AMBA 4 AXI4-流协议

版本：1.0

规范

序言

序言介绍了 AMBA4 AXI4-流协议规范。包含以下章节：

- *关于本文档*
- *反馈*

关于本文档

本文档讲述 AMBA4 AXI4 流协议规范。

计划的读者群

本文档面向打算了解先进微控制器总线结构(AMBA)的硬件和软件工程师，以及设计能够兼容 AMBA 4 AXI4 流协议系统的工程师。

使用本文档

本文档包含以下章节：

1 介绍

介绍 AXI4 流协议，并给出了一个流类型的例子。

2 接口信号

讲述了 AXI4 流信号以及管理信号的标准规则。

3 默认信号要求

讲述了默认信号的要求。

4 传输交错和排序

讲述了流交错和排序约束。

附录 A 与 AXI4 写数据通道比较

讲述了 AXI4 流接口和 AXI4 写数据通道之间的主要区别。

附录 B 版本

讲述了本文档发行版本之间的变化。

约定

本文档使用的约束描述如下：

- *排版*
- *时序图*
- *信号*

排版

排版约定如下：

斜体	突出了重要注意事项，介绍特殊术语，标示了内部前后对照、以及引用。
加粗	突出接口元素，比如菜单名。标示信号名。也用于分类表中的术语。
monospace 字体	表示可以用键盘输入的文本，比如命令、文件和程序名，以及源代码。
monospace 字体	表示允许的命令或选项的缩写。可以输入下划线文字代替命令或选项的全名。
monospace 斜体字体	代表 monospace 字体表示的参数，该参数可以被一个特殊值代替。
monospace 加粗字体	当用于示例代码以外时表示语言关键字。

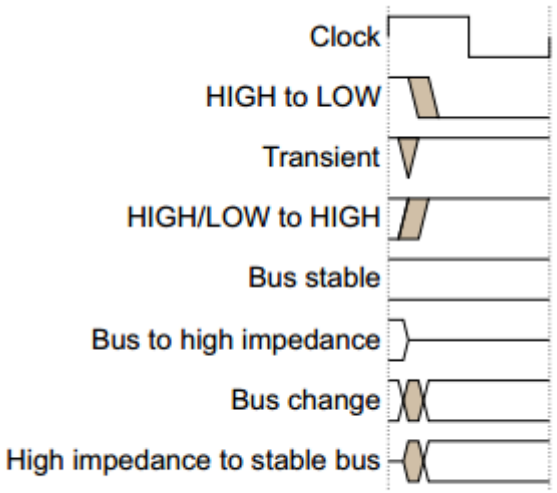
<and>　　出现在代码或代码段中，代表汇编语法中可替换的术语。例如：

MRC　p15, 0　<Rd>，　<CRn>，　<CRm>，　<Opcode_2>

时序图

以下图“Key to timing diagram conventions”解释了时序图中使用的约定。当有变化发生时会清晰标示出来。不能假定途中没有明确标示的任何时序信息。

共享的总线和信号区域没有被定义，因此在共享区域内，总线或信号可以假定为任何值。实际电压水平不是很重要，因为不会影响正常操作。



Key to timing diagram conventions

时序图有时候会将一个单 bit 信号同时显示为高和低，看起来类似于总线变化，如上图所示。如果一个时序图中一个单 bit 信号以这种方式显示，则它的值不会影响伴随的描述。

信号

信号约定有：

信号电平　　一个断言的信号的电平取决于该信号是高有效还是低有效。断言的意思是：

- 对于高有效的信号，为高
- 对于低有效的信号，为低

小写 n　　在信号名的开头或结尾，表示该信号为低有效

反馈

ARM 欢迎对于本产品及文档的反馈。

反馈关于本产品

如果你有任何关于本产品的看法和建议，请联系你的供应商并给出：

- 产品名
- 产品修订版本或版本
- 提供尽可能多的解释说明信息，如果合适的话，包括问题以及调试过程。

反馈关于内容

如果你有关于内容的意见，请发送邮件给 errata@arm.com，并给出：

- 标题：AMBA 4 AXI4-Stream Protocol Specification
- 数字：ARM IHI 0051A
- 你对内容有意见的页码
- 关于你的意见的一个简洁的解释说明

1 介绍

本章描述了 AXI4-流协议，并给出一些流类型的例子。本章包含以下章节：

- *关于 AXI4-流协议*
- *数据流*

1.1 关于 AXI4-流协议

AXI4-流协议作为一个标准接口，用于连接进行数据交换的组件。接口可以用来连接一个单一的主机，主机向接收数据的单一从机发送数据。协议也可用于连接若干个主机和从机的组件。协议支持共用一组信号线的多个数据流，允许构建一个通用互联(generic interconnect)，可以执行 upsizing、downsizing 以及路由操作。

AXI4-流接口也支持多种不同的流类型。流协议定义了传输和包之间的关系。

1.1.1 字节定义

以下字节定义用于本规范：

数据字节 在源设备和目标设备之间传输的、包含有效信息的一个字节的数据。

位置字节 在流中标示数据字节相对位置的一个字节。这是一个占位符，不会包含任何在源设备和目标设备之间传输的实质性的数据值。

空字节 在流中的一个字节，不包含任何数据信息或任何有关数据字节相对位置的信息。

1.1.2 流术语

以下流术语用于本规范：

传输 通过 AXI4-流接口进行的一个单一数据传输。一个单一数据传输由 **TVALID**，**TREADY** 握手定义。参见*握手过程*。

包 通过 AXI4-流接口被一起传输的一组字节。包类似于 AXI4 的突发。一个包可以由一个单一传输或多个传输组成。基础组件可以使用包来更有效地处理 packet-sized 组中的一个流。

帧 一个 AXI4-流中最高级别的字节编组。一个帧包含整数个数的包。一帧可以包含很大数量的字节数，例如，一个完整的视频帧缓存。

数据流 从一个源设备到一个目标设备传输的数据。

一个数据流可以是：

- 一系列独立字节传输
- 组合在包中的一系列字节传输

1.2 数据流

数据流有多种形式。本章节提供了一些使用定义的 AXI4-流类型的不同数据流风格的例子。

1.2.1 字节流

一个字节流是若干个数据和空字节的传输。在每个 **TVALID**，**TREADY** 握手期间，可以传输任何数量的数据字节。空字节不具有任何意义，可以被插入流中，或者从流中移除。图 1-1 展示了字节流的两个例子。图中，每一竖列代表在一个单一传输中的字节，在这个例子中使用了一个 4-byte 宽的数据总线。列的时间序为从左到右。

注意：

因为空字节不传输信息，也不出现在流中，所以图 1-1 中的两个例子传输的信息是相同的。

Null	Null	D-07	D-0A	Null	D-0F
D-01	Null	D-06	D-09	Null	D-0E
Null	D-03	D-05	D-08	D-0C	Null
D-00	D-02	D-04	Null	D-0B	D-0D

D-02	D-06	Null	D-0B	Null	Null
Null	D-05	Null	D-0A	D-0E	D-0F
D-01	D-04	Null	D-09	D-0D	Null
D-00	D-03	D-07	D-08	D-0C	Null

Figure 1-1 Example of byte streams

1.2.2 连续对齐的流

一个连续对齐的流是一个包含若干个数据字节的传输，这个传输中的每个包中都没有位置字节或空字节。图 1-2 展示了一个连续对齐流的例子。

D-03	D-07	D-0B	D-0F	D-13
D-02	D-06	D-0A	D-0E	D-12
D-01	D-05	D-09	D-0D	D-11
D-00	D-04	D-08	D-0C	D-10

Figure 1-2 Example of a continuous aligned stream

1.2.3 连续非对齐的流

一个连续非对齐的流是一个包含若干个数据字节的传输，这个传输中，每个包的第一个数据字节和最后一个数据字节之间没有位置字节。图 1-3 展示了连续非对齐流的两个例子。

注意：

一个连续非对齐的流可以在包的开始、结尾或开始和结尾处有任意数量的相邻位置字节

D-03	D-07	D-0B	D-0F	Position
D-02	D-06	D-0A	D-0E	Position
D-01	D-05	D-09	D-0D	D-11
D-00	D-04	D-08	D-0C	D-10

D-00	D-04	D-08	D-0C	D-10	Position
Position	D-03	D-07	D-0B	D-0F	Position
Position	D-02	D-06	D-0A	D-0E	Position
Position	D-01	D-05	D-09	D-0D	D-11

Figure 1-3 Example of continuous unaligned streams

1.2.4 稀疏的流

一个稀疏的流是一个包含若干个数据字节及位置字节的传输。所有的数据及位置字节都必须被保持并从源设备传送到目标设备。图 1-4 展示了一个稀疏流的例子。

注意：

一个稀疏流可以包含任何数据字节和位置字节的混合，但一般情况下多数字节为数据字节。一个稀疏流并不意味着只有少数字节是数据字节。

D-03	D-07	Position	D-0F	D-13
Position	D-06	D-0A	Position	D-12
D-01	Position	D-09	D-0D	Position
D-00	D-04	Position	D-0C	D-10

Figure 1-4 Example of a sparse stream

2 接口信号

本章讲述了 AXI4-流接口的信号要求。包含以下章节：

- *信号列表*
- *传输信号*
- *数据信号*
- *字节限定符*
- *包边界*
- *源设备和目标设备信号*
- *时钟和复位*
- *用户信号*

2.1 信号列表

接口信号列在表 2-1 中。关于这些信号的其他信息请参见本章其他章节。

表 2-1 使用以下参数来定义信号宽度：

n 以 byte 为单位表示的数据总线的宽度

I **TID** 宽度，推荐最大为 8-bits

d **TDEST** 宽度，推荐最大为 4-bits

u **TUSER** 宽度，推荐的 bit 数为以 byte 为单位表示的接口宽度的整数倍

信号	源	描述
ACLK	时钟源	全局时钟信号。所有信号都在 **ACLK** 的上升沿采样
ARESETn	复位源	全局复位信号。**ARESETn** 是低有效的
TVALID	主机	**TVALID** 表示主机发出一个有效的传输。当 **TVALID** 和 **TREADY** 都被断言时传输发生
TREADY	从机	**TREADY** 表示从机可以在当前周期接收一个传输
TDATA[(8n-1):0]	主机	**TDATA** 是主要的 payload，用来提供通过接口传输的数据。Data

		payload 的宽度是 byte 的整数倍。
TSTRB[(n-1):0]	主机	**TSTRB** 是字节限定符，表示 **TDATA** 字节对应的内容是否作为一个数据字节或位置字节被处理
TKEEP[(n-1):0]	主机	**TKEEP** 是字节限定符，表示 **TDATA** 字节对应的内容是否作为数据流的一部分来被处理。 如果对应的字节，其 TKEEP 字节限定符被取消断言，则该字节为空字节，可以从数据流中移除
TLAST	主机	**TLAST** 表示一个包的边界
TID[(i-1):0]	主机	**TID** 是数据流的识别码，表示数据的不同流
TDEST[(d-1):0]	主机	**TDEST** 为数据流提供路由信息
TUSER[(u-1):0]	主机	TUSER 是用户定义的边带(sideband)信息，可以与数据流一起被发送

2.2 传输信号

本章节给出了握手信号的细节，并定义了握手信号 **TVALID** 和 **TREADY** 之间的关系。

2.2.1 握手过程

TVALID 和 **TREADY** 的握手决定了信息合适通过接口传输。一个双向的流控制机制允许主机和从机都可以控制通过接口传送的数据和控制信息的传输速率。为了使传输能够发生，**TVALID** 和 **TREADY** 信号都必须被断言。**TVALID** 和 **TREADY** 都可以第一个被断言，或者两个信号在同一个 **ACLK** 周期内同时被断言。

在断言 **TVALID** 之前，主机不能等待 **TREADY** 被断言(也就是要先断言 **TVALID**，再断言 **TREADY**)。一旦 **TVALID** 被断言，其必须保持断言直到握手发生。

从机可以在断言 **TREADY** 之前等待 **TVALID** 被断言。

如果从机断言了 **TREADY**，允许在 **TVALID** 被断言之前从机取消断言 **TREADY**。

接下来章节给出了握手序列的例子。

TVALID 先于 TREADY 的握手

图 2-1 中，主机发出了数据和控制信息并断言了 **TVALID** 信号为高。一旦主机断言了 **TVALID**，主机发出的数据或控制信息必须保持不变，直到从机驱动 **TREADY** 信号为高表示可以接收数据和控制信息。在这种情况下，一旦从机断言 **TREADY** 为高，传输就会发生。箭头标示出了传输发生的位置。

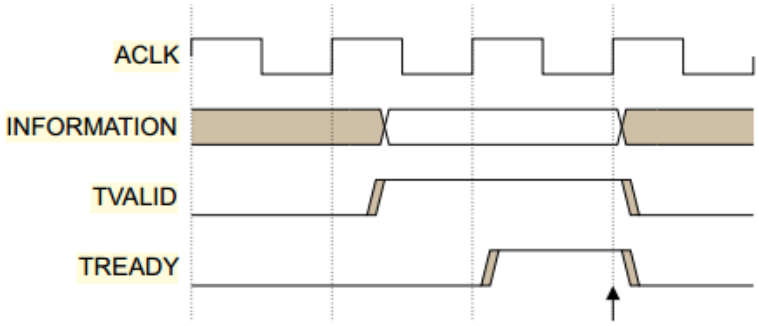


Figure 2-1 TVALID before TREADY handshake

TREADY 先于 TVALID 的握手

图 2-2 中，从机在数据和控制信息有效之前驱动 **TREADY** 为高。这表示目标设备可以在一个 **ACLK** 周期内接收数据和控制信息。在这种情况下，一旦主机断言 **TVALID** 为高，则传输就会发生。箭头标示出了传输发生的位置。

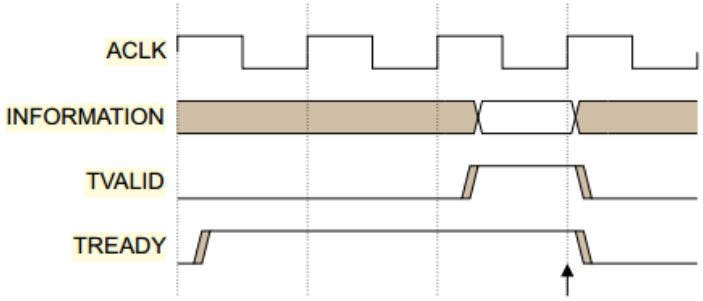


Figure 2-2 TREADY before TVALID handshake

TVALID 和 TREADY 同时发生的握手

图 2-3 中，主机断言 **TVALID** 为高，从机在同一 **ACLK** 周期内也断言 **TREADY** 为高。在这种情况下，如图中箭头标注，传输在同一周期内发生。

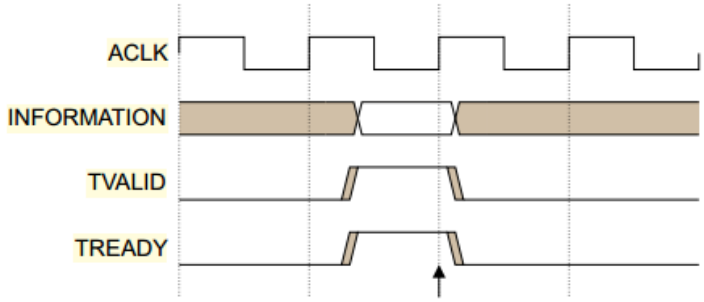


Figure 2-3 TVALID with TREADY handshake

2.3 数据信号

本章节讲述了 AXI4-流接口的数据信号要求。

TDATA 是 AXI4-流接口的主要 payload，用于从一个源设备到一个目标设备传输数据。

2.3.1 字节位置

在一个数据流中，数据总线的低位字节是流中的较早传输的。

对于一个没有空字节的、被充分打包的流，流中一个给定字节的位置可以使用以下方式来决定：

在一个数据流中：

- 字节 n 的序列从 0 开始向上编号
- 序列中的每个传输 t 从 0 开始向上编号
- 数据总线的宽度为 w bytes
- INT(x)是数值 x 向下取整（rounded-down——向绝对值减小的方向舍入取整）

因此字节 n 被包含在传输 t 中：

t = INT(n/w)

其(字节 n)在字节位置 b:

b = n – (t * w)

字节位置 b 被包含在：

TDATA[(8b+7):8b]

2.3.2 字节类型

AXI4-流协议定义了三种字节类型：

数据字节 一个数据字节必须被从源设备传输到目标设备。
据字节的数量、对应的 **TDATA** 数据值，以及流中数据字节的相对位置(相对其他数据字节和位置字节)必须被保持在源设备和目标设备之间。

位置字节 一个位置字节必须被从源设备传输到目标设备。
位置字节的数量和流中位置字节的相对位置(相对其他数据字节和位置字节)必须被保持在源设备和目标设备之间。
与一个字节位置对应的 **TDATA** 数据值不要求在源设备和目标设备之间被传输。

空字节 一个空字节不包含信息。
空字节可以被插入流中，也可以从流中被移除。不要求空字节在源设备和目标设备之间被传输。

一个互联(interconnect)不能修改流中的数据字节或位置字节的数量或相对位置。

允许互联(interconnect)插入或移除流中的空字节。空字节的插入可能会被某些操作请求，比如，传输中的数据和位置字节不足以组成完整的数据宽度而需要扩展宽度到较宽的数据总线宽度。

主机和从机组件不要求支持空字节，因此，任何具有在流中插入空字节能力的互联(interconnect)，在流到达不能处理空字节的目标设备之前，也应该能够移除空字节。

2.3.3 数据合并、组包及宽度转换

理想的是能够适应一个流接口的数据宽度。这将允许构建一个接口宽度固定的组件，之后该组件可以被重用在一个不同宽度接口的系统设计中。适应能力强的接口宽度还可以使具有不同本地宽度的组件被组合在一起并共用一个公共的链路层。

在大部分应用中，期望的接口宽度为 2 的次幂 byte 宽。但是该要求不是协议的要求，所以接口可以是一个 byte 整数倍的宽度。

所有扩展和缩小操作都要求保持任何其对应的 **TKEEP** 数据限定符被断言的字节。如果一个扩展或缩小操作没有足够多的数据字节来组成完整宽度的输出，则附加的字节可以在对应的 **TKEEP** 信号被取消断言时被添加进来。

合并考虑

合并就是将来自两个不同传输的字节混合到一个传输中的过程。

当一个传输中有可被移除的空字节时，可以进行合并操作，将后面的数据或位置字节包含进来。合并可以和组包一起发生，参见*组包考虑*。

合并的规则如下：

- 只有 **TID** 和 **TDEST** 识别码相同的传输才能被合并
- 如果当前传输的 **TLAST** 信号已发出，则该传输不能被合并到接下来的传输中。**TLAST** 还表示没有可被合并的接下来的传输，所以当前传输不能被延迟
- 正确的数据字节和位置字节的顺便必须被保持
- 正确的 **TLAST**、**TSTRB** 及 **TUSER** 必须被保持

允许部分合并。例如，如果当前四字节的传输只有三个数据字节，则允许从下一个传输中合并一个数据字节到当前传输中，即使下一个传输有多于一个的数据字节。

组包考虑

组包是将流中的空字节移除的过程。

组包通常与其他操作如扩展、缩小或合并操作一起发生。

一个使用 **TKEEP** 的数据流可以通过移除空字节来进行组包，以提供一个压缩度更高的数据流。完全组包的数据是不需要的，因此在组包前后，都有可能会出现空字节。

缩小考虑

缩小是从一个给定的数据总线宽度转换到更窄的数据总线宽度。

这个过程通常涉及为单个输入传输产生多输出传输。通常情况下，缩小比例为 2:1，但也可以是其他缩小比例。

缩小的规则如下：

- 输出流中的字节顺序必须与输入流的字节顺序相同
- **TSTRB** 必须以类似的方式缩小到数据的宽度，以保证数据字节和位置字节之间正确的关系被保持
- TLAST 只能与一个缩小操作中的最后一个传输相关联
- 所有输出传输的 **TID** 和 **TDEST** 必须与输入传输的值相同
- **TUSER** 信息必须与相同的字节保持关联

注意：

一个只包含空字节，并且 **TKEEP** 被取消断言，**TLAST** 也没有被断言的传输是被支持的

扩展操作

扩展是从一个给定的数据总线宽度转换到一个更宽的数据总线。

这一过程可以和合并操作混合进行，以便从多输入传输产生一个单个的输出传输。

如果扩展与合并操作混合进行，那么扩展操作可能会是一个更复杂的过程，因为在传输的接收端，扩展器不能始终判断下一个传输是否在一个相同的包中。

扩展的规则如下：

- 输出流中的字节顺序必须与输入流中的字节顺序相同
- **TSTRB** 必须以类似的方式被扩展到数据的宽度，以保证数据字节和位置字节之间正确的关系被保持
- **TLAST** 必须被保持
- 所有输出事物的 **TID** 和 **TDEST** 必须与输入事物的相同
- **TUSER** 信息必须与相同的字节保持关联

注意：

如果没有足够的传输来构成一个完整宽度的扩展流，则需要用 **TKEEP** 信号来表示空字节

2.4 字节限定符

本章节定义了 AXI4-流协议支持的两个字节限定符：

TKEEP	一个字节限定符，用来表示对应字节的内容是否必须被传输到目标设备。
TSTRB	一个字节限定符，用来表示对应字节的内容是一个数据字节还是一个位置字节。

TKEEP 和 **TSTRB** 的每一 bit 都与一个 payload 字节相对应：

- **TKEEP[x]**对应于 **TDATA[(8x + 7):8x]**
- **TSTRB[x]**对应于 **TDATA[(8x + 7):8x]**

2.4.1 TKEEP 限定符

TKEEP 被断言为高，表示对应的字节必须被发送到目标设备。

TKEEP 被断言为低，表示一个可以从流中被移除的空字节。

如果一个传输的所有 **TKEEP** 位都被断言为低，该传输也是合法的。

对于一个所有 **TKEEP** 位都被断言为低的传输，允许其传输被阻止，除非其 **TLAST** 被断言为高。参见*包边界*。

不推荐主机和从机处理空字节，因此，任何具有在流中插入空字节能力的互联(interconnect)，在流到达不能处理空字节的目标设备之前，也应该能够移除空字节。

2.4.2 TSTRB 限定符

当 **TKEEP** 被断言为高时，**TSTRB** 被用来表示对应的字节是否是一个数据字节或一个位置字节。**TSTRB** 被断言为高，表示对应的字节包含有效信息，是一个数据字节。**TSTRB** 被断言为低，表示对应的字节比包含有效信息，是一个位置字节。

一个位置字节用来表示流中正确的数据字节的相对位置。位置字节通常用于当数据流在目标设备上执行一个部分信息更新操作时。

如果数据对应的位置字节是无效的，则互联(interconnect)不需要传输 **TSTRB** 为低的 **TDATA** 对应的字节。

2.4.3 TKEEP 和 TSTRB 的组合

表 2-2 列出了 **TKEEP** 和 **TSTRB** 字节限定符的有效组合。

2-2 TKEEP 和 TSTRB 字节限定符

TKEEP	TSTRB	数据类型	描述
高	高	数据字节	对应的字节包含有效信息，必须在源设备和目标设备之间被传输。
高	低	位置字节	对应的字节表示流中数据字节的相对位置，但不包含任何实质性的数据值。
低	低	空字节	对应的字节不包含信息，可以从流中被移除
低	高	保留	不能被使用

并不是所有的组件都要求 **TKEEP** 和 **TSTRB** 数据限定符。参见*可选的 TKEEP 和 TSTRB*。

2.5 包边界

包是通过接口传输的一组字节。通过处理组合成包的传输，可以使得基础组件更有效率。AXI4-流包类似于 AXI4 突发。

在一个包传输过程中需要考虑的信号有 **TID**、**TDEST** 以及 **TLAST**。关于 **TID** 和 **TDEST** 的更多信息请参*考源设备和目标设备信号*。

TLAST 的使用如下：

- 当取消断言时，**TLAST** 表示接下来可以进行另一个传输，因此当前传输可以被延迟，以进行扩展、缩小或合并操作
- 当被断言时，**TLAST** 可以被一个目标设备用来表示包的边界
- 当被断言时，**TLAST** 表示在一个共享链路上进行仲裁改变的有效点

注意：

不要求仲裁只发生在一个 **TLAST** 边界，但是 **TLAST** 可用来通过将传输保持在相同的包中来潜在提高效率。

TLAST 可用于在源设备和目标设备之间传送信息。包的数量，以及 **TLAST** 断言的数量必须在主机和从机之间被保持。

协议中没有明确给出一个包边界的开始信号。一个包的开始由以下决定：

- Reset 之后发生的第一个 **TID** 和 **TDEST** 信号对
- 前一个传输结束之后的第一个传输，该传输具有唯一的一组 **TID** 和 **TDEST** 值

一个包中的所有字节都是来自相同的源设备，被发送到相同的目标设备，并且具有相同的 **TID** 和 **TDEST** 值。

合并属于不同包的传输是不允许的。这个要求表示，两个具有相同 **TID** 和 **TDEST** 值的传输，如果前一个传输断言了 **TLAST**，那么这两个传输也不能被合并。参见*合并考虑*。

绝对不允许合并两个具有不同 **TID** 和 **TDEST** 值的传输。

2.5.1 零数据或位置字节的传输

一个传输，其 **TLAST** 可以被断言但却不包含数据或位置字节。这可以用于：

- 当没有更多的数据或位置字节要传输时，表示一个包的结束
- 推进传输任何被保留在中间缓存区的数据
- 在终点完成一个操作，该终点即一个包的结尾、期待一个 **TLAST** 的位置

一个已经断言了 **TLAST**，但没有任何数据或位置字节的传输，可以和前一个具有相同 **TID** 和 **TDEST** 值、但却没有断言 **TLAST** 的传输合并。参见合并考虑。

注意：

由于不支持重新排序，因此，发送一个零数据字节的传输可以有效推进主机和从机之间的所有传输。参见*传输顺序*。

2.6 源设备和目标设备信号

与源设备和目标设备关联的信号有：

TID 提供了一个流标识符，可被用来区分通过相同接口传输的多流数据。

TDEST 提供了用于数据流的粗放的路由信息。

具有相同 **TID** 和 **TDEST** 值的传输都来自相同的流。不允许合并属于不同流的传输。

在每次传输的基础上，允许对来自不同流的传输进行交错，并且这种交错并不限于 **TLAST** 边界。参见*传输交错*。

互联组件可以控制 **TID** 和 **TDEST** 信号：

- 互联可以产生附加的 **TID** 信号。这允许两个其他方面相同的流被区分。
- 互联可以产生或控制 **TDEST** 信号来为一个流提供路由信息

任何 **TID** 或 **TDEST** 的控制必须确保两个不同的流保持唯一性。

注意：

一个共同的使用模型用于一个互联，用来根据输入流提供的 **TID** 信息，为输出流产生 **TDEST** 信息。

2.7 时钟和复位

本章节讲述了时钟和复位信号的要求。

2.7.1 时钟

每个组件都使用一个单一的时钟信号——**ACLK**。所有输入信号都在 **ACLK** 的上升沿上采样。所有输出信号必须在 **ACLK** 上升沿之后发生改变。

2.7.2 复位

协议包含一个单一的低有效的复位信号——**ARESETn**。复位信号可以被异步断言，但取消断言必须是在 **ACLK** 上

升沿之后、同步的。

在复位期间，**TVALID** 必须被驱动为低。

所有其他信号可以被驱动为任何值。

一个主机接口必须只能在一个 ARESETn 被断言为高的上升沿之后、在 ACLK 的一个上升沿来开始驱动 TVALID。图 2-4 展示了 reset 之后 TVALID 可以被驱动为高的第一个点。

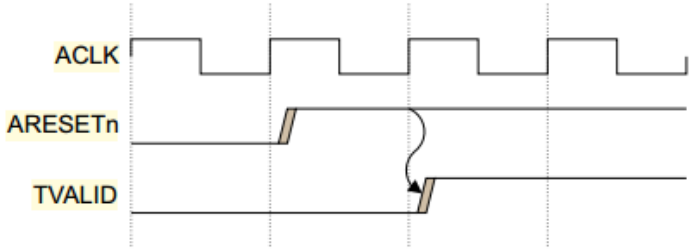


Figure 2-4 Exit from reset

2.8 用户信号

典型的一个流接口的使用要求某些用户边带信号。边带信号可以被用于数据字节、传输、包、或者基于帧的信息。

有几种用户信号的使用，例如：

- 标记位置或者特殊数据项类型
- 提供辅助信息，这种辅助信息必须伴随数据。比如，奇偶性，控制信号，及 flags
- 识别一个包的段

为了确保一个一致的方法来传输用户信息，协议规定用户信号在一个字节基础上传输。

推荐，但不要求 **TUSER** 位的数量是一个以字节表示的接口宽度的整数倍。每个字节的用户信号必须在 **TUSER** 内以相邻的方式被组合在一起。

用户 bits 的位置定义为：

- 每个数据 byte 有 m 个用户信号与之关联
- 接口的总宽度为 n 字节
- 用户 bits 的总数量为 u，这里 u=m*n

字节 x 的用户信号(这里 x=0 …(n-1))位于：

TUSER[((x * m) + (m – 1)) : (x * m)]

当关联的 **TKEEP** 信号被取消断言为低时，**TUSER** 位的传输不会被请求，或者不能被保证。

注意：

与一个空字节关联的用户位(通过关联的 **TKEEP** 位表示)，如果空字节从流中被移除，则该用户位也必须从数据流中被移除。

如果一个空字节被插入到数据流中，则适当数量的用户位也必须被插入。当插入附加位时，他们也必须被固定为低（谁被固定为低？？）。

2.8.1 基于传输信息的用户信号

TUSER 可被用来传送有关整个传输的信息，而不是对个别字节的信息。关于此的一个例子是，如果相同的信息适用于传输中的每个字节，此时，该信息一旦只针对整个传输而不是将该信息复制到传输中的每个字节，这种情况下将更有效率。

TUSER 可被用来传送基于传输的信息，但传输机制将会把被传送的数据字节之间的 **TUSER** 信息分离。基于传输

的 **TUSER** 信息的可靠传送只能通过以下约束来保证：

- 输入到互联(interconnect)的数据总线宽度必须和互联的输出数据总线宽度一致
- 在互联(interconnect)中发生的任何数据宽度的转换都不能修改互联输入和互联输出之间数据的组包

2.8.2 用户信号宽度的匹配

一个穿过复杂互联的传输，可能会跨越互联中不同的部分，该部分支持的 **TUSER** 宽度和主机或从机所支持的 **TUSER** 宽度都不相同。本章节讲述了用来保证穿越这样一个互联的可靠传输所需要的 **TUSER** 操作。

Padding(填充)和 trimming(裁剪)信息

本章节描述了在一个每字节的 **TUSER** 数不匹配的接口上，**TUSER** 信息是如何被填充或裁剪的。本章节中所有的例子都要求数据宽度已经做过转换，以保证接口两边数据字节的数目是匹配的。

TUSER 的填充和裁剪，是通过添加或移除每字节的高位，而不是整个 **TUSER** 数据的高位来完成的。当填充时，任何添加的 bits 必须被固定为低(即 0).

图 2-5 展示了对于一个 8 字节数据接口，从每字节 1bit 填充到每字节 2 bits。

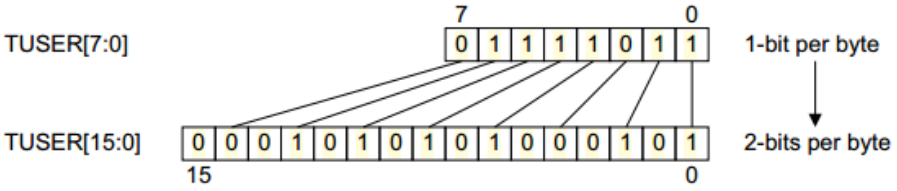


Figure 2-5 Example 1-bit to 2-bits padding for an eight byte data interface

图 2-6 展示了对一个 8 字节数据接口，从每字节 2 bits 裁剪到每字节 1 bit。

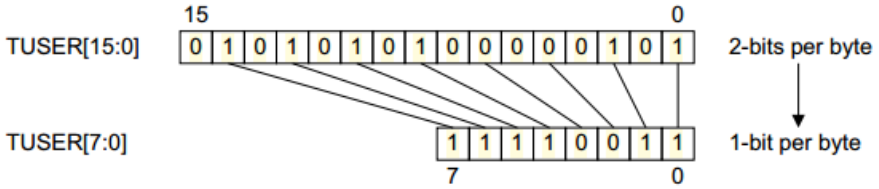


Figure 2-6 Example 2-bits to 1-bit trimming for an eight byte data interface

图 2-7 展示了对于一个 4 字节数据接口，从每字节 2 bits 填充到每字节 4 bits。

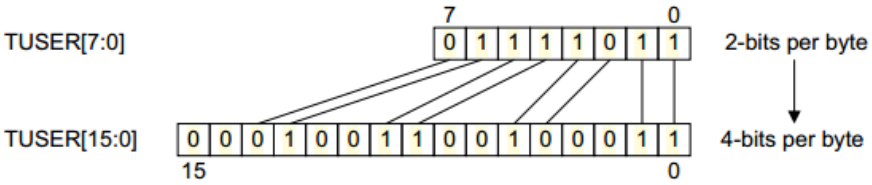


Figure 2-7 Example 2-bits to 4-bits padding for a four byte data interface

图 2-8 展示了对于一个 4 字节数据接口，从每字节 2 bits 填充到每字节 3 bits。

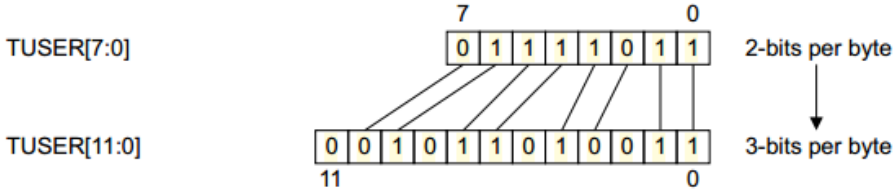


Figure 2-8 Example 2-bits to 3-bits padding for a four byte data interface

决定 TUSER 的宽度

对于一个具有若干个主机接口和从机接口的互联，互联必须支持的每字节 **TUSER** bit 数定义为：

MIN(MAX[主机每字节 **TUSER** 位数]，MAX[从机每字节 **TUSER** 位数])

在实际中：

1. 找出 **TDATA** 的每字节 **TUSER** bit 数最大的主机
2. 找出 **TDATA** 的每字节 **TUSER** bit 数最大的从机
3. 使用以上两个值中的最小值，即为互联必须支持的每 **TDATA** 字节 **TUSER** bit 数

如果是主机具有较小的最大值，则与可被任何主机产生的相比，一个或多个从机具有更大的每 **TDATA** 字节 **TUSER** 数。

如果是从机具有较小的最大值，则与可被任何从机接受的相比，一个或多个主机具有更大的每 **TDATA** 字节 **TUSER** 数。

TUSER 的适用准则有：

- **TUSER** 窄于互联的主机，必须将 **TUSER** 补 0 来匹配互联端口。
- **TUSER** 宽于互联的主机，必须裁剪 **TUSER** 来匹配互联端口。该准则必须只能适用于 TUSER 比最宽的从机还要宽的主机。
- 在 **TUSER** 窄于下行 **TUSER** 的互联中的任何点，**TUSER** 必须被补 0。
- 在 **TUSER** 宽于下行 **TUSER** 的互联中的任何点，**TUSER** 必须被裁剪。该准则必须只能适用于所有可能的下行从机具有一个相等的或者更窄的 **TUSER**。
- 在到达的一个从机上，如果从机的 **TUSER** 比互联提供的窄，则 **TUSER** 必须被裁剪。
- 在到达的一个从机上，如果从机的 **TUSER** 比互联提供的宽，则 **TUSER** 必须被补 0。该准则必须只能适用于 **TUSER** 比最宽的主机还要宽的从机。

如果在构造互联的过程中，知道哪个主机与哪个从机通信的，则允许对必须支持的 TUSER 宽度进行进一步的优化。

3 默认信号要求

本章描述了 AXI4-流接口的接口信号要求。包含以下章节：

- 默认值信号
- 兼容性考虑

3.1 默认值信号

本章描述了流接口的默认信号值。

3.1.1 可选的 TREADY

TREADY 信号在确定的环境中可以被省略。但是推荐始终实现 **TREADY**。

TREADY 的默认值是高。

从机接口考虑

对于一个从机接口，如果从机能够始终接收一个传输，则 **TREADY** 输出可以被省略。

注意：在省略 **TREADY** 信号之前，推荐所有使用模型都被考虑进来。例如，当时钟大于一个特定频率时，从机可能能够始终接收传输，但是当时钟低于一个特定阈值时，从机可能需要使用 **TREADY**。

主机接口考虑

对于一个主机接口，省略 **TREADY** 输入表示该主机接口不能接受背压(back-pressure)，并且该主机假定 **TREADY** 始终为高。

注意：即使主机不能接受背压，也推荐该主机接口包含 **TREADY** 信号。一个不能接受背压的主机接口，可以通过在传输期间将 **TREADY** 驱动为低，来用该 **TREADY** 标示一个错误条件。通过将 **TREADY** 信号包含到接口，可以使错误源被正确识别。

3.1.2 可选的 TKEEP 和 TSTRB

TKEEP 和 **TSTRB** 信号是可选的信号，并不是所有数据流类型都需要这两个信号。

注意：如果一个数据流是可扩展的(upsizing)，并且不能保证始终有足够的传输来构造一个完整的扩展流宽度，则 **TKEEP** 必须用来标示空字节。

默认值规则

默认值规则有：

- 当 **TKEEP** 省略时，**TKEEP** 默认对所有 bits 都是高
- 当 **TSTRB** 省略时，**TSTRB=TKEEP**
- 当 **TSTRB** 和 **TKEEP** 都省略时，**TSTRB** 和 **TKEEP** 默认对所有 bits 都是高

3.1.3 可选的 TLAST

对于不具有包和帧概念的数据流，**TLAST** 的默认值是未知的。下列选项可供选择：

- 设置 **TLAST** 为低。这表示所有传输都在同一个包中。该选项为合并和扩展提供了最大的可能，但这意味着传输有可能在流中被间歇突发延迟。一个恒定为低的 **TLAST** 信号可能也会影响在共享通道中的流的交错，因为互联可以使用 **TLAST** 信号来影响仲裁过程。
- 设置 **TLAST** 为高。这表示所有传输都是单个的包。该选项可以保证传输不会在基础组件中延迟。该选项也可以保证流不会通过影响仲裁过程而不必要地阻止一个共享通道的使用。该选项会阻止流合并，这些流来自于具有该设置的主机。该选项也会阻止有效的扩展。
- 自动产生一个脉冲的 **TLAST** 值。该选项在一个固定数量的传输之后断言 **TLAST**，例如，经过 2 个或 16 个传输。该选项可能证明是一个很好的妥协，允许有效地操作而不会过度地阻塞共享的通道。

推荐的方法是：

- 任何具有包边界概念的组件都必须包含一个 **TLAST** 信号。当被包含到一个组件的接口中时，**TLAST** 信号通过互联必须被保留。
- 当一个组件不支持 **TLAST** 信号，并且互联中的拓扑和功能是未知的，则 **TLAST** 信号必须默认为高。这将保证传输不会在互联中被使用 **TLAST** 信号来强制一个缓存 draining 操作的组件不确定地延迟。
- 当一个组件不支持 **TLAST** 信号，并且由于互联的构建——没有互联组件要求使用 **TLAST** 信号来阻止互联中传输的延迟——而可以被保证。这种情况下 **TLAST** 信号可以被固定为低。

3.1.4 可选的 TID，TDEST，及 TUSER

TID，**TDEST**，以及 **TUSER** 都是接口中可选的信号：

- 一个主机不要求支持这三个输出信号
- 一个具有多余 **TID**，**TDEST**，及 **TUSER** 输入信号的从机必须将这 3 个信号的所有 bits 固定为低。

3.1.5 可选的 TDATA

大部分 AXI4-流接口的应用都将传输一个数据 payload。但是，也允许开发一个比包含 **TDATA** 数据 payload 的接口。

如果没有 **TDATA**，则也不能有 **TSTRB**。

在 **TDATA** 被忽略的接口中，如果具有 **TKEEP**，则 **TKEEP** 的位宽被用来决定所有扩展和缩小操作的正确操作。

在 **TDATA** 和 **TKEEP** 都被忽略的接口中，所有扩大和缩小操作必须以相同的方式进行操作。

3.2 兼容性考虑

本章讨论 AXI4-流组件接口的兼容性考虑。

注意：接口兼容性不会为两个组件共同发挥作用提供保证，因为更高级别的考虑，比如共享数据结构的格式，同样需要被考虑进来。

因为 AXI4-流具有若干个可选的功能，它可能为一个主机和一个从机组件实现不同的功能级。

通过支持所有输入信号可以保证对任何单个组件的完全兼容。输出信号可以被有选择地支持，使用如*默认值信号*中描述的默认值可以确保兼容操作。

关于兼容性有两方面考虑：

- 直连兼容性。该考虑主要关于一个主机直接连接一个从机。
- 互联兼容性。该考虑关于一个互联的实现具有兼容两个组件的效果。

3.2.1 主机兼容性

一个主机和一个从机接口如果要相兼容，则接口的数据宽度必须相同。如果不是这种情况，则要求互联组件提供数据宽度转换来匹配数据宽度。

AXI4-流接口的单向性意味着，任何支持 **TREADY** 的主机组件，其接口可以和任何支持完整功能的从机组件相兼容。这是因为任何不是由主机组件提供的输出信号，都可以被给定一个默认值。参见*默认值信号*。

3.2.2 从机兼容性

兼容性问题主要源自于一个从机组件支持输出信号的能力，这些输出信号是连接到该从机的任何主机所产生的。

数据宽度 兼容性的第一个要求是，两个接口的数据宽度必须相同。如果不是这种情况，则要求互联组件提供数据宽度转换来匹配数据宽度。

源和目的信号 如果要求一个从机组件能够区分多个不同的流，则该从机必须支持足够的源和目标信令，这些源和目标信令分别使用 **TID** 和 **TDEST** 输入。通常来说，一个从机组件有能力处理任何级别的流交错。如果一个从机组件被要求能够区分多个流，则该从机组件必须包含合适的 **TID** 和 **TDEST** 输入。在这种情况下，从机将具有一个交错上限，这个上限必须不能被超出。参见*传输交错*。

空字节和位置字节

不要求从机支持空字节和位置字节。以下是为不支持两种字节的从机推荐的方法：

- 如果一个从机不支持位置字节，则所有字节必须被转换为数据字节。该方法不允许部分更新一个数据结构，这可能会导致被覆盖的数据字节损坏。但是，这会确保所有数据字节在流中保持正确地被放置。

- 如果一个从机不支持空字节，则一个执行组包的组件会被用来从流中移除空字节

注意：

一个执行组包的组件通常可能需要支持多个并发流。

3.2.3 互联兼容性

一个互联组件被用来从主机到从机传送数据流。要求互联要能够可靠地从主机到从机传输所有数据字节和位置字节。随意的扩大和缩小操作可能会引入空字节。

有两种技术可以用来确保空字节不会被传送到不支持空字节的从机：

- 互联拓扑结构可以被限制，以确保空字节只有在数量等于目标从机的数据宽度时才会被引入。这保证整个传输可以被抑制。
- 一个执行组包的组件可以被用来移除流中的空字节。

4 传输交错和排序

本章讲述 AXI4-流协议允许的传输交错和排序过程。包含以下章节：

- *传输交错*
- *传输排序*

4.1 传输交错

传输交错是交错来自不同流的传输的过程。

互联不需要限制流的交错，以便从机的能力不会被超过。

注意：

在某些互联拓扑中，传输交错可能被限制在包边界，以便通过合并传输提高效率。

4.1.1 从机考虑

从机可被设计为不限制其处理交错流的数量。组件上如果出现交错流，可能其操作效率会比较低，但是从功能上讲，从机将会正确操作。

当从机处理一个完整的包时，其操作会更有效率，在这样一种系统设计中，仲裁可被设计操作在 **TLAST** 边界上。

注意：

限制仲裁机制使其只能操作在 **TLAST** 边界上将会不能与 **TLAST** 永久固定为低的主机相兼容。推荐任何使用 **TLAST** 信号来影响选择过程的仲裁也应该包含一个覆盖(override)机制。

从机也可以被设计为限制交错。如果是这种情况，则必须使用以下其中一种技术来确保从机能力不会被超过：

- 只能通过一个主机来访问从机，并且该主机可以使用所有的交错功能。
- 从机可以被多个主机访问，每个主机不能交错包。系统设计，或着某些更高级的控制机制要能够保证一次访问从机的主机数量不操作从机的交错能力。
- 从机被多个主机访问，并且一个更高级的机制确保从机所有的交错能力不会被超过。

4.2 传输排序

AXI4-流协议要求所有传输保持顺序。不允许传输的重新排序。不能重新排序的有点有：

- 确保重新排序不会增加被从机观察到的流交错
- 系统整体可预测性被提高
- 通过使用传输的 **TID**，观察发自同一个主机、已经到达相同从机的后一个传输，可以判定一个给定的传输已经到达了目标设备
- 减少系统的复杂性

附录 A 与 AXI4 写数据通道的比较

本附录列出了 AXI4-流接口和 AXI4 写数据通道的注意区别。包含以下章节：

- *与 AXI4 写数据通道的区别*

A.1 与 AXI4 写数据通道的区别

AXI4-流接口与 AXI4 写数据通道有很多相似性。但是有一些重要的区别。这些区别总结如下：

- AXI4 写数据通道不允许交错
- AXI4-流接口不具有一个定义的或者最大的突发或包长度
- AXI4-流接口允许数据宽度为数据字节的任何整数
- AXI4-流接口包含 **TID** 和 **TDEST** 信号来分别表示源和目的
- AXI4-流接口定义了 **TUSER** 边带信号更精确的操作
- AXI4-流接口包含 **TKEEP** 信号以允许插入或移除空字节

写在最后：

本文档是本人花了一个星期翻译的，最初是为了学些 AMBA4 协议，了解协议新增的内容。翻译过程中尽量忠于原文，但水平有限，难免有些内容理解不到位的，或者是英文文意理解错的地方，如果有发现错误的，欢迎指出，共同学习。如果想自己更改，可以向本人索要本文的 WORD 源文档进行修改。本人邮箱：li.wsh@163.com。

欢迎各位读者一起交流学习。

风雨者

2016.04.15 于西安