1-1-1984

A study of switching function representations.

Suman Purwar

A STUDY OF SWITCHING FUNCTION REPRESENTATIONS

BY

SUMAN PURWAR

A THESIS

PRESENTED TO THE GRADUATE COMMITTEE

OF LEHIGH UNIVERSITY

IN CANDIDACY FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

LEHIGH UNIVERSITY

1984

A STUDY OF SWITCHING FUNCTION REPRESENTATIONS

BY

SUMAN PURWAR

A THESIS

PRESENTED TO THE GRADUATE COMMITTEE

OF LEHIGH UNIVERSITY

IN CANDIDACY FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

LEHIGH UNIVERSITY

1984

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science

3/29/84
(date)

Professor in Charge

Chairman of Department

ACKNOWLEDGMENT

The valuable guidance and suggestions given by Prof. A. K. Susskind, the Major Professor, are acknowledged. His encouragement and the many discussions were of invaluable assistance to me.

Finally, I wish to thank my husband for his understanding, encouragement and patience.

Table of Contents

List of Figures

ABSTRACT

The Walsh-Rademacher spectral transform is a transformation procedure for converting a Boolean function into a set of spectral coefficients. It is shown how certain problems in the Boolean domain are soluble in the spectral domain. An algorithm is presented to compute the Walsh spectrum directly from the Boolean expression. Some properties of the Walsh spectrum are investigated. The Binary Decision Diagram (BDD) is used to compute the Walsh spectrum and, conversely, from the complete Walsh spectrum, the BDD is constructed. An algorithm is presented to obtain the near optimal BDD of a Boolean function via Decision Tables.

1. INTRODUCTION

1.1 INTRODUCTION

The Boolean equation, truth table and Karnaugh map are commonly used to express the logical input-output relationship of a binary function - all of which have the disadvantage of growing rapidly in size with the increasing number of variables involved(Actually, we will be concerned here with the special case of bivalued variables and functions and so we will deal with a special case of Boolean algebra often called Switching algebra). Binary Decision Diagrams offer a concise way to specify the precise logical performance of a Boolean function [1, 2, 20]. The diagrams are essentially means to compute the output value of a function by examining the input values and can be used to determine the various properties of the functions they represent. These diagrams have been used for test generation and for obtaining various implementations of binary functions [1, 2, 13, 20].

To dervive these diagrams from the Boolean expression, repeated application of the classical expansion due to Shannon is used as follows[1] :

$$F(X_1,X_2,\dots,X_n) = X_1F_1 + \overline{X}_1F_0 \qquad (1.1)$$

where, $F_1 = F(1,X_2,\dots,X_n)$ and $F_0 = F(0,X_2,\dots,X_n)$. We begin by setting X_1=0 in F(X) to obtain the function F_0 and do the same for

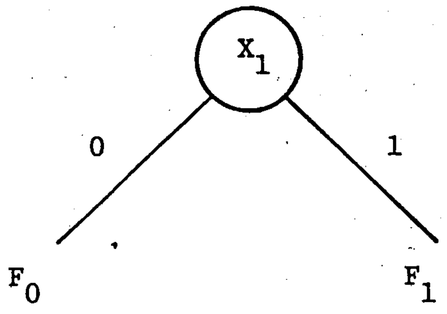


Figure 1-1. BDD OF F(X)

X_1=1 to obtain F_1 as shown in Fig.1-1.

Now the process is repeated for the variable X_2, and so on. In its most general form, all variables are represented as nodes on the diagram of which only one branch is active for any particular set of inputs. The output value is computed by entering the diagram at the root and then traversing downward through the diagram. There exists only one path through the diagram for any set of input values. All paths in the diagram terminate in F=1 or F=0.

Boolean functions are binary valued, conventionally taken as 0 and 1. However instead of involving two binary values, it is possible to transform this binary data into another mathematical domain, in which the resultant numbers lie within a larger range of values, not confined to (0,1). This is the "spectral domain", the data enumerated in this domain being the spectrum of the given binary function [14]. A close mathematical analogy is the transform of a complex a.c. waveform, whose magnitude varies with time, from the time domain to the frequency domain giving the frequency components or spectrum of the a.c. waveform.

The most relevant and compact digital transform is the Walsh-Rademacher transform [3, 4, 6, 7, 8, 9, 14, 15]. This transform is orthogonal, involving the numbers -1 and +1. The fundamental property of this transform is that no information content is lost in the transformation into the spectral domain. Thus a transformation back from the spectral domain to the more familiar binary domain is possible, by using the inverse of the forward transform. Mathematically we have :

$$[C] = [T].[F] \tag{1.2}$$

where [T] is the square transform matrix of dimension $2^n x 2^n$; [F] is the column matrix of dimension $2^n x1$, representing the output of the function F(X) of n variables. [C] is the column matrix of dimension $2^n x1$ corresponding to the spectrum of F(X).

The converse of Eq(1.2) is :

$$[F] = [T]^{-1}.[C] \tag{1.3}$$

where $[T]^{-1}$ is the inverse of the forward transform matrix [T].

For the purpose of the above mentioned transformation, we will code binary values (0,1) to (-1,+1). Under this coding the jth element of [F] is given by

$$F_j = 2.F(X_1,X_2,\ldots,X_n) - 1 \tag{1.4}$$

with $(X_1,X_2,\ldots,X_n)$ being the jth n-tuple(out of 2^n). The transform matrix [T] consists of 2^n row vectors,R,each of dimension 2^n, computed as follows [8].

$$R_0 = [1,1,\ldots 1]_{1x2^n}$$

$R_i = [2(X_i)_1-1, 2(X_i)_2-1, \ldots, 2(X_i)_{2^n}-1]_{1x2^n}$

with $i = 1$ to n and $(X_i)_j$ being the value of the variable X_i in the j^{th} n-tuple.

$R_{ik} = [(r_{i1}xr_{k1}), \ldots, (r_{i2^n}xr_{k2^n})]$

with $i,k = 1$ to n ; $i \neq k$; and r_{ij} ,r_{kj} being the j^{th} elements of R_i and R_k respectively.

.
.
.

$R_{12\ldots n} = [(r_{11}x\ldots r_{n1}), \ldots, (r_{12^n}x\ldots xr_{n2^n})]$

It is noted that the transform matrix is an orthogonal matrix, is independent of the Boolean function F(X), and only depends upon the number of variables in F(X). The ordering of the rows in [T] can be changed, as long as the ordering in the vector [C] is changed accordingly. The ordering of [F] and rows in [T] comes from the ordering of the n-tuples and again, one can change the sequence of n-tuples without affecting the end results(see [11]).

If the Walsh-Rademacher ordering(see [8]) is used and (0,1) is coded in (-1,+1), [T] can also be derived from the following recursive form

$$T_n = \begin{bmatrix} T_{n-1} & T_{n-1} \\ & \\ -T_{n-1} & T_{n-1} \end{bmatrix} \qquad (1.5)$$

where T_n is a $2^n x 2^n$ transform matrix, n is the number of variables, and $T_0 = 1$.

The Walsh Coefficients are denoted by $C_0, C_1, C_2, C_3, \ldots C_{12\ldots n}$. In the following chapters $C_{12\ldots n}$ will be denoted by C_{ALL} because it involves all the variables.The Walsh Coefficients have been used for fault detection [18, 16] and for synthesis [12, 19, 10] of the logic networks. In the next section we will discuss some properties of the Walsh coefficients.

1.2 SOME PROPERTIES OF WALSH COEFFICIENTS

With the introduction to the Walsh transformation, we now turn to some of their properties :

1. The arithmetic sum of all the (2^n) Walsh coefficients is given by

$$\Sigma C = 2^n . F_n \qquad (1.6)$$

Where F_n represents the last entry in the column vector [F] and corresponds to the mapped value of the function F(X) with all its inputs at 1. Thus,

$$\Sigma C = +2^n \quad \text{when } F_n = 1 \text{ and}$$
$$\Sigma C = -2^n \quad \text{when } F_n = -1.$$

PROOF

Premultiplying Eq(1.2) by a vector of all 1's of dimension $1x2^n$, we have

$[1\ 1\ \ldots\ 1]\ [C] = [1\ 1\ \ldots\ 1]\ [T]\ [F]$

or, $\sum C = [0\ 0\ \ldots\ 0\ 2^n]\ [F]$

or, $\sum C = 2^n\ F_n$

where F_n is the last element of the column vector [F], which corresponds to the mapped value of the function F(X) with all its inputs at 1. F_n can be either +1 or -1, so we have

$$\sum C = +2^n \text{ when } F_n = +1 \text{ and}$$
$$\sum C = -2^n \text{ when } F_n -1.$$

EXAMPLE 1

Consider a simple case of 2 variables. Eq(1.2) can be written as

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_{12} \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & +1 & -1 & +1 \\ +1 & -1 & -1 & +1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{1.7}$$

Premultipling on both sides by [1 1 1 1], we have

$$[1\ 1\ 1\ 1]\begin{bmatrix}C_0\\C_1\\C_2\\C_{12}\end{bmatrix}=[1\ 1\ 1\ 1]\begin{bmatrix}+1&+1&+1&+1\\-1&-1&+1&+1\\-1&+1&-1&+1\\+1&-1&-1&+1\end{bmatrix}\begin{bmatrix}F_1\\F_2\\F_3\\F_4\end{bmatrix}$$

$$C_0+C_1+C_2+C_{12}=[0\ 0\ 0\ 4]\begin{bmatrix}F_1\\F_2\\F_3\\F_4\end{bmatrix}$$

or, $\Sigma C = 2^2.F_4$

From Eq(1.6), some properties can be derived as follows :

a. All 2^n coefficients with zero values do not define a valid Boolean function.

b. If one of the coefficients is $+2^n$, then the remaining coefficients will all be zero.

2. The length of the vector [C] (square root of the sum of the squares of all the coefficients) is given by

$$\sqrt{\Sigma C^2} = 2^n \qquad (1.8)$$

<u>PROOF</u>

Since [T] is an orthogonal matrix, we can write(see [5])

$$[T]'.[T] = \underline{K}.[I] \qquad (1.9)$$

where [T]' is the transpose of the matrix [T], [I] is the identity matrix and K is a constant. In our case $K=2^n$. So, we have

$$[T]'.[T] = 2^n.[I] \tag{1.10}$$

Taking the transpose on both sides of Eq (1.2), we have

$$[C]' = [F]'.[T]' \tag{1.11}$$

Multiplying Eq(1.11) on both sides by [C], we have

$$[C]'.[C] = [F]'.[T]'.[T].[F]$$

using Eq(1.10), we have

$$\begin{aligned}[C]'.[C] &= [F]'.2^n.[I].[F] \\ &= 2^n.[F]'.[F]\end{aligned}$$

Since [C] and [F] are column matrices, we can write

$$\Sigma C^2 = 2^n.\ \Sigma F^2 \tag{1.12}$$

The dimension of [F] is 2^nx1 and the F's can be either +1 or -1, resulting in $F_i^2 = 1$. So there would be 2^n 1's in ΣF^2. Hence $\Sigma F^2 = 2^n$. Using this property in Eq(1.12), we have

$$\Sigma C^2 = 2^{2n}$$

$$\text{or, } \sqrt{\Sigma C^2} = 2^n$$

EXAMPLE 2

Consider the two variables case of EXAMPLE 1. Transposing Eq(1.7) on both sides, we have

$$[C_0\ C_1\ C_2\ C_{12}]=[F_1\ F_2\ F_3\ F_4]\begin{bmatrix}+1 & -1 & -1 & +1\\ +1 & -1 & +1 & -1\\ +1 & +1 & -1 & -1\\ +1 & +1 & +1 & +1\end{bmatrix}$$

Postmultiplying both sides by [C], and using $[T]'[T] = 2^2$, we have

$$[C_0\ C_1\ C_2\ C_{12}]\begin{bmatrix}C_0\\ C_1\\ C_2\\ C_{12}\end{bmatrix}= 4.[F_1\ F_2\ F_3\ F_4]\begin{bmatrix}F_1\\ F_2\\ F_3\\ F_4\end{bmatrix}$$

$$C_0^2+C_1^2+C_2^2+C_{12}^2 = 4(F_1^2+F_2^2+F_3^2+F_4^2)$$

Since $F_i^2 = 1$, for i=1 to 4, we finally have

$$\sqrt{C_0^2+C_1^2+C_2^2+C_{12}^2} = 2^2.$$

1.3 SUMMARY AND OUTLINE OF REPORT

In this chapter, we discussed some properties of the Walsh coefficients with a brief introduction to the Walsh transformation and the Binary Decision Diagram.

In Chapter 2, a simple approach to obtain a near-optimal BDD using Decision Tables is explored. Simple examples that illustrate the algorithm are discussed.

Chapter 3 presents algorithms to compute the Walsh coefficients of a function from its BDD and, conversely, construct the BDD of a function from its Walsh spectrum. Examples are given to illustrate both algorithms.

Chapter 4 presents an algorithm to compute the Walsh coefficients directly from the Boolean function F(X). The Walsh spectrum of the Boolean Difference, $(dF(X)/dX_i)$ of F(X), is computed from the spectrum of F(X). The spectra of $[X_i(dF(X)/dX_i)]$ and $[\bar{X}_i(dF(X)/dX_i)]$ are then computed to determine the true vertices of these two functions, as these vertices determine the probability of detecting a fault at any input lead.

Chapter 5 presents a summary of the results of the thesis. A list of references is included at the end of the thesis.

2. A METHOD FOR FINDING GOOD BDD'S

This chapter presents an algorithm for obtaining a "good" BDD of a Boolean function. A BDD is considered to be a good BDD when it has a small, possibly minimum, number of nodes.

Construction of the BDD of a function $F(X_1,\ldots,X_i,\ldots,X_j,\ldots,X_n)$ beginning with a variable X_i may lead to a BDD with a smaller number of nodes than that with some other initial choice X_j. Thus, the number of nodes in a BDD depends upon the order in which the variables are selected. The proper selection of the variables at every step will lead to a good BDD.

In the context of data processing, a similar problem of reducing computation time, based on a set of rules and conditions, was treated by Pollack [17]. Pollack solved this problem by forming a table with conditions in the rows and rules in the columns. He called this table a DECISION TABLE. The algorithm discussed in this chapter is based on Pollack's approach.

We begin with some definitions in Section 2.1. This is followed by the formulation of the algorithm in Section 2.2. The algorithm is illustrated with examples. This algorithm does not always lead to a good BDD and an example illustrating this limitation is also presented. Finally, a brief summary is presented in Section 2.3.

2.1 DEFINITIONS

Table

The Boolean function is first represented in the form of a table. Each product implicant in the function is represented by a column in the table. All the variables appear in the leftmost column as row coordinates. In every column, the entry 1(0), represents an uncomplemented(complemented), variable in that implicant. A dash in a column shows that the implicant is independent of that variable. For example, the table for the function $F=X_1X_2+X_2X_3+X_1X_3$ is shown in Fig2-1.

Subtable

For each variable X_i, there are two tables corresponding to its 0 and 1 branches. These tables are called subtables. The 0 branch subtable corresponds to the part of the table where X_i has 0 and dash entries whereas the 1 branch subtable corresponds to the part of the table where X_i has 1 and dash entries.

Implicant Count

The Implicant Count of an implicant is 2^r, where r is the number of dashes(variables missing) in that implicant. In Fig2-1, the Implicant Count for all three implicants is 2, because every implicant has only one dash.

X_1	1	-	1
X_2	1	1	-
X_3	-	1	1

Figure 2-1. Table of $F = X_1X_2 + X_2X_3 + X_1X_3$

Dash Count

For each variable X_i, the sum of the Implicant Counts taken over those implicants that are independent of X_i, is denoted as Dash Count. In Fig2-1, the Dash Count for X_1, X_2 and X_3 is 2.

1(0) COUNT

The 1(0) Count of a variable X_i is the sum of the Implicant Count taken over those implicants where X_i is uncomplemented (complemented). In Fig2-1, the 1 count for X_1, X_2 and X_3 is 4 and the 0 Count for all variables is 0.

DELTA

For any variable X_i, Delta is the absolute difference of the 1 count and the 0 count. In Fig2-1, Delta for X_1, X_2 and X_3 is 4.

2.2 ALGORITHM

The algorithm to determine the next node in the BDD consists of the following steps :

1. Express the Boolean function in the form of a table as defined above.
2. Combine any two implicants that can be merged. This is similar to Boolean minimization.
3. Select that variable which has a minimum dash count.
4. If two or more variables have the same minimum dash count, select the variable with minimum delta.

5. If two or more variables have the same minimum delta and the subtables corresponding to one variable is the same as some other subtable or part of the subtable , then select that variable after selecting other variables because equivalent subtables then can be combined.
6. Assign the variable so chosen to the node of the path being considered. If this is the first variable of the BDD being constructed, assign it to the root. This node or root has two branches, each of which leads to a subtable with one less variable than the original table.
7. At any stage if two subtables are equivalent, i.e., the variables and the entries are the same, then join them.
8. To simplify the subtables, the following rules can be applied :

   a. If a branch leads to a subtable containing all dashes in any implicant, terminate that branch with F=1.

   b. If a branch does not lead to a subtable, terminate that branch with F=0.

   c. If at any step a variable contains only dashes in all rows of a subtable, eliminate that variable from the subtable.

   d. For any variable X_i, the subtable containing only one variable, having the form $\begin{array}{|c|c|}\hline X_i & 0 \\ \hline\end{array}$ will be replaced by the node X_i having two branches. The 0 branch will be terminated with F=1 and the 1 branch will be terminated with F=0. Similarly, $\boxed{X_i \mid 1}$ will be replaced by the node X_i with its 0 branch terminated with F=0 and 1 branch terminated with F=1.

9. If a branch leads to a subtable containing more than one implicant, go back to step(2).

<u>EXAMPLE</u>

Consider the function $F=B(\bar{A}C+\bar{C}\bar{E})+\bar{E}(\bar{A}B+\bar{B}D)$. The BDD for this function, arbitrarily choosing the variables in lexicographical order, is shown in Fig.2-2. We can express this function in a table form as follows :

					Dash count
A	0	-	0	-	8
B	1	1	1	0	0
C	1	0	-	-	8
D	-	-	-	1	12
E	-	0	0	0	4
Implicant Count	4	4	4	4	

According to step(3), B would be the root, since it has the minimum dash count. Applying step(6), we have Fig.2-3.

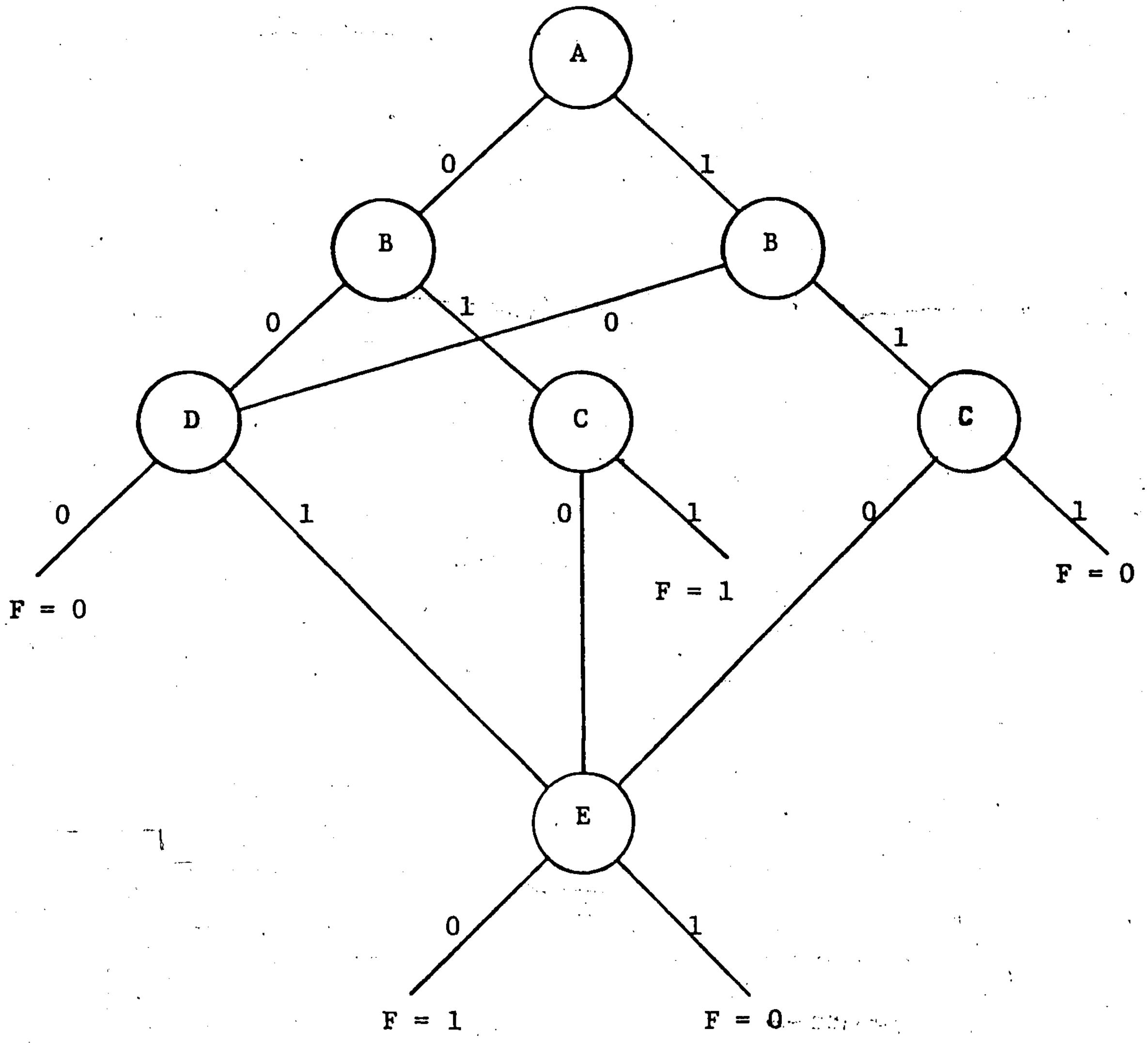


Figure 2-2.

In the subtable of the "0" branch of B(Fig.2-3), A and C conditions contain only dashes and D condition in the subtable of the "1" branch contains only dashes. Thus, the corresponding rows are eliminated and the new tables are shown in Fig.2-4.

In the right subtable of Fig.2-4, applying step(4), C would be our next condition to be tested. In the left subtable, since both conditions have the same delta, condition D would be tested before E, because E appears in the right subtable with the same value. Now we have Fig.2-5.

Since the second column is redundant in both subtables of the variable C, it is eliminated in Fig.2-6, according to step(2).

Finally, joining equivalent subtables and applying step 8(d), we have Fig.2-7.

The number of nodes in Fig.2-2 is 7 whereas the number of nodes in Fig.2-7 is 5, which is minimal in this case, since each variable appears exactly once.

COUNTER EXAMPLE

Consider the function $F=\bar{B}\cdot\bar{D}+\overline{A}\overline{D}+A\overline{B}\overline{C}$. We can express this function in a table form as follows:

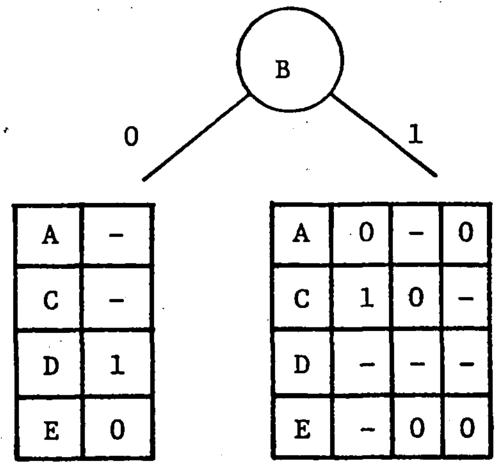
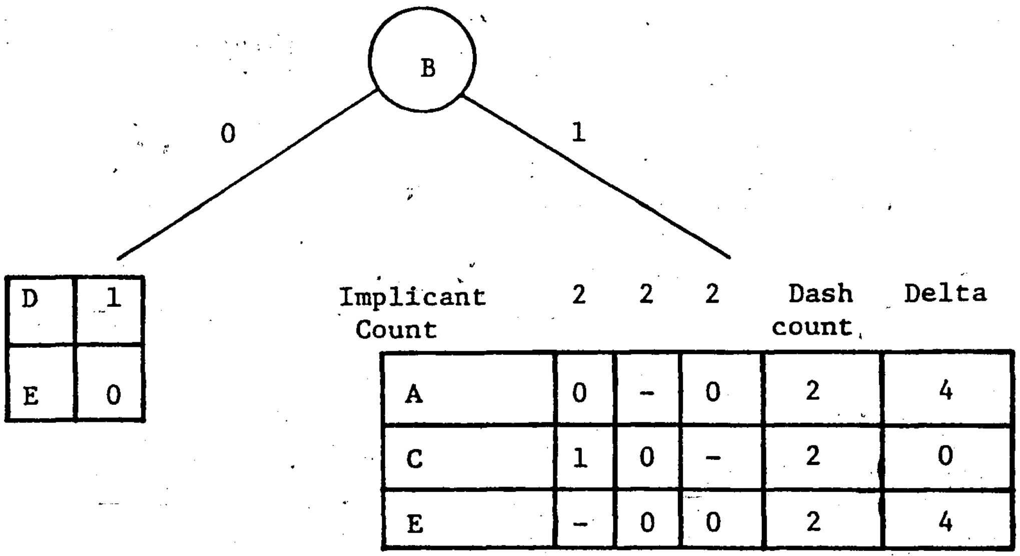


Figure 2-3.



Implicant Count	2	2	2	Dash count	Delta
A	0	-	0	2	4
C	1	0	-	2	0
E	-	0	0	2	4

Figure 2-4.

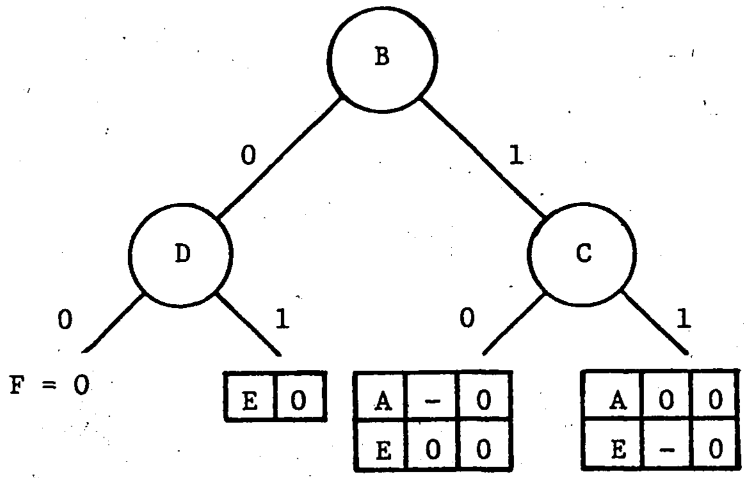


Figure 2-5.

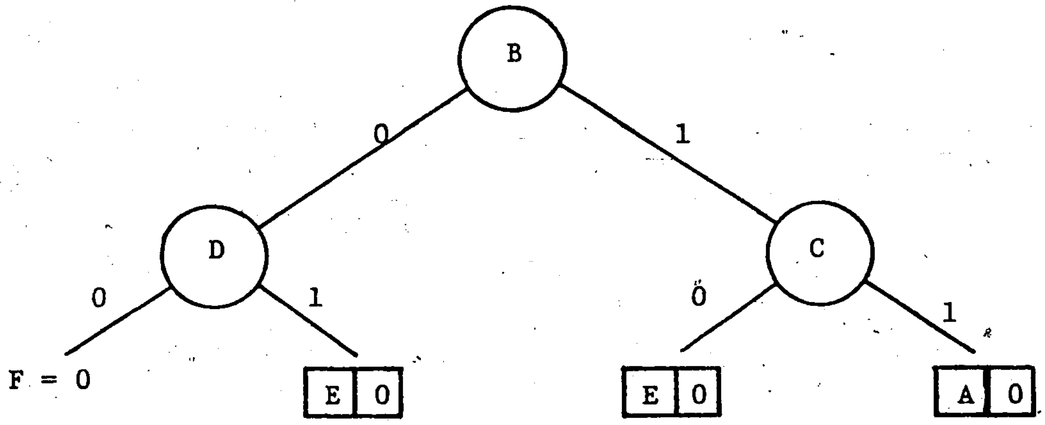


Figure 2-6.

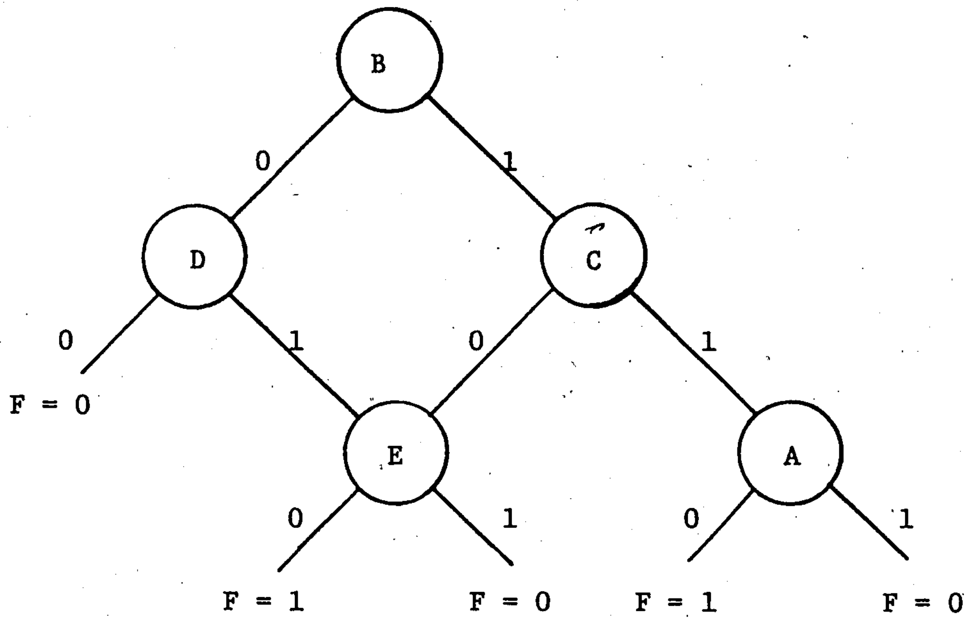


Figure 2-7.

				Dash count
A	-	0	1	4
B	0	-	0	4
C	-	-	0	8
D	0	0	-	2
Implicant Count	4	4	2	

According to step(3), variable D would be our root, since it has the minimum dash count. Applying step(6), we have Fig.2-8. In the subtable of the zero branch, the last column is redundant. So simplifying this table we have Fig2-9. Since variable B appears in both subtables with the same entry, we can choose either A or C in the 1 branch subtable. Choosing C in the right subtable of Fig2-9, we get Fig2-10.

Variable B appears in the two subtables with the same values, so according to step(5), it would be tested after testing A variable(Fig2-11). Finally, combining subtables, we get Fig2-12.

However, if we had chosen A variable as a root, we would have obtained Fig2-13.

Fig2-12 has 5 nodes and Fig2-13 has 4 nodes. Thus, we can conclude that our algorithm does not always lead to a minimum

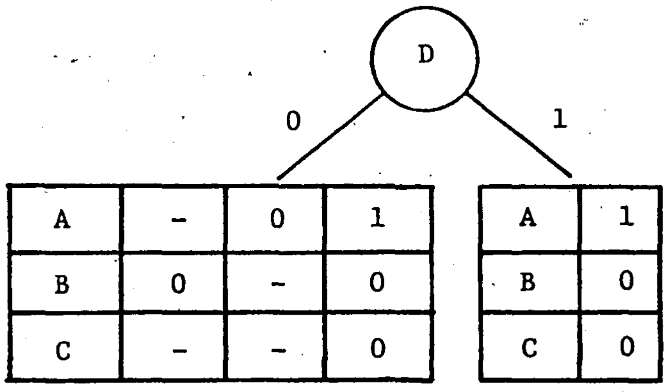


Figure 2-8.

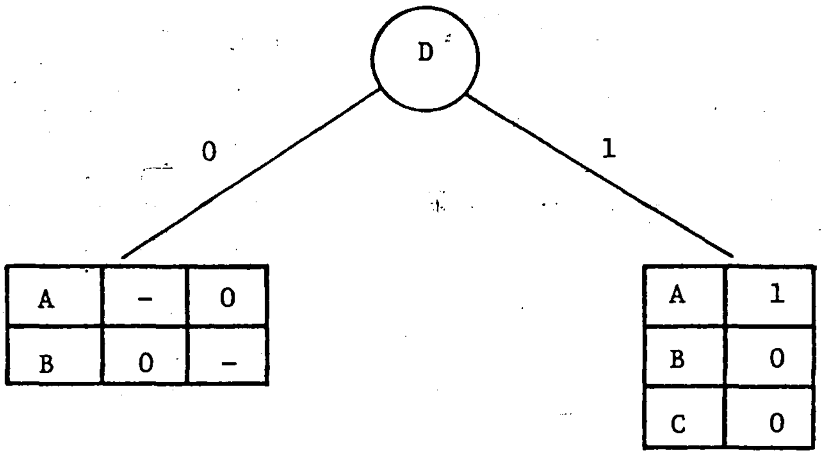


Figure 2-9.

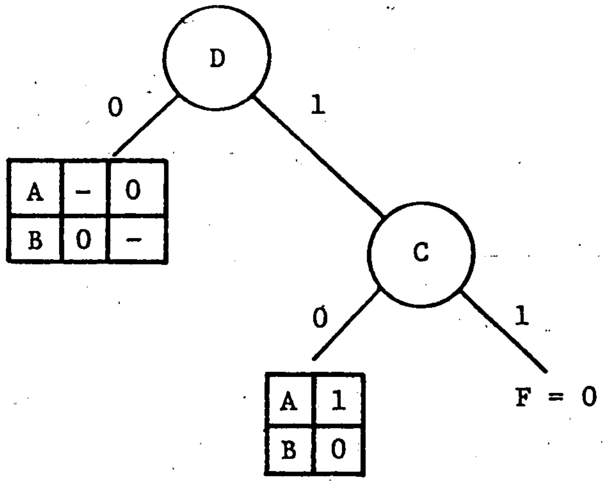


Figure 2-10.

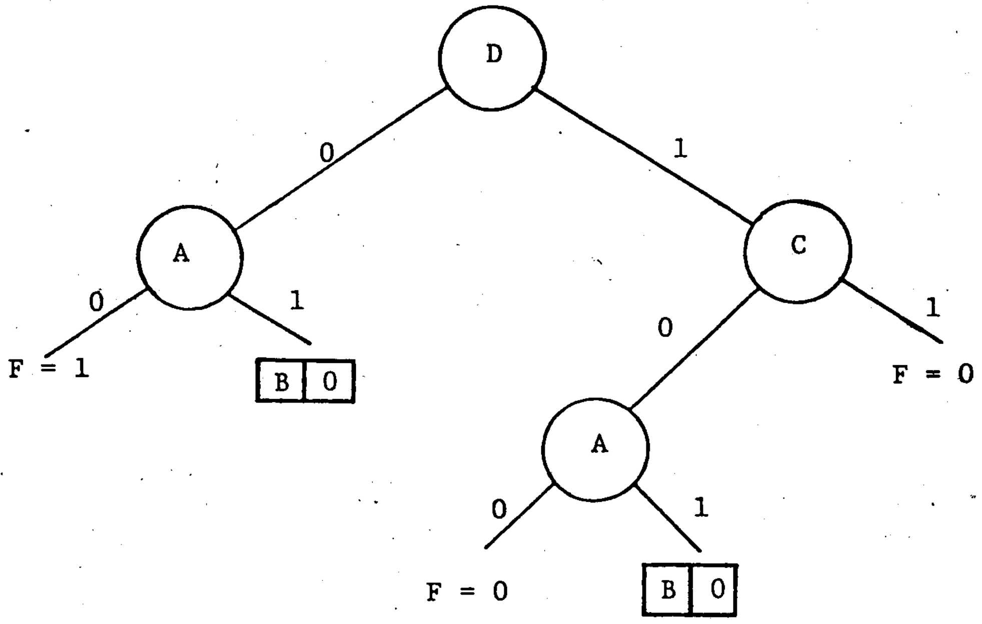


Figure 2-11.

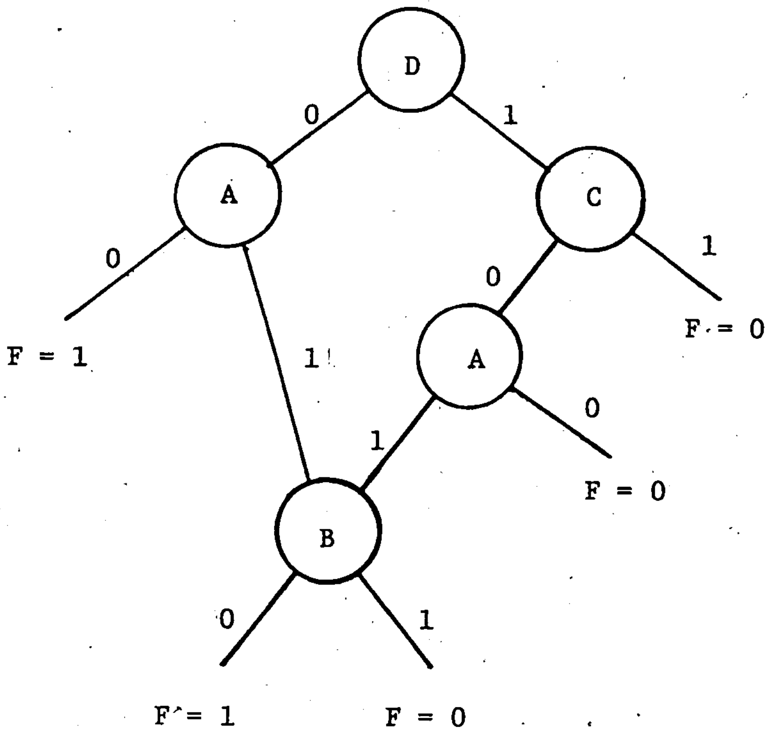


Figure 2-12.

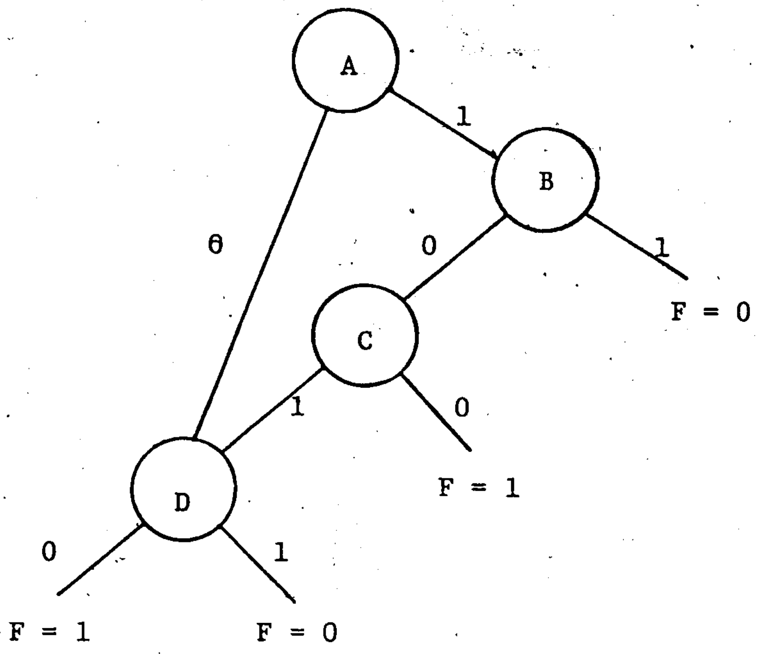


Figure 2-13.

structure, just good ones.

2.3 SUMMARY

In this chapter, we presented an algorithm to obtain good BDD's. The algorithm presented can be easily computer programmed.

From experience, we found that if the function consists of a minimal subset of implicants, then the algorithm usually gives us a good BDD. This means that to get a good BDD, we should first minimize the function and then apply the algorithm. This adds an extra step.

3. WALSH SPECTRUM FROM BDD AND BDD FROM WALSH SPECTRUM

This chapter presents two algorithms. The first algorithm describes the computation of the complete set of 2^n Walsh coefficients C of a function F(X) from its BDD. The second algorithm describes the construction of the BDD of a function F(X) from its complete set of 2^n Walsh coefficients C. This means that we can go from one representation to the other and illustrates that the properties of a function which can be determined from the BDD can also be derived from the Walsh spectrum.

3.1 COMPUTATION OF WALSH COEFFICIENTS FROM BDD

In this section, we describe the computation of the complete set of 2^n Walsh coefficients for a switching function F(X) of n variables directly from its BDD. An example of a BDD is shown in Fig.3-1 for the function $F = X_1X_2 + X_2X_3 + X_1X_3$. Branches are labeled as a,b,c,...h along with the branch value (0 or 1). For example; a;0 means that branch a has a value 0. An "exit branch" is defined as a branch that terminates in a value (0 or 1) of the function F. Exit branches in Fig.3-1 are c,f,g,h. Both outgoing branches of an "exit node" are exit branches. In Fig.3-1, X_3 is an exit node. The "Root" is a node with no incoming branches. In the example being considered, X_1 is a root.

The total number of input combinations (ICs) for a function of

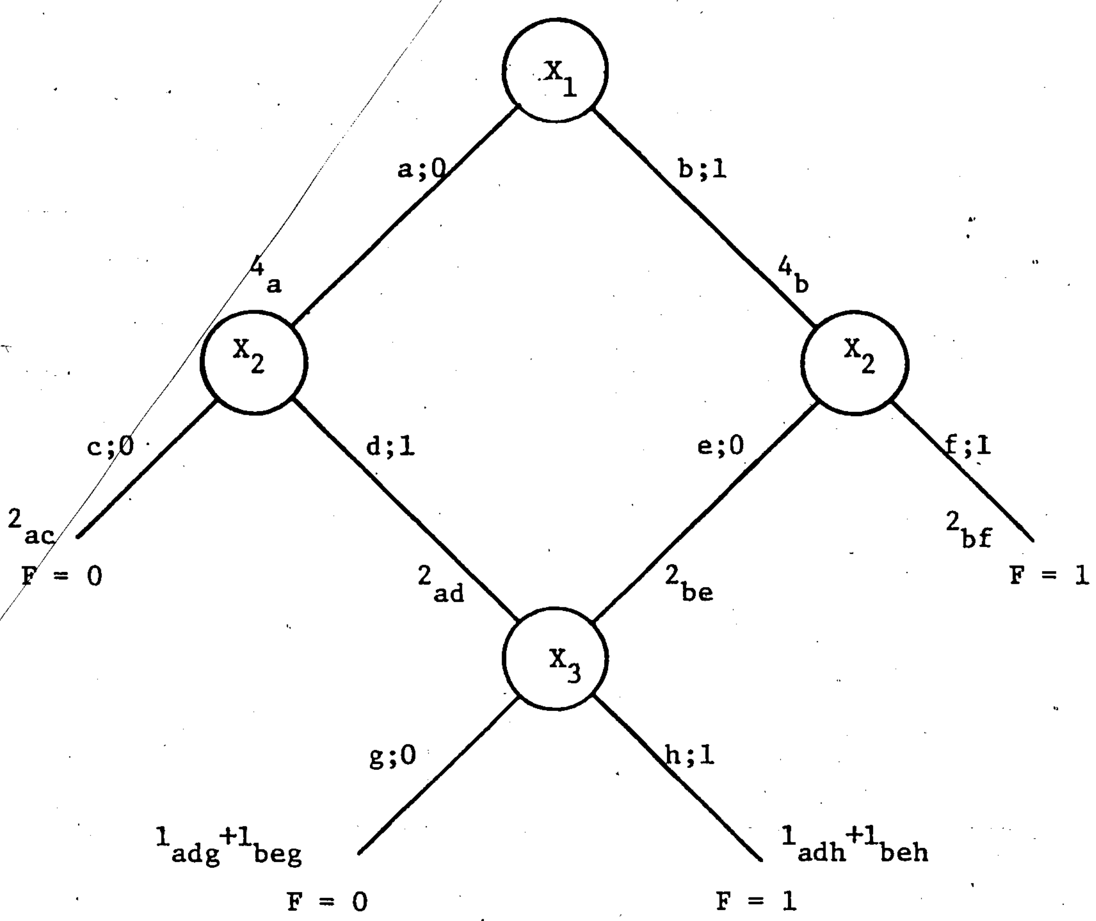


Figure 3-1. BDD OF $F = X_1X_2 + X_2X_3 + X_1X_3$

n variables is 2^n. For the example of Fig.3-1, it is $2^3=8$. The number of ICs associated with each of the outgoing branches of the root is 2^{n-1}. In Fig.3-1, ICs corresponding to the branches a and b are labeled as 4_a and 4_b, respectively. The outgoing branches of any node with only one incoming branch are associated with ICs equal to one half of the ICs of the incoming branch. In the present example, ICs of the outgoing branches c and d of the node X_2 are labeled as 2_{ac} and 2_{ad}, respectively. If any node has more than one incoming branch, the number of the ICs of each of the outgoing branch is equal to the sum of one half of the number of ICs corresponding to each incoming branch. In the present example, this case is illustrated by the ICs associated with the branch g labeled as $1_{adg}+1_{beg}$, indicating that 1_{adg} comes from the branch d and 1_{beg} comes from the branch e. The subscripts attached to particular ICs are indicative of the path of these ICs from root to the particular branch. Thus ac, for example, is indicative of the two ICs that have X_1 and X_2 equal to zero, i.e., $\bar{X}_1\bar{X}_2\bar{X}_3$ and $\bar{X}_1\bar{X}_2X_3$.

In order to facilitate the computation of Walsh coefficients from the BDD, we define a function η_{path}(branch values, F) of a path between any node and the terminal value of the switching function, F, which depends upon some or all branch values selected along the path. The value of η is defined to be +1(-1) if the branch values along the specified path and the terminal value of the function F contains an even(odd) number of zeroes. For example, in Fig.3-1, $\eta_{ac}(a,c,F) = -1$, but $\eta_{ac}(a,F) = 1$. A set of subscripts (attached to the ICs of a particular exit branch) is "complete" if it contains ~~one outgoing~~ branch from each of the nodes associated with the m^{th} order Walsh Coefficient being computed. For example, in Fig.3.1, consider the computation of C_{23}. The corresponding nodes to be considered are X_2 and X_3. The outgoing branches from these nodes are c,d,e,f,g and h. The subscripts attached to the ICs 1_{adg} of the exit branch g are "complete" because it contains the branch d corresponding to the node X_2 and the branch g corresponding to the node X_3. Likewise, the subscripts of ICs 1_{beg}, 1_{adh} and 1_{beh} are also "complete", whereas the ICs 2_{ac} and 2_{bf} are not "complete".

3.1.1 COMPUTATION OF C_0

The zeroth order Walsh Coefficients, C_0, is given by

$$C_0 = \Sigma M_i - \Sigma M_j \tag{3.1}$$

where, ΣM_i is the sum of the number of ICs associated with those exit branches whose terminal value is F=1 and ΣM_j is the sum of the ICs associated with those exit branches whose terminal value is F=0. Coefficient C_0, in fact is the difference between true vertices(F=1) and false vertices (F=0). It gives us a measure of the number of minterms associated with a function.

3.1.2 COMPUTATION OF OTHER SPECTRAL COEFFICIENTS

In order to compute the remaining 2^n-1 Walsh Coefficients corresponding to $m \leqslant n$ variables, the following rules are observed :

1. We begin by considering the ICs at all the exit branches. Remembering that the subscripts of a particular IC denote the path of the IC from the root node, we consider only those ICs(at the exit branches) that contain a "complete" set of subscripts associated with the m^{th} order Walsh coefficients.
2. From the collection of ICs defined in step(1), delete ICs associated with outgoing branches of exit nodes. (This step is used because the contributions along the two exit branches will cancel each other).
3. The m^{th} order Walsh Coefficient corresponding to m variables is given by :

$$C_{12...m} = \Sigma M_i \eta(\text{"complete" set , F}) \qquad (3.2)$$

where M_i is the number of ICs having a "complete" set of subscripts at a particular exit branch and η("complete" set , F) includes the branch values corresponding to the "complete" set and the terminal value of the function F.

The contributions of ICs at an exit branch that does not involve all the m variables will cancel each other. Any function for which any BDD does not contain any exit IC involving all the m variables will be characterized by $C_{12...m} = 0$.

EXAMPLE 1

With reference to Fig3-1, we wish to calculate the Walsh Coefficient C_1 corresponding to variable X_1. The outgoing branches associated with node X_1 are a and b. Branches a and b appear in the ICs associated with exit branches c,f,g,h. But, step(2) eliminates ICs at exit branches g and h. Applying Eq(3.2), we have

$$\begin{aligned} C_1 &= 2\eta_{ac}(a,F) + 2\eta_{bf}(b,F) \\ &= 2\eta_{ac}(0,0) + 2\eta_{bf}(1,1) \\ &= 2(+1) + 2(+1) \\ &= 4 \end{aligned}$$

To calculate C_2, corresponding to variable X_2, we need to consider ICs at exit branches c, f, g, h because the outgoing branches associated with X_2 are c, d, e, f. But, step(2) eliminates ICs at exit branches g and h. Thus, we can write

$$\begin{aligned} C_2 &= 2\eta_{ac}(c,F) + 2\eta_{bf}(f,F) \\ &= 2\eta_{ac}(0,0) + 2\eta_{bf}(1,1) \\ &= 2(+1) + 2(+1) \\ &= 4 \end{aligned}$$

Similarly C_3 can be calculated below :

$$C_3 = 1\eta_{adg}(g,F)+1\eta_{beg}(g,F)+1\eta_{adh}(h,F)+1\eta_{beh}(h,F)$$
$$= 1\eta_{adg}(0,0)+1\eta_{beg}(0,0)+1\eta_{adh}(1,1)+1\eta_{beh}(1,1)$$
$$=1 + 1 +1 +1$$
$$= 4$$

The second order Walsh Coefficients C_{12}, C_{23} and C_{13} corresponding to variables X_1X_2, X_2X_3 and X_1X_3 can be computed as follows :

The outgoing branches associated with variables X_1 and X_2 are (a,b) and (c,d,e,f), respectively. We need to consider ICs at exit branches c, f, g, h but, step(2) eliminates ICs at g, h.

$$C_{12} = 2\eta_{ac}(a,c,F) + 2\eta_{bf}(b,f,F)$$
$$=2\eta_{ac}(0,0,0) + 2\eta_{bf}(1,1,1)$$
$$= 2(-1) + 2(+1)$$
$$= 0$$

The outgoing branches associated with variables X_2 and X_3 are (c, d, e, f) and (g,h), respectively. The ICs at exit branches c and f will not be considered because they do not form a complete set.

$$C_{23} = 1\eta_{adg}(d,g,F)+1\eta_{beg}(e,h,F)$$
$$+1\eta_{adh}(d,h,F)+1\eta_{beh}(e,h,F)$$
$$=1\eta_{adg}(1,0,0)+1\eta_{beg}(0,0,0)$$
$$+1\eta_{adh}(1,1,1)+1\eta_{beh}(0,1,1)$$
$$=1-1+1-1$$

=0

To calculate C_{123}, the outgoing branches asoociated with X_1, X_2 and X_3 are (a,b) , (c,d,e,f) and (g,h), respectively.

$$\begin{aligned} C_{123} &= 1\eta_{adg}(a,d,g,F)+1\eta_{beg}(b,e,g,F) \\ &\quad +1\eta_{adh}(a,d,h,F)+1\eta_{beh}(b,e,h,F) \\ &=1\eta_{adg}(0,1,0,0)+1\eta_{beh}(1,0,0,0) \\ &\quad +1\eta_{adh}(0,1,1,1)+1\eta_{beh}(1,0,1,1) \\ &= -1 -1 -1 -1 \\ &= -4. \end{aligned}$$

EXAMPLE 2

With reference to Fig.3-2, we wish to calculate the Walsh Coefficient C_B. The outgoing branches associated with variable B are (c,d,e,f).The ICs at exit branches g, j, l will be considered.

$$\begin{aligned} C_B &= 4\eta_{acg}(c,F)+4\eta_{adj}(d,F)+4\eta_{beg}(e,F)+4\eta_{bfl}(f,F) \\ &= 4(+1) + 4(+1) + 4(+1) + 4(-1) \\ &= 8 \end{aligned}$$

Similarly, other coefficient are calculated below :

$$\begin{aligned} C_C &= 4\eta_{adj}(j,F) + 4\eta_{bfl}(l,F) \\ &= 4(+1) + 4(-1) \\ &= 0 \end{aligned}$$

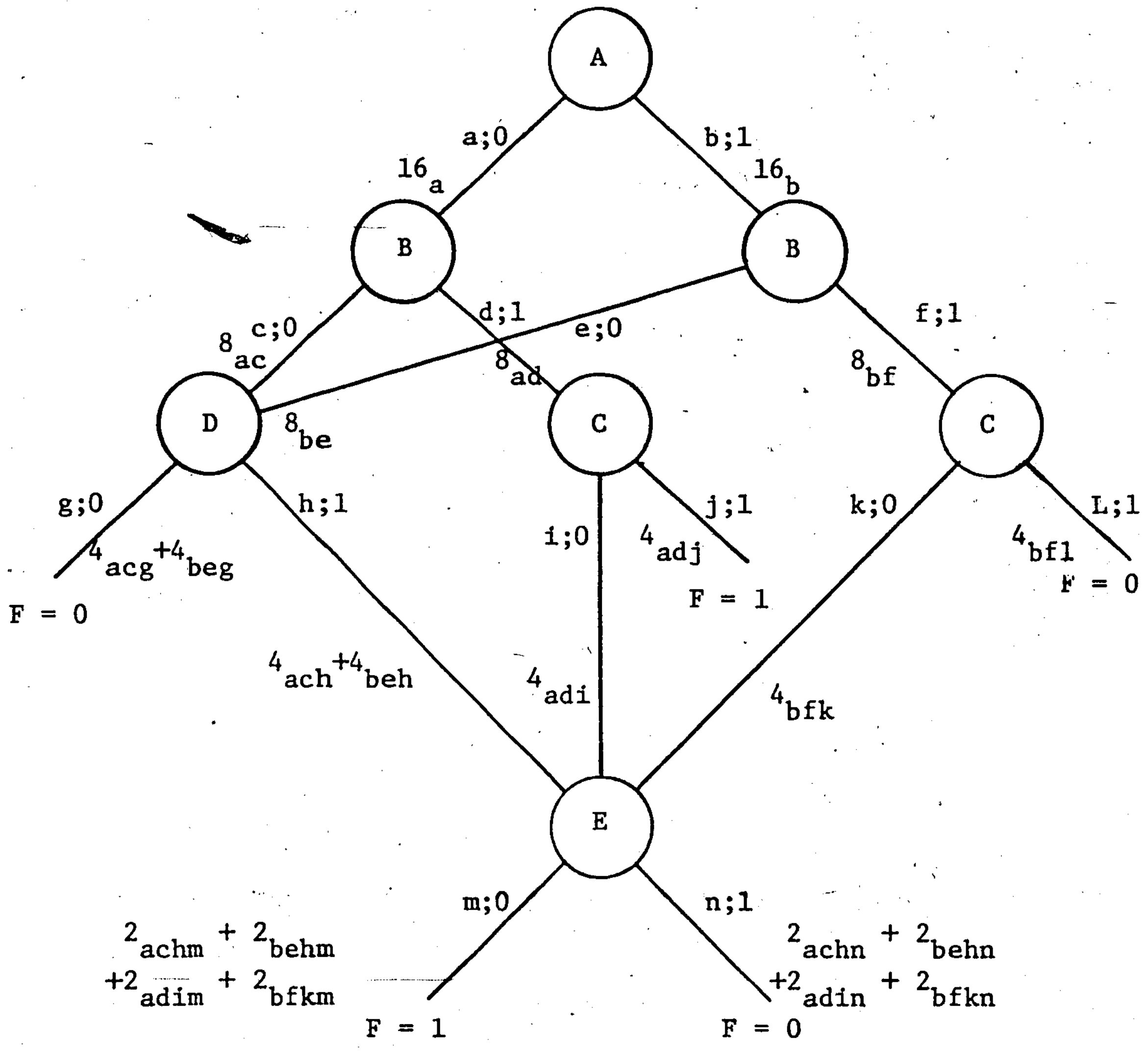


Figure 3-2. BDD OF $F=B(\bar{A}C+\bar{C}\bar{E})+\bar{E}(\bar{A}B+\bar{B}D)$

$C_D = 4\eta_{acg}(g,F) + 4\eta_{beg}(g,F)$

$= 4 + 4$

$= 8$

$C_{ABC} = 4\eta_{adj}(a,d,j,F) + 4\eta_{bfl}(b,f,l,F)$

$= 4(-1) + 4(-1)$

$= -8.$

<u>EXAMPLE 3</u>

With reference to Fig.3-3, the exit branches are d, f, g, h. Now, we wish to calculate C_C corresponding to variable C. The outgoing branches associated with node C are e and f. Only the ICs at exit branch f need to be considered.

$C_C = 2\eta_{bcf}(f,F)$

$=2\eta_{bcf}(0,1)$

$=-2$

To calculate C_{AD}, the outgoing branches associated with variables A and D are (a,b) and (g,h), respectively. Only the ICs at exit branches g and h need to be considered.

$C_{AD} = 4\eta_{ag}(a,g,F) + 1\eta_{bceg}(b,g,F)$

$+ 4\eta_{ah}(a,h,F) + 1\eta_{bceh}(b,h,F)$

$= 4 - 1 + 4 - 1$

$=6$

To compute C_{ABC}, the ICs at exit branch f only will be

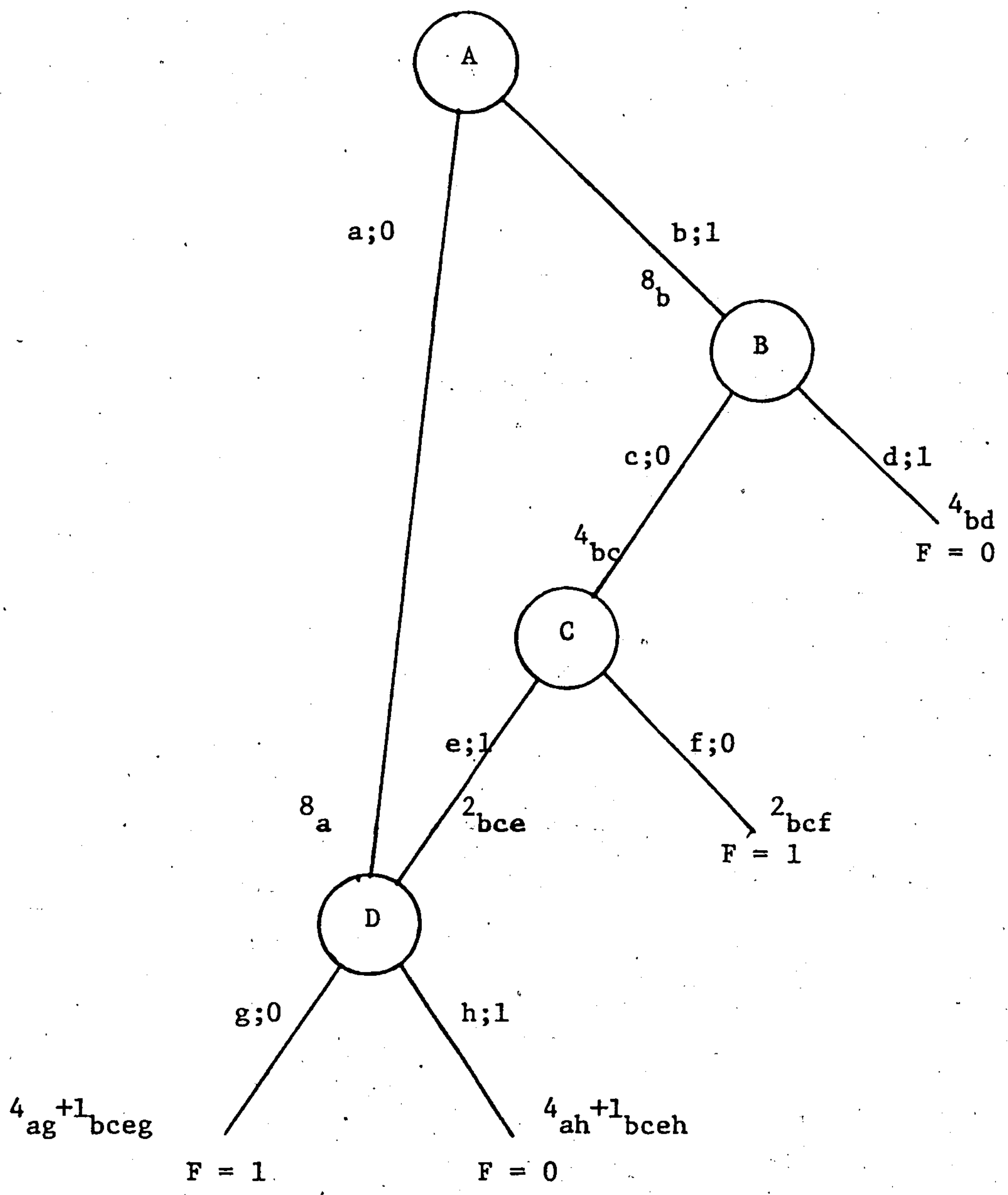


Figure 3-3. BDD OF $F = \bar{A}\bar{D} + A\bar{B}\bar{C} + A\bar{B}C\bar{D}$

considered.

$$C_{ABC} = 2n_{bcf}(b,c,f,F)$$
$$=2.$$

3.2 SYNTHESIS OF BDD DIRECTLY FROM THE WALSH SPECTRUM

In this section, we describe an algorithm to directly obtain the BDD of a Boolean function, $F(X)$, of n variables from its complete set of 2^n Walsh coefficients, C.

Consider the Shannon decomposition of $F(X)$ about a single variable X_i, i=1 to n

$$F(X_1,\ldots,X_n)=\bar{X}_iF_0 + X_iF_1 \tag{3.3}$$

where $F_0 = F(X_1,\ldots,0,\ldots,X_n)$ and $F_1 = F(X_1,\ldots,1,\ldots,X_n)$.

From Eq(1.2) and Eq(1.5), we have

$$C = \begin{bmatrix} T_{n-1} & T_{n-1} \\ & \\ -T_{n-1} & T_{n-1} \end{bmatrix} \begin{bmatrix} F_0 \\ \\ F_1 \end{bmatrix} \tag{3.4}$$

$$C = \begin{bmatrix} T_{n-1}F_0 + T_{n-1}F_1 \\ -T_{n-1}F_0 + T_{n-1}F_1 \end{bmatrix}$$

where, F_0 and F_1 are function vectors for $F_0(X)$ and $F_1(X)$, respectively. Let

$$T_{n-1}F_0 + T_{n-1}F_1 = V_0$$

$$-T_{n-1}F_0 + T_{n-1}F_1 \quad = \quad V_1$$

where, V_0 and V_1 are the ordered vectors which jointly give C. After simplifying, we have

$$T_{n-1}F_0 = (1/2)(V_0 - V_1)$$
$$T_{n-1}F_1 = (1/2)(V_0 + V_1)$$

If we know the complete spectrum and hence the two half spectra V_0 and V_1, we can calculate the spectrum corresponding to functions $F_0(X)$ and $F_1(X)$.

In general, the 2^{n-1} Walsh coefficients, C', of $F_0(X)$ are given by

$$C_j' = (1/2)(C_j - C_{ij}) \tag{3.5}$$

and the 2^{n-1} Walsh coefficients, C", of $F_1(X)$ are given by

$$C_j'' = (1/2)(C_j + C_{ij}) \tag{3.6}$$

where j=all possible coefficient subscript sets not involving i, including j=0. Coefficient C_{ij}, for j=0, = C_i(see[4]).

Consider the following truth table of a single variable function , let us say X

X	F
0	1
1	1

The value of the function is always 1, independent of the variable X. For this function, we have $C_0=2$, $C_X=0$. So, whenever we have coefficients (2,0), then the value of the function is always 1(F=1). Similarly, for the set (-2,0), the value of the function is always 0(F=0).

Applying the same idea to the set of coefficients (0,-2) and (0,2), we obtain their truth tables and their corresponding BDDs as shown in Fig.3-4 and Fig.3-5, respectively.

Whenever a function is independent of any variable, then all the subscripted coefficients involving that variable will be zero. For example, consider the function $F(X_1,X_2,X_3)=X_2X_3$. The 2^3 Walsh coefficients are

$C_0=-4$, $C_1=0$, $C_2=4$, $C_3=4$

$C_{12}=0$, $C_{13}=0$, $C_{23}=4$, $C_{123}=0$.

Here, $C_1,C_{12},C_{13},C_{123}$ are zero. Conversely, we can say that whenever all the coefficients involving a given variable are zero, then the function is independent of that variable. If we want the spectrum in terms of the remaining variables, then we delete the subscripted coefficients involving that variable and divide the remaining

X	F
0	1
1	0

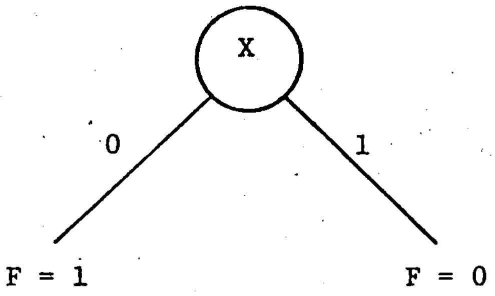


Figure 3-4.

X	F
0	0
1	1

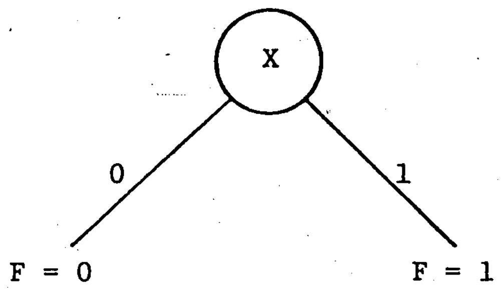


Figure 3-5.

spectrum by 2.

With this introduction, we now turn to the main algorithm that makes successive use of Equations(3.5) and (3.6). The algorithm is formulated as follows :

1. Select any variable X_i and assign it to the root of the BDD being constructed with two outgoing branches of value 0 and 1.
2. Calculate the two subsets of 2^{n-1} Walsh coefficients, C' and C" of the functions $F(X_1,\ldots,0,\ldots,X_n)$ and $F(X_1,\ldots,1,\ldots,X_n)$, respectively, corresponding to the branch values $X_i=0$ and 1 using (3.5) and (3.6).
3. If in any subset (C' or C"), all the coefficients involving k of the remaining variables are zero, then, eliminate them and divide the remaining coefficients by the factor of 2^K.
4. When any branch has a set of coefficients (-2,0) or (+2,0), terminate that branch with F=0 or F=1, respectively.
5. When any branch has a set of coefficients (0,2), connect that branch to the last remaining node variable. Terminate the 0 branch of this node with F=0 and the 1 branch with F=1.

   When any branch has a set of coefficients (0,-2), connect that branch to the last remaining node variable. Terminate the 0 branch with F=1 and 1 branch with F=0.

6. Whenever two sets corresponding to two distinct branches are the same, join the branches.
7. Select the next variable in each of the subsets C',C" and repeat steps (2)-(6).

EXAMPLE 1

Consider a set of (2^3) Walsh coefficients. Here n=3. Let $C_0=0$, $C_1=4$, $C_2=4$, $C_3=4$, $C_{12}=0$, $C_{13}=0$, $C_{23}=0$, $C_{123}=-4$.

In order to determine the BDD for the function corresponding to the given set of Walsh coefficients, we select the variable X_1 according to step(1), and calculate the subsets C' and C"as shown in Fig.3-6.

Since steps (3)-(6) are not applicable, X_2 is selected as the next variable and further subsets are calculated according to step(7) as shown in Fig.3-7. Next, applying steps (4)-(6), finally we obtain Fig.3-8.

The Walsh coefficients given in this example correspond to the function $F=X_1X_2+X_2X_3+X_1X_3$. Fig.3-8 corresponds to the BDD for this function.

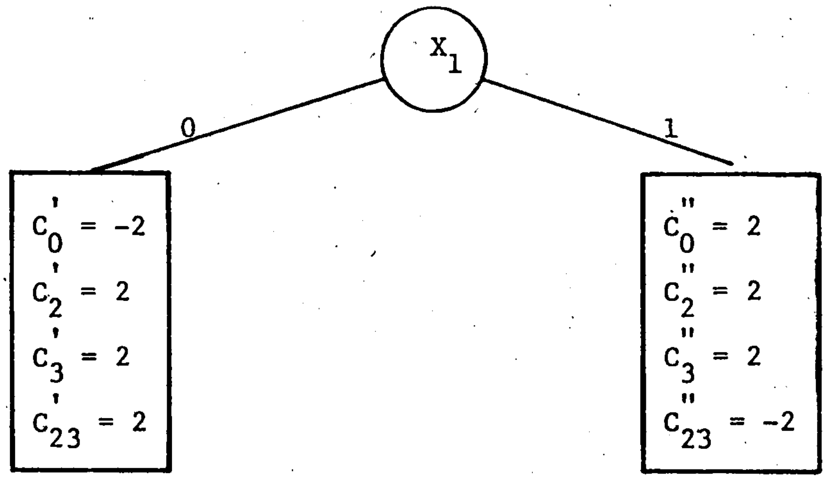


Figure 3-6.

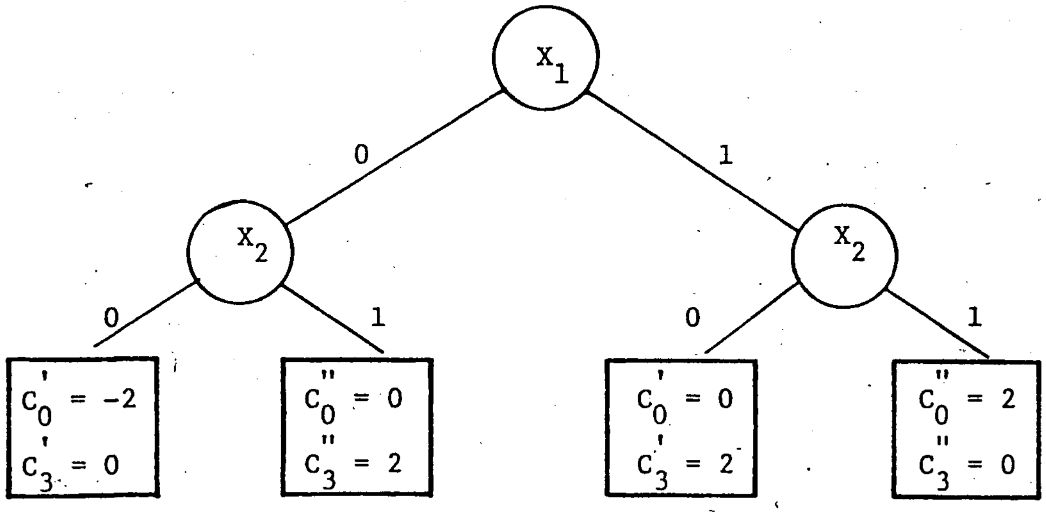


Figure 3-7.

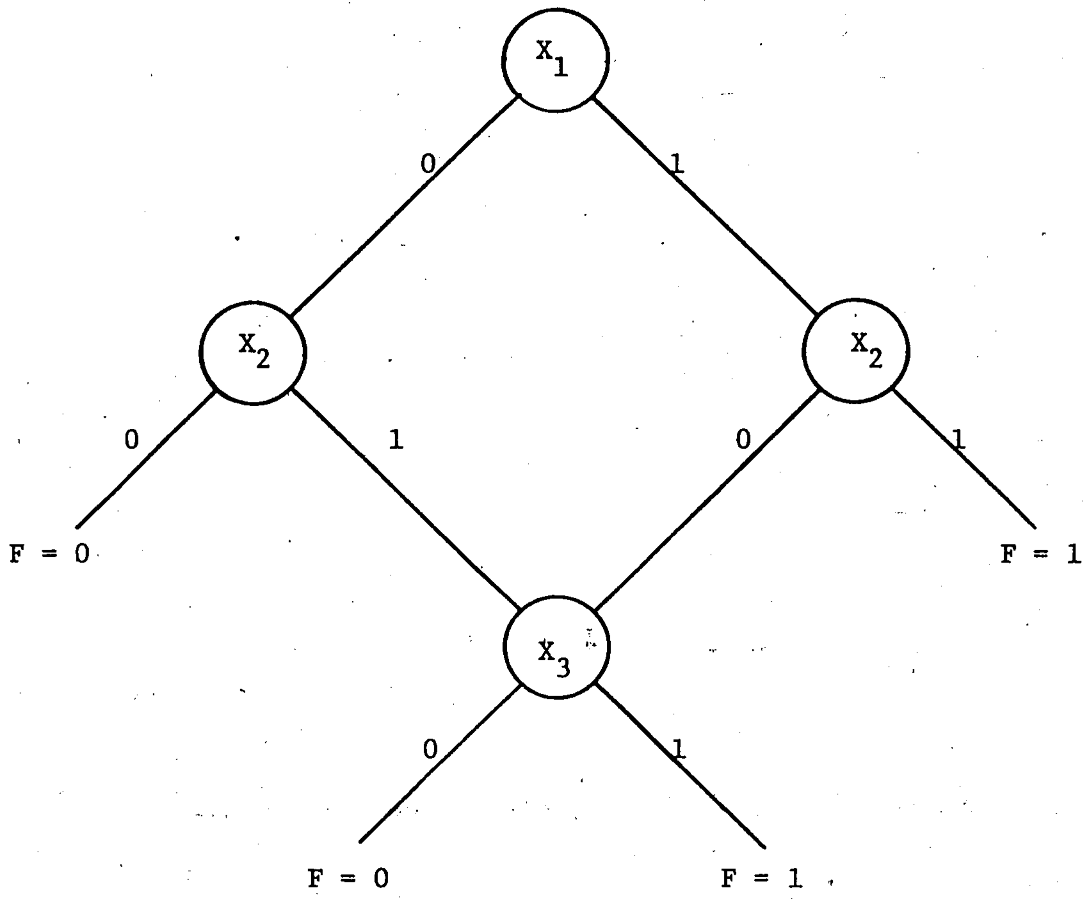


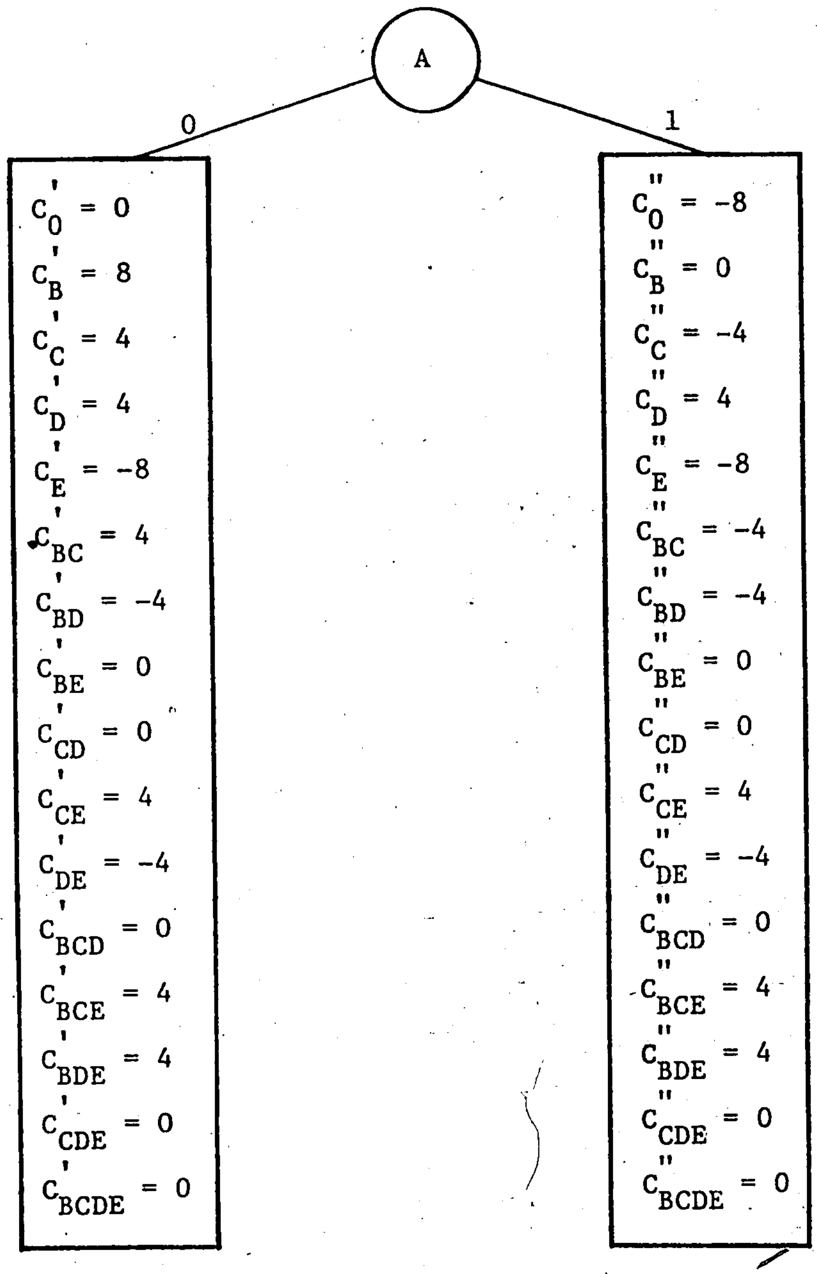
Figure 3-8.

EXAMPLE 2

Next, consider the set of 2^5 Walsh coefficients C_0=-8, C_A=-8, C_B=8, C_C=0, C_D=8, C_E=-16, C_{AB}=-8, C_{AC}=-8, C_{AD}=0, C_{AE}=0, C_{BC}=0, C_{BD}=-8, C_{BE}=0, C_{CD}=0, C_{CE}=8, C_{DE}=-8, C_{ABC}=-8, C_{ABD}=0, C_{ABE}=0, C_{ACD}=0, C_{ACE}=0, C_{ADE}=0, C_{BCD}=0, C_{BCE}=8, C_{BDE}=8. All remaining coefficients are zero.

In order to determine the BDD for the function corresponding to the given set of Walsh coefficients, we select the variable A according to step(1) and calculate the subsets C' and C" as shown in Fig.3-9. Next, we select variable B as a node on both of the outgoing branches of the variable A, as shown in Fig.3-10. Now, applying step(3), we obtain Fig.3-11. Next, the use of steps (5)-(7) results in Fig.3-12. Finally, applying steps (4)-(5), we obtain Fig.3-13.

The Walsh coefficients used in this example correspond to the function $F=\bar{B}D\bar{E}+B\bar{C}\bar{E}+\bar{A}BC$. Fig.3-13 corresponds to the BDD for this function.



A
0
1
$C_0' = 0$
$C_B' = 8$
$C_C' = 4$
$C_D' = 4$
$C_E' = -8$
$C_{BC}' = 4$
$C_{BD}' = -4$
$C_{BE}' = 0$
$C_{CD}' = 0$
$C_{CE}' = 4$
$C_{DE}' = -4$
$C_{BCD}' = 0$
$C_{BCE}' = 4$
$C_{BDE}' = 4$
$C_{CDE}' = 0$
$C_{BCDE}' = 0$
$C_0'' = -8$
$C_B'' = 0$
$C_C'' = -4$
$C_D'' = 4$
$C_E'' = -8$
$C_{BC}'' = -4$
$C_{BD}'' = -4$
$C_{BE}'' = 0$
$C_{CD}'' = 0$
$C_{CE}'' = 4$
$C_{DE}'' = -4$
$C_{BCD}'' = 0$
$C_{BCE}'' = 4$
$C_{BDE}'' = 4$
$C_{CDE}'' = 0$
$C_{BCDE}'' = 0$

Figure 3-9.

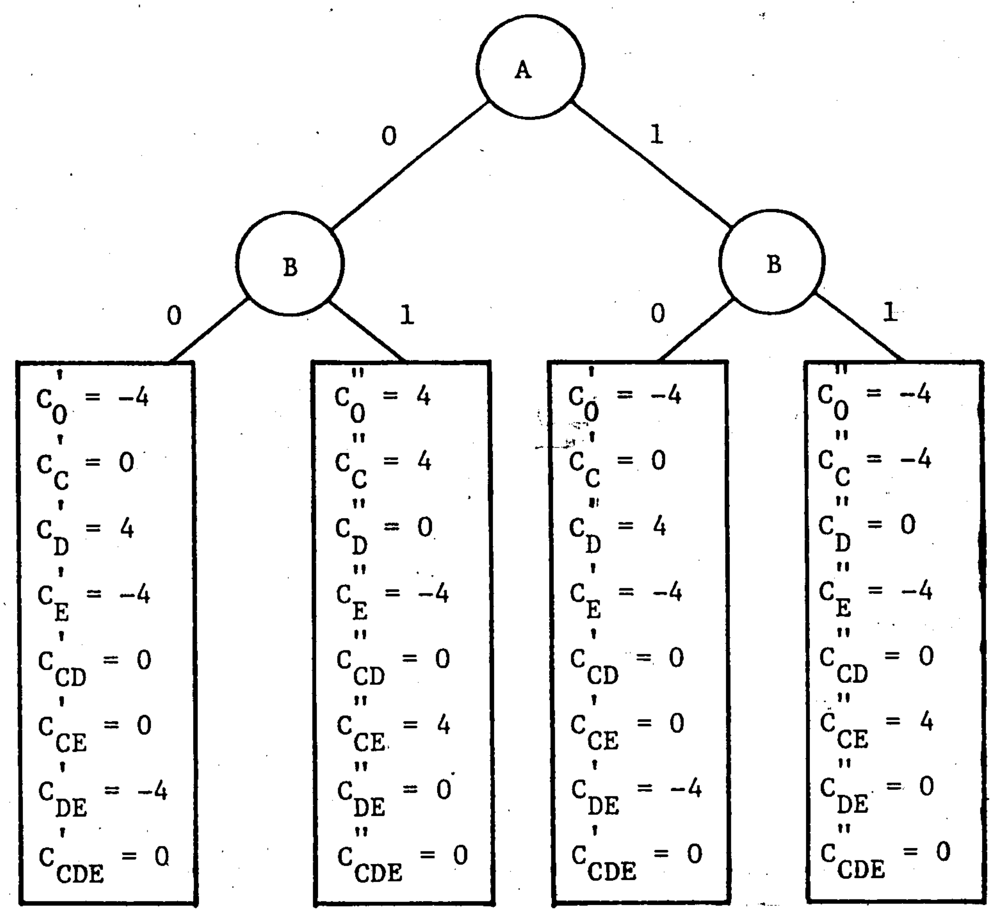

A
0
1
B
B
0
1
0
1
$C'_0 = -4$, $C'_C = 0$, $C'_D = 4$, $C'_E = -4$, $C'_{CD} = 0$, $C'_{CE} = 0$, $C'_{DE} = -4$, $C'_{CDE} = 0$
$C''_0 = 4$, $C''_C = 4$, $C''_D = 0$, $C''_E = -4$, $C''_{CD} = 0$, $C''_{CE} = 4$, $C''_{DE} = 0$, $C''_{CDE} = 0$
$C'_0 = -4$, $C'_C = 0$, $C''_D = 4$, $C'_E = -4$, $C'_{CD} = 0$, $C'_{CE} = 0$, $C'_{DE} = -4$, $C'_{CDE} = 0$
$C''_0 = -4$, $C''_C = -4$, $C''_D = 0$, $C''_E = -4$, $C''_{CD} = 0$, $C''_{CE} = 4$, $C''_{DE} = 0$, $C''_{CDE} = 0$

Figure 3-10.

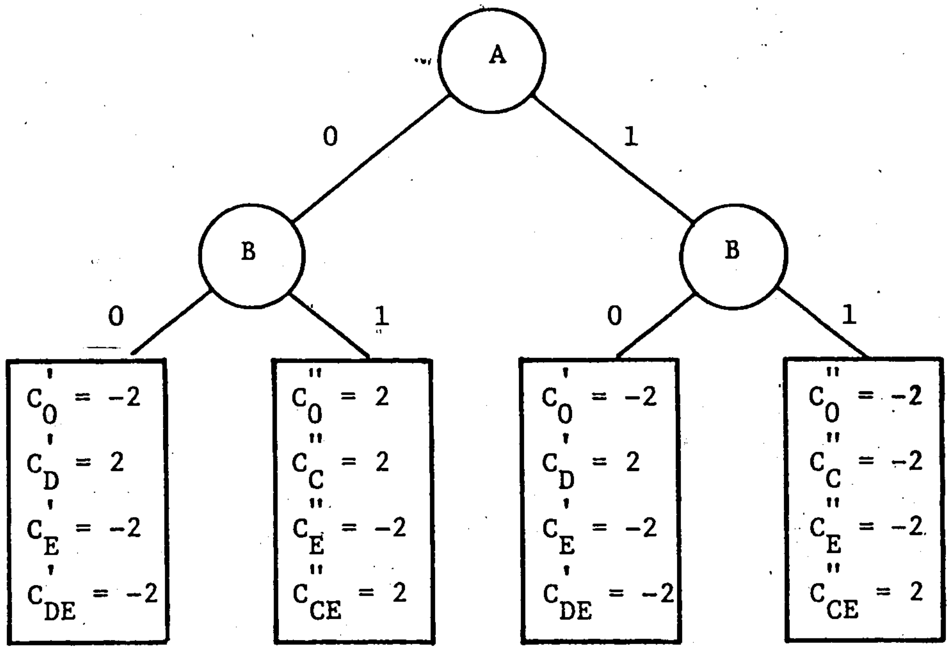

A
0
1
B
B
0
1
0
1
$C'_0 = -2$
$C'_D = 2$
$C'_E = -2$
$C'_{DE} = -2$
$C''_0 = 2$
$C''_C = 2$
$C''_E = -2$
$C''_{CE} = 2$
$C'_0 = -2$
$C'_D = 2$
$C'_E = -2$
$C'_{DE} = -2$
$C''_0 = -2$
$C''_C = -2$
$C''_E = -2$
$C''_{CE} = 2$

Figure 3-11.

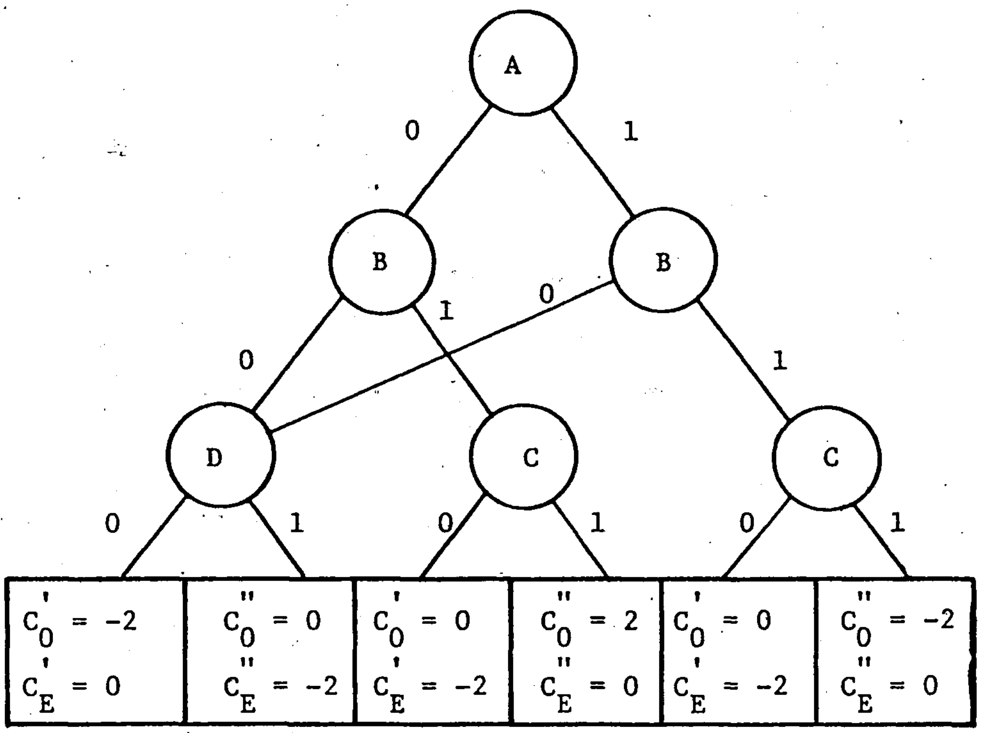

A
0
1
B
B
1
0
0
1
D
C
C
0
1
0
1
0
1
$C_0' = -2$
$C_E' = 0$
$C_0'' = 0$
$C_E'' = -2$
$C_0' = 0$
$C_E' = -2$
$C_0'' = 2$
$C_E'' = 0$
$C_0' = 0$
$C_E' = -2$
$C_0'' = -2$
$C_E'' = 0$

Figure 3-12.

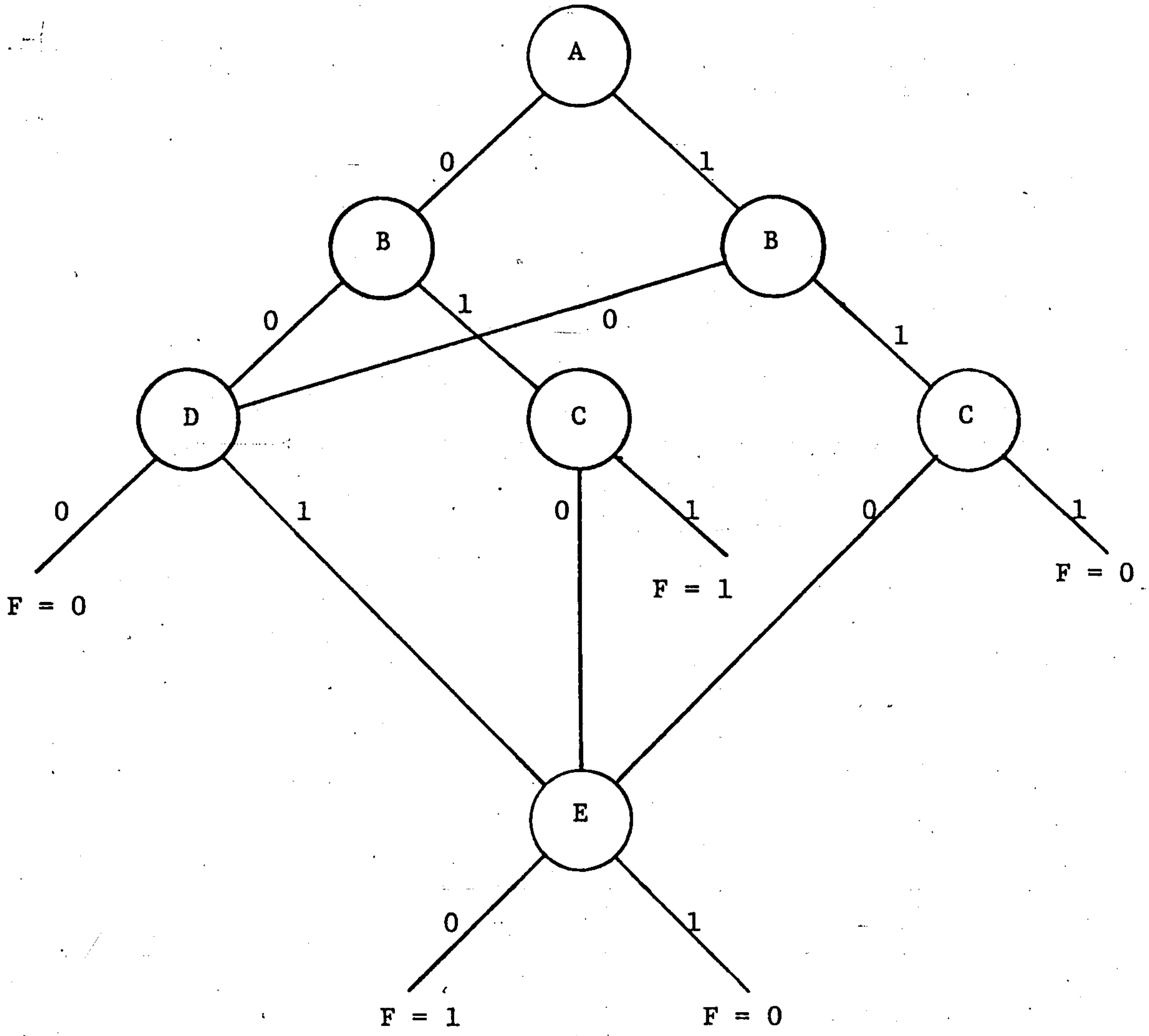


Figure 3-13.

3.3 SUMMARY

The Walsh coefficients are usually computed from the truth table. As the number of variables increases, the size of the truth table grows rapidly. It becomes tedious to compute the Walsh spectrum of a function of more than five variables. Since the BDD is a concise way to specify a Boolean function, the algorithm to compute the Walsh spectrum directly from the BDD is very useful. The second algorithm (constructing the BDD directly from the Walsh spectrum) illustrates the conversion from the spectral domain to the binary domain.

4. WALSH SPECTRUM FROM THE BOOLEAN EXPRESSION

This chapter contains an algorithm for computing the Walsh spectrum directly from the Boolean expression of the function $F(X)$. Then, we show how to find the spectrum of the Boolean Difference $dF(X)/dX_i$ from the spectrum of $F(X)$. Next, we show how to compute the number of true vertices of $X_i(dF(X)/dX_i)$ and $\bar{X}_i(dF(X)/dX_i)$.

The simple Boolean Difference of a function $F(X)$ with respect to one of its defining variables, X_i is defined as (see [7])

$$dF(X)/dX_i = F_a(X) \oplus F_b(X) \tag{4.1}$$

where $F_a = F(X_1,\ldots,X_i,\ldots,X_n)$, $F_b = F(X_1,\ldots,\bar{X}_i,\ldots,X_n)$ and $\oplus$ is the exclusive OR operator.

Solutions to $(dF(X)/dX_i)=1$ are independent of X_i and define the input for which a change in state of X_i causes a change of output state. The set of tests for a fault on X_i is given by

$$X_i(dF(X)/dX_i) = 1 \text{ , for } X_i \text{ stuck at } 0 \tag{4.2}$$

$$\bar{X}_i(dF(X)/dX_i) = 1 \text{ , for } X_i \text{ stuck at } 1 \tag{4.3}$$

4.1 COMPUTATION OF THE WALSH SPECTRUM DIRECTLY FROM THE BOOLEAN FUNCTION

In this section, we describe an algorithm to compute the complete Walsh spectrum directly from a non-canonical form of a given Boolean function F(X) of n variables. Our method is based on the observation that when the function is reduced to its disjoint form(sum of products form where products are mutually exclusive, i.e., no two product terms cover the same minterm), a variable appearing in an uncomplemented (complemented) form in a product term which has K missing variables contributes $+2^K(-2^K)$ to the appropriate partial product of the Walsh coefficients.

The algorithm can be expressed as follows :

1. Convert the given Boolean function to a disjoint form, if it is not in that form already(see [4]). There may be many disjoint forms. Choose any one .
2. In each product term with K missing variables, index each literal with $i=+2^K(-2^K)$, if uncomplemented (complemented).
3. The m^{th} ($m\leq n$) order Walsh coefficient can be written in the following form

$$C_{12...m} = 2\Sigma_{\text{prod terms}}\, \alpha(i) \qquad (4.4)$$

where, the sum is carried out over only those product terms that contain all 12...m variables and $\alpha(i)$ for each such product term is given by

$$\alpha(i) = \begin{cases} +|i|, & \text{if the number of negative variables in } 12..m \text{ is even in that product;} \\ -|i|, & \text{otherwise.} \end{cases}$$

If there is no product which involves all 12...m variables, then $C_{12...m}$ will be characterized by zero. The coefficient C_0 is given by

$$C_0 = 2[\Sigma_{\text{all prod terms}} 2^K - 2^{n-1}] \qquad (4.5)$$

where each product term contributes 2^K and summation is carried over all the product terms.

<u>PROOF</u>

The contribution to the Walsh coefficients of a product term which does not involve all the m variables will be zero, because half of the input combinations in the product term will contribute (+1) and other half will contribute (-1). Only the product terms which involve all the m variables will be considered. The product term, which has K missing variables, corresponds to 2^K input combinations associated with F=1.

There will be 2^{n-1} input combinations that have an even number of zeroes in the m variables and the remaining 2^{n-1} will have an odd number. Each input combinations having even number of zeroes associated with F=1 will contribute +1 and input combinations having an odd number of zeroes associated with F=1 will contribute -1. Conversely, the input combinations having an even number of zeroes associated with F=0 will contribute -1 and input combinations having an odd number of zeroes associated with F=0 will contribute +1.

If N_E is the number of input combinations having an even number of zeroes associated with F=1, then the number of input combinations having an even number of zeroes associated with F=0 will be $(2^{n-1}-N_E)$. Conversely, if N_O is the number of input combinations having an odd number of zeroes associated with F=1, then the number of input combinations having odd number of zeroes associated with F=0 will be $(2^{n-1}-N_O)$.

The mth order Walsh coefficients can be written as follows

$$C_{12\ldots m} = N_E - N_O - (2^{n-1}-N_E) + (2^{n-1}-N_O)$$
$$= 2(N_E-N_O)$$

In the terminology used in Eq(4.4), $+|i|$ corresponds to N_E and $-|i|$ corresponds to N_O.

The coefficient C_0 is the difference of the number of input combinations associated with F=1 and the number of input combinations associated with F=0. The total number of input combinations associated with F=1 is the sum of all product terms and each product term contributes 2^K.

Number of Input Combinations with F=1 $= \Sigma\, 2^K$
Number of Input Combinations with F=0 $= 2^n - \Sigma 2^K$

$$C_0 = \Sigma 2^K - (2^n - \Sigma 2^K)$$
$$= 2(\Sigma 2^K - 2^{n-1})$$

EXAMPLE 1

Consider the function $F=X_1X_2+X_2X_3+X_1X_3$. A disjoint form of F is given by $F=X_1X_2+\bar{X}_1X_2X_3+X_1\bar{X}_2X_3$. Each variable is indexed according to rule(2)as follows

$$X_1^{+2}X_2^{+2}+ X_1^{-1}X_2^{+1}X_3^{+1}+ X_1^{+1}X_2^{-1}X_3^{+1}$$

Rule(3) can now be used to compute the Walsh spectrum.

$C_0 = 2[2+1+1-4] = 0$,

$C_1 = 2[2-1+1] =4$,

$C_2 = 2[2+1-1] =4$,

$C_3 = 2[1+1] = 4$,

$C_{12} = 2[2-1-1] = 0$,

$C_{13} = 2[-1+1] = 0$,

$C_{23} = 2[1-1] = 0$,

$C_{123} = 2[-1-1] = -4$.

EXAMPLE 2

Consider the expression $F=\bar{B}D\bar{E}+B\bar{C}\bar{E}+\bar{A}BC$ which is already in a disjoint form. Each variable is indexed according to rule(2) as follows

$$B^{-4}D^{4}E^{-4} + B^{4}C^{-4}E^{-4} + A^{-4}B^{4}C^{4}$$

Rule(3) can now be used to compute the Walsh spectrum.

$C_0 = 2[4+4+4-16] = -8$,

$C_A = 2[-4] = -8$,

$C_B = 2[-4+4+4] = 8$,

$C_D = 2[+4] = 8$,

$C_E = 2[-4-4] = -8$,

$C_{AB} = 2[-4] = -8$,

$C_{BC} = 2[-4+4] = 0$,

$C_{CD} = 2[0] = 0$,

$C_{BE} = 2[+4-4] = 0$,

$C_{ABCDE} = 2[0] = 0.$

4.2 SPECTRUM OF $dF(X)/dX_i$

Let the spectrum of $F_a(X)$ in Eq(4.1) be C_a. As shown in [4], then the spectrum C_b of $F_b(X)$ is equal to C_a, except that the sign of all i-subscripted coefficients is reversed. It was shown in [4] that if in Eq(4.1), $F_a(X)$ and $F_b(X)$ are disjoint functions, then the spectrum of $(dF(X)/dX_i)$ can be easily calculated. To compute the $dF(X)/dX_i$ from the complete spectrum of $F(X)$, we can not apply this technique, because $F_a(X)$ and $F_b(X)$ may or may not be completely disjoint.

As was shown in [15], if $F_a(X)$ and $F_b(X)$ are two functions of n variables, in which (0,1) is coded in (-1,1) with corresponding spectra C_a and C_b, then the spectrum $C_{F_a \oplus F_b}$ of $F_a \oplus F_b$ is given as follows(for proof see [15]):

$$C_{F_a \oplus F_b} = -N'(C_a * C_b) \tag{4.6}$$

where $N'=1/N$ and $N=2^n$. $C_a * C_b$ is the convolution sum defined by

$$(C_a * C_b)_i = \sum_j C_{a_{i\oplus j}} \cdot C_{b_j} \tag{4.7}$$

where i and j denote all 2^n-1 combinations of the numbers 1,2,3,...,n and 0. $i \oplus j$ denotes the concatenation of the digits in i and j with the exclusion of any common digits. For example, $12\oplus34=1234$, $123\oplus345=1245$ and $0\oplus K=K$ for any K. Computation of $C_a * C_b$ can be understood from the following example:

<u>EXAMPLE1</u>

Consider the case of 3 variable function and let us determine the spectrum C' of $dF(X)/dX_1$. If C_a is given by

$$C_a=[C_0,C_1,C_2,C_{12},C_3,C_{13},C_{23},C_{123}]^T$$

Then C_b is given by

$$C_b=[C_0,-C_1,C_2,-C_{12},C_3,-C_{13},C_{23},-C_{123}]^T$$

In order to compute $dF(X)/dX_1$, using Eq(4.6) and Eq(4.7), we have

$$C'_0=-(1/8)[C_{0\oplus0}\cdot C_0+C_{0\oplus1}\cdot -C_1+C_{0\oplus2}\cdot C_2+C_{0\oplus12}\cdot -C_{12}+C_{0\oplus3}\cdot C_3 +C_{0\oplus13}\cdot -C_{13}+C_{0\oplus23}\cdot C_{23}+C_{0\oplus123}\cdot -C_{123}]$$

$$=-(1/8)[C_0^2-C_1^2+C_2^2-C_{12}^2+C_3^2-C_{13}^2+C_{23}^2-C_{123}^2]$$

$$C_1'=-(1/8)[C_{1\oplus 0}\cdot C_0+C_{1\oplus 1}\cdot -C_1+C_{1\oplus 2}\cdot C_2+C_{1\oplus 12}\cdot -C_{12}+ C_{1\oplus 13}\cdot -C_{13}+C_{1\oplus 23}\cdot C_{23}+C_{1\oplus 123}\cdot -C_{123}]$$
$$=0$$

Similarly, the remaining coefficients are computed as below:

$$C_2'=-(1/8)[2C_0C_2-2C_1C_{12}+2C_3C_{23}-2C_{13}C_{123}]$$

$$C_{12}'=0$$

$$C_3'=-(1/8)[2C_0C_3-2C_1C_{13}+2C_2C_{23}-2C_{12}C_{123}]$$

$$C_{13}'=0$$

$$C_{23}'=-(1/8)[2C_0C_{23}-2C_1C_{123}+2C_2C_3-2C_{12}C_{13}]$$

$$C_{123}'=0.$$

In general, C_0' of $dF(X)/dX_i$ can be written as follows:

$$C_0'=(1/N)[\sum_{i\subset j} C_j^2 - \sum_{i\not\subset k} C_k^2] \qquad (4.8)$$

where, the first summation is carried out on the squares of those coefficients which contain i in the subscript list and the second summation is carried out on the squares of those coefficients which do not contain i in the subscript list. As shown in EXAMPLE1, C_0' of $dF(X)/dX_1$ is

$C_0'=(1/8)[C_1^2+C_{12}^2+C_{13}^2+C_{123}^2-C_0^2-C_2^2-C_3^2-C_{23}^2]$

4.3 SPECTRUM OF $X_i(dF(X)/dX_i)$ AND $\bar{X}_i(dF(X)/dX_i)$

As was shown in [15], if F and G are functions of n variables with corresponding spectra C_F and C_G, then the spectrum C_{FG} of $F \wedge G$ is given by(for proof see [15]):

$$C_{FG} = (1/2)[N'(C_F * C_G) + (C_F+C_G) - J] \qquad (4.9)$$

To compute the spectra C'' of $X_i(dF(X)/dX_i$ and $\bar{X}_i(dF(X)/dX_i)$, we have $F = X_i$ or $\bar{X}_i$ and $G = dF(X)/dX_i$. Since F and G are supposed to be functions of n variables, we first convert C_F and C_G to n variable spectra having 2^n coefficients and then apply Eq(4.9) to compute the spectra of $X_i(dF(X)/dX_i)$ or $\bar{X}_i(dF(X)/dX_i)$.

<u>EXAMPLE 2</u>

Consider the three variable case of EXAMPLE 1. In order to compute the spectrum C'' of $X_1(dF(X)/dX_1)$, we have $F=X_1$, $G=dF(X)/dX_1$ and their corresponding spectra C_F and C_G are as follows :

$C_F=[0\ 8\ 0\ 0\ 0\ 0\ 0\ 0]^T$

$C_G=[C_0'\ 0\ C_2'\ 0\ C_3'\ 0\ C_{23}'\ 0]^T$

Using Eq(4.9), we finally have

$$\begin{bmatrix} C_0'' \\ C_1'' \\ C_2'' \\ C_{12}'' \\ C_3^3 \\ C_{13}'' \\ C_{23}'' \\ C_{123}'' \end{bmatrix} = (1/2) \begin{bmatrix} C_0' - 8 \\ C_0' + 8 \\ C_2' \\ C_2' \\ C_3' \\ C_3' \\ C_{23}' \\ C_{23}' \end{bmatrix}$$

To compute the spectrum of $\bar{X}_1(dF(X)/dX_1)$, we have

$$C_F = [0\ -8\ 0\ 0\ 0\ 0\ 0\ 0]^T$$
$$C_G = [C_0'\ 0\ C_2'\ 0\ C_3'\ 0\ C_{23}'\ 0]^T$$

Using Eq(4.9), we finally have

$$\begin{bmatrix} C_0'' \\ C_1'' \\ C_2'' \\ C_{12}'' \\ C_3'' \\ C_{13}'' \\ C_{23}'' \\ C_{123}'' \end{bmatrix} = (1/2) \begin{bmatrix} C_0' - 8 \\ -C_0' - 8 \\ C_2' \\ -C_2' \\ C_3' \\ -C_3' \\ C_{23}' \\ -C_{23}' \end{bmatrix}$$

4.4 COMPUTATION OF THE NUMBER OF TRUE VERTICES OF $X_i(dF(X)/dX_i)$ AND $\bar{X}_i(dF(X)/dX_i)$

In general, if X_i is an independent input then C_0'' of $X_i(dF(X)/dX_i)$ and $\bar{X}_i(dF(X)/dX_i)$, is given by

$$C_0'' = (1/2)[C_0' - 2^n] \tag{4.10}$$

C_0'', in fact is the difference between the number of true vertices and false vertices of $X_i(dF(X)/dX_i)$ or $\bar{X}_i(dF(X)/dX_i)$. True vertices of $X_i(dF(X)/dX_i)$ or $\bar{X}_i(dF(X)/dX_i)$ will give us the number of test patterns for which a stuck fault on X_i can be tested. In random testing, a large number of true vertices is desirable, because this fault will have high probability of being detected. We give a method of calculating the true vertices of these functions in terms of the spectrum of F(X).

$$C_0'' = N_1 - N_0$$

where N_1 is the number of true vertices where $X_i(dF(X)/dX_i) = 1$ and N_0 is the number of false vertices where $X_i(dF(X)/dX_i) = 0$.

But $N_1 + N_0 = 2^n$, so we have

$$C_0'' = 2N_1 - 2^n \tag{4.11}$$

Using Eq(4.10) and Eq(4.11), we have

$$N_1 = (1/4)C_0' + 2^{n-2}$$

Using Eq(4.8), we have

$$N_1 = (1/4N)[\sum_{i\subset j} C_j^2 - \sum_{i\not\subset k} C_k^2] + 2^{n-2}$$

EXAMPLE 3

To calculate the true vetices of $X_1(dF(X)/dX_1)$, we have for a three variable function

$$N_1 = -(1/32)[C_0^2 - C_1^2 + C_2^2 + C_3^2 - C_{12}^2 - C_{13}^2 + C_{23}^2 - C_{123}^2] + 2$$

Finally, given the spectrum of F(X), we can directly compute the true vertices N_1 of $X_i(dF(X)/dX_i)$ and $\bar{X}_i(dF(X)/dX_i)$ for any single input lead fault. To extend this concept to multiple faults, the spectrum of $X_iX_j(dF(X)/dX_iX_j)$ and $\bar{X}_i\bar{X}_j(dF(X)/dX_iX_j)$ can be similarly calculated.

4.5 SUMMARY

The algorithm to obtain Walsh spectra directly from the Boolean expression is very simple and fast. A lot of computation involved in computing Walsh spectra from matrix multiplication, is saved. We also showed that Walsh spectra can determine some parameters useful in random testing.

5. CONCLUSION

This chapter is a summary of the analytical results and the conclusions arrived at in preceding chapters. The principal aim of this thesis is to study bivalued functions in the spectral domain. In particular, the Walsh Rademacher transformation of Boolean functions is examined. Chapter 1 provides a brief introduction to the Walsh-Rademacher transformation. Any arbitrary set of 2^n numbers may not correspond to a valid Boolean function and two properties were derived that must be satisfied.

Next, an introduction to the representation of Boolean functions by Binary Decision Diagrams (BDD) is presented. The arbitrary choice of the starting variable does not always lead to the minimum number of nodes in a BDD. In chapter 2,an algorithm is presented to obtain a good BDD of a function. The algorithm presented can be easily computer programmed.

In chapter 3, the BDD of a function is used to compute the Walsh spectrum of the function. Conversely, from the Walsh spectrum of a function, the BDD of the function is constructed.

The computation of Walsh Coefficients usually requires a matrix multiplication. The size of the matrices increases rapidly ($\sim 2^n$) with the order, n, of the Boolean function. In chapter 4, a simple and direct algorithm to compute the Walsh spectrum from the Boolean expression is presented. All 2^n minterms are not required to compute the spectrum, only a disjoint form of the Boolean expression is needed. The Walsh Coefficients can be applied to the fault diagnosis and testing. It is shown that the number of test patterns, for a particular input lead being stuck, can be computed directly from the Walsh spectrum of the function.

REFERENCES

[1] S. B. Akers.
Binary Decision Diagram.
IEEE Trans. on Computers C-27(6):509-516, June, 1978.

[2] S. B. Akers.
Functional Testing with Binary Decision Diagram.
IEEE Proc. 8th International conference on Fault Tolerant computing :75-82, June, 1979.

[3] K. G. Beauchamp.
Walsh functions and their applications.
Academic Press, 1975.

[4] R. G. Benetts and S. L. Hurst.
The Rademacher-Walsh spectral transform : a new tool for problems in digital network fault diagnosis.
IEE Comput. Digital Technique. 1 :38-44, 1978.

[5] P. J. Davis.
The Mathmatics of matrices.
Xerox College Publishing, 1973.

[6] C. R. Edwards.
The application of the Rademacher-Walsh Transform to Boolean Function classification and Threshold Logic synthesis.
IEEE transaction on Computers C-24(1), January, 1975.

[7] C. R. Edwards.
The design of easily tested circuits using mapping and spectral techniques.
Radio and Electron Engg. 47(7):321-342, July, 1977.

[8] S. L. Hurst.
The application of chow parameters and Rademacher-Walsh matrices in the synthesis of binary functions.
Computer Journal 16(2):165-173, 1973.

[9] M. G. Karpovsky.
Finite orthogonal seies in the design of digital devices.
New York : Willey, 1976.

[10] A. M. Lloyd.
Spectral addition techniques for the synthesis of multivariable logic networks.
Computers and Digital Techniques 1(4):152-164, 1978.

[11] A. M. Lloyd.
A consideration of orthogonal matrices, other than the Rademacher-Walsh types, for the synthesis of digital networks.
Int. J. Electronics 47(3):205-212, 1979.

[12] A. M. Lloyd.
Design of multiple universal Logic-module networks using spectral techniues.
IEE Proceeding. Part.E 127(1), January, 1980.

[13] Jose S. Matos and John V. oldfield.
Binary Decision Diagram : From abstract representations to physical implementations.
20th design automation conference :567-570, 1983.

[14] J. C. Muzio and S. L. Hurst.
The Computation of Complete and reduced sets of orthogonal spectral coefficients for Logic Decision and pattern recognition purposes.
Comput. Elect. Engg. 5:231-249, 1978.

[15] J. C. Muzio.
Composite spectra and the analysis of switching circuits.
IEEE Trans. on Computer C-29(8), August, 1980.

[16] J. C. Muzio and D. M. Miller.
Spectral techniques for fault detection.
FTCS-12 , 1982.

[17] S. L. Pollack.
Conversion of Limited Entry Decision Tables to computer programs.
comm. of the ACM 8: 677-682, November, 1965.

[18] A. K. Susskind.
Testing by verifying Walsh coefficients .
IEEE Trans. on Computers , 1983.

[19] V. H. Tokmen.
The evaluation of the spectrum of Multilevel logic networks.
Comput. and Elect. Engg. 6:233-237, 1979.

[20] Arthur Willing.
Test generation using Binary Decision Diagram.
IEEE Autotestcon :336-346, 1981.

VITA

Suman Purwar was born in Allahabad in July, 1957. She received the B.E.(Hons) degree in Electrical Engineering from the M.N.R. Engg. College, Allahabad, India in 1978. She received the M.E.(Hons) degree in Control Engineering from the same Institute in 1980. After completing her M.E. programme, she joined Uttar Pradesh State Electricity Board in Sept 1980 and worked there till Dec 1981. In Jan 1982, she joined the Lehigh University to pursue her M.S. program in Computer Science and Electrical Engineering. She was a Teaching assistant from Jan 1982 to June 1984.