

长互连链延时功耗建模与基于混合粒子群算法的优化

秋小强^{①②} 杨海钢^{*①} 周发标^{①②} 谢元禄^①

^①(中国科学院电子学研究所 北京 100190)

^②(中国科学院研究生院 北京 100049)

摘要: 延时和功耗已经成为 VLSI 两个关键因素, 该文提出一种用于综合考虑功耗和延时的片上寄存器长链设计方法, 首先建立由基于 Elmore 延时模型和综合功耗模型的能量延时积模型。提出一种混合进化粒子群算法, 通过引入新的惯性权值算子和混合变异操作, 克服了传统算法收敛速度慢, 早熟及局部收敛等缺陷, 同时采用基准函数对该算法进行测试, 验证了算法的有效性。最后, 将该算法应用于基于最小能量延时积的长链设计中, 和最小延时模型相比, 该方法所得到的功耗延时积减少 26.34%, 能量延时积减少 18.74%, HSPICE 仿真验证了设计方法的有效性。

关键词: 片上互连; 缓冲器; 延时; 能耗; 粒子群算法

中图分类号: TN402

文献标识码: A

文章编号: 1009-5896(2011)06-1481-06

DOI: 10.3724/SP.J.1146.2010.01114

Analysis of Delay-power Model of Long Chain and Optimization Based on Hybrid Evolution Particle Swarm Algorithm

Qiu Xiao-qiang^{①②} Yang Hai-gang^① Zhou Fa-biao^{①②} Xie Yuan-lu^①

^①(Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China)

^②(Graduate University of the Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Interconnect delay and power consumption are two of the main issues in deep-submicron meter technology and nano-meter technology. This paper proposes a long chain design method which takes power consumption and delay into consideration. This paper proposes a hybrid evolution particle swarm algorithm which by introducing inertia weighted operator and hybrid mutant operation overcomes such drawbacks such as low convergent speed, prematurity and local convergence. Tests employing benchmark function prove that the proposed algorithm is valid and efficient. The algorithm is applied to long chain design based on minimum energy delay product, simulation results show that in minimum power delay model the PDP is 26.34% lower than in minimum delay model, while in minimum energy delay model the EDP is 18.74% lower than in minimum delay model, simulations indicate the efficacy of such design method with HSPICE.

Key words: On-chip interconnect; Repeater; Delay, Power consumption; Particle swarm algorithm

1 引言

半导体工艺进入深亚微米甚至纳米阶段后, 随着集成电路芯片频率和规模的不断扩大, 片上长连线在延时、功耗和信号完整性都面临着严峻的挑战, 已经成为阻碍集成电路性能提高的瓶颈之一^[1]。

减少长连线延时的常用方法是在连线上等距离地插入缓冲器, 使总延时随线长线性增长。插入缓冲器的方法简单可靠, 但随着工艺的缩小和连线长度的增加, 插入缓冲器的尺寸越来越大, 数量越来越多。对于大规模高性能的设计而言, 在深亚微米时缓冲器的数量将会大于 10^6 个, 同时缓冲器的功

耗也占到总功耗的 40%~70%, 将导致延时、功耗和面积等严重问题^[2,3]。近几年的一些研究^[4,5], 都是以一定的延时增加来换取功耗的减小, 但如何取舍并没有合理的分析和推导。另外一些研究则通过降低互连线上电压的摆幅来降低功耗和延时, 但会增加硬件开销和噪声的影响^[6-9]。因此, 如何合理地量化缓冲器的尺寸和数量, 使长连线的功耗延时积和能量延时积最小, 已经成为芯片设计所必须解决的关键问题之一。

本文以功耗延时积和能量延时积为优化目标, 基于 Elmore 连线延时模型和综合功耗模型^[1], 提出了一种基于混合进化策略粒子群算法 (Hybrid Evolution Particle Swarm Algorithm, HEP-PSO) 的长链设计方法。该方法可以准确和高效地得到最优的

缓冲器的尺寸和数量,使长链获得最小的能量延时积。

2 长链的最优结构模型的建立和优化方法

2.1 延时模型的构建

设长链共有 N 个负载,每个负载的电容为 C ,每个负载的电阻为 R ,标准反相器的输入栅电容为 C_g ,本征输出电容为 C_{int} ,反相器的导通电阻为 R_{ref} , n 为管子尺寸的增大倍数,且将长链分为 m 节,则在分布 RC 模型下,长链的总的互连延时为

$$\tau = m\tau_m = mR_{ref}(C_{int} + C_g) + \frac{N}{n}R_{ref}C + \frac{N^2}{2m}RC + NRn C_g \quad (1)$$

2.2 功耗模型的构建

缓冲器的功耗主要由 3 部分组成,分别为动态功耗 (switching power)、短路功耗 (short-circuit power) 和静态功耗 (leakage power) 组成^[5]。

当反相器尺寸增大 n 倍,且将长链分为 m 节,则总的功耗为

$$P_{total} = mk_1 n(C_{int} + C_g) + k_1 NC + k_2 n + mk_3 n \quad (2)$$

其中 $k_1 = \alpha V_{DD}^2 f_{clk}$, $k_2 = \frac{1}{2} \alpha \tau \ln \left(\frac{V_{DD} - |V_{tp}|}{V_{tn}} \right) V_{DD} W_{n_{min}}$

$\cdot I_{sat} f_{clk}$, $k_3 = (3/2) V_{DD} I_{off} W_{n_{min}}$ 。其中 V_{DD} 为电源电压, f_{clk} 为时钟频率,本文中 α 为时钟通道的开关活性因此取为 1, I_{off} 为 N 管的静态漏电流, I_{sat} 为 N 管的饱和电流, V_{tp} 和 V_{tn} 分别为 P 管和 N 管的阈值电压, $W_{n_{min}}$ 为 N 管的最小宽度。

2.3 功耗延时最优模型

通过对式(1)和式(2)的分析。可以看出,随着 m 和 n 的减小,延时虽有所增加,但功耗将大幅减小,我们要试图寻找一种折中,使能量延时积最小。根据式(1)和式(2)延时功耗和缓冲器尺寸增大倍数 n 与长链节数 m 的关系见图 1。

假设实际的设计采用 $m = xm_{opt}$, $n = yn_{opt}$; 其中

m_{opt} 和 n_{opt} 为延时最小时的缓冲器尺寸和长链节数, x 和 y 为设计的尺寸和节数同延时最优的尺寸和节数之比。

总的延时为

$$t_p(x, y) = 0.69\tau = 0.69N\sqrt{RR_{ref}CC_{int}}(x + 1/x + y + 1/y) \quad (3)$$

总的功耗为

$$P_{total}(x, y) = xyk_1 NC + k_1 NC + yk_2 \sqrt{\frac{R_{ref}C}{RC_{int}}} + \frac{1}{2} xyk_3 N \frac{C}{C_{int}} \quad (4)$$

则功耗延时积为

$$PDP(x, y) = P_{total} t_p \quad (5)$$

则能量延时积为

$$EDP(x, y) = P_{total} t_p^2 \quad (6)$$

通过选择合适的 x 和 y 值可得出最小能量延时积的缓冲器尺寸和长链的节数。

3 功耗和延时模型的 HSPICE 仿真

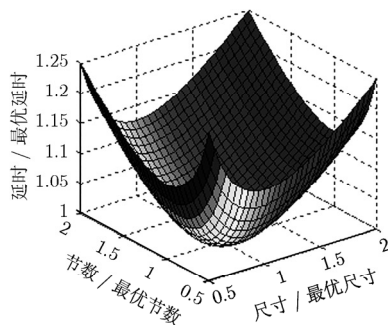
为验证本文提出的方法,我们在 1.5 V 0.13 μm CMOS 工艺下进行了 Hspice 仿真。其中 N 为 6000, $C=3$ fF, 单位连线电阻 $R=5 \Omega$, $f_{clk}=50$ MHz。模型参数来自 Chartered 0.13 μm 工艺^[10]。

同时,针对不同的单位负载,仿真得到的延时和功耗和模型计算出的延时和功耗对比如表 1。

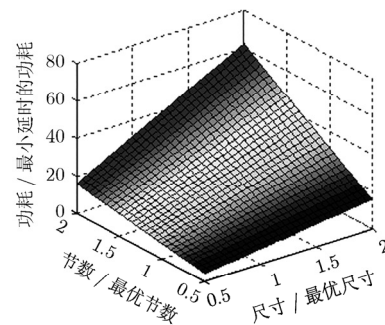
通过功耗和延时的仿真测量和模型计算的对比,可以看到,对于不同的单位负载,功耗的最大误差为 2.87%,延时的最大误差为 3.59%,说明建立的功耗和延时模型是可靠的。

4 基于混合进化的粒子群算法

粒子群优化 (Particle Swarm Optimization, PSO) 算法是由 Kennedy 和 Eberhart 于 1995 年提出的一种基于群智能的随机优化算法^[11]。适用于求解大量非线性、不可微和多峰值的复杂优化问题^[12,13]。



(a) 延时与尺寸、节数的关系



(b) 功耗与尺寸、节数的关系

图 1 延时和功耗与尺寸和节数的关系

表1 模型和仿真的功耗和延时对比

单位负载(fF)	模型延时(ns)	仿真延时(ns)	误差(%)	模型功耗(mW)	仿真功耗(mW)	误差(%)
3	11.84	11.43	3.58	4.31	4.19	2.86
4	13.67	13.32	2.63	5.78	5.66	2.12
5	15.28	14.77	3.45	7.18	6.98	2.86
6	16.74	16.26	2.95	8.64	9.41	2.44
7	18.08	17.56	2.96	10.06	9.78	2.86
8	19.33	18.67	3.53	11.53	11.21	2.85
9	20.51	19.81	3.53	12.92	12.57	2.78
10	21.62	20.87	3.59	14.36	13.96	2.86
11	22.67	21.90	3.51	15.76	15.33	2.80
12	23.68	22.88	3.49	17.20	16.72	2.87

由于 PSO 算法的程序实现异常简洁,需要调整的参数也少,因而已应用于多个科学和工程领域。粒子群优化算法首先初始化一群随机粒子,然后通过迭代找到最优解。在每一次迭代中,粒子通过跟踪当前自身找到的个体最优值 p_{best} 和整个种群找到的全局最优值 g_{best} , 并根据

$$v_k = wv_{k-1} + c_1 \text{Rand}(p_{best} - p_{present}) + c_2 \text{Rand}(g_{best} - p_{present}) \quad (7)$$

$$p_{present_k} = p_{present_{k-1}} + v_k \quad (8)$$

来更新自己的位置和速度。其中 v 是粒子的速度, $p_{present}$ 是粒子当前的位置, Rand 是(0,1)之间的随机数, c_1 和 c_2 被称为学习因子,表示粒子向个体极值和全局极值的学习程度, w 为惯性因子,表示粒子全局搜索能力。

但是, PSO 算法作为一种新的随机搜索算法, 仍旧存在着早熟收敛和收敛较慢这两个难题, 并且具有种群多样性随代数增加下降过快, 有可能不收敛到全局最优解等缺点。针对这一问题, 一些研究通过对算法的基本更新公式的改进来提高算法的寻优能力^[11,12], 还有一些研究通过引入其他的类型的优化算法来增加算法的全局搜索能力^[13,14], 虽然这些研究工作已经给出了提高算法全局搜索能力的方法, 但是它们很难在提高搜索速度和保持种群多样性之间达到平衡。

本文提出了一种基于混合进化的粒子群算法, 分析了惯性权重变化对算法全局和局部搜索能力的影响, 提出了混合惯性权重的概念, 平衡了算法的全局和局部搜索能力, 提高了算法的搜索速度。同时, 该算法借鉴了遗传算法的变异算子, 运用了两种变异算子, 有效地保持了算法种群的多样性, 克服算法的早熟收敛。最后, 对算法的个体学习因子

c_1 和社会学习因子 c_2 进行了不同的调整。在算法的迭代前期, 个体学习因子 c_1 较大而社会学习因子 c_2 较小, 使粒子保持较强的个体搜索能力, 防止陷入早熟收敛。而在算法的迭代后期, 个体学习因子 c_1 较小而社会学习因子 c_2 较大, 保证算法快速收敛。

4.1 惯性权值的改进

惯性权重 w 是用来控制粒子以前速度对当前速度的影响, 它将直接影响粒子的全局和局部搜索能力。惯性权重取值较大时, 全局寻优能力强, 局部寻优能力弱。反之, 则局部寻优能力增强, 而全局寻优能力减弱。一般而言, 在算法的初始阶段, 惯性权重较大, 使算法有较强的全局寻优能力, 在算法的后期阶段, 惯性权重较小, 以增强局部寻优能力。但这样也使算法容易早熟收敛, 无法求得全局最优解。

针对这一情况, 本文提出了混合惯性权重算子, 主导思想是在算法的奇次迭代时, 开始惯性权重较大, 算法后期惯性权重较小, 而在算法的偶次迭代时, 开始惯性权重较小, 算法后期惯性权重较大。这样, 在算法的不同时期, 算法都能保持较强的全局和局部寻优能力。改进后的混合权重算子为式(9)和式(10), 其中 i 为迭代次数, i_{max} 为最大迭代次数, w_{st} 和 w_{end} 为两个惯性权重。

算法的奇数次迭代

$$w(i) = (w_{st} - w_{end}) \times (i / i_{max})^2 - 2 \times (w_{st} - w_{end}) \times (i / i_{max}) + w_{st} \quad (9)$$

算法的偶数次迭代

$$w(i) = (w_{st} - w_{end}) \times (i / i_{max})^2 + w_{end} \quad (10)$$

4.2 两种变异算子的引入

针对算法运行后期容易早熟收敛, 无法搜索全局最优解的情况, 本文引入了两种变异算子, 当群

体的多样性低于预先给定的阈值时开始变异操作。变异分为两种情况,在算法奇数次迭代时,以概率 p_1 在最优解的附近变异,提高算法的局部搜索能力。其中 P_k 为第 k 个粒子当前的位置, $P_{k,best}$ 为第 k 个粒子历史最优位置。

$$P_k = P_{k,best}(1 + 0.5 \times \text{rand}n) \quad (11)$$

在算法的偶数次迭代时,以概率 p_2 在全局变异,提高算法的全局搜索能力,防止算法陷入早熟收敛。其中 p_2 远小于 p_1 ,防止算法退化为随机搜索算法。其中 SP 为整个算法的搜索空间。

$$P_k = \text{SP} \times \text{rand}n \quad (12)$$

4.3 学习因子的设置

学习因子设置为二次开口曲线。随着迭代次数的增加, c_1 从 c_{st} 变为 c_{end} , c_2 从 c_{end} 变为 c_{st} 。

$$c_1 = (c_{st} - c_{end}) / (i / i_{max})^2 + 2 \times (c_{end} - c_{st}) (i / i_{max}) + c_{st} \quad (13)$$

$$c_2 = (c_{end} - c_{st}) / (i / i_{max})^2 + 2 \times (c_{st} - c_{end}) (i / i_{max}) + c_{end} \quad (14)$$

4.4 仿真实验

为了测试 HEP SO 算法的优化能力,本文从两个方面来考察算法的性能:(1)固定函数评价次数,来考察算法的精度,(2)设定一定的精度要求,来考察算法所需要的函数评价次数。HEP SO 算法的参数选取为 $w_{st}=0.9$, $w_{end}=0.1$, $c_{st}=3$, $c_{end}=1$, $p_1=0.1$, $p_2=0.01$ 。

4.4.1 固定函数评价次数 本节测试是在一定的函数评价次数的前提下,评估 HEP SO 算法的优化能

力,对比算法为文献[12]提出的 PLOYPSO 算法和文献[11]提出的 PSO-FCI 算法。实验环境为:种群规模 30,函数评价次数 9 万次,采用文献[12]中采用的 4 个基准测试函数,函数维数为 30,每个函数运行 50 次,3 种算法的比较结果如表 2。

通过表 2 可以看出,本文提出的 HEP SO 算法在最优均值和最优方差上都优于 PLOYPSO 算法和 PSO-FCI 算法,尤其在较难优化的 Rosenbrock 函数也取得较优的结果。说明 HEP SO 算法具有较强的全局寻优能力。

4.4.2 给定一定精度 本节测试是在给定一定的精度下,比较算法的均值,方差以及函数评价次数,对比算法为 PSO-FCI 算法。实验环境为:种群数 30,最大迭代次数 3000,函数维数 30,每个函数独立运行 50 次。实验结果见表 3。

表 3 的实验结果表明本文提出的 HEP SO 算法所需要的函数评价次数优于 PSO-FCI 算法,也就是说对于指定精度,HEP SO 算法只需要较少的迭代次数就能满足计算要求,说明 HEP SO 算法具有较快的收敛能力。

5 基于混合进化粒子群优化算法求解 x 和 y 值

对于求解 x 和 y 值,如采用一般的数学解析法,则计算量较大,并且功耗和延时不是互相独立的。因此本文采用混合进化粒子群优化算法来求解 x 和 y 值。

算法设置种群数为 30。迭代次数为 50 次,通

表 2 HEP SO 与 POLYPSO 和 PSO-FCI 算法的性能比较

测试函数	最小值		均值			标准差		
	PSO-FCI	本文算法	PLOYPSO	PSO-FCI	本文算法	PLOYPSO	PSO-FCI	本文算法
Rosenbrock	1.30e-8	1.00e-8	87.08	1.09e-4	2.72e-5	43.64	2.24e-4	2.92e-5
Griewank	0	0	0.1245	0	0	0.2170	0	0
Rastrigin	0	0	49.79	4.77e-10	0	14.44	2.15e-9	0
Sphere	0	0	7.266e-4	0	0	1.859e-3	0	0

表 3 HEP SO 与 PSO-FCI 算法的性能比较

测试函数	要求精度	均值		方差		函数评价次数	
		PSO-FCI	本文算法	PSO-FCI	本文算法	PSO-FCI	本文算法
Rosenbrock	1e-3	8.58e-4	8.25e-4	3.64e-8	4.84e-8	81738	20540
Griewank	1e-4	7.22e-5	6.66e-5	2.07e-10	7.79e-10	4004	3960
Rastrigin	1e-4	8.01e-5	7.38e-5	1.66e-10	5.22e-10	5846	3630
Sphere	1e-4	7.27e-5	6.47e-5	3.71e-10	5.42e-10	2460	1600

过迭代求出一个最优二维粒子, 也就得出 x 和 y 的最优值。通过上文可知功耗延时积为 Pt , 能量延时积为 Pt^2 。因此, 更进一步可以用 $P^i t^j$ 表示能耗延时积, 这个表达式允许采用不同的指数 i 和 j 来确定能耗和延时的权重。通过图 2 可以看到, 当表示功耗权重的 i 较大时, 算法主要考虑功耗最小, 此时 x 和 y 值均较小, 意味着划分的节数较小且缓冲器的尺寸较小, 长链的功耗较小而延时较大。当表示延时权重的 j 较大时, 算法重点考虑延时最小, 此时 x 和 y 值均较大, 意味着划分的节数较大且缓冲器的尺寸较大, 长链的延时较小而功耗较大。

最小延时模型的 i 和 j 分别为 0 和 1, 意味着只考虑延时而不考虑功耗, 因此其仿真结果延时最小而功耗最大。采用最小能量延时模型时 i 和 j 分别为 1 和 2, 表示开始考虑功耗, 但延时仍比功耗重要。当采用最小功耗延时模型时 i 和 j 分别为 1, 表示功耗和延时同等重要, 其仿真结果延时最大而功耗最小。

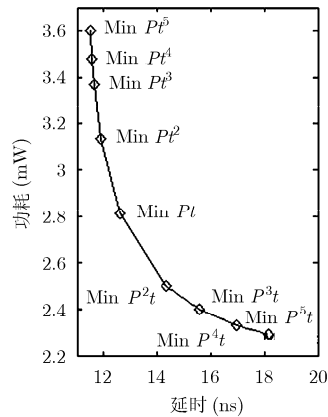
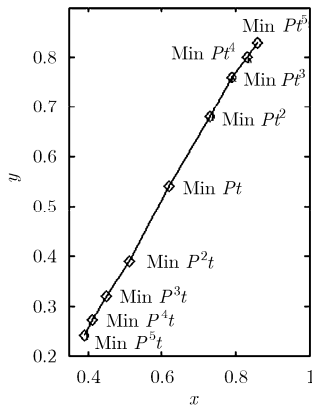


图 2 不同权重的最小能耗延时积对应的 x 和 y 值及功耗和延时

表 4 3 种典型模型的仿真数据

	x	y	延时(ns)	功耗(mW)	功耗延时积(fJ)	能量延时积(fJ.ns)
最小延时模型	1	1	11.39	4.19	47.72	543.53
最小功耗延时模型	0.66	0.50	12.69	2.77	35.15	446.07
最小能量延时模型	0.76	0.65	11.86	3.14	37.24	441.67

参考文献

[1] Rabaey J M and Chandrakasan A. Digital Integrated Circuits: A Design Perspective[M]. Second Edition, Prinitice Hall Electronics and VLSI Series, 2003: 19-23.
 [2] Banerjee K and Mehrotra A. A power-optimal repeater insertion methodology for global interconnects in nanometer designs[J]. *IEEE Transactions on Electron Devices*, 2002, 49(11): 2001-2007.

6 结论

本文提出了一种采用混合进化粒子群算法的长链设计方法。它综合考虑了功耗和延时对设计的影响, 采用量化的指标来确定其最优值。算法从惯性权值, 学习因子和选择变异 3 个方面进行改进, 仿真实验从两个方面验证了本文提出的 HEP SO 算法有较强的寻优能力。最后, 将该算法应用于求解最小能量延时模型下的最优缓冲器尺寸和长链的节数。仿真结果表明, 该方法采用的功耗模型和延时模型最大误差仅为 2.87%和 3.59%。对于具体设计而言, 该方法可以通过调整权重指数 i 和 j 来确定设计方向, 和最小延时模型相比, 该方法所得到的功耗延时积减少 26.34%, 能量延时积减少 18.74%。该方法适用于片上互连线(特别是带有较大负载的长连线)设计。随着深亚微米直至纳米工艺的广泛应用, 功耗问题尤其静态功耗将大幅增加, 该方法的可用性和有效性将进一步提高。

[3] Gelsinger P P. Microprocessors for the new millennium: challenges, opportunities and new frontiers[C]. *IEEE International Solid-State Circuits Conference 2001*, Hillsboro OR USA, Feb. 5-7, 2001: 22-25.
 [4] Li Rui-ming, Zhou Dian, and Liu Jin, *et al.* Power-optimal simultaneous buffer insertion/sizing and wire sizing for two-pin nets[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, 24(12):

- 1915-1924.
- [5] Cheung W T and Wong N. Power optimization in a repeater-inserted interconnect via geometric programming[C]. ISLPED'06, Tegernsee, Germany, October 4-6, 2006: 226-231.
- [6] Jiang Iris Hui-ru and Wu Ming-hua. Power-state-aware buffered tree construction[C]. 2008 IEEE International Conference on Computer Design (ICCD 2008), Squaw Greek, Lake Tahoe, California, October 12-15, 2008: 21-26.
- [7] Mohammad Moghaddam Tabrizi and Nasser Masoumi. Low-power and high-performance techniques in global interconnect signaling[J]. *Microelectronics Journal*, 2009, 40(10): 1487-1495.
- [8] Narasimhan A, Kasotiya M, and Sridhar R. A low swing differential signaling scheme for on chip global interconnects[C]. Proceedings of the 18th International Conference on VLSI Design jointly with 4th International Conference on Embedded Systems Design, Taj Bengal Kolkata India, Jan. 3-7, 2005: 634-639.
- [9] Dave V, Baghini M S, and Sharma D K. A process variation tolerant, high-speed and low-power current mode signaling scheme for on-chip interconnects[C]. GLSVLSI'09, Boston, Massachusetts, USA, May 10-12, 2009: 389-392.
- [10] Chartered Semiconductor Manufacturing Document Title: 0.13LP EP SPEC Rev1B.doc 0.13 μm CMOS low power logic /mixed signal/rf technology electrical parameters specification, www.foundryview.com/ebs/oa-Servlets /Appslogin.
- [11] 吕强, 刘士荣. 一种信息充分交流的粒子群优化算法[J]. 电子学报, 2010, 38(3): 664-667.
- LV Qiang and Liu Shi-rong. A particle swarm optimization algorithm with fully communicated information[J]. *Acta Electronica Sinica*, 2010, 38(3): 664-667.
- [12] 张长胜, 孙吉贵, 欧阳丹彤. 一种自适应离散粒子群算法及其应用研究[J]. 电子学报, 2009, 37(2): 299-304.
- Zhang Chang-sheng, Sun Ji-gui, and Ouyang Dan-tong. A self-adaptive discrete particle swarm optimization algorithm[J]. *Acta Electronica Sinica*, 2009, 37(2): 299-304.
- [13] Veenhuis C. Particle swarm optimization with polymorphic update rules[C]. 2009 International Conference on Adaptive and Intelligent Systems, ICAIS'09, Klagenfurt, Austria, Sep. 24-26, 2009: 135-140.
- [14] Hassan A and Phillips C. Chaotic particle swarm optimization for dynamic routing and wavelength assignment in all-optical WDM networks[C]. International Conference on Signal Processing and Communication Systems, 2009. Omaha, United States, Sep. 28-30, 2009: 1-7.
- 秋小强: 男, 1973年生, 博士生, 研究方向为大规模集成电路设计方法.
- 杨海钢: 男, 1960年生, 研究员, 博士生导师, 主要从事数模混合信号SoC设计和大规模集成电路设计等方面的研究工作.
- 周发标: 男, 1984年生, 博士生, 研究方向为大规模集成电路可测性设计方法.
- 谢元禄: 男, 1980年生, 研究实习员, 研究方向为可编程逻辑电路的配置系统.