第34卷 第16期 计 算 机 工 程 2008年8月
Vol.34 No.16 Computer Engineering August 2008
·博士论文· 文章编号：1000—3428(2008)16—0010—03 文献标识码：A 中图分类号：TP309.7

一种同步时序PLD逆向分析数据采集算法

李清宝，张 平，赵荣彩，曾光裕
(解放军信息工程大学信息工程学院，郑州 450002)

摘 要：采用逻辑分析法实现加密可编程逻辑器件(PLD)逆向分析的关键是为逻辑综合提供有效、完备的数据集，特别是对时序型PLD，在未知状态图的情况下，如何高效地采集到所有有效状态下的数据，是逆向分析研究的核心问题之一。该文在理论分析同步时序型PLD逆向分析可行性的基础上，提出一种适合多状态、复杂同步时序型PLD的高效数据采集算法，以动态建立非完全状态图为基础，求解状态驱动的最短路径，使得数据采集算法具有理想的时空开销。

关键词：可编程逻辑器件；同步时序；非完全状态图；最短路径；数据采集

Data Collecting Algorithm for Reverse Analysis of Synchronous Logic PLD

LI Qing-bao, ZHANG Ping, ZHAO Rong-cai, ZENG Guang-yu
(Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002)

【Abstract】The key problem in the reverse analysis of encrypted Programmable Logic Devices(PLD) using logic analysis techniques is to collect effective and self-contained data set, especially for timing PLD. An efficient data collecting algorithm is presented whichs suit for large scale and multi-state synchronous PLDs. The algorithm builds non-complete state graph and finds the shortest path for state migration form initial state to each effect state. The algorithm has ideal time and space cost and has been used in the PLD reverse engineering system.

【Key words】Programmable Logic Devices(PLD); synchronous logic; non-complete state graph; shortest path; data collecting

1 概述

近年来，可编程逻辑器件(Programmable Logic Device, PLD)在电子设备中大量使用，因为该类器件不仅功能可由设计者根据需要自行定制，而且还具有很强的加密功能。但是，加密器件的使用同时也给设备维护带来了很大困难，特别是进口设备一旦损坏，将难以修复，严重影响了设备的正常使用。为了解决这一矛盾，最好的方法就是将这些加密芯片进行逆向分析。常用的逆向分析方法主要有“克隆”法和逻辑分析法。“克隆”法就是采取某种技术手段避开加密位直接读取芯片内部编程码点。“克隆”法逆向分析出的芯片和原芯片完全一致，不足之处主要有2点：(1)对很多类型逻辑器件无能为力，如Lattice公司生产的带D后缀的、Atmel公司生产的带ATF前缀的和一些熔丝性OTP类逻辑器件；(2)“克隆”出的器件只能用于已知该芯片损坏后的替换，无法协助维修工程师了解其设计功能，从而引导系统的自动诊断与维护。

逻辑分析法是将待解析的器件看成一只“黑箱”，运用黑箱理论对其进行分析，通过输入不同组合、不同序列的激励信号，在输出端采集其对应输出，运用逻辑综合的方法推导出引脚间逻辑功能的一种安全有效的方法。该方法的优点是适用于各种工艺的PLD器件，对芯片没有任何损害，分析出的芯片引脚属性和内部逻辑功能有助于维修工程师实现对电子设备的自动诊断与维护和维修设备的测试程序集(Test Program Set, TPS)的开发。

“克隆”法需要有大量昂贵的物理设备支持，同时还要求操作者对集成电路制造工艺有一定的了解；另一方面还存在侵犯知识产权的嫌疑，故本文不作深入的讨论，仅对逻辑分析法进行讨论。采用逻辑分析法进行逆向分析时，数据采集是一个重要且困难的环节，特别对时序电路。时序电路的数据采集要求在判定芯片的输入、组合输出、时序输出和反馈引脚的基础上，为逻辑综合提供正确的、完整的数据，即要得到每一有效状态在所有可能输入下的输出、次态和高阻数据。

由于纯组合电路可看作是时序组合电路的一个子集，且较为简单，因此本文将重点讨论同步时序型PLD逆向分析中数据采集的若干问题，首先通过理论分析证明采用逻辑分析法进行同步时序性PLD逆向分析得可行性，然后从系统资源和算法实用性的角度出发，给出分状态数据采集算法，该算法基于数据采集过程中动态建立的非完全状态图，导出初始状态到各个状态迁移的最短路径，使得同步时序型PLD解析的数据采集有较理想的时空开销，最后给出算法的性能分析。

2 理论分析

2.1 同步时序型PLD

同步时序型PLD是数据处理和控制中最常用的一种编程形式[1]，其基本结构如图1所示，该类结构的显著特征是所有时序寄存器的输出由全局统一的时钟驱动[2]。其特征可以用5元组P来描述：$P=<\boldsymbol{I},\boldsymbol{O},S,F_o,F_s>$，其中：

基金项目：国家“863”计划基金资助项目“网络关键设备中核心芯片逆向分析与可控技术研究”(2006AA01Z404)

作者简介：李清宝(1967－)，男，博士研究生，主研方向：计算机软件与理论，信息安全；张 平，副教授；赵荣彩，教授、博士生导师；曾光裕，副教授

收稿日期：2007-09-20 **E-mail：**lqb215@vip.371.net

$\boldsymbol{I}$ 是一个有限的输入向量，定义形式为

$$\boldsymbol{I}=\{I_1,I_2,\cdots,I_n\},I_i\in T,i=1,2,\cdots,n,T=\{0,1\} \quad (1)$$

$\boldsymbol{O}$ 是一个有限的输出向量，定义形式为

$$\boldsymbol{O}=\{O_1,O_2,\cdots,O_m\},O_i\in B,i=1,2,\cdots,m,B=\{0,1,z\} \quad (2)$$

其中，z 表示高阻。

S 表示一个有限的有效状态集合：

$$S=\{s_1,s_2,\cdots,s_r\},r\le 2^k,k\le m \quad (3)$$

其中，k 为寄存器输出总数。

F_o 为输出映射函数：

$$F_o:\boldsymbol{I}\times S\to\boldsymbol{O} \quad (4)$$

F_s 为状态转换映射函数：

$$F_s:\boldsymbol{I}\times S\to S \quad (5)$$

根据式(5)，如果有 $s_j=F_s(\boldsymbol{I},s_i)$，其中，$s_i,s_j\in S$，$\boldsymbol{I}\in T^n$，则$S_i$称为现态，$S_j$称为次态，$\boldsymbol{I}$称为输入向量。

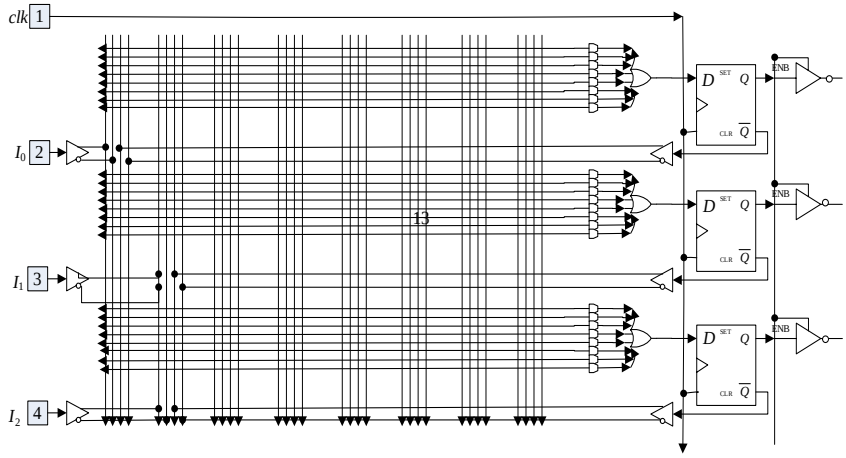

图 1　同步时序型 PLD 内部结构

2.2　逻辑分析法的可行性分析

同步时序型PLD逆向分析的关键在于采集到的同步寄存器输出端数据要完整。时序型PLD的数据采集与其他类型(如组合型)PLD的数据采集相比更为复杂，因为不仅要考虑输入输出间的映射关系，还要考虑状态间的转换关系[3]。为便于讨论，定义以下术语：

定义 1 在现态 $S_i\in S$ 条件下，施加一激励输入向量 $\boldsymbol{I}\in T^n$，采集状态为 $S_j\in S$，称 $\boldsymbol{I}$ 为已作用输入向量，指的是输出 $F_o(\boldsymbol{I},S_i)$ 已经由数据采集电路求得，否则称为未作用的输入向量。

定义 2 如果 $\forall\boldsymbol{I}\in T^n$，$\boldsymbol{I}$为$S_i$的已作用输入向量，则$S_i$为饱和状态，否则$S_i$为不饱和状态。

定义 3 对某状态S_i，在输入端穷举所有可能输入向量的过程称为状态S_i的输入加全，即该状态所有输入向量已穷举完成。

同步时序型PLD的逆向分析需要对所有有效状态完成输入加全，即要使所有有效状态达到饱和。当被分析逻辑器件处于某确定状态 $S_i\in S$ 时，可取任意的 $\boldsymbol{I}\in T^n$，求得 $F_o(\boldsymbol{I},S_i)$，但数据采集电路只能提供任意 $\boldsymbol{I}\in T^n$，不能提供任意的现态 $S_i\in S$，对于同步时序型PLD的逆向分析必须解决如何快速获得并加全所有可能出现的有效状态S_i的问题。

根据数字电路理论可知，按照规范设计的同步时序逻辑电路，其状态转移图必须是强连通的，把满足这种特性的状态图中的状态称为有效状态。针对这种同步时序型 PLD 有下面定理：

定理 1 同步时序的PLD满足对上电后的初始状态 S_0，$S_0\in S$，可以通过一系列操作将 S_0 驱动至另一状态S_i，$S_i\in S$。

定理 1 的正确性是非常直接的，因为实际应用的PLD上电后在正常工作前必须初始化进入某确定状态，即状态 S_0，$S_0\in S$，在初始状态 S_0 下可以通过一系列操作将 S_0 驱动至另一状态 S_i，$S_i\in S$，否则如果存在状态转移不可达，假设状态S_i (初始状态 S_0 可达S_i)不可达状态S_j，则必然存在初始状态 S_0 不可达状态S_j，电路系统正常工作时不会进入S_j状态，则状态S_j便成为无效状态，在电路系统中不起任何作用。所以对于一个有效状态S_i，可以通过序列操作由初始状态 S_0 驱动至S_i[4]。

根据定理 1，可以得出定理 2。

定理 2 同步时序寄存器型 PLD 可经由有限步操作加全所有有效状态，当全部有效状态达到饱和状态时，同步时序寄存器型 PLD 逆向分析数据采集完成，即得到了完整而且涵盖原始芯片等价逻辑功能的数据集合。

所以，根据所采集到的完整而且涵盖原始芯片等价逻辑功能的数据集合进行逻辑综合处理，生成的引脚逻辑布尔表达式必然和原设计功能等价，即逆向分析产生的结果可以替代原芯片正常工作，这充分证明了同步时序型 PLD 是可解析的。

虽然已证明逻辑分析法进行逆向分析是可行的，但是在具体实现时，还存在时间和空间复杂度的约束问题，下文将进一步讨论。

3　数据采集算法

针对同步时序型 PLD 的逆向分析，传统的逻辑分析法需要采集记录所有的数据之后，即所有有效状态处于饱和状态后，再通过对这些数据进行逻辑综合以获得分析芯片的逻辑功能表达式，换句话说就是有效状态遍历采集操作过程和逻辑综合操作过程独立分割、完全串行执行。这种方法对小规模 PLD 器件是可行的，但随着芯片集成度的提高，需要采集存储和处理的数据迅速增长，数据的存储与处理已成为采用逻辑分析法逆向解析 PLD 器件的瓶颈。

设待解析芯片的输入引脚数为n，寄存器输出引脚数为r，则最多有 2^r 个有效状态输出，每个状态的输入条件有 2^n 项，所以，需要记录的数据向量总量为 $2^n\times 2^r=2^{n+r}$，当n和r达到一定值，计算机系统的内存资源将无法满足存储和处理的需求。而且如此大的数据量，目前的逻辑综合算法也面临一定的困难[5-6]。

为了降低数据采集和处理过程中对内存资源的苛刻要求，本文提出了一种分状态数据采集算法，从上电初始状态开始记录状态的转移关系，通过查找状态转移图有目的地控制状态的转移，依次对出现的各个状态输入加全，使其达到饱和。这样在数据文件中，同一状态的所有数据是相邻的，可以以状态为单位进行内外存交换，对内存的需求可以降为 2^n。对于已饱和状态的数据即可递交逻辑综合算法进行化简，从而实现数据采集和逻辑综合的并发执行。分状态数据采集算法描述如下：

算法 1

(1)设芯片上电后正常工作前的初始状态为 S_0，令当前状态 S_i=S_0。

(2)选取状态S_i的未作用输入向量 $\boldsymbol{I} \in T^n$ 采集 $F_o(\boldsymbol{I},S_i)$，存储结果；采集状态S_i的次态 S_j=$F_s(\boldsymbol{I},S_i)$，调用动态状态图生成算法，更新状态图。

(3)判断S_i是否饱和，若饱和，转(4)；若不饱和，查找状态转移图是否存在有转移到状态S_j的路径，若有驱动实现状态转移到S_j，令 S_i=S_j 返回(2)，否则转(5)。

(4)判断是否所有的 $S_i \in S$ 饱和，若饱和，则数据采集过程结束；若不饱和，则继续向下执行。

(5)待解析的PLD器件重新上电，通过数据采集电路设置状态为 S_0，调用最短路径导出算法，搜索到当前最短路径最短的未饱和状态S_i。

(6)根据导出的最短路径通过接口电路驱动将待解析PLD从状态 S_0 驱动至状态S_i，转(2)。

4 状态图的动态生成与状态迁移最短路径算法

时序系统中常采用状态图来描述数字电路系统的行为，表示状态间转换关系，形象说明输入向量和现态与所对应次态的关系。时序电路状态图定义如下：

定义 4 时序电路状态图 G=<N,E>，其中，N 为顶点集，$s \in N$ 为已出现状态；E 表示有向边集，E={< s_i,s_j >|如果在状态 s_i 下施加输入向量 $\boldsymbol{I}_i$ 到达稳定状态 s_j，且< s_i,s_j >上的值为 $\boldsymbol{I}_i$ }。

对于一个有 n 个输入、r 个寄存器的同步时序机，其有效状态数最多可达 2^r 个，因此，在极端情况下时序机的完全状态图是一个含有 2^r 个顶点、 $2^r \times 2^n$ 条边的有向图。

在分状态数据采集过程中，为实现某状态下的输入加全，需要通过查找状态转移图，找出状态间转移关系，按照转移关系施加一定序列的激励信号实现状态迁移，因待分析芯片状态转移关系未知，没有现成已知状态图供迁移使用，需要在逆向分析过程中动态建立状态图。由于在数据采集过程中仅需要状态间迁移路径信息，因此从时空性能上考虑，在建图过程中可对所记录的状态信息进行简化，建立简化的非完全状态图。在实现中采用顺序表记录每个状态的路径迁移信息，每个已出现状态的记录结构如下：

状态值	最短路径长度	前趋状态	输入向量

其中，状态值项记录目标状态 s_i 的编码值，最短路径长度是指从开始状态 S_0 经过一系列状态到达目标状态所经过的边的数目(经过的状态数)，前趋状态记录最短路径上 s_i 的前驱状态 s，输入向量是指以 s 为现态到达次态 s_i 的输入向量。在数据采集过程中，如果出现新状态 s_i，则调用动态状态图生成算法，更新简化状态图。

算法 2 动态状态图生成算法

在现态 s 下，对于新采集到的状态 s_i ：

(1)判断 s_i 是否已在简化状态图中有记录。如果有，则继续执行；如果没有，则转(3)。

(2)d=s 的最短路径长度+1，比较 d 与 s_i 的当前最短路径长度。如果 d< s_i，则为当前最短路径长度，算法结束；否则继续执行。

(3)更新状态 s_i 的信息：

1)令状态值= s_i ；

2)最短路径长度=s 的最短路径长度+1；

3)路径上的前趋状态=s；

4)输入向量=s迁移到s_j的输入向量。

从简化状态图中可以方便地导出初始状态s_0到任意状态s_i的最短路径。

算法 3 最短路径导出算法

(1)取状态s_i的记录项。

(2)判断s_i的前趋s是否为s_0，如果为s_0，算法结束；否则继续执行。

(3)记录s→s_i的输入向量，转(1)。

按照算法 3 所记录的输入向量序列的反向顺序，施加一系列输入，即可将状态有s_0驱动至s_i。

根据算法 1~算法 3 可以得出，对任意状态 s，算法所求得的状态迁移路径是最短的。

5 性能分析

本文所提出的基于最短路径驱动状态迁移的分状态数据采集的算法，在时空性能上优于自然路径数据采集算法。自然路径数据采集是指状态机到达某一状态时，无需查找最短路径，可在此状态下输入该状态的下一个激励信号，观察是否转移到其他状态，若没有转移，继续输入下一个激励信号，否则到转移状态下重复上述过程。在自然路径法采集数据过程中需要记录状态饱和标志，如果现态已饱和，则不记录相应数据。2 种方法在性能上存在以下差别:

(1)从空间上考虑，自然路径数据采集所需内存空间为$O(2^{n+r})$，而分状态数据采集方法对记录数据所需内存空间的需求大大降低，所需内存空间为$O(2^n)$，新的数据采集算法采集存储和处理所需内存空间只与输入引脚数目直接相关，与状态输出数目无关，使得大规模PLD的逆向分析成为可能。

(2)分状态采集记录激励数据和状态信息可以真正实现同步时序型逻辑功能不明器件数据采集过程和逻辑综合处理过程的并行执行，这种并行执行不仅大大降低内存需求的复杂度还将大大缩短处理的时间，从而有效提高逆向分析过程的时效性能。

(3)状态图的简化存储方式降低了对临时存储空间的需求。对于状态数为 N 的同步时序机，算法的空间复杂度为$O(N)$。

(4)从简化状态图中可以方便地导出开始顶点到任意状态 s 的最短路径，算法 3 的平均时间复杂度为 $O(N/2)$。根据算法 1~算法 3 可以得出，对任意状态 s，算法所求得的状态迁移路径是最短的。

(5)由于同步时序型 PLD 芯片中状态出现是未知的，因此具有很大的随机性。图 2 是在数据采集过程中生成的一个状态转移图。

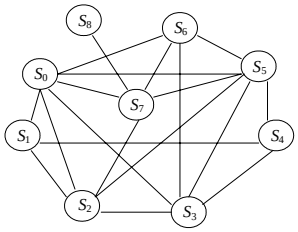

图 2　自然路径采集算法无法加全的情况

(下转第 21 页)