SoC 总线互联

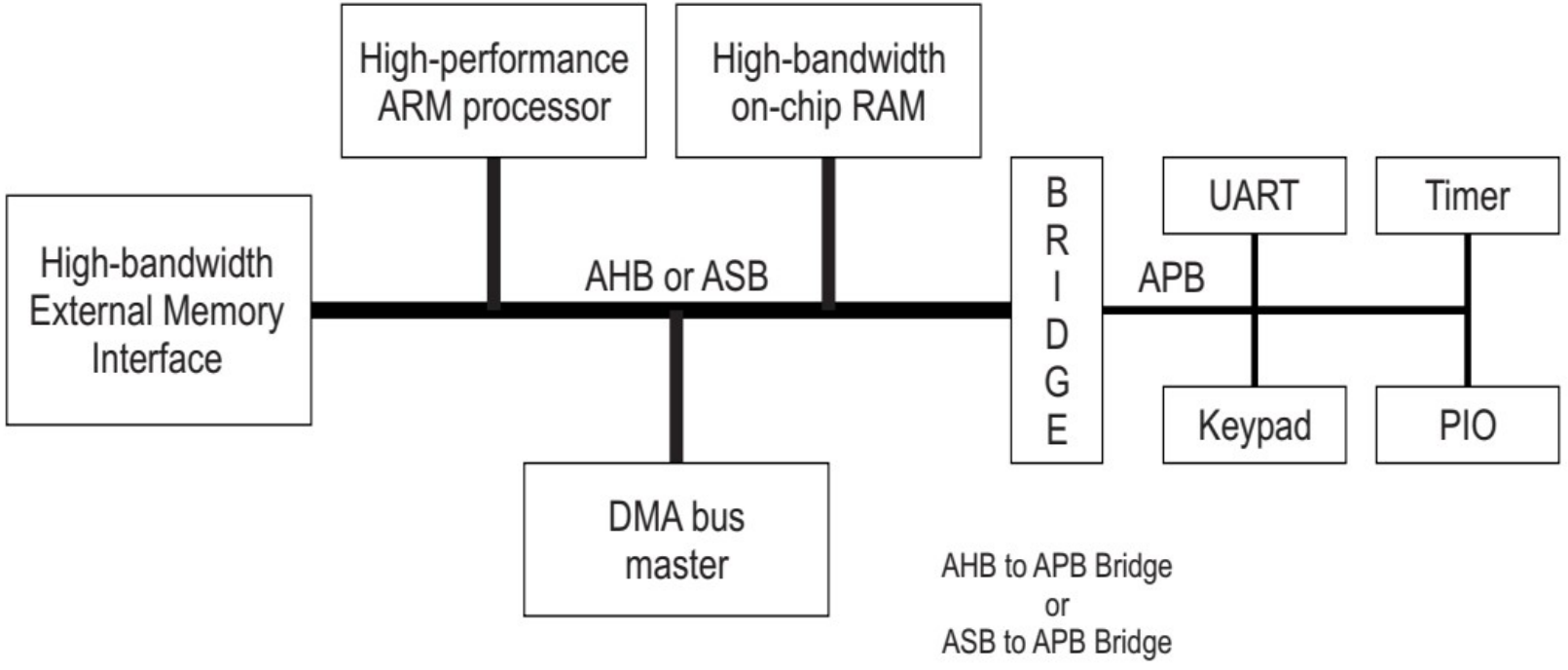
AMBA AHB

AMBA 总线简介

AMBA(Advanced Microcontroller Bus Architecture) 2.0 总线协议定义了三种总线

- 高级高性能总线（ AHB: Advanced High-performance Bus ）
- 高级系统总线（ ASB : Advanced System Bus ）
- 高级外设总线（ APB : Advanced Peripheral Bus ）

基于 AMBA AHB 的微控制器总线架构



AMBA AHB

* High performance
* Pipelined operation
* Multiple bus masters
* Burst transfers
* Split transactions

AMBA ASB

* High performance
* Pipelined operation
* Multiple bus masters

AMBA APB

* Low power
* Latched address and control
* Simple interface
* Suitable for many peripherals

术语

- 总线周期：总线时间的基本单元。对于 AHB 或者 APB 协议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： 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。 ASB 总线支持的传输大小包括字节（ 8 位）、半字（ 16 位）、字（ 32 位）。 AHB 又支持较宽的数据传输，包括 64 位和 128 位的数据传输。 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。

AMBA AHB 介绍

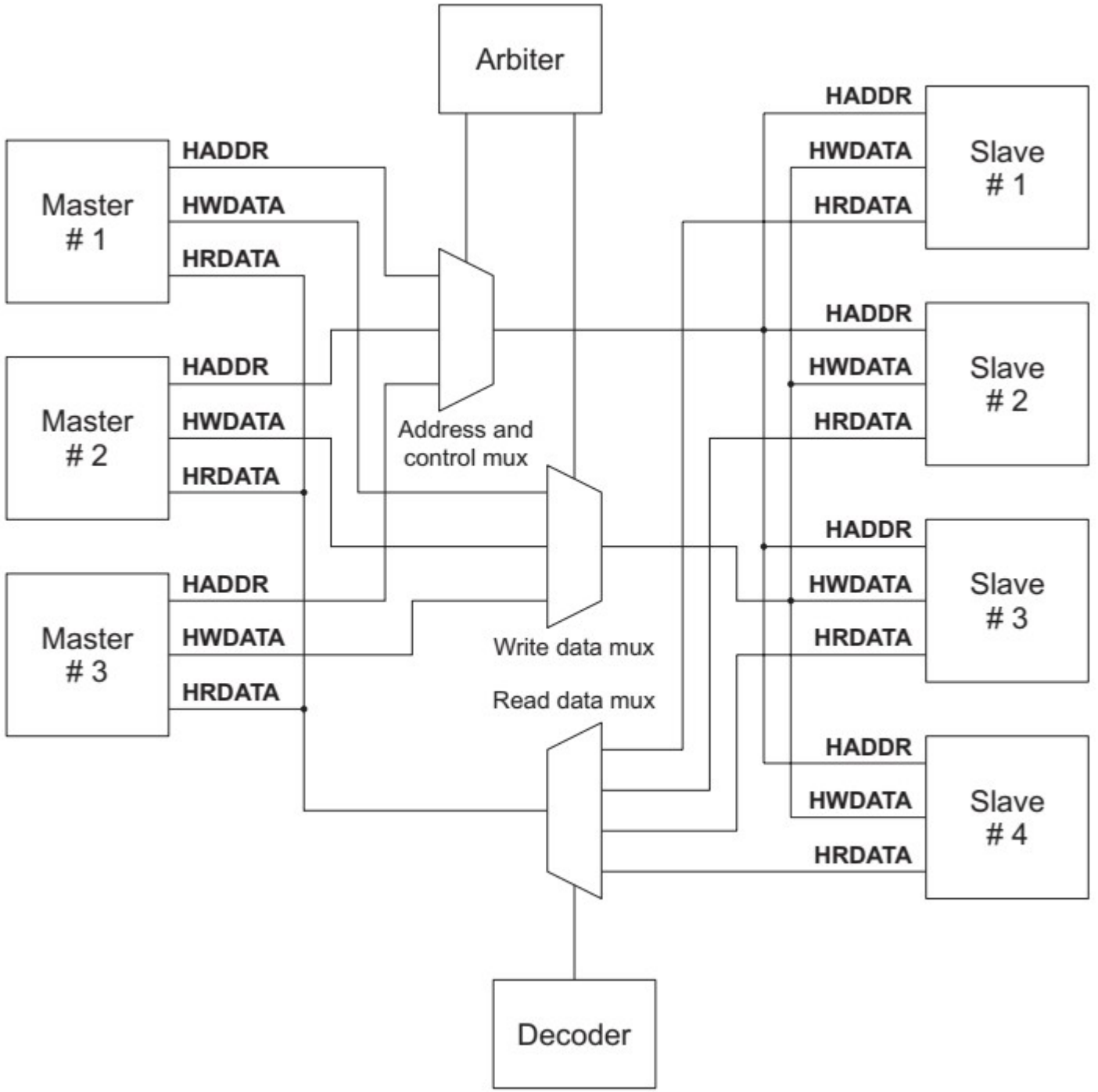
AHB 是为提出高性能可综合设计的要求而产生的新一代 AMBA 总线，支持多总线主机和提供高带宽操作（AHB: Advanced High-performance Bus），具有以下特点：

- 高性能、高速总线
- 突发传输（ burst transfers ）
- 分离处理（ split transactions ）
- 单周期总线主机移交（ single cycle bus master handover ）
- 单一时钟沿操作（ single clock edge operation ）
- 非三态实现（ non-tristate implementation ）
- 更宽的数据总线架构（ 8/16/32/64/128 位）

AHB 系统总线组成

- AHB Master
  - ✓发起 AHB 总线读写操作
  - ✓某一时刻只允许一个 Master 占用总线
- AHB Slave
  - ✓响应 AHB 总线读写操作
  - ✓通过地址映射选择使用 Slave
- AHB Arbiter
  - ✓允许某一个 Master 控制 AHB 总线
  - ✓具体的仲裁算法按需选取（ AMBA 协议中没有定义）
- AHB Decoder
  - ✓通过地址译码决定选择使用 Slave

AHB总线互联


Arbiter
Master # 1
Master # 2
Master # 3
HADDR
HWDATA
HRDATA
Address and control mux
Write data mux
Read data mux
Slave # 1
Slave # 2
Slave # 3
Slave # 4
Decoder

3.5 Private Peripheral Bus (PPB) memory map

The PPB is part of the defined Arm bus architecture and provides access to select processor-local modules. These resources are only accessible from the core; other system masters do not have access to them.

Table 3-3. PPB memory map for CM4

System 32-bit address range	Resource
0xE000_0000–0xE000_0FFF	Instrumentation Trace Macrocell (ITM)
0xE000_1000–0xE000_1FFF	Data Watchpoint and Trace (DWT)
0xE000_2000–0xE000_2FFF	Flash Patch and Breakpoint (FPB)
0xE000_3000–0xE000_DFFF	Reserved
0xE000_E000–0xE000_EFFF	System Control Space (SCS) (for NVIC and FPU
0xE000_F000–0xE003_FFFF	Reserved
0xE004_0000–0xE004_0FFF	Trace Port Interface Unit (TPIU)
0xE004_1000–0xE004_1FFF	Reserved
0xE004_2000–0xE004_2FFF	Reserved
0xE004_3000–0xE004_3FFF	Reserved
0xE004_4000–0xE007_FFFF	Reserved
0xE008_0000–0xE008_0FFF	Miscellaneous Control Module (MCM)
0xE008_1000–0xE008_1FFF	Reserved
0xE008_2000–0xE008_2FFF	Cache Controller (LMEM)
0xE008_3000–0xE00F_EFFF	Reserved
0xE00F_F000–0xE00F_FFFF	Arm Core ROM Table[1] - allows auto-detection of debug components

AHB 信号

Name	Source	Description
HCLK Bus clock	Clock source	This clock times all bus transfers. All signal timings are related to the rising edge of HCLK.
HRESETn Reset	Reset controller	The bus reset signal is active LOW and is used to reset the system and the bus. This is the only active LOW signal.
HADDR[31:0] Address bus	Master	The 32-bit system address bus.
HTRANS[1:0] Transfer type	Master	Indicates the type of the current transfer, which can be NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY.
HWRITE Transfer direction	Master	When HIGH this signal indicates a write transfer and when LOW a read transfer.
HSIZE[2:0] Transfer size	Master	Indicates the size of the transfer, which is typically byte (8-bit), halfword (16-bit) or word (32-bit). The protocol allows for larger transfer sizes up to a maximum of 1024 bits.
HBURST[2:0] Burst type	Master	Indicates if the transfer forms part of a burst. Four, eight and sixteen beat bursts are supported and the burst may be either incrementing or wrapping.
HPROT[3:0] Protection control	Master	The protection control signals provide additional information about a bus access and are primarily intended for use by any module that wishes to implement some level of protection. The signals indicate if the transfer is an opcode fetch or data access, as well as if the transfer is a privileged mode access or user mode access. For bus masters with a memory management unit these signals also indicate whether the current access is cacheable or bufferable.
HWDATA[31:0] Write data bus	Master	The write data bus is used to transfer data from the master to the bus slaves during write operations. A minimum data bus width of 32 bits is recommended. However, this may easily be extended to allow for higher bandwidth operation.

AHB 信号

Name	Source	Description
HSELx Slave select	Decoder	Each AHB slave has its own slave select signal and this signal indicates that the current transfer is intended for the selected slave. This signal is simply a combinatorial decode of the address bus.
HRDATA[31:0] Read data bus	Slave	The read data bus is used to transfer data from bus slaves to the bus master during read operations. A minimum data bus width of 32 bits is recommended. However, this may easily be extended to allow for higher bandwidth operation.
HREADY Transfer done	Slave	When HIGH the **HREADY** signal indicates that a transfer has finished on the bus. This signal may be driven LOW to extend a transfer. Note: Slaves on the bus require **HREADY** as both an input and an output signal.
HRESP[1:0] Transfer response	Slave	The transfer response provides additional information on the status of a transfer. Four different responses are provided, OKAY, ERROR, RETRY and SPLIT.

AMBA AHB also has a number of signals required to support multiple bus master operation (see Table 2-2). Many of these arbitration signals are dedicated point to point links and in Table 2-2 the suffix **x** indicates the signal is from module X. For example there will be a number of **HBUSREQx** signals in a system, such as **HBUSREQarm**, **HBUSREQdma** and **HBUSREQtic**.

Table 2-2 Arbitration signals

Name	Source	Description
HBUSREQx Bus request	Master	A signal from bus master x to the bus arbiter which indicates that the bus master requires the bus. There is an **HBUSREQx** signal for each bus master in the system, up to a maximum of 16 bus masters.
HLOCKx Locked transfers	Master	When HIGH this signal indicates that the master requires locked access to the bus and no other master should be granted the bus until this signal is LOW.
HGRANTx Bus grant	Arbiter	This signal indicates that bus master x is currently the highest priority master. Ownership of the address/control signals changes at the end of a transfer when **HREADY** is HIGH, so a master gets access to the bus when both **HREADY** and **HGRANTx** are HIGH.

AHB 信号

Name	Source	Description
HMASTER[3:0] Master number	Arbiter	These signals from the arbiter indicate which bus master is currently performing a transfer and is used by the slaves which support SPLIT transfers to determine which master is attempting an access. The timing of **HMASTER** is aligned with the timing of the address and control signals.
HMASTLOCK Locked sequence	Arbiter	Indicates that the current master is performing a locked sequence of transfers. This signal has the same timing as the **HMASTER** signal.
HSPLITx[15:0] Split completion request	Slave (SPLIT-capable)	This 16-bit split bus is used by a slave to indicate to the arbiter which bus masters should be allowed to re-attempt a split transaction. Each bit of this split bus corresponds to a single bus master.

AHB 信号

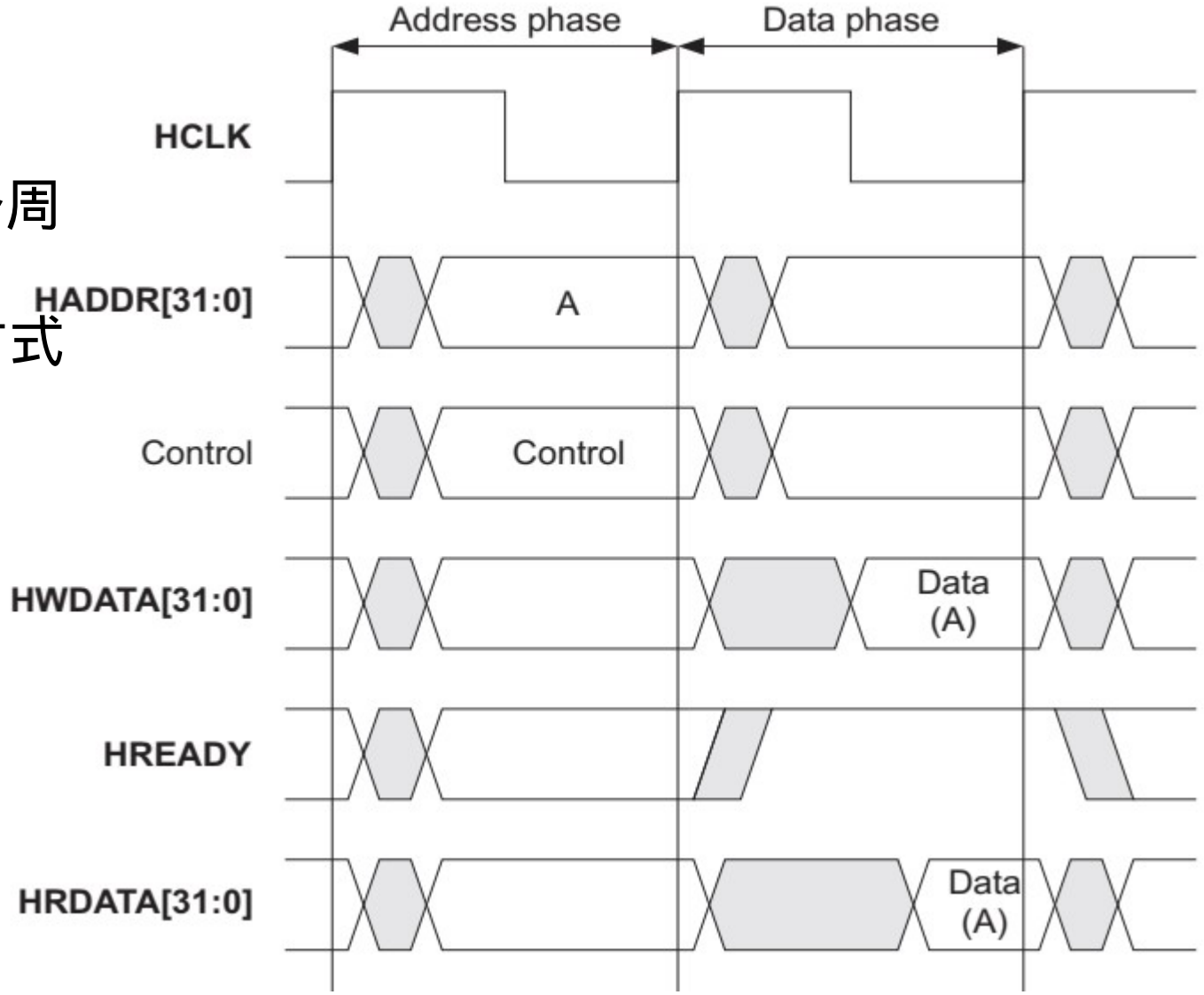
Name	Source	Description
HADDR[31:0]	Master	Address bus
HTRANS[1:0]	Master	Transfer type
HWRITE	Master	Transfer direction
HSIZE[2:0]	Master	Transfer size
HBURST[2:0]	Master	Burst type
HPROT[3:0]	Master	Protection control
HWDATA[31:0]	Master	Write data bus
HSELx	*Decoder*	Slave select
HRDATA[31:0]	Slave	Read data bus
HREADY	Slave	Transfer done
HRESP[1:0]	Slave	Transfer response
HBUSREQx	Master	Bus request
HLOCKx	Master	Locked transfers
HGRANTx	*Arbiter*	Bus grant
HMASTER[3:0]	*Arbiter*	Master number
HMASTLOCK	*Arbiter*	Locked sequence
HSPLITx[15:0]	Slave	Split completion request

AHB 总线操作概述

- AHB Master 发起总线访问请求
- AHB Arbiter 授权 AHB Master 使用总线
- AHB Decoder 根据 AHB 地址信号，选择对应的 AHB Slave
- AHB Slave 响应 AHB Master 的总线访问操作
  - **HRESP** ： **OKAY**、 **ERROR**、 **RETRY**、 **SPLIT**
    RETRY 和 SPLIT 两个传输响应都表示传输不能立刻完成，但是 Master 应该继续尝试传输。
  - **HREADY** ：高电平表明 **Slave** 传输完成；低电平表明 **Slave** 需要延长传输周期
- AHB Bus Transaction 完成，总线回到 Idle 状态

AHB 基本传输

- 一次传输，两个阶段，流水线
  - Address Phase ，单个周期
  - Data Phase ，可能持续多个周期， HREADY 表示传输结束
  - 地址与数据传输采用流水线方式

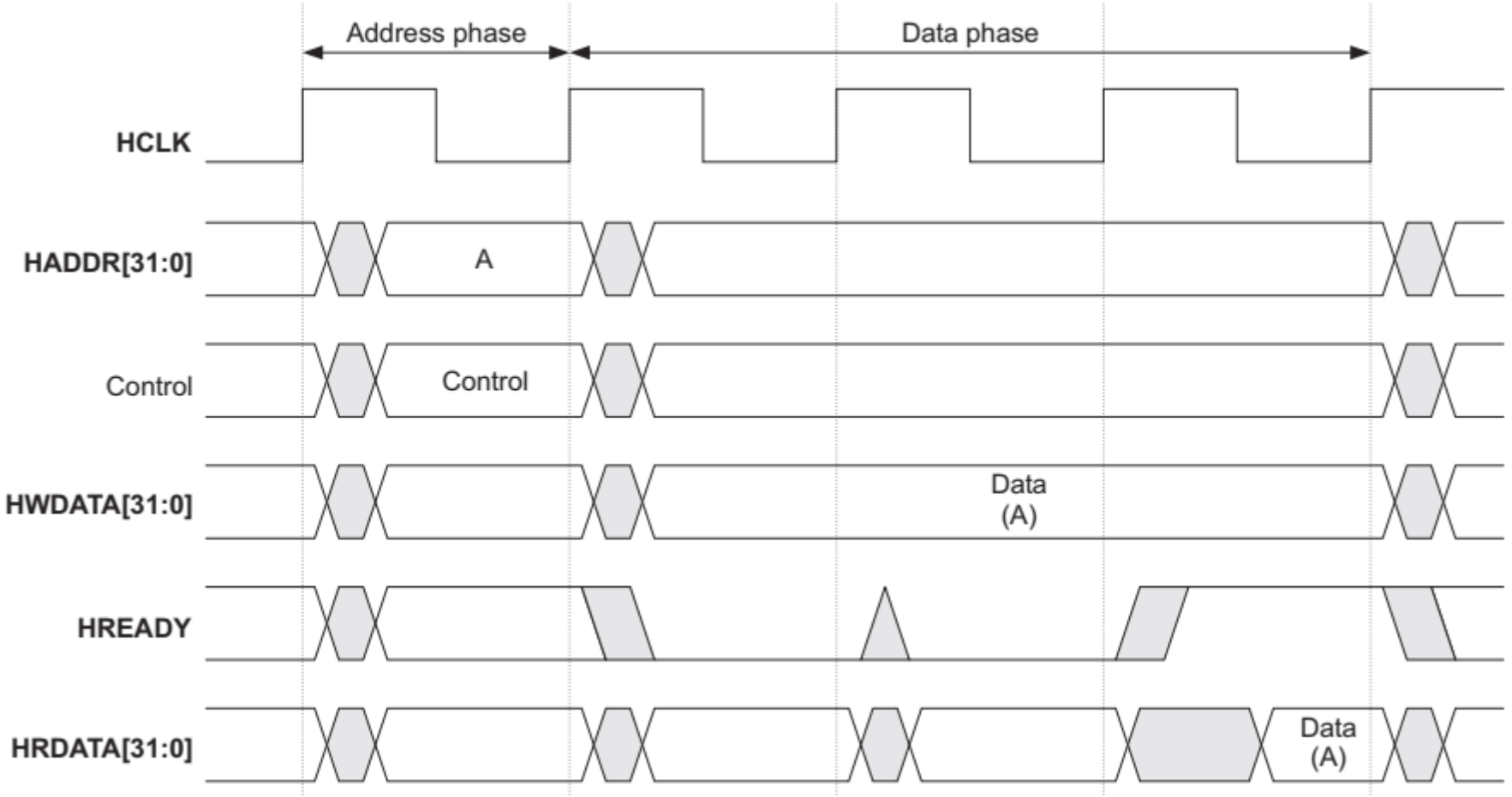


1. Master 在 **HCLK** 的上升沿之后输出地址和控制信号
2. Slave 在时钟的下一个上升沿采样地址和控制信息
3. Slave 接下来发出响应并且该响应被总线主机在第三个时钟的上升沿采样

无等待周期的简单传输

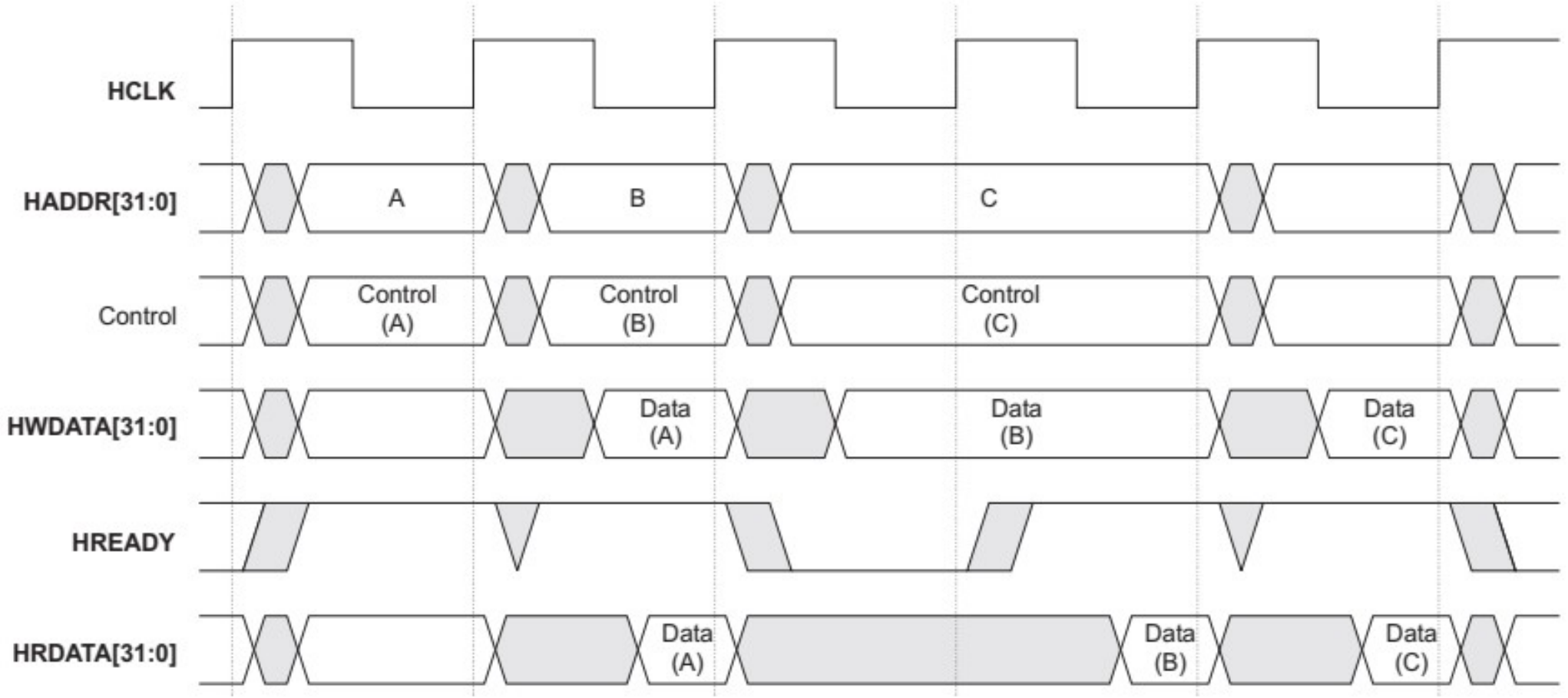
AHB 基本传输：有等待周期的简单传输

1. Slave 可能在数据周期中，插入等待周期
2. 对写操作， Master 必须保持数据在整个扩展周期中稳定
3. 对读传输， Slave 没必要提供有效数据直到传输将要结束时



AHB 基本传输：采用流水线的多次传输

- 在 Slave hready 为高电平的情况下， master 可以连续发起多次传输

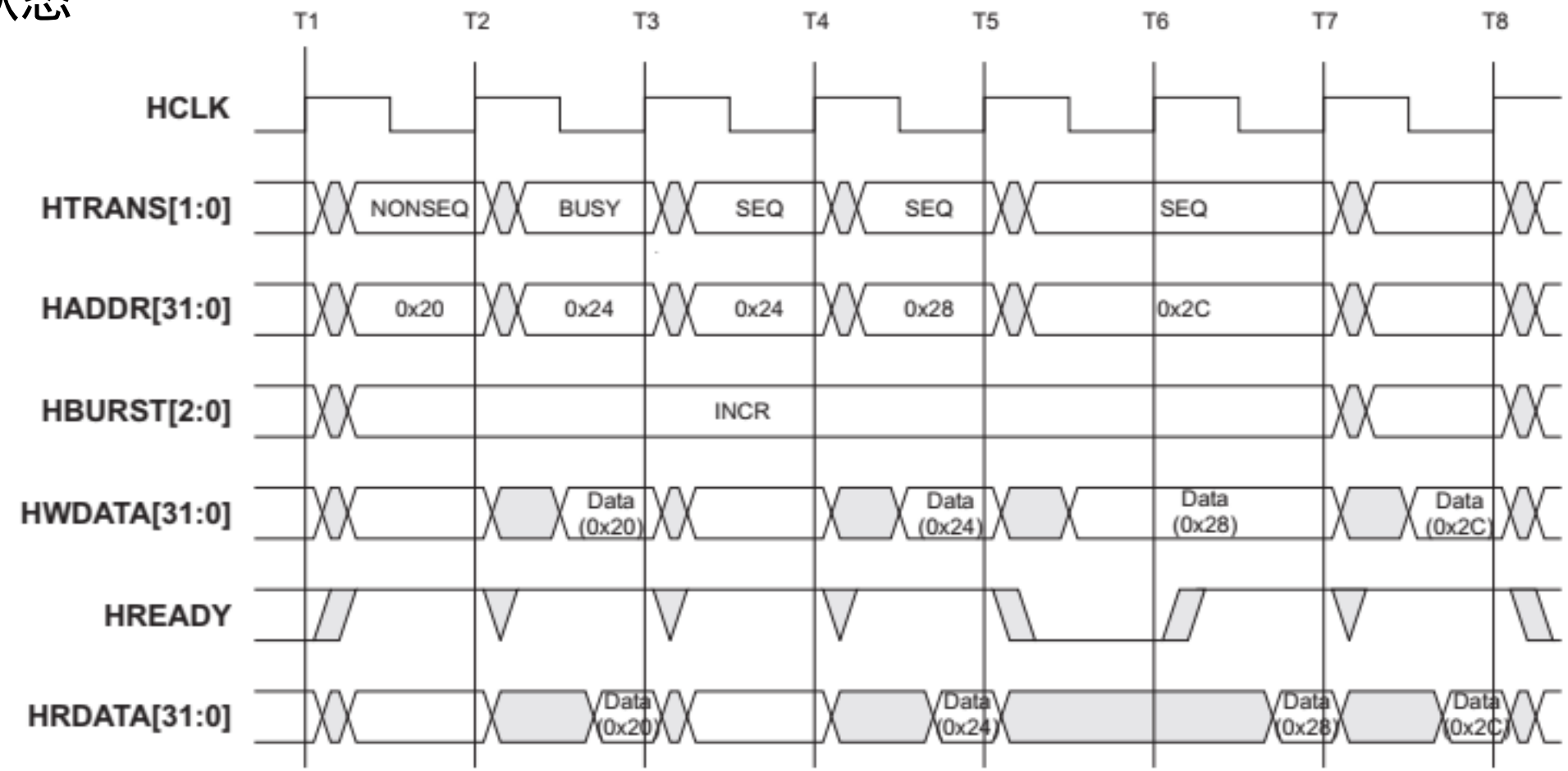


AHB 传输类型

HTRANS[1:0]	Type	Description
00	IDLE	Indicates that no data transfer is required. The IDLE transfer type is used when a bus master is granted the bus, but does not wish to perform a data transfer. Slaves must always provide a zero wait state OKAY response to IDLE transfers and the transfer should be ignored by the slave.
01	BUSY	The BUSY transfer type allows bus masters to insert IDLE cycles in the middle of bursts of transfers. This transfer type indicates that the bus master is continuing with a burst of transfers, but the next transfer cannot take place immediately. When a master uses the BUSY transfer type the address and control signals must reflect the next transfer in the burst. The transfer should be ignored by the slave. Slaves must always provide a zero wait state OKAY response, in the same way that they respond to IDLE transfers.
10	NONSEQ	Indicates the first transfer of a burst or a single transfer. The address and control signals are unrelated to the previous transfer. Single transfers on the bus are treated as bursts of one and therefore the transfer type is NONSEQUENTIAL.
11	SEQ	The remaining transfers in a burst are SEQUENTIAL and the address is related to the previous transfer. The control information is identical to the previous transfer. The address is equal to the address of the previous transfer plus the size (in bytes). In the case of a wrapping burst the address of the transfer wraps at the address boundary equal to the size (in bytes) multiplied by the number of beats in the transfer (either 4, 8 or 16).

AHB 传输类型

- T1 ： burst 传输的开始，传输类型为 NONSEQ
- T2 ： master 使用了 BUSY 传输来延时下一次传输的开始
- T4 ：执行 burst 的第三次传输， slave 不能完成该传输，用 **HREADY** 来插入一个等待状态



AHB Burst 传输

- 支持 4 、 8 、 16-beat 、单个和未定义长度的 burst 传输
- 支持 Incrementing 和 wrapping 两种 burst 传输行为
- Wrapping burst ，对于 4-beat 的 wrapping burst 字传输，在 16 字节边界回环
  - 0x34->0x38->0x3C->0x30
- Burst 传输的数据总量由 HBURST 和 HSIZE 共同确定
- 地址对齐：字传输必须对齐到字地址边界（A[1:0]=00），半字传输

HBURST[2:0]	Type	Description
000	SINGLE	Single transfer
001	INCR	Incrementing burst of unspecified length
010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat incrementing burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat incrementing burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat incrementing burst

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

AHB Burst 传输：地址计算

- 地址计算根据 HBURST 和 HSIZE 进行
- 假定 HSIZE=010 （ 32-bits ）
- 假定起始地址为 0x48

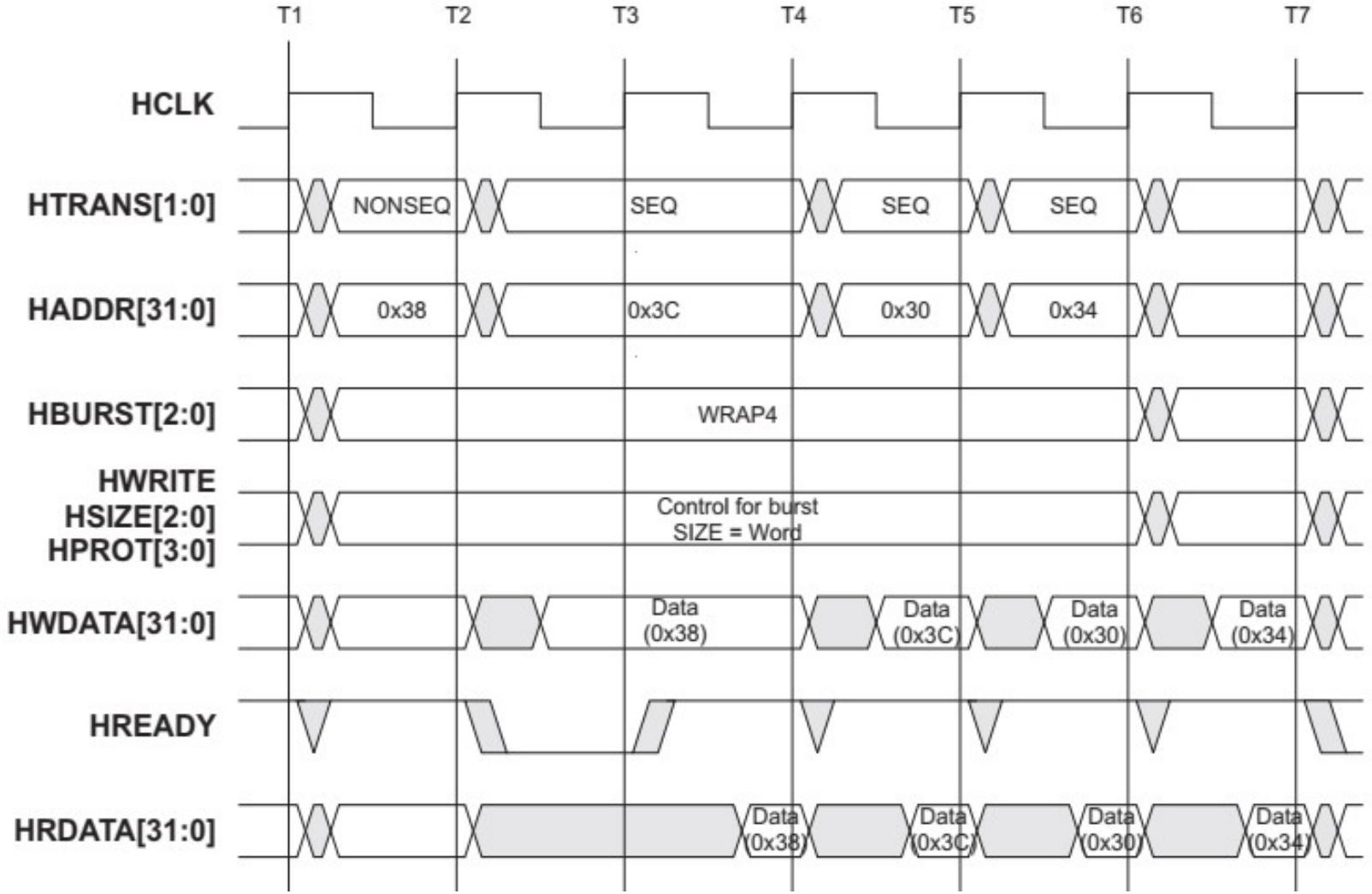
HBURST	Type	Address
000	SINGLE	0x48
001	INCR	0x48, 0x4C, 0x50,… The most useful
010	WRAP4	0x48, 0x4C, 0x40, 0x44
011	INCR4	0x48, 0x4C, 0x50, 0x54
100	WRAP8	0x48, 0x4C, 0x50, 0x54, 0x58, 0x5c, 0x40, 0x44
101	INCR8	0x48, 0x4C, 0x50, 0x54, 0x58, 0x5c, 0x60, 0x64
110	WRAP16	0x48, 0x4C,…, 0x7c, 0x40, 0x44
111	INCR16	0x48, 0x4C,…, 0x7c, 0x80, 0x84

AHB Burst 传输：地址计算

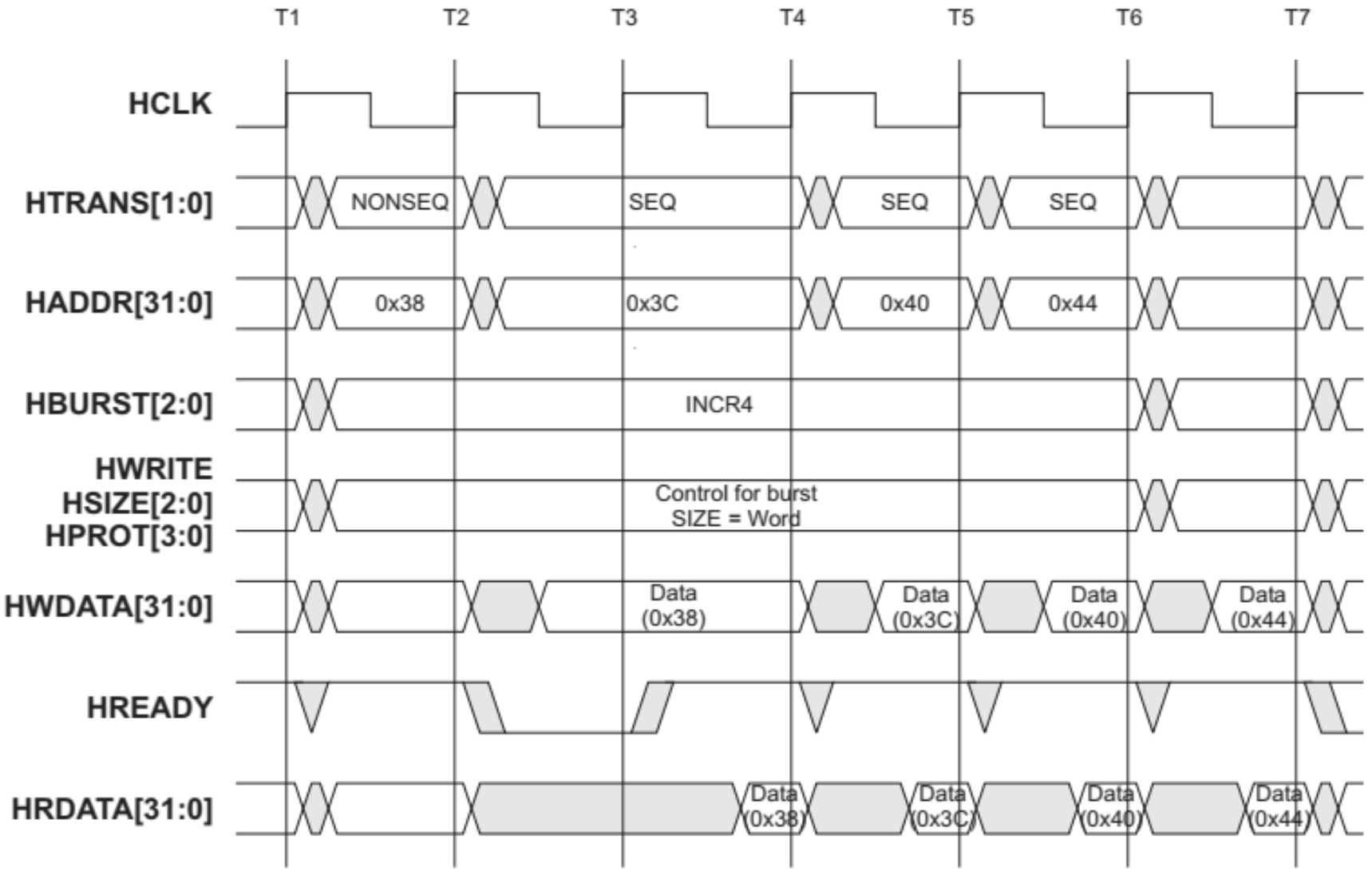
- 地址计算根据 HBURST 和 HSIZE 进行
- 假定 HSIZE=010 （ 32-bits ）
- 假定起始地址为 0x48

HBURST	Type	Address
000	SINGLE	0x48
001	INCR	0x48, 0x4C, 0x50,… The most useful
010	WRAP4	0x48, 0x4C, 0x40, 0x44
011	INCR4	0x48, 0x4C, 0x50, 0x54
100	WRAP8	0x48, 0x4C, 0x50, 0x54, 0x58, 0x5c, 0x40, 0x44
101	INCR8	0x48, 0x4C, 0x50, 0x54, 0x58, 0x5c, 0x60, 0x64
110	WRAP16	0x48, 0x4C,…, 0x7c, 0x40, 0x44
111	INCR16	0x48, 0x4C,…, 0x7c, 0x80, 0x84

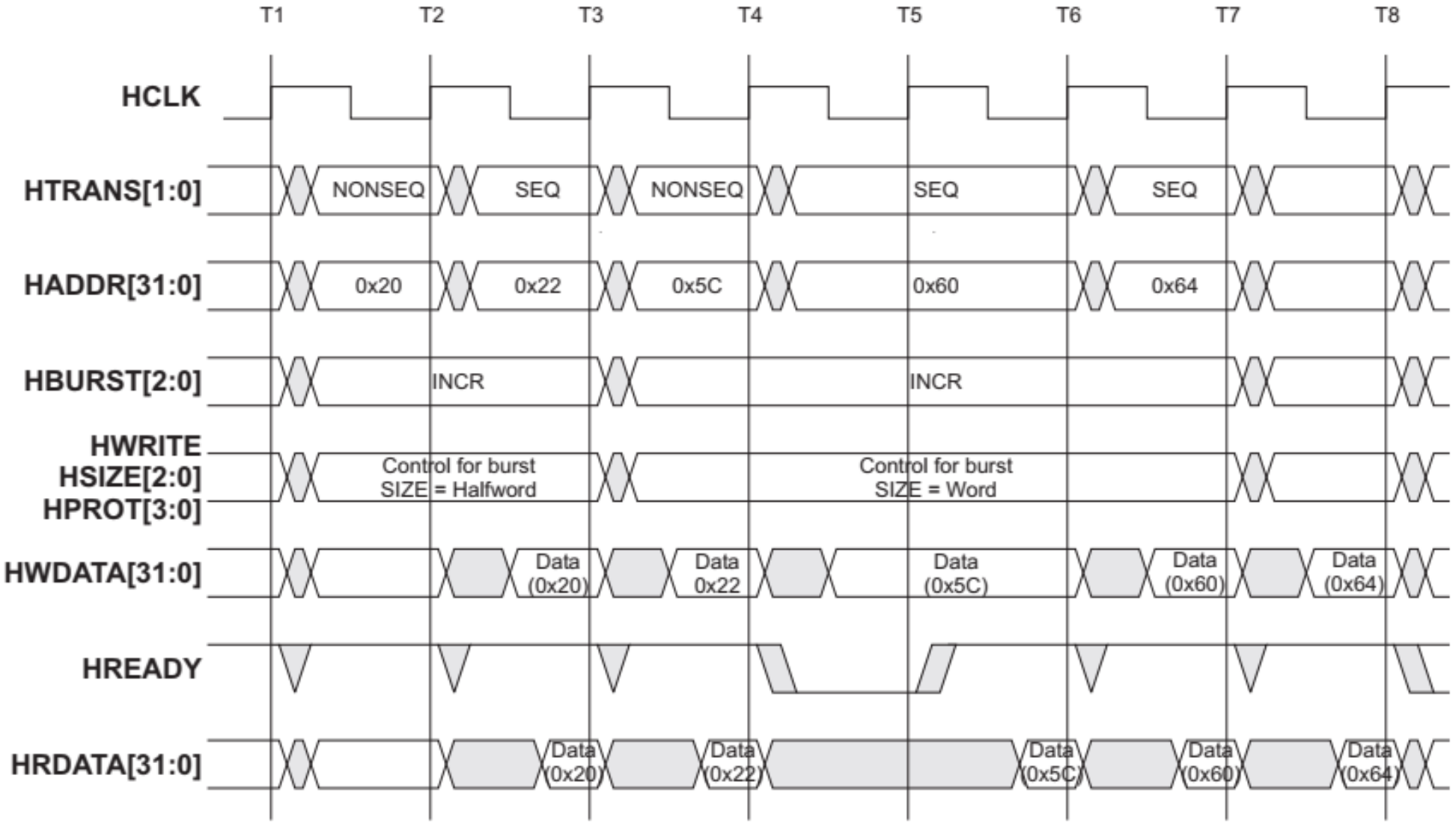
AHB Burst 传输： WRAP4 burst



AHB Burst 传输： INCR4 burst

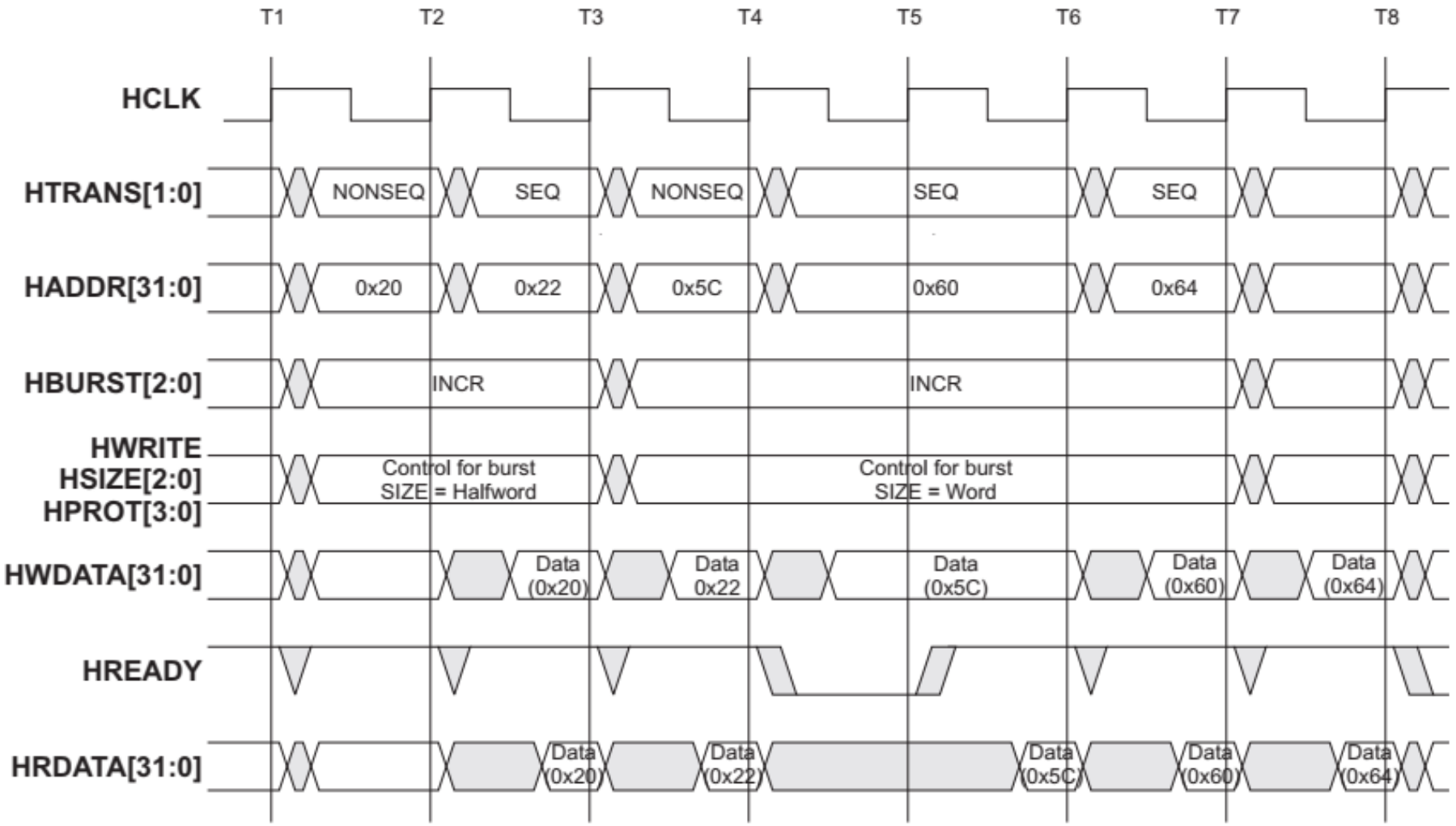


AHB Burst 传输：未定义长度的 burst



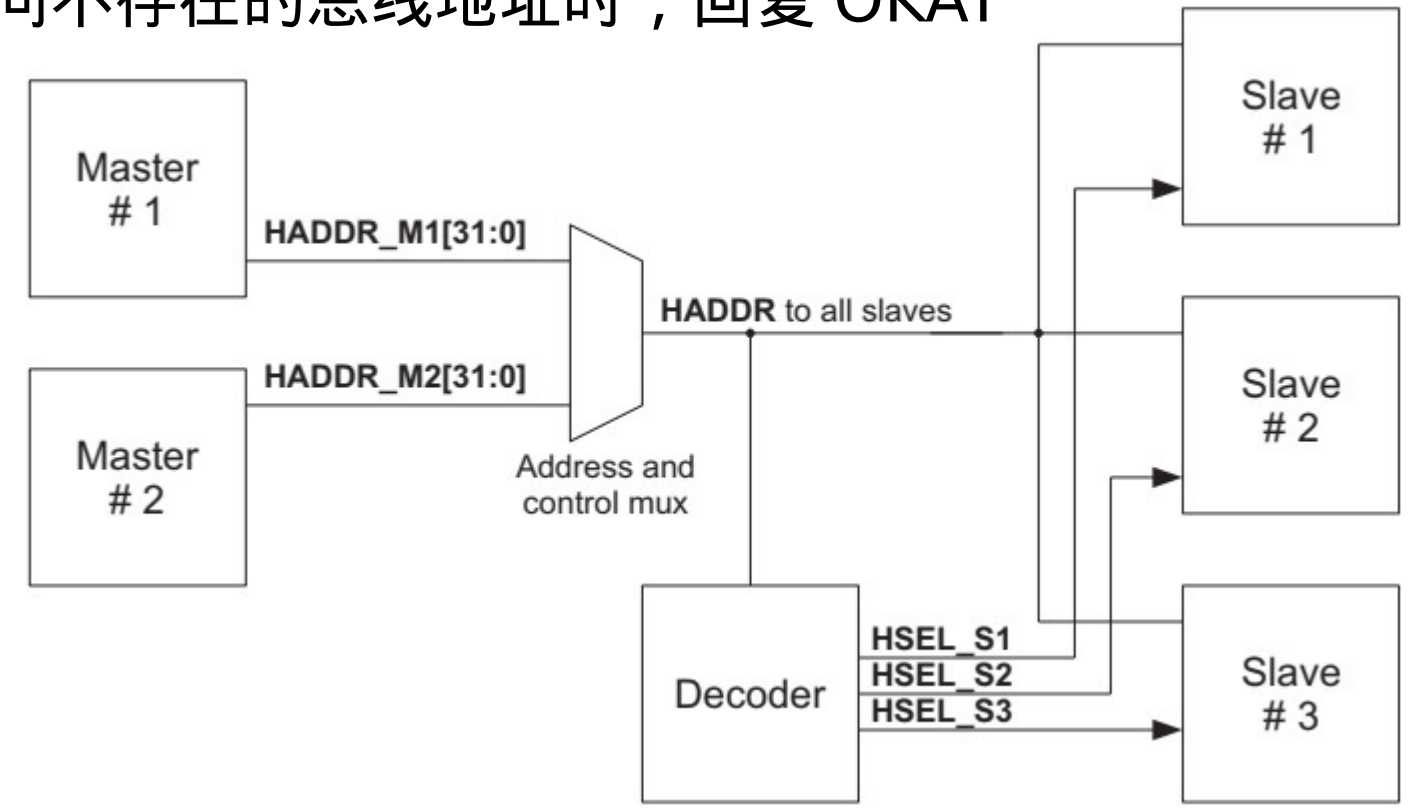
AHB Burst 传输：跨越 1K 边界

- Burst 传输不能跨越 1k-byte 边界
  - 当存在跨越边界的行为时，必须将其拆分为两次 burst 传输



地址译码

- 地址译码器用于产生 HSELx 信号，来选择 slave
- Slave 在 HSELx 和 HREADY 为高电平时，采样地址和控制信号
- Default Slave ，用于产生正常的总线响应
  - NOSEQ 或 SEQ 访问不存在的总线地址时，回复 ERROR
  - IDLE 或 BUSY 访问不存在的总线地址时，回复 OKAY



AHB Slave 响应

- 被选中的 AHB Slave 必须响应总线传输
  - 可能的响应
  - 传输完成
  - 插入等待状态（ HREADY ）
  - 发出错误信号，表示传输失败
  - 分离（ SPLIT ）传输，使得总线可用于其它传输

- HREADY ：传输完成信号

- HRESP[1:0] ：传输响应信号
  - 00 ： OKAY ， 传输成功
  - 01 ： ERROR ，传输失败
  - 10 ： RETRY ，传输未完成，请求 master 重新开始一次传输
  - 11 ： SPLIT ，传输未完成，请求 master 分离一次传输

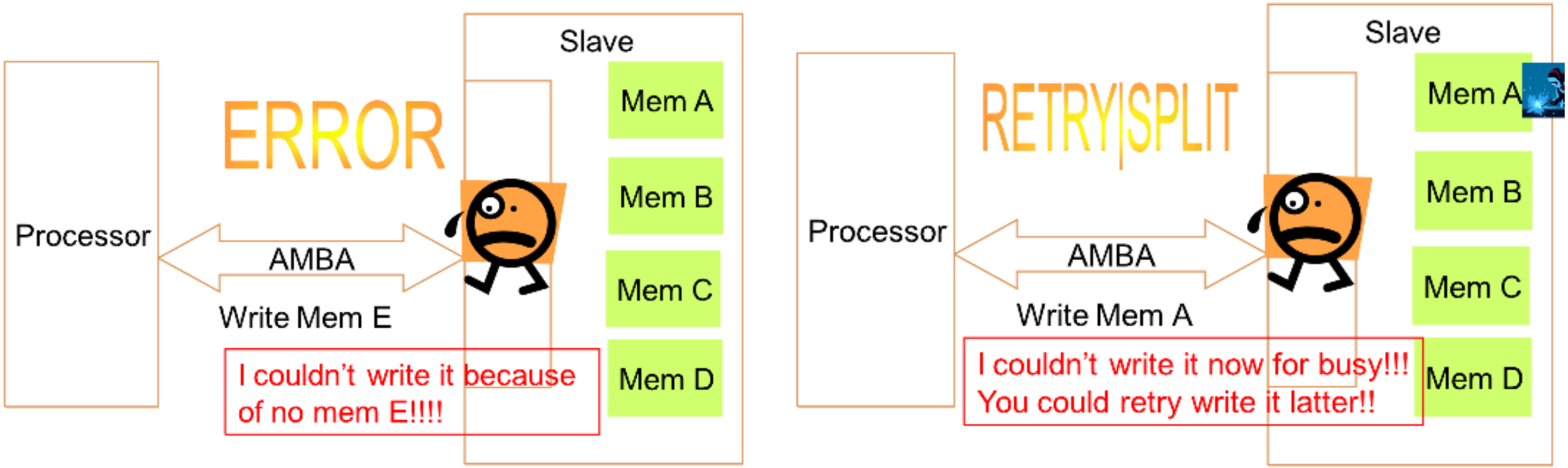
AHB Slave 响应

- HRESP[1:0] ：传输响应信号
  - OKAY ：单周期响应
  - ERROR ：两周期响应
  - RETRY ：两周期响应
  - SPLIT ：两周期响应

- 总线的流水特性 Slave 需要两个周期的响应

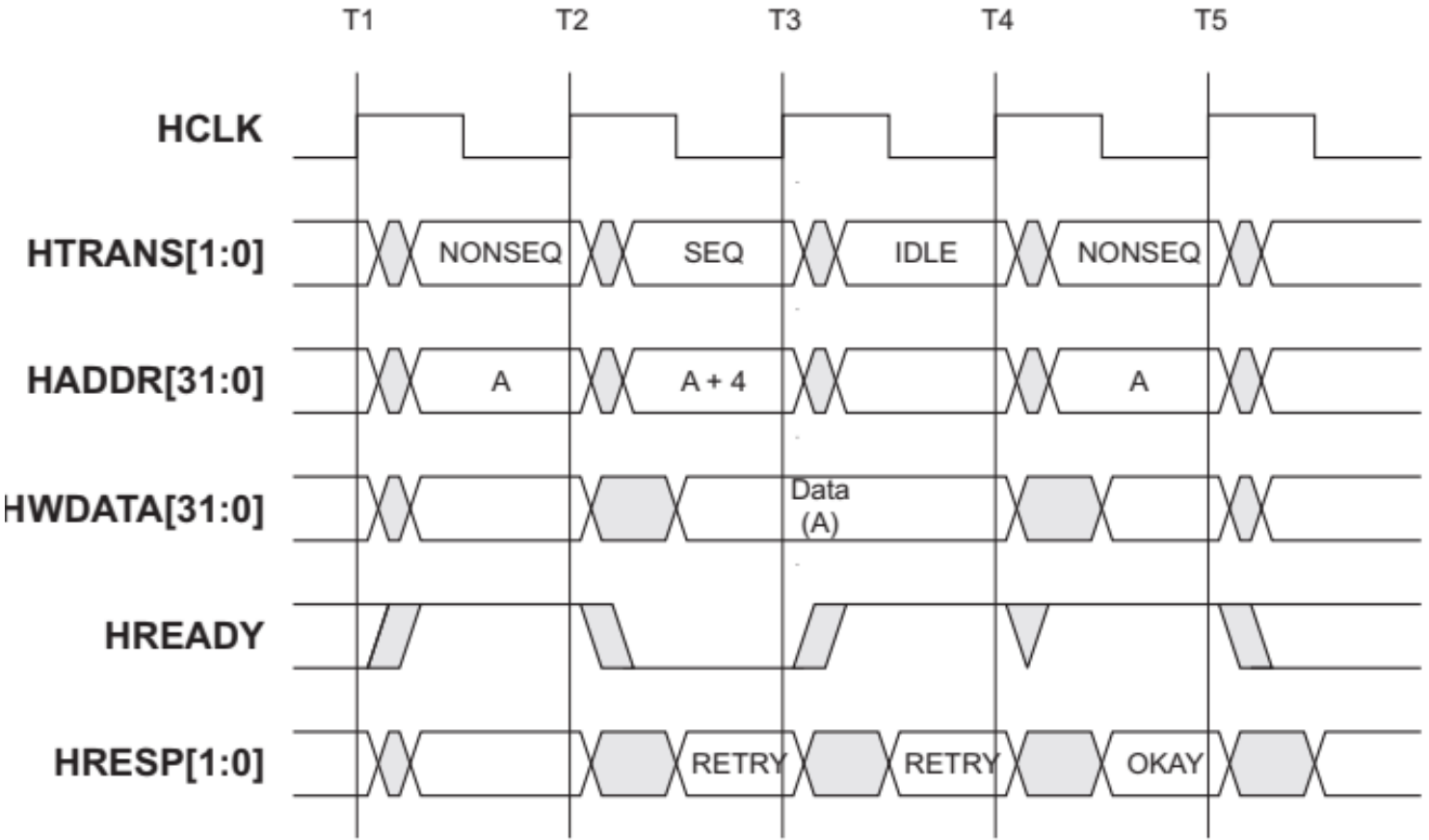
在回复 ERROR、 RETRY 或 SPLIT 之前， slave 已经收到下一次传输的地址，故需要在此处插入一个等待状态，使得 slave 有足够时间处理下一次传输；

AHB Slave 响应


Slave
ERROR
Processor
AMBA
Write Mem E
I couldn't write it because of no mem E!!!!
Mem A
Mem B
Mem C
Mem D
Slave
RETRY|SPLIT
Processor
AMBA
Write Mem A
I couldn't write it now for busy!!!
You could retry write it latter!!
Mem A
Mem B
Mem C
Mem D

AHB Slave 两周期响应： Retry 响应

- RETRY 和 SPLIT 不同之处主要存在于仲裁方式：
  - RETRY ： arbiter 通常会使用原有的优先级进行仲裁
  - SPLIT ： arbiter 会调整优先级，以方便其它请求总线的 master 可以访问总线
- 对于 master 而言，可以使用相同的方式处理 RETRY 和 SPLIT 响应



AHB 数据总线

- 非三态总线，读数据和写数据总线分开
- 字节序（ endianness ），协议中未定义，所有总线设备需保持一致

32-bit little-endian 数据总线中的有效字节

Transfer size	Address offset	DATA [31:24]	DATA [23:16]	DATA [15:8]	DATA [7:0]
Word	0	Y	Y	Y	Y
Halfword	0	-	-	Y	Y
Halfword	2	Y	Y	-	-
Byte	0	-	-	-	Y
Byte	1	-	-	Y	-
Byte	2	-	Y	-	-
Byte	3	Y	-	-	-

32-bit big-endian 数据总线中的有效字节

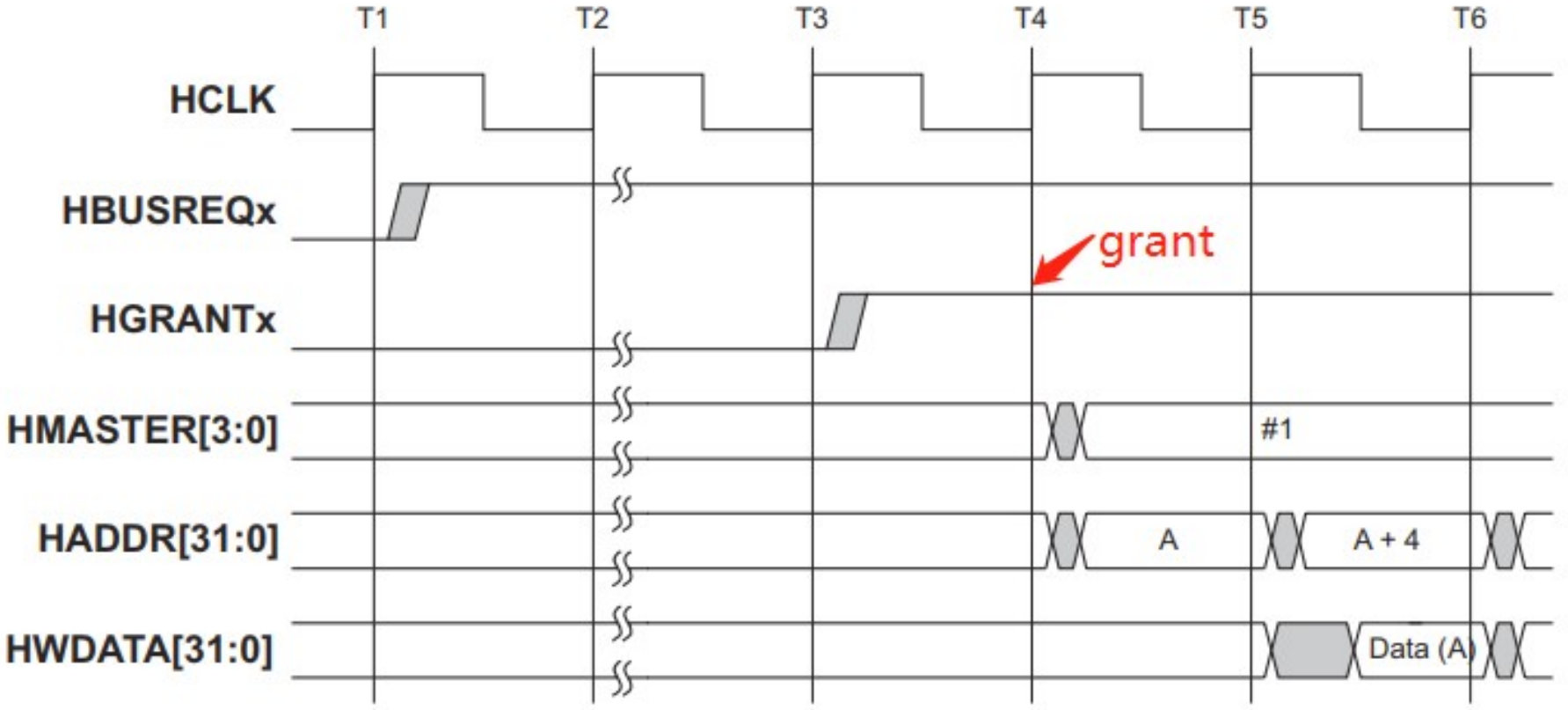
Transfer size	Address offset	DATA [31:24]	DATA [23:16]	DATA [15:8]	DATA [7:0]
Word	0	Y	Y	Y	Y
Halfword	0	Y	Y	-	-
Halfword	2	-	-	Y	Y
Byte	0	Y	-	-	-
Byte	1	-	Y	-	-
Byte	2	-	-	Y	-
Byte	3	-	-	-	Y

AHB 仲裁信号

- **HBUSREQx** ：
  - Master 总线请求信号，高电平有效
- **HLOCKx** ：
  - Master 请求总线锁定信号，高电平有效
- **HGRANTx** ：
  - Arbiter 给出该信号，表明 master-x 可以访问总线
  - Master-x 控制总线，当 HGRANTx=1 且 HREADY=1
- **HMASTER[3:0]** ：
  - Arbiter 给出该信号，表明哪一个 master 正在进行传输，提供进行 split 的信息
- **HMASTLOCK** ：
  - Arbiter 给出该信号，表明 master 正在进行一次锁定传输
- **HSPLIT[15:0]** ：
  - Slave 用这个信号告诉 arbiter 哪一个 master 允许重新尝试一次 split 传输
  - 每一位对应一个 master

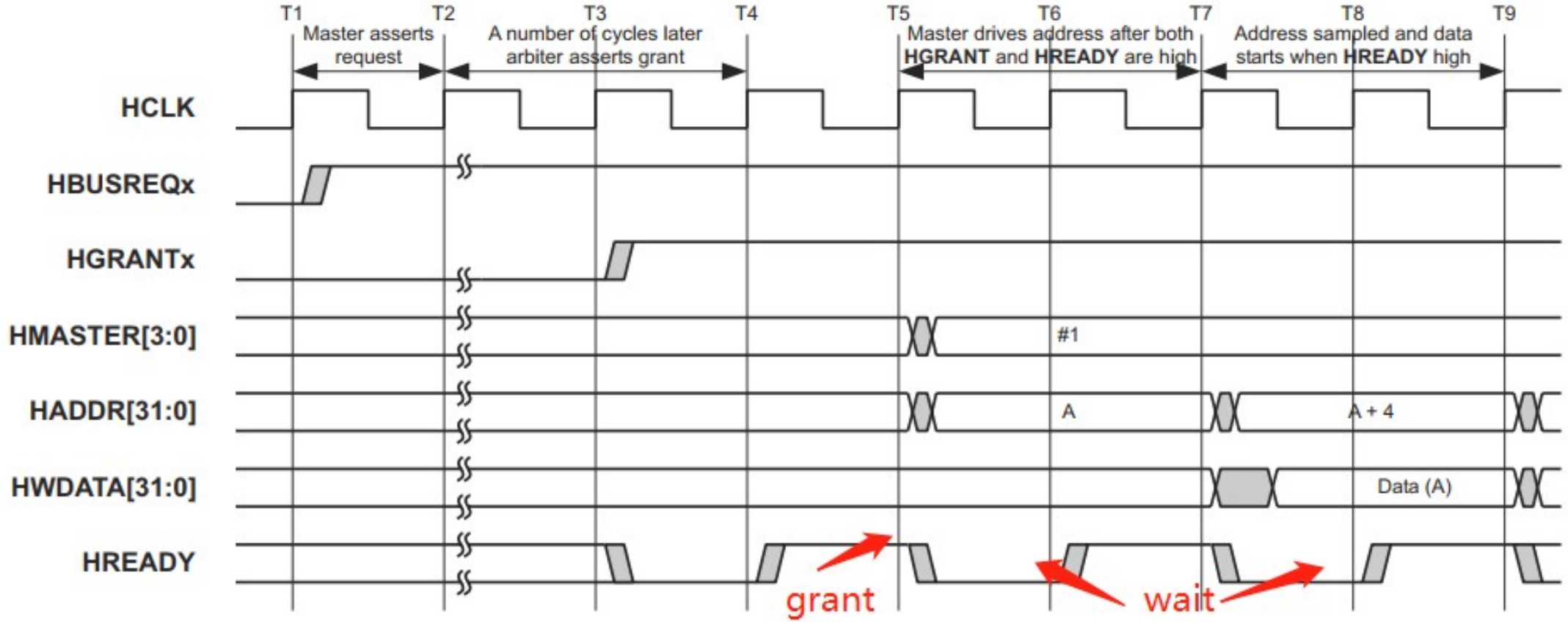
AHB 仲裁：无等待状态的 grant

- Master 发出总线请求 HBUSREQx
- Arbiter 根据总线情况和仲裁策略，将 HGRANTx 信号设置为高电平，允许 master 使用总线



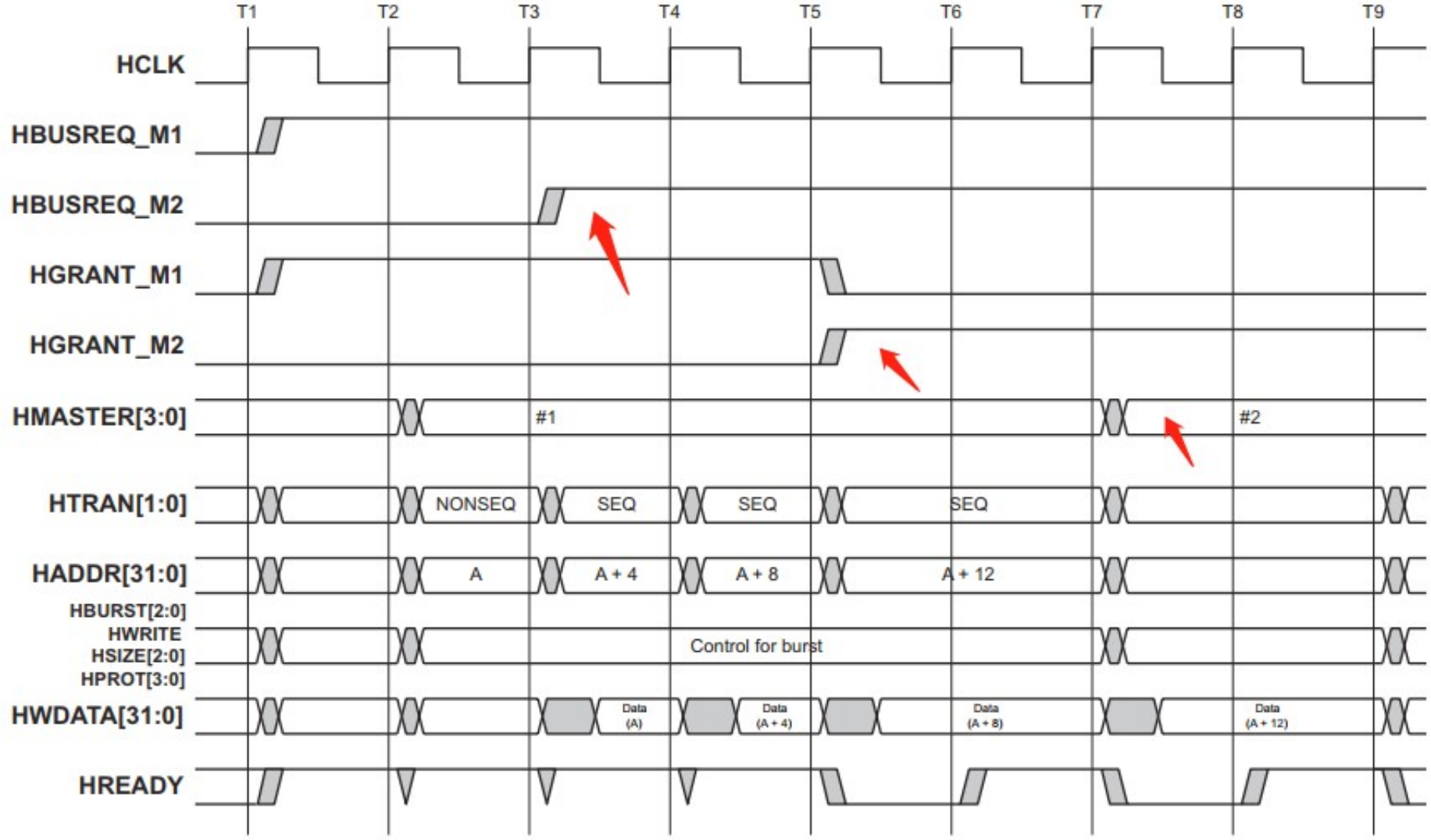
AHB 仲裁：有等待状态的 grant

- Master 发出总线请求 HBUSREQx
- Arbiter 根据总线情况和仲裁策略，将 HGRANTx 信号设置为高电平，允许 master 使用总线
- Master 在 HGRANTx 和 HREADY 同时为高电平时，占用总线进行数据传输

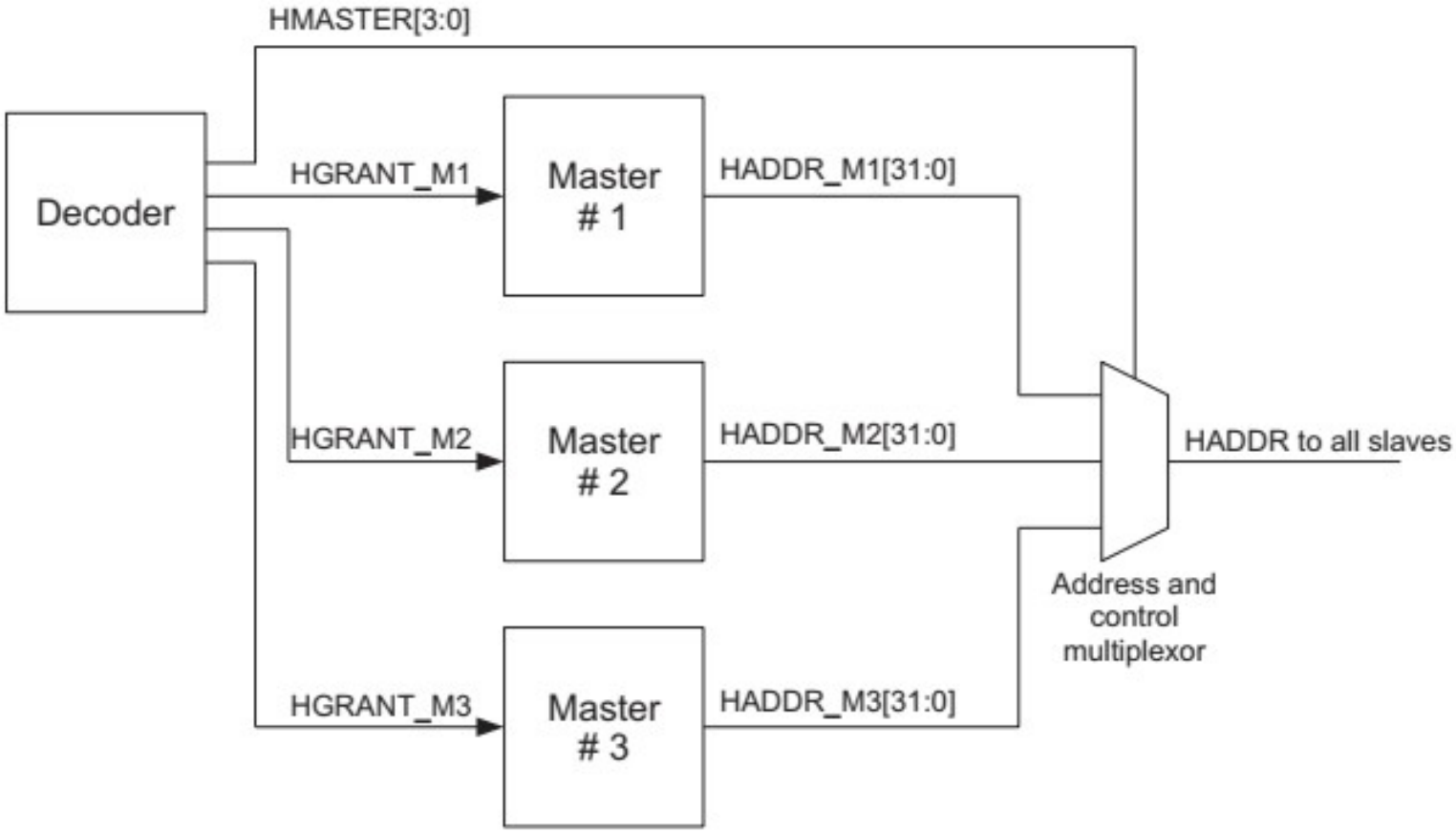


AHB 仲裁： Burst 传输完成后移交总线

- Arbiter 在倒数第二个地址被采样时改变 **HGRANTx** 信号，新的 **HGRANTx** 信息将与 burst 最后一个地址同时被采样。



AHB Master grant 信号



AHB 仲裁的几点说明

- 对于固定长度的 burst 传输
  - 不必持续请求总线
- 对于未定义长度的 burst 传输
  - master 应持续发出 bus 请求信号，直到最后一次传输
- 如果没有 master 请求总线（或所有 master 都在等待 SPLIT 传输完成）
  - 给 default master grant 信号（ AHB 系统必须包含一个 default master ）
  - HTRANS=IDLE

AHB Split 传输过程

1. Master 开始传输，发出地址和控制信息
2. 如 slave 需要多个周期才能获得数据，则 slave 可以给出一个 split 传输响应， slave 记录当前 master 的序号： HMASTER（比如待读取的数据位于外部 DDR 中， DDR 访问延迟很大）
3. 仲裁器改变当前 master 优先级，并 grant 其它的 master 使用总线
4. 当 slave 准备接受本次传输（即 1 中 split 响应对应的 master 数据传输），设置 HSPLITx 信号的相应位为高电平
5. 仲裁器发现 HSPLITx 信号有效之后，恢复对应 master 的优先级
6. 仲裁器 grant 对应 HSPLITx 的 master ，然后该 master 重新开始传输
7. 传输完成， slave 给出 OKAY 响应

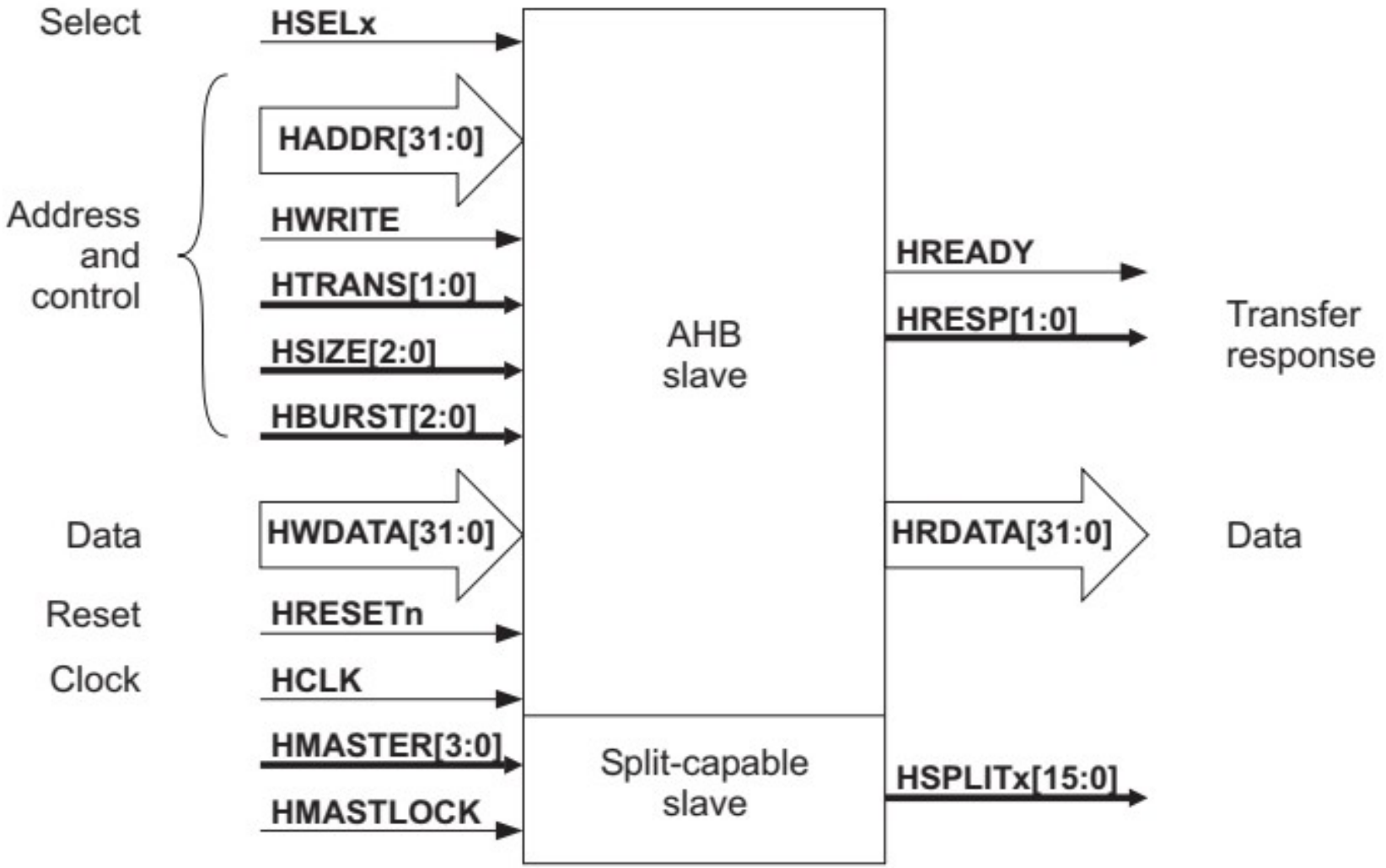
AHB 防止死锁（ Deadlock ）

- 当多个 master 试图访问同一个 slave ，该 slave 发出了 SPLIT 或 RETRY 响应，很可能导致死锁

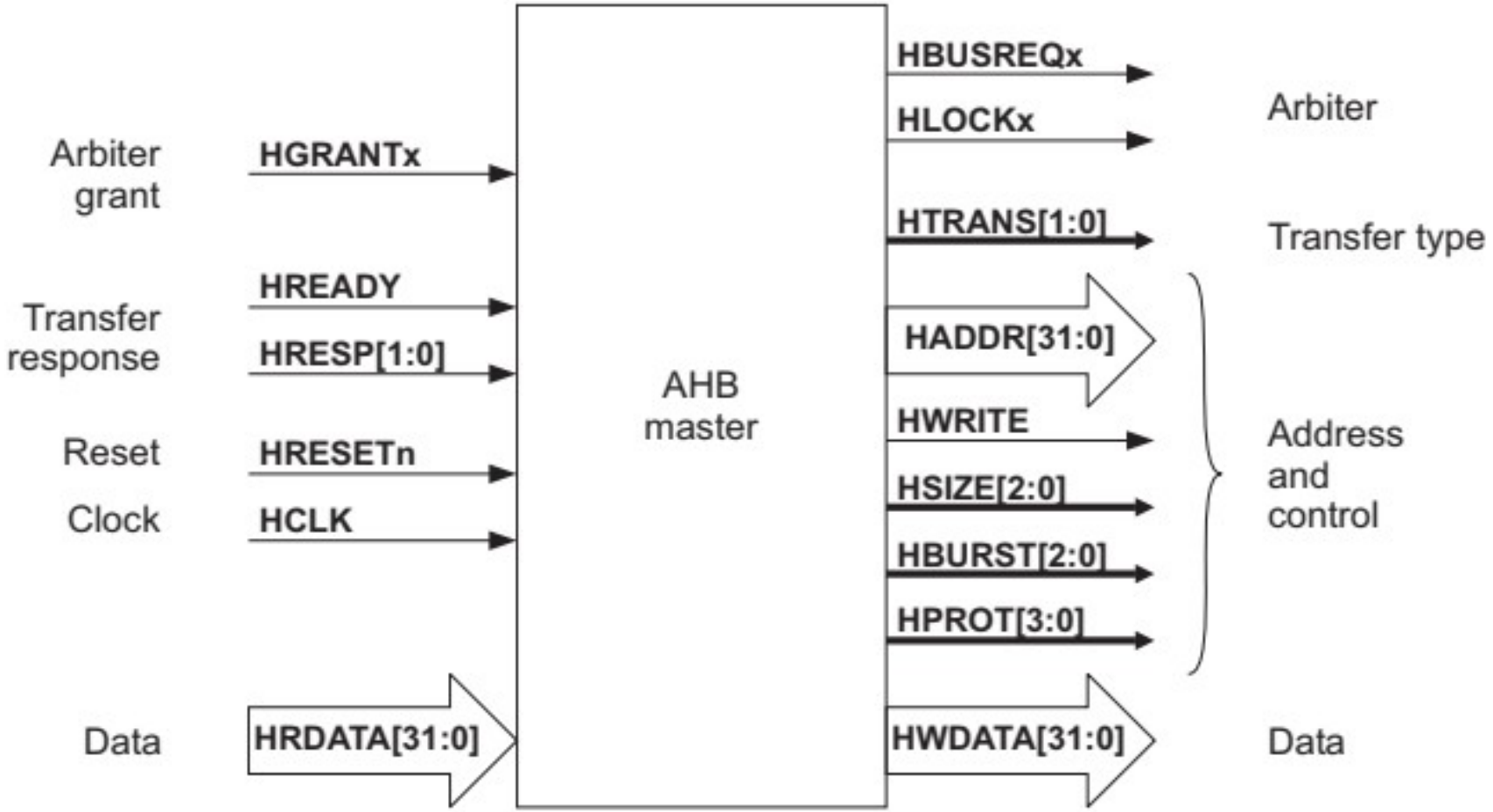
- 给出 SPLIT 响应的 slave ，通过确保能承受系统中每一个 master 的请求来避免死锁

- 给出 RETRY 响应的 slave 在某一时刻只能由一个 master 访问
  - 检查 RETRY 前后的 master 序号是否一致，选择使用硬件保护机制，比如 ERROR 响应，通知仲裁器，产生系统中断，让系统重启等；

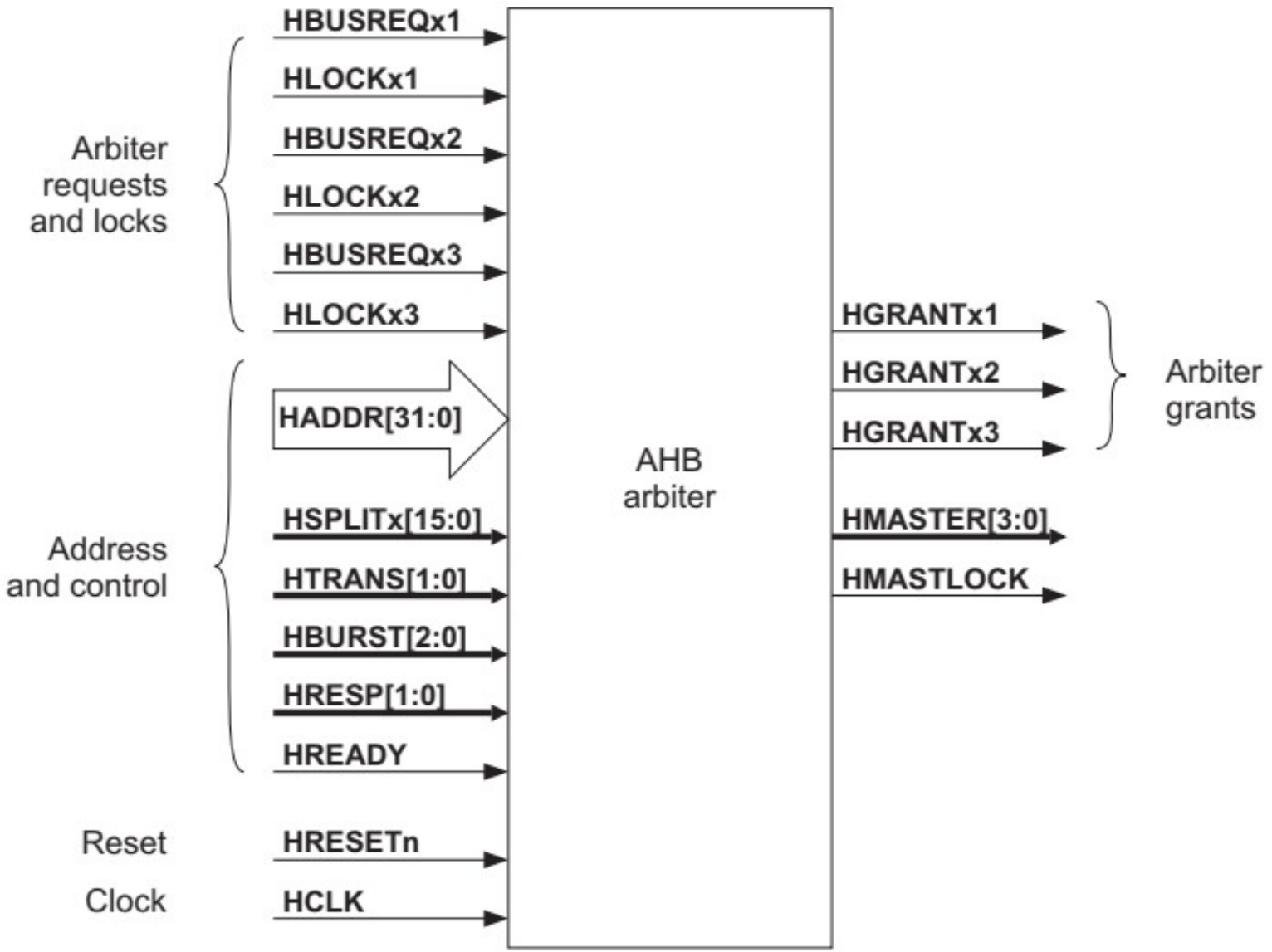
AHB 系统组成：Slave



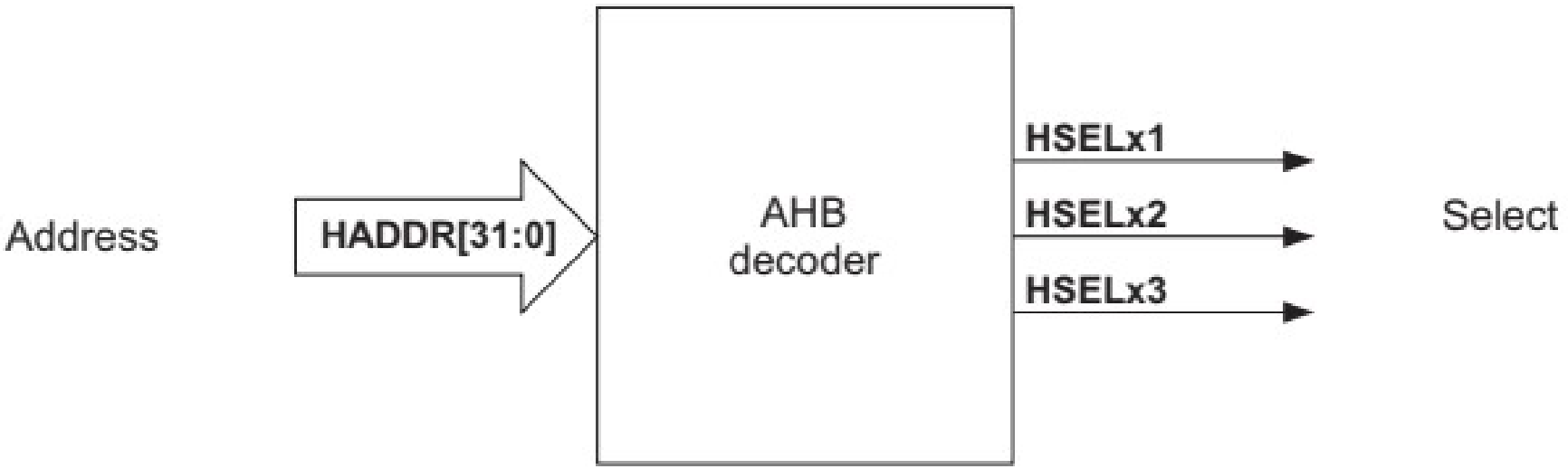
AHB 系统组成： master



AHB 系统组成： arbiter



AHB 系统组成：decoder



AHB 总结

- 主要组成部分：
  - Slave ， master ， arbiter ， decoder
- 传输过程
  - 采用流水线机制的地址、数据周期
- Burst 传输，提高总线读写性能
- 总线仲裁
- Slave 短时间无法响应
  - 插入等待， HREADY 信号拉低
- Slave 长时间无法响应
  - 插入 SPLIT/RETRY
- Master 短时间不能进行传输
  - 插入 BUSY
- 总线上如果只有一个 master ，可以只用 AHB lite 协议（ AMBA 3.0 中定义），无需仲裁