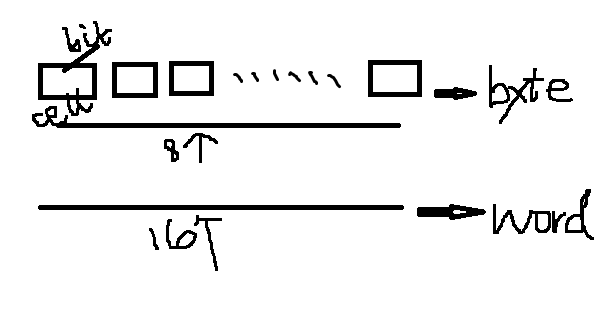
本文以S3C2410处理器和k9f1208系统为例，讲述NandFlash的读写方法。

NandFlash的数据是以bit 的方式保存在memory cell里的，一般来说，一个cell 中只能存储一个bit，这些cell 以8 个或者16 个为单位，连成bit line，形成所谓的byte(x8)/word(x16)，这就是NAND Device 的位宽。这些Line 组成Page， page 再组织形成一个Block。k9f1208的相关数据如下：



1block=32page；1page=528byte=512byte(Main Area)+16byte(Spare Area)。

总容量为=4096（block数量）\*32（page/block）\*512(byte/page)=64Mbyte

NandFlash以页为单位读写数据，而以块为单位擦除数据。按照k9f1208的组织方式可以分四类地址： Column Address、halfpage pointer、Page Address 、Block Address。A[0:25]表示数据在64M空间中的地址。

Column Address表示数据在半页中的地址，大小范围0~255，用A[0:7]表示；

halfpage pointer表示半页在整页中的位置，即在0~255空间还是在256~511空间，用A[8]表示；

Page Address表示页在块中的地址，大小范围0~31，用A[13:9]表示；

Block Address表示块在flash中的位置，大小范围0~4095，A[25:14] 表示；

二、读操作过程

K9f1208的寻址分为4个cycle（一次性取8个字节，所以寻址要4cycle）。分别是：A[0:7]、A[9:16]、A[17:24]、A[25]。

读操作的过程为: 1、发送读取指令；2、发送第1个cycle地址；3、发送第2个cycle地址；4、发送第3个cycle地址；5、发送第4个cycle地址；6、读取数据至页末。

K9f1208提供了两个读指令，‘0x00’、‘0x01’。这两个指令区别在于‘0x00’可以将A[8]置为0，选中上半页；而‘0x01’可以将A[8]置为1，选中下半页。

参考：<http://www.52rd.com/Blog/Detail_RD.Blog_gebi1218_20186.html>

对于BLOCK PAGE BYTE解释非常清楚的深度好文

二：NAND FLASH的基本操作接口函数

（1）在分析代码之前，必须弄清楚几个宏：

#define FLASH\_START\_ADDR (0) flash起始位置0x0

#define FLASH\_DATA\_SIZE (8\*1024\*1024) flash数据区域大小为8M，地址0x80000

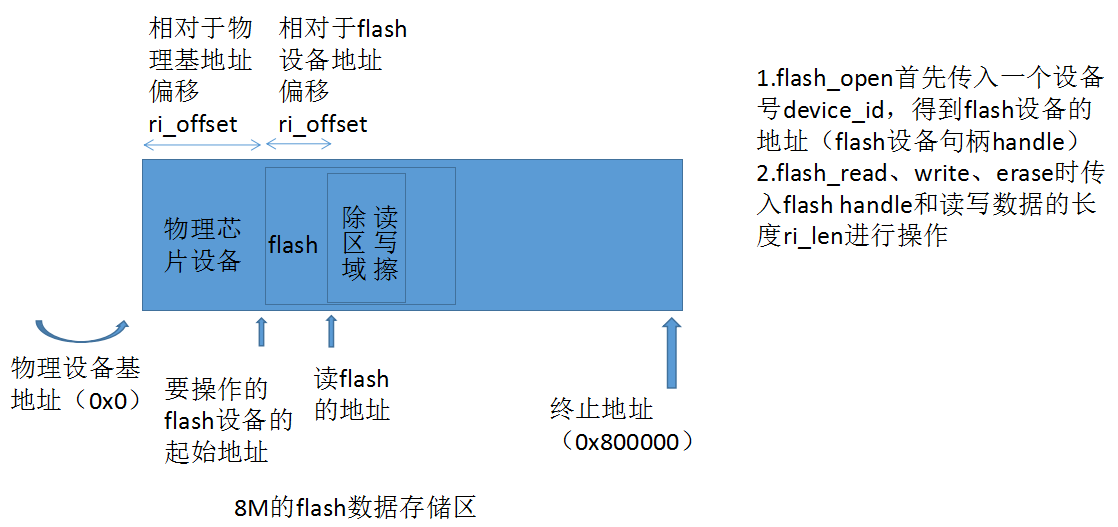
#define FLASH\_SECTION\_NUM\_MAX 20 一次性打开flash的大小，实际上只有一两个

#define CHARSTO\_SECTOR\_SIZE (64 \* 1024) 写入数据之前必须擦除块的大小为64K

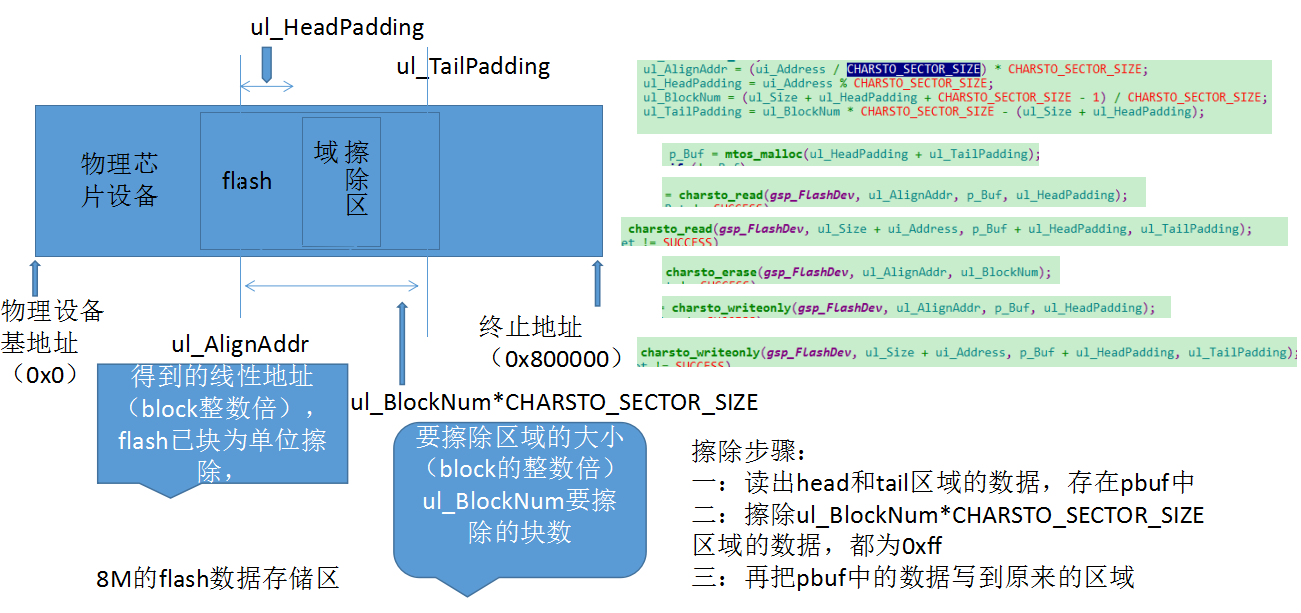
1block=64pages 1page=4sectors 1sector=512bytes

这里没有划分这么细，SECTION直接相当于块

1. 原理分析



擦除原理：



（3）相关接口函数

1. open 打开一个flash逻辑设备

传入参数：

renm\_DeviceId：设备号（0）

机顶盒可能不止内嵌一块物理芯片，可能有内部ram,外部ram,它们都有不同的设备号，一般从0、1、2、3·····编号，这里只有一块物理芯片，所以flash的设备号为0

ri\_Offset：相对于物理设备基地址的偏移

ri\_Len：打开flash区域的大小

rph\_FlashHandle：得到的flash的逻辑设备句柄



得到flash逻辑设备地址



1. read 从指定的flash逻辑设备读取数据

传入参数：

rh\_Handle FLASH设备句柄

ri\_Offset 读起始地址（相对FLASH 的偏移地址）

[out] rpuc\_Buffer 缓冲区指针，内存由调用者负责。

[in,out] rpi\_NumberToRead in:希望读取的字节数 out:实际读取的字节数





调用函数FLASH\_CheckParamter时：

ui\_Address = ri\_Offset + gstru\_FlashSection[i].i\_Offset;//得到从flash读取数据的地址（从flash哪个位置开始读取数据的）

1. Write 向指定的flash逻辑设备写入数据（需要以block为单位擦除数据）





调用函数FLASH\_CheckParamter时：

ui\_Address = ri\_Offset + gstru\_FlashSection[i].i\_Offset;//得到从flash写入数据的地址（从flash哪个位置开始写入数据的）

1. Erase 擦除某一段flash数据



ul\_AlignAddr = (ui\_Address / CHARSTO\_SECTOR\_SIZE) \* CHARSTO\_SECTOR\_SIZE;

//得到线性地址，必须是block的整数倍

ul\_HeadPadding = ui\_Address % CHARSTO\_SECTOR\_SIZE;

//读取位置到线性地址前端的长度

ul\_BlockNum = (ul\_Size + ul\_HeadPadding + CHARSTO\_SECTOR\_SIZE - 1) / CHARSTO\_SECTOR\_SIZE;

//擦除的块数

ul\_TailPadding = ul\_BlockNum \* CHARSTO\_SECTOR\_SIZE - (ul\_Size + ul\_HeadPadding);

//读取位置到线性地址后端的长度

ul\_HeadPadding=0，从ul\_AlignAdd处进行擦除，擦除长度为ri\_len

ul\_TailPadding=0，从（ul\_AlignAdd-ul\_HeadPadding）处开始擦除，一直擦除到ul\_AlignAdd+ul\_BlockNum \* CHARSTO\_SECTOR\_SIZE

如果ul\_HeadPadding=0，ul\_TailPadding=0时，擦除长度是ul\_BlockNum \* CHARSTO\_SECTOR\_SIZE

