

基于寄存器簇恢复的追踪信号选择方法

程 云^{1),2)} 李华伟^{1),2)} 王 颖¹⁾ 李晓维^{1),2)}

¹⁾(中国科学院计算技术研究所计算机体系结构国家重点实验室 北京 100190)

²⁾(中国科学院大学 北京 100049)

摘 要 在集成电路开发中,原型芯片硅后可观测性差,使得硅后调试变得异常困难,延长了芯片的开发周期.基于追踪的可调试性设计通过在芯片中添加追踪缓存以存储关键的追踪信号,可提高芯片的可观测性.但由于调试硬件开销有限,如何选择有限的追踪信号以提高芯片可观测性成为硅后调试的研究热点.该文提出了一种基于寄存器簇恢复的追踪信号选择方法,其可分为两步:寄存器簇生成、寄存器簇融合和选择,其中寄存器簇生成根据寄存器的反馈环信息构建初始寄存器簇,而寄存器簇融合和选择可以通过簇合并优化初始寄存器簇,并选择对状态恢复率提升最大的寄存器簇作为追踪寄存器簇.当追踪寄存器簇确定后,簇输入即为追踪信号,簇内寄存器即为需获取的快照信号.在基准电路 ISCAS89 上的实验结果表明,相对于现有的追踪信号选择方法,该方法可获得更高的状态恢复率,比现有最好的选择结果平均可提高 7%,最高可提高 57%,同时算法运行时间更短,比现有最快的选择方法平均也有 54 倍的加速,而仅需增加不到 2% 的存储开销和不到 1% 的逻辑开销.

关键词 追踪调试;追踪信号选择;状态恢复率;硅后调试

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2018.02318

Flip-Flop Cluster Restoration Based Trace Signal Selection

CHENG Yun^{1),2)} LI Hua-Wei^{1),2)} WANG Ying¹⁾ LI Xiao-Wei^{1),2)}

¹⁾(State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

²⁾(University of Chinese Academy of Sciences, Beijing 100049)

Abstract The limited observability of the prototype circuit after taped out, makes post silicon debug extremely difficult and time consuming, which has extended the development cycle. Trace based debug techniques try to overcome the challenge by storing some key signals to a dedicated trace buffer on chip during debug. Due to the limited hardware budget, how to select a limited set of trace signals to improve the observability has been a hot area of research for post-silicon. Conventional trace signal selection methods can be broadly classified as probability analysis based trace signal selection and simulation based trace signal selection, both methods try to maximize the state restoration ratio. Probability analysis based selection methods only consider the structure of the combinational connection among the flip-flops, and often lead to low restoration ratio. Simulation based selection methods use the simulation results to evaluate the real restoration ratio of candidate trace signals, and lead to much longer selection time. Under the observation that certain flip-flop clusters have very limited cluster inputs and large number of flip-flops in the cluster, we propose a cluster based restoration method. It can deterministically restore the cluster elements in the all trace cycles by the tracing of the cluster inputs and the initial state of flip-flops

收稿日期:2016-09-21;在线出版日期:2017-05-31. 本课题得到国家自然科学基金(61432017)资助. 程 云,男,1988 年生,博士研究生,主要研究方向为硅后调试、硬件安全、VLSI 可测试性设计. E-mail: chengyun@ict.ac.cn. 李华伟(通信作者),女,1974 年生,博士,研究员,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为 VLSI 测试、设计验证、可靠性设计、容错计算. E-mail: lihuawei@ict.ac.cn. 王 颖,男,1985 年生,博士,助理研究员,主要研究方向为 VLSI 设计、存储优化、硬件加速设计. 李晓维,男,1964 年生,博士,研究员,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为 VLSI 测试、可靠性设计、设计验证、可信计算及无线传感器网络.

in the cluster, which can improve the state restoration ratio of the whole circuit. We further propose a trace signal selection method based on cluster restoration. The trace selection procedure is divided into two stages, the first stage is the preparation stage which is to find the original flip-flop clusters in the circuit based on the feedback loops, and the second stage is the processing stage which is to optimize the clusters and select the trace clusters. Specifically, the preparation stage can be divided into five steps: netlist parsing, logic level computation for the combinational gates, fan-in flip-flops computation, the feedback loop computation and feedback loops merging. After feedback loops merging, candidate clusters are generated and every flip-flop will belong to only one flip-flop cluster at most. The processing stage can be divided into two steps: cluster merging and cluster selection. The cluster merging merges two candidate clusters which are identified based on the merging principles into one cluster to increase the cluster weight. The cluster selection is based on the cluster weight, and the cluster with the largest cluster weight will be selected as the tracing cluster. After the cluster selection, the inputs to the tracing clusters will be selected as the trace signals, meanwhile a snapshot of the flip-flops inside the tracing clusters will also be acquired as an initial partial-state of the circuit. The experimental results on ISCAS89 show that in comparison with the best results of prior methods, our method improves the state restoration ratio by 7% at average and by 57% at the most, and can reduce the runtime by an order of magnitude. The overhead of the added snapshot buffer is less than 2% of the original trace buffer and the overhead of added logic is less than 1%.

Keywords trace based debug; trace signal selection; state restoration ratio; post-silicon debug

1 引言

随着集成电路设计复杂度的增加和快速产品化压力的增大,可调试性设计作为硅后调试的支撑技术已成为集成电路设计领域的研究热点^[1-5]. 由于设计复杂度高、软件模拟速度慢、时延模型精度低等因素制约,硅前验证已无法保证硬件设计的正确性,一些设计错误遗漏到硅后,甚至在芯片投入市场后才被发现,造成巨大损失^[1]. 为了在量产前发现并修复这些遗漏错误,需要硅后调试技术保证芯片的正确性^[2,5]. 硅后调试可验证流片后芯片的正确性,并检测、定位和诊断硅前遗漏的错误. 由于流片后芯片可观测性差,使得硅后调试成为集成电路开发流程中的重要瓶颈,在 90 纳米工艺下需要耗费高达 30% 以上的开发时间^[3-4]. 可调试性设计通过在芯片设计阶段增加辅助硅后调试的调试电路,以提高硅后调试时的芯片的可观测性,缩短硅后调试时间^[3,6].

工业界和学术界近年来研究和使用了多种可调试性设计方案:基于物理探针的可调试性设计^[7]、基于扫描的可调试性设计^[8]和基于追踪的可调试性设计^[9-20]. 基于物理探针的可调试性设计通过特定的

物理布局使得在硅后调试中可使用物理探针调试技术^[7],但物理探针调试成本昂贵,操作复杂,需要耗费巨大的设备成本和人力成本. 基于扫描的可调试性设计复用芯片中已有的扫描链,可捕获芯片在特定时刻的内部状态并通过调试端口导出到芯片外部,使得硅后调试者可获知芯片的运行状态. 此设计方法成本低,但它只能捕获一拍内部状态,无法提供实时的可观测性,调试能力有限^[8]. 基于追踪的可调试性设计通过在芯片中增加追踪缓存,可在硅后调试时提供连续多拍的实时追踪能力,并已成为硅后调试的主要技术之一,并广泛应用于 ARM[®]、IBM Power^[9]等处理器中.

追踪信号选择是基于追踪的可调试性设计的关键,选取更有助于硅后错误检测和定位的追踪信号能进一步缩短调试周期^[10]. 工业界现有的追踪信号选择仅依靠设计者的经验选取重要的功能信号作为追踪信号,严重依赖于设计者对于硅后调试的理解程度,自动化程度低,而且追踪功能信号会导致细粒度的电路级的错误定位和诊断能力不足^[6,11]. 近年来学术界提出了基于状态恢复率(State Restoration

① ARM ETM, www.arm.com/products/system-ip/core-sight-debug-trace

Ratio, SRR) 的追踪信号选择以提高整个电路的可观测性并自动化追踪信号的选择过程^[10-16]. 目前方法主要可分为两类: 基于概率的追踪信号选择^[10-12]和基于模拟的追踪信号选择^[13-16]. 基于概率的选择方法通过综合考虑寄存器之间的组合电路的拓扑结构和逻辑门的逻辑特征, 使用概率分析方法估计状态恢复率, 并迭代地选择使得状态恢复率达到最大的追踪信号. 基于模拟的选择方法使用实际的模拟数据去预估状态恢复率, 并以此选择追踪信号. 基于概率的方法运行速度快, 但概率估计精度低, 得到的状态恢复率较模拟方法低^[10]; 基于模拟的方法可取得更高的状态恢复率, 但运行时间过长, 在 ISCAS89 基准电路上也需运行多个小时^[13]. 这些不足制约了基于状态恢复率的选择方法在实际中的运用.

本文提出了一种基于寄存器簇恢复的追踪信号选择方法, 不仅可取得更高的状态恢复率, 同时可缩短追踪信号选择的时间. 首先, 本文提出了一种基于寄存器簇, 结合簇快照和簇输入追踪的状态恢复方法. 此前的状态恢复方法依靠逻辑门的逻辑推导, 仅关注局部逻辑的恢复, 而忽略了寄存器间的时序依赖关系. 本文根据寄存器之间的时序关系构建寄存器簇, 并获取簇内寄存器的快照和簇输入从而确定性的恢复此寄存器簇, 可提高状态恢复率. 其次, 本文提出了一种新的基于寄存器簇恢复的追踪信号选择方法. 具体来说, 通过分析电路的拓扑结构, 寻找反馈环, 构建初始寄存器簇, 然后根据寄存器簇间关联融合寄存器簇, 并根据寄存器簇对状态恢复率的提升效果选择寄存器簇, 从而确定追踪信号和快照信号. 基准电路 ISCAS89 的实验结果表明, 本文提出的选择方法可以提高追踪信号的状态恢复率, 平均可提高 7%, 最高可提高 57%, 其运行时间相对目前最快的选择方法平均也有 54 倍的加速.

本文第 2 节介绍基于追踪的可调试性设计、状态恢复原理和追踪信号选择、寄存器聚类等相关工作; 第 3 节描述本文基于寄存器簇的状态恢复方法、追踪信号选择的框架、相关细节和硬件设计; 第 4 节给出本文方法的实验结果; 最后一节对本文进行总结.

2 相关工作

2.1 基于追踪的可调试性设计

基于追踪的可调试性设计使用片上追踪缓存存

储追踪信号, 并使用这些追踪数据验证芯片正确性. 一个完整的追踪设计通常包括触发单元、追踪控制器、追踪缓存和数据导出单元, 如图 1 所示. 触发单元用于监控调试中的触发事件或者触发序列, 当指定的触发事件或序列发生时, 触发单元会监测到触发信息并告知追踪控制器. 追踪控制器接收到触发单元的触发信号, 开启信号追踪, 将追踪数据存储到追踪缓存中. 追踪控制器还可以根据调试需求配置触发单元中的触发事件等调试参数. 追踪缓存可实时存储追踪数据. 当追踪缓存存满追踪数据后, 导出单元会将追踪缓存中的数据通过调试接口输出到片外, 以用于后续的状态恢复和错误调试.

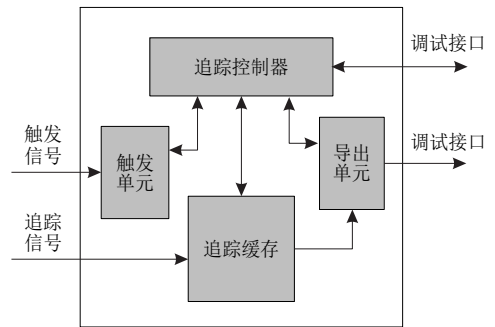


图 1 基于追踪的可调试性设计框架

2.2 状态恢复

状态恢复是利用已知的电路逻辑状态恢复未知电路逻辑状态的方法, 通过状态恢复可提高电路的可观测性^[19]. 其基本原理是利用逻辑单元的逻辑功能进行逻辑推导, 使用已知信号状态恢复未知信号状态. 通常有三种逻辑门的恢复策略: 前向恢复, 后向恢复和组合恢复. 前向恢复是利用逻辑门的输入推断逻辑门的输出, 后向恢复是利用逻辑门的输出推断逻辑门的输入, 组合恢复是结合逻辑门的输出和部分输入推断未知的输入. 这些恢复操作如图 2 所示. 对于时序逻辑门, 也可进行恢复操作, 但需考虑时序关系. 利用这些恢复原理, 可以构建状态恢复模拟器, 它以追踪缓存获取的追踪数据作为输入, 恢复未追踪的数据, 从而获知更多的门级信号状态, 有利于电路级错误的检测、定位和诊断.

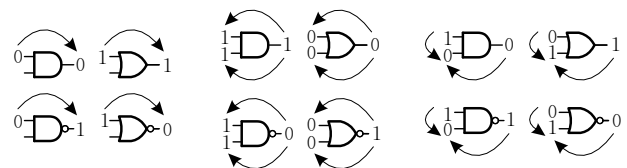
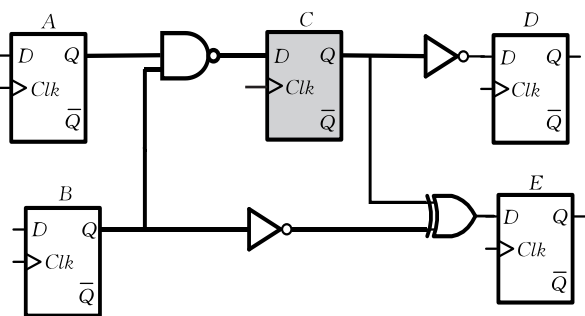


图 2 状态恢复的三种基本恢复操作

图 3 展示了一个针对实际电路的状态恢复的例子, 假设寄存器 C 是选定的追踪信号, 利用追踪缓

存获取了 C 在周期 $0 \sim 3$ 的状态值, 根据这些已知的状态按照状态恢复操作, 可以恢复出其他寄存器的一些未知状态值。



(a) 调试电路

	0	1	2	3	4
FFA	1	X	1	X	X
FFB	1	X	1	X	X
FFC	1	0	1	0	X
FFD	X	0	1	0	X
FFE	X	1	X	1	X

(b) 时序单元的数据恢复

图 3 状态恢复例图

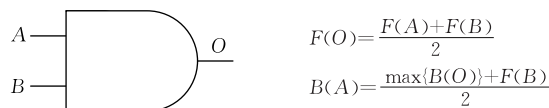
状态恢复率是评价追踪信号对未追踪信号的状态恢复能力的指标。状态恢复率定义为: 经过状态恢复后所有可知的寄存器状态的总数目与追踪获取的寄存器状态的总数目的比值。图 3 所示的状态恢复率为 $14/4=3.5$ 。在同样数目的追踪信号和同样的追踪周期下, 状态恢复率大, 就代表可以获知的内部寄存器状态更多, 内部的可观测性就更大, 更有助于电路级错误的调试。由于可在电路级清晰度量电路的可观测性, 状态恢复率成为硅后调试研究中评估追踪信号质量的最主要指标^[10-16]。

2.3 追踪信号选择

现有的追踪信号选择方法大致可分为基于概率的选择方法^[10-12]和基于模拟的选择方法^[13-16]。

基于概率的选择方法综合考虑组合电路拓扑和逻辑门的逻辑特征, 使用门级分析技术和恢复概率传播算法去估计实际的状态恢复率。它使用贪心的增量式的方法选择追踪信号, 在选择每一个追踪信号时, 需首先评估每一个候选信号作为追踪信号后全局状态恢复率的增加值, 然后选择可最大化状态恢复率的信号作为追踪信号。Ko 等人首次提出了状态恢复^[10], 并提出了基于概率分析的追踪信号选择方法。它分析每个信号的前向恢复概率和后向恢复概率, 如图 4 所示, 然后逐一选择可以获得最大状态

恢复率的信号作为追踪信号。Liu 等人提出了一种可见性的衡量指标^[11], 更细粒度的考虑信号值经过不同逻辑门的传播特性, 并以此指标选择追踪信号。Basu 等人提出了全局恢复性去选择追踪信号^[12]。这些方法只关注局部的组合电路并以此来评估全局的状态恢复, 估计精度低, 计算复杂, 得到的追踪信号集的状态恢复率较基于模拟的选择方法低。

图 4 基于概率分析的恢复率评估方法^[11]

基于模拟的选择方法使用模拟数据来评估候选信号的状态恢复率。Chatterjee 等人首先提出使用模拟的方法选择追踪信号^[13]。其文章认为借助有限周期的模拟数据可以评估出特定信号或者信号集在长周期内的平均状态恢复率。在此基础上, Chatterjee 等人提出了减量式的追踪信号选择方法, 逐渐剔除对状态恢复率贡献最小的信号, 直到追踪信号的数目等于目标追踪数目^[13]。因为基于模拟的选择方法需要多次全电路门级模拟, 而全电路门级模拟速度慢, 整个选择过程耗时很长。Rahmani 等人提出了一种使用机器学习的方法去减少模拟时间^[15], 它利用机器学习的方法从有限的仿真数据中学习并训练状态恢复率的模型, 并以此估计状态恢复率。Rahmani 等人还提出了在多个模拟数据中使用线性规划算法去选择追踪信号^[16]。Li 等人提出了使用综合评估的方法^[14], 即结合概率的方法和模拟的方法去评估追踪信号, 以减少追踪信号选择的时间。

国际上也出现一些基于特定目标的追踪信号选择和评估的方法。Prabhakar 等人提出了使用逻辑蕴含选择追踪信号, 其选择可以产生最多的新逻辑蕴含的信号作为追踪信号^[17]。Yang 等人提出了使用错误传播矩阵评估信号的错误检测能力, 并选择可观测到更多错误的信号作为追踪信号^[18]。Ma 等人提出了基于页排序的选择方法以提高断言覆盖率^[19]。以上的这些选择方法皆计算复杂, 扩展性差。

2.4 寄存器聚类

在集成电路布局领域已有一些寄存器聚类的方法。在芯片物理设计领域, 为了提高布局布线的效率, 研究者提出了寄存器聚类的方法划分寄存器簇, 将物理相近的寄存器布局到同一区域。例如文献^[20]使用 K -means 的方法将寄存器聚类以减少

时钟网络布线长度,从而节约功耗.这些聚类方法重点考虑寄存器的物理特性和电气特性.

3 基于寄存器簇恢复的追踪信号选择

本文提出一种基于寄存器簇恢复的追踪信号选择方法,利用寄存器簇快照提高追踪数据的状态恢复率,减少了选择算法的运行时间.

3.1 寄存器簇和簇状态恢复

一个寄存器簇是由电路中抽取的若干寄存器组成,这些寄存器称为簇内寄存器.簇状态表示所有簇内寄存器在当前时钟周期的状态集.而簇内寄存器的前驱寄存器称为簇输入,通过组合逻辑直接影响簇内寄存器的原始输入也被称为簇输入.在本文中,一个寄存器簇表示为 $C = (S_{FF}, S_{IN})$,其中 S_{FF} 表示组成寄存器簇 C 的簇内寄存器集, S_{IN} 表示簇输入集,对于时序电路,如果已知 S_{FF} ,根据电路拓扑结构易知 S_{IN} .图 5 所示为一个从电路中抽取的寄存器簇,它包括 5 个簇内寄存器,即 $S_{FF} = \{Q1, Q2, Q3, Q4, Q5\}$,而簇输入集 $S_{IN} = \{IN1\}$.

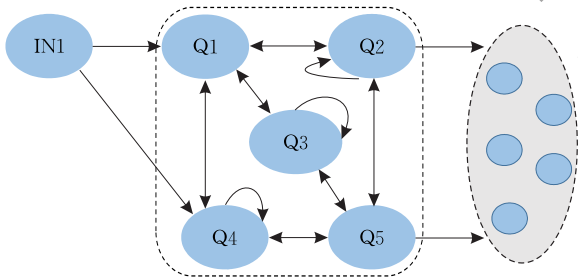


图 5 寄存器簇

易知如果可获取簇初始状态和簇输入在所有追踪周期的状态值,那么即可恢复出寄存器簇在所有追踪周期的簇状态值.因为根据簇初始状态和簇初始输入,可以得知下一追踪周期的簇状态,以此递推,可以得知此后所有追踪周期的簇状态.以图 5 为例,假设这些寄存器之间的逻辑关系如式(1)~(5)所示:

$$Q1^{n+1} = IN1^n \oplus Q2^n \oplus Q3^n \oplus Q4^n \quad (1)$$

$$Q2^{n+1} = Q1^n \oplus Q2^n \oplus Q5^n \quad (2)$$

$$Q3^{n+1} = Q1^n \oplus Q3^n \oplus Q5^n \quad (3)$$

$$Q4^{n+1} = IN1^n \oplus Q1^n \oplus Q4^n \oplus Q5^n \quad (4)$$

$$Q5^{n+1} = Q2^n \oplus Q3^n \oplus Q4^n \quad (5)$$

如表 1 所示,如果已知簇初始状态,即已知 $\{Q1, Q2, Q3, Q4, Q5\}$ 在周期 0 的状态,并且持续追踪簇输入,即已知 $IN1$ 在周期 0 到周期 4 的值,利用

这些值可以恢复出 $\{Q1, Q2, Q3, Q4, Q5\}$ 在周期 1 到 5 的状态,即可恢复出追踪周期内的所有簇状态,如表 1 中灰色阴影部分所示.如果采取传统的状态恢复方法,即不使用簇初始状态,那么选择图 5 中任何 1 个甚至任何 2 个寄存器作为追踪信号,都不能完全恢复出寄存器簇在追踪周期的所有状态.

表 1 基于寄存器初态和追踪的状态恢复示例

	0	1	2	3	4	5
IN1	1	0	1	1	1	X
Q1	1	0	0	0	1	1
Q2	1	1	0	0	1	0
Q3	1	1	0	0	1	0
Q4	1	0	1	0	0	0
Q5	1	1	0	1	0	0

这种利用簇初始状态,并结合簇输入的持续追踪,进行状态恢复的方法,称为基于簇的状态恢复方法.由于更多的寄存器状态可被恢复,基于簇的状态恢复比普通的状态恢复操作更有助于提高整个电路的状态恢复率,增加追踪调试的可观测性.本文定义了簇恢复率,代表了使用簇恢复方法时利用已知状态恢复未知状态的能力,记作 R_C . R_C 可用式(6)计算,其中 N_T 代表簇输入被追踪的周期数,同时也代表了经过簇恢复后可知的簇状态的周期数.在表 1 中,对于 5 个周期的追踪, R_C 为 3.5. 如果追踪 1000 个周期, R_C 将为 $(5 \times 1000) / (5 + 1000) + 1$, 也就是 5.97. 通过观察,本文还提出了一个粗略估计簇恢复率的指标:簇重(Cluster Weight, 简记为 CW). 簇重表示为簇内寄存器的数目与簇输入数目的比值,如式(7)所示.图 5 中的寄存器簇的簇重 CW 为 5. 易证,随着追踪周期 N_T 的无限增加, R_C 将无限接近 $(CW + 1)$, 所以可用 R_C 估计 CW .

$$R_C = \frac{|S_{FF}| \times N_T}{|S_{FF}| + |S_{IN}| \times N_T} + 1 \quad (6)$$

$$CW = \frac{|S_{FF}|}{|S_{IN}|} \quad (7)$$

3.2 基于寄存器簇恢复的追踪信号选择

基于寄存器簇恢复的追踪信号选择方法在寄存器簇生成后依据簇恢复率估计全局状态恢复率,并选择簇恢复率最大的簇作为追踪簇,簇输入作为追踪信号,簇内寄存器作为快照信号,即需捕获簇内寄存器的初始状态.如图 5 所示,追踪信号为簇输入 S_{IN} ,快照信号为簇内寄存器 S_{FF} .基于寄存器簇恢复的追踪信号选择问题可以分解为寄存器簇生成和寄存器簇选择的问题,追踪信号和快照信号由追踪寄

寄存器簇决定。由于调试设计开销有限,追踪寄存器簇同时需满足追踪信号宽度 W_T 和快照信号宽度 W_S 的约束。因此,可如下形式化基于寄存器簇恢复的追踪信号选择问题:

输入: 电路网表 G , 追踪信号宽度 W_T , 快照信号宽度 W_S

输出: 最大化簇恢复率的寄存器簇 $C = (S_{FF}, S_{IN})$, 其中 C 可由多个子寄存器簇组成, 即 $C = (C_1, C_2, \dots, C_n)$, $C_1 = (S_{FF}^1, S_{IN}^1), \dots, C_n = (S_{FF}^n, S_{IN}^n)$

约束条件: 簇输入等于追踪宽度, 子寄存器簇的簇内寄存器数目小于或等于快照宽度, 即 $|S_{IN}| = W_T, \max\{|S_{FF}^1|, |S_{FF}^2|, \dots, |S_{FF}^n|\} \leq W_S$ 。

寄存器簇生成一般可使用聚类方法, 但本文聚类的目标是优化簇恢复率, 跟簇内寄存器、簇输入和电路拓扑同时相关, 寻找最优解极为困难。现有的用于布局布线的聚类算法因应用环境和优化目标不同而并不适用。考虑到理想的寄存器簇具有簇内寄存器多而簇输入少的特性, 本文提出了一种基于反馈环生成寄存器簇的方法, 同时提出了一种簇聚合的方法去优化寄存器簇以增加簇重。

整个追踪信号选择的过程可以分为两个阶段:

(1) 寄存器簇生成阶段, 即利用反馈环信息生成初始寄存器簇; (2) 簇聚合和选择阶段, 即迭代的优化初始寄存器簇并选择追踪寄存器簇。

3.2.1 基于反馈环的寄存器簇生成

本节将详细介绍如何根据电路的反馈环信息生成寄存器簇。整个寄存器簇的生成过程如图 6 所示, 其主要思想是首先解析电路网表, 得到每个寄存器的扇入寄存器, 然后利用广度优先搜索的方法在输入锥中寻找反馈环, 合并反馈环并生成初始寄存器簇。整个过程可以分为 5 步, 详细介绍如下:

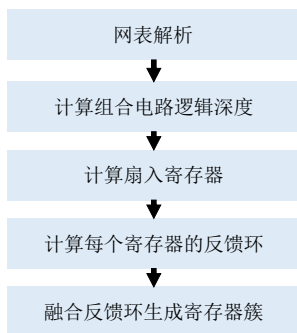


图 6 基于反馈环生成寄存器簇的流程

第 1 步: 网表解析。

首先读入设计网表, 依次计算每个逻辑门的所有的扇入和扇出, 并存储到对应的扇入和扇出链表

中, 如此可将网表转变成逻辑门之间的连接网络。本文提出的方法只关注逻辑门之间的连接关系, 而不考虑逻辑门的逻辑功能, 因此本文提出的方法可应用到包含复杂逻辑单元的设计中。

第 2 步: 计算组合电路逻辑深度。

每个组合逻辑门的输出信号的逻辑深度定义为其输入锥中的原始输入和寄存器为起始点的各条传播路径的最大逻辑深度。原始输入和寄存器的逻辑深度为 0, 每经过一个组合逻辑门, 输出的逻辑深度增加 1, 即输出的逻辑深度等于其所有输入的逻辑深度的最大值加 1。如图 7 所示, 举例说明了一个简单的组合电路逻辑深度计算的例子。

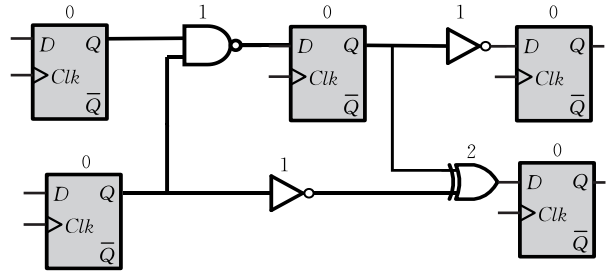


图 7 组合电路的逻辑深度计算

第 3 步: 计算扇入寄存器。

对于特定的逻辑门, 扇入寄存器指只经过组合电路直接影响此逻辑门的所有原始输入和寄存器。对于逻辑深度为 1 的逻辑门, 其扇入寄存器即为其扇入逻辑门的集合。对于逻辑深度大于 1 的逻辑门, 其扇入寄存器为其扇入逻辑门的扇入寄存器的合集。所以, 可根据逻辑深度由小到大的次序迭代计算扇入寄存器, 最终可以得到所有组合逻辑门的扇入寄存器。而寄存器的扇入寄存器则等于其扇入逻辑门的扇入寄存器的合集。

第 4 步: 计算每个寄存器的反馈环。

反馈环信息是构建寄存器簇的关键。本文观察到, 由在同一个反馈环的寄存器构成的寄存器簇往往有较高的簇重, 簇恢复率高。从电路拓扑结构上, 反馈环被定义为此寄存器的输入锥中包含此寄存器, 或者此寄存器的输出锥中包含此寄存器。本文通过搜索输入锥得到反馈环, 即使用广度优先策略遍历扇入寄存器。具体来说, 对于每一个寄存器, 首先遍历其扇入寄存器, 查找是否存在其自身, 如果存在, 就将所有在反馈路径上的寄存器组成一个反馈环并记录。然后继续迭代遍历每个扇入寄存器的扇入寄存器, 记录所有搜索到的反馈环, 直到扇入锥中的寄存器都被访问过或者对扇入寄存器的搜索到达原始输入。

第 5 步: 融合反馈环生成寄存器簇。

通过反馈环搜索可以找到每个寄存器的反馈环信息, 然后通过反馈环融合可生成初始寄存器簇。假设为每个寄存器搜索到的每个反馈环都生成一个寄存器簇, 那么某些寄存器因为存在多条反馈路径, 则该寄存器会存在于多个寄存器簇中; 此外属于同一个反馈环的寄存器会各自独立的为该反馈环生成一个反馈环簇。这些冗余的反馈环簇需要合并, 以使电路中的每个寄存器只存在于一个寄存器簇中, 便于后续的寄存器簇选择。具体的融合方法是: 对于一个特定的寄存器, 首先检查它是否存在反馈环, 如果不存在, 则可以跳过并处理下一个寄存器; 否则判断它是否属于一个已经存在的寄存器簇, 如果已经存在于某一簇中, 则可跳过并处理下一个寄存器, 如果不是, 则首先根据其簇重最大的反馈环建立一个新的寄存器簇。如果其有多个反馈环, 则需判断是否可将剩余的反馈环融合入现有的寄存器簇中, 融合条件是加入新反馈环后, 新簇的簇重有所增加, 其自身反馈环融合完毕后, 然后遍历簇内寄存器, 判断所有簇内寄存器的反馈环是否可以被继续融合到现有的寄存器簇中, 融合条件是加入簇内寄存器的反馈环后新寄存器簇的簇重有所增加。依次处理完所有的寄存器后, 可生成若干的初始寄存器簇。

3.2.2 寄存器簇融合和追踪信号选择

考虑到寄存器簇之间的连接关系, 可对寄存器簇实行簇间融合以提高单个寄存器簇的簇重。寄存器簇间连接关系有多种, 该文只考虑两种特殊的簇间关联的情况: 一是多个寄存器簇共享同样的簇输入, 二是某一寄存器簇的簇输入是另一寄存器簇的簇输出, 如图 8 所示。可以将具有以上特征的两个寄存器簇合并为一个寄存器簇, 本文称之为簇间融合。通过簇间融合, 可以增加簇内寄存器的数量, 同时不增加甚至可减少簇输入的数目, 所以相对于原始的寄存器簇, 融合后的寄存器簇的簇重有所增加, 增加了簇恢复率。

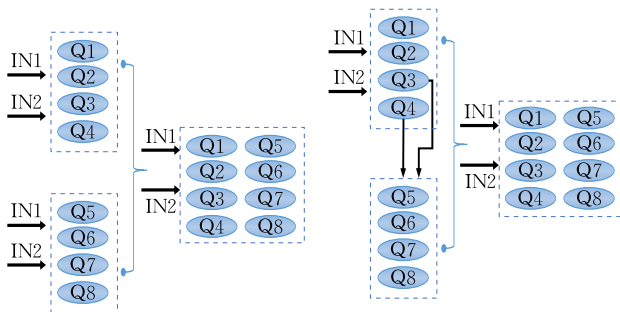


图 8 寄存器簇融合例子

寄存器簇选择将依据寄存器簇的簇重, 因为簇重可用来大致估计寄存器簇的全局状态恢复率。假设一个电路有多个寄存器簇, 那么拥有较大簇重的寄存器簇可以获得更大的簇恢复率, 可恢复出更多的簇状态, 这些被恢复的簇状态又可迭代的恢复出更多的内部状态, 也就更可能获得更高的全局状态恢复率。具体的簇选择过程为: 首先计算每一个寄存器簇的簇重, 然后对簇重进行排序, 选择具有最大簇重的寄存器簇作为目标簇, 其中簇输入将被确定为追踪信号, 而簇内寄存器将被确定为快照信号。同时需要考虑追踪信号宽度和快照信号宽度的限制, 如果此时追踪信号的总宽度超过预设的追踪宽度或者此寄存器簇的快照宽度超过预设的快照宽度, 此寄存器簇将被舍弃, 将选择剩余寄存器簇中簇重最大的寄存器簇。如果所有的寄存器簇都不满足限制要求, 将随机选择一个寄存器信号作为追踪信号。某一寄存器簇被选择后, 其簇内寄存器和簇输入将被视为已知信号, 并更新现有的候选寄存器簇, 即将这些已知信号从候选的寄存器簇中剔除。此后重复寄存器簇融合和寄存器簇选择的过程, 直至选择的各个寄存器簇的簇输入的总宽度等于追踪宽度。

3.3 算法复杂度和可扩展性

假设网表中有 N 个寄存器, M 个组合逻辑门和 E 个逻辑门间连接线, 追踪信号的宽度为 W_T 。图 6 中第 1 步~第 5 步的算法复杂度分别约为 $O(E)$, $O(E)$, $O(M)$, $O(N^2)$ 和 $O(N^2)$ 。因此寄存器簇生成的时间复杂度为 $O(N^2 + M + E)$ 。对于大型电路, M 和 E 往往是 N 的数十倍, 所以本文提出的寄存器簇生成方法的复杂度近似于 $O(N^2)$ 。假设生成的初始寄存器簇的数目为 C , 那么寄存器融合的复杂度约为 $O(C^2)$, 所以本文提出的融合和选择方法的复杂度约为 $O(C^2 \times W_T)$ 。因为 $C \ll N$ 并且 W_T 常常为 8, 16 或 32, 所以本文追踪信号选择方法的算法复杂度约为 $O(N^2)$ 。

本文提出的方法对于大电路有很好的扩展性, 尽管方法的复杂度为 $O(N^2)$ 。但实际的运行时间可以缩减, 因为基于反馈环的簇生成具有很好的局部性, 并可限制反馈搜索的深度, 减少搜索时间。对于大电路, 根据局部性和电路功能可以划分为多个子电路, 每一个子电路分开处理, 也可减少运行时间。

相对于此前的工作, 本文提出的方法可以显著的减少追踪信号选择的时间。基于概率分析的方法需要迭代的计算每个候选信号的恢复概率, 直至每个逻辑门的恢复概率趋于收敛, 所以需要大量时间计算恢复概率。基于模拟的方法需要大量的全电路

门级模拟和状态恢复去评估候选信号集的状态恢复率,而全电路门级模拟往往需要耗费大量的时间,特别是规模庞大的电路.本文提出的方法既不需要任何模拟,也不需要针对每个寄存器迭代地计算恢复概率,只需一次寄存器簇生成和数次簇选择,其中一次寄存器簇选择还可确定多个追踪信号,因此大大减少了追踪信号选择所需时间.

3.4 追踪方案的硬件设计

相对于传统追踪设计,本文提出的新追踪方案在硬件设计上有所区别.本文方案需要捕获两种不同类型的信号:追踪信号和快照信号,而传统的追踪设计不需捕获快照信号.追踪信号在追踪周期内每周期都需捕获,而对于快照信号只需捕获其初始状态.为了满足捕获要求,本文提出的追踪设计方案需要两种缓存:追踪缓存和快照缓存,如图9所示.

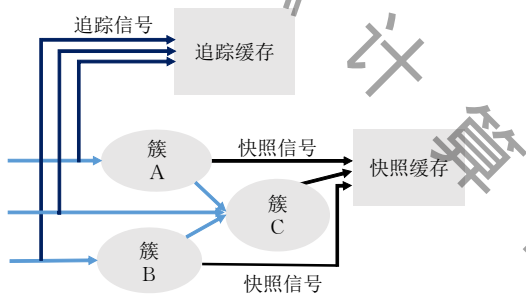


图9 追踪缓存与快照缓存

追踪缓存用来存储追踪信号,一般其宽度和深度均有限,宽度代表可以同时追踪的信号数,深度代表了可以追踪的时钟周期数.例如:16×1024的追踪缓存可以同时追踪16位追踪信号,并持续追踪1024个周期.

快照缓存用以存储快照信号.各个不同寄存器簇的快照信号可以在不同周期捕获.如果这些寄存器簇没有时序依赖关系,且其快照信号数目之和不大于快照缓存的宽度,则多个寄存器簇的快照也可同时捕获.图9中,可在三个追踪周期内先后捕获寄存器簇A、B和C的快照,如果满足快照缓存宽度约束,也可同时捕获寄存器簇A、B和C的快照.

4 实验结果

4.1 实验设置

为了评价追踪信号的质量,本文用C++实现了一个状态恢复模拟器.实验中使用Synopsys VCS生成原始的追踪数据和快照数据,由线性反馈移位

寄存器生成VCS仿真的电路输入,同时电路的某些控制性输入设置为固定的控制值.追踪缓存的追踪深度设置为4096个时钟周期,这些追踪数据和快照数据将作为状态恢复模拟器的输入以恢复未知的内部状态,以得出追踪信号的状态恢复率.

本文提出的追踪信号选择算法(记为CLUSTER)是由C++实现的,可输出追踪寄存器簇,追踪信号和快照信号.实验电路为ISCAS89中规模最大的三个电路:S38584、S38417和S35932,追踪宽度分别设为8位,16位和32位,快照宽度设为32位,并且和现有的选择方法的结果进行了比较.如第2节所述,文献[10]代表了基于概率的选择方法(记为PA),文献[16]表示基于模拟的选择方法(记为ILP),文献[14]代表了结合概率和模拟的选择方法(记为Hybrid).

在电路分析和模拟时,我们会将电路的控制性输入设置为固定值以使电路工作在正常模式.对于电路S38584,输入信号g35被视为复位信号,在VCS仿真中需要被置为1.假设不限制输入g35的取值,在仿真中则会反复出现复位值,导致大多数寄存器被重复复位,因而更易被恢复,导致虚高的状态恢复率,这与实际的应用环境不相符.除此之外,电路S35932有一个复位输入和两个模式控制输入(TM0和TM1),这些输入在仿真时都被置为1.

4.2 状态恢复率

状态恢复率是用来衡量追踪信号带来的芯片可观测性提升的主要指标.实验中PA的结果是直接文献[11]中获取的;对于Hybrid^[14]和ILP^[16],需被追踪的信号是从其公开的网站上^{①②}获取的,而最终的状态恢复率是通过本文设计的状态恢复模拟器获取的.表2的4~6列展示了最终的各种选择算法获得的状态恢复率.针对本文提出的寄存器簇恢复的选择方法展示了两种不同的恢复模式,分别为只使用追踪信号的数据进行状态恢复和同时使用追踪数据和快照数据进行状态恢复,分别标记为“簇输入”和“寄存器簇”,如7~8列所示.从表中可以看出,同时捕获追踪信号和快照信号相比现有的各种选择方法可获得更高的状态恢复率,比现有的最好结果,得到最高可达57%的提升,平均也有7%的提升.

① <http://www.cise.ufl.edu/~prabhat/>

② <http://homepages.cae.wisc.edu/~adavoodi/>

表 2 当控制输入确定时的 ISCAS89 电路的各追踪选择算法的状态恢复率

电路名	寄存器数目	追踪缓存宽度	状态恢复率					相对最好结果的提升率/%
			PA ^[10]	ILP ^[16]	Hybrid ^[14]	CLUSTER		
						簇输入	寄存器簇	
S38584	1452	8	19.0	9.2	1.4	4.7	23.5	24
		16	10.6	6.5	1.9	6.7	16.7	57
		32	6.3	8.6	12.4	5.9	14.2	15
S38417	1636	8	19.6	45.8	52.3	22.7	39.9	-24
		16	12.2	31.0	32.9	17.1	27.4	-17
		32	7.0	16.6	20.0	9.7	19.4	-5
S35932	1728	8	41.6	44.7	160.0	5.5	192.7	20
		16	39.5	12.3	114.0	3.8	103.3	-9
		32	24.8	6.14	61.6	2.6	64.3	4
平均		20.1	20.1	50.7	8.7	55.7	7	

4.3 追踪信号选择算法运行时间

本文算法运行在英特尔 4 核处理器 i5-2400 上, 处理器的频率是 3.1GHz, 内存是 4GB, 需要的运行时间如表 3 第 9 列所示. 其他追踪信号选择方法的运行时间是从其相关论文中获取的.

由表 3 可知, 基于概率的选择方法^[10]平均需

要运行数十个小时, 而基于模拟的选择方法^[13]也需数个小时. 基于概率的优化方法^[11]和基于模拟的优化方法^[16, 14]平均也需数十分钟. 本文基于寄存器簇恢复的选择时间只需要数秒. 平均而言, 本文提出的算法相对目前最快的算法有 54 倍的加速.

表 3 ISCAS89 电路上各追踪信号选择算法的运行时间

电路	寄存器数	追踪宽度	运行时间/s						相对最好结果的加速率
			PA ^[10]	Visibility ^[11]	Simulation ^[13]	ILP ^[16]	Hybrid ^[14]	CLUSTER	
S38584	1452	8	31440	162	18143	~1800	167	9.9	16X
		16	68700	369	18091	~7000	741	13.4	27X
		32	142800	1216	18003	~16200	752	21.1	35X
S38417	1636	8	28200	1179	24943	~200	434	8.2	24X
		16	69060	2647	24819	~500	2508	9.7	51X
		32	149940	5808	24734	~1200	2521	10.9	110X
S35932	1728	8	31440	1099	19857	~3000	139	1.6	87X
		16	68700	2484	19832	~5000	208	2.4	87X
		32	142800	5217	19801	~8000	217	3.8	57X
平均		81453	2242	20914	~4767	854	9.0	54X	

4.4 硬件开销

本节详细介绍了对于 S35932 电路的追踪的目标寄存器簇的详细情况和硬件开销, 实验结果如表 4 所示.

表 4 S35932 电路追踪寄存器簇的详细信息

追踪宽度	追踪簇数目	每个寄存器簇的簇内寄存器的数目
8	14	32,32,32,32,32,32,32,32,32,5,5,5,5,5,5
16	22	32,32,32,32,32,32,32,32,32,5,5,5,5,5,5,9,9,9,9,9,9
32	38	32,32,32,32,32,32,32,32,32,32,5,5,5,5,5,5,9,9,9,9,9,9,9,9,9,9,9,9,9,9,10,10,10,10,10,10,10,10,5,5

为了获得快照, 需要增加专门的快照缓存. 假设快照缓存的宽度为 32 位, 快照缓存的深度是根据寄存器簇的个数和寄存器簇的捕获方案确定的, 假设追踪缓存的深度为 4096. 当追踪缓存的宽度为 8 位时, 共需要捕获 14 个寄存器簇的快照, 那么可以设计一个 32×16 的快照缓存, 相对于原始的追踪设计, 需要增加 1.6% 的存储开销; 当追踪宽度为 16 位时, 共产生 22 个追踪寄存器簇, 那么可以使用 32×32 的快照缓存来完成快照捕获, 则需要增加 1.6% 的硬件开销. 同理, 当追踪宽度为 32 位时, 可以采用 32×64 的快照缓存, 存储开销增加 1.6%. 在这些配置中, 快照缓存都有剩余的

存储空间,同时还可以通过同时捕获多个寄存器簇,来进一步减少所需要的存储空间.总体来说,对于增加的存储开销不会超过 2%.

同时本文评估了快照信号传输网络和采集控制器的硬件开销.以表 4 中追踪宽度以 8 为例,由于需要分时捕获 14 个寄存器簇的快照信号,需要一个追踪控制器控制每个时钟周期捕获不同的寄存器簇,和 MUX 传输网络将快照信号传输到快照缓存.本文在基准电路中实现了这个控制器,带来的成本增加为 8 个时序寄存器和 650 个基本逻辑门,新增面积开销不及原电路的 1%.

4.5 ITC99 和开源电路上的实验结果

本文也在两个 ITC99 电路(b14、b15)和两个开源电路(tv80s、usb_func)上实现了本文提出的基于寄存器簇恢复的信号选择方法.这四个电路都是根据实际功能设计的.电路 b14 是 viper 处理器的一部分,电路 b15 是 80386 电路的一部分.电路 tv80s

是一个 8 位的微处理器核,usb_func 是 USB 功能核心.为了和现有方法作比较,本文根据文献[10]实现了一种基于概率的追踪信号选择方法.实验结果如表 5 所示.其中第 1 列表明电路名称,第 2 列代表寄存器数目,第 3 列代表本文提出的方法得到的追踪信号的数目和快照信号的总数目,第 4~6 列分别表示基于概率的方法^[10],只捕获簇输入,同时捕获簇输入和簇内寄存器快照得到的状态恢复率的结果,第 7 列是本文结合快照的选择方法相比概率方法的提升率.第 8~9 列分别表示基于概率的选择方法^[10]和本文方法的运行时间,第 10 列是本文方法的加速比.通过实验数据可以得到,本文提出的基于簇恢复的追踪信号选择方法在平均规模较小的基于实际功能的 ITC99 和开源电路上可以达到较高的状态恢复率,状态恢复率比基于概率的方法平均有 78%的提升,算法运行时间平均有 4.3 倍的加速.

表 5 ITC99 和开源电路实验结果

电路名	寄存器数目	追踪宽度/ 快照信号总数目	状态恢复率				运行时间/s		
			PA ^[10]	簇输入	寄存器簇	提升率/%	PA ^[10]	寄存器簇	加速比
b14	245	30/33	2.7	1.8	4.8	77.8	42.0	2.9	14.5 X
b15	449	41/172	9.2	1.8	14.0	52.2	122.0	61.4	2.0 X
tv80s	359	20/57	1.3	1.2	4.2	223.0	13.4	8.2	1.6 X
usb_func	1740	16/49	3.7	7.2	7.5	103.0	63.9	8.6	7.4 X
		32/117	3.4	4.9	6.0	76.5	164.0	15.0	10.9 X
平均			4.1	3.4	7.3	78.0	81.0	19.0	4.3 X

5 结 论

本文提出了一种基于寄存器簇恢复的状态恢复方法和一种基于簇状态恢复的追踪信号选择方案.寄存器簇恢复方法结合寄存器簇的初始状态和簇输入,可确定性的恢复整个寄存器簇,从而提高了全局的状态恢复率.追踪信号选择方法分为两步:寄存器簇生成、簇融合与簇选择,其中寄存器簇生成可通过分析电路反馈环信息获取,而簇融合可优化寄存器簇的簇重,融合后将依据簇重选择目标寄存器簇.追踪信号和快照信号都可由追踪寄存器簇确定.相对于此前的追踪信号选择方案,本文方法可以提高状态恢复率,减少选择方法的运行时间.在平均规模较大的 ISCAS89 电路上,状态恢复率相对目前最好结果最高可提升 57%,同时相对最短的选择时间,平均也可达 54 倍的加速,而仅需增加不到 2%的存储开销和不到 1%的逻辑开销.

参 考 文 献

- [1] Mitra S, Seshia S, Nicolici N. Post-silicon validation opportunities challenges and recent advances//Proceedings of the Design Automation Conference. Anaheim, USA, 2010: 12-17
- [2] Abramovici M. In-system silicon validation and debug. IEEE Design & Test of Computers, 2008, 25(3): 216-223
- [3] Abramovici M, Bradley P, Dwarakanath K, et al. A reconfigurable design-for-debug infrastructure for SoCs//Proceedings of the Design, Automation and Test in Europe. San Francisco, USA, 2006: 7-12
- [4] Josephson D. The good, the bad, and the ugly of silicon debug //Proceedings of the Conference on Design, Automation and Test in Europe. Munich, Germany, 2006: 3-6
- [5] Foutris N, Gizopoulos D, Vera X, Gonzalez A. Deconfigurable microprocessor architectures for silicon debug acceleration//Proceedings of the Annual International Symposium on Computer Architecture. Tel-Aviv, Israel, 2013: 631-642
- [6] Hsu Y, Tsai F, Jong W, Chang Y. Visibility enhancement

for silicon debug//Proceedings of the Design, automation and test in Europe, San Francisco, USA, 2006; 13-18

- [7] Lee C, Li T, Chen T. Design-for-debug routing for FIB probing//Proceedings of the Design, Automation and Test in Europe. Dresden, Germany, 2014; 1-4
- [8] Rootselaar G, Vermeulen B. Silicon debug: Scan chains alone are not enough//Proceedings of the International Test Conference. Atlantic City, USA, 1999; 892-902
- [9] Dusanapudi M, Fields S, Floyd M, et al. Debugging post-silicon fails in the IBM POWER8 bring-up lab. IBM Journal of Research and Development, 2015, 59(1): 1-10
- [10] Ko H, Nicolici N. Automated trace signals identification and state restoration for improving observability in post-silicon validation//Proceedings of the Conference on Design, Automation and Test in Europe. Munich, Germany, 2008; 1298-1303
- [11] Liu X, Xu Q. On signal selection for visibility enhancement in trace-based post-silicon validation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(8): 1263-1274
- [12] Basu K, Mishra P. RATS: Restoration-aware trace signal selection for post-silicon validation. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, 23(4): 605-613
- [13] Chatterjee D, McCarter C, Bertacco V. Simulation-based signal selection for state restoration in silicon debug//Proceedings of the International Conference on Computer-Aided Design,

San Jose, USA, 2011; 595-601

- [14] Li M, Davoodi A. A hybrid approach for fast and accurate trace signal selection for post-silicon debug. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014, 33(7): 1081-1094
- [15] Rahmani K, Mishra P, Ray S. Scalable trace signal selection using machine learning//Proceedings of the International Conference on Computer Design. Asheville, USA, 2013; 384-389
- [16] Rahmani K, Mishra P, Ray S. Efficient trace signal selection using augmentation and ILP techniques//Proceedings of the International Symposium on Quality Electronic Design. Santa Clara, USA, 2014; 148-155
- [17] Prabhakar S, Hsiao M. Using non-trivial logic implications for trace buffer-based silicon debug//Proceedings of the IEEE Asian Test Symposium. Taichung, China, 2009; 131-136
- [18] Yang J, Touba N. Automated selection of signals to observe for efficient silicon debug//Proceedings of the IEEE Very Large Scale Integration (VLSI) Test Symposium. Santa Cruz, USA, 2009; 79-84
- [19] Ma S, Pal D, Jiang R, et al. Can't see the forest for the trees: State restoration's limitations in post-silicon trace signal selection//Proceedings of the International Conference on Computer-Aided Design. Austin, USA, 2015; 1-8
- [20] Wu G, Xu Y, Wu D, et al. Flip-flop clustering by weighted K-means algorithm//Proceedings of the Design Automation Conference. Austin, USA, 2016; 1-6



CHENG Yun, born in 1988, Ph. D. candidate. His current research focus on post-silicon debug, hardware security and VLSI design for test.

LI Hua-Wei, born in 1974, Ph. D., professor, Ph. D. supervisor. Her main research interests include VLSI test,

design verification, design for reliability, and fault tolerant computing.

WANG Yin, born in 1985, Ph. D., assistant professor. His current research interests include VLSI design, memory optimization and hardware accelerator design.

LI Xiao-Wei, born in 1964, Ph. D., professor, Ph. D. supervisor. His main research interests include VLSI test, design for reliability, design verification, dependable computing and wireless sensor network.

Background

Trace signal selection for post-silicon debug is one of the most important problems in design for debug and it is challenging to use limited signal states to help post-silicon debug. The limited observability of the prototype circuit after taped out, makes post silicon debug extremely difficult and time consuming. Trace based debug techniques try to overcome the challenge by storing some key signals to a dedicated trace

buffer on chip during debug. It adds some trace buffers to record the pre-selected signals into the trace buffers, which will help the debugger to obtain more internal information and identify the root cause of silicon errors. Due to the limited hardware budget, only a small set of signals can be traced.

Conventional trace signal selection methods can be

broadly classified as probability analysis based trace signal selection and simulation based trace signal selection. Probability analysis based selection methods use some gate level probabilistic indexes to approximate the real restoration ratio and need to evaluate the probabilistic index of all candidate signals during the augmented greedy selection procedure. These methods only consider the structure of the combinational connection among the flip-flops, and often lead to low restoration ratio. Simulation based selection methods use the simulation results and state restoration to evaluate the real restoration ratio of candidate signals. Because lots of simulation efforts are needed to evaluate the candidate signals, simulation based methods usually need much long time, which makes them inapplicable to large circuits.

In this paper a novel trace signal selection method based on flip-flop cluster restoration is proposed, which can pick out trace signals with high restoration ratio at a short time. The key insight of the proposed method is to utilize the snapshot of cluster elements and the trace of cluster inputs to restore the cluster states deterministically and increase the restoration ratio for the whole circuit. Our methods can reduce the runtime for the selection by an order of magnitude compared to the prior works.

This work is supported in the National Natural Science Foundation of China under Grant No. 61432017. In the past years, our research group has published a lot of work on test generation, design for testability, silicon debug, and so on.

《计算机学报》