最简单的bootloader的编写-布布扣-bubuko.com

时间：2018-12-27 14:41:48 阅读：131 评论： 收藏：0 [点我收藏+]

标签：内核 最简 看门狗 col get 加载 阶段 light art

目标：写出bootloader的第一阶段代码和第二阶段代码，并测试。

最简单的bootloader的编写步骤:

1. 初始化硬件：关看门狗、设置时钟、设置SDRAM、初始化NAND FLASH
2. 如果bootloader比较大，要把它重定位到SDRAM
3. 把内核从NAND FLASH读到SDRAM
4. 设置"要传给内核的参数"
5. 跳转执行内核

1. 第一阶段：

编写start.S程序，主要用于初始化硬件：关看门狗、设置时钟、设置SDRAM、初始化NAND FLASH等

1.1 关看门狗

```
1     /* 关看门狗 */
2     ldr r0, =0x53000000
3     mov r1, #0
4     str r1, [r0]
```

s3c2440的看门狗寄存器地址为：0x53000000；1：把0x53000000这个地址写到r0中了 这时ldr是一个伪指令； 2：将数值0送入r1中；3：将r1里面的值，复制到以r0里面的值作为地址的内存里面。

1.2 设置时钟

```
      #define S3C2440_MPLL_200MHZ     ((0x5c<<12)|(0x01<<4)|(0x02))
 1    /* 设置时钟 */
 2    ldr r0, =0x4c000014
 3    mov r1, #0x03;            // FCLK:HCLK:PCLK=1:2:4, HDIVN=1,PDIVN=1
 4    str r1, [r0]
 5
 6    /* 如果HDIVN非0，CPU的总线模式应该从“fast bus mode”变为“asynchronous bus mode” */
 7    mrc    p15, 0, r1, c1, c0, 0          /* 读出控制寄存器 */
 8    orr    r1, r1, #0xc0000000            /* 设置为“asynchronous bus mode” */
 9    mcr    p15, 0, r1, c1, c0, 0          /* 写入控制寄存器 */
10
11    /* MPLLCON = S3C2440_MPLL_200MHZ */
12    ldr r0, =0x4c000004                   //MPLLCON寄存器地址
13    ldr r1, =S3C2440_MPLL_200MHZ
14    str r1, [r0]
```

这里以200MHz为例， FCLK:HCLK:PCLK分别为：50MHz，100MHz，200MHz。

1.3 初始化**SDRAM**

```
      #define MEM_CTL_BASE    0x48000000
1     /* 初始化SDRAM */
2     ldr r0, =MEM_CTL_BASE
3     adr r1, sdram_config     /* sdram_config的当前地址 */
4     add r3, r0, #(13*4)
5 1:
6     ldr r2, [r1], #4
7     str r2, [r0], #4
8     cmp r0, r3
   sdram_config:
     .long 0x22011110  //BWSCON
     .long 0x00000700  //BANKCON0
     .long 0x00000700  //BANKCON1
     .long 0x00000700  //BANKCON2
     .long 0x00000700  //BANKCON3
     .long 0x00000700  //BANKCON4
     .long 0x00000700  //BANKCON5
     .long 0x00018005  //BANKCON6
     .long 0x00018005  //BANKCON7
     .long 0x008C04F4  // REFRESH
```

```
    .long 0x000000B1  //BANKSIZE
    .long 0x00000030  //MRSRB6
    .long 0x00000030  //MRSRB7
```

1.4 重定位 **:** 把bootloader本身的代码从flash复制到它的链接地址去 */

```
 1     ldr sp, =0x34000000
 2
 3     bl nand_init   //NAND Flash初始化，在c程序中实现
 4
 5     mov r0, #0
 6     ldr r1, =_start
 7     ldr r2, =__bss_start
 8     sub r2, r2, r1
 9
10     bl copy_code_to_sdram
11     bl clear_bss
```

1.5 执行**main**

```
1 /*  执行main */
2     ldr lr, =halt
3     ldr pc, =main   //调用C程序中的main函数
4 halt:
5     b halt          // 循环
```

第二阶段:

第二阶段的代码由C语言完成，包括:

1. 帮内核设置串口； 2. 从NAND FLASH里把内核读入内存； 3. 设置传入参数； 4. 跳转执行kernel;

问题:

怎样查找内核加载地址和大小?

在uboot命令行下使用mtd命令，bootloader、params、kernel、root文件系统的存放地址，以便后面写程序时查找加载地址和大小。

待续。。。。。。。。。。。。

最简单的bootloader的编写

标签：内核 最简 看门狗 col get 加载 阶段 light art

原文：https://www.cnblogs.com/lxl-lennie/p/10181192.html