WALSH TYPE TRANSFORMS FOR COMPLETELY AND INCOMPLETELY SPECIFIED MULTIPLE-VALUED INPUT BINARY FUNCTIONS

Bogdan J. Falkowski

Marek A. Perkowski

Department of Electrical Engineering,
Portland State University,
P.O. Box 751,
Portland, OR 97207.

ABSTRACT

Spectral representation of multiple-valued input binary functions is proposed for the first time. Such a representation is composed of a vector of Walsh transforms, each of them is defined for one pair of the input variables of the function. The new representation has the advantage of being real-valued having by that an easy interpretation. Since two types of codings of values of binary functions are used then two different spectra are introduced. The meaning of each spectral coefficient in classical logic terms is discussed. The mathematical relationships between the number of true, false, and don't care minterms and spectral coefficients are stated. These relationships can be used to calculate the spectral coefficients directly from the graphical representations of binary functions. Similarly to the spectral methods in classical logic design, the new spectral representation of binary functions can find applications in many problems of analysis, synthesis, and testing of circuits described by such functions.

Indexing terms: *Logic design, Sum-of-products expression, Completely and incompletely specified multiple-valued binary functions, Standard trivial functions, Orthogonal functions, False, true, don't care minterms, Walsh transforms, Spectral coefficients*

1. INTRODUCTION

Spectral techniques (Walsh transforms) in digital logic design have been used for more than thirty years. They have been used to classification of logic functions, analysis, synthesis, and fault detection of logic circuits [8]-[13]. It has been shown by many authors that some problems that are difficult to solve in the logic domain can be solved quite easily in the spectral domain [8]-[13], [18]. Hence, the interest in developing of new mathematical descriptions and transforms is growing and manifested by recent publications [5]-[7].

A multiple-valued input binary function is an extension of a Boolean function. The multiple-valued input binary functions find several applications in logic design, pattern recognition, and other areas [3], [14]-[17], [20]. In logic design, they are primarily used for the minimization of PLA's that have 2-bit decoders on the inputs [16]. A PLA with r-bit decoders implements directly a sum-of-products expression (SOPE) of a 2^r-valued input binary function. As shown in [15], every set of m Boolean functions, where each of them has n binary variables, can be represented as a $n + 1$ multiple-valued input binary function with n binary inputs and one m-valued input.

The choice of an orthogonal transform for a given problem is very important since it affects the complexity of the calculations in the spectral domain as well as the calculation of the forward and inverse transforms. Currently, the main tools that are used for Boolean and multiple-valued (on input as well as on output) functions are the following transforms : Walsh, Chrestenson, Haar, Watari, polynomial Fourier transform, number-theoretic transform and the generalized discrete Fourier transform [2], [11], [12]. In general, the interpretation of the meaning of spectral coefficients for all but Walsh above transforms is cumbersome or impossible. Only Walsh spectrum, that can be applied for Boolean functions, has interpretation in classical logic terms and the spectral coefficients can be calculated either from minterms or disjoint cube representation of the function [5]-[7]. Special transforms for multiple-valued input binary functions have never been proposed.

In [15] the concept of Mixed-Radix Exclusive Sum of Products was introduced for multiple-valued input binary functions that points out the usefulness of multiple-valued generalizations of Reed-Muller transforms. Similarly, in this paper Walsh-type spectra for multiple-valued input binary functions are proposed. There exist two Walsh spectra for Boolean functions: S and R [8], [9], [11]. Conventionally, the first spectrum is used for analysis and synthesis of Boolean functions while the second one is used in the design for testability of digital circuits [11]. It should be noticed, however, that each of these spectra can be used interchangeably with one another since they are linearly related [7], [9]. One can expect the similar applications of the transforms S and R introduced here for multiple-valued input binary functions.

The main reason for the introduction of such type of new transforms was for the authors the requirement of having the transform which has a minimal number of coefficients and an easy to understand interpretation. This paper introduces a new approach to spectral methods. First, the transform for a binary function of n multiple-valued variables is a vector of $\left\lceil \frac{n}{2} \right\rceil$ *partial transforms* of all pairs of these variables what minimizes the necessary spectral information to be kept about the function (partial coefficients). Secondly, the partial coefficients describe some global properties of the function and can be used not only to generate final coefficients but also by themselves. Thirdly, the interpretation of each partial transform is given in classical logic terms. By investigating links between spectral techniques and classical logic design methods this interesting area of research is presented in a simple manner. Moreover, an algorithm is shown for easily handling the calculation of partial spectral coefficients for completely and incompletely specified multiple-valued input binary functions. All mathematical relationships between the number of true, false, don't care minterms of binary functions and spectral coefficients are stated.

All presented investigations can be applied to any multiple-valued input binary function. In the case that such a function has more than two multiple-valued variables then either the presented below approach can be applied to them or the multidimensional Walsh transform can be used [2]. The latter approach is similar to pattern analysis and image processing and is not a subject of the consideration in this paper.

2. MULTIPLE-VALUED INPUT BINARY FUNCTIONS

A multiple-valued input binary function *(binary function* for short) is a mapping $f(X_1, X_2, \ldots, X_n): P_1 \times P_2 \times, \ldots, P_n \rightarrow B$, where X_i is a multiple-valued variable that takes the values from the set $P_i = \{0, 1, \ldots, p_i - 1\}$ and $B = \{0, 1, -\}$ (where $-$ denotes a don't care value). This is then the generalization of an ordinary n-input incompletely specified Boolean function $f: B^n \rightarrow B$.

A *literal* of multiple-valued input variable X_i, denoted by $X_i^{S_i}$, is defined as follows:

$$X_i^{S_i} = \begin{cases} 1 & \text{if } X_i \in S_i \\ 0 & \text{if } X_i \notin S_i \end{cases}$$

where $S_i \subseteq P_i$.

A *product of literals*, $X_1^{S_1}, X_2^{S_2}, \ldots, X_k^{S_k}$, ($k \le n$) is referred to as a *product term* (also called as *term* for short). A product term that includes literals for all function variables $X_1, X_2, X_3, \ldots, X_n$ is called a *full term*. A *minterm* of a multiple-valued input binary function is a full term in which every set S_i reduces to a single logical value. The logical function has value 1 for a *true minterm*, value 0 for a *false minterm* and is not specified for a *don't care minterm*. A sum of products is denoted as a *sum-of-products expression (SOPE)* while a product of sums is called as a *product-of-sums expression (POSE)*.

3. DEFINITIONS AND BASIC PROPERTIES OF TWO-DIMENSIONAL MAPS FOR MULTIPLE-VALUED INPUT BINARY FUNCTIONS

Two spectral representations S and R are introduced in the next paragraph for multiple-valued input binary functions. The analogous definitions exist for Boolean functions (see for example [7]-[9]). In order to apply new spectral representations to binary functions, they are first represented by the set of two-dimensional maps on which the Walsh type transforms are performed. The auxiliary algorithm allowing to present any multiple-valued input binary function by corresponding to it set of two-dimensional maps will be introduced.

The approach presented in this article is not the only one possible to find new spectral representations of the multiple-valued input binary functions. For example, instead of presenting each multiple-valued input binary function in the form of two-dimensional maps it is possible to use high-dimensional Hadamard matrices and transforms [1]. The presented approach enables, however, the application of two-dimensional transforms to two-dimensional maps and, by that, each spectral coefficient has an easy interpretation in logical terms. Moreover, the spectra can be calculated by known fast Walsh algorithms that are used in many areas from image processing to designing of logical circuits [1], [2], [9], [12].

The following symbols will be used. Let n denote the number of different variables of a completely or incompletely specified multiple-valued input binary function. Let p_m denote the number of different logical values that can be assigned to any of the variables. It is obvious, that there exists only one such p_m that is maximal for the entire set of input variables, and that each input variable can take a different number of logical values.

The following properties are valid for any multiple-valued input binary function and can be proved by mathematical induction.

Property 3.1: Any multiple-valued input binary function can be represented by $\left\lceil \frac{n}{2} \right\rceil$ two-dimensional maps where each map has, as the coordinates, two different variables from the set of all variables and the dimension of $p_i \times p_j$ where p_i and p_j are the maximal number of logical values that can be assumed by two different variables X_i and X_j, accordingly.

Property 3.2: A full term of a multiple-valued input binary function of n variables is represented by $\left\lceil \frac{n}{2} \right\rceil$ two-dimensional maps. The number of cells (areas on these map) that correspond to the full term can be found according to the formula $\sum_{j=1}^{n-1} \left[m_j \sum_{i=j+1}^{n} m_i \right]$ where m_i is the number of logical values of the $i-^{th}$ literal. This property is valid for any term. However, for variables that are not included in this term, one has to take the value of p_i instead of m_i in the above formula for each variable X_i.

Property 3.3: A minterm of a multiple-valued input binary function of n variables is represented by $\left\lceil \frac{n}{2} \right\rceil$ two-dimensional maps and on each of these maps the minterm is represented by one cell. Hence the number of cells that correspond to such a minterm is equal to the number of two-dimensional maps.

Property 3.4: The two-dimensional $p_m \times p_n$ maps described above can be transformed to partial spectral coefficients by an orthogonal Walsh-type transform without loss of any information if both numbers p_m and p_n are some powers of 2, possibly different.

Let us observe, that while in classical tables for multiple-valued logic [14], [15] each minterm corresponds to a cell, in the proposed representation each minterm is represented by a set of cells, one cell from each map. The last requirement (Property 3.4) for the dimension of the map representing the function is due to the known ways of generation of Hadamard matrices and requirements for the orthogonality of such matrices [19]. Since such a requirement would limit the possible number of different logical values for input variables then the more general approach is presented below.

Proposition 3.1: The two-dimensional $p_m \times p_n$ map is expanded to the two-dimensional $p_m^* \times p_n^*$ map where p_m and p_n are any integer values describing the multiplicity of logical values of input variables, $p_m^* = p_m + 4 - (p_m \bmod 4)$ and $p_n^* = p_n + 4 - (p_n \bmod 4)$ accordingly. When the map is expanded then all introduced cells have don't care values.

Proposition 3.2: The number of additional don't care cells that have to be added during the expansion process of the two-dimensional $p_m \times p_n$ map is equal to

a) $p_n [4 - (p_m \bmod 4)]$ when $(p_n \bmod 4) = 0$ and $(p_m \bmod 4) \neq 0$

b) $p_m [4 - (p_n \bmod 4)]$ when $(p_m \bmod 4) = 0$ and $(p_n \bmod 4) \neq 0$

c) $p_n [4 - (p_m \bmod 4)] + p_m [4 - (p_n \bmod 4)] + [4 - (p_m \bmod 4)] [4 - (p_n \bmod 4)]$ when both $(p_m \bmod 4) \neq 0$ and $(p_n \bmod 4) \neq 0$.

The following algorithm describes how an arbitrary multiple-valued input binary function can be represented by the set of two-dimensional maps. It is assumed in the following description that the function is represented in the *SOPE* form. The dual algorithm can be derived for the function represented in the *POSE* form. It is obvious, that the algorithm can be applied to the binary function represented in the form of cubes as well.

Algorithm 3.1: Transformation of multiple-valued input binary function in SOPE form to the set of two-dimensional matrices of any dimensions.

1. Set *nl* (number of literals in the binary function) and *nt* (number of terms in *SOPE* form).
2. For each pair of literals of the function create a two-dimensional $p_{m_i} \times p_{m_j}$ map where p_{m_i} and p_{m_j} are the maximal values of the $i-^{th}$ and $j-^{th}$ literal. The number of such maps is equal to $\begin{bmatrix} nl \\ 2 \end{bmatrix}$ (Property 3.1).
3. For each term from the *SOPE* expression enter the true values into some cells of the two-dimensional maps. The total number of such cells in the maps having true values can be found for each term according to Property 3.2. If step 3 has been performed for each term (i.e, *nt* times) then stop.

Hence, by using Algorithm 3.1, one can represent any multiple-valued input binary function in the form of the set of two-dimensional maps. In general, both dimensions of such maps are equal to any integer numbers. Due to Property 3.4, the additional algorithm converts a two-dimensional map to its equivalent (from the point of view of the logical function this map is representing) that has the dimensions equal to 2^j where j is any integer number.

Algorithm 3.2: Conversion of any two-dimensional map having one or two dimensions different from a power of 2 to its logical equivalent having the dimensions that are powers of 2.

1. Set the value k - the number of two-dimensional maps for a given multiple-valued input binary function.
2. For each map do:
   $k = k - 1$.
   If either p_m or p_n or both these values are not some powers of 2 then modify either p_m or p_n or both to either p_m^* or p_n^* or both according to Proposition 3.1.
   If there was any modification of the dimensions of the map then fill the expanded map's cells with don't cares, and the number of such cells can be calculated according to Proposition 3.2.
3. If $k = 0$ then stop otherwise go to step 2.

The following proposition deals with the dimension of the transform matrix that is necessary to calculate the spectrum of a two-dimensional map. In the case when Algorithm 3.2 has not expanded the original map's dimension (dimensions) then instead of p_m^* or p_n^* the original values p_m and p_n should be used in the following formula.

Proposition 3.3: The Hadamard-Walsh matrix T of the dimension $2^n \times 2^n$ can transform the two-dimensional map having the dimension $p_m^* \times p_n^*$ iff $n = \log_2 \left\lceil p_m^* \, p_n^* \right\rceil$.

The application of both algorithms will be shown in the following example.

Example 3.1: Consider the following three-variable multiple-valued input binary function:

$f\,1 = X^{\{0\}}\,Y^{\{1\}} + X^{\{1,2\}}\,Y^{\{0,3\}}\,Z^{\{1,2,3\}}$

Then, according to Algorithm 3.1 $nl = 3$ and $nt = 2$. The three two-dimensional maps for this binary function that correspond to each pair of the variables are shown in Fig. 1. (the areas on the maps filled with ones and zeros only). For instance, the term $X^{\{0\}}\,Y^{\{1\}}$ corresponds on the map $X\,Y$ to the cell $X^{\{0\}}\,Y^{\{1\}}$, on the map $Y\,Z$ to the cells $Y^{\{1\}}\,Z^{\{0,1,2,3\}}$, and on the map $X\,Z$ to the cells $X^{\{0\}}\,Z^{\{0,1,2,3\}}$. Then all these areas are filled with ones. Originally, according to Algorithm 3.1 the dimensions of these maps are 4×3, 4×4 and 3×4. After the application of Algorithm 3.2 the dimension of the first and third maps have increased to 4×4 and the expanded areas are filled with don't cares. The final result is shown in Fig. 1.

Hence, by using Algorithm 3.1 and Algorithm 3.2 one can always represent a multiple-valued input binary function in the form of a set of two-dimensional maps having the dimensions equal to powers of 2. In the next paragraph it will be shown how to apply Walsh-type transforms to these two-dimensional maps.

4. BASIC PROPERTIES OF HADAMARD-WALSH SPECTRA S AND R FOR TWO-DIMENSIONAL MAPS REPRESENTING MULTIPLE-VALUED INPUT BINARY FUNCTIONS

In order to shorten the notation and make it similar to that of the other authors, it is assumed that the symbol n that is used in the sequel confirms to the requirements of Proposition 3.3. The Hadamard-Walsh spectrum S of a two-dimensional map is an alternative representation of this map. When the map is represented as a vector V formed of the consecutive rows, then the Hadamard-Walsh spectrum S is formed from the multiplication of the <+1, 0, -1> vector representation V^S (corresponding to the original vector V for an incompletely specified map) by a $2^n \times 2^n$ Hadamard-Walsh matrix T [2], [9], [12]. In the coding scheme, the conventional <0, 1, -> values correspond to <+1, -1, 0> coding, respectively (- stands for a don't care). In the case of a completely specified two-dimensional map the conventional <0, 1> values correspond to <+1, -1> coding only.

If one keeps the original coding scheme then the alternative spectrum R can be defined. The Hadamard-Walsh spectrum R of a two-dimensional map is an alternative representation of this map. When the map is represented as a vector V formed of the consecutive rows, then the Hadamard-Walsh spectrum R is formed from the multiplication of the <0, 1, 0.5> vector representation V^R (corresponding to the original vector V for an incompletely specified map) by a $2^n \times 2^n$ Hadamard-Walsh matrix T [2], [9], [12]. In the coding scheme, the conventional <0, 1, -> values correspond to <0, 1, 0.5> coding, respectively.

The principal properties of the spectra R and S for two-dimensional maps are described below. It will be assumed without loss of generality that each map has two 4-valued variables as the coordinates denoted in this description as X - horizontal variable and Y - vertical variable, accordingly. Also, for the simplicity of the used notation, instead of using the full set notation for the description of multiple-valued literals only the members of the set will be denoted. For example, the literal $X^{\{1,3\}}$ will be described as $X^{1,3}$ - the same abbreviation in the notation for spectral coefficients will be used as well. When the properties of the spectral coefficients from both spectra S and R are the same then such properties will be given for the coefficients from the spectrum S only and this fact will be noticed in the description of a given property. When these properties differ then both spectra will be described separately.

Given below new properties of the spectra for two-dimensional map representation of the multiple-valued input binary functions can be proved in an analogous way to the proofs derived for Walsh transforms used in Boolean logic [8], [9], [11]. The used below names of transforms, refer, of course, to the *multiple-valued* counterparts of the respective transforms known in the literature as Walsh-Kaczmarz, Walsh-Paley, Rademacher-Walsh, and Hadamard-Walsh. Only these four basic orderings are compared. Although the transform matrices for each of these four basic orderings are the same for multiple-valued input binary functions and Boolean functions, the former are described by the vector of spectra of each two-dimensional map (where each of such maps is treated as a separate two-variable binary function), while the latter is described by only one spectrum (a Boolean function can be treated as only one two-variable binary function and be represented by only one two-dimensional map). Due to the lack of space, the properties of spectra of two-dimensional maps are only given but not proved and the detailed algorithms of calculation of these spectra are not included. The careful reader can, however, reconstruct these proofs from the orthogonality of transform matrices and the requirements on their dimensions. Since in the classical Boolean domain the Hadamard ordering is preferred over other orderings, all our examples are given for this ordering only.

4.1. The transform matrix is complete and orthogonal, and therefore, there is no information lost in the spectra S and R, concerning the cells of the map.

4.2. Only the Hadamard-Walsh matrix has the *recursive Kronecker product structure* [1], [2], [9], [21], and for this reason is preferred over other possible variants of the Walsh transform, known in the literature as Walsh-Kaczmarz, Rademacher-Walsh, and Walsh-Paley transforms.

4.3. Out of the four considered orderings of Walsh functions, only the Rademacher-Walsh transform is not *symmetric* ; all other variants of Walsh transform are symmetric, so that, disregarding a scaling factor, the same matrix can be used for both the forward and inverse transform operations.

4.4. When the classical matrix multiplication method is used to generate the spectral coefficients for different Walsh transforms, then the only difference is the order in which particular coefficients are created. The values of all these coefficients are the same for every Walsh transform.

4.5. Each spectral coefficient s_I (as well as r_I) gives a *correlation value* between the two-variable input binary function F corresponding to a given two-dimensional map and a *standard trivial function* u_I corresponding to this coefficient. The standard trivial functions for the spectral coefficients are, respectively, for the *dc coefficient* (direct current coefficient) - the universe of the function (where all cells on the map have true value) denoted by u_0; for the coefficients $s_{X^{1,2}}$, $s_{X^{2,3}}$, $s_{Y^{1,2}}$, $s_{Y^{2,3}}$ etc. (first order coefficients) - the literals $X^{1,2}$, $X^{2,3}$, $Y^{1,2}$, $Y^{2,3}$ of the binary function shown on the map and denoted by $u_{X^{i,j}}$, $u_{X^{j,k}}$, $u_{Y^{i,j}}$, $u_{Y^{j,k}}$; for the coefficients $s_{X^{1,2} \oplus Y^{2,3}}$, $s_{X^{2,3} \oplus Y^{1,2}}$, $s_{X^{2,3} \oplus Y^{2,3}}$, $s_{X^{1,2} \oplus Y^{1,2}}$, etc. (second order coefficients) - the exclusive-or function between literals $X^{1,2} \oplus Y^{2,3}$, $X^{2,3} \oplus Y^{1,2}$, $X^{2,3} \oplus Y^{2,3}$, $X^{1,2} \oplus Y^{1,2}$ of the binary function shown on the map and denoted by $u_{X^{i,j} \oplus Y^{j,k}}$ or by $u_{X^{i,j} \oplus Y^{i,j}}$. In all the formulas, i, j, and k are different integer numbers, $i = 1$, $j = 2$, $k = 3$. In short, the dc coefficient can be denoted by s_I $(I = 0)$, first order coefficients by s_{L^I} $(I = i, j, i \neq 0, j \neq 0, i \neq j$, and L is a literal), second order coefficients by $s_{L1^I \oplus L2^I}$ $(I = i, j, i \neq 0, j \neq 0, i \neq j$, and $L1, L2$ are two different literals).

4.6. The sum of all spectral coefficients of spectrum S for any completely specified two-dimensional map is $\pm 2^n$.

4.7. The sum of all spectral coefficients of spectrum S for any incompletely specified two-dimensional map is not $\pm 2^n$.

4.8. The maximal/minimal value of any individual spectral coefficient of spectrum S is $\pm 2^n$. This happens when the binary function represented on a given two-dimensional map is equal to either a standard trivial function u_I (sign +) or to its complement (sign -). In either case, all the remaining spectral coefficients have zero values because of the orthogonality of the transform matrix T.

4.9. The maximal/minimal value of any but r_0 individual spectral coefficient r_I is $\pm 2^{n-1}$. This happens when the binary function represented on a given two-dimensional map is equal to either a standard trivial function u_I (sign -) or to its complement (sign +). In either case, all but r_0 remaining spectral coefficients have zero values because of the orthogonality of the transform matrix T.

4.10. The maximal value of r_0 spectral coefficient is 2^n. It happens when all the cells of the two-dimensional map have the logical value 1.

4.11. Each but u_0 standard trivial function u_I corresponding to a two-dimensional map has the same number of true and false minterms equal to 2^{n-1}.

4.12. The spectrum S of each true cell of a two-dimensional map is given by $s_0 = 2^n - 2$, and all remaining $2^n - 1$ spectral coefficients s_I are equal to ± 2.

4.13. The spectrum S of each don't care cell of a two-dimensional map is given by $s_0 = 2^n - 1$, and all remaining $2^n - 1$ spectral coefficients s_I are equal to ± 1.

4.14. The spectrum S of each false minterm of a two-dimensional map is given by $s_I = 0$.

Example 4.1: An example of a spectrum R of a completely specified two-dimensional map is shown in Fig. 2. The spectrum S for the same map is shown in Fig. 3. The next example of a spectrum R of an incompletely specified two-dimensional map is shown in Fig. 4. The spectrum S for the same incompletely specified map is shown in Fig. 5. All the examples are taken from the Fig. 1 and represent two maps of the multiple-valued input binary function considered previously. The spectrum for the third map of this function can be calculated in a similar way.

Recursive algorithms, data flow-graph methods and parallel calculations similar to Fast Fourier Transform [1], [2], [9], [12], [21] can also be used to calculate the transforms introduced above.

5. LINKS BETWEEN SPECTRAL TECHNIQUES AND CLASSICAL LOGIC DESIGN

The material presented in this paragraph is valid not only for two-dimensional maps representing multiple-valued input binary functions but for Boolean functions and their Karnaugh maps representation as well. If the latter is the case then in all the following formulas n corresponds to the number of variables of the Boolean function and the two-dimensional map corresponds to a Karnaugh map rewritten from Gray-code to straight binary code (in the case of 4×4 dimension of the map rows and columns second and third have to be mutually interchanged). For multiple-valued input binary functions n fulfills the requirements of Proposition 3.3 and when both p_m^* and p_n^* from this proposition are equal then n represents the number of different logical values that can be assumed by each of the literals of the binary function. The meaning of all other symbols that are going to be introduced below is the same for both Boolean functions and two-dimensional maps.

Hence, let us show more clearly in the classical logic terms what is the real meaning of spectral coefficients for each map. The following symbols will be used. Let a_I be the number of true cells in the two-dimensional map, where both the map and the standard trivial function u_I have the logical values 1; let b_I be the number of false cells in the two-dimensional map, where the map has the logical value 0 and the standard trivial function u_I has the logical value 1; let c_I be the number of true cells in the two-dimensional map, where the map has the logical value 1 and the standard trivial function u_I has the logical value 0; let d_I be the number of false cells in the two-dimensional map, where both the map and the standard trivial function u_I have the logical values 0, let e_I be the number of don't care cells in the two-dimensional map, where the standard trivial function u_I has the logical value 1, and f_I be the number of don't care minterms of two-dimensional map, where the standard trivial function u_I has the logical value 0. Then, for completely specified two-dimensional map having dimension $n \times n$, these formulas hold:

$$a_I + b_I + c_I + d_I = 2^n$$

and

$$a_I + b_I = c_I + d_I = 2^{n-1}.$$

Accordingly, for incompletely specified two-dimensional map having dimension $n \times n$, hold:

$$a_I + b_I + c_I + d_I + e_I + f_I = 2^n$$

and

$$a_I + b_I + e_I = c_I + d_I + f_I = 2^{n-1}.$$

The s_I spectral coefficients for a completely specified two-dimensional map can be defined in the following way :

$s_I = 2^n - 2 \times a_I$, when $I = 0$,

$s_I = 2 \times (a_I + d_I) - 2^n$, when $I \neq 0$.

The spectral coefficients for an incompletely specified two-dimensional map can be defined in the following way :

$s_I = 2^n - 2 \times a_I - e_I$, when $I = 0$

and

$s_I = 2 \times (a_I + d_I) + e_I + f_I - 2^n$, when $I \neq 0$.

As one can see, for the case when both $e_I = 0$, and $f_I = 0$, i.e., for the completely specified two-dimensional map the above formulas reduce to the formulas presented previously. And again, by easy mathematical transformations, one can define all but s_0 spectral coefficients in the following way:

$$s_I = 2 \times (a_I + d_I) + e_I + f_I - 2^n =$$
$$2 \times (a_I + d_I) + e_I + f_I - (a_I + b_I + c_I + d_I + e_I + f_I) =$$
$$(a_I + d_I) - (b_I + c_I), \text{ when } I \neq 0.$$

Simultaneously, the s_0 spectral coefficient can be rewritten in the following way:

$$s_I = 2^n - 2 \times a_I - e_I =$$
$$a_I + b_I + c_I + d_I + e_I + f_I - 2 \times a_I - e_I =$$
$$b_I + c_I + d_I + f_I - a_I = b_I - a_I,$$

since for $I = 0$, c_I, d_I, and f_I are always equal to 0.

Thus, in the final formulas, describing all s_I spectral coefficients, the number of don't care minterms $e_I + f_I$ can be eliminated from them. Moreover, the final formulas are exactly the same as the ones for the completely specified two-dimensional map. Of course, it does not mean that the spectral coefficients for the incompletely specified two-dimensional map do not depend on the number of don't care minterms. They do depend on those numbers, but the problem is already taken into account in the last two formulas themselves. Simply, the previously stated formula for the numbers a_I, b_I, c_I, d_I, e_I, and f_I bonds all these values together.

Let us show now the meanings of r_I spectral coefficients. The meanings of all the symbols a_I, b_I, c_I, d_I, e_I, and f_I are exactly the same as described previously.

The r_I spectral coefficients for a completely specified two-dimensional map can be defined in the following way :

$r_I = a_I$, when $I = 0$,

$r_I = c_I - a_I$, when $I \neq 0$.

The spectral coefficients for an incompletely specified two-dimensional map can be defined in the following way :

$$r_I = a_I + \frac{e_I}{2}, \text{ when } I = 0$$

and

$$r_I = c_I - a_I + \frac{f_I - e_I}{2}, \text{ when } I \neq 0.$$

As one can see, for the case when $e_I = 0$ and $f_I = 0$, i.e., for the completely specified two-dimensional map, the above formulas reduce to the formulas presented previously for r_I spectral coefficients.

The application of the above formulas will be shown in the following examples (all examples are for the spectra in Hadamard-Walsh order).

Example 5.1: Consider the completely specified two-dimensional map describing the relationship between four-valued variables Y and Z for the binary function from Fig. 1. All standard trivial functions and the corresponding values of a_I, c_I, and d_I for this map are shown in Fig. 6.

The spectrum R for this map can be calculated as follows:

$r_0 = a_0$,

and

$r_I = c_I - a_I$, when $I \neq 0$.

$r_0 = 10$, $r_{z^{2,3}} = 4 - 6 = -2$,
$r_{z^{1,3}} = 4 - 6 = -2$, $r_{z^{1,2}} = 4 - 6 = -2$,
$r_{y^{1,3}} = 3 - 7 = -4$, $r_{z^{2,3} \oplus y^{1,3}} = 5 - 5 = 0$,
$r_{z^{1,3} \oplus y^{1,3}} = 5 - 5 = 0$, $r_{z^{1,2} \oplus y^{1,3}} = 5 - 5 = 0$,
$r_{y^{2,3}} = 7 - 3 = 4$, $r_{z^{2,3} \oplus y^{2,3}} = 5 - 5 = 0$,
$r_{z^{1,3} \oplus y^{2,3}} = 5 - 5 = 0$, $r_{z^{1,2} \oplus y^{2,3}} = 5 - 5 = 0$,
$r_{y^{1,2}} = 6 - 4 = 2$, $r_{z^{2,3} \oplus y^{1,2}} = 4 - 6 = -2$,
$r_{z^{1,3} \oplus y^{1,2}} = 4 - 6 = -2$, $r_{z^{1,2} \oplus y^{1,2}} = 4 - 6 = -2$.

The spectrum S for this map can be calculated as follows:

$s_0 = 2^n - 2 \times a_0$,

and

$s_I = 2 \times (a_I + d_I) - 2^n$, when $I \neq 0$.

$s_0 = 16 - 20 = -4$, $s_{z^{2,3}} = 20 - 16 = 4$,
$s_{z^{1,3}} = 20 - 16 = 4$, $s_{z^{1,2}} = 20 - 16 = 4$,
$s_{y^{1,3}} = 24 - 16 = 8$, $s_{z^{2,3} \oplus y^{1,3}} = 16 - 16 = 0$,
$s_{z^{1,3} \oplus y^{1,3}} = 16 - 16 = 0$, $s_{z^{1,2} \oplus y^{1,3}} = 16 - 16 = 0$,
$s_{y^{2,3}} = 8 - 16 = -8$, $s_{z^{2,3} \oplus y^{2,3}} = 16 - 16 = 0$,
$s_{z^{1,3} \oplus y^{2,3}} = 16 - 16 = 0$, $s_{z^{1,2} \oplus y^{2,3}} = 16 - 16 = 0$,
$s_{y^{1,2}} = 12 - 16 = -4$, $s_{z^{2,3} \oplus y^{1,2}} = 20 - 16 = 4$,
$s_{z^{1,3} \oplus y^{1,2}} = 20 - 16 = 4$, $s_{z^{1,2} \oplus y^{1,2}} = 20 - 16 = 4$.

As one can find out, the obtained spectra R and S are exactly the same as the ones calculated by the classical method shown in Fig. 2 and Fig. 3.

Example 5.2: Consider the incompletely specified two-dimensional map describing the relationship between four-valued variables Y and X for the binary function from Fig. 1. All standard trivial functions and the corresponding values of a_I, c_I, d_I, e_I, and f_I for this map are shown in Fig. 7.

The spectrum R for this map can be calculated as follows:

$$r_I = a_I + \frac{e_I}{2}, \text{ when } I = 0$$

and

$r_I = c_I - a_I + \frac{f_I - e_I}{2}$, when $I \neq 0$.

$r_0 = 5 + 2 = 7$, $r_{x^{2,3}} = 3 - 2 + \frac{0-4}{2} = 1 - 2 = -1$,
$r_{x^{1,3}} = 1 - 2 = -1$, $r_{x^{1,2}} = -3 + 2 = -1$,
$r_{y^{1,3}} = -1$, $r_{x^{2,3} \oplus y^{1,3}} = -1$,
$r_{x^{1,3} \oplus y^{1,3}} = -1$, $r_{x^{1,2} \oplus y^{1,3}} = -1$,
$r_{y^{2,3}} = 1$, $r_{x^{2,3} \oplus y^{2,3}} = 1$,
$r_{x^{1,3} \oplus y^{2,3}} = 1$, $r_{x^{1,2} \oplus y^{2,3}} = 1$,
$r_{y^{1,2}} = 3$, $r_{x^{2,3} \oplus y^{1,2}} = -1$,
$r_{x^{1,3} \oplus y^{1,2}} = -1$, $r_{x^{1,2} \oplus y^{1,2}} = -5$.

The spectrum S for this map can be calculated as follows:

$s_I = 2^n - 2 \times a_I - e_I - f_I$, when $I = 0$

and

$s_I = 2 \times (a_I + d_I) + e_I + f_I - 2^n$, when $I \neq 0$.

$s_0 = 16 - 10 - 4 = 2$, $s_{x^{2,3}} = 14 + 4 - 16 = 2$,
$s_{x^{1,3}} = 14 + 4 - 16 = 2$, $s_{x^{1,2}} = 14 + 4 - 16 = 2$,
$s_{y^{1,3}} = 14 + 4 - 16 = 2$, $s_{x^{2,3} \oplus y^{1,3}} = 14 + 4 - 16 = 2$,
$s_{x^{1,3} \oplus y^{1,3}} = 14 + 4 - 16 = 2$, $s_{x^{1,2} \oplus y^{1,3}} = 14 + 4 - 16 = 2$,
$s_{y^{2,3}} = 10 + 4 - 16 = -2$, $s_{x^{2,3} \oplus y^{2,3}} = 10 + 4 - 16 = -2$,
$s_{x^{1,3} \oplus y^{2,3}} = 10 + 4 - 16 = -2$, $s_{x^{1,2} \oplus y^{2,3}} = 10 + 4 - 16 = -2$,
$s_{y^{1,2}} = 6 + 4 - 16 = -6$, $s_{x^{2,3} \oplus y^{1,2}} = 14 + 4 - 16 = 2$,
$s_{x^{1,3} \oplus y^{1,2}} = 14 + 4 - 16 = 2$, $s_{x^{1,2} \oplus y^{1,2}} = 22 + 4 - 16 = 10$.

As one can find out, the obtained spectra R and S are exactly the same as the ones calculated by the classical method shown in Fig. 4. and Fig. 5.

As the final example the spectra S and R for the third two-dimensional map representing the binary function f will be shown. This time, the calculations are not shown but can be performed by any of the methods already presented.

Example 5.3: Consider the incompletely specified two-dimensional map describing the relationship between four-valued variables X and Z for the binary function from Fig. 1.

The spectrum R for this map is as follows:

$r_0 = 12$, $r_{z^{2,3}} = -2$,
$r_{z^{1,3}} = -2$, $r_{z^{1,2}} = -2$,
$r_{x^{1,3}} = 2$, $r_{z^{2,3} \oplus x^{1,3}} = 0$,
$r_{z^{1,3} \oplus x^{1,3}} = 0$, $r_{z^{1,2} \oplus x^{1,3}} = 0$,
$r_{x^{2,3}} = 2$, $r_{z^{2,3} \oplus x^{2,3}} = 0$,
$r_{z^{1,3} \oplus x^{2,3}} = 0$, $r_{z^{1,2} \oplus x^{2,3}} = 0$,
$r_{x^{1,2}} = 0$, $r_{z^{2,3} \oplus x^{1,2}} = 2$,
$r_{z^{1,3} \oplus x^{1,2}} = 2$, $r_{z^{1,2} \oplus x^{1,2}} = 2$.

The spectrum S for this map is as follows:

$s_0 = -8$, $s_{z^{2,3}} = 4$,
$s_{z^{1,3}} = 4$, $s_{z^{1,2}} = 4$,
$s_{x^{1,3}} = -4$, $s_{z^{2,3} \oplus x^{1,3}} = 0$,
$s_{z^{1,3} \oplus x^{1,3}} = 0$, $s_{z^{1,2} \oplus x^{1,3}} = 0$,
$s_{x^{2,3}} = -4$, $s_{z^{2,3} \oplus x^{2,3}} = 0$,
$s_{z^{1,3} \oplus x^{2,3}} = 0$, $s_{z^{1,2} \oplus x^{2,3}} = 0$,
$s_{x^{1,2}} = 0$, $s_{z^{2,3} \oplus x^{1,2}} = -4$,
$s_{z^{1,3} \oplus x^{1,2}} = -4$, $s_{z^{1,2} \oplus x^{1,2}} = -4$.

Then the multiple-valued input binary function from the Fig. 1 is represented by a vector which is composed of three sets of partial spectral coefficients. The values of all three spectra are given in Example 5.1, Example 5.2, and Example 5.3, respectively.

6. CONCLUSION

A new concept of a spectral transform for a multiple-valued input binary function has been introduced. Such transform is composed of a vector of transforms of all pairs of the function input variables.

The class of these new transforms corresponds to the well-known transforms for Boolean logic and can find analogical to them applications in classification, analysis, synthesis, design for testability and test generation of multiple-valued input binary functions. Such functions have been implemented as PLA's with input-variable decoders [16], PLA's with programmable encoders [20], Mixed-Radix Exclusive Sums of Products [15], or multiple-level functions [17] and have found applications in state assignment and synthesis of any type of multiple-output Boolean functions. Since the spectral methods for Boolean functions have been used successfully to realize the PLA's, multi-level circuits, and circuits with EXOR gates and due to the fact that the multiple-valued input binary functions are generalizations of the Boolean functions, it seems natural that the spectral transforms for the multiple-valued input binary functions will find applications in analysis, synthesis and testing of all the circuits mentioned above.

Classical Walsh transforms have applications to the design with multiplexers [13], decomposition and design with EXOR pre-processing and post-processing circuits [10]-[12], [18]. One can expect that the similar applications can be found for multiple-valued technologies described above. Our interpretation of spectral coefficients from Section 5 is not only useful for hand calculations of coefficients but, what is even more important, it helps to formulate new theorems and algorithms in spectral domain by analogy to the ones in classical domain. Some of new developments deal with decomposition and testing of circuits described by multiple-valued input binary functions.

It would be also interesting to investigate relations of new transforms with multidimensional transforms used in image coding and application of the new transforms in image processing. There are also possible other formulations of transforms for multiple-valued input binary functions that do not use complex numbers as the coefficients. The work in these areas as well as formulations of the mutual relationships between different kinds of transforms are the topics of current research of the authors.

7. ACKNOWLEDGMENT

The authors wish to especially thank the anonymous referees for numerous helpful comments. This research was financially supported in part by the Faculty Development Grant 90-050-1009-2000 from Portland State University.

8. REFERENCES

[1] S. S. Agaian, *Hadamard Matrices and Their Applications.* Lect. Notes in Mathematics, No. 1168, Berlin: Springer-Verlag, 1980.

[2] K. G. Beauchamp, *Applications of Walsh and Related Functions.* New York: Academic Press, 1984.

[3] M. J. Ciesielski, S. Yang, and M. A. Perkowski, "Multiple-valued minimization based on graph coloring," *Proc. of IEEE Int. Conf. Computer Design (ICCD),* pp. 262-265, 1989.

[4] P. Davio, J. P. Deschamps, and A. Thayse, *Discrete and Switching Functions.* New York: George and McGraw-Hill, 1978.

[5] B. J. Falkowski, and M. A. Perkowski, "Algorithms for the calculation of Hadamard-Walsh spectrum for completely and incompletely specified Boolean functions," *Proc. of IEEE Int. Phoenix Conf. Comp. & Communications (IPCCC),* Scottsdale, AR, March 1990.

[6] B. J. Falkowski, and M. A. Perkowski, "Yet another method for the calculation of Hadamard-Walsh spectrum for com-

pletely and incompletely specified Boolean functions," *Accepted for publication in Int. Journal of Electronics*, 1990.

[7] B. J. Falkowski, and M. A. Perkowski, "Essential relations between classical and spectral approaches to analysis, synthesis, and testing of completely and incompletely specified Boolean functions," *Proc. of IEEE Int. Symp. Circ. & Systems (ISCAS)*, New Orleans, LA, May 1990.

[8] S. L. Hurst, *The Logical Processing of Digital Signals*. New York: Crane-Russak, 1978.

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

[10] Hurst, S.L., "Use of linearization and spectral techniques in input and output compaction testing of digital networks," *IEE Proc. E*, vol. 136, No. 1, pp. 48-56, 1989.

[11] M. G. Karpovsky, *Finite Orthogonal Series in Design of Digital Devices*. New York: Wiley, 1976.

[12] M. G. Karpovsky, (ed.), *Spectral Techniques and Fault Detection*. Orlando: Academic Press, 1985.

[13] A. M. Lloyd, "Design of multiplexer universal-logic-module networks using spectral techniques," *IEE Proc. E*, vol. 127, pp. 31-36, 1980.

[14] M. A. Perkowski, P. Wu, and K. A. Pirkl, "KUAI-EXACT: A new approach for multi-valued logic minimization in VLSI synthesis," *Proc. of IEEE Int. Symp. Circ. & Systems (ISCAS)*, Portland, OR, pp. 401-404, May 1989. No. 38, 1988.

[15] M. A. Perkowski, M. Helliwell, and P. Wu, "Minimization of multiple-valued input multi-output mixed-radix exclusive sums of products for incompletely specified Boolean functions," *Proc. of 19-th Int. Symp. Multiple-Valued Logic (ISMVL)*, pp. 256-263, 1989.

[16] T. Sasao, "An application of multiple-valued logic to a design of programmable logic arrays," *Proc. of 8th Int. Symp. Multiple-Valued Logic (ISMVL)*, 1978.

[17] T. Sasao, "MACDAS: Multi-level AND-OR circuit synthesis using two-variable function generators," *Proc. 23rd Design Automation Conference (DAC)*, pp. 86-93, 1986.

[18] D. Varma, and E.A. Trachtenberg, "Design automation tools for efficient implementation of logic functions by decomposition," *IEEE Trans. CAD*, vol. 8, no. 8, pp. 901-916, Aug. 1989.

[19] W. D. Wallis, A.P. Street, and J.S. Wallis, *Combinatorics: Room Squares, Sum-Free Sets, Hadamard Matrices*. Lect. Notes in Mathematics No. 292, Berlin: Springer-Verlag, 1972.

[20] S. Yang, and M.J. Ciesielski, "A generalized PLA decomposition with programmable encoders," *Proc. of Int. Workshop on Logic Synthesis*, 1989.

[21] L. P. Yaroslavsky, *Digital Picture Processing - An Introduction*. Berlin: Springer-Verlag, 1985.

Y\X	0	1	2	3
0	0	1	1	-
1	1	0	0	-
2	0	0	0	-
3	0	1	1	-

Y\Z	0	1	2	3
0	0	1	1	1
1	1	1	1	1
2	0	0	0	0
3	0	1	1	1

X\Z	0	1	2	3
0	1	1	1	1
1	0	1	1	1
2	0	1	1	1
3	-	-	-	-

$$f = X^{1,2}\, Y^{0,3}\, Z^{1,2,3} + X^{0}\, Y^{1}$$

Fig. 1. A set of two-dimensional maps representing three-variable multiple-valued input binary function f.

$$
\begin{array}{ccccc}
T & V^R & & R & \\
\left[\begin{array}{cccccccccccccccc}
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 \\
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 \\
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 \\
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{array}\right]
&
\left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{array}\right]
& = &
\left[\begin{array}{c} 10 \\ -2 \\ -2 \\ -2 \\ -4 \\ 0 \\ 0 \\ 0 \\ 4 \\ 0 \\ 0 \\ 0 \\ 2 \\ -2 \\ -2 \\ -2 \end{array}\right]
&
\begin{array}{l}
R_0 \\
R_{Z^{\{2,3\}}} \\
R_{Z^{\{1,3\}}} \\
R_{Z^{\{1,2\}}} \\
R_{Y^{\{1,3\}}} \\
R_{Z^{\{2,3\}} \oplus Y^{\{1,3\}}} \\
R_{Z^{\{1,3\}} \oplus Y^{\{1,3\}}} \\
R_{Z^{\{1,2\}} \oplus Y^{\{1,3\}}} \\
R_{Y^{\{2,3\}}} \\
R_{Z^{\{2,3\}} \oplus Y^{\{2,3\}}} \\
R_{Z^{\{1,3\}} \oplus Y^{\{2,3\}}} \\
R_{Z^{\{1,2\}} \oplus Y^{\{2,3\}}} \\
R_{Y^{\{1,2\}}} \\
R_{Z^{\{2,3\}} \oplus Y^{\{1,2\}}} \\
R_{Z^{\{1,3\}} \oplus Y^{\{1,2\}}} \\
R_{Z^{\{1,2\}} \oplus Y^{\{1,2\}}}
\end{array}
\end{array}
$$

Fig. 2. Spectrum R for the completely specified two-dimensional map of Z and Y variables from Fig. 1.

$$
\begin{array}{ccccc}
T & V^S & & S & \\
\left[\begin{array}{cccccccccccccccc}
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 \\
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 \\
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 \\
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{array}\right]
&
\left[\begin{array}{c} 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{array}\right]
& = &
\left[\begin{array}{c} -4 \\ 4 \\ 4 \\ 4 \\ 8 \\ 0 \\ 0 \\ 0 \\ -8 \\ 0 \\ 0 \\ 0 \\ -4 \\ 4 \\ 4 \\ 4 \end{array}\right]
&
\begin{array}{l}
S_0 \\
S_{Z^{\{2,3\}}} \\
S_{Z^{\{1,3\}}} \\
S_{Z^{\{1,2\}}} \\
S_{Y^{\{1,3\}}} \\
S_{Z^{\{2,3\}} \oplus Y^{\{1,3\}}} \\
S_{Z^{\{1,3\}} \oplus Y^{\{1,3\}}} \\
S_{Z^{\{1,2\}} \oplus Y^{\{1,3\}}} \\
S_{Y^{\{2,3\}}} \\
S_{Z^{\{2,3\}} \oplus Y^{\{2,3\}}} \\
S_{Z^{\{1,3\}} \oplus Y^{\{2,3\}}} \\
S_{Z^{\{1,2\}} \oplus Y^{\{2,3\}}} \\
S_{Y^{\{1,2\}}} \\
S_{Z^{\{2,3\}} \oplus Y^{\{1,2\}}} \\
S_{Z^{\{1,3\}} \oplus Y^{\{1,2\}}} \\
S_{Z^{\{1,2\}} \oplus Y^{\{1,2\}}}
\end{array}
\end{array}
$$

Fig. 3. Spectrum S for the completely specified two-dimensional map of Z and Y variables from Fig. 1.

$$
\overset{T}{\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{bmatrix}}
\overset{V^R}{\begin{bmatrix} 0\\ 1\\ 1\\ 0.5\\ 1\\ 0\\ 0\\ 0.5\\ 0\\ 0\\ 0\\ 0.5\\ 0\\ 1\\ 1\\ 0.5 \end{bmatrix}}
=
\overset{R}{\begin{bmatrix} 7\\ -1\\ -1\\ -1\\ -1\\ -1\\ -1\\ -1\\ 1\\ 1\\ 1\\ 1\\ 3\\ -1\\ -1\\ -5 \end{bmatrix}}
\begin{array}{l}
R_0\\
R_{X^{\{2,3\}}}\\
R_{X^{\{1,3\}}}\\
R_{X^{\{1,2\}}}\\
R_{Y^{\{1,3\}}}\\
R_{X^{\{2,3\}}} \oplus Y^{\{1,3\}}\\
R_{X^{\{1,3\}}} \oplus Y^{\{1,3\}}\\
R_{X^{\{1,2\}}} \oplus Y^{\{1,3\}}\\
R_{Y^{\{2,3\}}}\\
R_{X^{\{2,3\}}} \oplus Y^{\{2,3\}}\\
R_{X^{\{1,3\}}} \oplus Y^{\{2,3\}}\\
R_{X^{\{1,2\}}} \oplus Y^{\{2,3\}}\\
R_{Y^{\{1,2\}}}\\
R_{X^{\{2,3\}}} \oplus Y^{\{1,2\}}\\
R_{X^{\{1,3\}}} \oplus Y^{\{1,2\}}\\
R_{X^{\{1,2\}}} \oplus Y^{\{1,2\}}
\end{array}
$$

Fig. 4. Spectrum R for the incompletely specified two-dimensional map of X and Y variables from Fig. 1.

$$
\overset{T}{\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{bmatrix}}
\overset{V^S}{\begin{bmatrix} 1\\ -1\\ -1\\ 0\\ -1\\ 1\\ 1\\ 0\\ 1\\ 1\\ 1\\ 0\\ 1\\ -1\\ -1\\ 0 \end{bmatrix}}
=
\overset{S}{\begin{bmatrix} 2\\ 2\\ 2\\ 2\\ 2\\ 2\\ 2\\ 2\\ -2\\ -2\\ -2\\ -2\\ -6\\ 2\\ 2\\ 10 \end{bmatrix}}
\begin{array}{l}
S_0\\
S_{X^{\{2,3\}}}\\
S_{X^{\{1,3\}}}\\
S_{X^{\{1,2\}}}\\
S_{Y^{\{1,3\}}}\\
S_{X^{\{2,3\}}} \oplus Y^{\{1,3\}}\\
S_{X^{\{1,3\}}} \oplus Y^{\{1,3\}}\\
S_{X^{\{1,2\}}} \oplus Y^{\{1,3\}}\\
S_{Y^{\{2,3\}}}\\
S_{X^{\{2,3\}}} \oplus Y^{\{2,3\}}\\
S_{X^{\{1,3\}}} \oplus Y^{\{2,3\}}\\
S_{X^{\{1,2\}}} \oplus Y^{\{2,3\}}\\
S_{Y^{\{1,2\}}}\\
S_{X^{\{2,3\}}} \oplus Y^{\{1,2\}}\\
S_{X^{\{1,3\}}} \oplus Y^{\{1,2\}}\\
S_{X^{\{1,2\}}} \oplus Y^{\{1,2\}}
\end{array}
$$

Fig. 5. Spectrum S for the incompletely specified two-dimensional map of X and Y variables from Fig. 1.

Note: $P_i = \{0,1,2,3\}$

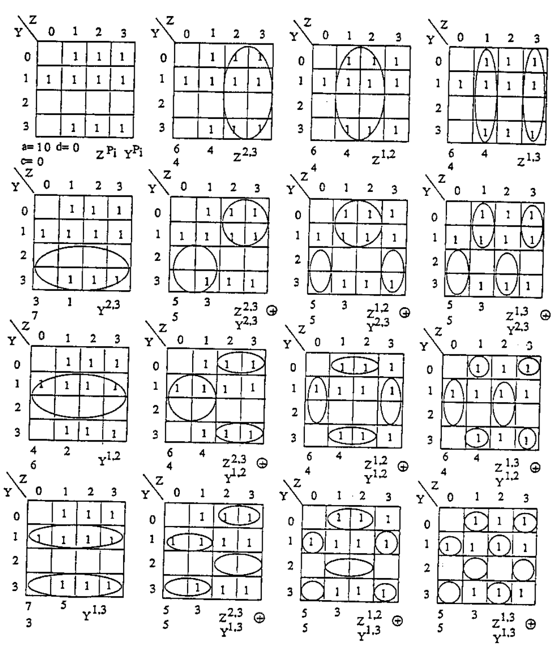

Fig. 6. Graphical representation of spectra R and S for completely specified two-dimensional map.

Note: $P_i = \{0,1,2,3\}$

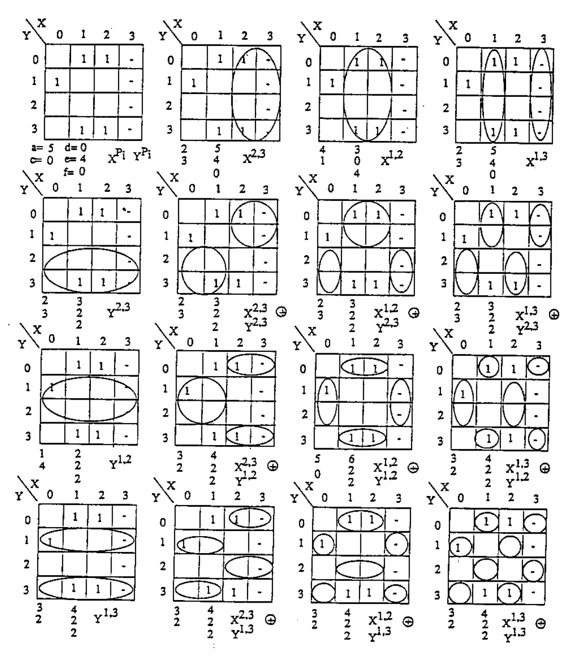

Fig. 7. Graphical representation of spectra R and S for incompletely specified two-dimensional map.