

结合用户约束文件的高效多 FPGA 系统分割方法

张倩莉^{1,2}, 于芳², 刘忠立¹, 李艳², 陈亮², 李明², 郭旭峰²

(1. 中国科学院 半导体研究所, 100083 北京; 2. 中国科学院 微电子研究所, 100029 北京)

摘要: 针对单个现场可编程门阵列 (Field Programmable Gate Array, FPGA) 可配置逻辑模块 (Configurable Logic Block, CLB) 和输入/输出 (I/O) 数目受限难以满足大规模复杂电路系统设计需要的问题, 提出了一种结合用户约束文件 (User Constraint File, UCF) 和单个 FPGA 的 EDA (Electronic Design Automation) 设计流程, 通过部分人为干涉, 对多 FPGA (multi-FPGA) 系统进行分割的方法. 应用这种分割方法, 可以提高 multi-FPGA 系统的划分效率, 简化设计的复杂度. 结果表明: 基于这种分割方法的分割工具, 结合 multi-FPGA 系统的硬件结构, 可实现一个最多为 9 芯片 multi-FPGA 系统的设计和配置.

关键词: 现场可编程门阵列; 电子设计自动化; 用户约束文件; 多 FPGA 系统; 分割

中图分类号: TP47

文献标志码: A

文章编号: 0367-6234(2012)07-0144-05

An efficient partitioning method with UCF for a multi-FPGA system

ZHANG Qian-li^{1,2}, YU Fang², LIU Zhong-li¹, LI Yan², CHEN Liang², LI Ming², GUO Xu-fen²

(1. Institute of Semiconductor, Chinese Academy of Sciences, 100083 Beijing, China; 2. Institute of Microelectronics, Chinese Academy of Sciences 100029 Beijing, China)

Abstract: FPGA has fixed number of I/O or CLB, which resulting in a limitation to the implementation of a very large complex system in a single FPGA. This paper addresses a partitioning method used for a multi-FPGA system implementation with partly guidance of User Constraint File and FPGA EDA flow. The partitioning method can improve the partitioning efficiency, and reduce the complexity of a multi-FPGA system design. Based on the proposed method and the multi-FPGA system architecture, a partitioning tool was implemented, which has the capability of designing a nine-chip multi-FPGA system.

Key words: FPGA; EDA; UCF; multi-FPGA system; partitioning

随着专用集成电路 (Application Specific Integrated Circuit, ASIC) 制造成本的日益增加以及 FPGA 在性能方面和 ASIC 差距逐渐缩小, 加上 FPGA 具有市场周期短, 成本低等明显优点, 因而 FPGA 正在成为各种实时设计平台的重要选择. 但是, 利用 FPGA 完成大规模复杂电路系统设计, 仍存在一定的困难. 其难点在于: 与 ASIC 相比, FPGA 的资源是有限的. 由于 FPGA 在输入/输出 (I/O) 和 CLB 的数量上都是固定的, 因此设计者在用 FPGA 作为平台时, 其电路规模受到限制. 然

而如果使用多个 FPGA 去实现大规模复杂电路系统, 上述问题就迎刃而解了^[1].

用户在利用 multi-FPGA 系统实现大规模复杂电路系统时, 电子设计自动化 (Electronic Design Automation, EDA) 成为了设计 multi-FPGA 系统的重要工具. 利用 EDA 工具设计 multi-FPGA 系统时需要参照系统的硬件结构, 因此在设计过程中不能按照传统设计方法那样在系统设计结束之后再设计 PCB 板, 而是要对 multi-FPGA 系统和 PCB 板进行同步设计和修正.

通常情况下, 利用 multi-FPGA 系统实现一个大规模复杂电路系统, 设计流程包含: 综合、分割、全局布局、全局布线、单个 FPGA 布局、单个 FPGA 布线^[2-3]. 其中, 分割步骤中必须考虑如何将分割后的电路分配到 multi-FPGA 系统的各个 FPGA

收稿日期: 2011-07-15.

作者简介: 张倩莉, (1986-), 女, 博士研究生;

于芳, (1960-), 女, 研究员, 博士生导师;

刘忠立, (1940-), 男, 研究员, 博士生导师.

通信作者: 张倩莉, zhangqianli@ime.ac.cn.

中. 分割步骤完成后, 按照所使用的 FPGA 规模和结构, 电路被划分为若干部分. 然后将分割后的若干电路在 multi-FPGA 系统平台上进行全局布局, 之后再通过全局布线进行线路调整 and 连接, 最后进行单个 FPGA 布局和布线, 从而完成 multi-FPGA 系统的设计工作.

multi-FPGA 系统分割算法的基本原则是: 使划分后的逻辑规模小于单个 FPGA 的容量上限, 同时保证划分后各逻辑模块之间信号线数量小于 FPGA 间的物理连线数量^[4]. 目前, 国内外采用的传统分割方法是在综合后的基本逻辑单元层面上进行的, 分割过程在物理层面上考虑了是否满足单个 FPGA 容量及 FPGA 之间信号连线数量等基本原则. 通常的分割步骤主要采用以下算法^[5]:

1) 共享资源分割. 将寄存器阵列的逻辑模块拆分, 然后将这些逻辑模块和其所控制的逻辑模块集成在一起. 这样, 不需要在不同 FPGA 边界之间设置宽的总线, 并且减少了 FPGA 在 I/O 总数上的限制. 2) 重新分组逻辑以减少 FPGA 之间的连接. 3) 复制逻辑以减少 FPGA 之间的连接. 4) 对 FPGA 边界的 I/O 采用时分复用.

上述分割算法存在着一些不足: 没有从电路设计者的设计意图出发, 不能在行为级分析整个电路的功能, 容易在分割过程中将一个具有独立功能的模块拆分开, 导致该模块内部延迟增大. 并且, 采用这种分割方法的后续工作较多, 对全局布局以及全局布线算法的要求很高, 由于以整个 multi-FPGA 系统为单位进行布局布线, 复杂度高, 延迟也不容易控制.

针对上述传统分割算法的不足, 本文提出了一种结合 UCF 的高效 multi-FPGA 系统分割方法. 经过本分割方法的划分以后, 可将 multi-FPGA 系统的设计转化为多个单 FPGA 的设计, 提高 multi-FPGA 系统的划分效率, 大大简化了系统设计的复杂度.

1 单 FPGA 的 EDA 设计流程

利用 FPGA 实现电路的设计需要依托于 EDA 工具, 本文自主开发的 EDA 软件流程如图 1 所示. 电路描述文件经综合、映射、布局布线及码流生成等步骤, 最后将生成的码流文件下载到 FPGA 中, 实现用户所描述的电路功能.

在每个步骤之间通过文件的形式来传递电路信息, 即利用综合工具读取电路描述文件, 经综合生成 edif 格式的文件作为映射工具的输入; 映射工具分析 edif 网表文件, 结合 FPGA 的硬件结构,

将基本逻辑单元 (Basic Logic Element, BLE) 进行打包 (packing)、装箱 (clustering), 完成符合可编程逻辑模块 (CLB) 结构的映射工作, 输出一个 .net 格式文件; 布局布线工具读取映射的输出文件, 从而获得可编程逻辑模块和输入输出模块 (IOB) 的网表信息, 根据 FPGA 的硬件结构进行布局布线, 生成 .p 和 .r 格式的输出文件; 根据布局布线的输出, 结合 FPGA 的可配置结构, 利用码流生成工具生成最终控制 FPGA 配置的码流文件, 经下载接口下载到 FPGA 内完成对 FPGA 的配置, 这样便完成了应用 FPGA 实现电路设计的工作.

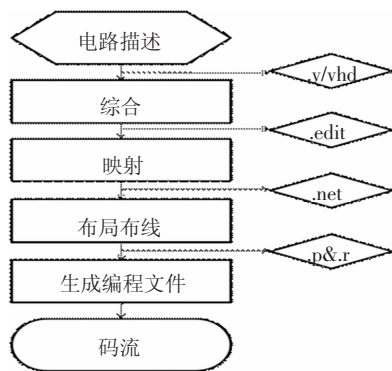


图 1 EDA 工具设计流程

2 结合用户约束文件的分割方法

基于结合 UCF 的 multi-FPGA 系统分割方法, 开发出用于 multi-FPGA 系统的分割工具, 此工具可以集成在自行开发的 EDA 流程内, 也可以与其他 FPGA 软件工具配合使用.

在分割工具中, UCF 文件的约束和指导内容由用户根据自己的设计特点和需求进行编写, 在电路的综合阶段对网表进行模块标注, 分割过程是通过部分人为干涉实现的. 分割后生成多个针对各个 FPGA 的子网表文件, 每个 FPGA 完全按照单个 FPGA 的 EDA 设计流程完成后续工作. 这种分割工具可以提高 multi-FPGA 系统的划分效率, 并简化系统设计的复杂度.

2.1 用户约束文件的内容及功能

用户约束文件是由用户即电路设计者, 按照自己的设计特点和需求, 参照 FPGA 的硬件结构, 对 EDA 设计流程中的部分步骤和内容做出约束和指导的文件.

现阶段也有部分 FPGA 软件采用 UCF 文件, 例如 Xilinx 的 ISE, 但是在它的 UCF 文件中只是对芯片 I/O 引脚和一些时序进行约束^[6]. 本文提出的多 FPGA 分割工具中使用的 UCF 不但可以对系统的结构信息进行描述, 还可以针对逻辑和

连线进行约束.

用户约束文件是 ucf 格式的文件,一部分内容 由用户利用本文提供的 EDA 工具进行编写,其 余部分内容由本文的工具按照一定规则自动生 成. 在用户编写过程中,EDA 工具能提供固定的 格式以及内容限定. 约束 multi-FPGA 系统分割 的 UCF 文件包括下列信息:划分子模块的个数、划 分后每个子模块的名称及对应 multi-FPGA 系统 中 FPGA 的序号、整个电路中的控制信号、电路中 指定要分割的连线、模块之间根据系统结构需要 增加的贯穿连线.

针对其中指定要分割的连线以及贯穿连线部 分,用户只需要把强调需求的内容编写在 UCF 内,分割工具会在初始化遍历网表的过程中分析 所有连线,并把需要进行处理的连线自动补充打 印在 UCF 中.

在 UCF 约束下,multi-FPGA 系统设计的流程 如图 2 所示. UCF 文件可以在设计初期编写,也 可以在 EDA 工具各步骤中,根据生成的中间结果 结合用户需求进行修改并重新载入运行.

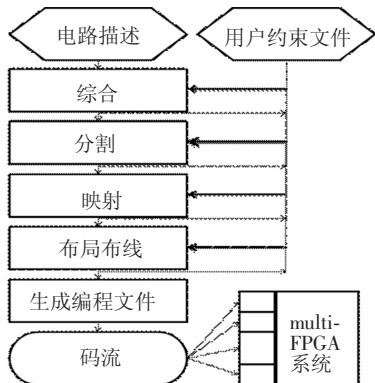


图 2 在 UCF 约束下 multi-FPGA 系统设计流程

在本设计流程中,综合器读取 UCF 的信息, 结合算法对电路进行综合,同时完成模块划分和 标注;UCF 内信息指导分割工具,结合综合结果 对电路进行分割,分割生成 N (N 为 multi-FPGA 系统中 FPGA 的个数)个子网表文件;映射、布局 布线以及码流生成工具分别针对每个子网表文件 继续 EDA 流程,最终生成 N 个配置文件并分别下 载到 multi-FPGA 系统中的 N 个 FPGA 中,完成 multi-FPGA 系统的设计工作.

2.2 multi-FPGA 系统分割的具体实施

multi-FPGA 系统设计的核心问题是将电路模 块化,使每个模块对应 multi-FPGA 系统中每一个 FPGA,而这种模块化通常在分割步骤中实 现^[7-8].

本文提出的分割方法除借鉴传统分割原则 外,还结合用户约束文件进行了部分人为干涉,不

仅从物理层面考虑,还考虑了模块功能的完整性.

综合工具和分割工具都会读取 UCF 文件中 关于 multi-FPGA 系统结构的信息,包括 multi-FPGA 系统由几个 FPGA 组成、每个 FPGA 在系统中 对应的编号及名称、电路的时钟信号名称等信息, 以指导后续工作的进行.

2.2.1 模块的划分与标注

综合过程对模块的划分和标注是分割工具 的主要参照标准,为了更好地还原用户的设计意图, 在用户设计电路时,需要在电路描述中对不同模 块进行标注命名,同时把这些作为标注的名称记 录在 UCF 文件中. 在综合过程中,综合工具将 RTL 级或行为级电路连接转换为门级电路连接, 同时按照电路的功能,参照 UCF 文件中的约束条 件,根据电路的自然边界将电路划分为若干子模 块,并且对子模块中每个基本逻辑单元进行标注 命名,这样综合产生的每一个基本逻辑单元的名 称都包括它所属于模块的名称.

根据用户约束文件的内容及功能所述,用户 可以在得到综合结果后结合自己的需求和目标修 改 UCF 文件和电路描述文件,重新载入综合器, 以得到更适合设计预期的结果,同时可以参照综 合结果在 UCF 文件中添加约束条件.

2.2.2 分割工具的具体算法

分割过程完成了将一个完整电路信息的网 表,分割成若干彼此相对独立的电路网表的工作.

分割工具以综合输出的 EDIF 网表和 UCF 文 件作为输入文件,首先遍历综合输出网表中的每 一个基本逻辑单元,辨识基本逻辑单元的名称,结 合 multi-FPGA 系统中 FPGA 的个数和 UCF 文件 中用户添加的模块标注名称以及 FPGA 的名称和 编号信息,给每一个基本逻辑单元增加一个分割 属性,属性值为该基本逻辑单元应属 multi-FPGA 系统中 FPGA 的编号;接下来遍历综合输出网表 中的每一个连线,首先对 UCF 文件中指定的需要 分割的连线进行分割. 分割工具将其他 UCF 文件 没有明确约束但不属于同一个 FPGA 的两个逻辑 单元之间的连线切断,在断开处分别加上 I/O;最 后,将属于同一 FPGA 的所有逻辑单元和连线进 行整合分类,生成若干个新的,并对应于 multi-FPGA 系统中各个 FPGA 的子网表文件.

在综合结果的网表中会有一部分衍生逻辑单 元由于所属模块不明确而没有标注命名. 分割工 具将通过分析该逻辑单元与相邻有标注逻辑单元 的连接紧密程度来进行分割分配. 具有控制功能 或与其他模块均紧密关联的模块将被分配到主芯

片 (main-chip) 中.

2.2.3 控制信号的分割与分配

分割过程中对时钟信号采用了在单个芯片内复制并在系统中全局控制的方法: 时钟信号是 FPGA 器件内部逻辑模块工作所必须的, 分割工具将时钟信号在各个 FPGA 内部重新生成. 所有 FPGA 的时钟信号由系统时钟控制. 这样, 就解决了信号在 FPGA 之间传递可能引起偏差的问题, 保证了整个 multi-FPGA 系统的时序同步性.

2.2.4 multi-FPGA 系统的配置

multi-FPGA 系统中各个 FPGA 的连接通过 FPGA 的 I/O 管脚和 PCB 板上预置的线路完成. 分割结束后生成 N 个子网表文件, N 为 multi-FPGA 系统中 FPGA 的个数.

在每个 FPGA 中, 将分割后的子网表作为输入, 结合单个 FPGA 的 EDA 设计流程, 完成对此芯片的映射、布局、布线、码流生成及下载.

上述结合 UCF 的 multi-FPGA 系统分割方法优点在于: 紧密地结合了用户的设计理念和需求, 从电路描述阶段开始进行人为的模块标注命名, 实现了人为干涉和设计自动化的结合; 分割思路简单清晰, 避免了在具有独立功能模块内部由于分割产生的不必要的延迟, 并大大提高了系统划分的效率; 分割结束后就把问题从 multi-FPGA 系统的设计简化成了几个对单 FPGA 设计的问题,

避免了算法复杂的全局布局 and 全局布线, 降低了系统设计的复杂度和难度.

3 multi-FPGA 系统的设计实例

利用 multi-FPGA 系统的 EDA 工具设计需要参照 multi-FPGA 系统的硬件结构, 即 PCB 板的设计结构. 图 3 为一个 9 芯片的 multi-FPGA 系统简图. 图中 main-chip 为主芯片, 围绕在主芯片周围的 8 个芯片以顺时针方向分别为 chip0 ~ chip7. 各芯片与相邻芯片之间的信号传输, 通过芯片的 I/O 和 PCB 板上的线路连接直接实现; 非相邻芯片之间的信号传输, 通过在介于这两个芯片之间的芯片中加贯穿连线来实现. 每个芯片的时钟控制端都引出并汇集在系统的时钟控制端, 电路的时钟信号通过系统的时钟控制电路输入, 从而保证系统的时序同步.

在利用 EDA 工具进行 multi-FPGA 系统设计之前, 用户需要按照 EDA 工具规定的格式及电路设计的内容, 设置用户约束文件的内容. UCF 的内容及形式包括: 系统被划分的模块数量, 即例如 FPGA 的个数 9, 每个模块的名称及其在 multi-FPGA 系统中对应的 FPGA 的编号, 例如 chip0 ~ chip7 以及 main-chip, 分割过程中必须分割的连线名称及分割后对应的 I/O, 需要添加的贯穿连线的名称及对应的 I/O.

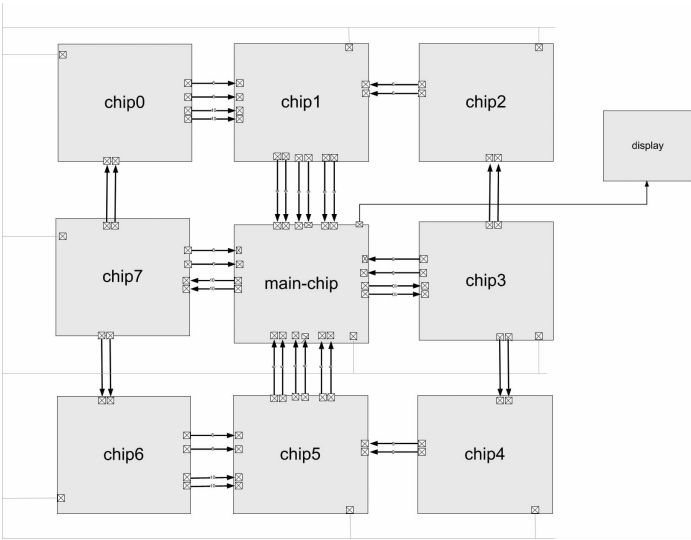


图 3 一个 9 芯片 multi-FPGA 系统简图

利用文中介绍的分割工具, 将一个规模较大的电路分割成了 9 个相对独立、分别对应于系统中 9 个芯片的子网表文件. 分割结果如表 1 所示, 其中 multi-FPGA 系统中 9 个 FPGA 全部使用本文自主研发的基于 SOI 工艺辐射加固的 FPGA VS1000 系列, 每个 FPGA 有 208 个 I/O 引脚以及 576 个逻辑资源. 所用实例名为 pingpong8. v, 内

容包括描述 8 个独立的乒乓球运动轨迹的电路以及系统控制端. 在电路描述中, 对 8 个乒乓球的模块分别命名为 pingpong0 ~ pingpong7, 并在 UCF 文件中做相应的标注. 分割结果显示, 分割工具将具有 3 347 个基本逻辑单元的原始文件按 8 个乒乓球及控制端的划分方式分成了 9 个子网表, 每个子网表所占用的逻辑资源及 I/O 数量满足单个

FPGA 的容量。

然后将 9 个网表文件按照单个 FPGA 设计流程逐个进行映射、布局布线、码流生成等操作,生成 9 个对应于各个 FPGA 的码流文件并下载到相应的芯片中,最终完成了本例 9FPGA 的 multi-FPGA 系统设计。

通过上述方式,就把设计任务由用一个系统实现一个大规模集成电路转换成为用 9 个芯片实现 9 个确定连接关系的电路,从而从基本思路上大大简化了 multi-FPGA 系统设计的复杂性,提高了划分效率。结合 UCF 文件的约束和指导,从用户的设计意图出发,保证电路各模块功能和结构上的完整性。

表 1 对实例 pingpong8 的分割结果

网表	基本逻辑 单元数量	连线数量	I/O 数量	占用 FPGA 逻辑资源
pingpong8 (原始网表)	3 347	4 888	9	
pingpong0 (chip0)	390	553	47	190
pingpong1 (chip1)	396	579	77	194
pingpong2 (chip2)	393	558	53	192
pingpong3 (chip3)	401	581	85	197
pingpong4 (chip4)	395	563	45	193
pingpong5 (chip5)	396	578	75	194
pingpong6 (chip6)	393	576	50	192
pingpong7 (chip7)	396	580	81	194
pingpong8 (main_chip)	202	455	168	96

4 结 论

1)分割过程以功能模块为单位进行划分,使各个 FPGA 的功能清晰独立,并且弥补了在寄存器级划分导致子系统间连线数目过多并容易产生延迟的不足;分割之后的流程完全简化为针对单个 FPGA 进行的设计,相对于以整个系统为对象进行全局布局布线,提高了系统划分的效率,大大降低了布局布线的难度和复杂性;加入用户约束文件的约束和指导,在整个 EDA 设计过程中参照 UCF 文件并实时反馈,可以做到由用户指导自动化设计工具,用户可以在一定程度上根据自己设计思路和需求,对整个系统设计做出指导。本分割方法适用于在电路描述中模块划分清晰、各模块功能定义较明确的大规模电路设计。

2)当用户的大规模集成电路设计模块化不够清晰或者模块之间的联系过于紧密时,利用本

算法分割出的各个子模块之间的连线会比较多,单个 FPGA 的 I/O 数目可能成为限制;贯穿连线的添加也会造成 FPGA 的 I/O 负担。针对这两个问题的对策是:分割过程中加强全局布局的考虑,对子模块的位置信息进行分析和调整,以减少贯穿连线的添加;以模块自然边界划分为基础,对于多连线边界进行拆分重组,适当调整模块归属,以减少 I/O 的使用。

参考文献:

[1] LADDHA S, De KUMAR B. EDA tools and design methodology for multi-fpga designing/prototyping[EB/OL]. [2009-11-16]. <http://www.design-reuse.com/articles/22074/for-multi-fpga-designing-prototyping.html>.

[2] HIDALGO J I, LANCHARES J, HERMIDA R. Partitioning and placement for multi-fpga systems using genetic algorithms[C]//Proceedings of the 26th Euromicro Conference. Washington, DC: IEEE Xplore, 2000: 204-211.

[3] HAUCK S, BORRIELLO G. Pin assignment for multi-FPGA systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1997, 16(9): 956-964.

[4] ARCE-NAZARIO R A, JIMENEZ M, RODRIGUEZ D. High-level partitioning of discrete signal transforms for multi-FPGA architectures[C]//Proceedings of the 2006 International Conference on Field-Programmable Logic and Applications. Washington, DC: IEEE Xplore, 2006: 1-4.

[5] 胡文彬, 吴剑旗, 洪一. 多 FPGA 验证平台引脚限制的解决方案[J]. 合肥工业大学学报(自然科学版), 2010, 33(10): 1519-1522.

[6] 薛小刚, 葛毅敏. Xilinx ISE 9. X FPGA/CPLD 设计指南[M]. 北京:人民邮电出版社. 2007: 173-207.

[7] 宋威, 方穗明, 姚丹, 等. 多 FPGA 设计的时钟同步[J]. 计算机工程, 2008, 34(7): 245-247.

[8] HAUCK S, BORRIELLO G. Logic partition orderings for multi-FPGA systems[C]//Proceedings of the 1995 ACM Third International Symposium on Field-Programmable Gate Arrays. New York, NY: ACM, 1995: 32-38.

[9] SUH Jinwoo, KANG Dong-In, CRAGO S P. A communication scheduling algorithm for multi-FPGA system[C]//Proceedings of the 2000 IEEE Symposium on Field-Programmable Custom Computing Machines. Washington, DC: IEEE Computer Society, 2000: 299-300.

[10] VOOTUKURU M, VEMURI R, KUMAR N. Resource constrained RTL partitioning for synthesis of multi-FPGA design[C]//Proceedings of the Tenth International Conference on VLSI Design: VLSI in Multimedia Applications. Washington, DC: IEEE Computer Society, 1997: 140-144.

(编辑 张 红)