

## 第五章 存储器控制器

### 5.1 概述

s3c2440A 的存储器控制器提供访问外部存储器所需的存储器控制信号。

s3c2440A 的存储器控制器有以下特性：

- 大小端（通过软件选择）
- 地址空间：每个 bank 有 128M 的字节（总共 1G 字节 /8 个 banks）
- 可编程的访问位宽，bank0(16/32 位)，其他 bank（8/16/32 位）
- 共 8 个存储器 banks
  - 6 个是 ROM，SRAM 等类型存储器 bank
  - 2 个是可以作为 ROM、SRAM、SDRAM 等存储器 bank
- 7 个固定的存储器 bank 起始地址
- 最后一个 bank 的起始地址可调整
- 最后两个 bank 大小可编程
- 所有存储器 bank 的访问周期可编程
- 总线访问周期可通过插入外部 wait 来延长
- 支持 SDRAM 的自刷新和掉电模式

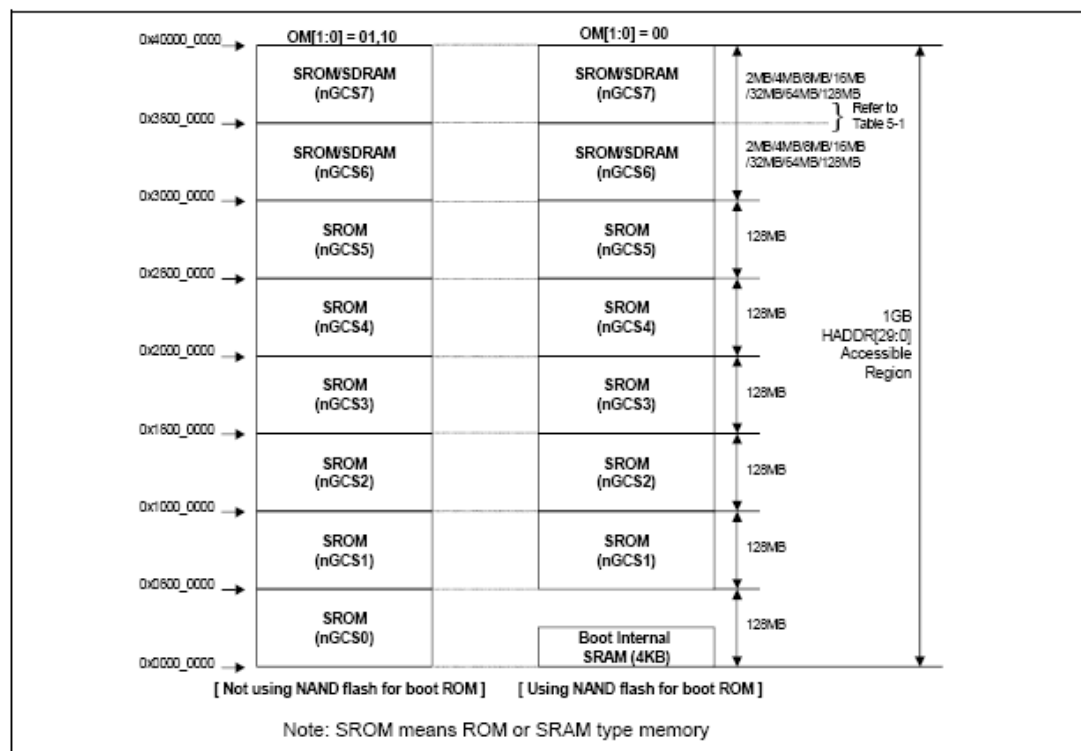


Figure 5-1. S3C2440A Memory Map after Reset

地址	2MB	4MB	8MB	16MB	32MB	64MB	128MB
<b>Bank6</b>							
起始地址	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000
结束地址	0x301f_ffff	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff
<b>Bank7</b>							
起始地址	0x3020_0000	0x3040_0000	0x3080_0000	0x3100_0000	0x3200_0000	0x3400_0000	0x3800_0000
结束地址	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff	0x3fff_ffff

## 5.2 功能描述

### 5.2.1 BANK0 总线宽度

BANK0 的数据总线 (nGCS0) 应该被配置为 16 位和 32 位中的一个。因为 BANK0 作为启动 ROM bank 工作时, BANK 的总线宽度应该在第一次 ROM 访问之前被决定, 其依赖于 OM[0:1]在重启时的逻辑电平。

OM1(操作模式 1)	OM2 (操作模式 2)	启动 ROM 数据宽度
0	0	Nand Flash 模式
0	1	16 位
1	0	32 位
1	1	测试模式

### 存储器 (SRAM/SDRAM) 地址引脚连接

存储器地址引脚	S3c2440A 地址 @8 位数据总线	S3c2440A 地址 @16 位数据总线	S3c2440A 地址 @32 位数据总线
A0	A0	A1	A1
A1	A1	A2	A2
...	...	...	...

### 存储器 (SDRAM) 地址引脚连接

Bank 大小	总线宽度	基本单元	存储器配置	Bank 地址
2MByte	x8	16Mbit	(1M x 8 x 2Bank) x 1	A20
	x16		(512K x 16 x 2B) x 1	
4MB	x16		(1M x 8 x 2B) x 2	A21
	x16		(1Mx 8 x 2B) x 2	
8MB	x16	16Mb	(2M x 4 x 2B) x 4	A22
	x32		(1Mx 8x 2B) x 4	
	x8	64Mb	(4Mx 8 x 2B) x 1	A[22:21]
	x8		(2M x 8 x 4B) x 1	
	x16		(2M x 16 x 2B) x 1	A22
	x16		(1M x 16 x 4B) x 1	A[22:21]
	x32		(512K x 32 x 4B) x 1	
16MB	x32	16Mb	(2Mx 4 x 2B) x 8	A23
	x8	64Mb	(8Mx 4 x 2B) x 2	
	x8		(4Mx 4 x 4B) x 2	A[23:22]
	x16		(4Mx 8 x 2B) x 2	A23
	x16		(2Mx 8 x 4B) x 2	A[23:22]
	x32		(2Mx 16 x 2B) x 2	A23
	x32		(1Mx 16x 4B) x 2	A[23:22]
	x8	128Mb	(4Mx 8 x 4B) x 1	
	x16		(2Mx 16 x 4B) x 1	
32MB	x16	64Mb	(8Mx 4 x 2B) x 4	A24
	x16		(4Mx 4 x 4B) x 4	A[24:23]
	x32		(4Mx 8 x 2B) x 4	A24
	x32		(2Mx 8 x 4B) x 4	A[24:23]
	x16	128Mb	(4Mx 8 x 4B) x 2	
	x32		(2Mx 16 x 4B) x 2	
	X8	256Mb	(8Mx 8x 4B) x 1	
	x16		(4Mx 16x 4B) x 1	
64MB	x32	128Mb	(4M x 8 x 4B) x 4	A[25:24]
	x16	256Mb	(8M x 8 x 4B) x 2	
	x32		(4M x 16 x 4B) x 2	
	X8	512Mb	(16M x 8 x 4B) x 1	
128MB	x32	256Mb	(8M x 8 x 4Bank) x 4	A[26:25]
	X8	512Mb	(32M x 4 x 4B) x 2	
	X16		(16Mx 8 x 4B) x 2	
	x32		(8M x 16 x 4B) x 2	

## 5.2.2 nWAIT 引脚操作

如果相对应每个存储器 bank 的 WAIT 位 (BWSCON 寄存器中的 WS<sub>n</sub> 位) 使能, 当存储器 bank 被激活, nOE 信号低电平有效持续时间可以被外部引脚 nWAIT 所延长。从 tacc-1 时刻开始检测 nWAIT 的状态。在采用 nWAIT 是高电平后的下一个时钟周期, nOE 也会被置高电平。nWE 与 nOE 也有同样的关系。

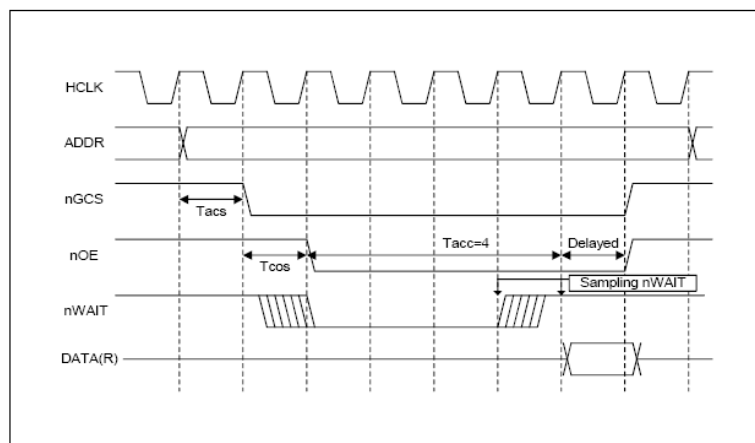


Figure 5-2. S3C2440A External nWAIT Timing Diagram (Tacc=4)

## 5.2.3 nXBREQ/nXBACK 引脚操作

如果 nXBREQ 引脚被拉低电平, s3c2440A 将通过置 nXBACK 引脚低电平对此响应。如果 nXBACK 为低电平, 则地址/数据总线和存储器控制信号将处于如表 1-1 所示的高阻态。当 nXBREQ 引脚被拉高以后, nXBACK 引脚也会恢复高电平。

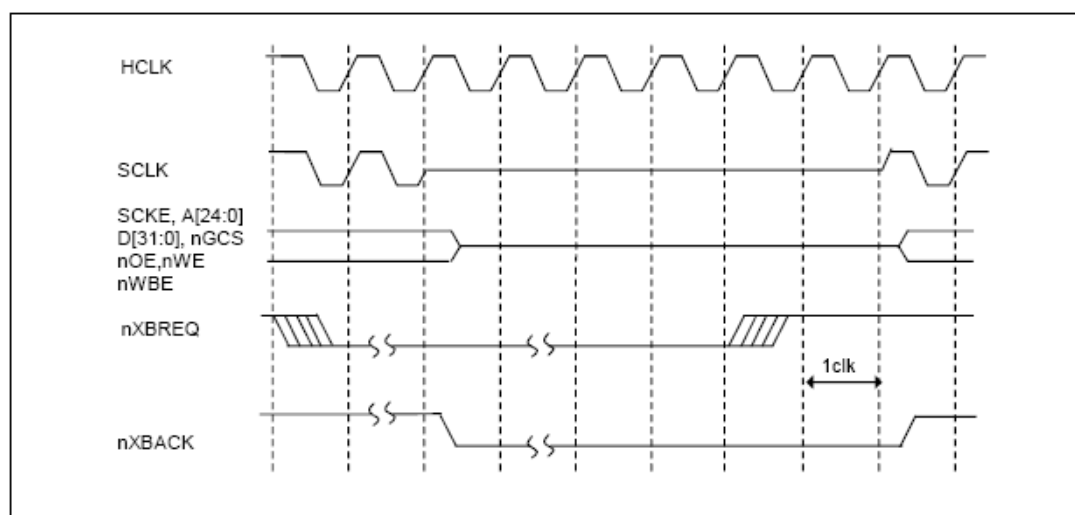


Figure 5-3. S3C2440A nXBREQ/nXBACK Timing Diagram

## 5.3 存储器接口示例

### 5.3.1 ROM 存储器接口示例

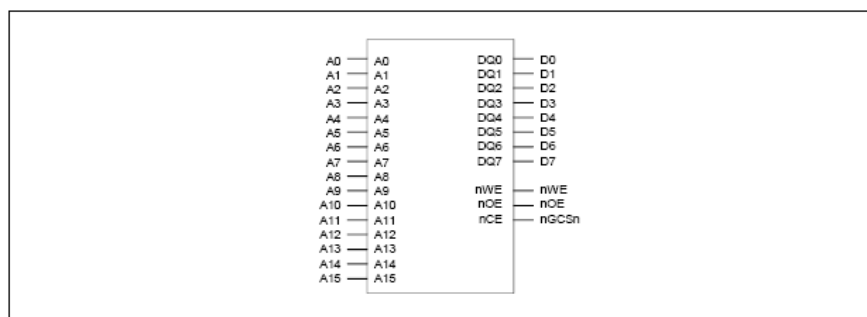


Figure 5-4. Memory Interface with 8-bit ROM

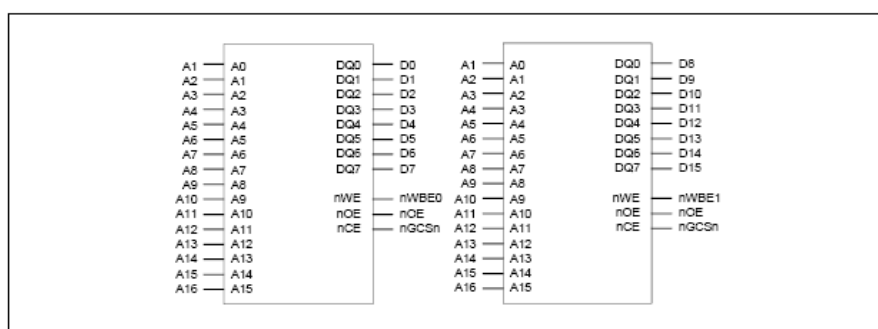


Figure 5-5. Memory Interface with 8-bit ROM x 2

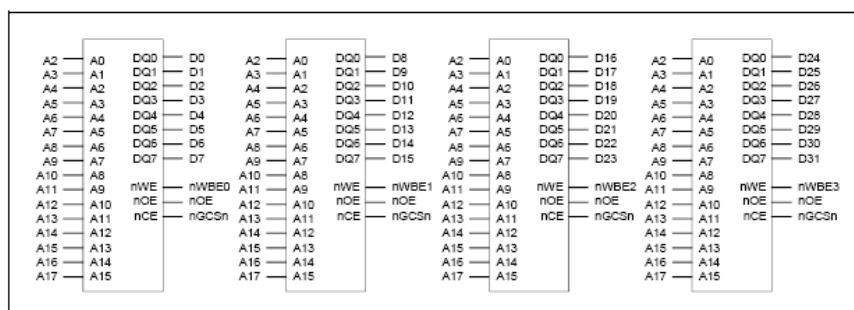


Figure 5-6. Memory Interface with 8-bit ROM x 4

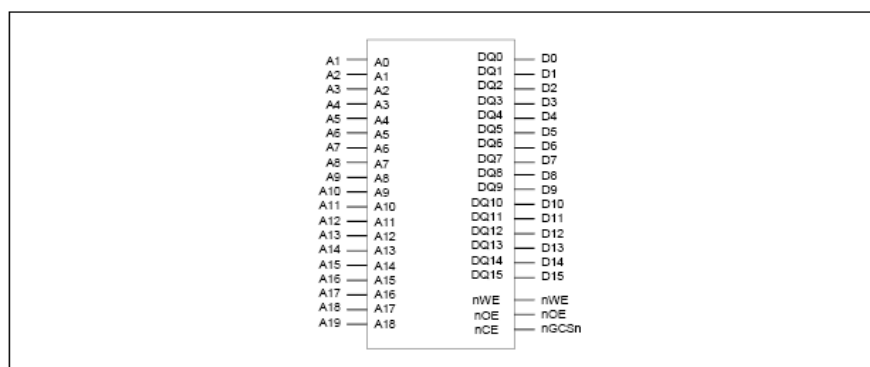


Figure 5-7. Memory Interface with 16-bit ROM

### 5.3.2 SRAM 存储器接口示例

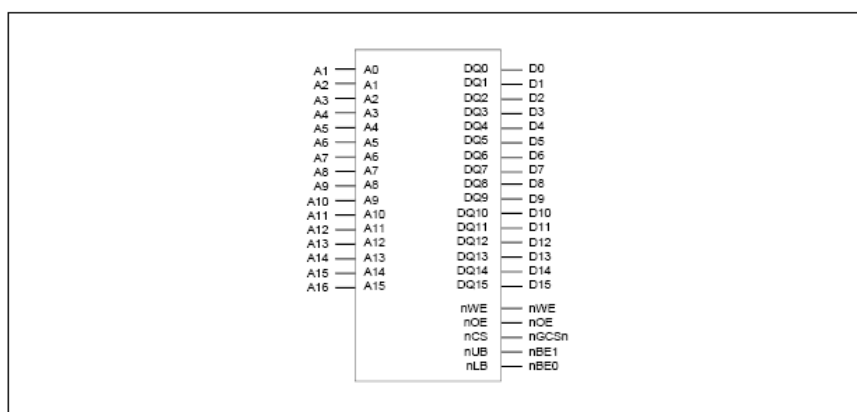


Figure 5-8. Memory Interface with 16-bit SRAM

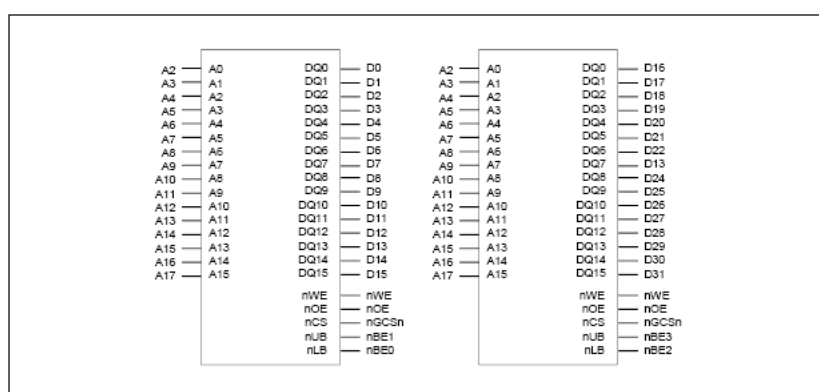


Figure 5-9. Memory Interface with 16-bit SRAM x 2

### 5.3.3 SDRAM 存储器接口示例

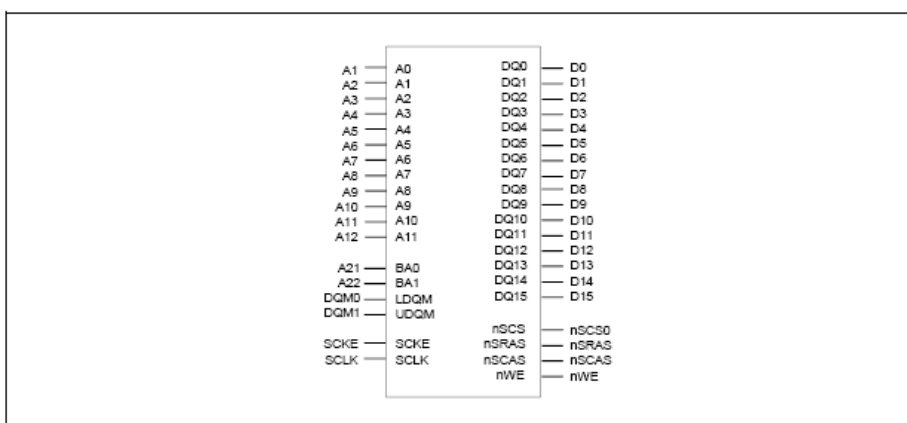


Figure 5-10. Memory Interface with 16-bit SDRAM (4Mx16, 4banks)

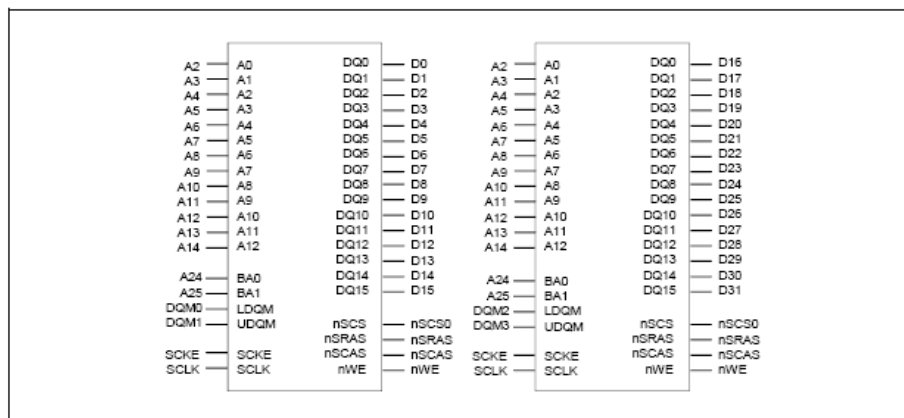


Figure 5-11. Memory Interface with 16-bit SDRAM (4Mx16x4Bank \* 2ea)

### 5.3.4 可编程存储器访问周期

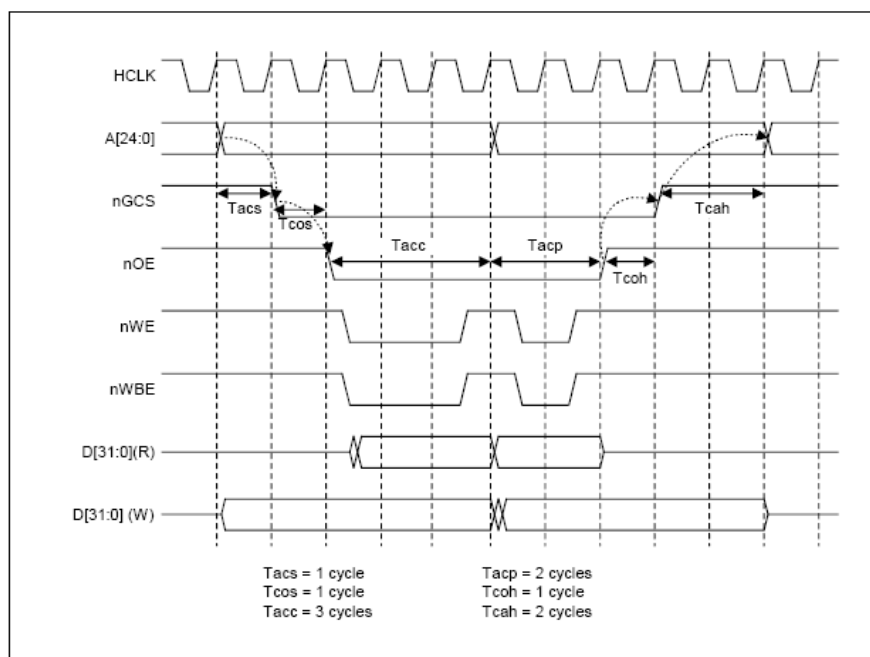


Figure 5-12. S3C2440A nGCS Timing Diagram

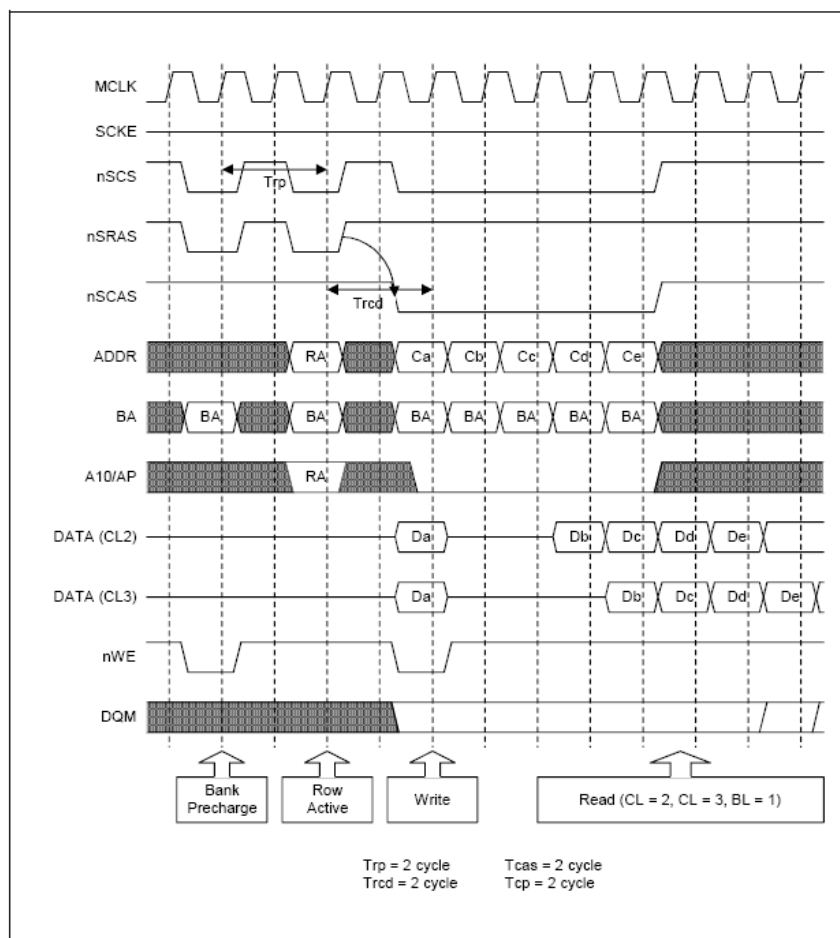


Figure 5-13. S3C2440A SDRAM Timing Diagram



## 5.4 存储器控制器寄存器

### 5.4.1 总线宽度&等待控制寄存器

#### BUS WIDTH & WAIT CONTROL REGISTER (BWSCON)

寄存器	地址	读写	描述	复位值
BWSCON	0x48000000	R/W	总线宽度&等待状态控制寄存器	0x000000

BWSCON	位	描述	初始值
ST7	[31]	决定SRAM对bank7 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS7	[30]	决定对于bank7 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW7	[29:28]	决定对于bank7 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST6	[27]	决定SRAM对bank6 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS6	[26]	决定对于bank6 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW6	[25:24]	决定对于bank6 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST5	[23]	决定SRAM对bank5 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS5	[22]	决定对于bank5 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW5	[21:20]	决定对于bank5 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST4	[19]	决定SRAM对bank4 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS4	[18]	决定对于bank4 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW4	[17:16]	决定对于bank4 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST3	[15]	决定SRAM对bank3 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS3	[14]	决定对于bank3 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW3	[13:12]	决定对于bank3 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0

ST2	[11]	决定SRAM对bank2 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS2	[10]	决定对于bank2 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW2	[9: 8]	决定对于bank2 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST1	[7]	决定SRAM对bank1 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS1	[6]	决定对于bank1 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW1	[5: 4]	决定对于bank1 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
DW0	[2:1]	决定对于bank0 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	-
保留	[0]	保留为 0	0

注:

1.在存储器控制器里的主时钟都对应总线时钟。例如, SRAM 中的 HCLK 与总线时钟一致, SDRAM 中的 SCLK 与总线时钟一致。在本章(存储器控制器)中, 一个时钟就意味一个总线时钟。

2.nBE[3:0]是 nWBE[3:0]和 nOE 的与信号。

## 5.4.2 Bank 控制寄存器

### BANK CONTROL REGISTER (BANKCONn: nGCS0-nGCS5)

寄存器	地址	读写	描述	复位值
BANKCON0	0x48000004	R/W	Bank0 控制寄存器	0x0700
BANKCON1	0x48000008	R/W	Bank1 控制寄存器	0x0700
BANKCON2	0x4800000C	R/W	Bank2 控制寄存器	0x0700
BANKCON3	0x48000010	R/W	Bank3 控制寄存器	0x0700
BANKCON4	0x48000014	R/W	Bank4 控制寄存器	0x0700
BANKCON5	0x48000018	R/W	Bank5 控制寄存器	0x0700

BANKCONn	位	描述	初始值
Tacs	[14:13]	nGCSn前的地址建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcos	[12:11]	nOE前的片选建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacc	[10:8]	访问周期 000 = 1 clock 001 = 2 clocks 010 = 3 clocks 011 = 4 clocks 100 = 6 clocks 101 = 8 clocks 110 = 10 clocks 111 = 14 clocks 注: nWAIT 信号被使用, Tacc 大于等于 4 clocks.	111
Tcoh	[7:6]	nOE后的片选保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcah	[5:4]	nGCSn后的地址保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacp	[3:2]	Page模式下的访问周期 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = 6 clocks	00
PMC	[1:0]	Page模式配置 00 = normal (1 data) 01 = 4 data 10 = 8 data 11 = 16 data	00

### 5.4.3 Bank 控制寄存器

#### BANK CONTROL REGISTER (BANKCONn: nGCS6-nGCS7)

寄存器	地址	读写	描述	复位值
BANKCON6	0x4800001C	R/W	Bank6 控制寄存器	0x18008
BANKCON7	0x48000020	R/W	Bank7 控制寄存器	0x18008

BANKCONn	位	描述	初始值
MT	[16:15]	Determine the memory type for bank6 and bank7. 00 = ROM or SRAM 01 = Reserved (Do not use) 10 = Reserved (Do not use) 11 = Sync. DRAM	11
<b>Memory Type = ROM or SRAM [MT=00] (15-bit)</b>			
Tacs	[14:13]	nGCSn前的地址建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcos	[12:11]	nOE前的片选建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacc	[10:8]	访问周期 000 = 1 clock 001 = 2 clocks 010 = 3 clocks 011 = 4 clocks 100 = 6 clocks 101 = 8 clocks 110 = 10 clocks 111 = 14 clocks	111
Tcoh	[7:6]	nOE后的片选保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcah	[5:4]	nGCSn后的地址保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacp	[3:2]	Page模式下的访问周期 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = 6 clocks	00
PMC	[1:0]	Page模式配置 00 = normal (1 data) 01 = 4 data 10 = 8 data 11 = 16 data	00
<b>Memory Type = SDRAM [MT=11] (4-bit)</b>			
Trcd	[3:2]	RAS to CAS 延时 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks	10
SCAN	[1:0]	列地址位数 00 = 8-bit 01 = 9-bit 10 = 10-bit	00

## 5.4.4 刷新控制寄存器

### REFRESH CONTROL REGISTER (REFRESH)

寄存器	地址	读写	描述	复位值
REFRESH	0x48000024	R/W	SDRAM 刷新控制寄存器	0xac0000

REFRESH	位	描述	初始值
REFEN	[23]	SDRAM 刷新使能 0 = 无效 1 = 有效 (自, CBR/自动刷新) 注:CBR (CAS Before RAS, 列提前于行定位) 式刷新	1
TREFMD	[22]	SDRAM刷新模式 0 = CBR/Auto Refresh 1 = Self Refresh 在自刷新模式下, SDRAM 控制信号被置于适当的电平	0
Trp	[21:20]	SDRAM RAS预充电时间 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = 不支持	10
Tsrc	[19:18]	SDRAM 半行周期时间Tsrc 00 = 4 clocks 01 = 5 clocks 10 = 6 clocks 11 = 7 clocks SDRAM 行周期时间: Trc=Tsrc+Trp 如果 Trp=3clocks & Tsrc=7clocks, Trc=3+7=10clocks.	11
Reserved	[17:16]	Not used	00
Reserved	[15:11]	Not used	0000
Refresh Counter	[10:0]	SDRAM 刷新计数值. 参考第六章SDRAM刷新控制寄存器总线优先级所在节 刷新时间 = $(2^{11}-refresh\_count+1)/HCLK$ Ex) 如果刷新时间是 7.8 us并且HCLK是 100 MHz, 刷新计数值如下: 刷新计数值 = $2^{11} + 1 - 100 \times 7.8 = 1269$	0

## 5.4.5 BANKSIZE 寄存器

### BANKSIZE REGISTER (BANKSIZE)

寄存器	地址	读写	描述	复位值
BANKSIZE	0x48000028	R/W	可调的 bank 大小寄存器	0x0

BANKSIZE	位	描述	初始值
BURST_EN	[7]	ARM 内核突发操作使能 0 = 无效突发操作 1 = 使能突发操作	0
Reserved	[6]	没用	0
SCKE_EN	[5]	SCKE 使能SDRAM power down 模式 0 = SDRAM power down 模式无效 1 = SDRAM power down 模式有效	0
SCLK_EN	[4]	SCLK仅在SDRAM访问周期内为减少电源消耗被使能。当SDRAM没有被访问时，SCLK变成低电平。 0 = SCLK 总是激活 1 = SCLK 仅当访问周期才被激活 (推荐).	0
Reserved	[3]	Not used	0
BK76MAP	[2:0]	BANK6/7 存储分布 010 = 128MB/128MB    001 = 64MB/64MB 000 = 32M/32M        111 = 16M/16M 110 = 8M/8M          101 = 4M/4M 100 = 2M/2M	

## 5.4.6 SDRAM 模式寄存器集寄存器

### SDRAM MODE REGISTER SET REGISTER (MRSR)

寄存器	地址	读写	描述	复位值
MRSRB6	0x4800002C	R/W	模式寄存器集寄存器 bank6	xxx
MRSRB7	0x48000030	R/W	模式寄存器集寄存器 bank7	xxx

BANKSIZE	位	描述	初始值
Reserved	[11:10]	没用	-
WBL	[9]	写突发长度 Write burst length 0: 突发(固定的) 1: Reserved	x
TM	[8:7]	测试模式 00: 模式寄存器集 (固定) 01, 10 and 11: 保留	xx
CL	[6:4]	CAS 反应时间 000 = 1 clock, 010 = 2 clocks, 011=3 clocks Others: 保留	xxx
BT	[3]	突发类型 0: 连续的(固定) 1: 保留	x
BL	[2:0]	Burst length 000: 1 (固定) Others: 保留	