

第5章 存储器与I/O接口原理

5.1 存储器概述

5.2 存储系统机制

5.3 S3C2410存储系统

5.4 S3C2410 I/O端口

5.1 存储器概述

5.1.1 存储器基本概念

5.1.2 SRAM和DRAM

5.1.3 NOR FLASH和NAND FLASH

5.1.1 存储器概述

存储器的一般概念和分类

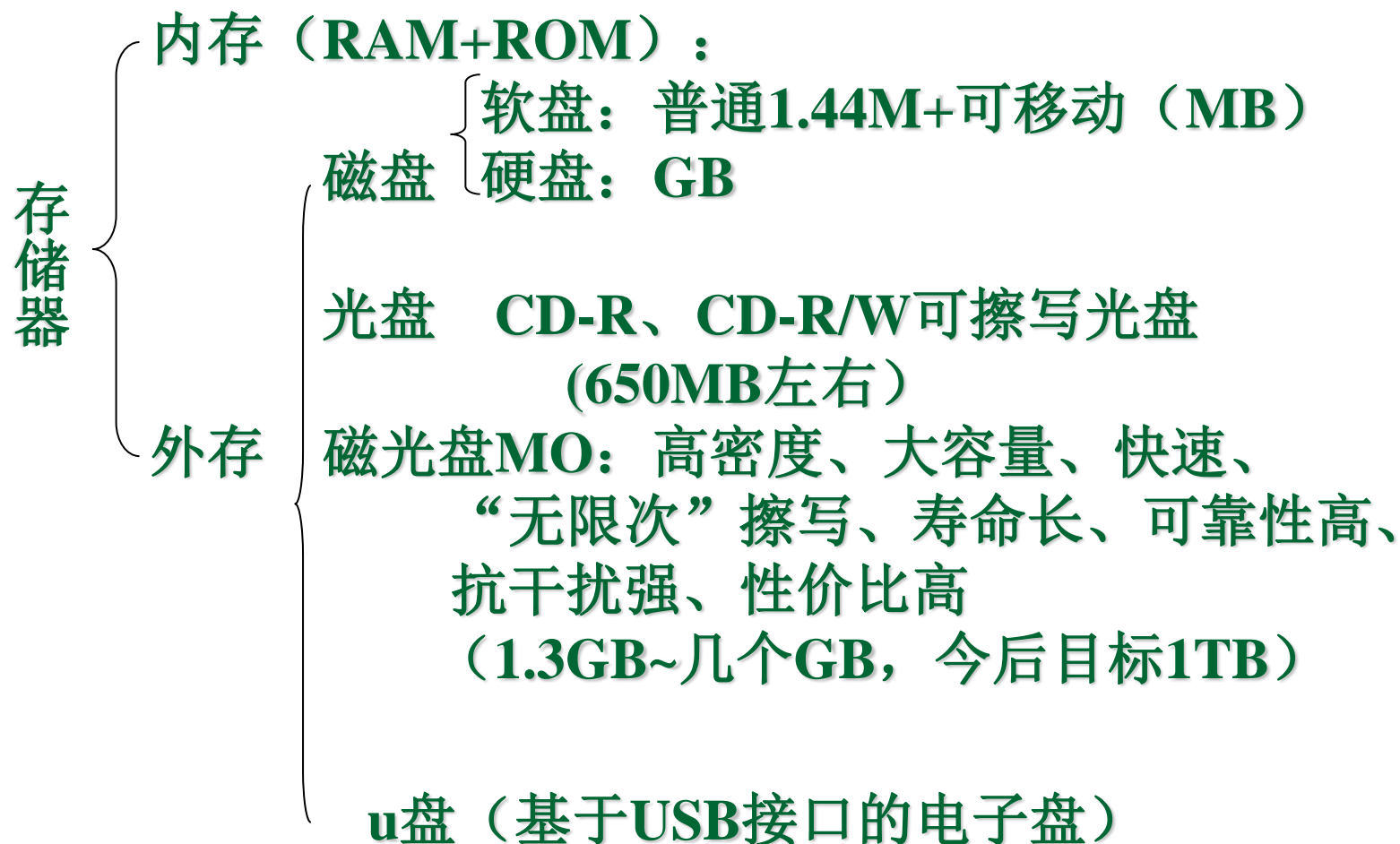
按存取速度和在计算机系统中的地位存储器分为两大类：

(1) **主存储器**：速度较快，容量较小，价格较高，用于存储当前计算机运行所需要的程序和数据，**可与CPU直接交换信息**，习惯上称为主存，又称内存(内部存储器)。

(2) **辅存储器**：速度较慢，容量较大，价格较低，用于存放计算机当前暂时不用的程序、数据或需要永久保持的信息，辅存又称外存(外部存储器)或海量存储器。

外存要配备专门的设备才能完成对外存的读写。通常，将外存归入到计算机外设一类。

存储器分类



■ 主存储器的分类

按存储
功能分

只读存储器（ROM）

随机存储器（RAM）

按制造
工艺分

双极性

MOS型

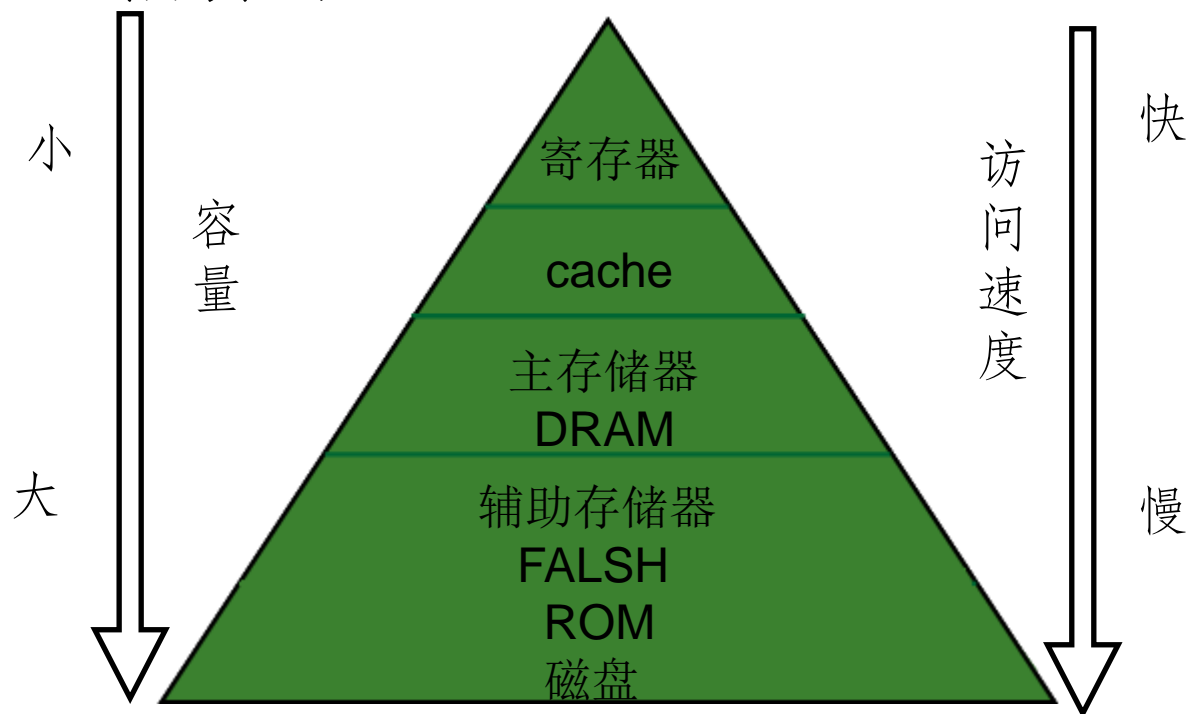
■ 主存储器的分类



存储系统的层次结构

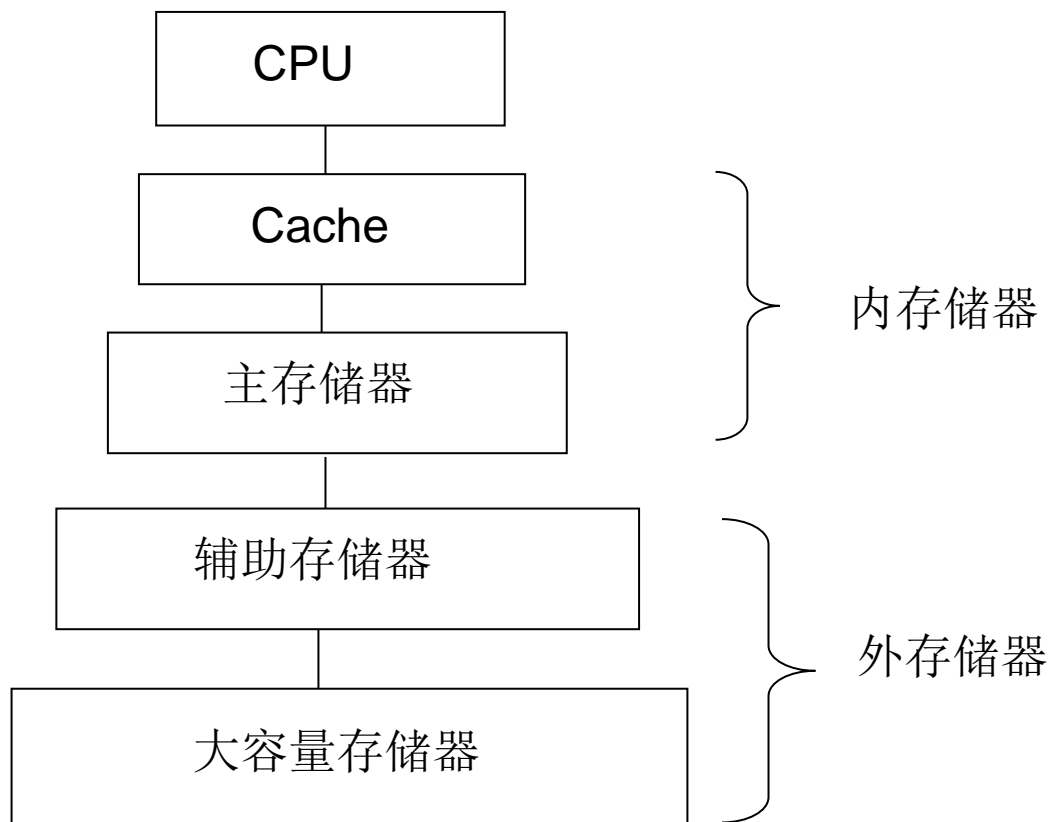
- 存储器是用来存储信息的部件，是嵌入式系统硬件中的重要组成部分。在复杂的嵌入式系统中，存储器系统的组织结构按作用可以划分为4级：

- 寄存器；
- cache ；
- 主存储器；
- 辅助存储器；



存储系统的层次结构

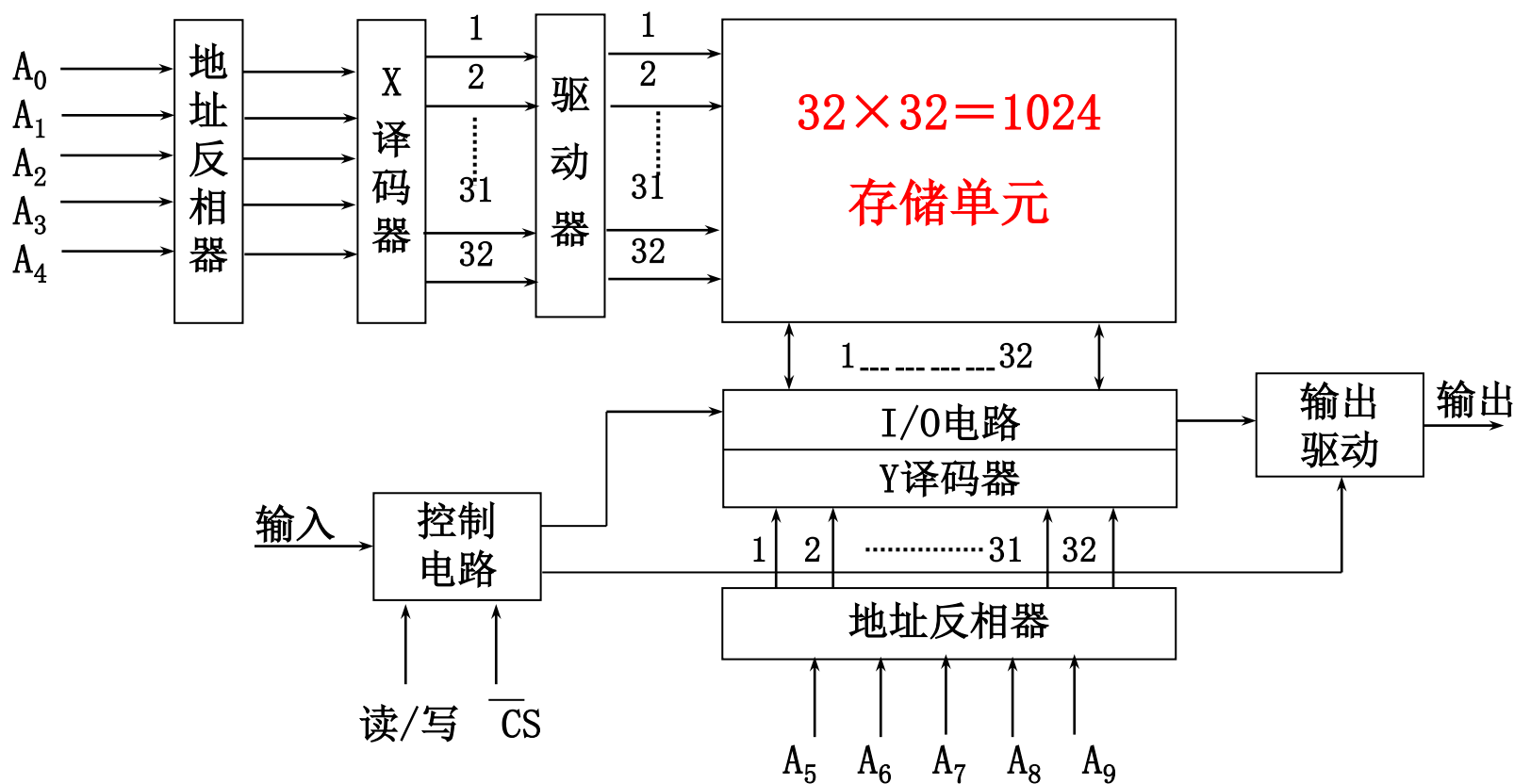
- 越靠近CPU的存储器速度越快而容量越小



5.1.2 SRAM和DRAM

- RAM：随机存取存储器
 - SRAM：静态随机存储器
 - DRAM：动态随机存储器
-
- SRAM读/写速度比DRAM读/写速度快；
 - SRAM比DRAM功耗大；
 - DRAM的集成度可以做得更大，则其存储器容量更大；
 - DRAM需要周期性的刷新，而SRAM不需要

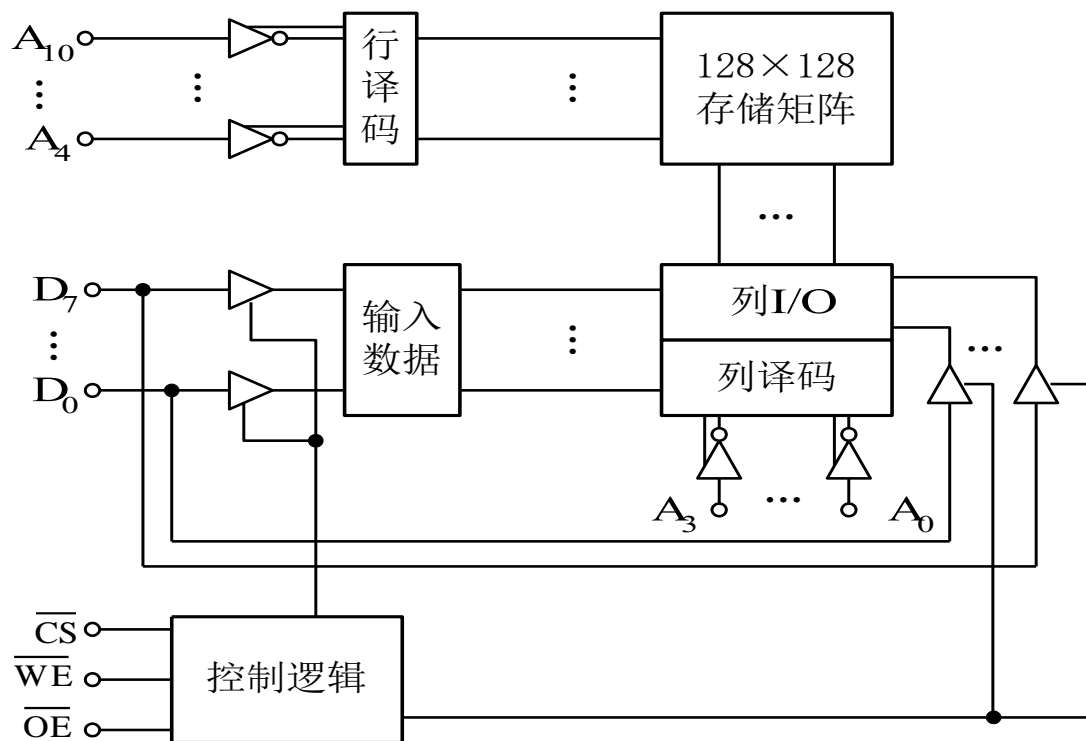
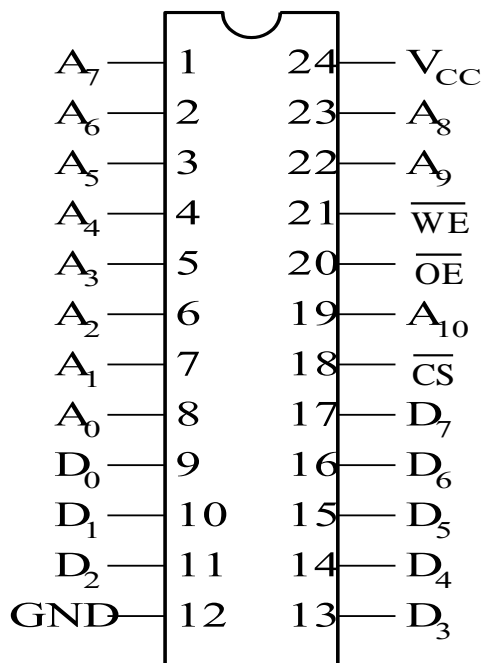
静态RAM的结构



由存储矩阵，地址译码器，控制逻辑和三态数据缓冲器组成。

典型SRAM芯片

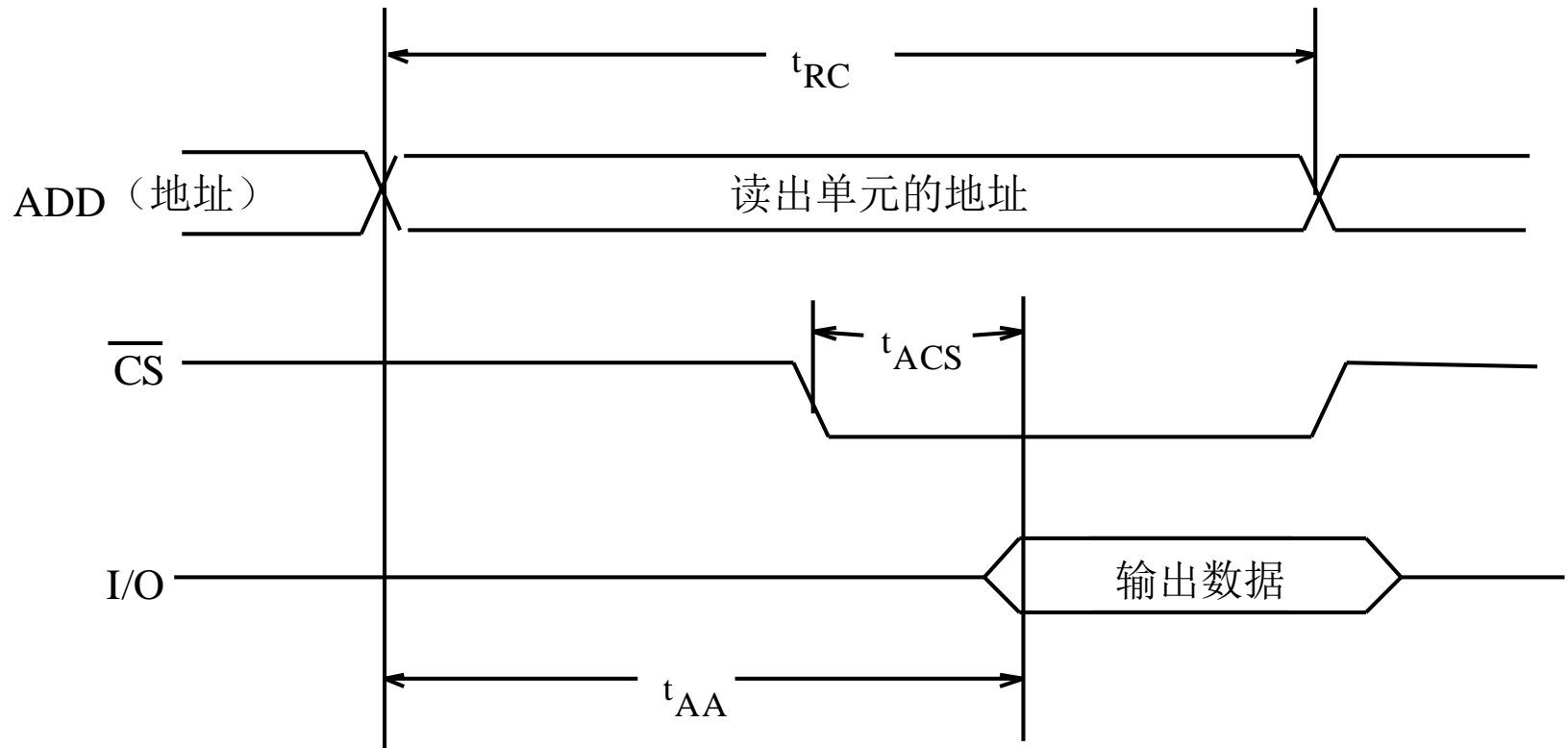
- 各SRAM芯片的引脚信号基本相同。
- 其存储容量不同,则地址线的根数不同;
- 其存储位数不同,则数据线的根数不同。



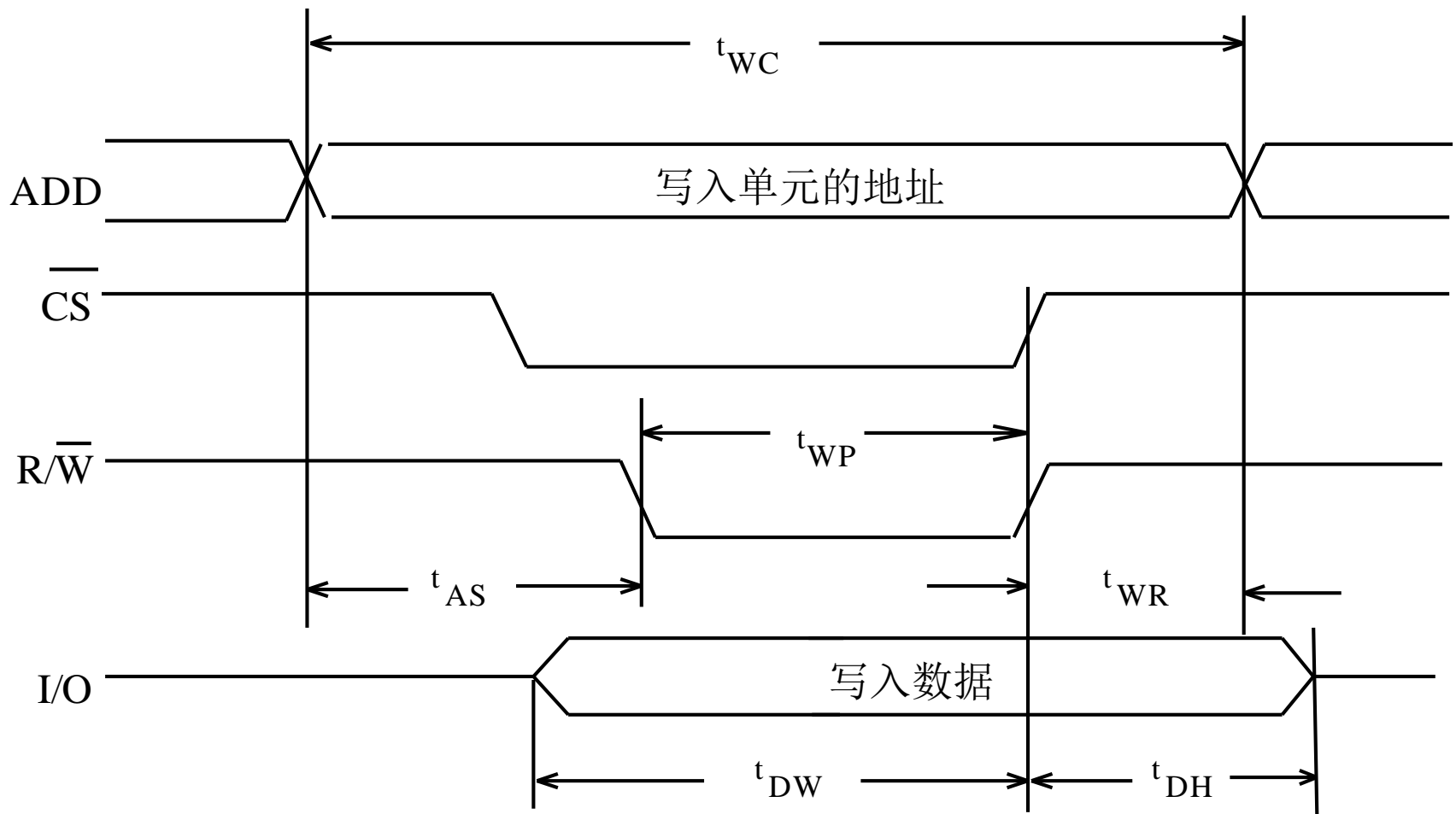
SRAM工作过程

- **读出:**
- 地址线 $A_{10} \sim A_0$ 送来的地址信号经译码后选中一个存储单元(其中有8个存储位), 由CS、OE、WE构成读出逻辑($CS=0$, $OE=0$, $WE=1$), 打开右面的8个三态门, 被选中单元的8位数据经I/O电路和三态门送到 $D_7 \sim D_0$ 输出。
- **写入:**
- 地址选中某一存储单元的方法和读出时相同, 不过这时 $CS=0$, $OE=1$, $WE=0$, 打开左边的三态门, 从 $D_7 \sim D_0$ 端输入的数据经三态门和输入数据控制电路送到I/O电路, 从而写到存储单元的8个存储位中。
- **当没有读写操作时, $CS=1$, 即片选处于无效状态, 输入输出三态门至高阻状态, 从而使存储器芯片与系统总线“脱离”**

SRAM的读时序

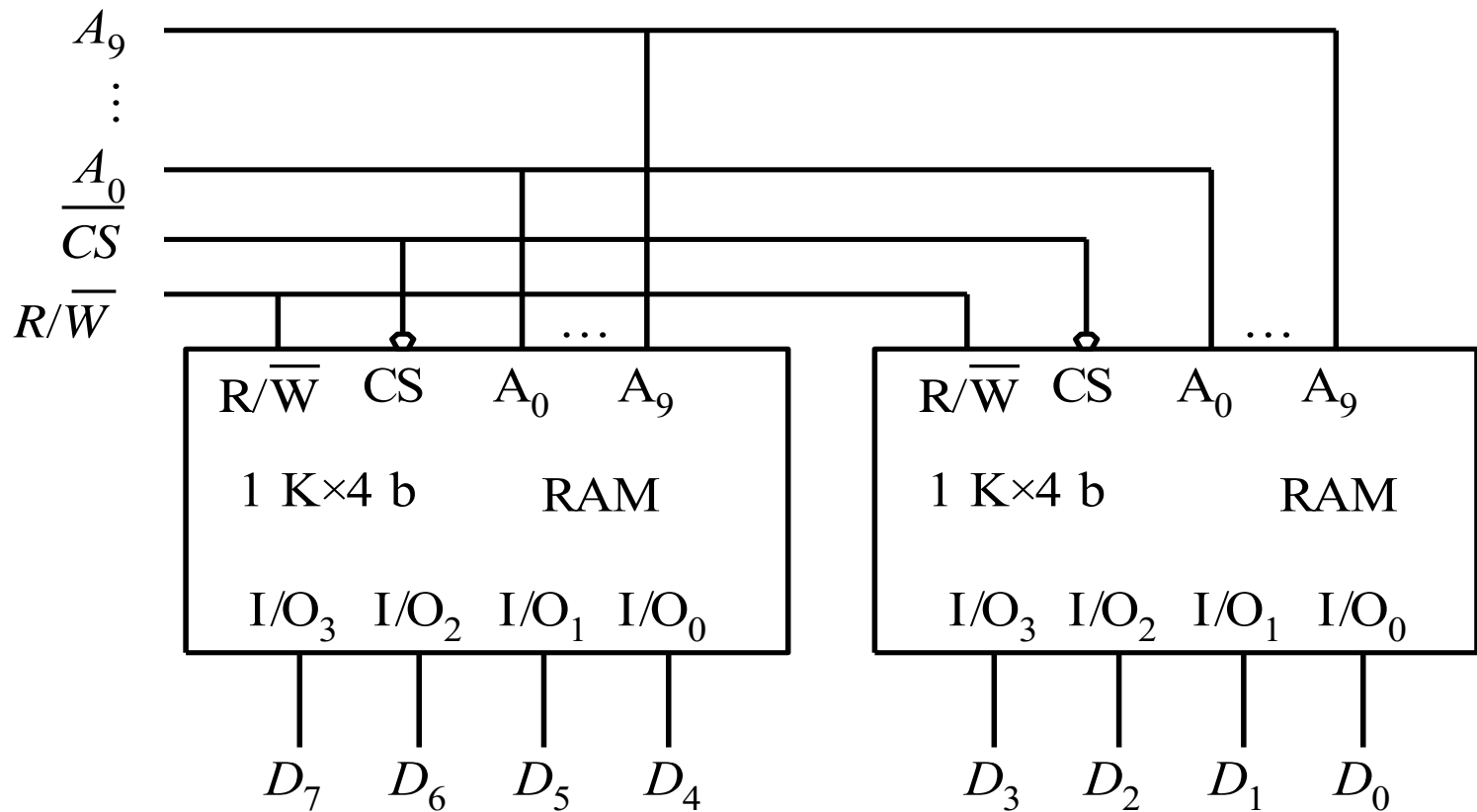


SRAM的写时序

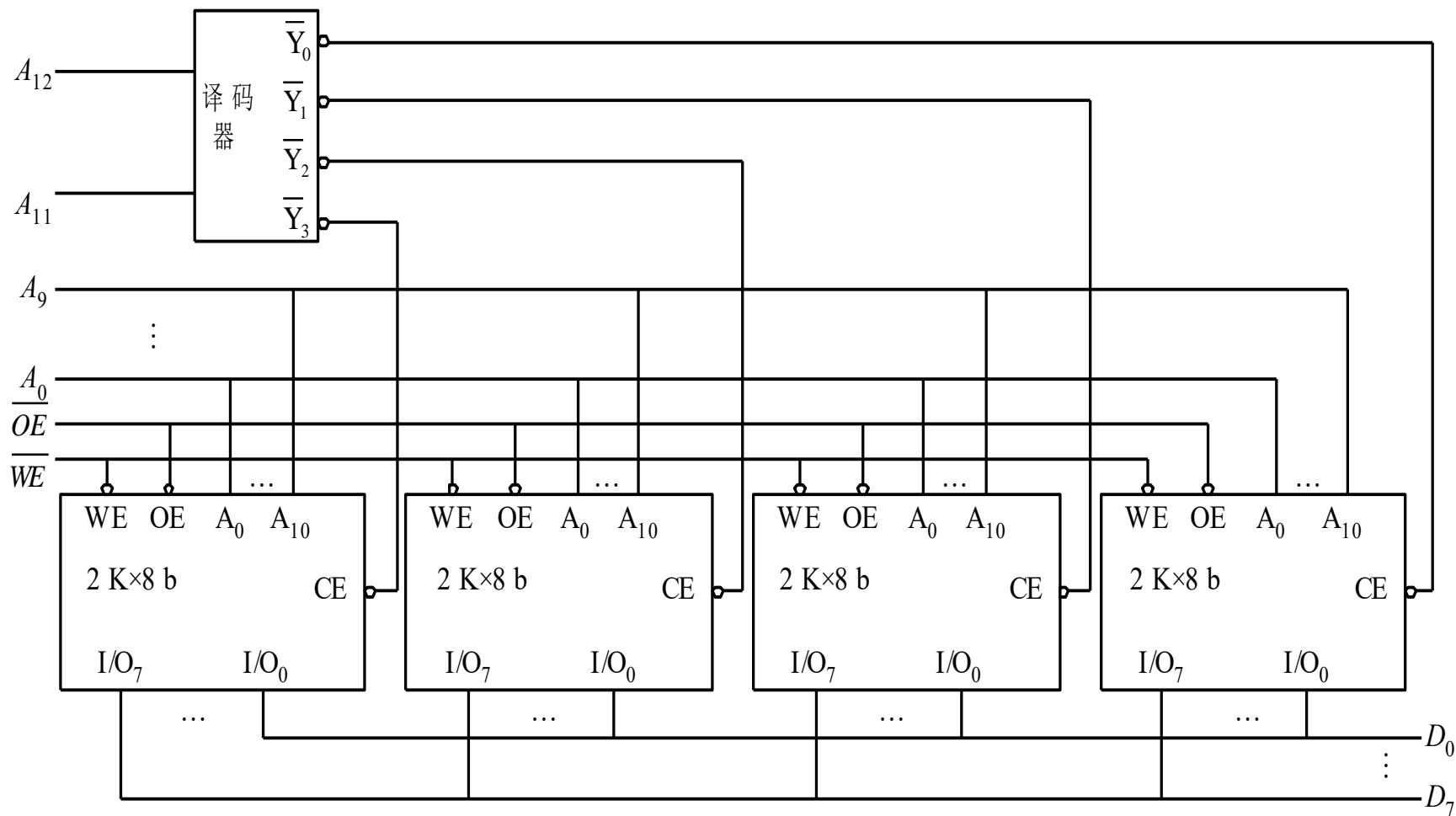


存储容量的扩展

位扩展:

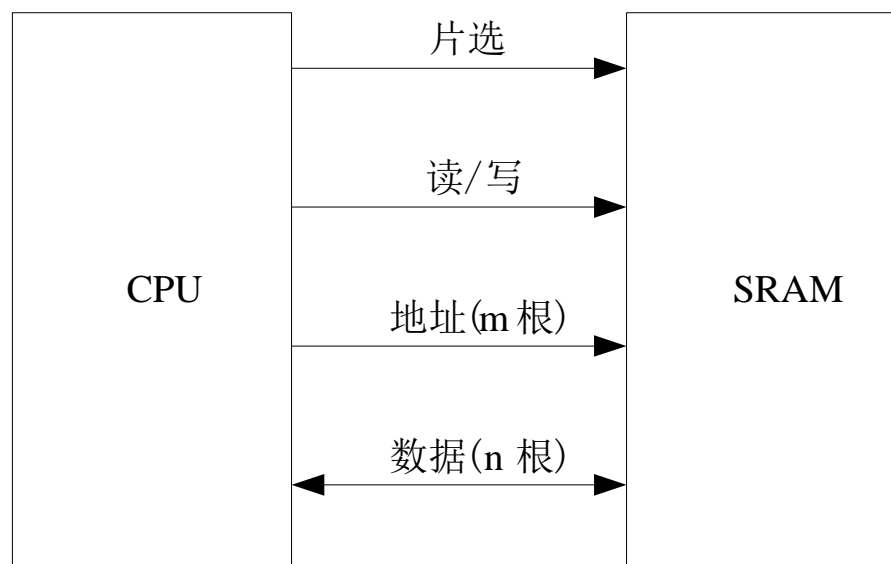


字扩展:



SRAM与CPU接口

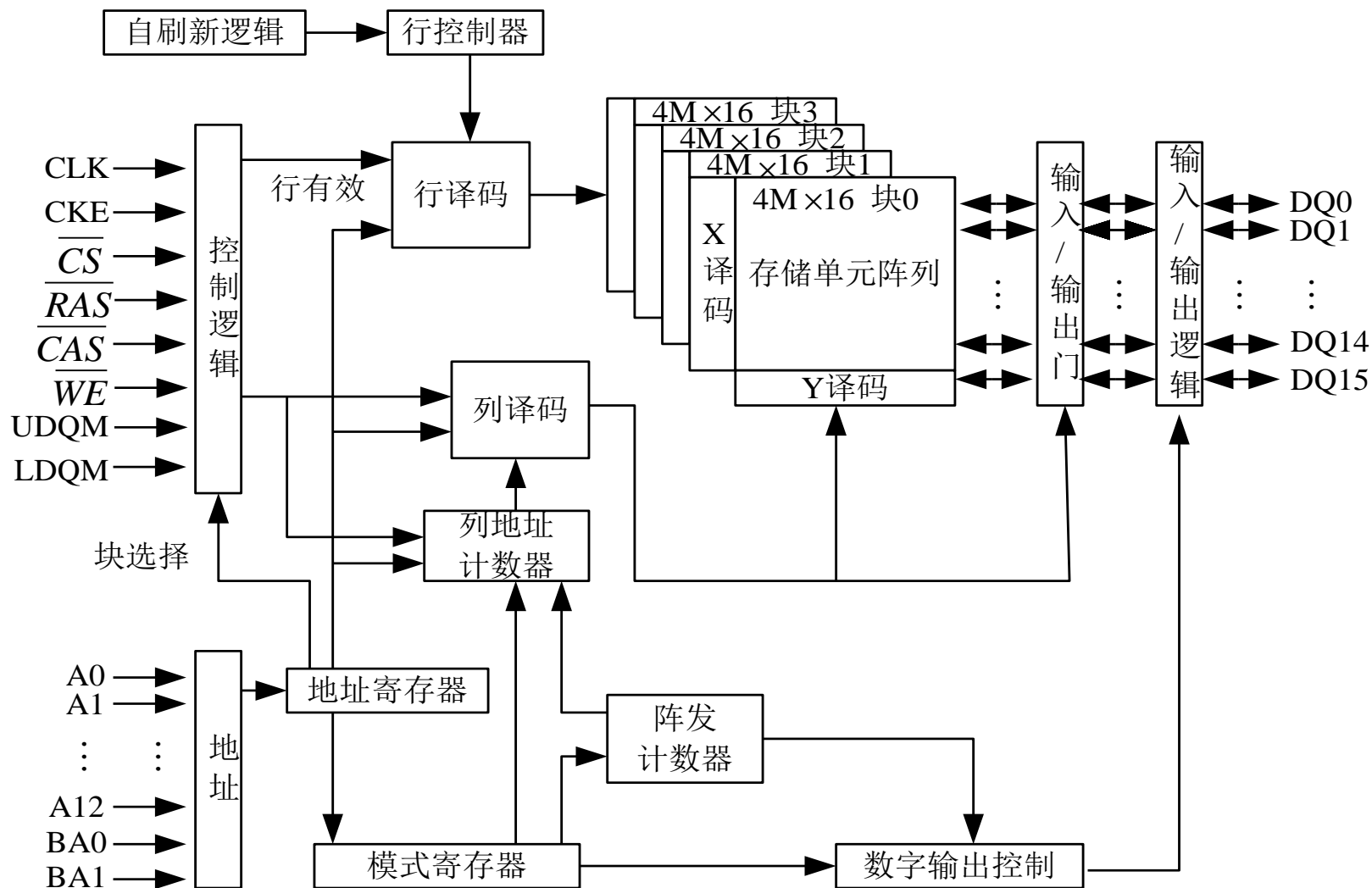
- 一般的CPU都具有和SRAM存储器接口相连的总线，因此连接方法也比较简单。微处理器与随机存储器接口的信号线一般有：
- 片选信号线CE：用于选中该芯片。
- 读/写控制信号线OE和WE：控制芯片数据引脚的传送方向。
- 地址线：用于指明读/写单元的地址。
- 数据线：双向信号线，用于数据交换。



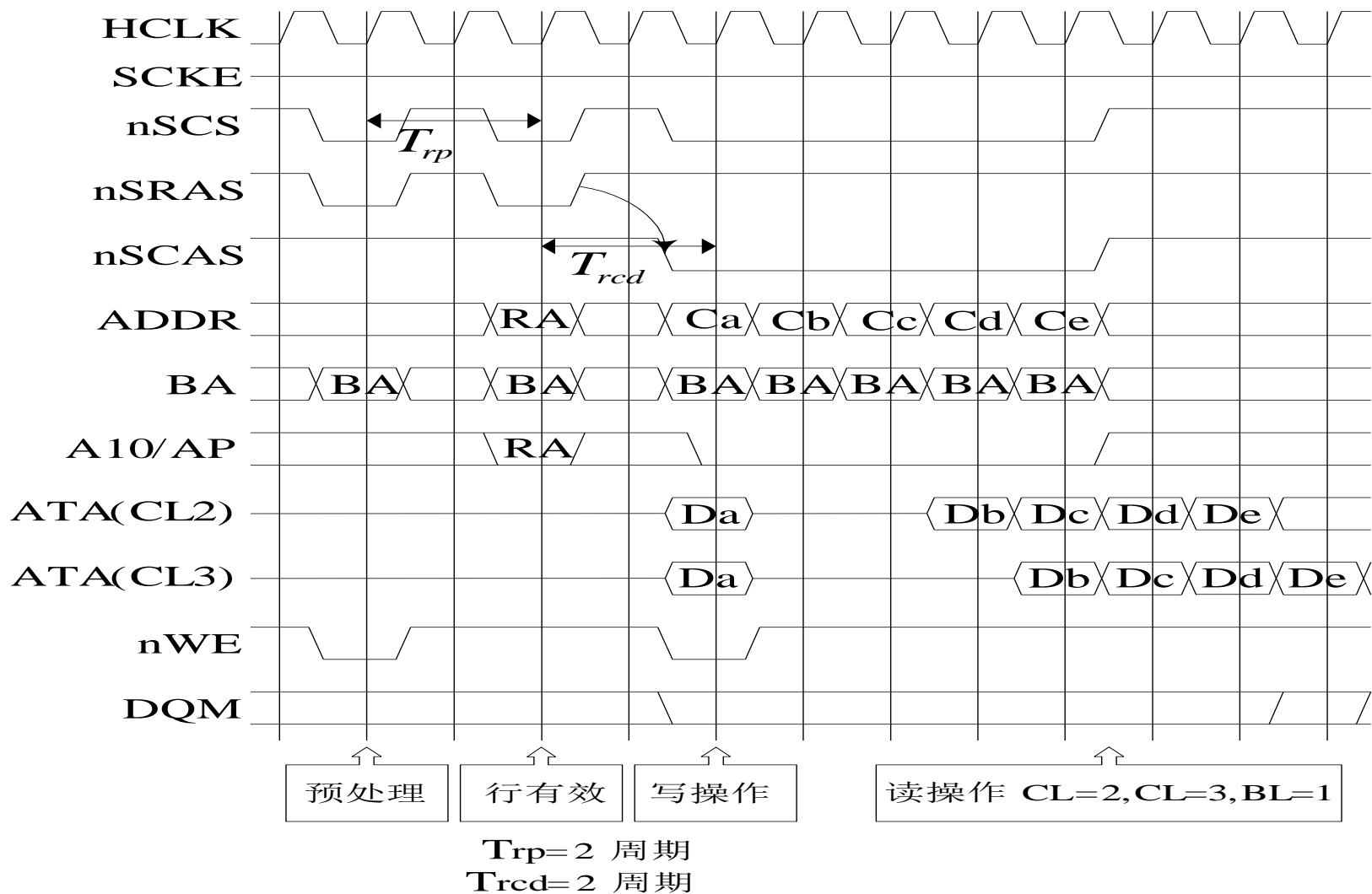
SDRAM存储器及其接口

- DRAM是动态存储器Dynamic RAM的缩写，SDRAM是Synchronous DRAM的缩写，即同步动态存储器的意思。
- BA0、BA1是块地址引脚，在 / RAS有效时，所选中的存储块被激活，在 / CAS有效时，所选中的存储块可进行读 / 写操作；
- / CS、 / WE、 / RAS、 / CAS分别是片选、写、行地址选通、列地址选通；
- LDQM、UDQM是用于控制输入 / 输出数据的；CLK是时钟信号引脚，SDRAM的所有输入是在CLK上升沿有效，CKE是时钟信号使能引脚，当其无效时，SDRAM处于省电模式。

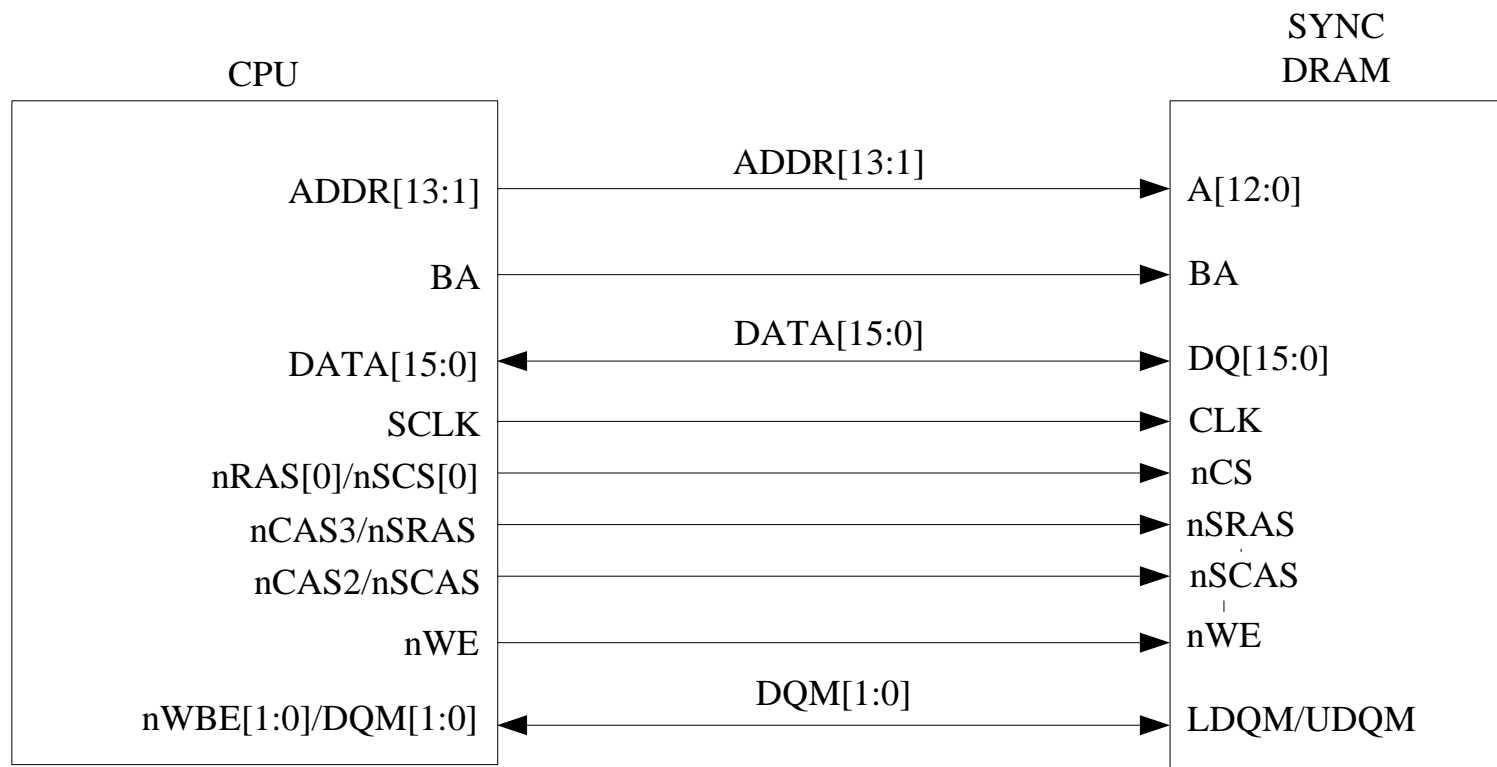
SDRAM结构



SDRAM工作时序



SDRAM存储器及其接口



16位数据存取，半字对齐。

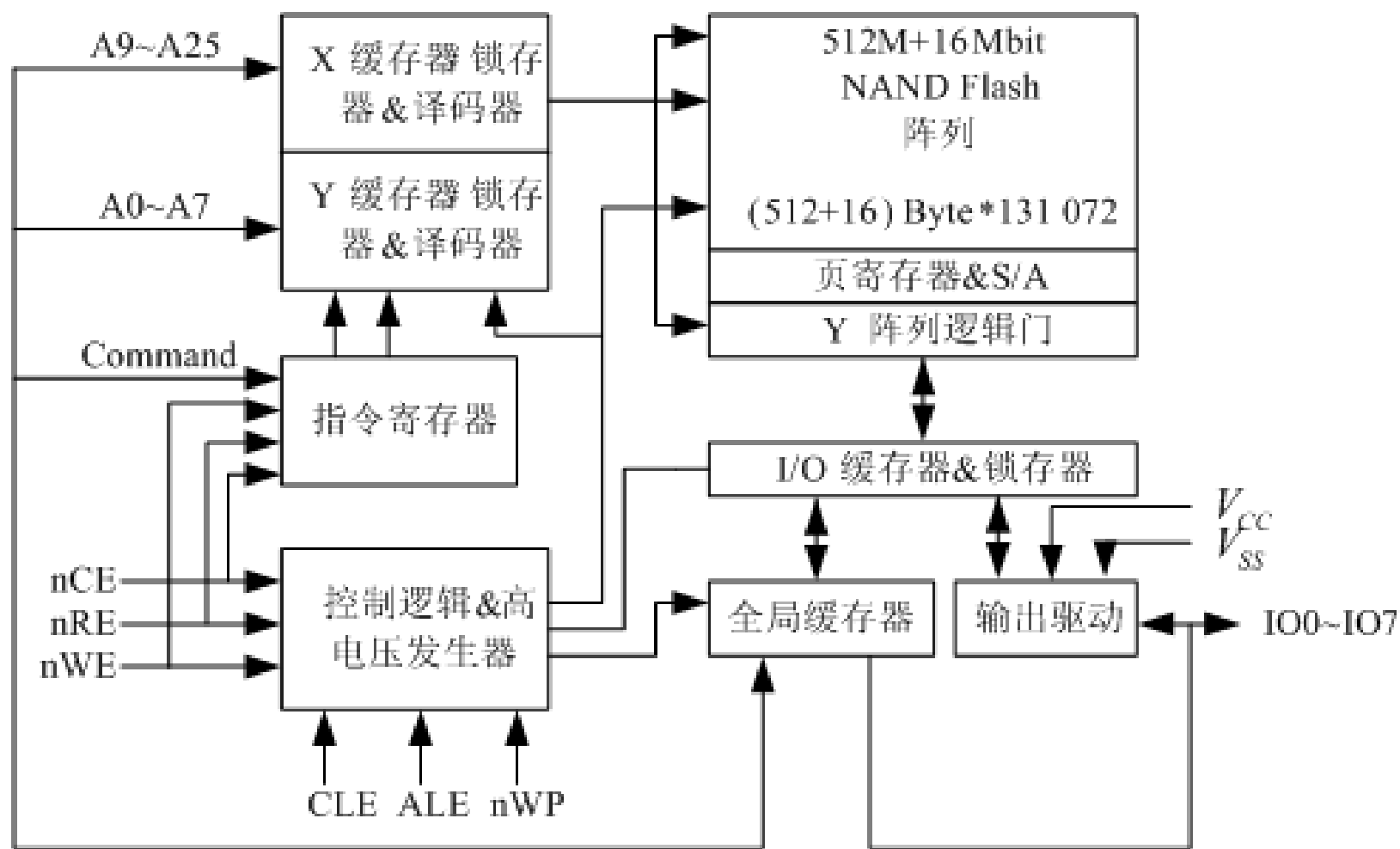
5.1.3 NOR FLASH和NAND FLASH

- NOR和NAND是现在市场上两种主要的非易失闪存技术。
- NOR Flash的读取速度比NAND Flash稍快一些，NAND Flash 的擦除和写入速度比NOR Flash快。
- NOR Flash带有SRAM接口，NAND Flash器件使用复杂的I/O口来串行的存取数据，。
- NAND Flash结构可以在给定的尺寸内提供更高的存储容量。
- NAND Flash中每个块的最大擦写次数是一百万次，而NOR Flash的擦写次数是十万次。

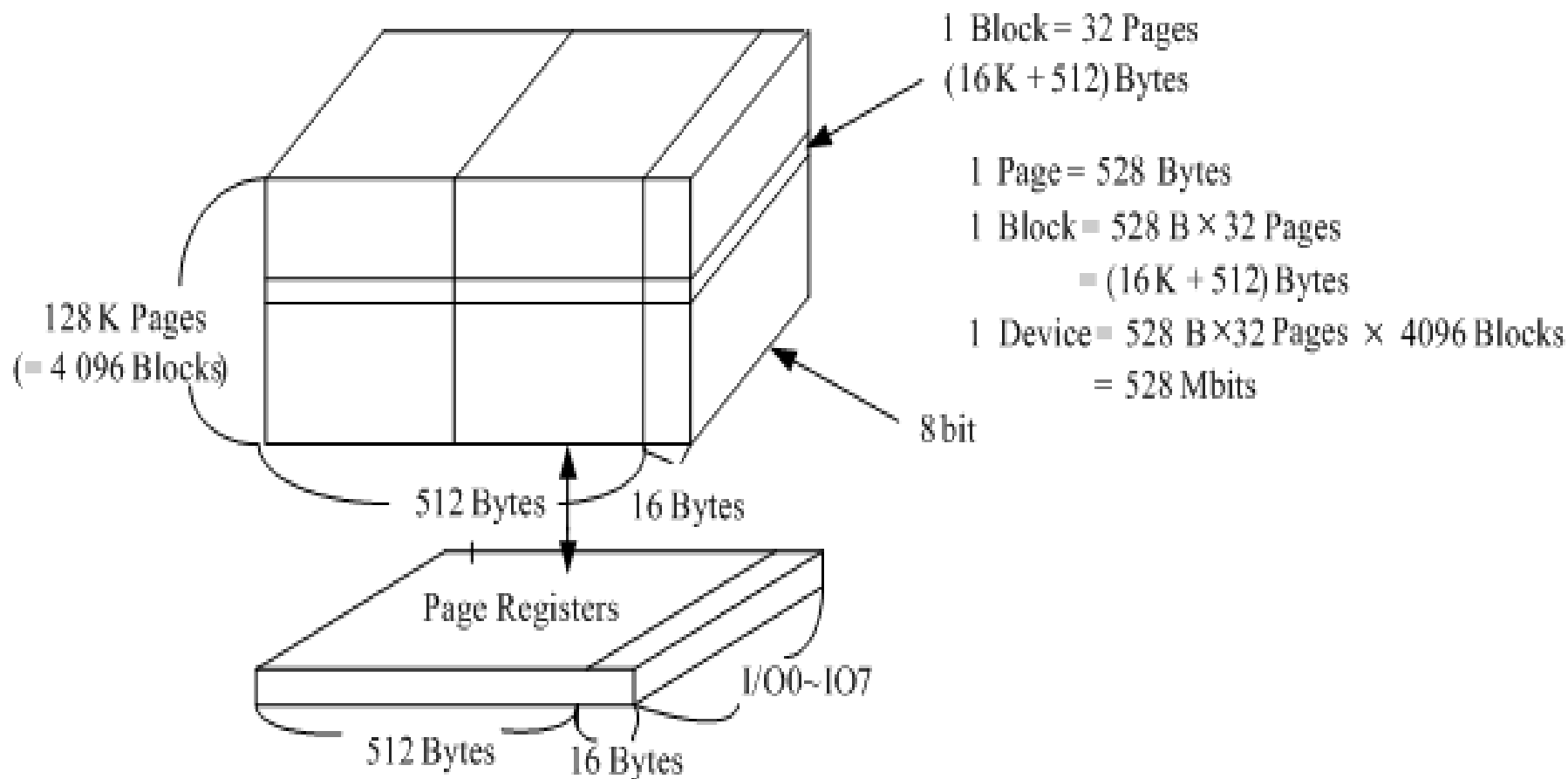
NAND Flash简介

- 以页为单位进行读和编程操作，以块为单位进行擦除操作。
- 数据、地址采用同一总线。实现串行读取。随机读取速度慢且不能按字节随机编程。
- 芯片尺寸小、引脚少，是位成本最低的固态存储器。
- 芯片包含有失效块。失效块不会影响有效块的性能，但设计者需要将失效块在地址映像表中屏蔽起来。

NAND Flash结构



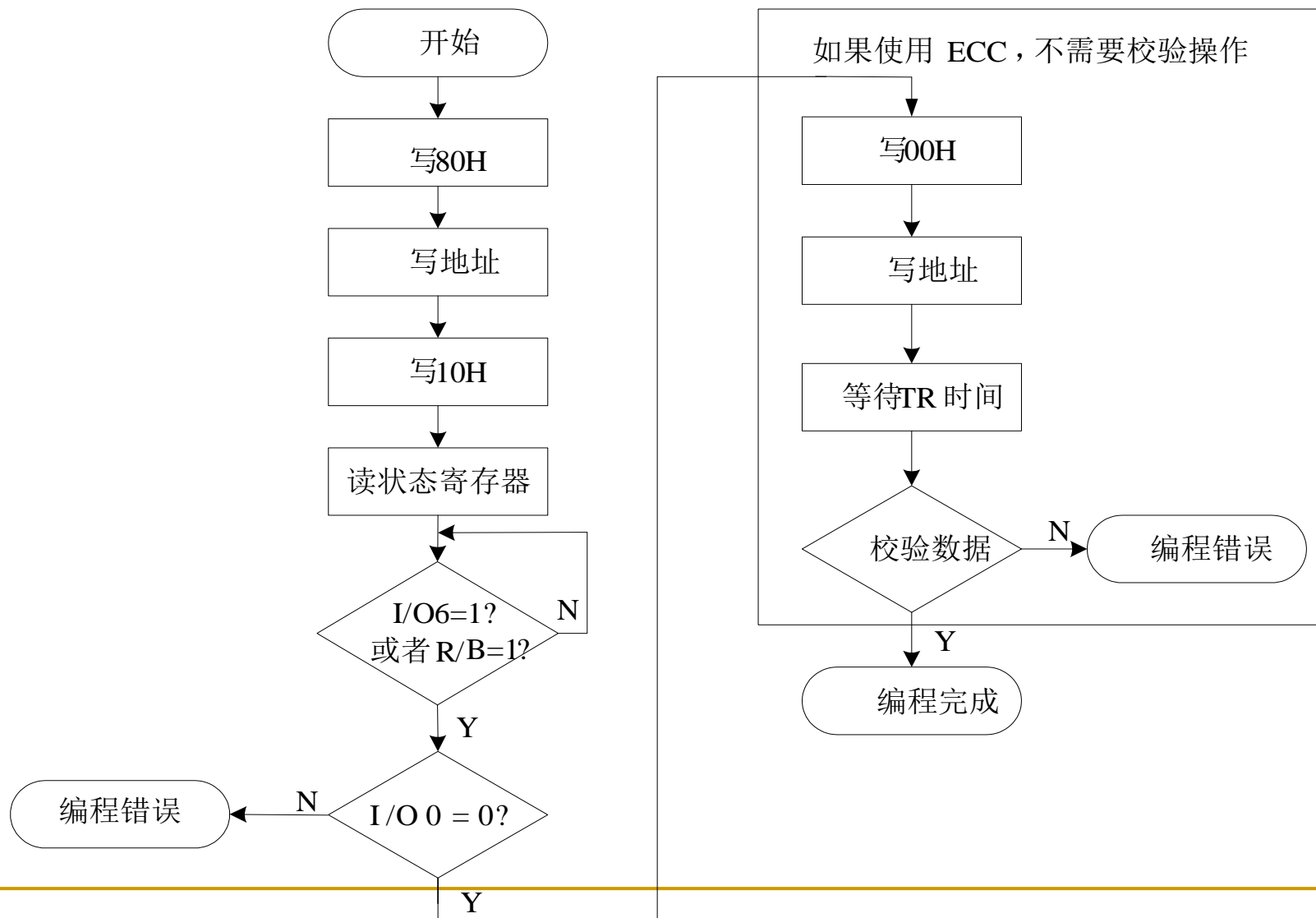
NAND Flash结构



NAND Flash操作

- K9F1208芯片有4096个Block，每个Block有32个Page，每个Page有528个Byte，Block是Nand Flash中最大的操作单元，擦除是以Block为单位完成的，而编程和读取是以Page为单位完成的。因此，对NAND Flash的操作要形成以下三类地址：
- 块地址（Block Address）；
- 页地址（Page Address）；
- 页内地址（Column Address）；
- 由于NAND Flash的数据线和地址线是复用的，因此，在传送地址时要用4个时钟周期来完成

NAND Flash写块操作流程



NAND Flash和NOR Flash比较

- 1988年，Intel首先开发出NOR Flash技术；
- 1989年，东芝公司发表了NAND Flash结构的存储器。
- 彻底改变了原先由EPROM和EEPROM一统天下的局面
- **NAND flash**结构,强调降低每比特的成本,更高的性能,并且象磁盘一样可以通过接口轻松升级

- NOR Flash的特点是**芯片内执行**(XIP, eXecute In Place), 这样应用程序可以直接在flash闪存内运行, 不必再把代码读到系统RAM中。
- NOR Flash的**传输效率很高**, 在1~4MB的小容量时具有很高的成本效益, 但是**很低的写入和擦除速度**大大影响了它的性能。
- NAND Flash结构能提供极高的单元密度, 可以达到高存储密度, 并且写入和擦除的速度也很快。应用NAND的困难在于flash的管理和需要特殊的系统接口

■ 接口方面差别:

- ❑ NOR Flash带有SRAM接口,有足够的地址引脚来寻址,可以很容易地存取其内部的每一个字节,可以像其他SRAM存储器那样与微处理器连接;
 - ❑ NAND Flash器件使用复杂的I/O口来串行地存取数据,各个产品或厂商的方法还各不相同,因此,与微处理器的接口复杂。
- ## ■ NAND Flash读和写操作采用512字节的块,这一点类似硬盘管理操作,很自然地,基于NAND Flash的存储器就可以取代硬盘或其他块设备。

■ 性能比较

- ❑ NOR的读速度比NAND快一些。
- ❑ NAND的写入速度比NOR快很多。
- ❑ NAND的4ms擦除速度远比NOR的5s快。
- ❑ 大多数写入操作需要先进行擦除操作。
- ❑ NAND的擦除单元更小,相应的擦除电路更少。

■ 容量和成本

- NAND flash的单元尺寸几乎是NOR flash器件的一半
- NOR flash占据了容量为1~16MB闪存市场的大部分，而NAND flash只是用在8~128MB的产品当中
- NOR flash主要应用在代码存储介质中
- NAND flash适合于数据存储，NAND flash在CompactFlash、Secure Digital、PC Cards和MMC存储卡市场上所占份额最大

5.2 存储系统机制

- **5.2.1 存储器接口方式**
- **5.2.2 高速缓存机制(CACHE)**
- **5.2.3 存储管理单元(MMU)**

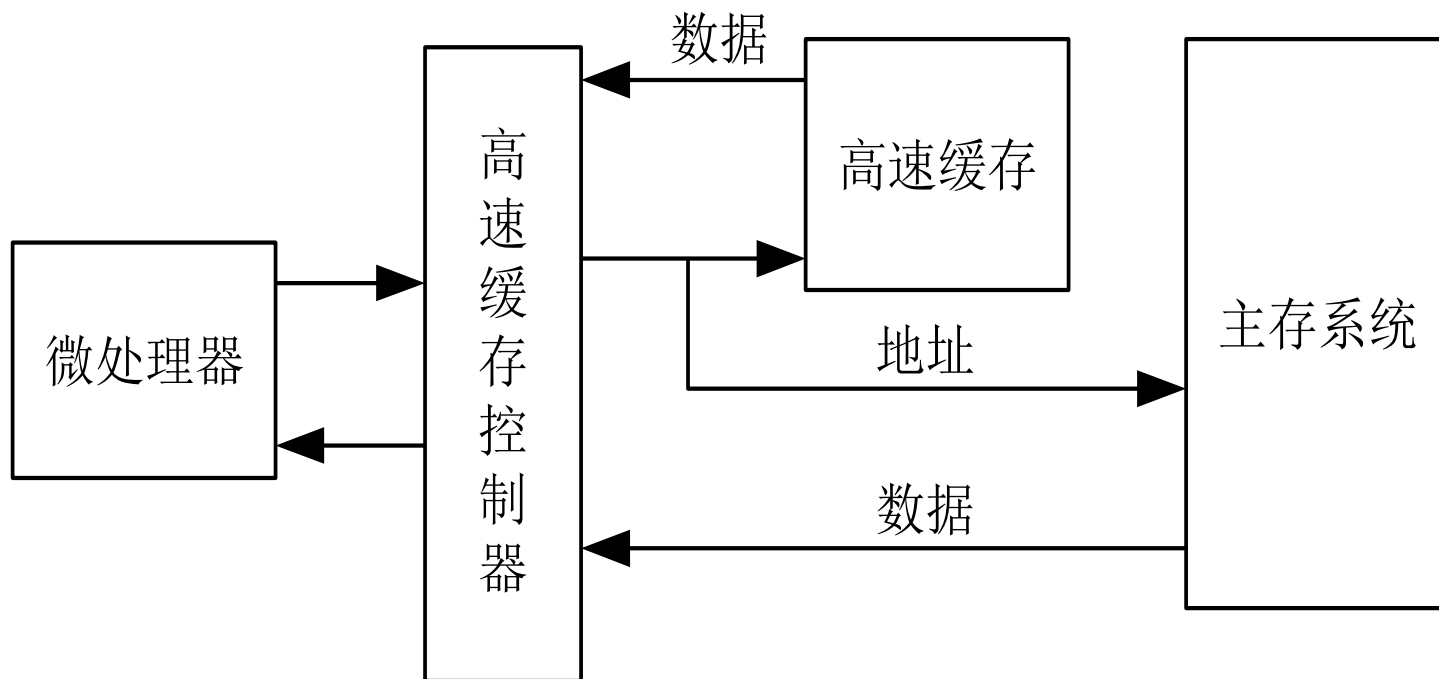
存储器接口方式

- **SRAM**型的全地址/数据总线接口：这种类型的地址线数目和片内存储单元数一一对应，接口比较简单。
- **DRAM**型动态存储器接口：存储单元需要定期地刷新。**CPU**与其接口的信号线除了有与**SRAM**相同的信号线外，还有**RAS**（行地址选择）信号线和**CAS**（列地址选择）信号线。
- 串行存储器接口：与**CPU**以串行的方式传送地址和数据，传送速度相对较慢，多用于嵌入式系统的辅助存储器。

高速缓存机制 (CACHE)

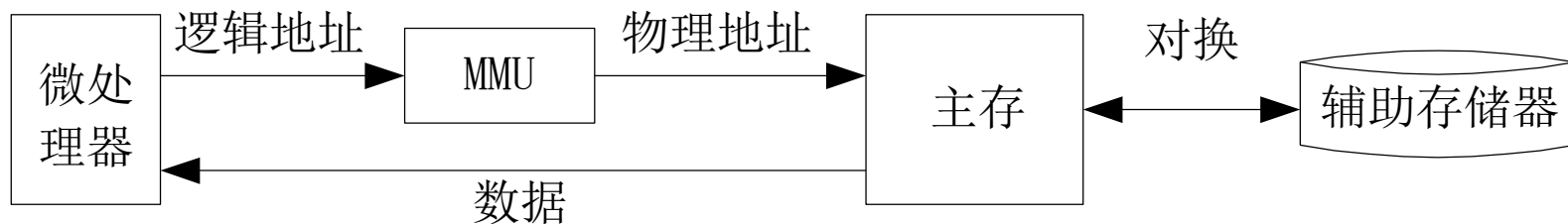
- ❑ 高速缓存控制器是微处理器用于控制访问高速缓存及主存系统的桥梁，它处于微处理器和高速缓存及主存系统之间
- ❑ 用于解决主存访问速度与**CPU**处理速度不相匹配的一种部件（由集成于**CPU**芯片中的专门的高速存取电路实现）。
- ❑ 或用于解决辅存访问速度与**CPU**处理速度不相匹配的一种部件（由主存的一部分实现）。
- ❑ 需要解决缓存内容与原内容不一致的问题

高速缓存机制 (CACHE)



存储管理单元(MMU)

- 存储管理单元（MMU）是集成在微处理器芯片内部、专门管理外部存储器总线的一部分硬件。主要用来完成虚实地址之间的转换。越来越多的微处理器芯片均带有MMU。
- MMU完成的主要功能有：
 - 将主存地址从虚拟存储空间映射到物理存储空间。
 - 存储器访问权限控制。
 - 设置虚拟存储空间的缓冲特性等。

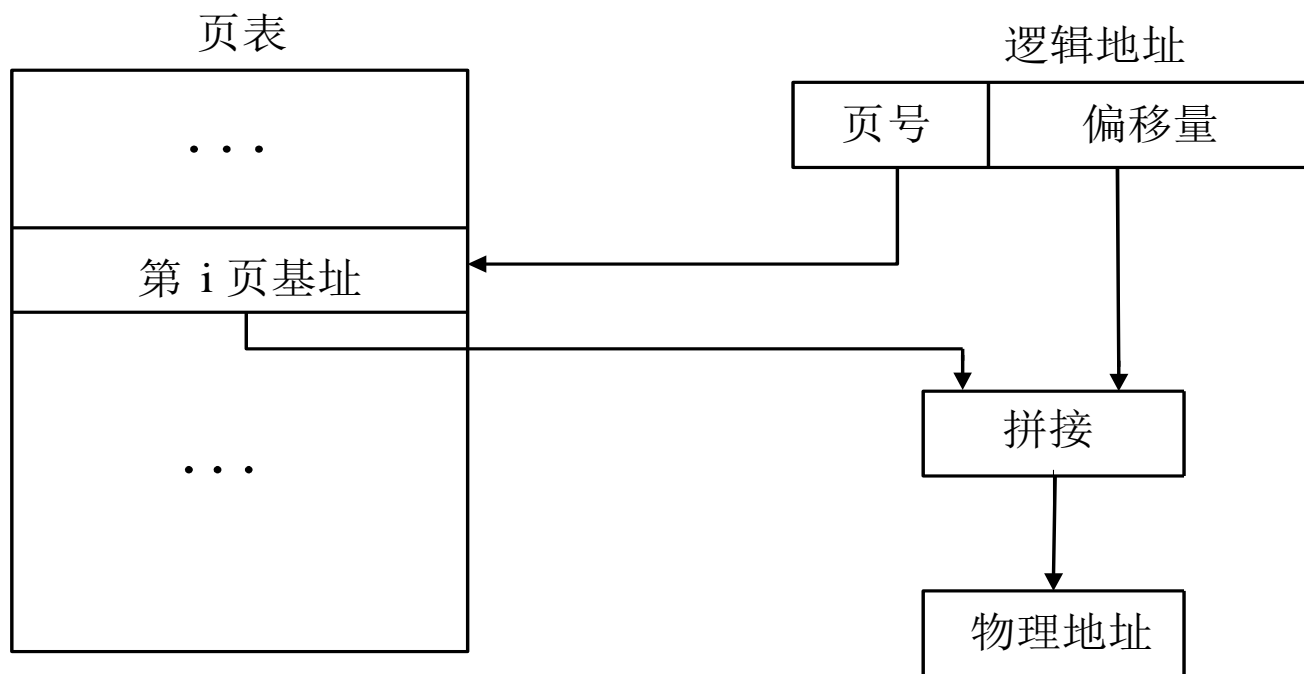


MMU管理方式

- 分段方式：分段方式支持较大的、任意大小的内存区域
- 分页方式：分页方式支持较小的、固定大小的内存区域
- 段页方式：段页方式介于分段方式和分页方式之间。等。每种方式都有其特点

分页虚拟存储管理

- 虚拟存储空间分成一个个固定大小的页，把物理主存储的空间也分成同样大小的一个个页。通过查询存放在主存中的页表，来实现虚拟地址到物理地址的变换。



■ MMU概括

- ❑ 负责虚拟地址到物理地址的映射，并提供硬件机制的内存访问权限检查。
- ❑ MMU使得每个用户进程拥有自己独立的地址空间，并通过内存访问权限的检查保护每个进程所用的内存不被其他进程破坏
- ❑ 大程序可以在较小的内存空间中运行
- ❑ 内存中可以同时装入更多的程序并发执行
- ❑ 重点就在于地址映射：页表的结构与建立、映射的过程
- ❑ 由于虚拟地址空间远大于物理地址，可能多块虚拟地址空间映射到同一块物理地址空间，或者有些虚拟地址空间没有映射到具体的物理地址空间上去（使用到时再映射）。

MMU概括

- CPU通过地址来访问内存中的单元，如果CPU没有MMU，或者有MMU但没有启用，CPU在访问内存时发出的地址将直接传送到地址总线上，使具有相同地址的物理存储器被读写。这称为物理地址
- 如果CPU启用了MMU，CPU核发出的地址将被MMU截获，从CPU到MMU的地址称为虚拟地址。
- 而MMU将这个地址翻译成真正的物理地址发送到地址总线上，也就是将虚拟地址映射成物理地址，前面的页表就起作用了

5.3 S3C2410存储系统

- 5.3.1 S3C2410存储空间
- 5.3.2 S3C2410存储器接口设计

嵌入式系统中，主存储器和辅助存储器中的存储单元是通过地址来识别的。在主存储器和辅助存储器的设计中，主要工作就是完成地址分配，确定各芯片对应的地址范围。

S3C2410芯片回顾:

1. 引脚介绍

S3C2410共有272个引脚, 主要分为总线控制信号、各类元器件接口信号以及电源时钟控制信号, 引脚分布底视图见右图。

其信号可以分成

addr0---addr26、

data0---data31、

GPA0---GPA22

GPB10、GPC15、

GPD15、GPE15、

GPF7、GPG15、

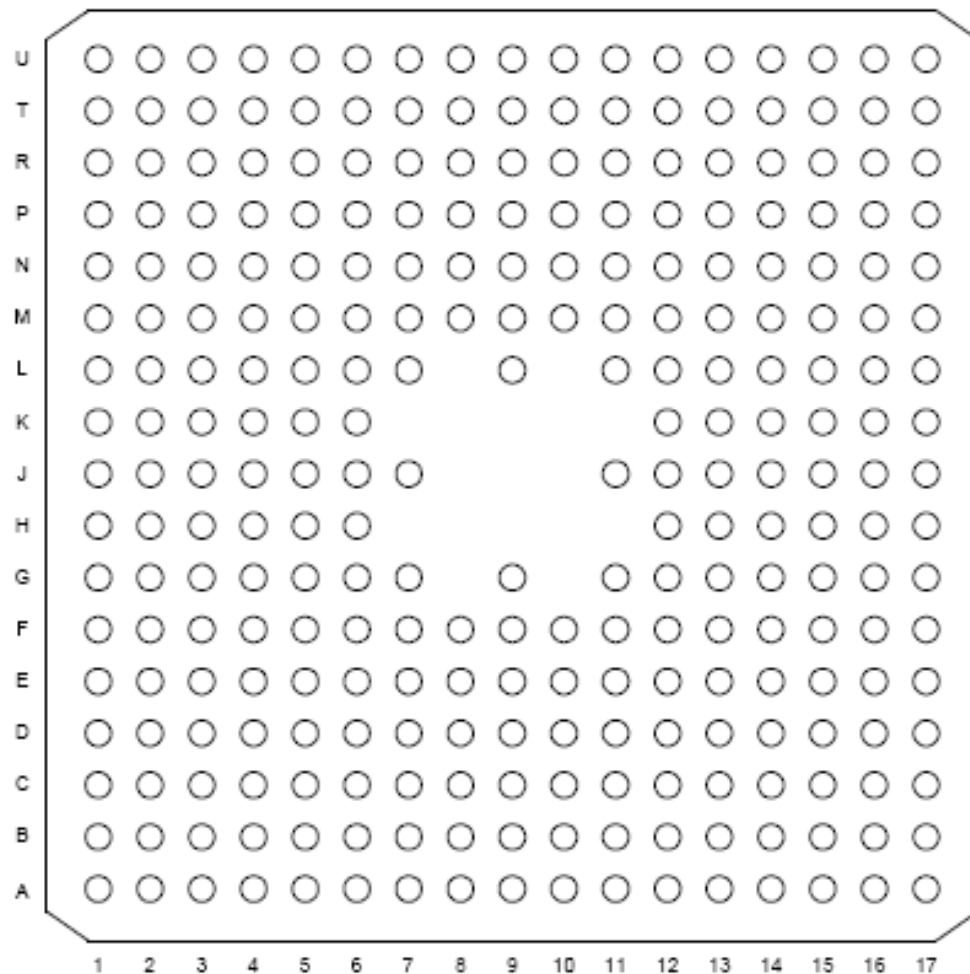
GPH10、EINT23、

nGCS0—nGCS7、

AIN7、IIC、SPI、

OM0---OM3

等, 大部分都是复用的。



BOTTOM VIEW

■ 2410主要特性

- ❑ 具有**16KB指令Cache**、**16KB数据Cache**和存储器管理单元**MMU**。
- ❑ 外部存储器控制器，可扩展**8组**，每组**128MB**，总容量达**1GB**；支持从**Nand flash**存储器启动。
- ❑ **55个**中断源，可以设定**1个**为快速中断，有**24个**外部中断，并且触发方式可以设定。
- ❑ **4通道**的**DMA**，并且有外部请求引脚。
- ❑ **3个通道**的**UART**，带有**16字节**的**TX/RX FIFO**，支持**IrDA1.0**功能。
- ❑ 具有**2通道**的**SPI**、**1个通道**的**IIC**串行总线接口和**1个通道**的**IIS**音频总线接口。
- ❑ 有**2个USB**主机总线的端口，**1个USB**设备总线的端口。

- ❑ 有**4**个具有**PWM**功能的**16**位定时器和**1**个**16**位内部定时器。
- ❑ **8**通道的**10**位**A/D**转换器，最高速率可达**500kB/s**；提供有触摸屏接口。
- ❑ 具有**117**个通用**I/O**口和**24**通道的外部中断源。
- ❑ 兼容**MMC**的**SD**卡接口。
- ❑ 具有电源管理功能，可以使系统以普通方式、慢速方式、空闲方式和掉电方式工作。
- ❑ 看门狗定时器。
- ❑ 具有日历功能的**RTC**。
- ❑ 有**LCD**控制器，支持**4K**色的**STN**和**256K**色的**TFT**，配置有**DMA**通道。
- ❑ 具有**PLL**功能的时钟发生器，时钟频率高达**203MHz**。
- ❑ 双电源系统：**1.8**内核供电，**3.3V**存储器和**I/O**供电。

2. 2410存储系统接口信号

■ SDRAM/SRAM接口信号

- 1) **nSRAS**(SDRAM Row Address Strobe, O): SDRAM行地址选通
- 2) **nSCAS**(SDRAM Column Address Strobe, O): SDRAM列地址选通
- 3) **nSCS[1:0]**(SDRAM Chip Select, O): SDRAM片选
- 4) **DQM[3:0]**(SDRAM Data Mask, O): SDRAM数据掩码
- 5) **SCLK[1:0]**(SDRAM Clock, o): SDRAM时钟
- 6) **SCKE[1:0]**(SDRAM Clock Enable, O): SDRAM时钟使能
- 7) **nBE[3:0]**(Upper Byte/Lower Byte Enable (In case of 16-bit SRAM), O): 使用16位SRAM时, 用作高字节/低字节使能信号
- 8) **nWBE[3:0]**(Write Byte Enable): 写字节使能

■ NAND Flash接口信号

- 1) **CLE**(Command Latch Enable, O):命令锁存允许
- 2) **ALE**(Address Latch Enable, O): 地址锁存允许
- 3) **nFCE**(NAND Flash Chip Enable, O): NAND Flash芯片使能
- 4) **nFRE**(NAND Flash Read Enable, O): NAND Flash读允许
- 5) **nFWE**(NAND Flash Write Enable, O):NAND Flash写允许
- 6) **nCON**(NAND Flash Configuration, O):NAND Flash配置
- 7) **R/nB**(NAND Flash Ready/Busy, O):NAND Flash准备就绪/忙

■ 总线控制器信号

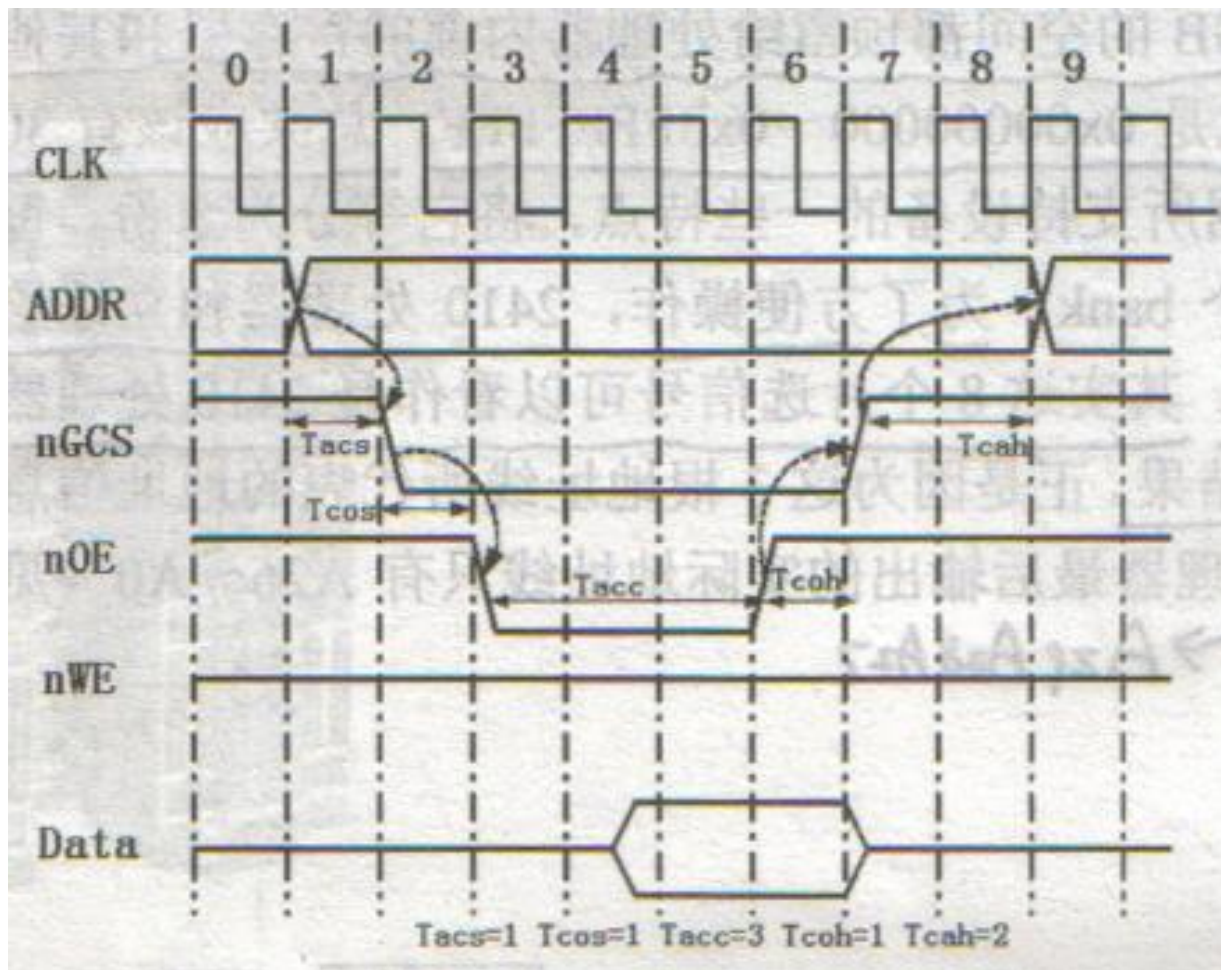
1) OM[1:0] (I): 启动方式选择(或见教材P114表4-6)

OM1	OM0	选择2410的启动方式
0	0	处理器从NAND Flash启动
0	1	处理器从16位宽的ROM启动
1	0	处理器从32位宽的ROM启动
1	1	处理器为测试模式

可直接将2410的这两个引脚接高电平或低电平。如将OM1和OM0接地，表示支持从NAND Flash引导系统，在该方式下，信号nGCS0不用。

- 2) **ADDR[26:0]** (O): 地址总线。输出对应存储器块(bank)的地址。
- 3) **DATA[31:0]** (I/O): 数据总线。总线宽度可编程为8位、16位或32位。
- 4) **nGCS[7:0]** (General Chip Select, O): 通用的片选。存储器地址在每块的地址范围内时有效, 访问周期数和存储器大小可编程。
- 5) **nWE** (I/O): 写允许/写有效信号。当前总线周期处于写周期时有效。
- 6) **nOE** (O): 输出允许/读有效信号。当前总线周期处于读周期时有效。
- 7) **nXBREQ** (Bus hold Request, I): 总线保持请求信号。允许另一个总线主控设备请求控制本地总线, **BACK**有效表示总线请求已被接受。
- 8) **nXBACK** (Bus Hold Acknowledge, O): 总线保持应答信号。有效时表示2410已允许其他主控设备控制本地总线。
- 9) **nWAIT** (I): 等待延迟信号。等待延长当前总线周期, 为低电平时当前总线周期尚未完成。

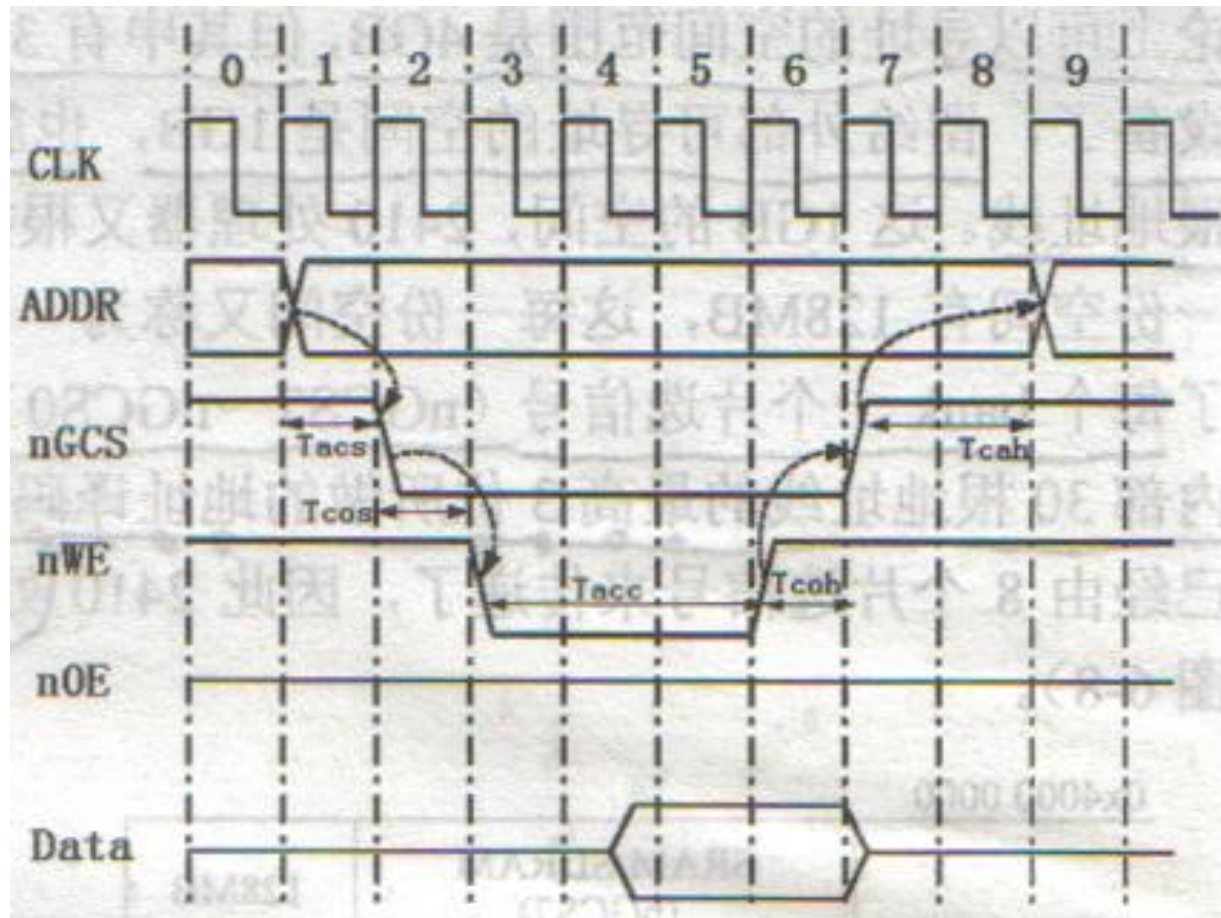
3. 2410总线操作时序



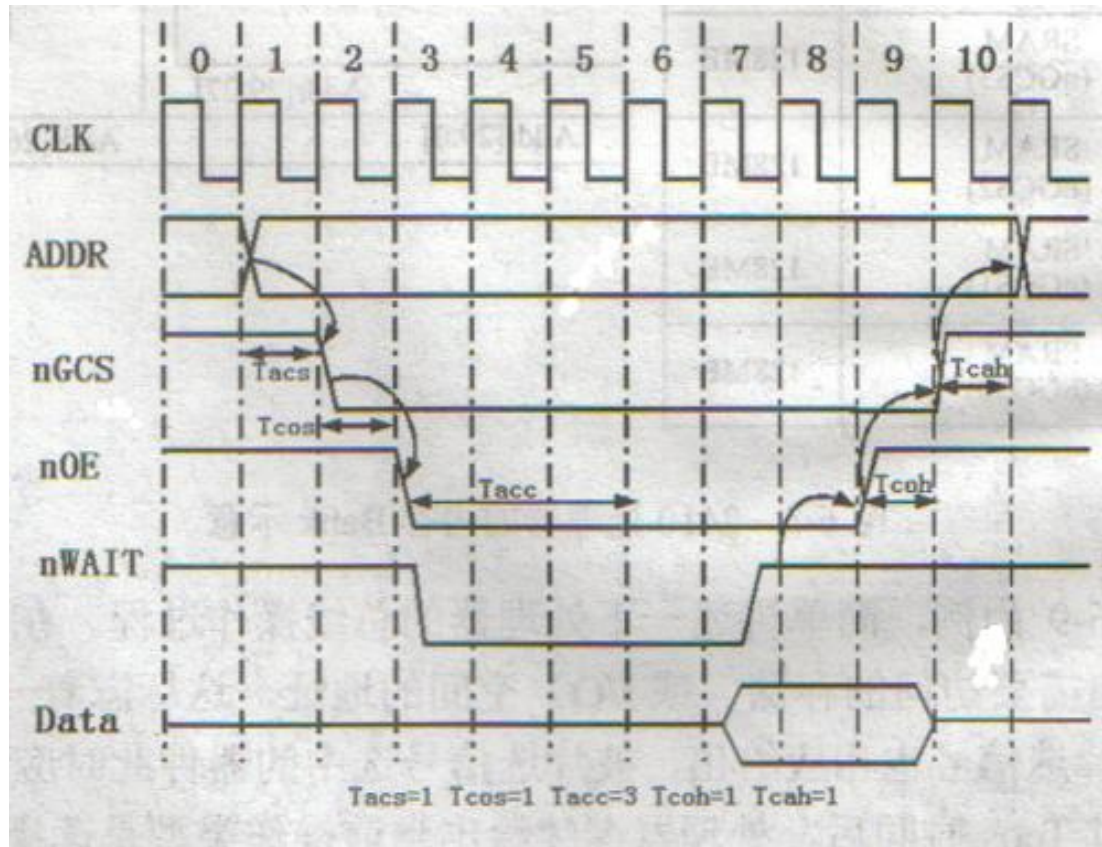
Notes:

T_{acs} : address set-up
time before nGCSn
 T_{cos} : chip selection set-up
time before nOE
 T_{acc} : access time
 T_{coh} : chip selection hold
time after nOE
 T_{cah} : Address hold time
after nGCSn

(a) 总线读操作基本时序



(b) 总线写操作基本时序



(c) 带 $nWAIT$ 信号的总线操作读操作时序

Notes:

Tacs: address set-up time before nGCSn

Tcos: chip selection set-up time before nOE

Tacc: access time

Tcoh: chip selection hold time after nOE

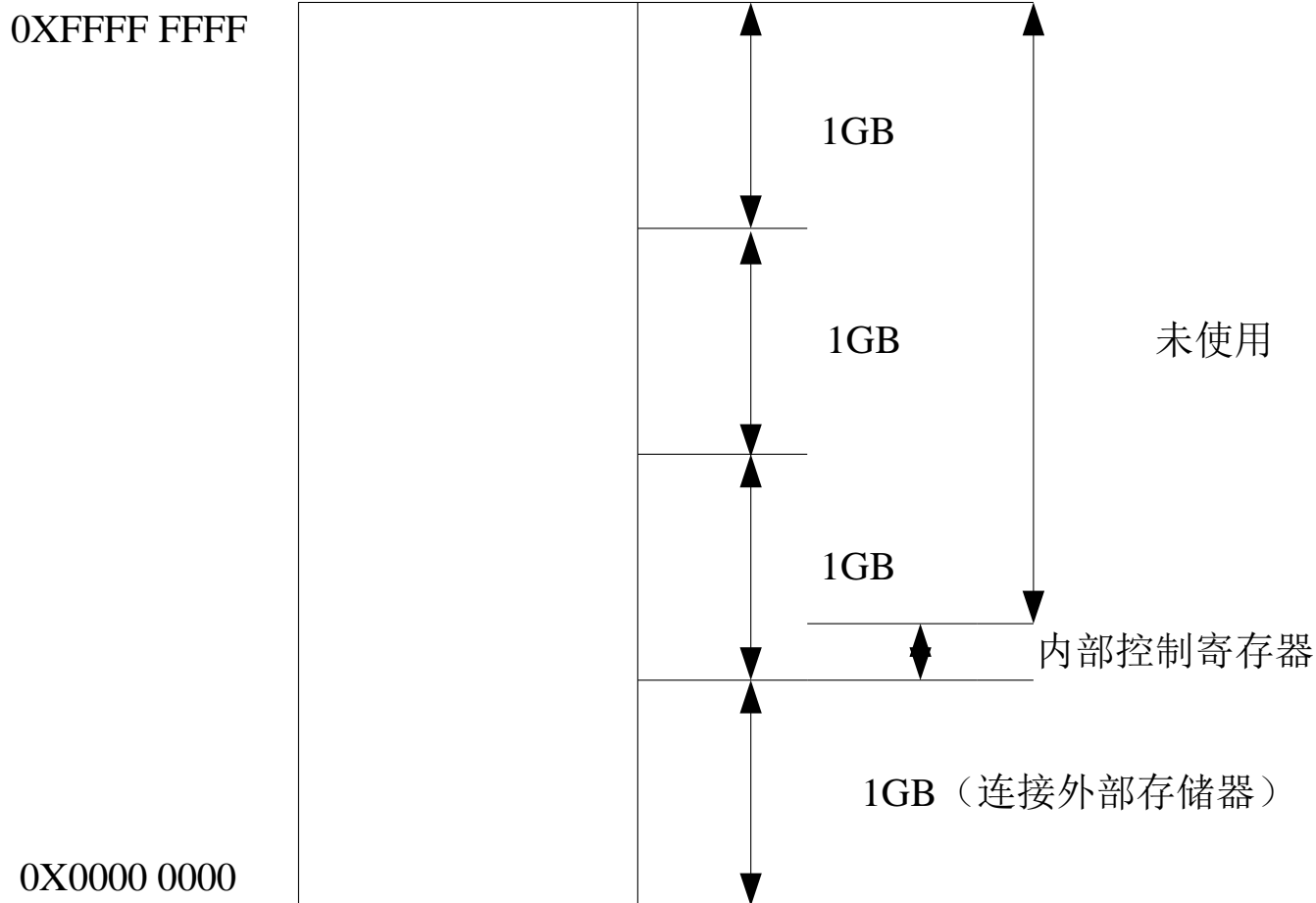
Tcah: Address hold time after nGCSn

需强调的是，在整个总线操作过程中，所涉及的所有时间变量，在**2410**处理器中都可以由软件设置并更改，这样就可以灵活方便地支持各种速率的器件。

4. 2410存储空间

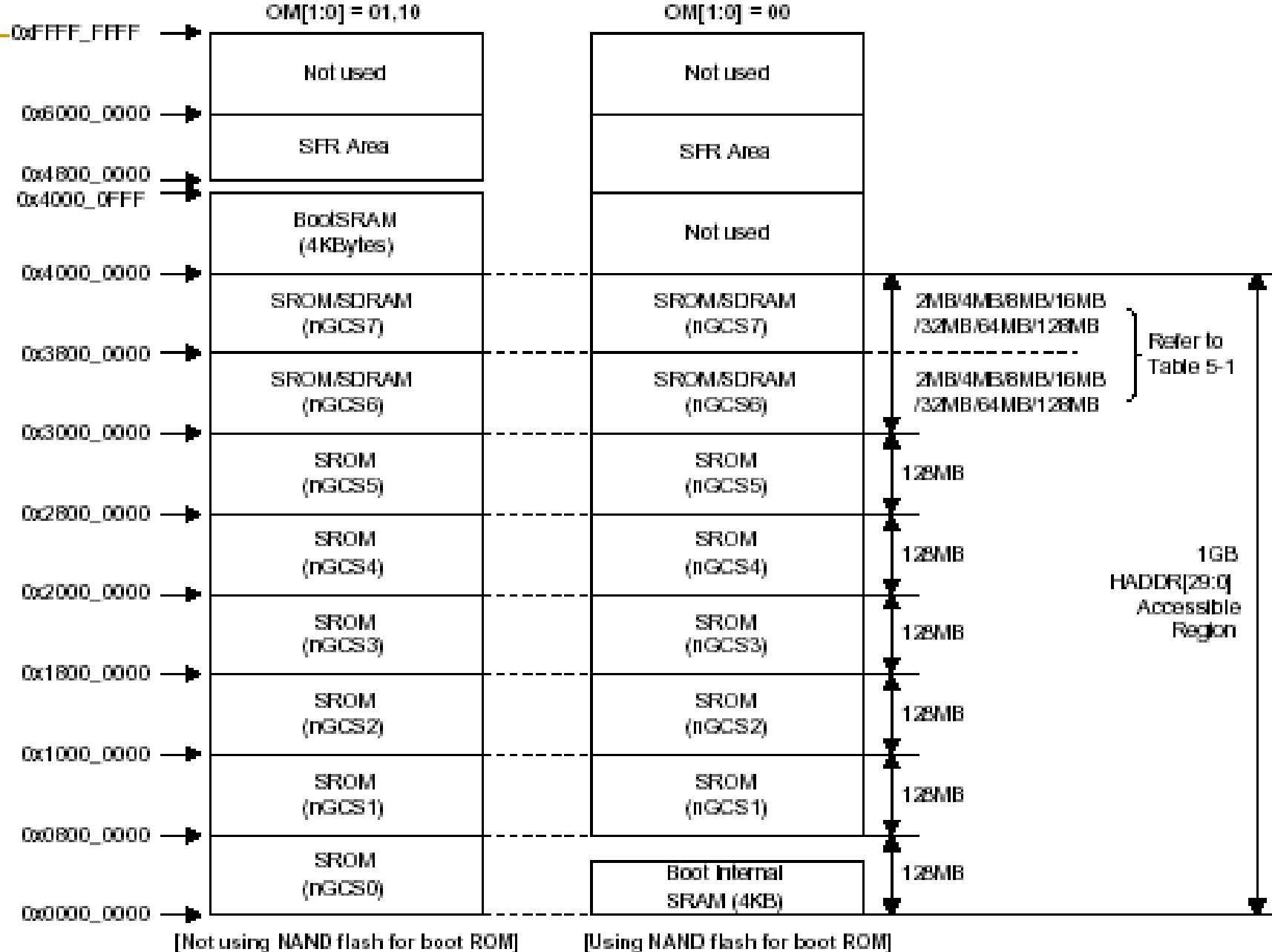
- S3C2410芯片采用的是ARM920T核，理论上地址空间总共有4GB (2^{32})。
- 但其中有3GB的空间都预留给处理器内部的寄存器和设备了。
- 留给外部可寻址的存储器空间是1GB，也就是0x00000000~0x3FFFFFFF总共有30根地址线
- 这1G空间被分成8个存储块，每块(bank)128MB。

S3C2410存储空间



S3C2410存储空间

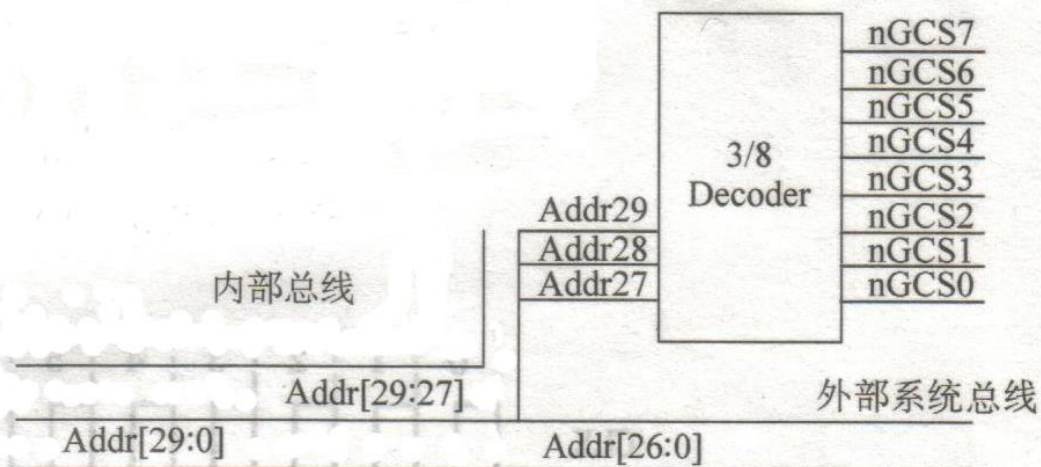
- **S3C2410整个地址空间（寻址范围）为4GB。**
- **S3C2410芯片可连接外部存储器的可寻址空间是1GB。**
- **有一部分地址微处理器内部占用。用于控制寄存器和I/O端口使用。**
- **有大部分地址空间未被使用或不能使用。**



NOTES:

1. SRAM means ROM or SRAM type memory.
2. SFR means Special Function Register.

0x4000 0000		
0x3800 0000	SRAM/SDRAM (nGCS7)	128MB
0x3000 0000	SRAM/SDRAM (nGCS6)	128MB
0x2800 0000	SRAM (nGCS5)	128MB
0x2000 0000	SRAM (nGCS4)	128MB
0x1800 0000	SRAM (nGCS3)	128MB
0x1000 0000	SRAM (nGCS2)	128MB
0x0800 0000	SRAM (nGCS1)	128MB
0x0000 0000	SRAM (nGCS0)	128MB



2410独立地给每个块(bank)一个片选信号(nGCS7~nGCS0)。在理解上，这8个片选信号可看作是2410内部30根地址线的最高3位(A29、A28、A27)所做的地址译码的结果（见上图）。

- **0号存储块**可以外接**SRAM**类型的存储器或者具有**SRAM**接口特性的**ROM**存储器（如**NOR Flash**），其数据总线宽度应设定为**16位**或**32位**中的一种。
- 当**0号存储块**作为**ROM**区，完成引导装入工作时（从**0x00000000**启动），**0号存储块**的总线宽度应在第一次访问**ROM**前根据**OM1**、**OM0**在复位时的逻辑组合来确定（见下表）

OM1 OM0	选择2410的启动方式
0 0	处理器从NAND Flash启动
0 1	处理器从16位宽的ROM启动
1 0	处理器从32位宽的ROM启动
1 1	处理器为测试模式

- 1号存储块到5号存储块也可以外接**SRAM**类型的存储器或者具有**SRAM**接口特性的**ROM**存储器（如**NOR Flash**），其数据总线宽度应设定为**8位、16位或32位**。
- 6号存储块、7号存储块可以外接**SDRAM**类型的存储器，它们的块容量可改变，且7号存储块的起始地址也可改变。
- 6号和7号块的存储地址分配情况

地址	2MB	4MB	8MB	16MB	32MB	64MB	128MB
Bank 6							
起始地址	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000
终址地址	0x301f_ffff	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff
Bank 7							
起始地址	0x3020_0000	0x3040_0000	0x3080_0000	0x3100_0000	0x3200_0000	0x3400_0000	0x3800_0000
终址地址	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff	0x3fff_ffff

S3C2410存储空间特点

- 支持小端 / 大端模式（可通过软件选择）。
- 8个存储块中，6个用于SRAM或ROM，另2个用于SDRAM、SRAM、ROM。
- 8个存储块中，7个存储块有固定起始地址，1个存储块起始地址可变。
- 支持异步定时，可用nWAIT(等待)信号来扩展外部存储器的读 / 写周期。
- 可编程的总线访问宽度8/16/32位，但Bank0不能通过软件编程方式设置。
- 在SDRAM中支持自主刷新和省电模式。
- 所有存储器Bank可编程访问周期。
- 存储器相关寄存器（见书）

2410存储器控制寄存器

- ❑ 地址空间 **0x48000000~0x5FFFFFFF** 范围有大量的特殊功能寄存器 **SFR (Special Function Registers)** 用于实现芯片内部及外围部件的控制。
- ❑ 如2410处理器的工作频率可达203MHz，但不是仅工作在于该频率，可以通过修改内部特殊寄存器的值，使处理器工作于不同的频率。
- ❑ 这里重点介绍2410存储器控制寄存器。

存储器的控制寄存器

内存控制器为访问外部存储空间提供存储器控制信号，S3C2410X存储器控制器共有13个寄存器。

寄存器	地 址	功 能	操作	复位值
BWSCON	0x48000000	总线宽度和等待控制	读/写	0x0
BANKCON0	0x48000004	BANK0控制	读/写	0x0700
BANKCON1	0x48000008	BANK1控制	读/写	0x0700
BANKCON2	0x4800000C	BANK2控制	读/写	0x0700
BANKCON3	0x48000010	BANK3控制	读/写	0x0700
BANKCON4	0x48000014	BANK4控制	读/写	0x0700
BANKCON5	0x48000018	BANK5控制	读/写	0x0700
BANKCON6	0x4800001C	BANK6控制	读/写	0x18008
BANKCON7	0x48000020	BANK7控制	读/写	0x18008
REFRESH	0x48000024	SDRAM刷新控制	读/写	0xAC0000
BANKSIZE	0x48000028	可变的组大小设置	读/写	0x0
MRSRB6	0x4800002C	BANK6模式设置	读/写	xxx
MRSRB7	0x48000030	BANK7模式设置	读/写	xxx

1、总线宽度和等待控制寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST7	WS7	DW7		ST6	WS6	DW6		ST5	WS5	DW5		ST4	WS4	DW4	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST3	WS3	DW3		ST2	WS2	DW2		ST1	WS1	DW1		X	DW0		X

STn: 控制存储器组n的UB/LB引脚输出信号。

1: 使UB/LB与nBE[3: 0]相连;

0: 使UB/LB与nWBE[3: 0]相连

WSn: 使用/禁用存储器组n的WAIT状态

1: 使能WAIT; 0: 禁止WAIT

DWn: 控制存储器组n的数据线宽

00: 8位; 01: 16位; 10: 32位; 11: 保留

2、BANKn---存储器组控制寄存器 (n=0--5)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Tacs		Tcos		Tacc				Tcoh		Tcah		TACP		PMC	

Tacs: 设置nGCSn有效前地址的建立时间

00: 0个; **01:** 1个; **10:** 2个; **11:** 4个时钟周期

Tcos: 设置nOE有效前片选信号的建立时间

00: 0个; **01:** 1个; **10:** 2个; **11:** 4个时钟周期

Tacc: 访问周期

000: 1个; **001:** 2个; **010:** 3个; **011:** 4个时钟

100: 6个; **101:** 8个; **110:** 10个; **111:** 14个

Tcoh: nOE无效后片选信号的保持时间

00: 0个; **01:** 1个; **10:** 2个; **11:** 4个时钟

Tcah: nGCSn无效后地址信号的保持时间

00: 0个; **01:** 1个; **10:** 2个; **11:** 4个时钟

Tacp: 页模式的访问周期

00: 2个; **01:** 3个; **10:** 4个; **11:** 6个时钟

PMC: 页模式的配置, 每次读写的数据数

00: 1个; **01:** 4个; **10:** 8个; **11:** 16个

注: **00**为通常模式。

注: 紫色为实验箱上的配置, 其值为**0x0700**

3、BANK6/7---存储器组6/7控制寄存器

31												17	16	15
保留													MT		
14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Tacs		Tcos		Tacc			Tcoh		Tcah		TACP/ Trcd		PMC/ SCAN		

MT: 设置存储器类型

00: ROM或者SRAM, [3: 0]为TACP和PMC;

11: SDRAM, [3: 0]为Trcd和SCAN; **01、10:**
保留

Trcd: 由行地址信号切换到列地址信号的延时时钟数

00: 2个时钟; **01:** 3个时钟; **10:** 4个时钟

SCAN: 列地址位数

00: 8位; **01:** 9位; **10:** 10位

4、REFRESH---刷新控制寄存器

31				24	23	22	21	20	19	18	17	16
保 留						REFEN	TREFMD	Trp		Tsrc		保留	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留					Refresh_count										

REFEN: 刷新控制。 **1:** 使能刷新； **0:** 禁止刷新

TREFMD: 刷新方式。 **1:** 自刷新 **0:** 自动刷新

Trp: 设置**SDRAM**行刷新时间（时钟数）

00: 2个时钟； **01:** 3个； **10:** 3个； **11:** 4个时钟

Tsrc: 设置**SDRAM**行操作时间（时钟数）

00: 4个时钟； **01:** 5个； **10:** 6个； **11:** 7个时钟

注： **SDRAM**的行周期= Trp + Tsrc。

Refresh_count: 刷新计数值

Refresh_count: 刷新计数器值

计算公式:

$$\text{刷新周期} = (2^{11} - \text{Refresh_count} + 1) / \text{HCLK}$$

例子: 设刷新周期=15.6 μ s, HCLK=60MHz

则 刷新计数器值= $2^{11} + 1 - 60 \times 15.6 = 1113$

$1113 = 0x459 = 0b10001011001$

5、BANKSIZE---BANK6/7组大小控制寄存器

7	6	5	4	3	2	1	0
BURST_EN	X	SCKE_EN	SCLK_EN	X	BK76MAP		

高24位未用。

BURST_EN: ARM突发操作控制

0: 禁止突发操作; **1:** 可突发操作

SCKE_EN: SCKE使能控制SDRAM省电模式

0: 关闭省电模式; **1:** 使能省电模式

SCLK_EN: SCLK省电控制, 使其只在SDRAM访问周期内使能SCLK

0: SCLK一直有效; **1:** SCLK只在访问期间有效

BK76MAP: 控制BANK6/7的大小及映射

BK76MAP: 控制BANK6/7的大小及映射

100: 2MB;

101: 4MB;

110: 8MB

111: 16MB;

000: 32MB;

001: 64MB

010: 128MB

6、MRSRB6/7---BANK6/7模式设置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						WBL	TM		CL			BT	BL		

WBL：突发写的长度。 0：固定长度；1：保留

TM：测试模式。 00：模式寄存器集；其它保留

CL：列地址反应时间

000：1个时钟；010：2个时钟；

011：3个时钟；其它保留

BT：猝发类型

0：连续；1：保留

BL：猝发时间 000：1个时钟；其它保留

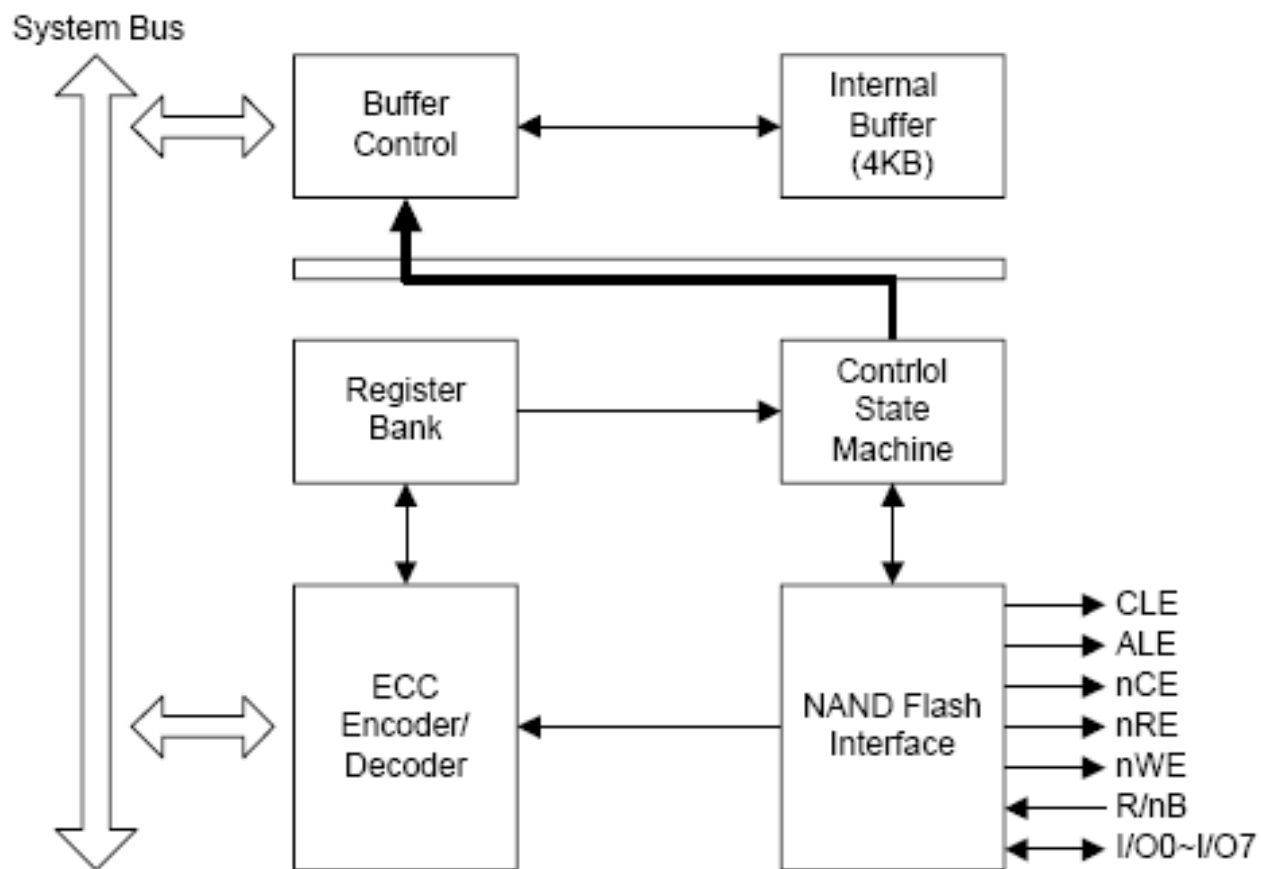
Nand Flash 控制器的寄存器

寄存器	地 址	功 能	操作	复位值
NFCON	0x4E000000	Nand Flash配置	读/写	-
NFCMD	0x4E000004	Nand Flash命令	读/写	-
NFADDR	0x4E000008	Nand Flash地址	读/写	-
NFDATA	0x4E00000C	Nand Flash数据	读/写	-
NFSTAT	0x4E000010	Nand Flash状态	读/写	-
NFECC	0x4E000014	Nand Flash纠错	读/写	-

NAND Flash控制器主要特性

- **Nand Flash模式：**支持读/擦/编程NAND Flash存储器。
- **自动导入模式：**复位后，引导代码被送入Steppingstone，传送后，引导代码在Steppingstone中执行。
- **具有硬件ECC（纠错码）功能：**硬件产生纠错代码。
- **内部4KB的SRAM缓冲器Steppingstone，**在NAND Flash引导后可以作为其他用途使用。

主要由6部分组成



2410的NAND Flash控制器方框图

■ NAND Flash控制器及NAND Flash的寻址

- 1.因为没有统一的接口标准，NAND Flash与微处理器间的接口较复杂。
- 2.NAND Flash接口信号分为三类：数据线、控制线和状态线。
- 3.NAND Flash接口信号比较少，其数据线的宽度只有8位。
- 4.NAND Flash无地址线，另外多出了命令锁存允许信号**CLE**(Command Latch Enable: 写操作时给出此信号表示进行写命令操作)和地址锁存允许信号**ALE**(Address Latch Enable: 写操作时给出此信号表示进行写地址或数据操作)用来区分总线上的数据类型。
- 5.就绪/忙信号**R/nB**(Nand Flash Ready/Busy)用于表示器件的工作状态。当数据写入、编程或随机读取时，R/nB为低电平，表示NAND Flash芯片正忙，否则输出高电平。

■ 2410 NAND Flash概述及NAND Flash的寻址

NAND Flash存储器读速度不如NOR Flash，而擦、写速度高，容量大，价格低。因此，现在不少用户从NAND Flash启动和引导系统，而在SDRAM上执行主程序代码。

2410 NAND Flash概述

(1) **S3C2410微处理器从NAND Flash的引导功能**：其内部有一个叫做“**起步石（Stepping stone）**”的**4KB SRAM缓冲器**。系统启动时，NAND Flash存储器的前面4KB将被自动加载到**起步石**中，然后系统自动执行这些载入的引导代码。引导代码执行完毕后，自动跳转到SDRAM执行。

(2) **NAND Flash操作的校验功能**：使用S3C2410X内部硬件ECC功能可以对Nand flash的数据进行有效性的检测。

■ Nand Flash 控制器

- ❑ S3C2410X支持Nand Flash启动，启动代码存储在Nand Flash上。
- ❑ 启动时，Nand Flash的前4KB将被装载到内部的固定地址中，利用ECC对数据的正确性检验，然后开始执行其中的启动代码。
- ❑ 一般情况下，该启动代码会把Nand Flash中的内容拷贝到SDRAM中去，拷贝完后，主程序将在SDRAM中执行。操作流程如下图所示。

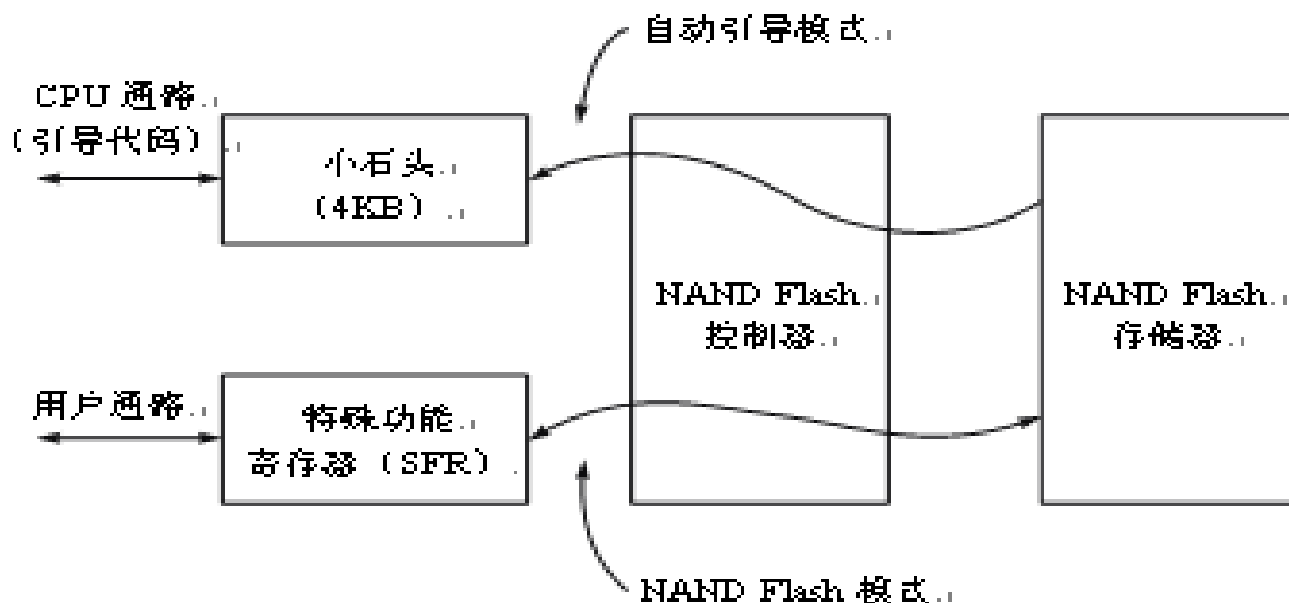


图 4.13 Nand Flash 执行流程

- 自动引导模式流程：
- 复位：
 - 如果自动引导模式使能，Nand Flash中的前4KB代码拷贝到内部的小石头区域；
 - 小石头映射到nGCS0；
 - CPU开始执行小石头区域中的代码。
- Nand Flash模式：
 - 通过NFCONF寄存器设置Nand Flash配置；
 - 把Nand Flash命令写入NFCMD寄存器；
 - 把Nand Flash地址写入NFADDR寄存器；
 - 读/写数据同时通过NFSTAT寄存器检测Nand Flash状态。读操作前或者程序执行后检查R/nB信号。

1、NFCON---Flash配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFEN	X		IECC	NFCE	TACLE			X	TWRPH0			X	PWRPH1		
0	-		0	0	0			-	0			-	0		

NFEN: NF控制器使能控制

0: 禁止使用;

1: 允许使用

IECC: 初始化ECC编码/解码器控制位

0: 不初始化ECC;

1: 初始化ECC

NFCE: NF片选信号nFCE控制位持续时间设置

0: nFCE为低有效;

0: nFCE为高无效

TACLE: CLE/ALE持续时间设置值 (0---7)

持续时间 = HCLK * (TACLS + 1)

CLE/ALE : 命令/地址锁存允许

TWRPH0: 写信号持续时间设置值 (0~7)

持续时间 = $HCLK * (TWRPH0 + 1)$

TWRPH1: 写信号无效后CLE/ALE保持时间设置值 (0~7)

持续时间 = $HCLK * (TWRPH1 + 1)$

2、NFCMD---Flash命令寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留								命令字							

3、NFADDR---Flash地址寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留								地址值							

高24位未用，低8位为Flash存储器地址值

4、NFDATA---Flash数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留								输入/输出数据							

高24位未用，低8位为读入或者写出的数据

5、NFSTAT---Flash状态寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留															RnB

RnB: Nand Flash存储器状态位

0: 存储器忙;

1: 存储器准备好

6、NFEC---Flash错误校正码寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保 留								错误校正码#2							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
错误校正码#1								错误校正码#0							

S3C2410启动方式

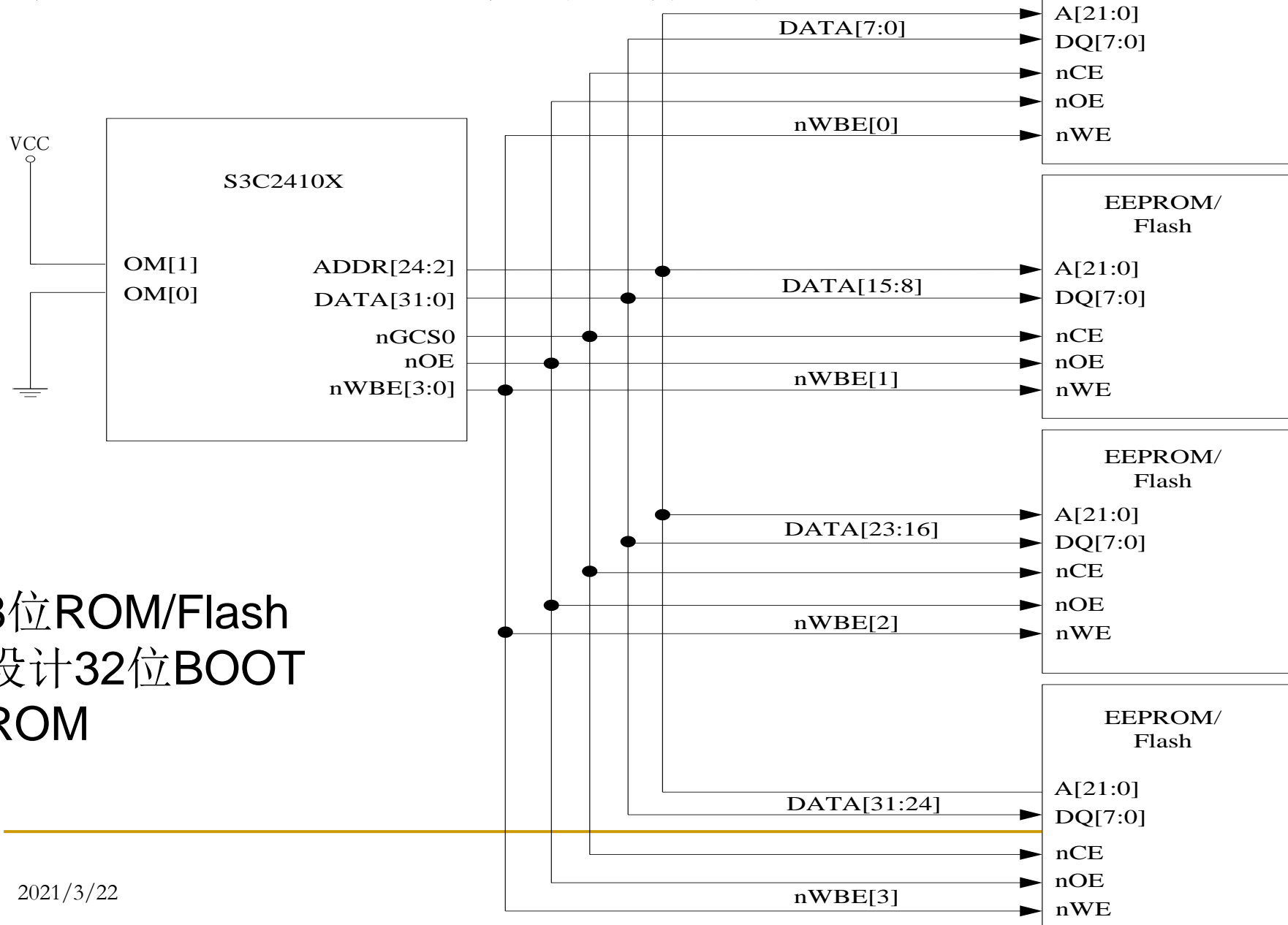
- 两种启动方式:
- 非NAND Flash启动方式, S3C2410访问0X0000 0000地址, 因此, 启动代码应该放在0X0000 0000地址上, BOOT ROM的总线宽度可以由OM[1:0]确定。
- NAND Flash启动方式, 此时, CPU将从NAND Flash中 (Stepping Stone) 读取代码来启动。

S3C2410启动方式

- Bank0 存储块可以外接SRAM类型的存储器或者具有SRAM接口特性的ROM存储器 (如NOR Flash)，其数据总线宽度应设定为16位或32位中的一种。
- 当0号存储块作为ROM区，完成引导装入工作时(从0x00000000启动)，Bank0存储块的总线宽度应在第一次访问ROM前根据OM1、OM0在复位时的逻辑组合来确定

OM1	OM0	引导ROM数据的宽度
0	0	NANDFlash模式
0	1	16位
1	0	32位
1	1	测试模式

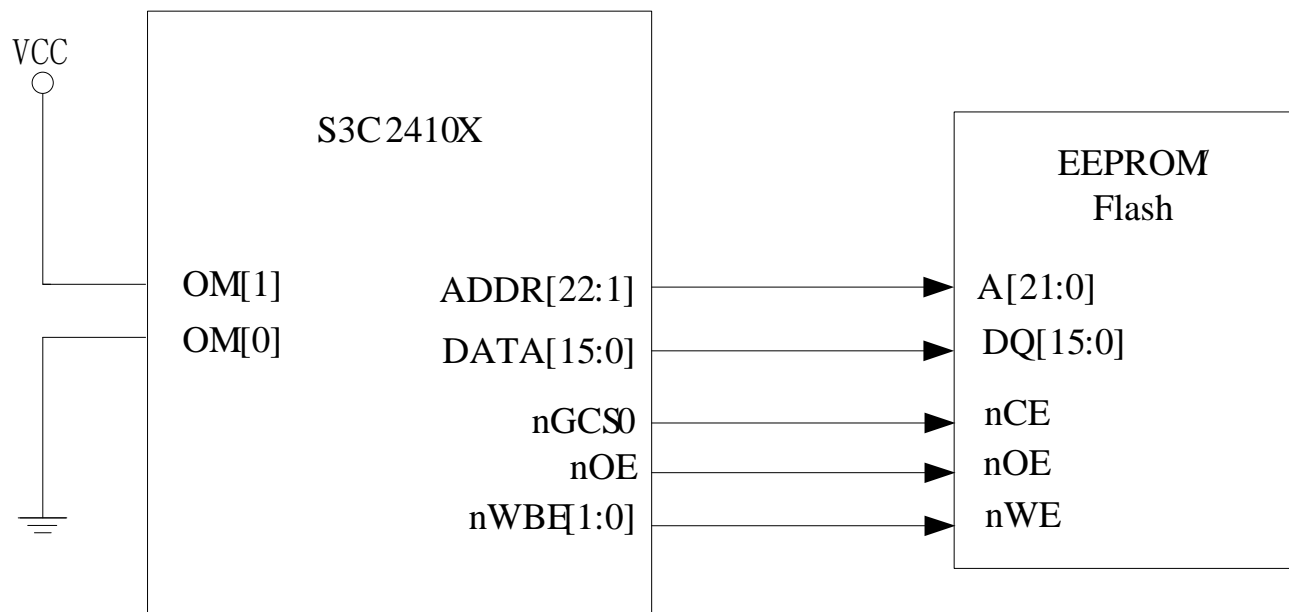
非NAND Flash启动方式设计



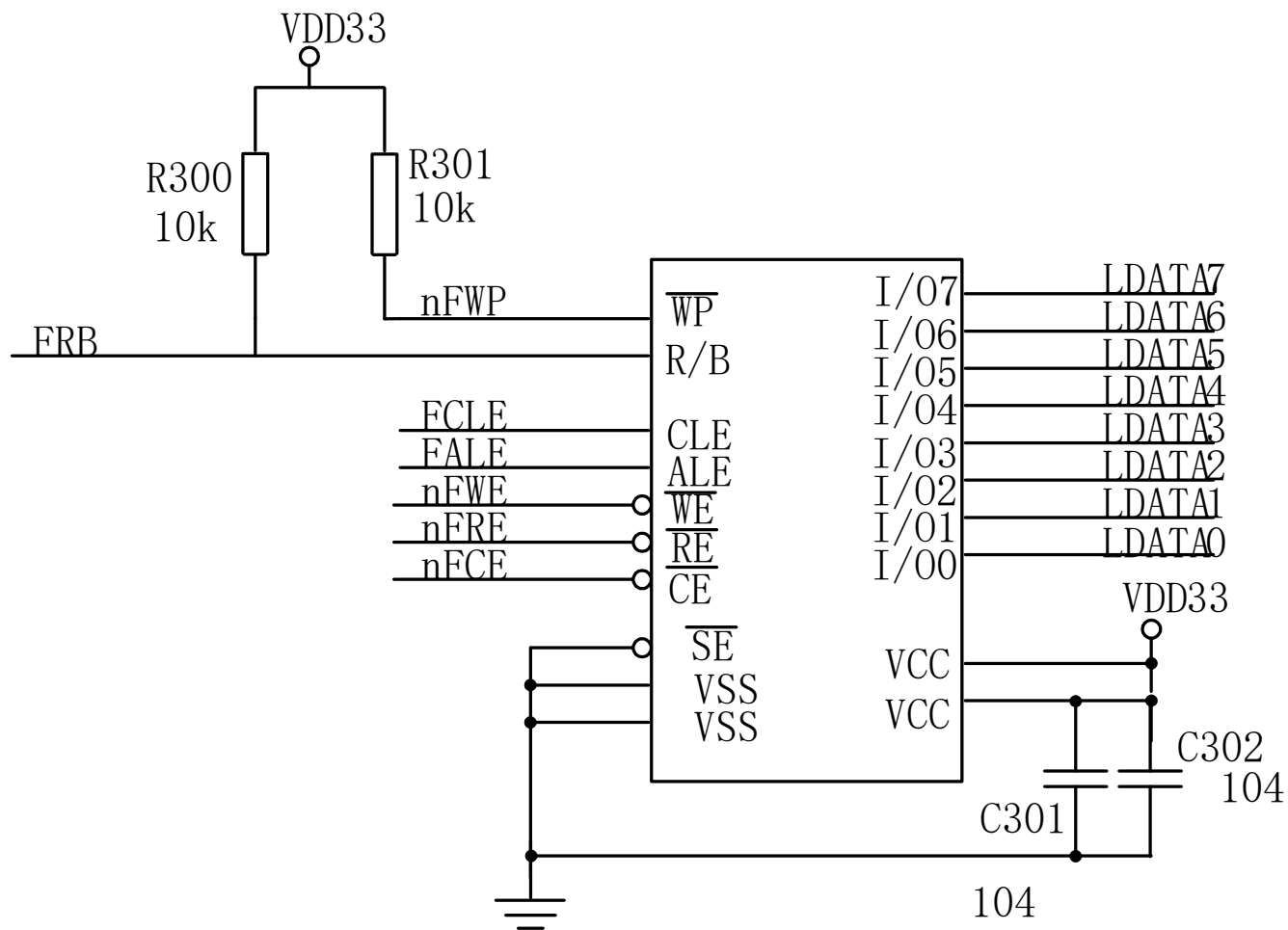
8位ROM/Flash
设计32位BOOT
ROM

非NAND Flash启动方式设计

用16位ROM设计16位BOOT ROM

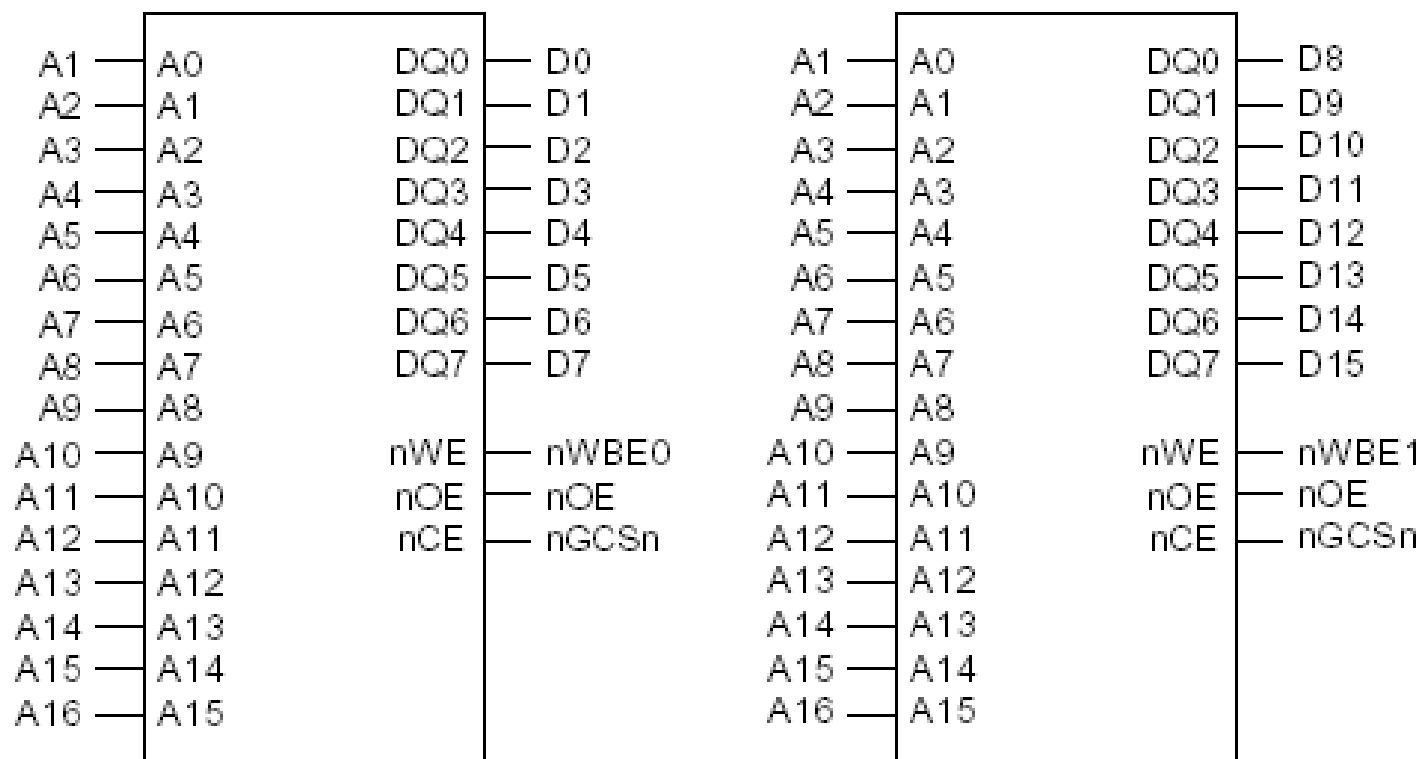


NAND Flash启动方式



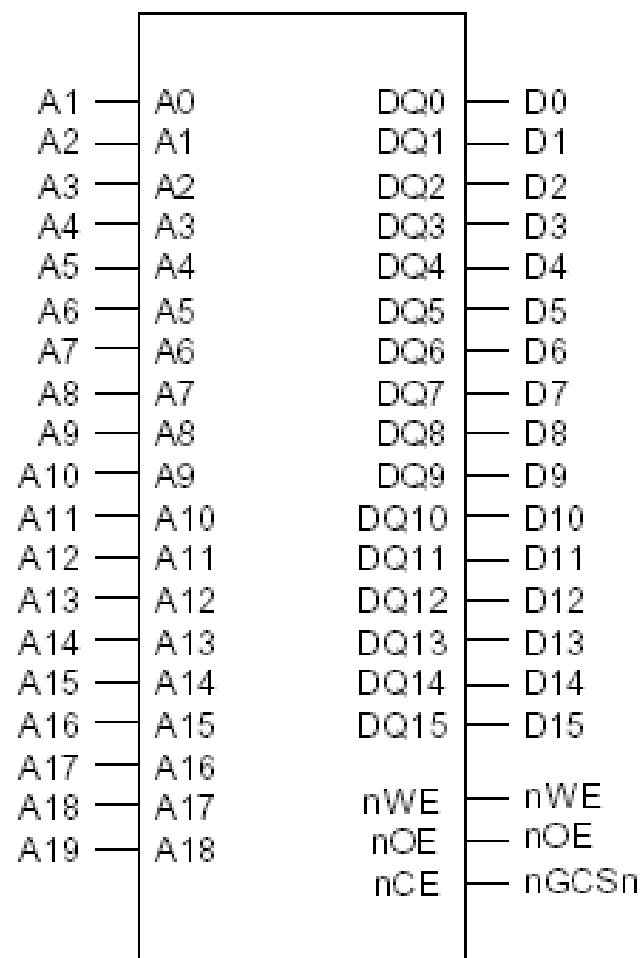
5.3.2 S3C2410存储器接口设计

■ 与2片8位的ROM连接方法



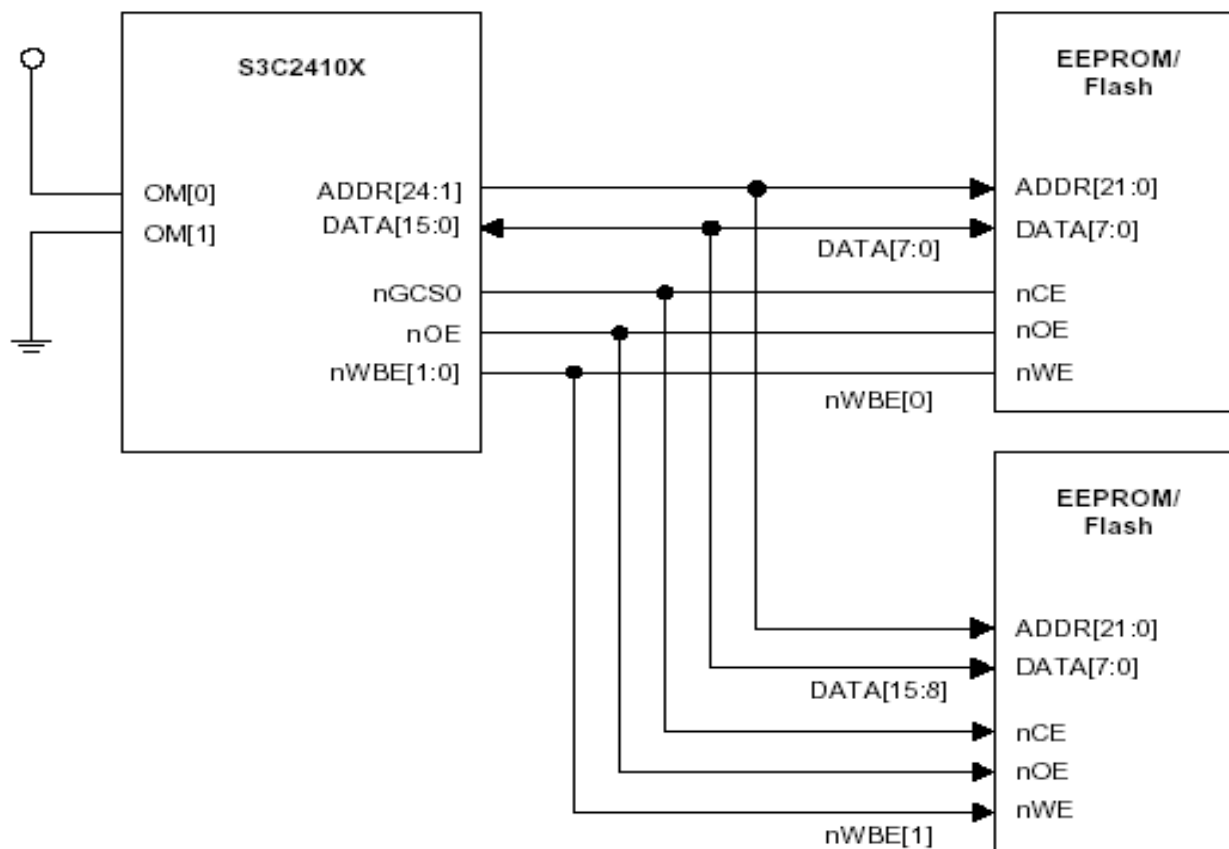
S3C2410存储器接口设计

■ 与1片16位的ROM连接



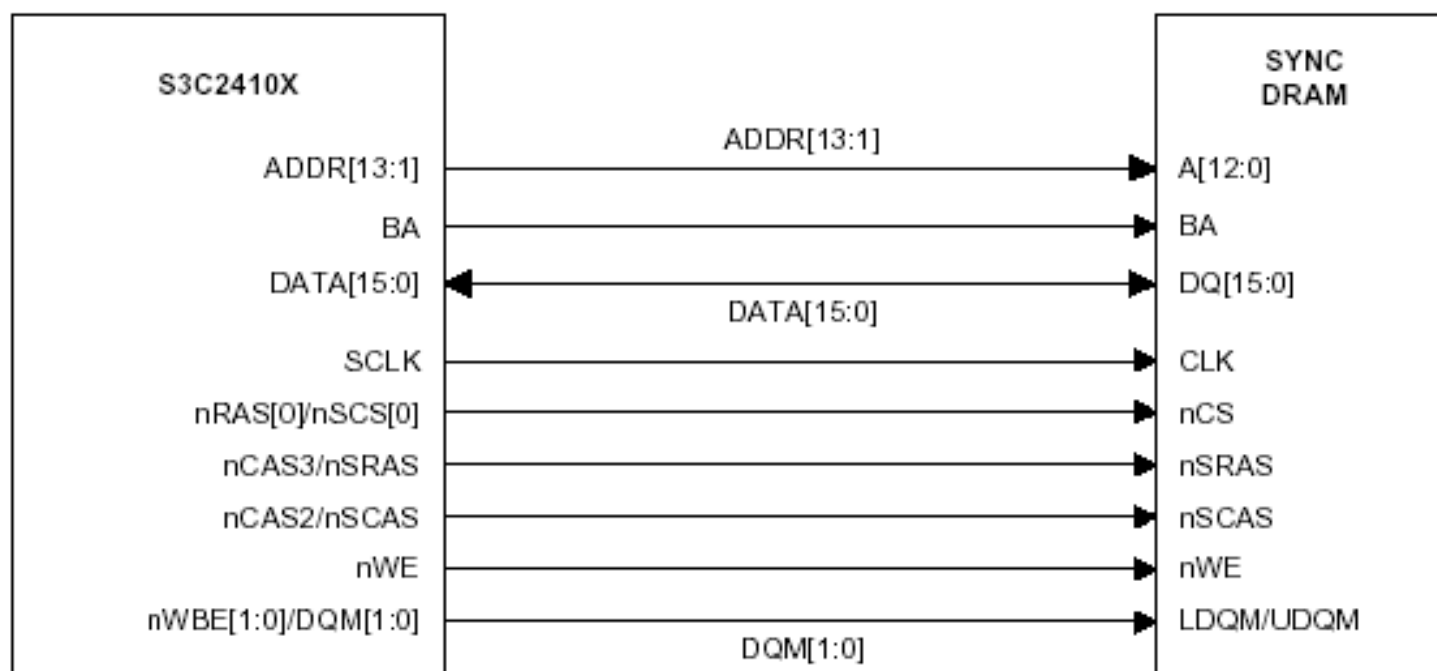
S3C2410存储器接口设计

■ 与2片8位FLASH的连接方法



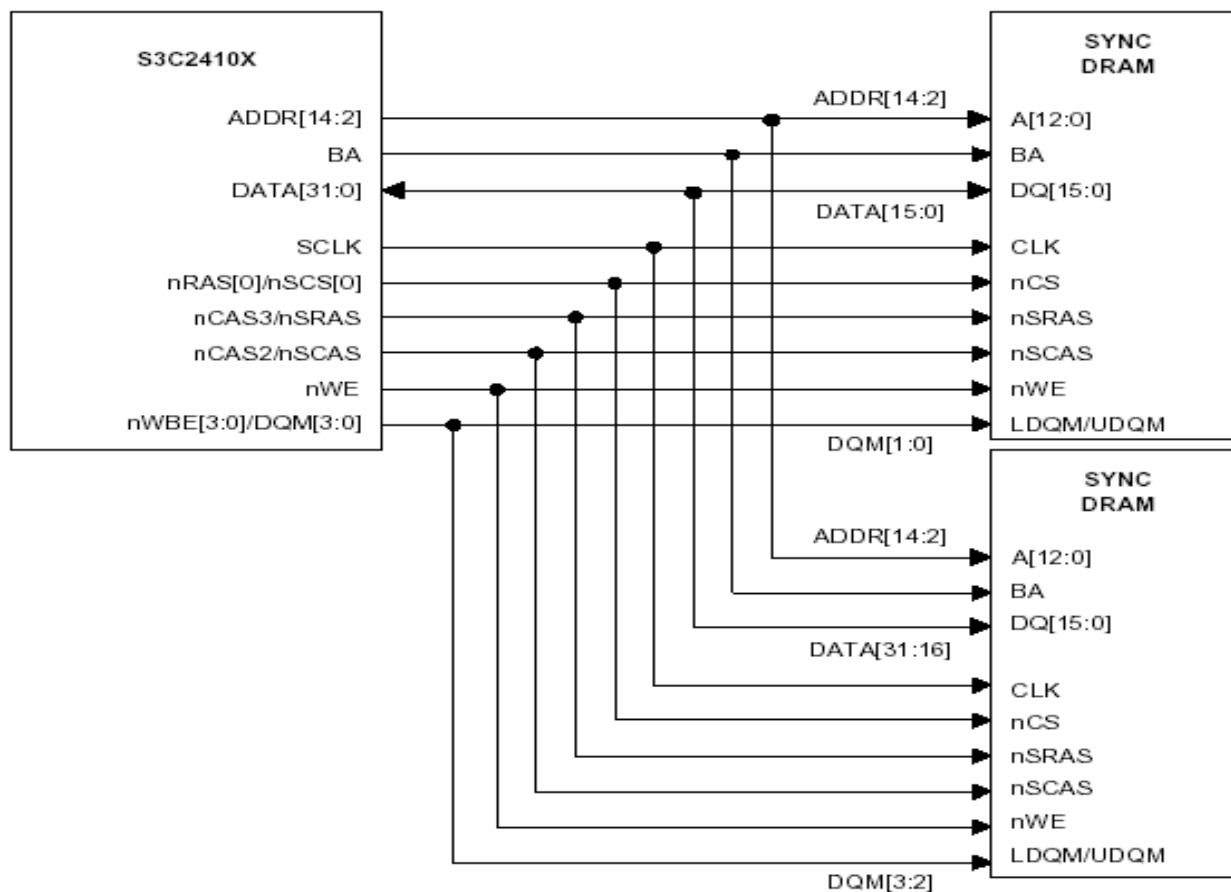
S3C2410存储器接口设计

■ 与1片16M的SDRAM的连接方法



S3C2410存储器接口设计

■ 与2片16M的SDRAM的连接方法



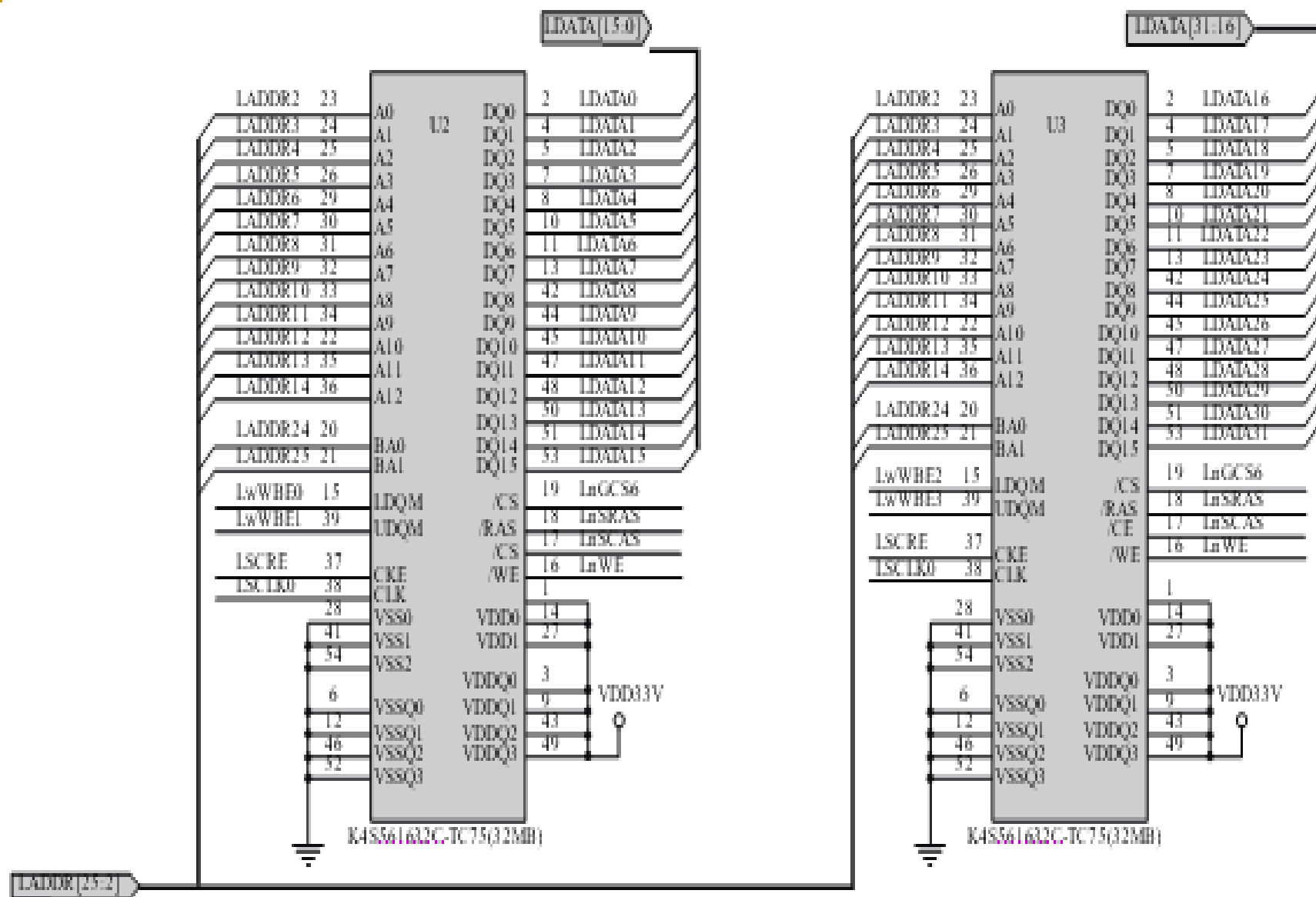


图 4.29 32 位 SDRAM 存储器系统的实际应用电路图

上图是存储容量为64MB的SDRAM接口电路图。注意：芯片内部的4 bank不是指该芯片需要占用S3C2410芯片的4个存储块，而是指HY57V561620芯片内部把32MB容量分成了4块存储区，每块存储区的容量为4M×16 bit。从图中可以看出，该存储器的nCS由S3C2410芯片的nGCS6引脚控制，因此，它占用了S3C2410芯片的6号存储块。实际占用地址空间为0x30000000 ~ 0x33FFFFFF。

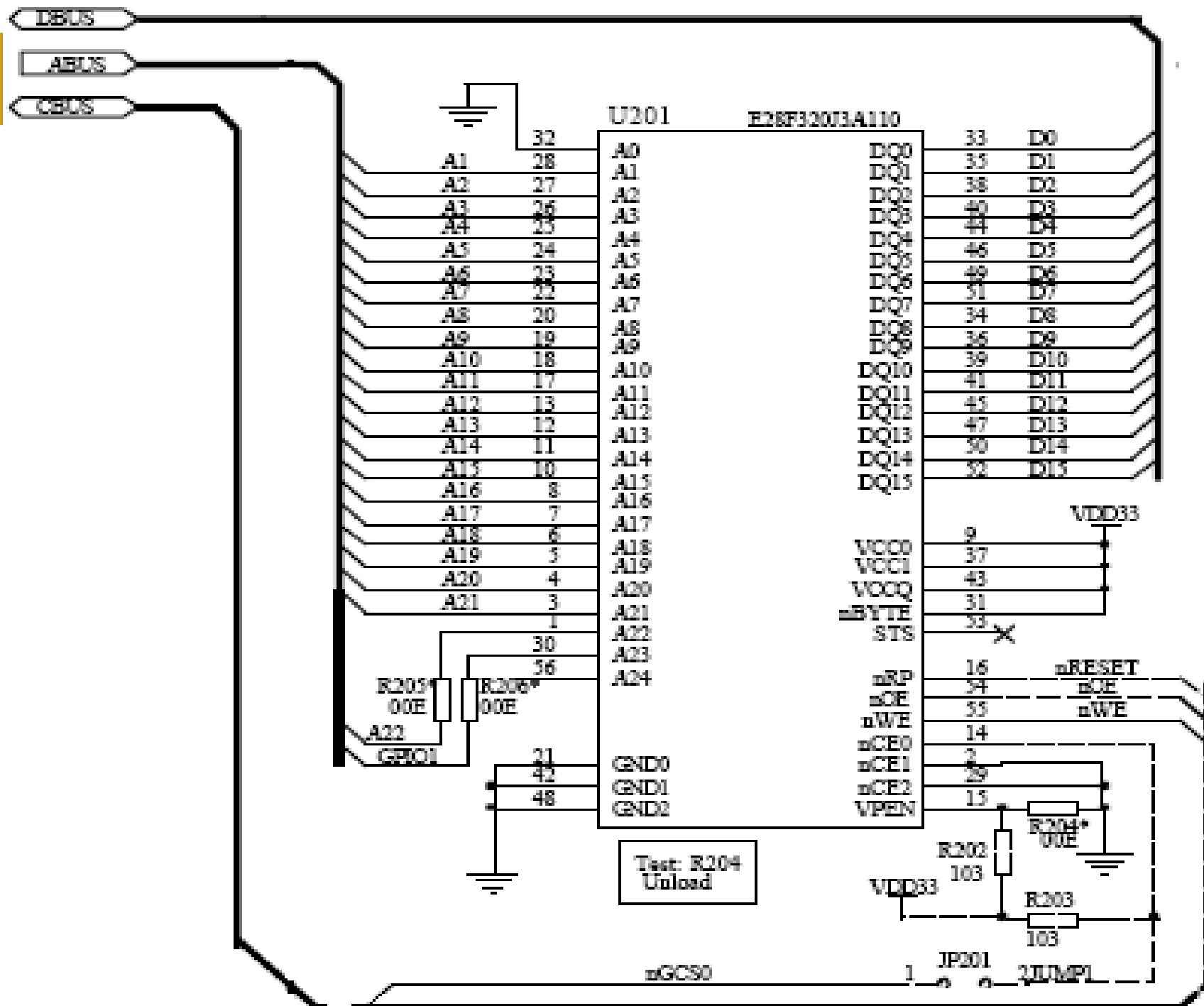
由上图可知，这里应将BWSCON中的DW6设置为10，即选择32位总线方式。

表：HY57V561620系列引脚信号描述

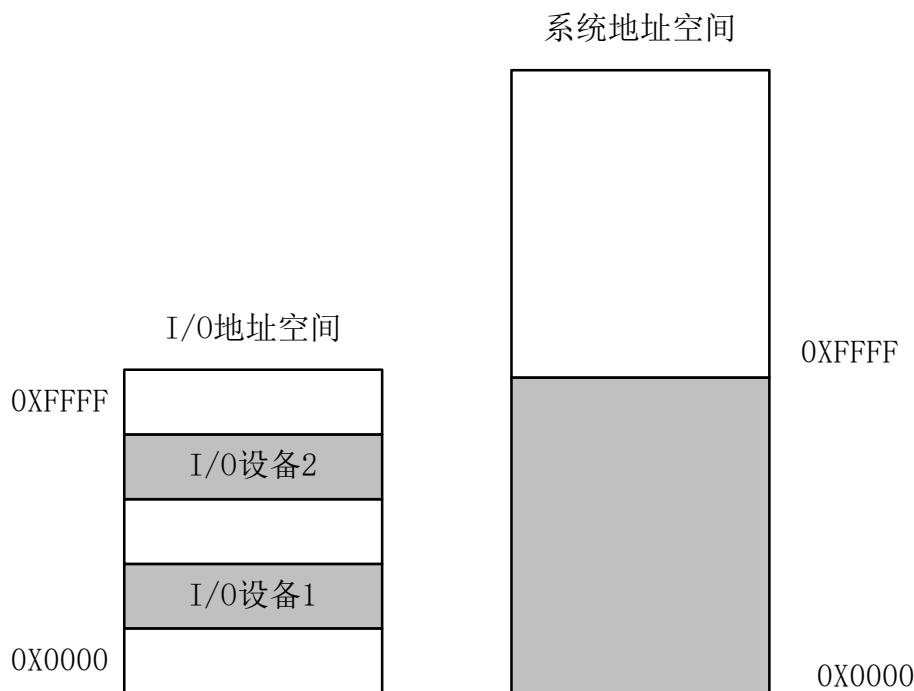
引脚	名称	描述
CLK	时钟	芯片时钟输入
CKE	时钟使能	片内时钟信号控制
nCS	片选	禁止或使能CLK、CKE和DQM外的所有输入信号
BA0, BA1	块地址选择	用于片内4个块的选择
A12~A0	地址总线	行地址为A12~A0， 列地址为A8~A0
nRAS	行地址锁存	时钟沿和nRAS有效时锁存行地址，允许行的访问或改写
nCAS	列地址锁存	时钟沿和nCAS有效时锁存列地址，允许列的访问
nWE	写使能	使能写信号和允许列改写，nWE和nCS有效时开始锁存数据
LDQM, UDQM	数据I/O屏蔽	在读模式下控制输出缓冲；在写模式下屏蔽输入数据
DQ15~DQ0	数据总线	数据输入输出引脚

NOR Flash存储器接口设计

- NOR Flash存储器芯片的接口特性类似于SRAM，与微处理器的接口电路相对于SDRAM类、NAND Flash类存储器的接口来说简单。
- 下面的图是以32MB容量的NOR Flash芯片E28F320J3为例的存储器接口电路。
- E28F320J3是16位的数据宽度，地址是半对准的，因此该芯片的地址线A0接地。此时，应将BWSCON中的DW0设置为01，即选择16位总线方式。



I/O接口的编址方式——端口映射



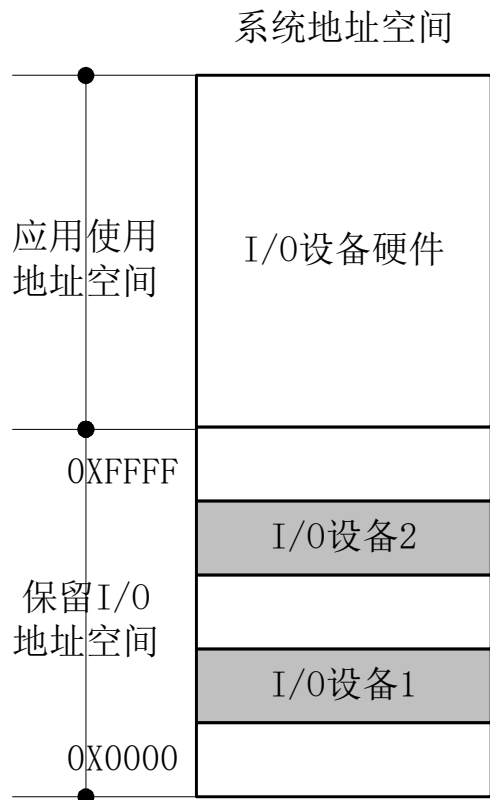
1) I/O接口独立编址——端口映射方式

- 这种编址方式是将存储器地址空间和I/O接口地址空间分开设置，互不影响。设有专门的输入指令（IN）和输出指令（OUT）来完成I/O操作。

主要优点：内存地址空间与I/O接口地址空间分开，互不影响，译码电路较简单，并设有专门的I/O指令，所以编程序易于区分，且执行时间短，快速性好。

缺点：只用I/O指令访问I/O端口，功能有限且要采用专用I/O周期和专用I/O控制线，使微处理器复杂化。

I/O接口的编址方式——内存映射



1) 2) I/O接口与存储器统一编址方式——内存映射

- 所有的I/O接口的端口都当作是存储器的一个单元对待
- 不设专门的输入/输出指令，所有传送和访问存储器的指令都可用来对I/O接口操作。

主要优点：访问内存的指令都可用于I/O操作，数据处理功能强；同时I/O接口可与存储器部分共用译码和控制电路。

缺点：一是I/O接口要占用存储器地址空间的一部分；二是因不用专门的I/O指令，程序中较难区分I/O操作。

5.4 S3C2410 I/O端口——内存映射

- S3C2410有117个有复用功能的I/O端口引脚：
- PortA (GPA) 23个输出端口；
- PortB (GPB) 11个I/O端口；
- PortC (GPC) 16个I/O端口；
- PortD (GPD) 16个I/O端口；
- PortE (GPE) 16个I/O端口；
- PortF (GPF) 8个I/O端口；
- PortG (GPG) 16个I/O端口；
- PortH (GPH) 11个I/O端口；

端口寄存器及引脚配置

每一个端口都有4个寄存器，它们是：引脚配置寄存器、数据寄存器、引脚上拉寄存器等。

Register	Address	R/W	Description	Reset Value
GPXCON	0x560000x0	R/W	端口X配置寄存器	X
GPXDAT	0x560000x4	R/W	端口X数据寄存器	X
GPXUP	0x560000x8	R/W	端口X上拉寄存器	X
RESERVED	0x560000xC	R/W	端口X保留寄存器	-

1、端口A寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPACON	0x56000000	R/W	端口A引脚配置寄存器	0x7FFFFFFF
GPADAT	0x56000004	R/W	端口A数据寄存器	-
RESERVED	0x56000008	-	端口A保留寄存器	-
RESERVED	0x5600000C	-	端口A保留寄存器	-

GPADAT寄存器为准备输出的数据

其值为23位[22: 0]

注意：（1）当A口引脚配置为非输出功能时，其输出无意义；（2）从引脚输入没有意义。

1、端口A寄存器及引脚配置

位号	位 名	位值:0	1	位号	位 名	位值:0	1
22	GPA22	输出	nFCE	10	GPA10	输出	ADDR25
21	GPA21	输出	nRSTOUT	9	GPA9	输出	ADDR24
20	GPA20	输出	nFRE	8	GPA8	输出	ADDR23
19	GPA19	输出	nFWE	7	GPA7	输出	ADDR22
18	GPA18	输出	ALE	6	GPA6	输出	ADDR21
17	GPA17	输出	CLE	5	GPA5	输出	ADDR20
16	GPA16	输出	nGCS5	4	GPA4	输出	ADDR19
15	GPA15	输出	nGCS4	3	GPA3	输出	ADDR18
14	GPA14	输出	nGCS3	2	GPA2	输出	ADDR17
13	GPA13	输出	nGCS2	1	GPA1	输出	ADDR16
12	GPA12	输出	nGCS1	0	GPA0	输出	ADDR0
11	GPA11	输出	ADDR26		FCE:	Flash片选	

2、端口B寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPBCON	0x56000010	R/W	端口B引脚配置寄存器	0x0
GPBDAT	0x56000014	R/W	端口B数据寄存器	-
GPBUP	0x56000018	R/W	端口B上拉寄存器	0x0
RESERVED	0x5600001C	-	端口B保留寄存器	-

GPBDAT---为准备输出或输入的数据

其值为11位[10: 0]

GPBUP---端口B上拉寄存器，位[10: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

注意： 当B口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口B引脚配置寄存器

位号	位 名	位值:00	01	10	11
21,20	GPB10	输入	输出	nXDREQ0	Reserved
19,18	GPB9	输入	输出	nXDACK0	Reserved
17,16	GPB8	输入	输出	nXDREQ1	Reserved
15,14	GPB7	输入	输出	nXDACK1	Reserved
13,12	GPB6	输入	输出	nXBACK	Reserved
11,10	GPB5	输入	输出	nXBREQ	Reserved
9,8	GPB4	输入	输出	TCLK0	Reserved
7,6	GPB3	输入	输出	TOUT3	Reserved
5,4	GPB2	输入	输出	TOUT2	Reserved
3,2	GPB1	输入	输出	TOUT1	Reserved
1,0	GPB0	输入	输出	TOUT0	Reserved

3、端口C寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPCCON	0x56000020	R/W	端口C引脚配置寄存器	0x0
GPCDAT	0x56000024	R/W	端口C数据寄存器	-
GPCUP	0x56000028	R/W	端口C上拉寄存器	0x0
RESERVED	0x5600002C	-	端口C保留寄存器	-

GPCDAT---为准备输出或输入的数据

其值为16位[15: 0]

GPCUP---端口C上拉寄存器，位[15: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

注意： 当C口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口C引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPC15	输入	输出	VD7	保留	15,14	GPC7	输入	输出	LCDVF2	保留
29,28	GPC14	输入	输出	VD6	保留	13,12	GPC6	输入	输出	LCDVF1	保留
27,26	GPC13	输入	输出	VD5	保留	11,10	GPC5	输入	输出	LCDVF0	保留
25,24	GPC12	输入	输出	VD4	保留	9,8	GPC4	输入	输出	VM	保留
23,22	GPC11	输入	输出	VD3	保留	7,6	GPC3	输入	输出	VFRAME	保留
21,20	GPC10	输入	输出	VD2	保留	5,4	GPC2	输入	输出	VLINE	保留
19,18	GPC9	输入	输出	VD1	保留	3,2	GPC1	输入	输出	VCLK	保留
17,16	GPC8	输入	输出	VD0	保留	1,0	GPC0	输入	输出	VEND	保留

4、端口D寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPDCON	0x56000030	R/W	端口D引脚配置寄存器	0x0
GPDDAT	0x56000034	R/W	端口D数据寄存器	-
GPDUP	0x56000038	R/W	端口D上拉寄存器	0xF000
RESERVED	0x5600003C	-	端口D保留寄存器	-

GPDDAT---为准备输出或输入的数据

其值为16位[15: 0]

GPDUP---端口D上拉寄存器，位[15: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

初始化时，[15: 12]无上拉功能，而[11: 0]有上拉

注意： 当D口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口D引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPD15	输入	输出	VD23	nSS0	15,14	GPD7	输入	输出	VD15	保留
29,28	GPD14	输入	输出	VD22	nSS1	13,12	GPD6	输入	输出	VD14	保留
27,26	GPD13	输入	输出	VD21	保留	11,10	GPD5	输入	输出	VD13	保留
25,24	GPD12	输入	输出	VD20	保留	9,8	GPD4	输入	输出	VD12	保留
23,22	GPD11	输入	输出	VD19	保留	7,6	GPD3	输入	输出	VD11	保留
21,20	GPD10	输入	输出	VD18	保留	5,4	GPD2	输入	输出	VD10	保留
19,18	GPD9	输入	输出	VD17	保留	3,2	GPD1	输入	输出	VD9	保留
17,16	GPD8	输入	输出	VD16	保留	1,0	GPD0	输入	输出	VD8	保留

5、端口E寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPECON	0x56000040	R/W	端口E引脚配置寄存器	0x0
GPEDAT	0x56000044	R/W	端口E数据寄存器	-
GPEUP	0x56000048	R/W	端口E上拉寄存器	0x0
RESERVED	0x5600004C	-	端口E保留寄存器	-

GPEDAT---为准备输出或输入的数据

其值为16位[15: 0]

GPEUP---端口E上拉寄存器，位[15: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

初始化时，各个引脚都有上拉功能。

注意： 当E口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口E引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPE15	输入	输出	IICSDA	保留	15,14	GPE7	输入	输出	SDDAT0	保留
29,28	GPE14	输入	输出	IICSCL	保留	13,12	GPE6	输入	输出	SDCMD	保留
27,26	GPE13	输入	输出	SPICLK0	保留	11,10	GPE5	输入	输出	SDCLK	保留
25,24	GPE12	输入	输出	SPISIO	保留	9,8	GPE4	输入	输出	ISSDO	保留
23,22	GPE11	输入	输出	SPISO0	保留	7,6	GPE3	输入	输出	ISSDI	保留
21,20	GPE10	输入	输出	SDDAT3	保留	5,4	GPE2	输入	输出	CDCLK	保留
19,18	GPE9	输入	输出	SDDAT2	保留	3,2	GPE1	输入	输出	ISSCLK	保留
17,16	GPE8	输入	输出	SDDAT1	保留	1,0	GPE0	输入	输出	ISLRCK	保留

6、端口F寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPFCON	0x56000050	R/W	端口F引脚配置寄存器	0x0
GPFDAT	0x56000054	R/W	端口F数据寄存器	-
GPFUP	0x56000058	R/W	端口F上拉寄存器	0x0
RESERVED	0x5600005C	-	端口F保留寄存器	-

GPFDAT---为准备输出或输入的数据

其值为8位[7: 0]

GPFUP---端口F上拉寄存器，位[7: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

初始化时，各个引脚都有上拉功能。

注意： 当F口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口F引脚配置寄存器

位号	位 名	位 值			
		00	01	10	11
15,14	GPF7	输入	输出	EINT7	保留
13,12	GPF6	输入	输出	EINT6	保留
11,10	GPF5	输入	输出	EINT5	保留
9,8	GPF4	输入	输出	EINT4	保留
7,6	GPF3	输入	输出	EINT3	保留
5,4	GPF2	输入	输出	EINT2	保留
3,2	GPF1	输入	输出	EINT1	保留
1,0	GPF0	输入	输出	EINT0	保留

7、端口G寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPGCON	0x56000060	R/W	端口G引脚配置寄存器	0x0
GPGDAT	0x56000064	R/W	端口G数据寄存器	-
GPGUP	0x56000068	R/W	端口G上拉寄存器	0xF800
RESERVED	0x5600006C	-	端口G保留寄存器	-

GPGDAT---为准备输出或输入的数据

其值为16位[15: 0]

GPGUP---端口G上拉寄存器，位[15: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

初始化时，[15:11]引脚无上拉功能，其它引脚有。

注意： 当G口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口G引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPG15	输入	输出	EINT23	nYPON	15,14	GPG7	输入	输出	EINT15	SPICLK1
29,28	GPG14	输入	输出	EINT22	YMON	13,12	GPG6	输入	输出	EINT14	SPISI1
27,26	GPG13	输入	输出	EINT21	nXPON	11,10	GPG5	输入	输出	EINT13	SPISO1
25,24	GPG12	输入	输出	EINT20	XMON	9,8	GPG4	输入	输出	EINT12	LCD- PEN
23,22	GPG11	输入	输出	EINT19	TCLK1	7,6	GPG3	输入	输出	EINT11	nSS1
21,20	GPG10	输入	输出	EINT18	保留	5,4	GPG2	输入	输出	EINT10	nSS0
19,18	GPG9	输入	输出	EINT17	保留	3,2	GPG1	输入	输出	EINT9	保留
17,16	GPG8	输入	输出	EINT16	保留	1,0	GPG0	输入	输出	EINT8	保留

LCD-PEN:POWER_ENABLE

nSS0:SPI0_SELECT

8、端口H寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPHCON	0x56000070	R/W	端口H引脚配置寄存器	0x0
GPHDAT	0x56000074	R/W	端口H数据寄存器	-
GPHUP	0x56000078	R/W	端口H上拉寄存器	0x0
RESERVED	0x5600007C	-	端口H保留寄存器	-

GPHDAT---为准备输出或输入的数据

其值为11位[10: 0]

GPHUP---端口H上拉寄存器，位[10: 0]有意义。

0: 对应引脚设置为上拉

1: 无上拉功能

注意： 当H口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

端口H引脚配置寄存器

位号	位 名	位值:00	01	10	11
21,20	GPH10	输入	输出	CLKOUT1	Reserved
19,18	GPH9	输入	输出	CLKOUT0	Reserved
17,16	GPH8	输入	输出	UCLK	Reserved
15,14	GPH7	输入	输出	RXD2	nCTS1
13,12	GPH6	输入	输出	TXD2	nRTS1
11,10	GPH5	输入	输出	RXD1	Reserved
9,8	GPH4	输入	输出	TXD1	Reserved
7,6	GPH3	输入	输出	RXD0	Reserved
5,4	GPH2	输入	输出	TXD0	Reserved
3,2	GPH1	输入	输出	nRTS0	Reserved
1,0	GPH0	输入	输出	nCTS0	Reserved

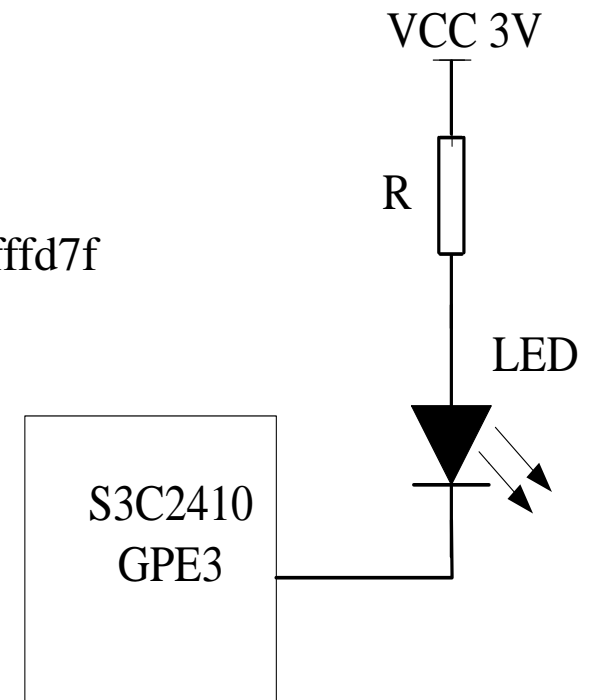
9、端口其它控制寄存器

Register	Address	R/W	Description	Reset Value
MISCCR	0x56000080	R/W	混合控制寄存器	0x10330
DCLKCON	0x56000084	R/W	D时钟控制寄存器	0x0

I/O端口应用举例

```
#include "def.h"
#include "2410lib.h"
#include "isr.h"
#define ledlight()    { rGPEDAT = rGPEDAT & 0xff7}
#define ledclear()   { rGPEDAT = rGPEDAT | 0x0008}

void Main(void)
{
    BoardInitStart(); //系统初始化, MMU初始化
    SystemClockInit(); //系统时钟初始化
    rGPECON = ( ( rGPECON | 0x00000140 ) & 0xffffd7f
//GPIO E初始化
    while (1)
    {
        ledlight ( ) ;           //LED灯亮
        Delay (3000) ;           // 延时
        ledclear ( ) ;           //LED灯灭
    }
}
```



例2:

有3个LED发光二极管，LED1~LED3由2410的GPC口GPC5~GPC7控制。试设计使LED1~LED3间断闪亮的汇编程序。

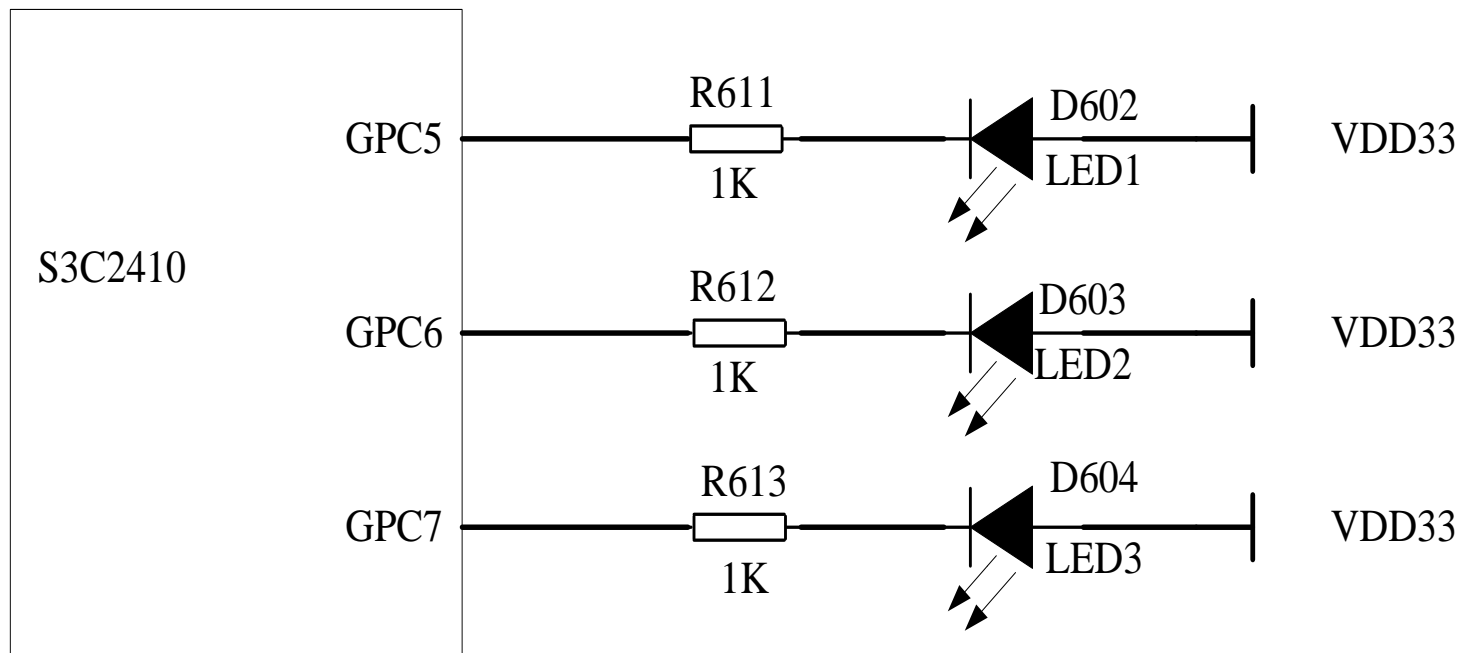


图 S3C2410 GPC7-GPC5引脚信号与LED的连接

本例汇编源程序：

;Filename: test2.s

AREA ABC, CODE, READONLY

ENTRY

CODE32

START	LDR R0, =0x56000020	;GPCCON: Port C control
	LDR R1, =0x56000024	;GPCDAT: Port C data
	LDR R2, =0x56000028	;GPCUP: Port C pull up control
	LDR R5, =0x5454	;GPC口控制字,设置GPC7~GPC0为输出
	STR R5, [R0]	
	MOV R5, #0xffff	;GPC口上拉电阻禁止
	STR R5, [R2]	
LOOP	MOV R5, #0x1F	;GPC7~GPC5输出0
	STR R5, [R1]	
	BL DELAY	;调用DELAY延时子程序
	MOV R5, #0xFF	;GPC7~GPC5输出1
	STR R5, [R1]	
	BL DELAY	;调用延时子程序
	B LOOP	

DELAY

;延时子程序开始

MOV R4, #100

DEL MOV R5, #2000

DEL1 NOP

NOP

NOP

NOP

SUBS R5, R5, #1

BNE DEL1

SUBS R4, R4, #1

BNE DEL

MOV R15, R14 ;延时子程序返回

;上面的返回指令也可写成MOV PC,LR

END