Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

An algorithm for identifying symmetric variables based on the order eigenvalue matrix*

Xiao-hua LI[1], Ji-zhong SHEN[†‡2]

([1]*Campus Information Center, Zhejiang University, Hangzhou 310027, China*)

([2]*College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: jzshen@zju.edu.cn

Received Mar. 5, 2016; Revision accepted Aug. 14, 2016; Crosschecked Nov. 3, 2017

Abstract: To simplify the process for identifying 12 types of symmetric variables in Boolean functions, we propose a new symmetry detection algorithm based on minterm expansion or the truth table. First, the order eigenvalue matrix based on a truth table is defined according to the symmetry definition of a logic variable. By analyzing the constraint conditions of the order eigenvalue matrix for 12 types of symmetric variables, an algorithm is proposed for identifying symmetric variables of the Boolean function. This algorithm can be applied to identify the symmetric variables of Boolean functions with or without don't-care terms. The proposed method avoids the restriction by the number of logic variables of the graphical method, spectral coefficient methods, and AND-XOR expansion coefficient methods, and solves the problem of completeness in the fast computation method. The algorithm has been implemented in C language and tested on MCNC91 benchmarks. The application results show that, compared with the traditional methods, the new algorithm is an optimal detection method in terms of the applicability of the number of logic variables, the Boolean function including don't-care terms, detection type, and complexity of the identification process.

Key words: Boolean function; Symmetric variable; Boolean logic algebra system; Order eigenvalue matrix; Truth table
https://doi.org/10.1631/FITEE.1601052 **CLC number:** TN431

1 Introduction

The symmetry of logic variables is a very important property of Boolean functions. Symmetric variables exist in most Boolean functions. For arbitrary symmetric variables x_i and x_j, the Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$ satisfies the condition $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)=f(x_1, \ldots, x_j, \ldots, x_i, \ldots, x_n)$. A Boolean function with symmetric variables has many characteristics. The existence of symmetric variables is favorable for logic synthesis and optimization of a Boolean function (Rice and Muzio, 2002; Rahaman *et al*., 2002; 2003; Blais *et al*., 2012). For Boolean functions with symmetric variables, there exist special logic synthesis procedures that can be used to improve the design (Young and Muroga, 1985; Kim and Dietmeyer, 1991), and thus the efficiency of the technology of mapping and equivalence testing can be improved (Lai and Pedram, 1992; Cheng and Sadowska, 1993). A Boolean function with symmetric variables has high algebraic immunity. This characteristic has been used in the field of cryptographic function construction (Peng *et al*., 2011; Wang and Peng, 2012). Detection of the symmetric variables of Boolean functions is the basis for all kinds of applications.

The Boolean logic algebra system based on AND-OR-NOT operation (Boole, 1854) and the

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61471314 and 61271124), the Zhejiang Provincial Natural Science Foundation (No. LY13F010001), and the National Key Technology R&D Program of China (Nos. 2013BAH27F01, 2013BAH27F02, and 2013BAH27F03)

ORCID: Ji-zhong SHEN, http://orcid.org/0000-0002-9031-2379

canonical Reed-Muller (CRM) algebra system based on the AND-XOR operation (Muller, 1954; Reed, 1954) are two of main algebra systems of Boolean functions. The graphical method (Mukhopadhyay, 1963), spectral coefficient methods (Hurst, 1977; Rahardja and Falkowski, 1998; Kannurao and Falkowski, 2002), and fast computation method (Mishchenko, 2003) are all used to identify symmetric variables in the Boolean logic algebra system. Since Reed and Muller proposed the CRM algebra system, symmetry detection methods have been studied regarding AND-XOR expansion coefficients (Falkowski and Kannurao, 1999; Kannurao and Falkowski, 2003), and the algorithm for identifying symmetric variables (Li and Shen, 2016) has been developed. There are 12 basic types of symmetry based on the relationship between four co-factors of a Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$ around two variables x_i and x_j, namely $N(x_i|x_j)$, $\mathrm{CN}(x_i|x_j)$, $E(x_i|x_j)$, $\mathrm{CE}(x_i|x_j)$, $S(x_i|x_j)$, $\mathrm{CS}(x_i|x_j)$, $S(x_j|x_i)$, $\mathrm{CS}(x_j|x_i)$, $S(x_i \mid \overline{x}_j)$, $\mathrm{CS}(x_i \mid \overline{x}_j)$, $S(x_j \mid \overline{x}_i)$, and $\mathrm{CS}(x_j \mid \overline{x}_i)$ (Hurst, 1977; Rice and Muzio, 2002). Generally, the graphical method is suitable for the functions where the number of logic variables is not larger than six. The process is complicated for calculating spectrum coefficients and separating spectral coefficients into different groups by the spectral coefficient methods. The fast algorithm is convenient and efficient, but only the $N(x_i|x_j)$, $\mathrm{CN}(x_i|x_j)$, $E(x_i|x_j)$, and $\mathrm{CE}(x_i|x_j)$ symmetries can be identified. The AND-XOR expansion coefficient methods need to separate the expansion coefficients into $n(n-1)/2$ groups. These methods are not suitable for multivariable Boolean functions. The algorithm for identifying symmetric variables is suitable for multivariable Boolean functions, but it cannot be used to identify the symmetric variables of the Boolean function including don't-care terms. Because the Boolean logic algebra system is the most commonly used logic algebra system, study on symmetry detection in that system is very important. So far, research into symmetric detection has not been perfected. To simplify the process for identifying 12 types of symmetric variables of an arbitrary n-variable Boolean function, a new symmetry detection algorithm based on minterm expansion or the truth table is proposed.

2 Representation of the logic function in the Boolean logic algebra system

2.1 Canonical minterm expansion of a Boolean function

An arbitrary n-variable Boolean function can be expressed as a canonical minterm expansion (Boole, 1854) as

$$f(x_0,\ldots,x_c,\ldots,x_{n-2},x_{n-1}) = \sum_{i=0}^{2^n-1} a_i m_i, \tag{1}$$

where the subscript i of m_i can be expressed with binary numbers as $i=(i_{n-1}, i_{n-2}, \ldots, i_c, \ldots, i_0)$, and $\sum$ is the OR operator. The minterm formula can be expressed as $m_i = \dot{x}_{n-1}\dot{x}_{n-2}\cdots\dot{x}_c\cdots\dot{x}_0$, and

$$\dot{x}_c = \begin{cases} \overline{x}_c, & i_c = 0, \\ x_c, & i_c = 1, \end{cases} \quad c = 1,2,\cdots,n. \tag{2}$$

2.2 Canonical minterm expansion of a Boolean function including don't-care terms

In practical digital systems, some combinations of input variables do not appear. These minterms of corresponding combinations are called 'bound terms'. In some combinations of input variables, the output values may take the value 0 or 1. These corresponding minterms are called 'arbitrary terms'. The bound terms and arbitrary terms are both called "don't-care terms" (Hurst, 1978).

An arbitrary n-variable Boolean function including don't-care terms can be expressed as a canonical minterm expansion (Hurst, 1978):

$$f(x_1,\cdots,x_c,\cdots,x_{n-1},x_n) = \sum_{i=0}^{2^n-1} a_i m_i + \sum_{i=0}^{2^n-1} d_i m_i, \quad a_i \cdot d_i \neq 1, \tag{3}$$

where m_i and d_i represent the minterms and don't-care terms, respectively. a_i=1 and d_i=1 indicate the presence of minterms and don't-care terms, respectively.

3 Method for identifying symmetric variables of the Boolean function

3.1 Related definitions

According to Hurst (1977) and Rice and Muzio (2002), there are 12 basic types of symmetry. Six

types of symmetric definitions and six types of anti-symmetric definitions are given in Tables 1 and 2, respectively.

Table 1 Six types of symmetry

Symbol	Symmetric definition
$E(x_i\|x_j)$	$f(x_1,\ldots,0,\ldots,0,\ldots,x_n)=f(x_1,\ldots,1,\ldots,1,\ldots,x_n)$
$N(x_i\|x_j)$	$f(x_1,\ldots,0,\ldots,1,\ldots,x_n)=f(x_1,\ldots,1,\ldots,0,\ldots,x_n)$
$S(x_i\|x_j)$	$f(x_1,\ldots,0,\ldots,1,\ldots,x_n)=f(x_1,\ldots,1,\ldots,1,\ldots,x_n)$
$S(x_i\|\overline{x}_j)$	$f(x_1,\ldots,0,\ldots,0,\ldots,x_n)=f(x_1,\ldots,1,\ldots,0,\ldots,x_n)$
$S(x_j\|x_i)$	$f(x_1,\ldots,1,\ldots,0,\ldots,x_n)=f(x_1,\ldots,1,\ldots,1,\ldots,x_n)$
$S(x_j\|\overline{x}_i)$	$f(x_1,\ldots,0,\ldots,0,\ldots,x_n)=f(x_1,\ldots,0,\ldots,1,\ldots,x_n)$

Table 2 Six types of antisymmetry

Symbol	Antisymmetric definition
CE($x_i\|x_j$)	$f(x_1,\cdots,0,\cdots,0,\cdots,x_n)=\overline{f(x_1,\cdots,1,\cdots,1,\cdots,x_n)}$
CN($x_i\|x_j$)	$f(x_1,\cdots,0,\cdots,1,\cdots,x_n)=\overline{f(x_1,\cdots,1,\cdots,0,\cdots,x_n)}$
CS($x_i\|x_j$)	$f(x_1,\cdots,0,\cdots,1,\cdots,x_n)=\overline{f(x_1,\cdots,1,\cdots,1,\cdots,x_n)}$
CS($x_i\|\overline{x}_j$)	$f(x_1,\cdots,0,\cdots,0,\cdots,x_n)=\overline{f(x_1,\cdots,1,\cdots,0,\cdots,x_n)}$
CS($x_j\|x_i$)	$f(x_1,\cdots,1,\cdots,0,\cdots,x_n)=\overline{f(x_1,\cdots,1,\cdots,1,\cdots,x_n)}$
CS($x_j\|\overline{x}_i$)	$f(x_1,\cdots,0,\cdots,0,\cdots,x_n)=\overline{f(x_1,\cdots,0,\cdots,1,\cdots,x_n)}$

Definition 1 (Minterm expansion coefficient matrix of the subfunction) Using Shannon's law, and considering the arbitrary n-variable Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$, the decomposition of the function around two variables x_i and x_j is given by

$$\begin{aligned} f=&\overline{x}_i\overline{x}_j f_{00}(x_1,\cdots,0,\cdots,0,\cdots,x_n)+\overline{x}_i x_j f_{01}(x_1,\cdots,0,\\ &\cdots,1,\cdots,x_n)+x_i\overline{x}_j f_{10}(x_1,\cdots,1,\cdots,0,\cdots,x_n)\\ &+x_i x_j f_{11}(x_1,\cdots,1,\cdots,1,\cdots,x_n). \end{aligned} \tag{4}$$

The minterm expansion coefficient matrix composed of coefficients $\boldsymbol{a}_i$ in ascending order of subscript i is represented as $\boldsymbol{C}_0$. The minterm expansion coefficient matrices of subfunctions $f_{00}(x_1, \ldots, 0, \ldots, 0, \ldots, x_n)$, $f_{01}(x_1, \ldots, 0, \ldots, 1, \ldots, x_n)$, $f_{10}(x_1, \ldots, 1, \ldots, 0, \ldots, x_n)$, and $f_{11}(x_1, \ldots, 1, \ldots, 1, \ldots, x_n)$ are represented as $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, $\boldsymbol{C}_3$, and $\boldsymbol{C}_4$, respectively.

Definition 2 (Truth table (Hurst, 1978)) A truth table is a representation of a Boolean function. Table 3 gives the truth table of the three-variable Boolean function.

Table 3 Truth table of the three-variable Boolean function

x_1	x_2	x_3	f	x_1	x_2	x_3	f
0	0	0	$\boldsymbol{a}_0$	1	0	0	$\boldsymbol{a}_4$
0	0	1	$\boldsymbol{a}_1$	1	0	1	$\boldsymbol{a}_5$
0	1	0	$\boldsymbol{a}_2$	1	1	0	$\boldsymbol{a}_6$
0	1	1	$\boldsymbol{a}_3$	1	1	1	$\boldsymbol{a}_7$

The truth table is composed of the coding of the variables and the corresponding values of the Boolean function.

Definition 3 (Eigenvalues of a truth table) When the values of the logic variables x_ix_j equal 00, 01, 10, and 11, the corresponding minterm expansion coefficients of a truth table (the corresponding values of the Boolean function) are called the 'eigenvalues of the truth table'.

Definition 4 (Feature coding of x_ix_j) When the values of logic variables x_ix_j equal 00, 01, 10, and 11, the coding of the variables of x_ix_j is called 'feature coding of x_ix_j'.

Definition 5 (Order eigenvalue matrix of a truth table) If the eigenvalues are arranged in ascending order of subscript i of $\boldsymbol{a}_i$, then the coefficients $\boldsymbol{a}_i$ constitute the order eigenvalue matrix of a truth table, represented as $[\boldsymbol{a}_i]_{x_ix_j}$. The $[\boldsymbol{a}_i]_{x_ix_j}$ are represented as $[\boldsymbol{a}_i]_{00}$, $[\boldsymbol{a}_i]_{01}$, $[\boldsymbol{a}_i]_{10}$, and $[\boldsymbol{a}_i]_{11}$ when x_ix_j equals 00, 01, 10, and 11, respectively.

Take a three-variable Boolean function as an example. The corresponding minterm coding and the eigenvalues of the truth table are highlighted in Table 4 when x_2x_3 equals 00, 01, 10, and 11, respectively.

From Table 4, $[\boldsymbol{a}_i]_{x_ix_j}$ around logic variables x_2 and x_3 can be obtained as follows: $[\boldsymbol{a}_i]_{00}=[\boldsymbol{a}_0, \boldsymbol{a}_4]$, $[\boldsymbol{a}_i]_{01}=[\boldsymbol{a}_1, \boldsymbol{a}_5]$, $[\boldsymbol{a}_i]_{10}=[\boldsymbol{a}_2, \boldsymbol{a}_6]$, and $[\boldsymbol{a}_i]_{11}=[\boldsymbol{a}_3, \boldsymbol{a}_7]$.

According to Definitions 1–4, it is easy to obtain the following relationship between $\boldsymbol{C}_0$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, $\boldsymbol{C}_3$, and $[\boldsymbol{a}_i]_{x_ix_j}$: $\boldsymbol{C}_0=[\boldsymbol{a}_i]_{00}$, $\boldsymbol{C}_1=[\boldsymbol{a}_i]_{01}$, $\boldsymbol{C}_2=[\boldsymbol{a}_i]_{10}$, and $\boldsymbol{C}_3=[\boldsymbol{a}_i]_{11}$.

Definition 6 (Determinate elements in $[\boldsymbol{a}_i]_{x_ix_j}$) In $[\boldsymbol{a}_i]_{x_ix_j}$, all the elements except the don't-care terms are called the 'determinate elements'.

Definition 7 (Exclusive operation of a matrix) The exclusive operation of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ means that the corresponding elements conduct exclusive operations.

If $\boldsymbol{A}=[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n]$, $\boldsymbol{B}=[\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$, then $\boldsymbol{A}\oplus\boldsymbol{B}=[\boldsymbol{a}_1\oplus\boldsymbol{b}_1, \boldsymbol{a}_2\oplus\boldsymbol{b}_2, \ldots, \boldsymbol{a}_n\oplus\boldsymbol{b}_n]$.

Table 4 Eigenvalues of the truth table around logic variables x_2 and x_3 for the three-variable Boolean function

x_1	x_2	x_3	f
0	**0**	**0**	$\boldsymbol{a}_0$
0	0	1	$\boldsymbol{a}_1$
0	1	0	$\boldsymbol{a}_2$
0	1	1	$\boldsymbol{a}_3$
1	**0**	**0**	$\boldsymbol{a}_4$
1	0	1	$\boldsymbol{a}_5$
1	1	0	$\boldsymbol{a}_6$
1	1	1	$\boldsymbol{a}_7$

(a) x_2x_3=00

x_1	x_2	x_3	f
0	0	0	$\boldsymbol{a}_0$
0	**0**	**1**	$\boldsymbol{a}_1$
0	1	0	$\boldsymbol{a}_2$
0	1	1	$\boldsymbol{a}_3$
1	0	0	$\boldsymbol{a}_4$
1	**0**	**1**	$\boldsymbol{a}_5$
1	1	0	$\boldsymbol{a}_6$
1	1	1	$\boldsymbol{a}_7$

(b) x_2x_3=01

x_1	x_2	x_3	f
0	0	0	$\boldsymbol{a}_0$
0	0	1	$\boldsymbol{a}_1$
0	**1**	**0**	$\boldsymbol{a}_2$
0	1	1	$\boldsymbol{a}_3$
1	0	0	$\boldsymbol{a}_4$
1	0	1	$\boldsymbol{a}_5$
1	**1**	**0**	$\boldsymbol{a}_6$
1	1	1	$\boldsymbol{a}_7$

(c) x_2x_3=10

x_1	x_2	x_3	f
0	0	0	$\boldsymbol{a}_0$
0	0	1	$\boldsymbol{a}_1$
0	1	0	$\boldsymbol{a}_2$
0	**1**	**1**	$\boldsymbol{a}_3$
1	0	0	$\boldsymbol{a}_4$
1	0	1	$\boldsymbol{a}_5$
1	1	0	$\boldsymbol{a}_6$
1	**1**	**1**	$\boldsymbol{a}_7$

(d) x_2x_3=11

3.2 Principle of the method

3.2.1 Symmetry detection principle of a Boolean function

Theorem 1 If an n-variable Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$ exists with symmetric variables around two logic variables x_i and x_j, then $\boldsymbol{C}_0$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{C}_3$ satisfy the corresponding conditions as shown in Table 5.

Proof Take the type of $E(x_i|x_j)$ symmetry as an example. From the symmetric definition in Table 1, if a Boolean function has $E(x_i|x_j)$ symmetry, then $f(x_1, \ldots, 0, \ldots, 0, \ldots, x_n)=f(x_1, \ldots, 1, \ldots, 1, \ldots, x_n)$, and $\boldsymbol{C}_0$ and $\boldsymbol{C}_3$ satisfy the following condition:

$$\boldsymbol{C}_0=\boldsymbol{C}_3. \tag{5}$$

Then expansion (5) can be rewritten as $\boldsymbol{C}_0\oplus\boldsymbol{C}_3=\boldsymbol{C}_3\oplus\boldsymbol{C}_3=[0, 0, \ldots, 0]$.

By the same argument, the numbered bullets 2–12 in Table 5 can be proved.

Table 5 The minterm expansion coefficient matrix of the subfunction of symmetric variables

Sequence number	Type of symmetry	Condition of the minterm expansion coefficient matrix of the subfunction
1	$E(x_i\|x_j)$	$\boldsymbol{C}_0\oplus\boldsymbol{C}_3$=[0,0,…,0]
2	$N(x_i\|x_j)$	$\boldsymbol{C}_1\oplus\boldsymbol{C}_2$=[0,0,…,0]
3	$S(x_i\|x_j)$	$\boldsymbol{C}_1\oplus\boldsymbol{C}_3$=[0,0,…,0]
4	$S(x_i \mid \overline{x}_j)$	$\boldsymbol{C}_0\oplus\boldsymbol{C}_1$=[0,0,…,0]
5	$S(x_j\|x_i)$	$\boldsymbol{C}_2\oplus\boldsymbol{C}_3$=[0,0,…,0]
6	$S(x_j \mid \overline{x}_i)$	$\boldsymbol{C}_0\oplus\boldsymbol{C}_1$=[0,0,…,0]
7	CE($x_i\|x_j$)	$\boldsymbol{C}_0\oplus\boldsymbol{C}_3$=[1,1,…,1]
8	CN($x_i\|x_j$)	$\boldsymbol{C}_1\oplus\boldsymbol{C}_2$=[1,1,…,1]
9	CS($x_i\|x_j$)	$\boldsymbol{C}_1\oplus\boldsymbol{C}_3$=[1,1,…,1]
10	CS($x_i \mid \overline{x}_j$)	$\boldsymbol{C}_0\oplus\boldsymbol{C}_1$=[1,1,…,1]
11	CS($x_j\|x_i$)	$\boldsymbol{C}_2\oplus\boldsymbol{C}_3$=[1,1,…,1]
12	CS($x_j \mid \overline{x}_i$)	$\boldsymbol{C}_0\oplus\boldsymbol{C}_1$=[1,1,…,1]

From the above discussions, the following deductions can be made:

Deduction 1 If the results of the elements in the $[\boldsymbol{a}_i]_{x_ix_j}$ conducting exclusive operation of the n-variable Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$ satisfy the conditions as shown in Table 6, then this Boolean function exists with the symmetric variables.

Proof Take the conditions $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{11}=[0, 0, \ldots, 0]$ to be satisfied as an example. If the $[\boldsymbol{a}_i]_{x_ix_j}$ of the n-variable Boolean function satisfies $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{11}=[0, 0, \ldots, 0]$, namely

$$\boldsymbol{C}_0\oplus\boldsymbol{C}_3=[0, 0, \ldots, 0], \tag{6}$$

then expansion (6) can be rewritten as

$$\boldsymbol{C}_0\oplus\boldsymbol{C}_3\oplus\boldsymbol{C}_3=[0, 0, \ldots, 0]\oplus\boldsymbol{C}_3,$$
$$\boldsymbol{C}_0\oplus(\boldsymbol{C}_3\oplus\boldsymbol{C}_3)=\boldsymbol{C}_3,$$
$$\boldsymbol{C}_0\oplus[0, 0, \ldots, 0]=\boldsymbol{C}_3,$$
$$\boldsymbol{C}_0=\boldsymbol{C}_3,$$

so $f(x_1, \ldots, 0, \ldots, 0, \ldots, x_n)=f(x_1, \ldots, 1, \ldots, 1, \ldots, x_n)$. From Table 1, the n-variable Boolean function exists with $E(x_i|x_j)$ symmetry.

By the same argument, the numbered bullets 2–12 in Table 6 can be proved.

Table 6 Symmetric conditions of the logic variable

Sequence number	Condition of $[\boldsymbol{a}_i]_{x_i x_j}$	Type of symmetry
1	$[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{11}=[0,0,\ldots,0]$	$E(x_i\|x_j)$
2	$[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{10}=[0,0,\ldots,0]$	$N(x_i\|x_j)$
3	$[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{11}=[0,0,\ldots,0]$	$S(x_i\|x_j)$
4	$[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{10}=[0,0,\ldots,0]$	$S(x_i \mid \overline{x}_j)$
5	$[\boldsymbol{a}_i]_{10}\oplus[\boldsymbol{a}_i]_{11}=[0,0,\ldots,0]$	$S(x_j\|x_i)$
6	$[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{01}=[0,0,\ldots,0]$	$S(x_j \mid \overline{x}_i)$
7	$[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{11}=[1,1,\ldots,1]$	$\mathrm{CE}(x_i\|x_j)$
8	$[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{10}=[1,1,\ldots,1]$	$\mathrm{CN}(x_i\|x_j)$
9	$[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{11}=[1,1,\ldots,1]$	$\mathrm{CS}(x_i\|x_j)$
10	$[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{10}=[1,1,\ldots,1]$	$\mathrm{CS}(x_i \mid \overline{x}_j)$
11	$[\boldsymbol{a}_i]_{10}\oplus[\boldsymbol{a}_i]_{11}=[1,1,\ldots,1]$	$\mathrm{CS}(x_j\|x_i)$
12	$[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{01}=[1,1,\ldots,1]$	$\mathrm{CS}(x_j \mid \overline{x}_i)$

Based on the above discussions, the method for identifying 12 types of symmetric variables based on the $[\boldsymbol{a}_i]_{x_i x_j}$ for the n-variable Boolean function is decomposed into the following three steps:

1. Draw the truth table of the n-variable Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$.

2. According to the truth table, obtain $[\boldsymbol{a}_i]_{x_i x_j}$.

3. Identify the 12 types of symmetric variables by judging whether $[\boldsymbol{a}_i]_{x_i x_j}$ satisfies Deduction 1.

3.2.2 Symmetry detection principle of a Boolean function including don't-care terms

Deduction 2 If the results of the determinate elements in $[\boldsymbol{a}_i]_{x_i x_j}$ conducting exclusive operation of n-variable Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$ satisfy the conditions as shown in Table 6, then this Boolean function exists with the symmetric variables.

Proof Also take the conditions $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{11}=[0, 0, \ldots, 0]$ as an example. According to the property of exclusive operation, the results are all don't-care terms when 0, 1, and don't-care terms conduct the exclusive operation with the don't-care term as shown in Table 7. When the elements in matrices $[\boldsymbol{a}_i]_{00}$ and $[\boldsymbol{a}_i]_{11}$ conduct an exclusive operation, the don't-care terms can be omitted.

The results of the determinate elements in matrices $[\boldsymbol{a}_i]_{00}$ and $[\boldsymbol{a}_i]_{11}$ conducted in an exclusive operation are proved in Deduction 1. So, the n-variable Boolean function including don't-care terms exists with $E(x_i|x_j)$ symmetry.

From Deduction 2, the identification method based on the $[\boldsymbol{a}_i]_{x_i x_j}$ for the n-variable Boolean function including don't-care terms is decomposed into the following three steps:

1. Draw the truth table of the n-variable Boolean function.

2. According to the truth table, obtain $[\boldsymbol{a}_i]_{x_i x_j}$.

3. If the element of $[\boldsymbol{a}_i]_{x_i x_j}$ is a don't-care term, then do not conduct an exclusive operation. Identify the 12 types of symmetric variables by judging whether $[\boldsymbol{a}_i]_{x_i x_j}$ satisfies Deduction 2.

Table 7 Exclusive operation

A	B	$A\oplus B$
0	X	X
1	X	X
X	X	X

3.2.3 Example of the method

Example 1 Consider a three-variable Boolean function $f_1(x_1, x_2, x_3)=\sum m(1,3,4,6,7)$, and apply $[\boldsymbol{a}_i]_{x_i x_j}$ to identify the symmetric variables.

Table 8 is the truth table of $f_1(x_1, x_2, x_3)$.

Table 8 Truth table of f_1

x_1	x_2	x_3	f_1	x_1	x_2	x_3	f_1
0	0	0	0	1	0	0	1
0	0	1	1	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1

From the truth table, $[\boldsymbol{a}_i]_{x_1 x_2}$ is listed as follows: $[\boldsymbol{a}_i]_{00}=[0, 1]$, $[\boldsymbol{a}_i]_{01}=[0, 1]$, $[\boldsymbol{a}_i]_{10}=[1, 0]$, and $[\boldsymbol{a}_i]_{11}=[1, 1]$. According to Deduction 1, $[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{10}=[1, 1]$, and then $\mathrm{CN}(x_1|x_2)$ symmetry exists; $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{10}=[1, 1]$, and then $\mathrm{CS}(x_1 \mid \overline{x}_2)$ symmetry exists; $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{01}=[0, 0]$, and then $S(x_2 \mid \overline{x}_1)$ symmetry exists.

By the same argument, $N(x_1|x_3)$, $\mathrm{CS}(x_1 \mid \overline{x}_3)$, $\mathrm{CS}(x_3 \mid \overline{x}_1)$, $\mathrm{CN}(x_2|x_3)$, $S(x_2 \mid \overline{x}_3)$, and $\mathrm{CS}(x_3 \mid \overline{x}_2)$ symmetries can be identified.

Example 2 Consider a three-variable Boolean function $f_2(x_1,x_2,x_3)=\sum m(0,2,5,6)+d(7)$, and apply $[\boldsymbol{a}_i]_{x_i x_j}$ to identify the symmetric variables.

Table 9 shows the truth table of $f_2(x_1, x_2, x_3)$.

Table 9 Truth table of f_2

x_1	x_2	x_3	f_2	x_1	x_2	x_3	f_2
0	0	0	1	1	0	0	0
0	0	1	0	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	0	1	1	1	X

From the truth table, $[\boldsymbol{a}_i]_{x_1x_2}$ is listed as follows: $[\boldsymbol{a}_i]_{00}$=[1, 0], $[\boldsymbol{a}_i]_{01}$=[1, 0], $[\boldsymbol{a}_i]_{10}$=[0, 1], $[\boldsymbol{a}_i]_{11}$=[1, X]. According to Deduction 2, if $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{11}$=[0, X], then $E(x_1|x_2)$ symmetry exists; if $[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{10}$=[1, 1], then CN($x_1|x_2$) symmetry exists; if $[\boldsymbol{a}_i]_{01}\oplus[\boldsymbol{a}_i]_{11}$=[0, X], then $S(x_1|x_2)$ symmetry exists; if $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{10}$=[1, 1], then $S(x_1|\overline{x}_2)$ symmetry exists; if $[\boldsymbol{a}_i]_{10}\oplus[\boldsymbol{a}_i]_{11}$=[1, X], then CS($x_2|x_1$) symmetry exists; if $[\boldsymbol{a}_i]_{00}\oplus[\boldsymbol{a}_i]_{01}$=[0, 0], then $S(x_2|\overline{x}_1)$ symmetry exists.

By the same argument, $E(x_1|x_3)$, CS($x_1|x_3$), CS($x_3|x_1$), CS($x_3|\overline{x}_1$), CE($x_2|x_3$), $S(x_2|x_3)$, CS($x_3|x_2$), and CS($x_3|\overline{x}_2$) symmetries can be identified.

4 Realization of the symmetry detection algorithm for Boolean functions with or without don't-care terms

Based on the discussions on the principle and example of the proposed symmetry detection method, the 12 types of symmetric variables can be identified by judging $[\boldsymbol{a}_i]_{x_i x_j}$. The symmetry detection algorithm for Boolean functions with or without don't-care terms is decomposed into the following steps:

Step 1: A 2D array $\boldsymbol{y}$ is defined, and is used to conserve the identified function $f(x_1-x_n)$. Array $\boldsymbol{y}$ has 2^n rows and n+1 columns. The first n columns conserve the coding of variables of $f(x_1-x_n)$. The last column conserves the minterm expansion coefficients of $f(x_1-x_n)$. If the Boolean function has don't-care terms, the input values are expressed as a blank. Take Example 2 as an example. A 2D array $\boldsymbol{y}_1$, which is used to conserve the identified function $f(x_1-x_3)$, can be represented as in Table 10. The first c represents the first column of array $\boldsymbol{y}_1$. The second c, third c, and fourth c represent the second, third, and fourth columns of array $\boldsymbol{y}_1$, respectively.

Table 10 Representation of columns of array y_1

First c	Second c	Third c	Fourth c
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	

Step 2: Define 1D arrays $\boldsymbol{C}_0$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{C}_3$. Arbitrarily choose two columns of array $\boldsymbol{y}$, namely the value coding of x_i and x_j ($1\le i<j\le n$), and compare the values of $y[k][i]$ and $y[k][j]$ ($1\le k\le 2^n$) to feature coding.

1. If the values of $y[k][i]$ and $y[k][j]$ equal 0, then store the value of $y[k][n+1]$ in array $\boldsymbol{C}_0$.

2. If the value of $y[k][i]$ equals 0, and the value of $y[k][j]$ equals 1, then store the value of $y[k][n+1]$ in array $\boldsymbol{C}_1$.

3. If the value of $y[k][i]$ equals 1 and the value of $y[k][j]$ equals 0, then store the value of $y[k][n+1]$ in array $\boldsymbol{C}_2$.

4. If the values of $y[k][i]$ and $y[k][j]$ both equal 1, then store the value of $y[k][n+1]$ in array $\boldsymbol{C}_3$.

Step 3: Judge the values of arrays $\boldsymbol{C}_0$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{C}_3$.

1. Conducting the exclusive operation of arrays $\boldsymbol{C}_0$ and $\boldsymbol{C}_3$, when an element of $\boldsymbol{C}_0$ and $\boldsymbol{C}_3$ is blank, the result is also blank. If all operation results of the determinate elements equal 0, then output $E(x_i|x_j)$; if the results equal 1, then output CE($x_i|x_j$).

2. Conducting the exclusive operation of arrays $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, when an element of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ is blank, the result is also blank. If all operation results of the determinate elements equal 0, then output $N(x_i|x_j)$; if the results equal 1, then output CN($x_i|x_j$).

3. Conducting the exclusive operation of arrays $\boldsymbol{C}_1$ and $\boldsymbol{C}_3$, when an element of $\boldsymbol{C}_1$ and $\boldsymbol{C}_3$ is blank, the result is also blank. If all operation results of the determinate elements equal 0, then output $S(x_i|x_j)$; if the results equal 1, then output CS($x_i|x_j$).

4. Conducting the exclusive operation of arrays $\boldsymbol{C}_0$ and $\boldsymbol{C}_2$, when an element of $\boldsymbol{C}_0$ and $\boldsymbol{C}_2$ is blank, the

result is also blank. If all operation results of the determinate elements equal 0, then output $S(x_i \mid \overline{x}_j)$; if the results equal 1, then output $\mathrm{CS}(x_i \mid \overline{x}_j)$.

5. Conducting the exclusive operation of arrays $\boldsymbol{C}_2$ and $\boldsymbol{C}_3$, when an element of $\boldsymbol{C}_2$ and $\boldsymbol{C}_3$ is blank, the result is also blank. If all operation results of the determinate elements equal 0, then output $S(x_j|x_i)$; if the results equal 1, then output $\mathrm{CS}(x_j|x_i)$.

6. Conducting the exclusive operation of arrays $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$, when an element of $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$ is blank, the result is also blank. If all operation results of the determinate elements equal 0, then output $S(x_j \mid \overline{x}_i)$; if the results equal 1, then output $\mathrm{CS}(x_j \mid \overline{x}_i)$.

Step 4: Repeat step 2.

Step 5: Terminate the loop.

The flowchart of the program implemented in C language is shown in Fig. 1.

5 Experimental results

The proposed algorithm was implemented in C language on the Linux platform and run on an IBM P5 560q server (8 cores, 64 GB memory, using the Linux enterprise server operating system, SCSI disk). To verify the proposed algorithm, it was tested on MCNC91 benchmarks. The results of the test are shown in Table 11 for single output functions up to 18 variables. The program of the algorithm can output the specific types of symmetry. Take test function 'qov' as an example. The types of symmetry $E(x_1|x_2)$, $N(x_7|x_8)$, $S(x_3|x_4)$, $S(x_9|x_{10})$, $S(x_5 \mid \overline{x}_6)$, $S(x_9|x_8)$, $\mathrm{CE}(x_2|x_3)$, $\mathrm{CN}(x_1|x_5)$, $\mathrm{CS}(x_3|x_7)$, $\mathrm{CS}(x_6 \mid \overline{x}_9)$, $\mathrm{CS}(x_8|x_4)$, $\mathrm{CS}(x_{10}|x_3)$, and $\mathrm{CS}(x_6 \mid \overline{x}_2)$ can be identified. The results identified by programming the algorithm are consistent with the conclusions identified by judging $[\boldsymbol{a}_i]_{x_i x_j}$ of the function. The validity of the algorithm is also proven. The symmetry detection time for the same number of logic variables in the test function depends on the number of symmetric variables. The more the symmetric variables, the greater the detection time required, as indicated by test functions 'prt' and 'yy9' in Table 11. The required detection time obviously increases with the increase of the number of logic variables in the test function.

6 Analysis and comparison of different identification methods

The performances of different methods for identifying symmetric variables are listed for comparison in Table 12. The graphical method is intuitive, the functions where the number of logic variables is not larger than six. The process is complicated for calculating the spectrum coefficients and separating the spectral coefficients into $n(n-1)/2$ groups by the spectral coefficient methods. The AND-XOR expansion coefficient methods also need to separate the AND-XOR expansion coefficients into $n(n-1)/2$

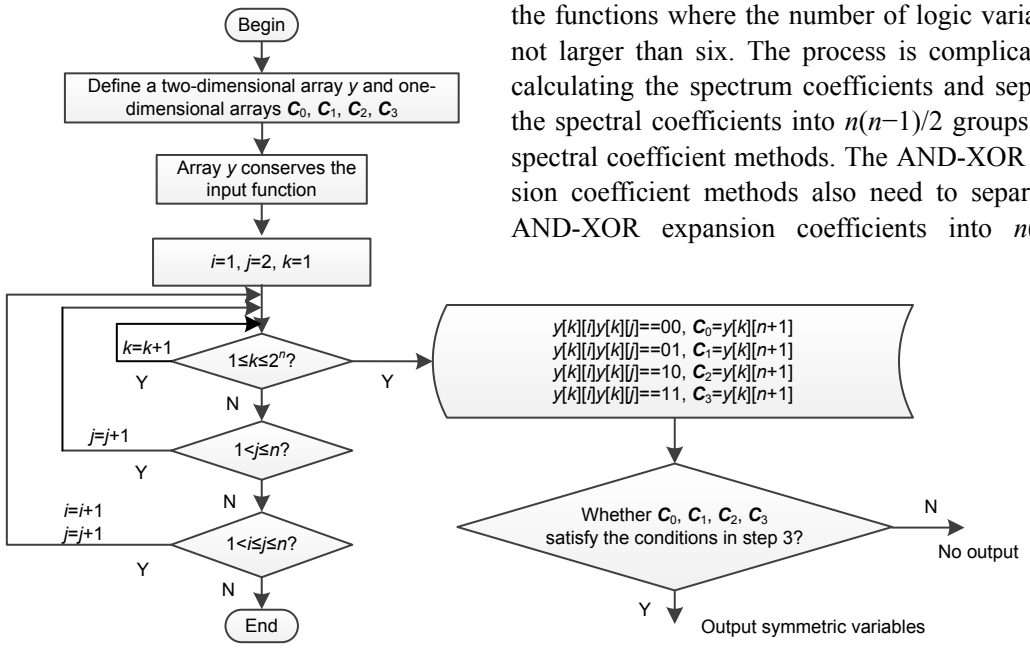

Fig. 1 Flowchart of the program

Table 11 Experimental results for the test function

Name	n	t	$E(x_i\|x_j)$	$N(x_i\|x_j)$	$S(x_i\|x_j)$	$S(x_i\|\overline{x}_j)$	$S(x_j\|x_i)$	$S(x_j\|\overline{x}_i)$	$CE(x_i\|x_j)$	$CN(x_i\|x_j)$	$CS(x_i\|x_j)$	$CS(x_i\|\overline{x}_j)$	$CS(x_j\|x_i)$	$CS(x_j\|\overline{x}_i)$
test1	3	1.12	0	1	0	1	0	0	0	2	1	2	0	2
test2	3	2.11	2	0	2	1	0	1	1	1	1	0	3	2
d	4	4.43	1	0	0	1	0	1	1	1	3	1	0	1
ae5	5	7.88	1	0	1	0	1	1	2	0	1	1	1	0
6tt	6	9.74	1	1	1	1	0	2	0	2	0	1	0	1
7fc	7	11.32	1	1	1	0	1	0	2	0	2	1	1	0
up8	8	15.77	2	1	0	1	1	0	2	1	2	1	1	0
prt	9	18.14	2	1	1	2	2	1	0	1	0	1	0	1
yy9	9	27.17	1	2	3	3	1	2	2	2	3	1	2	2
qov	10	29.16	1	1	2	1	1	0	1	1	1	1	2	1
ipt	11	31.54	1	0	1	2	1	1	1	1	1	0	1	1
sw1	12	33.08	1	2	0	2	1	1	2	1	2	2	2	2
shu	13	45.83	2	3	5	1	3	2	3	0	0	1	0	1
we2	14	58.77	3	2	3	4	1	6	1	1	4	1	1	3
iyr	15	69.88	4	2	3	5	6	2	1	1	5	6	1	1
qro	16	88.15	3	4	2	9	3	3	8	2	1	2	3	4
ll3	17	142.00	7	2	3	2	2	7	6	4	4	5	4	2
vc7	18	193.00	5	6	5	4	3	5	4	6	4	2	4	2

'name', 'n', and 't' express the test function, the number of logic variables in the test function, and the symmetry detection time (in seconds), respectively. Columns 4–15 give the statistical numbers of symmetric variables existing in the test function for the 12 types of symmetry. Test functions 'test1' and 'test2' are the same as those used in Examples 1 and 2

Table 12 Comparison of six methods

Detection method	Applicability of the number of logic variables	Number of detection types	Whether suitable for the Boolean function including don't-care terms	Complexity of the identification process	
				Foundation work	Time for condition judgment
Graphical method (Mishchenko, 2003)	Usually $n\leq6$	12	No	Draw the K-map of the n-variable Boolean function	$6n(n-1)$
Spectral coefficient methods (Muller, 1954; Mukhopadhyay, 1963; Peng *et al.*, 2011)	Usually $n\leq6$	12	No	Calculate the spectral coefficients and then separate the spectral coefficients into $n(n-1)/2$ groups	$6n(n-1)$
AND-XOR expansion coefficient methods (Reed, 1954; Rice and Muzio, 2002)	Usually $n\leq6$	12	No	Separate the AND-XOR expansion coefficients into $n(n-1)/2$ groups	$6n(n-1)$
Fast computation method (Rahardja and Falkowski, 1998)	No limit	4	No	Input the minterm expansion coefficients	$12n(n-1)$
Algorithm for identifying symmetric variables (Wang and Peng, 2012)	No limit	12	No	Input the number of logical variables in the test function and the AND-XOR expansion coefficients	$3n(n-1)$
Proposed method	No limit	12	Yes	Input the minterm expansion coefficients	$3n(n-1)$

Note: n represents the number of logic variables in the test function

groups. It is difficult to classify the coefficients by spectral coefficients methods and AND-XOR expansion coefficient methods when the number of logic variables exceeds six. The fast computation method is suitable for arbitrary n-variable Boolean functions, but can identify only $N(x_i|x_j)$, $E(x_i|x_j)$, $\mathrm{CN}(x_i|x_j)$, $\mathrm{CE}(x_i|x_j)$ symmetries. Research on the symmetric detection of Boolean functions including don't-care terms was not carried out in all the mentioned methods. The proposed method is the optimal detection method in terms of the applicability of the number of logic variables, Boolean functions including don't-care terms, detection type, and complexity of the identification process. If a Boolean function is expressed as minterm expansion, the proposed method is obviously the most efficient. If a Boolean function is expressed as AND-XOR expansion, by the transformation from AND-XOR expansion to minterm expansion (Wu *et al*., 1982), the proposed algorithm is also the most appropriate method.

Up to now, there have been no publications on an algorithm that can be programmed for identifying 12 types of symmetric variables in the Boolean logic algebra system for comparison.

7 Conclusions

The Boolean logic algebra system is the most commonly used algebra system for Boolean functions. Currently, the main tool of logic synthesis and the theory of constructing cryptographic functions in the Boolean logic algebra system are mostly based on the AND-OR-NOT operation. In this paper, a new symmetric detection algorithm was proposed based on the order eigenvalue matrix. The algorithm has been implemented in C language and tested on MCNC91 benchmarks. The proposed algorithm avoids the restriction by the number of logic variables in graphical methods, spectral coefficient methods, and AND-XOR expansion coefficient methods, and solves the problem of completeness in the fast computation method. Application results show that, compared with existing methods, the new algorithm is an optimal detection method in terms of the applicability of the number of logic variables, the Boolean function including don't-care terms, detection type, and complexity of the identification process.

References

Blais, E., Weinstein, A., Yoshida, Y., 2012. Partially symmetric functions are efficiently isomorphism-testable. Proc. IEEE 53rd Annual Symp. on Foundation of Computer Science, p.551-560. https://doi.org/10.1109/FOCS.2012.53

Boole, G., 1854. Investigation of the Lows of Thought. New York, USA.

Cheng, D.I., Sadowska, M., 1993. Verifying equivalence of functions with unknown input correspondence. *IEEE Trans. Comput.*, **41**(6):81-85. https://doi.org/10.1109/EDAC.1993.386496

Falkowski, B.J., Kannurao, S., 1999. Identification of Boolean symmetries in spectral domain of Reed-Muller transform. *Electron. Lett.*, **35**(16):1315-1316. https://doi.org/10.1049/el:19990955

Hurst, S.L., 1977. Detection of symmetries in combinatorial functions by spectral means. *IEE J. Electron. Circ. Syst.*, **1**(5):173-180. https://doi.org/10.1049/ij-ecs:19770026

Hurst, S.L., 1978. The Logic Processing of Digital Systems. Crane-Russak, New York.

Kannurao, S., Falkowski, B., 2002. Identification of complement single variable symmetry in Boolean functions through Walsh transform. Proc. IEEE Int. Symp. on Circuits and Systems, p.745-748. https://doi.org/10.1109/ISCAS.200.101081

Kannurao, S., Falkowski, B., 2003. Single variable symmetry conditions in Boolean functions through Reed-Muller transform. Proc. IEEE Int. Symp. on Circuits and Systems, p.680-683. https://doi.org/10.1109/ISCAS.2003.1206200

Kim, B., Dietmeyer, D., 1991. Multilevel logic synthesis of symmetric switching functions. *IEEE Trans. Comput.-Aided Des.*, **10**(9):436-446. https://doi.org/10.1109/43.75627

Lai, Y., Pedram, M., 1992. Boolean matching using binary decision diagrams with application to logic synthesis and verification. IEEE Int. Conf. on Computer Design: VLSI in Computers and Processors, p.452-458. https://doi.org/10.1109/ICCD.1992.276313

Li, X., Shen, J., 2016. An algorithm for identifying symmetric variables in the canonical Reed–Muller algebra system. *J. Circ. Syst. Comput.*, **25**(10):1650126. https://doi.org/10.1142/S0218126616501267

Mishchenko, A., 2003. Fast computation of symmetries in Boolean functions. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.*, **22**(11):1588-1593. https://doi.org/10.1109/TCAD.2003.818371

Mukhopadhyay, A., 1963. Detection of total or partial symmetry of a switching function with the use of decomposition charts. *IEEE Trans. Electron. Comput.*, **EC**(12): 553-557. https://doi.org/10.1109/PGEC.1963.263654

Muller, D.E., 1954. Application of Boolean algebra to switching circuit design and error detection. *IRE Trans. Electron. Comput*., **EC**(3):6-14. https://doi.org/10.1109/IREPGELC.1954.6499441

Peng, J., Wu, Q., Kan, H., 2011. On symmetric Boolean functions with high algebraic immunity on even number of variables. *IEEE Trans. Inform. Theory*, **157**(10):7205-7220. https://doi.org/10.1109/TIT.2011.2132113

Rahaman, H., Das, D.K., Bhattacharya, B.B., 2002. A new synthesis of symmetric functions. Proc. 7th Asia and South Pacific and 15th Int. Conf. on VLSI Design, p.160-165. https://doi.org/10.1109/ASPDAC.2002.994910

Rahaman, H., Das, D.K., Bhattacharya, B.B., 2003. Mapping symmetric functions to hierarchical modules for path-delay fault testability. Proc. 12th Asian Test Symp., p.284-289. https://doi.org/10.1109/ATS.2003.1250824

Rahardja, S., Falkowski, B., 1998. Symmetry conditions of Boolean functions in complex Hadamard transform electron. *Electron. Lett.*, **34**(17):1634-1635. https://doi.org/10.1049/el:19981164

Reed, I.S., 1954. A class of multiple-error-correcting code and the decoding scheme. *IRE Trans. Electron. Comput.*, **EC**(4):38-49. https://doi.org/10.1109/TIT.1954.1057465

Rice, J.E., Muzio, J.C., 2002. Antisymmetries in the realization of Boolean functions. Proc. IEEE Int. Symp. on Circuits and Systems, p.69-72. https://doi.org/10.1109/ISCAS.2002.1010390

Wang, H., Peng, J., 2012. On 2*k*-variable symmetric Boolean functions with maximum algebraic immunity. *IEEE Trans. Inform. Theory*, **58**(8):5612-5624. https://doi.org/10.1109/TIT.2012.2201350

Wu, X., Chen, X., Hurst, S.L., 1982. Mapping of Reed-Muller coefficients and the minimisation of exclusive OR-switching functions. *IEE Proc. E*, **129**(1):15-20. https://doi.org/10.1049/ip-e:19820004

Young, M., Muroga, S., 1985. Symmetric minimal covering problem and minimal PLA's with symmetric variables. *IEEE Trans. Comput.*, **34**(6):312-318. https://doi.org/10.1109/TC.1985.5009404