

AMBA™ 总线规范

(V2.0)

ARM

AMBA 总线规范

(V2.0)

©版权归 ARM 公司所有，保留所有权利。

发行信息

变更历史

日期	事件	变更
1999.5.13	A	第一次发布

版权声明

ARM 以及 ARM 所支持的 logo, Thumb 和 StrongARM 是 ARM 有限公司注册的商标。

ARM 的 logo 包括 AMBA、PrimeCell、Angel、ARMulator、EmbeddedICE、ModelGen、Multi-ICE、ARM7TDMI、ARM7TDMI-S、ARM9TDMI、TDMI 和 STRONG 都是 ARM 有限公司的商标。

这里所提到的所有产品和服务也可能是他们各自所有者的商标。

在这篇文档中所包含的信息或者是描述的产品无论是其中的部分或者全部都不可以改编或者以任何方式复制除非在这之前有版权所有者书面许可之外。

本文中描述的产品是遵从（产品的）连续发展和改进的。ARM 有限公司友好的提供本文中包含的产品细节和它们的用途。然而，它们并不用于所有内在或者外在保证（授权），包含但不限于内在保证（授权），商业或其他目的的使用。

本文仅用于帮助读者使用本公司的产品。但是 ARM 有限公司并不对由于使用本文中的信息，或者这些信息中的错误和疏忽以及任何错误使用产品所造成的损失或者损害负责。

本文的机密状态

本文公开阅览，对于发行没有任何限制。

产品状态

本文中的信息是最终版本（关于已经开发出的产品的信息）。

ARM 网站地址

<http://www.arm.com>

前言

这篇前言介绍了高级微控制器总线体系（Advanced Microcontroller Bus Architecture AMBA）规范。它由下面两个部分组成：

- 关于本文的介绍；
- 反馈。

关于本文

本文是 AMBA 规范

适合的读者

本文用来帮助有经验的硬件或者软件工程师设计符合 AMBA 规范的模块。

文档结构

本文由下面的章节组成：

- | | |
|-----|---|
| 第一章 | AMBA 总线简介 |
| | 这一章给出了 AMBA 总线概括。 |
| 第二章 | AMBA 信号 |
| | 这一章描述了 AMBA 设备用到的信号。 |
| 第三章 | AMBA AHB |
| | 这一章介绍了 AMBA 高级高性能总线(Advanced High-performance Bus)。 |
| 第四章 | AMBA ASB |
| | 这一章介绍了 AMBA 高级系统总线 (Advanced System Bus)。 |
| 第五章 | AMBA APB |
| | 这一章介绍了 AMBA 高级外设总线 (Advanced Peripheral Bus)。 |
| 第六章 | AMBA 测试方法 |
| | 这一章介绍了 AMBA 总线中用到的测试方法。 |

排版惯例

以下是本文中使用的排版惯例：

- | | |
|-------------|---|
| type | 文本形式高亮显示的 ARM 处理器信号，以及诸如选项名称之类的接口元素。也可以用来强调合适的描述对象。 |
| <i>type</i> | 术语、交叉引用、引用的集合。 |
| type | 表示键盘输入的文本，比如命令、文件名或者程序名以及源代码。 |
| <u>type</u> | 表示一个允许的命令或者选项的缩写。带下划线的文本表示完整命令或者选项名称的缩写。 |
| type | 表示函数或者命令的参数，这些参数可以用特定值来替代。 |
| type | 表示代码样例中程序语言的关键字。 |

时序图惯例

本文中包含一种或者多种时序图。下面的样例给出了时序图中的成分。时序图中发生任何变化都有明显的标记。因此，除了特别说明之外这些元素中没有别的附加信息。



序图 1 时序图惯例

总线和信号中的阴影部分为未定义，所以总线和信号在阴影区域时间段时可以是任何值。这时真实值并不重要同时也不影响正常操作。

反馈

ARM 有限公司欢迎反馈 AMBA 和 AMBA 总线规范中的信息。

对本文档的反馈

如果你对本文有任何评论，请发邮件给errata@arm.com，并给出下列信息：

- 文档标题；
- 文档号；
- 你的评论中引用的页码；
- 对你评论的简单解释。

同时也欢迎一般的建议和改进方法。

对 AMBA 总线标准的反馈

如果你对本产品有任何评论或者建议，请联系你的供应商并给出如下信息：

- 产品名称；
- 对你评论的简单说明。

目 录

第 1 章 AMBA总线简介.....	1
1.1 AMBA总线概况.....	1
1.1.1 高级高性能总线（AHB）	1
1.1.2 高级系统总线（ASB）	1
1.1.3 高级外设总线（APB）	1
1.2 AMBA规范的目的.....	1
1.3 基于AMBA 的典型微控制器	2
1.4 术语.....	2
1.5 AMBA AHB介绍.....	3
1.6 AMBA ASB介绍	4
1.7 AMBA APB介绍	4
1.8 为系统选择合适的总线.....	5
1.8.1 系统总线的选择.....	5
1.8.2 系统总线和外设总线.....	5
1.8.3 什么时候使用AMBA AHB/ASB或者APB	5
1.9 AMBA规范备忘.....	5
1.9.1 工艺独立性.....	6
1.9.2 电气特征.....	6
1.9.3 时序规范.....	6
第 2 章 AMBA信号.....	7
2.1 AMBA信号名称.....	7
2.1.1 AHB信号前缀	7
2.1.2 ASB信号前缀	7
2.1.3 APB信号前缀	7
2.2 AMBA AHB信号列表.....	7
2.3 AMBA ASB信号列表	9
2.4 AMBA APB信号列表	10
第 3 章 AMBA AHB.....	12
3.1 关于AMBA AHB.....	12
3.1.1 基于AMBA AHB的典型微控制器	12
3.2 总线互联.....	13
3.3 AMBA AHB操作概括	14
3.4 基本传输.....	14
3.5 传输类型.....	17
3.6 突发操作.....	18
3.6.1 突发提前终止.....	19
3.7 控制信号.....	22
3.7.1 传输方向.....	22
3.7.2 传输大小.....	22
3.7.3 保护控制.....	23
3.8 地址译码.....	23
3.9 从机传输响应.....	24

3.9.1	传输完成.....	24
3.9.2	传输响应.....	24
3.9.3	双周期响应.....	25
3.9.4	错误响应.....	27
3.9.5	分块和重试.....	27
3.10	数据总线.....	27
3.10.1	HWDATA[31: 0].....	27
3.10.2	HRDATA[31: 0].....	28
3.10.3	端结构.....	29
3.11	仲裁.....	29
3.11.1	信号描述.....	29
3.11.2	请求总线访问.....	30
3.11.3	授予总线访问.....	30
3.11.4	突发提前终止.....	32
3.11.5	锁定传输.....	33
3.11.6	默认总线主机.....	33
3.12	分块传输.....	33
3.12.1	分块传输顺序.....	34
3.12.2	多重分块传输.....	34
3.12.3	预防死锁.....	34
3.12.4	分块传输的总线移交.....	35
3.13	复位.....	36
3.14	关于AHB数据总线的位宽.....	36
3.15	在宽总线上实现窄从机.....	37
3.16	在窄总线上实现宽从机.....	37
3.16.1	主机.....	38
3.17	关于AMBA AHB的组成部分.....	38
3.18	AHB总线从机.....	38
3.18.1	接口框图.....	38
3.18.2	时序图.....	39
3.18.3	时序参数.....	40
3.19	AHB总线主机.....	42
3.19.1	接口框图.....	42
3.19.2	总线主机时序.....	42
3.19.3	时序参数.....	43
3.20	AHB仲裁器.....	44
3.20.1	接口框图.....	44
3.20.2	时序图.....	45
3.20.3	时序参数.....	46
3.21	AHB译码器.....	47
3.21.1	接口框图.....	47
3.21.2	时序图.....	48
3.21.3	时序参数.....	48
第 4 章	AMBA ASB.....	49

4.1	关于AMBA ASB	49
4.1.1	基于AMBA ASB的典型微控制器.....	49
4.1.2	AMBA ASB和APB.....	50
4.2	AMBA ASB描述	50
4.3	ASB传输	51
4.3.1	非连续传输.....	51
4.3.2	连续传输.....	52
4.3.3	仅地址传输.....	53
4.4	地址译码.....	55
4.5	传输响应.....	56
4.6	多主机操作.....	58
4.6.1	仲裁器.....	58
4.6.2	总线主机移交.....	59
4.6.3	默认总线主机.....	60
4.6.4	锁定传输.....	60
4.7	复位操作.....	60
4.7.1	退出复位.....	60
4.8	ASB信号描述.....	61
4.8.1	时钟.....	61
4.8.2	复位.....	61
4.8.3	传输类型.....	62
4.8.4	地址和控制信息.....	63
4.8.5	地址总线.....	63
4.8.6	传输方向.....	63
4.8.7	传输大小.....	63
4.8.8	保护信息.....	64
4.8.9	地址和控制信号时序.....	64
4.8.10	地址和控制信号的三态使能.....	67
4.8.11	从机选择信号.....	68
4.8.12	传输响应.....	70
4.8.13	数据总线.....	73
4.8.14	仲裁信号.....	75
4.9	关于ASB AMBA的组成部分	76
4.10	ASB总线从机	76
4.10.1	接口框图.....	77
4.10.2	总线从机接口描述.....	77
4.10.3	时序图.....	78
4.10.4	时序参数.....	79
4.11	ASB总线主机	80
4.11.1	接口框图.....	80
4.11.2	总线主机接口描述.....	81
4.11.3	总线接口状态机.....	81
4.11.4	总线主机时序图.....	84
4.11.5	时序参数.....	87

4.12	ASB译码器.....	89
4.12.1	接口框图.....	89
4.12.2	译码器描述.....	90
4.12.3	时序图.....	93
4.12.4	时序参数.....	95
4.13	ASB仲裁器.....	95
4.13.1	接口框图.....	96
4.13.2	仲裁器描述.....	96
4.13.3	时序图.....	96
4.13.4	时序参数.....	97
第 5 章	AMBA APB	99
5.1	关于AMBA APB	99
5.1.1	基于AMBA的典型微控制器.....	99
5.2	APB规范.....	100
5.2.1	状态图.....	100
5.2.2	写传输.....	101
5.2.3	读传输.....	101
5.3	关于AMBA APB的组成部分.....	102
5.4	APB桥.....	102
5.4.1	接口框图.....	102
5.4.2	APB桥描述.....	103
5.4.3	时序图.....	103
5.4.4	时序参数.....	104
5.5	APB从机.....	105
5.5.1	接口框图.....	105
5.5.2	APB从机描述.....	105
5.5.3	时序图.....	106
5.5.4	时序参数.....	106
5.6	APB到AHB的接口.....	107
5.6.1	读传输.....	107
5.6.2	写传输.....	109
5.6.3	背靠背传输.....	110
5.6.4	三态数据总线的实现.....	110
5.7	APB到ASB的接口.....	111
5.7.1	写传输.....	111
5.7.2	读传输.....	112
5.8	APB D版外设到APB 2.0 版外设的接口	113
第 6 章	AMBA测试方法.....	115
6.1	关于AMBA测试接口.....	115
6.2	外部接口.....	116
6.2.1	测试总线请求A.....	116
6.2.2	测试总线请求B	116
6.2.3	测试应答.....	116
6.2.4	测试时钟.....	117

6.2.5	测试总线.....	117
6.3	测试向量类型.....	117
6.4	测试接口控制器.....	118
6.4.1	测试传输参数.....	118
6.4.2	增量地址.....	118
6.4.3	进入测试模式.....	118
6.4.4	地址向量.....	119
6.4.5	控制向量.....	119
6.4.6	写测试向量.....	119
6.4.7	读测试向量.....	119
6.4.8	突发向量.....	120
6.4.9	改变突发的方向.....	120
6.4.10	退出测试模式.....	120
6.5	AHB测试接口控制器	120
6.5.1	控制向量.....	122
6.6	AMBA AHB测试序列样例.....	123
6.6.1	进入测试模式.....	124
6.6.2	写测试向量.....	124
6.6.3	读传输.....	125
6.6.4	控制向量.....	126
6.6.5	突发向量.....	127
6.6.6	读到写和写到读.....	127
6.6.7	退出测试模式.....	128
6.7	ASB测试接口控制器.....	128
6.7.1	控制向量位定义.....	129
6.8	AMBA ASB测试序列样例	130
6.8.1	进入测试模式.....	130
6.8.2	地址向量.....	130
6.8.3	控制向量.....	131
6.8.4	写测试向量.....	133
6.8.5	改变突发方向.....	138
6.8.6	退出测试模式.....	139
附录A	索引.....	141
附录B	后记.....	147
附录C	说明.....	148

第1章 AMBA总线简介

这一章主要介绍了高级微控制器总线体系（AMBA）规范。包含了以下内容：

- AMBA总线概况；
- AMBA规范的目的；
- 基于AMBA 的典型微控制器；
- 术语；
- AMBA AHB介绍；
- AMBA ASB介绍；
- AMBA APB介绍；
- 为系统选择合适的总线；
- AMBA规范备忘。

1.1 AMBA总线概况

高级微控制器总线体系（AMBA）规范定义了在设计高性能嵌入式微控制器时的一种片上通信标准。

根据 AMBA 标准定义了三种不同的总线：

- 高级高性能总线（AHB）；
- 高级系统总线（ASB）；
- 高级外设总线（APB）。

AMBA 规范还包含一种测试方法以提供对宏单元进行测试和诊断访问的下部构造。

1.1.1 高级高性能总线（AHB）

AMBA AHB 是用于高性能、高时钟频率的系统模块。

AHB 担当高性能系统的中枢总线。AHB 支持处理器，片上存储器，片外存储器以及低功耗外设宏功能单元之间的有效连接。AHB 也通过使用综合和自动测试技术的有效设计流来确保减轻使用负担。

1.1.2 高级系统总线（ASB）

AMBA ASB 是用于高性能的系统模块之间的。

AMBA ASB 是另外一种系统总线，用在并不要求 AHB 的高性能特征的地方。ASB 也支持处理器，片上存储器，片外存储器以及低功耗外设宏功能单元之间的有效连接。

1.1.3 高级外设总线（APB）

AMBA APB 是用于低功耗外设的。

AMBA APB 优化了最小功率消耗并且降低了接口复杂度以支持外设功能。APB 可以用来连接任意一种版本的系统总线。

1.2 AMBA规范的目的

AMBA 规范设计用于满足以下四个关键要求：

- 促进带一个或多个 CPU 或者信号处理器的嵌入式微控制器产品的第一时间开发；
- 技术上独立并且高复用度的外设和系统宏单元能在多样的 IC 工序之间方便的移植，以及适用于完整定制、标准宏单元和门阵列技术；

- 鼓励标准系统设计以提高处理器的独立性，提供高级 cache CPU 的发展路线图和外设库的发展；
- 降低硅的下部构造要求以支持用在操作和生产测试时有效的片上和片外通信。

1.3 基于AMBA 的典型微控制器

基于AMBA的微控制器典型地由一个高性能系统中枢总线（AMBA AHB或者AMBA ASB）组成，能够支持外部存储器带宽，包括CPU，片上存储器和其他直接数据存取（DMA）设备。这条总线为上述单元之间大多数的传输提供高带宽接口。在这条高性能总线上也有一个桥接器以连接低带宽的APB，而在APB上连接着大多数的系统外设（参见图 1.1）。

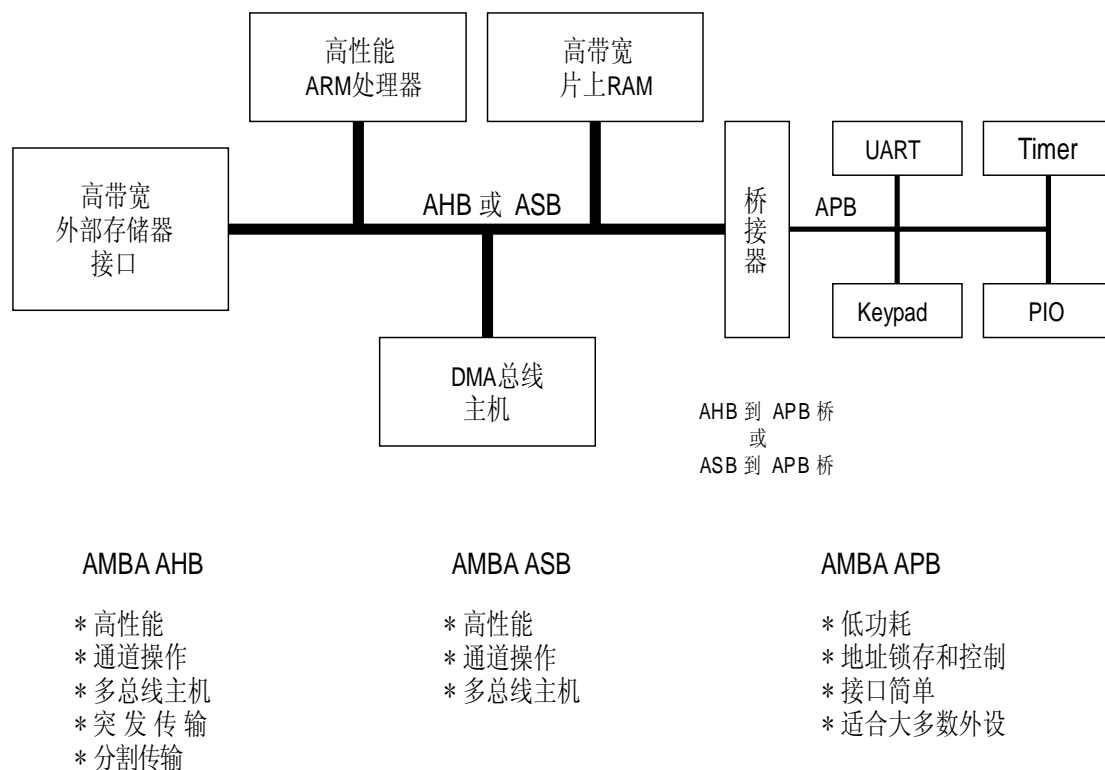


图 1.1 典型 AMBA 系统

AMBA APB 作为主总线高性能通道的二级总线，提供基本外设宏单元之间的通信接口。这些典型的外设具有以下特征：

- 有存储器映射的寄存器接口；
- 没有高带宽接口；
- 通过可编程的控制来访问。

外部存储器接口是受应用指定因而可能只有较窄的数据通道，但也可以支持测试访问模式以允许通过独立于系统的测试装备来测试内部 AMBA AHB、ASB 和 APB。

1.4 术语

下面的条目贯穿整个规范中都有用到。

总线周期 总线周期是总线时间的基本单元并且对于 AMBA AHB 或者 APB 协议的描述目的是定义为从上升沿到上升沿的转移。ASB 的总线周期定义为从下降沿到下降沿的转移。总线信号时序参考于总线周期时钟。

- 总线传输** AMBA AHB 或者 ASB 总线传输是对数据目标的读写操作,可能会持续一个或者多个总线周期。总线传输在收到从机地址的完成响应后终止。
- AMBA ASB 总线支持的传输大小包括字节 (8 位)、半字 (16 位)、字 (32 位)。AMBA AHB 又支持较宽的数据传输,包括 64 位和 128 位的数据传输。AMBA APB 总线传输是对数据目标的读写操作,总是需要 2 个总线周期。
- 突发操作** 突发操作定义为一个或多个数据传输,由总线主机发起,在地址空间增加时传输宽度保持一致。每次传输增加的 (地址) 步长由传输大小决定 (字节, 半字, 字), 对于 APB 而言不支持突发操作。

1.5 AMBA AHB 介绍

AHB 是为提出高性能可综合设计的要求而产生的新一代 AMBA 总线。它是一种支持多总线主机和提供高带宽操作的高性能总线。

AMBA AHB 实现了高性能, 高时钟频率系统的以下特征要求:

- 突发传输;
- 分块处理;
- 单周期总线主机移交;
- 单时钟沿操作;
- 非三态执行;
- 更宽的数据总线架构 (64 位或者 128 位)。

这条高级总线和当今的 ASB/APB 能够有效的桥接确保了能够方便集成任何现有的设计。

AMBA AHB 的设计可能包含一个或者多个主机, 一个典型的系统将至少包含处理器和测试接口。然而, 将直接数据存取 (DMA) 或者数字信号处理器 (DSP) 包含作总线主机也很普通。

外部存储器接口, APB 桥和任何内部存储器是最常见的 AHB 从机。然而, 低带宽的外设通常都是连接到 APB 上。

典型的 AMBA AHB 系统设计包含以下的成分:

AHB 主机 总线主机能够通过提供地址和控制信息发起读写操作。任何时候只允许一个总线主机处于有效状态并能使用总线。

AHB 从机 总线从机在给定的地址空间范围内响应读写操作。总线从机将成功、失败或者等待数据传输的信号返回给有效的主机。

AHB 仲裁器 总线仲裁器确保每次只有一个总线主机被允许发起数据传输。即使仲裁协议已经固定, 任何一种仲裁算法, 比如最高优先级或者公平访问都能够根据应用要求而得到执行。

AHB 必须只包含一个仲裁器, 尽管在单总线主机系统中这显得并不重要。

AHB 译码器 AHB 译码器用来对每次传输进行地址译码并且在传输中包含一个从机选择信号。

所有 AHB 执行都必须仅要求有一个中央译码器。

1.6 AMBA ASB介绍

ASB 是第一代 AMBA 系统总线。ASB 位于当前的 APB 之上并且实现高性能系统的以下要求：

- 突发传输；
- 通道传输操作；
- 多总线主机。

典型的 AMBA ASB 系统包括一个或者多个总线主机。例如，至少有处理器和测试接口。然而，将直接数据存取（DMA）或者数字信号处理器（DSP）包含作总线主机也很普通。

外部存储器接口，APB 桥和任何内部存储器是最常见的 ASB 从机。然而，低带宽的外设通常都连接到 APB 上。

AMBA ASB 系统设计典型的包括以下成分：

ASB 主机 总线主机能够通过提供地址和控制信息发起读写操作。任何时候只允许一个总线主机处于有效状态并能使用总线。

ASB 从机 总线从机在给定的地址空间范围内响应读写操作。总线从机将成功、失败或者等待数据传输的信号返回给有效的主机。

ASB 译码器 总线译码器将传输地址译码并且选择合适的从机。总线译码器也确保总线在没有总线传输时也保持运作。所有 ASB 工具都必须要求仅有一个中央译码器。

ASB 仲裁器 总线仲裁器确保每次只有一个总线主机被允许发起数据传输。即使仲裁协议已经固定，任何一种仲裁算法，比如最高优先级或者公平访问都能够根据应用要求而得到执行。

ASB 必须只包含一个仲裁器，尽管在单总线主机系统中这显得并不重要。

1.7 AMBA APB介绍

APB 是 AMBA 总线体系中的一部分并且优化了最小功率消耗和降低了接口复杂度。

AMBA APB 表现为一个局部二级总线，封装为 AHB 或者 ASB 的一个外设。APB 在 AHB 和 ASB 信号的基础上直接为系统总线提供了低功耗的延伸。

APB 桥作为一个从模块处理总线握手并且从局部外设总线的角度控制信号的时序更新。通过从系统总线起始点定义 APB 接口，可以有效的利用系统诊断和测试方法的好处。

AMBA APB 应该被用作任何低带宽和不需要通道总线接口的高性能的外设接口。

最新版本的 APB 规定所有信号的传输只和时钟的上升沿相关。这个改进确保了 APB 能够很容易地被集成到任何设计流中，并包含以下优点：

- 易于实现较高频率的操作；
- 性能和时钟的脉冲间隔比（占空比）无关；
- 通过使用单时钟沿来简化静态时序分析；
- 对自动测试插入无需考虑特殊情况；
- 许多专用集成电路（ASIC）库对上升沿寄存器可以优先选择；
- 易于集成到基于周期的仿真器中。

APB 的改变也使得它和新版 AHB 的接口变得更简单。

典型的 AMBA APB 工具包含单个 APB 桥，需要它来将 AHB 或者 ASB 传输转变成适合于 APB 从设备的形式。APB 桥提供所有地址、数据和控制信号的锁存，也提供一个二级译码以产生 APB 外设的从机选择信号。

所有 APB 模块均是 APB 从机。APB 从机包含以下接口规格：

- 整个访问中地址和控制信号有效（不分通道）；
- 当无外设总线活动时接口功耗为零（外设总线不使用时为静态）；
- 通过选通脉冲时序译码产生时序（无时钟接口）；
- 整个访问过程中写数据有效（允许无毛刺透明锁存工具）。

1.8 为系统选择合适的总线

在决定系统应该使用哪种总线之前，应该考虑下面的因素：

- 系统总线的选择；
- 系统总线和外设总线；
- 什么时候使用 AMBA AHB/ASB 或者 APB。

1.8.1 系统总线的选择

AMBA AHB 和 ASB 都可以作为主要系统的总线。典型的系统总线选择应该取决于系统模块的接口要求。

在新设计中建议使用 AHB，不仅是因为它提供了更高的带宽解决方案，也因为单时钟边沿协议使得在典型 ASIC 开发时使用的自动化工具更容易综合。

1.8.2 系统总线和外设总线

将所有外设整合成 AHB 或者 ASB 完整的功能模块是可行的但不可能总是令人满意的：

- 有大量外设宏单元的设计中，增加的总线负载会导致功耗增加和性能的下降；
- 当需要进行时序分析时，总线上（速度）最慢的单元会限制最大性能；
- 很多简单的外设宏单元仍需要锁存地址和控制信号尽管有悖于从通道信号传输中受益的高带宽宏单元；
- 很多外设单元仅只需要一个简单的选择脉冲来传递宏单元的选择信号和总线读/写操作，没有广播高频时钟信号到每个外设的要求。

1.8.3 什么时候使用AMBA AHB/ASB或者APB

完整的 AHB/ASB 用作：

- 总线主机；
- 片上存储模块；
- 片外存储器接口；
- 带 FIFO 接口的高带宽外设；
- DMA 从机外设。

简单的 APB 外设接口建议用作：

- 简单的寄存器映射从机设备；
- 时钟不能够全局布通的超低功耗接口；
- 分组窄总线外设应避免挂接到系统总线上。

1.9 AMBA规范备忘

以下是阅读 AMBA 规范时应该考虑的几点：

- 工艺独立性；
- 电气特征；
- 时序规范。

1.9.1 工艺独立性

AMBA 是一种工艺独立（与工艺无关）的片上协议。本规范仅详述了总线协议和时钟周期级别。

1.9.2 电气特征

AMBA 规范并不提供任何电气特征有关的信息，因为这完全取决于设计时选择的生产处理工艺。

1.9.3 时序规范

AMBA 协议在周期级别上定义了多种信号的行为。准确的时序要求取决于所使用的处理工艺和操作频率。

因为在 AMBA 协议中并未定义精确的时序要求，这就给了系统综合者对总线上的多种模块分配信号时序预算时最大的自由度。

第2章 AMBA信号

这一章介绍了 AMBA 信号。包括以下部分：

- AMBA信号名称；
- AMBA AHB信号列表；
- AMBA ASB信号列表；
- AMBA APB信号列表。

2.1 AMBA信号名称

所有 AMBA 信号的命名都用名称的第一个字母来指示信号和哪个总线相关联。信号名称中用一个小写的 **n** 表示该信号低电平有效，否则信号的名称总是用大写字母来表示。

测试信号有一个前缀**T**而与总线类型无关。关于测试信号的更多信息请参考第六章 AMBA测试方法。

2.1.1 AHB信号前缀

H 表示一个 AHB 信号。

例如，**HREADY** 是用来指示 AHB 部分数据传输完毕的信号。该信号高电平有效。

2.1.2 ASB信号前缀

A ASB 主机和仲裁器之间的单向信号。

B 表示一个 ASB 信号。

D 一个单向 ASB 译码器信号。

例如，**BnRES** 为 ASB 复位信号，低电平有效。

2.1.3 APB信号前缀

P 表示一个 APB 信号。

例如，**PCLK** 是 APB 使用的主时钟。

2.2 AMBA AHB信号列表

这一节包括了AMBA AHB信号的概括（表 2.1）。每个信号的完整描述可以本文档的后继章节中找到。

所有信号都以字母前缀 **H** 开头，在系统设计中要确保 AHB 信号不同于其他相似命名的信号。

表 2.1 AMBA AHB 信号

名称	来源	描述
HCLK 总线时钟	时钟源	时钟为所有总线传输提供时基。所有信号时序都和 HCLK 的上升沿相关。
HRESETn 复位	复位控制器	总线复位信号低有效并用来复位系统和总线。这是唯一的低有效的信号。

续上表

名称	来源	描述
HADDR[31: 0] 地址总线	主机	32 位系统地址总线。
HTRANS[1: 0] 传输类型	主机	表示当前传输的类型，可以是不连续、连续、空闲和忙。
HWRITE 传输方向	主机	当该信号为高时表示一个写传输，为低的时候表示一个读传输。
HSIZE[2: 0] 传输大小	主机	表示传输的大小，典型情况是字节（8 位）、半字（16 位）或者是字（32 位）。协议允许最大的传输大小可以达到 1024 位。
HBURST[2: 0] 突发类型	主机	表示传输是否组成了突发的一部分。支持四个、八个或者 16 个节拍的突发传输并且突发传输可以是增量或者是回环。
HPROT[3: 0] 保护控制	主机	提供总线访问的附加信息并且主要是打算给那些希望执行某种保护级别的模块使用的。这个信号指示当前传输是否为预取指或者数据传输，同时也表示传输是保护模式访问还是用户模式访问。对带存储器管理单元的总线主机而言这些信号也用来指示当前传输是高速缓存的（cache）还是缓冲的（buffer）。
HWDATA[31: 0] 写数据总线	主机	写数据总线用来在写操作期间从主机到总线从机传输数据。建议最小的数据总线宽度为 32 位。然而，在要求高带宽运行时扩展（数据总线）还是很容易的。
HSELx 从机选择	译码器	每个 AHB 从机都有自己独立的从机选择信号并且用该信号来表示当前传输是否是打算送给选中的从机。该信号是地址总线的简单组合译码。
HRDATA[31: 0] 读数据总线	从机	读数据总线用来在读操作期间从总线从机向总线主机传输数据。建议最小的数据总线宽度为 32 位。然而，在要求高带宽运行时扩展（数据总线）还是很容易的。
HREADY 传输完成	从机	当 HREADY 为高时表示总线上的传输已经完成。在扩展传输时该信号可能会被拉低。 注意：总线上的从机要求 HREADY 作为输入输出信号。
HRESP[1: 0] 传输响应	从机	传输响应给传输状态提供了附加信息。提供四种不同的响应：OKEY、ERROR、RETRY 和 SPLIT。

AMBA AHB 也有许多信号请求支持多主机操作（表 2.2）。许多这些仲裁信号用于点对点连接，表 2.2 中后缀 **x** 用来表示信号来自模块 **x**。例如，一个系统中会有许多的信号 **HBUSREQx**，比如 **HBUSREQarm**、**HBUSREQdma** 和 **HBUSREQtic**。

表 2.2 仲裁信号

名称	来源	描述
HBUSREQx 总线请求	主机	从总线主机 x 传向总线仲裁器用来表示该主机请求(控制) 总线的信号。系统中每个总线主机都有一个 HBUSREQx 信号, 最多 16 个总线主机。
HLOCKx 锁定的传输	主机	当该信号为高时表示主机请求锁定对总线的访问并且在信号为低之前其他主机不应该被允许授予总线。
HGRANTx 总线授予	仲裁器	该信号用来表示总线主机 x 目前是优先级最高的主机。当 HREADY 为高时传输结束, 地址/控制信号的所有权发生改变。所以主机应在 HREADY 和 HGRANTx 都为高时获得对总线的访问。
HMASTER[3: 0] 主机号	仲裁器	这些来自仲裁器的信号表示哪个总线主机正在执行传输和被支持分块传输的从机用来确定哪个主机正在尝试一次访问。 HMASTER 的时序和地址以及控制信号对齐。
HMASTLOCK 锁定顺序	仲裁器	表示当前主机正在执行一个锁定顺序的传输。该信号和 HMASTER 有相同的时序。
HSPLITx[15: 0] 分块完成请求	从机 (支持分块)	从机用这 16 位的分块总线来指示仲裁器总线主机应该被允许重试一个分块传输。 分块总线上的每一位对应一个总线主机。

2.3 AMBA ASB信号列表

表 2.3是AMBA ASB信号列表。

表 2.3 AMBA ASB 信号

名称	描述
AGNTx 总线授予	从总线仲裁器到总线主机 x 用来指示总线主机将在 BWAIT 为低时被授予总线的信号。和相关的总线请求信号 AREQx 一样, 系统中每个总线主机都有一个 AGNTx 信号。
AREQx 总线请求	从总线主机 x 到仲裁器用来表示总线主机请求总线的信号。和相关的总线授予信号 AGNTx 一样, 系统中每个总线主机都有一个 AREQx 信号。
BA[31: 0] 地址总线	系统地址总线, 由有效的总线主机驱动。
BCLK 总线时钟	时钟信号为所有总线传输提供时基。在总线上 BCLK 的高相位和低相位都用来控制传输。
BD[31: 0] 数据总线	这是双向的系统数据总线。数据总线在写传输时由当前的总线主机驱动而在读传输时由被选中的总线从机驱动。

续上表

名称	描述
BERROR 错误响应	被选中的总线从机使用 BERROR 信号来表示一个传输错误。当 BERROR 为高时发生一个传输错误，当 BERROR 为低时传输成功。该信号也和 BLAST 信号一起组合使用以表示一个总线撤消操作。当没有从机被选中时该信号由总线译码器驱动。
BLAST 末尾响应	该信号由被选中的总线从机驱动以表示当前传输是否是一次突发顺序的最后一次传输。当 BLAST 为高时译码器必须允许有足够的时间来进行地址译码。当 BLAST 为低时，下一次传输可能会继续一次突发顺序。这个信号也可以和 BERROR 信号一起组合使用以表示一个总线撤消操作。当没有从机被选中时该信号由总线译码器驱动。
BLOK 锁定传输	当该信号为高时表示当前传输和下一个传输是不可分割的并且其他总线主机不应该被授权访问总线。这个信号由总线仲裁器使用。该信号由有效的总线主机驱动。
BnRES 复位	总线复位信号为低有效并且用来复位系统和总线。这是唯一的低有效信号。
BPORT[1: 0] 保护控制	保护控制信号提供关于总线访问的附加信息并且主要打算是被总线译码器用作一个基本的保护单元。该信号表示传输是预取指还是数据访问，也表示传输是保护模式访问还是用户模式访问。该信号由有效的总线主机驱动并且必须和地址总线时序一致。
BSIZE[1: 0] 传输大小	传输大小信号指示传输的大小，可以是字节、半字或者字。该信号由有效的总线主机驱动并且必须和地址总线时序一致。
BTRAN[1: 0] 传输类型	这些信号表示下一次传输的类型，可以是仅地址、非连续或者连续。这些信号在适当的 AGNTx 信号被断言时由总线主机驱动。
BWAIT 等待响应	该信号由选中的总线从机驱动以表示当前传输是否完成。如果 BWAIT 为高时要求更多的总线周期，如果 BWAIT 为低时则传输可以在当前总线周期完成。当没有从机被选中时该信号由总线译码器驱动。
BWRITE 传输方向	该信号为高表示一个写传输而为低时表示一个读传输。这个信号由有效的总线主机驱动并且和地址总线有相同的时序。
DSELx 从机选择	从总线译码器到总线从机 x 用来表示该从机设备被选中并且要求一个数据传输的信号。每个 ASB 总线从机都有一个 DSELx 信号。

2.4 AMBA APB信号列表

所有的 AMBA APB 信号使用单个字母 **P** 作为前缀。有些 APB 信号，例如时钟，可能直接连接到系统总线上的对应信号。

表 2.4列出了AMBA APB信号名称，同时还描述了每个信号是如何使用的。

表 2.4 AMBA APB 信号

名称	描述
PCLK 总线时钟	PCLK 的上升沿用作所有 APB 传输的时基。
PRESETn APB 复位	APB 总线复位信号为低有效并且通常将该信号直接连接到系统总线复位信号。
PADDR[31: 0] APB 地址总线	这是 APB 地址总线，可高达 32 位宽度并由外设总线桥接单元驱动。
PSELx APB 选择	来自二级译码器的信号，从外设总线桥接单元内到每个外设总线从机 x 。该信号表示从机设备被选中并且要求一次数据传输。每个总线从机都有一个 PSELx 信号。
PENABLE APB 选通	这个选通信号用来给外设总线上的所有访问提供时间。使能信号用来表示一次 APB 传输的第二个周期。 PENABLE 的上升沿出现在 APB 传输的中间。
PWRITE APB 传输方向	该信号为高表示一次 APB 写访问而为低表示一次读访问。
PRDATA APB 读数据总线	读数据总线由被选中的从机在读周期（ PWRITE 为低）期间驱动。读数据总线可达到 32 位宽度。
PWDATA APB 写数据总线	写数据总线由外设总线桥接单元在写周期（ PWRITE 为高）期间驱动。写数据总线可达到 32 位宽度。

第3章 AMBA AHB

这一章介绍高级高性能总线体系（AHB），包含以下部分：

- 关于AMBA AHB；
- 总线互联；
- AMBA AHB操作概括；
- 基本传输；
- 传输类型；
- 突发操作；
- 控制信号；
- 地址译码；
- 从机传输响应；
- 数据总线；
- 仲裁；
- 分块传输；
- 复位；
- 关于AHB数据总线的位宽；
- 在宽总线上实现窄从机；
- 在窄总线上实现宽从机；
- 关于AMBA AHB的组成部分；
- AHB总线从机；
- AHB总线主机；
- AHB仲裁器；
- AHB译码器。

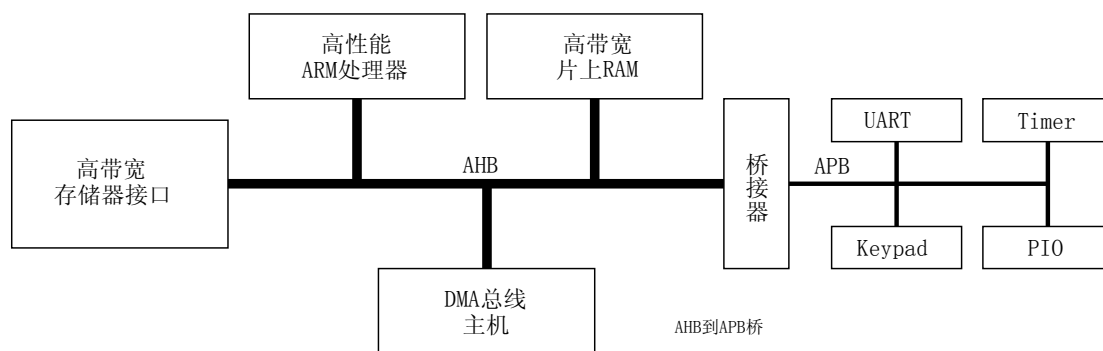
3.1 关于AMBA AHB

AHB 是为提出高性能可综合设计的要求而产生的新一代 AMBA 总线。AMBA AHB 是位于 APB 之上新一级的总线并且实现了高性能、高系统时钟频率系统的以下特征要求：

- 突发传输；
- 分块处理；
- 单周期总线主机移交；
- 单时钟沿操作；
- 非三态执行；
- 更宽的数据总线架构（64 位或者 128 位）。

3.1.1 基于AMBA AHB的典型微控制器

基于AMBA的微控制器典型地由一个高性能系统中枢总线组成，能够支持外部存储器带宽，包括CPU，片上存储器和其他直接数据存取（DMA）设备，加上一个桥接器连接到较窄的挂接着较低带宽外设的APB总线。图 3.1表示了包含AHB和APB的一个典型系统。



AMBA 高级高性能总线 (AHB)

- * 高性能
- * 通道操作
- * 突发传输
- * 多总线主机
- * 分块传输

AMBA 高级外设总线 (APB)

- * 低功耗
- * 锁存地址和控制
- * 接口简单
- * 适合大多数外设

图 3.1 典型基于 AMBA AHB 的系统

3.2 总线互联

AMBA AHB 总线协议设计使用一个中央多路选择器互联方案。使用该方案所有总线主机设备输出地址和控制信号来指示它们想执行的传输同时仲裁器决定哪一个主机能够将它地址和控制信号连通到所有的从机。当然也需要一个译码器来控制读数据和响应多路信号选择器，多路信号选择器选中来自传输中所包含从机的适当信号。

图 3.2 表示了实现包含三个主机和四个从机的 AMBA AHB 设计的结构要求。

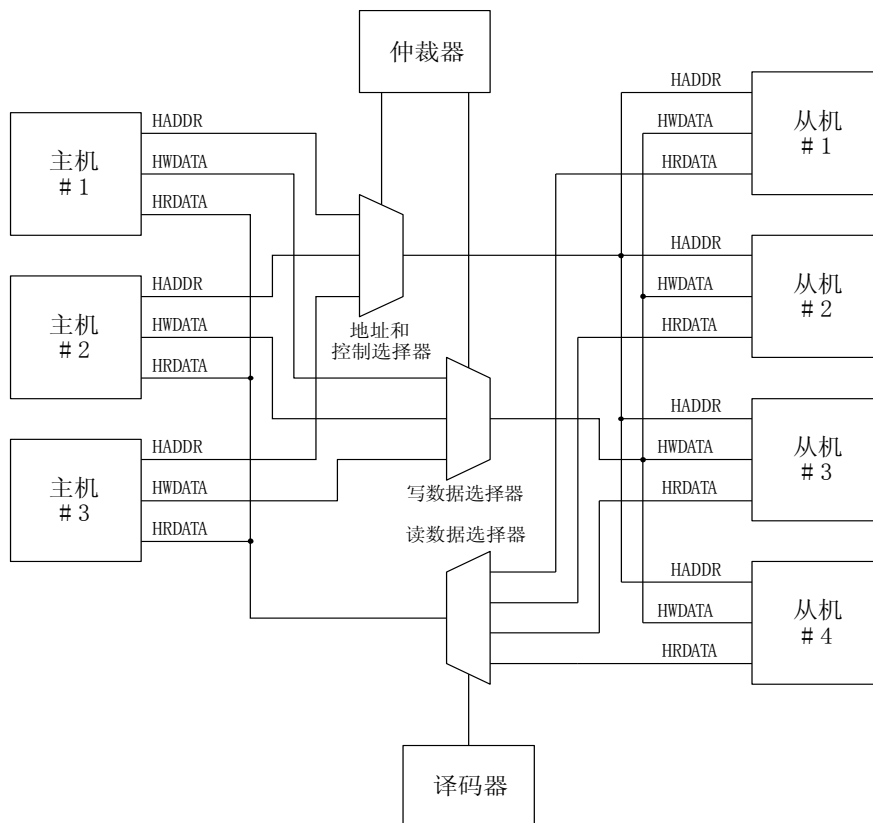


图 3.2 多路选择器互联

3.3 AMBA AHB操作概括

在一次 AMBA AHB 传输开始之前总线主机必须被授予访问总线。这个过程开始于总线主机向仲裁器断言一个请求信号。仲裁器指示主机何时能够被授予使用总线。被授权的总线主机通过驱动地址和控制信号来发起一次 AMBA AHB 传输。这些信号提供关于地址、方向和传输宽度的信息，以及表示传输类型是否为一次突发传输的部分。允许有两种不同类型的突发传输：

- 增量突发，在地址边界处不回环；
- 回环突发，在特定的地址边界上回环。

写数据总线用来将数据从主机传输到从机上，而读数据总线用来将数据从从机传输到主机上。

每次传输包含：

- 一个地址和控制周期；
- 一个或多个数据周期。

地址不长期有效所以所有从机必须在这个时段（传输地址时）采样地址。然而，通过 **HREADY** 信号可以延长数据。当该信号为低时导致在传输中插入等待状态同时允许从机有额外的时间提供或者采样数据。

在传输中从机通过使用响应信号来表示状态，**HRESP[1: 0]**：

OKAY	OKAY 响应用来表示传输进展正常并且当 HREADY 变高时表示传输成功完成。
ERROR	ERROR 响应表示发生了一个传输错误并且传输失败。
RETRY 和 SPLIT	RETRY 和 SPLIT 两个传输响应都表示传输不能立刻完成，但是总线主机应该继续尝试传输。

在常规操作中主机被允许在仲裁器授予另一个主机访问总线之前完成一个特定突发的所有传输。然而，为了避免过多的仲裁延时可能允许仲裁器打断一个突发并且这种情况下主机必须（申请）重新仲裁总线以完成剩下的突发传输。

3.4 基本传输

AHB 传输包含两个截然不同的部分：

- 地址相位，只持续单个周期；
- 数据相位，可能需要多个周期。这通过使用 **HREADY** 信号实现。

图 3.3表示了最简单的传输，没有等待状态。

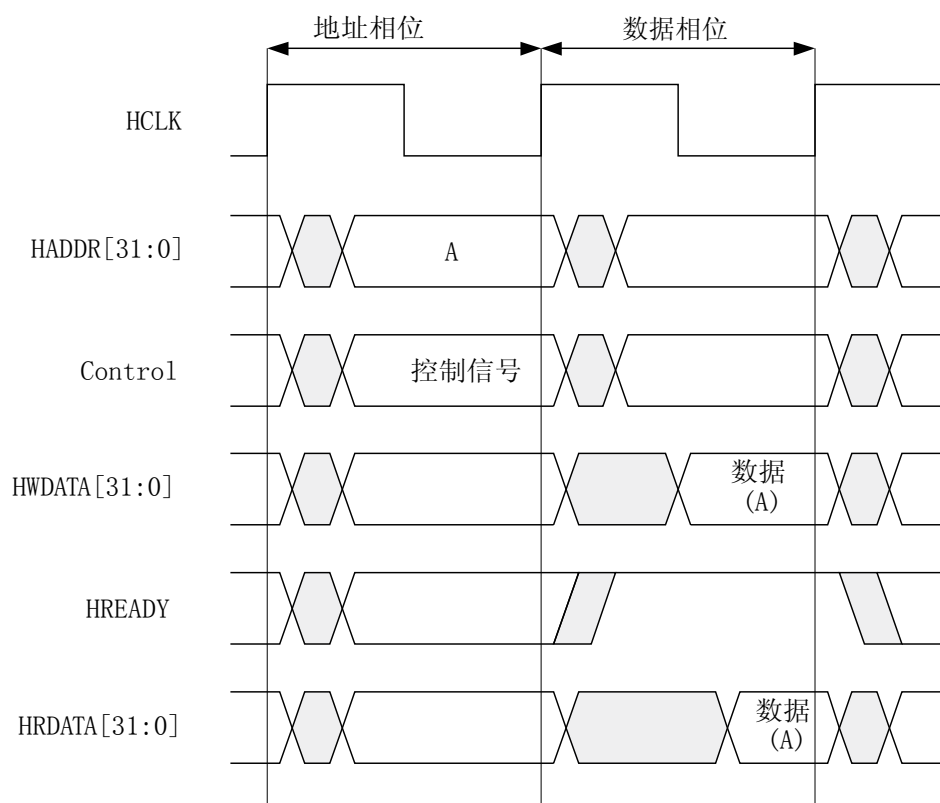


图 3.3 简单传输

在这个没有等待状态的简单传输中：

- 主机在 **HCLK** 的上升沿之后将地址和控制信号驱动到总线上；
- 然后在时钟的下一个上升沿从机采样地址和控制信息；
- 在从机采样了地址和控制信号后能够开始驱动适当的响应并且该响应被总线主机在第三个时钟的上升沿采样。

这个简单的例子演示了在不同的时钟阶段传输的地址和数据相位是如何产生的。事实上，任何传输的地址相位在前一次传输的数据相位期间出现。这种地址和数据的交叠是总线传输通道的基本性质同时允许高性能的操作。同时仍然给从机提供足够的时间来产生传输响应。

从机也可能插入等待周期到任意传输中，如图 3.4所示，这样扩展了传输完成允许的附加时间

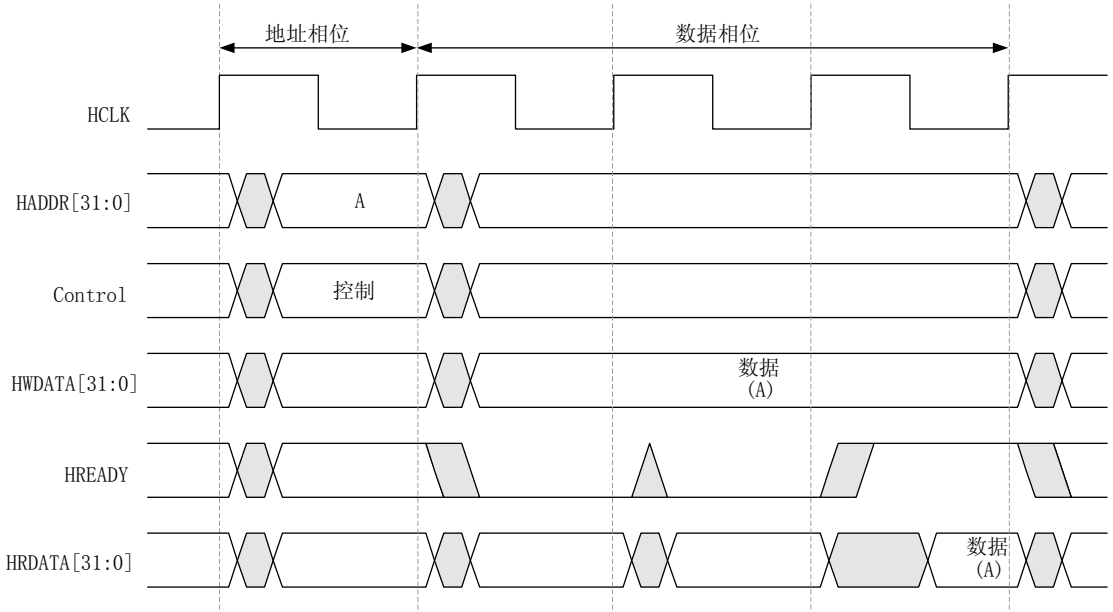


图 3.4 等待状态的传输

注：
对写操作而言总线主机必须保持数据在整个扩展周期中稳定。
对读传输从机没必要提供有效数据直到传输将要结束时。

当传输以这种方式扩展时随后的传输将会有地址相位扩展的副作用。如图 3.5所示，表示了三次传输到无关的地址A、B和C。

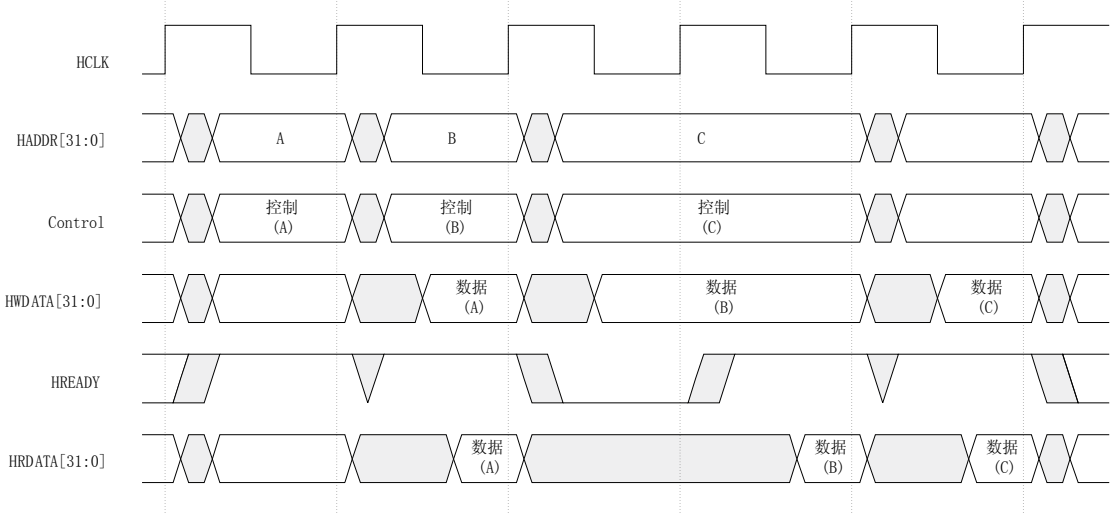


图 3.5 多重传输

- 在图 3.5中：
- 到地址 A 和 C 的传输都是零等待状态；
 - 到地址 B 的传输是一个等待状态；
 - 传输到地址 B 的扩展数据相位对传输到地址 C 的扩展地址相位有影响。

3.5 传输类型

每个传输都可以被分类为四个不同类型之一，如**HTRANS[1: 0]**信号所表示的状态，详见表 3.1:

表 3.1 传输类型编码

HTRANS[1: 0]	类型	描述
00	空闲 (IDLE)	表示没有数据传输的要求。空闲传输类型在总线主机被授予总线，但并不希望执行一个数据传输时使用。 从机必须总是提供一个零等待状态 OKAY 来响应空闲传输并且该传输应该被从机忽略。
01	忙 (BUSY)	忙传输类型允许总线主机在突发传输中间插入空闲周期。这种传输类型表示总线主机正在连续执行一个突发传输，但是下一次传输不能立即发生。当一个主机使用忙传输类型时地址和控制信号必须反映突发中的下一次传输。 这种传输应该被从机忽略。从机必须总是提供一个零等待状态 OKAY 响应，和从机响应空闲传输一样的方式。
10	非连续 (NONSEQ)	表示一次突发的第一个传输或者一个单一传输。地址和控制信号和前一次传输无关。 总线上的单一传输被视为（数据量为）一个的突发因此传输类型是非连续的。
11	连续 (SEQ)	突发中剩下的传输是连续传输并且地址是和前一次传输有关的。控制信息和前一次传输一样。地址等于前一次传输的地址加上传输大小（字节）。在回环突发的情况下传输地址在地址边界处回环，回环值等于传输大小乘以传输的次数（4、8 或者 16 其中之一）。 (译者注：这里回环指从传输的结束地址边界跳回到传输的起始地址边界。)

图 3.6表示了一组用到的不同传输类型:

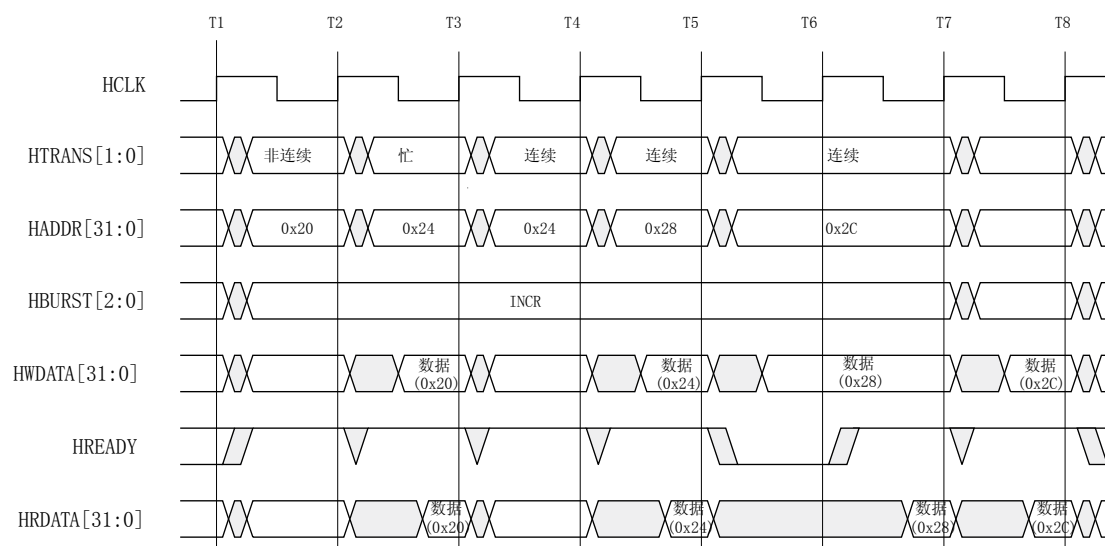


图 3.6 传输类型实例

在图 3.6中：

- 第一个传输是一次突发的开始所以传输类型为非连续；
- 主机不能立刻执行突发的第二次传输所以主机使用了忙传输来延时下一次传输的开始。在这个例子中主机在它准备还突发的下一次传输之前仅请求了一个忙周期，下一次传输的完成没有等状态；
- 主机立刻执行突发的第三次传输，但是这时从机不能完成（传输）并用 **HREADY** 来插入一个等待状态；
- 突发的最后一个传输以无等待状态完成。

3.6 突发操作

AMBA AHB 协议定义了四、八和十六拍突发，也有未定长度的突发和信号传输。协议对增量和回环都支持：

- 增量突发访问连续地址并且突发中的每次传输地址仅是前一次地址的一个增量；
- 对于回环突发，如果传输的起始地址并未和突发（x 拍）中字节总数对齐那么突发传输地址将在达到边界处回环。例如，一个四拍回环突发的字（4 字节）访问将在 16 字节边界回环。因此，如果传输的起始地址是 0x34，那么它将包含四个到地址 0x34、0x38、0x3C 和 0x30；

突发信息通过使用 **HBURST[2: 0]** 并且 8 中可能的类型在中定义如下：

表 3.2 突发信号编码

HBURST[2: 0]	类型	描述
000	SINGLE	单一传输
001	INCR	未指定长度的增量突发
010	WRAP4	4 拍回环突发
011	INCR4	4 拍增量突发

续上表

HBURST[2: 0]	类型	描述
100	WRAP8	8 拍回环突发
101	INCR8	8 拍增量突发
110	WRAP16	16 拍回环突发
111	INCR16	16 拍增量突发

突发禁止超过 1KB 的地址边界。因此重要的是主机不要尝试发起一个将要超过这个边界的定长增量突发。

将执行单个传输时使用未指定长度的增量突发理解为长度为一的突发比较合理。

一个增量突发可以是任何长度，但是（长度）上限由地址不能超过 1KB 边界这个事实限定了。

注：突发大小表示突发的节拍数量，并不是一次突发传输的实际字节数量。一次突发传输的数据总量可以用节拍数乘以每拍数据的字节数来计算，每拍字节数由 HSIZE[2: 0] 指示。

所有突发传输必须将地址边界和传输大小对齐。例如，字传输必须对齐到字地址边界（也就是 A[1: 0] = 00），半字传输必须对齐到半字地址边界（也就是 A[0] = 0）。

3.6.1 突发提前终止

当一个突发不允许完成的特定情况下对任一从机设计而言如果突发提前终止那么利用突发信息能够采取正确的动作显得很重要。从机能够通过监控 HTRANS 信号决定一个突发何时提前终止并且确保在突发开始之后每次传输有连续或者忙的标记。如果产生一个非连续或者空闲传输那么这表明一个新的突发已经开始因此前一次突发一定已经终止。

如果总线主机因为失去对总线的占有而不能完成一次突发那么它必须在下一次获取访问总线时正确地重建突发。例如，如果一个主机仅完成了一个四拍突发的一拍那么它必须用一个未定长度突发来执行剩下的三拍突发。

以下是一些例子：

- 图 3.7 表示了一个四拍回环突发；
- 图 3.8 表示了一个四拍增量突发；
- 图 3.9 表示了一个八拍回环突发；
- 图 3.10 表示了一个八拍增量突发；
- 图 3.11 表示了一个未定长度的突发。

图 3.7 表示了一个四拍回环突发并且第一次传输伴随一个附加等待状态。

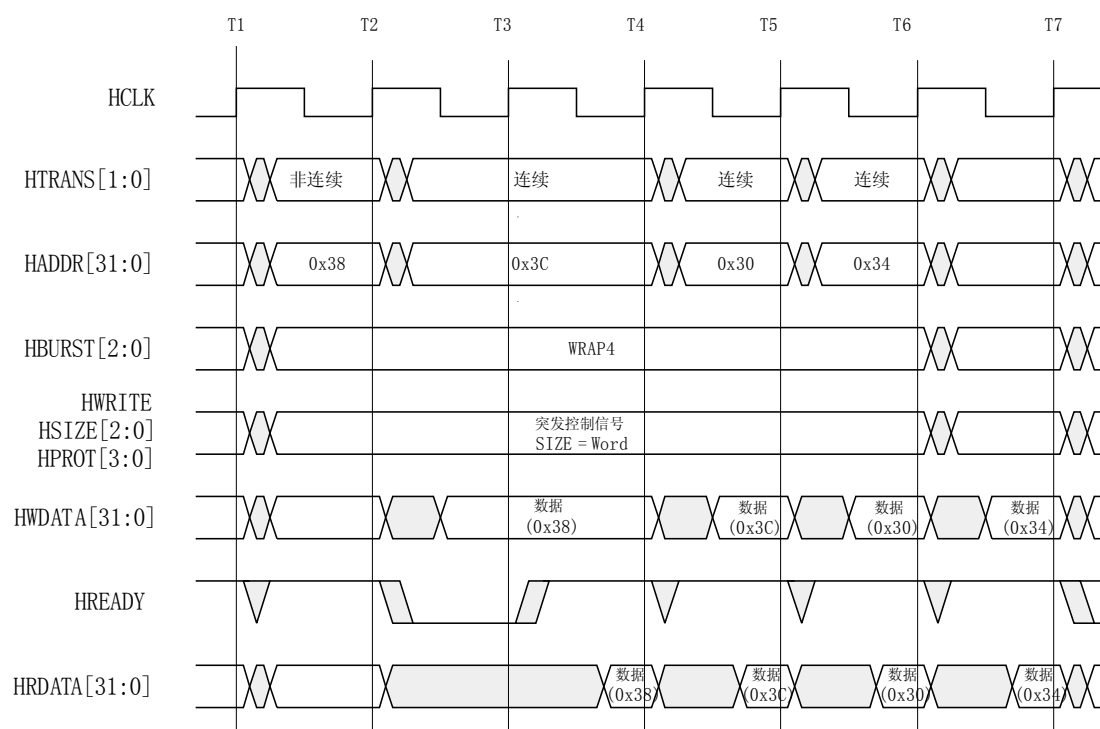


图 3.7 四拍回环突发

作为一次四拍字突发传输，地址将会在 16 字节边界回环，因此传输到地址 0x3C 之后接下来传输的地址是 0x30。图 3.8 表示了（回环突发）和增量突发的唯一不同，既是地址连续通过了 16 字节边界。

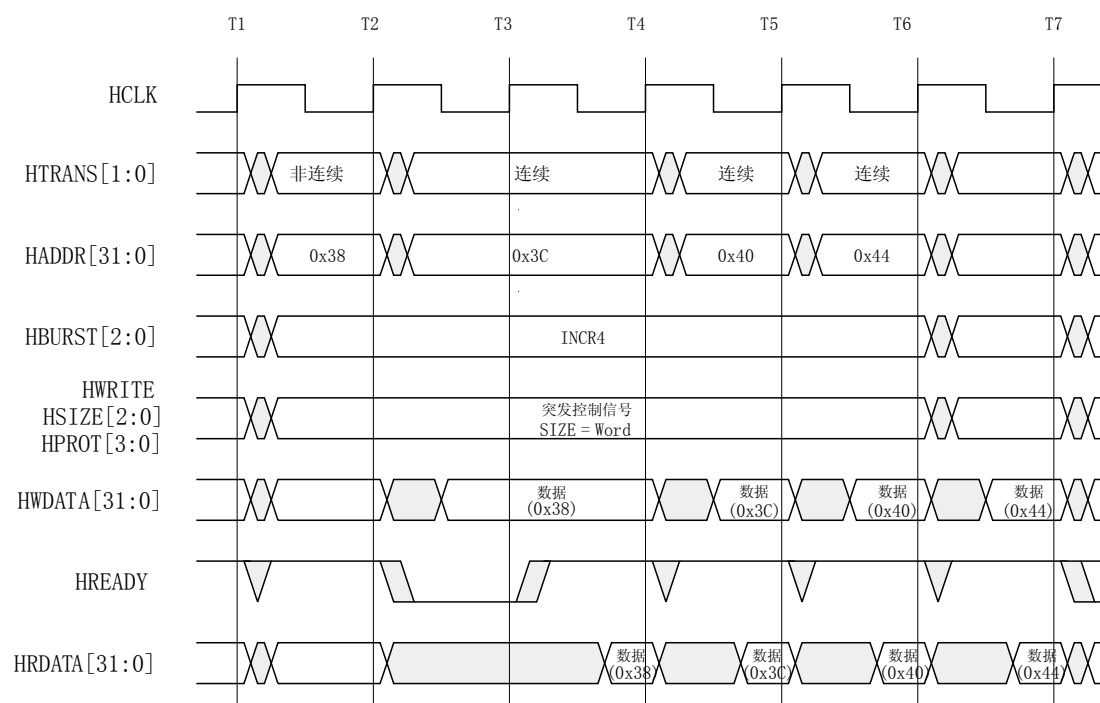


图 3.8 四拍增量突发

图 3.9 是一个八拍字传输突发。

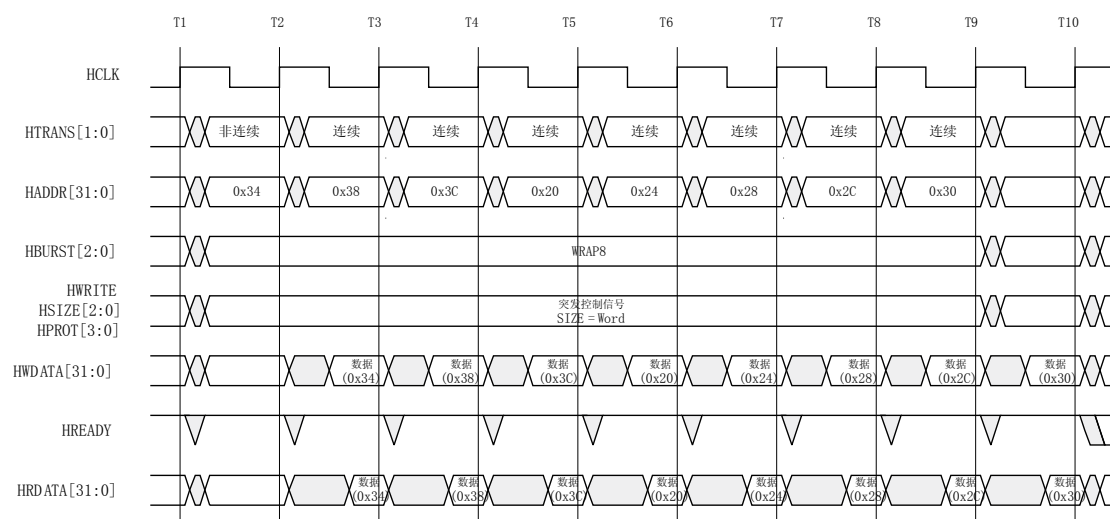


图 3.9 八拍回环突发

地址将在 32 字节边界处回环因此地址 0x3C 之后的地址是 0x20。

图 3.10中的突发使用半字传输，所以地址每次增加 2 个字节并且突发在递增因此地址连续增加通过了 16 字节边界。

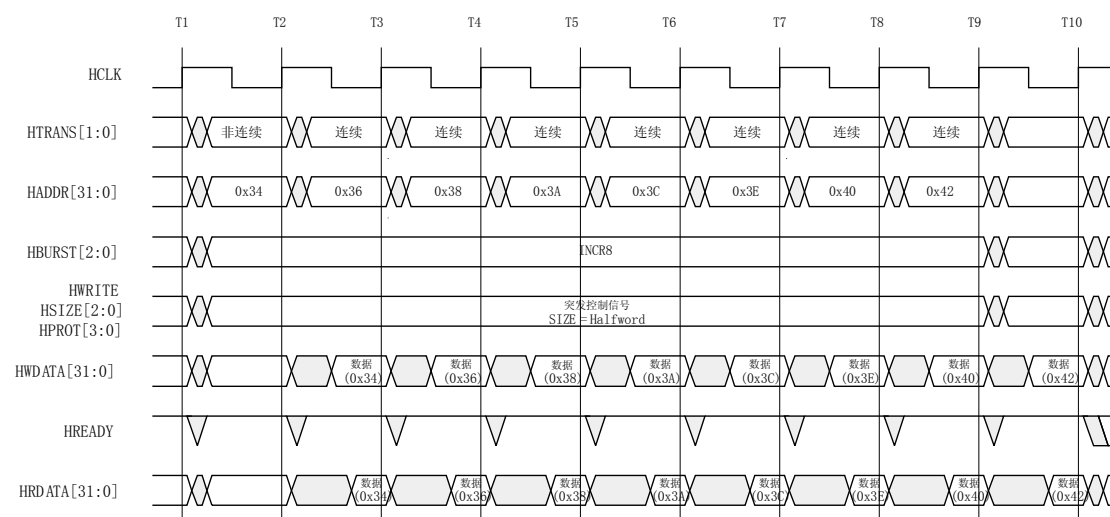


图 3.10 八拍增量突发

图 3.11表示未定义长度的增量突发。

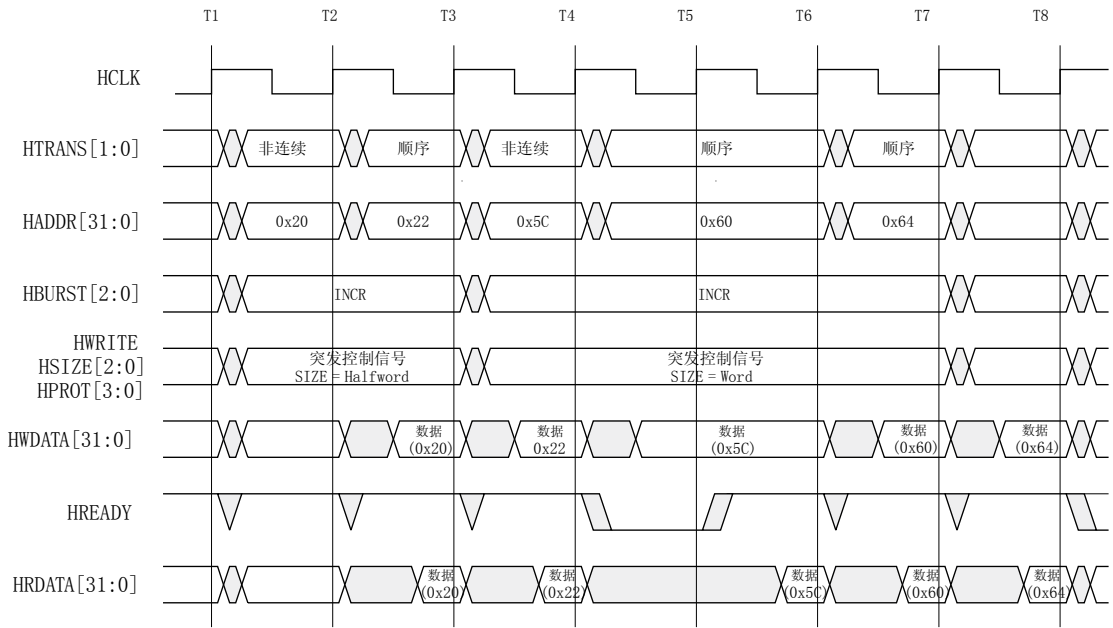


图 3.11 未定长度的突发

图 3.11表示两个突发：

- 两个半字传输在地址 0x20 处开始。半字传输地址增加为 2。
- 三个字传输在地址 0x5C 处开始。字传输地址增加为 4。

3.7 控制信号

和传输类型和突发类型一样每次传输将会有一组控制信号以提供传输的附加信息。这些控制信号和地址总线有严格一致的时序。然而，在一次突发传输过程中它们必须保持不变。

3.7.1 传输方向

当 **HWRITE** 为高，该信号表示一个写传输并且主机将数据广播到写数据总线上，**HWDATA[31: 0]**。当该信号为低时将会执行一个读传输并且从机必须产生数据到读数据总线 **HRDATA[31: 0]**。

3.7.2 传输大小

HSIZE[2: 0]表示传输的大小，见表 3.3。

表 3.3 大小编码

HSIZE[2]	HSIZE[1]	HSIZE[0]	大小	描述
0	0	0	8 位	字节
0	0	1	16 位	半字
0	1	0	32 位	字
0	1	1	64 位	-
1	0	0	128 位	4 字线
1	0	1	256 位	8 字线

续上表

HSIZE[2]	HSIZE[1]	HSIZE[0]	大小	描述
1	1	0	512 位	-
1	1	1	1024 位	-

（传输）大小被用来和 **HBURST[2: 0]** 信号一起决定回环突发的地址边界。

3.7.3 保护控制

保护控制信号，**HPROT[3: 0]**，提供总线访问的附加信息并且最初打算是给那些希望执行某种保护级别的模块使用的（见表 3.4）。

这些信号表示传输是否是：

- 一次预取指或者数据访问；
- 特权模式访问或者用户模式访问；

对于带有存储器管理单元的总线主机来说这些信号也表示当前访问是带高速缓存的（cache）还是带缓冲的（buffer）。

表 3.4 保护信号编码

HPROT[3] 高速缓存	HPROT[2] 带缓冲的	HPROT[1] 特权模式	HPROT[0] 数据/预取指	描述
-	-	-	0	预取指
-	-	-	1	数据访问
-	-	0	-	用户模式访问
-	-	1	-	特权模式访问
-	0	-	-	无缓冲
-	1	-	-	带缓冲
0	-	-	-	无高速缓存
1	-	-	-	带高速缓存

并不是所有总线主机都能产生正确的保护信息，因此建议从机在没有严格必要的情况下不要使用 **HPROT** 信号。

3.8 地址译码

对于每个总线上的从机来说使用一个中央地址译码器提供选择信号，**HSELx**。选择信号是高位地址信号的组合译码，并且建议使用简单的译码方案以避免复杂译码逻辑和确保高速操作。

从机只能在 **HREADY** 信号为高时采样地址和控制信号以及 **HSELx**，**HSELx** 为高表示当前传输已经完成。在特定的情况下有可能在 **HREADY** 为低时采样 **HSELx**，但是被选中的从机将会在当前传输完成后变更。

能够分配给单个从机的最小地址空间是 1KB。所有总线主机必须被设计为不能执行超过 1KB 地址边界的增量传输，因此确保了一个突发绝不会超过地址译码的边界。

在系统设计中如果有包含一个存储器映射并未完全填满（存储空间）的情况时应该设置一个额外的默认从机以在访问任何不存在的地址空间时提供响应。如果一个非连续或者连续传输试图访问一个不存在的地址空间时这个默认从机应该提供一个 **ERROR** 响应。空闲或者忙传输访问不存在的空间（默认从机）应该给出一个零等待状态的 **OKAY** 响应。典型默认从机的功能将以作为中央地址译码器的一部分来实现。

图 3.12表示了一个典型地址译码系统和从机选择信号。

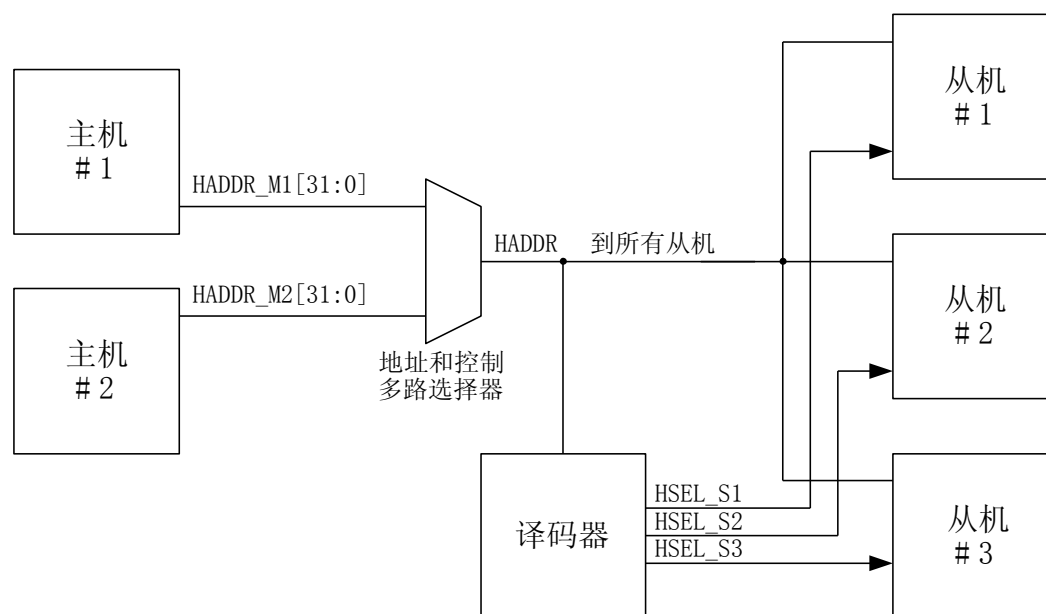


图 3.12 从机选择信号

3.9 从机传输响应

在主机发起传输后，由从机决定传输该如何进行。**AHB** 规范中没有做出总线主机在传输已经开始后取消传输的规定。

只要从机被访问那它必须提供一个表示传输状态的响应。**HREADY** 信号被用来扩展传输并且和响应信号 **HRESP[1:0]** 相结合，以提供传输状态。

从机能够用许多种方式来完成传输。它能：

- 立刻完成传输；
- 插入一个或者多个等待状态以允许有时间来完成传输；
- 发出一个错误信号来表示传输失败；
- 延时传输的完成，但是允许主机和从机放弃总线，把总线留给其他传输使用。

3.9.1 传输完成

HREADY 信号用来扩展一次 **AHB** 传输的数据部分。当 **HREADY** 信号为低时表示传输将被扩展而当其为高时表示传输完成。

注：每个从机必须有一个预先确定的在从机放弃总线之前插入的最大等待状态数目，以便能够计算访问总线的延时。建议但不强制规定，从机不要插入多于 16 个等待状态以阻止任何单个访问将总线锁定较长的时钟周期。

3.9.2 传输响应

典型的从机将会用 **HREADY** 信号来在传输中插入适当数量的等待状态而传输在 **HREADY** 为高时完成并且给出 **OKAY** 响应，表示传输成功完成。

ERROR 响应被从机用来表示某种形式的错误条件和相关的传输。典型的这被用作保护错误，例如试图写一个只读的存储空间。

SPLIT 和 **RETRY** 响应组合允许从机延长传输完成的时间，但是释放总线给其他主机使用。这些响应组合通常仅由有高访问延时的从机请求并且从机能够利用这些响应编码来确保其他主机在长时间内不被阻止访问总线。

关于 **SPLIT** 和 **RETRY** 的完整描述参见分块和重试。

HRESP[1: 0] 的编码、传输响应信号和每个响应的描述参见表 3.5。

表 3.5 响应编码

HRESP[1]	HRESP[0]	响应	描述
0	0	OKAY	当 HREADY 为高表示传输已经成功完成。 OKAY 响应也被用来插入任意一个附加周期，当 HREADY 为低时，优先给出其他三种响应之一。
0	1	ERROR	该响应表示发生了一个错误。错误条件应该发信号给总线主机以便让主机意识到传输失败。 一个错误条件需要双周期响应。
1	0	RETRY	RETRY （重试）信号表示传输并未完成，因此总线主机应该重试传输。主机应该继续重试传输直到完成为止。 要求双周期的 RETRY 响应。
1	1	SPLIT	传输并未成功完成。总线主机必须在下一次被授予访问总线时重试传输。当传输能够完成时从机将请求代替主机访问总线。 要求双周期的 SPLIT 响应。

当从机需要插入一定数量的等待状态优于决定将要给出何种响应时从机必须将响应驱动为 **OKAY**。

3.9.3 双周期响应

仅有 **OKAY** 响应可以在单个周期里给出。**ERROR**、**SPLIT** 和 **RETRY** 响应需要至少两个周期。为了完成这些响应中的任意一个那么在倒数第二个（最后一个的前一个）周期从机驱动 **HRESP[1: 0]** 以表示 **ERROR**、**RETRY** 或者 **SPLIT** 并同时驱动 **HREADY** 为低以给传输扩展一个额外的周期。在最后一个周期 **HREADY** 被驱动为高电平以结束传输，同时 **HRESP[1: 0]** 保持驱动以表示 **ERROR**、**RETRY** 或者 **SPLIT**。

如果从机需要两个以上的周期以提供 **ERROR**、**SPLIT** 或者 **RETRY** 响应那么额外的等待状态可能会在传输开始时被插入。在这段时间 **HREADY** 信号将为低电平同时响应必须被设为 **OKAY**。

需要双周期响应是因为总线通道的本质特征。在从机开始发出 **ERROR**、**SPLIT** 或者 **RETRY** 中任何一个响应时接下来传输的地址已经广播到总线上了。双周期响应允许主机有足够的时间来取消该地址并且在开始下一次传输之前驱动 **HTRANS[1:0]** 为空闲传输。

对于 **SPLIT** 和 **RETRY** 响应接下来的传输必须取消因为在当前传输完成之前禁止下一次传输发生。然而，对于 **ERROR** 响应，由于当前传输不被重复，所以可以选择完成接下来的传输。

图 3.13 表示了一次 **RETRY** 操作的例子。

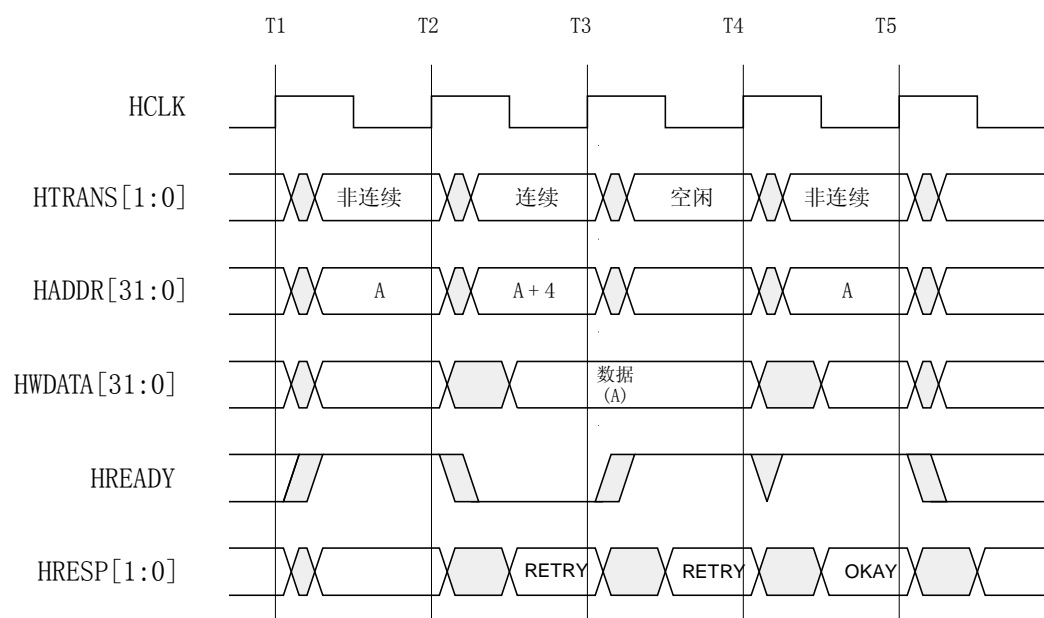


图 3.13 带 RETRY 响应的传输

图中包含以下事件：

- 主机从地址 A 发起传输；
- 这次传输在接收到响应之前主机将地址移动到 A + 4；
- 从机在地址 A 不能立刻完成传输因此从机发出一个 **RETRY** 响应。该响应指示主机在地址 A 的传输无法完成并且在地址 A + 4 的传输被取消而用空闲传输替代。

图 3.14 表示了一个传输中从机请求一个周期来决定将要给出的响应（在 **HRESP** 为 **OKAY** 的时间段），之后从机用一个双周期的 **ERROR** 响应结束了传输。

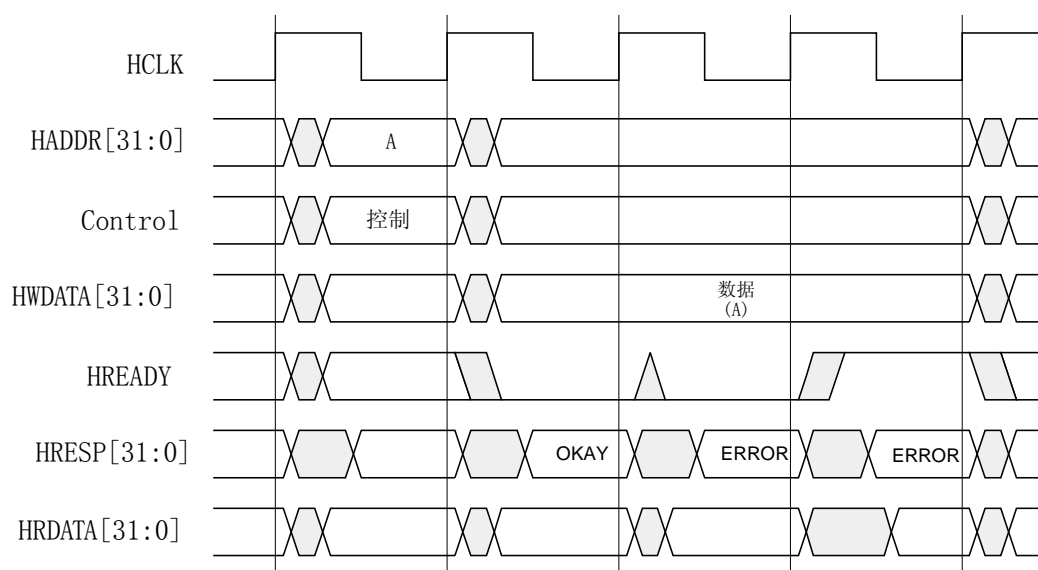


图 3.14 错误响应

3.9.4 错误响应

如果从机提供一个错误响应那么主机可以选择取消突发中剩下的传输。然而，这并不是一个严格要求同时主机继续突发中剩下的传输也是可以接受的。

3.9.5 分块和重试

分块和重试响应给从机提供了在无法立刻给传输提供数据时释放总线的机制。这两种机制都允许在总线上结束传输因此允许更高优先级的主机能够访问主机。

分块（SPLIT）和重试（RETRY）的不同之处在于仲裁器在发生 SPLIT 和 RETRY 后分配总线的方式：

- 对 RETRY 而言仲裁器将继续使用常规优先级方案因此只有拥有更高优先级的主机将获准访问总线；
- 对于 SPLIT 传输而言仲裁器将调整优先级方案以便其他任何主机请求总线即能获得访问（总线），即使是优先级较低的主机。为了完成一个 SPLIT 传输从机必须通知仲裁器何时数据可用。

SPLIT 传输都增加了仲裁器和从机的复杂性，但是却有可以完全释放总线给其他主机使用的优点，但是 RETRY（响应）的情况就只允许较高优先级的主机使用总线。

总线主机应该以同样的方式来对待 SPLIT 和 RETRY（响应）。主机应该继续请求总线并尝试传输直到传输成功完成或者遇到 ERROR 响应时终止。

3.10 数据总线

为了不使用三态驱动而又允许执行AHB系统所以要求分开读和写数据总线。最小的数据宽度规定为 32 位，但是总线宽度却可以增加，参见关于AHB数据总线的位宽这一节中的描述。

3.10.1 HWDATA[31: 0]

写数据总线在写传输期间由总线主机驱动。如果传输是扩展的那么总线主机必须保持数据有效直到传输完成，由 HREADY 为高表示。

所有传输必须对齐到和传输大小相等的地址边界。例如，字传输必须对齐到字地址边界（也就是 $A[1:0] = 00$ ），半字传输必须对齐到半字地址边界（也就是 $A[0] = 0$ ）。

对于宽度小于总线宽度的传输，例如一个在 32 位总线上的 16 位传输，那么总线主机仅需要驱动相应的字节通道。从机必须负责从正确的字节通道选择写数据。表 3.6 和表 3.7 分别表示了小端系统和大端系统中哪个字节通道有效。如果有要求，这些信息可以在更宽的总线应用中扩展。传输大小小于数据总线宽度的突发传输将在每拍突发中有不同有效字节通道。

有效字节通道取决于系统的端结构，但是 AHB 并不指定要求的端结构。因此，总线上所有主机和从机的端结构相同这点很重要。

3.10.2 HRDATA[31:0]

读数据总线在读传输期间由合适的从机驱动。如果从机通过拉低 **HREADY** 扩展读传输那么从机只需要在传输的最后一个周期提供有效数据，由 **HREADY** 为高表示。

对于宽度小于总线宽度的传输从机仅需要在有效的字节通道提供有效数据，如表 3.6 和表 3.7 所示。总线主机负责从正确的字节通道中选择数据。

当传输以 **OKAY** 响应完成时从机仅需提供有效数据。**SPLIT**、**RETRY** 和 **ERROR** 响应不需要提供有效的读数据。

表 3.6 32 位小端数据总线的有效字节通道

传输大小	地址偏移	DATA[31: 24]	DATA[23: 16]	DATA[15: 8]	DATA[7: 0]
字	0	√	√	√	√
半字	0	-	-	√	√
半字	2	√	√	-	-
字节	0	-	-	-	√
字节	1	-	-	√	-
字节	2	-	√	-	-
字节	3	√	-	-	-

表 3.7 32 位大端数据总线的有效字节通道

传输大小	地址偏移	DATA[31: 24]	DATA[23: 16]	DATA[15: 8]	DATA[7: 0]
字	0	√	√	√	√
半字	0	√	√	-	-
半字	2	-	-	√	√
字节	0	√	-	-	-
字节	1	-	√	-	-
字节	2	-	-	√	-
字节	3	-	-	-	√

3.10.3 端结构

为了使系统正确运行事实上所有模块都是相同端结构的并且任何数据通路或者桥接器也是相同端结构的。

不支持动态端结构，因为在大多数嵌入式系统中，这将导致明显的硅晶片较高，也就是多余的。

对于模块设计者而言建议只有应用场合非常宽泛的模块才应该被设计为双端结构的，通过一个配置引脚或者内部控制位来选择端结构。对于更多的特定用途的模块，固定端结构为大端或者小端将产生体积更小、功耗更低、性能更高的接口。

3.11 仲裁

仲裁机制被用来确保任意时刻只有一个主机能够访问总线。仲裁器的功能是检测许多不同的使用总线的请求和决定当前请求总线的主机中哪一个的优先级最高。仲裁器也接收来自从机需要完成 **SPLIT** 传输的请求。

任何没有能力执行 **SPLIT** 传输的从机不需要了解仲裁的过程，除非它们需要检测因为总线所有权改变而导致突发传输不能完成的情况。

3.11.1 信号描述

以下给出对每个仲裁信号的简短描述：

HBUSREQx	被总线主机用来请求访问总线的总线请求信号。每个总线主机都有自己的连接到仲裁器的 HBUSREQx 信号并且任何一个系统中都可以有高达 16 个独立的总线主机。
HLOCKx	由主机在请求总线的同时时断言的锁定信号。这提示仲裁器主机正在执行一系列不可分割的传输并且一旦锁定传输的第一个传输已经开始仲裁器不能授予任何其他主机访问总线。 HLOCKx 必须在涉及到的地址被寻址到之前至少断言一个周期，以防止仲裁器改变授予信号。
HGRANTx	授予信号由仲裁器产生并且表示相关主机是当前请求总线的主机中优先级最高的主机，（优先）考虑锁定传输和 SPLIT 传输。 主机在 HGRANTx 为高时获取地址总线的所有权并且在 HCLK 的上升沿 HREADY 为高电平。
HMASTER[3: 0]	仲裁器使用 HMASTER[3: 0] 信号表示哪一个主机当前被授予总线并且该信号可被用来控制中央地址和控制多路选择器。有 SPLIT 传输能力的从机也可以请求主机的序号以便它们能够提示仲裁器哪个主机能够完成一个 SPLIT 传输。
HMASTLOCK	仲裁器通过断言 HMASTLOCK 信号指示当前传输是一个锁定序列的一部分，该信号和地址以及控制信号有相同的时序。
HSPLIT[15: 0]	这 16 位有完整分块能力的总线被有分块（ SPLIT ）能力的从机用来指示哪个总线主机能够完成一个 SPLIT 传输。仲裁器需要这些信息以便于授予主机访问总线完成传输。

下列小节提供更多的信息：

- 请求总线访问；
- 授予总线访问；
- 突发提前终止；
- 锁定传输。

3.11.2 请求总线访问

总线主机使用 **HBUSREQx** 信号来请求访问总线并且可以在任何周期请求总线。仲裁器将在时钟的上升沿采样（主机的）请求然后使用内部优先级算法来决定哪个主机将会下一个获得访问总线。

如果主机请求锁定访问（总线）那么主机也必须断言 **HLOCKx** 信号来提示仲裁器其他主机不应该被授予总线。

当一个主机被授予总线并且正在执行一个固定长度的突发那么就没有必要继续请求总线以便完成传输。仲裁器监视突发的进程并且使用 **HBURST[2: 0]**信号来决定主机请求了多少个输。如果主机希望在当前正在进行的传输之后执行另一个突发那么主机需要在突发中重新断言请求信号。

如果主机在一次突发当中失去对总线的访问那么它必须重新断言 **HBUSREQx** 请求线以重新获取访问总线。

对未定长度的突发主机应该继续断言请求直到已经开始最后一次传输。在未定长度的突发结束时仲裁器不能预知何时改变仲裁。

对于主机而言有可能当它未申请总线时却被授予总线。这可能在没有主机请求总线并且仲裁器将访问（总线）授予一个默认的主机时发生。因此，如果一个主机并没请求访问总线那么它驱动传输类型 **HTRANS** 来表示空闲传输显得很重要。

3.11.3 授予总线访问

仲裁器通过断言适当的 **HGRANTx** 信号来表示请求总线的主机中哪个是当前优先级最高的。当当前传输完成，由 **HREADY** 为高时所表示，那么主机将被授予（总线）并且仲裁器将改变 **HMASTER[3: 0]**信号来表示总线主机序号。

图 3.15表示了当所有传输都为零等待状态并且**HREADY**信号为高时的处理过程。

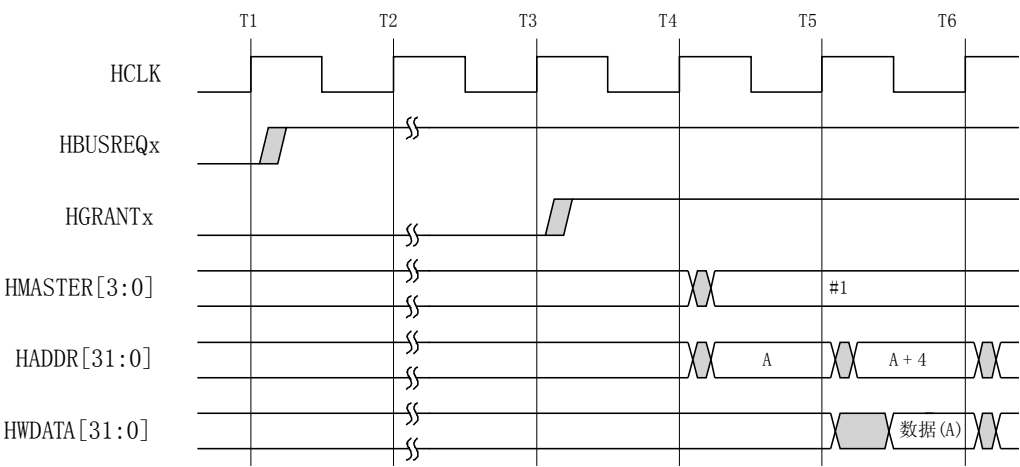


图 3.15 无等待状态访问授予

图 3.16表示了总线移交时等待状态的影响。

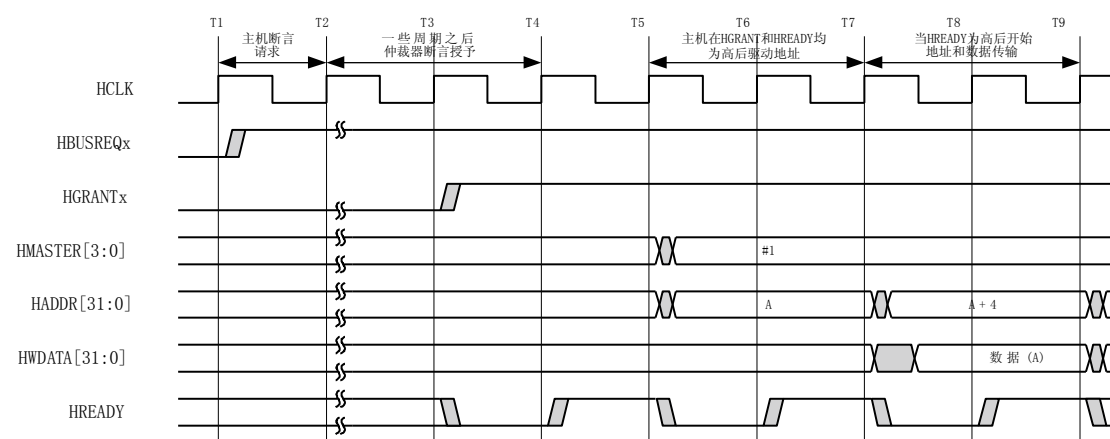


图 3.16 有等待状态的访授予

数据总线的所有权延时在地址总线的所有权之后。一次传输无论何时完成(由**HREADY**为高时所表示)然后占有地址总线的主机才能使用数据总线并且将继续占有数据总线直到传输完成。图 3.17表示当在两个总线主机之间移交总线时数据总线的所有权是如何转移的。

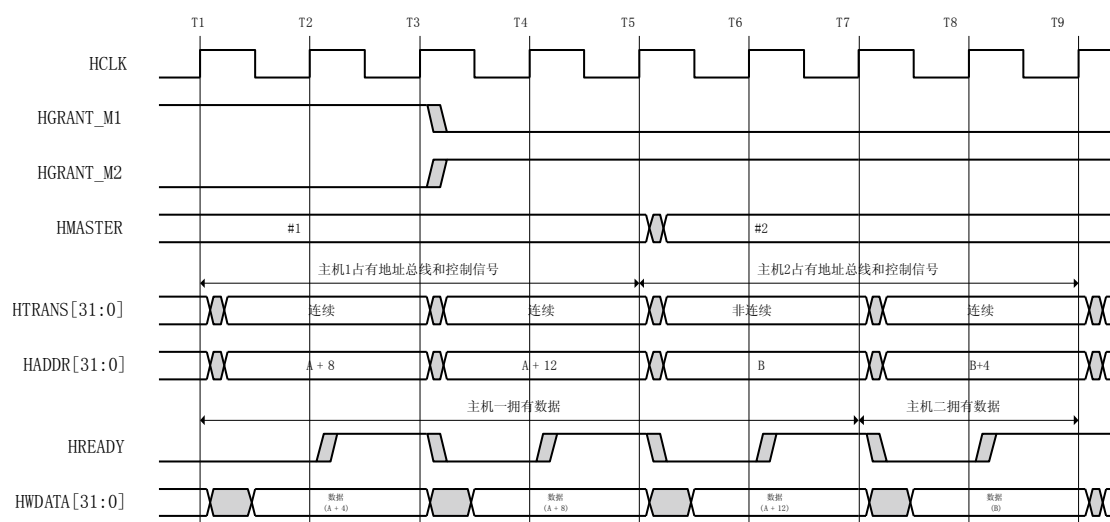


图 3.17 数据总线所有权移交过程

图 3.18表示一个仲裁器如何能在一次突发传输结束时移交总线的例子。

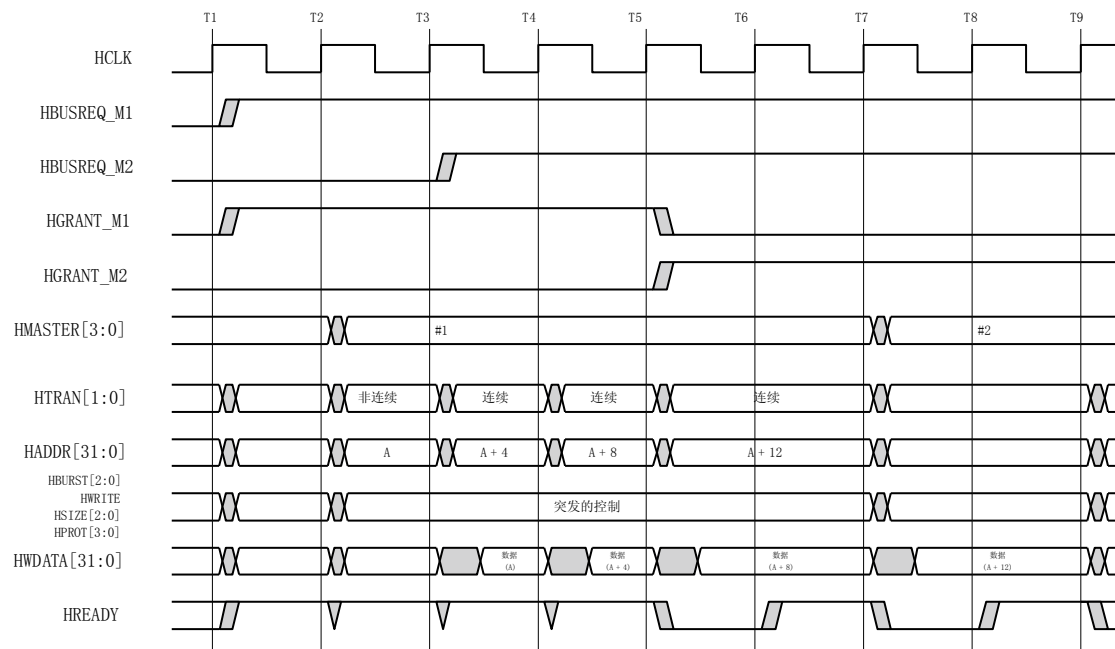


图 3.18 突发之后的移交

仲裁器在倒数第二个（最后一个之前的）地址被采样时改变 **HGRANT_x** 信号。新的 **HGRANT_x** 信息将在突发的最后一个地址被采样的同时被采样。

图 3.19表示了**HGRANT_x**和**HMASTER**信号是如何在系统中使用的。

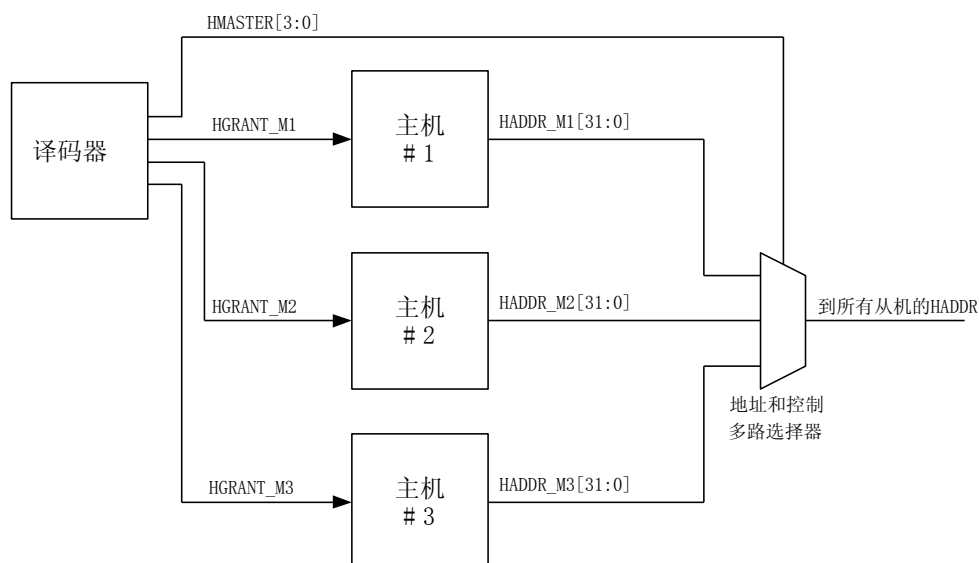


图 3.19 总线主机和授予信号

注: 因为使用了中央多路选择器, 每个主机可以立刻输出它希望执行的地址而不需要等到被授予总线。**HGRANT_x** 信号是仅被主机用来决定它何时拥有总线并因此需要考虑何时让地址被合适的从机采样。

HMASTER 总线的延时版本被用来控制写数据多路选择器。

3.11.4 突发提前终止

通常仲裁器在突发传输结束之前不会将总线移交给一个新的主机。但是，如果仲裁器决定突发必须被提前终止以防止过长的总线访问时间那么它可能会在一个突发完成之前将（总线）授予转移给另外一个总线主机。

如果主机在突发传输中间失去了对总线的所有权那么它必须重新断言总线（请求）以完成突发。主机必须确保 **HBURST** 和 **HTRANS** 信号都被更新以反映主机不再执行一个完整的 4、8 或者 16 拍的突发。

例如，如果一个主机仅能完成一个 8 拍突发的 3 个传输，那么当它重新获得总线时必须使用一个合法的突发编码来完成剩下的 5 个传输。主机可以使用任何合法组合，因此无论是 5 拍未定长度的突发或者是 4 拍固定长度的突发然后跟上一个单拍未定长度的突发都是可以接受的。

3.11.5 锁定传输

仲裁器必须监视来自各个主机的 **HLOCKx** 信号以确定何时主机希望执行一个锁定连续传输。之后仲裁器负责确保没有其他总线主机被授予总线直到锁定传输完成。

在一个连续锁定传输之后仲裁器将总是为一个附加传输保持总线主机被授予（总线）以确保锁定序列的最后一个传输成功完成并且没有接收到 **SPLIT** 或者 **RETRY** 响应。因此建议但不规定，主机在任何锁定连续传输之后插入一个空闲传输以提供给仲裁器在着手另外一个突发传输之前改变（总线授予）的机会。

仲裁器也负责断言 **HMASTLOCK** 信号，**HMASTLOCK** 信号和地址以及控制信号有相同的时序。该信号指示每个从机当前传输是锁定的因此必须在其他主机被授予总线之前被处理掉。

3.11.6 默认总线主机

每个系统必须包含一个默认总线主机，如果所有其他主机不能使用总线时该主机被授予总线。当被授予总线时，默认主机必须只能执行空闲（IDLE）传输。

如果没有请求总线那么仲裁器可以授予默认主机（访问总线）或者访问总线延时较低的主机将因此受益而被授予总线。

授予默认主机访问总线也为确保在总线上没有新的传输开始提供了一个有用的机制并且也是预先进入低功耗操作模式的有用步骤。

如果其他所有主机都在等待 **SPLIT** 传输完成时默认主机必须被授予总线。

3.12 分块传输

分块传输通过根据从机的响应操作来分离（或者分块）主机操作以给从机提供地址和合适的的数据，提高了总线的总体使用率。

当传输产生时如果从机认为传输的执行将占据大量的时钟周期那么从机能够决定发出一个 **SPLIT** 响应。该信号提示仲裁器尝试这次传输的主机不应该被授予访问总线，直到从机表示它准备好了完成传输时。因此仲裁器负责监视响应信号并且在内部屏蔽已经是 **SPLIT** 传输主机的任何请求。

在传输的地址相位期间仲裁器在 **HMASTER[3: 0]** 产生一个标记，或者总线主机序号，以表示正在执行传输的主机。任何一个发出 **SPLIT** 响应的从机必须表示它有能力完成这个传输，并且通过记录 **HMASTER[3: 0]** 信号上的主机序号来实现。

之后，当从机能够完成传输时，它将根据主机序号在从从机到主机的 **HSPLITx[15: 0]** 信号上断言适当的位。然后仲裁器使用这个信息来解除来自主机请求信号的屏蔽并且主机将

被及时授予访问总线以重试传输。仲裁器在每个时钟周期采样 **HSPLITx** 总线因此从机只需要断言适当的位一个周期以便仲裁器能够识别。

如果系统中有多个具有 **SPLIT** 能力的从机那么每个从机的 **HSPLITx** 总线可以逻辑或在一起以提供给仲裁器单个 **HSPLIT** 总线。

大多数系统中并没有用到最大 16 个总线主机的能力因此仲裁器仅要求一个位数和总线主机数量一样的 **HSPLIT** 总线。但是，建议所有有 **SPLIT** 能力的从机被设计成支持高达 16 个主机。

3.12.1 分块传输顺序

SPLIT 传输的基本步骤如下：

- 1、 主机以和其他传输一样的方式发起传输并发出地址和控制信息；
- 2、 如果从机能够立刻提供数据那么它可以马上提供数据。如果从机确认获取数据可能会占据较多的周期那么它给出一个 **SPLIT** 传输响应；
每次传输中仲裁器广播一个序号或者标记，表示哪个主机正在使用总线。从机必须记录该序号，以使用来在之后的一段时间重新发起传输；
- 3、 仲裁器授予其他主机使用总线并且 **SPLIT** 响应的动作允许主机移交总线。如果所有其他主机也接收到一个 **SPLIT** 响应那么默认主机将被授予总线；
- 4、 当从机准备完成传输那么它断言 **HSPLITx** 总线中的适当位给仲裁器以指示哪个主机应该被重新授予访问总线；
- 5、 仲裁器每个时钟周期监视 **HSPLITx** 信号，并且当 **HSPLITx** 中的任何一位被断言仲裁器将恢复对应主机的优先级；
- 6、 最后仲裁器将授予（**SPLIT** 的）主机总线，因此主机能重新尝试传输。如果一个优先级更高的主机正在使用总线的话这可能不会立刻发生；
- 7、 当传输终于开始后从机以一个 **OKAY** 传输响应来结束（传输）。

3.12.2 多重分块传输

总线协议只允许每个总线主机有一个未完成的处理。如果任何主机模块能够处理多于一个未完成的处理那么它需要为能够处理的每个未完成处理设置一个额外的请求和授予信号。在协议级上一个信号模块可以表现为许多不同总线主机，每个主机只能有一个未完成的处理。

然而，可能一个有 **SPLIT** 能力的从机会接收比它能并发处理的（传输）还要多的传输请求。如果这种情况发生那么从机可以不用记录对应传输的地址和控制信息而仅需要记录主机序号就发出 **SPLIT** 响应。之后从机可以通过断言 **HSPLITx** 总线中适当的位给之前被给出 **SPLIT** 响应的所有主机来表示它能处理另外一个传输，但是从机没有记录地址和控制信息。

之后仲裁器能够重新授予这些主机访问总线并且它们将重试传输，给出从机要求的地址和控制信息。这表示一个主机可以在它最终完成它要求的传输之前被多次授予总线。

3.12.3 预防死锁

SPLIT 和 **RETRY** 传输响应都必须在使用中注意预防总线死锁。单个传输决不会锁定 **AHB**，因为每个从机必须被设计成能在预先确定的周期数内完成传输。但是，如果多个不同主机试图访问同一个从机，从机发出 **SPLIT** 或者 **RETRY** 响应以表示从机不能处理，那么就有可能发生死锁。

分块传输

从机可以发出 **SPLIT** 传输响应，通过确保从机能够承受系统中每个主机（最多 16 个）的单个请求来预防死锁。从机并不需要存储每个主机的地址和控制信息，它只需要简单的记录传输请求已经被处理和 **SPLIT** 响应已经发出的事实即可。最后所有主机将处在低优先级然后从机可以有次序的来处理这些请求，指示仲裁器正在服务于哪个请求，因而确保了所有请求最终都被服务。

当从机有许多未完成的请求时它可能以任何顺序（随机的）来选择处理这些请求，尽管从机需要注意锁定传输必须在任何其他传输继续之前完成。

从机使用 **SPLIT** 响应而不用锁存地址和控制信息显得非常合法（合适）。从机仅需要记录特定主机做出的传输尝试并且稍后的时间段从机通过指示自己已经准备好完成传输就能获取地址和控制信息。主机将被授予总线并将重新广播传输，允许从机锁存地址和控制信息并且立刻应答数据，或者发出另外一个 **SPLIT** 响应如果还需要额外的一些周期的话。

理想情况下从机不应该有多于它能支持的未完成传输，但是要求支持这种机制以防止总线死锁。

重试传输

发出 **SPLIT** 响应的从机必须一次只能被一个主机访问。在总线协议中并没有强制而在系统体系结构中应该确保这一点。大多数情况下发出 **RETRY** 响应的从机必须是一次只能被一个主机访问的外设，因此这会在一些更高级协议中得到确保。

硬件保护和多主机访问 **RETRY**（响应）的从机相违背并不是协议中的要求，但是可能会在下文描述的设计中得到执行。仅有的总线级要求是从机必须在预先确定的时钟周期内驱动 **HREADY** 为高。

如果要求硬件保护那么这可以被 **RETRY**（响应）的从机自己执行。当一个从机发出一个 **RETRY** 信号后它能够采样主机序号。在这之后和传输最终完成之前 **RETRY** 的从机可以检查做出的每次传输尝试以确保主机序号是相同的。如果从机发现主机号不一致那么它可以选择下列的行动方式：

- 一个错误响应；
- 一个信号给仲裁器；
- 一个系统级中断；
- 一个完全的系统复位。

3.12.4 分块传输的总线移交

协议要求主机在接收到一个 **SPLIT** 或者 **RETRY** 响应后立刻执行一个空闲传输以允许总线转移给另外一个主机。图 3.20 表示了发生一个分块（**SPLIT**）传输的顺序事件。

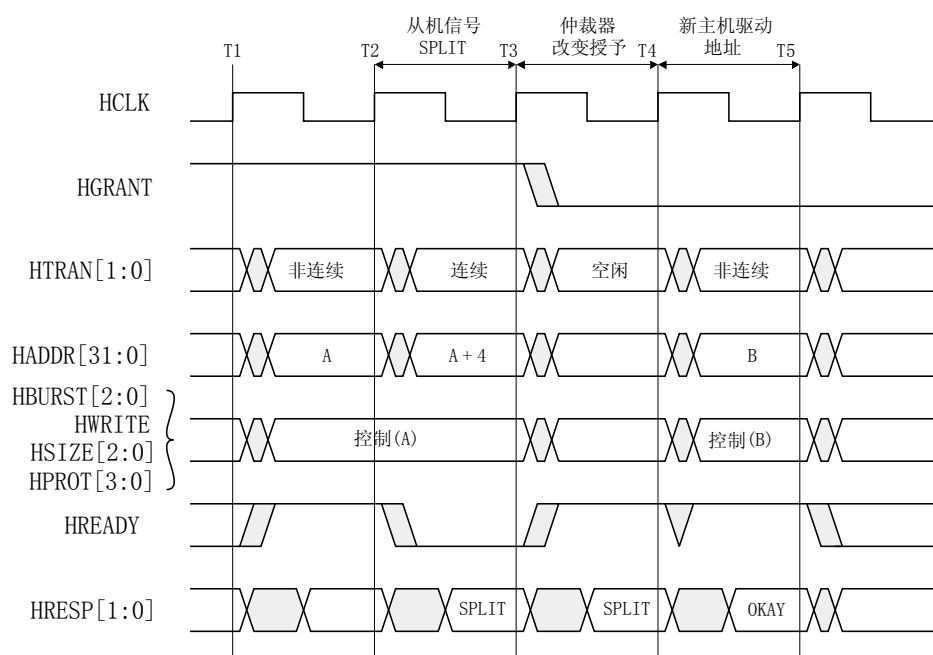


图 3.20 分块传输后的总线移交

需要注意以下的要点：

- 传输的地址在时间 T1 之后出现在总线上。在时钟沿 T2 和 T3 后从机返回两个周期的 **SPLIT** 响应；
- 在第一个响应周期的末尾，也就是 T3，主机能够检测到传输将会被分块因此（主机）改变接下来的传输控制信号以表示一个空闲传输；
- 同样也在时间 T3 处仲裁器采样响应信号并确定传输已经被分块。之后仲裁器可以调整仲裁优先权并且在接下来的周期改变授予信号，这样新的主机能够在时间 T4 后被授予地址总线；
- 新主机可以保证立刻访问（总线）因为空闲传输总是在一个周期内完成。

3.13 复位

复位信号，**HRESETn**，是 AMBA AHB 规范中唯一的低有效信号并且是所有总线设备的主要复位源。复位可以异步方式断言，但是却在 **HCLK** 的上升沿被同步地撤消断言。

在复位期间所有主机必须确保地址和控制信号在有效电平并且 **HTRANS[1:0]** 信号表示空闲。

3.14 关于AHB数据总线的位宽

一种能提高总线带宽而不用提高操作频率的方法是使片上总线的数据通道更宽。金属层的增加和大容量片上存储模块（例如嵌入式 **DRAM**）的使用都是更宽片上总线使用的推动因素。

指定一个固定宽度的总线将意味着在大多数场合下总线宽度在应用中并不是最佳的。因此允许可变总线宽度的途径已经被采纳，但是必须确保模块在设计中高移植性。

协议允许 AHB 数据总线可以是 8、16、32、64、128、256、512 或者 1024 位宽。然而，建议使用中最低的总线宽度为 32 位并且预计最大 256 位宽的总线将适合几乎所有应用。

对读和写传输而言接收模块都必须从总线上正确的字节通道选择数据。并不要求将数据复制到所有字节通道上。

3.15 在宽总线上实现窄从机

图 3.21表示了一个原始设计为以 32 位数据总线运行的从机模块是如何轻松的被转换以运行在较宽的 64 位总线上的。这仅需要增加外部逻辑，而不是任何内部的设计变更，因此该技术也可以用在难以设计的宏单元上。

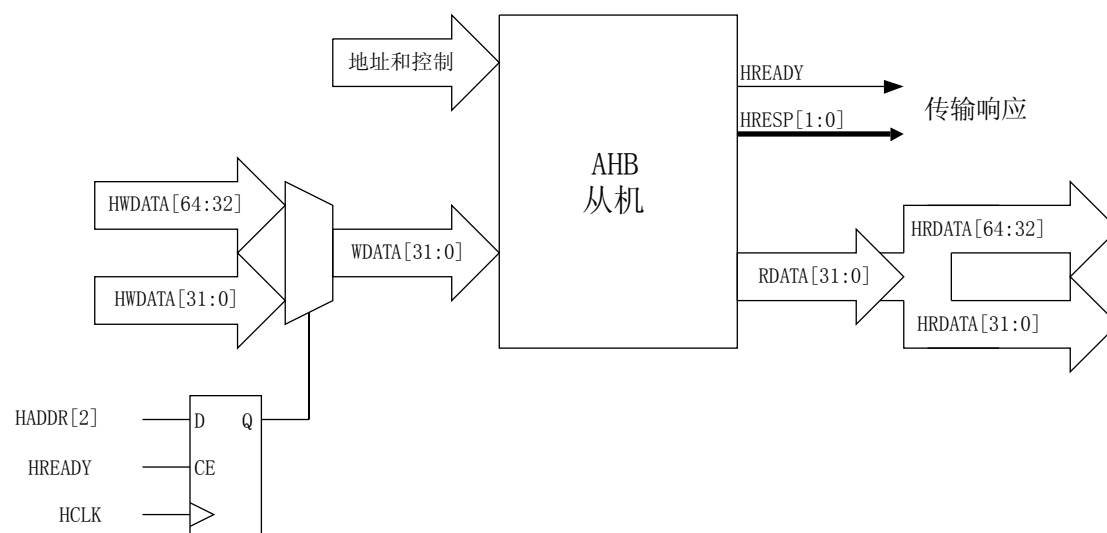


图 3.21 宽总线上的窄从机

对于输出，当从较窄的总线转换成较宽的总线时，要完成下列事件之一：

- 复制数据到宽总线上的两个半部分上（如上图所示）；
- 使用附加的逻辑电平来确保总线上只有适当的那一半被改变。这会导致功耗的降低。

从机可以只接收和它接口相同宽度的传输。如果一个主机尝试一个大于从机能支持的传输那么从机可以使用 **ERROR** 传输响应。

3.16 在窄总线上实现宽从机

图 3.22表示了一个在窄总线上实现宽从机的例子。同样只需要外部逻辑因此初步的设计或者导入的模块可以简单的修改来在不同宽度的数据总线上工作。

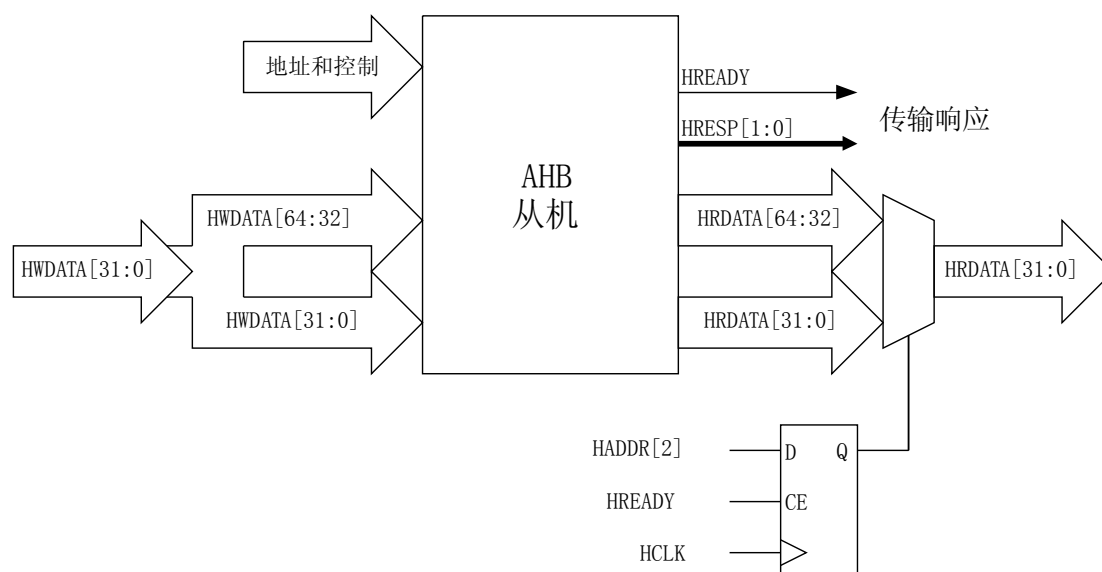


图 3.22 窄总线上的宽从机

3.16.1 主机

与最初打算通过用相同的方式修改从机以工作在宽总线上相比，总线主机可以用下面的方式被简单修改便能工作在宽总线上：

- 多路选择输入总线；
- 复制输出总线；

然而，总线主机不能工作在比原先设计要窄的总线上，除非有一些限制总线主机尝试传输的宽度的机制将主机也包含在内。主机禁止尝试宽度（由 **HSIZE** 表示）大于所连接的数据总线的传输。

3.17 关于AMBA AHB的组成部分

这部分描述了基于 AHB 的 AMBA 系统的各个元素并提供分析一个 AMBA 设计所要求的常规时序参数。

下面的符号用作时序参数：

- T_{is} – 输入建立时间；
- T_{ih} – 输入保持时间；
- T_{ov} – 输出有效时间；
- T_{oh} – 输出保持时间；

3.18 AHB总线从机

一个 AHB 总线从机在系统中应答由总线主机表示的传输。从机使用一个来自译码器的 **HSELx** 选择信号以确认何时响应总线主机。所有传输要求的其他信号，例如地址和控制信息，将由总线主机产生。

3.18.1 接口框图

图 3.23表示了一个AHB总线从机接口。

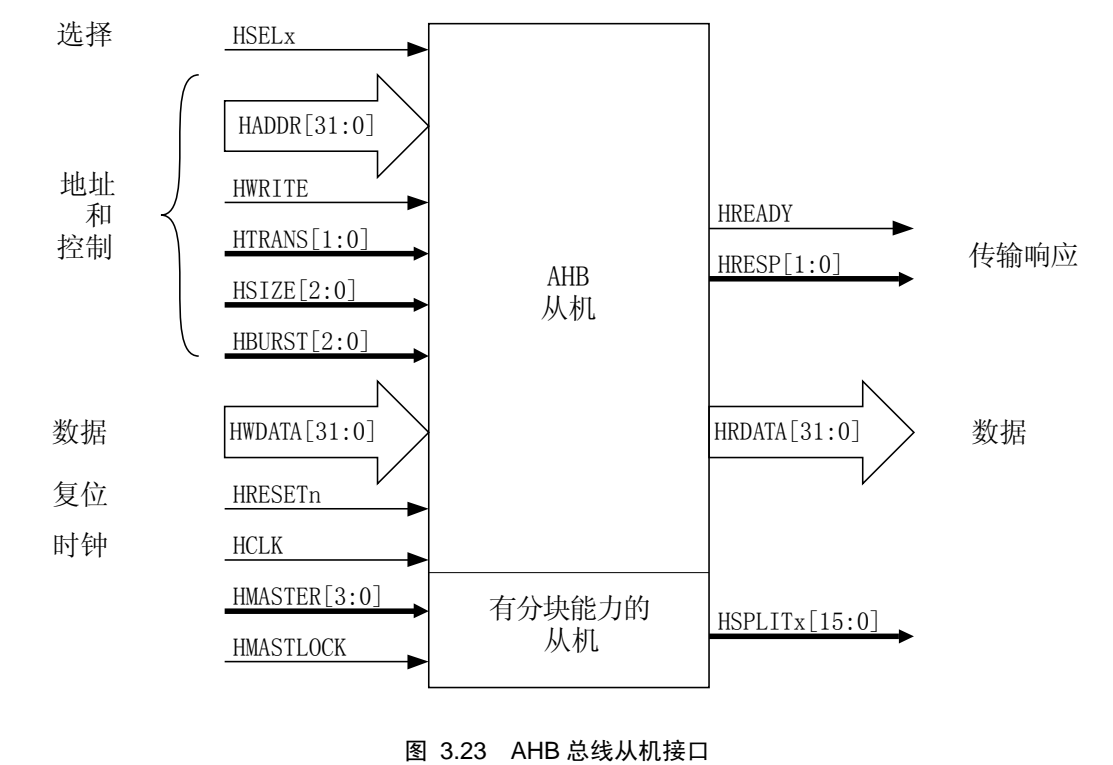


图 3.23 AHB 总线从机接口

3.18.2 时序图

下图表示了在 AMBA 系统中与访问 AHB 总线从机操作有关的时序参数：

- 图 3.24表示了AHB从机复位时序参数；
- 图 3.25表示了AHB从机主要的时序参数；
- 图 3.26表示了有分块能力的AHB从机的附加时序参数。

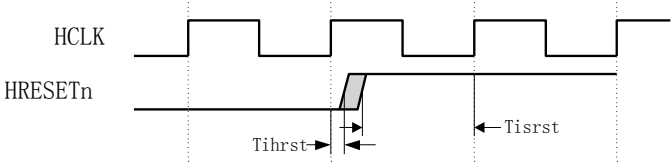


图 3.24 AHB 从机复位时序

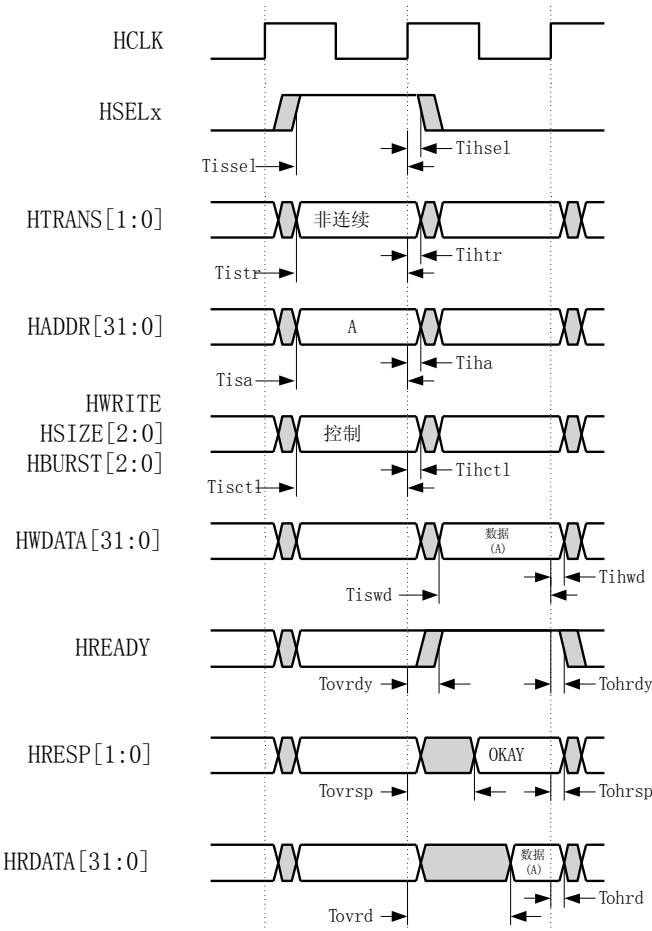


图 3.25 AHB 时序参数

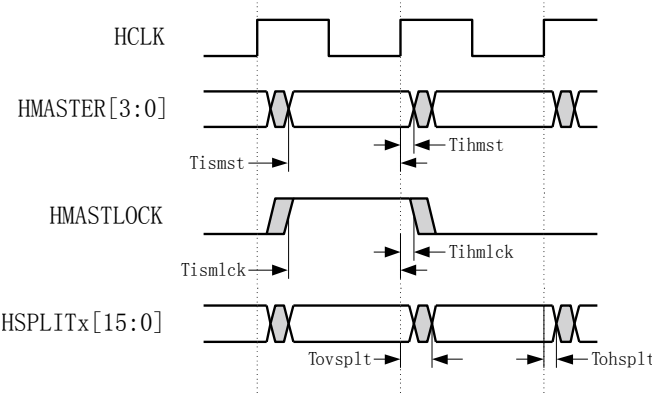


图 3.26 有分块能力的从机的附加参数

3.18.3 时序参数

下面给出关于AHB总线从机的时序参数，表 3.8为输入信号而表 3.9为输出信号。

表 3.8 AHB 从机输入参数

参数	描述
T _{clk}	HCLK 最小时钟周期

续上表

参数	描述
T_{isrst}	在 HCLK 之前 HRESETn 失效的建立时间
T_{ihrst}	在 HCLK 之后 HRESETn 失效的保持时间
T_{issel}	在 HCLK 之前 HSELx 的建立时间
T_{ihsel}	在 HCLK 之后 HSELx 的保持时间
T_{istr}	在 HCLK 之前传输类型的建立时间
T_{ihtr}	在 HCLK 之后传输类型的保持时间
T_{isa}	在 HCLK 之前 HADDR[31: 0] 的建立时间
T_{iha}	在 HCLK 之后 HADDR[31: 0] 的保持时间
T_{isctl}	在 HCLK 之前 HWRITE 、 HSIZE[2: 0] 和 HBURST[2: 0] 控制信号的建立时间
T_{ihctl}	在 HCLK 之后 HWRITE 、 HSIZE[2: 0] 和 HBURST[2: 0] 控制信号的保持时间
T_{iswd}	在 HCLK 之前写数据的建立时间
T_{ihwd}	在 HCLK 之后写数据的保持时间
T_{isrdy}	在 HCLK 之前准备信号的建立时间
T_{ihrdy}	在 HCLK 之后准备信号的保持时间
T_{ismst}	在 HCLK 之前主机序号的建立时间（有分块能力的才有）
T_{ihmst}	在 HCLK 之后主机序号的保持时间（有分块能力的才有）
T_{ismclk}	在 HCLK 之前主机锁定的建立时间（有分块能力的才有）
T_{ihmclk}	在 HCLK 之后主机锁定的保持时间（有分块能力的才有）

表 3.9 AHB 从机输出参数

参数	描述
T_{ovrsp}	在 HCLK 之后响应有效时间
T_{ohrsp}	在 HCLK 之后响应保持时间
T_{ovrdy}	在 HCLK 之后准备信号有效时间
T_{ohrdy}	在 HCLK 之后准备信号保持时间
T_{ovsplt}	在 HCLK 之后分块有效时间（有分块能力的才有）
T_{ohsplt}	在 HCLK 之后分块保持时间（有分块能力的才有）

3.19 AHB总线主机

在 AMBA 系统中 AHB 总线主机有最复杂的总线接口。典型的一个 AMBA 系统设计者应该使用预先设计的总线主机因此就不需要关注总线主机接口的细节。

3.19.1 接口框图

AHB 总线主机的接口框图表示了主要的信号群。

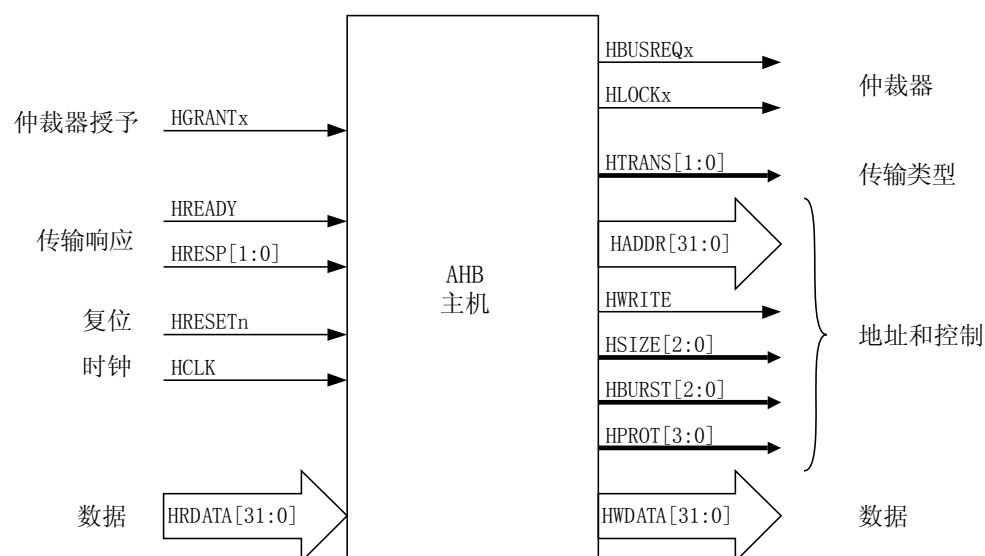


图 3.27 AHB 总线主机接口框图

3.19.2 总线主机时序

下面的插图表示了一个 AMBA 系统中关于 AHB 总线主机操作的时序参数：

- 图 3.28表示了AHB主机复位时序参数；
- 图 3.29表示了AHB主机传输时序参数；
- 图 3.30表示了AHB主机仲裁时序参数；

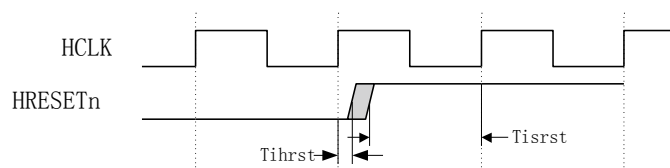


图 3.28 AHB 主机复位时序参数

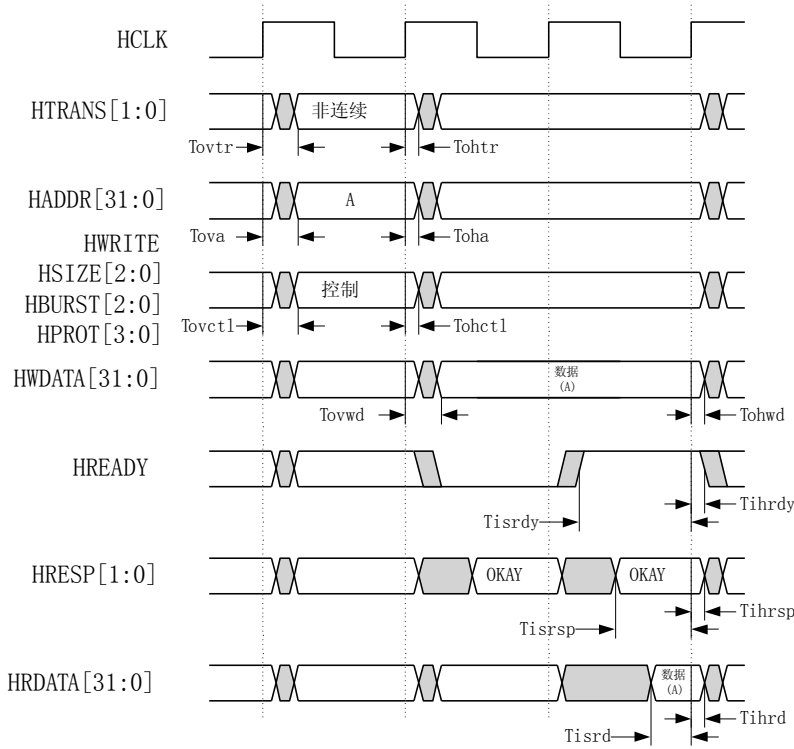


图 3.29 AHB 主机传输时序参数

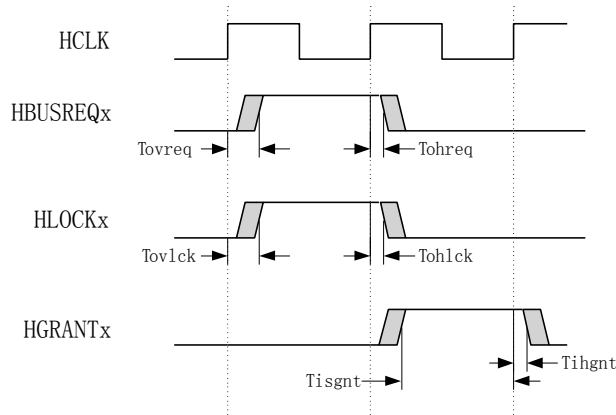


图 3.30 AHB 主机仲裁时序参数

3.19.3 时序参数

下面两个表格中用文本的方式表示了在AMBA系统中关于AHB总线主机操作的时序参数。表 3.10为输入信号的细节而表 3.11为输出信号的细节。

表 3.10 总线主机输入时序参数

参数	描述
T_{clk}	HCLK 最小时钟周期
T_{isrst}	在 HCLK 之前复位信号失效的建立时间
T_{ihrst}	在 HCLK 之后复位信号失效的保持时间
T_{isgnt}	在 HCLK 之前 HGRANTx 的建立时间

续上表

参数	描述
T_{ihgnt}	在 HCLK 之后 HGRANTx 的保持时间
$T_{isr dy}$	在 HCLK 之前准备信号的建立时间
$T_{ihr dy}$	在 HCLK 之后准备信号的保持时间
T_{isrsp}	在 HCLK 之前响应的建立时间
T_{ihrsp}	在 HCLK 之后响应的保持时间
T_{isrd}	在 HCLK 之前读数据的建立时间
T_{ihrd}	在 HCLK 之后读数据的保持时间

表 3.11 总线主机输出时序参数

参数	描述
T_{ovtr}	在 HCLK 之后传输类型的有效时间
T_{ohtr}	在 HCLK 之后传输类型的保持时间
T_{ova}	在 HCLK 之后地址的有效时间
T_{oha}	在 HCLK 之后地址的保持时间
T_{ovctl}	在 HCLK 之后控制信号的有效时间
T_{ohctl}	在 HCLK 之后控制信号的保持时间
T_{ovwd}	在 HCLK 之后写数据的有效时间
T_{ohwd}	在 HCLK 之后写数据的保持时间
T_{ovreq}	在 HCLK 之后请求信号的有效时间
T_{ohreq}	在 HCLK 之后请求信号的保持时间
T_{ovlck}	在 HCLK 之后锁定信号的有效时间
T_{ohlck}	在 HCLK 之后锁定信号的保持时间

3.20 AHB仲裁器

在 AMBA 系统中仲裁器的角色是控制哪个主机访问总线。每个总线主机有一个请求（REQUEST）/授予（GRANT）接口连到仲裁器并且仲裁器使用一个优先权分配方案来决定哪个总线主机是当前请求总线的主机中优先级最高的。

每个主机也可以产生用来表示主机请求独占总线访问的 **HLOCKx** 信号。

优先级方案的细节并没有规定而是由应用决定的。仲裁器使用其他信号，无论是 AMBA 的还是非 AMBA 的，来改变使用中的优先级方案都是可取的。

3.20.1 接口框图

图 3.31表示了一个AHB仲裁器的信号接口。

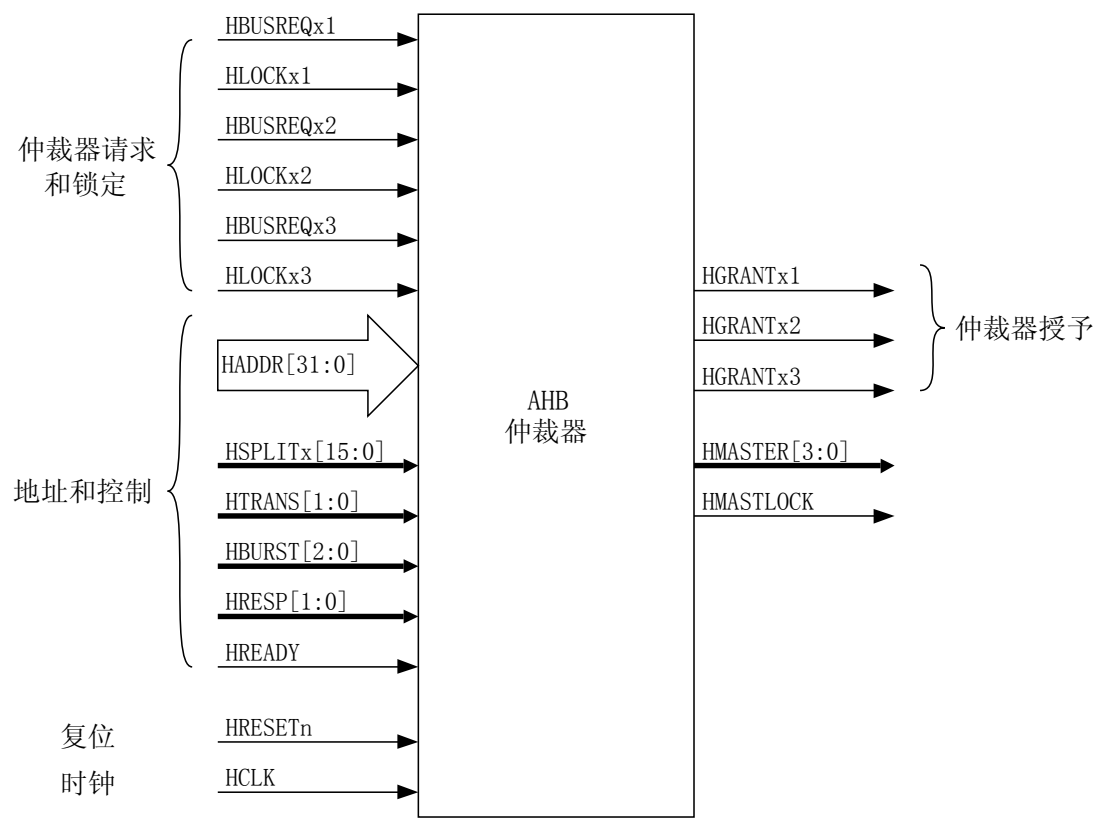


图 3.31 AHB 仲裁器接口框图

3.20.2 时序图

下面的插图表示了一个 AMBA 系统中关于 AHB 总线仲裁器操作的时序参数：

- 图 3.32表示了AHB仲裁器的复位时序参数；
- 图 3.33表示了AHB仲裁器的传输时序参数；
- 图 3.34表示了AHB仲裁器的分块时序参数。

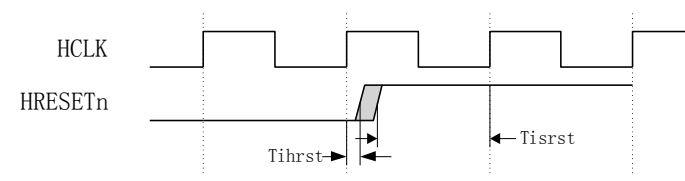


图 3.32 AHB 仲裁器复位时序参数

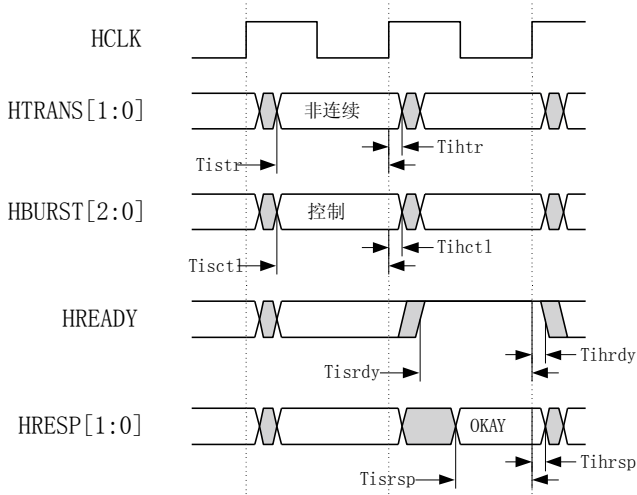


图 3.33 AHB 仲裁器传输时序参数

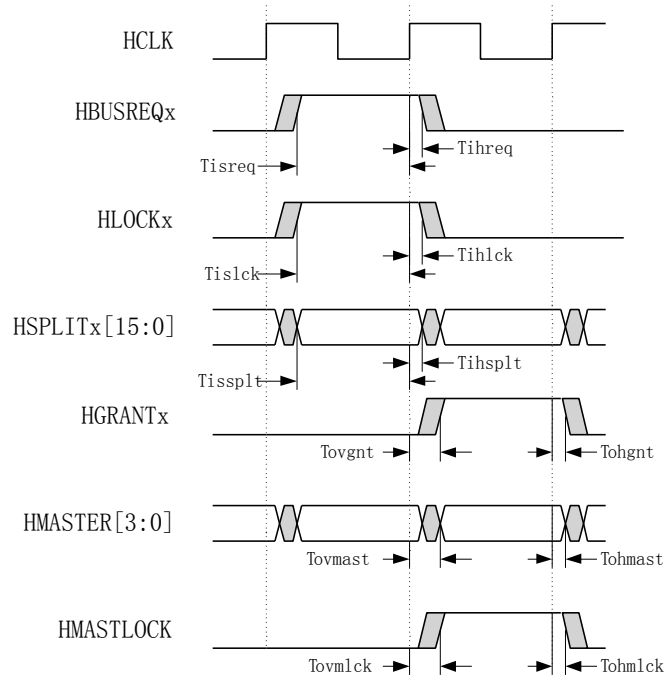


图 3.34 AHB 仲裁器分块时序参数

3.20.3 时序参数

下表给出关于 AHB 仲裁器的时序参数：

- 表 3.12是输入信号时序参数；
- 表 3.13是输出信号时序参数。

表 3.12 AHB 仲裁器输入参数

参数	描述
T _{clk}	HCLK 最小时钟周期
T _{isrst}	在 HCLK 之前复位信号失效的建立时间

续上表

参数	描述
T_{ihrst}	在 HCLK 之后复位信号失效的保持时间
$T_{isr dy}$	在 HCLK 之前准备信号的建立时间
T_{ihrdy}	在 HCLK 之后准备信号的保持时间
T_{isrsp}	在 HCLK 之前响应的建立时间
T_{ihrsp}	在 HCLK 之后响应的保持时间
T_{isreq}	在 HCLK 之前请求信号的建立时间
T_{ihreq}	在 HCLK 之后请求信号的保持时间
T_{islck}	在 HCLK 之前锁定信号的建立时间
T_{ihlck}	在 HCLK 之后锁定信号的保持时间
T_{issplt}	在 HCLK 之前分块信号的建立时间
T_{ihspkt}	在 HCLK 之后分块信号的保持时间
T_{istr}	在 HCLK 之前传输类型的建立时间
T_{ihtr}	在 HCLK 之后传输类型的保持时间
T_{isctl}	在 HCLK 之前控制信号的建立时间
T_{ihctl}	在 HCLK 之后控制信号的保持时间

表 3.13 AHB 仲裁器输出参数

参数	描述
T_{ovgnt}	在 HCLK 之后授予信号的有效时间
T_{ohgnt}	在 HCLK 之后授予信号的保持时间
T_{ovmst}	在 HCLK 之后主机序号的有效时间
T_{ohmst}	在 HCLK 之后主机序号的保持时间
T_{ovmlck}	在 HCLK 之后主机锁定的有效时间
T_{ohmlck}	在 HCLK 之后主机锁定的保持时间

3.21 AHB译码器

在 AMBA 系统中译码器被用来执行集中的地址译码功能，通过使它们独立于系统的存储器映射，可以提高外设的移植性。

3.21.1 接口框图

图 3.35表示了一个AHB译码器。

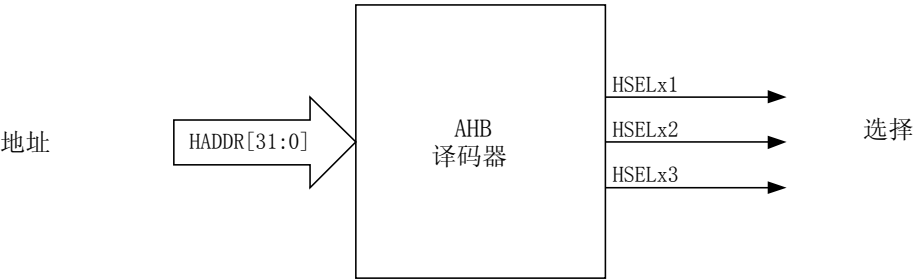


图 3.35 AHB 译码器接口框图

3.21.2 时序图

图 3.36给出了一个AHB译码器的时序参数。

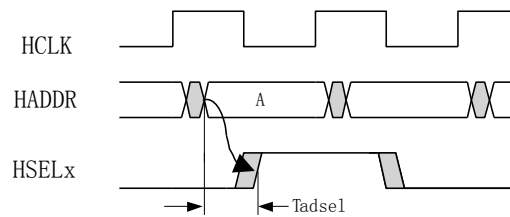


图 3.36 AHB 译码器时序参数

3.21.3 时序参数

表 3.14给出了关于AHB译码器的时序参数。

表 3.14 AHB 译码器输出参数

参数	描述
T _{adssel}	从地址到选择有效的延时

第4章 AMBA ASB

这章介绍了高级微控制器总线体系（AMBA）的高级系统总线规范。包含以下部分：

- 关于AMBA ASB；
- AMBA ASB描述；
- ASB传输；
- 地址译码；
- 传输响应；
- 多主机操作；
- 复位操作；
- ASB信号描述；
- 关于ASB AMBA的组成部分；
- ASB总线从机；
- ASB总线主机；
- ASB译码器；
- ASB仲裁器。

4.1 关于AMBA ASB

高级系统总线（Advanced System Bus, ASB）规范定义了一个用在设计高性能的 16 位和 32 位嵌入式微控制器中的高性能总线。

AMBA ASB 支持处理器、片上存储器和片外存储器接口以及低功耗外设宏单元之间的有效连接。该总线也提供用于模块宏单元测试和诊断的测试下部构造。

4.1.1 基于AMBA ASB的典型微控制器

一个基于AMBA的典型微控制器由一个能够支持外部存储器带宽（包括CPU，片上存储器和其他直接数据存取（DMA）设备）的高性能系统中枢总线，加上一个桥接器连接着较窄的挂接着较低带宽外设的APB总线组成。图 4.1表示了表示了一个包含AHB和APB的典型系统。

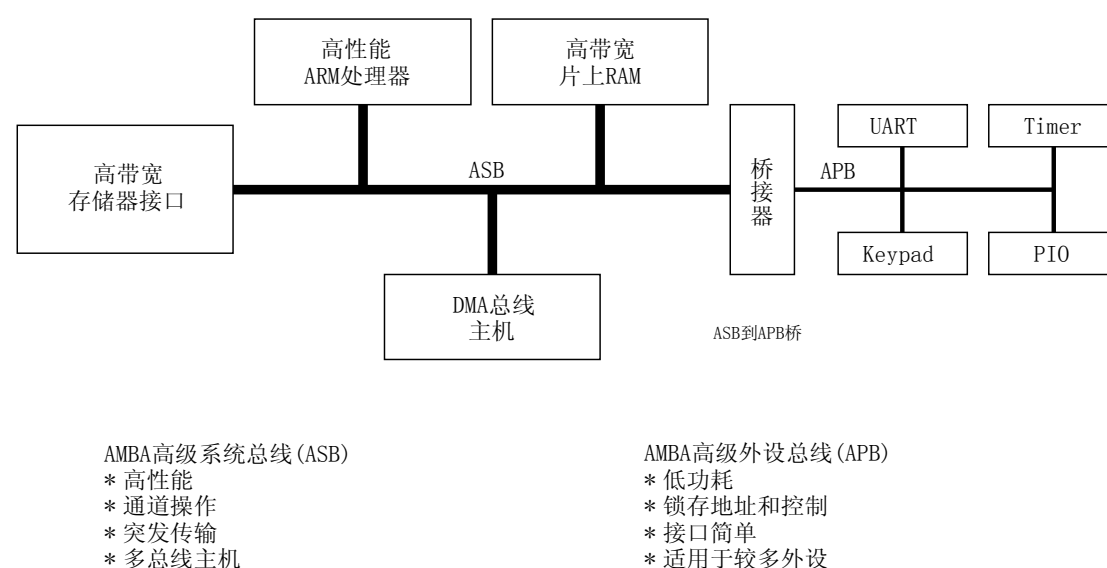


图 4.1 一个 AMBA 的典型系统

外部存储器接口是专用的并且可能有一个较窄的数据通道,但是它支持一种允许内部的 ASB 和 APB 模块与系统独立的测试装置分开测试的测试访问模式。

4.1.2 AMBA ASB和APB

APB 表现为一个局部二级总线,被封装成一个 ASB 的从机。APB 在 ASB 信号的基础上直接为系统总线提供了低功耗的扩展。

APB 桥作为一个从模块处理总线握手并且从局部外设总线的角度控制信号的时序更新。通过从系统总线起始点处定义 APB 接口,可以有效的利用系统诊断和测试方法带来的好处。

4.2 AMBA ASB描述

ASB 是一个高性能的通道总线,支持多总线主机。基本的总线操作流程是:

- 1、 仲裁器决定哪个主机被授予访问总线;
- 2、 当被授予总线,主机发起总线上的传输;
- 3、 译码器使用高位 (MSB) 地址线来选择总线从机;
- 4、 从机提供一个传输响应返回给主机并且在主机和从机之间传输数据。

在 ASB 上有三种传输类型:

- 非连续 (NONSEQUENTIAL) 用作单个传输或者是突发的第一个传输。
- 连续 (SEQUENTIAL) 用在突发传输中。连续传输的地址总是和前一个传输的地址有关。
- 仅地址 (ADDRESS-ONLY) 当没有数据传递的要求时使用。仅地址传输的三个主要用处是空闲 (IDLE) 传输,总线主机移交 (HANDOVER) 周期,以及不负责数据传输的随机地址译码。

图 4.2表示了在处理突发事务中非连续和连续传输的使用。

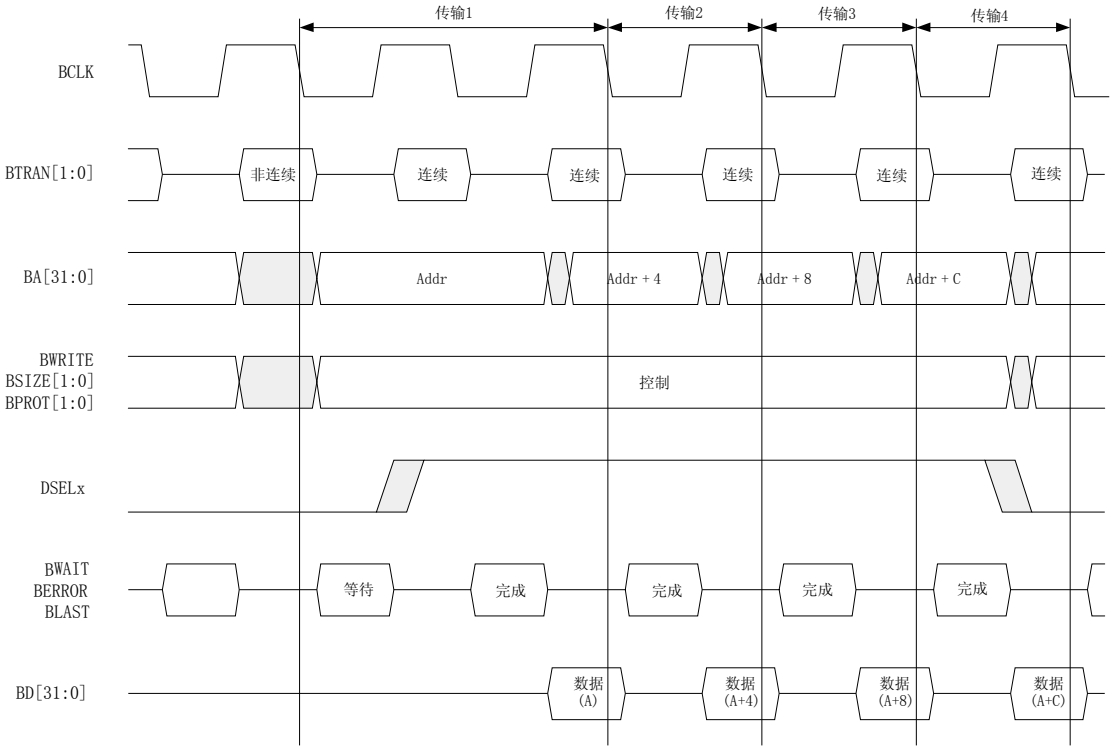


图 4.2 ASB 传输

突发以一个到地址 A 的非连续传输开始。接下来的连续传输是传送到连续地址 $A+4$, $A+8$, 和 $A+12$ 的。

4.3 ASB传输

当主机被授予总线时能够执行下面的传输：

- 非连续数据传输；
- 连续数据传输；
- 仅地址传输。

传输定义为在前一个传输完成之后（由 **BWAIT** 为低表示）的 **BCLK** 下降沿开始，并且一直运行到接收到一个传输完成响应后（也由 **BWAIT** 为低表示）的 **BCLK** 下降沿为止。

总线主机将会执行的传输类型可由在传输发起时的 **BTRAN** 信号的值来决定。在传输中 **BTRAN** 信号将会改变以表示下一个传输的类型。

4.3.1 非连续传输

非连续传输不是在单个传输就是在突发传输的开始处发生。图 4.3 表示了一个包含等待状态典型的非连续读传输。

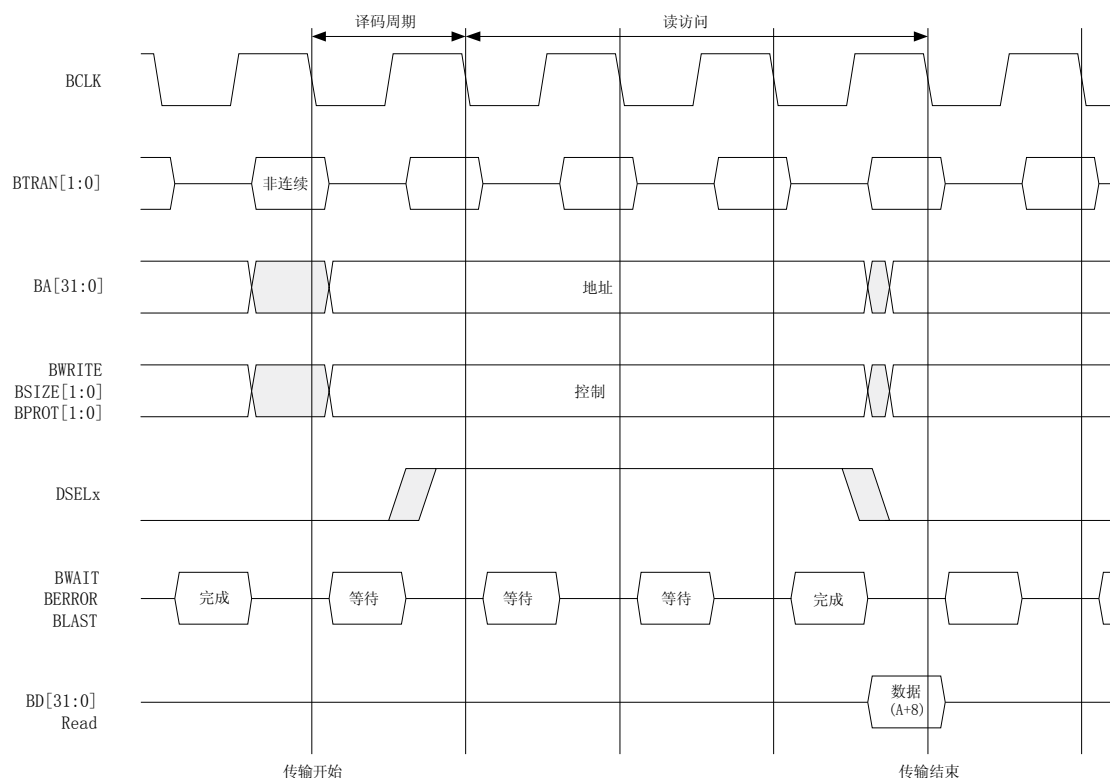


图 4.3 非连续传输

需要注意以下几点：

- 在传输开始之前地址和控制信号在 **BCLK** 为高相位期间开始改变；
- 对于非连续传输一个有效的地址可能会不可用，地址的不可用会一直持续到 **BCLK** 为高电平相位的末尾，甚至是在传输开始前时钟低电平相位的开始处；
- 译码器需要一个稳定的地址以便选择当前从机，会在非连续传输的第一个周期自动插入一个等待状态。这个等待状态被当作一个译码周期并且给译码器提供适当的时

间来检查地址线的高位并且在译码周期的高电平相位期间断言 **DSELx** 中适当的位；

- 对于传输中剩下的周期，从机将提供一个传输响应并且在主机和从机之间将发生数据交换；

注：在特定的系统设计中，比较典型的是那些带有低频率系统时钟的系统，传输开始前地址在 **BCLK** 的高电平相位期间很早就有效了，允许译码器在 **BCLK** 的下降沿之前产生有效的 **DSELx** 信号。这样的系统在非连续传输开始处并不需要一个额外的译码周期并且这类系统中的操作在地址译码中有更详细的描述。

- 数据总线，**BD[31: 0]**，必须在传输结束时 **BCLK** 的下降沿处有效。在一个写周期中，总线主机负责驱动数据总线，在时钟的高电平相位的起始处实行，以便从机可以在时钟的下降沿接收数据。在一个读周期中相应的从机必须驱动数据总线，以便使数据在（时钟的）高电平相位结束处有效；
- 因为有许多不同的总线从机会驱动数据到 **ASB** 上所以需要确认不同的从机在驱动数据到总线上时不会互相交迭。整个非交迭相位由从机提供并且主机不要在非连续传输开始时时钟为低电平相位期间驱动数据；
- 由于许多总线信号是共享的并且在没有活动驱动器时有转换周期，因此需要确认提供总线保持单元以防止浮空电平出现在总线上。

4.3.2 连续传输

当地址和前一个传输相关时发生连续传输。由 **BWRITE**、**BPORT**、和 **BSIZE** 表示的控制信号，将和之前的传输一样。

如果连续传输跟随在一个非连续传输或者另外一个连续传输之后，当前地址就可以用之前的（传输）大小和地址来计算。例如一个字突发访问将会有地址 **A**，**A+4**，**A+8**，而一个半字的突发访问将会有地址 **A**，**A+2**，**A+4**。

如果连续传输跟随在一个仅地址周期之后那么地址将和仅地址周期一样。这种仅地址后跟随着连续传输的组合既允许单个访问使用连续传输也允许突发传输以连续传输开始。图 4.6 表示了使用仅地址后跟随着连续传输的例子。

图 4.4 表示了一个带一个等待状态的连续传输。这近似于一个非连续传输。

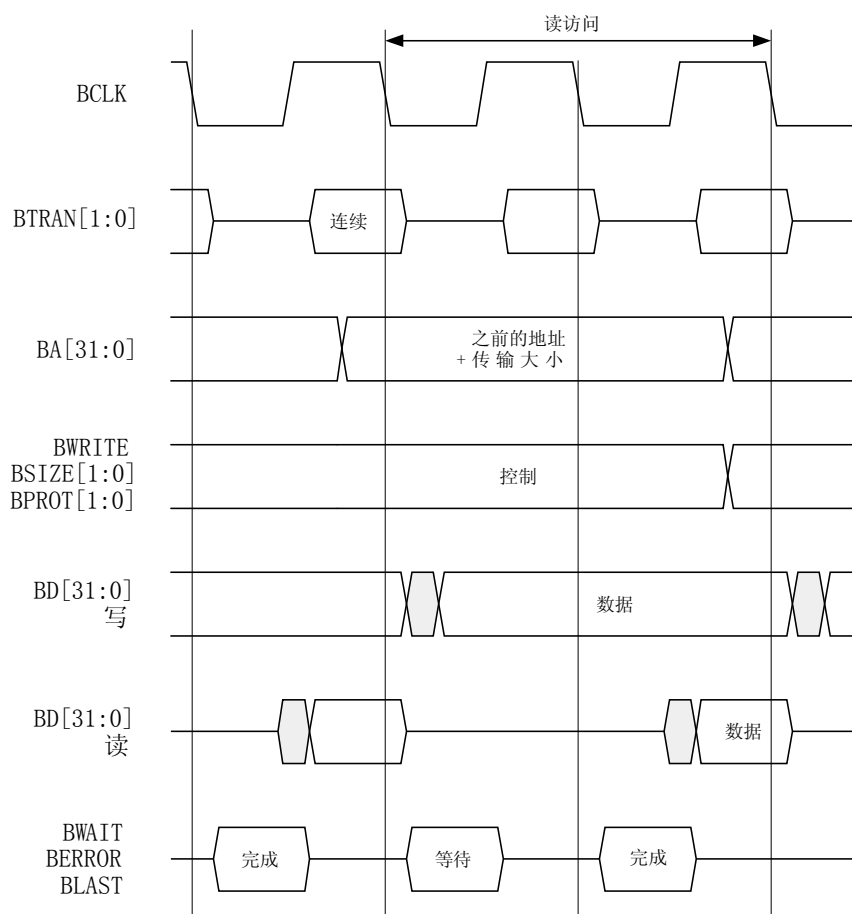


图 4.4 连续传输

主要的不同如下：

- **BTRAN** 信号表示一个连续传输；
- 传输开始时在 **BCLK** 为高电平相位期间地址总是有效的；
- 地址和前一个传输相关；
- 控制信息和前一个传输保持一致；
- 对于写操作而言数据总线在整个传输期间都被驱动。

数据总线，**BD[31: 0]**，能在整个传输期间被驱动，因为和非连续传输的情况不一样，并不需要提供一段时间来允许总线周转。

4.3.3 仅地址传输

仅地址传输表示（传输中）并不要求数据处理。在一个仅地址传输期间地址和控制信息也可能有效，必须被驱动为有效电平的信号仅有：

- **BTRAN** – 表示下一个传输类型；
- **BLOK** – 允许仲裁处理得以继续。

由于仅地址处理并不访问总线上的从机，所以仅要求一个时钟周期并且 **BWAIT** 信号将为低电平。该信号由总线译码器驱动，因为在仅地址传输周期内没有从机被选中。总线主机在不要求总线进行数据传输时可以连续执行许多个仅地址传输。

仅地址传输可以用在三个不同的方式：

- 作为一个确切的空闲周期（当总线主机不请求总线时）；

- 为下一个传输随机地广播一个地址，而不负责传输；
- 在总线主机移交期间提供一个周转的周期。

仅地址传输用作一个确切的空闲周期那么地址和控制信号并不要求在传输中的任何时刻都有效（见图 4.5）。

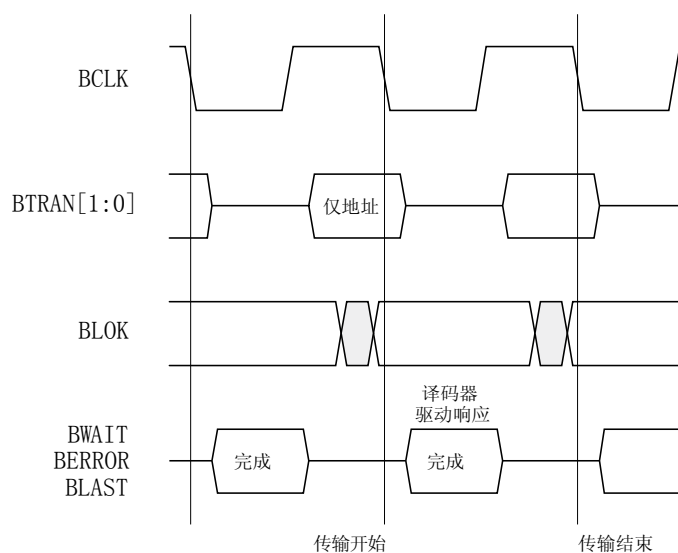


图 4.5 仅地址传输

BLOK 信号是唯一的例外并且在所有仅地址传输期间必须被驱动为有效电平以允许仲裁器处理得以继续。

仅地址传输的第二种用法是为传输随机地广播一个地址，而不负责传输（见图 4.6）。

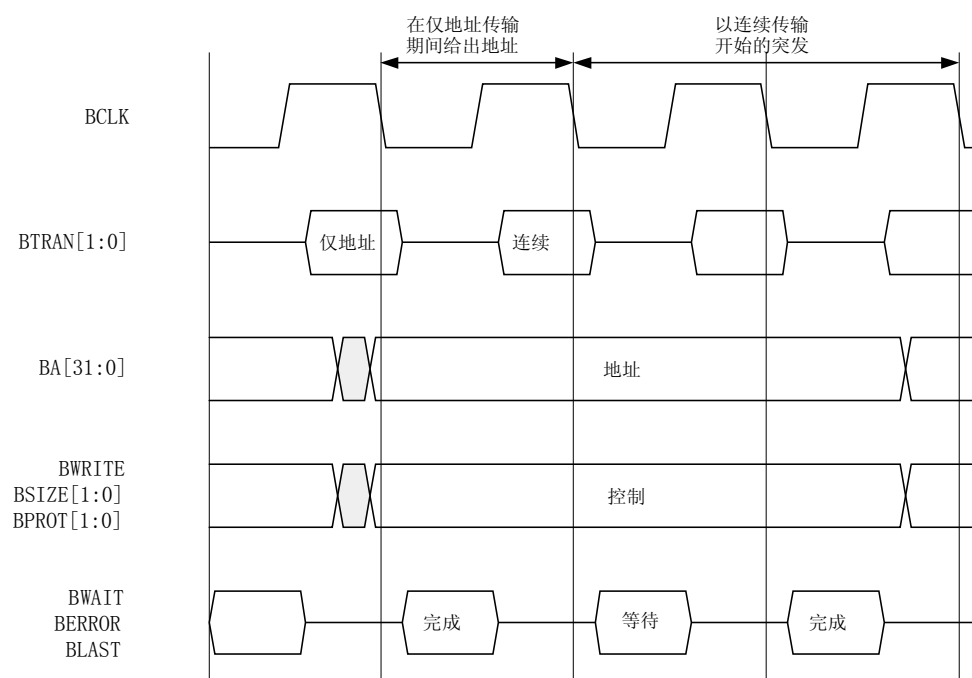


图 4.6 仅地址传输用来启动突发

使用仅地址传输来随机地广播地址允许在仅地址周期中由译码器执行地址译码。如果总线主机之后提交突发那么主机可能用一个连续传输来启动突发，因而去除了传输开始前一个额外的译码周期的需要。

仅地址传输的最后一种用法是在总线主机移交期间提供一个周转周期（见图 4.7）。

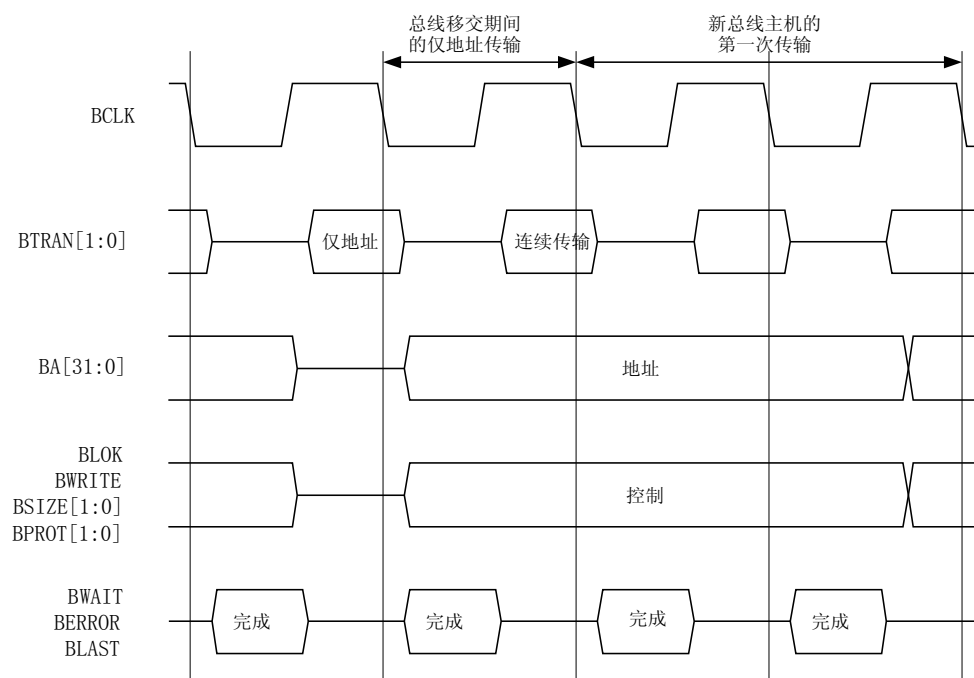


图 4.7 仅地址传输在总线主机移交中的使用

将被授予总线的主机在总线上必须发起一个仅地址传输，并且在这种情况下，新的总线主机并不立刻驱动地址和控制信号，但是在传输的低电平相位期间驱动信号之前提供一个周转相位。

注：在这种情况下，地址和控制信息将不再有效直到 **BCLK** 的低电平相位为止。

4.4 地址译码

在基于 ASB 的 AMBA 系统中地址译码由一个中央译码器来执行。

译码器使用每次传输的类型以确定执行以下功能之一：

- 对于仅地址传输译码器将会响应一个完成（DONE）传输响应并且没有从机会被选中。在仅地址传输期间译码器随机地执行一个地址译码如果仅地址传输之后立刻跟随着一个连续传输；
- 对于非连续传输（或者前一个传输被一个末尾（LAST）处理响应终止）译码器将会在传输发起前插入一个等待状态以允许有足够的时间来进行地址译码（尽管并不是所有的系统都要求这个额外的等待状态）。

译码器插入的额外的等待状态被当作一个译码周期（DECODE Cycle）并且在译码周期内没有选择信号，**DSELx**，被断言。

在传输的第二个周期译码器将不是选择合适的从机就是提供一个错误（ERROR）传输响应。

错误传输响应将在以下情况被提供：

- 在当前地址传输中不存在从机；

- 到受保护存储区域的传输；
- 传输的对齐形式不被存储器系统支持。

在大多数有效传输的常规情况下，译码器将断言相应的从机选择信号 **DSEL_x** 并允许被选中的从机为剩下的传输周期提供响应；

- 对于连续传输响应译码器将断言合适的 **DSEL_x** 信号并且被选中的从机提供传输响应。译码器并不需要译码地址因为这已经在前一个非连续或者仅地址传输就被执行了。

因为在连续传输中译码器并不执行地址译码所以如果传输将要跨越存储器边界时则需要从机提供一个末尾响应。当译码器检测到连续传输将跨越存储器边界时译码器也要负责产生一个内部译码的末尾 (**LAST**) 信号。

在非连续传输中插入的译码周期可以被用来提高系统的性能。在典型的设计中地址译码的时间要求将会增加到从机的访问的临界途径并且经常导致需要一个额外的等待状态。译码器可以被用来减少这种频繁的消费，仅通过自动的在非连续传输期间插入一个译码周期，但允许连续传输不用额外的等待的状态就能完成传输。

在一些系统中，比较典型的是那些带较低时钟频率的系统，地址译码并不要求额外的等待状态并且在这样的系统中译码器可以被简化，这样连续和非连续传输都可以不用额外的译码周期就可以发出。

4.5 传输响应

由总线主机指示的每次传输必须产生一个响应并且该响应不是由译码器就是由被选中的从机提供。传输响应通过使用 **BWAIT**、**BERROR** 和 **BLAST** 信号来提供，这些信号在时钟的低电平相位期间被驱动。

图 4.8 表示了插入三个等待状态以扩展传输的传输响应是如何使用的例子。

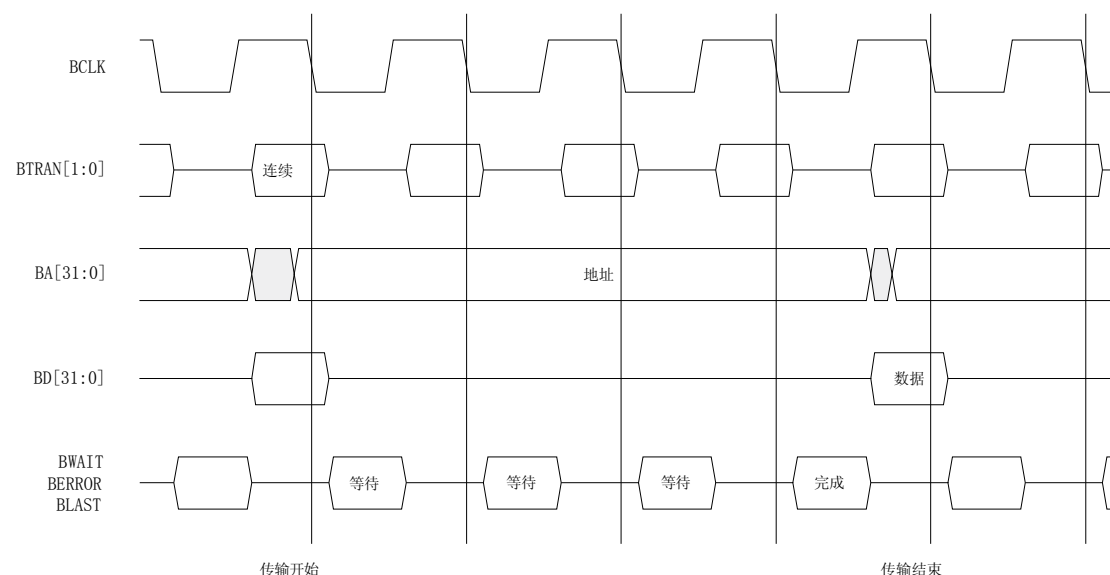


图 4.8 传输响应

下面的传输响应是可用的：

- WAIT** 传输在完成之前必须被扩展；
- DONE** 传输已经成功完成；

- ERROR** 传输已经完成，但是发生了一个错误。错误条件应该被总线主机采样因此要注意传输并没有成功；
- LAST** 传输已经成功完成，但是从机不能接受下一步的突发传输或者到达了存储的边界。该响应对主机而言和等价于 **DONE** 响应，但是指示译码器必须在发起下一次传输时插入一个译码周期；
- RETRACT** 传输响应并没有完成，因此总线主机应该重试传输。撤消（**RETRACT**）响应可以被从机用来发信号给总线主机表明传输可以完成，但是这可能会消耗许多总线周期。

使用 **RETRACT** 响应阻止总线被可能需要较长时间才能完成的传输锁定并且释放给更高优先级的总线主机使用。

和其他单周期响应的编码不同，**RETRACT**响应是一个两步响应，如图 4.9。

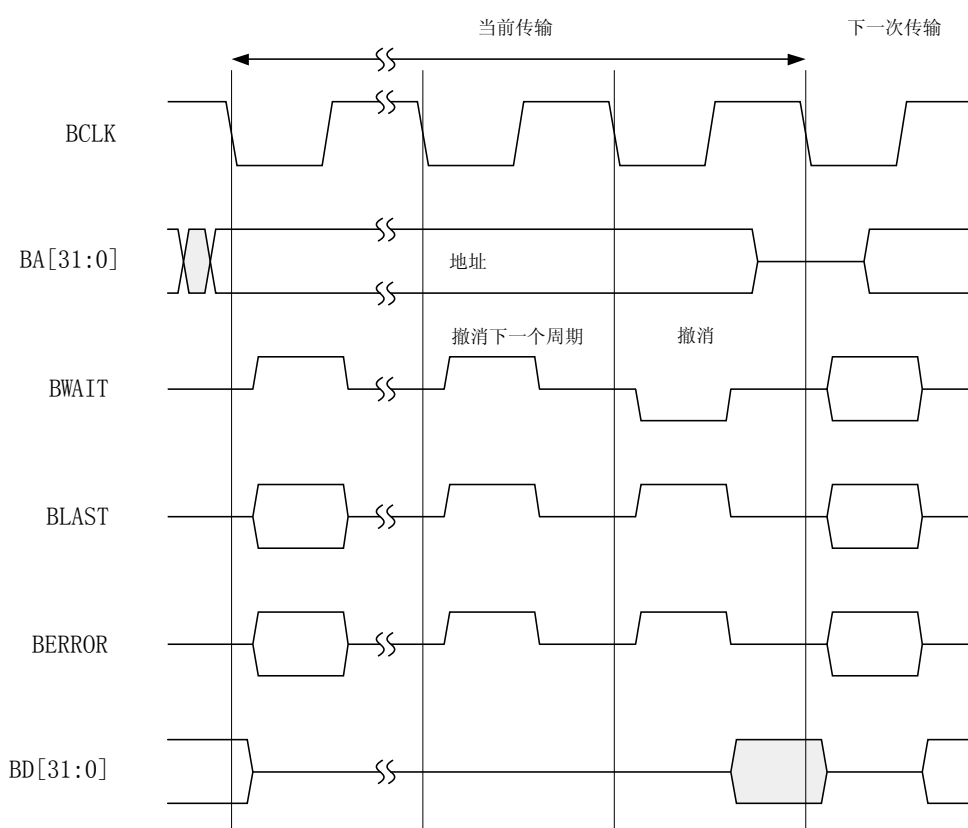


图 4.9 撤消响应

以下要点需要注意：

- 在 **RETRACT** 将要发生时从机给出总线主机信号的第一步，使用 **RETNEXT** 响应（**BWAIT**、**BLAST** 和 **BERROR** 全部为高）；
- 在第二步当从机提供 **RETRACT** 响应时传输完成（**BWAIT** 为低，**BLAST** 和 **BERROR** 都为高）。

这个两步响应给总线主机提供适当的警告，当 **BWAIT** 信号为低时主机不应该考虑传输已经完成。

所有的总线主机必须支持 **RETRACT** 机制，然而不是所有从机都要求能执行 **RETRACT** 响应。典型的，一个 **RETRACT** 响应将仅由没有短暂的保证完成时间的从机提供，因此可能会死锁总线相当一段时间。

对大多数传输，响应将由被选中的从机提供，但是译码器提供响应应当：

- 传输是仅地址（传输）；
- 传输是到没有总线从机的存储空间；
- 到受保护的存储区域的时发生访问违规。

4.6 多主机操作

AMBA 总线规范在高性能的 **ASB** 上支持多总线主机。在仲裁器和每个总线主机之间执行简单的两线请求和授予机制。仲裁器确保只有一个总线主机在总线上是有效的并且确保没有主机请求总线时授予一个默认主机总线。

规范也支持共享的锁定信号。这允许总线主机表示当前传输不应该和接下来的传输分开并且将阻止其他总线主机获取访问总线直到锁定传输已经完成。

有效率的仲裁机制在减少总线上主机相继有效之间的死区时间(**dead-time**)显得很重要。总线协议支持通道仲裁，这样下一次传输的仲裁可以在当前传输就被执行。

定义了仲裁协议，但是优先权顺序是固定的并由应用决定。然而，典型地，测试接口应该被分配最高优先级以确保在所有情况下的测试访问。每个系统也必须包含一个默认的总线主机，（用来）在没有总线主机请求总线时被授予总线。

从每个总线主机到仲裁器的请求信号，**AREQ_x**，表示总线主机请求总线。从仲裁器到总线主机的授予信号，**AGNT_x**，表示该总线主机是当前请求总线的主机中优先级最高的。

总线主机：

- 当 **AGNT_x** 为高时必须在 **BCLK** 的高电平相位期间驱动 **BTRAN** 信号；
- 当 **AGNT_x** 为高并且 **BWAIT** 为低时在 **BCLK** 的上升沿被授予总线。

共享的锁定信号，**BLOK**，指示仲裁器接下来的传输和当前传输是不能分割的并且其他主机不应该被允许访问总线。

当被授予总线后以确保仲裁进程可以继续时总线主机必须一直驱动 **BLOK** 信号为有效电平，即使总线主机不执行任何传输。

4.6.1 仲裁器

仲裁器的功能如下：

- 1、总线主机在 **BCLK** 的高电平相位期间断言 **AREQ_x**；
- 2、仲裁器在 **BCLK** 的下降沿采样所有的 **AREQ_x** 信号；
- 3、在 **BCLK** 的低电平相位期间仲裁器也采样 **BLOK** 信号并在之后断言对应的 **AGNT_x** 信号：
 - 如果 **BLOK** 为低，那么仲裁器将（总线）授予最高优先级的总线主机；
 - 如果 **BLOK** 为高，那么仲裁器将保持当前总线主机被授予（总线）。

仲裁器可以在每个总线周期更新授予信号。然而，新的总线主机只有在当前传输完成时（由 **BWAIT** 为低表示）才能被授予并开始驱动总线。因此，在等待传输过程中潜在的下一个总线主机可能会变更。

BLOK 信号在两个不同总线主机移交（总线）的单个时钟周期中被仲裁器忽略。

如果没有总线主机请求总线那么仲裁器必须授予默认总线主机（总线）。

定义了仲裁协议，但是优先权顺序是固定的并且留给应用处理。可以使用一个简单的固定优先级方案。当然，如果应用需要一个更加复杂的方案也可以被执行。

4.6.2 总线主机移交

当一个总线主机不是当前被授予总线的（主机），而是另一个新的总线主机被授予（总线）时，总线主机移交发生。

总线主机在 **AGNTx** 为高并且 **BWAIT** 为低时被授予总线。**AGNT** 为高表示该总线主机是当前请求总线的主机中优先级最高的而 **BWAIT** 为低表示前一个传输已经完成。

图 4.10表示了总线移交的进程。

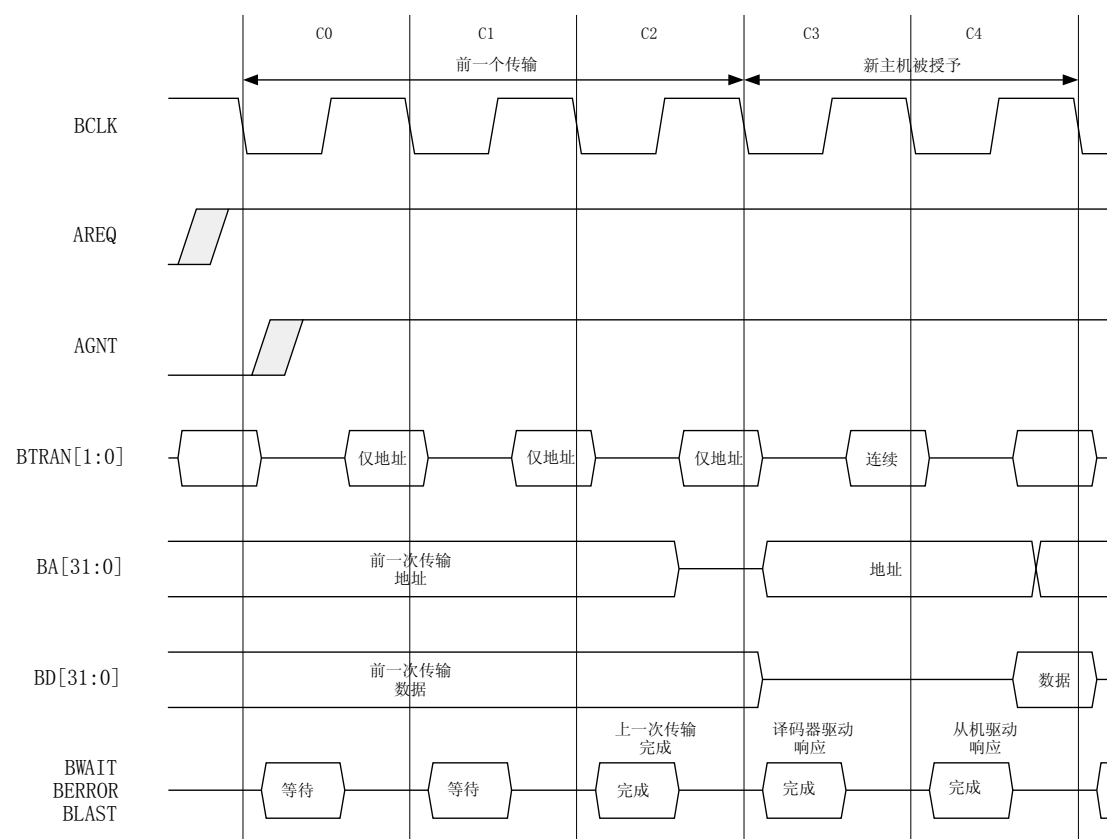


图 4.10 总线主机移交

下面的要点必须注意：

- 当 **AGNTx** 被断言时总线主机必须在 **BCLK** 为高时驱动 **BTRAN** 信号。如果在等待前一次传输这可能会持续许多周期；
- 在移交之前 **BTRAN** 必须表示一个仅地址周期因为新的总线主机必须从仅地址周期开始以允许总线周转；
- 当前一个传输完成时新的总线主机将被授予总线；
- 在前一个传输的最后一个时钟的高电平相位期间地址总线将被前一个总线主机停止驱动；
- 新的总线主机在时钟的低电平相位期间开始驱动地址总线和控制信号。第一个传输可能会从接下来的总线周期开始。

在等待传输期间，总线主机移交可能会被延时并且如果另外一个更高优先级的总线主机也在当前传输完成前请求总线，到特定总线主机的 **AGNTx** 信号可能会被断言然后被撤销。

4.6.3 默认总线主机

每个系统必须被设计为带有一个单一的默认总线主机，该主机将在没有其他总线主机请求总线时被授予总线。默认总线主机负责驱动下列信号以确保总线保持有效：

- **BTRAN** 必须被驱动以表示仅地址传输；
- **BLOK** 必须被驱动为低。

4.6.4 锁定传输

总线主机不要尝试执行到一个能够给出 **RETRACT** 响应的从机的锁定传输。有两点原因：

- 总线可能会被保持锁定许多周期；
- 如果 **RETRACT** 在锁定传输序列的最后一个传输发生，那么仲裁器可能会在传输完成前改变总线的所有权给下一个主机，因此传输序列中最后的传输将不会被锁定。

4.7 复位操作

复位信号，**BnRES**，为低电平有效并且可能会被异步地断言以保证总线处在安全状态下。在复位期间总线上出现下列动作：

- 仲裁器授予默认总线主机总线；
- 默认总线主机必须：
 - 驱动 **BTRAN** 以表示仅地址传输；
 - 驱动 **BLOK** 为低以允许仲裁。
- 所有总线主机三态共享总线信号；
- 译码器必须：
 - 取消断言所有的从机选择信号，**DSELx**；
 - 提供合适的传输响应；
- 所有从机三态共享总线信号。

4.7.1 退出复位

图 4.11表示了退出复位顺序的一个例子。

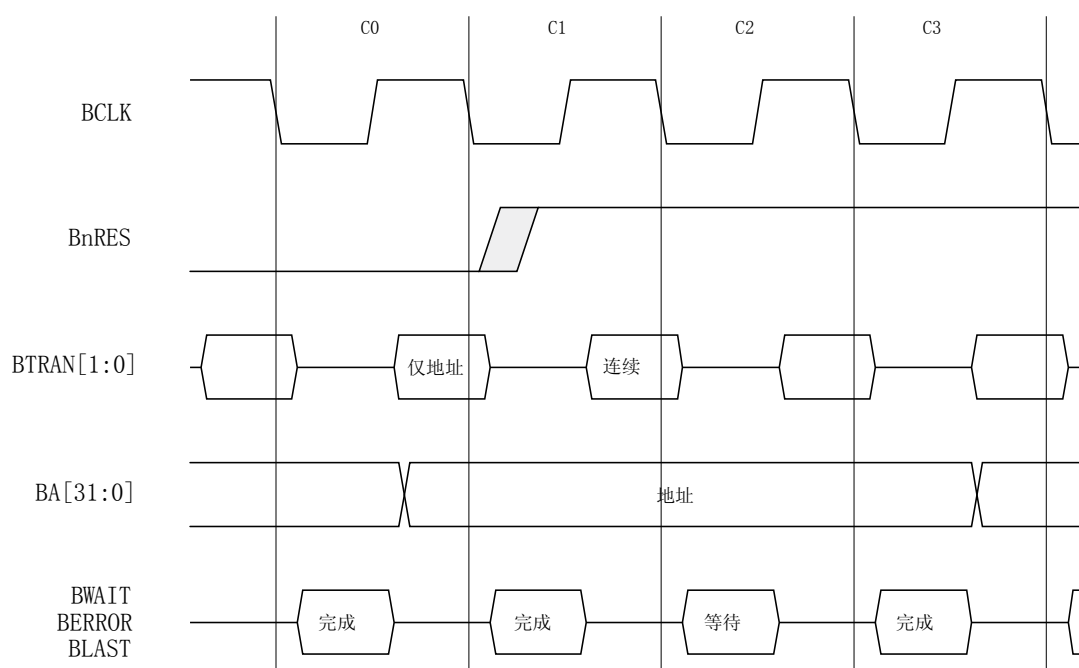


图 4.11 退出复位

下面的要点必须注意：

- 在周期 C1 中时钟的低电平相位期间 **BnRES** 被撤消断言；
- 在周期 C1 中时钟的高电平相位期间默认总线主机可以驱动 **BTRAN** 信号以表示它希望发起一个传输；
- 传输将在周期 C2 发起并且，如例中所示，传输等待并在周期 C3 继续。

4.8 ASB信号描述

这部分提供了更多的关于全部 AMBA ASB 信号的信息，包括它们的使用目的和相位精度级的时序要求。

需要确保提供一个总线保持单元以阻止浮空电平出现在总线上。这可能是因为许多总线信号是共享的，并且在没有活动驱动器时会有周转周期。

4.8.1 时钟

BCLK 是基本的时钟，用来给所有总线传输提供时序。时钟的两个边沿都被使用。

4.8.2 复位

支持低电平有效的复位信号，**BnRES**，以用来复位总线。复位信号可以在任意一个时钟相位期间被异步断言为低电平，但是该信号总是在 **BCLK** 的低电平相位期间失效。

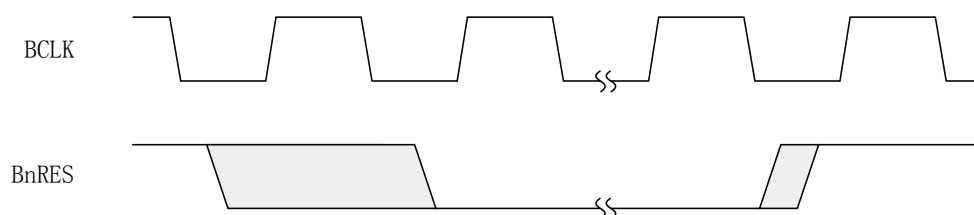


图 4.12 复位信号

在复位期间总线上出现下列动作：

- 仲裁器授予默认总线主机；
- 默认总线主机必须：
 - 驱动 **BTRAN** 以表示仅地址传输；
 - 驱动 **BLOK** 为低以允许仲裁。
- 所有其他总线主机三态共享总线信号；
- 译码器必须：
 - 撤消断言所有从机选择信号，**DSELx**；
 - 提供合适的传输类型。
- 所有从机三态共享总线信号。

BnRES 信号可以在超时情况期间被用来复位总线。

大多数总线主机和从机将用到 **BnRES** 信号来复位总线接口和主要的核心单元。然而，对一些系统单元，例如实时时钟，建议仅使用 **BnRES** 来复位总线接口。这样的系统单元典型的有一个二级复位输入以允许在初始上电和测试目的时复位核心组件。

4.8.3 传输类型

在传输发起前总线主机使用 **BTRAN[1: 0]** 指示是哪种类型的传输。可以设置下列类型的传输：

- 仅地址（ADDRESS-ONLY）；
- 非连续（NONSEQUENTIAL）；
- 连续（SEQUENTIAL）。

表 4.1 表示了 **BTRAN[1: 0]** 信号的编码：

表 4.1 BTRAN 编码

BTRAN[1: 0]	传输类型	描述
00	仅地址	当没有数据传输要求时使用。仅地址传输的三个主要用途是： <ul style="list-style-type: none"> ● 空闲周期； ● 总线移交周期； ● 不负责数据传输的随机地址译码
01	-	保留
10	非连续	用于单个传输或者一次突发的第一个传输。传输的地址和上一次总线访问的地址是无关的。
11	连续	在一个突发中用作连续传输。一个连续传输的地址总是和上一次传输相关。

从表中可以推导出 **BTRAN[1]** 可以用来确定下一个周期是否需要一个数据传输。

当 **AGNTx** 输入为高时 **BTRAN** 信号由总线主机在 **BCLK** 的高电平相位期间驱动（见图 4.13）。

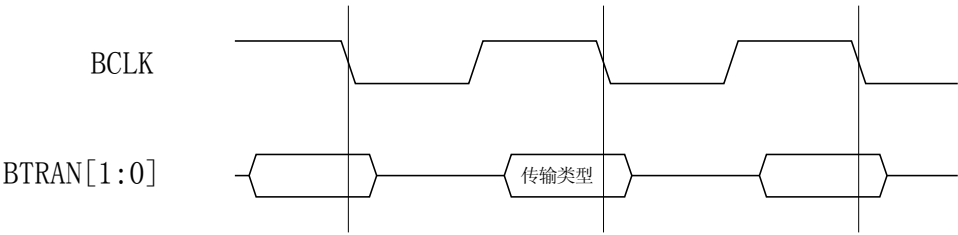


图 4.13 BTRAN 时序

在一个多主机系统中，在一个扩展的传输期间驱动 **BTRAN** 的总线主机可能会变更。因此，**BTRAN** 必须只能在前一次传输已经完成后（由 **BWAIT** 为低表示）才能被认为有效。

4.8.4 地址和控制信息

地址和控制信号如下：

- 地址总线 – **BA[31: 0]**;
- 传输方向 – **BWRITE**;
- 传输大小 – **BSIZE[1: 0]**;
- 保护信息 – **BPROT[1: 0]**。

4.8.5 地址总线

32 位的地址总线，**BA[31: 0]**，提供传输的地址信息。所有传输都经过存储器映射因此系统中所有的外设和存储器必须有一个它们被访问时所在的地址范围。译码器使用地址总线（通常是高位）来确定哪个总线从机被访问。

4.8.6 传输方向

BWRITE信号用来表示传输的方向（见表 4.2）。当**BWRITE**为低传输是一个读访问而当**BWRITE**为高时传输是一个写访问。

表 4.2 BWRITE 编码

BWRITE	传输方向
0	读传输
1	写传输

4.8.7 传输大小

BSIZE[1: 0]编码传输大小（见表 4.3）。字节，半字和字（传输大小）都被定义，而最后一个编码保留作将来使用。

表 4.3 BSIZE 编码

BSIZE[1: 0]	传输宽度
00	字节（8 位）
01	半字（16 位）
10	字（32 位）

续上表

BSIZE[1: 0]	传输宽度
11	保留

当执行比数据总线要窄的传输时，例如一个字节或者半字传输，总线主机可以通过总线复制数据，使总线主机成为更有效率的双端结构。当响应读周期时，典型的从机不会复制数据到总线上因此主机“认为”数据出现在和从机所驱动的相同的字节通道上显得很重要。

4.8.8 保护信息

总线主机可以使用**BPROT**信号来提供关于它正在执行的传输的附加信息（见表 4.4）。该信息最初打算是由译码器在作为一个总线保护单元时使用的并且大多数总线从机将不使用这些信号。

表 4.4 BPROT 编码

BPROT[1: 0]	传输特权
- 0	预取指
- 0	数据访问
0 -	用户访问
1 -	特权访问

4.8.9 地址和控制信号时序

地址和控制信息由总线主机在 **BCLK** 的上升沿产生。然而，对于非连续和连续传输类型，地址和控制信息的时序被认为是分开的。这是因为典型的总线主机在各自情况下有明显不同的时序参数。

对于连续传输总线主机有快速的地址和控制输出有效时序是普遍的特征，如图 4.14。这是因为总线主机在传输发起前通常能够很好地产生连续地址，因此从总线主机输出的有效时间将主要取决于驱动新的（地址）值出现在总线上的时间。

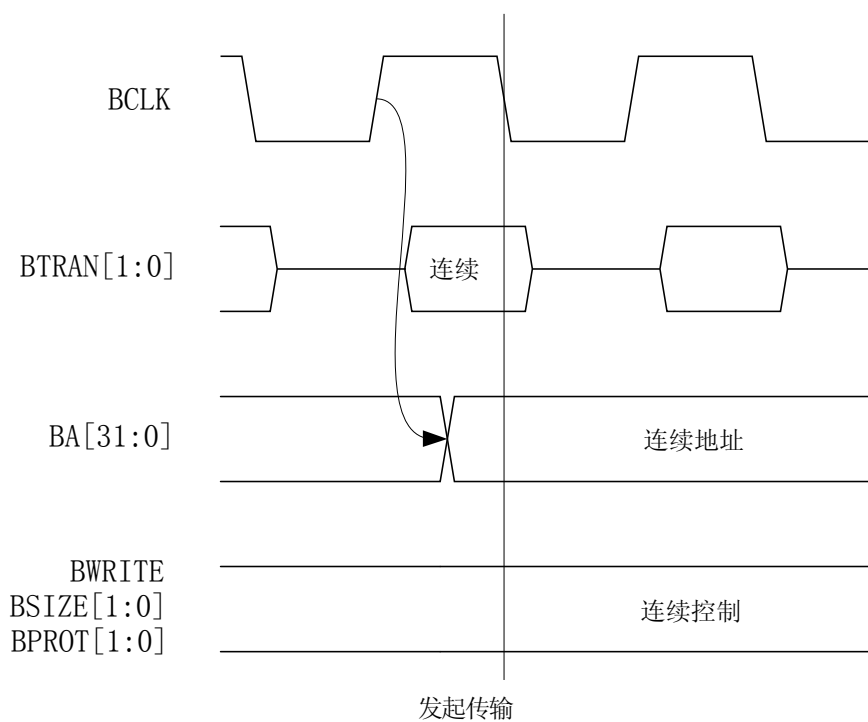


图 4.14 连续地址和控制时序

对于非连续传输，相对于连续传输而言通常总线主机将有明显变慢的地址和控制信号输出有效时间并且这些如图 4.15所示。

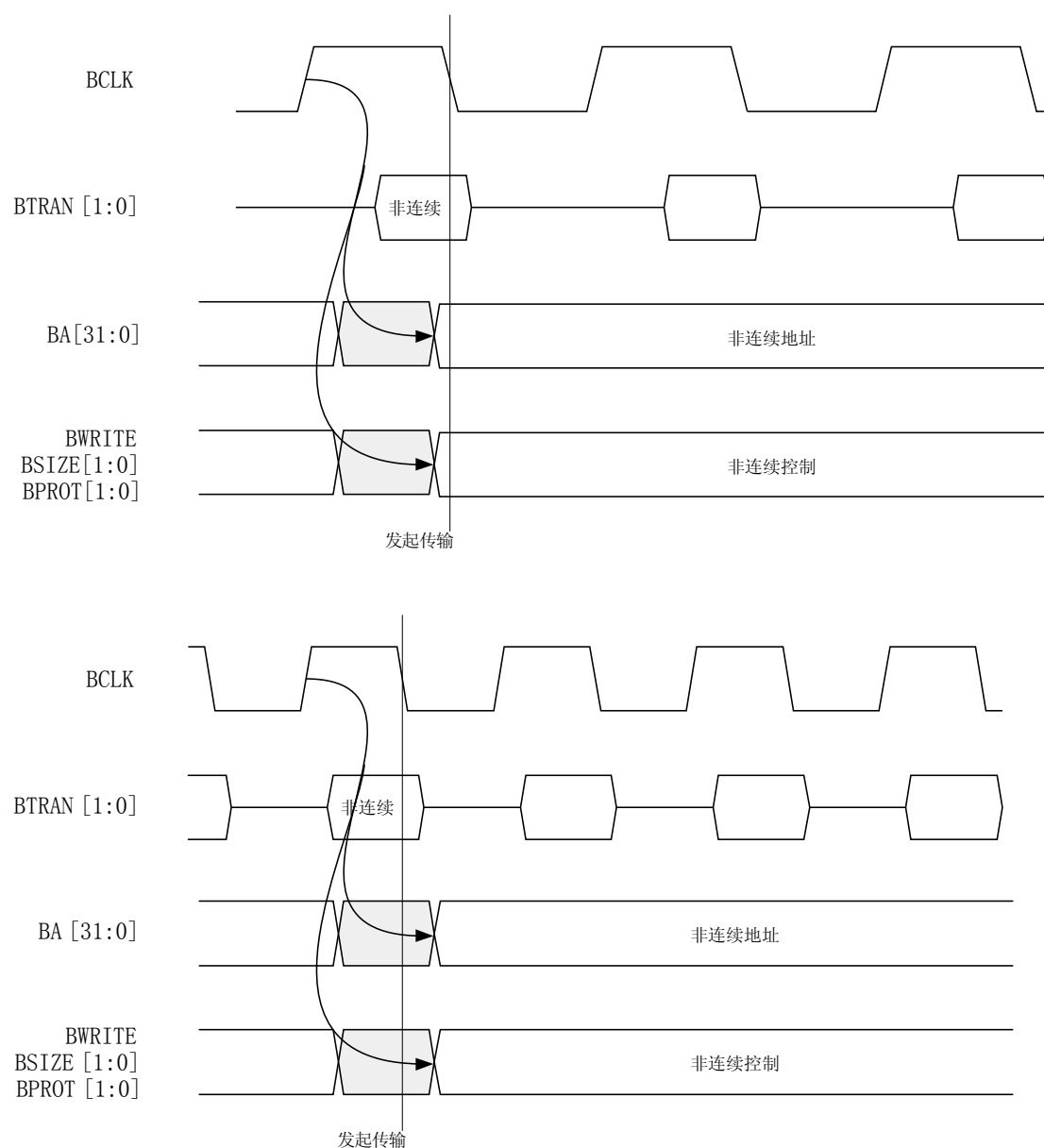


图 4.15 在低频和高频时钟下非连续传输的地址和控制信号

在时钟频率最大可能的系统中，通常地址和控制输出有效时间大于一个时钟相位，这导致传输开始时地址在**BCLK**的低电平相位之前不会有效，如图 4.15所示。

对于仅地址传输地址和控制信息是无效的。在仅地址后立刻跟着一个连续传输的特定情况下，如图 4.16所示，总线主机在仅地址传输期间产生地址和控制信息，这样在连续传输开始前这些信息在**BCLK**的整个高电平相位期间都有效。

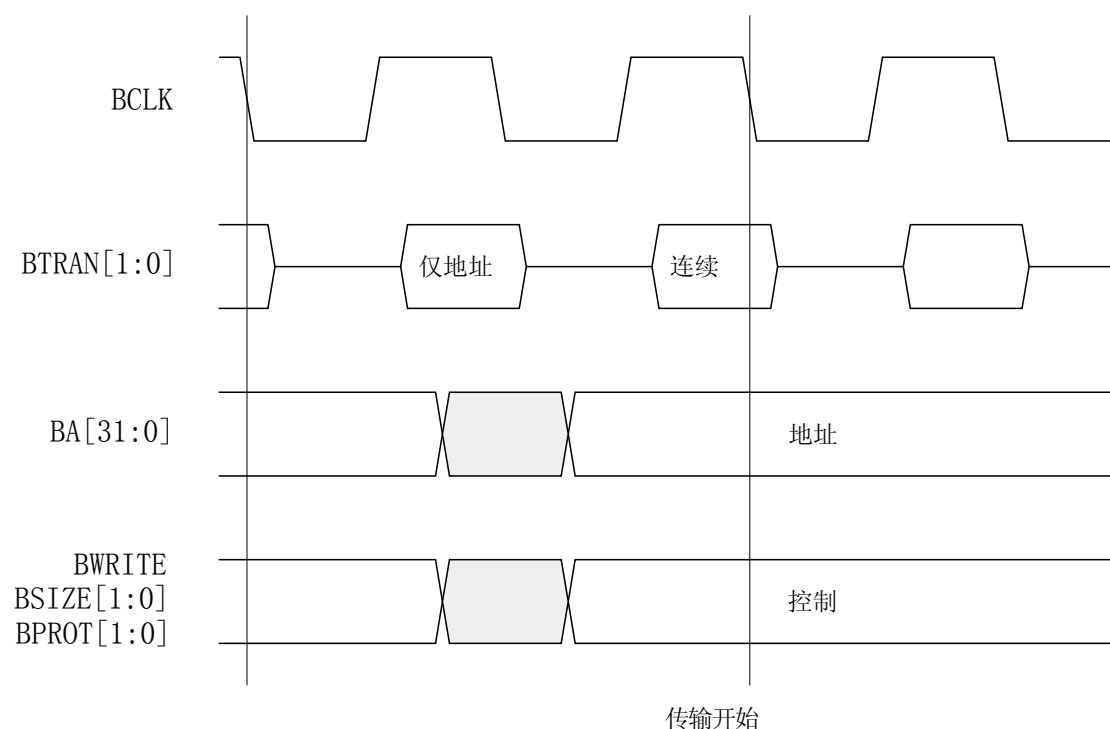


图 4.16 仅地址后跟着连续传输的地址和控制时序

4.8.10 地址和控制信号的三态使能

当总线主机被授予总线时可以仅驱动地址和控制信号。为允许一段时间的总线周转周期，当一个总线主机是第一次被授予总线时它不会在第一次传输之前驱动**BCLK**的高电平相位。作为代替，总线主机必须总是以一个仅地址传输来发起一段时间的总线所有权并且地址和控制信号不被驱动直到仅地址传输的**BCLK**低电平相位时（见图 4.17）。

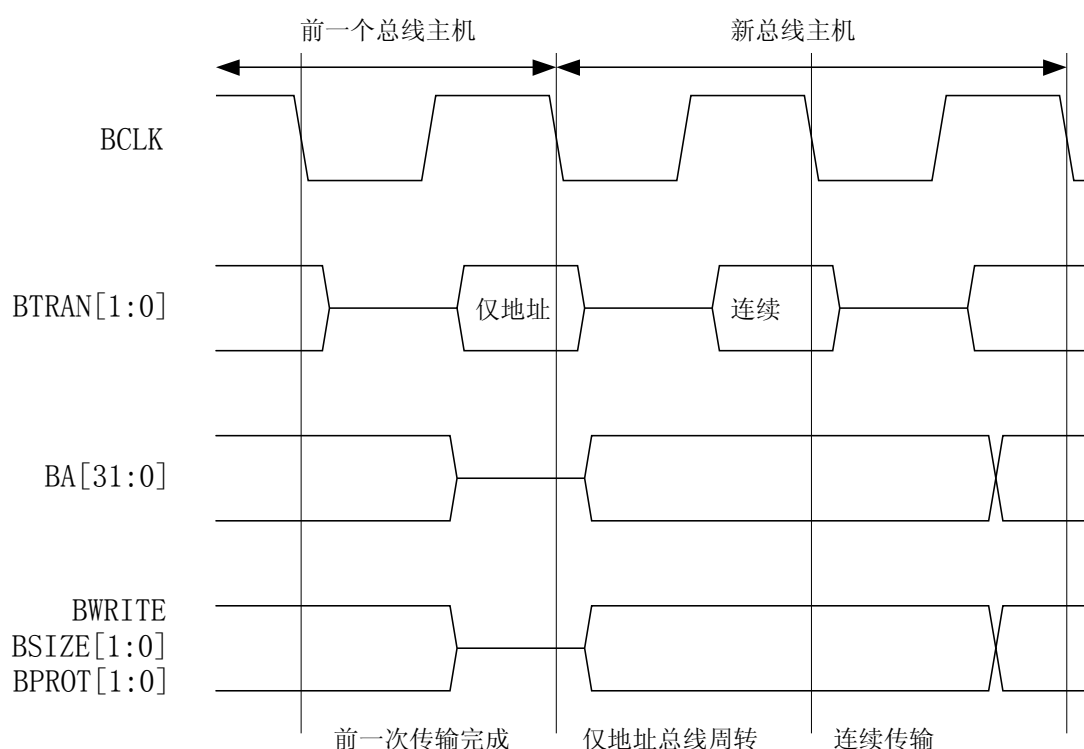


图 4.17 总线主机移交期间的地址和控制信号

对于多主机系统：

- 当主机第一次被授予（总线），当主机在总线上有效时信号并不驱动直到第一个 **BCLK** 的低电平相位时；
- 当总线主机被授予总线时它在 **BCLK** 的两个相位期间都驱动地址和控制信号；
- 总线主机在 **BCLK** 的最后一个高电平相位期间停止驱动信号，当主机失去总线的主机权限时。

4.8.11 从机选择信号

系统中每个 ASB 从机有一个 **DSEL** 选择输入信号。这个信号表示该从机负责提供一个传输响应和要求的数据传输。信号名称 **DSEL_x** 被用来表示到从机 **X** 的 **DSEL** 信号。

在 ASB 上每个从机只有一个 **DSEL_x** 信号并且这些信号由译码器产生。在一个传输中只有一个 **DSEL_x** 信号有效并且当 **DSEL_x** 无效时可能会持续一些周期，例如在仅地址传输中。

在传输发起前 **DSEL_x** 在 **BCLK** 的高电平相位期间改变并在传输中保持有效。它（**DSEL_x**）将在之后跟随着 **BWAIT** 为低的传输响应的 **BCLK** 高电平相位期间转换为下一次传输。

当设计一个系统时在实现一个 ASB 译码器时有两种选择：

- 带译码周期的译码器；
- 不带译码周期的译码器。

这种选择在设计阶段固定并将建立在一个系统的时序分析的基础上。总体上，一个运行在处理器最高频率下的系统将要求译码周期。仅在运行在频率明显低于可能的最高频率下的系统中才可以不需要译码周期。

带译码周期的译码器

在一个高时钟频率系统中译码器的关键通道、在单时钟相位内的地址和从机选择趋向于限制最大的总线时钟速度。在这样的系统中译码器可以在每个非连续传输的开始时被用来自动插入一个等待状态，或者译码周期。这种操作允许连续传输继续运行而没有一个额外的等待状态，因为众所周知，地址译码的关键通道可以在连续传输中被避免，因此导致了全面实现总线带宽。

在译码周期对于非连续传输DSELx在BCLK的高电平相位期间被断言，如图 4.18所示。

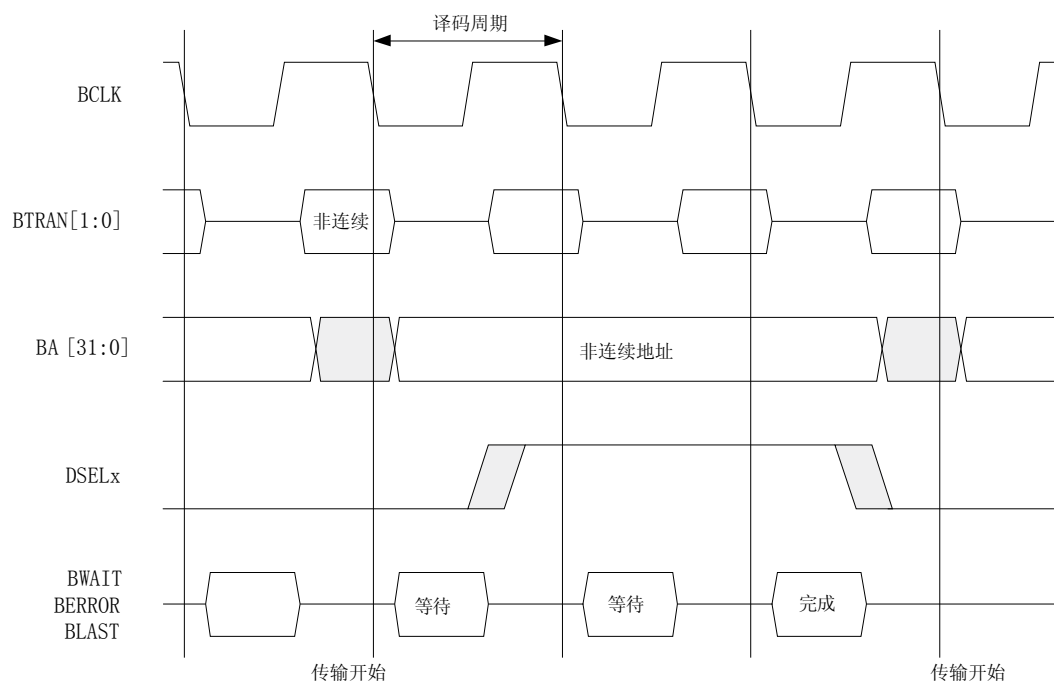


图 4.18 带译码周期的选择信号时序

- 当译码周期在执行 **DSELx** 的时序时仅取决于 **BTRAN[1:0]**并且地址和控制信号没有时序上的依赖性。这是因为仅地址传输中没有 **DSELx** 信号被断言；
- 对于一个非连续传输译码周期被插入以给地址和控制信号提供一个相位来变得有效；
- 对于连续传输使用来自前一个周期的地址和控制信息。

不带译码周期的译码器

在较低时钟频率的系统中地址和控制信息将在单个时钟相位内译码地址和从机选择时有效。这样的系统中译码周期并不需要（见图 4.19）。

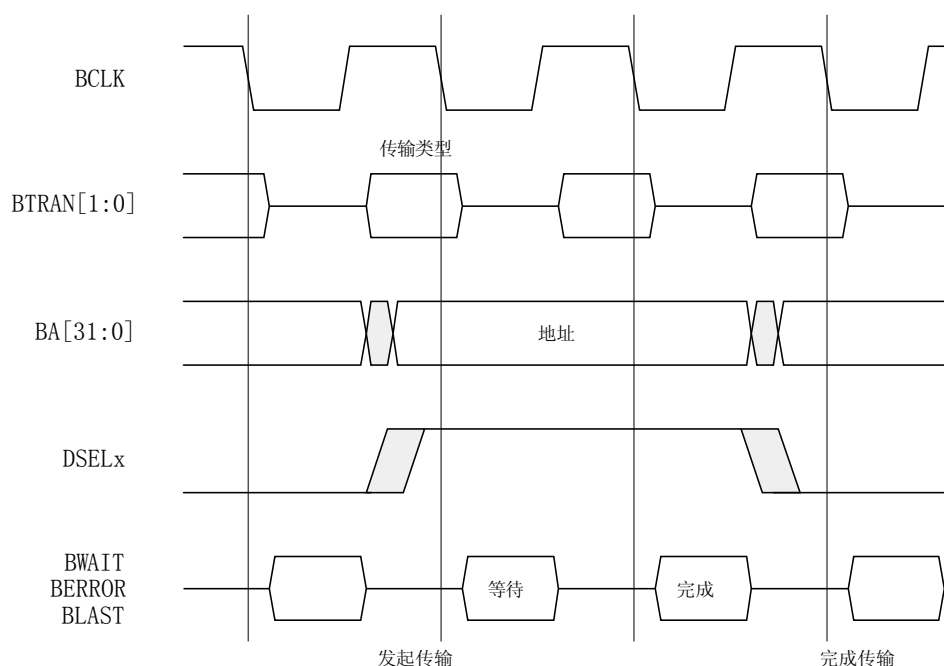


图 4.19 不带译码周期的选择信号时序

在传输开始前选择信号在 **BCLK** 的高电平相位期间生效并且在整个传输期间保持有效，直到最后一个周期的高电平相位时为止。

当译码器没有插入译码周期时，**DSELx** 的时序取决于由当前授予的总线主机产生的地址和控制信号的时序。

4.8.12 传输响应

传输响应信号由从机设备使用以表示一个传输的状态。

有效的传输响应必须在 **BCLK** 的低电平相位期间被提供。一个从机无论何时被选中（由 **DSELx** 被断言来表示），这个从机必须提供响应。当没有从机被选中时，例如在一个仅地址传输期间，响应由译码器提供。

等待响应

BWAIT 被用来表示一个传输可以在何时完成。当从机需要额外的总线周期以完成当前传输时 **BWAIT** 被断言为高电平。**BWAIT** 为低表示传输可以完成。传输是否成功完成只能通过其他传输响应信号来判断。

错误响应

通过 **BERROR** 信号来告知一个错误情况。这可以被用来表示一个失败的传输，一个到没有从机设备的地址的传输或者一个保护错误。

很多简单的总线从机将不会实现错误逻辑因此有一个固定的 **BERROR** 为低的响应。

BERROR 也和 **BLAST** 一起被用来表示一个撤消操作。当这两个信号都为高平时表示要求一个总线撤消。

末尾响应

如果当前传输必须是一次突发的末尾那么 **BLAST** 被用来发出（这个）信号。这可能会被典型的用来阻止一次突发连续传输而超越一个页边界或者其他突发长度限制。

BLAST 被译码器用来确保接下来的传输有和非连续类型传输一样的特征，而非一个突发传输。典型的这包括确保（译码器）有足够的时间来执行一个新的地址译码。

许多总线从机将能接收任何数量的突发访问并且这些从机将给出 **BLAST** 为低的固定响应。

BLAST 和 **BERROR** 一起也被用来表示一个撤消操作。当这两个信号都为高时表示要求一个总线撤消。

总线撤销

在少量等待状态中从机不能保证完成传输会潜在的阻塞总线并阻止高优先级的传输发生。为防止这些从机影响系统的总体延时而提供一种机制允许从机表示传输在当前不能完成，但是（传输）操作可以重试直到它能完成成功完成为止。

一个撤消分两步处理执行，如图 4.20所示。

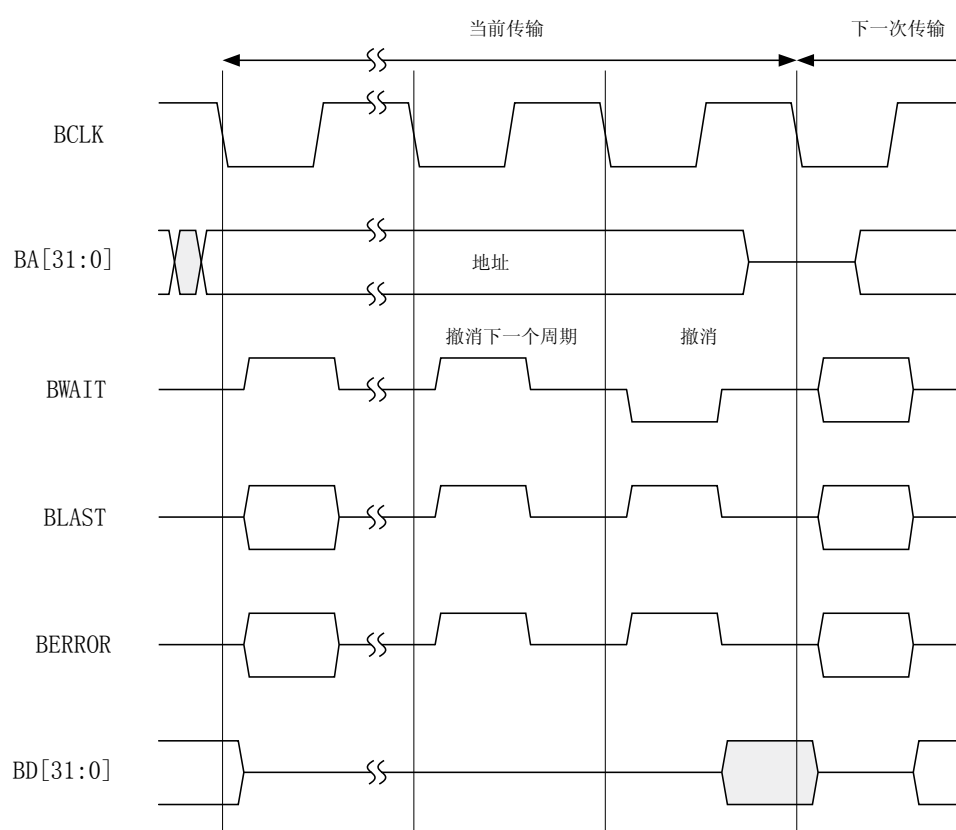


图 4.20 撤消操作

- 首先从机以 **BWAIT**、**BLAST** 和 **BERROR** 都为高来作为响应，表示一个撤消将要发生并且传输将在下一个总线周期结束；
- 在第二个周期传输响应是 **BWAIT** 为低，**BLAST** 和 **BERROR** 都为高。这表示传输被撤消并且总线是空闲的。

有一个保证完成时间的基本从机，并不需要支持总线撤消机制。

响应组合

表 4.5表示了三个从机传输响应信号的组合。

表 4.5 传输响应组合

BWAIT	BLAST	BERROR	状态	描述
0	0	0	DONE	完成，传输成功
0	0	1	ERROR	完成，传输错误
0	1	0	LAST	完成，不能继续突发
0	1	1	RETRACT	完成，总线撤消
1	0	0	WAIT	未完成，插入等待状态
1	0	1	-	保留
1	1	0	-	保留
1	1	1	RETNEXT	总线撤消的下一个周期

为了确保总线保持同步，传输响应必须每个周期都被驱动。在总线传输期间，当一个从机被选中并且它的 **DSELx** 信号被断言，该从机负责驱动传输响应信号。

总线译码器负责下列情况下驱动传输响应：

- 仅地址传输；
- 译码周期；
- 到没有从机被定义的地址空间的传输；
- 到保护区域的传输，且不满足访问许可时；
- 不被存储器系统支持的未对齐传输。

传输响应时序

传输响应信号必须在**BCLK**的上升沿变得有效（图 4.21）

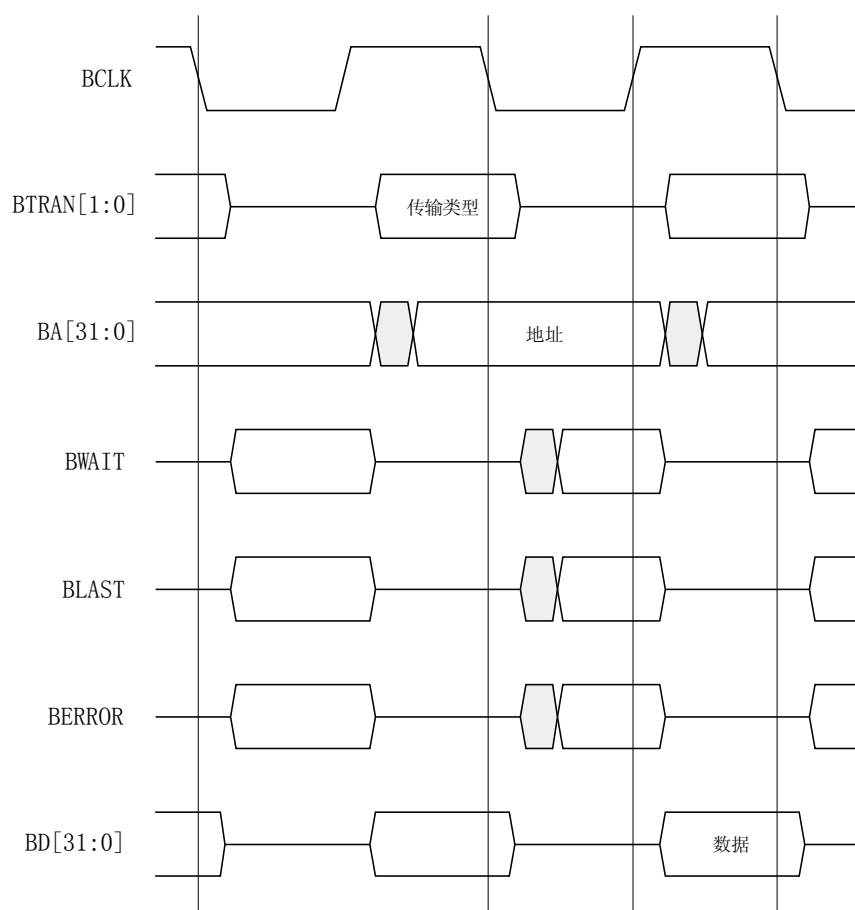


图 4.21 传输响应信号时序

信号不要在 **BCLK** 的高电平相位期间驱动以允许提供给信号驱动器之间一整个周转相位。

4.8.13 数据总线

双向数据总线，**BD[31: 0]**，被用来在总线主机和从机之间传输数据。传输大小和方向由控制信号给出，参见地址和控制信息部分的描述。

数据总线在一次非连续传输的第一个 **BCLK** 低电平相位期间并不驱动。它可以在除复位之外的其他时间段由合适的主机和从机驱动。

在写传输期间：

- 主机在传输的所有相位期间驱动数据总线，除了非连续传输的第一个 **BCLK** 低电平相位期间；
- 从机不驱动总线。

在读传输期间：

- 主机不驱动数据总线；
- 从机在传输的最后一个 **BCLK** 高电平相位期间必须驱动数据总线。对于传输的剩余部分，从机可以驱动数据总线或者让其保持三态，并规定从机不能在非连续传输的第一个 **BCLK** 低电平相位期间驱动数据总线。

下面的图表例子表示了数据总线是如何被驱动的。

图 4.22表示了一个非连续写传输的例子。

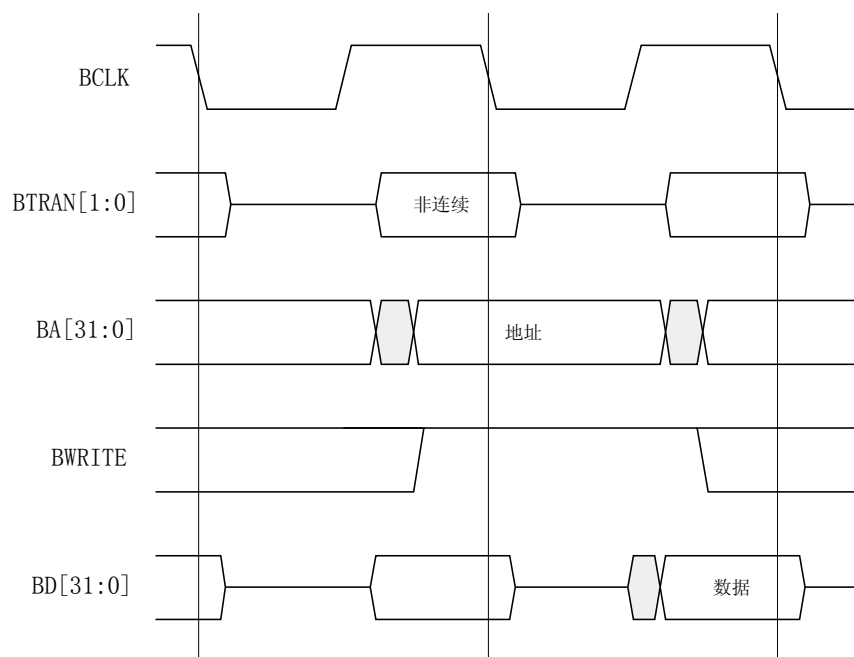


图 4.22 非连续写传输

数据总线由总线主机驱动，除了第一个 **BCLK** 的低电平相位期间。在非连续传输的开始不驱动数据总线为不同数据总线驱动器之间提供了一个完整的周转相位。

当一个写传输通过使用**BWAIT**扩展时，数据在需要完成传输的额外的**BCLK**低电平相位期间保持有效，如图 4.23所示。

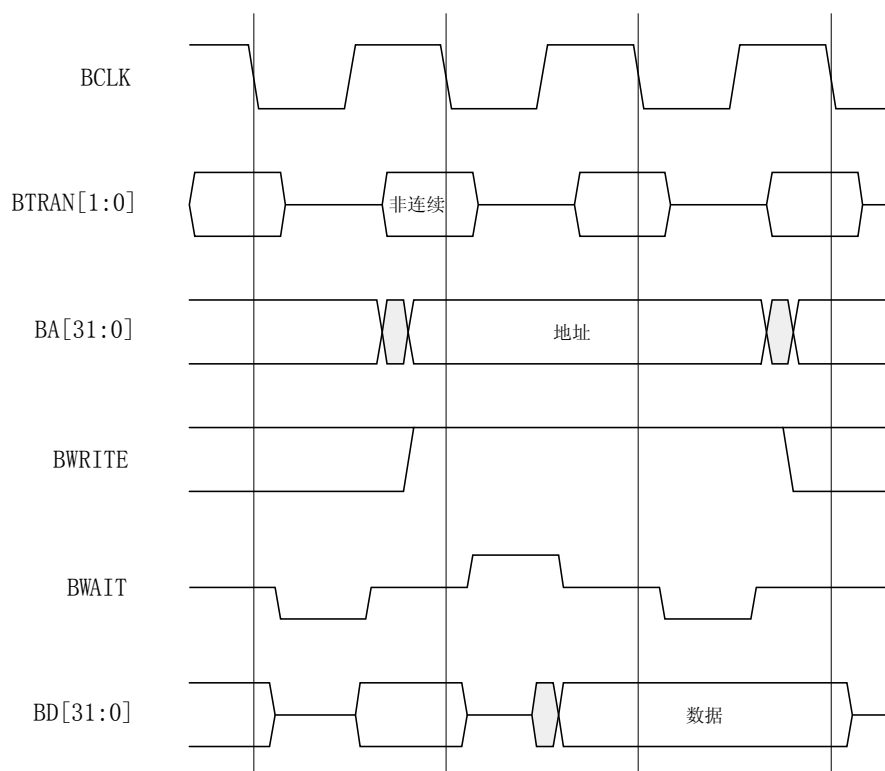


图 4.23 扩展的写传输

对于连续传输总线主机可以在传输发起时的**BCLK**低电平相位期间驱动数据总线，如图 4.24所示。这被允许作为一个相位周转并且对连续传输而言并不要求。

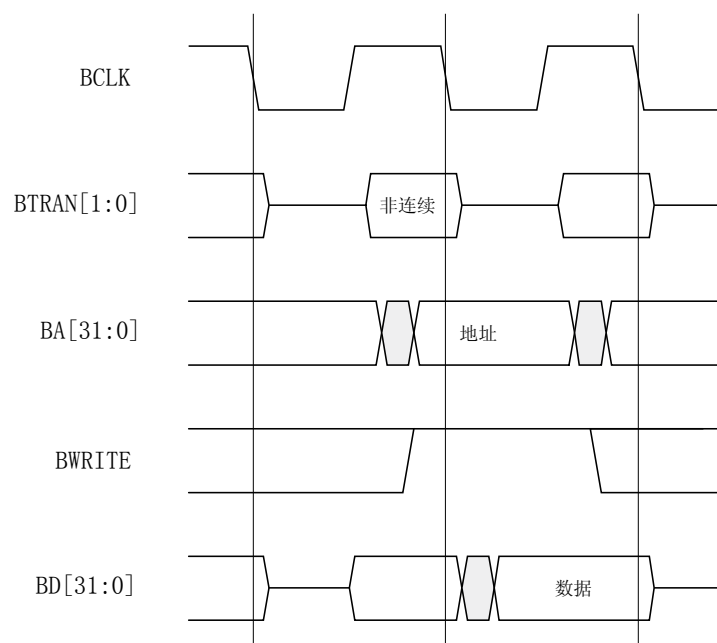


图 4.24 连续写传输

在一个读传输期间从机驱动数据总线，并且和写周期的情况一样，对于非连续传输的第一个周期的**BCLK**低电平相位期间数据总线不被驱动（见图 4.25）。之后在余下的传输期间总线从机可以驱动总线。

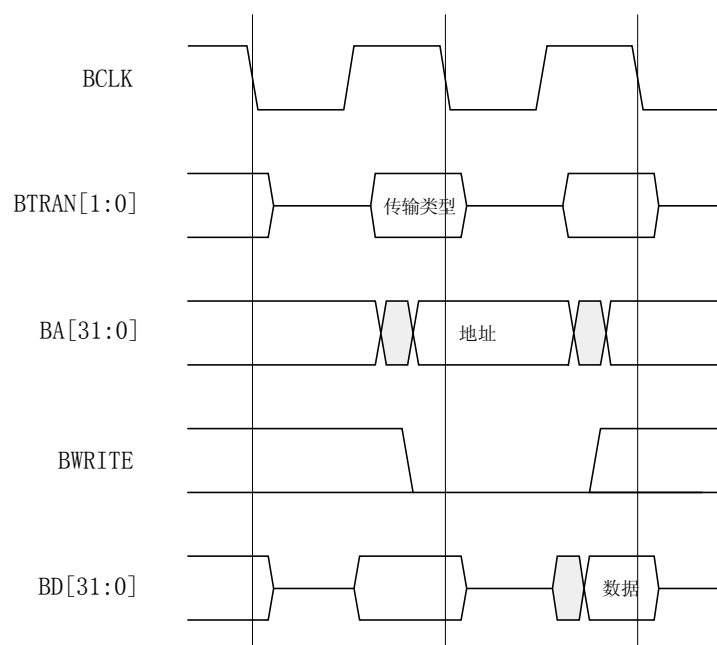


图 4.25 读传输

没有要求从机在整个传输期间驱动总线。仅有的要求是数据在传输的最后一个 **BCLK** 高电平相位时被被驱动。

4.8.14 仲裁信号

AREQ_x – 总线请求

AREQ_x 是从主机到仲裁器的表示主机请求总线的信号。每个主机有一个 **AREQ_x** 信号，该信号在 **BCLK** 的高电平相位期间改变。

AGNT_x – 总线授予

从仲裁器到总线主机的授予信号表示该总线主机是当前请求总线的主机中优先级最高的。在系统中每个总线主机都有一个 **AGNT_x** 信号。

需要注意 **AGNT_x** 并不表示对应主机是当前被授予总线的。相反，它表示该主机是当前优先级最高的并在传输完成时（由 **BWAIT** 为低表示），**AGNT_x** 被断言的主机被授予总线。

AGNT_x 在 **BCLK** 的低电平相位期间由仲裁器改变并且在高电平相位期间保持有效。

当 **AGNT_x** 为高时，主机必须：

- 在 **BCLK** 的高电平相位期间驱动 **BTRAN** 信号；
- 在 **BWAIT** 为低时被授予（总线）。

BLOK – 总线锁定

BLOK 是共享的总线锁定信号。该信号表示接下来的传输和当前传输是不可分割的并且其他总线主机不应该被授予访问总线。

当被授予总线时主机必须总是驱动 **BLOK** 信号为有效电平，即使主机并没在执行任何传输。这是必须的以确保仲裁处理可以继续。

如果 **BLOK** 为低仲裁器将授予请求总线的优先级最高的主机。

如果 **BLOK** 为高仲裁器将保持当前主机被授予。

BLOK 信号被仲裁器在 **BCLK** 的低电平相位期间采样并且它必须有效以便仲裁器能够在 **BCLK** 的上升沿之前产生有效的 **AGNT_x** 输出。**BLOK** 在总线主机移交周期中被仲裁器忽略。

4.9 关于ASB AMBA的组成部分

这部分描述了 AMBA 系统中的每个元素并提供了分析一个基于 ASB 的 AMBA 设计所要的求基本时序参数。

下面的被用作时序参数：

- T_{is} – 输入建立时间；
- T_{ih} – 输入保持时间；
- T_{ov} – 输出有效时间；
- T_{oh} – 输出保持时间。

除非其他声明，这些时序参数应用于信号的上升沿和下降沿。三态使能和禁能时间并没详细指定。所有三态禁能时间必须小于一个 **BCLK** 的相位以阻止总线冲突发生。在特定场合下如果使能三态驱动器是主要因素的话，三态使能时间可能需要被考虑到输出有效时间里。

4.10 ASB总线从机

系统中一个 ASB 总线从机响应由总线主机发起的传输。从机使用一个来自译码器的 **DSEL** 选择信号来确定它何时响应总线主机。所有其他传输要求的信号，例如地址和控制信号，将由总线主机产生。

译码器极大的简化了从机接口并去除了从机需要理解可能出现在总线上的不同传输类型的需要。

4.10.1 接口框图

图 4.26表示了一个ASB总线从机接口的框图。

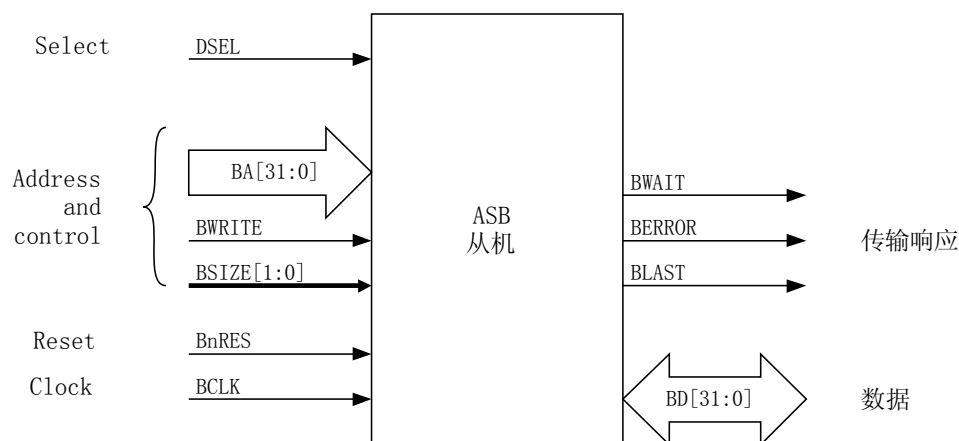


图 4.26 ASB 总线从机接口

4.10.2 总线从机接口描述

总线从机接口用下列术语来描述：

- 传输响应；
- 数据。

传输响应

当 **DSEL** 被断言时从机必须在 **BCLK** 的低电平相位期间提供传输响应。使用 **BWAIT**、**BERROR** 和 **BLAST** 告知下列响应将被产生。

- | | |
|----------------|--|
| WAIT | 传输在完成前必须被扩展； |
| DONE | 传输已经成功完成； |
| LAST | 传输成功完成，但是从机不能接受下一步地突发传输或则到达了存储器边界； |
| ERROR | 传输并未成功完成。错误情况将被发信号给总线主机以便使主机意识到传输并未正确完成； |
| RETRACT | 传输并未完成，因此总线主机应该重试传输。 RETRACT 响应被从机用来阻止总线被一个可能占据许多周期才能完成的传输锁上。 |

许多从机将仅使用 **WAIT** 和 **DONE** 响应并且在这种情况下，当一个传输响应被提供时，**BERROR** 和 **BLAST** 将都为低。

当从机未被选中时（由 **DSEL** 为低表示），传输响应信号必须为三态。响应信号在复位期间也必须为三态。

数据

从机接口实现为一个简单的状态机，运行在时钟的下降沿来确定何时数据传输会发生。在复位期间状态机进入 **NOT_SELECTED** 状态（见图 4.27）。

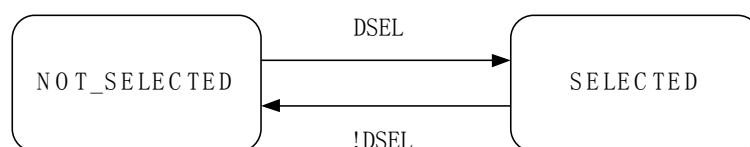


图 4.27 ASB 总线从机接口状态机

当在 **SELECTED** 状态时对于写传输从机在时钟的下降沿采样数据。如果有要求，从机可以使用上面描述的传输响应信号扩展传输。

对于读传输从机必须在传输的最后一个时钟的高电平相位期间驱动数据总线。

如果传输是通过插入等待状态而被扩展的，那么从机可以在传输的附加周期驱动数据总线，或者可以保持数据总线为三态直到传输的最后一个相位为止。

为避免从机不得不确认传输是连续或者非连续的，通常简单的做法是设计一个在任何传输的第一个相位期间不驱动数据总线的从机。

在复位期间当从机在 **NOT_SELECTED** 状态时数据总线必须为三态。

4.10.3 时序图

关于访问一个ASB总线从机的时序参数如图 4.28所示。

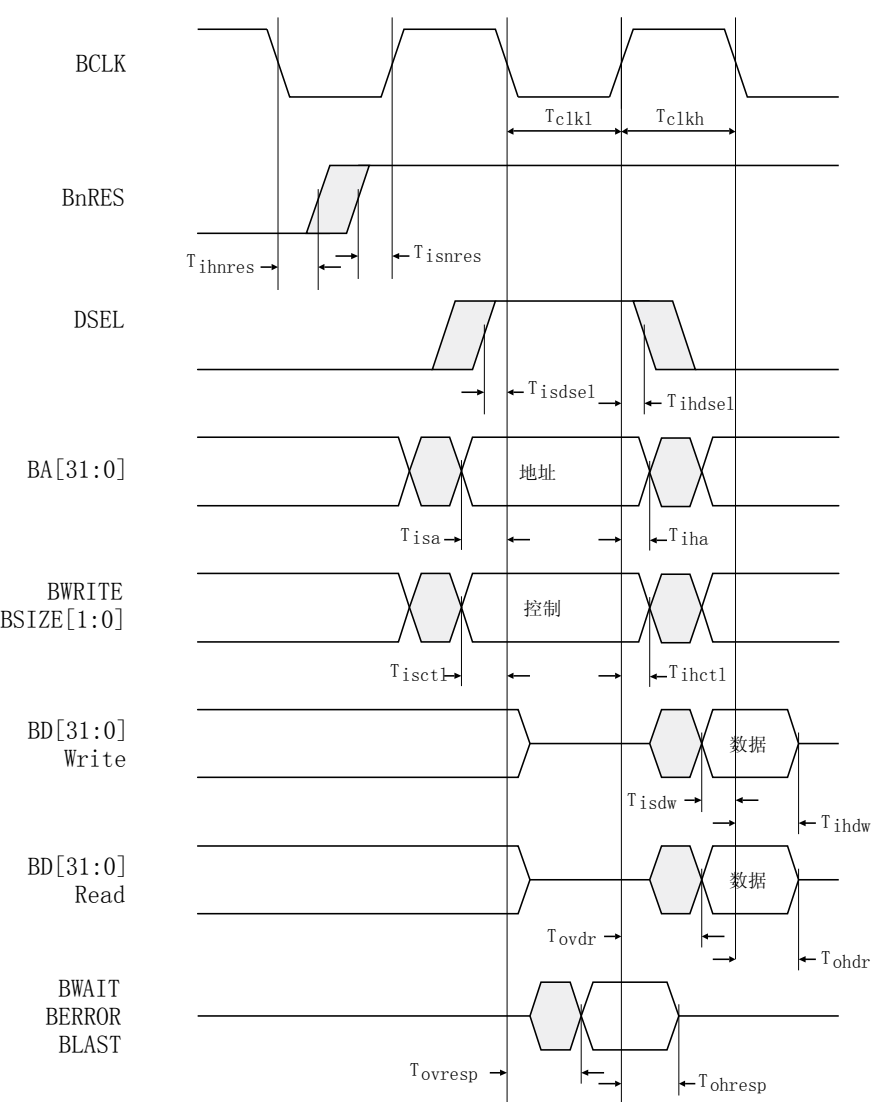


图 4.28 ASB 从机传输

4.10.4 时序参数

给出关于一个ASB总线从机的时序参数，表 4.6为输入信号而表 4.7为输出信号。双向信号可以在两个表中都能找到。

表 4.6 ASB 从机输入参数

参数	描述
$T_{clk\ell}$	BCLK 低电平时间
T_{clkh}	BCLK 高电平时间
T_{isnreS}	在 BCLK 上升沿前 BnRES 失效建立时间
T_{ihnres}	在 BCLK 下降沿后 BnRES 失效保持时间
$T_{isd\ell}$	在 BCLK 下降沿前 DSEL 建立时间
$T_{ihd\ell}$	在 BCLK 上升沿后 DSEL 保持时间

续上表

参数	描述
T _{isa}	在 BCLK 下降沿前 BA[31: 0] 建立时间
T _{iha}	在 BCLK 下降沿后 BA[31: 0] 保持时间
T _{isctl}	在 BCLK 上升沿前 BWRITE 和 BSIZE[1: 0] 建立时间
T _{ihctl}	在 BCLK 下降沿后 BWRITE 和 BSIZE[1: 0] 保持时间
T _{isdw}	对写传输，在 BCLK 下降沿前 BD[31: 0] 建立时间
T _{ihdw}	对写传输，在 BCLK 下降沿后 BD[31: 0] 保持时间

表 4.7 ASB 从机输出参数

参数	描述
T _{ovresp}	在 BCLK 下降沿后 BWAIT 、 BERROR 和 BLAST 有效时间
T _{ohresp}	在 BCLK 上升沿后 BWAIT 、 BERROR 和 BLAST 保持时间
T _{ovdr}	对读传输，在 BCLK 上升沿后 BD[31: 0] 有效时间
T _{ohdr}	对读传输，在 BCLK 下降沿后 BD[31: 0] 保持时间

注：如果总线从机被设计为译码器、地址和控制信号都在 **BCLK** 的下降沿采样那么输入保持时间的整个相位由总线协议来保证。

可以确认保持时间的整个相位是通过在传输中插入一个额外的等待状态而被提供在数据总线上。

4.11 ASB总线主机

在 AMBA 系统中 ASB 总线主机有最复杂的总线接口。一个典型的 AMBA 系统设计者将会使用预先设计好的总线主机，因此并不需要关注总线主机接口的细节。

一个总线主机接口也可以包含一个从机接口，不是用作测试就是用来编程总线主机的操作。在这样的情况下许多接口信号将在主机接口和从机接口之间共享。

4.11.1 接口框图

ASB 总线主机的接口框图显示了主要的信号群组。

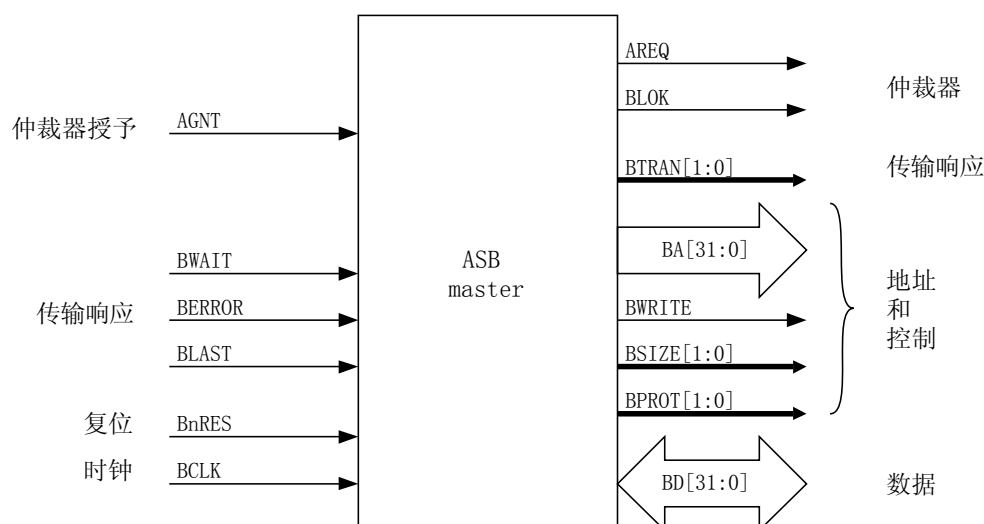


图 4.29 ASB 总线主机接口框图

4.11.2 总线主机接口描述

总线主机接口包含两个状态机：

- 第一个状态机决定主机当前是否被授予总线；
- 第二个，更加复杂的状态机被用来控制主机的总线接口。

授予状态机

授予状态机被用来确定总线主机是否被授予总线。它随着 **BCLK** 的上升沿同步并且只有两个状态，**GRANTED** 和 **NOT_GRANTED**。状态机的框图见图 4.30

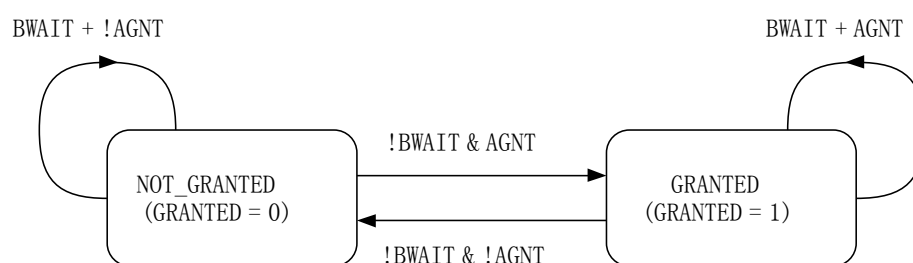


图 4.30 总线主机授予状态机

从这个状态机的输出是 **GRANTED** 信号，该信号在总线主机主状态机中使用。

注：**AGNT** 信号可以持续许多时钟周期有效，但是仅在当 **AGNT** 有效且 **BWAIT** 为低时总线主机才真正意义上被授予总线。

一个重要的设计考虑因素就是状态机可以异步复位到这两个状态中的任意一个，具体取决于 **AGNT** 信号的值。在复位期间系统中的一个总线主机被设置为默认总线主机，由 **AGNT** 在复位期间被断言来表示，并且（该主机）将被复位到 **GRANTED** 状态。所有其他总线主机将被复位到 **NOT_GRANTED** 状态。

4.11.3 总线接口状态机

总线主机主状态机是下降沿触发的并且包含六个状态。整个状态框图见图 4.32，非常复杂但可以在四个象限中来考虑，如图 4.31。

无传输请求 未授予	传输请求 未授予
无传输请求 授予	传输请求 授予

图 4.31 总线接口状态机象限图

传输请求（和）授予象限包含三个状态，这些状态处理总线周转和撤消操作。

两个内部总线主机信号，GRANTED和REQUEST，控制着围绕状态机的主要转换（见图 4.32）：

- GRANTED 由前述的简单状态机产生；
- REQUEST 直接由总线主机产生。

REQUEST 在总线主机请求一次总线上的传输时被断言为高电平而在总线主机不需要访问总线时为低电平。

状态机的转换不受 GRANTED 和 REQUEST 控制的唯一时间是当总线主机处于 ACTIVE 状态时。在这个状态下（状态机）转换到下一个状态是由接收到的传输响应决定的。图中所示的 WAIT、DONE、LAST、ERROR 和 RETNEXT 与传输响应信号的编码一致。

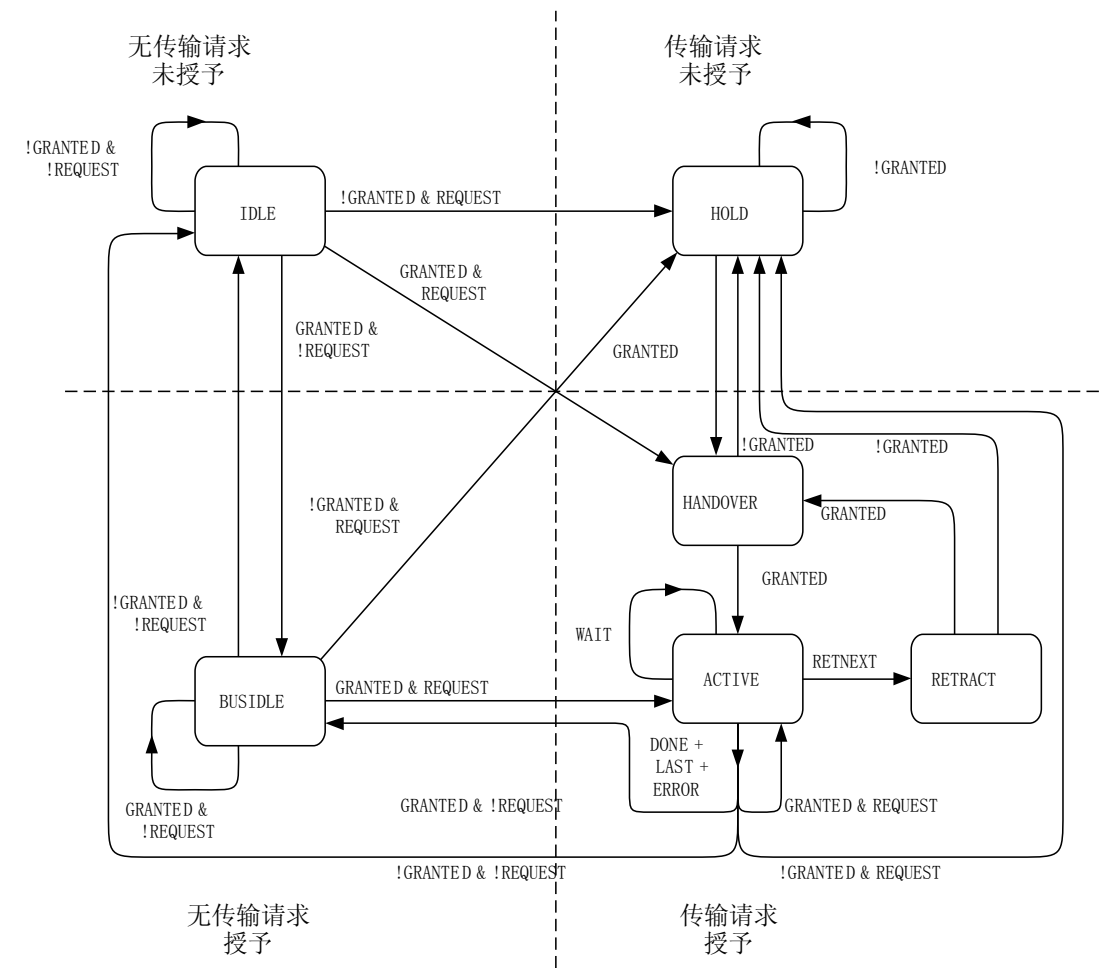


图 4.32 总线主机主状态机

需要注意状态框图假定一旦总线主机已经作出一次传输请求（由 **REQUEST** 表示），那么 **REQUEST** 将保持有效直到总线主机已经完成一次传输为止。

由于总线主机主状态机运行在时钟的下降沿所以它需要使用被锁存的传输响应信号 **BWAIT**、**BERROR** 和 **BLAST** 的译码以控制（状态机）退出 **ACTIVE** 状态。

复位条件并没在图中显示并且和授予状态机相类似，总线主机主状态机的复位条件比较复杂。如果 **AGNT** 在复位期间（当 **BnRES** 为低电平时）生效，总线主机是默认总线主机并进入 **BUSIDLE** 状态。然而，如果 **AGNT** 在复位期间无效那么总线主机将进入 **IDLE** 状态。

表 4.8表示了在每个状态期间必须发生的动作。

表 4.8 每个状态必须发生的动作

名称	描述	动作
IDLE	总线主机并不请求总线且为被授予	内部 BTRAN 为仅地址。 主机时钟使能。 主机地址总线为三态。 主机数据总线为三态。

续上表

名称	描述	动作
BUSIDLE	总线主机请求总线，但未被授予	内部 BTRAN 由主机指示。 主机时钟使能。 主机地址总线使能由 GRANTED 信号产生。 主机数据总线为三态。
HOLD	主机请求总线，但未被授予	内部 BTRAN 为仅地址。 主机时钟禁能。 主机地址总线为三态。 主机数据总线为三态。
HANDOVER	当不同总线主机之间改变（状态）时该状态提供总线周转	内部 BTRAN 为连续。 主机时钟禁能。 主机地址总线使能由 GRANTED 信号产生。 主机数据总线为三态。
ACTIVE	当数据传输发生时为活动状态。退出该状态取决于传输响应	内部 BTRAN 由主机指示。 主机时钟使能来自 BWAIT 。 主机地址总线使能由 GRANTED 信号产生。 如果为写操作则主机数据总线使能。
RETRACT	撤消状态，系统中其余部件留意传输结，但是总线主机并不高级。	内部 BTRAN 为仅地址。 主机时钟禁能。 主机地址总线使能由 GRANTED 信号产生。 如果为写操作则主机数据总线使能。

BTRAN[1: 0]三态驱动器在 **AGNT** 和 **BCLK** 都为高时使能。

主机地址总线使能是用来控制 **BA[31: 0]**、**BWRITE**、**BSIZE[1: 0]**和 **BLOK** 的三态使能。主机数据总线使能时用来控制 **BD[31: 0]**的三态使能。

4.11.4 总线主机时序图

下面的图表示了在 **AMBA** 系统中关于一个 **ASB** 总线主机操作的时序参数：

- 图 4.33表示了一个ASB总线主机非连续传输；
- 图 4.34表示了一个ASB总线主机连续传输；
- 图 4.35表示了一个ASB主机仅地址传输；
- 图 4.36表示了一个ASB总线主机仲裁和复位信号。

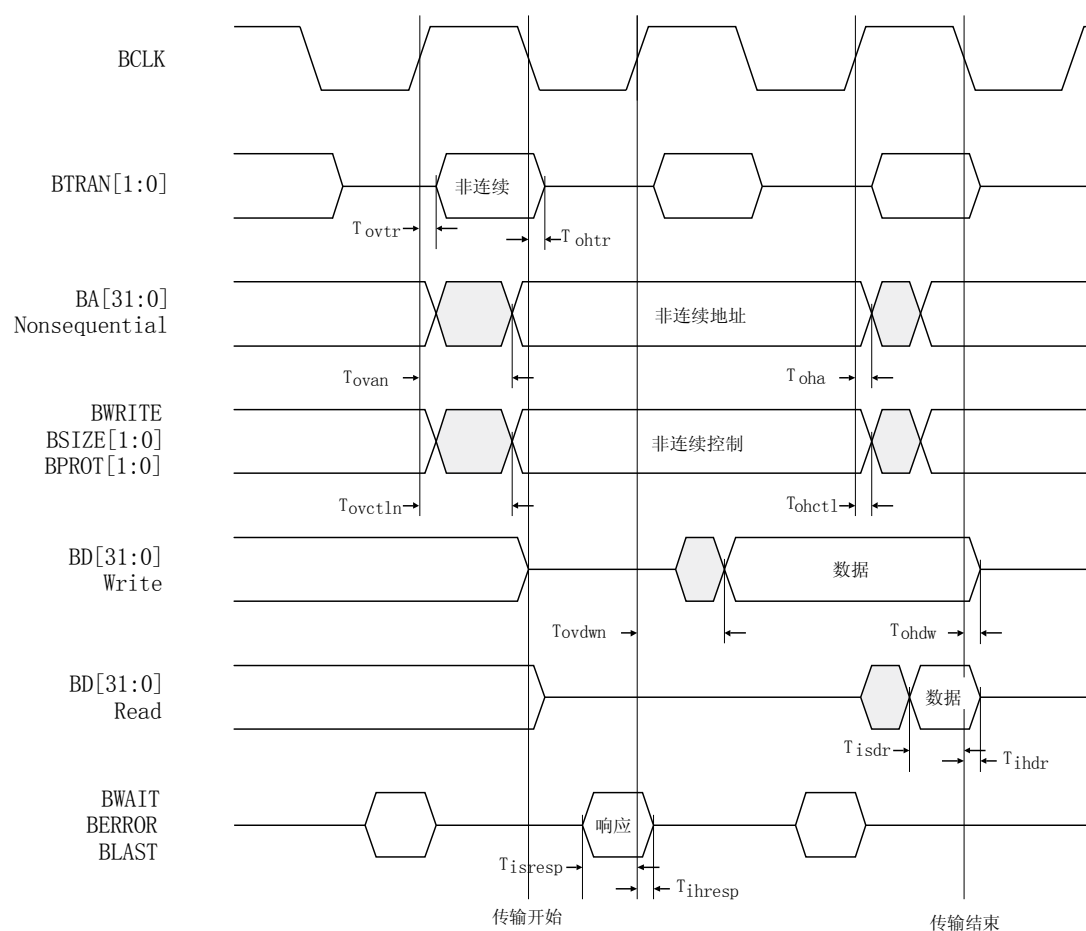


图 4.33 ASB 总线主机非连续传输

对于非连续传输，如图 4.33所示，在传输开始前地址和控制信号在**BCLK**的该高电平相位期间有效。**AMBA**协议的一个重要特征就是对非连续传输允许较少的输出有效时间，这种方式通过在每个非连续传输的开始由译码器自动插入一个等待状态来实现。

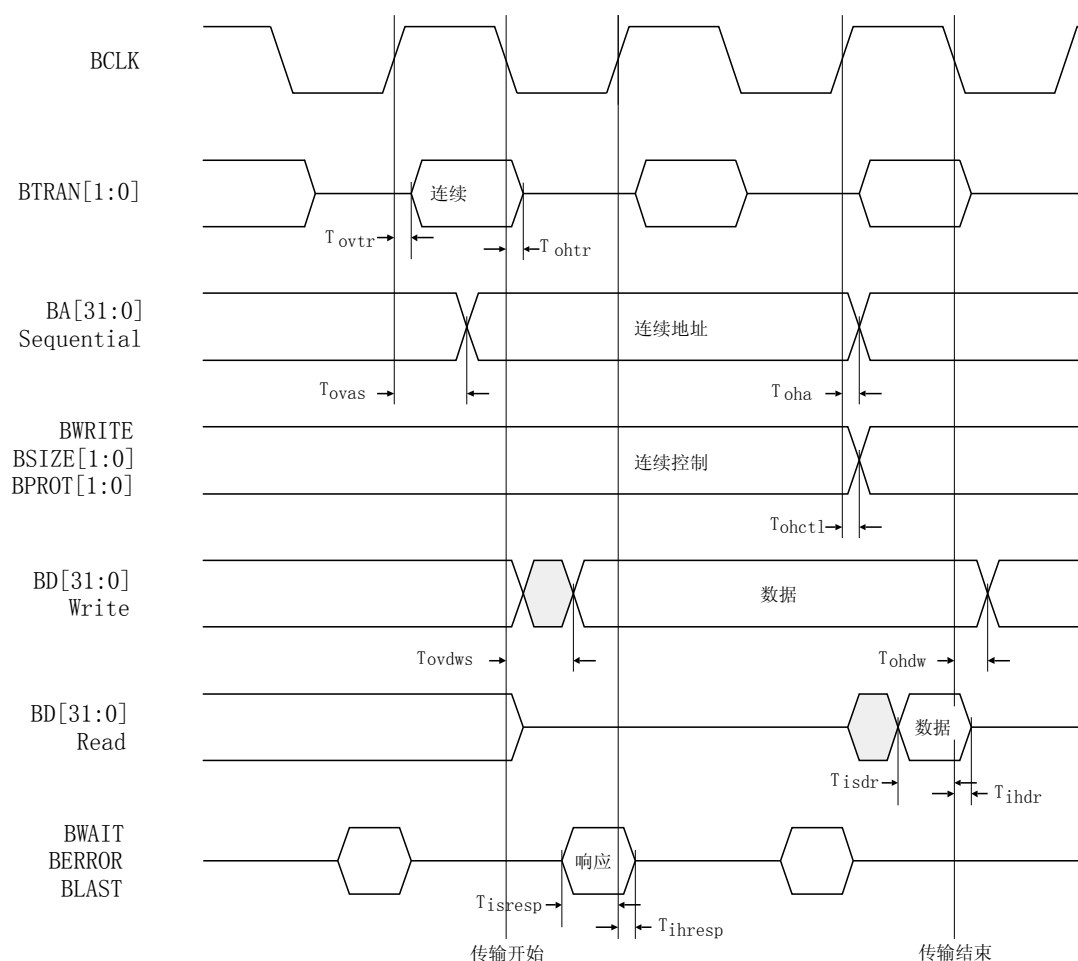


图 4.34 ASB 总线主机连续传输

对地址和控制信号的有效时间连续传输有不同的时序参数（见图 4.34）。在一个典型的总线主机中，连续传输的输出有效时间将远好于非连续传输。地址、控制 and 数据的输出保持时间是相同的并且独立于传输类型。

连续传输和非连续传输之间的其他不同之处是在连续传输期间可以在传输的第一个相位期间驱动，因此数据有效参数的详细说明不同于 **BCLK** 的下降沿。

在传输开始前对于仅地址传输的地址和控制信号可以在时钟的高电平相位驱动，或者处于主机移交的情况下可以在传输自身的时钟低电平相位期间驱动（见图 4.35）。地址和控制有效的时序参数仅与仅地址传输之后紧跟一个连续传输时有关，并且在这种情况下必须驱动地址和控制信号以便使它们在仅地址传输的低电平相位期间有效，换言之这意味着地址和控制信号在连续传输前时钟的高电平相位期间也有效。

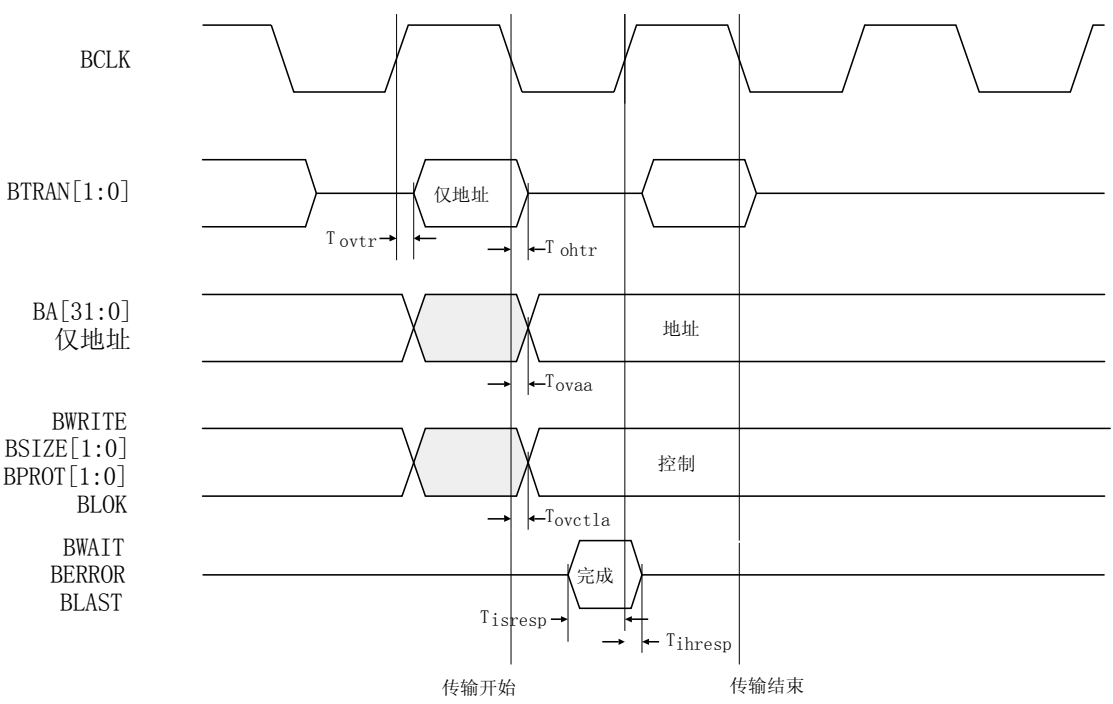


图 4.35 ASB 主机仅地址传输

图 4.36表示了ASB总线主机仲裁和复位信号。

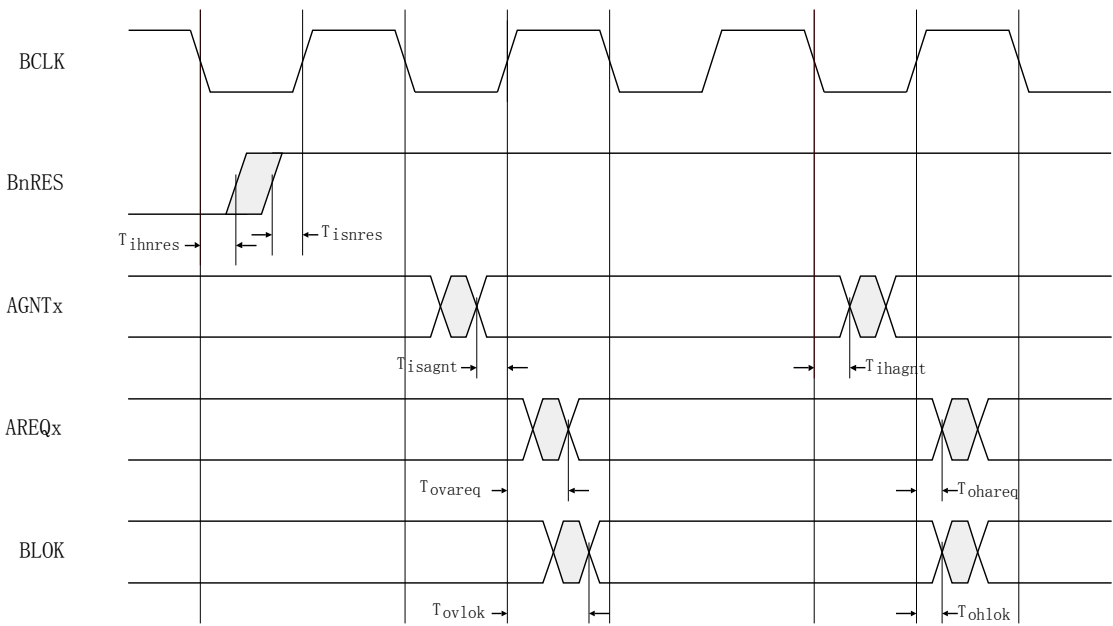


图 4.36 ASB 总线主机仲裁和复位信号

BnRES 信号可以被异步断言，因此没有关于该信号断言的建立和保持参数。总线主机输出的 **AREQ** 信号，在时钟的高电平相位期间改变而从仲裁器返回的 **AGNT** 信号在时钟的低电平相位期间改变。

4.11.5 时序参数

下面的两个表格中以文本方式表示了在一个AMBA系统中关于ASB总线主机操作的时序参数。表 4.9为输入信号。表 4.10为输出信号。双向信号在两个表中都能找到。

表 4.9 总线主机输入时序参数

参数	描述
T_{clkI}	BCLK 低电平时间
T_{clkH}	BCLK 高电平时间
T_{isnres}	在 BCLK 上升沿前 BnRES 失效建立时间
T_{ihnres}	在 BCLK 下降沿后 BnRES 失效保持时间
T_{isresp}	在 BCLK 上升沿前 BWAIT 、 BERROR 和 BLAST 建立时间
T_{ihresp}	在 BCLK 上升沿后 BWAIT 、 BERROR 和 BLAST 保持时间
T_{isdr}	对读传输，在 BCLK 下降沿前 BD[31: 0] 建立时间
T_{ihdr}	对读传输，在 BCLK 下降沿后 BD[31: 0] 保持时间
T_{isagnt}	在 BCLK 上升沿前 AGNT 建立时间
T_{ihagnt}	在 BCLK 下降沿后 AGNT 保持时间

表 4.10 总线主机输出时序参数

参数	描述
T_{ovtr}	在 BCLK 上升沿后 BTRAN 有效时间
T_{ohtr}	在 BCLK 下降沿后 BTRAN 保持时间
T_{ovan}	对非连续传输，在 BCLK 上升沿后 BA[31: 0] 有效时间
T_{ovas}	对连续传输，在 BCLK 上升沿后 BA[31: 0] 有效时间
T_{ovaa}	对仅地址传输，在 BCLK 下降沿后 BA[31: 0] 有效时间
T_{oha}	在 BCLK 上升沿后 BA[31: 0] 保持时间
T_{ovctlN}	对非连续传输，在 BCLK 上升沿后 BWRITE 、 BSIZE[1: 0] 和 BPROT[1: 0] 有效时间
T_{ovctlA}	对仅地址传输，在 BCLK 下降沿后 BWRITE 、 BSIZE[1: 0] 和 BPROT[1: 0] 有效时间
T_{ohctl}	在 BCLK 上升沿后 BWRITE 、 BSIZE[1: 0] 和 BPROT[1: 0] 保持时间
T_{ovdwn}	对非连续写传输，在 BCLK 上升沿后 BD[31: 0] 有效时间
T_{ovdws}	对连续写传输，在 BCLK 下降沿后 BD[31: 0] 有效时间
T_{ohdw}	对写传输，在 BCLK 下降沿后 BD[31: 0] 保持时间
T_{ovlok}	在 BCLK 上升沿后 BLOK 有效时间
T_{ohlok}	在 BCLK 上升沿后 BLOK 保持时间

续上表

参数	描述
T _{ovareq}	在 BCLK 上升沿后 AREQ 有效时间
T _{ohareq}	在 BCLK 上升沿后 AREQ 保持时间

4.12 ASB译码器

在一个 AMBA 系统中译码器被用来执行集中地址译码功能，主要有两个优点：

- 通过使外设独立于系统存储器映射，提高了外设的移植性；
- 通过集中地址译码和总线控制功能，简化了总线从机设计。

译码器的三个主要任务是：

- 地址译码；
- 默认传输响应；
- 保护单元。

ASB 译码器对每个在 ASB 总线上的从机产生一个选择信号并且，在特定情况下，将不选择任何从机并自身提供处理响应。

译码器极大的简化了从机接口并且去除了从机需要理解可能发生在总线上的不同传输类型的需要。

译码器的一个重要特征是它能够通过对地址译码提供译码周期来提高系统性能。因为译码器能够识别传输是连续或者非连续的，故对译码器而言仅在需要时增加一个译码周期是很简单的任务。

实际上译码器明显有助于提高系统性能。在一个非 AMBA 系统中的关键途径（步骤），例如一个读传输将是：

- 1、 主机输出地址；
- 2、 地址译码以选择从机；
- 3、 从从机返回给主机的数据输出和响应。

然而，在 AMBA 系统中只要总线主机正在执行一个连续传输时就可以移除中间的步骤，因为被选中的从机和前一次传输一样（这个条件）是已知的。译码器可以使用这个因素在需要时通过对地址译码仅插入一个等待状态来提高系统性能，这个过程被用作非连续传输。这就是所熟知的插入一个译码周期。

在时钟频率足够低因而地址译码不需要额外等待状态的设计中，那么译码器的任务可以简化。

译码器也用来提供许多总线维护功能。首先，译码器可以担当一个简单的保护单元，它可以发出一个错误响应给试图访问一个非法或则受保护的存储器映射区域的总线主机。译码器也在仅地址传输期间提供传输响应，这时没有从机被选中。

4.12.1 接口框图

图 4.37表示了一个ASB译码器。

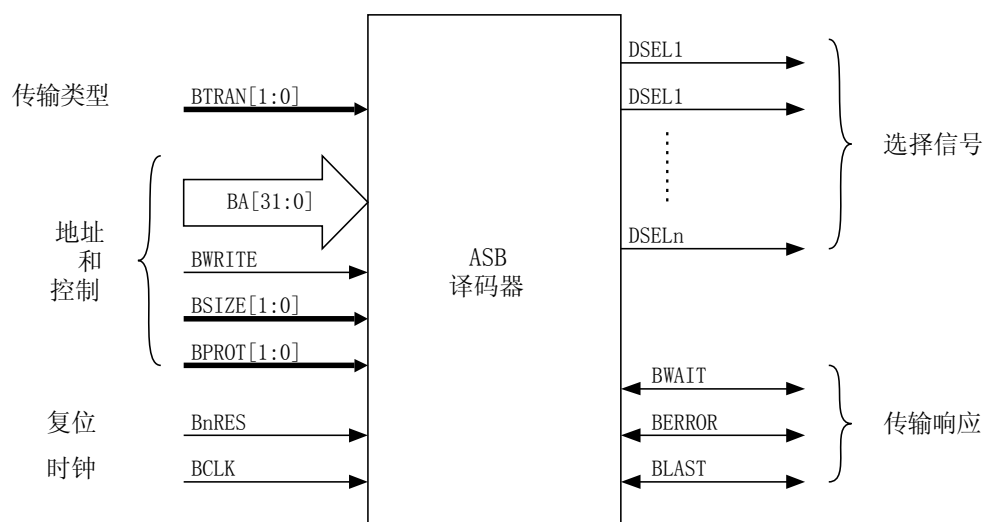


图 4.37 ASB 译码器接口框图

4.12.2 译码器描述

译码器有两种可能的实现方式，取决于系统设计的性能要求：

- 译码器的常规实现将包含在非连续传输时插入译码周期并且在存储器边界时打断突发传输；
- 在一些系统设计中，典型的带有较低的时钟频率，译码周期将不再要求因此可以实现一个简单的译码器。

带译码周期

译码器实现为一个运行在时钟下降沿的状态机并有四个状态（见图 4.38）。在复位期间状态机应该进入ADDRONLY状态。

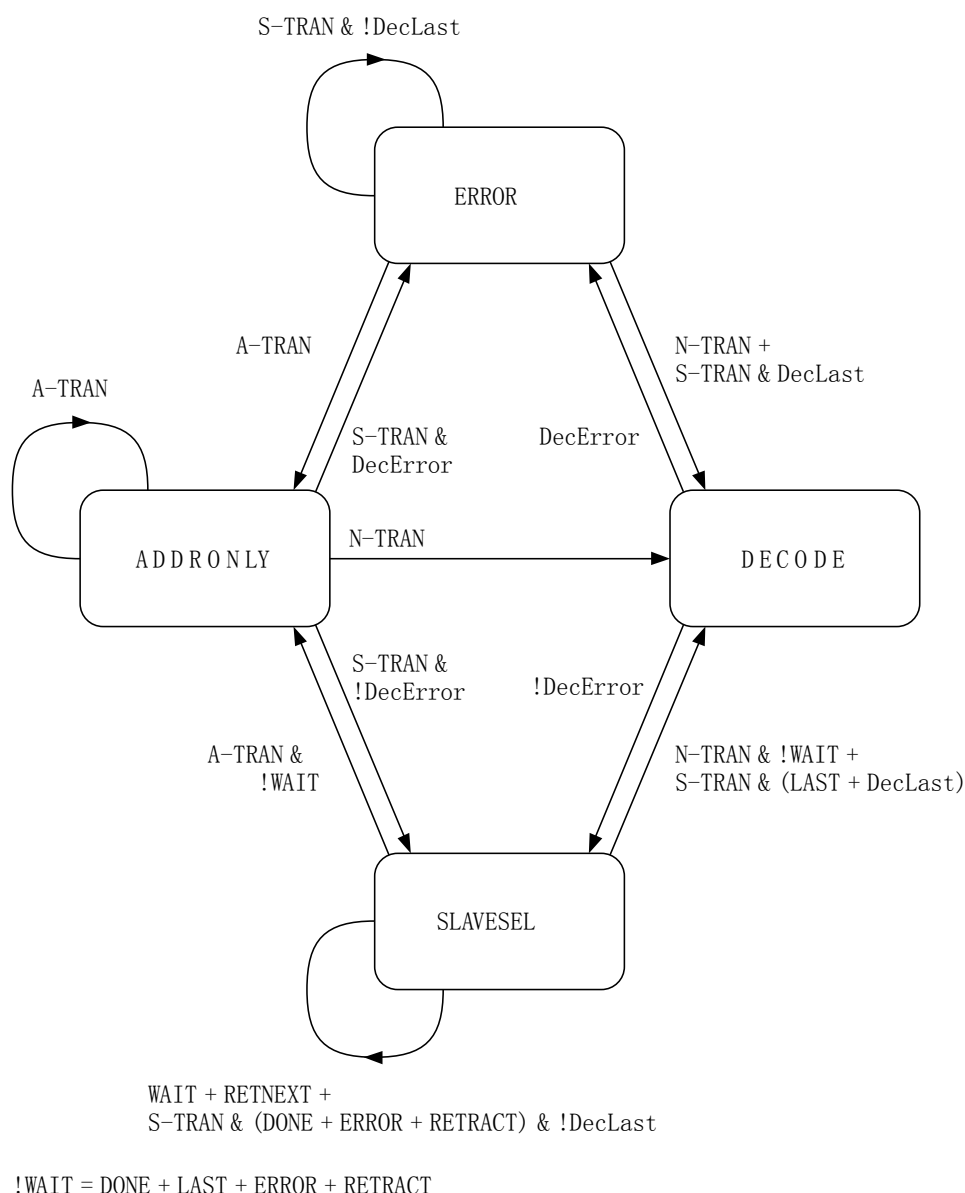


图 4.38 带译码周期的译码器状态机

围绕状态机的转换由下一次传输的类型、来自当前传输的传输响应和两个内部译码信号 **DecLast** 和 **DecError** 来控制。状态机框图中表示出的 **WAIT**、**DONE**、**LAST**、**ERROR**、**RETNEXT** 和 **RETRACT** 和传输响应信号的编码一致。

当译码器检测到一个连续传输将要越过存储器边界时由译码器产生 **DecLast** 信号并和外部的 **BLAST** 信号一起被用来影响地址译码，即使是连续传输。

DecError 是另一个地译码器内部信号并且当译码器检测到以下错误时产生：

- 在传输地址上没有从机存在；
- 到受保护的存储区域的传输；
- 传输的对齐形式不被存储器系统支持。

译码器执行以下功能：

- 在 **ADDRONLY** 状态：
 - 随机译码地址；

- 在 BCLK 的低电平相位期间提供一个 DONE 传输响应;
- 如果下一次传输的类型是连续传输并且地址有效则在 BCLK 的高电平相位期间断言 DSELx 信号。
- 在 DECODE 状态:
 - 译码地址;
 - 在 BCLK 低电平相位期间提供一个 WAIT 传输响应;
 - 如果地址有效则在 BCLK 高电平相位期间断言 DSELx 信号。
- 在 SLAVESEL 状态:
 - 由被选中的从机驱动传输响应;
 - 当传输处于等待状态, 或者如果下一次传输是连续传输且没有检测到 LAST 条件时保持 DSELx 有效。
- 在 ERROR 状态:
 - 在 BCLK 的低电平相位期间提供一个 ERROR 传输响应。

不带译码周期

没有实现译码周期的译码器移除了 DECODE 状态。这简化了状态机框图, 如图 4.39 所示。

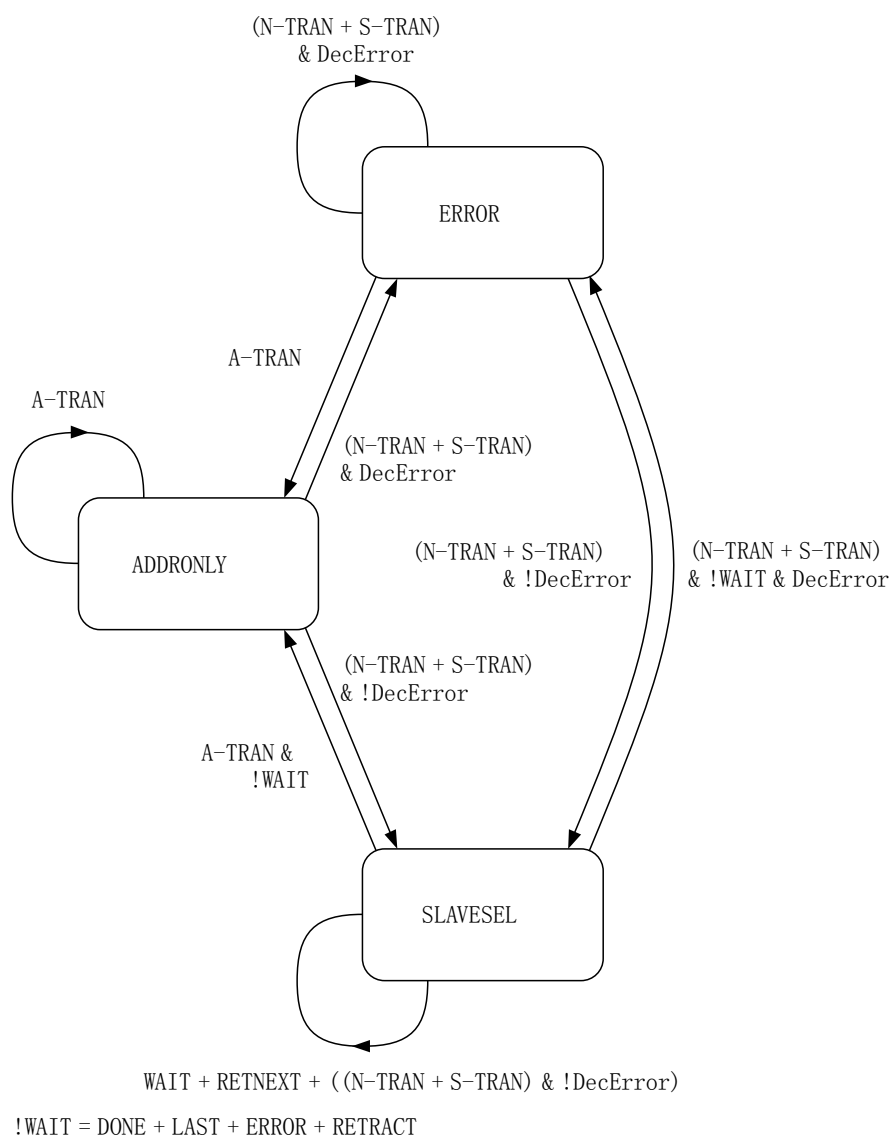


图 4.39 不带译码周期的译码器状态机

4.12.3 时序图

图 4.40表示了带译码周期的ASB译码器的时序参数。图 4.41表示了不带译码周期的译码器的时序参数。这两个图之间的主要不同是当译码周期并未插入时**DSEL**信号的时序取决于地址和控制信号的时序。

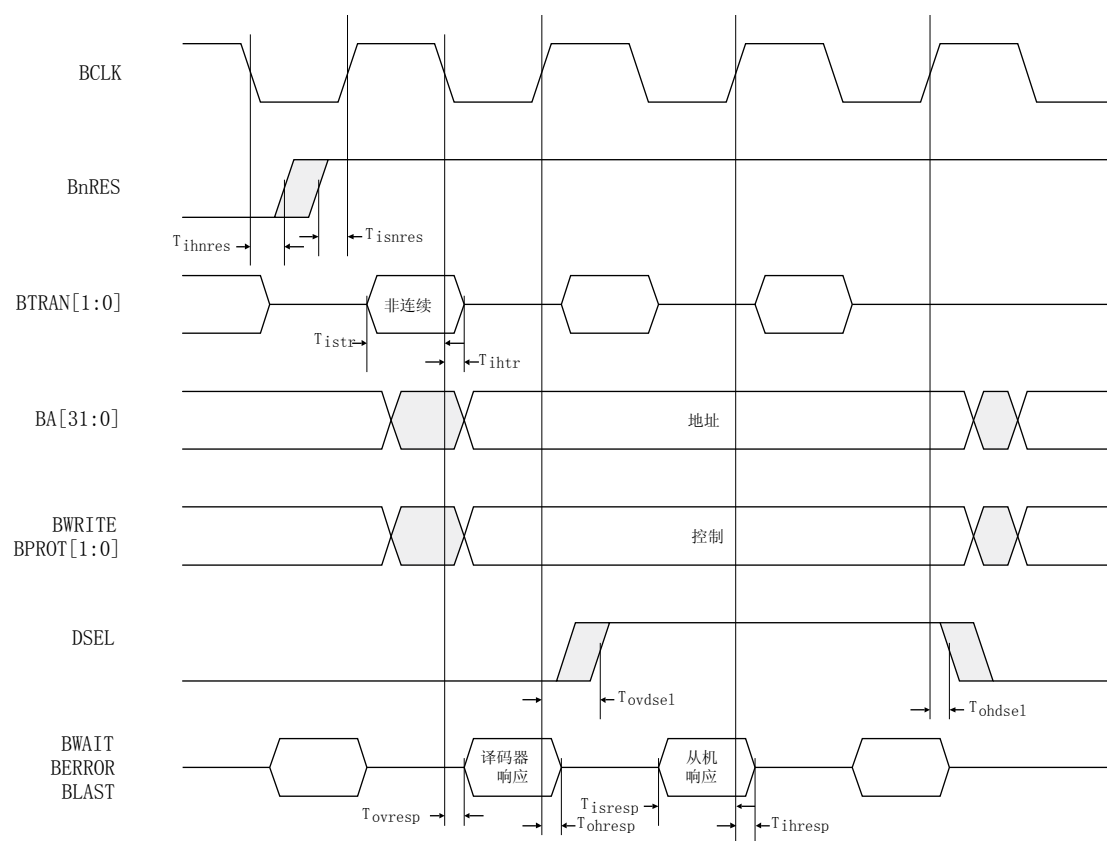


图 4.40 带译码周期的 ASB 译码器

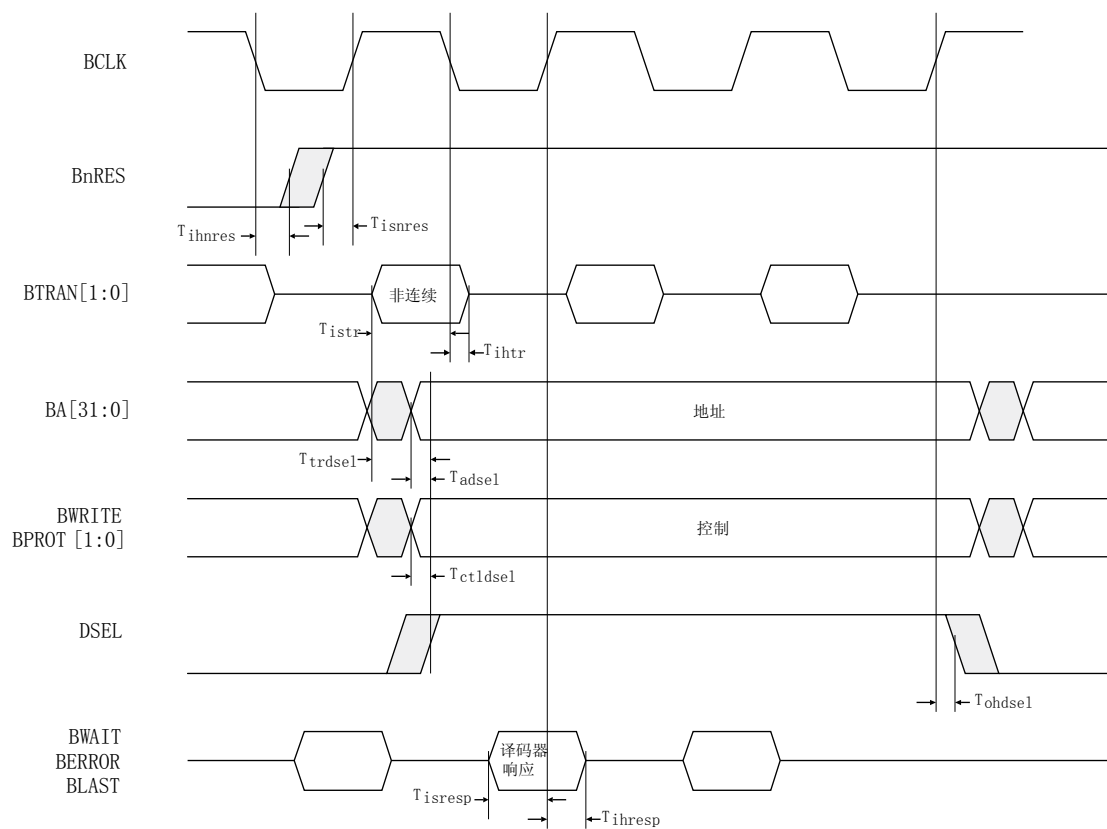


图 4.41 不带译码周期的 ASB 译码器

4.12.4 时序参数

下表关于一个 ASB 译码器的时序参数：

- 表 4.11为输入信号；
- 表 4.12为输出信号；
- 表 4.13为组合产生的输出；

表 4.11 ASB 译码器输入参数

参数	描述
$T_{\text{clk}l}$	BCLK 低电平时间
$T_{\text{clk}h}$	BCLK 高电平时间
T_{isnres}	在 BCLK 上升沿前 BnRES 失效的建立时间
T_{ihnres}	在 BCLK 下降沿后 BnRES 失效的保持时间
T_{istr}	在 BCLK 下降沿前 BTRAN 建立时间
T_{ihtr}	在 BCLK 下降沿后 BTRAN 保持时间
T_{isresp}	在 BCLK 上升沿前 BWAIT 、 BERROR 和 BLAST 建立时间
T_{ihresp}	在 BCLK 上升沿后 BWAIT 、 BERROR 和 BLAST 保持时间

表 4.12 ASB 译码器输出参数

参数	描述
T_{ovresp}	在 BCLK 下降沿后 BWAIT 、 BERROR 和 BLAST 有效时间
T_{ohresp}	在 BCLK 下降沿后 BWAIT 、 BERROR 和 BLAST 保持时间
T_{ovdtsel}	在 BCLK 上升沿后 DSEL 有效时间
T_{ohdtsel}	在 BCLK 上升沿后 DSEL 保持时间

表 4.13 ASB 译码器组合参数

参数	描述
T_{trdsel}	从 BTRAN 有效到 DSEL 有效的延时
T_{adsel}	从 BA 有效到 DSEL 有效的延时
T_{ctldsel}	从 BWRITE 和 BPROT[1: 0] 有效到 DSEL 有效的延时

4.13 ASB仲裁器

在一个 AMBA 系统中仲裁器的角色是控制哪个主机访问总线。每个总线主机都有一个到仲裁器的两线请求和授予接口并且在每个周期仲裁器使用一个优先级方案来决定哪个主机是当前请求总线的主机中优先级最高的。

一个共享的总线锁定信号，**BLOK**（由当前被授予总线的主机驱动）被用来表示当前传输和接下来的传输是不可分的并且他主机不应该被授予总线。

优先级方案的细节并未详细说明而是由具体应用决定。建议仲裁器使用其他信号，无论是否是 AMBA 的，来影响使用的优先级方案。

4.13.1 接口框图

图 4.42表示了ASB仲裁器接口框图。

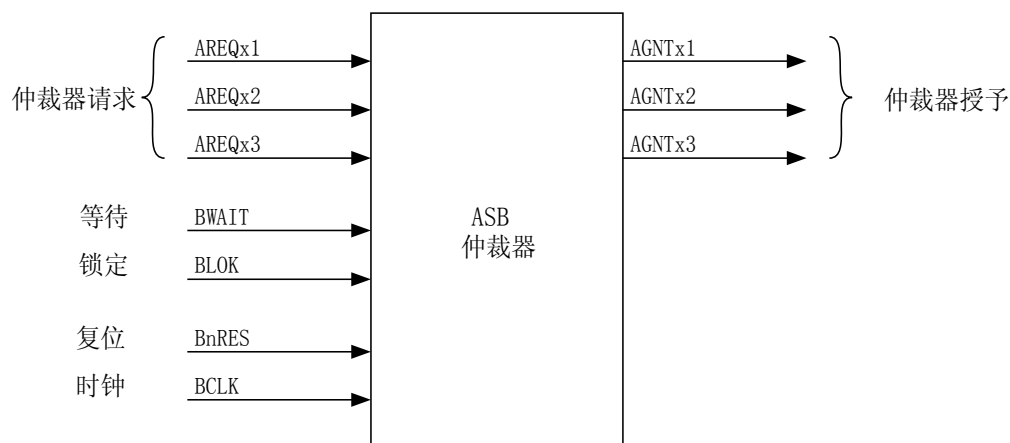


图 4.42 ASB 仲裁器接口框图

4.13.2 仲裁器描述

总线可以每个时钟周期都重新仲裁。仲裁器在 **BCLK** 的下降沿采样所有请求信号 **AREQx**，并在 **BCLK** 的低电平相位期间仲裁器使用内部优先级方案和 **BLOK** 的值来断言相应的 **AGNTx** 信号。

由于仲裁可以每个周期都改变，所以可能在一个扩展的传输期间，最高优先级的总线主机会在传输最终完成前改变好几次。当传输完成时 **AGNT** 生效的总线主机将成为下一个活动总线主机。

在总线主机移交期间 **BLOK** 信号并未驱动因此仲裁器必须假定该信号为低电平。

仲裁器必须保持当前被授予主机的一个拷贝以便仲裁器可以：

- 如果 **BLOK** 被断言则保持当前总线主机被授予；
- 决定何时总线主机改变，因此决定合适出现一个总线主机移交周期。

4.13.3 时序图

图 4.43表示了仲裁器的时序参数。

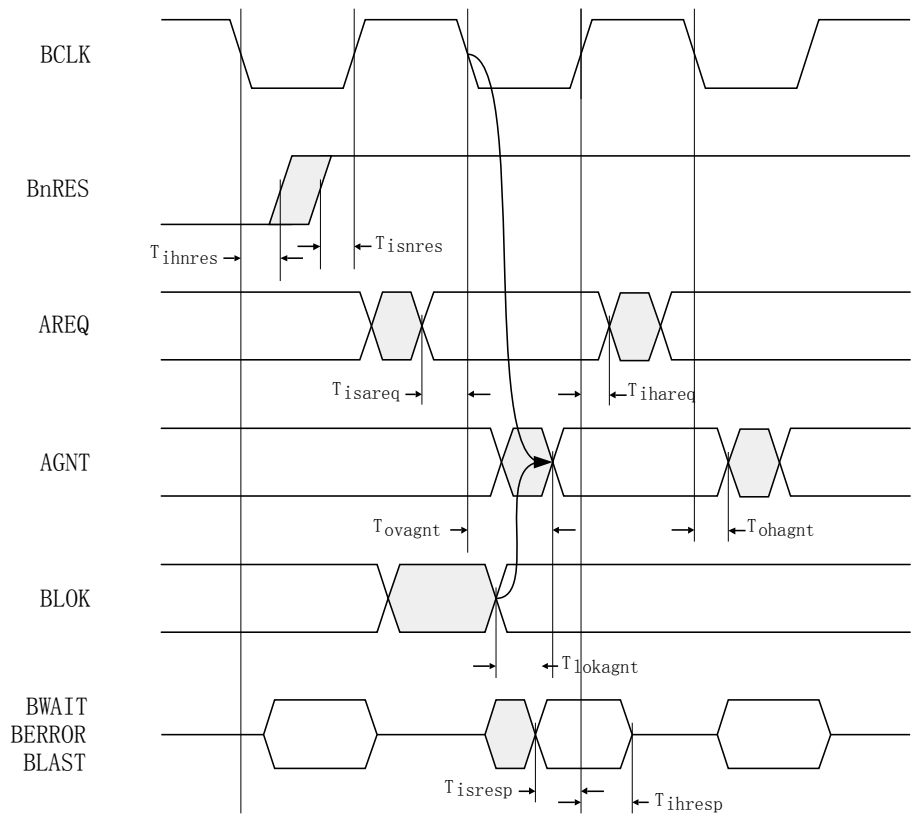


图 4.43 ASB 仲裁器参数

4.13.4 时序参数

下面的表格给出关于一个 ASB 仲裁器的时序参数：

- 表 4.14为输入信号；
- 表 4.15输出信号；
- 表 4.16为组合产生的输出。

表 4.14 ASB 仲裁器输入参数

参数	描述
$T_{clk\ell}$	BCLK 低电平时间
T_{clkh}	BCLK 高电平时间
T_{isnres}	在 BCLK 上升沿前 BnRES 失效的建立时间
T_{ihnres}	在 BCLK 下降沿后 BnRES 失效的保持时间
T_{isareq}	在 BCLK 下降沿前 AREQ 建立时间
T_{ihareq}	在 BCLK 上升沿后 AREQ 保持时间
T_{isresp}	在 BCLK 上升沿前 BWAIT 建立时间
T_{ihresp}	在 BCLK 上升沿后 BWAIT 保持时间

表 4.15 ASB 仲裁器输出参数

参数	描述
T_{ovagnt}	在 BCLK 下降沿后 AGNT 有效时间
T_{ohagnt}	在 BCLK 下降沿后 AGNT 保持时间

表 4.16 ASB 仲裁器组合参数

参数	描述
$T_{lokagnt}$	从 BLOK 有效到 AGNT 有效的延时

第5章 AMBA APB

这一章包含了高级微控制器总线体系（AMBA）的高级外设总线规范（APB）的以下部分：

- 关于AMBA APB；
- APB规范；
- 关于AMBA APB的组成部分；
- APB桥；
- APB从机；
- APB到AHB的接口；
- APB到ASB的接口；
- APB D版外设到APB 2.0 版外设的接口。

5.1 关于AMBA APB

高级外设总线（APB）是高级微控制器总线体系（AMBA）的总线层次的一部分并且为降低功率消耗和接口复杂性做了优化。

AMBA APB 应该被用来连接任何一种低带宽且不需要高性能通道总线接口的外设。

最新版本的 APB 确保了所有信号转变仅和时钟的上升沿有关。这中改进意味着 APB 外设可以方便的集成到任何设计流程，并有以下优点：

- 在高频操作时性能得到提高；
- 性能独立于时钟的占空比；
- 静态通过使用单时钟边沿简化了静态时序分析；
- 自动测试插件无需特别考虑因素；
- 许多特定用途集成电路（ASIC）库有较好的上升沿寄存器选择；
- 便于与基于周期的仿真器集成。

APB 的这些改变也使得它简化了到新的高级高性能总线（AHB）的接口。

5.1.1 基于AMBA的典型微控制器

一个典型的基于AMBA的微控制器包含一个高性能系统中枢总线，可以支持外部存储器带宽，包括CPU和其他直接存储器存取（DMA）设备，加上一个连接着较窄的APB总线，在该总线上挂接着较低带宽的外设。图 5.1表示了一个典型AMBA体系中的APB。

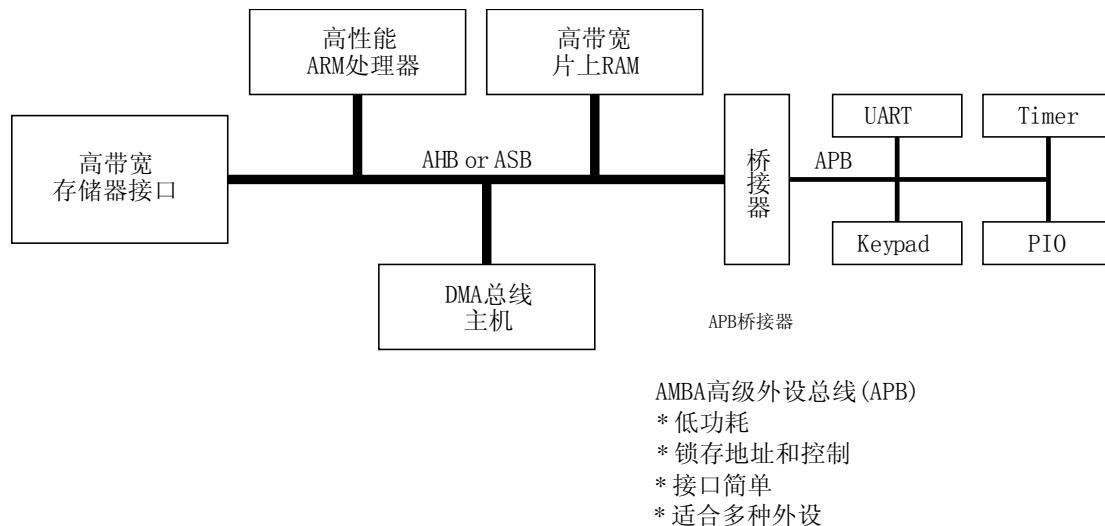


图 5.1 典型 AMBA 系统中的 APB

一个典型的总线包含一个测试接口控制器 (TIC) 允许模块组件测试系统总线和 APB 模块。

5.2 APB 规范

APB 规范以下列标题来描述：

- 状态图；
- 写传输；
- 读传输。

5.2.1 状态图

图 5.2 表示了状态图，可以用来表示外设总线的活动性。

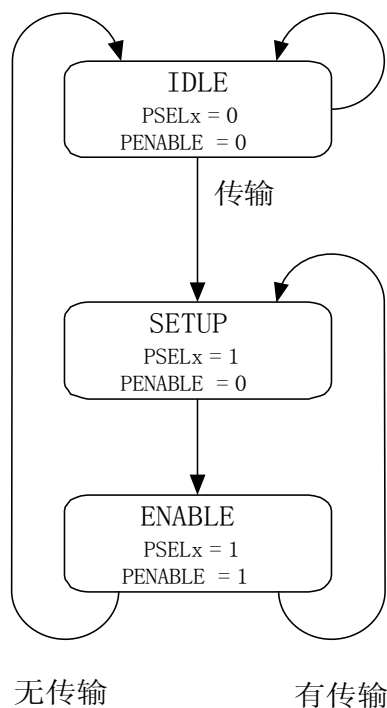


图 5.2 状态图

状态机运行在下面所描述的三个状态：

IDLE 外设总线的默认状态。

SETUP 当有传输要求时总线转移到 **SETUP** 状态，这时对应的选择信号，**PSELx** 被断言。总线仅在 **SETUP** 状态保持一个时钟周期并将总是在下一个时钟的上升沿移动到 **ENABLE** 状态。

ENABLE 使能信号 **PENABLE** 在 **ENABLE** 状态中被断言。地址、写和选择信号都在从 **SETUP** 到 **ENABLE** 状态的转变过程中保持稳定。

ENABLE 状态也仅持续一个周期并且在这个状态之后如果没有进一步的传输要求总线将返回 **IDLE** 状态。另外，如果紧跟着另一个传输那么总线移动到 **SETUP** 状态。地址、写和选择信号在从 **ENABLE** 到 **SETUP** 状态的转变期间有小跳变也是可以接受的。

5.2.2 写传输

图 5.3表示了基本的写传输。

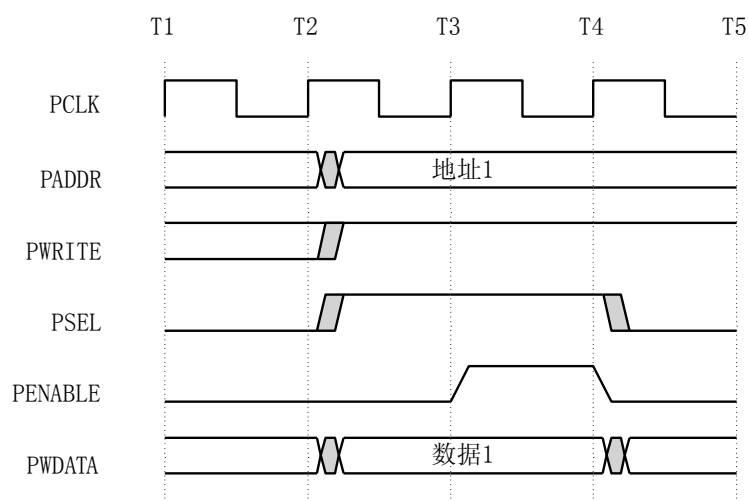


图 5.3 写传输

写传输开始于地址、写数据、写信号和选择信号在时钟上升沿之后的全部改变。传输的第一个时钟被称作 **SETUP** 周期。在下一个时钟边沿使能信号 **PENABLE** 生效，这表示 **ENABLE** 周期正在进行。地址、数据和控制信号全都在整个 **ENABLE** 周期保持有效。传输在这个周期结束时完成。

使能信号 **PENABLE**，将在传输结束时失效。选择信号也将变成低电平，除非当前传输之后紧跟着另一个到该外设的传输。

为了降低功率消耗地址信号和写信号将在传输之后不再改变，直到下一个传输发生为止。

协议仅要求在使能信号上有一个规则的转变（跳变）。在背靠背传输情况下选择和写信号可能有点跳变。

5.2.3 读传输

图 5.4表示了一个读传输。

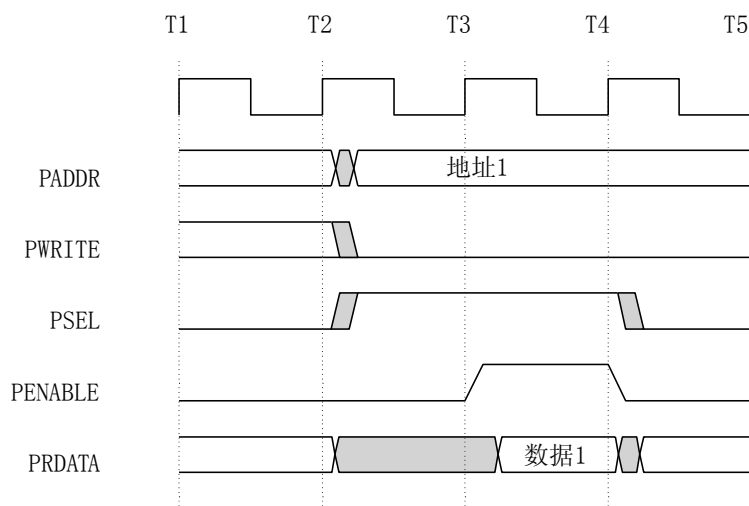


图 5.4 读传输

地址、写、选择和选通信号的时序都和写传输一样。在读传输的情况下，从机必须在 ENABLE 周期提供数据。数据在 ENABLE 周期末尾的时钟上升沿被采样。

5.3 关于AMBA APB的组成部分

下面的符号用作时序参数：

- T_{is} – 输入建立时间；
- T_{ih} – 输入保持时间；
- T_{ov} – 输出有效时间；
- T_{oh} – 输出保持时间。

5.4 APB桥

APB 桥是 AMBA APB 中的唯一总线主机。另外，APB 桥也是高级系统总线中的一个从机。

5.4.1 接口框图

图 5.5表示了APB桥的APB接口信号。

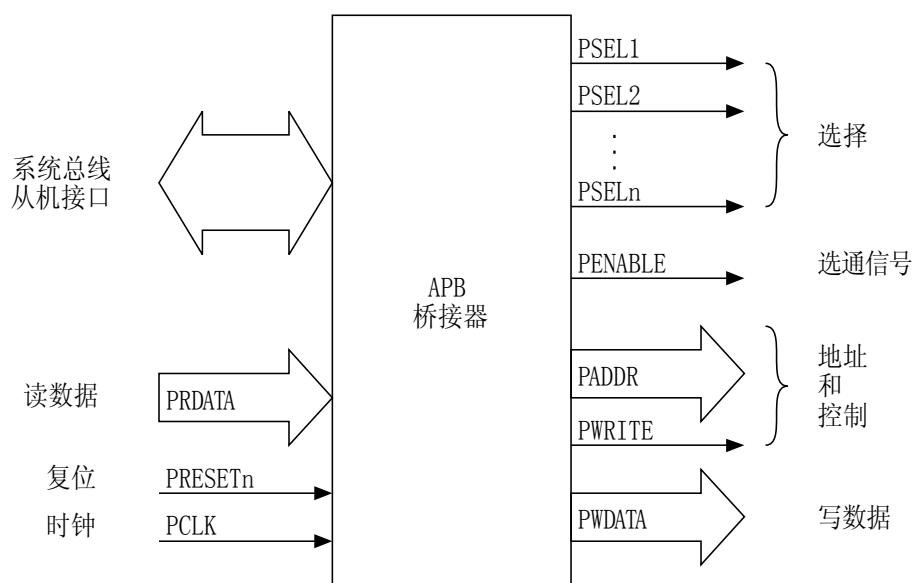


图 5.5 APB 桥接口框图

5.4.2 APB桥描述

桥接单元将系统总线传输转换成 APB 传输并实现下列功能：

- 锁存地址并使之在整个传输期间有效；
- 译码地址和产生一个外设选择信号，**PSELx**。在一个传输期间只有一个选择信号有效；
- 对于写传输驱动数据到 APB 上；
- 对于读传输驱动 APB 数据到系统总线上；
- 为传输产生一个时序选通信号，**PENABLE**。

5.4.3 时序图

图 5.6 表示了 APB 桥的时序参数。

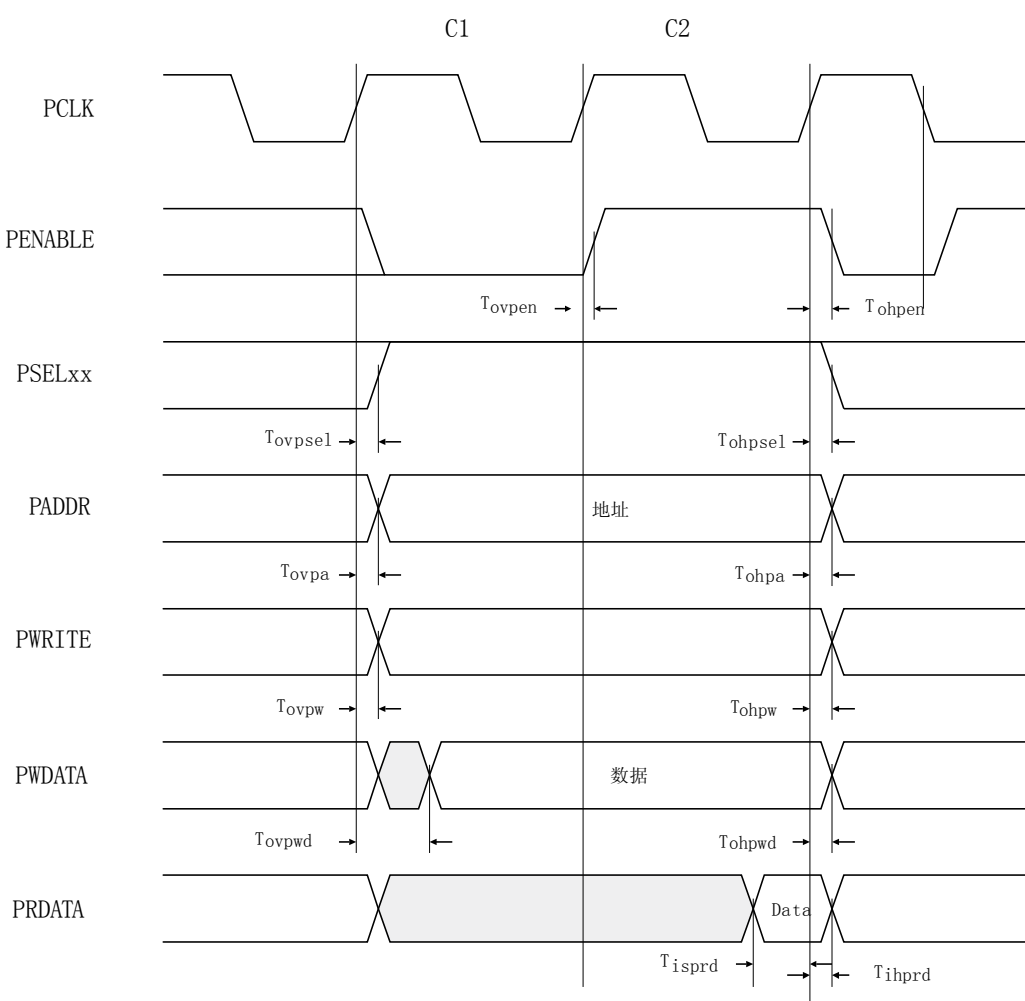


图 5.6 APB 桥传输

5.4.4 时序参数

给出关于APB桥的时序参数，表 5.1为输入信号，而表 5.2为输出信号。

表 5.1 APB 桥输入参数

参数	描述
T_{clki}	PCLK 低电平时间
T_{clkh}	PCLK 高电平时间
T_{isnres}	在 PCLK 上升沿前 PRESETn 失效的建立时间
T_{ihnres}	在 PCLK 下降沿后 PRESETn 失效的保持时间
T_{isprd}	对读传输，在 PCLK 上升沿前 PRDATA 建立时间
T_{ihprd}	对读传输，在 PCLK 上升沿后 PRDATA 保持时间

表 5.2 APB 桥输出参数

参数	描述
T _{ovpen}	在 PCLK 上升沿后 PENABLE 有效时间
T _{ohpen}	在 PCLK 上升沿后 PENABLE 保持时间
T _{ovpsel}	在 PCLK 上升沿后 PSEL 有效时间
T _{ohpsel}	在 PCLK 上升沿后 PSEL 保持时间
T _{ovpa}	在 PCLK 上升沿后 PADDR 有效时间
T _{ohpa}	在 PCLK 上升沿后 PADDR 保持时间
T _{ovpw}	在 PCLK 上升沿后 PWRITE 有效时间
T _{ohpw}	在 PCLK 上升沿后 PWRITE 保持时间
T _{ovpwd}	对写传输，在 PCLK 上升沿后 PWDATA 有效时间
T _{ohpwd}	对写传输，在 PCLK 上升沿后 PWDATA 保持时间

5.5 APB从机

APB 从机有一个即简单又灵活的接口。接口的准确实现将依赖于设计人员的设计风格
和许多不同的可能情况。

5.5.1 接口框图

图 5.7表示了APB从机的信号接口。

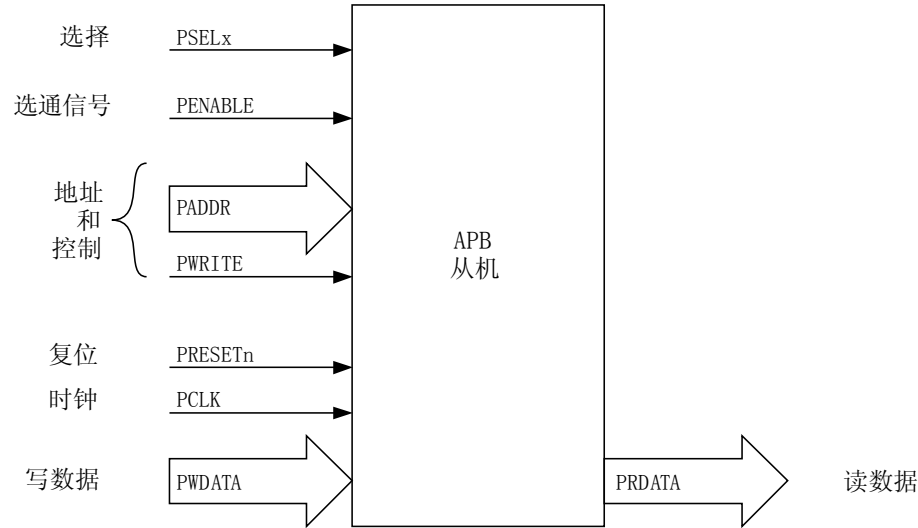


图 5.7 APB 从机接口描

5.5.2 APB从机描述

APB 从机接口非常灵活。
对写传输可以在下列要点时被锁存：

- 当 **PSEL** 为高时，在任意一个 **PCLK** 的上升沿；

- 当 **PSLE** 为高时在 **PENABLE** 的上升沿。

选择信号 **PSELx**，地址 **PADDR** 和写信号 **PWRITE** 可以组合来决定哪个寄存器由写操作来更新。

对读传输数据可以在 **PWRITE** 为低电平且 **PSELx** 和 **PENABLE** 都为高电平时被驱动到数据总线上。而此时 **PADDR** 被用来决定哪个寄存器应该被读取。

5.5.3 时序图

图 5.8表示了关于访问一个APB总线从机的时序参数。

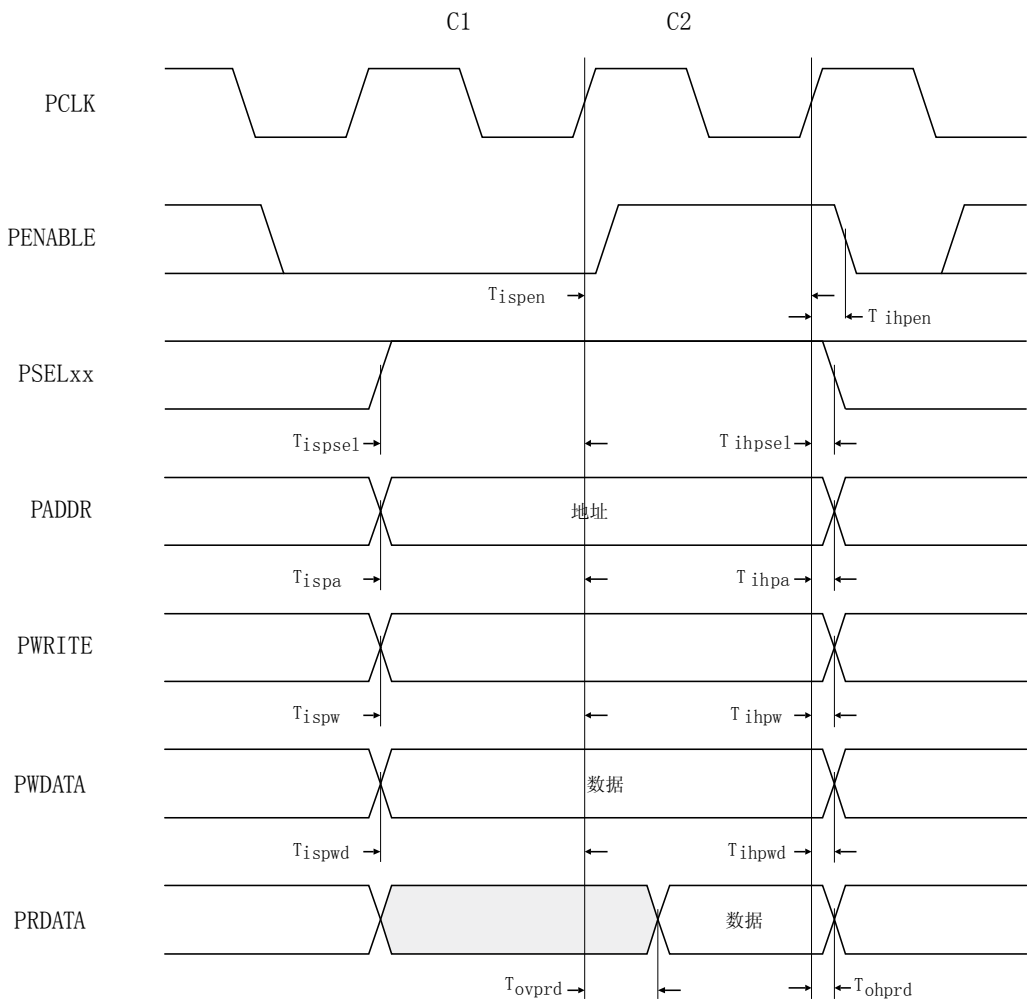


图 5.8 APB 从机传输

5.5.4 时序参数

下表给出关于APB从机的时序参数，表 5.3为输入信号而表 5.4为输出信号。

表 5.3 APB 从机输入参数

参数	描述
T _{clk_l}	PCLK 低电平时间
T _{clk_h}	PCLK 高电平时间
T _{isnres}	在 PCLK 上升沿前 PRESETn 失效的建立时间

续上表

参数	描述
T_{ihnres}	在 PCLK 下降沿后 PRESETn 失效的保持时间
T_{ispen}	在 PCLK 上升沿前 PENABLE 建立时间
T_{ihpen}	在 PCLK 下降沿后 PENABLE 保持时间
T_{ispsel}	在 PCLK 上升沿前 PSEL 建立时间
T_{ihpsel}	在 PCLK 上升沿后 PSEL 保持时间
T_{ispa}	在 PCLK 上升沿前 PADDR 建立时间
T_{ihpa}	在 PCLK 上升沿后 PADDR 保持时间
T_{ispw}	在 PCLK 上升沿前 PWRITE 建立时间
T_{ihpw}	在 PCLK 上升沿后 PWRITE 保持时间
T_{ispwd}	对写传输，在 PCLK 上升沿前 PWDATA 建立时间
T_{ihpwd}	对写传输，在 PCLK 上升沿后 PWDATA 保持时间

表 5.4 ASB 从机输出参数

参数	描述
T_{ovprd}	对读传输，在 PCLK 上升沿后 PRDATA 有效时间
T_{ohprd}	对读传输，在 PCLK 上升沿后 PRDATA 保持时间

5.6 APB到AHB的接口

AMBA APB 到 AHB 的接口描述如下：

- 读传输；
- 写传输；
- 背靠背传输；
- 三态数据总线的实现。

5.6.1 读传输

图 5.9阐明了一个读传输。

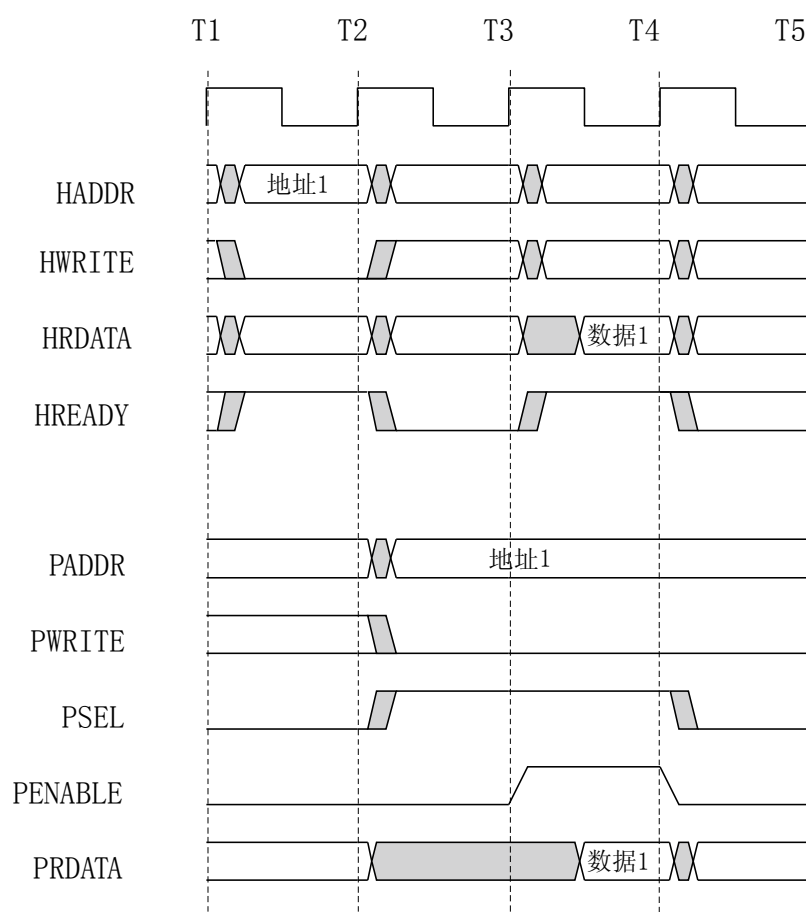


图 5.9 到 AHB 的读传输

传输在 AHB 上的时刻 T1 发起并且地址在 T2 被 APB 桥采样。如果传输是到外设总线那么该地址被广播并且产生合适的外设选择信号。在外设总线上的第一个周期成为 **SETUP** 周期，当 **PENABLE** 信号被断言后，该周期后跟随着 **ENABLE** 周期。

在 **ENABLE** 周期期间外设必须提供读数据。通常可能会将要读的数据直接发回到 AHB，总线主机在 **ENABLE** 周期结束时的时钟上升沿采样该数据，在图 5.9 中为时间 T4。

在时钟频率很高的系统中可能需要桥在 **ENABLE** 周期结束时寄存读数据然后桥在接下来的周期将该数据驱动回 AHB 总线主机。尽管对外设总线读传输而言这将要求一个额外的等待状态，但这允许 AHB 运行在较高的时钟频率，这也导致了系统性能的全面提高。图 5.10 表示了一次突发读传输。（图中）所有读传输都要求一个等待状态。

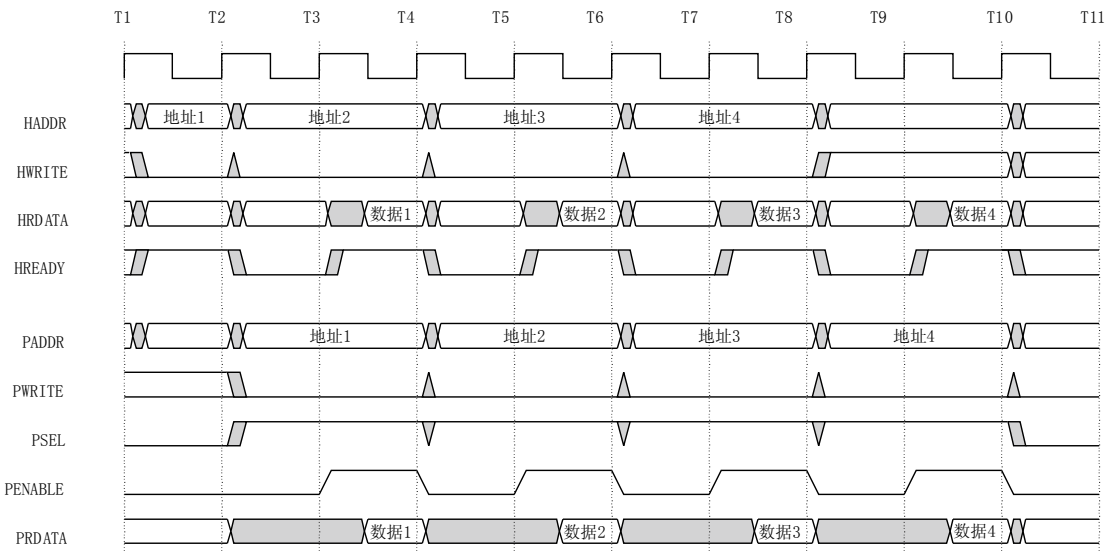


图 5.10 突发读传输

5.6.2 写传输

图 5.11表示了一个写传输

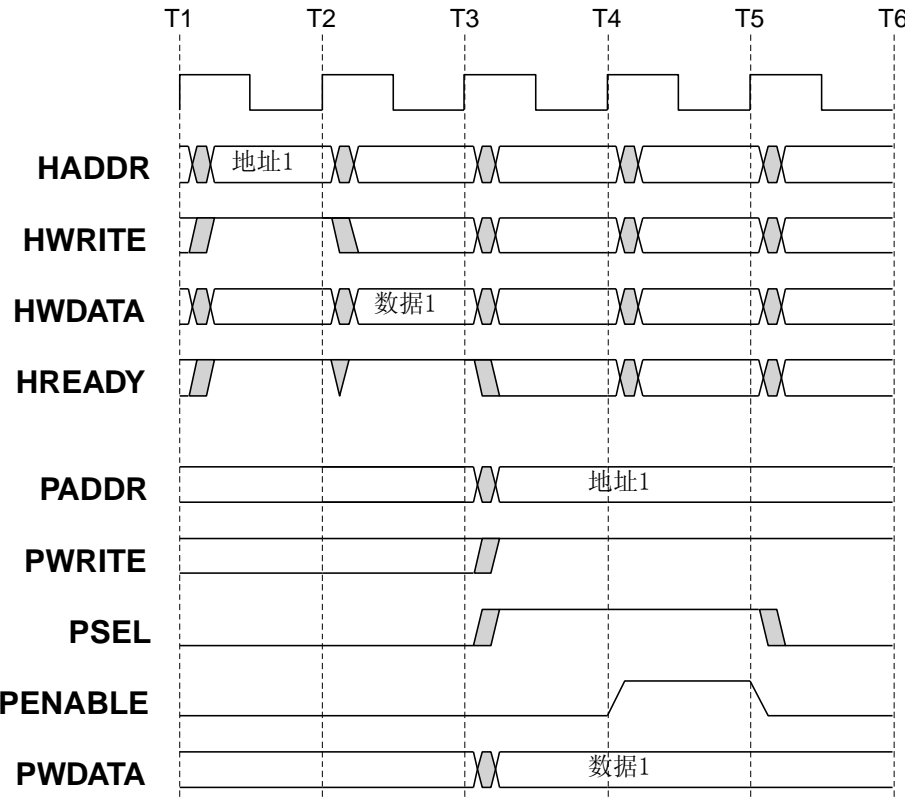


图 5.11 来自 AHB 的写传输

单个到 AHB 的写传输可以零等待状态发生。APB 桥负责采样传输的地址和数据并在 APB 的写传输期间保持这些数据值。

图 5.12表示了一个突发写传输。

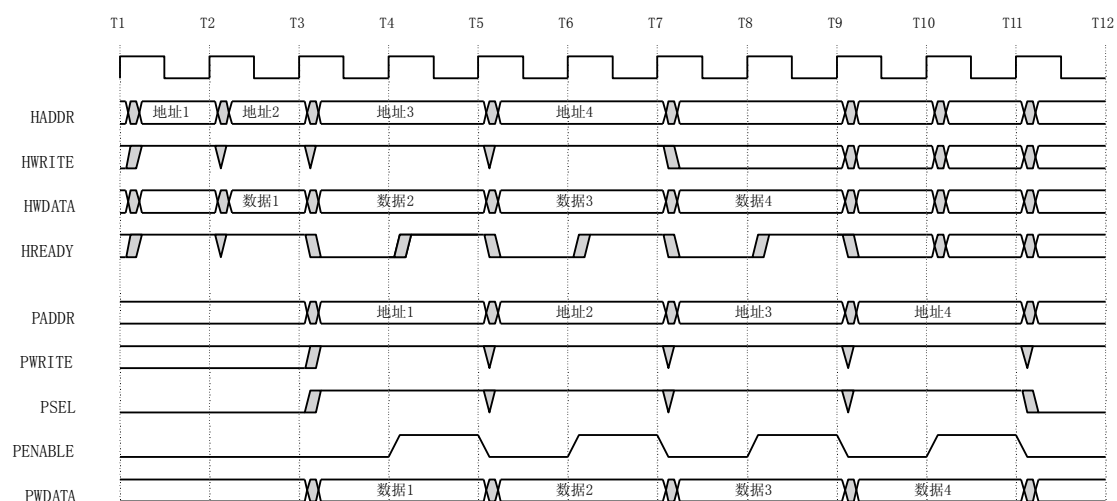


图 5.12 突发写传输

虽然第一个传输可以零等待状态完成,但之后到外设总线的传输将为每个传输的执行要求一个等待状态。

APB 桥需要包含两个地址寄存器,以便 APB 桥可以采样下一次传输的地址而同时当前传输继续在外设总线上(执行)。

5.6.3 背靠背传输

图 5.13表示了许多的背靠背传输。传输序列以一个写操作开始,之后跟随着一个读操作,然后是一个写操作,之后是一个读操作。

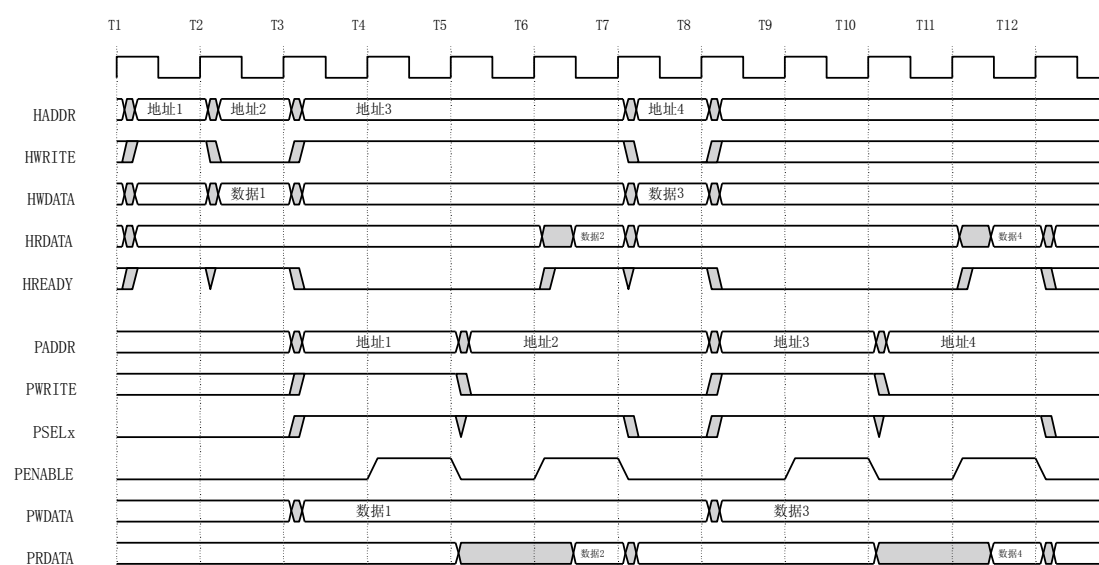


图 5.13 背靠背传输

图 5.13表示了如果一个读传输紧跟在一个写传输之后,那么需要 3 个等待状态来完成这次读操作。事实上,在基于处理器的设计中一个写传输后跟随着一个读传输并不经常发生因为处理器将在两个传输之间执行指令预取并且指令存储器不太可能挂接在 APB 总线上。

5.6.4 三态数据总线的实现

建议以分开的读和写数据总线来实现 AMBA APB，这允许使用一个多路复用总线或者或（OR）总线方案来互联许多 APB 总线上的从机。如果使用了三态总线那么读和写数据总线可以组合成单根总线，因为读和写数据绝不会同时发生。

图 5.14表示了不需要特别考虑的因素如果数据总线是使用三态缓冲来实现。如果数据总线在读传输的SETUP周期为三态并且只要总线在空闲状态那么周转的整个时钟周期总是在不同的数据驱动器之间发生。对突发写传输而言没有周转期因为APB桥将在每个传输的SETUP周期驱动数据，然而这非常合理因为APB桥是写传输唯一的数据总线驱动器因此不需要周转期。

图 5.14表示了读和写数据总线是如何成功的组合到单根三态数据总线的。

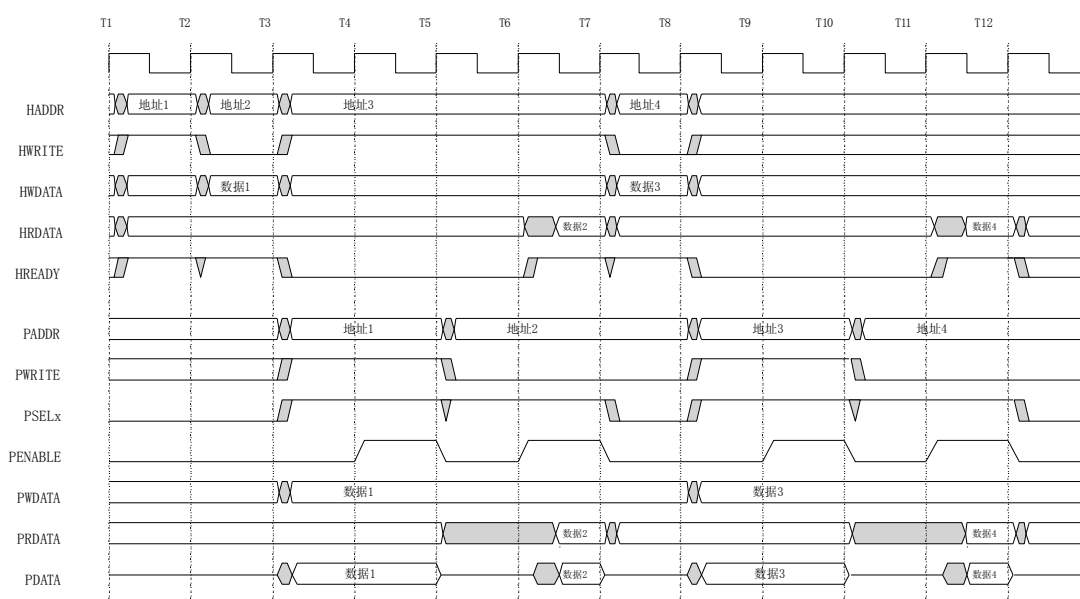


图 5.14 三态数据总线

5.7 APB到ASB的接口

AMBA APB 到 ASB 的接口描述如下：

- 写传输；
- 读传输。

5.7.1 写传输

图 5.15表示了如何构建一个从ASB到APB的接口。写传输可以零等待状态发生，然而对突发写传输而言需要一个附加的等待状态。

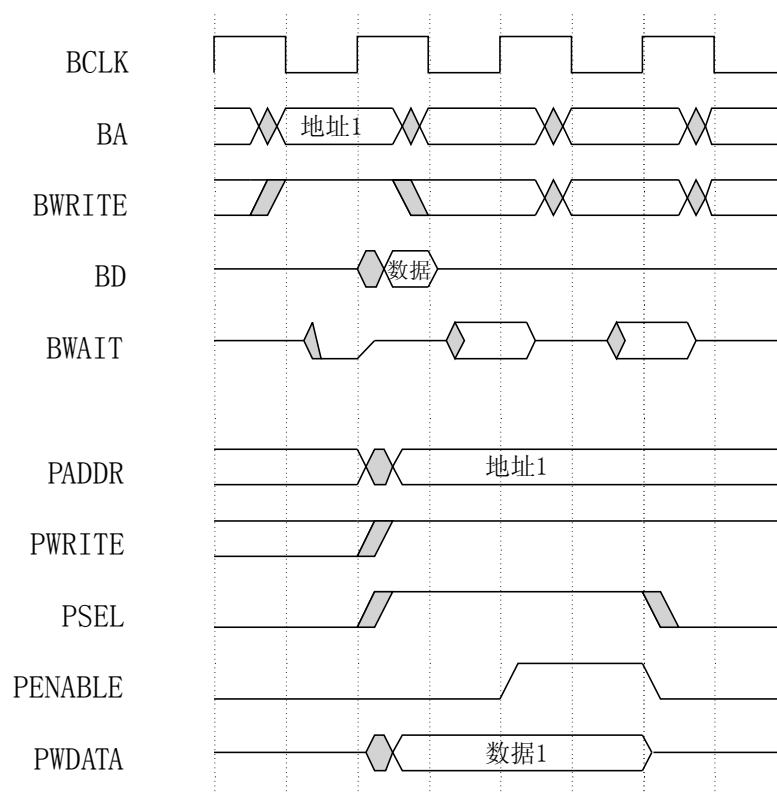


图 5.15 来自 ASB 的写传输

5.7.2 读传输

读传输将总是要求一个等待状态（见图 5.16）。在时钟频率较高的系统中可能需要插入一个附加的等待状态以确保读传输有足够的时间来通过APB桥并在ASB上有效。

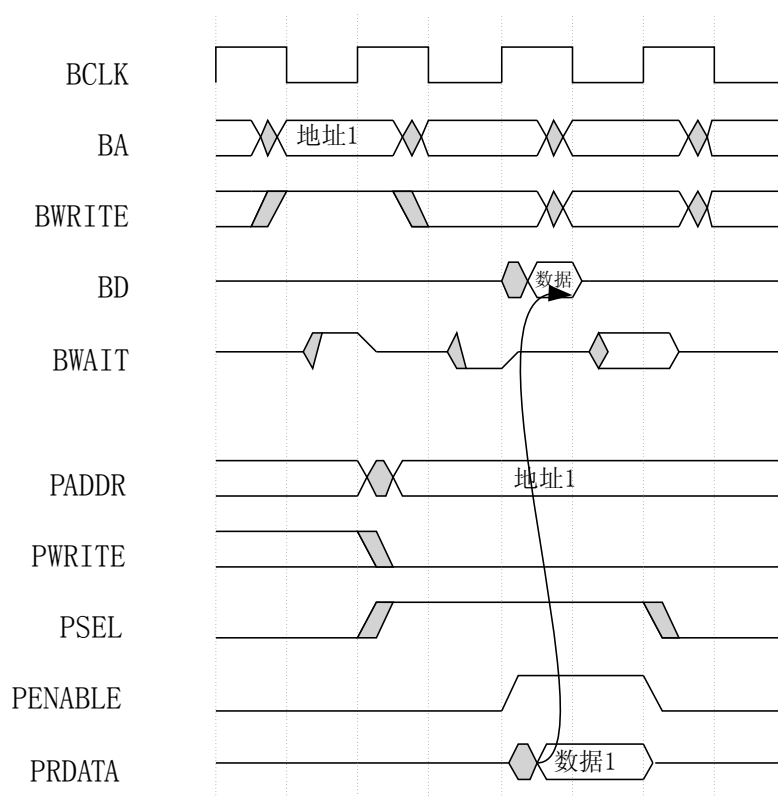


图 5.16 到 ASB 的读传输

5.8 APB D版外设到APB 2.0 版外设的接口

当使用组合外设时，一些外设被设计为 2.0 版本规范和其他则被设计为早期版本，建议使用一个 2.0 版本的桥，这样早期版本的外设可以被修改以用在新版本的桥上。

这部分说明了一个 D 版本外设可以如何被转换到最新版本的 APB 上的。如果许多外设都被转换那么在单个集中模块中执行转换的效率将更高。

在 D 版本和 2.0 版本 APB 规范之间有两个基本的不同：

- 选通信号的时序被称为使能信号；
- 读数据在哪个时间点被采样。

为了快速决定一个外设是被设计为D版本或者 2.0 版本规格，需注意该外设是否有一个 **PSTB**输入（在D版本的情况）或者一个**PENABLE**输入（在 2.0 版本的情况）。图 5.17表示了连接到一个现有的D版本外设所需要的两个步骤。

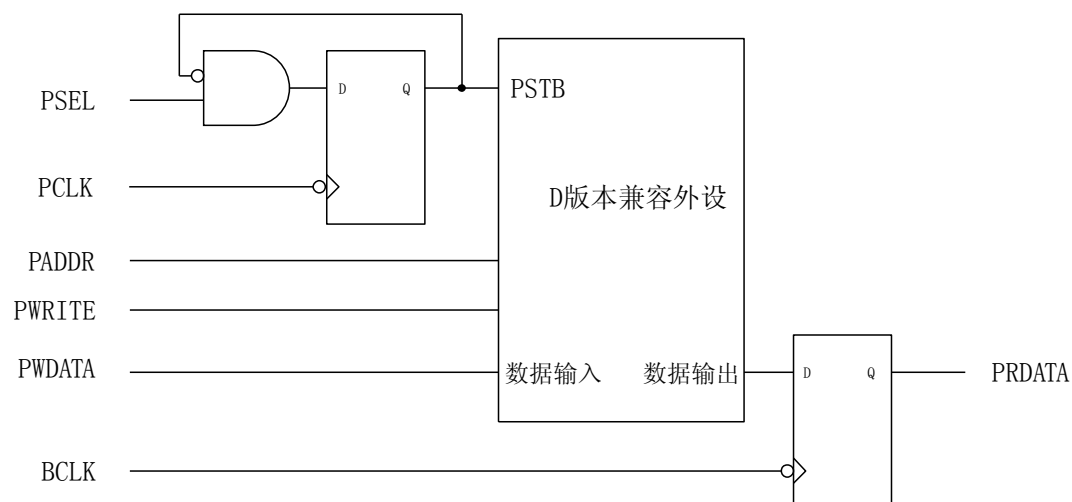


图 5.17 连接一个 D 版本的外设

首先，**PSEL** 信号可以用来产生一个 **PSTB** 信号。**PSTB** 信号的一个反馈版本可以用来确保信号只生效一个时钟周期。

可能要求的连接的第二步是一个下降沿触发寄存器或者是透明锁存来自外设的输出数据（读数据）。这仅被要求在如果外设在下沿之后改变输出数据的情况下。

第6章 AMBA测试方法

这一章描述了和 AMBA 模块设计一起使用的测试接口。它包含以下部分：

- 关于AMBA测试接口；
- 外部接口；
- 测试向量类型；
- 测试接口控制器；
- AHB测试接口控制器；
- AMBA ASB测试序列样例；
- ASB测试接口控制器；
- AMBA AHB测试序列样例。

6.1 关于AMBA测试接口

AMBA 测试的本质允许系统中的个体模块被隔离测试。每个模块经过设计因此能仅使用来自总线的传输而被测试并且不依赖于任何其他系统元素之间的相互作用。因此需要访问没有直接连接到总线上的外设的输入和输出而这由一个测试装具来提供。

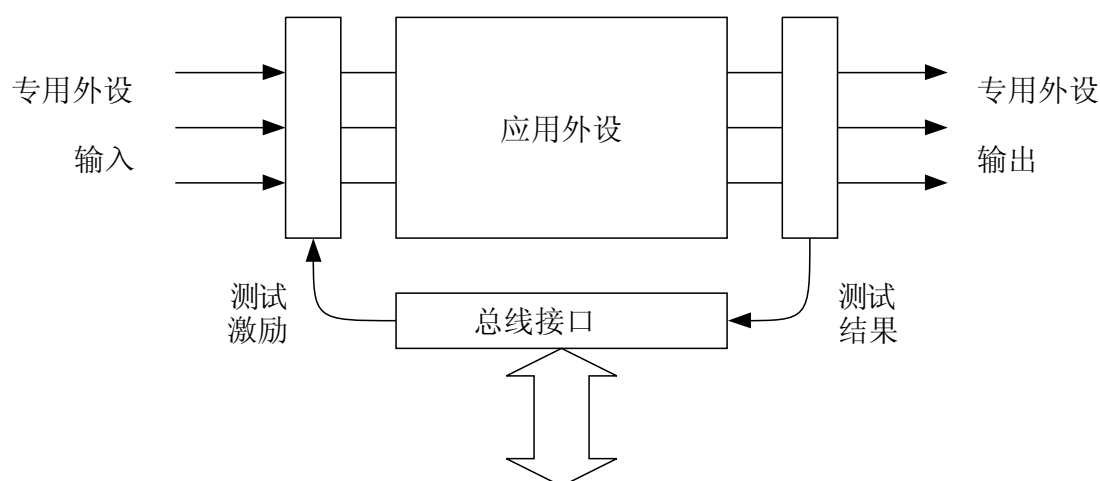


图 6.1 外设测试装具

系统中要求一个低门数的测试接口控制器（TIC）总线主机模块以允许外部应用的测试向量被转换成内部总线传输。

TIC 使用一个最少的三线握手机制来控制测试向量的应用而外部总线接口（EBI）的数据通道被用来提供高速 32 位并行向量接口。

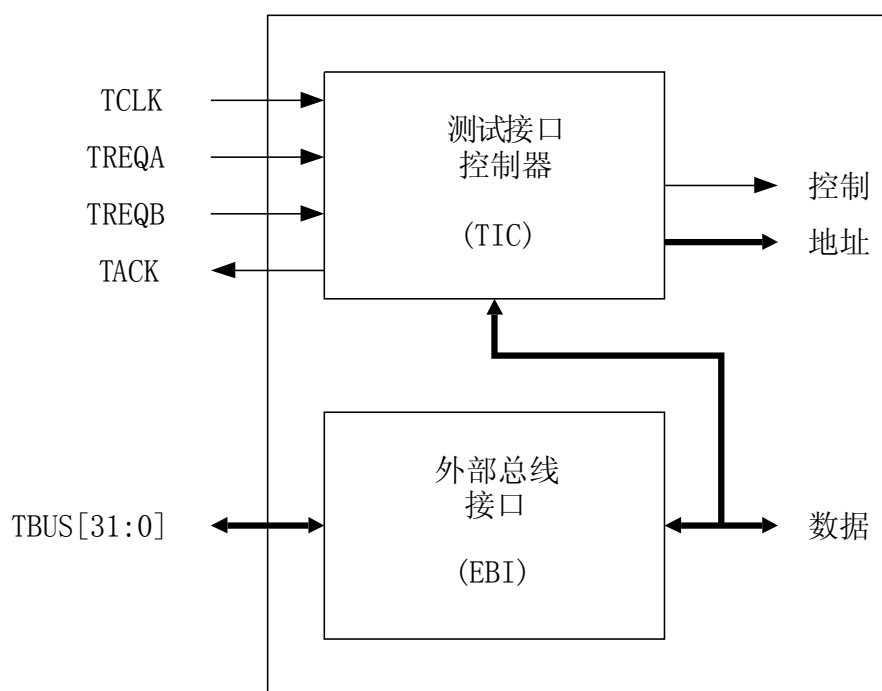


图 6.2 TIC 和外部总线接口的相互配合

为支持这种测试向量应用的方法在测试访问期间必须有一个 32 位双向端口可用。对于一个带 32 位外部数据总线接口的系统而言这比较简单。16 位和 8 位数据总线设计要求，例如，16 或者 24 根地址线被重新配置为测试模式访问的双向测试端口信号。

6.2 外部接口

外部测试接口由以下部分组成：

- 测试时钟；
- 三个控制信号；
- 一个 32 位测试总线。

只要求两个专用信号引脚（**TREQA** 和 **TACK**）来控制进入和退出测试模式。剩下的信号可以由复用现有的设备引脚来提供。

6.2.1 测试总线请求A

TREQA 是测试总线请求 A 的输入信号并要求为一个专用设备引脚。

在常规系统操作期间 **TREQA** 信号被用来请求进入测试模式。这将引起测试总线变成三态，允许测试向量的应用。

在测试期间该信号和 **TREQB** 一起结合使用来表示在随后周期将被应用的测试向量的类型。

6.2.2 测试总线请求B

TREQB 是测试总线请求 B 的输入信号。

在测试期间该信号和 **TREQA** 一起结合使用来表示在随后周期将被应用的测试向量的类型。

6.2.3 测试应答

TACK 是测试总线响应输出信号并且要求为一个专用设备引脚。

测试总线响应信号给出测试总线被授予的外部指示并且也在测试访问已经完成时给出指示。当 **TACK** 为低时当前测试向量必须被扩展到 **TACK** 变为高电平为止。**TREQA** 和 **TREQB** 信号仅在 **TACK** 为高电平时被 TIC 采样。

表 6.1和表 6.2表示了**TREQA**、**TREQB**和**TACK**信号的操作。这些信号根据是否进入测试模式有不同的功能。

表 6.1 在常规操作期间的测试控制信号

TREQA	TREQB	TACK	描述
0	0	0	常规操作
1	0	0	请求进入测试模式
0	1	0	保留（给外部主机请求）
-	-	1	已进入测试模式

表 6.2 在测试模式期间的测试控制信号

TREQA	TREQB	TACK	描述
-	-	0	当前访问未完成
1	1	1	地址向量、控制向量或者周转向量
1	0	1	写向量
0	1	1	读向量
0	0	1	退出测试模式

6.2.4 测试时钟

TCLK 是测试时钟输入信号。

在测试模式，内部总线时钟是由外部 **TCLK** 时钟源驱动。这个引脚可以是常规的晶振时钟源或者是一个端口替代信号。系统总线时钟在常规和测试模式选择期间禁止抖动（脉冲干扰）。

进入测试模式时 TIC 通过使 **TACK** 信号生效来表示它已经切换到测试时钟输入。

6.2.5 测试总线

TBUS[31: 0]是 32 位双向测试端口。

测试总线被用作一个输入以应用于地址、控制和写向量。对读向量测试总线被用作一个设备输出。测试接口协议确保当改变测试总线的方向时总是提供一个周转周期。

6.3 测试向量类型

有 5 种和测试接口相关的测试向量：

- 地址向量；
- 写向量；
- 读向量；

- 控制向量；
- 周转向量。

地址向量、控制向量和周转向量都由 **TREQA** 和 **TREQB** 信号上的相同值来表示。下面的规则可以被用来确定被使用的测试向量类型。

- 当单个地址/控制向量被使用时则是一个地址向量；
- 当一个突发地址/控制向量被使用时则都为地址向量，除了最后一个控制向量之外；
- 一个读向量，或者突发读向量，总是被一个周转向量跟着。这是周转向量仅有的发生情形。ASB 版本的测试接口要求单个周转向量，而 AHB 版本则要求两个。

6.4 测试接口控制器

测试接口控制器（TIC）是一个接收来自外部测试总线（**TBUS[31: 0]**）的测试向量的总线主机，并且指示总线传输。TIC 锁存地址向量并且在要求时增加地址以允许读和写突发测试向量。

6.4.1 测试传输参数

当进入测试模式时默认 TIC 总线主机的操作是：

- 32 位传输宽度；
- 特权系统访问。

这足够测试许多嵌入式系统设计并且最小化了片上测试支撑逻辑。在系统要求上述控制信号动态改变的情况下，使用一个控制向量机制来更新 TIC 内的控制信号。

控制向量的第 0 位被用来表示控制向量是否有效。因此，如果一个控制向量以第 0 位为低的方式被使用，该向量将被忽略并且不会更新控制信息。这种机制允许第 0 位为低的地址向量可以被使用许多周期而不用更新控制信息。

6.4.2 增量地址

为了支持使用测试接口的突发访问 TIC 可以支持总线地址增量。增加的地址位数目取决于经由测试接口要求的最大突发访问长度。这是独立于系统的，但是典型的实现将使用一个 8 位地址增量器，允许了使用字传输的突发访问（能够）达到 1KB 的边界。

控制向量也提供了一个机制来使能和禁能 TIC 内部的地址增量器。这允许突发访问增加地址，因此可以被用来测试内部 RAM。此外，地址增量可以被禁能，这样一次突发的连续访问将出现在同一个地址上，因此可以被用来连续读取单个外设寄存器。

如果传输大小是动态变更的那么任何支持突发模式访问的地址增量器必须能够支持以字节、半字和字偏移量方式的增量，因此要求适应地址增量器逻辑。

地址增量器默认是禁能的并且在使用前必须用一个控制向量来使能。

6.4.3 进入测试模式

在常规操作模式中 **TREQA** 将为低电平，表示不要求测试访问并且测试总线将被用作常规操作需要，通常是成为外部总线接口的一部分。进入测试模式允许测试向量由外部使用，这会引发在内部总线上的传输。

下面是进入测试模式要求的顺序：

- 1、 **TREQA** 生效以请求测试总线访问；
- 2、 当 TIC 被授予内部总线时进入测试模式并且这由 **TACK** 信号生效来表示；

- 3、 在这时 **TCLK** 将成为内部时钟信号源；
- 4、 当已经进入测试模式时 **TREQB** 生效以便初始化一个地址向量。

TIC 将不执行任何内部传输直到一个有效的地址向量被应用为止。

不要求一个同步测试器来给总线查询 **TACK**。通常 **TREQA** 信号将生效最小的一段周期数量来保证获取访问总线（最长等待状态外设访问的完成或者所有总线主机完成它们当前指令的最大周期数目）。

6.4.4 地址向量

地址向量必须在任何读或者写操作发生前被使用。下面是使用一个地址向量要求的顺序：

- 1、 **TREQA** 和 **TREQB** 都被断言为高电平，表示下一个周期为地址向量；
- 2、 在下一个周期地址被应用到 **TBUS[31: 0]**，而 **TREQA** 和 **TREQB** 都改变以反映接下来测试向量的类型。

在一些高速系统中可能必须连续使用一个地址向量以上，以允许地址有足够时间从外部测试总线传播到内部地址总线。在这样的情况下 **TIC** 可以取消地址向量的第一个周期，强迫使用地址向量的第二个周期。

6.4.5 控制向量

控制向量总是在一个地址向量序列的最后并且被用来更新 **TIC** 内的控制信息。序列如下：

- 1、 **TREQA** 和 **TREQB** 都被断言为高电平，指示下一个周期为一个地址向量；
- 2、 在下一个周期地址被应用到 **TBUS[31: 0]**。**TREQA** 和 **TREQB** 都保持高电平因为控制向量将在接下来的周期出现；
- 3、 在下一个周期控制信息被应用到 **TBUS[31: 0]**，而 **TREQA** 和 **TREQB** 改变以反应下一个周期测试向量的类型；
- 4、 最终传输出现在内部总线上。

6.4.6 写测试向量

一旦成功进入测试模式，读和写操作可以通过测试接口来执行。为了内部执行一个写操作需要提供一个地址，之后跟随着写数据。

写传输中使用的地址将取决于前一个向量而一个写向量可以在下列情形之后发生：

- 单个地址向量；
- 一个地址/控制向量序列；
- 另外一个写测试向量，形成一个写突发；
- 在单个读或者突发读之后的一个周转向量。

当一个内部总线传输通过插入等待状态被扩展时这由 **TACK** 信号变成低电平来外在表示。在等待条件期间 **TREQA** 和 **TREQB** 应该改变来表示当前向量完成之后将会跟随的向量类型。然而，需要重点注意这种情况下一个写向量的数据应该继续被应用到 **TBUS[31: 0]**。

6.4.7 读测试向量

和写向量的方式相同，读测试向量可以跟着许多不同向量之后，如下面所列，并且用作传输的地址将取决于这之前的向量：

- 单个地址向量；

- 一个地址/控制向量序列；
- 另一个读测试向量，形成一个读突发；
- 单个写或者突发写。

一个读，或者读突发，之后必须跟着一个周转向量以阻止外部 **TBUS** 信号上的总线冲突。至于一个写向量，如果外部传输被扩展那么这由 **TACK** 信号变低来外在表示。读数据不应该被外部采样直到内部传输已经完成为止。

6.4.8 突发向量

多个写向量或者读向量可以结合在一起形成突发向量。这可以通过移除每个读或写向量相关地址向量的需要来让测试向量能够应用在更快的速度。

突发传输可以使用增量地址或者静态地址，取决于 **TIC** 是否包含一个已使能的地址增量器。没有地址增量器的话 **TIC** 将执行一个到常量地址的无顺序传输。

如果 **TIC** 包含一个已使能的地址增量器那么用作每个连续传输的地址将增加合适的数量，由传输大小来指示。

6.4.9 改变突发的方向

改变一次突发的传输方向是可能的，从读到写或者写到读。

如果是从读到写的改变则需要插入一个周转向量。这不会加载一个新的地址，但是会内部引起一个新的突发的开始以允许内部从机来检测已经被更改的突发方向。

6.4.10 退出测试模式

使用下列顺序（步骤）来退出测试模式：

- 1、 使用一个单周期的地址向量，这确保了任何内部传输的完成；
 - 2、 **TREQA** 和 **TREQB** 都被驱动为低电平来表示测试模式已经退出；
 - 3、 当测试接口已经被配置为常规系统操作时 **TACK** 将变低来表示测试模式已经退出；
- 测试模式能够干净利落地进入和退出很重要因此可以执行在系统操作期间诊断测试。

6.5 AHB测试接口控制器

下面的状态图表示了 **TIC** 的操作。

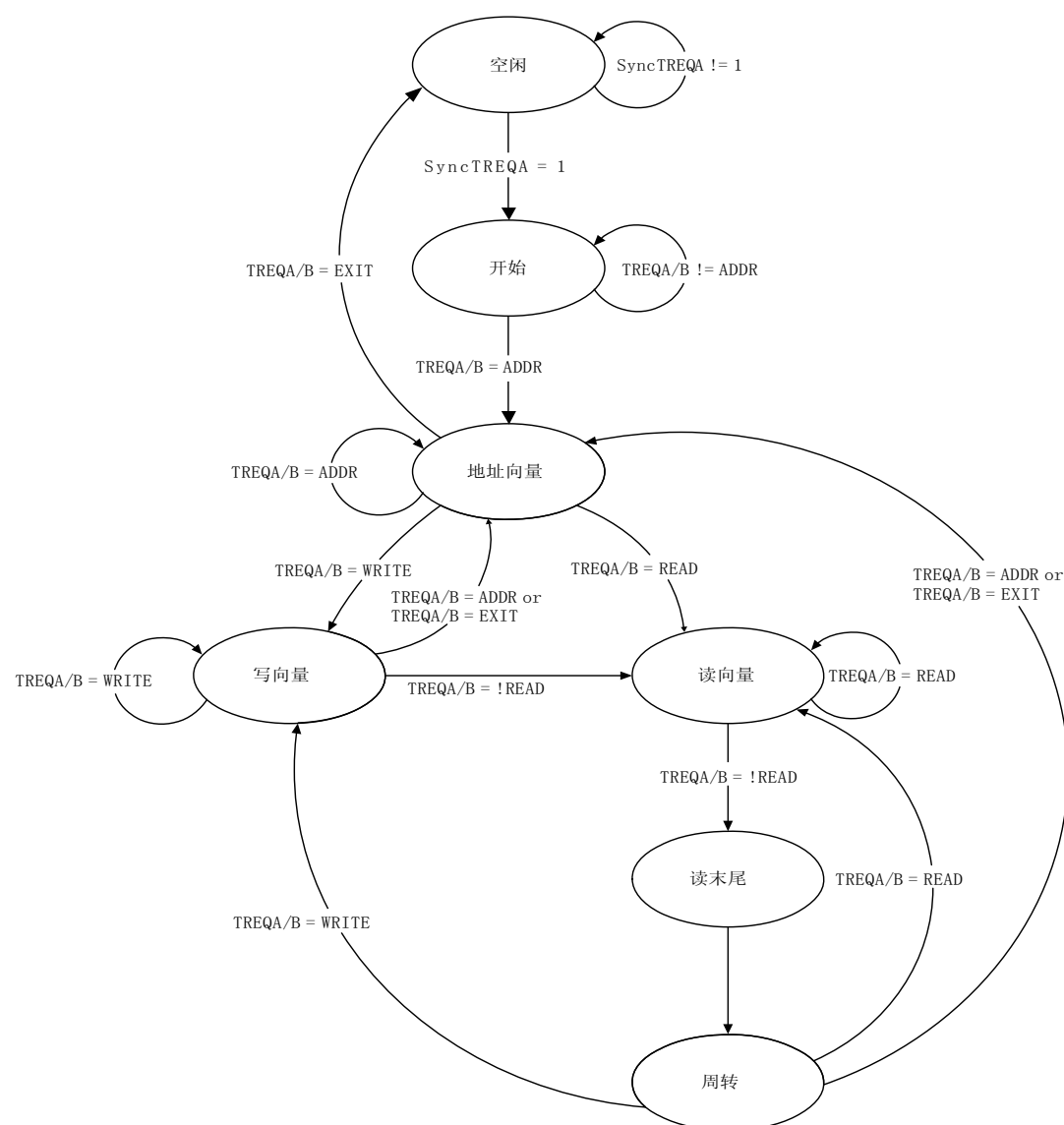


图 6.3 测试接口控制器状态图

下面的要点描述了 TIC 状态图的操作：

- 在复位时 TIC 在空闲状态并不在请求使用 AHB。当在空闲状态时 **TACK** 被驱动为低以表示测试接口不能被使用；
- **TACK** 信号被用来控制所有状态机的所有处理，除从空闲到开始的转变之外。在所有其他情况下如果 **TACK** 信号为低则状态机保持相同状态；
- **TREQA** 信号被用来从空闲状态移动到开始状态。这已经从以前的规范中修改过来，以前的规范要求 **TREQA** 为高电平且 **TREQB** 为低电平，好处是仅使用 **TREQA** 来从常规操作移动到测试模式成为可能；
- 在一些系统实现中需要从一个内部时钟源选择到一个用在测试模式期间的外部时钟 **TCLK**。当 **TREQA** 先变为高电平时这可以被用作时钟源应该被改变的指示并且一个表示时钟选择已经成功发生的返回信号可以被用来阻止阻止进入开始状态直到测试时钟正在使用时为止；

- 如果时钟选择（开关）被使用那么可能 **TREQA** 在进入测试模式之前异步于片上时钟因此使用一个同步器来产生一个同步版本的 **TREQA** 以控制从空闲状态到开始状态的移动；
- 使用开始状态以确保被应用的第一个向量是一个地址向量来阻止在地址被初始化之前一个读或者写向量的发生。开始状态仅在当 **TREQA/B** 表示一个地址向量且接下来的状态是地址向量时退出；
- 在地址向量状态中 **TIC** 寄存 **TBUS** 上的地址。地址向量状态被用于地址和控制向量，因此要求额外的逻辑来决定 **TBUS** 上的值是否应该被当作一个地址还是一个控制向量。如果前一个周期是一个地址向量且接下来的周期不是一个地址向量（由 **TREQA/B** 表示）那么当前周期是一个控制向量；
- 在地址向量状态可能会停留许多周期，但是通常一个地址向量将不是被一个读就是写传输跟着；
- 如果一个写传输正被执行那么 **TIC** 移动到写向量状态同时它初始化总线上的传输并且可以通过保持在写向量状态来执行多个写传输。通常写向量将被一个地址向量跟着，然而也可能直接移动到读传输，通过移动到读向量状态；
- 当执行了一个读、或者一个突发读时 **TIC** 进入读向量状态。这个状态表示 **TIC** 正开始一个在总线上的读传输并且到下一个周期为止读数据才会出现。当读向量状态是初次进入时 **TBUS** 将为三态，但在读向量状态中会被驱动更多的周期；
- 所有读向量之后必须跟着两个周转向量。在这些周期中的第一个中 **TIC** 将移动到读末尾状态，在这期间传输的最后一次读将完成且将被驱动到外部 **TBUS** 上。在读末尾状态期间没有内部传输将被启动且 **TIC** 将在总线上执行空闲传输；
- 紧跟着读末尾状态之后 **TIC** 移动到周转状态，在这期间外部的 **TBUS** 将为三态。通常周转状态之后将跟着一个地址向量，但是也可能立刻进入到一个写向量或者另外一个读（向量）；
- 退出测试模式测常规方法是回到地址向量状态之后设置 **TREQA/TREQB** 都为低来回到空闲状态并且有效地退出了测试。事实上，在任何时刻都可以通过设置 **TREQA** 和 **TREQB** 都为低来退出测试模式最终这会导致 **TIC** 退出测试。

注：当应用 **TIC** 向量时理论上可能会断言 **HLOCK** 的输出并且之后退出测试。如果这种情况发生了且之后 **TIC** 在常规操作下被授予总线这会有有效的锁定总线。**TIC** 内没有提供保护措施来阻止这种情况的发生。

6.5.1 控制向量

控制向量被包含在 **TIC** 内来确定它能执行的传输类型。控制向量被用来设置 **HSIZE**、**HPROT** 和 **HLOCK** 的值。

当进入测试模式时默认的 **TIC** 总线主机操作是：

- 32 位传输宽度 – **HSIZE[1: 0]**表示字传输；
- 特权系统访问 – **HPROT[3: 0]**表示特权数据访问，无快速缓冲且无缓冲。

控制向量的位 0 被用来表示控制向量是否有效。因此，如果一个控制向量被应用时位 0 为低，这个向量将被忽略并不会更新控制信息。这种机制允许位 0 为低的地址向量被应用许多周期而不用更新控制信息。

虽然默认设置足够测试许多嵌入式系统设计，但是控制向量也可以被用来改变传输的控制信号和确定 **TIC** 是否应该产生固定地址或者增量地址。

表 6.3 定义了控制向量的位的安排。控制向量位定义被设计为和早期 **TIC** 版本保持向下兼容因此并不是所有的控制位都是有定义的。

表 6.3 控制向量位定义

位的位置	描述
0	控制向量有效
1	保留
2	HSIZE[0]
3	HSIZE[1]
4	HLOCK
5	HPROT[0]
6	HPROT[1]
7	地址增量使能
8	保留
9	HPROT[2]
10	HPROT[3]

没有控制 **TIC** 能够执行的突发类型的机制而仅支持未定义长度的增量突发。**TIC** 仅支持 8 位、16 位和 32 位传输因此 **HSIZE[2]** 不能被更改且必须为低。

为了支持使用测试接口的突发访问测试接口控制器可以支持总线地址的增量。**TIC** 增加 8 个地址位则这个增量器能够覆盖的地址范围取决于被执行的传输大小。

控制向量提供一个机制来使能和禁能 **TIC** 内的地址增量器。这允许到增量地址的突发访问，因此可以被用来测试内部 **RAM**。另外，地址增量可以被禁能这样一个连续访问的突发可以出现在相同的地址上，因此可以被要求用来连续读单个外设寄存器。

如果 **HSIZE[1: 0]** 是动态改变的那么任何支持突发模式访问的地址增量器必须能够支持字节、半字和字偏移的增量，因此要求适应地址增量器逻辑。

地址增量器默认是禁能的且必须在使用之前用一个控制向量来使能。

注：控制向量最初是被用来改变和地址总线有相同时序的信号。然而控制向量也允许锁定要改变的信号，这实际上在锁定传输开始前被要求。如果 **HLOCK** 信号在测试期间被使用那么在它被要求之前应该被设置为一个传输。在 **HLOCK** 信号时序上的不同可能在一些情况下会在事实上应该被锁定的序列前和后都引起一个被锁定附加的传输。

6.6 AMBA AHB测试序列样例

AHB 测试序列的样例用下列标题来描述：

- 进入测试模式；
- 写测试向量；
- 读传输；
- 控制向量；
- 突发向量；
- 读到写和写到读；
- 退出测试模式。

6.6.1 进入测试模式

在常规操作模式 **TREQA** 将为低电平，表示没有要求测试访问且测试总线将被用作常规操作需要，通常会外部总线接口的一部分。进入测试模式允许测试向量被外部应用，这会引起在内部总线上的传输。

下面的顺序，如图 6.4所示，是为了进入测试模式而要求的：

- 1、 **TREQA** 被断言来请求测试总线访问；
- 2、 当 **TIC** 被授予内部总线时进入测试模式且这由 **TACK** 信号被断言来表示；
- 3、 在这时 **TCLK** 将成为内部 **HCLK** 信号的时钟源；
- 4、 当进入测试模式时 **TREQB** 被断言以表示一个地址向量；
- 5、 **TIC** 不会执行任何内部传输直到一个有效的地址向量被应用为止。

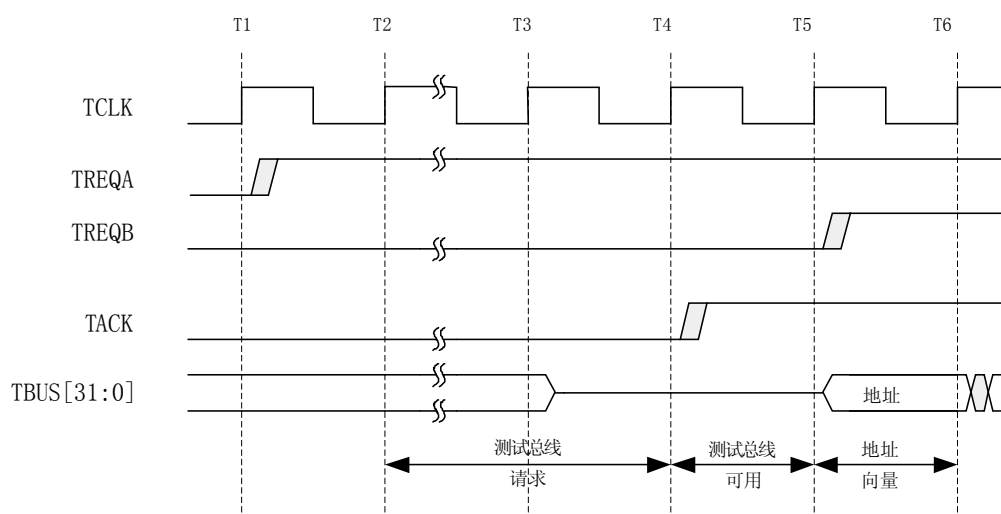


图 6.4 测试发起序列

并不要求一个同步测试器来为总线查询 **TACK**。

通常 **TREQA** 信号被断言最小的周期数量来保证访问总线(最长等待状态外设访问的完成或者所有总线主机完成它们当前指令的最大周期数目)。

6.6.2 写测试向量

图 6.5表示了当应用一组写测试向量的事件顺序。最初一个地址向量被应用并且这之后跟随着一个写测试向量。

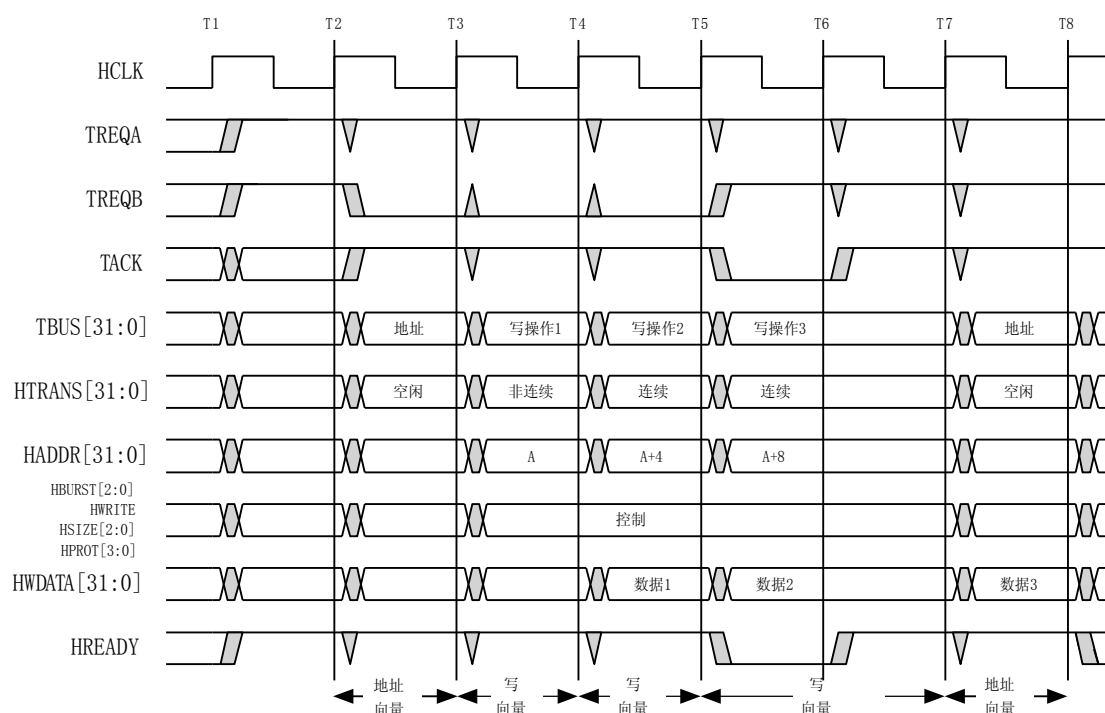


图 6.5 写测试向量

当正在写测试向量时应用下列要点：

- **TREQA**和**TREQB**信号被（读入）通道且被用来表示在接下来的周期将要应用的测试向量类型。图 6.5表示了许多写传输被执行的例子；
- **TIC** 在时刻 **T3** 采样地址和 **TREQA/B** 信号。这之后 **TIC** 可以初始化在 **AHB** 上的合适的传输；
- 在接下来的周期写数据被驱动到 **TBUS** 上且这之后数据在下一个时钟沿 **T4** 被采样，并被驱动到内部总线上；
- 如果内部传输不能完成那么 **TACK** 信号被驱动为低电平而这表示外部测试向量必须在另一个周期被应用；

6.6.3 读传输

读传输更加复杂因为它们要求**TBUS**反向驱动所以要求额外的周期来阻止当在**TBUS**的不同驱动器之间变更时的总线冲突。图 6.6表示了一个典型的读测试序列。

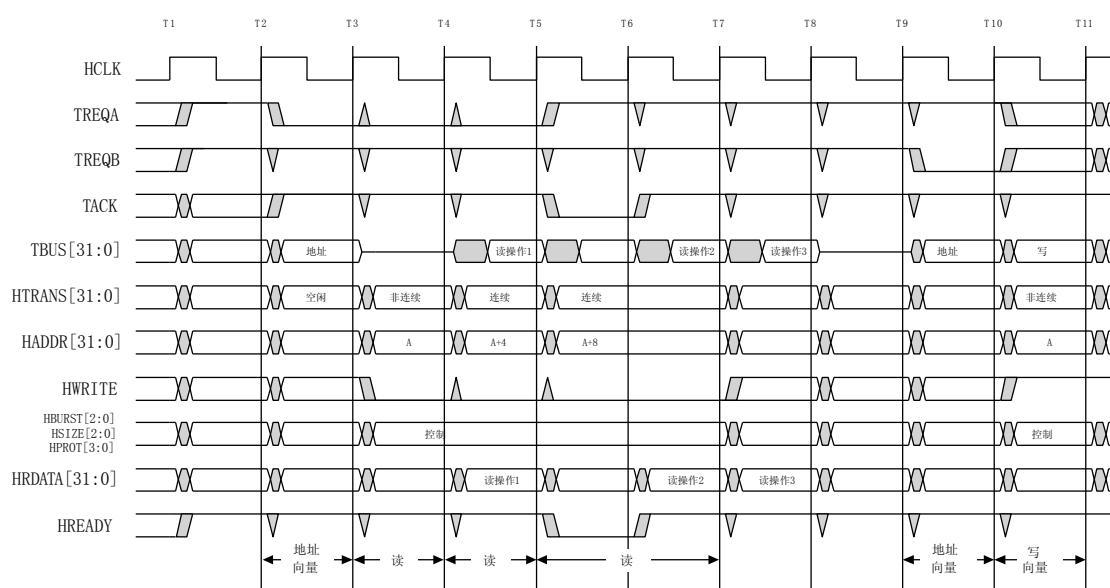


图 6.6 读测试向量

当正在读测试向量时应用下列要点：

- **TREQA** 和 **TREQB** 信号以相同方式被用作一个写传输。最开始，**TREQA/B** 被用来应用一个地址向量，在接下来的周期它们被用来表示要求一个读传输。读的第一个周期 **TBUS** 必须为三态，这确保了驱动 **TBUS** 的外部设备在 **TIC** 将使能片上缓冲以驱动读数据输出之前有一整个周期来使它的缓冲为三态；
- 在一个读突发的末尾也需要留给总线周转时间。这种情况下 **TIC** 必须关闭内部缓冲且在外部测试设备开始驱动前留出一整个周期；
- 读突发的结束由 **TREQA** 和 **TREQB** 都为高电平来表示，作为一个地址向量。实际上它们必须持续两个周期表示地址向量，这分配了在突发开始时的周转周期和突发结束时的周转周期。

6.6.4 控制向量

TIC 的操作可以通过使用一个控制向量而被更改。只要连续应用多于一个地址向量那么最后的向量是被当作一个控制向量并且不被当作地址而锁存。控制向量的位 0 是被用来确定控制向量是否应该被认为有效，这允许对个地址向量被应用而不用更改控制信息。

图 6.7表示了插入一个控制向量的过程。

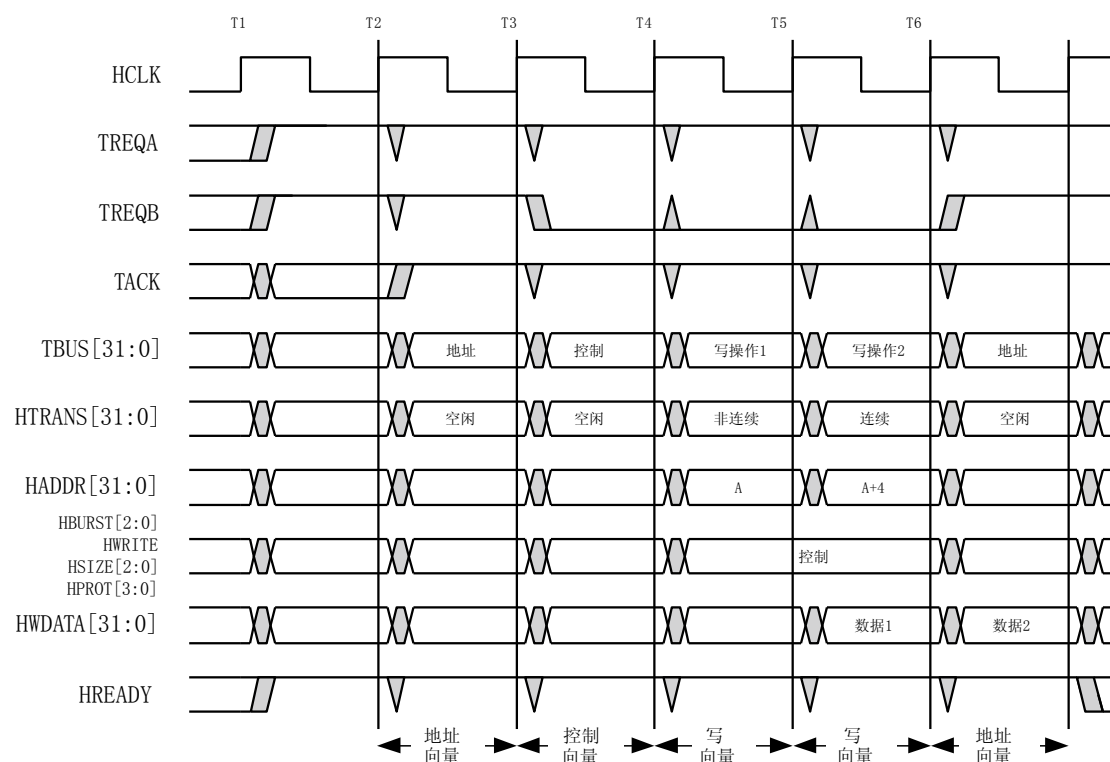


图 6.7 控制向量

在时刻 T4 TIC 可以确定 **TBUS** 包含一个控制向量。这是因为前一个周期是一个地址向量而 **TREQA/B** 都表示接下来的周期是一个读或者一个写因此当前周期必定是一个控制向量。

6.6.5 突发向量

图 6.5和图 6.6中读和写的例子也表示了额外的传输可以如何被使用来形成在总线上的突发传输。TIC限制了突发传输的容量并且只能执行未定义长度的增量突发。

TIC 包含一个 8 位的增量器并且如果作出执行一个增量器边界的突发尝试那么地址将会回环并且 TIC 将给出传输为非连续的信号。可能发生的这种情况下的确切边界取决于传输大小。对字传输增量器将在 1KB 的边界处溢出，对半字传输则是在 512 字节边界处溢出而对字节传输溢出则会在 256 字节边界处发生。

6.6.6 读到写和写到读

在读传输和写传输之间转换而不用应用新的地址向量是可能的。通常这会以地址增量器禁能的方式完成因此读传输和写传输都将是相同的地址。如果测试环境要求以增量器使能来完成这些也是可能的。

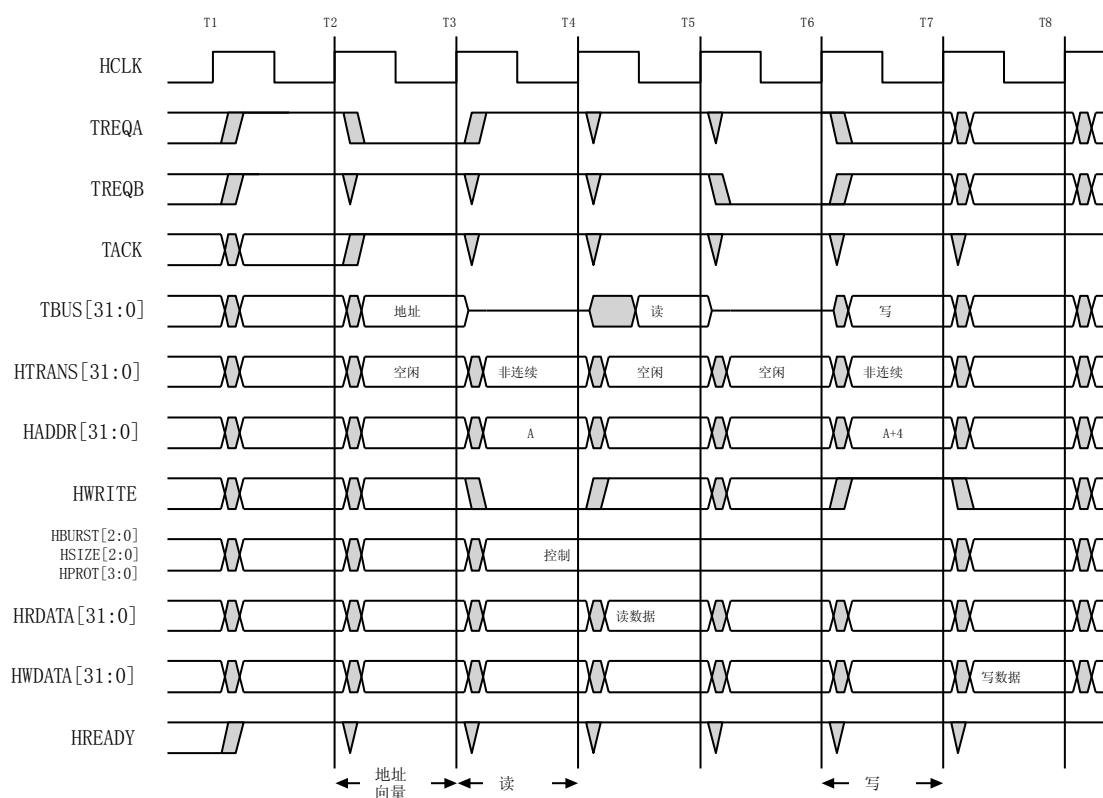


图 6.8 读之后写的传输

当从一个读传输转移到一个写传输时也需要分配两个周期给总线移交因此**TREQA**和**TREQB**应该在读之表示两个周期的地址向量。这不会引起地址改变直到这之后跟着第三个地址向量。图 6.8表示了（这些）事件的顺序。

6.6.7 退出测试模式

使用下列顺序退出测试模式：

- 1、应用单个周期的地址向量，这引起一个内部空闲周期，确保了任何内部传输都已经完成；
- 2、**TREQA** 和 **TREQB** 都被驱动为高电平来表示测试模式将要退出；
- 3、当测试接口被配置为常规系统操作 **TACK** 将变低来表示测试模式已经退出。

测试模式能干净地进入和退出很重要因此 **TIC** 也可以被用来在系统操作期间诊断测试，以及出厂测试。

6.7 ASB测试接口控制器

图 6.9表示了ASB测试接口控制器状态图。

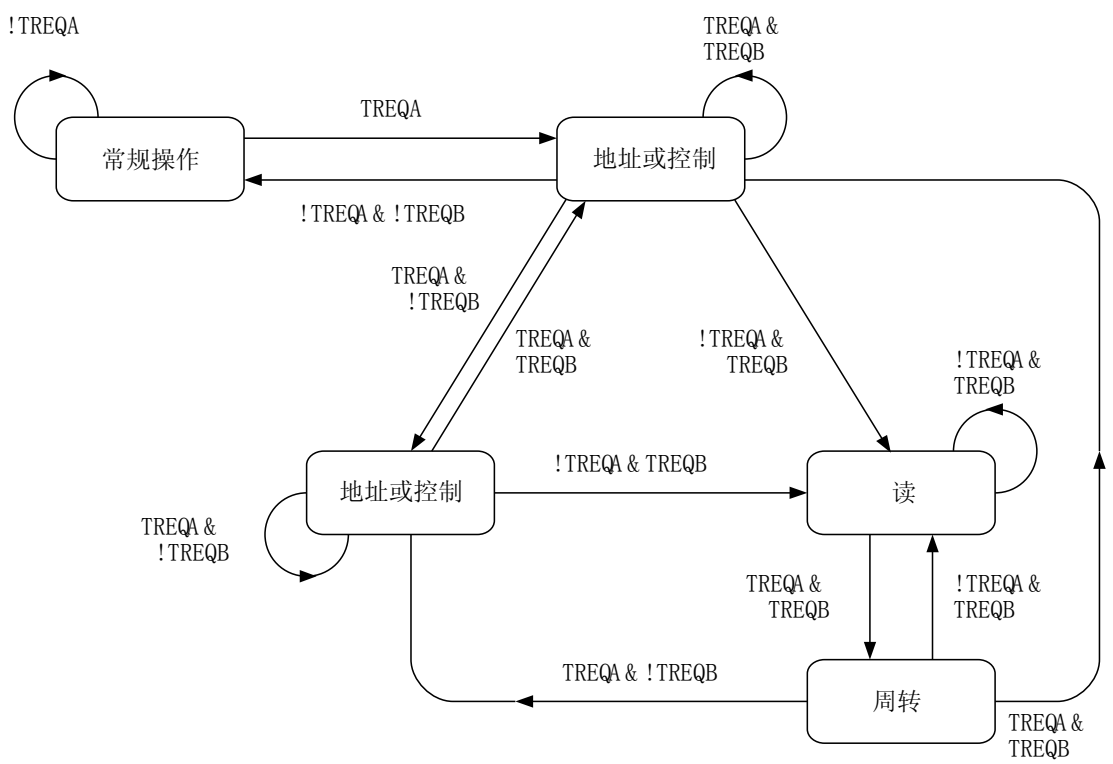


图 6.9 测试接口控制器状态图

当 **TACK** 为高时 **TREQBA** 和 **TREQB** 在 **TCLK** 的下降沿被采样，除了在常规操作状态中 **TREQA** 被异步用来转到地址或控制状态之外。复位状态是常规操作。

6.7.1 控制向量位定义

控制向量被包含在 **TIC** 内来确定 **TIC** 可以执行的传输类型。控制向量被用来设置 **BSIZE**、**BPROT** 和 **BLOK** 的值和控制地址增量。

控制包的位 0 被用来定义将要在内部系统总线上发生的访问。控制包的位 1 是保留作时钟控制和调试。

表 6.4表示了控制向量位分配。

表 6.4 控制向量位定义

位的位置	描述
0	控制向量有效
1	保留
2	BSIZE [0]
3	BSIZE [1]
4	BLOK
5	BPROT [0]
6	BPROT [1]
7	地址增量使能

6.8 AMBA ASB测试序列样例

ASB 测试序列样例用下列标题来描述：

- 进入测试模式；
- 地址向量；
- 控制向量；
- 写测试向量；
- 改变突发方向；
- 退出测试模式。

6.8.1 进入测试模式

测试模式的进入如图 6.10所示，使用下列顺序：

- 1、**TREQA** 被断言以请求测试总线访问；
- 2、当 **TIC** 被授予内部总线时进入测试模式并且这由 **TACK** 信号的断言来表示；
- 3、在这时 **TCLK** 将成为内部 **BCLK** 信号的时钟源；
- 4、当已经进入测试模式时 **TREQB** 被断言来初始化一个地址向量。

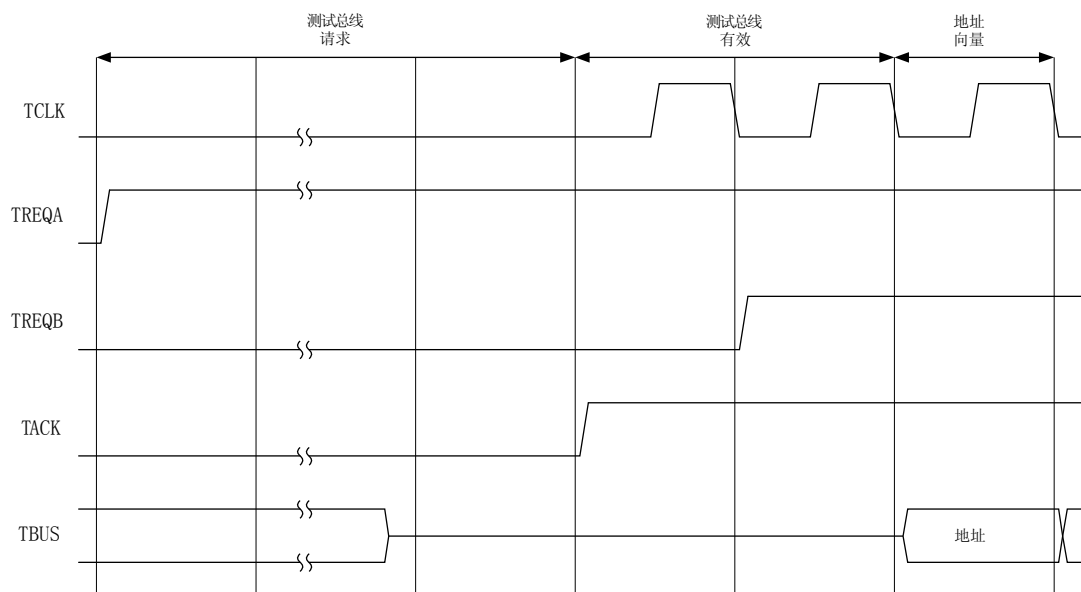


图 6.10 测试开始顺序

6.8.2 地址向量

地址向量必须在一个读或者写操作能发生前被应用。图 6.11表示了一个地址向量之后跟随一个写向量的例子，出现下面的顺序：

- 1、**TREQA** 和 **TREQB** 都被断言为高电平来表示下个周期为一个地址向量；
- 2、在下一个周期地址被应用，而 **TREQA** 和 **TREQB** 改变以表示接下来测试向量的类型。在这个周期期间地址出现在地址总线上；
- 3、在下一个周期写（或者读）向量被应用。

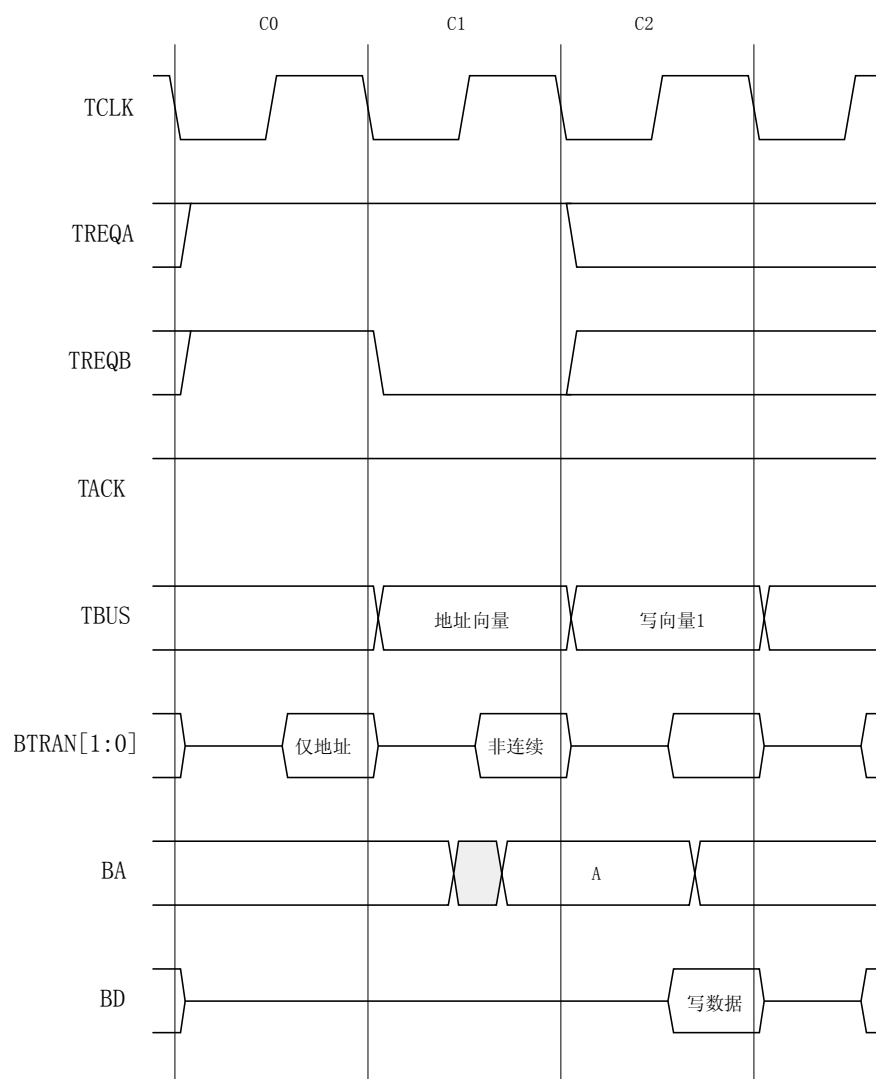


图 6.11 地址向量

6.8.3 控制向量

控制向量必须总是跟在一个地址向量之后。图 6.12表示了一个被写向量跟随着的地址和控制向量的顺序。发生下列顺序：

- 1、 在地址向量结束后 **TREQA** 和 **TREQB** 都保持高电平来表示下一个周期为控制向量；
- 2、 在下一个周期控制信息被应用到 **TBUS[31: 0]**，而 **TREQA** 和 **TREQB** 变更以反映接下来的测试向量类型。在这个周期期间任何被控制向量影响的内部信号，都会改变。

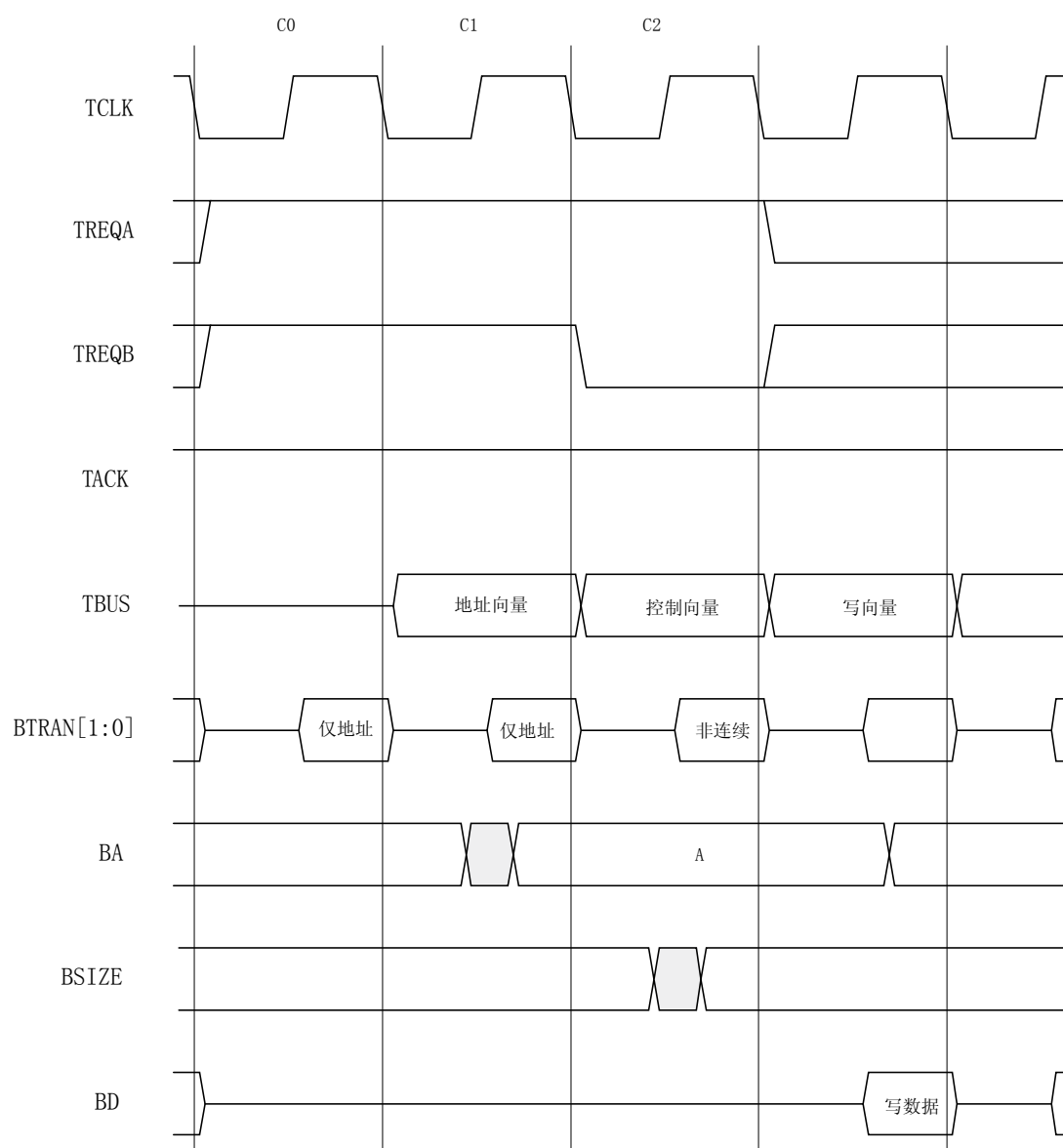


图 6.12 控制向量

图 6.13表示了一个无效控制向量后跟随着一个传输的例子。TIC执行一个在内部总线上的顺序传输因为控制信号没有改变。

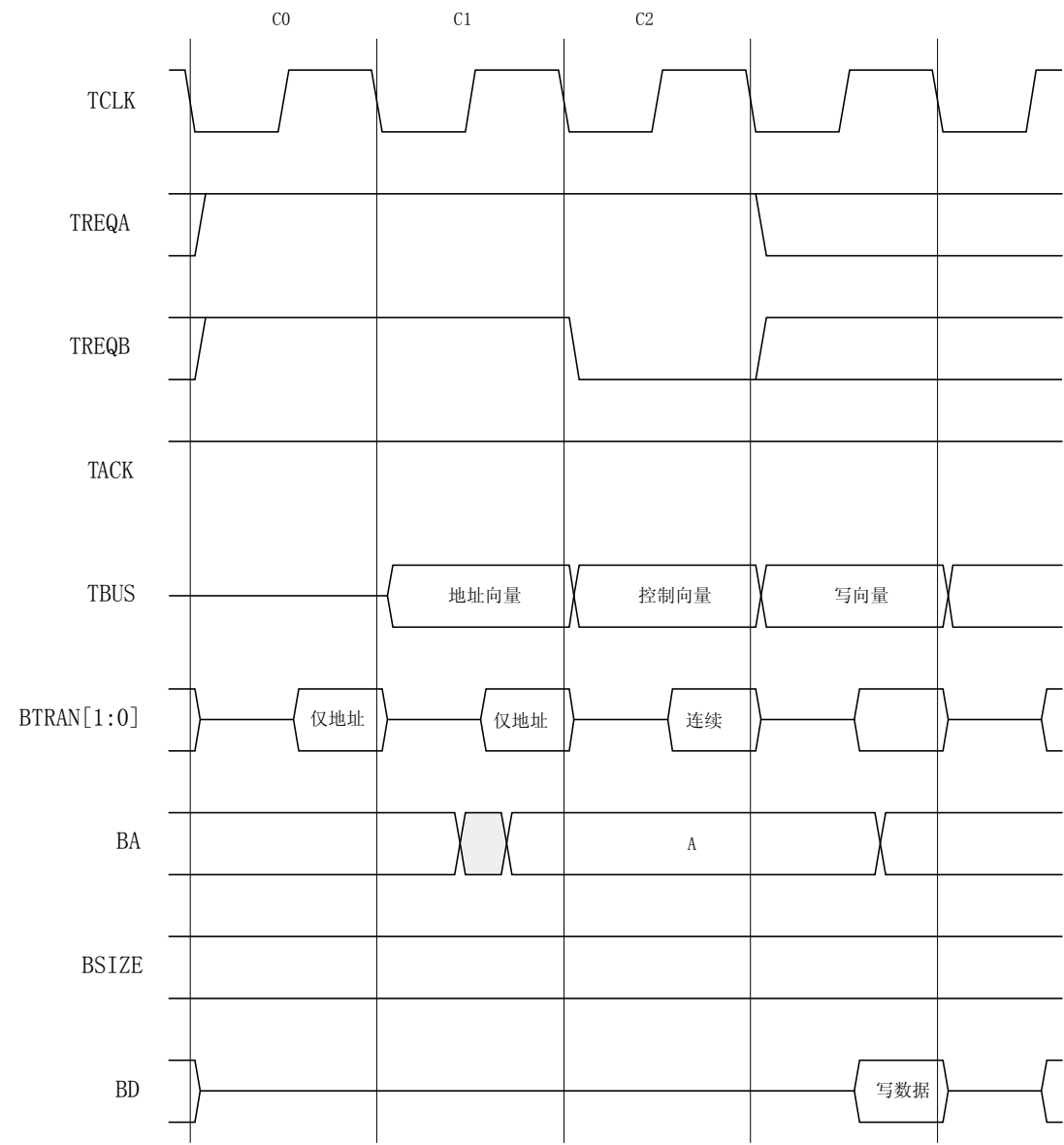


图 6.13 无效控制向量

6.8.4 写测试向量

图 6.14表示了一个单个写向量跟在单个地址向量之后的例子。

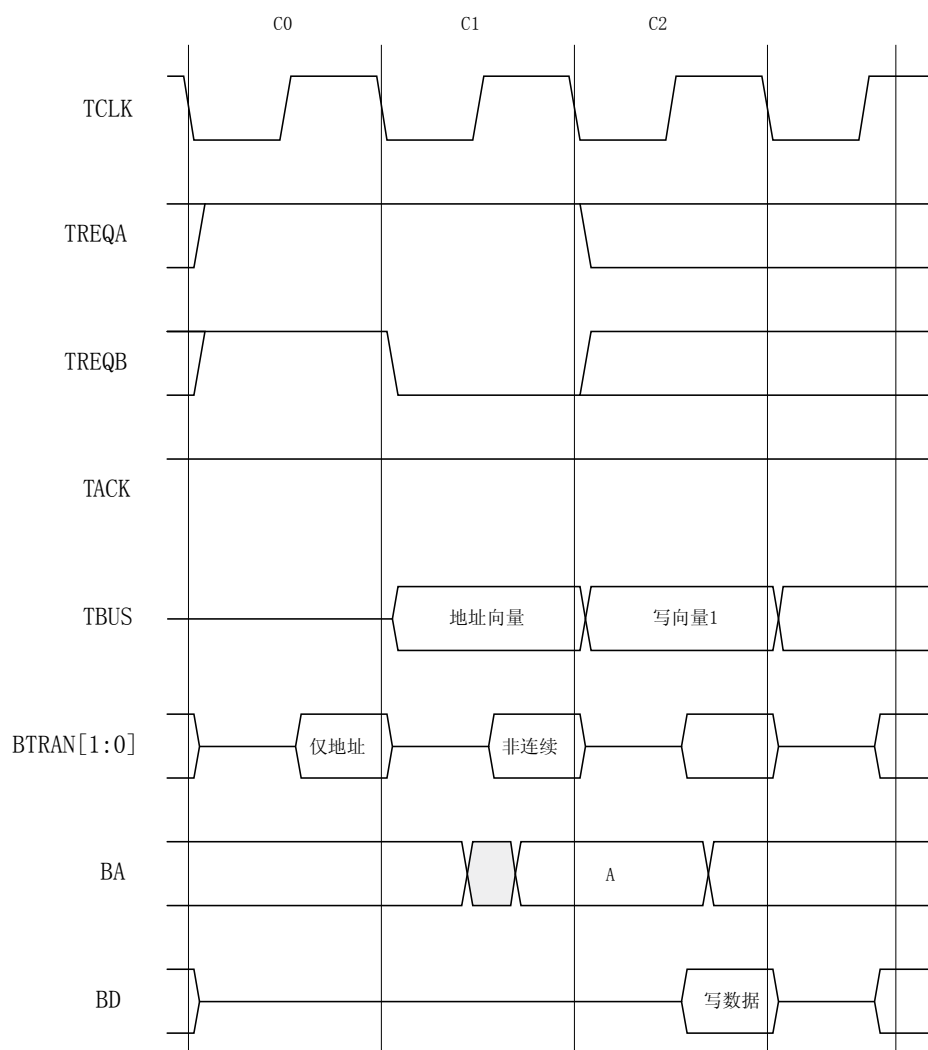


图 6.14 写测试向量

图 6.15表示了一个单个地址向量之后跟着扩展的写向量的例子。

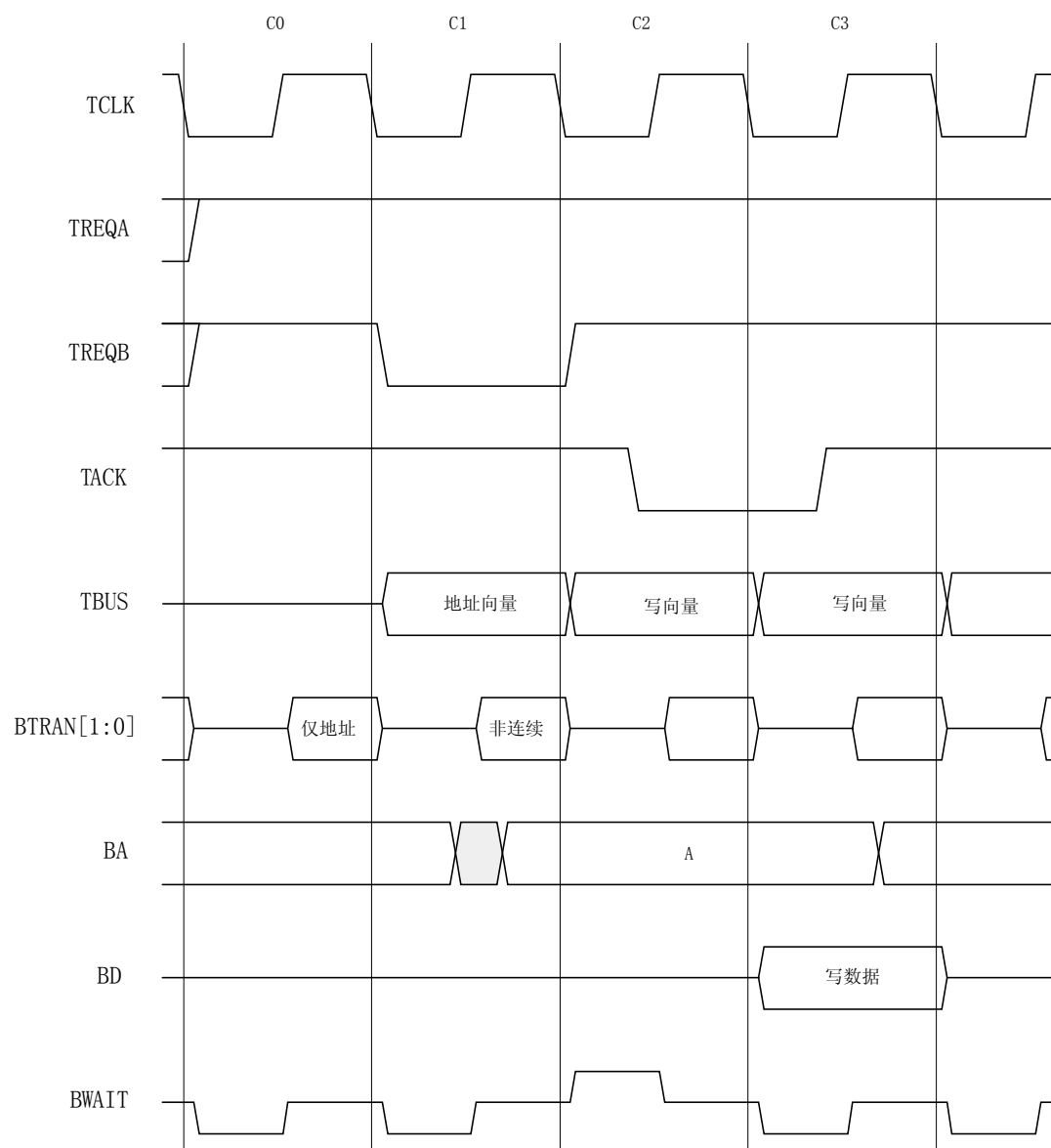


图 6.15 扩展的写测试向量

图 6.16表示了一个单地址向量的例子，之后跟着单个读向量并且（测试）以单个周转向量来终止。

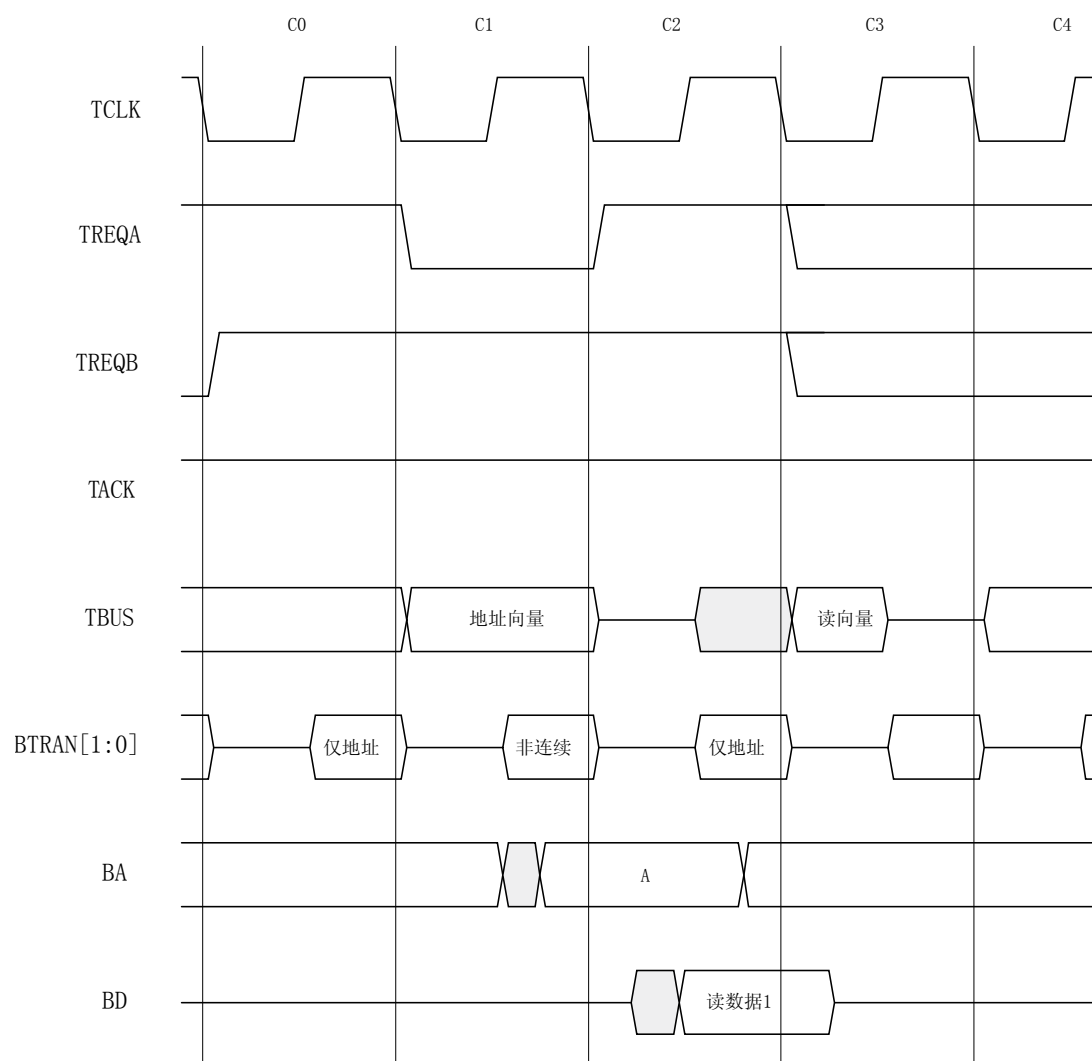


图 6.16 读测试向量

图 6.17表示了到非增量地址的连续传输。

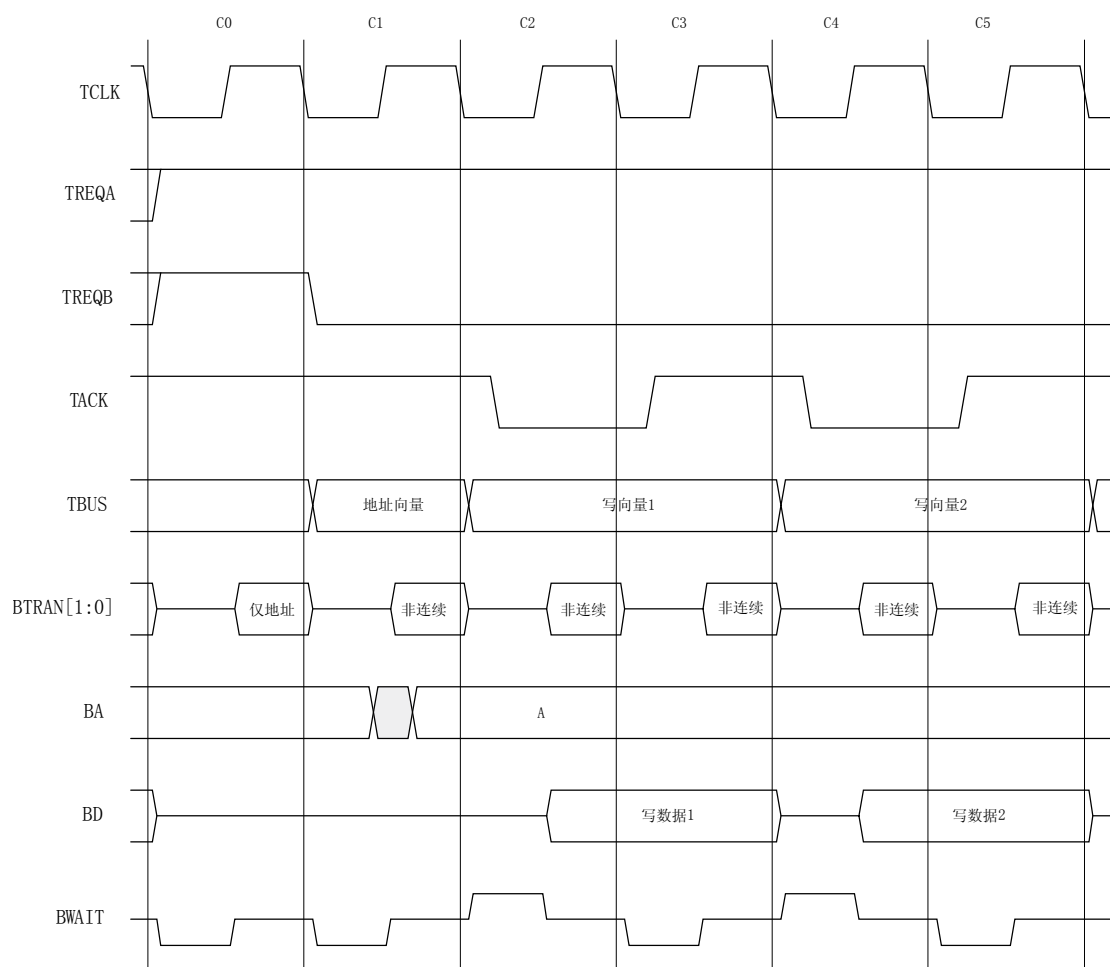


图 6.17 增量禁能的突发写向量

图 6.18表示了到增量地址的连续传输。

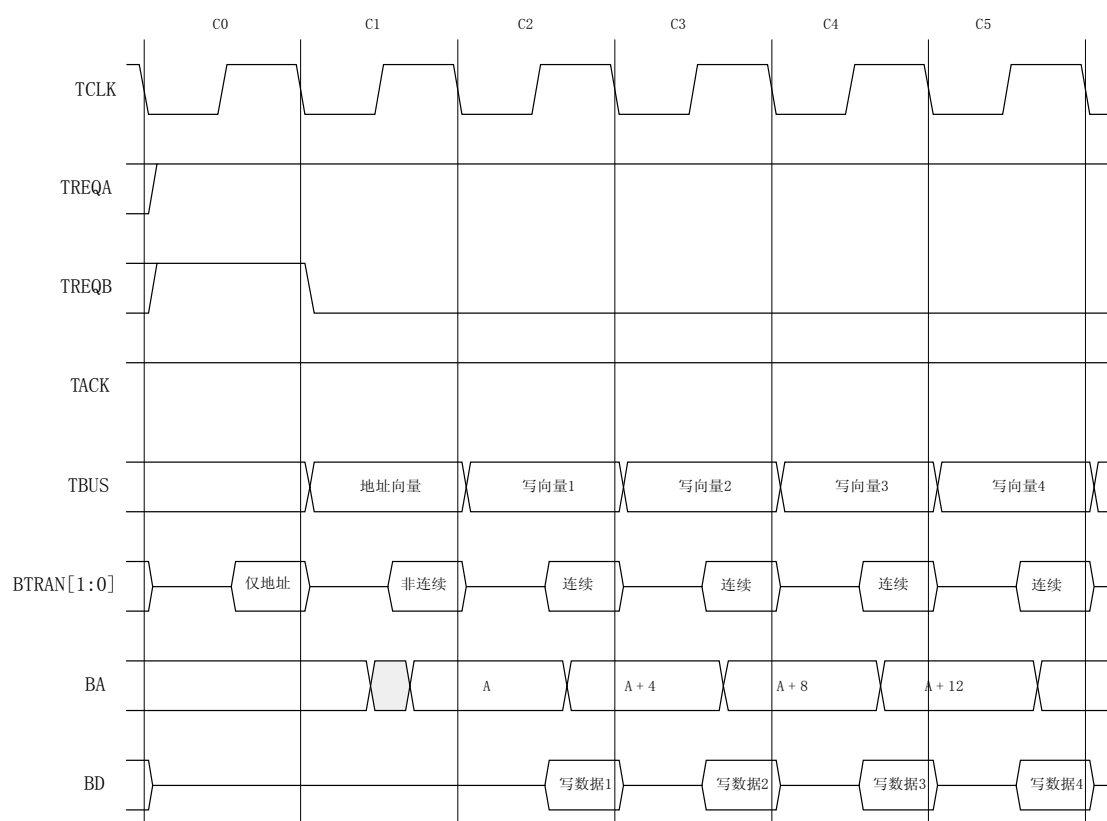


图 6.18 增量使能的突发写向量

6.8.5 改变突发方向

下面的图 6.19表示了一次从读到写方向改变的突发。

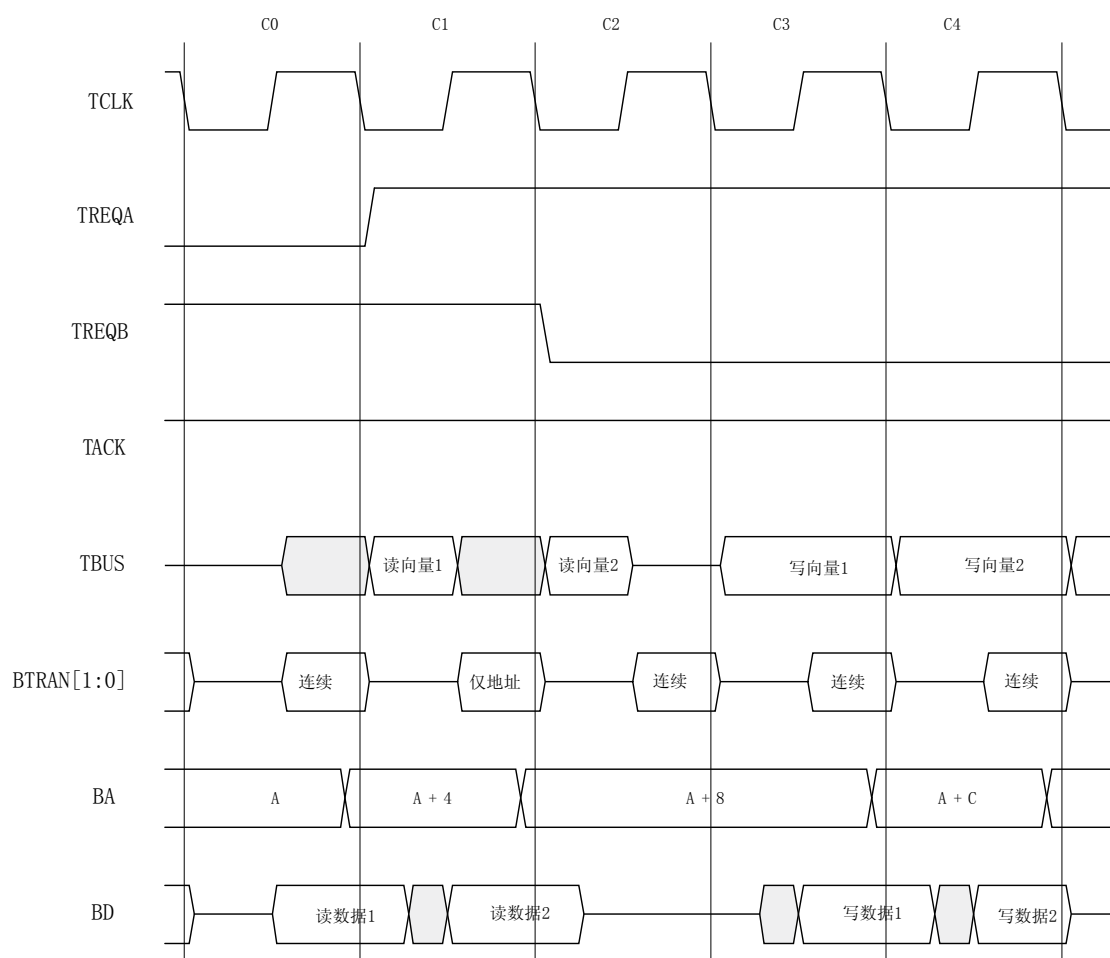


图 6.19 改变突发方向

6.8.6 退出测试模式

图 6.20表示了从测试模式退出的一个例子。

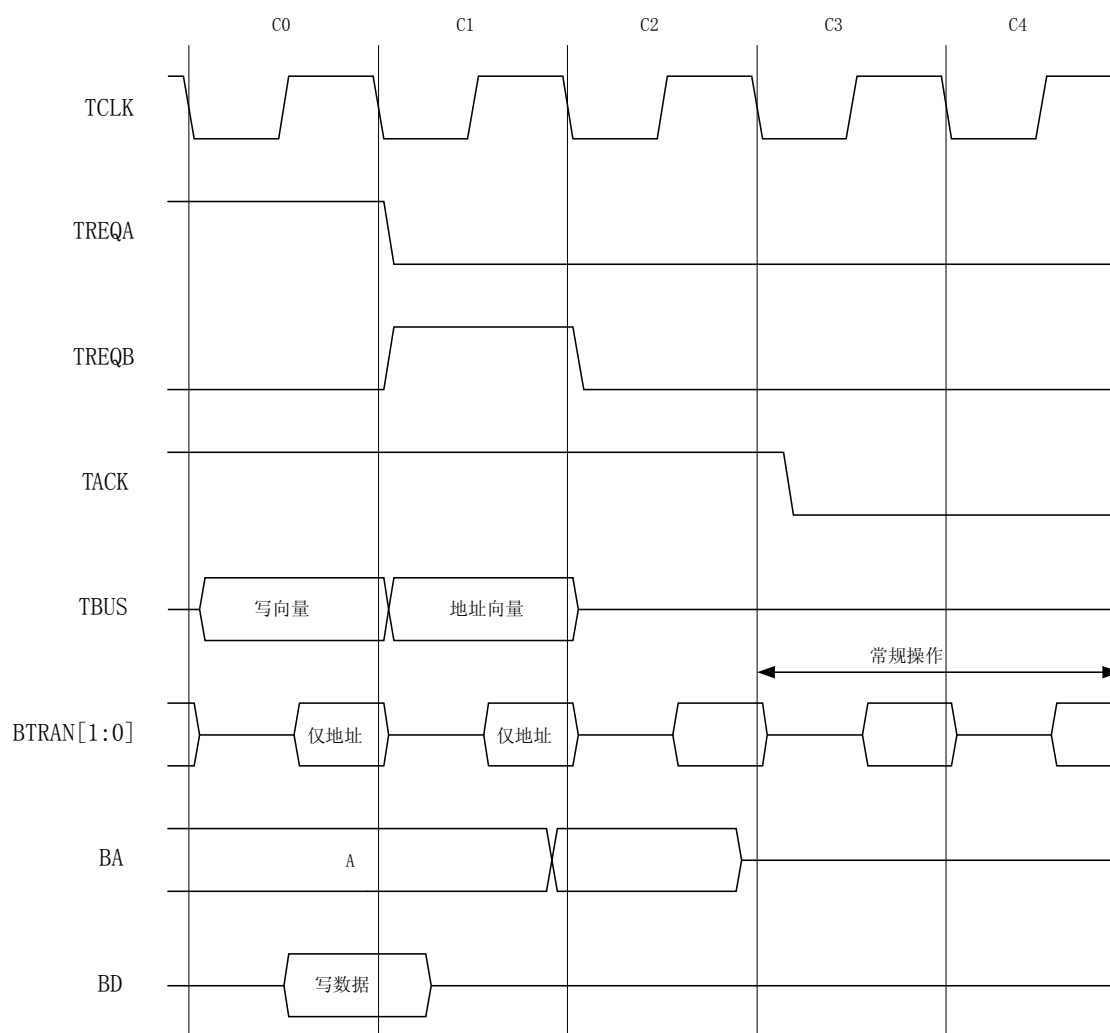


图 6.20 退出测试模式

附录A 索引

这个索引中的条目以字母表的顺序列出。

A

Active state

Address and control signals

ASB

timing

Address bus

AHB

APB

ASB

Address decoding

AHB

ASB

Address vectors

Address-only transfers

AGNTx

AHB

arbiter

arbitration signals

decoder

introduction to

master

operation

signal list

signal prefixes

slave

AHB/ASB or APB, when to use

AMBA signal names

AMBA system, typical

AMBA test interface

AMBA test methodology

APB

address bus

bridge

bridge interface diagram

bridge transfer

components

in a typical AMBA system

introduction to

read data bus

read transfers

select

signal list

signal prefixes

slave

slave interface diagram

strobe

timing parameters

transfer direction

write data bus

write transfers

Arbiter

AHB

ASB

Arbitration and reset signals

Arbitration signals

AHB

ASB

Arbitration, AHB

AREQx

ASB

and APB

arbiter

arbiter interface diagram

arbiter timing parameters

bus master

bus master interface diagram

bus slave

bus slave interface	Bursts
components	incrementing
decoder	of read transfers
decoder interface diagram	of write transfers
decoder timing diagrams	undefined length
decoder timing parameters	wrapping
description	Bus
introduction to	backbone
master	choosing
signal description	peripheral
signal list	Bus clock
signal prefixes	AHB
slave	APB
slave bus interface state machine	ASB
test sequence	Bus cycle
transfers	Bus grant
B	AHB
BA	ASB
Back to back transfers	Bus interface state machine, ASB
Backbone bus	Bus lock
Basic transfers	Bus master
BCLK	ASB
BD	default
BERROR	granted state machine
BLAST	handover
BLOK	interface, ASB
BnRES	main state machine
BPROT	timing diagrams, ASB
encoding	timing parameters, ASB
BSIZE	Bus request
encoding	AHB
BTRAN	ASB
encoding	Bus retract
timing	Bus slave interface, ASB
Burst operation	Bus transfer
Burst type, AHB	Busidle state
Burst vectors	BWAIT

BWRITE	External test interface
encoding	
C	G
Choosing the right bus	Grant signal, AHB
Clock ASB	Granted state machine
Control signals	
Control vectors	H
bit definitions	HADDR
D	Handover
Data bus	Handover state
AHB	Handover, bus master
ASB	HBURST
Deadlock	HBUSREQx
Decode cycles	HCLK
Decoder	HGRANTx
AHB	HLOCKx
ASB	HMASTER
state machine	HMASTLOCK
with decode cycles	Hold state
without decode cycles	HPROT
Default bus master	HRDATA
Direction of transfer	HREADY
APB	HRESETn
ASB	HRESP
Done response	HSELx
DSEL	HSIZE
DSELx	HSPLITx
E	HTRANS
Early burst termination	HWDATA
Electrical characteristics	HWRITE
Enable state	I
Enter test mode	Idle state
Error response	Incremental addressing
ASB	Incrementing burst
Exit from reset	Interfacing
Exit test mode	APB to AHB
	APB to ASB
	revD peripherals

L	Read data bus
Last response	AHB
ASB	APB
Lock signal, AHB	Read test vectors
Locked sequence, AHB	Read transfers
Locked transfers	APB
AHB	burst of
ASB	to ASB
M	Reset
Master	AHB
AHB	APB
ASB	ASB
Master number	exit from
AHB	Reset operation, ASB
Multi master operation, ASB	Response encoding
Multiple transfers	Retract response
N	Retract state
Nonsequential transfers	Retry transfers
P	Rev D peripherals
PADDR	S
PCLK	Select, APB
PENABLE	Sequential transfers
Peripheral bus	Setup state
Peripheral test harness	Signal list
PRDATA	AHB
PRESETn	APB
Protection control	ASB
AHB	Signal names
ASB	AMBA
Protection signals	Signal prefixes
ASB	AHB
PSELx	APB
PWDATA	ASB
PWRITE	Size encoding
R	Size of transfer
	ASB
	Slave

AHB	entering
ASB	exiting
transfer response	Test sequence
Slave select	ASB
AHB	Test vector types
ASB	TIC
Split completion request, AHB	Timing diagrams
Split transfers	APB bridge
State diagram	APB slave
TIC	ASB arbiter
State machine	ASB bus slave
ASB slave bus interface	ASB decoder
bus interface, ASB	Timing parameters
bus master, main	APB
decoder	APB slave
Strobe, APB	ASB
T	ASB arbiter
TACK	ASB bus master
TBUS	ASB bus slave
TCLK	Timing specification
Technology independence	Transfer direction
Termination, early burst	AHB
Terminology	APB
Test	ASB
transfer parameters	Transfer direction, ASB
Test acknowledge	Transfer done
Test bus	AHB
Test bus request	Transfer response
Test clock	AHB
Test harness	ASB
Test Interface Controller	combinations
ASB	timing
ASB state diagram	Transfer size
Test Interface Controller (TIC)	AHB
Test Interface Controller state	ASB
diagram	Transfer type
Test mode	AHB

ASB	Typical AMBA system
encoding	AHB based
Transfers	ASB based
address only	U
back to back	Undefined length burst
basic	W
multiple	Wait response
nonsequential	ASB
sequential	Wait states
split	Wrapping burst
with retry response	Write data bus
with wait states	AHB
TREQA	APB
TREQB	Write test vectors
Tristate	Write transfers
data bus	APB
enable of address and control signals	burst of
Two cycle response	from ASB
Type of transfer	
ASB	

附录B 后记

缘起

翻译这篇文档，是一时兴起。当时在 21IC 上面荡到了这篇文档，想到自己目前也在从事嵌入式产品开发的工作，在使用 NXP 公司的 LPC2000 系列 ARM 芯片时，对其中的 DMA 部分的一些概念很费解，因为其中多数是和 AMBA 总线传输相关的。再加上平时也经常看一些英文的数据手册，里面的一些专业术语看着很是头疼，同时也切身实地的感受到英语的重要性。自己的英语水平有点差劲，平时也更倾向于看中文资料。所以在下载了英文版的 AMBA 总线规范后，就想要是把这篇文档翻译出来，或许能对从事嵌入式的同行们有一些帮助，让大家更好的理解 ARM 结构和 AMBA 体系，同时，也能提高一下自己的英语水平。于是，想法变成了行动。

过程

这篇文档的翻译，大约持续了一个半月的时间。因为白天要上班，所以只有晚上回宿舍后才开始翻译。每天付出的工作量也不大，8 点多回到宿舍，休闲一会后，十点左右开始翻译，翻译到 12 点左右就停工了，因为第二天还得上班，不敢开夜车开的太晚。这样下来，每晚能够翻译七八页左右的英文原文。不过中间由于一些事情耽搁了一下，所以完成翻译的时候，已经快到中秋节了。本来翻译完了还想好好的检查一遍，可是后面项目比较紧张，所以只好匆匆的浏了一遍，在排版和一些明显错误的地方做了修改。

翻译的过程并不是很顺利。以前在学英语的时候，比较烦的就是做阅读理解了，要花很多的时间。所以这次看着这 230 页的英文 PDF 文档，翻译过程中有好几次都想放弃。不过转念一想，既然选择了做这件事情，不管结果是好还是坏，一定要坚持下去，把这件事情做完，半途而废是最可耻的失败。好在最后坚持下来了，或许有些事情就是这样，就是在你最想放弃的时候，再坚持一会就撑过去了。

英文的翻译，并不是想象中的那么简单。一个是那些难懂的术语，看着上下文，能够理解它们的意思，但是要找一个合适的词汇来准确的描述这些术语，实在是有困难。另外就是文档中许多很长的从句，修饰的对象一个叠一个，要调整好语序，使之前后通顺，也不大容易。而且有些地方翻译出来在读的时候，也要注意断句，否则会很难看懂。还有一个问题就是对原文的忠实程度，这个不好把握，因为如果全部按照字面意思来翻译，即使语序调整好了，阅读起来仍然很拗口，所以有些地方还是要换成易懂的表述方式，不过这有一个缺点就是可能会丢失英文中生动而形象的表达效果。所以在做这个权衡时，我还是倾向于尽量忠实于原文。

结果

整个过程走下来，感觉还是收获了很多东西。不仅仅是对更多专业术语的理解、英文阅读能力的提高，还对 WORD 的排版、文档编写的规范有了更深的认识，老外的一篇简单技术规范就能写这么多东西，而且整个文档相当规范，人家细致而专注的精神是值得我们学习的。当然，最重要的收获还是对于“世上无难事，只怕有心人”这句话的理解。

其实在这之前，自己也翻译过一些文档，比如在本科期间的专业英语以及毕设时的外文翻译，加上这次，让我对老外的行文笔法和表达效果产生了兴趣，以后有空，还会再翻译一些外文资料的。

附录C 说明

由于译者水平有限，加上时间匆忙，本文档中难免有错误和不足之处，欢迎感兴趣的同志们批评和指正。请把您的意见和建议发送到kongsuozt@126.com，我也会及时更新大家的意见和建议，争取把这篇文档做得更好。

本文档仅供从事相关行业的人员作学习和交流之用，不得用于出版、发行或者其他商业目的。