Use of Gray Decoding for Implementation of Symmetric Functions

Osnat Keren, Ilya Levin, and Radomir S. Stankovic

***Abstract*— This paper discusses reduction of the number of product terms in representation of totally symmetric Boolean functions by Sum of Products (SOP) and Fixed Polarity Reed-Muller (FPRM) expansions. The suggested method reduces the number of product terms, correspondingly, the implementation cost of symmetric functions based on these expressions by exploiting Gray decoding of input variables. Although this decoding is a particular example of all possible linear transformation of Boolean variables, it is efficient in the case of symmetric functions since it provides a significant simplification of SOPs and FPRMs. Mathematical analysis as well as experimental results demonstrate the efficiency of the proposed method.**

***Index Terms*— Symmetric function, Gray code, linear transformation, autocorrelation.**

I. Introduction

Linearization of switching functions based on linear transformation of variables is a classical method of optimization in circuit synthesis originating already in 1958 [22]. It has been recently efficiently exploited by several authors and discussed for different aspects due to its :

1) *Effectiveness*. When properly performed, the method provides considerable savings in complexity of the representation of functions with respect to different optimization criteria.
2) *Simplicity of the implementation.* The overhead comprises EXOR circuits required to perform the selected linear combination of variables. The overhead is usually quite negligible compared to the overall complexity of the implementation [12].

The linearization can be performed over different data structures used to represent functions. For example, it has been performed over Sum-of-Product (SOP) expressions [10], [13], [15], [28], AND-EXOR expressions [5], word-level expressions [27] as well as decision diagrams [7], [14], [18].

In spectral techniques, this method is studied as a mean to reduce the number of non-zero coefficients in spectral expressions for discrete functions [8], [12]. In [12], [20], and [21] the extensions to multiple-valued logic functions are discussed.

The complexity of determining an optimal non-singular binary matrix that defines the optimal linear transformation of variables is NP-complete. For this reason different strategies have been suggested in exploiting this method.

In searching for exact optimum, some restrictions should to be made on the number of variables in functions processed. For example, it has been reported in [7] that the complete search over all possible linear transformations is feasible for functions up to seven variables within reasonable space and time resources.

Another approach is to restrict considerations to particular classes of functions. For instance, in [10], [27] a method has been used for specific circuits, such as n-bit adders and an optimal linear transform has been found.

Alternatively, nearly optimal solutions can be provided by deterministic algorithms if analysis of additional information about the functions can be provided. In this direction, research has been reported by analyzing besides the functions their Walsh coefficients and autocorrelation coefficients, see, for instance, [8], [12] and [14] and references therein.

In this paper, we discuss a compromising approach. We show that when the class of functions is the totally symmetric Boolean functions, then an efficient linear transformation of variables can be determined analytically, it reduces to Gray decoding of input variables.

A justification to consider symmetric functions can be found in the following considerations. Symmetric Boolean functions represent an important fraction of Boolean functions. There are 2^{n+1} binary-valued symmetric functions out of 2^{2^n} functions. There are efficient circuit-based methods and complete BDD-based methods for identifying symmetries of completely and incompletely specified functions [11], [17], [19], [23], [29], [32].

In last several years, symmetric functions have been studied from different aspects. Optimal Fixed Polarity Reed-Muller (FPRM) expansions for totally symmetric functions are discussed in [4], [31] and references therein. A lower bound on the number of gates in conjunctive (disjunctive) normal form representation of symmetric Boolean functions is given in [30] and a method for generating a minimal SOP cover is presented in [3]. A multilevel synthesis of symmetric functions which exploits the disjoint decomposability and weight dependency of the functions is presented in [16] and a mapping of symmetric and partially symmetric functions to the CA-type FGPAs was suggested in [2]. A new expansion of symmetric functions and their application to non-disjoint functional decompositions for LUT-type FPGAs is presented in [25]. In this paper we show that the Gray decoding of the input variables almost always reduces the complexity in terms of the above three measures: the number of gates in two-level realization, the number of FPRM terms and the number of FPGA LUTs.

The paper is organized as follows. Section II gives basic definitions of symmetric Boolean functions and Gray codes. Section III presents the implementation of a symmetric function as a superposition of a Gray decoder and a non-linear function.

This work was supported by BSF under grant 2002259

978-1-4244-1710-0/07/$25.00

ection IV presents an illustrative example discussing in detail application of the proposed method. In Section V we discuss features of the proposed method and prove that the solutions produced can never increase complexity of representation of SOPs compared to the given initial representations. Section VI contains experimental results and Section VII concludes the paper.

II. Preliminaries

A. Totally symmetric functions

Let $f(x) = f(x_{n-1}, \dots x_0)$ a Boolean function of $n \geq 2$ inputs and a single output. The function f is *symmetric* in x_i and x_j iff

$$f(x_{n-1} \dots x_i \dots x_j \dots x_0) = f(x_{n-1} \dots x_j \dots x_i \dots x_0). \quad (1)$$

The function f is *totally* symmetric iff it is symmetric in all pairs of its variables.

A function $f(x) = S_i(x)$ is called an *elementary* symmetric function with working parameter i iff

$$S_i(x) = \begin{cases} 1 & ||x|| = i \\ 0 & otherwise \end{cases}$$

where $||x||$ is the Hamming weight of x. There are $n+1$ elementary symmetric functions satisfying

$$\sum_x S_i(x) S_j(x) = \begin{cases} \binom{n}{i} & i = j \\ 0 & otherwise \end{cases}.$$

Any symmetric function can be represented as a linear combination of elementary symmetric functions, i.e. $f(x) = \oplus_{i=0}^{n} a_i S_i(x)$ where $a_i \in \{0, 1\}$. Hence, there are 2^{n+1} symmetric functions out of 2^{2^n} functions.

Example 1: Consider an elementary 5-inputs symmetric function $f(x) = S_3(x)$. The K-map of the function is given in Table I. The minimal SOP representation of the function consists of 10 minterms of 5 literals.

A Fixed Polarity Reed-Muller (FPRM) expansion is an EXOR of product terms, where no two products consists of the same variables and each variable appears in complemented or un-complemented form, but not in both [24]. In matrix notation [1], the FPRM expansion of a function $f(x_{n-1}, \dots x_0)$ with a given polarity vector $h = (h_{n-1}, \dots h_1, h_0)$, is defined as

$$f(x_{n-1}, \dots x_0) = \left(\otimes_{i=0}^{n-1} [1, x_{n-1-i}^{h_{n-1-i}}] \right) \left(\otimes_{i=0}^{n-1} R^{h_{n-1-i}}(1) \right) F$$

where $\otimes$ is a Kronecker product,

$$x_i^{h_i} = \begin{cases} x_i & if\ h_i = 0 \\ x_i' & otherwise \end{cases}$$

and

$$R^{h_i}(1) = \begin{cases} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} & if\ h_i = 0 \\ \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} & otherwise \end{cases}$$

and F is the truth vector. The number of product terms in the FPRM depends on the polarity vector.

Example 2: The FPRM expansion of the *3-out-of-5* function in Example 1 with a positive polarity ($h = 0$) comprises 10 terms,

$$\begin{aligned} f &= x_4x_3x_2 \oplus x_4x_3x_1 \oplus x_4x_2x_1 \oplus x_3x_2x_1 \oplus x_4x_3x_0 \\ &\oplus x_4x_2x_0 \oplus x_3x_2x_0 \oplus x_4x_1x_0 \oplus x_3x_1x_0 \oplus x_2x_1x_0. \end{aligned}$$

The positive polarity produces the minimal number of terms, all the other 31 polarity vectors produces FPRM expansions of at least 16 product terms.

B. Gray code

The reflected binary code, also known as Gray code after Frank Gray [6], is used for listing n-bit binary numbers so that successive numbers differ in exactly one bit position. The definition of the Gray encoding and decoding is the following: Elements of a binary vector of length n, $z = (z_{n-1}, \dots z_0)$ and the vector $x = (x_{n-1}, \dots x_0)$ derived by Gray encoding are related as

$$x_i = \begin{cases} z_i & i = n-1 \\ z_i \oplus z_{i+1} & otherwise \end{cases}$$

and

$$z_i = \begin{cases} x_i & i = n-1 \\ x_i \oplus z_{i+1} & otherwise \end{cases}.$$

This relation can be written using matrix notation as $x = G_E\, z$ and $z = G_D\, x$ where $G_E = (\tau_{n-1}, \dots, \tau_1, \tau_0)$ is a non-singular matrix of the form

$$G_E = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 1 & 1 \end{pmatrix}. \quad (2)$$

and $G_D = G_E^{-1}$. The matrices G_E and G_D are called the Gray encoding and the Gray decoding matrices, respectively. The implementation of the Gray encoder (decoder) requires $n-1$ two-input EXOR gates.

Example 3: Let $n = 4$ and $z = (1, 1, 0, 1)$ then

$$\begin{aligned} x_3 &= z_3 = 1 \\ x_2 &= z_3 \oplus z_2 = 0 \\ x_1 &= z_2 \oplus z_1 = 1 \\ x_0 &= z_1 \oplus z_0 = 1 \end{aligned}$$

or

$$x = (\tau_3, \tau_2, \tau_1, \tau_0) z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

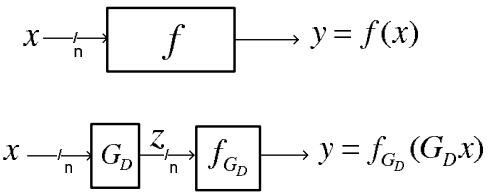

Fig. 1. Implementation of a Boolean function with a Gray decoding of the input variables

III. Implementation of symmetric functions by Gray decoded inputs

In this paper we introduce an implementation of a symmetric function as a superposition of two functions: a Gray decoder defined by the matrix G_D, and the corresponding function f_{G_D} whereas $f(x) = f_{G_D}(G_D x)$ (see Figure 1).

The main idea behind this approach is the following: A Boolean function maps elements of the vector space $\{0,1\}^n$ to $\{0,1\}$. The vector space $\{0,1\}^n$ is spanned by n base vectors, usually the binary vectors $\{\delta_i\}_{i=0}^{n-1}$ corresponding to the integer value 2^i are used. The set of δ_i's is called the initial basis. This basis is used in definition of SOP expressions.

Any set of n linearly independent vectors forms a basis, and in particular, the columns $\{\tau_i\}_{i=0}^{n-1}$ of the matrix G_E.

Since $Ix = G_E\ z$, the vector x can be interpreted as the coefficient vector that defines an element of $\{0,1\}^n$ using the initial basis, and z can be interpreted as the coefficient vector representing an element with the set of τ's. Thus, the matrices G_E and G_D define a linear transformation between the coefficient vectors.

Example 4: In Example 3, the element $(1,0,1,1) \in \{0,1\}^4$ can be represented as a linear combination of the initial base vectors $\delta_3 = (1,0,0,0), \delta_2 = (0,1,0,0), \delta_1 = (0,0,1,0)$ and $\delta_0 = (0,0,0,1)$, or as a linear combination of the columns of G_E. Namely,

$$(1,0,1,1) = 1{\cdot}\delta_3 + 0{\cdot}\delta_2 + 1{\cdot}\delta_1 + 1{\cdot}\delta_0 = 1{\cdot}\tau_3 + 1{\cdot}\tau_2 + 0{\cdot}\tau_1 + 1{\cdot}\tau_0,$$

thus, $x = (1011)$ and $z = (1101)$.

In theoretical considerations, complexity of circuit realization of a Boolean function is usually estimated without referring to a specific implementation technology. It is, therefore, often expressed in the number of two-input gates (AND/OR) that are required for the realization of the function considered. Formally, this criterion can be written in terms of a cost function [12], [26]

$$\mu(f) = |\{x | x, \tau \in \{0,1\}^n, f(x) = f(x+\tau), ||\tau|| = 1\}|$$

where $+$ stands for a bitwise EXOR of two binary vectors and $||\tau||$ is the Hamming weight of a binary vector τ. The autocorrelation function of f, is defined as $R(\tau) = \sum_{x\in\{0,1\}^n} f(x)f(x\oplus\tau)$. For a given function f, the value of μ can be related to the values of the autocorrelation function of f, at points corresponding to the base vectors,

$$\mu(f) = \sum_{i=0}^{n-1} R(\delta_i).$$

TABLE I

K-map of a *3-out-of-5* function

$x_4x_3x_2$ / x_1x_0	000	001	011	010	110	111	101	100
00						1		
01			1		1		1	
11		1		1				1
10			1		1		1	

TABLE II

K-map of Gray coded *3-out-of-5* function

$z_4z_3z_2$ / z_1z_0	000	001	011	010	110	111	101	100
00								
01		1	1	1	1	1		1
11				1	1		1	1
10								

In the case of initial basis, these are points 2^i, and linear transformation of variables performs the shift of these values.

There is a variety of minimization procedures that construct a linear transformation deterministically, see, for instance [14], [15] and [28] and references therein. It should be noticed that implementation of such procedures may be a space and time demanding task, and therefore, it is useful to take into considerations specific features of functions to be realized. In particular, we point out that for totally symmetric Boolean functions the linear transformation of variables derived from the Gray code almost always reduce the implementation cost. The same transformation often reduces the number of terms in Fixed polarity Reed-Muller expressions.

IV. Motivation example

Consider the *3-out-of-5* function in Example 1. Let G_E and the G_D be the 5×5 Gray encoding and decoding matrices. The columns of G_E are binary vectors of length 5 corresponding to the integer values $1, 3, 6, 12$ and 24. Let $z = G_D x$ be the Gray decoded inputs. Table II shows the K-map of f_{G_D}. The minimal SOP representation of f_{G_D} consists of 5 products,

$$f_{G_D}(z_4, z_3, z_2, z_1, z_0) = z_3 z_2' z_0 + z_3 z_1' z_0 + z_4 z_2' z_0 + z_4' z_2 z_1' z_0 + z_4 z_3' z_1 z_0.$$

The FPRM expansion of f_{G_D} with a polarity vector $h = (11000)$ is

$$f_{G_D}(z_4, z_3, z_2, z_1, z_0) = z_0 \oplus z_2 z_1 z_0 \oplus z_3' z_2 z_0 \oplus z_4' z_3' z_0.$$

The values of the autocorrelation function of the original *3-out-of-5* function are shown in Figure 2 (top figure). The values of $R(\tau)$ at positions $\tau = 1, 2, 4, 8$ and 16 corresponding to the initial base vectors are all zero , thus, the minimal SOP comprises 10 minterms. The autocorrelation values at positions $\tau = 1, 3, 6, 12$ corresponding to the new base vectors (τ_0, τ_1, τ_2 and τ_3) are equal to 6.

Applying the Gray decoding on the inputs is equivalent to permuting the autocorrelation values so that high autocorrelation values are now placed at positions 2^i. The autocorrelation function of f_{G_D} is shown at the bottom of Figure 2. The sum of the autocorrelation values of f_{G_D} at positions 2^i ,$i =$

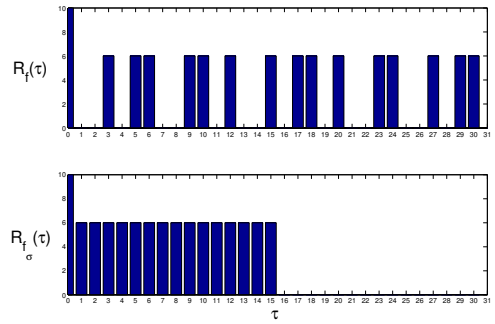

Fig. 2. Autocorrelation function values of the original *3-out-of-5* symmetric function f (top) and the values of the autocorrelation function corresponding to f_{G_D} with the Gray decoded inputs (bottom).

$0, \ldots, 4$ is $4 \cdot 6 + 0$, therefore, the number of pairs in the first merging step of the Quine-McClusky minimization algorithm is now 12 which leads to a minimal SOP representation.

V. Analysis

Let $f(x) = f(x_{n-1}, \ldots x_0) \sum_{i=0}^{n} a_i S_i(x)$, $a_i \in \{0,1\}$, a totally symmetric Boolean function of n variables and a single output. The autocorrelation function of $S_i(x)$ is [12]

$$\begin{aligned} R_{S_i}(\tau) &= \sum_{x \in \{0,1\}^n} S_i(x) S_i(x \oplus \tau) \\ &= \begin{cases} \binom{n-||\tau||}{i-||\tau||/2}\binom{||\tau||}{||\tau||/2} & ||\tau|| \text{ is even} \\ 0 & otherwise \end{cases} \end{aligned}$$

where $\binom{a}{b} = 0$ for $b < 0$.

The cross correlation between $S_i(x)$ and $S_j(x)$ is

$$\begin{aligned} R_{S_i,S_j}(\tau) &= \sum_{x \in \{0,1\}^n} S_i(x) S_j(x \oplus \tau) \\ &= \begin{cases} \binom{n-||\tau||}{i-w}\binom{||\tau||}{w} & i-j+||\tau|| \text{ is even} \\ 0 & otherwise \end{cases} \end{aligned}$$

where $w = (i - j + ||\tau||)/2$.

The autocorrelation function of f is

$$\begin{aligned} R_f(\tau) &= \sum_{x \in \{0,1\}^n} f(x) f(x \oplus \tau) \\ &= \sum_{i=0}^{n} a_i R_{S_i}(\tau) + \sum_{\substack{i,j=0 \\ i \neq j}}^{n} a_i a_j R_{S_i,S_j}(\tau). \quad (3) \end{aligned}$$

Therefore, the autocorrelation values in positions corresponding the the initial set of base vectors $\{\delta_i\}_{i=0}^{n-1}$ is

$$\begin{aligned} R_f(\delta_i) &= 2\sum_{k=1}^{n-1} a_k a_{k+1} R_{S_k,S_{k+1}}(\tau) \\ &= 2\sum_{k=1}^{n-1} a_k a_{k+1} \binom{n-1}{k}. \quad (4) \end{aligned}$$

On the other hand, the autocorrelation values at positions corresponding to the base vectors $\tau_i = \delta_i + \delta_{i+1}$, $i = 0, \ldots n-2$, defined by the columns of the Gray encoding matrix G_E, are

$$R_f(\tau_i) = 2\sum_{k=0}^{n} a_k \binom{n-2}{k-1} + 2\sum_{k=1}^{n-2} a_k a_{k+2} \binom{n-2}{k} \quad (5)$$

The following Theorem states that the realization cost of f_{G_D} with the Gray decoded inputs is less or equal to the realization cost of f for any totally symmetric function.

Theorem 1: Let $f(x) = \sum_{i=1}^{n} a_i S_i(x)$, $a_i \in \{0,1\}$ a totally symmetric function, and let f_{G_D} the corresponding function with the Gray decoded inputs, i.e. $f(x) = f_{G_D}(G_D x)$. Then,

$$\mu_f \leq \mu_{f_{G_D}}.$$

Proof: The proof is based the fact that $R_{f_{G_D}}(\delta_i) = R_f(G_D^{-1}\delta_i) = R_f(\tau_i)$. Let $\Delta_i = R_f(\tau_i) - R_f(\delta_i)$, clearly, $\Delta_{n-1} = 0$ and for $0 \leq i < n-1$, $\Delta_i = 2\sum_{k=0}^{n} d_k$ where

$$d_k = a_k \left(\binom{n-2}{k-1} - a_{k+1} \binom{n-1}{k} + a_{k+2} \binom{n-2}{k} \right). \quad (6)$$

We now show that $\Delta_i \geq 0$ for all i. From 6, if $a_k = 0$ than $d_k = 0$, otherwise, there are four possible cases:

1) If $a_{k+1} = a_{k+2} = 0$ than $d_k > 0$.
2) If $a_{k+1} = a_{k+2} = 1$ than $d_k = 0$ since
$$\binom{a}{b} = \binom{a-1}{b} + \binom{a-1}{b-1}.$$
3) If $a_{k+1} = 0$ and $a_{k+2} = 1$ than $d_k > 0$.
4) If $a_{k+1} = 1$ and $a_{k+2} = 0$ than we may consider the sum $d_k + d_{k+1}$ and get
$$\binom{n-2}{k-1} - \binom{n-1}{k} + \binom{n-2}{k} + a_{k+2}\binom{n-2}{k} \geq 0 \quad (7)$$

Therefore, $R_{f_{G_D}}(\delta_i) = R_f(\tau_i) \geq R_f(\delta_i)$. From [12], the cost function μ_f of a function $f : \{0,1\}^n \rightarrow \{0,1\}$ equals to $\mu_f = 2^n - 2R_f(0) + 2\sum_{i=0}^{n-1} R_f(\delta_i)$, and thus $\mu_{f_{G_D}} \geq \mu_f$. ■

VI. Experimental results

In this section, we compare the implementation cost of the original and Gray-coded functions in terms of:

a) The number of Look-Up-Tables (LUTs) required to implement the function by *SPARTAN3 xcs200ft256* as computed by LeonardoSpectrum.

b) The number of literals (L) in its minimal SOP representation as produced by ESPRESSO.

c) The number of nonzero terms in the optimal Fixed-Polarity Reed-Muller ($FPRM$) expansion.

Tables III and IV show the number LUTs for several totally symmetric functions of 8 and 12 input variables, the number of literals in the minimal SOP expression and the number of non-zero FPRM terms as computed with and without the Gray decoding. The improvement in those parameters is given in percentage. The symmetric functions $f = \sum_i a_i S_i(x)$ are specified by a set I, $I = \{i | a_i \neq 0\}$, of working parameters, I is written in the left column of Tables III and IV.

TABLE III

TOTALLY SYMMETRIC FUNCTIONS OF 8 INPUTS

I	LUT $orig$	LUT $Gray$	%	L $orig$	L $Gray$	%
3	12	7	41.7	448	92	79.5
4	13	9	30.8	560	106	81.1
3, 4	18	13	27.8	490	185	62.2
3, 5	15	8	46.7	896	45	95.0
3, 4, 5	18	15	16.7	336	123	63.4
2, 3, 5, 7	19	10	47.4	904	74	91.8
0, 2, 3, 5, 8	18	11	38.9	856	109	87.3

I	$FPRM$ $orig$	$FPRM$ $Gray$	%
3	64	24	62.5
4	107	15	86.0
3, 4	96	31	67.7
3, 5	104	17	83.6
3, 4, 5	162	49	69.7
2, 3, 5, 7	36	40	-11.1
0, 2, 3, 5, 8	107	25	76.6

TABLE IV

TOTALLY SYMMETRIC FUNCTIONS OF 12 INPUTS

I	LUT $orig$	LUT $Gray$	%	L $orig$	L $Gray$	%
3	65	26	60.0	2640	470	82.2
4	32	41	-28.1	5940	800	86.5
3, 4	143	71	50.3	5445	1225	77.5
3, 5	37	57	-54.1	12144	584	95.2
3, 4, 5	204	118	40.2	7920	1170	85.2
0, 2, 3, 5, 8	217	101	53.5	17876	1582	91.1

I	$FPRM$ $orig$	$FPRM$ $Gray$	%
3	232	200	13.8
4	794	166	79.1
3, 4	562	306	45.6
3, 5	1024	136	86.7
3, 4, 5	1354	356	73.7
0, 2, 3, 5, 8	738	328	55.6

Table V shows how the Gray decoding reduces the implementation cost of several totally symmetric LGSynth93 benchmark functions. Given a polarity vector, the number of non-zero FPRM terms of a k-output function is defined as the size of the union of the non-zero terms in the FPRM expansion of each one of the k single-output functions. For example, the original benchmark function *rd84* has four outputs, the number of non-zero FPRM terms of each is $28, 8, 1$ and 70 and the size of the union of these terms is 107. The number of non-zero terms of the corresponding Gray coded single-output functions is $14, 4, 1$ and 38 and the size of their union is 39.

TABLE V

BENCHMARK FUNCTIONS

	in	out	LUT $orig$	LUT $Gray$	L $orig$	L $Gray$
rd53	5	3	6	4	140	35
rd73	7	3	24	8	756	141
rd84	8	4	51	13	1774	329
9sym	9	1	36	36	504	135

	in	out	$FPRM$ $orig$	$FPRM$ $Gray$
rd53	5	3	20	12
rd73	7	3	63	24
rd84	8	4	107	39
9sym	9	1	173	33

VII. CONCLUSIONS

The problem of linearization of logic functions may be considered as a determining a linear transform for variables in a given function, which produces a representation of the function appropriate for particular applications. However, it is not always necessary to determine the best possible linear transformation for a class of functions. For many practical applications it is sufficient to find a suitable transform producing acceptable solutions.

In this paper we consider the class of symmetric functions and point out a suitable linear transformation of variables resulting in considerably reduced number of product terms in AND-OR and Reed-Muller expressions.

We propose a method to represent a symmetric logic function as a superposition of a linear portion that realize the Gray decoding of input vectors and a non-linear portion. Being a particular case of the linear transformation, the described Gray decoding transform enables to achieve very compact implementations of the initial symmetric function.

We have shown that the use of the Gray transform improves the complexity of the initial function implementation in terms of a specific cost function. Experimental results show that for majority of benchmarks the proposed method improves also a LUT based implementation of the function. The suggested approach can be extended to partially symmetric functions by the Gray decoding of each symmetry class separately.

REFERENCES

[1] J.T. Astola and R.S. Stankovic, *Fundamentals of Switching Theory and Logic Design: A Hands on Approach,* Springer-Verlag New York, 2006.

[2] M. Chrzanowska-Jeske and Z. Wang, " Mapping of symmetric and partially-symmetric functions to theCA-type FPGAs" *proc. of the 38th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 290-293, Aug 1995.

[3] D. L. Dietmeyer, "Generating minimal covers of symmetric functions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 5, pp. 710-713, May 1993.

[4] R. Drechsler, B. Becker, "Sympathy: fast exact minimization of fixed polarity Reed-Muller expressions for symmetric functions," *Proc. of the European Design and Test Conference*, pp. 91 - 97, March 1995.

[5] R. Drechsler and B. Becker, "EXOR transforms of inputs to design efficient two-level AND-EXOR adders," *IEE Electronic Letters*, vol. 36, no. 3, pp. 201-202, Feb. 2000.

[6] F. Gray, " Pulse code communication," March 17, 1953 (filed Nov. 1947). U.S. Patent 2,632,058.

[7] W. Günther, R. Drechsler, "BDD minimization by linear transforms", *Advanced Computer Systems*, pp. 525-532, 1998.

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

[9] J. Jain, D. Moundanos, J. Bitner, J.A. Abraham, D.S. Fussell and D.E. Ross, "Efficient variable ordering and partial representation algorithm," *Proc. of the 8th International Conference on VLSI Design*, pp. 81-86, Jan. 1995.

[10] J. Jakob, P.S. Sivakumar, V.D. Agarwal, "Adder and comparator synthesis with exclusive-OR transform of inputs", *Proc. 1st Int. Conf. VLSI Design*, pp. 514-515, 1997.

[11] S. Kannurao and B. J. Falkowski, "Identification of complement single variable symmetry in Boolean functions through Walsh transform," *Proceedings of the International Symposium on Circuits and Systems* (ISCAS), 2002.

[12] M.G. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*, John Wiley, 1976.

[13] M.G. Karpovsky, E.S. Moskalev, "Utilization of autocorrelation characteristics for the realization of systems of logical functions," *Avtomatika i Telemekhanika*, No. 2, 1970, 83-90, English translation *Automatic and Remote Control*, Vol. 31, pp. 342-350, 1970.

[14] M.G. Karpovsky, R.S. Stankovic and J.T. Astola, "Reduction of sizes of decision diagrams by autocorrelation functions," *IEEE Trans. on Computers*, vol. 52, no. 5, pp. 592-606, May 2003.

[15] O. Keren, I. Levin and R.S. Stankovic, "Linearization of Functions Represented as a Set of Disjoint Cubes at the Autocorrelation Domain," *Proc. of the 7th International Workshop on Boolean Problems*, pp. 137-144, Sept. 2006.

[16] B. G. Kim, D.L. Dietmeyer, "Multilevel logic synthesis of symmetric switching functions," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.10, No. 4O, pp. 436-446, Apr. 1991.

[17] E.J., Jr. McCluskey, "Detection of group invariance or total symmetry of a Boolean function," *Bell Systems Tech. Journal*, vol. 35, no. 6, pp. 1445-1453, 1956.

[18] Ch. Meinel, F. Somenzi, T. Tehobald, "Linear sifting of decision diagrams and its application in synthesis," *IEEE Trans. CAD*, Vol. 19, No. 5, 2000, 521-533.

[19] D. Moller, J. Mohnke, and M. Weber, "Detection of symmetry of Boolean functions represented by ROBDDs," *Proc. of the International Conference on Computer-Aided Design (ICCAD),* 1993, pp. 680684.

[20] C. Moraga, "Introducing disjoint spectral translation in spectral multiple-valued logic design", *IEE Electronics Letters*, 1978, Vol. 14, No. 8, pp. 248-243, 1978.

[21] C. Moraga, "On some applications of the Chrestenson functions in logic design and data processing", *Mathematic and Computers in Simulation*, Vol. 27, pp. 431-439, 1985.

[22] E.I. Nechiporuk, "On the synthesis of networks using linear transformations of variables", *Dokl. AN SSSR*. Vol. 123, No. 4, pp. 610-612, Dec. 1958.

[23] S. Panda, F. Somenzi, and B. Plessier, "Symmetry detection and dynamic variable ordering of decision diagrams," *Proc. of the International Conference on Computer-Aided Design (ICCAD)*, 1994.

[24] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, Feb. 1999

[25] T. Sasao, "A new expansion of symmetric functions and their application to non-disjoint functional decompositions for LUT type FPGAs", *IEEE Int. Workshop on Logic Synthesis, IWLS-2000*, May 2000.

[26] C. E. Shannon, "The Synthesis of Two-Terminal Switching Circuits," *Bell System Technical Journal*, Vol. 28, pp. 59-98, Jan. 1949.

[27] R.S. Stankovic,J.T. Astola, "Some remarks on linear transform of variables in adders," *Proc. 5th Int. Workshop on Applications of Reed-Muller Expression in Circuit design*, Starkville, Mississippi, USA, Aug. 10-11, pp. 294-302, 2001.

[28] D. Varma and E.A. Trachtenberg, "Design automation tools for efficient implementation of logic functions by decomposition," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 8, pp. 901-916, Aug. 1989.

[29] K.H. Wang and J.H. Chen, "Symmetry Detection for Incompletely Specified Functions," *Proc. of the 41st Conference on Design Automation Conference, (DAC'04)*, pp. 434-437, 2004.

[30] G. Wolfovitz "The complexity of depth-3 circuits computing symmetric Boolean functions," *Information Processing Letters*, vol. 100, No. 2, pp. 41 - 46, Oct. 2006.

[31] S.N. Yanushkevich, J.T. Butler, G.W. Dueck, V.P. Shmerko, "Experiments on FPRM expressions for partially symmetric logic functions", *Proc.30th Int. Symp. on Multiple-Valued Logic*, Portland, Oregon USA, 141-146, May 2000.

[32] J. S. Zhang, A. Mishchenko, R. Brayton, M. Chrzanowska-Jeske, "Symmetry detection for large Boolean functions using circuit representation, simulation, and satisfiability", *Proceedings of the 43rd annual conference on Design automation* , pp. 510 - 515, 2006.