基于 FPGA 的局部动态可重构技术研究

王仪洁 王 烈 许晓洁

（广西大学计算机与电子信息学院 南宁 530004）

摘 要 可重构技术作为嵌入式系统中软硬件结合的设计方法，在可靠性、系统高集成度方面有很大优势。现场可编辑门阵列（Field Programmable Gate Array，FPGA）不仅可以满足这些客观需求，还加强了系统的自适应性，降低了开发成本。文章介绍了动态局部重构的实现方法，并在早期获取部分可重构（Early Access Partial Reconfiguration，EAPR）方法的基础上加以改进。之后使用 Xilinx 生产的 Virtex-ML403 开发板实现整个设计，验证该方法的有效性，保证系统的稳定，在实际应用的实现中有利于对资源有效的管理和合理的利用。

关键词 现场可编辑门阵列；动态局部可重构；早期获取部分可重构；Virtex-ML403 开发板

Study on Local Dynamic Reconfiguration of FPGA

WANG Yijie WANG Lie XU Xiaojie

(*School of Computer*, *Electronics and Information*, *Guangxi University*, *Nanning* 530004, *China*)

Abstract As a combination method of hardware and software in embedded systems design, reconfiguration technology has a great advantage for its reliability and system integration. Dynamic reconfiguration technology of FPGA can not only satisfy these requirements, but also strengthen the adaptivity of the system and reduce costs. In this paper, dynamic local reconfiguration was introduced and the EAPR design process was presented with improvement. Then the entire design was implemented with the Virrtex-ML403 development board produced by Xilinx to verify the validity of the method and guarantee the stability of the system. It is good for efficient management and rational utilization of resources in practical applications.

Keywords FPGA; local dynamic reconfiguration; EAPR; Virtex-ML 403 development board

1 引 言

动态可重构技术是 FPGA 设计中的一种新型设计理念，它可快速实现器件的逻辑重建，为处理大规模计算问题提供一种兼具通用处理器灵活性和 ASIC 电路高速性的解决方案[1]。传统的重构方式是停止正在运行的系统，将整个系统中完整的文件重新下载到 FPGA 芯片中，这样内部所有可编程逻辑单元都会被重新配置，重构的时间和功耗开销太大，无法达到高效低耗的要求。而目前具有动态局部重构技术的 FPGA 芯片提供了灵活的在线编程方法和重组功能，它不需要改变原本的设计就能灵活地实现新的功能，完成局部动态重构。具体方法是将部分比特流文件下载到正在运行的 FPGA 芯片上，进而改变其操作功能。也就是说根据需要通过配置已完成的 FPGA 中的部分比特流文件就可以切换不同的功能，而不改变同一块芯片中其他部分的正常运行。为了用更小的芯片完成更多的功能，降低消耗，节约成本，越来越多的开发商对可重构技术进行了研究。用它实现的容错系统[2]可以提高设备的安全性和可靠性，保证了系统的

基金项目：广西自然科学基金（2013GXNSFAA019339）。

作者简介：王仪洁，硕士研究生，研究方向为 FPGA 设计与开发；王烈，副教授，研究方向为 FPGA 动态可重构技术；许晓洁（通讯作者），硕士研究生，研究方向为动态可重构技术，E-mail：xuxiaojiewin@163.com。

稳定运行。可重构技术以其特有的灵活性和高效性被广泛用于通信、航天、交通、医疗等多个领域。

2　可重构技术设计与实现

兼具硬件高效性和软件灵活性的局部动态可重构技术在利用模块化思想的同时采用了自底向上的设计方法，瞬时改变系统的任务，实现所需要的功能。在 FPGA 设计中主要分为两个大区域：重构区域和静态区域，其中重构区域用来存放动态模块。在整个重构过程中，静态模块部分都不会受到任何影响，只有动态重构模块可以根据当前的需要使用部分比特流文件来动态地改变其数字逻辑系统功能。

2.1　设计方法

Xilinx 公司首先提出了在 FPGA 中实现动态部分重构的方法。目前，使用较多的是 Xilinx 公司的 Virtex 系列，其中芯片本身的控制系统能够执行芯片上其他部分区域的重配置[3]。目前使用的动态局部重构的设计方法主要有四种[4]：基于 Jbits 的动态局部可重构，基于差异的动态局部可重构，基于模块的动态局部可重构以及基于 EAPR（早期获取部分可重构）的动态局部可重构技术。四种方式的使用范围不同，其中第一种基于 Jbits 的方法由于缺少技术支持，现在已经基本不再使用；第二种基于差异的方法不能修改与布线资源相关的配置信息，而且可供改动的空间较小，不利于现在大规模集成器件的发展；基于模块化的方法设计思想独特，在目前的使用较为广泛，但也有其无法克服的局限性；于是基于 EAPR 的局部动态可重构技术成为了当今社会的主流设计方法，Xilinx 也推出了多种产品来支持这种动态重构设计方法。在实际系统的设计过程中可以根据重构粒度的大小，实时性要求以及重构功能的特点等多方面来考虑选取相应的设计方法，完成设计。

2.1.1 EAPR 简介

基于 EAPR 的设计方法是 Xilinx 公司 2006 年提出的一种最新的动态重构设计方法[5]，它在不需要人为修改配置的基础上进一步改进基于模块化设计方法，并成为目前 Xilinx 最为通用的设计方式。基于 EAPR 的设计方法在包含了基于模块化动态重构方法中并行实现的方式以及可直接对比特流操作等特点的同时，又去除了基于模块方法的局限性的缺陷，拥有自身的优势，适合于复杂系统的设计[6]。首先，它允许局部重构区域为任意矩阵的设计方式十分灵活。其次，对全局信号也允许其直接穿越可重构区域，无需使用总线宏就能在分界线两边搭起通信的桥梁。再次，它所使用的总线宏是基于查找表结构的。另，用于传输信息的双向通信的总线宏更密，保证模块间正常快速的通讯。同时这种以粗粒度任务模块为基本重构单位的重构技术将各模块紧密地排布在芯片上，支持新型的 FPGA 器件（如 Virtex-4，Virtex-5 等），用于比较复杂的情况中，保证了大规模、高密度系统的可靠性和稳定性，实现起来也比较简单。

2.1.2 设计过程

实现具体可以分为初始化阶段、模块设计和网表生成阶段、比特流的生成合并、下载四大阶段。在设计初期需要考虑模块的划分，运用基于 EAPR 的思想设计实现流程如图 1 所示，在设计中主要包括顶层模块、静态模块和重构模块这三大部分。实现各个模块的布线布局都被保留，用来保证每个模块的性能。在实现的过程中每个模块单独设计，独立综合，尽可能做到高内聚、低耦合、接口简单，才能高效的完成整个系统。虽然 ISE 软件环境支持同一个工程中可以同时含有 VHDL 和 VerilogHDL 文件，且不会影响设计的完成，但是为了系统易于设计，这里统一使用 Verilog。

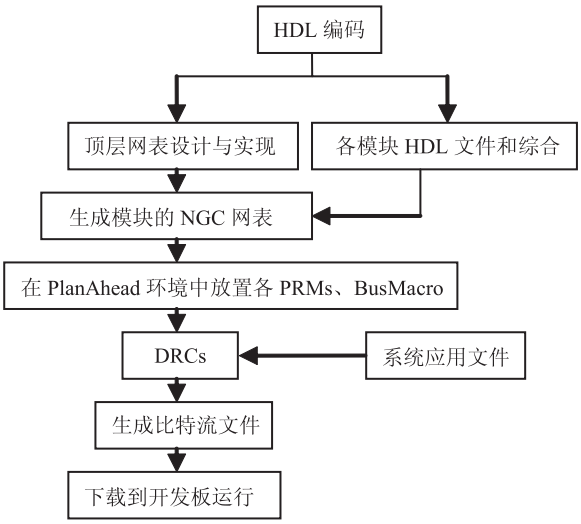

图 1　动态重构实现流程图

设计中首先需要考虑设计描述系统在 FPGA 区域中的划分情况，保证布局布线的合理。其次在实际动态可重构过程中还需要注意以下几个问题：（1）I/O、原始时钟、信号声明，各个模块都要以黑盒的形式初始化在顶层文件中，并且将多个可重构配置模块例化时只例化为一个可重构区域即可。（2）虽然很多模块都可以被重构，但是全局时钟逻辑必需设计在静态

区域中，且不能被重构。同一动态区域中需要被重构的不同模块接口必须相同，所给出的命名也要保持一致。(3)使用基于 SLICE 来实现的总线宏也要事先在顶层设计中例化，再将它用于静态逻辑和可重构区域间的通信，同时在总线宏中要将不必使用的输入引脚接高或低电平。除了全局时钟以外的其他信号中间进行通信时必须通过总线宏，这种总线宏允许 8、16 和 24 位宽的配置。(4)约束文件 .ucf 可以先不在 ISE 中设置，在 PlanAhead 中再添加面积、时间和管脚约束即可，但可以针对相应的器件先设置空的 .ucf 文件，方便后面动态重构的实现。(5)在设计过程中，要确保每个用于下载到 FPGA 设计中的局部动态重构区域的模块的配置文件都必须是完整的，才能真正在系统电路和时间上保持动态连续，且不影响系统的正常运行。使用这样的动态重构方法可以简化设计进程，提高时序性能，并且支持最新的 FPGA 开发板。

2.2 软件中的设计过程

使用 Xilinx 配套的 ISE 设计组件来作为局部动态重构的软件开发工具，用来完成 HDL 设计描述和综合，这些必须在模块激活阶段之前就应该完成。本文使用方便且容易理解的 Verilog HDL 硬件语言来完成 HDL 的设计和综合。采用目前较为流行的动态重构软件 ISE 实现整个设计流程，其中包括静态逻辑和重构区域中的动态重构模块。通常的方法是先完成设计后综合顶层文件，再用同样的方式完成静态模块和各个可重构模块的设计和综合，且在其中不能包含任何时钟逻辑，之后再将生成的网表保存。按照前面所述的设计流程，可以完成动态局部可重构的初始化模块设计。在实际的实现过程中使用具体 paobiao 实例完成整个设计，如图 2 所示，有利于观察分析整个系统。

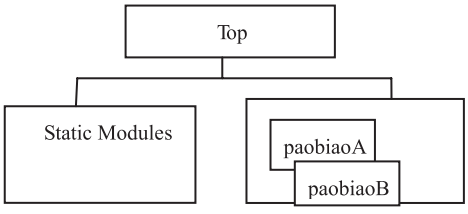

图 2 分层设计图

虽然基于 EAPR 的设计方法较为清晰明确，但是用于实例具体过程的实现中仍然比较复杂，容易出错，考虑实际实现过程中会遇到不同的问题，在设计前期需要做较多的相关工作。所以怎样简化设计就成了需要进一步考虑的问题，同时在具体的实例设计中顶层设计和综合的步骤基本与静态模块设计方法是一致的，而分开设计综合各个部分就加大了整个设计的工作量，所以考虑将静态子模块的设计和综合放入顶层模块来实现。这样做不仅使设计简化，无需做两次相同的工作，更重要的是在顶层例化中也减掉了例化静态模块，采取直接实现的方式，可降低由于例化和实现的复杂所导致出错的概率。并且在实现了顶层模块和静态模块后再实现动态模块，同时要保持综合后的动态模块的网表名要相同。在产生网表的过程中，一个模块只能产生唯一与之对应的网表。接着将生成的各模块网表，包括静态模块在内的顶层网表、动态模块以及总线宏文件都拷贝到相应 netlists 目录下的 TandS 文件目录下。在设计局部可重构区域时，为了便于观察结果，采用设计整列的方法，这样在具备 EAPR 优势的同时简化过程，可以更好的实现和观察布局布线，便于完成在芯片中的实现。

在动态区域中，有 2 个动态模块：paobiaoA 和 paobiaoB，其中 paobiaoA 模块完成数字跑表功能，可以用于百分秒到秒、秒到分钟的计数计时。为了防止紧急情况中 paobiaoA 由于自身状况发生错误，可以设置 paobiaoB 模块完成与 paobiaoA 同样的功能，保证系统不受影响、安全可靠的运行。在此过程中的输入和输出都包含在静态逻辑中，同时用静态逻辑驱动数字计数功能。静态模块和动态区域之间可以共用全局时钟，而其他信号通信使用总线宏实现。但是在实际重构动态模块时需要考虑到配置时间，在减少占用资源的同时缩短重构时间是需要进一步探讨的问题，所以实例中采用重构配置代替重构模块，以及采用重新配置比特流的的方式实现动态重构。初始化设计：

```
TandS.v     //顶层模块初始化，包含静态模块的实现
PRR         //动态可重构区域中模块实例化
  paobiaoA.v and paobiaoB.v
```

使用 PlanAhead 工具可以实现动态可重构设计步骤的自动化，对模块进行物理布局布线、分配区域，并做时钟约束和面积约束相关的设计约束、设置重构区域划分，放置合适的总线宏，完成运行规则检查，实现各模块并将其装配整合。在合并阶段中将单独实现的静态模块和激活的重构模块建立成一个完整的比特流，再整合成为完整的系统。在这个阶段中，系统需要生成后缀为 .bit 的二进制配置流文件，包含芯片和内部的全部相关信息。在这个过程中同时自动优化各个子模块，将不需要的信号去掉，使整个系统的系统得到提高。

2.3　硬件中的实现

软件实现后还需要映射到 FPGA 芯片中，也就是先对网表文件进行设计规则检查，确保无误后映射到目标器件中，将逻辑表示转化为物理表示。整个动态局部可重构系统使用内部访问接口 ICAP 来完成配置，通过 iMPACT 软件将生成的比特流文件按照顺序下载到 FPGA 开发板 Virtex-4 ML403 上，运行系统。利用当前新型的 FPGA 芯片中内嵌有软/硬核处理器，来减少综合布线的时间。在重构过程调用重构比特流需要在第一次调用时就下载完成的比特流配置信息，具体方法是：首先下载固定的 TandS.bit，再下载可重构的 paobiaoA.bit 观察执行情况，完成数字跑表功能；然后假设 piaobiaoA 模块无法正常执行，需要使用动态重构让 paobiaoB 模块代替 paobiaoA 模块动态下载改变模块执行，这是通过静态模块的嵌入式系统完成局部比特流加载，在此过程中静态模块保持运行，仅需调用局部配置文件实现重构。将已经实现的各个配置数据文件比特流用于整个设计，观察系统整体的执行情况，验证其是否能按照原来的方式继续工作来达到预定目标，进而完成整个硬件系统。这种将软件加载到硬件中实现动态局部重构的方法可以随时重构系统，保证系统安全可靠的运行。

3　结果和分析

使用 V4 系列 ML403 XC4VFX12 开发板，参数设置：Package：FF668；Speed：-10；Top-Level Source Type：HDL；Preferred Language：Verilog。

在 Xilinx ISE Simulator 仿真重构前和重构后的部分波形如图 3 所示，重构前完成数字跑表功能，当系统遇到故障时，paobiaoA 受损，无法正常工作，需要瞬时动态重构；这样使用先前预设的 paobiaoB 同样也能正确地完成了数字跑表功能，不影响系统的正常工作。完成验证后观察 FPGA 中资源的使用情况，主要通过所使用的 Slice 的数目来衡量资源的利用情况。动态重构衡量标准主要就是资源和时间两个方面。表 1 给出了单个动态模块 paobiaoA 资源消耗情况，可以看出每个模块占用的资源并不多，采用动态局部可重构就只需要提供 paobiaoA 或者 paobiaoB 占用资源即可。但是如果不使用动态重构的设计方法，而是采用原始方式保证系统正常运行就需要下载所有配置文件，paobiaoA 和 paobiaoB 同时暂用资源数，是现在资源数目的两倍，浪费了系统资源。如果系统比较复杂，模块数目比较多，就有可能会导致 FPGA 芯片中的资源不够使用，无法及时正确的完成系统相应的功能。例中使用所提出的局部动态重构方式对芯片资源进行合理规划，并且将重构模块放入重构区域中，延迟较小，兼具了硬件的速度和软件的灵活特点，及时保证了系统功能的可靠性，同时可以延长芯片的寿命，体现了该方法的优势。在时间方面，由于使用传统重构方式需要下载较大的文件，所以使用时间 3s 远远比动态重构设计方法使用时间(仅 1s)久。从整体考虑，局部动态重构方式大大节省了资源的开销，也防止的资源不够使用的情况发生，较大地改善

表 1　动态模块资源消耗

资源类型	资源数目
Slice Flip Flops	26
4 input LUTs	28
Slices containing only related logic	16
Bonded	27
BufGMUXs	1

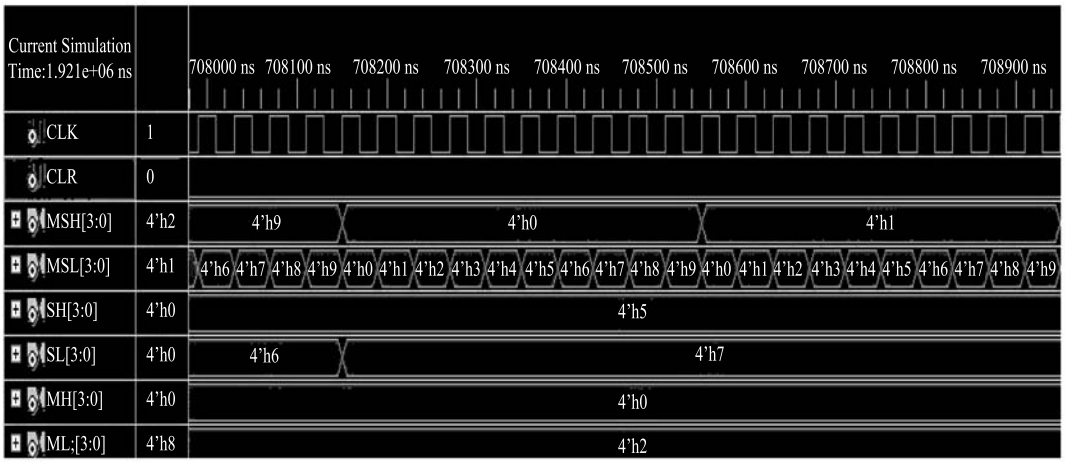

图 3　功能波形图

了在运行时出现突发情况而及时处理的速度，具有明显优势。

4 结束语

目前业界对 FPGA 的研究已越来越深，技术也越发成熟，特别是基于其局部动态可重构特点的探讨，已经逐步从理论研究发展到应用于实际的产品中。本文介绍了一种基于 FPGA 的局部动态可重构的设计方法，使用一个数字跑表的实例验证了该方法的可行性。验证结果表明了动态局部重构的节约资源、时分复用特点，适用于大型复杂的系统设计，对以后应用于容错系统的研究提供了一定的理论依据。

参 考 文 献

[1] Bondalapati K, Prasanna VK. Reconfigurable computing systems [J]. Proceedings of the IEEE, 2002, 90(7): 1201-1217.

[2] Gericota MG, Alves GR, Silva ML, et al. DRAFT: an on-line fault detection method for dynamic and partially reconfigurable FPGAs [C] // Proceedings of the 7th International On-Line Testing Workshop, 2001: 33-36.

[3] 薛建伟, 张杰, 关永. 基于 EAPR 流程的动态局部可重构实现 [J]. 计算机工程, 2010, 36(23): 252-254.

[4] 吴凤艳. 基于 FPGA 的动态局部可重构系统研究 [D]. 广西：广西大学, 2010.

[5] Xilinx Inc. UG208: Early access partial reconfiguration user guide [EB/OL]. (2006-03-16). http://www.xilinx.com.

[6] Xilinx Inc. Early access partial reconfiguration user guide UG208(v1.1) [EB/OL]. (2007-03-01). http://www.xilinx.com/xInx/xil_entry2.jsp?sMode=login&group=prealounge.