

AXI-FULL协议详解

1、概述

前文对axi_lite协议进行了讲解，并且使用 Verilog HDL编写了一个常规数据流接口转axi_lite的模块。本文在此基础上对axi_full协议进行讲解，axi_full协议与axi_lite的主要区别在于支持突发读写（传输指定长度数据只需要发送首地址），从而提高数据读写速率。还支持一些其他功能，但是这些功能对于fpga开发来说一般不会使用，作为了解即可。

Axi_full的信号也比较多，与axi_lite一样分为5个通道，其中写地址通道和读地址通道的信号含义和时序均一致。

下图是axi_full写流程图，首先通过写通道发送首地址，然后在通过写数据通道发送数据，数据传输完成后，从机的写应答通道会回复主机此次写过程是否成功。每个通道的数据均需要主机与从机握手，可以保证数据的稳定写入。

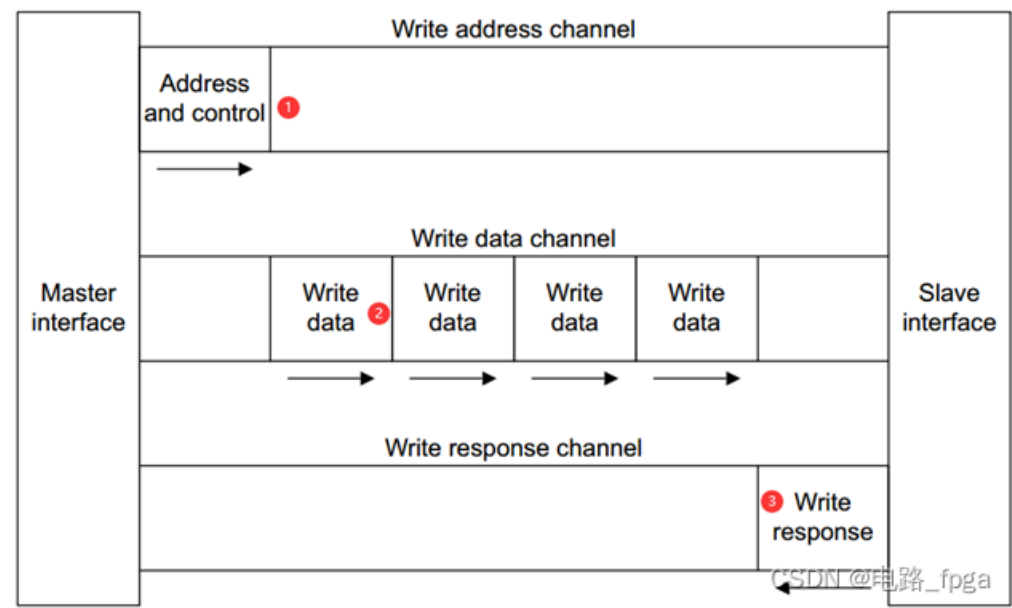


图1 axi_full写流程

下图是axi_full的读流程图，首先通过读地址通道发送读数据的首地址、突发长度、突发读取数据大小等信息。从机应答之后，会读数据通道输出对应地址的数据，主机通过握手接收数据。

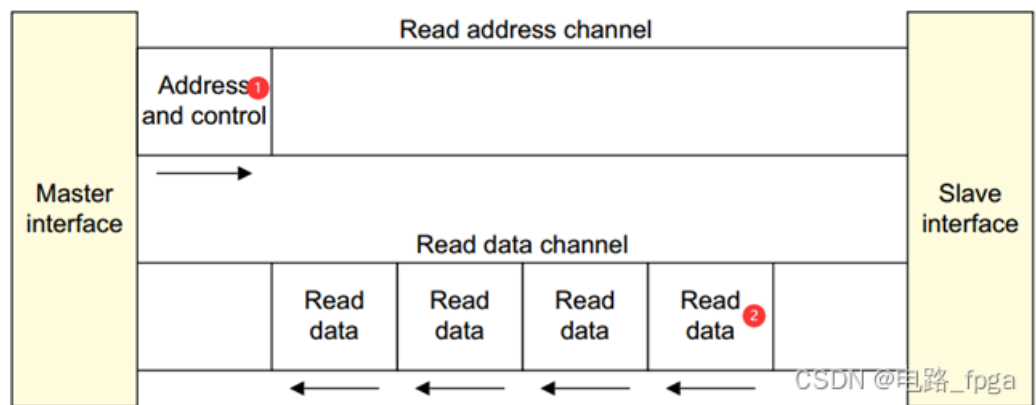


图2 axi_full读流程

由此可以，读写数据的流程与axi_lite保持一致，区别在于读写支持突发长度大于1的突发传输。后文将对其每个通道的信号做详细讲解，有部分信号是为了兼容AXI3总线或者给ARM提供的信号，这部分将简要讲解，fpga需要重点关注的信号会重点标出。

2、写地址通道

Axi_full写地址通道的信号如下表所示，相对axi_lite来说增加了很多信号，但在实际使用时，很多信号都是固定取值，不用过多关注。注意下表中信号的输入、输出方向是以主机位参考对象的。

表1 写地址通道信号

信号	方向	含义
AWID	O	写地址ID，用于对应写地址与写应答。
AWADDR	O	写地址，突发写的首地址。
AWLEN	O	写突发传输长度，单位字节。
AWSIZE	O	一次写数据的字节数。
AWBURST	O	写突发传输类型，一般取值2'd1。
AWLOCK	O	锁类型，只为了兼容AXI3协议，取值为0。
AWCACHE	O	写突发传输的数据存储方式（针对从机怎样存数据）。
AWPROT	O	写突发的优先级和保护类型，一般取值3'd0。

信号	方向	含义
AWQOS	O	服务质量，一般取值4'd0。
AWREGION	O	区域标识符，允许从机上的单个物理接口用于多个逻辑接口，一般不使用。
AWUSER	O	用户自定义信号。
AWVALID	O	写地址有效指示信号，高电平有效。
AWREADY	I	写地址应答信号，高电平有效。

上表对写地址通道信号做了大致说明，下面将对这些信号的取值含义做一些讲解，对于FPGA设计需要注意的点会重点标出。

AXI4协议 的写突发长度**AWLEN**，如果突发类型为递增突发，那么突发长度可以设置为[1,256]。如果是其他突发类型，那么突发长度只支持[1,16]。

注意突发长度**AWLEN**信号位宽为8，且**AWLEN**的值从0开始，所以实际突发长度等于**AWLEN+1**。

AWSIZE表示每次写数据时最大数据位宽对应字节数，即写数据位宽除以8，取值范围如下所示。

表2 awsize取值

写数据(bit)	写数据(byte)	awsize
1024	128	3'b111
512	64	3'b110
256	32	3'b101
128	16	3'b100
64	8	3'b011
32	4	3'b010
16	2	3'b001
8	1	3'b000

从上表可知，axi_full协议中写数据的最大数据位宽可以支持1024bit。

突发类型WSTRB一般分为三种，为0时表示固定地址（FIXED）写突发，即向一个地址中写多次数据，最终只会留下最后写入的数据。比较常用的是**取值为1的地址递增(INCR)突发传输，当写入一次数据后，地址会自动加1，从而将突发传输数据写入连续地址中**。还有一种回绕突发类型（WRAP），与递增突发类似，只不过当地址累加到最大地址后会回到最初的地址，在FPGA中一般不会使用。

因此FPGA内部一般使用递增突发，WSTRB的值为2'b01。

AWCACHE一般用于告诉从机，突发传输的数据该以怎样的方式存储，需不需要缓冲等等，**FPGA进行传输时一般不需要使用缓冲，取值为4'd0或4'd2**。

然后就是权限以及保护类型的信号AWPROT，该信号的各位取值及含义如下表所示，bit[0]表示此次突发传输的优先级，bit[1]表示访问类型，bit[2]表示此次传输的是指令还是数据，**一般取值为3'b000**即可。

表3 AWPROT信号各位含义

awprot位	值	含义
Bit[0]	0	非特权访问
Bit[0]	1	特权访问
Bit[1]	0	安全访问
Bit[1]	1	非安全访问
Bit[2]	0	数据访问
Bit[2]	1	指令访问

上述就是一些常用信号和部分不常用信号的讲解，对于常规使用，写地址通道只需要关注写地址、写突发长度、写突发类型、写突发大小、写地址有效指示信号、写地址应答信号即可，其余信号均为固定数值，需要使用其余功能时在通过手册去查找具体使用方式即可。

3、写数据通道

写数据相关信号如下表所示，相关信号都比较简单，相对于axi_lite主要增加了WLAST信号，用于指示写突发传输的最后写入数据。

表4 写数据通道信号

信号	I/O	含义
WID	O	写ID，只在AXI3协议中支持。
WDATA	O	写数据，数据位宽与AWSIZE信号设置一致。

信号	I/O	含义
WSTRB	O	写数据掩码，低电平有效。
WLAST	O	指示一次写突发传输的最后写入的一个数据。
WUSER	O	写数据通道用户自定义信号。
WVALID	O	写数据有效指示信号，高电平有效。
WREADY	I	写数据应答信号，高电平有效。

写通道的相关信号与axi_lite协议基本一致，此处就不再赘述各个信号的含义和时序了，WVALID与WREADY的握手即可。

4、写应答通道

写应答通道的相关信号如下表所示，与axi_lite协议写应答通道的信号基本一致，多了个ID信号，这是因为axi_full支持乱序传输、交织传输等等，需要通过ID来确定本次应答的是哪次写突发。

表5 写应答通道信号

信号	I/O	含义
BID	I	写响应ID，必须与AWID的数值匹配。
BRESP	I	写响应，为0表示从机正确接收写入数据。
BUSER	I	写响应通道用户自定义数据。
BVALID	I	写通道数据有效指示信号，高电平有效。
BREADY	O	写通道数据应答信号，高电平有效。

突发写状态信号BRESP为0表示突发写正常，为1表示独占写入成功，为2表示从机错误，为3表示解码器解码错误。可以根据此信号的数值判断与BID相等的AWID写突发是否正确写入从机。

5、读地址通道

读地址通道的信号与写地址通道的信号完全一致，取值也可以保持一致，对应信号如下表所示。

表6 读地址通道信号

信号	方向	含义
ARID	O	读地址ID，与读数据的ID对应。
ARADDR	O	读地址，突发读的首地址。
ARLEN	O	读突发传输长度，单位字节。
ARSIZE	O	一次读数据的字节数。
ARBURST	O	读突发传输类型，一般取值2'd1。
ARLOCK	O	锁类型，只为了兼容AXI3协议，取值为0。
ARCACHE	O	读突发传输的数据存储方式（针对从机怎样存数据）。
ARPROT	O	读突发的优先级和保护类型，一般取值3'd0。
ARQOS	O	服务质量，一般取值4'd0。
ARREGION	O	区域标识符，允许从机上的单个物理接口用于多个逻辑接口，一般不使用。
ARUSER	O	用户自定义信号。
ARVALID	O	读地址有效指示信号，高电平有效。
ARREADY	I	读地址应答信号，高电平有效。

写地址通道信号以AW开头，而读地址通道信号以AR开头，其余一致，各个信号的含义与时序不再赘述。

6、读数据通道

读状态信号RRESP也在读数据通道中，该信号的含义与WRESP一致，为0表示读数据正常。其余信号与写数据通道的含义一致。

表7 读数据通道信号

信号	I/O	含义
RID	I	读ID，必须与ARID保持一致。
RDATA	I	读数据，数据位宽与AWSIZE信号设置一致。
RRESP	I	读应答信号，为0表示读数据正常。

信号	I/O	含义
RLAST	I	指示一次读突发传输的最后写入的一个数据。
RUSER	I	读数据通道用户自定义信号。
RVALID	I	读数据有效指示信号，高电平有效。
RREADY	O	读数据应答信号，高电平有效。

7、总结

Axi_full协议相关信号的简要介绍完毕，从上面可知，大部分信号与axi_lite还是一样的，因为支持突发传输，读写数据位宽可变，因此多了突发长度，读写数据大小，突发结束等信号。

而对于服务质量、区域标识符、用户自定义等信号，大多时候要么为固定数值，要么就没有这些信号。

由于axi_full支持乱序传输，所以在实际使用时，需要判断ID是否对应，从而进行响应操作，如果ID保持为常数，那可以忽略。

关于axi_full协议的详细情况，还是推荐查看ARM公司提供的手册，可以在ARM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。

后文将通过两个实例在FPGA中实现axi_full协议，进一步掌握该协议。

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！