第 36 卷　第 1 期　　电 子 科 技 大 学 学 报　　Vol.36　No.1
2007 年 2 月　　Journal of University of Electronic Science and Technology of China　　Feb. 2007

数字多路选择器树形网络设计理论和算法

姜文彬，姜恩华

(淮北煤炭师范学院物理系　安徽 淮北　235000)

【摘要】利用多路选择器网络可以实现任意逻辑函数的原理和布尔代数运算，提出了基于逻辑函数不相交的简化的积之和形式的数字多路选择器树形网络设计的一种代数方法，该方法可以使待设计的数字多路选择器网络化简到最小树形网络。给出的设计实例说明了该方法是有效的，容易实现数字多路选择器网络的自动综合。

关　键　词　布尔代数;　多路选择器;　不相交的简化的积之和形式;　计算机辅助设计

中图分类号　TN79+1　　　　　文献标识码　A

Theory and Algorithm for the Design of Digital Multiplexer Tree-Type Networks

JIANG Wen-bin，JIANG En-hua

(Department of Physics, Huaibei Coal Industry Teachers College　Huaibei Anhui　235000)

Abstract　In this paper, an algebraic method for the design of digital multiplexer tree-type networks based on disjoint reduced sum-of-products forms of logic functions is presented by using principle realizing any logic function with multiplexer network and Boolean algebraic operations. By using the method, the digital multiplexer networks can be simplified to minimal tree-type networks. The example given in the paper shows the method is effective. Using the method, digital multiplexer network logical synthesis can easily be accomplished automatically on a computer.

Key words　Boolean algebra;　multiplexers;　disjoint reduced sum-of-products forms;　computer aided design

多路选择器(Multiplexers，MUX)亦称数据选择器，是一种多功能通用逻辑器件(模块)。在应用多路选择器通用逻辑模块实现多变量函数时，由于受多路选择器模块的控制(地址)端个数的限制，需要选取变量集合的一个划分，并将各划分块中的变量分别配置(分配)给网络中每级网络的各个选择器的控制端，作为控制(地址)变量，以树形网络(或级联网络)实现。然而，多路选择器网络的繁简程度与该网络中每级网络的各个选择器的控制变量的配置有关。如果构成多路选择器树形网络所用的多路选择器的数目为最小，则称此网络为最小树形网络。为了合理地选择MUX网络中各个选择器的控制变量的配置，文献[1]提出了卡诺图分割法，当函数中变量超过6个时，图形分割变得复杂。文献[2]提出了Walsh谱方法，当函数中变量增加时，谱系数迅速增加。文献[3-4]是利用计算机的自动综合方法，但算法较复杂。因此，本文提出了基于逻辑函数不相交的简化的积之和(Sum-of-Products，SOP)形式的多路选择器树形网络设计的一种代数方法。讨论了对于逻辑函数的简化的SOP形式，求解某个积项(或子SOP形式)的函数限制的有关定理。在此基础上，分析了如何合理地选择MUX网络中各个选择器的控制变量的配置，使待实现函数的MUX网络的设计达到最小的树形网络，并减少搜索次数。

1　理　论

利用多路选择器通用逻辑模块实现任意布尔函数的理论基础是基于布尔函数的Shannon展开式[5-6]。对于一个定义在变量集$X=\{x_1,x_2,\cdots,x_n\}$上的布尔函数f，以及变量集X上的一个划分为：

$$\prod_{i_2\cdots i_n}=\{x_{i_1};x_{i_2},\cdots,x_{i_n}\}=\{B_1;B_2\} \tag{1}$$

式中　$i_s\in\{1,2,\cdots,n\}$，且$1\leq s\leq n$；变量集X的子集

收稿日期：2005 - 01 -13

作者简介：姜文彬(1941 –)，男，教授，主要从事数字逻辑和计算机应用方面的研究；姜恩华(1974 –)，男，硕士，讲师，主要从事信号处理和计算机网络方面的研究.

B_1和B_2为划分块，满足$B_1 \cap B_2 = \varnothing$且$B_1 \cup B_2 = X$。关于划分式(1)，函数$f$的Shannon展开式为：

$$f = \bar{x}_{i_1} f_{\bar{x}_{i_1}} + x_{i_1} f_{x_{i_1}} \tag{2}$$

式中　$f_{\bar{x}_{i_1}}$ 和 $f_{x_{i_1}}$ 分别为函数f关于文字 $\bar{x}_{i_1}$ 和 x_{i_1} 的限制[6]，及变量 x_{i_1} 被赋值为0和1时所得到的f的结果。此时，x_{i_1} 为限制变量。于是有：

$$\left.\begin{array}{l} f_{\bar{x}_{i_1}} = f(0; x_{i_2}, x_{i_3}, \cdots, x_{i_n}) = f|_{x_{i_1}=0} \\ f_{x_{i_1}} = f(1; x_{i_2}, x_{i_3}, \cdots, x_{i_n}) = f|_{x_{i_1}=1} \end{array}\right\} \tag{3}$$

一般地，对于一个定义在变量集X上的函数f及X的子集上的某个积项 $P = \dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}$，单个文字为积项$P$的特例，求函数$f$关于$P$的函数限制 $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}}$ 的运算，称为限制运算，被定义为：

$$f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}} = f|_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}=1\ 1\ \cdots\ 1} \tag{4}$$

式中　文字 $\dot{x}_{i_s}$ 表示变量 x_{i_s} 的极性，或者为文字 $\bar{x}_{i_s}$，或者为文字 x_{i_s}，即 $\dot{x}_{i_s} \in \{\bar{x}_{i_s}, x_{i_s}\}$，且1≤$s$≤$r$。式(4)表示对限制变量集合中的各限制变量被分别赋值为 $x_{i_s} = e_s$，$e_s \in \{0,1\}$，使 $\dot{x}_{i_s} = 1$, 且 $1 \le s \le r$ 时所得到的f的结果。它可以为一个平凡子函数(即常量0，1及单个文字)，或者为一个非平凡子函数。

定理 1　对于一个定义在变量集X上的函数f，以及变量集X上的一个划分为：

$$\prod_{i_{r+1} i_{r+2} \cdots i_n} = \{x_{i_1}, x_{i_2}, \cdots, x_{i_r}; x_{i_{r+1}}, x_{i_{r+2}}, \cdots, x_{i_n}\} \tag{5}$$

在函数f的某个简化的积之和(RSOP)形式[6]中，若有一个积项 $P_a = Pa = (\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r})a$，其中，$a \in \{1, P_k\}$，$P_k$是含 $\dot{x}_{i_{r+1}}, \dot{x}_{i_{r+2}}, \cdots, \dot{x}_{i_n}$ (部分或全体)的积项(单个文字为积项的特例)；并且f的这个SOP形式中除积项P_a之外的每个积项P_l中，都至少含有一个文字 $\dot{x}_{j_l} \in \{\bar{x}_{i_1}, \bar{x}_{i_2}, \cdots, \bar{x}_{i_r}\}$；则函数$f$关于积项$P$的函数限制为 $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}} = a$。

证明　设函数f的RSOP形式为：

$$\begin{aligned} f = P_a + P_1 + P_2 + \cdots + P_m = (\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}) \cdot a + \\ P_1 + P_2 + \cdots + P_m \end{aligned} \tag{6}$$

式中　积项P_1～P_m中都至少含有一个文字 $\dot{x}_{j_l} = \{\bar{x}_{i_1}, \bar{x}_{i_2}, \cdots, \bar{x}_{i_r}\}$，且$1 \le l \le m$，不妨假设 $P_l = \dot{x}_{j_l} q_l$，且$1 \le l \le m$，其中，q_1为P_l提出因子 $\dot{x}_{j_1}$ 的剩余部分，于是式(6)可写为：

$$f = (\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}) \cdot a + \dot{x}_{j_1} q_1 + \dot{x}_{j_2} q_2 + \cdots + \dot{x}_{j_m} q_m \tag{7}$$

由式(4)得：当 $\dot{x}_{i_s} = 1$，且 $1 \le s \le r$ 时，$\bar{x}_{i_s} = 0$，且1≤s≤r，所以此时 $\dot{x}_{j_l} = 0$，且 $1 \le l \le m$，$(\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r})$ =1。由式(4)、(7)得f关于积项 $P = \dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}$ 的函数限制为 $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}} = a$。　　证毕。

定理 2　对于一个定义在变量集X上的函数f，以及变量集X上的一个划分式(5)，在函数f 的某个RSOP形式中，若有两个积项 P_{t_1} 和 P_{t_2} 含有公因子 $P = \dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}$，在它们组成的子SOP形式中提出公因子后为 $P_t = P_{t_1} + P_{t_2} = Pt = (\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r})t$，其中，$t \in \{1, \dot{x}_{i_{r+1}}, \dot{x}_{i_{r+2}}, \cdots, \dot{x}_{i_n}, (P_{k_1} + P_{k_2})\}$，$P_{k_1}$ 和 P_{k_2} 为含 $\dot{x}_{i_{r+1}}, \dot{x}_{i_{r+2}}, \cdots, \dot{x}_{i_n}$ (部分或全体)的两个积项；并且在f的这个SOP形式中除 P_{t_1} 和 P_{t_2} 之外的每个积项 P_l 中，都至少含有一个文字 $\dot{x}_{j_l} \in \{\bar{x}_{i_1}, \bar{x}_{i_2}, \cdots, \bar{x}_{i_r}\}$，则函数关于积项$P$的函数限制为 $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}} = t$。

证明　类似定理1的证明可得。

定理 3　对于一个定义在变量集X上的函数f，以及变量集X上的一个划分式(5)，在函数f的某个RSOP形式中，若有积项 $P = \dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}$ 不含在该RSOP形式中，并且，在f的这个SOP形式中的每个积项P_l中，都至少含有一个文字 $\dot{x}_{j_l} \in \{\bar{x}_{i_1}, \bar{x}_{i_2}, \cdots, \bar{x}_{i_r}\}$，则函数关于积项$P$的函数限制为$f_P$= $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}} = 0$。

证明　类似定理1的证明可得。

由定理1(或定理2)可知，当一个函数f的某个RSOP形式满足定理1(或定理2)时，其中的积项P_a(或子SOP形式P_t)与该式中其余的每个积项是不相交的。若函数f的某个RSOP形式中，任何两个积项都不相交，则称此RSOP形式为不相交的RSOP(DRSOP)形式。

函数f的某个DRSOP形式中的一个积项P的尺度(大小)定义为P中所含的函数f中的最小项的数目，并记为‖P‖。若积项P中的文字的数目为r，函数f中所含变量的数目为n，则得‖P‖=2^{n-r}。

函数f中所含的变量的个数即为定义该函数的布尔空间的维数，在本文中，把它简称为函数f的维数。例如，含有n个变量的函数f称为n维函数。

定理 4　对于一个定义在变量集X上的函数f，以及变量集X上的一个划分式(5)，在函数f的某个DRSOP形式中，若有某个积项 $P_a = \dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}$，且‖$P_a$‖≥‖$P_l$‖，且1≤$l$≤$m$，其中，$m$为该DRSOP形式中，除积项$P_a$外其余的全部积项的个数。则按照划分式(5)将函数$f$分解所得到的各个函数限制(子函数)中，有平凡子函数 $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}} = 1$。

证明　利用DRSOP形式的性质以及定理1即可得证。

上述平凡子函数 $f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}}=1$ 的维数为$(n-r)$，称之为$(n-r)$维平凡子函数。

可以看出，对于定理4中的DRSOP形式，以及其中的积项P_a；若P_a可以分解为 $(\dot{x}_{j_1}\dot{x}_{j_2}\cdots\dot{x}_{j_{r-1}})\,\dot{x}_{j_r}$，其中，$\dot{x}_{j_c}\in\left\{\dot{x}_{i_1},\dot{x}_{i_2},\cdots,\dot{x}_{i_r}\right\}$，$1\le c\le r$；并且积项 $(\dot{x}_{j_1}\dot{x}_{j_2}\cdots\dot{x}_{j_{r-1}})$ 与DRSOP形式中除P_a之外的每一个积项P_l均不相交，$1\le l\le m$，则由定理1可知，若取划分{ $x_{j_1},x_{j_2},\cdots,x_{j_{r-1}}$; $x_{j_r},x_{i_{r+1}},\cdots,x_{i_n}$ }将函数f分解，所得到的各个函数限制中，有平凡子函数 $f_{\dot{x}_{j_1}\dot{x}_{j_2}\cdots\dot{x}_{j_{r-1}}}=\dot{x}_{j_r}$，该平凡子函数为一个$(n-r+1)$维平凡子函数。

可以看出，平凡子函数的维数越高，对MUX网络的化简越有利，即可省去的选择器的数目越大。

2 算法与实例

本文提出的MUX网络的设计方法是先求出函数f,用二选—选择器M(1)实现的最小树形网络(或级联网络)。然后把该网络转化为用M(c)(c>1)实现的最小MUX网络。最小化目标是使待实现函数f的MUX网络中每个选择器的数据输入端连接最大数目的平凡子函数，以省去尽可能多的选择器M(1)器件。

2.1 算 法

(1) 利用常规(经典)的最小化方法将待实现的函数f化简为DRSOP形式。

(2) 由MUX网络的最小化目标，利用定理1或(和)定理2或(和)定理3，以及定理4求函数f的变量集上的划分。该划分应满足使定理1中的a为一个平凡子函数，即 $a\in\{1,\dot{x}_{i_{r+1}},\dot{x}_{i_{r+2}},\cdots,\dot{x}_{i_n}\}$，或使$a$为可用单个M(1)实现的积项；或(和)使定理2中的t为一个平凡子函数，即 $t\in\{1,\dot{x}_{i_{r+1}},\dot{x}_{i_{r+2}},\cdots,\dot{x}_{i_n}\}$，或使$t$为可用单个M(1)实现的子SOP形式，此时$t$的一般形式可表示为 $t=\dot{x}_{l_1}\dot{x}_{l_2}+\overline{\dot{x}}_{l_1}\dot{x}_{l_3}$，其中，$\dot{x}_{l_s}\in\{\dot{x}_{i_{r+1}},\dot{x}_{i_{r+2}},\cdots,\dot{x}_{i_n}\}$，$1\le s\le 3$；或(和)使定理3中的$f_p=f_{\dot{x}_{i_1}\dot{x}_{i_2}\cdots\dot{x}_{i_r}}=0$；或(和)能生成若干分支共享的非平凡子函数(当器件带负载能力允许时)。

(3) 利用限制运算求函数限制 $f_{\overline{x}_{i_1}}$ 和 $f_{x_{i_1}}$ (或求 $f_{\overline{x}_{i_1}\overline{x}_{i_2}}$、$f_{\overline{x}_{i_1}x_{i_2}}$、$f_{x_{i_1}\overline{x}_{i_2}}$、$f_{x_{i_1}x_{i_2}}$)。

(4) 对于 $f_{\overline{x}_{i_1}}$ 或(和) $f_{x_{i_1}}$，若是不能用单个M(1)实现的非平凡子函数，则重复算法(2)～(3)，直到各函数限制或者为平凡子函数，或者为可用单个M(1)实现为止。

2.2 应用实例

例 用M(2)设计实现下述函数：

f=∑(1,3,5,7,9,10,11,17,19,21,23,25,26,27,28,29,30,31,36,37,38,39, 40,41,42,52,53,54,55,57,58)的最小MUX网络。

算法的执行过程为：

(1) 利用常规的最小化方法将待实现的函数f化简为DRSOP形式为：$f=\overline{x}_1\overline{x}_4x_6+x_1\overline{x}_3x_4+\overline{x}_1\overline{x}_3x_4x_6+\overline{x}_1x_2x_3x_4+x_3\overline{x}_4x_5\overline{x}_6+x_1x_3\overline{x}_4\overline{x}_5x_6+x_1\overline{x}_2x_3\overline{x}_4\overline{x}_5\overline{x}_6$。

(2) 由MUX网络的最小化目标，利用定理1或(和)定理2或(和)定理3，以及定理4求函数f的变量集上的划分。对于上式的积项 $x_1\overline{x}_3x_4=(x_1\overline{x}_3)x_4$，除此积项外，该式中其余各积项中都至少含有一个属于集合$\{\overline{x}_1,x_3\}$中的文字，并且 $x_4\in\{0,1,\dot{x}_2,\dot{x}_4,\dot{x}_5,\dot{x}_6\}$；对于积项 $\overline{x}_1\overline{x}_4x_6=\overline{x}_1\overline{x}_3\overline{x}_4x_6+\overline{x}_1x_3\overline{x}_4x_6$ 以及积项 $\overline{x}_1\overline{x}_3x_4x_6$，由于 $\overline{x}_1\overline{x}_3\overline{x}_4x_6+\overline{x}_1\overline{x}_3x_4x_6=(\overline{x}_1\overline{x}_3)x_6$，而积项 $\overline{x}_1x_3\overline{x}_4x_6$ 以及f中除积项 $\overline{x}_1\overline{x}_4x_6$ 和积项 $\overline{x}_1\overline{x}_3\,x_4x_6$以外其余各项都至少含有一个属于集合$\{x_1, x_3\}$中的文字，并且 $x_6\in\{0,1,\dot{x}_2,\dot{x}_4,\dot{x}_5,\dot{x}_6\}$；于是得所求的划分为$\{x_1, x_3; x_2, x_4, x_5, x_6\}$。

(3) 利用限制运算，求函数f关于 $\overline{x}_1\overline{x}_3$、$\overline{x}_1x_3$、$x_1\overline{x}_3$、$x_1x_3$ 的函数限制为 $f_{\overline{x}_1\overline{x}_3}=\overline{x}_4x_6+x_4x_6=x_6$，$f_{\overline{x}_1x_3}=\overline{x}_4x_6+x_2x_4+\overline{x}_4x_5\overline{x}_6$，$f_{x_1\overline{x}_3}=x_4$，$f_{x_1x_3}=\overline{x}_4x_5\overline{x}_6+\overline{x}_4\overline{x}_5x_6+\overline{x}_2\overline{x}_4\overline{x}_5\overline{x}_6$。

(4) 对于 $f_{\overline{x}_1x_3}$ 和 $f_{x_1x_3}$，不能用单个M(1)实现，重复算法执行过程(2)～(3)，得到结果为：$f_{\overline{x}_1x_3\overline{x}_4}=x_6+x_5\overline{x}_6$，$f_{\overline{x}_1x_3x_4}=x_2$；$f_{x_1x_3\overline{x}_4\overline{x}_6}=x_5+\overline{x}_2\overline{x}_5=x_2x_5+\overline{x}_2$，$f_{x_1x_3\overline{x}_4x_6}=\overline{x}_5$，$f_{x_1x_3x_4\overline{x}_6}=0$，$f_{x_1x_3x_4x_6}=0$。

由于得到的函数限制 $f_{\overline{x}_1x_3\overline{x}_4},f_{x_1x_3\overline{x}_4\overline{x}_6}$ 均可用单个M(1)实现，而其余的函数限制均为平凡子函数，故算法结束。

根据上面求出的各个函数限制，作出待实现的函数f的用M(1)实现最小树形网络如图1所示。将它转化为用M(2)实现的最小树形网络如图2所示。

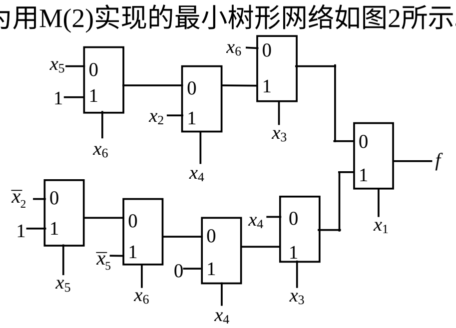

图1 函数f 用M(1)实现的最小树形网络

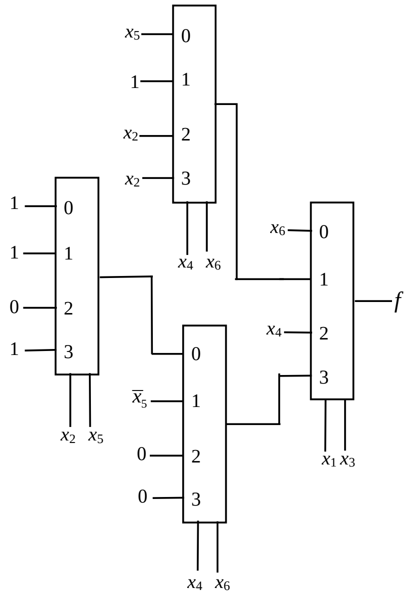

图2　函数f 用M(2)实现的最小树形网络

3　结 束 语

本文利用多路选择器网络可以实现任意逻辑函数的原理和布尔代数运算，提出了基于逻辑函数的DRSOP形式的数字多路选择器树形网络设计的一种代数方法，这种方法可以使待设计的MUX网络达到最小树形网络。然后给出了设计实例，从给出的设计实例可以看出，这种设计方法是有效的。并且由于使用的方法为一种解析方法，因而容易实现多路选择器网络的自动综合。

参　考　文　献

[1] TOSSER A J, AOUAD-SYAD D. Cascade networks of logic functions built in multiplexer units[J]. IEE Proc Pt E, 1980, 127(2): 64-68.

[2] LLOYD A M. Design of multiplexer universal-logic-module networks using spectral techniques[J]. IEE Proc Pt E, 1980, 127(1): 31-36.

[3] VOITH R P. ULM implicants for minimization of universal logic module circuits[J]. IEEE Trans Comput,1977, C-26(5): 417-424.

[4] ALMAINI A E A, MILLER J F, Xu L. Automated synthesis of digial multilexer networks[J]. IEE Proc Pt E, 1992,139(4): 329-334.

[5] SHANNON C E. The synthesis of two-terminal switching circuits [J]. Ball Syst Tech J, 1949, 28: 59-98.

[6] WANG Y, MCCROSKY C. Solving Boolean equtions using ROSOP forms[J]. IEEE Trans Comput, 1998, C-47(2): 171-177.

编　辑　孙晓丹

(上接第121页)

用伪代码描述了A^2FD算法的实现，此算法不仅达到了节点故障检测的目的，而且不额外增加系统的通信开销，并动态适应系统运行状况的变化。A^2FD算法已被应用于分布式并行数据库系统—DPSQL。为了实现分布式并行服务器节点容错，可进一步就节点容错算法等进行深入研究。

参　考　文　献

[1] LYU M R, MENDIRATTA V B. Software fault tolerance in a clustered architecture: techniques and reliability modeling [C]// In: Proc. IEEE Aerospace Applications Conference, 1999, 5: 141-150.

[2] GUPTA I, CHANDRA T D, Goldszmidt G S. On scalable and efficient distributed failure detectors[C]// In: Proc. the Annual ACM Symposium on Principles of Distributed Computing, 2001: 170-179.

[3] SU W C, WANG S C, KUO S Y. Failure detection mechanism for distributed object computing using CORBA[C]// In: Proc. 2001 Pacific Rim International Symposium on Dependable Computing, 2001: 273-280.

[4] 赵立杰，王　纲，输入训练神经网络PCA故障检测方法[J]. 系统仿真学报(增刊), 2001(13): 149-151.

[5] 李胜利，张前峰，韩宗芬，等. DRT-UNIX的故障检测技术[J]. 华中理工大学学报, 2000, 28(12): 42-44.

[6] 赵　超，张君昌. 控制系统故障检测与多模型混合故障方法[J]. 系统工程与电子技术, 2001, 23(7): 63-86.

编　辑　熊思亮