

DDR2 SDRAM

操作时序规范

基本功能

对 DDR2 SDRAM 的访问是基于突发模式的; 读写时, 选定一个起始地址, 并按照事先编程设定的突发长度 (4或8) 和突发顺序来依次读写. 访问操作开始一个激活命令, 后面紧跟的就是读或者写命令。和激活命令同步送达的地址位包含了所要存取的簇和行 (BA0, BA1 选定簇; A0-A13 选定行). 和读或写命令同步送达的地址位包含了突发存取的起始列地址, 并决定是否发布自动预充电命令。

在进行常用的操作之前, 要先对 DDR2 SDRAM 进行初始化. 下面的几小节介绍初始化的详细信息, 寄存器的定义, 命令的描述和芯片的操作。

上电和初始化

DDR2 SDRAM必须以预定义的时序进行上电和初始化。不按规定的操作将导致不可预期的情况出现。。

上电和初始化时序

1. 对于上电和初始化来说, 下列时序是必须的。

供电且保持 CKE 低于 $0.2 \cdot V_{DDQ}$, ODT^{*1} 要处于低电平状态 (所有的其余脚可以都没有定义.) 电源上升沿不可以有任何翻转, 上升沿时间不能大于 200mS; 并且要求在电压上升沿过程中满足, $V_{DD} > V_{DDL} > V_{DDQ}$ 且 $V_{DD} - V_{DDQ} < 0.3$ volts.

- V_{DD}^{*2} , V_{DDL}^{*2} 和 V_{DDQ} 必须由同一个电源芯片供电, 并且
- V_{TT} 最大只能到 0.95 V, 并且
- V_{ref} 要时刻等于 $V_{DDQ}/2$, 紧跟 V_{DDQ} 变化。

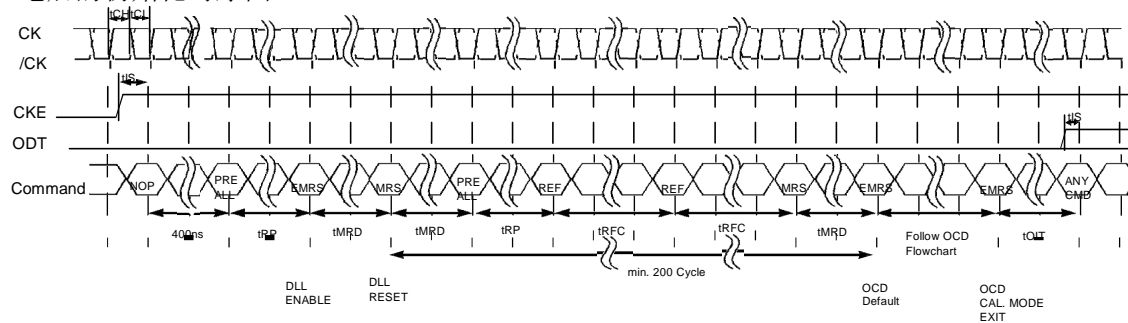
或者

- 在给 V_{DDL} 上电的同时或之前就给 V_{DD}^{*2} 上电
- 在给 V_{DDQ} 上电的同时或之前就给 V_{DDL}^{*2} 上电
- 在给 V_{TT} & V_{REF} 上电的同时或之前就给 V_{DDQ} 上电。

上面的两个条件至少要满足一个。

2. 开始时钟信号并保持信号稳定。
 3. 在稳定电源和时钟 (CK, /CK) 之后至少 200 μ S, 然后发布 NOP 或者取消选定命令 & 拉高 CKE.
 4. 等待至少 400ns 然后发布预充电所有簇命令. 在等待的 400ns 过程中要发布 NOP 或者取消选定命令。
 5. 发布 EMRS(2) 命令. (EMRS(2) 命令, 需要将 BA0 拉低, 将 BA1 拉高.)
 6. 发布 EMRS(3) 命令. (为了发布 EMRS(3) 命令, 将 BA0 和 BA1 拉高.)
 7. 发布 EMRS 命令以激活 DLL. (为了发布 "DLL 激活" 命令, 将 A0 拉低, BA0 拉高 并且将 BA1-2 和 A13-A15 置低.)
 8. 发布 MRS 命令实现 "DLL 复位"².
(为了发布 DLL 复位命令, 需要将 A8 拉高 并使 BA0-1 为低)
 9. 发布预充电所有簇命令。
 10. 至少发布两次自动刷新命令。
 11. 将 A8 拉低, 发布模式寄存器设定命令 (MRS) 对芯片进行初始化操作. (也就是不对 DLL 复位, 编程芯片的操作参数)
 12. 在第 8 步之后至少过 200 个时钟周期, 执行 OCD 校准 (片外驱动电阻调校).
如果不使用 OCD 校准, EMRS OCD 校准模式结束命令 (A9=A8=A7=0) 必须在 EMRS OCD 默认命令 (A9=A8=A7=1) 之后发布, 用来设定 EMRS 的其它操作参数。
 13. 现在, DDR2 SDRAM 就准备好可以进行普通的操作了。。
- *1) 为了保证 ODT 关闭, V_{REF} 必须有效 并且 ODT 脚必须拉低.
- *2) 如果 V_{DDL} 或 V_{DD} 的电平值在正常操作过程中人为改变, (例如 e, 为了 V_{DD} 相交测试, 或者节省功率) 则必须执行 "DLL 复位".

上电后的初始化时序图



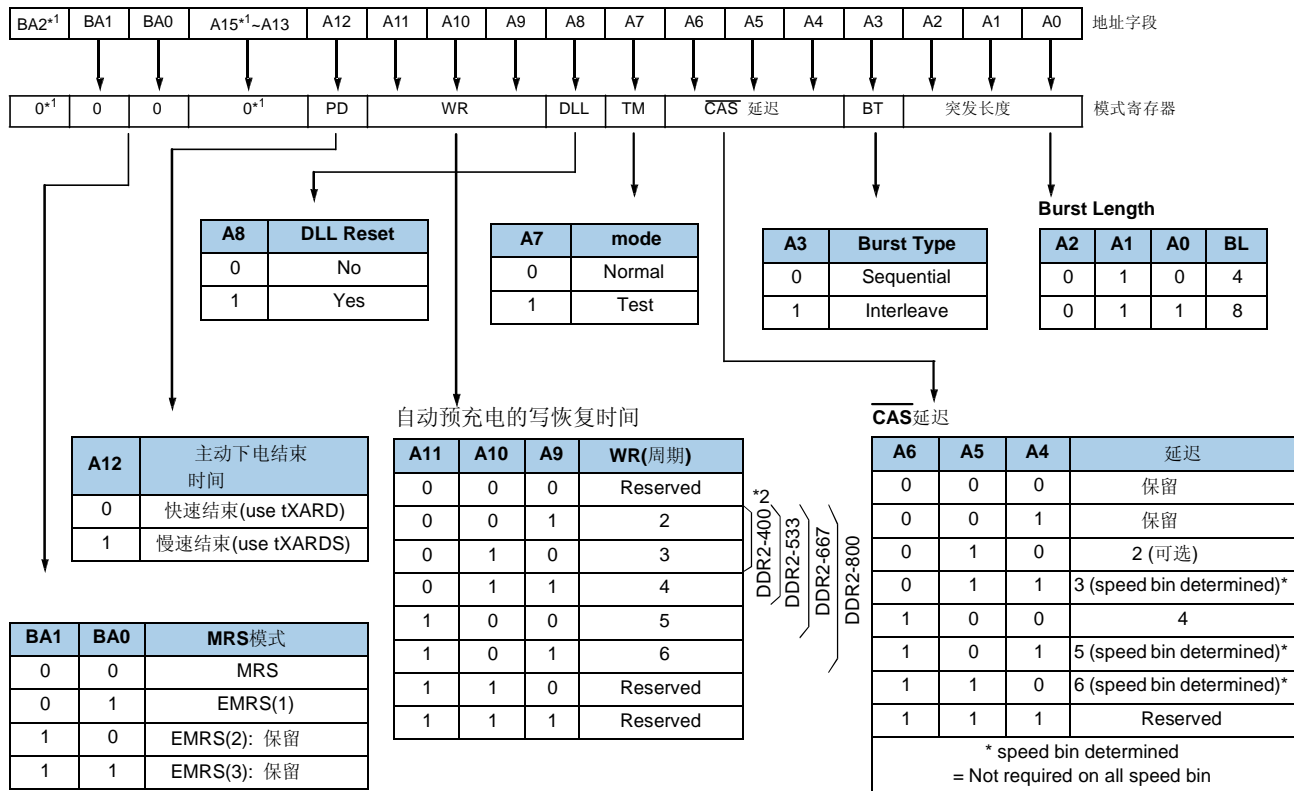
编程（扩展）模式寄存器

为了增强使用灵活性，突发长度，突发类型，/CAS延迟，DLL 复位功能，写恢复时间(WR) 都是可以定义的变量。通过预先编程模式寄存器就可以设定它们。除此之外，DLL 禁止功能，驱动电阻，附加 CAS 延迟，ODT(终结电阻)，单线选通，和 OCD（片外驱动电阻调整）同样也是可以定义的变量，通过预先编程扩展模式寄存器就可以设定它们。模式寄存器或者扩展模式寄存器的内容可以通过对他们进行重新设定来改变它们。如果用户只是想改变寄存器中某些特定的位，也必须通过MRS或EMRS命令对全部寄存器位进行重新设定。

MRS, EMRS和t DLL 复位这些命令并不会影响存储阵列的内容，这意味着上电后的任意时间执行初始化操作不会改变存储的内容。

DDR2 SDRAM 模式寄存器设定 (MRS)

模式寄存器中的数据控制着 DDR2 SDRAM 的操作模式. 它控制着 CAS 延迟, 突发长度, 突发顺序, 测试模式, DLL 复位, WR 等各种选项, 支持着 DDR2 SDRAM 的各种应用. 模式寄存器的默认值没有被定义, 所以上电之后必须按规定的时序规范来设定模式寄存器的值. 通过将 CS, RAS, CAS, WE, BA0, BA1 置低来发布模式寄存器设定命令, 操作数通过地址脚 A0 ~ A15 同步送出. DDR2 SDRAM 在写模式寄存器之前, 应该通过拉高 CKE 而完成了所有簇的预充电. 模式寄存器设定命令的命令周期 (tMRD) 必须满足完成对模式寄存器的写操作. 在进行正常操作时, 只要所有的簇都已经处于预充电完成状态, 模式寄存器都可以使用同一命令重新设定. 模式寄存器不同的位表示不同的功能. A0 ~ A2 设定突发长度是 4 还是 8. 突发长度的译码规则与 DDR SDRAM 相同. A3 定义了突发地址顺序. A4 ~ A6 定义了 CAS 延迟. DDR2 不支持半时钟延迟. A7 设定测试模式. A8 设定 DLL 复位. 对通常的 MRS 操作, A7 必须设定为低. A9 ~ A11 定义了写恢复时间 WR. 详细情况见下面的示意图: .



*1: A13 现在保留, 以后备用, 设置模式寄存器时要设为 0.

BA2 和 A14 对于 512Mb DDR2 不使用, 但对于 1Gb 和 2Gb DDR2 SDRAMs 使用. A15 保留, 以后备用.

*2: WR (自动预充电的写恢复时间) 的最小值由 tCK 的最大值决定, 而 WR 最大值由 tCK 最小值决定. WR 的计算方法是看 tWR 有几个时钟周期, 如果是小数, 则进位为大一的整数 (WR[cycles] = tWR(ns)/tCK(ns)). 模式寄存器必须编程为这个值. 同样的方法用 tRP 决定 tDAL.

DDR2 SDRAM 扩展模式寄存器设定

EMRS(1)

扩展模式寄存器(1) 存储着激活或禁止DLL的控制信息, 输出驱动强度, ODT 值的选择 和附加延迟等信息. 扩展寄存器(1) 的默认值没有被定义, 因此, 上电之后, 扩展模式寄存器(1) 的值必须按正确的步骤来设定. 写扩展模式寄存器(1) 是通过拉低CS, RAS, CAS, WE, 置低 BA0, 同时控制地址线 A0 ~ A13的状态. 在写扩展模式寄存器(1) 之前, DDR2 SDRAM 应该通过将 CKE拉高完成所有簇的预充电. 扩展模式寄存器(1) 设定命令的命令周期 (tMRD)必须满足完成对扩展模式寄存器(1) 的写操作. 在进行正常操作时, 只要所有的簇都已经处于预充电完成状态, 扩展模式寄存器(1) 都可以使用同一命令重新设定.. A0控制着DLL 激活或禁止. A1被用于激活数据输出驱动能力为一半. A3~A5 决定着附加延迟, A2和 A6 用语 ODT 值的选定, A7~A9 用于控制 OCD, A10 被用于禁止 DQS#, A11 被用于 RDQS 的激活.

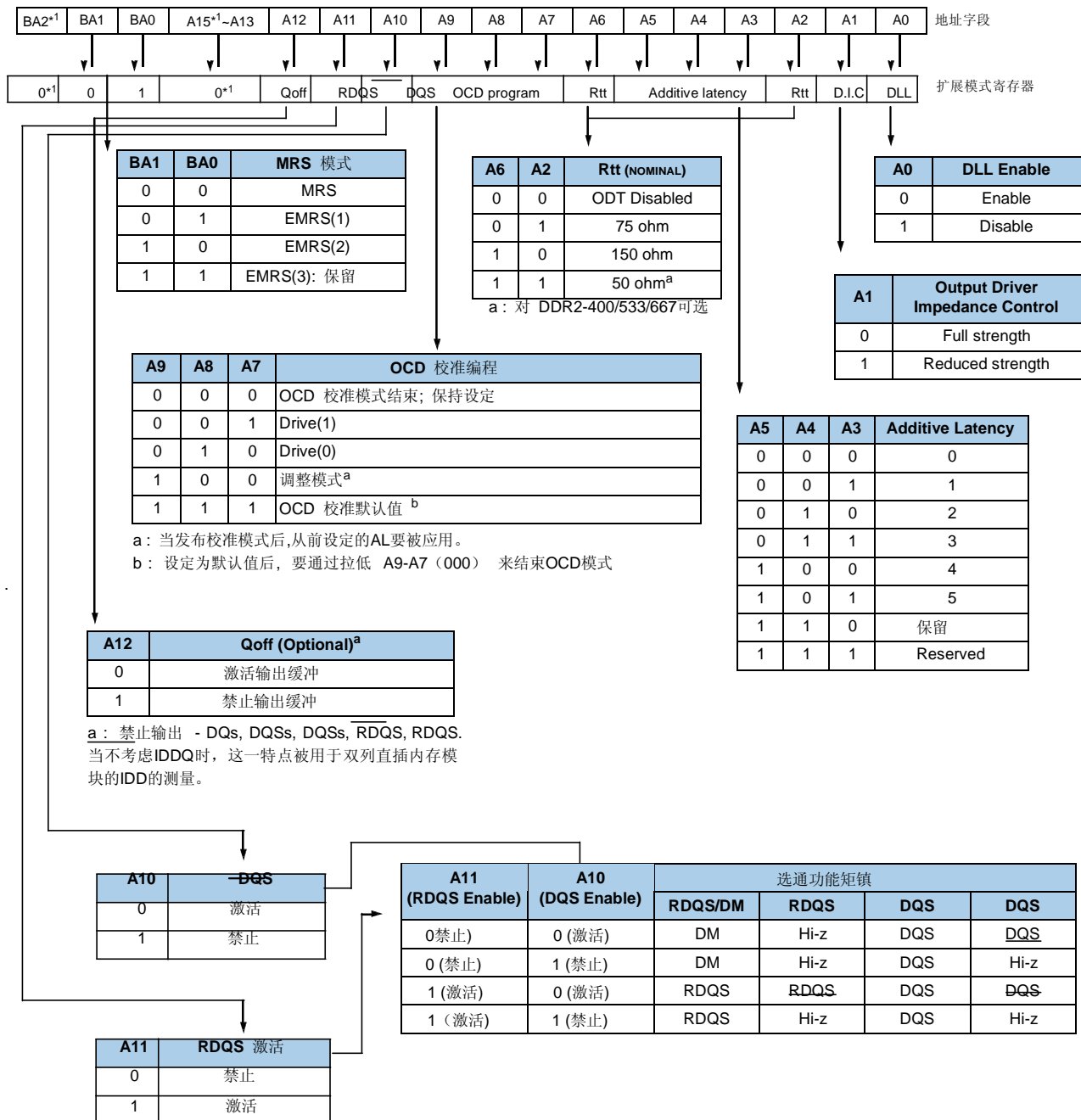
DLL 激活/禁止

对通常的操作, DLL必须被激活. 在上电初始化过程中, 必须激活 DLL, 在开始正常操作时, 要先关闭DLL. 在进入自我刷新操作时, DLL会被自动禁止, 当结束自我刷新时, DLL会被自动激活. 一旦 DLL被激活(随之将复位), 为了使外部时钟和内部始终达到同步, 在发布读命令之前必须至少要过200个时钟周期. 没有等待同步可能会导致tAC 或 tDQSCK参数错误.

EMRS(2)

扩展模式寄存器(2)控制着刷新和相关的特性. 扩展模式寄存器(2)的默认值没有被定义, 因此在上电后, 必须按规定的时序对扩展模式寄存器(2) 进行设定. 通过拉低S, RAS, CAS, WE,置高 BA1 拉低 BA0来发布扩展模式寄存器(2) 的设定命令. 同时控制地址线A0 ~ A15的状态. 在写扩展模式寄存器(2) 之前, DDR2 SDRAM 应该通过将 CKE拉高完成所有簇的预充电. 扩展模式寄存器(2) 设定命令的命令周期 (tMRD)必须满足完成对扩展模式寄存器(2) 的写操作. 在进行正常操作时, 只要所有的簇都已经处于预充电完成状态, 扩展模式寄存器(2) 都可以使用同一命令重新设定.

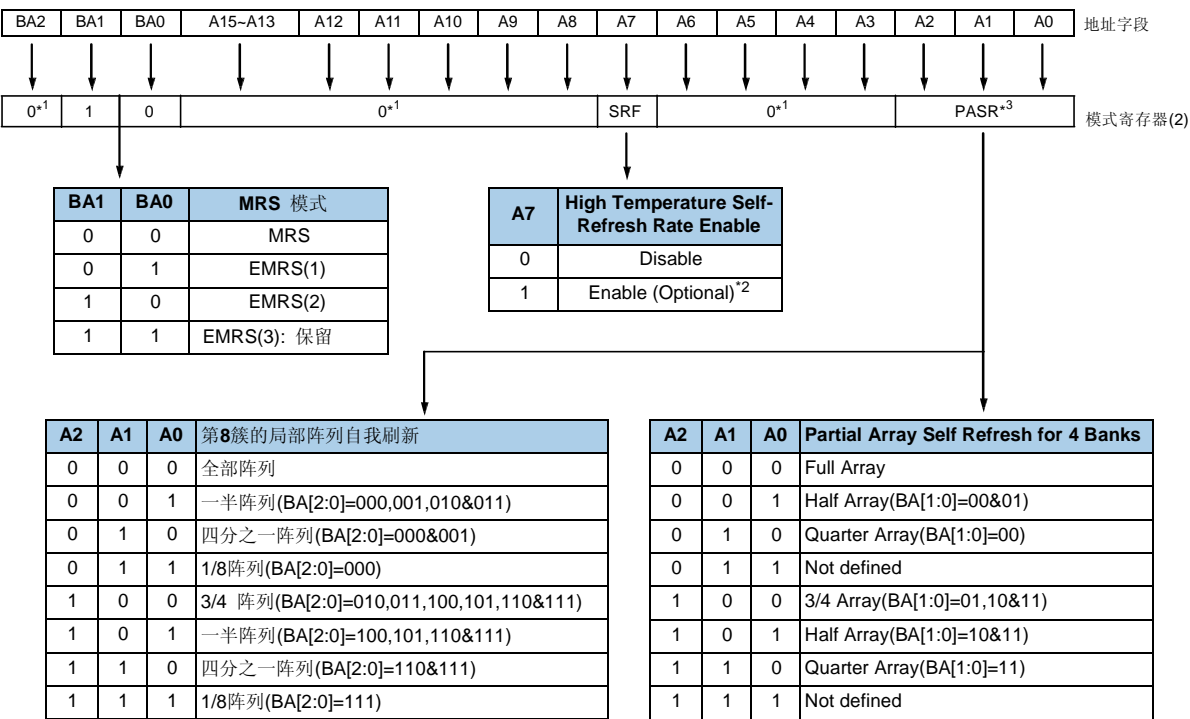
EMRS(1) 编程



*如果 RDQS被激活, 那么 DM 功能将被禁止. 当写数据时 RDQS 是活动的, 写命令时则不关心RDQS的状态。

*1: BA2 和 A13~A15是保留以被以后扩展用, 编程时要设置为0.

EMRS(2) 编程

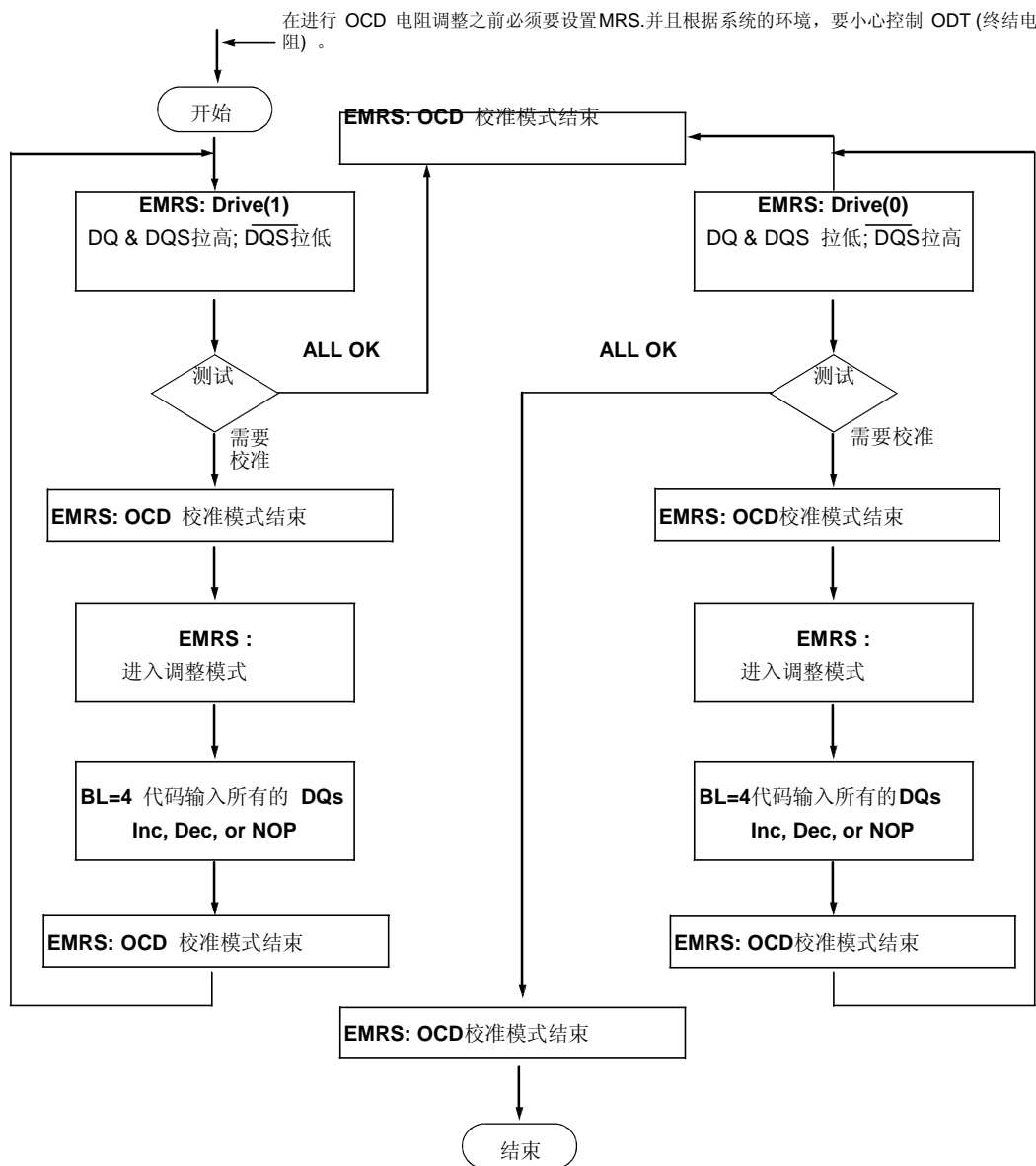


Device Operations

片外驱动 (OCD) 电阻调整

DDR2 SDRAM

DDR2 SDRAM 支持驱动校准特性， 调整的流程如下图所示。 每执行一次校准命令，都要在之后紧跟着“OCD 校准模式结束”命令，之后才可以发布其它的命令。 在进行 OCD 电阻调整之前必须要设置MRS.并且根据系统的环境，要小心控制 ODT (终结电阻)。



针对 **OCD**电阻调整的扩展模式寄存器设定

可以使用下面的 **EMRS** 模式完成OCD电阻调整。在驱动模式（drive）下，所有的输出都由DDR2 SDRAM驱动，且RDQS的驱动依赖于EMRS寄存器的RDQS激活位。在驱动(1) 模式下，所有的 DQ, DQS (和 RDQS) 信号都被拉高，并且所有的/DQS都被置低电平。 在驱动(0)模式下，所有的 DQ, DQS (和 RDQS) 信号都被清0并且所有的/DQS都被拉高。在调整（adjust）模式下，BL = 4 的操作码必须被使用。如果 OCD 校准失败，那么在标准温度和电压情况下，输出驱动有一个18欧姆的标准电阻。OCD 校准失败情况下的输出驱动标准在表格中有详细说明。OCD 应用只是针对由EMRS(1)定义的最大驱动力的情况。如果定义为半驱动力，那么OCD默认的输出启动特性将无效。当 OCD 校准调试模式使用之后，OCD默认的输出驱动特性将无效。当OCD校准完成或者驱动力被设置成为默认的情况，随之而来的EMRS命令不可以再调整OCD.所以A9-A7必须设置为000。这样就可以保持默认的值或者之前调整好的值。

片外驱动编程

A9	A8	A7	Operation
0	0	0	OCD 校准模式结束
0	0	1	Drive(1) DQ, DQS, (RDQS) 高 且 \overline{DQS} 为低
0	1	0	Drive(0) DQ, DQS, (RDQS) 为低切 \overline{DQS} 为高
1	0	0	调整模式
1	1	1	OCD 校准默认值

OCD 电阻调整

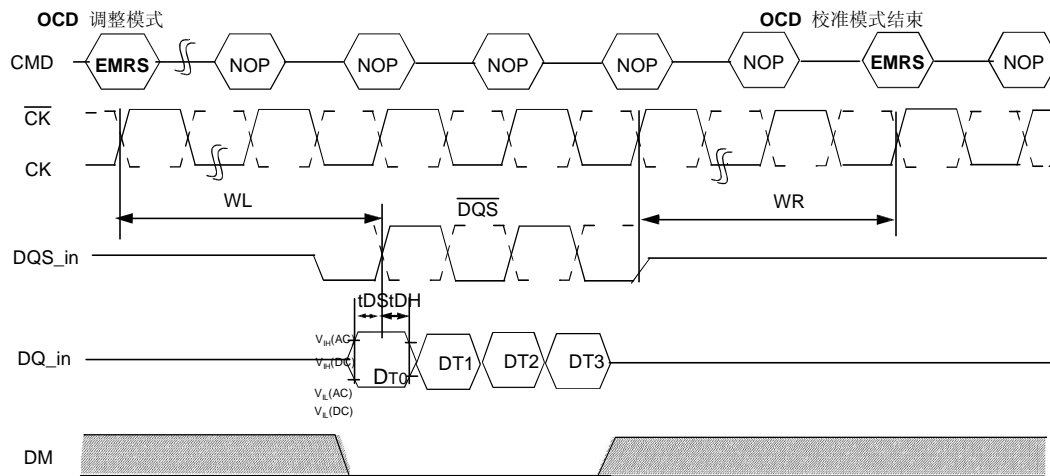
为了调整输出驱动电阻,控制器必须同EMRS调整命令同步发送4位的突发码到DDR2 SDRAM,如下表。为进行此步操作,必须在激活OCD之前通过设定MRS命令将突发长度必须设定为 BL = 4，并且控制器必须同时驱动突发码到所有的 DQs 。下表中的DT0 意思是所有DQ在位时间0的时刻，DT1在时间1，依次类推。 在OCD之后或同时，DDR2 SDRAM所有的DQs的驱动输出电阻的值被调整好。

这样，所有的DDR2 SDRAM 的DQs的驱动能力被调整为一致。每一步最大可调整16，如果达到了这个限制，则要求更快的增大或减小则是无效的。默认的设置可以是16以内的任何值。当校准命令发布以后，以前设定的AL值必须被应用起来。

OCD 编程

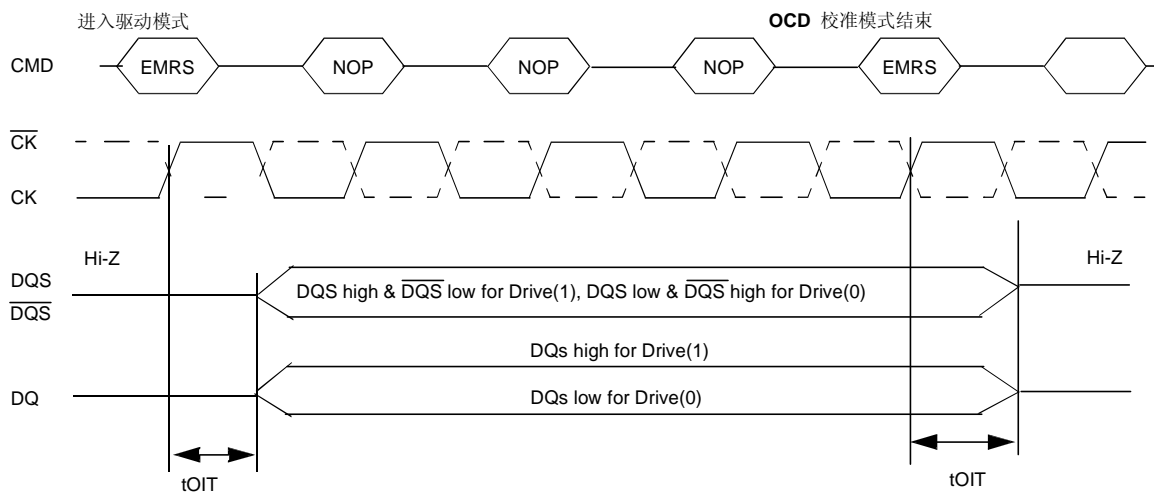
所有 DQs的四位突发码				操作	
DT0	DT1	DT2	DT3	上拉驱动能力	下拉驱动力
0	0	0	0	NOP (无操作)	NOP (无操作)
0	0	0	1	按1累加	NOP
0	0	1	0	按1递减	NOP
0	1	0	0	NOP	按1累加
1	0	0	0	NOP	按1递减
0	1	0	1	按1累加	按1累加
0	1	1	0	按1递减	按1累加
1	0	0	1	按1累加	按1递减
1	0	1	0	按1递减	按1递减
其余组合				保留	

为了正确的操作调整模式, $WL = RL - 1 = AL + CL - 1$ 个时钟和 tDS/tDH 要满足下面的时序图.对于调整模式的输入数据来说, DT0 - DT3是固定顺序的, 并且不会被MRS的地址模式所影响。(ie.连续情况或间隔情况).



驱动模式

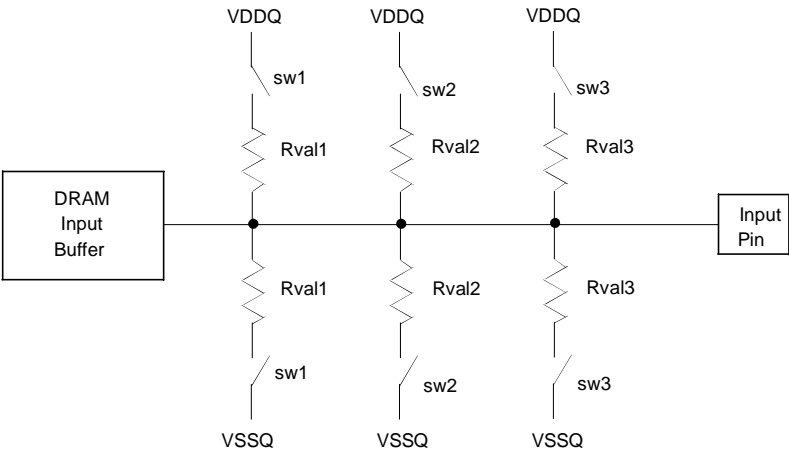
驱动模式, 包括 Drive(1) 和 Drive(0), 被控制器用来测量 DDR2 SDRAM 驱动电阻。在这种模式下, 执行完“进入驱动模式”命令之后, 所有的输出都被逐出 $tOIT$ 。并且在“OCD校准模式结束”命令之后, 所有的输出驱动都被关闭 $tOIT$.具体详情见下图。



ODT (终结电阻)

终结电阻 (ODT) 是 DRAM 的一个特点, 它允许通过 ODT 控制脚对每一个或每一对 DQ, DQS/DQS, RDQS/RDQS, 和 x4/x8 的 DM 打开和关闭电阻, 对于 x16 , 它允许通过 ODT 控制脚对每一个或每一对 DQ, UDQS/UDQS, LDQS/LDQS, LDM 和 UDM 信号。ODT 特性用来提升存储通道的信号完整性, 因为它可以允许 DRAM 控制器独立的打开或关闭所有 DRAM 芯片的终结电阻。
ODT 功能指出 ACTIVE 和 STANDBY 两种模式, 在 SELF REFRESH 模式下不支持关闭功能。.

ODT 的功能



开关 (sw1, sw2, sw3) 由 ODT 脚激活。
开关 sw1, sw2 and sw3 的选择由 EMRS 寄存器中的 “Rtt (nominal)” 决定。
终结电阻包括所有的 DQs, DM, DQS, DQS, RDQS, RDQS 脚

ODT DC 电气特性

参数/条件	Symbol	Min	Nom	Max	Units	Note
Rtt 有效电阻值 for EMRS(A6,A2)=0,1; 75 ohm	Rtt1(eff)	60	75	90	ohm	1
Rtt 有效电阻值for EMRS(A6,A2)=1,0; 150 ohm	Rtt2(eff)	120	150	180	ohm	1
Rtt 上拉和下拉对之间的不匹配容限	Rtt(mis)	-3.75		+3.75	%	1

注意 1 : Rtt 测量的测试条件

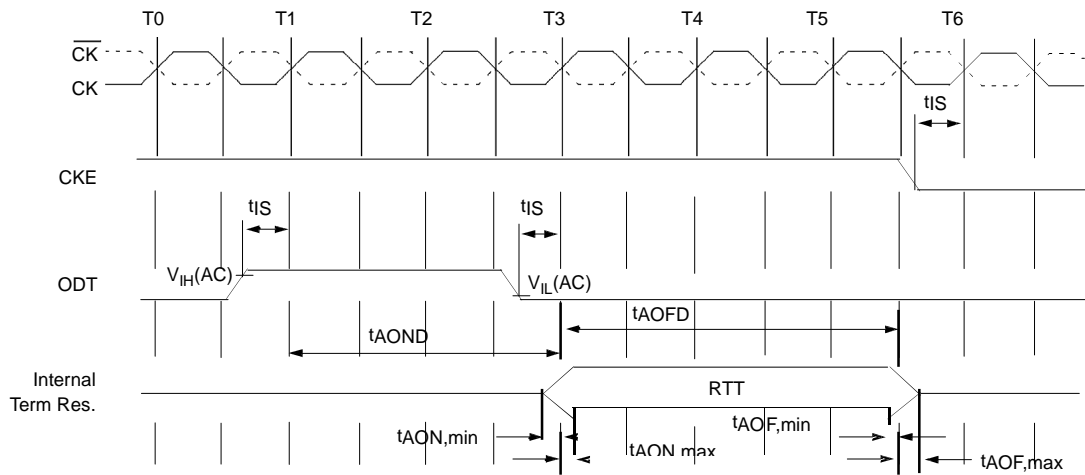
Rtt(eff) 的测试定义: 在测试脚分别测试 V_{IH} (AC) 和 V_{IL} (AC), 然后分别测量电流 I(V_{IH} (AC)) and I(V_{IL} (AC)).
V_{IH} (AC), V_{IL} (AC), 和 VDDQ 的值在 SSTL_18 中定义

$$R_{tt}(eff) = \frac{V_{IH} (AC) - V_{IL} (AC)}{I(V_{IH} (AC)) - I(V_{IL} (AC))}$$

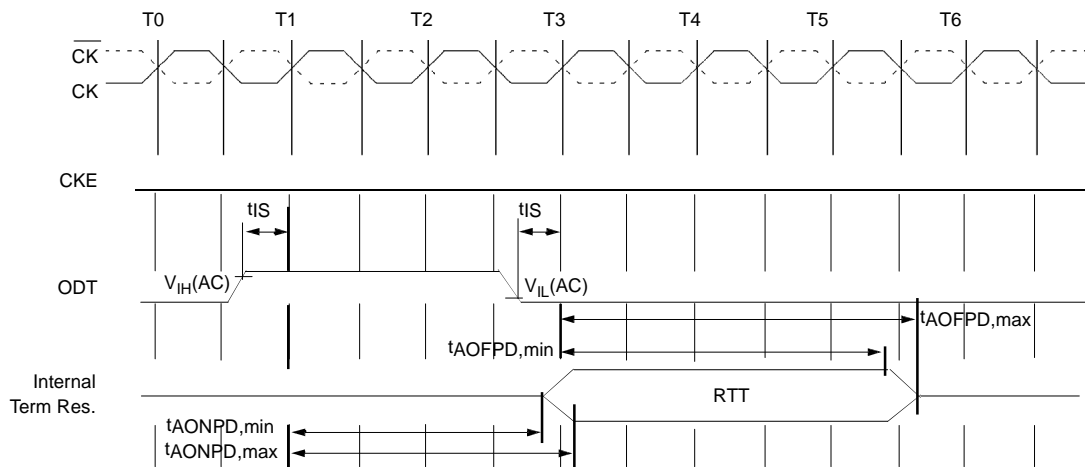
VM 的测试定义: 测试脚空负载是测量其电压 (VM)。

$$\text{delta VM} = \left(\frac{2 \times V_m}{V_{DDQ}} - 1 \right) \times 100\%$$

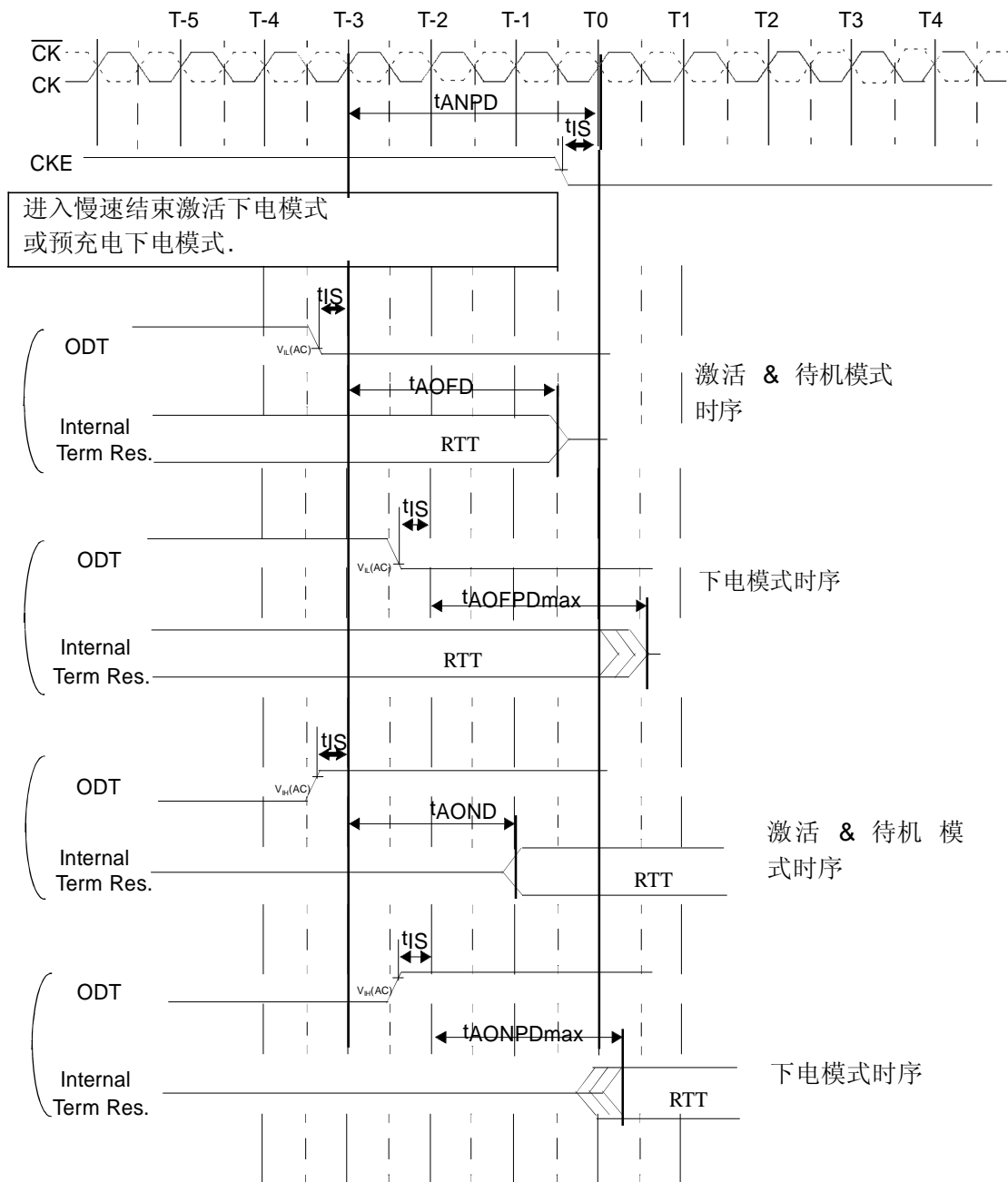
激活/待机模式下的ODT 时序



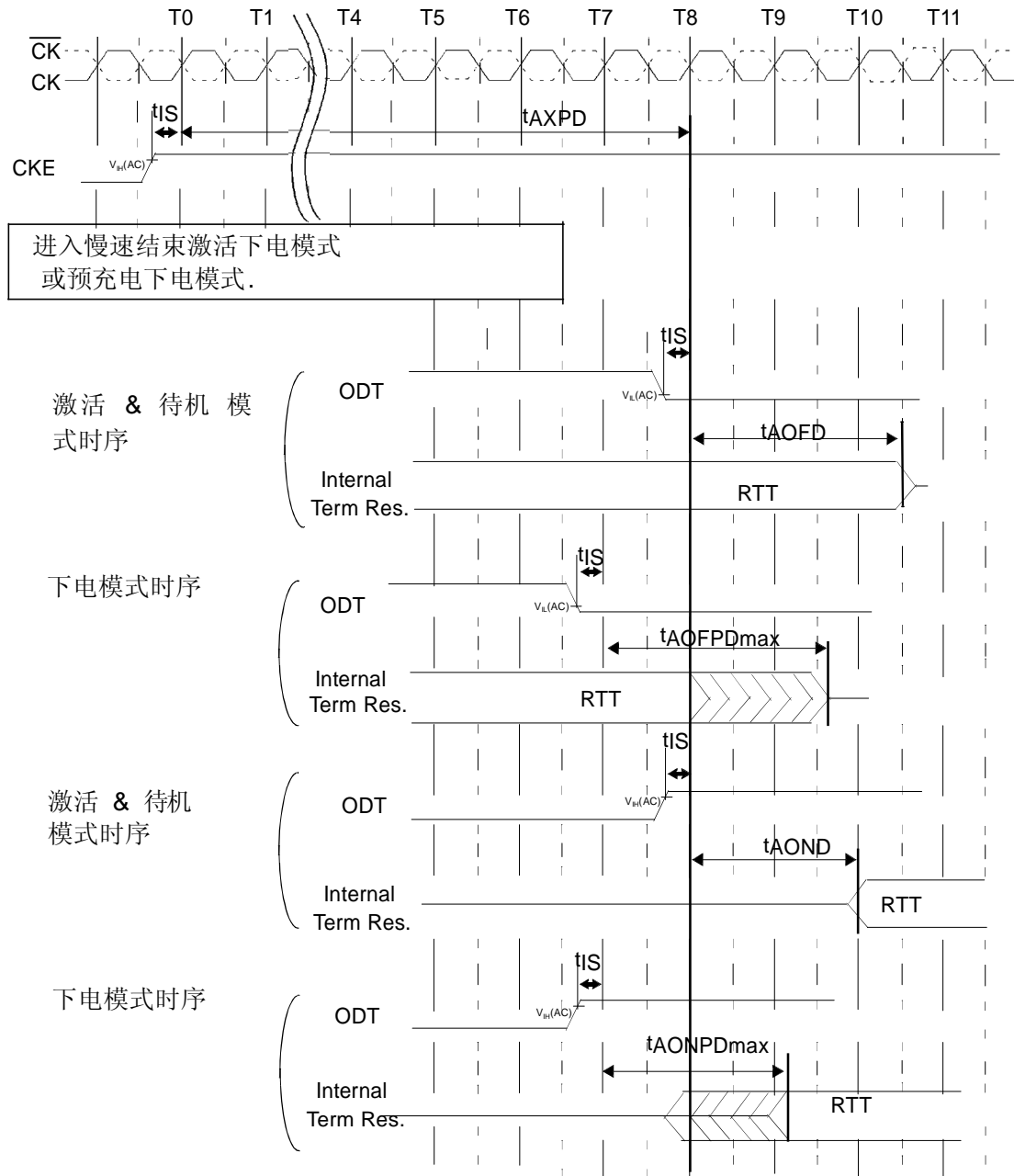
下电模式下的ODT时序



进入下电模式时的ODT 时序模式变法



结束下电模式时的ODT时序模式变化



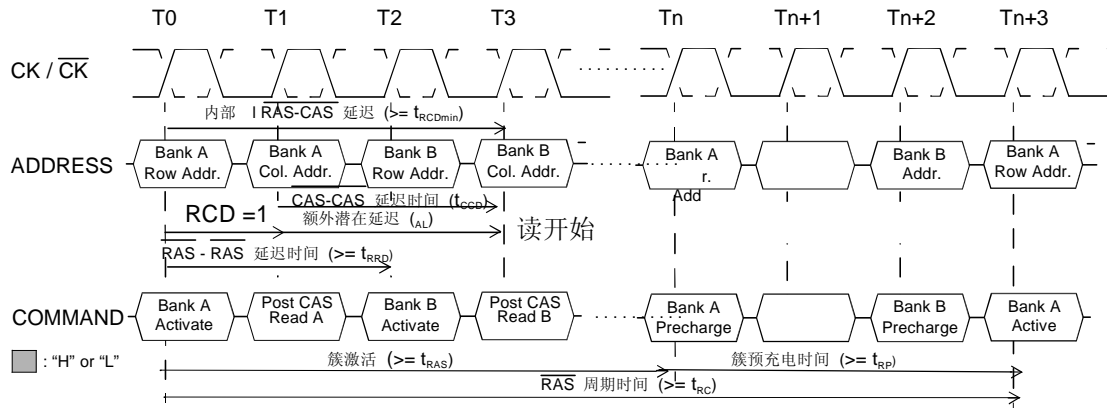
簇激活命令

通过在时钟上升沿时保证 $/CS$ 和 $/WE$ 为高, $/CS$ 和 $/RAS$ 为低即可发布簇激活命令。簇地址 $BA0$ 和 BA 选定希望激活哪一簇。行地址 $A0$ 到 $A13$ 决定要激活该簇的哪些行, 在执行读写命令之前, 必须先执行簇激活命令。在簇激活命令发布之后, DDR2 SDRAM 在下一个时钟周期就可以接受读或者写命令。如果 R/W 命令不能满足 t_{RCD} 最小值的规格, 那么, 在编程时, 一定要在 R/W 命令后加上附加延迟。附加延迟的值必须满足 t_{RCD} 最小值的要求。附加延迟的值可以是 0, 1, 2, 3, 4。一旦某一个簇被激活了, 那么在下一个簇激活命令应用到这一个簇之前, 要先对其预充电。簇激活和预充电时间被分别定义为 t_{RAS} 和 t_{RP} 。

对同一个簇的两个成功的簇激活命令的时间间隔由芯片的 RAS 周期时间来决定 (t_{RC})。T 簇激活命令之间的最小时间间隔是 t_{RRD} 。

为了保证8簇的芯片的瞬时电流不会超过4簇芯片的电流供应能力, 针对8簇的芯片在操作上有一定的限制。有两条规则, 第一条是连续可以发布激活命令条数的显示, 另一个是对于预充电所有簇命令允许更多的 RAS 预充电时间。详细的描述如下:

- 8 簇芯片的连续的簇激活命令的约束: 在一个滚动的 t_{FAW} 窗内, 最多可以激活4个簇。转化为时钟个数就是将 t_{FAW} 除以 t_{CK} , 得到的数取相邻的大的整数。举一个滚动窗的实例, 如果 (t_{FAW}/t_{CK}) 达到10, 并且一个簇激活命令在第 N 个时钟发布, 那么, 在 N 到 $N+9$ 个时钟内, 最多只能发布3个簇激活命令。
- 8簇芯片最大能力预充电: t_{RP} for a Precharge All command for an 8 Bank device will equal to $t_{RP} + 1 * t_{CK}$, where t_{RP} is the value for a single bank pre-charge.

簇激活命令周期: $t_{RCD} = 3$, $AL = 2$, $t_{RP} = 3$, $t_{RRD} = 2$, $t_{CCD} = 2$ 

读写访问模式

簇激活命令之后, 就可以执行读或者写周期了。通过在时钟上升沿设置 $/RAS$ 为高, $/CS$ 和 $/CAS$ 为低, 同时还必须定义 $/WE$ 的值以决定访问动作是读还是写 ($/WE$ 为高表示读操作, $/WE$ 为低表示写操作。)

DDR2 SDRAM 提供了一种快速的列访问操作。单个的读或者写命令会导致在接下来的时钟周期中进行一系列的读或者写操作。突发周期的边界被严格限制到指定页的详细位置。例如, 32Mbit x 4 I/O x 4 Bank 的芯片一页的大小是 2048 位 (由 $CA0-CA9$, $CA11$ 所定义)。每一页的 2048 位依据突发长度被分成 512 或 256 个可寻址的片段, 突发长度为 4 位时是 512 个, 突发长度为 8 位是 256 个。一个 4 位或 8 位的突发访问将对一页的 512 或 256 分之一, 进行完全的访问, 访问的开始地址就是在读或者写命令中的 $CA0-CA9$, $CA11$, 随后的此片段的第二个, 第三个和第四个地址也被访问。突发的顺序由之前的寄存器来定义。

新的突发访问操作不可以中断正在进行的突发长度为 4 的存取操作。然而, 如果正在进行的是突发长度为 8 的操作, 那么在两种情况下, 可以对其中断。第一种情况是, 正在进行的是读操作, 而中断它的也是读操作。第二种情况是, 正在进行的是写操作, 而中断它的是突发长度为 4 的写操作。最小的 CAS 到 CAS 的延迟被定义做 t_{CCD} , 对于读或者写命令来说至少要 2 个时钟。任何的系统或者应用都必须设计为能够随机的对内存芯片进行操作或测试, 如果过多的对某些特定的地址进行读或者写可能会导致缩短产品的使用寿命。

登记 $\overline{\text{CAS}}$

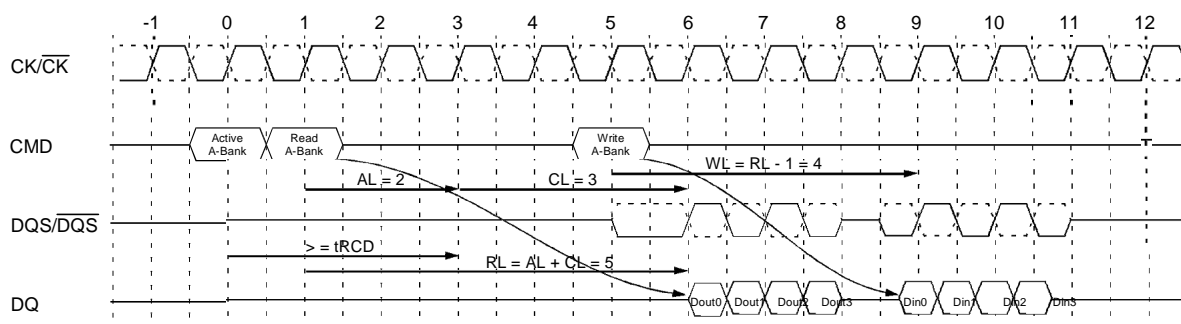
登记 $\overline{\text{CAS}}$ 操作命令可以使DDR2 SDRAM的数据线有效的保持足够的带宽。在此操作中, DDR2 SDRAM 允许在/RAS簇激活命令之后立即发布一个CAS读或者写命令 (或者是在RAS-CAS 延迟时间tRCD周期内的任意时间)。在真正在芯片内部发布之前, 命令要被锁定一个附加延迟时间 (AL)。读延迟(RL) 等于AL和CAS 延迟 (CL) 的和。因此如果用户选择在tRCD最小值之前发布读或者写命令的话, AL(大于0)一定要先写进EMR(1)。写延迟 (WL) 始终都被定义成 $RL - 1$ (读延迟 -1), 这里读延迟被定义为附加延迟和CAS延迟的和($RL=AL+CL$)。使用 AL 的读或者写操作允许无缝的突发 (参考在读突发和写突发的章节的无缝操作时序图的示例)

登记 $\overline{\text{CAS}}$ 操作实例

例 1

针对同一簇读命令后紧跟写命令

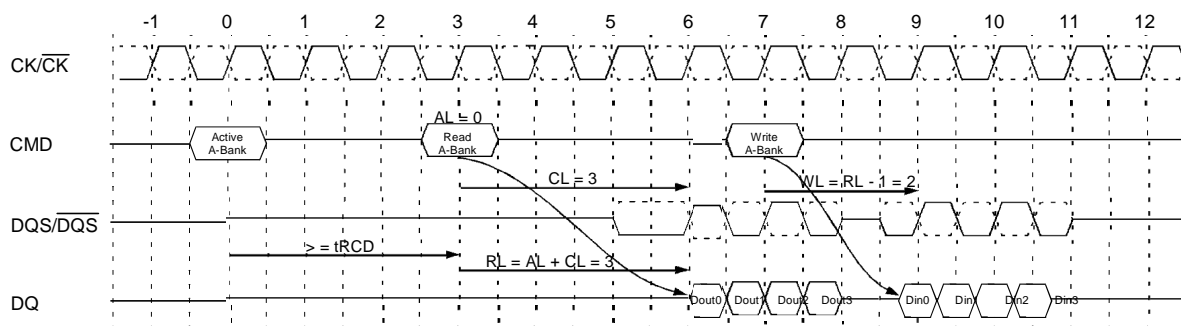
[AL = 2 and CL = 3, RL = (AL + CL) = 5, WL = (RL - 1) = 4, BL = 4]



Example 2

针对同一簇读命令后紧跟写命令

[AL = 0 and CL = 3, RL = (AL + CL) = 3, WL = (RL - 1) = 2, BL = 4]



突发模式操作

突发操作可以向存储器的地址中提供一种恒定的数据流 (写周期), 或者从存储区中得到一种恒定的数据流 (读周期). 寄存器的 参数定义了突发模式包括突发长度和突发顺序。DDR2 SDRAM 支持 4位突发和 8位突发模式。 对于8位突发模式, 支持全间隔地址顺序, 然而, 基于执行的方便, 通常使用连续的地址顺序。突发类型, 或者是间隔的或者是连续的, 由 MRS寄存器的第3位的值决定, 这一点同DDR SDRAM 的操作相似。同样也支持无缝的突发读或者写操作。同DDR器件不同, 在BL=4时, 中断操作是被禁止的。然而当BL = 8 时,在两种情况下读命令可以中断读操作, 写命令可以中断写操作。因此, DDR2 SDRAM并不支持突发中断命令。
突发长度和顺序:

BL = 4

突发长度	开始地址 (A1 A0)	连续地址 (10进制)	间隔地址 (10进制)
4	0 0	0, 1, 2, 3	0, 1, 2, 3
	0 1	1, 2, 3, 0	1, 0, 3, 2
	1 0	2, 3, 0, 1	2, 3, 0, 1
	1 1	3, 0, 1, 2	3, 2, 1, 0

BL = 8

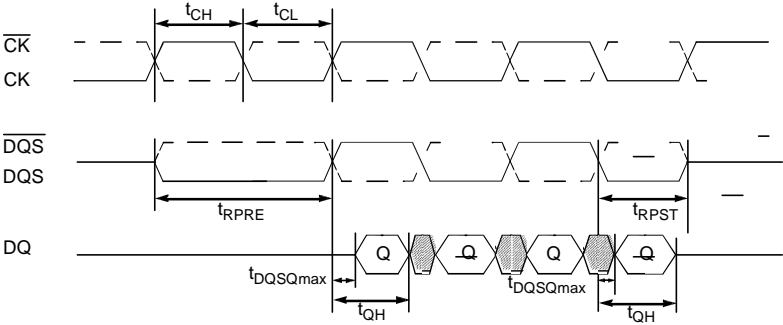
突发长度	开始地址 (A2 A1 A0)	连续地址 (10进制)	间隔地址 (10进制)
8	0 0 0	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7
	0 0 1	1, 2, 3, 0, 5, 6, 7, 4	1, 0, 3, 2, 5, 4, 7, 6
	0 1 0	2, 3, 0, 1, 6, 7, 4, 5	2, 3, 0, 1, 6, 7, 4, 5
	0 1 1	3, 0, 1, 2, 7, 4, 5, 6	3, 2, 1, 0, 7, 6, 5, 4
	1 0 0	4, 5, 6, 7, 0, 1, 2, 3	4, 5, 6, 7, 0, 1, 2, 3
	1 0 1	5, 6, 7, 4, 1, 2, 3, 0	5, 4, 7, 6, 1, 0, 3, 2
	1 1 0	6, 7, 4, 5, 2, 3, 0, 1	6, 7, 4, 5, 2, 3, 0, 1
	1 1 1	7, 4, 5, 6, 3, 0, 1, 2	7, 6, 5, 4, 3, 2, 1, 0

注意:页长度是一种I/O 组织和列地址的功能。

突发读命令

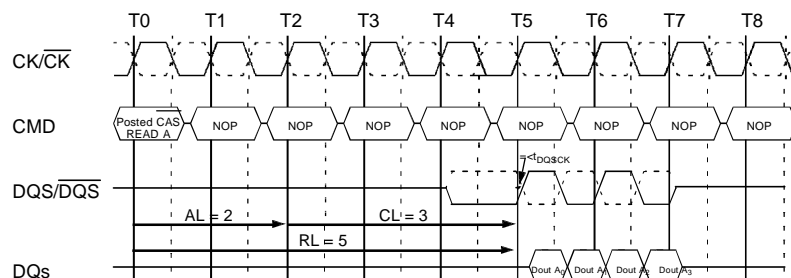
在系统时钟的上升沿, 保持 /CS 和/ CAS 低电平和 /RAS和/ WE 高电平既发布了突发读命令。 同时的地址输入决定了突发操作开始的列地址。从命令开始到第一个数据出现在输出端的延迟时间叫做读延迟 (RL)。在数据脚DQ给出有效数据前一个时钟周期DQS脚被拉低。第一位有效数据将同DQS信号的上升沿同步。以后依次出现的有效数据输出都同DQS脚保持源同步模式。RL等于附加延迟 (AL)和 CAS 延迟 (CL)的和。 CL的值由模式寄存器设定 (MRS), 同 SDR 和 DDR SDRAMs 相似。 AL的值有扩展模式寄存器1设定 (1)(EMRS(1))。

DDR2 SDRAM 管脚时序或者是单端的或者是差分的, 这一点由扩展模式寄存器的EMRS的“激活 DQS” 模式位来决定。
在系统设计时, 差分模式有它的优点。测量DDR2 SDRAM管脚时序的方法根据是单端的还是差分的而不同。在单端模式下, 时序关系在DQS下降或上升到VREF的时候测量。在差分模式下, 时序关系在DQS和/DQS相交的时候测量。这种时序上的差别在设计 and 描述的时候都要考虑到。注意当差分数据选通模式被EMRS禁止的时候, /DQS脚必须连一个20-10K欧姆大小的电阻, 并接地以确保DDR2的正确工作。

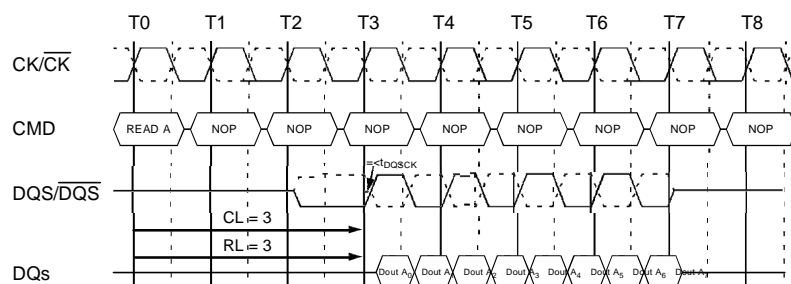


数据输出 (读)时序

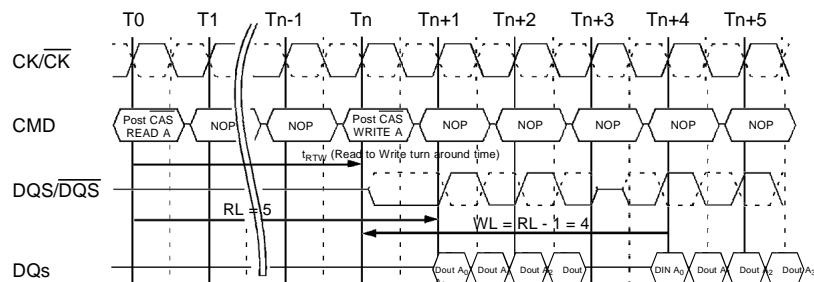
突发读操作: $RL = 5$ ($AL = 2$, $CL = 3$, $BL = 4$)



突发读操作: $RL = 3$ ($AL = 0$ and $CL = 3$, $BL = 8$)

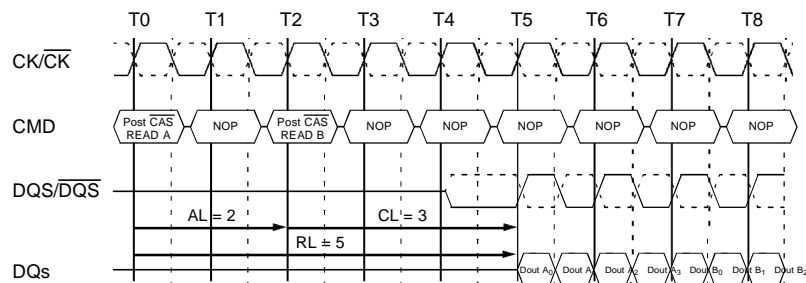


突发读操作，后面跟突发写操作: $RL = 5$, $WL = (RL-1) = 4$, $BL = 4$



从突发读命令到突发写命令的最小时间被定义为读到写的转换时间，在 $BL=4$ 时是4个时钟，在 $BL=8$ 时是6个时钟。

无缝突发读操作: $RL = 5$, $AL = 2$, and $CL = 3$, $BL=4$

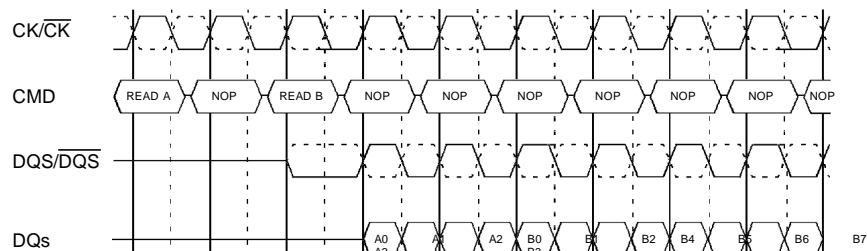


无缝的突发读操作可以通过在每隔一个时钟 ($BL=4$) 或两个时钟 ($BL=8$) 发布一个读命令来实现。只要簇被激活，这种操作允许在同一簇内或不同的簇之间动作。

读命令中断读操作

突发的读操作只可以被另一个突发长度为4位的突发读命令中断。任何其它的情况都不可以被中断。

突发读中断时序示例: (CL=3, AL=0, RL=3, BL=8)



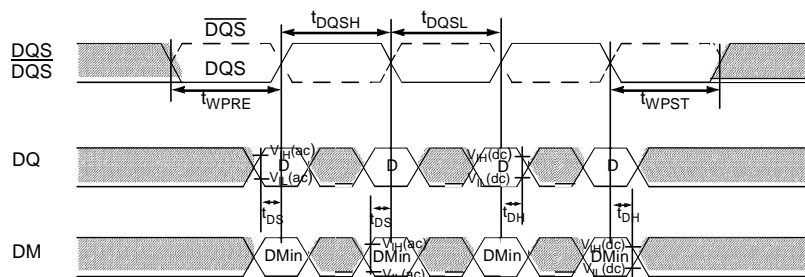
注意:

1. 读突发中断只是在突发长度为8时允许，突发长度为4时是禁止的。
2. 长度为8个读突发操作只可以被另一个读命令所中断。读突发是不可以被写命令或者预充电命令所中断的。
3. 读突发中断只可以发生在从前的读命令发布了2个时钟周期之后。任何其它的时序都是错误的。
4. 读突发中断可以发生在 DRAM内的任何簇内。
5. 带自动预充电的读突发操作是不可以被中断的。
6. 读突发操作可以被另一个在自动预充电的读突发操作所中断。
7. 突发长度由模式寄存器来设定。他们不一定就是实际的情况。例如，最小的读到预充电的时间是 $AL + BL/2$ ，这里BL是在模式寄存器中设定的突发长度而不是实际的突发长度(实际的短，因为被中断了)。

突发写操作

通过在时钟上升沿时保持 $/CS$ 、 $/CAS$ 和 $/WE$ 为低并且 $/RAS$ 为高可以发布突发写命令。同时的地址输入决定了突发操作开始的列地址。写延迟 (WL) 被定义为读延迟 (RL) 减去1，等于 $(AL + CL - 1)$; 也就是从命令发布到第一个 DQS 上升沿的时钟数目。在数据脚 DQ 给出有效数据前一个时钟周期 DQS 脚被拉低。tDQSS 的大小必须满足 DQS 的每一个上升沿同时钟周期的上升沿相关联。随后的数据都在 DQS 上升沿采样，知道达到突发的长度 (4 或者 8)。当突发结束之后，任何加到 DQ 脚上的数据都将被忽略。突发操作结束之后的所有 DQ 信号都被忽略。从完成突发写操作到下一次的簇预充电命令被称做写恢复时间 (WR)。

DDR2 SDRAM 管脚时序或者是单端的或者是差分的，这一点由扩展模式寄存器的 EMRS 的“激活 DQS”模式位来决定。在系统设计时，差分模式有它的优点。测量 DDR2 SDRAM 管脚时序的方法根据是单端的还是差分的而不同。在单端模式下，时序关系在 DQS 下降或上升到 VREF 的时候测量。在差分模式下，时序关系在 DQS 和 \overline{DQS} 相交的时候测量。这种时序上的差别在设计 and 描述的时候都要考虑到。注意当差分数据选通模式被 EMRS 禁止的时候， $/DQS$ 脚必须连一个 20-10K 欧姆大小的电阻，并接地以确保 DDR2 的正确工作。



数据输入(写) 时序

The diagram illustrates the timing for a burst write operation across clock cycles T0 to Tn. It shows the relationship between the clock (CK/CK̄), command (CMD), data strobe (DQS), and data bus (DQs).

Case 1: with tDQSS(max)

- CK/CK̄:** A periodic clock signal.
- CMD:** A sequence of commands: "Posted CAS WRITE A" at T0, followed by "NOP" from T1 to T6, and "Precharge" at Tn.
- DQS/DQS̄:** A strobe signal that starts at T4. The pulse width is labeled $W_L = R_L - 1 = 4$. The period of the strobe is labeled $\geq W_R$.
- DQs:** Data signals are shown as hexagonal pulses starting at T4. The first data signal is labeled $DIN A_0$ and $DIN A_3$. The time from the start of the strobe to the first data signal is t_{DQSS} . The time from the start of the strobe to the last data signal is t_{DS} .

Case 2: with tDQSS(min)

- CK/CK̄:** A periodic clock signal.
- CMD:** A sequence of commands: "Posted CAS WRITE A" at T0, followed by "NOP" from T1 to T6, and "Precharge" at Tn.
- DQS/DQS̄:** A strobe signal that starts at T4. The pulse width is labeled $W_L = R_L - 1 = 4$. The period of the strobe is labeled $\geq W_R$.
- DQs:** Data signals are shown as hexagonal pulses starting at T4. The first data signal is labeled $DIN A_0$ and $DIN A_3$. The time from the start of the strobe to the first data signal is t_{DQSS} . The time from the start of the strobe to the last data signal is t_{DS} .

Timing diagram for a Read operation. The diagram shows signals CK/CK-bar, CMD, DQS/DQS-bar, and DQ over clock cycles T0 to T9. Key parameters shown include Write to Read delay (CL-1 - BL/2 - tWTR), NOP commands, Post CAS READ A, WL = RL - 1 = 4, AL = 2, CL = 3, RL = 5, and tWTR = 1/2.

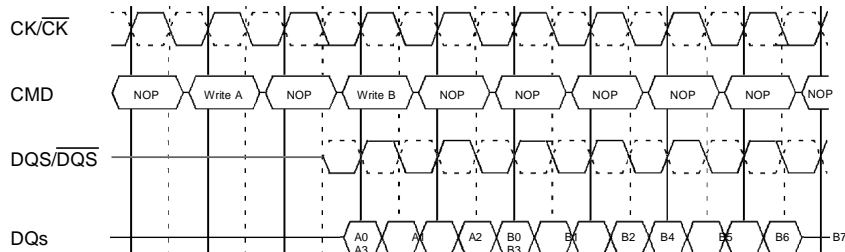
Timing diagram for a 16-bit burst read operation. The diagram shows signals CK/CK-bar, CMD, DQS/DQS-bar, and DQs over time intervals T0 to T8. CK/CK-bar is a periodic clock. CMD shows 'Post CAS Write A' at T0, 'NOP' at T1, 'Post CAS Write B' at T2, and 'NOP' from T3 to T8. DQS/DQS-bar shows a pulse at T1 and T2, then remains high. DQs shows data being sampled at T3 (DIN A0), T4 (DIN A1), T5 (DIN A2), T6 (DIN A3), T7 (DIN B0), and T8 (DIN B1). A note indicates WL = RL - 1 = 4.



写操作被写命令打断

突发的写操作只可以被另一个突发长度为4位的突发写命令中断。任何其它的情况都不可以被中断。

写突发中断实例: (CL=3, AL=0, RL=3, WL=2, BL=8)



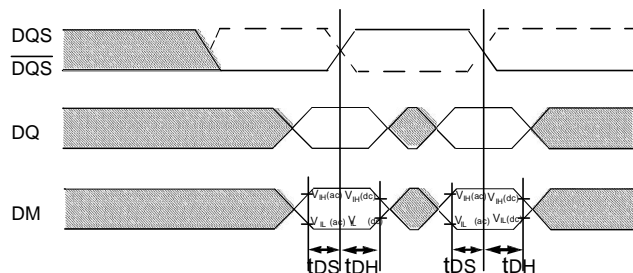
注意:

1. 写突发中断只是在突发长度为8时允许，突发长度为4时是禁止的。
2. 长度为8个写突发操作只可以被另一个写命令所中断。写突发是不可以被读写令或者预充电命令所中断的。
3. 写突发中断只可以发生在从前的写命令发布了2个时钟周期之后。任何其它的时序都是错误的。
4. 写突发中断可以发生在 DRAM内的任何簇内。
5. 带自动预充电的写突发操作是不可以被中断的。
6. 写突发操作可以被另一个在自动预充电的写突发操作所中断。
7. 突发长度由模式寄存器来设定。他们不一定是实际的情况。例如，最小的写到预充电的时间是 $AL + BL/2 + WR$ ，这里BL是在模式寄存器中设定的突发长度而不是实际的突发长度(实际的短，因为被中断了)。

写数据隐藏

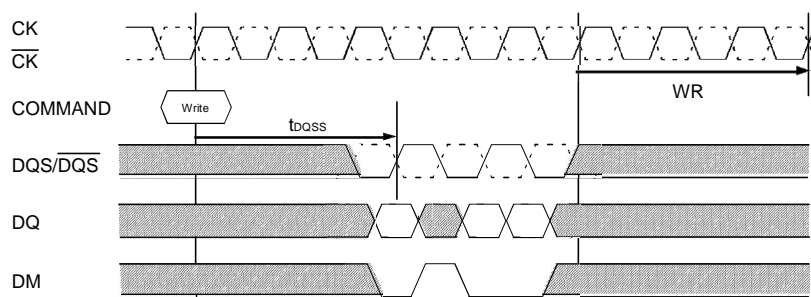
在DDR2 SDRAM中，每8个DQ都有一个写数据隐藏脚（DM），同DDR SDRAM的情况一致。他同数据位有同一的操作时序。虽然是在双向行为中使用，它内部下载数据确保系统时序。在读周期时，x4和x16位组织时，并不使用DM功能。因此，x8的DM位组织可以通过EMRS(1)设定在读周期时使用RDQS。

数据隐藏时序

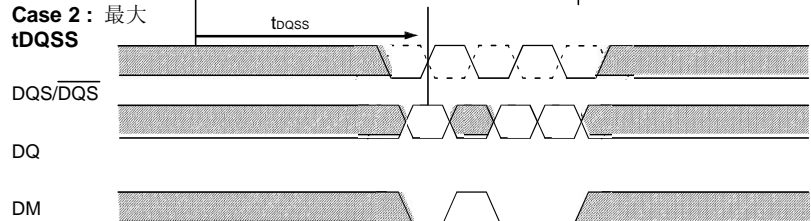


数据隐藏功能：WL=3, AL=0, BL=4 显示

Case 1：最小 tDQSS



Case 2：最大 tDQSS



预充电操作

可以通过激活预充电命令对某一簇进行预充电或者关闭某一簇。在时钟上升沿时保持 /CS、/RAS 和 /WE 为低并且 /WE 为高就可以发布预充电命令。预充电命令可以对某一个簇操作，也可以对所有簇同时操作。在发布预充电命令时，256Mb 和 512Mb 的 3 个地址位 A10、BA0 和 BA1 或者 1Gb 大小以上的器件的地址位 A10、BA0 ~ BA2 被用于定义对哪一个簇预充电。对于 8 簇器件，参考簇激活的章节。

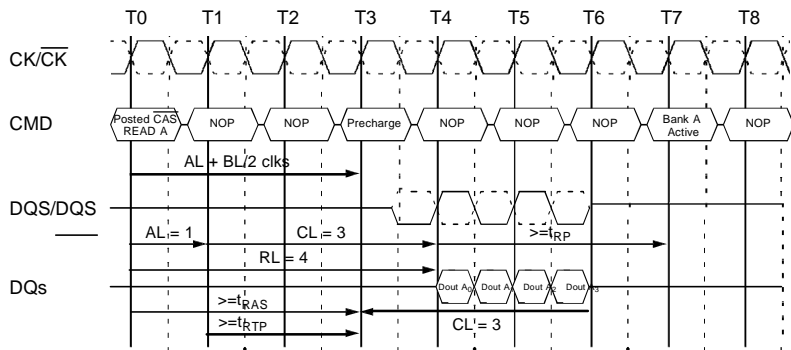
通过地址位选择预充电的簇

A10	BA2	BA1	BA0	Precharged Bank(s)	Remarks
低	低	低	低	只是0簇	
低	低	低	高	只是1簇	
低	低	高	低	只是2簇	
低	低	高	高	只是3簇	
低	高	低	低	只是4簇	1 Gb 或更大
低	高	低	高	只是5簇	1 Gb或更大
低	高	高	低	只是6簇	1 Gb或更大
低	高	高	高	只是7簇	1 Gb或更大
高	不关心	不关心	不关心	所有簇	

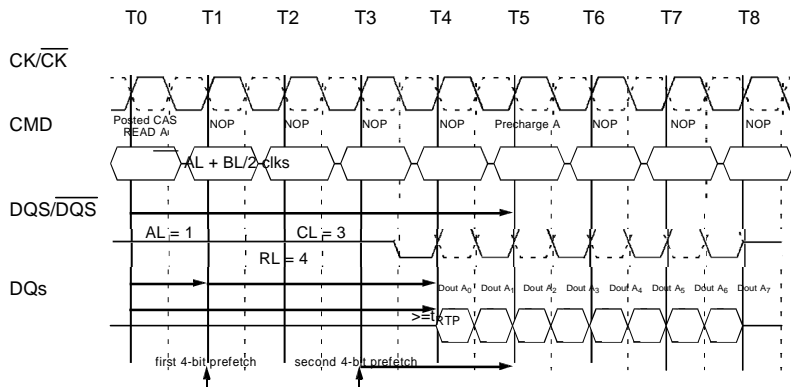
突发读操作后跟预充电命令

针对同一个簇从读命令到预充电命令的最小时间 = $AL + BL/2 + \max(RTP, 2) - 2$ 个时钟。
为了尽快的预充电，预充电命令在读命令发布之后的“附加延迟 (AL) + BL/2 个时钟”的上升沿发布。针对同一簇的簇激活命令可以在预充电之后 (t_{RP}) 长时间发布。直到 t_{RAS} 条件满足才可以发布预充电命令。
最小的读到预充电必须满足一个从开始最新的 4 位读命令的预取到下一次预充电的最小的模拟时间，对于 $BL = 4$ 这就是实际读 (读命令之后的 AL) 到预充电命令的时间。对于 $BL = 8$ 来说，这就是读命令之后的 AL + 2 个时钟到预充电命令。

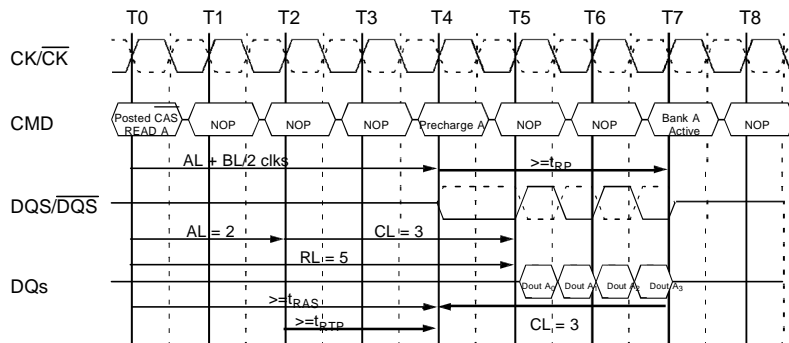
例 1 :突发读操作后面紧跟预充电：
RL = 4, AL = 1, CL = 3, BL = 4, $t_{RTP} \leq 2$ clocks



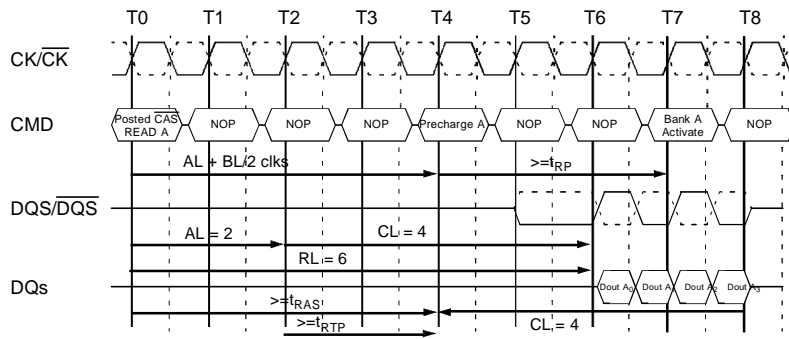
例 2 : 突发读操作后面紧跟预充电：
RL = 4, AL = 1, CL = 3, BL = 8, $t_{RTP} \leq 2$ clocks



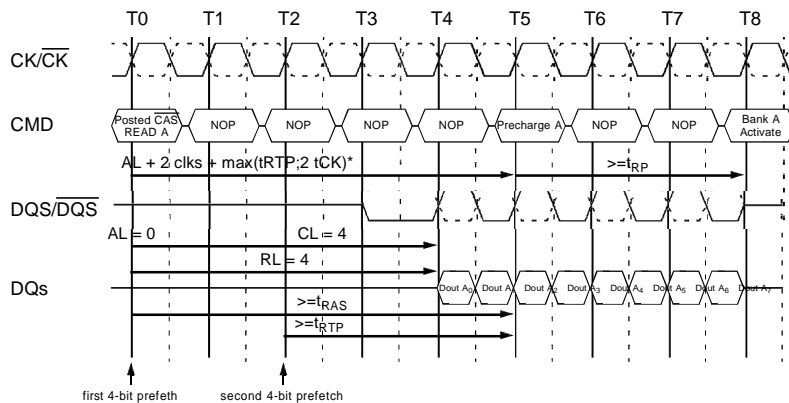
例 3:突发读操作紧跟着预充电: $RL = 5$, $AL = 2$, $CL = 3$, $BL = 4$,
 $t_{RTP} \leq 2$ 个时钟



例 4: 突发读操作紧跟着预充电:
 $RL = 6$, $AL = 2$, $CL = 4$, $BL = 4$, $t_{RTP} \leq 2$ clocks



例 5: 突发读操作紧跟着预充电:
 $RL = 4$, $AL = 0$, $CL = 4$, $BL = 8$, $t_{RTP} > 2$ 个时钟

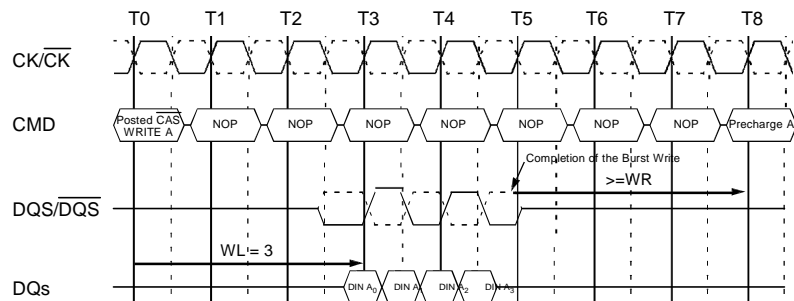


*: 贴近大一点的整数.

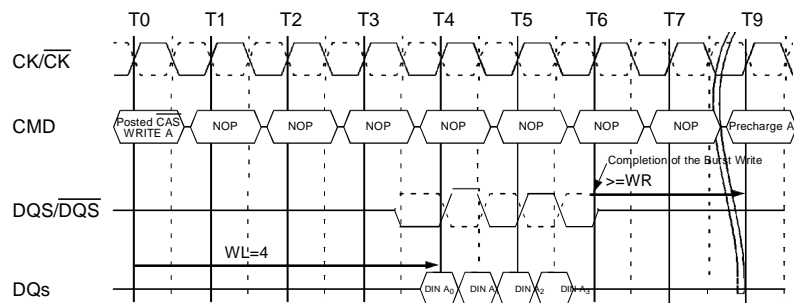
突发写后面紧跟预充电

从写命令到对同一簇的预充电命令最小的时间 = $WL + BL/2 \text{ cks} + WR$ ，对于写周期来说，必须满足从上一个写命令到能够发布预充电命令的延迟。这个延迟被称为写恢复时间（WR），是从上一个写命令到能够发布预充电命令的延迟。在WR延迟之前不可以发布预充电命令。

例 1：突发写后面跟预充电：WL = (RL-1) = 3



例 2：突发写后面跟预充电：WL = (RL-1) = 4



自动预充电操作

在对一个处于激活状态的簇的新的行打开之前，这个处于激活状态的簇必须被预充电，可以用预充电命令或自动预充电命令。当对 DDR2 SDRAM 进行读或者写操作之前，CAS 时序接受附加的列地址 A10，以允许在突发读或者写周期时，被激活的簇可以自动的尽快进行预充电。如果在读写命令发布的时候，A10 是低，执行通常的读写操作，并且在完成突发操作后，那一簇依然保持激活状态。如果在读写命令发布的时候，A10 是高，那么自动预充电功能就被激活了。在自动预充电过程中，读命令被照常执行，只是处于激活状态的行在突发读结束之前的 CL 个时钟的上升沿就开始预充电。在写命令时同样可以执行自动预充电操作。在突发写操作的最后一个数据正确的存入存储器之前，不会执行自动预充电命令的充电操作。这种特点允许在突发读操作时部分的或者全部的隐藏预充电操作（依赖于 CAS 延迟）。这样，提升了随机数据访问时的系统性能。RAS 保持电路了内部的延迟直到阵列恢复操作完成 (tRAS 得到满足)，因此自动预充电操作可以同任何的读或者写命令一起发布。

带自动预充电的突发读命令

如果在发布读命令时，A10 是高电平，那么一个带自动预充电功能的读命令就开始了。如果 tRAS 和 tRTP 满足条件，那么 DDR2 SDRAM 将在带自动预充电功能的读命令之后的 AL-BL/2 个周期的上升沿开始自动预充电操作。

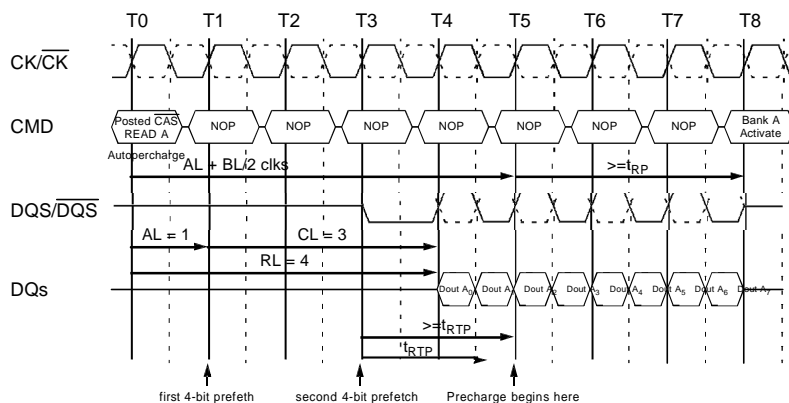
如果 tRAS(min) 在时钟边沿不满足，那么自动预充电命令将延后直到 tRAS(min) 条件得到满足时开始。

如果 tRTP(min) 在时钟边沿不满足，那么自动预充电命令将延后直到 tRTP(min) 条件得到满足时开始。尽管内部预充电由 tRTP 所限制，tRP 开始于内部预充电发生的时候 (并不是那一时候的下一个时钟上升沿)。因此对于 BL = 4 从自动预充电的读到下一个激活命令的最少时间是 $AL + (tRTP + tRP)^*$ (参见例2)，对于 BL = 8 从自动预充电的读到下一个激活命令的最少时间是 $AL + 2 + (tRTP + tRP)^*$ ，这里 “*” 的意思是：“比自己大的邻近的整数”。在任何情况下，内部预充电操作不会早于最新的 4 位数据预存取之前的两个周期。

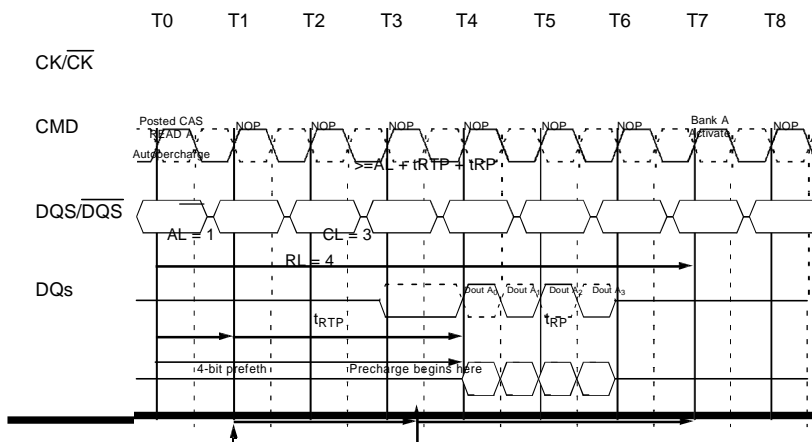
如果下面两个条件同时满足，那么针对同一个簇的簇即命令就可以发布了。(1) 在自动预充电开始的时候，RAS 预充电时间 (tRP) 被满足。

(2) 从之前的簇激活的 RAS 周期时间 (tRC) 条件得到满足。

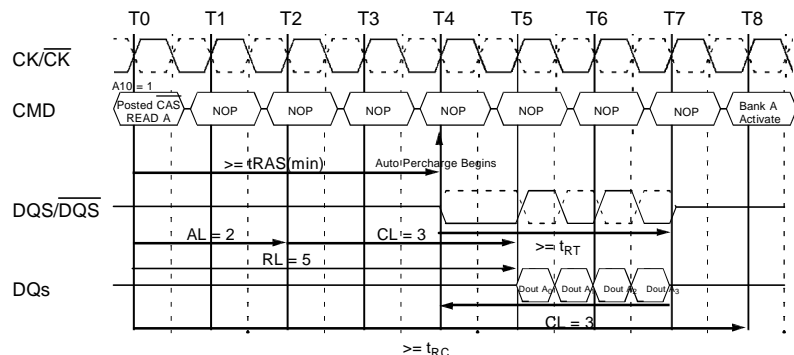
例 1: 带自动预充电的突发读操作: RL = 4, AL = 1, CL = 3, BL = 8, tRTP ≤ 2 个时钟



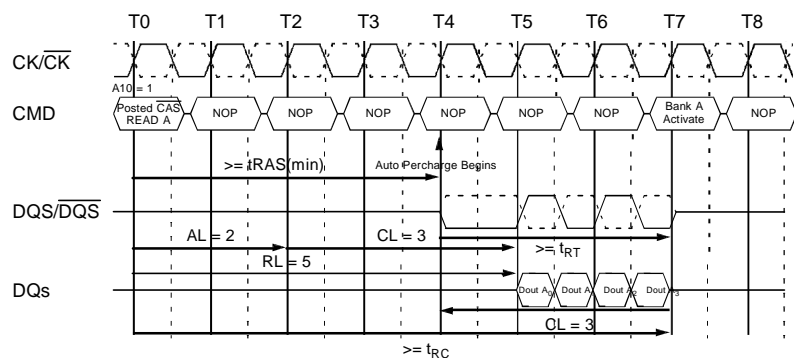
例 2: 带自动预充电的突发读操作: RL = 4, AL = 1, CL = 3, BL = 4, tRTP > 2 个时钟



例 3: 在自动预充电的突发读操作，后面跟针对同一簇的簇激活命令 (**tRC Limit**): $RL = 5$ ($AL = 2$, $CL = 3$, internal $tRCD = 3$, $BL = 4$, $t_{RTP} \leq 2$ clocks)



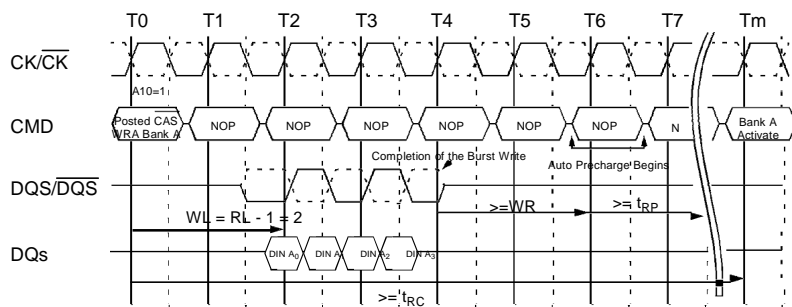
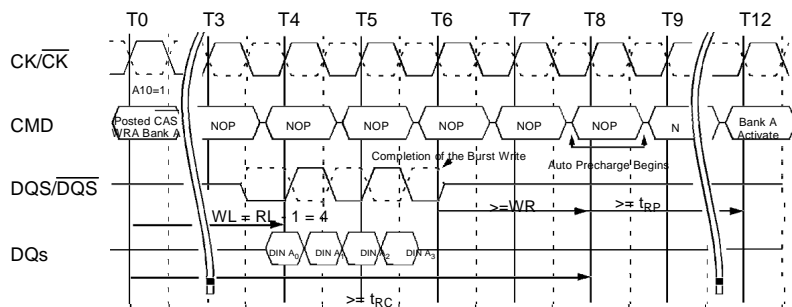
例 4: 在自动预充电的突发读操作，后面跟针对同一簇的簇激活命令 (**tRP Limit**): $RL = 5$ ($AL = 2$, $CL = 3$, internal $tRCD = 3$, $BL = 4$, $t_{RTP} \leq 2$ clocks)



带自动预充电的突发写

当发布写命令时，如果A10是高电平，那么就是执行了带自动预充电的突发写操作。DDR2 SDRAM 在突发写操作完成之后的写恢复时间满足之后就开始自动预充电操作。如果下面两个条件满足，那么在写周期完成之后进行自动预充电操作。

- (1) 数据输入到簇的激活延迟时间 ($WR + tRP$) 满足条件。
- (2) 之前簇的激活的/RAS 周期时间 (tRC) 条件被满足。

带自动预充电的突发写操作 (tRC Limit): $WL = 2, WR = 2, tRP = 3, BL = 4$ 带自动预充电的突发写操作 ($WR + tRP$): $WL = 4, WR = 2, tRP = 3, BL = 4$ 

预充电 & 自动预充电净化

From Command	To Command	Minimum Delay between "From Command" to "To Command"	Unit	Notes
读	预充电 (同读命令作用于相同的簇)	$AL + BL/2 + \max(RTP, 2) - 2$	clks	1, 2
	预充电所有簇	$AL + BL/2 + \max(RTP, 2) - 2$	clks	1, 2
带自动预充电的读	预充电(同读命令作用于相同的簇)	$AL + BL/2 + \max(RTP, 2) - 2$	clks	1, 2
	预充电所有簇	$AL + BL/2 + \max(RTP, 2) - 2$	clks	1, 2
写	预充电 (同写命令作用于相同的簇)	$WL + BL/2 + WR$	clks	2
	预充电所有簇	$WL + BL/2 + WR$	clks	2
带自动预充电的写	预充电 (同写命令作用于相同的簇)	$WL + BL/2 + WR$	clks	2
	预充电所有簇	$WL + BL/2 + WR$	clks	2
预充电	预充电(同前一个预充电命令作用于相同的簇)	1	clks	2
	预充电所有簇	1	clks	2
预充电所有簇	预充电	1	clks	2
	预充电所有簇	1	clks	2

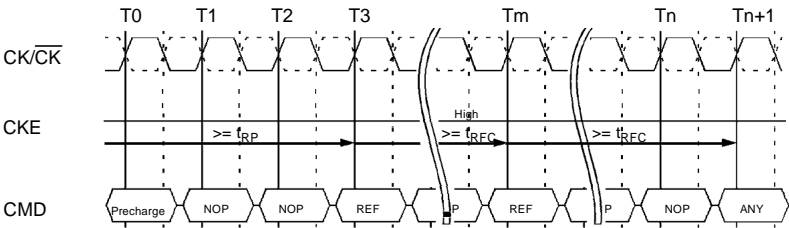
- 注意：
- 1. $RTP[cycles] = RU\{tRTP(ns)/tCK(ns)\}$, 这里 RU 支持向邻近的大的整数靠拢.
 - 2. 对于一个给定的簇，预充电周期应该从最近的预充电命令开始算起，或者是预充电某一簇或者是预充电所有簇，最近的发布到某一簇上的预充电命令之后要满足 tRP 或者 tRP 的条件。

刷新命令

当在时钟上升沿时保持 $/CS/$ RAS 和 $/CAS$ 为低并且 $/WE$ 为高，那么芯片就进入了刷新模式 (REF).在执行刷新模式前所有的簇都要被预充电并且至少要保持 (tRP) 的空闲状态。芯片内部的地址计数器在刷新周期中提供簇地址。一旦命令周期开始，则不需要任何的外部地址线。

当刷新周期结束以后，DDR2 SDRAM 的所有簇都处于空闲状态。一个刷新命令到下一个激活命令或刷新命令的延迟必须大于或者等于刷新周期时间 ($tRFC$)。

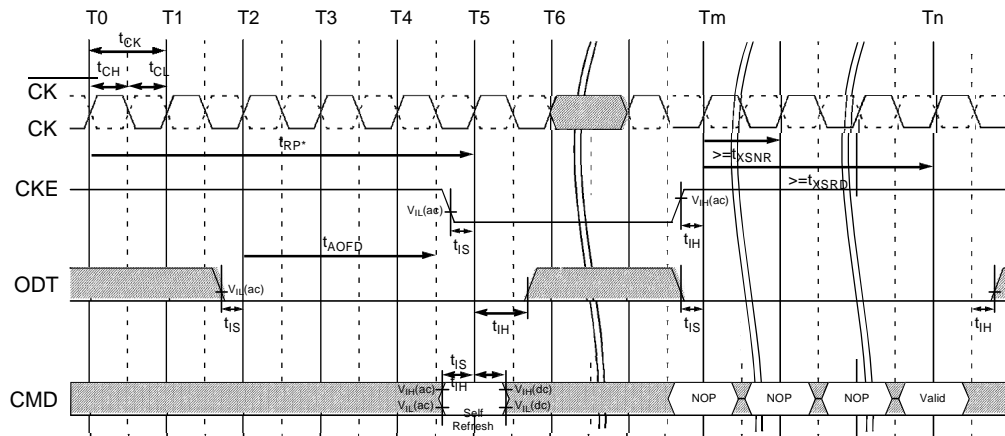
为了提升在时序安排和多任务切换是的效率，在完全刷新的间隔提供了一些灵活的选择。对于任何给定的DDR2 SDRAM,最多可以连续发布8次刷新命令，也就是说，任何的刷新命令和接下来的刷新命令的最大时间间隔是 $9 * tREFI$ 。



自我刷新操作

当系统其余部分断电时，自刷新指令是用来维持数据的。处于自刷新模式，DDR2无需外部时钟就可维持数据。DDR2使用内建计数器进行自刷新超作。在CS,RAS,CAS与CKE为0，WE为高时，自刷新指令在时钟上沿写入。在自刷新指令执行前ODT要关闭，将ODT脚拉低或使用EMRS指令。一旦指令进入寄存器，CKE要拉低以使器件保持在自刷新模式。进入自刷新模式后DLL自动禁用，推出后自动打开。DDR2进入自刷新模式后所有的外部信号（除了CKE）都被忽略。所有电源脚(VDD, VDDQ, VDDL and Vref)必须供电以保证自刷新超作正确进行。进入自刷新模式后，DRAM在TCKE时期内会激活几条内部刷新指令。为了省电，时钟也会被禁止。DDR2维持自刷新的最小时间是TCKE。自刷新入口被存入寄存器，用户可以改变外部时钟频率或停止一个外部时钟周期。但在退出自刷新模式前必须恢复时钟。

退出自刷新模式需要一系列指令。首先，在CKE为高前时钟必须稳定。一旦自刷新出口进入被寄存，在有效指令允许器件内部刷新以前，至少要满足tXSNR的延时。在整个自刷新出口期间TXSRD，CKE必须为高，除非重新进入自刷新。一旦退出自刷新，在等待至少TXSNR后，并写入一条刷新指令（TRFC刷新时间）后，DDR2才能再次进入自刷新。在自刷新出口期间TXSNR，NOP或去选择指令必须在上沿写入。在自刷新期ODT因该关闭。出了自刷新后，想要再次进入，DDR2需要至少需要一条的自刷新指令。



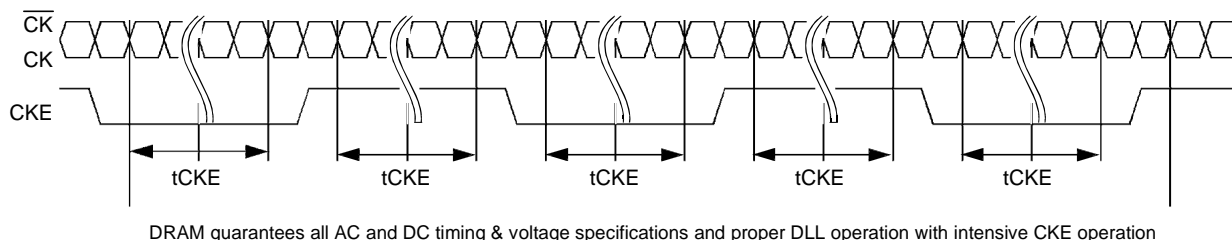
- 进入自刷新状态前，芯片必须处于所有块空闲状态。
- 在进入自刷新模式前，ODT要被关闭TAFD，当TXSRD时序满足时才能被打开。
- tXSRD是针对读或者预冲读指令。
- tXSNR是针对除了读或者预冲读指令外的其他指令的。

下电

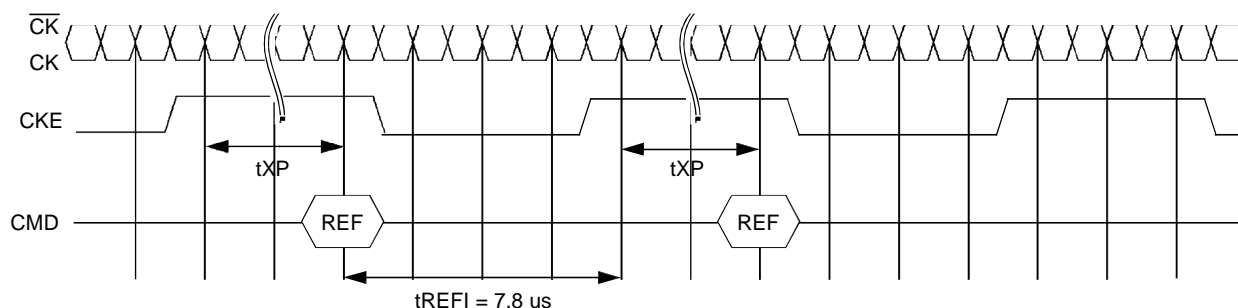
CKE为低时（随NOP或DESELET指令），同时进入下电模式。当执行模式寄存器或扩展寄存器指令，或读写超作时，CKE不能为低。其他超作时，如行激活，预冲电，自充或者自刷新，CKE可以为低；但是直到这些操作结束后，才能施加下电IDD spec。详细情况在时序图中显示。

进入下电模式后，DLL应处于锁定状态。否则在退出下电模式时DLL应该被重启。只要DRAM控制器与DRAM规范兼容，DRAM设计保证所有AC，DC时序和电压要求以及正确的DLL操作伴随CKE精确操作。下图显示两个CKE应用。

<Example of CKE intensive environment 1>



<Example of CKE intensive enviroment 2>

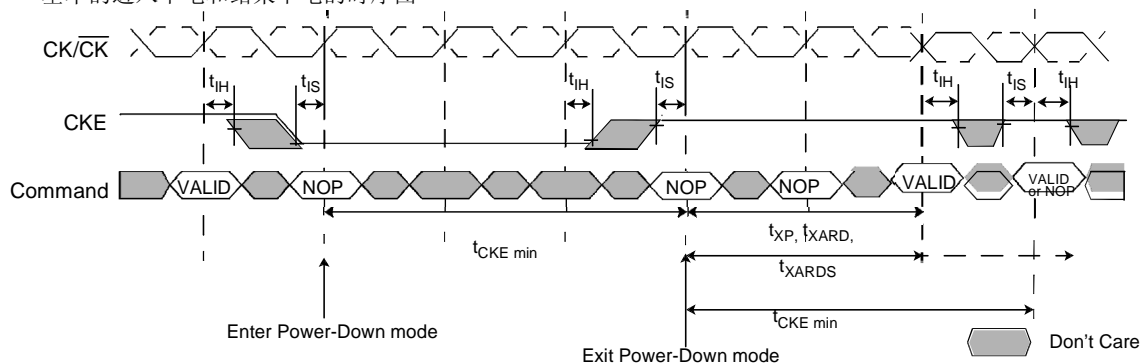


上图可以在很长时间里重复。DRAM保证所有AC，DC时序和电压要求以及正确的DLL操作在温度和电压漂移特性。

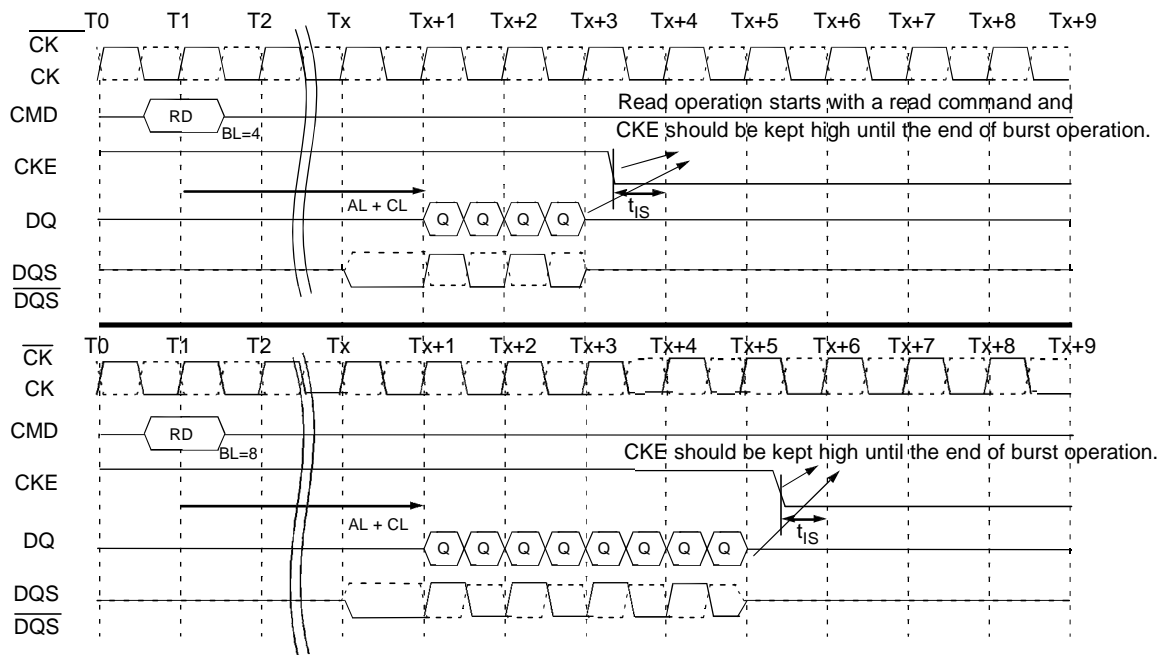
当所有的块空闲时,如果进入下电,这时是指预冲下电;如果是某一行被选中时,这是指选中(active)下电.下电模式禁用输入输出缓冲,但CK,-CK,ODT,CKE除外。一旦进入预冲下电或慢退出选中下电模式时DLL被禁用。但是在快速退出选中下电模式时,DLL是开启状态。下电模式时,CKE为低,时钟必须维持稳定,ODT有效,但所有的输入信号是“无关”,CKE须为低直到TCKE满足要求。下电时期必须在 $9 \times \text{TREFI}$ 内。

CKE为高时（伴随NOP,DESELET指令），退出下电状态,CKE须为高直到TCKE满足要求。CKE为高后,就可以参考下电退出时序（TXP,TXARD,TXARDS),施加有效的指令。下电退出时序参考AC特性表。

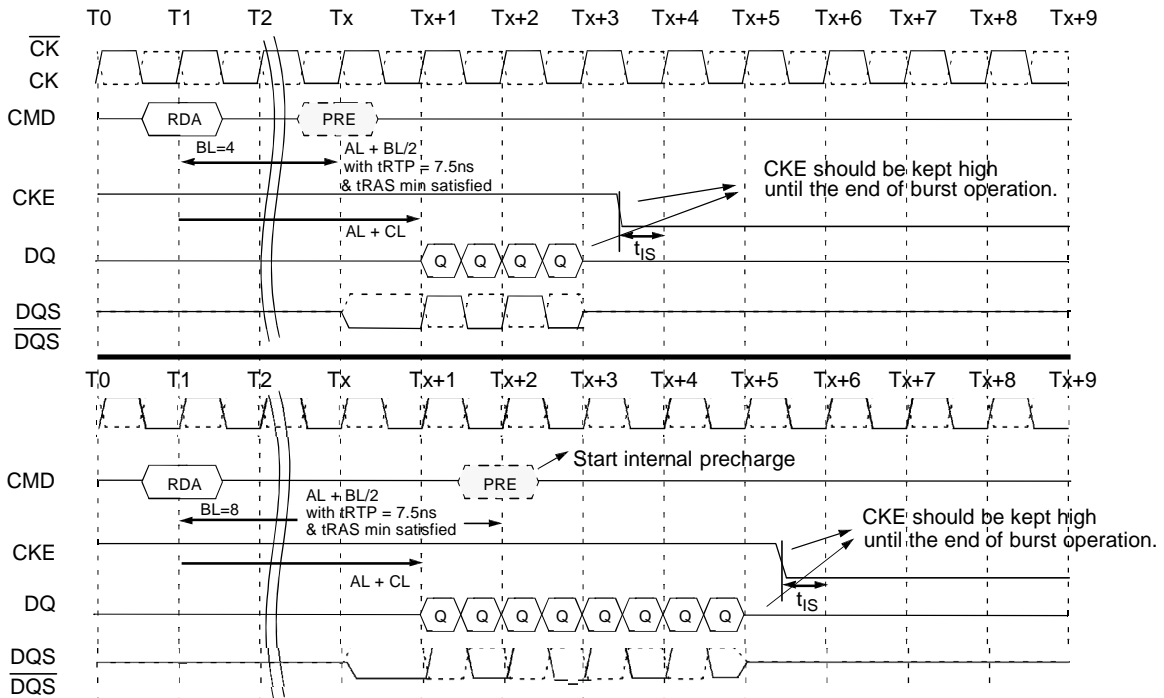
基本的进入下电和结束下电的时序图



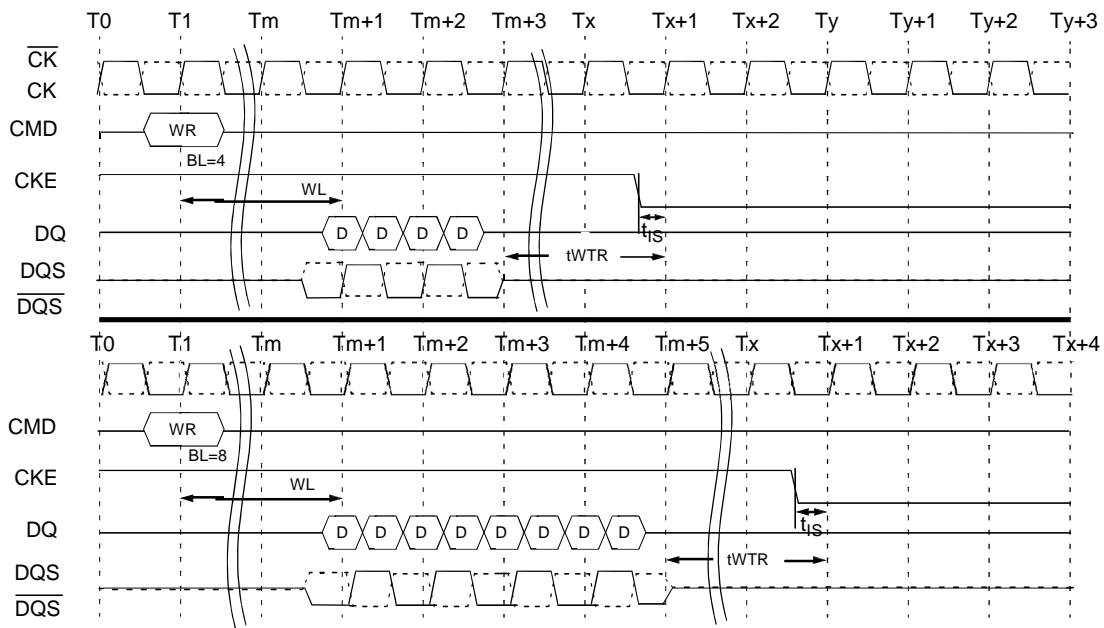
读命令到进入下电命令



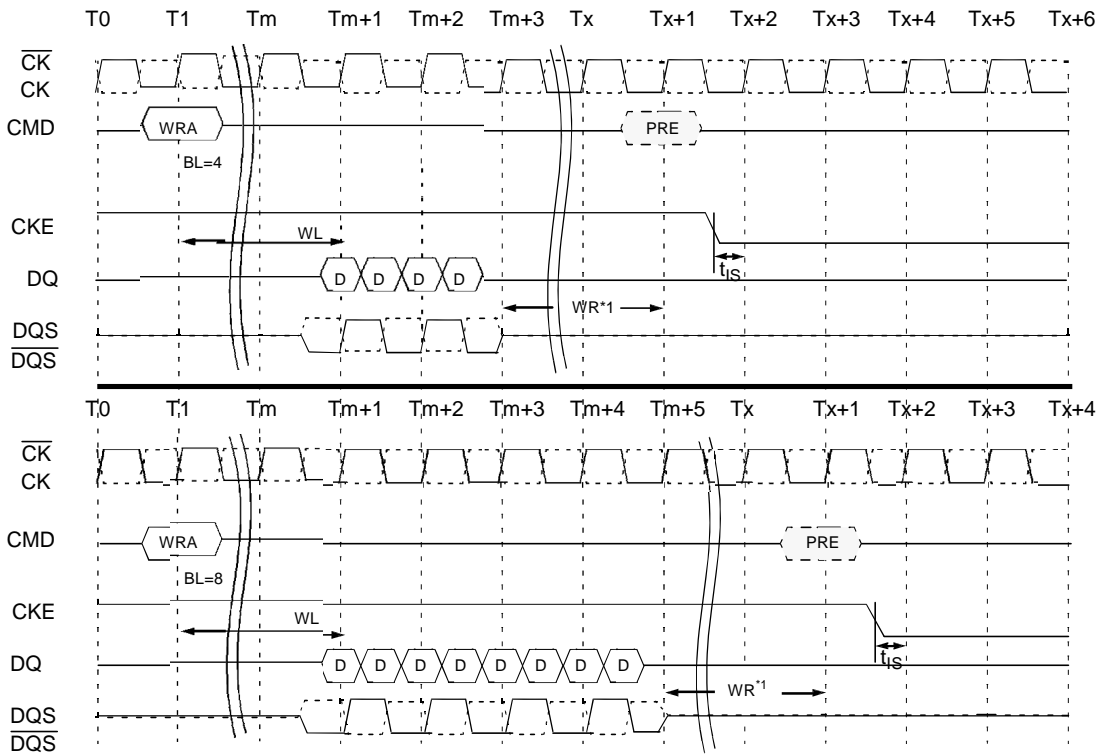
带自动预充电的读命令到进入下电命令



写命令到进入下电命令

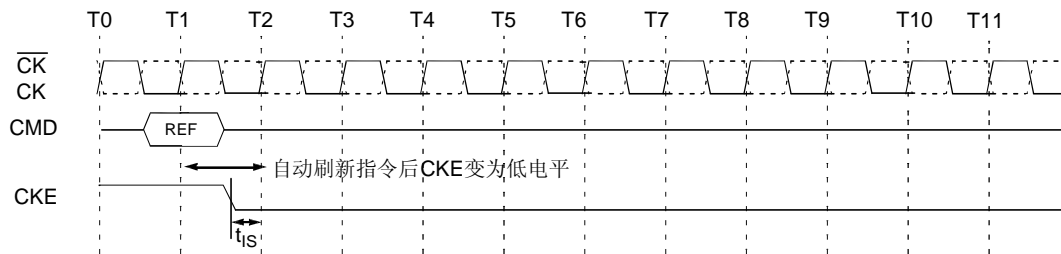


带自动预充电的读命令到进入下电命令

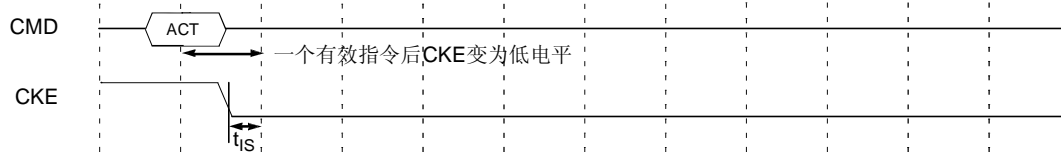


* 1: WR is programmed through MRS

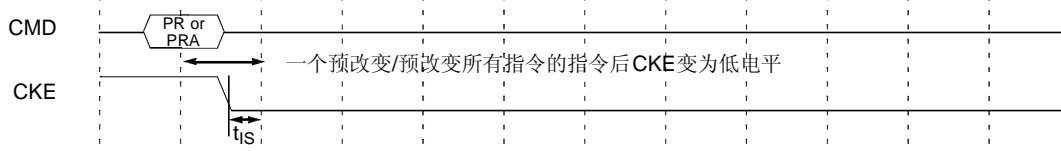
刷新命令到进入下电命令



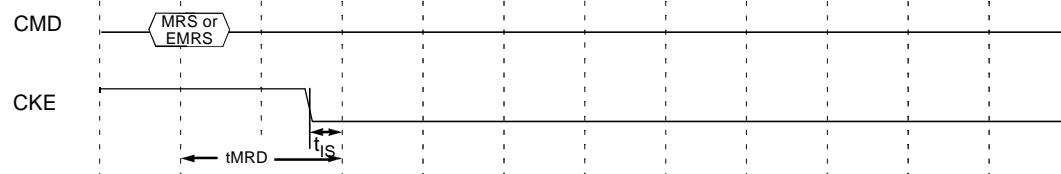
激活命令到进入下电命令



预充电/预充电所有簇命令到进入下电命令

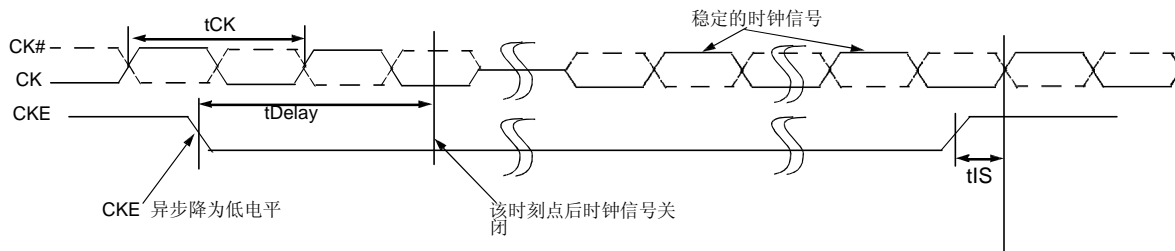


MRS/EMRS 命令到进入下电命令



异步CKE低电平事件

数据表显示DRAM要求CKE始终保持高电平，以维持所有的有效操作。如果在任何有效操作过程中，CKE异步地变为低电平，则DRAM将无法保证所保存的数据内容有效。如果这样的事件发生了，则存储控制器必须在关掉时钟信号之前满足DRAM的定时规范。CKE再次升为高电平之前，在进行DRAM输入时，必须保证稳定时钟信号存在。正如初始化程序中描述，DRAM必须完全重新初始化（步骤4到13）。初始化程序后，DRAM可供正常运行。见AC定时参数表中的 t_{Delay} 规范。

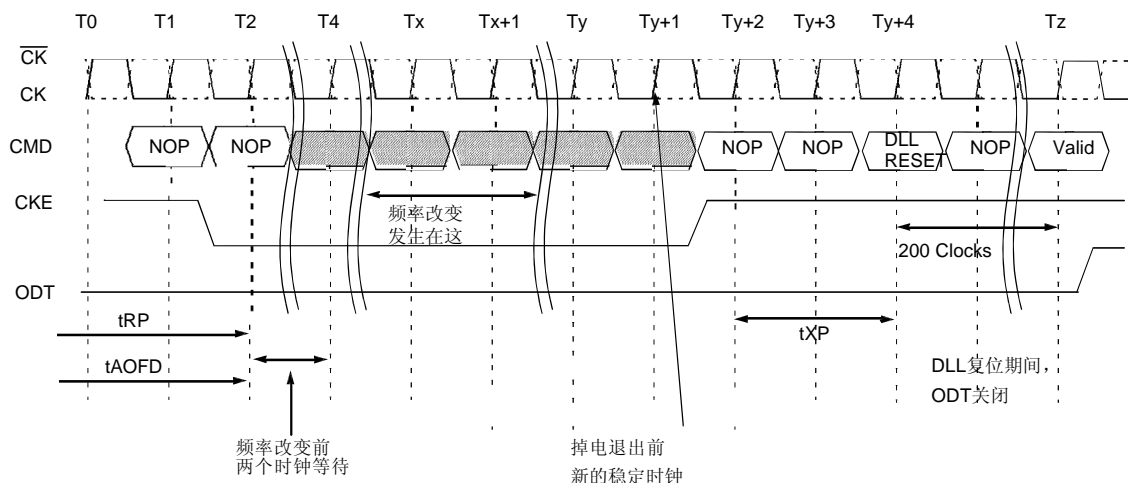


在预充电下电时改变输入时钟频率

根据以下条件，可以改变DDR2 SDRAM输入时钟频率：

DDR2 SDRAM是在预掉电模式下。必须关闭ODT，并且CKE必须为逻辑低电平。CKE变为低电平后必须等待至少两个时钟，时钟频率才能发生改变。只有当最小和最大频率为特定的速度等级时，SDRAM输入时钟频率才允许改变。输入频率改变时，ODT和CKE必须保持稳定的低电平。一旦输入频率改变，在退出预掉电之前，必须提供给DRAM新的稳定时钟，并且在退出预掉电之后，通过EMRS将DLL重新复位。依靠新的时钟频率，一个额外的MRS指令用以设置适合的WR,CL等。DLL重新锁定时期，ODT必须关闭。DLL锁定时间后，DRAM在新的时钟频率下提供正常运行。

预充电下电模式下时钟频率变化



空操作指令

空操作指令主要应用于当DDR2SDRAM处于空闲或等待的状态时。空操作指令的目的是为了防止DDR2 SDRAM在运行期间执行任何不需要的指令。必须在CS,RAS,CAS,且时钟处于上升沿时，才能执行无操作指令。一个空操作指令将不会终止之前的exe文件切割操作，如一个读写周期。

取消指令

取消指令与空操作指令具有相同的功能。取消指令发生于当时钟处于上升沿时，CS为高电平。此时，RAS,CAS,和WE等都处于无效状态。

指令真值表.

功能	CKE		CS	RAS	CAS	—WE	BA0 BA1 BA2	A15-A11	A10	A9 - A0	注意事 项
	先前周期	当前周期									
(扩展) 模式寄存器	H	H	L	L	L	L	BA	OP Code			1,2
刷新(REF)	H	H	L	L	L	H	X	X	X	X	1
进入自动刷新命令	H	L	L	L	L	H	X	X	X	X	1
退出自动刷新命令	L	H	H	X	X	X	X	X	X	X	1,7
			L	H	H	H					
单簇预充电	H	H	L	L	H	L	BA	X	L	X	1,2
所有簇预充电	H	H	L	L	H	L	X	X	H	X	1
簇激活	H	H	L	L	H	H	BA	Row Address			1,2
写	H	H	L	H	L	L	BA	Column	L	Column	1,2,3,
带自动预充电的写命令	H	H	L	H	L	L	BA	Column	H	Column	1,2,3,
读取	H	H	L	H	L	H	BA	Column	L	Column	1,2,3
带自动预充电的读命令	H	H	L	H	L	H	BA	Column	H	Column	1,2,3
空操作	H	X	L	H	H	H	X	X	X	X	1
取消指令	H	X	H	X	X	X	X	X	X	X	1
进入掉电状态命令	H	L	H	X	X	X	X	X	X	X	1,4
			L	H	H	H					
结束掉电状态命令	L	H	H	X	X	X	X	X	X	X	1,4
			L	H	H	H					

- 注意事项:
- 1 DDR2 SDRAM的所有指令都由时钟处于上升沿时， /CS, /RAS,/CAS,/WE和CK的状态来决定。
 - 2.簇地址BA0, BA1, BA2 (BA) 确定哪一个簇进行操作。(E)MRS BA 用于选择一个模式寄存器。
 - 3.突发读写在BL=4 时，不能够被终止或中断。
 - 4.在掉电模式下，不进行任何刷新操作。刷新要求叙述对掉电的持续时间有所限制。
 - 5. ODT的状态对该表所描述的状态无任何影响。自刷新中，ODT的功能不起作用。
 - 6. “X” 意思为 “高电平 or低电平 (必须是一个确定的电平)”.
 - 7. 自刷新输出是异步的。
 - 8. 自刷新操作时必须提供参考电压。

同步变化的使能真值表

当前状态 ²	CKE		指令 (N) ³ RAS, CAS, WE, CS	操作 (N) ³	注意 事项
	先前周期 ¹ (N-1)	当前周期 ¹ (N)			
掉电	L	L	X	保持掉电	11, 13, 15
	L	H	中止或空	出掉电	4, 8, 11, 13
自刷新	L	L	X	保持自刷新	11, 15
	L	H	中止或空	出自刷新	4, 5, 9
簇激活	H	L	中止或空	掉电启动	4, 8, 10, 11, 13
所有簇空闲	H	L	中止或空	进入预充电掉电	4, 8, 10, 11, 13
	H	L	刷新	自刷新启动	6, 9, 11, 13
	H	H	参考指令真值表		7

- 注意事项:
- 1. CKE (N) 是第N个时钟沿的CKE的逻辑状态; CKE (N-1)是先前时钟沿得CKE的状态。
 - 2. 当前状态是先于第N个时钟沿时DDR SDRAM的状态。
 - 3. COMMAND (N) 是在第N个时钟沿积存的指令, ACTION (N)是COMMAND (N)的结果。
 - 4. 如果在这个文件中没有明确的描述, 所有的状态或序列都属于非法的。
 - 5. 在T_{xsnr}、时期, 每一个时钟沿都必须执行自刷新结束中的中止或空指令。t_{xSRD}(200个时钟)后, 执行读取指令。
 - 6. 所有簇处于空闲状态时, 才能进入自刷新模式。
 - 7. 在指令真值表中所定义的必须是一个合法指令。
 - 8. 掉电启动和结束的有效指令只有空和中止。
 - 9. 自刷新结束的有效指令只有空和中止。
 - 10. 当正在运行读取或写入操作时, 掉电和自刷新操作不能执行。
 - 11. 三个时钟的 tCKEmin表示CKE必须存储于三个连续的正时钟边沿。整个时钟期间, CKE必须保持有效输入。因此, 在tIS+ 2*tCK + tIH 时期, 任何CKE过渡后, CKE不必过渡有效输入。
 - 12. 该表所描述的状态均不受ODT的状态影响。自刷新中, ODT的功能不起作用。
 - 13. 在掉电模式下, 不进行任何刷新操作。刷新要求叙述对掉电的持续时间有所限制。
 - 14. SDRAM在OCD模式时, CKE必须保持高电平。
 - 15. “X”表示在自刷新和掉电中不起作用。 然而, 如果ODT实现功能, 则ODT必须在掉电中以高电平或低电平启动。
 - 16. 自刷新操作时必须提供参考电压。

DM 真值表

名称 (功能)	DM	DQs	注意
写入使能	L	有效	1
写入禁止	H	X	1

注意1 :用于隐藏数据, 并提供相应的数据。

绝对最大直流率

符号	参数	额定值	单位	备注
V_{DD}	V_{DD} 脚相对的电压 V_{SS} 的电压	- 1.0 V ~ 2.3 V	V	1
V_{DDQ}	V_{DDQ} 脚相对的电压 V_{SS} 的电压	- 0.5 V ~ 2.3 V	V	1
V_{DDL}	V_{DDL} 脚相对的电压 V_{SS} 的电压	- 0.5 V ~ 2.3 V	V	1
V_{IN}, V_{OUT}	任何脚相对的电压 V_{SS} 的电压	- 0.5 V ~ 2.3 V	V	1
T_{STG}	储存温度	-55 to +100	°C	1, 2

- 注意事项:
- 1.若电压大于上述所列绝对最大电压率可能会对设备造成永久的损坏 。这是一个设备在这些条件和上述其他条件下的仅电压率和功能操作 ， 延长的周期完全暴露在最大电压级下会影响稳定性。
 2. 存储器温度指的是中央/顶部面的表面温度。测量条件，请参见JESD51-2 标准。

交流和直流的操作条件

推荐的直流操作条件 (SSTL - 1.8)

符号	参数	级别			单位	备注
		最小	典型	最大		
V_{DD}	提供电压	1.7	1.8	1.9	V	1
V_{DDL}	DLL供电	1.7	1.8	1.9	V	5
V_{DDQ}	输出供电	1.7	1.8	1.9	V	1, 5
V_{REF}	输入参考电压	$0.49 \cdot V_{DDQ}$	$0.50 \cdot V_{DDQ}$	$0.51 \cdot V_{DDQ}$	mV	2, 3
V_{TT}	终止电压	$V_{REF}-0.04$	V_{REF}	$V_{REF}+0.04$	V	4

- 注 : SSTL-1.8 没有特殊设备 V_{DD} 供电需求。但是在任何条件下 V_{DDQ} 必须低于或者等于 V_{DD} 。
1. 用户可以选择 V_{REF} 的值以提供系统中最佳的噪声容限。通常 V_{REF} 的值设定为高于0.5 x 发射设备的 V_{DDQ} ，并且 V_{REF} 随着 V_{DDQ} 而变化
 2. V_{REF} 上的峰峰交流噪声不会超过 +/-2% $V_{REF}(DC)$ 。
 3. 发射设备的 V_{TT} 必须随着接受设备的 V_{REF} 变化。
 4. V_{DDQ} 跟随 V_{DD} , V_{DDL} 随 V_{DD} . 交流参数必须与 V_{DD} , V_{DDQ} 和 V_{DDL} 相符。

输入直流逻辑电平

符号	参数	最小	最大	单位	备注
$V_{IH}(DC)$	DC 输入逻辑高	$V_{REF} + 0.125$	$V_{DDQ} + 0.3$	V	
$V_{IL}(DC)$	DC 输入逻辑低	- 0.3	$V_{REF} - 0.125$	V	

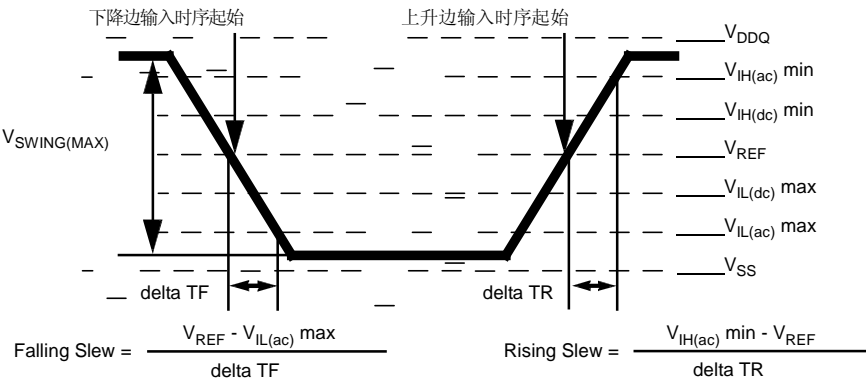
输入交流逻辑电平

符号	参数	DDR2-400, DDR2-533		DDR2-667, DDR2-800		单位	备注
		最小	最大	最小	最大		
$V_{IH}(AC)$	AC 输入逻辑高	$V_{REF} + 0.250$	-	$V_{REF} + 0.200$		V	
$V_{IL}(AC)$	AC 输入逻辑低	-	$V_{REF} - 0.250$		$V_{REF} - 0.200$	V	

交流输入测试条件

符号	条件	值	单位	备注
V_{REF}	输入参考电压	$0.5 * V_{DDQ}$	V	1
$V_{SWING(MAX)}$	输入信号最大峰峰振幅	1.0	V	1
SLEW	输入信号最小回速率	1.0	V/ns	2, 3

- 备注:
- 1. 输入波形时序参考应用在测试设备上的 $V_{IH/IL}(AC)$ 电平的输入交叉信号。
 - 2. 输入信号最小回速率保持在下图所示的范围，上升边从 V_{REF} 到 $V_{IH}(AC)$ 和下降边从 V_{REF} 到 $V_{IL}(AC) \max$ 。
 - 3.直流时序参考正跃迁的 $V_{IL}(AC)$ 到 $V_{IH}(AC)$ 和负跃迁 $V_{IH}(AC)$ 到 $V_{IL}(AC)$ 转换的输入波形。



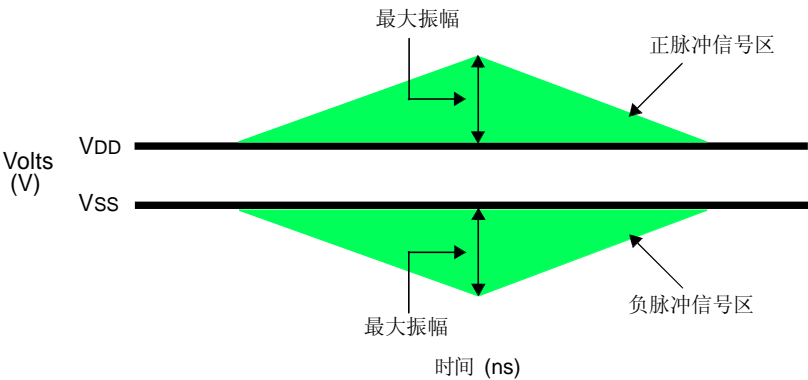
< 交流输入测试信号波形 >

正/负脉冲信号规格

地址和控制引脚 **A0-A15, BA0-BA2, CS, RAS, CAS, WE, CKE, ODT** 的交流正/负脉冲信号规格

参数	规格			
	DDR2-400	DDR2-533	DDR2-667	DDR2-800
正脉冲信号区允许的最大峰值振幅 (见下图):	0.5(0.9)*V	0.5(0.9)*V	0.5(0.9)*V	0.5(0.9)*V
负脉冲信号区允许的最大峰值振幅 (见下图):	0.5(0.9)*V	0.5(0.9)*V	0.5(0.9)*V	0.5(0.9)*V
最大正脉冲信号区高于 VDD (见下图):	1.33 V-ns	1.0 V-ns	0.8 V-ns	0.66 V-ns
最大负脉冲信号区低于 VSS (见下图):	1.33 V-ns	1.0 V-ns	0.8 V-ns	0.66 V-ns

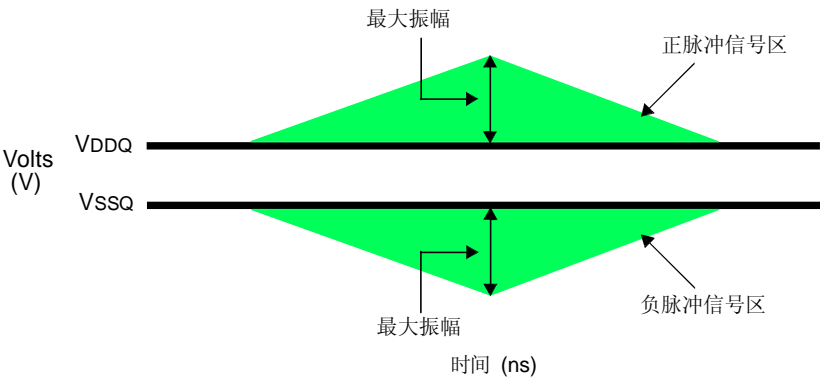
注：峰值振幅的最大需求从0.9V减少到 0.5V 。寄存器数据表将说明特殊RDIMM应用程序中检测出的最大正/负脉冲信号。DRAM 数据表也会表明DRAM所容许的最大正/负脉冲信号。这能够使RDIMM提供者知道DRAM是否能承受特殊RDIMM应用程序中寄存器感应出的正脉冲信号。



地址和控制引脚的交流正/负脉冲信号定义

时钟, 数据, 闸, 和Mask 引脚 **DQ, DQS, DM, CK, CK** 的交流正/负脉冲信号说明

参数	规格			
	DDR2-400	DDR2-533	DDR2-667	DDR2-800
正脉冲信号区允许的最大峰值振幅 (见下图):	0.5V	0.5V	0.5V	0.5V
负脉冲信号区允许的最大峰值振幅 (见下图):	0.5V	0.5V	0.5V	0.5V
最大正脉冲信号区高于 VDDQ (见下图):	0.38 V-ns	0.28 V-ns	0.23 V-ns	0.23 V-ns
最大负脉冲信号区低于 VSSQ (见下图):	0.38 V-ns	0.28 V-ns	0.23 V-ns	0.23 V-ns



时钟, 数据, 闸, 和Mask 引脚的交流正/负脉冲信号定义

表1. 满力驱动默认下拉驱动特性

电压 (V)	下拉电流 (mA)			
	最小(23.4 Ohms)	表征缺省低(18 ohms)	表征缺省高(18 ohms)	最小(12.6 Ohms)
0.2	8.5	11.3	11.8	15.9
0.3	12.1	16.5	16.8	23.8
0.4	14.7	21.2	22.1	31.8
0.5	16.4	25.0	27.6	39.7
0.6	17.8	28.3	32.4	47.7
0.7	18.6	30.9	36.9	55.0
0.8	19.0	33.0	40.9	62.3
0.9	19.3	34.5	44.6	69.4
1.0	19.7	35.5	47.7	75.3
1.1	19.9	36.1	50.4	80.5
1.2	20.0	36.6	52.6	84.6
1.3	20.1	36.9	54.2	87.7
1.4	20.2	37.1	55.9	90.8
1.5	20.3	37.4	57.1	92.9
1.6	20.4	37.6	58.4	94.9
1.7	20.6	37.7	59.6	97.0
1.8		37.9	60.9	99.1
1.9				101.1

图 1. 满力度 驱动下DDR2 缺省的下拉特性

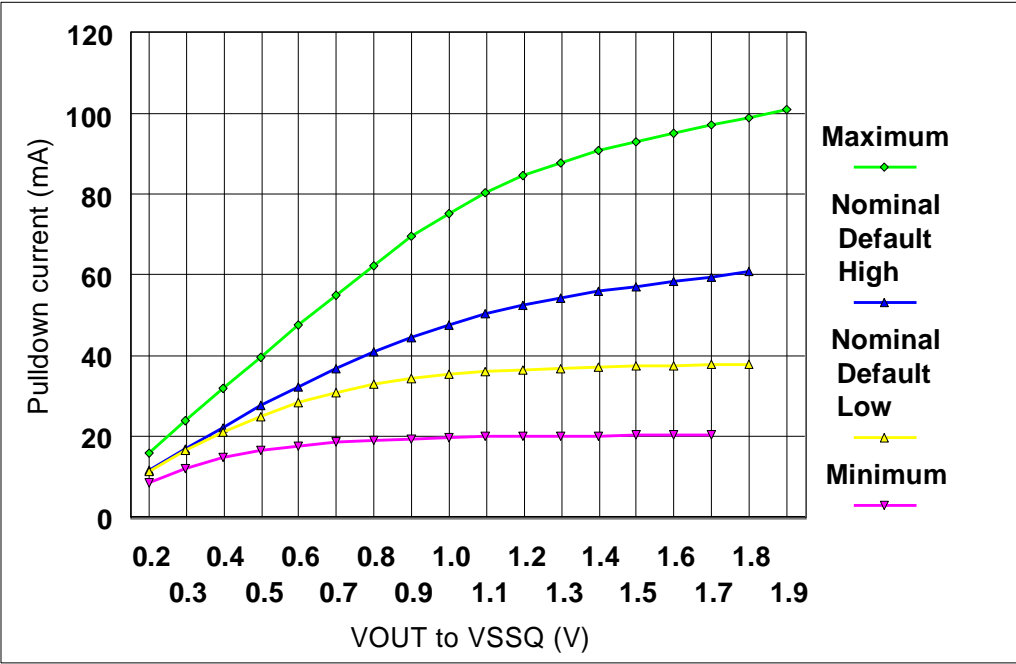


表 2. 满力驱动默认上拉驱动特性

Voltage (V)	Pullup Current (mA)			
	Minimum(23.4 Ohms)	Nominal Default Low (18 ohms)	Nominal Default High (18 ohms)	Maximum(12.6 Ohms)
0.2	-8.5	-11.1	-11.8	-15.9
0.3	-12.1	-16.0	-17.0	-23.8
0.4	-14.7	-20.3	-22.2	-31.8
0.5	-16.4	-24.0	-27.5	-39.7
0.6	-17.8	-27.2	-32.4	-47.7
0.7	-18.6	-29.8	-36.9	-55.0
0.8	-19.0	-31.9	-40.8	-62.3
0.9	-19.3	-33.4	-44.5	-69.4
1.0	-19.7	-34.6	-47.7	-75.3
1.1	-19.9	-35.5	-50.4	-80.5
1.2	-20.0	-36.2	-52.5	-84.6
1.3	-20.1	-36.8	-54.2	-87.7
1.4	-20.2	-37.2	-55.9	-90.8
1.5	-20.3	-37.7	-57.1	-92.9
1.6	-20.4	-38.0	-58.4	-94.9
1.7	-20.6	-38.4	-59.6	-97.0
1.8		-38.6	-60.8	-99.1
1.9				-101.1

图 2. 满力度 驱动下DDR2 缺省的上拉特性

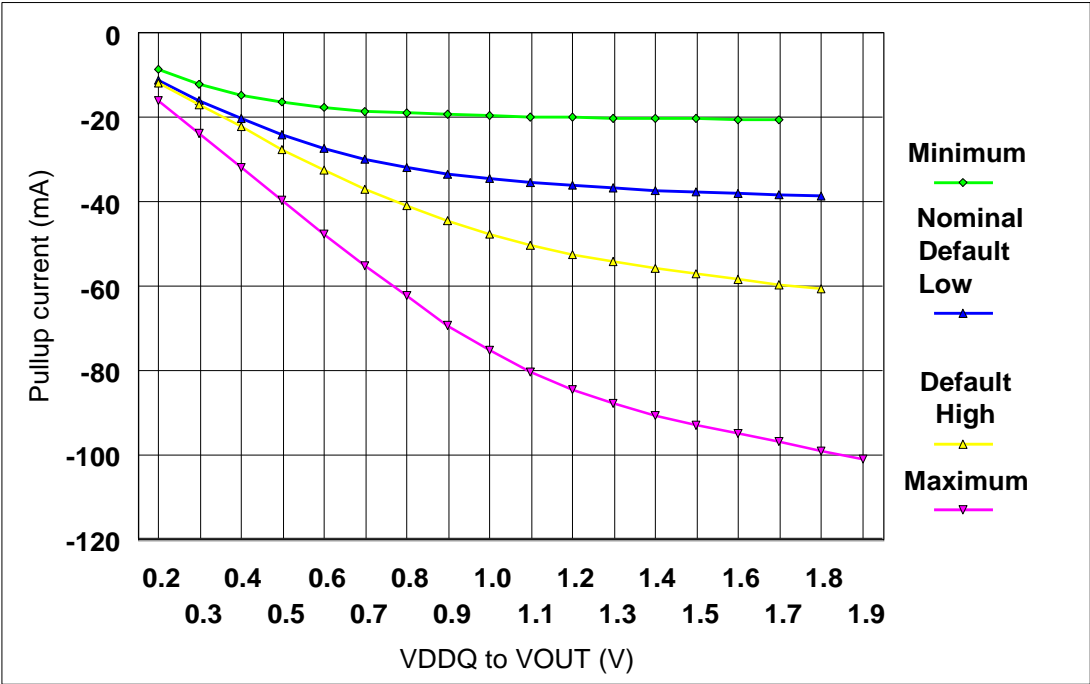


表 3. 递减驱动力度下的默认下拉驱动特性。

Voltage (V)	Pulldown Current (mA)			
	Minimum	Suggested IBIS Target Low	Suggested IBIS Target High	Maximum
0.0	0.00	0.00	0.00	0.00
0.1	1.72	3.24	4.11	4.77
0.2	3.44	6.25	8.01	9.54
0.3	5.16	9.03	11.67	14.31
0.4	6.76	11.52	15.03	19.08
0.5	8.02	13.66	18.03	23.85
0.6	8.84	15.41	20.61	28.62
0.7	9.31	16.77	22.71	33.33
0.8	9.64	17.74	24.35	37.77
0.9	9.89	18.38	25.56	41.73
1.0	10.09	18.80	26.38	45.21
1.1	10.26	19.06	26.90	48.21
1.2	10.41	19.23	27.24	50.73
1.3	10.54	19.35	27.47	52.77
1.4	10.66	19.46	27.64	54.42
1.5	10.77	19.56	27.78	55.80
1.6	10.88	19.65	27.89	57.03
1.7	10.98	19.73	27.97	58.23
1.8		19.80	28.02	59.43
1.9				60.63

图e 3.递减驱动力度下的 DDR2 默认下拉特性

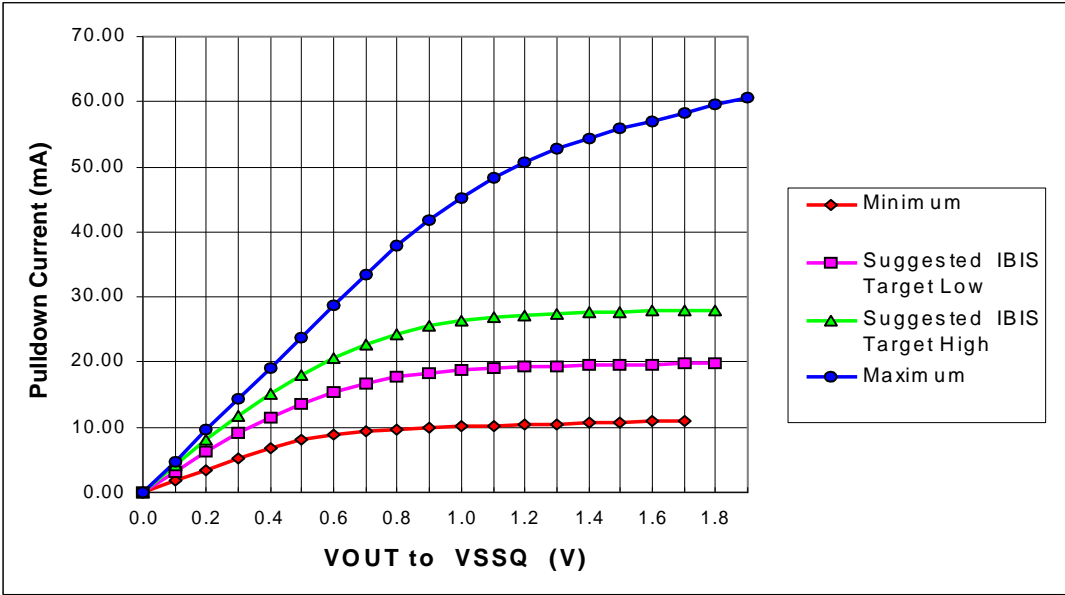
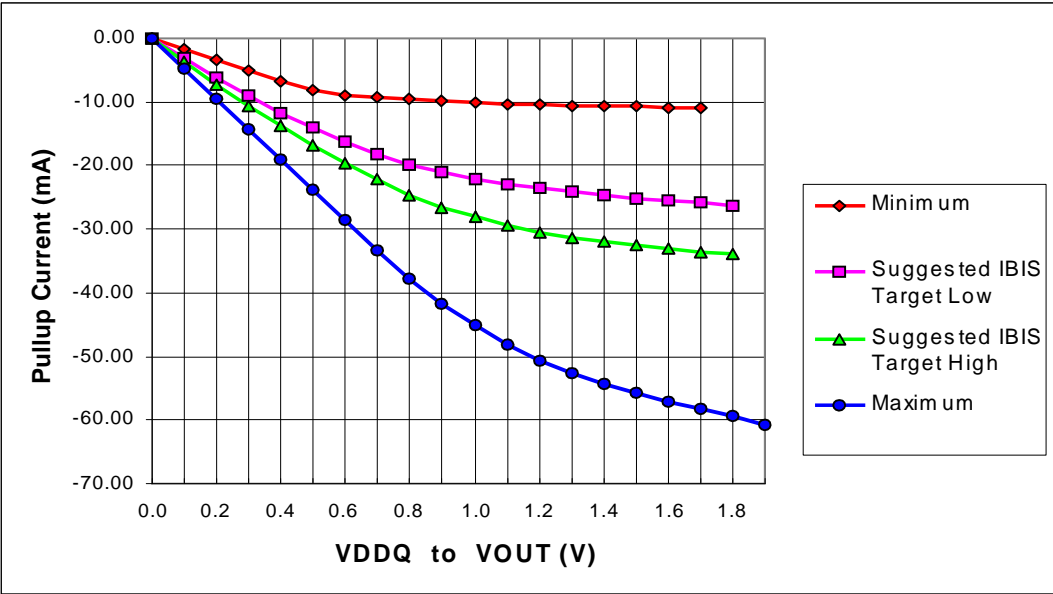


表 4. 递减驱动力度下的默认下拉驱动特性

Voltage (V)	Pulldown Current (mA)			
	Minimum	Suggested IBIS Target Low	Suggested IBIS Target High	Maximum
0.0	0.00	-0.00	0.00	0.00
0.1	-1.72	-3.200	-3.70	-4.77
0.2	-3.44	-6.200	-7.22	-9.54
0.3	-5.16	-9.040	-10.56	-14.31
0.4	-6.76	-11.690	-13.75	-19.08
0.5	-8.02	-14.110	-16.78	-23.85
0.6	-8.84	-16.270	-19.61	-28.62
0.7	-9.31	-18.160	-22.20	-33.33
0.8	-9.64	-19.770	-24.50	-37.77
0.9	-9.89	-21.100	-26.46	-41.73
1.0	-10.09	-22.150	-28.07	-45.21
1.1	-10.26	-22.960	-29.36	-48.21
1.2	-10.41	-23.610	-30.40	-50.77
1.3	-10.54	-24.160	-31.24	-52.77
1.4	-10.66	-24.640	-31.93	-54.42
1.5	-10.77	-25.070	-32.51	-55.80
1.6	-10.88	-25.470	-33.01	-57.03
1.7	-10.98	-25.850	-33.46	-58.23
1.8		-26.210	-33.89	-59.43
1.9				-60.63

图 4. 递减驱动力度下的DDR2 默认下拉特性



DDR2 SDRAM 默认输出驱动 V-I 特性

DDR2 SDRAM 输出驱动特性被定义为满力度的默认操作，这通过设定EMRS1的 A7-A9 = ‘111’来实现。图 1 和 2 展示了这种驱动特性。表1和表2以表格形式展示了同样的数据。驱动特性的评估条件如下：

默认 25 °C (摄氏度), VDDQ = 1.8 V, 典型值
最小 T_{OPER(MAX)}, VDDQ = 1.7 V, 慢慢 进程
最小0 °C (T case), VDDQ = 1.9 V, 快-快 进程

默认输出驱动特性曲线图：

- 1) 在驱动电流从最小到最大的变化过程中，温度和电压将影响V-I特性
如图 1 和 2。
- 2)推荐使用 "典型的" IBIS V-I曲线图 。

表 3.满力度校准下拉驱动特性

Voltage (V)	Calibrated Pulldown Current (mA)				
	Nominal Minimum	Nominal Low (18.75 ohms)	Nominal (18 ohms)	Nominal (17.25 ohms)	Nominal Maximum (15 ohms)
0.2	9.5	10.7	11.5	11.8	13.3
0.3	14.3	16.0	16.6	17.4	20.0
0.4	18.7	21.0	21.6	23.0	27.0

表 4. 满力度校准上拉驱动特性

Voltage (V)	Calibrated Pullup Current (mA)				
	Nominal Minimum (21 ohms)	Nominal Low (18.75 ohms)	Nominal (18 ohms)	Nominal (17.25 ohms)	Nominal Maximum (15 ohms)
0.2	-9.5	-10.7	-11.4	-11.8	-13.3
0.3	-14.3	-16.0	-16.5	-17.4	-20.0
0.4	-18.7	-21.0	-21.2	-23.0	-27.0

DDR2 SDRAM 校准输出驱动 V-I 特性

DDR2 SDRAM 输出驱动特性被定义为满力度校准操作，有OCD阻抗调整特性来选择。表 3和 4以表格形式展示了适合输入到仿真工具的数值。标称点代表了器件精确在18欧姆。标称的低和标称的高值代表了在无错误的情况下最大每步可以调整 1.5欧姆。(i.e. 如规格书上所标，最大正确的单步调整大小是1.5欧姆)。实际的系统校正错误也要加到这些值上面。我们必须明白，为了修正系统的校准错误，模型需要校准到一个很宽的范围。由于这是一个系统的细节状况，所以在这里不能够作到完全量化。校准表中的值代表的仅仅是只关注 DQ脚时，DRAM的部分特性，有不确定性。

如果使用了校准过程，可能导致芯片工作在表中所列的默认特性曲线以外。在这种情况下，就不能保证规格书中所列的时序参数的正确性。所以在实习系统应用时，要确保芯片在任何时间都要被校准在最大和最小默认值之间。如果在系统校正过程或重校正过程中这一点无法得到保证，那么建议大家使用默认的值。标示的最大和最小的值代表了当电压和温度在最小和最大值之间变化时，阻抗的最小和最大的变化。如果是在极限条件下校准，那么校准的量就相当于标示的最小和最大值之间的差。

驱动特性的评估条件是：

- : Nominal 25 °C (T case), VDDQ = 1.8 V, 典型过程
- Nominal Low and Nominal High 25 °C (T case), VDDQ = 1.8 V, 任何过程
- Nominal Minimum T_{OPER(MAX)}, VDDQ = 1.7 V, 任何过程
- Nominal Maximum 0 °C (T case), VDDQ = 1.9 V, 任何过程。

Device Operations

DDR2 SDRAM