SoC Design
Lecture 11: SoC Bus Architectures

Shaahin Hessabi

Department of Computer Engineering

Sharif University of Technology

On-Chip bus topologies

- Shared bus: Several masters and slaves connected to a shared bus. Requires arbiter.
  - \+ simple topology, extensibility, low area cost, easy to build, efficient to implement.
  - – Larger load per data bus line, longer delay for data transfer, larger energy consumption, and lower bandwidth.
- Hierarchical bus: several shared buses interconnected by bridges to form a hierarchy.
  - ➢ SoC components placed at appropriate level in the hierarchy according to the performance level they require.
  - – Transactions across the bridge involve additional overhead,
  - – During the transfer both buses remain inaccessible to other SoC components.
  - \+ Large throughput over the shared buses due to:
    1. decreased load per bus;
    2. potential for parallel transactions on different buses; and multiple communications can be preceded across the bridge in a pipelined manner.
- Ring: each node component (master/slave) communicates using a ring interface.
  - ➢ Usually implemented by a tokenpass protocol.

SoC Bus Architectures

- Current SoCs are advanced enough to need a hierarchy of buses.
- 2 approaches have been proposed:
  1. Companies have promoted their On-Chip-Buses (OCB) as potential standards (ARM, IBM, Palmchip, etc.).
     - These buses allow for higher performance than traditional tri-state buses.
  2. VSIA and Sonics (resp. VCI and OCP) have chosen to develop a standard communication protocol and a bridge to link IPs and the bus.
- Single bus advocates: protocols incur performance and area overheads.
- Bus protocol advocates: no single OCB can address the needs of all SoCs.
- Agreeing upon one all-purpose standard unsuccessful due to:
  - Commercial issues
  - Disagreement over required features
  - Different applications require different trade-offs.

Main Architectures

- Bus approach:
  - ARM AMBA (Advanced Microcontroller Bus Architecture)
  - Altera AVALON
  - IBM CORECONNECT
  - Silicore Corporation's WISHBONE
- Standard communication protocol approach:
  - VCI
  - OCP

AMBA 2.0 Bus Standard

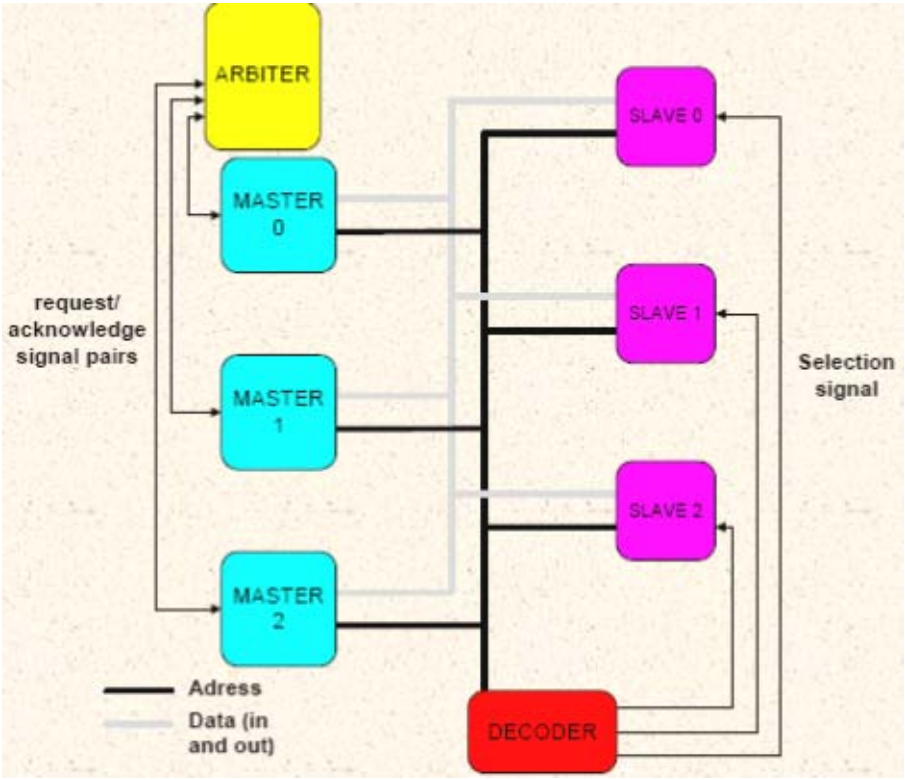
- Three buses are defined:
  - Advanced High-Performance Bus (AHB)
  - Advanced System Bus (ASB)
  - Advanced Peripheral Bus (APB)
- A test methodology is included within AMBA which provides an infrastructure for modular macrocell test and diagnostic access.
- Split-transaction protocols for high performance buses.
  - Bus mastership is released just after a request has completed.
  - Slave has to gain access to the bus to respond, possibly several bus cycles later.
  - Bus supports multiple outstanding transactions.
  - Bus masters and bus interfaces are much more complex.

AMBA Buses: AHB

- Advanced High-speed Bus (new standard)
  - Provides high-bandwidth communication channel between embedded processor (ARM, MIPS, AVR, DSP 320xx, 8051, etc.) and high performance peripherals/ hardware accelerators (ASICs MPEG, color LCD, etc), on-chip SRAM, on-chip external memory interface, and APB bridge.
    - In the Bluetooth SoC, only the processor (ARM7TDM1) is a bus master
  - Data bus width: 32-64-128-256 bit, Address bus width: 32 bit
  - Data bus protocol: Single READ/WRITE transfer, Pipelined, Byte/half-word/word transfer support.
  - Interconnection: multiplexed implementation
  - Supported interconnection: non-tristate, separate data read & write bus required.
- Advantages:
  - Two multiplexed data buses
  - Only uses the rising edge of the clock
  - Burst and split transfers are supported.
- Disadvantage:
  - Area overhead increases rapidly.

AMBA Buses: ASB

- Advanced System Bus (older standard).
  - Data bus width: 32- 64- 128- 256-bit
  - Address bus width: 32 bit
  - Interconnection: not defined
  - Supported interconnection: tristate-bus, common data read & write bus required
- Advantages:
  - Supported by a large number of common ARM processors and microcontrollers
    - ARM7TDMI, ARM940T, ARM9TDMI
  - Relatively simple to implement
  - Burst transfers supported
- Disadvantages:
  - Single, tri-state data bus
  - Latch-based instead of register-based design
  - Uses both clock edges
  - Split transfers not supported

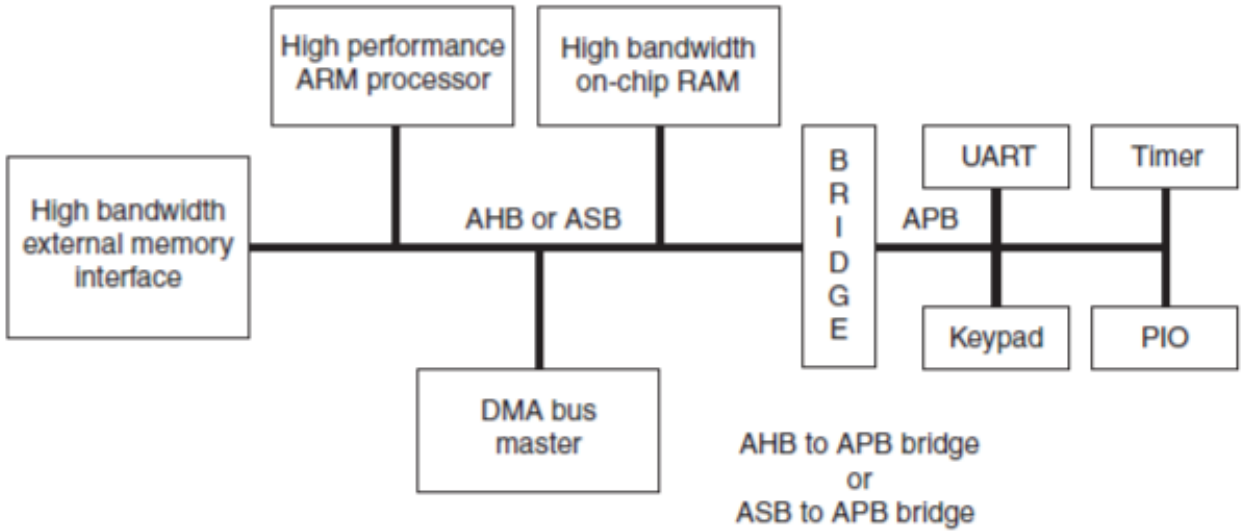


AMBA Buses: APB

- Advanced Peripheral Bus
  - Optimised for minimal power consumption and reduced interface complexity to support peripheral functions
  - data bus width: 8-16-32-bit
  - address bus width: 32 bit
  - architecture (Single) MASTER (bridge) / (Multi) SLAVE
  - data bus protocol: 2 cycle single READ/WRITE transfer, no burst transfer, non-Pipelined
  - Interconnection: not defined
  - Supported interconnection: non-tristate-bus recommended, separate data read & write bus recommended
  - Power consumption: zero, when not in use

System Based on an AMBA Bus

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



AMBA AHB

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

AMBA ASB

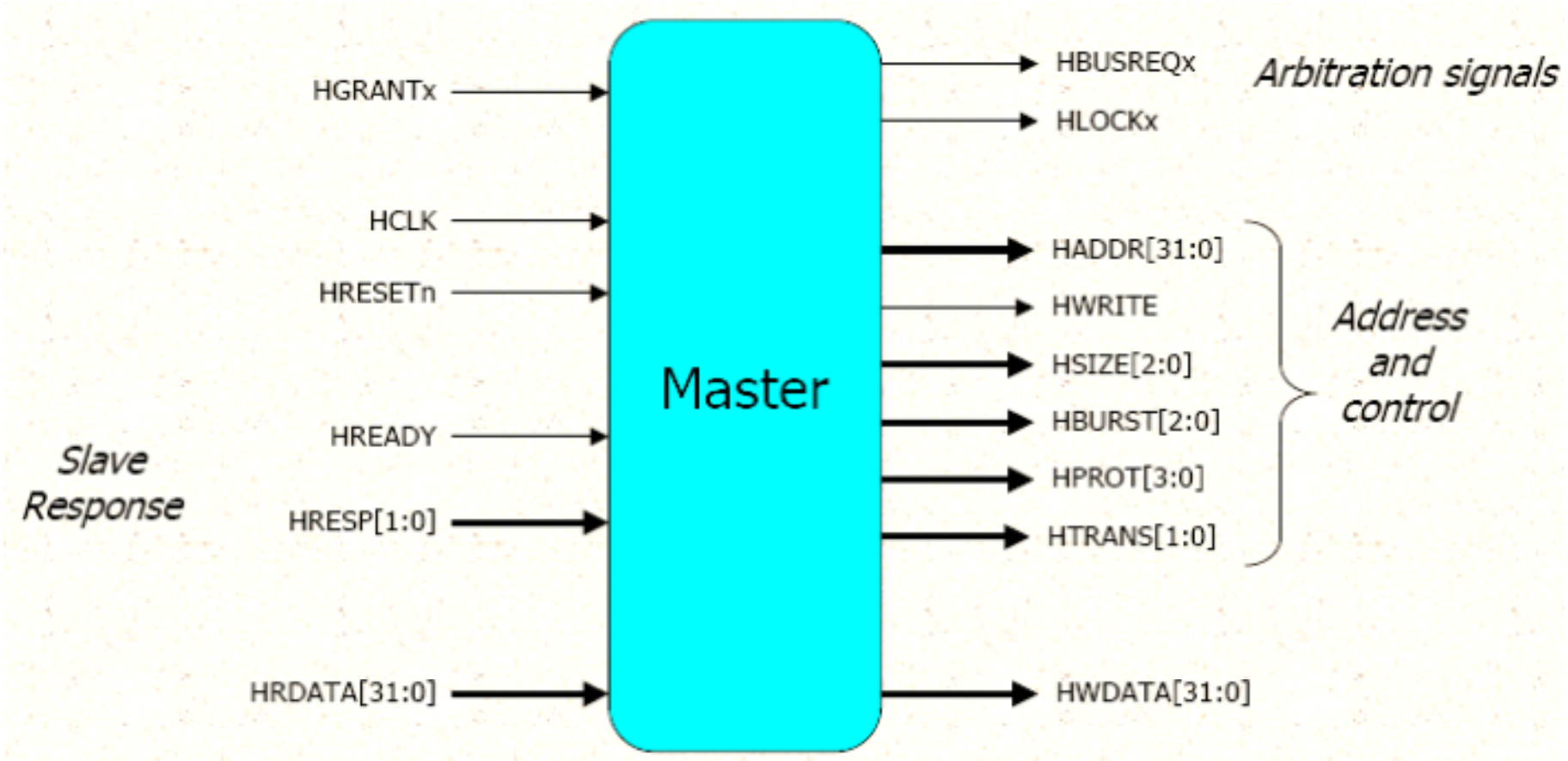
* High performance
* Pipelined operation
* Multiple bus masters

AMBA APB

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

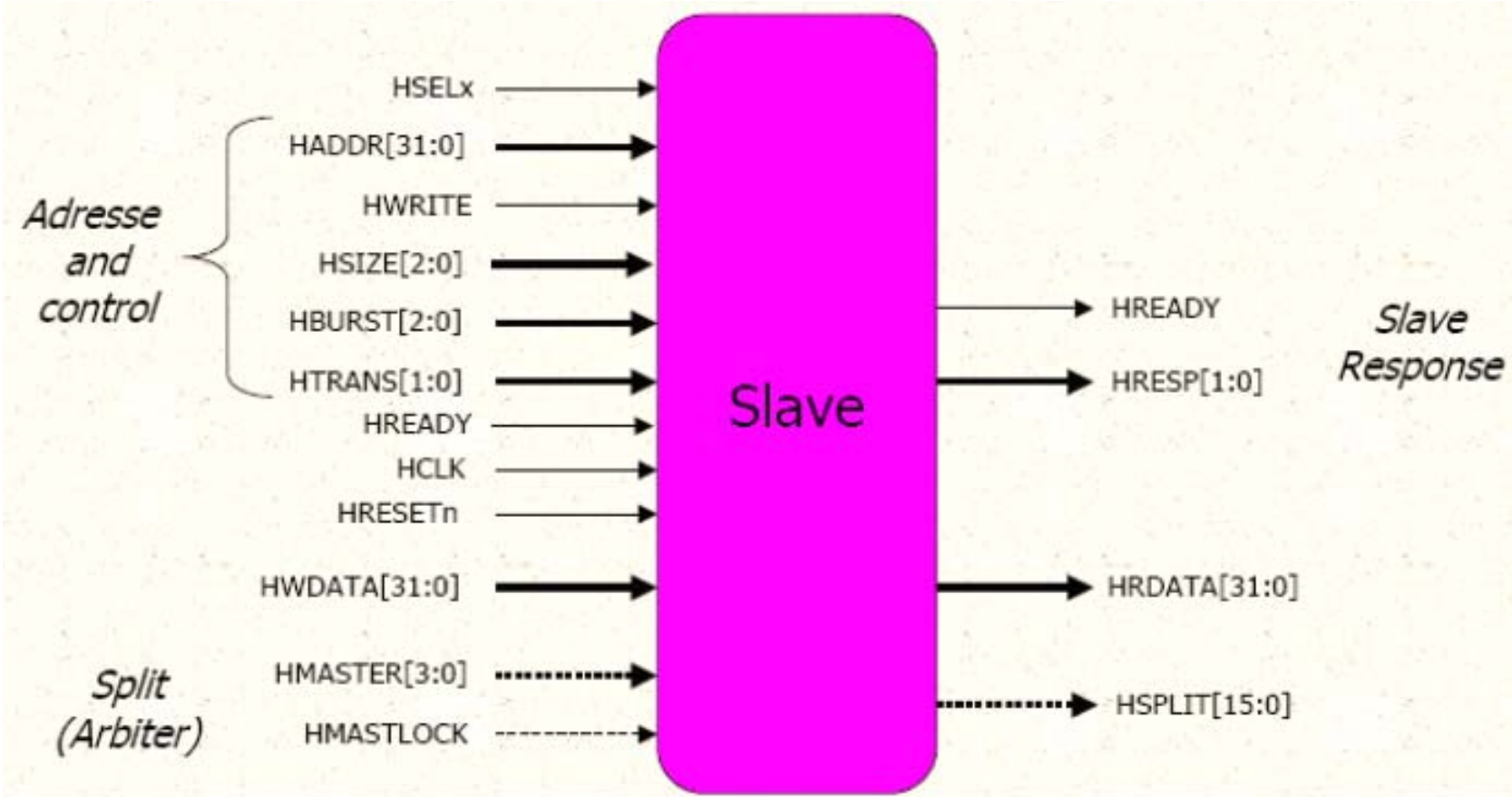
AMBA AHB Master

- Can initiate read and write information by providing address and control information.



AMBA AHB Slave

- Responds to a read and write operation within a given address-space range.
- Signals back to the active bus master the success, failure or waiting of the data transfer.

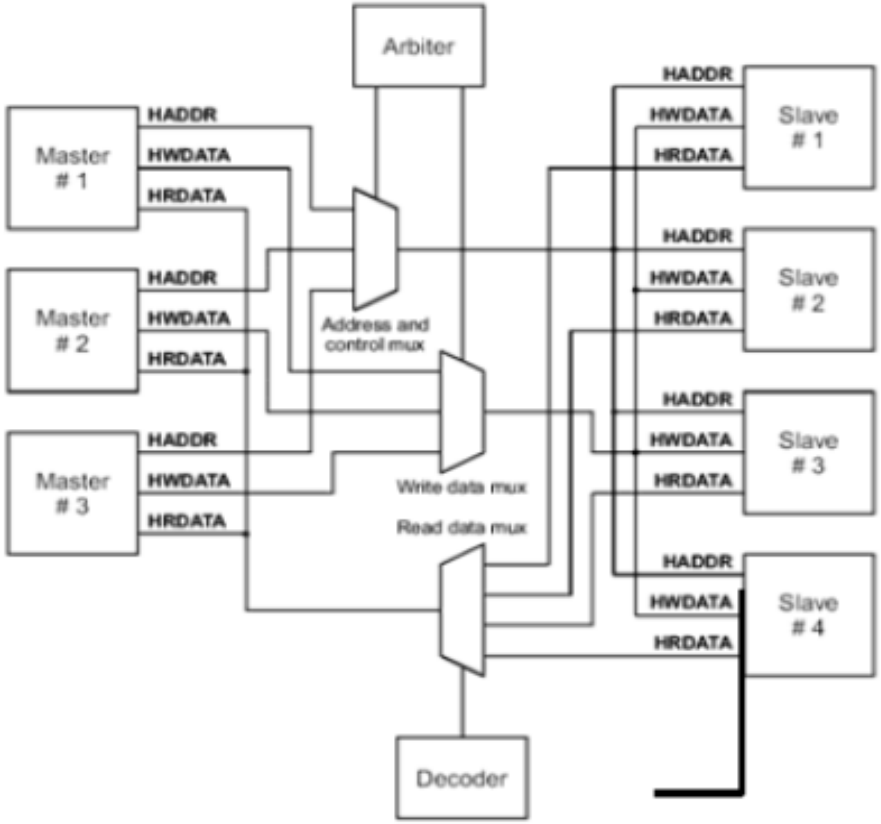


AMBA Arbiter and Decoder

- AHB Arbiter
  - Ensures that only one bus master at a time is allowed to initiate data transfers.
  - Arbitration protocol is fixed, but any arbitration algorithm (highest priority, fair access,…) can be implemented depending on the application requirements.
  - An AHB includes only one arbiter.
- AHB Decoder
  - Decodes the address of each transfer, and provides a select signal for the involved slave.
  - A single centralized decoder is required in all AHB implementations, to provide a select signal, HSELx, for each slave on the bus.
  - A slave must only sample the address and control signals and HSELx when HREADY is HIGH, indicating that the current transfer is completing.

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.

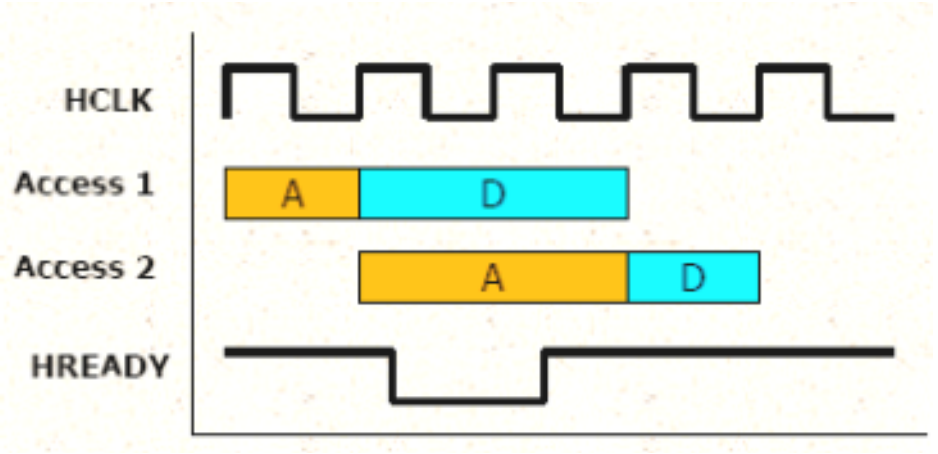


Split Transfers

1. If the slave decides that it may take a large number of cycles to obtain the data, it returns a SPLIT response (instead of OK) to the arbiter and memorizes HMASTER,
2. The arbiter grants the use of the bus to another master that requests it,
3. When the slave is ready to complete the transfer, it asserts the appropriate bit of the HSPLITx (between 0 and 16) bus to the arbiter,
4. The arbiter observes the HSPLITx signals every cycle. It will allow the master to complete the transfer if no higher priority master is using the bus,
5. When the transfer eventually takes place, the slave finishes with an OKAY transfer response (on HRESP).

Pipelined Burst Transfers

- The pipelining of memory accesses allows for a higher transfer rate, at the cost of an initial latency.
- For certain types of memory, the first access in a burst requires several cycles, and subsequent accesses take only one.
- AMBA uses a pipelined memory model.
- AMBA AHB uses a two-phase pipeline:
  - Address Phase: lasts only a single cycle.
  - Data Phase: may require several cycles. This is achieved using the HREADY signal.

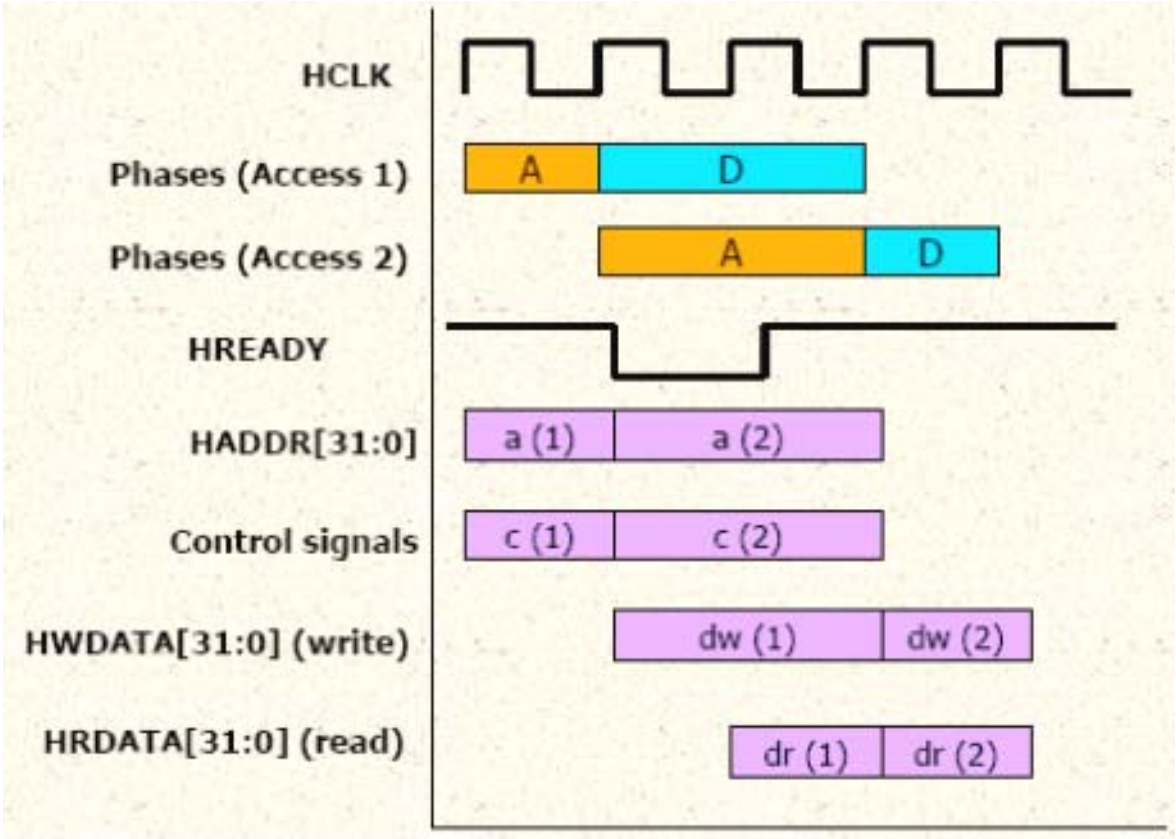


Pipelined Burst Transfers: Address Phase

- During the address phase:
- The master places the address and the other control signals on the bus
- The decoder selects the appropriate slave
- At the next rising edge of the clock, the slave stores the address and control signals
- The data phase begins
- The address phase can be stretched if:
  - The bus is not immediately granted to the master (HGRANT = 0)
  - The preceding data phase is not yet complete (HREADY = 0)

Pipelined Burst Transfers: Data Phase

- During the data phase:
  - The slave reacts to the access, according to the stored control signals.
  - A slow slave can request several cycles by setting HREADY = 0.
  - When HREADY = 1, the access is complete. If the access is a read, the master captures the data at that moment.
- The phases:



AXI: The AMBA 3 Bus Protocol

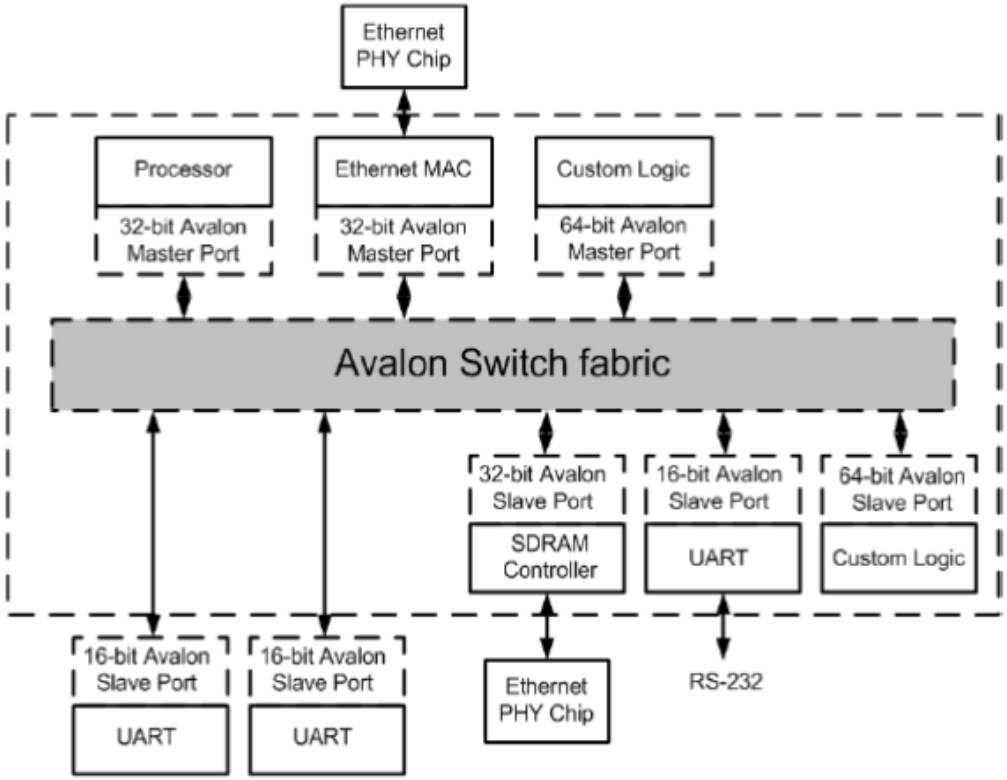
- Based on point-to-point connection concept.
- Objectives:
  - Be suitable for high-bandwidth and low-latency designs.
  - Enable high-frequency operation without using complex bridges.
  - Meet the interface requirements of a wide range of components.
  - Be suitable for memory controllers with high initial access latency.
  - Provide flexibility in the implementation of interconnect architectures.
  - Be backward-compatible with existing AHB and APB interfaces.
- Key features of the AXI protocol:
  - separate address/control and data phases
  - support for unaligned data transfers using byte strobes
  - burst-based transactions with only start address issued
  - separate read and write data channels to enable low-cost DMA
  - ability to issue multiple outstanding addresses
  - out-of-order transaction completion
  - easy addition of register stages to provide timing closure

Altera Avalon Bus

- ALTERA interface bus, used in Nios embedded processor
- Designed to accommodate peripheral development for the system-on-a-programmable-chip (SOPC) environment.
- The generated switch fabric logic includes several chipselect signals for
  - data-path multiplexing, address decoding, wait-state generation, interrupt-priority assignment, dynamic-bus sizing, multi-muster arbitration logic and advanced switch fabric transfer.
- Can only be implemented on Altera devices using SOPC Builder tool
- Avalon Bus is generated automatically, when a new Nios II core with peripherals is created in SOPC-builder

Altera Avalon Bus

- Features
  - Up to 128-bit wide data
  - Synchronous operation
  - Separate address, data and control lines
- Open Standard
  - Specification specifies communication between
    - Master and switch-fabric
    - Slave and switch-fabric
  - Third party vendors can develop their Avalon devices



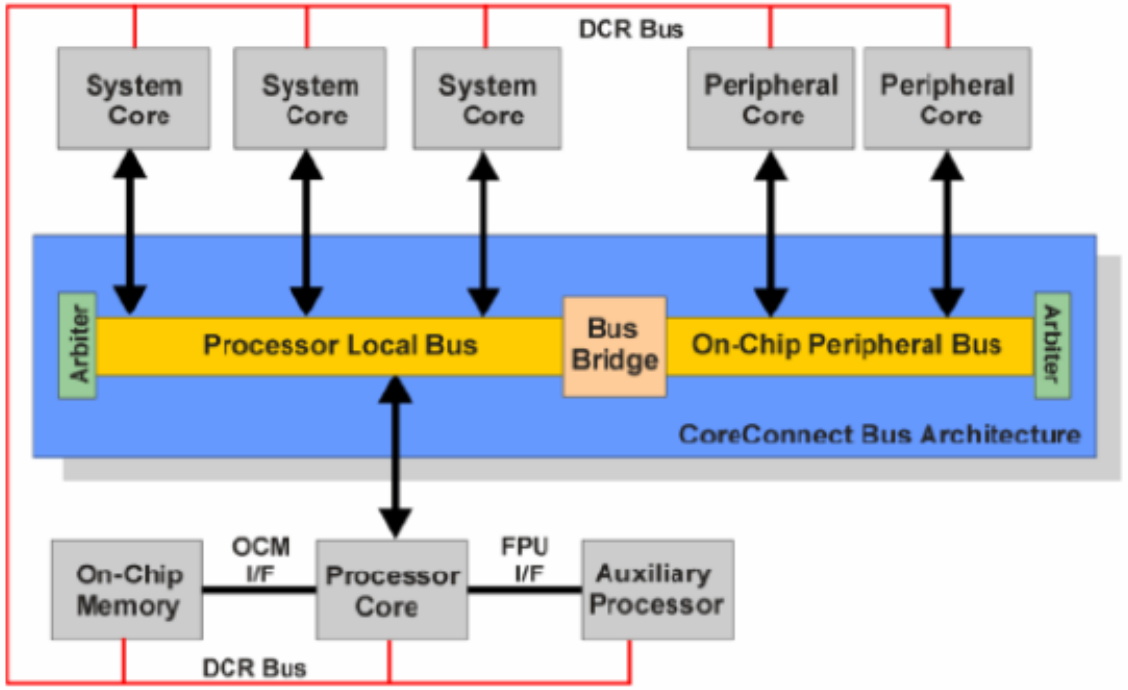
- Avalon specification allows (among others) the following transfer modes:
  - Wait-states: Fixed or variable (slave only)
  - Pipeline: Fixed or variable latency
  - Burst
  - Tristate (devices with a shared read/write channel)

IBM CoreConnect BUS

- Base bus scheme for the IBM Blue Logic Core Library
- Supports standard bus transactions (Burst/DMA/Split/Pipelined)
- Eases the integration and reuse of processor, system and peripheral cores, within standard product platform design
- Arbitration based on a static priority, with programmable priority fairness.
- Three types:
  - Processor local bus (PLB)
    - high-performance bus, with low latency system modules,
    - Separate address, and data buses support concurrent read/write transfers.
  - On-chip peripheral bus (OPB)
    - Optimized to connect to low peripheral and low power consumption
  - Device control register bus (DCR)
    - Interconnects General Purpose Register of CPU, with the Device Control Register of the slaves.
    - Removes the configuration registers from memory map ➔ reduces the loading and improves the bandwidth for Processor Local Bus

Diagram of CoreConnect BUS

- PLB: main system bus.
  - Synchronous,
  - Multi-master (up to 16),
  - No restriction on # of slaves,
  - Central arbitrated.
- OPB:
  - Synchronous,
  - Multi-master,
  - Separate data and address bus.

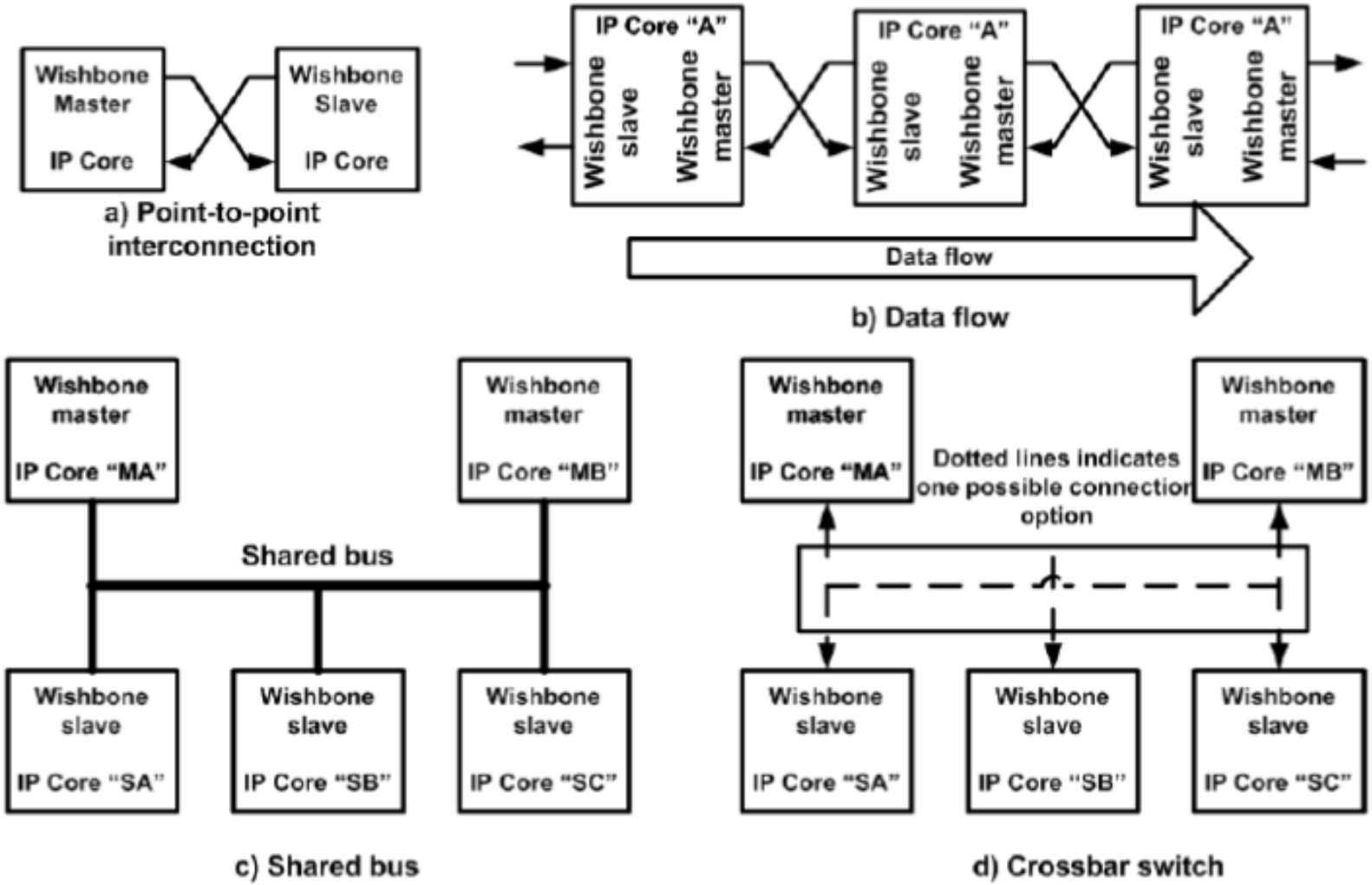


- PLB masters access the peripherals on OPB through OPB bridge macro.
- DCR: ring topology, single-master, relatively low speed datapath for:
  - Passing status and setting configuration information between the Processor Core and others SoC constituents,
  - DFT purposes.

Wishbone Bus

- Silicore Corporation's SoC interconnect architecture for portable IP core.
- Put in public domain by OpenCores in August 2002.
- Improves the portability and reliability of the system by using a common logical interface between IP cores.
- Wishbone specification:
  - Set of rules, recommendations, suggestions, permissions and observations.
  - Allows Wishbone to be a simple, open, highly configurable interface.
  - Defines only one interface for high and low speed bus (not hierarchical).
    - If 2 buses should exist, one slow and one fast, two separated Wishbone interfaces could be created.
  - Users might have to create their own substandard of wishbone, specifying data order, meaning of tags, and additional features.

Possible Wishbone Interconnections

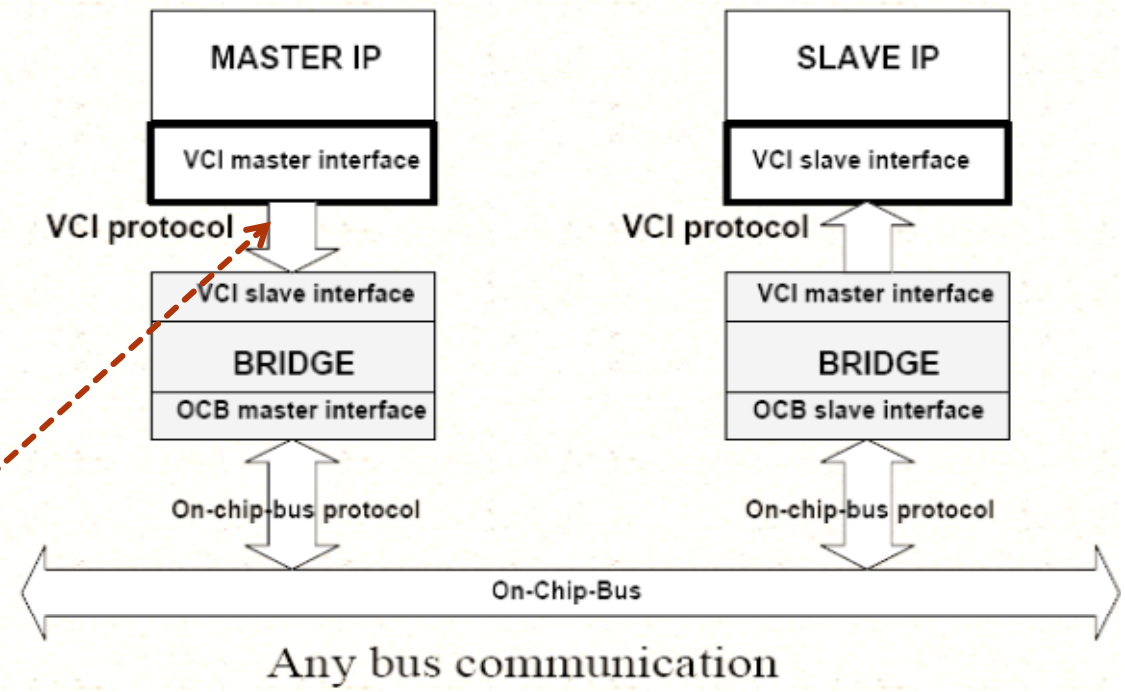


Disadvantages of Standard Bus Protocol Approach

- Agreeing upon one all-purpose interconnection standard has proven to be unsuccessful due to
  - Commercial issues
  - Disagreement over required features
  - Different applications require different trade-offs
- Bus protocols limit design reuse of IP.
- Alternative to standard bus protocol: standard component protocols
  - Virtual Component Interface (VCI) from Virtual Socket Interface Alliance (VSIA)
    - Basically a handshake protocol
  - Open Core Protocol (OCP) from Sonics
    - Bus interface for IP cores
    - Reconfigurable interface

VCI

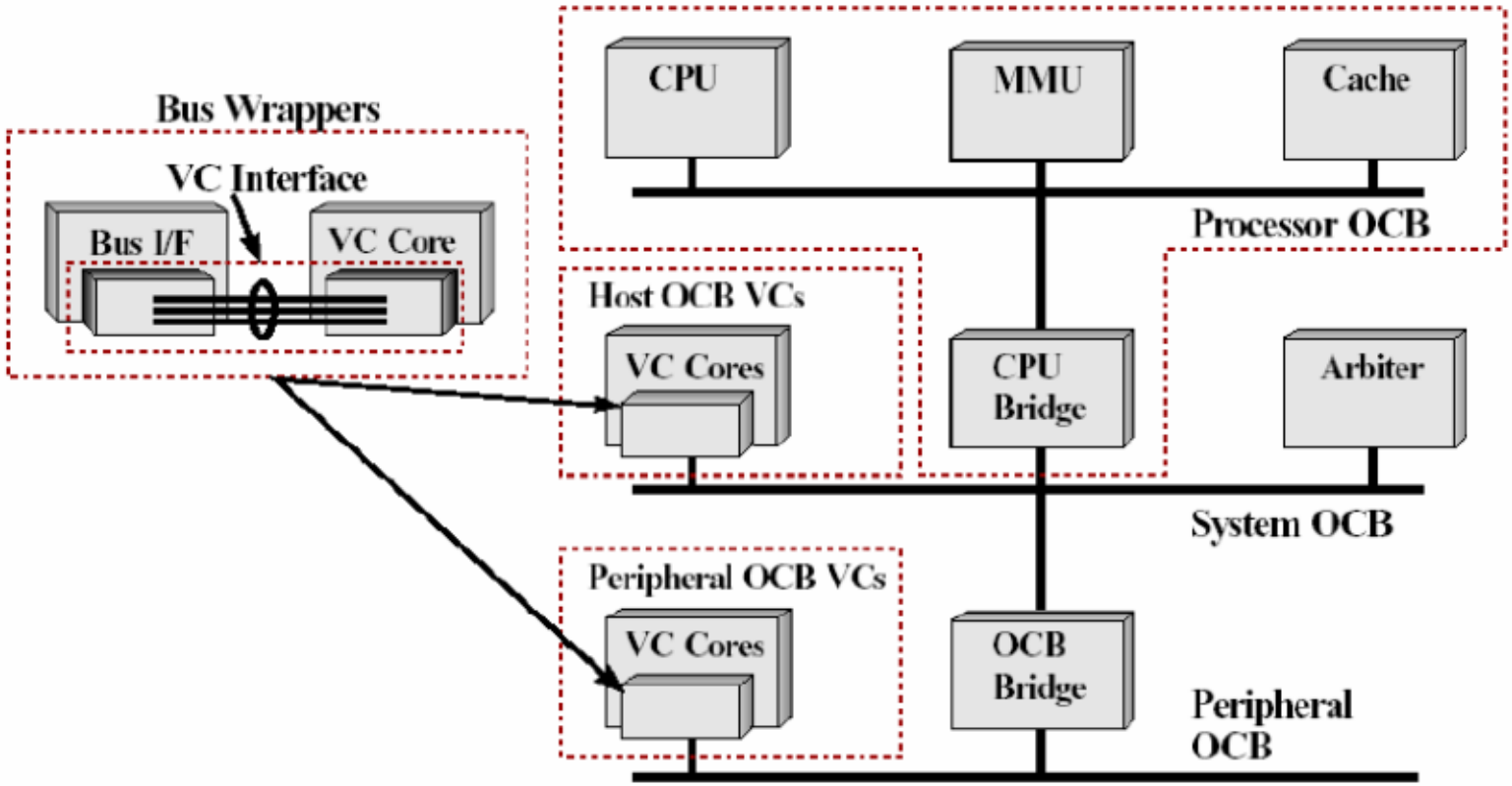
- VCI is an interface rather than a bus.
  - Separating bus-specific logic from the VC.
- VCI specifies:
  - a request-response protocol,
  - a protocol for the transfer of requests and responses,
  - contents and coding of these requests and responses.



Point-to-point communication

- To support true mix-and-match of VCs, VC provider should not need to know the system interconnect. VC provider can design to the VCI as a single interface and the integrator understands how this is translated to his OCB standard.
- Tuning the VC connection to the system's communication channel (bus) by building of the wrapper is left to the system integrator.

VCI



- Bus hierarchy:
  - Processor OCB : Processor/Cache
  - System OCB : RISC/DSP/DMA
  - Peripheral OCB

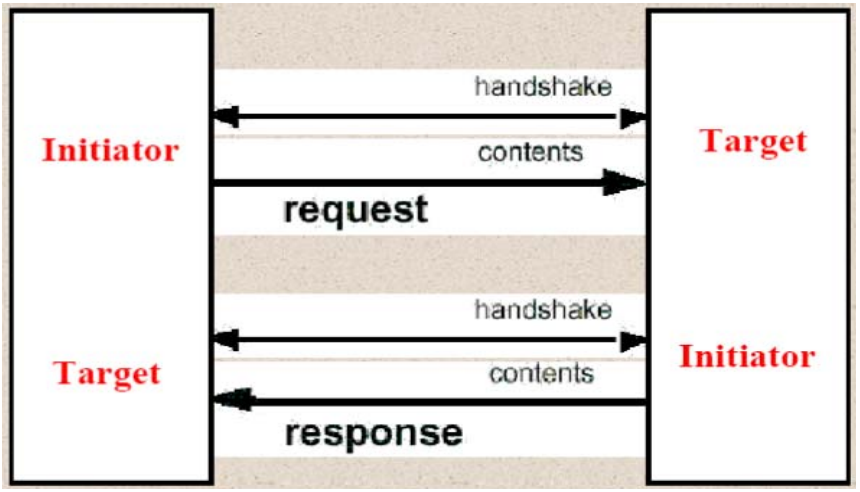
VCI Protocols

1. PVCI = Peripheral VCI: ≈ APB
2. BVCI = Basic VCI: between ASB and AHB
   - Superset of PVCI; adds support for SPLIT transactions, additional burst modes (e.g., wrapped, fixed), additional data transfer modes (e.g., locked/exclusive read), and data bus widths up to 128 bits.
3. AVCI = Advanced VCI (≈ AHB but it is probably better to use OCP)
   - Superset of BVCI; adds additional data transfer modes (e.g., new wrap, defi ned transfer modes), and support for advanced features.

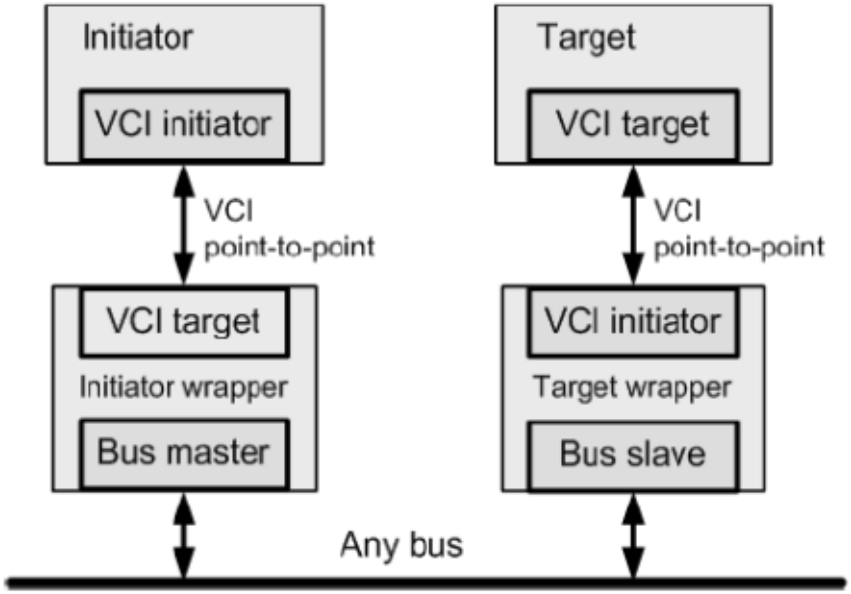
- However, the VCI interface only contains data fl ow signals and does not address issues pertaining to test and control.

BVCI Protocol

- Request and response handshakes are completely independent of each other.
  - Initiator only requests
  - Target only responds
  - If a VC needs both, implement parallel initiator and target interfaces.
- Star topology
- VC connection to any bus:
  - Initiator connects to bus using a bus initiator wrapper.
  - Target connects to that bus using a bus target wrapper.
  - Then, any IPs can be connected to that bus.
    - OCB suppliers provide VCI wrappers



- Two VCI connections are used to realize a bus connection

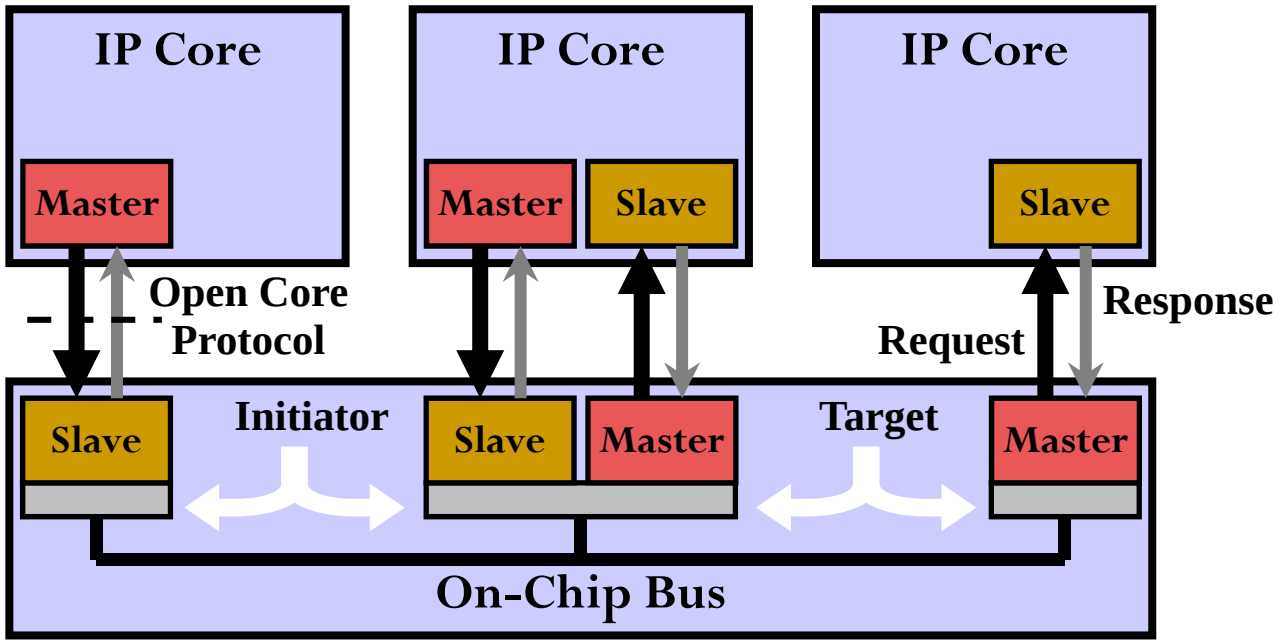


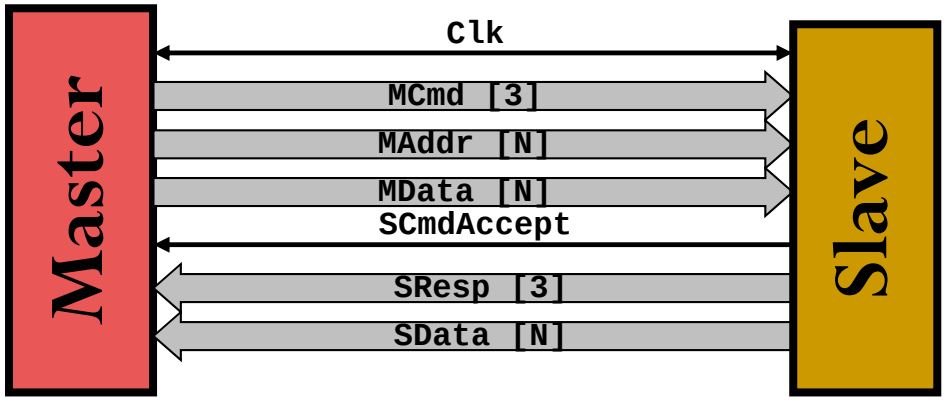
BVCI Protocol: Advantages and Disadvantages

- Advantages
  - Any bus structure can be interfaced
  - Symmetrical signaling enables direct point-to-point communication between 2 cores (without OCB)
  - Burst and split transfers are implicit in the protocol but packets are not trivial to implement
- Disadvantages
  - Can increase the communication time if burst transfers are not used:
    - time for a load on an ARM7 is increased by 67%
    - time for a store on an ARM7 is increased by 50%
  - Increases the circuit area (recommended for large IP)
  - Request and response packets must have the same number of cells (unlike AVCI)

Open Core Protocol (OCP) Goals

- Bus Independent
- Scalable
- Configurable
- Master - slave interface with unidirectional signals
- Signals are strictly point-to-point (except clock & reset)
- OCP and its 4 extensions encompass entire core/system interface needs (data, control, test flows)
- OCP is VCI superset





SoC Buses Features Overview

	Topology						Arbitration						Bus width		Transvers					
Name	Point-to-point	Ring	Unilevel shared bus	Hierarchical bus	Interconnection network	Synchronous/Asinhronous	Static priority	TDMA	Lottery	Round-robin	Token Passing	CDMA	Data bus width [bit]	Address bus width [bit]	Handshaking	Split transfer	Pipelined transfer	Burst transfer	Broadcast	Multicast	Operating frequancy
AMBA	-	-	-	×	-	S	7*	7*	7*	7*	7*	7*	8*	32	×	×	×	×	n/a	n/a	11*
Avalon	×	-	-	-	-	S	13*	13*	13*	13*	13*	13*	1-128	1-32	-	-	×	×	-	-	n/a
Core Connect	-	1*	-	1*	-	S	4*	-	-	-	-	-	9*	10*	×	×	×	×	n/a	n/a	12*
Wishbone	×	×	×	-	×	S	3*	3*	3*	3*	3*	3*	8,16,32,64	1-64	×	n/a	-	×	n/a	n/a	11*
Silicon Backplane	-	-	-	-	×	S	-	6*	-	6*	-	-	8,16,32,64	n/a	×	×	×	×	×	×	n/a
Core Frame	14*	-	-	-	-	S	3*	3*	3*	3*	3*	3*	n/a	n/a	2*	-	n/a	×	×	n/a	n/a
Marble	-	-	-	×	-	A	×	-	-	-	-	-	n/a	n/a	×	×	×	×	×	n/a	n/a
PI bus	-	-	×	-	-	S	3*	3*	3*	3*	3*	3*	1-32	1-32	×	-	×	-	-	-	n/a
OCP	×	-	-	-	-	S	-	-	-	-	-	-	n/a	n/a	×	-	×	×	×	-	n/a
VCI	n/a	n/a	n/a	n/a	n/a	S	3*	3*	3*	3*	3*	3*	n/a	n/a	×	×	×	n/a	-	n/a	n/a
Lotterybus	-	-	-	×	-	S	-	-	×	-	-	-	n/a	n/a	n/a	n/a	n/a	×	n/a	n/a	n/a

Exceptions for Table:1* Data lines shared, control lines point-to-point ring; 2* Palmbus uses handshaking, Mbus does not; 3* Application specific, arbiter can be designed regarding to the application requirements; 4* Programmable priority fairness; 5* Two level arbitration, first level TDMA, second level static priority; 6* Two level arbitration, first TDMA, second round-robin token passing; 7* Application specific except for APB which requires no arbitration; 8* For AHB and ASB bus width is 32, 64, 128 or 256 byte, for APB 8, 16 or 32 byte; 9* For PLB bus width is 32, 64, 128 or 256 byte, for OPB 8, 16 or 32 byte and for DCR 32 byte; 10* For PLB and OPB bus width is 32 byte, and for DCR 10 byte; 11* User defined operating frequency; 12* Operating frequency depending on PLB width; 13* Slave side arbitration; 14* System of buses, Palmbus and Mbus, both are point-to-point;

Milica Mitiˊc and Mile Stojˇcev, An Overview of On-Chip Buses

Limitations of Bus-Based On-Chip Networks

- Lack of Software interface
  - Only low-level hardware interface available.
- Lack of SW Interrupt handling
  - Only low-level hardware interrupt available.
- Lack of Debugging features
- Lack of interfacing with off-chip bus
- Lack of Error detection/correction.

Network-on-Chip tries to solve these problems.