一、S3C2440的两种启动方式

（1）Nor Flash

对于从NorFlash启动，由于NorFLash可以直接在片上运行，所以可以直接从NorFlash启动即程序直接下载到NorFlash中。NorFlash基地址为0，片内RAM地址为0x4000000，CPU首先读出NorFlash上的第一条指令（前4字节），执行指令，然后CPU继续读出其他指令执行，当然这里一般会把程序运行的数据部分拷贝到SRAM或者SDRAM中；也可以把全部的程序拷贝到SDRAM中，然后跳到SDRAM中继续执行。

它的数据线和地址线分开，可以实现ram一样的随机寻址功能，可以读取任何一个字节，但是擦除仍要按块来擦。其容量也较小，大都是2～12M。

（2）Nand Flash会存在坏块 导致有些地址不能烧录程序

对于从NandFlash启动，本质上还是从SRAM中启动。（NandFlash不具有片上直接运行的功能）因为系统在NandFlash启动时，会自动从NandFlash中复制前4K的内容进入系统内部的SRAM中，然后从SRAM中执行程序（CPU从0地址去除第一条指令执行）。这前面的4KB（注意只有4KB），一般来说，需要把整个程序复制到SDRAM中，然后跳到SDRAM中执行。

一般来说，复位之后，程序开始执行党的第一条指令都是厂家固化在芯片内部存储区的一段代码。根据不同的功能，这代码有不同的名字。C51和STM32叫做ISP检测程序，主要用来下载程序之用。S3C2440叫做RBL(ROM boot loader)，它是SOC上电后开始运行的地方，它会判断是哪种启动方式，如果是nand启动，就会从nand的起始地址处读取UBL（user boot loader）并且复制到ARM的内存里面，也就是上面说的片内SRAM。（以前我一直以为是硬件自动加载的，原来还是软件部分加载的UBL）

nand flash同样是按块擦除，但是数据线和地址线复用，不能利用地址线随机寻址。读取只能按页来读取。它内部电路更简单，因此数据密度大，体积小，成本也低。因此大容量的flash都是nand型的。

1. Nor Flash和Nand Flash的烧写方式（H-JTAG J-Link OpenJTAG）

1.H-JTAG

（1）只能烧写Nor Falsh

（2）其硬件是物理电平的转换接口，软件是负责协议转换的调试代理工具（RDI->JTAG），软件包括H-JTAG SERVER 和H-FLASHER。其中，H-JTAG SERVER 实现调试代理的功能，而H-FLASHER则实现了FLASH 烧写的功能。

（3）H-JTAG支持所有基于ARM7 和ARM9的芯片的调试，并且支持大多数主流的ARM调试软件，硬件简单，速度较慢

2.J-Link

（1）他就是平常用的ARM仿真器

（2）做一个板，用此板直接接收来自IAR、KEIL和ADS等软件的调试命令，由此板做RDI->JTAG协议的转换。然后与目标板通信，这就是JLINK的工作原理，硬件复杂，速度较快

（3）烧写程序至Nand Flash J-Command

usb--------连接目标板

  r----------重启目标板

  halt-------停止cpu运行的程序

  loadbin----加载可执行的二进制文件

  g----------跳到代码段地址执行

  s----------单步执行（调试用）

  setpc-----设置pc寄存器的值（调试用）

  setbp-----设置断点

  Regs-------读寄存器组织

  wreg-------写寄存器

  mem--------读内存

  w4---------写内存

  power off mmu---关闭mmu，这个对于裸板调试很重要

  w4 cpsr,0x0000001f------切换到系统模式

  speed------设置jtag的传输速率

  rce 0,c0,c0,0-----设置cp15寄存器的第1个寄存器为0

第一：r //防止提示 CPU is not halted! 正确找打CPU

第二：speed 12000 //设置TCK为12M

第三：loadbin 文件地址（E:\u-boot.bin\_openjtag） 烧写首地址（例如：0x33f80000）--SDRAM

第三：setpc 烧写首地址（0x33f80000）

第四：g //运行程序

第五：h //查看寄存器 PC R0-R7 USR FIQ SVC ABT IRQ UND

顺序：init.bin(0)-> u-boot.bin\_openjtag(0x33f80000)-> u-boot.bin(0x30000000)->超级终端

以上是将.bin文件烧写进SDRAM，需借助u-boot.bin\_openjtag和超级终端，将u-boot.bin烧写进Nand Flash

超级终端：nand scrud //擦除整个Nand Flash

nand erase 0 0x30000 //擦除从0 到 0x30000（192KB）的Nand Flash扇区 0x40000（256KB）

nand write.jffs2 0x30000000 0 0x30000 // 将SDRAM首地址为0x30000000的u-boot.bin烧写到Nand Flash Black0区 大小为192KB

（4）烧写supervivi到Nor Flash J-Command

Init.bin(0x40000000)->u-boot.bin(0x33f80000)->supervivi(0x30000000)->超级终端

超级终端：

protect of all //解锁

erase 0 0x4000 //擦除Nor Flash扇区

cp.b 0x30000000 0x4000 //烧写到Flash中

除此之外 也可以通过J-Flash ARM 向Nor Falsh中烧入程序

3.OpenJTAG

（1）简介

OpenJTAG能用在所有支持GDB调试协议的工具上，比如IAR、Eclipse、winarm，不能用在支持RDI协议的工具上，比如ADS。结合OpenOCD.exe向S3C2440中烧录程序

（2）使用

OpenJTAG+OpenOCD+sjf24x0\_ft2232.exe+dnw.exe（友善之臂自己的exe），要烧录的镜像文件要和sjf24x0\_ft2232.exe在同一目录下

1. 命令终端（在SecureCRT下进行命令操作）
2. 2440.test程序

通过MiniTools软件和USB线进行烧录，该软件还可以给开发板刷系统

1. Nand Flash下的u-boot指令

类似于Linux下的命令行操作 可控制按键 灯 文件系统 网络连接 文件传输

四、Nand中的坏块

Nand Flash有坏块是它的天然缺点，无法避免。有的坏块多，有的少，纯属几率。

http://baike.baidu.com/link?url=Tw5ldEU9l3qtNwCpzC5NV2Ha\_jPeunjFtxMaTy9yrrCmRvoxGs4CTVf7i\_4kdLJaC\_s\_eRIn0Oo9zjlnvomvga

Nand Flash专用的YAFFS文件系统可以识别、管理坏块, 可以避免坏块对系统产生影响。

在u-boot中，可以执行 " nand bad " 命令查看有哪些坏块,

Nand Flash怎么标记某一个BLOCK是坏的?

它使用该BLOCK中第1个扇区的OOB数据中某一个字节来标记: 其值为0xff表示该BLOCK是好的， 其值为非0xff表示该BLOCK是坏的。

有时候我们会误写这个OOB区的值导致有些BLOCK被误认为是"坏块",

可以在u-boot中执行"nand scrub"后, 根据提示信息输入小写字母'y'并回车,

它会强制擦除整个Nand Flash(包括把OOB擦除为0xff), 这样就可以恢复被误标为坏块的区域了。