专题4-Bootloader设计

一、Bootloader设计蓝图

Bootloader	Boot parameters	Kernel	Root filesystem

1.1、Bootloader作用
用于系统上电时开始执行，初始化硬件设备，准备好软件环境，最后调用操作系统内核。

1.2、Bootloader的设计方法
模仿U-Boot（Universal Boot Loader）

1.3、U-Boot
可以引导多种操作系统，支持多种架构的CPU

1.4、使用Source Insight建立U-Boot代码工程
具体操作需要实际练习

二、ARM处理器启动流程（S3C6410）

2.1、Stepping Stone 垫脚石（8kB）

2.2、启动方式

Table 3-1. Device operating mode selection at boot-up

XSELNAND	OM[4:0]	GPN[15:13]	Boot Device	Function	Clock Source
1	0000X	XXX	RESERVED	RESERVED	XXTIpll if OM[0] is 0. XEXTCLK if OM[0] is 1.
1	0001X			RESERVED	
1	0010X			RESERVED	
1	0011X			RESERVED	
X	0100X		SROM(8bit)	-	
X	0101X		SROM(16bit)	-	
0	0110X		OneNAND[1]	Don't use NAND Device	
X	0111X		MODEM	Don't use Xm0CSn2 for SROMC	
X	1111X	000	IROM[2]	SD/MMC(CH0)	
0		001		OneNAND	
1		010		NAND(512Byte, 3-Cycle)	
1		011		NAND(512Byte, 4-Cycle)	
1		100		NAND(2048Byte, 4-Cycle)	
1		101		NAND(2048Byte, 5-Cycle)	
1		110		NAND(4096Byte, 5-Cycle)	
X		111		SD/MMC(CH1)	

Note 1) Only 6410X PoP D type doesn't support OneNAND booting.

Note 2) 6410X PoP A type doesn't support IROM booting based on NAND Flash. 6410X PoP D type doesn't support IROM booting based on OneNAND Flash.

2.3、地址分布

2.2 DEVICE SPECIFIC ADDRESS SPACE

Table 2-2. Device Specific Address Space

Address		Size(MB)	Description	Note
0x0000_0000	0x07FF_FFFF	128MB	Booting Device Region by XOM Setting	Mirrored Region
0x0800_0000	0x0BFF_FFFF	64MB	Internal ROM	
0x0C00_0000	0x0FFF_FFFF	64MB	Stepping Stone (Boot Loader)	
0x1000_0000	0x17FF_FFFF	128MB	SROMC Bank0	
0x1800_0000	0x1FFF_FFFF	128MB	SROMC Bank 1	
0x2000_0000	0x27FF_FFFF	128MB	SROMC Bank 2	
0x2800_0000	0x2FFF_FFFF	128MB	SROMC Bank 3	
0x3000_0000	0x37FF_FFFF	128MB	SROMC Bank 4	
0x3800_0000	0x3FFF_FFFF	128MB	SROMC Bank 5	
0x4000_0000	0x47FF_FFFF	128MB	Reserved	
0x4800_0000	0x4FFF_FFFF	128MB		
0x5000_0000	0x5FFF_FFFF	256MB	DRAM Controller of the Memory Port1	
0x6000_0000	0x6FFF_FFFF	256MB		

2.4、NAND FLASH时启动流程
    首先把IROM的内容（固化好的代码，BL0）复制到0x0地址处，然后执行IROM的流程，如以下内容:
2.2 iROM(BL0) boot-up sequence
Perform the following steps for iROM (BL0) boot-up:
1. Disable the Watch-Dog Timer
2. Initialize the TCM. (Please refer to "memory map" section of chapter 2.4)
3. Initialize the Block Device Copy Function. (Please refer to "Device Copy Function" section of chapter 2.6)
4. Initialize the stack region (Please refer to "memory map" section of chapter 2.4)
5. Initialize the PLL. (Please refer to "clock configuration" section of chapter 2.7)
6. Initialize the instruction cache
7. Initialize the heap region. (Please refer to "memory map" section of chapter 2.4)
8. Copy the BL1 to the stepping stone region (Please refer to "Device Copy Function" section of chapter 2.6)
9. Verify the integrity of BL1
10. Jump to the stepping stone
Note: ECC error and bootloader verification fail are referred to chapter 4

三、U-Boot工作流程分析

3.1、程序入口

    3.1.1、根据顶层目录Makefile查找板级信息
1956 forlinx_sd_ram256_config :   unconfig
1957   @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 SD ram256

    3.1.2、在板级目录（board/samsung/smdk6410/）通过链接文件查找起始文件
26 OUTPUT_ARCH(arm)
27 ENTRY(_start)
28 SECTIONS
29 {
30   . = 0x00000000;
31
32   . = ALIGN(4);
33   .text   :
34   {
35   cpu/s3c64xx/start.o   (.text)
36   cpu/s3c64xx/s3c6410/cpu_init.o   (.text)
37   cpu/s3c64xx/onenand_cp.o   (.text)
38   cpu/s3c64xx/nand_cp.o (.text)
39   cpu/s3c64xx/movi.o (.text)
40   *(.text)
41   lib_arm/div0.o
42   }

可知起始文件在 cpu/s3c64xx/ 目录下，源文件是start.S。又根据入口标号 ENTRY(_start) 在start.S中找到入口为

```
.globl _start
_start: b reset
ldr pc, _undefined_instruction
ldr pc, _software_interrupt
ldr pc, _prefetch_abort
ldr pc, _data_abort
ldr pc, _not_used
ldr pc, _irq
ldr pc, _fiq
```

G-Boot架构设计

1. 第1阶段程序设计

1.1. 核心初始化
- 1.1.1. 设置中断向量表
- 1.1.2. 设置处理器为svc模式
- 1.1.3. 关闭看门狗
- 1.1.4. 关闭所有中断
- 1.1.5. 关闭mmu和cache
- 1.1.6. 外设基地址初始化（6410）

1.2. C语言编程环境设置
- 1.2.1. 设置堆栈
- 1.2.2. 清除bss段

1.3. LED初始化

1.4. 初始化系统时钟

1.5. 进行内存初始化
- 1.5.1. 内存初始化
- 1.5.2. 取消存储保护区（210）
- 1.5.3. iram-irom初始化（210）

1.6. 复制nand flash中的bl到内存中
- 1.6.1. 简单初始化nand flash
- 1.6.2. 复制代码到内存
- 1.6.3. 跳转到第2阶段入口

2. 第2阶段程序设计

2.1. mmu初始化

2.2. 中断初始化
- 2.2.1. 中断初始化
- 2.2.2. 按键初始化

2.3. 初始化串口
- 2.3.1. 串口初始化
- 2.3.2. 移植printf函数

2.4. 网卡初始化

2.5. LCD初始化
- 2.5.1. 触摸板的初始化
- 2.5.2. LCD初始化

2.6. 解析执行用户命令
- 2.6.1. 移植tftp命令
- 2.6.2. 移植bootm命令