Walsh Spectrum Computations Using Cayley Graphs

Whitney J. Townsend and Mitchell A. Thornton
Mississippi State University
Starkville, MS
{wjt1,mitch}@ece.msstate.edu

***Abstract*-- The Walsh spectrum for a Boolean function has found many uses in VLSI CAD. A graph-based approach to calculating this spectrum using Cayley graphs is extended here to include alternate encoding, direct computation of the spectrum for the inverse of a function and a "fast" method of calculation for the adjacency matrix of a graph.**

***Index Terms*--Boolean Function, Cayley Graph, Walsh Spectrum**

I. INTRODUCTION

SPECTRAL methods have been used in logic design for synthesis [16, 9, 12, 7, 10], testing [4, 13, 8, 14] and function classification [9, 6]. However spectral methods have seen little practical application until recently due to the computational cost for calculating the spectrum. Graph-based methods utilizing decision diagram (DD) [2] structures have been developed which decrease the cost for calculating the spectrum [15, 11, 5].

An alternative graph-based method using Cayley graphs to compute the spectrum for a function was presented in [1]. This technique is of theoretical interest because it demonstrates the equivalence of the spectra of Cayley graphs and the Walsh spectra for Boolean functions.

In this paper, several extensions to the graph-based method in [1] are shown. In particular, alternative encodings and analysis of other possible field relations are explored. A group yielding a Cayley graph representing the spectrum for the inverse of a function is presented and a "fast" method for producing the adjacency matrix for the Cayley graphs of both groups is described.

The organization of the paper is as follows. Section II presents the necessary background information of the matrix-based method for the calculation of the Walsh spectrum described in [9] and the graph-based method described in [1]. In Section III, the extensions are presented and illustrated by examples. Concluding observations are presented in Section IV.

II. BACKGROUND

A. Calculation of the Walsh Spectrum

A function can be transformed from the Boolean domain into a number of alternative spectral domains. The traditional technique for the calculation of the Walsh spectrum for a Boolean function is presented in [9]. The use of this technique for an example function, $f = \bar{x}_1 \bar{x}_3 + x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3$ is shown in Figure 1.

$$\begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \\ -2 \\ 0 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$

Figure 1. Calculation of the Walsh Spectrum for the Example Function

B. R-encoding and S-encoding

The example function as shown in Figure 1 above is represented by *R*-encoding in which logic 1 is coded by a 1 and logic 0 is coded by a 0. An alternative representation known as *S*-encoding can also be defined in which logic 1 is coded by a -1 and logic 0 is coded by a +1 [9]. The *S*-encoded spectrum for a function can be obtained directly by encoding both the transformation matrix and the output vector for the function utilizing *S*-encoding or the *R*-encoded coefficients can be converted to *S*-encoded coefficients by the following equations.

$$s_0 = 2^n - 2\, r_0$$
$$s_i = -2\, r_i \;\forall\; i \subset \{1, 2, \ldots, n\}$$

C. Cayley Group

An alternative approach for the computation of the Walsh spectrum for a Boolean function based on algebraic groups and graph theory is described in [1]. The Cayley graph (or Cayley color graph) is a structure that is used to relate an algebraic group to graph theory [17, 3]. This technique for the computation of the Walsh spectrum for a function, *f*, relies upon representing the Boolean function based upon a specific definition of a group.

Recall that a group, $(M, *)$, consists of a set, *M*, and a binary operator on *M*, $*$, such that closure, associativity and

This work was supported in part by the NSF under grant CCR-0000891.

identity hold and that inverses exist. That is, for all m_i, $m_j \in M$, the element $m_i * m_j$ is a uniquely defined element of M (closure). That $m_i * (m_j * m_k) = (m_i * m_j) * m_k$ holds for all m_i, m_j, $m_k \in M$ (associativity). That there exists an identity element, $e \in M$, such that, $e * m_i = m_i$ and $m_i * e = m_i$ for all $m_i \in M$ (identity). Finally that there exists an inverse element $m_i^{-1} \in M$ such that $m_i * m_i^{-1} = e$ and $m_i^{-1} * m_i = e$ for each $m_i \in M$ (inverses exist).

The Cayley group, $(M, \oplus)$, used in the technique described in [1] characterizes the Boolean function, f: $B^n \rightarrow B$. $(M, \oplus)$ is a group in which M consists of all possible minterms in B^n, that is, all points in the space defined by B^n and $\oplus$ is the binary operator for the group. This group has an identity element corresponding to an n-length bit string of all zeros and additionally for each element $m_i \in M$, $m_i^{-1} = m_i$.

D. Cayley graph

The Cayley graph corresponding to this group representing the Boolean function, f, has a vertex set, V, in which each $v_i \in V$ uniquely corresponds to an element of the set $m_i \in M$. The edge set, E, is given by the equation below.

$$E = \{(m_i\, m_j) \in B^n \times B^n \mid f(m_i \oplus m_j) = 1\}$$

The adjacency matrix, A, for this Cayley graph is a matrix of size $2^n \times 2^n$ with $a_{ij} = 1$ if $f(m_i \oplus m_j) = 1$ and with $a_{ij} = 0$ otherwise. A is a symmetric matrix because $m_i \oplus m_j = m_j \oplus m_i$. The adjacency matrix for the example function is shown in Figure 2 and the corresponding Cayley graph described by A is shown in Figure 3.

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Figure 2. Adjacency Matrix for the Example Function Utilizing *R*-encoding

The spectrum of a graph is defined as the set of eigenvalues for the adjacency matrix representing it in [3]. The theorems and proofs given in [1] demonstrate that the spectrum of the Cayley graph representing the group as defined in [1], which in turn represents some Boolean function, f, is identical to the Walsh spectrum utilizing *R*-encoding for the Boolean function.

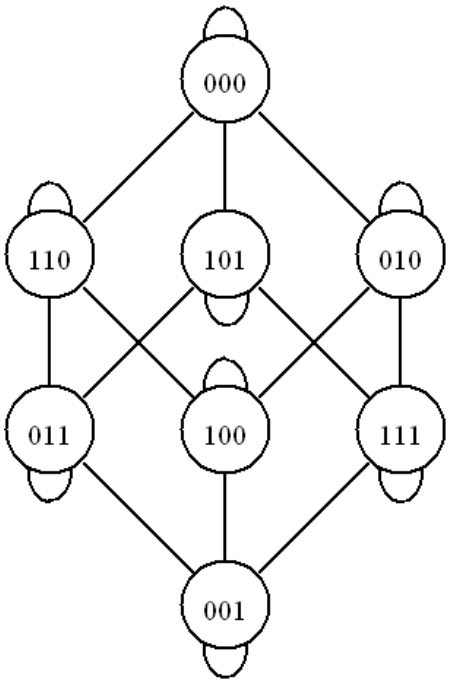

Figure 3. Cayley Graph for the Example Function Utilizing *R*-encoding

The characteristic polynomial $C(\lambda)$ for the adjacency matrix given in Figure 2 is shown below.

$$C(\lambda) = \lambda^8 - 8\lambda^7 + 16\lambda^6 + 16\lambda^5 - 80\lambda^4 + 64\lambda^3$$

Solving $C(\lambda) = 0$ yields the eigenvalues, $\lambda_i \;\forall\; i = \{1,2,...,8\} = \{4, 2, 0, -2, 0, 2, 0, 2\}$. These eigenvalues are the Walsh spectral coefficients for f as verified in Figure 1.

III. EXTENSIONS

A. S-encoding

The first extension to the technique presented in [1] was to verify that in a manner analogous to that used for the matrix-based calculation of the Walsh spectrum as discussed in [9], *S*-encoding of each element, $m_i \in M$, results in a graph whose eigenvalues directly yields the *S*-encoded coefficients for the Boolean function, f. The adjacency matrix, B, resulting for the example function when utilizing *S*-encoding is shown in Figure 4 and the corresponding Cayley graph described by B, is shown in Figure 5. Solving for the characteristic polynomial for this graph yields the *S*-encoded Walsh coefficients $\lambda_i \;\forall\; i = 1, 2, \ldots, 8 = \{0, -4, 0, 4, 0, -4, 0, -4\}$. Note that the topology of the graph in Figure 5 is unchanged from that of Figure 3, only the encoding of the vertices is different.

$$B = \begin{bmatrix} -1 & +1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & -1 & -1 & +1 & +1 & -1 \\ -1 & +1 & -1 & +1 & -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & -1 & +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & -1 & +1 & -1 & +1 & -1 \\ -1 & +1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & -1 & +1 & +1 & -1 & +1 & -1 \end{bmatrix}$$

Figure 4. Adjacency Matrix for the Example Function Utilizing *S*-encoding

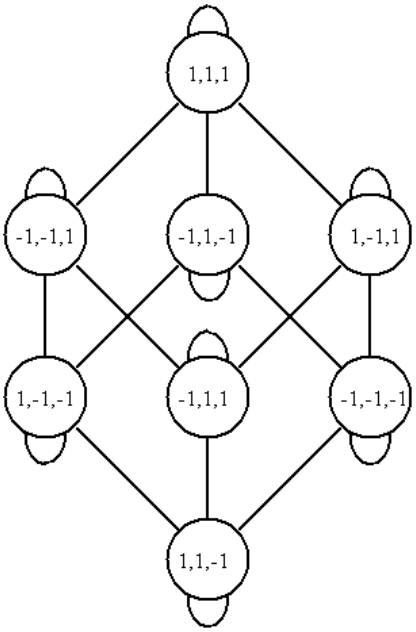

Figure 5. Cayley Graph for the Example Function Utilizing *S*-encoding

B. Other possible operators

Next all the remaining fifteen Boolean functions of two variables were considered as possible alternative operators to $\oplus$ in the formation of other Cayley groups. Only the two non-unate functions, EXOR ($\oplus$) and equivalence (EXNOR, $\equiv$) were found to satisfy the definition of a group using the mapping operation in [1].

C. Equivalence

The Cayley group defined using equivalence as the Boolean operator, $(M, \equiv)$, proved to have properties similar to the correspondence between the two operators, $\oplus$ and $\equiv$. This group has an identity element corresponding to an *n*-length bit string of all ones and for each element $m_i \in M$, $m_i^{-1} = m_i$. The adjacency matrix, *C*, for the example function using this definition for the Cayley group is shown in Figure 6 and the corresponding Cayley graph is shown in Figure 7.

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 6. Adjacency Matrix for the Inverse of the Example Function

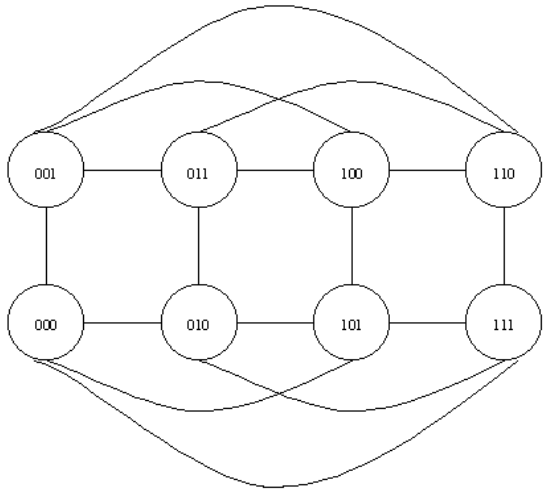

Figure 7. Cayley Graph for the Inverse of the Example Function

As is shown by Figure 7, the topology of this graph, for the same example function is quite different from the one produced by the Cayley group, $(M, \oplus)$, as shown in Figure 3. Of particular interest is the presence of self-loops in Figure 3 and their absence in Figure 7. This is determined by the value of $m_0 \oplus m_0$. For the example function, *f*, and the Cayley group, $(M, \oplus)$,

$$(m_0 \oplus m_0) = (000 \oplus 000) = 000$$

which is a minterm for the example function, *f*, while for the Cayley group, $(M, \equiv)$,

$$(m_0 \equiv m_0) = (000 \equiv 000) = 111$$

which is not a minterm for the example function. It is the value of this calculation that determines the presence or absence of self-loops in the corresponding Cayley graph for the function. If the result is a minterm of the function, self-loops will appear at all vertices in the graph; if the result is not a minterm of the function, self-loops will not appear in the corresponding Cayley graph.

Solving the characteristic polynomial for the new Cayley group, $(M, \equiv)$, produces the Walsh spectrum for the inverse of the example function directly, $\lambda_i \ \forall \ i = \{1,2,...,8\} = \{4, -2, 0, 2, 0, -2, 0, -2\}$.

As in the previous discussion on *S*-encoding, if the example function, *f*, is *S*-encoded using the Cayley group, $(M, \equiv)$, the *S*-encoded Walsh coefficients for the inverse of the example function can also be obtained directly using $(M, \equiv)$.

D. Direct Calculation of the Adjacency Matrix

During the computations required to obtain the adjacency matrix for a function using the definition of a Cayley group, a method was discovered which greatly minimizes the computational cost of producing the adjacency matrix for a function under consideration. In a method similar to the "fast transform butterfly diagrams" described in [15] it becomes possible to obtain all the other $n_1,..., n_n$ rows of the adjacency matrix from the n_0 row by a series of transpositions as shown in Figure 8. Additionally, because the first row of the adjacency matrix for the Cayley group, $(M, \oplus)$, consists of the equation, $e * m_i = m_i \ \forall \ i \subset \{1, 2, \ldots, n\}$, the first row of the adjacency matrix can be obtained directly from a transposition of the output vector for the function. Conversely, for the Cayley group, $(M, \equiv)$, the equation, $e * m_i = m_i \ \forall \ i \subset \{1, 2,..., n\}$, occurs in the n_n row of the adjacency matrix and thus the transformation can proceed in a similar manner from right to left.

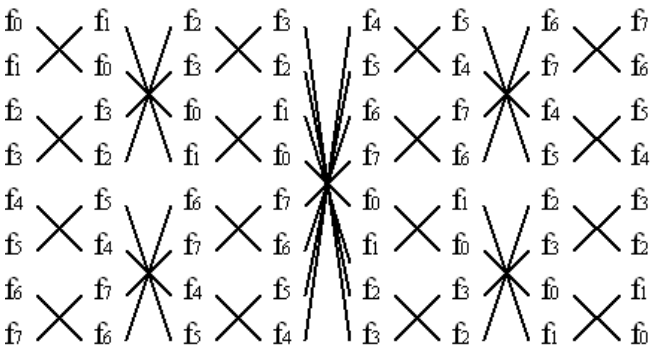

Figure 8. Transformation Diagram

IV. CONCLUSION

A new Cayley group whose graph represents the spectrum for the inverse of a function is presented. Additionally it is shown that both the group, $(M, \oplus)$, and the group, $(M, \equiv)$, can be used to directly calculate *S*-encoded coefficients by *S*-encoding of each element, $m_i \in M$. Finally a "fast" method for calculating the adjacency matrix by transposition is presented.

REFERENCES

[1] A. Bernasconi and B. Codenotti, "Spectral analysis of Boolean functions as a graph eigenvalue problem," *IEEE Trans. on Comp*., 48:345-351, 1999.

[2] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. on Comp*., 35(8):677-691, 1986.

[3] D. M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs*. Academic Press, 1979.

[4] T. Damarla, "Generalized transforms for multiple valued circuits and their fault detection," *IEEE Trans. on Comp*., 41(9):1101-1109, 1992.

[5] R. Drechsler and B. Becker, *Binary Decision Diagrams - Theory and Implementation*. Kluwer Academic Publishers, 1998.

[6] C. R. Edwards, "The application of the Rademacher-Walsh transform to Boolean function classification and threshold logic synthesis," *IEEE Trans. on Comp*., 24: 48-62, 1975.

[7] C. R. Edwards, "The design of easily tested circuits using mapping and spectral techniques," *Radio and Electronic Engineer*, 7:321-342, 1977.

[8] T. C. Hsiao and S. C. Seth, "An analysis of the use of Rademacher-Walsh spectrum in compact testing," *IEEE Trans. on Comp.*, 33:931-937, 1984.

[9] S. L. Hurst, D. M. Miller, and J. C. Muzio, *Spectral Techniques in Digital Logic*. Academic Press Publishers, 1985.

[10] M. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*. Wiley and JUP, 1976.

[11] D. M. Miller, "Graph algorithms for the manipulation of Boolean functions and their spectra," in *Congressus Numerantium*, pp 177-199, Winnipeg Canada, 1987.

[12] D. M. Miller, "A spectral method for Boolean function matching," in *European Design & Test Conf*., pg 602, 1996.

[13] D. M. Miller and J. C. Muzio, "Spectral fault signatures for single stuck-at faults in combinational networks," *IEEE Trans. on Comp*., 33:765-768, 1984.

[14] A. K. Susskind, "Testing by verifying Walsh coefficients," *IEEE Trans. on Comp*., 32:198-201, 1983.

[15] M. A. Thornton and R. Drechsler, "Spectral decision diagrams using graph transformations", in *Design, Automation and Test in Europe*, pp. 713-717, 2001.

[16] M. A. Thornton and V. S. S. Nair, "Efficient calculation of spectral coefficients and their application," *IEEE Trans. on CAD*, 14(11):1328-1341, 1995.

[17] A. T. White, *Graphs, Groups and Surfaces*. North-Holland Publishing Company, 1973.