

基于 Close Page Policy 的 SDRAM 控制器设计原理

复旦大学 梁晨 整理于 2012/04/08

1. SDRAM 的概述

SDRAM，意为同步的 DRAM，最早的样品，由三星在 1993 年生产。在 1996-2002 年期间，SDRAM 逐步取代了异步的 FPM DRAM、EDO DRAM，称雄 PC 内存市场。在 2003 年之后，逐渐被 DDR SDRAM 取代。它与前代异步 DRAM 的不同包括：同步时钟、多 Bank 机制、Burst 读写的引入。由于每个时钟周期，只在上升沿传送一次数据，它也被称为 SDR SDRAM，以便与 DDR SDRAM 区别。

1.2 SDRAM 基本硬件参数

SDRAM 的常见容量包括：16Mbit、64Mbit、128Mbit、256Mbit、512Mbit。其中，除了 16Mbit 分为 2 个 Bank 外，其他容量的 SDRAM 都分为 4 个 Bank。

SDRAM 的常见数据位宽包括：4bit、8bit、16bit、32bit。如果系统位宽 64bit，用 16 个 512Mbit 容量 4bit 位宽的 SDRAM 并联，可以实现 8Gbit 的存储空间；而如果用 2 个 512Mbit 容量 32bit 位宽的 SDRAM 并联，只能实现 1Gbit 的存储空间。

SDRAM 的 Burst Length，即读写突发传送的周期数，一般可设置为 1、2、4、8 以及 Full Page。Burst Length 是可编程的，在初始化时设置即可。需要注意的是，如果地址没有与 BL 对齐，SDRAM 会自行 Wrap，导致意外的结果。

SDRAM 的 Cas Latency，即从发送读指令到有效数据出现的延时，一般为 2 或 3 个时钟周期，少数器件可以为 1 个周期。大部分 SDRAM 的 CL 可在初始化时编程设置。一般 CL=3 时，对应的 AC 延时参数较小。

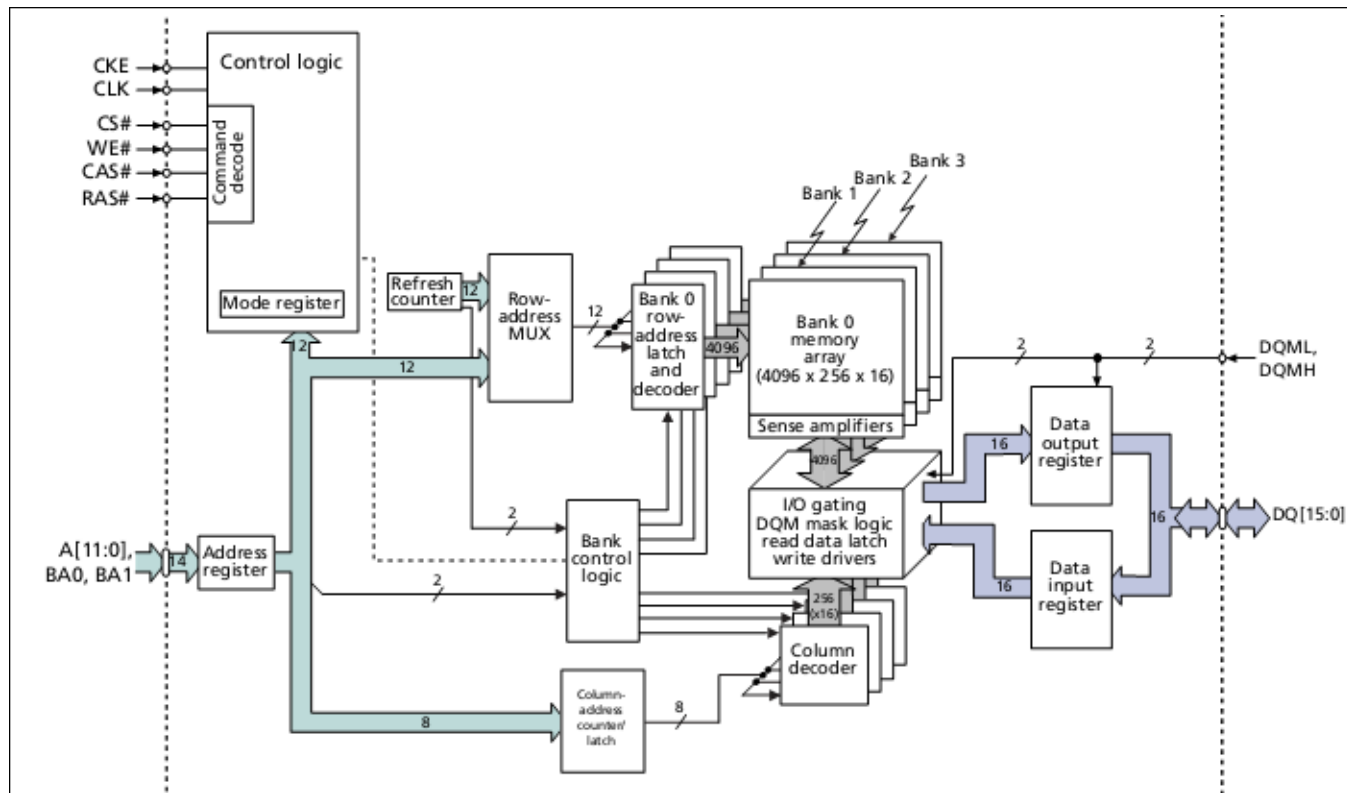
SDRAM 的常见速度等级包括：100MHz (T=10ns)、125MHz(T=8ns)、133MHz (T=7.5ns)、143MHz (T=7ns)、166MHz (T=6ns)、183MHz(T=5.5ns)、200MHz (T=5ns)。速度等级一般对应 CL=3 时的最高工作频率。一般最高工作频率越高，对应的 AC 延时参数越小。

1.3 SDRAM 的三维地址空间

当代 SDRAM 的地址空间由 Bank、Row、Column 三个维度。64Mbit 及以上容量的 SDRAM 一般分为 4 个 Bank。为了与内存模组中 Physical Bank (Rank) 的概念区别，SDRAM 的 Bank 有时也叫 Logic Bank。每个 Bank 都是一个 Row 与 Column 构建的阵列；阵列的每个单元都是一组数据，这组数据的位宽与 SDRAM

数据接口位宽一致。

例如，一个容量 64Mbit 位宽 16bit 的 SDRAM 地址空间为： $4\text{bank} * (2^{12})\text{Row} * (2^8)\text{Column} * 16\text{ bit} = 64\text{Mbit}$ ，器件手册一般写为 $4 * 1\text{M} * 16\text{bit} = 64\text{Mbit}$ 。下图摘自镁光 SDRAM 器件手册：



1.2.3 芯片规格与 SDRAM 芯片组的容量

关于 SDRAM 芯片寻址的术语——Bank、Row、Column 已在上一节给出了解释。下面解释几个 SDRAM 芯片组中常用的术语。

Rank: 又叫 Physical Bank，指的是一组被同时操作的 DRAM 芯片。通常，它们的数据总线被合并，以提供系统所需要的数据位宽，如 Intel 处理器要求的 64bit。

Page: 一般指同一个 Rank 中 Bank address、Row address 相同的空间。可以认为将 Rank 中所有 SDRAM 芯片中的 Row Buffer 合并在一起，就组成了 Rank 中的 Page Buffer。关于 Row Buffer、Page Buffer 对 DRAM 访问的影响将在下一节讲述。

以 64bit 位宽的系统为例，综合列出不同规格 SDRAM 的 3 维地址空间，以及组成 Rank 后的特性：

Chip Size	Chip DQ/ Fab	Bank Num	Row Num	Col Num	Row Buf Size	64bit Page Size	64bit Rank Size
16Mb	16 ISSI	2	2048	256	4kb	16kb	64Mb

Chip Size	Chip DQ/ Fab	Bank Num	Row Num	Col Num	Row Buf Size	64bit Page Size	64bit Rank Size
64Mb	32 Micron	4	2048	256	8kb	16kb	128Mb
	16 Micron	4	4096	256	4kb	16kb	256Mb
	8 Micron	4	4096	512	4kb	32kb	512Mb
	4 Micron	4	4096	1024	4kb	64kb	1Gb
128Mb	32 Micron	4	4096	256	8kb	16kb	256Mb
	16 Micron	4	4096	512	8kb	32kb	512Mb
	8 Micron	4	4096	1024	8kb	64kb	1Gb
	4 Micron	4	4096	2048	8kb	128kb	2Gb
256Mb	32 ISSI	4	4096	512	16kb	32k	512Mb
	16 Micron	4	8192	512	8kb	32kb	1Gb
	8 Micron	4	8192	1024	8kb	64kb	2Gb
	4 Micron	4	8192	2048	8kb	128kb	4Gb
512Mb	32 ISSI	4	8192	512	16kb	32kb	1Gb
	16 Micron	4	8192	1024	16kb	64kb	2Gb
	8 Micron	4	8192	2048	16kb	128kb	4Gb
	4 Micron	4	8192	4096	16kb	256kb	8Gb

注：其中 Micron 64Mb: x32 的芯片虽然是 2048 行，计算刷新周期时要按 4096 行算。而 ISSI 16Mb: x16 的芯片也是 2048 行，刷新时确实按 2048 行算。关于刷新的原理见后文。

1.2.4 Row Buffer 对 SDRAM 读写的影响。

SDRAM 的每个 Bank 都有一个 Row Buffer，其实就是一排灵敏放大器(Sense Amplifier)。每次激活(Activate)一个 Row 地址后，就打开 Row Buffer 把整个 Row 的数据保存在其中；Precharge 指令可以关闭指定 Bank 的 Row Buffer，Precharge All 可以关闭所有 Bank 的 Row Buffer；我们称 Row Buffer 关闭的 Bank 为空闲(Idle)的 Bank。

由于 Row Buffer 的存在，SDRAM 的访问分为如下三种情况：

(1) 当前访问的 Row 所在的 bank 中，row buffer 关闭。此时，需要先激活 Row，再发读写指令和 Column 地址。此情况，读写延时中等，也称为 Page Hit。

(2) 当前访问的 Row，正好保存在相应 bank 的 row buffer 里。此时，不需要激活 Row，直接发读写指令和 Column 地址，就可以访问，这就是 Back to Back 读写。此情况，读写延时最小，也称为 Page Fast Hit、Page Direct Hit。

(3) 当前访问的 Row 所在的 bank 中，row buffer 打开，存的是另一个 Row 的数据。此时，必须先发送 Precharge 或 Precharge All 指令关闭 row buffer，再激活 Row，最后发读写指令和 Column 地址。此情况，读写延时最大，也称为 Page Miss。

1.3 Close Page Policy 的原理与有效带宽分析

1.3.1 Close Page Policy 概述

Close Page Policy 是一种简单有效的 SDRAM 控制器设计策略。它的性能与访问地址的空间分部无关，带宽利用率固定，不考虑 Refresh 影响时读写延时恒定，适合于实时性要求较高的应用。

它保证每次 SDRAM 访问都是读写延时中等的 Page Hit 情况，手段很简单：每次读写后自动 Precharge，关闭 row buffer。这样不会因为同一 Bank 内 Row 地址切换而有 Penalty，也不会因为访问地址的空间 locality 有 Gain，所以性能稳定性好。但另一方面，大量的 Precharge 给 SDRAM 带来较大的功耗。

1.3.2 Close Page Policy 的实现与性能分析

典型的 Close Page Policy 操作，只涉及 Refresh、Activate & Read with Autoprecharge、Activate & Write with Autoprecharge 三种情况。以下讲解这三种基本操作，相关的时序参数，并推导出理想情况下的有效读写带宽公式。

注：此处先假设控制器实现时未使用 SDRAM 指令流水线。

1.3.2.1 Refresh 操作和它对有效带宽的影响

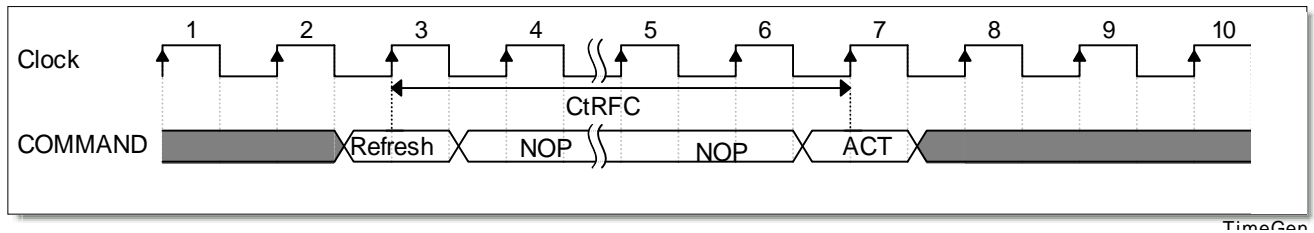
由于 DRAM 的 1T1C 结构存在漏电，必须周期性地刷新(Refresh)以持久保持数据。通用 SDR SDRAM 要求在 64ms 内将所有 Row 刷新一遍，这个周期称为 tREF。SDRAM 的 row 数一般为 4096 或 8192，故如果均匀地刷新，每隔 15.625us 或 7.8125us 要发送一次 Refresh 指令。有时，即使 SDRAM 的 row 数不到 4096，刷新周期也按 4096 row 的情况计算。

现代 DRAM 内部都包含刷新计数器，发 Refresh 指令时，不需要发送 Row 地址，故把 Refresh 也称为 Column Before Row 操作。每次 Refresh 指令发送后，必须等待 tRFC 时间才能发送下一条指令，这就是刷新延时。只要满足 tRFC、tREF，连续的突发 Refresh 也是允许的。

执行 Refresh 操作前，必须保证每个 Bank 都空闲，如果任意一个 Bank 中打开的 Row Buffer，必须先发送 Precharge All 或 Precharge 指令。因为 Close Page Policy 以读写后立刻关闭 Row Buffer 为特点，所以它在 Refresh 之前，不需要这些指令。

按照器件手册的规定，AC 参数折合为周期数时要向上取整。下文中，将以 CtREF 表示 tREF 对应的周期数，以 CtRFCm1 表示 tREF 对应的周期数减 1，其他以此类推，不再赘述。

刷新操作的时序图如下：



于是，刷新导致的带宽损失计算如下：

$$BW_{available} = BW \times \left(1 - \frac{(CtRFCm1 + 1) \times RowNum}{CtREF}\right)$$

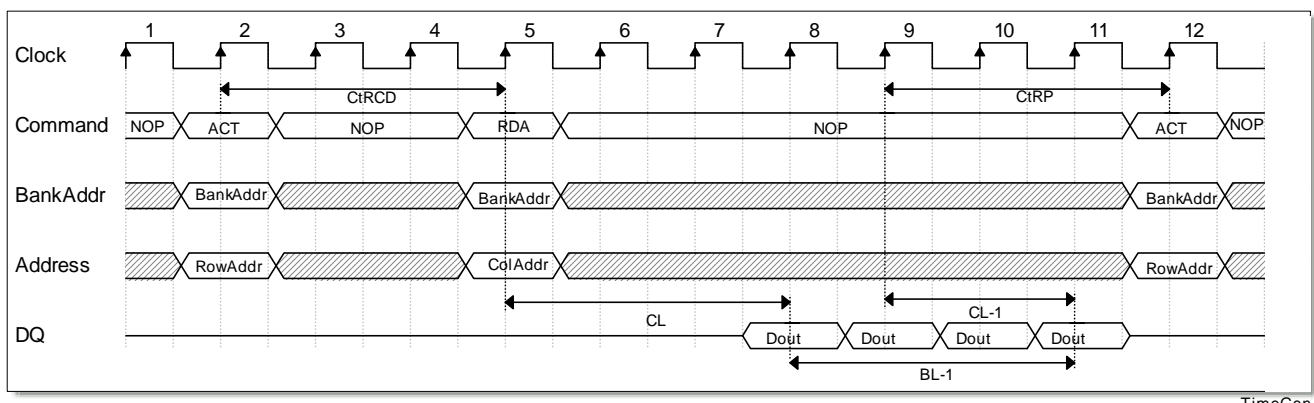
1.3.2.2 Activate & Read with Autoprecharge 操作与 Close Page Policy 的读效率分析

Close Page Policy 中，连续的读操作可以认为是 Activate 和 Read with Autoprecharge 操作的循环。

Activate 命令用于激活 Bank 中的一个 Row。发此命令的同时，要给出 Bank 地址和 Row 地址。Activate 命令后要隔 tRCD 才能发送读写指令。

Read with Autoprecharge 命令的作用是：读取数据，然后自动执行 Precharge 关闭 Row Buffer。发此命令的同时，要给出 Bank 地址和 Column 地址。Read with Autoprecharge 命令后，隔 CL 个周期才会有有效数据输出，输出是连续的。自动的 precharge 发生在出现最后一个有效数据的时钟上升沿前的(CL-1)个周期，同时 Precharge 延时参数 tRP 必须被满足。

以 CL=3, BL=4 为例，画出整个读操作周期的时序图如下：



由于每次读写后都关闭 Row Buffer，故对于 Close Page Policy，读延时恒定为 Activate 延时加 Cas Latency，即 $1+CtRCDm1+1+(CL-1)$ ，即 $CtRCD+CL$ 。

考虑到 Refresh 操作的影响，假设指令数据不重合，推得理想的读有效带宽公式如下：

$$BW_{effrd} = BW \times \frac{BL}{1+CtRCDm1+1+(CL-1)+BL+\max((CtRPm1-(CL-1)),0)} \times (1 - \frac{(CtRFCm1+1) \times RowNum}{CtREF})$$

由时序图以及以上公式可知，选取较大的 BL 可以提高带宽利用率。而由于时序参数与 CL 相关，选取较小的 CL 不一定能提高提高带宽利用率。选择较高的时钟频率，可以提高硬件带宽，但可能导致命令延时对应的周期数增加，故需要综合考虑。

更普适的带宽计算需要补充 5 点条件：

(1) SDRAM 允许指令与数据重叠：可以在 SDRAM 输出数据时进行指令操作；

(2) Activate 指令与 Precharge 指令之间必须相隔 $tRAS(min)$ 的时间，在 BL=1 或 2 时可能会成为限制条件；

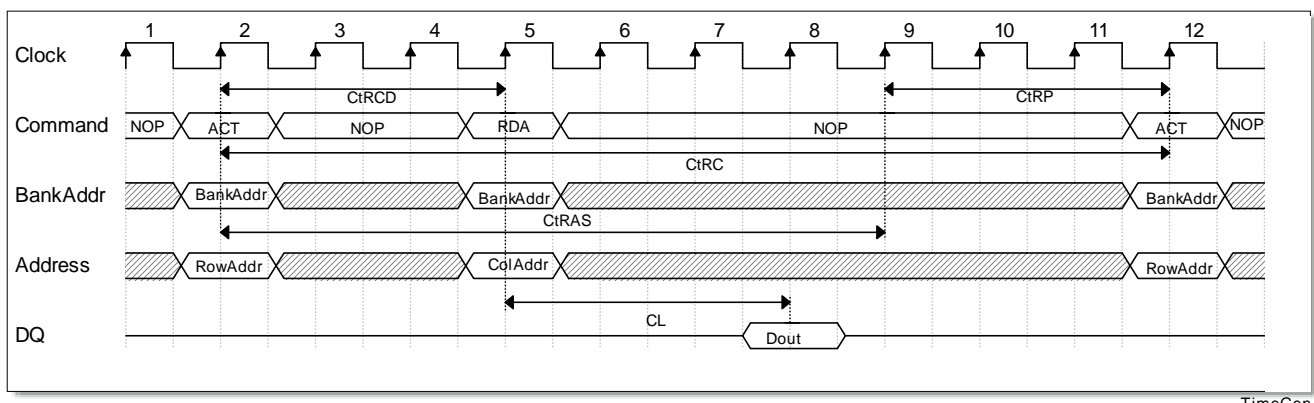
(3) 一个 Row Buffer 被打开的时间不可超过 $tRAS(max)$ 。 $tRAS(max)$ 远大于其它 AC 延时参数，除非时钟频率低到离谱，对 Close Page Policy 不产生约束；

(4) 相同 Bank 的两条 Activate 指令之间，必须相隔 tRC 的时间。一般 tRC 约等于 $tRAS(min)+tRP$ ，这就是 $tRAS(min)$ 期限至作用时，Close Page Policy 的 Read Cycle，故只考虑 $tRAS(min)$ 就行了。但是，为了计算精确，也将其列入考虑范围；

(5) 不同 Bank 的两条 Activate 指令之间必须相隔 $tRRD$ 的时间，这个时间与 $tRCD$ 大致相同，对 Close Page Policy 不产生约束。

从前文 CL=3，BL=4 的波形图中，我们可以计算出 Autoprecharge 与 Activate 相隔的时钟周期数为： $(1+CtRCDm1+CL-1+BL-(CL-1))=(1+CtRCDm1+BL)$ ，如果这个值小于 $1+CtRASm1$ ，则自动 Precharge 会推迟以满足 $tRAS(min)$ 延时。

以 CL=3，BL=1 为例，画出整个读操作周期的时序图，以体现 $tRAS(min)$ 和 tRC 的影响：



故，修正后的理想读有效带宽公式如下：

$$BW_{effrd} = BW \times \frac{BL}{\max(1 + CtRCm1, \max(1 + CtRASm1, 1 + CtRCDm1 + 1 + BL) + CtRPm1)} \times (1 - \frac{(CtRFCm1 + 1) \times RowNum}{CtREF})$$

我们发现：

1. 读带宽利用率与 **CL** 不直接相关，故应选取较大的 **CL**，以换取较小的 **AC** 延时参数。
2. 与之前相同，选取较大的 **BL** 可以提高带宽利用率。
3. 选择较高的时钟频率，可以提高硬件带宽，但也可能导致命令延时对应的周期数增加，故需要综合考虑。

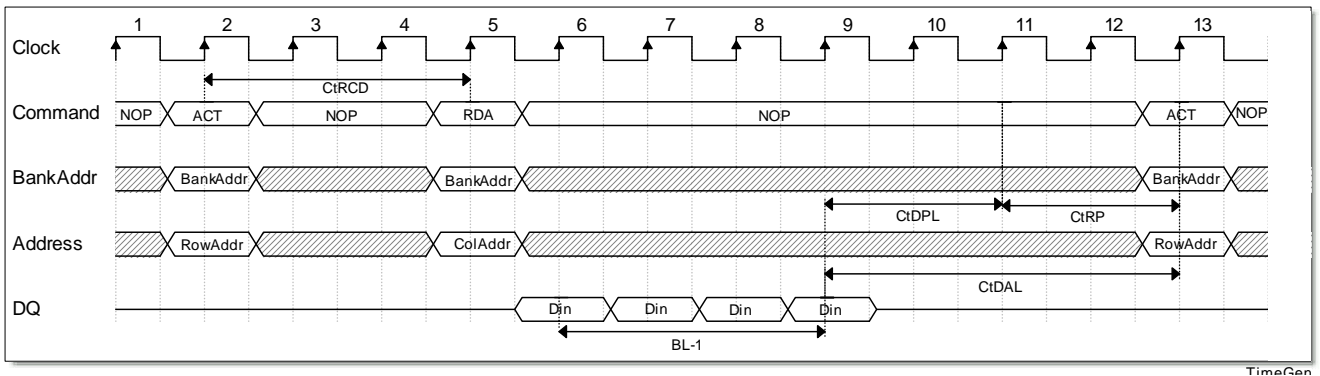
1.3.2.3 Activate & Write with Autoprecharge 操作与 Close Page Policy 的写效率分析

Close Page Policy 中，连续的写操作可以认为是 Activate 和 Write with Autoprecharge 操作的循环。

Activate 命令用于激活 Bank 中的一个 Row。发此命令的同时，要给出 Bank 地址和 Row 地址。Activate 命令后要隔 tRCD 才能发送读写指令。

Read with Autoprecharge 命令的作用是：写入数据，然后自动执行 Precharge 关闭 Row Buffer。发此命令的同时，要给出 Bank 地址和 Column 地址。Read with Autoprecharge 命令后，需要立刻传送数据，持续时间由 Burst Length 决定。自动的 precharge 发生在出现最后一个有效数据的时钟上升沿后的 CtDPL 个周期(也叫 tWR 延时)，之后 Precharge 延时参数 tRP 也必须被满足。有时候用 tDAL 标示 tDPL+tRP。

以 CL=3, BL=4 为例，画出整个写操作周期的时序图如下：



由于每次读写后都关闭 Row Buffer，故对于 Close Page Policy，写延时只由 Activate 延时决定，即 $1 + CtRCDm1 + 1$ ，即 $CtRCD + 1$ 。

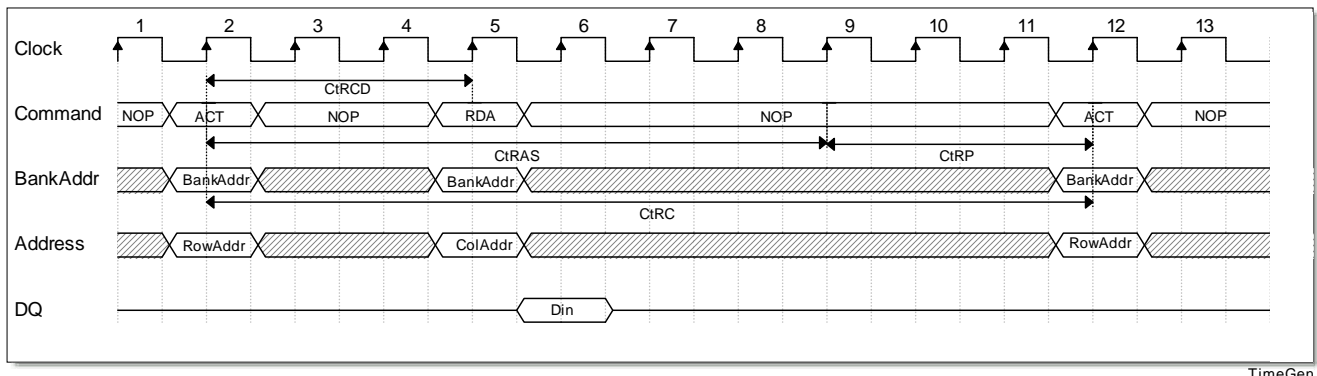
考虑到 Refresh 操作的影响，理想的写有效带宽公式如下：

$$BW_{effwr} = BW \times \frac{BL}{1 + CtRCDm1 + 1 + BL + CtDALm1} \times (1 - \frac{(CtRFCm1 + 1) \times RowNum}{CtREF})$$

另外，与读操作一样，写操作也有 tRC 和 tRAS 的制约。当 BL 较小时，如果 $(1 + CtRCDm1 + 1 + (BL - 1))$ 小于 Activate 与 Precharge 指令的最小间隔要求 $(1 + CtRASm1)$ ，则自动 Precharge 会推迟以满足 tRAS(min)

延时。

以 CL=3, BL=1 为例, 画出受 tRC 或 tRAS 制约的时序图:



故, 修正后的理想写有效带宽公式如下:

$$BW_{effwr} = BW \times \frac{BL}{\max(1 + CtRCm1, \max(1 + CtRASm1, 1 + CtRCDm1 + 1 + BL + CtDPLm1) + CtRPm1)} \times (1 - \frac{(CtRFCm1 + 1) \times RowNum}{CtREF})$$

我们发现:

1. 与之前相同, 选取较大的 BL 可以提高带宽利用率。
2. 选择较高的时钟频率, 可以提高硬件带宽, 但也可能导致命令延时对应的周期数增加, 故需要综合考虑。

1.3.3 Close Page Policy 性能计算实例

1.3.3.1 Samsung 64Mb: x16 单片 SDRAM 计算示例

1.3.3.1.1 系统设定

系统只是用一片 Samsung 64Mb: x16 的 SDRAM。系统时钟 166MHz, 要求 BL=4 或 8。SDRAM 速度等级为 166MHz, 若取 CL=3, 它的 AC 延时参数如下表:

tXX	tXX(ns)	CtXX(clks)	CtXXm1(clks)
tRCD	18	3	2
tRP	18	3	2
tRAS(min)	42	7	6
tWR(a.k.a. tDPL)	2clk	2	1
tDAL	2clk+tRP	5	5
tRC	60	10	9
tRFC	60	10	9
tRRD	12	2	1

次 SDRAM 的三位地址空间为：4bank * (2¹²)Row * (2⁸)Column * 16 bit = 64Mbit。需要在 64ms 内将 4096 行刷新一遍。

1.3.3.1.2 非流水线 Close Page Policy 性能计算

读延时 = CtRCD + CL = 3 + 3 = 6 clks

写延时 = CtRCD + 1 = 3 + 1 = 4 clks

回顾读带宽公式：

$$BW_{effrd} = BW \times \frac{BL}{\max(1 + CtRCm1, \max(1 + CtRASm1, 1 + CtRCDm1 + 1 + BL) + CtRPm1)} \times \left(1 - \frac{(CtRFCm1 + 1) \times RowNum}{CtREF}\right)$$

BL = 4 时：

$$BW_{effrd} = BW * 4/10 * (1 - 4096*60/64000000) = 39.85\% * BW$$

BL = 8 时：

$$BW_{effrd} = BW * 8/14 * (1 - 4096*60/64000000) = 56.92\% * BW$$

回顾写带宽公式：

$$BW_{effwr} = BW \times \frac{BL}{\max(1 + CtRCm1, \max(1 + CtRASm1, 1 + CtRCDm1 + 1 + BL + CtDPLm1) + CtRPm1)} \times \left(1 - \frac{(CtRFCm1 + 1) \times RowNum}{CtREF}\right)$$

BL = 4 时：

$$BW_{effwr} = BW * 4/11 * (1 - 4096*60/64000000) = 36.22\% * BW$$

BL = 8 时：

$$BW_{effwr} = BW * 8/15 * (1 - 4096*60/64000000) = 53.13\% * BW$$