基于阈值逻辑的逻辑函数综合算法研究

韦 一 沈继忠*
(浙江大学信息与电子工程学系 杭州 310027)

摘 要：阈值逻辑门由于具有强大的逻辑功能且独自构成完备集而备受关注。为了设计以阈值逻辑门为单元结构的电路，该文首先分析了谱技术与阈值函数的关系，并通过零次、一次谱系数计算阈值函数的权值和阈值。对于非阈值函数，该文提出了新的逻辑函数综合算法，可以将任意非阈值函数转化为几个阈值函数和的形式。因此，使用一个或多个阈值逻辑门组成的网络可以实现任意布尔逻辑函数。该算法为共振隧穿二极管的电路设计提供一种新方法。

关键词：电路设计；阈值逻辑；谱技术；逻辑综合
中图分类号：TP331 **文献标识码**： A **文章编号**：1009-5896(2011)07-1775-04
DOI: 10.3724/SP.J.1146.2010.01199

Research of Logic Function Synthesis Algorithm Based on Threshold Logic

Wei Yi Shen Ji-zhong
(*Department of Information Science* & *Electronic Engineering*, *Zhejiang University*, *Hangzhou* 310027, *China*)

Abstract: Threshold Logic Gate (TLG) is receiving much attention because of its logic versatility and functionally complete. For the circuit design based on TLG, a method is described to determine whether a function is threshold or not with the spectral technology. The weights and threshold can be calculated by spectral coefficients. As for non-threshold function, a novel logic synthesis algorithm is proposed, which can transform non-threshold function to the sum of some threshold functions. Furthermore, any Boolean logic function can be realized by a collection of TLG using the method in this paper. Proposed algorithm provides a method for circuit design of resonant tunneling diode.

Key words: Circuit design; Threshold logic; Spectral technology; Logic synthesis

1 引言

阈值逻辑门具有较强的逻辑功能，且与人工神经元有类似的结构，受到广泛的关注[1,2]。在阈值逻辑门的硬件电路设计中，共振隧穿二极管(Resonant Tunneling Diode, RTD)因为其电流控制和负内阻特性，体现了巨大的优势[3−5]。RTD 是一种利用量子共振隧穿效应制成的纳米电子器件。它具有速度快、频率高、电压低、功耗少和实现功能多等特点，因而在未来 VLSI 设计中有着巨大的发展潜力[6,7]。

部分文献已提出了基于 RTD 阈值逻辑门的电路综合方法[8,9]，然而对于阈值函数判断和非阈值函数分解都存在不足之处。本文结合谱技术，提出基于阈值逻辑门的布尔函数综合算法。通过此算法，任意逻辑函数可以用一个或多个阈值逻辑门组成阈值网络予以实现。

2010-11-05 收到，2011-03-10 改回
国家自然科学基金(61071062)资助课题
*通信作者：沈继忠 jzshen@zju.edu.cn

2 阈值逻辑和谱技术

2.1 阈值逻辑

一个阈值逻辑门由 n 个二值输入变量，$\{x_i\}$ $(i=1,2,\cdots,n)$，一组为整数的权值，$\{\omega_i\}(i=1,2,\cdots,n)$，一个阈值 T 和输出 f 组成，其输入输出关系为

$$f=\begin{cases}1, & \sum_{i=1}^{n}\omega_i x_i \ge T\\ 0, & \text{其他}\end{cases} \tag{1}$$

同时，阈值逻辑函数也可以由权值-阈值向量(weight-threshold vector)表示，$[\omega_1,\omega_2,\cdots,\omega_n;T]$。

如果一个布尔逻辑函数能用单一的阈值逻辑门实现，则此函数为阈值函数。否则，则称为非阈值函数。由于阈值逻辑门能够实现与、或、非逻辑，所以其具有函数完备性[8]。

2.2 谱技术

谱技术是一种数学方法，它通过转换矩阵将二值函数和变量从传统的布尔域转换到新的谱域，而

在转换过程中没有损失任何信息。由于相对于布尔域，谱域采用更大的取值范围，在 n 变量函数中谱系数 r_i 可以取从 -2^n 到 $+2^n$ 中的偶数，因此每个谱系数拥有更多的信息，很多函数性质不能在布尔域中发现，而可以在谱域中表现[10]。对于函数 $f(x_1,\cdots,x_n)$，它的谱系数列阵 $\boldsymbol{R}$ 可由式(2)得到

$$\boldsymbol{R}=\boldsymbol{T}^n\cdot\boldsymbol{F} \tag{2}$$

其中 $\boldsymbol{T}^n$ 为 n 阶 Rademacher-Walsh 转换矩阵，$\boldsymbol{F}$ 为函数在布尔域中的输出值列阵。以三变量谱系数为例，$\boldsymbol{R}=(r_0\ r_1\ r_2\ r_3\ r_{12}\ r_{13}\ r_{23}\ r_{123})^{\mathrm{T}}$，其中 r_0 称为零次谱系数，r_i 称为一次谱系数，r_{ij},r_{ijk} 称为高次谱系数。谱系数具有以下性质[11]：

(1)当 $f\leftrightarrow\overline{f}$，则全部谱系数改变符号；

(2)当 $x_i\leftrightarrow\overline{x}_i$，则相应的谱系数变换如下：

$r_i\leftrightarrow -r_i$，$r_{ij}\leftrightarrow -r_{ij}$

(3)当 $f(x)\oplus x_i$，则相应的谱系数变化如下：

$r_i\leftrightarrow r_0$，$r_{ij}\leftrightarrow r_j$，$r_{ijk}\leftrightarrow r_{jk}$

(4)当 $x_i\leftrightarrow x_i\oplus x_j$，则相应的谱系数变化如下：

$r_i\leftrightarrow r_{ij}$，$r_{ik}\leftrightarrow r_{ijk}$

3 逻辑函数综合算法

在这一节中，首先讨论如何通过谱系数区分阈值函数和非阈值函数。对于非阈值函数，结合谱系数性质，提出了新的函数综合算法，可以将非阈值函数分解成几个阈值函数和的形式。通过以上两步，任意逻辑函数都可以由一个或多个阈值逻辑门实现。在论文的以下部分，均以三变量函数为例，而新算法同样适用于更多变量的函数。

3.1 阈值函数判断

文献[10]提出基于谱系数的阈值函数分类情况，如表 1 所示。如果一个三变量函数的零次和一次谱系数，经过取绝对值并按降序排列后，其结果与表 1 中某一行的 $|r_i|$ 数值相同，则此函数为阈值函数。而阈值函数的权值与阈值可以通过表中 $|a_i|$ 得到，具体方法是：权值 $|\omega_i|=|a_i|$，且其极性与所对应的一次谱系数 r_i 相同；而阈值 T 为

$$T=\frac{1}{2}\left(\sum_{i=0}^{3}a_i+1\right) \tag{3}$$

例 1 对于三变量函数 $f=x_1x_2+x_1\overline{x}_3$，计算可得其零次和一次谱系数为(2 6 2 −2)，取绝对值并降序排列后的数据，与表 1 中 $|r_i|$ 的第 2 行相同，所以 f 为阈值函数。通过计算得到此函数的权值-阈值向量是[2,1,−1;2]。

表 1 基于谱系数 $|r_i|$ 的三变量阈值函数分类情况

$\lvert r_i\rvert$				$\lvert a_i\rvert$			
8	0	0	0	1	0	0	0
6	2	2	2	2	1	1	1
4	4	4	0	1	1	1	0

3.2 非阈值函数分解算法

在讨论如何将非阈值函数分解为阈值函数的方法之前，首先提出几个定理：

定理 1 对于阈值函数 $f(x_1,\cdots,x_n)$，如果存在以下等式：

$$f(x_1,\cdots,x_n)=x_if'(x_1,\cdots,x_n)+\overline{x}_if''(x_1,\cdots,x_n) \tag{4}$$

其中 $f'(x_1,\cdots,x_n)$ 和 $f''(x_1,\cdots,x_n)$ 为两个新函数，$1\le i\le n$，则函数 $x_if'(x_1,\cdots,x_n)$ 和 $\overline{x}_if''(x_1,\cdots,x_n)$ 均为阈值函数。

证明 从式(4)可得

$$x_if'(x_1,\cdots,x_n)=x_if(x_1,\cdots,x_n) \tag{5}$$

$$\overline{x}_if''(x_1,\cdots,x_n)=\overline{x}_if(x_1,\cdots,x_n) \tag{6}$$

由于 $f(x_1,\cdots,x_n)$ 为阈值函数，则存在以下关系：

$$f=\begin{cases}1, & \displaystyle\sum_{k=1}^{n}\omega_kx_k\ge T\\ 0, & \displaystyle\sum_{k=1}^{n}\omega_kx_k< T\end{cases} \tag{7}$$

其中 ω_k 为相应 x_k 的权值，T 为阈值。构造一个阈值函数 h 满足

$$h=\begin{cases}1, & \displaystyle\sum_{k=1,k\ne i}^{n}\omega_kx_k+\left(|T|+\sum_{k=1}^{n}|\omega_k|+\omega_i\right)x_i\\ & \displaystyle\ge T+\left(|T|+\sum_{k=1}^{n}|\omega_k|\right)\\ 0, & \displaystyle\sum_{k=1,k\ne i}^{n}\omega_kx_k+\left(|T|+\sum_{k=1}^{n}|\omega_k|+\omega_i\right)x_i\\ & \displaystyle< T+\left(|T|+\sum_{k=1}^{n}|\omega_k|\right)\end{cases} \tag{8}$$

当 $x_i=0$，则 $\sum_{k=1,k\ne i}^{n}\omega_kx_k<T+\left(|T|+\sum_{k=1}^{n}|\omega_k|\right)$，$h=0$；当 $x_i=1$ 时，式(8)与式(7)相同，$h=f$。由此可得

$$h=x_if(x_1,\cdots,x_n)=x_if'(x_1,\cdots,x_n) \tag{9}$$

所以 $x_if'(x_1,\cdots,x_n)$ 为阈值函数，同理可证 $\overline{x}_if''(x_1,\cdots,x_n)$ 也为阈值函数。 证毕

定理 2 对于一个非阈值函数 $f(x_1,\cdots,x_n)$，如果将 $x_i\oplus x_j$，$1\le i\ne j\le n$，替换 x_i 得到的新函数 $f'(x_1,\cdots,x'_i,\cdots,x_n)$ 为阈值函数，则 $x_jf(x_1,\cdots,x_n)$ 和 $\overline{x}_jf(x_1,\cdots,x_n)$ 也均为阈值函数。

证明 新函数 $f'(x_1,\cdots,x_i',\cdots,x_n)$ 中 $x_i'=x_i\oplus x_j$，则根据香农定理可得

$$f'(x_1,\cdots,x_i',\cdots,x_n)=f(x_1,\cdots,x_i\oplus x_j,\cdots,x_n)$$
$$=x_jf(x_1,\cdots,x_i\oplus 1,\cdots,x_{j-1},1,x_{j+1},\cdots,x_n)+\overline{x}_jf(x_1,\cdots,x_i\oplus 0,\cdots,x_{j-1},0,x_{j+1},\cdots,x_n)=x_jf(x_1,\cdots,\overline{x}_i,\cdots,x_n)+\overline{x}_jf(x_1,\cdots,x_n) \quad (10)$$

由于函数 $f'(x_1,\cdots,x_i',\cdots,x_n)$ 为阈值函数，根据定理 1，则 $\overline{x}_jf(x_1,\cdots,x_n)$ 为阈值函数，而 $x_jf(x_1,\cdots,\overline{x}_i,\cdots,x_n)$ 也为阈值函数。

根据前述谱系数的性质(2)可知，函数 $x_jf(x_1,\cdots,x_n)$ 与 $x_jf(x_1,\cdots,\overline{x}_i,\cdots,x_n)$ 中除了下标含有 i 的谱系数数值相同、符号相反外，其余谱系数均相同。根据阈值函数判断方法得到 $x_jf(x_1,\cdots,x_n)$ 也为阈值函数。　　证毕

定理 3　对于一个非阈值函数 $f(x_1,\cdots,x_n)$，如果经过 $f(x_1,\cdots,x_n)\oplus x_i, 1\le i\le n$，得到的新函数 $f'(x_1,\cdots,x_n)$ 为阈值函数，则 $x_if(x_1,\cdots,x_n)$ 和 $\overline{x}_if(x_1,\cdots,x_n)$ 也均为阈值函数。

证明　有函数 $f'(x_1,\cdots,x_n)$ 定义可得

$$f'(x_1,\cdots,x_n)=f(x_1,\cdots,x_n)\oplus x_i$$
$$=x_i\overline{f(x_1,\cdots,x_n)}+\overline{x}_if(x_1,\cdots,x_n) \quad (11)$$

因为 $f'(x_1,\cdots,x_n)$ 为阈值函数，根据定理 1 可得，$\overline{x}_if(x_1,\cdots,x_n)$ 和 $x_i\overline{f(x_1,\cdots,x_n)}$ 均为阈值函数。

通过谱系数的性质(1)可知，$x_i\overline{f(x_1,\cdots,x_n)}$ 的反函数 $\overline{x}_i+f(x_1,\cdots,x_n)$ 也为阈值函数。同时，

$$\overline{x}_i+f(x_1,\cdots,x_n)=\overline{x}_i+x_if(x_1,\cdots,x_n) \quad (12)$$

根据定理 1 可知 $x_if(x_1,\cdots,x_n)$ 为阈值函数。　　证毕

通过定理 2 和定理 3 可知，基于阈值逻辑的函数综合算法的关键就是如何将非阈值函数转换成阈值函数。我们以三变量函数为例，将所有函数的谱系数取绝对值、降序排列后可得表 2 所示结果。从表中可知，三变量函数谱系数的取值只有 3 种可能。所以任何一个三变量非阈值函数都可以通过性质(3)和性质(4)转换成阈值函数，然后由定理 2 和定理 3 得到综合结果。算法的具体步骤如下：

(1)计算函数的谱系数；

(2)根据函数的零次和一次谱系数判断是非为阈值函数。如果是阈值函数，由表 1 和式(3)可得函数的权值-阈值向量；

表 2 三变量谱系数分布情况

编号	谱系数							
1	8	0	0	0	0	0	0	0
2	6	2	2	2	2	2	2	2
3	4	4	4	4	0	0	0	0

(3)对于非阈值函数，根据谱系数性质(3)和性质(4)，将原函数转换成阈值函数；

(4)通过步骤(3)中参与转换的输入变量，以及定理 2 和定理 3，即可将原函数分解为若干个阈值函数和的形式。

例 2　对于三变量函数 $f=\overline{x}_1x_3+\overline{x}_2x_3+x_1x_2\overline{x}_3$，其谱系数为

0	0	0	4	0	4	4	−4
r_0	r_1	r_2	r_3	r_{12}	r_{13}	r_{23}	r_{123}

根据性质(3)，得到新函数 $f'(x_1,x_2,x_3)=f(x_1,x_2,x_3)\oplus x_3$，其谱系数为

4	4	4	0	−4	0	0	0
r_0	r_1	r_2	r_3	r_{12}	r_{13}	r_{23}	r_{123}

从其零次和一次谱系数可知，新函数为阈值函数。则根据定理 3，原函数可以用两个阈值函数和的形式表示。

$$f=f_1+f_2=x_3f+\overline{x}_3f \quad (13)$$

其中 $f_1=\overline{x}_1x_3+\overline{x}_2x_3$，其权值-阈值向量为[−1,−1,2;1]；$f_2=x_1x_2\overline{x}_3$，其权值-阈值向量为[1,1,−1;2]。

对于更多变量的函数中，以上算法依然成立。当然也存在部分非阈值函数无法直接转换成阈值函数的情况。对于这些函数，可以将原函数分解为多个函数[10]，其中每个函数都可以转换为相应的阈值函数。然后使用本文提到的算法，就可以由多个阈值逻辑门实现这些函数。

4　结束语

本文首先分析了谱技术与阈值函数的关系：通过计算和处理函数的零次、一次谱系数，可以判断其是否为阈值函数。如果是阈值函数，则可以通过计算直接得到其权值-阈值向量；而对于非阈值函数，本文提出了新的基于阈值网络的函数综合算法。此算法可以将任意非阈值函数转化为多个阈值函数和的形式。新的综合算法适合于以阈值逻辑门为单元结构的电路设计。利用 RTD 器件设计通用阈值逻辑门电路，并以此构建阈值网络将是下一步的研究方向。

参 考 文 献

[1]　潘张鑫, 马汝星, 陈偕雄. 三变量通用阈值逻辑门的设计[J]. 浙江大学学报(理学版), 2005, 32(1): 42–44.

Pan Z X, Ma R X, and Chen X X. Design of three-variable universal- threshold- logic gates [J]. *Journal of Zhejiang University* (*Science Edition*), 2005, 32(1): 42–44.

[2]　Pettenghi H, Avedillo M J, and Quintana J M. Using multi-threshold threshold gates in RTD-based logic design: a

case study [J]. *Microelectronics Journal,* 2008, 39(2): 241–247.

[3] Beiu V, Quintana J M, and Avedillo M J. VLSI implementations of threshold logic — a comprehensive survey [J]. *IEEE Transactions on Neural Networks*, 2003, 14(5): 1217–1243.

[4] Zheng Y X and Huang C. Complete logic functionality of reconfigurable RTD circuit elements [J]. *IEEE Transactions on Nanotechnology,* 2009, 8(5): 631–642.

[5] Mirhoseini S M, Sharifi M J, and Bahrepour D. New RTD-based general threshold gate topologies and application to three-input XOR logic gates [J]. *Journal of Electrical and Computer Engineering*, 2010, 35(1): 1–4.

[6] Likharev K K. Hybrid CMOS/nanoelectronic circuits: opportunities and challenges [J]. *Journal of Nanoelectronics and Optoelectronics,* 2008, 3(3): 203–230.

[7] Lee J, Choi S, and Yang K. A new low-power RTD-based 4:1 multiplexer IC using an InP RTD/HBT MMIC technology [C]. 2010 International Conference on Indium Phosphide & Related Materials, Kagawa, Japan, May 31–June 4, 2010: 1–3.

[8] Zhang R, Gupta P, and Zhong L, *et al.*. Threshold network synthesis and optimization and its application to nanotechnologies [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2005, 24(1): 107–118.

[9] Bawiec M A and Nikodem M. Boolean logic function synthesis for generalised threshold gate circuit [C]. Proceedings of the 46th Design Automation Conference, San Francisco, USA, July 26–31, 2009: 83–86.

[10] Hurst S L, Muzio J C, and Miller D M. Spectral Techniques in Digital Logic [M]. London: Academic Press Inc., 1985: 1–8.

[11] 陈偕雄, 沈继忠. 近代数字理论[M]. 杭州: 浙江大学出版社, 2001: 85–101.
Chen X X and Shen J Z. Modern Digital Theory [M]. Hangzhou: Zhejiang University Press, 2001: 85–101.

韦 一： 男，1983 年生，博士生，从事新型纳米器件电路设计的研究工作.

沈继忠： 男，1965 年生，教授，博士生导师，从事新型纳米器件电路设计、数字集成电路设计、低功耗设计、模糊逻辑和神经网络的研究工作.