

基于量子进化算法的层次型 SOC 测试结构优化

许川佩,戴葵,马丽

XU Chuan-pei, DAI Kui, MA Li

桂林电子科技大学 电子工程学院, 广西 桂林 541004

School of Electronic Engineering, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

E-mail: dancki_sin@sina.com

XU Chuan-pei, DAI Kui, MA Li. Quantum-inspired evolutionary algorithm based on optimization of hierarchical SOC test architecture. Computer Engineering and Applications, 2008, 44(14): 96-99.

Abstract: Multilevel Test Access Mechanism (TAM) optimization is studied for the hierarchical SOC while aiming at reduction of test application time of system-on-a-chip. According to the classification of embedded IP cores, the optimization of hierarchical SOC test architecture is transformed into the optimization of flow SOC test architecture and a mathematical model using quantum-inspired evolutionary algorithm is set up. By observing the state of swarm, the results observed represent the assignment of IP core on test access mechanism and the best chromosome in current swarm. The paper realizes the optimization of hierarchical SOC test architecture while the hierarchical SOC including TAM-ed and wrapped embedded cores. The experimental results for SOC benchmark show that the proposed algorithm holds a shorter testing time when compared to the GA, ILP and heuristic algorithm.

Key words: quantum-inspired evolutionary algorithm; test architecture; hierarchical SOC

摘要:以减少系统芯片 SOC 测试时间为目标,研究了层次型 SOC 的多层次 TAM 优化问题。根据嵌入式 IP 核的分类,将层次型 SOC 测试结构优化转变成了平铺型 SOC 测试结构优化,并建立了基于量子进化算法的数学模型。通过对群体的观测,决定 IP 核在测试访问机制上的分配以及当前群体中的最佳个体,实现了包含 TAM-ed 且 wrapped 的嵌入式核的层次型 SOC 测试结构优化。针对国际标准片上系统芯片验证表明,与 GA、ILP 和启发式算法相比,该算法能够获得更短的测试时间。

关键词:量子进化算法;测试结构;层次型 SOC

DOI: 10.3778/j.issn.1002-8331.2008.14.027 **文章编号:** 1002-8331(2008)14-0096-04 **文献标识码:** A **中图分类号:** TN47

1 引言

深亚微米技术的进展允许在单一芯片上设计一个完整的电子系统,这便是系统芯片(System-On-a-Chip, SOC)。SOC 作为一种解决方案得到了越来越广泛的应用,但 SOC 的测试却是制约其设计和验证的“瓶颈”。

SOC 测试集成的普遍性问题包括外壳(Wrapper)/测试访问机制(TAM)结构的设计优化和测试调度两个方面。目前的许多测试优化研究^[3,4]都建立在 IP 核处在同一层次的假设上,即单级 TAM 的优化。但在实际中,大多 SOC 都是层次型,内部的 IP 核都不是平铺处在同一个层面上,那么多层次 TAM 的优化更符合实际的 SOC。文献[1,2]提出和建立了多层次 TAM 优化问题的一些重要概念和方法。根据嵌入式 IP 核的分类,本文将层次型 SOC 测试结构优化转变成了平铺型 SOC 测试结构优化,并建立了基于量子进化算法的数学模型。通过实验验证,该算法能够较好地减少 SOC 的测试时间,与遗传算法^[1]、ILP 和启发式算法^[2]相比,该算法能够较好地减少

SOC 的测试时间。

2 多层次 TAM 优化的分类

SOC 设计开发商向 IP 核提供商购买他们所需要的 IP 核(这里多指层次型的 IP 核),应用于他们的设计中。这些提供的 IP 核可能处于不同程度的测试结构中。例如,嵌入式核测试标准 IEEE P1500 定义了两种级别的服从认证:IEEE 1500 Wrapped Core 和 IEEE 1500 Unwrapped Core。基于 IEEE P1500 的两极服从认证,由核提供商提供的 IP 核分成三种不同的类型^[2]。类型描述中 wrapped 表示 IP 核已经存在一个 wrapper, TAM-ed 表示 IP 核包含一个内部的 TAM 结构。

(1)未 TAM-ed 且未 wrapped 的核。对于此类型的 IP 核, SOC 系统设计者必须设计一个 wrapper 和一个内部 TAM 结构。

(2)TAM-ed 但未 wrapped 的核。此类型的 IP 核内部已经设计了一个 TAM 结构,优化内部子核分配,但是核的 wrapper 结构需要由 SOC 系统设计者来设计。因此核的顶层扫描链的

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No. 60266001);广西自然科学基金(the Natural Science Foundation of Guangxi of China under Grant No. 桂科自 0542051)。

作者简介:许川佩(1968-),女,博士,教授,主要研究方向为集成电路测试理论与技术;戴葵(1981-),男,硕士研究生,主要研究方向为集成电路测试理论与技术;马丽(1982-),女,硕士研究生,主要研究方向为集成电路测试理论与技术。

收稿日期: 2007-09-04 **修回日期:** 2007-10-25

条数和长度以及内部 TAM 上的子核的总测试时间必须提供给 SOC 系统设计者,用于设计核顶层的平衡 wrapper 结构。

(3) TAM-ed 且 wrapped 的核。此类型的核已经由核提供商设计了一个内部的 TAM 结构和核顶层的平衡 wrapper 结构。对于一个曾经是 SOC 的 IP 核,尤其非常适合以此种方式提供给 SOC 系统设计者。此时,核的测试宽度是固定的。由于缺乏 I/O 和扫描链的相关信息,假定此类的 IP 核的测试数据无法再做进一步的串行或并行化处理,那么核的测试宽度无法做适应性的改变,在系统级的测试结构设计上,必须分配足够的测试宽度。

对于一个层次型 SOC,可以将嵌入式 IP 核作为一个整体参与到整个 SOC 的优化结构中。这样嵌入式 IP 核内部的子核不需要在整个 SOC 的测试优化结构中进行调度,而只在父核中进行调度,嵌入式 IP 核在 SOC 的测试优化中进行调度。层次型 SOC 测试结构优化就转化为了平铺类 SOC 测试结构优化。对于第二、三分类情况,都可以这样进行简化,区别在于第二分类情况嵌入式 IP 核内的优化需要系统设计人员进行设计优化,而属于第三分类的嵌入式 IP 核优化结构已经确定,尤其对于一个曾经是 SOC 的 IP 核。本文重点研究第三种类型情况的 TAM 优化。此种情况下,核提供商只需要提供 IP 核的测试宽度 W 和相应的测试时间 T 。根据核提供商提供的核信息不同,SOC 的测试优化分为非交互式 and 交互式的两种方式。

2.1 非交互式方式

在非交互式方式中,所提供的层次型 IP 核的信息仅仅只包括测试宽度和测试时间,记为 (W_{IP}, T_{IP}) 。在 SOC 系统级的测试调度优化中,分配给此 IP 核的 TAM 测试宽度必须满足它本身的测试宽度。多层次 TAM 优化的非交互式模型因此描述如下:

$P_{non-int}$:已知各 IP 核的测试信息,总的测试宽度 W ,为每一个核(不包括第三种类型的核,因为它们已经存在 wrapper 且无法重新设计)设计 wrapper,决定测试总宽度 W 的一个划分 $\{w_1, w_2, \dots, w_n\}$ 以及各核在此划分上的调度,以使测试时间最短,同时要满足以下约束:第三种类型的核分配测试宽度 w_i 必须满足它本身的测试宽度,即 $w_i \geq W_{IP}$ 。

2.2 交互式方式

在交互式方式中,所提供的层次型 IP 核的信息是在测试宽度 w 下 IP 核的测试时间 t 的一个集合 $\{(w^*, t^*)\}$ 。在 SOC 系统级的测试调度优化中,分配给此 IP 核的测试宽度 w 及相应的测试时间 t 必须满足 $(w, t) \in \{(w^*, t^*)\}$ 。多层次 TAM 优化的交互式模型因此描述如下:

P_{int} :已知各 IP 核的测试信息,总的测试宽度 W ,为每一个核(不包括第三种类型的核,因为它们已经存在 wrapper 且无法重新设计)设计 wrapper,决定测试总宽度 W 的一个划分 $\{w_1, w_2, \dots, w_B\}$ 以及各核在此划分上的调度,以使测试时间最短,同时要满足以下约束:第三种类型的核分配的测试宽度 w 及相应的测试时间 t 必须满足 $(w, t) \in \{(w^*, t^*)\}$ 。

3 基于量子进化算法的算法模型

3.1 SOC 优化模型划分

根据平铺型 SOC 测试结构优化问题的划分^[3],本文将转化为平铺型 SOC 测试结构优化的层次型 SOC 测试结构优化问题划分为 4 个子问题: P_W, P_{AW}, P_{PAW} 和 P_{NPAW} 。

P_W :对于一个给定的 IP 核,在 TAM 宽度一定的情况下,设计一个测试 Wrapper,使 IP 核测试时间最小。

P_{AW} :已知测试总宽度 W ,TAM 条数 B 且各条 TAM 宽度固定,在 P_W 问题基础上,将 SOC 的 IP 核合理地分配到各条 TAM 上,使整个 SOC 测试时间最小。

P_{PAW} :已知测试总宽度 W ,TAM 条数 B ,决定 W 的一个划分 $\{w_1, w_2, \dots, w_B\}$,并将 SOC 的 IP 核合理地分配到各条 TAM 上,使整个 SOC 测试时间最小。

P_{NPAW} :已知测试总宽度 W ,决定 TAM 条数 B 及一个划分 $\{w_1, w_2, \dots, w_B\}$,将 SOC 的 IP 核合理地分配到各条 TAM 上,使整个 SOC 测试时间最小。

考虑包含 N 个 IP 核和测试总宽度为 W 的 SOC,以及测试宽度 W 上的一个划分 $\{w_1, w_2, \dots, w_B\}$,其中 B 为 TAM 组数。 $T_i(w_j)$ 表示核 i 在第 j 条 TAM 上的测试时间,如果 IP 核是一个单核,则 $T_i(w_j)$ 的计算采用 P_W 问题的方法 BFD^[4],而当 IP 核是一个层次型嵌入式核,在非交互式方式中 $T_i(w_j)$ 的计算采用下面的公式:

$$T_i(w_j) = \begin{cases} \infty, & w_j < w \\ T_w, & w_j \geq w \end{cases} \quad (1)$$

其中 (w, T_w) 为层次型嵌入式核的测试宽度及测试时间。在交互式方式中:

$$T_i(w_j) = \begin{cases} \infty, & w_j \notin \{w^*\} \\ T_w, & w_j = w, w \in \{w^*\} \end{cases} \quad (2)$$

对于 P_{AW} ,本文在下一节建立基于量子进化算法的模型。 P_{PAW} 是为了求解产生最好核调度的一个划分 $\{w_1, w_2, \dots, w_B\}$,而当划分 $\{w_1, w_2, \dots, w_B\}$ 确定时, P_{PAW} 就转化为了 P_{AW} 。本文的解决中并没有采用枚举所有划分的方法^[4],而是根据 P_{AW} 中搜索算法的一个特性假设,建立了一个启发式方法,目的是降低 CPU 计算时间。 P_{NPAW} 是为了求解产生最好核调度的 TAM 分组 B 的值,当 B 已知时, P_{NPAW} 转化为了 P_{PAW} 。本文根据 P_{NPAW} 问题的理论分析^[4],只对 $2 \leq B \leq 6$ 进行了计算。

3.2 P_{AW} 问题的量子进化算法模型

为了适用多进制,本文对量子进化算法做了相应改进,以符合问题模型的要求^[5]。

设待测试的 SOC 包含 N 个 IP 核,测试时 B 条 TAM 宽度分别是 w_1, w_2, \dots, w_B ,TAM 宽度总和为 W 。 P_{AW} 问题中的个体编码表示设为:

$$q_j^t = \begin{bmatrix} \varphi 1_1 & \varphi 1_2 & \dots & \varphi 1_N \\ \varphi 2_1 & \varphi 2_2 & \dots & \varphi 2_N \\ \dots & \dots & \dots & \dots \\ \varphi B_1 & \varphi B_2 & \dots & \varphi B_N \end{bmatrix}$$

编码中的每一分量代表一个 IP 核,故个体长度等于 N 。依照观测规则生成观测值 $P_j^t = (x_1^t, x_2^t, \dots, x_N^t)$,其中 x_i^t 为整数,且 $(1 \leq i \leq N, 1 \leq x_i^t \leq B)$,表示观测值中第 i 个核被分配到第 x_i^t 条 TAM 上,测试时间为 $T_i(w_{x_i^t})$ 个时钟周期。这样就生成了 IP 核在 TAM 上的一个组合分配,即为 P_{AW} 问题解空间中的一点。本文引入二值变量 $\delta_{x_i^t}$,有

$$\delta_{x_i^t} = \begin{cases} 1, & \text{核 } i \text{ 分配到第 } x_i^t \text{ 条 TAM 上} \\ 0, & \text{其他} \end{cases}$$

那么分配在第 x_i^t 条 TAM 上的所有 IP 核完成测试所用时间如

式(3)所示

$$\sum_{i=1}^N T_i(w_{x_i^t}) \cdot \delta_{x_i^t} \quad (3)$$

因为所有的 TAM 可以同时测试,所以整个 SOC 测试的最小时间如式(4)所示:

$$\max_{1 \leq x_i^t \leq B} \left(\sum_{i=1}^N T_i(w_{x_i^t}) \cdot \delta_{x_i^t} \right) \quad (4)$$

其中 $(1 \leq i \leq N, 1 \leq x_i^t \leq B)$ 。

公式(3)就可作为 P_{AW} 问题中适应度公式。

3.3 P_{PAW} 问题中的启发式算法流程

考虑量子进化算法是一个搜索过程算法,对于每一个 TAM 的划分,通常情况下它都具有与 TAM 划分相符合的搜索速度。当 TAM 的划分是使 SOC 测试时间最短的那个划分时,在迭代次数相同的情况下,它的搜索速度应该最快,得到的结果最好。依据此点设想,本文建立了如下的算法流程。

(1) 首先设定 P_{AW} 问题的量子进化算法迭代次数为 100 进行计算,寻找可能的一些最好 TAM 划分。

(2) 随机产生一定数目的 TAM 划分。

(3) 对具体的 TAM 划分,采用 P_{AW} 问题的量子进化算法进行计算,得出各 TAM 划分的最好结果。

(4) 在步骤(3)中的结果中选取 5~8 个最好的 TAM 的划分。

(5) 设定 P_{AW} 问题的量子进化算法迭代次数为 3 000,对步骤(4)中选取的各 TAM 划分进行最终计算。

(6) 选取最好的结果即最终的结果,相对应的 TAM 划分为 TAM 的最终所求划分。

在本文算法中,对于 TAM 的所有划分只是随机选取了一定数目的划分,也可以枚举所有的 TAM 划分,因为初次设定的迭代次数为 100,计算步骤(2)中产生的所有 TAM 划分所花费的时间非常短,相比按照步骤(5)中的迭代次数设定计算,节约了大量的 CPU 计算时间。步骤(4)在计算的结果中设定了选取 5~8 个最好结果的弹性措施保证最好的划分包括在里面。量子进化算法只采用了量子门来作为更新算子,算法本身存在很大随机性。对算法中的参数经过反复试验,本文将步骤(5)中的迭代次数设定为 3 000,保证足够的迭代次数在一定程度上克服量子进化算法本身的随机性缺点。

4 实验结果

本算法实验仿真的对象选用 SOC 标准集 ITC2002 SOC Benchmark^[3] 中的 3 个 SOC:p22810、p93791 和 p34392。这 3 个 SOC 都是多层次型的,包含有层次型的嵌入式核。本文分别列出了非交互式方式和交互式方式的实验结果。实验所有结果是在 Pentium 4 CPU 2.50 GHz,512 M 内存,VC++ 6.0 的环境下运行得到的。

在非交互式方式中,p22810 中层次型嵌入式核的测试宽度设定为 8,p93791 和 p34392 的设定为 16,同文献[1,2]设定一致,结果能够进行比较。各层次型嵌入式核的测试时间由本文的方法计算得到。表1中表(a)、表(b)和表(c)分别列

表1 非交互式方式的结果

(a) p22810

W	本文的方法					文献[1]的结果			文献[2]的结果		
	B	TAM 划分	测试时间	相比[1]的提高幅/%	相比[2]的提高幅/%	B	TAM 划分	测试时间	B	TAM 划分	测试时间
16	3	8,5,3	464 789	1.52	8.11	3	5,3,8	471 993	2	8,8	505 858
24	3	8,8,8	334 049	-0.07	19.10	3	8,8,8	333 787	2	8,16	412 682
32	3	16,8,8	284 627	0	28.20	3	8,9,15	284 627	2	9,23	396 473
40	3	17,12,11	284 627	0	22.30	4	12,9,9,10	284 627	2	8,32	366 260
48	3	26,13,9	284 627	0	22.30	4	10,14,9,15	284 627	2	8,40	366 260
56	3	24,24,8	284 627	-	-	-	-	-	-	-	-
64	3	45,10,9	284 627	0	22.30	6	7,11,9,9,13,15	284 627	2	8,56	366 260

(b) p93791

W	本文的方法					文献[1]的结果			文献[2]的结果		
	B	TAM 划分	测试时间	相比[1]的提高幅/%	相比[2]的提高幅/%	B	TAM 划分	测试时间	B	TAM 划分	测试时间
32	2	16,16	918 867	0	10.00	2	16,16	918 867	2	16,16	1 021 320
40	3	16,16,8	738 897	-0.04	19.40	3	16,8,16	738 535	4	20,16,2,2	916 852
48	3	16,16,16	615 237	-0.12	9.76	3	16,16,16	614 488	3	16,16,16	681 816
56	3	24,16,16	579 953	0	8.25	3	24,16,16	579 953	5	3,3,16,16,18	632 125
64	4	16,16,16,16	461 572	3.39	11.40	4	16,16,16,16	477 773	4	16,16,16,16	521 064

(c) p34392

W	本文的方法					文献[1]的结果			文献[2]的结果		
	B	TAM 划分	测试时间	相比[1]的提高幅/%	相比[2]的提高幅/%	B	TAM 划分	测试时间	B	TAM 划分	测试时间
32	2	16,16	776 537	0	1.56	2	16,16	776 537	2	16,16	788 873
40	3	17,16,7	683 183	0.95	6.21	3	19,1,20	689 772	3	23,16,1	728 426
48	3	16,16,16	606 261	0	1.99	3	16,16,16	606 261	3	16,16,16	618 597
56	3	20,16,20	606 261	0	1.99	3	19,18,19	606 261	3	24,16,16	618 597
64	3	23,18,23	606 261	0	1.99	3	21,19,24	606 261	3	32,16,16	618 597

表2 交互式方式的结果

(a) p22810

W	本文的方法				文献[2]的结果		
	B	TAM 划分	测试 T	提高幅/%	B	TAM 划分	测试 T
16	3	8,5,3	463 285	4.24	2	5,11	483 795
24	4	11,10,2,1	302 625	16.80	2	9,15	363812
32	4	10,10,8,4	240 272	25.80	2	14,18	324 079
40	5	13,11,10,5,1	194 193	32.60	2	9,31	288 159
48	4	17,13,12,6	173 331	36.50	2	17,31	272 892
56	4	22,16,12,6	164 588	-	-	-	-
64	6	26,16,8,6,4,4	149 265	43.30	2	29,35	263 443

(b) p93791

W	本文的方法				文献[2]的结果		
	B	TAM 划分	测试 T	提高幅/%	B	TAM 划分	测试 T
16	2	8,8	1 783 572	7.44	5	4,3,3,3,3	1 927 010
24	3	12,9,3	1 193 293	7.78	4	3,4,8,9	1 293 990
32	4	12,9,8,3	887 451	-2.74	3	24,5,3	863 765
40	3	23,9,8	713 135	5.94	2	24,16	758 156
48	3	23,23,2	602 514	5.22	2	24,24	635 710
56	4	23,16,15,2	517 830	-	-	-	-
64	3	46,12,6	453 953	5.80	2	48,16	481 896

(c) p34392

W	本文的方法				文献[2]的结果		
	B	TAM 划分	测试 T	提高幅/%	B	TAM 划分	测试 T
24	2	16,8	853 459	-4.18	3	10,9,5	819 176
32	3	21,10,1	654 780	8.18	3	22,5,5	713 174
40	3	17,15,8	606 261	0	3	17,17,6	606 261
48	3	22,17,9	593 925	0	3	25,17,6	593 925
56	3	31,15,10	581 588	0	3	30,20,6	581 588
64	3	38,17,9	581 588	0	3	38,20,6	581 588

出了 p22810、p93791 和 p34392 的实验结果,表中同时还列出了文献[1,2]中的结果。从结果对比看,本文的方法与文献[1]所述方法的结果基本一致,较之文献[2]的结果都有很大的提高。对于 p22810 的结果,在 $W=32$ 时,改进幅度最大,达到了 28.2%,最小幅度也有 8.11%;对于 p93791,在 $W=40$ 时,达到了最大 19.4%;对于 p34392,在 $W=40$ 时,达到了最大 6.21%。

在交互式方式中,对 3 个 SOC 中的各层次型嵌入式核提供了测试宽度 w 与测试时间 t 对应的一组数据 $\{(w^*, t^*)\}$,

其中设定 $3 \leq w^* \leq 48$,且 $w^* \in Z$ 。各层次型嵌入式核对应测试宽度 w 的测试时间 t 由本文的方法计算得到。表 2 中表(a)、表(b)和表(c)分别列出了 p22810、p93791 和 p34392 的实验结果,由于文献[1]未能给出了交互式方式的实验结果,表中只列出了文献[2]的结果进行比较。本文的实验结果相比文献[2]的结果都有很大幅度的提高。对于划分数 B 较大时,本文的方法显示出了良好的计算能力,结果明显要比划分数 B 较小时结果好,尤其在 p22810 的实验结果中体现出来。对于 p22810,在 $W=64$ 时,改进幅度最大,达到了 43.3%;对于 p93791,在 $W=24$ 时,达到了最大 7.78%,在 $W=24$ 时,有微弱减小;对于 p34392,在 $W=24$ 有一定减小,在 $W=32$ 时,达到了最大 8.18%。

5 结束语

综上所述,层次性 SOC 的测试结构优化可以将内部嵌入式 IP 核作为整体进行调度,转化为了平铺型 SOC 的测试结构优化。针对第三种类型的 SOC,本文建立了基于量子进化算法的算法模型。对 Benchmark SOC 的实验表明,本算法能合理的调度各个嵌入式核,获得更短的测试时间。

参考文献:

- [1] Chandan Giri, Dilip Kumar Reddy Tipparthi, Santanu Chattopadhyay. Genetic algorithm based approach for hierarchical SOC test scheduling [C]//Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07), 2007.
- [2] Chakrabarty K, Iyengar V, Krasniewski M D. Test planning for modular testing of hierarchical SOC's [J]. IEEE Transactions on CAD of Integrated Circuits and Systems, 2005, 24(3).
- [3] Goel S K, Marinissen E J. Effective and efficient test architecture design for SOC's [C]//Proc Int Test Conf, 2002: 529-538.
- [4] Iyengar V, Chakrabarty, Marinissen E J. Test wrapper and test access mechanism co-optimization for system-on-chip [J]. Journal of Electronic Testing: Theory and Applications, 2002, 18.
- [5] 申抒含, 金炜东, 陈维荣. 一种基于量子进化算法的概率进化算法[J]. 计算机工程与应用, 2005, 41(33): 64-67.
- [6] Marinissen E J, Iyengar V, Chakrabarty K. A set of benchmarks for modular testing of SOC's [C/OL]//Proc Int Test Conf, 2002: 519-528. <http://www.extra.research.philips.com/itc02sochenchm>.

(上接 64 页)

族 $L-\lambda-R_0$, 给出了其 FMT 模型的三 I 支持算法、 α -三 I 支持算法。且这种算法是较简单的。算法的过程体现了这一算子族的在实际应用中的优越性。

参考文献:

- [1] 王作真. 基于含参数蕴涵算子族 $L-\lambda-R_0$ 的三 I 支持算法——仅就 FMP 模型进行讨论 [J]. 计算机工程与应用, 2008, 44(5): 102-104.
- [2] Zadeh L A. Outline of new approach to the analysis of complex systems and decision processes [J]. IEEE Trans on Systems, Man and Cybernetics, 2003, 3: 28-33.
- [3] 王国俊. 模糊推理的全蕴涵三 I 算法 [J]. 中国科学: E 辑, 1999, 29(1): 43-53.
- [4] 王国俊. 非经典数理逻辑与近似推理 [M]. 北京: 科学出版社, 2000.

- [5] 王国俊. 模糊推理的一个新方法 [J]. 模糊系统与数学, 1999, 13(3): 1.
- [6] 王国俊. 一种新型的三 I 算法及其逻辑基础 [J]. 自然科学进展, 2003, 13(6): 575.
- [7] 张兴芳, 孟广武. 蕴涵算子族及其应用 [J]. 计算机学报, 2007, 30(3): 448-453.
- [8] 彭家寅. 基于某些常见蕴涵算子的反向三 I 算法 [J]. 自然科学进展, 2005, 15(4): 404.
- [9] 彭家寅. 基于某些常见蕴涵算子的模糊推理全蕴涵三 I 约束算法 [J]. 自然科学进展, 2005, 15(5): 539.
- [10] 王国俊. 系统 H_α 中的广义重言式理论 [J]. 陕西师范大学学报, 2003, 31(2): 1-11.
- [11] 吴望名. 参数 Kleene 系统中的广义重言式 [J]. 模糊系统与数学, 2000, 14(1): 1-7.