### 1 总线信号

#### 1.1 系统信号

* ACLK：AXI总线全局时钟信号；所有组件均在上升沿采样信号和驱动信号；
* ARESETn：全局复位信号，低电平有效；

#### 1.2 写地址通道

* AWID：写传输id标识，位宽自定义；
* AWADDR：写地址总线，位宽2幂次方；
* AWLEN：突发传输长度，位宽4bit(AXI3)或8bit(AXI4)；
* AWSIZE：每次传输数据位宽，位宽3bit；
* AWBURST：突发传输类型，位宽2bit；
* AWLOCK：锁传输，进行原子访问，位宽1bit；
* AWCACHE：访问地址内存属性，位宽4bit；
* AWPROT：访问保护信号，位宽3bit；
* AWVALID：写地址通道有效信号，位宽1bit；
* AWREADY：slave准备接收数据信号；

#### 1.3 写数据通道

* WID：写数据id标识符，位宽自定义；
* WDATA：写数据总线，位宽2幂次方，最大为1024bit；
* WSTRB：写探测信号，指示哪些字节有效；
* WLAST：突发传输最后一拍；
* WVALID：同上；
* WREADY：同上；

#### 1.4 写回应通道

* BID：写回应id标识，位宽自定义；
* BRESP：指示传输完成情况，位宽2bit；
* BVALID：slave写回应通道有效信号，位宽1bit；
* BREADY：master准备接收信号，位宽1bit；

#### 1.5 读地址通道

* ARID：读传输id标识，位宽自定义；
* ARADDR：读地址总线，位宽2幂次方；
* ARLEN：突发传输长度，位宽4bit(AXI3)或8bit(AXI4)；
* ARSIZE：每次传输数据位宽，位宽3bit；
* ARBURST：突发传输类型，位宽2bit；
* ARLOCK：锁传输，进行原子访问，位宽1bit；
* ARCACHE：访问地址内存属性，位宽4bit；
* ARPROT：访问保护信号，位宽3bit；
* ARVALID：master读地址通道有效信号，位宽1bit；
* ARREADY：slave准备接收数据信号；

#### 1.6 读回应通道

* RID：读回应id标识，位宽自定义；
* RDATA：读数据总线，位宽2幂次方；
* RRESP：传输完成状态，位宽2bit；
* RLAST：突发传输最后一拍，位宽1bit；
* RVALID：slave读回应通道有效信号，位宽1bit；
* RREADY：master准备接收信号，位宽1bit；

### 2 接口信号特点

#### 2.1 时钟和rst信号

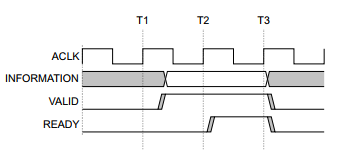
**时钟信号：**AXI只支持单一时钟；上升沿采样输入信号和改变输出信号；

**重置信号：**可以异步设置为低电平，只能同步设置为高电平；重置期间master必须将ARVALID、AWVALID和 WVALID信号设置为低电平；slave必须将RVALID和BVALID设置为低电平；

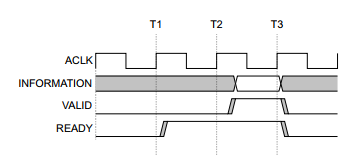
#### 2.2 基础读写操作

##### 2.2.1 握手机制

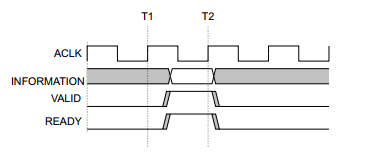
发送数据方将valid设置为高电平表示信号有效，接收方设置ready为高电平表示可以接收数据。只有在valid和ready均为高电平时，数据传输必须才会发生。



如上图所示，数据传输只会在T3上升沿处发生。T1时刻发送方传输信息并将valid设置为高电平；T2时刻接收方检查valid为高电平并将ready设置为1；在T3时刻



如上图所示，接收方也可以提前设置ready信号为高表示可以接收数据。T1时刻接收方设置ready为高电平，T2时刻发送方发送数据并设置valid为高电平；T3时刻接收方检测valid信号为高并接收数据。接收方也可以在发送方设置valid信号前取消ready信号。



如上图所示，接收方和发送方都在T1时刻设置信号，在T2时刻进行数据传输。

##### 2.2.2 通道握手信号

1. **写地址通道**

发送方传输地址数据和控制信息时必须设置AWVALID信号；直到检测到接收方设置AWREADY信号的时钟上升沿才可以取消AWVALID信号。建议AWREADY信号默认为高电平。

1. **写数据通道**

进行突发传输时，发送方必须在最后一次传输数据时将WLAST设置为高电平。

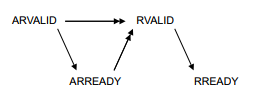
#### 2.3 各通道之间的关系

**AXI协议规定**：写回应总是在写数据传输之后；读取数据必须在其读取地址传输之后；通道握手信号必须符合一定的依赖关系。

当master发出写请求时必须能够提供写数据并且不依赖于其它传输，读请求同理。但是master可以使用相同id要求读数据按顺序返回。

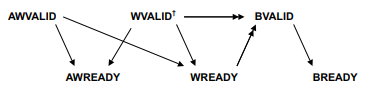
发送方的读写请求均不受其它数据传输的影响。AXI总线中valid信号不依赖ready信号；可以等待valid信号来接受数据。

1. **读请求**

****

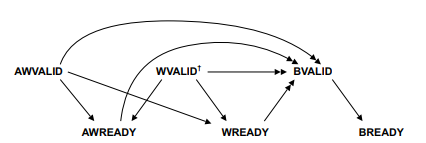
如图所示，双方必须完成地址传输后才可以进入数据传输，不允许地址信息还未发出就进行数据传输。

1. **AXI3写请求**



同理，master进行写操作时需要双方依次完成地址传输、数据传输和回应传输阶段。

1. **AXI4和AXI5写请求**



AXI4和AXI5中，slave在设置BVALID之前AWVALID、AWREADY、WVALID和WREADY必须均被设置，以防slave未接收到任何地址时响应写请求。

#### 2.4 传输数据结构

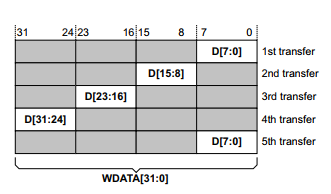
##### 2.4.1 地址组成

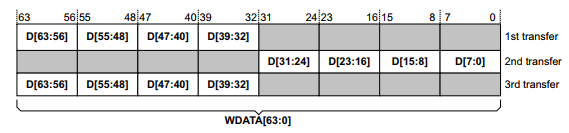
AXI总线支持突发传输，但不允许跨4KB地址边界。在突发传输时，接收方必须能够计算相应地址。突发长度必须为2、4、8或16；不支持提前终止突发传输。在进行写操作时可以将strb设置为0来减少数据传输；读操作时可以直接丢弃数据。

##### 2.4.2 数据读写结构

1. **窄传输**

当master发出比数据总线宽度更小的传输时，地址和控制信息决定传输使用的字节通道：在递增或wrap突发传输时，每次传输使用不同的字节通道；在Fixed突发传输时每次传输使用相同的字节通道。



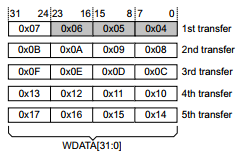


1. **字节序不变**

对于数据结构中的任何多字节元素：元素使用相同的连续字节存储，与数据的字节序无关；字节序决定了内存中这些字节的顺序；数据传输时不会改变字节序。

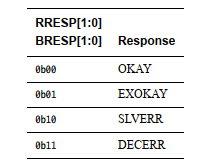
1. **非对齐传输**

AXI支持非对齐传输。Master可以使用对齐的地址和strb信号来表示未对齐的起始地址。



##### 2.4.3 数据传输回应

AXI使用RRESP[1:0]和BRESP[1:0]分别作为读回应和写回应信号。



SLVERR表示接收方出现错；DECERR表示解码错误。

### 3 传输属性

#### 3.1 传输类型和属性

内存设备：需要正确处理每一次传输；

外部设备：外部设备具有自己实现的访问方法；这些设备的访问需要兼容协议；

ARCACHE和AWCACHE信号指定传输属性，包含传输如何在系统中进行以及系统缓存如何处理传输。

#### 3.2 内存属性

1. **不可缓冲设备内存**

* 写响应必须从最终目的地获取；
* 读数据必须从最终目的地获取；
* 传输是不可修改的；
* 不能预读取；写操作不能合并；

1. **可缓冲设备内存**

* 写入响应可以从中间节点获取；
* 写操作必须及时在最终目的地可见；
* 读数据必须从最终目的地获取；
* 传输是不可修改的；
* 不能预读取，写操作不能合并；

1. **不可缓存不可缓冲的普通内存**

* 写响应必须从最终目的地获取；
* 读数据必须从最终目的地获取；
* 传输是可修改的；
* 写操作可以合并；

### 4 传输标识

#### 4.1 AXI传输标识

Master通过使用传输标识划分需要按顺序进行的传输。对于相同标识的传输必须按顺序进行，而不同标识则不需要。使用标识进行传输时不需要等待上一次数据传输完成，这可以提高系统性能。

#### 4.2 标识信号

1. **读数据排序**

发送方必须保持返回数据RID与ARID一致。互连结构必须确保同一ARID的数据传输按照地址发出顺序进行。

1. **写数据排序**

Master必须按照地址发出的顺序传输写入的数据。互连结构必须使不同数据传输按照相应顺序发出。

### 5 AXI排序模型

#### 5.1 概述

AXI排序模型基于传输标识，即ARID和AWID。在同一通道中，具有相同id和目的地数据传输需要保序。返回相同id的回应必须与请求顺序保持一致。如果Master要求在没有排序规定的数据传输之间进行排序，则必须等待第一次数据传输完成后才能进行接下的数据传输。

#### 5.2 内存和设备

1. **内存区域属性**

* 读取内存返回上一次写入的数据；
* 写入数据会更新接下来读取的数据；
* 读写内存不会对其它地方产生影响；
* 任何位置的读写操作都可以被观测；
* 内存读写大小对于其单拷贝原子性大小；

1. **设备区域属性**

* 从外设地址区域读取并不一定返回写入该地址的最后一个值；
* 对外设地址区域的字节地址进行写操作不一定会将该地址的值更新为后续读操作获得的新值；
* 访问外设地址区域可能会对该区域内的其他地址产生边界作用；

#### 5.3 传输和排序

协议只保证同一内存区域的读写操作是按顺序的；跨越多个区域的读写操作需要多个排序保证；对外围设备的读写操作必须包含在该区域内。

#### 5.4 观察

* 如果W2在W1之后生效，则称写操作W1将被写操作W2观察；
* 当W2在W3之后，如果R1从写操作W3返回数据，则称读操作R1将被写操作W2观察到；
* 当W3在W1后面，如果R2从写操作W1或写操作W3返回数据，则称读操作R2观察写操作W1；

#### 5.5 排序保证模型

##### 5.5.1 收到回应之间的保证

* 当DW2在DW1之后发出并且到达相同的外设区域时，则设备写操作DW1保证在写操作DW2之前到达目的地；
* 当DR2在DR1之后发出并且到达相同的外围区域时，则设备读取操作DR1保证在设备读取操作DR2之前到达目的地；
* 当W2在W1之后发出并且到达相同的内存位置时，则写操作W1必须被写操作W2观察到；
* 当R3在R2之后发出并且到达相同的内存位置时，则读操作R2观察到的写操作W1必须被读操作R3观察到；

##### 5.5.2 收到回应的保证

* 对读请求的响应必须保证对来自任何master的后续读或写请求都是可见的；
* 对写请求的响应必须保证对任何master的后续读或写请求都是可见的；

##### 5.5.3 回应排序保证

* 当R2在R1之后由同一个master发出且具有相同ID，则读操作R1保证在对读操作R2的响应之前接收响应；
* 当W2在W1之后由同一个master发出且具有相同的ID，则写W1保证在对写W2的响应之前收到响应；

#### 5.6 排序要求

##### 5.6.1 slave排序请求

对于外设内存，外设位置的传输执行顺序是实现定义的。执行顺序通常与到达顺序相相同，但这不是必需的。

普通内存，相同ID且同一位置的请求按照fifo顺序进行处理，回应也相同：

* W1和W2具有相同ID且同一位置，当先接收到W1请求，则W1必须在的W2之前进行；
* W1和W2具有相同ID且同一位置，当先接收到W1回应，则W1必须在的W2之前进行；

##### 5.6.2 互连排序请求

基本同上。

### 6 原子访问

#### 6.1 单拷贝原子大小

AXI4协议引入单拷贝原子大小，定义了传输原子性更新的最小字节数。AXI4协议要求数据必须以至少单拷贝原子大小的块进行传输。任何master都不能观察到原子数据的部分更新。

AXI4协议可以将系统设计为支持不同的单拷贝原子性大小。不同的组件组可以具有不同的单拷贝原子大小，以便组内进行通信。Strb信号不会影响单拷贝原子大小。

#### 6.2 独占访问

#### 6.3 锁访问

#### 6.4 原子访问信号

### 7 AXI4-Lite接口协议

#### 7.1 协议定义

AXI4-Lite关键功能：

* 所有传输的突发长度为1；
* 所有数据访问都使用数据总线的全宽度，32位或64位；
* 所有的访问都是不可修改的，不可缓冲的；
* 不支持独占访问。

### 参考资料

[1] AMBA AXI and ACE Protocol Specification.