

# AMBA 3 AHB-Lite协议

**V1.0**

中文版



版权© 2001, 2006 ARM 有限公司 版权所有

ARM IHI 0033A

chnlw111 翻译

chnlw111@sina.com

# 目录

目录.....	ii
前言.....	iv
关于本书.....	v
反馈.....	vii
第一章 简介.....	1-1
1.1关于本协议.....	1-2
1.1.1 主设备.....	1-2
1.1.2 从设备.....	1-3
1.1.3 译码器.....	1-3
1.1.4 多路器.....	1-4
1.2 操作.....	1-4
1.3 多层AHB-Lite.....	1-5
第二章 信号描述.....	2-1
2.1 全局信号.....	2-2
2.2 Master信号.....	2-2
2.3 Slave信号.....	2-4
2.4 译码器信号.....	2-5
2.5 多路器信号.....	2-6
第三章 传输.....	3-1
3.1 基本传输.....	3-2
3.2 传输类型.....	3-4
3.3 锁定传输.....	3-6
3.4 传输大小.....	3-7
3.5 Burst操作.....	3-8
3.5.1 BUSY传输后Burst终止.....	3-8
3.5.2 Burst 提前终止.....	3-9
3.5.3 Burst举例.....	3-9
3.6 等待传输.....	3-13
3.6.1更改传输类型.....	3-13
3.6.2 在等待状态期间地址改变.....	3-15
3.7 保护控制.....	3-17
第四章 总线互连.....	4-1
4.1 地址译码.....	4-1
4.1.1默认Slave.....	4-1
4.2 总线互连.....	4-2
第五章 Slave响应信号.....	5-1
5.1 Slave传输响应.....	5-2
5.1.1传输完成.....	5-2
5.1.2 传输挂起.....	5-2
5.1.3 错误响应.....	5-3
第六章 数据总线.....	6-1

---

6.1数据总线 .....	6-2
6.1.1 写总线 .....	6-2
6.1.2 读总线 .....	6-2
6.1.3 大小端 .....	6-3
6.2 数据总线位宽 .....	6-4
6.2.1 在宽总线上实现窄位宽的Slave.....	6-4
6.2.2 在窄总线上实现宽Slave.....	6-4
6.2.3 在宽总线实现Master .....	6-5
第七章 时钟和复位 .....	7-1
7.1 时钟和复位需求 .....	7-2
7.1.1 时钟 .....	7-2
7.1.2 复位 .....	7-2
术语 .....	1

# 前言

前言介绍了Advanced microcontroller Bus Architecture(AMBA)3 AHB-Lite Protocol Specification.包含以下两节:

关于本书

反馈

## 关于本书

本书是AMBA3 AHB-Lite协议的规格说明书

### 预期读者

本书的写作目的是帮助软硬件工程师设计遵从AHB-Lite协议的系统和模块。

### 本书使用

本书由以下章节构成：

#### 第一章 介绍

阅读此章了解AHB-Lite协议梗概。

#### 第二章 信号描述

阅读此章了解信号描述。

#### 第三章 传输

阅读此章了解由遵从AHB-Lite协议的主设备发起的不同种类的传输的信息。

#### 第四章 总线互联

阅读此章了解AHB-Lite协议要求的互联逻辑。

#### 第五章 从设备响应信号

阅读此章了解从设备响应信号的信息。

#### 第六章 数据总线

阅读此章了解读写数据总线的信息，了解怎样对接不同位宽的数据总线。

#### 第七章 时钟和复位

阅读此章了解时钟和复位信号信息

#### 术语

阅读此章了解本文的术语短语的定义

### 约定

本节描述了本文时钟的约定信息：

字体信息

时序图解

信号

### 字体

字体约定如下：

*斜体* 重要的注解，引入特殊的术语，表示文档内部的引用，交叉引用。

**黑体** 重要的交互元素，例如目录名字。表示信号名字。也用来for terms in descriptive lists, when appropriate.

`monospace` 表示你可以在键盘输入的文本，例如命令，文件和程序名，和源代码。

`monospace` 表示命令或选项的缩写。你可以用下划线的文本而不是命令或选项的全名。

*`monospace italic`* 表示monospace的参数，参数可以被具体的特定数值取代。

**`monospace bold`** 当用到举例说明的源代码时的源码语言的关键字。

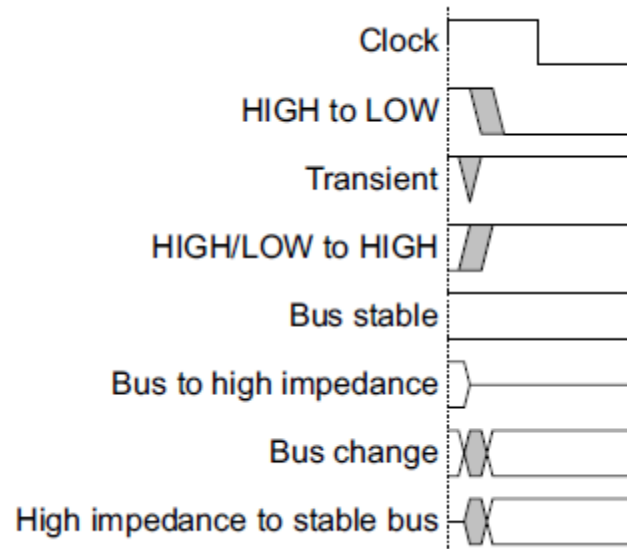
<and> 尖括号括起来表示当其出现在源代码或代码段中，在语法上可以替代的短语。它们通常用在可执行文件中，例如：

MRC p15, 0 <Rd> <CRn> <CRm> <Opcode\_2>

Opcode\_2的值是可以通过寄存器访问的变量。

时序图解

在下图命名为*Key to timing diagram*的时序图解释了时序图中用到的约定。时序变化已经在此图清楚标记。用户禁止假设任何没有在时序图中明确标示出来的时序。阴影部分的总线和（单bit）信号区域是未定义的时序，因此阴影部分可以假定任何值，它是高低电平并不重要别且不会影响总线操作。



Key to timing diagram convention

注：如图Key to timing diagram convention所示，单bit信号有时同时画出了HIGH高电平和LOW低电平，看起来有点像总线信号。如果一个单bit信号像这样表示，则表示当前此信号的值并不影响附随的描述。

### 信号

信号约定如下：

Lower-case n 小写字母n 表示低有效的信号

**Prefix H** 前缀H的信号 表示Advanced High-performance Bus(AHB) 信号

**Prefix P** 前缀P的信号 表示Advanced Peripheral Bus(APB)信号

**Signal Level** 信号电平 被asserted的信号的电平取决于信号是高有效还是低有效。

Asserted 意思是对高有效的信号赋值为高，对低有效的信号赋值为低。

### 扩展阅读

本节列出了ARM公司和第三方公司出版的相关资料

ARM公司定期更新和修正文档。登录<http://www.arm.com>来查询勘误，遗漏和被问到的问题列表

ARM 出版

对于相关信息，请查询以下文档：

AMBA 3 APB Protocol Specification

AMBA AXI Protocol Specification

Muliti-layer AHB Overview

## 反馈

ARM公司欢迎AHB-Lite协议和本文档的反馈.

### 关于协议的反馈

对于AHB-Lite 协议，如果您有任何意见和建议请联系ARM公司.

### 关于原文的反馈

关于原英文文档，如果您有任何意见，请发送电子邮件 [errata@arm.com](mailto:errata@arm.com) 请给出：

标题；

编号；

您的意见涉及的页数；

您的意见的简明描述。

ARM公司也欢迎普通建议以帮助提高和改进。

### 关于本文的反馈

关于本翻译文档，如果您有任何意见和建议，请发送电子邮件 [chnlw111@sina.com](mailto:chnlw111@sina.com) 请给出：标题；编号；您的意见涉及的页数；您的意见的简明描述。多谢。

让我们一起学习AMBA总线协议和相关内容。

# 第一章 简介

本章提供了AHB-Lite协议的概述. 包括以下几节:

*1.1 关于本协议在1-2页*

*1.2 操作在1-4页*

*1.3 多层AHB-Lite在1-5页*

注: 为了举例说明, 在本文用32位总线. 增加数据总线位宽也是允许的, 在6.2 数据总线位宽在6-4页有阐述.



## 1.1关于本协议

AMBA AHB-Lite是面向高性能的可综合设计，为支持一个总线主设备提供总线接口并提供高带宽操作。

注：想要了解如何实现基于AHB-Lite总线接口的多主设备的系统，参考1.3 多层AHB-Lite在1-5页。

AHB-Lite实现了高性能的特性，高时钟频率系统包括：

- Burst传输
- 单时钟沿操作
- 非三态实现
- 宽数据位，包括64、128、256、512和1024位

最常见的 AHB-Lite 从设备包括内存器件、外存接口和高带宽外设。尽管低带宽外设可以连接到AHB-Lite从设备，为了系统性能考虑，人们通常将其连接到AMBA *Advanced Peripheral Bus*(APB)。在高级别总线和APB之间实现桥接功能的是一种特殊的AHB-Lite从设备，即APB桥。

图1-1展示了一个具有一个主设备的AHB-Lite系统，包括一个主设备和三个从设备。总线互联逻辑包括一个地址译码器和一个从设备到主设备的多路选择器。译码器监控从主设备输出的地址线，来传输给从设备选择信号；多路器将相应的从设备的输出数据反馈给主设备。

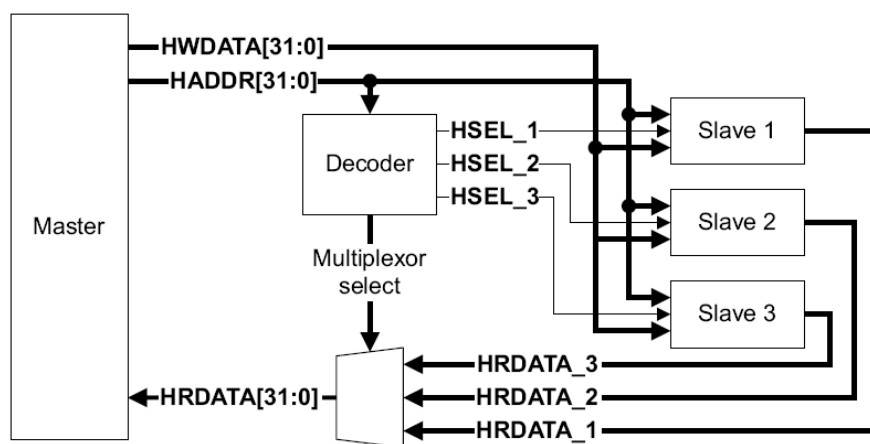


图1-1 AHB-Lite 方框图

注：在图1-1没有显示出主设备的控制信号。在此澄清。

AHB-Lite系统的主要组件将在下文描述：

- 1.1.1 主设备
- 1.1.2 从设备
- 1.1.3 译码器
- 1.1.4 多路器

### 1.1.1 主设备

一个AHB-Lite主设备提供了地址线和控制信息来发起读操作或写操作。图1-2显示AHB-Lite主设备接口。

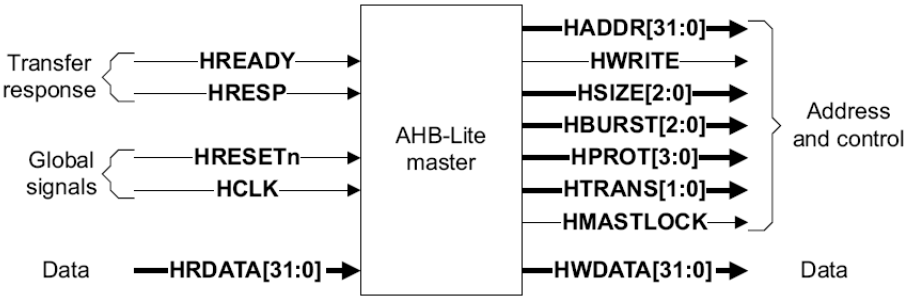


图1-2 Master 接口

1.1.2 从设备

在总线系统中，AHB-Lite从设备响应由主设备发起的传输。从设备使用从解码器输出的HSELx信号来控制何时响应总线传输。从设备信号反馈给主设备的信息包括：

- 成功
- 失败
- 等待数据传输

在图1-3显示了一个AHB-Lite从设备接口。

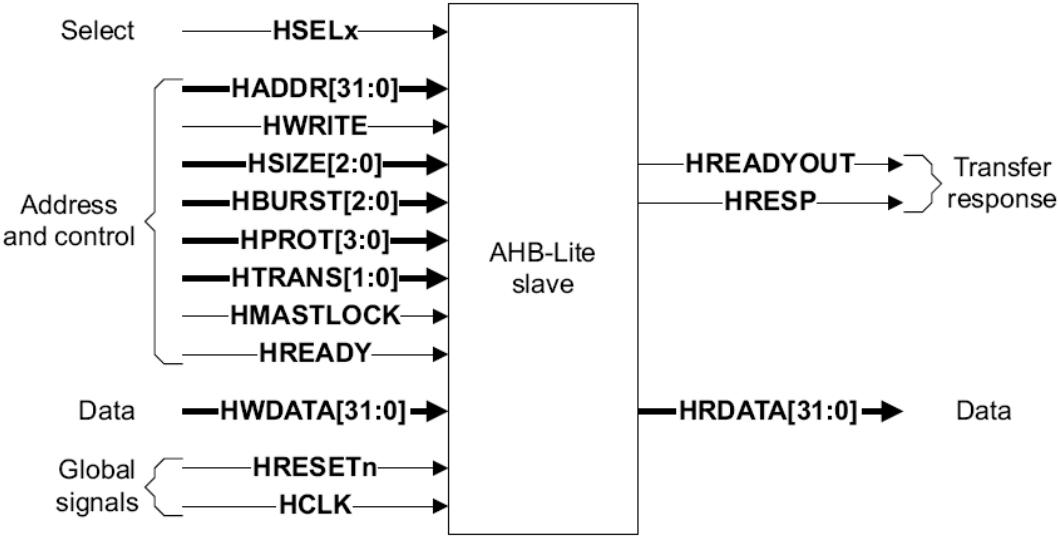


图1-3 Slave 接口

1.1.3 译码器

该组件对每个传输的地址进行译码，给当前传输相关的从设备提供选择信号，该组件也给多路器提供了控制信号。

当AHB-Lite系统包含两个或更多的从设备时，一个单个的集中控制的译码器是必需的。在4.1 地址译码这一节可以获得更多信息。

注：一个多层AHB-Lite的实现，译码功能通常包含在多层互联组件中。

### 1.1.4 多路器

Slave-to-Master多路转换器是必需的, 为了实现从设备发送到主设备的读数据总线和响应信号的多路选择. 译码器给多路器提供控制信号.

当AHB-Lite系统包含两个或更多的从设备时, 一个单个的集中控制的多路器是必需的. 在4.2 总线互连这一节可以获得更多信息.

注: 一个多层AHB-Lite的实现, 多路转换功能通常包含在多层互联组件中.

## 1.2 操作

主设备通过驱动地址和控制信号来发起一个传输. 这些信号提供了传输地址、传输方向、传输位宽的信息, 并且指示是否是Burst传输的一部分. 传输可以是:

- 单一传输
- 递增Burst传输, 在地址边界不回卷
- 回卷Burst传输, 在特殊的地址边界回卷

写数据总线从主设备向从设备搬数据, 读数据总线从Slave向Master搬数.

每一个传输有如下组成:

**Address Phase** 地址相位 一个地址和控制周期

**Data Phase** 数据相位 一个或多个数据周期

Slave无法请求扩展地址相位, 因此所有的Slave必须具有在地址相位采集完地址信息的能力. 但是, Slave可以使用**HREADY**信号请求Master扩展数据相位. **HREADY**信号, 为低时, 将会在传输过程中插入等待状态, 从而可以使Slave有足够的时间提供数据或采集数据.

Slave通过**HRESP**信号指示传输成功或者失败.

### 1.3 多层AHB-Lite

因为AHB-Lite是一个单Master总线接口，如果需要多个Master的系统，就需要一个部件来将各个Master互相隔离。为了实现这种隔离功能，每个Master可以考虑为处在它自己的层，这样隔离组件必须建立多层互连，使的多有的Master之间互相隔离但是可以共享Slave。在多层AHB-Lite的互连组件中，Slave仲裁是必须要实现的。

图1-4是一个多层AHB-Lite系统的例子：

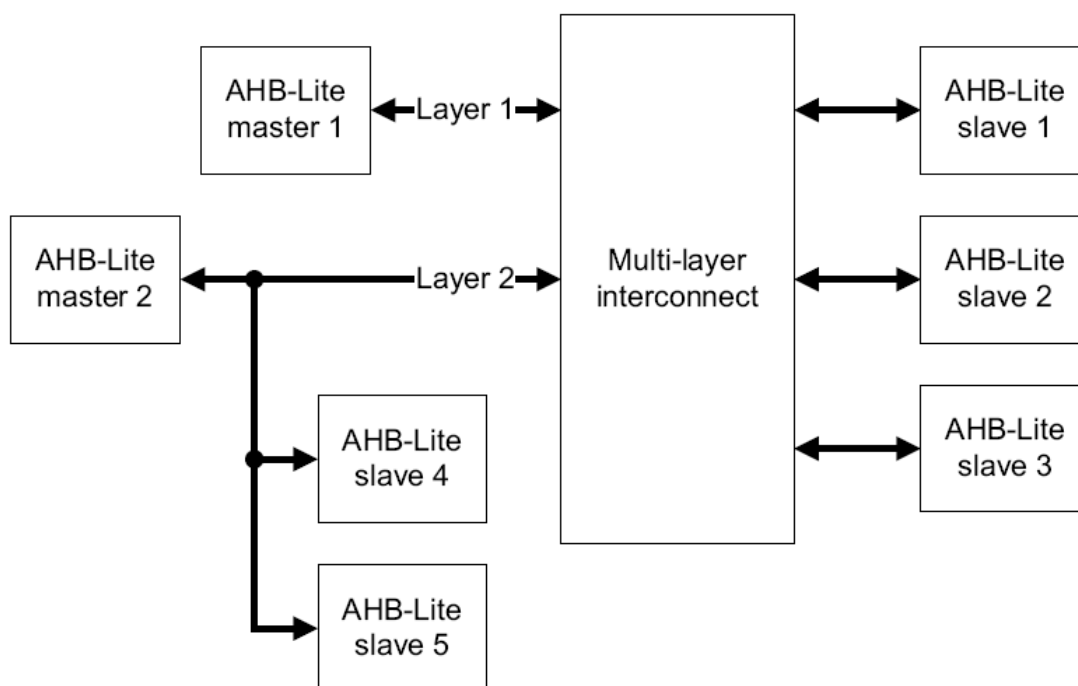


图1-4 多层AHB-Lite框图

在图1-4中，Master1和Master2都可以访问Slave1、Slave2、Slave3。多层互连组件必须通过实现三个共享Slave的仲裁电路来避免两个Master同时访问同一个Slave。

Master1不能请求访问Slave4和Slave5，因此可以作为Master2的本地连接设备。这样减少了多层互连组件的复杂度。

多层互连组件的设计超出了本文的范围。可以查询[Multi-layer AHB Overview](#) 来获取关于实现多层AHB-Lite互连的信息。

## 第二章 信号描述

本章描述了协议信号，包括以下小节：

- 2.1 全局信号
- 2.2 *Master* 信号
- 2.3 *Slave* 信号
- 2.4 译码器信号
- 2.5 多路器信号

注：所有的AHB-Lite信号都由字母H前缀来区分系统设计中其他相似名字的信号。

## 2.1 全局信号

表2-1 列出了协议中的全局信号

表2-1 全局信号

名称	源	描述
HCLK	Clock Source	总线时钟, 所有的信号时序变化都在HCLK的上升沿触发. 在7.1.1 时钟 节可以获得更多信息.
HRESETn	Reset controller	总线复位信号, 低有效, 复位整个系统和总线. 这是AHB-Lite信号中唯一的低有效信号. 在7.1.2 复位 节可以获得更多信息.

## 2.2 Master信号

表2-2 列出了master生成的信号

表2-2 Master信号

名称	目的	描述
HADDR[31:0]	Slave和译码器	系统的32位地址总线.
HBURST[2:0]	Slave	Burst类型表示当前传输是否是single传输还是构成burst传输的一拍. 支持固定长度为4、8和16的burst. Burst可以是递增型或者是回卷型. 系统同样支持未定义长度的递增型burst. 在3.5 Burst操作 节可以获得更多信息.
HMASKLOCK	Slave	当为高电平时, 表示当前传输是一个锁定序列的一部分. 它与地址总线和控制信号有着相同的时序. 在3.3 锁定传输 节可以获得更多信息.
HPROT[3:0]	Slave	保护控制信号提供总线访问的额外信息, 设置它的初衷是方便一些模块需要实现多种的级别的保护功能. 保护控制信号, 表示当前传输是否是一个操作码获取还是数据访问, 还可以表示传输是否是一个特权模式访问还是用户模式访问. 对于带MMU功能的Master, 保护信号也可以指示是否当前访问可以高速缓存(Cachable)或者Buffer缓存. 在3.7 保护控制 节可以获得更多信息.
HSIZE[2:0]	Slave	表示传输的大小, 典型值有字节、半字和字. 协议支持更大的传输大小, 支持最大值为1024位. 在3.4 传输大小 节可以获得更多信息.
HTRANS[1:0]	Slave	指示当前传输的传输类型, 可以是: <ul style="list-style-type: none"> <li>• IDLE</li> <li>• BUSY</li> </ul>

		<ul style="list-style-type: none"><li>• NONSEQUENTIAL</li><li>• SEQUENTIAL.</li></ul> 在3.2 传输类型 节可以获得更多信息.
HWDATA[31:0] <sup>a</sup>	Slave	写数据总线 在总线写操作时从Master向Slave搬数. 建议最小的数据总线位宽为32位. 当然, 位宽也可以扩展到支持更高带宽的操作. 在6.1数据总线 节可以获得更多信息.
HWRITE	Slave	表示传输方向. 高电平表示写操作, 低电平表示读操作. 它需要和地址总线同时有效, 但在burst传输中必须保持不变. 在3.1 基本传输 节可以获得更多信息.

a. 写数据总线位宽并不限制在32位. 在6.2 数据总线位宽 节列出了允许的其他数据位宽值.

2.3 Slave信号

表2-3 列出了Slave生成的信号

表2-3 Slave信号

名称	目的	描述
HRDATA[31:0] <sup>a</sup>	多路器	在读操作时，此信号从被选中的Slave向多路器传输数据。多路器再将数据传输到Master。 建议最小的数据总线位宽为32位。当然，位宽也可以扩展到支持更高带宽的操作。 在6.1数据总线节可以获得更多信息。
HREADYOUT[2:0]	多路器	当信号为高电平，表示总线上的传输结束了。此信号可以通过驱动为低电平来增加传输的拍数。 在4.2 总线互连节可以获得更多信息。
HRESP	多路器	传输的响应信号，在通过多路器后，给Master提供传输的状态信息。 当信号为低电平，表示传输的状态正常。 当信号为高电平，表示传输的状态错误。 在5.1 Slave传输响应节可以获得更多信息。

a. 读数据总线位宽并不限制在32位。在6.2 数据总线位宽节列出了允许的其他数据位宽值。



2.4 译码器信号

表2-4 列出了译码器生成的信号：

表2-4 译码器信号

名称	目的	描述
HSELx <sup>a</sup>	Slave	每个AHB-Lite Slave 拥有它自己的选择信号HSELx，并且这个信号指示了当前传输意图选择的Slave. 当Slave被选中，它必须也要监控HREADY信号的状态来确保在它响应当前传输之前的那个总线传输已经完成. HSELx信号是地址总线的组合逻辑译码. 在4.1 地址译码 小节可以获得更多信息.

a. 在一个系统中，**HSELx**中的字母x必须每个Slave有一个特定的值. 例如，HSEL\_S1，HSEL\_S2, HSEL\_Memory.

注：通常译码器也要提供给多路器**HSELx**信号，或者是由**HSELx**信号驱动的单bit 或多bit信号，用来使能多路器透传从被选中的Slave到Master的信号. 特别重要的是：这些额外的给多路器的控制信号必须在总线传输数据相也要有效.

2.5 多路器信号

表2-5 列出了多路器生成的信号：

表2-5 多路转换器信号

名称	目的	描述
HRDATA[31:0] <sup>a</sup>	Master	读数据总线，由译码器输入的选择信号选出 <sup>a</sup> .
HREADY[2:0]	Master和Slave	当信号为高电平，HREADY信号指示给Master和所有的Slave，前一个传输已经完成了. 在4.2 总线互连 小节可以获得更多信息.
HRESP	Master	传输的响应信号，由译码器输入的选择信号选出 <sup>a</sup> .

a. 由于HRDATA[31:0]和HRESP信号只是通过多路器，且保留相同的名字，在表2-3 提供了信号的详细描述.

## 第三章 传输

本章描述了AHB-Lite读写传输,包括以下小节:

- 3.1 基本传输
- 3.2 传输类型
- 3.3 锁定传输
- 3.4 传输大小
- 3.5 *Burst*操作
- 3.6 等待传输
- 3.7 保护控制

3.1 基本传输

一个AHB-Lite传输由以下两个相位组成：  
**地址相位** 保持一个HCLK周期除非被前一个传输延长；  
**数据相位** 也许需要若干个HCLK周期. Slave用HREADY信号控制完成这个传输的时钟周期数.

- HWRITE 控制数据传输的方向：
- HWRITE为高电平时，表示当前传输是写传输，由Master发送数据到写数据总线，HWDATA[31:0].
  - HWRITE为低电平时，表示要执行一个读传输，Slave必须产生数据在读数据总线，HRDATA[31:0].

最简单的传输由一个时钟周期的地址相位和一个时钟周期的数据相位组成，没有等待状态. 图3-1是一个简单的读传输，图3-2是一个简单的写传输.

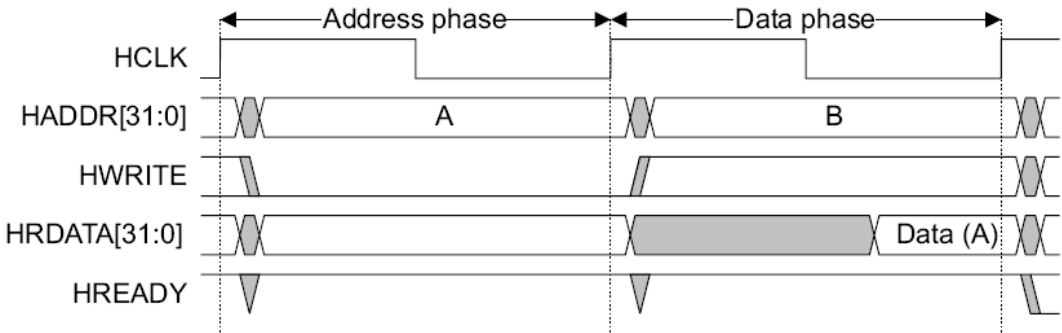


图3-1 读传输

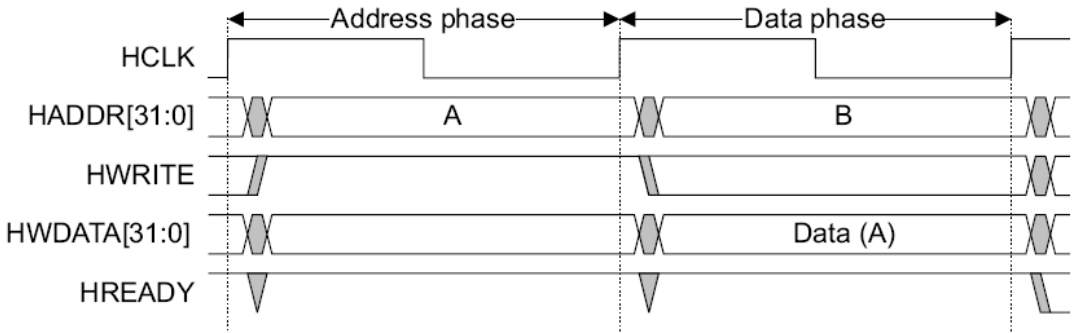


图3-2 写传输

- 一个不含有等待状态的简单传输：
1. 在**HCLK**的上升沿，Master驱动地址和控制信号到总线上；
  2. Slave在下一个**HCLK**上升沿采集到地址和控制信息；
  3. 当Slave采集到地址和控制后，它可以驱动**HREADY**信号做出相应. Master将在第三个**HCLK**上升沿采集到响应信号.

这个简单的例子阐述了传输中的地址相位和数据相位是怎么在不同的时钟周期发生的. 一个传输的地址相位可以同时也是上一个传输的数据相位. 这种地址相位和数据相位的重叠是总线流水属性的基本原则，由此提高总线操作的性能，还为Slave提供了足够的时间来响应传输.

Slave可以给任一个传输插入等待状态来为完成总线传输获得更多的时间.

图3-3 展示了一个具有两个等待状态的读传输：

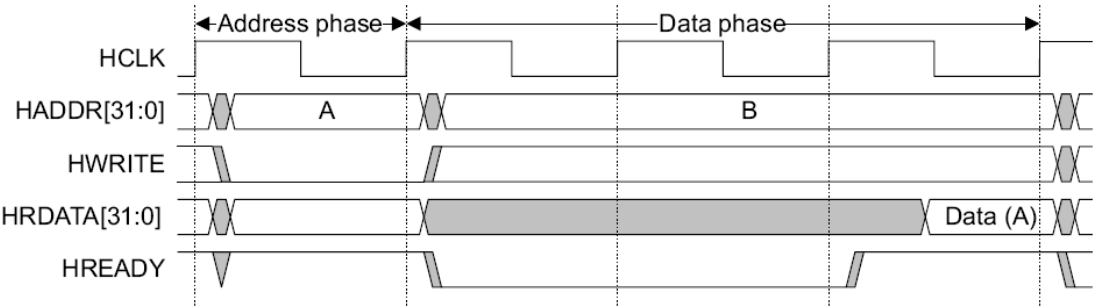


图3-3 具有等待状态的读传输

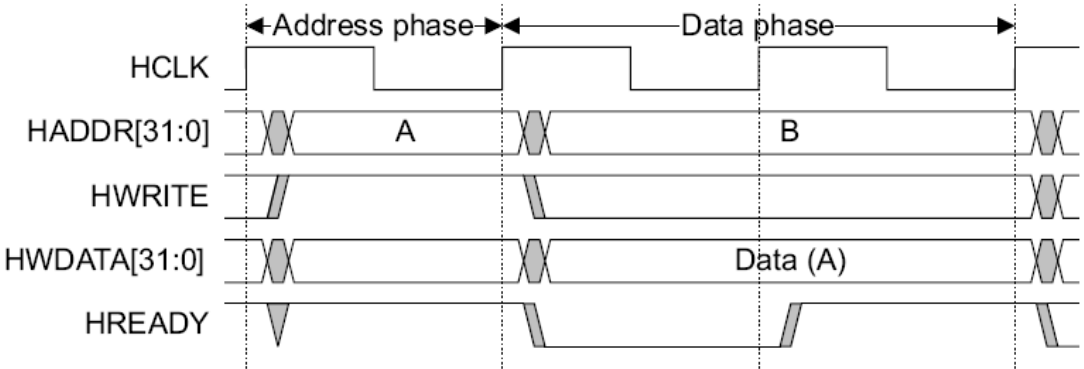


图3-4 具有等待状态的写传输

注：对于写操作，Master在被扩展的时钟周期要保持数据稳定。对于读操作在读传输完成前Slave不提供读数据。

当一个传输以这种方式扩展了传输拍数，它将对下一个传输产生副作用。如图3-5所示，图中三个传输，且三个地址A、B、C没有任何关联，地址C的地址相位(因为B的数据相位扩展了一拍)被扩展一拍了。

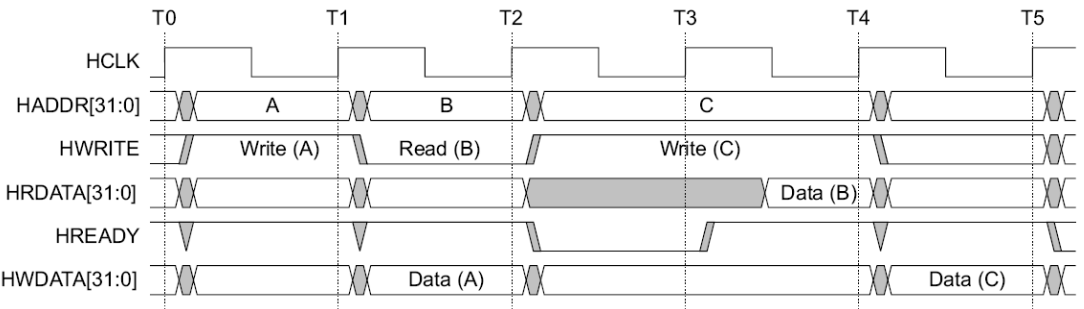


图3-5 多传输

在图3-5中：

- 地址A和地址C的传输是不包括等待状态的；
- 地址B的传输包含一拍等待状态；
- 由于地址B的传输在数据相位增加了一个时钟周期的等待扩展，影响到了地址C的传输，使得地址C的传输在地址相位增加了一个时钟周期的等待扩展。

3.2 传输类型

传输类型被分为4类，由**HTRANS[1:0]**控制。由表3-1列出：

表3-1 传输类型编码

HTRANS[1:0]	类型	描述
2'b00	IDLE	表示没有数据传输请求. Master利用IDLE传输表示它不想启动一个数据传输. 推荐Master用IDLE传输结束 锁定传输. Slave必须提供零等待状态的OKAY响应给IDLE传输，并且Slave必须忽略IDLE传输，不做任何操作.
2'b01	BUSY	BUSY传输类型，使Master可以在一个Burst中间插入空的周期. 这种传输类型表示Master将继续Burst传输，但不能马上进行下一拍传输. 当Master使用BUSY传输类型时，地址和控制信号必须是下一拍burst的信号. 只有未定义长度的Burst可以将Busy传输作为一个Burst的最后一拍. 在Burst在BUSY传输终止节可以获得更多信息. Slave必须始终为BUSY传输提供零等待的OKEY响应，并且必须忽略BUSY传输，不做任何操作.
2'b10	NONSEQ	表示当前传输是一个Single传输，或者是一个Burst传输的第一拍. 地址和控制信号与前一拍传输没有关联. Single传输可以看作长度为1的Burst传输，因此传输类型是NONSEQUENTIAL(非连续).
2'b11	SEQ	Burst传输中NONSEQ后续的传输是SEQUENTIAL传输，地址必须和前一拍传输相关. 控制信息与前一拍传输完全相同. 地址=前一个传输的地址+传输大小，单位是字节，传输大小由 <b>HSIZE[2:0]</b> 定义. 在回卷Burst的场景中，传输地址在地址边界回卷.

图3-6展示了NONSEQ、BUSY和SEQ传输类型的使用

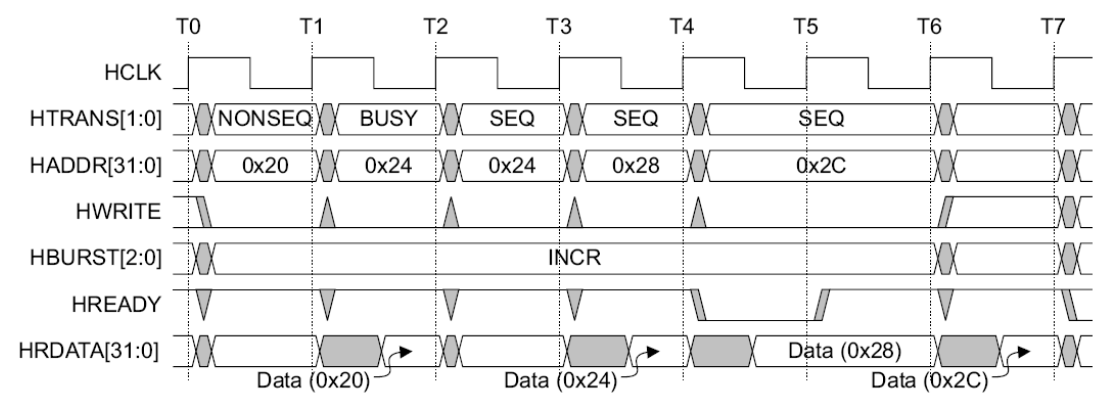


图3-6 传输类型举例

在图3-6中:

- T0-T1 由一个NONSEQ传输开始一个4拍的读操作.
- T1-T2 Master不能立即提供第二拍的数据, 插入BUSY传输延迟第二拍的数据, Slave提供第一拍的读数据
- T2-T3 Master现在可以开始第二拍的传输, 因此发出一个SEQ的传输. Master忽略Slave在读数据总线HRDATA[31:0]提供的任何信息.
- T3-T4 Master执行第三拍传输. Slave提供第二拍的读数据.
- T4-T5 Master执行最后一拍传输. Slave无法马上提供第三拍的数据, 用拉低**HREADY**插入一拍等待状态.
- T5-T6 Slave提供第三拍的读数据.
- T6-T7 Slave提供最后一拍的读数据.

3.3 锁定传输

如果Master要求锁定访问，则它必须置位**HMASTLOCK**信号。这个信号指示给所有Slave当前传输不可被打断，必须在任何其他传输执行之前执行此传输。  
典型的锁定传输用在保持信号的完整性，确保在微处理器SWP指令期间Slave不执行其他操作。

图3-7展示了**HMASTLOCK**信号在SWP指令中的例子：

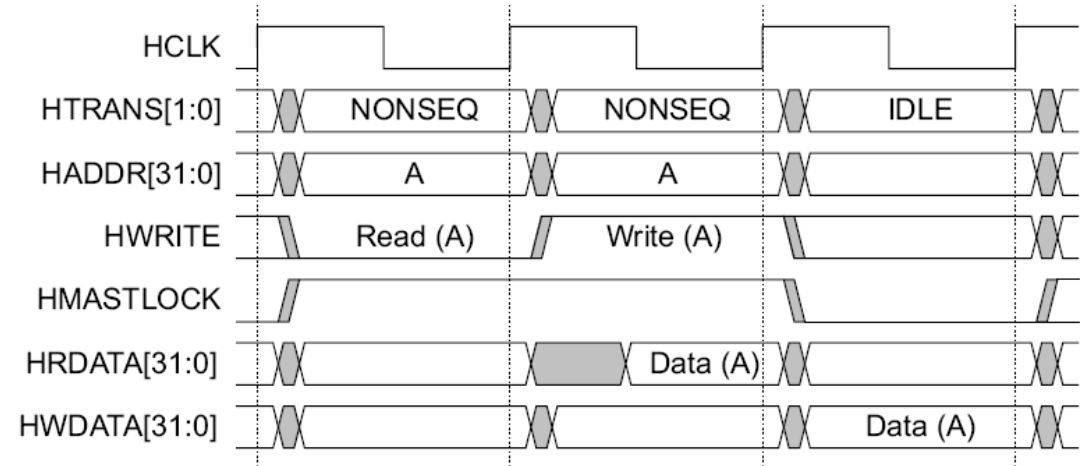


图3-7 锁定传输

注：在一个锁定传输后，建议Master插入一拍IDLE传输。

大部分Slave不必实现**HMASTLOCK**，因为它们只能按照接收的请求顺序进行传输。但是在以一个包含多个Master的系统中，例如*Muli-Port Memory Controller*，Slave可以被多个Master访问，则必须实现**HMASTLOCK**功能。



3.4 传输大小

HSIZE[2:0]信号指示了一个数据传输的大小. 表3-2列出了所有传输大小的可能:

表3-2 传输大小编码

HSIZE[2]	HSIZE[1]	HSIZE[0]	Size(bits)	描述
0	0	0	8	Byte
0	0	1	16	Halfword
0	1	0	32	Word
0	1	1	64	Doubleword
1	0	0	128	4-word line
1	0	1	256	8-word line
1	1	0	512	-
1	1	1	1024	-

注: HSIZE设置的传输大小必须小于等于数据总线的位宽. 例如, 一个32位数据总线的系统, HSIZE只能配置为b000, b001, b010.

HSIZE信号和HBURST信号关联起来, 决定了回卷Burst传输的地址边界.

HSIZE信号和地址总线保持严格的时序相同. 但是, 在Burst传输期间必须保持常值.

3.5 Burst操作

本协议定义了长度为4、8、16的Burst，还有未定义长度Burst(undefined length bursts)，和单传输。固定长度Burst支持递增和回卷两种模式：

- 递增型Burst访问连续位置，并且每拍传输的地址是前一个地址递增。
- 回卷Burst在跨越地址边界的时候回卷。地址边界由传输大小和在这个burst的拍数计算得出。传输大小由**HSIZE**控制，传输中的拍数递增值由**HBURST**控制。例如，一个传输大小为字(4字节)的4拍回卷burst，在16字节边界发生回卷。因此，如果一个burst传输的开始地址是0x34，构成它的四个传输地址依次是：0x34，0x38，0x3C和0x30。

**HBURST[2:0]**控制burst类型。表3-3列出了所有burst类型。

表3-3 Burst信号编码

HBURST[2:0]	类型	描述
3'b000	SINGLE	单拍传输。
3'b001	INCR	未定义长度的递增Burst
3'b010	WRAP4	4拍回卷Burst
3'b011	INCR4	4拍递增Burst
3'b100	WRAP8	8拍回卷Burst
3'b101	INCR8	8拍递增Burst
3'b110	WRAP16	16拍回卷Burst
3'b111	INCR16	16拍递增Burst

Master禁止发起一个跨越1KB地址边界的递增型Burst。

Master可以用以下任何一种方式发起一个单拍传输：

- Single burst
- 长度为1的未定义长度的Burst。

注：burst大小指的是在一个burst中传输拍数，而不是传输的字节数。计算一次burst传输中数据的总数的方法是，拍数X每一拍的字节数。每一拍的字节数由**HSIZE[2:0]**指示。

一个Burst的所有传输必须对齐**HSIZE**的地址边界。例如，必须对齐word传输的地址到word地址边界(**HADDR[1:0]=2'b00**)，对齐halfword传输的地址到halfword地址边界(**HADDR[0]=1'b0**)。IDLE传输的地址也必须对齐，否则在仿真时，总线监视器将报警告。

3.5.1 BUSY传输后Burst终止

在一个burst启动后，当Master需要更多的时间准备下一拍传输时，它可以插入BUSY传输。

一个未定义长度的burst，INCR，Master可以插入BUSY传输，然后再判断没有更多的数据要传输了。在这种情况下，Master可以再发起NONSEQ或者IDLE传输来终止未定义长度的burst传输。

协议不允许Master用BUSY传输终止固定长度的Burst传输：

- 递增型burst：INCR4，INCR8和INCR16

- 回卷型burst: WRAP4, WRAP8和WRAP16.

这些固定长度的burst必须由一个SEQ传输终止.

Master不允许在一个单拍burst传输后立刻发起一个BUSY传输. 单拍Burst后必须是一个IDLE或者NONSEQ传输.

### 3.5.2 Burst 提前终止

Burst可以在以下情况被终止:

- Slave error response错误响应
- 多层互连的终止

#### Slave错误响应

如果Slave提供了ERROR响应, 则Master必须取消掉剩下的传输. 但是, 这并不是强制要求, Master仍然可以继续传输完剩下的内容.

如果Master没有完成Burst, 当它再次访问相应的Slave时, 没有必要重新发起之前的Burst. 例如, 如果Master只完成了8拍burst的3拍传输, 当它再次访问对应的Slave时不需要完成剩下的5拍传输.

#### 多层互连终止

尽管Master不允许提前终止一个Burst传输, Slave必须设计为burst传输完成之前就一直可以正常工作.

在一个多个Master系统中, 包含多层互连组件, 它可以终止Burst传输这样另一个Master可以获得Slave的访问权. 当这种情况发生时, Slave必须终止前一个Master的Burst操作, 然后马上响应新的Master的访问.

### 3.5.3 Burst举例

以下几节是多种Burst的例子:

- 4拍回卷Burst, WRAP4
- 4拍递增Burst, INCR4
- 8拍回卷Burst, WRAP8
- 8拍递增Burst, WRAP8
- 未定义长度Burst, INCR

## 4拍回卷Burst, WRAP4

图3-8是一个4拍回卷Burst写传输, 在第一拍传输插入了一拍等待状态.

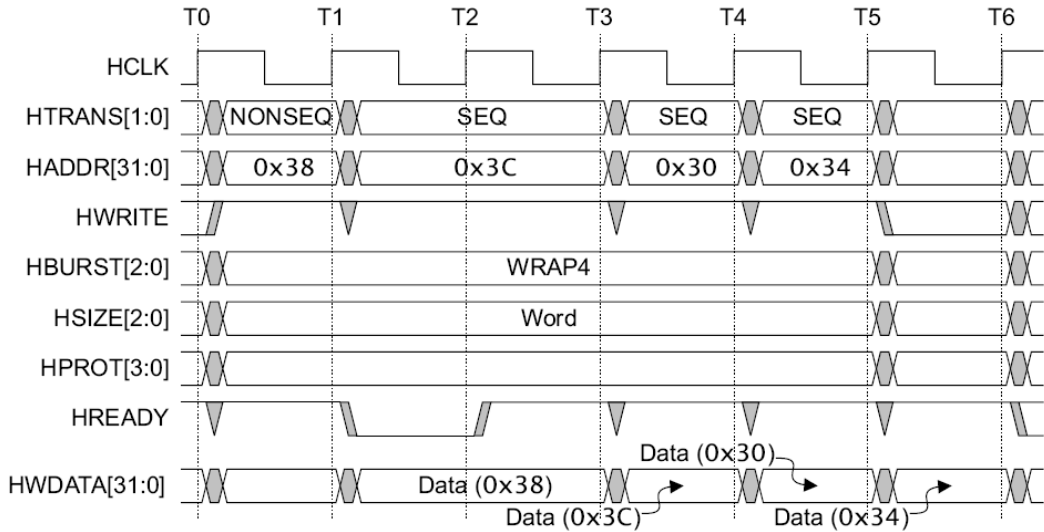


图3-8 4拍回卷Burst

因为上图的Burst是4拍回卷的word传输，地址在16字节边界回卷，所以在地址0x3C后下一拍地址是0x30。

4拍递增Burst， INCR4

图3-9是一个4拍递增读传输的Burst，在第一拍传输插入了一拍等待状态。在此例中，地址在16字节边界不回卷，地址0x3C后下一拍地址是0x40。

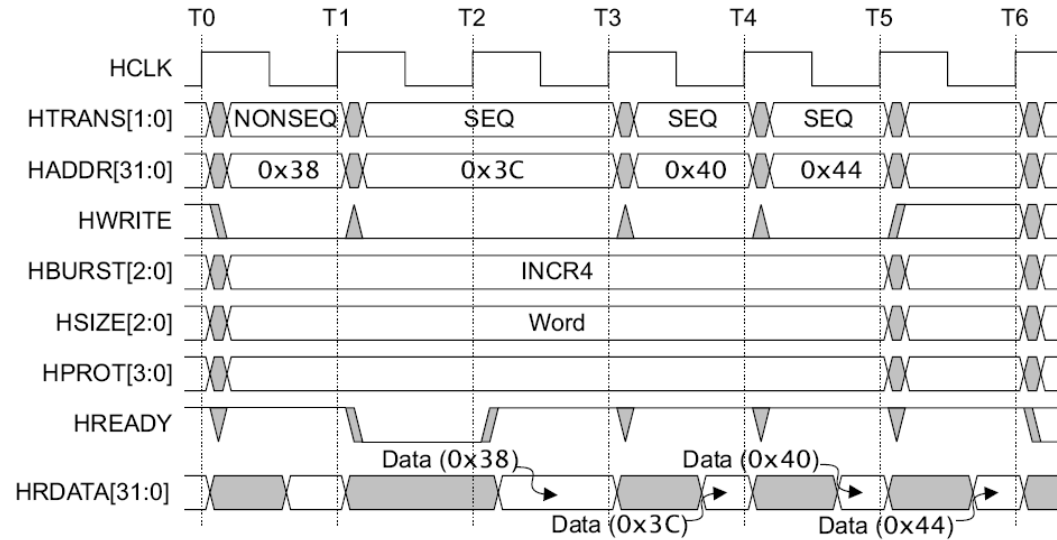


图3-9 4拍递增Burst

8拍回卷Burst， WRAP8

图3-10是一个8拍回卷Burst的读传输。

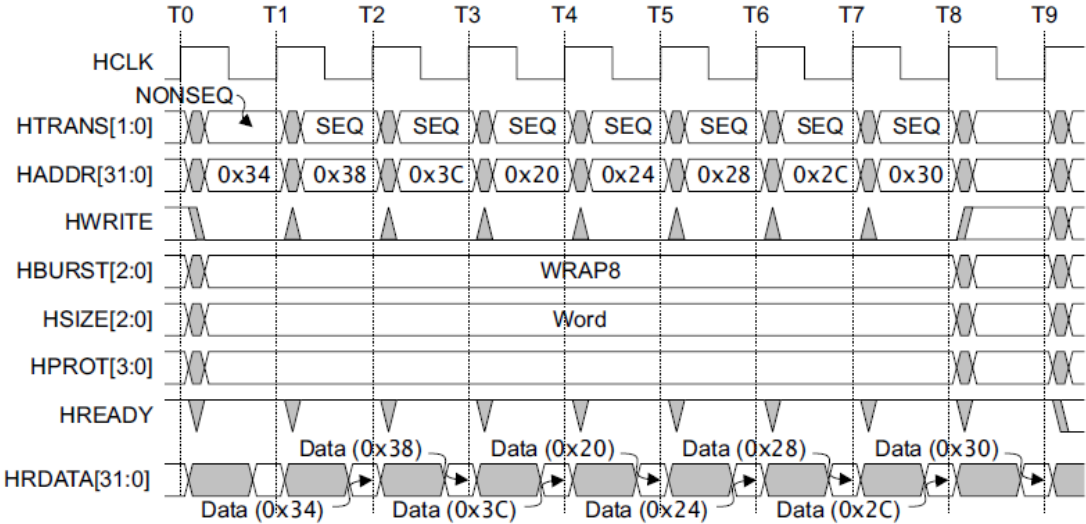


图3-10 8拍回卷Burst

由于这是一个8拍Burst的Word传输，地址在32字节边界回卷，因此传输地址0x3C的下一拍地址是0x20。

8拍递增Burst，WRAP8

图3-11是一个8拍递增写传输的Burst.

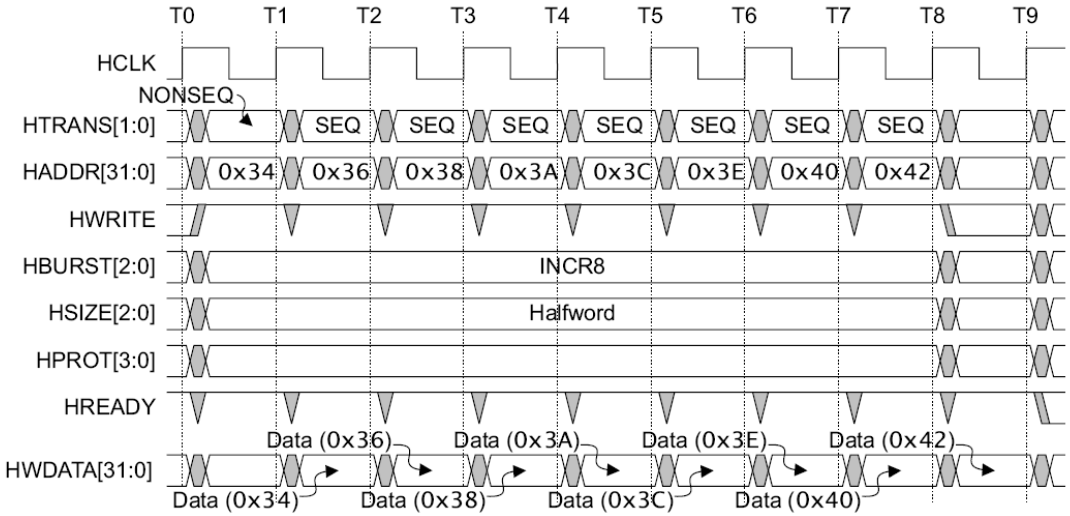


图3-11 8拍递增Burst

这个Burst是一个半字传输，因此地址递增补偿为2字节。因为burst是一个递增型传输，地址在16字节边界递增。

未定义长度Burst，INCR

图3-12是未定义长度的Burst.

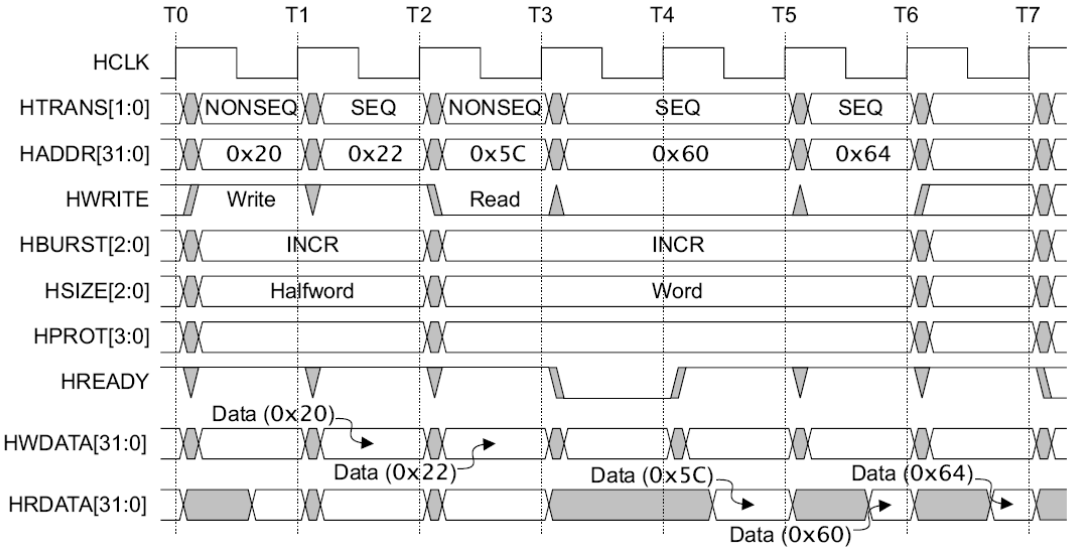


图3-12 未定义长度Burst

图3-12展示了两个Bursts:

- 第一个Burst是一个写传输，包括2拍半字传输，地址从0x20开始. 传输地址递增步长为2字节.
- 第二关Burst是一个读传输，包括3拍字传输，起始地址0x5C. 传输地址递增补偿为4字节.

3.6 等待传输

当Slave需要更多的时间提供数据或采集数据的时候，它用**HREADY**信号插入等待状态. 在一个等待传输期间，**Master**被限制更改传输类型和传输地址. 下面两节将详细描述：

- 3.6.1在等待状态期间更改传输类型
- 3.6.1在等待状态期间更改传输类型 .

3.6.1在等待状态期间更改传输类型

当Slave请求等待状态，**Master**禁止更改传输类型，除非以下情况：

- *IDLE* 传输
- *BUSY* 传输，固定长度burst
- *BUSY* 传输，未定义长度的Burst

IDLE传输

在等待状态期间，**Master**被允许将传输状态从IDLE更改为NONSEQ. 当**HTRANS**更改为NONSEQ，**Master**必须保持**HTRANS**不变，直到**HREADY**为高电平.

图3-13所示，一个插入等待状态的Single传输，**Master**将传输类型由IDLE变为NONSEQ.

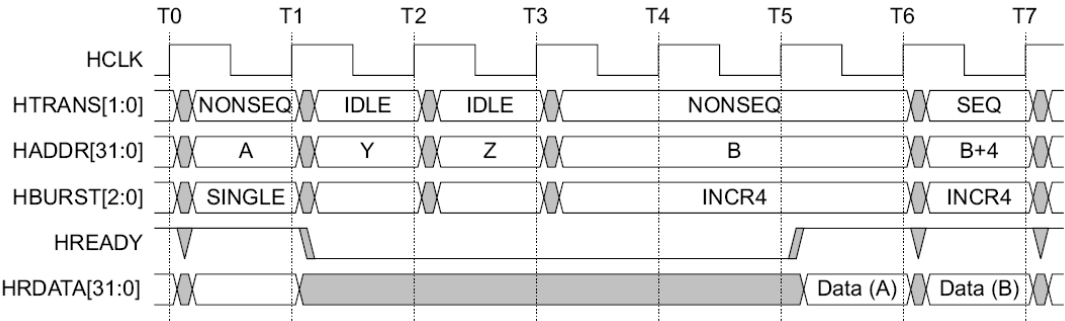


图3-13 等待传输，IDLE到NONSEQ

在图3-13中：

- T0-T1 Master发起了一个Single传输，地址为A.
- T1-T2 Master插入一个IDLE传输，地址为Y.  
Slave插入一个等待状态，**HREADY**为低电平.
- T2-T3 Master插入一个IDLE传输，地址为Z.
- T3-T4 Master改变传输类型为NONSEQ，并且发起一个INCR4的Burst传输到地址B.
- T4-T6 由于**HREADY**为低电平，**Master**保持**HTRANS**不变.
- T5-T6 地址A的Single传输完成，**HREADY**为高电平，**Master**开始Burst传输地址B的第一拍.
- T6-T7 完成INCR4 Burst传输的地址B的传输，**Master**开始地址为B+4的下一拍传

输.

BUSY传输，固定长度burst

一个固定长度的Burst插入等待传输期间，Master可以改变传输类型，又BUSY改为SEQ. 当HTRANS传输类型改为SEQ，Master必须保持HTRANS不变，直到HREADY为高电平.

注：由于BUSY传输只能插入到Burst传输相邻两拍之间，所以此节场景不能应用在SINGLE类型的Burst. 因此场景有以下几种：

- INCR4，INCR8和INCR16
- WRAP4，WRAP8和WRAP16.

图3-14所示，一个含有等待传输的固定长度Burst，传输类型由BUSY变为SEQ.

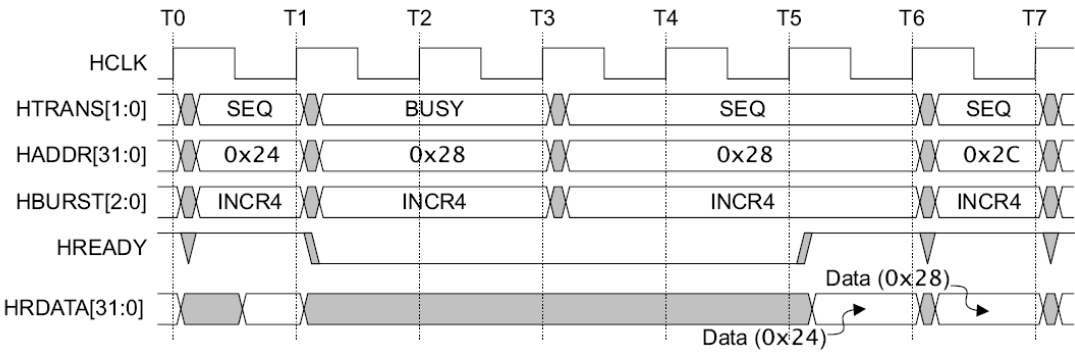


图3-14 等待传输，BUSY到SEQ，固定长度Burst

在图3-14中：

- T0-T1 Master发起了INCR4传输的新的一拍，地址为0x24.
- T1-T3 Master插入了一个BUSY传输，地址为0x28.  
Slave插入等待状态，HREADY为低电平.
- T3-T4 Master改变传输类型为SEQ，发起了下一拍传输，地址为0x28.
- T4-T6 Master保持HTRANS不变，直到HREADY为高.
- T5-T6 地址为0x24的这一拍传输完成，HREADY为高电平.
- T6-T7 第三拍INCR4的传输，地址为0x28的传输完成，Master发起最后一拍传输，地址0x2C.

BUSY传输，未定义长度的Burst

在一个未定义长度的Burst等待传输期间，当HREADY为低电平时，Master允许改变传输类型，从BUSY到其他任何类型. 如果传输类型改为SEQ，Burst继续传输；如果传输类型改为IDLE或者NONSEQ，Burst被终止.

图3-15所示，在一个未定义长度的Burst等待传输期间，传输类型由BUSY改为NONSEQ.



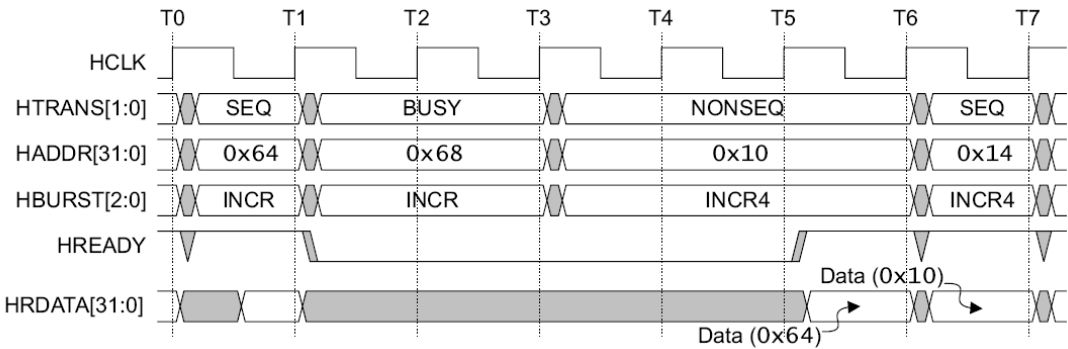


图3-15 等待传输，**BUSY**到**NONSEQ**，未定义长度**Burst**

在图3-15中：

- T0-T1 Master发起了INCR传输的新的一拍，地址为0x64.
- T1-T3 Master插入了一个BUSY传输，地址为0x68.  
Slave插入等待状态，HREADY为低电平.
- T3-T4 Master改变传输类型为NONSEQ，发起了下一拍传输，地址为0x10.
- T4-T6 Master保持**HTRANS**不变，直到**HREADY**为高.
- T5-T6 地址为0x64的这一拍传输完成，**HREADY**为高电平，未定义长度的Burst传输完成，Master开始了地址为0x10的新的一拍传输.
- T6-T7 第一拍INCR4传输，地址为0x10的传输完成，Master发起下一拍传输，地址0x14.

3.6.2 在等待状态期间改变地址

当Slave请求等待状态，Master可以只改变一次传输地址，如下描述

- 在IDLE传输期间
- 在ERROR响应后

在IDLE传输期间

在一个等待状态传输期间，Master可以改变IDLE传输地址. 当HTRANS传输类型改变为NONSEQ，Master必须保持地址不变，直到HREADY为高电平.

图3-16所示，一个包含等待传输的SINGLE传输，地址在IDLE传输时发生改变.

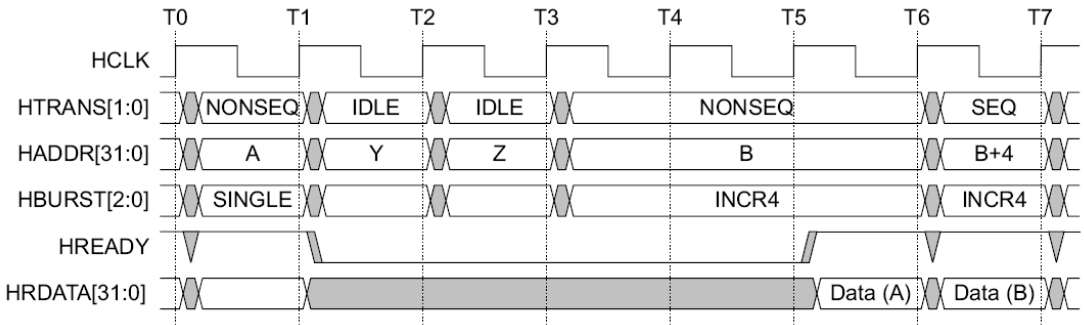


图3-16 在等待传输中改变地址，具有一个IDLE传输

- 在图3-16中：
- T0-T1 Master发起了SINGLE BURST传输到地址A.
  - T1-T2 Master插入了一个IDLE传输到地址Y.  
Slave插入等待状态，**HREADY**为低电平.
  - T2-T3 Master插入了一个IDLE传输到地址Z.
  - T3-T5 Master改变**HTRANS**为**NONSEQ**，指示这是一个**INCR4**的传输从地址B开始. 直到**HREADY**为高，不允许再有地址改变.
  - T5-T6 地址A的SINGLE Burst传输完成，**HREADY**为高电平，Master开始地址B传输的第一拍.
  - T6-T7 第一拍**INCR4**传输，地址为B的传输完成，Master发起下一拍传输，地址B+4.

在ERROR响应后

在等待传输期间，如果Slave回复**ERROR**响应，Master可以在**HREADY**为低电平的时候改变地址. 在5.1.3 错误响应小节可以获得更多信息.  
图 3-17 所示，一个等待传输，地址在 Slave 的 **ERROR** 响应后发生改变.

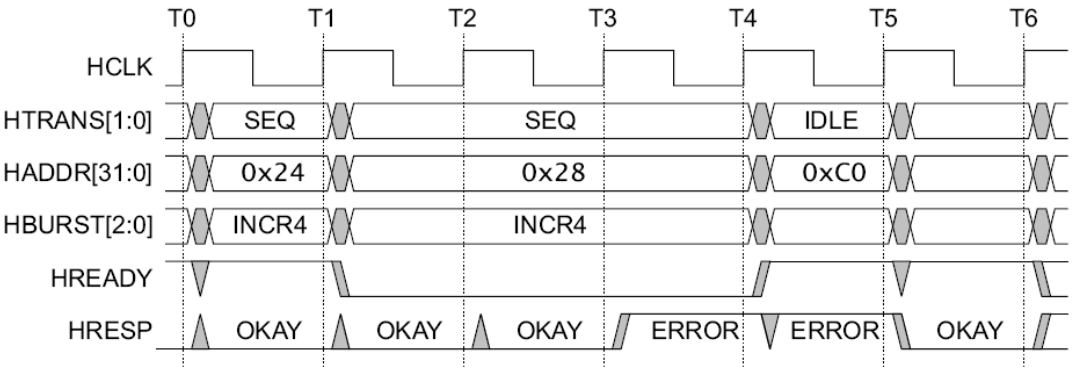


图3-17 **ERROR**之后，在等待传输期间改变地址

- 在图 3-17 中：
- T0-T1 Master发起了地址为0x24的Burst传输地址.
  - T1-T3 Master发起了下一拍Burst传输地址0x28.  
Slave的响应是**OKAY**.
  - T3-T4 Slave响应**ERROR**.
  - T4-T5 Master改变**HTRANS**为**IDLE**，并且允许**HREADY**为低电平的时候改变地址.  
Slave完成**ERROR**响应.
  - T5-T6 地址0xC0的Slave响应为**OKAY**.

3.7 保护控制

保护控制信号，**HPROT[3:0]**，提供总线访问的附加信息，设置保护信号的初衷是方便实现多种级别的保护。

当总线传输为以下场景保护信号起作用：

- 操作码获取还是数据访问
- 特权模式访问还是用户模式访问。

对于带MMU单元的Master，保护信号也可以指示当前访问是否是Cacheable或者Bufferable。表3-4列了**HPROT**信号编码。

表3-4 保护信号编码

HPROT[3] Cacheable	HPROT[2] Bufferable	HPROT[1] Privileged	HPROT[0] Data/opcode	描述
-	-	-	0	操作码获取
-	-	-	1	数据访问
-	-	0	-	用户访问
-	-	1	-	特权访问
-	0	-	-	不缓冲
-	1	-	-	可缓冲
0	-	-	-	不可高速缓存
1	-	-	-	可高速缓存

注：很多Master不具备产生准确保护信号的能力。如果Master不支持产生精确的保护信号，ARM公司建议：

- Master设置HPROT为4'b0011给相应的non-cacheable，non-bufferable，特权数据访问
- Slave不使用HPROT信号除非有绝对的需要。

HPROT控制信号和地址总线有完全一样的时序。但是，它们在一个Burst传输中必须保持不变。

## 第四章 总线互连

本章描述了AHB-Lite系统需要的互连逻辑, 包括以下小节:

- 4.1 地址译码
- 4.2 总线互连

### 4.1 地址译码

中央地址译码器提供选择信号, HSEL<sub>x</sub>, 给总线上每一个Slave. 选择信号是地址总线高位的组合逻辑译码. 鼓励使用简单的地址译码电路而不是复杂译码逻辑, 以确保高速操作.

Slave必须只能在HREADY为高电平(表示当前传输已经完成)时, 采样HSEL<sub>x</sub>, 地址和控制信号. 在某种情况下, 有可能HSEL<sub>x</sub>在HREADY为低电平时置位为高, 但是必须当前传输完成时改变选中的Slave.

单个Slave可以分配的最小地址空间是1KB. 所有的Master设计为执行的递增传输地址不能跨越1KB地址边界. 这样保证了Burst传输不会跨越地址译码边界.

图4-1所示, HSEL<sub>x</sub>信号由译码器产生.

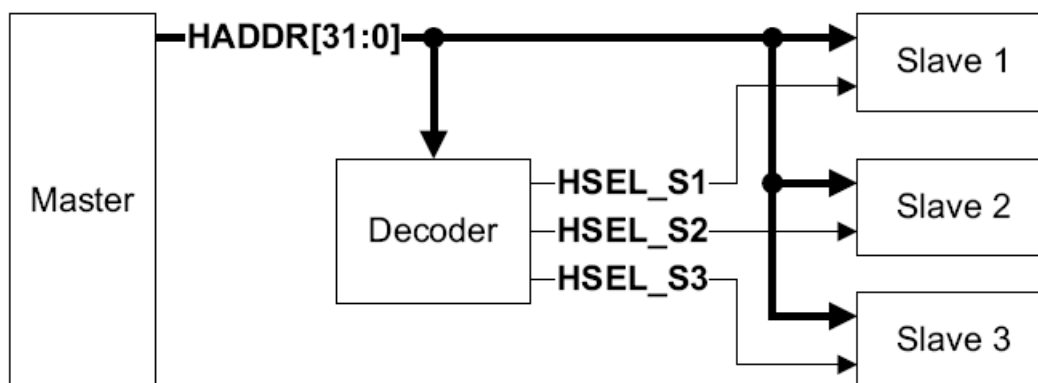


图4-1 Slave选择信号

#### 4.1.1 默认Slave

如果一个系统不包含一个完整的可以填满的地址空间, 设计者必须实现一个附加的默认的Slave, 当Master访问到不存在的地址空间时提供响应.

如果一个NONSEQUENTIAL或者SEQUENTIAL传输师徒访问一个不存在的地址, 默认Slave将提供一个错误响应.

IDLE或者BUSY传输访问不存在的地址, 默认Slave回复一个无等待状态的OKAY.

## 4.2 总线互连

AHB-Lite协议用一个中央读数据多路器互连电路. Master驱动地址和控制信号给所有的Slave, 译码器选择相应的Slave. 任何从选中的Slave响应的数据, 通过读数据多路器传输给Master.

图4-2所示, 一个具有3个Slave的AHB-Lite, 需要的多路器互连结构:

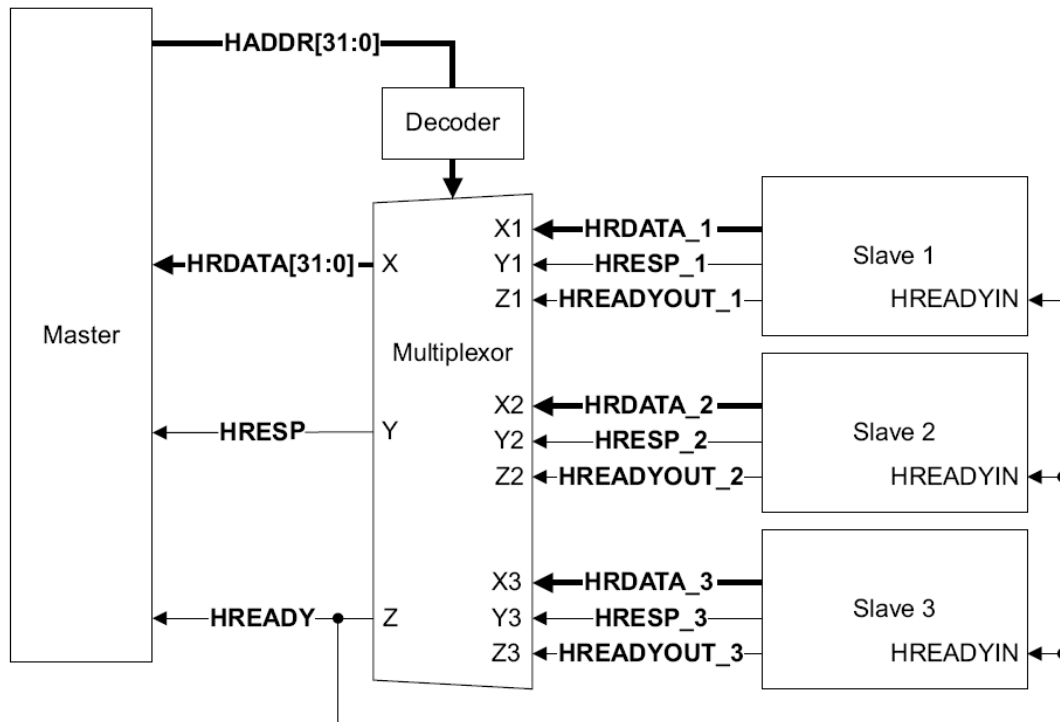


图4-2 多路转换器互连

## 第五章 Slave响应信号

本章描述了Slave响应信号，包含以下小节：

- 5.1 *Slave* 传输响应

5.1 Slave传输响应

在Master开始一个传输之后，Slave控制这个传输的进程。Master不能取消已经开始的传输。

当Slave被访问时，必须提供响应，来指示传输的状态。传输状态由**HRESP**信号提供。表5-1列出了**HRESP**的状态：

表5-1 HRESP信号

HRESP	响应	描述
0	OKAY	传输成功完成或者Slave需要更多的时钟周期才能完成请求。 <b>HREADY</b> 信号指示传输是否在等待还是已完成。
1	ERROR	传输期间发生了ERROR。错误情况必须通知到Master，这样Master察觉到传输没成功。 发生错误，需要2个时钟周期的ERROR响应信号，在第二个时钟周期 <b>HREADY</b> 置位。

表5-1所示，完整的传输响应是**HRESP**和**HREADY**信号的组合。表5-2列出了基于这两个信号的完整传输状态。

表5-2 传输响应

HRESP	HREADY	
	0	1
0	传输等待	传输成功完成
1	ERROR 响应，第一个时钟周期	ERROR 响应，第二个时钟周期

这意味着Slave可以用以下三种方式完成总线传输：

- 立即完成传输
- 插入一个或多个等待状态，为完成传输提供足够的时间
- 发送错误响应来指示传输失败

这三种Slave传输响应在下面分别描述：

- 传输完成
- 传输挂起
- 错误响应

5.1.1传输完成

成功完成传输，通过置位**HREADY**为高电平并且**HRESP**状态为**OKAY**。

5.1.2 传输挂起

通常Slave使用**HREADY**信号来插入适当数量的等待状态在数据相位中。等到**HREADY**信号为高电平并且响应**OKAY**，指示成功完成了传输。

当Slave在完成响应之前插入若干等待状态，它必须驱动**HRESP**为**OKAY**。

注：通常Slave在它撤销占用总线之前，预先确定它要插入的等待状态数量. 这样就可以计算访问总线的延时.

建议Slave不要插入超过16拍等待时间，防止单一访问操作长期占用总线. 但是，在某些设备这个建议不适用，例如，boot ROM. 这种设备通常用来做系统启动，如果使用了超过16拍的等待状态对整个系统性能而言是微不足道的.

5.1.3 错误响应

Slave使用ERROR响应指示，相关联的传输的一些错误情况. 在试图写访问一个只读存储器时，这通常表示一周保护错误.

OKEY响应只需要一个时钟周期，ERROR响应需要两个. 发起一个ERROR响应，Slave驱动**HRESP**为高电平指示ERROR，同事驱动**HREADY**为低将传输扩展一个时钟周期. 在下一个时钟周期，**HREADY**驱动为高电平来结束传输，**HRESP**继续驱动为高电平来指示错误.

因为总线的流水线特性，所以要求这种两个时钟周期的响应机制. 当Slave开始发出一个ERROR响应的时候，下一个传输的地址已经发布在总线上，这种两周期响应机制为Master提供了足够的时间通过驱动**HTRANS[1:0]**为IDLE状态来取消下一拍总线访问.

如果 Slave 需要超过两个时钟周期的时间提供 ERROR 响应信号，则从传输开始需要额外插入等待状态. 此时，**HREADY** 为低电平，响应必须驱动为 OKAY.

图 5-1 所示，一个 ERROR 响应的传输：

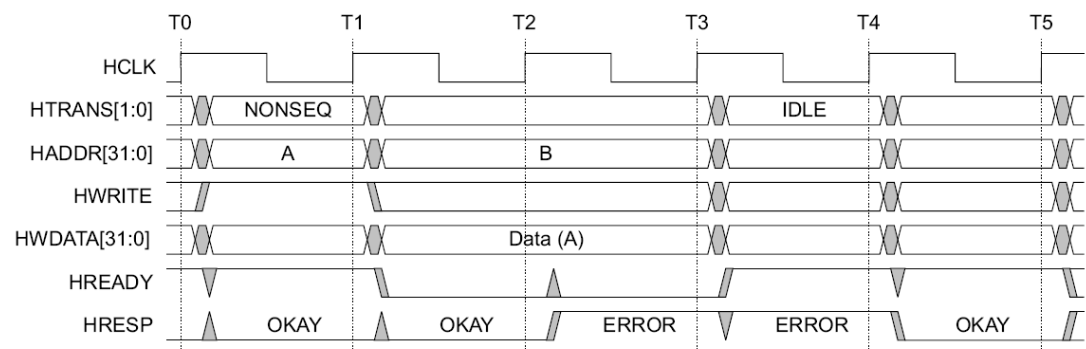


图5-1 ERROR response

- 在图 5-1 中：
- T1-T2     Slave插入了一个等待状态，提供了一个OKAY响应.
  - T2-T3     Slave发布一个ERROR响应. 这是ERROR响应两拍机制的第一拍，因为**HREADY**为低电平.
  - T3-T4     Slave发布一个ERROR响应. 这是ERROR响应两拍机制的最后一拍，因为**HREADY**为高电平.  
Master改变传输类型为IDLE. 取消了原计划对地址B的传输，这是由于Slave在T2是发布的错误.
  - T4-T5     Slave回应一个OKAY响应.

当 Slave 响应了 ERROR 响应，Master 可以取消掉一个 Burst 中剩余的传输. 然而，这并不是严格的要求，Master 仍然可以继续 Burst 的剩余传输.



## 第六章 数据总线

本章描述了AHB-Lite数据总线，包括以下小节：

- 6.1 数据总线
- 6.2 数据总线位宽

## 6.1 数据总线

分离写总线和读总线，是为了避免AHB-Lite系统实现中使用三态驱动。建议最小的数据总线位宽为32位，可以按照6.2 数据总线位宽小节修改总线位宽。

数据总线描述：

- 6.1.1 写总线
- 6.1.2 读总线
- 6.1.3 大小端

### 6.1.1 写总线

在写传输期间，Master驱动写总线。如果传输被扩展了，Master必须保持数据不变直到由HREADY信号为高电平指示传输完成。

传输的数据位宽小于总线宽度，例如，在32位总线上请求16位传输，Master必须将数据驱动到相应的字节lane。Slave也是从相应的字节lanes取数据。

表6-1和表6-2分别列出了小端和大端列出了32位总线的所有字节lanes。

哪个字节lanes有效，取决于系统的大小端。因为AHB-Lite没有指定具体的大端还是小端，所以在一个系统中所有的Master和Slave使用相同的大小端很重要。

### 6.1.2 读总线

在读传输期间，相应的Slave驱动读总线。如果Slave通过保持HREADY为低电平扩展了传输周期，Slave只需要在整个传输的最后一拍(即HREADY为高电平)提供有效数据。

传输的数据位宽小于总线宽度，Slave只需要提供数据到相应的字节lanes。Master从正确的字节lane选择数据。

Slave只需要当响应OKEY时提供有效的数据。ERROR响应不需要提供读数据。

表6-1列出了小端系统32位总线的字节lanes：

表6-1 32位小端模式下的有效位

传输大小	地址偏移	DATA[31:24]	DATA[23:16]	DATA[15:8]	DATA[7:0]
Word	0	Active	Active	Active	Active
Halfword	0	-	-	Active	Active
Halfword	2	Active	Active	-	-
Byte	0	-	-	-	Active
Byte	1	-	-	Active	-
Byte	2	-	Active	-	-
Byte	3	Active	-	-	-

表6-2列出了大端系统32位总线的字节lanes：

表6-2 大端模式下32位总线有效位

传输大小	地址偏移	DATA[31:24]	DATA[23:16]	DATA[15:8]	DATA[7:0]
Word	0	Active	Active	Active	Active

Halfword	0	Active	Active	-	-
Halfword	2	-	-	Active	Active
Byte	0	Active	-	-	-
Byte	1	-	Active	-	-
Byte	2	-	-	Active	-
Byte	3	-	-	-	Active

如果有需要，可以扩展32位数据总线到位宽更宽的总线实现。小于总线位宽的Burst传输，在burst的不同拍可能需要不同的Byte使能。

6.1.3 大小端

在同一个系统中，所有的模块必须使用相同的大小端，任何数据通路和桥接页必须使用相同的大小端。

动态切换大小端不支持，因为在绝大多数嵌入式系统中，支持动态切换大小端将引入很大的芯片冗余。

建议只是在宽范围应用设计支持大小端，或者使用配置引脚或者使用内置控制位来选择大小端。对于更多特定应用，固定的大小端为小端或者是大端，可以收获面积更小，功耗更低，性能更高的接口。

## 6.2 数据总线位宽

需要提高总线带宽而不提高时钟频率的方法是提高芯片上的数据线的位宽。提高了金属层数，大容量片上存储单元(嵌入式DRAM)的使用，是支持更宽片上数据总线应用的因素。

指定总线为固定位宽，意味着，更多情况下，在应用中总线的位宽不能再优化。因此，有一个方法可以使用灵活的总线位宽，仍然能保证在模块设计间高性能地切换。协议允许AHB-Lite数据总线位宽为8,16,32,64,128,256,512和1024位。建议最小用位宽为32位的总线。最大数据位宽为256位，这样可以满足几乎所有应用的需求。对于读传输和写传输，接收模块必须从总线上正确的字节 lane 选择数据。不必全Byte line 复制数据。

以下章节描述：

- 6.2.1 在宽总线上实现窄位宽的Slave
- 6.2.2 在窄总线上实现宽Slave
- 6.2.3 在宽总线实现Master

### 6.2.1 在宽总线上实现窄位宽的Slave

图6-1所示，一个原本设计为数据总线位宽32位的Slave模块，怎样转接到位宽64位的总线系统。只需要添加一些额外的逻辑，而不需要模块内的设计的改变，此技术适用于Hard macrocell。

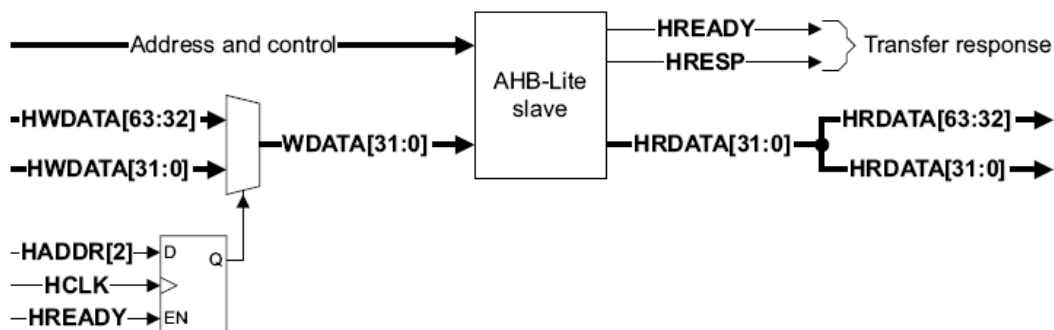


图6-1 宽总线上的窄Slave

对于输出，当转换窄的数据总线到宽总线时，做以下两件事：

- 复制数据到宽总线的每一半总线上，如图6-1所示
- 用附加逻辑保证总线上只有一半变化。这样将导致更多的功耗。

Slave只可以接收它本身总线宽度的数据传输。如果Master试图传输比Slave能支持的更宽的总线传输，Slave将回复Master ERROR响应。

### 6.2.2 在窄总线上实现宽Slave

可以通过采用额外的逻辑来预设计或引入Slave，使得可以在一个窄的总线系统工作。图6-2所示，在位宽窄的总线系统实现宽的Slave。

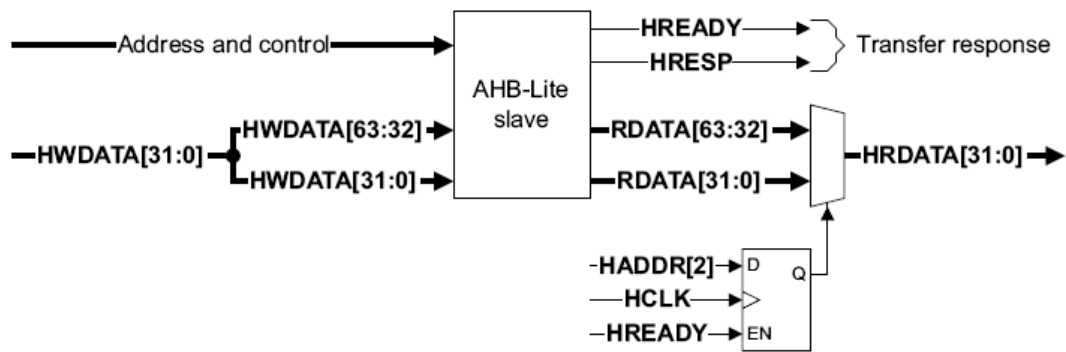


图6-2 窄总线上的宽Slave

### 6.2.3 在宽总线实现Master

可以按照在宽总线实现窄Slave的方法，在位宽更宽的总线上实现Master：

- 复用输入总线
- 复制输出总线

注：不可以让Master在窄总线上工作，除非在Master中包含了限制总线位宽的机制。Master禁止试图发起一个通过**H SIZE**指示的位宽比互连的总线实际位宽还要宽的传输。

## 第七章 时钟和复位

本章描述了协议的时钟和复位信号的时序. 包含以下章节:

- 7.1 时钟和复位需求

## 7.1 时钟和复位需求

本节描述了实现**HCLK**和**HRESETn**信号的需求.

### 7.1.1 时钟

每个AHB-Lite组件使用一个单独的时钟信号, **HCLK**. 所有的输入信号都在**HCLK**的上升沿采样. 所有的输出信号的改变都必须在**HCLK**的上升沿后发生.

### 7.1.2 复位

复位信号, **HRESETn**, 是AHB-Lite系统中仅有的低有效信号, 它是总线所有元素的初始复位. 复位 异步复位, 在**HCLK**上升沿同步撤销.

在复位期间, 所有的Master确保地址和控制信号都处于有效的电平值, 并且**HTRANS[1:0]**类型指示为IDLE.

在复位期间, 所有的Slave必须确保**HREADY**为高电平.

# 术语

本部分描述了ARM公司技术文档的一些短语.

内容略

Advanced eXtensible Interface

有些术语的翻译:

Master翻译为主设备

Slave翻译为从设备

Beat翻译为拍

翻译不好的地方:

a/an 翻译成一个是否合理

transfer, transaction, none/verb 都翻译为“传输”

一个完整的传输过程叫“传输”，一拍传输也叫传输？