ON-CHIP COMMUNICATION STRUCTURE CONCEPTS

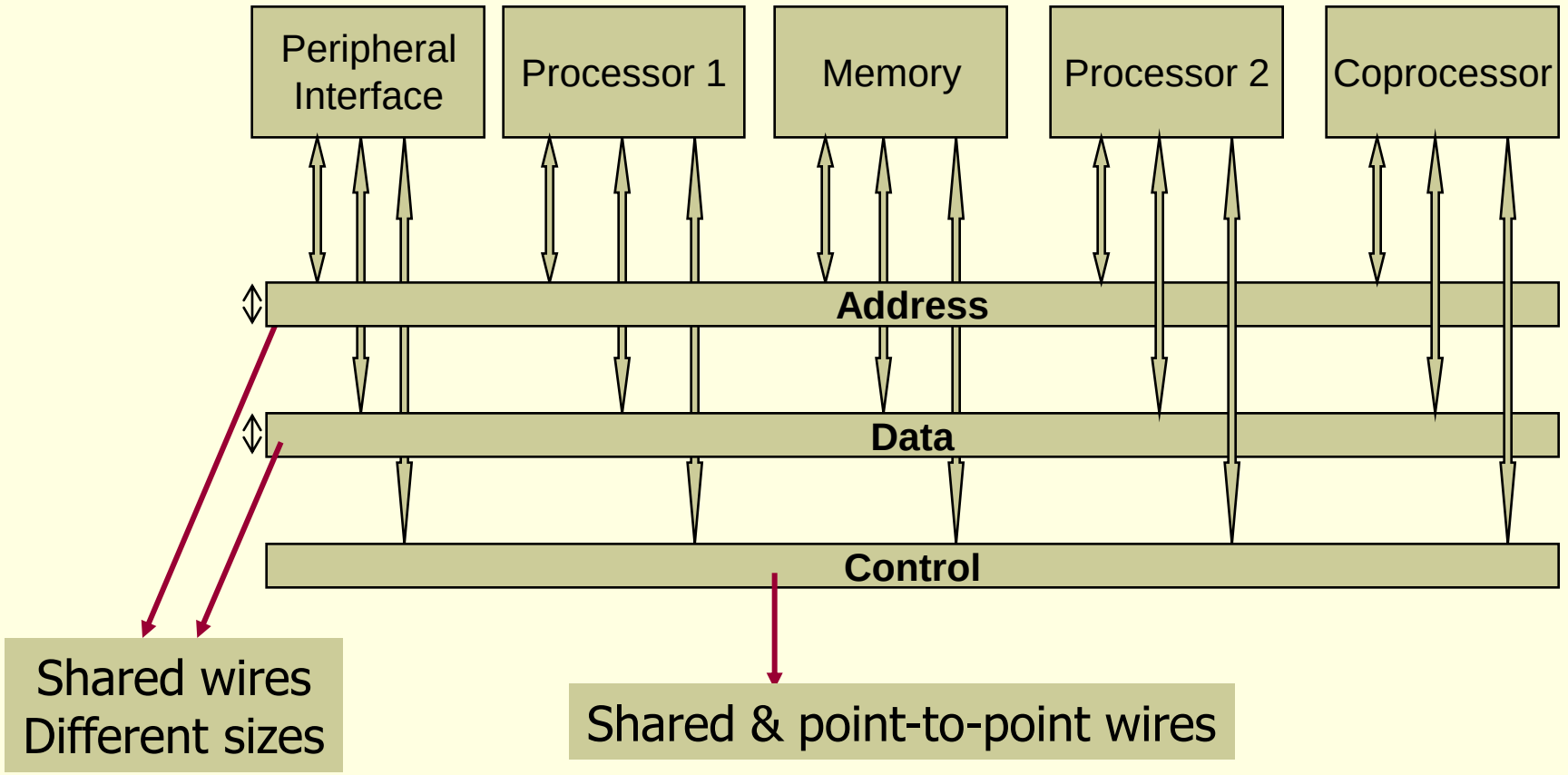
Prof. Sebastian Eslava Ph.D.

Universidad Nacional de Colombia

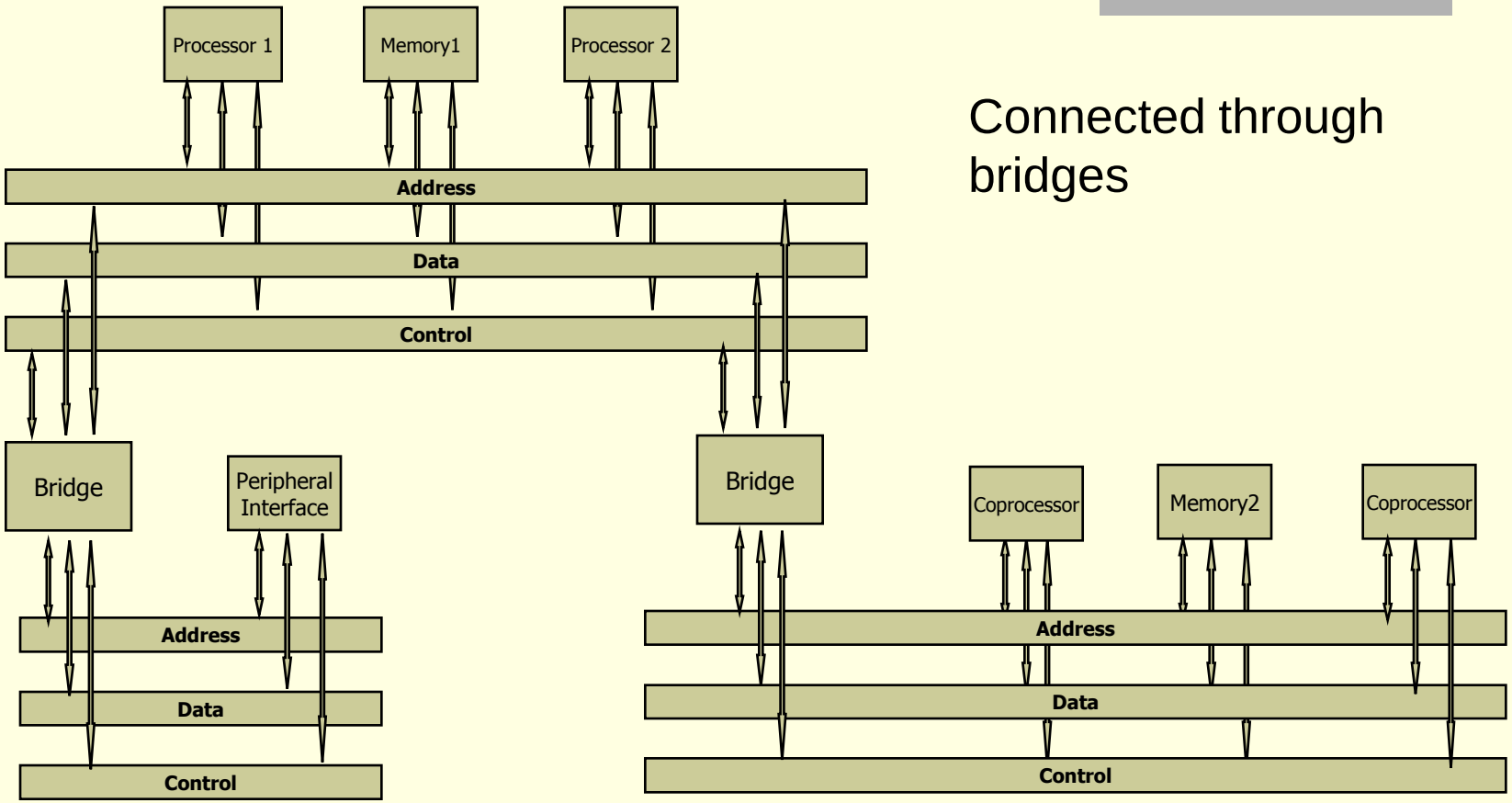
Bus: definition

- Set of *wires* connected to the *bus elements*
- Wires are functionally grouped
  - *Address*
  - *Data*
  - *Control*
- Wires are shared
  - Access protocols are required
  - (Control logic + control wires) embody the protocol

High-level bus view



High-level hierarchical bus view



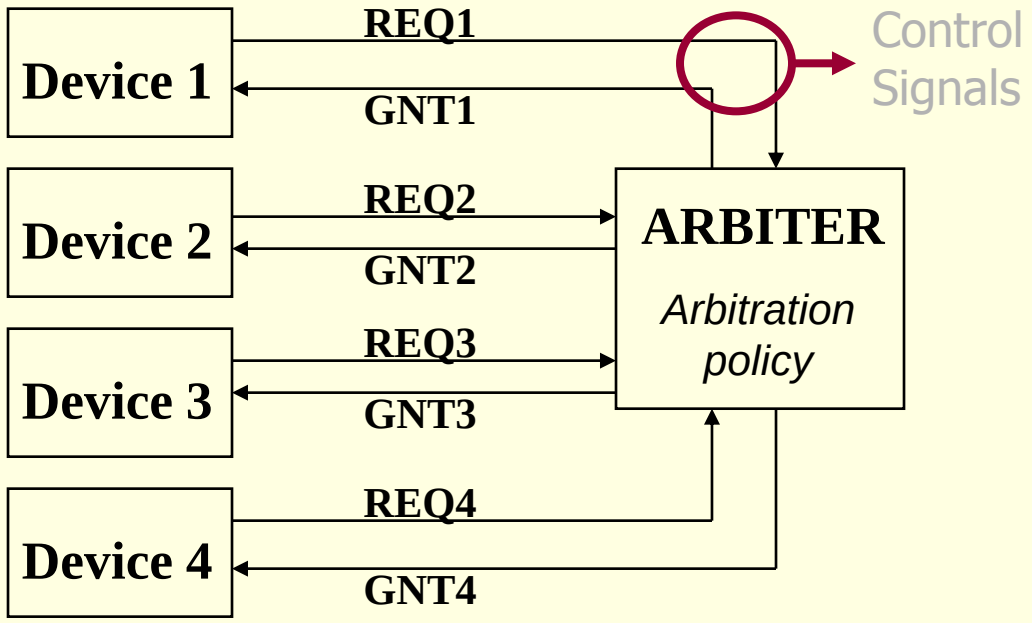
Connected through bridges

High/ low speed types

Bus Arbitration

- Buses can support multiple initiators (master)
- Conflicts resolution requires:
  - *Bus arbitration*
- That needs a decision procedure:
  - *Arbitration policy (choose)*

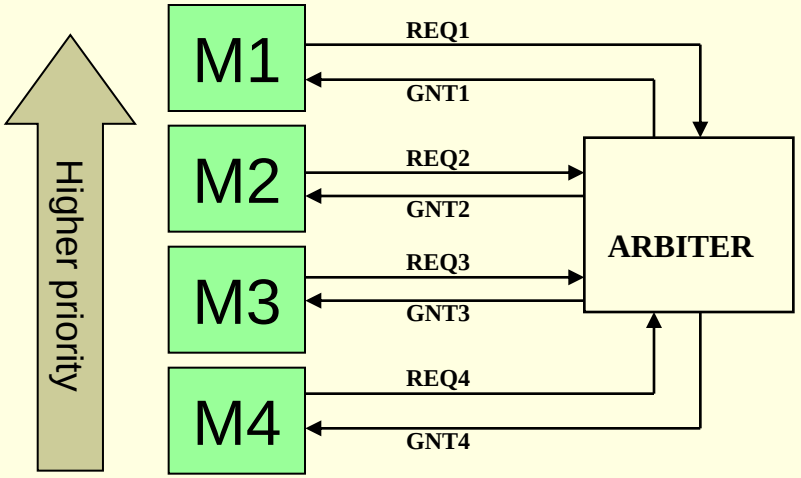
Bus Arbitration Example



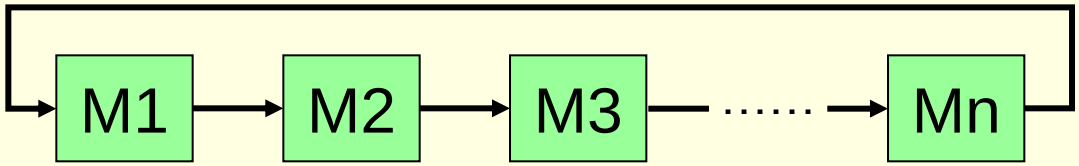
- Each bus master has a pair of dedicated REQ and GNT signals
- Needs at least 1 p2p control signal for (REQi) and (GNTi)

Arbitration policy and configuration

- Fixed priority
  - Master's identification

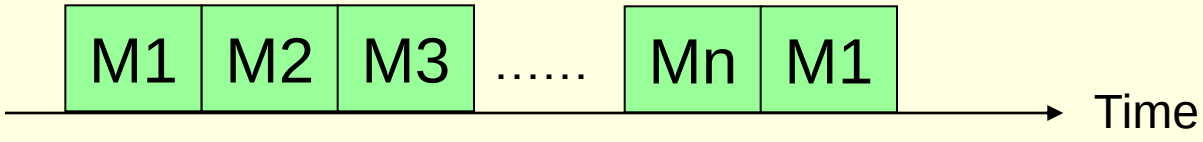


- Round-Robin (RR)
  - Master's sequence

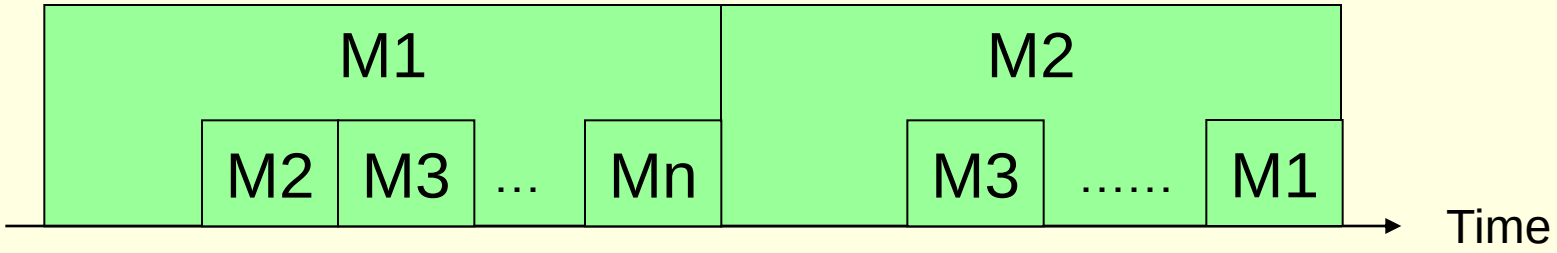


Arbitration policy and configuration (cont.d)

- TDMA (time division multiple access)
  - Master's sequence
  - Time slots



- Two level (RR + TDMA)
  - Master's sequence
  - Time slots

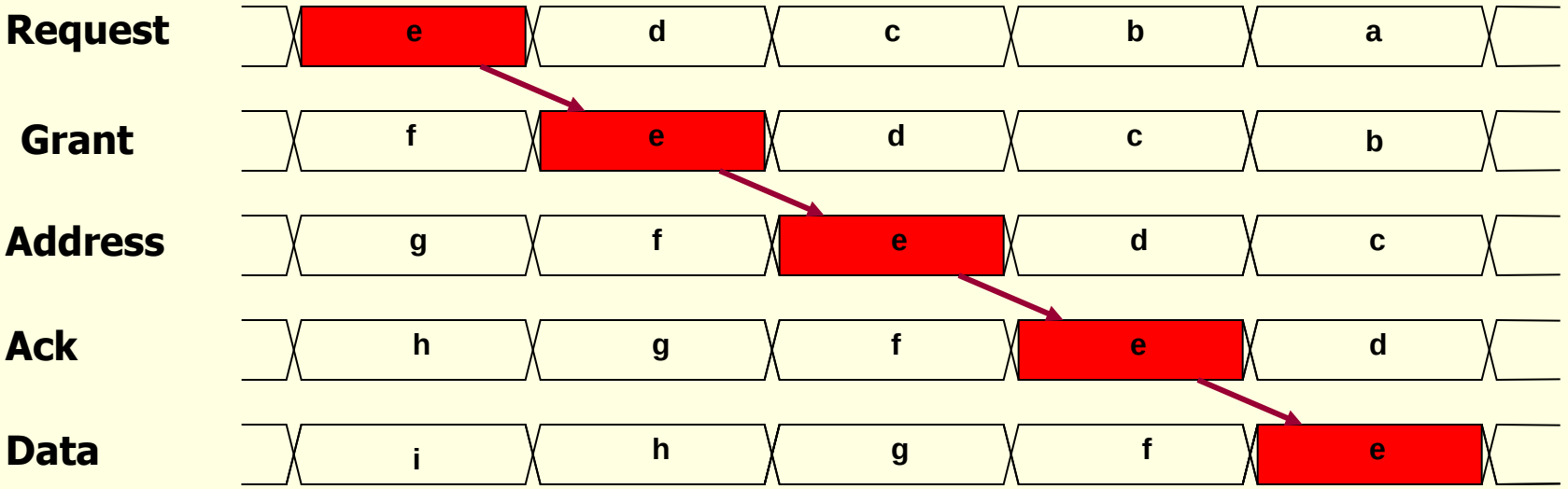


Bus attributes

- **Latency**
  - Time required to initiate a transaction on the bus
- **Maximum Bandwidth**
  - Maximum capacity for data transfer in bit/s
- **Effective Bandwidth**
  - Accounts for contention
- **Energy per bit**
  - Average energy for transferring one bit of information
- **Pipelining**
  - Splitting of a bus transaction over multiple clock cycles
- **Endianness**
  - Determines how bytes are ordered in a word

Pipelining Bus Transactions

Example of a 5-stage pipeline

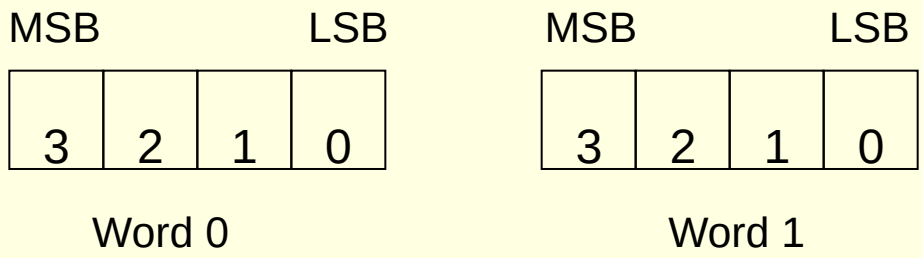


- 5 transactions
- Increase throughput but increase complexity
- AMBA 3-stage: request, address, data

Endianness

- Big endian

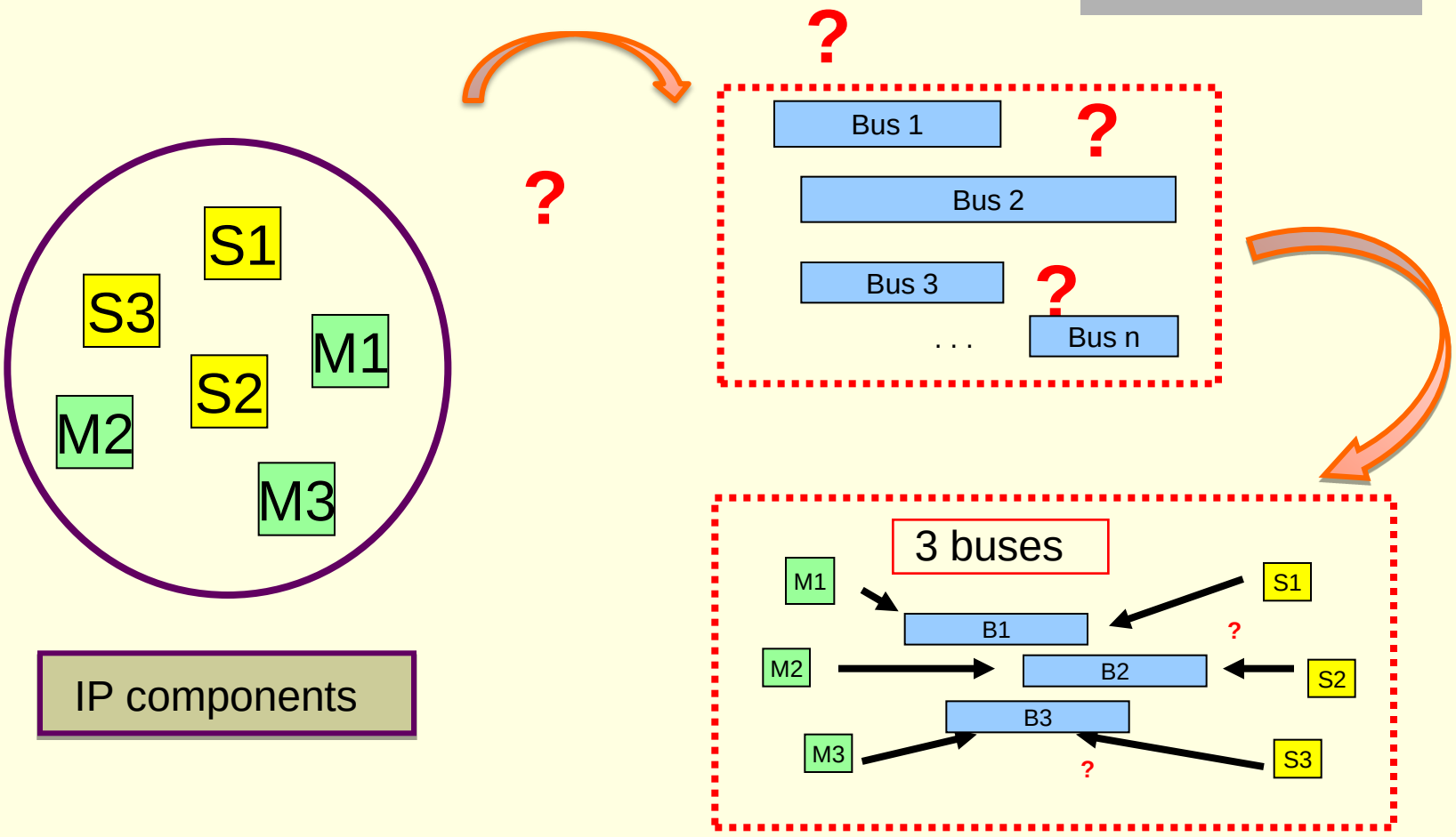
- Little endian



Bus Configuration Parameters

- Global
  - HBUS size
    - # buses
    - Bus_type
  - IP mapping
    - IP/bus/bus_type
    - IP/router
- Local

Hbus Size and Mapping



Bus local parameters

- Logical parameters
  - Arbitration policy (fixed priority, round-robin, TDMA, two-level)
  - Arbitration configuration (master´s id, time-slots)
  - Bridge properties (identification, priority)
  - Protocol interfaces
    - Burstiness (incremental, wrapper)
    - Split transfer (yes, no)
- Physical parameters
  - Bus Width (16,32,64,128,256)
  - Bus Type (high, low throughput, control)
    - Pipeline stages (5,3,1)
  - Bridge Type, width

Bridge properties

- # bridges
  - # buses
- Type
  - High speed – High speed
  - High speed – Low speed
- Bridge width
  - Similar (32-32 bits)
  - Variable (32-16 bits)

Bus type and bus width

Bus type

- High speed
  - AHB, AXI – AMBA
  - PLB – CoreConnect
- Low speed
  - APB – AMBA
  - OCB – CoreConnect
- Control
  - DCR – CoreConnect

Bus width

- Standard
  - 16
  - 32
  - 64
  - 128
  - 256
- Custom

Protocol interfaces

- Burstiness
  - Wrapper
  - Incremental
  - Different sizes
- Split Transfers
  - Used when the slave cannot complete the transfer
  - Bus can be re-arbitered
  - Any master can access the bus.
  - Arbiter must know when the slave is ready to terminate
- Retry
  - Only higher-priority masters can access the bus

Bus Configuration Parameters

Protocol Interfaces:
burstiness, split transfers

Arbitration policy

Arbitration configuration

Bus width

Bus type
(Pipeline stages)

bridges
Types

HBUS size
- # buses

IP mapping
- IP/bus/bus_type

BUS PERFORMANCE METRICS

Bus performance metrics

- Latency
  - Time required to initiate a transaction on the bus
- Average Duration
  - Time required to execute a transaction on the bus

Bus performance metrics (cont.d)

- Throughput
  - **Maximum Bandwidth**
    - Maximum capacity for data transfer in bit/time
  - **Effective Bandwidth**
    - Accounts for contention
  - **Data**

Bus performance metrics (cont.d)

- Utilization Level
  - Resource utilization
- Elements participation
  - On the utilization level
- Communication Locality
  - Partial traffic between pairs

BUS COMMERCIAL ARCHITECTURES

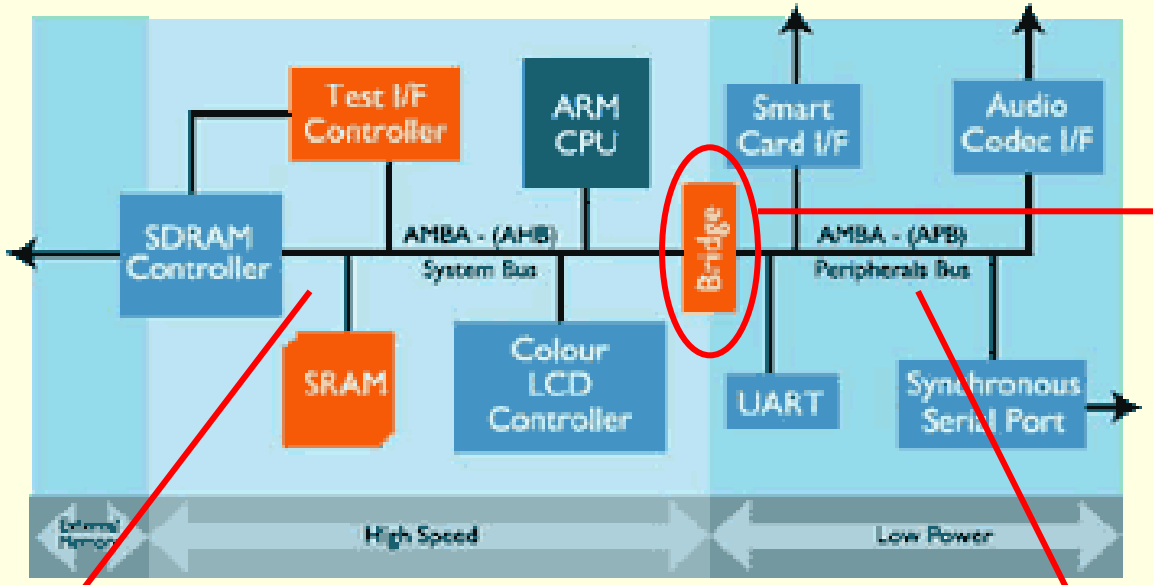
Standard “Bus” Architecture

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

AMBA EXAMPLE

AMBA bus

Amba is a bridged bus



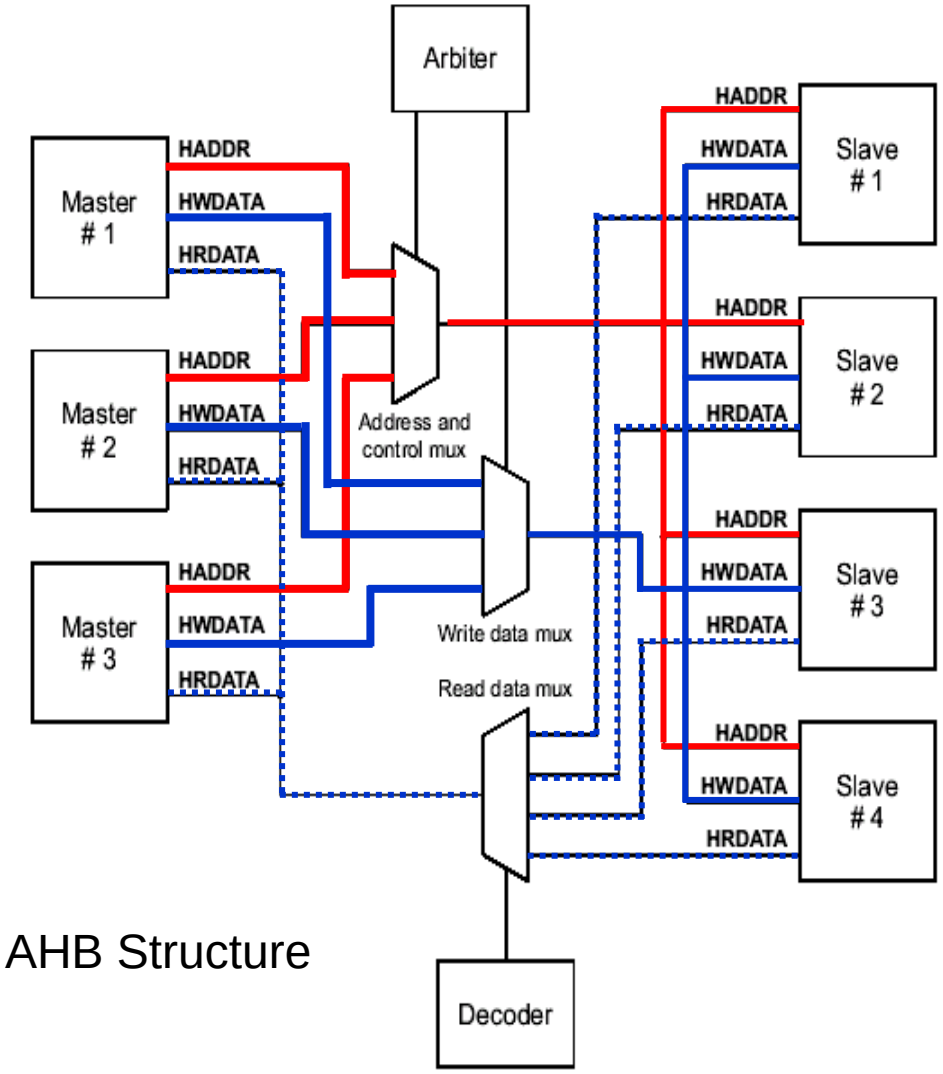
System-Peripheral Bridge

AHB: *high-speed high-bandwidth multi-master bus*

APB: *Simplified processor for general purpose peripherals*

AMBA Bus (cnt.d)

- Signal/Wires:
  - **Address**
  - **Data**
  - Control
- Shared Elements
  - Contention
  - Arbiter
  - Decoder



AHB Structure

AMBA AHB vs. APB

AHB

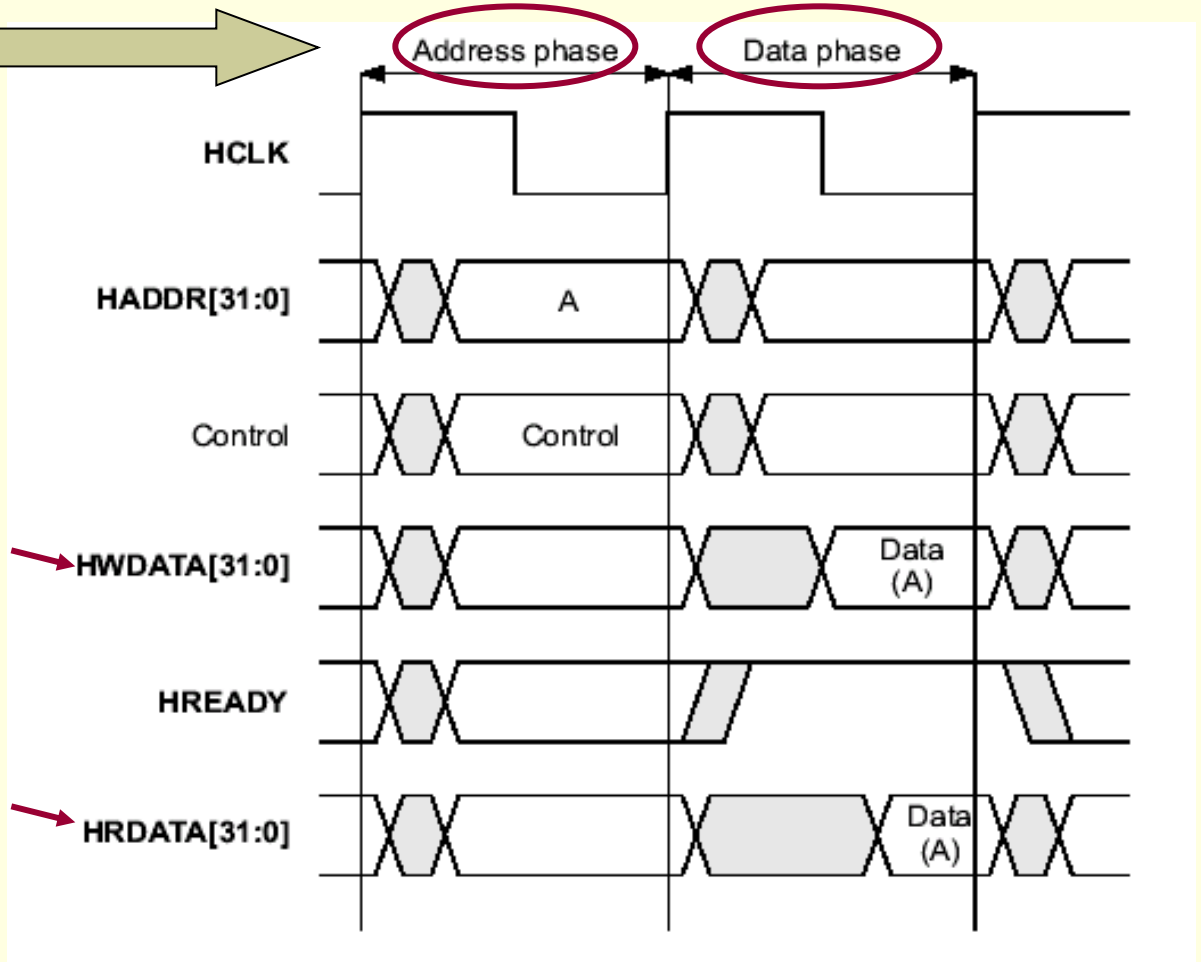
- High Performance
- Multiple Bus Masters
- Pipelined Operation
- Burst Transfers
- Split Transactions
- Used for:
  - Processors, on-chip memory, interface to off-chip memory

APB

- Slave-only bus
- Not pipelined
- Low power interface
- Used for:
  - Register-mapped slaves
    Narrow-bus peripherals

AMBA basic transfer

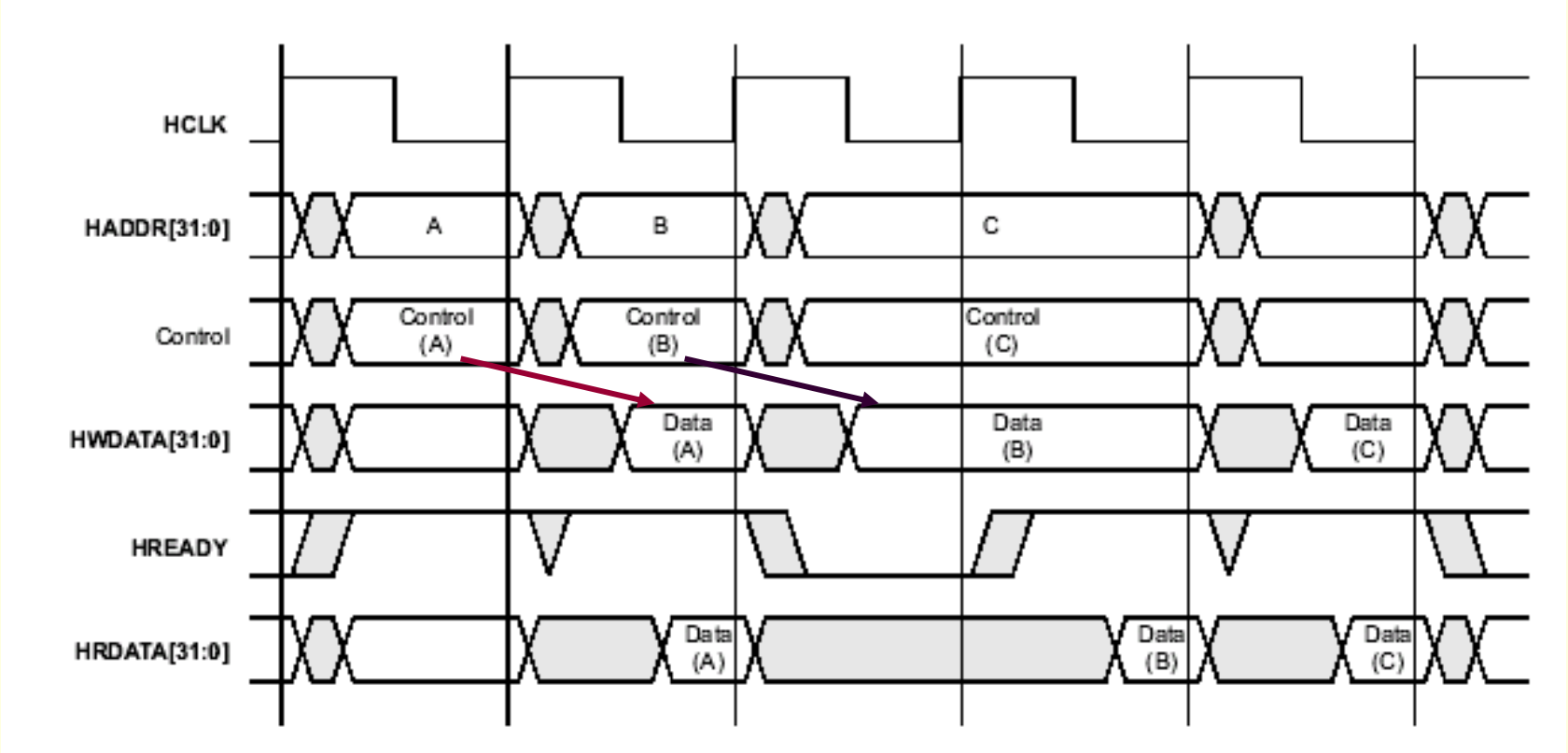
Pipeline stages
- Request
- Address
- Data



For a write

For a read

Multiple transfers



Pipelined (2 CLK)

Burst types

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

Fixed length bursts

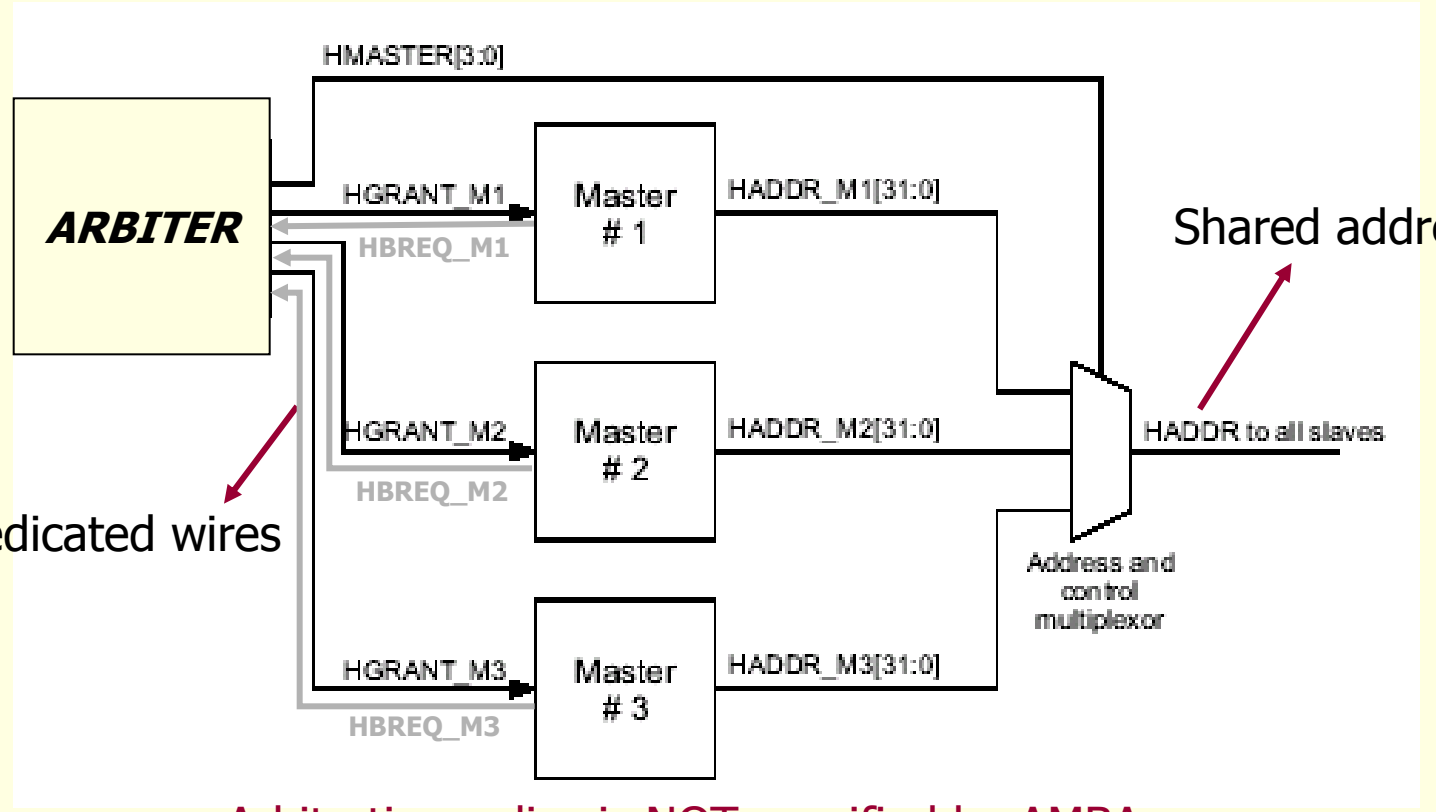
Slave responses

HRESP[1]	HRESP[0]	Response	Description
0	0	OKAY	When **HREADY** is HIGH this shows the transfer has completed successfully. The OKAY response is also used for any additional cycles that are inserted, with **HREADY** LOW, prior to giving one of the three other responses.
0	1	ERROR	This response shows an error has occurred. The error condition should be signalled to the bus master so that it is aware the transfer has been unsuccessful. A two-cycle response is required for an error condition.
1	0	RETRY	The RETRY response shows the transfer has not yet completed, so the bus master should retry the transfer. The master should continue to retry the transfer until it completes. A two-cycle RETRY response is required.
1	1	SPLIT	The transfer has not yet completed successfully. The bus master must retry the transfer when it is next granted access to the bus. The slave will request access to the bus on behalf of the master when the transfer can complete. A two-cycle SPLIT response is required.

Not yet done, the master should retry

Not yet done, the master releases the bus the slave will request access when ready

Arbitration circuitry



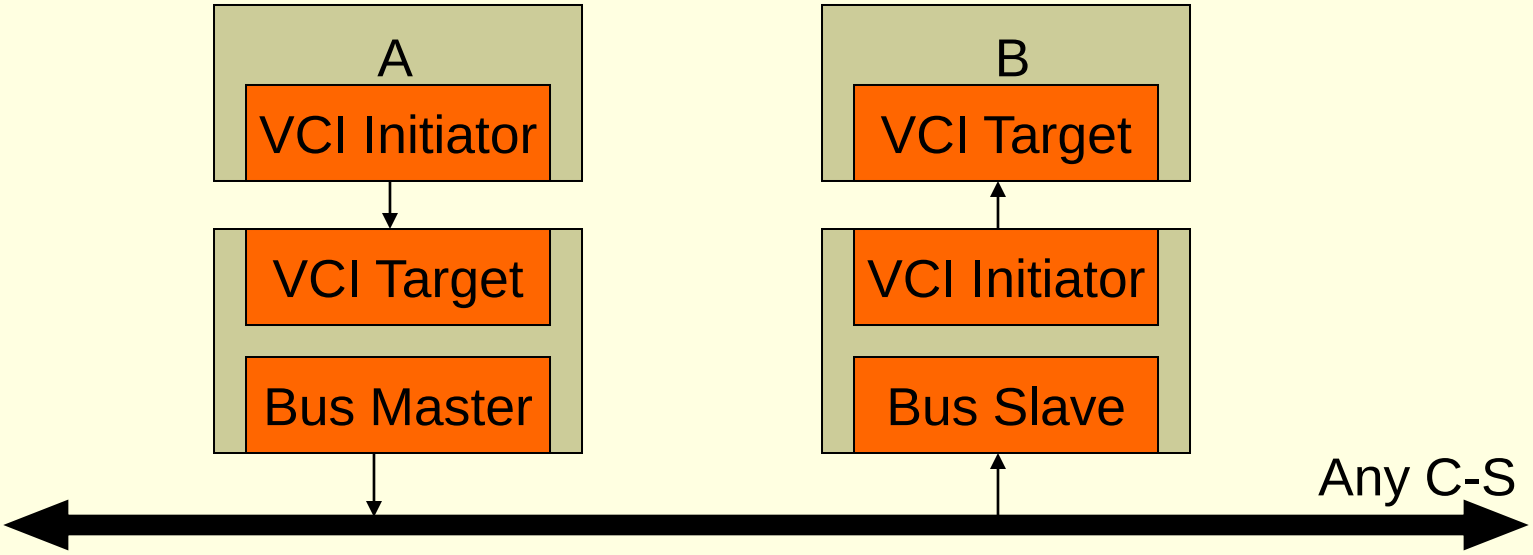
Arbitration policy is NOT specified by AMBA

OCP

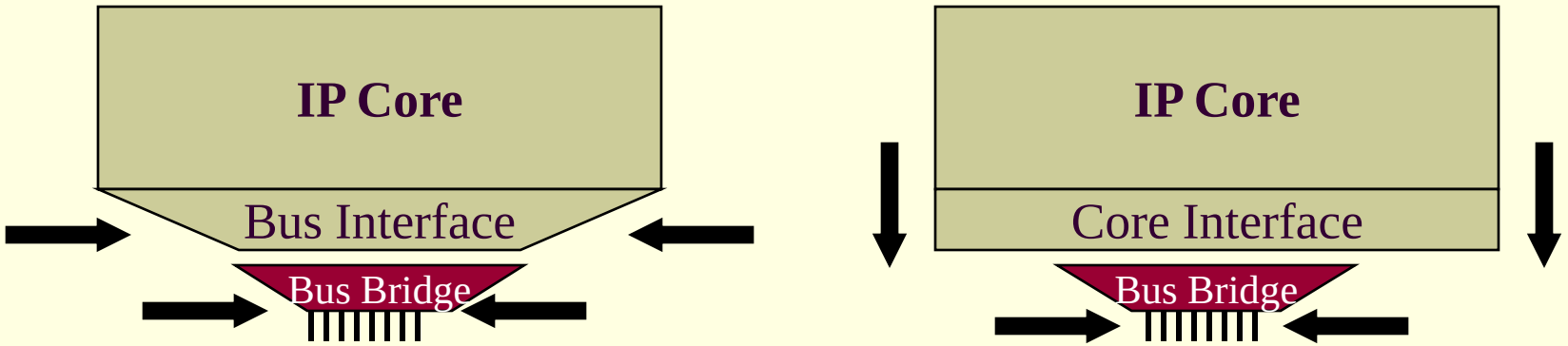
Standardization

- VSIA: Virtual Socket Interface Alliance
  - Specifies Virtual Component Interface
    - Peripheral VCI
    - Basic VCI
    - Advanced VCI

http://www.vsi.org



Bus-Centric vs. Core-Centric



- Bus-Centric Protocol Interface:
  - Forces core interface to the specs of a particular bus
- Core-Centric Protocol Interface:
  - Provides an abstract communication channel for the core
  - Enables unconstrained interface bridge to any interconnect structure

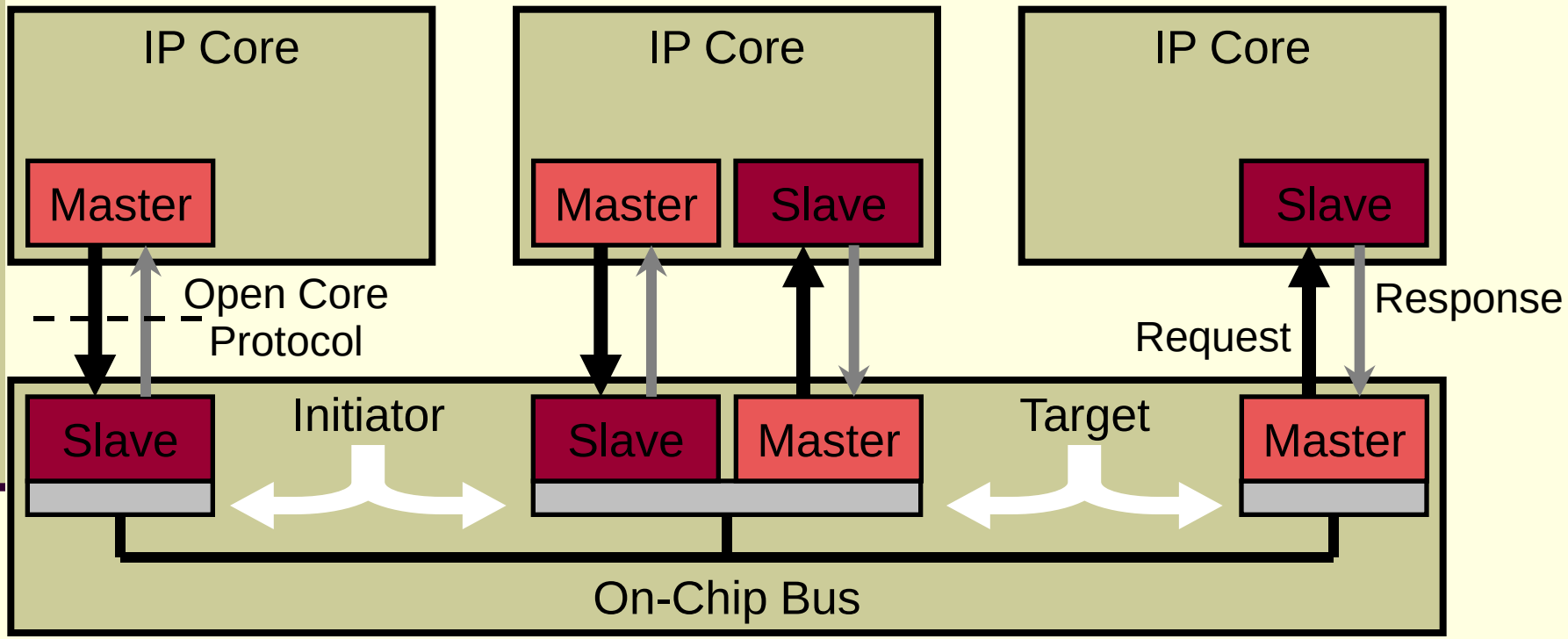
Reference: [SonicsInc]

Basic OCP Concepts

- Point-to-point, unidirectional, synchronous
  - Easy physical implementation
- Master/slave, request/response
  - Well-defined, simple roles
- Extensions
  - Added functionality to support cores with more complex interface requirements
- Configurability
  - Match a core's requirements exactly
  - Tailor design to required features only

Reference: [SonicsInc]

Master vs. Slave



OCP Extensions

- Simple Extensions
  - Byte Enables
  - Bursts
  - Flow Control
  - Data Handshake
- Complex Extensions
  - Threads and Connections
- Sideband Signals

Configurability

- OCP is configurable to tailor to the features required by the core
  - Basic OCP is very simple
  - Many extensions exist for cores with more complex interface requirements
- OCP is configured with a set of parameters
  - Control the presence of a set of signals
    - Example: core makes use of byte enables
  - Control the width of a set of signals
    - Example: address width is 14 bits
  - Control protocol features
    - Example: core uses data handshaking to pipeline write data

Conversion of existing cores

- New cores can be targeted directly at OCP, but existing cores must be converted
- Typical conversion tasks are:
  - Turning a bi-directional data bus into the two uni-directional data buses of OCP
  - Recoding the command field
  - Adding a small state machine to enforce the OCP phases
- OCP configurability eases core conversion
  - Choose features according to core's needs
  - Typical bridges are 3 to 300 gates