

U-Boot 在 MicroBlaze 处理器上的移植方法

邓伟^{1,2}, 许扬婧^{1,2}

(1. 通信信息控制和安全技术重点实验室, 浙江 嘉兴 314033;

2. 中国电子科技集团公司第三十六研究所, 浙江 嘉兴 314033)

摘 要:在通信干扰系统中,为了分析干扰效果,通常需要对被干扰的信号进行采集和分析。由于文中所涉及的采集系统使用了 Linux 操作系统,因此需要用 Bootloader(引导加载程序)将操作系统加载到采集系统的程序存储器,使系统投入运行。文中主要研究了该信号采集系统的 Bootloader(U-Boot)的移植过程,针对采集系统所使用的硬件修改了 U-Boot 代码的文件,给出了 U-Boot 在移植过程中的详细步骤和修改要点,并对 U-Boot 启动后的采集系统进行了验证。结果表明,文中所移植的 U-Boot 程序可以正确引导 Linux 操作系统,并且能够提供操作系统所需参数,解决了该数据采集系统的操作系统无法加载的问题。文中的工作不仅为 FPGA 器件在嵌入式系统中的应用提供了一个较好的解决方案,同时文中所述的移植过程为相关的嵌入式系统设计人员提供了一定的参考。

关键词: U-Boot; MicroBlaze; Xilinx; FPGA; Bootloader; SP605 评估板

中图分类号: TN332

文献标识码: A

文章编号: 1673-629X(2012)07-0071-04

Porting Method of U-Boot on MicroBlaze Processor

DENG Wei^{1,2}, XU Yang-jing^{1,2}

(1. Key Lab of Communication Information Control and Security Technology, Jiaxing 314033, China;

2. No. 36 Research Institute of CETC, Jiaxing 314033, China)

Abstract: Usually, the signal acquisition and analysis must be done in order to analyze interference effects in a communications jamming system. Since the acquisition system involved the use of Linux operating system, so need a Bootloader to load the operating system into acquisition system that can be put into operation. It studies the transplant process of the Bootloader (U-Boot) for the signal acquisition system, modifies U-Boot code file for the hardware and gives the detail transplant steps, and verified the transplanted system. Results show that the migration of U-Boot program can correctly boot the Linux operating system, and provide the necessary parameters to the operating system. Thus, this solution solves the problem, which the acquisition system unables to load the operating system. This work not only provides a better solution for the FPGA device in embedded system applications, but also the migration process described in this article provides some reference for embedded system designers.

Key words: U-Boot; MicroBlaze; Xilinx; FPGA; Bootloader; SP605 EVB

1 U-Boot 简介

Bootloader 代码是芯片复位后进入操作系统之前执行的一段代码,主要用于完成基本的硬件初始化,并通过板载存储器或网络完成嵌入式操作系统的加载,其功能类似于 PC 机的 BIOS 程序^[1~3]。由于嵌入式系统的多样性,因此不可能有通用的 Bootloader 程序,开发者需要根据具体的嵌入式系统情况进行移植。嵌入式系统中常用的 Bootloader 有 redboot, U-Boot, vivi 等,其中 U-Boot 是当前比较流行、功能比较强大的 Bootloader。

U-Boot, 全称 Universal Bootloader, 是遵循 GPL 条款的开放源码项目,其源代码可以从 <http://sourceforge.net/projects/u-boot> 获得。U-Boot 支持多种处理器架构,如 x86、ARM、PowerPC、MIPS、Microblaze、NIOS 等;支持 VxWorks、Linux、QNX 等多种嵌入式操作系统的引导;具有较高的可靠性和稳定性;具有丰富的设备驱动代码,如串口、以太网、FLASH、RTC、LCD 等;支持 TFTP、NFS 网络协议^[4,5]。正是 U-Boot 有如此多的优点,它成为了当前在嵌入式系统中应用最广泛的 Bootloader。

2 U-Boot 系统启动流程

U-Boot 实现分为 stage 1 和 stage 2 两部分。依赖于 CPU 体系结构的代码,如 CPU、Cache 等初始化代码

收稿日期:2011-12-05;修回日期:2012-03-10

基金项目:国家“863”高技术发展计划项目(2011AA7071034)

作者简介:邓伟(1978-),男,甘肃兰州人,硕士,工程师,研究方向为嵌入式系统设计、宽带通信信号处理。

都放在 stage 1, 采用汇编语言实现。而其他部分, 如大多数设备驱动程序、命令处理程序和网络通信等放在 stage 2, 并用 C 语言实现。这样的代码组织方式具有更好的可读性和移植性。

MicroBlaze 处理器 U-Boot 的 stage 1 代码放在 arch/micorblaze/cpu/start.S 中, 采用汇编语言, 其主要功能包括:

(1) 定义 U-Boot 程序全局入口地址(_start), 该地址是系统上电后, U-Boot 执行的第一条指令, 这个入口地址必须放在处理器复位后程序指针(PC)指向的地址, 对于 MicroBlaze 来说, 该地址为 0。

(2) 设置 reset、interrupt、hardware exception 等向量。

(3) 初始化堆栈。

(4) 如果 U-Boot 执行地址与复位地址不同, 则将 U-Boot 代码复制到指定的执行地址处。

(5) 调用 board_init 函数, 使 U-Boot 执行 stage 2 阶段代码。

arch/micorblaze/lib/board.c 中的 board_init 函数是 stage 2 的起始点, 也是整个 U-Boot 代码的主函数, 该函数主要完成:

(1) 初始化串口、中断处理器、定时器、Flash、网口等外设。

(2) 初始化输入输出终端。

(3) 进入命令循环, 接收用户从终端输入的命令, 然后执行相应的流程。

3 U-Boot 移植过程

3.1. 硬件平台介绍

文中采用 Xilinx 公司提供的 Spartan®-6 FPGA SP605 评估板作为目标硬件平台。SP605 评估板主要包括一块 XC6SLX45T-3FGG484 FPGA、128M 字节 DDR RAM、32M 字节 Flash、10M/100M/1000M 自适应以太网芯片、串口芯片等^[6], 该评估板功能框图如图 1 所示。

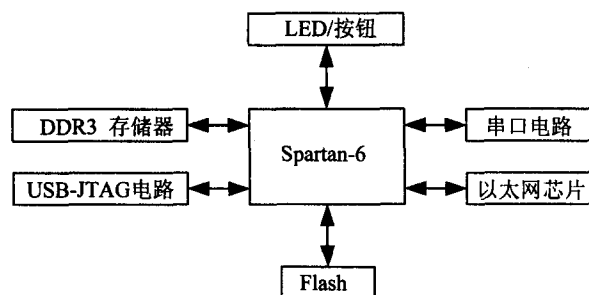


图 1 SP605 评估板功能框图

3.2 SoC 系统的生成

文中利用 Xilinx EDK 开发工具搭建一个包括 MicroBlaze 处理器以及 DDR 控制器、以太网 MAC 协议控制器、串口通信协议控制器、中断控制器、定时器等 IP 核的 SoC 系统作为 U-Boot 程序运行的平台, 所用 EDK 工具版本为 13.2。经过 EDK 中的工具编译综合后, 生成针对 Spartan6 FPGA 的配置文件, 配置后的 Spartan6 内部系统结构示意图如图 2 所示, 其中 MicroBlaze 处理器作为 SoC 系统的控制中心, 承载 U-Boot 代码的运行, DDR2 控制器、以太网控制器、串口控制器、GPIO 控制器作为 SoC 系统的外设接口, 分别驱动 SP605 评估板上相应的部件, 完成内存访问、以太网通信、串口通信、LED 控制等功能。

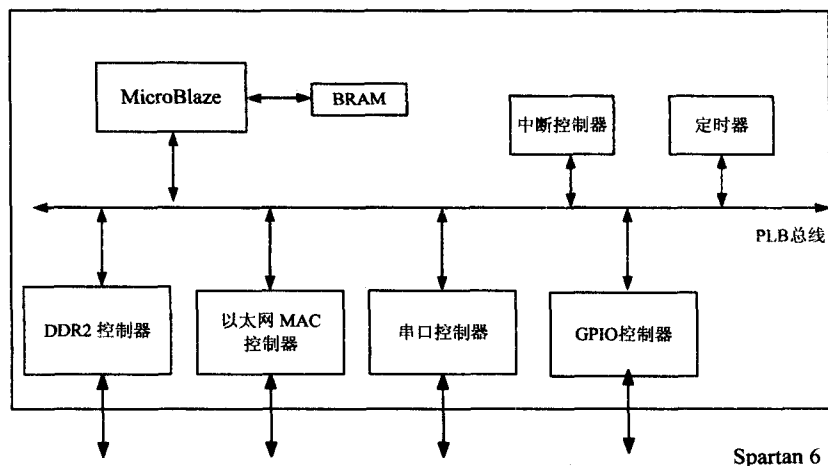


图 2 SoC 系统框图

目前最新的 MicroBlaze 处理器可以加入 MMU 部件^[7], 这意味着当前的 MicroBlaze 处理器不仅支持 uclinux 嵌入式操作系统, 同时也支持 Linux 嵌入式操作系统的运行。为了使硬件平台能有更好的适应性, 文中选择了具有 MMU 部件的 MicroBlaze 处理器, 这样该 SoC 系统不仅可以运行 uclinux 操作系统, 同时也支持 Linux 系统。此外, SoC 系统将网络控制器、串口控制器和定时器设置为支持中断方式, 并链接到中断控制器上, 以通知 MicroBlaze 处理器中断事件的发生。

3.3 U-Boot 交叉编译环境的建立

采用交叉开发环境 (Cross Development Environment) 是嵌入式应用软件开发时的一个显著特点。首先在通用计算机上编写嵌入式应用软件程序, 然后通过交叉编译生成目标平台上可以运行的二进制代码格式, 最后再下载到目标平台上运行^[8-10]。对于 U-Boot 源代码, 其交叉编译环境一般为 Linux, 文中采用 Fedora 15 Linux 操作系统作为 MicroBlaze 的编译环境, 用已经编译好的 MicroBlaze GNU 交叉编译工具链作为开发工具进行开发。MicroBlaze GNU 交叉编译工具获取方法如下:

(1) 在 Fedora 15 环境下安装 git 工具。

(2) 执行 `git clone git://git.xilinx.com/xldk/microblaze_v1.0.git`。

从 `git.xilinx.com` 网站下载 `microblaze-unknown-linux-gnu.tar.gz`。

(3) 将 `microblaze-unknown-linux-gnu.tar.gz` 解压到 `/root` 目录下:

```
tarxvf microblaze-unknown-linux-gnu.tar.gz
```

(4) 将工具链的地址加入系统路径,并设置 `CROSS_COMPILE` 环境变量

```
export PATH=/root/microblaze-unknown-linux-gnu/bin:$PATH
```

```
export CROSS_COMPILE=microblaze-unknown-linux-gnu-
```

3.4 U-Boot 源代码下载

U-Boot 源代码同样可以通过 git 工具从 `git.xilinx.com` 网站下载,方法如下:

(1) 在 `/root` 目录下执行 `git clone git://git.xilinx.com/u-boot-xlnx.git`,该命令将 U-Boot 源代码复制到 `/root/u-boot-xlnx` 目录下。

(2) 执行 `cd U-Boot-xlnx` 切换到 `UBboot-xlnx` 目录下,然后执行 `git checkout -b microblaze origin/microblaze`。

将 U-Boot 代码的版本切换到 `microblaze` 分支下,就可以针对特定目标板进行相关文件的修改和编译。

3.5 xparameters.h 文件的生成

`xparameters.h` 是 `microblaze` U-Boot 代码中最重要的文件,该文件主要定义了目标板上所有部件的地址、参数等信息,如 `MicroBlaze` 处理器的频率、版本、是否使用 MMU、是否使用 cache、DDR RAM 存储器基地址、串口控制器基地址、以太网控制器基地址等。U-Boot 代码通过解析该文件即可确定相关部件的访问地址和特性。`xparameters.h` 头文件的编写可以参考 `board/xilinx/microblaze-generic` 目录下的 `xparameters.h` 文件,也可以通过 EDK 工具自动生成,方法如下:

(1) 在 EDK 开发界面中选择 `Project -> Export Hardware Design to SDK` 菜单,然后选择 `Export & Launch SDK`,将综合好的 FPGA 配置文件(.bit 文件)导出到指定目录。

(2) 启动 SDK 工具,选择 `File -> New -> Xilinx Board Support Package` 生成一个针对上一步生成的 SoC 系统的板级支持包。

(3) 在板级支持包目录下的 `include` 文件夹中即可看到 `xparameters.h` 文件。

3.6 移植操作内容

U-Boot 移植所需的开发环境建立好后,可以开始

移植操作。U-Boot 的移植,实际上就是根据嵌入式系统硬件资源提供的数据,对有关的文件进行修改。U-Boot 移植工作中修改的文件主要包括汇编文件和 C 文件两部分^[11,12]。汇编文件的修改主要是针对 `arch/microblaze/cpu/` 目录下文件的修改,C 语言部分主要是针对 `board` 目录下相关文件的修改。

由于 U-Boot 代码里面已经包含 `MicroBlaze` 的移植,所以对 SP605 评估板的移植主要是针对 `board` 的移植,步骤如下:

(1) `arch/microblaze/cpu` 文件夹里已经有 `cache.c`、`cpu.c`、`exception.c`、`interrupts.c`、`irq.S`、`start.S`、`timer.c` 几个文件,因而不需要新建 CPU 相关的目录。

(2) 在 `board/xilinx/` 目录下创建 `sp605` 目录,将同一级的 `microblaze-generic` 目录下的文件复制到新建的 `sp605` 目录下。

(3) `include/configs` 目录下添加 `sp605.h`,在这里可放入全局的宏定义等。`sp605.h` 文件可以参考 `microblaze-generic.h` 文件来进行相关的修改。

(4) 将 `xparameters.h` 文件复制到 `board/xilinx/sp605` 目录下。如上所述,`xparameters.h` 文件定义了 SP605 评估板上各部件的硬件地址,U-Boot 代码通过对该文件的解析完成代码的编译。

(5) 修改 U-Boot 根目录下的 `boards.cfg`,加入对 `sp605` 评估板的声明。`boards.cfg` 文件包含了 U-Boot 所支持的所有目标板的声明,任何新加入的目标板必须在该文件中加入相关的声明,其声明格式如下:

<编译目标名称> <CPU 架构> <CPU 型号> <目标板名称> <生产商> <SoC 平台名称> <参数>

文中按照目标板声明格式,在 `boards.cfg` 中加入了 `sp605` 评估板的声明,如下:

```
sp605microblaze microblaze sp605 xilinx-
```

其中<参数>一项省略,U-Boot 编译时会自动根据评估板前几个选项,自动搜索相应目录下的配置文件,完成编译。

(6) 修改 `board/xilinx/sp605/config.mk` 文件中的 `CONFIG_SYS_TEXT_BASE` 宏定义,将该宏定义的值修改为 SoC 系统 DDR RAM 基地址,例如文中 EDK 所生成的 DDR RAM 基地址 `0x50000000`。

(7) 运行 `make clobber`,清除以前生成的依赖关系文件。

(8) 运行 `make sp605_config`,对 U-Boot 进行配置。

(9) 运行 `make`,即可完成 U-Boot 代码的编译。

4 U-Boot 烧写及运行检查

首先将综合好的 FPGA 配置文件下载到 Spartan

⑥-6 FPGA 中,建立系统的硬件平台,然后在 xmd 命令窗口里键入如下命令,将编译好的 U-Boot 文件下载到 SP605 评估板的 DDR RAM 中:

(1) connect mb mdm

(2) down U-Boot

如果下载没有错误,则在 xmd 命令窗口里键入 run 命令,运行 U-Boot 代码。如果能从串口输出正确的启动信息,就表明 U-Boot 移植基本成功,文中所移植的 U-Boot 程序在 SP605 评估板上的运行情况如图 3 所示。

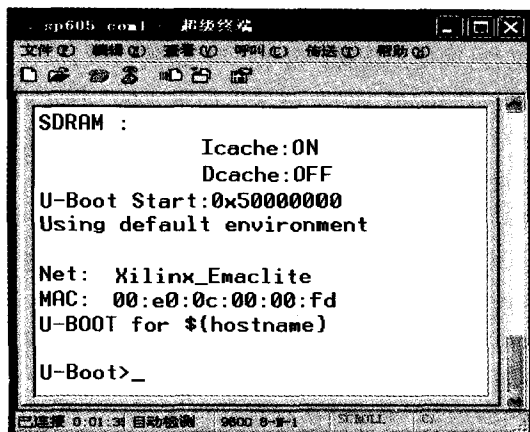


图 3 运行在 SP605 评估板上的 U-Boot 程序

U-Boot 运行后,可以通过它提供的命令测试目标板的一些基本情况,如检查内存是否读写正常、网络通信是否正常。此外用户也可以添加自定义的功能,将 U-Boot 作为主控程序,应用到某些对软件系统需求不高的地方,完成简单的嵌入式系统的控制。

5 结束语

U-Boot 是一个功能强大的 Bootloader 开发软件,

适用的 CPU 平台及支持的嵌入式操作系统很多。文中是笔者在实际开发过程中根据相关资料进行摸索,并在成功移植了 U-Boot 的基础上总结出来的。对于不同的 CPU 和开发板,其基本的方法和步骤是相同的,希望能对相关嵌入式系统的设计人员有所帮助。

参考文献:

- [1] 华清远见嵌入式培训中心. 嵌入式 Linux 应用程序开发标准教程[M]. 第 2 版. 北京:人民邮电出版社,2009.
- [2] Šimek M. Embedded Operating System for MicroBlaze[J]. Acta Polytechnica,2006,46:37-39.
- [3] Bmyninckx H, LeuvenK U. Real-time and Embedded Guide[J]. Mechanical Engineering,2001,56:5-9.
- [4] 张进,姜威. U-boot 的启动流程及移植[J]. 国外电子元件,2005(5):11-14.
- [5] 徐亚鹏,谢凯年. 用 U-Boot 构建 IXP2350 目标系统的引导程序[J]. 计算机技术与发展,2007,17(5):9-14.
- [6] Xilinx Company. SP605 Evaluation Platform[M]. [s. l.]: Xilinx Company,2007.
- [7] Xilinx Company. MicroBlaze Processor Reference Guide[M]. [s. l.]: Xilinx Company,2009.
- [8] 叶肇晋. 基于 XILINX FPGA 片上嵌入式系统的用户 IP 开发[M]. 西安:西安电子科技大学出版社,2008.
- [9] 赵峰. FPGA 上的嵌入式系统设计实例[M]. 西安:西安电子科技大学出版社,2008.
- [10] 杨强浩. 基于 EDK 的 FPGA 嵌入式系统开发[M]. 北京:机械工业出版社,2008.
- [11] 周涛,刘有源. 基于 MIPS 的 Au1500 NC 板上的启动程序 U-Boot 设计[J]. 计算机技术与发展,2006,16(8):94-96.
- [12] 吴桂清,张峰铭,秦丽,等. 基于 MicroBlaze 的振动信号采集系统的设计与实现[J]. 电子设计工程,2010(3):37-39.

(上接第 70 页)

- 对象定位模型[J]. 软件学报,2005,16(5):894-902.
- [5] 刘业,杨鹏. 基于自组织聚类的结构化 P2P 语义路由改进算法[J]. 软件学报,2006,17(2):339-348.
- [6] 杨静. 基于用户兴趣的 P2P 搜索机制研究[D]. 西安:西安电子科技大学,2010.
- [7] 朱晓姝,周娅,黄桂敏. 基于小世界层次分布式路由模型研究[J]. 计算机工程,2006,32(15):120-122.
- [8] 宋雪梅. 蚁群算法的改进及应用研究[D]. 唐山:河北理工大学,2006.
- [9] 王新生,李学,贾冬艳. 基于蚁群算法的非结构化 P2P 资源搜索机制[J]. 计算机工程,2009,35(7):189-194.
- [10] Bontoux B, Feillet D. Ant colony optimization for the traveling purchaser problem[J]. Computer & Operations Research, 2008,35(2):628-637.
- [11] 蓝慧琴,钟诚,李智. 一种基于蚁群算法的非结构化 P2P 网络搜索算法[J]. 计算机技术与发展,2006,16(10):26-28.
- [12] 青桂仙,苏筱蔚,陈淑艳. 中文文本挖掘的无词典分词的算法及其应用[J]. 吉林工程学院学报,2002,23(1):16-18.
- [13] 何盈捷,王珊,杜小勇. 纯 Peer to Peer 环境下有效的 Top-k 查询[J]. 软件学报,2005,16(4):540-552.
- [14] PeerSim [EB/OL]. 2010. <http://peersim.sourceforge.net/>.