Automation and Remote Control, Vol. 65, No. 6, 2004, pp. 1007–1017. Translated from Avtomatika i Telemekhanika, No. 6, 2004, pp. 188–200.

DISCRETE MODELING

Mixed-radix MVL Function Spectral and Decision Diagram Representation[1]

M. A. Thornton

Southern Methodist University, Dallas, USA
Received July 21, 2003

Abstract—Discrete finite-valued functions are increasingly important in applications involving automation and control. In particular, it is evident that industry is focusing on "Systems-on-a-Chip" (SoC) where the integration of analog (infinite-valued) and digital (binary-valued) circuits must co-exist. As designers struggle with these interfacing issues, it is natural to consider the intermediate circuits that can be modeled as multi-valued, discrete logic-level circuits. This viewpoint is not unprecedented as such principles have been used for at least the past twenty years in telecommunications protocols. If an analogous approach is considered in control systems implemented in "Integrated Circuit" (IC) designs, it is proposed that spectral analysis may provide an important role and efficient methods for computing such mixed-radix function spectra are described here. These methods are formulated as transformations of word-level decision diagrams representing the underlying arithmetic expressions and can be implemented as graph traversal algorithms. The theoretical foundation of the spectral transform of a mixed-radix function is presented and the equivalence of the resulting spectrum and the spectrum of a Cayley graph is shown.

1. INTRODUCTION

It is shown that the analysis and synthesis of functions with discrete domain and range spaces may be considered using algebraic group theory and spectral analysis. It is proposed that such theory provides a natural framework for the use of arithmetic expressions that represent multi-valued (i.e., greater than binary-valued) functions. In particular it is conjectured that this evolution of modern circuitry will first be composed of infinite-valued (analog) circuitry that is interfaced with digital (binary-valued) circuits and later be generalized to consist of circuitry that can be modeled as a discrete mixed-radix arithmetic expression. It is argued that the wealth of knowledge obtained using discrete transforms for sampled analog signals may be well suited for the synthesis and analysis of these conjectured new systems.

It has been shown that the Walsh spectrum of a binary-valued function $f(x_1, x_2, \ldots, x_n)$ may be computed as the spectrum of a Cayley graph over the elementary additive Abelian group $\mathbf{Z}_{2^n}$ using a generator based on f [1]. These results were also generalized to a technique to compute the Chrestenson spectrum [3,4] of finite discrete-valued functions in [2]. In general, these techniques can be proven to yield these spectra through the use of group character theory as described in [5] where the resulting spectral values are shown to be equivalent to inner products of the Cayley graph color vectors and the rows of the group character tables. If a proper generator is used in the formation of the Cayley graph, the corresponding color graph vector can be defined such that it is equivalent to a discrete function truth vector. Because the rows of the Walsh and Chrestenson transformation matrices are defined as the rows of the group character tables describing the elementary additive

[1] This work was supported by the United States National Science Foundation, project nos. CCR-0243358 and CCR-0243365.

Abelian group [8], the spectrum of the Cayley color graph is equivalent to the spectrum of the discrete function.

Mixed-radix "Multiple Valued Logic" (MVL) functions are assumed to be finite and discrete-valued and depend on a finite-valued variable support set $\{x_i, \dots, x_j\}$ such that x_i is q_i-valued and x_j is q_j-valued with $q_i \neq q_j$. The spectra of such MVL functions is of interest to circuit designers and automated design tool researchers and developers. Spectral transforms are described that are applicable to such functions over the elementary additive (mod(p)) Abelian groups. Three formulations of such transforms are described here; a linear transformation matrix derived from a group character table, a Kronecker-based expansion allowing for a "fast" transform algorithm, and a Cayley graph spectrum computation. It is shown that a particular spectral transformation of a discrete mixed-radix function over $\mathbf{Z}_6$ is equivalent to that over $\mathbf{Z}_2 \times \mathbf{Z}_3$ within a permutation. Also, it is shown that a Cayley graph may be formed over $\mathbf{Z}_6$ with a generator corresponding to the discrete function of interest.

The results in [1, 2] considered the spectra of p-valued functions with variable support sets consisting of q-valued variables. Here the subject of computing the spectra of p-valued functions with support sets of variables $\{x_1, x_2, \dots, x_n\}$ are considered such that each variable x_i is q_i-valued and $q_i \neq q_j$ for various pairs of (i, j). Here, such functions are referred to as *mixed-radix* since the support variables are $x_i \in \mathbf{Z}_i$ for different values of i. It is noted that the case of the function f being p-valued is not as of much interest as the case of different-valued variables in the support set since for $p = 2$ a Cayley graph results and for $p > 2$ a Cayley color graph results which can simply be considered as a disjoint set of Cayley graphs.

The techniques described here allow for input-output signals of discrete control systems represented as word-level decision diagrams to be directly transformed into graphs representing corresponding spectral arithmetic expressions. Furthermore, efficient algorithms exist for the transformation of functions represented as word-level decision diagrams into their spectral counterparts. One application for these arithmetic expressions is use in system identification computations. Nonparametric estimation can be achieved through the use of spectral and correlation analysis methods for estimation of frequency functions. As a practical example, spectral estimates are used in the formulation of control system models for biological processes as described in [6]. Another example of the use of spectral techniques is in the analysis of time-varying linear systems as described in [7].

In the remainder of this paper, mixed-radix transformations are considered and reviewed using transformation matrices formed as Kronecker products of elementary Walsh and Chrestenson matrices. It is also demonstrated that such transforms may be specified as a group character table over an appropriately ordered group and that the Cayley graph technique as described in [1,2] may be generalized for the mixed-radix case.

2. TRANSFORMATIONS OF MVL FUNCTIONS OVER AN ADDITIVE ABELIAN GROUP

The simplest case of a mixed-radix function where $q_i \geq 2$ is the binary-valued function $f(x_1, x_2)$ where $q_1 = 2$ and $q_2 = 3$. A straight forward extension of the techniques described in [1, 2] is to encode each minterm describing f as a unique element in $\mathbf{Z}_6$ and then to formulate the group character table for the additive Abelian group mod(6) and utilize this table as a transformation matrix. This involves mapping each minterm to one of six roots of unity in the set

$$\left\{e^{j2\pi\times\frac{0}{6}}, e^{j2\pi\times\frac{1}{6}}, e^{j2\pi\times\frac{2}{6}}, e^{j2\pi\times\frac{3}{6}}, e^{j2\pi\times\frac{4}{6}}, e^{j2\pi\times\frac{5}{6}}\right\}.$$

A graphical depiction of these points is shown in Fig. 1. Past work involving the definition of discrete transforms using finite groups is available in [9–11]. Such transforms have been considered in the past [8, 12] and referred to as the *generalized transform* or *generalized Fourier transform*;

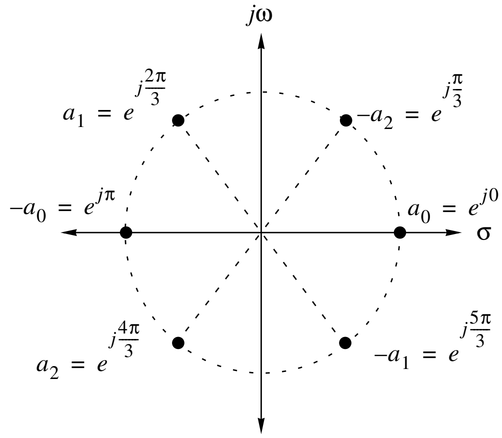

Fig. 1. Diagram of six roots of unity.

however, the formulation and relationship of these discrete function spectra to graph spectra offers a new viewpoint. It is noted that the formulation of the generalized Fourier transform typically includes the use of scaling constants referred to as "twiddling" factors. These are not included in the following without loss of generality.

A group character table is written with rows corresponding to irreducible representations and columns corresponding to conjugacy classes [9,13]. Traditionally the rows are labeled with χ_i which represent the irreducible representations as is done here. Each column (i.e., the conjugacy classes) is labeled with the notation as shown in Table 1. The purpose for this notation should become apparent later in this section of the paper.

Table 1. Group character table over $\mathbf{Z}_6$

	x_0	x_1	x_2	x_3	x_4	x_5
χ_0	a_0	a_0	a_0	a_0	a_0	a_0
χ_1	a_0	$-a_2$	a_1	$-a_0$	a_2	$-a_1$
χ_2	a_0	a_1	a_2	a_0	a_1	a_2
χ_3	a_0	$-a_0$	a_0	$-a_0$	a_0	$-a_0$
χ_4	a_0	a_2	a_1	a_0	a_2	a_1
χ_5	a_0	$-a_1$	a_2	$-a_0$	a_1	$-a_2$

Using the results of [5], a transformation matrix may be formulated using the rows of the group character table as shown in Table 1 as rows in a transformation matrix. This can be accomplished using the so-called "R-encoding" where the transformation matrix contains complex-valued elements or with "S-encoding" where both the transformation matrix and the function truth vector are encoded into complex values [2]. In both cases, a linear transformation, vector-matrix product may be formulated and computed. The resulting transformed function is then an arithmetic expression that represents the original function.

2.1. Kronecker Product Formulation

An alternative way of computing the transformation matrix for a function $f(x_1, x_2)$ where $x_1 \in \mathbf{Z}_2$ and $x_2 \in \mathbf{Z}_3$ is to utilize the Kronecker (or tensor) product [14] to combine the transformation matrices for the elementary additive Abelian group mod(2) and the elementary additive Abelian group mod(3) (e.g., note that these are also known as the Walsh and Chrestenson transformation matrices of functions of one variable). Equation (1) illustrates the computation of the transformation matrix T where $a_0 = e^{j2\pi\times\frac{0}{6}}$, $a_1 = e^{j2\pi\times\frac{2}{6}}$ and $a_2 = e^{j2\pi\times\frac{4}{6}}$

$$T = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \bigotimes \begin{bmatrix} a_0 & a_0 & a_0 \\ a_0 & a_1 & a_2 \\ a_0 & a_2 & a_1 \end{bmatrix}. \tag{1}$$

Carrying out the calculation in Eq. (1), the transformation matrix T becomes that as shown in Eq. (2)

$$T = \begin{bmatrix} a_0 & a_0 & a_0 & a_0 & a_0 & a_0 \\ a_0 & a_1 & a_2 & a_0 & a_1 & a_2 \\ a_0 & a_2 & a_1 & a_0 & a_2 & a_1 \\ a_0 & a_0 & a_0 & -a_0 & -a_0 & -a_0 \\ a_0 & a_1 & a_2 & -a_0 & -a_1 & -a_2 \\ a_0 & a_2 & a_1 & -a_0 & -a_2 & -a_1 \end{bmatrix}. \tag{2}$$

It is easily shown that the transformation matrix T is unique as follows.

Lemma 1. *The inverse of matrix T exists and it is orthogonal (with a scale factor of $\frac{1}{2^n\times 3^n}$).*

Proof. In Eq. (1), T is formed as the Kronecker product of the 2×2 Walsh transformation matrix W and the 3×3 Chrestenson transformation matrix C. Therefore the inverse of T is given as shown

$$T^{-1} = (W\otimes C)^{-1} = W^{-1}\otimes C^{-1}. \tag{3}$$

It is known that the inverses of W and C are given as follows

$$W^{-1} = \frac{1}{2^n}W, \tag{4}$$

$$C^{-1} = \frac{1}{3^n}C^*. \tag{5}$$

Therefore T^{-1} is given as shown in Eq. (6)

$$T^{-1} = \frac{1}{2^n\times 3^n}(W\otimes C^*). \tag{6}$$

□

Several other properties are noted with respect to the matrix T and are given in the following lemmas.

Lemma 2. *A linear transformation matrix formed using the rows and columns of the group character table given in Table 1 is identical to matrix T under a set of row and column permutations.*

Proof. Let G represent the matrix formed using the rows and columns of the group character table given in Table 1. Equation (7) holds where U_1 and U_2 are elementary permutation matrices

$$G = U_1TU_2. \tag{7}$$

The matrices U_1 and U_2 are given in Eqs. (8) and (9)

$$U_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$U_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (9)$$

□

This technique is computationally advantageous to the other methods presented here since the decomposition of the transformation matrix into a Kronecker product of smaller dimension matrices allows for "fast-transform" techniques to be applied [15, 16]. As has been done traditionally, a butterfly diagram may be used to characterize the "fast-transform." Such diagrams are described in detail in [17] where the notion of "signal flow graphs" is introduced. Butterfly diagrams are a signal flow graph where vertices represent summation operations and edges carry multiplicative weights. The butterfly diagram corresponding to Eq. (1) is shown in Fig. 2.

Alternatively, Eq. (1) may be rearranged using the permutation operations described in Lemma 2. When the permutation matrices are not included, the same spectral vector components result but in a different order. Equation (10) contains the relationship describing the Kronecker expansion of

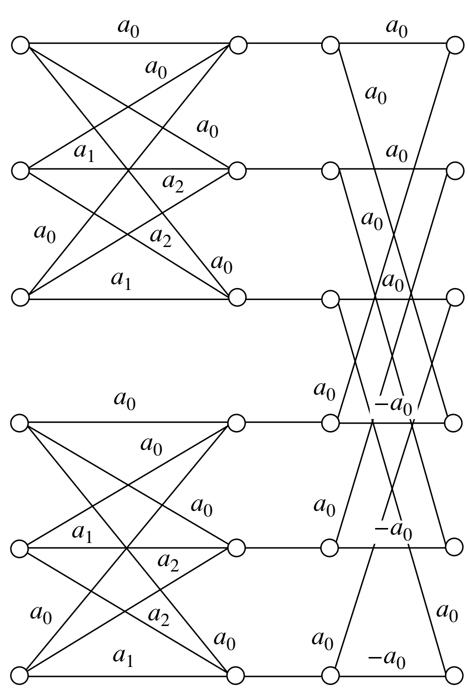

Fig. 2. Butterfly diagram of fast transform.

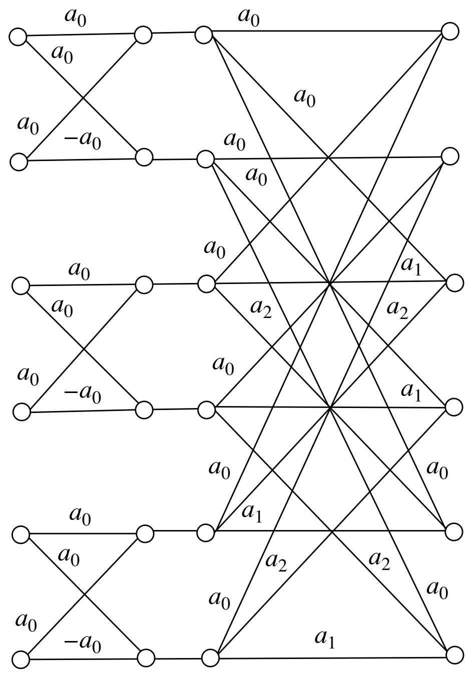

Fig. 3. Alternative butterfly diagram of fast transform.

the transformation matrix in this form and Fig. 3 is a an illustration of the corresponding butterfly diagram

$$T = \begin{bmatrix} a_0 & a_0 & a_0 \\ a_0 & a_1 & a_2 \\ a_0 & a_2 & a_1 \end{bmatrix} \bigotimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{10}$$

2.2. Cayley Graph Spectrum Formulation

To generalize the methods described in [1] for the binary-valued case and [2] for the discrete p-valued case where $p > 2$, a Cayley graph must be specified with an appropriate generator over the group $\mathbf{Z}_{q_0 \times q_1 \times \ldots \times q_{n-1}}$ where each variable x_i is q_i-valued.

The following definitions are used:

Definition 1. A *Cayley graph* represents an algebraic group $G = \{g_i, *\}$ and is denoted as $Cay(V, E)$ where V is the vertex set and E is the edge set. Each $v_i \in V$ uniquely corresponds to the group element g_i. Each $e_i \in E$ corresponds to a colored edge with some associated color g_i. The set of edges E is a subset of ordered pairs of elements in V that are *generated* by some binary operation(s) over elements in G. This binary operator need not be the same as the group product operator $*$.

In [1, 2] Cayley graphs were formed representing the elementary additive Abelian groups with generators that were the evaluation of some discrete function of the same group. The function argument was formulated as the digit-by-digit modulo-p difference of all possible minterms. For the binary case, the generator function yielding the adjacency matrix edge colors is $a_{ij} = f(m_i \oplus m_j)$ for all possible pairs of function domain values (m_i, m_j). Likewise for the non-binary p-valued case, adjacency matrix edge colors are generated as $a_{ij} = f(m_i \ominus_p m_j)$. $\ominus_p$ denotes the digit-by-digit difference of two minterms modulo-p.

The generalization to the mixed-radix case is quite natural. If each minterm is composed of various polarities of n different q-valued variables, then argument of the generator function f is formed as a concatenation of $(x_{i1} \ominus_{q_1} x_{j1}), (x_{i2} \ominus_{q_2} x_{j2}), \ldots, (x_{in} \ominus_{q_n} x_{jn})$. This operation will be denoted by the symbol $\ominus$ with no subscript, but with the understanding that it is applied digit-by-digit and modulo-q_i. In this case the generator is given in Eq. (11)

$$a_{ij} = f(m_i \ominus m_j). \tag{11}$$

2.3. Example Spectrum Computation

Using Eq. (11), the adjacency matrix for the Cayley graph over $\mathbf{Z}_6$ is given in Eq. (12)

$$A = \begin{bmatrix} f(0) & f(2) & f(1) & f(3) & f(5) & f(4) \\ f(1) & f(0) & f(2) & f(4) & f(3) & f(5) \\ f(2) & f(1) & f(0) & f(5) & f(4) & f(3) \\ f(3) & f(5) & f(4) & f(0) & f(2) & f(1) \\ f(4) & f(3) & f(5) & f(1) & f(0) & f(2) \\ f(5) & f(4) & f(3) & f(2) & f(1) & f(0) \end{bmatrix}. \tag{12}$$

As an example consider the mixed-radix function described in Table 2.

Table 2. Example function truth-table

x_1	x_2	X	f
0	0	0	0
0	1	1	1
0	2	2	1
1	0	3	0
1	1	4	1
1	2	5	0

The corresponding adjacency matrix is given as shown by Eq. (13) and an illustration of the Cayley graph is shown in Fig. 4,

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{13}$$

The characteristic equation of A is $C(\lambda) = \lambda^6 - 6\lambda^4 - 10\lambda^3 + 6\lambda + 9$. Solving for the roots of $C(\lambda)$, the spectrum of f is found to be $\lambda_i = \{a_0, 3a_0, a_1, a_2, 2a_1 + a_2, a_1 + 2a_2\}$. It is easily verified that the same set of spectral coefficients result when the truth vector of f is multiplied with the transformation matrix given in Eq. (2) or the one formed from Table 1.

2.3.1. Decision diagram method. The "ast-transform" methods described previously may also be implemented in a graphical manner using decision diagram data structures resulting in further savings [18, 19]. This formulation leads to a word-level graphical representation of the arithmetic expressions (i.e., the spectral function) of the transformed discrete mixed-radix function. Although the minimum number of required operations for obtaining the spectral representation of a discrete function is documented and has been studied in detail [20, 21], decision diagrams can offer an

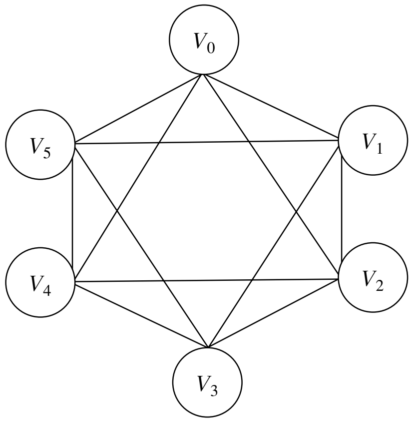

Fig. 4. Cayley graph of example function.

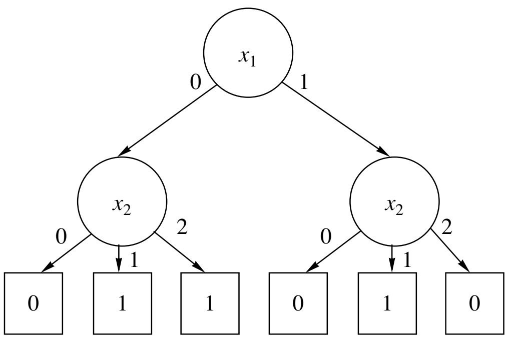

Fig. 5. Decision tree representation of example function.

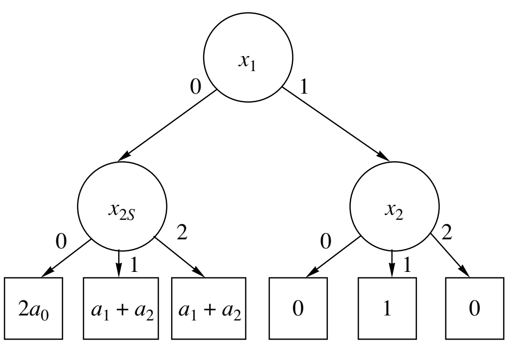

Fig. 6. Decision tree with one transformed vertex.

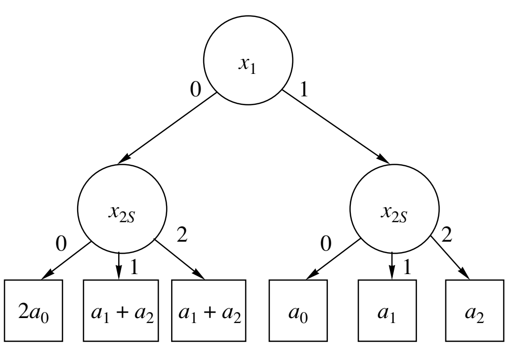

Fig. 7. Decision tree with ternary variables transformed.

advantage by avoiding the cases of addition with a constant-0 and multiplication by a constant-1 since these operations are implicit due to the reduction rules of the data structure. This technique can be viewed as an extension of that applied to the constant radix discrete functions as described in [22–24] resulting in a word-level decision diagram.

To illustrate the technique, we will first describe the computation methodology using a word-level decision tree which can be thought of as a non-reduced decision diagram. Note the tree will have internal vertices with different numbers of exiting edges. This is due to the mixed-radix function. Each internal vertex will have p edges where p corresponds to the total number of logic levels represented by each variable. Figure 5 contains a diagram of a decision tree representation of an example function.

The decision tree structure representing function f may be transformed into another decision tree representing the spectrum of f. This can be implemented as a depth-first traversal where the bottom-most vertices are transformed to the spectral domain first. As an example, the bottom vertices in Fig. 5 represent ternary-valued variables; hence, they must be transformed using the group table for the elementary additive Abelian group mod(3) (or, the ternary Chrestenson transformation matrix for a function of one variable). The Chrestenson transformation matrix for a function of one ternary variable is given in Eq. (14)

$$T_{C1} = \begin{bmatrix} a_0 & a_0 & a_0 \\ a_0 & a_1 & a_2 \\ a_0 & a_2 & a_1 \end{bmatrix}. \tag{14}$$

This will result in a tree as shown in Fig. 6 and is constructed by replacing the left-most group of terminal nodes with the following values (in respective order):

$$0 \times a_0 + 1 \times a_0 + 1 \times a_0 = 2a_0,$$
$$0 \times a_0 + 1 \times a_1 + 1 \times a_2 = a_1 + a_2,$$
$$0 \times a_0 + 1 \times a_2 + 1 \times a_1 = a_1 + a_2.$$

Note that the transformed vertex in Fig. 6 now is denoted by a subscript of S to denote that it represents a decomposition in terms of the matrix in Eq. (14). Performing the same transformation on the remaining vertex labeled x_2 yields a tree as depicted in Fig. 7.

Finally, the top variable of tree must be transformed yielding the decision tree representing the en-

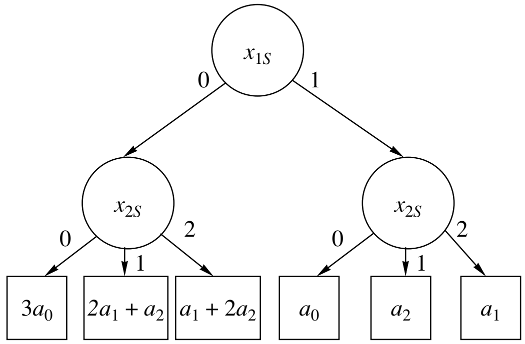

Fig. 8. Decision tree with ternary variables transformed.

tire spectrum. Because the top variable is binary-valued, the transformation matrix is formed from the character group table representing the elementary additive Abelian group mod(2) (or, the binary Hadamard-Walsh transformation matrix for a function of one variable). The Hadamard-Walsh transformation matrix for a function of one binary variable is given in Eq. (15). This results in replacing the leftmost tree with the sum of the left and right subtrees and the right-most subtree with the difference of the left versus the right subtree as given by the transformation matrix T_{HW1}. The word-level decision tree representing the transformed mixed-radix function is shown in Fig. 8,

$$T_{HW1} = \begin{bmatrix} a_0 & a_0 \\ a_0 & -a_0 \end{bmatrix}. \tag{15}$$

The spectral transformation over the decision tree illustrates the basic principles used to develop a decision diagram-based algorithm for spectral transformations. The key issue is the absence of vertices representing subsequent variables in the ordering of the reduced decision diagram [25]. Reduced decision diagrams have paths from the root (start) vertex to the terminal (labeled with a constant) that may "skip" variables. This feature is, in large part, responsible for their compactness. In transforming a decision diagram representing a discrete function into a spectral transformation, it is important to consider such "missing nodes" during the traversal.

During the postfix traversal of the decision diagram, it is important to maintain an order of the p-value of each variable so that the appropriate transformation matrix can be employed. Not surprisingly, the number of exiting edges from a non-terminal vertex is also the order of the square transformation matrix for the particular transformation of that graph node. The reduced decision diagram algorithm requires that an implicit "missing" node be inserted during the traversal, thus, allowing for the transformation to take place and perhaps resulting in the introduction of new non-terminal vertices in the resulting graph. Alternatively, the reduction rules allow for the elimination, and potentially, the reduction of the spectral decision graph size.

3. CONCLUSION

Spectral transformations of mixed-radix discrete functions result in an arithmetic expression uniquely representing the function as a weighted superposition of alternative basis functions. These arithmetic expressions can be conveniently represented using word-level decision diagrams and often offer compact representations in terms of required memory storage. Furthermore, the computation of the word-level decision diagram has a temporal complexity equivalent to that of the traditional "fast" transform methods which have been shown to utilize a minimal number of intermediate computations.

It has been shown that a particular spectral transformation of a discrete mixed-radix function over $\mathbf{Z}_6$ is equivalent to that over $\mathbf{Z}_2 \times \mathbf{Z}_3$ within a permutation. Also, it is shown that a Cayley graph may be formed over $\mathbf{Z}_6$ with a generator corresponding to the discrete function of interest. The spectrum of the Cayley graph is equivalent to the spectrum of the discrete mixed-radix function.

The motivation of this work is to provide additional ways of evaluating discrete multi-valued functions in the spectral domain so that practitioners in the design and analysis of automation and remote control of devices may have a new set of techniques for use in applications such as system identification and other tasks involving spectral and correlation analysis for system estimation.

ACKNOWLEDGMENTS

The author would like to thank Professor R. Stanković, Professor Claudio Moraga, and Professor D. Michael Miller for comments and suggestions related to this work.

REFERENCES

1. Bernasconi, A. and Codenotti, B., Spectral Analysis of Boolean Functions as a Graph Eigenvalue Problem, *IEEE Trans. Comp.*, 1999, no. 48, pp. 345–351.
2. Thornton, M.A., Miller, D.M., and Townsend, W., Chrestenson Spectrum Computation Using Cayley Color Graphs, *Int. Symp. on Multiple-Valued Logic*, 2002.
3. Chrestenson., H.E., A Class of Generalized Walsh Functions, *Pacific J. Math.*, 1955, no. 5, pp. 17–31.
4. Vilenkin, N.Ya., Concerning a Class of Complete Orthogonal Systems, *Dokl. Akad. Nauk SSSR, Ser. Math.*, 1947, no. 11.
5. Babai, L., Spectra of Cayley Graphs, *J. Combinatorial Theory, Series B*, 1979, no. 27, pp. 180–189.
6. Head, G.A., Lukoshkova, E.V., Burke, S.L., Malpas, S.C., Lambert, E.A., and Janssen, B.J.A., Comparing Spectral and Invasive Estimates of Baroreflex Gain, *IEEE Eng. Medicine Biology Magaz.*, 2001, no. 20(2), pp. 43–52.
7. Wirth, F., Dynamics of Time-Varying Discrete-Time Linear Systems: Spectral Theory and the Projected System, *SIAM J. Control Optimiz.*, 1998, no. 36(2), pp. 447–487.
8. Karpovsky, M.G., *Finite Orthogonal Series in the Design of Digital Devices*, New York: Wiley, 1976.
9. Pontryagin, L.S., *Nepreryvnye Gruppy* (Continuous Groups), Moscow: Ghostehizdat, 1954.
10. Garaev, M.U. and Faradzhev, R.G., On an Analog of Fourier Expansion over Galois Fields and Its Applications to Problems of Generalized Sequential Machines, *Izv. Akad. Nauk Azerb. SSR, Ser. Fiz.-Techn. Mat. Nauk*, 1968, no. 6, pp. 69–75.
11. Maslen, D.K. and Rockmore, D.N., Generalized FFTs—A Survey of Some Recent Results, *DIMACS Workshop in Groups and Computation*, 1995, pp. 183–238.
12. Ahmed, N. and Rao, K.R., *Orthogonal Transforms for Digital Signal Processing*, New York: Springer-Verlag, 1975.
13. Dummit, D.S. and Foote, R.M., *Abstract Algebra*, Upper Saddle River: Prentice Hall, 1999.
14. Graham, A., *Kronecker Products and Matrix Calculus with Applications*, New York: Wiley, 1981.
15. Cooley, J.W. and Tukey, J.W., An Algorithm for the Machine Calculation of Complex Fourier Series, *Math. Computation*, 1965, no. 19, pp. 297–301.
16. Shanks, J.L., Computation of the Fast Walsh-Fourier Transform, *IEEE Trans. Comp.*, 1969, no. 18, pp. 457–459.
17. Mason, S.J. and Zimmermann, H.J., *Electronic Circuits, Signals, and Systems*, Massachusetts: Massachusetts Inst. Techn., 1960.
18. Stanković, R.S., Sasao, T., and Moraga, C., Spectral Transforms Decision Diagrams, in *Representation of Discrete Functions*, Sasao, T. and Fujita, M., Eds., New York: Kluwer, 1996.

19. Thornton, M.A., Drechsler, R., and Miller, D.M., *Spectral Techniques in VLSI CAD*, New York: Kluwer, 2001.

20. Clausen, M., Fast Generalized Fourier Transforms, *Theor. Computer Sci.*, 1989, no. 67, pp. 55–63.

21. Maslen, D.K., The Efficient Computation of Fourier Transforms on the Symmetric Group, *Math. Comp.*, 1998, no. 67(223), pp. 1121–1147.

22. Miller, D.M., Graph Algorithms for the Manipulation of Boolean Functions and Their Spectra, in *Congressus Numerantium*, Winnipeg: Utilitas Mathematica, 1987, pp. 177–199.

23. Miller, D.M., Spectral Transformations of Multiple-Valued Decision Diagrams, *Int. Symp. on Multiple-Valued Logic*, 1994, pp. 89–96.

24. Thornton, M.A. and Drechsler, R., Spectral Decision Diagrams Using Graph Transformations, *Design, Automation, and Test in Europe*, 2001, pp. 713–717.

25. Bryant, R.E., Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Trans. Comp.*, 1986, no. 35(8), pp. 677–691.

This paper was recommended for publication by O.P. Kuznetsov, a member of the Editorial Board