

支持软硬件透明编程的可重构资源管理器的实现

焦 铭^{1,2}, 李仁发¹, 彭日光¹, 李 浪^{1,2}

(1 湖南大学 计算机与通信学院, 湖南 长沙 410082; 2 衡阳师范学院 计算机科学系, 湖南 衡阳 400044)

摘 要: 设计并实现了一种支持可重构软硬件透明编程的操作系统资源管理器, 它通过系统调用封装了底层硬件细节, 让程序设计者以透明的方式开发应用程序. 实验表明可以减少硬件函数的配置等待时间, 提高软硬资源利用率和系统性能.

关键词: 动态可重构系统; 软硬件透明编程; 资源管理器

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2010)09-0133-05

Implementation of Reconfigurable Resource Management Support for Hardware and Software Transparent Programming

JIAO Ge^{1,2}, LI Ren-fa¹, PENG Ri-guang¹, LI Lang^{1,2}

(1 College of Computer Science and Communication, Hunan University, Changsha Hunan Province 410082, China;

2 Department of Computer Science, Hengyang Normal University, Hengyang Hunan Province 421008, China)

Abstract: This article is designed and implemented a transparent support for programming reconfigurable hardware and software operating system resource management, and system calls by the underlying hardware details of the package, so that process designers to develop a transparent manner applications. Experiments show that the function can reduce the hardware configuration of waiting time and increase utilization of hardware resources and system performance.

Key words: Dynamic Reconfigurable System; Software /Hardware Transparent Programming; Resource Management

1 引言

在透明编程模型的基础上, 设计了一个支持编程模型的原型系统, 在移植的 Linux 实时操作系统内实现了原型系统的可重构资源管理器, 该资源管理器对硬件函数和可重构资源进行管理, 上层的应用开发可以直接调用操作系统内可重构资源管理器提供的接口来管理系统资源^[1-2].

2 可重构软硬件透明编程模型

硬件透明编程, 是指用户(应用程序开发人员)能够在不需要关注底层硬件实现细节的情况下, 通过调用硬件函数来使用可重构硬件. 而且使用的方式在形式上与使用其它软件函数库中的函数一样,

应用程序开发人员看到的只是硬件函数库中的硬件函数. 因此底层硬件实现细节对程序开发人员编写代码是透明的, 也就是硬件透明编程.

对于软硬件编程模型, 可以采用建立软硬件协同函数库的方式为统一的软硬件函数描述提供支持, 如图 1 所示. 对于关键的硬件函数声明, 采用与软件函数相同的语法, 这样可以把软硬件函数统一到相同形式的函数声明上, 形成一个抽象的软硬件协同函数库, 设计人员可以直接调用这个库中的函数, 而不关心具体的实现.

对于软硬件函数的定义, 则采用不同的代码, 以便运行时可动态链接到不同的实现上. 软件函数可以采用该函数本身的实体代码, 编译时会生成与平常软件无异的指令流. 硬件函数则提供访问可重构

收稿日期: 2009-06-30; 修回日期: 2009-09-05

基金项目: 国家自然科学基金项目(60673061); 国家“八六三”计划项目(2007AA01Z104)

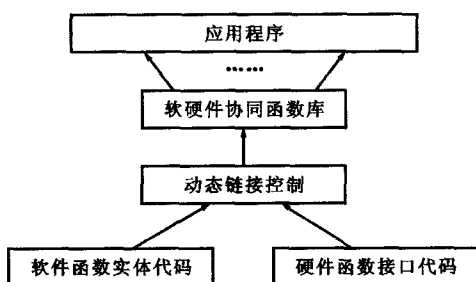


图1 软硬件协同函数库

资源上硬件模块的接口代码,主要是通过可重构资源管理器获得硬件模块在系统存储空间中分配的端口地址,然后把输入、输出参数传送到硬件模块相应端口上,并启动硬件模块。程序运行时,由动态链接控制负责将对协同函数库的调用与软硬件函数的实现链接起来。动态链接控制通过动态链接器调用软硬件划分算法,将应用程序中对抽象函数的调用映射到具体的软件实体代码或硬件接口代码上。

编程模型如图2所示,应用程序主程序通过调用软硬件协同函数库来完成特定的可能需要硬件加速的计算任务,软硬件协同函数库中函数通过操作系统提供的系统调用接口检查并完成硬件加速器的配置或解除配置。可重构资源管理器管理资源链表,记录可重构资源的使用状况,文中的重点工作就是实现可重构资源管理器。

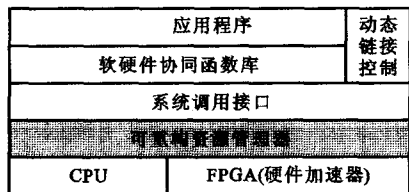


图2 透明编程模型实现框架

3 系统模型设计

在透明编程模型基础之上,参考文献[3]和文献[4],构建整个可重构系统的系统模型,如图3所示。可重构系统的任务统一调度工作通常由调度器、放置器和加载器共同完成。调度器的工作是将到达系统的任务按照一定的调度策略分配给系统各个处理单元进行计算。可重构资源管理器管理系统中的可重构资源,为划分到硬件执行的任务寻找合适的放置位置和分配硬件资源,并负责已完成任务的资源回收工作。可重构资源管理器对硬件任务链表以及可重构资源进行更新,以反映可重构资源的动态变化。加载器负责调度到可重构器件上执行的硬件任务的配置工作,并将配置后的数据加载到共享存储

器中。共享存储器被用来存储每一个任务的配置数据和任务的执行结果。

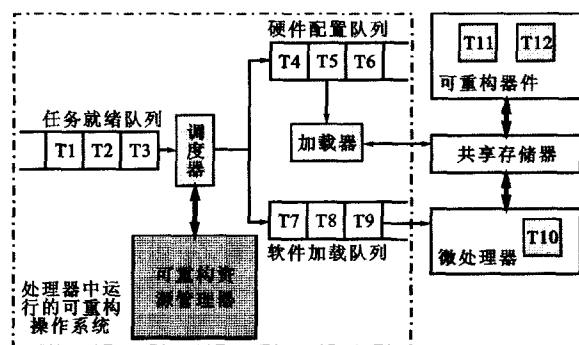


图3 系统模型设计

当可重构操作系统请求一个任务执行时,首先将其加入任务就绪队列中,然后调度器根据采用的软硬件划分算法,决定任务是由软件执行还是由硬件执行。如果由软件执行,按照通常的软件任务的策略将其加载到微处理器执行即可。如果任务队划分到硬件执行,其处理过程参考文献[5-6],如图4所示。

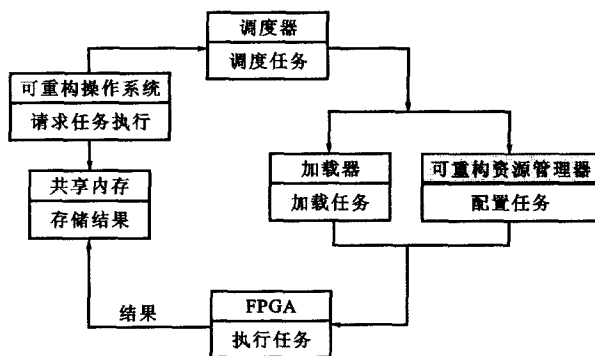


图4 任务处理过程

调度器调用资源管理器来检测是否有可行放置位置,如果存在,则根据特定的任务放置机制从候选的位置中,选择最佳的放置顶点,同时资源管理器负责更新任务链表和可重构资源区域。调度器根据资源管理器返回的结果,如果存在可行放置和调度,则将任务添加到执行链表中或者将任务添加到配置链表中,否则该任务被拒绝。加载器根据调度的情况,如果任务调度成功,则将该任务的配置数据加载到FPGA上,接着硬件任务开始执行,执行完成后,加载器最后将执行结果写入共享存储器中。

4 可重构资源管理器的实现

可重构资源管理器主要负责跟踪系统中硬件函数和可重构资源的状态,以决定新调用的硬件函数是否需要配置,如果需要配置,则为硬件函数分配合

适的硬件资源,并控制可重构控制器实现硬件函数的配置.由于系统同一时刻能够配置多个任务,在对可重构区域进行管理时,需要根据采用的可重构资源管理方法,判断哪个区域的资源可用和够用.操作系统通过动态的维护三个记录硬件函数状态情况的链表,称为硬件函数状态链表,对可重构资源的使用情况进行跟踪和管理,如图 5 所示.通过对硬件函数状态链表的插入、删除、查找等操作,可以使系统获得动态的 FPGA 资源使用情况,完成相应的预配置以及最大限度实现重用已经配置的硬件函数.

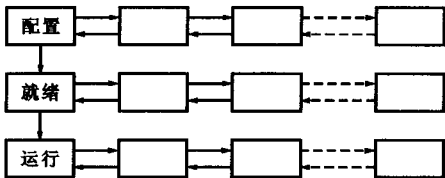


图 5 状态链表

当调用一个硬件函数时,硬件函数接口代码会首先查询这三条链表,如果发现该函数处于配置或运行状态,那么等待其完成.如果处于就绪态,则马上可以启动执行.当在这三条链表中都不能找到该函数时,则表明该函数处于缺失状态,那么就需要调用配置控制器对其进行配置,该函数会进入配置状态.

在系统中了创建三个状态表,分别与这三种状态相对应:

```
LinkList ConfigList, RdyList, RunningList;
CreateList(&ConfigList); /* 正在配置的函数名列表 */
CreateList(&RdyList); /* 配置完成的函数名列表 */
CreateList(&RunningList); /* 正在运行的函数名列表 */
```

系统定义可重构资源控制块 Freeframe_Contrl_Block 来记录可重配置资源的状态.它表示链表中的节点,数据结构如下:

```
typedef struct Freeframe_Contrl_Block
{
    INT16U Begin;
    INT16U End;
    INT16U HFStatus;
} Freeframe;

Begin:指一个硬件函数使用条形可重构资源开始位置的编号.

End:指明这硬件函数使用条形可重构资源结束位置的编号.

HFStatus:表明硬件函数现在所处的状态.
```

FPGA 资源状态描述信息如下:
INT8U FreeframeMaxNum;
FPGA 资源的总条数(按一维模式条形分配资源).

FPGACB FPGA [FreeframeMaxNum;];
用数组来存储整个 FPGA 的资源信息.
INT8U FreeCount = FreeframeMaxNum;
指 FPGA 内处于空闲状态的模块的条数,初始化为最大可重构模块的数目.

硬件函数在 FPGA 上的状态信息如下:
Config:表示硬件函数为正在配置状态;
Rdy:表示硬件函数处于配置完成状态;
Running:表示硬件函数为正在运行状态;
Absent:表示硬件函数在可重构器件上不存在.
硬件函数的状态转换如图 6 所示:

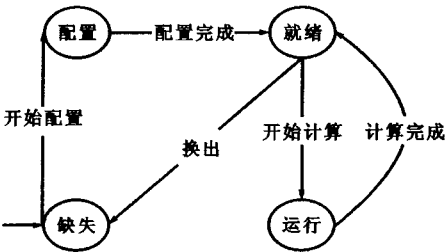


图 6 硬件函数状态图

调用一个硬件函数时,硬件函数接口代码首先要调用可重构管理器,它以硬件函数的名字作为参数传给可重构管理器,可重构管理器要对该硬件函数的状态进行判断,以决定是否对其重构.如果函数正在运行或者配置,则等待其完成;如果硬件任务处于缺失状态,并且可重构资源有足够的空闲区域,则将对任务进行配置;如果任务处于就绪状态,则直接返回配置硬件任务的基地址.硬件任务的调用过程如图 7 所示,由此可见对硬件函数的管理,主要是状态管理与可重构资源管理.

5 系统实验与结果分析

5.1 系统平台的构建

硬件平台是由 Xilinx 公司大学计划提供的 XUP Virtex-II Pro XC2VP30 实验板. XUP Virtex-II Pro 开发板以 XC2VP30-FF896 FPGA 为核心,在此平台上,用户以快速创建和评估软硬件设计. XUP Virtex-II Pro 开发系统的相关软件环境以 Xilinx ISE 和 Xilinx EDK 为主,文中研究使用的软件开发环境是 Xilinx Platform Studio 9.1i 版本,选择的实验处理器是 Power PC405,并在其上移

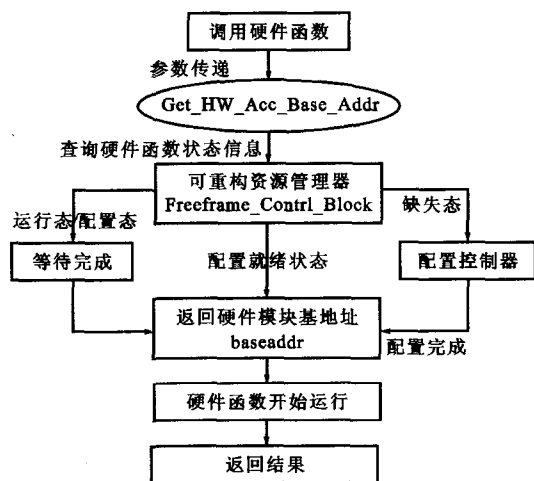


图7 硬件函数调用过程

植 Linux 2.6.24 版的实时操作系统^[7].

硬件平台配置包括:XUP Virtex-II Pro 开发板、ACE CF 控制系统、512MB CF 卡、FPGA USB 下载线、RS-232 连接线等.测试和评估系统使用的基本配置如下:CPU 时钟频率为 300MHz,系统总线为 100MHz,片上存储 128KB.

文中利用 Xilinx Platform Studio 把位流文件下载到实验平台.为了方便查看实验输出结果,使用平台上面的串行输出端口链接到 PC 机上.在 Windows XP 环境下的超级终端中输出结果.

5.2 移植 Linux 操作系统

建立硬件平台:在移植 Linux 之前,我们需要对硬件系统进行基本的配置,这些工作是通过 Xilinx 的 xps 开发平台中的 BSB 步骤完成的.在 BSB 中,选择以下配置:CPU 为 powerpc(300MHz),其中总线频率为 100MHz.其他外设选择:

RS232_Uart_1(opb_uartlite_9600),sysACE_CompactFlash(OPB_SYSACE),DDR_256MB_32MX64_rank1_row13_col11_cl2_5(plb_DDR),plb_bram_if_cntlr(128K),都选中支持中断的复选框.选择一个示例程序,并且生成系统位流文件,最后测试基本硬件系统是否存在问题.

通过配置软件平台并生成相应 BSP,建立交叉编译环境,下载 Linux 内核源代码进行 patch 并配置内核,交叉编译.首先,对 BSP 中 xparameters_300.h 文件进行部分修改,选择相应的体系结构和交叉编译器,对内核进行相应的配置;然后,将之前编译的 BSP 拷贝到源代码树目录中,覆盖原有文件,只需要拷贝 xparameter300.h 这个文件即可;最后,进行交叉编译.

生成 ACE 文件:与 Parallel IV 电缆和 XMD 编

程相比,ACE 文件能够更容易在 FPGA 中编程和装载根文件系统.

生成根文件系统:嵌入 linux 启动时需要一个根文件系统,可以以多种方式组织这个根文件系统.嵌入 linux 的文件系统一般包含以下几个目录:/bin,/sbin,/etc,/boot,/dev,/lib,/mnt,/proc,/usr, 同时还要提供一些实用的基本工具,像 ls、cd 等.

将生成的 system.ace 文件拷贝到 CF 卡中,本地引导 Linux,载入运行,在开发板上看到 Linux 的提示符后,说明移植成功.

5.3 测试实例

测试用例采用实验室项目组开发的图形图像函数库,该函数库主要用于空间图形图像元素的建模、显示、缩放和移动等基本计算任务,它主要包括图形元素的建模、图形元素的运算、图形元素的叠加 3 大类.

上述函数库被进行有效的封装后,并配置二进制比特流下载到 CF 卡,根据需要,可从 CF 读取不同的硬件函数比特流文件,将其加载到 FPGA 上执行.

5.4 测试结果

利用 XPS 把位流文件下载到实验平台,存储在 CF 卡中.系统加电之后,加载操作系统,系统开始运行.在 Windows XP 环境下的超级终端中输出并显示结果.部分显示结果如图 8 所示.

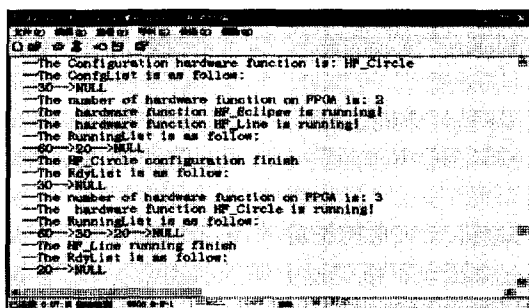


图8 可重构资源管理测试实例图

可以看出,随着硬件函数的不断被调用,可重构资源上的硬件模块越来越多,硬件函数状态链表上的节点也越来越多了.这些节点记录这此硬件函数所处的状态、配置的时间、以及所占据的区域等信息.随着部分任务的完成,相应硬件资源被释放,可重构资源管理模块会进行资源回收,将运行结束的模块所占据的区域置为空闲状态.硬件函数状态链表的节点也相应减少了.

6 结束语

由于当前 Xilinx 公司 XUP Virtex - II Pro XC2VP30 开发板平台的资源限制,所以文中实现的资源管理器是在一维任务模型基础上,对任务进行的管理,实际片上运行的任务较少.随着 Xilinx 公司技术的成熟,这些问题需要进一步的研究和改进.

参考文献:

- [1] Vuletic M, Pozzi L. Seamless hardware software integration in reconfigurable computing systems[J]. IEEE Design & Test of Computers, 2005, 22(2): 102 - 113.
- [2] Vuletic M, Pozzi L. Virtual memory window for application - specific reconfigurable coprocessors[C]//Proc. of DAC. London, 2004: 948 - 953.
- [3] Tabero, Mecha H, Mozos D. A vertex - list approach to 2D HW multitasking management in RTR FPGAs[C]//DCIS 2003. Ciudad Real, Spain, 2003, 545 - 550.
- [4] Juanjo Noguera, Rosa M Badia. HW/SW codesign techniques for dynamically reconfigurable architectures [J].

IEEE Trans. on VLSI Systems, 2002, 10(4): 399 - 415.

- [5] Mitsuru Tomono, Masaki Nakanishi, Shigeru Yamashita, et al. A new approach to online FPGA placement[C]//40th Annual Conference on Information Sciences and Systems. New Jersey Prinveton, 2006: 145 - 150.
- [6] 傅光宁, 葛思攀. 基于 ARM_FPGA 的可重构控制器设计及其在加载系统中的应用[J]. 微电子学与计算机, 2006, 23(7): 28 - 29.
- [7] Windstorm. PowerPC 平台 Linux 的移植[EB/OL]. [2009 - 11 - 10]. <http://www.forwind.cn/2007/08/13/powerpc-linux1/>.

作者简介:

焦 铭 男, (1979 -), 硕士研究生, 讲师. 研究方向为嵌入式计算.

李仁发 男, (1956 -), 教授, 博士生导师. 研究方向为嵌入式计算、无线网络等.

彭日光 男, (1981 -), 硕士研究生. 研究方向为可重构计算等.

李 浪 男, (1970 -), 博士研究生. 研究方向为嵌入式计算等.

(上接第 132 页)

从理论上分析了其检测性能. 仿真结果表明, 该方法在检测性能上与非量化情况下的检测性能非常接近, 同时具有较小计算和通信开销的优势. 在经典衰落信道下的检测性能是下一步研究计划.

参考文献:

- [1] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal. Wireless sensor network survey[J]. Computer Networks, 2008, 52(12): 2292 - 2330.
- [2] 孟中楼, 王殊, 赵峰, 等. 分簇式无线传感器网络睡眠调度机制研究[J]. 微电子学与计算机, 2009, 26(7): 9 - 16.
- [3] 杨晓飞, 吴晓蓓, 黄锦安. 数据融合技术在无线传感器网络中的应用[J]. 微电子学与计算机, 2009, 26(5): 190 - 192.

[4] Zhou G, Stankovic J A, Son S H. Crowded spectrum in wireless sensor networks [C]// Proc. of the Third Workshop on Embedded Networked Sensors. Harvard, 2006.

[5] FCC. Spectrum policy task force. Report of the spectrum efficiency working group[EB/OL]. [2009 - 11 - 10]. <http://www.fcc.gov/sptf/reports.html>

[6] Mitola I, Maguire J, G Q. Cognitive radio: making software radios more personal[J]. IEEE Personal Communications, 1999, 6(4): 13 - 18.

作者简介:

胡富平 男, (1983 -), 博士研究生. 研究方向为认知无线网络.

王 殊 男, (1956 -), 教授, 博士生导师. 研究方向为智能信号处理和认知无线网络.