

基于FPGA的DDR相关知识导航界面

个人觉得想要真正理解DDR，那就要从SDRAM入手，SDRAM读写 **控制器** 也可以作为入门FPGA的一个判断标准，说实话I2C、UART、SPI这些接口难度太低了，会写这些接口说明不了什么。

很常见的问题，现在 **DDR3** 、SDRAM、DDR4官方都有IP，还需要自己写SDRAM驱动吗？意义在哪？

如果是工程使用，那当然不需要自己写驱动，但如果你是初学者，刚接触 **FPGA** ，写过UART、SPI等低速接口。那么推荐你读一下SDRAM手册，写一下SDRAM驱动，altera很多C4的板子都带有SDRAM芯片，用signal tap抓一下读写时序。

注意看手册时，要详细了解SDRAM读、写时序，模式寄存器、自刷新、预充电、突发读写、还有各个指令后的一些时序参数，这些参数与SDRAM的型号直接挂钩。后续的DDR只不过是SDRAM的基础上进行的一些改进，了解SDRAM的工作原理之后，也就了解了DDR的大部分工作原理，仅仅需要补充一下具体区别即可。

首先需要阅读SDRAM手册，SDRAM的手册中有很多时序图，这些图可以清晰讲解SDRAM的指令时序和读写时序。这篇文章是当时自己编写SDRAM驱动之前阅读手册后总结在word中的，时间过了很久了，有需要的可以看看。

特别要注意SDRAM的管脚，各个管脚的含义含义，后续DDR只是在此基础上新增几个管脚而已，多数管脚并没有变化。

SDRAM技术手册阅读总结-基于IS42S16400J

下文是一年前通过SDRAM的框图分析其工作原理所写的一篇文章，内容比较简单，但原理分析的比较到位。在阅读手册时，最好多注意配图，理解图片想要表达的意思，之后即使忘记了，通过一张图也能得知其各个功能。

如何通过SDRAM内部结构图了解工作方式？

当年我写SDRAM驱动的时候还没有CSDN账号，所以就没有保留SDRAM驱动相关代码和设计思路。但不用慌，就SDRAM驱动设计方面，可以看看明德扬的这篇文章，设计思路还是不错的。

SDRAM读写控制器

为了提高读写速率，SDRAM推出多年之后，DDR出现了，与SDRAM的区别在于DDR的接口时序采用双沿传输数据，注意指令依旧在时钟上升沿传输。DDR内部时钟与接口时钟频率还是相同的，内部数据在内部时钟的上升沿进行传输，内部数据位宽变为接口数据位宽2倍，来达到数据传输速率平衡。

为了保证传输数据的精确性，采用差分时钟，增加DQS、延时锁存回路、电平变低，功耗优化等等，详情看下文。

从内部结构分析SDRAM到DDR的变化

在DDR提出之后，如何继续提升数据传输速率？最直接的方式就是通过提升时钟频率来提升数据传输速率，但是DDR内部的时钟频率由于功耗等原因不能一直提升。

因此出现了DDR2，DDR2把内部时钟与接口时钟分离，接口时钟频率变为内部时钟2倍，内部数据位宽变为接口数据的4倍，即4倍预取，达到数据传输速率平衡。

由于接口时钟频率进一步提升，将DQS信号变为差分信号，DDR2支持8个bank，容量相比DDR提升一倍，将终端电阻引入到芯片内部，减小PCB设计的难度。DDR与DDR2的具体变化可以查看下文。

从内部结构分析DDR到DDR2的变化

DDR3将接口时钟频率变为内部时钟频率的4倍，相比DDR2在提升一倍，内部数据位宽变为接口数据位宽的8倍，即8倍预取。接口传输8次数据，内部只需要传输一次数据。

DDR3相对DDR2新增了复位引脚、ZQ校准引脚、写入均衡等等，详情看下文。文中还对米联客的DDR3原理图做了简要分析，包括上电时序等。

从内部结构分析DDR2到DDR3的变化及DDR3原理图分析

DDR3的驱动设计很复杂，如果不是项目有特殊要求，直接调用官方的IP即可。xilinx给用户提供了mig IP，用于驱动DDR2、DDR3等，7系列芯片不能驱动DDR4。

下文对mig IP的配置参数结合手册进行详细讲解，如果用户看过前面几篇理论文章，曾经写过SDRAM驱动，那么看这些参数的配置将没有任何难度。

详细讲解Xilinx DDR3 的MIG IP生成步骤及参数含义

上面生成了mig IP，本文仿真该IP，xilinx官方提供的所有IP都有配套的示例工程，内部会提供仿真模型，用户可以借助该仿真模型，完成自己DDR3驱动的仿真。

本文主要是对app接口的各个信号含义进行讲解，最后使用示例工程对IP仿真。了解该IP读、写数据的时序，便于后续开发设计。

Xilinx DDR3的MIG IP信号分析及仿真和上板测试

mig IP比较简单，但是提供的app接口和axi接口在传输数据时，都需要等待握手，用户直接使用该接口传输数据流很不方便。

因此在实际使用时，更多时候需要将接口封装成FIFO或者RAM的接口，如果传输图像数据等对数据丢失要求不高的数据，一般采用FIFO接口，更加简单。

下文将MIG的APP接口封装成FIFO接口，有乒乓操作的模式，用户在使用DDR3 IP时，只需要读写FIFO即可，该文章详细分析了每个always块的设计，再加上工程中注释比较齐全，理解起来用过没有难度。

将Xilinx DDR3 MIG IP核的APP接口封装成FIFO接口（含源码）

下文是关于DDR3模块的应用，上位机通过UDP协议向FPGA传输图像数据，FPGA接收数据后在DDR3中暂存，最后读取图像在HDMI接口上显示。文末上板时对是否使用乒乓操作的结果做了对比。方便大家知道在图像传输时为什么要使用乒乓操作存储图像数据。

基于FPGA的以太网传输图片通过HDMI显示（含源码）

在千兆网传输图片的工程中，因为FIFO深度设置原因，曾经出现过一个bug。下文记录了利用ILA查找该bug的原因的方式，由此可以看出，对于FPGA的使用，一定要知道每个模块每个信号的含义，在出现问题之后，你才能够根据一些关键信号的状态，判断出现问题的可能原因。

即使是别人的代码，对于代码内容也要比自己写的代码更加熟悉才行，不然需要bug就凉凉，别人也不会给你去排除bug，没有这个义务与责任。

以太网传输图片工程出现的问题总结(含源码)

下面案例是使用OV7725摄像头采集图像数据，通过DDR3暂存，最后显示在HDMI显示器上。如果了解过HDMI和OV7725后，相关内容就比较简单。

由于是二手开发板送的一个摄像头，摄像头本身有点问题，导致成像效果不理想，大家知道原理即可，可以使用自己手册摄像头进行验证。

基于FPGA的OV7725摄像头的HDMI显示（含源码）

前文讲解的DDR3 mig IP都是使用的app接口，但是mig IP还有一个axi4_full接口，下文将该接口封装为FIFO接口，替换以太网传输图像数据的APP接口的mig IP。

其实两者都差不多，只不过提供给用户的axi接口的读写通道是独立的，支持用户同时进行读写操作，mig IP内部会多读写操作进行仲裁，在配置IP时可以配置仲裁方式。app接口需要用户自己确定读写仲裁逻辑，其余方面没有区别。

将Xilinx DDR3 MIG IP核的AXI_FULL接口封装成FIFO接口（含源码）

目前关于DDR的使用就这么多，后续在新增一些案例，如果换了开发板在新增DDR4相关知识。下面的图片用于封面！！！！



如果对文章内容理解有疑惑或者对代码不理解，可以在评论区或者后台留言，看到后均会回复！

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