AN APPROACH TO UNIFIED METHODOLOGY OF COMB. SW. CIRCUITS   E. Cerny   Ph. D.

AN APPROACH TO UNIFIED METHODOLOGY OF

COMBINATIONAL SWITCHING CIRCUITS

by

Eduard Cerny, M.Eng.

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy.

Department of Electrical Engineering,

McGill University,

Montreal, Quebec.

March 1975.

ABSTRACT

A methodology based on the theory of Boolean equations has been developed which permits a unified approach to the analysis and synthesis of combinational logic circuits. The type of circuits covered by the approach include both the classical loopless combinational networks as well as those which contain closed feedback loops and thus have internally a sequential character. To that end a general multiple output circuit represented by a Mealy-type machine is studied using characteristic equations (functions) that describe its internal structure. It is shown how behavioral properties of the circuit are reflected through the solutions of these equations. Moreover, it is demonstrated that a multiple output incompletely specified switching function is realized if a $\leq$ relation is satisfied between the corresponding characteristic functions. This leads to a new unified outlook on functional decomposition as used in modular synthesis procedures. Although the building modules are allowed to be sequential circuits, it is shown under which conditions the feedback loops are redundant with respect to the realization of a given output characteristic function, and thus the existence conditions of non-degenerate combinational circuits with loops are stated.

The power of the methodology of characteristic functions is then illustrated by showing that computational procedures based on that approach can easily be transcribed into operations on either maps or cubical complexes.

Moreover, the methodology is applied to derive procedures for modular synthesis of combinational circuits and for fault detection.

RESUME

Une méthodologie basée sur la théorie des équations Booléennes est developpée. Cette méthodologie permet une approche unifiée d'analyse et de synthèse de circuits logiques combinatoires. Les types de circuits traités par cette approche incluent le réseau classique de circuits combinatoires sans boucle ainsi que ceux qui possèdent des boucles de rétroaction (feedback loops) et qui ont un caractère interne séquentiel.

A cette fin, et un utilisant des équations caractéristiques (fonctions), nous étudions la structure interne d'une circuit général à sorties multiples représentée par une machine du type Mealy. En particulier, nous prouvons de quelle façon les propriétés de comportement du circuit se reflètent à travers les solutions de ces équations. De plus, nous prouvons qu'une fonction de commutation non-complètement spécifiée est réalisable si une relation $\leq$ existe entre les fonctions caractéristiques correspondantes. Cette relation nous permet de considérer d'une façon nouvelle et unifiée la décomposition fonctionnelle telle qu'utilisée dans les procédures de synthèse modulaire. Quoique les modules de construction du circuit peuvent être de nature séquentielle, nous déterminons dans quelles conditions les boucles de rétroaction sont redondantes par rapport à la réalisation d'une fonction caractéristique de sortie. Conséquemment, les conditions d'existence de circuits combinatoires avec boucles sont établies.

La puissance de la méthodologie des fonctions caractéristiques est illustrée par les procédures de calculs basés sur cette approche qui peuvent être aussi facilement effectués sur des diagrammes standards ou sur des complexes cubiques. De plus, la méthodologie est appliquée à la dérivation des procédures de synthèse modulaire de circuits combinatoires et à la découverte d'erreurs.

ACKNOWLEDGMENTS

The author would like to express his deep appreciation for the guidance provided throughout the project by Dr. M.A. Marin. He is also indebted to Dr. E.L. Sigurdson for his helpful critical comments, to Dr. S.J. Kubina for his ability to give strength and faith, and to Mrs. Louise McNeil for careful typing of the manuscript.

Thanks should also go to the staff of the Faculty of Engineering, Loyola College, which created a suitable atmosphere for undertaking this kind of work.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Historical Background

The application of Boolean algebra to the design and analysis of switching networks is considered to be an indivisible part of the theory of switching circuits since the founding work by Shannon [41]. The structure and properties of Boolean functions have been studied in order to design more reliable, economical and faster digital circuits and at the same time technological advancements and mere curiosity were posing still new problems to the switching theorists. The solutions to some of the problems gave rise to new theoretical tools and computational procedures whose only mutual link was that they were based on the laws of Boolean algebra. Thus the point of view obtained with each such theory was usually limited to a particular area, and it did provide little understanding of characteristics in common with other problems. At that point, the Boolean differential calculus [26, 34] was developed, and it was shown that it could be applied in a number of different areas and thus to form a common philosophy underlying the solutions. The solution steps could be described using formally simple formulae, but unfortunately the computations involved in solving actual problems are rather complex. Nevertheless, some practical results were obtained mainly in the area of fault detection [6, 26, 34]. Consequently, there is a need for some other approach which would be not only formally simple, but would also yield relatively easy computational algorithms.

The potential for the development of such a unifying methodology was seen by a number of authors in the theory of Boolean equations (B.E.), whose power was even compared with that of differential equations in electric circuit theory [17]. Enough mathematical background has been accumulated with respect to the properties and methods of solution of Boolean equations, as nicely summarized in the monograph by Rudeanu [26], but relatively few switching theorists know about these techniques and realize their power. Consequently, not too many practical applications of B.E. to switching circuit design have been discovered so far. Even then, however, the results obtained were of a rather topical character, without gaining any global insight into the relationship between the behaviour of switching circuits and the properties of Boolean equations. A uniform survey of the applications can be found in [26] and to a limited extent in [10]. Svoboda [31] devised a simple tabular method for solving B.E., and he proposed a hardware processor [39] which can be used to produce all elementary solutions of a given equation. The machine was simulated by Marin [19, 20] who at the same time developed some of its first practical applications to the synthesis of switching circuits. A summary and extensions of these techniques can be found in Klir [16, 18]. Brown [3] obtained a method for generating a reduced general solution to B.E. which, combined with some of the ideas from [19, 20], was used to produce an algorithmic procedure for modular synthesis of combinational switching circuits by decomposition [5]. A specialized case of synthesis using B.E. was also treated by Brown [4]. The properties and applications of sequential Boolean equations were studied in [8, 35].

The work to be presented here represents an effort to develop an understanding of the relationship between the theory of B.E. and the structure of combinational circuits. The result is intended to be a methodology which would permit a unified approach to the solution of problems related to the analysis and synthesis of combinational circuits.

The thesis should not be considered as a closed system of rules which would provide solutions to a fixed number of problems even though two direct applications will be shown. Rather, it is to provide a philosophical base, from which various types of switching theory problems could be tackled in a unified manner. Its development was initially stimulated by the fact that current methods used in the design and analysis of combinational circuits were unable to give :

(1) A unique point of view on modular synthesis by decomposition,

(2) A clear understanding of the behaviour of combinational circuits containing closed feedback loops,

(3) An approach to fault detection that would be consistent with circuit design procedures and simultaneously cover various types of faults under one methodology.

Therefore, in order to place the work into proper perspective, it is felt that before outlining the content of the subsequent chapters it is necessary to present an overview of past advances in the above mentioned areas.

The foundation for the synthesis of combinational switching functions by decomposition was laid down in the work by Ashenhurst [1], and later expanded by Curtis [40]. This method of synthesis proved to be suitable when logic networks were to be designed in terms of circuit modules which implement Boolean functions different from the basic AND, OR and NOT connectives. Therefore, an effort was made to state precise algorithmic procedures based on the theory of decomposition which could be applied to synthesize a given switching network using a fixed complete set of circuit module functions. Some of these procedures employed modified decomposition charts [13, 15, 29] as defined in [1, 40], others were based on algebraic techniques derived from the original decomposition theory as applied to a particular form of representing Boolean functions - the cubical complexes (arrays) [7, 21, 24, 25, 28, 36]. In general, the direction taken in these synthesis procedures was to start building the network from the problem input side, and then to progress towards the outputs by adding modules as guided by the decomposition theory and various circuit constraints. The process terminated when a network was obtained which realized the particular function, and which satisfied all the circuit constraints imposed by the designer. This direction should be contrasted with that used in the synthesis procedures based on Boolean equations [5, 16, 19, 20]. There, the decomposition proceeds from the problem output side by selecting a building module, and then by solving the corresponding B.E. a second level function is obtained, etc., until all free module inputs are satisfied directly by the problem input variables. The synthesis is terminated by the same conditions as before. Using the new methodology of this thesis, however, these two opposing approaches to modular synthesis can be unified to produce a rather flexible decomposition technique (Chapters 4 and 6).

Until recently, combinational circuits were constructed in such a way that they contained no closed feedback loops, characteristic to sequential machines. Also, the actual building modules used in synthesis procedures were purely combinational circuits. However, it has been pointed out [12, 14, 30] that some circuits having such closed loops could still have an overall combinational character, if only their stable output states were observed. Kautz [14] considered a particular cellular array structure, and he demonstrated that not only by closing the feedback loop would a combinational switching function be realized, but also that the number of gates required to synthesize some multiple output functions would be smaller than if done in the loopless conventional way. Even though some necessary conditions for the existence of loops in combinational circuits were stated, their scope was limited to that cellular array structure. In other words, the tools used for designing classical combinational networks are not directly usable in the synthesis of circuits with loops. As will be seen, however, the methodology developed here is general enough to be applicable to both types of combinational circuits.

Due to technological advances the circuits synthesized grew in complexity, and eventually they were placed on a single integrated circuit chip. Then, however, the problem of fault detection was encountered, for it was no longer possible to directly observe the inside activity of such complex monolithic modules. Nevertheless, in the case of combinational circuits, it would still be theoretically possible to perform exhaustive tests by applying all the possible combinations of input stimuli and simultaneously verifying that correct responses were received at the outputs. With an increasing number of inputs, though, the time needed to test all the combinations becomes rather prohibitive, and hence new techniques had to be developed. These took into account the internal structure of the circuit under test, and a necessary subset of the possible stimuli was determined so as to allow for detection of certain types of faults on the internal circuit lines [6, 9, 34, 37, 38, and many others]. Although the problem of determining these sets has been satisfactorily solved for combinational circuits, the methods used differ in approach from the techniques used in designing the networks, and thus they cannot be efficiently merged into synthesis algorithms. Moreover, there seems to be no simple general procedure which would cover various types of faults such as single stuck-at-(0, 1) and bridging, as well as multiplicity of those faults. Again, by applying the new methodology to this problem, a fault detection procedure has been obtained which is not only consistent with the synthesis method which will be presented in Chapter 6, but it also permits a unified treatment of the forementioned types of faults.

1.2 A Concise Outline of the Thesis

The presentation begins in Chapter 2 with a review and in some cases a deeper development of certain topics in the theory of Boolean equations, which are pertinent to the subject matter. An algebraic approach is stressed throughout, at the same time recognizing, however, the important visual information content of map techniques in cases where the number of variables involved is relatively small (See also Chapter 5). (In that sense, there is a certain disagreement with Rudeanu [26] who seems to be pressing for an algebraic approach only, vis a vis switching circuit applications). The concept of a characteristic equation is re-introduced in the sense that it is defined as a single equation of the form $\Phi = 1$, which is solution-equivalent to a given system of equations tied together by some prespecified mutual relation. Subsequently, the technique of characteristic equations (functions Φ) is fully employed in Chapter 3 to describe a general switching circuit with feedback, as represented by a Mealy-type machine. It is shown, that information about the steady and transient states of the circuit can be extracted from the corresponding characteristic equations, simply by studying their solution properties. The circuit's overall steady state output behaviour is then described through a circuit characteristic function Φ_C (equation $\Phi_C = 1$). Furthermore, the analysis of functional realization as it is performed in Chapter 4 leads to an important definition of an output characteristic function Φ. It is shown then, that a circuit realizes a particular multiple output incompletely specified switching function provided that the subsumming relation $\leqslant$ is satisfied between the corresponding circuit and output characteristic functions ($\Phi_C \leqslant \Phi$). This simple concept is the corner stone upon which the unification property of the methodology is based. A generalized decomposition theory for modular synthesis is then derived, which further clarifies the relationship between the internal structure of circuits and the properties of Boolean equations. Chapter 4 ends by presenting an inquiry into the necessity of closed loops in circuits realizing some output characteristic function.

Since any theory would have little meaning in engineering design unless it can be used to solve some practical problems, Chapter 5 is devoted to the development of computational tools, while two applications of the methodology are described in Chapter 6. It is demonstrated that the computational steps which underline the methodology of characteristic functions can easily be converted to either map manipulation procedures or to operations on cubical complexes (arrays). This ease of conversion is mainly due to the initial independence of the theory with respect to a particular data structure. An important point to note is that multiple output functions need not be treated in any special way, because the methodology permits a unified representation of any function by describing the functional mappings between Boolean spaces in terms of the corresponding "single output" characteristic functions.

Chapter 6 describes two applications. One deals with a unified approach to algorithmic synthesis of combinational switching functions by modular decomposition, and the other with the development of a technique for the detection of various types of faults in combinational networks (without feedback at the present time). The solution steps in both cases are unified by the $\leqslant$ relation between characteristic functions which signifies functional realization. Consequently, the procedure for generating fault detection test sets could be merged with the synthesis algorithm, thus sharing the same routines and data.

As a conclusion, a summary of the results obtained and plans for future research are presented in Chapter 7. It also brings to light an interesting relation between characteristic functions and function arrays.

CHAPTER 2

BOOLEAN EQUATIONS (B.E.)

To facilitate the analysis of switching circuits with the aid of Boolean Equations, some important definitions and theorems dealing with B.E. are presented. The scope of the presentation is limited only to those aspects of B.E. which are pertinent to the material in the subsequent chapters.

2.1 Notation

For dealing with two state switching elements the simplest Boolean Algebra $B_2 = \langle (0, 1), +, \cdot \rangle$ will be considered. The elements of B_2 then satisfy the standard axioms of Boolean Algebras [7, 11, 16, 21, 26]. Here are some important relations derived from those axioms:

Let $x, y, z \in B_2$ be elements of B.A., then

$$x + y = 0 \quad \leftrightarrow \quad x = y = 0 \tag{2.1}$$

$$x \cdot y = 1 \quad \leftrightarrow \quad x = y = 1 \tag{2.2}$$

$$x \leqslant x \quad \text{order relation} \tag{2.3}$$

$$x \leqslant y \quad \text{and} \quad y \leqslant x \quad \rightarrow \quad x = y \tag{2.4}$$

$$x \leqslant y \quad \text{and} \quad y \leqslant z \quad \rightarrow \quad x \leqslant z \tag{2.5}$$

$$x \leqslant x + y \quad ; \quad y \leqslant x + y \tag{2.6}$$

$$x \leqslant z \quad \text{and} \quad y \leqslant z \quad \rightarrow \quad x + y \leqslant z \tag{2.7}$$

$$x \leqslant y \quad \text{and} \quad x \leqslant z \qquad x \leqslant y\, z \tag{2.8}$$

$$x \leqslant y \quad \rightarrow \quad x + z \leqslant y + z \quad \text{and} \quad x\, z \leqslant y\, z \tag{2.9}$$

$$x \leqslant y \quad \leftrightarrow \quad x + y = y \quad \leftrightarrow \quad x\, y = x \tag{2.10}$$

$$x \leqslant y \quad \leftrightarrow \quad x\, \bar{y} = 0 \quad \leftrightarrow \quad \bar{x} + y = 1 \tag{2.11}$$

$$x = y \quad \leftrightarrow \quad x\, y + \bar{x}\, \bar{y} = 1 \quad \leftrightarrow \quad x\, \bar{y} + \bar{x}\, y = 0 \tag{2.12}$$

$$x \leqslant y \quad \rightarrow \quad \bar{y} \leqslant \bar{x} \tag{2.13}$$

$$0 \leqslant x \leqslant 1 \tag{2.14}$$

$$x + y = 1 \quad \leftrightarrow \quad x = 1 \quad \text{or} \quad y = 1 \tag{2.15}$$

$$x \cdot y = 0 \quad \leftrightarrow \quad x = 0 \quad \text{or} \quad y = 0 \tag{2.16}$$

The notation :

x, x_i, y_i — Boolean variables

$\underline{x} = (x_1, x_2, \ldots, x_r)$
$\underline{y} = (y_1, y_2, \ldots, y_q)$
$\underline{z} = (z_1, z_2, \ldots, z_n)$
— Vectors of variables (sets)

α_i, β, γ — Constants in B_2

$\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_r)$
$\underline{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$
— Vectors of constants (sets)

Φ , $\underline{f}$ — Scalar Boolean functions $B_2^m \rightarrow B_2$

$\underline{f}, \underline{g}$ — Vectors of functions $B_2^m \rightarrow B_2^n$

X, Y, Z — Sets of states of $\underline{x}$, $\underline{y}$, $\underline{z}$ respectively

$\emptyset$	Empty set
$\cup$, $\cap$	Set union, intersection
Σ	Boolean (+) summation
Π	Boolean ($\cdot$) product
$\Sigma\Pi$	Sum of product form
$x \oplus y$	Symmetric difference (exclusive "or")
$m_\alpha(\underline{x})$	Minterm function of $\underline{x}$ such that $m_\alpha(\underline{\beta}) = 1$ if $\underline{\alpha} = \underline{\beta}$ and 0 otherwise
$M_\alpha(\underline{x})$	Maxterm function of $\underline{x}$ such that $M_\alpha(\underline{\beta}) = 0$ if $\underline{\alpha} = \underline{\beta}$, 1 otherwise
$\sum_{\underline{\alpha} \in B_2^r} f(\underline{\alpha}) = f(\alpha_1) + f(\alpha_2) + \cdots$	Summation of $f(\underline{x})$ over all states of $\underline{x}$
$\prod_{\underline{\alpha} \in B_2^r} f(\underline{\alpha}) = f(\alpha_1) \cdot f(\alpha_2) \cdots$	Product of $f(\underline{x})$ over all states of $\underline{x}$
$\ulcorner \log_2(y)$	An integer such that $(\ulcorner \log_2(y)) - 1 < \log_2(y) \leq \ulcorner \log_2(y)$
R_i, R	Binary relation (=, $\leq$, $\geq$)
$\lvert\underline{y}\rvert$, $\lvert X\rvert$	The norm of $\underline{y}$ and X

2.2 Definitions and Theorems

Let $\underline{f}(\underline{x}, \underline{y})$, $\underline{g}(\underline{x}, \underline{y})$ be Boolean functions $\underline{f} = (f_1, f_2, \ldots, f_n)$, $\underline{g} = (g_1, g_2, \ldots, g_n)$ of the variables

$\underline{x} = (x_1, \ldots, x_r)$, $\underline{y} = (y_1, \ldots, y_q)$, that is, $\underline{f}, \underline{g} : B_2^{r+q} \to B_2^n$.
Furthermore, let $f_i \, R_i \, g_i$ be a binary relation which represents either

$$f_i = g_i$$

or

$$f_i \leqslant g_i \qquad\qquad i = 1, \ldots, n$$

or

$$f_i \leqslant g_i$$

Application of (2.11) and (2.12) to the above relations yields

$$f_i = g_i \quad \to \quad f_i \, g_i + \bar{f}_i \, \bar{g}_i = 1 \tag{2.17}$$

$$f_i \leqslant g_i \quad \to \quad \bar{f}_i + g_i = 1 \tag{2.18}$$

$$f_i \geqslant g_i \quad \to \quad f_i + \bar{g}_i = 1 \, . \tag{2.19}$$

The original system of relations can thus be transformed into an equivalent system of Boolean equalities of the form $\phi_i(\underline{x}, \underline{y}) = 1$, $i = 1, \ldots, n$, where ϕ_i stands for one of (2.17), (2.18) or (2.19).

<u>Definition 2.1</u>: Solution to B.E.

Let $\underline{f}(\underline{x}, \underline{y}) \;\; R \;\; \underline{g}(\underline{x}, \underline{y})$ be a system of n Boolean relations with its equivalent system of Boolean equations $\underline{\phi}(\underline{x}, \underline{y}) = 1$, as derived above. Then a function $\underline{y} = \underline{y}(\underline{x})$ is a solution to this system if $\underline{f}(\underline{x}, \underline{y}(\underline{x}))$ R $\underline{g}(\underline{x}, \underline{y}(\underline{x}))$ or equivalently $\phi(\underline{x}, \underline{y}(\underline{x})) = 1$ holds identically.

Definition 2.2 :

Characteristic function $\Phi(\underline{x}, \underline{y})$ [10, 26] of the system of n simultaneous equations $\phi_i(\underline{x}, \underline{y}) = 1$, $i = 1, \ldots, n$, (Def. 2.1) is defined as $\Phi(\underline{x}, \underline{y}) = \prod_{i=1}^{n} \phi_i(\underline{x}, \underline{y})$.

Lemma 2.1 : [10, 16, 26, 31]

The characteristic equation $\Phi(\underline{x}, \underline{y}) = 1$ is equivalent to the system $\underline{f}(\underline{x}, \underline{y})$ R $\underline{g}(\underline{x}, \underline{y})$. That is, a function $\underline{y} = \underline{y}(\underline{x})$ is a solution to $\underline{f}$ R $\underline{g}$ iff it is a solution to $\Phi = 1$.

Remark : The characteristic equation $\psi(\underline{x}, \underline{y}) = \bar{\Phi}(\underline{x}, \underline{y}) = 0$ could have been chosen instead; however, the form $\Phi = 1$ seems to be more convenient for describing switching circuits, as will be seen later.

Definition 2.3 : Consistency.

The equation $\Phi(\underline{x}, \underline{y}) = 1$ is said to be consistent if it has a solution $\underline{y} = \underline{y}(\underline{x})$ over all $\underline{x} \in B_2^r$.

Theorem 2.1 : [10, 26]

The equation $\Phi(\underline{x}, \underline{y}) = 1$ is consistent iff

$$c(\underline{x}) = \sum_{\underline{\beta} \in B_2^q} \Phi(\underline{x}, \underline{\beta}) = 1 \text{ identically, or equivalently } \bar{c}(\underline{x}) = 0.$$

<u>Remark</u> : If $c(\underline{x}) = 1$ does not hold for all $\underline{x} \in B_2^r$, the equation $\Phi = 1$ is constrained by $c(\underline{x}) = 1$, and it has a solution only for such $\underline{x} \in B_2^r$ which satisfy $c(\underline{x}) = 1$.

<u>Definition 2.4</u> : Don't care states.

Let $X' \subseteq B_2^r$ define the states of $\underline{x}$ such that the solutions $\underline{y} = \underline{y}(\underline{x})$ to $\Psi(\underline{x}, \underline{y}) = 1$ can have a "don't care" value for $\underline{x} \in X'$. That is, $\underline{x} \in X' \rightarrow \underline{y}$ can assume any value in B_2^q. Let then a function $d(\underline{x})$ be defined as $d(\underline{x}) = \sum_{\underline{\alpha} \in X'} m_{\alpha}(\underline{x})$.

<u>Theorem 2.2</u> : [5]

The characteristic function of the equation $\Psi(\underline{x}, \underline{y}) = 1$ with d.c. states defined by $d(\underline{x}) = 1$ is given as $\Phi(\underline{x}, \underline{y}) = \Psi(\underline{x}, \underline{y}) + d(\underline{x})$.

<u>Remark</u> : If the consistency condition $c(\underline{x}) = 1$ is not satisfied identically, and the input states are restricted (constrained) to $\underline{x} \in X$ where $X = \{\underline{x} \mid c(\underline{x}) = 1\}$, then in effect the states in $X' = B_2^r - X$ become don't care in the sense of Definition 2.4. Hence the characteristic function of the constrained equation becomes $\Phi(\underline{x}, \underline{y}) + \bar{c}(\underline{x})$ (Theorem 2.2).

Thus the resulting characteristic equation becomes consistent. Moreover, if a system of B.E. is not consistent, and don't care states are introduced so that $d(\underline{x}) \geqslant \bar{c}(\underline{x})$, then through Theorem 2.2 the system is again consistent.

The following section will describe some of the methods for obtaining the solutions $\underline{y} = \underline{y}(\underline{x})$ of $\Phi(\underline{x}, \underline{y}) = 1$.

2.3 Methods of Solution

Once a problem has been formulated in the form of an equivalent Boolean equation, then it is usually desired to obtain the functions $\underline{y}(\underline{x})$ which satisfy the equation. A number of methods will be presented here with a concise evaluation regarding their applicability.

Definition 2.5 : Types of solutions.

An elementary solution to $\Phi(\underline{x}, \underline{y}) = 1$ is a vector function $\underline{y}(\underline{x})$ such that $\Phi(\underline{x}, \underline{y}(\underline{x})) = 1$ identically.

A general solution $\underline{n}(\underline{x}, \underline{p})$ is a vector function such that for every value $\underline{p}^*$ of the parameter vector $\underline{p}$ the function $\underline{n}(\underline{x}, \underline{p}^*)$ is an elementary solution to $\Phi = 1$. Also, given any elementary solution $\underline{y}(\underline{x})$ there exists a valuation $\underline{p}^*$ of $\underline{p}$ such that $\underline{y}(\underline{x}) = \underline{\eta}(\underline{x}, \underline{p}^*)$.

2.3.1 The Method of Successive Elimination [10, 26]

Consider the equation $y\, a + \bar{y}\, b = 1$ of one unknown y. The equation is consistent if the relation $a + b = 1$ holds identically. The general solution is then given by

$$y = \bar{b} + a\, p, \quad a, b \in B,$$

where

$$p \text{ is an arbitrary parameter}, \quad p \in B .$$

An elementary solution may be obtained by substitution of a value $p^* \in B$ for p.

The equation $\Phi(\underline{x}, \underline{y}) = 1$ is then solved by a repeated application of the above procedure. Namely, by expanding $\Phi(\underline{x}, \underline{y})$ about y_q, the equation can be written as

$$\Phi(\underline{x}, y_1, y_2, \ldots, y_{q-1}, 1)\, y_q + \Phi(\underline{x}, y_1, y_2, \ldots, y_{q-1}, 0)\, \bar{y}_q = 1$$

with B being the algebra of all functions $B_2^{\,r} \to B_2^{\,q}$, (2.20)

and it is consistent for y_q if

$$\Phi(\underline{x}, y_1, y_2, \ldots, y_{q-1}, 1) + \Phi(\underline{x}, y_1, y_2, \ldots, y_{q-1}, 0) = 1 \tag{2.21}$$

The same procedure is now applied to the Equation (2.21) of $q - 1$ unknowns, and another consistency condition obtained, etc., until only the variable y_1 remains. The consistency condition thus becomes

$$\sum_{\underline{\beta} \in B_2^q} \Phi(\underline{x}, \underline{\beta}) = 1 \qquad (2.22)$$

which is identical to the result of Theorem 2.1. If (2.22) holds, the general solution for y_1 is $y_1 = \bar{b} + a\, p_1$, where a and b are determined through the elimination process just described. It can be shown that

$$a = \sum_{(\alpha_2, \ldots, \alpha_q) \in B_2^{q-1}} \Phi(\underline{x}, 1, \alpha_2, \ldots, \alpha_q),$$

and

$$b = \sum_{(\alpha_2, \ldots, \alpha_q) \in B_2^{q-1}} \Phi(\underline{x}, 0, \alpha_2, \ldots, \alpha_q).$$

The function $y_1(\underline{x}, p_1)$ is back substituted into the equation for y_2 and a general solution $y_2 = y_2(\underline{x}, p_1, p_2)$ obtained. Then y_1, y_2 substituted into the equation for y_3, etc., until $y_q = y_q(\underline{x}, p_1, \ldots, p_q)$ is generated. That is, the general solution

$$\underline{y} = \underline{\eta}(\underline{x}, \underline{p})$$

where

$$\underline{p} = (p_1, \ldots, p_q)$$

is a vector of arbitrary parameters ranging over all functions $\underline{p} : B_2^r \to B_2^q$.

If an elementary solution is desired the procedure remains the same, except an elementary solution $y_1 = y_1(\underline{x})$ is chosen by substituting a value for p_1, then for p_2 at the second step of back substitution (for y_2), etc. Finally, the solution $\underline{y} = \underline{y}(\underline{x})$ is obtained. Furthermore, since B_2^q is finite, all elementary solutions can be generated by a tree-like structure. (For large number of variables this may be quite time consuming). Also, it should be noted that the parameter vector $\underline{p}$ may be redundant in the sense that different values of $\underline{p}$ may produce the same elementary solution. For a brief discussion of irredundant encodings of the parameters see the end of Section 2.3.2.

2.3.2 Methods Based on Canonical Forms of Φ [3, 10, 16, 18, 26, 31]

The first method to be described produces elementary solutions of $\Phi(\underline{x}, \underline{y}) = 1$. The solutions can very easily be extracted when the function $\Phi(\underline{x}, \underline{y})$ is represented as a map (Marquand or other) [16, 26, 31].

Let the equation $\Phi = 1$ be given in its disjunctive canonical form

$$\sum_{\underline{\alpha} \in B_2^r} \left(m_{\alpha}(\underline{x}) \cdot \sum_{\underline{\beta} \in S(\underline{\alpha})} m_{\beta}(\underline{y})\right) = 1 \qquad (2.23)$$

where

$$S(\underline{\alpha}) \subseteq B_2^q \quad \text{for each } \underline{\alpha} \in B_2^r .$$

A solution $\underline{y} = \underline{y}(\underline{x})$ is obtained by a correspondence relating each $\underline{\alpha} \in B_2^r$ with a vector $\underline{y}(\underline{\alpha})$ such that $\Phi(\underline{\alpha}, \underline{y}(\underline{\alpha})) = 1$ identically. The Equation (2.23) then reduces to $\sum_{\underline{\beta} \in S(\underline{\alpha})} m_{\underline{y}(\underline{\alpha})}(\underline{\beta}) = 1$. This equality to hold, $\underline{y}(\underline{\alpha})$ has to be chosen as one of the vectors from $S(\underline{\alpha})$, for each $\underline{\alpha} \in B_2^r$. The total number of distinct solutions is thus given as $n_s = \prod_{\underline{\alpha} \in B_2^r} |S(\underline{\alpha})|$ (Π stands for arithmetic multiplication here). Clearly, the Equation (2.23) is consistent when $\forall \underline{\alpha}$, $S(\underline{\alpha}) \neq \emptyset$, which corresponds to Theorem 2.1.

The above method as translated into the form of map manipulation [16, 26, 31] :

Let $\Phi(\underline{x}, \underline{y})$ be expressed in the form of a Marquand map, the states of $\underline{x}$ labeling the columns (2^r of them) and the states of $\underline{y}$ labeling the rows (2^q) of the map. A 1 is placed in the square with coordinates $(\underline{x}^*, \underline{y}^*)$ if $\Phi(\underline{x}^*, \underline{y}^*) = 1$, the square is marked 0 otherwise. The resulting map is the discriminant D of the equation [31]. Let $\underline{d}(\underline{\alpha})$ be a column vector of the map D associated with the state $\underline{x}$. Then the 1's in $\underline{d}(\underline{\alpha})$ define the set $S(\underline{\alpha})$ through the associated states of $\underline{y}$. Let the number of 1's be n_{α}, $n_{\alpha} = |S(\underline{\alpha})|$, then $n_s = \prod_{\underline{\alpha} \in B_2^r} n_{\alpha}$.

The elementary solutions are obtained by decomposing the discriminant D into its sub-maps D_j, $j = 1, \ldots, n_s$, such that each column of the sub-maps has exactly one 1, and $D_j \leq D$, $D_i \neq D_j$ for $i \neq j$. Now the states of $\underline{y}$ associated with each 1 in D_j determine the elementary solution $\underline{y}^j = \underline{y}^j(\underline{x})$, $j = 1, \ldots, n_s$, of the original equation.

Given a complete set of elementary solutions to $\Phi(\underline{x}, \underline{y}) = 1$, a general solution $\underline{\eta}(\underline{x}, \underline{p})$ may be obtained by introducing a set of orthonormal functions of some parameters $\underline{p}$ with each set $S(\underline{\alpha})$. If a particular value $\underline{p}^*(\underline{x})$ for $\underline{p}$ is assigned then a single state $\underline{\beta} \in S(\underline{\alpha})$, $\underline{x} = \underline{\alpha}$ is identified, and hence an elementary solution $\underline{y} = \underline{y}(\underline{x}) = \underline{\eta}(\underline{x}, \underline{p}^*(\underline{x}))$ generated. The general solution $\underline{\eta}$ is similar to that obtained by the elimination method. Also the number of parameters $|\underline{p}| = q$ may be the same. However, by proper (economic) encoding of the orthonornal functions the number of parameters may in some cases be reduced. (E.g. if $\max\limits_{\underline{\alpha} \in B_2^r} (n_\alpha) = 2^q/2$ then only $q - 1$ parameters may be needed.) [3, 26]

As an example of such an economic encoding the method of Brown [3] will be shown here. As a matter of fact, that method was programmed for a computer and used in a combinational circuit synthesis program [5].

Assume $\Phi(\underline{x}, \underline{y}) = 1$ consistent. Let $\underline{p}$ be a vector of t parameters (t defined later). The (reduced) general solution in terms of $\underline{x}$ and $\underline{p}$ is given using Boolean matrix operations [3, 26] as:

$$\underline{y} = \underline{n}(\underline{x}, \underline{p}) = K_q \cdot A(\underline{p}) \cdot \underline{m}(\underline{x})$$

where

$$|\underline{y}| = q,$$

$\underline{m}(\underline{x})$ is the minterm vector, $|\underline{m}(\underline{x})| = 2^r$.

The number of parameters t is determined as follows:

Let $\underline{d}_j$ be the j^{th} column of the discriminant D (map considered as a matrix), and let s_j be the number of 1's in $\underline{d}_j$; $s = \max_j (s_j)$, $j = 0, 1, \ldots, 2^{r-1}$.

Then $2^{t-1} < s \leq 2^t$.

$A(\underline{p})$ is a matrix of 2^q rows and 2^r columns obtained from D as follows:

Let $\underline{a}_j$ be the j^{th} column vector of A. If $s_j = 1$ or 0 then $\underline{a}_j = \underline{d}_j$, else if $s_j > 1$ then form a set O_j of s_j orthonormal functions of the t parameters $\underline{p}$.

$$O_j = \{ \zeta_{j1}(\underline{p}), \ldots, \zeta_{js_j}(\underline{p}) \}$$

with

$$\zeta_{ji} \cdot \zeta_{jk} = 0 \qquad \text{if } i \neq k$$

$$\sum_{i=1}^{sj} \zeta_{ji} = 1 \qquad \text{being the orthonormality condition.}$$

Form now $\underline{a}_j$ by the following algorithm:

$$
\begin{array}{l}
k = 1 \; ; \quad \textbf{for} \quad m = 1 \quad \text{to} \quad 2^r \quad \text{do} \\
\qquad \text{if} \quad d_{mj} = 0 \quad \text{then} \quad a_{mj} = 0 \\
\qquad \qquad \qquad \qquad \quad \text{else} \quad a_{mj} = \zeta_{jk}(\underline{p}) ; \\
\qquad k = k + 1 \; ; \\
\qquad \text{end} \quad m \; ;
\end{array}
$$

The decomposition matrix is defined recursively as:

$$
\begin{aligned}
K_1 &= (0 \quad 1) \\
K_2 &= \left[\begin{array}{cc|cc} \multicolumn{2}{c|}{k_1} & \multicolumn{2}{c}{k_1} \\ 0 & 0 & 1 & 1 \end{array}\right] \\
&\ \vdots \\
K_{i+1} &= \left[\begin{array}{c|c} k_i & k_i \\ 0 \ldots 0 & 1 \ldots 1 \end{array}\right]
\end{aligned}
$$

For more detail and examples see Reference [3, 5].

2.3.3 Application of Löwenheim's Theorem [26, 27]

Consider a consistent equation $\Phi(\underline{x}, \underline{y}) = 1$ (If not consistent then apply Theorem 2.2 or as in [26, 27].) Löwenheim's theorem states:

Given a consistent B.E. $\Psi(\underline{y}) = 1$ in an arbitrary Boolean algebra B, then a general solution has the form

$$\underline{\eta}(\underline{p}) = \bar{\Psi}(\underline{p}) \cdot \underline{h} + \Psi(\underline{p}) \cdot \underline{p}$$

where

$\underline{h}$ is an elementary solution, $\Psi(\underline{h}) = 1$, and

$\underline{p}$ is an arbitrary parameter over B^q.

According to [27], if B is considered to be the algebra of all functions of $\underline{x}$ then the general solution to $\Phi(\underline{x}, \underline{y}) = 1$ is given as:

$$\underline{\eta}(\underline{x}, \underline{p}) = \bar{\Phi}(\underline{x}, \underline{p}(\underline{x})) \cdot \underline{y}(\underline{x}) + \Phi(\underline{x}, \underline{p}(\underline{x})) \cdot \underline{p}(\underline{x}) \qquad (2.24)$$

where

$\underline{y}(\underline{x})$ is any elementary solution of $\Phi = 1$,

$\underline{p}(\underline{x})$ is an arbitrary vector over all functions

$B_2^r \to B_2^q$, $|\underline{p}| = q$.

If this approach (L) to the generation of $\underline{\eta}(\underline{x}, \underline{p})$ is compared with the method of successive eliminations (SE), and with the methods similar to Brown's (B), then:

(L) Requires knowledge of an elementary solution, thereafter the procedure for obtaining $\underline{\eta}(\underline{x}, \underline{p})$ is very simple, however $|\underline{p}| = q$.

(SE) No knowledge of an elementary solution is required; however, algebraic manipulation is more complicated than in (L). $|\underline{p}| = q$.

(B) No explicit knowledge of an elementary solution required, $|\underline{p}|$ can also be smaller than q in some cases. It requires the formation of D and the associated sets of orthonormal functions. (Actually all elementary solutions are known implicitly through D, their enumeration must be made to determine t.) A higher number of variables requires large maps (matrices) D and A - the dimensions being $2^r \times 2^q$.

Comparison of the methods generating elementary solutions, namely, Svoboda's method (S)[17, 26, 31] and that of successive elimination (SE), yields:

(S) Construction of D as in (B), very simple for small number of variables, hyperplanes of solutions can be obtained [17, 18] . A hardware processor has been designed and simulated [16, 19, 20]. (The method could be transcribed into algebraic notation rather than that of maps.)

(SE) The method is algebraic in nature, no large maps required. The number of steps in the procedure is smaller than in (S); however, hyperplanes of solutions are more difficult to identify.

2.3.4 Some Other Methods. [26]

If the system of relations, $f_i \; R_i \; g_i$, $i = 1, \ldots, n$, can be transformed into a set of linear equations

$$\sum_{j=1}^{q} a_{ij} \, y_j = b_i \, , \qquad i = 1, \ldots, n \, ,$$

where a_{ij}, b_i may be functions of $\underline{x}$, then the problem can be expressed in the form of a linear Boolean matrix equation $AX = B$. The solution(s) may thus be obtained by applying the theory of matrix equations. However, for applications in switching circuit design, this approach does not have much value, since the problem can seldom be simply stated in the form of a system of linear equations.

The equation $\Phi(\underline{x}, \underline{y}) = 1$ may also be solved by the method of undetermined coefficients. However, $q \cdot 2^r$ unknown coefficients c_i, $i = 1, \ldots, q \cdot 2^r$, must be introduced through which the original equation is transformed into a system of 2^r truth equations $e_i(\underline{c}) = 1$. Solutions to this system then yield the elementary solutions

$\underline{y}(\underline{x})$, by a back substitution of the coefficient values. Although simple, the method seems to complicate the problem by introducing $q \cdot 2^r$ unknowns instead of the q original ones. Therefore, except for problems with small r and q, the metnod has a very little practical value.

2.4 Some Properties of Solutions of B.E.

The material to be presented here is mostly original and will be used quite extensively in the subsequent chapters. Assume $\Phi(\underline{x}, \underline{y}) = 1$ consistent.

Lemma 2.2: Identity solution.

The equation $\Phi(\underline{x}, \underline{y}) = 1$ has an elementary solution(s) such that some $y_i \in \underline{y}$ is an identity ($y_i = 1$ or $y_i = 0$) iff the equation

$$\Phi(\underline{x}, y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_q) = 1 \tag{2.25}$$

or

$$\Phi(\underline{x}, y_1, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_q) = 1$$

is consistent.

Proof: Trivial.

To determine whether the equation $\Phi(\underline{x}, \underline{y}) = 1$ has an identity solution for some subset of the variables $\underline{y}$, the above Lemma 2.2 can be applied repeatedly with respect to the variables in the subset. If, however, an identity solution for all $y_i \in \underline{y}$ is sought (simultaneously) then a more compact procedure is stated in the following theorem.

Theorem 2.3: Identity in $\underline{y}$.

The equation $\Phi(\underline{x}, \underline{y}) = 1$ has an identity solution $\underline{y}^* \in B_2^q$ if the equation

$$\Psi(\underline{x}, \underline{y}) = \bar{\Phi}(\underline{x}, \underline{y}) = 1$$

is inconsistent with respect to the inverse solutions $\underline{x} = \underline{x}(\underline{y})$. The identity solution(s) $\underline{y}^*$ is (are) then obtained as a solution of the equation

$$c(\underline{y}) = \sum_{\underline{\alpha} \in B_2^r} \Psi(\underline{\alpha}, \underline{y}) = 0 \tag{2.26}$$

or equivalently

$$\bar{c}(\underline{y}) = \prod_{\underline{\alpha} \in B_2^r} \Phi(\underline{\alpha}, \underline{y}) = 1 \tag{2.27}$$

Proof:

Let $\underline{y}^* = \underline{\beta} \in B_2^q$ be an identity solution, then $\Phi(\underline{x}, \underline{\beta}) = 1$ holds independently of $\underline{x}$. $\underline{\beta}$ is also the unique solution to

$$\sum_{\underline{\alpha} \in B_2^r} m_\alpha(\underline{x}) \cdot m_\beta(\underline{y}) = 1 \tag{2.28}$$

hence

$$\sum_{\alpha \in B_2^r} m_\alpha(\underline{x}) \cdot m_\beta(\underline{y}) \leq \Phi(\underline{x}, \underline{y}) . \tag{2.29}$$

Consistency condition of (2.28) w.r.t. $\underline{x} = \underline{x}(\underline{y})$ is

$$\sum_{\underline{Y} \in B_2^r} \overline{\sum_{\underline{\alpha} \in B_2^r} m_\alpha(\underline{Y})\, m_\beta(\underline{y})}$$

$$= \bar{m}_\beta(\underline{y}) + \sum_{\underline{Y} \in B_2^r} \sum_{\underline{\alpha} \in B_2^r} \overline{m_\alpha}(Y)$$

$$= \bar{m}_\beta(\underline{y})$$

which is not identically equal to 1, hence (2.28) is inconsistent.

Negation of (2.29) and a summation over all $\underline{Y} \in B_2^r$ yields

$$\bar{m}_\beta(\underline{y}) \geq \sum_{\underline{Y} \in B_2^r} \bar{\Phi}(\underline{Y}, \underline{y})$$

That is,

$$\underset{\sim}{c}(\underline{y}) = \sum_{\underline{\alpha} \in B_2^r} \Psi(\underline{\alpha}, \underline{y}) \leq \bar{m}_\beta(\underline{y}),$$

meaning that $\Psi(\underline{x}, \underline{y}) = 1$ is inconsistent. Also, $c(\underline{\beta}) \leq \bar{m}_{\beta}(\underline{\beta}) = 0$; therefore, $\underline{\beta}$ is a solution to (2.26).

Consider now $\Psi(\underline{x}, \underline{y}) = 1$ to be inconsistent with respect to $\hat{\underline{x}} = \underline{x}(\underline{y})$. Let $\underline{\beta}_1, \ldots, \underline{\beta}_t$ be the solutions to $c(\underline{y}) = 0$. Then

$$\sum_{\underline{\alpha} \in B_2^r} \Psi(\underline{\alpha}, \underline{\beta}_i) = 0 \qquad \text{for all } i = 1, \ldots, k.$$

That is,

$$\sum_{\underline{\alpha} \in B_2^r} \overline{\Phi}(\underline{\alpha}, \underline{\beta}_i) = 0 \;,$$

and thus

$$\overline{\Phi}(\underline{x}, \underline{\beta}_i) = 0 \qquad \text{identically for all } \underline{x} \in B_2^r \text{ and } i = 1, \ldots, k.$$

Or

$$\Phi(\underline{x}, \underline{\beta}_i) = 1 \qquad \text{by negating the above equality.}$$

Q.E.D.

Remark:

The existence of an identity solution means that the variables $\underline{x}$ are redundant in some solutions.

Another important property of a consistent equation $\Phi(\underline{x}, \underline{y}) = 1$ is that of having a unique elementary solution. This property can easily be verified by analyzing the discriminant map D - a unique solution exists if each column of D contains a single 1. However, for algebraic processing a different approach is needed, which is stated in the following theorem and its corollary.

<u>Theorem 2.4</u> : Unique solution.

The equation $\Phi(\underline{x}, \underline{y}) = 1$ has a unique solution for a $y_i \in \underline{y}$ iff

$$\sum_{(\alpha_1,\dots,\alpha_{i-1},\dots,\alpha_q)\in B_2^{q-1}} \Phi(\underline{x}, \alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_q) \cdot \sum_{(\alpha_1,\dots,\alpha_{i-1},\alpha_{i+1},\dots,\alpha_q)\in B_2^{q-1}} \Phi(\underline{x}, \alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_q) = 0 \qquad (2.30)$$

identically.

<u>Proof</u>:

Apply the method of successive eliminations (Section 2.3.1) to solve the equation, with y_i being eliminated at the last step before subsumming. This step can be written as

$$\bar{y}_i\, b(\underline{x}) + y_i\, a(\underline{x}) = 1$$

where

$$b(\underline{x}) = \sum_{(\alpha_1,\dots,\alpha_{i-1},\alpha_{i+1},\dots,\alpha_q)\in B_2^{q-1}} \Phi(\underline{x}, \alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_q)$$

and

$$a(\underline{x}) = \sum_{(\alpha_1,\ldots,\alpha_{i-1},\alpha_{i+1},\ldots,\alpha_q)\in B_2^{q-1}} \Phi(\underline{x}, \alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_q)$$

The general solution is then $y_i = \bar{b}(\underline{x}) + a(\underline{x})\, p_i$. There is a unique elementary solution $y_i(\underline{x})$ if the general solution is independent of p_i, that is, when $\bar{b}(\underline{x}) \geqslant a(\underline{x})$. Or equivalently (2.11) as $b(\underline{x}) \cdot a(\underline{x}) = 0$.

Q.E.D.

Corollary:

There is a unique solution $\underline{y}(\underline{x})$ to $\Phi(\underline{x}, \underline{y}) = 1$ if

$$\sum_{i=1}^{q} b_i(\underline{x}) \cdot a_i(\underline{x}) = 0 \quad \text{, identically,}$$

where a_i and b_i are the same as in Theorem 2.4.

Proof: Trivial.

The following discussion will now cover a property of B.E. which is of importance for the definition of circuit characteristic function as introduced in Chapters 3 and 4.

Consider the following problem: Let $\Phi_1(\underline{x}, \underline{z}) = 1$ and $\Phi_2(\underline{x}, \underline{z}, \underline{y}) = 1$ be two consistent equations with respect to the solutions $\underline{z} = \underline{z}(\underline{x})$ and $\underline{y} = \underline{y}(\underline{x}, \underline{z})$ respectively, where $|\underline{x}| = r$, $|\underline{z}| = n$, $|\underline{y}| = q$. It is desired to determine all the functions $\underline{y} = \underline{y}(\underline{x})$ generated by the solutions $\underline{z}(\underline{x})$ of $\Phi_1(\underline{x}, \underline{z}) = 1$ through the second equation $\Phi_2(\underline{x}, \underline{z}(\underline{x}), \underline{y}) = 1$.

An approach to this problem would be to solve $\Phi_1 = 1$ for all $\underline{z}(\underline{x})$, substitute all these elementary solutions for $\underline{z}$ in $\Phi_2 = 1$, and then solve the resulting equation for $\underline{y}(\underline{x})$. However, a more compact method is obtained by first finding the characteristic function (equation) of the system of two equations $\Phi_1 = 1$, $\Phi_2 = 1$ related as stated above. That is, to find an equation $\Phi(\underline{x}, \underline{y}) = 1$ such that its solutions are precisely the solutions to the original problem.

Lemma 2.3 :

The characteristic equation $\Phi(\underline{x}, \underline{y}) = 1$ (function Φ) of a system of two consistent equations $\Phi_1(\underline{x}, \underline{z}) = 1$ and $\Phi_2(\underline{x}, \underline{z}, \underline{y}) = 1$ such that the solutions $\underline{y}(\underline{x})$ of $\Phi = 1$ are equivalent to the solutions of $\Phi_2(\underline{x}, \underline{z}(\underline{x}), \underline{y}) = 1$ with $\Phi_1(\underline{x}, \underline{z}(\underline{x})) = 1$ is given by

$$\Phi(\underline{x}, \underline{y}) = \sum_{\underline{Y} \in B_2^n} (\Phi_1(\underline{x}, \underline{Y}) \cdot \Phi_2(\underline{x}, \underline{Y}, \underline{y})) \;\cancel{=}\; 1 \qquad (2.31)$$

Proof :

The consistency condition: $\Phi_1(\underline{x}, \underline{z}) = 1$ has at least one solution $\underline{z}(\underline{x})$; $\Phi_2(\underline{x}, \underline{z}, \underline{y}) = 1$ has at least a single solution $\underline{y}(\underline{x}, \underline{z})$. Hence the canonical forms of Φ_1 and Φ_2 can be expressed as

$$\Phi_1(\underline{x}, \underline{z}) = \sum_{\underline{\alpha} \in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\underline{\gamma} \in S_\alpha} m_\gamma(\underline{z}))$$

where

$$S_\alpha = \{\underline{\gamma} \mid \Phi_1(\underline{\alpha}, \underline{\gamma}) = 1\} \neq \emptyset ,$$

$$\Phi_2(\underline{x}, \underline{z}, \underline{y}) = \sum_{\underline{\alpha} \in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\underline{\gamma} \in B_2^n} (m_\gamma(\underline{z}) \cdot \sum_{\underline{\beta} \in S_{\alpha\gamma}} m_\beta(\underline{y})))$$

where

$$S_{\alpha\gamma} = \{\underline{\beta} \mid \Phi_2(\underline{\alpha}, \underline{\gamma}, \underline{\beta}) = 1\} \neq \emptyset.$$

By (2.31)

$$\Phi(\underline{x}, \underline{y}) = \sum_{\underline{\gamma} \in B_2^n} \{ \sum_{\underline{\alpha} \in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\underline{\gamma} \in S_\alpha} (m_\gamma(\underline{z}) \cdot \sum_{\underline{\beta} \in S_{\alpha\gamma}} m_\beta(\underline{y})))\}$$

$$= \sum_{\underline{\alpha} \in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\substack{\underline{\gamma} \in S_\alpha \\ \underline{\beta} \in S_{\alpha\gamma}}} m_\beta(\underline{y})) = 1 \tag{2.32}$$

(a) Show that for any solution $z(x)$ of $\Phi_1 = 1$ the solutions $\underline{y}(\underline{x})$ of $\Phi_2(\underline{x}, \underline{z}(\underline{x}), \underline{y}) = 1$ are also solutions of $\Phi(\underline{x}, \underline{y}) = 1$:

$$\Phi_2(\underline{x}, \underline{z}(\underline{x}), \underline{y}) = \sum_{\underline{\alpha}\in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\underline{\gamma}\in B_2^n} (m_\gamma(\underline{z}(\underline{\alpha})) \cdot \sum_{\underline{\beta}\in S_{\alpha\gamma}} m_\beta(\underline{y})))$$

$$\leqslant \sum_{\underline{\alpha}\in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\substack{\underline{\gamma}\in S_\alpha \\ \underline{\beta}\in S_{\alpha\gamma}}} m_\beta(\underline{y})) = \Phi(\underline{x}, \underline{y})$$

Therefore, any $\underline{y}(\underline{x})$ solution of $\Phi_2(\underline{x}, \underline{z}(\underline{x}), \underline{y}) = 1$ is a solution of $\Phi(\underline{x}, \underline{y}) = 1$ as well.

(b) The converse - for any solution $\underline{y}(\underline{x})$ of $\Phi(\underline{x}, \underline{y}) = 1$ there exists a function $\underline{z}(\underline{x})$ such that $\Phi_2(\underline{x}, \underline{z}(\underline{x}), \underline{y}(\underline{x})) = 1$ and $\Phi_1(\underline{x}, \underline{z}(\underline{x})) = 1$ identically :

Let $\underline{y}(\underline{x})$ be a solution of (2.31). Then

$$\Phi_2(\underline{x}, \underline{z}, \underline{y}(\underline{x})) = \sum_{\underline{\alpha}\in B_2^r} (m_\alpha(\underline{x}) \cdot \sum_{\underline{\gamma}\in B_2^n} (m_\gamma(\underline{z}) \cdot \sum_{\underline{\beta}\in S_{\alpha\gamma}} m_\beta(\underline{y}(\underline{\alpha}))))$$

$$= \sum_{\underline{\alpha}\in B_2^r} m_\alpha(\underline{x}) \cdot (\sum_{\underline{\gamma}\in S'_\alpha} m_\gamma(\underline{z})) ,$$

where

$$S'_\alpha = \{ \underline{\gamma} \mid \underline{y}(\underline{\alpha}) \in S_{\alpha\gamma} \}$$

Clearly, $\forall\ \underline{\alpha},\ S'_{\alpha} \neq \emptyset$, since $\underline{y}(\underline{x})$ is a solution of (2.32), that is, $\Phi_2(\underline{x}, \underline{z}, \underline{y}(\underline{x})) = 1$ is consistent w.r.t. a solution $\underline{z}(\underline{x})$. Also, if $\underline{y}(\underline{\alpha}) \in S_{\alpha\gamma}$ then $\underline{\gamma} \in S_{\alpha}$ for the same reason. Hence $S'_{\alpha} \subseteq S_{\alpha}$, and $\Phi_2(\underline{x}, \underline{z}, \underline{y}(\underline{x})) \leq \Phi_1(\underline{x}, \underline{z})$. Thus any solution $\underline{z}(\underline{x})$ of $\Phi_2(\underline{x}, \underline{z}, \underline{y}(\underline{x})) = 1$ is a solution of $\Phi_1(\underline{x}, \underline{z}) = 1$ too.

Q.E.D.

2.5 Evaluating Remarks

It has been shown how a characteristic function (equation) of a system of Boolean relations can be formed. Also, some theorems stating basic properties of characteristic equations, such as consistency, inclusion of don't care states, etc., have been presented. Once a problem is stated in the form of a system of B.E., and its characteristic function obtained, it is usually desired to obtain all or only some of the solutions. For this reason, a concise description of the main methods of solution was included. The methods can be subdivided into those producing just elementary solutions, and those which generate a general solution. The latter can further be subdivided into 3 groups, namely;

(1) No knowledge of an elementary solution needed - successive eliminations.

(2) A single elementary solution needed - Lowenheim's Theorem.

(3) Knowledge of all elementary solutions needed (even if only in an implicit form of a canonical expression of Φ) - Brown's method or others with economic encoding.

Depending on the tools used, all the methods may be classified as :

(1) Using maps (Marquand or other) - canonical forms of Φ, e.g., Svoboda's method.

(2) Algebraic - Φ can be in any form - successive eliminations, Lowenheim.

(3) Combination of 1 and 2 - economic encoding, e.g., Brown's method.

A method which is most appropriate to a given application will be used in the subsequent chapters. Algebraic methods will be used exclusively in the theoretical parts and in the examples which require more than 6 variables.

Certain important types of solutions were discussed in Section 2.4. The identity and the unique-solution theorems, and the Lemma 2.3 will play a key rôle in the chapters to follow.

CHAPTER 3

ANALYSIS OF COMBINATIONAL CIRCUITS

3.1 Introduction

It is usually assumed that combinational switching circuits contain no feedback loops, their response being dependent only on the present state of the input variables. In turn, sequential circuits contain feedback loops, hence their response depends not only on the present input state, but also on its past values. However, as a number of authors have pointed out [12, 14, 30], the presence of closed loops in switching circuits will not affect their overall combinational character in some cases. Moreover, the circuit implementation may require less gates than the loopless equivalent which realizes the same combinational function.

Kautz [14] has considered a one dimensional unilateral cellular array of identical cells connected in a closed loop. He stated conditions under which such a circuit with a closed loop would not degenerate. Also, it was shown that the same functions implemented in the more classical way would require a higher total number of NOR gates.

Furthermore, the possibility of loops in combinational circuits may arise as a natural phenomenon when proper general synthesis procedures using fixed libraries of building modules (combinational and sequential) are used. However, before such a general synthesis procedure can be developed, a close inquiry into the behaviour and decomposition of such generalized combinational circuits is required. An attempt of such an analysis is presented in this and the following chapter.

The analysis methodology is centered around the theory of Boolean equations and their characteristic functions as presented in Chapter 2. It will be applied to a general sequential circuit (asynchronous) with the aim of determining the circuit's properties pertinent to combinational behaviour.

3.2 Basic Concepts

Without loss of generality, let a general switching circuit be represented by an asynchronous Mealy-type machine.

Resolution Level:

Let all variables describing the behaviour of the circuit be considered at the level given by the two distinct values 0, 1 of B_2. Furthermore, let the activity of the circuit be observed at discrete time instances $0, 1, \ldots, t, t+1, \ldots;$ the time interval between two consecutive instances t_i, t_{i+1} being defined by the transition time Δt

needed by the circuit to go from one state of the internal state variables to the next one, regardless of whether that next state is stable or is going to change in the t_{i+1}, t_{i+2} transition period. If the state is stable then let the intervals be fixed and equal to the time interval needed for the last transition to take place, until the next change of the input stimulus - next transition period.

Definition 3.1:

General Switching Circuit (GSC) is represented by a first order finite state asynchronous Mealy machine (Figure 3.1)

$$M = (X, Y, Z, \underline{f}, \underline{g})$$

where

X - finite set of input states (stimuli)

Y - finite set of output states (responses)

Z - finite set of internal states

$\underline{f}$ - output vector function

$\underline{g}$ - state transition vector function.

Furthermore, let

$$\underline{y}^t = \underline{f}(\underline{x}^t, \underline{z}^t)$$

$$\underline{z}^{t+1} = \underline{g}(\underline{x}^t, \underline{z}^t)$$

describe the behaviour of the circuit at time t, where

$$\underline{x}^t = (x_1^t, x_2^t, \ldots, x_r^t) \in X$$

$$\underline{y}^t = (y_1^t, y_2^t, \ldots, y_q^t) \in Y$$

$$\underline{z}^t = (z_1^t, z_2^t, \ldots, z_n^t) \in Z$$

are the stimuli, the response, and the internal state of GSC at time t, respectively, and

$$\underline{z}^{t+1} = (z_1^{t+1}, z_2^{t+1}, \ldots, z_n^{t+1}) \in Z$$

the next internal state generated by $\underline{g}$.

Also, let the functions $\underline{f}$, $\underline{g}$ be represented by ideal combinational circuits (without feedback loops) with no internal delays, the transition times then being determined and fixed to Δt by external delay elements (a fundamental model of M).

Moreover, let

$\underline{z}$ represent the present state of the circuit at any time,

$\underline{z}'$ the next state generated by $\underline{g}$, $\underline{z}' = \underline{g}(\underline{x}, \underline{z})$, $\underline{x} \in X$; $\underline{z}$, $\underline{z}' \in Z$,

Z_α be the set of all internal states which can possibly be reached from any state $\underline{z} \in Z$ for the input $\underline{x} = \underline{\alpha} \in X$,

$$Z = \bigcup_{\underline{\alpha} \in X} Z_{\alpha}, \qquad \bar{Z} = B_2^n - Z.$$

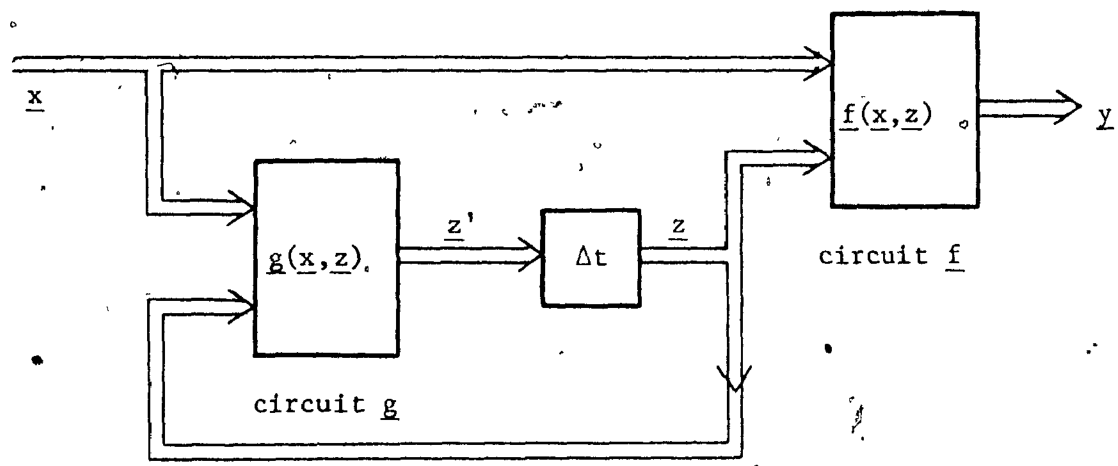


FIGURE 3.1. GENERAL SWITCHING CIRCUIT.

Definition 3.2:

State transition characteristic function $\Phi_g(\underline{x}, \underline{z}, \underline{z}')$ [8, 26, 35] is defined as the characteristic function (Definition 2.2) of the system of B.E.

$$\underline{z}' = \underline{g}(\underline{x}, \underline{z}).$$

That is,

$$\Phi_g(\underline{x}, \underline{z}, \underline{z}') = \prod_{i=1}^{n} (z_i' \cdot g_i(\underline{x}, \underline{z}) + \bar{z}_i' \cdot \bar{g}_i(\underline{x}, \underline{z}))$$

for $\underline{x} \in X$ and $\underline{z} \in Z$, and $\Phi_g(\underline{x}, \underline{z}, \underline{z}') = 0$ for $\underline{x} \in \bar{X}$, $\underline{z} \in \bar{Z}$.

Definition 3.3:

Stable state characteristic function $\Phi_g^s(\underline{x}, \underline{z}')$ is defined as the characteristic function of the system of B.E.

$$\underline{z}' = \underline{g}(\underline{x}, \underline{z}').$$

That is,

$$\Phi_g^s(\underline{x}, \underline{z}') = \prod_{i=1}^{n} (z_i' \cdot g_i(\underline{x}, \underline{z}') + \bar{z}_i' \cdot \bar{g}_i(\underline{x}, \underline{z}'))$$

for $\underline{x} \in X$, $\underline{z}' \in Z$, and $\Phi_g^s(\underline{x}, \underline{z}') = 0$ for $\underline{x} \in \bar{X}$, $\underline{z}' \in \bar{Z}$.

Definition 3.4:

Next state characteristic function $\Phi_g^z(\underline{x}, \underline{z}')$ is defined in terms of $\Phi_g(\underline{x}, \underline{z}, \underline{z}')$ as

$$\Phi_g^z(\underline{x}, \underline{z}') = \sum_{\underline{\beta} \in B_2^n} \Phi_g(\underline{x}, \underline{\beta}, \underline{z}').$$

<u>Definition 3.5</u>: Terminal state function.

$G(I_k, \underline{z}_R)$ is defined as a function giving the terminal state $\underline{z}_T = \underline{z}^{t+k+1}$ reached by an input sequence $I_k = (\underline{x}^t, \underline{x}^{t+1}, \ldots, \underline{x}^{t+k})$ from an initial state $\underline{z}_R = \underline{z}^t \in Z$.

$$\underline{z}_T = G(I_k, \underline{z}_R) = \underline{g}(\underline{x}^{t+k}, \underline{g}(\underline{x}^{t+k-1}, \ldots, \underline{g}(\underline{x}^t, \underline{z}_R)) \ldots).$$

<u>Definition 3.6</u> : Stability.

Given an initial state $\underline{z}_R$ and a stimulus $\underline{x} = \underline{\alpha} \in X$, the GSC is said to be stable for $\underline{\alpha}$ and $\underline{z}_R$ if for an input sequence $I_k = (\underline{x}^t, \underline{x}^{t+1}, \ldots, \underline{x}^{t+k})$ such that $\underline{x}^{t+i} = \underline{\alpha}$, $i = 0, 1, \ldots, k$, there exists a finite k such that

$$G(I_k, \underline{z}_R) = G(I_{k+j}, \underline{z}_R)$$

for $j = 1, 2, 3, \ldots$ That is, with a constant stimulus $\underline{\alpha}$ a stable state is reached after a finite number of transitions (k).

<u>If the circuit is stable</u> for all possible $\underline{z}_R \in Z$ then it is stable at $\underline{x} = \underline{\alpha}$.

If there is no such finite k then the circuit is oscillatory (with respect to $\underline{z}$) for that $\underline{x} = \underline{\alpha}$ and $\underline{z}_R$.

Even though a circuit may be oscillatory for some $\underline{\alpha}$ and $\underline{z}_R$, not all internal variables $z_i \in \underline{z}$ have to oscillate. Therefore, stability of a variable $z_i \in \underline{z}$ can be defined similarly as in Definition 3.6.

Definition 3.7 : Simple oscillations.

If the circuit is oscillatory (Definition 3.6) for some $\underline{x} = \underline{\alpha}$ and $\underline{z}_R$, then it has simple oscillations at $\underline{\alpha}$ if there is no other initial state $\underline{z}_{R2} \neq \underline{z}_{R1}$ such that the circuit would be stable for $\underline{\alpha}$ and $\underline{z}_{R2}$.

For instance, the state diagram as shown in Figure 3.2

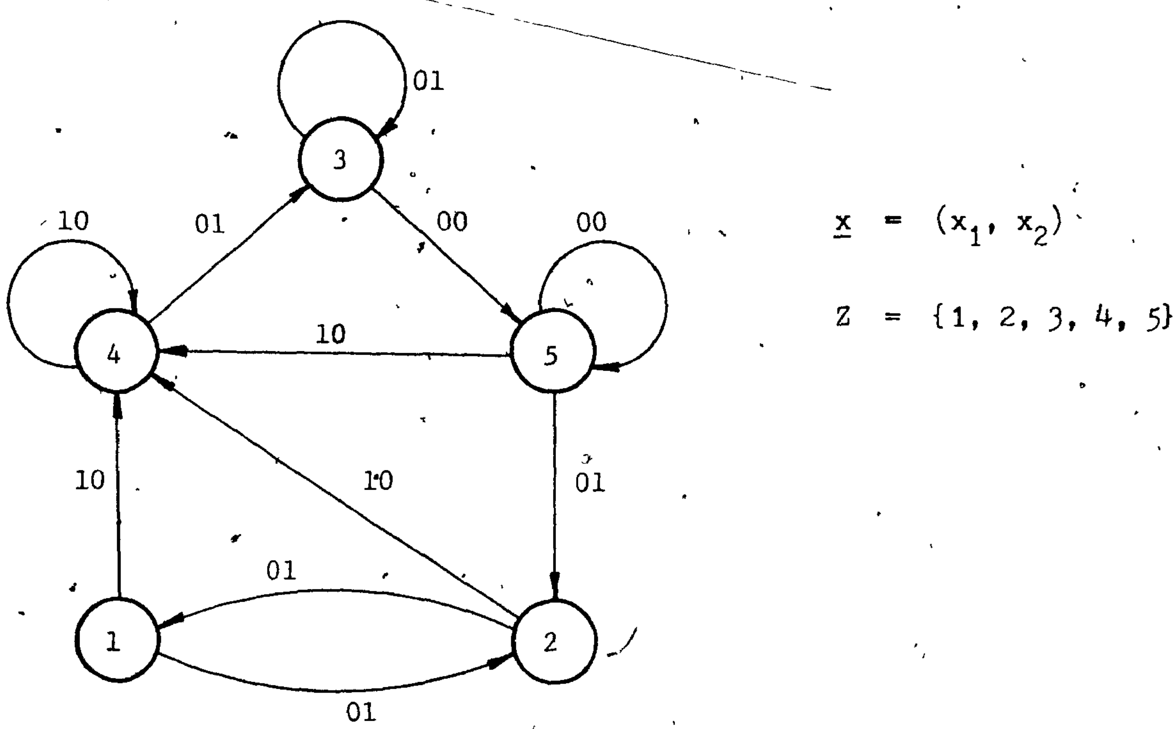


FIGURE 3.2. ILLUSTRATION FOR DEFINITION 3.7.

describes a circuits which is oscillatory for $\underline{x} = (0, 1)$ and $\underline{z}_{R1} = 5$, and stable for the same $\underline{x}$, but $\underline{z}_{R2} = 4$. Therefore, by Definition 3.7 it does not have simple oscillations at $\underline{x} = (0, 1)$. By changing the circuit so that the stimulus causing the $4 \to 3$ and $3 \to 3$ transitions is $(1, 1)$, the new circuit would have simple oscillations at $\underline{x} = (0, 1)$, since there is no other transition caused by that $\underline{x}$ which would bring the circuit to a stable state.

<u>Definition 3.8</u> : Steady state.

A circuit has reached steady state (conditions) for a stimulus $\underline{x} = \underline{\alpha} \in X$ if it is in a stable state or is oscillatory. Furthermore, let all the states $\underline{z} \in Z$ through which the circuit is passing while in steady state, be called steady states. All other states are then transient.

<u>Definition 3.9</u> :

A k-transition circuit is such that it passes through at most k transient states when changing steady states due to a change in the input stimulus $\underline{x}$.

Lemma 3.1 :

$$\text{If} \quad |\underline{z}| = n \quad \text{then} \quad k \leqslant 2^n - 2.$$

Proof : A steady state at $\underline{x}_1 \in X$ consists of at least one (stable) state $\underline{z}_1$. Assume that a change of $\underline{x}_1$ to $\underline{x}_2 \neq \underline{x}_1$ would bring the circuit to a steady state $\underline{z}_2$. Then there remains at most $|B_2^n - \{\underline{z}_1, \underline{z}_2\}|$ different transient states through which the circuit may pass before reaching the total steady state $\underline{x}_2, \underline{z}_2$.

Q.E.D.

$$\text{If} \quad |Z| < 2^n \text{ is known then } k \leqslant |Z| - 2.$$

Definition 3.10 : 0-transition circuits.

Clearly, 0-transition circuits have the fastest transitions between steady states. The most common type is the normal circuit defined as:

(1) It has at least one stable state for each input stimulus $\underline{x} \in X$.

(2) Each unstable state passes directly into a stable state.

If oscillations are permitted, then a class of quasi-normal circuits (equivalent to the class of 0-transition circuits) is defined as:

(1) The circuits can reach steady state in finite time for each $\underline{x} \in X$ (Lemma 3.1).

(2) If a state becomes unstable due to a change in the input stimuli the transition leads directly to a steady state. That is,

(i) either to a stable state

or

(ii) to a state which is a member of the set of states defining the oscillations for that $\underline{x}$. Hence, if $\underline{x}$ does not change then the circuit would keep entering periodically that state.

Remark : The class of normal circuits is a sub-class to the class of the quasi-normal circuits.

Definition 3.11 : Operation of a circuit.

It will be assumed that the circuit is operated in the following way:

(1) The stimulus $\underline{x}$ cannot change unless the circuit is in a steady state.

(2) The state $\underline{z}$ and hence the response $\underline{y}$ too are of interest to the external environment only when the circuit is in a steady state.

The following are some properties related to the characteristic functions defined before:

<u>Lemma 3.2</u> :

The equation $\Phi_g(\underline{x}, \underline{z}, \underline{z}') = 1$ (Definition 3.2) has a unique solution for $\underline{z}'$, defined over $\underline{x} \in X$, $\underline{z} \in Z$. That solution is equivalent to the function $\underline{g}(\underline{x}, \underline{z})$.

<u>Proof</u>: The way in which Φ_g is formed, and then Theorem 2.4.

<u>Lemma 3.3</u> :

Given a stimulus $\underline{x} = \underline{\alpha} \in X$ then all the possible states the circuit may reach from any initial state $\underline{z}_R \in Z$ are obtained as the solutions $\underline{z}'$ to the truth equation

$$\Phi_g^z (\underline{\alpha}, \underline{z}') = 1 .$$

Proof : By Definition 3.4, Lemma 3.2 and summation over all $\underline{z} \in B_2^n$.

Lemma 3.4 :

The solution set $\{\underline{z}'\}$ of the equation $\phi_g^s(\underline{\alpha}, \underline{z}') = 1$, $\underline{\alpha} \in X$ (Definition 3.3) is equivalent to the set of all stable states the circuit can possible reach for $\underline{x} = \underline{\alpha}$ and any initial state.

Proof : The equation $\phi_g^s(\underline{x}, \underline{z}') = 1$ is solution equivalent to the system of equations $\underline{z}' = \underline{g}(\underline{x}, \underline{z}')$ (Definition 3.3, 2.2, Lemma 2.1). That is, if $\underline{z}^*$ is a solution then $\underline{z}^* = \underline{g}(\underline{x}, \underline{z}^*)$ identically. Assume that $\underline{x} = \underline{\alpha}$ does not change, and that $\underline{z}^*$ was reached after k transitions from an initial state at time t, $\underline{z}_R = \underline{z}^t$. Thus $\underline{z}^{t+k} = \underline{z}^*$ and $\underline{z}^{t+k+1} = \underline{g}(\underline{\alpha}, \underline{z}^{t+k}) = \underline{g}(\underline{\alpha}, \underline{z}^*) = \underline{z}^*$, and also

$$\underline{z}^{t+k+2} = \underline{g}(\underline{\alpha}, \underline{z}^{t+k+1}) = \underline{g}(\underline{\alpha}, \underline{z}^*) = \underline{z}^*$$

$$\vdots$$

$$\underline{z}^{t+k+j} = \underline{z}^*$$

However, by Lemma 3.1, k is finite, and from the above,

$$G(I_k, \underline{z}_R) = G(I_{k+j}, \underline{z}_R) \quad ;$$

therefore, by Definition 3.6 the circuit is in a stable state $\underline{z}^*$. Similarly, it can be shown that any stable state $\underline{z}^*$ at $\underline{x} = \underline{\alpha}$

satisfies the equality $\underline{z}^* = \underline{g}(\underline{\alpha}, \underline{z}^*)$, hence $\Phi_g^S(\underline{\alpha}, \underline{z}^*) = 1$ identically.

Q.E.D.

Lemma 3.5 :

The circuit $\underline{g}$ has simple oscillations at $\underline{x} = \underline{\alpha} \in X$ iff $\Phi_g^S(\underline{\alpha}, \underline{z}') = 1$ is inconsistent.

Proof : Simple oscillations (Definition 3.9) $\leftrightarrow$ no stable state at $\underline{\alpha}$. By Lemma 3.4, the solution set to $\Phi_g^S(\underline{\alpha}, \underline{z}') = 1$ is the set of stable states at $\underline{\alpha}$. No stable states $\leftrightarrow$ no solution to $\Phi_g^S(\underline{\alpha}, \underline{z}') = 1$.

Q.E.D.

Definition 3.12 :

Steady state characteristic function $\Phi_g^C(\underline{x}, \underline{z}')$ is defined so that the equation $\Phi_g^C(\underline{\alpha}, \underline{z}') = 1$, $\underline{\alpha} \in X$, has as its solutions all the steady states the circuit $\underline{g}$ may possibly assume at $\underline{\alpha}$.

Theorem 3.1 :

(1) If $\underline{g}$ is quasi-normal then

$$\Phi_g^C(\underline{x}, \underline{z}') = \Phi_g^Z(\underline{x}, \underline{z}') \qquad (3.1)$$

(2) If $\underline{g}$ is stable (no oscillations at any $\underline{x} \in X$) then

$$\Phi_g^c(\underline{x}, \underline{z}') = \Phi_g^s(\underline{x}, \underline{z}') \tag{3.2}$$

(3) In general (any circuit $\underline{g}$)

$$\Phi_g^s(\underline{x}, \underline{z}') \leq \Phi_g^c(\underline{x}, \underline{z}') \leq \Phi_g^z(\underline{x}, \underline{z}') \tag{3.3}$$

Proof :

(1) By Definition 3.10 and Lemma 3.3.

(2) By Lemma 3.4 and the fact that there are no oscillatory states.

(3) $\Phi_g^s(\underline{x}, \underline{z}') = 1$ describes the stable states only, hence $\Phi_g^s(\underline{x}, \underline{z}') \leq \Phi_g^c(\underline{x}, \underline{z}')$. In turn, $\Phi_g^z(\underline{x}, \underline{z}') = 1$ gives all the transient and steady states, hence $\Phi_g^c \leq \Phi_g^z$.

Q.E.D.

Corollary 1 : Generation of Φ_g^c.

The steady state characteristic function $\Phi_g^c(\underline{x}, \underline{z}')$ can be obtained as a result of the following sequence:

$$\Phi_g^1(\underline{x}, \underline{z}, \underline{z}^1) = \Phi_g(\underline{x}, \underline{z}, \underline{z}')$$

$$\Phi_g^i(\underline{x}, \underline{z}, \underline{z}^i) = \sum_{\underline{z}^{i-1} \in B_2^n} \Phi_g^{i-1}(\underline{x}, \underline{z}, \underline{z}^{i-1}) \cdot \Phi_g(\underline{x}, \underline{z}^{i-1}, \underline{z}^i)$$

for $i = 2, \ldots, 2^n$; $n = |\underline{z}|$

Then

$$\Phi_g^c(\underline{x}, \underline{z}') = \sum_{i=1}^{2^n} \Phi_g^i(\underline{x}, \underline{z}', \underline{z}').$$

Proof : The function $\Phi_g^i(\underline{x}, \underline{z}, \underline{z}^i)$ relates the initial state $\underline{z}$ with the next state after i transition periods. Thus

$i = 1$	gives all stable states $\Phi_g(\underline{x}, \underline{z}', \underline{z}') = \Phi_g^s(\underline{x}, \underline{z}')$
$i > 1$	gives all states involved in oscillations passing at most through i states.

Note, that the form $\sum_{\underline{z}^{i-1} \in B_2^n} \Phi_g^{i-1} \cdot \Phi_g$ results from the application of Lemma 2.3.

3.3 Combinational Behaviour

It will be shown here, under which conditions a general circuit (Definition 3.1, Figure 3.1) would behave as a combinational circuit.

Definition 3.13 : Combinational behaviour.

A circuit has a combinational behaviour with respect to the output, for the stimulus $\underline{x} = \underline{\alpha} \in X$, if a unique response $\underline{y}$ is associated with $\underline{\alpha}$.

That is, the circuit response is independent of the internal states $\underline{z} \in Z_{\alpha}$. Otherwise, the circuit has a sequential behaviour at $\underline{\alpha}$.

Remarks :

(1) If a circuit has a combinational behaviour for all $\underline{\alpha} \in X$ then it has combinational behaviour - it represents a combinational switching function.

(2). If a particular $\underline{\alpha}$ is not specified then combinational behaviour is assumed for all $\underline{\alpha} \in X$.

(3) Similarly as in Definition 3.6, combinational behaviour of a single variable $y_i \in \underline{y}$ can be defined.

(4) The following nomenclature will be used:

combinational	- means combinational behaviour with respect to the outputs.
purely combinational	- same as above except that the circuit contains no feedback loops.

The first theorem to be presented here discusses the trivial case of a circuit exhibiting combinational behaviour. It is the case where the next state transition function $\underline{g}(\underline{x}, \underline{z})$ is independent of $\underline{z}$ (no closed loop).

Theorem 3.2 :

A circuit has a combinational behaviour if the equation $\Phi_g^c(\underline{x}, \underline{z}') = 1$ has a unique solution $\underline{z}'(\underline{x})$.

Proof : Trivial, apply Definition 3.4, Lemma 3.3, and Definition 3.13. Since $\underline{f}$ is a combinational function then the output $\underline{y}$ has a unique state associated with each $\underline{x} \in X$.

Q.E.D.

The situation becomes more complicated when there exists a closed feedback loop, that is, when $\Phi_g^c(\underline{x}, \underline{z}') = 1$ does not have a unique solution. However, even in such a case the circuit may still have combinational behaviour with respect to $\underline{y}$. (The function $\underline{f}$ has to be independent of some of the states of $\underline{g}$). The following theorem will investigate such a case. It will be assumed that the circuit is operated as specified in Definition 3.11. First, another definition is needed.

Definition 3.14 :

Output generator characteristic function $\Phi_f(\underline{x}, \underline{z}', \underline{y})$ is defined as

$$\Phi_f(\underline{x}, \underline{z}', \underline{y}) = \prod_{i=1}^{q} (\bar{y}_i \cdot \bar{f}_i(\underline{x}, \underline{z}') + y_i \cdot f_i(\underline{x}, \underline{z}'))$$

$$\text{for} \quad \underline{x} \in B_2^n, \quad \underline{z}' \in B_2^n.$$

(It is the characteristic function of the system $\underline{y} = \underline{f}(\underline{x}, \underline{z}')$ of B.E.)

Theorem 3.3 : Combinational behaviour.

Let

$$\Phi_c(\underline{x}, \underline{y}) = \sum_{\underline{\beta} \in B_2^n} [\Phi_g^c(\underline{x}, \underline{\beta}) \cdot \Phi_f(\underline{x}, \underline{\beta}, \underline{y})] . \tag{3.4}$$

The general circuit of Definition 3.1 has a combinational behaviour if and only if the equation $\Phi_c(\underline{x}, \underline{y}) = 1$ has a unique solution $\underline{y}(\underline{x})$, $\underline{x} \in X$. That solution is then the combinational function generated by the circuit.

Proof : $\Phi_g^c(\underline{x}, \underline{z}') = 1$ describes all the steady states $\underline{z}'$ the circuit $\underline{g}$ may assume for a particular $\underline{x} \in X$. Considering then each such state $\underline{z}'$, the response $\underline{y}$ is generated through the circuit $\underline{f}$ - the output is thus obtained as a solution $\underline{y}$ of $\Phi_f(\underline{x}, \underline{z}', \underline{y}) = 1$.

Therefore, by Lemma 2.3, the possible states of $\underline{y}$ are obtained as the solutions to the equation

$$\Phi_c(\underline{x}, \underline{y}) = 1 \tag{3.5}$$

where Φ_g^c and Φ_f replace Φ_1, and Φ_2, respectively, in the Lemma.

If $\Phi_c = 1$ has a unique solution then a unique response is associated with each $\underline{x} \in X$; consequently, the circuit has combinational behaviour (Definition 3.13). Because of Lemma 2.3, that solution is then the combinational function generated by the entire circuit. The converse - a unique response at $\underline{x}$ means that all steady states $\underline{z}'$ (at that $\underline{x}$) generate the same output state $\underline{y}$. But by Definition 3.12 the function Φ_g^c contains only steady states, hence (3.5) must have a unique solution.

Q.E.D.

<u>Remark</u>: If the function Φ_g^c as defined in Definition 3.12 is not available, but another function $\Phi_g^*(\underline{x}, \underline{z}')$ which contains not only the steady states, but also some or all of the transient ones ($\Phi_g^*(\underline{x}, \underline{z}') \geqslant \Phi_g^c(\underline{x}, \underline{z}')$), then the condition of combinational behaviour becomes sufficient only. That is, if the equation

$$\sum_{\underline{\beta} \in B_2^n} [\Phi_g^*(\underline{x}, \underline{\beta}) \cdot \Phi_f(\underline{x}, \underline{\beta}, \underline{y})] = 1 \tag{3.6}$$

has a unique solution then the circuit has combinational behaviour. Again, the solution is the function generated by the circuit.

The material so far presented was intended to give more insight into the relationship between a circuit and its characteristic functions. Also, the case when a sequential circuit has combinational behaviour with respect to its outputs has been studied. (See Chapter 5 for illustrative examples).

It is seldom the case, however, that one is presented with a circuit and then requested to determine whether it has combinational behaviour. More often the design of such circuits is performed in engineering practice. The problem is then the interconnection of some available building modules (circuits) so as to realize a given combinational switching function. Thus the following chapter will deal with the realization of switching functions.

CHAPTER 4

REALIZATION OF COMBINATIONAL SWITCHING FUNCTIONS

It has been shown so far how Boolean equations can be used for analyzing combinational circuits, even those containing feedback loops. It will be demonstrated here that B.E. yield very compact expressions stating functional realization, and at the same time unifying the various approaches to functional decomposition.

4.1 Introduction

A decomposition of a switching function $\phi(\underline{x})$ is considered to be a sequence of functions

$$\eta_1(\underline{y}_1, \underline{z}_1), \quad \eta_2(\underline{y}_2, \underline{z}_2), \ \ldots, \ \eta_m(\underline{y}_m, \underline{z}_m) \tag{4.1}$$

where

(1) $\underline{z}_j \subseteq \underline{x}$

(2) $\underline{y}_j \subseteq \{\eta_1, \ldots, \eta_{j-1}\} \subseteq \underline{\eta}$

(3) For every $\underline{x} \in B_{(2)}^n$ for which $\phi(\underline{x})$ is defined, η_m is defined and equal to $\phi(\underline{x})$ [21]. Thus the network realizes $\phi(\underline{x})$.

It would be a rather difficult task to obtain the complete decomposition all at once; however, it was shown that the complete decomposition can be generated by a repeated application of simple decompositions [5, 7, 11, 13, 15, 16, 19, 20, 21, 28], having the form:

$$\phi(\underline{x}) = \eta_2(\eta_1(\underline{y}), \underline{z}) \quad ; \qquad \underline{y} \subseteq \underline{x}, \quad \underline{z} \subseteq \underline{x} \tag{4.2}$$

$$\underline{y} \cup \underline{z} = \underline{x}$$

or

$$\phi(\underline{x}) = \eta_2(\eta_1^1(\underline{y}^1),\ \eta_1^2(\underline{y}^2),\ \ldots,\ \eta_1^t(\underline{y}^t),\ \underline{z}); \tag{4.3}$$

$$\underline{y}^i \subseteq \underline{x}, \quad \underline{z} \subseteq \underline{x}$$

$$\bigcup_i \underline{y}^i \cup \underline{z} \quad \underline{x}$$

Even more restrictions can be placed on the decomposition by requiring the sets $\underline{y}^i$ and $\underline{z}$ to be disjoint, i.e.,

$$\underline{y}^i \cap \underline{y}^j = \emptyset, \qquad i \neq j;$$

$$\underline{y}^i \cap \underline{z} = \emptyset, \qquad i = 1, \ldots, k.$$

The concepts above may easily be extended over a vector of functions

$\underline{\phi}(\underline{x}) = (\phi_1(\underline{x}), \ldots, \phi_q(\underline{x}))$.

Determination of the functions η_1 and η_2 in (4.2) was first approached by Ashenhurst [1] and Curtis [40] through modified Karnaugh maps - so called decomposition charts. The method was then expanded and brought into the form of computer aided synthesis procedures by a number of authors. The decomposition was obtained either through the charts [11, 13, 15, 29], or by translating the theory into algebraic topological methods operating on cubical complexes [7, 21, 24, 25, 28, 36] . Furthermore, either any decomposition was sought, that is, η_1 and η_2 were allowed to be any functions that form a feasible decomposition, or restrictions were placed so that the functions had to be chosen from a set of available functions realized by some circuit modules. Theorems were stated which determined η_1 for a given ϕ and η_2, similarly η_2 could be determined for that ϕ and a known η_1 [7, 11, 15, 21, 23, 25]. This type of approach is rather suitable when the actual circuits are to be implemented using integrated circuits, where a fixed number of different combinational functions is available to the designer. These functions then form the set of modules (a library) from which the functions η_1, η_2 must be selected, and by a proper interconnection the function $\phi(\underline{x})$ realized.

The decomposition procedure usually started by selecting the function η_1 and an assignment (mapping) of $\underline{x}$ to $\underline{y}$, and after that the function η_2 was sought by applying the decomposition theory. If an η_2 existed, it was tested against the set of available modules, and if the search was successful - a module realizing η_2 was found - then the synthesis of $\phi(\underline{x})$ was completed. Otherwise, the function η_2 was further decomposed by the same procedure, and so on, until a complete decomposition (4.1) of ϕ was obtained. If no decomposition existed for the η_1 chosen, then another function η_1 was selected from the set of modules, and another test for decomposition was performed. Provided, that the set of modules represented a complete set of Boolean functions, the synthesis was completed in a finite number of iterations, described above.

A more recent method of functional decomposition is based on Boolean equations [5, 16, 19, 20, 26]. It was shown that a simple decomposition can be obtained as a solution to a system of B.E. derived from $\phi(\underline{x})$ and the module function η_2. Marin [19, 20] used the method of Svoboda [31] to solve the equations, in order to generate a simple decomposition. The procedure was then expanded into a more complete computer program for the synthesis of combinational circuits[5]; however, the general solution to B.E. procedure was used [3] . In both cases, though, the decomposition proceeded in a direction opposite to that used in the methods based on Ashenhurst's work. That is, for a given ϕ, the function η_2 in (4.3) was selected from a set of modules, and then the function $\underline{\eta}_1 = \{\eta_1^i\}$, i = 1,..., k, and the set $\underline{z}$ was determined by solving the equation

$$\phi(\underline{x}) = \eta_2 (\underline{\eta}_1, \underline{z})$$

for the unknowns $\underline{\eta}_1$, $\underline{z}$. The solutions were then tested against the library of available modules. If a match was found, the decomposition was complete, otherwise the procedure was applied iteratively until a successful realization was obtained. The method has a straight forward extension for incorporating multiple output incompletely specified functions $\underline{y}(\underline{x}) = <\underline{\phi}(\underline{x}), \quad \underline{\phi}(\underline{x}) + \underline{d}(\underline{x})>$.

When the resulting circuit was to be optimal under certain predetermined criteria, both of the forementioned methods had to use more-or-less exhasutive searches aided by some heuristic rules. Therefore, the actual optimal circuit could be obtained in a reasonable time only in the case of a relatively simple function $\underline{y}(\underline{x})$. The method of B.E. has an advantage, however, since circuit constraints can easily be implemented by adding constraint equations to the original system. The number of possible decompositions can thus be reduced, which in turn lowers the search time for an optimal solution. The selection of constraints can be guided by the actual circuit layout, fan-out, availability of signal lines, etc. [5].

Another possible way to synthesize switching circuits is obtained by formulating the decomposition as a linear 0 - 1 integer programming problem [22]. The interconnections between gates are specified as the 0 - 1 variables, gate functions and input-output connections then as linear inequalities. A solution to the variables is sought within the system of inequalities, so that a certain cost function representing the designer's criterion of optimality would be minimized. Even though the optimal solution can be found in a finite number of steps, the approach requires a very large number of variables, growing at least exponentially with $|\underline{x}|$ and $|\underline{\phi}|$. Therefore, its usefulness is limited to small problems.

A rather compact form of the decomposition theorems can be obtained by applying the theory of Boolean differential calculus [26, 34]. Although the description is formally very concise, the computations involved in solving even a simple circuit are quite complex, not suited to practical problems. (Generation of prime implicants, etc.)

It will be demonstrated here, that an extended application of the theory of B.E. to the problem of functional realization and decomposition yields both formally and computationally feasible formulae. The theory is rather general, incorporating the various directions taken in the synthesis procedure (~~from~~ inputs to outputs and vice versa), as well as allowing for the function modules to be sequential circuits. It also covers the synthesis of multiple output incompletely specified functions; any circuit constraints can be implemented in terms of additional equations as in [5], since the procedure is unified under

B.E. Moreover, the concept of degenerate and direct transition circuits is introduced, and an important conclusion about the necessity of feedback loops in such circuits is derived.

4.2 Realization of Switching Functions

Let an incompletely specified function

$$\underline{y}(\underline{x}) = < \underline{\phi}(\underline{x}), \ \underline{\phi}(\underline{x}) + \underline{d}(\underline{x})>$$

be given, and let a realization of that function by a circuit C be considered, as shown in Figure 3.1, Definition 3.1. That realization would occur if the responses of the circuit map into the states $\underline{y}$ of the function (the permissible states) for each $\underline{x} \in B_2^r$. Thus the first assumption to be made is that the set X of possible input stimuli is the whole space B_2^r, unless the states in $\bar{X}$ are covered by $d_i(\underline{x})$, $i = 1, \ldots, q$. To simplify the discussion, $X = B_2^r$ will be assumed since the modifications required for the other case mentioned above are rather minor. (In case that $X \subset B_2^r$, the set X would also limit the function $\underline{y}(\underline{x})$.)

Definition 4.1 :

The circuit characteristic function $\Phi_C(\underline{x}, \underline{y})$ of C is defined so that the solutions to the equation $\Phi_C(\underline{\alpha}, \underline{y}) = 1$, $\underline{\alpha} \in X$ are the possible steady state responses the circuit may generate for $\underline{x} = \underline{\alpha} \in X$, provided that the conditions of Definition 3.11 are satisfied.

As shown in Theorem 3.3 and its proof,

$$\Phi_C(\underline{x}, \underline{y}) = \sum_{\underline{\beta} \in B_2^n} \Phi_g^C(\underline{x}, \underline{\beta}) \cdot \Phi_f(\underline{x}, \underline{\beta}, \underline{y}) \quad ; \qquad (4.4)$$

however, a sequential behaviour is assumed in general, hence $\Phi_C(\underline{x}, \underline{y}) = 1$ need not have a unique solution $\underline{y}(\underline{x})$.

If the function, $\Phi_g^*(\underline{x}, \underline{z}')$ (Remark to Theorem 3.3) is used instead of $\Phi_g^C(\underline{x}, \underline{z}')$ in (4.4) then all statements which will be made about functional realization/decomposition become sufficient only. (The same reason as the one stated in the above referenced remark).

Note, that $\Phi_C(\underline{x}, \underline{y}) = 1$ is always consistent with respect to solutions $\underline{y} = \underline{y}(\underline{x})$, since it represents an actual circuit which has at least one steady state associated with each $\underline{x} \in X$.

Definition 4.2 :

The output characteristic function $\Phi(\underline{x}, \underline{y})$ is formally defined so that the equation $\Phi(\underline{\alpha}, \underline{y}) = 1$, $\alpha \in X$, has as its solutions such states of $\underline{y}$ which a circuit C is permitted to assume at most as its steady state responses at that $\underline{x} = \underline{\alpha} \in X$. If a particular circuit C satisfies this condition then it is said that C realizes $\Phi(x, y)$.

Lemma 4.1 :

A circuit C realizes $\Phi(\underline{x}, \underline{y})$ iff

$$\Phi_C(\underline{x}, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y}). \tag{4.5}$$

Proof : Let $\underline{Y}$ be a response generated by C at $\underline{x} = \underline{\alpha}$. Then $\Phi_C(\underline{\alpha}, \underline{Y}) = 1$. But C realizes Φ, hence by Definition 4.1, $\Phi(\underline{\alpha}, \underline{Y}) = 1$. If $\underline{Y}$ cannot be generated by C then $\Phi_C(\underline{\alpha}, \underline{Y}) = 0$ and $\underline{Y}$ may or may not be in Φ. Therefore, $\Phi_C(\underline{x}, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y})$. The converse is proven similarly: $\Phi_C \leqslant \Phi$ implies that the responses of C are solutions to $\Phi = 1$, hence Φ is an output characteristic function, in other words, C realizes Φ.

Q.E.D.

Theorem 4.1 : Realization of $\underline{y}(\underline{x})$.

The function $\underline{y}(\underline{x}) = \langle \underline{\phi}(\underline{x}), \underline{\phi}(\underline{x}) + \underline{d}(\underline{x}) \rangle$ is realized by a circuit C if and only if the output characteristic function

$$\Phi(\underline{x}, \underline{y}) = \prod_{i=1}^{q} [\, y_i \cdot \phi_i(\underline{x}) + \bar{y}_i \cdot \bar{\phi}_i(\underline{x}) + d_i(\underline{x}) \,] \quad (4.6)$$

is realized by C, that is iff

$$\Phi_C(\underline{x}, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y})$$

Proof : The specification of $\underline{y}(\underline{x})$ as an incompletely specified function can be written as a system of Boolean relations

$$y_i \geqslant \phi_i(\underline{x}) \quad \text{and} \quad y_i \leqslant \phi_i(\underline{x}) + d_i(\underline{x}), \quad i = 1, \ldots, q,$$

which have to be satisfied simultaneously. By Lemma 2.1, the characteristic function of the system is the function (4.6). That is, the solutions to $\Phi(\underline{\alpha}, \underline{y}) = 1$, $\underline{\alpha} \in B_2^r$ are the permissible states $\underline{y}$ of the function $\underline{y}(\underline{x})$ at $\underline{\alpha}$. Therefore, the realization of Φ is equivalent to the realization of $\underline{y}(\underline{x})$ (Definition 4.2), since $\Phi_C(\underline{x}, \underline{y}) = 1$ is consistent. Consequently, the function $\underline{y}(\underline{x})$ is realized if and only if $\Phi_C(\underline{x}, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y})$, by Lemma 4.1. (Note, that $\Phi(\underline{x}, \underline{y}) = 1$ is always consistent).

Q.E.D.

Let now a cascade realization of $\underline{y}(\underline{x})$ be considered as shown in Figure 4.1. By a repeated application of Lemma 2.3, the overall circuit characteristic function $\Phi_C(\underline{x}, \underline{y}^k)$ can be obtained as

$$\Phi_C(\underline{x}, \underline{y}^k) = \sum_{\underline{\beta}^{k-1} \in B_2^{q k-1}} \left(\Phi_{Ck}(\underline{x}, \underline{\beta}^{k-1}, \underline{y}^k) \cdot \sum_{\underline{\beta}^{k-2} \in B_2^{q k-2}} \left(\Phi_{Ck-1}(\underline{x}, \underline{\beta}^{k-2}, \underline{\beta}^{k-1}) \right.\right.$$

$$\cdot \ldots \sum_{\underline{\beta}^1 \in B_2^{q_1}} \left(\Phi_2(\underline{x}, \underline{\beta}^1, \underline{\beta}^2) \cdot \Phi_1(\underline{x}, \underline{\beta}^1)\right) \ldots)) \quad (4.7)$$

And hence if $\Phi_C(\underline{x}, \underline{y}^k) \leqslant \Phi(\underline{x}, \underline{y}^k)$ then $\underline{y}(\underline{x})$ is realized. By looking at the network from the output side and taking each C_i separately, then $\Phi(\underline{x}, \underline{y}^k)$ is realized if $\Phi_{Ck}(\underline{x}, \underline{y}^{k-1}, \underline{y}^k) \leqslant \Phi(\underline{x}, \underline{y}^k)$, provided that the output characteristic function $\Phi_{k-1}(\underline{x}, \underline{y}^{k-1})$ defining the permissible input states of C_k is realized by C_{k-1}, i.e.,

$$\Phi_{C\,k-1}(\underline{x}, \underline{y}^{k-2}, \underline{y}^{k-1}) \leqslant \Phi_{k-1}(\underline{x}, \underline{y}^{k-1}) \quad ,$$

which holds provided that $\Phi_{k-2}(\underline{x}, \underline{y}^{k-2})$ is realized, etc., until $\Phi_1(\underline{x}, \underline{y}^1)$ is realized by C_1.

The particular output characteristic functions $\Phi_i(\underline{x}, \underline{y}^i)$ can be derived as stated in the following Lemma.

<u>Lemma 4.2</u> : Determination of $\Phi_i(\underline{x}, \underline{y}^i)$. (Figure 4.1).

Let $\Phi^+_{Ci}(\underline{x}, \underline{y}^i, \underline{y})$ be the overall circuit characteristic function of the combined circuits $C_{i+1}, \ldots, C_k$. Then the output

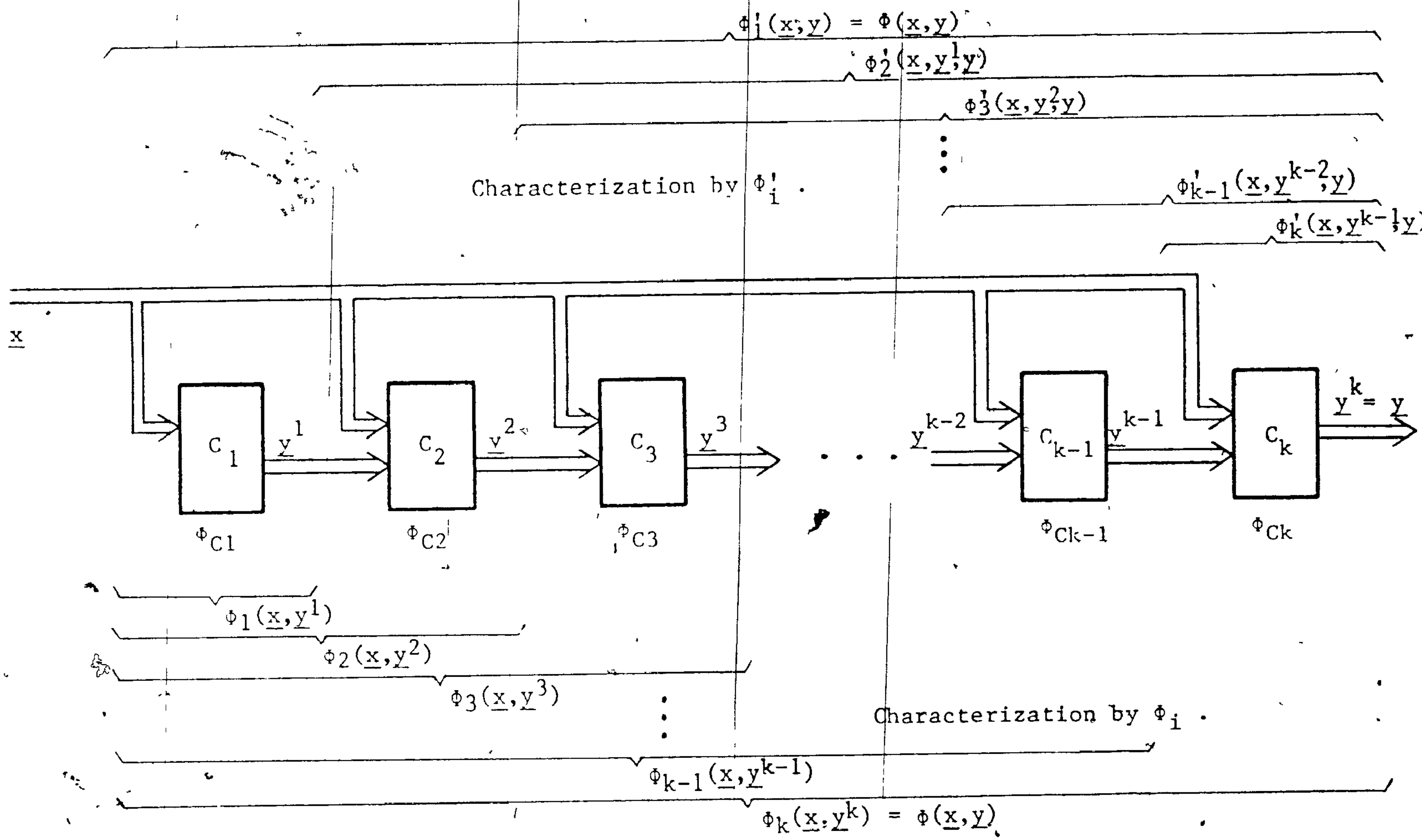


FIGURE 4.1. CASCADE REALIZATION OF $\Phi(\underline{x},\underline{y})$.

characteristic function Φ_i is equivalent to the characteristic function of the relation $\Phi^{+}_{Ci}(\underline{x}, \underline{y}^i, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y})$, assumed valid for all $\underline{y} \in B_2^q$. That is,

$$\Phi_i(\underline{x}, \underline{y}^i) = \prod_{\underline{\beta} \in B_2^q} [\, \overline{\Phi^{+}_{Ci}}(\underline{x}, \underline{y}^i, \underline{\beta}) + \Phi(\underline{x}, \underline{\beta}) \,] \qquad (4.8)$$

and

(1) There exists a circuit C which completes the realization of $\underline{y}(\underline{x})$ iff $\Phi_i(\underline{x}, \underline{y}^i) = 1$ is consistent for solutions $\underline{y}^i(\underline{x})$.

(2) A circuit C represented by its circuit characteristic function $\Phi_C(\underline{x}, \underline{y}^i)$ will complete the realization iff $\Phi_C(\underline{x}, \underline{y}^i) \leqslant \Phi_i(\underline{x}, \underline{y}^i)$.

Proof : The second part can be proven as follows:

(2) Assume that $\Phi_C(\underline{x}, \underline{y}^i) \leqslant \Phi_i(\underline{x}, \underline{y}^i)$. It will be shown that such C completes the realization of $\underline{y}(\underline{x})$. By Lemma 2.3, the combined circuits C and $C_{i+1}, \ldots, C_k$ have the overall characteristic function

$$\Phi^{\bullet}_C(\underline{x}, \underline{y}) = \sum_{\underline{\beta} \in B_2^{q_i}} [\, \Phi_C(\underline{x}, \underline{\beta}) \cdot \Phi^{+}_{Ci}(\underline{x}, \underline{\beta}, \underline{y}) \,].$$

Thus by Theorem 4.1, $\underline{y}(\underline{x})$ is realized if

$$\bar{\Phi}_C(\underline{x}, \underline{y}) + \Phi(\underline{x}, \underline{y}) = 1 \quad \text{identically.}$$

After substitution

$$\prod_{\underline{\beta} \in B_2^{q_i}} (\bar{\Phi}_C(\underline{x}, \underline{\beta}) + \bar{\Phi}^+_{Ci}(\underline{x}, \underline{\beta}, \underline{y})) + \Phi(\underline{x}, \underline{y})$$

$$= \prod_{\underline{\beta} \in B_2^{q_i}} (\bar{\Phi}_C(\underline{x}, \underline{\beta}) + \bar{\Phi}^+_{Ci}(\underline{x}, \underline{\beta}, \underline{y}) + \Phi(\underline{x}, \underline{y})) = 1 \tag{4.9}$$

The equality must be satisfied for all $\underline{x} \in X$ and $\underline{y} \in B_2^q$. Therefore, from (4.9)

$$\bar{\Phi}_C(\underline{x}, \underline{\beta}) + \bar{\Phi}^+_{Ci}(\underline{x}, \underline{\beta}, \underline{y}) + \Phi(\underline{x}, \underline{y}) = 1 \quad \text{identically}$$

for all $\underline{\beta} \in B_2^{q_i}$ and $\underline{y} \in B_2^q$. In other words,

$$\prod_{\underline{y} \in B_2^q} [\bar{\Phi}_C(\underline{x}, \underline{\beta}) + \bar{\Phi}^+_{Ci}(\underline{x}, \underline{\beta}, \underline{y}) + \Phi(\underline{x}, \underline{y})]$$

$$= \bar{\Phi}_C(\underline{x}, \underline{\beta}) + \underbrace{\prod_{\underline{y} \in B_2^q} [\bar{\Phi}^+_{Ci}(\underline{x}, \underline{\beta}, \underline{y}) + \Phi(\underline{x}, \underline{y})]}_{\Phi_i(\underline{x}, \underline{\beta})} = 1, \ \forall \underline{\beta}.$$

It is equivalent to

$$\Phi_C(\underline{x}, \underline{\beta}) \leq \Phi_i(\underline{x}, \underline{\beta}),$$

which must be satisfied for all $\underline{\beta} \in B_2^{q_i}$; however, that is guaranteed by the initial assumption. Therefore, C completes the realization of Φ.

To prove the converse, let it be assumed that C completes the realization. Then the equality (4.9) is satisfied identically by Lemma 4.1, which leads to $\Phi_C(\underline{x}, \boldsymbol{\beta}) \leqslant \Phi_i(\underline{x}, \boldsymbol{\beta})$ being satisfied for all $\boldsymbol{\beta} \in B_2^{q_i}$. Hence

$$\sum_{\boldsymbol{\beta} \in B_2^{q_i}} \Phi_C(\underline{x}, \boldsymbol{\beta}) \cdot m_{\boldsymbol{\beta}}(\underline{y}^i) \leqslant \sum_{\boldsymbol{\beta} \in B_2^{q_i}} \Phi_i(\underline{x}, \boldsymbol{\beta}) \cdot m_{\boldsymbol{\beta}}(\underline{y}^i).$$

$$\underline{\underline{\Phi_C(\underline{x}, \underline{y}^i) \leqslant \Phi_i(\underline{x}, \underline{y}^i)}}$$

(1) The proof of the existence of a circuit is derived from the proof of 2, as follows:

Given any circuit C then its characteristic equation $\Phi_C(\underline{x}, \underline{y}^i) = 1$ is always consistent for $\underline{x} \in X$ (Definition 4.1). Thus if $\Phi_i(\underline{x}, \underline{y}^i) = 1$ is inconsistent, i.e., $\Phi_i(\underline{\alpha}, \underline{y}^i) = 0$ identically for some $\underline{\alpha} \in X$, then $\Phi_C(\underline{\alpha}, \underline{y}^i) \leqslant \Phi_i(\underline{\alpha}, \underline{y}^i) = 0$. However, it implies $\Phi_C(\underline{\alpha}, \underline{y}^i) = 0$ identically, meaning that $\Phi_C(\underline{x}, \underline{y}^i) = 1$ is inconsistent at $\underline{\alpha}$, a contradiction.

Q.E.D.

Remark : If $\Phi^{+}_{Ci}(\underline{x}, \underline{y}^{i}, \underline{y}) = 1$ has a unique solution $\underline{y}(\underline{x}, \underline{y}^{i})$, that is, it represents a combinational circuit, then

$$\Phi_{i}(\underline{x}, \underline{y}^{i}) = \sum_{\underline{\beta} \in B_2^q} [\ \Phi^{+}_{Ci}(\underline{x}, \underline{y}^{i}, \underline{\beta}) \cdot \Phi(\underline{x}, \underline{\beta})\ ]$$

$$= \prod_{\underline{\beta} \in B_2^q} [\ \overline{\Phi}^{+}_{Ci}(\underline{x}, \underline{y}^{i}, \underline{\beta}) + \Phi(\underline{x}, \underline{\beta})\ ].$$

A rather interesting consequence is obtained by applying Lemma 4.2 for $i = 1, \ldots, k$, namely, it can be shown that if the ordering

$$\Phi_{C1}(\underline{x}, \underline{y}^1) \leqslant \{\ \Phi_{C2}(\underline{x}, \underline{y}^1, \underline{y}^2) \leqslant [\ \Phi_{C3}(\underline{x}, \underline{y}^2, \underline{y}^3) \quad \ldots\ldots$$

$$\ldots\ldots \leqslant (\ \Phi_{Ck}(\underline{x}, \underline{y}^{k-1}, \underline{y}^k) \leqslant \Phi(\underline{x}, \underline{y}^k)) \ldots]\} \quad (4.10)$$

is satisfied for all $\underline{x} \in X$, $\underline{y}^i \in B_2^{q_i}$, $i = 1, \ldots, k$, then the cascade network of Figure 4.1 realizes the output characteristic function $\Phi(\underline{x}, \underline{y})$.

By looking at the network of Figure 4.1 from the C_1 end, and by assuming that the circuits $C_1, C_2, \ldots, C_{i-1}$ are known, then the output characteristic function $\Phi'_i(\underline{x}, \underline{y}^{i-1}, \underline{y})$ which defines the permissible states of the remaining portion of the network $(C_i, \ldots, C_k)$ can be derived as follows:

Lemma 4.3 : Determination of $\Phi_i'(\underline{x}, \underline{y}^{i-1}, \underline{y})$. (Figure 4.1)

Let $\Phi^*_{Ci-1}(\underline{x}, \underline{y}^{i-1})$ be the overall circuit characteristic function of $C_1, C_2, \ldots, C_{i-1}$ combined. Then the output characteristic function $\Phi_i'(\underline{x}, \underline{y}^i, \underline{y})$ of the rest of the cascade is given as

$$\Phi_i'(\underline{x}, \underline{y}^{i-1}, \underline{y}) = \overline{\Phi}^*_{Ci-1}(\underline{x}, \underline{y}^{i-1}) + \Phi(\underline{x}, \underline{y}) \qquad (4.11)$$

That is, a circuit C characterized by $\Phi_C(\underline{x}, \underline{y}^{i-1}, \underline{y})$ will complete the realization of $\Phi(\underline{x}, \underline{y})$ iff

$$\Phi_C(\underline{x}, \underline{y}^{i-1}, \underline{y}) \leqslant \Phi_i'(\underline{x}, \underline{y}^{i-1}, \underline{y}) .$$

Note, that $\Phi_i' = 1$ is always consistent - any circuit which realizes Φ will realize Φ_i' too.

Proof : Assume $\Phi_C(\underline{x}, \underline{y}^{i-1}, \underline{y}) \leqslant \Phi_i'(\underline{x}, \underline{y}^{i-1}, \underline{y})$. By Lemma 2.3 and 4.1 the output characteristic function $\Phi(\underline{x}, \underline{y})$ is realized if

$$\sum_{\underline{\beta}^{i-1} \in B_2^{q_i-1}} [\Phi^*_{Ci-1}(\underline{x}, \underline{\beta}^{i-1}) \cdot \Phi_C(\underline{x}, \underline{\beta}^{i-1}, \underline{y})] \leqslant \Phi(\underline{x}, \underline{y}) \qquad (4.12)$$

That is, if

$$\overline{\Phi}^*_{Ci-1}(\underline{x}, \underline{\beta}^{i-1}) + \overline{\Phi}_C(\underline{x}, \underline{\beta}^{i-1}, \underline{y}) + \Phi(\underline{x}, \underline{y}) = 1$$

identically for all $\underline{\beta}^{i-1} \in B_2^{qi-1}$, which is equivalent to

$$\Phi_C(\underline{x}, \underline{\beta}^{i-1}, \underline{y}) \leq \bar{\Phi}^*_{Ci-1}(\underline{x}, \underline{\beta}^{i-1}) + \Phi(\underline{x}, \underline{y}) = \Phi'_i(\underline{x}, \underline{\beta}^{i-1}, \underline{y}).$$

However, that relation is satisfied by the initial assumption, thus Φ is realized.

The converse follows from (4.12) by a similar procedure as in the proof to Lemma 4.2.

Q.E.D.

In order to consider a general type of a network where the function $\Phi(\underline{x}, \underline{y})$ is realized by a combination of a cascade and parallel interconnection of circuit modules (possibly with feed forward lines), let the purely parallel case as shown in Figure 4.2 be analyzed first. The primary inputs $\underline{x}$ are assumed to represent all input lines, some possibly coming from the preceeding logic levels of a cascade. The outputs $\underline{y}_1$, $\underline{y}_2$ of the circuits C_1, C_2, respectively, form a disjoint partition on the set $\underline{y}$, i.e.,

$$\underline{y} = \underline{y}_1 \cup \underline{y}_2, \qquad \underline{y}_1 \cap \underline{y}_2 = \emptyset .$$

Furthermore, let $\Phi_{C1}(\underline{x}, \underline{y}_1)$ and $\Phi_{C2}(\underline{x}, \underline{y}_2)$ be the corresponding circuit characteristic functions of C_1 and C_2. Since the circuits operate in parallel, the overall circuit characteristic function is the characteristic function of the system of simultaneous equations

$$\Phi_{C1}(\underline{x}, \underline{y}_1) = 1 \quad \text{and} \quad \Phi_{C2}(\underline{x}, \underline{y}_2) = 1 ,$$

that is,

$$\Phi_{C1}(\underline{x}, \underline{y}_1) \cdot \Phi_{C2}(\underline{x}, \underline{y}_2) .$$

Thus Φ is realized if

$$\Phi_{C1}(\underline{x}, \underline{y}_1) \cdot \Phi_{C2}(\underline{x}, \underline{y}_2) \leqslant \Phi(\underline{x}, \underline{y}_1, \underline{y}_2) .$$

Assuming now that only C_2 is known, then the output characteristic function $\Phi_1(\underline{x}, \underline{y}_1)$, describing the limits on the states C_1 may at most assume is obtained as :

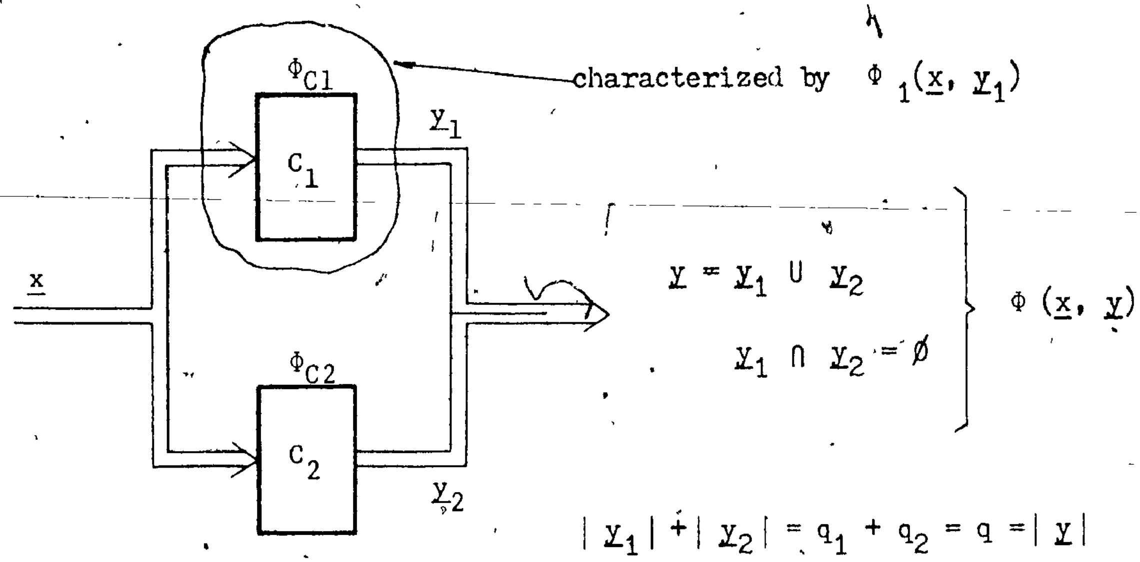


FIGURE 4.2. PARALLEL REALIZATION.

<u>Lemma 4.4</u> : Determination of Φ_1 in parallel realization (Figure 4.2).

Assuming C_2 being known then

$$\Phi_1(\underline{x}, \underline{y}_1) = \prod_{\underline{\beta} \in B_2^{q2}} [\ \overline{\Phi}_{C2}(\underline{x}, \underline{\beta}) + \Phi(\underline{x}, \underline{y}_1, \underline{\beta})\ ] \qquad (4.13)$$

That is,

(1) The realization of Φ may be completed by some C_1 iff $\Phi_1(\underline{x}, \underline{y}_1) = 1$ is consistent with respect to solutions $\underline{y}_1(\underline{x})$.

(2) A circuit C_1 represented by $\Phi_{C1}(\underline{x}, \underline{y}_1)$ will complete the realization of Φ iff

$$\Phi_{C1}(\underline{x}, \underline{y}_1) \leq \Phi_1(\underline{x}, \underline{y}_1) .$$

<u>Proof</u> :

(1) Similar to proof of (1) in Lemma 4.2.

(2) Assume $\Phi_{C1}(\underline{x}, \underline{y}_1) \leq \Phi_1(\underline{x}, \underline{y}_1)$. Φ would be realized if

$$[\ \Phi_{C1}(\underline{x}, \underline{y}_1) \cdot \Phi_{C2}(\underline{x}, \underline{y}_2)\ ] \leq \Phi(\underline{x}, \underline{y}_1, \underline{y}_2) \qquad (4.14)$$

that is if for all states of $\underline{y}_1$ and $\underline{y}_2$

$$\overline{\Phi}_{C1}(\underline{x}, \underline{y}_1) + \overline{\Phi}_{C2}(\underline{x}, \underline{y}_2) + \Phi(\underline{x}, \underline{y}_1, \underline{y}_2) = 1$$

identically. Or equivalently, if

$$\prod_{\underline{\beta} \in B_2^{q2}} [\ \overline{\Phi}_{C1}(\underline{x}, \underline{y}_1) + \overline{\Phi}_{C2}(\underline{x}, \underline{\beta}) + \Phi(\underline{x}, \underline{y}_1, \underline{\beta})\ ]$$

$$= \overline{\Phi}_{C1}(\underline{x}, \underline{y}_1) + \prod_{\underline{\beta} \in B_2^{q2}} [\ \overline{\Phi}_{C2}(\underline{x}, \underline{\beta}) + \Phi(\underline{x}, \underline{y}_1, \underline{\beta})\ ] = 1$$

$$\updownarrow$$

$$\Phi_{C1}(\underline{x}, \underline{y}_1) \leqslant \Phi_1(\underline{x}, \underline{y}_1),$$

which is satisfied by the initial assumption.

To prove the converse - assume that C_1 completes the realization of Φ. Then (4.14) holds, by Lemma 4.1, and (4.14) is equivalent to $\Phi_{C1} \leqslant \Phi_1$, as shown above.

Q.E.D.

<u>Remark</u>: If $\Phi_{C2}(\underline{x}, \underline{y}_2) = 1$ has a unique solution $\underline{y}_2(\underline{x})$, i.e., it represents a combinational circuit, then (4.13) can be replaced by a simpler expression

$$\Phi_1(\underline{x}, \underline{y}_1) = \sum_{\underline{\beta} \in B_2^{q2}} [\ \Phi_{C2}(\underline{x}, \underline{\beta}) \cdot \Phi(\underline{x}, \underline{y}_1, \underline{\beta})\ ].$$

Finally, a general theorem of functional decomposition can be stated as a combined application of the preceeding three Lemmas 4.2, 4.3 and 4.4. For that purpose, let it be assumed that an output characteristic function $\Phi(\underline{x}, \underline{y})$ is to be realized by a network as shown in Figure 4.3. Furthermore, it is assumed that a partial realization of Φ is attempted using some particular circuits C_1, C_2^1 and C_3, and thus it is required to determine the permissible steady output states for C_2^2, so as to complete the realization. That is the output characteristic function $\Phi_2^2(\underline{x}, \underline{y}^1, \underline{y}_2^2)$ is sought.

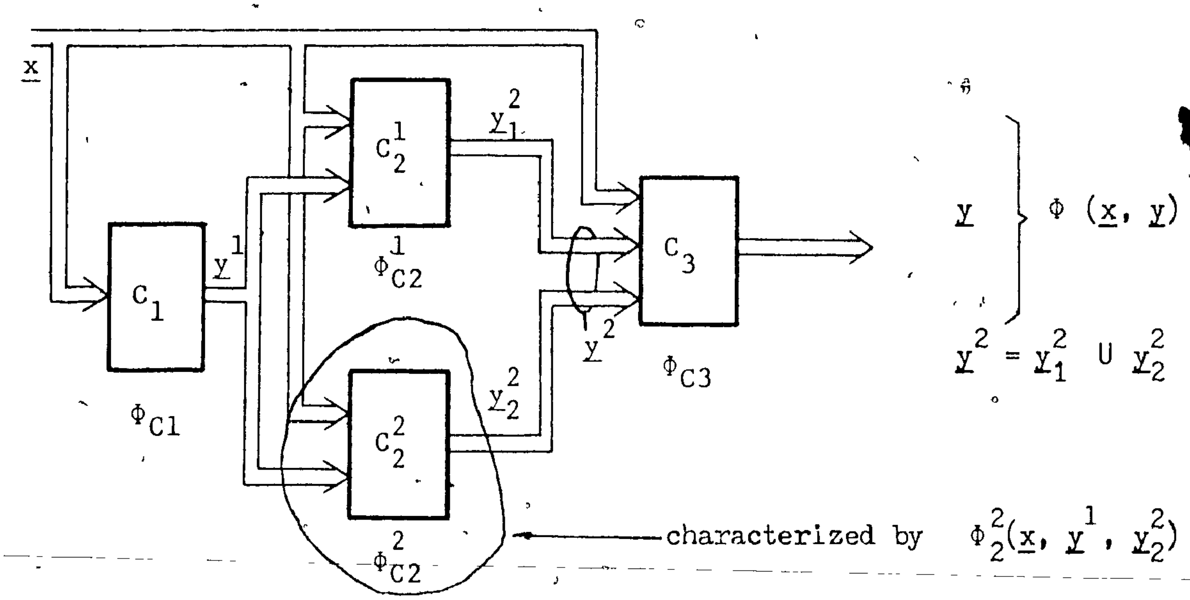


FIGURE 4.3. ILLUSTRATION FOR THEOREM 4.2.

Theorem 4.2 : Decomposition. (Figure 4.3).

Let $\Phi(\underline{x}, \underline{y})$ be an output characteristic function to be partially realized using some circuits C_1, C_2^1, C_3 represented by their circuit characteristic functions $\Phi_{C1}(\underline{x}, \underline{y}^1)$, $\Phi_{C2}^1(\underline{x}, \underline{y}^1, \underline{y}^2)$, $\Phi_{C3}(\underline{x}, \underline{y}^2, \underline{y})$, respectively. Then the output characteristic function $\Phi_2^2(\underline{x}, \underline{y}^1, \underline{y}_2^2)$ describing the permissible states of the still unknown circuit C_2^2 is given as

$$\Phi_2^2(\underline{x}, \underline{y}^1, \underline{y}_2^2) = \prod_{\underline{\beta} \in B_2^{q21}} [\bar{\Phi}_{C1}(\underline{x}, \underline{y}^1) + \Phi_{C2}^1(\underline{x}, \underline{y}^1, \underline{\beta})$$

$$+ \prod_{\underline{\gamma} \in B_2^{q3}} (\bar{\Phi}_{C3}(\underline{x}, \underline{\beta}, \underline{y}_2^2, \underline{\gamma}) + \Phi(\underline{x}, \underline{\gamma}))] \quad (4.15)$$

That is,

(1) There exists a circuit C_2^2 which would complete the realization of Φ iff $\Phi_2^2(\underline{x}, \underline{y}^1, \underline{y}_2^2) = 1$ is consistent with respect to solutions $\underline{y}_2^2(\underline{x}, \underline{y}^1)$.

(2) A particular C_2^2 represented by $\Phi_{C2}^2(\underline{x}, \underline{y}^1, \underline{y}_2^2)$ will complete the realization iff

$$\Phi_{C2}^2(\underline{x}, \underline{y}^1, \underline{y}_2^2) \leqslant \Phi_2^2(\underline{x}, \underline{y}^1, \underline{y}_2^2) \quad (4.16)$$

Proof : Apply Lemma 4.2 to get the characteristic function of the combined circuits C_1, C_2^1, C_2^2, then Lemma 4.3 to characterize (C_2^1, C_2^2), and finally Lemma 4.4 to generate Φ_2^2 as specified. The existence of solution condition (1) is carried over from Lemma 4.2 and 4.4.

Q.E.D.

For further illustration of the methodology the following corollary is included here.

Corollary 1 :

Assuming that the circuit C_2^2 is known as a partial realization of Φ, the output characteristic function $\Phi^*(\underline{x}, \underline{y}^1, \underline{y}_2^2, \underline{y})$ describing the rest of the network is obtained as

$$\Phi^*(\underline{x}, \underline{y}^1, \underline{y}_2^2, \underline{y}) = \bar{\Phi}_{C2}^2(\underline{x}, \underline{y}^1, \underline{y}_2^2) + \Phi(\underline{x}, \underline{y}) \qquad (4.17)$$

That is, if C_1, C_2^1, C_3 is represented by the overall characteristic function $\Phi_C^*(\underline{x}, \underline{y}^1, \underline{y}_2^2, \underline{y})$ then Φ is realized iff $\Phi_C^* \leqslant \Phi^*$. .

Proof : The proof is similar to those for Lemma 4.2, 4.3 and 4.4.

Remarks :

(1) Depending on the presence of either of the circuits C_1, C_2^2 C_3 in Figure 4.3, the theorem degenerates into one of the preceeding Lemmas.

(2) All the circuits involved may in general be of a sequential character, possibly with oscillations, provided that the corresponding circuit characteristic functions properly

describe their steady states. (Definition 4.1). In some cases, however, the theorem places a stronger condition on the realization than necessary. It is due to the fact that some input states may never occur (they are not generated by the preceeding circuit as its outputs), and thus some input sequences may not ever occur either, even if the problem inputs $\underline{x}$ are varied at random. Consequently, some of the steady output states of the particular sequential circuit may not be reachable, and hence the circuit could then be represented by its effective characteristic function $\Phi_{C\ eff}$ such that $\Phi_{C\ eff} \leq \Phi_C$. It means, that the Theorem 4.4 as well as the Lemmas 4.2 and 4.4 become sufficient only, since there certainly may exist cases such that $\Phi_C \not\leq \Phi$, but $\Phi_{C\ eff} \leq \Phi$. However, $\Phi_{C\ eff}$ cannot be obtained a priori, especially in Lemma 4.2, since the preceeding circuits are not yet known at the time of generating their output characteristic function. In case of Lemma 4.3, though, $\Phi_{C\ eff}$ could possibly be obtained using Φ_g (Definition 3.3).

(3) A complex multilevel realization of $\underline{y}(\underline{x})$ can be obtained by a repeated application of the decomposition theorem or

its degenerate forms, the lemmas.

(4) Any feed-forward line from a lower stage to a higher one (e.g. from C_1 to G_2) can be incorporated as a circuit described by the characteristic function of the equation of a pure interconnection (e.g. $y_j^2 = y_j^1$).

(5) Any combinational circuit (without feedback) which is to realize an output characteristic function $\Phi(\underline{x}, \underline{y})$ can be obtained as an elementary solution of the equation $\Phi(\underline{x}, \underline{y}) = 1$ by any method mentioned in Chapter 2. In such a case then, the Lemmas 4.2 and 4.4 represent implicitly the decomposition procedure as used in [5, 16, 19, 20]. More precisely, the particular decompositions are obtained via the expressions as stated in the Remarks following the above referenced lemmas.

The last Theorem (4.2) has completed the discussion covering the realization and decomposition of combinational switching functions. The next section will investigate the inside structure of a sequential circuit which is to realize a given output characteristic function. Namely, a more detailed study of the presence and necessity of feedback loops in such circuits will be performed.

4.3 Necessity of Loops, Degenerate Circuits

In the previous sections, all circuits were assumed to be generally of the type shown in Figure 3.1 (possibly with feedback loops, stable or oscillatory), and represented by an overall circuit characteristic function relating the steady state outputs with the input stimuli. Further investigation will now be made into the necessity of closed feedback loops inside such circuits. It will be shown, what the character of a circuit with loops should be, in order to bring any saving on gates as compared to purely combinational circuits realizing the same function. That such circuits exist was shown in examples by Kautz [14] and others [12, 30].

Let it be assumed that an output characteristic function $\Phi(\underline{x}, \underline{y})$ is to be realized by a circuit C represented by $\Phi_C(\underline{x}, \underline{y})$. (Definition 4.1, 4.2, Lemma 4.1). The discussion can be limited to the case where the circuit $\underline{f}(\underline{x}, \underline{z}')$ of Figure 3.1 is just a pure interconnection, i.e., $\underline{y} = \underline{z}'$, because any other case may be transformed into this one by applying Lemma 4.2 with $k = 2$ and C_2 being $\underline{f}$. Thus the new output characteristic function is to be realized by the circuit $\underline{g}$. Therefore, in general, the circuit C may be represented by $\underline{y}' = \underline{g}(\underline{x}, \underline{y})$, $|\underline{y}| = q$ (a Moore machine), where $\underline{y}$ is the present and $\underline{y}'$ the next output/internal state of C. Also, $\Phi_C(\underline{x}, \underline{y}) = \Phi_g^C(\underline{x}, \underline{y})$ (Definition 3.12, 4.1).

Theorem 4.3 : Degenerate circuits with feedback.

Let $\underline{y}' = \underline{g}(\underline{x}, \underline{y})$ represent a sequential circuit C realizing an output characteristic function $\Phi(\underline{x}, \underline{y})$. If there exists a state $\underline{Y} \in B_2^q$ such that

$$\Phi(\underline{x}, \underline{g}(\underline{x}, \underline{Y})) = 1, \qquad (4.18)$$

identically, then the circuit $\underline{g}$ is redundant in the sense that it can be made to degenerate into a purely combinational circuit (without feedback) represented by $\underline{g}^*(\underline{x}) = \underline{g}(\underline{x}, \underline{Y})$, also realizing Φ and having lower complexity than $\underline{g}$.

Note that the state $\underline{g}^*(\underline{x}) = \underline{g}(\underline{x}, \underline{Y})$ for some $\underline{x} \in X$ may be a transient state of the original circuit.

Proof : It has to be shown that (1) $\underline{g}^*(\underline{x})$ realizes Φ, and (2) that the circuit $\underline{g}^*(\underline{x})$ obtained by freezing the feedback inputs of $\underline{g}(\underline{x}, \underline{y})$ at $\underline{y} = \underline{Y}$ has a lower complexity (gate-lead cost).

(1). Since $\underline{g}^*$ represents a purely combinational circuit then

$$\Phi^*_C(\underline{x}, \underline{y}) = \prod_{i=1} (y_i \cdot g_i^*(\underline{x}) + \bar{y}_i \cdot \bar{g}_i^*(\underline{x}))$$

(Definition 4.1, Theorem 3.1.)

Assuming

$$\Phi(\underline{x}, \underline{g}(\underline{x}, \underline{y})) = \Phi(\underline{x}, \underline{g}^*(\underline{x})) = 1$$

identically, then $\Phi^*_C(\underline{x}, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y})$, because $\Phi^*_C(\underline{x}, \underline{y}) = 1$ has a unique solution. Therefore, $\underline{g}^*(\underline{x})$ realizes Φ (Lemma 4.1).

(2) Let it be initially assumed that the internal structure of the circuit $\underline{g}(\underline{x}, \underline{y})$ corresponds to a simple sum of product form of the function $\underline{g}$:

$$y_i' = g_i(\underline{x}, \underline{y}) = \sum_{j \in T_i} s_{ij}(\underline{x}) \cdot t_{ij}(\underline{y}), \quad i = 1, \ldots, q$$

where $s_{ij}(\underline{x})$, $t_{ij}(\underline{y})$ are the j^{th} (in the sequence of writing) products in $\underline{x}$ and $\underline{y}$, respectively, T_i being the index set of terms for each g_i. Thus for every $j \in T_i$, $s_{ij}(\underline{x}) \cdot t_{ij}(\underline{y})$ is an implicant of g_i.

Assuming (4.18),

$$g_i^*(\underline{x}) = g_i(\underline{x}, \underline{y}) = \sum_{j \in T_i} s_{ij}(\underline{x}) \cdot t_{ij}(\underline{y})$$

That is,

$$g_i^*(\underline{x}) = \sum_{j \in T_i^*} s_{ij}(\underline{x}), \qquad i = 1, \ldots, q.$$

where

$$T_i^* = \{ j \mid j \in T_i \text{ and } t_{ij}(\underline{y}) = 1 \}.$$

Clearly $T_i^* \subseteq T_i$, for all i. Therefore, the implementation of the circuit $\underline{g}^*$ will require at most as many AND gates as $\underline{g}$, but each gate requiring lower fan-in because of the missing terms $t_{ij}(\underline{y})$. Also, the OR gate would possibly have a lower fan-in. Consequently, the resulting circuit $\underline{g}^*$ will be less complex than the circuit $\underline{g}$. Moreover, the circuit would probably have a faster response than that with feedback loops, since $\underline{g}(\underline{x}, \underline{y})$ may have a number of transient states to go through before reaching a steady state.

If now the internal structure of the circuit $\underline{g}$ is represented by some form other than $\Sigma\Pi$, a procedure similar to that above can again be applied to produce $\underline{g}^*$, whose implementation will always require lower fan-in of the gates even if the total number of gates should stay the same as in $\underline{g}$.

Q.E.D.

The theorem yields some rather interesting results which are summarized in the following corollaries :

Corollary 1 :

Any circuit $\underline{y}' = \underline{g}(\underline{x}, \underline{y})$ with feedback having

$$\Phi_g^z(\underline{x}, \underline{y}') \leqslant \Phi(\underline{x}, \underline{y}') \quad ,$$

where Φ_g^z is the next state characteristic function (Definition 3.4), is degenerate with respect to that Φ. Thus all 0-transition circuits (Definition 3.10) realizing Φ are degenerate, since

$$\Phi_g^C = \Phi_g^z$$

is such a case. (Any choice of $\underline{y} \in Y$ will satisfy (4.18)).

Remark : Circuits with $\Phi_g^z \leqslant \Phi$ will be referred to as direct transition circuits, because any transition from an unstable state leads directly to a state covered by Φ.

Corollary 2 :

Any sequential circuit $\underline{y}' = \underline{g}(\underline{x}, \underline{y})$ which realizes some $\Phi(\underline{x}, \underline{y}')$ and is not degenerate must satisfy :

(i) $\Phi_C(\underline{x}, \underline{y}') \leqslant \Phi(\underline{x}, \underline{y}')$ so as to realize Φ.

(ii) For each $\underline{y} \in B_2^q$ there exists a state $\underline{\alpha} \in X$ such that

$$\Phi(\underline{\alpha}, \underline{g}(\underline{\alpha}, \underline{y})) = 0 \qquad (4.19)$$

(If not then that $\underline{y}$ satisfies (4.18) $\rightarrow$ degeneration).

(iii) The (next) state $\underline{g}(\underline{\alpha}, \underline{y})$ satisfying (4.19) is a transient state. (Otherwise Φ would not be realized - condition(i)).

(iv) $|\underline{y}| \geq 2 \rightarrow$ there exists no non-degenerate sequential circuit if $|\underline{y}| = 1$.

Corollary 3 :

As a consequence of Corollary 2 above, any circuit $\underline{g}$ with feedback which is to bring about any savings on gates as compared to a purely combinational realization of Φ (without feedback) must have transient states not covered by Φ. That is, such a circuit is inherently hazardous during its transition periods. If this would be considered as an undesirable property for some applications, then the circuit having a minimal number of gates must be sought only among the purely combinational realizations of Φ. As such, its functional representation $\underline{y}' = \underline{g}(\underline{x})$ is one of the elementary solutions to the equation $\Phi(\underline{x}, \underline{y}') = 1$. (This approach was used so far in all existing synthesis procedures, either explicitly or implicitly).

The design of non-degenerate sequential circuits satisfying certain criteria of optimality is considered by the author as a topic of its own, and as such it is beyond the scope of this presentation. Nevertheless, an example of such a circuit was shown by Kautz [14], and it will be analyzed in Example 5.3 of Chapter 5 using the author's methodology. As will be seen therein, the transient states are in fact induced by the spatial iterative structure of the circuit. Most of these states are not covered by the corresponding function Φ. It is a non-degenerate sequential circuit with hazardous transition periods.

Although it was shown that direct transition circuits with feedback are always degenerate, and as such they seem to have very little practical value when minimal realizations are sought, their generation is at least of theoretical interest, especially since they relate directly to the general solution of the output characteristic equation $\Phi(\underline{x}, \underline{y}') = 1$. Therefore, a discussion covering the topic will conclude this chapter.

Theorem 4.4 : Direct transition circuits.

Let $\Phi(\underline{x}, \underline{y}') = 1$ be a consistent output characteristic equation, and $\underline{y}' \cong \underline{n}(\underline{x}, \underline{p})$ its general solution (Definition 2.5).

Furthermore, let $\underline{p} = \underline{p}(\underline{x}, \underline{y})$ be any Boolean function of $\underline{x}$ and $\underline{y}$. Then the function

$$\underline{y}' = \underline{g}(\underline{x}, \underline{y}) = \underline{n}(\underline{x}, \underline{p}(\underline{x}, \underline{y}))$$

represents a direct transition circuit realizing $\Phi(\underline{x}, \underline{y})$. Conversely, for any direct transition circuit $\underline{y}' = \underline{g}(\underline{x}, \underline{y})$ realizing Φ there exists a function $\underline{p}^*(\underline{x}, \underline{y})$ such that

$$\underline{g}(\underline{x}, \underline{y}) = \underline{n}(\underline{x}, \underline{p}^*(\underline{x}, \underline{y})).$$

Proof : Assume the Löwenheim's form of $\underline{n}(\underline{x}, \underline{p})$ (Section 2.3.3):

$$\underline{y}' = \underline{n}(\underline{x}, \underline{p}) = \Phi(\underline{x}, \underline{p}) \cdot p + \overline{\Phi}(\underline{x}, \underline{p}) \cdot \xi(\underline{x})$$

where $\xi(\underline{x})$ is any elementary solution to $\Phi(\underline{x}, \underline{y}') = 1$. Then

$$\underline{g}(\underline{x}, \underline{y}) = \Phi(\underline{x}, \underline{p}(\underline{x}, \underline{y})) \cdot \underline{p}(\underline{x}, \underline{y}) + \overline{\Phi}(\underline{x}, \underline{p}(\underline{x}, \underline{y})) \cdot \xi(\underline{x}).$$

To prove the first part, it has to be shown that for any present state $\underline{y}^*$ the next state $\underline{y}'$ is such that $\Phi(\underline{\alpha}, \underline{y}') = 1$, $\underline{\alpha} \in B_2^r$,

(a) $\underline{y}^*$ is such that

$$\Phi(\underline{\alpha}, \underline{p}(\underline{\alpha}, \underline{y}^*)) = 1,$$

then

$$\underline{y}' = \underline{p}(\underline{\alpha}, \underline{y}^*) \rightarrow \Phi(\underline{\alpha}, \underline{y}') = 1.$$

(b) $\underline{y}^*$ is such that

$$\Phi(\underline{\alpha}, \underline{p}(\underline{\alpha}, \underline{y}^*)) = 0,$$

then

$$\underline{y}' = \xi(\underline{x}).$$

But $\xi(\underline{x})$ is an elementary solution to

$$\Phi(\underline{x}, \underline{y}) = 1 \quad \rightarrow \quad \Phi(\underline{\alpha}, \underline{y}') = 1.$$

Thus all the states generated by $\underline{g}$ (steady and transient) are contained in Φ; therefore, that $\underline{g}$ is a direct transition circuit with respect to $\Phi(\underline{x}, \underline{y})$.

The converse of the theorem : Let $\underline{y}' = \underline{g}(\underline{x}, \underline{y})$ be a direct transition circuit. The choice of $\underline{p}^*(\underline{x}, \underline{y}) = \underline{g}(\underline{x}, \underline{y})$ will make

$$\underline{\eta}(\underline{x}, \underline{p}^*(\underline{x}, \underline{y})) = \underline{g}(\underline{x}, \underline{y});$$

$$\underline{\eta}(\underline{x}, \underline{g}(\underline{x}, \underline{y})) = \Phi(\underline{x}, \underline{g}(\underline{x}, \underline{y})) \cdot \underline{g}(\underline{x}, \underline{y}) + \bar{\Phi}(\underline{x}, \underline{g}(\underline{x}, \underline{y})) \cdot \xi(\underline{x}),$$

But $\underline{g}$ is a direct transition circuit, hence

$$\Phi(\underline{x}, \underline{g}(\underline{x}, \underline{y})) = 1,$$

$\forall \underline{x}$ and $\forall \underline{y}$, thus

$$\underline{\eta}(\underline{x}, \underline{g}(\underline{x}, \underline{y})) = \underline{g}(\underline{x}, \underline{y}).$$

Q.E.D.

The class of direct transition circuits could be enlarged by permitting the transitions from the states contained in

$$\bar{Y} = B_2^q - \left(\bigcup_{\underline{\alpha} \in B_2^r} Y_{\alpha} \right)$$

to be don't care (unspecified). However, if the circuit would accidentally enter a state contained in $\bar{Y}$ (say, at power-up time) then the response of the circuit is unpredictable (a reset mechanism required). The inclusion of $\bar{Y}$ states does not alter the result of Theorem 4.4, provided that $\underline{y} \in Y$, hence there seems to be no reason to elaborate on direct transition circuits any more. It should be noticed, though, that if $\underline{p}(\underline{x}, \underline{y})$ is chosen to depend on $\underline{x}$ only, i.e. $\underline{p}(\underline{x}, \underline{y}) = \underline{p}(\underline{x})$, then the resulting circuit is purely combinational - all elementary solutions to $\Phi(\underline{x}, \underline{y}) = 1$ can thus be obtained as in [5].

CHAPTER 5

COMPUTATION TECHNIQUES AND EXAMPLES

The purpose of this chapter is to show some examples illustrating the use and power of the methodology developed in the previous sections. First, however, it is necessary to specify certain techniques and tools, through which all the computations involved in solving actual problems could efficiently be formulated. Two approaches will be considered here - for problems with few variables (≤ 6) Marquand maps will be used to represent Boolean functions [5, 16, 17, 27, 31, 32], with all the necessary operations defined on them. Problems which involve a larger number of variables (> 6) will then be solved using algebraic techniques. That is, cubical complexes with the associated operations will be employed [7, 21, 24, 25, 28, 36]. The reason for selecting the algebraic approach for larger problems is rather obvious - map methods require the construction of large maps (6 variables $\rightarrow$ 64 bit map), even when the particular Boolean function is expressable algebraically as a single product term [26]. Also, the algebraic method provides a simple form for describing computational algorithms in a way similar to a programming language. In turn, an advantage of maps for smaller problems is their visual information content - properties of functions and mutual relations between them (e.g. $\leqslant$ relation) can easily be recognized by visual examination of the maps.

5.1 Map Techniques

From the large number of possible map representations of Boolean functions, the so called Marquand map will be used here because of its simplicity of construction; its format remains the same for different number of variables. Also, such a map is suitable for solving Boolean equations (Section 2.3.2) [5, 16, 18, 19, 26, 31].

The rows and columns of a Marquand map are labelled in an increasing order of minterm identifiers. E.g., considering 4 variables, the minterm $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ has the identifier $0000_2 = 0_{10}$; $x_1 x_2 x_3 x_4 \rightarrow 1111_2 = 15_{10}$. If the general circuit of Figure 3.1 is considered again as the model of the circuit structure desired, then four types of variables can be recognized:

(1) Input variables $\underline{x}$, $\underline{y}^{i-1}$.

(2) State variables - present state $\underline{z}$.

(3) State variables - next state $\underline{z}'$.

(4) Output variables $\underline{y}^i$.

The maps of a given problem will always be organized so that the current independent variables will label the columns, the dependent variables then the rows of the maps.

Operations on Maps

Negation of a function Φ	- bit by bit complement of the map
Intersection of $\Phi_1 \cdot \Phi_2$	- bit by bit AND of the maps Φ_1, Φ_2.
Union $\Phi_1 + \Phi_2$	- bit by bit OR of the Φ_1, Φ_2 maps.
$\Phi_1 \leq \Phi_2$ relation	- either $\Phi_1 \cdot \bar{\Phi}_2$ results in a map with all zeros or by inspection that any time Φ_1 contains a 1 in a particular bit then Φ_2 also has a 1 there.

If $\Phi(\underline{x}, \underline{y}) = 1$ is a B.E. with $\underline{x}$ being the independent and $\underline{y}$ the dependent variables; the map of Φ having the columns labelled by the states of $\underline{x}$, the rows by the states of $\underline{y}$, then : (Chapter 2), [16, 26, 21] .

(1) The equation is consistent for $\underline{y} = \underline{y}(\underline{x})$ solutions if there is no all zero column in the map.

(2) The equation has an identity solution $\underline{y} = \underline{y} \in B_2^q$ if there is a row containing all 1's.

(3) The equation has a unique solution if each column contains exactly a single 1.

(4) To obtain elementary solutions, the method of Svoboda [26, 31] may be used. For a general solution, the method of Brown [3] seems to be easily applicable, since the discriminant of the equation is the map of Φ.

All other operations needed for applying the theory presented in Chapters 3 and 4, such as formation of the characteristic functions, summation over all states of a vector variable, augmentation of the maps by redundant variables, etc., are rather trivial, and they will be shown implicitly in the examples presented in Section 5.4.

5.2 Cubical Complexes and Operations [7, 21, 25, 36]

Boolean functions can algebraically be represented using encoded product terms of a sum-of-product form of the particular function. Each such product can be topologically viewed as a multidimensional cube, its dimensions being determined by the number of missing variables in the term as compared to the total number of variables of the function. The encoding of variables in a term will be as follows:

i^{th} position in a term	the i^{th} variable
0	complemented
1	uncomplemented
x	missing

It will be assumed here, that the sequence of variables appearing in a term will always be $\underline{x}$, $\underline{z}$, $\underline{z}'$, $\underline{y}'$, $\underline{y}^2$, ..., $\underline{y}^k$. However, if some of the vector variables are not required by a particular problem then the corresponding columns will be left out in the cubical array. For instance, in a problem with

$$\underline{x} = (x_1, x_2, x_3); \quad \underline{z}, \ \underline{z}' = (z_1, z_2); \quad \underline{y} = (y_1, y_2)$$

the term $x_1 \bar{x}_2 \bar{z}_1 z_2 z_1' \bar{y}_2$ will be represented by a cube

$$C = \underbrace{1\ 0\ x}_{\underline{x}}\ \ \underbrace{0\ 1}_{\underline{z}}\ \ \underbrace{1\ x}_{\underline{z}'}\ \ \underbrace{x\ 0}_{\underline{y}} = c_{\underline{x}} \circ c_{\underline{z}} \circ c_{\underline{z}'} \circ c_{\underline{y}}$$

In any case, the current order of variables will be indicated with each example shown. A Boolean function in its $\Sigma\Pi$ form is thus represented by an array of the above mentioned cubes. An efficient encoding (binary) of the cubes, as well as a data structure for the arrays designed for computer aided processing, can be found in [36].

Operations on Cubical Arrays

Having determined the representation of Boolean functions, it remains to define the Boolean algebra equivalent operations on the data structure. A concise review of such operations will be presented here. More detail can be found in the forementioned references. Since the examples in Section 5.5 will be done by hand, all the operations will be performed as in [7]; however, for automated processing, the more efficient encoding as in Reference [36] should be used instead.

Let a, b, c represent cubes of n coordinates (dimension n), a_i, b_i, c_i the i^{th} coordinate of a, b, c, respectively, and let A, B, C be cubical arrays. Then the following operations can be defined [7]:

(1) Absorption:

$$a \subseteq b \quad \text{if} \quad (a_i \sqsubseteq b_i) = \varepsilon \quad \text{for all } i,$$

$$a \nsubseteq b \quad \text{if} \quad (a_i \sqsubseteq b_i) = \emptyset \quad \text{for any } i.$$

The coordinate subsumming table :

$a_i \subseteq b_i$		b_i: 0	1	x
	0	ε	$\emptyset$	ε
a_i	1	$\emptyset$	ε	ε
	x	$\emptyset$	$\emptyset$	ε

Absorb operation on an array

$B = \mathbf{A}(A)$ B is the absorbed array A

(cover equivalent)

(2) <u>Cube Union</u> : (Boolean OR)

If

$$A = \{ a^1, a^2, \ldots \} , \quad B = \{ b^1, b^2, \ldots \}$$

then

$$C = A \sqcup B = \mathbf{A}(a^1, a^2, \ldots, b^1, b^2, \ldots)$$

(3) <u>Cube Intersection</u> : (Boolean AND)

$$a \sqcap b = \begin{cases} \emptyset & (\text{empty if any } a_i \sqcap b_i = \emptyset \\ c & \text{otherwise, } c_i = a_i \sqcap b_i \end{cases}$$

The coordinate cube intersection table :

$\sqcap$		b_i 0	1	x
	0	0	$\emptyset$	0
a_i	1	$\emptyset$	1	1
	x	0	1	x

Intersection of two arrays :

$$C = A \sqcap B = \{\{A \sqcap b^1\} \sqcup \{A \sqcap b^2\} \sqcup \ \ldots \}$$

where

$$A \sqcap b = \{(a^1 \sqcap b) \sqcup (a^2 \sqcap b) \sqcup \ \ldots \}$$

(4) <u>Sharp Product</u> : (Boolean $A \cdot \bar{B}$)

$$a \# b = \begin{cases} a & \text{if } a \sqcap b = \emptyset, \text{ i.e., } a_i \# b_i = \emptyset \text{ for some } i \\ \emptyset & \text{if } a \sqsubseteq b, \text{ i.e., } a_i \# b_i = \varepsilon, \forall i \\ \bigsqcup_{i \in I} (a_1, a_2, \ldots, \bar{b}_i, \ldots, a_n) & \text{otherwise,} \end{cases}$$

where $I = \{i \mid a_i \# b_i = \alpha_i \in \{0, 1\}\}$

The coordinate sharp product table :

$a_i \# b_i$		b_i: 0	b_i: 1	b_i: x
	0	ε	$\emptyset$	ε
a_i	1	$\emptyset$	ε	ε
	x	1	0	ε

Then

$$A \# b = \{ \{a^1 \# b\} \sqcup \{a^2 \# b\} \sqcup \ldots \}$$

$$a \# B = \{ \ldots \{\{a \# b^1\} \# b^2\} \# \ldots \}$$

$$A \# B = \{ \ldots \{\{ A \# b^1\} \# b^2\} \ldots \}$$

$$= \{ \{a^1 \# B\} \sqcup \{a^2 \# B\} \ldots \}$$

(5) Bookkeeping Operations :

(a) The replacement operation $\leftarrow$:

$A \leftarrow \mathbf{A}(A)$ absorbed array A replaces A

$C \leftarrow A \sqcap B$ intersection of A, B placed in C.

(b) The delete operation $\mathbf{D}_\lambda(A)$ removes the columns identified by λ from the cubes of array A. $\mathbf{D}_{\underline{z}}(A)$ will delete the columns associated with $\underline{z}$.

Note that $\mathbf{D}_{\underline{z}}(A)$ is equivalent to $\sum_{\underline{z} \in B_2^n} (A)$ - the summation over all states of the variables $\underline{z}$.

(c) The insert operation $\mathsf{I}_j^i(A)$ inserts j new columns of $\underline{x}$'s to the right of the i^{th} column of A. If j not specified, only one column is inserted.

(d) The permute operation $\mathsf{P}_{i,j}(A)$ interchanges the i^{th} and the j^{th} columns of A.

(e) The change operation $\mathsf{C}_{\lambda}^{b_1 b_2 b_3}(A)$ replaces all 0's with b_1, 1's with b_2, and $\underline{x}$'s with b_3 in the columns of A identified by λ.

(f) The split operation S identifies and transfers to another array all cubes which subsume a given mask cube.

$A \leftarrow B \;\mathsf{S}\; a$ will transfer to A all cubes of b which subsume a.

(g) Special cubes :

U_n - n-dimensional space cube (n x's)

χ_{λ}^{b} - a cube of x's with $b \in \{0, 1\}$ in the columns identified by λ

ψ_{λ} - a cube of 0's with 1's in the columns identified by λ

$\emptyset$ - an empty cube (0 function)

(h) The Cartesian product $\times$ is used to append a fixed cube to each element of an array, i.e., $A \times U_n$ will append n columns of x's to each cube of A. $A \times \{0\ 1\}$ would append $\{0\ 1\}$ to each cube of A.

5.3 Related Algorithms

5.3.1 Algorithms for Chapter 3

Let the function $\underline{f}$, $\underline{g}$, $\underline{\phi}$, etc., be represented by their ON arrays for each component of the function vector, and in addition, by the DC arrays in case of incompletely specified functions [7, 21]. E.g., the function $\underline{y}(\underline{x})$ will be represented by the arrays ON_{ϕ_i}, DC_i, $i = 1, \ldots, q$. The so called function arrays of [7] will not be considered at the moment; however, the relationship between characteristic functions and function arrays will be discussed in Chapter 7. The sets X, Z of input stimuli and internal states will be described by a function (array) $D = D_x \circ D_z$ being equal to 1 for $\underline{x} \in \bar{X}$, $\underline{z} \in \bar{Z}$, and 0 otherwise. The characteristic functions of Chapter 3 are in fact completely specified functions, representing a mapping from a multi-dimensional space into B_2. Thus, only the ON arrays are needed to describe them completely.

Algorithm 5.1 : Formation of the Array Φ_g of $\Phi_g(\underline{x}, \underline{z}, \underline{z}')$:

(Definition 3.2)

$$\underline{g}(\underline{x}, \underline{z}) : \quad ON_{gi} = G_i = (G_x \circ G_z)_i, \quad i = 1, \ldots, n$$

$$D = D_x \circ D_z$$

χ_λ^b of n coordinates

(1) $\Phi_g \leftarrow U_{r+n+n}$.

(2) for $i = 1$ to n

(3) $\Phi_g \leftarrow \Phi_g \sqcap ((G_i \times \chi_i^1) \sqcup ((U_{r+n} \# G_i) \times \chi_i^0))$

(4) next i

(5) $\Phi_g \leftarrow \Phi_g \# (D \times U_n)$.

(6) end

Algorithm 5.2 : Formation of Φ_g^s : (Definition 3.3)

$$\underline{g}(\underline{x}, \underline{z}) : \quad G_i, \qquad i = 1, \ldots, n$$

$$D = D_x \circ D_z$$

χ_λ^b of $r + n$ coordinates

(1) $\Phi_g^s \leftarrow U_{r+n}$

(2) for $i = 1$ to n

(3) $\Phi_g^s \leftarrow \Phi_g^s \sqcap (U_{r+n} \# (\overbrace{(G_i \# \chi_i^1)}^{G_i \sqcap \chi_i^0} \sqcup (\chi_i^1 \# G_i)))$

(4) next i

(5) $\Phi_g^s \leftarrow \Phi_g^s \# D$

(6) end

Algorithm 5.3 : Formation of $\Phi_g^z(\underline{x}, \underline{z}')$: (Definition 3.4)

Use Algorithm 5.1 to generate Φ_g, then

(1) $\Phi_g^z \leftarrow A(D_{\underline{z}}(\Phi_g))$

(2) end

Algorithm 5.4 : Determination of Simple Oscillations : (Lemma 3.5)

Use Algorithm 5.2 to form Φ_g^s.

The states of $\underline{x}$ at which simple oscillations occur are covered by the array T obtained as follows:

(1) $T \leftarrow D_{z'}(\Phi_g^s)$; delete columns $r + 1$ to $r + n$

(2) $T \leftarrow U_r \# T$

(3) $T \leftarrow T \# D_x$

(4) if T empty then no simple oscillations, otherwise T contains the cover

(5) end

<u>Algorithm 5.5</u> : Determination of Φ_g^c :

If Theorem 3.1 is applicable then Φ_g^c is generated by Algorithm 5.2 or Algorithm 5.3. If an otherwise stable circuit has simple oscillations at the states of $\underline{x}$ covered by T (Algorithm 5.4) then an approximation to Φ_g^c can be obtained by :

$$\Phi_g^c \leftarrow \Phi_g^s \cup (\Phi_g^z \sqcap (T \times U_n))$$

In other cases (not very often) Φ_g^c must be obtained from a state transition table or using Corollary 1 to Theorem 3.1.

<u>Algorithm 5.6</u> : Algorithm for Theorem 3.2 : (Theorem 2.4, Corollary)

Use Algorithm 5.5 to form Φ_g^c.

$X_{\underline{x}}^b$ of $r + n$ coordinates.

(1) $T \leftarrow \emptyset$; initialize.

(2) for $i = 1$ to n

(3) $T \leftarrow T \cup (D_{\underline{z}'}(\Phi_g^c \sqcap X_{i+r}^0) \sqcap D_{\underline{z}'}(\Phi_g^c \sqcap X_{i+r}^1))$

(4) next i

(5) $T \cancel{\leftarrow T \# D_{\underline{x}}}$

(6) if T empty then $\Phi_g^c = 1$ has a unique solution $\rightarrow \underline{g}$ is combinational

(7) end

Algorithm 5.7 : Formation of Φ_f : (Definition 3.14)

$\underline{f}(\underline{x}, \underline{z}', \underline{y})$: F_i , $i = 1, \ldots, q$

χ_λ^b of q coordinates .

(1) $\Phi_f \leftarrow U_{r+n+q}$

(2) for $i = 1$ to q

(3) $\Phi_f \leftarrow \Phi_f \sqcap (F_i \times \chi_i^1) \sqcup ((U_{r+n} \# F_i) \times \chi_i^0))$

(4) next i

(5) end

Algorithm 5.8 : Algorithm for Theorem 3.3.

χ_λ^b of $r + q$ coordinates.

(1) $\Phi_c \leftarrow \Phi_g^c \times U_q$

(2) $\Phi_c \leftarrow \Phi_c \sqcap \Phi_f$

(3) $\Phi_c \leftarrow D_{\underline{z}'}(\Phi_c)$; Φ_c now contains the circuit char. function

(4) $T \leftarrow \emptyset$; initialize

(5) for $i = 1$ to q ; check for unique solution

(6) $T \leftarrow \cancel{T \sqcup} (D_{\underline{y}}(\Phi_c \sqcap \chi_{r+i}^0) \sqcap D_{\underline{y}}(\Phi_c \sqcap \chi_{r+i}^1))$

(7) next i

(8) $T \leftarrow T \# D_x$; limit $\underline{x}$ to X.

(9) if T empty then the circuit has combinational behaviour,

the characteristic function is in Φ_c.

(11) end

5.3.2 <u>Algorithms for Chapter 4</u>

<u>Algorithm 5.9</u> : Formation of the Circuit Characteristic Function

$\Phi_c(\underline{x}, \underline{y})$: (Definition 4.1)

Use step 1 to 4 of Algorithm 5.8, then

$$\Phi_c \leftarrow \Phi_c \# (D_x \times U_q)$$

<u>Algorithm 5.10</u> : Formation of $\Phi(\underline{x}, \underline{y})$ of $\underline{y}(\underline{x})$ (Theorem 4.1)

$$\left.\begin{array}{lll} \underline{y}(\underline{x}) & : & ON_{\phi i} \\ & & DC_i \\ & & OFF_i \end{array}\right\} \quad i = 1, \ldots, q$$

where

$$OFF_i = U_r \# (ON_{\phi i} \sqcup DC_i)$$

χ_λ^b of q coordinates

(1) $\Phi \leftarrow U_{r+q}$

(2) for $i = 1$ to q

(3) $\Phi \leftarrow \Phi \sqcap ((ON_{\phi\, i} \times X_i^1) \sqcup (OFF_i \times X_i^0) \sqcup (DC_i \times U_q))$.

(4) next i

(5) end

Line #3 could be replaced by

$$(3) \quad \Phi \leftarrow \Phi \sqcap ((ON_{\phi\, i} \times X_i^1) \sqcup ((U_r \# ON_{\phi\, i}) \times X_i^0) \sqcup (DC_i \times U_q))$$

which produces a cover equivalent array, in case that OFF_i arrays are not known. The calculations are then shorter than if OFF_i were to be calculated separately.

<u>Algorithm 5.11</u> : Realization of an Output Characteristic Function Φ by Φ_c : (Theorem 4.1)

(a) (1) $T \leftarrow \Phi_c \# \Phi$

(2) if T empty then C realizes Φ.

(3) end

(b) The computations involved in Lemmas 4.2, 4.3, 4.4 and Theorem 4.2 are quite simple, but it should be noted that the expression of the form

$$\Phi(\underline{x}, \underline{y}) = \prod_{\underline{z} \in B_2^n} [\, \overline{\Phi}_1(\underline{x}, \underline{y}, \underline{z}) + \Phi_2(\underline{x}, \underline{z}) \,]$$

(appearing in Lemma 4.2 and 4.4, as well as in Theorem 4.2) is best evaluated by double complementing it first, that is,

$$\Phi(\underline{x}, \underline{y}) = \overline{\sum_{\underline{z} \in B_2^n}} [\, \Phi_1(\underline{x}, \underline{y}, \underline{z}) \cdot \overline{\Phi}_2(\underline{x}, \underline{z}) \,] \ .$$

The reason is that such a form has a simple equivalent in terms of array operations, namely,

(1) $\Phi_2 \leftarrow I_q^r(\Phi_2)$; $|\underline{x}| = r, \ |\underline{y}| = q, \ |\underline{z}| = n$

(2) $\Phi \leftarrow U_{r+q} \# [\, D_{\underline{z}}(\Phi_1 \# \Phi_2) \,]$;

The resulting array Φ has an interesting property - it consists of all the prime implicants of the function $\Phi(\underline{x}, \underline{y})$. It is due to the last # operation performed [7]. Thus the size of the array Φ can be reduced by applying standard reduction algorithms.

<u>Algorithm 5.12</u> : Existence of Solution Test.

Let $\Phi(\underline{x}, \underline{y})$ be an output characteristic function (array) obtained either through Lemma 4.2, 4.4 or Theorem 4.2. The following simple algorithm will determine whether $\Phi = 1$ is consistent - the existence of decomposition test :

(1) $T \leftarrow D_{\underline{y}}(\Phi)$

(2) $T \leftarrow U_r \# T$

(3) if T is empty then $\Phi = 1$ is consistent.

(4) end.

<u>Algorithm 5.13</u> : Test for Degeneration of a Circuit : (Theorem 4.3)

Assume that a circuit $\underline{g}$ realizes a given output characteristic function $\Phi(\underline{x}, \underline{y}')$. Let $\underline{g}$ be represented by the array Φ_g (Algorithm 5.1), and the array Φ is either obtained through Algorithm 5.10 or as a result of the decomposition theorem/lemmas application. The following algorithm will now perform a test to determine whether the feedback in $\underline{g}$ is redundant, i.e., whether the sequential circuit is degenerate with respect to Φ .

(1) $T \leftarrow I_q^r(\Phi)$; Insert q x's to equalize dimension of Φ to that of Φ_g.

(2) $T \leftarrow D_{\underline{y}'}(\Phi_g \sqcap T)$; Calculate $\Phi(\underline{x}, \underline{g}(\underline{x}, \underline{y}))$

$$= \sum_{\underline{y}' \in B_2^q} [\, \Phi_g(\underline{x}, \underline{y}, \underline{y}') \cdot \Phi(\underline{x}, \underline{y}') \,]$$

If $\Phi(\underline{x}, \underline{g}(\underline{x}, \underline{y})) = 1$ has an identity solution for $\underline{y}$ then that solution is the state $\underline{y}$ sought in Theorem 4.3.

(3) $T \leftarrow U_{r+q} \# T$; Use Theorem 2.3.

(4) $T \leftarrow U_q \# D_{\underline{x}}(T)$

(5) If T is empty then $\underline{g}$ is non-degenerate else T contains a cover for the possible states $\underline{y} \in B_2^q$.

(6) end

As mentioned in Theorem 4.4, direct transition circuit functions $\underline{g}$ can be generated through the general solution to the output characteristic equation $\Phi(\underline{x}, \underline{y}) = 1$. Therefore, the next two algorithms will generate $\underline{\eta}(\underline{x}, p)$, the first one by Löwenheim's theorem (Section 2.3.3) if an elementary solution is known, the second one then by the successive elimination method (Section 2.3.1) for cases where a trivial solution is not known in advance. In either algorithm, the solution $\underline{\eta}(\underline{x}, \underline{p})$ will be returned in the arrays

$$Z_i \quad , \qquad i = 1, \ldots, q$$

the cubes in Z_i having $r + q$ coordinates $(\underline{x}, \underline{p})$.

<u>Algorithm 5.14</u> : Lowenheim's Theorem.

Let the elementary solution $\underline{\xi}(\underline{x})$ be contained in the arrays

$$Y_i \quad , \qquad i = 1, \ldots, q$$

having r coordinates $(\underline{x})$. Also, let it be assumed that $\Phi(\underline{x}, \underline{y}) = 1$

is consistent.

χ_λ^b of $r + q$ coordinates corresponding to $\underline{x}$ and $\underline{p}$.

(1) for $i = 1$ to q

(2) $Z_i \leftarrow (Y_i \times U_q) \# \Phi$

(3) $Z_i \leftarrow Z_i \sqcup (\Phi \sqcap \chi^1_{r+i})$

(4) next i

(5) end

Algorithm 5.15 : Successive Eliminations.

(1) $T \leftarrow \Phi \times U_q$

(2) for $i = 1$ to q

(3) if $i = 1$ go to 6

(4) $T \leftarrow T \sqcap (C^{111}_{r+i-1}(Z_{i-1}) \sqcup C^{000}_{r+i-1}(U_{r+2q} \# Z_{i-1}))$

(5) $T \leftarrow C^{xxx}_{r+i-1}(T)$

(6) $Z_i \leftarrow (C^{xxx}_{r+i \ldots r+q}(T \sqcap \chi^1_{r+i})) \sqcap \chi^1_{r+q+i}$

(7) $Z_i \leftarrow Z_i \sqcup (U_{r+2q} \# (C^{xxx}_{r+i \ldots r+q}(T \sqcap \chi^0_{r+i})))$

(8) next i ; $r+i \ldots r+q$ is an implied DO loop

(9) for $i = 1$ to q

(10) $Z_i \leftarrow D_{r+i \ldots r+q}(Z_i)$

(11) next i

(12) end

Clearly, Algorithm 5.14 is preferable to Algorithm 5.15, the latter algorithm being incomparably more complicated. However, an elementary solution must be known beforehand if Algorithm 5.14 is to be used.

Algorithm 5.16 : Direct Transition Circuits (Theorem 4.4).

Assuming now, that $\underline{n}(\underline{x}, \underline{p})$ is known, the following algorithm will produce the direct transition circuit function

$$\underline{y}' = \underline{g}(\underline{x}, \underline{y})$$

corresponding to a given function $\underline{p}(\underline{x}, \underline{y})$. Let

P_i, $i = 1, \ldots, q$, be the q arrays of $\underline{p}$ (~~r +~~ q coordinates),

G_i, $i = 1, \ldots, q$, the q arrays of $\underline{g}$, and

Z_i, $i = 1, \ldots, q$, the q arrays of $\underline{n}(\underline{x}, \underline{p})$.

χ^b_λ of q coordinates.

(1) $T \leftarrow U_{r+2q}$

(2) for $i = 1$ to q

(3) $T \leftarrow T \sqcap ((P_i \times \chi_i^1) \sqcup ((U_{r+q} \# P_i) \times \chi_i^0))$

(4) next i

(5) for $i = 1$ to q

(6) $G_i \leftarrow D_{r+q+1 \ldots r+2q}(T \sqcap I_r^q(Z_i))$

(7) next i

(8) end

Steps #1 to 4 form the characteristic function $T(\underline{x}, \underline{y}, \underline{p})$ of the equation $\underline{p} = \underline{p}(\underline{x}, \underline{y})$. The function T is then used to filter out the states of $\underline{y}$ in Z_i which do not produce the correct values of $\underline{y}'$.

Any constraints on the circuit can be induced by adding constraint equations on $\underline{p}$ or directly on $\Phi(\underline{x}, \underline{y}) = 1$, as in [5].

If the function $\underline{p}(\underline{x}, \underline{y})$ is independent of $\underline{y}$ (columns $\underline{y}$ in P_i are all x's) then the resulting circuit $\underline{g}$ is purely combinational, with no feedback loops. However, even certain choices of $\underline{p}$ dependent on $\underline{y}$ could produce circuits without closed loops, e.g., iterative arrays of cells.

5.4 Examples

A number of examples will be presented here, illustrating the theoretical results obtained in Chapters 3 and 4. They will be solved with the aid of the techniques developed in the previous sections of this chapter. No attention will be paid to hazards in the circuits, these can be treated in the usual way when actually implementing a particular circuit - the theory of decomposition is independent of the internal structure of the circuits, as long as the characteristic functions properly reflect the behaviour of the circuit with respect to the steady output states.

Example 5.1 :

Let an asynchronous sequential circuit be given by the following encoded state transition table.

P.S. $z_2\ z_1$	N.S. $z_2'z_1'$; $x_2\ x_1$ = 0 0	0 1	1 0	1 1
0 0	0 0	0 1	1 0	0 0
0 1	0 0	0 1	0 1	0 0
1 0	0 0	1 0	1 0	1 1
1 1	0 0	0 1	1 1	1 1

$|\underline{x}| = 2, \quad X = B_2^2$

$|\underline{z}| = 2, \quad Z = B_2^2$

The excitation equations :

$$z_1' = g_1(\underline{x}, \underline{z}) = \bar{x}_2 x_1(\bar{z}_2 + z_1) + x_2 \bar{x}_1 z_1 + x_2 x_1 z_2$$

$$z_2' = g_2(\underline{x}, \underline{z}) = \bar{x}_2 x_1 z_2 \bar{z}_1 + x_2 \bar{x}_1(z_2 + \bar{z}_1) + x_2 x_1 z_2$$

The next state characteristic function $\Phi_g(\underline{x}, \underline{z}, \underline{z}')$ in the form of a map : (Definition 3.2)

$z_2' z_1'$ \ $z_2 z_1 x_2 x_1$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1			1	1			1	1				1			
1		1				1	1							1		
2			1							1	1					
3												1			1	1

Map #1 : Φ_g

$z_2' z_1'$ \ $x_2 x_1$	0	1	2	3
0	1			1
1		1	1	
2		1	1	
3			1	1

Map #2 : Φ_g^z

(Definition 3.4)

$z_2' z_1'$ \ $x_2 x_1$	0	1	2	3
0	1			1
1		1	1	
2		1	1	
3			1	1

Map #3 : Φ_g^s

(Definition 3.3)

$\Phi_g^s = 1$ is consistent $\rightarrow$ no simple oscillations, actually, the circuit is a normal (stable, 0-transition) circuit, since $\Phi_g^z = \Phi_g^s$. Also, since $\Phi_g^s = 1$ does not have a unique solution, the circuit does not have combinational behaviour (Theorem 3.2).

However, if, for instance, the output generating function

$$y = f(\underline{x}, \underline{z}') = x_2 + x_1(z_2' + z_1')$$

is connected to the circuit $\underline{g}$ (as in Figure 3.1) then :

The circuit characteristic function $\Phi_f(\underline{x}, \underline{z}', \underline{y})$ is (Definition 3.14) :

$z_2'\ z_1'\ x_2\ x_1$

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
y	0	1	1			1				1				1			
	1			1	1		1	1	1		1	1	1		1	1	1

Map #4 : Φ_f

The overall circuit characteristic function is thus

$$\Phi_c(\underline{x}, \underline{y}) = \sum_{\underline{Y} \in B_2^2} \left(\Phi_f(\underline{x}, \underline{Y}, \underline{y}) \cdot \Phi_g^s(\underline{x}, \underline{Y}) \right)$$

$x_2\ x_1$

		0	1	2	3
y	0	1			
	1		1	1	1

Map #5 : $\Phi_c(\underline{x}, \underline{y})$

Since $\Phi_c = 1$ has a unique solution, then by Theorem 3.3 the entire circuit has combinational behaviour. The function generated is thus

$$\underline{y = x_1 + x_2}$$

It is obvious that from a practical point of view it is rather difficult to imagine that such a complicated circuit would ever be used to generate this function. However, it was meant to serve only as an illustration of the method.

Example 5.2 :

To illustrate the decomposition theorem, let the same function $y = x_1 + x_2$ as in Example 5.1 be considered.

The output characteristic function of $y(\underline{x})$ is

$$\Phi(\underline{x}, \underline{y}) = y(x_1 + x_2) + \bar{y}\,\bar{x}_1\,\bar{x}_2$$

the map is identical to the Map #5 before. Clearly, the realization theorem (Theorem 4.1) holds if y is realized by the circuit of Example 5.1, there

$$\Phi(\underline{x}, y) = \Phi_c(\underline{x}, y),$$

so that $\Phi_c \leqslant \Phi$ holds. Now, given the circuit $\underline{f}$ of Example 5.1, let the Lemma 4.2 be applied to generate the output characteristic function for the circuit C_1 if C_2 is $\underline{f}$ in the 2 stage realization of $y(\underline{x})$.

$$\Phi_{c2} = \Phi_f \quad \text{(Map \#4)}$$

Hence,

$$\Phi_1(\underline{x}, \underline{z}) = \prod_{\underline{\beta} \in B_2^2} (\overline{\Phi_f}(\underline{x}, \underline{z}, \underline{\beta}) + \Phi(\underline{x}, \underline{\beta})),$$

as shown in Map #6. Clearly $\Phi_1 = 1$ is consistent.

		$x_2\, x_1$			
		0	1	2	3
	0	1		1	1
$z_2\, z_1$	1	1	1	1	1
	2	1	1	1	1
	3	1	1	1	1

Map #6 : $\Phi_1(\underline{x}, \underline{z})$

If the circuit $\underline{g}$ from Example 5.1 is taken as C_1 then

$$\Phi_{C1} = \Phi_g^s = \Phi_g^z$$

(Map #2 and 3), but $\Phi_{C1} \leq \Phi_1$, so that C_1 completes the realization of y, just confirming the known fact. Since C_1 is a normal circuit then it falls into the class of direct transition circuits realizing Φ_1. Therefore, by Theorem 4.3, any choice of $\underline{z} = \underline{y} \in B_2^2$ will degenerate the sequential circuit into a purely combinational one, maintaining the realization of Φ_1, however.

For instance the choice $z_1 = 1$, $z_2 = 0$, will generate

$$\underline{g}^*(\underline{x}) = \underline{g}(\underline{x}, \underline{y}) : \quad g_1^*(\underline{x}) = \bar{x}_2 x_1 + x_2 \bar{x}_1$$

$$g_2^*(\underline{x}) = 0 ,$$

which can be obtained as an elementary solution of $\Phi_1 = 1$ - the particular solution is encircled on the map of Φ_1 (Map #6). As a matter of fact, any elementary solution of $\Phi_1 = 1$ represents a purely combinational circuit completing the realization of $y(\underline{x})$. (There are 192 different solutions). For instance, there are 3 identity solutions (rows with all 1's) :

$$z_2 = 0, \quad z_1 = 1 \quad \text{or}$$

$$z_2 = 1, \quad z_1 = 0 \quad \text{or}$$

$$z_2 = 1, \quad z_1 = 1 .$$

In order to obtain all direct transition circuits C_1 in "one package", Theorem 4.4 will be applied now :

Let $\underline{\xi}(\underline{x})$: $z_1 = 0$, $z_2 = 1$ be an elementary solution, then Lowenheim's general solution to $\Phi_1 = 1$ is

$$\underline{n}(\underline{x}, \underline{p}) = \Phi_1(\underline{x}, \underline{p})\, \underline{p} + \bar{\Phi}_1(\underline{x}, \underline{p}) \cdot \underline{\xi}(\underline{x})$$

i.e.,

$$n_1(\underline{x}, \underline{p}) = p_1$$

$$n_2(\underline{x}, \underline{p}) = p_2 + \bar{p}_1 x_1 \bar{x}_2 .$$

Any choice of $\underline{p}(\underline{x}, \underline{z}) : B_2^4 \rightarrow B_2^2$ will produce a direct transition circuit.

For example,

$$p_1 = \bar{x}_2 x_1 + x_2 \bar{x}_1 , \quad p_2 = 0$$ generates $\underline{g}^*(\underline{x})$ derived before.

$$\left.\begin{aligned} p_1 &= g_1(\underline{x}, \underline{z}) \\ p_2 &= g_2(\underline{x}, \underline{z}) \end{aligned}\right\} \quad \text{of Example 5.1 will make}$$

$$\underline{n}(\underline{x}, \underline{p}(\underline{x}, \underline{z})) = \underline{g}(\underline{x}, \underline{z})$$

Another circuit may be generated by $p_1 = x_1 \bar{z}_2$, $p_2 = x_2 \bar{z}_1$. Then

$$g_1 = x_1 \bar{z}_2$$

$$g_2 = x_2 \bar{z}_1 + x_1 \bar{x}_2 z_2$$

whose characteristic function $\Phi_g(\underline{x}, \underline{z}, \underline{z}')$ is shown in Map #7.

$z_2 \, z_1 \, x_2 \, x_1$

$z_2' \, z_1'$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1				1		1		1				1		1	1
1		1				1		1								
2			1							1	1	1		1		
3				1												

$p_1 = x_1 \bar{z}_2$, $\quad p_2 = x_2 \bar{z}_1$

Map #7 : Φ_g

And

$x_2 \, x_1$

$z_2' \, z_1'$	0	1	2	3
0	1		1	1
1		1		1
2		1		1
3			1	1

Map #8 : Φ_g^z

$x_2 \, x_1$

$z_2' \, z_1'$	0	1	2	3
0	1			
1		1		1
2		1	1	1
3				

Map #9 : Φ_g^s

The resulting direct transition circuit is not normal, the total state $x_1\, x_2\, z_1\, z_2 = 0\ 1\ 0\ 0$ is transient, also, the circuit has oscillations for $x_1\, x_2 = 1\ 1$ with the steady states being $z_1\, z_2 = 0\ 0$ and $1\ 1$. All other states are stable. Note, that the oscillations are not simple.

The combined steady state characteristic function Φ_g^c is shown in Map #10.

$z'\ z'$ \ $x_2\ x_1$	0	1	2	3
0	1			1
1		1		1
2		1	1	1
3				1

$p_1 = x_1\, \bar{z}_2$,

$p_2 = x_2\, \bar{z}_1$

Map #10 : Φ_g^c

Again, $\Phi_g^c \leqslant \Phi_1 \rightarrow$ the oscillatory circuit realizes $y(\underline{x})$.

The choice of $p_1 = x_1\, x_2\, \bar{z}_2$, $p_2 = x_1\, x_2\, \bar{z}_1$, leads to

$$g_1(\underline{x}, \underline{z}) = x_1\, x_2\, \bar{z}_2$$

$$g_2(\underline{x}, \underline{z}) = x_1\, x_2\, \bar{z}_1 + x_1\, \bar{x}_2$$

which is a quasi-normal circuit with simple oscillations at $x_1\, x_2 = 1\ 1$, everywhere else the circuit is stable. The steady state characteristic function Φ_g^c (see Algorithm 5.5) is shown in Map #11.

$x_2\ x_1$

$z'_2\ z'_1$	0	1	2	3
0	1		1	1
1				1
2	1	1		1
3				1

$p_1 = x_1\, x_2\, \bar{z}_2$

$p_2 = x_1\, x_2\, \bar{z}_1$

Map #11 : Φ_g^c

Consider now the case where the circuit C_1 is given with its characteristic function Φ_C and the circuit C_2 is sought. The output characteristic function $\Phi_2(\underline{x}, \underline{z}, \underline{y})$ of C_2 is then (Lemma 4.3):

$$\Phi_2(\underline{x}, \underline{z}, \underline{y}) = \overline{\Phi}_{C1}(\underline{x}, \underline{z}) + \Phi(\underline{x}, \underline{y})$$

For instance, let

Φ : Map #5,

Φ_{C1} : Map #10.

The function Φ_2 is shown in Map #12.

$z_2\ z_1\ x_2\ x_1$

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1		1		1		1				1	1	1	
1			1	1	1	1	1	1	1	1	1	1	1	1	1	1

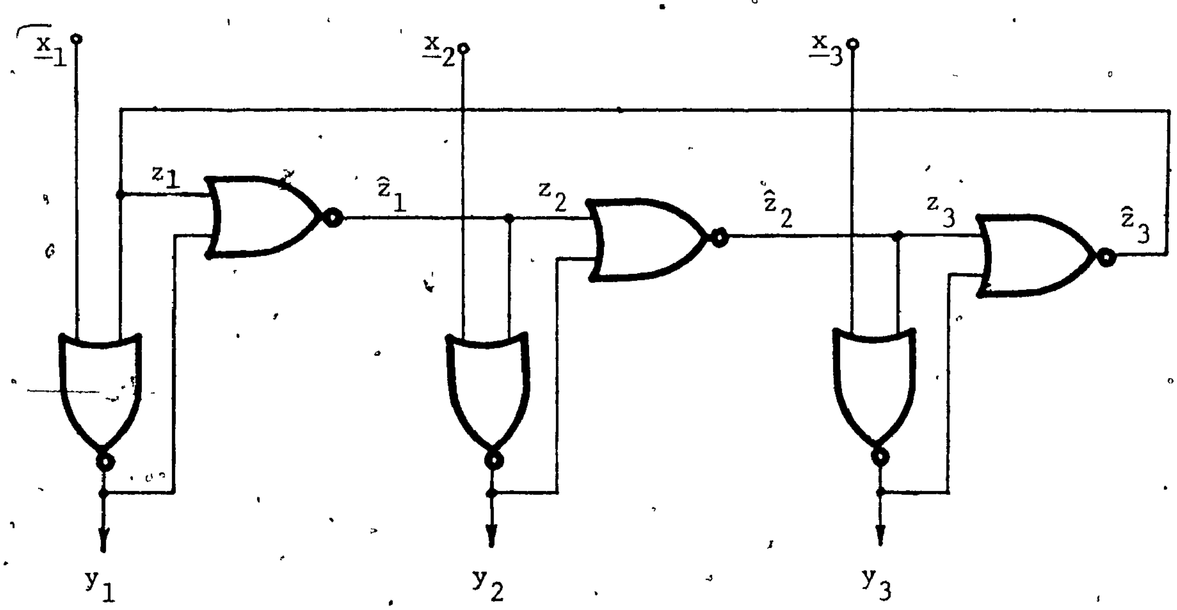
Map #12 : Φ_2

Clearly, the former function $\underline{f}$ satisfies the realization theorem - $\Phi_f \leq \Phi_2$. Note also, that $\Phi_2 = 1$ is always consistent, no matter what C1 is.

Any direct transition circuit C_2 can be obtained through the same procedure as when C_1 was sought before (Theorem 4,4). Or for any circuit C_2 selected from a library of modules, the realization of Φ_2 can be tested by seeing that the relation $\Phi_{C2} \leqslant \Phi_2$ is satisfied.

Example 5.3 :

Having shown an application of the theory on simple examples, a more complicated case will be investigated using cubical arrays - the unilateral cellular array with a closed loop as described in [14]. The particular circuit built from NOR gates is shown in Figure 5.1.



$y_1 = \bar{x}_1 . \bar{z}_1$ $\quad y_2 = \bar{x}_2 \, \bar{z}_2$ $\quad \hat{z}_i = \bar{z}_i \, x_i$

$y_3 = \bar{x}_3 . \bar{z}_3$

FIGURE 5.1 : CIRCUIT FOR EXAMPLE 5.3.

The interconnection of the cells is such that

$$z_i = \hat{z}_{i-1}$$

thus the next state equations of its $\underline{g}$ circuit are

$$z_1' = \bar{z}_3 x_1$$

$$z_2' = \bar{z}_1 x_2$$

$$z_3' = \bar{z}_2 x_3$$

And the output generating circuit function $\underline{f}$:

$$y_1 = \bar{z}_3 \bar{x}_1$$

$$y_2 = \bar{z}_1 \bar{x}_2$$

$$y_3 = \bar{z}_2 \bar{x}_3$$

Formation of the characteristic functions :

Φ_g (Algorithm 5.1) :

		$\underline{x}$	$\underline{z}$	$\underline{z}'$
(1)	$\Phi_g =$	x x x	x x x	x x x
(2)	i = 1			
(3)	$\Phi_g =$	1 x x	x x 0	1 x x
		x x x	x x 1	0 x x
		0 x x	x x x	0 x x
(2)	i = 2			

(3) Φ_g =

1 1 x	0 x 0	1 1 x
x 1 x	0 x 1	0 1 x
0 1 x	0 x x	0 1 x
1 x x	1 x 0	1 0 x
x x x	1 x 1	0 0 x
0 x x	1 x x	0 0 x
1 0 x	x x 0	1 0 x
x 0 x	x x 1	0 0 x
0 0 x	x x x	0 0 x

(2) i = 3

(3) Φ_g =

x	z	z'
1 1 1	0 0 0	1 1 1
x 1 x	0 0 1	0 1 1
0 1 1	0 0 x	0 1 1
1 x 1	1 0 0	1 0 1
x x 1	1 0 1	0 0 1
0 x 1	1 0 x	0 0 1
1 0 1	x 0 0	1 0 1
x 0 1	x 0 1	0 0 1
0 0 1	x 0 x	0 0 1
1 1 x	0 1 0	1 1 0
x 1 x	0 1 1	0 1 0
0 1 x	0 1 x	0 1 0
1 x x	1 1 0	1 0 0
x x x	1 1 1	0 0 0
0 x x	1 1 x	0 0 0
1 0 x	x 1 0	1 0 0
x 0 x	x 1 1	0 0 0
1 1 0	0 x 0	1 1 0
x 1 0	0 x 1	0 1 0
0 1 0	0 x x	0 1 0
x x 0	1 x 1	0 0 0
0 x 0	1 x x	0 0 0
1 x 0	x x 0	1 0 0
x 0 0	x x 1	0 0 0
0 0 0	x x x	0 0 0

(5) D = $\emptyset$

(6) end; Φ_g as above.

Φ_g^z (Algorithm 5.3) :

(1) $\Phi_g^z \leftarrow A(D_{\underline{z}}(\Phi_g))$

	$\underline{x}$	$\underline{z'}$
=	1 1 1	1 1 1
	x 1 x	0 1 1
	1 x 1	1 0 1
	x x 1	0 0 1
	1 1 x	1 1 0
	x 1 x	0 1 0
	1 x x	1 0 0
	x x x	0 0 0

(2) end

Φ_g^s (Algorithm 5.2) :

		$\underline{x}$	$\underline{z'}$
(1)	Φ_g^s =	x x x	x x x

(2) i = 1

(3)	Φ_g^s =	0 x x	0 x x
		1 x x	1 x 0
		1 x x	0 x 1
		x x x	0 x 1

(2) i = 2

(3) Φ_g^s =
$$\begin{matrix} 0 & 0 & x & 0 & 0 & x \\ x & 0 & x & 0 & 0 & 1 \\ 1 & x & x & 1 & 0 & 0 \\ 0 & 1 & x & 0 & 1 & x \\ x & 1 & x & 0 & 1 & 1 \end{matrix}$$

(2) $i = 2$

(3) Φ_g^s =
$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & x & 0 & 1 & 0 & 0 \\ 0 & 1 & x & 0 & 1 & 0 \\ \underline{x} & \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{1} \end{matrix}$$

(5) $D = \emptyset$

(6) end

Simple oscillations test on $\underline{g}$ (Algorithm 5.4) :

(1) $T = D_{\underline{z}}(\Phi_g^s) = \begin{matrix} 0 & 0 & 0 \\ 1 & x & 0 \\ 0 & 1 & x \\ x & 0 & 1 \end{matrix}$

(2) $T = U_r \# T = \{x\ x\ x\} \# \begin{Bmatrix} 0 & 0 & 0 \\ 1 & x & 0 \\ 0 & 1 & x \\ x & 0 & 1 \end{Bmatrix}$

$$= \underline{1\ 1\ 1}$$

(3) $D_x = \emptyset$

(4) $T \neq \emptyset$ = simple oscillations at $x_1 = x_2 = x_3 = 1$, which just confirms a known fact about the circuit.

(5) end.

Except for the state $\underline{x} = 1\ 1\ 1$, the circuit g is stable as can be checked through Φ_g. Therefore, an approximation to Φ_g^c - the steady state characteristic function - can be generated by Algorithm 5.5.

$\underline{\Phi_g^c \text{ (Algorithm 5.5)}}$:

$$\Phi_g^c = \Phi_g^s \cup (\Phi_g^z \sqcap (T \times U_3))$$

$$= \left\{\begin{matrix} 0\,0\,0 & 0\,0\,0 \\ 1\,x\,0 & 1\,0\,0 \\ 0\,1\,x & 0\,1\,0 \\ x\,0\,1 & 0\,0\,1 \end{matrix}\right\} \cup \left\{\begin{matrix} 1\,1\,1 & 1\,1\,1 \\ 1\,1\,1 & 0\,1\,1 \\ 1\,1\,1 & 1\,0\,1 \\ 1\,1\,1 & 0\,0\,1 \\ 1\,1\,1 & 1\,1\,0 \\ 1\,1\,1 & 0\,1\,0 \\ 1\,1\,1 & 1\,0\,0 \\ 1\,1\,1 & 0\,0\,0 \end{matrix}\right\}$$

$$= \begin{matrix} 0\,0\,0 & 0\,0\,0 \\ 1\,x\,0 & 1\,0\,0 \\ 0\,1\,x & 0\,1\,0 \\ x\,0\,1 & 0\,0\,1 \\ 1\,1\,1 & x\,x\,x \end{matrix}$$

(Note, that the array was reduced by an application of a standard minimization procedure for single output functions).

Simply by inspection of Φ_g^c, it can be seen that Theorem 3.1 fails - multiple steady states at $\underline{x} = 1\ 1\ 1$. Therefore, before a test for combinational behaviour (Theorem 3.3) can be made, the characteristic function Φ_f has to be formed.

Φ_f (Algorithm 5.7) :

$F_1 = \{0\ x\ x \quad x\ x\ 0\}$

$F_2 = \{x\ 0\ x \quad 0\ x\ x\}$

$F_3 = \{x\ x\ 0 \quad x\ 0\ x\}$

		x	z'	y
(1)	Φ_f =	x x x	x x x	x x x
(2)	i = 1			
(3)	Φ_f =	0 x x	x x 0	1 x x
		x x x	x x 1	0 x x
		1 x x	x x x	0 x x
(2)	i = 2			
(3)	Φ_f =	0 0 x	0 x 0	1 1 x
		x 0 x	0 x 1	0 1 x
		1 0 x	0 x x	0 1 x
		0 x x	1 x 0	1 0 x
		x x x	1 x 1	0 0 x
		1 x x	1 x x	0 0 x
		0 1 x	x x 0	1 0 x
		x 1 x	x x 1	0 0 x
		1 1 x	x x x	0 0 x

(2) i = 3

(3) Φ_f =

X	Z'	Y
0 0 0	0 0 0	1 1 1
x 0 0	0 0 1	0 1 1
1 0 0	0 0 x	0 1 1
0 x 0	1 0 0	1 0 1
x x 0	1 0 1	0 0 1
1 x 0	1 0 x	0 0 1
0 1 0	x 0 0	1 0 1
x 1 0	x 0 1	0 0 1
1 1 0	x 0 x	0 0 1
0 0 x	0 1 0	1 1 0
x 0 x	0 1 1	0 1 0
1 0 x	0 1 x	0 1 0
0 x x	1 1 0	1 0 0
x x x	1 1 1	0 0 0
1 x x	1 1 x	0 0 0
0 1 x	x 1 0	1 0 0
x 1 x	x 1 1	0 0 0
1 1 x	x 1 x	0 0 0
0 0 1	0 x 0	1 1 0
x 0 1	0 x 1	0 1 0
1 0 1	0 x x	0 1 0
0 x 1	1 x 0	1 0 0
x x 1	1 x 1	0 0 0
1 x 1	1 x x	0 0 0
0 1 1	x x 0	1 0 0
x 1 1	x x 1	0 0 0
1 1 1	x x x	0 0 0

Combinational behaviour of the entire circuit (Theorem 3.3) (Algorithm 5.8):

(1) $\Phi_c \leftarrow \Phi_g^c \times U_3$

	$\underline{x}$	$\underline{z}'$	$\underline{y}$
=	0 0 0	0 0 0	x x x
	1 x 0	1 0 0	x x x
	0 1 x	0 1 0	x x x
	x 0 1	0 0 1	x x x
	1 1 1	x x x	x x x

(2) $\Phi_c \leftarrow \Phi_c \sqcap \Phi_f$

=	0 0 0	0 0 0	1 1 1
	1 x 0	1 0 0	0 0 1
	0 1 x	0 1 0	1 0 0
	x 0 1	0 0 1	0 1 0
	1 1 1	x x x	0 0 0

(3) $\Phi_c \leftarrow D_{\underline{z}'}(\Phi_c)$

	$\underline{x}$	$\underline{y}$
=	0 0 0	1 1 1
	1 x 0	0 0 1
	0 1 x	1 0 0
	x 0 1	0 1 0
	1 1 1	0 0 0

(5) T = $\emptyset$

(6) i = 1 (6) i = 2 (6) i = 3

(7) T = $\emptyset$ (7) T = $\emptyset$ (7) T = $\emptyset$

(9) T = $\emptyset$

(10) T empty $\rightarrow$ the circuit has combinational behaviour, the

characteristic function is in Φ_c at step #4.

(11) end.

Namely,

$$\begin{cases} y_1 = \bar{x}_1(x_2 + \bar{x}_3) \\ y_2 = \bar{x}_2(x_3 + \bar{x}_1) \\ y_3 = \bar{x}_3(x_1 + \bar{x}_2) \end{cases}$$

which is precisely the function as given in [14].

To show that the feedback in the circuit $\underline{g}$ is not redundant, i.e., $\underline{g}$ is not degenerate with respect to the corresponding output characteristic function $\Phi_1(\underline{x}, \underline{z}')$, let Theorem 4.3 be applied through the Algorithm 5.13. First, however, $\Phi_1(\underline{x}, \underline{z}')$ has to be formed as (Lemma 4.2) :

$$\Phi_1(\underline{x}, \underline{z}') = \prod_{\underline{\beta} \in B_2^3} (\overline{\Phi_f}(\underline{x}, \underline{z}', \underline{\beta}) + \Phi(\underline{x}, \underline{\beta})),$$

where $\Phi(\underline{x}, \underline{y})$ is the overall output characteristic function of $\underline{y}(\underline{x})$ (Theorem 4.1), i.e., in this case it is the array Φ_c as generated before. Hence, (Algorithm 5.11b),

(1) $\Phi_1 \leftarrow \Phi_f \# I_3^3(\Phi)$; $\Phi_f \cdot \bar{\Phi}$

(2) $\Phi_1 \leftarrow U_6 \# D_{\underline{y}}(\Phi_1)$; $\overline{\sum_{\underline{\beta} \in B_2^3}} [\Phi_f(\underline{x}, \underline{z}', \underline{\beta}) \cdot \overline{\Phi}(\underline{x}, \underline{\beta})]$

(3) end.

After substituting the actual arrays :

$$
\Phi_1 = \begin{array}{ccc c ccc}
\multicolumn{3}{c}{\underline{x}} & & \multicolumn{3}{c}{\underline{z}'} \\
1 & 1 & 1 & & x & x & x \\
1 & 1 & x & & x & 0 & x \\
1 & x & 1 & & 0 & x & x \\
x & 1 & 1 & & x & x & 0 \\
x & 0 & 1 & & 0 & x & 1 \\
1 & x & 0 & & 1 & 0 & x \\
1 & 0 & 1 & & 0 & 1 & x \\
0 & 1 & x & & x & 1 & 0 \\
0 & 0 & 0 & & 0 & 0 & 0
\end{array}
$$

Now Algorithm 5.13 :

(1) $T \leftarrow I_3^3(\Phi_1)$

(2) $T \leftarrow D_{\underline{z}'}(\Phi_g \sqcap T)$; Φ_g as generated previously

$$
T = \begin{array}{ccc c ccc}
\multicolumn{3}{c}{\underline{x}} & & \multicolumn{3}{c}{\underline{z}} \\
1 & 1 & 1 & & 0 & 0 & 0 \\
1 & 1 & 1 & & 1 & 0 & 0 \\
1 & 1 & x & & 1 & 1 & x \\
1 & 1 & 0 & & 1 & x & 1 \\
1 & 1 & 1 & & 0 & 0 & 1 \\
1 & x & 1 & & 1 & 0 & 1 \\
1 & 0 & 1 & & x & 0 & 1 \\
1 & 1 & 1 & & 0 & 1 & 1 \\
1 & x & 1 & & 1 & 1 & 1 \\
1 & 0 & 1 & & x & 1 & 1 \\
1 & 1 & 1 & & 0 & 1 & 0 \\
x & 1 & 1 & & 0 & 1 & 1 \\
0 & 1 & 1 & & 0 & 1 & x
\end{array}
$$

x 1 1	1 1 1
0 1 1	1 1 x
0 0 1	1 0 x
x 0 1	x 0 1
0 0 1	x 0 x
1 x 0	x x 0
0 1 x	0 1 1
0 1 0	0 x x
0 0 0	x x x

(3) $T \leftarrow U_{3+3} \# T$

$T =$

0 0 1	x 1 0
0 1 0	1 x x
0 1 1	x 0 1
0 1 x	1 0 1
0 1 1	0 0 0
1 0 0	x x 1
1 0 1	x x 0
x 0 1	x 1 0
1 x 0	0 x 1
1 0 0	0 0 1

(4) $T \leftarrow U_3 \# D_{\underline{x}}(T)$

$T = \emptyset$

(5) Since T is empty then $\underline{g}$ is non-degenerate with respect to Φ_1.

Therefore, the feedback loop is not redundant, and it cannot be directly replaced by a combinational circuit $\underline{g}^*(\underline{x}) = \underline{g}(\underline{x}, \underline{y})$. Actually, there is no purely combinational circuit which would realize $\Phi(\underline{x}, \underline{y})$ with less NOR gates than $\underline{g}(\underline{x}, \underline{z})$ [14]. Furthermore, **since $\underline{g}$ is non-degenerate the circuit may have hazardous transitions** between steady states.

Any direct transition circuit $\underline{g}$ realizing Φ_1 (with or without feedback) can be generated by applying Algorithm 5.13 (or 5.14), and then Algorithm 5.15 for a selected function $\underline{p}(\underline{x}, \underline{z})$.

CHAPTER 6

APPLICATIONS

The theoretical results and computational techniques of the preceeding chapters established a firm link between Boolean equations and the internal structure of combinational switching circuits. The link is expressed by the $\leqslant$ relation between the characteristic functions of the corresponding Boolean equations. This then provides the methodological base for solving practical problems related to the design of switching circuits. Two such applications will be discussed here, namely, an approach to the algorithmic synthesis of combinational switching functions by decomposition, and a method for generating test sets of input stimuli for detecting faults in combinational circuits.

6.1 Application #1 : Modular Synthesis of Combinational Circuits

6.1.1 Problem Statement

Given an incompletely specified multiple output combinational function $\underline{y}(\underline{x}) = \langle \underline{\phi}(\underline{x}), \underline{\phi}(\underline{x}) + \underline{d}(\underline{x}) \rangle$, $|\underline{y}| = q$, $|\underline{x}| = r$, and a set S of combinational and sequential circuit modules M_i, $i = 1, \ldots, |S|$, it is required to synthesize a circuit C using only the modules in S, such that C would realize $\underline{y}(\underline{x})$. Furthermore, it may be required that the resulting network should satisfy a number of prespecified constraints such as total cost, signal propagation delay, loading of input lines, number of external interconnections, etc.

As mentioned in a number of previous studies [2, 5, 7, 13, 15, 16, 20, 21, 24, 26, 28, 29] determination of an absolutely optimal circuit (under any criteria) implies more or less exhaustive searching over all possible realizations of $\underline{y}$, using modules in S. Therefore, to reduce the number of trials, and yet to obtain a reasonably good (satisfies constraints) realization, some heuristic searching techniques must be applied [5, 15, 28]. It is not the intention, however, to develop these search algorithms here. The reason is that these heuristic algorithms provide a strategy for judging the goodness of a particular partial decomposition and for choosing the next step to take towards obtaining a final satisfactory circuit. Thus up to certain extent, they can be considered separately from the method used in generating a partial decomposition. Rather, it will be shown how the methodology of characteristic functions as developed here could be applied to unify the steps common to most algorithms based on functional decomposition (Section 4.1).

The decomposition related problems may be summarized as follows:

(1) Formation of a library of available circuit modules.

(2) Representation of the function $\underline{y}(\underline{x})$.

(3) Selection of a subset of the input (output) variables of $\underline{y}(\underline{x})$, and the mapping between these and the module variables.

(4) Application of the module under the selected mapping to the function $\underline{y}(\underline{x})$ in order to test whether a decomposition exists.

(5) Testing of "goodness" of the decomposition, that is, realization of some of the outputs $\underline{y}$ or the module inputs, determination of redundant variables, satisfaction of circuit constraints, etc. The results of these tests are then used to guide the heuristic ~~search~~ for an optimal realization.

6.1.2. Development of Solution Steps

(1) The library: Let the available module and submodule functions be represented by their circuit characteristic functions (Definition 4.1). A submodule is obtained from a module in S by tying some of its inputs together or bringing them to

constant levels 0, 1. If a particular module is combinational then the various submodule characteristic functions can be obtained by imposing constraint equations of the before mentioned type on the module characteristic function (equation). In case of sequential modules, the constraint equations should be applied to the state transition characteristic function (Definition 3.2), and then the corresponding circuit characteristic function obtained. Otherwise the submodule characteristic function might describe steady states never reachable by any input sequence.

In either case, however, the library L can be viewed as a list of circuit characteristic functions Φ_{Ci}, $i = 1, \ldots, |L|$, each entry containing additional information describing the number of inputs and outputs, loading factors, delay, symmetry information, etc.

If the Library were to contain only the characteristic functions of the original modules, then the submodules would have to be generated during the synthesis procedure. However, it was shown [5, 28] that the time required to do so is rather long, and that the procedure must be repeated with each new function synthesized. Therefore, it will be assumed that the preprocessing is done while forming the library, thus only one to one mappings need to be considered between the function $\underline{y}(\underline{x})$ and the module variables [7, 28].

(2) Form of $\underline{y}(\underline{x})$: Whatever the initial description of $\underline{y}(\underline{x})$ is, it should be converted into the form of an output characteristic function Φ (as shown in Theorem 4.1). Additional information with regards to the maximum loading, delay, cost, etc, should also be supplied to define the properties of the target circuit C.

(3)
(4)
(5) As mentioned in Section 4.1, two main paths were considered in the past when performing synthesis by decomposition. Either a mapping μ_x from a subset of the $\underline{x}$ variables to the module inputs was chosen, the resulting function generated by the circuit was applied to $\underline{y}(\underline{x})$ to form a decomposition, and then a test for realization of some of the outputs $\underline{y}$ was made [2, 7, 15, 21, 24, 28]. The other way was to choose a mapping μ_y from module outputs to a subset of problem outputs $\underline{y}$, and then seek a decomposition via systems of B.E., with the highest number of inputs to the module being realized directly by some of the variables $\underline{x}$ [5, 16, 20].

Both of the methods had a disadvantage that with each new mapping the decomposition procedure had to be repeated. Moreover, the position of a module inside of the final circuit was a priori determined to have either all the inputs fed by $\underline{x}$ directly or the outputs tied directly to $\underline{y}$. The Lemmas 4.2 - 4.4 and Theorem 4.2 unify the two opposing approaches under one methodology, and allow for the module to be placed anywhere inside of the future circuit. Let it be assumed that a (sub)module characteristic function $\Phi_C(\underline{v}, \underline{w})$, $|\underline{v}| = m$, $|\underline{w}| = n$ was selected from the library to form a partial realization of an output characteristic function $\Phi(\underline{x}, \underline{y})$, $|\underline{x}| = r$, $|\underline{y}| = q$. Without performing any mapping between the module and problem variables, the module may be left "floating" in the future circuit, and the output characteristic function (Corollary to Theorem 4.2)

$$\Phi_1(\underline{x}, \underline{w}, \underline{v}, \underline{y}) = \overline{\Phi}_C(\underline{v}, \underline{w}) + \Phi(\underline{x}, \underline{y})$$

may be formed, where $\underline{x}$, $\underline{w}$ are new domain and $\underline{v}$, $\underline{y}$ new range variables. Now, a mapping μ_x from <u>some</u> $\underline{x}$ to <u>some</u> $\underline{v}$ and a mapping μ_y from <u>some</u> $\underline{w}$ to <u>some</u> $\underline{y}$ may be selected. (The selection can be guided by similar criteria as in [5, 20]). The mappings may be expressed by the equations

$$v_k = x_l , \qquad w_i = y_j$$

for all the interconnections to be made. Let then

$$\mu_x(\underline{x}, \underline{v}) = 1 \quad \text{and} \quad \mu_y(\underline{w}, \underline{y}) = 1$$

be the corresponding characteristic equations of the two systems of mapping equations. (Note that μ_x and μ_y represent a combinational circuit). The application of the mappings can then be expressed by a new output characteristic function

$$\Phi_2(\underline{x}, \underline{w}, \underline{v}, \underline{y}) = \mu_x(\underline{x}, \underline{v}) \cdot \mu_y(\underline{w}, \underline{y}) \cdot \Phi_1(\underline{x}, \underline{w}, \underline{v}, \underline{y})$$

with the variables from $\underline{v}$ and $\underline{y}$ used in the mappings being deleted. If

$$\Phi_2(\underline{x}, \underline{w}, \underline{v}, \underline{y}) = 1$$

is consistent with respect to solutions $\underline{v}(\underline{x})$ and $\underline{y}(\underline{x}, \underline{w})$ then the particular mappings μ_x, μ_y with the module Φ_C form a partial realization of $\Phi(\underline{x}, \underline{y})$. Thus by applying various mappings μ_x and μ_y and then selecting the best ones (under some heuristic criteria) which form a consistent $\Phi_2 = 1$, the process may be repeated to realize Φ_2, etc.

Note, that if maps are used to represent Boolean functions then the largest map required would have $2^{r+q+m+n}$ bits. If cubical complex representation is used then the largest array will be that of Φ_1. It would require $r + q + m + n$ columns and a maximum of $2^r + 2^m$ rows - a very conservative estimate considering the fact that if Φ_1 is evaluated as $U_{r+m+n+q} \# (\Phi_C \# \Phi)$ then the resulting array is formed from all the prime implicants of Φ_1 (Section 5.3.2) [7]. It also means, that standard reduction (extraction) techniques could be applied to reduce the size of the array even more. If a quaternal encoding of the variables in arrays is used [36], then the upper limit on the bit requirement for storing the array Φ_1 is

$$2 \cdot (r + q + m + n) \cdot (2^r + 2^m) .$$

Hence for a reasonably sized problem of

$$r = 9, \quad q = 10, \quad m = 5, \quad n = 4,$$

Φ_1 can be represented as a 2^{28} bit map or as an array with at most $2 \cdot 28 \cdot 2^{14} < 2^{20}$ bits. Considering the additional advantages of the array representation such as easy manipulation (Section 5.1, 5.2), then it seems that the more suitable form of representing Boolean functions for computer processing is that of cubical complexes.

The sequential application of the various mappings μ_x, μ_y to a given Φ_1 may be a rather time consuming task, even if the sequence of applications is trimmed by using the inconsistency information of Φ_1 under past mappings. It will be shown here, however, that by increasing the memory requirements and by incorporating special (hypothetical) mapping modules, parallel processing of all the possible mappings can be performed. The result of the operations being an encoded list of all mappings which form a consistent equation

$$\Phi_2 = 1 .$$

The proposed structure is shown in Figure 6.1. The modules Ξ and Ω are hypothetical, their function is to perform the mappings μ_x and μ_y, respectively. Their operation is controlled

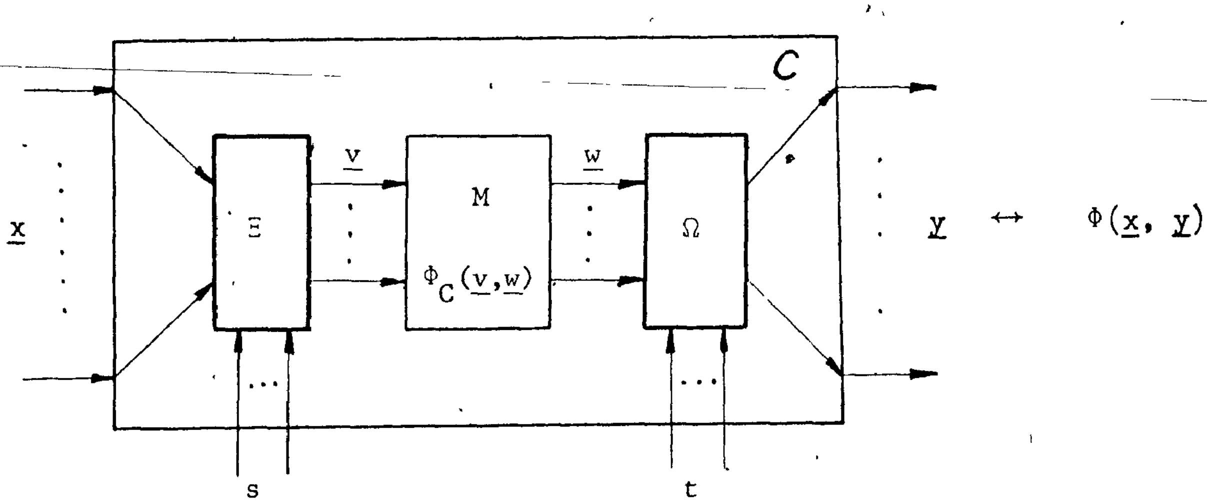


FIGURE 6.1. MAPPING MODULE APPLICATION.

by the parameters $\underline{s}$ and $\underline{t}$ in such a way that a constant value of $\underline{s}$ and $\underline{t}$ selects a particular mapping from the module variables to the problem variables. The internal structure of the modules Ξ and Ω will be derived by first considering a general mapping module from a set $\underline{a} = \{a_1, a_2, \ldots, a_\alpha\}$ to a set $\underline{b} = \{b_1, \ldots, b_\beta\}$ in terms of some parameters $\underline{\nu} = \{\nu_1, \ldots, \nu_\gamma\}$.

The simplest arrangement would be to select $\gamma = \alpha \cdot \beta$, assign

$$\underline{\nu} = \{\nu_1^1, \nu_2^1, \ldots, \nu_\alpha^1, \nu_1^2, \ldots, \nu_\alpha^2, \ldots, \nu_\alpha^\beta\},$$

and then let

$$b_i = a_1 \nu_1^i + a_2 \nu_2^i + \ldots + a_\alpha \nu_\alpha^i, \qquad i = 1, \ldots, \beta \tag{6.1.1}$$

with an orthonormality constraint imposed over all $i = 1, \ldots, \beta$ subsets of $\underline{\nu}$, so as to force only a single variable to be connected to b_i for a given value of $\underline{\nu}$. That is, the parameters must satisfy [3, 4, 5, 27]

$$\sum_{j=1}^{\alpha} \nu_j^i = 1, \quad \nu_j^i \cdot \nu_k^i = 0, \quad j \neq k, \; i = 1, \ldots, \beta.$$

The equations can be combined and represented by a characteristic equation

$$\prod_{i=1}^{\beta} \left[\sum_{j=1}^{\alpha} \left(\nu_j^i \prod_{k \neq j} \bar{\nu}_k^i \right) \right] = 1 \tag{6.1.2}$$

An additional constraint can be obtained by considering one to one mapping only. That is,

$$\nu_k^i \cdot \nu_k^j = 0 \quad \text{for all} \quad i = 1, \ldots, \beta, \quad j \neq i, \quad k = 1, \ldots, \alpha$$

Or in one equation as

$$\prod_{k=1}^{\alpha} \cdot \prod_{i=1}^{\beta} \left[\prod_{j \neq i} (\bar{\nu}_k^i + \bar{\nu}_k^j) \right] = 1 \tag{6.1.3}$$

For (6.1.3) to be applicable $\alpha \geq \beta$ must be satisfied. The module characteristic function Φ_M obtained from (6.1.1) is thus

$$\Phi_M = \prod_{i=1}^{\beta} c_i = \prod_{i=1}^{\beta} \left[b_i(a_1 \nu_1^i + \ldots + a_\alpha \nu_\alpha^i) + \bar{b}_i \overline{(a_1 \nu_1^i + \ldots + a_\alpha \nu_\alpha^i)} \right] \tag{6.1.4}$$

and constrained by (6.1.2), (6.1.3).

The total number of parameters $\alpha \cdot \beta$ is very high even for relatively small problems; however, due to (6.1.2) an economic encoding of the orthonormal sets [26] may reduce the number of necessary parameters by a considerable factor. Two such encodings will be mentioned hereafter.

Encoding 1 :

Consider each b_i separately, then the corresponding subsets of $\underline{\nu}$ may be encoded by β sets of distinct parameters $\underline{p}^i$. The number of parameters needed in each set is $\lceil \log_2(\alpha)$, bringing the total number of parameters to $\beta \cdot \lceil \log_2(\alpha)$. The variables $\underline{\nu}$ thus become functions of $\underline{p}$ with the orthonormality constraint (6.1.2) embedded in them. A simple algorithm for the generation of the orthonormal functions can be found in [5] . An actual assignment will be shown in the following example.

Example : Let $\alpha = 3$, $\beta = 2$, then $\beta \cdot \lceil \log_2(3) = 4$, thus

$\underline{p} = p_1^1, p_2^1, p_1^2, p_2^2$, and

$$\nu_1^1 = p_1^1 \, p_2^1 \qquad\qquad \nu_1^2 = p_1^2 \, p_2^2$$

$$\nu_2^1 = \bar{p}_1^1 \, p_2^1 \qquad\qquad \nu_2^2 = \bar{p}_1^2 \, p_2^2$$

$$\nu_3^1 = \cancel{\bar{p}_1^1 \, \bar{p}_2^1 + p_1^1 \, \bar{p}_2^1} = \bar{p}_2^1 \qquad\qquad \nu_3^2 = \bar{p}_2^2$$

The module characteristic function is (6.1.4)

$$\begin{aligned}\Phi_M(\underline{a}, \underline{b}, \underline{p}) = \prod_{i=1}^{2} [& b_i(a_1 \, p_1^i \, p_2^i + a_2 \, \bar{p}_1^i \, p_2^i + a_3 \, \bar{p}_2^i) \\ & + \bar{b}_i(\bar{a}_1 \, \bar{a}_2 \, \bar{a}_3 + \bar{a}_2 \, \bar{a}_3 \, \bar{p}_1^i + \bar{a}_1 \, \bar{a}_3 \, \bar{p}_1^i + \bar{a}_1 \, \bar{a}_2 \, p_2^i \\ & + \bar{a}_3 \, \bar{p}_2^i + \bar{a}_2 \, \bar{p}_1 \, p_1 + \bar{a}_1 \, p_1 \, p_2)]\end{aligned}$$

The constraint (6.1.3) will take the form

$$(\bar{p}_1^1 + \bar{p}_2^1 + \bar{p}_1^2 + \bar{p}_1^2)\cdot(p_1^1 + \bar{p}_2^1 + p_1^2 + \bar{p}_2^2)\,(p_2^1 + p_2^2) = 1$$

Any additional constraints could be expressed in terms of the parameters in a similar way.

Encoding 2 :

If $\alpha \geq \beta$ is assumed then even higher reduction of the number of parameters required may be achieved by incorporating the constraint (6.1.3) into (6.1.2). The two constraints describe the total of $\frac{\alpha!}{(\alpha - \beta)!}$ distinct states of $\underline{\nu}$; therefore, there are

$$|\underline{p}| = \lceil \log_2\left(\frac{\alpha!}{(\alpha - \beta)!}\right)$$

parameters $\underline{p}$ needed to encode the variables $\underline{\nu}$ as functions $\underline{\nu}(\underline{p})$. The particular functions can be obtained in a similar way as for Encoding 1 - by solving (6.1.2) and (6.1.3) in the space of the parameters, and at the same time preserving the mutual exclusiveness of the $\underline{\nu}$ minterms. The procedure is illustrated in the following example.

Example : As before, let $\alpha = 3$, $\beta = 2$, then

$$|\underline{p}| = \lceil \log_2 \frac{3!}{(3-2)!} = 3 \rightarrow \underline{p} = \{p_1, p_2, p_3\}$$

(compared with 4 parameters in Encoding 1). To obtain the functions $\underline{\nu}(\underline{p})$ the following correspondence between the minterms of $\underline{\nu}$ (satisfying (6.1.2) and (6.1.3)) and the minterms of $\underline{p}$ may be selected:

$$\nu_1^1\,\bar\nu_2^1\,\bar\nu_3^1 \quad \bar\nu_1^2\,\nu_2^2\,\bar\nu_3^2 \quad = \quad p_1\,p_2\,p_3$$

$$\nu_1^1\,\bar\nu_2^1\,\bar\nu_3^1 \quad \bar\nu_1^2\,\bar\nu_2^2\,\nu_3^2 \quad = \quad p_1\,p_2\,\bar p_3$$

$$\bar\nu_1^1\,\nu_2^1\,\bar\nu_3^1 \quad \nu_1^2\,\bar\nu_2^2\,\bar\nu_3^2 \quad = \quad p_1\,\bar p_2\,p_3$$

$$\bar\nu_1^1\,\nu_2^1\,\bar\nu_3^1 \quad \bar\nu_1^2\,\bar\nu_2^2\,\nu_3^2 \quad = \quad p_1\,\bar p_2\,\bar p_3$$

$$\bar\nu_1^1\,\bar\nu_2^1\,\nu_3^1 \quad \nu_1^2\,\bar\nu_2^2\,\bar\nu_3^2 \quad = \quad \bar p_1\,p_2\,p_3 + \bar p_1\,\bar p_2\,p_3 \quad = \quad \bar p_1\,p_3$$

$$\bar\nu_1^1\,\bar\nu_2^1\,\nu_3^1 \quad \bar\nu_1^2\,\nu_2^2\,\bar\nu_3^2 \quad = \quad \bar p_1\,p_2\,\bar p_3 + \bar p_1\,\bar p_2\,\bar p_3 \quad = \quad \bar p_1\,\bar p_3$$

The individual function ν_j^i obtained by solving the above system are

$$\nu_1^1 = p_1\,p_2 \qquad \nu_2^1 = p_1\,\bar p_2 \qquad \nu_3^1 = \bar p_1$$

$$\nu_1^2 = \bar p_2\,p_3 + \bar p_2\,p_3 \qquad \nu_2^2 = p_1\,p_2\,p_3 + \bar p_1\,\bar p_3 \qquad \nu_3^2 = p_1\,\bar p_3$$

The module characteristic function being

$$\begin{aligned}\Phi_M = {} & [b_1(a_1\,p_1\,p_2 + a_2\,p_1\,\bar p_2 + a_3\,\bar p_1) \\ & + \bar b_1(\bar a_1 + \bar p_1 + \bar p_2)(\bar a_2 + \bar p_1 + p_2)(\bar a_3 + p_1)] \\ & \cdot [\, b_2(a_1\,\bar p_2\,p_3 + a_1\,\bar p_1\,p_3 + a_2\,p_1\,p_2\,p_3 + a_2\,\bar p_1\,\bar p_3 + a_3\,p_1\,\bar p_3) \\ & + \bar b_2(\bar a_1 + p_1\,p_2 + \bar p_3)(\bar a_2 + p_1\,\bar p_2 + p_1\,\bar p_3 + \bar p_1\,p_3 + \bar p_2\,p_3) \\ & \cdot (\bar a_3 + \bar p_1 + p_3)]\end{aligned}$$

For instance, a choice of $p_1 = 0$, $p_2 = 0$, $p_3 = 0$ will generate the mapping $b_1 = a_3$, $b_2 = a_2$. Any additional constraints could again be added by formulating them as equations in $\underline{p}$.

To compare the encodings, Table 6.1 shows the number of parameters required for various α and β, and the two types of encoding.

		$\lvert \underline{p} \rvert$	
α	β	Enc. 1	Enc. 2
2	1	1	1
2	2	2	1
3	1	2	2
3	2	4	3
3	3	6	3
4	1	2	2
4	2	4	4
4	3	6	5
4	4	8	5
5	5	15	7
7	5	15	12
10	5	20	15
5	10	30	$\alpha < \beta$!

Application to modules Ξ and Ω :

Module Ξ :

In order to allow for free inputs to the module M, the domain $\underline{a}$ is composed of the variables $\underline{x}$ and the variable v_i to which the mapping is to be made, i.e. $\underline{a} = \{\underline{x}, v_i\}$ for $b_i = v_i$. Thus $m \cdot (r + 1)$ functions ν^i_j are needed, (6.2.1) yields

$$v_i = \sum_{j=1}^{r} x_i \nu^i_j + v_i \nu^i_{r+1} , \qquad i = 1, \ldots, m.$$

If Encoding 1 is used then $m \cdot \ulcorner \log_2 (r + 1)$ parameters $\underline{p}$ are required. Encoding 2 would reduce the number of parameters $\underline{s}$ to $\log_2 \frac{(r + 1)!}{(r + 1 - m)!}$; however, only one free module input would be allowed in any mapping thus obtained. The Encoding 1 does not have this disadvantage, provided that (6.2.3) is modified before applying. Such a modification is rather difficult to make in case of Encoding 2 where even the number of parameters may have to be changed.

Considering again a problem with $r = 9$ and $m = 5$ then 20 parameters are required for Encoding 1, and only 15 for Encoding 2. That would bring the total number of variables in the module characteristic function to 34 and 29, respectively.

Let $\Xi\,(\underline{x}, \underline{v}, \underline{s})$ be the characteristic function of the module with any encoding in terms of the parameters $\underline{s}$. It should be noted that due to the presence of the v_i variable in $\underline{a}$, the function Ξ has multiple output states $\underline{v}$ associated with the states of $\underline{x}$ and $\underline{s}$ - it resembles a sequential module.

Module Ω :

Similarly as in module Ξ, the domain $\underline{a}$ consists of the variables $\underline{w}$ and y_i so that

$$\underline{a} = \{ \underline{w}, y_i \} \qquad \text{for} \quad b_i = y_i .$$

Hence

$$y_i = \sum_{j=1}^{n} w_j \nu_j^i + y_i \nu_{n+1}^i , \qquad i = 1, \ldots, q.$$

The Encoding 1 can be used if the constraint (6.1.3) is modified to permit multiple free outputs. As far as Encoding 2 is concerned, it could be used only if $q \leqslant n + 1$, and even then it is restricted to the maximum of 1 free output y_i only. Therefore, it seems that the Encoding 1 is more suitable here, since usually $q > n + 1$, and the modifications required to permit multiple free outputs in Encoding 2 are not simple to perform.

Let $\Omega(\underline{w}, \underline{y}, \underline{t})$ be the module characteristic function, the mappings being determined by the parameters $\underline{t}$ via Encoding 1.

Again, due to the presence of a free y_i in $\underline{a}$ the function Ω has a sequential character.

For a problem with $q = 10$ and $n = 4$ as before, some 30 parameters would be needed, bringing the total number of variables in Ω to 44. The only practically possible representation might be using cubical complexes, since a map would require over 10^{13} bits!

Determination of Feasible Mappings - Module Application :

The overall circuit characteristic function of (Ξ, M) in Figure 6.1 is given by

$$\Phi'_C(\underline{x}, \underline{s}, \underline{w}) = \sum_{\underline{r} \in B_2^m} \left(\overline{\Xi(\underline{x}, \underline{v}, \underline{s})} \cdot \Phi_C(\underline{v}, \underline{w}) \right) \qquad \text{(Lemma 2.3)}$$

This circuit has to realize the output characteristic function

$$\Phi_1(\underline{x}, \underline{w}, \underline{t}) = \prod_{\underline{y} \in B_2^q} \left(\overline{\Omega}\,(\underline{w}, \underline{y}, \underline{t}) + \Phi(\underline{x}, \underline{y}) \right)$$

$$= \overline{\sum_{\underline{y} \in B_2^q}} \left(\Omega \cdot \overline{\Phi} \right) \qquad \text{(Lemma 4.2)}$$

Hence $\Phi'_C \leftarrow \Phi_1$ must hold for all states of $\underline{w}$ and $\underline{x}$ with $\underline{s}$ and $\underline{t}$ taking constant values. Therefore, feasible mappings μ_x and μ_y exist if the equation $\prod_{\underline{w} \in B_2^n} \left(\overline{\Phi}'_C + \Phi_1 \right) = 1$ has an identity solution (constant) for the parameters $\underline{s}$ and $\underline{t}$. (The particular mappings are obtained by substituting the identity solution(s) to the mapping functions). By Theorem 2.3, the identity solutions are equivalent to the solutions of the truth equation

$$\sum_{\underline{x} \in B_2^r} \sum_{\underline{w} \in B_2^n} \left[\sum_{\underline{v} \in B_2^m} (\Xi \cdot \Phi_C) \cdot \sum_{\underline{y} \in B_2^q} (\Omega \cdot \overline{\Phi}) \right] = 0 \quad (6.1.5)$$

Hence by scanning the identity solutions, the most suitable mappings (under some criteria as in [5, 28]) can be selected. The choice may also be guided by the number of redundant domain variables which each such mapping would introduce [15, 28]. To determine whether an output characteristic function $\Phi(\underline{x}, \underline{y})$ has some solutions with $x_i \in \underline{x}$ redundant the following test can be performed.

If

$$\Phi(\underline{x}, \underline{y})\Big|_{x_i=1} \cdot \Phi(\underline{x}, \underline{y})\Big|_{x_i=0} = 1$$

is consistent then

$$\Phi(\underline{x}, \underline{y}) = 1$$

contains solutions with redundant x_i.

6.1.3 Evaluation

The method of module application without parametric encoding of the mappings μ_x, μ_y is rather simple and it should be possible to incorporate the procedure into any of the existing synthesis programs, such as [5, 15, 28].

The size of the arrays to be stored in [23] would be increased by m columns and at most by 2^m rows, where m is the number of module inputs. The module application is done once only, and then various mappings in the form of constraint equations can be applied in a search for a feasible decomposition. The optimizing principles could remain without any major change.

In order to estimate the complexity of parametric encoding of μ_x and μ_y, the size of the arrays/maps required to evaluate equation (6.1.5) will be examined. Estimation of the size of functions involved:

$\Phi(\underline{x}, \underline{y})$ - $r + q$ variables, maximum approximately 2^r rows in an array.

$\Phi_C(\underline{v}, \underline{w})$ - $m + n$ variables, maximum approximately 2^m rows.

$\Xi(\underline{x}, \underline{v}, \underline{s})$ - $r + m + |\underline{s}|$ variables, approximately $\dfrac{(r+1)!}{(r+1-m)!}$ rows (Encoding 2).

$\Omega(\underline{w}, \underline{y}, \underline{t})$ - $n + q + |\underline{t}|$ variables, maximum $(n+1)^q$ rows (Encoding 1).

Intermediate results in (6.1.5) :

$\sum_{\underline{v}}(\Xi \cdot \Phi_C)$ - since $\underline{v}$ can be deleted while forming the intersection, the total number of variables (to be stored) is $r + n + |\underline{s}|$

$\sum_{\underline{y}}(\Omega \cdot \overline{\Phi})$ - since $\underline{y}$ can be deleted while forming the relative complement (# operation), only $r + n + |\underline{t}|$ variables need to be stored.

$\sum_{\underline{x}} \sum_{\underline{w}} [\cdot]$ - the variables $\underline{x}$ and $\underline{w}$ may be deleted in the process of intersecting. Thus it is necessary to store $|\underline{s}| + |\underline{t}|$ variable maps or arrays.

Since $|\underline{s}| + |\underline{t}|$ is larger than any of the other variable requirements (see discussion about the Encoding 1 and 2) then the largest map to be stored is $2^{|\underline{s}| + |\underline{t}|}$ bits.

By examining the row requirements of the various arrays of functions, it seems that the largest array would be that formed at the last operation in (6.1.5). That would have $|s| + |t|$ columns, and a very conservative estimate of maximum $(n + 1)^q \cdot \frac{(r + 1)!}{(r + 1 - m)!}$ rows, which is based on the total number of possible module variable to problem variable mappings. If a 2 bit per variable encoding [35] is used then there are at most $2 \cdot (n+1)^q \cdot \frac{(r+1)!}{(r+1-m)!} \cdot (|\underline{s}| + |\underline{t}|)$ bits needed to store the result of (6.1.5). The actual bit requirement would very likely be much smaller than that, however.

Considering again the problem with

$$r = 9, \quad q = 10, \quad m = 5, \quad n = 4$$

then

$$|\underline{s}| = 15, \quad |\underline{t}| = 30,$$

and a $2^{45} \simeq 3.52 * 10^{13}$ bit map is needed (independently whether any mappings are feasible), compared to the maximum of

$$2 \cdot 5^{10} \cdot \frac{10!}{5!} \cdot (15 + 30) \simeq 2.7 * 10^{13}$$

bits for the array.

No computational examples are presented, since even for trivial problems the number of variables involved is rather high for hand manipulation. As far as computer aided synthesis is concerned then for simple problems, the forementioned modification to [28] could be made; however, problems of practical interest having large number of variables $r \geq 10$, $q \geq 10$, would require long computational times and have high memory requirements. If parametric encoding of the mappings is performed to allow for simultaneous processing of the various possibilities, the storage requirements are close to current technological limits. Hence, at present, the use of such synthesis programs seems to be limited to rather small problems where a particular type of a solution is sought, and where the task could not be performed "intuitively" because of the large number of possibilities involved. Therefore, the development of such a computer program did not seem to be justified for economic reasons.

A rather interesting possibility arises, however, if the computational capability of a digital computer is combined with human intuition via a computer graphics system. There, the computer would perform the computational tasks (array operations) and minor decisions based on consistency conditions of Boolean equation. The human designer could then guide the choice of modules, their placement, and also set the limits on possible mappings of problem to module variables. All the decisions could be guided by the future layout of the circuit board - a process rather similar to designing integrated circuit masks with the aid of a computer graphics system.

6.2 Application #2 : Fault Detection in Combinational Circuits

A current problem in switching circuit manufacturing and maintenance is fault detection [6, 9, 37, 38]. If the number of inputs to a given circuit is high then its testing by application of all possible input states to a combinational circuit or all input sequences to a sequential circuit becomes impractical. Considering only combinational circuits, it has been shown that it is sufficient to apply a subset of the possible input states to detect all single and multiple faults [6, 9, 37, 38].

A method will be shown, that will generate first all the input states which could detect a particular fault, and second, it will be extended to generate a minimum length test set T_s of input stimuli which would detect all single stuck at 0 and 1 faults in a combinational circuit built from combinational modules. The procedure is based on the relationship between circuit and output characteristic functions as developed in Chapter 4. It is especially easy to use if the circuit characteristic functions of the modules comprising the network are directly available, for instance, from the library of modules as in Application #1. Moreover, it will also be shown how the same methodology can be applied to generate tests for detecting multiple "stuck at" faults and bridging faults.

6.2.1 Detection of a Single Stuck at 0 or 1 Fault.

Let a combinational circuit be given, as shown in Figure 6.2. It is required to find all states of the input variables $\underline{x}$ (stimuli) which would detect a "stuck at" fault on a line z feeding into a module G of the circuit. That is, if any of those states $\underline{x}$ is applied to the circuit then a fault on the line z would produce an incorrect output $\underline{y}$.

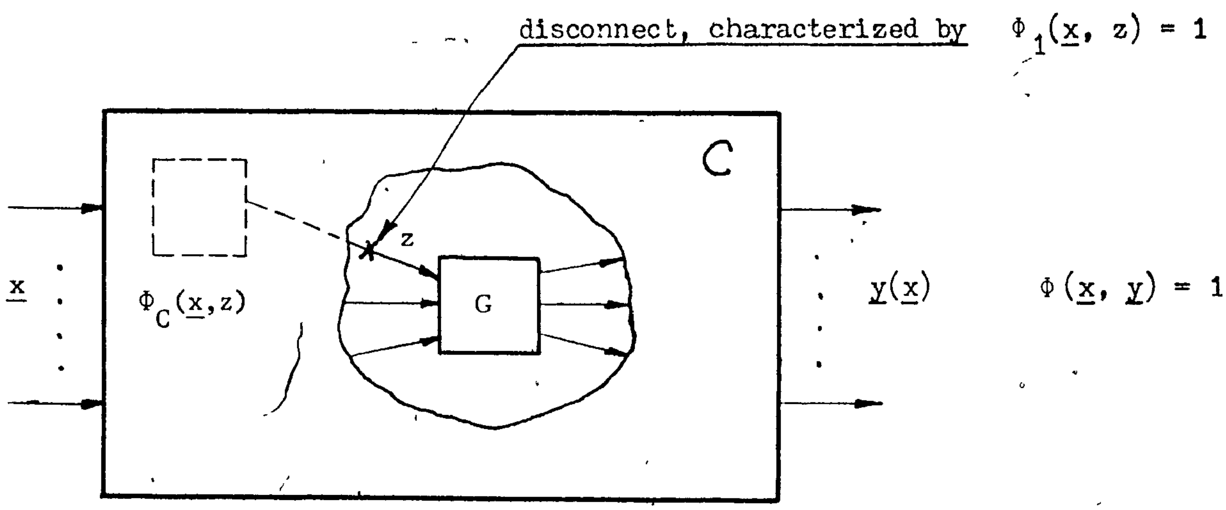


FIGURE 6.2. SINGLE FAULT DETECTION.

Let $\Phi(\underline{x}, \underline{y})$ be the overall circuit characteristic function without any faults present. (The circuit normally realizes that Φ). Furthermore, let the circuit characteristic functions of all the combinational modules which compose C be known. Since the internal structure of C is known, then by applying Theorem 4.2 and the Lemmas 4.2 - 4.3, the output characteristic function $\Phi_1(\underline{x}, z)$, describing the line z when disconnected from the rest of the circuit, can be obtained.

If $\Phi_C(\underline{x}, z)$ is the circuit characteristic function normally realizing z then

$$\Phi_C(\underline{x}, z) \leqslant \Phi_1(\underline{x}, z)$$

identically. A stuck at 1 line z actually produces $z = 1$, whose characteristic function is $\Phi_C'(\underline{x}, z) = z$. Therefore, the realization condition is

$$z \leqslant \Phi_1(\underline{x}, z).$$

Two cases can now occur, either the relation is satisfied identically, then z is redundant in C and the fault cannot be detected, or the relation holds only for some states of $\underline{x}$. For all the other states of $\underline{x}$ the function $\Phi_1(\underline{x}, z)$ is not realized, thus $\Phi(\underline{x}, \underline{y})$ is not realized, and consequently, an incorrect output $\underline{y}$ is produced. Therefore, these input states would detect the fault. The relation,

$$z \leqslant \Phi_1(\underline{x}, z)$$

is not realized at the inconcistency points of the equation

$$\prod_{z \in B_2} (\bar{z} + \Phi_1(\underline{x}, z)) = 1 \tag{6.2.1}$$

After simplification - all input states $\underline{x}$ which would detect $z = 1$

form the solutions to the truth equation

$$\overline{\Phi}_1(\underline{x}, 1) = 1 \tag{6.2.2}$$

Similarly, a stuck at 0 fault is represented by $z = 0$, and the test states are the solutions to

$$\overline{\Phi}_1(\underline{x}, 0) = 1 \tag{6.2.3}$$

<u>Example</u> : Consider a circuit shown in Figure 6.3 [6, 7]. The function $y(a, b, c, d)$ generated by the circuit is

$$y = a\,b\,\bar{c} + a\,b\,\bar{d} + \bar{a}\,c\,d + \bar{b}\,c\,d$$

Hence

$$\Phi(a, b, c, d, y) = y \cdot (a\,b\,\bar{c} + a\,b\,\bar{d} + \bar{a}\,c\,d + \bar{b}\,c\,d)$$
$$+ \bar{y} \cdot (\bar{a}\,\bar{c} + \bar{b}\,\bar{c} + \bar{a}\,\bar{d} + \bar{b}\,\bar{d} + a\,b\,c\,d)$$

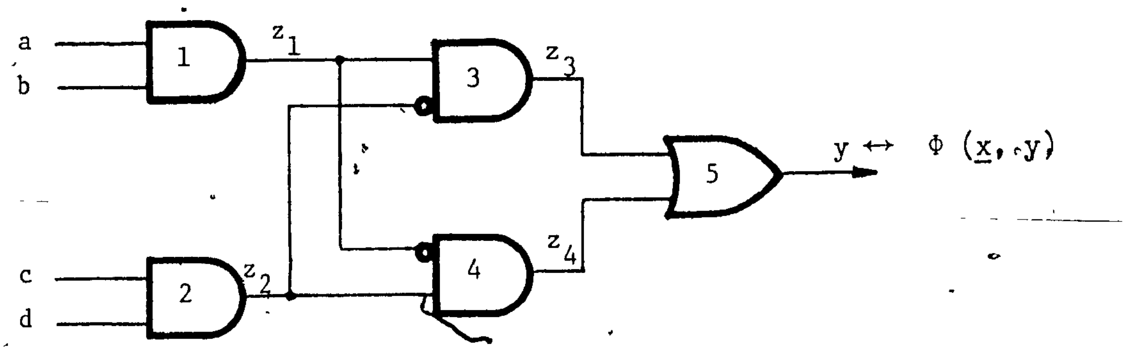


FIGURE 6.3. EXAMPLE FOR SECTION 6.2.1.

The circuit characteristic functions for gates 1, 2, 3, 4 and 5 are

$$\Phi_{C1}(a, b, z_1) = z_1 a b + \bar{z}_1(\bar{a} + \bar{b})$$

$$\Phi_{C2}(c, d, z_2) = z_2 c d + \bar{z}_2(\bar{c} + \bar{d})$$

$$\Phi_{C3}(z_1, z_2, z_3) = z_1 \bar{z}_2 z_3 + \bar{z}_1 \bar{z}_3 + z_2 \bar{z}_3$$

$$\Phi_{C4}(z_1, z_2, z_4) = \bar{z}_1 z_2 z_4 + z_1 \bar{z}_4 + \bar{z}_2 \bar{z}_4$$

$$\Phi_{C5}(z_3, z_4, y) = y z_3 + y z_4 + \bar{y} \bar{z}_3 \bar{z}_4$$

Note that all gates are combinational so that the simpler form of the expression in Lemma 4.2 can be used (see Remark following Lemma 4.2). Let a fault be assumed on line z_3. Then the following steps will generate the test input states of (a, b, c, d). Circuits 1 and 2 combined:

$$\begin{aligned}
\Phi_{C1,2}(a, b, c, d, z_1, z_2) &= \Phi_{C1}(a, b, z_1) \cdot \Phi_{C2}(c, d, z_2) \\
&= z_1 z_2 a b c d + \bar{z}_1 z_2(\bar{a} c d + \bar{b} c d) \\
&+ z_1 \bar{z}_2(a b \bar{c} + a b \bar{d}) \\
&+ \bar{z}_1 \bar{z}_2(\bar{a} \bar{c} + \bar{a} \bar{d} + \bar{b} \bar{c} + \bar{b} \bar{d})
\end{aligned}$$

Circuits 1, 2 and 4 combined : (Lemma 2.3)

$$\Phi_{C1,2,4}(a, b, c, d, z_4)$$

$$= \sum_{z_1 z_2 \in B_2^2} (\Phi_{C1,2} \cdot \Phi_{C4})$$

$$= z_4(\bar{a}\, c\, d + \bar{b}\, d\, c)$$

$$+ \bar{z}_4(a\, b\, \bar{c} + a\, b\, \bar{d} + \bar{a}\, \bar{c} + \bar{a}\, \bar{d} + \bar{b}\, \bar{c} + \bar{b}\, \bar{d} + a\, b\, c\, d)$$

The output characteristic function $\Phi_{1,2,3,4}$ generated through gate 5 : (Lemma 4.2)

$$\Phi_{1,2,3,4}(a, b, c, d, z_3, z_4)$$

$$= \sum_{y \in B_2} (\Phi' \cdot \Phi_{C5})$$

$$= (z_3 + z_4)(a\, b\, \bar{c} + a\, b\, \bar{d} + \bar{a}\, c\, d + \bar{b}\, c\, d)$$

$$+ \bar{z}_3\, \bar{z}_4(\bar{a}\, \bar{c} + \bar{b}\, \bar{c} + \bar{a}\, \bar{d} + \bar{b}\, \bar{d} + a\, b\, c\, d)$$

Hence the output characteristic function for z_3 (Lemma 4.4) :

$$\Phi_3(a, b, c, d, z_3)$$

$$= \sum_{z_4 \in B_2} (\Phi_{1,2,3,4} \cdot \Phi_{C1,2,4})$$

$$= \underline{\underline{z_3(a\, b\, \bar{c} + a\, b\, \bar{d}) + \bar{z}_3(\bar{a}\, \bar{c} + \bar{b}\, \bar{c} + \bar{a}\, \bar{d} + \bar{b}\, \bar{d} + a\, b\, c\, d)}}$$

$$\underline{\underline{+ \bar{a}\, c\, d + \bar{b}\, c\, d}}$$

z_3 stuck at 0 :

$$\overline{\Phi}_3(a,\ b,\ c,\ d,\ 0) = a\, b\, \bar{c} + a\, b\, \bar{d}$$

Therefore, any of the following stimuli would detect that s-a-0 fault :

a	b	c	d
1	1	0	0
1	1	0	1
1	1	1	0

z_3 stuck at 1 :

$$\overline{\Phi}_3(a,\ b,\ c,\ d,\ 1) = \bar{a}\, \bar{c} + \bar{b}\, \bar{c} + \bar{a}\, \bar{d} + \bar{b}\, \bar{d} + a\, b\, c\, d$$

Hence any of the following stimuli would detect the s-a-1 fault :

a	b	c	d
1	1	1	1
0	0	0	0
0	0	0	1
0	1	0	0
0	1	0	1
1	0	0	1
0	0	1	0
0	1	1	0
1	0	1	0

The states 0 0 1 1, 0 1 1 1 and 1 0 1 1 would detect neither fault.

6.2.2 Minimal Length Single Fault Test Sets

It has been shown [9] that all single faults in a combinational circuit are detected if all single faults on the circuit's checkpoints are detected. The checkpoints being defined as all primary inputs which do not fanout and all fanout branches.

Let z_i, $i = 1, \ldots, k$ be some variables marking all the lines associated with the checkpoints of a circuit C. Before proceding to generate a minimal length test set covering all single faults on the k checkpoints, the sets of test stimuli for each line z_i must be generated first by the method shown in Section 6.2.1. Therefore, let

$$\Phi_i(\underline{x}, z_i) , \qquad i = 1, \ldots, k$$

be the output characteristic functions associated with these lines. Furthermore, let

$$\Psi_i^0(\underline{x}) = \overline{\Phi}_i(\underline{x}, 0) ,$$

$$\Psi_i^1(\underline{x}) = \overline{\Phi}_i(\underline{x}, 1) .$$

Hence a stimulus $\underline{x}^*$ will detect a stuck at 0 or 1 fault on line z_i if

$$\Psi_i^0(\underline{x}^*) = 1$$

or

$$\Psi_1^1(\underline{x}^*) = 1 ,$$

respectively.

A minimal length test set T_s is defined as the smallest set of input stimuli which would detect all single "stuck at" faults on the checkpoints. Therefore, in terms of the Ψ_i functions, it is the smallest set such that

$$\forall\ i\ , \quad \exists\ \underline{x}^1 \in T_s \ \ni\ \Psi_i^0(\underline{x}^1) = 1$$

$$\text{and} \quad \exists\ \underline{x}^2 \in T_s \ \ni\ \Psi_i^1(\underline{x}^2) = 1.$$

Hence the determination of T_s can be formulated as a covering problem. As such it could be solved using a covering table with its rows labelled by all states of $\underline{x}$ (B_2^n), and the columns labelled by the functions Ψ_i^0, Ψ_i^1, $i = 1, \ldots, k$. A check mark is then placed in an entry $(\underline{x}^j, \Psi_i^{0(1)})$ if $\Psi_i^{0(1)}(\underline{x}^j) = 1$. A minimal cover of the functions Ψ_i must then be obtained by any extraction procedure.

A major disadvantage of the tabular method above is that the table is extended over all states of $\underline{x}$. A smaller covering table could be obtained by first forming pairwise intersections [9] of the functions Ψ_i so as to find the minterms common to the largest possible subsets of these functions.

A purely algebraic method based on mutual intersections of the functions Ψ_i could be developed as follows:

Let $\underline{p}$ be a set of 2k parameters

$$\underline{p} = \{p_1^0, p_1^1, p_2^0, p_2^1, \ldots, p_k^0, p_k^1\},$$

and form a function

$$P = \prod_{i=1}^{k} (p_i^0 + \Psi_i^0(\underline{x})) \cdot (p_i^1 + \Psi_i^1(\underline{x})).$$

Delete all terms in the $\Sigma\Pi$ form of P, whose $\underline{p}$ part of the product subsums another term's $\underline{p}$ part. After all such terms were deleted the modified function P^- contains only the largest non-zero products of the function Ψ_i. The variables from $\underline{p}$ contained in a term mark those functions Ψ_i which did not take part in that product term.

At this point, let P* be the smallest subfunction (implicant) of P, such that the set formed as a union of all the missing parameters p_i^0 or p_i^1 from all the product terms of P* is equal to the complete set $\underline{p}$. A set T_s is then formed as follows: For each term in P* select a single state $\underline{x}^j$ for which the term becomes dependent only on the parameters $\underline{p}$ (the $\underline{x}$ part is equal to 1), then $\underline{x}^j$ is a member of T_s.

The method was presented without a formal proof, but it closely follows the method of pairwise intersections and finding the smallest cover.

6.2.3 Detection of Other Types of Faults.

Multiple "stuck at" Faults :

From all the possible multiplicity of faults, only double faults will be considered here for reasons of simplicity. However, the method is easily extendable to cover any fault multiplicity.

Let z_1 and z_2 be two checkpoints in a circuit C for which a double fault test is to be generated. Similarly as in Section 6.2.1, let the lines z_1 and z_2 be disconnected and their output characteristic function $\Phi_{12}(\underline{x}, z_1, z_2)$ obtained.

$$\Psi^{11}(\underline{x}) = \bar{\Phi}_{12}(\underline{x}, 1, 1)$$

$$\Psi^{10}(\underline{x}) = \bar{\Phi}_{12}(\underline{x}, 1, 0)$$

$$\Psi^{01}(\underline{x}) = \bar{\Phi}_{12}(\underline{x}, 0, 1)$$

$$\Psi^{00}(\underline{x}) = \bar{\Phi}_{12}(\underline{x}, 0, 0) .$$

A double fault $z_1 = \alpha$, $z_2 = \beta$; α, $\beta \in B_2$, can thus be detected by all states of $\underline{x}$ for which

$$\Psi^{\alpha\beta}(\underline{x}) = 1.$$

Minimal length test sets can then be obtained by a similar method as discussed in Section 6.2.2.

Bridging Faults :

A bridge between lines z_1 and z_2 will force $z_1 = z_2$. Such fault would be then detected by all the states of $\underline{x}$ which are the inconsistency points of the equation

$$\Phi_{12}(\underline{x}, z_1, z_1)(z_1 z_2 + \bar{z}_1 \bar{z}_2) = 1$$

with respect to the solutions $z_1(\underline{x})$, $z_2(\underline{x})$. In other words, let

$$\Psi(\underline{x}) = \overline{\sum_{z_1, z_2 \in B_2^2} \Phi_{12}(\underline{x}, z_1, z_2)(z_1 z_2 + \bar{z}_1 \bar{z}_2)} ,$$

then a stimulus $\underline{x}^*$ will detect the particular bridging fault if

$$\Psi(\underline{x}^*) = 1.$$

6.2.4 Concluding Remarks

Using the methodology described in the preceeding sections, test sets of input stimuli for detecting various types of faults can be generated. Minimum length test set can then be obtained by a covering procedure mentioned in Section 6.2.2. The entire test generating procedure could easily be programmed for a digital computer, especially if cubical complexes are used to represent the various functions. The algorithm could also be incorporated into the computer aided synthesis program mentioned in Section 6.1, since both procedures would use the same library of module characteristic functions.

CHAPTER 7

CONCLUSION

In this final chapter, an overall summary of the original contributions described in the preceeding sections is presented, and natural extensions of these contributions are put forward as topics for further research.

7.1 Summary

7.1.1 Theoretical Aspects

The roots of the work lie in the theory of systems of Boolean equations. Therefore, some major topics related to the formation of a characteristic equation $\Phi(\underline{x}, \underline{y}) = 1$ (function $\Phi(\underline{x}, \underline{y})$) of a system of B.E., to its consistency, and to the methods of solution were reviewed in the first three sections of Chapter 2. The following properties of B.E. pertinent to the research were then elaborated upon in Section 2.4:

- Detection of a unique solution and of identity solutions by a simple algebraic method.
- Determination of a characteristic equation equivalent to a system of two equations related by the fact that the solutions of the first equation are required to form the domain of the second equation (Lemma 2.3).

The second case played an important role when an analysis of switching circuits was performed in Chapters 3 and 4. First, however, the concept of a characteristic function (equation) of a system of B.E. was applied to characterize a general circuit (Figure 3.1) represented by a Mealy type machine (Definition 3.2). It resulted in the definitions of a number of special characteristic functions which related the input stimuli with the internal states and the output responses of the circuit. (Definitions 3.2, 3.3, 3.6, 3.12, 3.14). Mutual relationship between these functions was analyzed in Lemma 3.2, Theorem 3.1, while Lemma's 3.1, 3.3, 3.4 and 3.5 showed how a description of the stable, oscillatory and transient states of the circuit can be obtained from the corresponding characteristic equations, by analyzing their solutions and consistency. The above steps eventually lead to the formulation of a circuit characteristic function $\Phi_c(\underline{x}, \underline{y})$ (Theorem 3.3, Definition 4.1) which related, through $\Phi_c = 1$, the input states $\underline{x}$ with the steady output states $\underline{y}$ that may possibly be assumed by the circuit if a proper input sequence is applied.

A trivial case when a circuit has combinational behaviour was studied in Theorem 3.2. Furthermore, it was shown in Theorem 3.3 that combinational behaviour (Definition 3.13) of an internally sequential circuit is characterized by the existence of a unique solution to $\Phi_c(\underline{x}, \underline{y}) = 1$. Assuming the internal structure of the circuit to consist of a next state generator $\underline{g}$ and an output generator $\underline{f}$ (Figure 3.1), the function $\Phi_c(\underline{x}, \underline{y})$ was formed as

$$\sum_{\underline{z}'} (\Phi_g^c \cdot \Phi_f)$$

(Definitions 5.12, 3.14; Theorem 3.3) by application of Lemma 2.3. The expression gives an insight into the mechanism by which a sequential circuit may produce a combinational output. Namely, that the output generator $\underline{f}$ filters out the possible multiple internal steady states $\underline{z}$ of $\underline{g}$ for each input state $\underline{x}$, and thus a unique output state $\underline{y}$ is generated.

Realization of multiple output incompletely specified switching functions $\underline{y}(\underline{x})$ by combinational and sequential networks was then treated in Chapter 4. An output characteristic function $\Phi(\underline{x}, \underline{y})$ (Definition 4.2) was formally defined in such a way that the maximum permissible steady output states which a circuit may assume so as to satisfy some requirements are described by the solutions of

$$\Phi(\underline{x}, \underline{y}) = 1.$$

If a particular circuit (represented by Φ_c) operates within these states, then that fact was defined as a realization of the output characteristic function, and it was shown that in such a case the relation

$$\Phi_c(\underline{x}, \underline{y}) \leqslant \Phi(\underline{x}, \underline{y})$$

is satisfied (Lemma 4.1). The original requirement placed on a circuit was to realize $\underline{y}(\underline{x})$; however, it was demonstrated in Theorem 4.1 that the realization of $\underline{y}(\underline{x})$ corresponds to the realization of an output characteristic function derived from the system of relations

$$\underline{y}(\underline{x}) = < \underline{\phi}(\underline{x}), \ \underline{\phi}(\underline{x}) + \underline{d}(\underline{x}) > .$$

Consequently, the function $\underline{y}(\underline{x})$ is realized by a circuit C if the $\leqslant$ relation between the respective characteristic functions is satisfied. This notion was further extended to cover realizations by circuits which consist of cascade and parallel interconnections of functional modules. By assuming that some of the modules in the circuit are still unknown, the problem of functional decomposition was shown to be equivalent to the problem of solving a system of 2 Boolean equations related by the order ($\leqslant$) relation. Thus the various approaches to decomposition as mentioned in Section 4.1 were unified and expressed under one methodology in Lemma's 4.2, 4.3 and 4.4. Their combined effect was then stated in the main decomposition theorem (Theorem 4.2) and its Corollary.

The unifying impact of the methodology of characteristic functions on functional decomposition can be summarized as follows :

Independently of the position of the still unknown module inside the network two conditions must be satisfied for a decomposition/realization to exist, namely,

(1) a decomposition exists if the corresponding output characteristic equation $\Phi(\underline{x}, \underline{y}) = 1$ derived for the unknown module is consistent;

(2) a circuit module C will complete the realization provided that $\Phi_C \leqslant \Phi$ is satisfied.

The particular output characteristic function Φ can be obtained as specified in the forementioned Lemma's or Theorem 4.2.

The condition (2) above is valid even if the module is of a sequential character, provided that the circuit characteristic function properly describes its steady output states under the possible input stimuli. Furthermore, if a combinational circuit is desired as the missing module, then its output function (of $\underline{x}$ only) can be obtained as an elementary solution of the corresponding output characteristic equation $\Phi = 1$.

Having shown that circuits with feedback loops (and thus with multiple output states) could be used to realize combinational switching functions, a more detailed inquiry into the necessity of such loops in minimal circuits realizing a given output characteristic function was performed in Section 4.3. There, Theorem 4.3 stated a necessary condition under which the feedback in a sequential circuit g realizing some Φ is redundant. If the condition is satisfied then g could be made to degenerate into a purely combinational equivalent g^* obtained by freezing the feedback inputs of g at some constant value. Moreover, it was demonstrated that the resulting feedbackless circuit would not be as complex as the sequential original. A subclass of the degenerate cases (for some Φ) was named "direct transition circuits". They are characterized by the property that any transition from an unstable state leads directly to a state permitted by the output characteristic function. Since any quasi-normal (0-transition) circuit realizing Φ belongs to this subclass, any non-degenerate circuit must be passing through transient states ($k \geqslant 1$ transition). Furthermore, some of those states must not be permitted by Φ (Corollary 2 to Theorem 4.3), that is, the output characteristic function must not be realized during the transitions. The immediate implication is that although feedback in combinational networks might possibly reduce the overall cost, the behaviour of such circuits would be inherently hazardous during their transition periods (Corollary 3 to Theorem 4.3). As a matter of fact, the existence of such transient states describes the main difference between parallel and serial information processing systems.

That is, parallel processing (in space) generates the steady state responses faster (no transient states), but more hardware may be required as compared to a serial case where the processing is done by iterations in time. There, however, some internal memory (feedback) is required to store the intermediate results (transient states) which do not yet form the correct final output. A simple example is a fast parallel adder as compared with its serial equivalent.

Even though the direct transition circuits are of no practical value, for they are always degenerate with respect to the output characteristic function realized, it was shown in Theorem 4.4 that any such circuit can be obtained via a general solution $\underline{y} = \underline{\eta}(\underline{x}, \underline{p})$ of the output characteristic equation

$$\Phi\,(\underline{x}, \underline{y}) = 1.$$

Unfortunately, a similar procedure has not yet been discovered for the non-degenerate cases; however, as already mentioned, the Corollary 2 to Theorem 4.3 stated some of the necessary conditions that such circuits must satisfy.

7.1.2 Computational Aspects

The theoretical work of Chapters 2, 3 and 4 was done independently of any particular form of Boolean function representation. However, in order to illustrate the results on examples, some computational form with associated operations had to be selected. Two typical forms were reviewed in Chapter 5, Section 5.1, namely, the Marquand maps and the cubical complexes (arrays). It was shown then in Section 5.2 (Algorithms) and Section 5.3 (Examples) that the formal descriptions of the previous chapters could easily be transcribed into either of the computational forms. This ease of conversion is due to the fact that the methodology is independent of any data structure. Nevertheless, cubical complexes and operations seem to yield the most transparent representation of the algorithms, since some of the operations defined on arrays have an immediate meaning in the context of characteristic functions (e.g.,

$$D_{\underline{y}} \; , \leftrightarrow \sum_{\underline{y} \in B_2^{|\underline{y}|}} \; ; \qquad \Phi_1 \cdot \bar{\Phi}_2 \leftrightarrow \Phi_1 \# \Phi_2 \; , \text{ etc }).$$

Also, the array operators allow for easy description of computational procedures in a way similar to programming languages.

The above mentioned close relationship between characteristic functions and cubical complexes becomes especially apparent when the so called "function array" [7] of $\underline{y}(\underline{x})$ is compared with the array of the output characteristic function of $\underline{y}(\underline{x})$ as used in Theorem 4.1. It can be shown that the two arrays are cover equivalent (compare Algorithm 5.10 here with Algorithm 3.5 of [7]). Similarly, the circuit characteristic function of a purely combinational circuit corresponds to the function array describing that circuit. However, the characteristic function as a concept is more general, since it allows for representing both combinational and sequential circuits in a unified manner. Even more important is the fact, though, that characteristic functions can be analyzed using the theory of B.E., independently of the form in which they would eventually be represented. Thus consistency of an array, functional realization, decomposition, redundancy of input and state variables, and combinational behaviour can be precisely defined using the methodology of characteristic functions as corresponding to the consistency of a Boolean equation, satisfaction of the $\leqslant$ relation between two equations, the existence of a solution in a system of two equations related by $\leqslant$, the existence of a solution to a B.E. under related constraints, and the presence of a unique solution in a circuit characteristic equation, respectively. Similar definitions can be developed to describe a number of other properties of switching functions and circuits (e.g. symmetry, existence of disjoint decompositions, etc.). In other words, if "function arrays" are consistently replaced by arrays of characteristic functions then the properties of the cubical complex representation of multiple output functions can be studied using a powerful tool - the theory of B.E.

7.1.3 Applications

In order to demonstrate that the methodology can be used to solve problems of practical interest, the material in Chapter 6 concentrated on developing two particular applications. It was shown that:

(1) The methodology of characteristic functions permits a rather flexible approach to modular synthesis of combinational circuits by decomposition, and that

(2) the output characteristic functions (equations) describing the internal structure of a combinational circuit carry enough information to allow for a unified approach to the generation of test sets of input stimuli for detecting various types of faults inside such circuits.

In both cases above, the development of solution steps was done without any dependence on a data structure; however, due to the properties of the methodology as stressed in Section 7.1.2, the conversion to either map or array algorithms is rather trivial.

The first application showed that by unifying the various directions taken in functional decomposition (Section 4.1) under one theory, the position of a building module (combinational or sequential) inside the future circuit need not be determined in advance. Thus an output characteristic function describing a "floating" module is obtained first, and then the module inputs and outputs can be fixed by applying mapping constraint equations so as to yield the best conditions for satisfying the particular circuit optimization goals currently in use. Moreover, the mapping constraints could be combined into two hypothetical mapping modules (Ξ and Ω) which are controlled by constant parameters. It was demonstrated that if these modules are used in conjunction with the building module, then a truth equation for the parameters can be constructed so that any solution to the equation describes a feasible mapping, and thus a valid decomposition. Either procedure is relatively simple to incorporate into the optimizing routines of some of the existing computer aided synthesis algorithms. However, the storage/time requirements of these programs would still remain high (by present day standards) to be feasible for solving problems of practical interest. Therefore, no actual computer program was developed, since its design was not justified on economic grounds.

More practical results were obtained in the second application dealing with fault detection. It was shown that a set of stimuli which would detect a particular fault corresponds to the set of inconsistency points (singularities) of an output characteristic function (equation) derived for the line ~~at~~ which the fault is assumed. Furthermore, it was demonstrated that single and multiple "stuck at" as well as bridging faults could be detected using the same method. A covering procedure was then proposed for selecting minimal length test sets suitable for detecting all single "stuck at" faults. The procedure could be extended to include other types of faults mentioned before, and its conversion to a computer algorithm should be relatively easy, since the methodology of characteristic functions is used throughout.

7.2 Future Research

The following areas seem to be immediately suitable for further exploration based on the methodological tools developed here.

7.2.1 Circuit Synthesis

(a) The design of non-degenerate sequential circuits which would realize a given output characteristic function.

(b) Improvements in general synthesis programs as outlined in Chapter 6, with the possibility of including external feed-back loops, depending on the results obtained in (a) above.

(c) Application of the methodology of characteristic functions to the decomposition of sequential circuits with associated hazard analysis.

(d) Due to the similarity between array representation of characteristic functions and "function arrays" as discussed in Section 7.1.2, a possibility for research arises in the development of efficient algorithms for generating minimal solutions to Boolean equations. It seems likely that some of the procedures used for minimizing or reducing multiple output functions in an array form could be altered so as to perform the task.

7.2.2 Fault Detection

(a) Development of efficient computer algorithms which would generate minimal length test sets for detecting various types of faults in combinational circuits, as discussed in Section 6.2.

(b) Expansion of the fault detection method of Chapter 6 to cover sequential circuits. It seems that the state transition characteristic function (Definition 3.2) would be a suitable tool for such an analysis.

(c) A corollary to fault detection - the design of easily testable circuits by monitoring faults of their submodules using circuits which implement the output characteristic functions associated with the submodules, as shown in Figure 7.1. The method might be especially attractive in the case of LSI circuits whose internal structure is very complex, and where there is no access to the various sub-circuits contained on the chips. However, there is a number of major problems which have to be resolved first, namely :

(i) What form the circuits implementing an output characteristic function should have.

(ii). How complex such a monitoring technique would be as compared to current methods, which usually implement a given circuit twice (or more) and then use comparators to decide whether a correct output response is obtained.

(iii) What output characteristic function should be used if the module monitored is redundant in the overall network.

Possible advantages of the characteristic function approach could be summarized as follows :

(a) The checking circuit would have a structure different from that of the module monitored, thus the chance of inducing the same manufacturing faults in both circuits would be decreased, resulting in more reliable performance.

(b) If Φ_C is the circuit characteristic function and Φ the output characteristic function of the module, then all the states in $\Phi \cdot \overline{\Phi}_C$ can be considered as don't care ($\Phi_C \leqslant \Phi$ is assumed) when designing the checking circuit.

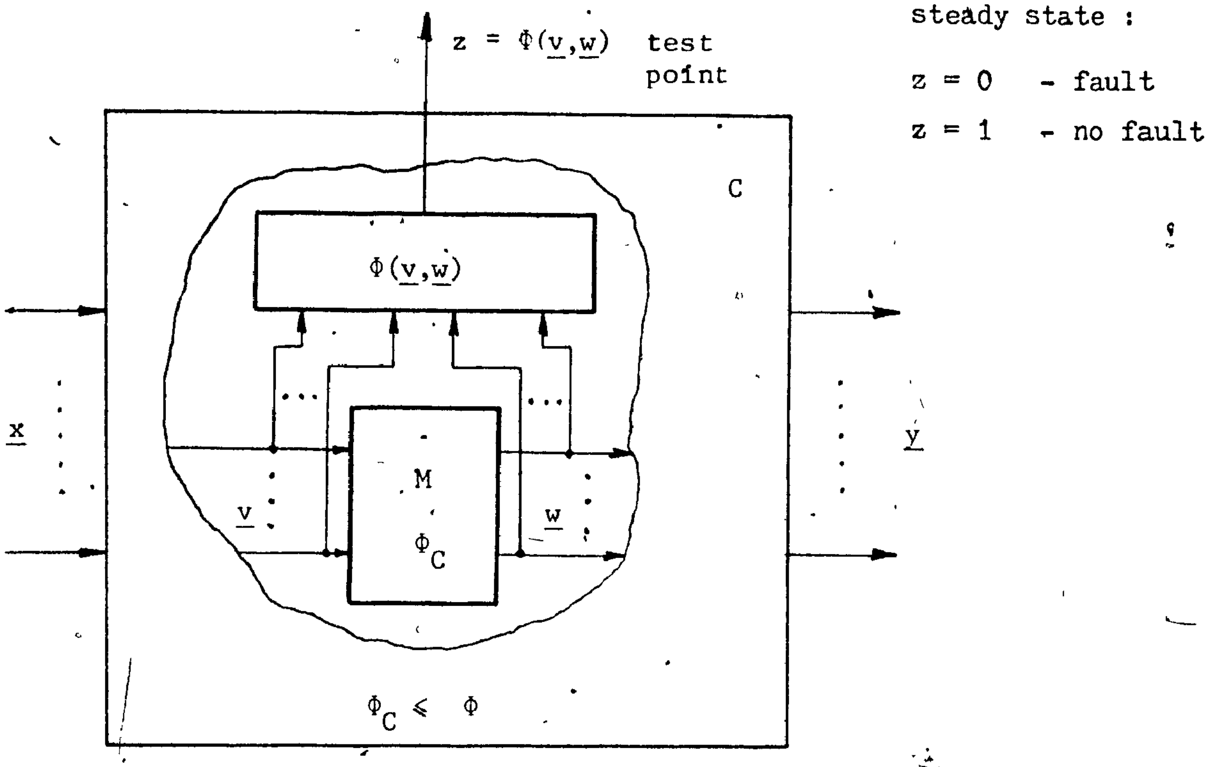


FIGURE 7.1. FAULT MONITORING.

7.3 Final Remarks

In order to underline the title of the thesis, the following are the main unifying aspects of the methodology of characteristic functions :

(a) The representation of combinational and sequential circuits can be unified with respect to the synthesis of combinational networks.

(b) The various directions taken in the synthesis of combinational circuits (Section 4.1) may be approached using one methodology.

(c) Since "function arrays" were shown to be just special cases of characteristic functions represented as arrays, the methodological tools of Boolean equations can be used for developing and analyzing computational procedures which are based on that data structure.

(d) The formulation of detection procedures for various types of faults in combinational circuits may be done using a unified approach.

Though the methodology might seem to be computationally more complex, especially in the case of simple circuits, its advantage lies in the underlying philosophy which allows for describing solutions to various logic design problems in terms of simple concepts related to the properties of Boolean equations.

REFERENCES

1. Ashenhurst, R.L., "The Decomposition of Switching Functions", Proc. International Symp. Theory of Switching, Vol. 29 of Annals of Computation, Laboratory of Harvard Univ., Cambridge, Massachusetts, 1959, pp. 74.

2. Brewer, M.A., (Editor) Design Automation of Digital Systems, Prentice-Hall, 1972.

3. Brown, F.M., "Reduced Solutions of Boolean Equations, IEEE TC, October 1970, pp. 976.

4. Brown, F.M., "Equational Logic", IEEE TC, Vol. C-23, #12, December 1974, pp. 1228.

5. Cerny, E., Marin, M.A., "A Computer Algorithm for the Synthesis of Memoryless Switching Circuits", IEEE TC, May 1974, C-23, pp. 455.

6. Chang, H.Y., Manning, E., Metze, G., Fault Diagnosis of Digital Systems, Wiley-Interscience, 1970.

7. Dietmeyer, D., Logic Design of Digital Systems, Allyn and Bacon, 1970.

8. Even, S., Meyer, A.R., "Sequential Boolean Equations", IEEE TC, March 1969, C-18, #3, pp. 230.

9. Fridrich, M., Davis, W.A., "Minimal Fault Tests for Combinational Networks", IEEE TC, Vol. C-23, #8, August 1974, pp. 850.

10. Hammer, P.L., Rudeanu, S., Boolean Methods in Operations Research, Springer Verlag, New York, 1968.

11. Hu, S.T., Mathematical Theory of Switching Circuits and Automata, UCLA Press, Berkeley, 1968.

12. Huffman, D.A., "Logical Design with One NOT Element", Proc. 2nd Hawaii Int. Conf. on Syst. Sciences, Honolulu, Hawaii, 1969, pp. 735.

13. Karp, R.M., et al, "A Computer Program for the Synthesis of Combinational Switching Circuits", Proc. AIEE Annual Symp. Switching Circuit Theory, 1961, pp. 182.

14. Kautz, W.H., "The Necessity of Closed Circuit Loops in Minimal Combinational Circuits", IEEE TC, February 1970, pp. 162.

15. Kjelkerud, E., "A Computer Program for the Synthesis of Switching Circuits by Decomposition", IEEE TC, Vol. C-21, #6, June 1972, pp. 568.

16. Klir, G.J., Introduction to the Methodology of Switching Circuits, D. Van Nostrand, 1972.

17. Klir, G.J., Marin, M.A., "New Considerations in Teaching Switching Theory", IEEE Trans. on Education, Vol. E-12, #4, December 1969, pp. 257.

18. Klir, G.J., "On the Solution of Boolean and Pseudo-Boolean Relations", IEEE TC, Vol. 23, #10, October 1974, pp. 1098.

19. Marin, M.A., "Investigation of the Field of Problems for the Boolean Analyser", Ph.D. Dissertation, UCLA, Los Angeles, Rep. #68-28, 1968.

20. Marin, M.A., "Some New Applications of the Boolean Analyser", Proc. International Symposium on Design and Applications of Logic Systems, Brussels, September 1969.

21. Miller, R.E., Switching Theory, Vol. I and II, J. Wiley & Sons, New York, 1969.

22. Muroga, S., Ibaraki, T., "Design of Optimal Switching Networks by Integer Programming", IEEE TC, Vol. C-21, #6, June 1972, pp. 573.

23. Patt, Y.N., "Synthesis of Switching Functions Using Complex Logic Modules", Spring Joint Computer Conference, AFIPS Conf. Proc., 1967, pp. 699.

24. Roth, J.P., et al, "A Computer Program for the Synthesis of Combinational Switching Circuits", Proc. AIEE Symp, Sw. Circuit Theory and Logical Design, 1961.

25. Roth, J.P., "Algebraic Topological Methods for the Synthesis of Switching Systems in n Variables", The Institute of Advanced Studies, Princeton, ECP 56-02, April 1956.

26. Rudeanu, S., Boolean Functions and Equations, North-Holland, 1974.

27. Rudeanu, S., "An Algebraic Approach to Boolean Equations", IEEE TC, Vol. C-23, #2, February 1974, pp. 206

28. Schneider, P.R., Dietmeyer, D.L., "An Algorithm for Synthesis of Multiple Output Combinational Logic", IEEE TC, Vol. C-17, #2, February 1968, pp. 117.

29. Shen, V.Y., McKellar, A.C., "An Algorithm for the Disjunctive Decomposition of Switching Functions", IEEE TC, Vol. C-19, #3, March 1970, pp. 237.

30. Short, R.A., "A Theory of Relations Between Sequential and Combinational Realizations of Switching Functions", Stanford Electronic Laboratories, Menlo Park, California, Tech. Rep. #098-1, 1966.

31. Svoboda, A., "An Algorithm for Solving Boolean Equations", IEEE TC, Vol. EC-12, October 1963, pp. 557.

32. Svoboda, A., "Logical Instruments for Teaching Logic Design", IEEE Trans. on Education, Vol. E-12, #4, September 1969, pp. 262.

33. Svoboda, A., "Ordering of Implicants", IEEE TC, February 1967.

34. Thayse, A., Davio, A., "Boolean Differential Calculus and Its Application to Switching Theory", IEEE TC, Vol. C-22, #4, April 1973, pp. 409.

35. Ulrich, J.W., "A Note on the Solution of Sequential Boolean Equations", IEEE TC, February 1967.

36. Ulug, M.E., Bowen, B.A., "A Unified Theory of the Algebraic Topological Methods for the Synthesis of Switching Systems", IEEE TC, Vol. C-23, #3, March 1974, pp. 255

37. Special Issue on Fault-Tolerant Computing; IEEE TC, Vol. C-23, #7, July 1974.

38. Special Issue on Fault-Tolerant Computing; IEEE TC, Vol. C-20, November 1971.

39. Svoboda, A., "Boolean Analyzer", Proc. of the IFIP Congress, Edinburgh, Scotland, August 1968, Booklet D, North Holland Publ, Co., 1968.

40. Curtis, H.A., A New Approach to the Design of Switching Circuits, Van Nostrand Reinhold, New York, 1962.

41. Shannon, C.E., "A Symbolic Analysis of Relay and Switching Circuits", Trans. AIEE 57, 1938.