Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An algorithm for identifying symmetric variables in the canonical OR-coincidence algebra system*

Xiao-hua LI[1], Ji-zhong SHEN[†‡2]

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

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

[†]E-mail: jzshen@zju.edu.cn

Received Mar. 16, 2014; Revision accepted May 26, 2014; Crosschecked Nov. 13, 2014

Abstract: To simplify the process for identifying 12 types of symmetric variables in the canonical OR-coincidence (COC) algebra system, we propose a new symmetry detection algorithm based on OR-NXOR expansion. By analyzing the relationships between the coefficient matrices of sub-functions and the order coefficient subset matrices based on OR-NXOR expansion around two arbitrary logical variables, the constraint conditions of the order coefficient subset matrices are revealed for 12 types of symmetric variables. Based on the proposed constraints, the algorithm is realized by judging the order characteristic square value matrices. The proposed method avoids the transformation process from OR-NXOR expansion to AND-OR-NOT expansion, or to AND-XOR expansion, and solves the problem of completeness in the d_j-map method. The application results show that, compared with traditional methods, the new algorithm is an optimal detection method in terms of applicability of the number of logical variables, detection type, and complexity of the identification process. The algorithm has been implemented in C language and tested on MCNC91 benchmarks. Experimental results show that the proposed algorithm is convenient and efficient.

Key words: Symmetric variable, d_j-map, Canonical OR-coincidence algebra system, Boolean function
doi:10.1631/jzus.C1400093 **Document code:** A **CLC number:** TN431

1 Introduction

The symmetry of logical variables is a very important property. Symmetric variables exist in most Boolean functions. The symmetry of logical variables has been widely used in the fields of logical synthesis and optimization, ordered binary decision diagram (OBDD) simplification, fault detection, circuit analysis, construction of cryptographic functions, etc. (Heinrich-Litan and Molitor, 2000; Rice and Muzio, 2002; Rahaman *et al.*, 2003; Acharya *et al.*, 2011; Peng *et al.*, 2011; Shpilka amd Tal, 2011; Blais *et al.*, 2012; Kowshik and Kumar, 2013). Detection of the symmetric variables is the basis for all types of applications. There are three main algebra systems of Boolean functions, namely the Boolean logic algebra system based on AND-OR-NOT expansion (Mukhopadhyay, 1963), the canonical Reed-Muller (CRM) algebra system based on AND-XOR expansion (Wu *et al.*, 1982), and the canonical OR-coincidence (COC) algebra system based on OR-NXOR expansion (Cheng *et al.*, 2003). There are also the graphic method (Mukhopadhyay, 1963), the spectral coefficient methods (Hurst, 1977; Rahardja and Falkowski, 1998; Kannurao and Falkowski, 2002), and the tabular method (Lian *et al.*, 2005), which are all used to identify symmetric variables in the traditional AND-OR-NOT algebra system. Since

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61471314 and 61271124), the National Social Science Foundation of China (No. 12AZD121), the Zhejiang Provincial Natural Science Foundation of China (No. LY13F010001), and the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Nos. 2013BAH27F01 and 2013BAH27F02)

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

Reed (1954) and Muller (1954) proposed the CRM algebra system, the detection algorithm has been researched in the AND-XOR expansion (Falkowski and Kannurao, 1999; Kannurao and Falkowski, 2003). Cheng *et al.* (2003) proposed the COC algebra system which has the advantages of fault detection and simplified design. Zhao *et al.* (2006) discussed the detection method based on the d_j-map in the COC algebra system. There are 12 basic types of symmetry based on a relationship between four co-factors of the Boolean function around two arbitrary variables x_i and x_j. The d_j-map method can be used to identify the type of N($x_i|x_j$) symmetry, but cannot identify the type of CN($x_i|x_j$), E($x_i|x_j$), CE($x_i|x_j$), S($x_i|x_j$), CS($x_i|x_j$), S($x_j|x_i$), CS($x_j|x_i$), S($x_i\,|\,\overline{x}_j$), CS($x_i\,|\,\overline{x}_j$), S($x_j\,|\,\overline{x}_i$), or CS($x_j\,|\,\overline{x}_i$) symmetry. Up to now, the research into symmetric detection has not been perfected for the COC algebra system. To simplify the process for identifying the 12 types of symmetric variables in the COC algebra system, a new symmetry detection algorithm based on OR-NXOR expansion is proposed.

2 OR-NXOR expansion of the Boolean function and its coefficient feature

In the COC algebra system, the expansion of the n-variable Boolean function can be written as

$$\begin{aligned} f(x_1\sim x_n)=&(d_0+x_1^1+x_2^1+\cdots+x_n^1)\\ &\odot(d_1+x_1^0+x_2^1+\cdots+x_n^1)\odot\cdots\\ &\odot(d_{2^n-1}+x_1^0+x_2^0+\cdots+x_n^0), \end{aligned} \tag{1}$$

where $x_i^0=0$, $x_i^1=x_i$, $d_j\in\{0, 1\}$ (j=0, 1, …, 2^n-1), and '$\odot$' represents the coincidence operation.

Take a three-variable Boolean function as an example:

$$\begin{aligned} f(x_1\sim x_n)=&(d_0+x_1+x_2+x_3)\odot(d_1+x_1+x_2)\odot(d_2+x_1+x_3)\\ &\odot(d_3+x_1)\odot(d_4+x_2+x_3)\odot(d_5+x_2)\\ &\odot(d_6+x_3)\odot d_7. \end{aligned} \tag{2}$$

To satisfy the correspondence between the subscript of coefficients and the variables in the sum of the terms, expansion (1) can be rewritten as

$$\begin{aligned} f(x_1\sim x_n)=&(h_{12\cdots n}+x_1^1+x_2^1+\cdots+x_n^1)\\ &\odot(h_{2\cdots n}+x_1^0+x_2^1+\cdots+x_n^1)\odot\cdots\\ &\odot(h_0+x_1^0+x_2^0+\cdots+x_n^0). \end{aligned} \tag{3}$$

Based on Eqs. (1) and (3), the following transformational relationship between coefficients d_j and h_j of a three-variable Boolean function can be obtained: If we transform the subscript of d_j to the corresponding binary number, the corresponding variables of the 0 value coding can be used to obtain the subscript of h_j. For example, the binary subscript of d_1 is '001', and the corresponding variables of the 0 value coding are x_1 and x_2. Thus, the corresponding coefficient h_j is represented as h_{12}.

Using Shannon's law, and considering an arbitrary Boolean function $f(x_1, …, x_i, …, x_j, …, x_n)$, the decomposition of the function around two arbitrary variables x_i and x_j ($1\le i<j\le n$) is given by

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

Definition 1 The OR-NXOR expansion coefficient matrices of the sub-functions $f_{00}(x_1, …, 0, …, 0, …, x_n)$, $f_{01}(x_1, …, 0, …, 1, …, x_n)$, $f_{10}(x_1, …, 1, …, 0, …, x_n)$, and $f_{11}(x_1, …, 1, …, 1, …, x_n)$ are represented as $\boldsymbol{C}_0$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{C}_3$, respectively.

Definition 2 The order OR-NXOR expansion coefficient matrix of the Boolean function $f(x_1, …, x_i, …, x_j, …, x_n)$, which is composed of coefficients d_j with the subscript in an ascending order, is represented as $\boldsymbol{C}_I$. The subscript of $\boldsymbol{C}_I$ is titled the index of the variables in the sum of the terms. Regard $\boldsymbol{C}_I$ as a block matrix and decompose it to submatrices corresponding to two arbitrary variables x_i and x_j. The order OR-NXOR expansion coefficient subset matrices of $\boldsymbol{C}_I$ ($i\in I, j\in I$), $\boldsymbol{C}_I$ ($i\in I, j\notin I$), $\boldsymbol{C}_I$ ($i\notin I, j\in I$), and $\boldsymbol{C}_I$ ($i\notin I, j\notin I$) are represented as $\boldsymbol{C}^0$, $\boldsymbol{C}^1$, $\boldsymbol{C}^2$, and $\boldsymbol{C}^3$, respectively.

Take a three-variable Boolean function as an example. Decomposing the order OR-NXOR expansion coefficient matrix around variables x_2 and x_3, the order OR-NXOR expansion coefficient subset matrices $\boldsymbol{C}^0$, $\boldsymbol{C}^1$, $\boldsymbol{C}^2$, and $\boldsymbol{C}^3$ can be obtained as follows:

$$\boldsymbol{C}^0=[h_{123}, h_{23}]=[d_0, d_4],\ \boldsymbol{C}^1=[h_{12}, h_2]=[d_1, d_5],$$
$$\boldsymbol{C}^2=[h_{13}, h_3]=[d_2, d_6],\ \boldsymbol{C}^3=[h_1, h_0]=[d_3, d_7].$$

Theorem 1 The n-variable Boolean function $f(x_1 \sim x_n)$ which represents the relationship between the OR-NXOR expansion coefficient matrices of the sub-functions and the order OR-NXOR expansion coefficient subset matrices around two arbitrary logical variables x_i and x_j satisfies the following equation:

$$[\boldsymbol{C}_0\ \boldsymbol{C}_1\ \boldsymbol{C}_2\ \boldsymbol{C}_3]=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\begin{bmatrix}1&0&0&0\\1&1&0&0\\1&0&1&0\\1&1&1&1\end{bmatrix}. \quad (5)$$

Proof According to Kannurao and Falkowski (2003), the relationship between the AND-XOR expansion coefficient matrices of the sub-functions and the order AND-XOR expansion coefficient subset matrices around two arbitrary logical variables x_i and x_j satisfies the following equation:

$$[\boldsymbol{C}_0\ \boldsymbol{C}_1\ \boldsymbol{C}_2\ \boldsymbol{C}_3]=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3](\boldsymbol{T}^{\otimes 2})^{\mathrm{T}}, \quad (6)$$

where $\boldsymbol{T}=\begin{pmatrix}1&0\\1&1\end{pmatrix}$. The transformational relationship between the AND-XOR expansion coefficients and the minterm expansion coefficients can be written as

$$\boldsymbol{b}_n=\begin{pmatrix}\mathbf{1}&\mathbf{0}\\\mathbf{1}&\mathbf{1}\end{pmatrix}^{\otimes n}*\boldsymbol{c}_n=\boldsymbol{T}^{\otimes n}*\boldsymbol{c}_n, \quad (7)$$

where $\boldsymbol{b}_n$ represents a column matrix constituted by coefficients b_j, $\boldsymbol{c}_n$ represents a column matrix constituted by minterm expansion coefficients, '*' represents the corresponding element of the two matrices multiplication and then adding modulo-2, and $\otimes n$ represents n times the Kronecker matrix product.

The transformational relationship between the OR-NXOR expansion coefficients and the maxterm expansion coefficients can be written as

$$\boldsymbol{d}_n=\begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes n}*\boldsymbol{M}_n=\begin{pmatrix}1&1\\0&1\end{pmatrix}^{\otimes n}*c_n=\left(\boldsymbol{T}^{\mathrm{T}}\right)^{\otimes n}*\boldsymbol{c}_n, \quad (8)$$

where $\boldsymbol{d}_n$ represents a column matrix constituted by coefficients d_j, and $\boldsymbol{M}_n$ represents a column matrix constituted by maxterm expansion coefficients. $\boldsymbol{M}_n$ equals $\boldsymbol{c}_n$ because they are the values of the function under the same input. By comparing Eq. (7) with Eq. (8), and applying $\boldsymbol{T}^{\mathrm{T}}$ instead of $\boldsymbol{T}$ in Eq. (7), the relationship between the OR-NXOR expansion coefficient matrices of the sub-functions and the order OR-NXOR expansion coefficient subset matrices around two arbitrary logical variables x_i and x_j can be revealed as follows:

$$\begin{aligned}[\boldsymbol{C}_0\ \boldsymbol{C}_1\ \boldsymbol{C}_2\ \boldsymbol{C}_3]&=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\left(\boldsymbol{T}^{\mathrm{T}\otimes 2}\right)^{\mathrm{T}}\\&=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\left(\left(\boldsymbol{T}^{\mathrm{T}}\right)^{\mathrm{T}}\right)^{\otimes 2}\\&=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\boldsymbol{T}^{\otimes 2}\\&=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\begin{pmatrix}1&0\\1&1\end{pmatrix}^{\otimes 2}\\&=[\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\begin{bmatrix}1&0&0&0\\1&1&0&0\\1&0&1&0\\1&1&1&1\end{bmatrix}.\end{aligned}$$

3 Symmetric and antisymmetric conditions of logical variables

According to Falkowski and Kannurao (1999) and Kannurao and Falkowski (2003), there are 12 basic types of symmetry, four of which are two-variable symmetry and eight of which are single variable symmetry. Six types of symmetric definitions and six types of antisymmetric definitions are formulated in Tables 1 and 2, respectively.

Theorem 2 If an n-variable Boolean function $f(x_1, \dots, x_i, \dots, x_j, \dots, x_n)$ exists with symmetric variables, then the order OR-NXOR expansion coefficient subset matrices around two arbitrary logical variables x_i and x_j satisfy the conditions listed in Table 3.

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 the type of $E(x_i|x_j)$ symmetry, then $f(x_1, \dots, 0, \dots, 0, \dots, x_n)=f(x_1, \dots, 1, \dots, 1, \dots, x_n)$, and hence $\boldsymbol{C}_0=\boldsymbol{C}_3$. According to Theorem 1,

$$\boldsymbol{C}_0=\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2 \odot \boldsymbol{C}^3,\ \boldsymbol{C}_3=\boldsymbol{C}^3,$$
$$\text{or } \boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2 \odot \boldsymbol{C}^3=\boldsymbol{C}^3.$$

Table 1 The six types of symmetry

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

Table 2 The six types of antisymmetry

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

Table 3 The order OR-NXOR expansion coefficient subset matrices of symmetric variables

Type of symmetry	Satisfied conditions of the order OR-NXOR expansion coefficient subset matrices
$E(x_i\|x_j)$	$\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2=[1 1 \dots 1]^T$
$N(x_i\|x_j)$	$\boldsymbol{C}^1 \odot \boldsymbol{C}^2=[1 1 \dots 1]^T$
$S(x_i\|x_j)$	$\boldsymbol{C}^1=[1 1 \dots 1]^T$
$S(x_i\|\bar{x}_j)$	$\boldsymbol{C}^0 \odot \boldsymbol{C}^1=[1 1 \dots 1]^T$
$S(x_j\|x_i)$	$\boldsymbol{C}^2=[1 1 \dots 1]^T$
$S(x_j\|\bar{x}_i)$	$\boldsymbol{C}^0 \odot \boldsymbol{C}^2=[1 1 \dots 1]^T$
$CE(x_i\|x_j)$	$\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2=[1 0 \dots 0]^T$
$CN(x_i\|x_j)$	$\boldsymbol{C}^1 \odot \boldsymbol{C}^2=[1 0 \dots 0]^T$
$CS(x_i\|x_j)$	$\boldsymbol{C}^1=[1 0 \dots 0]^T$
$CS(x_i\|\bar{x}_j)$	$\boldsymbol{C}^0 \odot \boldsymbol{C}^1=[1 0 \dots 0]^T$
$CS(x_j\|x_i)$	$\boldsymbol{C}^2=[1 0 \dots 0]^T$
$CS(x_j\|\bar{x}_i)$	$\boldsymbol{C}^0 \odot \boldsymbol{C}^2=[1 0 \dots 0]^T$

So, $\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2=[1\ 1\ \dots\ 1]^T$.

By the same argument, the other 11 symmetry types in Table 3 can be proved.

4 Algorithm for identifying symmetric variables based on d_j-map

4.1 Principle of the algorithm

Definition 3 (d_j-map (Cheng *et al.*, 2003)) The d_j-map is a representation of a Boolean function in the COC algebra system. Fig. 1 gives the d_j-map of the three-variable Boolean function.

The d_j-map has the following features:

1. Each square corresponds to the sum of the terms. If a d_j=0 entry is included within an x_i=0 area, then variable x_i appears in uncomplemented form in the sum of the terms. On the other hand, if within an x_i=1 area, then variable x_i does not appear in the sum of the terms.

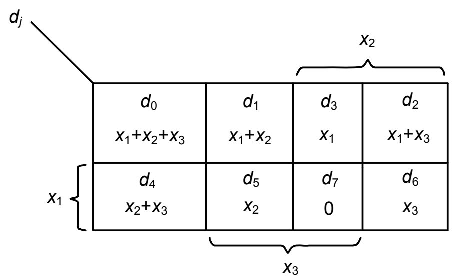

Fig. 1 The d_j-map of a three-variable Boolean function

2. Adjacent squares differ in one variable.

When the logical variables x_ix_j equal 00, 01, 10, or 11, the corresponding squares are titled the characteristic squares of the d_j-map. The filled values of the characteristic squares are titled the characteristic square values.

Definition 4 (The order characteristic square value matrix $[d_j]_{x_ix_j}$) If the characteristic square values are arranged in ascending order of the subscript of d_j, then cofficients d_j constitute the order characteristic square value matrix, represented as $[d_j]_{x_ix_j}$. The order characteristic square value matrices are represented as $[d_j]_{00}$, $[d_j]_{01}$, $[d_j]_{10}$, and $[d_j]_{11}$ when x_ix_j equal 00, 01, 10, and 11, respectively.

From the d_j-map features and Definitions 1–4, it is easy to obtain the following relationship between the order OR-NXOR expansion coefficient subset matrices $\boldsymbol{C}^0$, $\boldsymbol{C}^1$, $\boldsymbol{C}^2$, $\boldsymbol{C}^3$, and $[d_j]_{x_ix_j}$:

$$C^0=[d_j]_{00},\ C^1=[d_j]_{01},\ C^2=[d_j]_{10},\ C^3=[d_j]_{11}.$$

Taking a three-variable Boolean function as an example, the characteristic squares of the d_j-map around logical variables x_1 and x_2 demonstrated by oblique lines are shown in Fig. 2.

From the above discussions, the following deduction can be obtained:

Deduction 1 If the order characteristic square value matrix $[d_j]_{x_ix_j}$ of the n-variable Boolean function $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n)$ satisfies the conditions listed in Table 4, then there exist corresponding symmetric variables.

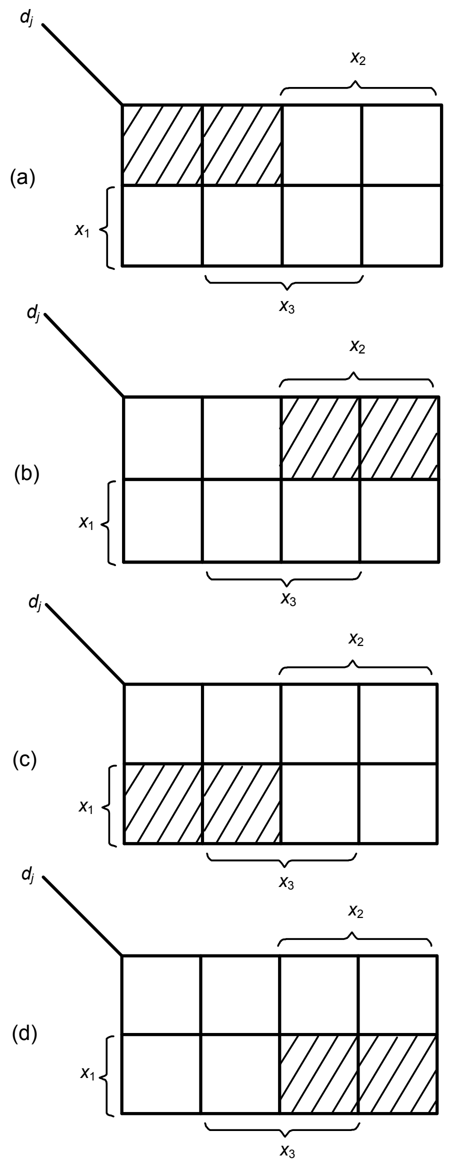

Fig. 2 The characteristic squares around logical variables x_1 and x_2
(a) x_1x_2=00; (b) x_1x_2=01; (c) x_1x_2=10; (d) x_1x_2=11

Proof Take the satisfied conditions $[d_j]_{00} \odot [d_j]_{01} \odot [d_j]_{10}=[1\ 1\ \ldots\ 1]^{\mathrm{T}}$ as an example. If the order characteristic square value matrix $[d_j]_{x_ix_j}$ of the n-variable Boolean function satisfies $[d_j]_{00} \odot [d_j]_{01} \odot [d_j]_{10}=[1\ 1\ \ldots\ 1]^{\mathrm{T}}$, namely

$$\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2 = [1 1 \ldots 1]^{\mathrm{T}}, \tag{9}$$

then expansion (9) can be rewritten as

$$\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2 \odot \boldsymbol{C}^3 = [1 1 \ldots 1]^{\mathrm{T}} \odot \boldsymbol{C}^3 = \boldsymbol{C}^3.$$

According to Theorem 1, $\boldsymbol{C}^0 \odot \boldsymbol{C}^1 \odot \boldsymbol{C}^2 \odot \boldsymbol{C}^3=\boldsymbol{C}_0$ and $\boldsymbol{C}^3=\boldsymbol{C}_3$. So, $\boldsymbol{C}_0=\boldsymbol{C}_3$, namely $f(x_1, \ldots, 0, \ldots, 0, \ldots, x_n)=f(x_1, \ldots, 1, \ldots, 1, \ldots, x_n)$. From Table 1, the n-variable Boolean function exists with the E($x_i|x_j$) symmetry.

By the same argument, the other 11 satisfied conditions of $[d_j]_{x_ix_j}$ in Table 4 can be proved.

Table 4 The symmetric conditions of logical variables

The satisfied conditions of $[d_j]_{x_ix_j}$	Existing symmetry type
$[d_j]_{00} \odot [d_j]_{01} \odot [d_j]_{10} = [1 1 \ldots 1]^{\mathrm{T}}$	E($x_i|x_j$)
$[d_j]_{01} \odot [d_j]_{10} = [1 1 \ldots 1]^{\mathrm{T}}$	N($x_i|x_j$)
$[d_j]_{01} = [1 1 \ldots 1]^{\mathrm{T}}$	S($x_i|x_j$)
$[d_j]_{00} \odot [d_j]_{01} = [1 1 \ldots 1]^{\mathrm{T}}$	S($x_i \mid \bar{x}_j$)
$[d_j]_{10} = [1 1 \ldots 1]^{\mathrm{T}}$	S($x_j|x_i$)
$[d_j]_{00} \odot [d_j]_{10} = [1 1 \ldots 1]^{\mathrm{T}}$	S($x_j \mid \bar{x}_i$)
$[d_j]_{00} \odot [d_j]_{01} \odot [d_j]_{10} = [1 0 \ldots 0]^{\mathrm{T}}$	CE($x_i|x_j$)
$[d_j]_{01} \odot [d_j]_{10} = [1 0 \ldots 0]^{\mathrm{T}}$	CN($x_i|x_j$)
$[d_j]_{01} = [1 0 \ldots 0]^{\mathrm{T}}$	CS($x_i|x_j$)
$[d_j]_{00} \odot [d_j]_{01} = [1 0 \ldots 0]^{\mathrm{T}}$	CS($x_i \mid \bar{x}_j$)
$[d_j]_{10} = [1 0 \ldots 0]^{\mathrm{T}}$	CS($x_j|x_i$)
$[d_j]_{00} \odot [d_j]_{10} = [1 0 \ldots 0]^{\mathrm{T}}$	CS($x_j \mid \bar{x}_i$)

Based on the above discussions, the method for identifying 12 types of symmetric variables based on d_j-map for the n-variable Boolean function is decomposed into the following two steps:

1. Draw the d_j-map of the n-variable Boolean function.

2. Identify the 12 types of symmetric variables by judging whether $[d_j]_{x_ix_j}$ satisfies Deduction 1.

4.2 An example of the algorithm

Example 1 Consider a three-variable Boolean function $f_1(x_1\sim x_3)=\odot\Pi 0, 3, 4$, and apply the d_j-map to identify symmetric variables.

The d_j-map of $f_1(x_1\sim x_3)$ is shown in Fig. 3. From the d_j-map, $[d_j]_{x_1x_2}$ is listed as follows:

$[d_j]_{00}=[0\ 0]^T$, $[d_j]_{01}=[1\ 0]^T$, $[d_j]_{10}=[0\ 1]^T$, $[d_j]_{10}=[1\ 1]^T$. According to Deduction 1, $[d_j]_{01}=[1\ 0]^T$ and then the CS($x_1|x_2$) symmetry exists; $[d_j]_{00}\odot[d_j]_{10}=[1\ 0]^T$, then the CS($x_2\,|\,\bar{x}_1$) symmetry exists; $[d_j]_{00}\odot[d_j]_{01}\odot[d_j]_{10}=[1\ 1]^T$, then the E($x_1|x_2$) symmetry exists. By the same argument, the CN($x_1|x_3$), CS($x_1\,|\,\bar{x}_3$), S($x_3\,|\,\bar{x}_1$), CE($x_2|x_3$), CS($x_2\,|\,\bar{x}_3$), and S($x_3|x_2$) symmetries can be identified.

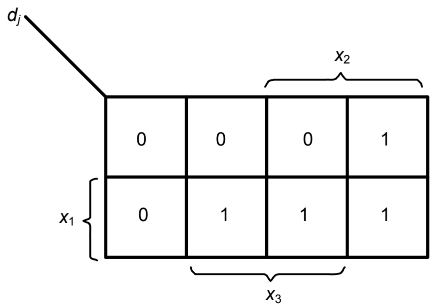

Fig. 3 The d_j-map of f_1

5 Algorithm realization

Based on the discussions on the principle and example of the algorithm, the 12 types of symmetric variables can be identified by judging $[d_j]_{x_ix_j}$. The algorithm is decomposed into the following steps:

Step 1: A 2D array $\boldsymbol{y}$ is defined which is used to conserve the identified function $f_1(x_1\sim x_n)$. Array $\boldsymbol{y}$ has 2^n rows and n+1 columns. The first n columns conserve the variables coding of $f_1(x_1\sim x_n)$. The last column conserves coefficients d_j of $f_1(x_1\sim x_n)$. Taking Example 1 as an example, a 2D array $\boldsymbol{y}_1$, which is used to conserve the identified function $f_1(x_1\sim x_3)$, can be represented as shown in Table 5.

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 $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ ($1\le i<j\le n$), and compare the values of $y[k][i]$, $y[k][j]$ ($1\le k\le 2^n$) to the characteristic coding.

Table 5 Representation of columns of array $\boldsymbol{y}_1$

Column 1	Column 2	Column 3	Column 4
0	0	0	0
0	0	1	0
0	1	1	0
0	1	0	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. If the values of $y[k][i]$ and $y[k][j]$ all 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 $y[k][j]$ all 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. The corresponding elements of arrays $\boldsymbol{C}^0$, $\boldsymbol{C}^1$, and $\boldsymbol{C}^2$ carry out the coincidence operation. If all operation results equal 1, then output E($x_i|x_j$); If the operation results of the first column equal 1, and the others equal 0, then output CE($x_i|x_j$).

2. The corresponding elements of arrays $\boldsymbol{C}^1$ and $\boldsymbol{C}^2$ carry out the coincidence operation. If all operation results equal 1, then output N($x_i|x_j$); If the operation results of the first column equal 1, and the others equal 0, then output CN($x_i|x_j$).

3. The corresponding elements of $\boldsymbol{C}^0$ and $\boldsymbol{C}^1$ carry out the coincidence operation. If all operation results equal 1, then output S($x_i\,|\,\bar{x}_j$); If the operation results of the first column equal 1, and the others equal 0, then output CS($x_i\,|\,\bar{x}_j$).

4. If all elements of array $\boldsymbol{C}^2$ equal 1, then output S($x_j|x_i$); If the first element of array $\boldsymbol{C}^2$ equals 1 and the others equal 0, then output CS($x_j|x_i$).

5. The corresponding elements of $\boldsymbol{C}^0$ and $\boldsymbol{C}^2$ carry out the coincidence operation. If all operation results equal 1, then output S($x_j\,|\,\bar{x}_i$); If the operation results of the first column equal 1, and the others equal 0, then output CS($x_j\,|\,\bar{x}_i$).

6. If all elements of array $\boldsymbol{C}^1$ equal 1, then output S(x_i|x_j); If the first element of array $\boldsymbol{C}^1$ equals 1 and the others equal 0, then output CS(x_i|x_j).

Step 4: Repeat step 2.

Step 5: Termination the loop.

The flow chart of the program implemented in C language is shown in Fig. 4.

6 Experimental results

The proposed algorithm was implemented in C language on the Linux platform and run on an IBM P5 560Q server (eight 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 6 for single-output functions up to 18 variables.The test function 'test 1' is the same as that used in Example 1 in Section 4.2. The results identified by programming the algorithm are consistent with the conclusions identified by judging $[d_j]_{x_ix_j}$ of the function. The validity of the algorithm is also proven. The symmetry detection time for the same number of logical variables in the test function depends on the number of symmetric variables. The more the symmetric variables, the longer the detection time required, as indicated by the test functions 'pde' and 'ig9' in Table 6. The required detection time increases obviously along with the increasing number of logical variables in the test function.

Up to now, there have been no publications on the algorithm which can be programmed for

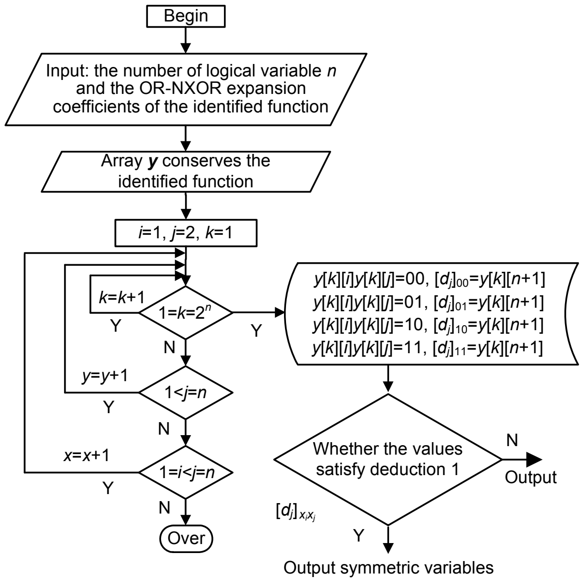

Fig. 4 The flow chart of the program

Table 6 The expermental results for the test function

Name	n	t (s)	Number of existing symmetric variables											
			E(x_i|x_j)	N(x_i|x_j)	S(x_i|x_j)	S(x_i|$\bar{x}_j$)	S(x_j|x_i)	S(x_j|$\bar{x}_i$)	CE(x_i|x_j)	CN(x_i|x_j)	CS(x_i|x_j)	CS(x_i|$\bar{x}_j$)	CS(x_j|x_i)	CS(x_j|$\bar{x}_i$)
test1	3	1.08	1	0	0	0	1	1	1	1	1	2	0	1
c4	4	3.12	1	0	1	1	0	1	0	1	0	1	0	0
ie3	5	6.28	0	1	1	0	0	0	2	0	0	1	1	0
6fe	6	8.21	1	1	1	1	0	0	0	1	0	1	0	0
7ec	7	9.45	0	0	1	0	1	0	1	0	2	0	1	0
dc8	8	11.22	0	1	0	0	1	0	0	1	0	1	0	1
pde	9	15.26	1	0	1	0	0	1	0	0	0	0	0	1
ig9	9	22.83	1	0	1	0	1	0	1	1	1	0	1	1
qtt	10	24.61	1	1	0	1	1	0	1	0	0	1	0	1
exp	11	27.56	0	2	1	1	1	0	0	1	0	1	0	0
ss1	12	30.8	2	0	0	1	1	2	0	1	0	0	1	1
ort	13	37.83	1	0	1	1	1	0	1	0	1	0	1	0
we2	14	45.2	0	0	1	0	1	0	1	0	0	0	1	0
xxt	15	57.21	1	0	0	0	0	1	0	1	0	0	0	0
ujl	16	72	0	0	0	0	1	1	0	0	0	1	1	0
le5	17	102	0	1	0	0	0	1	0	0	0	1	0	0
vcc	18	148	0	0	0	0	0	0	0	0	0	0	0	0

Name: the test function; n: the number of logical variables in the test function; t: the symmetry detection time (s) identifying 12 types of symmetric variables in the COC algebra system for comparison.

7 Comparison of different identification methods

The performance comparisons of different methods for identifying symmetric variables are listed in Table 7. The graphic method and the d_j-map method are restricted by the number of logical variables. Generally, these two methods are suitable for the functions where the number of logical variables is no more than six. The process is complicated for calculating spectrum coefficients and separating the spectral coefficients into $n(n-1)/2$ groups by spectral coefficient methods. The AND-XOR expansion coefficient methods (Falkowski and Kannurao, 1999; Kannurao and Falkowski, 2003) also need to separate the AND-XOR expansion coefficients into $n(n-1)/2$ groups. It is difficult to classify the coefficients by spectral coefficient methods and the AND-XOR expansion coefficient method when the number of logical variables exceeds six. Only the $E(x_i|x_j)$ and $N(x_i|x_j)$ symmetries can be identified by the tabular method. From Table 7, the proposed method is the optimal detection method in terms of the applicability of the number of logical variables, detection type, and the complexity of the identification process. If a Boolean function is expressed as an OR-NXOR expansion, the proposed method which avoids the transformation process from the OR-NXOR expansion to AND-OR-NOT expansion or to the AND-XOR expansion and solves the completeness of the d_j-map method, is obviously the most efficient. If a Boolean function is expressed as an AND-OR-NOT expansion or AND-XOR expansion, by the transformation process from the AND-OR-NOT expansion or AND-XOR expansion to the OR-NXOR expansion, the proposed algorithm is also the most appropriate method.

8 Conclusions

The Boolean logic algebra system, the CRM algebra system, and the COC algebra system are three algebra systems for Boolean functions. The logical expansions have great differences in the complexity of the same function in the three algebra systems. This leads to differences of circuit realization in the circuit area, power consumption, and circuit speed. Detection of the symmetric variables in different types of algebra is very important. In this paper, we propose a new symmetry detection algorithm for identifying 12 types of symmetric variables based on the OR-NXOR expansion. The algorithm has been implemented in C language. Experimental results show that the proposed algorithm is convenient and efficient.

References

Acharya, J., Jafarpour, A., Orlitsky, A., 2011. Expected query complexity of symmetric Boolean functions. Proc. 49th Annual Allerton Conf. on Communication, Control, and Computing, p.26-29. [doi:10.1109/Allerton.2011.6120145]

Table 7 Six methods for comparison

Detection method	Applicability of the number of logical variables	Number of detection types	Foundation work	Time for condition judgment
Graphic method	Usually $n\leq6$	12	Draw the decomposition charts of n-variable Boolean function	$6n(n-1)$
Spectral coefficients method	Usually $n\leq6$	12	Calculate the spectral coefficients, then separate the spectral coefficients into $n(n-1)/2$ groups	$6n(n-1)$
Tabular method	Usually $n\leq10$	2	Draw the 1-valued mintern table	$9n(n-1)/2$
CRM expansion coefficients method	Usually $n\leq6$	12	Separate the AND-XOR expansion coefficients into $n(n-1)/2$ groups	$6n(n-1)$
d_j-map method	Usually $n\leq6$	1	Draw the d_j-map	$n(n-1)/2$
Proposed method	No limit	12	Input the number of logical variable in the test function and the OR-NXOR expansion coefficients	$3n(n-1)$

n: the number of logical variables in the test function

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. [doi:10.1109/FOCS.2012.53]

Cheng, J., Chen, X., Faraj, K.M., *et al.*, 2003. Expansion of logical function in the OR-coincidence system and the transform between it and maxterm expansion. *IEE Proc.-Comput. Dig. Tech.*, **150**(6):397-402. [doi:10.1049/ip-cdt:20030969]

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

Heinrich-Litan, L., Molitor, P., 2000. Least upper bounds for the size of OBDDs using symmetry properties. *IEEE Trans. Comput.*, **49**(4):360-368. [doi:10.1109/12.844348]

Hurst, S.L., 1977. Detection of symmetries in combinatorial functions by spectral means. *IEE J. Electron. Circ. Syst.*, **1**(5):173-180.

Kannurao, S., Falkowski, B.J., 2002. Identification of complement single variable symmetry in Boolean functions through Walsh transform. IEEE Int. Symp. on Circuits and Systems, p.745-748. [doi:10.1109/ISCAS.2002.1010811]

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

Kowshik, H., Kumar, P.R., 2013. Optimal computation of symmetric Boolean functions in collocated networks. *IEEE J. Sel. Area Commun.*, **31**(4):639-654. [doi:10.1109/JSAC.2013.130403]

Lian, Y., Li, X., Chen, X., 2005. Detection of partial symmetric functions based on tabular method. *Bull. Sci. Tech.*, **21**(2):214-217 (in Chinese).

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**(5):553-557. [doi:10.1109/PGEC.1963.263654]

Muller, D.E., 1954. Application of Boolean algebra to switching circuit design and to error detection. *IRE Trans. Electron. Comput.*, **EC-3**(3):6-12. [doi: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*, **57**(10):7205-7220. [doi:10.1109/TIT.2011.2132113]

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. [doi:10.1109/ATS.2003.1250824]

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

Reed, I., 1954. A class of multiple-error-correcting code and the decoding scheme. *IRE Trans Inform. Theory*, **4**(4):38-49. [doi:10.1109/TIT.1954.1057465]

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

Shpilka, A., Tal, A., 2011. On the minimal Fourier degree of symmetric Boolean functions. Proc. IEEE 26th Annual Conf. on Computational Complexity, p.200-209. [doi:10.1109/CCC.2011.16]

Wu, X., Chen, X., Hurst, S.L., 1982. Mapping of Reed-Muller coefficients and the minimisation of exclusive OR-switching functions. *IEE Proc.-Comput. Dig. Tech.*, **129**(1):15-20. [doi:10.1049/ip-e.1982.0004]

Zhao, M., Lou, J., Chen, X., 2006. Denotation and application for d_j-map of symmetric function. *J. Zhejiang Univ. (Sci. Ed.)*, **33**(1):62-65 (in Chinese).