自动复位快速环树数据采集算法

陈志锋 李清宝 王 炜 林夕杰
(解放军信息工程大学 郑州 450002)

摘 要 获取时序可编程逻辑器件内部状态转移关系是可编程逻辑器件逆向分析的关键和安全缺陷分析的基础。在分析现有可编程逻辑器件数据采集算法的基础上,基于环树数据采集算法,提出一种自动复位机制,以实现可编程逻辑器件功能全集数据的采集;针对环树数据采集算法驱动路径动态变化的问题,提出一种动态最短路径算法,以减少状态转移次数,提高采集效率。实验结果显示,自动复位快速环树数据采集算法不仅实现了可编程逻辑器件功能全集数据的正确采集,而且数据采集效率比现有算法提高了约9%。

关键词 可编程逻辑器件,数据采集,自动复位,最短路径
中图法分类号 TP309.5 **文献标识码** A

Quick Data Collecting Algorithm Based on Ring-tree with Auto-reset

CHEN Zhi-feng LI Qing-bao WANG Wei LIN Xi-jie
(PLA Information Engineering University, Zhengzhou 450002, China)

Abstract Acquisition of inside-states-transition-functions of sequential PLD is the main point of PLD reverse analysis and the foundation of security vulnerability analysis. With the analysis of existing data collecting algorithms, an auto-reset mechanism was proposed to collect the PLD functional corpora data based on the ring-tree data collecting algorithm. For the dynamic change problems of driver paths of the ring-tree data collecting algorithm, a novel dynamic shortest path algorithm was proposed which reduces the times of state transition and improves the efficiency of data collection. Experiment results show that quick data collecting algorithm based on ring-tree with auto-reset can not only realize the collection of functional corpora data correctly, but also increase about 9% of data collection efficiency than existing algorithms.

Keywords Programmable logic device, Data collection, Auto-reset, Shortest path

1 引言

随着技术的发展和应用的普及,信息安全已经从传统的软件系统安全扩展至包括软件系统和硬件设备在内的整个电子系统,大量的研究已逐步转向对IC芯片内部的安全性漏洞和权限的分析[1-4]。常用的IC芯片安全缺陷检测方法可分为侵入式检测法[5,6]、非侵入式检测法[7-10]。对于可编程逻辑器件(Programmable Logic Device, PLD)的安全缺陷检测而言,侵入式检测法需要如开封、制样、拍照、电子探针、扫描电镜等硬件设备,费用代价高,而且检测过程需要消耗很长时间(解析一个芯片需要耗费多个月时间);由于植入的安全缺陷通常很小,最小仅需两个门即可实现,而随着集成电路精度的提升,集成电路中包含成千上万亿个门[22],用这种方法"寻找"安全缺陷的难度很大且局限性明显,显然不适用于PLD安全缺陷检测。非侵入式检测法可分为运行时和测试时两种,运行时技术采用一个在线的监测系统监测运行过程中的异常行为,测试时技术一般用于产品部署前的安全缺陷监测。非侵入式检测法相对于侵入式检测法代价降低了,检测过程需要一个不包含安全缺陷的Golden Chip,但很难保证一个芯片是Golden Chip的,即使检测到存在安全缺陷也无法精确定位到哪个芯片上存在安全缺陷,因此该方法不适于PLD内部结构,无法实现缺陷的准确定位。基于状态转移图的PLD安全缺陷检测方法[11]在获取PLD逻辑数据的基础上,绘制状态转移图,分析状态间的转移关系,通过与硬件环境结合检测安全缺陷,不仅能够快速准确地检测出PLD中存在的安全缺陷,而且大大降低了成本。

数据采集是基于状态转移图安全缺陷检测方法的工作前提。为确保PLD安全缺陷分析的正确性,数据采集要提供PLD内部的工作全集数据,即在所有可能输入条件下的组合输出、寄存器状态、高阻输出等数据信息。

本文简要分析比较了几种典型的PLD数据采集算法,指出了数据采集算法存在的问题和不足。结合PLD状态转移随机性特点和安全缺陷检测要求,针对PLD功能全集数据不完全的问题,提出自动复位快速环树数据采集算法,实现了

到稿日期:2011-12-14 返修日期:2012-03-17 本文受国家高技术研究发展(计划)(863)项目(2009AA01Z434)资助。
陈志锋(1986—),男,硕士生,主要研究方向为信息安全,E-mail:xiaohouzi06@163.com;李清宝(1967—),男,博士,教授,主要研究方向为信息安全;王 炜(1975—),男,博士,讲师,主要研究方向为计算机系统结构、片上多处理器与片上系统、信息安全;林夕杰(1987—),男,硕士生,主要研究方向为信息安全。

PLD功能全集数据的采集。实验结果与实际应用均显示，自动复位快速数据采集算法解决了功能全集数据不完全的问题，提高了采集算法的效率。

2 数据采集算法分析

2.1 现有数据采集算法

数据采集的过程可以等价于完备输入空间的构建过程。在对组合逻辑电路进行数据采集时，由于外部激励可控，施加激励全集即可构建完备输入空间，因而若无特殊说明，下文所述数据采集均针对时序逻辑电路。对于时序逻辑电路，仅外部输入与外部输出可见，状态转移不可控，如何构建有效的状态空间是时序逻辑电路数据采集的关键。目前主要有4种数据采集算法：①基于环的数据采集算法；②基于树的数据采集算法；③分状态数据采集算法；④基于环树的数据采集算法。

基于环的数据采集算法[13]主要是基于保存环的思想，发现有环路时把环中所有状态规约为一个新的状态节点。在算法中，为每个状态设置编码值，编码值按照状态出现的次序编号，新状态的编码值为当前最大值。算法结束后，所有状态的编码均为初始值。由采集过程中的有效状态集以及记录的边构成的状态图称为环图。故该算法称为基于环的数据采集算法。这种方法在最坏情况下找到环的数目最多为 $N-1$（N为状态数），每个环内仅有2个节点（包含被视为节点的环），因此该算法最坏情况下记录 $2N-2$ 条边，$N=|SA|$。

基于树的采集算法[14]采用树形结构记录状态转移关系，该树是一个多叉树。算法在搜索未加全状态时，按照树的结构从根节点出发，一步一步地搜索其子树。只要数据采集过程中存在未加全状态，算法必能在有限步搜索中寻得。该算法不记录重复的边且不记录自身到自身的转移关系，在边的数目上达到了最少。因此，基于树的采集算法，仅需记录 $N-1$ 条边，$N=|SA|$。

文献[15]提出一种基于内向树的采集算法，该算法思想上与基于树的采集算法类似，主要的区别在于结构上的差异。基于内向树的采集算法采用内向树记录状态转移关系。一般的树通常称为外向树，表现为根向各个结点发散，根到任意结点可达；内向树则表现为任意结点到根结点可达，方向与外向树相反。该算法也仅需记录 $N-1$ 条边，$N=|SA|$。

文献[16]提出了一种分状态数据采集算法，它亦可称为状态逐次加全算法。该算法以复位态（根结点）为初始状态 s_0，逐次加全所有状态。在采集过程中，算法依托状态转移图维护一个路径信息表，记录未加全状态至根结点 s_0 的路径，用于实现从根结点至状态 s 的转移。若某次状态转移中次态不在路径表中，则记录转移信息，根结点 s_0 至次态的路径即由 s_0 至现态的路径与该激励构成。该算法也不记录重复的边和自身到自身的转移关系，因此该算法仅需记录 $N-1$ 条边，$N=|SA|$。

该算法主要包含环树结构的创建和层次驱动策略两部分。环树结构通过环内和环间结构的调整，在确保规模为 $O(N)$ 的前提下缩短了平均路径长度；层次驱动算法充分利用了树形结构的优点，通过对子环树出口的调整，构建驱动路径，使得无需搜索未加全状态，就可以直接按环边和出树边进行状态驱动，减少了驱动尝试次数，提高了算法的效率。

基于环树的数据采集算法[17]结合了基于环的数据采集算法和基于树的数据采集算法的优点。如图1所示，环树是一种不包含自圈和重边的有向图，兼有环与树的特点，与内向树相似，区别在于环树中内节点的孩子构成一个有向圈。

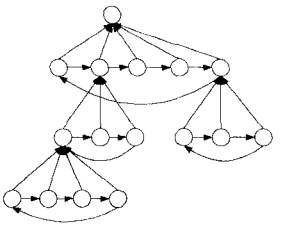

图1 环树结构

数据采集算法的时间包括驱动路径时间和状态加全时间。状态加全与状态个数和输入引脚个数相关，上述4种算法的状态加全时间均为 $O(MN)$，其中 $M=|I_A|$，$N=|S_A|$。因此决定采集算法的时间复杂度的关键是驱动路径时间，即搜索从加全状态到未加全状态所需的时间。最坏情况下，每次搜索到的驱动路径都是最长的，即从初始状态到未加全状态的路径为经过所有加全状态后转入未加全状态的状态数目，则驱动路径时间复杂度为：

$$T_{max}=(M-1)(1+2+\cdots+N-1)=\frac{(M-1)(N-1)N}{2}$$
$$=O(MN^2)$$

式中，$M=|I_A|$，$N=|S_A|$。

该结论对上述4种算法均适用，故4种算法的最坏时间复杂度为 $O(MN^2)$。

此外，基于树和内向树的数据采集算法以及状态逐次加全算法都需借助于复位状态，若电路中不存在复位引脚，则必须通过上电复位，但频繁的复位电路易对芯片造成损害。

2.2 现有数据采集算法的局限性

现有数据采集算法虽然能够实现大部分PLD的数据采集，但是在某类PLD的数据采集和最短驱动路径的选取上存在局限性。

1. 功能全集数据不完全问题

通常情况下，设计者在PLD中设计的状态机对应的状态转移图是强连通的，但在某些情况下，设计者基于某种目的而不按照常规的设计方法设计状态机。例如，根据文献[11]对PLD安全缺陷的存在形式的分析，存在孤立状态（环）存在形式，即某一状态 S 在输入激励下跳转到另一状态 $S0$ 或者由若干个状态组成的一个环 SC，却不存在从 $S0$ 和 SC 中跳转到其它状态的转移关系，这种情况在一般的状态机设计中是不可能出现的，但可被用来设计安全缺陷。

如果采用上述4种采集算法采集含孤立状态安全缺陷的PLD，将导致采集数据不完全，状态转移图不完整，从而影响孤立状态安全缺陷分析的准确性。

2. 最短驱动路径动态变化问题

在数据采集的过程中，从加全状态寻找一条到未加全状态的路径是数据采集的关键之一。由于存在多条这样的路径选择，因此选取一条最短的驱动路径是解决该问题的重点，最短驱动路径算法的优劣直接影响着数据采集的效率。

上述4种数据采集算法在最短驱动路径算法选取上主要采用Dijkstm算法[18]、Floyd算法[19]、A* 算法[20,21]等。但由于在数据采集的过程中状态图是动态变化的，在状态加全的

过程中，随着时间的推移，更多的状态从未加全状态转变为加全状态，使得加全状态到未加全状态的最短路径随之变化，同时新的未加全状态的加入也使得最短路径产生变化。因此，上述4种数据采集算法采用的最短驱动路径算法计算得到的并不是真正的最短路径，这影响着采集算法的效率，故纯粹采用上述最短路径算法不能很好地解决最短驱动路径动态变化的问题。

3 自动复位快速环树数据采集算法

数据采集算法因进入孤立状态后无法实现状态的转移使得采集的数据不完全，并且因状态的动态变化和转移关系的随机性使得采集算法在最短驱动路径的选择上存在不足。针对这些问题和不足，本文提出自动复位快速采集算法（见3.1节）来解决功能全集数据的不完全问题，提出动态最短驱动路径算法（见3.2节）来解决最短驱动路径的选取问题，其中3.2节算法是3.1节算法的子算法，是3.1节算法实现的一部分。

3.1 自动复位环树数据采集算法

对于含有孤立状态安全缺陷的PLD，数据采集算法无法采集功能全集数据，导致采集的状态不全或者转移关系缺失。因此本节提出自动复位环树数据采集算法，其利用芯片自动复位功能解决功能全集数据采集的不完全问题。下面从两个方面讨论自动复位环树数据采集算法。

(1)算法结构

自动复位环树数据采集算法中涉及的结构主要有表头节点、表内节点和搜索队列，以及一个临时结构。表头结点数据结构如图2所示。

状态节点 Q	输入向量 I	指针 P	指针 R
路径存在标志	已搜索标志	路径长度	加全标志

图2 表头节点结构图

输入向量 I：当前状态 Q 下所需要施加的激励；路径存在标志 F：表示该状态的最短路径信息是否有效，$F=1$ 为有效，$F=0$ 为无效；已搜索标志 H：为1表示该状态已搜索过；指针 R：当最短路径标志 F 有效时，指针 R 指向其最短路径队列的队头；路径长度 L：最短路径上当前状态至未加全状态的距离。

表内结点数据结构如图3所示。

状态节点 Q	输入向量 I′	指针 P	指针 R
加全状态	孤立标志	路径存在标志	
路径长度	已搜索标志	孤立标志为1跳转状态 Q′	

图3 表内节点结构图

I'：从表头结点转移到表内结点所施加的激励；其余同上。

搜索队列数据结构如图4所示。

指针 P	层次 C	输入向量 I	指针 P1	指针 P2

图4 搜索队列结构图

临时结构记录转移至孤立状态的前一个状态和激励，如图5所示。

状态 Q	前一状态 Q′	输入向量 I

图5 临时结构图

(2)算法描述

从上电状态开始，当搜索到状态 Q 时，如果状态 Q 不在表头数组中，则将其加入表头数组，开始对其进行输入加全；否则，从上次输入加全的终止处开始顺序施加输入激励，如果在施加激励1时发生状态转移至次态 Q'，则检测状态 Q' 是否已经存在于 Q 的邻接表中，若存在，则只需将激励1加入 Q' 节点的输入向量集合中，否则，开辟新的状态节点空间，将状态 Q' 插入 Q 的邻接表中，如果 Q' 是新出现的状态，还需要将它加入到表头数组中。当出现孤立状态时，对孤立状态进行标志，并自动复位芯片至初始状态。当时序机中所有出现的状态均已成为加全状态之后，状态网络就自动生成了，并且在状态网络动态生成的任何阶段，利用已有的局部状态网络都可为状态驱动提供支持。

算法描述如图6所示。

算法1 自动复位环树数据采集算法

输入：芯片各引脚属性（如输入、输出、时钟等）

输出：芯片的逻辑数据

1. 初始化数据，置所有状态的当前激励值均为0，设芯片上电状态为初始状态 S_0。
2. 令当前状态 $S=S_0$。
3. 若上电后芯片未复位过，转4；否则转5。
4. 若现态 Q 为未加全状态，取 Q 的当前激励并加1，转6；否则，调用动态最短驱动路径算法，确定下一状态，若该状态为未加全状态，以该未加全状态为现态，转3；否则寻找进入该状态的上一个状态 Q″ 和激励 I′，自动复位芯片，转2。
5. 取复位前芯片对应的状态 Q″ 和激励 I′，若 Q″=Q 且 I′=I，，则激励 I 加1，转6；否则转4。
6. 对现态 Q 施加激励和时钟，若状态 Q 发生转移至 Q′，转7；否则，转8。
7. 若 Q′ 已在 Q 的邻接表中，则 Q−>Q′，为重边，丢弃，将当前输入激励加入节点 Q′ 的输入激励数组域内；否则，分配新的节点空间将 Q 插入 Q 的邻接表中，记录边 Q−>Q′，如果 Q′ 为新出现的状态，则将它加入到表头数组中，转6。
8. 若现态当前激励值为全1，置现态为加全状态，转10；否则，转9。
9. 以次态为现态，转4。
10. 判断是否所有出现的状态均已加全，若是，结束；否则，以次态为现态，转3。

图6 自动复位环树数据采集算法

3.2 动态最短驱动路径算法

在数据采集过程中，由于状态网络随着新状态的加入、状态的加全、边的增加等使得状态网络在不断更新，因此状态间的最短路径也在不断变化。为此需要对最短路径算法进行修改以适应这种情况。

数据采集过程中，在某个状态成为加全状态后，状态转移图发生变化，用 Dijkstm 算法寻找该状态经过其余加全状态（若直接连接到未加全，可以不经过加全状态）到所有可达未加全状态的最短路径，从所有最短路径中选取一条相对的最短路径，进而驱动电路按照该路径转移到未加全目标状态。所得到的相对最短路径必为最短路径，这是由 Dijkstm 算法保证的。

每当状态转移图发生上述情况，均按照该方法计算最短驱动路径，这是一个动态寻找最短路径的过程，因此称该算法

为动态最短驱动路径算法。算法实现过程中要充分利用现有状态网络信息，仅计算当前加全状态到未加全状态的路径，减少路径计算量，并且在相对最短路径选取过程中，边计算边比较，相对于得到所有最短路径后再进行比较可以减少时间开销。动态最短路径快速驱动算法描述如图 7 所示。

算法 2　动态最短驱动路径算法

输入：局部状态转移图

输出：最短路径

1. 以当前加全状态为源点，从状态转移图中找出所有的未加全状态，并以这些未加全状态为目标节点；
2. 利用现有状态网络信息计算源点到目标节点之一的最短路径 L_i；
3. 重复 2，得到路径 L_j，比较 L_i 与 L_j，将较小值赋予相对最短路径 L_{min}；
4. 重复 3 直到源点到所有目标节点的最短路径计算完毕，此时得到最终相对最短路径 L_{min}；
5. 按照该相对最短路径 L_{min} 驱动芯片进入未加全状态，继续数据采集；
6. 每遇到有状态加全，动态调用算法步骤 1－5，直到所有状态均成为加全状态，算法结束。

图 7　动态最短驱动路径算法

3.3　算法分析

算法 1 和算法 2 分别针对数据的不完全问题和驱动路径选择问题提出了解决的方法。算法 1 利用芯片自动复位实现了功能全集数据的采集，算法 2 提出了动态计算加全状态到未加全状态的最短路径的方法来提高驱动效率，解决最短驱动路径选择问题。

算法 2 是算法 1 的组成。算法 2 动态计算最短路径，每一次得到的都是当前状态转移图中加全状态到未加全状态的最短路径，这是对数据采集算法最短驱动路径选择的改进。尽管该驱动路径算法的最坏时间复杂度仍与原来的驱动路径算法一致，为 $O(N^2)$，但是系数得到了减小，从而减少了路径驱动次数，间接提高了采集算法的效率。

算法 1 采用的自动复位环树结构与环树结构类似，故其最坏时间复杂度与基于环树的数据采集算法一样，为 $O(MN^2)$。但是该算法解决了功能全集数据不完全问题，且采集过程中不需要频繁地复位，仅在发现无法驱动芯片进一步数据采集时才进行复位，这比基于树和分状态的数据采集算法减少了对芯片的损坏。

4　实验及结果分析

1. 实验环境

实验硬件环境为实验室自制的脱机式逆向分析平台，软件放置在配置为 AMD Athlon 64 X2 Dual Core Processor 4200＋ 2.21Ghz、1G 主存的 PC 机上，软件包括驱动程序、原数据采集程序和自动复位快速环树数据采集程序。驱动程序实现软件与硬件的交互，实现命令控制、数据交换等功能。

2. 实验与分析

(1)算法正确性验证

该实验主要是测试自动复位快速环树数据采集算法的正确性。测试对象为 XX 卫星接收机上的 PLD 以及特定功能的时序 PLD，从两种角度分别检验算法的正确性。

利用工具将卫星接收机上的 PLD 从接收机上取下来放置到脱机式逆行分析平台上，进行数据采集，逻辑综合，经编译后将结果通过编程器烧录到芯片，并替换原芯片，将替换后的芯片放置在该设备上，启动设备并观察其是否正常工作，如果其连续工作时间超过一定时间，则说明替换的芯片是正确的，即自动复位快速采集算法是正确的。实验结果如表 1 所列。

表 1　自动复位快速环树数据采集算法正确性测试结果

设备名	所在位置	编号	型号	输入输出规模				是否成功
				I	C	Q	CLK	
XX 卫星接收机	接口板	U34	PALCE16V8H	14	3	0	0	是
		U35	PALCE20V8H	14	8	0	0	是
		U36	PALCE20V8H	13	5	0	0	是
		U116	PALCE16V8H	4	8	0	0	是
	解调板	U73	TIBPAL20R4	12	3	4	1	是
		U86	TIBPAL16R8	0	0	8	1	是
		U91	PALCE16V8H	7	7	0	0	是

表 1 列出了卫星接收机不同功能不同规模的 PLD 的名称、型号等信息以及采集算法是否能实现数据的正确采集，I 为输入引脚数，C 为组合输出引脚数，Q 为寄存器输出引脚数，CKL 为时钟引脚数。

实验结果表明，对于该卫星接收机上的 PLD，数据采集算法与原数据采集算法一样实现了 PLD 功能全集的正确采集，这表明自动复位快速环树数据采集算法实现了数据采集算法的功能，是正确的。

对于特定功能的不同规模不同逻辑的 PLD，总共设计了 6 类不同逻辑的 PLD，并且对于同一种型号的 PLD 设计了两种逻辑功能，一种设计为正常功能，另一种设计为含孤立状态安全缺陷，分别用原数据采集算法和自动复位快速环树数据采集算法采集这 6 类 PLD。编译自行设计的程序，将其通过编程器烧录到空白芯片中，同样将烧录好的芯片放置到脱机式逆向分析平台，利用采集算法采集数据，经逻辑综合后比较分析的结果与原程序是否等价，如果等价，则表示采集算法正确。实验结果如表 2 所列。

表 2　原采集程序与自动复位快速环树采集程序正确性测试结果

芯片型号	输入输出规模				原采集算法	自动复位快速环树采集算法
	I	C	Q	CLK		
GAL16V8(正常)	8	3	4	1	成功	成功
GAL16V8(含孤立状态)	4	2	5	1	失败	成功
GAL22V10(正常)	11	6	2	1	成功	成功
GAL22V10(含孤立状态)	8	0	10	1	失败	成功
EPM7032(正常)	16	0	16	2	成功	成功
EPM7032(含孤立状态)	10	2	10	1	失败	成功

从表 2 可知，对于 GAL16v8(正常)、GAL22V10(正常)和 EPM7032(正常) 3 种芯片，原数据采集算法和自动复位快速环树数据采集算法均能够实现数据的正确采集；但是对于 GAL16v8(含孤立状态)、GAL22V10(含孤立状态)和 EPM7032(含孤立状态)这 3 种芯片，通过结果的比较，原数据采集算法采集的数据逻辑综合后的结果与设计的程序不一致，说明原采集算法采集的数据是不正确的，而自动复位快速环树数据采集算法采集的数据经逻辑综合后得到的结果与原程序等价，即自动复位快速环树数据采集算法是正确的。实验结果表明现有数据采集算法无法完成含孤立状态安全缺陷 PLD 的数据采集，这与数据采集算法存在问题的分析是一致的；而自动复位快速环树数据采集算法解决了这一问题，实现

了含孤立状态 PLD 功能全集数据的采集。

(2)算法性能对比

该实验测试基于环树的数据采集算法与自动复位快速环树数据采集算法的性能，主要是指时间性能。实验对象为特定功能的 PLD，为表 2 列出的 3 种正常功能的 PLD。通过采集正常功能 PLD 的数据，记录采集过程所耗费的时间，实验结果如表 3 所列。

表 3　原采集算法与改进的采集算法采集耗时结果

芯片型号	输入输出规模				基于环树采集算法	自动复位快速环树采集算法	百分比
	I	C	Q	CLK			
GAL16V8（正常）	8	3	4	1	14350ms	13082ms	8.84%
GAL22V10（正常）	11	6	2	1	32580ms	29685ms	8.89%
EPM7032（正常）	16	0	16	2	6325704ms	5731362ms	9.40%

表 3 列出了 3 种不同规模的 PLD 不同的采集算法所需要的时间和百分比。根据表 3 的时间对比可以看出，自动复位快速环树数据采集算法消耗的时间比基于环树的采集算法低，并且提高的百分比约为 9%。从时间差值来看，对于小规模的 PLD，差值较小，对于较大规模的 PLD，则差值较大，这表明对于较大规模 PLD 的数据采集可以节省比较可观的时间。但是时间上的优化是以增加辅助的节点记录为代价的。由于数据采集算法效率的提高可以加快芯片安全缺陷的检测，因此这种代价是值得的。

结束语　本文针对现有数据采集算法存在的问题提出自动复位快速环树数据采集算法。实验结果表明，自动复位快速环树数据采集算法与现有数据采集算法一样实现了功能全集数据的采集。通过自动复位机制解决了现有数据采集算法全集功能数据采集不完全的问题。

通过引入动态最短驱动路径算法，提高了自动复位快速环树数据采集算法的效率，这对提高基于状态转移图的安全缺陷检测方法的效率是非常有利的。

该算法已成功应用于国家“863”项目，对提高采集算法和安全缺陷检测效率发挥了重要作用。

目前该算法主要用于中小规模 PLD(20 引脚至 44 引脚)的数据采集，对于大规模 PLD 因采集硬件环境受限还未能实现，这是下一步需要解决的问题。

参 考 文 献

[1] Chakraborty R S, Paul S, Bhunia S. On-demand transparency for improving hardware Trojan detectability[A]// Proceedings of IEEE International Workshop on Hardware-Oriented Security and Trust[C]. 2008:48-50

[2] Stefan D, Mitchell C, Almenar C G. Trojan Attacks for Compromising Cryptographic Security in FPGA Encryption Systems [EB/OL]. http://199.98.20.129/~stefan/projects/csaw08/csaw08_cooper_submission/cooper_csaw08.pdf, 2008-09-27

[3] Wolff F, Papachristou C, Bhunia S, et al. Towards trojan-free trusted ics: problem analysis and detection scheme[A]// Proceedings of Design Automation and Test in Europe[C]. 2008: 1362-1365

[4] Adamov A, Saprykin A, Melnik D, et al. The Problem of Hardware Trojans Detection in System-on-Chip[A]// Proceedings of CADSM'2009[C]. Polyana-Svalyava UKRAINE, 2009:178-179

[5] Kash J A, Tsang J C, Knebel D R. Method and Apparatus for Reverse Engineering Integrated Circuits by Monitoring Optical Emission[P]. United States Patent Number 6. 2002, 022 B1:496

[6] Chipworks Inc. Semiconductor Manufacturing-Reverse Engineering of Semiconductor components, parts and process[EB/OL]. http://www.chipworks.com

[7] Abramovici M, Bradley P. Integrated Circuit Security: New Threats and Solutions[A]// Proceedings of 5th Ann. Workshop Cyber Security and Information Intelligence Research: Cyber Security and Information Challenges and Strategies[C]. ACM Press, 2009:55

[8] Chakraborty R S, et al. MERO: A Statistical Approach for Hardware Trojan Detection[A]// Proceedings of Workshop on Cryptographic Hardware and Embedded Systems[C]. LNCS 5747, 2009:396-410

[9] Agarwal D, Baktir S, Karakoyunlu D, et al. Trojan detection using IC fingerprinting[C]// IEEE Symp. on Security and Privacy. 2007:20-23

[10] Banga M, Chandrasekhar M, Fang Lei, et al. Guided test generation for isolation and detection of embedded trojans in ICs[A]// Proceedings of the 18th ACM Great Lakes Symposium on VLSI [C]. 2008:363-366

[11] 陈志锋，李清宝，曾光裕. 时序 PLD 安全缺陷检测方法研究[J]. 计算机科学，2012，39(5)：53-56，79

[12] Zeng Guang-yu, Fan Min, Li Qing-bao, et al. Research on the Algorithm of Decrypting Encrypted Synchronous Sequential Machine[A]// 2008 International Symposium on Computer Science and Computational Technology[C]. Shanghai, China: IEEE, 2008:599-602

[13] 徐仑峰，熊光泽，等. 未知时序电路状态图生成算法及状态间路径的递归导出[J]. 计算机辅助设计与图形学学报，1998，10(2)：173-179

[14] 郭敏，肖梓样. PLD 解析技术研究[J]. 计算机工程与科学，1998，20(4)：66-69

[15] 王征东. MACH110 芯片解析系统研制[D]. 郑州：解放军信息工程学院，1999

[16] 李清宝，张平，等. 一种同步时序 PLD 逆向分析数据采集算法[J]. 计算机工程，2008，34(16)：10-12

[17] 樊敏. OLRDS 结构技术研究[D]. 郑州：解放军信息工程大学，2009

[18] Dijkstra E W. A note on two problems in connection with graphs [J]. Numerische Mathematik, 1959, 1:269-271

[19] Floyd R W. Algorithm 97: Shortest path[J]. Communications of the ACM, 1962, 5:345-350

[20] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4:100-107

[21] Nilsson N J. Artificial intelligence: a new synthesis[M]. Beijing: China Machine Press, 1999

[22] Sturton C, Hicks M, Wagner D, et al. Defeating uci: Building Stealthy and Malicious Hardware[A]// Proceedings of the 32nd IEEE Symposium on Security and Privacy[C]. 2011