一种开源微处理器 OR1200 的嵌入式 SoC 设计 *

焦汉明,陈新华,沈国新,方翰华
(山东科技大学 电子信息科学与技术系,山东 青岛 266510)

摘 要:介绍了一种基于 32 位开放源代码的 OpenRISC1200 处理器嵌入式 SoC 的设计方案。系统以 OR1200 为核心,开发基于 Wishone 总线的 IP 核,并集成到 OR1200 系统中,构成了系统的硬件平台。软件部分构建了基于 OR1200 系统的交叉编译环境,开发了系统的启动程序 Bootloader,移植了 μC/OS-II 操作系统。

关键词:OR1200 微处理器;嵌入式系统;Bootloader;μC/OS-II

中图分类号:TP368.1 **文献标识码**:A

Embedded system design of open source microprocessor OR1200

JIAO Han Ming, CHEN Xin Hua, SHEN Guo Xin, FANG Han Hua
(School of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China)

Abstract: This paper introduces the design schedule of embedded SoC based on OpenRISC1200 which is a 32 bit open source code. The hardware platform of the system is as follows. The processor system core is OR1200. And the development is based on Wishone bus IP cores, and then integrated into OR1200 system. The software builds the cross-compile environment, develops the system's bootloader, and transplants ports the μC/OS-II operate system.

Key words: OR1200 microprocessor; embedded system; bootloader; μC/OS-II

高性能的处理器核是 SoC 设计中最为关键和核心的部分。绝大多数 SoC 的处理器都采用了 RISC 体系结构。RISC 处理器具有指令效率高、电路面积小和功率消耗低等特点,满足了 SoC 高性能、低成本和低功耗的设计要求。目前在 SoC 设计中广泛使用的 32 位 RISC 处理器(如 ARM 公司的 ARM 处理器、IBM 的 Power PC 处理器、MIPS 公司的 MIPS 处理器等)均属于商业内核,使用者必须支付相对昂贵的授权费。

近年来开放源代码运动迅速发展,开放性源码的概念已经从软件领域(如 Linux, GCC 等)扩展到了硬件领域,出现了专门发布免费的 IP 核源代码的组织——OpenCores。OR1200 以其完全开放的源代码和编译器吸引了设计者。其结构简单、通用性和可移植性强,具备完整的开发平台,非常适合嵌入式 SoC 设计[1]。关于 OR1200 的研究也受到学术界和工业界越来越多的关注。

1 32 位开放源代码处理器核 OpenRISC1200

OpenRISC1000 系列处理器是开放 IP 核源代码组织 OpenCores 公布的 32/64 位处理器软核。OpenRISC1200 采用了自主设计的 OpenRISC1000 体系结构和自定义的 ORBIS32 指令集。它是一种 32 位、标量、哈佛结构、5 级整数流水线的 RISC,支持 Cache、MMU 和基本的 DSP 功能。OR1200 具有较好的可配置性,使用者可以根据自己的需要定制自定义的指令,确定是否使用 MMU,配置 Cache 的大小(1 KB~64 KB 之间)。OR1200 同时还提供了 1 个用于降低功耗的电源管理单元和 1 个支持片内调试的调试单元[2]。

OR1200 有完善的软件开发环境和操作系统的支持。使用者可以通过包括 GCC、Binutils 和 GDB 等在内的 GNU tool chains 进行基于 OR1200 内核的编译和调试。同时,OR1200 拥有专门的仿真器 Or1ksim,可以进行指令集一级的仿真,并支持 Linux, μClinux, μC/OS-Ⅱ等任

* 基金项目:青岛市科技局 2008 年重点科研项目(项目编号:07-2-3-1-jch)

何一种现代操作系统。

OR1200 典型应用情况[3]：使用 0.18 μm 和 6 层金属工艺时，主频运行达到 300 MHz，可以提供300Dhrystone、2.1MIPS 和 300 Hz 的 32 bit×32 bit 的 DSP 乘加操作的能力。OR1200 默认配置有 100 万个晶体管。

2 系统的总体方案

本文构建以 OR1200 微处理器为核心，包含软硬件平台的嵌入式 SoC 系统。硬件系统以开源 32 位 RISC 核+Wishbone 总线为主干，进行增量迭代开发，将仿真验证过的模块逐个加入到 OR1200 嵌入式系统中，然后在 FPGA 上进行验证。软件部分进行交叉编译环境的建立、操作系统的移植、应用程序开发并利用驱动程序、函数库，经交叉编译工具最后生成可执行程序下载到内存中。最后在 Altera 的 DE2-70 开发板上验证系统能否稳定运行。系统开发流程如图 1 所示：

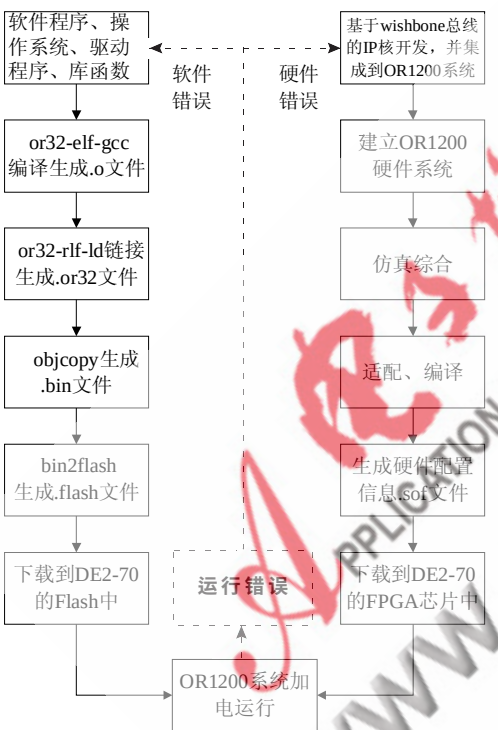

图 1 基于 OR1200 的嵌入式 SOC 开发流程

3 硬件平台的构建

OR1200Q 嵌入式 SoC 系统组件包括：OR1200、Wishbone 互联模块、Memory Controller、Flash、SDRAM、UART、GPIO、DM9000A 接口模块以及 DM9000A 控制器。图 2 所示系统的硬件架构。

系统外接 50 MHz 的时钟，经过 PLL 分频，系统工作在 25 MHz 频率。由于 OR1200 的可配置性，配置了硬件乘法功能，选择 8 KB 的 IC(指令缓存)、8 KB 的 DC(数据缓存)，选择 NO_DMMU，NO_IMMU，即未加内存管理单元。起始地址设为 0x100，对系统做了硬件映射，复位时，0x04000000 地址映射到 0x0 地址，所以系统可以从

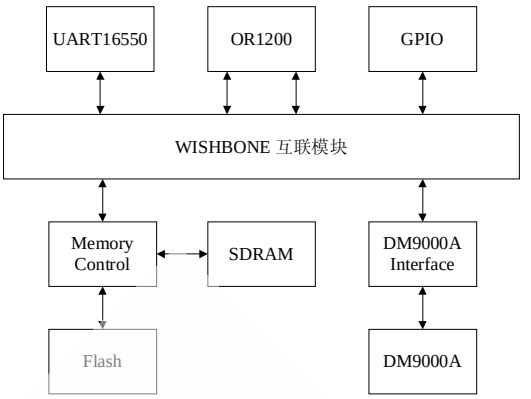

图 2 系统的硬件架构

Flash 启动。Memory Control 为统一的存储控制器模块，控制着 SDRAM 和 Flash 2 种存储器。本系统针对 AlteraDE2-70 开发板硬件配置选择 64 MB 的 SDRAM 和 8 MB 的 Flash。Flash 用于存储压缩的内存映像文件及启动程序 Bootloder。Bootloder 把压缩的内存映像文件从 Flash 复制到 SDRAM 中，并跳到起始位置解压执行可执行文件、启动操作系统。UART 是嵌入式中重要的 I/O 设备，本系统选择由 OpenCores 组织维护的 UART16550 IP 核，它与国家工业标准兼容并且可配置。UART 通信协议的要求是：波特率 9600 b/s、8 个数据位、1 个停止位、无奇偶校验位。GPIO 在系统中主要用于控制 LED 显示状态。DM9000A 支持 8 bit 和 16 bit 数据接口以适用于不同的微处理器，但是没有合适的开源 IP 模块来连接 DM9000A 和 Wishbone。因此需要开发 1 个 IP 接口模块能将 Wishbone 从设备信号转换为 DM9000A 控制信号。这个接口模块有 2 个接口：1 个连接在 Wishbone 从设备上；1 个连接 DM9000A 控制器。DM9000A 本身是 little-endian，OR1200 是 big-endian，所以数据线接法应该做交叉接法。

Wishbone 互联模块有很多种，比如 Wishbone Conbuse、Wishbone Traffic Cop、Wishbone Interconnect Matrix。鉴于本系统外设不多，选择 Wishbone Traffic Cop，它有 8 个主设备、9 个从设备。系统 2 个主设备即指令总线主设备、数据总线主设备，主设备通过 Wishbone 的总线译码器和仲裁器与从设备进行数据通信，总线仲裁器选择控制总线的主设备，保证每个周期最多只有 1 个主设备获得总线控制权。系统有 4 个从设备，分别是 UART、Flash、SDRAM 和 DM9000A。每个从设备都由 32 bit 地址线的最高 8 bit 来编址，总线译码器对主设备发出的地址的最高 3 bit 进行译码来决定访问哪个从设备。

4 系统的软件开发

软件开发过程包括交叉编译环境的搭建，Bootloader 的设计、μC/OS-II 的移植和应用程序开发等。

4.1 GNU 交叉编译环境的构建

OR1200 系统使用的是 OR32 工具链，它由 GCC，

GNU Binutils 和 GDB 组成，另外还提供了 OR32 模拟仿真工具。gcc-3.4.4 用于程序的编译；Binutils-2.16.1 提供了链接等各种工具；gdb-5.3 用于程序的调试。OR32 模拟仿真工具 or1ksim 工具可以模拟 OR1200 处理器的行为，让程序脱离处理器在模拟工具上运行，从而实现 OR1200 与程序并行开发。OR32 可从 opencores 网站上下载。

4.2 启动代码 Bootloder 的设计

4.2.1 移植分析

对于计算机系统来说，从开机到操作系统启动需要一个引导过程，嵌入式系统同样离不开引导程序，这个引导程序就是 Bootloader。通过这段小程序，可以初始化硬件设备、建立内存空间的映射表，从而建立适当的系统软硬件环境，为最终调用操作系统内核做好准备。

结合前面构建的 OR1200 嵌入式 SoC 系统，具体的移植过程中需要修改或编写以下代码：

spr_defs.h OR1200 相关寄存器的设置；

board.h 系统验证所用的 DE2-70 开发板的定义，启动参数等；

flash_boot.S，修改初始化代码和入口；

flash.ld，代码在 flash 中的地址空间安排；

mc.h 存储控制器的初始化；

uart.h UART 的初始化；

copier.c 下载的程序从 Flash 拷贝到 SDRAM 执行的拷贝代码；

在 Makefile 中添加 Bootloder 的编译支持。

4.2.2 启动分析

0x100 处是复位入口地址，但是 Flash 的初始地址为 0x04000000，通过设置寄存器对系统做了硬件映射。复位时，0x04000000 地址映射到 0x0 地址，所以可从 Flash 启动。系统加电后，CPU 从该地址开始执行。该地址也就是系统异常处理向量表第 1 项（复位）。跳到了 init_mc 中，此时完成了系统的大部分初始化，例如设置寄存器、CPU、SDRAM、Flash。然后把 Flash 中的拷贝代码（copier）拷贝到 SDRAM，跳到 SDRAM 中的 Copier 处，copier 负责把 Flash 中的用户程序（userprog.bin）拷贝到 SDRAM 中去，并跳到 0x100 处，解压执行用户程序，启动操作系统。此时的内存映射为：SDRAM 为 0x00000000～0x04000000，Flash 还是 0x04000000～0x08000000。Bootloader 的启动如图 3 所示。

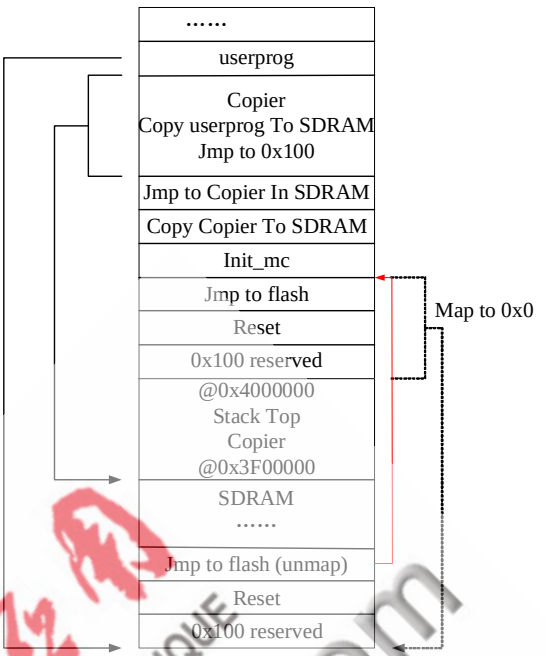

图 3 Bootloader 的启动

4.3 实时操作系统 μC/OS-II 的移植

μC/OS-II 是一个代码公开、内核精简、实时性强、支持多任务的操作系统，非常适合嵌入式系统开发。μC/OS-II 是由与处理器相关的代码，与处理器无关的代码及与应用程序相关的代码组成。它在设计之初就已经充分考虑了可移植性，所以 μC/OS-II 在不同处理器上的移植是比较容易的，主要是编写与处理器相关的代码，即 OS_CPU_A_ASM、OS_CPU.H、OS_CPU_C.C。因此对于 μC/OS-II 的移植可以参考文献[4]中第 13 章，明确 OR1200 微处理器的数据声明类型、OR1200 微处理器支持的堆栈增长方向、临界区处理方式。

5 系统运行测试

系统通过验证操作系统移植的正确性来测试所构建的 OR1200 嵌入式 SoC 系统能否正常运行，编写 main.c 实现 3 个任务以及任务切换。函数的主要功能包括完成 μC/OS-II 操作系统的初始化、硬件资源的初始化、创建相关任务和启动 μC/OS-II 操作系统这几部分。main.c 的主函数部分如下：

```
void startup(void)
{
    initns16450();                    //初始化 or1200 系统
    putstr("Hello world! \n");
    OSInit ();                        //初始化 μC/OS-II

    OSTaskCreate (task1,(void*)&data1, (OS_STK*)
(stack1+STACK1_SIZE),
    TASK1_PRIO);
    OSTaskCreate (task2,(void*)&data2,(OS_STK*)
(stack2+STACK2_SIZE),
    TASK2_PRIO);
    OSTaskCreate (task3,(void*)&data3,(OS_STK*)
(stack3+STACK3_SIZE),
    TASK3_PRIO);
                                      //创建 3 个任务
    OSStart();                        //启动 μC/OS-II
}
```

主函数中3个任务函数非常简单，优先级从高到底分别为1、2、3，系统提供的函数OSTimeDly()分别为10、20、30，OsTimeDly()是将1个任务挂起，延时若干个时钟节拍，CPU可以去执行其它任务。

图4 OR1200系统运行测试结果

图4给出的是通过串口工具输出μC/OS-II多任务调度的信息，3个任务根据优先级和延迟时间正确执行并打印出执行信息，测试证明OR1200嵌入式SoC系统能够稳定的运行μC/OS-II。

本文介绍了一种基于OR1200微处理器的嵌入式SoC系统的软硬件设计。系统经测试运行稳定。系统的硬件核心选用了开源的32位微处理器核OR1200，并定位于嵌入式系统，性能良好，也可适应其他的开放源码IP，对于掌握具有自主知识产权和自主创新的处理器具有重大的意义，本系统已应用于青岛市重大科技攻关项目“基于OR1200嵌入式SoC网关集成电路的设计及AVS实现”。

参考文献

[1] 刘军，郭立，郑东飞，等.开放性32位RISC处理器IP核的比较与分析[J].电子器件，2005，28(4)：50-52.

[2] 孙恺，王田苗，魏洪兴，等.嵌入式CPU软核综述[J].计算机工程，2006，32(7)：8-9.

[3] 徐敏，孙恺，潘峰.开源软核处理器OpenRisc的SOPC设计[M].北京：北京航空航天大学出版社，2008.

[4] LABROSSEZ J J.嵌入式实时操作系统uCOS-II[M].邵贝贝，译.第2版.北京：北京航空航天大学出版社，2003.

(收稿日期：2009-05-09)