

FPGA 嵌入式系统的引导实现^①

Boot Realization of FPGA Embedded System

王淑贞 姚 铭 (厦门大学 信息科学与技术学院 福建 厦门 361005)

摘 要: 基于 xilinx 公司 ml403 板子, 实现 FPGA 嵌入式系统的引导, 采用一种新型的加载引导方式—System ACE 方式。通过修改 u-boot 源码, 制作.ace 文件, 分区和格式化 CF(Compact Flash)卡, 并将位流文件.bit 文件、u-boot(Universal Boot Loader)映像文件、内核映像文件、文件系统都固化于 CF 卡上。最终实现以 System ACE 方式加载 u-boot 到 RAM 中运行, 再由 u-boot 通过此方式加载内核和文件系统, 成功地实现了 FPGA 嵌入式系统的引导。

关键字: System ACE u-boot CF 卡 引导实现

FPGA(Field Programmable Gate Array)现场可编程门阵列是可编程半导体器件, 具有逻辑单元灵活、集成度高以及适用范围宽等特点, 与其它门阵列相比, 它又具有开发周期短、成本低、开发工具先进、质量稳定以及可实时在线检验等优点。因此 FPGA 产品得到广泛的应用, 其应用领域已经从原来的通信扩展到消费电子、汽车电子、工业控制、测试测量等广泛的领域。FPGA 嵌入式系统也是 FPGA 产品得到广泛应用很重要的原因之一, 而嵌入式系统又必然涉及到引导程序(bootloader), 它的作用是初始化硬件设备, 为最终调用操作系统内核做好准备。根据加载引导方式的不同, 选择 System ACE 方式, 用此方法进行系统启动引导设计, 既利用了 FPGA 的速度快、I/O 口丰富的特点, 又利用了 CF 卡的容量大、非易失性和即插即用的特点。而且 CF 卡读写速度快、功耗低、抗震能力强, 同时与 Windows 98/Windows XP 操作系统有良好的兼容性, 目前已被广泛应用于大中规模的嵌入式系统设计中。

1 新型bootloader启动引导方式—System ACE 方式

在嵌入式系统研发阶段, 现有的 BootLoader 下载引导方式可根据加载途径的不同细分为以下几种:

①通过 Ethernet 网口从宿主机下载内核到目标板;

②通过串口从宿主机下载内核到目标板;

③直接从 Flash 中提取已存储的内核

前两种在系统调试时需要进行宿主机与目标板间的硬件线路的实际连接, 使用起来不够方便; 而第三种需要大容量的 Flash 的支持, 增加了研发成本, 而且烧片的速度比较缓慢。这三种引导方式的调试的效率不高, 进行内核更新时也不够灵活。而采用可移动的存储介质 CF 卡对系统内核进行存储, 在调试时无需建立宿主机和目标板的硬件连接, 提高了调试的效率, 使用起来更便捷、更灵活; 进行内核更新时也更为灵活, 只需把更新内核转存到指定目录^[1]。

对于引导程序 bootloader, 根据代码工具的不同, 可以有以下两种方法:

①简单的 bootloader 程序: 在 EDK 工程文件中, 加入一简单的应用程序 bootloader(存放于 bram), 并选中其 Mark to Initialize BRAMS, 这样一旦.bit 被加载, bootloader 就自动运行。

②用 u-boot 来引导启动内核并加载文件系统 U-boot 全称 Universal Boot Loader, 是开放源代码、支持多种嵌入式操作系统内核、支持多种处理器系列、具有较高的稳定性、高度灵活的功能设置、

^① 收稿时间:2008-12-22

丰富的设备驱动源码，另外 u-boot 对操作系统和产品研发提供了灵活丰富的支持，主要表现在：可以引导压缩或非压缩系统内核，可以灵活设置/传递多个关键参数给操作系统，适合系统在不同开发阶段的调试要求，支持多种文件系统，支持多种目标板环境参数存储介质，提供多种控制台接口，使用户可以在不需要 ICE 的情况下通过串口/以太网/USB 等接口下载数据并烧录到存储设备中去(这个功能在软件现场升级的时候是很实用的)^[2]。鉴于 u-boot 这一系列优点，本文选择用它来引导 FPGA 上的系统，就如何通过 System ACE 方式加载引导内核进行了详解。

2 System ACE CF

System ACE(Advanced Configuration Environment)由两部分组成：ACE 控制器芯片、ACE CF 卡(Compact Flash)。ACE 控制器具有以下特性：

- ①支持绝大多数第三方标准 CF 卡；
- ②通过 JTAG 方式配置 FPGA；
- ③具有 MPU(Microprocessor Interface Unit)接口，可以直接和 IBM PowerPC405 等 MPU 连接；
- ④支持 FAT12/16 文件系统；
- ⑤功耗较低，适应嵌入式环境。

ACE 控制器不仅可以从 CF 卡读取配置文件，通过 JTAG 方式配置 FPGA，还提供了 MPU 接口，方便通过 MPU 管理和配置 FPGA。System ACE 对使用的 CF 卡并无特殊要求，符合 ATA 接口标准的 CF 卡大多可以使用，适用于 Xilinx SPARTAN 系列、VIRTEX 系列的任何 FPGA。它与 CPU、CF 卡之间的连线如图 1

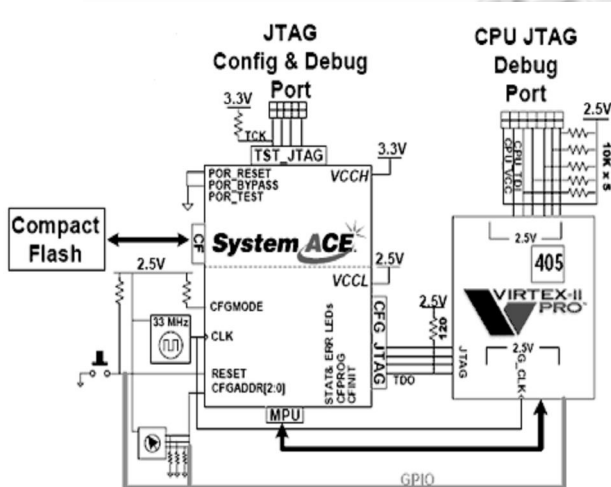


图 1 System ACE 与 CPU 的接线图

EDK(Embedded Development Kit)提供的 IP 核中，有专门的 opb_sysace 核，通过 OPB 总线，处理器不但可以直接使用 CF 卡作为配置器件，还可以访问、操作 CF 卡。在大中型嵌入式 FPGA 设计中，System ACE 以其低廉的价格、简单的接口成为首选配置方案^[3]。当 CF 卡用于配置 FPGA 时，其目录结构如图 2 所示：

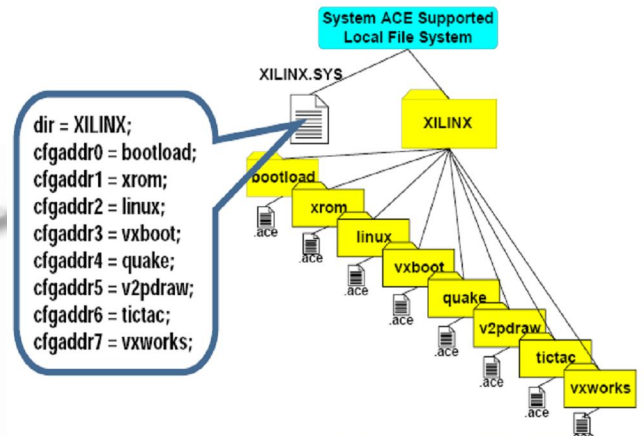


图 2 System ACE 的目录结构

整个 XILINX 目录称之为‘集’，一个集最多可以包含八种配置，八个子目录对应八种配置，每个子目录下的 .ace 文件即为对应的配置文件，可以给每个 .ace 文件分配不同的地址。System ACE CF 卡可以有多个集，但是在任一时刻却只有一个集被激活，通过查看 XILINX.SYS 文件就知道目前哪个“集”处于激活状态，同时它用来控制当前集下有哪些配置文件，及每个配置文件的路径信息。为此 xilinx.sys 文件至关重要，是必不可少的，且文件名只能为 xilinx.sys，否则 FPGA 读取 CF 卡时会出错，文件里的内容也不可随意涂改。这些目录结构的产生是通过 iMPACT 工具自动产生的，包括 .sys 文件和 .ace 文件。为了让 System ACE 一复位能自动运行引导程序，需把引导程序写到 CF 卡的 0 地址上，xilinx.sys 中的 cfgaddr0 即代表 0 地址所存放的配置文件，复位后首先读取此文件并将 0 地址所对应的 .ace 文件，然后通过 MPU 接口拷贝到 RAM 让其运行。事实上格式化后的 CF 卡，其 FAT 文件系统是由如下 4 部分组成的：

- 0 区：保留区(引导区)。
- 1 区：FAT 区。
- 2 区：根目录区(FAT32 中存在的是浮动的 FDT

表)。

3 区: 文件、目录与数据区。

而引导程序是需要把它写到 0 区即引导区, 这样引导程序才能自动而正常运行, 进而启动内核的。

3 基于PowerPC405的嵌入式环境

(1) 硬件

因为采用的是 xilinx 公司的 ML403 板, 其 FPGA 型号是 V4FX12, 用 EDK 构造工程时需选对板子类型, 需要的基本组成部分及相关参数如下所示:

① ppc405_virtex4 核

• 100MHZ 输入时钟, 200MHZ CPU 频率, 133MHZ 外设总线频率;

- 使用 CACHE;
- 使用浮点计算 APU(辅助处理器单元)。

② plb_ddr 核(使用中断)

③ opb_uart16550(使用中断)

• 波特率 9600 bit/s;

• 数据位 8 位, 1 位停止位, 无奇偶校验位, 无流控

④ opb_ethernet(10/100M Ethernet MAC)

- 非 DMA 方式, 使用中断

⑤ opb_sysace(16 位, 使用中断)

(2) 软件环境

成功用 u-boot 引导 CF 卡的内核及加载文件系统, 所涉及到的软件环境:

- EDK 9.1, iMPACT, VMware Workstation
- u-boot-1.3.2
- linux-2.6-xlnx.git
- busybox-1.1.2

4 嵌入式系统的引导实现

4.1 u-boot 配置及编译

u-boot 具有通用性, 包含有很多种体系结构, 很多种 CPU 及各种必需的设备驱动程序, 因此我们必须进行板级配置, 使之与所要移植的板子相对应。而对于 ML403 板子, 其配置主要涉及以下几个文件:

首先是 board/ml403 目录下的重要文件

xparameters.h: 此文件是由 EDK 软件 BSP 产生的, 包含关于 ML403 对应的硬件信息, 是板级配置的基础文件。

ml403.c: 这个文件主要是初始化板子的板级信息, 包括内存大小、CPU 频率、总线频率等, 这些需与板子的实际信息相对应, 否则 u-boot 不能正常运行。

serial.c: 这个是标准串口 opb_uart16550 的驱动程序, 若是有误, 则串口将不能正常运行, 终端也将无输出。

config.mk: 定义 u-boot 在内存中的加载地址 另外, u-boot 配置所涉及到的一个重要文件 include/configs/ml403.h, 此头文件包含有很多重要的配置信息, 可以根据不同需要进行修改。例如有以下几个重要定义:

```
关于 System ACE
#define CFG_SYSTEMACE_BASEXPAR_SYSA
CE_COMPACTFLASH_BASEADDR
#define CFG_SYSTEMACE_WIDTH
XPAR_SYSACE_COMPACTFLASH_MEM_WIDTH
#if defined(CONFIG_SYSTEMACE)
#define CONFIG_CMD_FAT
#endif
```

其中, CONFIG_CMD_FAT 必须定义, 唯有定义它, u-boot 命令中才有关于 CF 卡的命令如 fatls, fatinfo, fatload, 而采用 System ACE 方式引导内核, 要求 fatload 命令能正常使用, 否则将无法从 CF 卡启动内核。

```
关于 uart16550
#define CFG_DUART_CHAN 0
#define CFG_NS16550_REG_SIZE -4
#define CFG_NS16550 1
#define CFG_INIT_CHAN1 1
```

必须要有如上定义, 串口 uart16550 的驱动程序 board/ml403/serial.c 才能正常编译, 也才能正常运行。

关于命令参数的设定

```
#define CONFIG_BOOTCOMMAND "fatload
ace 0 0x3000000 linux/ulmage;bootm 0x300
0000"
```

如果在设定的时间范围内, 用户没有按键, u-boot 就会进入启动加载模式, 自动执行 CONFIG_BOOTCOMMAND 所定义的参数 "fatload ace 0 0x3000000 linux/ulmage;bootm 0x3000000"

即将 CF 卡上 linux/ulmage 文件拷贝到内存地址 0x3000000(可以自己指定), 然后通过 u-boot 命令 bootm 启动。为了让 u-boot 在启动加载模式下能自主启动内核, 则必须设定此参数, 这样 System ACE 复位后 u-boot 会自动启动存于 CF 卡上的内核映像文件 ulmage。

```
#define CONFIG_BOOTARGS"console=ttyS0,
9600 ip=off root=/dev/xsa2 rw"
```

CONFIG_BOOTARGS 相当于内核命令行参数, 若定义不正确将无法启动内核并加载文件系统, root=/dev/xsa2 rw 定义的就是文件系统的存放位置即 system ACE CF 卡的第二个分区。

配置完之后, 通过如下命令即可对 u-boot 进行编译:

```
[root@localhost u-boot-1.3.2]# make clean
[root@localhost u-boot-1.3.2]# make
ml403_config
```

```
[root@localhost u-boot-1.3.2]# make
```

编译成功之后将产生 u-boot, u-boot.bin, u-boot.srec, u-boot.map, 在这里只用 u-boot, 它是一个 elf 格式的二进制映像文件。

4.2 通过 System ACE 固化 u-boot

(1) .ace 文件的制作

通常 u-boot 都是在 XMD 环境下调试的, 直接通过 xmd 命令将其下载到内存中让其运行, 看是否正常运行, 其命令如下:

```
XMD% dow u-boot
XMD% con
```

但是当调试完毕, 就需把它保存到 CF 卡中, 以便脱离 XMD 环境, 加电后可以独立运行。CF 卡上的文件都是以 .ace 的格式存储的, 而 .ace 文件的产生是通在 EDK shell 下键入如下命令:

```
xmd -tcl genace.tcl -opt genace.opt
```

其中, genace.opt 的内容如下所示:

```
-jprog
-target ppc_hw
-board ml403
-hw implementation/download.bit
-elf u-boot/u-boot
-ace u-boot_ml403.ace
```

这样命令行会提示成功创建 u-boot_ml403.

ace, 如:

```
Converting SVF file 'u-boot_ml403.svf' to SystemACE file 'u-boot_ml403.ace'
Executing 'inpect -batch svf2ace.scr'

SystemACE file 'u-boot_ml403.ace' created successfully
```

(2) System ACE CF 卡的分区

u-boot_ml403.ace, 还有编译好的内核 ulmage, 以及文件系统都需存储于 CF 卡(2GB)上, 但文件格式并不一样, u-boot_ml403.ace 和 ulmage 是以 fat16/fat12 的格式存放于 CF 卡上, 而 ramdisk 文件系统是以 ext2 格式存放于 CF 卡上, 因此就必须得对 CF 卡进行分区, 在这里, 我把 CF 卡分成两个区 FAT16(128M), EXT2(1.79G), 而分区主要是用 linux 下的 fdisk 命令来完成的, 其分区步骤如下:

① 进入 linux 系统, 打开终端, 打入如下命令:

- fdisk -l (查看 CF 卡的设备名), 如 /dev/sdc
- fdisk /dev/sdc (开始对 CF 卡进行分区)

其信息显示如下:

```
command (m for help): m
```

打入 m 即可显示所有与分区有关的命令, 如 p (显示当前所有的分区), d (删除分区), n (创建分区), t (改变文件系统格式, 默认为 linux 系统格式) 等等, 根据这些命令创建和修改分区。

② 分区后, 还需对分区格式化, 命令如下:

```
mkfs.vfat /dev/sdc1 (将第一个分区格式化为 fat16)
mkfs.ext2 /dev/sdc2 (将第二个分区格式化为 ext2)
```

(3) 固化 u-boot

分区后, 接下来就应该把 u-boot_ml403.ace 和内核映像文件 ulmage 拷贝到 CF 卡上, 但是除了这两个文件, 还需 xilinx.sys 和一定的目录结构, 而 xilinx.sys 是由 iMPACT 自动产生, 因此还需借助 iMPACT 工具, CF 卡整个目录结构如图 3 所示。

在所有的文件都被拷贝到 CF 卡上之后, 当然也包括 linux 分区上的文件系统, 然后修改一下板子上的跳线, 让其从 JTAG 启动, System ACE 复位后, 整个加载流程如下图 4 所示。

其启动信息如下所示:

```
U-Boot 1.3.2 (Sep 23 2008 - 10:51:13)
### No HW ID - assuming ML403
```

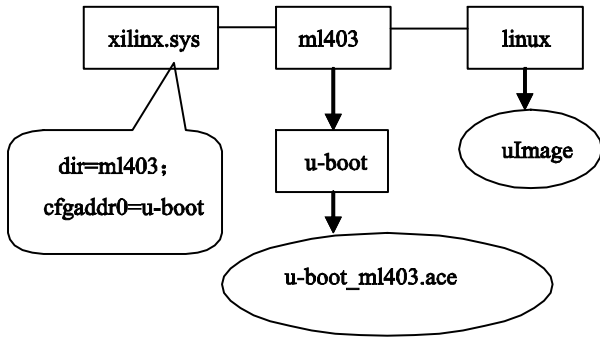


图 3 System ACE CF 卡的目录结构图

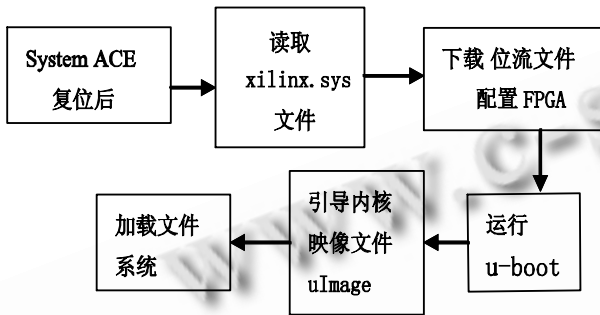


图 4 System ACE 复位后加载流程图

DRAM: 64 MB

*** Warning - bad CRC, using default environment

In: serial

Out: serial

Err: serial

Net: Hit any key to stop autoboot: 0

reading linux/uImage

```
## Booting image at 03000000 ...
Image Name:   Linux-2.6.26-rc8
Image Type:   PowerPC Linux Kernel Image
              (gzip compressed)
Data Size:    1289376 Bytes = 1.2 MB
Load Address: 00000000
Entry Point:  00000000
Uncompressing Kernel Image ... OK
.....
root@ml403:~#
```

5 结语

在 FPGA 平台上,通过 System ACE 方式成功实现嵌入式系统的引导加载工作。System ACE 一复位,系统将加载 u-boot_ml403.ace 程序,事先设定好启动参数,u-boot 会自动启动内核并加载文件系统。CF 卡具有价格便宜,容量大,方便存储等优点,所有文件都存放于 CF 卡上,若需修改,只需将 CF 卡取出,借助读卡器进行修改即可,既灵活又方便。更者由于 CF 卡容量大的特点,将文件系统存放于 CF 卡,可以较大范围地扩展文件系统。

参考文献

- 1 李波.一种新的 bootloader 启动方式引导研究.单片机与嵌入式系统应用,2008,7.
- 2 郑灵翔.嵌入式系统设计与应用开发.北京:北京航空航天大学出版社,2006:209-210.
- 3 杨强浩.基于 EDK 的 FPGA 嵌入式系统开发.北京:机械工业出版社,2008:71-72.