第 38 卷 第 2 期
Vol.38 No.2

计 算 机 工 程
Computer Engineering

2012 年 1 月
January 2012

• 工程应用技术与实现 • 文章编号：1000—3428(2012)02—0245—03 文献标识码：A 中图分类号：TP303

一种时序优化的通用 FPGA 装箱算法

刘 垚

(复旦大学专用集成电路与系统国家重点实验室，上海 201203)

摘 要：提出一种时序优化的通用 FPGA 装箱算法。将配置电路与用户电路转化为有向图，解决子图同构问题。将线网延时作为变量，定义关键度，以此为代价函数进行装箱，达到优化时序的目的。在 VPR 平台上进行实验，结果表明，该算法的时序性能较优，并可应用于不同的可配置逻辑块结构中。

关键词：现场可编程门阵列；工艺映射；装箱算法；时序优化

Universal FPGA Packing Algorithm of Timing Optimization

LIU Yao

(State Key Laboratory of ASIC & System, Fudan University, Shanghai 201203, China)

【Abstract】 A universal timing optimization oriented FPGA packing algorithm is proposed in this paper. Configure and user circuits are converted to directed graphs to solve the sub-graph isomorphism problem. Aiming for timing optimization, net delay is used as variable to define the criticality, which is used for cost function to guide packing procedure. Experimental results on VPR platform prove this algorithm performs less timing delay than other similar packing algorithms, and can applied in various kinds of FPGA CLBs.

【Key words】 Field Programmable Gate Array(FPGA); technology mapping; packing algorithm; timing optimization

DOI: 10.3969/j.issn.1000-3428.2012.02.082

1 概述

现场可编程门阵列(Field Programmable Gata Array, FPGA)是由若干个可配置逻辑块(Configurable Logic Block, CLB)和布线资源组成。CLB 中包含了 FPGA 功能电路单元，这些电路单元由编程点控制，可以配置成不同的逻辑。FPGA 中的基本逻辑单元有查找表(Look Up Table, LUT)、多路选择器(MUX)、触发器等，触发器中以 D 触发器(D Flip Flop, DFF)最为常见。

当用户使用 FPGA 设计实际电路时，其工艺映射步骤又可细分为映射和装箱 2 个过程：映射是将门级网表转化至 FPGA 功能电路元件 LUT、MUX 和 DFF 中；装箱是将 LUT、MUX 和 DFF 级别的网表映射到相应 FPGA 的 CLB 中。FPGA 专用综合工具可以将电路网表直接综合成 FPGA 元件级的电路网表，因此，工艺映射与逻辑综合在映射这个部分发生了一部分交叠，本文研究的工艺映射专指装箱。

FPGA 装箱研究多集中在装箱后电路的面积、时序或功耗等方面。与商用 FPGA 结构不同，学术研究多采用 BLE (Basic Logic Element)学术模型。文献[1]提出了基于 BLE 的 T-VPack 算法[1]，结合 VPR(Versatile Place and Route)学术模型平台[2]，提供了学术模型解决方案。后人在 T-VPack 基础上进行了深入研究，如文献[3]提出基于布通率的 RPack 算法，文献[4]基于开关活跃度提出了 CAP 算法以降低动态功耗，文献[5]提出 T-RDPack，综合考虑了时序和布通率等。

但 T-VPack 算法只提供了一个理论上的指导，无法应用在实际的 FPGA 上，不具通用性。因此，提出一种通用、高性能的装箱算法非常必要。文献[6]提出 XC3000 和 XC4000 系列芯片的通用装箱算法 FDUMap。文献[7]基于子图同构思想，提出更具广泛适应性的 PLBMAP。本文基于 PLBMAP 子图同构的思想，结合 T-V Pack 算法中的代价函数，提出一种时序优化的通用 FPGA 装箱算法。

2 时序优化的通用 FPGA 装箱算法

2.1 术语说明

本文算法中用到的术语说明如下：

(1)目标电路：等价于用户电路的 FPGA 基本逻辑元件级电路。

(2)样本电路：描述 CLB 配置情况的电路，将所有可用到的样本电路集合起来建立样本电路库，用于和目标电路进行匹配。

(3)有向图：将电路图转换成有向简单图 $G=(V,E)$，将电路元件当作一个图的顶点，其中，V 是顶点的集合，一个电路元件就对应有向图中的一个顶点；E 表示边的集合，从一个元件的输出端到另一个元件的输入端只会存在一个连接，所以产生的边均为简单边。将它们之间的连接关系标记为图的边，得到的图称为有向图。以 BLE 模型为例，在建模时将元件抽象为点，将线网抽象为边，转化为有向图。给有向图添加约束信息形成的图叫带权图。

(4)子图同构：子图同构是一个经典的数学问题，就是在一个给定的目标图中，找出另一个样本图同构的像。该问题是 NP 完全问题。给定图 $G1$ 和 $G2$，如果存在一个从 $G1$ 的顶点集到 $G2$ 的顶点集上的双射函数 f，和一个从 $G1$ 的边集到 $G2$ 的边集上的双射函数 g，使得 $G1$ 中的边 e 与顶点 v 和 w 相关联当且仅当 $G2$ 中的边 $g(e)$ 与顶点 $f(v)$ 和 $f(w)$ 相关联，就说明图 $G1$ 和 $G2$ 是同构的。因此，装箱问题等价于子图同

作者简介：刘 垚(1987－)，男，硕士，主研方向：FPGA 工艺映射

收稿日期：2011-07-12 **E-mail：**082052049@fudan.edu.cn

构问题。

2.2 有向图的建立

首先需要将 CLB 的各种实际电路配置建立为有向图。以 Xinlinx SpartanII 的 FPGA 为例，其 CLB 结构见图 1。

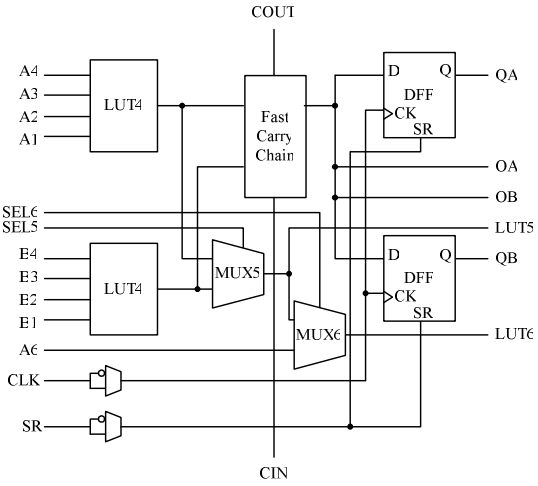
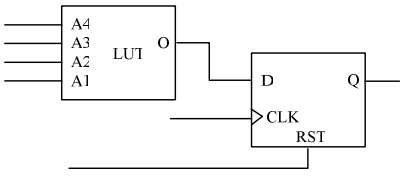
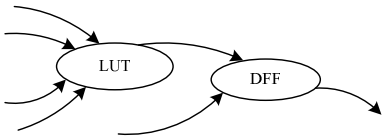


图 1 SpartanII 的 CLB 结构

将 FPGA 的 CLB 进行配置后可以得到不同的功能电路。简单时序功能配置电路与建模后的有向图如图 2 所示。由 2 个 LUT4 以及选择器构成的 LUT5 功能电路及其建模后的有向图如图 3 所示。

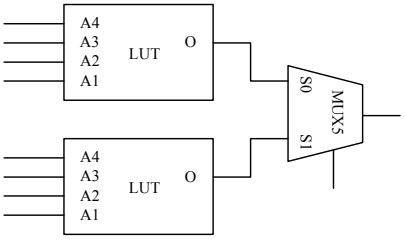


(a)简单时序功能配置电路

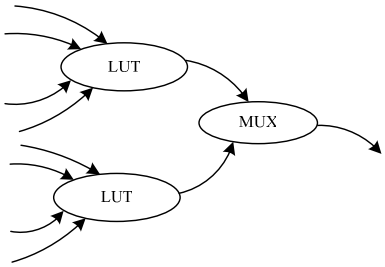


(b)建模后的有向图

图 2 简单时序功能配置电路及其有向图



(a) 2 个 LUT4 组成的 LUT5 电路



(b)建模后的有向图

图 3 2 个 LUT4 组成的 LUT5 电路及其有向图

建立好样本电路有向图后，将样本电路写入 CLB 配置文件用于装箱时的调用。不同的 CLB 结构需要构建不同的配置文件，同一种 CLB 结构根据实际使用情况可酌情加减配置。同理可将用户电路转化为有向图，这个过程在程序读入用户电路网表时执行，不需要预先执行。由于 LUT 的 4 个输入端并不一定都会用上，在程序中可以针对用户电路的实际情况灵活匹配。

2.3 代价函数

子图同构问题的解不是唯一的，这表示着同一个电路对应着不同的装箱结果，而不同的结果表示同一个用户电路经过不同的装箱实际性能迥异。因此，不仅仅需要解决子图同构的问题，还要需要寻找最优解。这里引入关键度作为代价函数来指导装箱。PLBMAP 是在 FDUMAP 贪婪算法的基础上，以减少 CLB 间线网数为目标，达到减少装箱后面积的目的。本文算法采用以线网延时作为变量的代价函数达到优化时序的目的。

在进行装箱前，首先对目标电路进行时序分析，从输入端口开始，与输入端口相连的线网设为 0，顺序遍历路径，碰到 1 个 LUT，该值加 1，碰到 1 个 MUX，该值加 0.5，碰到 DFF 标记为时序端口，截断该时序路径，如此遍历所有时序路径，最终每个线网都会得到一个或多个值，取其最大值作为该线网的延时(此时是对用户电路进行分析，不考虑互连和 CLB 延时情况)，标记到线网对应有向图的边上去。

对于处在同一时序路径上的线网 a 和线网 b，定义关键度为：

$$Criticality = 1 - \frac{Slack}{MaxDelay}$$

其中，$Slack$ 为线网 a 与线网 b 之间的延时差；$MaxDelay$ 取整个电路的最大延时。这表示关键度越接近 1，2 条线网越相关，装箱时应优先将这些线网放入一个 CLB 中。

2.4 算法流程

本文算法的流程如下：

(1)读入 CLB 配置文件，将所有配置电路转化为有向图，存储该配置下输入输出端口间延时(给输出文件使用，装箱时不考虑)。所有样本电路有向图组成集合 P。

(2)读入用户电路网表，将用户电路转化为有向图 $G1$。对 $G1$ 进行分析，得到每条边的理想最大延时，反标到边上。

(3)取 $G1$ 值最大的边作为当前边 e，与之相连接的点作为当前点 v，取一个新的 CLB 并开始匹配。设 G 为匹配图，此时 $G=e+v$。

(4)计算通过 v 与 e 相连的边的关键度。设关键度值最大的边为 e'，通过 e' 与 v 相连的点为 v'。

(5)遍历 P，尝试寻找图 $G' \in P$，使 $G' = G + v'$。

1)若不存在，表示当前 CLB 已满，标记 G 为已匹配，将 $G1 = G1 - G$，转至步骤(3)；

2)若存在且该时序路径未完全处理，表示当前 CLB 未满，将 $G = G'$，$e = e'$。转至步骤(4)；

3)若存在且该时序路径已完全处理，标记 G 为已匹配，将 $G1 = G1 - G$，转至步骤(6)。

(6)判断是否 $G1$ 为空集：

1)若 $G1 = \varnothing$，表示装箱完成，转至步骤(7)；

2)若 $G1 \neq \varnothing$，表示装箱未完成，转至步骤(3)。

(7)输出装箱后电路网表，并根据样本实际情况输出端口间延时信息。算法结束。

3　实验与分析

现代 FPGA 互连延时远高于 CLB 内部延时，如果忽略互连延时，只考虑 CLB 内部延时，会使数据不明确，甚至导致数据出错。因此，延时数据应与整个布局布线过程相结合。本文从 MCNC 测试用例中选取了 10 个常用的测试用例，分别使用本文装箱算法、PLBMAP 和 T-VPack 3 种装箱算法，应用在不同 FPGA 中，经过布局布线程序[8]后，对得到的关键路径延时信息进行比较，3 种算法在 VPR 平台上的结果如表 1 所示。

实验结果表明，在 VPR 平台上，本文算法表现出了较优的性能，而且仅仅需要更改配置库文件就可以针对不同 FPGA 的 CLB 结构进行装箱，表现出了 T-VPack 所不具备的通用性，其时序相对于 PLBMAP 时序也优化了 12%以上。

表 1　3 种算法的关键路径延时对比

测试电路	关键路径延时/ns			延时优化/(%)	
	本文算法	PLBMAP 算法	T-VPack 算法	PLBMAP 算法	T-VPack 算法
alu4	67.31	78.19	69.26	14.0	2.8
c1196	41.79	45.53	42.08	8.2	0.6
c1238	49.03	57.18	51.27	14.3	4.4
ex5p	74.66	88.52	77.13	15.7	3.2
apex2	77.98	90.49	79.40	13.8	1.8
apex4	73.43	87.39	76.38	16.0	3.9
misex3	70.58	77.46	73.12	8.9	3.5
seq	75.17	83.68	76.14	10.2	1.3
decod	24.83	26.81	24.39	7.4	−1.8
e64	39.87	45.11	41.09	11.6	3.0
平均	—	—	—	12.01	2.27

4　结束语

本文提出了一种时序优化的通用 FPGA 装箱算法，已成功应用于多种 FPGA 结构中，算法性能较优且具有广泛的通用性。对不同的 FPGA，如何更好地使用特定的 CLB 结构资源是下一步的研究方向。

参考文献

[1] Betz V, Marquardt A, Rose J. Using Cluster-based Logic Blocks and Timing-driven Packing to Improve FPGA Speed and Density[C]//Proc. of ACM International Symposium on FPGA. Monterey, USA: [s. n.], 1999.

[2] Betz V. VPR and T-VPack User's Manual[EB/OL]. [2011-05-15]. http://www.eecg.utoronto/vpr/.

[3] Bozorgzadeh E, Ogrenci-Memik S, Sarrafzadeh M. RPack: Routability-Driven Packing for Cluster-based FPGAs[C]//Proc. of ASP-DAC'01. Yokohama, Japan: [s. n.], 2001.

[4] Hassan H, Anis M, El Daher A, et al. Activity Packing in FPGAs for Leakage Power Reduction[C]//Proc. of DATE'05. Munich, Germany: [s. n.], 2005.

[5] Pandit A, Easwaran L, Akoglu A. Concurrent Timing Based and Routability Driven Depopulation Technique for FPGA Packing[C]//Proc. of ICECE'08. [S. l.]: IEEE Press, 2008.

[6] 倪　刚, 童家榕, 来金梅. 基于对可编程逻辑块建模的 FPGA 通用装箱算法[J]. 计算机工程, 2007, 33(6): 239-241.

[7] 蔡　丹, 来金梅, 童家榕. PLBMAP: 高性能通用 FPGA 可编程逻辑块映射算法[J]. 微电子学与计算机, 2008, 25(8): 40-44.

[8] 徐嘉伟, 来金梅, 童家榕. 可配置宏的快速 FPGA 布局算法[J]. 计算机工程, 2009, 35(16): 228-230.

编辑　顾姣健

(上接第 244 页)

储单元和运算单元都进行了复用，有效地节约了资源。复用后的 IP 用了 30 850 个 Slice Flip Flops，占总量的 24%，存储资源占用 222 个 FIFO16/RAM16s，占总资源的 40%。本文处理器与其他具有代表性的 FFT 处理器的性能比较如表 3 所示。可以看出，本文设计的 FFT 处理器在处理速度上优于其他设计。

表 3　FFT 处理器的性能比较

处理器	点数	位宽/bit	频率/MHz	处理时间/μs
Xilinx FFT 处理器	16 384	16	211	233.9
Altera FFT 处理器	4 096	16	94	36.0
文献[3]的处理器	4 096	16	130	16.0
文献[6]的处理器	16 384	16	130	221.0
本文的处理器	16 384	16	100	73.8
	65 536	16	100	337.5

6　结束语

本文设计并实现了一种高性能并行 FFT 处理器，采用 4 个蝶算单元，将处理速度提高为单蝶形处理器的 4 倍。对于 N 点 FFT，每一级蝶形运算所用时间仅为 $N/16$ 周期。改进的无冲突操作数地址映射方式保证了 4 个基 4 的蝶算单元所需要的 16 个输入数据可以在一个时钟周期内从存储器中读取到，并且同时将 16 个蝶算输出保存到存储器中。本文设计的处理器针对的是 16 384 点和 65 536 点 FFT 处理，对于其他 $N=4^M$ 点的 FFT 处理，只要配置相应参数即可。但是，并行处理程度的提高导致占用的逻辑资源和系统功耗相应增加，如何根据系统的实际需求找到一个平衡点是下一步的研究方向。

参考文献

[1] 胡广书. 数字信号处理理论、算法与实现[M]. 北京: 清华大学出版社, 1997.

[2] Johnson L G. Conflict Free Memory Addressing for Dedicated FFT Hardware[J]. IEEE Trans. on Circuits and System, 1992, 39(5): 313-316.

[3] 邓　波, 戎蒙恬, 汤晓峰. 可配置高速高精度 FFT 的硬件实现[J]. 计算机工程, 2006, 32(17): 254-256.

[4] 段小东, 顾立志. 高性能基 4 快速傅里叶变换处理器的设计[J]. 计算机工程, 2008, 34(24): 238-240.

[5] Andrake R. A Survey of CORDIC Algorithms for FPGA Based Computers[C]//Proc. of FPGA'98. Monterey, USA: [s. n.], 1998.

[6] 傅　博, 李　栋, 谢应科. 一种高速定点 FFT 处理器的设计与实现[J]. 计算机工程, 2005, 31(11): 52-54.

编辑　顾姣健