Embedded Busses

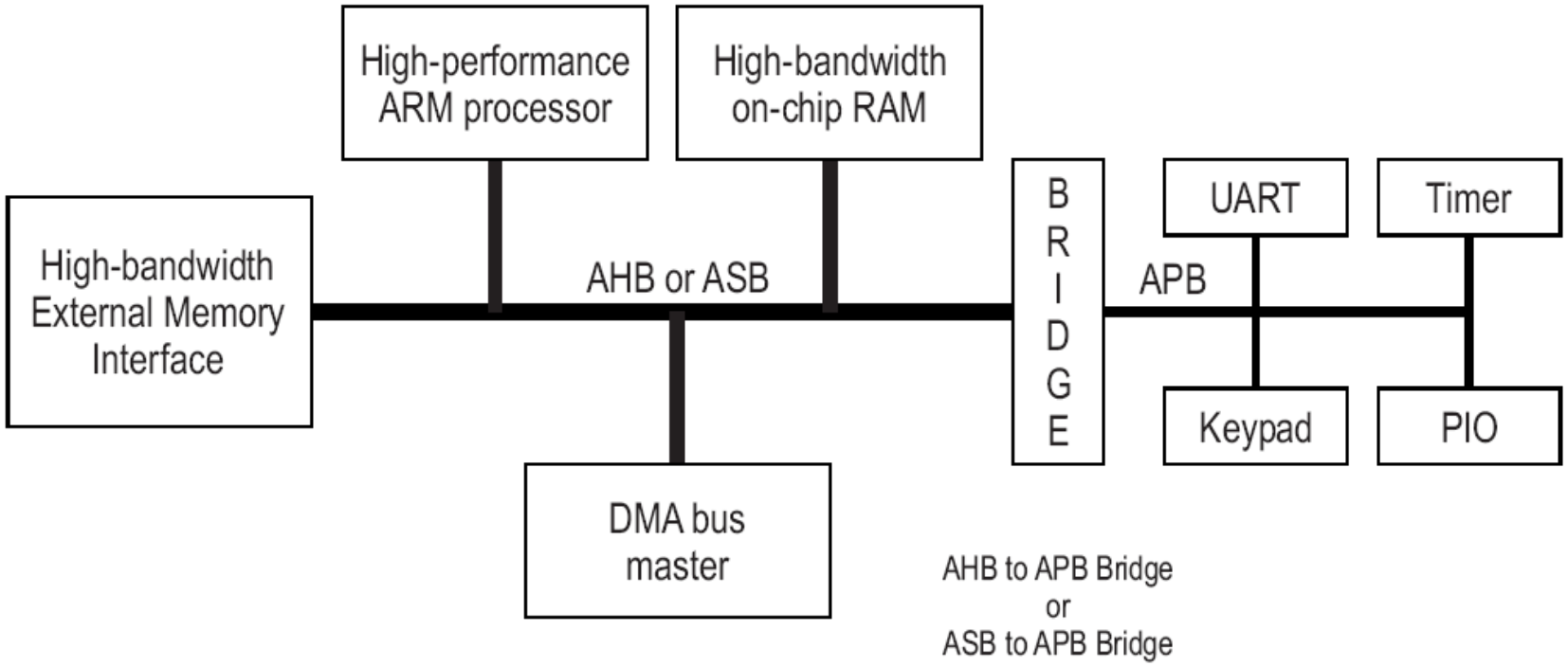
- Large semiconductor
  - CoreConnect (IBM)
  - STBUS (STMicroelectronics)
- Core vendors
  - AMBA (ARM Ltd.)
- Interconnect IP vendors
  - CoreFrame (Palmchip)
  - WishBone (Silicore)
  - SiliconBackPlane (Sonics)
  - Many others!

AMBA Specifications

- The AMBA specification defines an on-chip communications standard for designing high-performance embedded micro-controllers
- Three buses are defined
  - Advanced High-Performance Bus (AHB)
  - Advanced System Bus (ASB)
  - Advanced Peripheral Bus (APB)

System based on an AMBA Bus

- An AMBA system typically contains a high speed bus (ASB or AHB) for CPU, fast memory and DMA and a bus for peripherals (APB), which is connected via a bridge to the high-speed bus



AMBA Buses

- AMBA AHB (new standard)
  - High Performance
  - Pipelined Operation
  - Multiple Bus Masters
  - Burst Transfers
  - Split Transactions
- AMBA ASB (older standard)
  - High Performance
  - Pipelined Operation
  - Multiple Bus Masters
- AMBA APB
  - Low Power
  - Latched Address and Control
  - Simple Interface
  - Suitable for many peripherals

AMBA AHB System

- AHB Master
  - ➔ A bus master is able to *initiate* read and write operations by providing address and control information
    - ✓ Only one bus master can use the bus at the same time
- AHB Slave
  - ➔ A bus slave *responds* to a read and write operation within a given address-space range
    - ✓ The bus slave signals back to the active bus master the success, failure or waiting of the data transfer

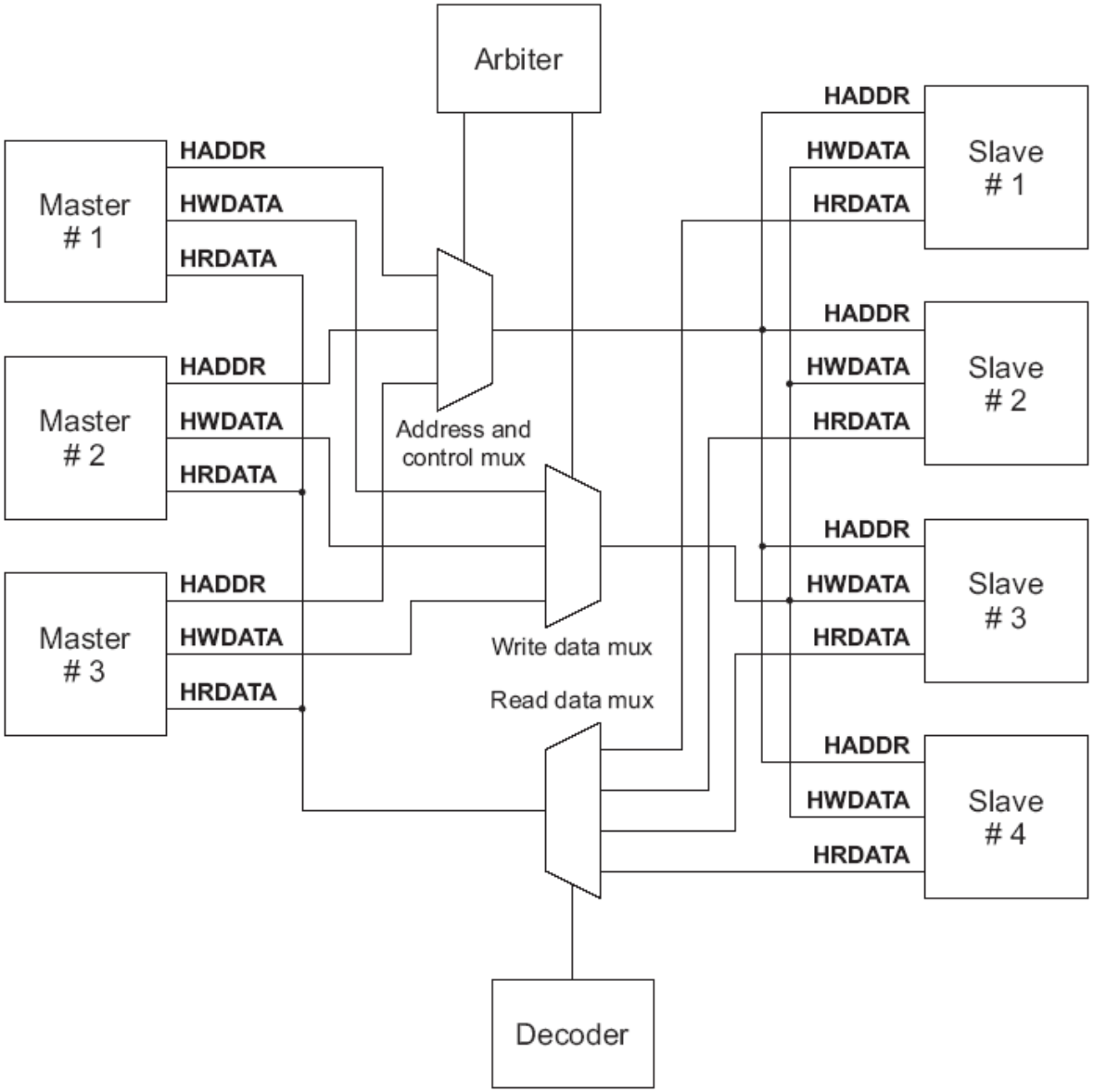
AMBA AHB System

- AHB Arbiter
  - The bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. Even though the arbitration protocol is fixed, any arbitration algorithm, such as highest priority or fair access can be implemented depending on the application requirements
  - An AHB includes only one arbiter
- AHB Decoder
  - The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer
  - A single centralized decoder is required in all AHB implementations

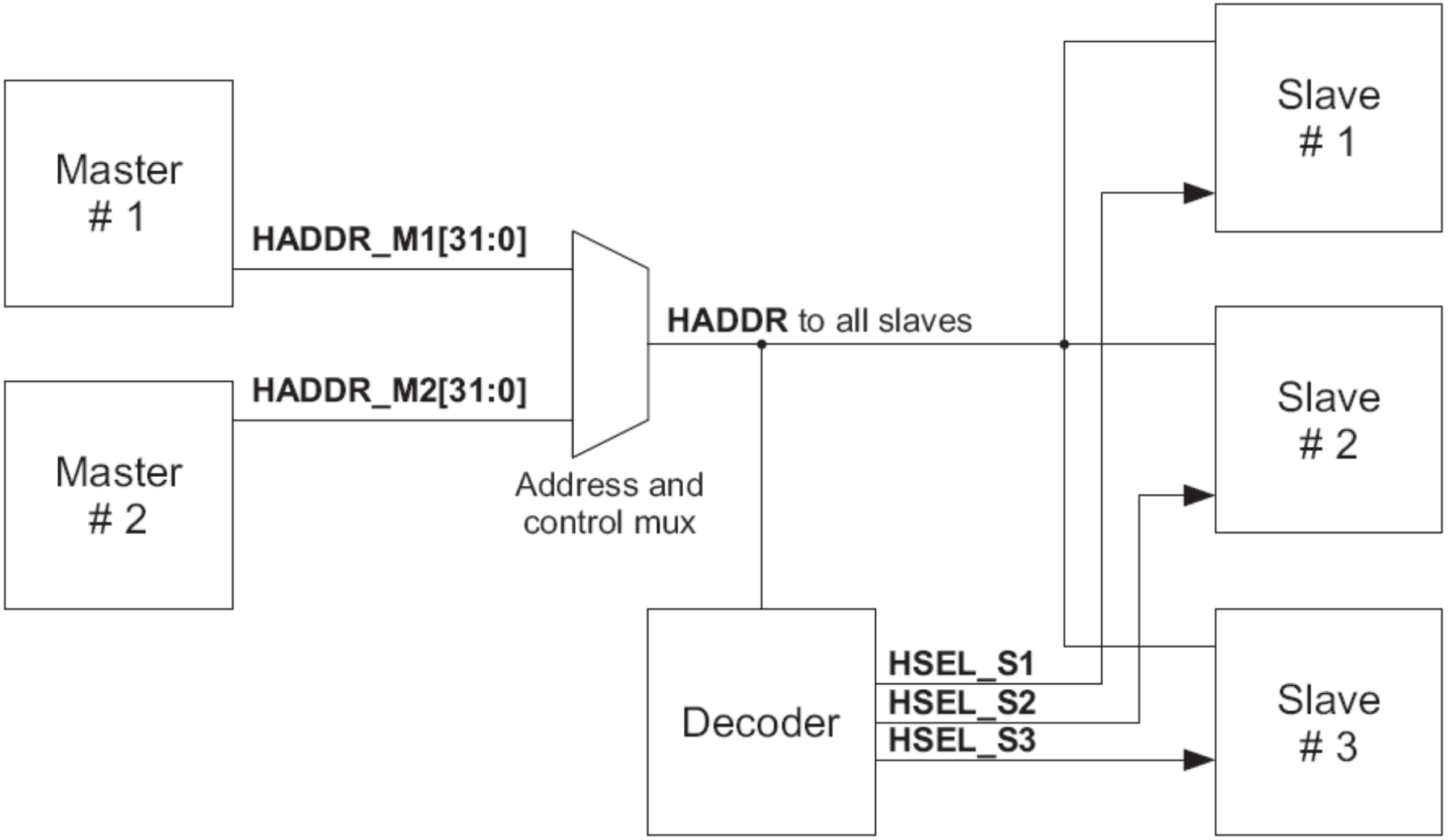
AMBA AHB Bus Interconnection

- AHB Protocol is based on a central multiplexer interconnection scheme
- All bus masters send their request in form of address and control signals
- The arbiter chooses one master. The address and control signals are routed to all slaves
- The decoder selects the signals from the slave that is involved in the transfer with the bus master

AMBA AHB Bus Interconnection

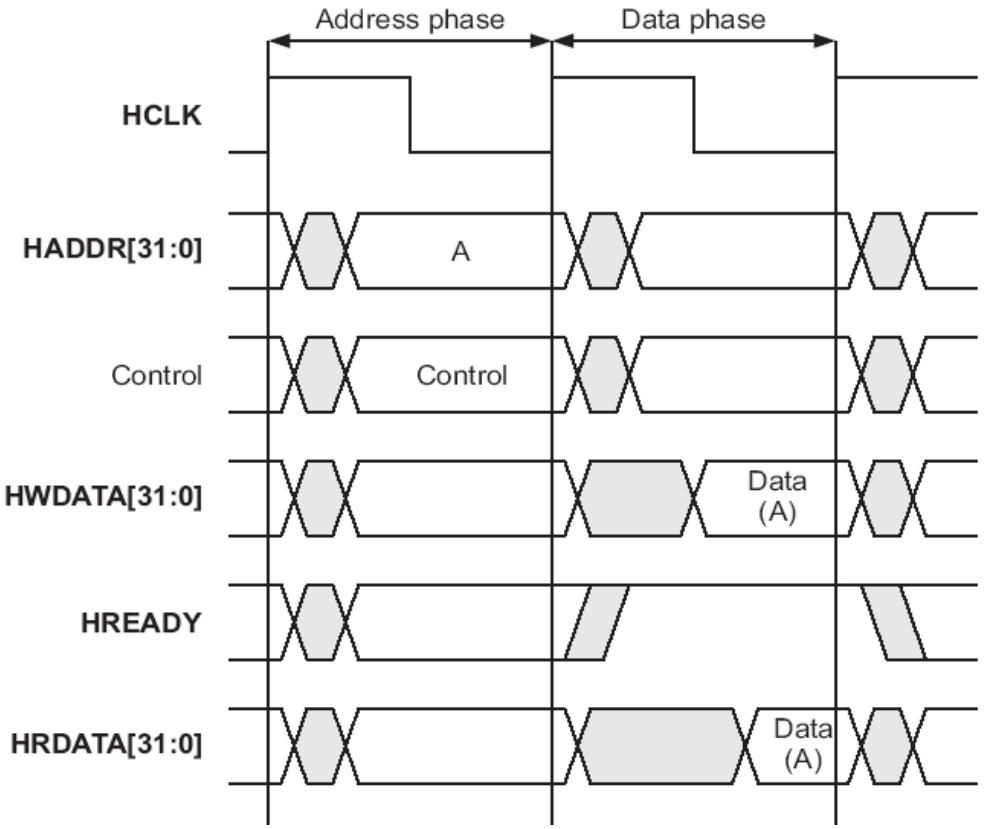


AMBA Address Decoding System

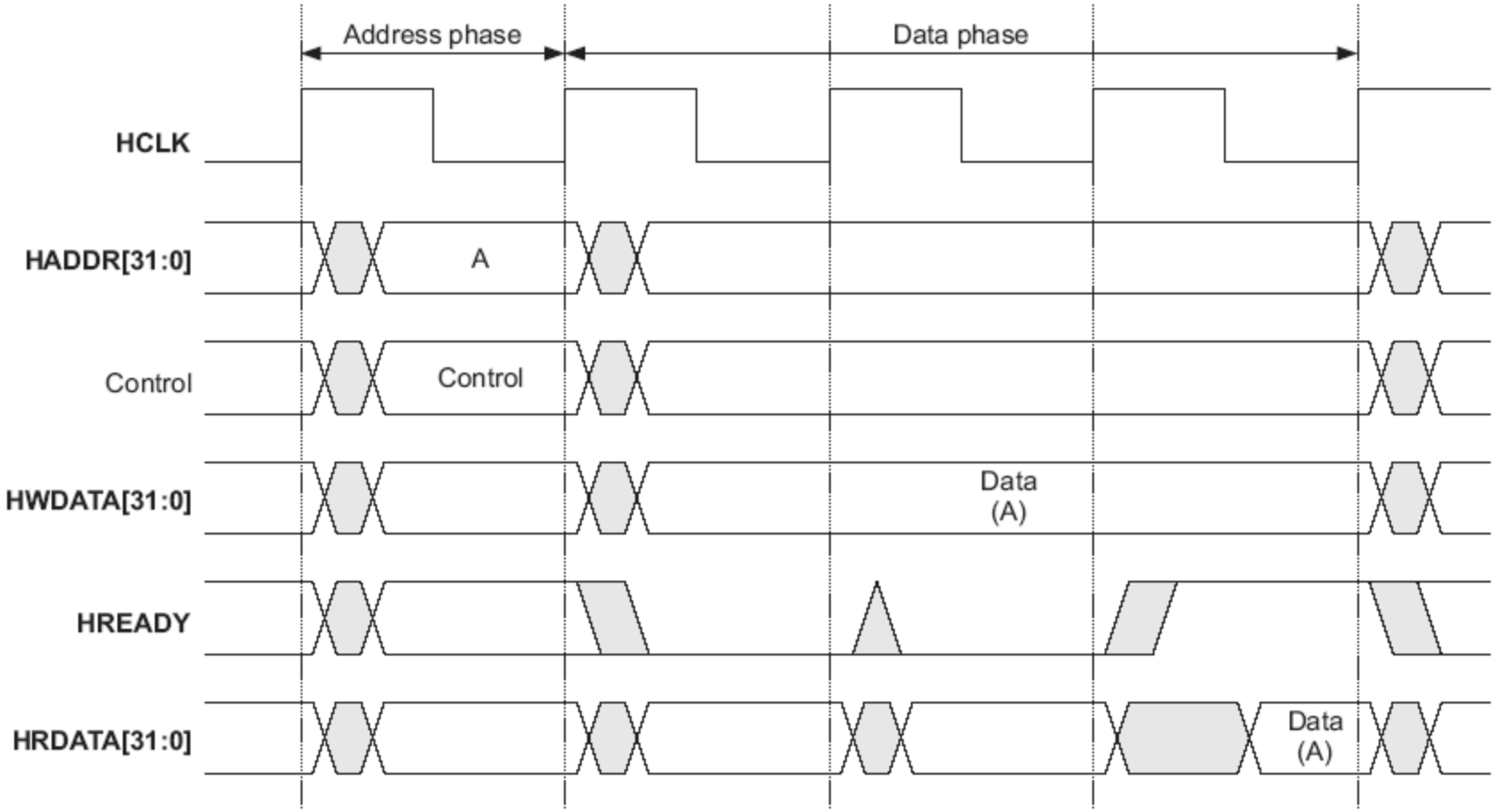


Basic Transfer

- An AHB transfer consists of two distinct sections
  - ➔ The address phase, which lasts only a single cycle
  - ➔ The data phase, which may require several cycles. This is achieved using the HREADY signal

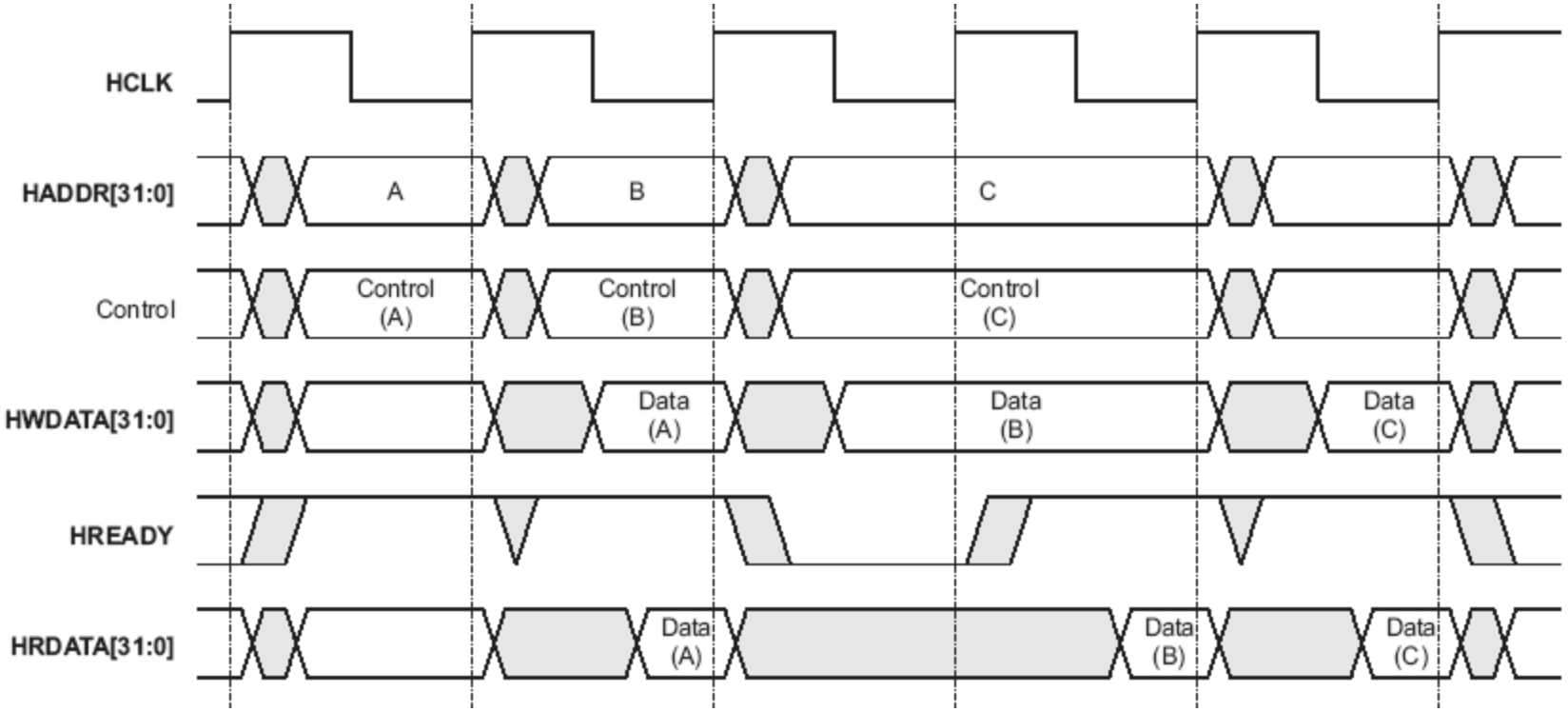


Transfer with wait states



Multiple Transfers

- When a transfer is extended in this way it will have the side-effect of extending the address phase of the following transfer



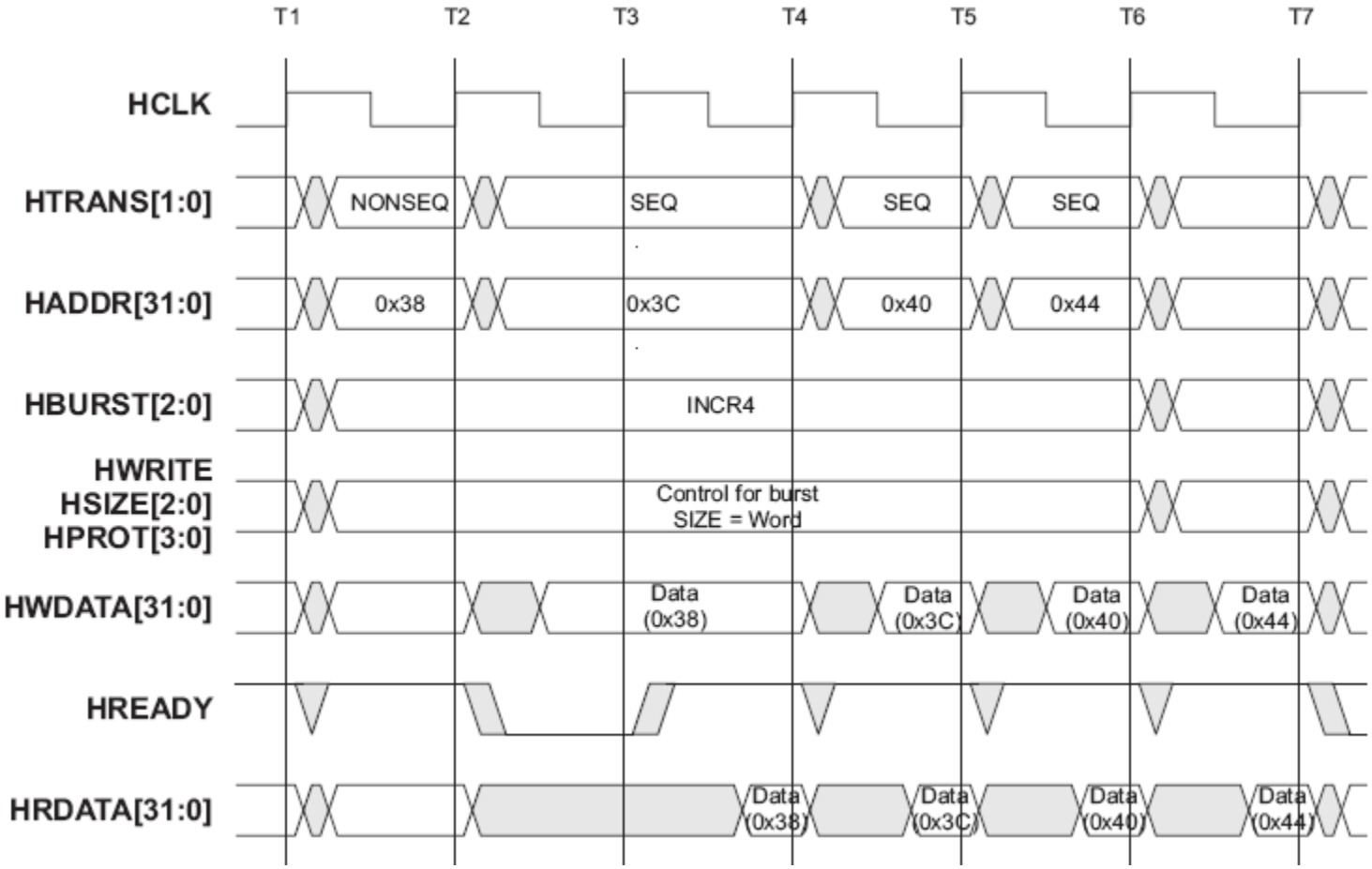
Burst Signal Encoding

- Both incrementing and wrapping bursts are supported in the protocol
  - ➔ *Incrementing bursts* access sequential locations
  - ➔ For *wrapping bursts*, if the start address of the transfer is not aligned to the total number of bytes in the burst (size x beats) then the address of the transfers in the burst will wrap when the boundary is reached

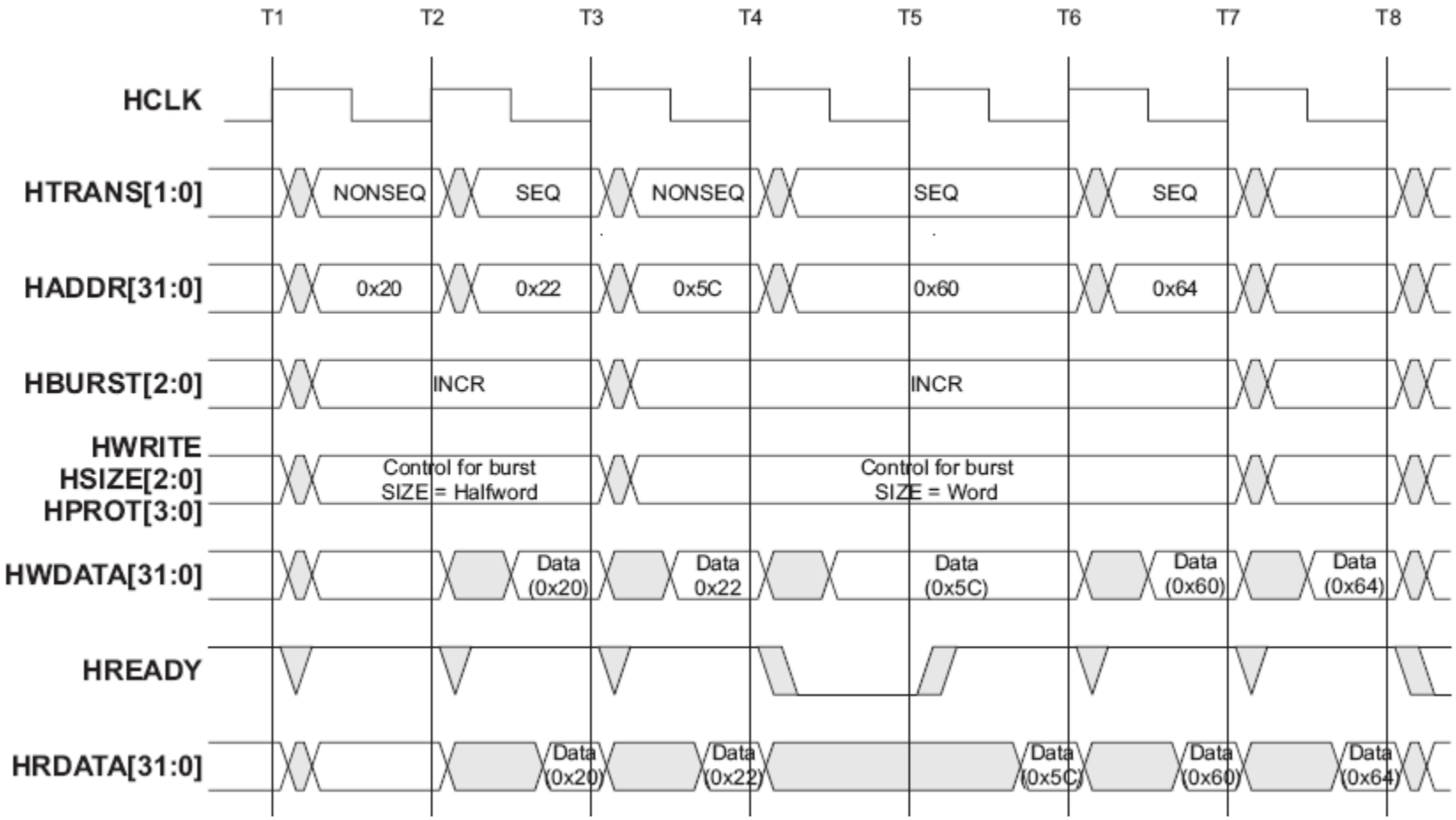
Burst Signal Encoding

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

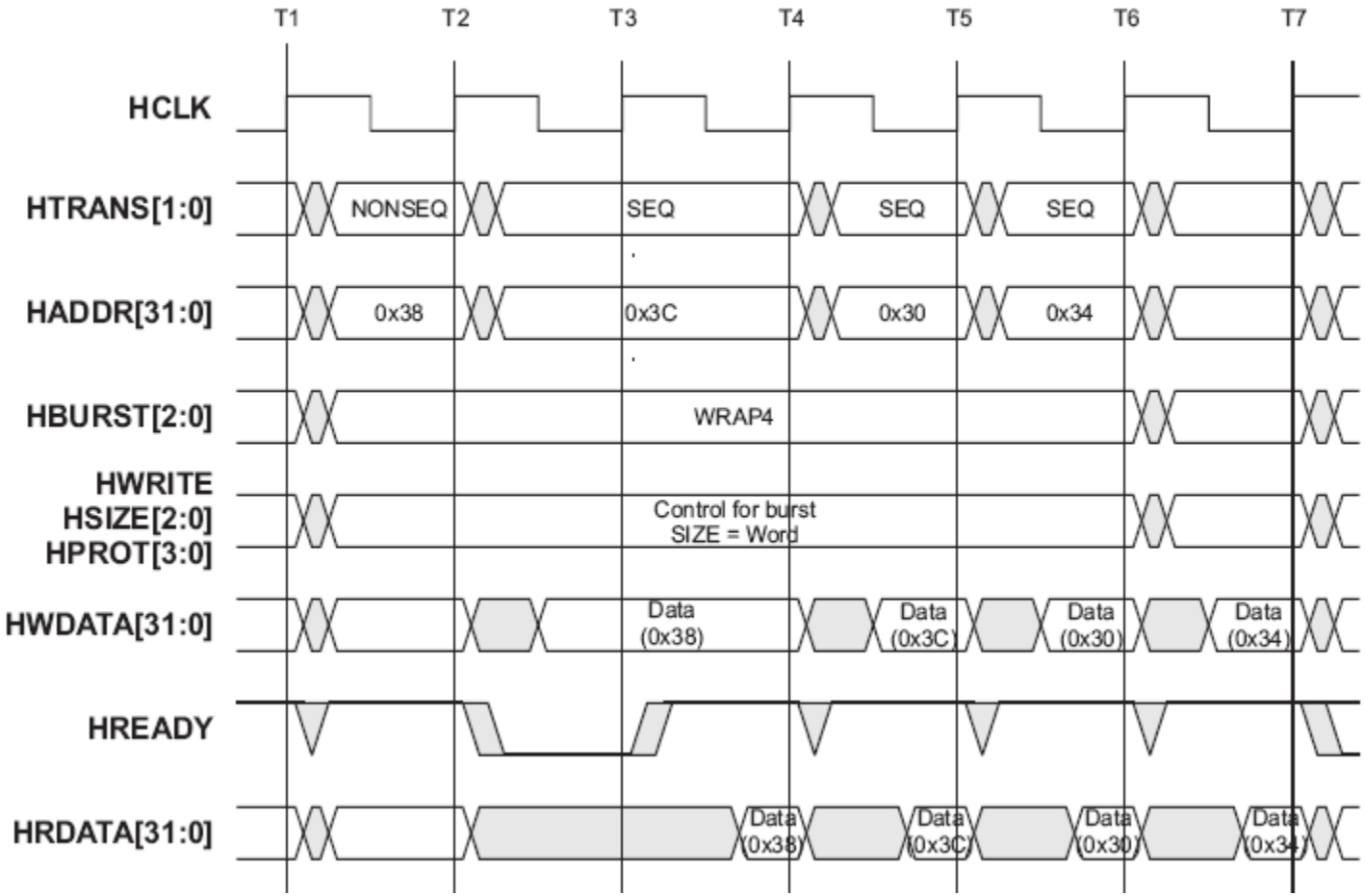
Four-beat incrementing burst



Undefined-length bursts

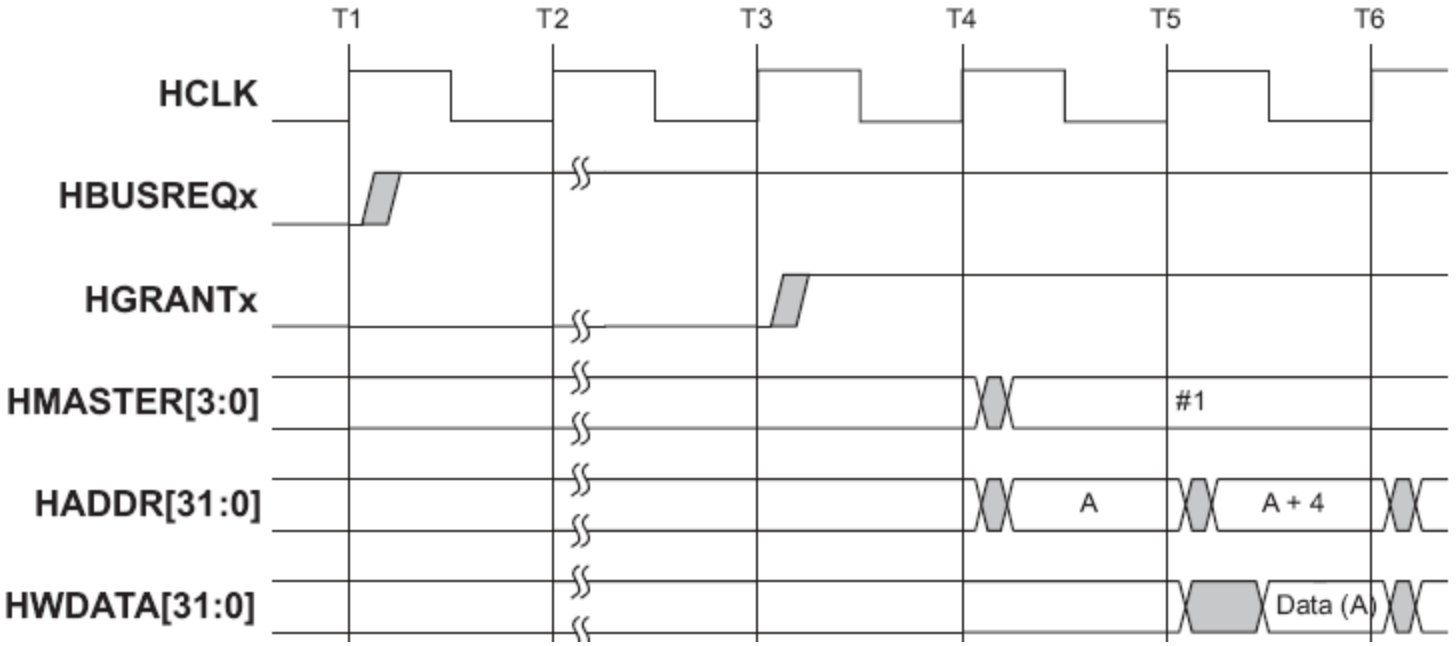


Four-beat wrapping burst

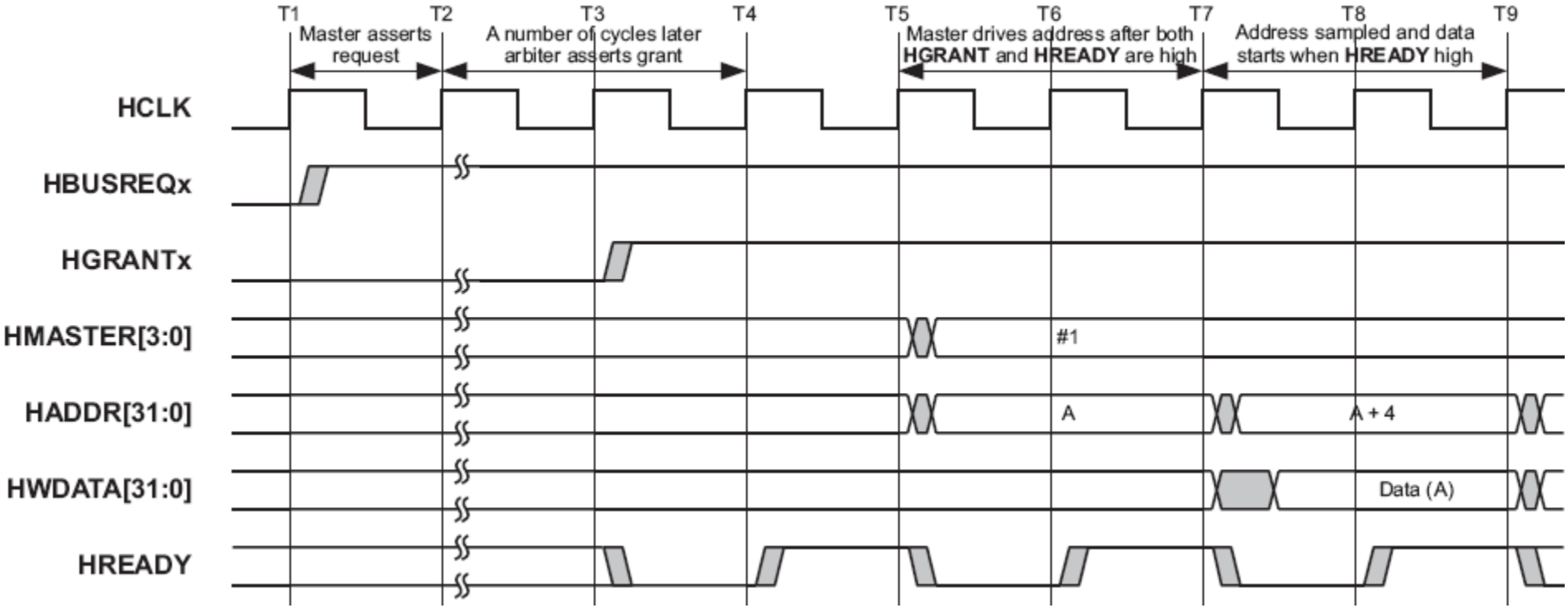


Arbitration

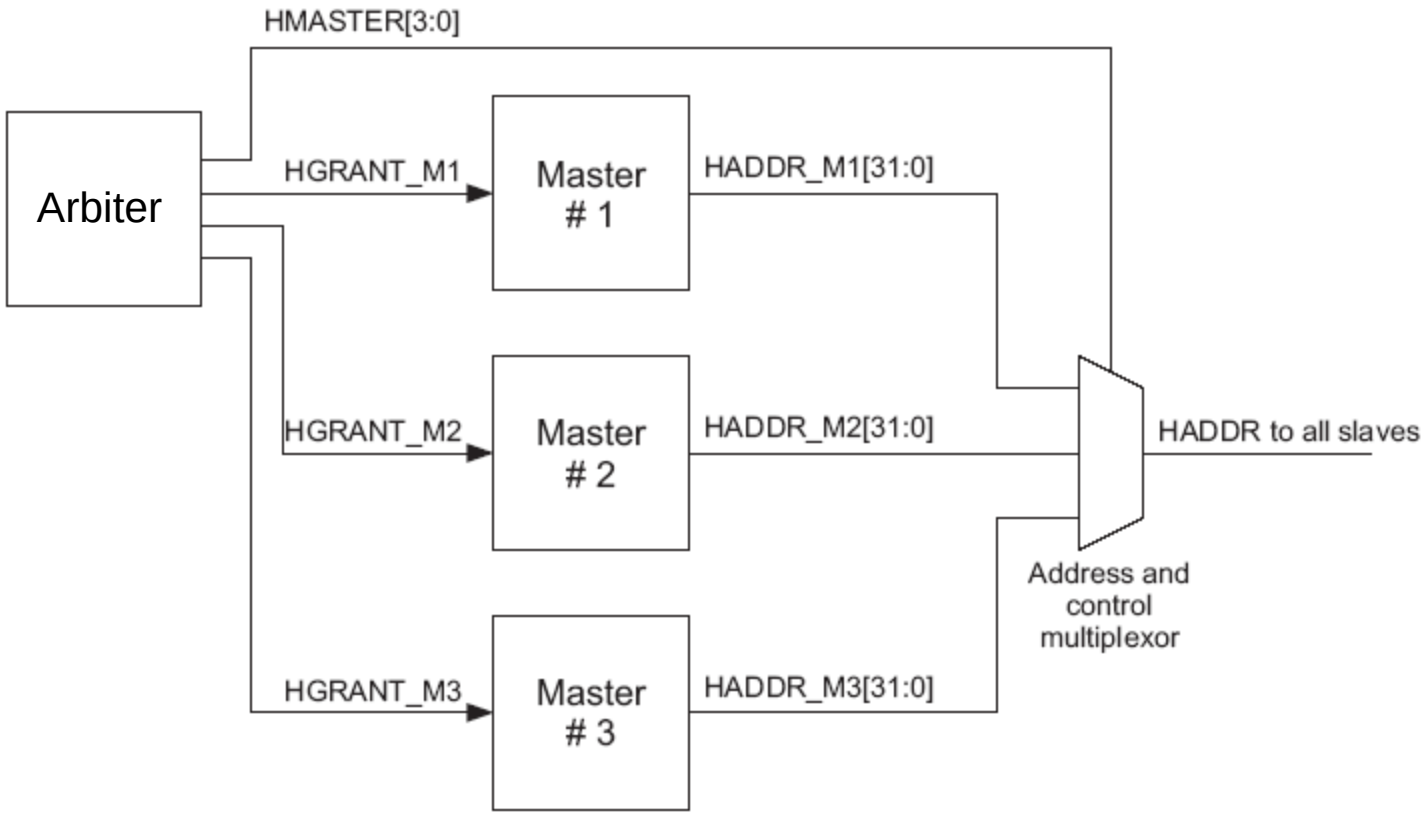
- The *arbitration* mechanism is used to ensure that only one master has access to the bus at any one time
- When a master is granted the bus and is performing a fixed length burst it is not necessary to continue to request the bus in order to complete the burst



Granting access with wait states



Bus master grant signals



Split Transfers

- SPLIT transfers improve the overall utilization of the bus
  - ➔ Separating (or splitting) the operation of the master providing the address to a slave from the operation of the slave responding with the appropriate data

Split Transfers

- When a transfer occurs the slave can decide to issue a SPLIT response if it believes the transfer will take long time
- This signals to the arbiter that the master which is attempting the transfer should not be granted access to the bus until the slave indicates it is ready to complete the transfer
- The arbiter is responsible for observing the response signals and internally masking any requests from masters which have been SPLIT

Split Transfer Sequence

- **1.** The master starts the transfer in an identical way to any other transfer and issues address and control information
- **2.** If the slave is able to provide data immediately it may do so. If the slave decides that it may take a number of cycles to obtain the data it gives a SPLIT transfer response
- **3.** The arbiter grants other masters use of the bus

Split Transfer Sequence

- **4.** When the slave is ready to complete the transfer it asserts the appropriate bit of the HSPLITx bus to the arbiter to indicate which master should be regranted access to the bus

- **5.** The arbiter observes the HSPLITx signals on every cycle, and when any bit of HSPLITx is asserted the arbiter restores the priority of the appropriate master

- **6.** Eventually the arbiter will grant the master so it can re-attempt the transfer. This may not occur immediately if a higher priority master is using the bus

- **7.** When the transfer eventually takes place the slave finishes with an OKAY transfer response

Handover after split transfer