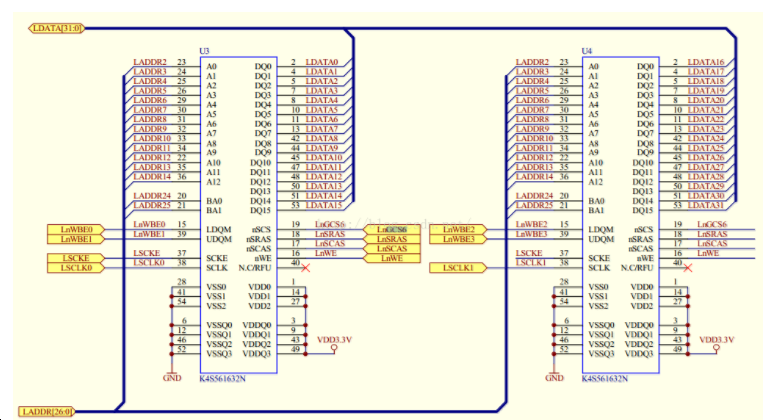
**SDRAM和处理器寻址的几点理解**



每个芯片的是16跟数据线，x2就组成32位数据线。地址地址13根，BA0 BA1 接到ADDR24 ADDR25   【这个是片选择哪一个bank】，为什么？

<http://blog.csdn.net/zmnqazqaz/article/details/51298289>

<http://blog.csdn.net/kickxxx/article/details/7231621>

<http://blog.csdn.net/blueice8601/article/details/7377092>

1. 先弄清楚第一个概念，对于CPU来说:

 一个地址用32位来表示：0x30000000、0x40000000; 但是这个地址代表的地址空间是1个字节，也就是说CPU读取0x30000000这个地址得到的是一个字节的数据； 一个地址是多少位的地址跟这个地址代表的地址空间有多大是2码事。

CPU：  
          0x30000000         1字节  
          0x30000001         1字节  
          0x30000002         1字节

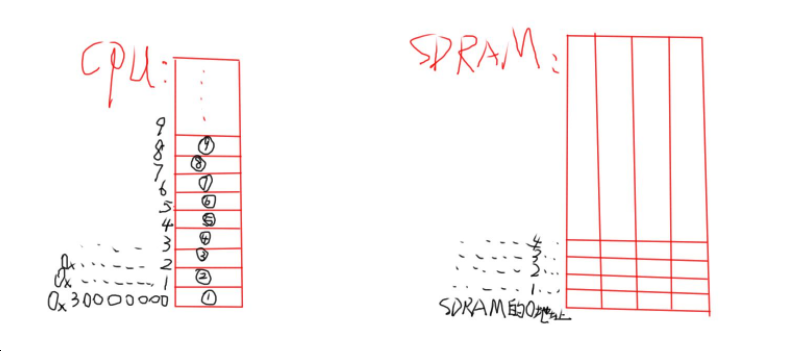
2，

  ① 对于JZ2440来说，SDRAM有2片，每一片的数据输出宽度是16bit，总共构成了具有**32bit数据**输出的内存，现在这块内存1次输出32bit的数据。

  ② 在这种情况下，读取SDRAM的一个地址得到的数据是4个字节

现在就出现了一个矛盾：  对于CPU来说一个地址代表1个字节的数据，但是我读一次SDRAM，它会给我4字节的数据，怎么办？

**假设：**下图黑色带序号的表示CPU想得到的数据，①②③④⑤...数据大小都是1个字节



0x30000000是SDRAM的首地址，现在CPU想得到①开始到⑨的9个字节数据。①②③...⑨这9个字节在SDRAM实际是怎么存储的呢？



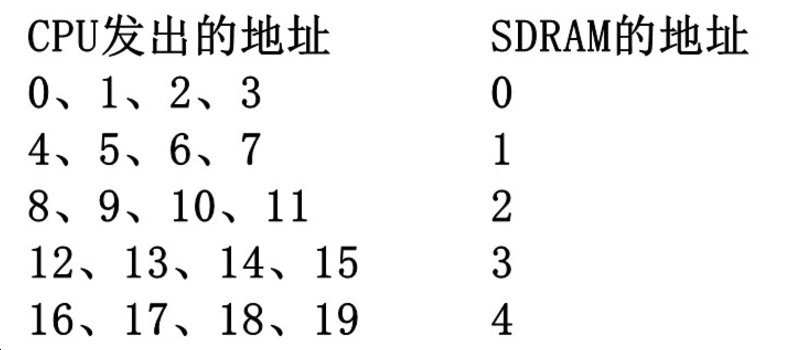
 也就是说①②③④这四个字节的数据在SDRAM的0地址，⑤⑥⑦⑧这四个字节的数据在SDRAM的1地址，⑨这个字节在SDRAM的2地址。

那么，

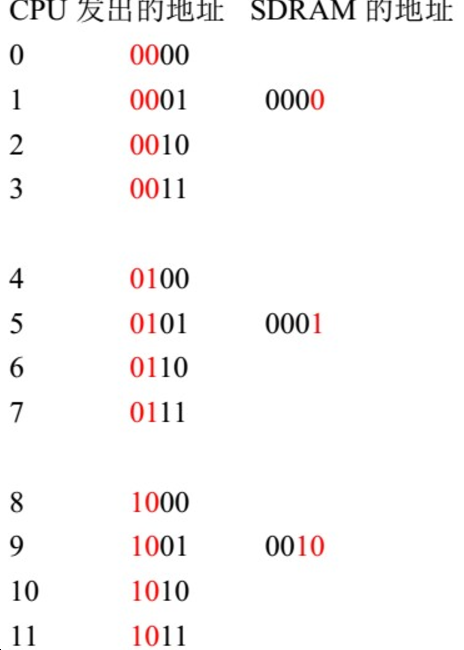
 ⑴   **CPU发出0x30000000、0x30000001、0x30000002、0x30000003这4个地址的时候，访问的都必须是SDRAM的0地址**

 ⑵   **发出0x30000004、0x30000005、0x30000006、0x30000007这4个地址的时候，访问的都必须是SDRAM的1地址**

⑶   **发出0x30000008这个地址的时候，访问的必须是SDRAM的2地址。**       而CPU的地址线是直接连在SDRAM的地址线上的，中间又没有其他硬件或者软件做转换，要做到上面这3点，怎么办？       我们把这个地址对应关系写的更简单一些：



**0x30000000、0x30000001....这些地址是32位的，每一位对应一根地址线（这句话很重要）。** **如果我们把上面的地址对应表以二进制的形式表示出来会更加直观：**



最左边的数字是地址的10进制表示，中间是这个地址代表的二进制，最右边是对应的SDRAM的地址。

CPU要想发出地址后得到正确数据，**那么发出的地址的bit0、bit1是无效的，地址从bit2开始有效。**

实际有效地址的最低位是从bit2开始的：

  无论bit0、bit1的值是多少，bit2为0的时候访问的是SDRAM的0地址；

                 bit2为1的时候访问的是SDRAM的1地址；

                 bit3、bit2为10的时候访问的是SDRAM的2地址；

                ......

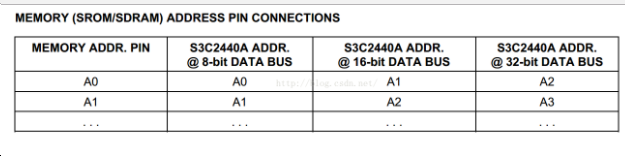
 既然实际有效地址的最低位从bit2开始，那么把CPU的ADDR2地址线接到SDRAM的AD0;

          把CPU的ADDR3地址线接到SDRAM的AD1;

 把CPU的ADDR4地址线接到SDRAM的AD2;

        .......

 CPU的ADDR0和ADDR1不用接。

**其实这个也只跟位宽有关系。**

**总结一下：CPU一定是字节寻址（8bit）**

**内存芯片的寻址是看数据位宽**

**我们把统一单位的地址线连接起来就可以了。**

**比如A0是2的0次方字节    1字节**

**A1是2的1次方字节    2字节**

**A2是2的2次方字节    4字节**

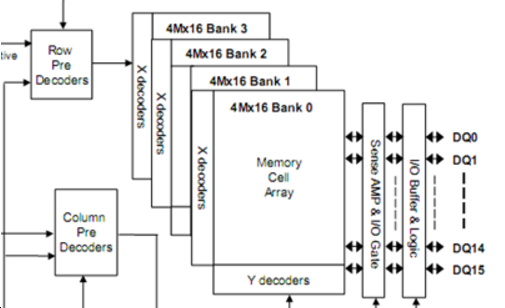
**那么位宽8位  就是1字节  从A0开始连接**

**16位  就是2字节  从A1开始连接**

**以此类推**

**我们看一下SDRAM的原理 就是一个立方体， 行x列x几（bank）**

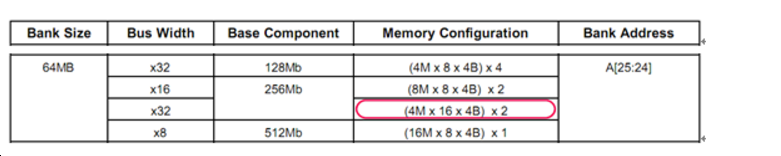
**寻址就非常简单，首先CPU发送第几个bank信号，对应ba0 ba1的值，然后通过地址依次发 行   列  信号，怎么知道是行还是列就看**nSRAS  nSCAS 行/列 有效管教是不是被置位了



一片DRAM其实是由4个4Mx16的存储块也就是bank组成，其中4M指的是4Mbit个存储单元，16指的是每个存储单元存储的位数，即2个字节。需要知道的是这4片存储块是相互独立的，并且每个bank都是阵列结构，所谓阵列结构指的是它是由行列组成的。一个bank看做是由许多方格组成的存储块，而方格的个数就是4Mbit,每个方格的容量则为16bit，你要想确定一个方格的地址首先要确定它的行地址，再确定它的列地址，两线相交确定一点，这一点就是你要的地址。一般SDRAM的地址线都采用分时复用原则，即先送行地址，再送列地址，K4S561632C这款芯片Row address : RA0 ~ RA12, Column address : CA 0 ~ CA8,即前9个地址线是被行列复用的。所以寻址的时候先发送bank地址和RA0 ~ RA12行地址，然后再发送CA 0 ~ CA8列地址。按照行地址13个，列地址9个，总容量为4 \* 2^13 \* 2 ^ 9 \* (16 / 8) = 32MB

3、为什么bank0和bank1与2440的ADDR24和ADDR25连接?

我们已经了解到bank0和bank1是用来选择存储块 (bank)的，32M的空间需要25根地址线寻址,但是对于K4S561632C只需要24根,因为它每个存储单元存储2个字节,我们来分一下,这24根地址线的前13根行地址,紧接着的9根为列地址,那么最后两根是干嘛？显而易见它是bank地址,用来选择存储块,所以BANK0接23, BANK1接24,但是由于我们是从ADDR2开始连得原因前面已经解释过了,所以每个地址加2.最后的地址连接方法,就是ADDR2~ ADDR14对应行地址, ADDR15~ ADDR23对应列地址,(由于管脚的分时复用其实是DDR2~ ADDR14对应行地址, ADDR2~ ADDR10对应列地址)ADDR24~ ADDR25对应BANK0~ BANK1,这就是bank0和bank1与2440的ADDR24和ADDR25连接的原因。



Bank Size:外接内存容量大小（4Mbit\*16bit\*4Bank\*2Chips/8=64MB）

Bus Width:总线宽度 （两片16位HY57561620，并联成32位）  
Base Component：单个芯片容量(bit)（256Mb）  
Memory Configration：内存配置 (（4M\*16\*4banks）\*2Chips )

4、由两片K4S561632C组合在一起的,这样做的效果是什么？

我们已经知道1片K4S561632C有4个bank,两片K4S561632C就有8个bank了,当连接一个K4S561632C时,一个地址对应2个字节,现在连接了两片K4S561632C那么,一个地址就对应4个字节了,即一个字,总共64MB,当一个地址存放1个字节,我们需要64Mbit个地址才能达到64MB的容量,现在容量不变,每个地址里存放的字节数增多了,那么必然需要的地址数就减少了,根据存储容量（BYTE）=地址数\*每个地址能存的字节数,可知当每个地址里存放的字节数增大4倍时,我们的地址数会变为原来的1/4, 当一个地址存放4个字节,我们只需要16Mbit个地址就能达到64MB的容量,16Mbit代表24根地址线(2^24=16Mbit=16\*1024\*1024)