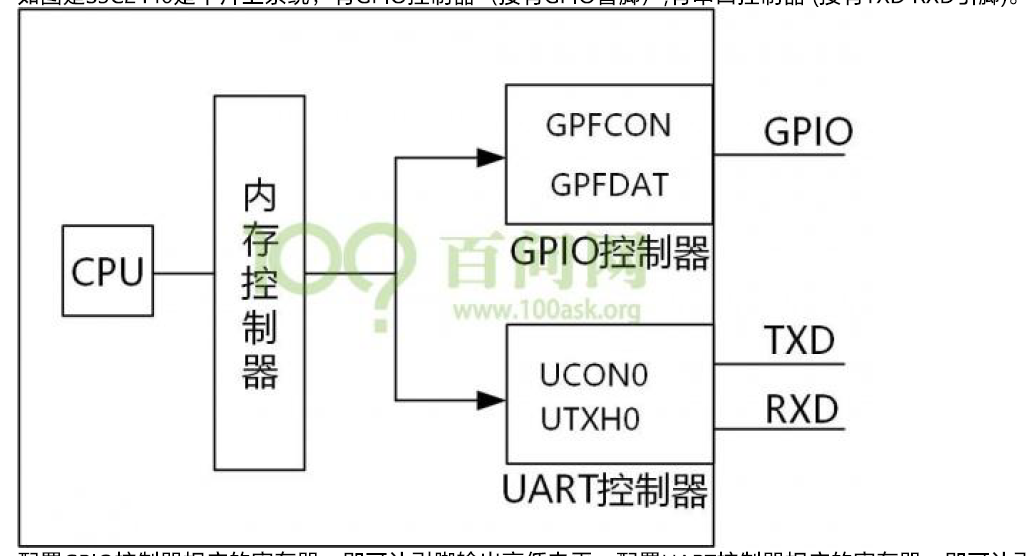
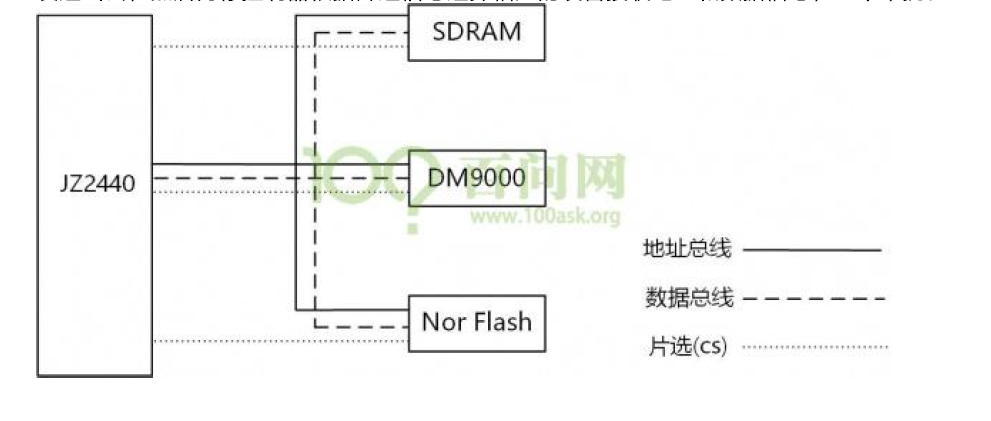
### 内存接口概念



Cpu发出地址，内存控制器根据地址选择模块，接着从模块得到数据或者发送数据。

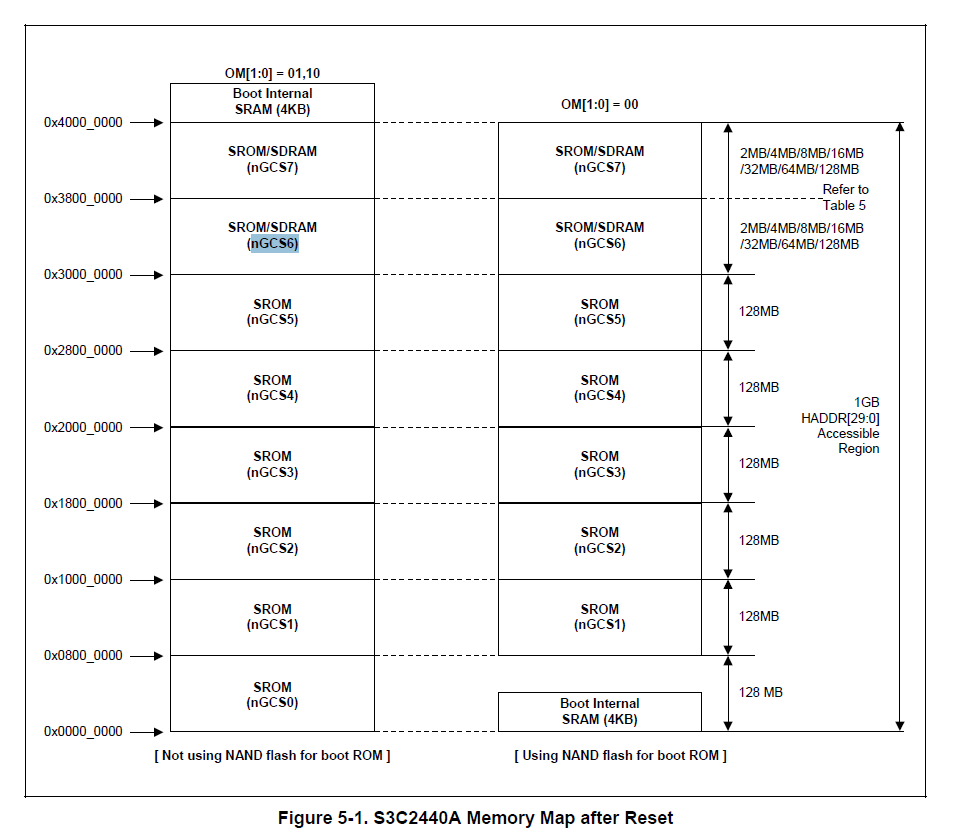
这种门电路接口（gpio），协议类接口不会把地址输出到外部，

下面的内存类接口会把地址输出到外部，比如nor flash、网卡、sdram



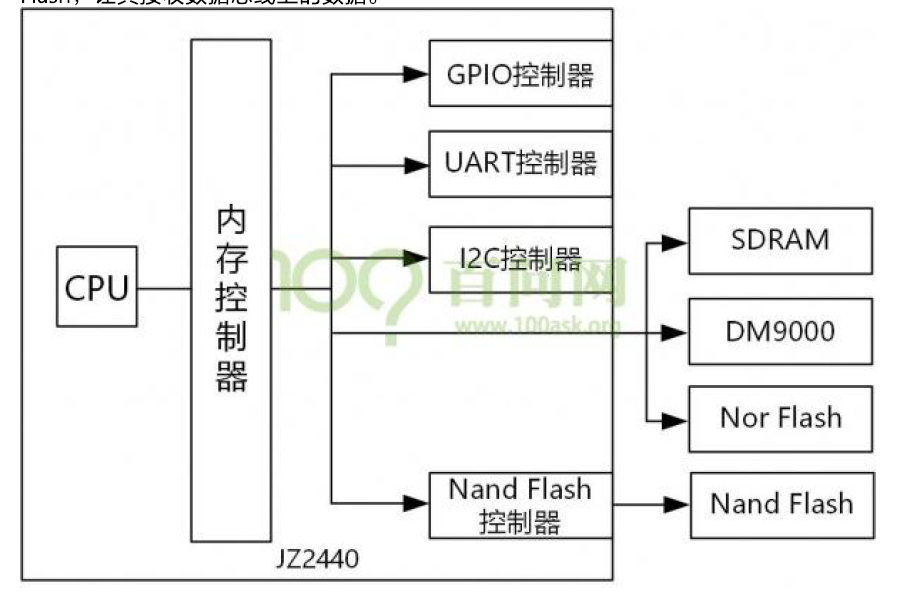
这种直接接在数据总线和地址总线上，CPU把数据和地址发送出去，然后内存控制器根据片选信号选择相应的设备接收地址和数据信号。

片选信号和地址的关系

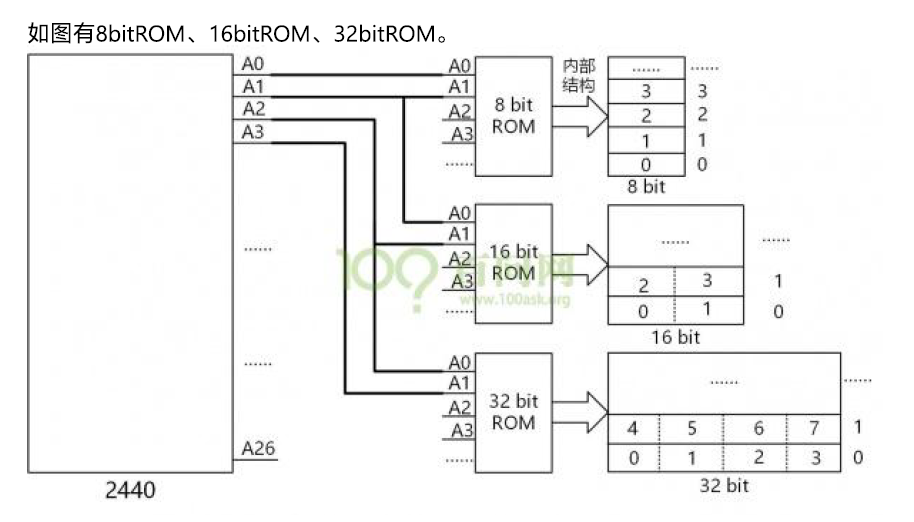
当选择nor启动时，CPU发出的指令地址范围处于0~0x0600 0000，内存控制器就会使ngcs0处于低电平，即片选引脚被选中，nor flash被选中。

当CPU发出指令地址范围处于0x20000000~0x26000000，存控制器就会使ngcs4处于低电平，即片选引脚被选中，网卡被选中。

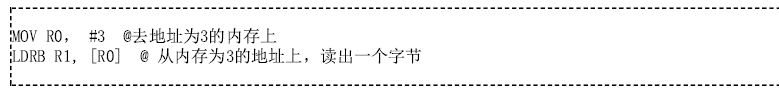
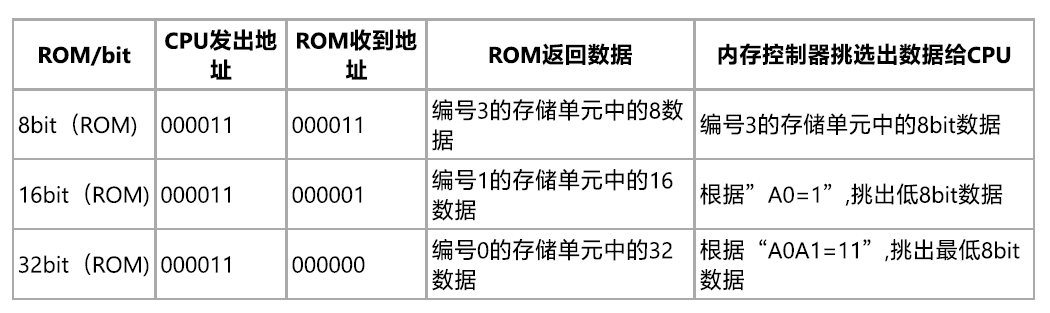
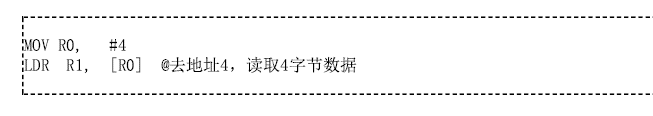
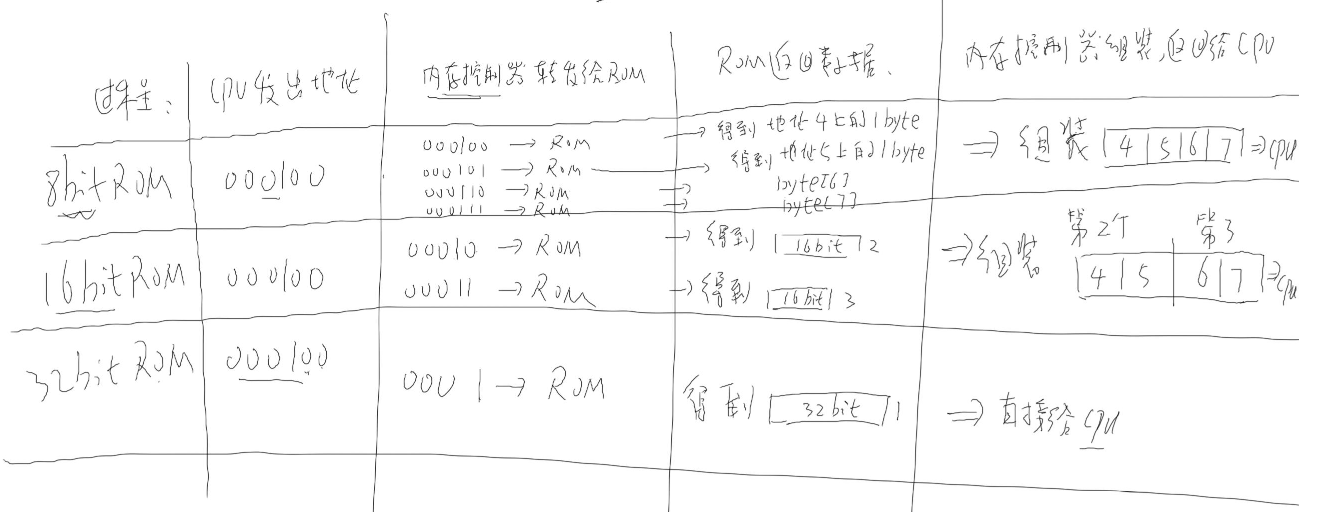
当CPU发出指令地址范围处于0x30000000~0x36000000，存控制器就会使ngcs6处于低电平，即片选引脚被选中，sdram被选中。



### 不同位宽设备的连接



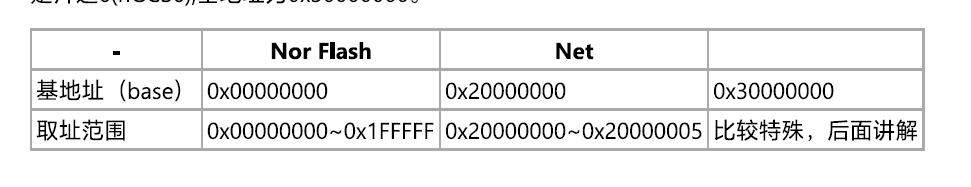
举个例子

怎么确定芯片的访问地址：1、根据片选信号确定基地址2、根据芯片所接地址线确定范围

例如：nor flash 接到ngcs0，基地址为0，用到的地址线为A0~A20

网卡接到ngcs4，基地址为0x2000 0000，用到的地址线为A0，A1两根地址线

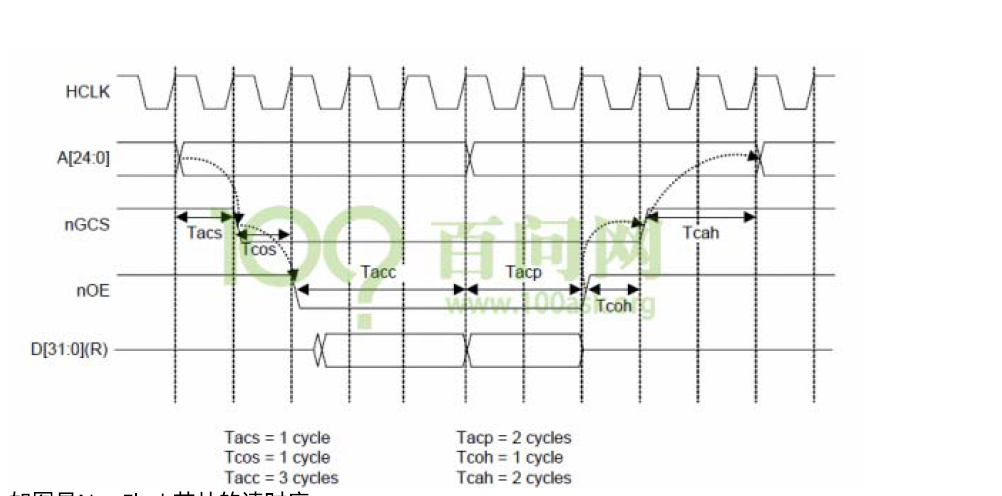


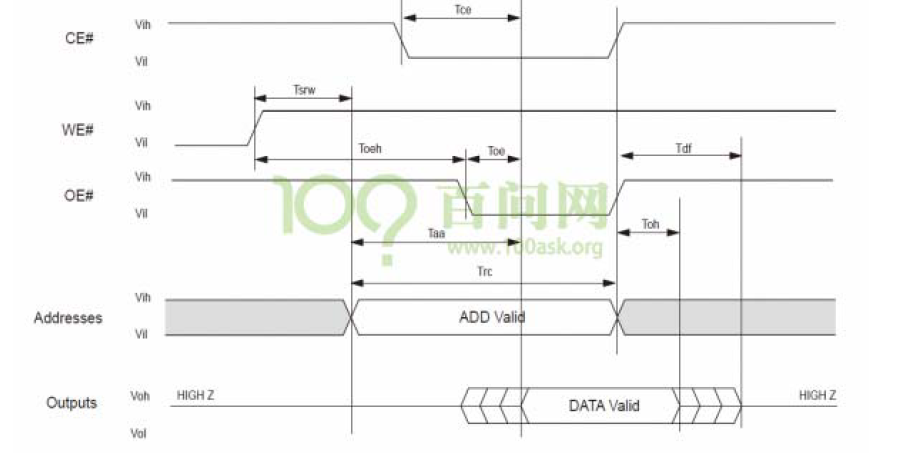
### 时序图分析

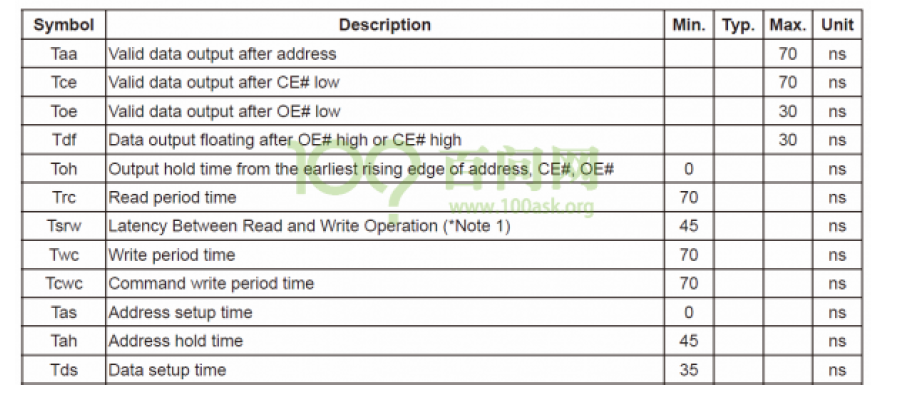
以nor flash为例分析如何设置时序



设置nor flash控制器时序去满足nor flash芯片的时序







可以看到

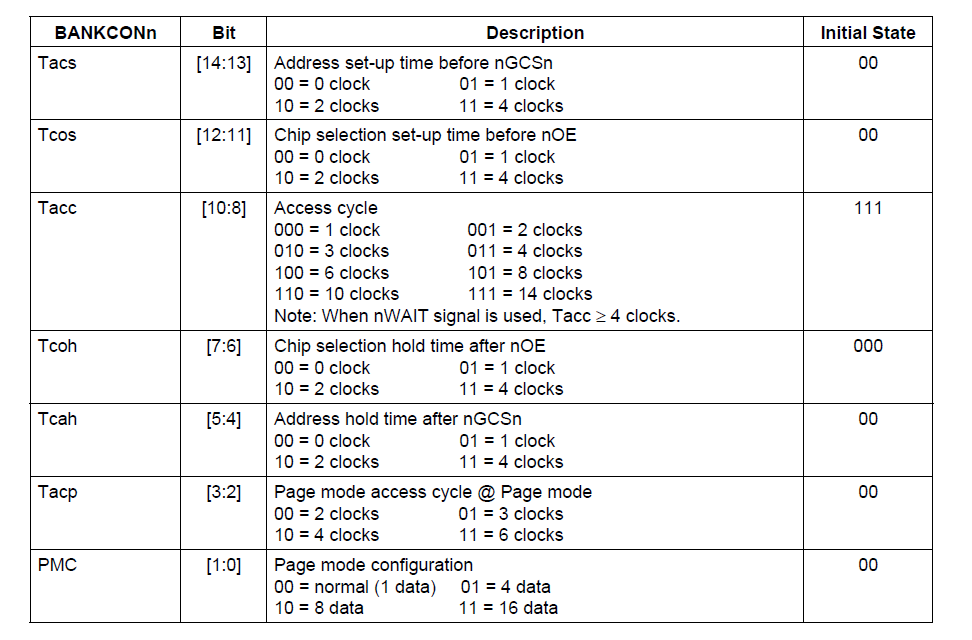
发出地址数据（addresses）后，要等待Taa（要求大于等于70ns）时间。地址数据才有效

发出片选信号（CE#）后，要等待Tce（要求大于等于70ns）时间。片选信号才有效

发出读信号（OE#）后，要等待Toe（要求大于等于30ns）时间，读信号才有效

为了简单直接把addresses ，CE# ，OE#同时发出，再同时等待70ns

查看手册，发现Nor flash是接在BANKCON0上的



这里设置tacc为70ns

HCLK = 100MHZ T = 1000/100 = 10ns 所以tacc取8 clocks

创建init.c

#include “s3c2440\_soc.h”

void bank0\_tacc\_set (int val)

{

BANKCON0 = val << 8;

}

init.h

#ifndef

#define

void ban0\_tacc\_set(int val);

#endif

最后在主函数里面，通过串口获取输入的值，传入bank0\_tacc\_set()函数里，设置tacc，然后再读取nor flash上的led程序

#include “s3c2440\_soc.h”

#include “uart.h”

#include “init.h”

int main(void)

{

unsigned char c;

uart0\_init();

puts(“ENTER TACC VAL:\n\r”);

while(1)

{

c = getchar();

putchar(c);

if(c >= ‘0’ && c <= ‘7’)

{

bank0\_tacc\_set(c - ‘0’);

led\_test();

}

else

{

puts(“ERR,VAL BRTWEEN 0 – 7\n\r”);

puts(“ENTER TACC VAL:\n\r”);

}

}

return 0;

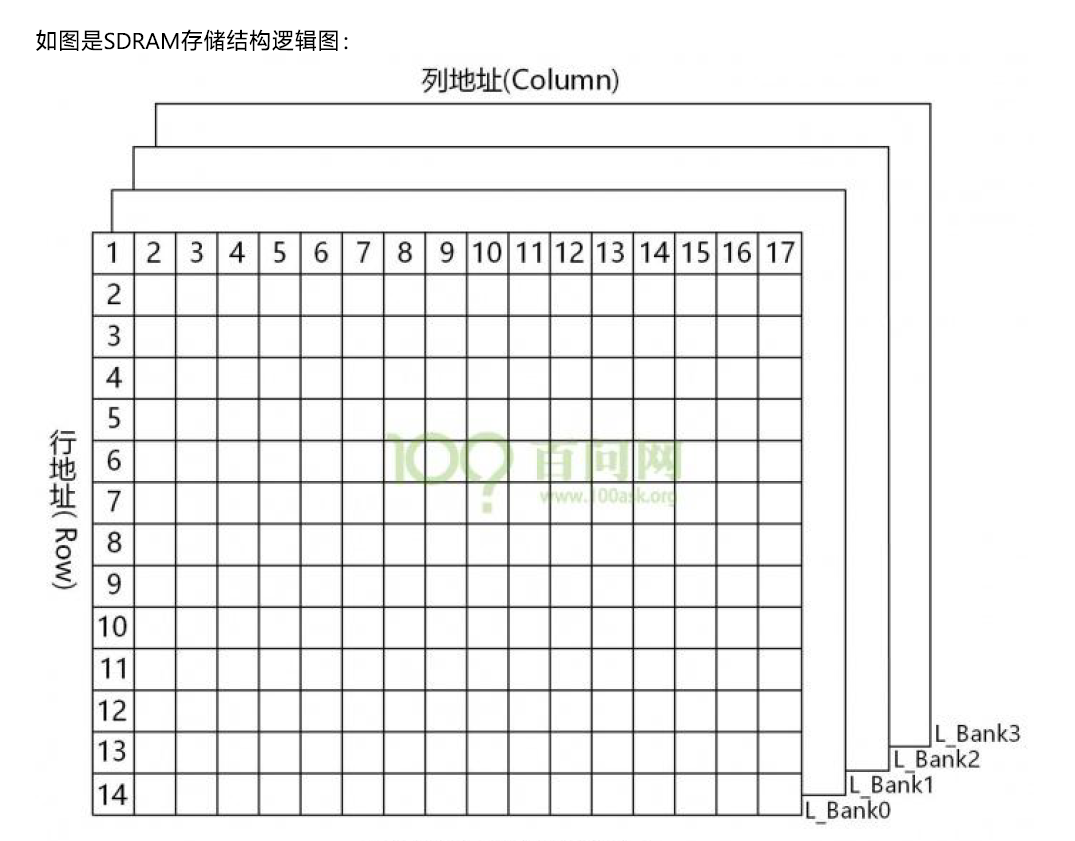
}

输入0 – 4，tacc小于70ns，无法读取nor flash上数据，led无法闪烁

输入4 – 7，tacc大于70ns，值越小闪烁越快 不明显

### SDRAM的设置

结合前面所说的，想要对SDRAM进行设置就得对内存控制器做设置

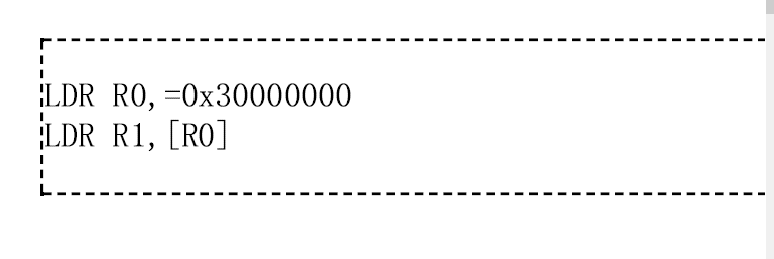


SDRAM总共有4个bank，每个格子表示的是16bit数据

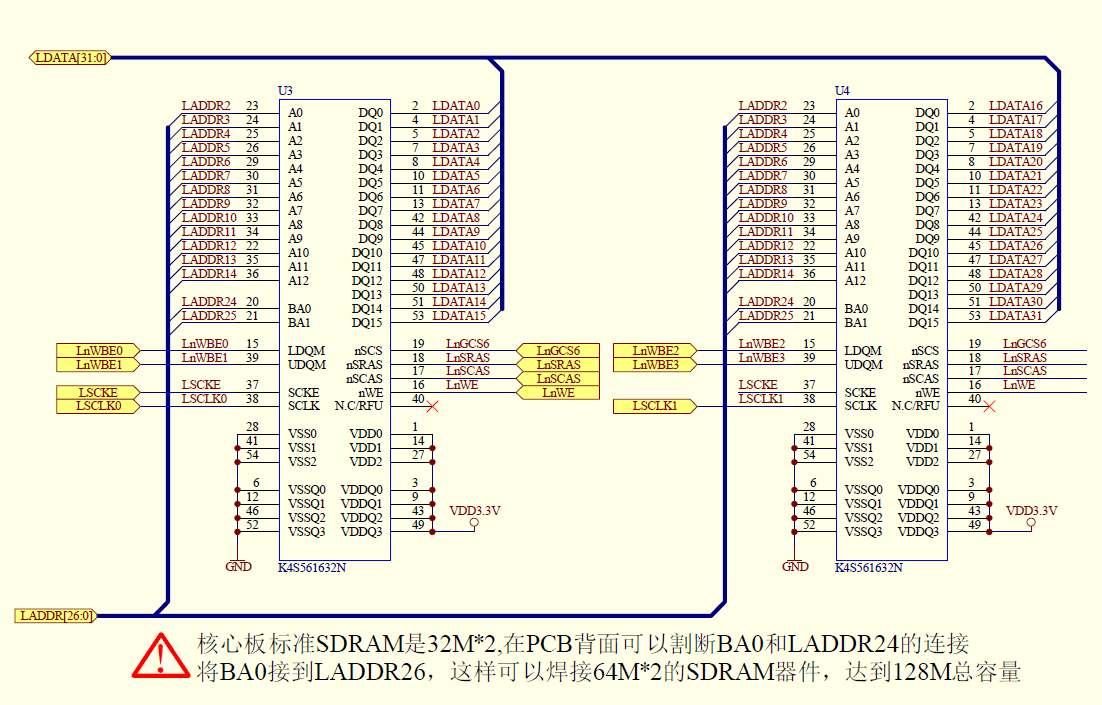
怎样访问某个格子？

1. 首先发出片选信号，选中整个芯片
2. 发出bank地址，选中一个bank
3. 发出行地址
4. 发出列地址，最后选中块

以上这么多信号都是由内存控制器发出，因此需要去设置它，cpu只是简单的执行读写内存的命令，



1. cup发出的片选信号nSCS6有效，选中SDRAM芯片
2. SDRAM有4个L-bank，需要两根地址信号选中其中一个L-bank，根据原理图，addr24.25
3. 对被选中的芯片进行统一的行列（存储单元）寻址



行列地址线公用addr2 – addr14 ，使用nSRAS,nSCAS来区分

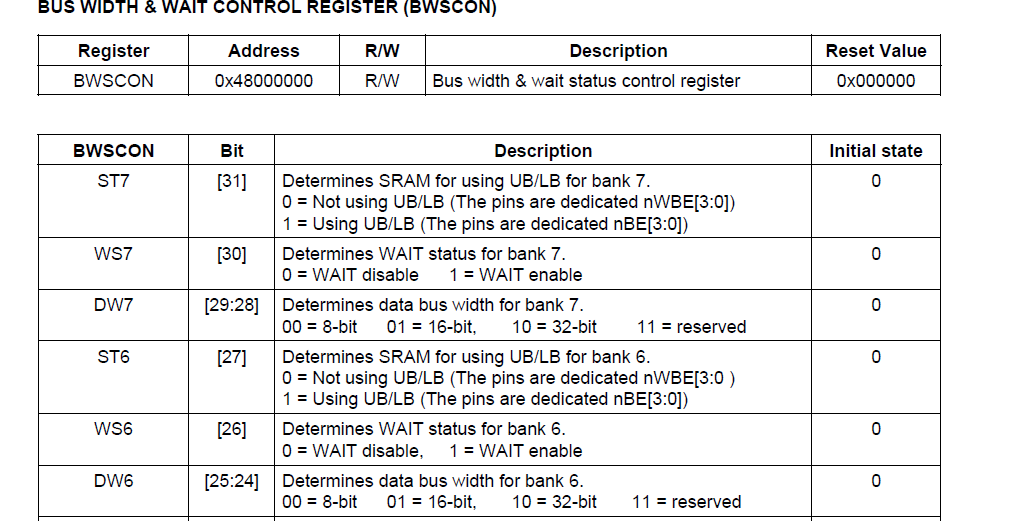
Addr24 addr25选择L-bank

内存控制器一共有13个寄存器

上面nor flash是接在bankcon0上的

一般bank0 – bank5只需要设置BWSCON 和BANKCONx就好了

SDRAM接bank6、bank7 时除了上面的两个寄存器还需要设置REFRESHA、BANKSIZW、MRSRB6、MRSRB7



最高四位没有使用，不用设置（ ？）

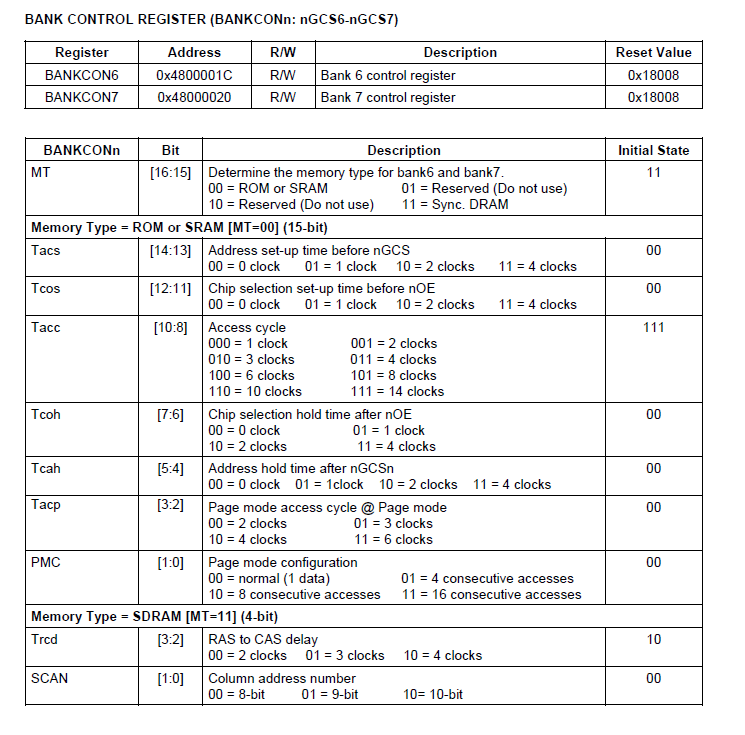
St6 [27]: 禁止/启动sdram的数据掩码引脚，sdram 为0、sram为1

Ws6 [26]：是否使用存储器的wait信号 设0

dw6[25:24]：用来设置bank的位宽，选10 = 32-bit

因此BWSCON = 0x22000000

（ dw7[29:28] = 10 ）



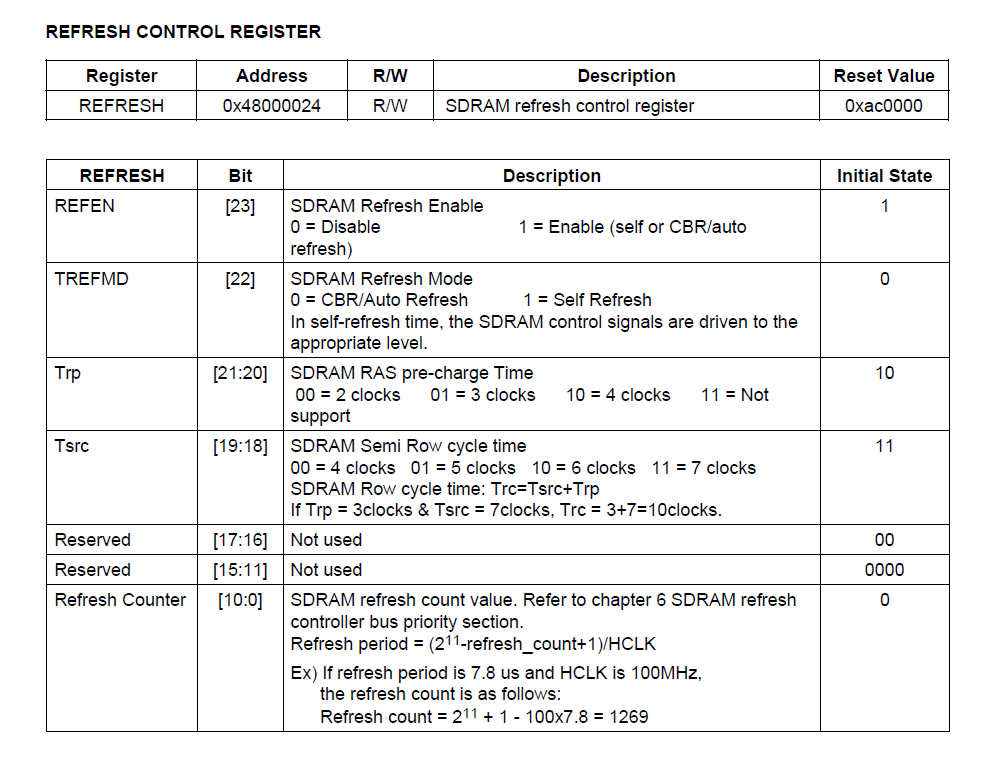
MT [ 16:15 ] 选择 11

Trcd [3:2] 行地址和列地址间隔多长时间，查找芯片手册20ns，选择00

Scan [1:0] 列地址位数，这里是01 = 9bit

所以BANKCON6 = 0x00018001

BANKCON7 = 0x00018001



REFEN [23] 1 开启刷新功能

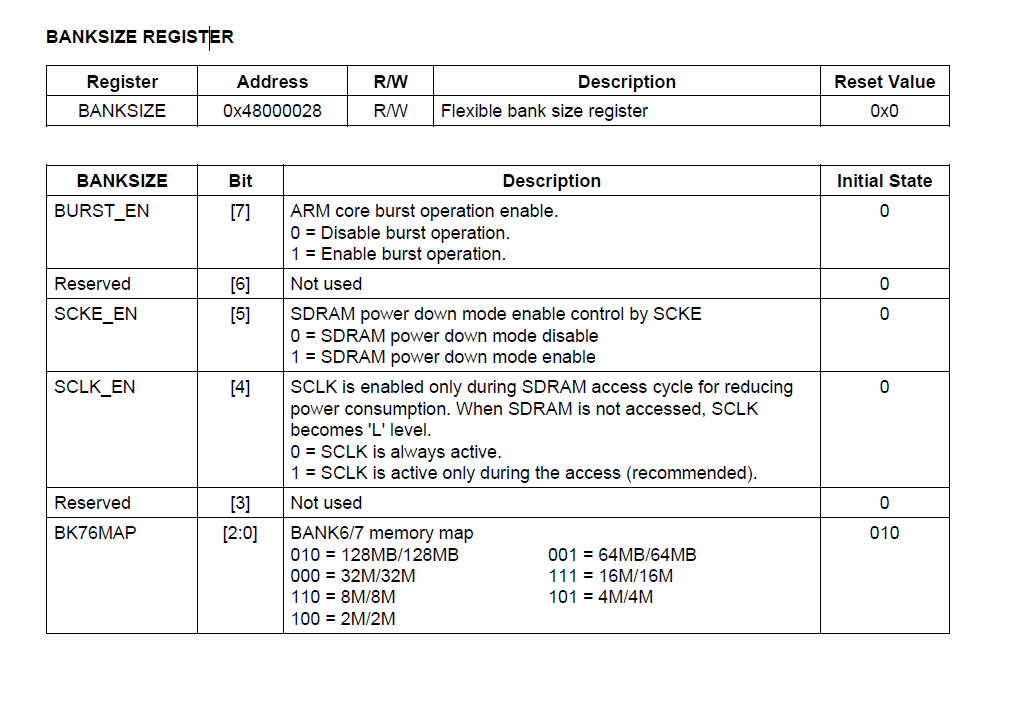
TREFMD [22] 刷新模式 1 一般用于休眠模式 这里使用默认值

TRP[21:20] 看sdram手册 选00

TSRC[19:18] 选 01

REFRESH COUNTER[10:0] 这里1269 = 0x4f5

所以REFRESH = 0x8404F5



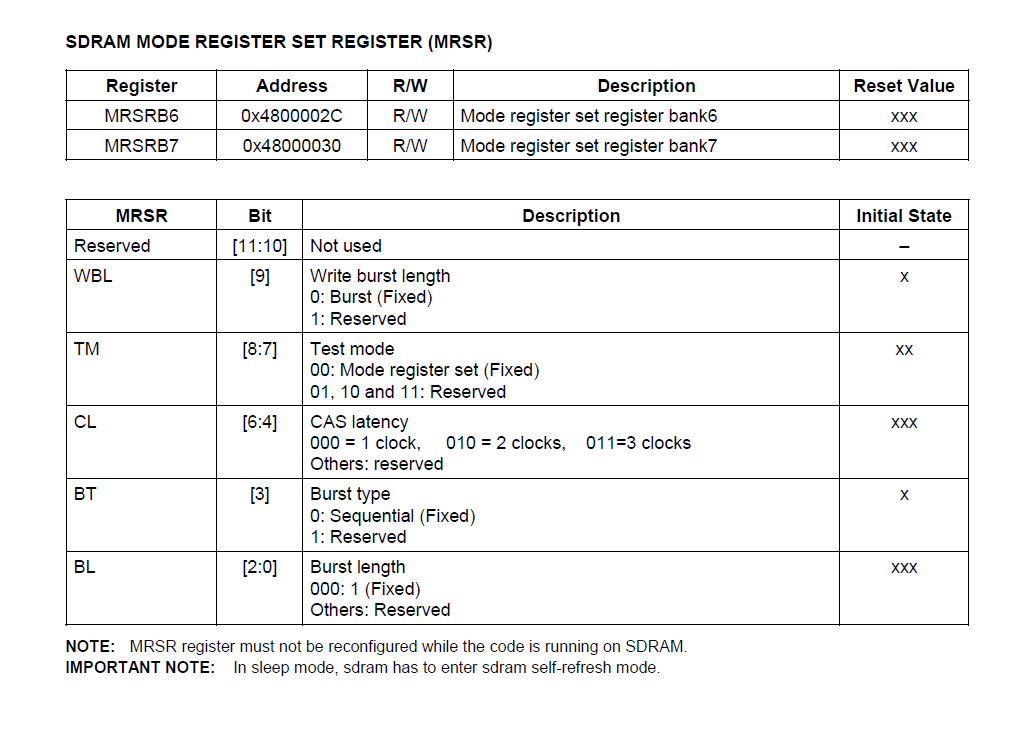
BURST\_EN [7] :禁止/允许arm核突发传输 选1

SCKE\_EN [5] :禁止/循序SCKE信号进入省电模式 选1

SCLK\_EN[4] : 0= 时刻发出sclk信号 1= 仅在访问sdram器件发出sclk信号 选1

BK76MAP[2:0] ：设置bank6大小，外接64M的sdram，选 001

因此BANKSIZE= 0xb1



这里只设置CL[6:4]选010

所以MRSRB6 = 0x20 MRSRB7 = 0x20

写程序

在上面init.c里面添加

Void sdram\_init(void)

{

BWSCON = 0x22000000;

BANKCON6 = 0x00018001;

BANKCON7 = 0x00018001;

REFRESH = 0x8404F5;

BANKSIZE= 0xb1;

MRSRB6 = 0x20 ;

MRSRB7 = 0x20;

}

再写一个测试函数

Int test\_sdram(void)

{

volatile unsigned char \*p = ( volatile unsigned char \*)0x30000000;

int i;

for(i = 0 ;i < 1000;i++ )

p[i] = 0x55;

for(i=0;i<1000;i++)

if(p[i]!=0x55)

return -1;

return 0;

}

主函数里测试

int main(void)

{

uart0\_init();

sdram\_init();

if(sdram\_test() == 0)

led\_test();

return 0;

}