

# AMBA4.0 总线

## 概述

AMBA4.0 总线是AMBA3.0 AXI 总线的继承和提升，增加了三个新接口协议：AXI4 总线，AXI4-Lite 总线和AXI4-Stream 总线。在不使用复杂的总线桥设计下，比AMBA3.0 总线拥有更高的传输速率；满足通用情况下的接口要求；适用于复杂的存储器控制器；具备灵活的互连结构设计；兼容旧版本的AMBA 总线接口（AHB 接口和APB 接口）。

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

## AXI特点

1. 单向通道体系结构。信息流以单方向传输，简化时钟域之间的桥接，减少门数量，减少延时。
2. 支持多项数据交换。通过并行执行猝发操作，极大提高了数据吞吐能力，可在更短的时间内完成任务，在满足高性能的同时，又减小了功耗。
3. 独立的地址和数据通道。地址和数据通道分开，能对每一个通道进行单独优化，可以根据需要控制时序通道，将频率提到最高。

## AXI4总线简介

AXI4 总线是一种多通道并以突发传输为机制的总线，读地址、写地址、读数据、写数据、写反馈在不同的通道中传输。独立的传输通道能够同时读写传输，具有更小的传输延迟。不同的操作访问之间顺序可以打乱，用总线ID来表示写响应和读数据反馈的归属。主设备在上一个传输没有完成的情况下可以连续发起多个未完成读写操作。对于实际应用中的简单低速外设则可以采用AXI4-Lite 总线。

AXI4-Lite 总线也是精简的AXI4 总线。AXI4-Lite 总线标准不支持突发传输方式，可以将其突发传输长度视为1，因此不具备突发传输的信号端口。与AXI4 的规定类似，AXI4-Lite 总线也拥有五个独立的传输通道，分别为读地址通道、读数据通道、写地址通道、写数据通道和写响应通道。特别值得注意的是AXI4-Lite 总线只支持32位和64 位的数据位宽。

AXI4-Stream 总线是以突发传输为机制并且没有突发长度限制的总线。没有地址通道，没有读写使能，一般情况都是写数据从主设备到从设备的传输。

AXI4 协议包括了主设备（master）与互连结构之间，从设备（slave）与互连结构之间，以及主设备与从设备之间的连接。互连结构的设计具备多个主设备接口和从设备接口，能够处理设备之间的连接问题。

典型的互连方式有三种：分享型（share bus）的地址和分享型的数据，分享型的地址和交叉开关互连型（crossbar switch）的数据，交叉开关互连型的地址和交叉开关互连型的数据。

## AXI协议和AXI 协议定义的基础事务

1. AXI共有5个通道；
2. 每一个事务都有地址和控制信息在地址通道，用来描述传输事务的性质

3. 读事务的结构图：

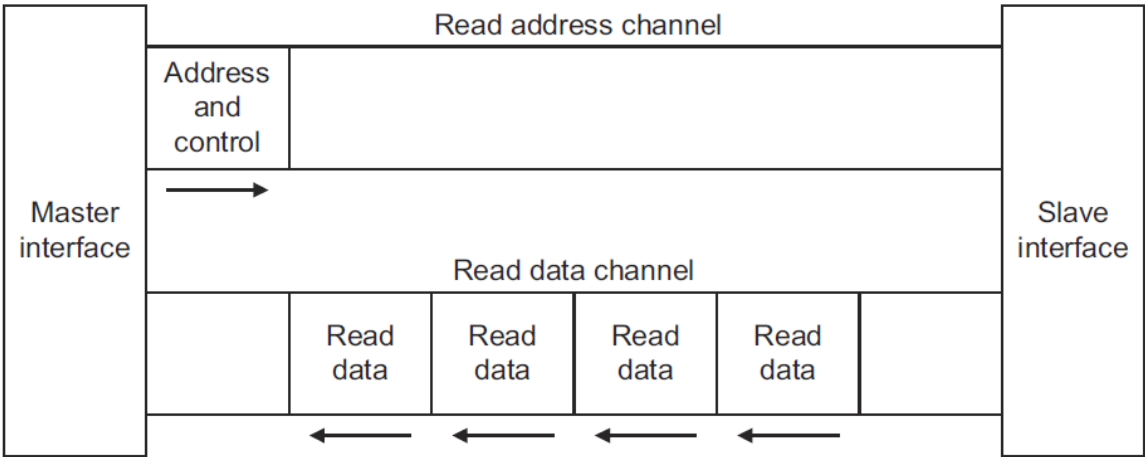


Figure 1-1 Channel architecture of reads

4. 写事务的结构图：

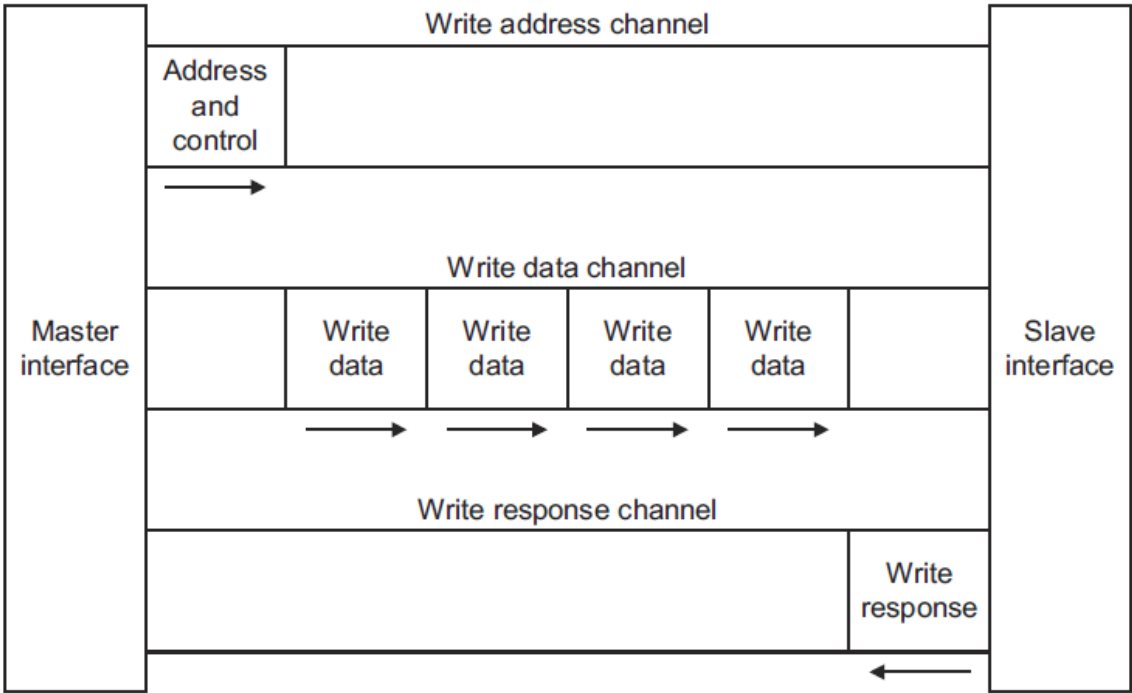


Figure 1-2 Channel architecture of writes

AXI4 总线信号描述

1. 全局信号

信号	源	描述
ACLK	Clock Source	全局时钟信号
ARESETn	Reset Source	全局复位信号，低有效

2. 写地址通道信号

信号	源	描述
AWID[3:0]	主机	写地址ID，这个信号是写地址信号组的ID tag
AWADDR[31:0]	主机	写地址
AWLEN[7:0]	主机	突发式写的长度，决定了写所传输数据的个数,最高256
AWSIZE[2:0]	主机	突发式写的大小
AWBURST[1:0]	主机	突发式写的类型
AWLOCK[1:0]	主机	锁类型
AWCache[3:0]	主机	Cache类型，指明事务的信息
AWPROT[2:0]	主机	保护类型
AWVALID	主机	写地址有效，这个信号会一直保持，直到AWREADY变为高
AWREADY	设备	写地址准备好，表明设备已经接受好接受地址和控制信号了

### 3. 写数据通道信号

信号	源	描述
WID[3:0]	主机	写ID tag，WID 的值必须和AWID 的值匹配
WDATA[31:0]	主机	写数据
WSTRB[3:0]	主机	写阀门，待确认
WLAST	主机	写的最后一个数据
WVALID	主机	写有效
WREADY	设备	写就绪

### 4. 写响应通道信号

信号	源	描述
BID[3:0]	设备	响应ID，必须和AWID的数值相同
BRESP[1:0]	设备	写响应，这个信号指明写事务的状态，如OKAY、EXOKAY、SLVERR、DECERR
BVALID	设备	写响应有效
BREADY	设备	主机可以接收信息

### 5. 读地址通道信号

信号	源	描述
ARID[3:0]	主机	读地址ID
ARADDR[31:0]	主机	读地址

信号	源	描述
ARLEN[7:0]	主机	突发式读长度
ARSIZE[2:0]	主机	突发式读大小
ARBURST[1:0]	主机	突发式读类型
ARLOCK[1:0]	主机	锁类型
ARCACHE[3:0]	主机	Cache类型
ARPORT[2:0]	主机	保护类型
ARVALID	主机	读地址有效
ARREADY	主机	读地址就绪

## 6. 读数据通道信号

信号	源	描述
RID[3:0]	设备	读ID tag，必须与ARID一致
RDATA[31:0]	设备	读数据
RRESP[1:0]	设备	读响应
RLAST[1:0]	设备	读事务的最后一个数据
RVLAID	设备	读数据有效
RREADY	主机	主机接收准备好

## 7. 低功耗接口信号

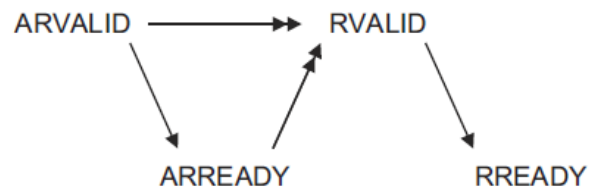
信号	源	描述
CSYSREQ	CLOCK CONTROLLER	系统低功耗请求，此信号来自系统时钟控制器，使外围设备进入低功耗状态
CSYSACK	外围设备	低功耗请求应答
CACTIVE	外围设备	

## 通道握手

- 5个通道都是使用相同的握手方式，以及READY和VALID握手信号的关系和默认值。
- 通道之间的关系
  - 地址、读、写和写响应之间的关系是灵活的。写数据可以出现在接口上早于与其关联的写地址。
  - 有两种关系一定要确定：
    - 读数据必须总是跟在与其数据相关联的地址之后；
    - 写响应必须跟在与其相关的写事务的最后出现。
- 通道握手信号之间关系 单箭头表示箭头指向的信号可以在前一个信号之前或之后置为有效，而双箭头则表示箭头指向的信号必须在前一个信号有效后才能置为有效。

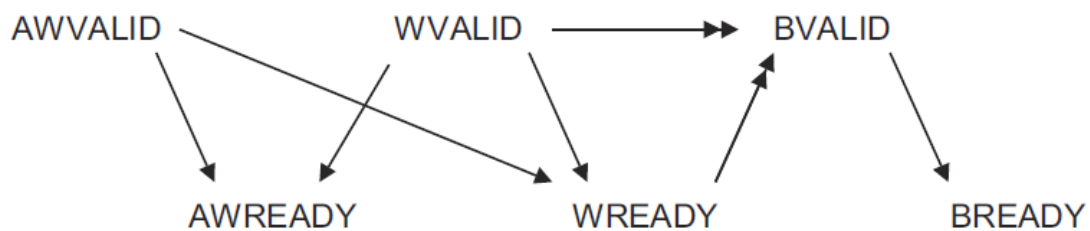
Figure 3-4 shows that, in a read transaction:

- the slave can wait for **ARVALID** to be asserted before it asserts **ARREADY**
- the slave must wait for both **ARVALID** and **ARREADY** to be asserted before it starts to return read data by asserting **RVALID**.



**Figure 3-4 Read transaction handshake dependencies**

Figure 3-5 on page 3-8 shows that, in a write transaction:



**Figure 3-5 Write transaction handshake dependencies**

## 地址计算

主要介绍AXI突发式读写的类型和一次突发读写事务如何计算地址和byte lanes。

1. 突发式读写地址必须4KB对齐；
2. AWLEN 和 ARLEN 指定换一次突发传输所传输数据的个数

Table 4-1 Burst length encoding

ARLEN[3:0] AWLEN[3:0]	Number of data transfers
b0000	1
b0001	2
b0010	3
.	
.	
.	
b1101	14
b1110	15
b1111	16

3. AWSIZE 和 ARSIZE 信息制定了每一个时钟节拍所传输的data bytes

Table 4-2 Burst size encoding

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

3. 突发式传输模式 AXI 协议定义了三种突发式读写类型：固定式的突发读写、增值式突发读写、包装式突发读写。用ARBURST和AWBURST来选择突发式读写的类型。具体的信息如下：

Table 4-3 Burst type encoding

ARBURST[1:0] AWBURST[1:0]	Burst type	Description	Access
b00	FIXED	Fixed-address burst	FIFO-type
b01	INCR	Incrementing-address burst	Normal sequential memory
b10	WRAP	Incrementing-address burst that wraps to a lower address at the wrap boundary	Cache line
b11	Reserved	-	-

- 固定式的突发读写指的是地址固定，每一次的传输地址相同。是对一个相同地址的位置进行存取，例如FIFO
- 增值式突发读写是每一次读写的地址都比上一次的地址增加一个固定值。
- 包装式突发读写和增值式类似，地址是包数据的低地址达到一个包边界，有两个限制
  - 其实地址必须以传输的SIZE对其
  - 突发读写长度必须是2，4，8，16

#### 4. 关于地址的计算公式

##### 1. 名词定义：

名词	含义
start_address	主机发送其实地址
Number_Bytes	每一次数据传输最大数据字节数
Data_Bus_Bytes	数据总线字节数量
Aligned_Address	起始地址对齐版本
Burst_Length	一次突发传输数据个数
Address_N	每一次突发事件读写所传输的地址数量，2-16
WRAP_BOUNDARY	包装式突发读写的最低地址
Lower_Byte_Lane	传输的最低地址的byteLane
Upper_Byte_Lane	传输的最高地址的byteLane
Int (X)	对x乡下取整

## 控制信息

- Cache
- 保护单元

## AXI 的设备响应

- AXI 对读事务和写事务都有响应。对于读事务，读响应和读数据一起发送给主机，而将写事务响应通过写响应通道传送，AXI的响应类型有：OKAY、EXOKAY、SLVERR、DECERR。

2. 通过RRESP，BRESP来编码响应信号

RRESP[1:0] BRESP[1:0]	Response	Meaning
b00	OKAY	Normal access okay indicates if a normal access has been successful. Can also indicate an exclusive access failure.
b01	EXOKAY	Exclusive access okay indicates that either the read or write portion of an exclusive access has been successful.
b10	SLVERR	Slave error is used when the access has reached the slave successfully, but the slave wishes to return an error condition to the originating master.
b11	DECERR	Decode error is generated typically by an interconnect component to indicate that there is no slave at the transaction address.

1. OKAY : 正常读取成功

2. EXOKAY: 独占式存取

3. SLVERR: 设备错误

4. DECERR: 译码错误
3. AXI协议要求即使传输错误，也要将数据数据传输完成。

多地址与乱序操作

AXI协议支持无序事务完成和发出多个未完成的地址。这些特性可以实现高性能的互连，最大限度地提高数据吞吐量和系统效率。

ID信号支持无序事务，允许每个端口充当多个有序端口。必须对具有给定ID的所有事务进行排序，但是对具有不同ID的事务的排序没有限制。五个事务id。

能够发出多个未完成的地址意味着，主服务器可以发出事务地址，而不需要等待较早的事务完成。这个特性可以提高系统性能，因为它支持事务的并行处理。 不按顺序完成事务的能力意味着更快的内存区域的事务可以在不等待更早的内存区域的事务的情况下完成。该特性还可以提高系统性能，因为它减少了事务延迟的影响。 事务的重新排序总是相对于其他事务进行的。没有在突发事件中重新排序数据传输的工具。定义突发控制的地址和控制信号，突发控制中的传输顺序。

1. 5种事务IDs:

1. AWID : 写地址群组信号

2. WID : 写ID tag在写事务中，与写数据一起，主机传送一个WID去匹配与地址相一致的AWID

3. BID : 写响应事务中。设备传送BID去匹配与AWID和WID 相一致的事务。

4. ARID : 读地址群组信号

5. RID : ID tag在读事务中，设备传送与RID 去匹配ARID相一致的事务。
2. 主机可以

传输ID字段

AXI协议提供了一个ID字段，使主人能够发出许多单独的事务，每个事务都必须按顺序返回。master可以使用事务的ARID或AWID字段来提供有关master的订购需求的附加信息。管理交易次序的规则如下:

1. 来自不同主机的事务没有顺序限制。他们可以按任何顺序完成。

2. 来自同一主机，但ID值不同的事务没有顺序限制。他们可以按任何顺序完成。

3. 具有相同AWID值的写事务序列的数据必须按照主发出地址的相同顺序完成。

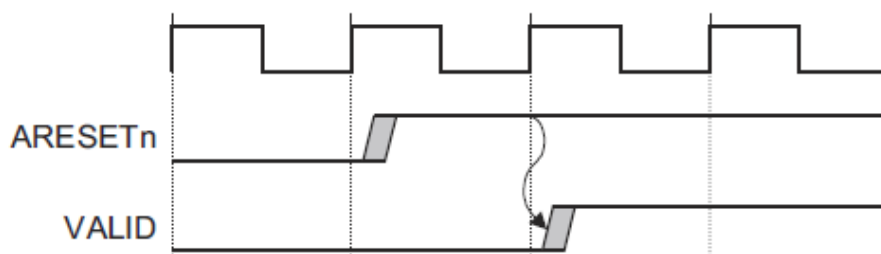
4. 必须返回具有相同ID的读事务序的数据。



5. 在具有相同AWID和ARID的读和写事务之间没有顺序限制。如果主服务器需要一个顺序限制，那么它必须确保在发出第二个事务之前完全完成第一个事务。

## 复位时序

1. 复位期间，所有valid信号要为低
2. 主机接口必须将ARVLIAD、AWVLIAD、WVALID，在第一个ACK的上升沿。



**Figure 11-1 Exit from reset**

## 参考文献

1. [AMBA AXI4 总线的研究与实现](#)
2. [AMBA AXI Protocol Version 2.0](#)
3. [AXI3与AXI4区别](#)
4. [ZYNQ的AXI\\_Lite 总线详解](#)