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

Function synthesis algorithm based on RTD-based three-variable universal logic gates*

Mao-qun YAO[†‡1], Kai YANG[1], Ji-zhong SHEN[2], Cong-yuan XU[2]

([1]*Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou 310036, China*)

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

[†]E-mail: yaomaoqun@163.com

Received Nov. 21, 2016; Revision accepted Dec. 20, 2016; Crosschecked Oct. 31, 2017

Abstract: Compared with complementary metal–oxide semiconductor (CMOS), the resonant tunneling device (RTD) has better performances; it is the most promising candidate for next-generation integrated circuit devices. The universal logic gate is an important unit circuit because of its powerful logic function, but there are few function synthesis algorithms that can implement an *n*-variable logical function by RTD-based universal logic gates. In this paper, we propose a new concept, i.e., the truth value matrix. With it a novel disjunctive decomposition algorithm can be used to decompose an arbitrary *n*-variable logical function into three-variable subset functions. On this basis, a novel function synthesis algorithm is proposed, which can implement arbitrary *n*-variable logical functions by RTD-based universal threshold logic gates (UTLGs), RTD-based three-variable XOR gates (XOR3s), and RTD-based three-variable universal logic gate (ULG3s). When this proposed function synthesis algorithm is used to implement an *n*-variable logical function, if the function is a directly disjunctive decomposition one, the circuit structure will be very simple, and if the function is a non-directly disjunctive decomposition one, the circuit structure will be simpler than when using only UTLGs or ULG3s. The proposed function synthesis algorithm is straightforward to program, and with this algorithm it is convenient to implement an arbitrary *n*-variable logical function by RTD-based universal logic gates.

Key words: Resonant tunneling device (RTD); Disjunctive decomposition algorithm; Universal logic gate; Truth value matrix; Function synthesis algorithm
https://doi.org/10.1631/FITEE.1601730 **CLC number:** TN47

1 Introduction

With the improvement of integrated circuit, the complementary metal–oxide semiconductor (CMOS) technology is gradually approaching its physical limitations. Compared with CMOS, the resonant tunneling device (RTD) has better performances, including negative differential resistance, high-speed switching, self-latching, and low power consumption (Mazumder *et al.*, 1998; Likharev, 2008; Iwai, 2013), so it is more likely to be the main electronic device of next-generation integrated circuits (Muramatsu *et al.*, 2005; Zheng and Huang, 2009). The universal logic gate, due to its powerful logic function, has become an important unit circuit for implementing the *n*-variable logical function, and RTD is more suitable for implementing the universal logic gate because of its negative differential resistance (Beiu *et al.*, 2003; Zheng and Huang, 2009; Mirhoseini *et al.*, 2010).

Though using the universal logic gate to implement an *n*-variable logical function can simplify the circuit, using different universal logic gates requires different algorithms. Some function decomposition algorithms have been proposed in the literature (Files and Perkowski, 2000; Ngwira and Tshabalala, 2002; Czajkowski and Brown, 2008; Liu *et al.*, 2011; Altun and Riedel, 2012; Nikodem, 2013; Fan *et al.*, 2014), but these algorithms cannot be

‡ Corresponding author
* Project supported by the Zhejiang Provincial Natural Science Foundation, China (No. LY15F010011) and the National Natural Science Foundation of China (Nos. 61771179, 61471314, and 61271124)
ORCID: Mao-qun YAO, http://orcid.org/0000-0001-6484-4972

applied to the RTD-based universal threshold logic gate (UTLG) (Wei and Shen, 2011) or RTD-based three-variable universal logic gate (ULG3) (Yao *et al.*, 2015) to implement the n-variable logical function. By using the disjunctive decomposition algorithm (Kolodzinski and Hrynkiewicz, 2009; Hrynkiewicz and Kolodzinski, 2010; Falkowski and Kannurao, 2001) to decompose the n-variable logical function, and combining the features of the RTD-based three-variable universal logic gate, an algorithm that can implement an arbitrary n-variable logical function by RTD-based three-variable universal threshold logic gates can be developed. Such an algorithm should have the following features: (1) It must be able to decompose an arbitrary n-variable logical function into three-variable subset functions which can be implemented by RTD-based three-variable universal logic gates; (2) The decomposition result must be relatively simple to simplify the circuit.

In this study, a new concept 'truth value matrix' is proposed. With it a novel disjunctive decomposition algorithm is further proposed, which can decompose an arbitrary n-variable logical function into three-variable subset functions. Then, a novel function synthesis algorithm is proposed, which can implement the arbitrary n-variable logical function by UTLGs, ULG3s, and RTD-based three-variable XOR gates (XOR3) (Yao *et al.*, 2015). The proposed function synthesis algorithm provides a new scheme for the design of integrated circuits by RTD devices.

2 Background

2.1 Disjunctive decomposition algorithm

Definition 1 (Disjunctive decomposition (Bertacco and Damiani, 1997)) If an n-variable logical function $f(x_1, x_2, \ldots, x_n)$ can be decomposed into two functions, one is its subfunction g_1, and the other is $F(g_1, x_l, \ldots, x_n)$ which consists of the remaining variables besides g_1, it is called a 'disjunctive decomposition function'. The diagram of a disjunctive decomposition function is shown in Fig. 1, where x_1, x_2, …, x_{l-1} are bound variables and x_l, …, x_n are free variables. We have $f(x_1, x_2, \ldots, x_n)=F(g_1, x_l, \ldots, x_n)$.

Definition 2 (Subset function) If an n-variable logic function $f(x_1, x_2, \ldots, x_n)$ is decomposed into some subfunctions by the disjunctive decomposition algorithm, the sub-functions are called 'subset functions'. For example, in a five-variable function $f(x_1, x_2, x_3, x_4, x_5)$, x_1, x_2, and x_3 are bound variables and x_4, x_5 are free variables. After decomposition, $f(x_1, x_2, x_3, x_4, x_5)= F(g_1, x_4, x_5)$. $F(g_1, x_4, x_5)$ and $g_1(x_1, x_2, x_3)$ are called 'three-variable subset functions'.

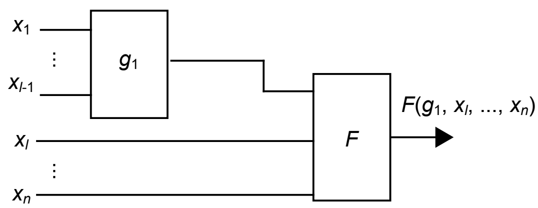

Fig. 1 Disjunctive decomposition function

2.2 Threshold logic

A threshold logic gate is defined as a logic gate with a single binary output and n binary input variables, $\{x_i\}$ (i=1, 2, …, n). Its internal parameters are a set of n positive or negative integer weights, $\{w_i\}$ (i=1, 2, …, n), and a threshold T, such that its input-output relationship can be expressed as

$$y=\begin{cases}1, & \text{if } \sum_{i=1}^{n} w_i x_i \geq T,\\ 0, & \text{otherwise.}\end{cases} \tag{1}$$

The threshold logic gate can also be presented as $f=\langle w_1x_1+ w_2x_2+\ldots+w_nx_n\rangle_T$.

If a logical function can be implemented with a single threshold logic gate, the function is called a 'threshold function'; otherwise, it is called a 'non-threshold function' (Zhang *et al.*, 2005).

3 Algorithm of n-variable logical function disjunctive decomposition

In this section, the truth value matrix is introduced, as well as an algorithm of n-variable logical function disjunctive decomposition with the truth value matrix.

Disjunctive decomposition algorithms in the Reed-Muller spectral domain or the Walsh spectrum have been studied (Bertacco and Damiani, 1997; Falkowski and Kannurao, 2001; Kolodzinski and

Hrynkiewicz, 2009; Hrynkiewicz and Kolodzinski, 2010), but they are not suitable for implementing an n-variable logical function by RTD-based universal logic gates. We propose a new algorithm in Boolean algebra, which can implement an n-variable logical function by RTD-based universal logic gates. This algorithm should meet two requirements: (1) The function should be easily expressed with an increasing number of variables; (2) The algorithm should be suitable for all kinds of functions.

3.1 Truth value matrix

To satisfy the first request, a new matrix named the 'truth value matrix (**TM**)' is defined to express functions. According to the usual expression of matrices, the truth value matrix of an n-variable logic function is defined as a matrix with $[n/2]$ rows and $[(n+1)/2]$ columns ([ ] represents integer notation). The elements of the matrix are assigned the values in a truth table first from left to right and then from top to bottom. The row coordinate indicates the sequence code of the first $[n/2]$ variables, and the column coordinate indicates the sequence code of the last $[(n+1)/2]$ variables; e.g., for the three-variable function $f(x_1, x_2, x_3)=\bar{x}_1\bar{x}_2\bar{x}_3+x_1\bar{x}_2x_3+\bar{x}_1x_2\bar{x}_3+x_1x_2x_3$, its truth value matrix is $\mathbf{TM}=\begin{bmatrix}1&0&1&0\\0&1&0&1\end{bmatrix}$, the row coordinate indicates x_1: 0, 1, and the column coordinate indicates x_2x_3: 00, 01, 10, 11.

3.2 Improved disjunctive decomposition algorithm

To satisfy the second request, XOR decomposition is introduced and, according to the algorithm proposed by Bertacco and Damiani (1997), we also divide the n-variable function into two categories: one is the directly disjunctive decomposition function, and its corresponding truth value matrix has only one or two different columns or rows; the other is the non-directly disjunctive decomposition function, and its corresponding truth value matrix has three or more different columns or rows.

Note that if the variables of a function are in a different order, the corresponding truth value matrix of the function will be different; thus, as a three-variable function $f(x_1, x_2, x_3)=x_1x_2x_3$, it can be expressed as $f(x_1, x_2, x_3)=x_3x_2x_1$. The corresponding truth value matrices of these two expressions are different, so the same function may belong to a different type of function because the variables of the function are in a different order. Hrynkiewicz and Kolodzinski (2010) proposed an algorithm that can be used to choose the optimal variable order of a function.

Before we discuss the decomposition algorithm of these two kinds of function, some concepts of the algorithm should be defined:

Definition 3 (Basic and non-basic columns or rows) The columns or rows of the truth value matrix which are used to compare with other columns or rows, if the function is a directly disjunctive decomposition one.

Definition 4 (Reference row r_i and non-reference row $\bar{r}_i$) Reference row r_i is the row of the truth value matrix used to compare with other rows, if the function is a non-directly disjunctive decomposition one. All the values of $\bar{r}_i$ elements are the NOT of the values of the corresponding r_i elements, e.g., r_i=[0 1] and $\bar{r}_i$=[1 0].

Definition 5 (XOR of two matrices) The XOR of two matrices is a matrix with its elements being the XOR of the corresponding elements of the two matrices. For example, when $\boldsymbol{A}=\begin{bmatrix}0&0\\1&1\end{bmatrix}$, $\boldsymbol{B}=\begin{bmatrix}0&1\\0&1\end{bmatrix}$, the XOR of $\boldsymbol{A}$ and $\boldsymbol{B}$ is $\boldsymbol{A}\oplus\boldsymbol{B}=\begin{bmatrix}0&1\\1&0\end{bmatrix}$.

1. For the directly disjunctive decomposition function whose corresponding truth value matrix has only one or two different columns (Bertacco and Damiani, 1997), we propose the following decomposition algorithm:

(1) Write the truth value matrix **TM** of $f(x_1, x_2, \ldots, x_n)$.

(2) Select the basic and non-basic columns: If the columns of **TM** are all the same, select any one of them as the basic column, and construct the column whose element values are all 0's as the non-basic column. If **TM** has two different columns, select the column which repeats the most times in **TM** as the basic column, and the other as the non-basic column; if two different columns of **TM** repeat the same number of times, select the column that has more 1's as the basic column, and the other as the non-basic column. If the number of 1's is the same in the two different columns, select either of them as the basic

column, and the other as the non-basic column.

(3) Construct transition matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$: Decompose $f(x_1, x_2, \ldots, x_n)=f(x_1, x_2, \ldots, x_m, g)$, where g is an $[(n+1)/2]$-variable function $g(x_{m+1}, x_{m+2}, \ldots, x_n)$. $\boldsymbol{M}_1$ is the matrix of $f(x_1, x_2, \ldots, x_m, g)$ with $[n/2]$ rows and two columns. Its row coordinate is the same as **TM**, and its column coordinate is g: 0, 1. Fill in the first column of $\boldsymbol{M}_1$ with the basic column, and the second column of $\boldsymbol{M}_1$ with the non-basic column. $\boldsymbol{M}_2$ is the matrix of $g(x_{m+1}, x_{m+2}, \ldots, x_n)$ with one row and $[(n+1)/2]$ columns. Its column coordinate is the same as **TM**, and its row coordinate expresses that each column is the basic or non-basic one. According to **TM**, to fill in $\boldsymbol{M}_2$, if the column is the basic column in **TM**, fill in the corresponding column of $\boldsymbol{M}_2$ with 0, and if the column is the non-basic column in **TM**, fill in the corresponding column of $\boldsymbol{M}_2$ with 1.

(4) Construct truth value matrices $\mathbf{TM}_1$ and $\mathbf{TM}_2$: $\mathbf{TM}_1$ is the matrix with $[l/2]$ rows and $[(l+1)/2]$ columns ($l=[n/2]+1$), and $\mathbf{TM}_2$ is the matrix with $[h/2]$ rows and $[(h+1)/2]$ columns ($h=[(n+1)/2]$). Fill in $\mathbf{TM}_1$ and $\mathbf{TM}_2$ with $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ from left to right and then from top to bottom.

(5) The function can be decomposed into $f(x_1, x_2, \ldots, x_m, g)$, $f(x_1, x_2, \ldots, x_n)=f(x_1, x_2, \ldots, x_m, g)$, where g is $g(x_{m+1}, x_{m+2}, \ldots, x_n)$. $f(x_1, x_2, \ldots, x_m, g)$ and $g(x_{m+1}, x_{m+2}, \ldots, x_n)$ are the corresponding functions of $\mathbf{TM}_1$ and $\mathbf{TM}_2$.

If the truth value matrix of the directly disjunctive decomposition function has only one or two different rows, using the proposed directly disjunctive decomposition algorithm we need only to replace column with row during the decomposition process.

Example 1 Decompose the four-variable function

$$f(x_1,x_2,x_3,x_4)=\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4+\overline{x}_1\overline{x}_2\overline{x}_3x_4+\overline{x}_1\overline{x}_2x_3\overline{x}_4+\overline{x}_1\overline{x}_2x_3x_4+x_1x_2\overline{x}_3\overline{x}_4+x_1x_2\overline{x}_3x_4.$$

The truth value matrix of the function is

$$\mathbf{TM}=\begin{bmatrix}1&1&1&1\\0&0&0&0\\0&0&0&0\\1&1&0&0\end{bmatrix}.$$

TM has only two kinds of different columns, so the function is a directly disjunctive decomposition one:

$$\boldsymbol{A}=\begin{bmatrix}1\\0\\0\\1\end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}.$$

$\boldsymbol{A}$ and $\boldsymbol{B}$ repeat the same number of times, and $\boldsymbol{A}$ has more 1's than $\boldsymbol{B}$. $\boldsymbol{A}$ is selected as the basic column, and $\boldsymbol{B}$ the non-basic column. Then, M_1 and M_2 are

$$\boldsymbol{M}_1=\begin{bmatrix}1&1\\0&0\\0&0\\1&0\end{bmatrix},\ \boldsymbol{M}_2=[0\ \ 0\ \ 1\ \ 1].$$

The row coordinate of $\boldsymbol{M}_1$ is x_1x_2: 00, 01, 10, 11; the column coordinate of $\boldsymbol{M}_1$ is g: 0, 1. The column coordinate of $\boldsymbol{M}_2$ is x_3x_4: 00, 01, 10, 11.

The truth value matrix $\mathbf{TM}_1$ of $f(x_1, x_2, g)$ is

$$\mathbf{TM}_1=\begin{bmatrix}1&1&0&0\\0&0&1&0\end{bmatrix}.$$

The row coordinate of $\mathbf{TM}_1$ is x_1: 0, 1; the column coordinate of $\mathbf{TM}_1$ is x_2g: 00, 01, 10, 11. The truth value matrix $\mathbf{TM}_2$ of $g(x_3, x_4)$ is

$$\mathbf{TM}_2=\begin{bmatrix}0&0\\1&1\end{bmatrix}.$$

The row coordinate of $\mathbf{TM}_2$ is x_3: 0, 1; the column coordinate of $\mathbf{TM}_2$ is x_4: 0, 1. Then, $f(x_1, x_2, x_3, x_4)$ in Example 1 can be decomposed as $f(g,x_1,x_2)=\overline{x}_1\overline{x}_2\overline{g}+x_1x_2\overline{g}+\overline{x}_1\overline{x}_2g,\ \ g(x_3,x_4)=x_3\overline{x}_4+x_3x_4$.

2. For the non-directly disjunctive decomposition function whose corresponding truth value matrix has three or more different columns or rows (Bertacco and Damiani, 1997), we propose the following decomposition algorithm:

(1) Write the truth value matrix **TM** of $f(x_1, x_2, \ldots, x_n)$.

(2) Select reference row r_1 and non-reference row $\overline{r}_1$: If **TM** has the same rows, select the row that repeats the most number of times as reference row r_1; if **TM** has two different rows that repeat the same

number of times, select the row that has more 1's as reference row r_1, and if the number of 1's is the same in the two different rows, select any one of them as reference row r_1; if all the rows in **TM** are different, select the row that has the most 1's as reference row r_1, and if there are some rows with the same number of 1's, select any one of them as reference row r_1. Write the corresponding non-reference row $\overline{r_1}$ of r_1.

(3) First XOR decomposition: Construct a new truth value matrix $\mathbf{TM}_1$ that has the same structure as **TM**. Comparing all the rows of **TM** with r_1 and $\overline{r_1}$, $\mathbf{TM}_1$ is constructed as follows: (i) If a row of **TM** has more elements identical to their counterparts in r_1, fill in the corresponding row of $\mathbf{TM}_1$ with r_1; else, fill in the corresponding row of $\mathbf{TM}_1$ with $\overline{r_1}$. (ii) If a row of **TM** has an equal number of elements identical to those in r_1 and $\overline{r_1}$, fill in the corresponding row of $\mathbf{TM}_1$ with the row that has more 1's in r_1 or $\overline{r_1}$. (iii) If the number of 1's is the same in r_1 and $\overline{r_1}$, the corresponding row of $\mathbf{TM}_1$ can be filled in with either r_1 or $\overline{r_1}$. Calculate the new truth value matrix $\mathbf{TM}_{11}$ by $\mathbf{TM}_{11}=\mathbf{TM}_1\oplus\mathbf{TM}$.

(4) Step 1 of the second XOR decomposition: If $\mathbf{TM}_{11}$ has the same number of rows except the row whose element values are all 0's, select this row as reference row r_2, and select the row whose element values are all 0's as the other reference row r_0; if the rows of $\mathbf{TM}_{11}$ are all different except the row whose element values are all 0's, select the row that has the most 1's as reference row r_2; if there are two or more different rows that have the same number of 1's, select any of them as reference row r_2, and the other reference row is still r_0.

(5) Step 2 of the second XOR decomposition: Construct a new truth value matrix $\mathbf{TM}_2$ which has the same structure as $\mathbf{TM}_{11}$, by comparing all the rows of $\mathbf{TM}_{11}$ with r_2 and r_0. If a row of $\mathbf{TM}_{11}$ has more elements identical to their counterparts in r_2, fill in the corresponding row of $\mathbf{TM}_2$ with r_2; else, fill in the corresponding row of $\mathbf{TM}_2$ with r_0. If a row of $\mathbf{TM}_{11}$ has an equal number of elements that are identical to their counterparts in r_2 and r_0, fill in the corresponding row of $\mathbf{TM}_2$ with r_2. Calculate the new truth value matrix $\mathbf{TM}_{22}$ by $\mathbf{TM}_{22}=\mathbf{TM}_2\oplus\mathbf{TM}_{11}$.

(6) Repeat steps (4) and (5) until the calculated truth value matrix has only two different rows.

(7) The function can be decomposed into the XOR of multiple directly disjunctive decomposition functions, i.e., $f(x_1, x_2, \ldots, x_n)=f_1(x_1, x_2, \ldots, x_n)\oplus f_2(x_1, x_2, \ldots, x_n)\oplus\ldots\oplus f_i(x_1, x_2, \ldots, x_n)$. $\mathbf{TM}_1, \mathbf{TM}_2, \ldots, \mathbf{TM}_i$ are the corresponding truth value matrices of $f_1, f_2, \ldots, f_i$.

Example 2 Decompose the four-variable function

$$f(x_1,x_2,x_3,x_4)=\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4+\overline{x}_1\overline{x}_2\overline{x}_3x_4+\overline{x}_1\overline{x}_2x_3x_4 +\overline{x}_1x_2\overline{x}_3x_4+\overline{x}_1x_2x_3x_4+x_1\overline{x}_2\overline{x}_3x_4+x_1\overline{x}_2x_3x_4+x_1x_2\overline{x}_3\overline{x}_4.$$

The truth value matrix of $f(x_1, x_2, x_3, x_4)$ is

$$\mathbf{TM}=\begin{bmatrix}1&1&0&1\\0&1&0&1\\0&1&0&1\\1&0&0&0\end{bmatrix}.$$

So, the function is a non-directly disjunctive decomposition one. Select row r_1=[0 1 0 1] as the reference row, which is repeated twice in **TM**, and non-reference row $\overline{r_1}$=[1 0 1 0]. Constructing the new truth value matrix $\mathbf{TM}_1$, and comparing all the rows of **TM** with r_1 and $\overline{r_1}$, it is noted that the first row has three elements identical to their counterparts in r_1, and the fourth row has three elements identical to their counterparts in $\overline{r_1}$.Thus, $\mathbf{TM}_1$ is filled in with r_1 and $\overline{r_1}$ as

$$\mathbf{TM}_1=\begin{bmatrix}0&1&0&1\\0&1&0&1\\0&1&0&1\\1&0&1&0\end{bmatrix}.$$

Calculate $\mathbf{TM}_{11}$ by $\mathbf{TM}_{11}=\mathbf{TM}_1\oplus\mathbf{TM}$:

$$\mathbf{TM}_{11}=\begin{bmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\end{bmatrix}.$$

For the second decomposition, selecting reference row $\boldsymbol{r}_2$=[1 0 0 0] and the other reference row $\boldsymbol{r}_0$=[0 0 0 0], the truth value matrix $\mathbf{TM}_2$ is

$$\mathbf{TM}_2=\begin{bmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}.$$

Calculate $\mathbf{TM}_{22}$ by $\mathbf{TM}_{22}=\mathbf{TM}_2\oplus\mathbf{TM}_{11}$ as

$$\mathbf{TM}_{22}=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\end{bmatrix}.$$

Thus, we have $\mathbf{TM}_3$=$\mathbf{TM}_{22}$. Then $f(x_1,x_2,x_3,x_4)=f_1(x_1,x_2,x_3,x_4)\oplus f_2(x_1,x_2,...,x_n)\oplus f_3(x_1,x_2,x_3,x_4)$, and $\mathbf{TM}_1$, $\mathbf{TM}_2$, $\mathbf{TM}_3$ are the corresponding truth value matrices of f_1, f_2, f_3, with $f_1=x_1x_2\overline{x}_4+\overline{x}_1x_4+\overline{x}_2x_4$, $f_2=\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$, $f_3=x_1x_2x_3\overline{x}_4$.

3.3 Algorithm of *n*-variable logical function disjunctive decomposition

Because ULG3 has a powerful logical function and its circuit is not very complex, it is an ideal universal logic gate (Yao *et al.*, 2015). We now propose the function synthesis algorithm that can be used to implement an arbitrary n-variable function by the RTD-based universal threshold logic gates (UTLGs) (Wei and Shen, 2011) and ULG3 (Yao *et al.*, 2015). According to the proposed improved disjunctive decomposition algorithm, we propose the algorithm that can be used to decompose the n-variable function into three-variable subset functions, and the decomposition process is as follows:

1. Write the truth value matrix **TM** of $f(x_1, x_2, \ldots, x_n)$.

2. Judge whether the function is a directly disjunctive decomposition one.

3. If the function is a directly disjunctive decomposition one, use the improved disjunctive decomposition algorithm to decompose it, and we have $f(x_1, x_2, \ldots, x_n)$=$f(x_1, x_2, \ldots, x_m, g)$, where g is $g(x_{m+1}, x_{m+2}, \ldots, x_n)$.

4. If the function is a non-directly disjunctive decomposition one, use the improved disjunctive decomposition algorithm to decompose it, and we have $f(x_1, x_2, \ldots, x_n)$=$f_1(x_1, x_2, \ldots, x_n)\oplus f_2(x_1, x_2, \ldots, x_n)\oplus\ldots\oplus f_i(x_1, x_2, \ldots, x_n)$.

5. If executing step 3, reassess whether $f(x_1, x_2, \ldots, x_m, g)$ and $g(x_{m+1}, x_{m+2}, \ldots, x_n)$ are directly disjunctive decomposition functions; if executing step 4, rejudge whether $f_1, f_2, \ldots, f_i$ are directly disjunctive decomposition functions.

6. Repeat steps 3–5 until all the subset functions $f(x_1, x_2, \ldots, x_m, g)$, $g(x_{m+1}, x_{m+2}, \ldots, x_n)$ and $f_1, f_2, \ldots, f_i$ are decomposed into three-variable subset functions.

4 Circuit design of an *n*-variable function

In this section, we propose a function synthesis algorithm that can be used to implement an arbitrary n-variable logical function by UTLGs, XOR3s, and ULG3s.

4.1 UTLG, ULG3, and XOR3

Wei and Shen (2011) proposed an RTD-based universal threshold logic gate (UTLG), which can implement all the three-variable threshold functions with one UTLG. Fig. 2 shows the schematic and symbol of UTLG, where only input bits need to be reconfigured and there is no need to change the parameters of RTD or MOSFET while implementing a threshold function with UTLG.

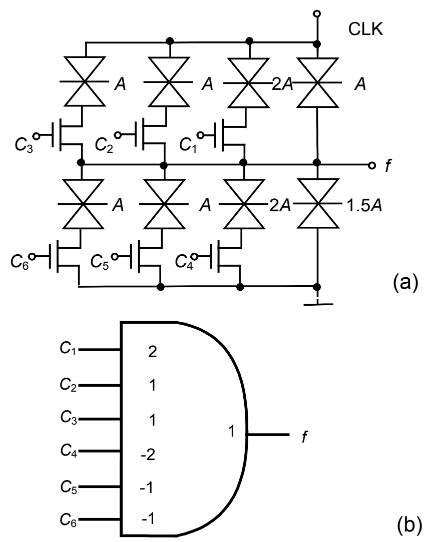

Fig. 2 Universal threshold logic gate (UTLG): (a) schematic; (b) symbol

Yao *et al*. (2015) proposed an RTD-based three-variable universal logic gate (ULG3), which can implement all the three-variable functions with one ULG3. The circuit in which all the three-variable non-threshold functions are implemented by a ULG3 is relatively simple, but if the function is a three-variable threshold one, it is better to choose UTLG. Fig. 3 shows the schematic and symbol of ULG3.

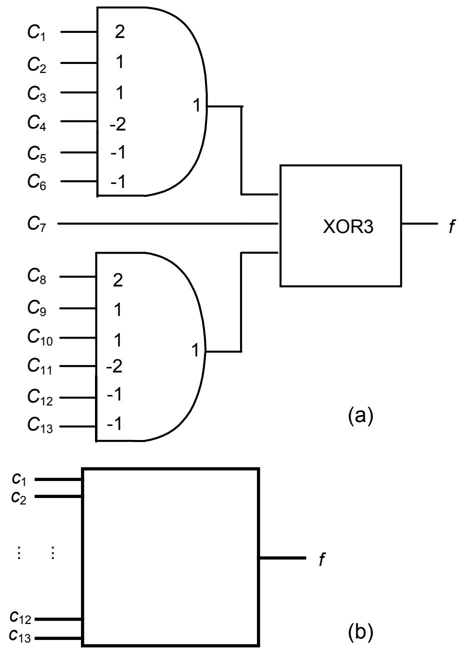

Fig. 3 Three-variable universal logic gate (ULG3): (a) schematic; (b) symbol

Yao *et al*. (2015) also proposed an RTD-based three-variable XOR gate (XOR3). All the three-variable XOR functions can be implemented with XOR3. Fig. 4 shows the schematic and symbol of XOR3.

4.2 Function synthesis algorithm for an *n*-variable logical function

All the *n*-variable functions can be implemented by ULG3, but after decomposition its sub-functions may have some threshold functions, and using the UTLG to implement them will be simpler. So, we propose a function synthesis algorithm of the *n*-variable logical function by ULG3s, UTLGs, and XOR3s.

The function synthesis algorithm that can implement an arbitrary *n*-variable logical function by UTLGs, XOR3s, and ULG3s is as follows:

1. Use the proposed algorithm of *n*-variable logical function disjunctive decomposition to decompose the *n*-variable function $f(x_1, x_2, \ldots, x_n)$ into three-variable subset functions.

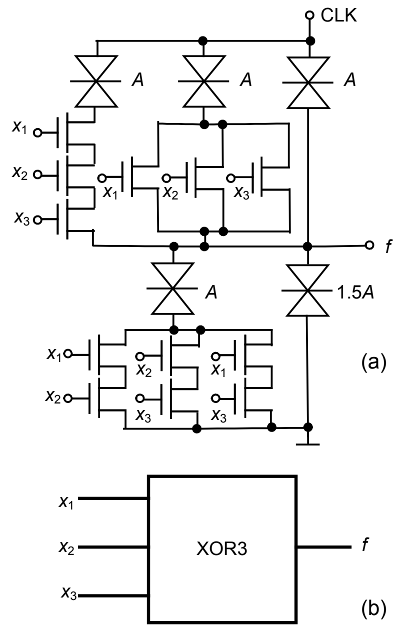

Fig. 4 Three-variable XOR gate (XOR3): (a) schematic; (b) symbol

2. Judge all the three-variable subset functions. If the three-variable subset function is the threshold function, it will be implemented by UTLG; if the three-variable subset function is a non-threshold function except the two special functions $f=x_1\oplus x_2\oplus x_3$ and $f=\overline{x_1\oplus x_2\oplus x_3}$, it will be implemented by ULG3; if the three-variable subset function is the XOR function, it will be implemented by XOR3.

Example 3 Implement the five-variable function

$$f(x_1, x_2, x_3, x_4, x_5)=\sum_i m_i\ , \quad i=0, 2, 3, 5, 6, 9, 12, 15, 24, 25, 26, 27, 28, 29, 30, 31,$$

where m_i is the minimum item of the function.

The truth value matrix of $f(x_1, x_2, x_3, x_4, x_5)$ is

$$\mathbf{TM}=\begin{bmatrix}1&0&1&1&0&1&1&0\\0&1&0&0&1&0&0&1\\0&0&0&0&0&0&0&0\\1&1&1&1&1&1&1&1\end{bmatrix}.$$

$f(x_1, x_2, x_3, x_4, x_5)$ has only two different columns and it is a directly disjunctive decomposition function. Select these two columns as basic column $\boldsymbol{A}$ and non-basic column $\boldsymbol{B}$:

$$\boldsymbol{A}=\begin{bmatrix}1\\0\\0\\1\end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix}0\\1\\0\\1\end{bmatrix}.$$

$\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are

$$\boldsymbol{M}_1=\begin{bmatrix}1 & 0\\0 & 1\\0 & 0\\1 & 1\end{bmatrix},\ \boldsymbol{M}_2=[0\ 1\ 0\ 0\ 1\ 0\ 0\ 1].$$

The row coordinate of $\boldsymbol{M}_1$ is x_1x_2: 00, 01, 10, 11; the column coordinate of $\boldsymbol{M}_1$ is g: 0, 1. The column coordinate of $\boldsymbol{M}_2$ is $x_3x_4x_5$: 000, 001, 010, 011, 100, 101, 110, 111. The truth value matrix $\mathbf{TM}_1$ of $f(x_1, x_2, g)$ is

$$\mathbf{TM}_1=\begin{bmatrix}1 & 0 & 0 & 1\\0 & 0 & 1 & 1\end{bmatrix}.$$

The row coordinate of $\mathbf{TM}_1$ is x_1: 0, 1; the column coordinate of $\mathbf{TM}_1$ is x_2g: 00, 01, 10, 11. The truth value matrix $\mathbf{TM}_2$ of $g(x_3, x_4, x_5)$ is

$$\mathbf{TM}_2=\begin{bmatrix}0 & 1 & 0 & 0\\1 & 0 & 0 & 1\end{bmatrix}.$$

The row coordinate of $\mathbf{TM}_2$ is x_3: 0, 1; the column coordinate of $\mathbf{TM}_2$ is x_4x_5: 00, 01, 10, 11.

Then, $f(x_1, x_2, x_3, x_4, x_5)$ can be decomposed into

$$f(x_1,x_2,g)=\overline{x}_1\overline{x}_2\overline{g}+x_1x_2\overline{g}+\overline{x}_1x_2g+x_1x_2g,$$
$$g(x_3,x_4,x_5)=\overline{x}_3\overline{x}_4x_5+x_3\overline{x}_4\overline{x}_5+x_3x_4x_5.$$

$f(x_1, x_2, g)$ and $g(x_3, x_4, x_5)$ are three-variable non-threshold functions. They can be implemented by ULG3s. To determine the inputs of ULG3s, we can decompose $f(x_1, x_2, g)$ into the XOR of two threshold functions (Yao *et al.*, 2015), i.e., $f(x_1, x_2, g)=f_1\oplus f_2$, $f_1=\langle g-x_1-2x_2\rangle_0$, $f_2=\langle 2g+x_1+x_2\rangle_2$, and decompose $g(x_3, x_4, x_5)$ into the XOR of two threshold functions $g(x_3, x_4, x_5)=f_3\oplus f_4$, $f_3=\langle 2x_3-x_4+x_5\rangle_2$, $f_4=\langle -x_4+x_5\rangle_1$. Then, $f(x_1, x_2, x_3, x_4, x_5)$ can be implemented by two ULG3s. Fig. 5 shows the ULG3 implementation of this function.

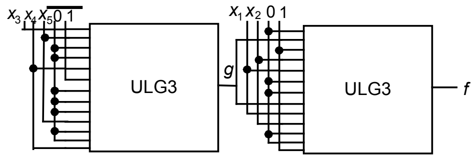

Fig. 5 The circuit for Example 3

From this example, we can find that if the function is a directly disjunctive decomposition one, the circuit of the function will be very simple by applying the proposed algorithm of n-variable logical function disjunctive decomposition.

Example 4 Implement the five-variable function

$$f(x_1, x_2, x_3, x_4, x_5)=\sum_i m_i,\ i=7, 11, 13, 14, 15, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31,$$

where m_i is the minimum item of the function.

The truth value matrix of $f(x_1, x_2, x_3, x_4, x_5)$ is

$$\mathbf{TM}=\begin{bmatrix}0&0&0&0&0&0&0&1\\0&0&0&1&0&1&1&1\\0&0&0&1&0&1&1&1\\0&1&1&1&1&1&1&1\end{bmatrix}.$$

$f(x_1, x_2, x_3, x_4, x_5)$ is a non-directly disjunctive decomposition function, and we use the proposed algorithm of n-variable logical function disjunctive decomposition to decompose the function into three-variable subset functions. So, the truth value matrices of the subset functions are

$$\mathbf{TM}_1=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&1&0&1&1&1\\0&0&0&1&0&1&1&1\\1&1&1&1&1&1&1&1\end{bmatrix},$$

$$\mathbf{TM}_2=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0\end{bmatrix},$$

$$\mathbf{TM}_3=\begin{bmatrix}0&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}.$$

Then, $f(x_1, x_2, x_3, x_4, x_5)$ is decomposed into

$$f(x_1,x_2,x_3,x_4,x_5)=f_1(x_1,x_2,x_3,x_4,x_5)$$
$$\oplus f_2(x_1,x_2,x_3,x_4,x_5)\oplus f_3(x_1,x_2,x_3,x_4,x_5).$$

Continue to decompose functions f_1, f_2, f_3. We have $f_1=\bar{x}_1x_2\bar{g}_1+x_1\bar{x}_2\bar{g}_1+x_1x_2\bar{g}_1+x_1x_2g_1$, $g_1=\bar{x}_3\bar{x}_4\bar{x}_5+\bar{x}_3\bar{x}_4x_5+\bar{x}_3x_4\bar{x}_5+x_3\bar{x}_4\bar{x}_5$, $f_2=x_1x_2g_2$, $g_2=\bar{x}_3\bar{x}_4\bar{x}_5$, $f_3=\bar{x}_1\bar{x}_2g_3$, $g_3=x_3x_4x_5$.

All the three-variable subset functions are threshold ones: $f_1=\langle x_1+x_2-g_1\rangle_1$, $g_1=\langle -x_3-x_4-x_5\rangle_{-1}$, $f_2=\langle x_1+x_2+g_2\rangle_3$, $g_2=\langle -x_3-x_4-x_5\rangle_0$, $f_3=\langle -x_1-x_2+g_3\rangle_1$, $g_3=\langle x_3+x_4+x_5\rangle_3$. Then, $f(x_1, x_2, x_3, x_4, x_5)$ can be implemented by six UTLGs and one XOR3. Fig. 6a shows the UTLG and XOR3 implementations of this function. The function can also be implemented by eight UTLGs (Wei and Shen, 2011). Fig. 6b shows the UTLGs implementation of this function.

From this example we can see that if the function is a non-directly disjunctive decomposition one, the circuit of this function will be simpler when the function is implemented by UTLGs, ULG3s, and XOR3 than when it is implemented by only UTLGs or ULG3s.

Example 5 Implement the six-variable function

$$f(x_1,x_2,x_3,x_4,x_5,x_6)=\sum_i m_i,$$
$$i=0,2,3,5,6,9,10,11,12,14,15,16,17,20,22,$$
$$25,26,27,29,31,32,34,35,37,38,40,41,43,44,$$
$$45,47,48,50,51,53,54,57,60,61,$$

where m_i is the minimum item of the function.

The truth value matrix of $f(x_1, x_2, x_3, x_4, x_5 x_6)$ is

$$\mathbf{TM}=\begin{bmatrix}1&0&1&1&0&1&1&0\\0&1&1&1&1&0&1&1\\1&1&0&0&1&0&1&0\\0&1&1&1&0&1&0&1\\1&0&1&1&0&1&1&0\\1&1&0&1&1&1&0&1\\1&0&1&1&0&1&1&0\\0&1&0&0&1&1&0&0\end{bmatrix}.$$

$f(x_1, x_2, x_3, x_4, x_5 x_6)$ is a non-directly disjunctive decomposition function. Using the proposed

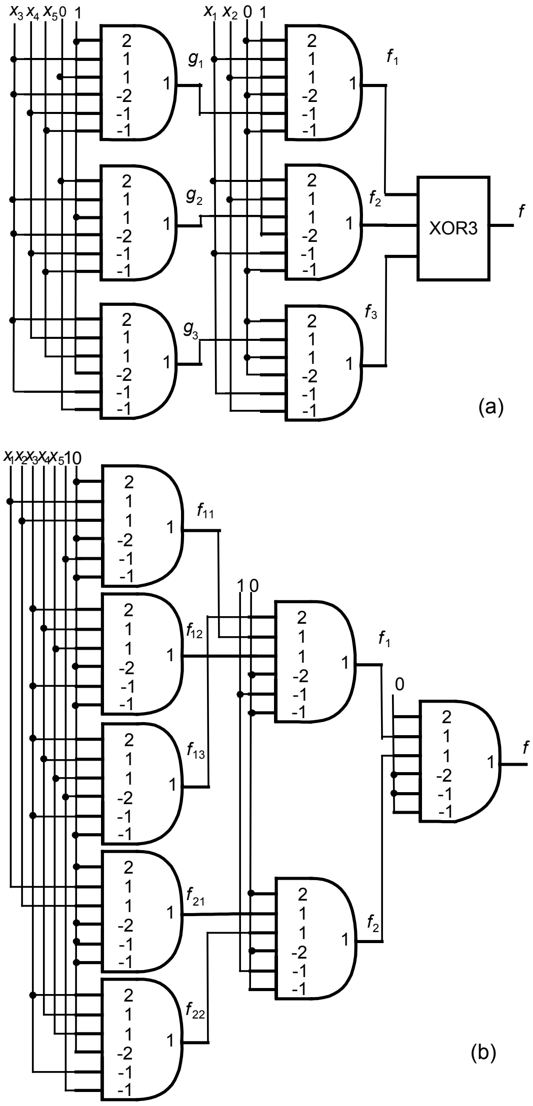

Fig. 6 The circuit for Example 4: (a) UTLG and XOR3 implementation; (b) UTLGs implementation

algorithm of n-variable logical function disjunctive decomposition to decompose the function into three-variable subset functions, the truth value matrices of the subset functions are

$$\mathbf{TM}_1=\begin{bmatrix}1&0&1&1&0&1&1&0\\0&1&0&0&1&0&0&1\\0&1&0&0&1&0&0&1\\1&0&1&1&0&1&1&0\\1&0&1&1&0&1&1&0\\0&1&0&0&1&0&0&1\\1&0&1&1&0&1&1&0\\0&1&0&0&1&0&0&1\end{bmatrix},$$

$$\mathbf{TM}_2=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\1&1&0&0&0&0&1&1\\1&1&0&0&0&0&1&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix},$$

$$\mathbf{TM}_3=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&1&1&0&0&1&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix},$$

$$\mathbf{TM}_4=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\1&0&0&1&0&1&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix},$$

$$\mathbf{TM}_5=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&1\end{bmatrix},$$

$$\mathbf{TM}_6=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}.$$

Then, $f(x_1, x_2, x_3, x_4, x_5, x_6)$ is decomposed into $f(x_1, x_2, x_3, x_4, x_5, x_6)=f_1\oplus f_2\oplus f_3\oplus f_4\oplus f_5\oplus f_6$, $f_1=\overline{x}_2x_3\overline{g}_1+\overline{x}_2x_3\overline{g}_1+x_1\overline{x}_3\overline{g}_1+x_1x_3g_1+\overline{x}_1x_2\overline{x}_3g_1+\overline{x}_1x_2x_3\overline{g}_1$, $g_1=\overline{x}_3\overline{x}_4x_5+x_3\overline{x}_4\overline{x}_5+x_3x_4x_5$, $f_2=\overline{x}_1x_3\overline{g}_2$, $g_2=\overline{x}_3x_4+x_3\overline{x}_4$, $f_3=\overline{x}_1\overline{x}_2x_3g_3$, $g_3=\overline{x}_3x_4+x_4\overline{x}_5$, $f_4=x_1\overline{x}_2x_3g_4$, $g_4=\overline{x}_3\overline{x}_4\overline{x}_5+\overline{x}_3x_4x_5+x_3\overline{x}_4x_5$, $f_5=x_1x_2x_3g_5$, $g_5=x_3x_5$, $f_6=\overline{x}_1x_2\overline{x}_3g_6$, $g_6=\overline{x}_3\overline{x}_4x_5$.

The three-variable subset functions g_1, g_2, and g_4 are non-threshold functions and can be implemented by ULG3s: $g_1=f_{11}\oplus f_{12}$, $f_{11}=\langle x_3+x_4-2x_5\rangle_1$, $f_{12}=\langle x_3+x_4\rangle_2$, $g_2=f_{21}\oplus f_{22}$, $f_{21}=\langle x_3+2x_4-x_5\rangle_1$, $f_{22}=\langle 2x_3+x_4+x_5\rangle_3$, $g_4=f_{41}\oplus f_{42}$, $f_{41}=\langle x_3+x_4-x_5\rangle_1$, $f_{42}=\langle x_3+x_4-x_5\rangle_0$. The three-variable subset functions g_3, g_5, and g_6 are three-variable threshold functions and can be implemented by UTLGs, $g_3=\langle -x_3+2x_4-x_5\rangle_1$, $g_5=\langle x_3+x_5\rangle_2$, $g_6=\langle -x_3-x_4+x_5\rangle_1$.

Continue to decompose f_1, f_2, f_3, f_4, f_5, f_6: $f_1=\overline{x}_2\overline{g}_7+\overline{x}_1x_2g_7$, $g_7=g_1$, $f_2=\overline{x}_1x_2\overline{g}_8$, $g_8=x_3\overline{g}_2+\overline{x}_3g_2$, $f_3=\overline{x}_1\overline{x}_2g_9$, $g_9=x_3g_3$, $f_4=x_1\overline{x}_2g_{10}$, $g_{10}=x_3g_4$, $f_5=x_1x_2g_{11}$, $g_{11}=x_3g_5$, $f_6=\overline{x}_1x_2g_{12}$, $g_{12}=\overline{x}_3g_6$. The three-variable subset function f_1 is a non-threshold one. It can be implemented by ULG3, $f_1=f_{13}\oplus f_{14}$, $f_{13}=\langle -g_7\rangle_0$, $f_{14}=\langle -x_1+2x_2-g_7\rangle_2$. The three-variable subset functions f_2, f_3, f_4, f_5, f_6 are threshold ones and can be implemented by UTLGs, $f_2=\langle -x_1+x_2-g_8\rangle_1$, $f_3=\langle -x_1-x_2+g_9\rangle_1$, $f_4=\langle x_1-x_2+g_{10}\rangle_2$, $f_5=\langle x_1+x_2+g_{11}\rangle_3$, $f_6=\langle -x_1+x_2+g_{12}\rangle_2$. The three-variable subset function g_8 is a non-threshold one and can be implemented by ULG3, $g_8=f_{81}\oplus f_{82}$, $f_{81}=\langle x_3+g_2\rangle_1$, $f_{82}=\langle x_3+g_2\rangle_2$. The three-variable subset functions g_7, g_9, g_{10}, g_{11}, g_{12} are threshold ones and can be implemented by UTLGs, $g_7=\langle g_1\rangle_1$, $g_9=\langle x_3+g_3\rangle_2$, $g_{10}=\langle x_3+g_4\rangle_2$, $g_{11}=\langle x_3+g_5\rangle_2$, $g_{12}=\langle -x_3+g_6\rangle_1$.

Finally the six-variable function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ can be implemented by 5 ULG3s, 13 UTLGs, and 3 XOR3s.

The proposed algorithm can implement arbitrary n-variable logical functions, and the circuit implemented by UTLGs, ULG3s, and XOR3s is simpler than by only UTLGs or ULG3s. When the n-variable function is implemented with the proposed algorithm, we need only to judge the function and decompose it. The processing of this algorithm for an arbitrary n-variable logical function is the same, so the algorithm is straightforward to program, providing a simpler algorithm for implementing an arbitrary

n-variable logical function by RTD-based universal logic gates.

5 Conclusions

A novel function synthesis algorithm is proposed which can implement an arbitrary n-variable logical function using RTD-based universal logic gates. First, a new concept of 'truth value matrix' is introduced. With the truth value matrix, a novel disjunctive decomposition algorithm is proposed, which can decompose an arbitrary n-variable logical function into three-variable subset functions. Then, a novel function synthesis algorithm is proposed, which can implement an arbitrary n-variable logical function by UTLGs, ULG3s, and XOR3s. When this proposed function synthesis algorithm is used to implement an n-variable function, if the n-variable logical function is a directly disjunctive decomposition one, the circuit structure will be very simple, and if the n-variable logical function is a non-directly disjunctive decomposition one, the circuit structure will be simpler than when using only UTLGs or ULG3s to implement the n-variable logic function. The proposed algorithm provides a new solution to implement an arbitrary n-variable logical function by RTD-based universal logic gates.

References

Altun, M., Riedel, M.D., 2012. Logic synthesis for switching lattices. *IEEE Trans. Comput.*, **61**(11):1588-1600. https://doi.org/10.1109/TC.2011.170

Beiu, V., Quintana, J.M., Avedillo, M.J., 2003. VLSI implementations of threshold logic—a comprehensive survey. *IEEE Trans. Neur. Networks*, **14**(5):1217-1243. https://doi.org/10.1109/TNN.2003.816365

Bertacco, V., Damiani, M., 1997. The disjunctive decomposition of logic functions. IEEE/ACM Int. Conf. on Computer-Aided Design, p.78-82. https://doi.org/10.1109/ICCAD.1997.643371

Czajkowski, T.S., Brown, S.D., 2008. Functionally linear decomposition and synthesis of logic circuits for FPGAs. *Comput.-Aided Des. Integr. Circ. Syst.*, **27**(12):2236-2249. https://doi.org/10.1109/TCAD.2008.2006144

Falkowski, B.J., Kannurao, S., 2001. Analysis of disjoint decomposition of balanced Boolean functions through the Walsh spectrum. *Comput. Dig. Techn.*, **148**(2):71-78. https://doi.org/10.1049/ip-cdt:20010205

Fan, D.L., Sharad, M., Roy, K., 2014. Design and synthesis of ultralow energy spin-memristor threshold logic. *IEEE Trans. Nanotechnol.*, **13**(3):574-583. https://doi.org/10.1109/TNANO.2014.2312177

Files, C.M., Perkowski, M.A., 2000. New multivalued functional decomposition algorithms based on MDDs. *Comput.-Aided Des. Integr. Circ. Syst.*, **19**(9):1081-1086. https://doi.org/10.1109/43.863648

Hrynkiewicz, E., Kolodzinski, S., 2010. An Ashenhurst disjoint and non-disjoint decomposition of logic functions in Reed-Muller spectral domain. Proc. 17th Int. Conf. on Mixed Design of Integrated Circuits and Systems, p.200-204.

Iwai, H., 2013. Future of nano CMOS technology. Proc. Symp. on Microelectronics Technology and Devices, p.1-10.

Kolodzinski, S., Hrynkiewicz, E., 2009. An utilisation of Boolean differential calculus in variables partition calculation for decomposition of logic functions. 12th Int. Symp. on Design and Diagnostics of Electronic Circuits & Systems, p.34-37. https://doi.org/10.1109/DDECS.2009.5012095

Likharev, K.K., 2008. Hybrid CMOS/nanoelectronic circuits: opportunities and challenges. *J. Nanoelectron. Optoelectron.*, **3**(3):203-230. https://doi.org/10.1166/JNO.2008.301

Liu, M.C., Lin, D.D., Pei, D.Y., 2011. Fast algebraic attacks and decomposition of symmetric Boolean functions. *IEEE Trans. Inform. Theory*, **57**(7):4817-4821. https://doi.org/10.1109/TIT.2011.2145690

Mazumder, P., Kulkarni, S., Bhattacharya, M., 1998. Digital circuit applications of resonant tunneling devices. *Proc. IEEE*, **86**(4):664-686. https://doi.org/10.1109/5.663544

Mirhoseini, S.M., Sharifi, M.J., Bahrepour, D., 2010. New RTD-based general threshold gate topologies and application to three-input XOR logic gates. *J. Electr. Comput. Eng.*, **35**(1):1-4. https://doi.org/10.1155/2010/463925

Muramatsu, N., Okazaki, H., Waho, T., 2005. A novel oscillation circuit using a resonate-tunneling diode. IEEE Int. Symp. on Circuits and Systems, p.2341-2344. https://doi.org/10.1109/ISCAS.2005.1465094

Ngwira, S.M., Tshabalala, P., 2002. Neural network analysis for the identification of optimal variable orderings in the decomposition of complex logic functions. *Comput. Dig. Techn.*, **149**(5):240-244. https://doi.org/10.1049/ip-cdt:20020405

Nikodem, M., 2013. Synthesis of multithreshold threshold gates based on negative differential resistance devices. *IET Circ. Dev. Syst.*, **7**(5):232-242. https://doi.org/10.1049/iet-cds.2012.0368

Wei, Y., Shen, J.Z., 2011. Novel universal threshold logic gate based on RTD and its application. *Microelectron. J.*, **42**: 851-854. https://doi.org/10.1016/j.mejo.2011.04.005

Yao, M.Q., Yang, K., Xu, C.Y., *et al.*, 2015. Design of a novel RTD-based three-variable universal logic gate. *Front. Inform. Technol. Electron. Eng.*, **16**(8):694-699. https://doi.org/10.1631/FITEE.1500102

Zhang, R., Gupta, P., Zhong, L., 2005. Threshold network synthesis and optimization and its application to nanotechnologies. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.*, **24**(1):107-118. https://doi.org/10.1109/TCAD.2004.839468

Zheng, Y.X., Huang, C., 2009. Complete logic functionality of reconfigurable RTD circuit elements. *IEEE Trans. Nanotechnol.*, **8**(5):631-642. https://doi.org/10.1109/TNANO.2009.2016563