第35卷 第14期 计 算 机 工 程 2009年7月
Vol.35 No.14 Computer Engineering July 2009
·开发研究与设计技术· 文章编号：1000—3428(2009)14—0258—03 文献标识码：A 中图分类号：TP301.6

基于布线资源图的FPGA互连测试算法

代 莉，梁绍池，王伶俐
(复旦大学专用集成电路国家重点实验室，上海 201203)

摘 要：分析基于静态随机访问内存的FPGA开关盒互连资源，提出一种自动生成且与应用无关的测试配置集算法，通过建立布线资源图，根据线网的走向动态设定各边的权重，利用改进的Kruskal算法，自动产生测试配置集。对于FPGA不同的互连结构，该算法对互连资源中的开路和短路故障的覆盖率能够达到100%，且具有测试配置个数少、运行速度快以及与具体硬件结构无关等优点。

关键词：布线资源图；Kruskal算法；静态随机访问内存

FPGA Interconnect Test Algorithm Based on Routing Resource Graph

DAI Li, LIANG Shao-chi, WANG Ling-li
(State Key Lab of Application Specific Integrated Circuit, Fudan University, Shanghai 201203)

【Abstract】 The interconnect resource for Field Programmable Gate Array(FPGA) based on Static Random Access Memory(SRAM) is analyzed. A test matched sets algorithm with automatic generation is presented, which has no relation to application. By constructing the routing resource graph, this algorithm assigns a dynamic weight of each edge by the direction of nets and runs the modified Kruskal algorithm to obtain the minimum numbers of test configurations. For different FPGA routing architectures, the cover rate to faults of stuck-open and stuck-closed reaches 100%, and has some characteristics such as few number of testing configurations, high operating speed and has no relation to hardware structure.

【Key words】 routing resource graph; Kruskal algorithm; Static Random Access Memory(SRAM)

1 概述

基于静态随机访问内存(Static Random Access Memory, SRAM)的现场可编程门阵列(Field Programmable Gate Array, FPGA)具有电路功能可重配置、设计周期短、通用灵活等特点，在国防武器装备、民用通信、汽车和医疗等领域中有着广泛应用。

FPGA的硬件测试保证了其对于任意用户配置都能正确实现其功能。在测试FPGA芯片时，需要设计一系列测试配置，即测试配置集，并将每个测试配置下载到芯片中，在输入端加入一组测试向量，同时对输出结果进行分析，测试芯片是否存在故障。

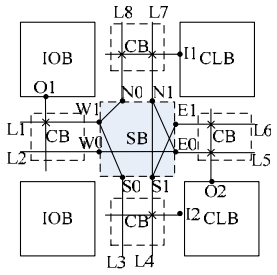
由于FPGA中可编程点(Programmable Interconnect Points, PIPs)大部分存在于互连中[1]，使得互连资源的测试面临巨大挑战：用尽可能少的测试配置文件覆盖所有要求的故障类型。

针对Xilinx公司Virtex系列芯片，文献[2]运用最大流算法，对横向、竖向及对角线方向的开关矩阵分别建立图，但针对互连线段的短路故障，利用了Virtex芯片硬件结构的特殊性，所有未使用到的互连资源可以被驱动为“0”或“1”，具有硬件结构上的局限性，同时各方向的建图复杂度也较高。

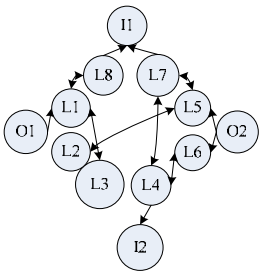
本文基于布线资源图，提出一种与硬件结构无关的测试配置集快速生成算法，它不仅能够覆盖所有的故障类型，而且配置个数较少。

2 背景知识

FPGA通常包括大量可编程逻辑单元(Configurable Logic Block, CLB)和可编程互连资源[3]，经典对称式的FPGA互连资源包括开关盒(Switch Box, SB)、连接盒(Connection Box, CB)以及互连线段，如图1所示。



(a)互连结构



(b)布线资源结构

图1 经典FPGA互连结构及其布线资源示意图

在图1(a)中，I1,I2是CLB的输入管脚，O1,O2是IOB/CLB的输出管脚，L1~L8是互连线段，每条边代表节点间一种可

基金项目：国家自然科学基金资助项目(60676020)
作者简介：代 莉(1984－)，女，硕士研究生，主研方向：FPGA软件系统开发；梁绍池，硕士研究生；王伶俐，副教授
收稿日期：2009-02-04 **E-mail：**llwang@fudan.edu.cn

能的连接关系，即在 2 个节点间存在一个 PIP，由于 PIP 可以由传输管或缓冲器构成，因此连接关系用有向边表示，如 O1 与 L1 存在连接关系，由于 O1 为输出管脚，因此信号只会从 O1 流向 L1，在 O1 与 L1 间存在一条单向边，而互连线段 L1 与 L8 之间通过传输管连接，另外，CB 将 CLB 或输入输出单元(Input/Output Block, IOB)的管脚连接到互连线段，SB 则连接不同的互连线段，随着 FPGA 阵列规模不断扩大，FPGA 的结构也有较大改进。

在图 1(b)中，节点的集合代表 FPGA 的互连资源，这些节点包括 CLB, IOB 的输入输出管脚以及互连线段。互连线段之间的连接方向可以分为横向连接、竖向连接、左对角线连接和右对角线连接 4 类。

FPGA 互连资源中存在的故障一般分为 2 类：开路和短路。开路故障包括互连线段的开路故障和 PIP 的常开故障；短路故障包括互连线段的短路故障和 PIP 的常闭故障。由于 PIP 的常开和常闭故障是指 SRAM 的编程信息不能改变此开关的状态，因此故障类型还包括互连线段的固定 0 故障和固定 1 故障，两者可以分别等效为与电源线 VDD 和地线 GND 的短路故障。

3 测试步骤

3.1 测试配置生成

为检测 FPGA 互连资源中存在的上述故障，测试配置需要满足以下条件：

条件 1 对于互连线段开路故障，在生成的所有测试配置文件中，所有的待测互连线段都必须有信号通过至少一次。

条件 2 对于 PIP 常开故障，在生成的所有测试配置文件中，所有的 PIPs 点都必须至少接通一次。

条件 3 为检测 PIP 的常闭故障,先要将 PIP 配置为断开，然后将 PIP 连接的 2 段互连线段分配到不同的被测线网，对 2 条被测线网施加不同的激励，根据结果能够检测出 PIP 的常闭故障[2]。

条件 4 互连线段的桥接故障，可采用与 PIP 常闭故障类似的方法。对每个开关盒连接的互连线段，两两分配到不同的线网中来检测桥接故障。

在实际情况中，当条件 2 满足时，条件 1 一定满足；当条件 4 满足时，条件 3 一定满足。

3.2 测试向量生成

当测试配置下载到 FPGA，还需要生成相应的测试向量。文献[4]说明了对于 n 条线网，$1+\lceil \mathrm{lb}n \rceil$ 个测试向量就可以检测出任何开路和短路故障。需要注意的是，在 n 条线网中，没有包含线段 VDD 和 GND，因而实际测试的线网总数为 $n+2$，为检测包括固定 0 和固定 1 在内的所有故障问题，所需施加的测试向量个数为 $1+\lceil \mathrm{lb}(n+2) \rceil$。

4 算法实现

4.1 Kruskal 算法的改进

Kruskal 算法是生成无向连通图的最小生成树问题的经典算法。该算法采用一种不相交集合数据结构，以维护几个互不相交的元素集合[5]，初始化每个节点为一棵树，以权重排序后依次遍历每条边，由于只有不会构成回路的边加入正在生长的森林中，因此保持了树的特性，当遍历边结束后，形成一棵最小生成树，包含所有的点。

由于 FPGA 芯片中的 SB 互连资源一般由传输管构成，因此线资源图中这些节点间的双向边可等效为无向边。针对 SB 的互连资源，本算法先从布线资源图中生成仅包含 SB 单元待测试的图 $G_{\mathrm{test}}=(V,E)$，其中，V 是所有节点的集合；E 是所有边的集合。在该图上运行 Kruskal 算法。每个元素集合维护一棵树，这些元素连接成的边没有回路，满足线网的基本要求，遍历边后，一棵树就是一条完整的线网，而所生成的森林就是一种测试配置，运用 Kruskal 算法能够快速地生成测试配置集。

为保证故障检测的覆盖率达到 100%，需要对 Kruskal 算法增加一些限制条件。由于运行一次 Kruskal 算法不能保证所有边的使用，因此根据条件 1 和条件 2。当仍存在边没有构成待测线网时，需要多次运行 Kruskal 算法，直到生成的测试配置文件涵盖了所有的边。同时，由于条件 4 包含条件 3，根据条件 4，对每个点 $v \in V$，与该点 v 在同一个 SB 中的点都必须至少一次属于不同元素集合，这样该点与同一个 SB 中其他点至少一次包含于不同的待测线网，以检测互连线段的短路以及 PIP 的常闭故障。根据 Kruskal 算法的特点，遍历所有边完成后，所有的节点都包含在配置文件中，从一个 SB 的角度来看，若同一个 SB 中的边属于不同的待测线网，则该节点与其他节点包含于不同的待测线网的个数最多，则每个配置文件都检测了尽可能多的互连线的短路以及 PIP 的常闭故障。

根据上述分析，生成测试配置还需要满足：

条件 5 在一个配置文件中，每条线网经过同一个 SB 至多一次。

根据条件 5，给出安全边的定义如下：

定义 1(安全边) 每次加入的边，必须保证加入后每个元素集合维护的仍然是一棵树，即没有回路。同时，该边所联系的两棵树合并后，这棵树所构成的线网经过同一个 SB 至多一次。

在进行 Kruskal 算法时，需要根据以上定义确定边所联系的两棵树是否能够合并。

4.2 权重的设置

运行 Kruskal 算法先要根据各边的权重进行排序。如何设定边的权重直接影响所需的配置文件数。

本文采用的方法是当运行 Kruskal 算法时，先设定该线网的走向，然后根据线网的走向，设定各边权重如下：

Wedge=SB 序号×边方向种类的个数+方向因子

由于互连线段的连接方向有 4 种，因此图 G_{test} 中边的方向种类也有 4 种，对应特定的 FPGA 模型，其边方向的种类个数是个常数。运行 Kruskal 算法时，选择横向、竖向、左对角线或右对角线方向其中一个方向为线网走线的方向。由于横竖向方向走线各不影响，因此可合并为一个线网方向。本文所采用的模型，其边方向的种类个数为 3，分别是横竖向、左对角线方向和右对角线方向。另外，式中 SB 序号及方向因子的定义如下：

定义 2(SB 序号) 根据线网走线的方向，对 SB 根据其坐标位置设定其序号。

定义 3(方向因子) 根据线网走线的方向，一旦边的方向与线网走线一致，则设定方向因子的值小，其他方向次之。

图 2 给出了 SB 中右对角线方向连接的 SB 序号和方向因子的设定方法。为使算法在连接时尽量偏向右对角线方向，SB 编号由左下角依次向右上角编号。由于线网连接的方向为右对角，因此该方向的边其方向因子为 0，横竖向及左对角

线方向的边方向因子依次为 1 和 2。

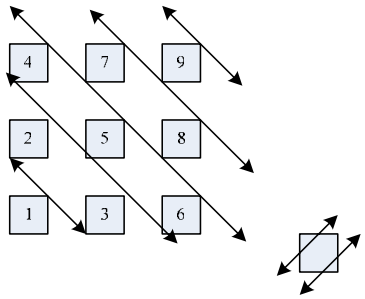


图 2　右对角线情况下的 SB 序号设置

对于使用过的边，需要加上一个固定的权值，从而能将其与未使用到的边区分开来，以便下一次运行 Kruskal 算法时优先选取没用到的边。该权值的设定与该芯片中待测试的 PIP 总个数 T 有关，一般设定该权值大于 T。

4.3　实现步骤

根据上述分析，算法的整体流程如图 3 所示。

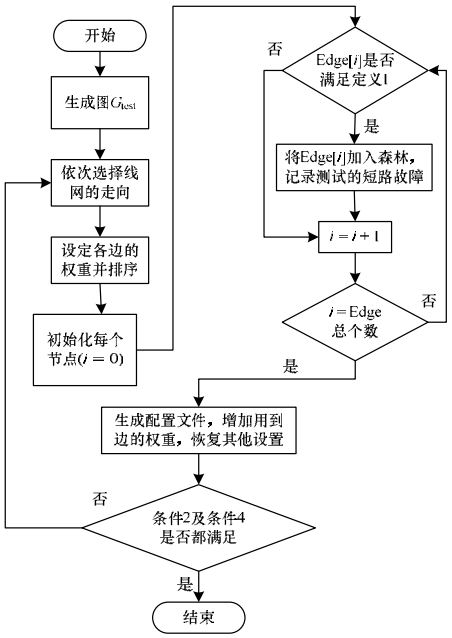


图 3　算法流程

在生成 G_{test} 时，对于每个节点，需要记录与该节点可能存在短路故障的节点。当边成功加入森林后，节点与记录的其他节点处于不同的线网，这些节点可从表中删除。当所有节点的表为空时，表示目前生成的配置文件已满足条件 4。

另外，Kruskal 算法运行的时间为 $O(|E|\lg|E|)$，假设运行 Kruskal 算法的次数为 k，则算法运行的时间为 $O(k|E|\lg|E|)$。由于 $k<|E|/|V|$，因此整个算法运行的时间为 $O(|E|^2\lg|E|/|V|)$。

5　实验结果

本文算法对一些小规模的例子生成了相应测试配置文件，这些互连结构规模及各种线型的数量规格如表 1 所示，其所需的测试配置文件个数如图 4 所示。以 3×3 规模的 FPGA 模型为例。在选择横竖向时，生成的线网结构如图 5 所示。

表 1　互连结构规模及各种线型的数量

模型	规模	通道线型及条数 (2 倍线/4 倍线/长线)	待测 PIP 个数
1	3×3	2/0/1	176
2	5×5	2/4/1	951
3	6×6	2/4/2	1 500
4	8×8	4/8/4	5 345
5	10×10	4/8/4	7 985
6	12×12	6/12/6	16 717
7	14×14	6/12/6	22 259
8	16×16	8/16/8	38 113
9	18×18	8/16/8	47 613
10	20×20	12/24/12	89 612

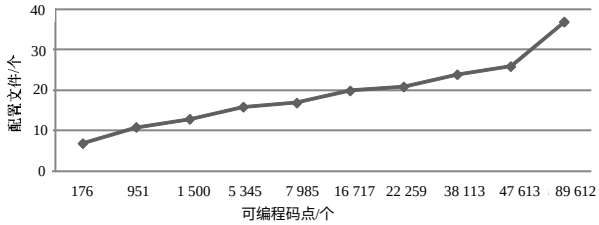


图 4　PIP 个数与生成测试文件个数的关系

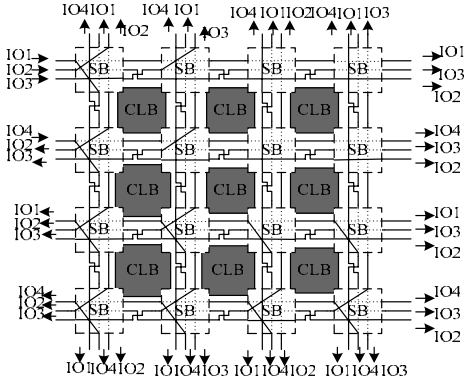


图 5　3×3 互连规模时横竖向连线配置结果

6　结束语

本文为 FPGA 中的 SB 布线资源提出一种基于布线资源图的测试配置集的自动生成算法。实验结果表明，当测试码点个数为 89 612 时，36 个配置文件能够达到 100%的故障覆盖率。

参考文献

[1] Asadi G. Soft Error Rate Estimation and Mitigation for SRAM-based FPGAs[C]//Proc. of ACM Int'l Symp. on Field Programmable Gate Arrays. California, USA: [s. n.], 2005.

[2] Tahoori M. Application-independent Testing of FPGA Interconnects[J]. IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems, 2005, 24(11): 1774-1783.

[3] 邢　虹, 童家榕, 王伶俐. 一种 FPGA 配置文件压缩算法[J]. 计算机工程, 2008, 34(11): 260-262.

[4] Kautz W. Testing for Faults in Wiring Networks[J]. IEEE Trans. on Computers, 1974, 23(4): 358-363.

[5] Cormen T H. Introduction to Algorithms[M]. Cambridge, MA, USA: MIT Press, 1990.

编辑　陈　文