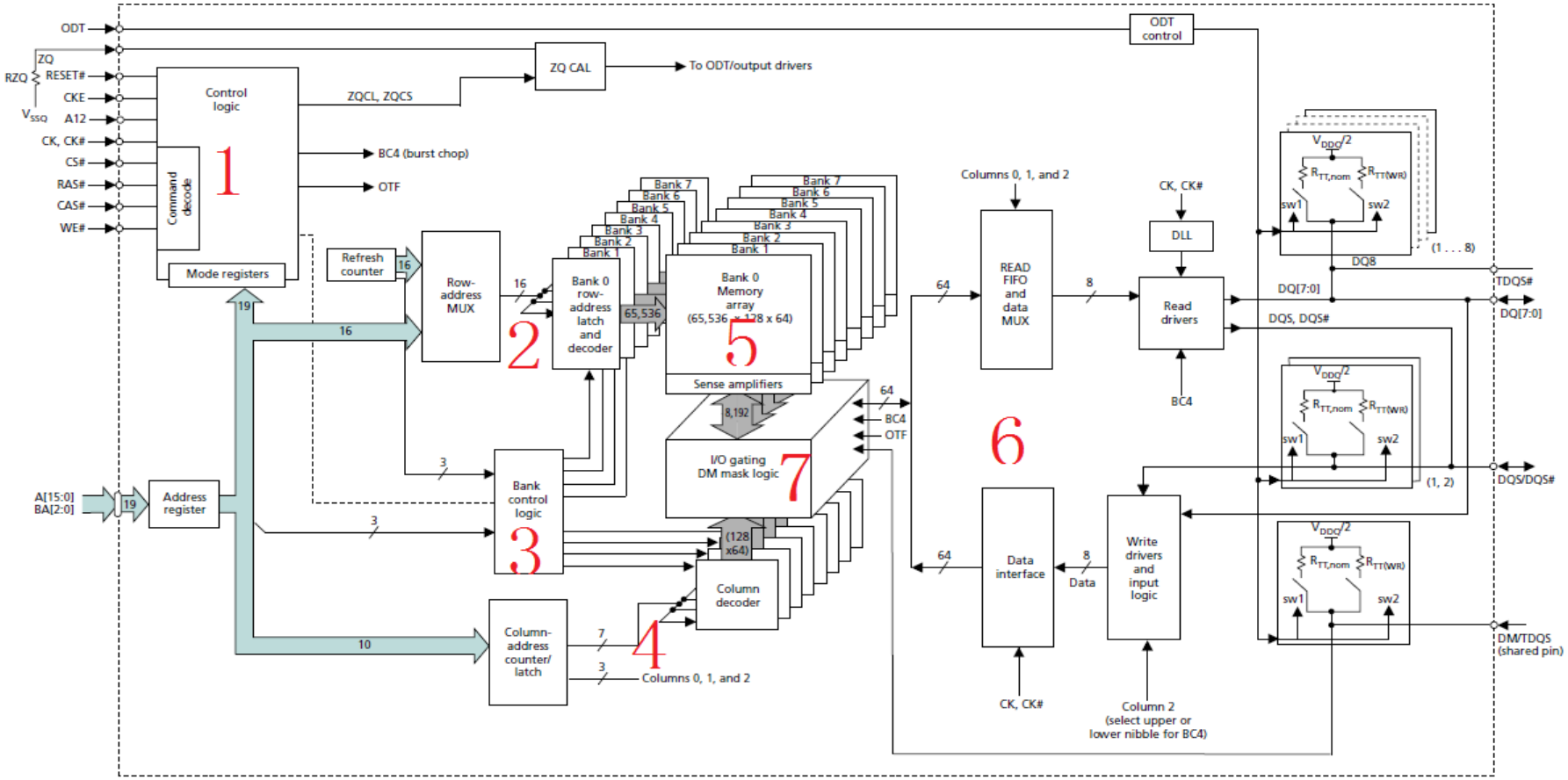
硬件十万个为什么
—— DDR3的工作原理

杜盼
dzplay@qq.com

目录

DDR的片内结构框图



上图是一片容量4Gb，数据位宽为8bit的DDR3 SDRAM的结构图

DDR的片内结构说明

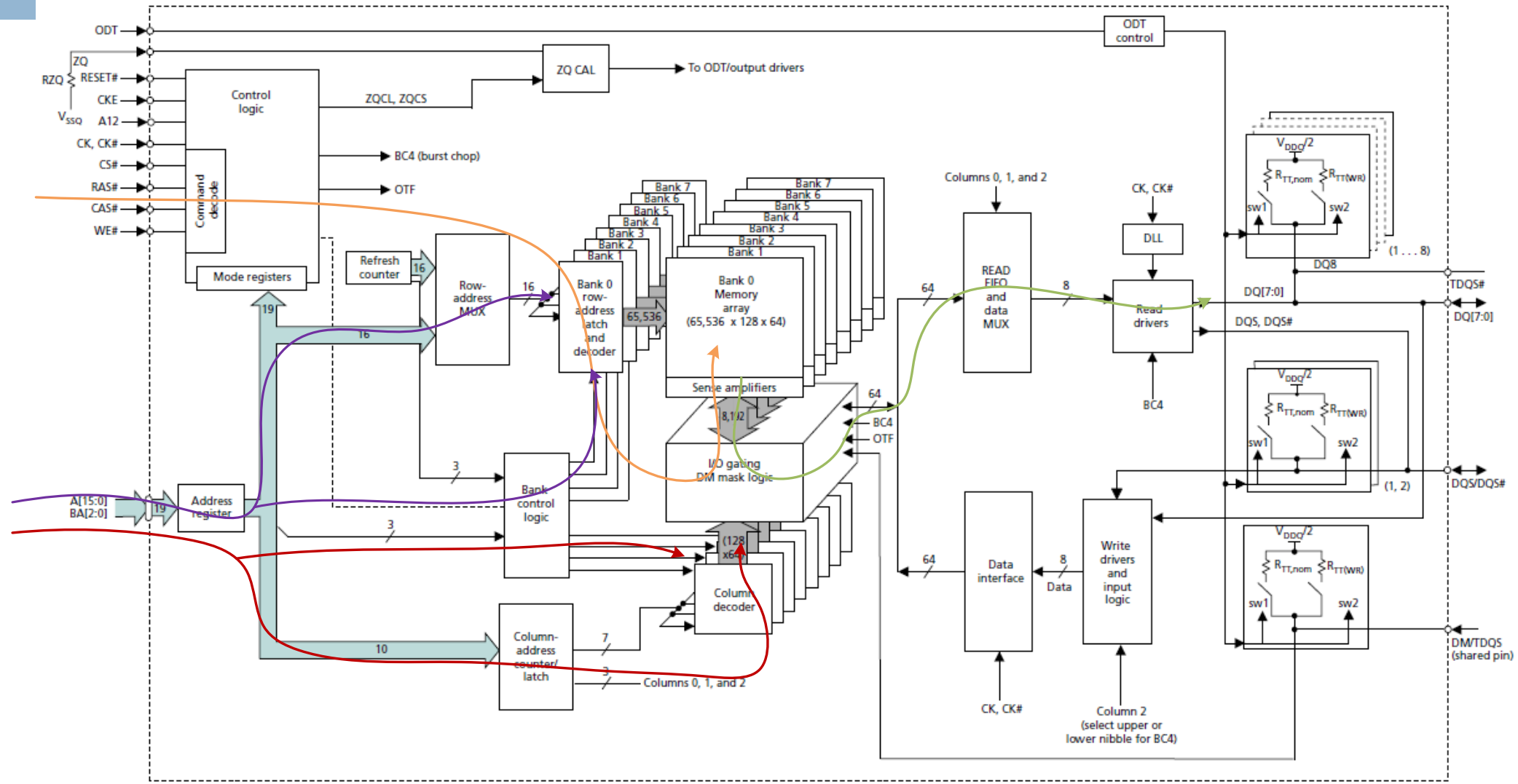
1. 控制器单元：包括输入命令解析，模式配置&控制部分；
2. 行地址选通单元：行激活通过此处操作；
3. Bank控制逻辑：行/列地址解码用到bank选通
4. 列地址选择单元，读写操作同时在打开列地址的时候送到1；
5. 内部存储阵列，此处分8个bank，已4g8bit的颗粒为例；每个bank分65536行，128列，每个cell存储8*bl的数据宽度；
6. 读写数据缓存及接口驱动；dq数据在此变换位宽后内外交互；
7. 锁存与控制逻辑：刷新与预充电用到该模块。

目录

DDR的操作

- 启动：上电->解复位->初始化->ZQCL-> LEVELING->IDLE(READY)
- **读：IDLE->行激活->读数据(1次或多次突发)->预充电->IDLE**
- **写：IDLE->行激活->写数据(1次或多次突发)->预充电->IDLE**
- 刷新：IDLE->REF->IDLE
- 自刷新的进入与退出：IDLE->SFR->IDLE
- 定期校正：IDLE->ZQCS->IDLE，一般外部温度或电压改变时操作
- 动态更改配置：IDLE->MRS/MPR->IDLE

DDR的读操作(内部流程图)

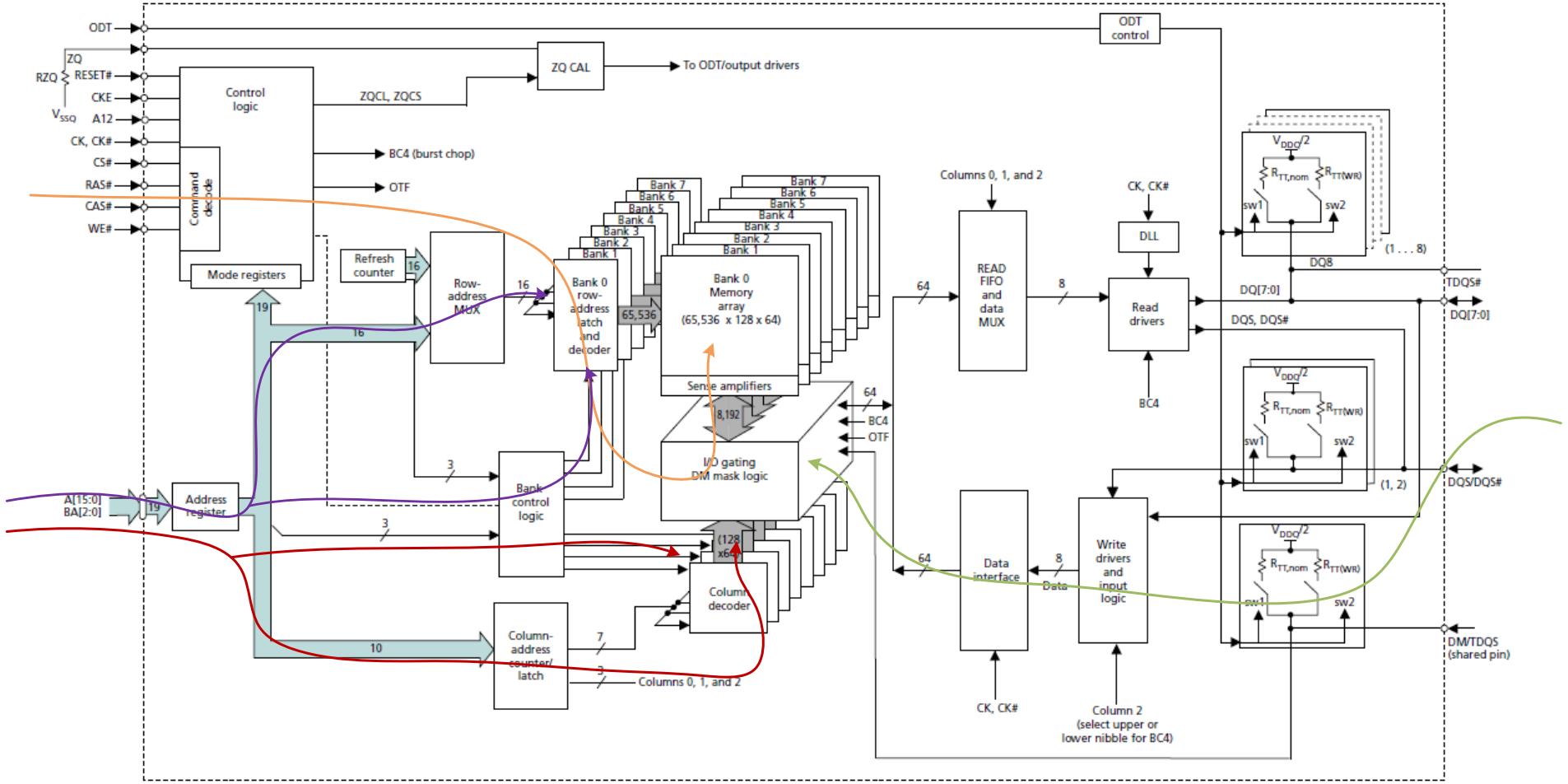


DDR的读操作(步骤说明)

- ddr发起一次读的过程包含一系列命令有：

1. 操作开始于Active命令——Active命令同时并发含带地址位，以选择Bank和Row地址（BA0-BA2选择BANK、A0-A15选择Row）。用于打开一个工作行。
2. 随后是一个Read命令——Read命令并发含带突发操作的起始Column地址和bank号，打开对应gating的列；
3. 数据经过特定的延时（CL+AL，后面详细介绍）；将数据传出IO；
4. 完成数据传出后，需将当前cache的数据刷回存储整列并关闭当前工作行（携带命令和BA信息）；【是否发布自动预充电命令（通过A10）】

DDR的写操作(内部流程图)



DDR的写操作(步骤说明)

- ddr发起一次写操作与读类似，除写数据，其他步骤参照读。

1. 和读操作一样。
2. 随后是一个write命令——Read命令并发含带突发操作的起始Column地址和bank号，打开对应gating的列；
3. 经过特定的延时（CWL+AL，后面详细介绍）；数据从IO写入io gating；这之间又有一个延时tWR。
4. 完成数据写入后，预充电的操作和读一样。

目录

行激活

- 初始化完成后，不管是读还是写，都需要对L-Bank中的阵列进行寻址，首先就要确定行（Row），使之处于活动状态（Active），然后再确定列。
- 虽然之前要确定L-Bank的定址，但它与行有效同时进行。

读命令

- 前面讲到的读操作是由一系列命令组成的，读命令是其必不可少的一部分（注意区分这2个定义的区别哈）
- 读命令包含命令本身，bank选择，和列地址进行寻址。
- 但是，地址线仍然是行地址所用的A0-A9（本例）。没错，在SDRAM中，行地址与列地址线是共用的。
- 不过，读/写的命令是怎么发出的呢？其实没有一个信号是发送读或写的明确命令的，而是通过芯片的可写状态的控制来达到读/写的目的。显然WE#信号就是一个关键。WE#无效时，当然就是读取命令。CS#,RAS#,CAS#,WE#：0101

写命令

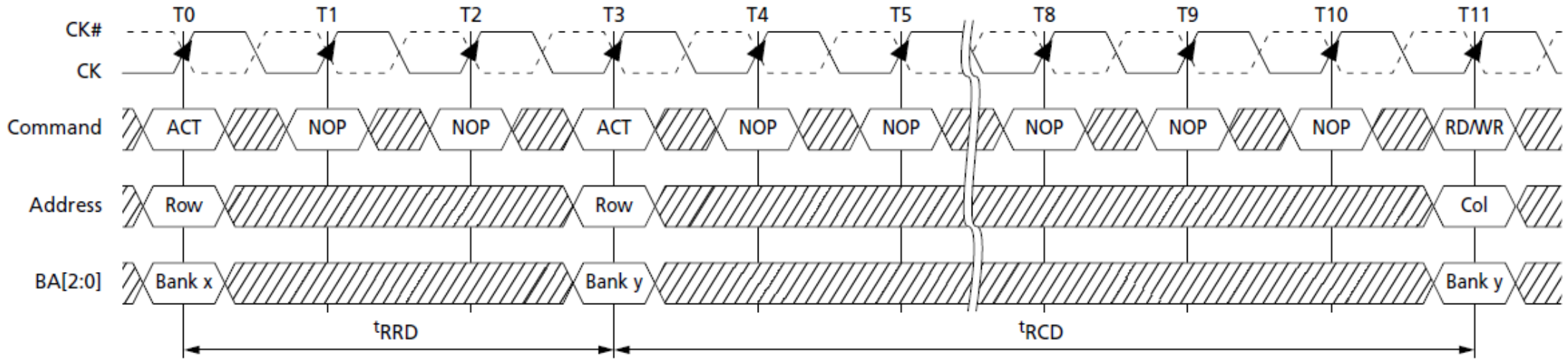
- 前面讲到的读操作是由一系列命令组成的，读命令是其必不可少的一部分（注意区分这2个定义的区别哈）
- 写命令包含命令本身，bank选择，和列地址进行寻址。
- CS#,RAS#,CAS#,WE#：0100；
- 需要注意的是，命令发出后，在固定的延时WL后要向DDR发送数据。

预充电

- 由于SDRAM的寻址具体独占性，所以在进行完读写操作后，如果要对同一L-Bank的另一行进行寻址，就要将原来有效（工作）的行关闭，重新发送行/列地址。
- L-Bank关闭现有工作行，准备打开新行的操作就是预充电（Precharge）。
- 预充电可以通过单独的命令控制，也可以通过辅助设定让芯片在每次读写操作之后自动进行预充电。（A10）
- 【本质】预充电是一种对工作行中所有存储体进行数据重写，并对行地址进行复位，同时释放S-AMP。

目录

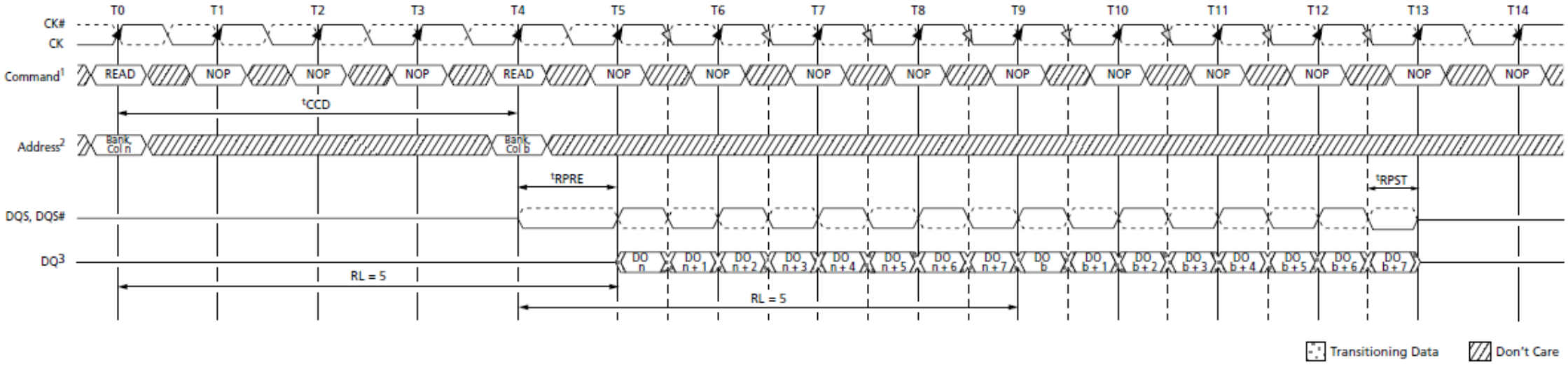
【时序】激活


T0
T1
T2
T3
T4
T5
T8
T9
T10
T11
CK#
CK
Command
ACT
NOP
NOP
ACT
NOP
NOP
NOP
NOP
NOP
RD/WR
Address
Row
Row
Col
BA[2:0]
Bank x
Bank y
Bank y
tRRD
tRCD

tRCD

- 在发送列读写命令时必须要与行有效命令有一个间隔，这个间隔被定义为tRCD，即RAS to CAS Delay（RAS至CAS延迟）；
- 大家也可以理解为行选通周期，这是根据芯片存储阵列电子元件响应时间（从一种状态到另一种状态变化的过程）所制定的延迟。

【时序】读操作BL


T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14
CK#
CK
Command[1]
READ
NOP
tCCD
Address[2]
Bank, Col n
Bank, Col b
tRPRE
tRPST
DQS, DQS#
DQ[3]
DO n
DO n + 1
DO n + 2
DO n + 3
DO n + 4
DO n + 5
DO n + 6
DO n + 7
DO b
DO b + 1
DO b + 2
DO b + 3
DO b + 4
DO b + 5
DO b + 6
DO b + 7
RL = 5
RL = 5
Transitioning Data
Don't Care

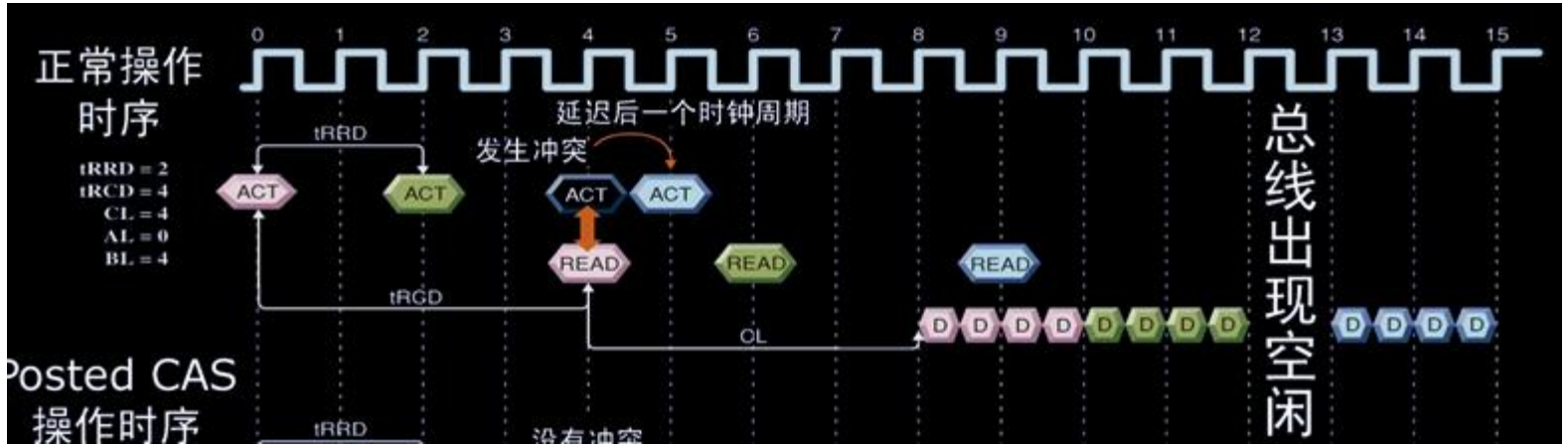
【参数】列选择延时——CL/CWL

- **CL**
- 但是在CAS发出之后，仍要经过一定的时间才能有数据输出，从CAS与读取命令发出到第一笔数据输出的这段时间，被定义为CL（CAS Latency，CAS潜伏期）。
- 由于CL只在读取时出现，所以CL又被称为读取潜伏期（RL，Read Latency）。
- CL的单位与tRCD一样，为时钟周期数，具体耗时由时钟频率决定。
- **CWL**
- 和读数据类似，在CAS发出之后，仍要经过一定的时间才允许数据进入，从CAS与写命令发出到第一拍数据写入的这段时间，被定义为CWL（CAS Write Latency）

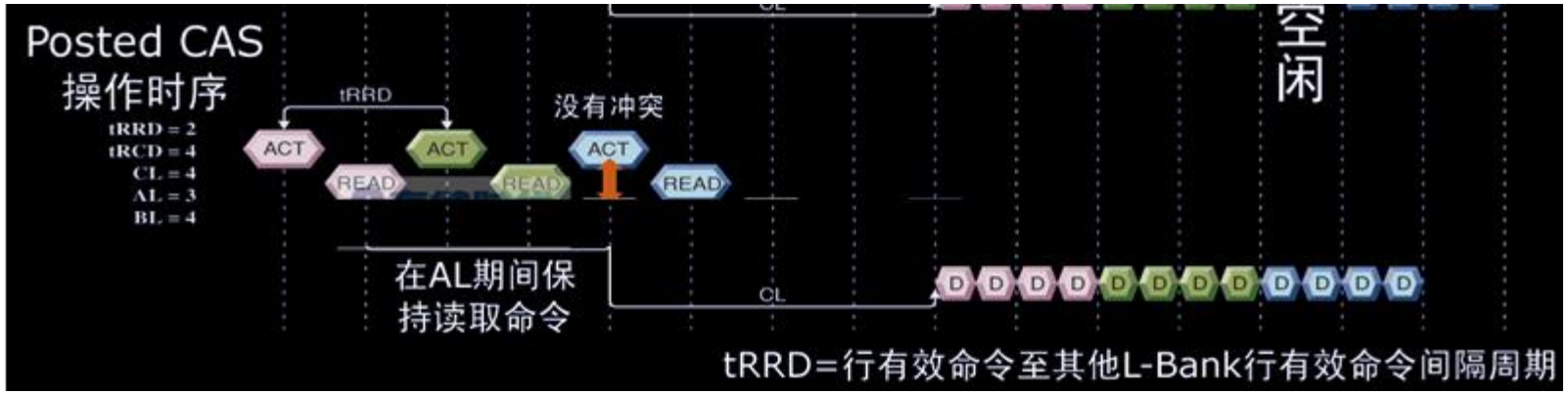
【参数】附加延时——AL

- (为啥会有这个东西，为啥可以是0)
- （为了解决DDR内存中指令冲突而设计的功能。它允许CAS信号紧随RAS发送，相对于以往的DDR等于将CAS前置了。这样，地址线可以立刻空出来，便于后面的行有效命令发出；读/写操作并没有因此而提前，仍有要保证有足够的延迟/潜伏期）
- DDR的2种读操作方式interleaved与Sequential；interleaved顾名思义是交错操作，也就是同时打开多个bank； Sequential是顺序操作，一次只操作1个bank；开启AL可在交错操作时减少冲突率而进一步提高效率。

【参数】附加延时——AL（disable）



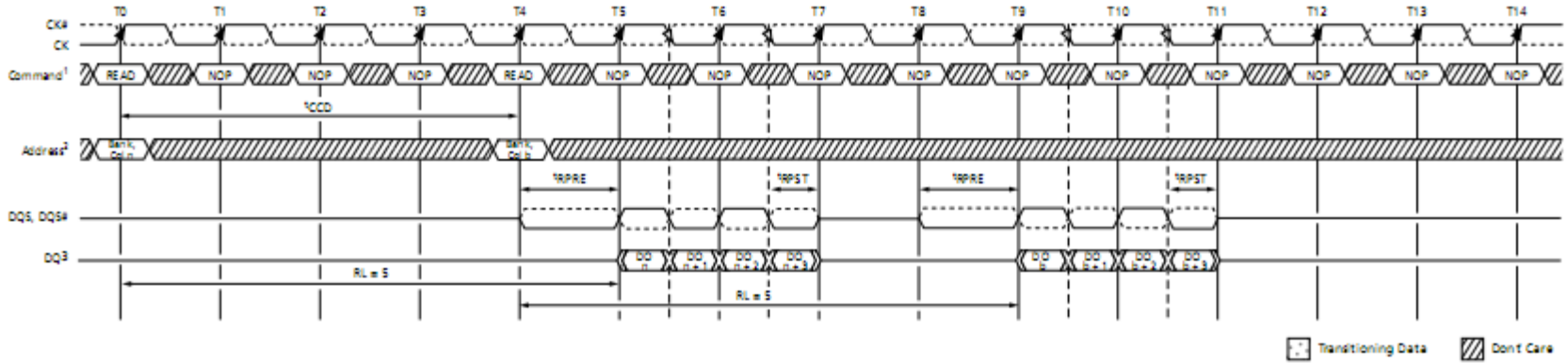
【参数】附加延时——AL（enable）



【参数】附加延时——AL（内部操作）

【时序】读操作BC

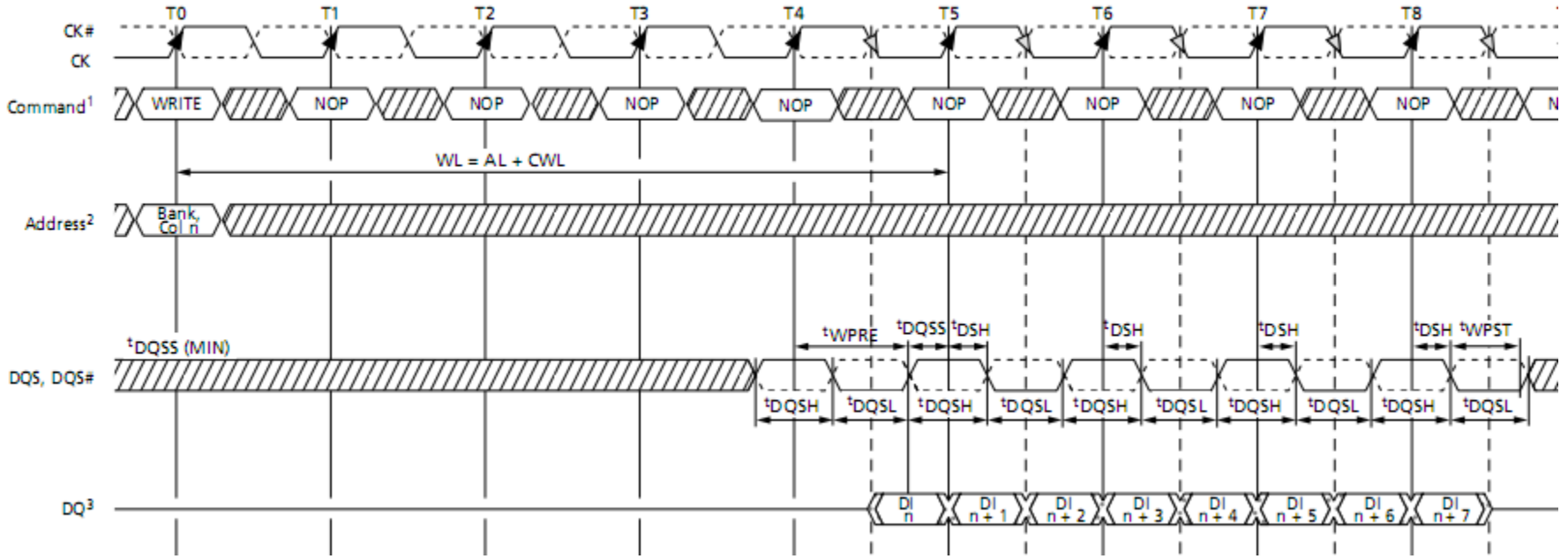
Figure 69: Consecutive READ Bursts (BC4)



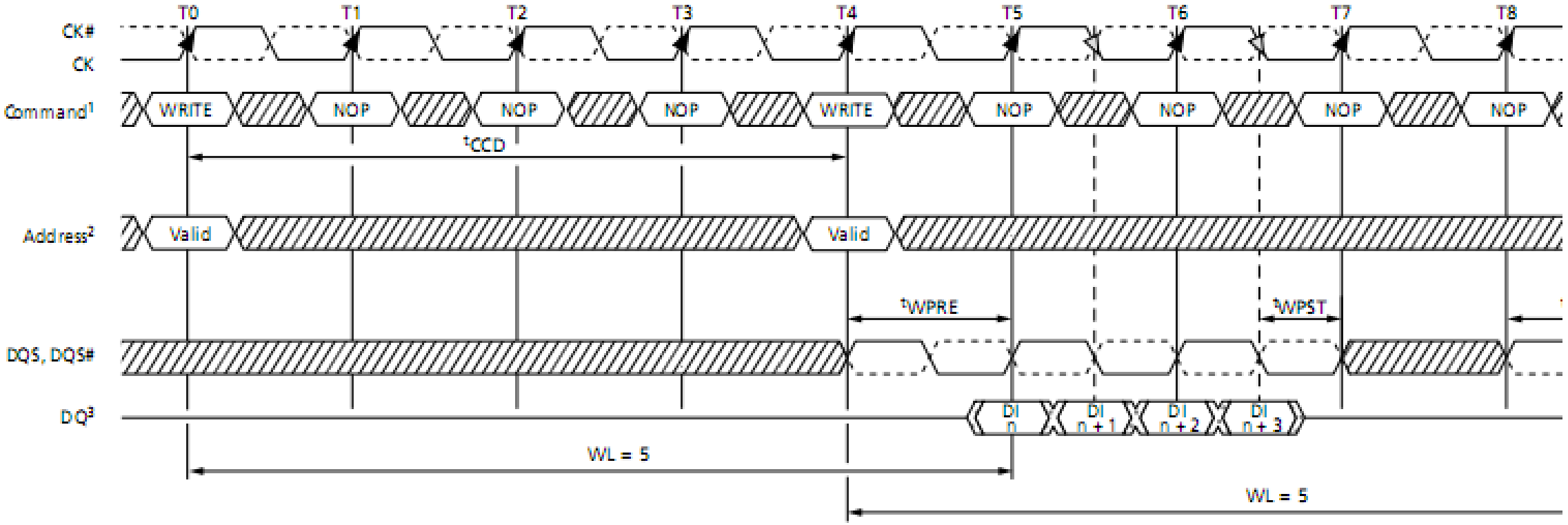
BL与BC

- 由于DDR3的预取为8bit，所以突发传输周期（Burst Length，BL）也固定为8
- 而对于DDR2和早期的DDR架构系统，BL=4也是常用的，DDR3为此增加了一个4bit Burst Chop（突发突变）模式
- 即由一个BL=4的读取操作加上一个BL=4的写入操作来合成一个BL=8的数据突发传输，届时可通过A12地址线来控制这一突发模式。

【时序】写操作BL


T0
T1
T2
T3
T4
T5
T6
T7
T8
CK#
CK
Command[1]
WRITE
NOP
WL = AL + CWL
Address[2]
Bank, Col n
tDQSS (MIN)
DQS, DQS#
tWPRE
tDQSS
tDSH
tWPST
tDQSH
tDQSL
DQ[3]
DI n
DI n + 1
DI n + 2
DI n + 3
DI n + 4
DI n + 5
DI n + 6
DI n + 7

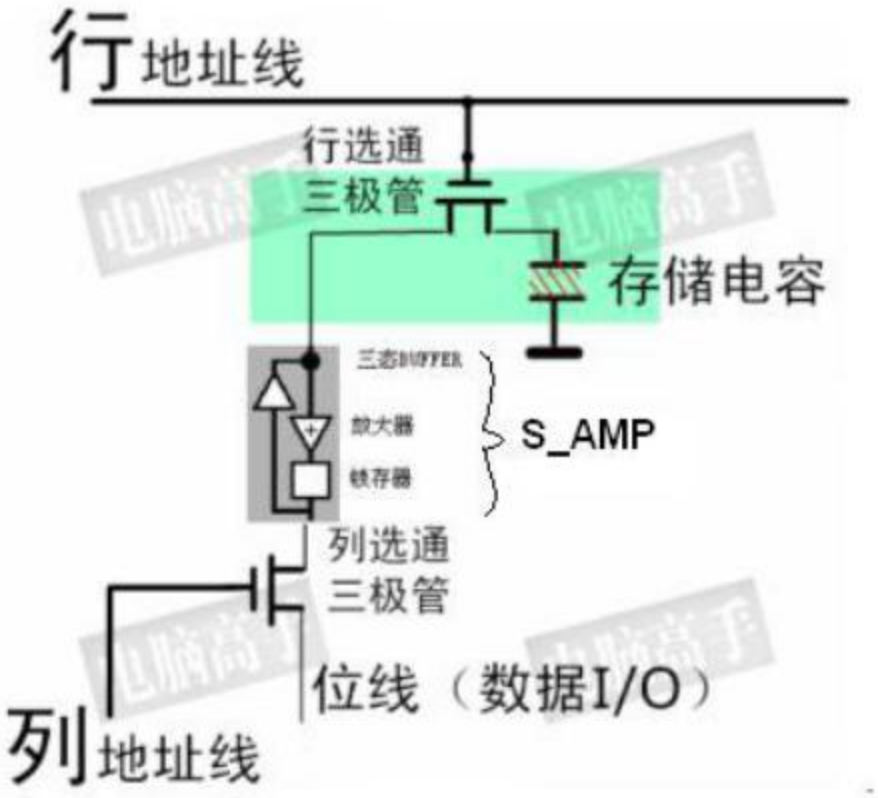
【时序】写操作BC


T0
T1
T2
T3
T4
T5
T6
T7
T8
CK#
CK
Command[1]
WRITE
NOP
NOP
NOP
WRITE
NOP
NOP
NOP
NOP
tCCD
Address[2]
Valid
Valid
tWPRE
tWPST
DQS, DQS#
DQ[3]
DI n
DI n + 1
DI n + 2
DI n + 3
WL = 5
WL = 5

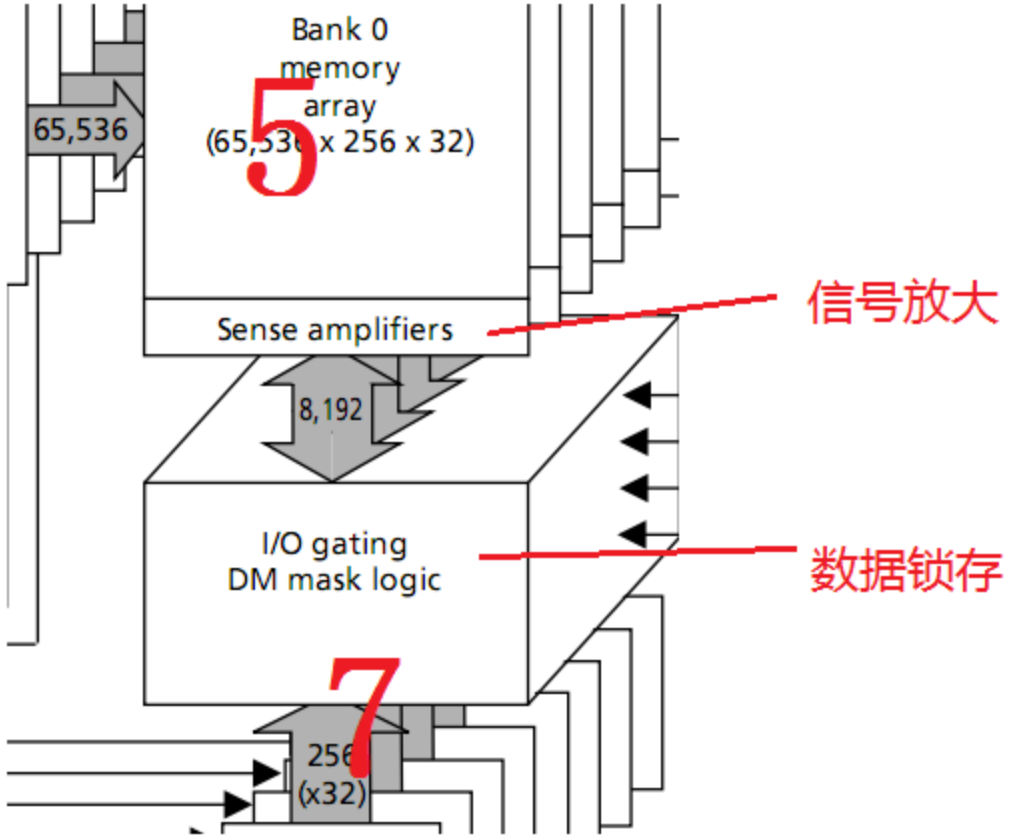
刷新与预充电

- 刷新操作与预充电中重写的操作一样，都是用S-AMP先读再写。因为它要不断进行刷新（Refresh）才能保留住数据，因此它是DRAM最重要的操作。
- 预充电是对一个或所有L-Bank中的工作行操作，并且是不定期的，而刷新则是有固定的周期，依次对所有行进行操作，以保留那些久久没经历重写的存储体中的数据。

感应放大器的读写（结构图1）



感应放大器的读写（结构图2）



感应放大器的读写（说明）

- 从第二张图中可以看到，每个bank的S-AMP读写整行的数据；对应有独立的存储单元。
- 当行激活时，数据从5到7；S-AMP是通过一个参考电压与存储体位线电压比较，来判断逻辑值。然后将值写入gating处锁存。后面依据读写命令来决定是将锁存的数据输出还是更新。
- 当要写回时，数据从7到5；对逻辑1单元，进行重写（充电），否则为逻辑0，不进行重写。

Thank you

扫二维码关注　硬件十万个为什么