

基于 Virtex-II Pro 系列 FPGA 的动态部分可重构系统设计 设计与实现

赵远宁, 吴强, 邹祎

(湖南大学计算机与通信学院, 湖南长沙 410082)

E-mail: zhaoyuanning@126.com

摘要: Xilinx Virtex-II Pro 系列 FPGA 具有支持局部重构的特点。在 XC2VP30 平台上利用 FPGA 局部重构技术实现了动态可重构系统。在平台内嵌的 PowerPC 处理器控制下, 通过内部配置访问通道 (ICAP) 对 OPB 总线上的 IP 模块进行动态重构。采用了 slice 总线宏实现重构模块与静态模块之间的通讯。系统实现了硬件资源的分时复用, 有效的提高系统资源利用率。

关键词: Virtex-II Pro; 动态可重构; OPB 总线; IP 模块; slice 总线宏

中图分类号: TP391.7

1 引言

可重构计算技术是当前热门的研究领域, 它是指数字系统制造完成以后, 其硬件结构可以根据需要重新配置的技术, 包括全部重构和部分重构两种设计技术^[1]。其中, 部分重构能够改变系统部分功能的同时不影响系统其余部分的工作。部分重构根据在配置过程中是否需要中断程序的运行又分为静态重构和动态重构。比较而言, 动态部分重构是在程序运行过程中进行, 能实时改变硬件的配置, 以使系统在不同时刻能够完成不同的功能, 具有节约硬件资源和增强系统灵活性的优点。

目前, 最适合进行可重构技术设计的平台是基于 SRAM 的 FPGA。SRAM 型 FPGA 具有可以反复多次编程的优点。只需在系统上电时, 给 FPGA 加载不同的配置数据, 可以完成不同的硬件功能。

Xilinx Virtex-II Pro 系列 FPGA 正是基于 SRAM 的。Virtex-II Pro 系列 FPGA 是基于查找表的, 内部包括可编程逻辑块 (CLB、IOB), SRAM, 乘法器模块, 时钟管理模块 (DCM), 用于实现动态重构的是一个 ICAP 接口, 在嵌入式的微处理器控制下, 从 PC 或者片上存储器中读取配置数据写入该 ICAP 接口即可完成芯片的动态重构^[2]。

Xilinx 公司提供了关于动态部分重构芯片的详细文档, 但动态部分重构系统还需要用户根据实际需要自行设计并实现。本文选择在 Xilinx Virtex-II Pro XC2VP30 平台上利用 FPGA 局部重构技术实现了一个动态部分可重构系统, 对如何利用内部配置访问通道 (ICAP) 对 OPB 总线上的 IP 模块进行动态重构作了详细的介绍, 其中重构模块与静态模块之间的通讯采用 slice 总线宏实现。后文的组织如下: 第二部分是系统总体结构及设计流程的介绍, 第三部分介绍了动态重构实现的具体细节, 第四部分是实验及分析, 最后第五部分对全文进行总结。

2 系统的总体结构及设计流程

动态可重构计算系统的一种体系结构是由一个或者多个软核微处理器与其它一些专用处理模块一起集成到可重构逻辑模块里面;专用处理模块通过共享总线或者一些特殊的内部网络连接和微处理器核通信。微处理器和专用处理模块可能用到一些片上内存作为局部缓存;另外,由于可编程逻辑器件能够支持动态的重构,那么上述专用处理模块能够被动态的改变,所以需要相应的控制和管理组件来处理动态添加或删除的专用处理模块,以及重构之后内部的通信等问题。这种紧密耦合的体系结构图如图 1 中所示,其中专用硬件模块可以是 IP 核,也可以是自行设计的硬件模块,用虚线框表示说明专用处理模块可动态改变。

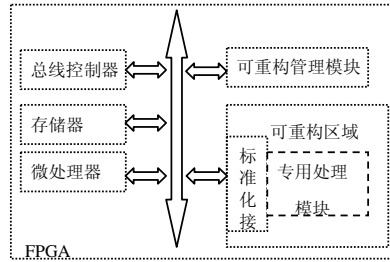


图 1 系统结构图

完成对系统结构的分析,我们在Xilinx Virtex-II Pro XC2VP30的具体平台上来实现动态可重构系统。图2为本文所设计的一个示例动态可重构计算系统的结构图。该系统是IP核的有效集成,其中包括微处理器、总线、BRAM、串行通信模块、通用输入输出模块(GPIO)等。通过Xilinx的开发工具EDK,可以根据设计需求添加相应模块,构建动态可重构计算系统。

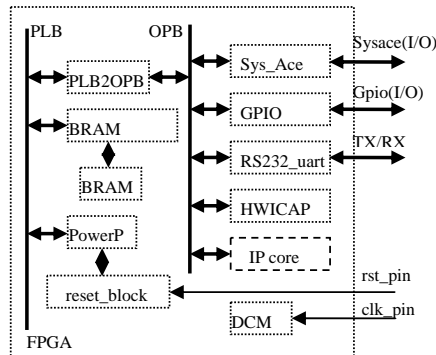


图 2 EDK 生成的具体结构图

在 EDK 生成的结构图中,对应图 1 的系统结构图,其中 PowerPC 为系统图中所示微处理器单元,存储器单元则包括 BRAM 以及 BRAM 控制器;SYS_ACE 用于管理重构过程及重构模块的配置数据;GPIO 属于系统图的可重构管理模块,负责组件的控制和管理;最后我们所说的专用硬件处理模块在示例中就是 IP 核,它可被动态改变。

实现动态部分重构存在两种设计方法。一种是基于模块设计,另一种就是基于差分设计^[3]。相对而言,模块设计具有并行设计和实现各模块,各模块间可最大程度上互不干扰,提高系统开发效率;各个模块有独立的优化目标;改变某些模块功能时,不影响其他模块的设

静态模块之间的信号就是对应IP核的OPB总线端口信号。如图4所示：

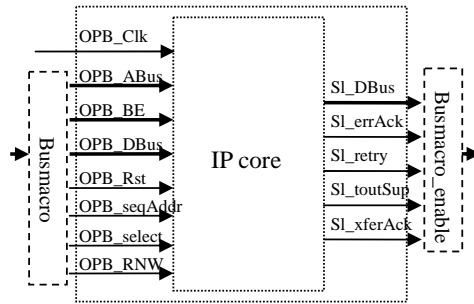


图4 重构模块的结构

从图4可见，除了时钟信号，重构模块和静态模块间的信号通信都应通过总线宏实现。

3.1.2 静态模块的设计

静态模块是系统设计中的固定逻辑模块。在重构过程中，模块功能不改变，并全程负责控制重构操作的完成。本文的静态模块结构如图5所示：

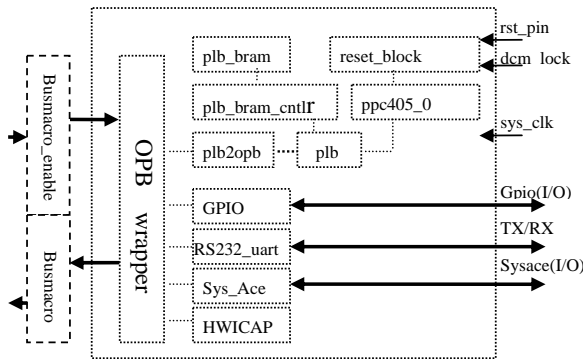


图5 静态模块的结构

图5中，系统在PowerPC微处理器（ppc405_0）的控制下，利用ICAP（HWICAP）向重构单元写入配置文件来完成重构操作。

静态模块为了实现与重构模块的通信，需在静态模块中添加与重构模块对应的端口信息，以保证模块间的信号完整性，见表1。

表1 静态模块中与重构模块对应的端口信息

输出信号	输入信号
OPB_Rst	S1_DBus[0..31]
OPB_ABus[0..31]	S1_errAck
OPB_BE[0..31]	S1_retry
OPB_DBus[0..31]	S1_toutSup
OPB_RNW	S1_xferAck
OPB_select	
OPB_seqAddr	

其中，OPB_开头的是静态模块向重构模块发送的信号，S1_是重构模块返回给静态模块的信号。

由于全局时钟资源（DCM和BUFG）被放置在顶层模块中，因此，静态模块还需为模块内部的各IP核提供统一的时钟端口信号。

```
sys_clk : in std_logic;
dcm_lock : in std_logic;
```

其中，dcm_lock是reset_block的驱动信号。

带格式的：两端对齐，缩进：首行缩进：4字符，行距：多倍行距 1.25 字行

带格式的：两端对齐，行距：多倍行距 1.25 字行

3.1.3 总线宏

模块设计中，模块间的通信至关重要。系统中，不同模块间的通信方式不一样。静态模块间的通信可以通过简单的直接路由来实现。但对于重构模块与静态模块或者重构模块间的通信则需要固定的布线资源，原因在前文3.1节中已经解释。

基于FPGA的动态部分重构系统开发中，使用的总线宏主要有基于三态缓冲的总线宏和基于slice的总线宏。

基于三态缓冲的总线宏具有结构简单、实现方便等优点。Virtex-II Pro系列FPGA传统的动态部分重构系统开发中，就是使用基于三态缓冲的总线宏来实现模块通信的。但是，它信号传输量小，同时它只能控制总线宏信号的传输方向，并不能控制信号的断开和连接，因此必须在静态模块中添加特定的控制模块，阻止配置过程中来自重构模块的无序信号，只接受完成配置后来自重构模块的信号。

基于Slice的总线宏利用了FPGA内部的预留连线来实现模块之间通信的通路，通过增加专门的使能信号来控制信号的传输，与基于三态缓冲的总线宏比较，它能够更好的控制重构过程中来自重构区域的状态不稳定信号进入静态模块，有效的保证了重构的顺利完成。因此，本文采用基于slice的总线宏。

本文中使用了基于slice总线宏的两种形式，这里着重介绍带使能信号的总线宏。在Virtex-II Pro系列FPGA中，仅支持水平方向的总线宏，即总线宏的方向只能从左至右或者从右至左。同时，由于包含在静态模块中完成配置工作的内部配置访问通道（ICAP）位于XC2VP30的右下方^{[4][5]}，静态模块需约束在FPGA的右边。因此，重构模块通向静态模块信号的总线宏结构如图6所示：

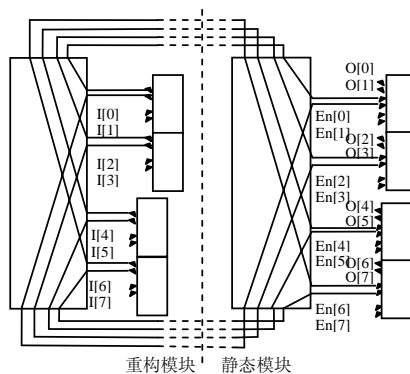


图6 带使能信号的基于 Slice 的总线宏结构^[6]

图中所示的是基于 Slice 的 8-bits 总线宏。其中，I[7:0]和 O[7:0]分别是信号的输

入端口和输出端口，En[7:0]是控制端口，控制通过总线宏信号的连接与断开，处于静态模块中，保证重构模块在配置过程中，重构区域发射的信号不通过总线宏影响静态模块的运行。使用总线宏进行通信时，某些位不需要时，在控制端口相应位赋值 1，即可切断重构模块向静态模块传输的信号。

在顶层模块中，图6所示总线宏的HDL实现如图7所示：

```
component busmacro_l2r_enable is
port (
    input0 : in std_logic;    input1 : in std_logic;
    input2 : in std_logic;    input3 : in std_logic;
    input4 : in std_logic;    input5 : in std_logic;
    input6 : in std_logic;    input7 : in std_logic;
    enable0 : in std_logic;   enable1 : in std_logic;
    enable2 : in std_logic;   enable3 : in std_logic;
    enable4 : in std_logic;   enable5 : in std_logic;
    enable6 : in std_logic;   enable7 : in std_logic;
    output0 : out std_logic;  output1 : out std_logic;
    output2 : out std_logic;  output3 : out std_logic;
    output4 : out std_logic;  output5 : out std_logic;
    output6 : out std_logic;  output7 : out std_logic
);
end component;
```

图7 使能信号总线宏的HDL

用户只需将重构模块通向静态模块方向的信号以及指定的使能信号赋予相应的端口即可。

3.1.4 约束文件的设定

FPGA 的设计过程中，通过附加约束可以有效控制逻辑的综合、映射、布局和布线，减小逻辑和布线延时，提高系统工作频率；同时也可以根据 FPGA 设计平台上静态时序分析工具获得映射或布局布线后的时序分析报告，从而对设计的性能做出评估。

动态局部重构的设计过程中，除了常用的时间约束以外，系统还必须为模块定义相应的区域约束以及总线宏的 LOC 约束。区域约束在 FPGA 上规划各个模块的实现区域，通过物理布局布线约束，完成模块化设计。本文使用的约束包括 AREA GROUP, AREA GROUP RANGE, MODE 约束。

AREA GROUP, AREA GROUP RANGE, MODE 3 种约束，重构模块和静态模块都必须具有。AREA GROUP 是为了避免重构模块和静态模块之间产生共有逻辑。AREA GROUP RANGE 则是定义重构模块和静态模块的硬件资源区域，包括 slice 和 RAM 块两种。

其语法格式为：

```
AREA_GROUP "AG_PRregionA" RANGE
           =SLICE_(minX)(minY):SLICE_(maxX)(maxY);
```

对于实验平台XC2VP30, minX必须满足 $4*n(n=0, 1\cdots 39)$ slice, (maxX-minX+1)需满足 $4*n(n=1, 2\cdots 40)$ 。

```
AREA_GROUP "AG_PRregionA" RANGE
    =RAMB16_(minX)(minY):RAMB16(maxX)(maxY);
```

参数(minx, minY), (maxX, maxY)的取值取决与slice参数的取值, 必须将slice范围内所有的RAM块资源都包含在内。

对于定义了AREA GROUP, AREA GROUP RANGE的模块, 必须同时定义MODE约束。

其语法格式为:

```
AREA_GROUP "AG_PRregionA" MODE=RECONFIG;
```

LOC 约束 所有的 I/O 端口, DCM、BUFGs 以及总线宏定义都要设定 LOC 约束。尤其对于总线宏, 必须设定适当的位置以连接重构模块和静态模块。另外, 总线宏只可放置在重构模块的左右边, 但不可放在上面和下面。总线宏的放置不仅决定系统重构的成功, 还对于系统的时间性能也有一定的影响。

对于实验平台 XC2VP30, 基于 slice 的总线宏 LOC 约束的语法为:

```
INST "bus_macro" LOC = "SLICE_X aYb" ;
```

在总线宏满足同时连接重构模块和静态模块条件下, a 和 b 需为偶数, 且小于重构区域的范围。

3.2 系统的实现

经过合理的模块划分后, 就可以开始进入硬件的模块化设计阶段, 同时通过加载应用程序来完成完整的系统。其详细流程图如图 8 所示:

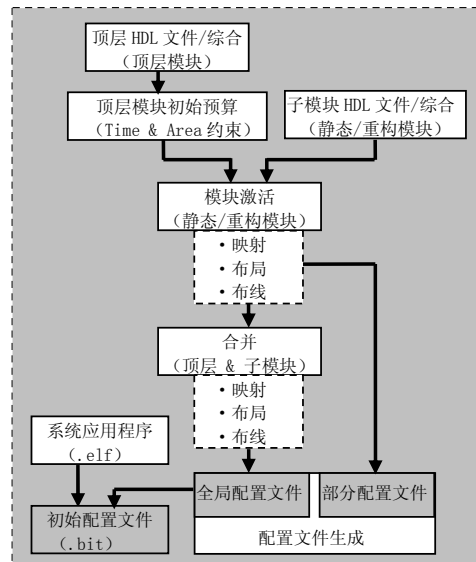


图 8 系统实现流程图

图中, 模块 HDL 文件以及综合阶段是根据划分的模块, 用户分别用 HDL 语言表述各个模

块, 并使用综合工具进行综合, 为模块设计的实现阶段做准备; 初始预算对系统进行全局约束, 确定用户设定的约束条件是否满足硬件要求; 模块激活阶段实现静态模块和重构模块的布线工作; 合并阶段将单独实现的静态模块和重构模块合并成一个完整的系统, 使系统能最终在 FPGA 中运行; 配置文件阶段生成系统所需的各类配置数据文件。

4. 实验及分析

实验使用的硬件平台是由 Xilinx 公司大学计划提供的 XUP Virtex-II Pro XC2VP30 实验板。平台的软件开发环境是 Xilinx Platform Studio 9.1i 和 ISE 9.1。

实验以加密算法 DES^[7]和 128 位分组对称加密算法 AES^[7]为动态重构应用研究对象, 将它们以基于寄存器的实现方式封装成 IP 模块, 并挂载在 OPB 总线上。用户通过调用 IP 模块对应的硬件函数^{[8][9]}即可实现不同的加密功能, 对应的硬件函数分别为 OPB_DES () 和 OPB_AES ()。当应用程序调用硬件函数时, 系统先检查对应的 IP 模块是否已配置在 FPGA 中, 如存在则直接运行; 否则系统先从 Flash 卡中读取相应的部分配置比特流文件到片内缓冲区, 通过配置控制器 ICAP 完成重构配置后再实现硬件函数功能。

通过最终的实验, 表 2 显示了在处理器时钟为 100MHz 情况下, 各模块所占用的 XC2VP30 FPGA 的片内资源情况:

表 2 系统资源消耗

资源类型	资源数						
	AES		DES		静态模块		片内资源
	Total	%	Total	%	Total	%	
SLICEs	8,594	62	4,694	34	1,271	9	13,696
Slice Flip Flops	1,448	5	5,475	19	1,301	4	27,392
4 input LUTs	15,671	57	4,575	16	1,116	4	27,392
PPC405s	0	0	0	0	1	50	2
RAMB16s	0	0	0	0	65	47	136
ICAP	0	0	0	0	1	100	1

从表中可以看到, AES 占用了芯片 62% 的资源, DES 占用 34%, 静态模块占用 9%, 其总资源需求量超过了 XC2VP30 FPGA 可提供的资源总数, 所以如果想使得系统中同时运行 AES 和 DES 两个 IP 模块, 不采用动态重构技术是无法在单片的 XC2VP30 FPGA 芯片上实现的。

当然, 占用资源少是以一定的时间开销为代价的。这个时间开销主要是动态可重构模块的配置时间。对于待处理数据量小, 运行时间不长的应用, 重构配置时间很可能会显著增加系统的整体运行时间; 但是当待计算数据量很大, 系统运行时间较长时, 重构配置的时间将远小于重构模块的运行时间。

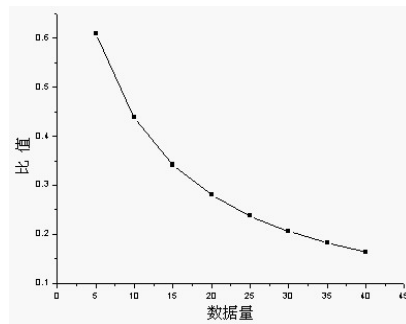


图 10 DES 模块重构开销占系统总运行时间的比值随加解密数据量变化的情况

在实验中, XC2VP30 FPGA上, DES的配置需时为1.22秒。图中所示为DES模块的重构配置开销随着数据量大小变化所占的比值图, 数据量表示处理数据的个数。易见, 随着数据量的增大, 重构配置开销在系统的总的运行时间比变小。

5. 结论

本文对动态部分可重构系统的设计与实现进行了研究, 讨论了开发过程中模块划分、模块间通信及硬件实现等三个问题的解决办法, 总结了基于模块的动态部分可重构系统设计流程。实验中设计了包含两个重构模块, 加密算法 DES 和 128 位分组对称加密算法 AES, 的动态部分可重构系统, 并在 XC2VP30 平台上根据所研究的设计流程和方法进行了实现。实验结果表明, 系统有效地实现了动态部分重构, 达到了分时复用 FPGA 硬件资源, 提高了硬件资源利用率的目标。

但是, 在实验过程中, 重构配置开销稍大, 因此如何有效减少在重构过程中的配置开销对系统总体性能的影响, 将成为我们下一步的研究工作重点。

参考文献

- [1] 黄海鹰, 常青, 卢焕章. 可重构计算技术[J]. 电子技术应用, 1998, 5: 4-5
- [2] 覃祥菊, 朱明程, 张太镒等. FPGA 动态可重构技术原理及实现方法分析[J]. 电子器件, 2004, 27(2): 277-282
- [3] Xilinx Inc. Two Flows for Partial Reconfiguration: Module Based or Difference Based[J], Xilinx Application Note XAPP290 v1.1 Nov 2003
- [4] Xilinx Inc. OPB HWICAP (V1.00.B), [J], DS280 April 4. 2005
- [5] Xilinx Inc. Xilinx University Program Virtex-II Pro Development System. [J], UG069 (v1.0) March 8, 2005
- [6] Xilinx Inc. Early Access Partial Reconfiguratino User Guide for ISE 8.1.01i [J] March 6, 2006. 37-42
- [7] IP cores for DES and AES. [EB/OL]. <http://www.opencores.org>
- [8] Wei Wang, Qiang Wu, Wei Xie. Hardware-Software Co-design for Dynamic Reconfigurable Computing with Collaborative Supports of Architecture and Operating System. [J]. Proceedings of the 11th International Conference on Computer Supported Cooperative Work in Design. Melbourne, Australia, April, 2007
- [9] Wei Xie, Qiang Wu, Wei Wang. An Architecture and Programming Framework for Dynamic

Reconfigurable Computing Systems. [J]. 9th Joint Conference on Information Sciences. OCT 8-11, 2006. KAOHSIUNG CITY, TAIWAN, ROC.

Design and Implementation of Dynamical Partial Reconfigurable System Based on Virtex-II Pro FPGA

Zhao Yuan-ning, Wu Qiang, Zou Yi

(College of Computer and Communication, Hunan University, Changsha, China 410082)

Abstract

Xilinx Virtex-II Pro FPGAs support active partial reconfiguration. A partial reconfiguration system is realized on XC2VP30. PowerPC embedded in the XC2VP30 dynamically reconfigures IP modules which are connected to on-chip peripheral bus (OPB) by internal configuration access port (ICAP). Communications between reconfigurable and static modules are implemented with slice bus macro. By using partial reconfiguration, it allows multiple design modules to time-share physical resources, improving the utility of FPGA hardware resources.

Key words: Virtex-II Pro; Dynamical Reconfiguration; on-chip peripheral bus; IP module; slice bus macro

作者简介: 赵远宁, 男, 1982 年生, 硕士研究生, 主要研究方向是可重构计算。