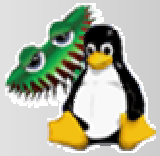
嵌入式系统引导程序移植

华清远见 刘洪涛



www.farsight.com.cn

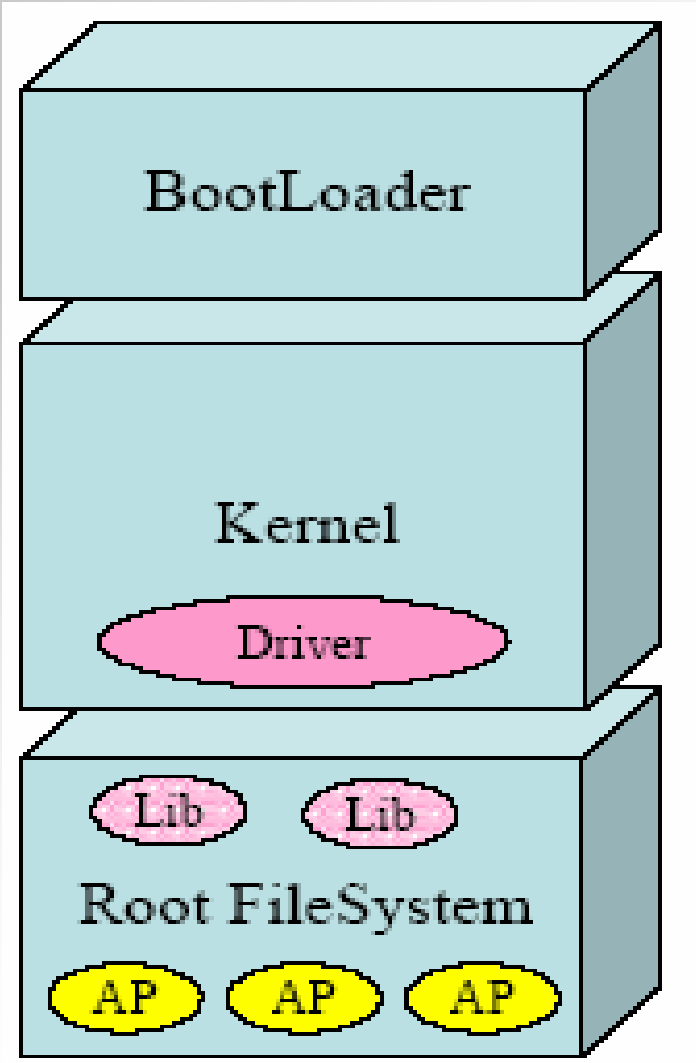
报告内容安排

- 引导程序介绍
- u-boot的功能及常用命令
- u-boot启动及初始化过程
- u-boot在nandflash上的移植要点
- 在u-boot中命令实现的原理

（以上内容均以arm体系结构为例）

FARSIGHT

嵌入式系统组成（软件）



开源的Bootloaders

Bootloader	Monitor	Description	x86	ARM	PowerPC
LILO	No	Main disk bootloader for Linux	Yes	No	No
GRUB	No	GNU's successor to LILO	Yes	No	No
Loadlin	No	Loads Linux from DOS	Yes	No	No
BLOB	No	Loader from the LART hardware project	No	Yes	No
U-boot	Yes	Universal loader	Yes	Yes	Yes
RedBoot	Yes	eCos-based loader	Yes	Yes	Yes

FARSIGHT

Bootloader任务

- 初始化处理器以及外设的硬件资源,配置SDRAM控制器,为主程序提供运行环境
  - 串口，提供交互终端
  - 网络，传输镜像文件
  - 其它I/O设备
- 执行系统自检，报告检测结果
- 引导操作系统
- 根据系统命令烧写镜像文件

U-BOOT常用命令1/2

- ？得到所有命令列表
- help: help nand 列出nand功能的使用说明
- ping： 只能开发板PING别的机器
- setenv: 设置环境变量：

  setenv serverip 192.168.0.1

  setenv ipaddr 192.168.0.56
- tftp: tftp 30008000 zImage
- nand: nand erase 40000 1c0000

  nand write 30008000 40000 1c0000

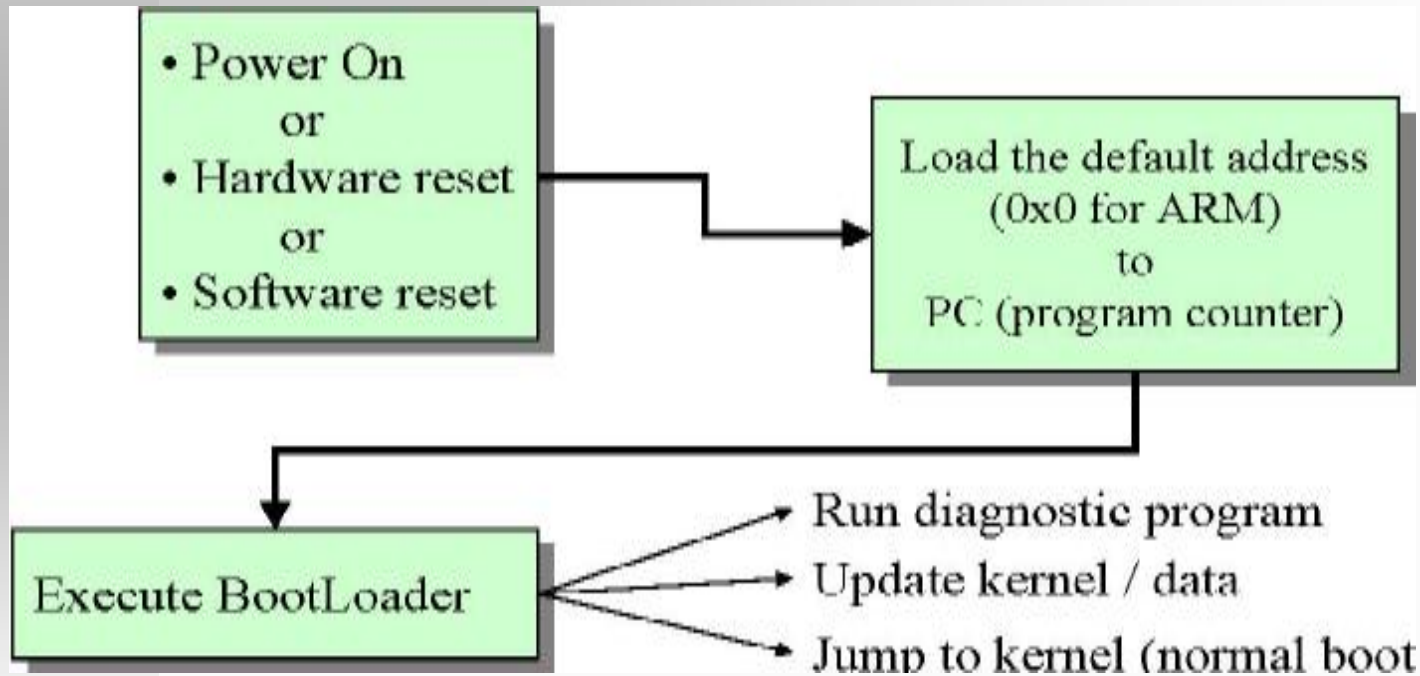
U-BOOT常用命令2/2

- setenv bootcmd nand read 30008000 40000 1c0000\; go 30008000
- setenv bootargs root=/dev/mtdblock2 console=ttySAC0,115200 init=/linuxrc
- saveenv

FARSIGHT

引导程序启动

- 系统上电复位，CPU在默认地址读取第一条指令。



FARSIGHT

初始化代码程序主要流程概述

- (1) 定义程序进入点
- (2) 设置一场向量表
- (3) 初始化sdram控制器
- (4) 设置各种堆栈指针寄存器
- (5) 初始化各种关键的I/O设备
- (6) 初始化C程序需要的存储器
- (7) 必要的话，使能中断
- (8) 必要的话，改变处理器的运行模式
- (9) 进入C代码

FARSIGHT

NOR flash和Nand flash接口比较

- FLASH存储器又称闪存，主要有两种：Nor flash和Nand flash
- Nor flash带有通用的SRAM接口
- Nand flash器件使用复杂的I/O口来串行地存取数据

Nand flash移植要点

- 明确CPU在系统复位时能否获得nand flash中的指令
- 在初始化代码中实现nandflah的代码搬运功能
- 阅读README.nand，明确我们需要实现的nand flash接口函数
- 参照CPU nand flash控制器及nand flash芯片的硬件手册，实现系统要求我们实现的接口函数
- 配置系统支持nand flash命令

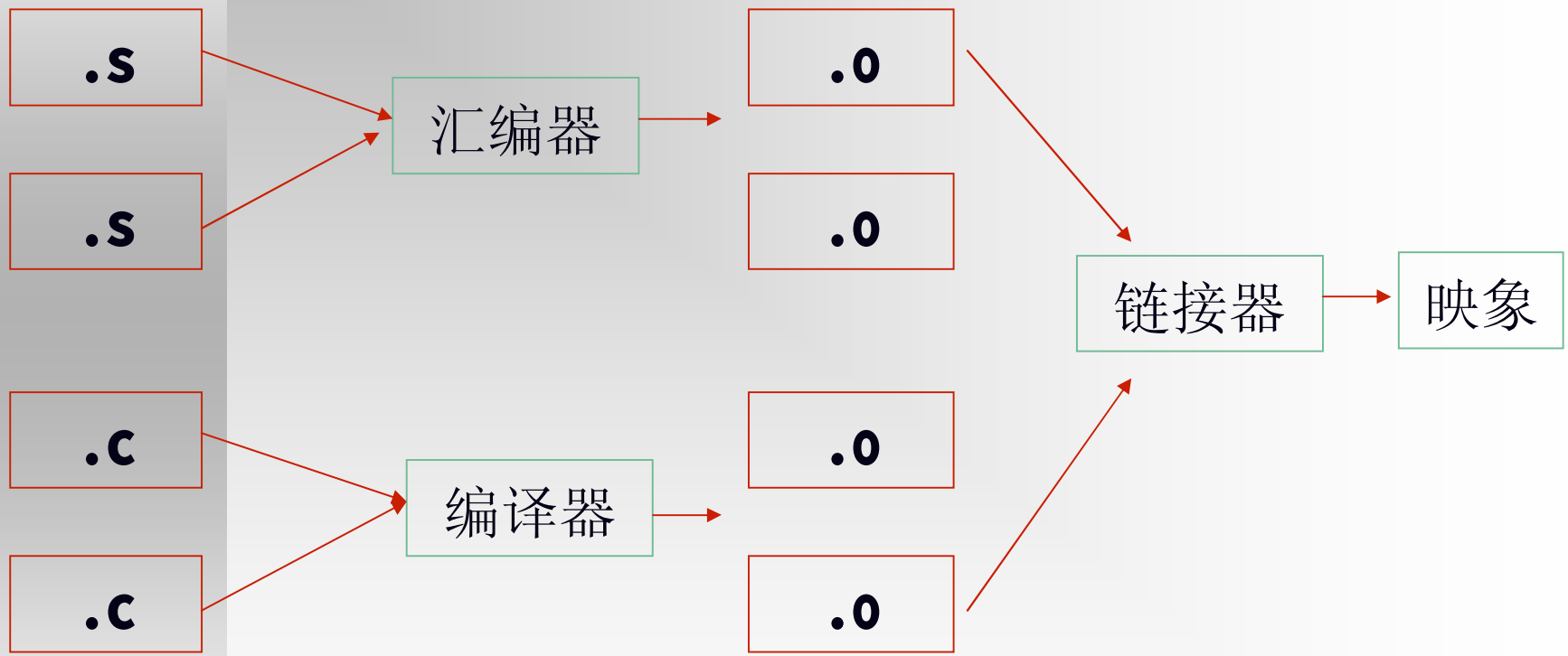
FARSIGHT

基于Nand flash的启动代码分析

- 根据源码讲解

FARSIGHT

Arm可执行映像的产生



Uboot源码的编译

- 针对你的目标平台配置 u-boot
  - make <BOARD_NAME>_config
    - make smdk2410_config
- 交叉编译
  - make CROSS_COMPILE = arm-linux-
- 生成的镜像文件
  - *System.map* The symbol map
  - *u-boot* U-Boot in ELF binary format
  - *u-boot.bin* U-Boot raw binary image
  - *u-boot.srec* U-Boot image in Motorola's S-Record format

解析u-boot.lds

```
OUTPUT_ARCH(arm)
ENTRY(_start)
SECTIONS
{
    . = 0x00000000;
    . = ALIGN(4);
    .text
……
```

FARSIGHT

Config.mk

- (1)smdk2410/config.mk

  TEXT_BASE = 0x33F80000

- (2)config.mk

  CPPFLAGS := DTEXT_BASE=$(TEXT_BASE） …

  CFLAGS := $(CPPFLAGS)　　…

FARSIGHT

深入分析u-boot启动代码

- 重新分析启动源码

FARSIGHT

加入u-boot命令方法

- U_BOOT_CMD(
  go, CFG_MAXARGS, 1, do_go,
  "go - start application at address 'addr'\n",
  "addr- start application at address 'addr'\n"
  " passing 'arg' as arguments\n"
  );
- int do_go (…)

FARSIGHT

u-boot命令的实现原理1/3

- U-boot.lds

```
__u_boot_cmd_start = .;
.u_boot_cmd : { *(.u_boot_cmd) }
__u_boot_cmd_end = .;
```

FARSIGHT

u-boot命令的实现原理2/3

- 在include/command.h中有:

```
#define U_BOOT_CMDU_BOOT_CMD(name,maxargs,\
  rep,cmd,usage,help)  \
 cmd_tbl_t __u_boot_cmd_##name Struct_Section =\
    {#name, maxargs, rep, cmd, usage, help}

#define Struct_Section  __attribute__ ((unused,section\
    (".u_boot_cmd")))
```

FARSIGHT

u-boot命令的实现原理3/3

- **common/main.c**

```
int run_command (const char *cmd, int flag)
{       …
                find_cmd(argv[0])
        …
}
```

- **common/command.c**

```
cmd_tbl_t *find_cmd (const char *cmd)
{    …
for (cmdtp = &__u_boot_cmd_start;cmdtp !=
&__u_boot_cmd_end;cmdtp++) …
 }
```

FARSIGHT

Any questions?

lht@farsight.com.cn

FARSIGHT