# Parity Function Detection and Realization Using a Small Set of Spectral Coefficients<sup>\*</sup>

M. A. Thornton, V. S. S. Nair Southern Methodist University Dept. of Computer Science and Engineering Dallas, Texas 75275 (214)768-2861 FAX:(214)768-3085 mitch@seas.smu.edu, nair@seas.smu.edu

# Abstract

A technique to detect and realize a parity logic function using a linear number of spectral coefficients is presented. Recent advances in computation methods for the spectra of a Boolean function have resulted in the determination of a single coefficient very efficiently. The use of a small set of spectral coefficients offers low cost and attractive alternatives to more traditional digital logic design and analysis techniques when coupled with the use of the new spectral computation approach. The detection and realization of the class of parity functions is an important problem since it generally requires the computation of  $2^n$  Walsh coefficients when spectral methods are used, or, extensive symbolic algebraic manipulation when other methods are used. The method presented in this paper only requires the computation of n + 2 Walsh coefficients and n + 1 non-Walsh coefficients to detect and realize a parity function.

# 1 Introduction

With the recent advent of an efficient method for the computation of a single spectral coefficient of a Boolean function comes a renewed interest in the use of these quantities for circuit design and analysis tasks [1]. In particular, methods that use a small subset of spectral coefficients are especially attractive since the computational efficiency provided by the new calculation technique is augmented by reducing the amount of storage required and the number of times the computation is performed. This paper provides a methodology for detecting and realizing the class of logic functions that compute the parity of a set of literals. This class of functions is usually difficult to detect efficiently when common graphical or symbolic approaches to logic analysis and design are applied.

The spectrum of a Boolean function is defined as a vector of integer values obtained through the linear mapping of a specific transformation matrix. Common transformation matrices include the Reed-Muller transformation matrix [2], or the Walsh family of matrices such as the Rademacher-Walsh or Hadamard [3]. As developed in [4], each row of the transformation matrix may be considered to be the output vector of a logic function, and thus the transformation matrices may be viewed as being composed of a collection of logic functions, called "constituent functions". Using this viewpoint, a general transformation matrix may be constructed with any arbitrary set of constituent functions. In this paper, we use n + 2 Walsh coefficients to detect whether a given n-variable function may be realized as a parity circuit and n+1 general spectral coefficients to determine the characteristics necessary for the actual realization.

The particular set of coefficients used to detect the parity function are the  $0^{th}$  and  $1^{st}$  ordered Walsh coefficients and a single  $i^{th}$  ordered Walsh coefficient, where *i* may range from 2 to *n*. The  $0^{th}$  and  $1^{st}$  ordered Walsh coefficients are usually referred to as the Chow parameters in recognition of the work in [5] although the original definition differs slightly from that originally defined by Chow. The realization portion of this process uses n + 1 spectral coefficients corresponding to constituent functions that are not used in any of the common transformation matrices.

This problem has great practical importance when spectral coefficients are used to synthesize a logic cir-

 $<sup>^{*}</sup>$  This project was supported in part by the NSF under grant number MIP-9410822.

cuit. The advent of methods to compute spectral coefficients with a complexity generally much less than  $O(2^n)$  has provided the motivation for the investigation into the use of the coefficients for logic synthesis. However, regardless of the efficiency of the computation of the coefficients, if all  $2^n$  are required, the synthesis algorithm will not prove to be practical. Since the Chow parameters correspond to the simplest set of constituent functions, it is interesting to try to exploit the information they provide to solve digital logic problems. We are currently working on a synthesis methodology that uses a reduced set of spectral coefficients including the Chow parameters. The particular technique that is the subject of this paper arose from this investigation and is a solution to a 'sub-problem' encountered in the development of the synthesis technique. Specifically, when the Chow parameters are all zero-valued, particular difficulties arise since algebraic sign information is not present and some variables could be redundant. In this paper we show that all parity functions have zero-valued Chow parameters, however the converse of this statement is not necessarily true. In order to detect whether a logic function is a parity function or not, we compute an  $i^{th}$  ordered Walsh coefficient that is determined by the properties of the additional n + 1 non-Walsh coefficients. If the detection criteria indicates the presence of a parity function, we then realize it since the additional n + 1non-Walsh coefficients can be used to determine the redundant inputs and the output polarity of the the circuit.

Many logic circuit optimizers require a structural input and provide an output that is also in a netlist form. Recently, some efficient spectral computation methods have been proposed that use BDDs as input [1] [6] [7] [8]. However, if the circuit is in the form of a netlist initially, it may be more work to obtain a suitable variable ordering and then build a BDD representation than it would be to compute the spectral coefficients directly from the netlist. In addition, some logic functions require an exponential number of BDD vertices versus a much smaller netlist. For these classes of circuits, the use of an efficient spectral computation technique using a structural input such as that described in [9] is clearly advantageous.

The remainder of this paper is organized as follows. Section 2 contains a brief discussion of the notation and definitions used in this development. Section 3 contains the mathematical results used for the detection and realization of parity functions using the small set of spectral coefficients. In Section 4, these results are summarized in an algorithmic form suitable for implementation as a computer program and an illustrative example is provided. Finally, Section 5 provides the conclusions of this work and indicates some related areas of future research.

# 2 Notation and Definitions

The formulation of the efficient method for computing spectral coefficients requires alternative definitions and notation than is typically used when discussing Boolean function spectra [1]. In order to present these results in a manner compatible with the efficient calculation technique, the same notation is used in the remainder of this paper.

- n is the number of variables in a Boolean function.
- Small case variables such as  $x_0$ ,  $x_1$ , etc. denote Boolean variables that have logic values of "1" or "0".
- The operator symbol, "+", will refer to the Boolean OR function or the addition of real numbers depending upon the context of the equation in which it is used.
- The operator symbol, ".", will refer to the Boolean AND operation. The absence of an operator between two adjacent values in a Boolean equation implies the presence of the "." operator.
- The operator symbol, "⊕", will refer to the Boolean XOR operation.
- The spectral coefficient of a function, f, with respect to a constituent function,  $f_c$ , is denoted as  $S_f[f_c]$ .
- $N_m$  is an integer value that indicates the number of times two functions yield the same output value for all possible input combinations. Likewise,  $N_{mm}$  is the number of times the outputs differ thus  $N_m + N_{mm} = 2^n$ .
- A Boolean function is said to be *degenerate* if its output is independent of one or more of its inputs.
- A particular Boolean function input is said to be *redundant* if its value does not affect the output.
- A Boolean function that may be formed as the XOR of some or all of its single literals is referred to as a parity function in the remainder of this paper. In mathematical terms, the parity function has the form:

$$f(x) = a_1 \hat{x}_1 \oplus a_2 \hat{x}_2 \oplus \dots \oplus a_n \hat{x}_n \tag{1}$$

where

$$a_i \in \{0, 1\} \tag{2}$$

and  $\hat{x}_i$  denotes  $x_i$  may or may not be inverted.

There are varying definitions of the spectrum of a Boolean function. The different spectra are classified based upon the transformation matrices used to compute them. As discussed in [4], the transformation matrices may be viewed as a collection of "constituent" functions whose output vectors are used as row vectors in the transformation matrix. Most of the commonly used transformation matrices always include constituent functions that correspond to the constant function (either  $f_{c0} = 0$ , or,  $f_{c0} = 1$ ) and functions that are equal to each primary input ( $f_{ci} = x_i$  for i = 1...n). This set of spectral coefficients form the Chow parameters [5]. As an example, the following transformation matrix could be used to compute the Chow parameters of a three-input function:

1	1	1	1	1	1	1	1	1]
$x_1$	0	0	0	0	1	1	1	1
$x_2$	0	0	1	1	0	0	1	1
$x_3$	0	1	0	1	0	1	0	1

Notice that the constituent functions corresponding to each row vector of the transformation matrix are shown to the left of the matrix. As an example of the calculation of the Chow parameters, consider the circuit whose logic diagram is shown in Figure 1.



Figure 1: Logic Circuit Example for Chow Parameter Computation

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 3 \\ 2 \end{bmatrix}$$
(3)

The computation in Equation 3 is representative of the original definition of the Chow parameters as given in [5]. In this form, the Chow parameters yield information regarding the total number of minterms present in the circuit's logic equation (given by the coefficient corresponding to  $f_{c0} = 1$ ), and, the number of minterms in which each primary input,  $x_i$ , appears in an uncomplemented form (given by each respective  $f_{ci} = x_i$ ).

The family of Walsh transformation matrices are more general than the matrix used in Equation 3 in that they contain only the real values of 1 and -1. The logic "1" values are replaced with -1 and the logic "0" values are replaced with +1 in the formation of these matrices. This allows the -1 products in the evaluation of the inner products to accumulate thus providing more information about the function being transformed. This form causes the spectral coefficients to lie in the range  $[-2^n, 2^n]$  and it has the desirable feature that the coefficient corresponding to a particular  $x_i$  is always zero-valued if that  $x_i$  is a redundant input. However, the fact that a particular coefficient is computed to be 0 does not necessarily imply that the corresponding  $x_i$  is redundant. This property is formally proven in the work given in [10]. The fact that degenerate functions will always have at least one zero-valued Chow parameter is used in the method we develop here, thus, in the remainder of this paper we assume the Chow parameters are computed using this form. As an example, the Chow parameters computed in Equation 3 would become:

It is useful to see that the original form of the Chow parameters can be used to compute the form in Equation 4 directly. Note that the  $0^{th}$ -ordered spectral coefficient in Equation 3 corresponds to the constituent function,  $f_{c0} = 1$ , while the same spectral coefficient in Equation 4 corresponds to  $f_{c0} = 0$ . In order to define algebraic relationships between the form of the Chow parameters in Equations 3 and 4 we will denote them as  $\{C_f[f_{c0}], C_f[f_{c1}], C_f[f_{c2}], \ldots C_f[f_{cn}]\}$  and  $\{S_f[f_{c0}], S_f[f_{c1}], S_f[f_{c2}], \ldots S_f[f_{cn}]\}$  respectively. The relationships are given as:

$$S_f[f_{c0}] = 2^n - 2C_f[f_{c0}] \tag{5}$$

$$S_f[f_{ci}] = 4C_f[f_{ci}] - 2C_f[f_{c0}]$$
(6)

# 3 Mathematical Basis for the Technique

This section provides a discussion and a derivation of the results used to develop the parity function detection and realization technique. First, we show that a necessary condition for a function to be considered a candidate for realization as a parity circuit is that all of its Chow parameters must be zero-valued. Next, we state the obvious fact that the spectral coefficient of the candidate function with respect to a constituent function describing the parity circuit must have the value of  $2^n$ . Finally, we show how the additional n + 1non-Walsh coefficients can be used to determine which parity circuit is a probable candidate to use for realizing the logic function.

It has been shown that the Chow parameters can provide information regarding input and output inversions and variable permutations for an NPNequivalent class of functions [11]. Unfortunately, this information cannot be exploited for the class of parity functions because all of the Chow parameters are zero valued for this group. However, the fact that all of the Chow parameters are necessarily zero valued can be used to determine if a given logic function is suitable for realization as a parity function. In order to formulate a detection technique for parity functions, we prove that for any possible parity function the Chow parameters must be zero valued.

The first rather obvious result concerns the identification of inverted literals. Lemma 1 can be used to show that all parity functions may be represented by two unique forms regardless of the polarity of the literals. This Lemma may be used to disregard the task of determining individual polarities of literals, and instead, translates the problem to the determination of whether or not the output of the parity circuit should be inverted.

**Lemma 1** All non-degenerate parity functions are equivalent to the following even or odd parity functions regardless of the polarity of the input literals.

$$f_1 = x_1 \oplus x_2 \oplus \dots \oplus x_n \tag{7}$$

 $f_2 = \overline{x_1 \oplus x_2 \oplus \dots \oplus x_n} \tag{8}$ 

**Proof:** The non-degenerate parity functions simply compute the even (or odd) parity of the inputs. If a

single input is inverted, the corresponding overall parity bit is inverted since  $x \oplus \overline{x} = 1$ . Hence, the function represented by  $f_1$  initially may now be represented by  $f_2$  (and vice versa). If an even number of inputs are inverted, the original function remains the same since  $\overline{x_1} \oplus \overline{x_2} \oplus \overline{x_3} \oplus \cdots \oplus \overline{x_m} \oplus x_{m+1} \oplus \cdots \oplus x_n = x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_m \oplus x_{m+1} \oplus \cdots \oplus x_n$ for m even. Conversely,  $\overline{x_1} \oplus \overline{x_2} \oplus \overline{x_3} \oplus \cdots \oplus \overline{x_m} \oplus x_{m+1} \oplus \cdots \oplus x_n = x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_m \oplus x_{m+1} \oplus \cdots \oplus x_n$ for m odd.

The following results prove that a necessary condition for a given logic function to be realized as a parity circuit is that all Chow parameters must be zero-valued.

**Lemma 2** A parity function always has a "0"-valued  $0^{th}$ -order Chow parameter.

**Proof:** Consider a non-degenerate, non-inverted parity function. The output of this function is the even parity bit of all inputs. The  $0^{th}$ -order Chow parameter may be defined as the difference of the number of function outputs at logic "0" and those at logic "1". Thus, for this spectral coefficient to equal 0, half of all possible input terms should have even parity with the remaining half having odd parity. Therefore, the lemma is proved upon the condition that for all possible binary strings of n bits, half have an even number of 1s and half have an odd number of 1s.

The number of binary strings containing an even number of bits may be computed as:

$$n_{even} = \begin{pmatrix} n \\ 0 \end{pmatrix} + \begin{pmatrix} n \\ 2 \end{pmatrix} + \dots + \begin{pmatrix} n \\ 2\lfloor \frac{n}{2} \rfloor \end{pmatrix}$$
(9)

Likewise, the number of possible binary strings of length n with odd parity is:

$$n_{odd} = \begin{pmatrix} n \\ 1 \end{pmatrix} + \begin{pmatrix} n \\ 3 \end{pmatrix} + \ldots + \begin{pmatrix} n \\ 2\lfloor \frac{n}{2} \rfloor + 1 \end{pmatrix} (10)$$

Note that,

$$n_{even} + n_{odd} = \begin{pmatrix} n \\ 0 \end{pmatrix} + \begin{pmatrix} n \\ 1 \end{pmatrix} + \begin{pmatrix} n \\ 2 \end{pmatrix} + \dots + \begin{pmatrix} n \\ n \end{pmatrix}$$
(11)

$$n_{even} + n_{odd} = \sum_{i=0}^{n} \binom{n}{i} = 2^{n}$$
(12)

From the Binomial Theorem:

$$(1-1)^n = 0 = \binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \dots - (-1)^n \binom{n}{n}$$
(13)

or,

Or equivalently,

$$\begin{pmatrix} n \\ 0 \end{pmatrix} + \begin{pmatrix} n \\ 2 \end{pmatrix} + \dots + \begin{pmatrix} n \\ 2\lfloor \frac{n}{2} \rfloor \end{pmatrix}$$
$$= \begin{pmatrix} n \\ 1 \end{pmatrix} + \begin{pmatrix} n \\ 3 \end{pmatrix} + \dots + \begin{pmatrix} n \\ 2\lfloor \frac{n}{2} \rfloor + 1 \end{pmatrix} \quad (14)$$

Using the definitions of  $n_{even}$  and  $n_{odd}$  in Equations 9 and 10 and substituting into Equation 14, we see that

$$n_{even} = n_{odd} \tag{15}$$

Solving for  $n_{even}$  and  $n_{odd}$  simultaneously using Equations 15 and 11 yields the result:

$$n_{even} = n_{odd} = 2^{n-1} \tag{16}$$

Therefore, the  $0^{th}$ -ordered Chow parameter is:

$$S_0 = n_{even} - n_{odd} = 2^{n-1} - 2^{n-1} = 0$$
 (17)

The non-degenerate, inverted parity function case also holds for this lemma since the only change is that the  $n_{even}$  minterms produce a logic "1" output and the  $n_{odd}$  terms produce a logic "0" output. In the most general case, a degenerate parity function, the lemma also holds. Consider a parity function of mvariables with 1 redundant input. If all  $2^n$  minterms are tabulated and the redundant input is removed, the list becomes the concatenation of 2 lists of all possible minterms of length m - 1. Thus, substituting m - 1 for  $n_{even}$  and  $n_{odd}$  in Equation 17 will result in a 0-valued,  $0^{th}$ -ordered Chow parameter. Further, by induction, the same argument may be applied for degenerate functions with more than 1 redundant input.  $\square$ 

The following lemma shows that all of the  $1^{st}$  ordered Walsh coefficients must have a value of zero for a parity function.

**Lemma 3** All Chow parameters of a parity function are zero-valued.

**Proof:** Consider the  $i^{th}$  Chow parameter corresponding to the spectral coefficient formed by using the constituent function,  $f_{ci} = x_i$ . If the  $x_i$  input and the first input,  $x_1$ , are interchanged and the resulting minterms are (and corresponding function outputs) are rewritten in ascending order, the first  $2^{n-1}$  minterms correspond to  $x_i = 0$  and the remaining  $2^{n-1}$  minterms correspond to  $x_i = 1$ .

Using the results of Lemma 1, it is apparent that each subset produces an equal number of "1" and "0" outputs. Thus, the number of output values matching the  $x_i$  input is  $2^{n-2}$  in each subset. The total number of outputs that match  $x_i$  (denoted by  $N_m$ ) is:

$$N_m = 2^{n-2} + 2^{n-2} = 2^{n-1} \tag{18}$$

It is shown in [4] that any spectral coefficient may be computed as:

$$S_f[f_c(x)] = 2N_m - 2^n$$
(19)

Substituting Equation 18 into Equation 19 yields:

$$S_f[f_c(x)] = 2(2^{n-1}) - 2^n = 0$$
(20)

The previous lemmas have proven that a necessary condition is that a parity function will always have 0-valued Chow parameters. However, the presence of a set of 0-valued Chow parameters is not sufficient to guarantee that the corresponding function may be realized as a parity function. As an example consider the function given in Equation 21.

$$f(x) = x_1 \oplus x_3 \oplus x_2 x_4 \tag{21}$$

Clearly, Equation 21 cannot be represented by a parity function as defined in this paper, but it does have all zero-valued Chow parameters. The following derivations show how a set of non-Walsh spectral coefficients may be used to determine a candidate parity function that may possibly realize the original logic function. Once the parity function is determined, it is used as a constituent function for the calculation of one additional spectral coefficient. If this spectral coefficient has a value of  $||2^n||$ , the function is realized. If the value of the additional coefficient is not  $||2^n||$ , the candidate logic function is not in the class of parity functions. It should be noted that all higher ordered Walsh coefficients utilize constituent functions that are parity functions. Therefore this method indicates which of the higher ordered Walsh coefficients should be computed to determine if a parity circuit can be used for the realization.

The application of traditional Walsh type spectral computations require the computation of  $2^n$  coefficients in to find the suitable parity function. However, by determining two additional facts about the logic function under consideration, we can determine which Walsh spectral coefficient is needed before resorting to computing the entire spectrum. The two additional facts are needed to uniquely identify the exact form of the function are:

• Determination of which (if any) inputs are redundant • Determination of whether or not the output of the parity function should be inverted. This determination indicates whether an odd or even parity function will be used.

The following lemmas show that the two criteria given in the preceding may determined by examining the properties of n + 1 non-Walsh spectral coefficients.

**Lemma 4** The set of n + 1 spectral coefficients computed using the following constituent functions:

$$f_{cn0} = x_1 x_2 \dots x_n$$

$$f_{cn1} = \overline{x}_1 x_2 \dots x_n$$

$$f_{cn2} = x_1 \overline{x}_2 \dots x_n$$

$$\vdots$$

$$f_{cnn} = x_1 x_2 \dots \overline{x}_n$$

with respect to a parity function, always have a magnitude of ||2||.

**Proof:** From Lemma 3, the  $2^n$  minterms may always be rewritten to form two disjoint subsets with only the  $x_i$  input differing in value. Since the parity function output is a parity value, the output will always result in different values when the two minterms that differ only in  $x_i$  are used as input.

The output vector of the  $f_{cni}$  functions given in the lemma always contain  $2^n - 1$  logic "0" values and a single logic "1" value since the  $f_{cni}$  functions are a single minterm. In terms of the number of matching values in the output vector of the parity function and the output vector of the constituent function, two cases arise. In the first case, the singular output logic value of "1" of the constituent function matches a logic "1" value on the output vector of the parity function. It follows that the  $2^{n-1}$  logic "0" output values of the parity function match  $2^{n-1}$  logic "0" values in the output vector of the constituent function. The total number of matching values becomes:

$$N_m = 2^{n-1} + 1 \tag{22}$$

Substituting this result into Equation 19, the corresponding spectral coefficient is computed as:

$$S_f[f_c(x)] = 2(2^{n-1} + 1) - 2^n = 2$$
(23)

In the other case, the single logic "1" output value of the constituent function mismatches with a logic "0" output of the parity function and the remaining  $2^{n-1}-1$  logic "0" outputs of the parity function match with logic "0" output values of the constituent function. Thus, the total number of matches in this case becomes:

$$N_m = 2^{n-1} - 1 \tag{24}$$

Substituting this result into Equation 19, the corresponding spectral coefficient is computed as:

$$S_f[f_c(x)] = 2(2^{n-1} - 1) - 2^n = -2$$
(25)

**Lemma 5** If a parity function has a redundant input,  $x_i$ , then

$$S_f[x_1x_2\dots x_i\dots x_n] = S_f[x_1x_2\dots \overline{x}_i\dots x_n] \quad (26)$$

**Proof:** Consider a parity function with a single redundant input,  $x_i$ . The output of the function is the same for the two inputs corresponding to  $x_1x_2\ldots x_i\ldots x_n$  and  $x_1x_2\ldots \overline{x_i}\ldots x_n$ . Since by definition a parity function with a non-redundant  $x_i$  input would compute differing parity values for these two constituent functions, if  $S_f[x_1x_2\ldots x_i\ldots x_n] = S_f[x_1x_2\ldots \overline{x_i}\ldots x_n]$ , the  $x_i$  input must be redundant.  $\Box$ 

The ramifications of Lemmas 4 and 5 are that redundant inputs may be determined by examining two of the additional spectral coefficients and noting whether or not they are identical, and, that the values of the additional coefficients may be determined by simply evaluating the function output for n + 1minterms. Specifically, the n + 1 minterms are those given by the additional constituent functions. Since it is proven that they will produce spectral values of ||2||, it is only necessary to determine the arithmetic sign. The arithmetic sign is negative if the function is evaluated to be a logic "0", and it is positive if the function is evaluated to be a logic "1" for a particular constituent function.

Using these results, the output polarity and the redundant inputs can be identified. Therefore, a unique candidate parity function is specified. Since the presence of zero valued Chow parameters is only a necessary condition for allowing a logic function to be represented as a parity circuit, it must be determined if the candidate parity function does indeed represent the initial logic function. This determination is easily carried out by computing an additional spectral coefficient using the candidate parity function as a constituent function. The well documented properties of spectral coefficients dictate that this coefficient must have a value of  $2^n$  if the parity function does indeed realize the logic function.

# 4 Implementation

The results of the previous section show that any parity function may be represented by two distinct forms function and that any redundant inputs are easily determined as well as whether the resulting parity should be even or odd. This information is determined by computing 2n + 3 spectral coefficients. In algorithmic form the steps required to analyze a particular function are:

#### Parity Function Detection and Realization Method

- **STEP 1** Determine function is a candidate by observing that all Chow parameters are zero valued.
- **STEP 2** Compute the additional n + 1 spectral coefficients. The additional coefficients may be computed using a spectrum calculation technique, or, by simply evaluating the function for n + 1 different inputs.
- **STEP 3** Determine the total number of redundant inputs (if any) and count the number of non-redundant inputs,  $N_{NR}$ . This is accomplished by determining the number of spectral coefficients for the constituent functions:

$$f_{cn1} = \overline{x}_1 x_2 \dots x_n$$
  

$$f_{cn2} = x_1 \overline{x}_2 \dots x_n$$
  
:

 $f_{cnn} = x_1 x_2 \dots \overline{x}_n$ 

that have the same value as the spectral coefficient obtained by using the constituent function:

 $f_{cn0} = x_1 x_2 \dots x_n$ Then,  $N_{NR}$  is the difference of this number and n, the total number of parity function inputs.

- **STEP 4** If  $N_{NR}$  is even and  $S_f[x_1x_2...x_n] = 2$ , the output of the candidate parity function should be inverted, otherwise it is not. Likewise, if  $N_{NR}$  is odd and  $S_f[x_1x_2...x_n]$ = -2, the output of the parity function should be inverted, otherwise it is not.
- **STEP 5** Compute the Walsh coefficient corresponding to the constituent function specified by the candidate parity function. If the value is  $||2^n||$ , the function may be realized as a parity function.
- **STEP 6** For the final step, all redundant inputs identified in STEP 3 are either left

unused in the resulting function, or their corresponding inputs can be set to logic "0".

As an example consider the function whose structural representation is given in Figure 2. Application of the spectral coefficient calculation algorithm given in [9] results in the set of 6 Chow parameters. Upon examination, it is noted that all of the Chow parameters are zero-valued. By Lemma 1 it is seen that this function may be represented by a parity function. Once this observation has been made, the additional n+1 coefficients are determined by observing the output of the logic function when inputs correspond to the minterms defined by the n + 1 additional constituent functions. Table 1 contains the Chow parameters and the additional spectral coefficients. The output of the function is shown for the additional spectral values to illustrate how the values may be obtained by simply evaluating the circuit output. Since the spectral values will have a magnitude of ||2||, the resulting function outputs dictate the arithmetic sign. The constituent functions are denoted by  $f_c(x)$ , the output of the function being transformed is denoted by f(x), and, the corresponding spectral coefficients are labeled as  $S_f(f_c)$ .



Figure 2: Example Logic Circuit for Parity Function Analysis

After the additional coefficients have been obtained, the number of non-redundant inputs,  $N_{NR}$ , is computed. Since there are 5 inputs present in the example circuit and 3 of the additional coefficients match the coefficient given by the constituent function,  $x_1x_2x_3x_4x_5$ , the total number of nonredundant inputs is computed to be  $N_{NR} = 5 - 3 = 2$ .

$f_c(x)$	f(x)	$S_f(f_c)$
0	-	0
$x_1$	-	0
$x_2$	-	0
$x_3$	-	0
$x_4$	-	0
$x_5$	-	0
$x_1\oplus x_2\oplus x_3\oplus x_4\oplus x_5$	-	0
$x_1 x_2 x_3 x_4 x_5$	1	2
$\overline{x}_1 x_2 x_3 x_4 x_5$	1	2
$x_1\overline{x}_2x_3x_4x_5$	1	2
$x_1x_2\overline{x}_3x_4x_5$	0	-2
$x_1x_2x_3\overline{x}_4x_5$	1	2
$x_1x_2x_3x_4\overline{x}_5$	0	-2

Table 1: Spectral Coefficients for the Example Circuit

Next, it is noted that the value of  $N_{NR}$  is even and that  $S(x_1x_2x_3x_4x_5) = 2$ , thus step 4 of the method given in the preceding indicates that the output of the function should be inverted. Now the circuit is completely identified to be equivalent to the odd parity function given in Equation 27 and the analysis is complete.

$$f = \overline{x_3 \oplus x_5} \tag{27}$$

This section has shown that the analysis and classification of parity functions may be easily accomplished by computing 2n + 3 spectral coefficients. The initial n + 1 coefficients are the Chow parameters. When the Chow parameters are all 0-valued and the  $i^{th}$  Walsh coefficient has a value of  $2^n$ , a parity function is indicated. The additional n + 1 spectral coefficients are computed to determine the candidate parity function. In addition, they indicate which inputs are redundant and whether the output of parity function should be inverted. Further, the additional n + 1 coefficients may be easily determined by evaluating the function to be classified for n + 1 minterms.

### 5 Conclusion

This paper has presented a methodology for the detection and realization of parity functions by using a small subset of spectral coefficients. In the past, such methodologies were overlooked due to the high computational cost of computing spectral coefficients. Due to recent advances in the computation of the spectrum of a Boolean function, approaches such as these provide a viable means for using spectral quantities in digital systems engineering tasks. This result is useful since this particular class of logic functions is quite difficult to handle using traditional spectral transformation matrices or other methods such as graphically or symbolically based techniques.

We have shown that by computing 2n + 3 spectral coefficients, degenerate parity functions may be realized in one of two distinct forms. Further, the n + 1 non-Walsh type spectral coefficients utilize very simple constituent functions and therefore their computation is reduced to simply evaluating the output of the function for n + 1 inputs.

Finally, the technique presented in this paper is very general in that it may be used with a variety of input representations. All that is required is some means to compute the coefficients thus, any type of suitable input may be used, either a structural or a functional representation is sufficient.

# References

- M. A. Thornton and V. S. S. Nair. Efficient Spectral Coefficient Calculation Using Circuit Output Probabilities. *Digital Signal Processing: A Re*view Journal, vol. 4:245-254, October 1994.
- [2] D. Green. Modern Logic Design. Addison-Wesley, Reading, Massachusetts, 1986.
- [3] M. Davio, J.-P. Deschamps, and A. Thayse. Discrete and Switching Functions. Mc-Graw-Hill, New York, New York, 1978.
- [4] M. A. Thornton and V. S. S. Nair. An Iterative Combinational Logic Synthesis Technique Using Spectral Information. *Proceedings of the European Design Automation Conference*, pages 358-363, September 1993.
- [5] C. K. Chow. On the Characterization of Threshold Functions. *IEEE Special Publication S.134*, pages 34-38, 1961.
- [6] M. A. Thornton and V. S. S. Nair. Efficient Calculation of Spectral Coefficients and Their Applications. *IEEE Trans. on CAD*, to appear.
- [7] E. M. Clarke, K. L. McMillan, X. Zhao, M. Fujita, and J. Yang. Spectral Transformations for Large Boolean Functions with Applications to Technology Mapping. *Proceedings of ACM/IEEE Design Automation Conference*, pages 54–60, 1993.

- [8] M. Fujita, J. Yang, E. M. Clarke, X. Zhao, and P. McGeer. Fast spectrum computation for logic functions using binary decision diagrams. To Appear in Proceedings of the International Conference on Circuits and Systems, 1994.
- [9] M. A. Thornton and V. S. S. Nair. The Computation of a Boolean Function Spectrum Using a Structural Input. *Technical Report*, 94-CSE-40, 1994.
- [10] S. Muroga. Threshold Logic and its Application. Wiley, New York, 1971.
- [11] S. L. Hurst, D. M. Miller, and J. C. Muzio. Spectral Techniques in Digital Logic. Academic Press, Orlando, Florida, 1985.