



暨南大学  
JINAN UNIVERSITY

# 第6章 嵌入式系统硬件 (2)

杨光华

物联网与物流工程研究院 / 电气信息学院

办公室：行政楼 631

电邮：ghyang@jnu.edu.cn

电话：8505687

声明：课件中的部分文字、图片、视频等源于网络，相应版权属于原创人

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

## 6 S3C2410X的存储器

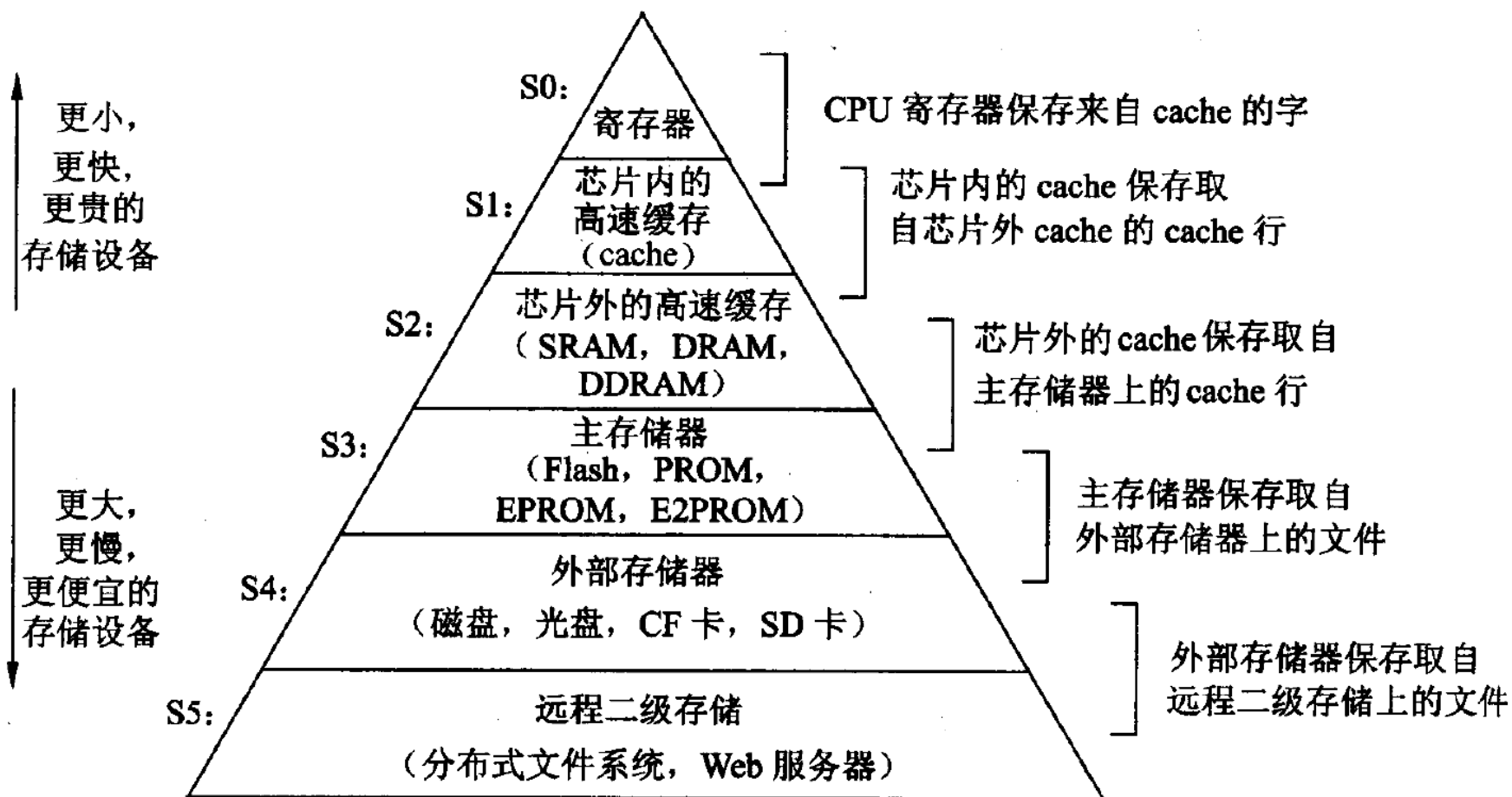
---

### 主要内容

- 存储器类型与选择
- 存储器配置
- 存储器模块硬件设计及控制
- Nand Flash及控制器
- 存储器模块驱动程序设计

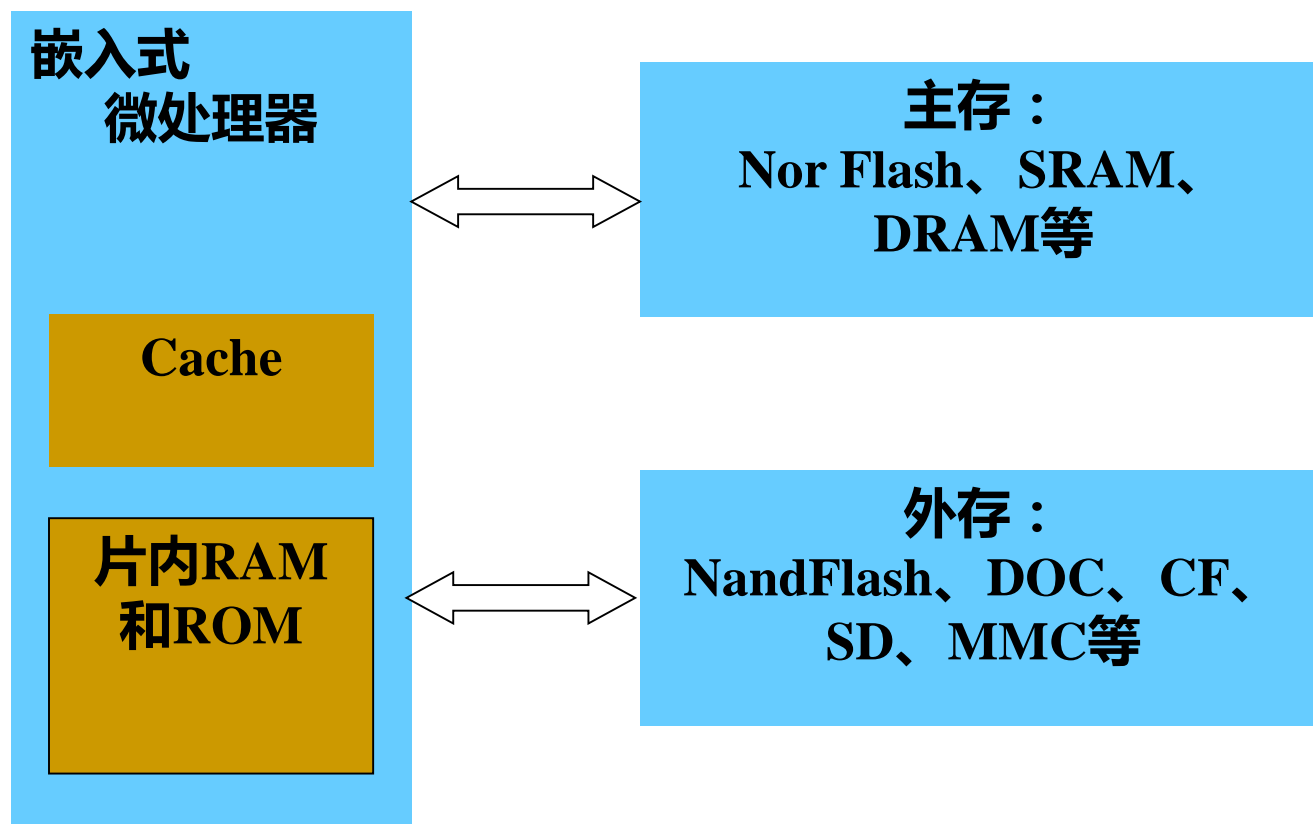
# 存储器系统的层次结构

## 存储器系统的层次结构



# 嵌入式系统存储设备分类

- 嵌入式系统典型的存储结构



# 嵌入式系统存储设备分类

- 主存

- 主存是处理器能直接访问的存储器，用来存放系统和用户的程序和数据
- 嵌入式系统的主存可位于SoC内和SoC外，片内存储器存储容量小、速度快，片外存储器容量大
- 可以做主存的存储器有：
  - RAM类：SRAM、DRAM、SDRAM等
  - ROM类：Nor Flash、EPROM、E2PROM、PROM等

- 外存

- 外存储器也称辅助存储器，简称外存或辅存
- 外存主要指那些容量比主存大、读取速度较慢、通常用来存放需要永久保存的或相对来说暂时不用的各种程序 and 数据的存储器

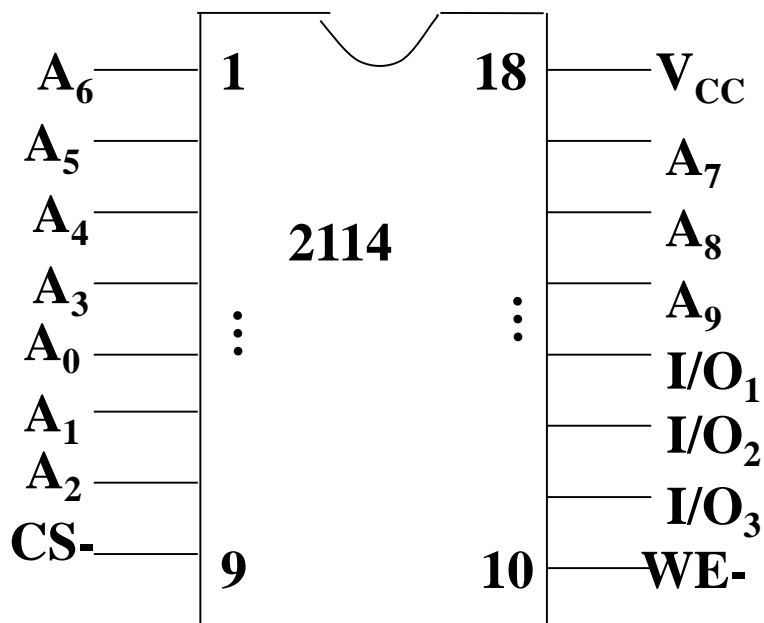
# RAM的种类与选型

- 常见RAM的种类有：

- SRAM (Static RAM, 静态随机存储器)
- DRAM (Dynamic RAM, 动态随机存储器)
- SDRAM (Synchronous DRAM, 同步动态随机存储器)
- DDRAM (Double Data Rate SDRAM, 双倍速率随机存储器)

# RAM的种类与选型 - SRAM

- 静态随机存取存储器（SRAM）
  - 存储信息：六管存储电路
  - 与DRAM比较，SRAM的存取速度快，但制造成本高
  - 典型芯片规格：2114（1KX4） 6116（2KX8） 6264（8KX8） 62128（16KX8） 62256（32KX8）



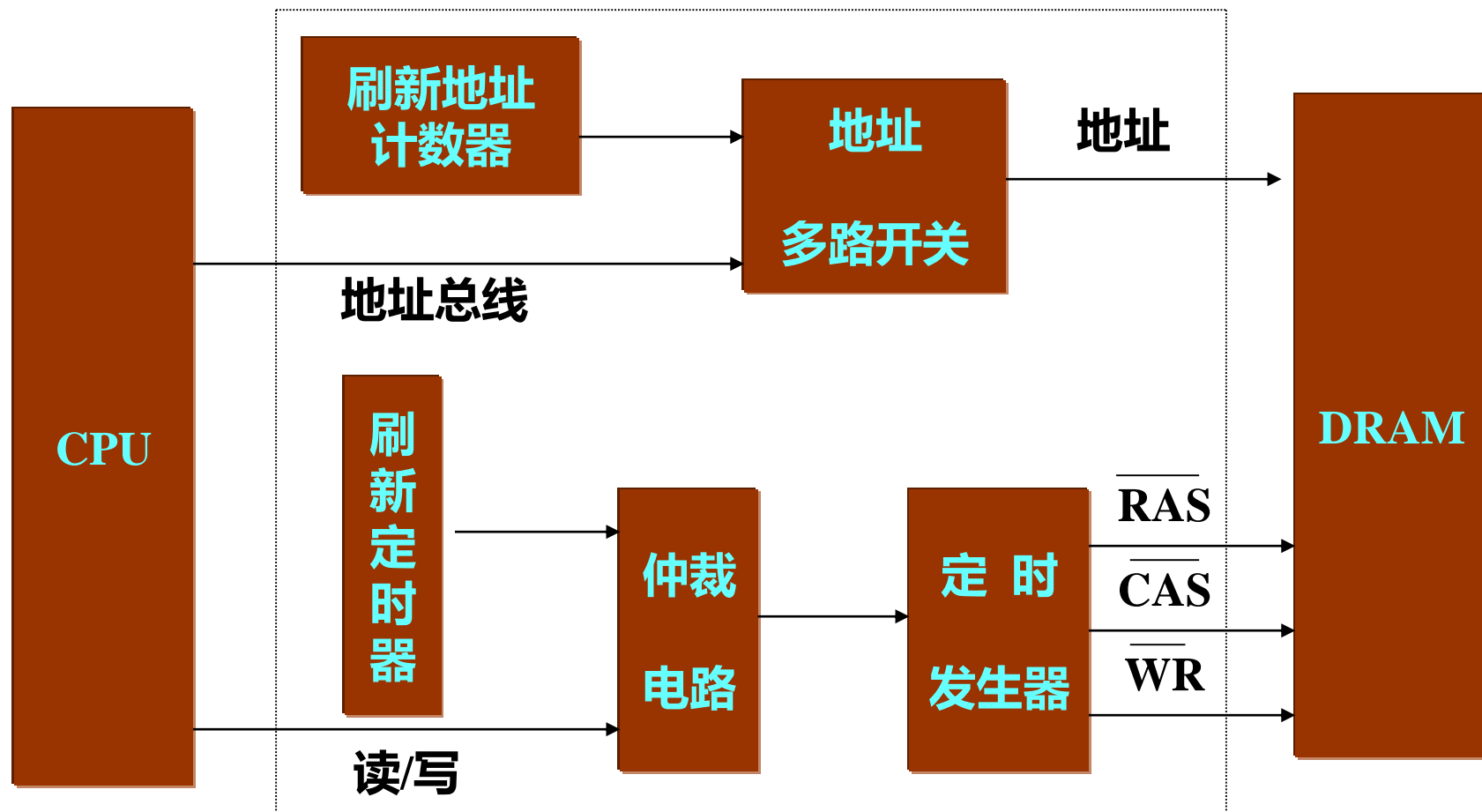
符 号	引脚名
$A_0 \sim A_9$	地址输入
$I/O_1 \sim I/O_4$	数据输入/输出
$CS-$	片选
$WE-$	写允许
$V_{CC}$ 、 $GND$	电源、地



# RAM的种类与选型 - DRAM

- 动态随机存取存储器（DRAM）
  - 存储信息的基本单元（1位）电路可采用4管、3管和单管电路
  - 需要不断刷新（为维持动态存储单元所存储的信息，必须设法使信息再生，这即所谓的刷新）
  - 与SRAM不同的是：为节省外部引脚，同样容量的DRAM外部地址线引脚是SRAM一半
  - DRAM采用行/列地址选通，将地址通过内部分成两路
  - DRAM控制器：解决刷新和多路
  - DRAM的应用较广，经常用做嵌入式计算机系统内存使用

# RAM的种类与选型 - DRAM



DRAM Controller的逻辑框图

# RAM的种类与选型 - SDRAM

- SDRAM (Synchronous RAM)

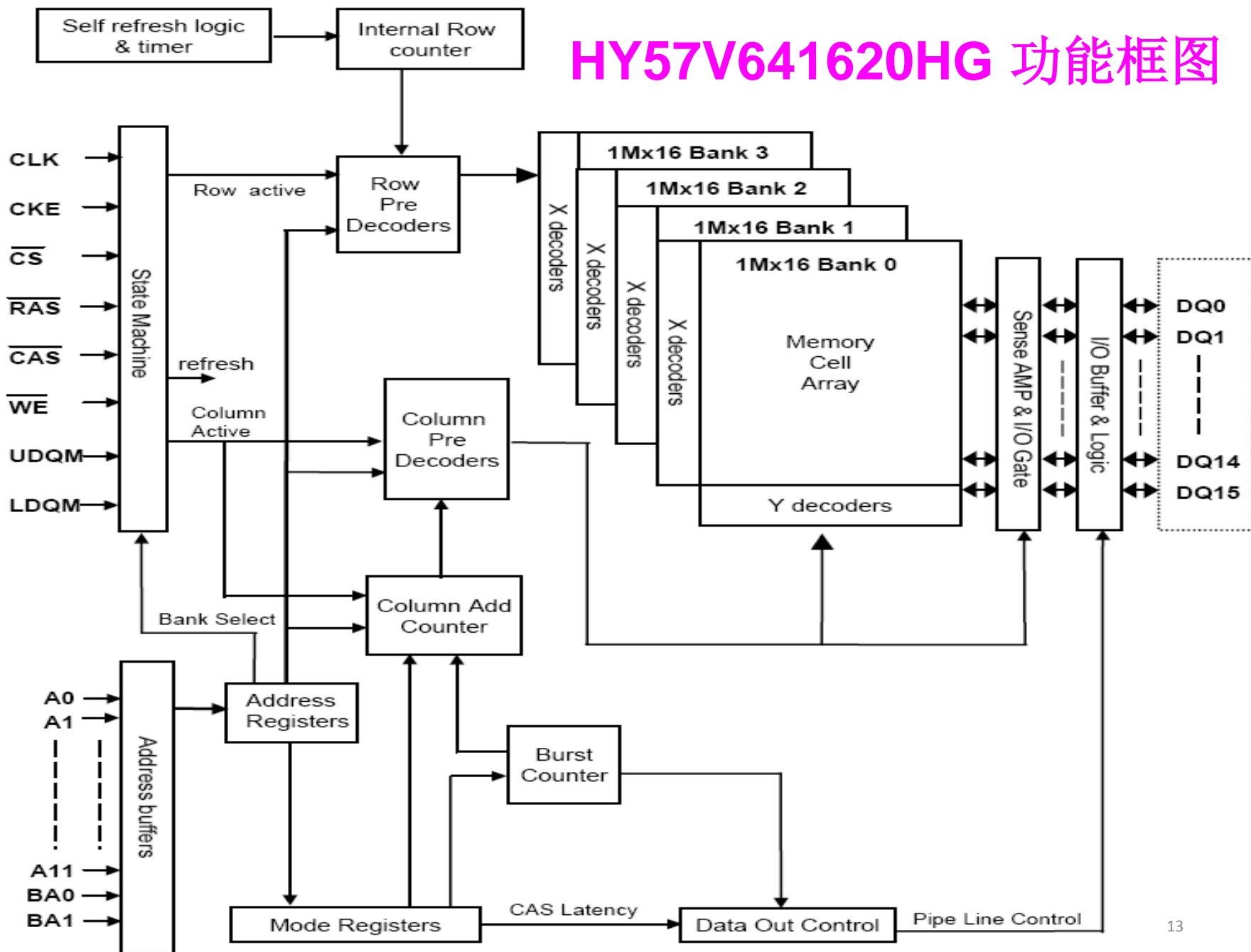
- CPU和RAM共享相同的时钟周期，以相同的速度同步工作
- 基于双存储器结构，内含两个交错的存储阵列，读取效率得到成倍提高
- DDRAM是基于SDRAM的一种新技术，在本质上和SDRAM完全相同
- DDRAM内存比普通SDRAM的带宽提升了一倍，也即是说在同样的时间内传送的数据量增加了一倍

# HY57V641620HGT-H SDRAM

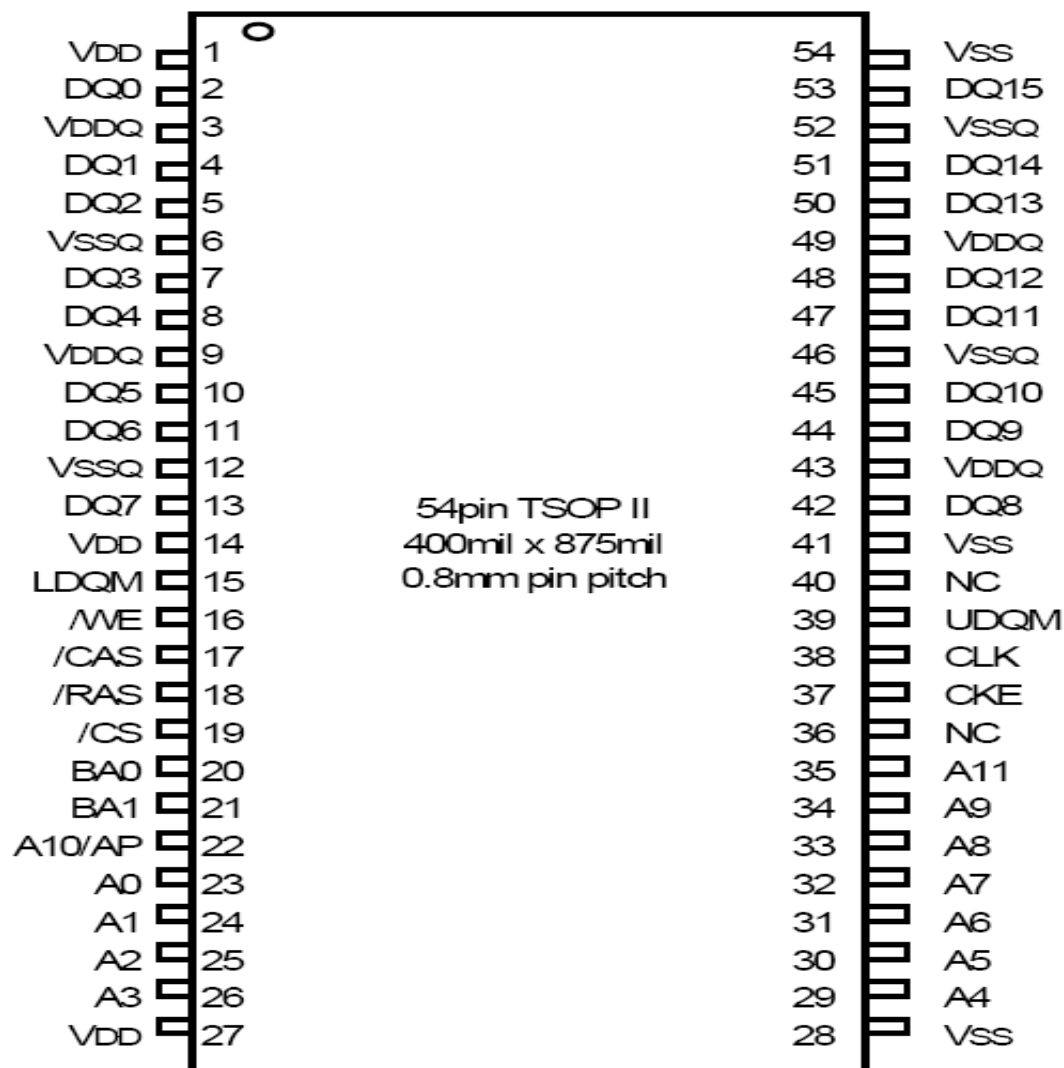
## • FEATURES

- Organized as 4 Banks x 1M x 16Bit Synchronous DRAM
- Single  $3.3 \pm 0.3V$  power supply
- All device pins are compatible with LVTTL interface
- JEDEC standard 400mil 54pin TSOP-II with 0.8mm of pin pitch
- All inputs and outputs referenced to positive edge of system clock
- Data mask function by UDQM or LDQM
- Internal four banks operation
- Auto refresh and self refresh
- 4096 refresh cycles / 64ms
- Programmable Burst Length and Burst Type
  - 1, 2, 4, 8 or Full page for Sequential Burst
  - 1, 2, 4 or 8 for Interleave Burst
  - Programmable CAS Latency ; 2, 3 Clocks

# HY57V641620HG 功能框图



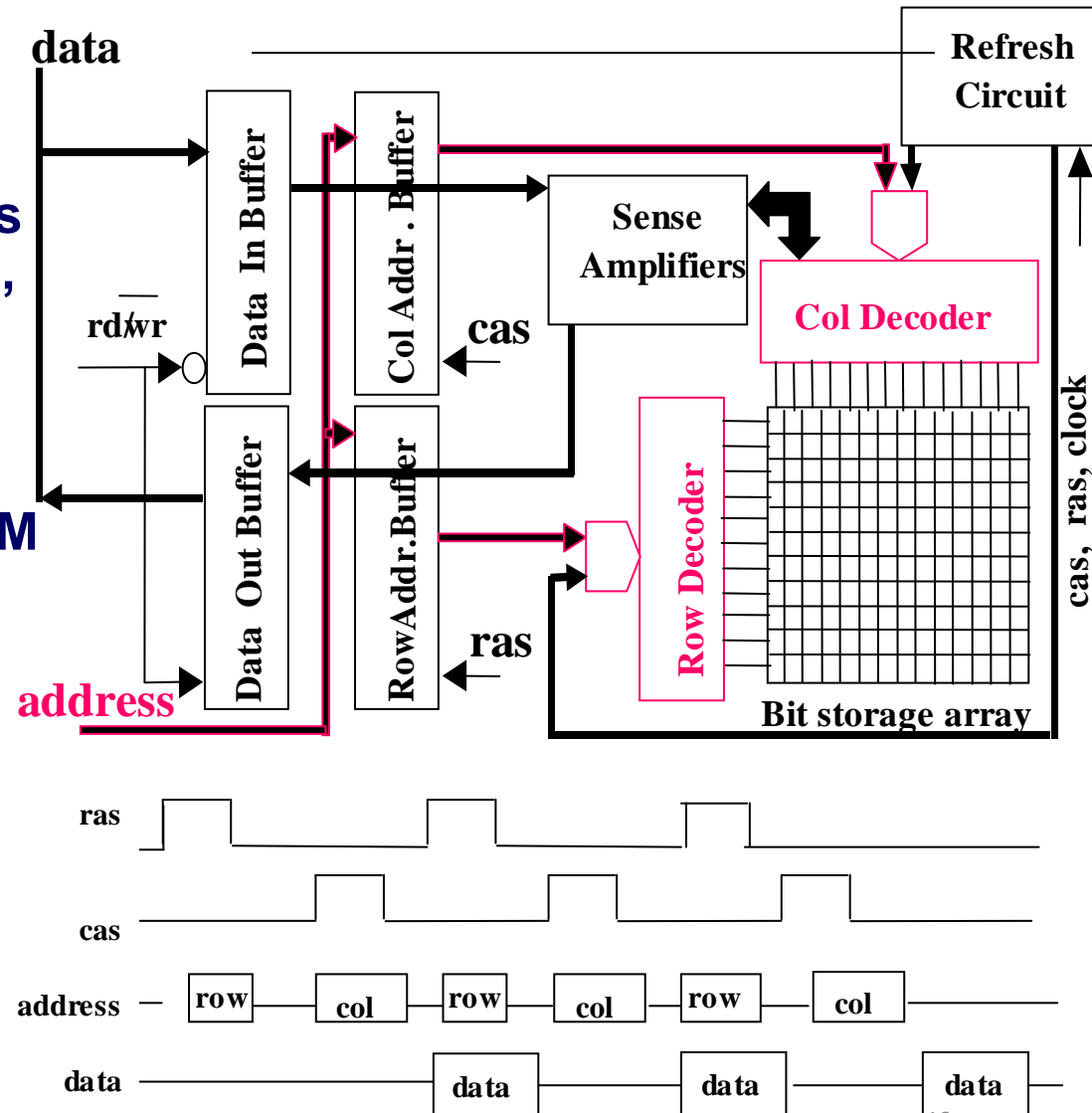
# 管脚及管脚信号描述



PIN	PIN NAME	DESCRIPTION
CLK	Clock	The system clock input. All other inputs are registered to the SDRAM on the rising edge of CLK
CKE	Clock Enable	Controls internal clock signal and when deactivated, the SDRAM will be one of the states among power down, suspend or self refresh
$\overline{\text{CS}}$	Chip Select	Enables or disables all inputs except CLK, CKE and DQM
BA0,BA1	Bank Address	Selects bank to be activated during $\overline{\text{RAS}}$ activity Selects bank to be read/written during $\overline{\text{CAS}}$ activity
A0 ~ A11	Address	Row Address : RA0 ~ RA11, Column Address : CA0 ~ CA7 Auto-precharge flag : A10
$\overline{\text{RAS}}$ , $\overline{\text{CAS}}$ , $\overline{\text{WE}}$	Row Address Strobe, Column Address Strobe, Write Enable	$\overline{\text{RAS}}$ , $\overline{\text{CAS}}$ and $\overline{\text{WE}}$ define the operation Refer function truth table for details
LDQM, UDQM	Data Input/Output Mask	Controls output buffers in read mode and masks input data in write mode
DQ0 ~ DQ15	Data Input/Output	Multiplexed data input / output pin
VDD/VSS	Power Supply/Ground	Power supply for internal circuits and input buffers
VDDQ/VSSQ	Data Output Power/Ground	Power supply for output buffers
NC	No Connection	No connection

# Basic DRAM

- Address bus multiplexed between row and column components
- Row and column addresses are latched in, sequentially, by strobing *ras* and *cas* signals, respectively
- Refresh circuitry can be external or internal to DRAM device
  - strokes consecutive memory address periodically causing memory content to be refreshed
  - Refresh circuitry disabled during read or write operation





# ROM的种类与选型

- ROM(只读存储器)的特点：
  - 在烧入数据后，无需外加电源来保存数据
  - 断电数据不丢失，但速度较慢
  - 适合存储需长期保留的不变数据
- 常见ROM的分类如下
  - Mask ROM (掩模ROM)
  - PROM (Programmable ROM, 可编程ROM)
  - EPROM (Erasable Programmable ROM, 可擦写ROM)
  - EEPROM (电可擦除可编程ROM)
  - Flash ROM (闪速存储器)

# ROM的种类与选型

- **掩模ROM**：一次性由厂家写入数据，用户无法修改
- **PROM(可编程ROM)**：出厂时厂家并没有写入数据，而是保留里面的内容为全0或全1，由用户来编程一次性写入数据，也就是改变部分数据为1或0
- **EPROM(可擦写ROM)**：EPROM是通过紫外光的照射，擦掉原先的程序。芯片可重复擦除和写入，解决了ROM芯片只能写入一次的弊端
- **EEPROM (E<sup>2</sup>PROM, 电可擦除可编程ROM)**：通过加电擦除原数据，通过高压脉冲可以写入数据。使用方便但价格较高，而且写入时间较长，写入较慢。
- **Flash ROM(闪速存储器)**：具有结构简单、控制灵活、编程可靠、加电擦写快捷的优点，而且集成度可以做得很高，它综合了前面的所有优点：断电不丢失数据(NVRAM)，快速读取，电可擦写可编程(EEPROM)，因此在手机，PC，PPC等电器中成功地获得了广泛的应用。

# Flash存储器的种类与选型

- Flash Memory的中文称为快闪存储器或快速擦写存储器
- Flash Memory由Toshiba于1980年申请专利，并在1984年的国际半导体学术会议上首先发表
- 目前在Flash Memory技术上主要发展了两种非易失性内存
  - 一种叫NOR（逻辑或），是Intel于1988年发明的
  - 另一种叫NAND（逻辑与）是Toshiba于1999年创造的

# Flash存储器的种类与选型

- **NOR Flash**

- 支持随机访问，速度快、电压低、功耗低、稳定性高，主要用于主存
- 应用程序可以直接在闪存内运行，不需要再把代码读到系统RAM中运行
- 优点：传输效率很高，在1 ~ 4MB 的小容量时具有很高的成本效益
- 缺点：写入和擦除速度很低

- **NAND Flash**

- NAND 支持顺序访问
- NAND Flash具有高存储密度，容量大、回写速度快、芯片面积小等特点，主要用于外存
- 应用NAND Flash的困难在于闪存和需要特殊的系统接口

# Flash存储器的种类与选型

- NOR和NAND型Flash的典型特征和不同点
  - (1)性能
  - (2)接口差别
  - (3)容量和成本
  - (4)可靠性和耐用性
  - (5)易用性
  - (6)软件支持
  - (7)市场定位
- NOR Flash用于对数据可靠性要求较高的代码存储、通信产品、网络处理等领域
- NAND Flash则用于对存储容量要求较高的MP3等；也可组成其他各种类型的电子盘如USB盘、CF、SD和MMC存储卡等

	NOR	NAND
<b>写入/擦除一个块的操作时间</b>	1 ~ 5s	2 ~ 4ms
<b>读性能</b>	1200 ~ 1500KB	600 ~ 800KB
<b>写性能</b>	<80KB	200 ~ 400KB
<b>接口/总线</b>	SRAM接口/独立的地 址数据总线	8位地址/数据/控制总线， I/O接口复杂
<b>读取模式</b>	随机读取	串行地存取数据
<b>成本</b>	较高	较低，单元尺寸约为NOR的一半， 生产过程简单，同样大小的芯片 可以做更大的容量
<b>容量及应用场合</b>	1 ~ 64MB，主要用于 存储代码	8MB ~ 4GB，主要用于存 储数据
<b>擦写次数(耐用性)</b>	约10万次	约100万次
<b>位交换(bit位反转)</b>	少	较多，关键性数据需要错误探 测/错误更正(EDC/ECC)算法
<b>坏块处理</b>	无，因为坏块故障率少	随机分布，无法修正

# SST39VF160 Nor Flash

- **FEATURES:**

- Organized as 1M x16
- Superior Reliability
  - Endurance: 100,000 Cycles (typical)
  - Greater than 100 years Data Retention
- **Low Power Consumption**(typical values at 14 MHz)
  - Active Current: 12 mA (typical) (动态)
  - Standby Current: 4  $\mu$ A (typical) (静态)
  - Auto Low Power Mode: 4  $\mu$ A (typical)
- **Sector-Erase Capability**
  - Uniform 2 KWord sectors (统一2K字)
- **Fast Read Access Time**
  - 55 ns for SST39LF160
  - 70 and 90 ns for SST39VF160

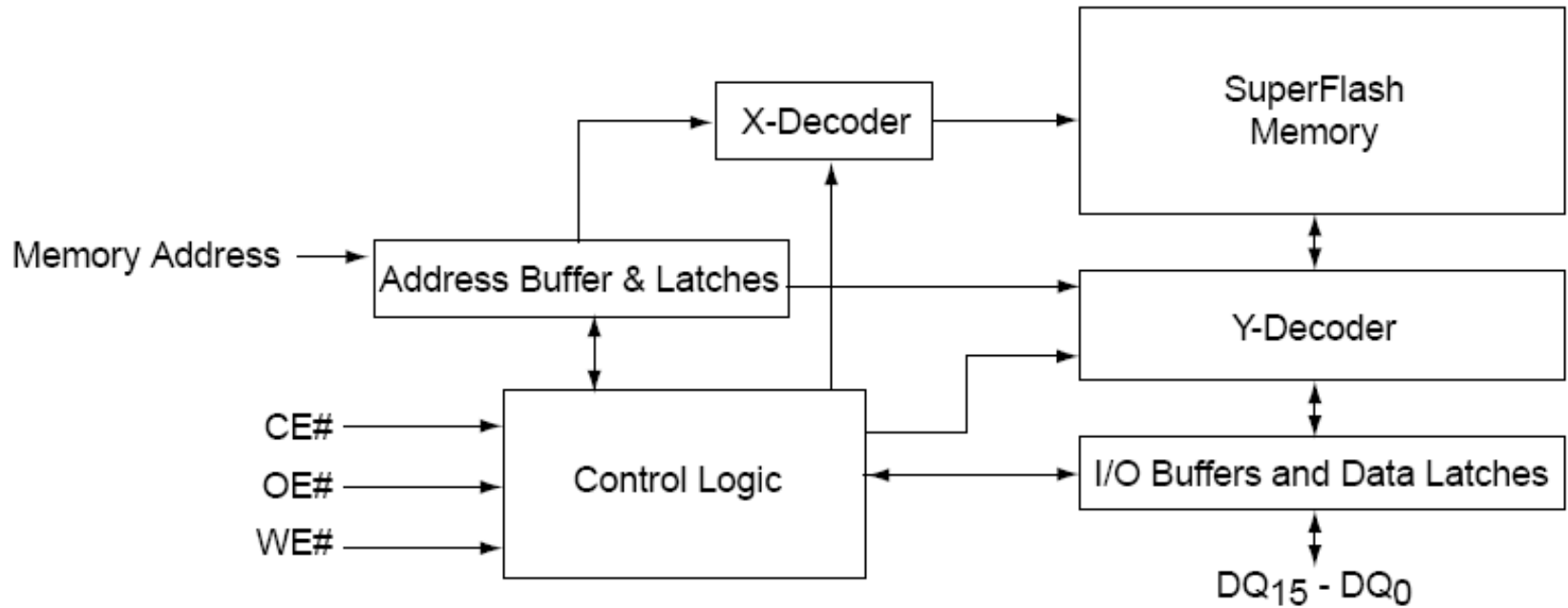
# SST39VF160 Nor Flash

- Fast Erase and Word-Program
  - Sector-Erase Time: 18 ms (typical)
  - Block-Erase Time: 18 ms (typical)
  - Chip-Erase Time: 70 ms (typical)
  - Word-Program Time: 14  $\mu$ s (typical)
  - Chip Rewrite Time: 15 seconds (typical) for SST39LF/VF160
- End-of-Write Detection
  - Toggle Bit(翻转)
  - Data# Polling (数据轮流检测)
- CMOS I/O Compatibility
- JEDEC Standard
  - Flash EEPROM Pinouts and command sets
- Packages Available
  - 48-lead TSOP (12mm x 20mm)
  - 48-ball TFBGA (8mm x 10mm)

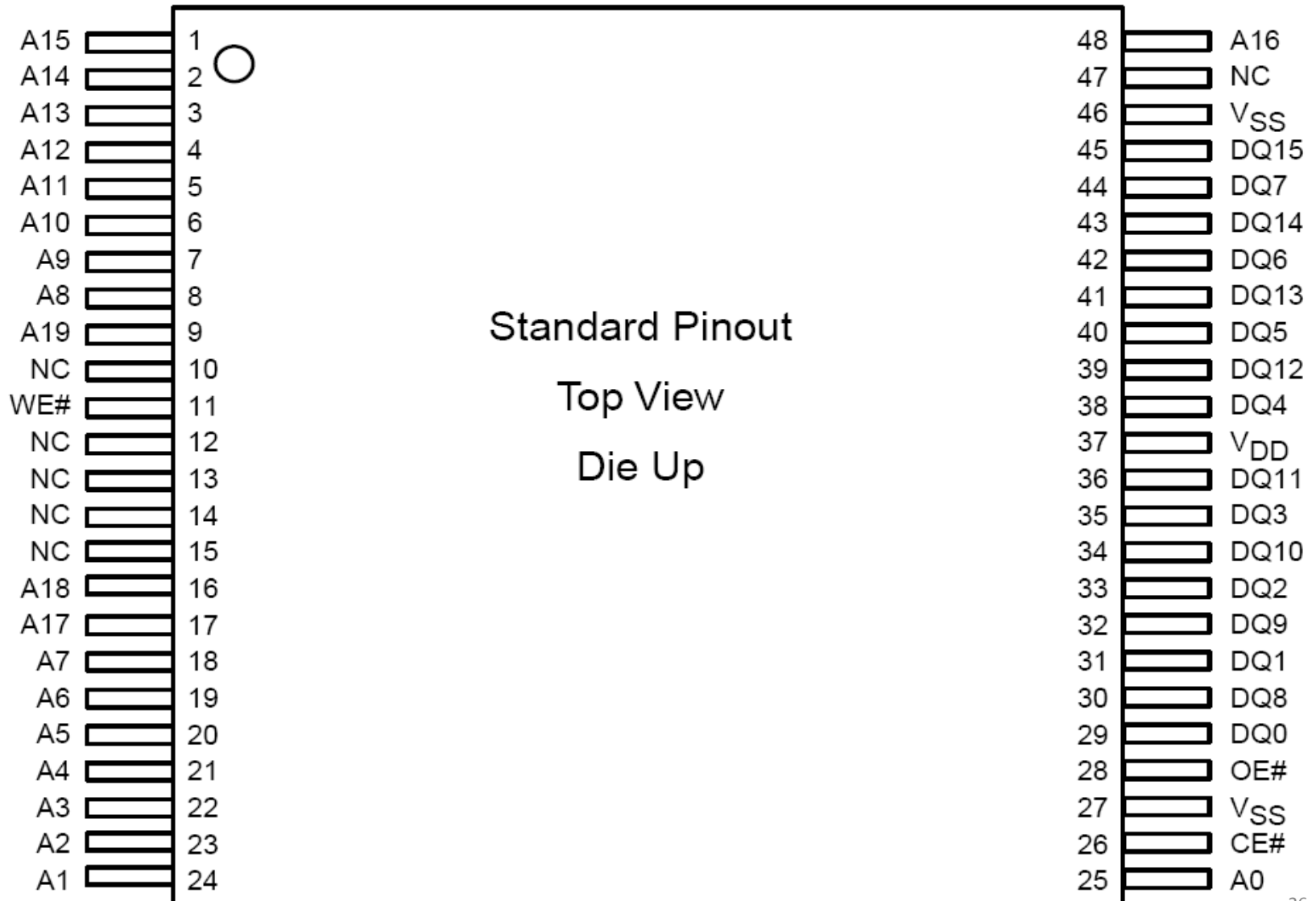




# SST39VF160 Diagram



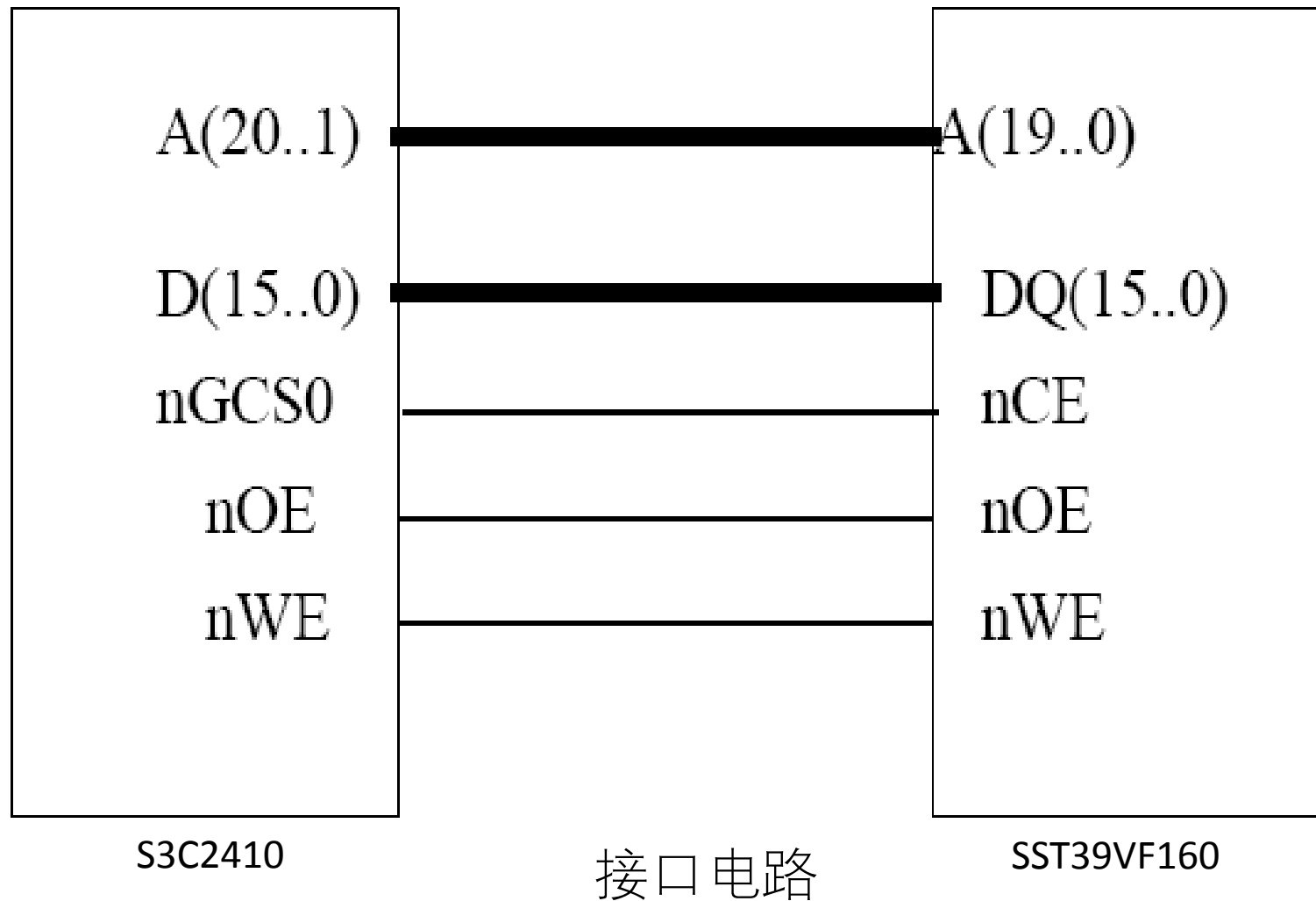
# SST39LF160/SST39VF160



# SST39VF160 Flash

Symbol	Pin Name	Functions
A <sub>19</sub> -A <sub>0</sub>	Address Inputs	To provide memory addresses. During Sector-Erase A <sub>19</sub> -A <sub>11</sub> address lines will select the sector. During Block-Erase, A <sub>19</sub> -A <sub>15</sub> address line will select the block.
DQ <sub>15</sub> -DQ <sub>0</sub>	Data Input/output	To output data during Read cycles and receive input data during Write cycles. Data is internally latched during a Write cycle. The outputs are in tri-state when OE# or CE# is high.
CE#	Chip Enable	To activate the device when CE# is low
OE#	Output Enable	To gate the data output buffers
WE#	Write Enable	To control the Write operations
V <sub>DD</sub>	Power Supply	To provide power supply voltage:   3.0-3.6V for SST39LF160 2.7-3.6V for SST39VF160
V <sub>SS</sub>	Ground	
NC	No Connection	Unconnected pins

# SST39VF160 接口电路



# 外部存储器的种类与选型

- 在嵌入式系统中常用的外存有：

- 磁盘存储器

- 光盘存储

- 电子盘、数据存储卡（闪存卡）等



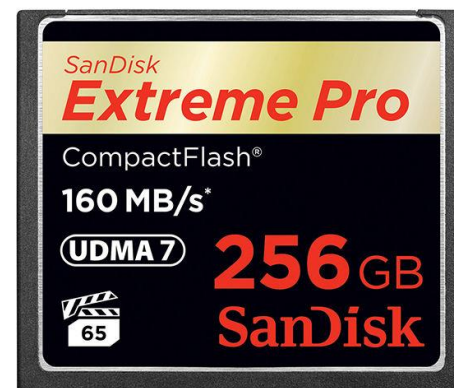
# 外部存储器的种类与选型 - 电子盘

- 电子盘采用半导体芯片来存贮数据，具有体积小、功耗低和极强的抗震性等特点
- 在嵌入式系统中普遍采用各种电子盘作为外存
- 常用的电子盘有：SD、CF、MMC、Memory Stick等



# Compact Flash (CF) 卡

- Compact Flash的诞生比较早，由最大的Flash Memory卡厂商之一美国SanDisk于1994年首次推出
- 大小仅为43mm x 36mm x 3.3mm，体积只有PCMCIA卡的1/4，看起来就像是PCMCIA卡的缩小版
- CompactFlash提供了完整的PCMCIA-ATA功能而且通过ATA/ATAPI-4兼容TrueIDE
- 和68针接口的PCMCIA卡不同，同样遵从ATA协议的CompactFlash的接口只有50针



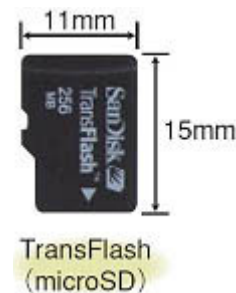
# SD卡

- SD (Secure Digital) 卡由松下电器、东芝和SanDisk联合推出，1999年8月首次发布。
- 2000年2月1日成立了SD协会 (SDA)，成员公司超过90个，其中包括Hewlett-Packard, IBM, Microsoft, Motorola, NEC、Samsung Electronics, Toyota Motor等巨头。
- SDA是开放式的，缴纳1500美元就可以成为一般会员，缴纳4000美元可以成为执行会员。SD存储卡的详细规范并没有公开，只有SDA会员或签定了保密协议才能获得。
- SD卡需付版税

32×24mm

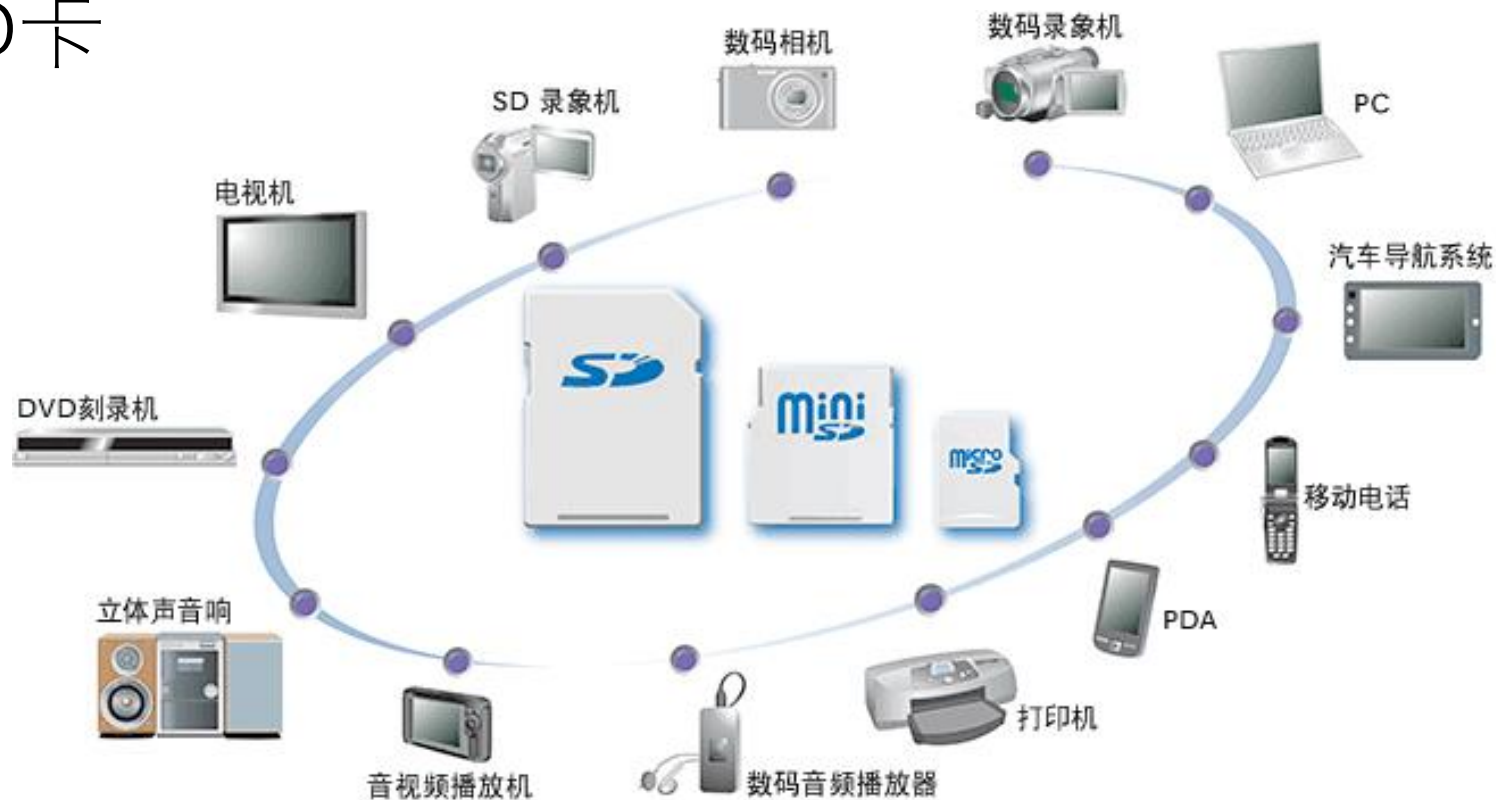


15×11mm





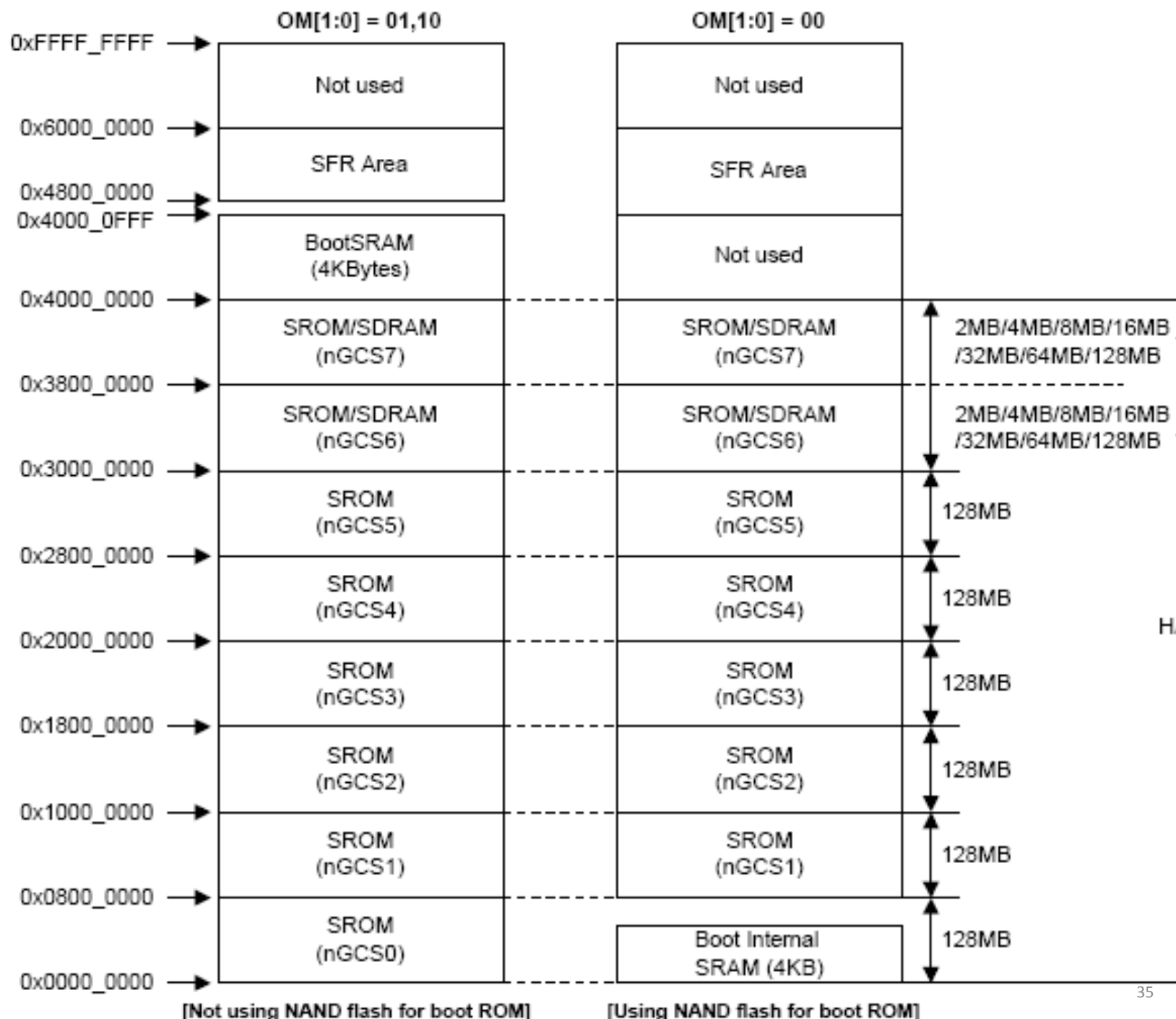
# SD卡



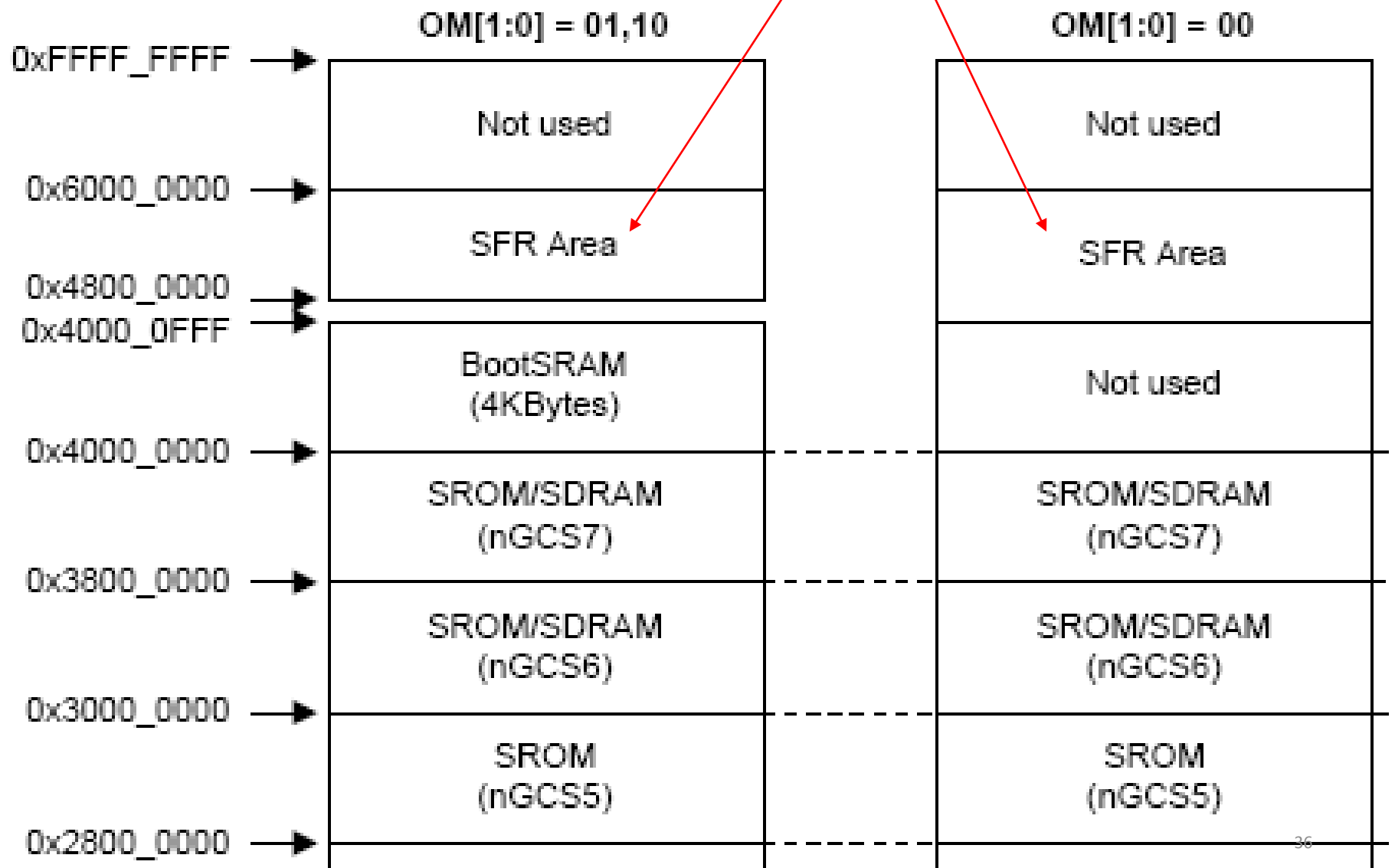
## 6.2 S3C2410X存储器配置

---

- 存储器管理器控制信号：27位地址、32位数据、8个片选、以及读/写控制信号等
- 存储空间：分成8组，最大容量是1GB
  - bank0---bank5为固定128MB
  - bank6和bank7的容量可编程改变，可以是2、4、8、16、32、64、128MB，并且bank7的开始地址与bank6的结束地址相连接，但是二者的容量必须相等
- Bank0可以作为引导ROM，其数据线宽只能是16位和32位，复位时由OM0、OM1引脚确定；其它存储器的数据线宽可以是8位、16位和32位
- 存储器格式：可编程设置为大端格式或小端格式



# 各个接口的控制寄存器



## 6.3 存储器模块硬件设计及控制

- Bank0 总线宽度: Bank0 的数据总线 (nGCS0) 必须首先设置成16 位或32 位的。因为bank0 通常作为引导ROM 区 (映射到地址0x0000-0000) , 在复位时, 系统将检测OM[1:0]上的逻辑电平, 并依据这个电平来决定bank0 区存储器的总线宽度

OM1 (Operating Mode 1)	OM0 (Operating Mode 0)	Booting ROM Data width
0	0	Nand Flash Mode
0	1	16-bit
1	0	32-bit
1	1	Test Mode

# SDRAM 线连接

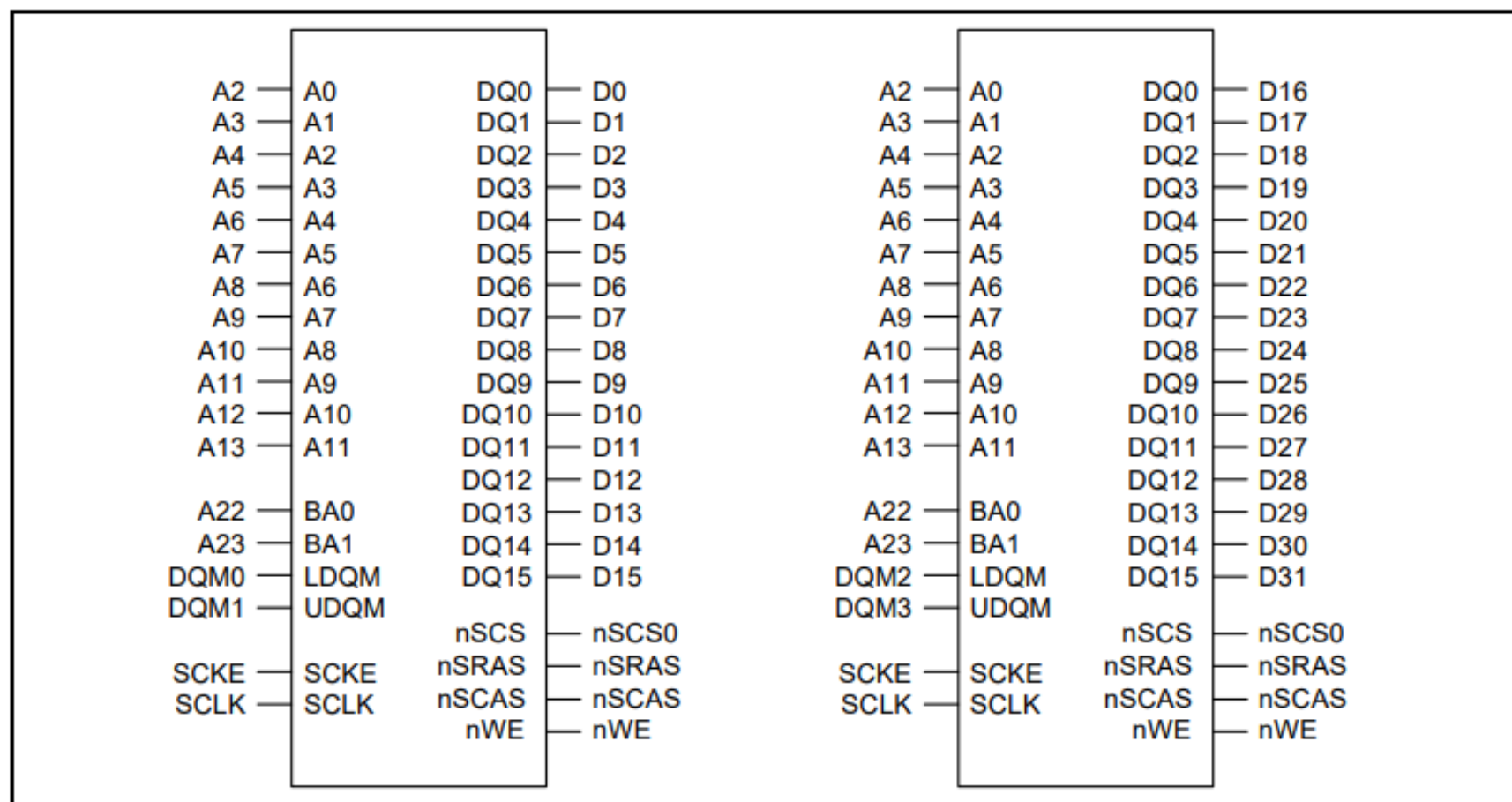
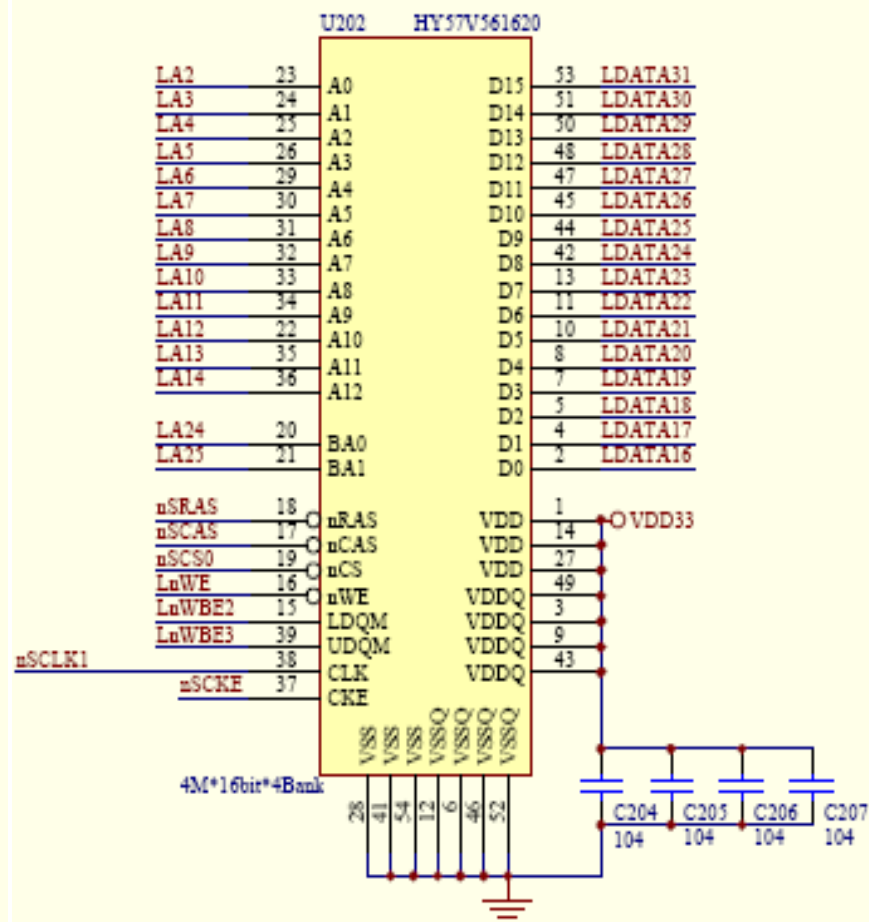
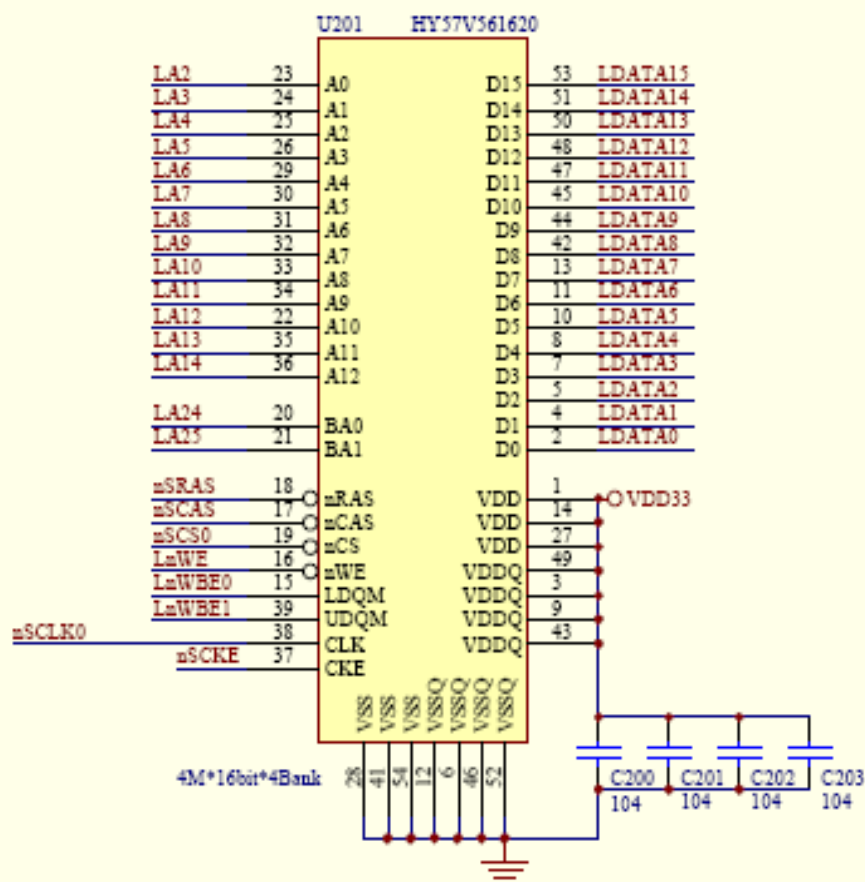


Figure 5-11. Memory Interface with 16-bit SDRAM (16MB: 1Mb ' 16 ' 4banks ' 2ea)

# SDRAM模块硬件设计

- 例：SDRAM为4Mx16Bitx4Bank



# 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
BWCON	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 <sub>0</sub>



# 1、总线宽度和等待控制寄存器BWSCON

用于设定各存储块的数据宽度以及是否使能nwait

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

## 2、BANKn---存储器组控制寄存器 (n=0--5)

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

**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。

## 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										

**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=15.6μs，HCLK=60MHz

则 刷新计数器值=2<sup>11</sup>+1-60×15.6=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

## 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		

**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

# SDRAM的初始化

SDRAM 在工作前需要进行初始化

- 1、SDRAM上电，时钟稳定，DQM高，CKE高，开始初始化
- 2、至少100us等待时间，在这个等待时间之内不能有除了COMMAND INHIBIT 或 NOP 以外的任何命令，COMMAND INHIBIT 或 NOP命令需持续生效
- 3、在100us等待时间结束之后，所有的 bank 需要预充电，（执行PRECHARGE 需要  $t_{RP}$  的时间）
- 4、至少两个自动刷新命令（每个自动刷新都需要 $t_{RC}$ 的时间）
- 5、模式寄存器的配置（需要 $t_{MRD}$ 的时间）。
- 6、初始化完成以后处于idle的状态

## 6.4 Nand Flash及其控制器

---

### 主要内容

- 1、Nand Flash控制器概述
- 2、Nand Flash控制器硬件设计
- 3、控制器的寄存器
- 4、控制器的工作原理

## 6.4 Nand Flash及其控制器

**Nor flash存储器：**读速度高，而擦、写速度低，容量小，价格高

**Nand flash存储器：**读速度不如Nor flash，而擦、写速度高，容量大，价格低。有取代磁盘的趋势

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

# 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**
  - 当OM1、OM0都是低电平时，S3C2410从NAND Flash启动
  - 复位后，NAND Flash开始的4k代码会被自动地复制到内部SRAM中的Steppingstone，传送后，引导代码在Steppingstone中执行
  - 使用这4k代码来把更多的代码从NAND Flash中读到SDRAM中去运行
- **具有硬件ECC（纠错码）功能**
  - 具备硬件ECC（校验码，Error Correction Code）生成模块（硬件生成校验码，通过软件校验）
- **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

# Nand Flash控制器主要特性

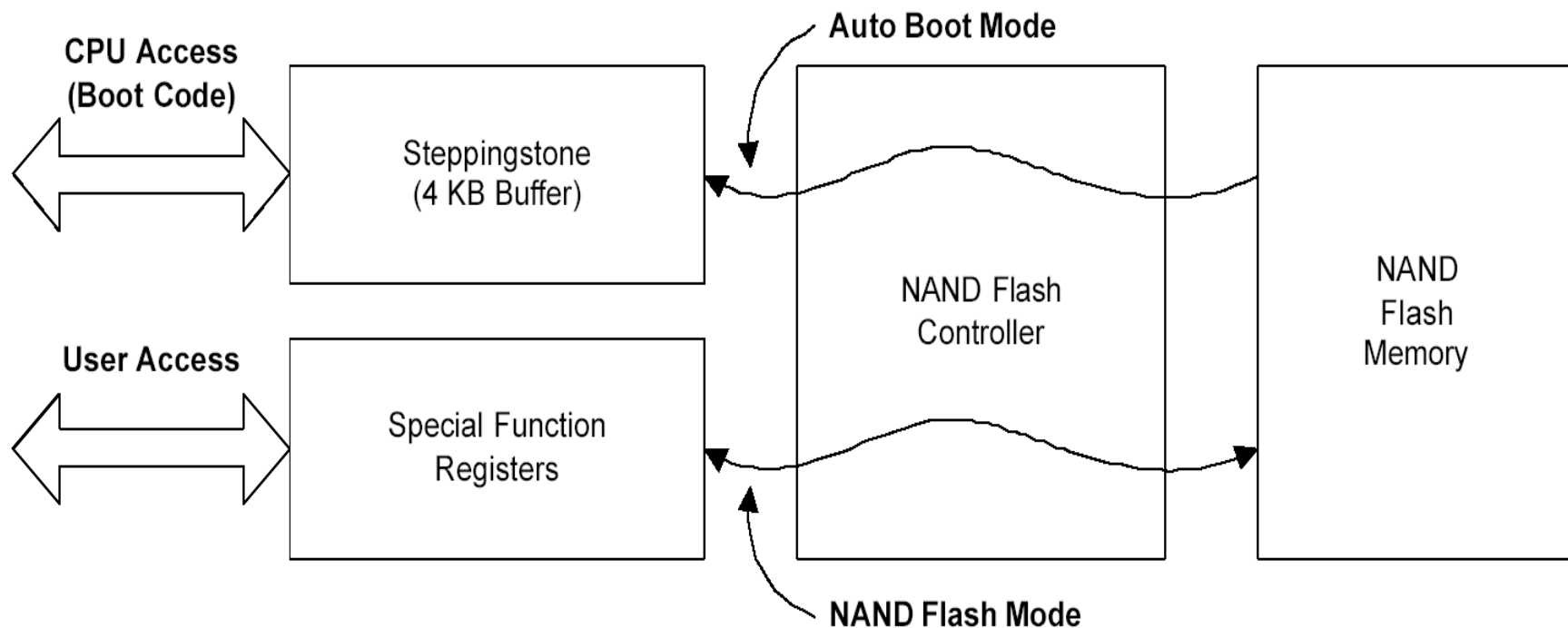
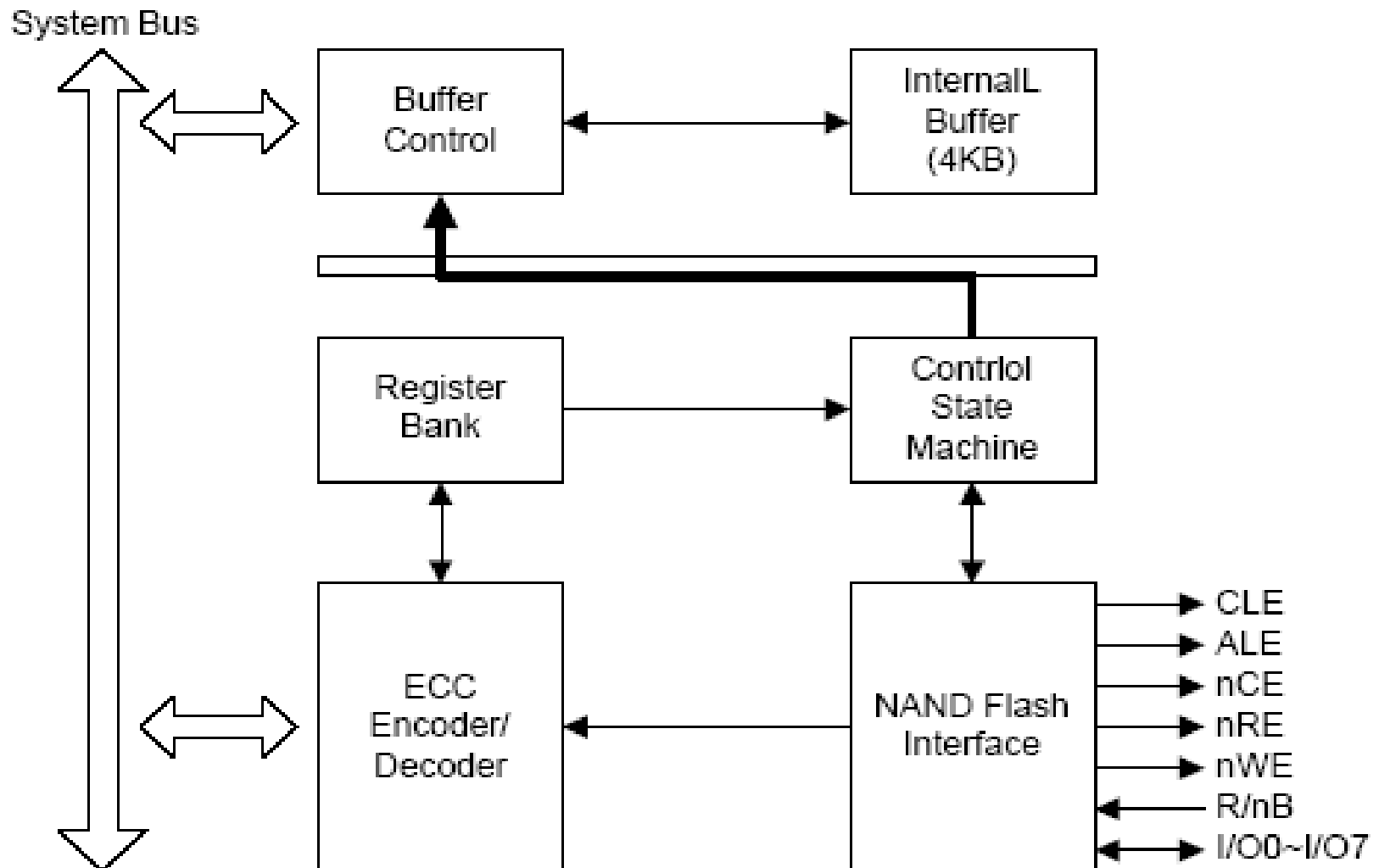


图4.4.2 NAND Flash的操作模式

# Nand Flash控制器硬件设计



# Nand Flash控制器硬件设计

- NAND接口信号：

I/O0 ~ I/O7：用于输入地址/数据/命令，输出数据

CLE：Command Latch Enable，命令锁存使能

ALE：Address Latch Enable，地址锁存使能

CE#：Chip Enable，芯片使能

RE#：Read Enable，读使能

WE#：Write Enable，写使能

WP#：Write Protect，写保护

R/B#：Ready/Busy Output，就绪/忙，主要用于在发送完编程/擦除命令后，检测这些操作是否完成：忙，表示编程/擦除操作仍在进行中；就绪，表示操作完成

Vcc：Power，电源

Vss：Ground，接地



# Nand Flash 控制器的寄存器

- NAND Flash控制器不能通过DMA访问，可以使用LDM/STM指令来代替DMA操作
- NAND Flash的操作通过NFCONF、NFCMD、NFADDR、NFDATA、NFSTAT和NFECC六个寄存器来完成

# Nand Flash 控制器的寄存器

寄存器	地 址	功 能	操作	复位值
NFCONF	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	-

# 1、NFCONF---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； 1：nFCE为高有效

**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**

# 1、NFCNF---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		

**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、NFECC---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							

# Nand Flash 控制器的工作原理

## 1、自动导入启动代码步骤

- 完成复位
- 如果自动导入模式使能，Nand flash存储器的前面4K字节被自动拷贝到Steppingstone内部缓冲器中
  - Steppingstone被映射到nGCS0对应的BANK0存储空间
  - CPU在Steppingstone的4-KB内部缓冲器中开始执行引导代码

**注意：**在自动导入模式下，不进行ECC检测。因此，Nand flash的前4KB应确保不能有位错误（一般Nandflash厂家都确保）

# Nand Flash 控制器的工作原理

## 2、Nand FLASH模式配置

- 通过NFCNF寄存器配置Nand flash
- 写Nand flash命令到NFCMD寄存器
- 写Nand flash地址到NFADDR寄存器
- 在读写数据时，通过NFSTAT寄存器来获得Nand flash的状态信息。应该在读操作前或写入之后检查R/nB信号（准备好/忙信号）
- 在读写操作后要查询校验错误代码，对错误进行纠正



# Nand Flash 控制器的工作原理

## 3、系统引导和 Nand FLASH 配置

- OM[3:0]为芯片引脚，设置引导模式、存储器bank0的数据宽度、时钟模式等
  - OM[1:0] = 00b：使能Nand flash控制器自动导入模式；
  - OM[1:0] = 01b\10b： bank0数据宽度为16\32位
  - OM[1:0]=11b： 测试模式
- Nand flash的存储页面大小为512字节
- NCON： Nand flash 寻址步骤数选择
  - 0： 3步寻址；    1： 4步寻址

# Nand Flash 控制器的工作原理

## 4、Nand Flash操作的校验问题

**S3C2410A在写/读操作时，每512字节数据自动产生3字节的ECC奇偶代码(24位)。**

**24位 ECC 奇偶代码 = 18位行奇偶 + 6位列奇偶**

**ECC产生模块执行以下步骤：**

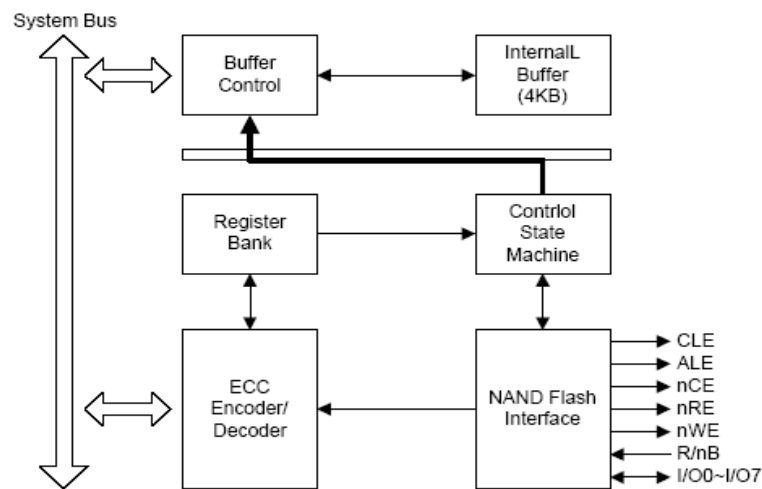
- 当MCU写数据到Nand时，ECC产生模块生成ECC代码
- 当MCU从Nand读数据时，ECC产生模块生成ECC代码同时用户程序将它与先前写入时产生的ECC代码比较

## 6.5 存储器模块驱动程序设计

- 存储器模块驱动程序包括初始化程序及读写程序
- Flash与SDRAM的驱动程序有很大的不同：
  - SDRAM驱动程序包括初始化、读与写程序。初始化复杂，读写简单，无需擦除。
  - Flash驱动程序包括读、写与擦除程序（无初始化程序）

# Flash存储器的读写与擦除

- **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（随机访问）
- 应用程序可以直接在**NOR Flash**内运行，不需要再把代码读到系统RAM中运行
- **NAND Flash**接口**逻辑复杂**，地址、数据和命令共用8位总线/16位总线，每次读写都要使用复杂的I/O接口串行地存取数据，8位总线/16位总线用来传送控制、地址和资料信息
- NOR Flash的读速度比NAND Flash稍快一些，NAND Flash的随机读取能力差，适合大量数据的连续读取



# Flash存储器的读写与擦除

- 除了NOR Flash的读，Flash Memory的其他操作不能像RAM那样，直接对目标地址进行总线操作。例如执行一次写操作，它必须输入一串特殊的指令（NOR Flash），或者完成一段时序（NAND Flash）才能将数据写入到Flash Memory中
- Flash Memory在物理结构上分成若干个区块，区块之间相互独立，以区块为单位的内存单元进行擦写和再编程
  - NOR Flash把整个存储区分成若干个扇区（Sector）
  - NAND Flash把整个存储区分成若干个块（Block）
- 写
  - Flash Memory的写操作只能将数据位从1写成0，而不能从0写成1，在对存储器进行写入之前必须先执行擦除操作，将预写入的数据位初始化为1
- 擦除
  - 擦操作的最小单位是一个区块，而不是单个字节
  - NAND Flash执行擦除操作是十分简单的
  - NOR Flash要求在在进行擦除前先要将目标块内所有的位都写为0

# Nor Flash 驱动程序设计

Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
	Addr <sup>1</sup>	Data <sup>2</sup>	Addr <sup>1</sup>	Data <sup>2</sup>	Addr <sup>1</sup>	Data <sup>2</sup>	Addr <sup>1</sup>	Data <sup>2</sup>	Addr <sup>1</sup>	Data <sup>2</sup>	Addr <sup>1</sup>	Data <sup>2</sup>
Word-Program	5555H	AAH	2AAAH	55H	5555H	A0H	WA <sup>3</sup>	Data				
Sector-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	SA <sub>X</sub> <sup>4</sup>	30H
Block-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	BA <sub>X</sub> <sup>4</sup>	50H
Chip-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	5555H	10H
Software ID Entry <sup>5,6</sup>	5555H	AAH	2AAAH	55H	5555H	90H						
CFI Query Entry <sup>5</sup>	5555H	AAH	2AAAH	55H	5555H	98H						
Software ID Exit <sup>7</sup> / CFI Exit	XXH	F0H										
Software ID Exit <sup>7</sup> / CFI Exit	5555H	AAH	2AAAH	55H	5555H	F0H						

# Nor Flash 驱动程序设计

- Flash驱动程序包括读、写与擦除程序（无初始化程序）

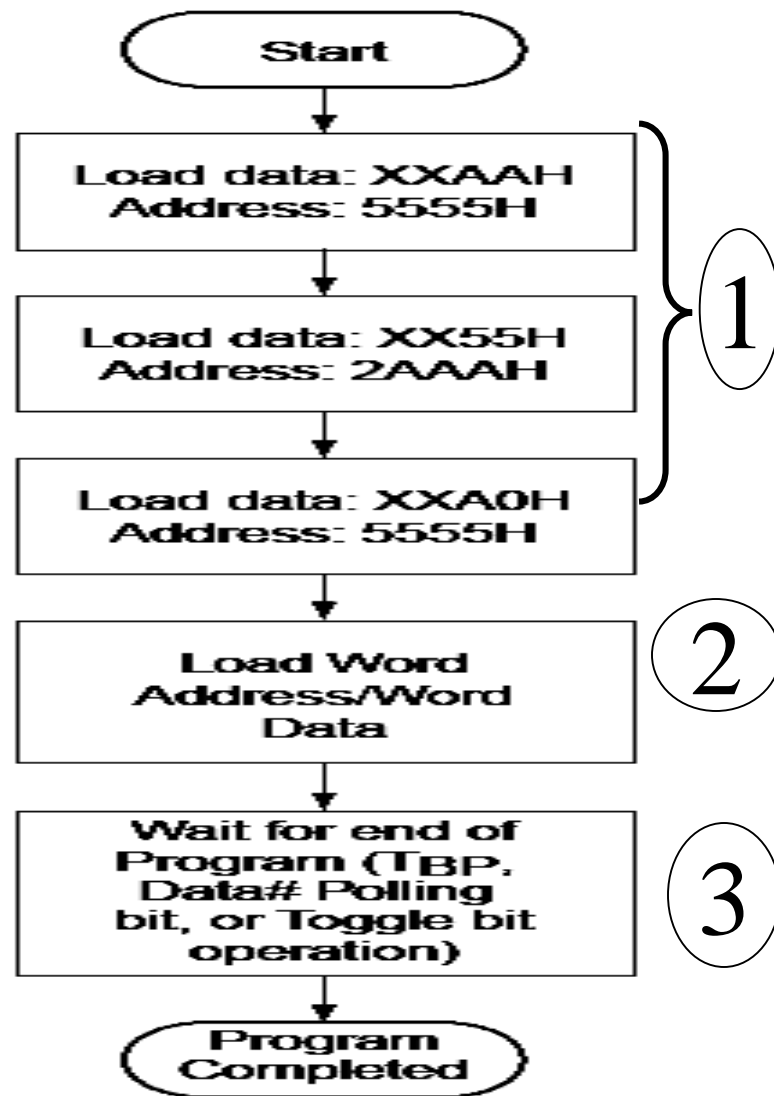
## ① 读

- 输出地址信号选中所需读出单元的地址
- CE和OE同时变低即可从数据线DQ15~DQ0读取数据

# Nor Flash 驱动程序设计

## ②写

- 每次写入一个字；写之前，若扇区中有数据，必须先进行擦除
- 写操作由3步组成
  - (1) 写入数据保护字节
    - 0xAA;0x55;0xA0
  - (2) 写入地址和数据
  - (3) 等待写入结束





# Nor Flash 驱动程序设计

## ③扇区擦除 (2k Word)

- 6字节指令系列
  - AAh@5555h;
  - 55h@2AAA;
  - 80h@5555h;
  - AAh@5555h;
  - 55h@2AAAh;
  - 30h@SAn

扇区擦除指令

### Sector-Erase Command Sequence

Load data: XXAAH  
Address: 5555H

Load data: XX55H  
Address: 2AAAH

Load data: XX80H  
Address: 5555H

Load data: XXAAH  
Address: 5555H

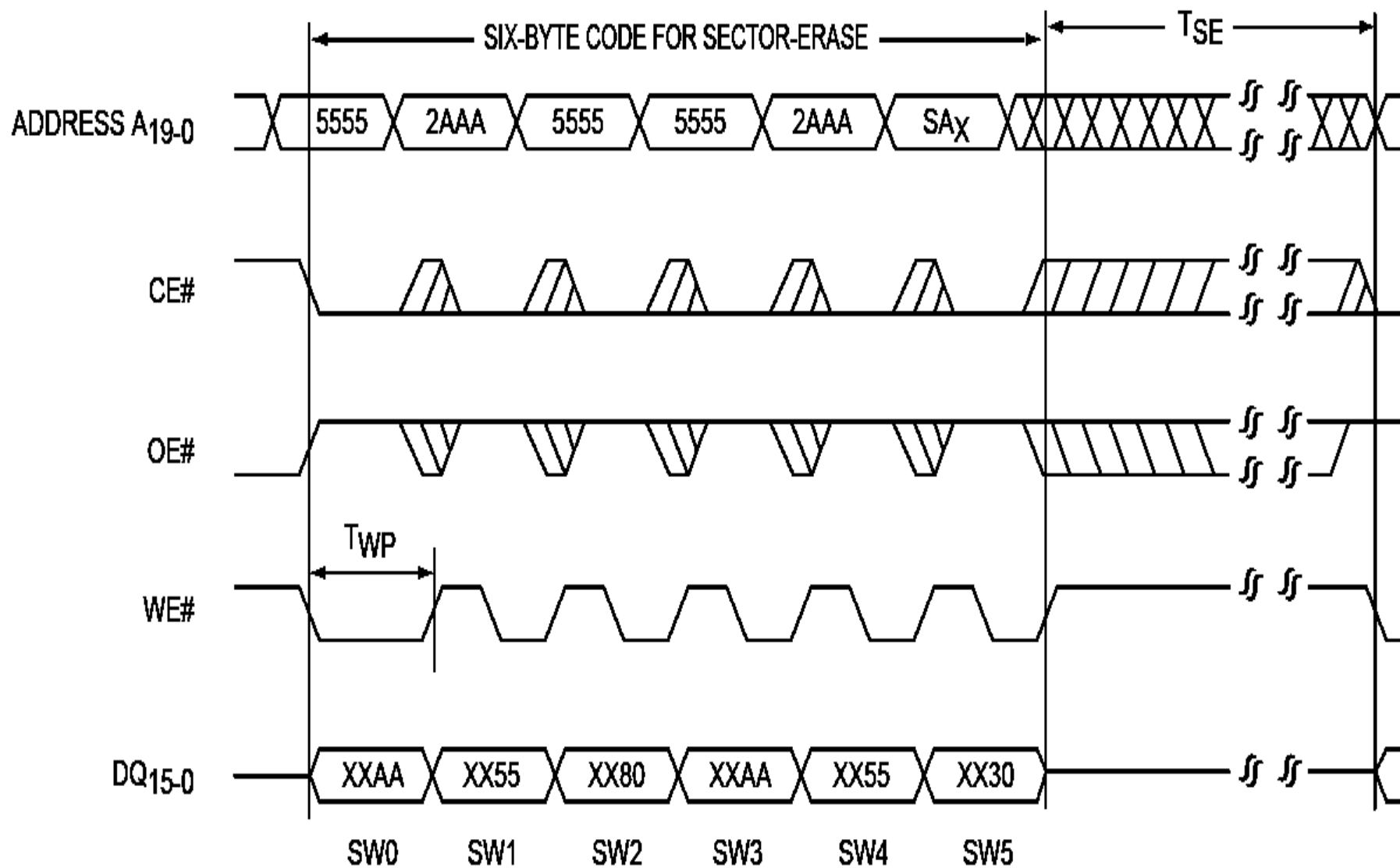
Load data: XX55H  
Address: 2AAAH

Load data: XX30H  
Address: SAn

Wait TSE

Sector erased  
to FFFFH

# Nor Flash 驱动程序设计



**WE# CONTROLLED SECTOR-ERASE TIMING DIAGRAM**

# Nor Flash 驱动程序设计

## ④块擦除 (32k Word)

• 6字节指令系列

- AAh@5555h;
- 55h@2AAA;
- 80h@5555h;
- AAh@5555h;
- 55h@2AAAh;
- 50h@BAx

块擦除指令

### Block-Erase Command Sequence

Load data: XXAAH  
Address: 5555H

Load data: XX55H  
Address: 2AAAH

Load data: XX80H  
Address: 5555H

Load data: XXAAH  
Address: 5555H

Load data: XX55H  
Address: 2AAAH

Load data: XX50H  
Address: BAx

Wait T<sub>BE</sub>

Block erased  
to FFFFH

# Nor Flash 驱动程序设计

## ⑤片擦除

• 6字节指令系列

- AAh@5555h;
- 55h@2AAA;
- 80h@5555h;
- AAh@5555h;
- 55h@2AAAh;
- 10h@5555h

片擦除指令

### Chip-Erase Command Sequence

Load data: XXAAH  
Address: 5555H

Load data: XX55H  
Address: 2AAAH

Load data: XX80H  
Address: 5555H

Load data: XXAAH  
Address: 5555H

Load data: XX55H  
Address: 2AAAH

Load data: XX10H  
Address: 5555H

Wait T<sub>SCE</sub>

Chip erased  
to FFFFH

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

# 7 DMA控制器

---

## 主要内容

1、概述

2、工作原理

3、寄存器

重点：

- (1) DMA优点
- (2) DMA工作过程

# 一、概 述

S3C2410X有4个通道的DMA控制器，位于系统总线和外设总线之间

每个DMA通道都能没有约束的实现系统总线或者外设总线之间的数据传输，即每个通道都能处理下面四种情况：

- (1) 源器件和目的器件都在系统总线
- (2) 源器件在系统总线，目的器件在外设总线
- (3) 源器件在外设总线，目的器件在系统总线
- (4) 源器件和目的器件都在外设总线

**DMA的主要优点是：可以不通过CPU的中断来实现数据的传输**，DMA的运行可以通过软件或者通过外围设备的中断和请求来初始化

## 二、DMA工作原理

### 1、DMA的服务对象

每个DMA通道都有4个DMA请求源，通过设置，可以从中挑选一个服务。每个通道的DMA 请求源如表4-1所示

表4-1 各通道的DMA 请求源

通道 \ 源	请求源0	请求源1	请求源2	请求源3	请求源4
通道0	nXDREQ0	UART0	SDI	Timer	USB设备EP1
通道1	nXDREQ1	UART1	IIS/SDI	SPI0	USB设备EP2
通道2	IISDO	IISDI	SDI	Timer	USB设备EP3
通道3	UART2	SDI	SPI1	Timer	USB设备EP4



## 2、DMA的工作过程

(1) 外设向DMAC发出请求

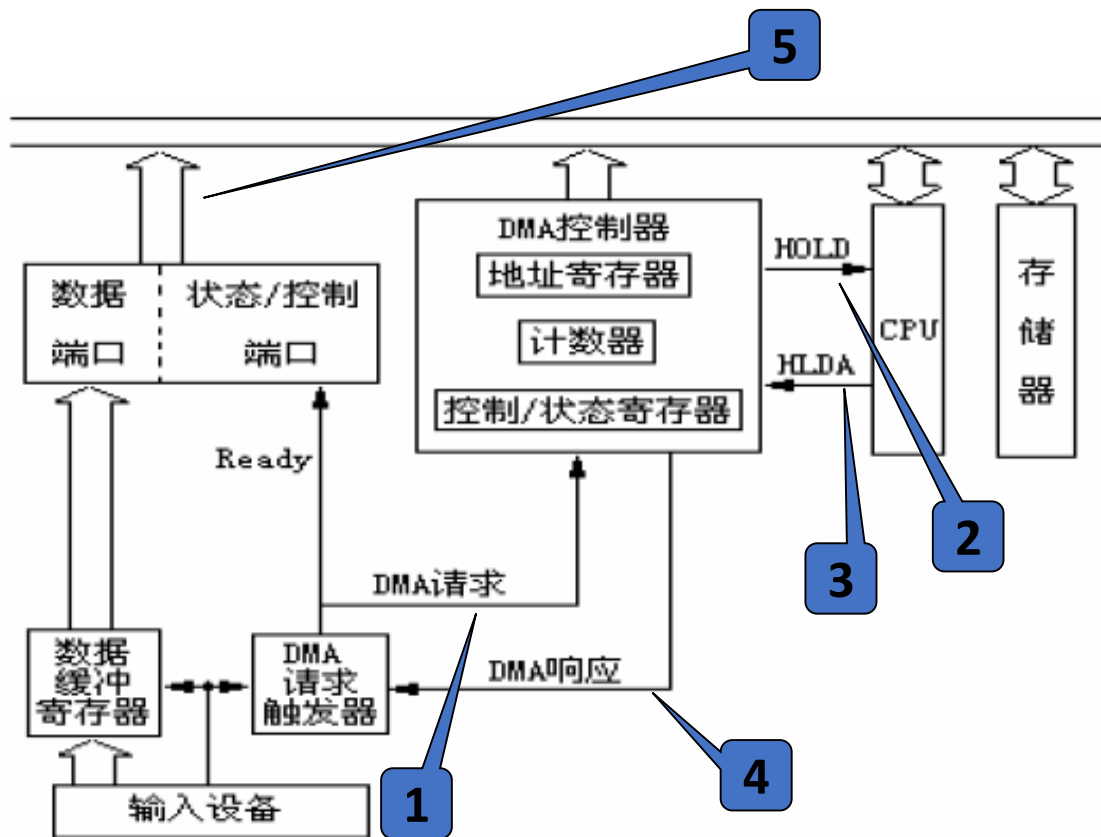
(2) DMAC通过HOLD向CPU发出总线请求

(3) CPU响应释放三总线，并且发应答HLDA

(4) DMAC向外设发DMA应答

(5) DMAC发出地址、控制信号，为外设传送数据

(6) 传送完规定的的数据后，DMAC撤销HOLD信号，CPU也撤销HLDA信号，并且恢复对三总线的控制



## 2、DMA的工作过程

S3C2410X的DMA工作过程可以分为三个状态：

- **状态1：等待状态。** DMA 等待一个DMA请求。如果有请求到来，将转到状态2。在这个状态下，DMA ACK和INT REQ为0
- **状态2：准备状态。** DMA ACK变为1，计数器（CURR\_TC）装入 DCON[19:0]寄存器。

注意：DMA ACK保持为1直至它被清除

- **状态3：传输状态。** DMA控制器从源地址读入数据并将它写到目的地址，每传输一次，CURR\_TC数器（在DSTAT中）减1，并且可能做以下操作：

## 2、DMA的工作过程

- **重复传输**：在全服务模式下，将重复传输，直到计数器CURR\_TC变为0；在单服务模式下，仅传输一次
- **设置中断请求信号**：当CURR\_TC变为0时，DMAC发出INT REQ信号，而且DCON[29]即中断设定位被设为1
- **清除DMA ACK信号**：对单服务模式，或者全服务模式CURR\_TC变为0

**注意**：在单服务模式下，DMAC的3个状态被执行一遍，然后停止，等待下一个DMA REQ的到来。如果DMA REQ到来，则这些状态被重复操作，直到CURR\_TC减为0

### 3、外部DMA请求/响应规则

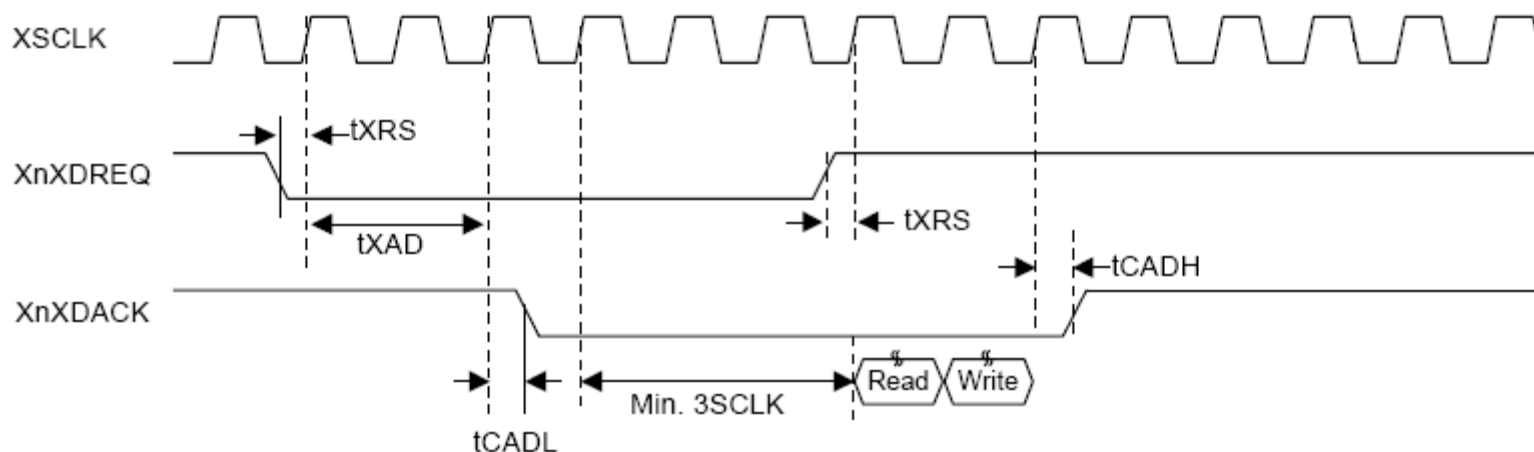
- DMAC有3种类型的外部DMA请求/响应规则：
  - single service demand, 单服务请求（对应于需求模式）
  - single service handshake, 单服务握手（握手模式）
  - whole service handshake, 全服务握手（全服务模式）

每种类型都定义了像DMA请求和DMA响应这些信号怎样与这些规则相联系

## 4、DMA时序要求

基本时序要求：

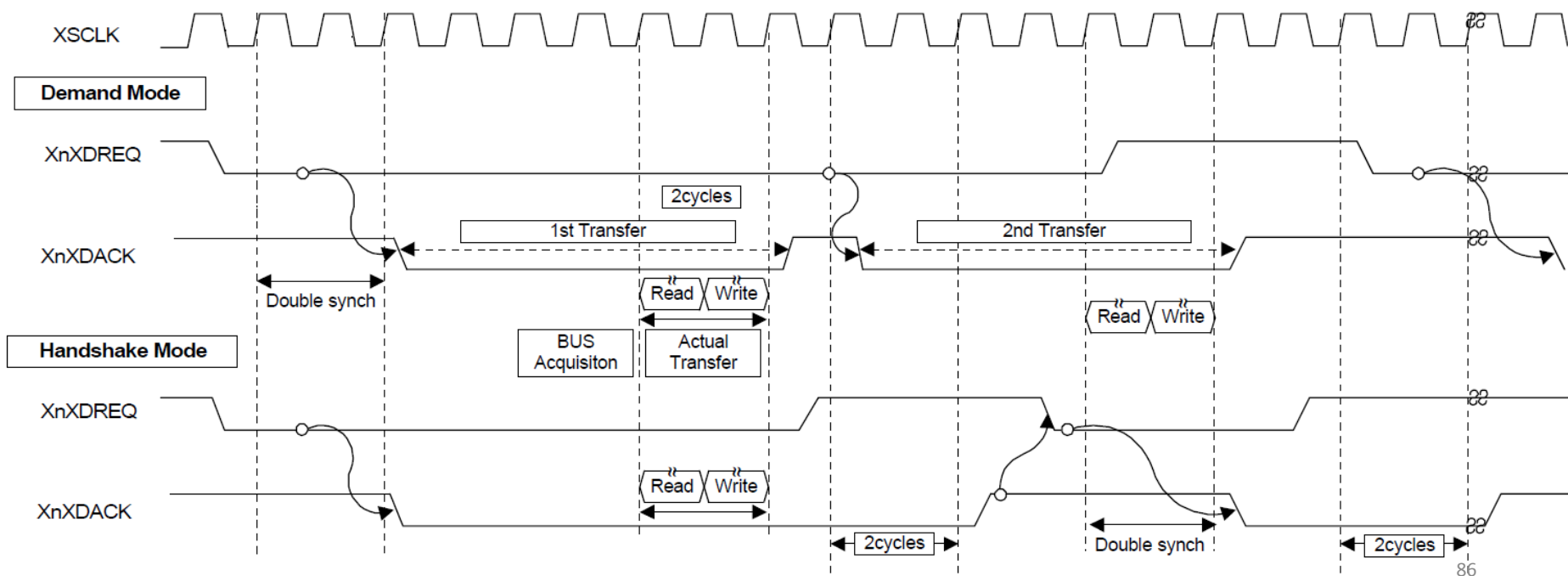
- DMA请求信号（ $xnxDREQ$ ）和响应信号（ $xnxDACK$ ）的Setup时间与delay时间在不同的模式下是不同的
- 如果DMA请求信号的setup时间满足要求，则在两个周期内实现同步，然后DMA响应信号变得有效
- 在DMA响应信号有效后，DMA向CPU请求总线。如果它得到总线就执行操作。DMA操作完成后，DMA响应信号变得无效



## 5、Demand 与 Handshake模式的比较

在一次传输结束时，DMA检查DMA请求（ $xnxDREQ$ ）信号的状态：

- demand模式：若DMA请求仍然有效，则传输马上再次开始。否则等待
- handshake模式：若DMA请求无效，DMA在两个时钟周期后将DMA响应（ $xnxDACK$ ）信号变得无效。否则，DMA等待直到DMA请求信号变得无效。每请求一次传输一次。



### 三、DMA控制器的相关寄存器

每个DMA 通道有9 个控制寄存器（4 个通道共计36 个寄存器），6 个控制DMA传输，其它3 个监视DMA 控制器的状态

Register	Address	R/W	Description	Reset Value
DISRCn	0x4B0000x0	R/W	初始源基地址寄存器	0x00000000
DISRCCn	0x4B0000x4	R/W	初始源控制寄存器	0x00000000
DIDSTn	0x4B0000x8	R/W	初始目的基地址寄存器	0x00000000
DIDSTCn	0x4B0000xC	R/W	初始目的控制寄存器	0x00000000
DCONn	0x4B0000y0	R/W	DMA控制寄存器	0x00000000
DSTATn	0x4B0000y4	R	状态/计数寄存器	0x00000000
DCSRCn	0x4B0000y8	R	当前源地址寄存器	0x00000000
DCDSTn	0x4B0000yC	R	当前目的地址寄存器	0x00000000
SKTRIGn	0x4B0000z0	R/W	DMA掩码/触发寄存器	0b000

# 1、DISRCn---DMA源基地址寄存器

原名： DMA初始源寄存器

寄存器	地 址	R/W	意 义	初 值
DISRC0	0x4B000000	R/W	DMA0源基地址寄存器	0x00000000
DISRC1	0x4B000040	R/W	DMA1源基地址寄存器	0x00000000
DISRC2	0x4B000080	R/W	DMA2源基地址寄存器	0x00000000
DISRC3	0x4B0000C0	R/W	DMA3源基地址寄存器	0x00000000

31	30	.....	0
0	S_ADDR---源数据基地址 (在CURR_SRC为0、 并且DMA ACK为1时装载入CURR_SRC)		



## 2、DISRCCn---DMA源控制寄存器

寄存器	地 址	R/W	意 义	初 值
DISRCC0	0x4B000004	R/W	DMA0初始源控制寄存器	0x00000000
DISRCC1	0x4B000044	R/W	DMA1初始源控制寄存器	0x00000000
DISRCC2	0x4B000084	R/W	DMA2初始源控制寄存器	0x00000000
DISRCC3	0x4B0000C4	R/W	DMA3初始源控制寄存器	0x00000000

31	.....	2	1	0
保留（为0）			LOC---源总线选择	INC---源地址变化设置

LOC---源所在总线选择

0: AHB; 1: APB

INC---源地址变化设置

0: 源地址增加;

1: 源地址不变

### 3、DIDSTn---DMA目的基地址寄存器

原名： DMA初始目的寄存器

寄存器	地 址	R/W	意 义	初 值
DIDST0	0x4B000008	R/W	DMA0目的基地址寄存器	0x00000000
DIDST1	0x4B000048	R/W	DMA1目的基地址寄存器	0x00000000
DIDST2	0x4B000088	R/W	DMA2目的基地址寄存器	0x00000000
DIDST3	0x4B0000C8	R/W	DMA3目的基地址寄存器	0x00000000

31	30	.....	0
0	D_ADDR---目标基地址，会被载入CURR_DST (当CURR_DST的值为0、并且 DMA ACK 的值为1时)		

## 4、DIDSTCn---DMA初始目的控制寄存器

寄存器	地 址	R/W	意 义	初 值
DIDSTC0	0x4B00000C	R/W	DMA0初始目的控制寄存器	0x00000000
DIDSTC1	0x4B00004C	R/W	DMA1初始目的控制寄存器	0x00000000
DIDSTC2	0x4B00008C	R/W	DMA2初始目的控制寄存器	0x00000000
DIDSTC3	0x4B0000CC	R/W	DMA3初始目的控制寄存器	0x00000000

31	.....	2	1	0
保留（为0）			LOC---目的总线选择	INC---目的地址变化设置

**LOC---目的地址所在总线选择**      0 : AHB ; 1 : APB

**INC---目的地址地址变化设置**

0 : 目的地址增加 ;

1 : 目的地址不变

## 5、DCONn---DMA控制寄存器

寄存器	地 址	R/W	意 义	初 值
DCON0	0x4B000010	R/W	DMA 0 控制寄存器	0x00000000
DCON1	0x4B000050	R/W	DMA 1 控制寄存器	0x00000000
DCON2	0x4B000090	R/W	DMA 2 控制寄存器	0x00000000
DCON3	0x4B0000D0	R/W	DMA 3 控制寄存器	0x00000000

31	30	29	28	27	26	25	24	23	22	21	20
DMD _HS	SYNC	INT	TSZ	SERV MODE	HWSRCSEL			SWHW _SEL	RE LOAD	DSZ	

19	18	17	16	15	14	13	12	11	10	9	8	8	7	6	5	4	3	2	1	0
TC---传输次数初值																				

31	30	29	28	27	26	25	24	23	22	21	20
DMD_HS	SYNC	INT	TSZ	SERV_MODE	HWSRCSEL			SWHW_SEL	RE_LOAD	DSZ	

### DMD\_HS---DMA与外设握手模式选择

- 0: 需求模式。为单服务，但只要DREQ信号有效便传输
- 1: 握手模式。为单服务，要等待DREQ信号变为无效，DREQ再有效时才传输。

### SYNC---DREQ 和DACK信号与系统总线时钟同步选择

- 0: DREQ和DACK与PCLK(APB clock)同步。慢速外设
- 1: DREQ和DACK与HCLK(AHB clock)同步。高速外设

### INT---CURR\_TC的中断请求控制

- 0: 禁止CURR\_TC产生中断请求
- 1: 当所有的传输结束时，CURR\_TC产生中断请求

### TSZ---传输长度类型选择

- 0: 执行单数据传输
- 1: 执行四数据长的突发传输

31	30	29	28	27	26	25	24	23	22	21	20
DMD_HS	SYNC	INT	TSZ	SERV_MODE	HWSRCSEL			SWHW_SEL	RE_LOAD	DSZ	

## SERV\_MODE---传输模式选择

0: 单服务传输模式，每传输一次都要查询DREQ

1: 全服务传输模式，不查询DREQ，但传输一次也要释放总线。

## HWSRCSEL ---各DMA通道请求源设置

HWSRCSEL	000	001	010	011	100
通道0	nXDREQ0	UART0	SDI	Timer	USB设备EP1
通道1	nXDREQ1	UART1	IISDI	SPI0	USB设备EP2
通道2	IISDO	IISDI	SDI	Timer	USB设备EP3
通道3	UART2	SDI	SPI1	Timer	USB设备EP4

31	30	29	28	27	26	25	24	23	22	21	20
DMD_HS	SYNC	INT	TSZ	SERV_MODE	HWSRCSEL			SWHW_SEL	RE_LOAD	DSZ	

## SWHW\_SEL--- DMA源选择方式（软件或硬件） 设置

0：以软件software方式产生DMA请求，需要用DMASKTRIG控制寄存器中的SW\_TRIG位设置触发。

1：由位[26:24]提供的DMA源触发DMA操作

## RELOAD---再装载选择

0：自动再装载，当传输次数减为0时自动装载DMA初值

1：不自动再装载，传输结束关闭DMA通道。

## DSZ---传输数据类型设置

00：字节；    01：半字；    10：字；    11：保留

## 6、DSTATn---DMA状态/计数寄存器

原名：DMA状态寄存器

寄存器	地 址	R/W	意 义	初 值
DSTAT0	0x4B000014	R	DMA0状态/计数寄存器	0x00000000
DSTAT1	0x4B000054	R	DMA1状态/计数寄存器	0x00000000
DSTAT2	0x4B000094	R	DMA2状态/计数寄存器	0x00000000
DSTAT3	0x4B0000D4	R	DMA3状态/计数寄存器	0x00000000

21	20	19	18	17	16	15	14	13	12	11	10	9	8	8	7	6	5	4	3	2	1	0
STAT		CURRTC---当前传输次数计数值																				

**STAT---DMA状态**

00：就绪态，可进行传输；

01：DMA正在传输；

1X：保留

**CURRTC---当前传输计数值**

每传输一次其值减1。其初值在DCONn中低20位。



## 7、DCSRCn---DMA当前源地址寄存器

寄存器	地 址	R/W	意 义	初 值
DCSRC0	0x4B000018	R	DMA0当前源地址寄存器	0x00000000
DCSRC1	0x4B000058	R	DMA1当前源地址寄存器	0x00000000
DCSRC2	0x4B000098	R	DMA2当前源地址寄存器	0x00000000
DCSRC3	0x4B0000D8	R	DMA3当前源地址寄存器	0x00000000

31	30	.....	0
0	CURR_SRC---当前数据源地址		

### CURR\_SRC---当前数据源地址

注意：（1）DMA每传输一次，其地址可能增加(1、2、4)、可能不变；（2）在CURR\_SRC为0、且DMA ACK为1时，将S\_ADDR源基地址的值装入。

## 8、DCDSTn---DMA当前目的地址寄存器

寄存器	地 址	R/W	意 义	初 值
DCDST0	0x4B00001C	R	DMA0当前目的地址寄存器	0x00000000
DCDST1	0x4B00005C	R	DMA1当前目的地址寄存器	0x00000000
DCDST2	0x4B00009C	R	DMA2当前目的地址寄存器	0x00000000
DCDST3	0x4B0000DC	R	DMA3当前目的地址寄存器	0x00000000

31	30	.....	0
0	CURR_DST---当前数据目的地址		

### CURR\_DST---当前数据目的地址

注意： (1) DMA每传输一次，其地址可能增加(1、2、4)  
 (2) 在CURR\_DST为0、且DMA ACK为1时，将  
 D\_ADDR的值装入

## 9、DMASKTRIGn---DMA掩码(Mask)触发寄存器

寄存器	地 址	R/W	意 义	初 值
DMASKTRIG0	0x4B000020	R/W	DMA0掩码触发寄存器	0x00000000
DMASKTRIG1	0x4B000060	R/W	DMA1掩码触发寄存器	0x00000000
DMASKTRIG2	0x4B0000A0	R/W	DMA2掩码触发寄存器	0x00000000
DMASKTRIG3	0x4B0000E0	R/W	DMA3掩码触发寄存器	0x00000000

31	.....	3	2	1	0
保留 (为0)			STOP	ON/OFF	SW_TRIG

## STOP---DMA运行停止位

1: DMA将当前数据传输完立即停止, 并且CURR\_TC变为0。

**注意:** 如果ON/OFF设置为OFF, 则DMA也停止传输。

## ON/OFF---DMA通道屏蔽位

0: 关闭通道;          1: 开放通道

如果DCONn[22]设为非自动重装, DMA则传输完成后STOP位置1、并且关闭通道。

**注意:** 在DMA运行期间, 不要改变其值, 并且也不要使用该位停止DMA传输, 正确的方法应该使用STOP位。

## SW\_TRIG: DMA软件触发位

设为1时, 实现软件触发DMA请求。

**注意:** 只有当DCONn[23]设为软件触发DMA请求时, 其软件触发才有效

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

# 8 A/D转换与触摸屏

---

## 主要内容

- 1、概述
- 2、结构与工作原理
- 3、寄存器
- 4、应用举例

重点：

- (1) 触摸屏测量原理

# 一、S3C2410X的A/D 转换器概述

**S3C2410X中集成了一个 8 通道10 位A/D 转换器，A/D 转换器自身具有采样保持功能。并且S3C2410X的A/D 转换器支持触摸屏接口。**

## **A/D转换器的主要特性：**

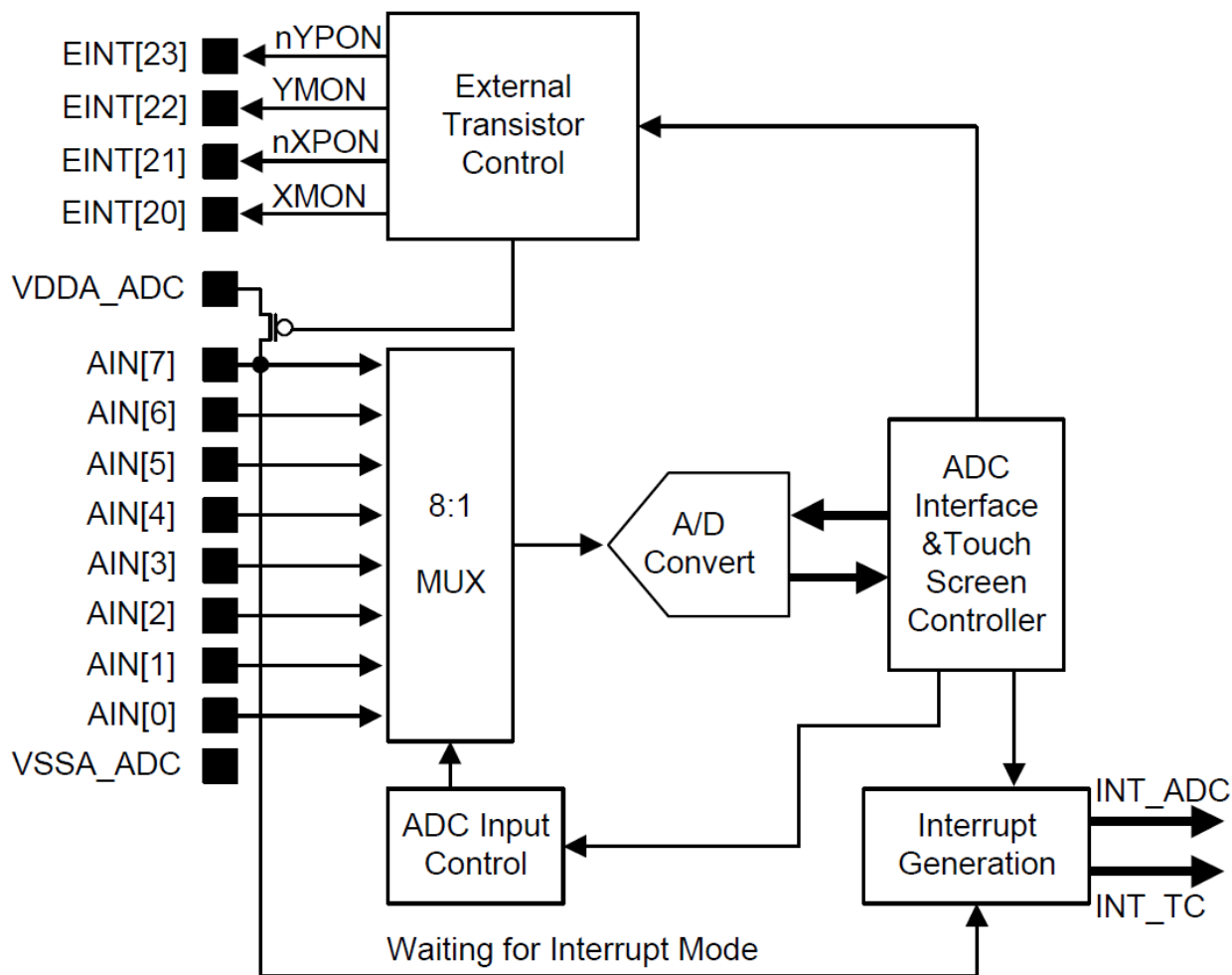
- 分辨率：10位；精度： $\pm 1\text{LSB}$
- 线性度误差： $\pm 1.5\text{---}2.0\text{LSB}$ ；
- 最大转换速率：500KSPS；
- 输入电压范围：0~3.3v；
- 系统具有采样保持功能；
- 常规转换和低能源消耗功能；
- 独立/自动的X/Y 坐标转换模式。

## 二、A/D转换器结构与工作原理

### 1、结构

主要由6部分构成：

- 信号输入通道
- 8转1切换开关
- A/D转换器
- 控制逻辑
- 中断信号发生器
- 触摸屏接口





## 2、工作原理

### (1) A/D 转换时间计算和分辨率

当PCLK 频率为50MHz，预分频值是49，10 位数字量的转换时间如下：

$$\text{A/D 转换频率} = 50\text{MHz} / (49+1) = 1\text{MHz}$$

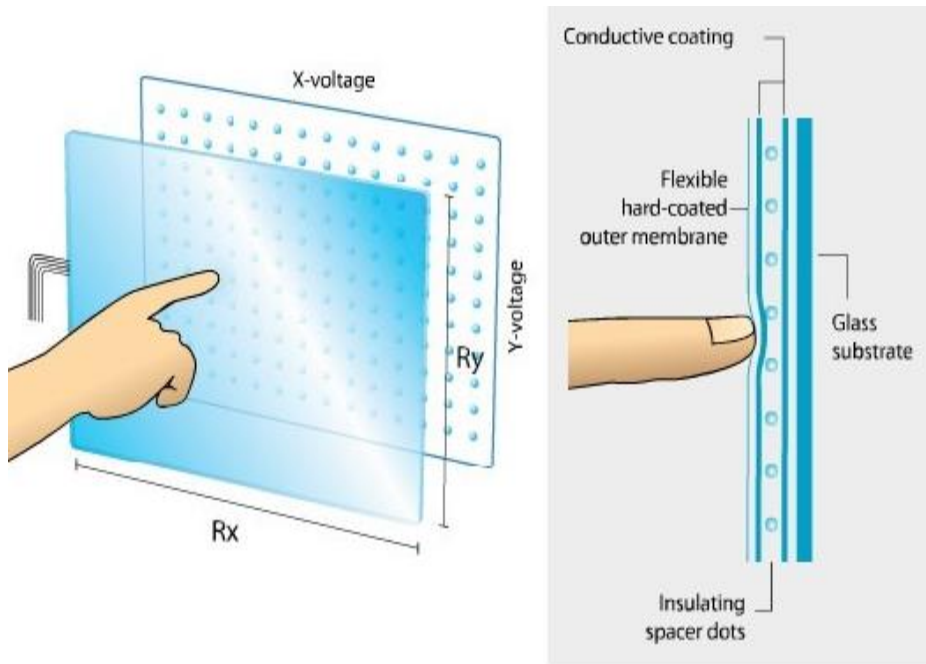
$$\text{转换时间} = 1 / (1\text{MHz} / 5 \text{ 个周期}) = 1 / 200\text{KHz} = 5\mu\text{s}$$

A/D 转换器最大可以工作在2.5MHz 时钟下，所以转换速率可以达到500KSPS

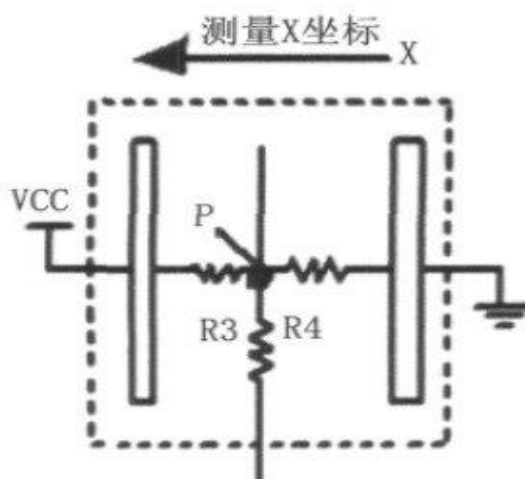
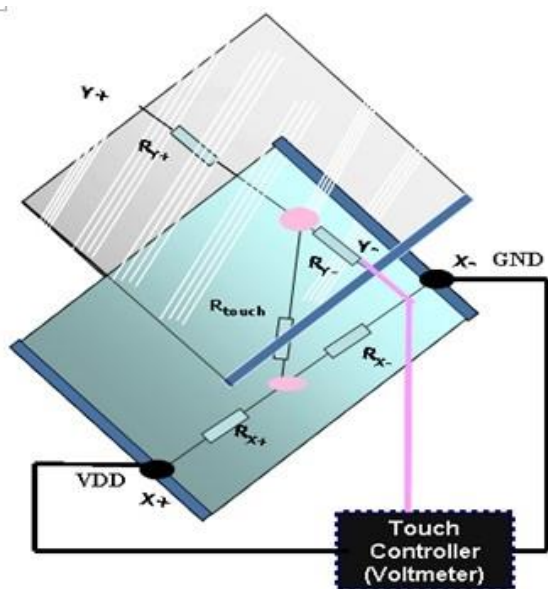
## 2、工作原理

### (2) 触摸屏的结构及工作原理

**原理：**对于电阻式触摸屏，由3层透明薄膜构成，有一层是电阻层，还有一层是导电层，它们中间有一隔离层，当某一点被按压时，在按压点电阻层与导电层接触，如果在电阻层的一边接电源，另一边接地，便可测量出按压点的电压，从而可算出其坐标。

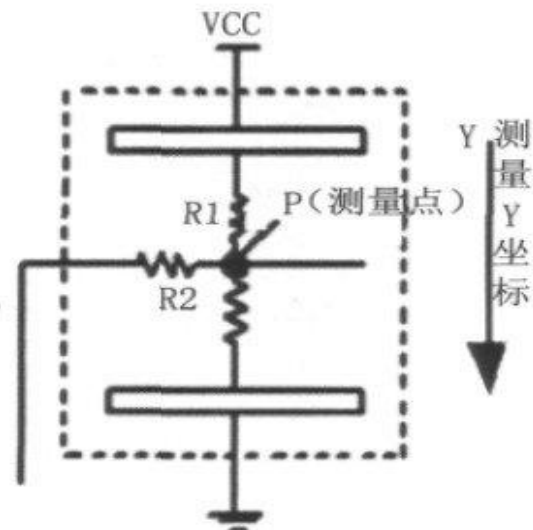


## 2、工作原理



模拟电压测量点  
(Y+电极)

$$V_{PX} = R4 \times V / (R3 + R4)$$



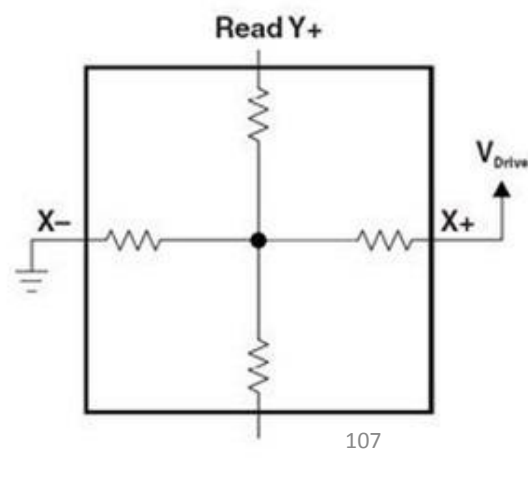
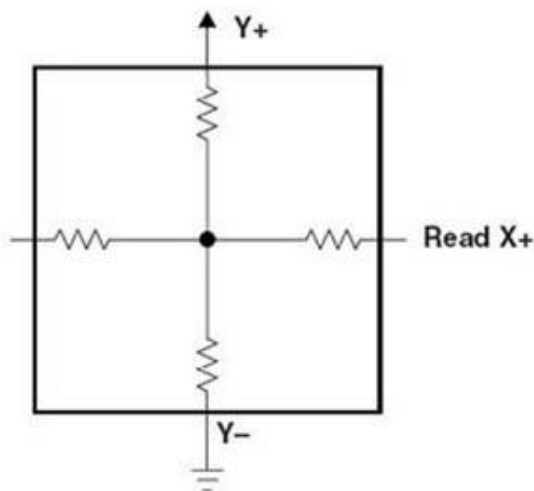
模拟电压测量点  
(X+电极)

$$V_{PY} = R1 \times V / (R1 + R2)$$

newmaker

$$y = \frac{V_{Y+}}{V_{Drive}} \times \text{height}_{screen}$$

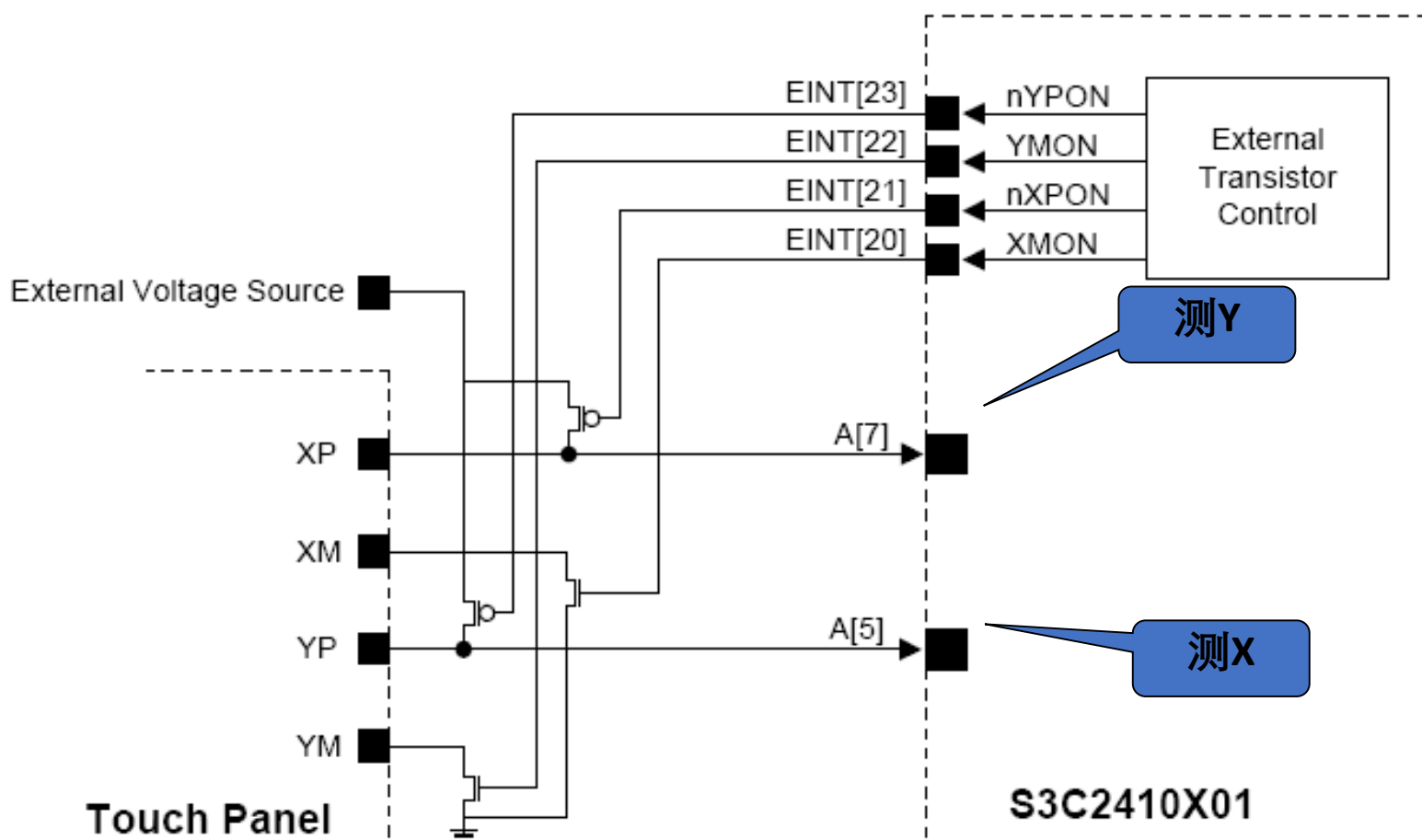
$$x = \frac{V_{X+}}{V_{Drive}} \times \text{width}_{screen}$$



**测量X坐标：** 从XP输出电压给X+端，从XM输出地电位给X-端；从YP脚输入按压点电压。

控制信号： nYPON=1； nYMON=0

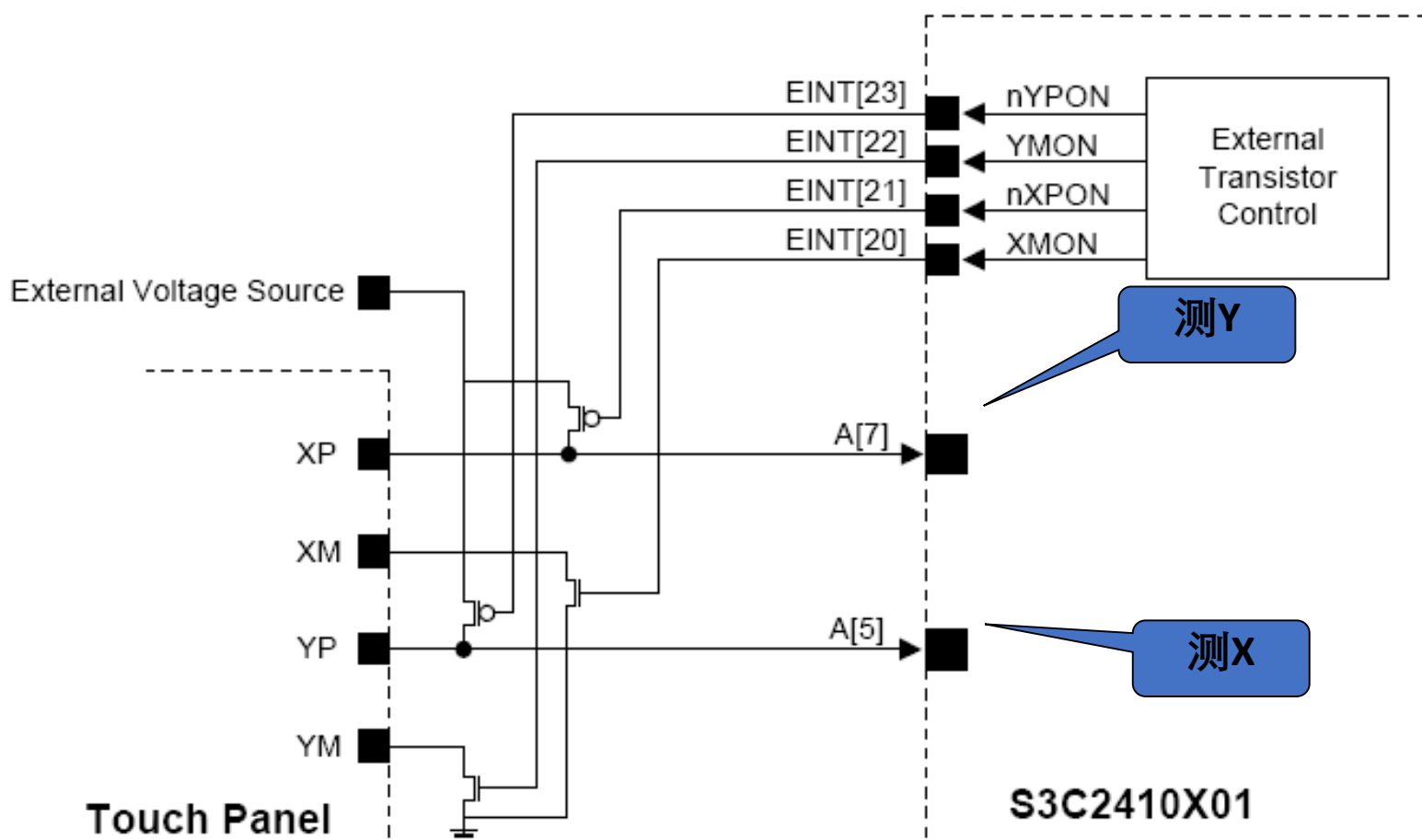
nXPON=0； nXMON=1



**测量Y坐标：**从YP输出电压给Y+端，从YM输出地电位给Y-端；从XP脚输入按压点电压。

控制信号：nYPON=0；nYMON=1

nXPON=1；nXMON=0



### 3、S3C24120X A/D转换器的的工作模式

5种模式：普通转换模式、分离的X/Y坐标转换模式、连续的X/Y坐标转换模式、等待中断模式、静态模式。第2---4种是用于触摸屏。

#### (1) 普通转换模式

用于一般A/D转换，不是用于触摸屏。转换结束后，其数据在ADCDAT0中的XPDATA域

#### (2) 分离的X/Y坐标转换模式

分两步进行X/Y坐标转换，其转换结果分别存于ADCDAT0中的XPDATA域中和ADCDAT1中的YPDATA域中，并且均会产生INT\_ADC中断请求

#### (3) 自动（连续）的X/Y坐标转换模式

X坐标转换结束启动Y坐标转换，其转换结果分别存于ADCDAT0中的XPDATA域中和ADCDAT1中的YPDATA域中，然后产生INT\_ADC中断请求

### 3、S3C24120X A/D转换器的的工作模式

#### (4) 等待中断转换模式

在该模式下，转换器等待使用者按压触摸屏，一旦触摸屏被按压，则产生INT\_TC触摸屏中断请求。中断后，在中断处理程序中再将转换器设置为分离的X/Y坐标转换模式、或者连续的X/Y坐标转换模式进行处理。

触摸屏接口信号：

XP=上拉

XM=高阻

YP=AIN[5]

YM=接地

#### (5) 静态模式

当ADCCON中的STDBM设为1时，转换器进入静态模式，停止A/D转换。其数据域的数据保持不变。

## 三、ADC和触摸屏专用寄存器

有5个专用寄存器

Register	Address	R/W	Description	Reset Value
ADCCON	0x58000000	R/W	ADC控制寄存器	0x3FC4
ADCTSC	0x58000004	R/W	触摸屏控制寄存器	0x058
ADCDLY	0x58000008	R/W	ADC起始延迟寄存器	0x00FF
ADCDAT0	0x5800000C	R	ADC转换数据0寄存器	-
ADCDAT1	0x58000010	R	ADC转换数据1寄存器	-



# 1、ADCCON---ADC控制寄存器

15	14	13	.....	6
ECFLG	PRSCEN	PRSCVL		

5	4	3	2	1	0
SEL_MUX			STDBM	READ_START	ENABLE_START

**ECFLG---转换结束标志（只读）**

0：转换操作中；

1：转换结束

**PRSCEN---转换器预分频器使能**

0：停止预分频器；

1：使能预分频器

**PRSCVL---转换器预分频器数值 数值N范围：1---255**

注意：（1）实际除数值为N+1

（2）对N数值的要求：转换速率应该 $<PCLK/5$

5	4	3	2	1	0
SEL_MUX			STDBM	READ_START	ENABLE_START

### SEL\_MUX ---模拟输入通道选择

000: AIN0;                      001: AIN1                      010: AIN2  
011: AIN3                      .....                      111: AIN7

### STDBM---备用模式设置

0: 正常工作模式;                      1: 备用模式, 不做A/D转换

### READ\_START---通过读取启动转换

0: 停止通过读取启动转换; 1: 使能通过读取启动转换

### ENABLE\_START---通过设置该位启动转换

0: 无效;                      1: 启动A/D转换 (启动后被清0)

注意: 如果READ\_START为1, 则该位无效

## 2、ADCTSC---ADC触摸屏控制寄存器

8	7	6	5	4	3	2	1	0
保留0	YM_SEN	YP_SEN	XM_SEN	XP_SEN	PULL_UP	AUTO_PST	XY_PST	

**YM\_SEN---选择YMON的输出值**

0: 输出0 (YM=高阻) ; 1: 输出1 (YM=GND)

**YP\_SEN---选择nYPON的输出值**

0: 输出0 (YP=外部电压) ; 1: 输出1 (YP连接  
AIN[5])

**XM\_SEN---选择XMON的输出值**

0: 输出0 (XM=高阻) ; 1: 输出1 (XM=GND)

**XP\_SEN---选择nXP的输出值**

0: 输出0 (XP=外部电压) ; 1: 输出1 (XP连接  
AIN[7])

8	7	6	5	4	3	2	1	0
保留0	YM_SEN	YP_SEN	XM_SEN	XP_SEN	PULL_UP	AUTO_PST	XY_PST	

### **PULL---**上拉切换使能

**0:** XP上拉使能;

**1:** XP上拉禁止

### **AUTO\_PST---**自动连续转换X轴和Y轴坐标模式选择

**0:** 普通A/D转换;

**1:** 连续X/Y轴转换模式

### **XY\_PST---**手动测量X轴和Y轴坐标模式选择

**00:** 无操作模式;

**01:** 对X坐标测量;

**10:** 对Y坐标测量;

**11:** 等待中断模式

### 3、ADCDLY---ADC起始延迟寄存器

31	.....	16	15	14	13	12	11	10	9	8	8	7	6	5	4	3	2	1	0
保留为0			起始延迟数值---分两种情况																

#### 第一情况：

对普通转换模式、分离的X/Y轴坐标转换模式、连续的X/Y轴坐标转换模式，**为转换延时数值。**

#### 第二种情况：

对中断转换模式，为按压触摸屏后到产生中断请求的延迟时间数值，**其时间单位为ms。**

## 4、ADCDAT0---ADC转换数据0寄存器

第11次到此

15	14	13	12	11	10	9	.....	0
UPDOWN	AUTO_PST	XY_PST	保留 (0)			XPDATA或普通ADC值		

### UPDOWN---等待中断模式的按压状态

0: 触笔点击;

1: 触笔提起

### AUTO\_PST---自动X/Y轴转换模式指示

0: 普通转换模式;

1: X/Y轴坐标连续转换

### XY\_PST---手动X/Y轴转换模式指示

00: 无操作;

01: 为X轴坐标转换

10: 为Y轴坐标转换

11: 为等待中断转换

XPDATA[9: 0]: 为X轴坐标转换数值、或普通ADC转换数值  
具体意义由其它位指示。其值为: 0---0x3FF

## 5、ADC DAT1---ADC转换数据1寄存器

15	14	13	12	11	10	9	.....	0
UPDOWN	AUTO_PST	XY_PST	保留 (0)			YPDATA		

**UPDOWN---等待中断模式的按压状态**

0: 触笔点击;

1: 触笔提起

**AUTO\_PST---自动X/Y轴转换模式指示**

0: 普通转换模式;

1: X/Y轴坐标连续转换

**XY\_PST---手动X/Y轴转换模式指示**

00: 无操作;

01: 为X轴坐标转换

11: 为Y轴坐标转换

11: 为等待中断转换

**YPDATA[9: 0]: 为10位Y轴坐标转换结果**

其值为: 0---0x3FF

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器



# 9 中断

---

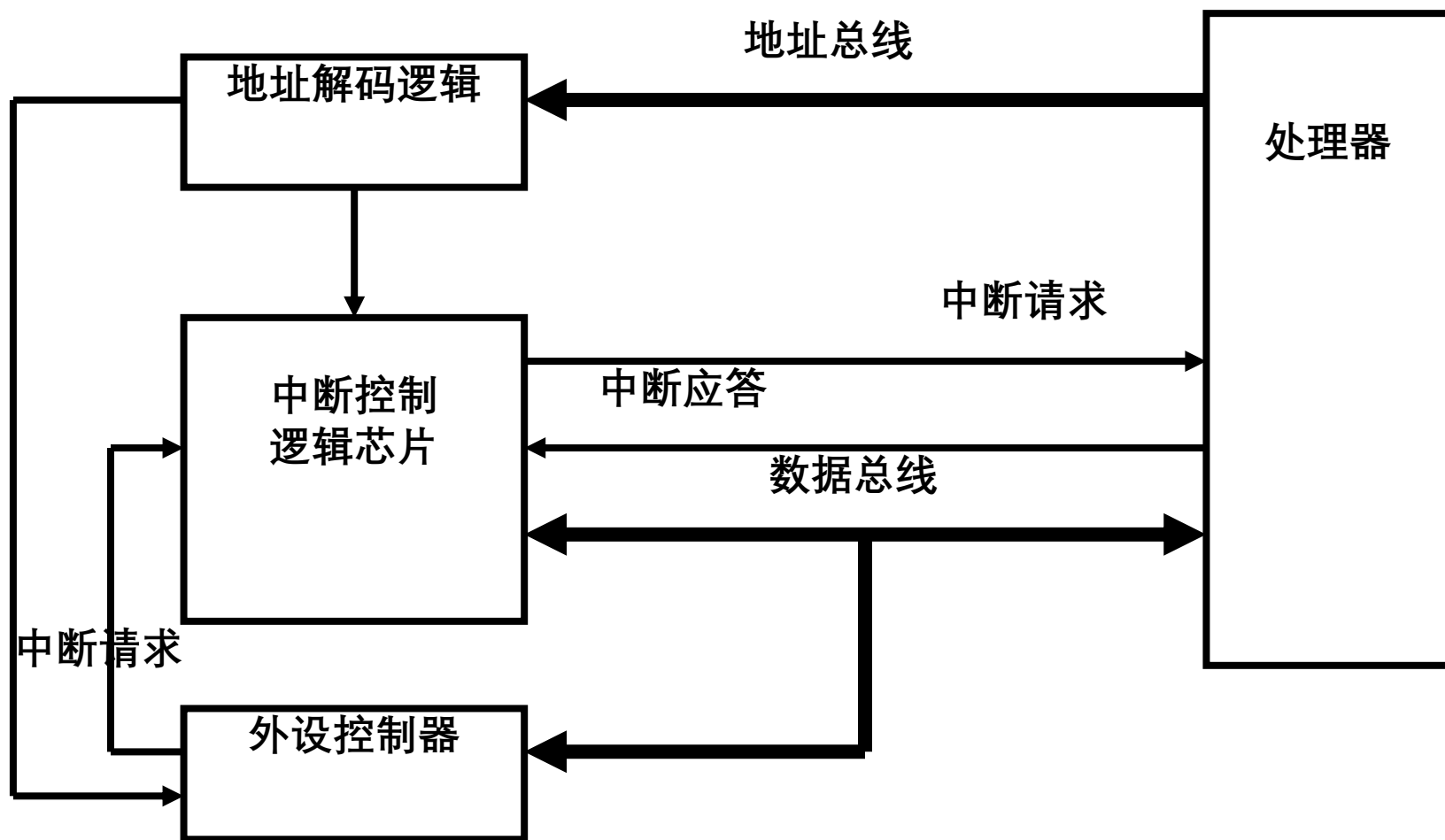
## 主要内容

1. 概述
2. 结构与工作原理
3. 寄存器
4. 应用举例

# 一、概 述

- S3C2410X中断控制器有56个中断源，对外提供24个外中断输入引脚，内部所有设备都有中断请求信号，例如DMA控制器、UART、IIC等等
- S3C2410X的ARM920T内核有两个中断，IRQ中断和快速中断FIQ（FIQ用于进行大批量的数据传输）
- 中断仲裁：当中断控制器接收到多个中断请求时，其内的优先级仲裁器裁决后向CPU发出优先级最高的中断请求信号或快速中断请求信号

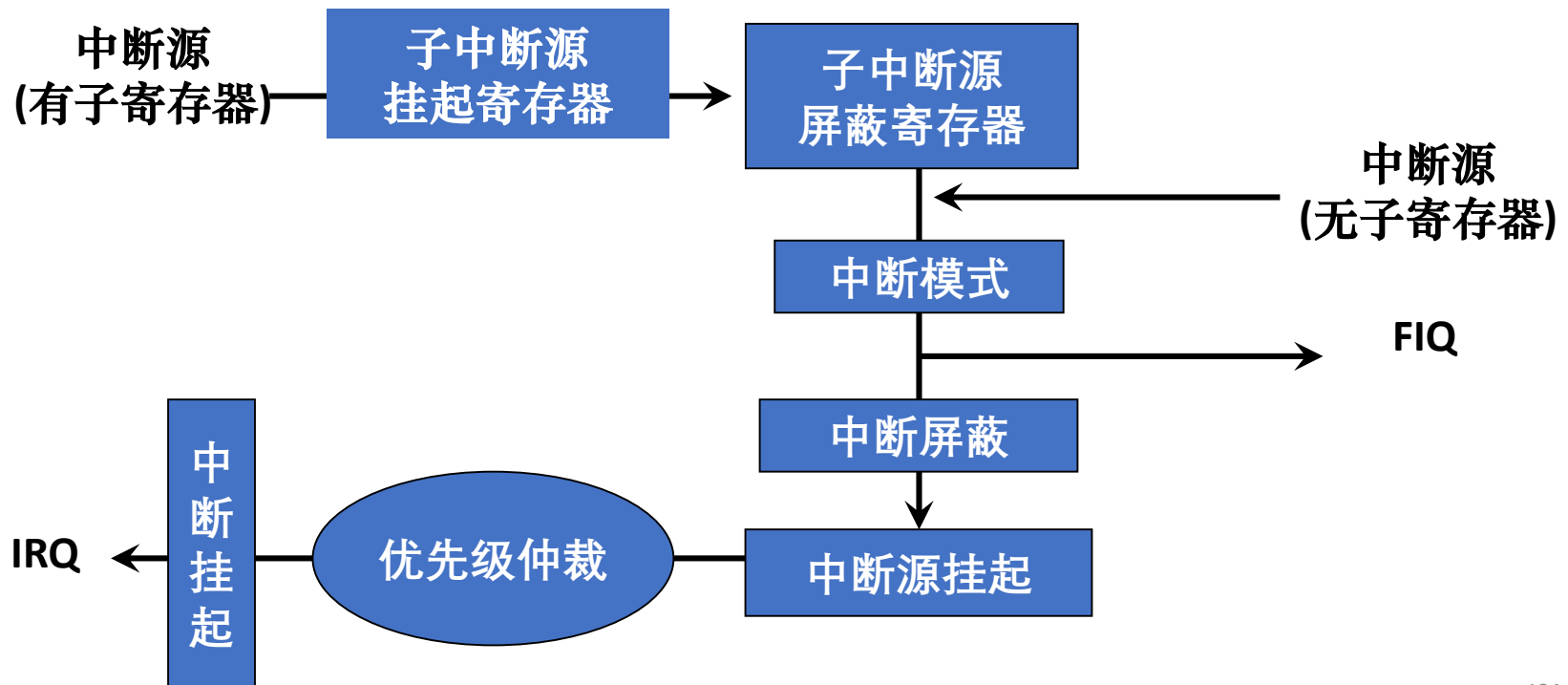
# 中断的工作过程



## 二、S3C2410X中断系统结构

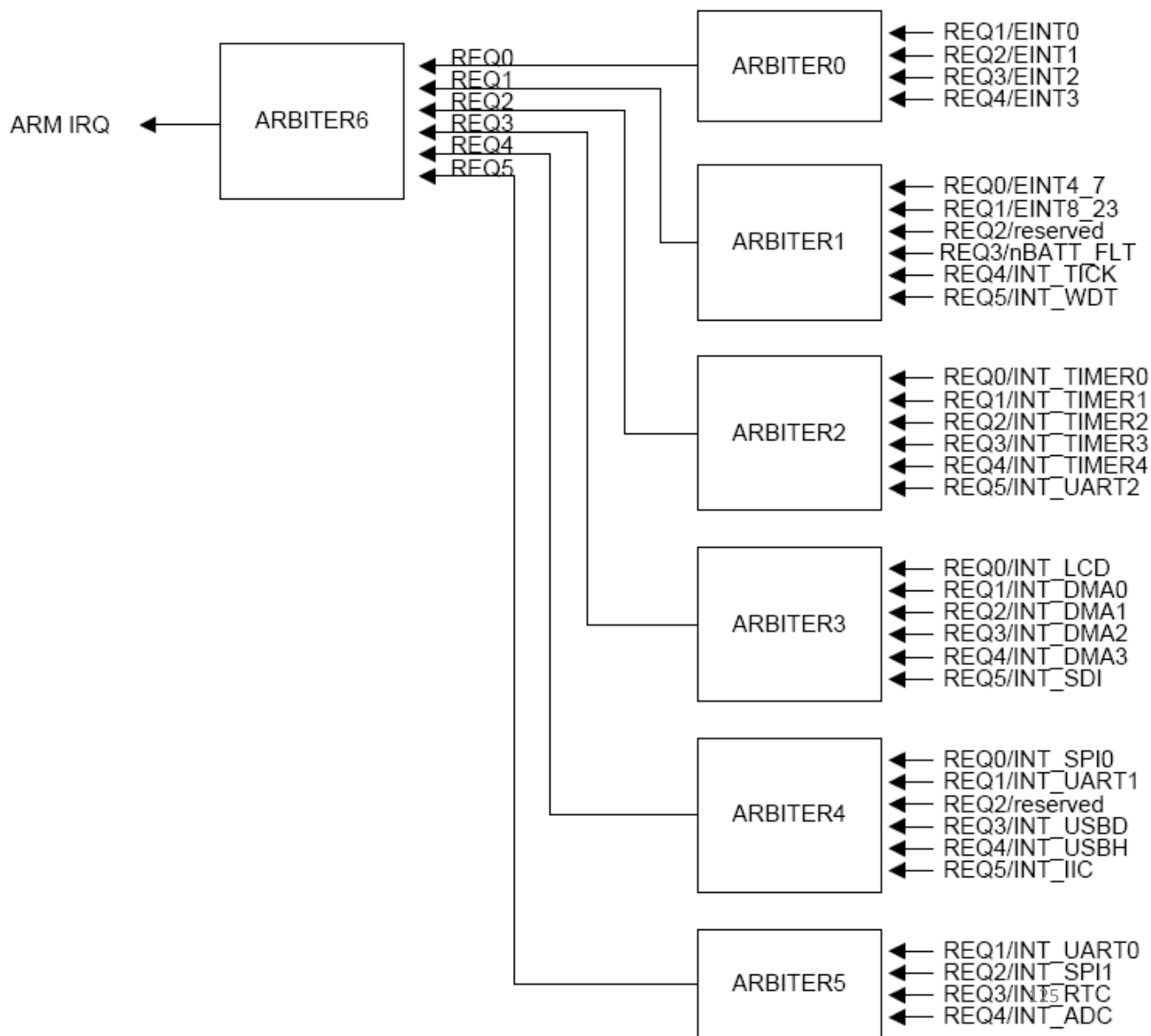
### 1、中断系统结构

主要由中断源和控制寄存器两大部分构成，其寄存器主要有4种：模式、屏蔽、优先级、挂起（标志）寄存器等



## 2、中断优先级仲裁器及工作原理

中断系统有6个分仲裁器和1个总仲裁器，每一个仲裁器可以处理6路中断。



## 三、中断控制器专用寄存器

有8个专用寄存器

Register	Address	R/W	Description	Reset Value
SRCPND	0x4A000000	R/W	中断标志寄存器	0x00000000
INTMOD	0x4A000004	R/W	中断模式寄存器	0x00000000
INTMSK	0x4A000008	R/W	中断屏蔽寄存器	0xFFFFFFFF
PRIORITY	0x4A00000C	R/W	中断优先级寄存器	0x7F
INTPND	0x4A000010	R/W	中断服务寄存器	0x00000000
INTOFFSET	0x4A000014	R	中断偏移寄存器	0x00000000
SUBSRCPND	0x4A000018	R/W	子源挂起寄存器	0x00000000
INTSUBMSK	0x4A00001C	R/W	中断子源屏蔽寄存器	0x7FF

主要使用前5个寄存器

# 1、SRCPND---中断源挂起（标志）寄存器

位号	中断源	位号	中断源	位号	中断源	位号	中断源
31	INT_ADC	23	INT_UART1	15	INT_UART2	7	nBATT_FLT
30	INT_RTC	22	INT_SPI0	14	INT_TIM4	6	保留
29	INT_SPI1	21	INT_SDI	13	INT_TIM3	5	EINT8_23
28	INT_UART0	20	INT_DMA3	12	INT_TIM2	4	EINT4_7
27	INT_IIC	19	INT_DMA2	11	INT_TIM1	3	EINT3
26	INT_USBH	18	INT_DMA1	10	INT_TIM0	2	EINT2
25	INT_USBD	17	INT_DMA0	9	INT_WDT	1	EINT1
24	保留	16	INT_LCD	8	INT_TICK	0	EINT0

该寄存器也就是中断标志寄存器

各位：1：对应中断源有中断请求

0：对应中断源无中断请求

注意：必须在中断处理程序中对其标志位清0。其方法为写1。

## 2、INTMOD---中断模式寄存器

位号	中断源	位号	中断源	位号	中断源	位号	中断源
31	INT_ADC	23	INT_UART1	15	INT_UART2	7	nBATT_FLT
30	INT_RTC	22	INT_SPI0	14	INT_TIM4	6	保留
29	INT_SPI1	21	INT_SDI	13	INT_TIM3	5	EINT8_23
28	INT_UART0	20	INT_DMA3	12	INT_TIM2	4	EINT4_7
27	INT_IIC	19	INT_DMA2	11	INT_TIM1	3	EINT3
26	INT_USBH	18	INT_DMA1	10	INT_TIM0	2	EINT2
25	INT_USBD	17	INT_DMA0	9	INT_WDT	1	EINT1
24	保留	16	INT_LCD	8	INT_TICK	0	EINT0

该寄存器是设置各中断源是FIQ中断还是IRQ中断

各位：

- 1：对应中断源设为FIQ中断模式
- 0：对应中断源设为IRQ中断模式



### 3、INTMSK---中断屏蔽寄存器

位号	中断源	位号	中断源	位号	中断源	位号	中断源
31	INT_ADC	23	INT_UART1	15	INT_UART2	7	nBATT_FLT
30	INT_RTC	22	INT_SPI0	14	INT_TIM4	6	保留
29	INT_SPI1	21	INT_SDI	13	INT_TIM3	5	EINT8_23
28	INT_UART0	20	INT_DMA3	12	INT_TIM2	4	EINT4_7
27	INT_IIC	19	INT_DMA2	11	INT_TIM1	3	EINT3
26	INT_USBH	18	INT_DMA1	10	INT_TIM0	2	EINT2
25	INT_USBD	17	INT_DMA0	9	INT_WDT	1	EINT1
24	保留	16	INT_LCD	8	INT_TICK	0	EINT0

各位：  
1：屏蔽对应中断源  
0：开放对应中断源

## 4、PRIORITY---中断优先级寄存器

位号	含 义	位号	含 义	位号	含 义
31:21	保 留	12:11	ARB_SEL2	4	ARB_MODE4
20:19	ARB_SEL6	10:9	ARB_SEL1	3	ARB_MODE3
18:17	ARB_SEL5	8:7	ARB_SEL0	2	ARB_MODE2
16:15	ARB_SEL4	6	ARB_MODE6	1	ARB_MODE1
14:13	ARB_SEL3	5	ARB_MODE5	0	ARB_MODE0

### ARB\_SELn---n组优先级顺序控制位

00: REQ0, 1, 2, 3, 4, 5

01: REQ0, 2, 3, 4, 1, 5

10: REQ0, 3, 4, 1, 2, 5

11: REQ0, 4, 1, 2, 3, 5

### ARB\_MODEn---n组优先级循环控制位

0: 优先顺序固定不变

1: 优先顺序循环，每响应一次中断，其顺序循环改变一次，但REQ0、REQ5位置不变。

## 5、INTPND---中断服务（挂起）寄存器

位号	中断源	位号	中断源	位号	中断源	位号	中断源
31	INT_ADC	23	INT_UART1	15	INT_UART2	7	nBATT_FLT
30	INT_RTC	22	INT_SPI0	14	INT_TIM4	6	保留
29	INT_SPI1	21	INT_SDI	13	INT_TIM3	5	EINT8_23
28	INT_UART0	20	INT_DMA3	12	INT_TIM2	4	EINT4_7
27	INT_IIC	19	INT_DMA2	11	INT_TIM1	3	EINT3
26	INT_USBH	18	INT_DMA1	10	INT_TIM0	2	EINT2
25	INT_USBD	17	INT_DMA0	9	INT_WDT	1	EINT1
24	保留	16	INT_LCD	8	INT_TICK	0	EINT0

各位：           1：对应的中断源被响应，且正在执行中断服务  
                  0：对应中断源未被响应

注意：必须在中断处理程序中对其服务标志位清0。方法为对某位写1便清除为0。

即在清除SRCPND中相应位后，要清除该寄存器相应位。

## 6、INTOFFSET---中断偏移寄存器

中断源	偏移值	中断源	偏移值	中断源	偏移值	中断源	偏移值
INT_ADC	31	INT_UART1	23	INT_UART2	15	nBATT_FLT	7
INT_RTC	30	INT_SPI0	22	INT_TIM4	14	保留	6
INT_SPI1	29	INT_SDI	21	INT_TIM3	13	EINT8_23	5
INT_UART0	28	INT_DMA3	20	INT_TIM2	12	EINT4_7	4
INT_IIC	27	INT_DMA2	19	INT_TIM1	11	EINT3	3
INT_USBH	26	INT_DMA1	18	INT_TIM0	10	EINT2	2
INT_USBD	25	INT_DMA0	17	INT_WDT	9	EINT1	1
保留	24	INT_LCD	16	INT_TICK	8	EINT0	0

该寄存器的偏移值指示在INTPND中显示的中断源

各位： 1： 对应的中断源，在INTPND中被置位

**说明：**当在中断服务程序中对SRCPND、INTPND中的标志位清0时，该寄存器的对应位自动清0。

## 7、SUBSRCPND---子中断源请求标志寄存器

位号	中断源	位号	中断源	位号	中断源
31:11	保 留	7	INT_TXD2	3	INT_RXD1
10	INT_ADC	6	INT_RXD2	2	INT_ERR0
9	INT_TC	5	INT_ERR1	1	INT_TXD0
8	INT_ERR2	4	INT_TXD1	0	INT_RXD0

对有多个中断源的外设，显示其具体的中断请求

各位：           1： 对应的子中断源有请求  
                  0： 对应的子中断源无请求

**注意：** 在中断服务程序中，需要对其置1的标志位清0。

## 8、INTSUBMSK---子中断源屏蔽寄存器

位号	中断源	位号	中断源	位号	中断源
31:11	保 留	7	INT_TXD2	3	INT_RXD1
10	INT_ADC	6	INT_RXD2	2	INT_ERR0
9	INT_TC	5	INT_ERR1	1	INT_TXD0
8	INT_ERR2	4	INT_TXD1	0	INT_RXD0

对有多个中断源的外设，对具体的中断源进行屏蔽

各位：  
1：屏蔽对应的子中断源  
0：开放对应的子中断源



Thank you