

AXI

https://support.xilinx.com/s/article/1074583?language=zh_CN#

1. 定义

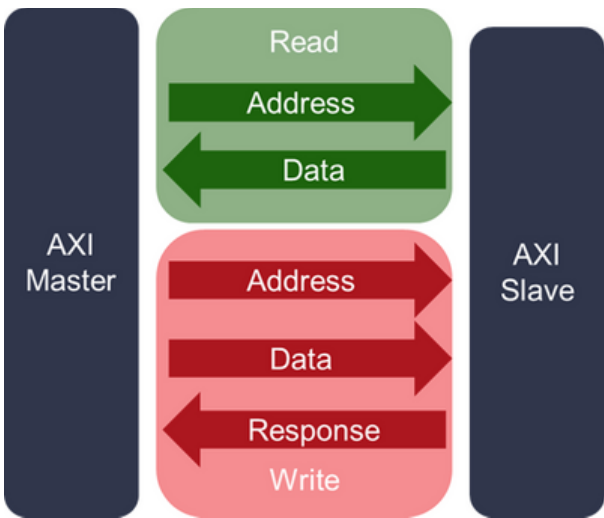
AXI 表示 Advanced eXtensible Interface（高级可扩展接口），它是由 Arm 定义的接口协议，包含在“高级微控制器总线架构 AMBA”标准中。

AXI4 接口 (AMBA 4.0) 分 3 种类型：



该图展示了 AXI 接口的分类。顶层是一个绿色的方框，标有“AXI”。从该方框下方引出三条线，分别连接到三个子方框：左侧是一个红色的方框，标有“AXI3/AXI4 Memory Map”；中间是一个深蓝色的方框，标有“AXI4 Streaming”；右侧是一个深蓝色的方框，标有“AXI4 Lite”。

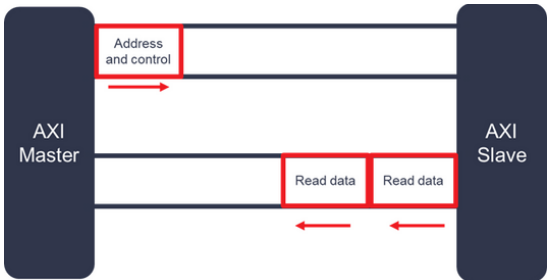
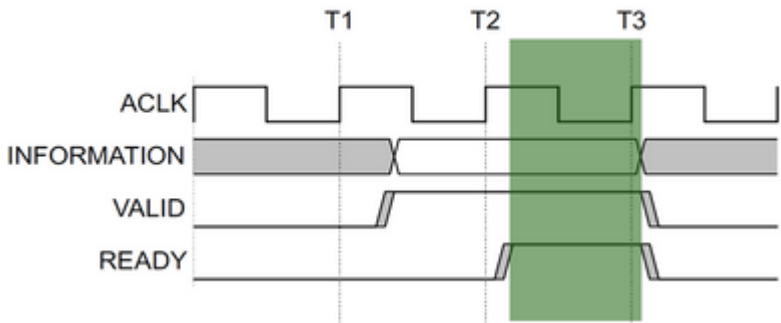
- AXI4 (AXI4-Full)：用于满足高性能存储器映射需求。
- AXI4-Lite：用于简单的低吞吐量存储器映射通信（例如，往来于状态寄存器的通信）。
- AXI4-Stream：用于高速流传输数据。



- AXI 协议定义了 5 条通道：
- 其中 2 条用于读取传输事务
 - 读地址
 - 读数据
 - 另 3 条用于写入传输事务
 - 写地址
 - 写数据
 - 写响应

通道 (channel) 是与 VALID 和 READY 信号关联的 AXI 信号的独立集合。

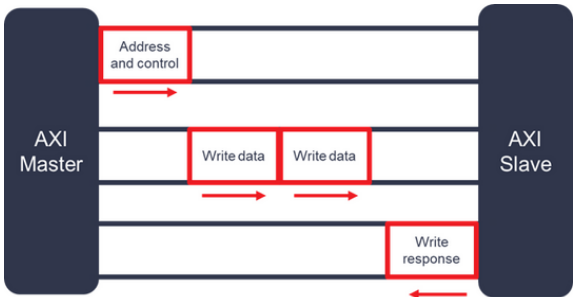
任一通道上发射的每一条数据都称为一次传输 (transfer)。当 **VALID** 和 **READY** 信号均高电平并且时钟存在上升沿时，就会发生传输。例如，在下图中，在 T3 处正在发生传输：



- AXI **读传输事务**需要在 2 条读取信道上发生多次传输。
- 首先，地址读通道 (Address Read Channel) 从主设备 (Master) 发送到从设备 (Slave)，以便设置地址和部分控制信号。

- 然后，此地址的数据通过读数据通道 (Read data channel) 从从设备发送到主设备。

请注意，根据下图所示，每个地址中可发生多次数据传输。此类型的传输事务称为突发 (burst)。



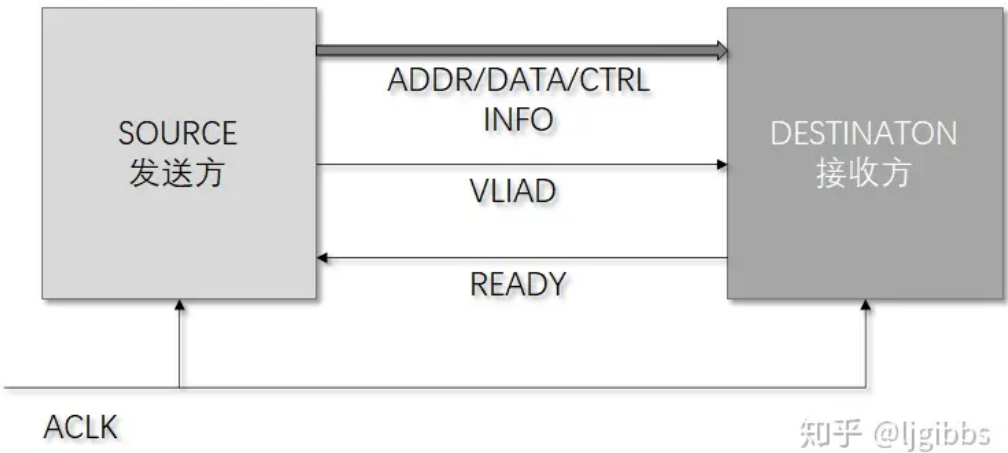
AXI **写入传输事务**需要在 3 条读取信道上存在多次传输。

- 首先，写地址通道 (Address Write Channel) 从主设备发送到从设备，以便设置地址和部分控制信号。
- 然后，此地址的数据通过写数据通道 (Write data channel) 从主设备发射到从设备。
- 最后，写入响应通过写响应通道 (Write Response Channel) 从从设备发送到主设备，以指示传输是否成功。

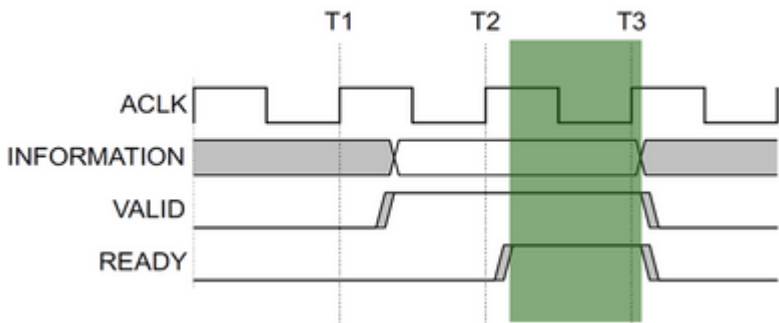
2. VALID/READY 握手机制

作为一种双向流控机制，VALID/READY机制可以使发送接收双方都有能力控制传输速率。

双向流控机制，指的是**发送方**通过置起**VALID**信号控制发送的时机与速度，**接收方**也可以通过**READY**信号的置起与否控制接收速度。



任一通道上发射的每一条数据都称为一次传输 (transfer)。当 **VALID** 和 **READY** 信号均高电平并且时钟存在**上升沿**时，就会发生传输。例如，在下图中，在 T3 处正在发生传输：



2.1 VALID/READY的三种情况

2.1.1 VALID信号先到达

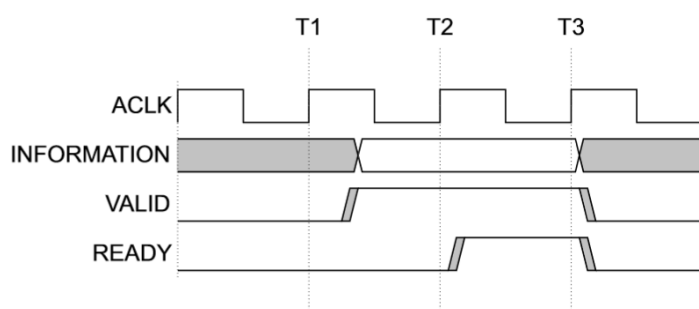


Figure A3-2 VALID before READY handshake

在这种情况下，接收方通过 READY 信号控制了传输速度，反压了发送速度。在T2前就准备好的数据直到T3时刻才被接收方接收。

在设计接收方逻辑时，检测到 VALID 信号置起，如果系统正忙，完全可以让发送方等待，**发送方在完成传输之前都不会置低 VALID 信号**，不需要考虑发送方撤销传输的可能。

! A source is not permitted to wait until READY is asserted before asserting VALID.

（发送方不能通过等待接收方 READY 信号来确定置起 VALID 信号的时机）说明了VALID信号的主动性，不受READY信号的影响。

如果VALID信号受READY的影响，那么会出现死锁（deadlock）。假如接收方在等发送方的VALID信号来拉高READY（按照ARM的协议，可以依赖发送方 VALID 信号），两方互相等，陷入死循环里，永远无法接收数据。

```

1 module axi_src...
2 //assign VALID = SRC_CONDITION && READY; //NOT permitted and may
  cause deadlock
3 assign VALID = SRC_CONDITION;//permitted
4 ...endmodule
5
6 module axi_dst...
7 assign READY = CONDITION && VALID;//permitted
8 ...endmodule

```

2.1.2 READY信号先到达

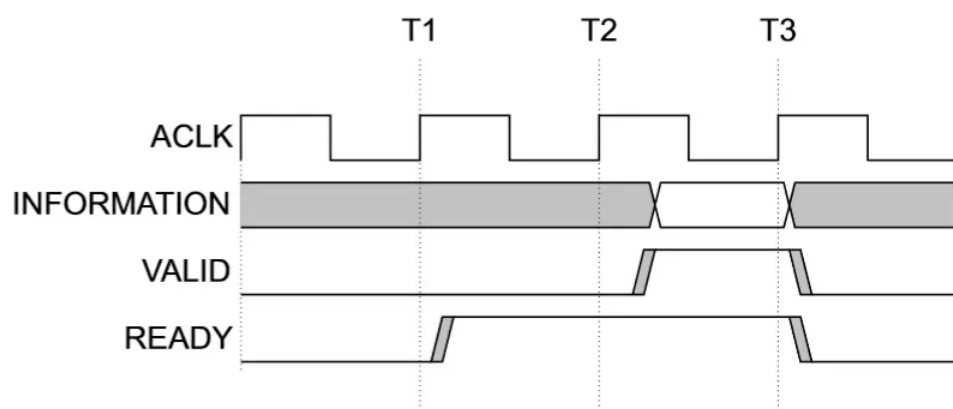


Figure A3-3 READY before VALID handshake

READY 信号很自由，可以等待 VALID 信号到来再做响应，但也完全可以在 VALID 信号到来前就置高，表示接收端已经做好了。

而且，READY 信号与 VALID 不同，接收方可以置起 READY 之后发现：其实我好像还挺忙，然后置低 READY 信号。只要此时 VALID 信号没有置起，这种操作是完全可以。

2.1.3 同时到达

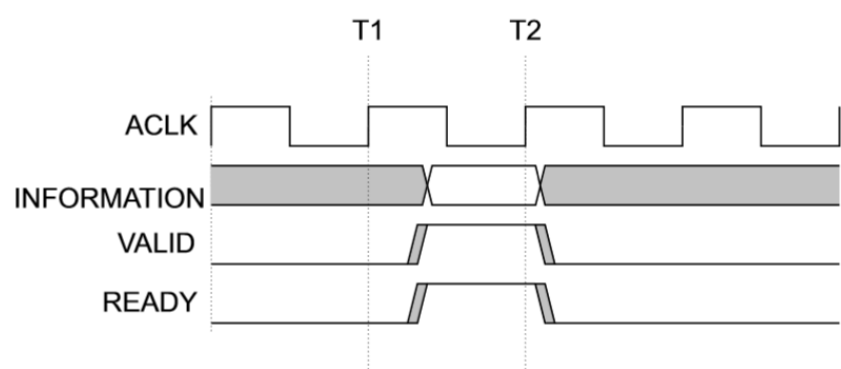
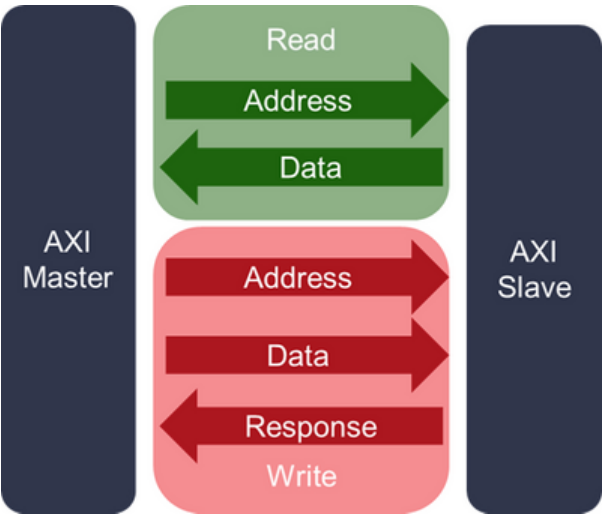


Figure A3-4 VALID with READY handshake

正常传输即可

3. 架构

3.1 5个独立通道



AXI 协议定义了 5 个独立通道，都只支持单向传输：

- 其中 2 条用于读取传输事务
 - 读地址（AR） read address
 - 读数据（RD） read data
- 另 3 条用于写入传输事务
 - 写地址（AW） write address
 - 写数据（WD） write data
 - 写响应（R） write response

通道 (channel) 是与 VALID 和 READY 信号关联的 AXI 信号的独立集合。

其中读回复，借用了读数据的通道。因此缺少1个

3.1.1 写传输（Write transcation）

Figure A1-2 shows how a write transaction uses the write address, write data, and write response channels.

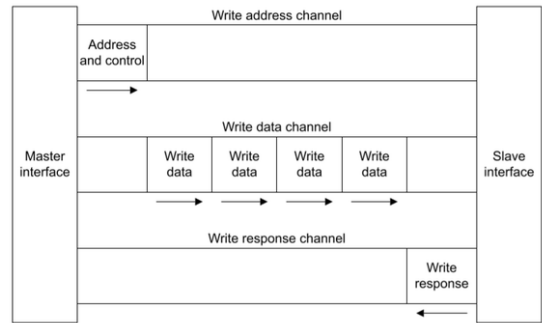


Figure A1-2 Channel architecture of writes

所有传输操作中，RC 与 WDC 的关系都如上图所示，写回复必然是在收到传输操作中最后一个写数据之后触发。

主机首先在写地址通道（AWC）上告知从机本次传输操作（transcaction，对应后文中的"传输操作"）的特性，包括地址和控制信息。

然后，在写数据通道（WDC）向从机写入数据，一次传输操作中可能包括多个数据传输（data transfer）。

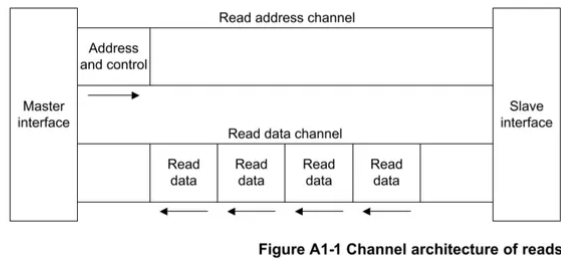
最后，从机在接收到写数据后，通过写回复通道（RC）将本次传输操作的响应告知主机。主机以收到从

机的响应信号，作为本次传输操作结束的标志。

手册强调，写回复是针对整个传输操作（transcation）的，而不是针对每个写入数据（data transfer）。

3.1.2 读传输（Read transcation）

Figure A1-1 shows how a read transaction uses the read address and read data channels.



首先主机在读地址通道（ARC）上写入本次传输操作（Transcation）待读取数据的地址以及控制信息。

从机在接收到地址后，将该地址上的数据通过读数据通道（RDC）传输给主机。

AR 虽然名字为读地址通道，但实际上仍由主机写入地址，只不过是写入要读取数据的地址。

无论是读写操作，AXI 总线支持，或者说基于**突发传输（Burst Transaction）**。简单来说，主机可以写入**起始地址**以及**突发传输长度**等信息，从机将起始地址开始，依次接收主机传输的写数据，或者读取连续地址上的数据，作为读数据传输给主机。所以上面两张图中，一次传输操作中（Transcation）中包括了一次**地址与控制信息（Address & Control）**、**多个数据（data transfer）**。

3.2 通道定义

- 都使用 VALID/READY 握手机制
- 在读写数据（WC&RC）两条数据通道中，传输突发传输（Burst Transaction）中的最后一个数据，必须要给出 LAST 信号，来标识这是此次突发传输中的最后一个数据（data transfer）。
- 读地址写地址（AR/AW） read address / write address：本次传输操作所需的地址和控制信息
- 读数据（RC） read data：从机发送给主机的读数据，以及从机对于本次读传输操作的回复
- 写数据（WC） write data：

WC 有一点 RC 所不具有的特性是拥有 STROBE 信号，用于标识写数据中有效的传输字节。即有些无效的数据，出于减少主机工作量的目的，或者在读写宽度不对称时，被放到写数据通道上和有效数据一起发送。而 STROBE 的信号的作用就是标识出这些无用的数据，告知从机不需要接收无用数据。（Master：我太懒，以至于把所有信号都送过来了）

写数据通道设计有缓存，可超前于从机响应本次传输操作，发起下一次写传输操作。

- 写响应（R） write response：用于从机将写操作响应回复给主机。所有写传输操作都需要以写回复通道上接收写响应作为完成信号。

为什么没有读相应

从**数据流向**看出来，主机在读取数据时，数据在读通道上传输，流向为从机到主机。而读回复由从机向主机报告读操作的情况，信号的数据流向也是从机到主机，所以**读回复可**

以合并在读数据通道中，搭个顺风车。

但写回复通道的数据流向就和写数据相反。写数据是从主机到从机，而写回复为从机报告写操作的完成情况，流向为从机到主机，无法合并到写数据通道中，另一方面，写回复又是不可或缺的，所以就有了一条独立的写回复通道。

3.3 通道信号

3.3.1 全局信号

ACLK，全局的时钟信号，所有的传输操作都发生在 ACLK 的上升沿。

ARESETn，全局复位信号，低电平有效。

Table A2-1 shows the global AXI signals. These signals are used by the AXI3 and AXI4 protocols.

Table A2-1 Global signals

Signal	Source	Description
ACLK	Clock source	Global clock signal. See <i>Clock</i> on page A3-36.
ARESETn	Reset source	Global reset signal, active LOW. See <i>Reset</i> on page A3-36.

3.3.2 写地址通道（AW）

写地址通道的信号可以分为 3 部分：经常用到的基础信号、突发传输的控制信号、内存访问相关以及其他的在基础阶段不是很常用的信号。

基础信号：

- AWADDR：传输操作的**起始地址**
- AWVALID 以及 AWREADY：所有通道都使用的**握手信号**。

突发传输：

- AWLEN：突发传输的长度，即在一次突发传输中**数据传输的个数**。
- AWSIZE：每次突发传输中的数据传输的**位宽**。
- AWBURST：突发传输的**类型**。

Table A2-2 Write address channel signals

Signal	Source	Description
AWID	Master	Write address ID. This signal is the identification tag for the write address group of signals. See <i>Transaction ID</i> on page A5-77.
AWADDR	Master	Write address. The write address gives the address of the first transfer in a write burst transaction. See <i>Address structure</i> on page A3-44.
AWLEN	Master	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address. This changes between AXI3 and AXI4. See <i>Burst length</i> on page A3-44.
AWSIZE	Master	Burst size. This signal indicates the size of each transfer in the burst. See <i>Burst size</i> on page A3-45.
AWBURST	Master	Burst type. The burst type and the size information, determine how the address for each transfer within the burst is calculated. See <i>Burst type</i> on page A3-45.

3.3.3 读地址通道（AR）

与3.3.2类似

3.3.4 写数据通道（WC）

WDATA 的可使用位宽可以见上文。WSTRB 信号用于标记传输数据中有效的字节，1 个WSTRB 位对应一个字节（8bit）的数据位宽，比如数据位宽为 64 位，那么WSTRB 信号的位宽就是 8位，1 个字节。

WLAST 标识一次突发传输中最后一次数据传输，如果没有正确的 WLAST 的信号，就会造成写入地址的混乱，导致从机无法正确接收写数据，从而造成**从机不再拉高 READY** 信号的现象。

Table A2-3 Write data channel signals

Signal	Source	Description
WID	Master	Write ID tag. This signal is the ID tag of the write data transfer. Supported only in AXI3. See Transaction ID on page A5-77 .
WDATA	Master	Write data.
WSTRB	Master	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus. See Write strobes on page A3-49 .
WLAST	Master	Write last. This signal indicates the last transfer in a write burst. See Write data channel on page A3-39 .
WUSER	Master	User signal. Optional User-defined signal in the write data channel. Supported only in AXI4. See User-defined signaling on page A8-100 .
WVALID	Master	Write valid. This signal indicates that valid write data and strobes are available. See Channel handshake signals on page A3-38 .
WREADY	Slave	Write ready. This signal indicates that the slave can accept the write data. See Channel handshake signals on page A3-38 .

3.3.5 写回复通道（R）

3.3.6 读数据通道（RC）

Table A2-6 Read data channel signals

Signal	Source	Description
RID	Slave	Read ID tag. This signal is the identification tag for the read data group of signals generated by the slave. See Transaction ID on page A5-77 .
RDATA	Slave	Read data.
RRESP	Slave	Read response. This signal indicates the status of the read transfer. See Read and write response structure on page A3-54 .
RLAST	Slave	Read last. This signal indicates the last transfer in a read burst. See Read data channel on page A3-39 .
RUSER	Slave	User signal. Optional User-defined signal in the read data channel. Supported only in AXI4. See User-defined signaling on page A8-100 .
RVALID	Slave	Read valid. This signal indicates that the channel is signaling the required read data. See Channel handshake signals on page A3-38 .
RREADY	Master	Read ready. This signal indicates that the master can accept the read data and response information. See Channel handshake signals on page A3-38 .

因为读回复信息在读数据通道上传递，所以集成了 RRESP 信号，用于返回读状态，值得注意的是读回复信号和读数据一样，发送方（source）为从机（slave）。

3.4 通道间的关系

AXI 的通道间需要保证以下三种联系（relationship）：

- 写回复必须在其所属传输的最后一个写数据完成后（write response must follow the last write transfer in the transaction）
- 读数据必须在接收到读地址信号后产生
- 通道间的握手需要满足通道间的握手依赖性(handshake dependencies)，目的是为了**防止死锁**（deadlock）

除了这三种联系外，通道之间保持独立关系。

3.4.1 握手

两大原则

- 发送方 VALID **一定不能依赖接收方** READY 信号
- 接收方 READY 信号可以检测到 VALID 置起后再置起有效，换句话说，**可以依赖** VALID 信号

3.4.1.1 读传输

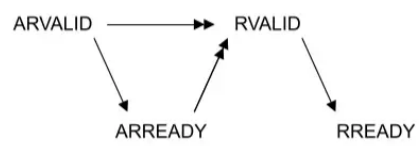


Figure A3-5 Read transaction handshake dependencies

读地址通道中主机为发送方、从机为接收方；读数据通道中主机为接收方、从机为发送方。

图中的单头箭头表示：其**指向的信号**可以在箭头**起始信号**置起之前或之后置起（无依赖）

图中的双头箭头表示：其**指向的信号**必须在箭头**起始信号**置起之后置起（指向信号**依赖起始信号**）

4. Burst 传输

用 burst 来表示一种传输模式：在一段时间中，**连续地传输多个地址相邻的数据**。此时可译为突发传输

在 AXI 传输事务（**Transaction**）中，数据以突发传输（**Burst**）的形式组织。一次突发传输中可以包含一至多个数据（**Transfer**）。每个 transfer 因为使用一个周期，又被称为一拍数据（**Beat**）。

4.1 传输事务结构（Transaction structure）

4.1.1 读写地址结构

主机首先将接下来 burst 传输的控制信息以及数据首个字节的地址传输给从机，这个地址被称为起始地址。在本次 burst 后续传输期间，从机将根据控制信息计算后续数据的地址。

!

单次 burst 传输中的数据，其地址不能跨越 **4KB** 边界。

ARLEN	Master	Burst length. This signal indicates the exact number of transfers in a burst. This changes between AXI3 and AXI4. See <i>Burst length</i> on page A3-44.
ARSIZE	Master	Burst size. This signal indicates the size of each transfer in the burst. See <i>Burst size</i> on page A3-45.
ARBURST	Master	Burst type. The burst type and the size information determine how the address of each transfer within the burst is calculated. See <i>Burst type</i> on page A3-45.

- 突发传输长度 (burst length),指一次突发传输中包含的数据传输(transfer)数量，在协议中使用 AxLen 信号控制。在 AXI4 中，INCR最大支持长度为 256，位宽为 8bit。
- 突发传输宽度(burst size),指传输中的数据位宽，具体地，是每周期传输数据的字节数量，在协议中使用 AXSIZE 信号控制。传输宽度 = 2 ^ AXSIZE

Table A3-2 Burst size encoding

AxSIZE[2:0]	Bytes in transfer
0b000	1
0b001	2
0b010	4
0b011	8
0b100	16
0b101	32
0b110	64
0b111	128

知乎 @ljgibbs

- 突发传输类型（AxBURST），类型共有 3 种，分别为 FIXED，INCR 以及 WRAP。使用 2 位二进制表示。

Table A3-3 Burst type encoding

AxBURST[1:0]	Burst type
0b00	FIXED
0b01	INCR
0b10	WRAP
0b11	Reserved

知乎 @ljgibbs

FIXED 类型中，burst 中**所有数据都使用起始地址**。该模式适合**对某个固定地址进行多次数据更新**，比如读写一个 fifo 时，读写地址就是固定的。

INCR 类型最为常用，后续数据的地址在**初始地址的基础上进行递增，递增幅度与传输宽度相同**。适合对于 RAM 等通过地址映射（mapped memory）的存储介质进行读写操作。

WRAP 类型比较特殊，首先根据起始地址得到绕回边界地址（wrap boundary）与最高地址。当前地址小于最高地址时，WRAP 与 INCR 类型完全相同，地址递增。但到递增后的地址到达最高地址后，地址直接回到绕回边界地址，再进行递增，就这样循环往复。最高地址由绕回边界地址计算得到： $\text{wrap boundary} + (\text{N_bytes} \times \text{burst_len})$

WRAP



张艺骞 6月1日 09:59
先不管了

4.1.2 数据读写传输

(1) Narrow Transfer 窄位宽数据传输

当本次传输中**数据位宽小于通道本身的数据位宽**时，称为窄位宽数据传输

在窄位宽写传输中，主机需要告知从机数据通道中哪些字节是有效的，需要使用到写数据通道中的 WSTRB 信号。WSTRB 信号中的**单个 bit 置起**，表示**对应位置上的字节有效**。

WSTRB[n] 对应 WDATA[8n+7:8n]，也就是：当 WSTRB[n] 为 1 时，WDATA[8n+7:8n]有效。

WSTRB 信号比特**位宽等于数据通道位宽的字节数量**，比如 32bit 位宽的数据通道，对应 WSTRB 信号位宽为 4bit。

对于非对齐传输，主机会进行两项操作：

- 即使起始地址非对齐，也保证所有传输是对齐的

- 在首个 transfer 中增加填充数据，将首次传输填充至对齐，填充数据使用 WSTRB 信号标记为无效