

动态可重构 FPGA 布局算法

张寒¹, 吴岩松², 赵森³, 孟成栋⁴

(1. 北京南瑞智芯微电子科技有限公司, 北京 100192; 2. 大庆油田有限责任公司第一采油厂, 黑龙江 大庆 163001;
3. 大庆油田矿区服务事业部物业管理一公司, 黑龙江 大庆 163453; 4. 空军装备研究院, 北京 100085)

摘要: SRAM(Static Random Access Memory)型FPGA凭借其动态结构调整的灵活性等特点,被广泛应用于工业领域。针对动态可重构功能单元的布局问题,分析了模拟退火解决方案的局限性,提出了基于电路分层划分和时延驱动的在线布局算法。算法首先按最小分割原则将电路划分为一定数目的层,然后按自顶向下的原则在芯片的每一层中布局划分出的层,同时保证电路关键路径的延时最小。实验结果表明,所述算法在时延、线长和运行时间方面均优于VPR算法。

关键词: 大规模集成电路; 动态可重构; FPGA; 布局

中图分类号: V271.4; TN47 **文献标志码:** A **文章编号:** 1671-637X(2014)04-0077-04

A Layout Algorithm for Dynamically Reconfigurable FPGA

ZHANG Han¹, WU Yan-song², ZHAO Sen³, MENG Cheng-dong⁴

(1. Beijing NARI Smartchip Microelectronics Company Ltd, Beijing 100192, China; 2. No. 1 Oil Plant of Daqing Oil Field Company Ltd, Daqing 163001, China; 3. No. 1 Property Management Company of Daqing Oil Field and Mine Services Division, Daqing 163453, China; 4. Air Force Equipment Research Institute, Beijing 100085, China)

Abstract: Due to their flexibility in dynamic structural adjustment, Static Random Access Memory (SRAM) based Field Programmable Gate Arrays (FPGAs) are widely applied in industry and other fields. Aimed at the layout problem of modules on the reconfigurable functional unit, a hierarchical circuit partitioning-based timing-driven placement algorithm was proposed based on analysis to the limitation of the simulated annealing algorithm. The circuit was first divided into limited number of tiers based on principle of minimum cut, and then the layout of each tier was implemented according to top-down principle. The algorithm can also attain the minimum critical path delay. Experimental results show that: compared VPR algorithm, the proposed algorithm achieves better results in delay, wire-length and runtime.

Key words: LSIC; Dynamic Reconfiguration; Field Programmable Gate Array (FPGA); placement

0 引言

随着大规模集成电路的迅猛发展,尤其是可编程逻辑门阵列FPGA的出现,为动态地改变电路功能和结构提供了硬件基础,从而在工业界引发了对可重构系统研究的热潮^[1-2]。在以FPGA(Field Programmable Gate Array)为基础的可重构系统中,电路的构建是通过硬件描述语言来完成的,硬件信息可以像软件程序模块一样被动态调用^[3-5],从而可以在系统逻辑电路资源相对既定的条件下在时间轴方向上对资源实现分

时复用,这极大地扩展了FPGA的应用范畴,提高了配置的灵活性。

然而,在实际应用中动态可重构技术还存在诸多问题需要解决。其中,最基本的问题是如何对可重构逻辑功能单元进行布局^[6],而现有布局算法的主要目标几乎都是提高布局的质量和缩减布局花费的时间,缺少对可重构结构单元及动态可重构的评价。本文首先分析了传统布局算法,并以此为基础建立了动态可重构模型,从而提出了一种基于电路分层划分和时延联合驱动的布局算法,将电路在Z维度上分层并通过线性布局实现重新组织各层,达到在线实时布局的目的。仿真实验的结果显示,与传统布局算法相比,本文所述方法在时延、线长和运行时间上分别减少了25%、33%和42%,具有一定的应用价值。

收稿日期:2013-06-04

修回日期:2013-06-25

作者简介:张寒(1982—),女,黑龙江哈尔滨人,硕士,工程师,研究方向为动态可重构系统的设计与测试。

1 FPGA 布局算法

1.1 FPGA 基本结构

以 Xilinx 公司的 FPGA 岛型 (island style) 结构芯片为例,对 FPGA 的布局算法进行阐述,其典型结构如图 1 所示。每一个逻辑单元块被布线轨道包围,输入/输出单元分布在 FPGA 的四周,数字功能由每个可编程逻辑块完成,并利用可编程布线资源连接这些逻辑块的输入/输出端口,从而形成更大的逻辑功能电路,同时每个输入/输出单元也通过可编程布线资源与逻辑块相连^[7]。FPGA 的可编程特性是通过 3 种方式来实现的^[8],分别对应于利用 SRAM 单元来控制传输门、多路选择器和三态缓冲器。

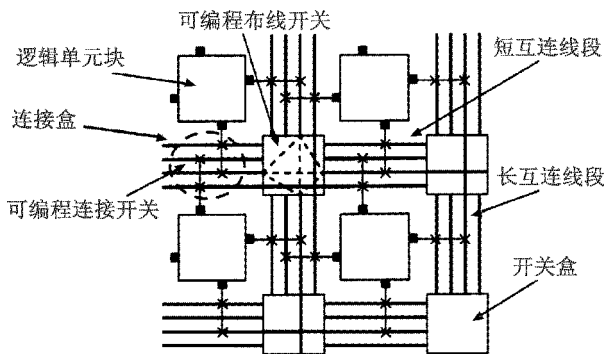


图 1 岛型 FPGA 结构图

Fig. 1 The architecture of island-style FPGAs

1.2 FPGA CAD 流程

在 FPGA 逻辑功能设计中,一个电路是通过将成千上万个可编程单元和配置位设置成合适的状态而实现的。通常情况下,采用高级的抽象描述语言来实现电路功能,并利用计算机辅助软件将其转化为可编程的位流文件,从而配置 FPGA 中可编程单元的状态。其设计流程如图 2 所示,分别为逻辑综合、工艺映射、布局和布线 4 大部分,其中,布局布线算法在整个设计过程中占主导地位。

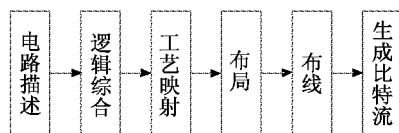


图 2 FPGA CAD 流程

Fig. 2 Flow chart of FPGA CAD

1.3 FPGA 布局算法

布局算法的功能是决定每个逻辑单元块的物理位置。布局模块首先读入逻辑单元块的网表和需要布局的器件信息,然后通过布局算法来决定电路中不同逻辑块在 FPGA 芯片中的分布,从而为布线过程提供输入。

布局算法通常是一个不断优化迭代的过程,其优化的直观目标通常是将相互关联的逻辑单元块放在一

起,从而尽可能地减少和缩短连线^[4-5]。在布局过程中也需要综合考虑布局的面积和线路延迟等其他因素,以获得较快的电路速度,常用的布局算法有数学分析方法,基于最小分割的布局算法和模拟退火算法等。目前,基于模拟退火的布局算法由于其在有限的时间内可以得到相对最优的布局结果被广泛使用,很好地解决了 FPGA 的布局问题^[9]。

模拟退火算法通过在模拟金属冷却的过程中逐渐产生高质量金属结构的特点实现逻辑单元的布局^[10],布局器在第一次布局时随机分配逻辑单元块和输入/输出块在 FPGA 芯片上的物理位置,然后不断地随机交换逻辑单元块和输入/输出块来迭代改善布局结果,在每次随机交换中采用代价函数来衡量交换的优劣。如果交换降低了布局代价,则接受此次交换;反之,此次交换仍然存在一定的机会被接受,用于防止布局器陷入局部最小化。在 VPR 中采用线性拥塞的代价函数,该代价函数可以在合理的时间内产生最佳的效果,其函数表达形式为

$$C = \sum_{n=1}^{N_{nets}} q(n) \left[\frac{bb_x(n)}{C_{av,x}(n)} + \frac{bb_y(n)}{C_{av,y}(n)} \right] \quad (1)$$

式中:对于线网 n , $bb_x(n)$ 和 $bb_y(n)$ 分别表示水平和垂直方向上边界盒跨越的距离; $q(n)$ 表示线网大于 3 个终端节点后的边界盒评估校正因子; $C_{av,x}(n)$ 和 $C_{av,y}(n)$ 分别表示水平和垂直方向上线网边界盒的平均通道宽度。

2 可重构计算模型

在可重构系统中,一个电路被划分成许多小的硬件任务,这些任务可以布局到 FPGA 芯片上,在系统中相互独立地并行执行,执行结束后被系统删除,这种部分可重构的能力显著增加了 FPGA 芯片的利用率,从而提高了系统性能。

但是系统资源、任务的调度和任务的布局必须由一个可重构操作系统完成,通常情况下,这种可重构计算系统由可重构功能单元 (Reconfiguration Function Unit, RFU) 管理器、共享内存、操作系统和可重构 FPGA 芯片组成,如图 3 所示。

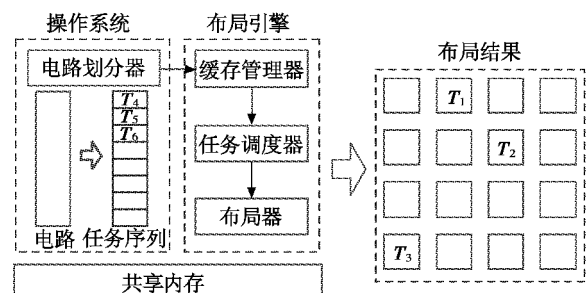


图 3 可重构计算系统模型

Fig. 3 The model of reconfigurable computing system

布局引擎部分由缓存管理器、任务调度器和布局器3部分组成,用于管理FPGA的资源,提供任务的管理和调度等操作;共享内存用于存储任务执行过程中的数据,缓存管理器用于管理任务的配置数据。

3 动态可重构 FPGA 布局算法

如前文所述,传统芯片的布局是将网表描述的电路模块映射到芯片物理位置的过程,以线长为主要目标,而动态可重构的布局问题则与传统布局不同,主要区别为电路模块执行顺序必须符合动态可重构FPGA的约束条件^[11]。本文所述的动态可重构FPGA芯片在空间上被划分成多个层结构,每层用一个由若干个配置内存单元组成的配置内存块来表示一个配置。于是,执行顺序包含任务的调度、配置内存块之间的通信、任务之间不能重叠等。在实际应用过程中,任务的到达时间是无法预知的,是在运行时确定的,所以对动态可重构FPGA的布局是动态进行的。

本文提出的动态可重构在线布局算法流程如图4所示。

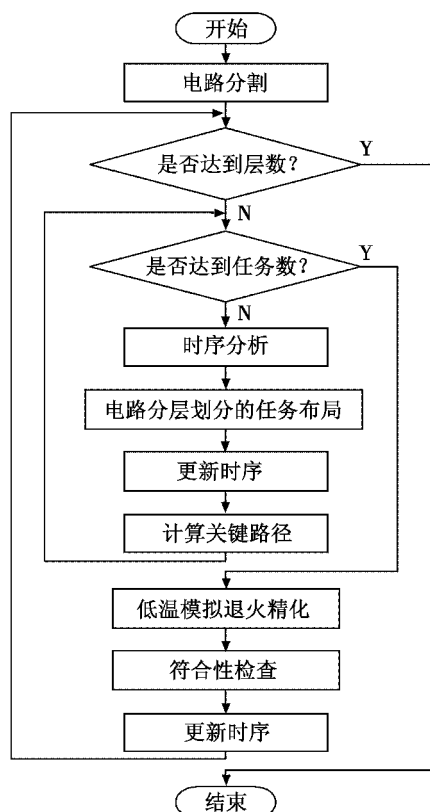


图4 电路分层划分和时延驱动的布局算法

Fig.4 Hierarchical circuit partitioning-based and timing driven placement algorithm

算法首先采用最小分割划分方式将电路划分成细小的任务,进而利用线性布局方法实现线长和各任务之间的分割尺寸最小;其次,进行时序分析得到各线网

的时延情况,以便在布局过程中周期性地更新关键路径的延时,保证时延驱动的正确性;然后从顶层开始,自上而下在每一层中分别进行布局,当任务数达到要求后,停止层的划分,进入低温模拟退火精化布局阶段,用于改善线长和布通率;在每次精化后进行符合性检查和时序更新,当所有的任务划分和层数达到要求后,结束布局过程,得到最好的布局效果。

为了实现电路的有效划分,即线长和各层之间割尺寸最小,本文用图的形式对电路进行标示,设电路为 $G(V, E)$,其中, V 表示线网中的节点, E 表示线网中的边。于是,可以构造 m, n 阶EV矩阵($m = |E|, n = |V|$),当且仅当对应的节点和边是关联的,矩阵中元素为1,否则为0。同时,设第 i 行带宽 $B(r_i)$ 为该行非0元素之间的距离,那么矩阵带宽 B 为矩阵所有行的最大带宽值,即 $B = \max(B(r_i)), 0 < i < m$ 。

于是,电路的有效划分可以等效为矩阵带宽的最小化问题。尽管带宽最小化问题是NP完全问题^[12],但通过启发式的方法可以寻求到满足线长和割尺寸的解决方案。这种方法的实质就是不断地交换矩阵的行和列,将矩阵的非0元素向对角线移动,最终产生电路节点的线性排列,达到最小化割尺寸的目的;同时在不断地交换行列的过程中,使得矩阵带宽最小,即所有线网的线长最小,得到最终的划分方案。

在电路分割结束后,对每层进行逐层布局,其代价函数同时考虑关键路径和时延,每次布局结束后重新进行时序分析,确保布局算法能够动态周期性地更新。

与传统布局方式不同的是,本文由于对FPGA增加了 Z 方向维度上的划分,即在3个维度进行动态可重构FPGA的布局,于是对应的代价函数需修正为

$$C = \sum_{n=1}^{N_{\text{nets}}} q(n) \left[\frac{bb_x(n)}{C_{av,x}(n)} + \frac{bb_y(n)}{C_{av,y}(n)} + \frac{bb_z(n)}{C_{av,z}(n)} \right] \quad (2)$$

式中, $bb_z(n)$ 和 $C_{av,x}(n)$ 分别表示 Z 方向上边界盒跨越的距离和线网边界盒的平均通道宽度。

4 仿真结果

仿真实验采用MCNC基准电路中的10个电路作为被测电路,仿真实验过程为:首先将高层次的电路描述转换为基本门网表;再将其转换成由逻辑单元块构成的网表,即综合和逻辑单元块打包过程,采用的软件分别为SIS和T-Vpack^[7];然后采用本文所述的算法对此网表进行布局,布局过程中还需要芯片结构描述文件;最终将电路布局到6层FPGA芯片结构上。

仿真实验与传统VPR算法进行对比,分别比较电路平均时延、线长和运行时间,其结果如表1所示。

表 1 仿真实验结果对比
Table 1 Simulation results

电路	VPR		本文			
	Delay/10 ⁻⁷ s	wire	Runtime/s	Delay/10 ⁻⁷ s	wire	Runtime/s
apex2	0.79	22892	15.12	0.52	16236	17.98
bigkey	0.39	23659	32.87	0.21	12802	25.71
clma	1.54	114234	521.41	1.02	78025	236.93
des	0.66	36013	12.76	0.38	14637	12.82
diffeq	0.71	14764	16.92	0.58	9137	16.75
ex5p	0.57	15018	8.71	0.45	9651	9.12
pdc	1.02	68124	87.13	0.68	49961	64.15
misex3	0.52	15369	11.76	0.39	10238	12.85
frise	1.16	43075	95.92	0.99	28757	61.07
ε38417	0.75	59238	270.13	0.85	46106	163.32
均值	0.81	41239	107.27	0.61	27555	62.07
比例	1.00	1.00	1.00	0.75	0.67	0.58

实验结果表明,本文所述算法在时延、线长和运行时间上均优于 VPR 算法,其中时延和线长平均值分别减少了 25% 和 33%,而平均运行时间更是缩短了 42%。

5 结束语

本文对动态可重构的布局问题进行研究,在传统布局算法的基础上提出了一种基于电路分层划分和时延联合驱动的布局算法。仿真实验结果表明,本布局算法和传统的布局算法相比,能够有效地降低平均时延约 25%,平均线长降低约 33%,运行时间降低约 42%。因此,该算法具有一定的应用价值。

参考文献

[1] 彭晓明,庞建民,郭浩然. 动态可重构技术研究综述[J]. 计算机工程与设计,2012,33(12):4514-4519.
PENG X M, PANG J M, GUO H R. Survey of dynamic reconfiguration technology[J]. Computer Engineering and Design, 2012, 33(12):4514-4519.

[2] CARDOSO J M P, DINIZ P C, WEINHARDT M. Compiling for reconfigurable computing: A survey[J]. ACM Computing Surveys, 2010, 42(4):13. 1-13. 65, doi: 10. 1145/1749603. 1749.

[3] CHEN W, WANG Y, WANG X, et al. A new placement approach to minimizing FPGA reconfiguration data[C]//IEEE International Conference on Embedded Software and

Systems, 2008:169-174.

[4] TAN H, DEMARA R F. A physical resource management approach to minimizing FPGA partial reconfiguration overhead[C]//IEEE International Conference on Reconfigurable Computing and FPGA's, 2006:1-5.

[5] ZHAO L, WANG Z, YANG L. Reliability-aware placement and fault tolerant reconfiguration in FPGAs [C]//IEEE 14th International Conference on Communication Technology, 2012:541-545.

[6] LOTFIFAR F, SHAHHOSEINI H S, KHANZADI H. Dependency aware placement in reconfigurable computing systems[C]//The Second IEEE International Conference on Intelligent Systems, Modelling and Simulation, 2011:272-276.

[7] BETZ V, ROSE J, MARQUARDT A. Architecture and CAD for deep-submicron FPGAs [M]. Kluwer Academic Publishers, 1999.

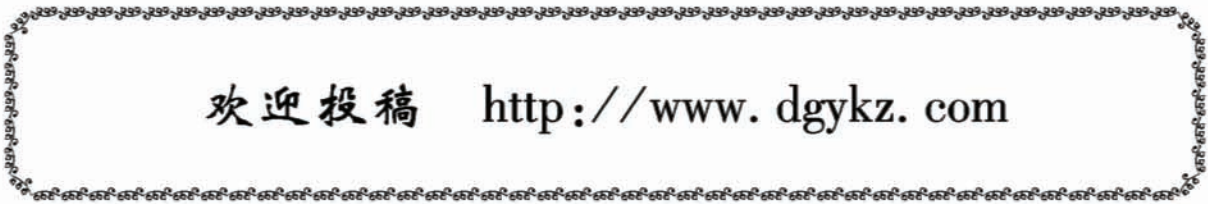
[8] 高海霞. 基于 SRAM 技术的现场可编程门阵列器件设计技术研究[D]. 西安:西安电子科技大学,2005.
GAO H X. Study on design techniques of SRAM-based field programmable gate array[D]. Xi'an:XiDian University, 2005.

[9] BETZ V, ROSE J. VPR: A new packing, placement and routing tool for FPGA research[C]// Field-Programmable Logic and Applications, Springer Berlin Heidelberg, 1997: 213-222.

[10] SANJABI M, JAHANIAN A, AMANOLLAHI S, et al. ParSA: Parallel simulated annealing placement algorithm for multi-core systems[C]//The 16th IEEE CSI International Symposium on Computer Architecture and Digital Systems, 2012:19-24.

[11] WU G M, LIN J M, CHANG Y W. Performance-driven placement for dynamically reconfigurable FPGAs[J]. ACM Transactions on Design Automation of Electronic Systems, 2002, 7(4):628-642.

[12] BAZARGAN K, KASTNER R, SARRAFZADEH M. 3-D floorplanning: Simulated annealing and greedy placement methods for reconfigurable computing systems[C]//IEEE International Workshop on Rapid System Prototyping, 1999:38-43.



欢迎投稿 <http://www.dgykz.com>