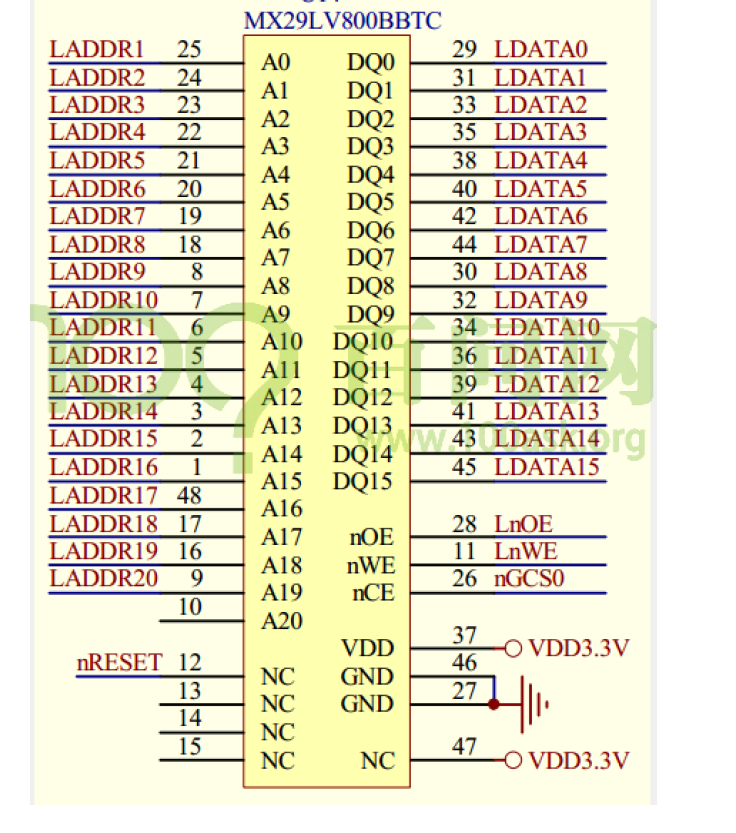
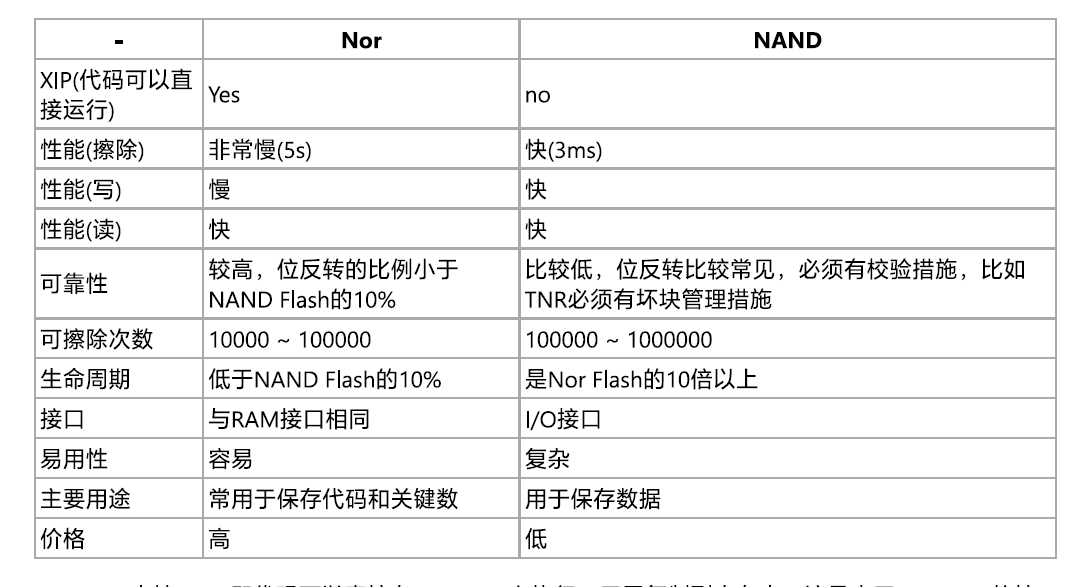
## nor简介

nor原理图如下



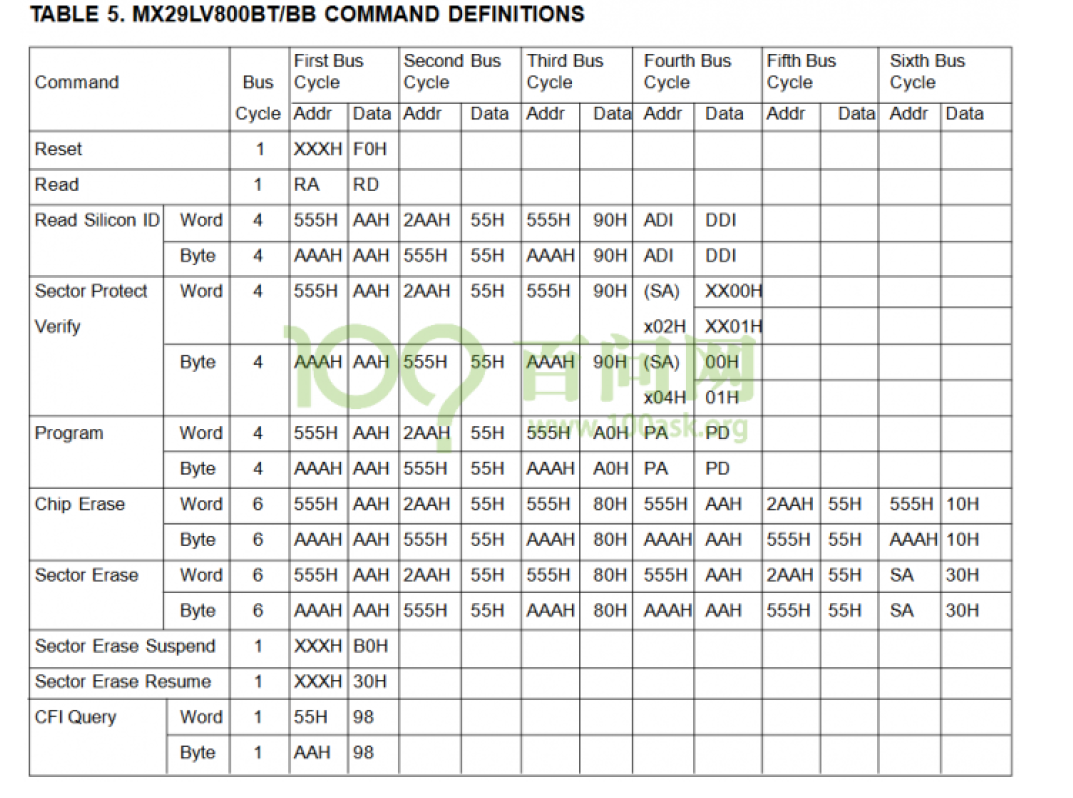
nor连接线有地址线，数据线，片选信号读写信号等，nor flash接口属于内存类接口，可以像内存一样读，但不能像内存一样写，需要做一些特殊操作

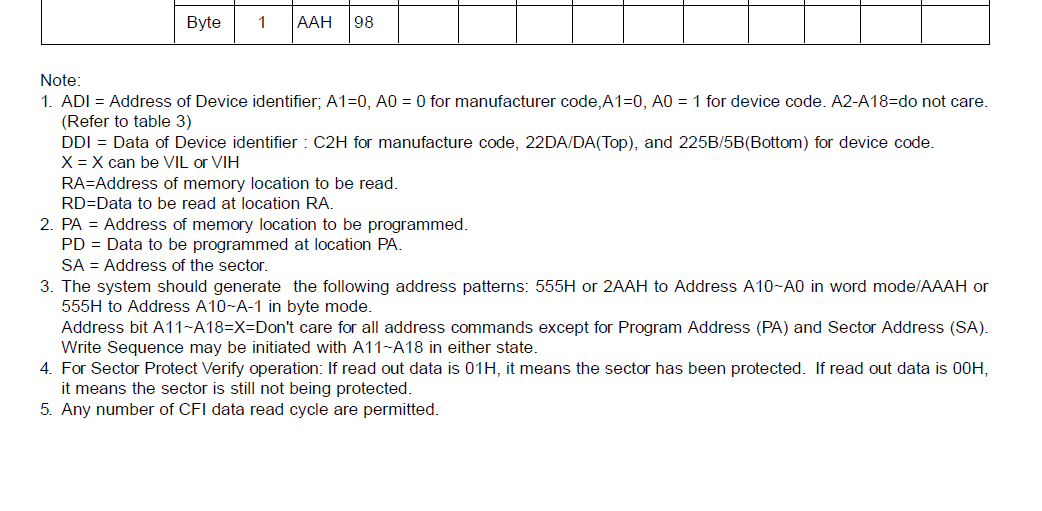
flash一般分为nor，nand两种



nor flash芯片手册命令表格，用于

操作

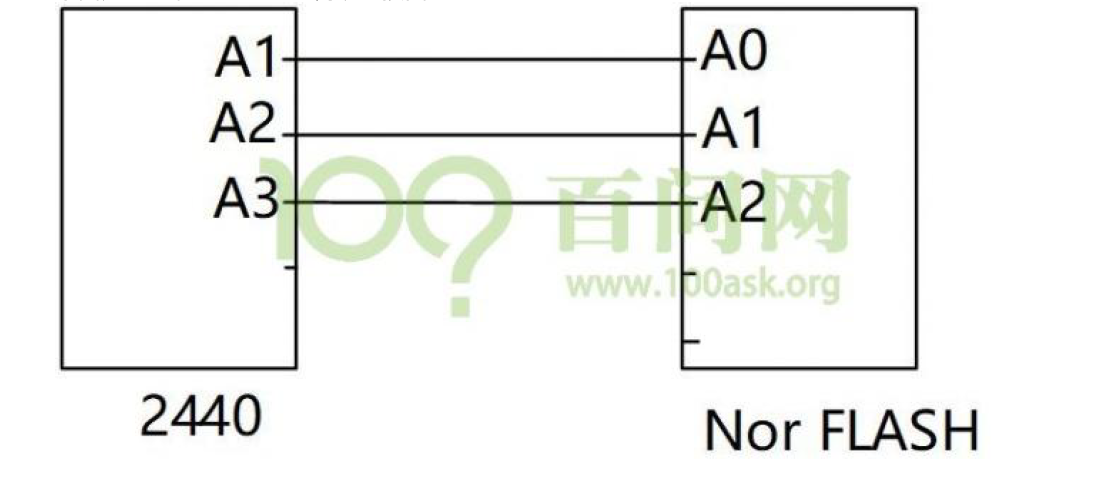




可以看到reset：往任意地址写F0，

read siliconl id ：往555H写AAH（解锁），往2AAH写55H（解锁），往555H写90H（命令），然后就可以读取ADI地址，读到数据为DDI。最后reset退出

下图为2440和nor flash的简易连接图

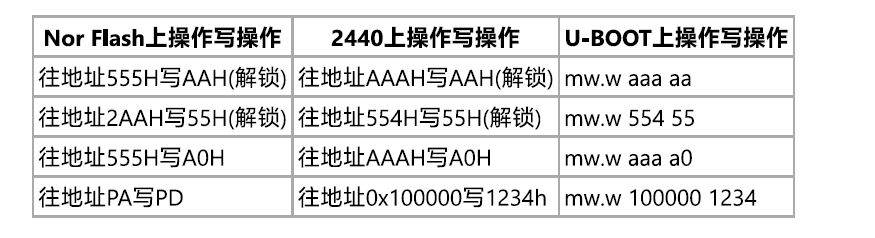


注意：2440的A1连接到NOR的A0，如果nor收到555H地址，那么2440发出的就是（555H << 1）地址

识别nor flash的两种规范：

1. jedec探测，就是在内核里事先定义一个数组，该数组里面放有不同厂家各个芯片的一些参数，探测的时候将flash的id和数组里的id一一比较
2. cfi探测，直接发各种命令来读取芯片的信息，

nor写操作，先擦除：



## nor编程（16字节）

实现擦除读写操作

首先看原理图，nor接到nGCS0，所以基地址为0

写发送命令函数（调用写数据函数），读取命令函数（调用读数据函数）

写nor\_test

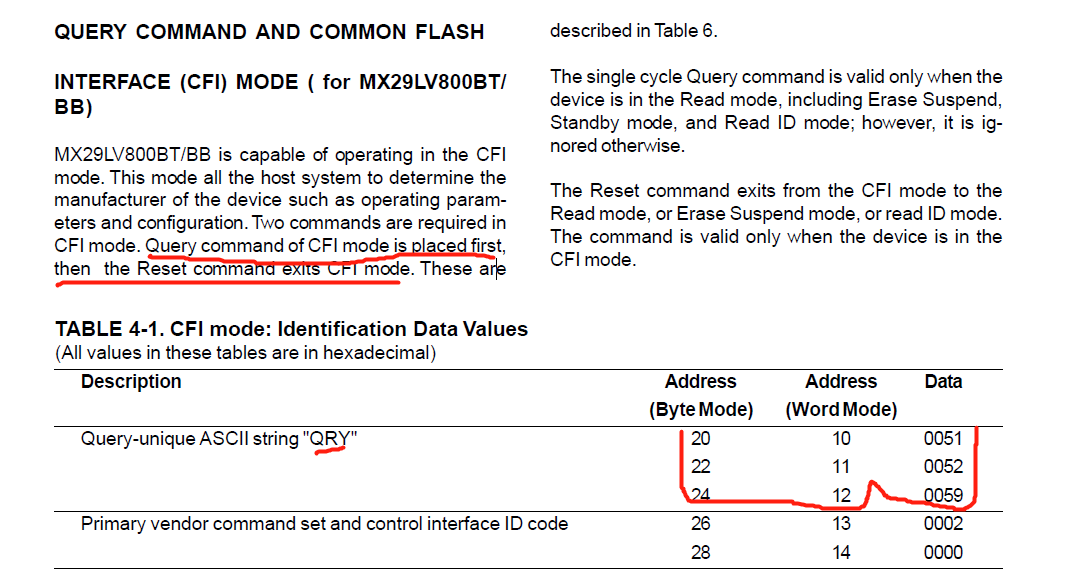
看操作表写擦除，读写函数

### scan函数：

读设备id

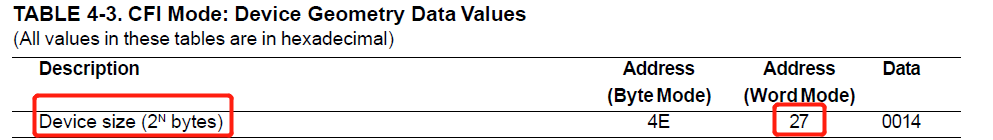
reset

进入cfi模式



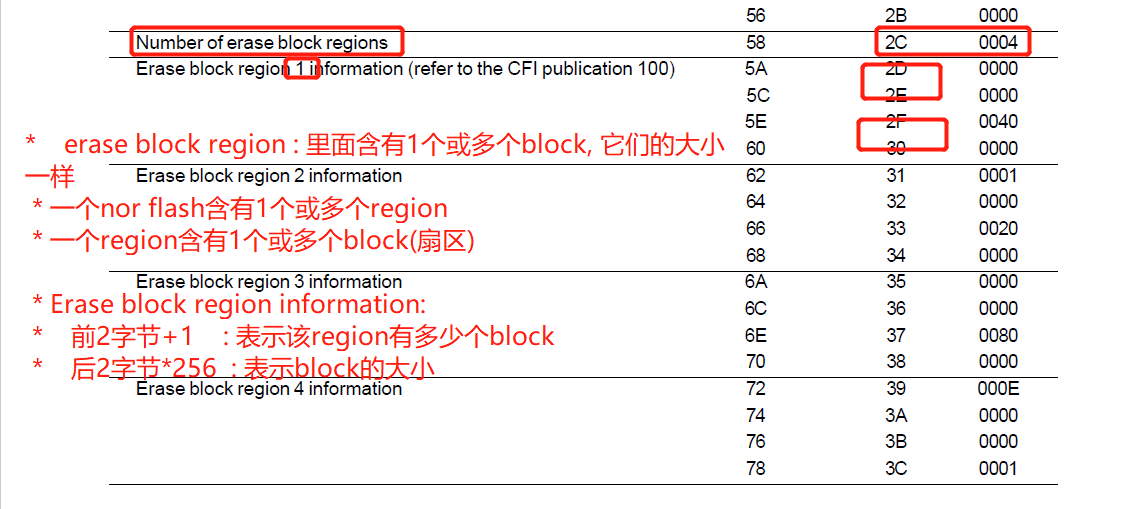
首先进入查询模式query command，可以根据表打印出内容做对比看是否成功

查询nor容量



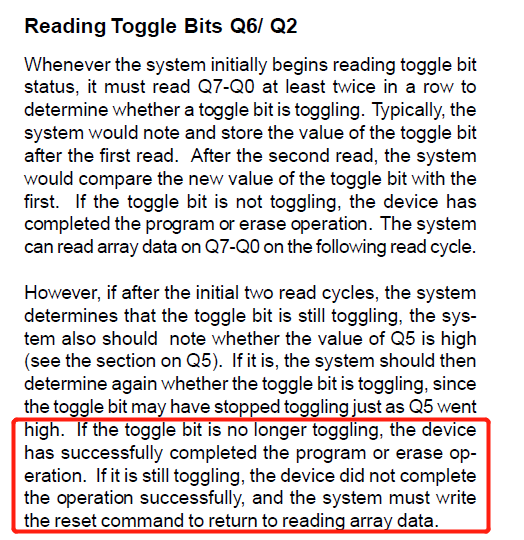
由于读取的是N的值，那么要将该值向左平移1位得出最终值

打印各个扇区起始地址



最后退出 reset

### wait\_ready函数



q6是切换位（toggle）

当q6不再变化时，表明设备完成擦除工作或完成program（write）

### erase函数

按照表来就行了，最后加wait\_ready函数

### write函数

获取地址，获取写入的字符，构造16位数据，按操作表来，最后wait\_ready

### read函数

不用对nor进行操作，打印64个数据，每行16个，后接可视不可视字符

由于文件太大，最终只能烧写到nand，并且进行擦除写入操作可能使按键中断失效