



Bootloader 在嵌入式linux系统的使用

王海宁

版权



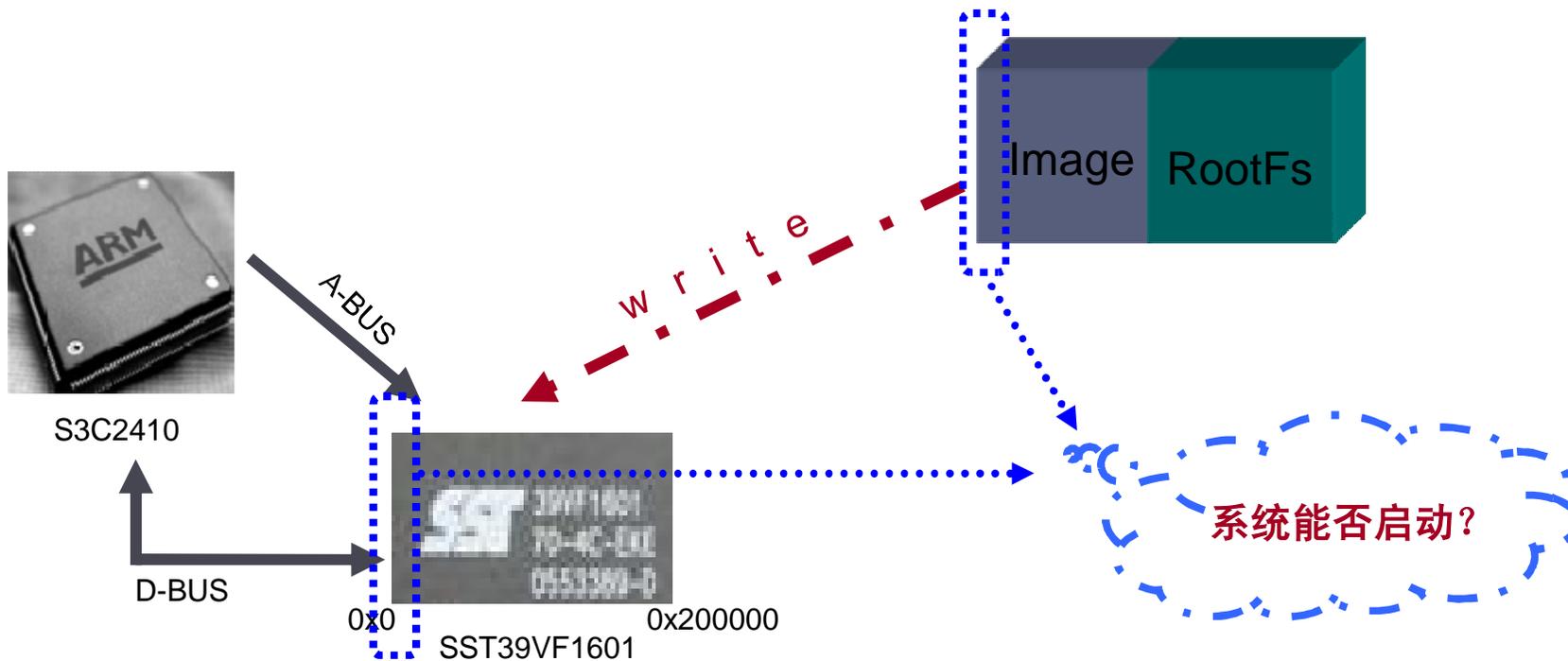
- } 华清远见嵌入式培训中心版权所有；
- } 未经华清远见明确许可，不能为任何目的以任何形式复制或传播此文档的任何部分；
- } 本文档包含的信息如有更改，恕不另行通知；
- } 保留所有权利。

内容大纲

- } **Bootloader**在嵌入式linux设备中的作用
 - } 嵌入式linux系统的组成
 - } S3C2410的简介
- } **u-boot**概述
 - } u-boot代码框架分析
 - } u-boot 引导初始化
 - } u-boot 链接脚本
- } **U-boot**加载linux内核
 - } linux内核参数传递

Bootloader在嵌入式linux设备中的作用

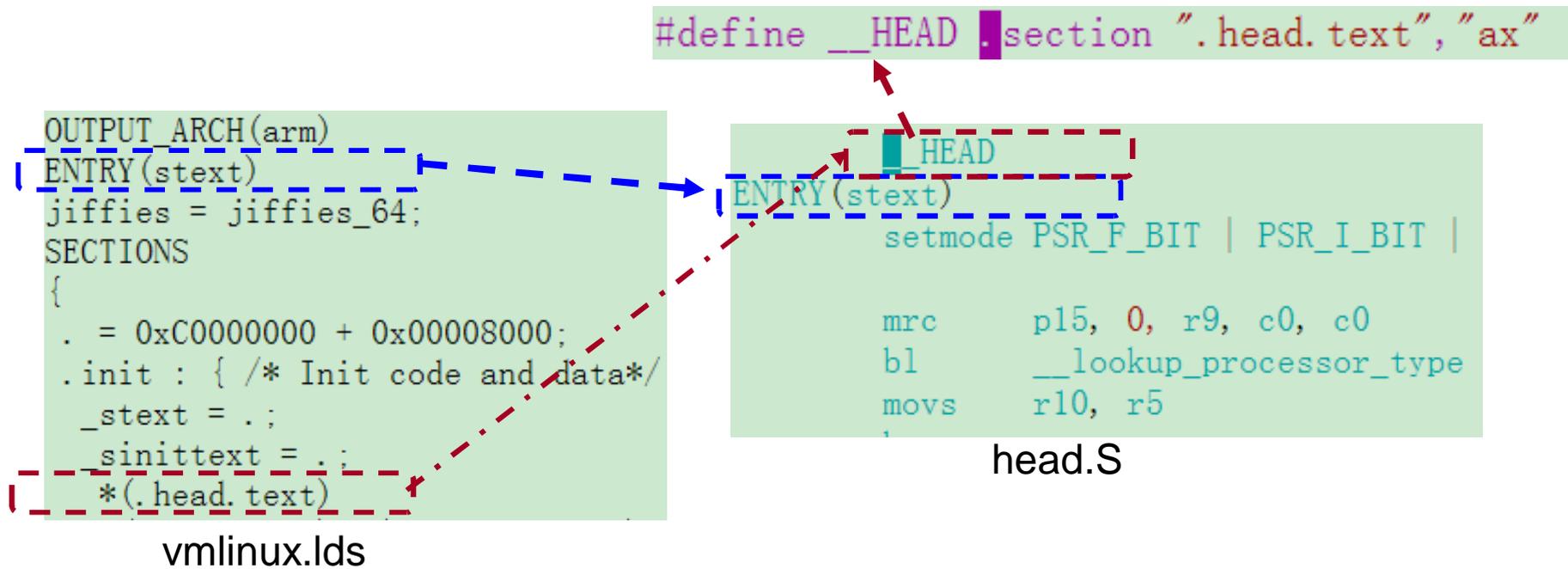
- } 嵌入式linux系统的组成
 - } linux内核Image (core)
 - } linux根文件系统 (app)



Bootloader在嵌入式linux设备中的作用

} 未压缩内核镜像Image的起始部分:

- } 链接脚本: arch/arm/kernel/vmlinux.lds
- } 起始部分Makefile: arch/arm/kernel/Makefile
- } 起始文件head.S: arch/arm/kernel/head.S

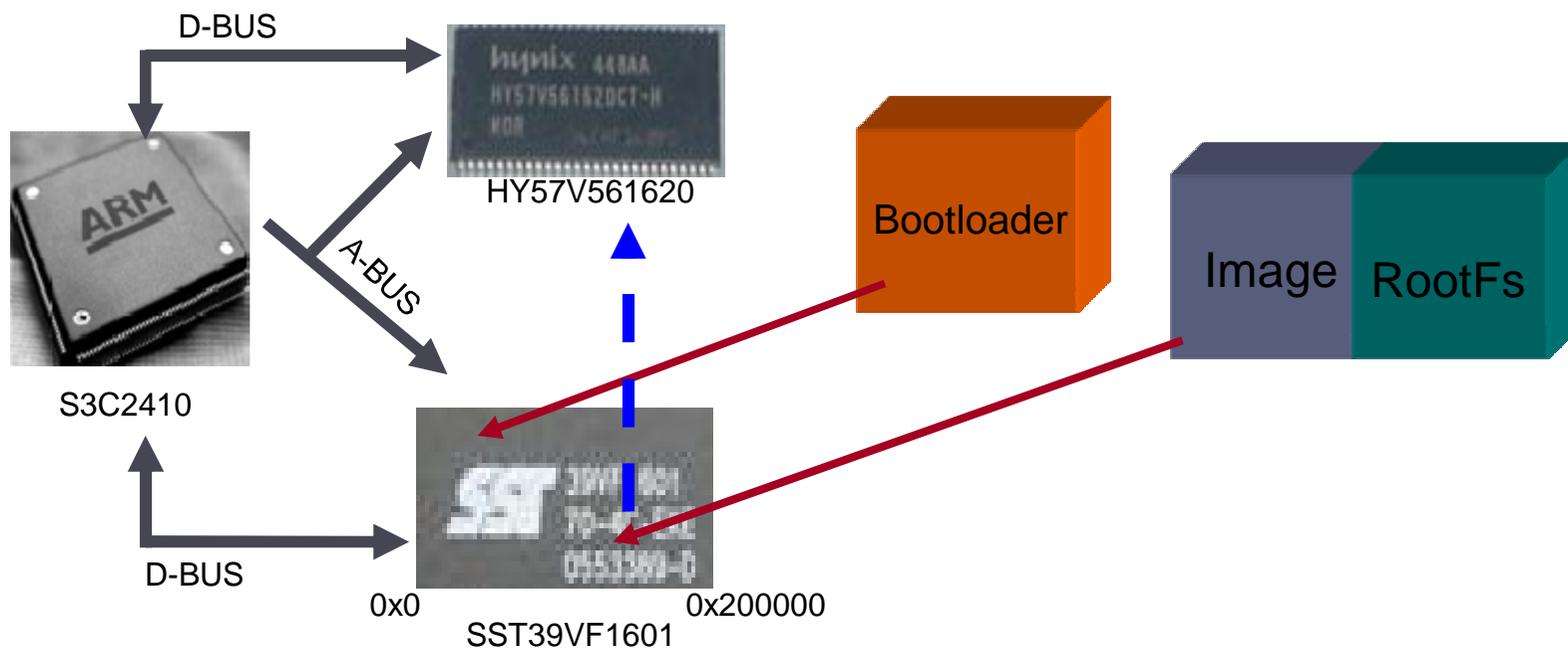


Bootloader在嵌入式linux设备中的作用

} linux内核镜像不具备自启动性

- } 内核镜像的起始地址没有异常向量表
- } 内核镜像的起始函数，需要r0,r1,r2这3个寄存器初始化相应的值

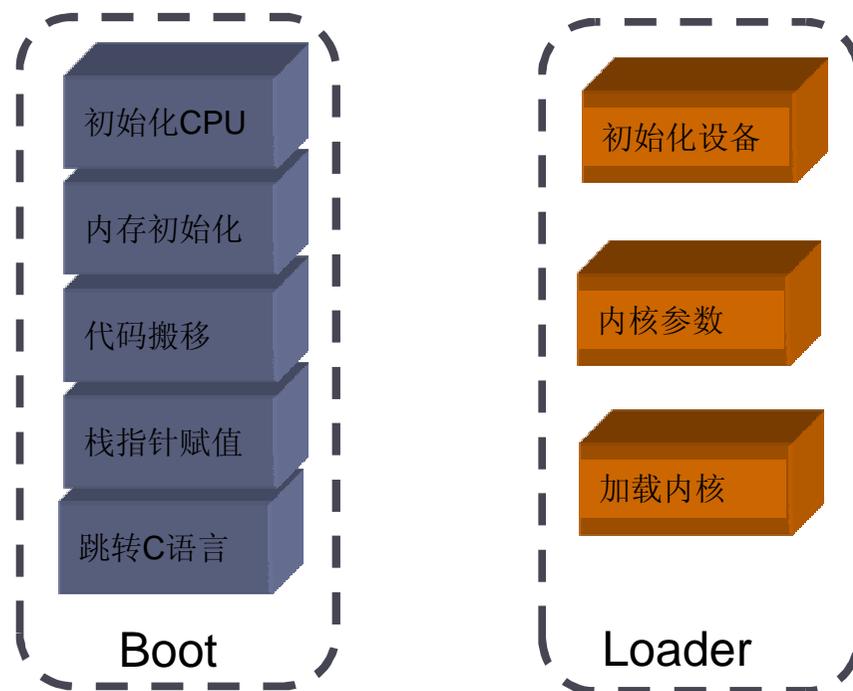
} linux内核镜像需要被动加载



Bootloader在嵌入式linux设备中的作用

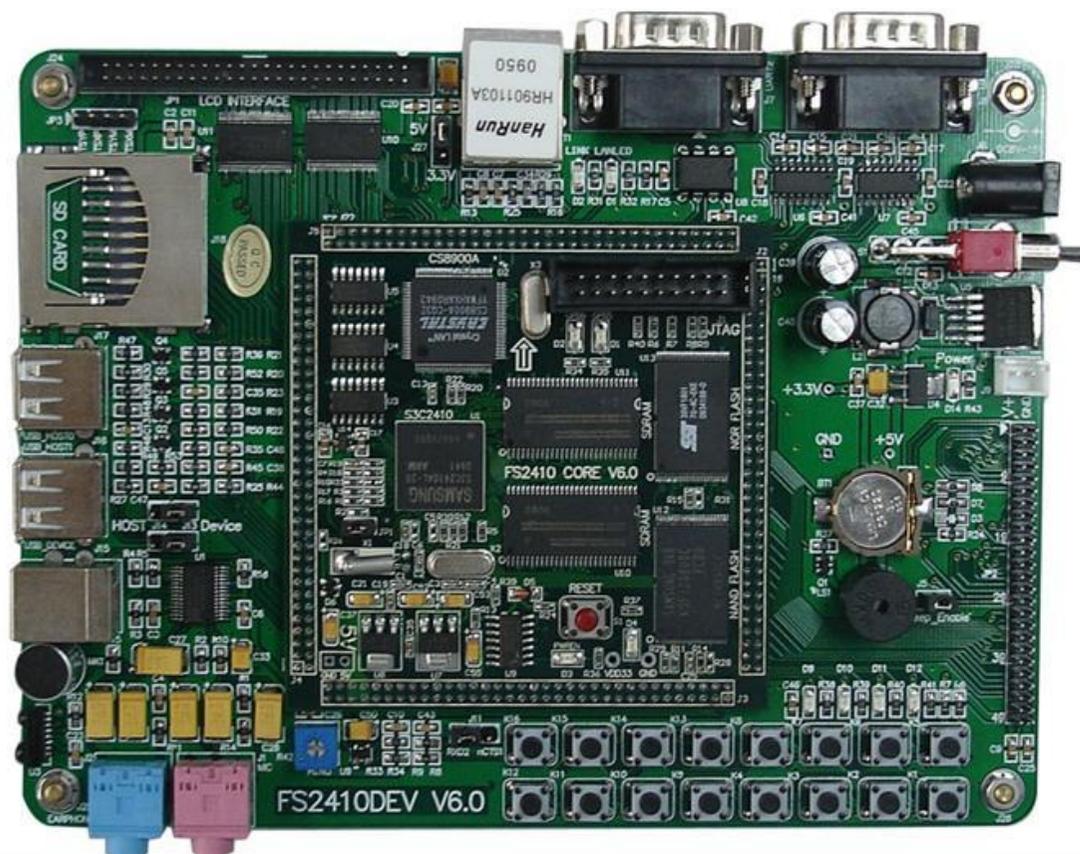
} Bootloader就是这样一段程序：

- } 在开发板启动时，第一个执行的程序
- } 初始化开发板硬件资源，为C代码提供相应环境
- } 加载内核镜像，初始化内核启动环境，启动内核



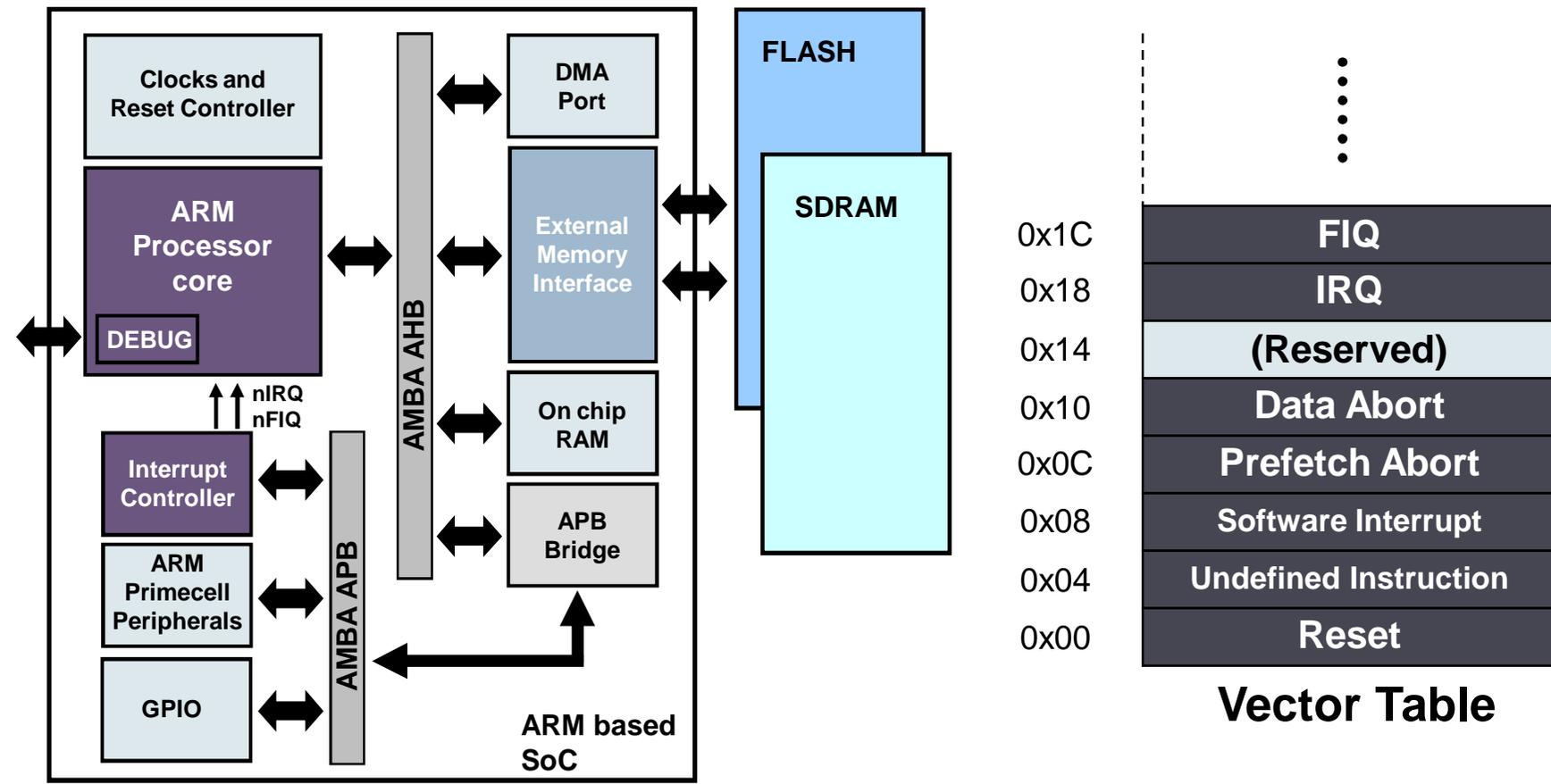
Bootloader在嵌入式linux设备中的作用

- } Bootloader依赖具体CPU体系和外围设备
- } FS2410开发板的情况：



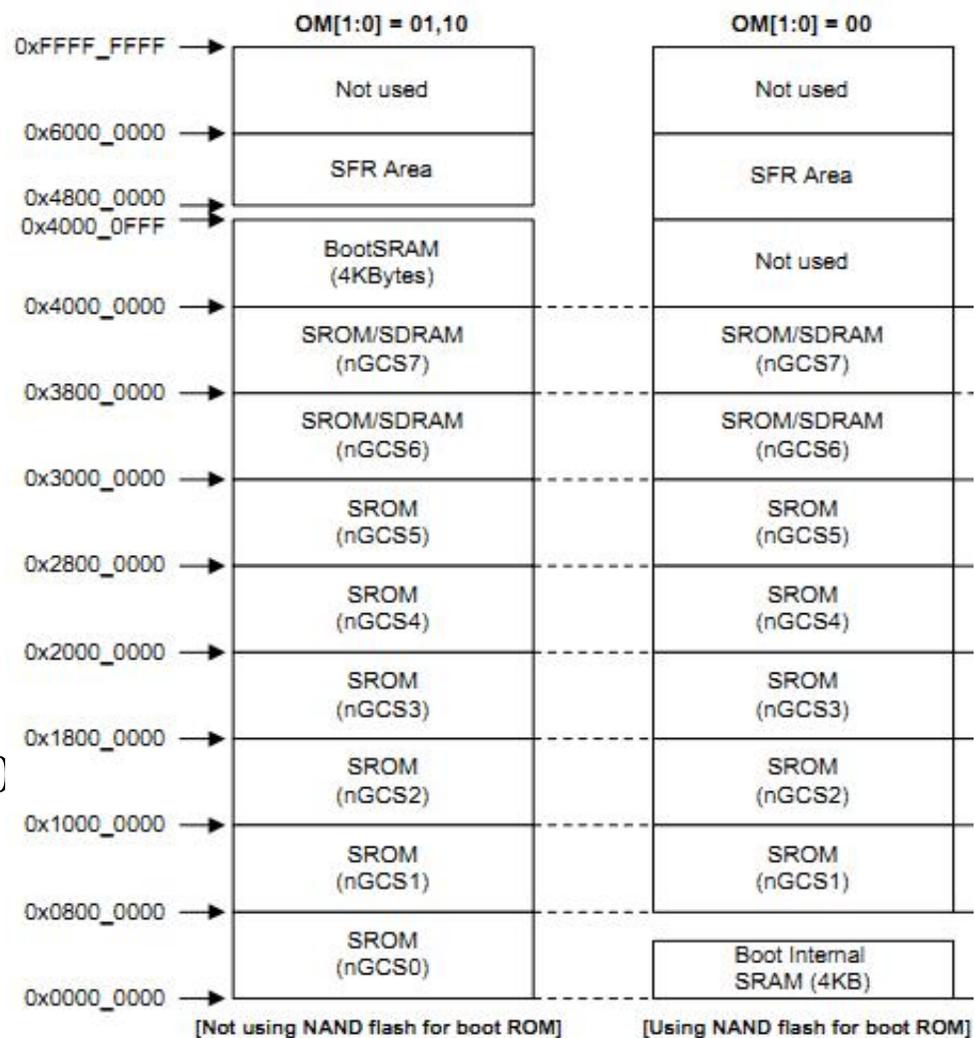
FS2410开发板情况

} CPU: ARM920T



FS2410开发板情况

- } Nor-Flash
 - } SST39VF1601
- } SDRAM
 - } HY57V561620
- } 内存映射:
 - } code:
 - } 0x0 ~ 0x20 0000
 - } data:
 - } 0x3000 0000 ~ 0x3400 0000



U-boot 概述

- } u-boot(Universal Boot Loader)是德国DENX小组开发的用于多种嵌入式CPU的bootloader程序。遵循GPL条款。
- } U-boot的特点
 - } 代码结构清晰、易于移植（见程序结构）
 - } 支持多种处理器体系结构（见程序结构cpu目录）
 - } 支持众多开发板（目前官方包中有200多种，见程序结构board目录）
 - } 命令丰富、有监控功能
 - } 支持网络协议、USB、SD等多种协议和设备
 - } 支持文件系统
 - } 更新较活跃，使用者多，有助于解决问题
- } U-boot下载：
 - } <ftp://ftp.denx.de/pub/u-boot/>

U-boot 概述

} U-boot目录结构分类:

- } 1、平台相关的或开发板相关的
- } 2、通用的函数
- } 3、通用的设备驱动
- } 4、uboot工具、实例程序、文档

} Porting U-boot

- } 配置
- } 编译
- } 烧写



U-boot 启动分析

} 配置U-boot

} 关键参考点:

} 1、顶层Makefile

- .. make smdk2410_config

```
smdk2410_config :      unconfig
    @$(MKCONFIG) $(@:_config=) arm arm920t smdk2410 samsung s3c24x0
```

- .. 实际执行:

- .. mkconfig smdk2410 arm arm920t smdk2410 samsung s3c24x0

} 2、顶层mkconfig

- .. shell脚本

- .. 在include目录下，建立软连接，构建asm目录到asm-arm目录等

- .. 生成config.h，其中间接包含configs/smdk2410.h的板级开发包配置文件

U-boot 启动分析

} 编译U-boot

} 编译规则（编译规则、链接规则等）

} 顶层Makefile会包含一些相关的config.mk文件，该文件里面定义了很多编译要使用的变量值。

} 链接规则

```
166 # load other configuration  
167 include $(TOPDIR)/config.mk
```

Makefile



```
ifdef ARCH  
sinclude $(TOPDIR)/lib_$(ARCH)/config.mk  
endif
```

config.mk



```
LDSCRIPT := $(SRCTREE)/cpu/$(CPU)/u-boot.lds
```

lib_arm/config.mk

U-boot 启动分析

} 链接脚本 [cpu/arm920t/u-boot.lds](#)

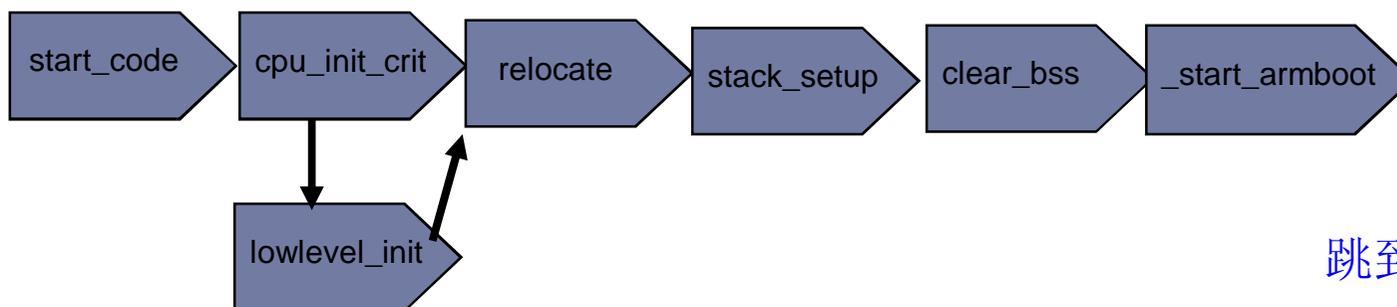
```
OUTPUT_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")
OUTPUT_ARCH(arm)
ENTRY(_start)
SECTIONS
{
    . = 0x00000000;

    . = ALIGN(4);
    .text :
    {
        {
            cpu/arm920t/start.o (.text)
            *(.text)
        }
    }
}
```

U-boot 启动分析

} Boot过程 (stage1) 【cpu/arm920t/start.s】

- } 硬件设备初始化
- } 为加载 Boot Loader的stage2准备RAM空间
- } 拷贝 Boot Loader的stage2到RAM 空间中
- } 设置好堆栈
- } 跳转到 stage2 的 C 入口点



U-boot 启动代码

} 跳到stage2阶段的代码

```
        ldr    pc, _start_armboot  
  
_start_armboot: .word start_armboot
```

```
} lib_arm/board.c  
void start_armboot (void)  
{  
    .....  
    size = flash_init ();  
    display_flash_config (size);  
    .....  
    for (;;)   
        main_loop ();  
}
```



U-boot加载linux内核

- } 在stage 2阶段，u-boot的主要目的就是启动内核
- } 需要由boot_loader设置的参数有：
 - } 内核参数，如页面大小、根设备
 - } 内存映射情况
 - } 命令行参数
 - } initrd映像参数
 - } 起始地址，大小
 - } Ramdisk参数
 - } 解压后的大小



U-boot加载linux内核

} 调用方法:

} 直接跳转到内核的第一条指令处，也即RAM中内核被加载的地址处

} 对于ARM Linux系统，在跳转之前必须满足:

1. CPU 寄存器的设置:

1. R0=0;
2. R1=机器类型 ID;
3. R2=传递给内核的启动参数起始地址;

2. CPU 模式:

1. 必须禁止中断（IRQs和FIQs）;
2. CPU必须处于SVC 模式;

3. Cache 和 MMU 的设置:

1. MMU 必须关闭;
2. 指令 Cache 可以打开也可以关闭;
3. 数据 Cache 必须关闭;

华清远见培训相关课程

- } 嵌入式Linux工程师就业班
- } 嵌入式linux工程师精英班 (周末班)
- } 嵌入式Linux系统开发班
- } ARM开发培训班(ATC)
- } 嵌入式Linux驱动开发班

谢谢!

