相关基础知识

DDR3全称第三代双倍速率动态随机存储器

一般DDR是由多个存储颗粒组成的，这些存储颗粒也被称为SDRAM，存储颗粒的类型或者说型号由不同的FBGA代码所唯一确定，存储颗粒的内部结构也是bank、行、列，要注意颗粒的存储单位一般是bit。

掉电无法保存数据，需要周期性刷新，支持brust传输，一般brust长度为8

DDR3存储：由bank、行地址、列地址三个参数确定具体存储位置。

如何理解呢？bank相当于一个存储块，一个DDR中有若干个存储块，在每个存储块中通过行地址和列地址可以唯一确定一个最小存储单元。DDR3的最小存储单元是16bit。（一种情况）

因此DDR3的容量为bank数\*行数量\*列数量\*存储单元容量。

另外DDR3的命名规则中可以解析出具体型号的一些参数：但是更重要的是通过名称型号去得到对应的用户手册，查询内部参数，并计算颗粒的存储量。

不同的DDR（内存条）会使用数量、型号不一的颗粒去组成。因此会出现使用相同颗粒的不同内存条（DDR）内存大小不同。需要注意的是内存条的内存大小一般用Byte表示。

时钟问题：

Mig IP核会从外部接入两个时钟，其中一个是sys\_clk（系统时钟）是mig工作的时钟；另外一个时钟是FPGA给的（ref\_clk，且必须为200M）叫做参考时钟。这两个时钟都是晶振产生的100MHZ时钟先经过差分，再经过倍频得到的。其中sys\_clk是mig工作的输入时钟。参考时钟目前不知道什么用处。差分时钟指的是等大反相的两个时钟信号，经过差分之后会生成一个较为精确的单端时钟，这个单端时钟经过倍频在FPGA内到处使用。

DDR3工作的时钟由Mig的系统时钟由FPGA输入到DDR3芯片，是差分时钟。

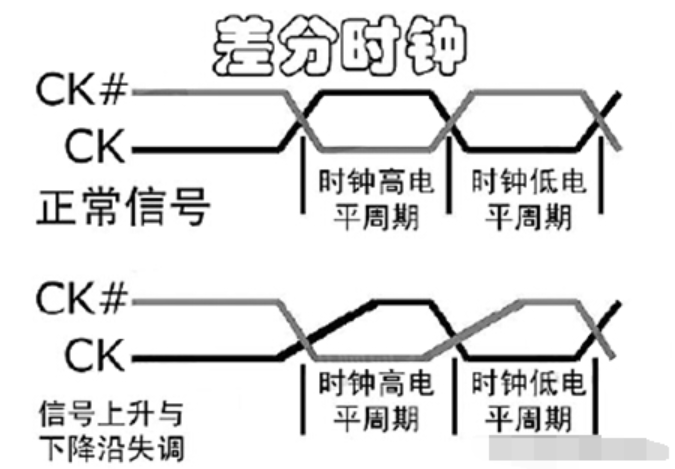
用户端也会有自己的用户时钟，是MIG ip核输出给用户端的时钟。被命名为ui\_clk

就是我们自己写代码的时候用的时钟

DDR3芯片工作的时钟与用户端时钟有一个比例关系：

DDR3芯片工作的频率：用户时钟频率为4：1或者2：1，当DDR3芯片工作时钟为800M的时候，比例只能为4：1。这个比例关系由mig核来配，mig核会把接入进去的sys\_clk通过倍频和分频的方式生成两条时钟，分别给用户和ddr，这个时钟分配还和数据带宽以及数据位宽有关，是他们共同决定的。（2:1比例的用户接口数据位宽是存储器物理接口数据位宽的4倍。而4:1比例的总线宽度是物理存储接口数据宽度的8倍）

另外补充一下差分时钟的原理：差分时钟的作用是对触发时钟进行校准，DDR是上下沿都采样，因此必须保证传输周期的稳定，使用与触发时钟反相的时钟与之进行差分，可以对因为温度、电阻而改变的clk上升下降沿进行纠正。同时出现在两个信号上面的干扰在差值中会大部分被抵消。这样就把干扰对差分信号的质量的影响降到了最低，大大提高了信号的抗干扰能力。使得上升沿和下降沿变得精确如图所示：



对于时钟，有三种不同的模式，nobuffer、single end 以及difference，含义分别为:

FPGA（或是其他设备）内部一般会有几个晶振，频率一般会使用其中的 100MHZ，而在板子上也有一个PPL（倍频器）如果时钟信号来自于FPGA内部的晶振（或是经过倍频的晶振信号）这个信号是没有被加上ibuff原语的，因此选择nobuffer，如果时钟信号由绑管脚的方式接入（其实就是直接接晶振），那么有单端（single end）和差分（difference）两种。

Mig ip核配置原理：

Mig IP核的工作原理：

工作方式是作为用户端和DDR芯片之间的中间层。这样用户只需要设计用户端与mig核之间的接口与时序，就可以方便地通过mig核对ddr进行读写。（纯FPGA类型）

配置mig核的方式可以通过修改源代码也可以通过在vivado中点点点。具体是配置时钟和管脚，之后再将mig模块在用户端顶层模块中进行例化，接好用户端与mig之间的接口。

需要注意的一点是：我们通过ip核连接用户端和复杂的设备端，这个其实并不是数据传输过程中涉及到的，而是数据到达对应的节点之后，在物理层上（逻辑、时序）去考虑的问题，以网络层数据包传输为例：ip数据报，在数据传输过程中涉及到路由算法这些，传输到了对应的节点处之后，通过一些接口协议比如AXI协议，将完整的数据包送到对应的设备接受端以内之后，对数据包进行解包