文章编号:1005－0523(2005)05－0092－04

U－Boot 在 S3C2410 上的移植分析

宋 凯,甘 岚,严丽平

(华东交通大学 信息工程学院,江西 南昌 330013)

摘要:介绍了一款优秀的嵌入式 BootLoader－U－Boot,详细讲解了它的运行原理,着重讨论了其在 S3C2410 上的移植过程,并对移植结果进行了测试.

关 键 词:U－Boot;S3C2410;移植;BootLoader
中图分类号:TP368.1 **文献标识码**:A

0 引言

BootLoader 是嵌入式系统软件开发的第一个环节,它紧密地将软硬件衔接在一起,对于一个嵌入式设备后续的软件开发至关重要.BootLoader 还涉及到许多硬件相关的知识,对于普通的嵌入式开发板,它又是不可跳过的步骤,所以做好它的移植工作是必须的,对于后续的开发工作也是有益的.U－Boot 是当前比较流行、功能强大的 BootLoader,它操作简便,可以支持多种体系结构的处理器,同时提供了完备的命令体系.S3C2410 是三星公司一款基于 ARM920T 的嵌入式通用处理器.本文的移植平台就是以 S3C2410 为核心的 HHARM9－EDU－R2 开发板,这块开发板的硬件资源配置较为完善.主要硬件资源有:S3C2410 处理器;1 片 Intel TE28f128 FLASH (16M);2 片 Hynix HY57V561620 SDRAM (64M);10/100M 自适应网络芯片 DM9000;USB Host/Device; RS232 × 2/RS485 × 1 串口;LQ035FLM08L 256K 色 TFT 真彩 LCD 显示屏;全功能 JTAG 调试口等.

1 U－Boot 简介

U－Boot 是 Das U－Boot 的简称,是由 denx 软件中心依照 GPL 发布的公共软件,作为系统启动的引导模块,U－Boot 支持多种处理器架构,比如 PowerPC、ARM、MIPS 和 x86 等.目前,U－Boot 源代码在 sourceforge 网站的社区服务器中,Internet 上有一群自由开发人员对其进行维护和开发,它的项目主页是 http://sourceforge.net/projects/u－boot.当下载并解压 U－Boot 源码包后,会形成如下的目录结构:board,和一些已有开发板有关的文件;common,实现各种 U－Boot 命令的 C 文件;cpu,CPU 相关文件,其中的子目录都是以 U－Boot 所支持的 CPU 命名的;disk,disk 驱动的分区处理代码;doc,文档;drivers,通用设备驱动程序;fs,支持文件系统的文件;include,头文件,对各种硬件平台支持的汇编文件,系统的配置文件和对文件系统支持的文件;net:与网络有关的代码;lib-arm,与 ARM 体系结构相关的代码;tools,创建 S－Record 格式文件和 U－Boot Images 的工具.

本文中 U－Boot 的移植就是根据 HHARM9－EDU－R2 开发板的硬件资源在以上的目录中修改

收稿日期:2005－02－08
作者简介:宋凯(1980－),男,河南泌阳人,华东交通大学在读研究生,研究方向:嵌入式系统.

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

或者添加相关源文件,并且重新编译的过程.移植工作开始之前,了解 U－Boot 的运行过程是十分必要的.

2　U－Boot 运行过程分析

U－Boot 编译后的代码定义一般不超过 100 kB,并且这 100 kB 又分成两个阶段来执行.第一阶段的代码在 start.s 中定义,大小不超过 10 kB,它包括从系统上电后在 0x00000000 地址开始执行的部分.这部分代码运行在 Flash 中,它包括对 S3C2410 的一些寄存器的初始化和将 U－Boot 的第二阶段代码从 Flash 拷贝到 SDRAM 中.除去第一阶段的代码,剩下的部分都是第二阶段的代码.第二阶段的起始地址是在第一阶段代码中指定的,被复制到 SDRAM 后,就从第一阶段跳到这个入口地址开始执行剩余部分代码.第二阶段主要是进行一些 BSS 段设置,堆栈的初始化等工作,最后会跳转到 main_loop 函数中,接受命令并进行命令处理.图 1 给出了 U－Boot 的详细的运行过程包括对内核的设置、装载及调用过程.

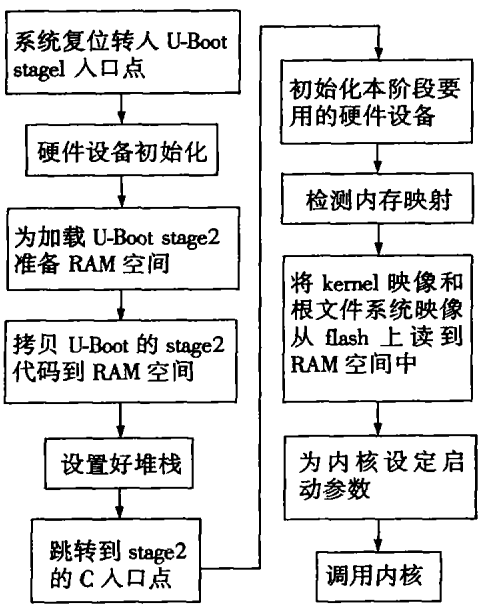

图 1　U-Boot 运行过程

了解了 U－Boot 的运行过程以后,我们还必须确定开发板的地址空间分布,才可以进行源码的修改和移植工作.地址空间的分布部分依赖于开发板的硬件配置及 CPU 的复位地址.本文中开发板的地址空间如图 2 所示.

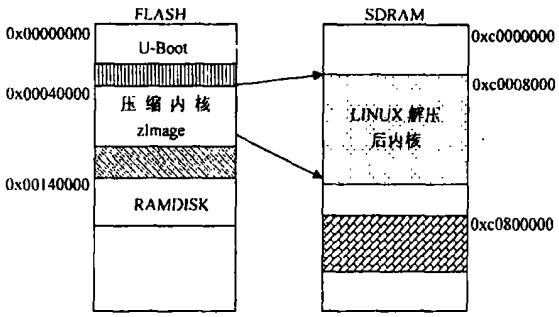

图 2　开发板地址空间分布

3　U－Boot 的移植与测试

为了使移植工作更加快捷,应当选择 U－Boot 当前发布的最新版本 1.1.2(尽管通过 CVS 可以得到 U－Boot1.1.3,但其正在开发,尚未发布,不宜使用).因为最新的版本可以提供尽可能多的处理器核及开发板的支持.对于 U－Boot－1.1.2 而言,它不仅提供对 ARM－920T 内核的支持,而且直接提供了对于 S3C2410 的板级支持,这使移植工作量相对减少.

3.1　支持 ARM－920T 内核的代码修改

由于 U－Boot－1.1.2 提供对 ARM－920T 内核的直接支持,所以本步骤不需要做任何工作,本文为了让读者了解 BootLoder 移植的通用模式,在此只是稍加提示.

3.2　配置自己的开发板

建立自己开发板的目录和相关文件.

1) 在 include/configs 目录中以 smdk2410.h 为模板添加头文件 S3C2410.h (cp smdk2410.h S3C2410.h).这个文件是开发板的配置文件,它包括开发板的 CPU、系统时钟、RAM、Flash 系统及其它相关的配置信息.

2) 在 board/ 目录下创建 S3C2410 目录.拷贝 smdk2410 目录下所有文件到 S3C2410 目录下,共有如下六个文件:flash.c、memsetup.c、S3C2410.c、Makefile、U－Boot.lds 和 config.mk,根据开发板实际情况对各个文件进行修改.

◆ flash.c.U－Boot 读、写和删除 Flash 设备的源代码文件.由于不同开发板中 Flash 存储器的种类各不相同,所以,修改 flash.c 时需参考相应的 Flash 芯片手册.它包括如下几个函数:

unsigned long flash_init (void),Flash 初始化;

void flash_print_info (flash_info_t * info),打印 Flash 信息;

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

int flash_erase(flash_info_t * info, ints_first, ints_last),Flash 擦除;

volatile static int write-hword(flash_info_t * info, ulong dest, ulong data),Flash 写入;

int write_buff (flash_info_t * info, uchar * src, ulong addr, ulong cnt),从内存复制数据.

由于本文开发板所用 flash 芯片为 Intel TE28f128,在 board/cmi 目录中有此 flash.c,只需对其稍加修改即可使用.

◆memsetup.c .初始化时钟、SMC 控制器和 SDRAM 控制器.为了以后能用 U-Boot 的 GO 命令执行修改过的用 loadb 或 tftp 下载的 U-Boot.在标记符"0:"上加入五句:

```
mov r3, pc
ldr r4, =0x3FFF0000
and r3, r3, r4 //以上三句得到实际起动的内存地址
aad r0, r0, r3 //用 GO 命令调试 uboot 时,启动地址在 RAM
add r2, r2, r3 //把初始化内存信息的地址,加上实际起动地址
```

◆S3C2410.c.设置各种总线时钟,打开数据 Cache 和指令 Cache ,并设置相关内存参数.

◆Makefile.修改:OBJS := S3C2410.o flash.o memsetup.o

◆U-Boot.lds.作如下修改:

```
.text
{
cpu/arm920t/start.o (.text)
*(.text)
}
```

◆config.mk.用于设置程序连接的起始地址,因为会在 U-Boot 中增加功能,所以留下 6M 的空间,修改 33F80000 为 33A00000.

3.3 实现网卡的驱动程序

在 drivers/ 目录中以 dm9000x.c 和 dm9000x.h 为模板添加网口设备控制程序 dm9000.c 和 dm9000.h,其中 dm9000.c 主要包括以下函数:

int eth_init (bd_t *bd),初始化网络设备;

int eth_send(volatile void *, int),发送数据包;

int eth_rx(void),接收数据包.

void eth_halt(void),关闭网络设备;

为了方便网络设备的数据读写操作,还定义了如下函数:

static int dm9000_probe(void),搜索 DM9000 芯片,分配空间并登记之;

static u16 phy_read(int),从 Phyxcer 寄存器读取一个字;

static void phy_write(int, u16),写一个字到 Phyxcer 寄存器;

static u16 read_srom_word(int),从 SROM 读取一个字数据;

static u8 DM9000_ior(int),从 I/O 口读取一个字节;

static void DM9000_iow(int reg, u8 value),写一个字节到 I/O 口;

最后在 drivers/Makefile 中加入 dm9000.o.

3.4 修改 Makefile 文件

在 U-Boot-1.1.2/Makefile 中 ARM92xT Systems 注释下面加入以下两行:

```
S3C2410_config : unconfig
        @./mkconfig $(@:_config=) arm arm920t S3C2410
```

其中"arm"是 CPU 的种类,arm920t 是 ARM CPU 对应的代码目录,S3C2410 是自己开发板对应的目录.

交叉编译器安装在:/path/armv4l-unknown-linux-目录下,所以把 CROSS_COMPILE 设置成相应的路径:CROSS_COMPILE = /path/arm4l-unknown-linux-

3.5 生成目标文件并进行测试

依次运行以下命令:

```
#make clean
# make S3C2410-config
# make
```

之后会生成三个文件:

U-Boot——ELF 格式的文件,可以被大多数 Debug 程序识别;

U-Boot.bin——二进制文件,纯粹的 U-Boot 二进制执行代码,不保存 ELF 格式和调试信息.这个文件一般用于烧录到用户开发板中;

U-Boot.srec——Motorola S-Record 格式,可以通过串行口下载到开发板中.

3.6 测试与应用

1) 测试

利用编制好的 Flash 烧写程序,通过 JTAG 口将生成的二进制文件 U-Boot.bin 烧入 Flash 的零地址.烧录成功后,拔掉 JTAG 调试线并复位开发板,从 Minicom 终端输出如下信息:

(C)1994-2025 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
U—Boot 1.1.2 (Jul 20 2005 — 09:34:21)
U—Boot code: 33F00000 —> 33F1952C BSS: —> 33F1D870
RAM Configuration:
Bank #0: 30000000 64 MB
Flash Memory Start 0x0000000
Device ID of the Flash is 18
Flash: 16 MB
Write 18 to Watchdog and it is 18 now
In: serial
Out: serial
Err: serial
SMDK2410 #
```

串口输出的以上信息表明,CPU 和串口已正常工作.通过 U—Boot 提供的命令 flinfo 和 mtest 可以测试 Flash 和 RAM.经过测试,可以正确地读出 Flash 信息及读写 RAM,表明 Flash 和 DRAM 已正确初始化.用 tftp 命令传输宿主机 tftpboot 目录下任一小文件到 RAM 成功,说明网卡芯片也成功驱动.

2) 简单应用

U—Boot 的主要作用是用来引导内核.因此,通过 U—Boot 引导一个特定的内核,可以进一步测试其移植的稳定性.而使用 U—Boot 引导内核有两种不同的方法.第一种方法是直接将内核映象文件和根文件系统烧写入 Flash,使用此方法,U—Boot 在启动时将 Flash 中的内核映象及根文件系统读入 RAM 指定位置并从同一位置启动内核.第二种方法是将内核映象文件和根文件系统下载至 RAM 中直接启动(而不是从 Flash 中读入 RAM),此种方法不需要烧写 Flash.笔者为了减少烧写 Flash 的次数,在本文中采用第二种方法,其步骤如下:

```
SMDK2410 # tftp 30008000 zImage
SMDK2410 # tftp 30800000 ramdisk.image.gz
SMDK2410 # go 30008000
```

上述指令执行的过程中,未出现异常,内核成功启动,并最终进入 Shell 提示符"#".在 Shell 提示符下输入内核编译时定制的各个命令,均可以正常运行.另外编写简单 C 程序,并用交叉编译器编译之,最终生成的可执行文件能够在开发板上正常运行.上述事实说明内核经过 U—Boot 引导已稳定运行在开发板上.此次应用,进一步验证了 U—Boot 移植的稳定性.至此,移植工作告一段落.

4　结　语

目前,笔者移植的 U—Boot 已经能稳定地运行在开发板上,这使得 Linux 内核的调试脱离了 BDM 调试器,节约了大量的开发时间,大大提高了效率,是对后续嵌入式开发的有力支持.当然,U—Boot 只是一款好用的 BootLoader,嵌入式 Linux 的开发存在很多技术细节,只有根据实际情况不断修改、调试、总结,才能获得更大的成功.

参考文献:

[1] 曾宏安,等.用 U—Boot 构建嵌入式系统的引导装载程序[J].单片机与嵌入式系统应用,2005,2.

[2] 赵俊良,李哲英.Blob 在 S3C44B0 上的移植[J]. 单片机与嵌入式系统应用,2004,9.

[3] 王俊卿,等.NIOS 软核处理器的 Linux 引导程序 U—Boot 设计[J]. 单片机与嵌入式系统应用,2004,9.

[4] 吕波等.基于 PPCBOOT 的嵌入式 Linux 系统的开发[J].单片机与嵌入式系统应用,2003,8.

[5] Karim Yaghmour.Building Embedded Linux System. O'Reilly.2004.

Analysis of Porting U—Boot to S3C2410

SONG Kai, GAN Lan, YAN Li-ping

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: This paper introduces an excellent embedded BootLoader—U—Boot and detailedly describes its operating theory. Then it emphasizes on how to port U-Boot to S3C2410 and tests for the porting results.

Key words: U-Boot; S3C2410; port; BootLoader

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net