



3.1 AXI 协议简介

AMBA AXI(Advanced eXtensible Interface)协议是一种面向高性能、高带宽系统设计的总线协议,能够满足各种高速系统的总线互连。

AXI 协议的主要特点有:

- 独立的地址、控制和数据接口;
- 支持使用字节选通的不对齐数据的传输;
- 基于特定地址进行的突发传输;
- 通过独立的读和写通道实现低成本直接内存访问(DMA);
- 支持无序数据传输;
- 提供多级寄存器锁存的支持,实现更好的时序收敛。

3.1.1 AXI 版本介绍

AXI 协议是 Xilinx 从 6 系列的 FPGA 开始引入的一个接口协议(AXI3),且在很多 Xilinx 提供的 IP 核中都有使用,目前很多都支持更高的版本 AXI4。AXI4-Lite 是 AXI4 的一个简化版本,实现 AXI4 运行起来的最少接口信号,适用于对传输控制要求不高的应用。

3.1.2 基本结构

AXI 协议是基于突发传输的。每一次传输时,地址通道上有地址和控制信息,描述了被传输数据的特性。数据将在主机和从机之间传输,数据传输的方向是从写数据通道到从机或从读数据通道到主机。在写传输中,主机向从机发送数据流。写响应通道是从机给主机的反馈信号,指示当前写传输的状态,完成写传输。

AXI 协议可以实现以下功能:

- 在有效数据传输前提供地址信息;
- 支持多个数据的传输;

- 支持无序传输。

主机到从机的读数据传输如图 3.1 所示。主机先发起一次地址的传输,随后从机将待读取的数据逐个送往主机。主机的地址发送由读地址通道实现,从机的数据发送由读数据通道实现。

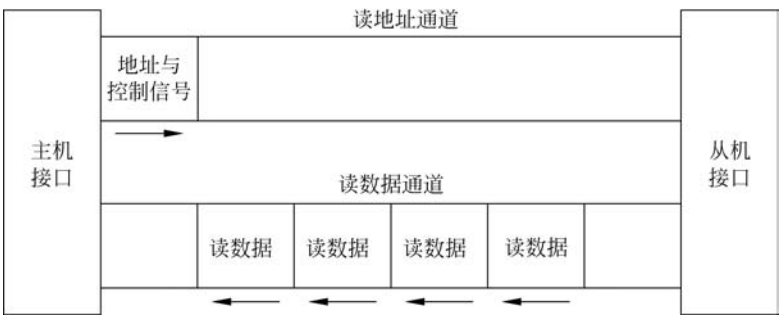


图 3.1 AXI 的读操作

主机到从机的写数据传输如图 3.2 所示。主机先发起一次地址的传输,随后主机将待写入的数据逐个送往从机,从机在完成一次完整的地址和数据接收后,发出一个写响应反馈给主机。主机的地址发送由写地址通道(Write address channel)实现,主机的数据发送由写数据通道(Write data channel)实现,从机的响应信号由写响应通道(Write response channel)实现。

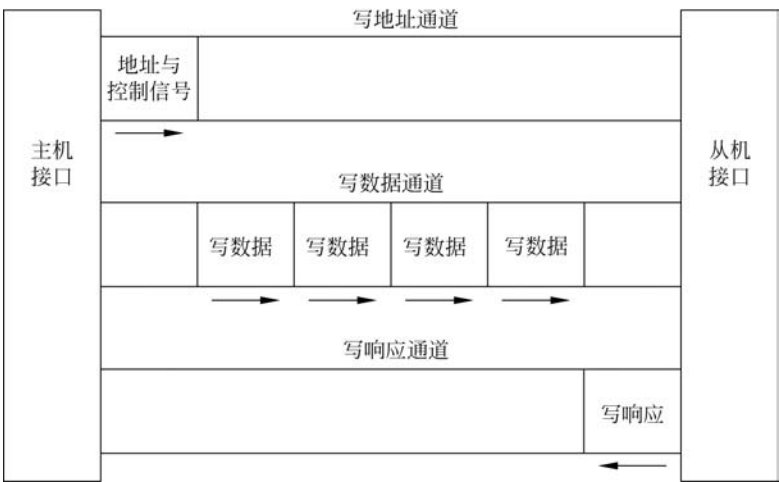


图 3.2 AXI 的写操作

1. 通道定义

AXI 协议一共定义了 5 个独立的通道,每一个通道都是由一组控制和响应信号组成的,使用双向的有效(VALID)信号和准备好(READY)信号实现握手机制。

发送端使用 VALID 信号指示何时通道中的数据或控制信号是有效的,接收端使用 READY 信号指示何时可以接收数据。读数据通道和写数据通道都包含一个结束(LAST)信号指示一个传输中的最后一个数据何时出现。

1) 读或写地址通道

读或写传输都有独立的地址通道。地址通道包含一次传输所需要的全部地址和控制信息。AXI 协议支持以下地址传输机制:

- 1~16 个可变数据个数的突发传输;
- 8~1024 位可变数据位宽的突发传输;
- 跳变、递增和非递增的突发传输;
- 专用的或锁定的传输控制;
- 系统级的缓存、缓冲控制;
- 安全的、专有的传输控制。

2) 读数据通道

读数据通道包含读数据和从机反馈给主机的读响应信息。读数据通道包含:

- 读数据总线,总线宽度可以是 8、16、32、64、128、256、512 或 1024 位;
- 用于指示读传输完成状态的一个读响应信号。

3) 写数据通道

写数据通道实现主机向从机的数据写入传输。写数据通道包含:

- 写数据总线,总线宽度可以是 8、16、32、64、128、256、512 或 1024 位;
- 为每 8 位数据提供一个有效标识位,标示数据总线的每个字节(8 位)是否有效。

写数据通道信息会被接收端缓存,因此主机在进行写传输时,无须确认上一次写传输的状态。

4) 写响应通道

写响应通道提供了一种从机对写传输作出响应的机制。所有的写传输都必须基于完成信号的状态确认传输是否成功。每次突发传输都有一次传输完成的信号响应,注意完成信号只在一次突发传输完成后才产生,而不是对突发传输中的每个独立的数据做出响应。

2. 接口和互连

如图 3.3 所示,一个典型的系统包含数个主机和从机设备,这些设备通过互连总线的形式连接在一起。

AXI 协议提供单一接口定义的形式来描述这种互连:

- 在主机与互连总线之间;
- 在从机与互连总线之间;
- 在主机与从机之间。

大多数系统使用以下三种互连方式中的一种:

- 地址和数据总线共享;



图 3.3 主机和从机互联

- 地址共享, 有多个数据总线;
- 多层互连, 即有多个地址和数据总线。

在大多数系统中, 对地址通道的带宽需求要明显低于数据通道的带宽。此类系统能够在系统性能与互连复杂性之间达成很好的平衡, 通过共享的地址总线与多个数据总线来达到平行数据传输。

3. 插入寄存器

每个 AXI 通道都只支持单向传输, 因此无须考虑不同通道复用的影响。这也使得在任意通道间插入寄存器成为可能, 虽然这样可能会产生一拍或多拍的时钟延时, 但这可以确保在时钟延时和更高的时钟频率实现之间做一些灵活的调整, 以实现系统设计的最优化。

也可以在互连中的任何必要的地方插入寄存器。若使用简单的寄存器隔离时序设计中的关键路径, 便可以更好地实现处理器和高性能存储器间直接、快速的连接。

3.1.3 基本传输

下面对 AXI 协议的每个基本传输一一做介绍。每个传输实例都有 VALID 和 READY 信号的握手机制。不论是地址或是数据的传输都在 VALID 和 READY 信号为高电平时发生。

1. 突发读传输实例

如图 3.4 所示, 这是一次 4 个数据的突发读操作。在这个传输中, 主机首先在地址通道上发送地址信息 (ARVALID 拉高时 ARADDR 有效), 然后从机在一个时钟周期之后接收到地址信息 (ARREADY 和 ARVALID 同时为高电平时, 表示从机接收到了 ARADDR)。完成地址传输后, 数据传输则在读数据通道实现。从机保持 RVALID 信号为低电平, 一直到读数据总线 RDATA 是有效的, 则拉高 RVALID 信号, 向主机发送有效的数据 RDATA。对于突发传输的最后一个读数据, 从机通过将 RLAST 信号拉高, 指示此时数据总线 RDATA 上传输的是最后一个读数据。

主机可以通过拉低 RREADY 信号来减慢从机送数据的速度。在 RVALID 为高电平时, 从机会判断此时 RREADY 信号是否也为高, 若为高, 表明此次传输的 RDATA 数据已经被接收; 若为低, 则继续保持 RVALID 为高, 并且保持当前的读数据 RDATA 不变, 直到 RREADY 拉高为止, 接着才会传送下一个有效数据。

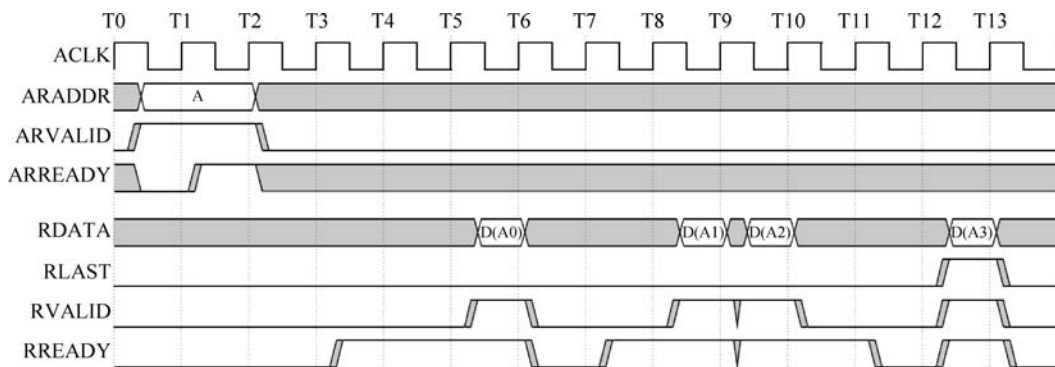


图 3.4 突发读操作

2. 连续突发读传输实例

如图 3.5 所示,在从机接收第一个地址后(ARVALID 和 ARREADY 都为高电平),主机可以接着发起第二个地址,实现连续的突发读操作。读数据通道送出的数据会遵循突发传输地址写入的先后,即“先来后到”的原则顺序传输。

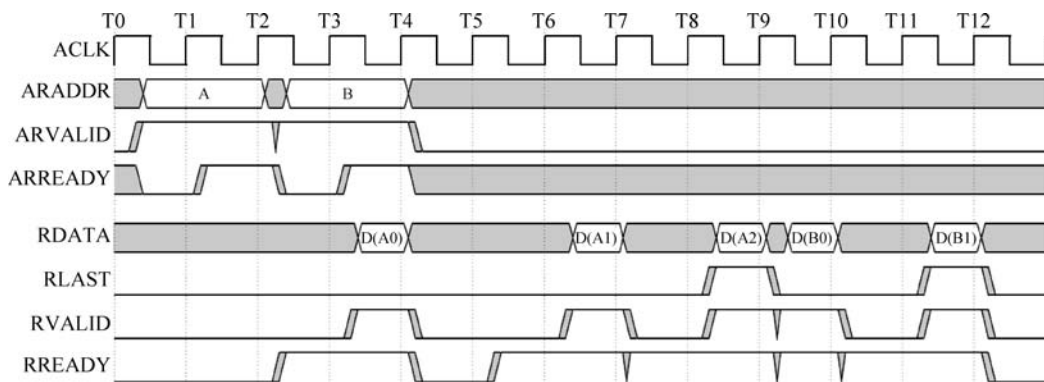


图 3.5 连续突发读操作

3. 突发写传输实例

如图 3.6 所示,这是一次 4 个数据的突发写操作。在这个传输中,主机首先在写地址通道上发送地址信息(AWVALID 拉高时 AWADDR 有效),然后从机在一个时钟周期之后接收到地址信息(AWREADY 和 AWVALID 同时为高电平时,表示从机接收到了 AWADDR)。完成地址传输后,主机接着在写数据通道实现写入数据的传输。主机拉高 WVALID 的同时,数据 WDATA 有效。在从机输出的 WREADY 为高电平且 WVALID 也为高电平时,表示当前数据 WDATA 已经被从机正确接收到了。在主机传输突发写入的最后一个数据(第 4 个数据)时,WLAST 信号需要同时拉高。当从机接收好所有的数据后,从机通过写响应通道将信息(BVALID 拉高时,BRESP 指示 OKAY 表示传输正常)反馈给主机,指示写数据已经被接收,写传输完成。

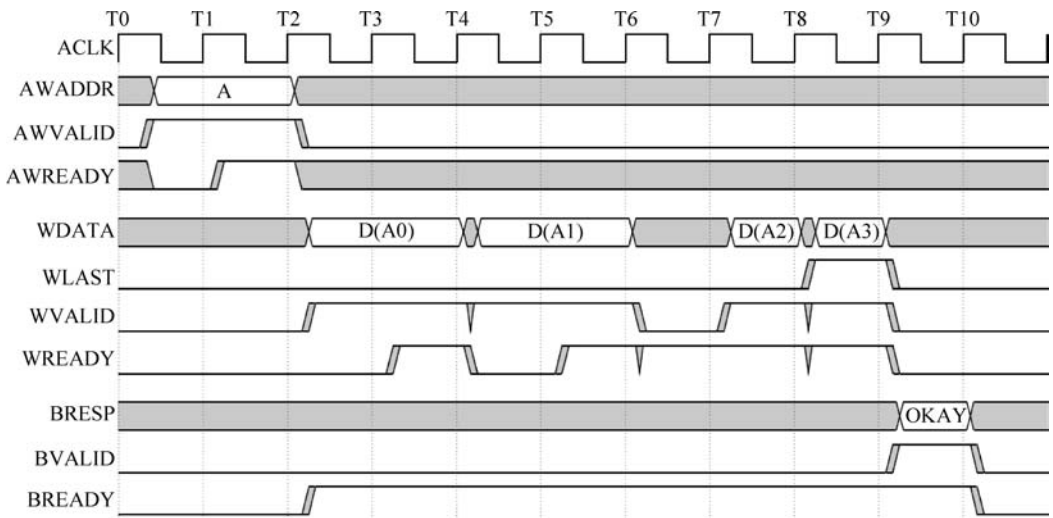


图 3.6 突发写传输操作

3.2 信号描述

1. 全局信号

全局信号及描述如表 3.1 所示。

表 3.1 全局信号及描述

信 号	发 送 端	描 述
ACLK	时钟源	全局时钟信号。所有的信号都在全局时钟的上升沿采样
ARESETn	复位源	全局复位信号。低电平有效

2. 写地址通道信号

写地址通道信号及描述如表 3.2 所示。

表 3.2 写地址通道信号及描述

信 号	发送端	描 述
AWID[3:0]	主机	写地址 ID。该信号是写地址传输的唯一标识
AWADDR[31:0]	主机	写地址。写地址总线为突发写传输的第一个有效数据传输提供地址。相关控制信号决定了后续传输的有效数据相对于第一个地址的关系
AWLEN[3:0]	主机	突发长度。突发长度提供突发传输的数据长度
AWSIZE[2:0]	主机	突发位宽。突发位宽提供突发传输中每个有效数据的位宽
AWBURST[1:0]	主机	突发类型

续表

信 号	发送端	描 述
AWLOCK[1:0]	主机	锁定类型
AWCACHE[3:0]	主机	缓存类型
AWPROT[2:0]	主机	保护类型
AWVALID	主机	写地址有效。该信号指示此时传输的是否为有效的写地址和控制信息：1=地址和控制信息是有效的；0=地址和控制信息无效。地址和控制信息保持稳定直到地址响应信号 AWREADY 为高电平
AWREADY	从机	写地址准备好。这个信号指示从机是否已经准备好接收地址和相关的控制信号：1=从机准备好；0=从机未准备好

注：对于一般的应用，AWBURST、AWLOCK、AWCACHE 和 AWPROT 通常赋一个特定的值，即设定为固定的工作模式即可。

3. 写数据通道信号

写数据通道信号及描述如表 3.3 所示。

表 3.3 写数据通道信号及描述

信 号	发送端	描 述
WID[3:0]	主机	写 ID。这个信号是写数据传输的唯一标识。WID 值必须与写传输的 AWID 值匹配
WDATA[31:0]	主机	写数据。写数据总线可以是 8、16、32、64、128、256、512 或 1024 位带宽
WSTRB[3:0]	主机	写数据有效字节数。这个信号指明写数据的每个字节数据是否写入最终的地址中。在写数据总线中每 8 位有一个WSTRB 位相对应，指示该信号写入的数据字节是否更新到最终的写地址中
WLAST	主机	最后一个数据写入指示信号。这个信号高电平时，表明一次突发写传输的最后一个数据正在传输
WVALID	主机	写有效信号。这个信号指明写数据是否有效：1=写数据有效；0=写数据无效
WREADY	从机	写准备好。这个信号指明从机是否可以接收写数据：1=从机准备好；0=从机未准备好

4. 写响应通道信号

写响应通道信号及描述如表 3.4 所示。

表 3.4 写响应通道信号及描述

信 号	来源	描 述
BID[3:0]	从机	响应 ID。这是写响应的唯一标识。BID 值必须与写传输的 AWID 值匹配
BRESP[1:0]	从机	写响应。这个信号表明写传输的状态。可用的状态是 OKAY、EXOKAY、SLVERR 和 DECERR

续表

信 号	来源	描 述
BVALID	从机	写响应有效信号。这个信号表明写响应是否有效：1=写响应有效；0=写响应无效
BREADY	主机	响应准备好。这个信号表明主机是否可以接收响应信息：1=主机是准备好的；0=主机未准备好

5. 读地址通道信号

读地址通道信号及描述如表 3.5 所示。

表 3.5 读地址通道信号及描述

信 号	来源	描 述
ARID[3:0]	主机	读地址 ID。该信号是读地址信号的唯一标识
ARADDR[31:0]	主机	读地址。读地址总线提供一个突发读传输的初始地址。只提供了突发读传输的起始地址,其余读数据的地址通过控制信号可以计算出来
ARLEN[3:0]	主机	突发长度。突发长度提供了突发读传输的数据个数
ARSIZE[2:0]	主机	突发位宽。这个信号指示了突发传输的数据位宽
ARBURST[1:0]	主机	突发类型
ARLOCK[1:0]	主机	锁定类型
ARCACHE[3:0]	主机	缓存类型
ARPROT[2:0]	主机	保护类型
ARVALID	主机	写地址有效。该信号为高电平时,写地址和控制信息有效,并将保持稳定,直到地址响应信号 ARREADY 拉高。1=地址和控制信息有效；0=地址和控制信息无效
ARREADY	从机	写地址准备好。该信号表明从机是否准备好接收地址和相关的控制信号：1=从机准备好；0=从机未准备好

6. 读数据通道信号

读数据通道信号及描述如表 3.6 所示。

表 3.6 读数据通道信号及描述

信 号	来源	描 述
RID[3:0]	从机	读 ID 标签。该信号是读数据的唯一标识。RID 值是由从机产生的,必须与读地址通道传输的 ARID 值相匹配
RDATA[31:0]	从机	读数据。读数据总线可以是 8、16、32、64、128、256、512 或 1024 位带宽
RRESP[1:0]	从机	读响应。这个信号表明读传输的状态。可用的反馈状态是 OKAY、EXOKAY、SLVERR 和 DECERR
RLAST	从机	最后一个读数据有效标志位。这个信号表明一次突发读传输的最后一个数据正在传输

续表

信 号	来 源	描 述
RVALID	从机	读有效信号。这个信号表明读数据是否有效：1=读数据有效；0=读数据无效
RREADY	主机	读数据准备好。这个信号表明主机是否可以接收从机送出的读数据：1=主机是准备好的；0=主机未准备好

3.3 握手过程

全部 5 个通道使用同样的 VALID 和 READY 握手机制来实现数据和控制信息的传输。这个双向的流控制机制确保主机和从机有效地把控好数据和控制信息的传输速度。发送端产生 VALID 信号指示数据或控制信息有效,接收端产生 READY 信号表明它可以或已经接收数据或控制信息。只有在 VALID 和 READY 信号都为高电平的时候才能进行有效的传输。

如图 3.7 所示,这是一个握手的实例。发送端送出数据或控制信息(INFORMATION),并将 VALID 信号拉高。来自发送端的数据或控制信息保持稳定直到接收端将 READY 信号拉高,指明接收端已经接收数据或控制信息。箭头表明了传输实际发生的时机。

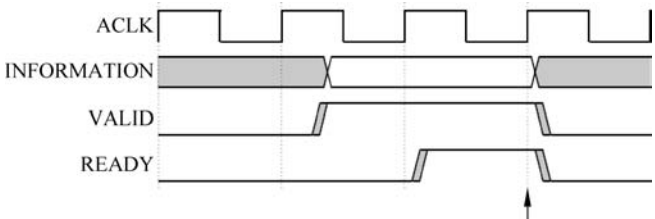


图 3.7 VALID 信号早于 READY 信号的握手操作

必须注意,若发送端要发起一次传输,不允许 VALID 信号一直等待 READY 变为高电平才执行拉高操作。正确的操作:将 VALID 拉高,一直保持高电平直到 READY 拉高,完成一个完整的握手操作。

如图 3.8 所示,接收端在 VALID 信号拉高之前,已经将 READY 信号拉高,那么数据或控制信息(INFORMATION)的传输在 1 个时钟周期就完成了。箭头表明传输发生的时机。

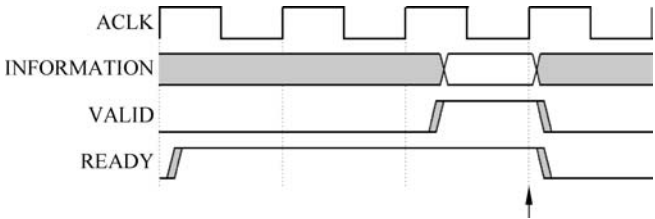


图 3.8 READY 信号早于 VALID 信号的握手操作

接收端送出的 READY 信号在 VALID 拉高之后才拉高,在这个握手机制中是允许的。如果 READY 为高电平,在 VALID 拉高之前将 READY 拉低也是允许的。如图 3.9 所示,在这个实例中,VALID 和 READY 信号同一个时钟周期拉高了,那么这正好是一次有效的数据或控制信息的传输。

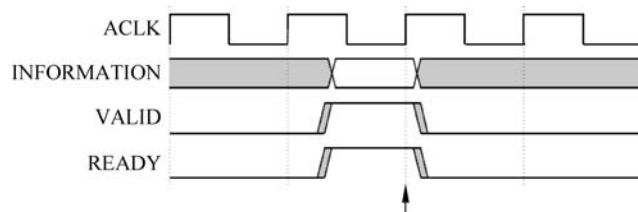


图 3.9 READY 和 VALID 信号同时拉高的握手操作