文章编号:1000 - 582X(2005)11 - 0035 - 04

基于 S3C4510B 的嵌入式系统启动设计*

李培毅,甘育裕
(重庆大学 通信工程学院,重庆 400030)

摘 要:近年来各种新型 32 位嵌入式系统设备不断涌现.但由于 32 位嵌入式系统的不通用性,使得各个系统设备设计者都必须设计独有的启动软件.作者剖析了 32 位微处理器 S3C4510B 的特性及嵌入式系统的基本结构和启动程序的特性,并详细地论述了启动程序的整个流程以及其中的地址重映射的实现过程和启动代码中异常向量的处理等关键步骤,设计出具有良好模块性和可移植性的基于 S3C4510B 嵌入式系统的启动程序.测试结果表明该设计具有很好的稳定性和高效性.

关键词:引导加载程序;地址重映射;异常向量
中图分类号:TP731.11 文献标识码:A

1 嵌入式系统组成

嵌入式系统通常由包括微控制器及外围设备的硬件和包括引导加载程序、操作系统及应用程序的软件两大部分组成,如图 1 所示.三星公司设计的 S3C4510B 是 32 位嵌入式网络微处理器,其内含一个由 ARM 公司设计的低功耗、高性能 16/32 位 ARM7TDMI RISC 处理器核,适用于对价格及功耗敏感的应用场合.由 Linux 改造而来的免费嵌入式操作系统 ucLinux 因其可靠的稳定性、良好的兼容性、可移植性、开放性,以及无需内存管理单元(MMU)的支持,故已经成为低端嵌入式操作系统首选.S3C4510B 控制器加载 ucLinux 即可构成价廉物美的嵌入式网络设备.

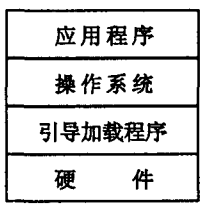

图 1 嵌入式系统结构示意图

2 引导加载程序(Bootloader 程序)

一个嵌入式系统的软件通常可以分为 3 个部分:引导加载程序,嵌入式操作系统,用户应用程序.引导加载程序是系统加电后运行的第一段软件代码[1].简单地说,就是在操作系统内核运行之前运行的一段小程序(在嵌入式系统中通常叫做 Bootloader).通过这段小程序,可以初始化硬件设备、建立内存空间的映射图,从而将系统的软硬件环境配置到一个合适的状态,以便为最终调用操作系统内核设置适当的环境.而在嵌入式系统中,通常并没有像 PC 机中 BIOS 那样的固件程序(有的嵌入式 CPU 也会内嵌一段短小的启动程序),因此整个系统的加载启动任务就完全由 Bootloader 来完成.由于每个用户对嵌入式 CPU 的应用千差万别,每个用户对系统的启动有不同的要求,所以 Bootloader 程序一般都由用户根据自己的实际要求编写.

3 系统存储器映射(System Memory Map)

S3C4510B 采用了统一编址的方式,将系统的片外存储器、片内存储器、特殊功能寄存器和外部的 I/O 设备,都映射到 64 MB 的地址空间[2].同时,为便于管理,又将地址空间分为如若干个存储器组,可以通过配置相关特殊功能寄存器,设定每个存储器组的大小和位置.在上电或系统复位后,所有组的地址指针寄存器都被初始化到其缺省值.这时,所有的组指针(ROM/SRAM/Flash 组 0 和特殊功能寄存器组除外)都被清零,即所有这些存储器组在系统启动时都处于禁用状

* 收稿日期:2005 - 05 - 20
基金项目:重庆市信息产业发展资金支持项目(200313003)
作者简介:李培毅(1975 -),男,重庆人,重庆大学助教,硕士,主要从事计算机网络技术的研究.

态.用户在进行程序设计时,应当首先通过配置相应特殊寄存器来定义各个存储器组系统的地址空间.

S3C4510B 内 ROM/SRAM/Flash 组 0 寄存器中的尾指针和基指针上电后的初始值分别为 0x200 和 0x00.这意味着一旦系统上电或复位后将自动设置 ROM/SRAM/Flash 组 0 的地址空间为从 0x0000000 开始的 32 MB 内.ROM/SRAM/Flash 组 0 的这种初始化定义使得系统在上电或复位后,将系统的控制权交给了由用户编写的启动代码,当然这些启动代码必须存放在 ROM/SRAM/Flash 组 0 控制的外部 Flash ROM 中.当起动代码运行时,它主要执行各种系统初始化任务,并根据应用系统的外部存储器和外设的实际情况来重新配置系统的存储器地址空间.而特殊功能寄存器组的基址指针在系统上电复位时被初始化为 0x3FF0000,一般不再改动.

开始启动时,存有启动代码的 Flash ROM 被映射到 0x0000000 地址,系统从此开始运行.但在实际应用中,为提高系统的实时性,加快代码的执行速度,系统启动后程序往往要被搬移到存取速度要比 Flash ROM 快得多的 SDRAM 中,以此大幅提高系统的实时性能.又由于 S3C4510B 芯片中的异常中断入口地址被固定在 0x0000000 开始的 8 个字(每个字包括 4 个字节)中,因而系统只能将存储器进行地址重映射(Remap),把 SDRAM 映射到 0x000000 地址处,把 Flash ROM 的地址映射到系统高端地址.

S3C4510B 硬件系统有 4 MB 的 Flash ROM,使用 16 位数据总线;16 MB 的 SDRAM,使用 32 位数据总线.如图 2 所示,Remap 前系统地址 0x0000000 ~ 0x03FFFFF 分配给 Flash ROM, 0x0400000 ~ 0x13FFFFF 分配给 SDRAM; Remap 后 0x0000000 ~ 0x0FFFFFF 分配给 SDRAM, 0x1000000 ~ 0x13FFFFF 分配给 Flash ROM.

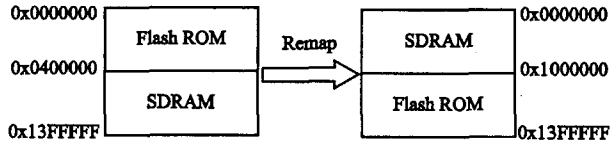

图 2 地址重映射示意图

4 S3C4510B 系统启动程序设计

图 3 为 S3C4510B 系统的启动程序流程图.

4.1 存储器初始化

存储器初始化主要是配置如前文所述的用于实现地址空间和芯片内外存储介质映射的特殊寄存器.根据本系统情况,其配置如下[3]:

```
SYSCFG = 0xE7FEFF80
EXTDBWTH = 0x3002
ROMCON0 = 0x04000060
DRAMCON0 = 0x14010010
```

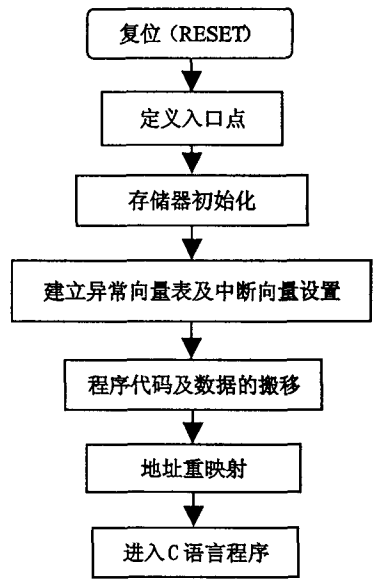

图 3 启动程序流程图

4.2 异常中断处理

异常中断的处理在嵌入式系统中异常重要需要特别注意.在 S3C4510B 中,异常中断的入口地址是固定的,它们的地址被固定设置在 0x00000000 ~ 0x0000001C 之间,具体设置请查看数据手册.

为保证写入 SDRAM 中的异常处理程序的入口地址.异常向量表绝对不能被从 Flash ROM 搬移到 SDRAM 中的代码和数据所覆盖,为此,异常向量表一般被定义在 SDRAM 中的高地址部分.

地址重新映射之后,入口地址和异常中断服务程序都被搬移到 SDRAM 内.系统初始化时,必须将异常处理服务程序入口地址填入异常中断向量表内,下面是 FIQ 异常中断部分的源程序[4].

```
;//异常向量表定义//
VECTOR_ADDR EQU 0x0FFFD00
SYS_RST_VECTOR Field 4
UDF_INS_VECTOR Field 4
SWI_SVC_VECTOR Field 4
INS_ABT_VECTOR Field 4
DAT_ABT_VECTOR Field 4
RESERVED_VECTOR Field 4
IRQ_SVC_VECTOR Field 4
FIQ_SVC_VECTOR Field 4

;//异常初始化代码(FIQ 部分)//
    …
B FIQ_HANDLER ;0x1C
```

```
    …
FIQ_HANDLER
    SUB sp,sp,#4
    STMFD sp!,{r0}
    LDR r0, = FIQ_SVC_VECTOR ;//读取中断向量//
    LDR r0,[r0]
    STR r0,[sp,#4]
    LDMFD sp!,{r0,pc};//转到中断处理入口//

;//异常处理入口代码//
    …
SysFIQHandler
    IMPORT ISR_FIQ
    STMFD sp!,{r0 - r12,lr}
    BL ISR_FIQ  ;//跳转到 FIQ 中断服务程序//
    LDMFD sp!,{r0 - r12,lr}
    SUBS pc,lr,#4
    …
```

以上中断处理程序具有极好的灵活性,无论系统是否进行了地址重映射,异常中断向量都可以在运行时动态改变,并且在运行时可以指向不同的异常处理代码入口.

在系统打开中断前还应设置堆栈,而用户主要根据实际需求对将使用的工作模式中的堆栈指针进行设置.一般来说,管理者(SVC)和 FIQ 堆栈必须设置.但通常为保证 Remap 后程序运行正常,一般将 SDRAM 的高地址设为系统的堆栈区.

4.3 程序代码及数据的搬移

Remap 后, SDRAM 被映射到 0x0000000 ~ 0x0FFFFFF 的地址空间,而 Flash ROM 则被映射到高于 0x0FFFFFF 的地址上.为保证 Remap 后程序能够正常运行,整个过程必须确保 Flash ROM 中的代码和数据地址不变地被移到 SDRAM 中[5].

程序编译时链接器 ARMLink 将自动生成 RO 段、RW 段和 ZI 段 3 种段,同时还将产生这 3 种段的起始和终止定位信息: Image $ $ RO $ $ Base、Image $ $ RO $ $ Limit、Image $ $ RW $ $ Base、Image $ RW $ Limit、Image $ ZI $ Base 和 Image $ $ ZI $ $ - Limit.使用这些定位信息,将 Flash ROM 中的代码和数据搬移到 SDRAM 中,源程序如下:

```
    Start_Flash DCD  |Image $ $ RO $ $ Base|
    End_Flash   DCD  |Image $ $ RO $ $ Limit|
    Start_BSS   DCD  |Image $ $ RW $ $ Base|
    End_BSS     DCD  |Image $ $ RW $ $ Limit|
    Start_Zero  DCD  |Image $ $ ZI $ $ Base|
    End_Zero    DCD  |Image $ $ ZI $ $ Limit|
    ;//复制 Flash ROM 中的代码到 SDRAM 中//
;   LDR r0, = Start_Flash
    LDR r1, = End_Flash
    LDR r2, = Start_BSS
    LDR r3, = End_BSS
    SUB r1, r1, r0
    SUB r3, r3, r2
    ADD r1, r1, r3
    LDR r2, = 0x400000;//Remap 前 SDRAM 起始地址
COP LDR r3, [r0], #4
    STR r3, [r2], #4
    SUBS r1, r1, #4
    BNE COP
    LDR r0, = End_Flash
    LDR r1, = Start_BSS
    LDR r3, = Start_Zero
    CMP r0, r1
    BEQ LOOP1
;//复制 RW 段到 SDRAM//
LOOP CMP r1, r3
    LDRCC r2, [r0], #4
    STRCC r2, [r1], #4
    BCC LOOP
;//复制 ZI 段并清 0//
LOOP1 LDR r1, = End_Zero
    MOV r2, #0
LOOP2 CMP r3, r1          ;
    STRCC r2, [r3], #4
    BCC LOOP2
```

在实际应用系统中,由于 Flash ROM 存储器相对而言比较小速度慢且价格高,为了节约成本提高运行速度,通常将会把应用程序和在存储器中的数据先压缩然后再写入 Flash ROM 中同时将解压缩程序也写进去.当系统上电启动后,在搬移程序和数据时解压程序先将被压缩过的程序和数据解压后再搬到 SDRAM 的适当位置中.由于压缩解压程序比较复杂不在这里讨论.

4.4 地址的重映射

在本系统中只需重新设置特殊寄存器 ROMCON0 和 DRAMCON0,即可完成 S3C4510B 系统的 Remap.但如果其他系统还使用了另外的存储器组则必须再设置相应的特殊寄存器.下面是地址从映射的源程序:

```
LDR r1, =ROMCON0_RE ;//Remap 后的 Flash 地址
LDR r2, = DRAMCON0 _ RE;//Remap 后的 SDRAM 地址
LDR r0, =ROMCON0
STMIA r0, {r1 - r2}
```

4.5 进入 C 语言程序

经过以上步骤后通常用跳转语句(例如 B main)进入 C 语言程序. 此时所有的程序已经被搬到 SDRAM 中,在此处的用 C 语言编写的程序一般为用户的应用程序. 在系统中,C 语言程序一般先运行与系统密切相关的初始化程序或特殊外围设备的测试程序等,例如 I/O 测试程序,I^2C 器件初始化和自检程序. 而后运行操作系统的入口函数,将系统的控制权交给操作系统. 从而完成整个嵌入式系统的启动,可调用 uclinux 的启动函数 start_kernel 函数使得 uclinux 启动[6].

上面的步骤可以根据实际需要进行适当的添加或删节.

5 结束语

此启动程序在 ARM 开发软件 ADS1.2 下编译通过,并下载运行实际验证可行,完全达到设计要求. 此程序结构合理,模块性强,易于增删修改. 可以根据硬件系统的实际需求灵活加入各种特殊的应用程序,也可方便的移植到其他 S3C4510B 的系统中.

参考文献:

[1] 詹荣开. 嵌入式 BootLoader 技术内幕[EB/OL]. http://www-900.ibm.com/developerWorks/cn, 2003-12-25.

[2] 李驹光. ARM 应用系统开发详解—基于 S3C4510B 的系统设计[M]. 北京:清华大学出版社,2003.

[3] 杜春雷. ARM 体系结构与编程[M]. 北京: 清华大学出版社, 2003.

[4] SAMSUNG ELECTRONICS. htm. 32-bit RISC Microcontroller for Network Solution[EB/OL]. http://www.samsung.com/Products/Semiconductor/SystemLSI/CommunicationProcessor/S3C4510B/S3C4510B, 2001-10.

[5] 曹伟. 地址重映射在 S3C4510B 系统中的实现[EB/OL]. http://21ic.com/news/c63p8.aspx, 2004-8-20.

[6] 谭浩强. C 语言程序设计教程[M]. 北京:高等教育出版社,1992.

Design of Bootloader for Embedded Systerm on S3C4510B

LI Pei-yi, GUN Yu-yu

(College of Communication Engineering, Chongqing University, Chongqing 400030, China)

Abstract: Recent years, there emerged many new types of 32-bit embedded system services. But due to the non-commonality of 32-bit embedded system, the designers of every system service should design particular boot software. The authors analyze the characteristic of 32-bit microprocessor S3C4510B, and the basic structure of embedded system and the characteristic of boot. The authors discuss the realization process of remap during the boot process and the management of abnormal vector in boot code. At last, they design boot program based on S3C4510B embedded system with good modularity and transplantablity. The test result shows good stability and effectivity of this design.

Key words: bootloader; remap; abnormal vector

(编辑 吕赛英)