Optimization of Kronecker Expressions using the Extended Dual Polarity Property

Dragan Jankovic[1] , Radomir S. Stankovic[1], Claudio Moraga[2]

Abstract - Đ Fixed Polarity Reed-Muller (FPRM) expressions are polynomial expressions for Boolean functions. A method for optimization of FPRMs using dual polarity property has been published in [4]. FPRMs are a subclass of Kronecker expressions. Analogously to dual polarity for FPRMs we introduce the notion of extended dual polarity for Kronecker expressions and present a method for optimization of Kronecker expressions for Boolean functions exploiting this concept.

Keywords - **Boolean function, Kronecker expression, optimization of Boolean functions, dual polarity**

I. INTRODUCTION

AND-EXOR realizations have some advantages over AND-OR expressions, such as high testability [1,2], low cost for arithmetic and symmetric functions in the number of product count, detection of symmetric variables [5], Boolean matching [6], etc. Fixed Polarity Reed-Muller expressions (FPRMs) are an important class of AND-EXOR expressions. For some classes of functions used in practice, FPRMs require fewer products than AND-OR expressions [3]. For an n-variable Boolean function there are 2^n FPRMs. Optimization of FPRMs is a problem of finding the FPRM with the smallest number of product terms.

Kronecker expressions are potentially better than FPRMs in optimization of Boolean functions if criterion is the number of non-zero terms. There are 3^n different Kronecker expressions for an n-variable Boolean function. FPRMs are a subclass of Kronecker expressions.

In [4], it is given a method for optimization of FPRMs using the dual polarity property. In this method, all of 2^n possible FPRMs are calculated by using relationship between two FPRMs whose polarities are dual. Method starts from the zero polarity FPRM.

In this paper, we extend the term "dual polarity" into the "extended dual polarity" and show the relationship between two Kronecker expressions with extended dual polarities. Based on these relationships, we generate a new algorithm for optimization of Kronecker expressions of Boolean functions. The algorithm starts from the truth-vector of a given Boolean functions and calculate all 3^n Kronecker expressions using route in which each two neighbours polarities are extended dual. We denote this route as the "extended dual route" and present a procedure for determination of this route.

The algorithm proposed is an exhaustive-search algorithm, but conversion from one Kronecker expression to another extended dual polarity Kronecker expression is carried out using one-bit checking. Due to that, and a simple processing this algorithm appears efficient. Experimental results show efficiency of the proposed algorithm.

II. BASIC DEFINITION

Definition 1: Each n-variable switching function f given by the truth-vector $\mathbf{F} = \left[f_0, \ldots, f_{2^n-1}\right]^T$ can be represented by the positive polarity Reed-Muller expression (PPRM) defined as

$$f(x_1, \ldots, x_n) = \mathbf{X}(n)\mathbf{R}(n)\mathbf{F}$$

where $\mathbf{X}(n) = \bigotimes_{i=1}^{n}\begin{bmatrix}1 & x_i\end{bmatrix}$, and $\mathbf{R}(n) = \bigotimes_{i=1}^{n}\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$,

where $\otimes$ denotes the Kronecker product, and addition and multiplication are performed modulo 2. $\mathbf{R}(n)$ is the Reed-Muller transform matrix of order ($2^n \times 2^n$).

If each variable can appear as complemented or uncomplemented, but not both, the related expressions is denoted as the fixed-polarity Reed-Muller (FPRM) and is given as

$$f(x_1, \ldots, x_n) = \mathbf{X}_H(n)\mathbf{R}_H(n)\mathbf{F},$$

where

$$\mathbf{X}_H(n) = \bigotimes_{i=1}^{n}\begin{bmatrix}1 & x_i^{h_i}\end{bmatrix}, \qquad x_i^{h_i} = \begin{cases} x_i, & h_i = 0,\\ \overline{x}_i, & h_i = 1,\end{cases}$$

and

$$\mathbf{R}_H(n) = \bigotimes_{i=1}^{n}\mathbf{R}^{h_i}(1) \qquad \mathbf{R}^{h_i}(1) = \begin{cases}\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}, & h_i = 0,\\ \begin{bmatrix}0 & 1\\ 1 & 1\end{bmatrix}, & h_i = 1.\end{cases}$$

Therefore, FPRMs are uniquely characterized by the polarity vectors $\mathbf{H} = \left[h_1, \ldots, h_n\right]^T$, $h_i \in \{0,1\}$, where $h_i = 1$ shows that the i-th variable is complemented and written as $\overline{x}_i$. (In

[1] Dragan Jankovic and Radomir S. Stankovic are with the Faculty of Electronic Engineering, University of Nis, 18000 Nis, Yugoslavia, Emails: gaga@elfak.ni.ac.yu , rstankovic@elfak.ni.ac.yu

[2] Claudio Moraga is with the University of Dortmund, Germany and with the Department of Artificial Intelligence, Technical University of Madrid, Spain under Grant SAB2000/0048 SEEU-Ministry of Education, Culture and Sport. Email: claudio@moraga.de

the case of Kronecker expressions –(see definition 2)- h_i will be allowed also to take the value 2, meaning that the I-th variable may appear both in complemented and non-complemented form.)

A FPRM can be given by the FPRM spectrum $\mathbf{R}_f^H$ calculated as

$$\mathbf{R}_f^H = \mathbf{R}_H(n)\mathbf{F}.$$

Example 1: The FPRM of a 2-variable Boolean function *f*, given by the truth-vector $\mathbf{F} = [0,1,1,0]^T$, for a polarity vector $\mathbf{H} = (0,1)$ is given by

$$f(x_1, x_2) = 1\cdot 1 \oplus 1\cdot \bar{x}_2 \oplus 1\cdot x_1 \oplus 0\cdot x_1\bar{x}_2 = 1 \oplus \bar{x}_2 \oplus x_1.$$

The corresponding FPRM spectrum is given by $\mathbf{R}_f^H = [1,1,1,0]$.

The FPRM of function *f* can be represented by the set of binary strings here called terms. A variable that is present in a product term is replaced by 1, and 0 replaces an absent variable. Therefore the FPRM for a function *f* and polarity vector given in Example 1 is represented by the following term set

$$\{00, 01, 10\}.$$

Definition 2: Each *n*-variable switching function *f* given by the truth-vector $\mathbf{F} = \left[f_0, \ldots, f_{2^n-1}\right]^T$ can be represented by the Kronecker expression defined as

$$f(x_1, \ldots, x_n) = \left(\bigotimes_{i=1}^{n} \mathbf{X}^{h_i}\right)\left(\bigotimes_{i=1}^{n} \mathbf{T}^{h_i}\right)\mathbf{F}$$

where

$$\mathbf{X}^{h_i} = \begin{cases} \begin{bmatrix}1 & x_i\end{bmatrix}, & h_i = 0, \\ \begin{bmatrix}1 & \bar{x}_i\end{bmatrix}, & h_i = 1, \\ \begin{bmatrix}\bar{x}_i & x_i\end{bmatrix}, & h_i = 2, \end{cases}$$

and

$$\mathbf{T}^{h_i} = \begin{cases} \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}, & h_i = 0, \\ \begin{bmatrix}0 & 1\\ 1 & 1\end{bmatrix}, & h_i = 1, \\ \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, & h_i = 2, \end{cases}$$

Example 2: The Kronecker expression of a 2-variable Boolean function *f*, given in Example 1, for a polarity vector $\mathbf{H} = (2,1)$ is given by

$$\begin{aligned} f(x_1, x_2) &= \left(\begin{bmatrix}\bar{x}_1 & x_1\end{bmatrix} \otimes \begin{bmatrix}1 & x_2\end{bmatrix}\right)\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}\right)\cdot \mathbf{F} \\ &= \bar{x}_1 x_2 \oplus x_1 \oplus x_1 x_2 \end{aligned}$$

The corresponding Kronecker spectrum $\mathbf{K}_f^{(2,1)}$ is given by $\mathbf{K}_f^{(2,1)} = [0,1,1,1]$ and the set of terms is $\{01, 10, 11\}$.

III. Extended Dual Polarity

Each Kronecker expression is characterized by its polarity vector. Two polarity vectors are dual if they differ in only one bit. We introduce the term "extended dual polarity" which will be used in our minimization method.

Definition 2: $\mathbf{H}' = (h'_1, \ldots, h'_{i-1}, h'_i, h'_{i+1}, \ldots, h'_n)$ is the dual of $\mathbf{H} = (h_1, \ldots, h_{i-1}, h_i, h_{i+1}, \ldots, h_n)$ iff $h'_j = h_j$, $j \neq i$ and $h'_i \neq h_i$.

Example 3: Extended dual polarities for polarity $\mathbf{H} = (1,0)$ are the polarities (0,0), (2,0), (1,1), and (1,2).

The number of polarity vectors, which characterize all possible Kronecker expressions for an *n*-variable Boolean function, is 3^n. It is possible to order all 3^n polarities in a way that each two successive polarities are extended dual polarities. This order we denote as the "extended dual polarity route". Traversing the 3-valued *n*-dimensional hypercube can generate one of several possible extended dual polarity routes.

Example 4: An extended dual polarity route generated by traversing a 3-valued *n*-dimensional hypercube is given by

(000)—(001)—(002)—(012)—(011)—(010)—(020)—
(021)—(022)—(122)—(121)—(120)—(110)—(111)—
(112)—(102)—(101)—(100)—(100)—(201)—(202)—
(212)—(211)—(210)—(220)—(221)—(222)

An extended dual polarity route can be constructed by using the recursive procedure ***route***(*level*, *direction*) given in Fig. 1, called as *route*(0,0).

IV. Method for Calculation of Kronecker Expression

Let $m = m_1^{i-1} m_i m_{i+1}^n$ be the compact representation of a term in the Kronecker expression for a given function *f* for the polarity $p = (p_1 \cdots p_{i-1} p_i p_{i+1} \cdots p_n)$. Term *m* produces new terms in the Kronecker expression of the function *f* for the extended dual polarity $p' = (p'_1 \cdots p'_{i-1}\, p'_i\, p'_{i+1} \cdots p'_n)$ depending on the value of p_i and p'_i. Table I shows all cases. In some of them , the term produces a new term while in some other cases, the term is modified so that only one of its bits is complemented. These processing rules are simple and ensure efficiency of the method. After processing all terms, by using these rules, a procedure for the deleting of equal terms starts.

```
void route(int level, int direction)
{
  if( direction == 0)
  { if(level == no_variable)
    { -- out new polarity vector h }
    else
    {  h[level] = 0;
       route(level+1, 0);
       h[level] = 1;
       route(level+1, 1);
       h[level] = 2;
       route(level+1, 0);
     }
  }
  else
  {    if(level == no_variable)
       { -- out new polarity vector h }
       else
       {   h[level] = 2;
           route(level+1, 1);
           h[level] = 1;
           route(level+1, 0);
           h[level] = 0;
           route(level+1, 1);
       }
  }
}
```

Figure 1: Procedure ***route***

Example 4: Let the (012)-polarity Kronecker expression for a 3-variable Boolean function f given by the truth vector $\mathbf{F} = [1,0,1,0,1,0,0,1]^T$ be given by

$$f = \bar{x}_3 \oplus x_1\bar{x}_3 \oplus x_1x_3 \oplus x_1\bar{x}_2\bar{x}_3 \oplus x_1\bar{x}_2x_3$$

i.e. the Kronecker spectrum is $[10001111]^T$.

The extended dual polarities and the corresponding extended dual polarity Kronecker expressions are given in the Table II.

Calculation procedures for these extended dual polarity Kronecker expressions are shown in Tables III, IV, V, VI, VII, and VIII. Note that deleted terms are marked with a simple line while changed terms, with a double line.

TABLE I

PROCESSING RULE.

<table>
<tr><th>p_i</th><th>p'_i</th><th>new terms</th></tr>
<tr><td>0</td><td>1</td><td rowspan="2">if m_i =1 then generate $m_1^{i-1}0m_{i+1}^n$</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>0</td><td>2</td><td rowspan="2">if m_i =0 then generate $m_1^{i-1}1m_{i+1}^n$</td></tr>
<tr><td>2</td><td>0</td></tr>
<tr><td>1</td><td>2</td><td>if m_i =1 then set m_i =0
if m_i =0 then generate $m_1^{i-1}1m_{i+1}^n$</td></tr>
<tr><td>2</td><td>1</td><td>if m_i =0 then set m_i =1
if m_i =1 then generate $m_1^{i-1}0m_{i+1}^n$</td></tr>
</table>

TABLE II

EXTENDED DUAL POLARITY KRONECKER EXPRESSIONS.

polarity	spectrum
112	(01111111)
212	(10000111)
002	(10000011)
022	(10100011)
010	(11001010)
011	(01001010)

TABLE III

POLARITY (012) TO (112).

polarity (012)	new terms	polarity (112)
~~000~~		001
100	~~000~~	010
101	001	011
110	010	100
111	011	101
		110
		111

TABLE IV

POLARITY (012) TO (212).

polarity (012)	new terms	polarity (212)
000	~~100~~	000
~~100~~		101
101		110
110		111
111		

TABLE V

POLARITY (012) TO (002).

polarity (012)	new terms	polarity (002)
000		000
~~100~~		110
~~101~~		111
110	~~100~~	
111	~~101~~	

TABLE VI

POLARITY (012) TO (022).

polarity (012)	new terms	polarity (022)
000	010	000
~~100~~	110	010
~~101~~	111	110
~~110~~	~~100~~	111
~~111~~	~~101~~	

TABLE VII

POLARITY (012) TO (010).

polarity (012)	new terms	polarity (010)
~~000~~	001	000
100	~~101~~	001
~~101~~		100
110	~~111~~	110
~~111~~		

TABLE VIII

POLARITY (012) TO (011).

polarity (012)	new terms	polarity (011)
~~000~~	001	001
~~100~~	~~101~~	100
~~101~~	100	110
~~110~~	~~111~~	
~~111~~	110	

V. OPTIMIZATION ALGORITHM

Example 4 shows that it is possible to calculate all possible Kronecker expressions using the proposed method for transforming fixed polarity Kronecker expressions into extended dual polarity Kronecker expression along the route without repetitive calculations. Therefore, we can perform optimization of Kronecker expressions by using the following exhaustive-search algorithm.

Algortihm

1. Initialization:
   - set the polarity vector p' to **H**=(2,2,…,2),i.e., p_{opt} =**H**(2,2,…,2).
   - $C_{\min} = 2^n$ - number of non-zero coefficients
2. List all the minterms for an n-variable switching functions f.
3. Obtain the Kronecker expansion for the polarity p' based on the proposed rule. Calculate the total number of non-zero coefficients $C'_{\min}$.

   If $C'_{\min} < C_{\min}$ then $C_{\min} = C'_{\min}$.
4. Stop if all the polarities have been treated. Otherwise go to the step 5.
5. Determine the next polarity p', of the Kronecker expansion according to the recursive route and go to the step 3.

VI. EXPERIMENTAL RESULTS

In this section, we present some experimental results estimating features and efficiency of the proposed algorithm for the minimization Kronecker expressions. We developed a program in C for determination of optimal Kronecker expression for arbitrary Boolean functions represented by minterms. The experiments were carried out on 400 MHz PC Celeron with 64Mb of main memory and all runtimes are given in CPU seconds. Table IX gives the runtimes for the Kronecker expression optimization for the simple functions taking the value 1 at first three minterms (0,1,2), randomly generated functions with 25% of all possible minterms, and randomly generated functions with 75% of all possible minterms, where the number of variables n ranged from 5 to 10. It can be concluded that the number of minterms strongly influences the runtime of proposed algorithm.

TABLE IX

EXPERIMENTAL RESULTS.

n	(012)	25%	75%
5	< 0.01	0.01	0.01
6	< 0.01	0.04	0.05
7	0.04	0.52	0.56
8	0.25	6.24	5.93
9	1.41	73.27	75.19
10	8.19	890.00	917.09
11	47.93	10831.21	11199.35

VII. CONCLUDING REMARKS

We have introduced the notion of extended dual polarities in Kroencker expressions for Boolean functions and present a method for conversions of Kronecker expressions from one polarity to another. Based on this method, we determine an algorithm for calculation of all Kronecker expressions. Calculation is performed starting from the truth-vector. All Kronecker expressions are calculated along the route that provides calculation of each Kronecker expressions exactly once.

The proposed method for transformation of Kronecker expressions from one to another one extended dual polarity is simple. Therefore, our exhaustive-search Kronecker expression optimization method is very efficient. Experimental results confirm this. Future work will be in extension of the proposed method and related algorithm to polynomial expressions for multiple-valued functions.

REFERENCES

[1] Drechsler, R., Hengster, H., Schaefer, H., Hartman, J., Becker, B., "Testability of 2-Level AND/EXOR Circuits", in *Journal of Electronic Testing, Theory and Application (JETTA)*, Vol. 14, No. 3, June 1999, 173-192.

[2] Sarabi, A., Perkowski, M.A., "Fast exact and quasy-minimal minimization of highly testable fixed polarity AND/XOR canonical networks", *Proc. Design Automation Conference*, June 1992, 30-35.

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

[4] Tan, E.C., Yang, H., "Optimization of Fixed-polarity Reed-Muller circuits using dual-polarity property", *Circuits Systems Signal Process*, Vol. 19, No. 6, 2000, 535-548.

[5] Tsai, C., Marek-Sadowska, M., "Generalized Reed-Muller forms as a tool to detect symmetries", *IEE Proc. Computers and Digital Techniques*, Vol. 141, No. 6, November 1994, 369-374.

[6] Tsai, C., Marek-Sadowska, M., "Boolean functions classification via fixed polarity Reed-Muller forms", *IEEE Trans.Comput.,* Vol. C-46, No. 2, February 1997, 173-186.