ARM 启动代码的比较与实现

作者：
张新宇　张平
张石　孙萍
鲍喜荣
东北大学信息科学与工程学院

摘要：
本文对ARM处理器的启动代码作了比较与分析，重点阐述了它们的地址重映射机制。并且给出了在S3C2410X上实现NAND flash启动的一种方式。

关键词：
ARM、地址重映射、启动代码、Bootloader

一、引言

ARM是一个采用RISC体系结构的处理器内核，是一个IP核。众多的半导体厂商采用ARM内核加上自己的技术生产出种类繁多的微处理器芯片。如Atmel公司的AT91M55800A；三星公司的S3C4510B、S3C44B0X、S3C2410X；Intel公司的SA-1110、PXA255等。由于它们采用ARM内核，所以它们的汇编指令集是相同的。这给开发人员带来了极大方便。同时，由于生产厂商的不同，各种处理器之间的差别也是显而易见的。首先表现在地址重映射机制上。应该说，这是理解ARM体系结构最重要的地方，也是一个难点。

二、ARM 地址重映射机制

ARM体系结构中，系统上电或复位后，处理器将从地址0x0处取第一条指令。因此，上电的时候，地址0x0处必须是非易失性的ROM或flash。但是，为了加快中断响应速度，方便更改中断向量表，有时需要把中断向量表复制到RAM中去，然后把RAM重新映射到地址0x0处，这就用到了地址重映射机制。因为地址重映射是在程序执行过程中进行的，必须考虑程序执行流程的连续性。

ARM处理器的地址重映射机制可分为三种情况：

1、处理器内部有专门的寄存器完成Remap，只需将Remap寄存器相应位置“1”，由硬件逻辑来完成地址的重新映射。

2、没有专门的Remap控制寄存器，需要重新改写用于控制Memory起始地址的Bank寄存器，来实现Remap。

3、没有Remap寄存器，而Bank寄存器的起始地址固定，这就需要MMU实现地址重映射。另外，有一些ARM处理器没有地址重映射机制。

由于处理器地址重映射机制不同，其具体实现方式也是不同的。AT91M55800A采用一个专用寄存器EBI_RCR(重映射控制寄存器)，通过向EBI_RCR的RCB（该寄存器的第0位）写“1”实现地址重映射。系统复位瞬间，系统将FLASH地址映射在0x0的位置，而将内部SRAM映射在0x00300000的位置。启动代码需将FLASH中的中断向量拷贝到内部SRAM中，然后，执行重映射命令，把内部SRAM的地址重新映射到0x0的位置，而将FLASH的地址映射到其他位置，如0x01000000(由NCS0片选，配置EBI——CSR0寄存器)的位置。这样便可以在内部SRAM中0x0开始的位置修改中断向量。同时，为了保证程序执行流程的连续性，必须调整PC指针，即调整到程序真正被连接的地方（0x01000000）去。这时，程序仍在FLASH中运行。

重映射命令前后AT91M55800A的存储器配置如图1：

▲ 图 1

三星S3C4510B最大可寻址空间

为64MB，它把64MB的地址空间分为10个块（Bank），每个块的起始地址和大小都可以通过寄存器来设置。上电时默认的是flash映射到地址空间为0x0000,0000～0x0200,0000的位置，这样，程序从flash中的启动代码开始执行。在启动代码中，把SDRAM映射到地址空间的其他位置，如0x0040,0000～（0x0140,0000-1）处。接着要做的一项工作是将flash中从0x0开始的程序一一对应地复制到SDRAM中从0x0040,0000开始的地址空间，然后，把SDRAM映射到从0x0开始的地址空间，flash映射到其他位置，如从0x0100,0000开始的地址空间，至此，完成了地址重映射。而它的PC指针不需要调整，只是这时的PC指针已经指向SDRAM中了。

三星S3C44B0X的地址空间分成8个块，Bank0~Bank6的起始地址固定，大小可变。Bank7虽起始地址可变，但那只是为了保证与Bank6地址连续。因为Bank6、Bank7主要用于DRAM。S3C44B0X没有提供地址重映射机制。

三星S3C2410X把1GB的地址空间分为8个块（Bank0~ Bank7），它们的起始地址固定，大小可变。S3C2410X没有进行地址重映射的控制寄存器，但它有内存管理单元MMU。MMU把物理地址空间映射到虚拟地址空间，通过这种方式可以实现地址重映射。

Intel公司StrongARM SA-1110和Xscale PXA255微处理器存储区域分成若干块，每个块的起始地址固定。它们都没有地址重映射机制，它们是通过存储器管理单元来完成地址映像的重映射。

三、S3C2410X的NAND flash启动

S3C2410X有两种启动方式：NOR flash启动和NAND flash启动。

- **NOR flash启动：**

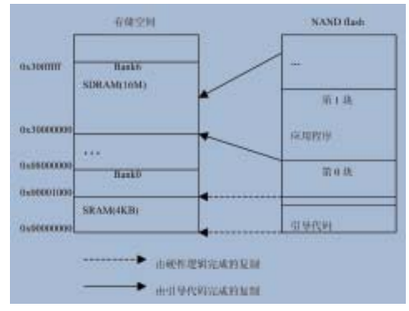
  NOR flash可以象SDRAM那样随机读取，且读取速度快，不但可以存储程序，还可以运行程序。上电复位时，NOR flash被映射到地址0x0处，程序就可以从NOR flash中的第一条语句开始执行。程序即可以在NOR flash运行，也可以复制到SDRAM中运行。

- **NAND flash启动：**

  NAND flash容量大，价格低，广泛应用于嵌入式系统中。S3C2410X集成了NAND flash控制器，可方便编程。但是，NAND flash随机读取速度慢，需专用I/O接口，只能存储程序，无法运行程序。为了能够从NAND flash启动，上电复位时，S3C2410X通过硬件逻辑把NAND flash的前4K的内容复制到片内SRAM中，而片内SRAM被映射到地址0x0，这样就可以从地址0x0处取到有效指令。因此，采用NAND flash启动时，必须利用片内SRAM中的代码把NAND flash中的程序代码复制到SDRAM中去。

这里，笔者参考了Bootloader的方式，把代码分成两部分，第一部分作为引导代码，包括ARM所要求的连续8个字的中断向量表。它主要负责把NAND flash中的程序代码复制到SDRAM中，其代码很小，远小于4K。第二部分是应用程序。调试时可分别单独调试，互不影响。调试完成后，分别烧写，引导代码的RO_Base设置为0x0，烧写到NAND flash的第0块。第二部分应用程序的RO_Base设置为0x3000,0000，烧写到NAND flash第一块开始的地址空间。

上电复位时，引导代码由硬件逻辑复制到片内SRAM中，于是，ARM所要求的连续8个字的中断向量表就位于0x0地址开始的连续空间内。接着从第一条指令开始执行，除了一些必要的初始化以及设置中断向量表，它把NAND flash中第一块开始的程序代码复制到起始地址为0x3000,0000的SDRAM中。地址0x3000,0000既是RO_Base的地址，也是SDRAM在整个地址空间的起始地址。复制完成后，引导代码也该结束退出了，退出之前需调整PC指针，为了简单起见，把PC指针直接调整到地址0x3000,0000，即从应用程序的启动代码开始执行。需要注意的是，引导代码把中断向量表复制到SDRAM中_ISR_STARTADDRESS处即地址0x30ffffff，而应用程序的启动代码也把中断向量表复制到SDRAM中同一地址_ISR_STARTADDRESS处。这样，当异常发生时，PC指针首先跳到地址0x0开始8个字的异常向量表，这是在片内SRAM中，是在引导代码里。接着，跳到SDRAM中的中断向量表，这是在应用程序里，然后转到中断处理程序，PC指针的跳转跨越了两部分程序。引导代码和应用程序在存储空间和NAND flash的分布情况如图2：



▲ 图2

实现复制和PC指针调整的代码如下：

```
        IMPORT    nand read ll              ; 引入外部 C 函数

ldr     r0,       =SDRAM Base Address       ; r0 指向 SDRAM 的基地址，即地址 0x3000,0000
mov     r1,       =APP Start  Address       ; r1 指向 NAND  Flash 中应用程序的开始地址
                                            ; 即第 1 块的开始地址
mov     r2,       =APP End  Address         ; r2 指向 NAND  Flash 中应用程序的结束地址
bl      nand read ll                        ; 调用复制函数开始复制

ldr     r12,      =SDRAM Base Address
mov     pc,  r12                            ; pc 指向 SDRAM 的基地址
                                            ; 引导代码到此结束，接着从应用程序启动代码的第一条语句开始执行。
```

▲ 图 3

同样道理，NOR flash 启动也可以采用这种方式，不同的是，引导代码在NOR flash中，它把NOR flash中的程序代码复制到SDRAM中，然后，也是跳到SDRAM中开始执行。

四、总结

采用 NAND flash 启动，可以节省 NOR flash 芯片，即降低成本、功耗，又减小电路板尺寸。本文实现的引导代码为采用 NAND flash 启动提供了一种简单可行的方法。

该引导代码已应用到三导联移动心电监护项目中，此项目作为第九届“挑战杯”飞利浦全国大学生课外学术科技作品竞赛参赛项目，已获得辽宁省金奖，并将于 2005 年 11 月份去上海参加全国总决赛。

参考文献

1 马忠梅等.AT91 系列 ARM 核微控制器结构与开发[M].北京:北京航空航天大学出版社,2002

2 陈章龙等.嵌入式系统 –Intel StrongARM 结构与开发[M].北京:北京航空航天大学出版社,2002

3 陈章龙等.嵌入式技术与系统 –Intel XScale 结构与开发[M].北京:北京航空航天大学出版社,2004

4 李驹光等.ARM 应用系统开发详解[M].北京:清华大学出版社,2003

5 杜春雷.ARM 体系结构与编程[M].北京:清华大学出版社,2003

6 曹伟等.地址重映射在 S3C4510B 系统中的实现[J]. 单片机与嵌入式系统应用,2004.3

7 费浙平.基于 ARM 的嵌入式系统程序开发要点(二).[J].单片机与嵌入式系统应用,2003.9