

AXI Lite协议详解

AXI Lite协议详解

axi (Advanced eXtensible Interface) 是一种总线协议，该协议是 ARM 公司提出的amba (Advanced Microcontroller Bus Architecture) 3.0协议中最重要的部分，是一种面向高性能、高带宽、低延迟的片内总线。

1、概述

axi的地址/控制和数据相位是分离的，支持不对齐的数据传输，在突发传输中，只需要首地址，同时分离的读写数据通道、支持显著传输访问、乱序访问。

amba总线包括apb、ahb、axi，其中axi向下兼容其余两种总线，xilinx在推出zynq芯片之后，ARM端就使用 axi总线 互联外设，PS端也可以通过axi总线与PL端进行数据交换。因此对于开发ZYNQ FPGA来说，AXI总线是必须掌握的。

axi4分为axi_lite、axi_full、axi_stream，axi_full是一种全模式的数据传输总线，相对来说比较复杂，支持突发传输，但是对于FPGA来说很多模式是不会被使用的，会使用到的信号和模式还是比较简单的。

而axi_lite可以说是axi_full简化版本，不支持突发传输，每次只能读、写单个地址的数据，相对来说更加简单。

axi_stream是一种流式传输协议，一般只在FPGA内部进行数据传输，各个信号比较简单。

本文通过对手册的解读，对axi_lite协议进行讲解，通过一些时序图，加深对信号变化的理解。

2、axi_lite协议

axi_lite接口的地址与数据通道是分离的，各自包含一组信号，数据的传输需要握手应答。包含写地址通道、写数据通道、写应答通道、读地址通道、读数据通道。因为读应答信号包含在读数据通道中，因此读过程少一个通道，每个通道的数据传输都需要握手，保证从机能正确接收数据。

首先包括时钟信号ACLK和复位信号ARESETn，剩下的就是几个通道相关的信号了。

2.1、写地址通道

写地址通道的作用就是将写操作的地址信号写入从机，写地址相关信号如下表所示，注意输入输出的方向是以主机模块为对象的。

表1 写地址通道信号

信号	I/O	位宽	含义
AWADDR	O	32	写地址信号

信号	I/O	位宽	含义
AWPROT	O	3	写端口信号，lite协议中该信号一般为0。
AWVALID	O	1	写地址有效指示信号，高电平有效。
AWREADY	I	1	写地址应答信号，高电平表示从机接收到主机输入的地址信号。

需要写地址有效指示信号和写地址应答信号同时为高电平，表示从机接收到主机写入的地址数据，对应的时序有三种。

如下所示，当写地址有效指示信号为高电平时，写地址应答信号开始为低电平，此时需要保持写地址有效指示信号，直到写应答信号拉高为止。

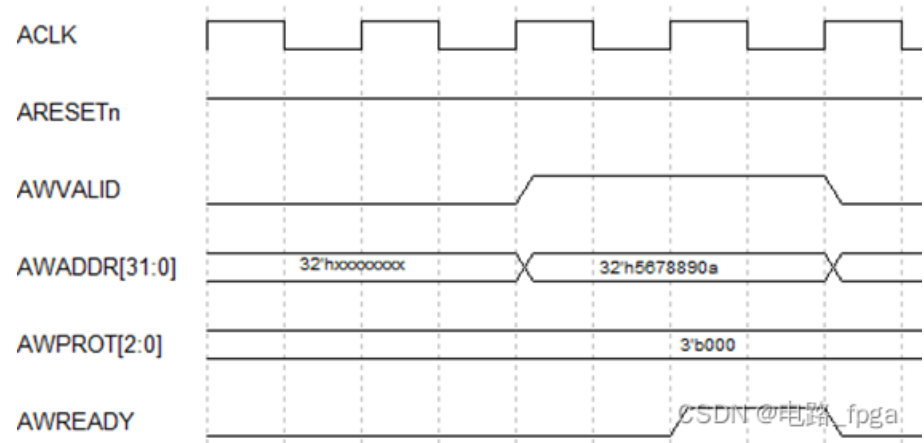


图1 写地址时序

下图是第二种情况，写地址有效指示信号拉高时，写地址信号也为高电平，此时表示写入的地址已经被接收。

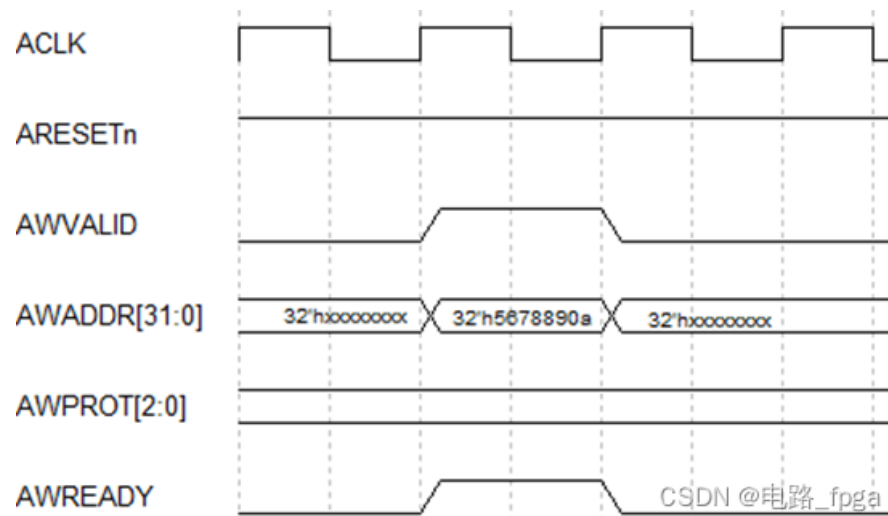


图2 写地址时序

下图中写应答信号可能会提前拉高，那么写入地址有效后，下个时钟可以立即拉低。

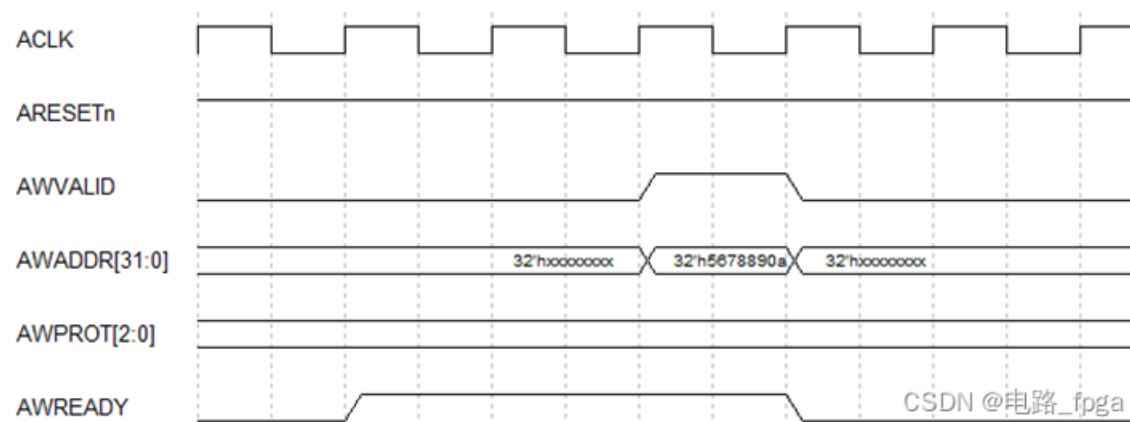


图3 写地址时序

注意：虽然说AWREADY信号可能会提前拉高，但是主机写入地址后必须主动拉高AWVALID信号，AWVALID不能等待AWREADY拉高后才拉高。

当写地址通道写入地址数据后，必须等待从机的应答信号拉高后，地址信号和有效指示信号才能发送变化。

后续通道的握手原理与该通道都一致，后续只会给其中一种时序图，希望能够理解。

2.2、写数据通道

axi_lite总线可以通过掩码信号屏蔽部分数据线上的数据，1位掩码信号控制1字节数据的写入，掩码信号为低电平表示该字节数据不写入寄存器。昔日数据通道相应信号如下表所示：

表2 写数据通道信号

信号	I/O	位宽	含义
WDATA	O	32	写数据信号
WSTRB	O	4	写数据掩码信号，低电平有效。
WVALID	O	1	写数据有效指示信号，高电平有效。
WREADY	I	1	写数据应答信号，高电平表示从机接收到主机写入数据。

写数据相关信号的时序如下图所示，只有当写入数据的应答信号为高电平时，才能表示主机将数据写入到从机中，此时才能将数据有效指示信号拉低，否则保持不变。

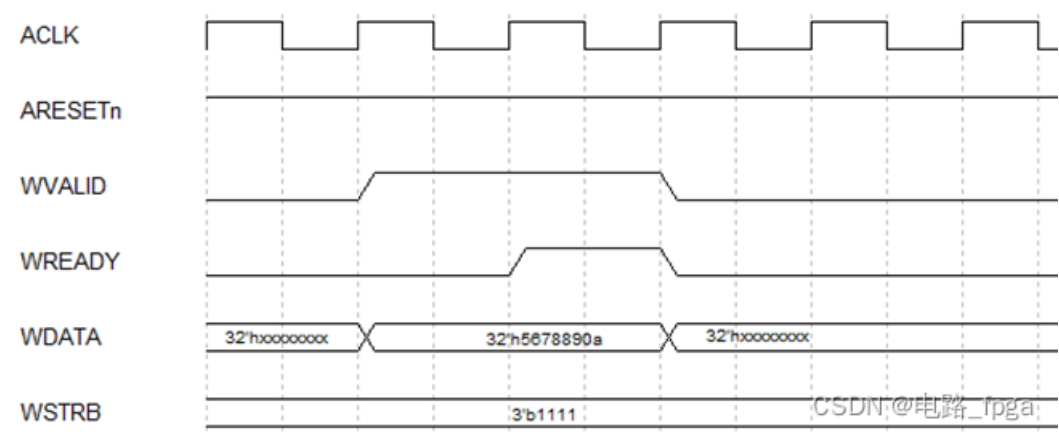


图4 写数据时序

上述讲解了写地址和写数据通道后，是否会产生这样的疑问，必须在写地址完成后才能写入数据吗？

手册中这幅图告诉了我们答案，如下图所示，图中有两种箭头。

单向箭头指向可以在箭尾信号拉高之前或之后拉高的信号，意思就是两者的拉高顺序没有要求。即**WVALID**信号可以在**WREADY**信号拉高之前或者之后拉高，因此写数据有效指示信号可以在写地址应答信号有效之前拉高，也表明写地址和写数据的有效信号可以同时拉高。

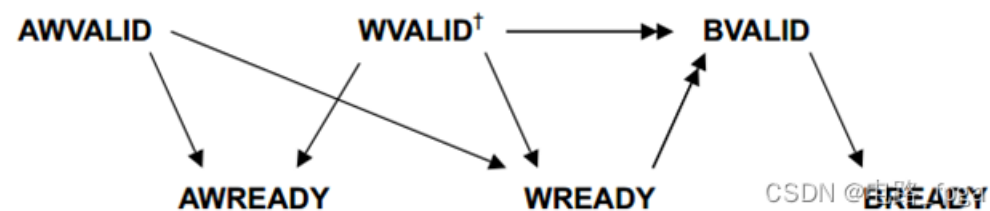


图5 写事务握手依赖关系

双箭头指向必须在箭尾处信号拉高后才拉高的信号，比如**BVALID信号必须在WVALID和WREADY均为高电平时才能拉高，即写数据完成后才能拉高写应答通道的主机应答信号。**

通过上文分析，写地址和写数据通道的信号时序可以如下图所示，地址和数据有效指示信号同时拉高，各自的应答信号拉高后立即拉低，其余时间保持不变。也是传输数据最快的方式，一般FPGA可以使用该时序。

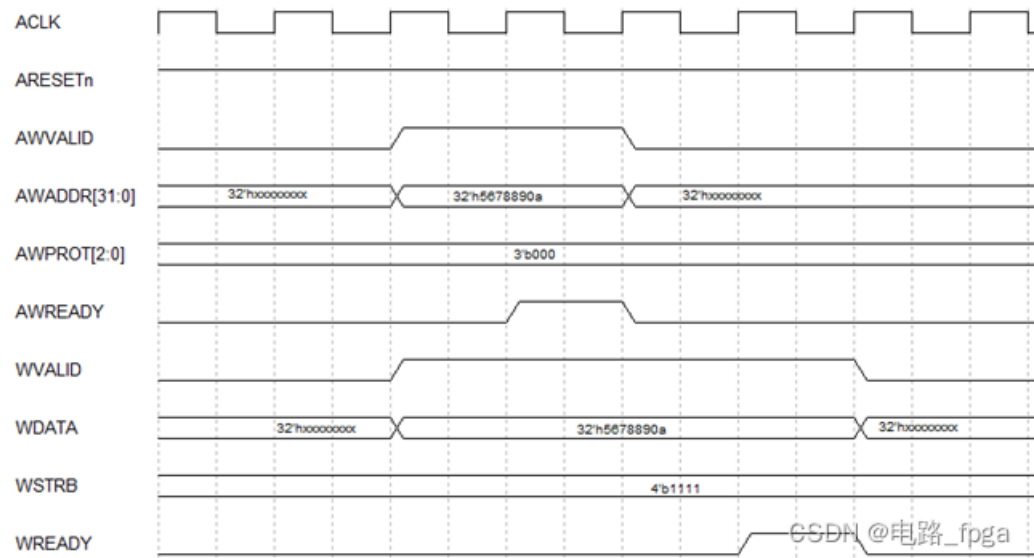


图6 写地址、数据通道时序

2.3、写应答通道

当主机向从机写入地址和数据之后，需要等待从机的写应答通道响应，告知主机此次写入是否正确，通过一个信号的状态可以查看从机是否正确接收数据。

表3 写应答通道信号

信号	I/O	位宽	含义
BRESP	I	2	应答状态信号，为0表示从机正确接收写数据。
BVALID	I	1	应答状态信号指示信号，高电平有效。
BREADY	O	1	主机应答信号，表示接收到从机应答通道的信号。

经过上面分析，BVALID信号必须在主机写入数据完成后，才能拉高，当写入BVALID有效时拉低，对应的时序图如下所示。

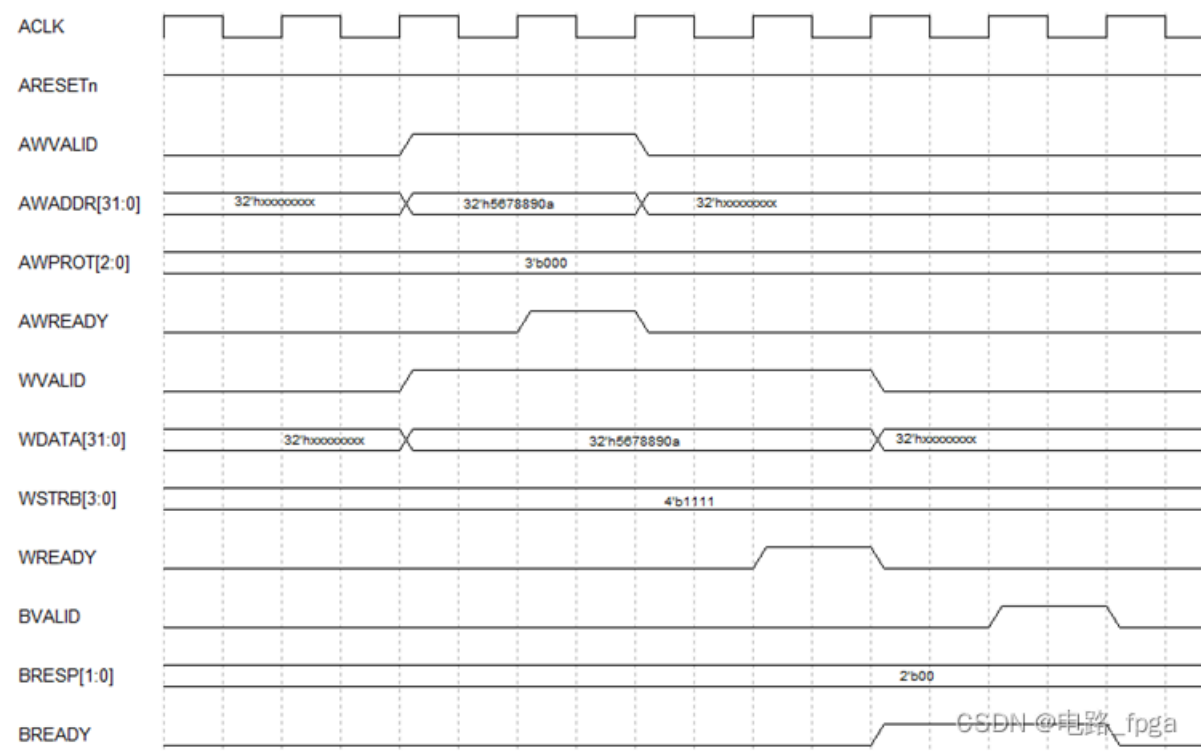


图7 写过程时序

2.4、读地址通道

读地址通道相关信号与写地址通道相关信号含义一致，时序也一致，如下表所示，信号以AR开头，A表示地址，R表示读吧。

表4 读地址通道信号

信号	I/O	位宽	含义
ARADDR	O	32	读地址信号
ARPROT	O	3	读端口信号，lite协议中该信号一般为0。
ARVALID	O	1	读地址有效指示信号，高电平有效。
ARREADY	I	1	读地址应答信号，高电平表示从机接收到主机输入的地址信号。

由于信号与写地址基本一致，所以不再赘述。

2.5、读数据通道

读数据通道相关信号如下表所示，相比写数据通道，多了一个应答信号。

表5 读数据通道信号

信号	I/O	位宽	含义
RDATA	I	32	读数据信号
RRESP	I	2	读取传输的状态信号，成功读取从机数据时为0。
RVALID	I	1	读数据有效指示信号，高电平有效。
RREADY	O	1	读数据应答信号，高电平表示主机接收到从机读出数据。

由于读数据通道包含了RRESP信号，因此读过程就不存在读应答通道了。

读过程的应答关系如下图所示，从机的读数据输出有效信号必须在读地址有效指示信号和读地址应答信号拉高之后才能拉高，而读数据应答信号可以在读数据有效指示信号拉高之前拉高。

即读数据应答信号RREADY可以与读地址有效指示信号ARVALID同时拉高。

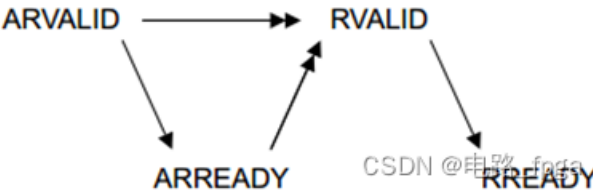


图8 读取事务握手依赖项

相关的时序如下图所示，下图中读数据应答信号与读地址有效指示信号同时拉高，当然读数据应答信号也可以在读地址应答信号拉高之后才拉高，都没有影响。

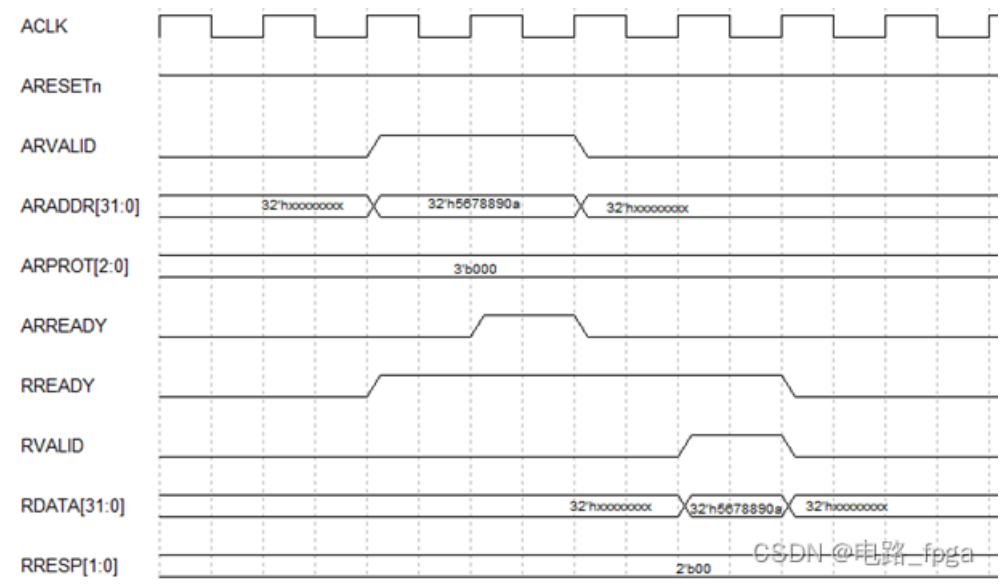


图9 读时序

手册中并没有对读写时序的顺序做相关要求，由于读、写时两个独立的通道，因此读写操作可以同时进行，但是在实际测试时发现，读写同一个地址的数据时会导致读出数据错误。

因此在实际使用时，应该避免同时读、写同一地址数据，可以在写完之后在进行读操作，这样会降低总线效率，但实现逻辑简单，时序如下图所示。另一种方式就是在读操作时，如果当前也要进行写操作，对比读写地址是否一致，如果地址一样，则优先进行写操作，这种方式逻辑会复杂一点，还需要知道当前的写地址。

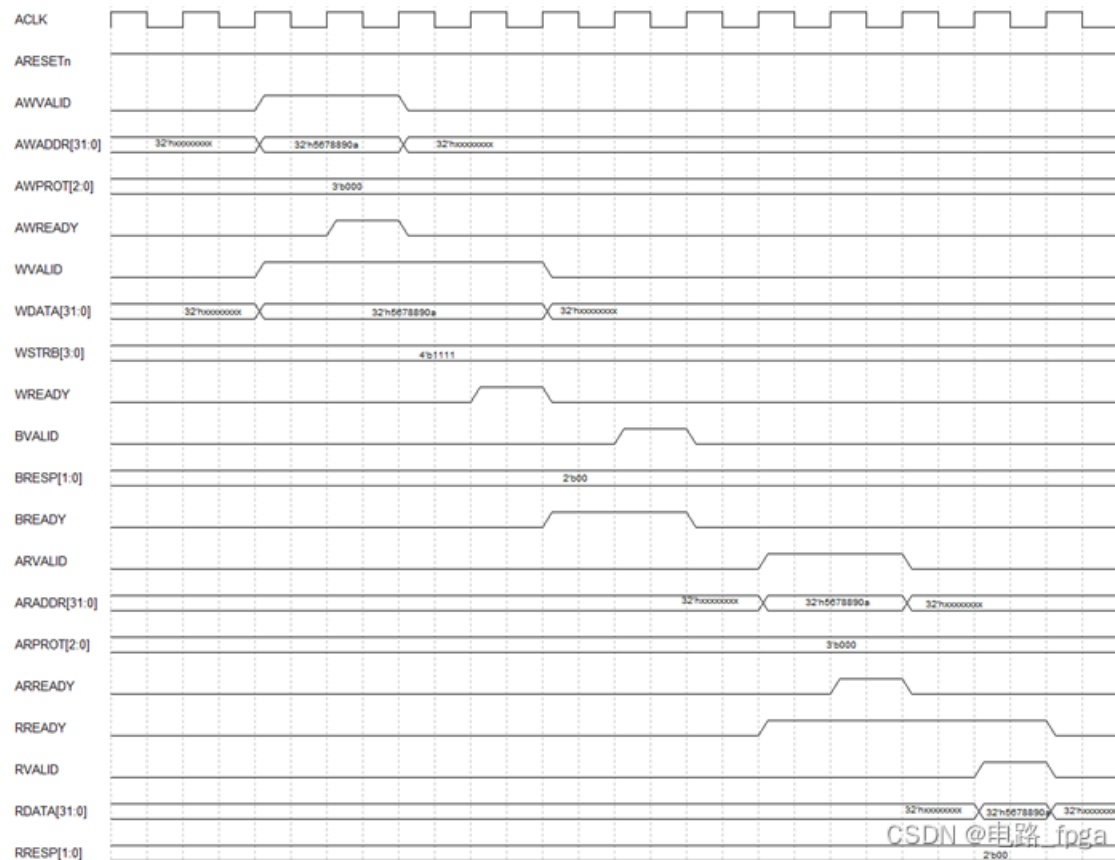


图10 读写时序

3、总结

前面已经详细讲解axi_lite接口的各个信号时序及注意事项，由此可知，axi_lite协议其实也比较简单，每个通道都需要两个信号进行握手，每次只能读写单个地址的数据。

以下对要点进行总结：

1. 主机的有效指示信号（写地址有效、写数据有效、读地址有效）必须主动拉高，不能等待从机的应答信号拉高之后在拉高（原因在于部分从机的应答信号信号可能需要等待主机有效信号拉高后才拉高，就会造成主机和从机均在等待对方拉高）。
2. 主机写数据和写地址有效指示信号可以同时拉高，加快写入数据效率。
3. 主机支持同时进行读写操作，但是不能同时读写同一地址数据。

由于xilinx的部分IP只支持axi总线，PL端在调用该IP时，就必须将普通接口转换为axi接口时序。另外也可以通过axi总线将PL端的模块挂在PS的总线上，因此下文将通过Verilog HDL实现axi_lite_master模块。

AXI手册可以在ARM公司官网下载，也可以在公众号后台回复“**AXI手册**”（不包括引号）获取。

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！