

基于聚类的异步时钟 SoC 测试

凌 立¹, 江建慧¹, 张 颖¹, 王 真²

(1. 同济大学软件学院, 上海 201804; 2. 上海电力学院计算机科学与技术学院, 上海 200090)

摘 要: 为进一步减少片上系统(System-on-Chip, SoC)测试耗时、降低测试成本, 本文结合异步时钟测试机制, 提出一种基于聚类的测试调度方法. 该方法利用了 SoC 各测试的特征以及异步时钟测试的特点, 对测试数据进行预处理. 在 ITC'02 基准 SoC 集上, 将本文方法与未采用异步时钟机制以及基于混合整型线性规划模型求解的方法进行对比. 结果表明, 本文的方法分别能平均减少测试耗时 20.39% 和 5.53%, 提升了调度算法的优化效率. 并且在功耗约束较强时, 最终调度结果与耗时下界仅相差 0.9%.

关键词: SoC 测试调度; 异步时钟; 混合整型线性规划模型; 聚类

中图分类号: TP391.7 **文献标识码:** A **文章编号:** 0372-2112 (2017)03-0740-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.03.034

Clustering-Based Asynchronous Clock Periods SoC Testing

LING Li¹, JIANG Jian-hui¹, ZHANG Ying¹, WANG Zhen²

(1. School of Software Engineering, Tongji University, Shanghai 201804, China;

2. College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200030, China)

Abstract: The clustering-based SoC test scheduling algorithm combined with the asynchronous clock periods testing is proposed to further reduce the SoC (System-on-Chip) test application time (TAT) and test cost. The scheduling algorithm pre-processes the test data by exploiting the characteristics of the tests. After conducting experiments on the ITC'02 SoC benchmark, we find out that the proposed scheduling method based on clustering can reduce TAT by 20.39% and 5.53% on average, when comparing with a synchronous clock testing method and an asynchronous method based on the MILP model, respectively. Besides, when the power constraint is tight, there is only a difference of 0.9% between the scheduling result and the lower bound.

Key words: SoC test scheduling; asynchronous clock; MILP model; clustering

1 引言

随着测试成本在芯片生产流程中所占比例日趋上升, 集成了多个芯核的 SoC 测试问题也逐渐复杂, 如需要面临测试数据量增加、测试耗时增长、测试功耗和温度增高等问题. 此外, 大部分芯片的测试仍然需要昂贵的自动测试仪 (Automatic Test Equipment, ATE). 因此减少应用测试所耗的时间 (Test Application Time, TAT) 是降低测试成本的重要环节. 就 SoC 而言, 利用测试间的并行性, 通过测试调度进行并发测试 (concurrent test) 既是学术界关注的热点^[1,2], 也是 ATE 制造商^[2]、电子设计自动化 (Electronic Design Automation, EDA) 企业^[3]、SoC 生产商^[4,5] 等实际和未来

产品中不可或缺的功能.

随着器件特征尺寸的进一步降低, 功耗过高逐渐成为了 SoC 测试时不可避免的问题^[6]. SoC 的测试调度问题也从最原始的仅考虑资源约束扩展到需要同时考虑功耗约束. 功耗约束一般是指在 SoC 测试的任何时期, 其功耗都不超过事先给定的阈值. 功耗约束测试调度的研究大体可以分为三个可以互补的方向: 发掘测试机制、建立调度模型和设计求解算法.

在发掘和设计测试机制方面, 调节测试频率以控制测试功耗是一种常用的方法. 文献[4]提出一种基于时分复用 ATE 扫描频率的测试调度算法. 文献[11]中认为频率的调节粒度必须大于 30 μ s 才不会使调节切换开销严重影响最终的测试耗时. 而为了达到更细粒度

的调节,文献[5,7]通过扩展 ATE 设备的功能,提出一种异步时钟周期测试的方法,该方法针对单一电路的测试用时优化效果在 ISCAS'89 电路的仿真实验中得到了验证.其中,文献[5]描述了两种应用这一技术的方案,分别是基于硬件的内建活跃度监测器(built-in activity monitor)和基于软件的存储预模拟测试数据(stored pre-simulated test data)的方案.

在调度模型方面,较为成熟的方法是使用 MILP 模型^[8].其他的方法如使用装箱模型,将测试资源和功耗的约束抽象为测试兼容图(Test Compatibility Graph, TCG),采用基于图论的模型^[12],假设多块 SoC 芯片可以同时 ATE 上进行测试而建立的多组测试(multi-site testing)模型^[4]等.

在求解算法方面,由于测试调度的 NP 特性,使得改进传统调度算法的难度较大.结合新调度模型的特性,通过对测试的预处理,将各测试转换为调度算法更易处理的输入形式,是一种优化问题的途径.因而,学术界也有将问题求解分为多个阶段进行的方法,在测试数据预处理阶段采用了贪心算法^[4]、聚类算法^[13]、分类器(classifier)^[14]等.

本文的工作属于 SoC 测试调度,旨在通过减少 TAT 降低测试成本,围绕着测试机制、调度模型和求解算法三个方向展开.其中测试机制采用异步时钟周期机制^[7],这是一种新型 ATE 所具备的能细粒度调节各测试周期长度的功能;调度模型采用混合整型线性规划(Mixed Integer Linear Programming, MILP)模型^[8],它相比其他模型的优点在于能获得确定性的解,并且避免局部最优;求解算法上则创新性地测试数据预处理阶段,对 SoC 的各测试进行聚类^[9],通过分治的方法,不仅可以降低问题的规模,同时能更有效地利用异步时钟测试的特点.

在 ITC'02 基准 SoC 集^[10]上的实验结果表明,本文提出的基于聚类的分治方法(Clustering-Based Divide and Conquer Approach, C-BDC)能更有效地降低 SoC 测试耗时,并且在功耗约束较强时,其测试耗时与理论下界非常接近.

2 调度问题建模

在建立模型前,先就功耗约束 SoC 的测试调度问题进行必要的假设.本文所做的假设基本与文献[7,11]相同,并利用文献[5,12]中的部分模型,其中一些假设如下:(1)各测试之间的资源约束用 TCG 表示^[12];(2)使用 TSMC 180nm 的工艺,暂不考虑静态功耗和工艺偏差造成的影响;(3)采用存储预模拟测试数据^[5]的异步时钟周期测试方法;(4)供电电压不变,并且可以支持给定频率缩放范围内的所有频率.

2.1 异步时钟周期

2.1.1 符号和约束

本文中测试调度的对象是单个 SoC,调度的最小单位是该 SoC 测试集中的各个测试(例如,若 IP 核 Core-A、Core-B 分别有测试 θ_A 和 θ_B ,以及集成测试 θ_X ,显然, θ_X 与 θ_A 、 θ_B 无法并行执行,其 TCG 对应顶点间也无连线; θ_X 各周期的功耗约等于 θ_A 和 θ_B 对应周期功耗之和等,但这 3 个测试在调度中不区别对待).而异步时钟技术则对各测试中的各个测试周期进行更细粒度的优化.

我们采用文献[7]对测试属性的一些定义,并扩展到 SoC 测试调度问题中.

Θ :整个 SoC 测试集;

N : S 中包含的测试数量;

θ_i :SoC 测试集中的测试 i ; $i \in [1, N]$;

T_{θ_i} : θ_i 的测试耗时; $i \in [1, N]$;

C_i : θ_i 的测试周期数; $i \in [1, N]$;

E_i : θ_i 所释放的总能量; $i \in [1, N]$;

$L_{i,j}$: θ_i 第 j 个测试周期的长度; $j \in [1, C_i]$;

$P_{i,j}$: θ_i 第 j 个测试周期的功耗; $j \in [1, C_i]$;

$E_{i,j}$: θ_i 第 j 个测试周期释放的能量; $j \in [1, C_i]$;

$f_{i,j}$: θ_i 第 j 个测试周期频率缩放系数; $j \in [1, C_i]$;

T_{SoC} :该 SoC 完成所有测试所需用时,即最终的优化目标.且有

$$T_{\theta_i} = \sum_{j=1}^{C_i} L_{i,j}.$$

其中,测试周期是一个抽象的逻辑概念,其最短可以与 SoC/ATE 时钟频率对应的周期相同,但在这种情况下,可能导致测试调度问题规模急剧增加.反之,则会导致丢失该测试周期内的细节.以文献[11]为例,其规定在测试时钟频率 200MHz 下,400 个时钟周期为一个测试周期,那么一个测试周期时长为 2 μ s.考虑功耗约束时,假设预先设定了功耗约束为 P_{max} ,则有 $(P_{i,j} \times f_{i,j}) \leq P_{\text{max}}$.

2.1.2 下界分析

考虑功耗约束,当测试 SoC 时,每个测试周期的测试功耗均等于功耗约束时,测试耗时最少.该理论下界可以通过分析 SoC 测试时释放的总能量得出.各个测试周期的能量如下:

$$E_{i,j} = \frac{L_{i,j}}{f_{i,j}} \times (P_{i,j} \times f_{i,j}).$$

而整个 SoC 测试时所释放的能量为:

$$\sum_{i=1}^N E_i = \sum_{i=1}^N \sum_{j=1}^{C_i} E_{i,j} = \sum_{i=1}^N \sum_{j=1}^{C_i} (L_{i,j} \times P_{i,j}).$$

令理论的测试耗时下界(lower bound)记为 T_{LB} ,不考虑电压调节的情况下,由于整个测试过程释放的总能量守恒,则有:

$$T_{LB} = \frac{\sum_{i=1}^N E_i}{P_{max}}$$

在实际的测试过程中,测试耗时无法达到 T_{LB} 的原因主要有以下几点:

(1) ATE 所支持的异步时钟频率是离散的. 如文献 [7] 中所使用的 Advantest T2000GS ATE 设备仅支持 4 种时钟频率, 对应的周期分别是 2, 3, 4, 1, 5 ns.

(2) 某些低功耗模块的测试没有与其他模块的测试并行, 导致其即使运行在最高的测试频率时, 测试功耗仍然远低于功耗约束. 本文的调度算法就旨在尽可能地减少这种情况的发生.

(3) 功耗约束较弱, 使得资源约束等其他约束成为限制测试并行性的瓶颈. 此时, 功耗约束 SoC 测试调度问题退化为资源约束的调度问题, 因此由功耗约束问题推导得出的理论下界值 T_{LB} 自然失去其对问题优化的指导意义.

在后文中, 仅考虑原因 (2) 得到的实际测试用时下界, 记为 T_{Lbp} . 对 (1) 中的设备, 则当所有周期的功耗与功耗阈值之比在 $[0.4, 1]$ 范围内时能取得 T_{Lbp} . 假设此时, 该比值在 $[0.4, 1]$ 之间均匀分布, 则 $T_{Lbp} \approx 1.17 \times T_{LB}$.

2.2 MILP 模型

由于篇幅所限, 仅举例说明功耗约束 SoC 测试调度问题中的部分结构, 完整的 MILP 可参见文献 [8]. 一个 MILP 模型需要包括以下数据类型:

(1) 已知量: 如 SoC 的测试集 Θ 及其中包含的 N 个测试 $\theta_i: i \in [1, N]$;

(2) 直接变量: 如各测试的开始时间 t_{start} 、结束时间 t_{finish} ;

(3) 间接变量: 如各测试的测试时长 L_{length} ;

(4) 目标函数: 如 T_{SoC} ;

(5) 约束条件: 如描述各测试开始时间、结束时间与测试时长的关系为 $\forall \theta \in \Theta, t_{finish}(\theta) = t_{start}(\theta) + L_{length}(\theta)$; 而测试总耗时与各测试结束时间的关系为 $\forall \theta \in \Theta, T_{SoC} \geq t_{finish}(\theta)$.

2.3 分治和聚类

资源约束和功耗约束的测试调度问题都是 NP 完全问题, 采用线性规划等传统方法进行求解的时间复杂度为 $O((F)^a)^{[4]}$, 其中 F 是可供选择测试频率的数量, a 是一个 TCG 中最大连通子图的节点数^[12]. 通过分治策略降低问题的复杂度是降低求解开销的途径之一.

基于异步时钟周期的 SoC 测试调度的特殊性, 可以将 SoC 中的各个测试根据其功耗、测试频率及功耗约束分为两大类: 第 1 类测试能通过异步时钟达到 T_{Lbp} ; 第 2 类测试不能. 其区别如图 1 所示.

在图 1 的示例中, 假设 ATE 能提供的测试频率归

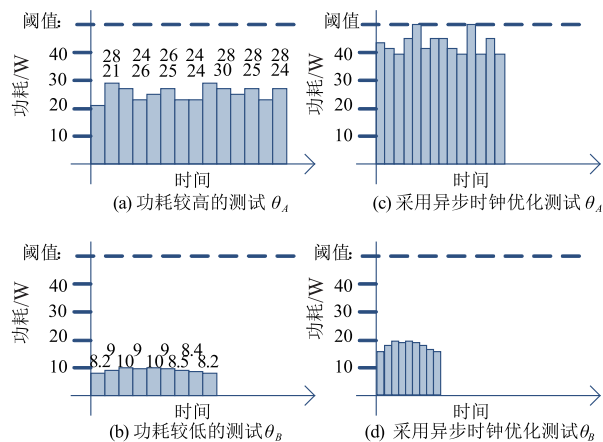


图1 高低功耗下应用异步时钟机制测试的比较

一化后的缩放系数 $f \in [1, 1.5, 2]$, 功耗约束指定为 50W. 则对图 1(a) 中的测试 θ_A , 能通过设定合适的测试频率, 使得测试功耗接近阈值功耗. 其理论测试耗时下界为 $337W / (50W/cycle) = 6.74cycle$; 而在实际的测试过程中, 对于 7 个功耗高于 25W 的测试周期, $f = 1.5$, 其余 7 个周期 $f = 2$, 其测试用时是 $7 \times (1/2 + 1/1/5) = 8.16cycle$, 与理论下界值相差 21% (如图 1(c)). 而图 1(b) 的示例中, 理论下界与实际测试用时分别是 1.43cycle 和 4.5cycle (如图 1(d) 所示), 相差 215%. 对于如 θ_B 的测试, 需要尽可能与其他测试并行执行.

图 2 给出了一个典型的测试调度示例. 在该例中, 共有 3 个测试需要调度, 分别记为 θ_A, θ_B 和 θ_C , 其测试功耗与时长组成二元组分别为 (3, 12), (1, 6), (3, 12). 异步时钟频率调节的缩放系数范围为 $[1, 4]$. 当不考虑异步时钟周期测试机制时, 若采用 3.2 节中的 MILP 模型进行求解, 其结果如图 2(a) 所示, 相比图 2(b) 所示的调度方法, 显然 2(a) 方法的 T_{SoC} 更小. 而对比两种调度方法采用异步时钟机制进行优化后的结果, 即图 2(c) 和 2(d), 可以发现, 图 2(b) 所对应的调度方法反而有更优的 T_{SoC} .

其原因是该例中的 θ_A 和 θ_C 均属于第 1 类测试, 调度问题需要解决的重点在于尽可能地使 θ_B 得到并行. 而现有的调度方法都没有针对这一情形进行优化. 通过预处理尽可能获得如图 2(b) 所示的解是本文的主要研究内容之一.

针对图 2 的例子, 通过基于功耗的分治方法 (Power consumption-Based Divide and Conquer approach, P-BDC) 可以达成该目的, 它分为以下 3 步执行: (1) 调度第 2 类测试. 获得调度结果后, 搜索未充分利用功耗约束的测试周期; (2) 遍历第 1 类测试, 寻找可并行的测试周期; (3) 调度所有剩余的第 1 类测试.

然而, P-BDC 存在以下 3 个主要问题: (1) 只能将各模块测试聚为两类; (2) 当功耗约束较弱时, 第 1 类

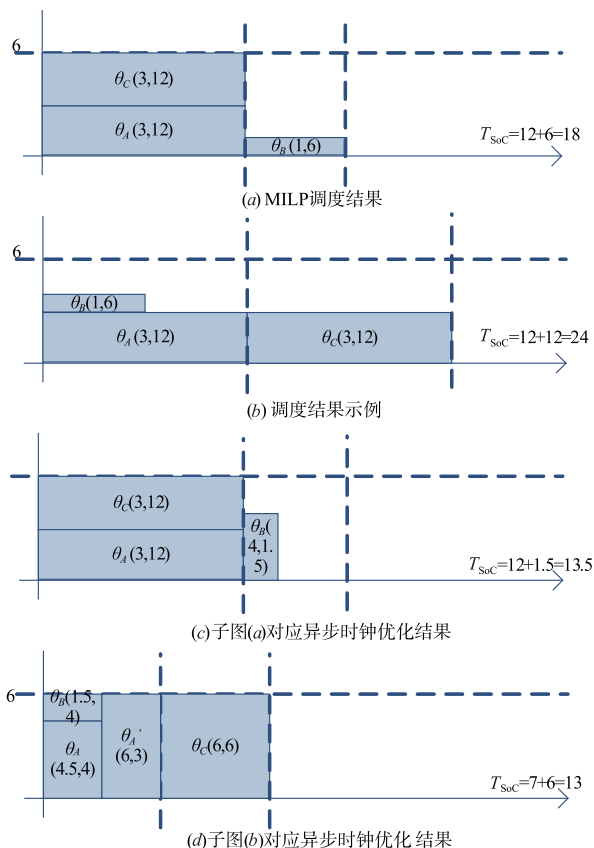


图2 两种测试调度结果应用异步时钟前后的对比

的测试数量会明显减少;(3) 分类规则仅通过平均功耗的高低,没有充分结合其他的测试属性;

聚类在识别数据的内在结构方面具有极其重要的作用^[9]. 通过聚类算法,能有效地弥补以上 3 个不足. 本文所提出的测试调度方法通过使用 K -means 聚类算法对 SoC 的各模块进行聚类,结合前面提出的分治策略,利用异步时钟机制对功耗约束的 SoC 测试调度问题进行优化. 对应本文的研究工作,聚类的各步骤简要介绍如下:

(1) 数据准备:采用 ITC'02 基准 SoC 测试集.

(2) 特征选择:IP 核数量,各 IP 核面积、功耗密度、所在层次,扫描链数量及长度,测试矢量数,输入输出管脚数,各测试的周期数、功耗等.

(3) 特征提取:与异步时钟测试密切相关的数据有测试周期数、各测试周期功耗、TCG.

(4) 聚类:采用 K -means 聚类算法实现.

(5) 聚类结果评估:本文并不具体到考查各聚类算法及其参数设置对于聚类效果的影响,而将聚类作为整体的一部分. 若应用聚类算法后,测试调度能获得更小的 T_{soc} ,则认为聚类结果有效.

综上所述,C-BDC 方法的总体流程如图 3 所示:

当 $K=1$ 时,所有的测试归为一类,则该方法等同

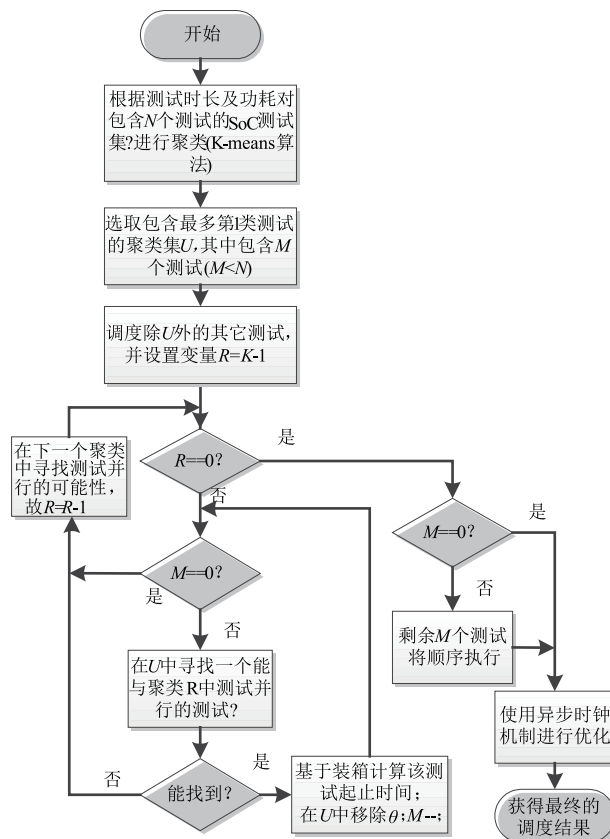


图3 基于聚类算法的异步时钟SoC测试调度流程

于基于 MILP 模型的测试调度方法. 当 $K=N$ 时,该方法约等于一种低功耗优先调度的贪心策略.

3 实验结果与分析

3.1 实验设计

基于章节 3 中的假设,与参考文献[8,11]采用相同的实验环境. 基准程序使用 ITC'02 测试集^[10]. 结合文献[15]所提供的面积、功耗信息,基准 SoC 信息如表 1 所示.

表 1 基准 SoC 信息

SoC	测试数量	最长测试周期 /cycle	功耗密度 /(W/mm^2)	峰值功耗 /W
q12710	4	5561	1.79	125.11
f2126	4	842	1.57	160.28
u226	9	3049	1.09	46.44
a586710	7	19353	1.89	87.60
h953	8	299	0.97	44.15
d281	15	164	1.37	103.75
p22810	28	256	1.70	17.93
p34392	19	1363	1.56	59.47
t512505	31	13070	1.89	48.67
p93791	32	287	1.43	39.47

当测试数量小于 10 时, K -means 聚类算法设定参数 $K=2$, 其余情况 $K=3$. 这样的参数设置是基于文献 [9] 中的结论, 对于 UCI 数据集 Iris, Wine 都是在 $K=3$ 时达到最高正确率. Iris 和 Wine 数据集都具有一定的规模, 而对于 SoC 测试调度问题, 测试的数量仍然比较少, 高的聚类数容易破坏原有的测试并行性.

本文方法中使用的 MILP 模型由 yalmip 工具^[16] 实现, 并通过 Gurobi^[17] 进行求解. 其他部分采用 C++ 编写. 异步时钟频率选取文献 [7] 的 ATE 设备特性, 分别为 [2, 3, 4, 1, 5] ns 所对应的频率, 归一化后的频率缩放系数分别是 [2.5, 1.67, 1.22, 1].

3.2 实验结果和分析

表 2 给出了不同调度方法下的 T_{SoC} 对比结果.

表 2 各方法 T_{SoC} 对比

SoC	功耗阈值 /W	T_{SoC}			T_{LB}
		MILP ^[8]	MILP ^[11] + 异步时钟	本文方法	
q12710	220	5745 (22.5%)	4794.67 (7.2%)	4449.75 (4957.12)	3784
f2126	200	875 (36.3%)	646.73 (13.9%)	557.11 (613.35)	466
u226	140	1376 (0.9%)	1369.47 (0.4%)	1364.21 (1364.21)	496
a586710	140	28497 (15.2%)	24558.15 (0.0%)	24558.15 (24558.15)	19832
h953	50	377 (37.2%)	271 (12.6%)	236.93 (275.99)	181
d281	120	174 (30.9%)	125.37 (4.1%)	120.21 (125.37)	67
p22810	30	534 (17.4%)	446.48 (1.2%)	441.20 (448.22)	308
p34392	110	1623 (23.7%)	1266.81 (2.3%)	1238 (1276.12)	1059
t512505	50	9041 (14.5%)	8963.27 (13.7%)	7733.61 (8963.27)	4706
p93791	90	323 (6.7%)	301.26 (0.0%)	301.26 (301.26)	160

表 2 中第 1 列给出各基准 SoC 的标识. 第 2 列给出了不同的功耗约束条件. 第 3 列是仅使用 MILP 模型求解获得的结果. 第 4 列是在第 3 列获得的结果基础上, 直接使用异步时钟机制进行测试耗时优化. 第 5 列括号内的数据是 P-BDC 方法得出的结果, 而加粗的数据是 C-BDC 方法的实验结果. 最后 1 列给出了耗时的理论下

限值. 第 3、4 列括号内的数据是 C-BDC 相较该结果, TAT 减少的百分比. 以 f2126 为例, 图 4 直观给出了不同调度策略在异步时钟周期机制优化后的 T_{SoC} 对比图, 图 4(a) 是第 4 列的数据, 而图 4(b) 反映 C-BDC 方法的实验结果.

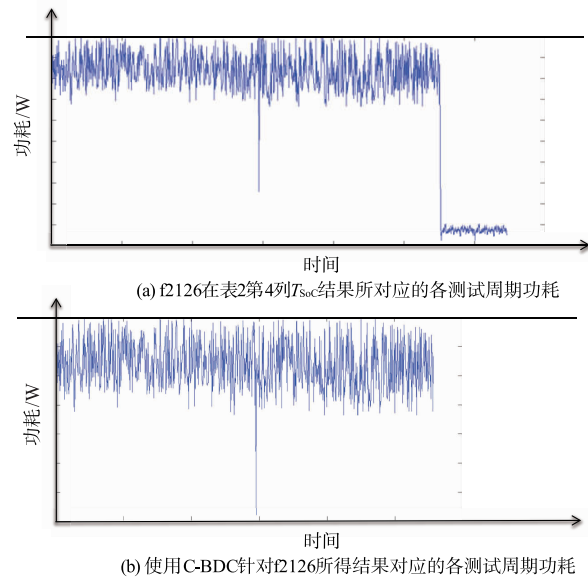


图4 f2126的TAT对比

从表 2 中第 4 列与第 3 列的比较可以看出, 异步时钟机制能有效地减少测试耗时, 优化幅度平均达到 15.89%. 此外, 第 5 列的 T_{SoC} 数据明显优于第 4 列, 其相较第 3 列的优化幅度平均达到 20.39%, 可以看出, 由于采用“分治”策略选出的部分测试在调度时不需要服从 MILP 模型给出的强约束条件, 从而能更灵活地利用功耗约束和异步时钟机制, 达到测试耗时优化的目的. 比较第 5 列的数据还可以看出, 应用聚类算法相比简单的基于功耗对测试进行区分, 在大多数情况下能更有效地对目标进行优化.

值得一提的是, 部分 SoC (如 a586710) 采用本文提出的方法并没有得到优化, 其原因主要有 2 点: (1) 该类 SoC 各模块间存在较明显的测试资源约束, 使得在图 3 所示的聚类簇 U 中, 无法找到可以并行执行的测试; (2) 其各模块的大小、复杂度、测试难度区别过大, 使得各测试的时长和功耗不在同一数量级上, 最终的调度结果也主要由周期长、功耗高的测试决定.

另外, 通过与 T_{LB} 值进行对比可以发现, 3 种调度方法都存在着进一步优化的空间. 特别的, 如 u226, 其 T_{LB} 值与实验获得的结果差距非常大, 原因在于 u226 的情形与图 1(b) 非常类似, 其长耗时 (3049 测试周期) 的测试 $\theta_0, \theta_1, \theta_2$ 均只有 2.11W 的最大功耗, 而最大测试功耗达到 113.28W 的测试 θ_6 却只有 10 个测试周期, 这一特殊的现象导致在大部分测试时间内, 测试均在较低

的功耗水平下进行。

分析原因 1 的影响,可以通过间接分析施加最高频率后仍然无法达到功耗阈值的测试周期占总周期数的比例(在后文中用 $rate$ 表示)来进行. 该比例越小,则结果越接近 T_{LBp} . 而原因 3(资源约束等)造成的影响可以通过理想实验的方式消除,如测试资源充足时,其 TCG 图是完全图. 基于以上两个假设,a586710 在不同功耗约束条件下部分实验的结果如图 5 所示.

其中,柱形图对应图 5 左侧的主纵坐标轴,表示不同功耗约束条件下, T_{SoC} 及 T_{LB} ;折线图对应右侧的次纵坐标轴,分别表示低功耗的测试周期占总测试周期的比例($rate$)及调度结果与理论下界值之间的差异($gap = (T_{SoC} - T_{LB})/T_{LB}$),各柱形上方给出了 gap 的精确数值. 从图 5 也可以看出,随着功耗阈值的提升,低功耗测试周期的比例也随之上升,但调度结果与理论下界值的差距并没有随之增大且平均为 18.0%. 结合 2.1.2 节中估算 $T_{LBp} \approx 1.17 \times T_{LB}$,则此时,调度结果与 T_{LBp} 仅相

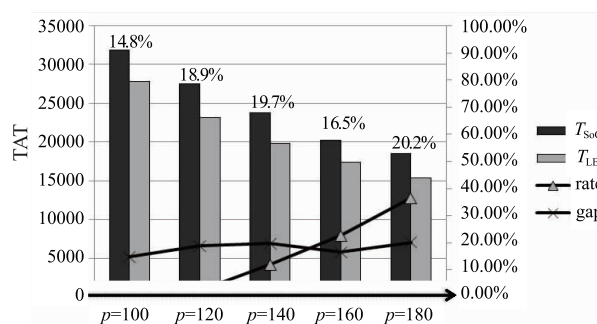
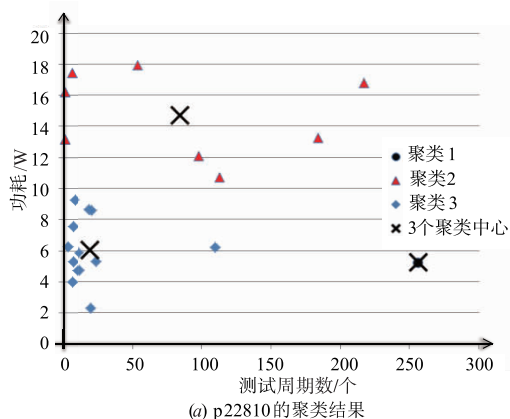


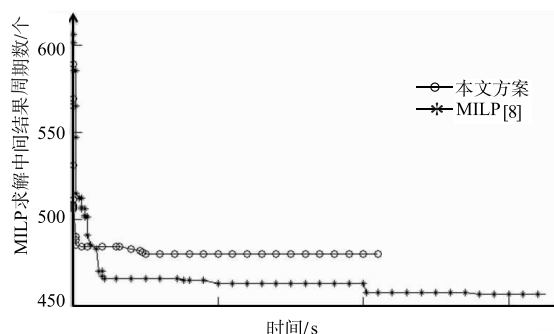
图5 a586710在两个附加假设条件下的实验结果

差 0.9%. 这在一定程度上说明了本文所提出算法的有效性几乎不受功耗约束条件的影响.

最后,以 p22810 为例,图 6 给出了聚类情形以及求解开销的对比,其中图 6(b)仅包含 MILP 模型求解的时间开销. 空间开销及其它步骤的时间开销(均在 1s 内完成)在该问题上几乎可以忽略不计.



(a) p22810的聚类结果



(b) p22810不同方法CPU计算时间开销对比

图6 聚类结果及CPU开销示例

4 结论

异步时钟机制能高效地处理 SoC 测试调度中的功耗约束问题. 而基于“分治”的思想,通过聚类对测试数据进行预处理,能更充分利用异步时钟测试调节功耗和测试时长的能力. 理论分析和实验结果表明,本文的方法相比传统的 MILP 模型方法,在降低求解时间开销的前提下,最高还能减少 TAT 13.9%,并且当功耗约束较强时,所获得的最终调度结果与实际下界值仅相差 0.9%.

参考文献

- [1] 胡瑜,韩银和,李晓维. SoC 可测试性设计与测试技术 [J]. 计算机研究与发展,2005,42(1):153-162.
Hu Yu,Han Yin-he,Li Xiao-wei. Design-for-testability and test technologies for system-on-a-chip [J]. Journal of

Computer Research and Development,2005,42(1):153-162. (in Chinese)

- [2] 裴颂伟,李兆麟,李圣龙,魏少军. 基于 V93000 的 SoC 中端口非测试复用的 ADC 和 DAC IP 核性能测试方案 [J]. 电子学报,2013,41(7):1358-1364.
Pei Song-wei, Li Zhao-lin, Li Sheng-long, Wei Shao-jun. Performance parameter testing for ADC and DAC IP cores without I/O multiplexing in SoC using Verigy 93000 [J]. Acta Electronica Sinica, 2013, 41(7): 1358-1364. (in Chinese)
- [3] Trawka M, Mrugalski G, Mukherjee N. High-speed serial embedded deterministic test for system-on-chip designs [A]. Proceedings of the 23rd IEEE Asian Test Symposium [C]. Hangzhou, China, Piscataway, NJ: IEEE, 2014: 74-80.
- [4] Vartziotis F, Kavousianos X, Chakrabarty K, et al. Time-division multiplexing for testing DVFS-based SoCs [J].

- IEEE Transactions on Computers-Aided Design of Integrated Circuits and Systems, 2015, 34(4): 668 – 681.
- [5] Shanmugasundaram P, Agrawal V D. Dynamic scan clock control for test time reduction maintaining peak power limit [A]. Proceedings of the 30th VLSI Test Symposium [C]. Maui, Hawaii, USA, Piscataway, NJ: IEEE, 2012. 248 – 253.
- [6] 邓立宝, 乔立岩, 俞洋, 彭喜元. 一种改进的层次化 SoCs 并行测试封装扫描单元[J]. 电子学报, 2012, 40(5): 949 – 954.
Deng Li-bao, Qiao Li-yan, Yu Yang, Peng Xi-yuan. A modified parallel wrapper cell for hierarchical SoCs test [J]. Acta Electronica Sinica, 2012, 40(5): 949 – 954. (in Chinese)
- [7] Venkataramani P, Agrawal V D. ATE test time reduction using asynchronous clock period [A]. Proceedings of the 2013 IEEE International Test Conference [C]. Anaheim, CA, USA, Piscataway, NJ: IEEE, 2013. 1 – 10.
- [8] Chakrabarty K. Test scheduling for core-based systems using mixed integer linear programming [J]. IEEE Transactions on Computers-Aided Design of Integrated Circuits and Systems, 2002, 19(10): 1163 – 1174.
- [9] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究 [J]. 软件学报, 2008, 19(1): 48 – 61.
Sun Ji-gui, Liu Jie, Zhao Lian-yu. Clustering algorithm research [J]. Journal of Software, 2008, 19(1): 48 – 61. (in Chinese)
- [10] Marinissen E, Iyengar E, Chakrabarty K. A set of benchmarks for modular testing of SoCs [A]. Proceedings of the 2002 IEEE International Test Conference [C]. Baltimore, MD, USA, Piscataway, NJ: IEEE, 2002. 521 – 528.
- [11] Millican S K, Saluja K K. Formulating optimal test scheduling problem with dynamic voltage and frequency scaling [A]. Proceedings of the 22nd Asian Test Symposium [C]. Yilan County, Taiwan: IEEE, 2013. 165 – 170.
- [12] Chou R M, Saluja K K, Agrawal D D. Scheduling tests for VLSI systems under power constraints [J]. IEEE Transactions on Very Large Scale Integration Systems, 1997, 5(2): 175 – 185.
- [13] Bild D R, Misra S. Temperature-aware test scheduling for multiprocessor systems-on-chip [A]. Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design [C]. San Jose, California, USA: IEEE, 2008. 59 – 66.
- [14] Lin F, Hsu CK, Cheng KT. AdaTest: An efficient statistical test framework for test escape screening [A]. Proceedings of the 2015 IEEE International Test Conference [C]. Anaheim, CA, USA: IEEE, 2015. 1 – 8.
- [15] Millican S K, Saluja K K. 3D-IC benchmarks [OL]. <http://3dsocbench.ece.wisc.edu/>, 2013-5-10/2014-06-27.
- [16] Löfberg J. YALMIP: A toolbox for modeling and optimization in MATLAB [A]. Proceedings of the 2004 IEEE CACSD Conference [C] Taipei: IEEE, 2004.
- [17] Optimization G. Gurobi Optimizer Reference Manual [OL]. <http://www.gurobi.com>, 2012-6-10/2014-06-27.

作者简介



凌立 男, 1988 年出生. 现为同济大学博士研究生. 主要研究方向为 VLSI 测试与容错计算.
E-mail: 072506@tongji.edu.cn



江建慧 (通信作者) 男, 1964 年出生. 现为同济大学教授、博士生导师、CCF 高级会员. 主要研究方向为可信系统与网络、软件可靠性工程、VLSI 测试与容错.
E-mail: jgjiang@tongji.edu.cn

张颖 男, 1984 年出生, 现为同济大学副教授. 主要研究方向为嵌入式系统设计、处理器的程序自测试.
E-mail: yingzhang@tongji.edu.cn

王真 女, 1984 年出生, 博士、高级工程师. 现为上海电力学院讲师. 主要研究方向为容错计算及计算机系统性能评估.
E-mail: wangzhenqq@hotmail.com