

## 从内部结构分析SDRAM到DDR的变化

基于FPGA的DDR相关知识导航界面，[点击查看](#)。

## 1、结构概述

在此之前，曾经通过一篇文章从SDRAM的内部芯片框图出发，分析过SDRAM的功能实现，本文开始继续分析DDR、DDR2、DDR3 的芯片内部框图，从而认识他们各自的区别，便于后续使用。

下图是镁光的128Mb的SDRAM内存芯片，前文已经对该框图做了详细讲解。

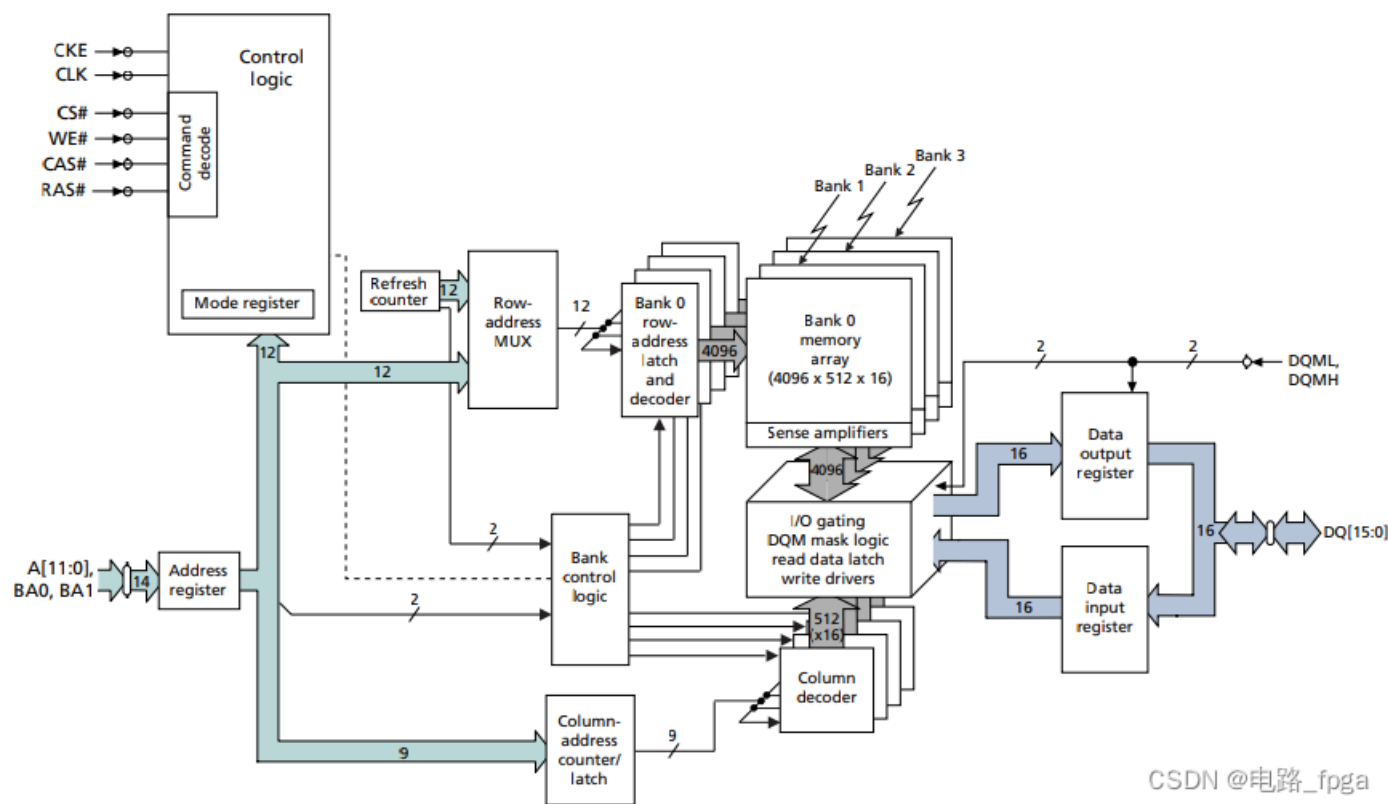


图1 SDRAM框图

在 **计算机** 运算速度发展的过程中，需要提高内存的读写速率，只能通过提高时钟频率来提高SDRAM的读写速率。由于温度等因素的影响，SDRAM的内核时钟频率受限，无法进一步提升。SDRAM只在时钟上升沿传输数据，为了提高数据传输速率，DDR的IO端口在时钟的上升沿和下降沿都传输数据，而内核还是只在时钟上升沿传输数据。为了匹配内核（上升沿传输）和IO端口（双沿传输）数据传输的速率，内核的数据位宽变为IO数据位宽的2倍，这就是DDR器件的设计原理。

下图是镁光DDR芯片的内部框图，型号为M146V32M16P-6T。该芯片数据位宽也是16位，容量为512Mb，忽略容量大小，两者的区别在于红框部分，DDR采用差分时钟信号。

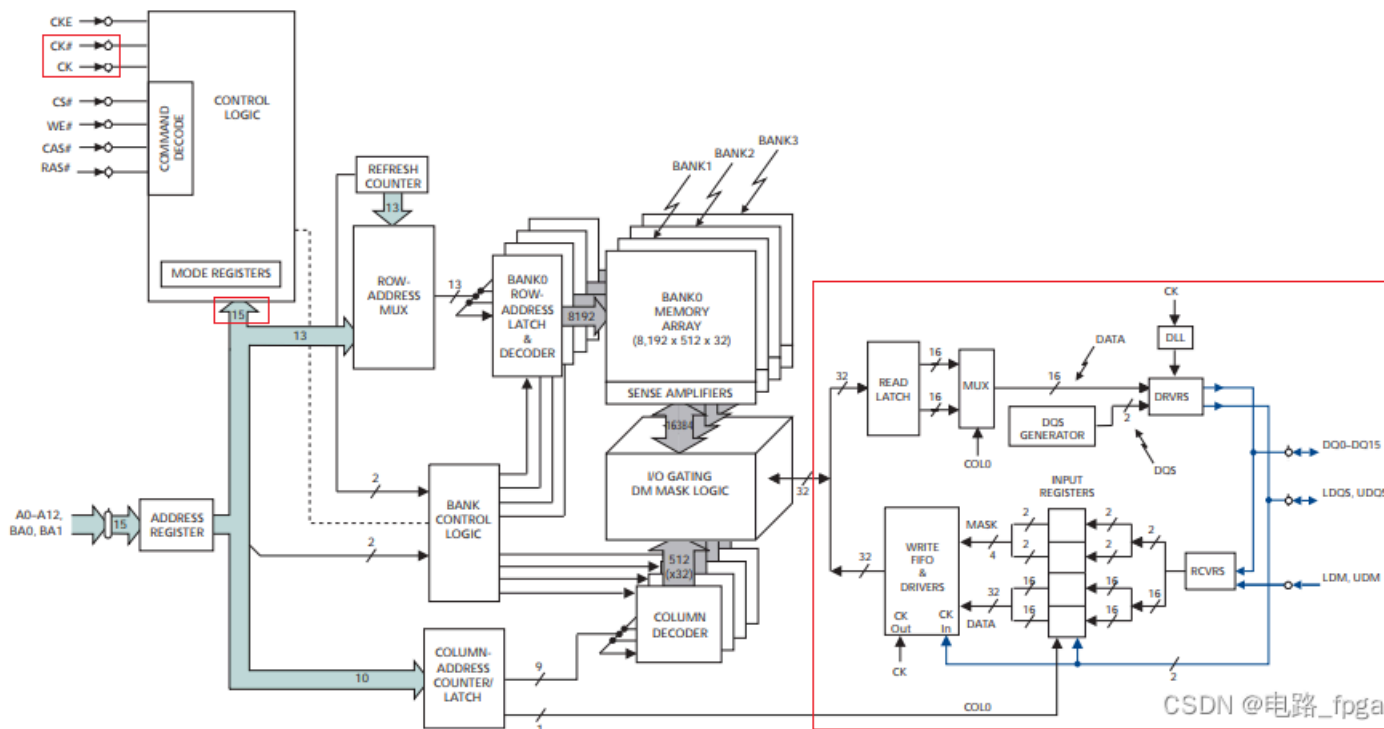


图2 DDR内部结构框图

在数据写入过程中，接收器（RCVRS）接收到的数据（DQ0~15）和数据掩膜信号（LDM/UDM）被暂存于接收寄存器中。该寄存器的功能类似于IDDR，它将DQS双沿传输的16位数据转换为DQS单沿传输的32位数据。同时，接收寄存器将2位的DM信号转换为4位MASK信号。

随后，这32位的DQS单沿数据和4位的MASK信号被送入写FIFO。写FIFO的作用是处理异步时钟，将DQS时钟下的DATA和MASK信号同步到内核时钟CK下，确保数据在正确的时序下被写入内部存储阵列。

注意输入寄存器和写FIFO的输入端都是把DQS作为数据的时钟信号，因为该信号的边沿与写入数据DQ和数据掩膜信号DM的中心对齐。

此处可能会有人有疑问，DQ不应该是在CK的双沿下传输的，为什么这里是在DQS的双沿采集DQ的数据？

DQS是数据选通信号，在读数据时，DQS的边沿与DQ的边沿对齐，在写数据时，DQS的边沿与写数据DQ的中心对齐。所以写数据的时候在DQS的边沿采集DQ的数据就比较稳定。

在读取时，L-Bank 在内部时钟信号的触发下，一次传送 32位的数据给读取锁存器（READ LATCH），再分成两路16位数据传给数据选择器(MUX)，数据选择器(MUX)选择其中一路，然后由发送器(DQS GENERATOR)在DQS的控制下在外部时钟上升、下降沿分两次传输16位的数据给主控芯片。

如果时钟频率为100MHz，那么在I/O端口处，由于在时钟上升、下降沿传输数据，那么传输频率就是200MHz。现在基本明白DDR SDRAM的工作原理了吧，这种内部存储单元容量（芯片内部总线位宽）= 2×芯片位宽（芯片I/O总线位宽）的设计，就是所谓的两位预取（2-bit Prefetch），有的公司称之为 2-n Prefetch（n代表芯片位宽）。

上述框图其实还有一个区别，SDRAM的框图中，12根地址线作为模式寄存器的数据信号，而DDR框图中，却有15根线连接到模式寄存器，其中13根是地址线，2根bank地址线，这是因为DDR除了模式寄存器(MR)，还增加了一个扩展模式寄存器（EMR），配置时bank地址线作为MR和EMR的地址线，区分A12~A0的数据是配置MR还是EMR的数据。

## 2、功能区别

DDR SDRAM和SDRAM的很多功能和指令都是一样，比如激活指令，自动预充电等指令，但是功能也有一些区别，本文不对两者相同的地方做讲解了，仅对不同的地方进行讲解。

### 2. 1、时钟

SDRAM采用单个时钟信号，而DDR采用差分时钟，CK#起到触发时钟校准的作用。由于数据在时钟CK的上升、下降沿传输，传输周期缩短了一半，因此必须要保证传输周期的稳定以确保数据的正确传输，这就要求CK的上升沿与下降沿的间距要精确控制。

由于温度、电阻性能的改变等影响，CK上升沿与下降沿的间距可能发生变化，此时与其反相的CK#就起到纠正的作用（CK上升快下降慢，CK#则是上升慢下降快）。

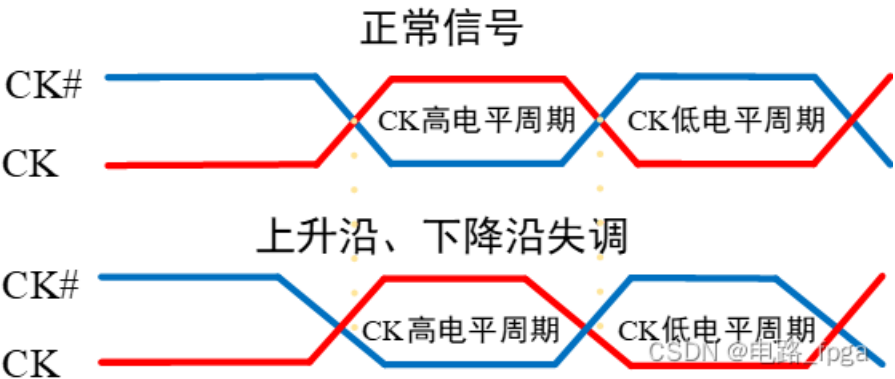


图3 时钟信号

在CK与CK#交叉时采集数据就能解决上述间距变化问题，这就是CK#的纠正思路。

由于数据在时钟的上升沿和下降沿传输，CL可以取1.5和2.5。数据DQ相关信号在时钟的双沿传输，而指令相关信号依旧是在时钟上升沿传输。

2.2、模式寄存器和初始化时序

DDR与SDRAM一样，在开机时需要进行MRS（模式寄存器设置），因为DDR的操作功能增加，相比SDRAM多了一个EMRS阶段（扩展模式寄存器设置），这个扩展模式寄存器控制着DLL的有效、输出驱动强度、QFC有效等。

下图是SDRAM的模式寄存器，各位的含义前文已经做过详细讲解，此处就不再赘述。

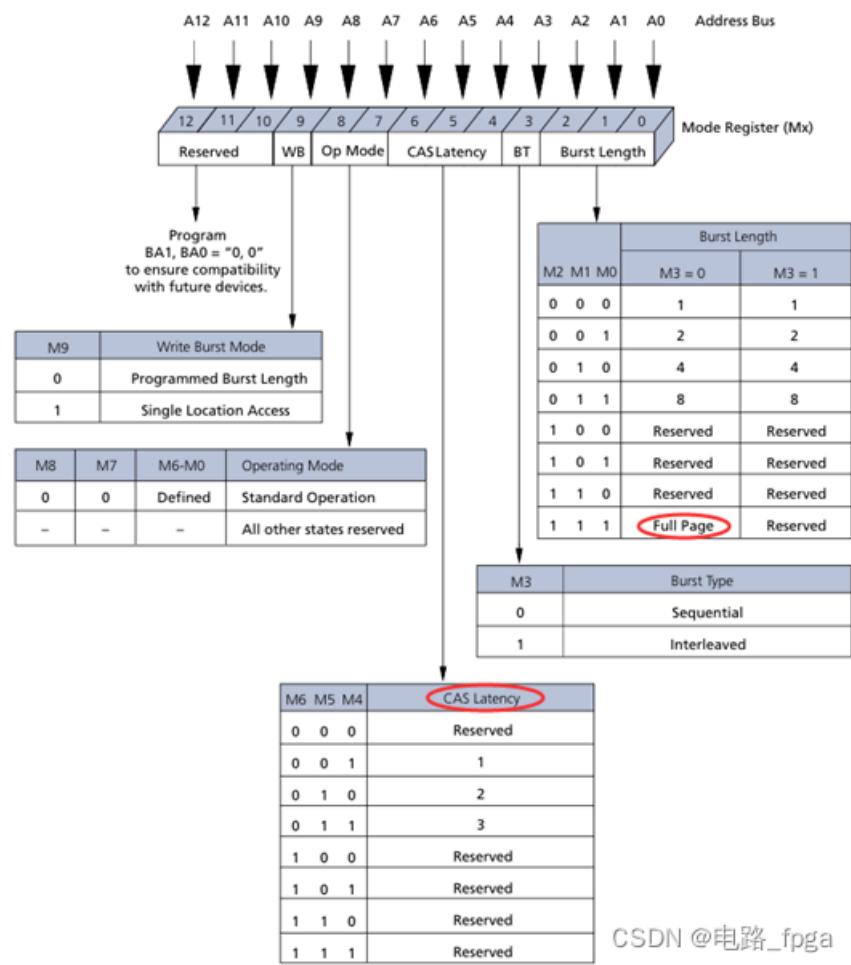


图4 SDRAM模式寄存器

下图是DDR的模式寄存器，模式寄存器的内容相对SDRAM也有一些变化，突发长度不支持全页突发了，CL的长度也发生了一些变化。

DDR中存在MRS和EMRS，那如何分辨这两个寄存器的呢？

在配置时，可以理解为在配置模式寄存器时，bank地址线就是模式寄存器的地址线。当BA1~BA0为00时，此时地址线An~A0的内容就是配置模式寄存器，为01时配置扩展模式寄存器。

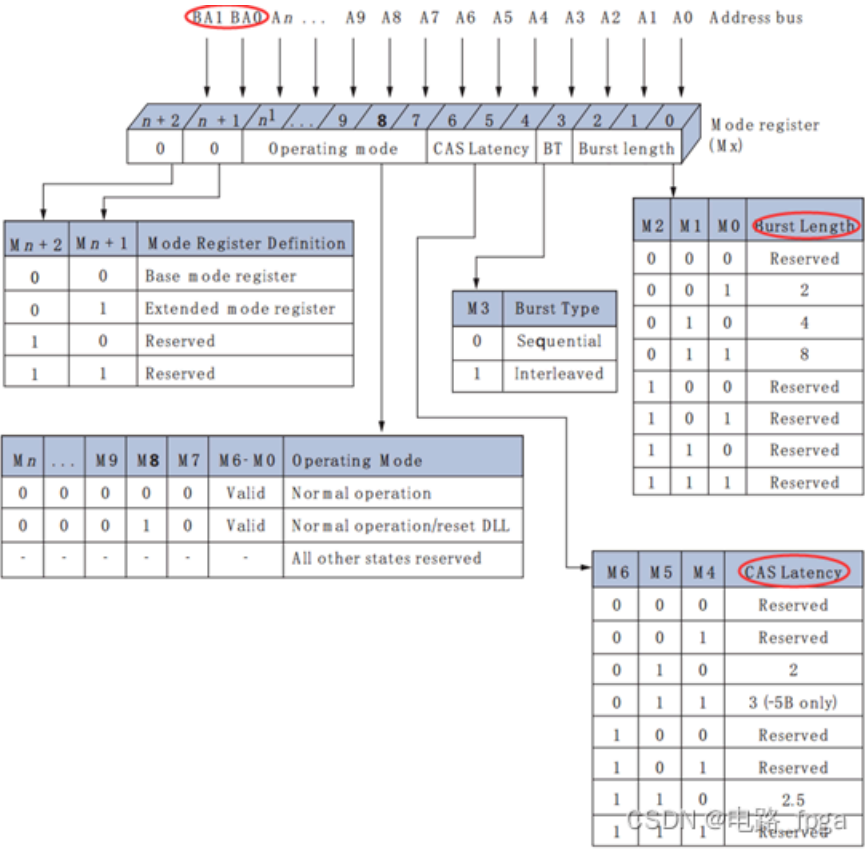


图5 DDR模式寄存器

下图是该芯片的扩展模式寄存器，包含DLL的启用/禁止，QFC的启用/禁止，其中DLL的作用在后文会讲解。

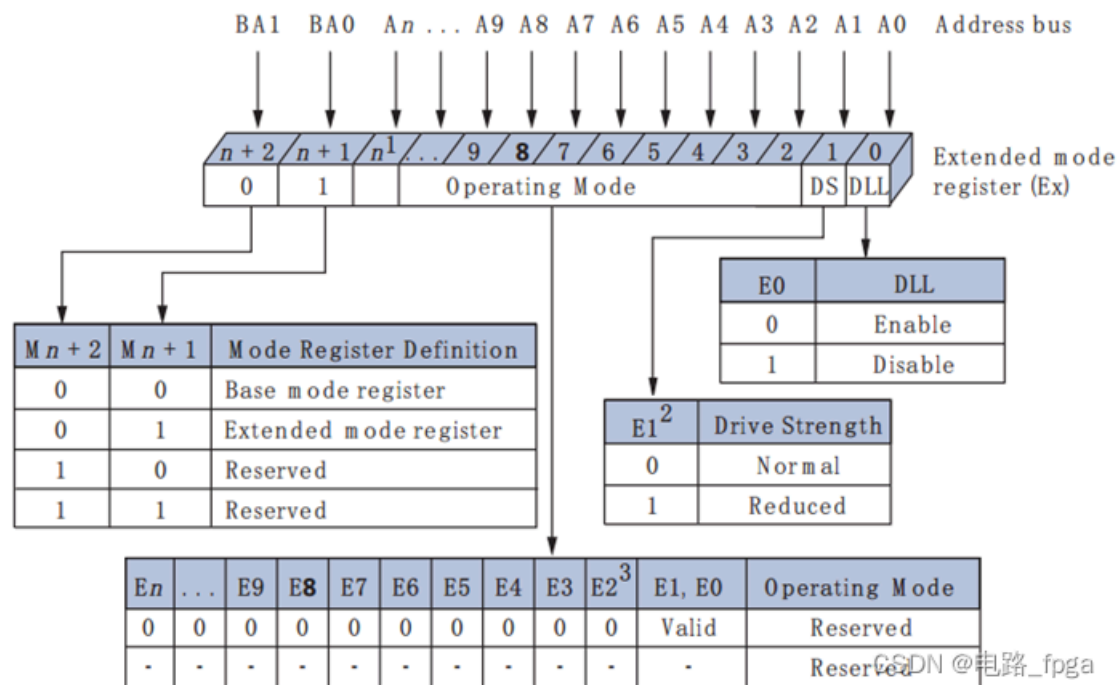


图6 DDR扩展模式寄存器

QFC是指FET Switch Controllar (FET开关控制)，低电平有效。用于借助外部FET开关控制内存模组上芯片在没有读、写操作时进入隔离状态，以确保芯片间不受相互干扰。

QFC是一个特选功能，厂商在接到芯片买家的指定要求后，才在芯片中加入此功能，并且需要在模组装配时进行相关的设计改动（如增加V<sub>ddQ</sub>的上拉电阻），所以市场上很少见到支持这一功能的DDR。在JEDEC最新发布的DDR规范中，已经不在有QFC的定义，这颗芯片的手册中也表示不支持QFC功能，作为了解即可。

既然DDR增加了扩展模式寄存器，那相应的初始化时序也相应的有所改变。相比SDRAM多了一个配置扩展模式寄存器的状态，初始化时序图如下所示，当上电电源稳定后，经过200us延时后把时钟使能，然后对所有bank进行预充电，之后依次配置扩展模式寄存器、模式寄存器，之后再次预充电，最后进行两次自刷新操作，就可以进入正常工作状态了。

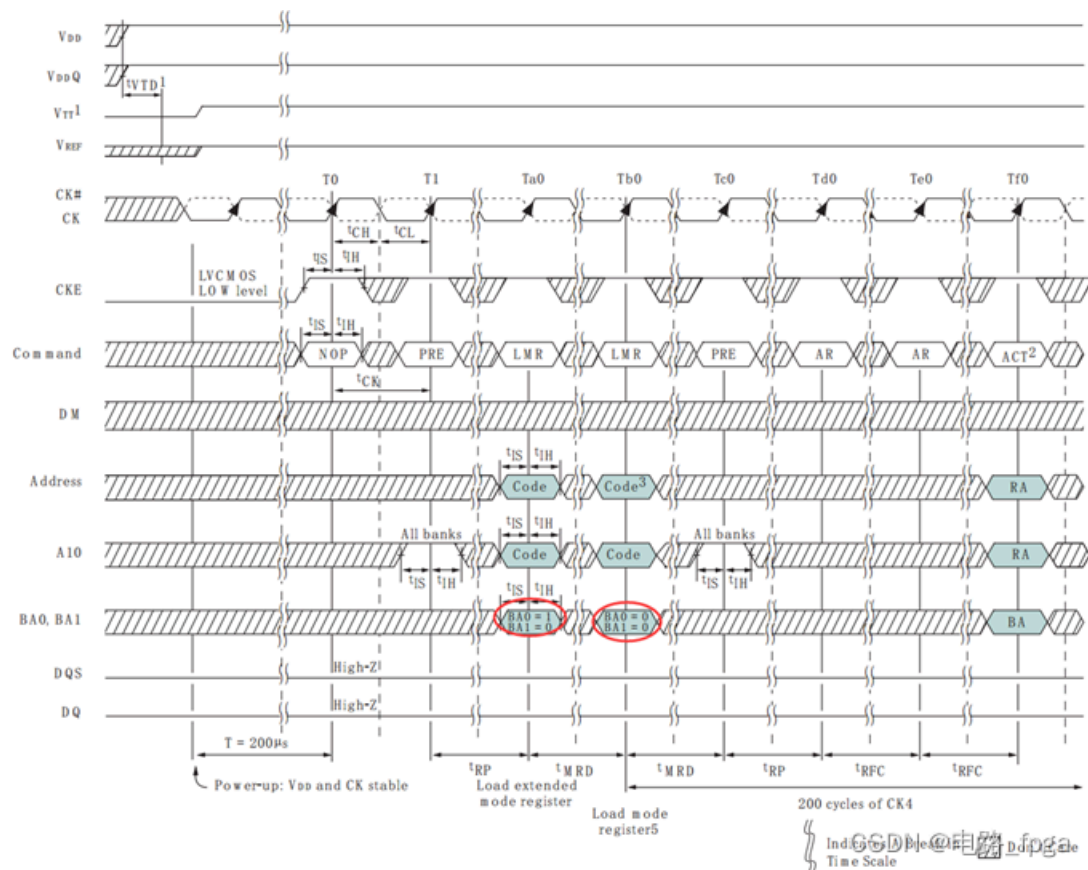


图7 DDR初始化时序

## 2.3、突发长度与写入掩码

由前文知SDRAM突发读写的长度有1、2、4、8、全页，而DDR的突发长度只有2、4、8。注意SDRAM突发长度是指连续寻址的存储单元数量，而DDR的突发长度是指连续的传输数据个数，每个数据宽度就是芯片IO的数据宽度。

因为DDR每次存储两倍芯片IO宽度的数据，芯片每次至少传输两次数据，所以不存在长度为1的读写操作。全页突发在图像领域可能会比较常用，但是在PC内存中一般不会被使用，毕竟DDR当时主要被用作PC内存，所以被砍掉了。

DDR在突发写入数据时，如果其中有不想存入的数据，可以像SDRAM一样使用掩膜信号DM进行屏蔽。通过下图知，掩膜信号DM与数据信号DQ同时发出，DDR芯片在DQS的上升沿和下降沿来检测DM的状态，如果DM为高电平，则此时的数据DQ就被丢弃了，不会写入DDR内部。



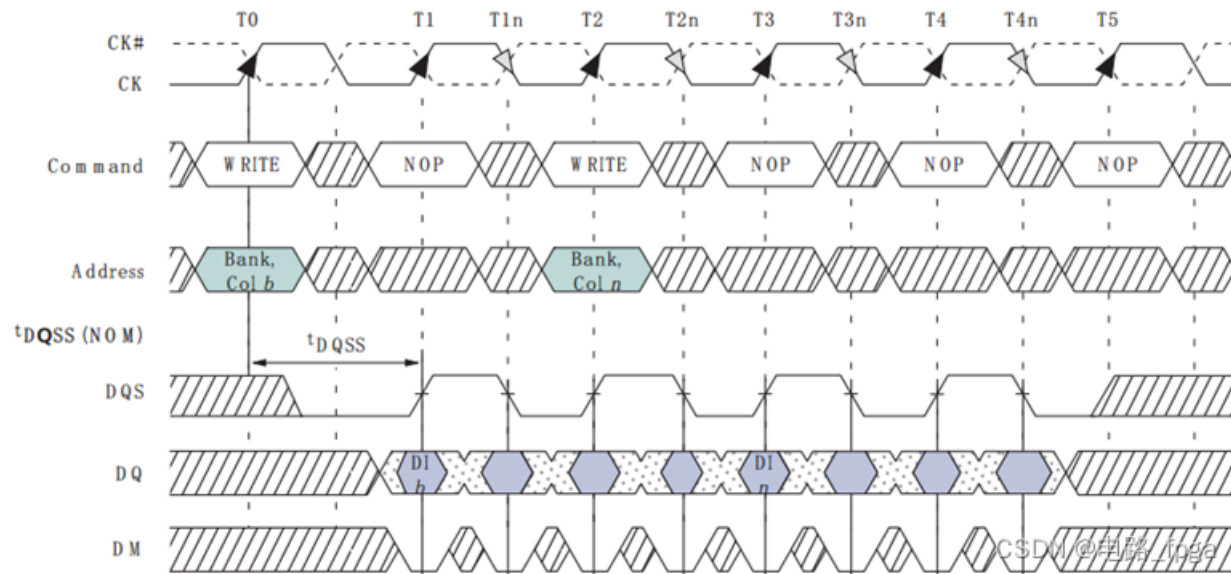


图8 DDR写时序

SDRAM对于读操作也是有掩膜信号的，但是DDR读操作就没有掩膜功能了，原因在于需要读哪个数据是由主控芯片控制的，如果不想读哪个地址的数据，完全可以主控芯片自己控制。

## 2.4、延迟锁定回路 (Delay Locked Loop)

这部分的功能讲解来自“电脑高手”的一篇文章，鉴于别人的文章内容，图片中的LOGO保留，讲的很简单，也推翻了很多网络上对DDR中DLL的错误理解。

DDR对时钟精确性要求极高，DDR有两个时钟：外部总线时钟和内部工作时钟。理想情况下，这两个时钟应保持同步，但由于温度和电压波动等因素，实际中很难实现完美的同步。此外，时钟频率本身的不稳定性也增加了同步的难度。SDRAM也有内部时钟，由于SDRAM工作频率较低，其内部时钟与外部时钟的同步问题不那么显著。

在实际应用中，时钟不同步可能表现为正向或负向的延迟。为应对这种情况，可以设定一个固定的延迟值，例如一个时钟周期，那么内外时钟的上升沿和下降沿仍然保持同步。由于外部时钟周期并非绝对一致，**DDR需要动态调整内部时钟的延迟，以匹配外部时钟，这一过程由延迟锁定环路（DLL）负责**，它能够实时监测并调整内部时钟，确保与外部时钟的精确同步。

**DLL不涉及频率与电压转换，而是生成一个延迟量给内部时钟。** 目前DLL有时钟频率测量法（CFM）和时钟比较法（CC）两种实现方法。

CFM（Clock Frequency Monitor）负责监测外部时钟的频率，并据此调整内部时钟的延迟，确保内外时钟之间仅相差一个周期，实现同步。DLL通过不断测量和调整延迟值，动态地维持内部时钟与外部时钟的精确同步。CFM的校正速度快，仅用两个时钟周期，但容易受到噪音干扰，如果测量失误，则内部的延迟就会永远错下去。



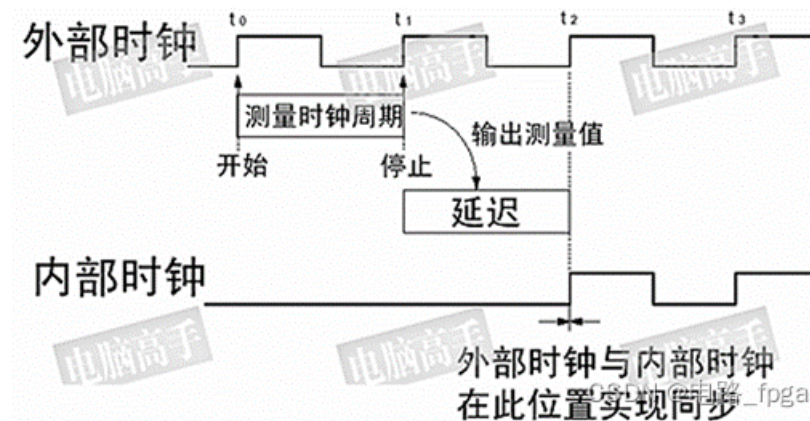


图9 时钟频率测量法

CC (Clock Correction) 方法通过比较内外时钟周期的差异来调整内部时钟。当检测到内部时钟周期短于外部时钟时，会在下一个内部时钟周期中补充缺失的延迟，反之则减少多余的延迟。这一过程不断循环，直至内外时钟达到同步。

CC的优势在于其稳定性和可靠性，即使在比较过程中出现偏差，也仅影响单个数据点，不会对后续的延迟修正产生连锁反应。然而，CC的调整过程相对较慢，需要更多时间来完成同步。

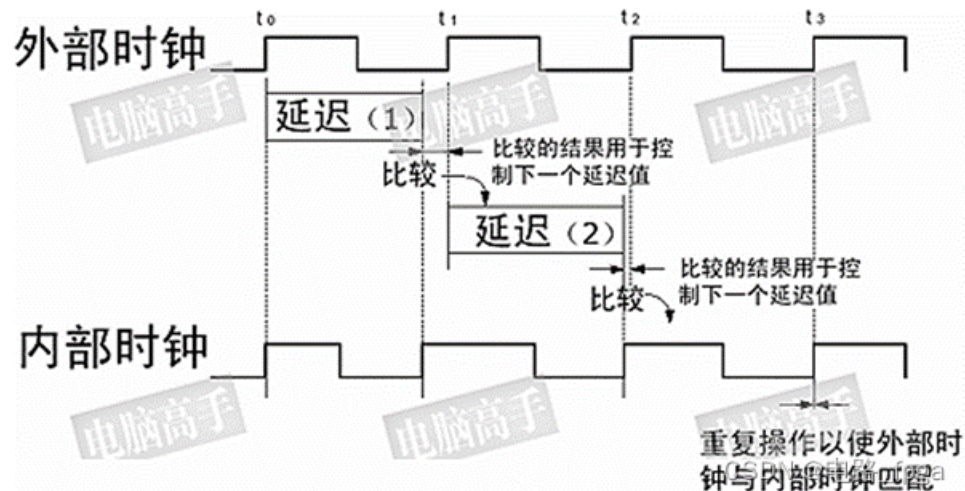


图10 时钟比较法

DDR在排错与评估操作时可以禁用DLL功能，正常工作状态是自动有效的。DLL的功能很简单，就是通过延时同步DDR内部和外部时钟。

2.5、数据选通脉冲 (DQS)

DDR SDRAM芯片增加了DQS选通脉冲信号，该信号在SDRAM中是不存在的。每个字节的数据对应一个DQS信号，对于16位数据线的芯片来说，每次传输两个字节数据，所以就有LDQS和UDQS信号。主要用来在一个时钟周期内准确区分出每个传输周期，并便于接收方准确接收数据。

DQS是双向信号，与DQ数据信号双向同步。从图2的框图分析知，在读取DDR数据时，DQS与数据信号同时生成（在CK与CK#的交叉点）。而DDR中的CL指从CAS发出到DQS生成的间隔，如下图所示，数据真正出现在数据I/O总线上相对于DQS触发的时间间隔被称为tAC。注意，DDR的tAC与SDRAM的tAC含义不同。

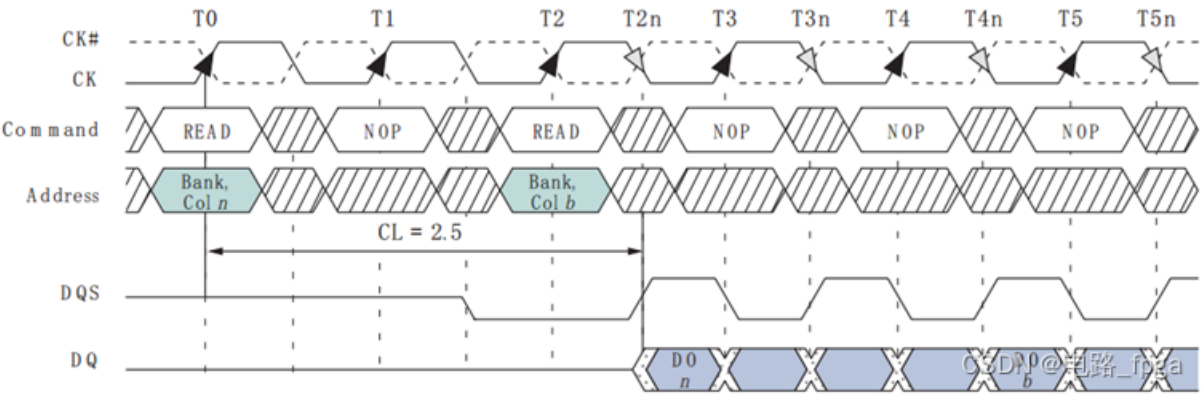


图11 DDR读时序

实际上，DQS生成时，芯片内部的预取已经完毕了，tAC指图2中红框部分的数据输出时间，由于预取的原因，实际的数据传出可能会提前于DQS产生（数据提前于DQS传出）。

如下图所示，向DDR写入数据时，DQS的边沿与DQ的中部对齐，此时数据线上的数据会比较稳定，DDR可以在DQS的边沿将DQ的数据存储。

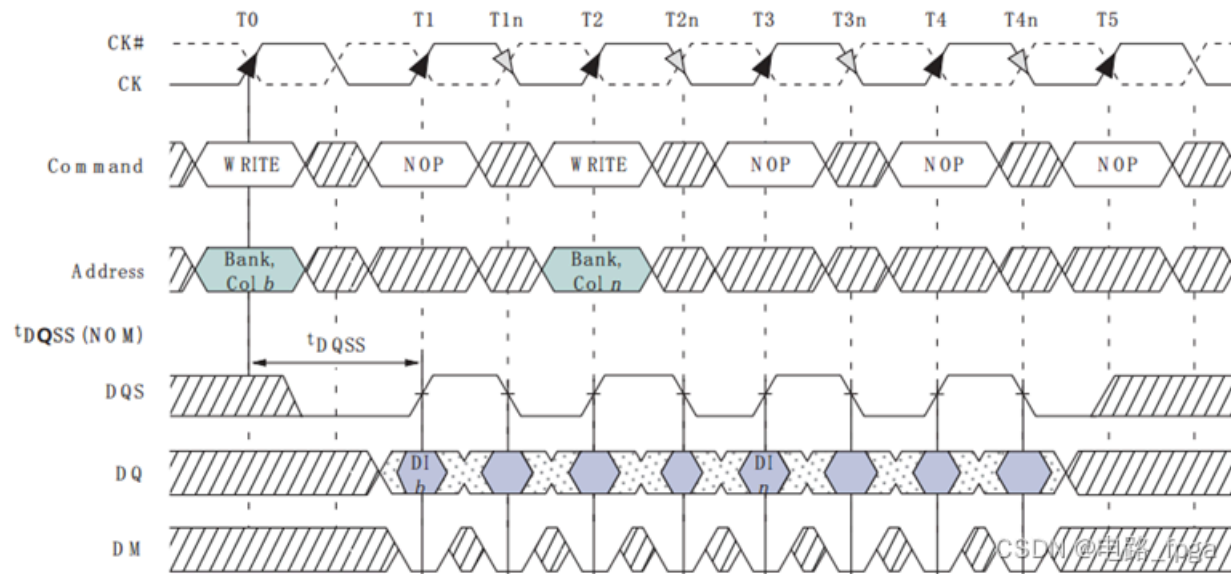


图12 DDR写时序

## 2.6、写入延迟 (tDQSS)

下图是SDRAM芯片的写时序，发出写数据和写命令同时出现在数据线上，没有延时。

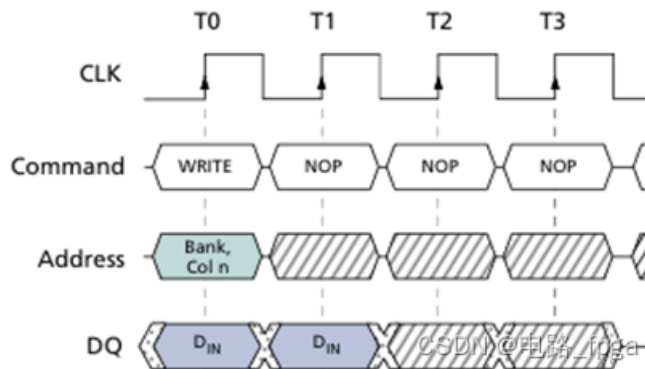


图13 SDRAM写时序

在看图12中DDR的写入时序，在发出写命令之后需要经过tDQSS，DQS、写入数据DQ、数据掩膜才会出现，这段时间被称为DQS相对写入命令的延时时间（tDQSS，WRITE Command to the first corresponding rising edge of DQS）。

这个延迟设计的作用也是在于同步，一个时钟周期传两次数据，控制精度要求较高，必须让接收方做好充分准备才能传输数据。 $t_{DQSS}$ 是DDR写入操作的一个重要参数，太短可能造成接受错误，太长则会造成总线空闲。 $t_{DQSS}$  大于等于0.75个时钟周期，小于等于1.25个时钟周期。

经过上面的延时，芯片内部的时钟很可能与DQS信号不同步了。一般情况下， $t_{DQSS}$ 是一个时钟周期，DDR的时钟信号一般只用来同步控制命令。在前文分析过写入的数据DQ完全通过DQS进行同步，后面通过一个写FIFO将数据从DQS下同步到时钟CK下，所以DQS与CK不同步对数据的写入是没有影响的。

注意数据在时钟的双沿传输，但是指令指令在时钟的上升沿传输，因此当CL为2.5时，读操作完成后，下一个指令需要延迟半个时钟周期后才能发出。如下图所示，读后写操作延迟的增加，如果CL=2.5，还要在 $t_{DQSS}$ 基础上加入半个时钟周期。

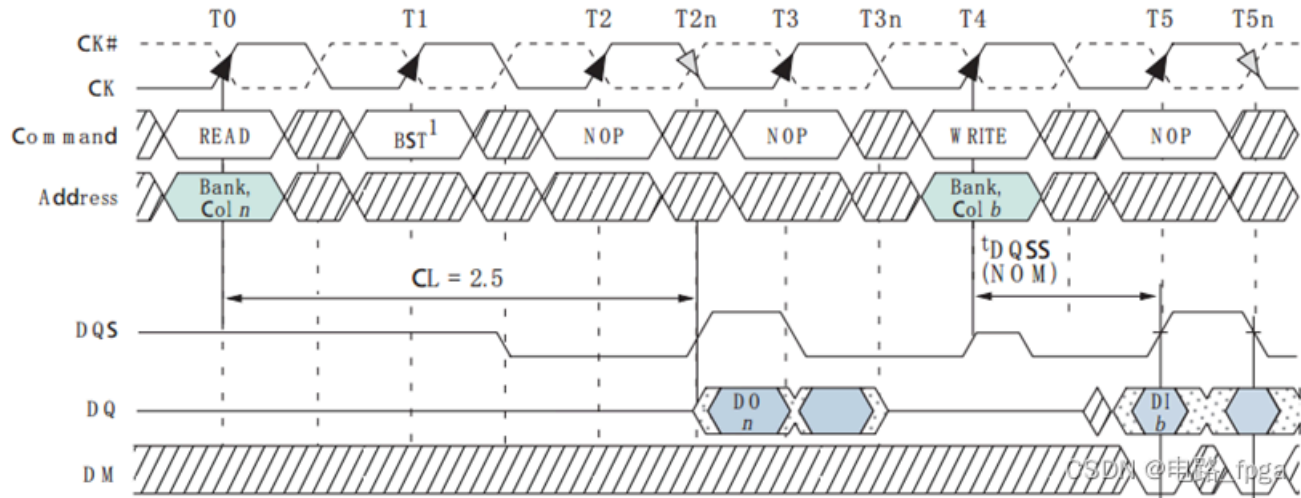


图13 DDR读写时序

另外，DD内存的数据真正写入由于要经过更多步骤的处理，写入时间（ $t_{WR}$ ）也明显延长，一般在3个时钟周期左右，在DDR-II规范中将 $t_{WR}$ 列为模式寄存器的一项。

### 3、总结

经过上面的讲述，对DDR与SDRAM的区别稍微总结一下：

1. 由于DDR采用双沿传输数据，每个时钟周期传输2次IO宽度的数据，而SDRAM只在上升沿传输数据，每个时钟只传输一次数据。
2. 由于DDR在双沿传输数据，对时钟精确度的要求较高，采用差分时钟，时钟不支持挂起操作。
3. DDR的数据IO在双沿传输数据，而内部数据只在上升沿传输，就需要将IO双沿传输的数据转换为单沿传输的数据，所以内部传输数据位宽是IO传输数据位宽的2倍，即两位预取。
4. DDR不支持单次突发和全页突发传输，在读取数据时也不支持数据掩膜功能。

5. 由于采用双沿传输，所以读潜伏期支持2.5这种两个半的时钟周期延时。SDRAM在写入数据时没有延时，写命令和写数据同时有效，而DDR发出写命令后，需要经过tDQSS延时后，数据DQ、数据掩膜DM、数据选通信号DQS才会有效。

6. DDR采用双沿传输数据，增加了数据选通信号DQS，读数据时DQS边沿与读出数据DQ边沿对齐，写入数据时，DQS的边沿与写入数据DQ的中心对齐，DDR可以在DQS的边沿采集数据DQ，前文详细分析过写入数据的过程。

7. 延时锁存回路（DLL），由于DDR对时钟精度要求较高，增加了DLL来调节内部时钟与外部时钟延时，确保两个时钟同步。

8. 另外SDRAM采用3.3V的LVTTTL电平，而DDR采用2.5V的SSTL\_2电平，DDR的功耗更低。SDRAM采用TSOP-II封装，而DDR有TSOP-II封装和BGA封装两种。

本文主要参考DDR与SDRAM的手册以及电脑高手的一篇文章，在使用过IDDR之后就会发现双沿传输数据转换为单沿传输数据其实也比较简单，分析数据的读写也就不难了。

**FPGA** 使用DDR3这些器件来说，可以很简单，直接调用DDR IP即可，连DDR怎么工作都可以不用了解，只需要写一下IP的用户接口时序即可。但如果DDR的核心板是自家公司设计的，下载程序之后不能正常工作，如果对DDR如何工作都不了解，如何确定问题出在FPGA还是硬件？

DDR最初是作为PC的内存出现，所以了解很多内容要结合一下PC端，比如全页突发为什么会被砍掉等。

本文只对DDR与SDRAM的区别做了讲解，相同部分就不做介绍了，毕竟内容挺多的，DDR就把SDRAM的单沿传输变为双沿传输就引发了这么多改变。下一节继续分析DDR如何升级成DDR2。

**您的支持是我更新的最大动力！将持续更新工程，如果本文对您有帮助，还请多多点赞👍、评论💬和收藏★！**