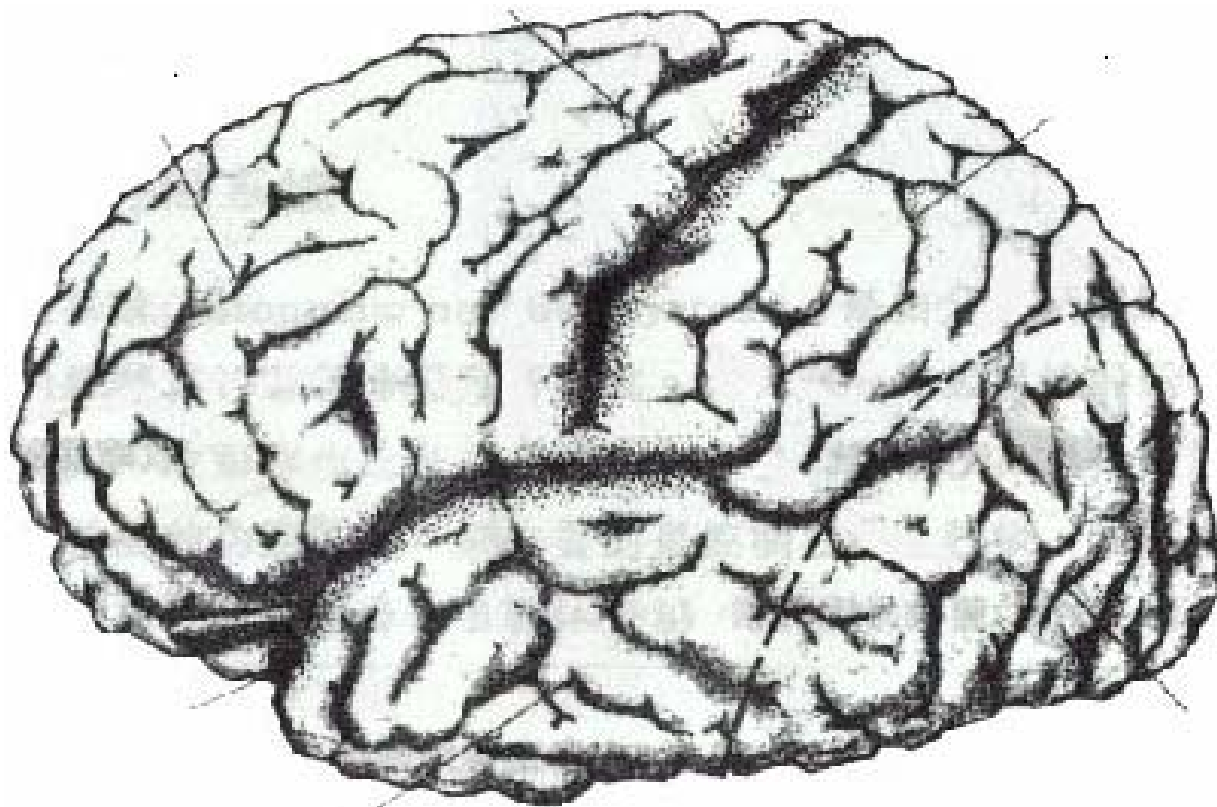




嵌入式系统设计

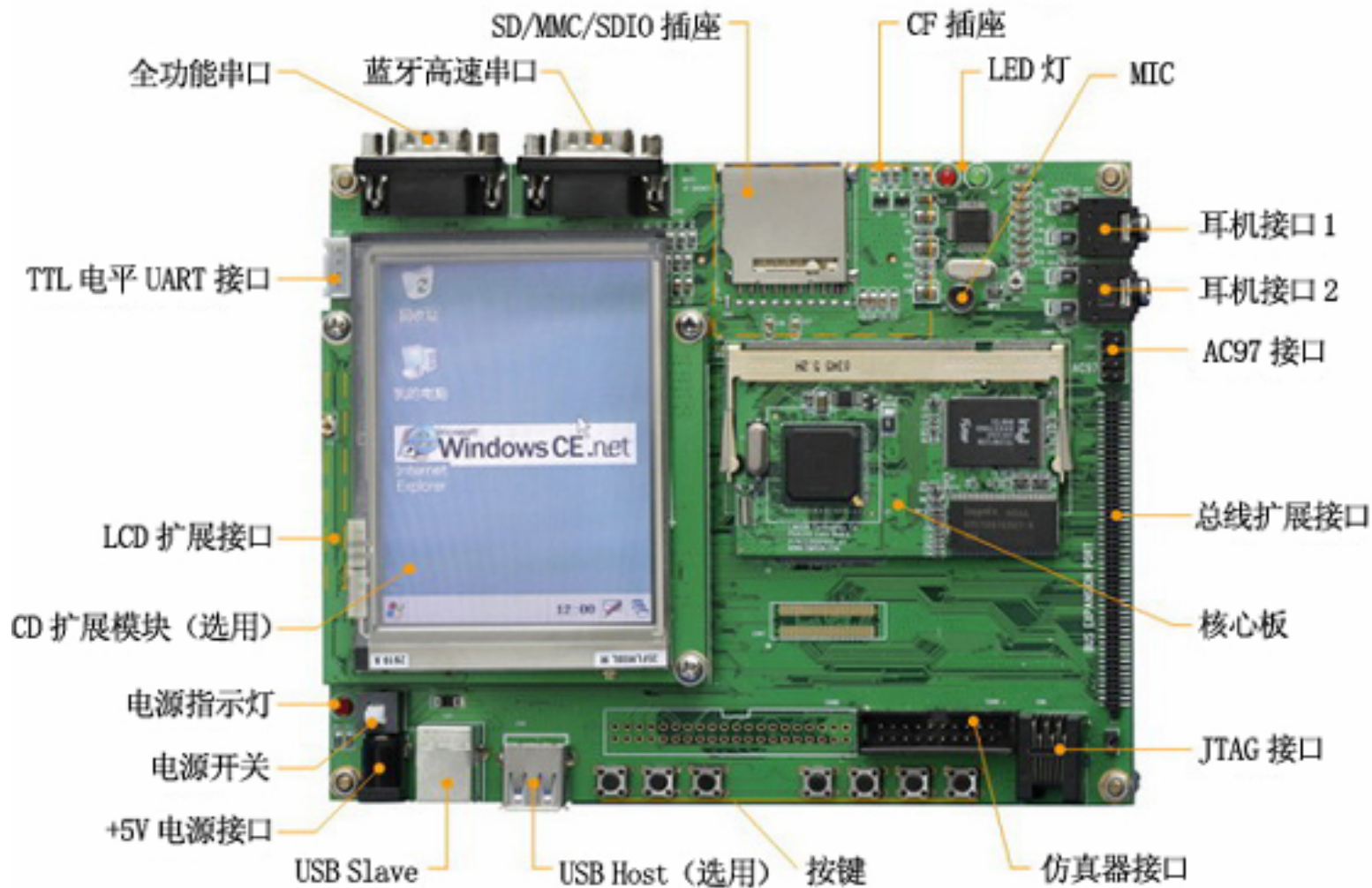
数字接口电路设计与编程 (1)

数字接口-引言 (1)



离开了躯体它能做什么？

数字接口-引言 (2)



与人的差距有多大?

北京大学软件与微电子学院 2006.2

数字接口-引言 (3)

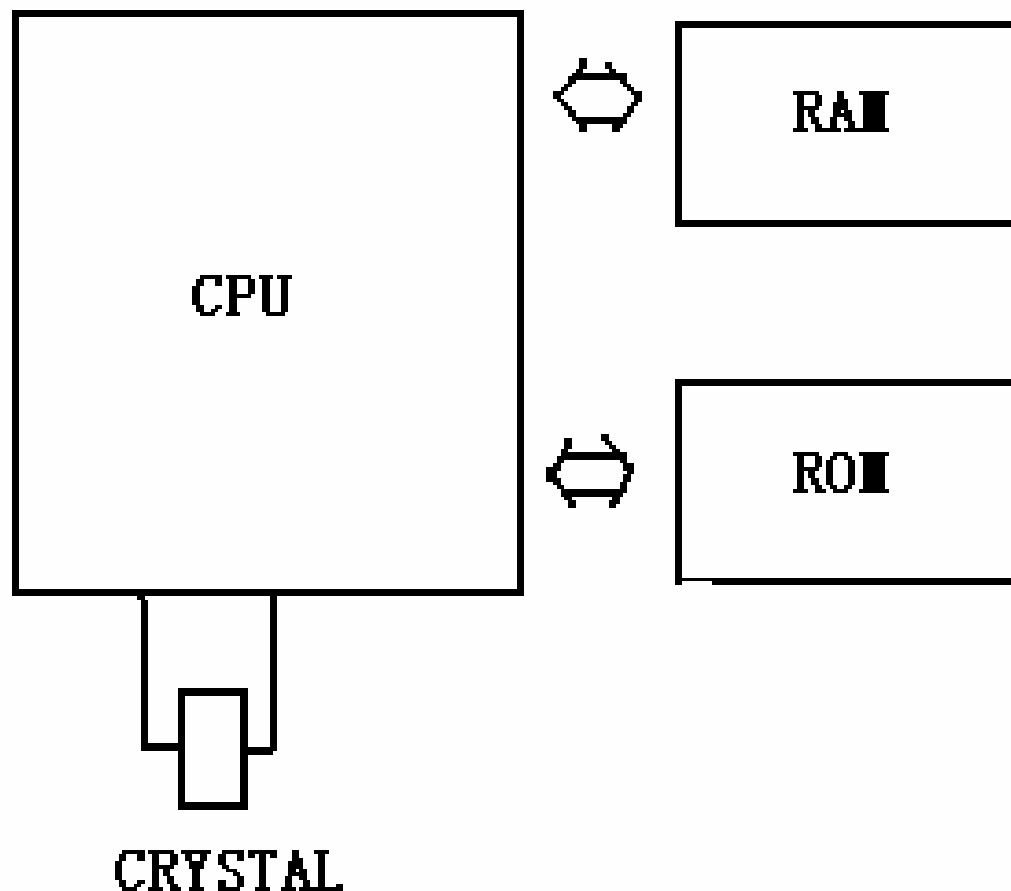
- 能够使处理器工作的最基本的条件有哪些？
- 为什么需要不同种类的存储器，它们各有什么特点？
- 选择外存储器需要考虑哪些因素？
- 有哪些常用的接口技术，在系统设计和编程时如何实现？

数字接口-主要内容

- 嵌入式系统的组成
- 处理器接口方式
- 存储器技术
- 通讯接口技术
- 人机交互技术
- 系统技术
- 小结

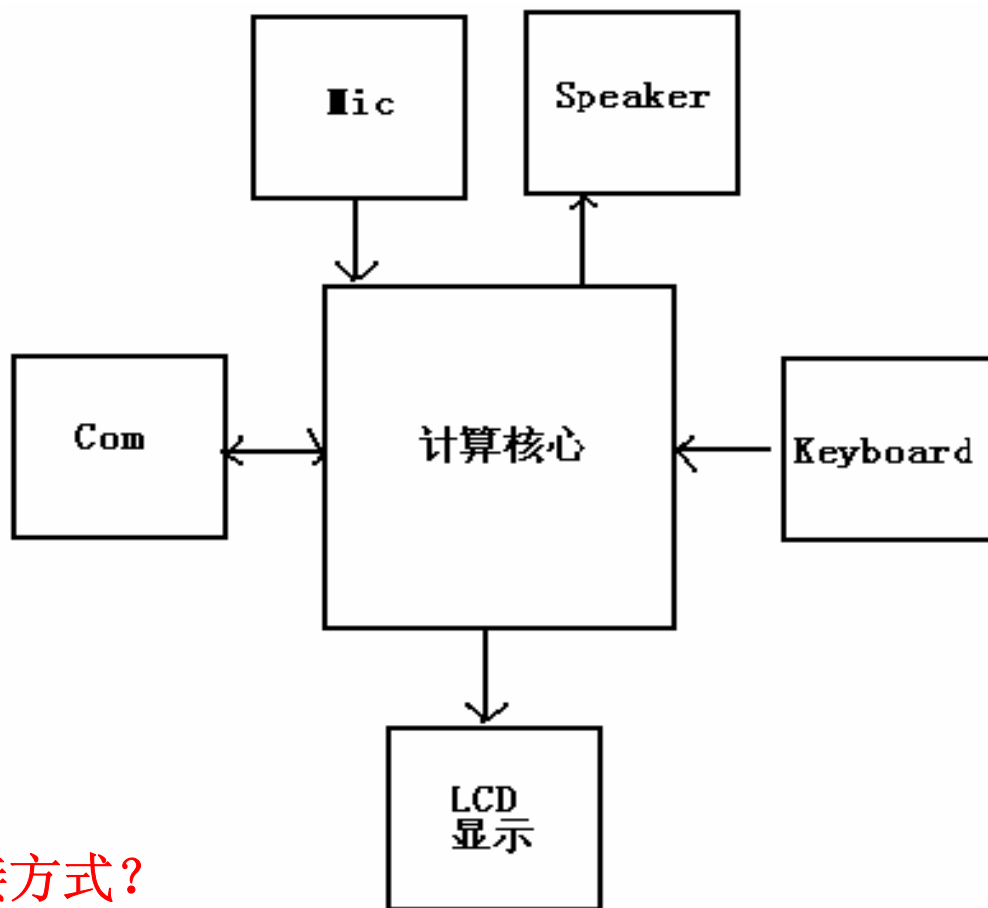
数字接口-系统组成 (1)

1. 系统计算核心



数字接口-系统组成 (2)

2. 基本的PDA系统



外设与CPU的连接方式?

数字接口-系统组成 (3)

3. 外设与CPU的连接方式

通过**接口电路**，连接到**CPU总线**上。

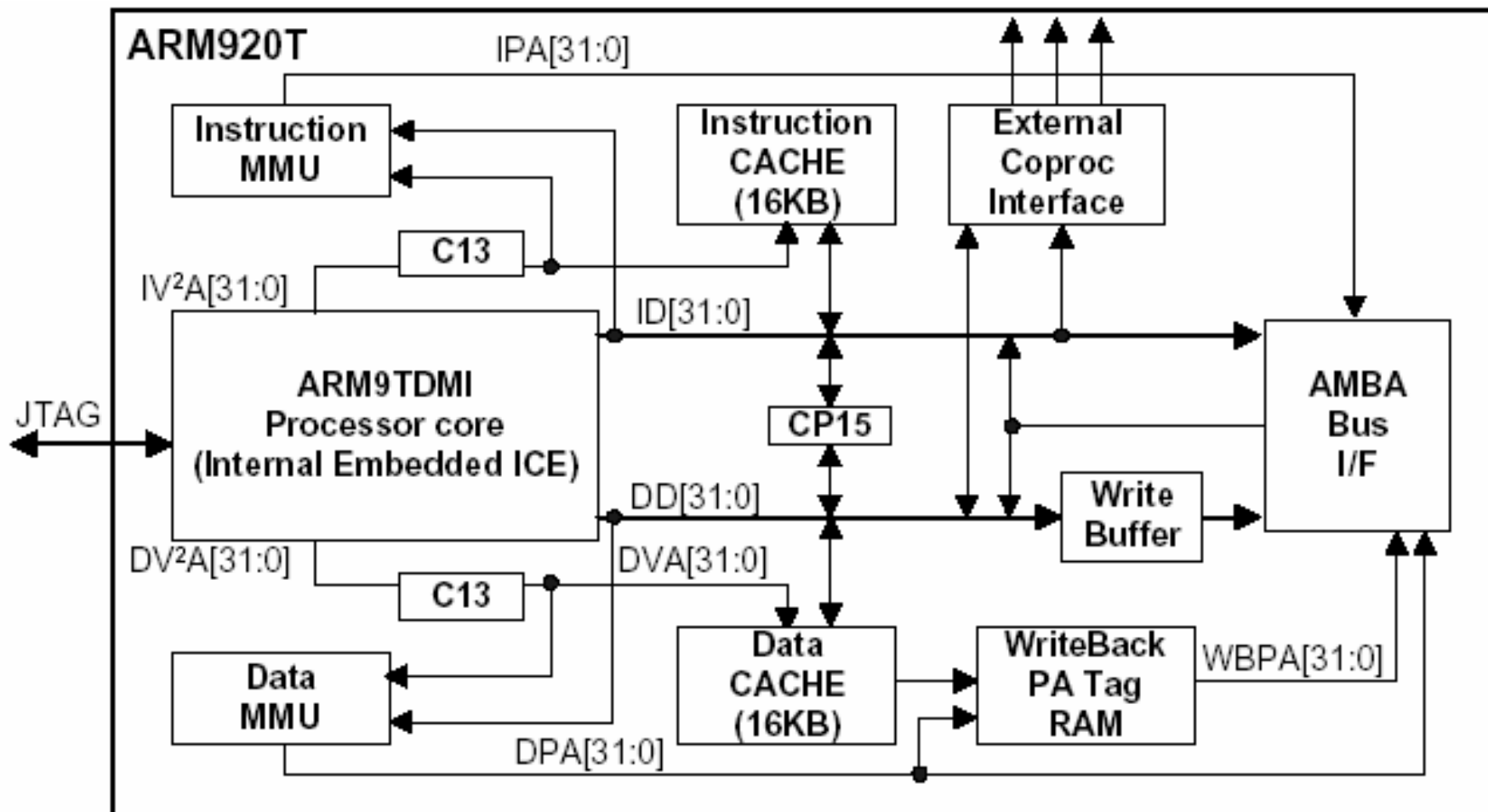
映射到处理器的**IO**空间，或**memory**空间。

数字接口-主要内容

- 嵌入式系统的组成
- 处理器接口方式
- 存储器技术
- 通讯接口技术
- 人机交互技术
- 系统技术
- 小结

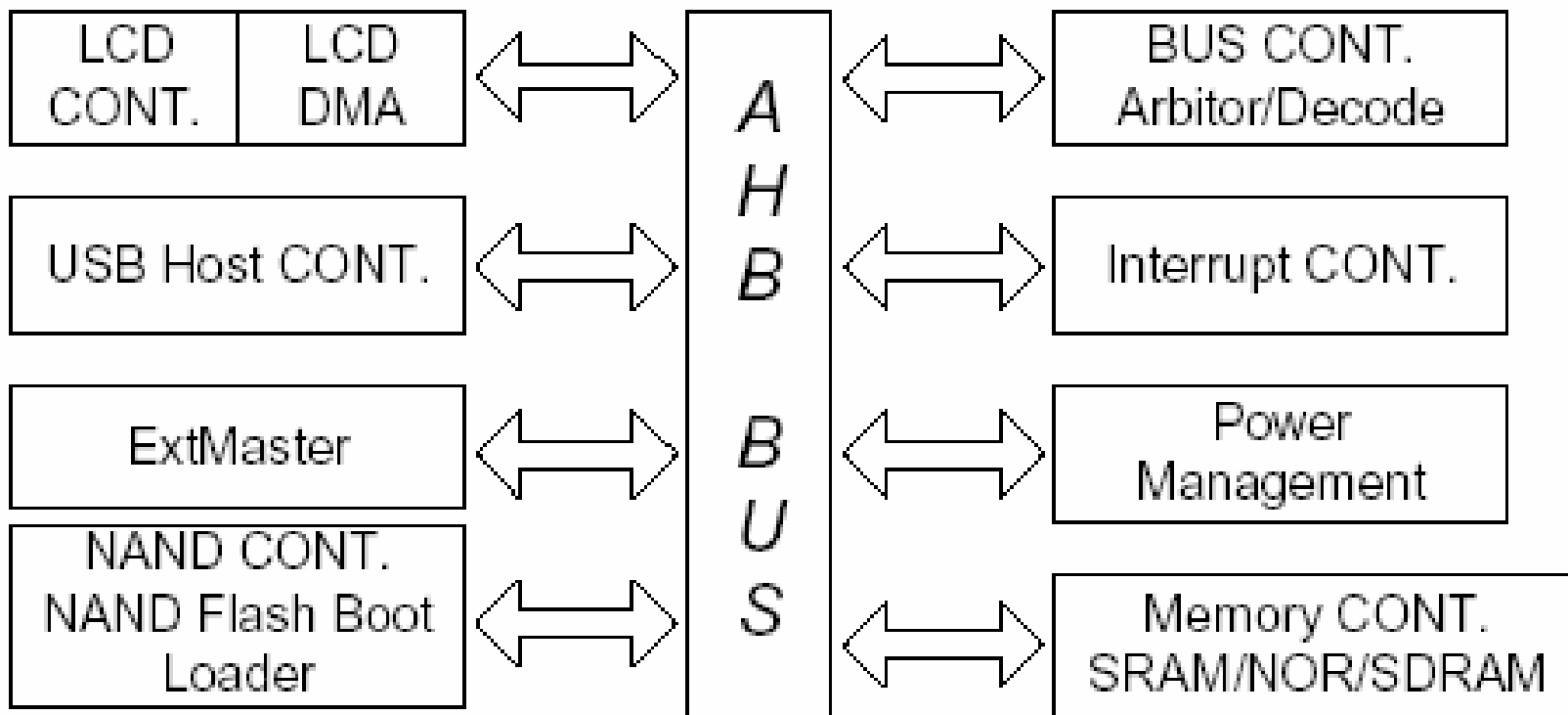
数字接口-处理器 (1)

■ S3c2410 (core)



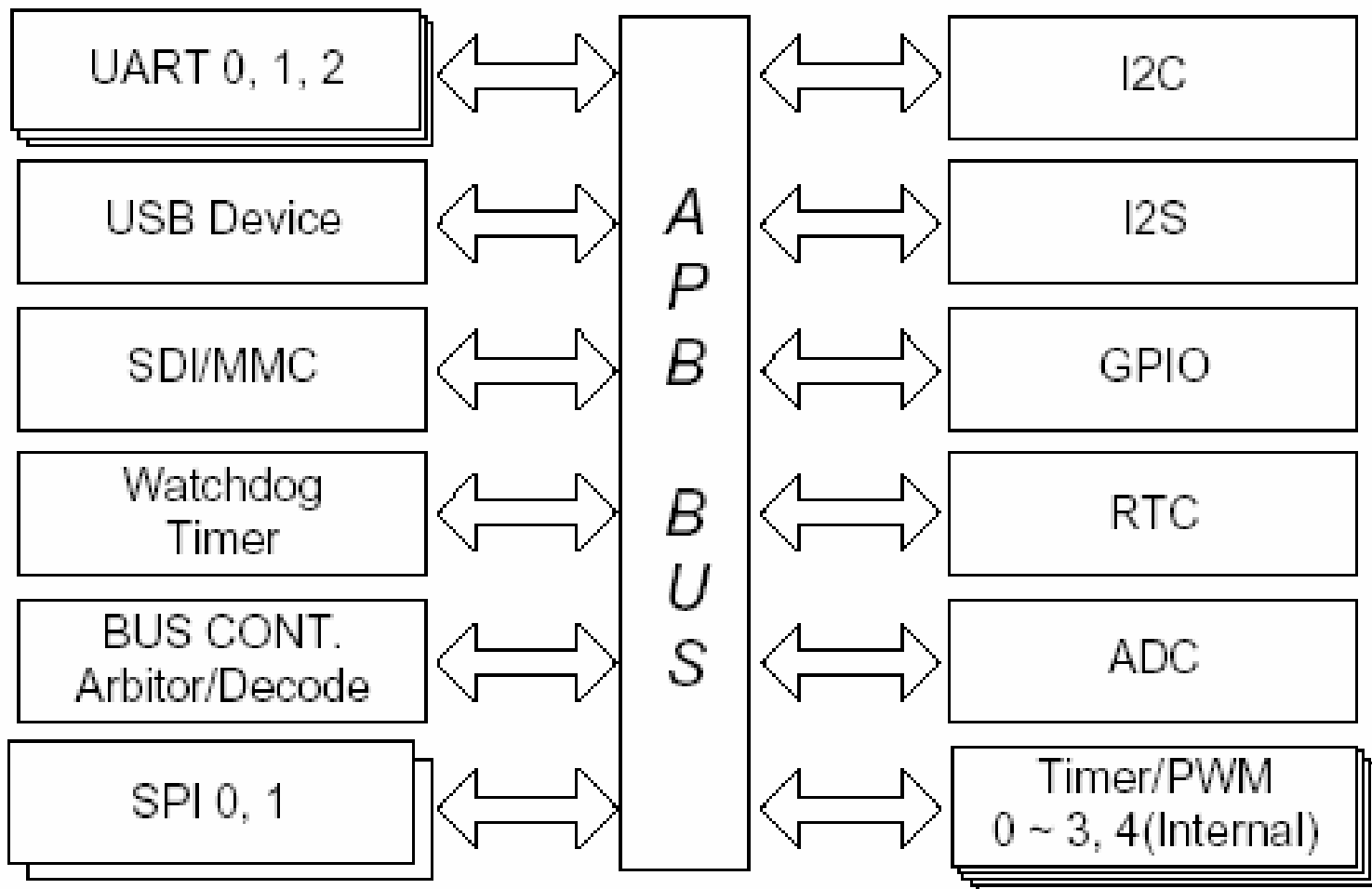
数字接口-处理器 (2)

- S3c2410 (on-chip peripheral devices)



数字接口-处理器 (3)

■ S3c2410 (on-chip peripheral devices)



数字接口-处理器 (4)

- S3c2410 (on-chip peripheral devices)

Bridge & DMA (4Ch)



数字接口-处理器 (5)

- S3c2410 (Bus Pins)
 - ADDR[26:0] (Address Bus)
 - DATA[31:0] (Data Bus)
 - nGCS[7:0] 0 nGCS[7:0] (General Chip Select)
 - nWE (Write Enable)
 - nOE (Output Enable)
 - nXBREQ (Bus Hold Request).
 - nXBACK
 - nWAIT

扩展外部设备的总线信号

这些信号的驱动源?

数字接口-主要内容

- 嵌入式系统的组成
- 处理器接口方式
- 存储器技术
- 通讯接口技术
- 人机交互技术
- 系统技术
- 小结

存储器接口-概述 (1)

1. 存储器类型

- DRAM

- DRAM , SDRAM

- SRAM

- SRAM, SBSRAM, NVRAM

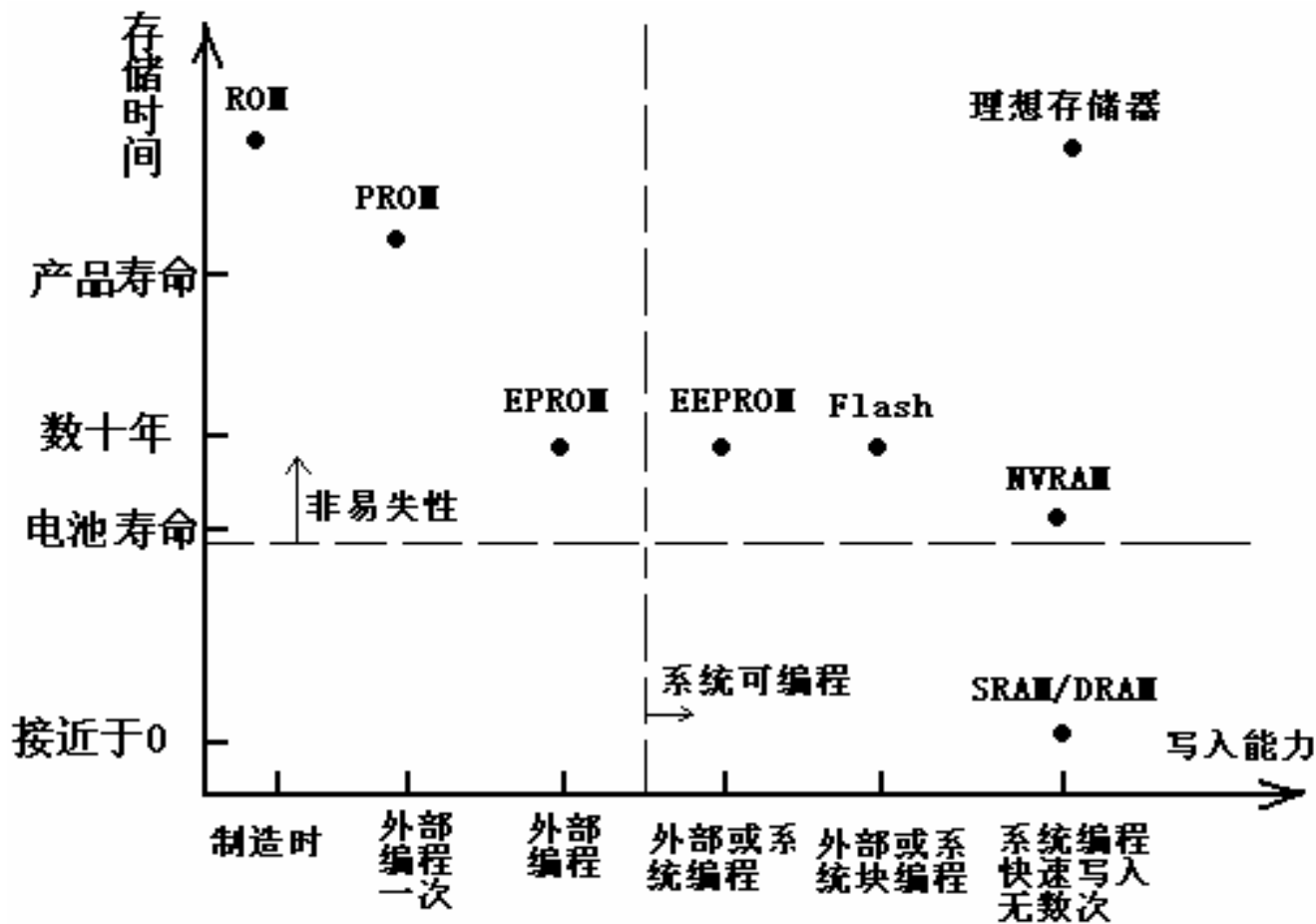
- ROM

- ROM, PROM, EPROM, EEPROM, FLASH

DRAM、SRAM和ROM数据存特性的不同点在哪儿？

存储器接口-概述 (2)

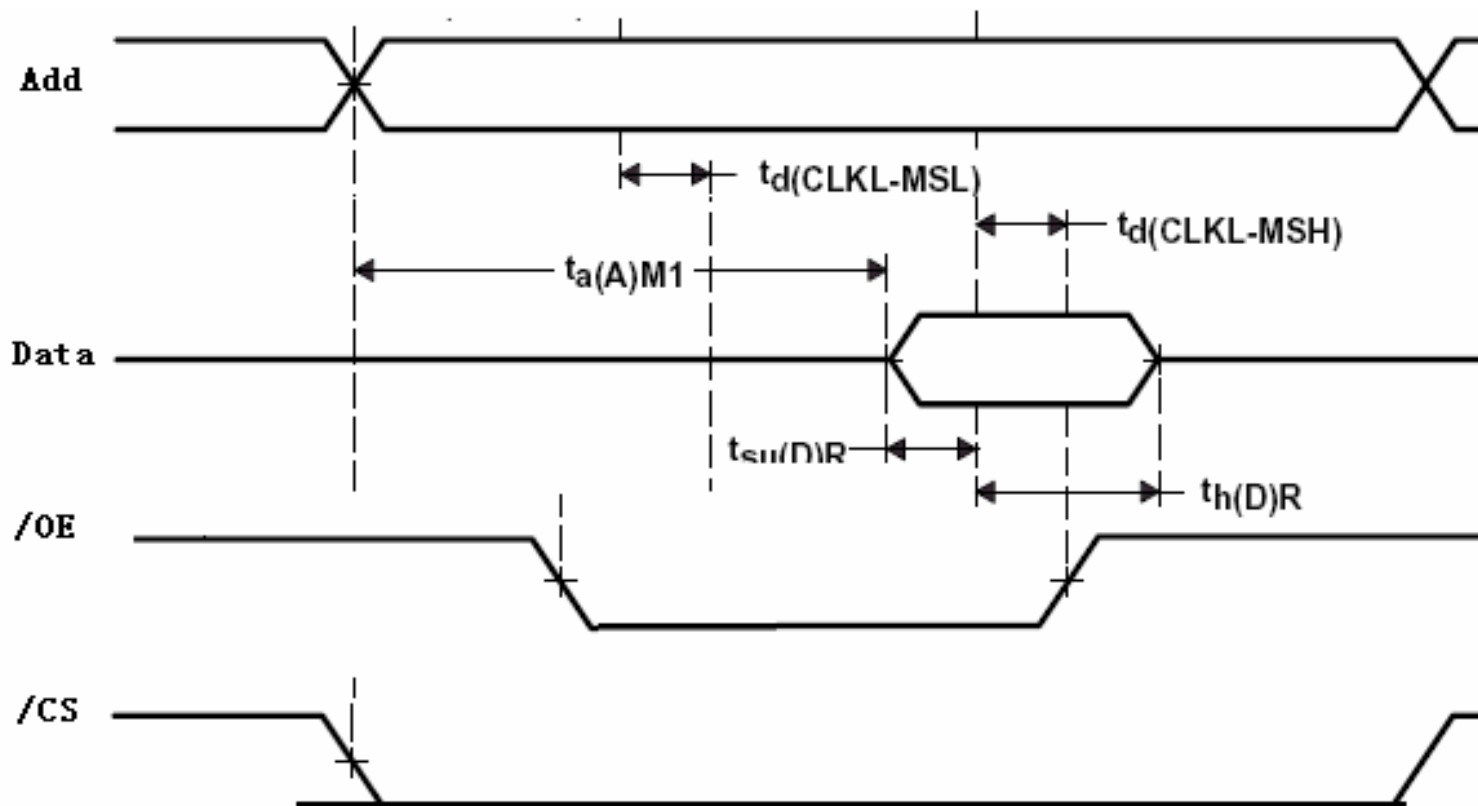
2. 不同存储器比较



存储器接口-概述 (3)

3. 典型访问时序

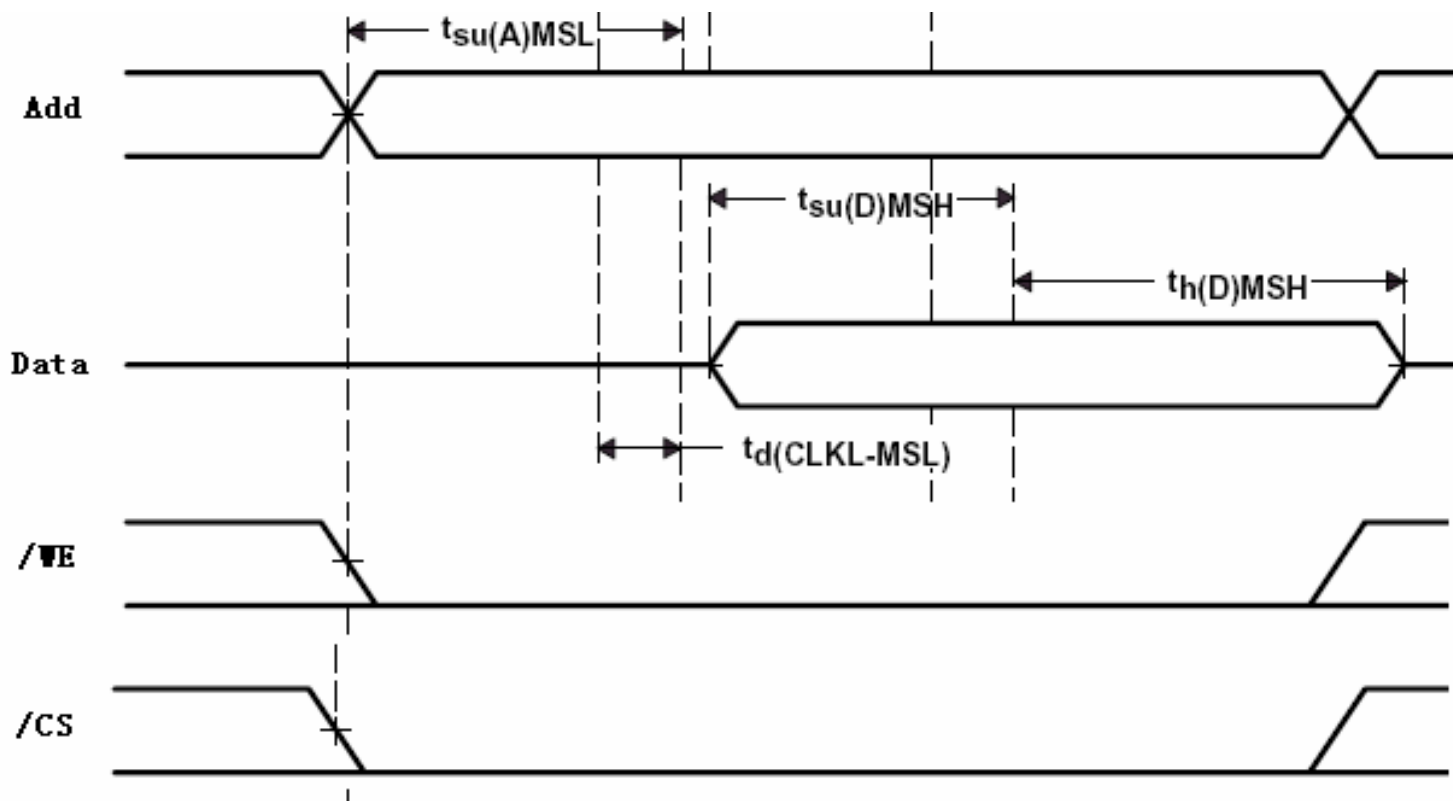
■ Read



存储器接口-概述 (4)

3. 典型访问时序 (续)

■ Write



存储器接口-SRAM (1)

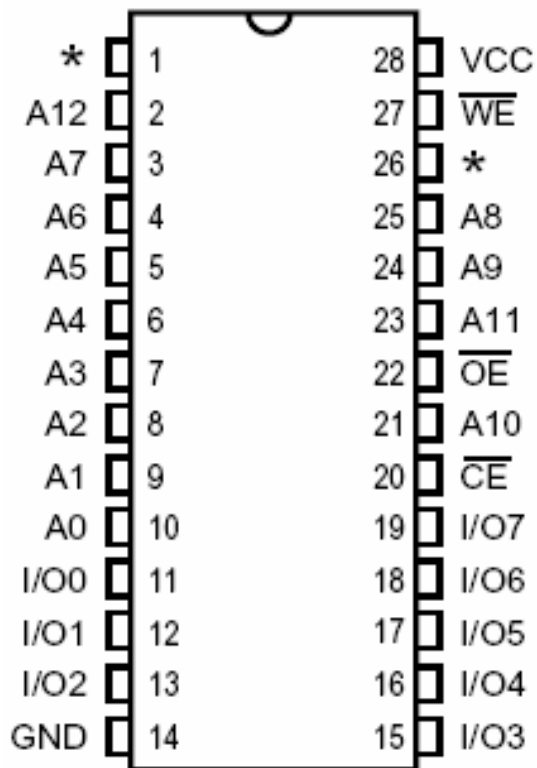
1. 概述

- SRAM: Static Random Access Memory
- Type
 - ASRAM (Asynchronous SRAM)
 - SSRAM (Synchronous SRAM)
- Interface Signal
 - Data (I/O)
 - Address (I)
 - Control (I)
 - /R (R/W), /W, /CS
 - MEMCLK, Mode, (SSRAM)

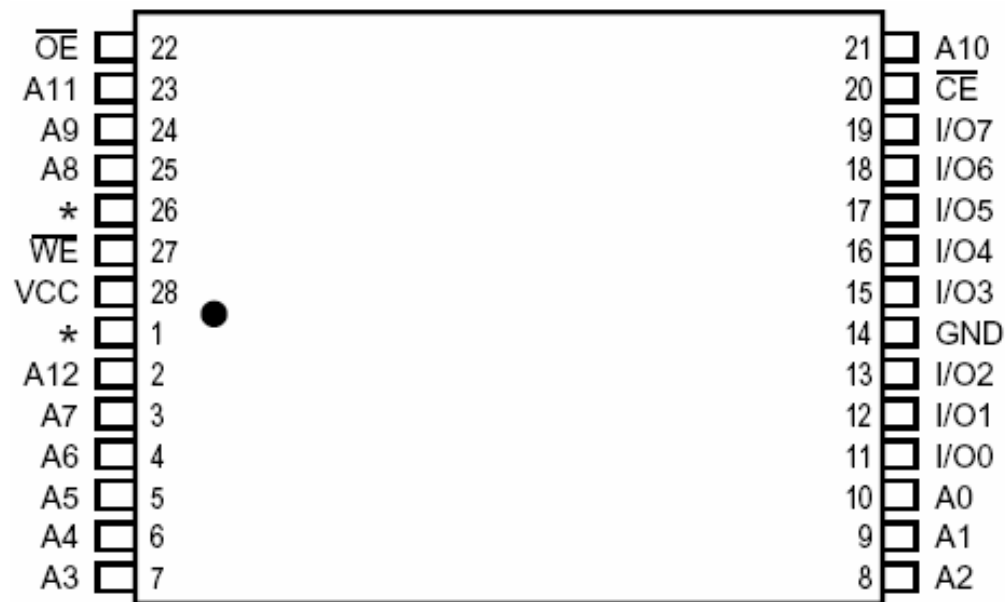
存储器接口-SRAM (2)

2. ASRAM-封装 (IS61C64B)

■ SOJ

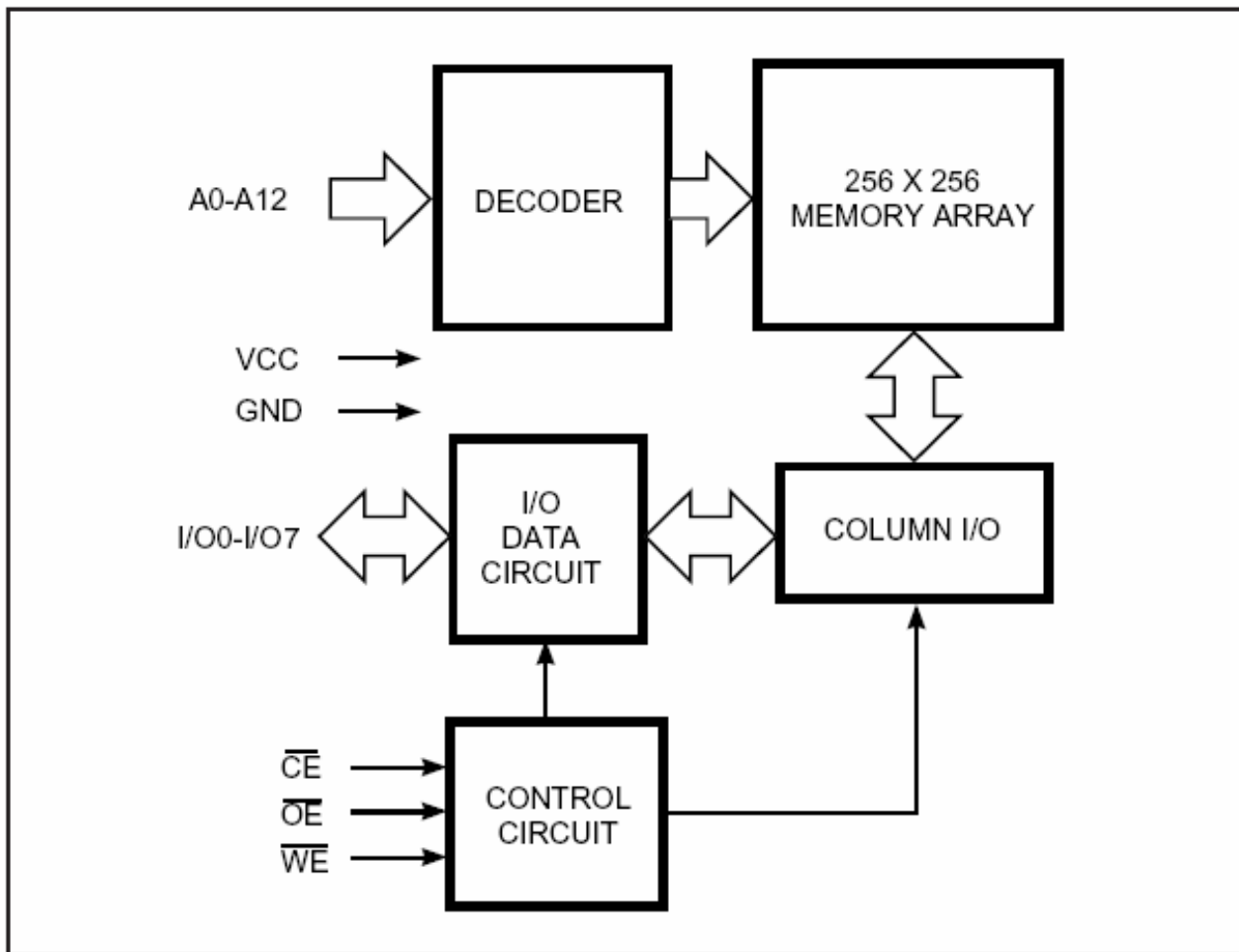


TSOP



存储器接口-SRAM (3)

2. ASRAM-结构 (IS61C64B)



存储器接口-SRAM (4)

2. ASRAM-真值表 (IS61C64B)

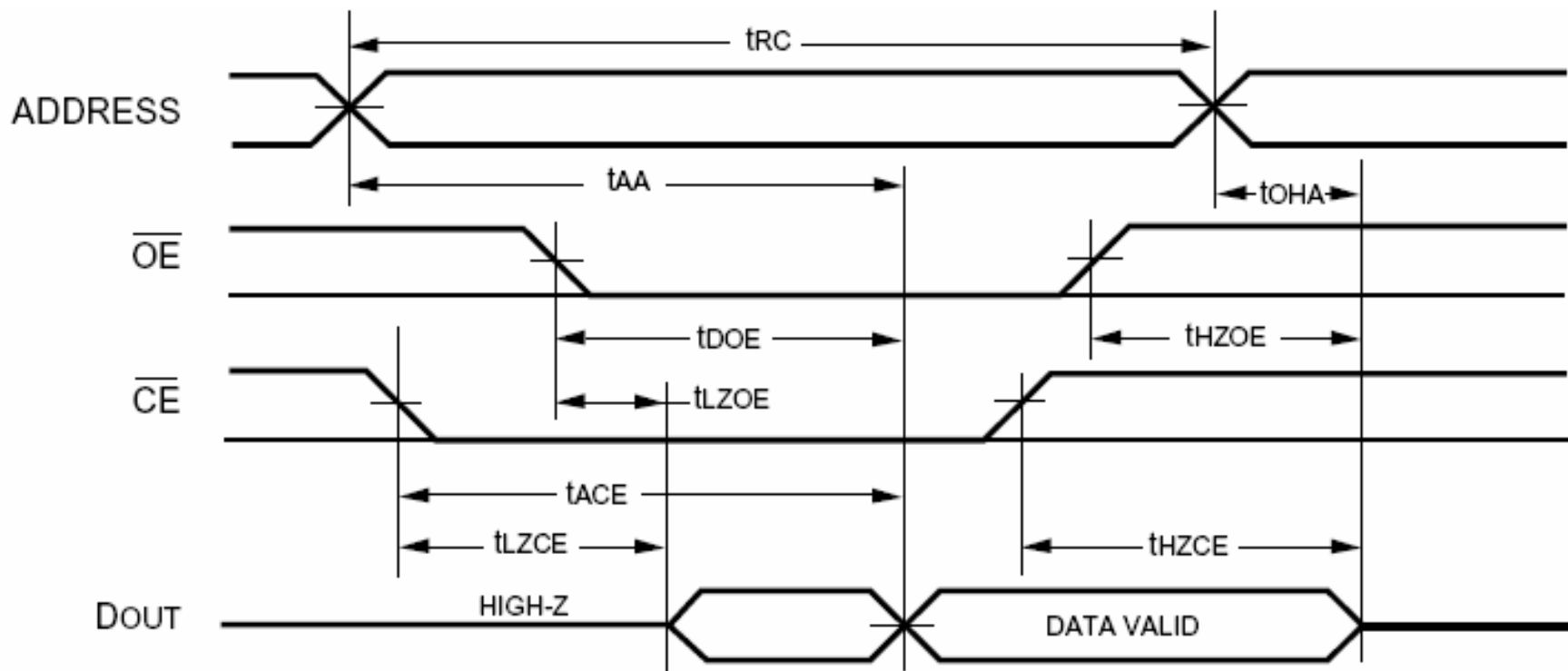
Mode	\overline{WE}	\overline{CE}	\overline{OE}	I/O Operation	Vcc Current
Not Selected	X	H	X	High-Z	I_{SB1}, I_{SB2}
(Power-down)	X	X	X	High-Z	I_{SB1}, I_{SB2}
Output Disabled	H	L	H	High-Z	I_{CC}
Read	H	L	L	Dout	I_{CC}
Write	L	L	X	Din	I_{CC}

如果 \overline{WE} 和 \overline{OE} 同时有效，会出现什么结果？

存储器接口-SRAM (5)

2. ASRAM-时序 (IS61C64B)

■ Read



这些参数对设计有什么帮助？

存储器接口-SRAM (6)

2. ASRAM-时序 (ISSI61C64B) (续)

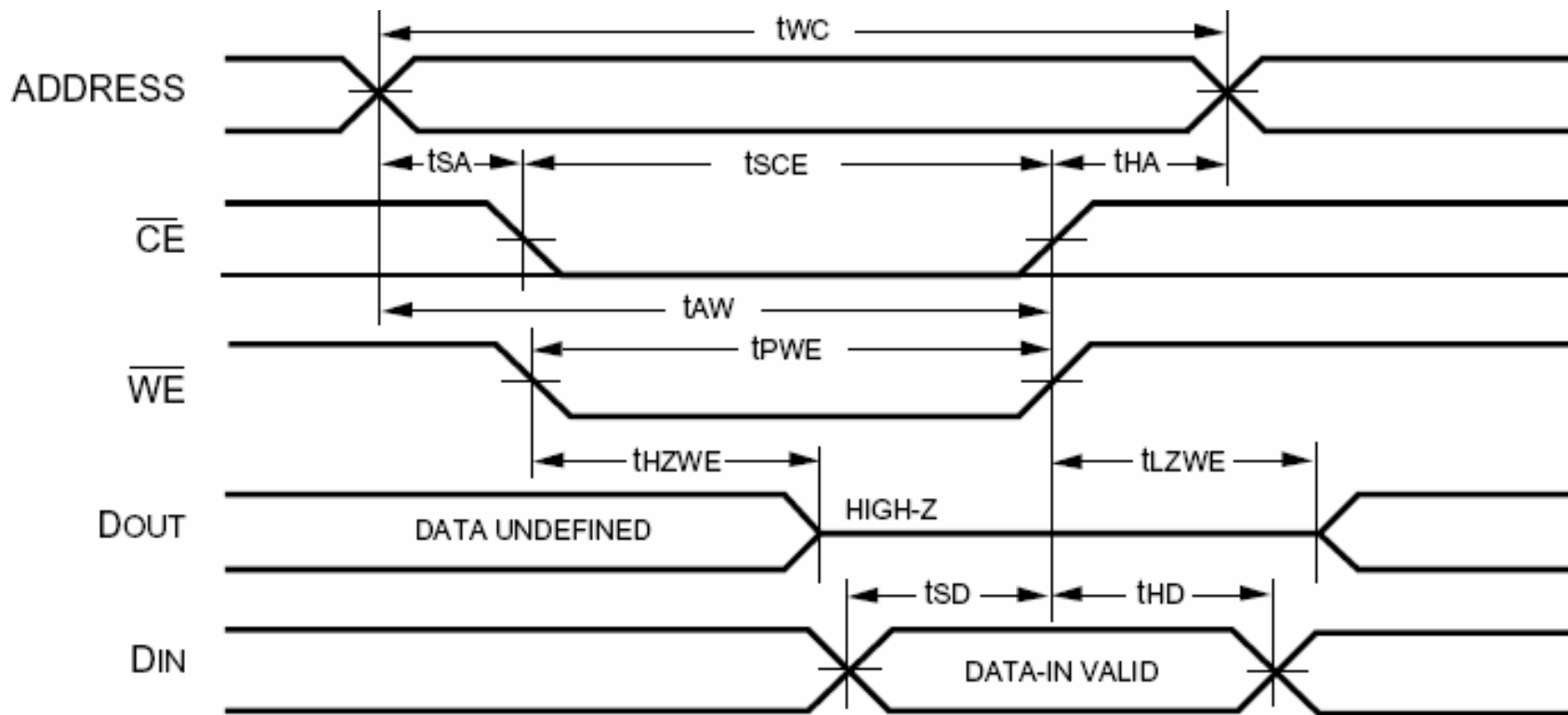
■ Read (续)

Symbol	Parameter	-10ns	
		Min.	Max.
t_{RC}	Read Cycle Time	10	—
t_{AA}	Address Access Time	—	10
t_{OHA}	Output Hold Time	2	—
t_{ACE}	\overline{CE} Access Time	—	10
t_{DOE}	\overline{OE} Access Time	—	5
$t_{LZOE}^{(2)}$	\overline{OE} to Low-Z Output	0	—
$t_{HZOE}^{(2)}$	\overline{OE} to High-Z Output	—	5
$t_{LZCE1}^{(2)}$	\overline{CE} to Low-Z Output	2	—
$t_{HZCE}^{(2)}$	\overline{CE} to High-Z Output	—	5

存储器接口-SRAM (7)

2. ASRAM-时序 (ISSI61C64B) (续)

Write



存储器接口-SRAM (8)

2. ASRAM-时序 (ISSI61C64B) (续)

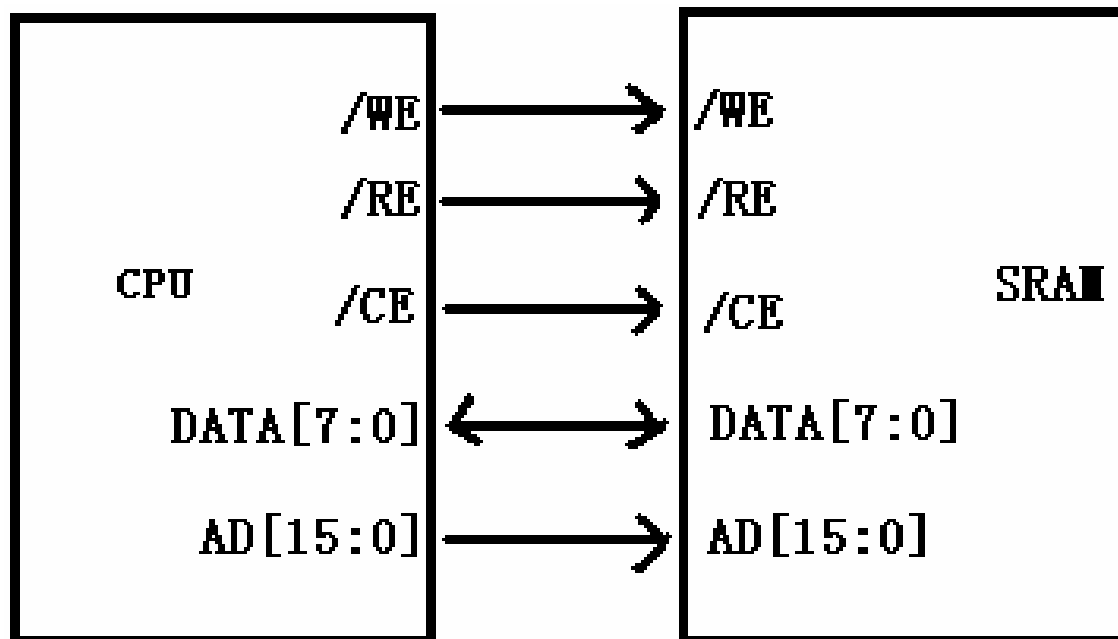
■ Write (续)

Symbol	Parameter	-10ns	
		Min.	Max.
t _{wc}	Write Cycle Time	10	—
t _{sCE}	$\overline{\text{CE}}$ to Write End	9	—
t _{AW}	Address Setup Time to Write End	9	—
t _{HA}	Address Hold from Write End	0	—
t _{SA}	Address Setup Time	0	—
t _{PWE} ⁽⁴⁾	$\overline{\text{WE}}$ Pulse Width	8	—
t _{SD}	Data Setup to Write End	8	—
t _{HD}	Data Hold from Write End	0	—
t _{HZWE} ⁽²⁾	$\overline{\text{WE}}$ LOW to High-Z Output	—	6
t _{LZWE} ⁽²⁾	$\overline{\text{WE}}$ HIGH to Low-Z Output	0	—

存储器接口-SRAM (9)

2. ASRAM-CPU连接

- 地址与数据总线分离

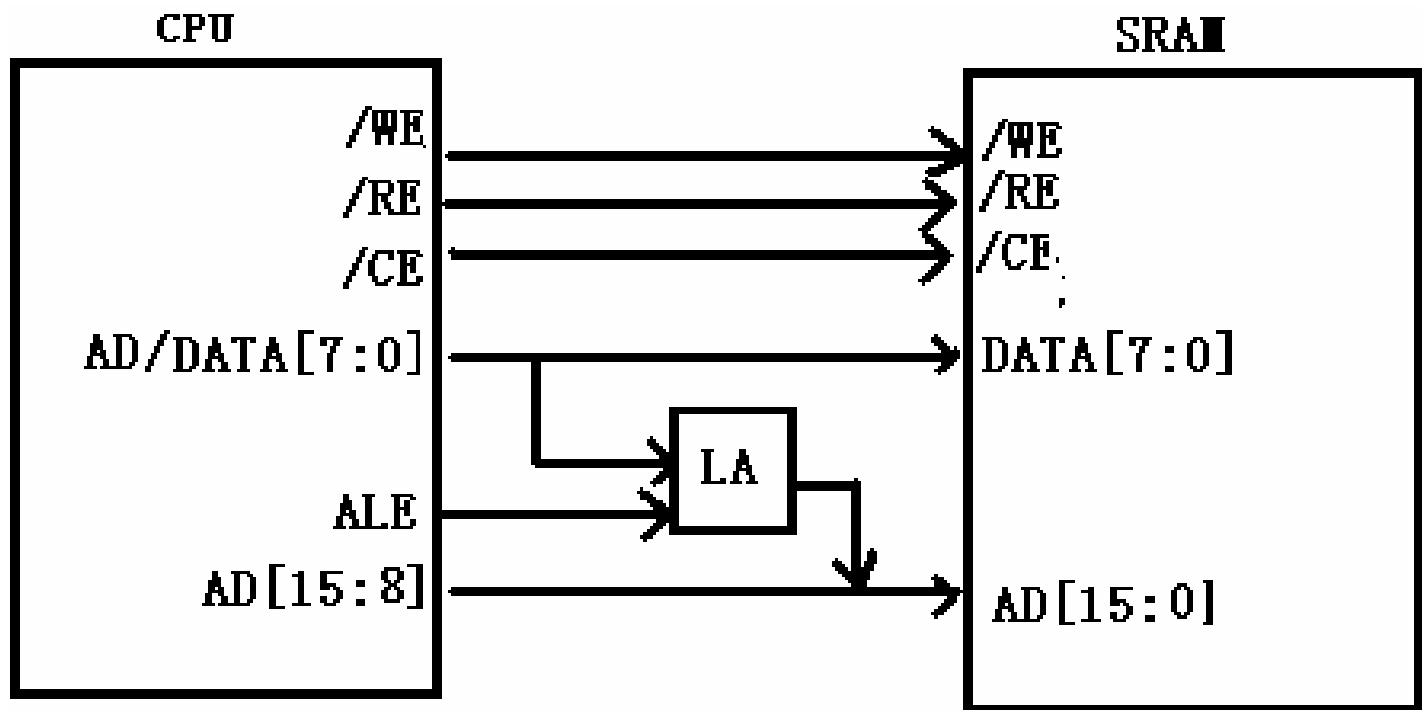


如果CPU用一个R/W信号的H和L分别表示Read和Write，如何连接？

存储器接口-SRAM (10)

2. ASRAM-CPU连接

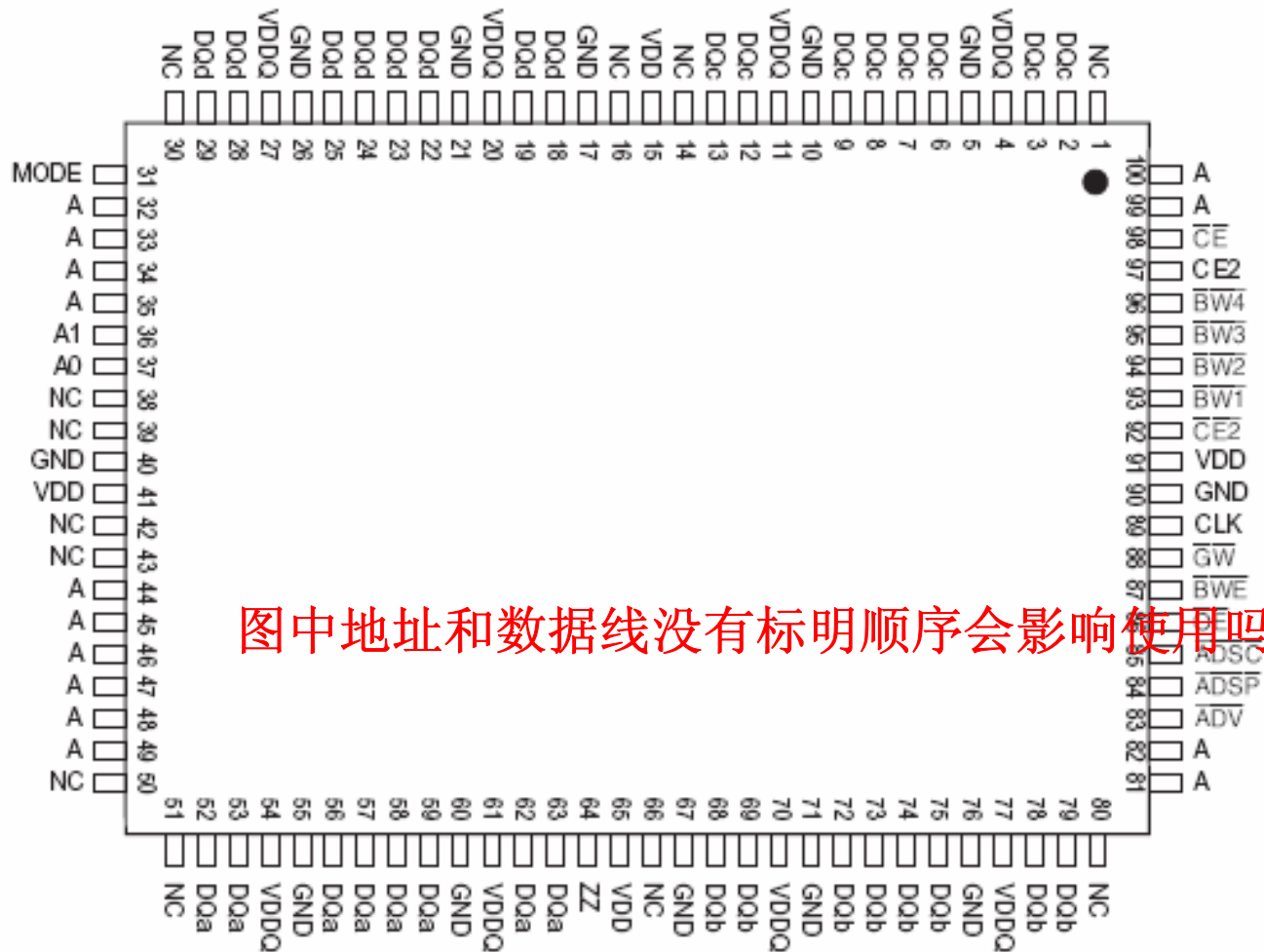
- 地址与数据共享总线（低性能处理器）



如果有多片RAM分布在不同的地址范围，在设计上需要如何处理？

存储器接口-SRAM (11)

3. SSRAM-封装 (IS64LF6432)



存储器接口-SRAM (12)

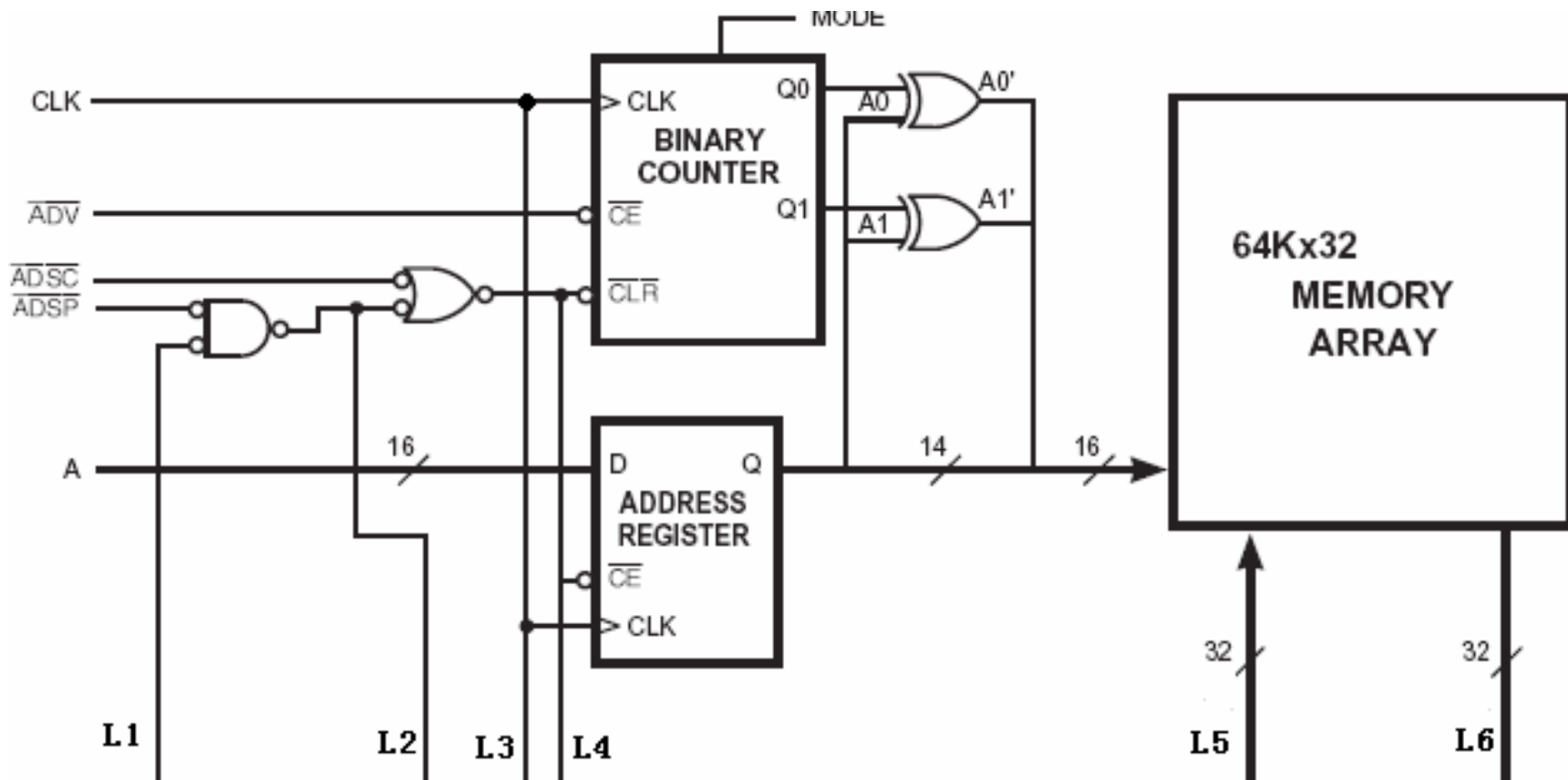
3. SSRAM-Pin description (IS64LF6432)

A0, A1	Synchronous Address Inputs. These pins must tied to the two LSBs of the address bus.
A	Synchronous Address Inputs
CLK	Synchronous Clock
$\overline{\text{ADSP}}$	Synchronous Processor Address Status
$\overline{\text{ADSC}}$	Synchronous Controller Address Status
$\overline{\text{ADV}}$	Synchronous Burst Address Advance
$\overline{\text{BW1}}\text{-}\overline{\text{BW4}}$	Individual Byte Write Enable
$\overline{\text{BWE}}$	Synchronous Byte Write Enable

$\overline{\text{GW}}$	Synchronous Global Write Enable
$\overline{\text{CE}}, \overline{\text{CE2}}, \text{CE2}$	Synchronous Chip Enable
$\overline{\text{OE}}$	Output Enable
DQa-DQd	Synchronous Data Input/Output
MODE	Burst Sequence Mode Selection
V _{DD}	+3.3V Power Supply
GND	Ground
V _{DDQ}	Isolated Output Buffer Supply: +3.3V or 2.5V
ZZ	Snooze Enable
DQPa-DQPd	Parity Data I/O

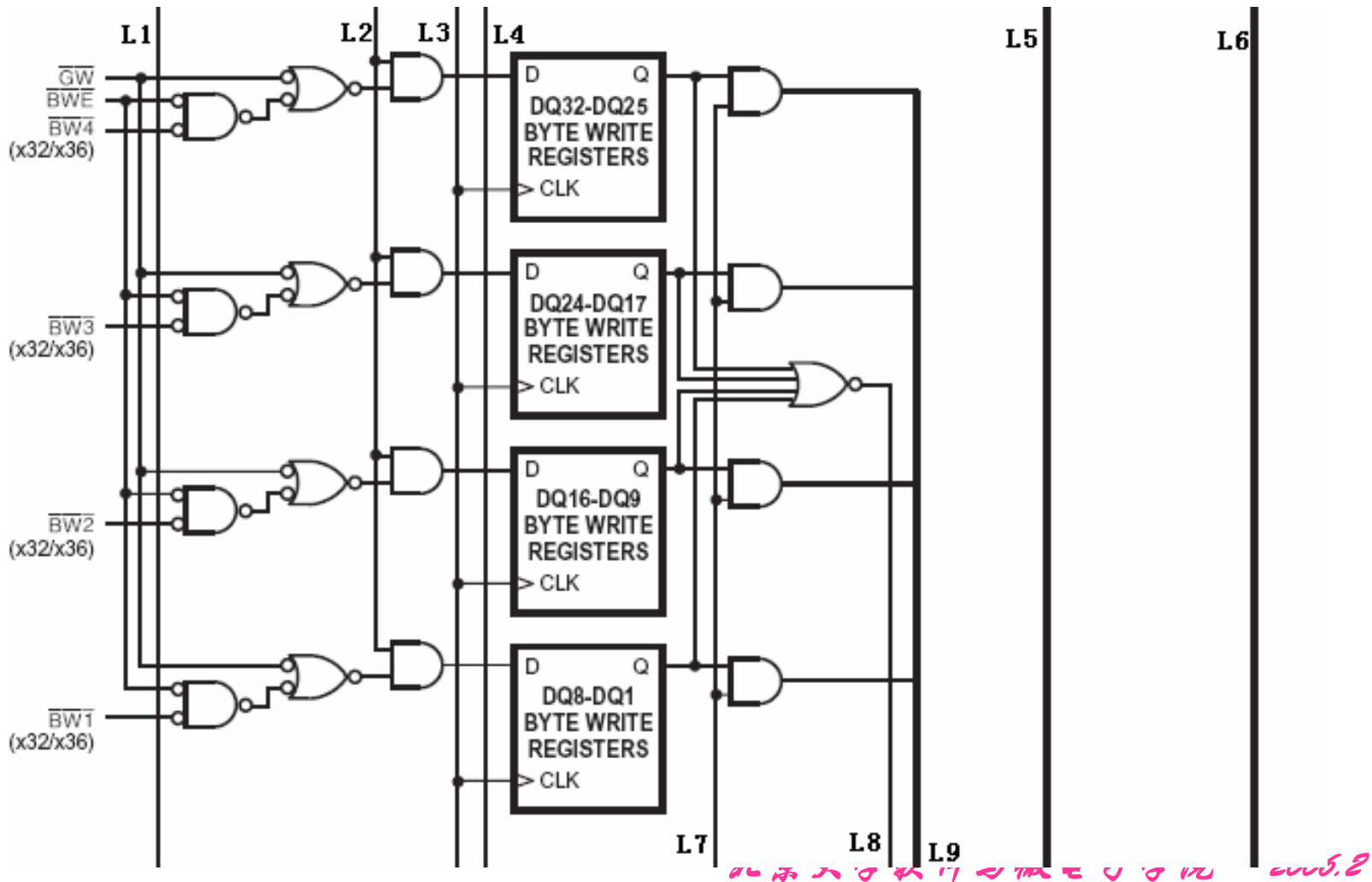
存储器接口-SRAM (13)

3. SSRAM-结构(3-1) (IS64LF6432) 地址控制



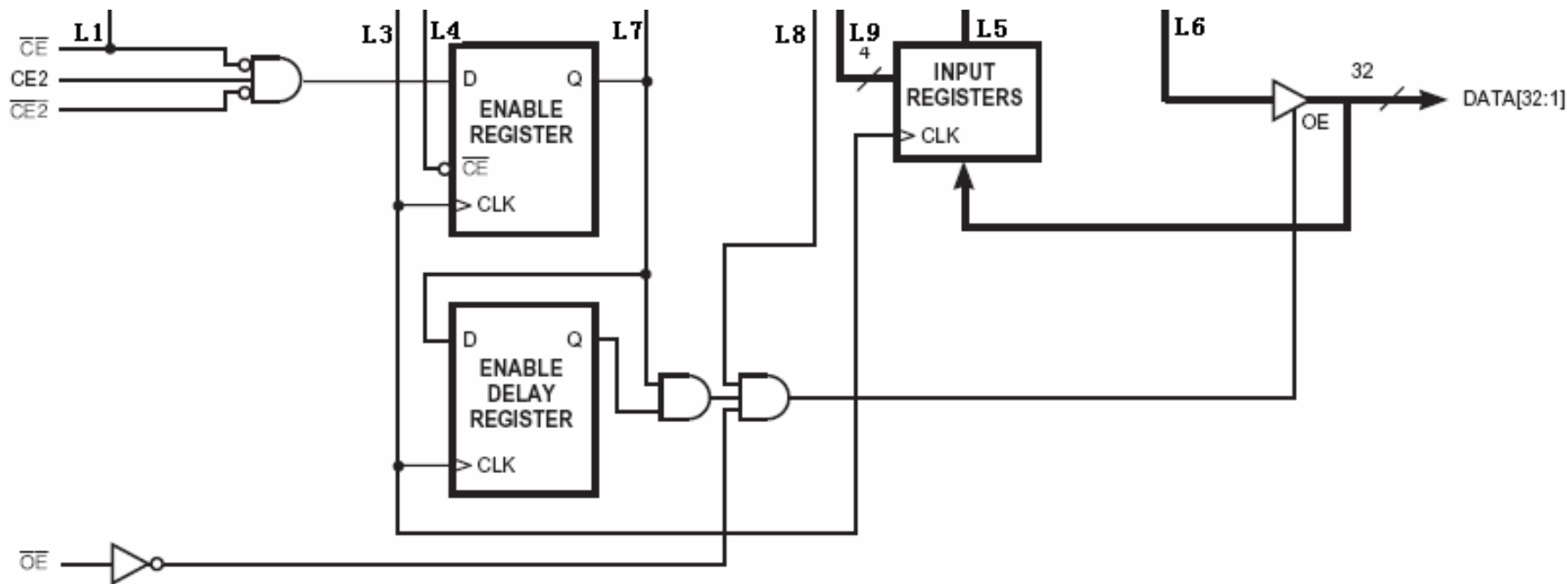
存储器接口-SRAM (14)

3. SSRAM-结构 (3-2) (IS64LF6432) (数据位段控制)



存储器接口-SRAM (13)

3. SSRAM-结构(3-3) (IS64LF6432) (数据I/O控制)



存储器接口-SRAM (16)

3. SSRAM-True Table 1 (IS64LF6432)

Operation	Address									
	Used	\overline{CE}	CE2	$\overline{CE2}$	\overline{ADSP}	\overline{ADSC}	\overline{ADV}	WRITE	\overline{OE}	DQ
Deselected, Power-down	None	H	X	X	X	L	X	X	X	High-Z
Deselected, Power-down	None	L	X	H	L	X	X	X	X	High-Z
Deselected, Power-down	None	L	L	X	L	X	X	X	X	High-Z
Deselected, Power-down	None	X	X	H	H	L	X	X	X	High-Z
Deselected, Power-down	None	X	L	X	H	L	X	X	X	High-Z
Read Cycle, Begin Burst	External	L	H	L	L	X	X	X	X	Q
Read Cycle, Begin Burst	External	L	H	L	H	L	X	Read	X	Q
Write Cycle, Begin Burst	External	L	H	L	H	L	X	Write	X	D
Read Cycle, Continue Burst	Next	X	X	X	H	H	L	Read	L	Q
Read Cycle, Continue Burst	Next	X	X	X	H	H	L	Read	H	High-Z
Read Cycle, Continue Burst	Next	H	X	X	X	H	L	Read	L	Q
Read Cycle, Continue Burst	Next	H	X	X	X	H	L	Read	H	High-Z
Write Cycle, Continue Burst	Next	X	X	X	H	H	L	Write	X	D
Write Cycle, Continue Burst	Next	H	X	X	X	H	L	Write	X	D
Read Cycle, Suspend Burst	Current	X	X	X	H	H	H	Read	L	Q
Read Cycle, Suspend Burst	Current	X	X	X	H	H	H	Read	H	High-Z
Read Cycle, Suspend Burst	Current	H	X	X	X	H	H	Read	L	Q
Read Cycle, Suspend Burst	Current	H	X	X	X	H	H	Read	H	High-Z
Write Cycle, Suspend Burst	Current	X	X	X	H	H	H	Write	X	D
Write Cycle, Suspend Burst	Current	H	X	X	X	H	H	Write	X	D

存储器接口-SRAM (17)

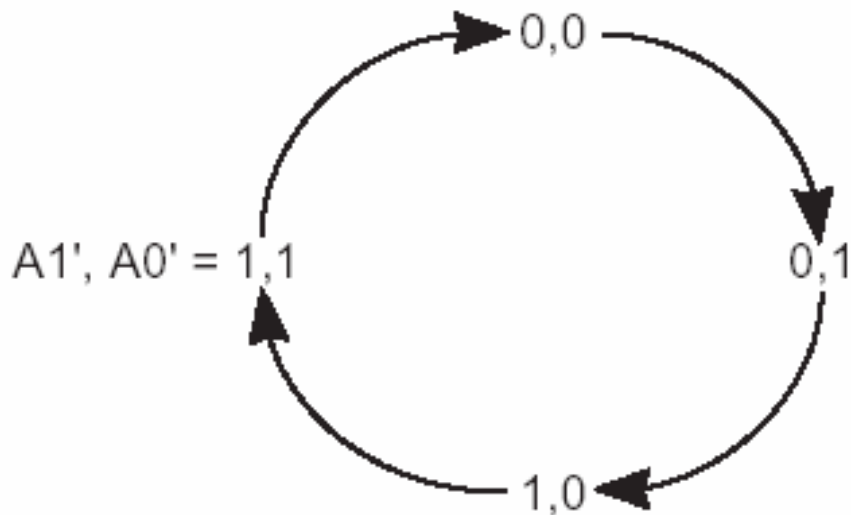
3. SSRAM-True Table 2 (IS64LF6432)

Function	\overline{GW}	\overline{BWE}	$\overline{BW1}$	$\overline{BW2}$	$\overline{BW3}$	$\overline{BW4}$
Read	H	H	X	X	X	X
Read	H	L	H	H	H	H
Write Byte 1	H	L	L	H	H	H
Write All Bytes	H	L	L	L	L	L
Write All Bytes	L	X	X	X	X	X

存储器接口-SRAM (18)

3. SSRAM-Interleaved Burst Address Table

External Address A1 A0	1st Burst Address A1 A0	2nd Burst Address A1 A0	3rd Burst Address A1 A0
00	01	10	11
01	00	11	10
10	11	00	01
11	10	01	00

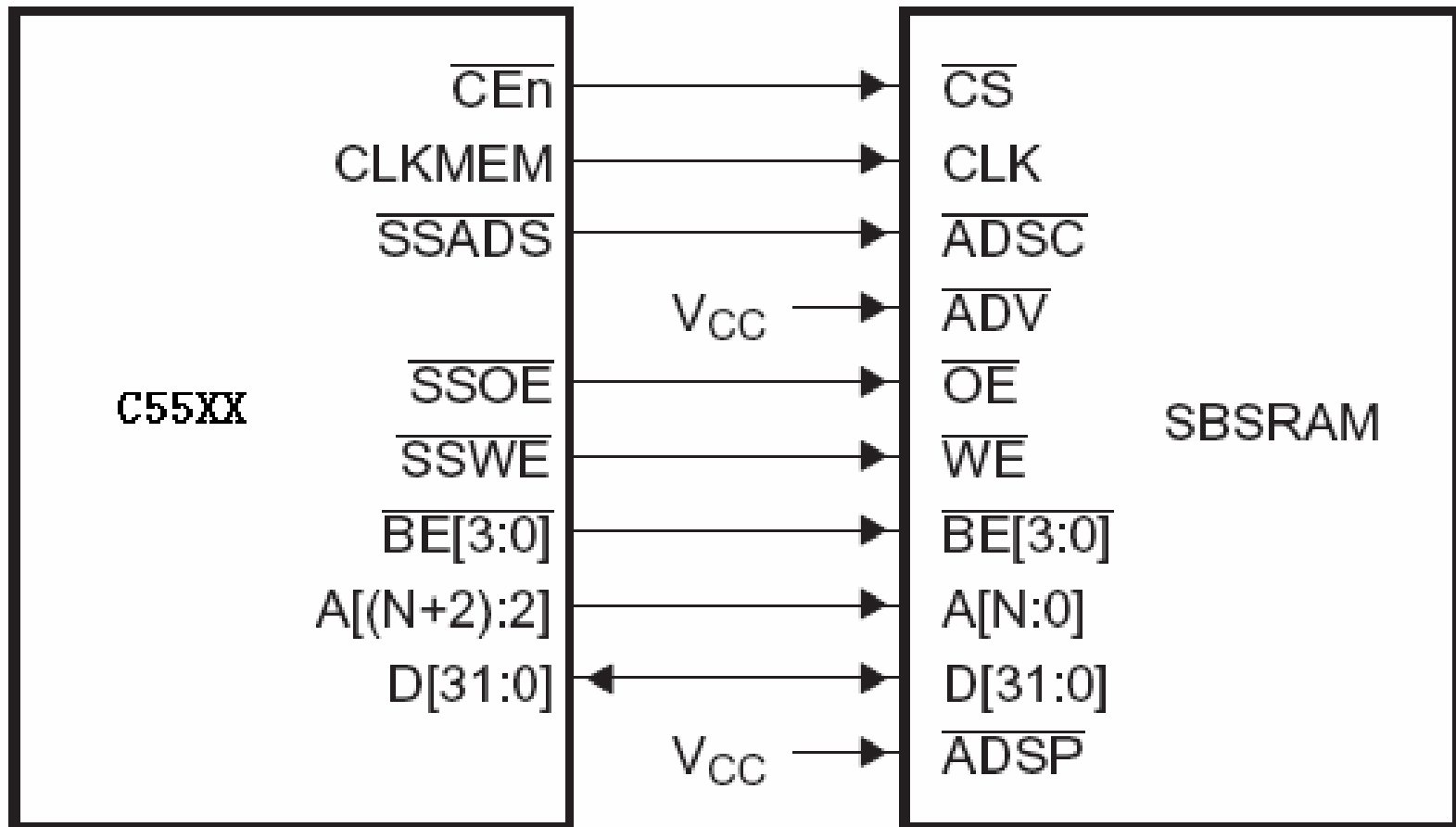


How about liner burst address?

存储器接口-SRAM (18)

3. SSRAM-The Interface With CPU

- Processor with SSRAM Interface



存储器接口-SRAM (19)

4. 编程

- 指针

```
int * mydata_ad= (int *) 0x800000;  
*mydata_ad= 11;
```

- 变量

```
int mydata;  
mydata=11;
```

存储器接口-DRAM (1)

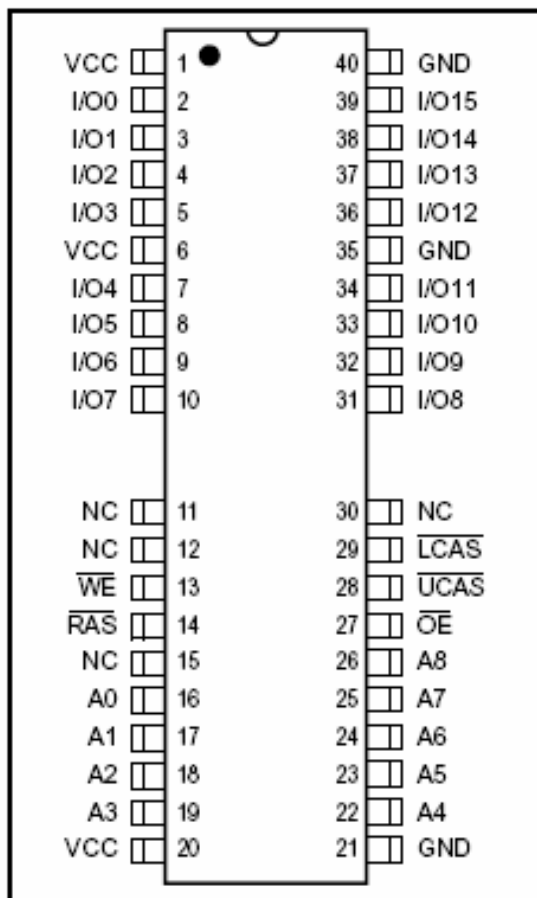
1. 概述

- DRAM (Dynamic RAM)
 - DRAM
 - SDRAM
- Interface
 - /RE, /WE
 - /RAS, /CAS
 - Add
 - Data
 - CLK, (for SDRAM)

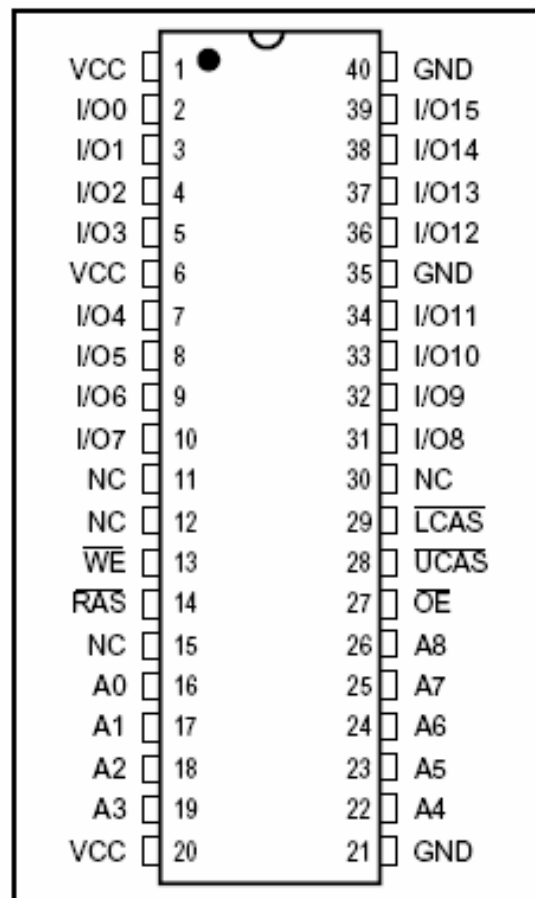
存储器接口-DRAM(2)

2. DRAM (41C16256)-Package

40-Pin TSOP (Type II)

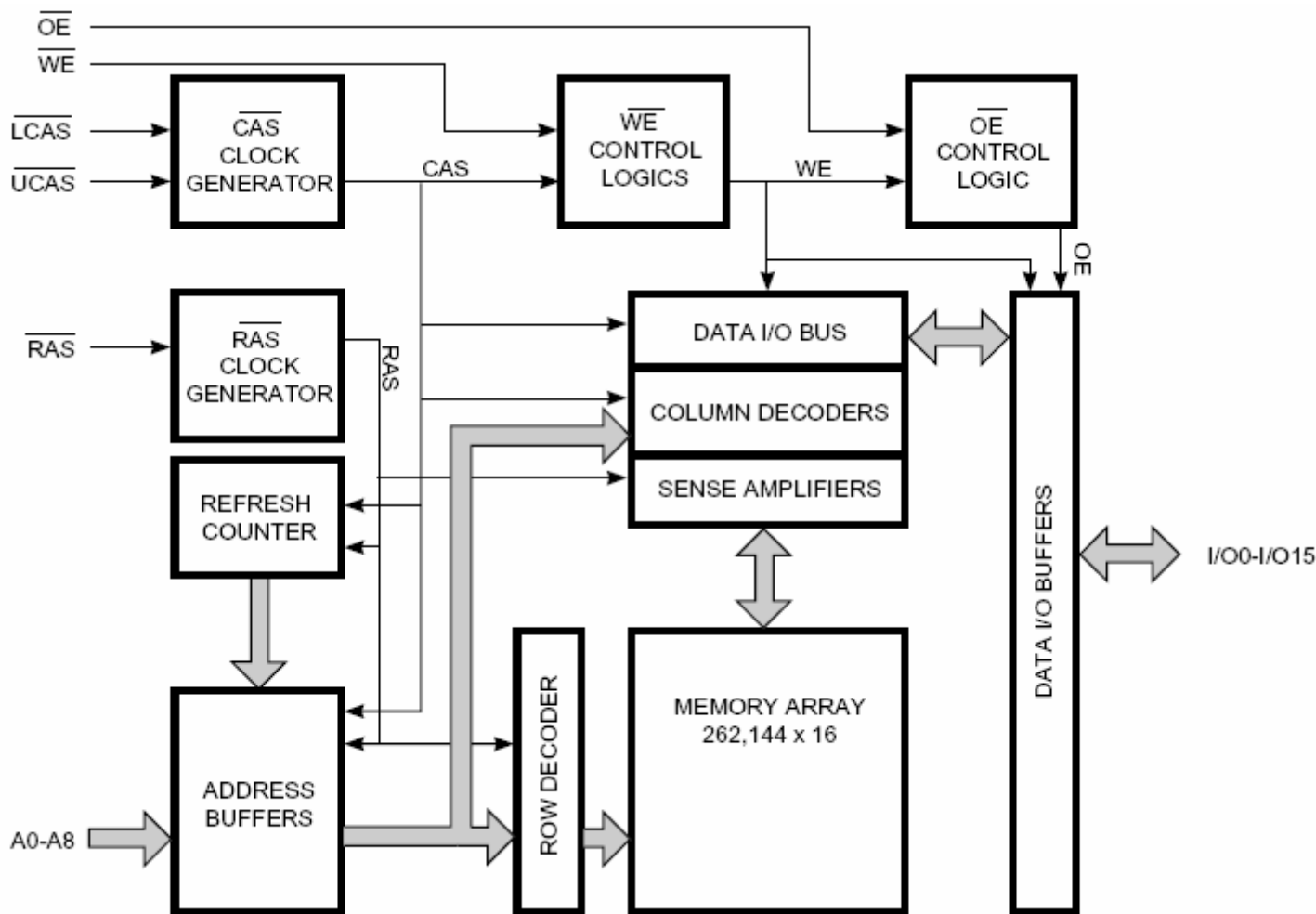


40-Pin SOJ



存储器接口-DRAM(3)

3. DRAM (41C16256)-Function Block



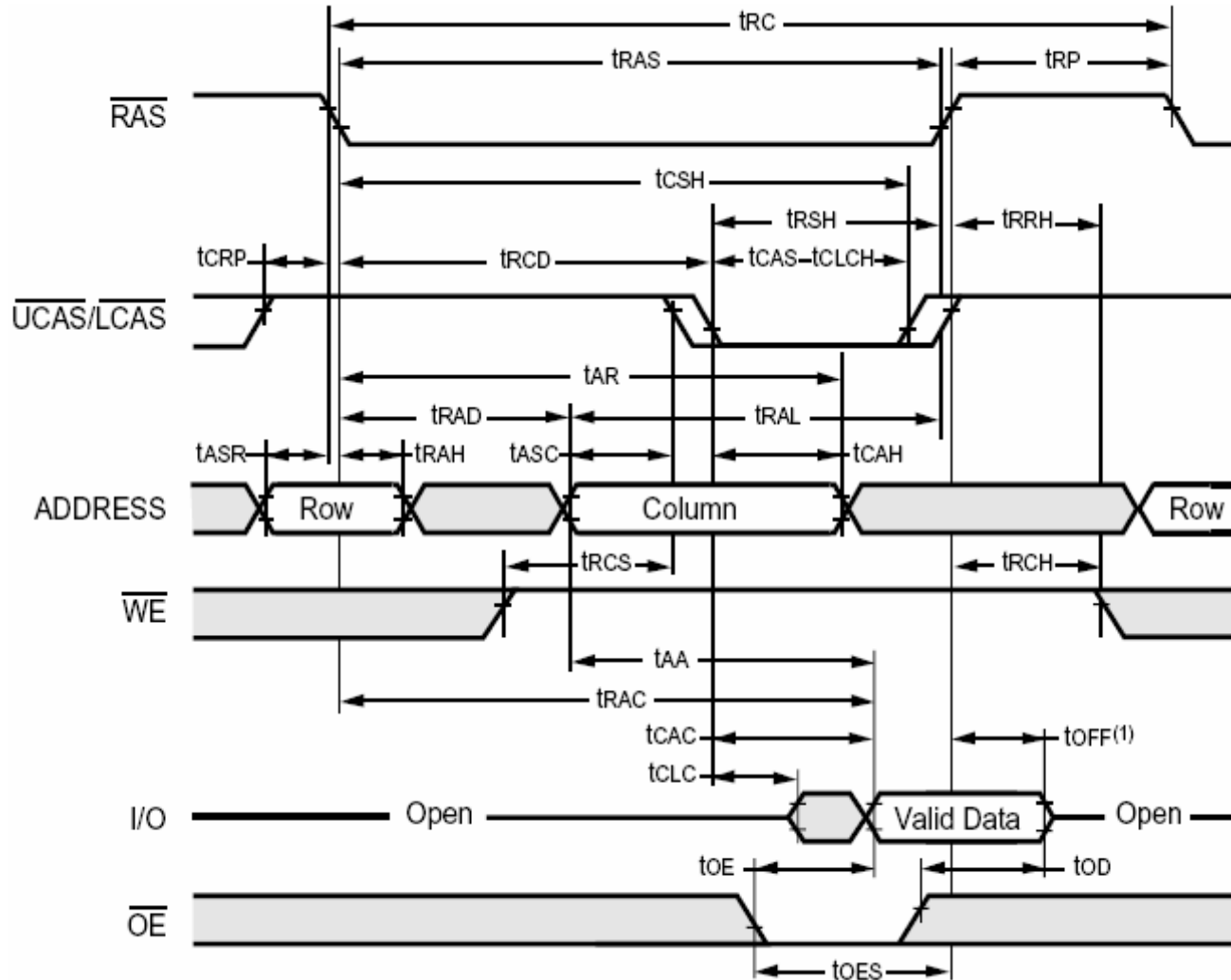
存储器接口-DRAM(4)

3. DRAM (41C16256)-True Table

Function	\overline{RAS}	\overline{LCAS}	\overline{UCAS}	\overline{WE}	\overline{OE}	Address	tr/tc	I/O
Standby	H	H	H	X	X	X		High-Z
Read: Word	L	L	L	H	L	ROW/COL		Dout
Read: Lower Byte	L	L	H	H	L	ROW/COL		Lower Byte, Dout Upper Byte, High-Z
Read: Upper Byte	L	H	L	H	L	ROW/COL		Lower Byte, High-Z Upper Byte, Dout
Write: Word (Early Write)	L	L	L	L	X	ROW/COL		Din
Write: Lower Byte (Early Write)	L	L	H	L	X	ROW/COL		Lower Byte, Din Upper Byte, High-Z
Write: Upper Byte (Early Write)	L	H	L	L	X	ROW/COL		Lower Byte, High-Z Upper Byte, Din
Read-Write	L	L	L	H→L	L→H	ROW/COL		Dout, Din
Hidden Refresh	Read	L→H→L	L	L	H	L	ROW/COL	Dout
	Write	L→H→L	L	L	L	X	ROW/COL	Dout
\overline{RAS} -Only Refresh	L	H	H	X	X	ROW/NA		High-Z
CBR Refresh	H→L	L	L	X	X	X		High-Z

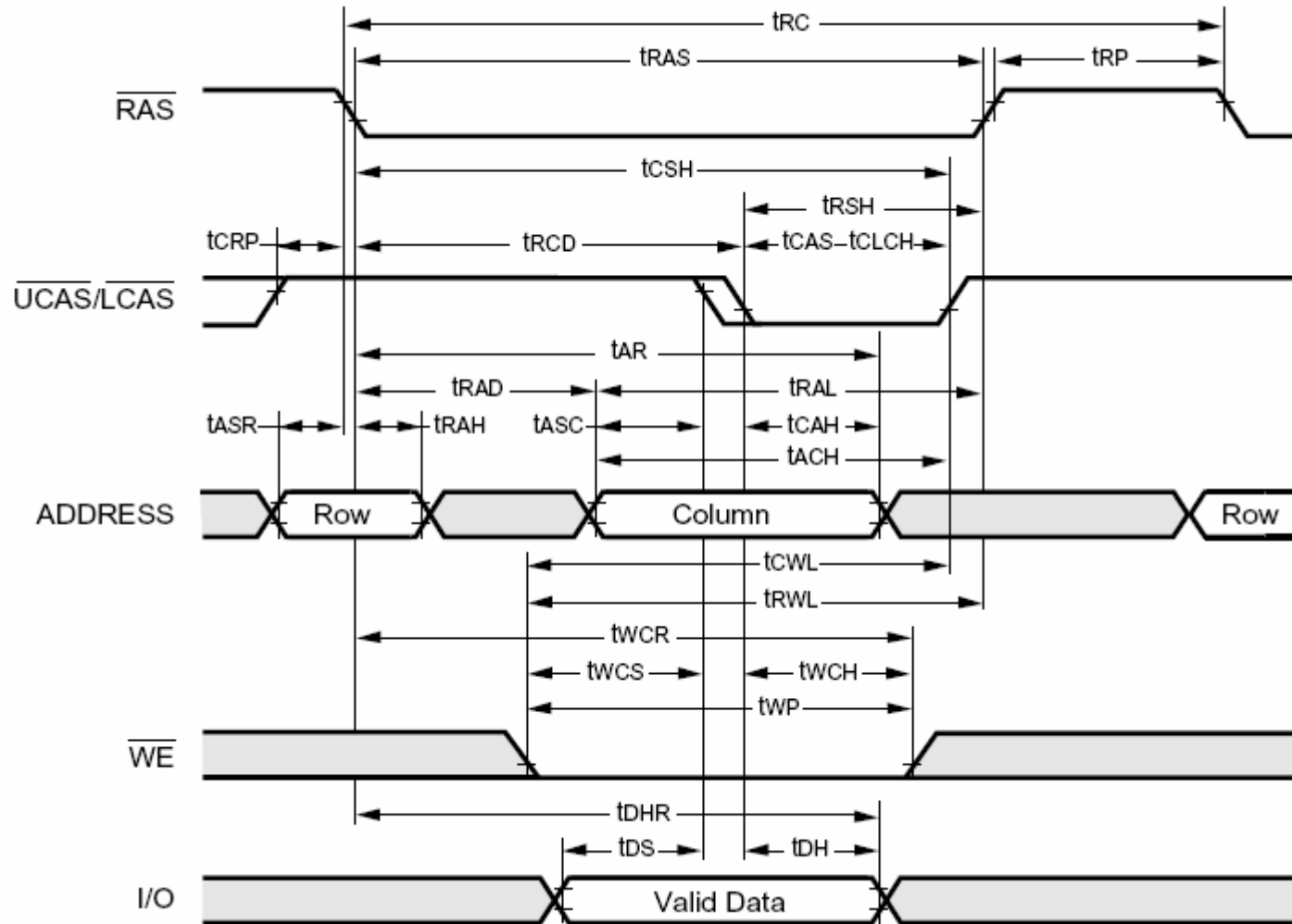
存储器接口-DRAM(5)

3. DRAM (41C16256)-Timing (Read Cycle)



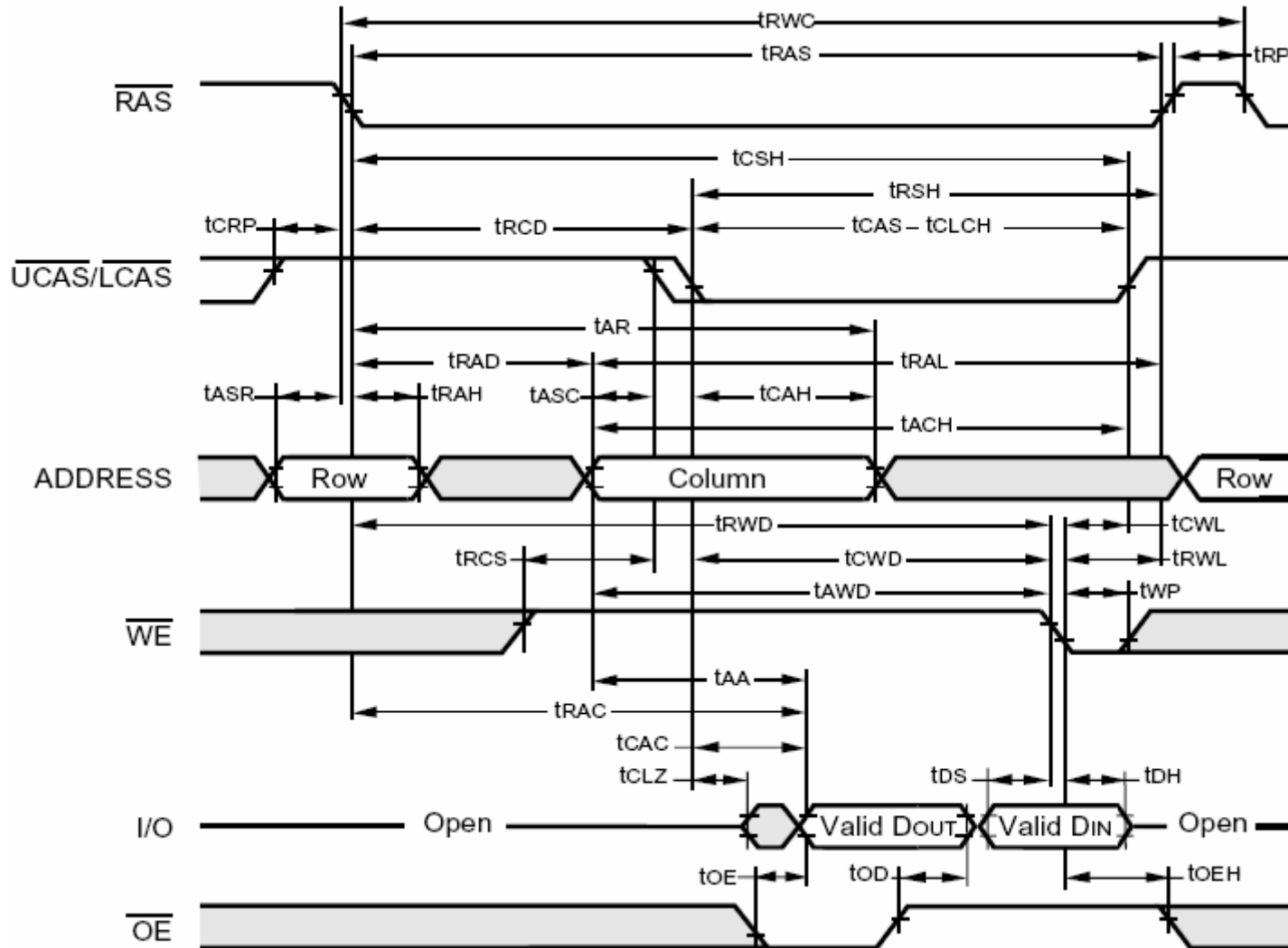
存储器接口-DRAM(6)

3. DRAM (41C16256)-Timing (Write Cycle)



存储器接口-DRAM(7)

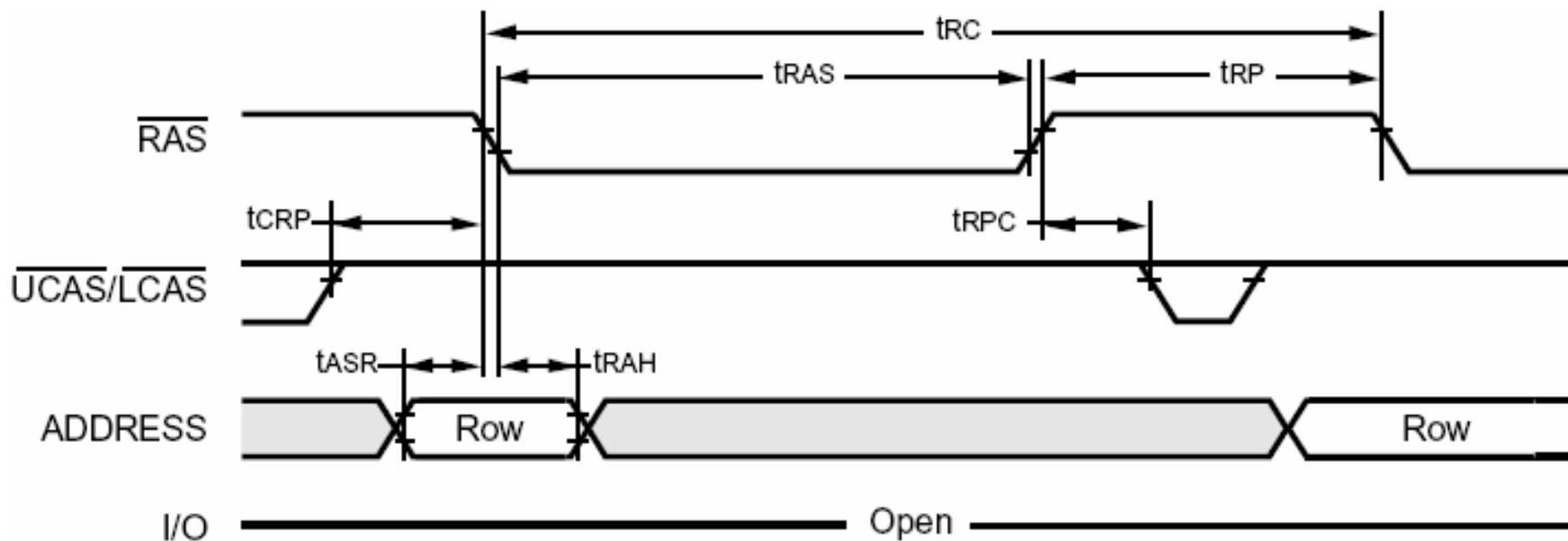
3. DRAM (41C16256)-Timing (Read Write Cycle)



存储器接口-DRAM(8)

3. DRAM (41C16256)-Timing (Refresh)

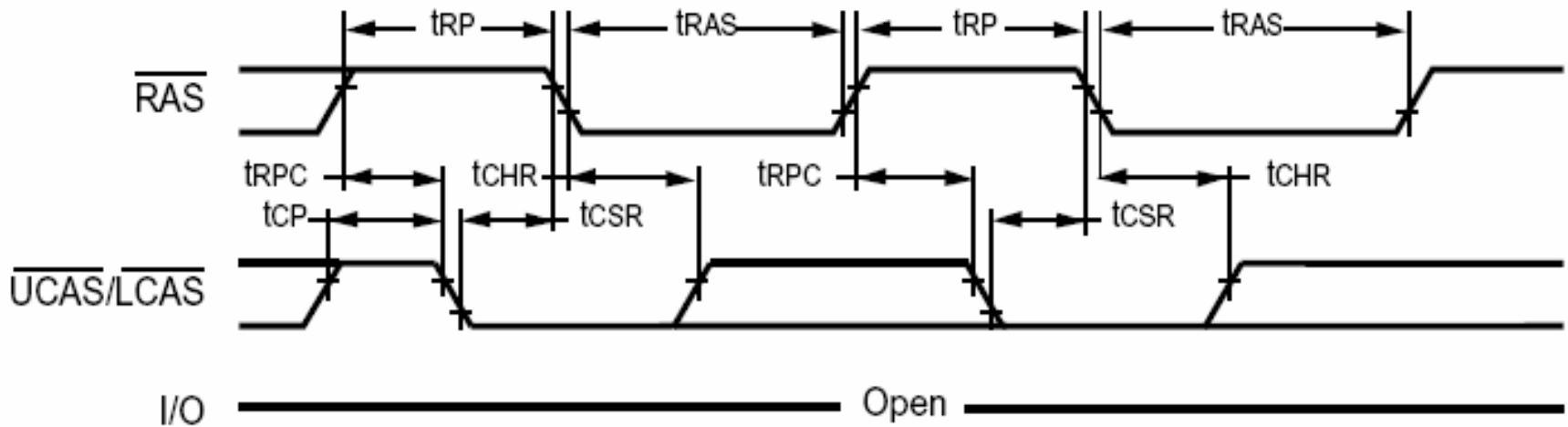
- /RAS-only



存储器接口-DRAM(9)

3. DRAM (41C16256)-Timing (Refresh) (Cont.)

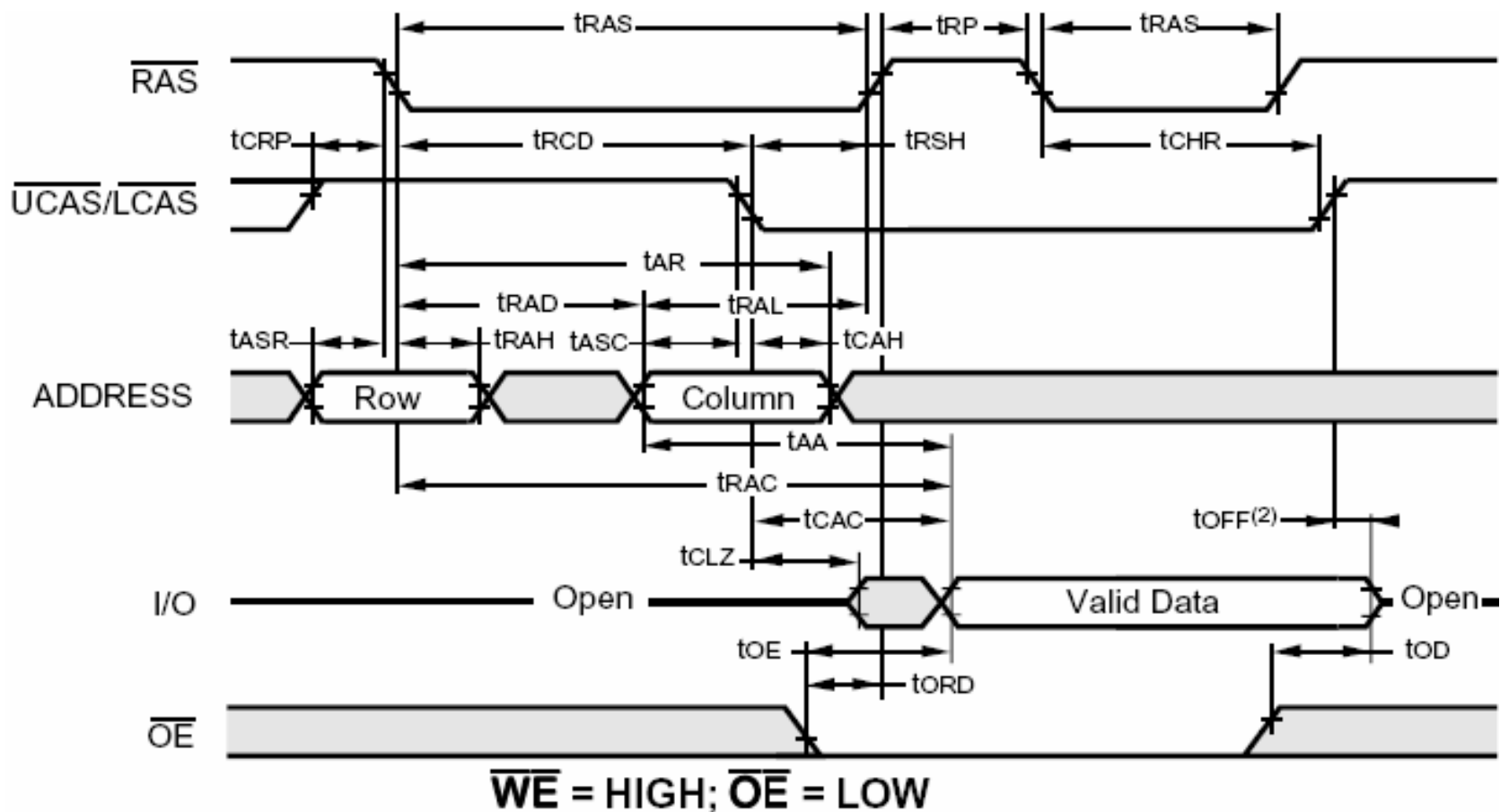
- /CAS-Before-/RAS



存储器接口-DRAM(10)

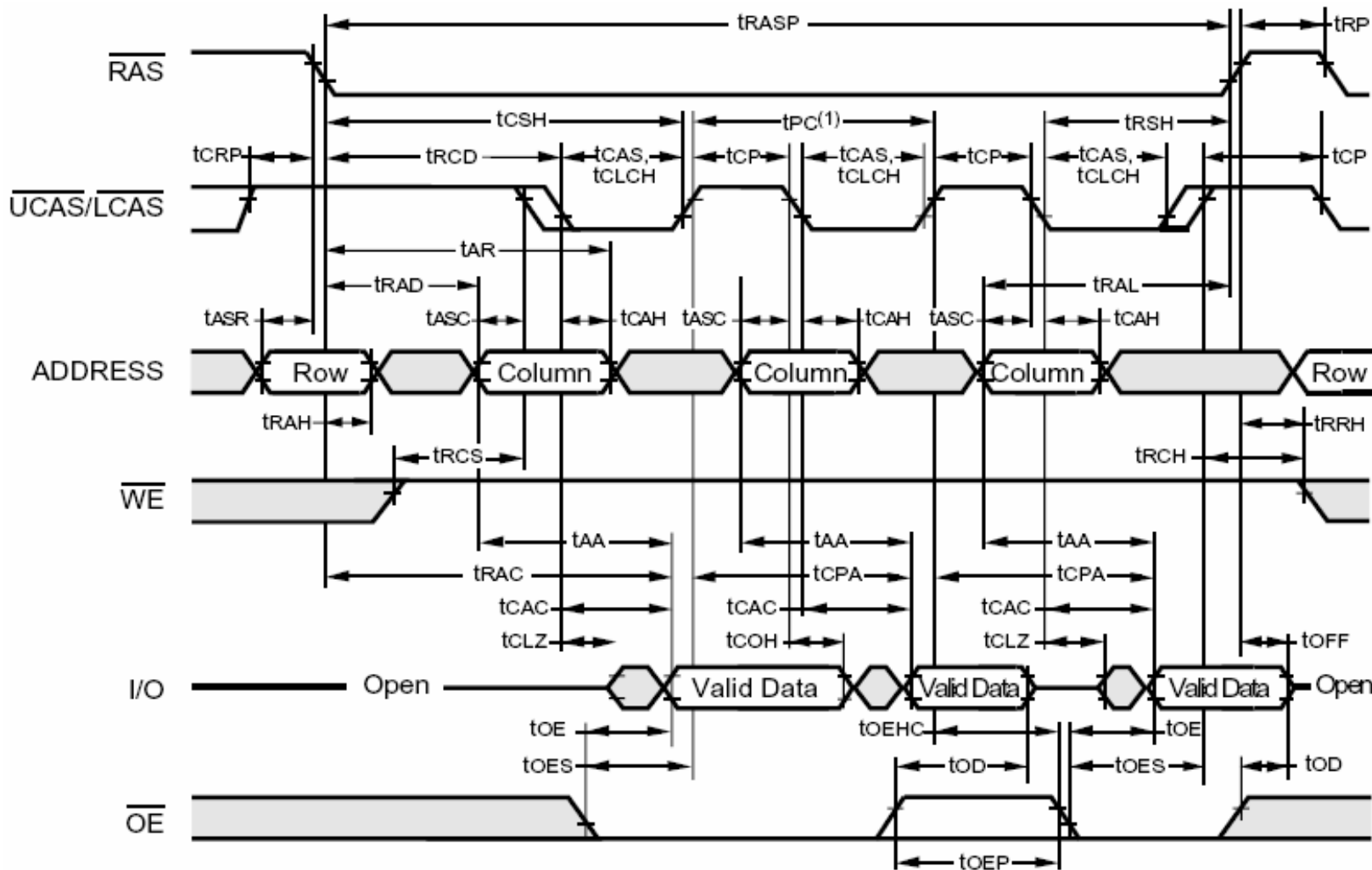
3. DRAM (41C16256)-Timing (Refresh) (Cont.)

- Hidden



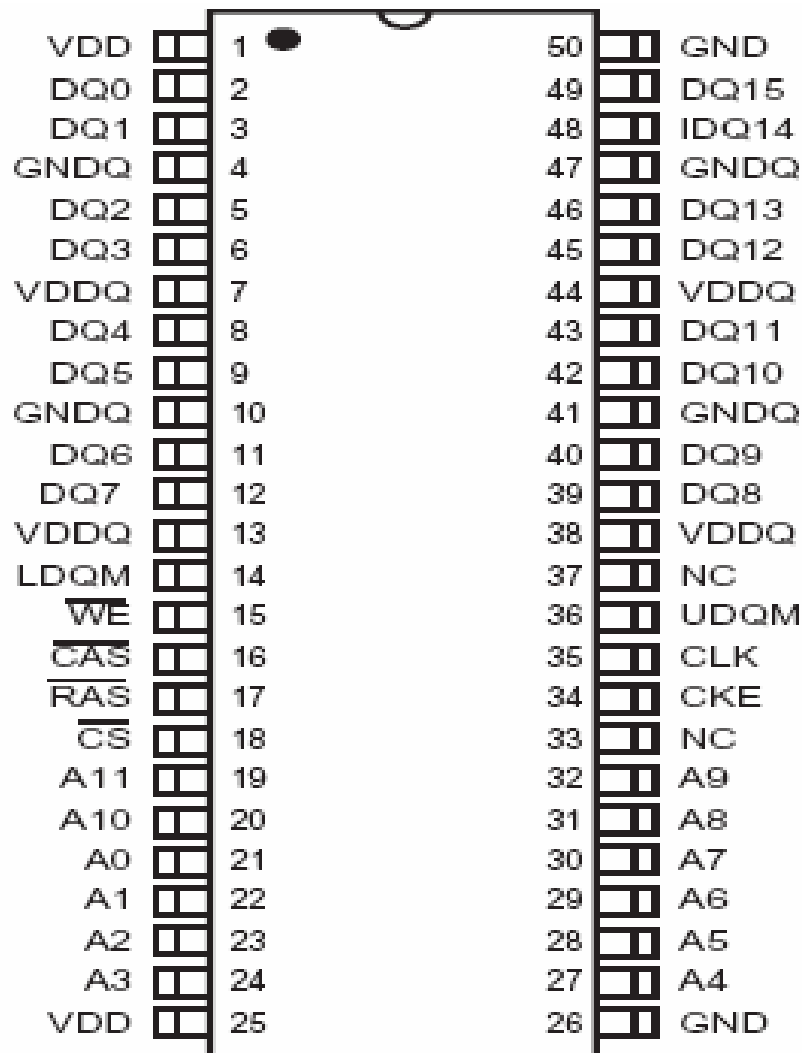
存储器接口-DRAM(11)

3. DRAM (41C16256)-Timing (page mode Read)



存储器接口-DRAM(12)

4. SDRAM (42S16100)-Package



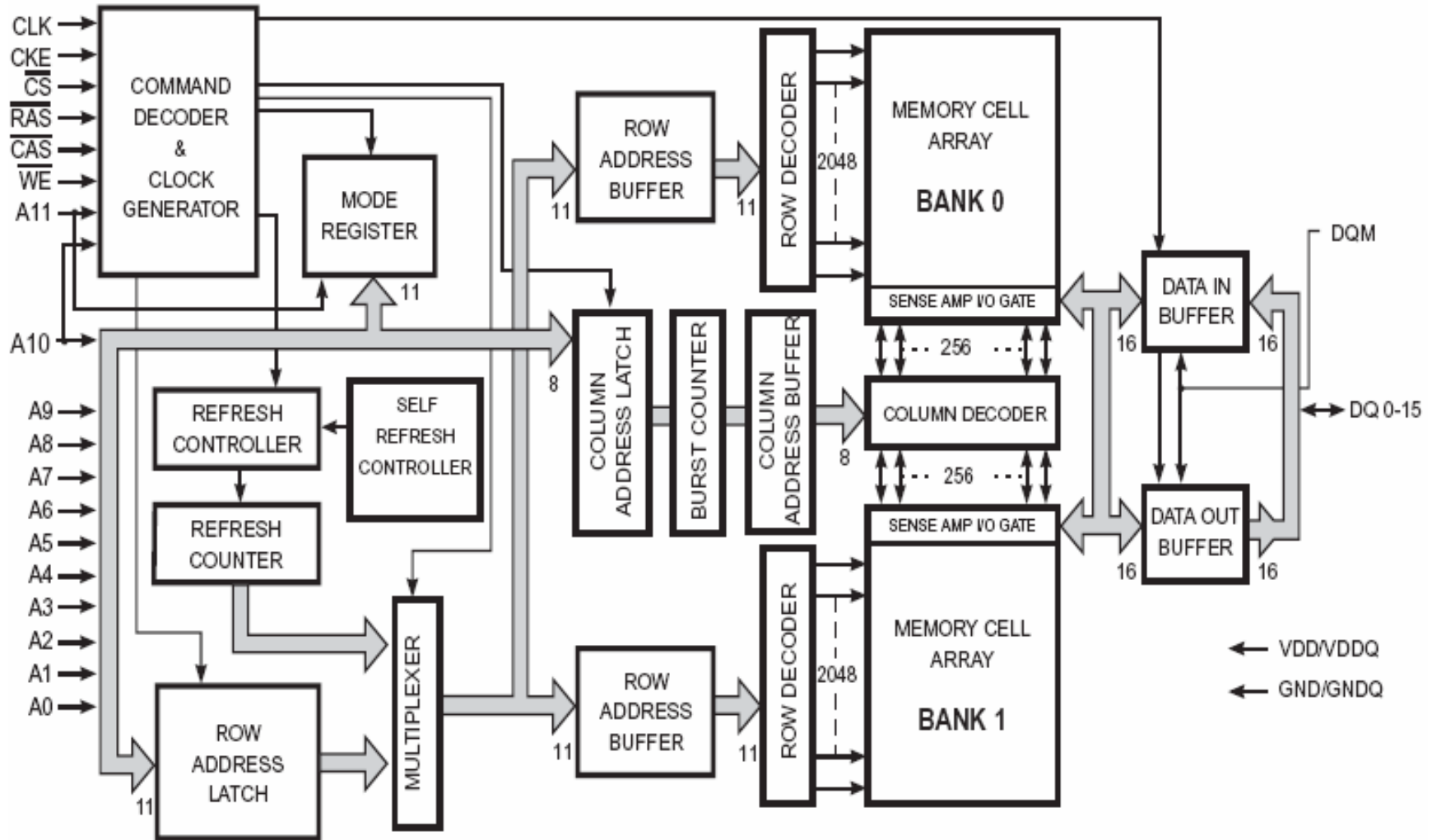
存储器接口-DRAM(13)

4. SDRAM (42S16100)-Specific Pins

- CKE: 时钟使能控制
 - HIGH: 有效
 - LOW: 无效 (睡眠, 刷新)
- LDQM, UDQM: 高低字节控制位
 - Low: 有效
 - Read: same /OE, output data
 - Write: in buffer enabled, data can be write into
 - High: 无效
 - Read: output high impedance
 - Write: data can't be write into
- A11: Bank selection
- A10: R/W Selection (复用)

存储器接口-DRAM(14)

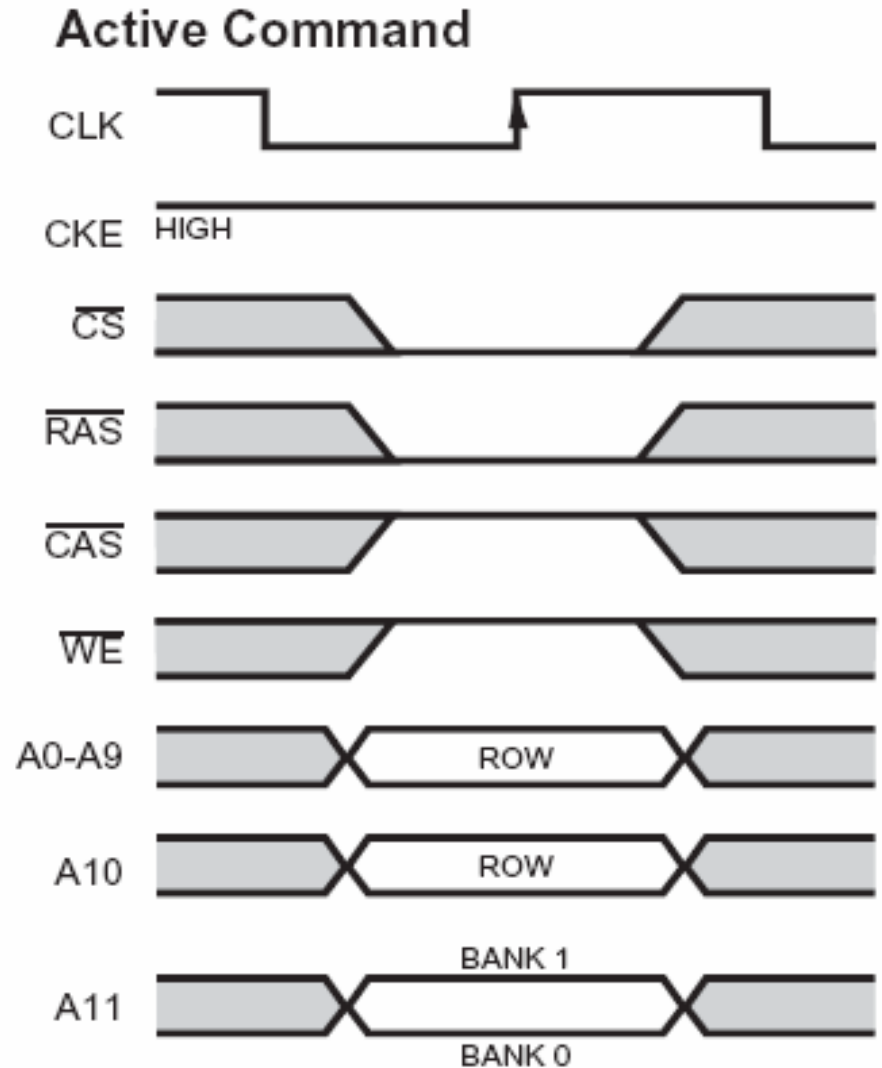
4. SDRAM (42S16100)-Function Block



存储器接口-DRAM(15)

4. SDRAM (42S16100)-COMMAND

- Format (the state of relative pins at rising edge of CLK)



存储器接口-DRAM(16)

4. SDRAM (42S16100)-COMMAND (cont.)

■ Commands

Symbol	Command	CKE		\overline{CS}	\overline{RAS}	\overline{CAS}	\overline{WE}	DQM	A11	A10	A9-A0	I/On
		n-1	n									
MRS	Mode Register Set	H	X	L	L	L	L	X	OP CODE			X
REF	Auto-Refresh	H	H	L	L	L	H	X	X	X	X	HIGH-Z
SREF	Self-Refresh	H	L	L	L	L	H	X	X	X	X	HIGH-Z
PRE	Precharge Selected Bank	H	X	L	L	H	L	X	BS	L	X	X
PALL	Precharge Both Banks	H	X	L	L	H	L	X	X	H	X	X
ACT	Bank Activate	H	X	L	L	H	H	X	BS	Row	Row	X
WRIT	Write	H	X	L	H	L	L	X	BS	L	Column	X
WRITA	Write With Auto-Precharge	H	X	L	H	L	L	X	BS	H	Column	X
READ	Read	H	X	L	H	L	H	X	BS	L	Column	X
READA	Read With Auto-Precharge	H	X	L	H	L	H	X	BS	H	Column	X
BST	Burst Stop	H	X	L	H	H	L	X	X	X	X	X
NOP	No Operation	H	X	L	H	H	H	X	X	X	X	X
DESL	Device Deselect	H	X	H	X	X	X	X	X	X	X	X
SBY	Clock Suspend / Standby Mode	L	X	X	X	X	X	X	X	X	X	X
ENB	Data Write / Output Enable	H	X	X	X	X	X	L	X	X	X	Active
MASK	Data Mask / Output Disable	H	X	X	X	X	X	H	X	X	X	HIGH-Z

存储器接口-DRAM(17)

4. SDRAM (42S16100)-COMMAND (cont.)
 - MRS (Mode Register Setting)

Input Pin	Field
A11, A10, A9, A8	Mode Options
A6, A5, A4	$\overline{\text{CAS}}$ Latency
A3	Burst Type
A2, A1, A0	Burst Length

存储器接口-DRAM(18)

4. SDRAM (42S16100)-COMMAND (cont.)

■ Mode Register (cont.)

	M6	M5	M4	$\overline{\text{CAS}}$ Latency
Latency Mode 0	0	1	0	2
	0	1	1	3

M11	M10	M9	M8	M7	Write Mode
X	X	0	0	0	Mode Register Set
X	X	1	0	0	Burst Read & Single Write
0	0	0	0	0	Reserved Test Set

Note: The reserved values are not listed here

存储器接口-DRAM(19)

4. SDRAM (42S16100)-COMMAND (cont.)

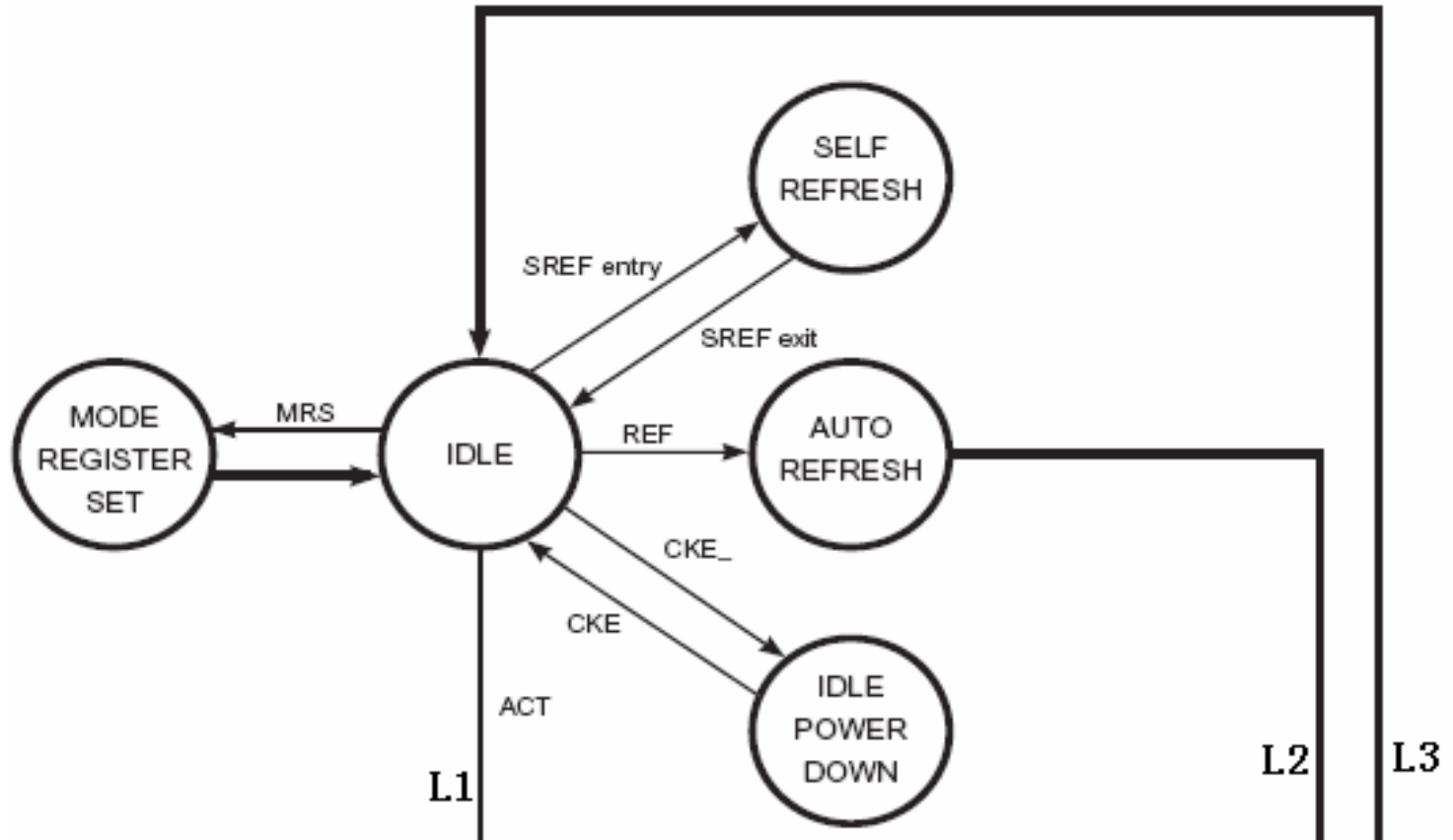
■ Mode Register

	M2	M1	M0	Sequential	Interleaved
BurstLength	0	0	0	1	1
	0	0	1	2	2
	0	1	0	4	4
	0	1	1	8	8
	1	1	1	Full Page	Reserved

	M3	Type
Burst Type	0	Sequential
	1	Interleaved

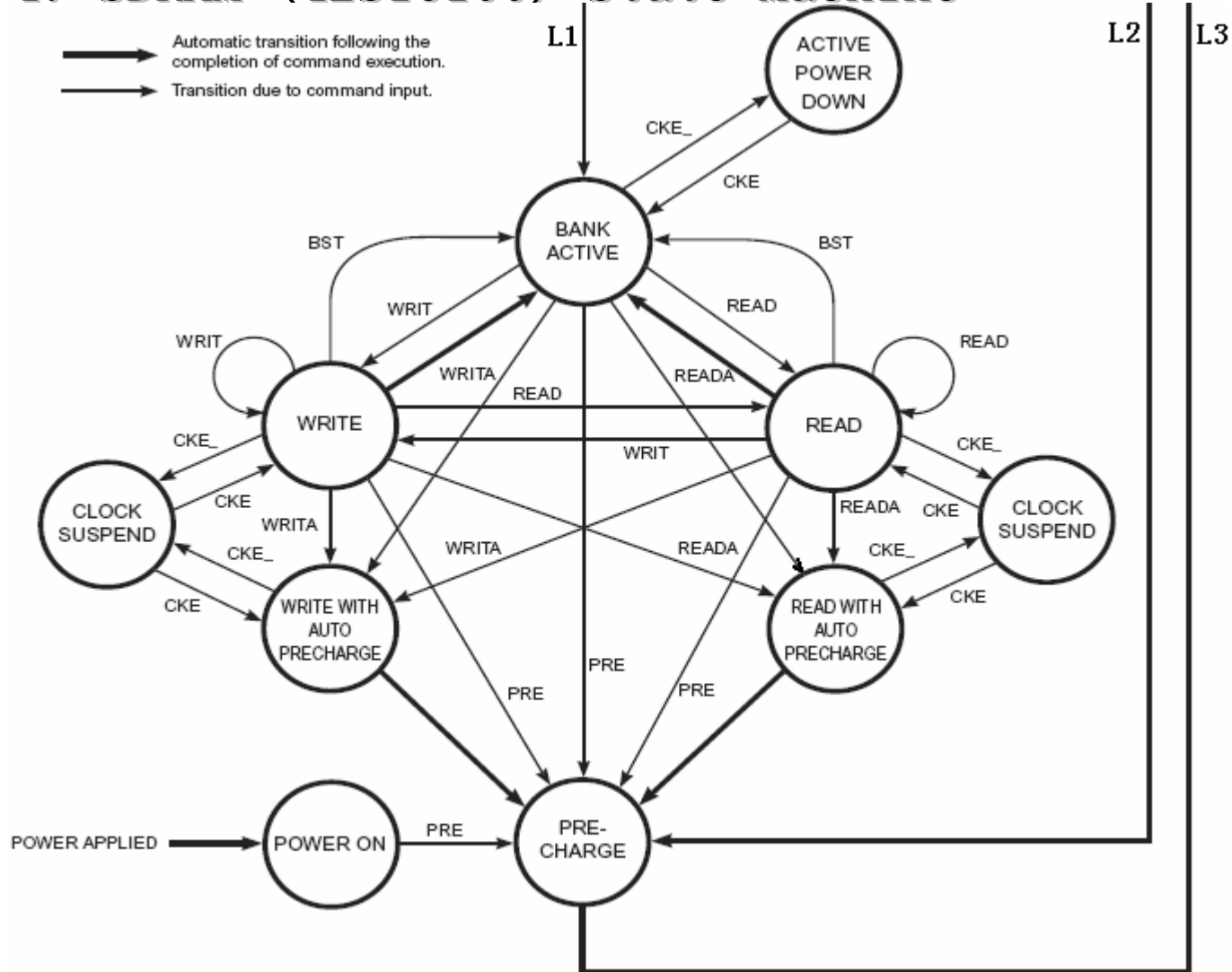
存储器接口-DRAM(20)

4. SDRAM (42S16100)-State machine-1



存储器接口-DRAM(21)

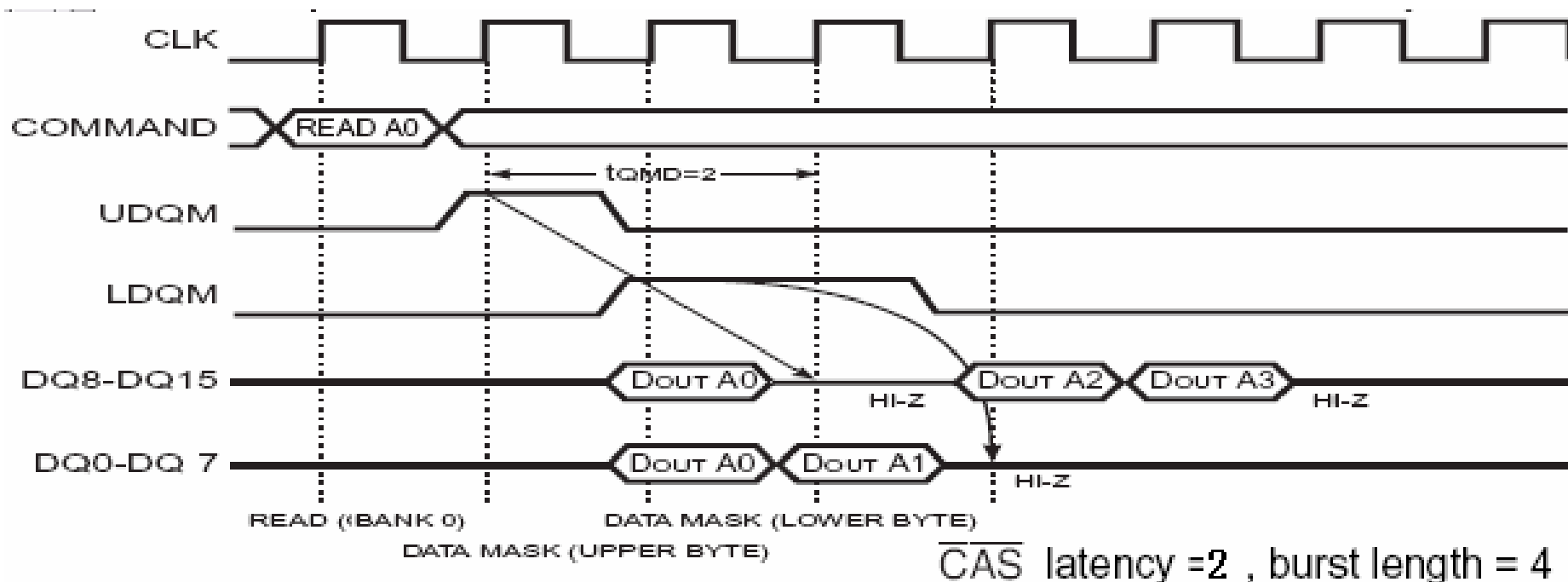
4. SDRAM (42S16100)-State Machine



存储器接口-DRAM(22)

4. SDRAM (42S16100)-Timing

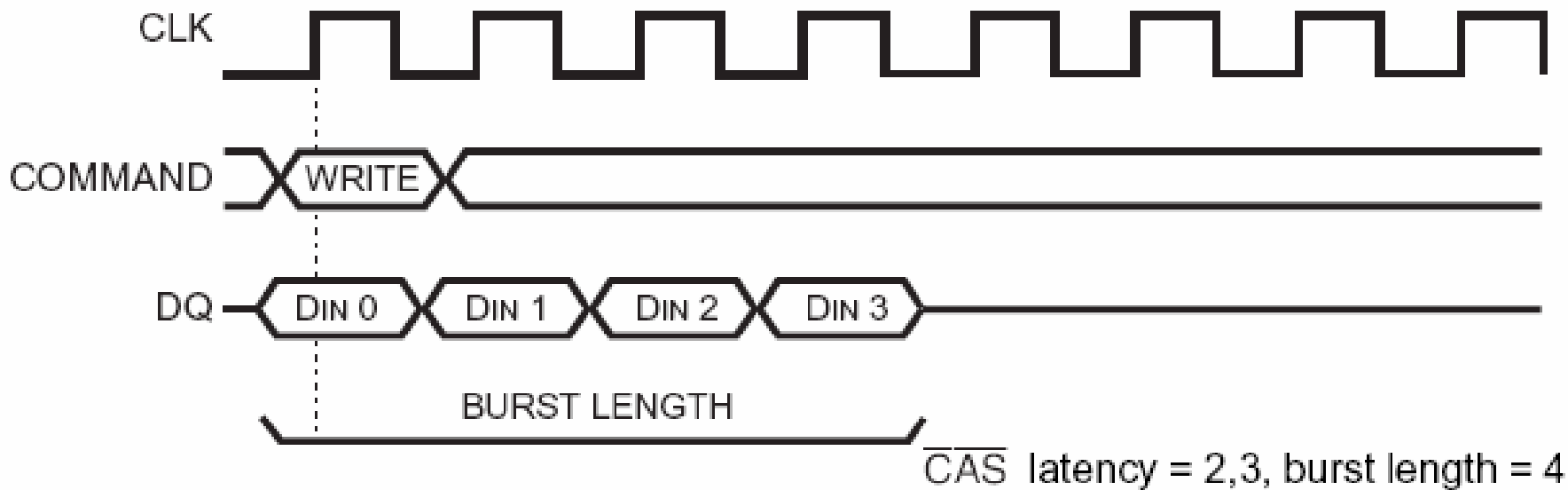
- Burst read



存储器接口-DRAM(23)

4. SDRAM (42S16100)-Timing (cont.)

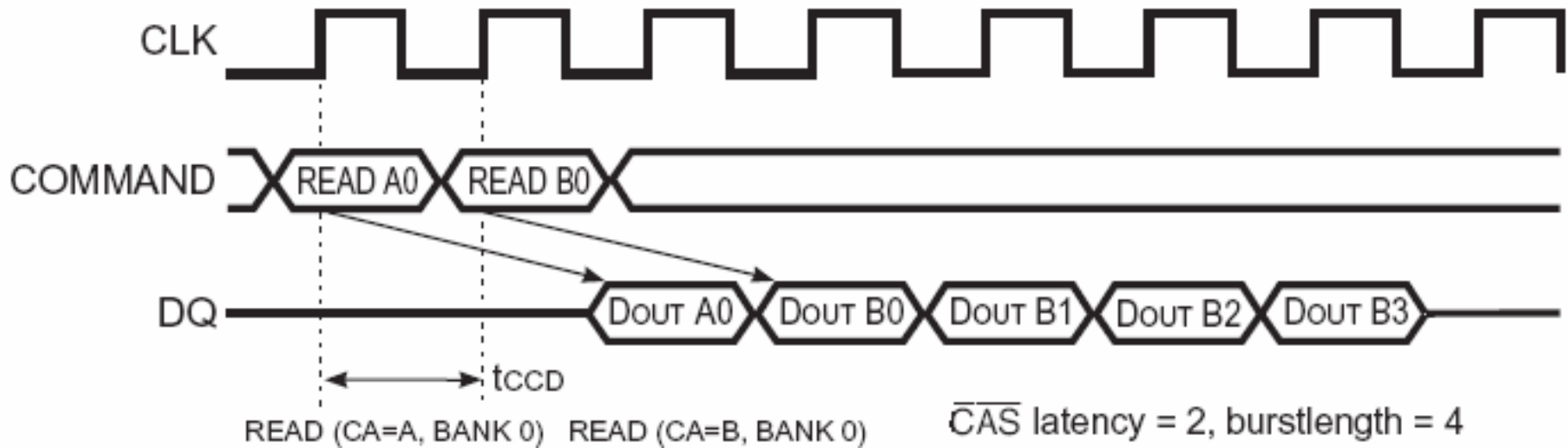
- Burst Write



存储器接口-DRAM(24)

4. SDRAM (42S16100)-Timing (cont.)

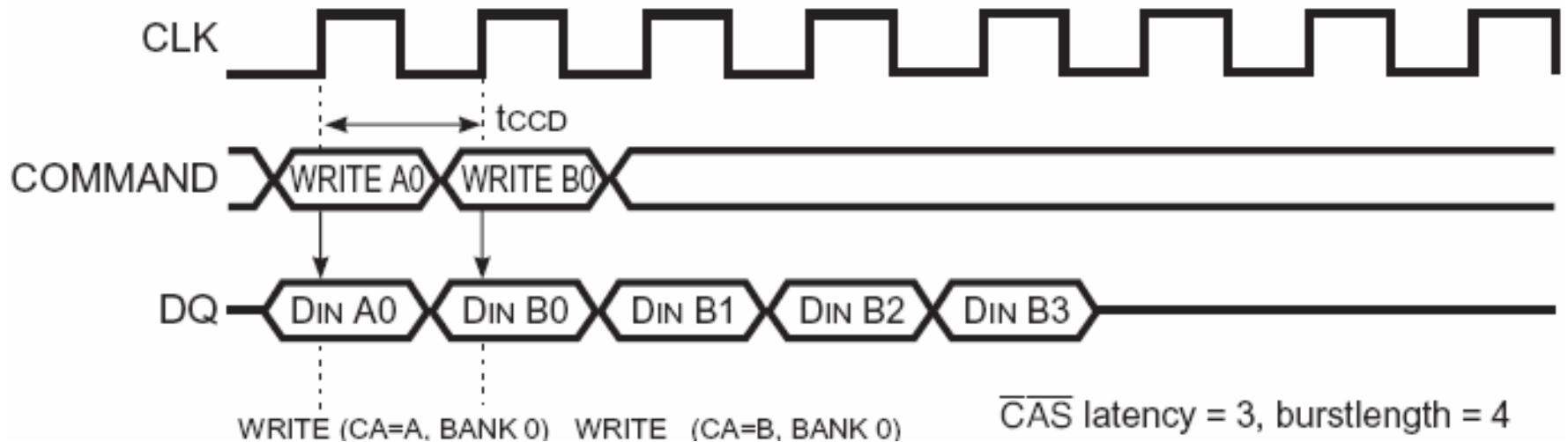
- Interval between read command



存储器接口-DRAM(25)

4. SDRAM (42S16100)-Timing (cont.)

- Interval between write command



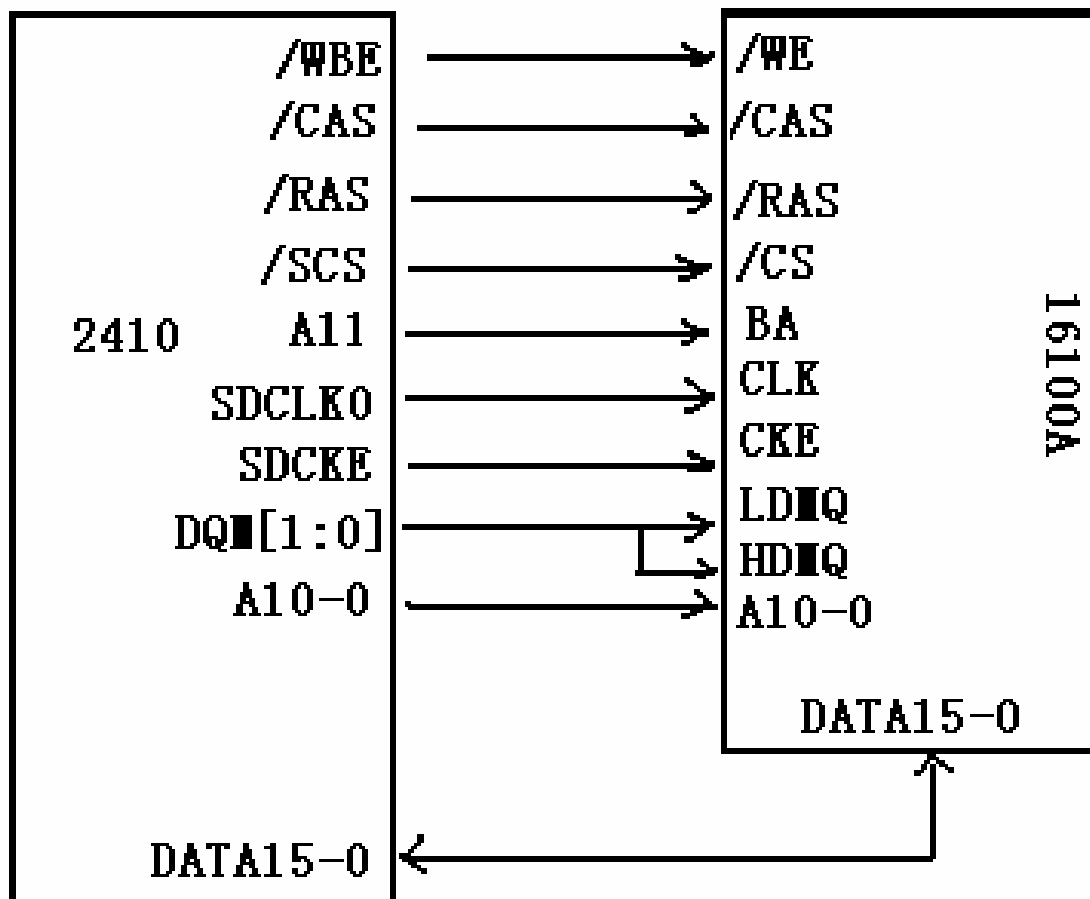
tccd at least one clock!

If WRITE before READ or WRITE after READ what will happen?

存储器接口-DRAM(26)

4. SDRAM (42S16100)- Connection

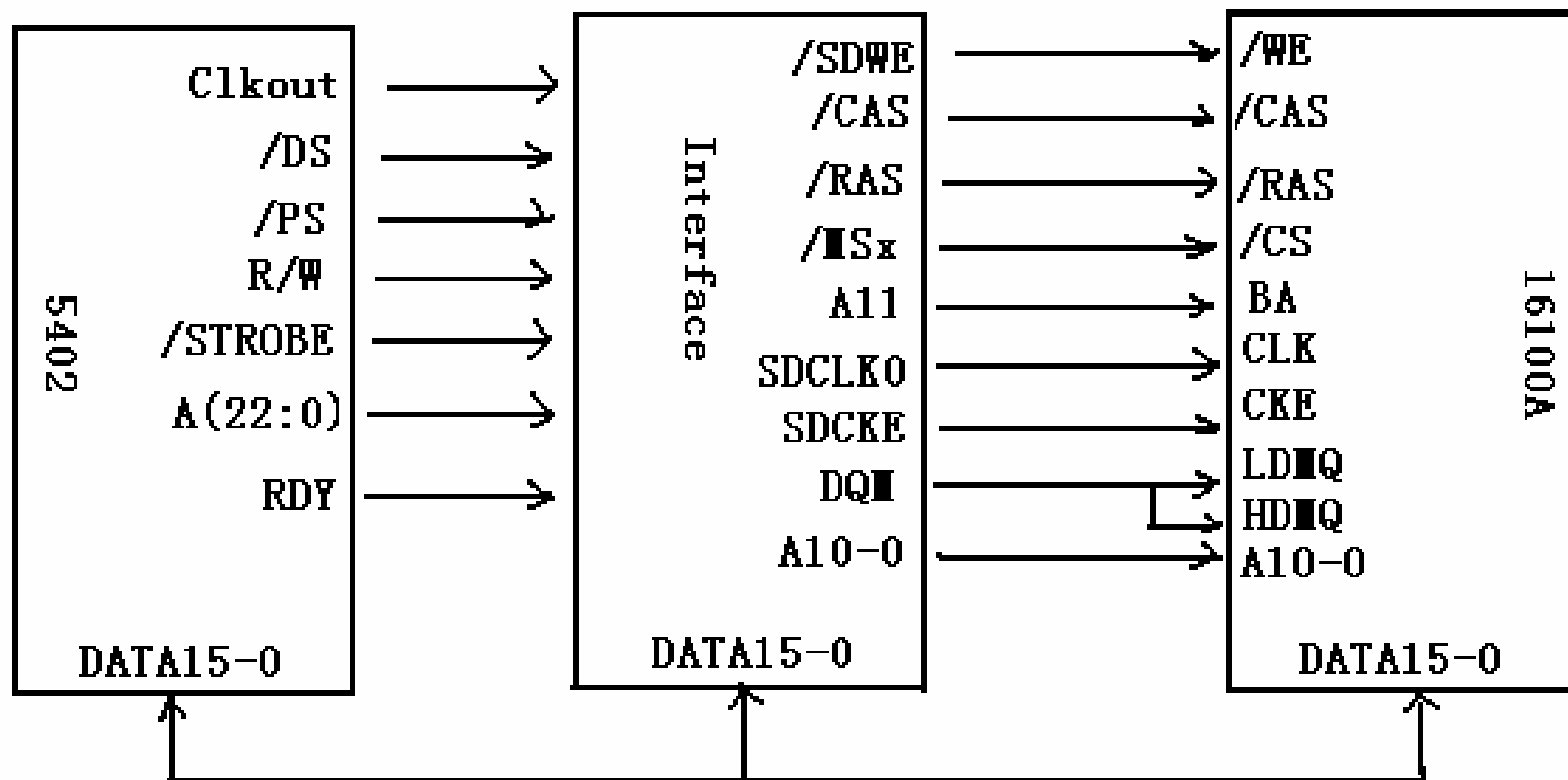
- With on chip interface

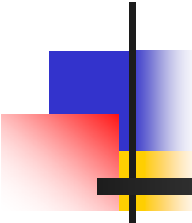


存储器接口-DRAM(27)

4. SDRAM (42S16100)- Connection

- Without on chip interface





存储器接口-DRAM(28)

使用**DRAM**，对编程有影响吗？



存储器接口-DRAM(28)

比较SRAM和DRAM在使用过程中的区别?

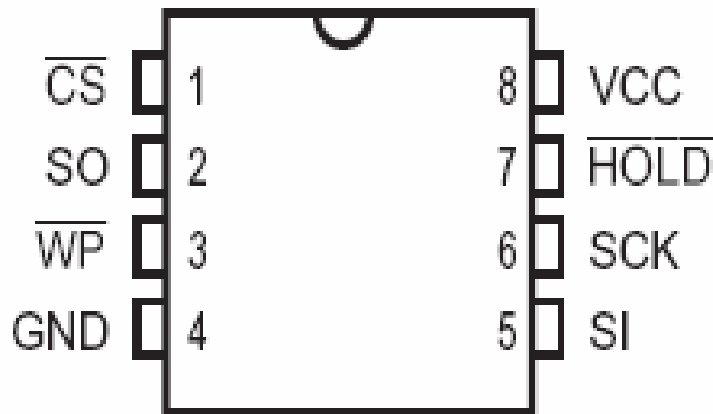
存储器接口-ROM(1)

1. 概述

- ROM (Read only memory)
 - Program type
 - ROM, PROM, EPROM, EEPROM
 - FLASH (NOR, NAND)
 - Data bus
 - Parellel, Serial
 - Interface
 - Parellel (same as SRAM)
 - Serial
 - SPI (Serial port interface)
 - I2C

存储器接口-ROM(2)

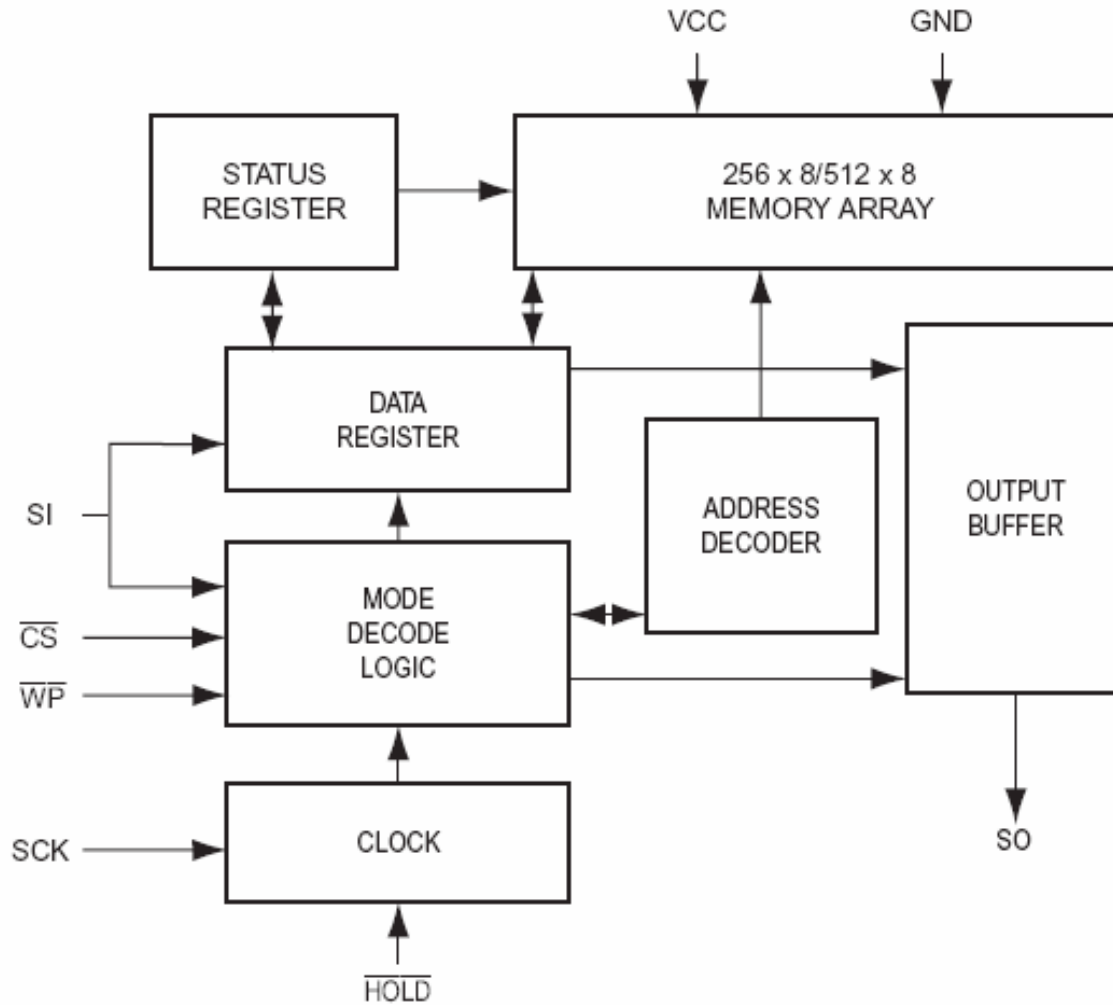
2. EEPROM (25C02, SPI) -Package



\overline{CS}	Chip Select
SCK	Serial Data Clock
SI	Serial Data Input
SO	Serial Data Output
GND	Ground
Vcc	Power
\overline{WP}	Write Protect
\overline{HOLD}	Suspends Serial Input

存储器接口-ROM(3)

2. EEPROM (25C02) -Function blocks



存储器接口-ROM(4)

2. EEPROM (25C02) -Serial port

- **MASTER**: the device that provide clock (OP-CODE)
- **SLAVE**: Input clock (OP-CODE)
- **Transmitter**: S0
- **Receiver**: SI
- **Rate**: the frequency of clock
- **MSB**: the most significant bit, generally the first Transmit or receive bit
- **OP-CODE**: the first byte transmit to slave following /CS transfer to LOW

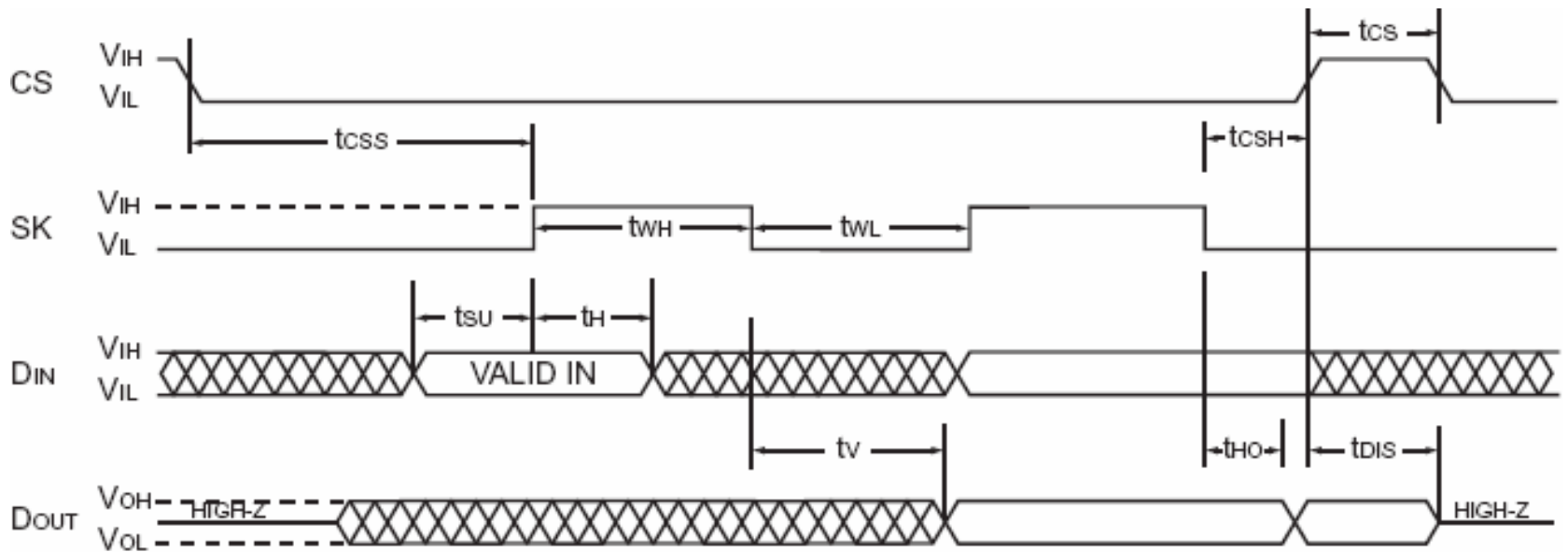
存储器接口-ROM(5)

2. EEPROM (25C02) -Op-Code

Name	Op-code	Operation	Address	Data(SI)	Data (SO)
WREN	0000 X110	Set Write Enable Latch	-	-	-
WRDI	0000 X100	Reset Write Enable Latch	-	-	-
RDSR	0000 X101	Read Status Register	-	-	D7-D0,...
WRSR	0000 X001	Write Status Register	-	D7-D0	-
READ	0000 Ag011	Read Data from Array	A7-A0	-	D7-D0,...
WRITE	0000 Ag010	Write Data to Array	A7-A0	D7-D0,...	-

存储器接口-ROM(6)

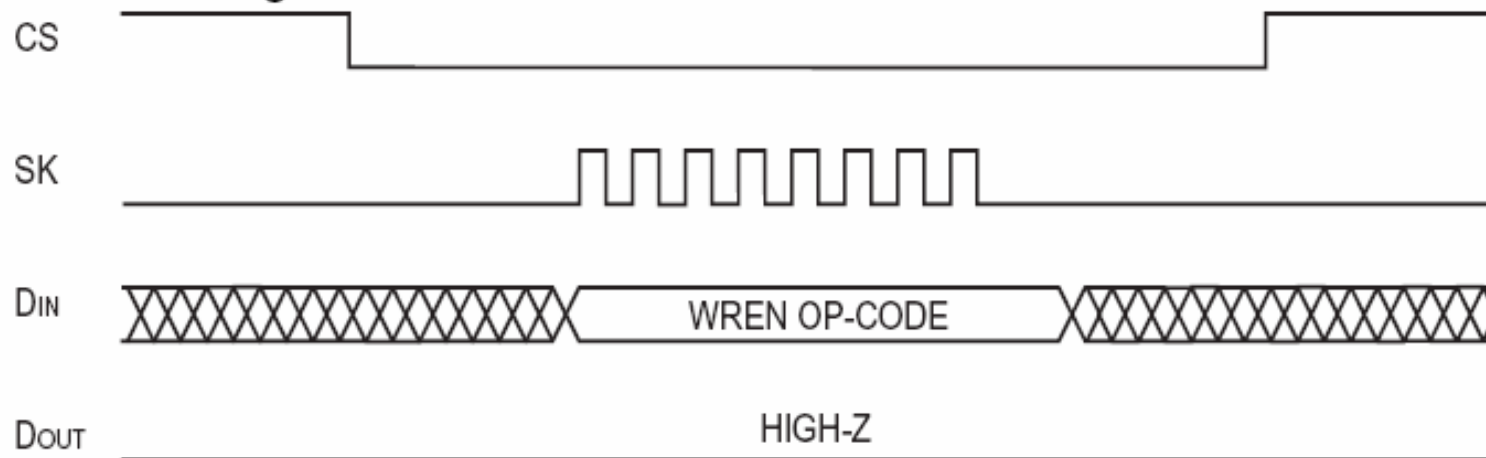
2. EEPROM (25C02) -Synchronous data timing



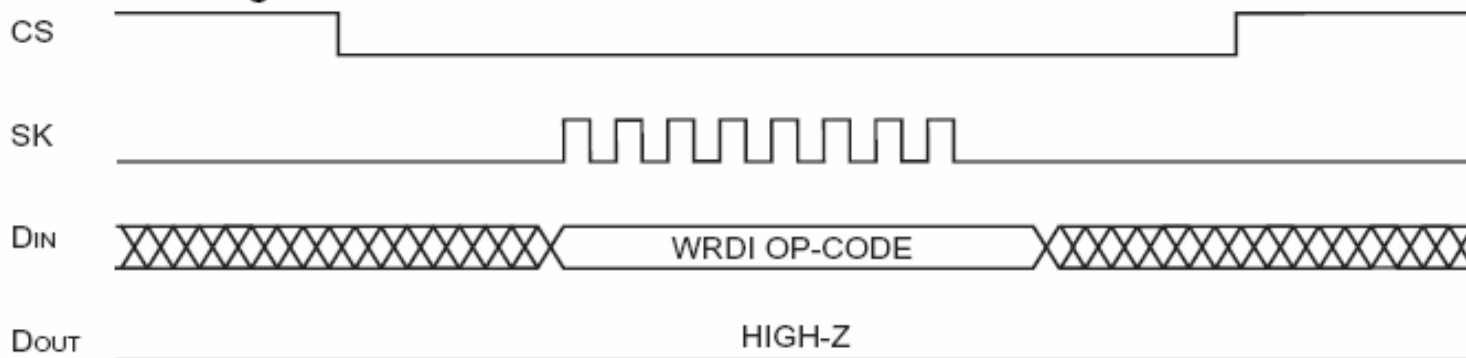
接存储器接口-ROM(7)

2. EEPROM (25C02) -Timing-WREN/WRDI

WREN Timing

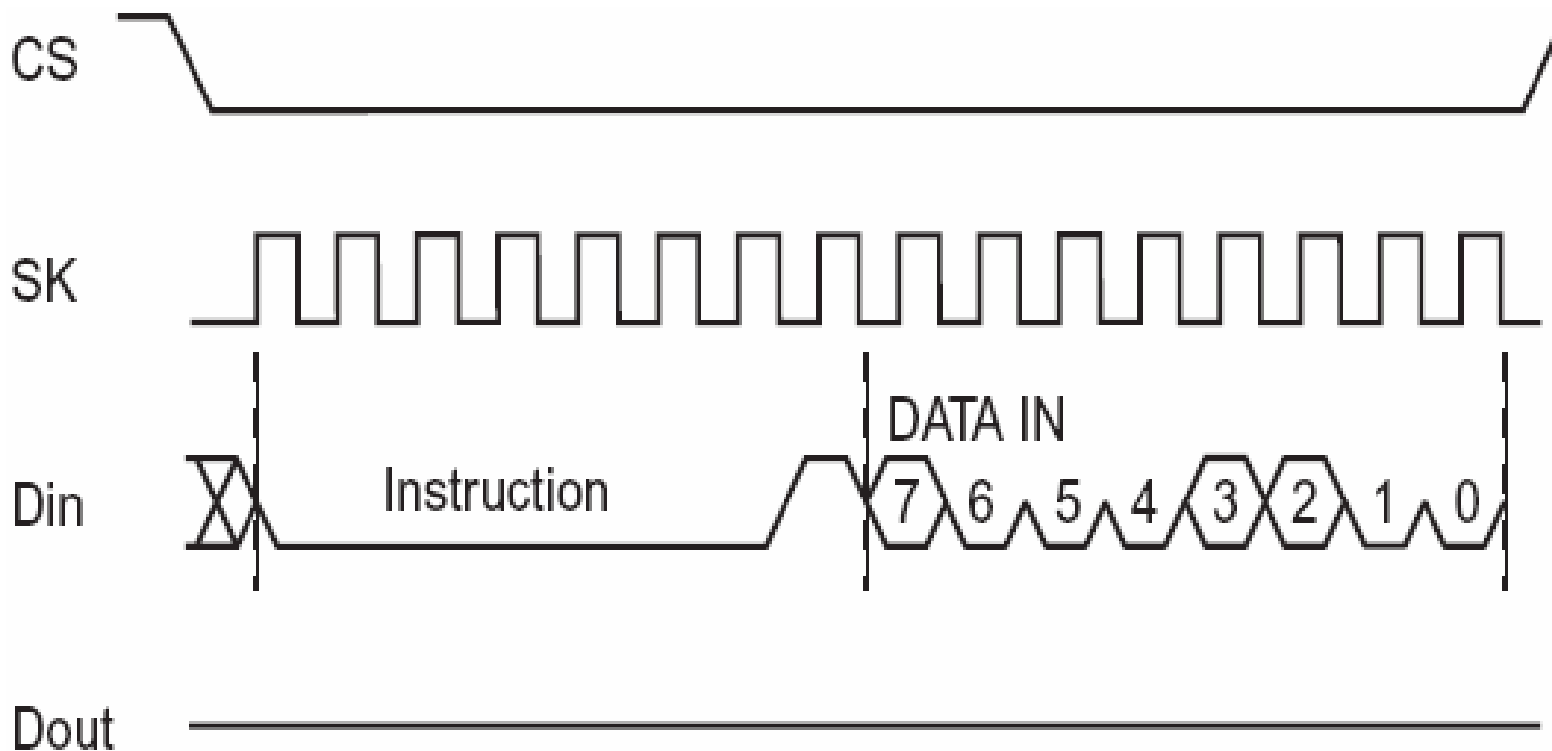


WRDI Timing



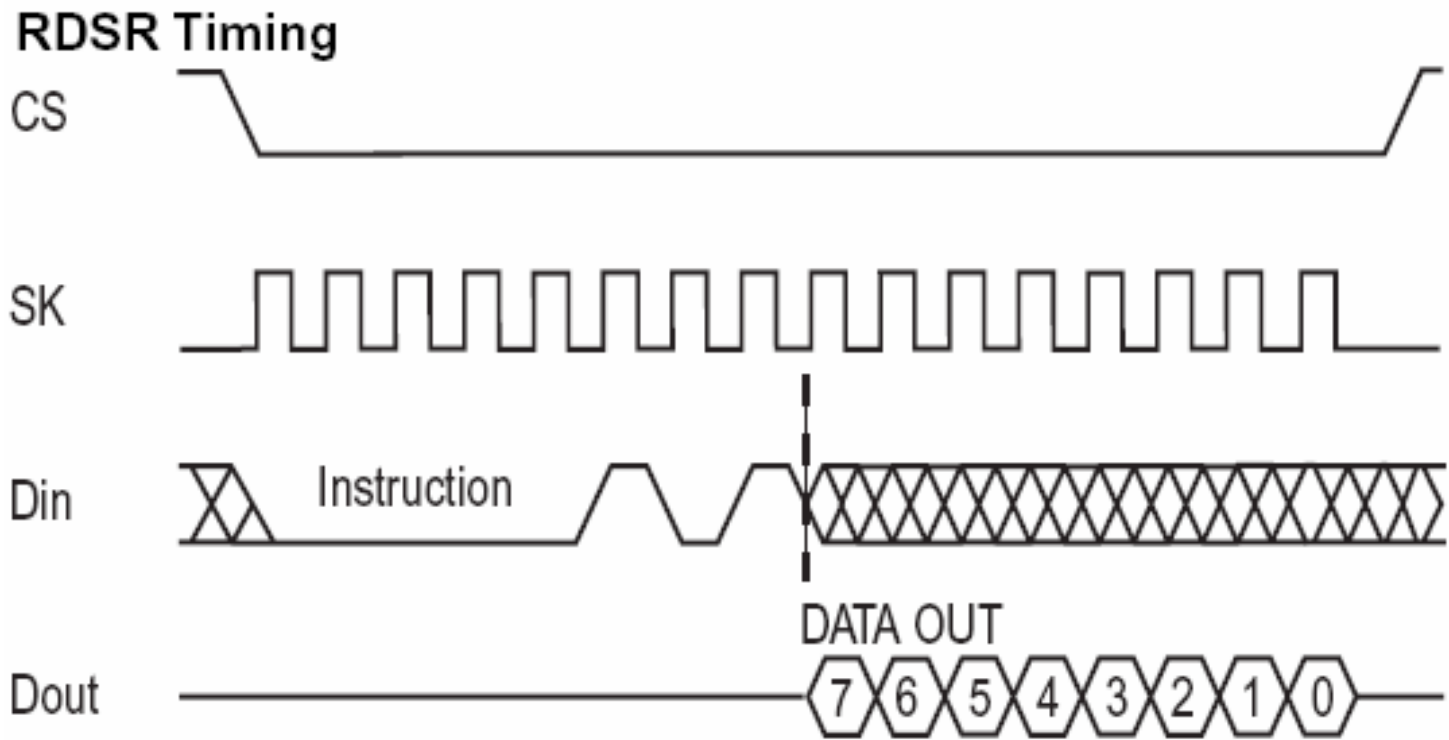
存储器接口-ROM(8)

2. EEPROM (25C02) -Timing-WDSR



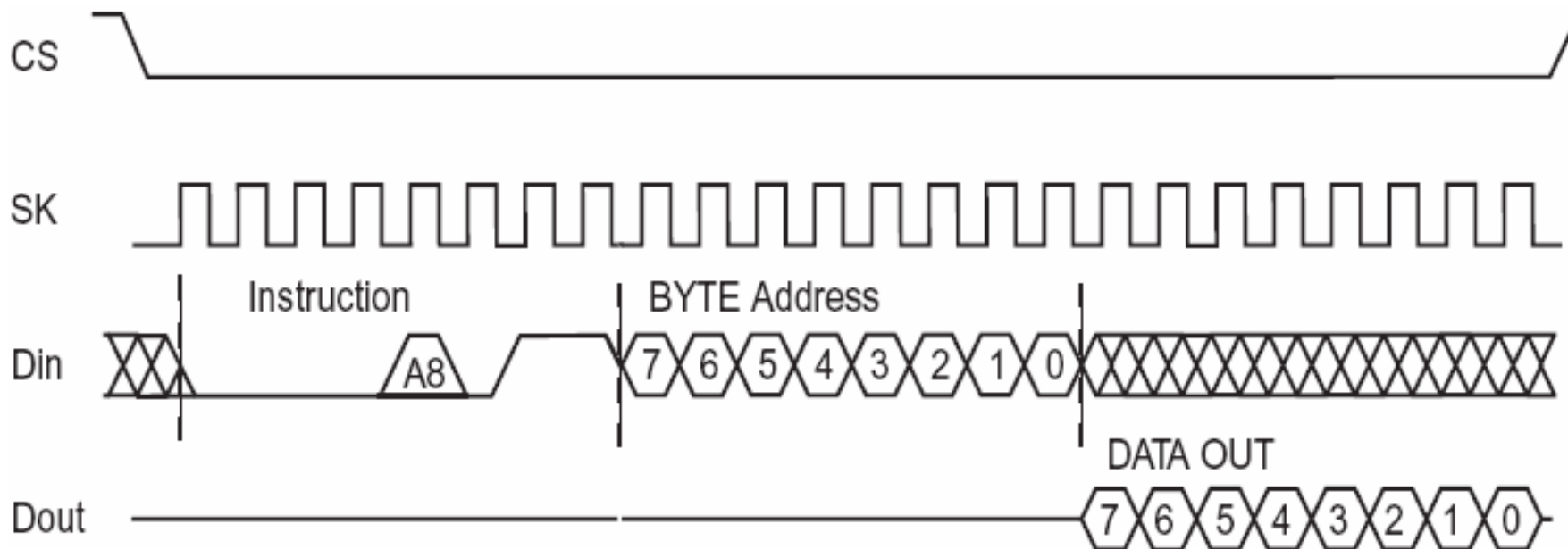
存储器接口-ROM(8)

2. EEPROM (25C02) -Timing-RDSR



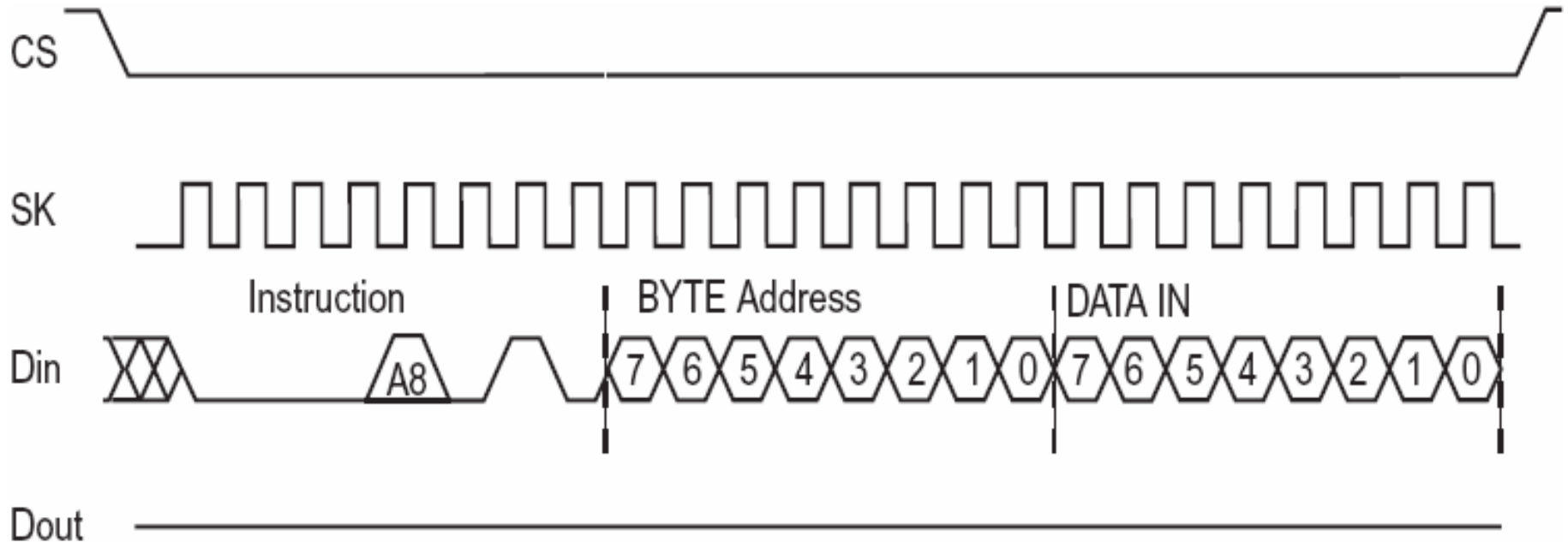
存储器接口-ROM(10)

2. EEPROM (25C02) -Timing-READ



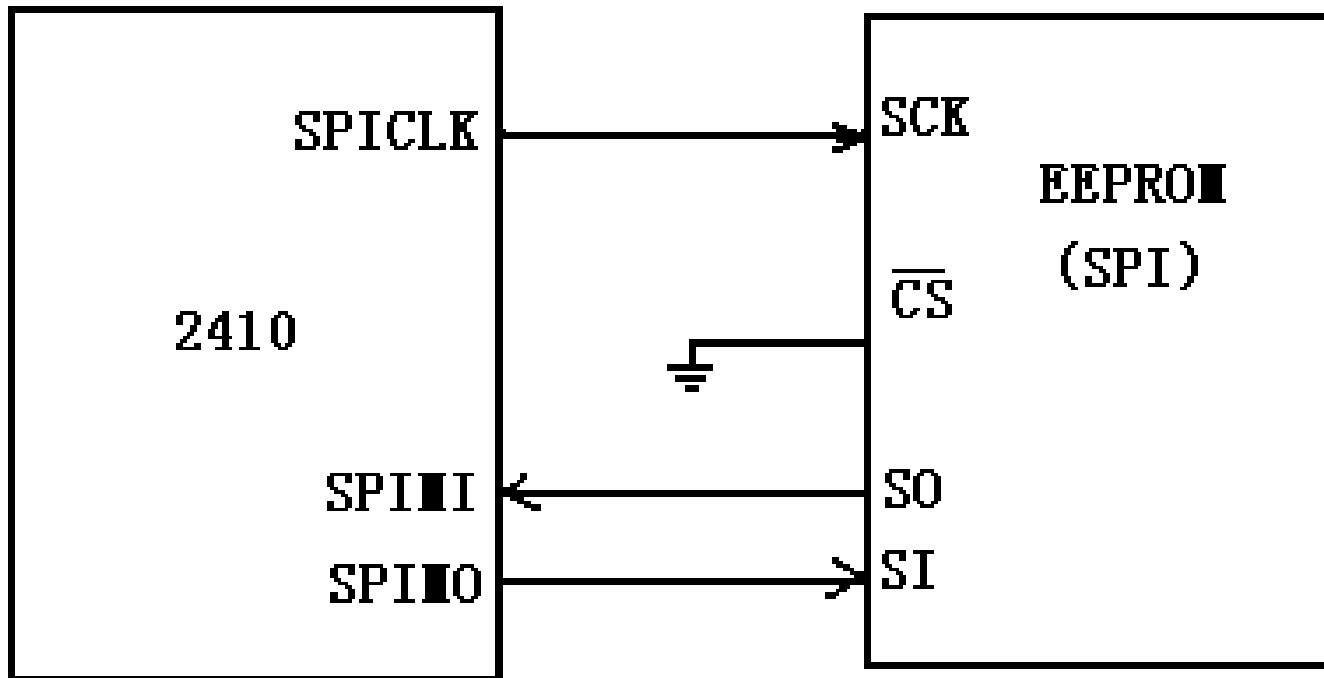
存储器接口-ROM(11)

2. EEPROM (25C02) -Timing-WRITE



存储器接口-ROM(12)

2. EEPROM (25C02) -Connection with DSP



/CS常接有效是否会出现意外读写?

如果处理器不带有**SPI**接口，如何使用**25C02**?

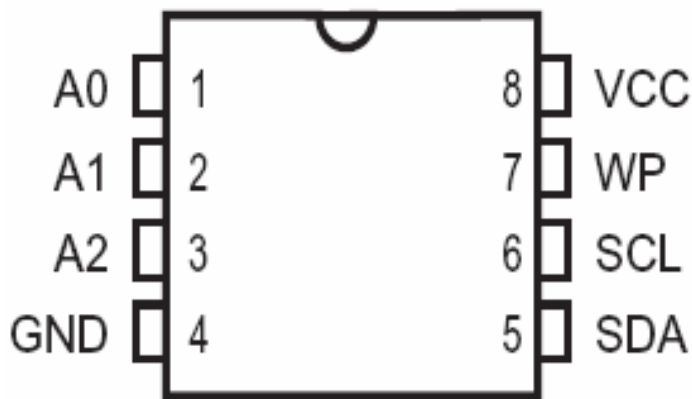
存储器接口-ROM(13)

2. EEPROM (25C02) -Programming

- Initial SPI port of 2410
- Send WREN to EEPROM
- Write data
 - Send command
 - Send address
 - Send data
- Read data
 - Send command
 - Send address
 - Get data

存储器接口-ROM(14)

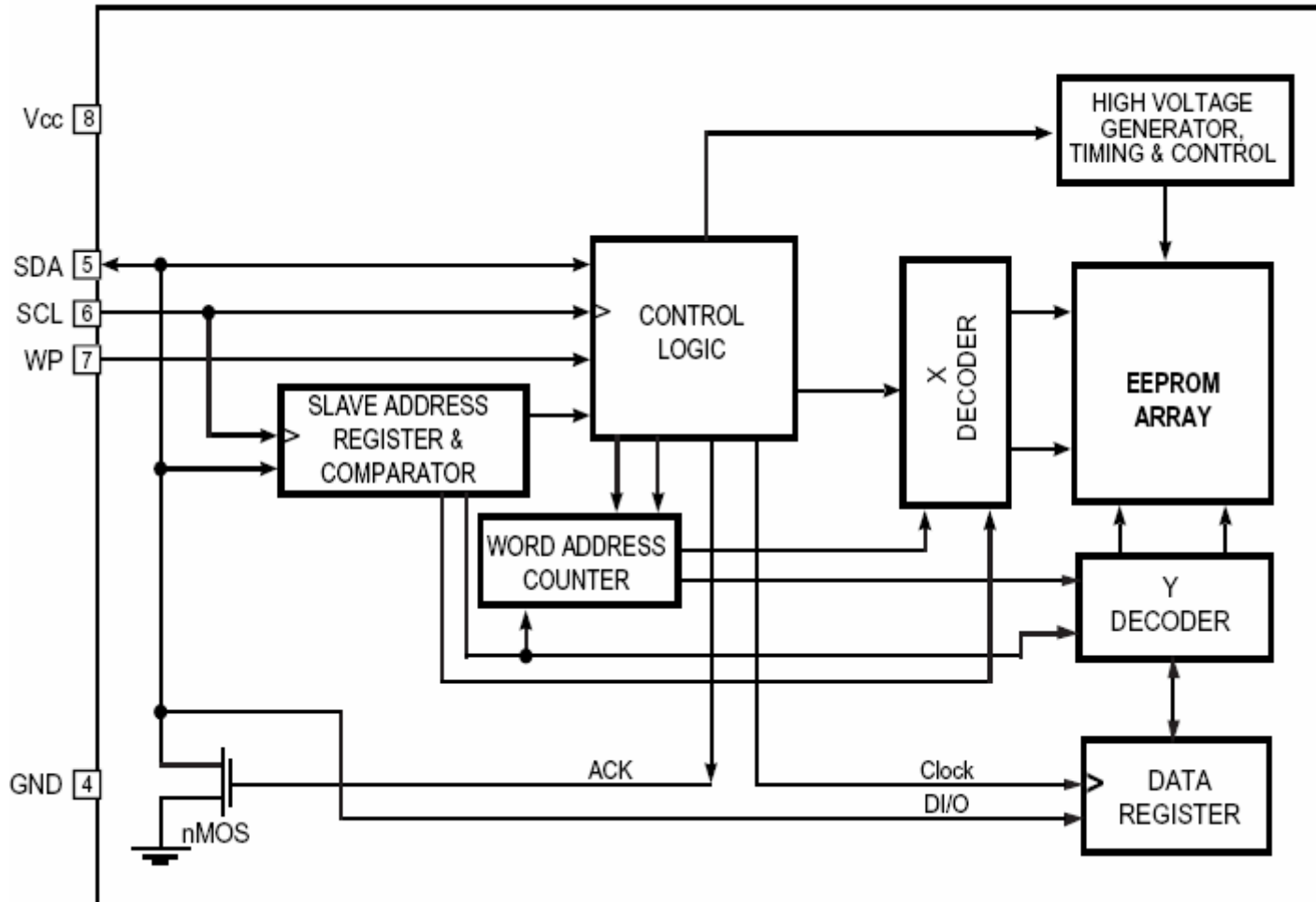
3. EEPROM (24C01, IIC)-Package



A0-A2	Address Inputs
SDA	Serial Address/Data I/O
SCL	Serial Clock Input
WP	Write Protect Input
Vcc	Power Supply
GND	Ground

存储器接口-ROM(15)

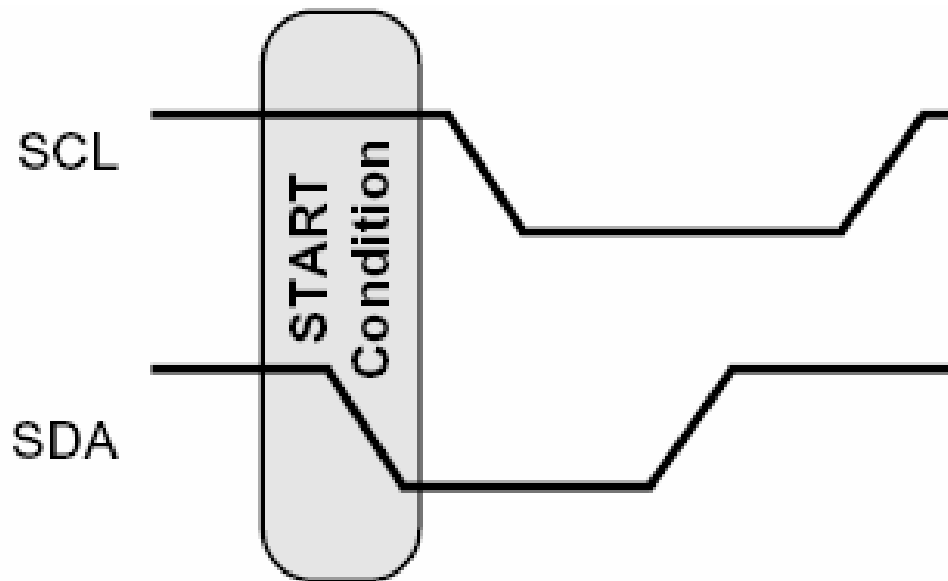
3. EEPROM (24C01)-Function Block



存储器接口-ROM(16)

3. EEPROM (24C01)-Bus protocol

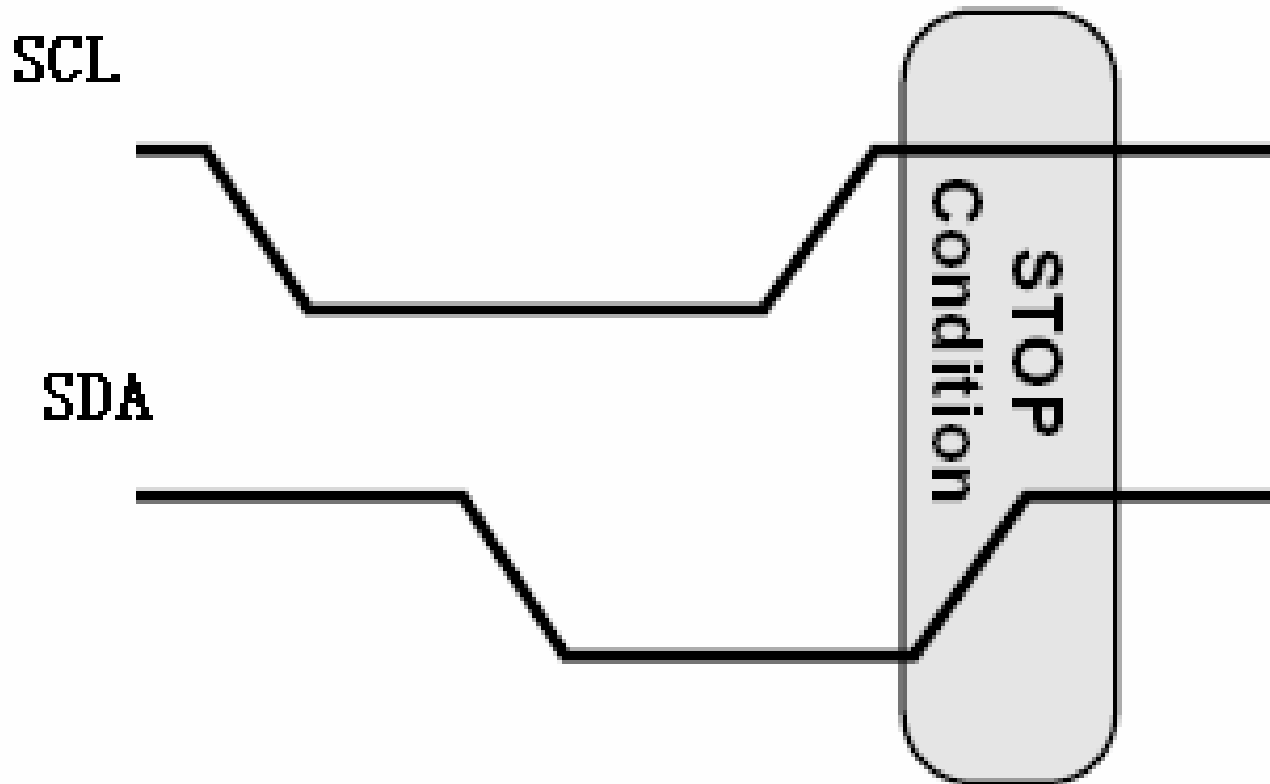
- Start condition



存储器接口-ROM(17)

3. EEPROM (24C01)-Bus protocol

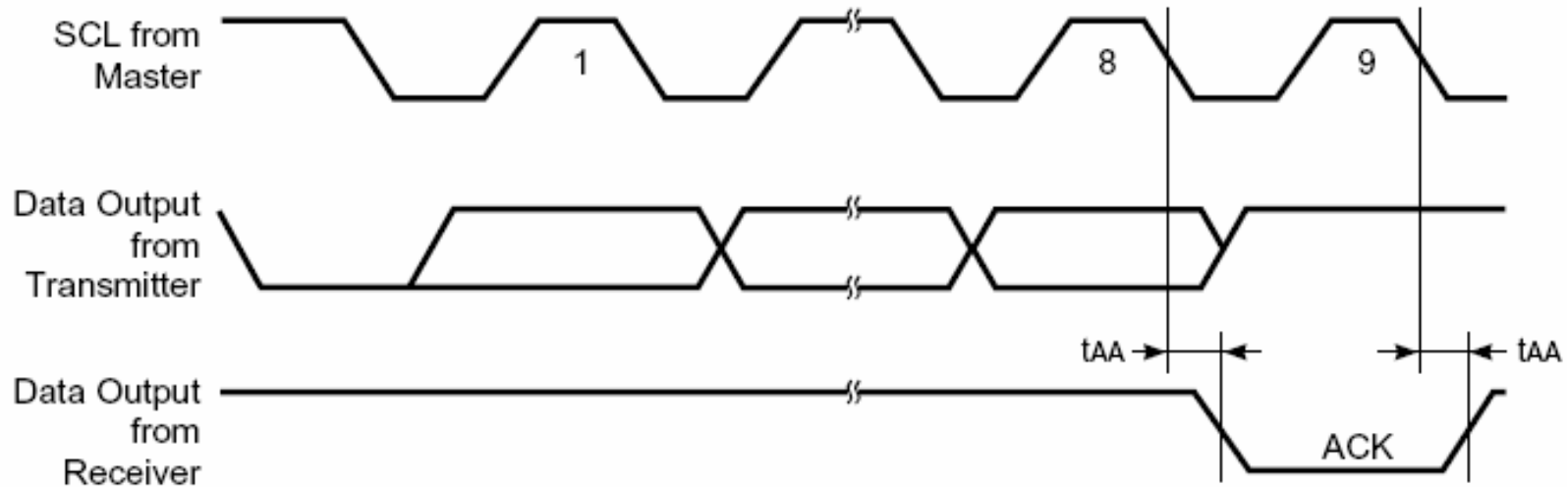
- Stop condition



存储器接口-ROM(18)

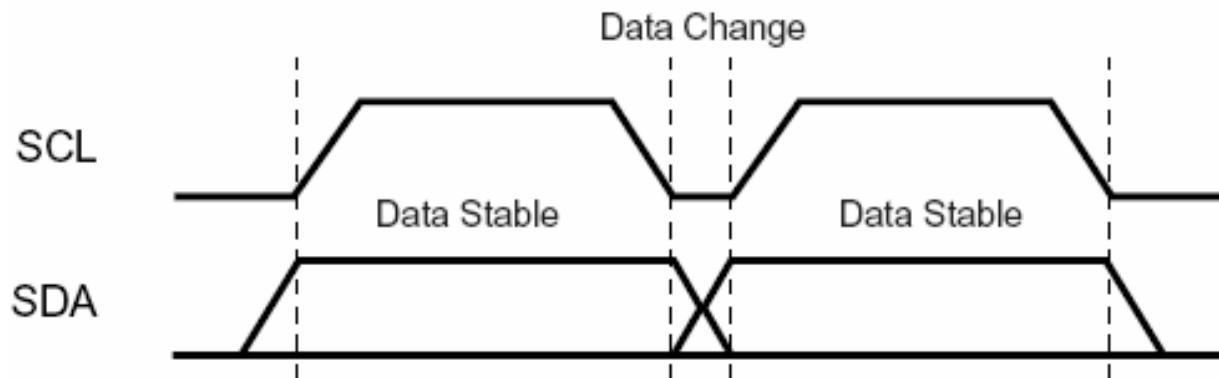
3. EEPROM (24C01)-Bus protocol

- Acknowledge



存储器接口-ROM(19)

3. EEPROM (24C01)-Data valid protocol



存储器接口-ROM(20)

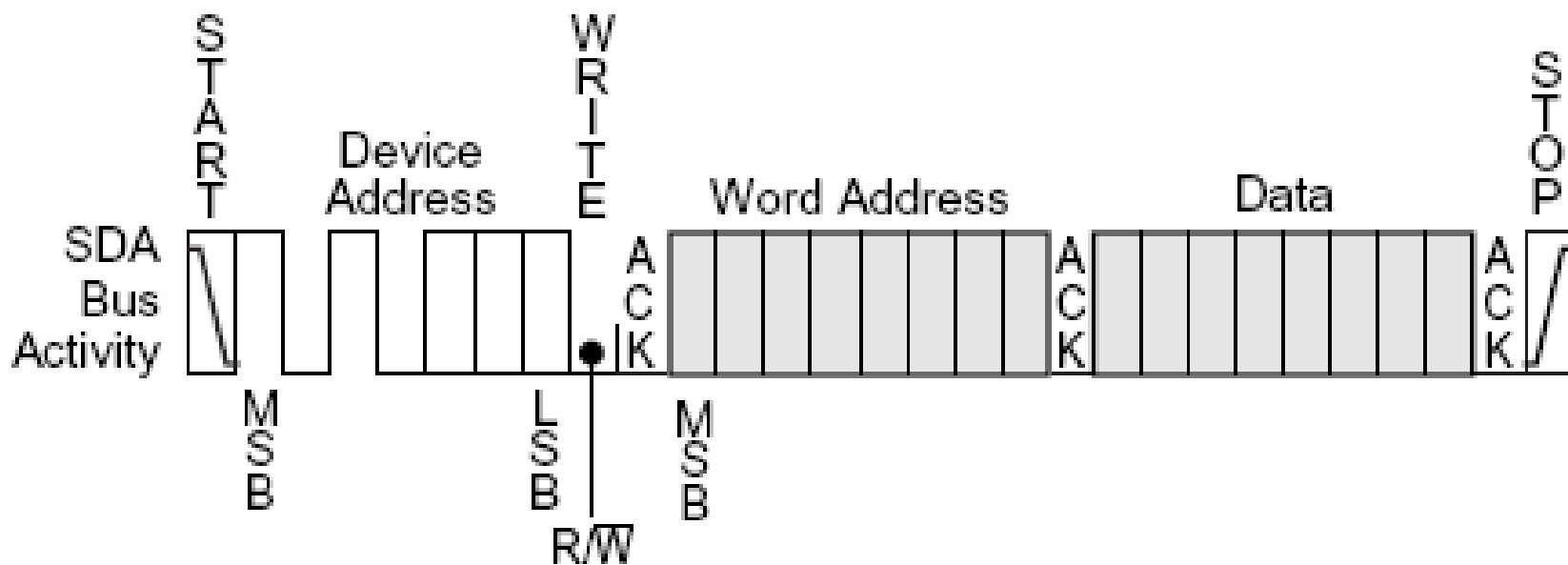
3. EEPROM (24C01)-Device address

BIT	7	6	5	4	3	2	1	0	
	1	0	1	0	A2	A1	A0	$\overline{R/W}$	IS24C01 IS24C02

存储器接口-ROM(21)

3. EEPROM (24C01)-Format of data R-W

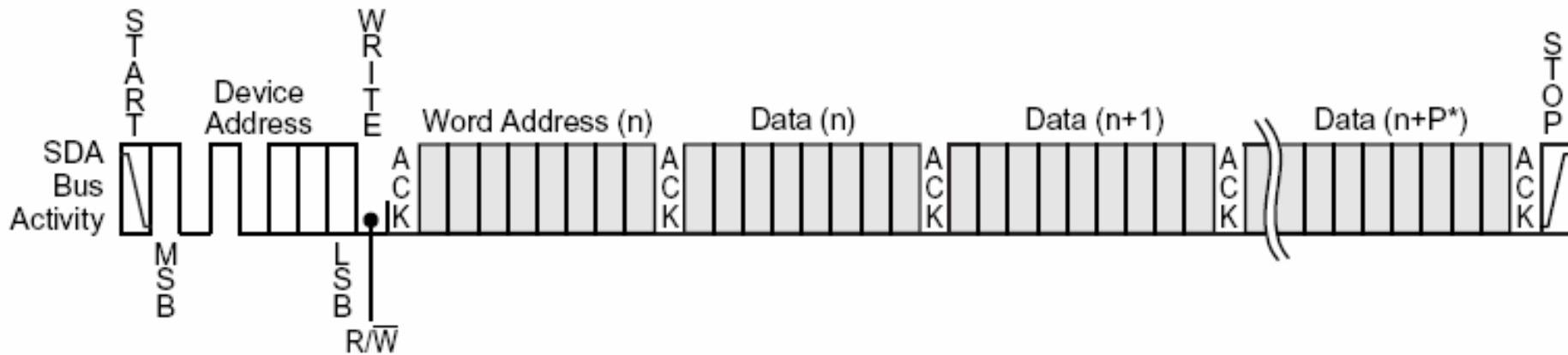
- Byte write



存储器接口-ROM(22)

3. EEPROM (24C01)-Format of data R-W

- Page write



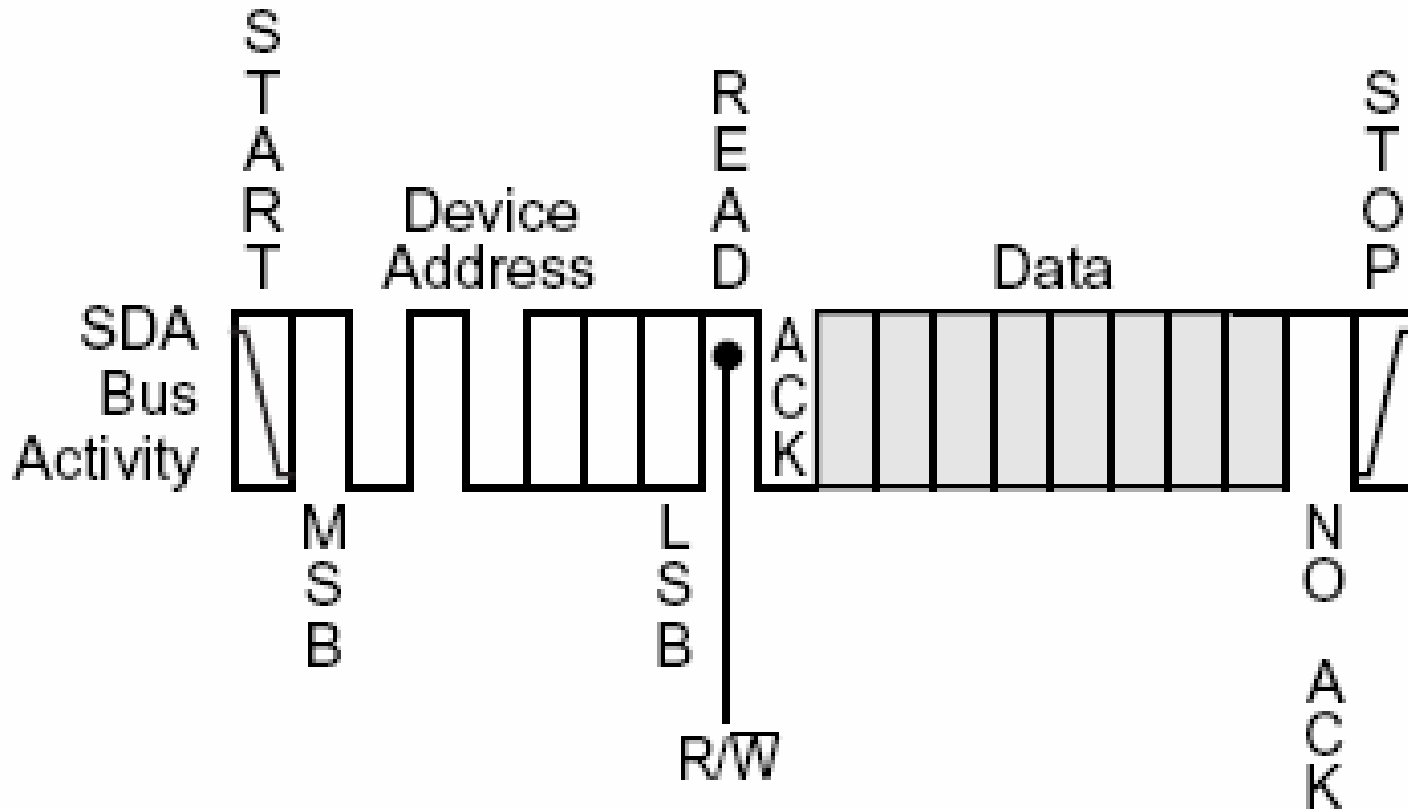
P =7: 24C01/02

=15: 24C04/08

存储器接口-ROM(23)

3. EEPROM (24C01)-Format of data R-W

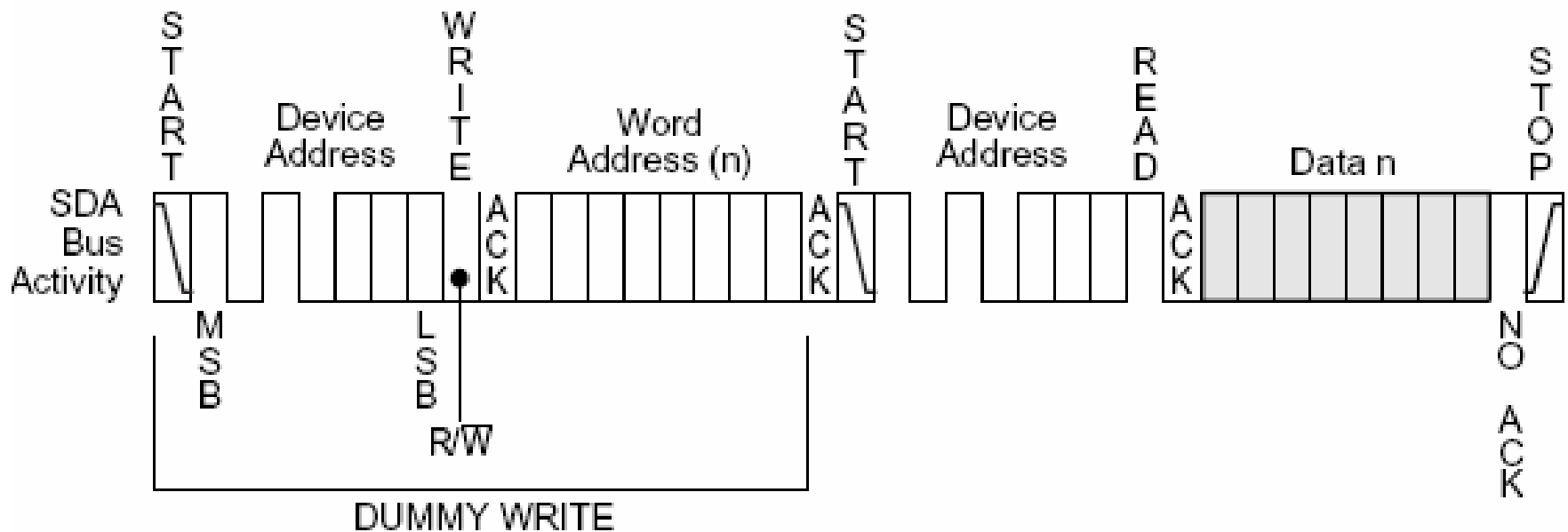
- Current address read



存储器接口-ROM(24)

3. EEPROM (24C01)-Format of data R-W

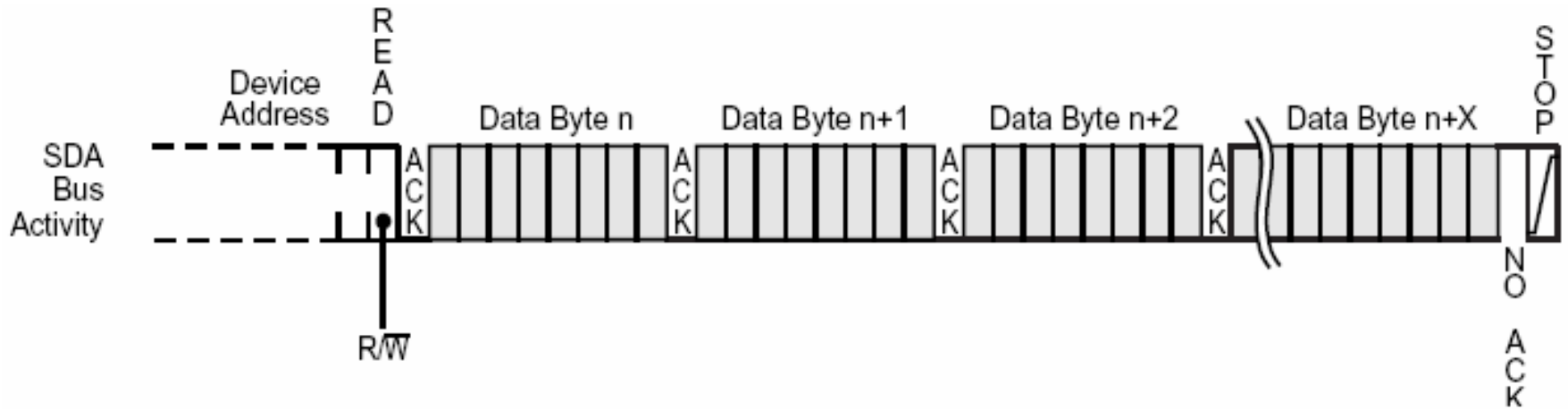
- Random address read



存储器接口-ROM(25)

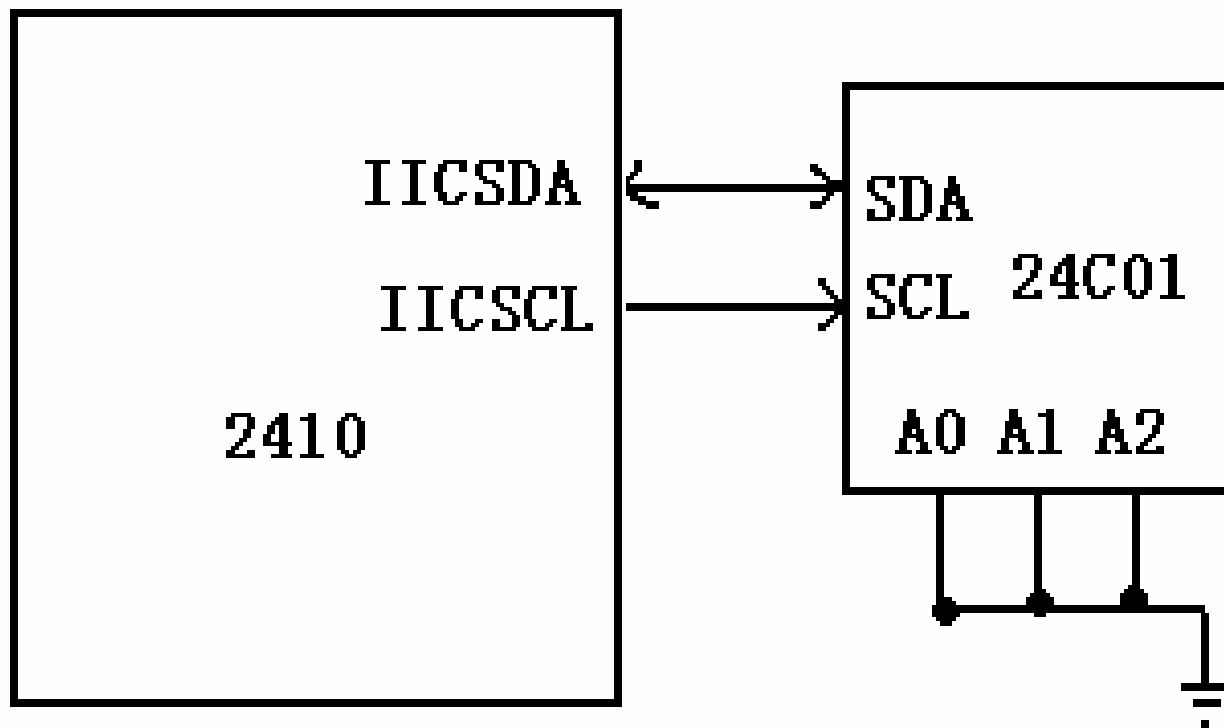
3. EEPROM (24C01)-Format of data R-W

- Sequential read



存储器接口-ROM(26)

3. EEPROM (24C01)-Connection



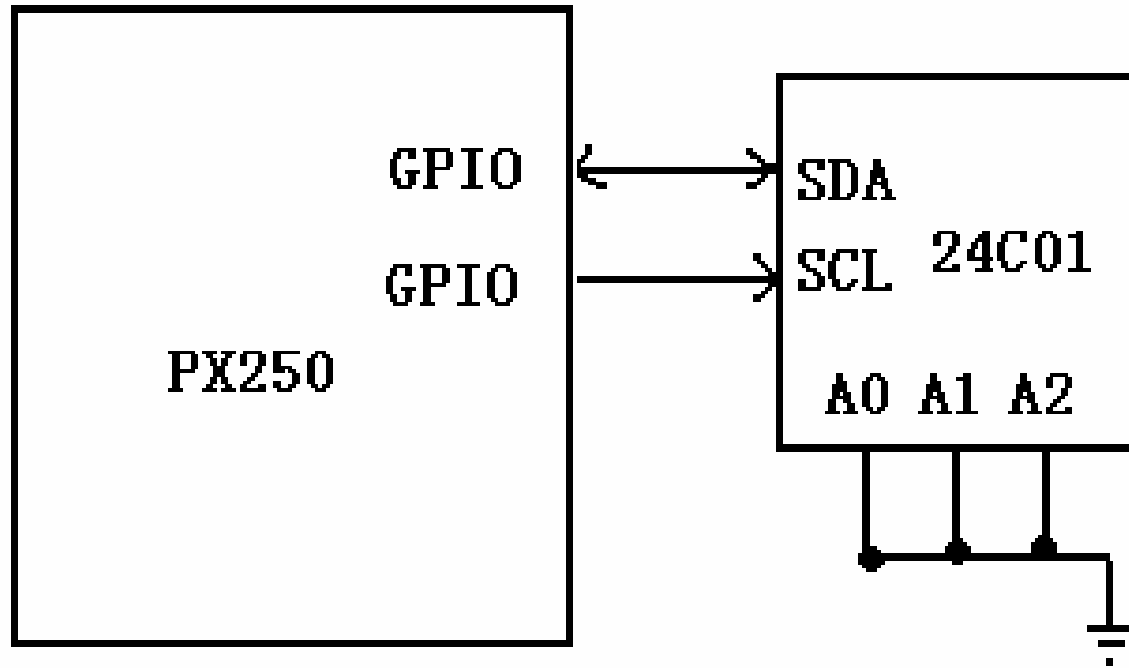
IIC器件有无主从区别？

对于没有**IIC**接口的**CPU**，如何连接**IIC**设备？

存储器接口-ROM(27)

3. EEPROM (24C01)-IIC Master

- Connection



存储器接口-ROM(28)

3. EEPROM (24C01)-IIC Master

- Programming
 - Control interface
 - Initial Gpio Port
 - generate SCLK and SDA
 - Generate Start, stop, and acknowledge condition
 - read or write a bit
 - Read or write a byte by bit operation

存储器接口-ROM(29)

3. EEPROM (24C01)-IIC Master

- Programming sample

```
int i2cstart();
int i2cstop();
int i2cwrite(int cc);
int i2cstart()
{
    i2c_con=0x7;        //scl=1,sdacon=1,sda=1
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    i2c_con=0x3;        //scl=1,sda=0
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    i2c_con=0x2;        //scl=0;sda=0
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    return 0;}
```

存储器接口-ROM(30)

3. EEPROM (24C01)-IIC Master

- Programming sample (续)

```
int i2cstop()
{
    i2c_con=0x2;      //scl=0,sdacon=1,sda=0
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    i2c_con=0x3;      //scl=1,sda=0
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    i2c_con=0x7;      //scl=1;sda=1
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    return 0;
}
```

存储器接口-ROM(31)

3. EEPROM (24C01)- IIC Master

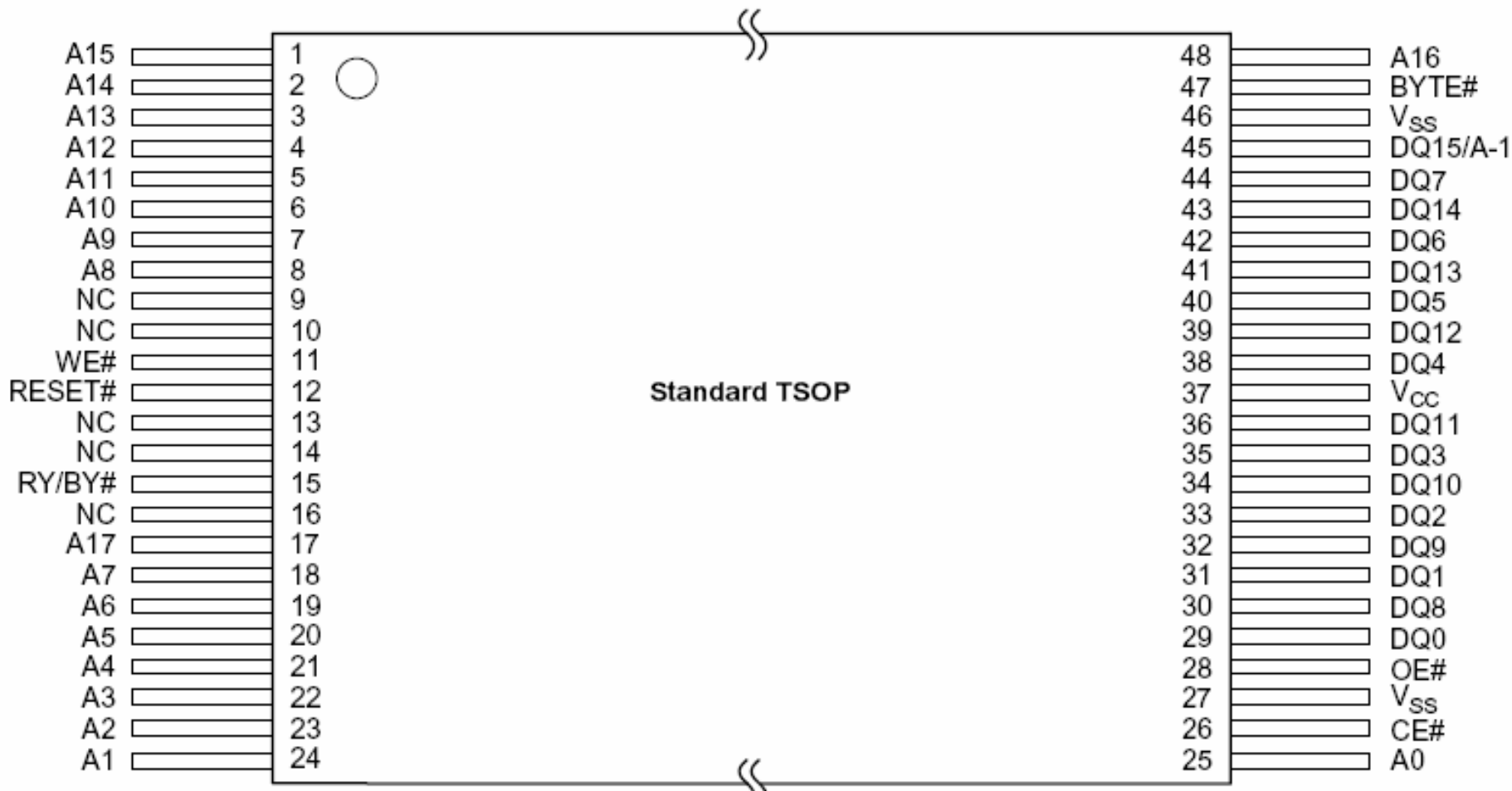
- Programming sample (续)

```
int i2cwrite(int cc)
{ int i;
  char j;      //sda=1;scl=1
  for(i=0;i<8;i++)
  {
    updatewatchdog();
    i2c_con=i2c_con&0xfe|2;    //scl=0,sdacon=1,sda=x
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    j=(cc>>(7-i))&1;
    j=(j<<2)&4;
    i2c_con=i2c_con&0xfb|j;    //scl=0,sdacon=1,sda=j
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
    i2c_con=i2c_con|1; //scl=1,sda=j
    *(I2C_CON)=i2c_con;mdelay(i2ctimes);
  }

  . . . ; . . .
return i;
}
```

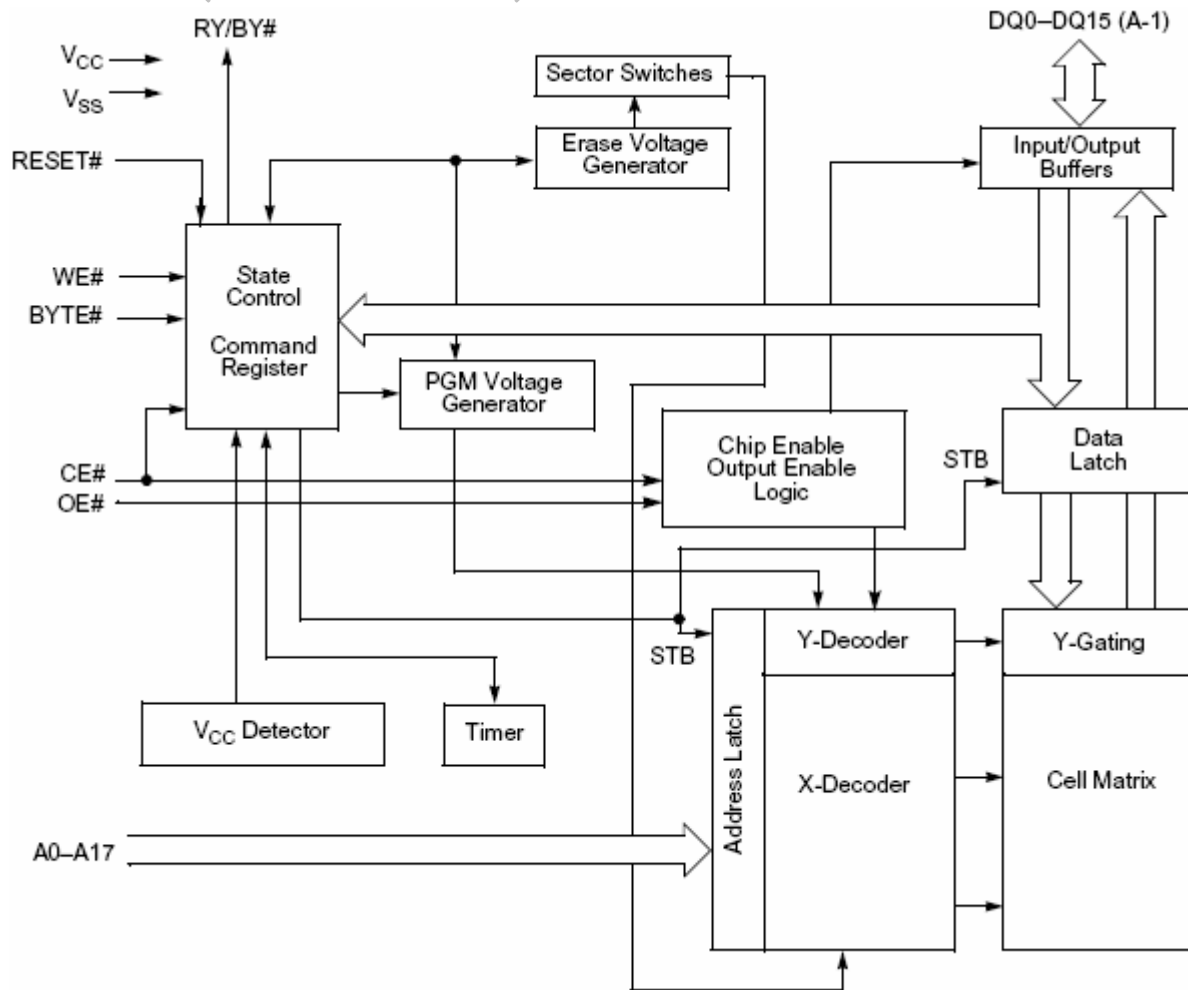
存储器接口-ROM(32)

4. Flash (29LV400B, NOR) -Package



存储器接口-ROM(33)

4. Flash (29LV400B) -Function Block



存储器接口-ROM(34)

4. Flash (29LV400B)-True Table

Operation	CE#	OE#	WE#	RESET#	Addresses (Note 1)	DQ0- DQ7	DQ8-DQ15	
							BYTE# = V _{IH}	BYTE# = V _{IL}
Read	L	L	H	H	A _{IN}	D _{OUT}	D _{OUT}	DQ8-DQ14 = High-Z, DQ15 = A-1
Write	L	H	L	H	A _{IN}	D _{IN}	D _{IN}	
Standby	V _{CC} ± 0.3 V	X	X	V _{CC} ± 0.3 V	X	High-Z	High-Z	High-Z
Output Disable	L	H	H	H	X	High-Z	High-Z	High-Z
Reset	X	X	X	L	X	High-Z	High-Z	High-Z
Sector Protect (Note 2)	L	H	L	V _{ID}	Sector Address, A6 = L, A1 = H, A0 = L	D _{IN}	X	X
Sector Unprotect (Note 2)	L	H	L	V _{ID}	Sector Address, A6 = H, A1 = H, A0 = L	D _{IN}	X	X
Temporary Sector Unprotect	X	X	X	V _{ID}	A _{IN}	D _{IN}	D _{IN}	High-Z

存储器接口-ROM(35)

4. Flash (29LV400B)–Sectors

Am29LV400BB Bottom Boot Sector Address Table

Sector	A17	A16	A15	A14	A13	A12	Sector Size (Kbytes/ Kwords)	Address Range (in hexadecimal)	
								(x8) Address Range	(x16) Address Range
SA0	0	0	0	0	0	X	16/8	00000h–03FFFh	00000h–01FFFh
SA1	0	0	0	0	1	0	8/4	04000h–05FFFh	02000h–02FFFh
SA2	0	0	0	0	1	1	8/4	06000h–07FFFh	03000h–03FFFh
SA3	0	0	0	1	X	X	32/16	08000h–0FFFFh	04000h–07FFFh
SA4	0	0	1	X	X	X	64/32	10000h–1FFFFh	08000h–0FFFFh
SA5	0	1	0	X	X	X	64/32	20000h–2FFFFh	10000h–17FFFh
SA6	0	1	1	X	X	X	64/32	30000h–3FFFFh	18000h–1FFFFh
SA7	1	0	0	X	X	X	64/32	40000h–4FFFFh	20000h–27FFFh
SA8	1	0	1	X	X	X	64/32	50000h–5FFFFh	28000h–2FFFFh
SA9	1	1	0	X	X	X	64/32	60000h–6FFFFh	30000h–37FFFh
SA10	1	1	1	X	X	X	64/32	70000h–7FFFFh	38000h–3FFFFh

存储器接口-ROM(36)

4. Flash (29LV400B)-Device code

Am29LV400B Autoselect Codes (High Voltage Method)

Description	Mode	CE#	OE#	WE#	A17 to A12	A11 to A10	A9	A8 to A7	A6	A5 to A2	A1	A0	DQ8 to DQ15	DQ7 to DQ0
Manufacturer ID: AMD		L	L	H	X	X	V _{ID}	X	L	X	L	L	X	01h
Device ID: Am29LV400B (Top Boot Block)	Word	L	L	H	X	X	V _{ID}	X	L	X	L	H	22h	B9h
	Byte	L	L	H									X	B9h
Device ID: Am29LV400B (Bottom Boot Block)	Word	L	L	H	X	X	V _{ID}	X	L	X	L	H	22h	BAh
	Byte	L	L	H									X	BAh
Sector Protection Verification		L	L	H	SA	X	V _{ID}	X	L	X	H	L	X	01h (protected)
													X	00h (unprotected)

L = Logic Low = V_{IL}, H = Logic High = V_{IH}, SA = Sector Address, X = Don't care.

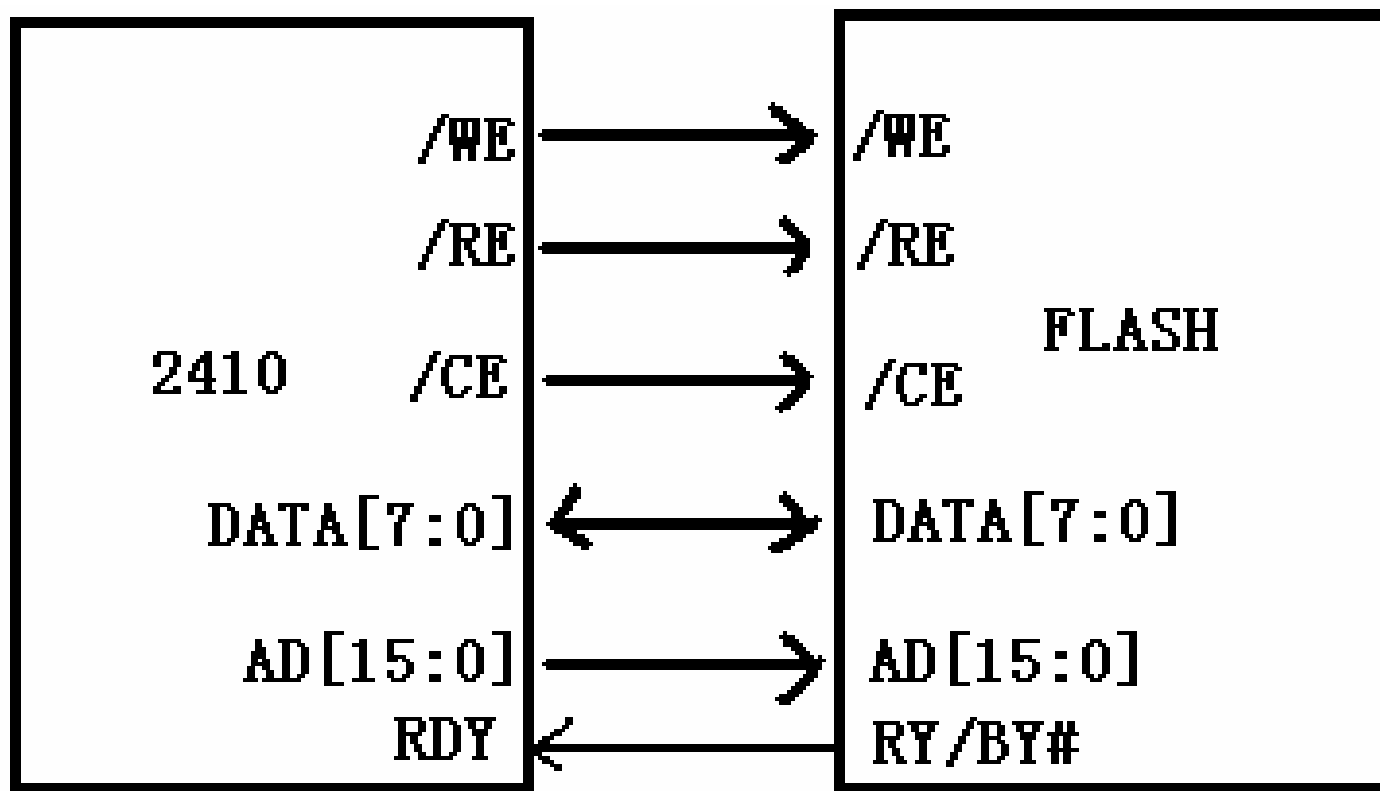
存储器接口-ROM(37)

4. Flash (29LV400B)-command

Command Sequence (Note 1)		Cycles	Bus Cycles (Notes 2-5)													
			First		Second		Third		Fourth		Fifth		Sixth			
			Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data		
Read (Note 6)		1	RA	RD												
Reset (Note 7)		1	XXX	F0												
Autoselect (Note 8)	Manufacturer ID	Word	4	555	AA	2AA	55	555	90	X00	01					
		Byte		AAA		555		AAA								
	Device ID, Top Boot Block	Word	4	555	AA	2AA	55	555	90	X01	22B9					
		Byte		AAA		555		AAA		X02	B9					
	Device ID, Bottom Boot Block	Word	4	555	AA	2AA	55	555	90	X01	22BA					
		Byte		AAA		555		AAA		X02	BA					
	Sector Protect Verify (Note 9)	Word	4	555	AA	2AA	55	555	90	(SA) X02	XX00 XX01					
		Byte		AAA		555		AAA		(SA) X04	00 01					
Program	Word	4	555	AA	2AA	55	555	A0	PA	PD						
	Byte		AAA		555		AAA									
Unlock Bypass	Word	3	555	AA	2AA	55	555	20								
	Byte		AAA		555		AAA									
Unlock Bypass Program (Note 10)		2	XXX	A0	PA	PD										
Unlock Bypass Reset (Note 11)		2	XXX	90	XXX	00										
Chip Erase	Word	6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10		
	Byte		AAA		555		AAA		AAA		555		AAA			
Sector Erase	Word	6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30		
	Byte		AAA		555		AAA		AAA		555					
Erase Suspend (Note 12)		1	XXX	B0												
Erase Resume (Note 13)		1	XXX	30												

存储器接口-ROM(38)

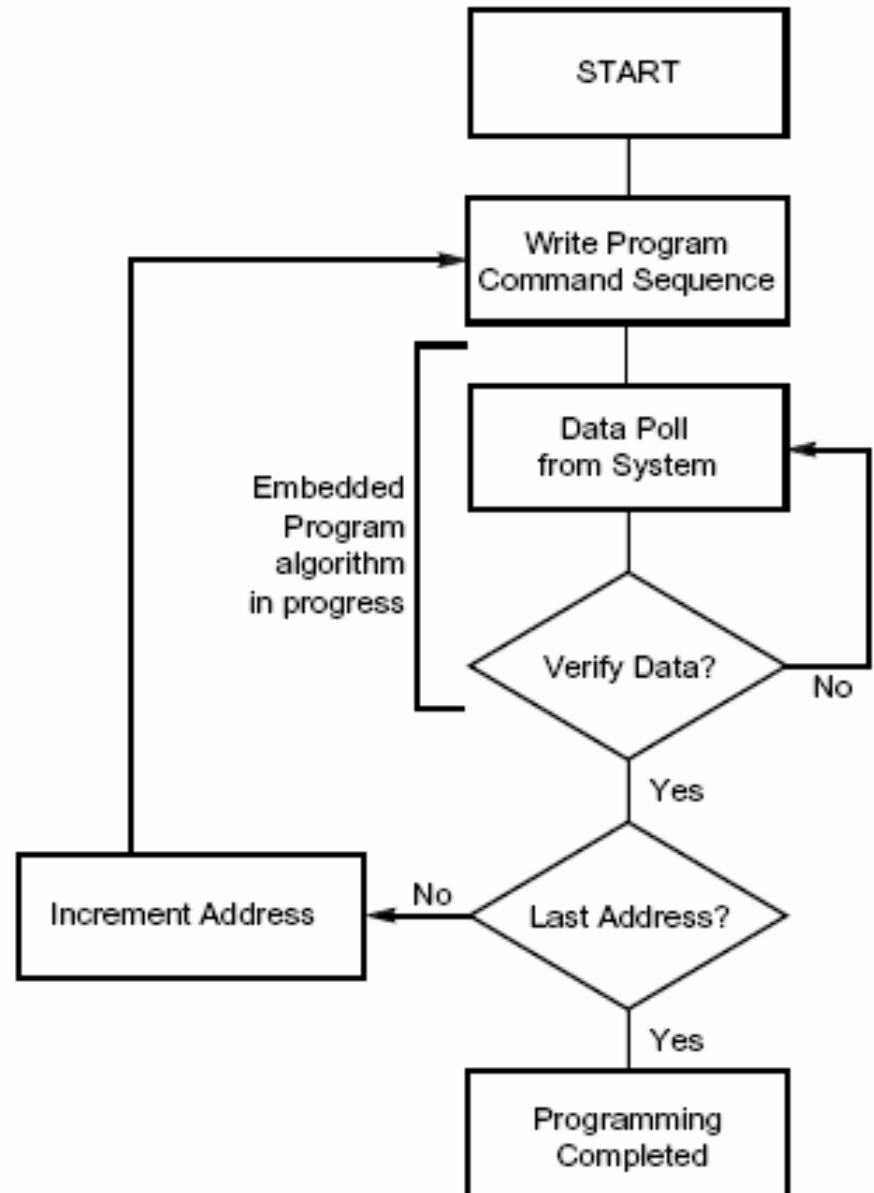
4. Flash (29LV400B)-Connection



存储器接口-ROM(39)

4. Flash (29LV400B)

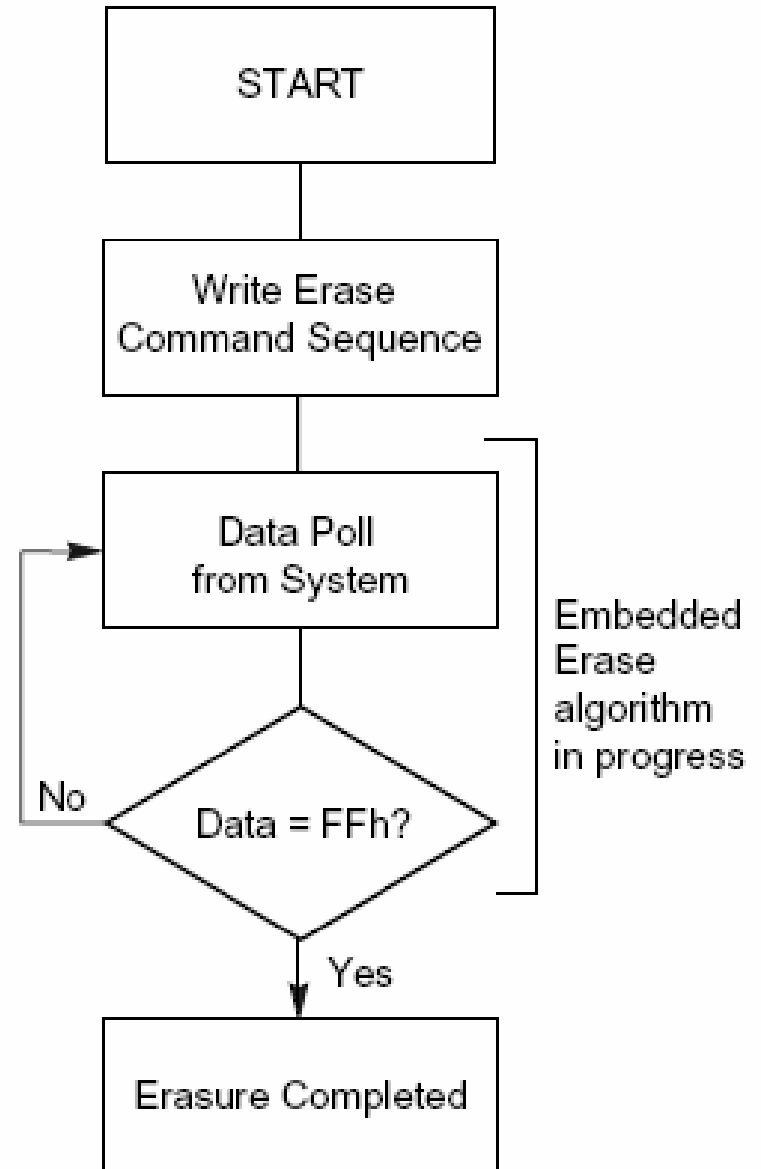
- Programming
 - Read
 - Same as SRAM
 - Programming



存储器接口-ROM(40)

4. Flash (29LV400B)

- Programming (cont.)
 - Erase



存储器接口-ROM(41)

4. Flash (29LV400B)-Erase

```
int erase_sector(int add) //写入命令
{
    unsigned char *p;
    unsigned char data=0;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x0aa;
    p=(unsigned char *)FLASH_BASE+0x555;
    *p=0x55;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x080;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x0aa;
    p=(unsigned char *)FLASH_BASE+0x0555;
    *p=0x055;
    p=(unsigned char *)FLASH_BASE+add;//sector 地址
    *p=0x30; //擦除命令
```

存储器接口-ROM(42)

4. Flash (29LV400B)-Erase

Do //检测状态

{

data = *p;

}while((data&0x20) == 0); //内部算法否结束?

if(*p != 0xff);

return FAIL;

return TRUE;

}

存储器接口-ROM(43)

4. Flash (29LV400B)-Programming

```
int programm(int add, unsigned char data)
{
    unsigned char *p;
    unsigned char data1;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x0aa;
    p=(unsigned char *)FLASH_BASE+0x555;
    *p=0x55;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x0a0;
    p=(unsigned char *)FLASH_BASE+add;
    *p=data; //数据
    do
    { data1=*p;
      if((data1^data)==0) return TRUE;
    }while((data1&0x20)==0);
    return FAIL;
}
```

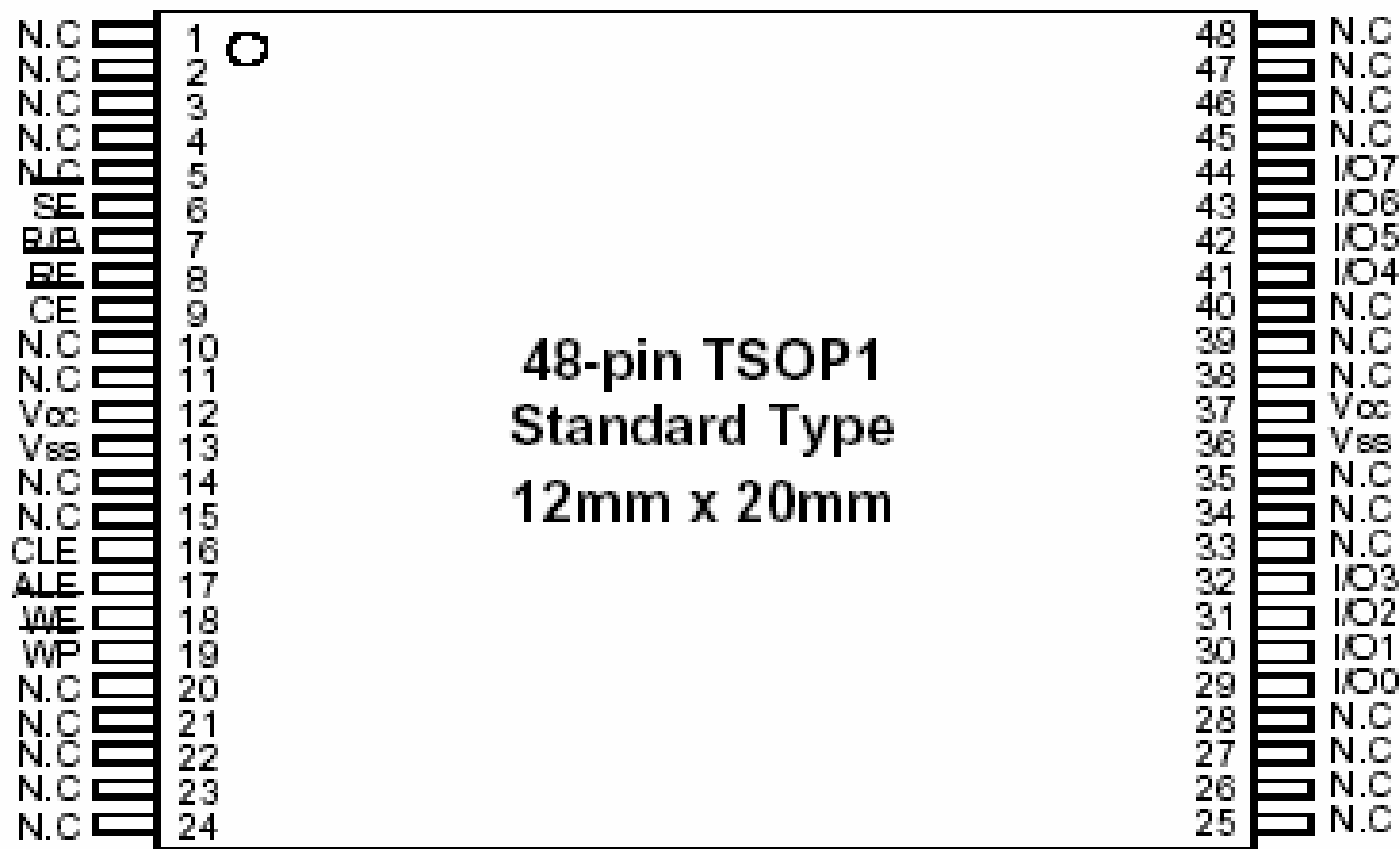
存储器接口-ROM(45)

4. Flash (29LV400B)-Bypass

```
int unlock_bypass(void)
{
    unsigned char *p;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x0aa;
    p=(unsigned char *)FLASH_BASE+0x555;
    *p=0x55;
    p=(unsigned char *)FLASH_BASE+0x0aaa;
    *p=0x20;
    return 1;
}
```

存储器接口-ROM(46)

5. Flash (K9F2808, NAND) –Package



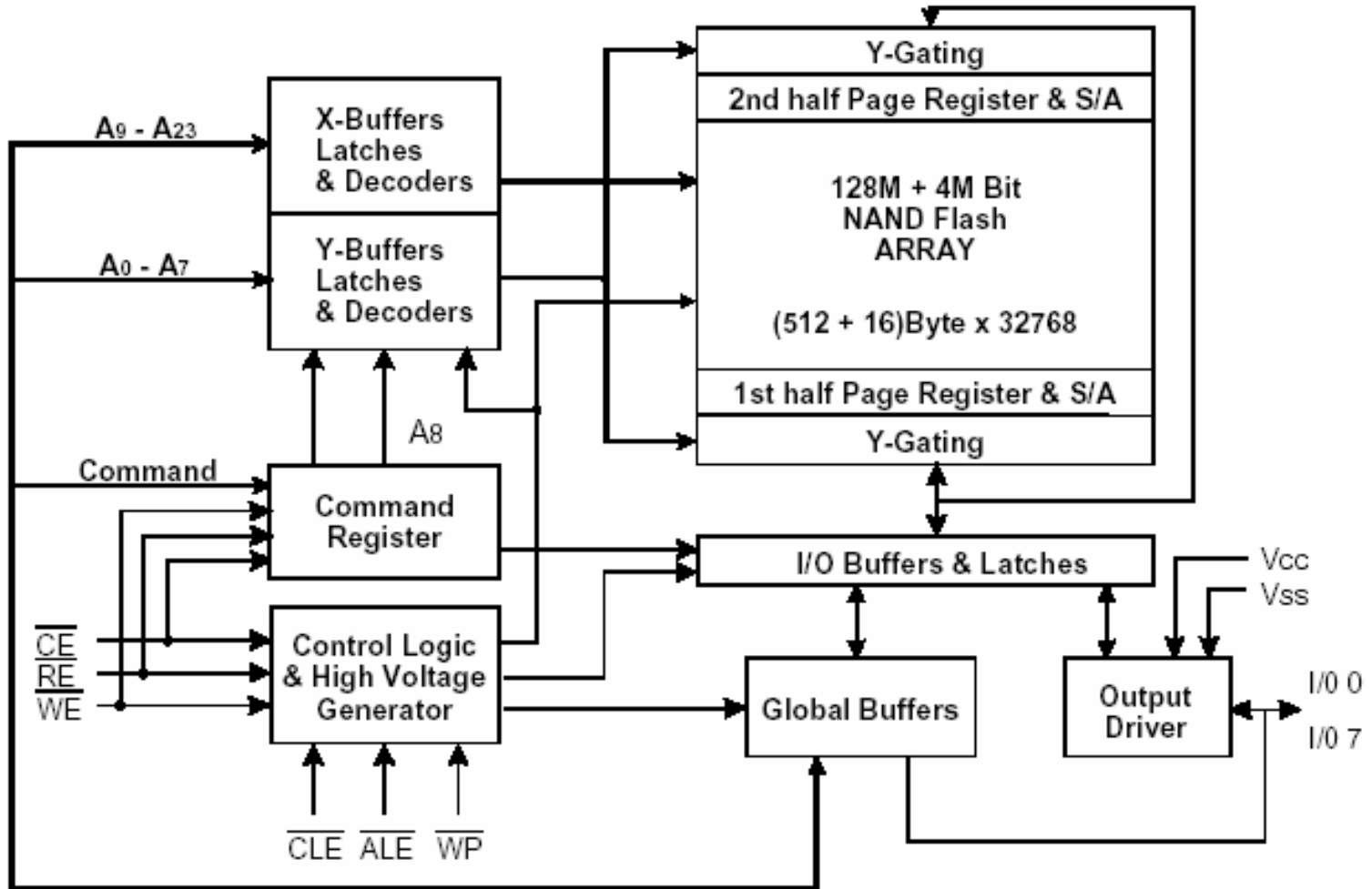
存储器接口-ROM(47)

5. Flash (K9F2808, NAND) - Pins

Pin Name	Pin Function
I/O ₀ - I/O ₇	Data Input/Outputs
CLE	Command Latch Enable
ALE	Address Latch Enable
$\overline{\text{CE}}$	Chip Enable
$\overline{\text{RE}}$	Read Enable
$\overline{\text{WE}}$	Write Enable
$\overline{\text{WP}}$	Write Protect
$\overline{\text{SE}}$	Spare area Enable
R/ $\overline{\text{B}}$	Ready/Busy output
V _{cc}	Power
V _{ss}	Ground
N.C	No Connection

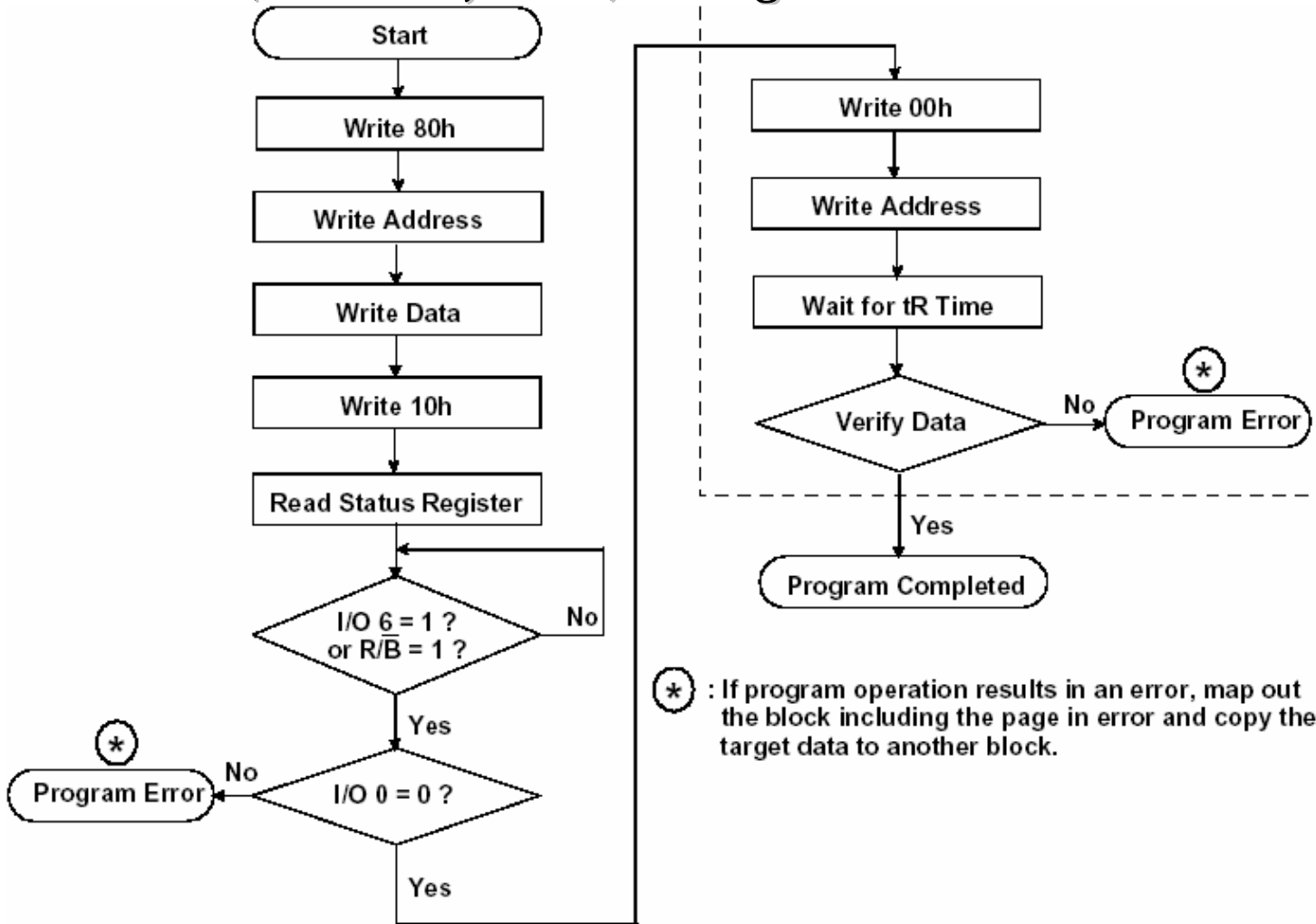
存储器接口-ROM(48)

5. Flash (K9F2808, NAND) -Blocks



存储器接口-ROM(49)

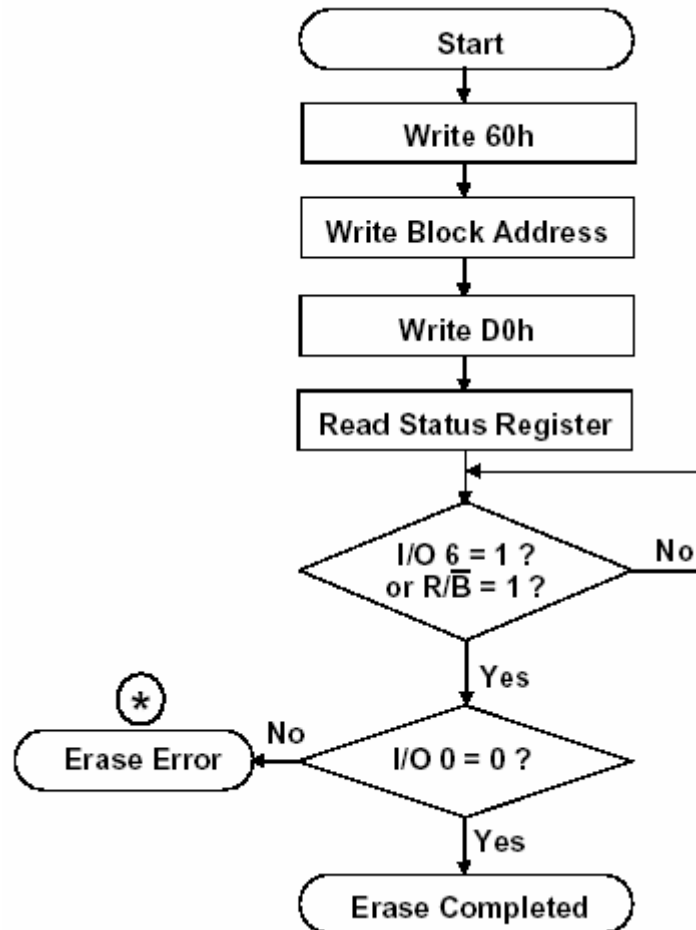
5. Flash (K9F2808, NAND) -Program



(*): If program operation results in an error, map out the block including the page in error and copy the target data to another block.

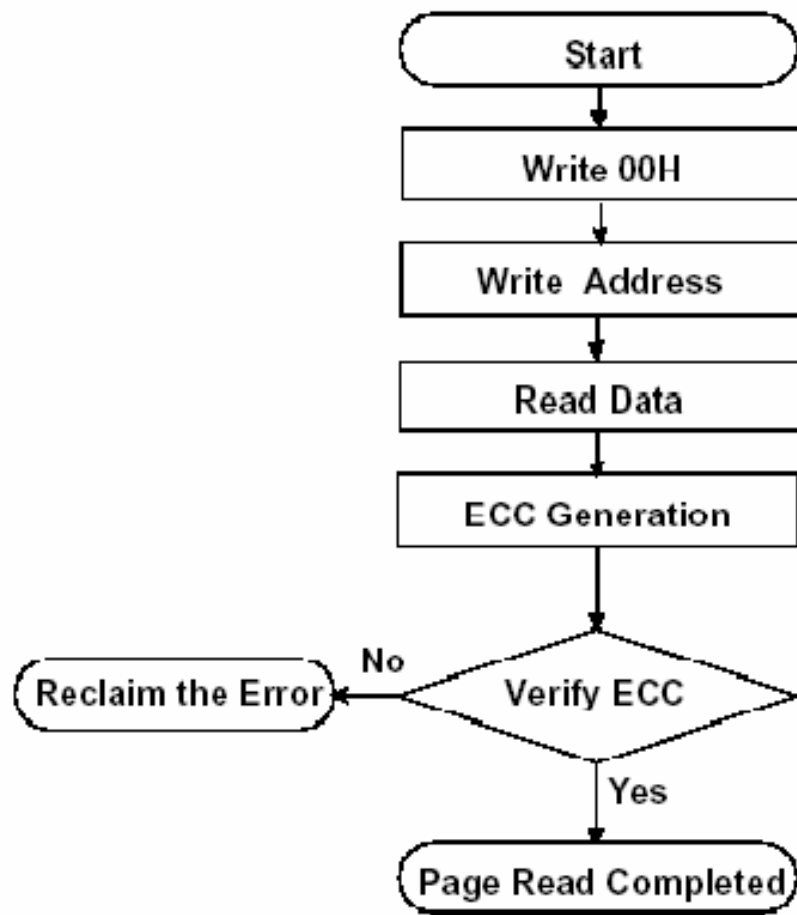
存储器接口-ROM(50)

5. Flash (K9F2808, NAND) -Erase



存储器接口-ROM(51)

5. Flash (K9F2808, NAND) -Read



存储器接口-ROM(52)

5. Flash (K9F2808, NAND) –Command

Function	1st. Cycle	2nd. Cycle	Acceptable Command during Busy
Read 1	00h/01h ⁽¹⁾	-	
Read 2	50h ⁽²⁾	-	
Read ID	90h	-	
Reset	FFh	-	0
Page Program	80h	10h	
Block Erase	60h	D0h	
Read Status	70h	-	0

1. The 00h command defines starting address of the 1st half of registers.
The 01h command defines starting address of the 2nd half of registers.
2. The 50h command is valid only when the SE(pin 6) is low level.

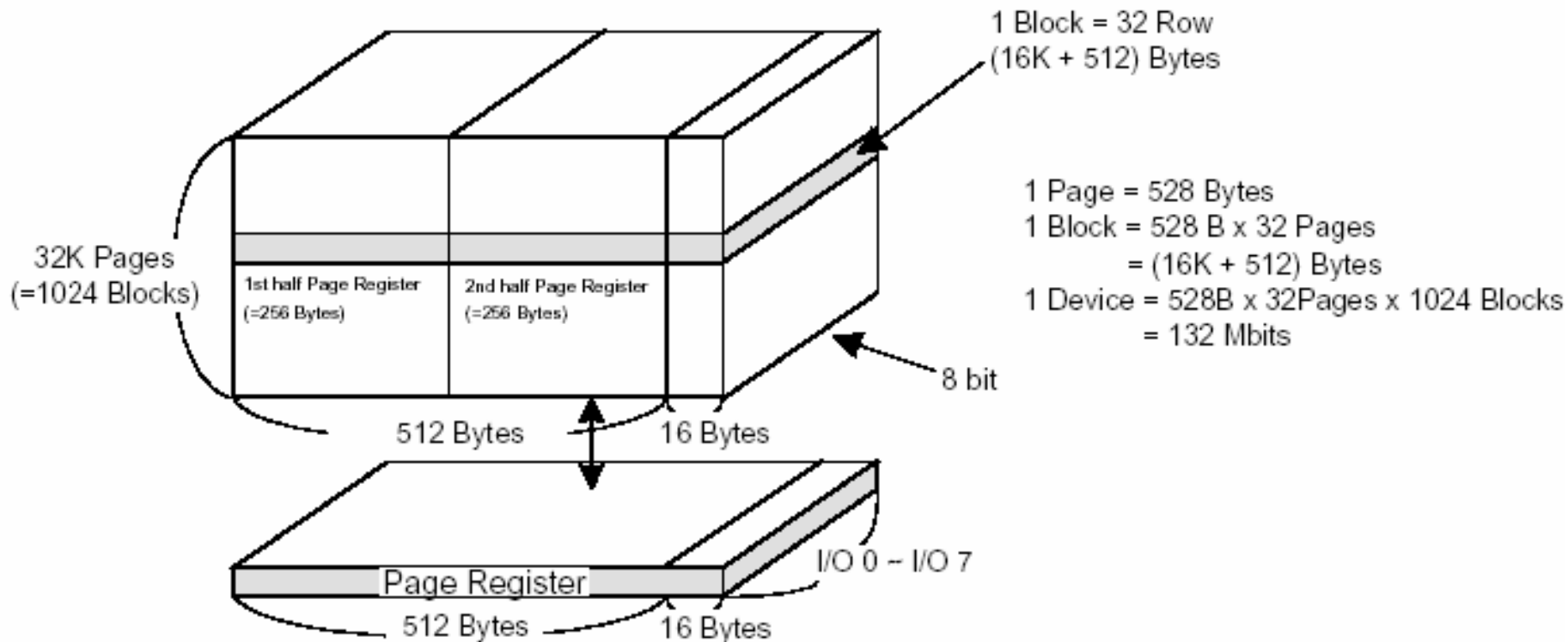
存储器接口-ROM(53)

5. Flash (K9F2808, NAND) -Address

	I/O 0	I/O 1	I/O 2	I/O 3	I/O 4	I/O 5	I/O 6	I/O 7	
1st Cycle	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	Column Address
2nd Cycle	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	Row Address (Page Address)
3rd Cycle	A ₁₇	A ₁₈	A ₁₉	A ₂₀	A ₂₁	A ₂₂	A ₂₃	*X	

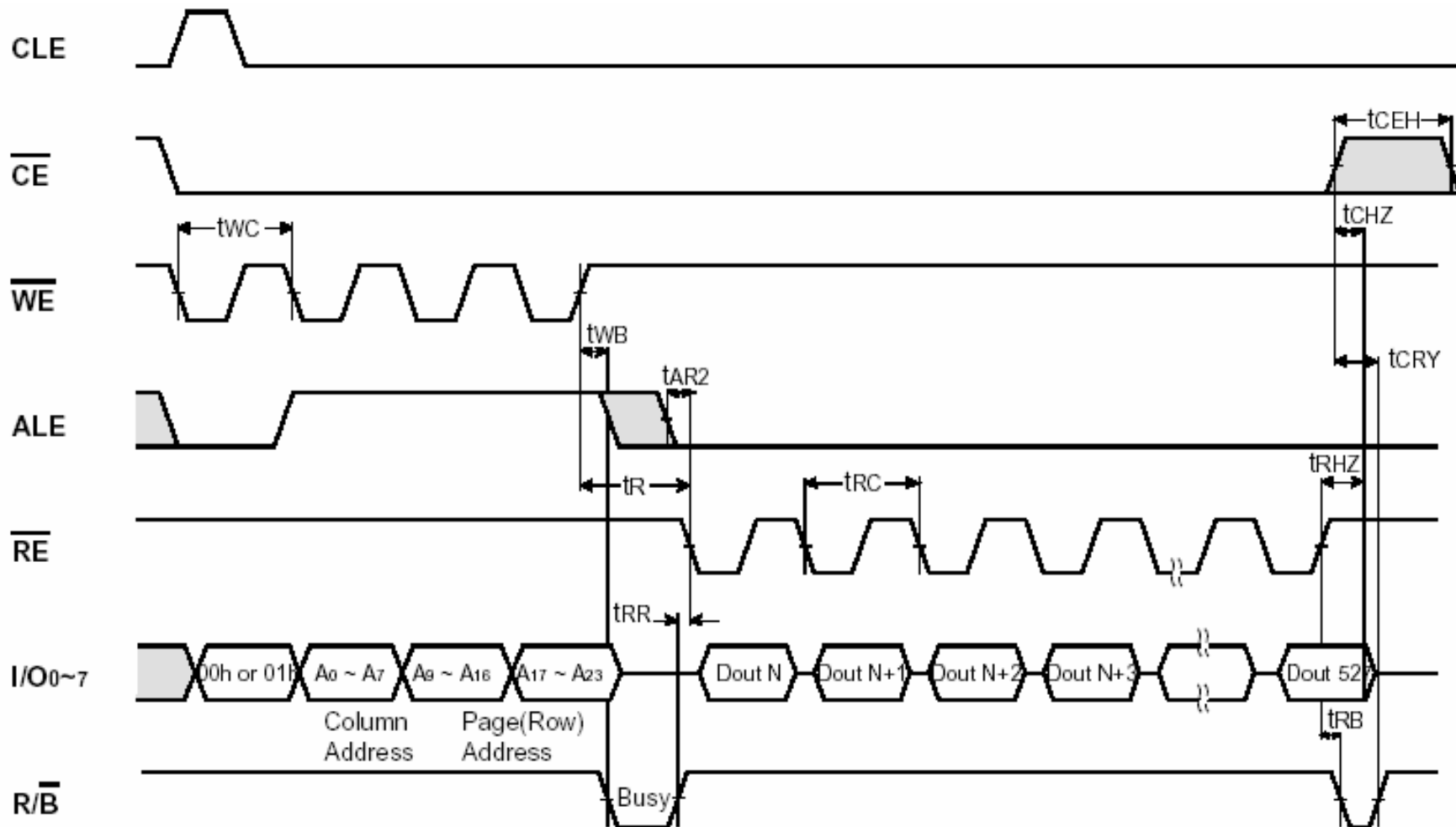
存储器接口-ROM(54)

5. Flash (K9F2808, NAND) - 数据结构



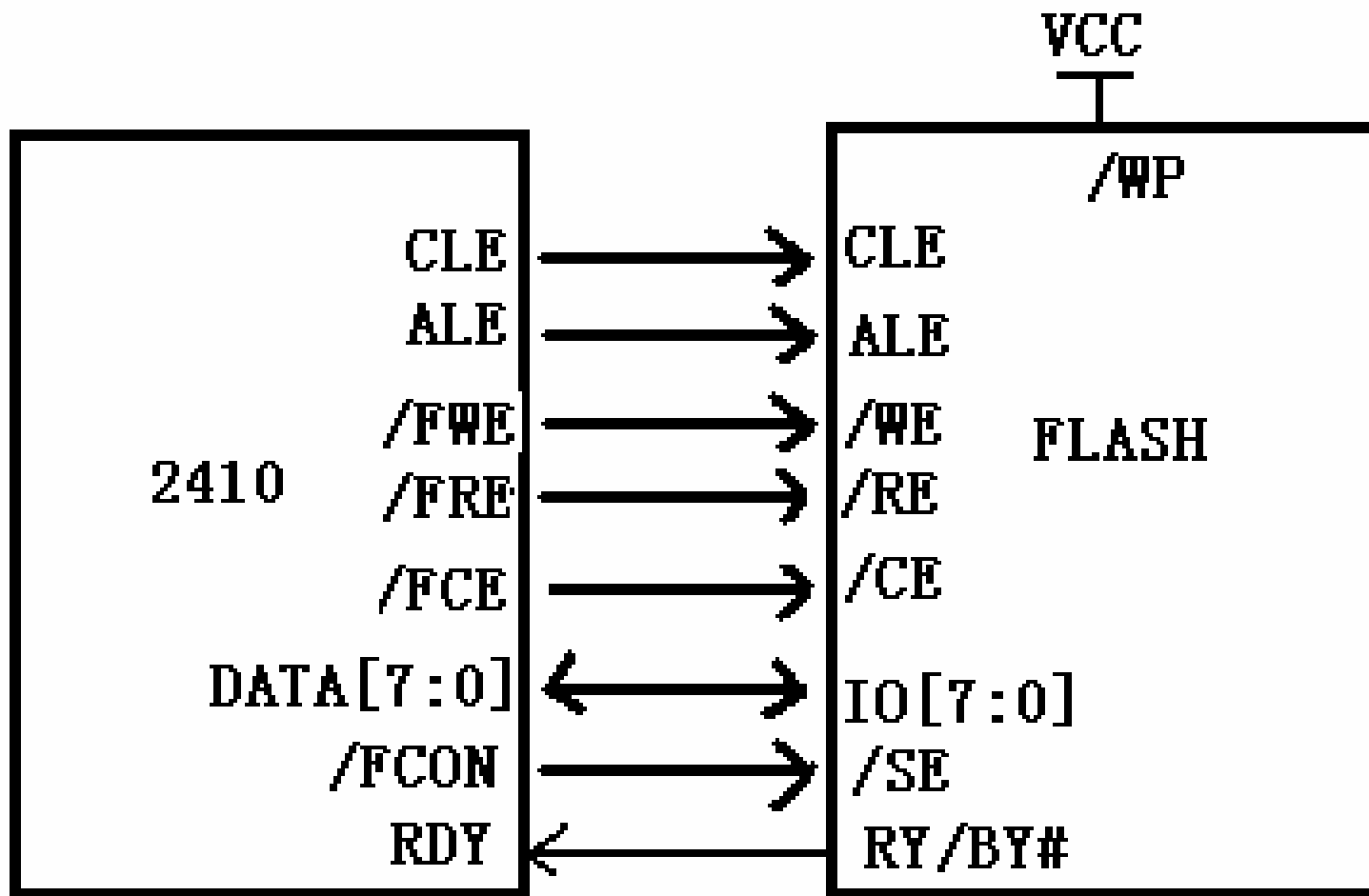
存储器接口-ROM(55)

5. Flash (K9F2808, NAND) -Timing



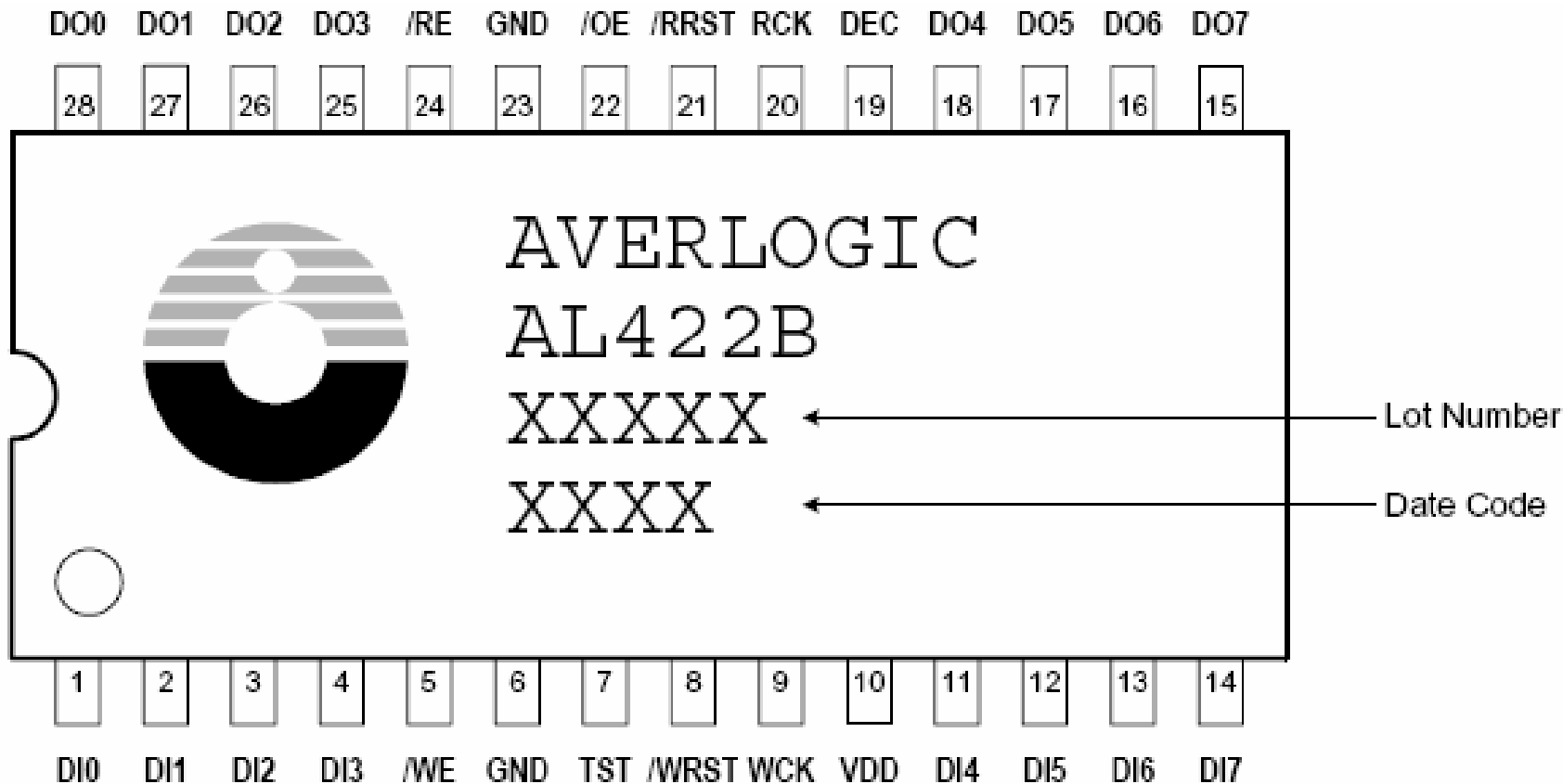
存储器接口-ROM(56)

5. Flash (K9F2808, NAND) -Connection



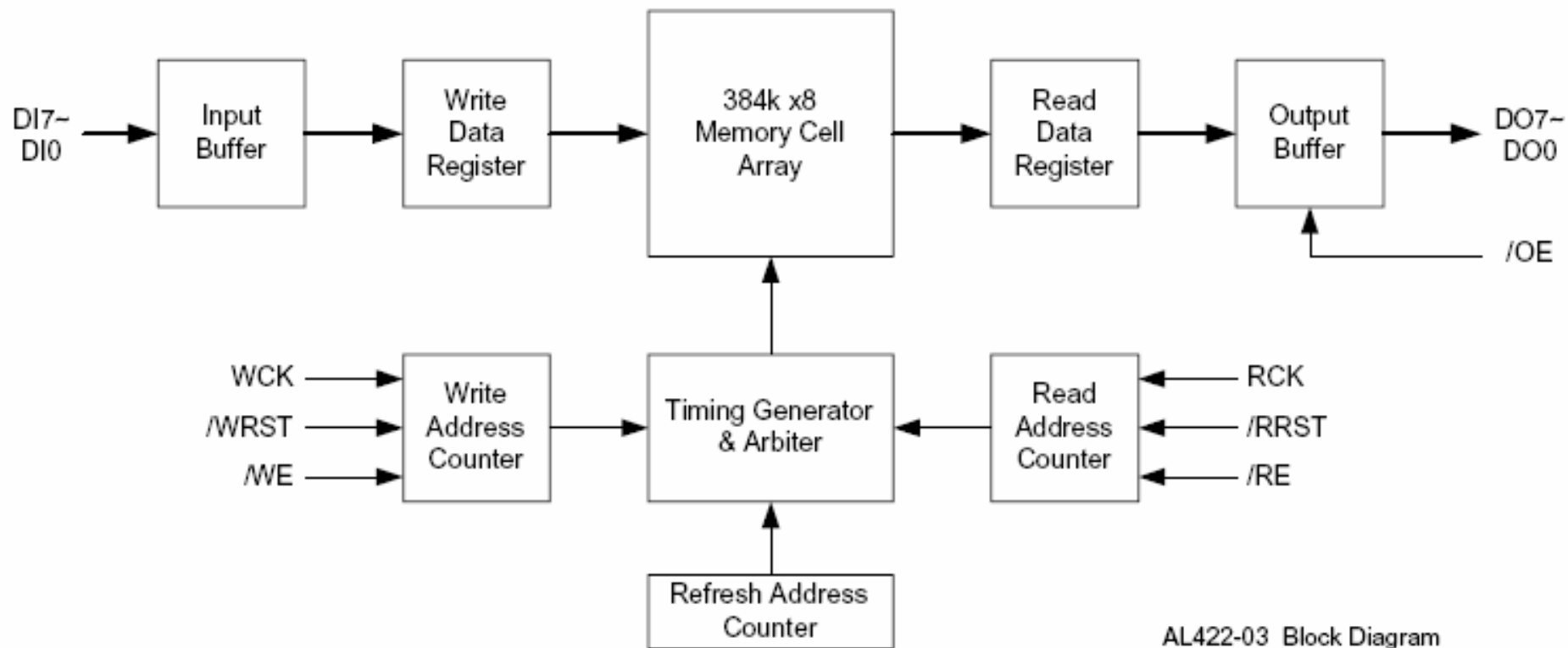
存储器接口--特殊Memory (1)

1. FIFO-AL422B-Package



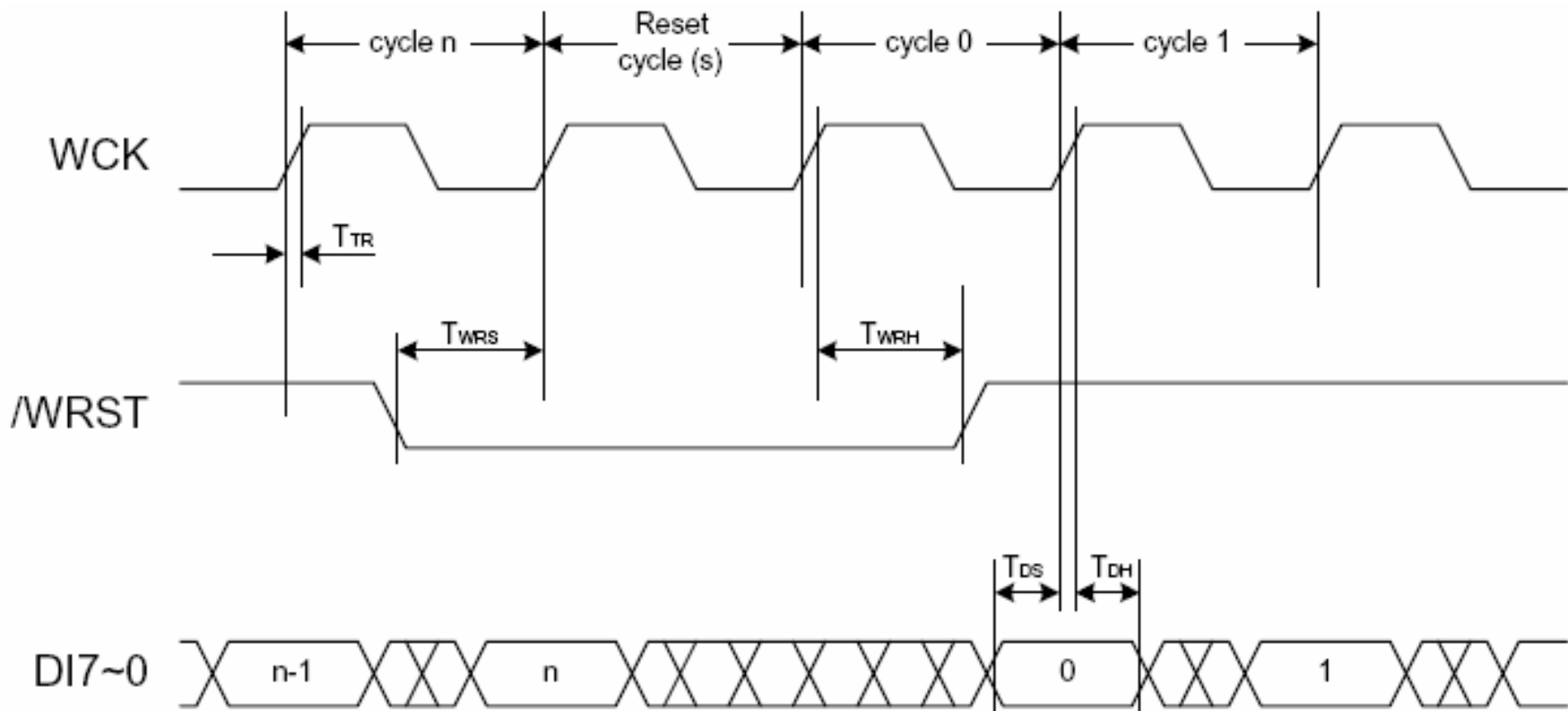
存储器接口--特殊Memory (2)

1. FIFO-Function Block



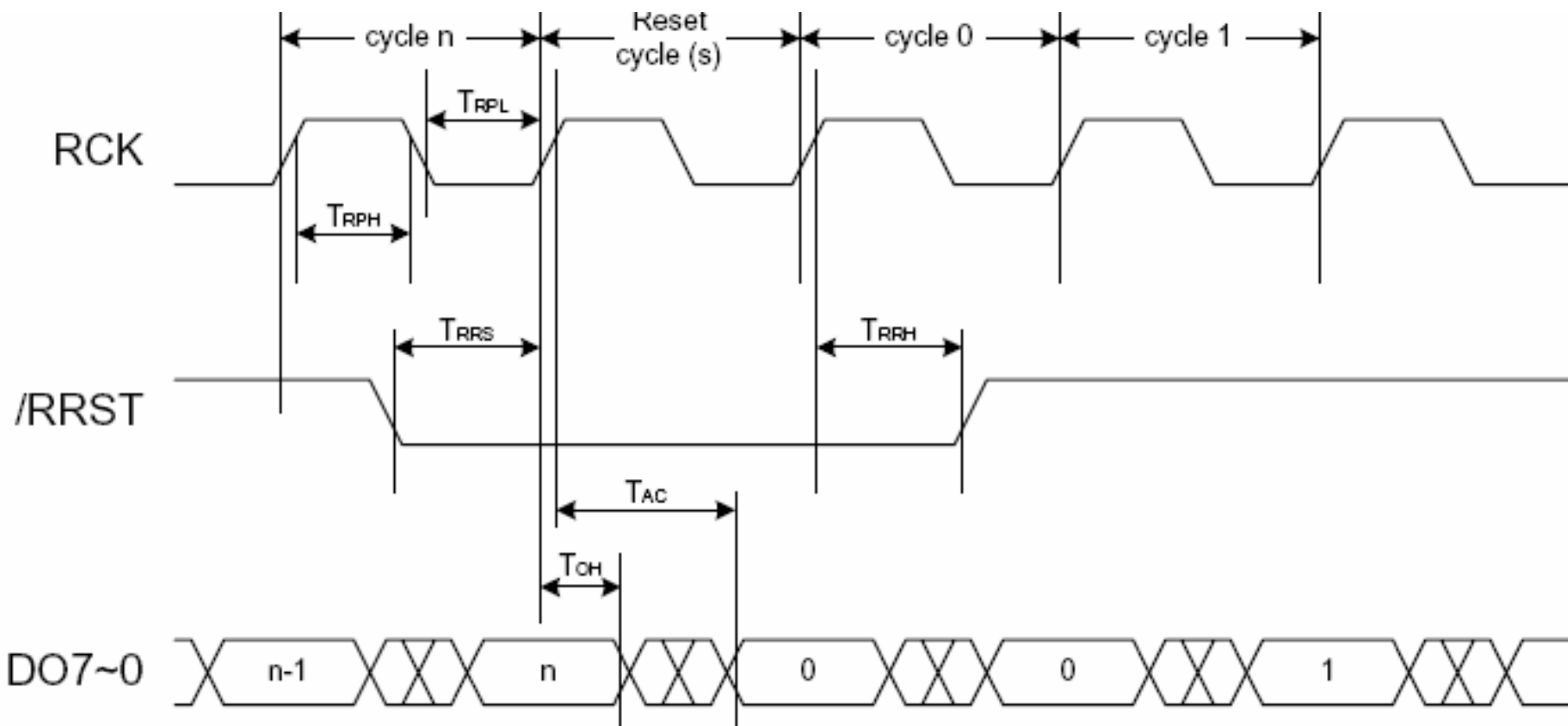
存储器接口--特殊Memory (3)

1. FIFO-Timing-Write



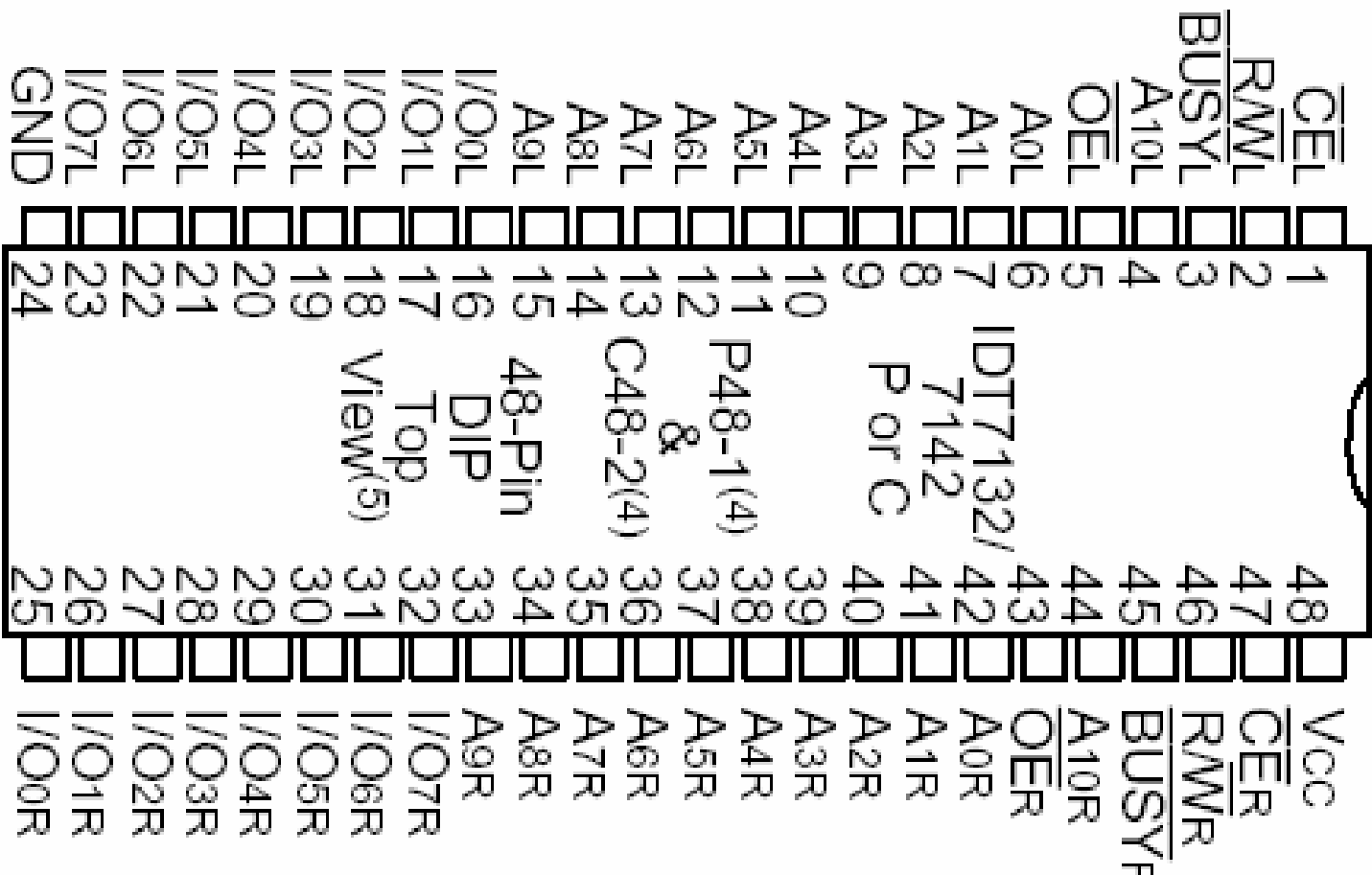
存储器接口--特殊Memory (4)

1. FIFO-Timing-Read



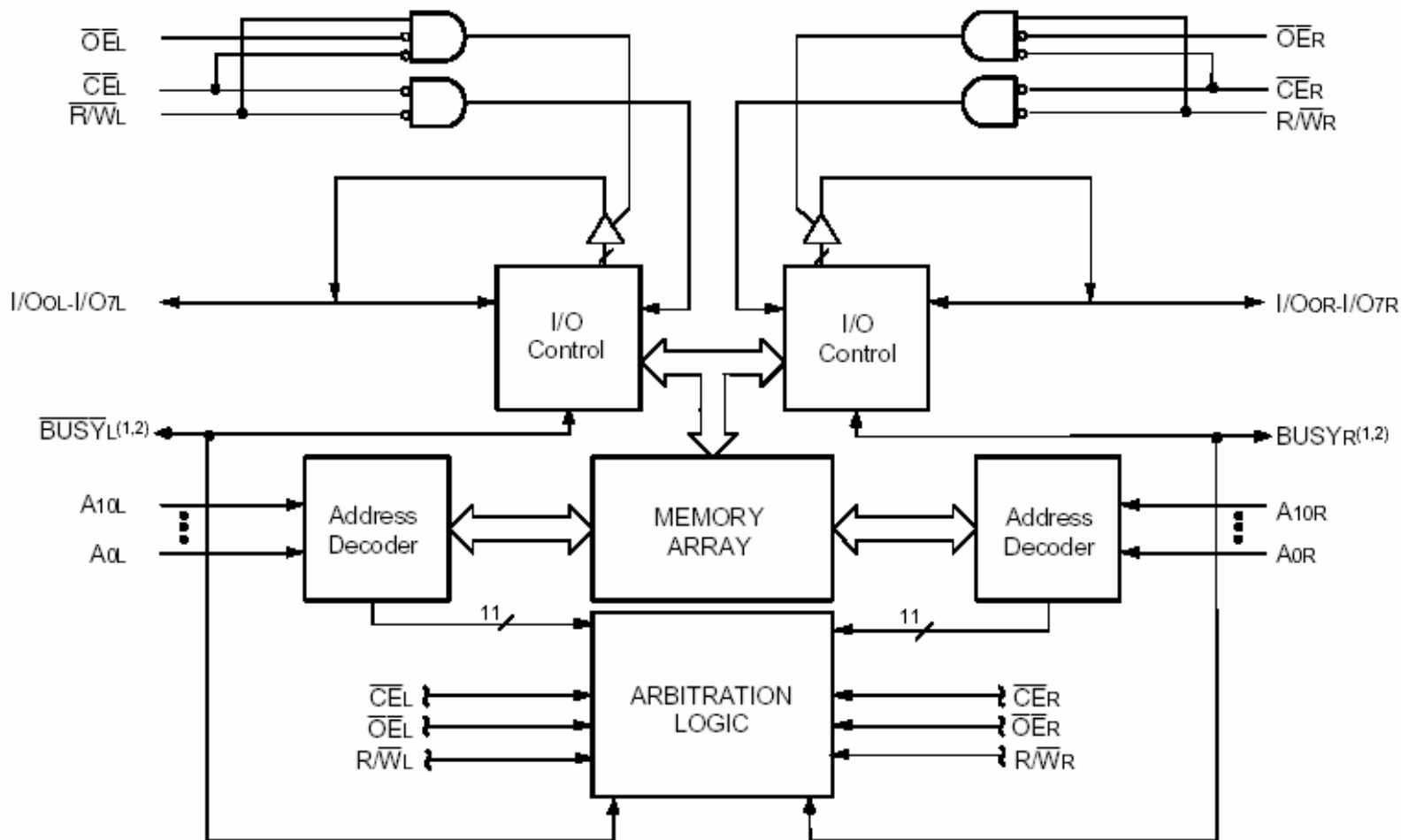
存储器接口--特殊Memory (5)

2. Dual Port-IDT7132-Package



存储器接口--特殊Memory (6)

2. Dual Port-IDT7132-Block





存储器接口--特殊Memory (7)

在哪些情况下需要用到这两种特殊存储器？

存储器接口—用好存储器 (1)

1. 选择-RAM

- Size:
 - Code, Data
 - Heap (dynamic allocating space)
 - Stack (template variables)
 - Base space (global variables)
- Speed:
 - Accessing time $T_{ac} < \text{extra bus cycle time } T_{cyc}$
(no waiting cycles)
 - Generally, $T_{ac} = T_{cyc} / 2$

存储器接口—用好存储器 (2)

1. 选择-RAM (续)

- Type

- Convenience (simple interface CPU)

SRAM > DRAM

- Speed (random access)

SRAM > DRAM

- Cost

SRAM < DRAM

- Power

SRAM < DRAM

存储器接口—用好存储器 (3)

1. 选择-ROM

- Size
 - Code
 - Data
- Speed
 - $T_{ac} < \max (T_{cyc} + \text{waiting cycles});$

存储器接口—用好存储器 (4)

1. 选择-ROM (续)

- Type

- Package

Serial < Parellel
(small)

- In-System Programmable

FLASH, EEPROM

- Reliability

ROM > PROM > EPROM > EERPOM/FLASH

- Running from rom

Parallel (ROM, PROM, NOR Flash)

- Cost

ROM < PROM < EPROM < EEPROM (FLASH)

存储器接口—用好存储器 (5)

2. 提高访问的可靠性

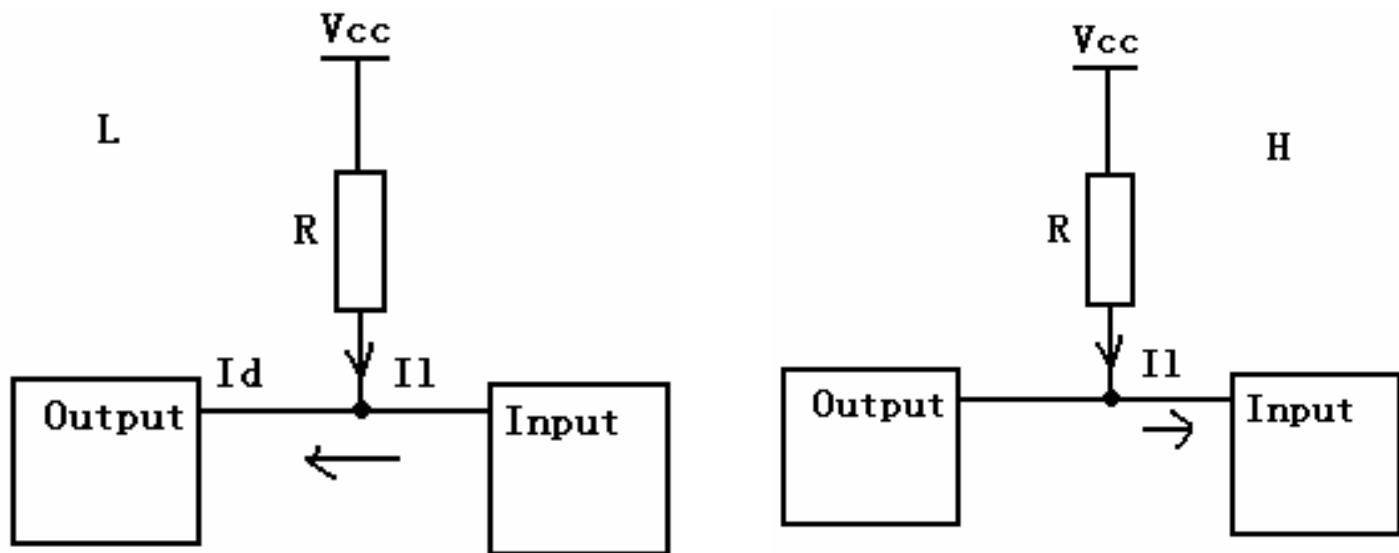
- 避免不确定状态
 - /CE, /WE, /RE 接上拉电阻, 控制线无效时disable ram
- 保证数据的有效时间
 - 访问时间大于存储器 T_{ac}
- 防止总线冲突
 - 总线控制权切换时加保护时间间隔
- 减少总线反射
 - 在数据总线上串联匹配电阻 (高频)

为什么大部分RAM的片选、读写控制信号采用低有效?

存储器接口—用好存储器 (5)

2. 上拉电阻的选择

- 什么是上拉电阻



存储器接口—用好存储器 (6)

2. 上拉电阻的选择

■ 计算方法

$$R_{\max} = \frac{V_{CC} \times (1 - 95\%)}{I_{l\max}}$$

$$R_{\min} = \frac{V_{CC} \times (1 - 5\%)}{I_{d\max}}$$

■ 常用

- TTL: 5K
- CMOS: 20-100K

CMOS SRAM

$V_{CC} = 3.3V$

$I_{l\max} = 2\mu A$

$I_{d\max} = 8mA$

求:

R_{\max}

R_{\min}

上拉电阻偏小有什么坏处?

存储器接口—用好存储器 (7)

3. EEROM的使用

- 延长使用寿命
 - 均衡使用整个区间
 - 减少擦写次数
- 防止系统异常时数据丢失
 - 先备份再擦写

4. 降低RAM功耗

- 关断时钟
- Disable CS
- Standby

存储器接口—小结

- 为什么需要不同种类的存储器，它们各有什么特点？
- 选择外存储器需要考虑哪些因素？

存储器接口—实验 (1)

- 内容
 - 编写ram测试程序。
 - 编写程序实现XOR Flash ROM的擦除、写入和读取。
- 目的
 - 掌握RAM的测试方法
 - 学会使用Flash ROM
- 实验平台
 - Embest ARM9实验版

存储器接口—实验 (2)

■ 相关知识

■ 内存测试

■ 目标

- 数据错误 (错误类型?)
- 地址错误
- 控制错误 (如何区别数据错误)

■ 方法

- 在存储器中写入数据然后读取验证。

■ 原则

- 不要在用不变的数进行测试
- 不要对同一地址先写接着读

■ 起始地址

- 0x3000_0000-0x3DF_FFFF (BankCon6)

存储器接口—实验 (3)

■ 相关知识

■ 内存初始化

- 设置SDRAM接口寄存器

■ Flash

- 型号: AT491614A (2M*8Byte, NOR)
- 起始地址: 0x0000_0000- 0x7FF_FFFF (Bankcon0)

■ 报告内容

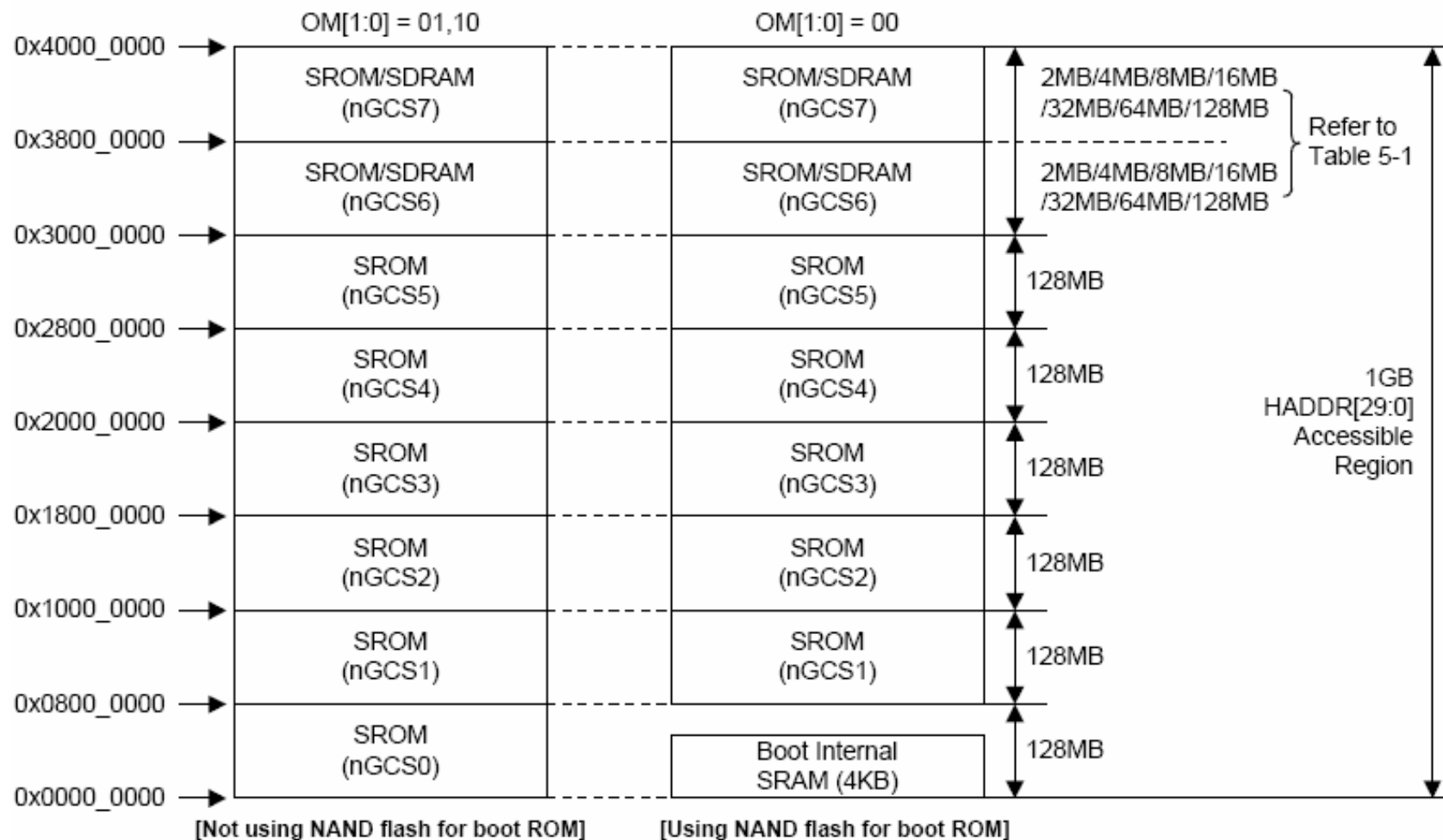
■ 测试程序及注释

■ 回答问题

- 如何分辨出内存错误类型?
- 在嵌入式考勤系统中, 每一次打卡记录都要保存, 并更新原有数据。由于Flash擦除时间较长, 用户觉得有等待时间, 如何解决这个问题?

存储器接口—实验 (4)

■ 附-2410存储器空间分配



存储器接口—实验 (5)

■ 附-SDRAM地址选择

Address	2MB	4MB	8MB	16MB	32MB	64MB	128MB
Bank 6							
Start address	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000
End address	0x301f_ffff	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff
Bank 7							
Start address	0x3020_0000	0x3040_0000	0x3080_0000	0x3100_0000	0x3200_0000	0x3400_0000	0x3800_0000
End address	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff	0x3fff_ffff

NOTE: Bank 6 and 7 must have the same memory size.

存储器接口—实验 (6)

■ 附-ROM空间位宽与Memory Pin

BANK0 BUS WIDTH

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

MEMORY (SROM/SDRAM) ADDRESS PIN CONNECTIONS

MEMORY ADDR. PIN	S3C2410X ADDR. @ 8-bit DATA BUS	S3C2410X ADDR. @ 16-bit DATA BUS	S3C2410X ADDR. @ 32-bit DATA BUS
A0	A0	A1	A2
A1	A1	A2	A3
...

存储器接口—实验 (7)

- 附-控制寄存器的初始值
 - 寄存器定义 (2410的Datasheet)
 - 初始化: (实验手册 p148)
 - 0x5300_0000: 0x0000_0000 //stop watchdog
 - 0x4A00_0008: 0xFFFF_FFFF //disable interrupt
 - 0x4C00_0004: 0x0005_c042
 -
 - 0x4800_0000: 0x2211_1110 //BWSCON
 - 0x4800_0004: 0x0000_0700 //BANKCON0
 -
 - 0x4800_001c: 0x0001_8005 //BANKCON6
 - 0x4800_0020: 0x0000_0700 //BANKCON7
 - 0x4800_0024: 0x008e_0459 //REFRESH
 - 0x4800_0028: 0x0000_00b2 //BANKSIZE
 - 0x4800_002c: 0x0000_0030 //MRSRB6
 - 0x4800_0030: 0x0000_0030 //MRSRB7