TU WIEN Informatics

Advance Computer Architecture

F1 On-chip Buses

Daniel Müller-Gritschneder

Motivation

- Most chips feature a range of processing elements (PEs) / multi-cores
- PEs needs to communicate with each other
- On-chip Interconnect architecture and type play crucial role in performance.
- Chips and devices are connected via different types of interconnects

Agenda

- Interconnect types
- On-chip buses
- Networks-on-chip (NoC)

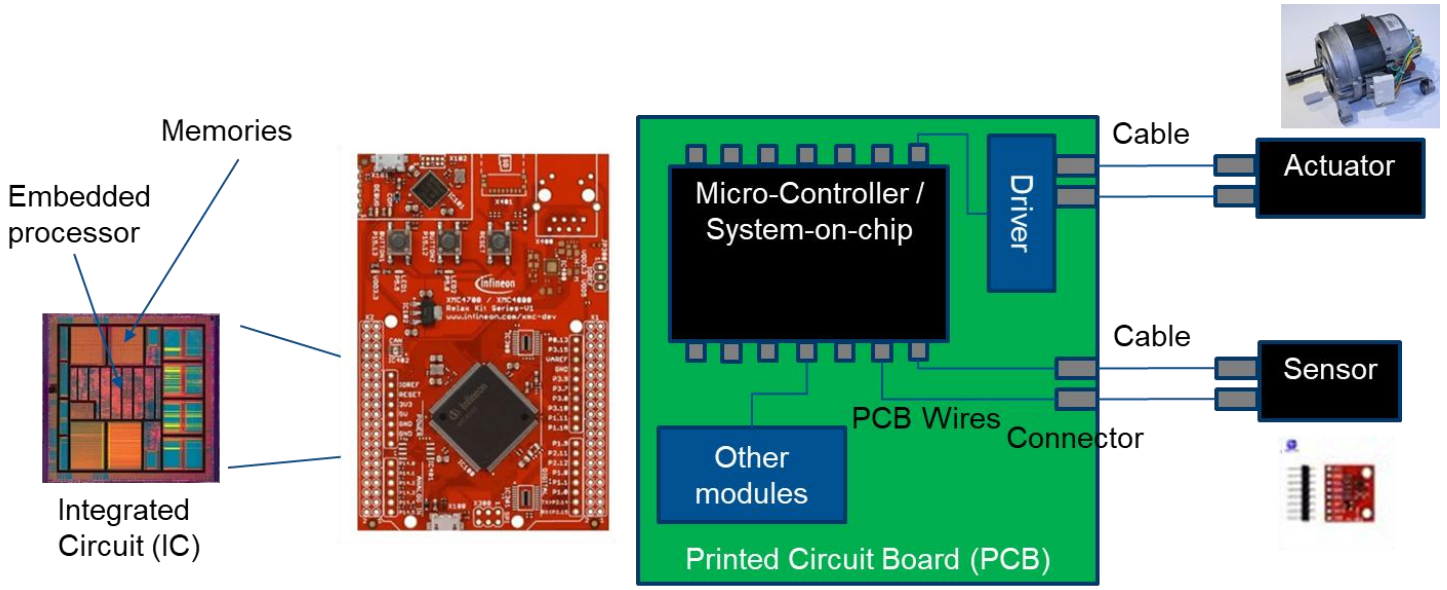
- A look at real on-chip interconnects

Optional, not relevant for exam

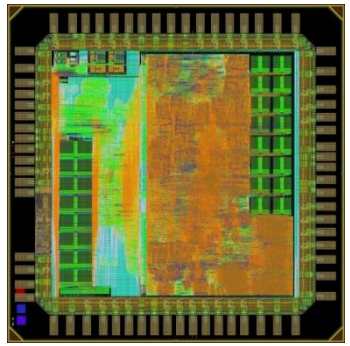
F1.1 Types of Interconnects

Interconnect Types

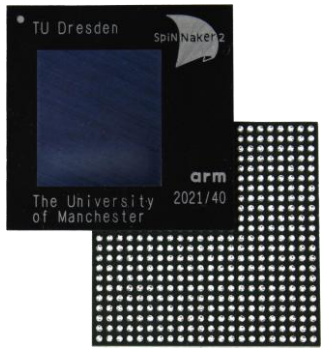
- On-Chip: Connects modules that are integrated into the same chip (IC: integrated circuit)
- PCB-level: Connects different ASICs + connectors and other component all mounted on one Printed Circuit Board (PCB).
- Many other interconnects (board to board, rack to rack): PCIe, Ethernet, CAN, UART, I2C, SPI, GPIO, …



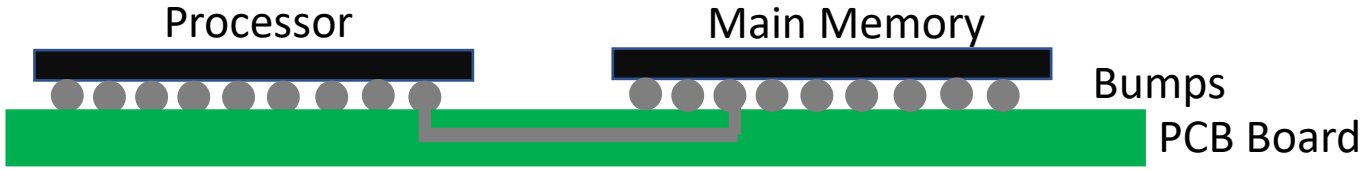
Different Scales of Interconnects



On-chip Interconnect

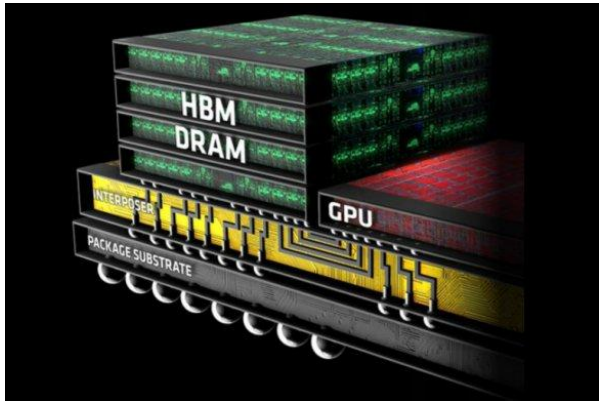


Chip Package



Chip2Chip

Chip2Chip (3D Stacked)



Source AMD

Board2Board

Rack2Rack

Sources: Pulp, SpiNNCloud

F1.2 On-chip Buses - Introduction

Memory-mapped Buses

- Purpose:
  - Read or write a value from or to a certain address
  - Value can be data or peripheral control information
- Memory-mapped Bus has several (sub-)buses (group of signals) and a defined bus protocol
  - Address bus
  - Data bus for reading data
  - Data bus for writing data
  - Control signals: Indicate if access is read or write, bust length, ID, bus grant, …
- Modules on the bus can either act as initiators or targets
  - Typical initiators: CPUs, DSPs, DMAs, bus bridges, …
  - Typical targets: Memory, accelerators, interface peripheral, bus bridges, …

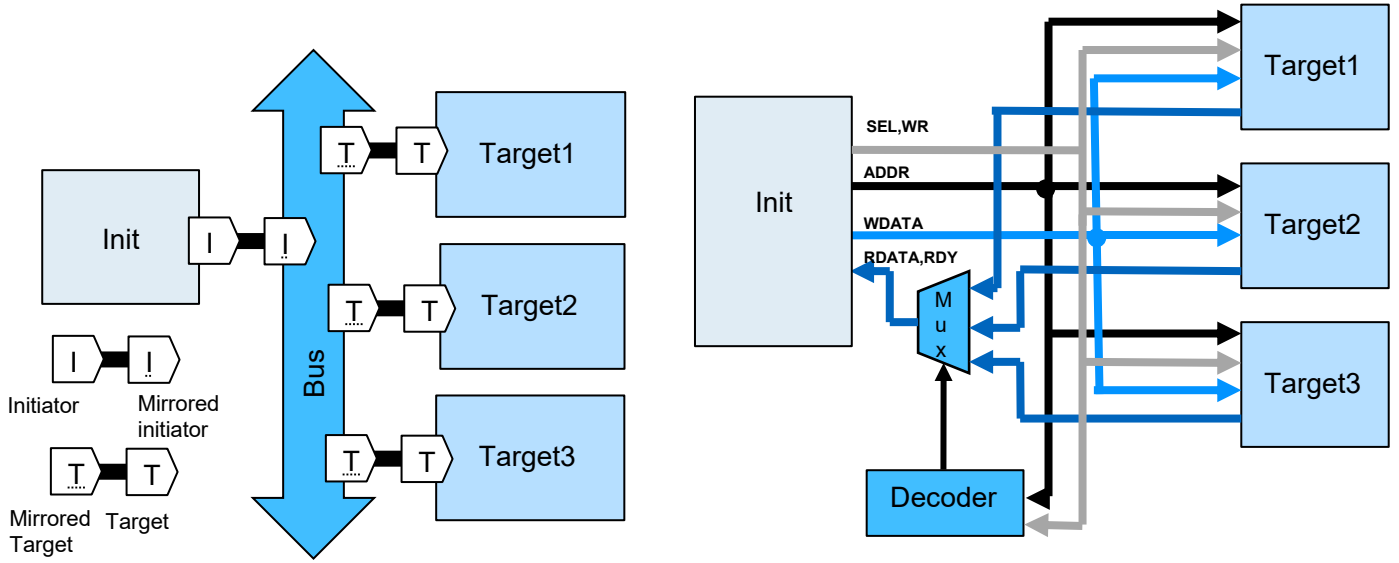
Classes of Memory-mapped Buses

- Single-initiator bus:
  - One initiator component can address different target components, which are mapped to different addresses
- Shared bus:
  - There are several initiators on the bus
  - An arbiter decides which initiator module is granted access to the bus
  - Only one initiator can access one slave via the bus at a time
- Layered bus:
  - There is more than one arbiter such that more than one initiator is granted access on the bus
  - Only one target component on each layer can be accessed at a time
- Crossbar/ bus matrix
  - Each target component has its own arbiter
  - Each target component can be accessed by one initiator at a time

F1.3 On-chip Buses – Single Initiator

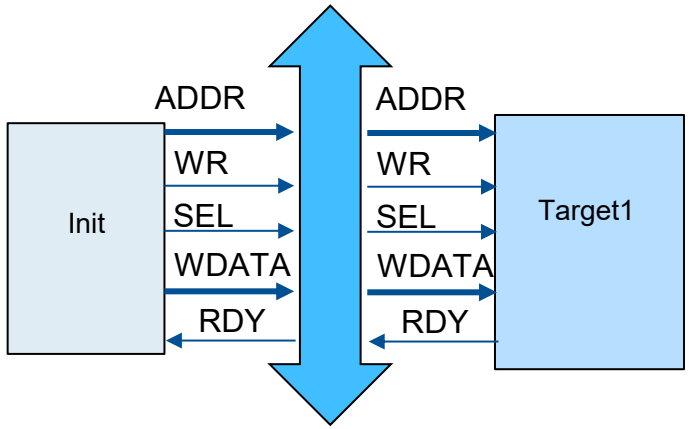
Single-Initiator Bus

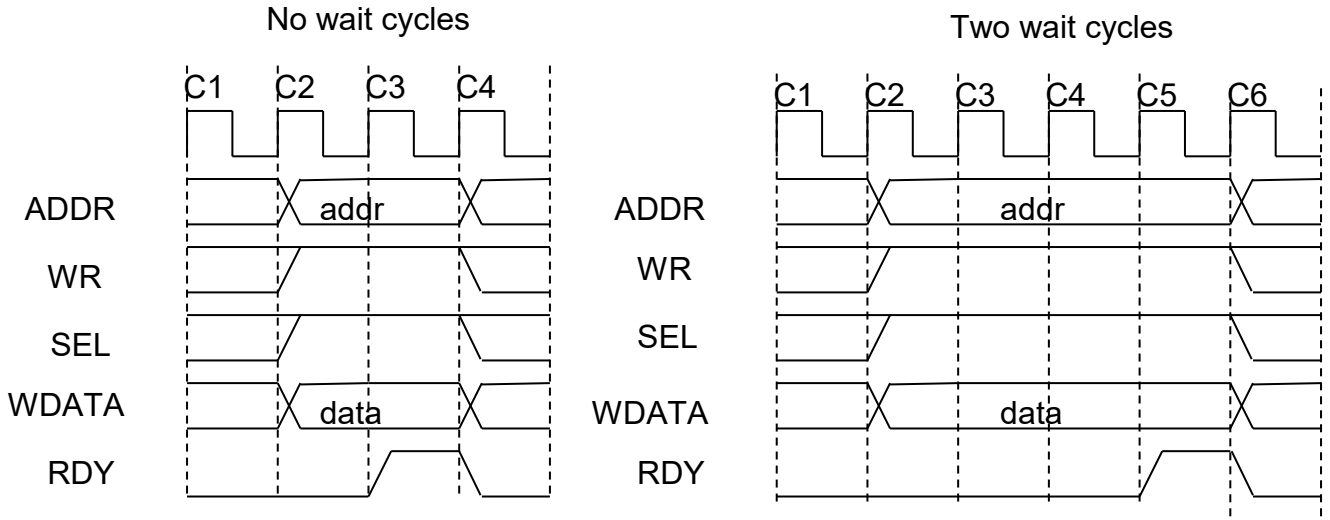
- Target knows
  - if it is addressed by observing the address bus ADDR
  - or decoder generates SEL signal for targets based on address bus ADDR
- Target can receive data on write data bus WDATA
- Decoder forwards the data from the addressed target by multiplexing it to the read data bus RDATA
- Additional control bus CTRL for signals related to bus protocol (e.g. WR, SEL, RDY)



Simple Write Access

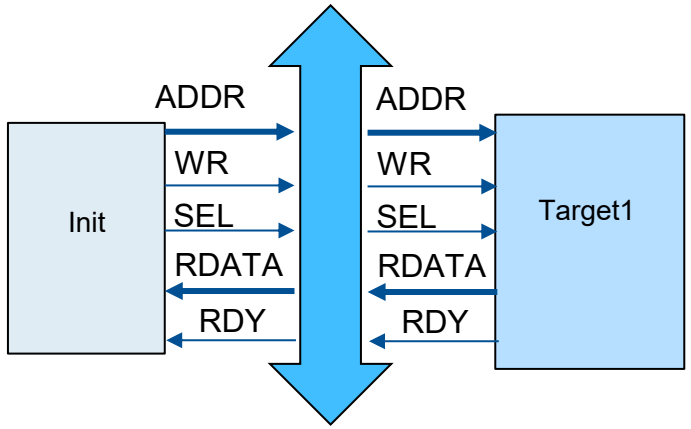
1. Initiator places address and data on the ADDR and WDATA bus
   Initiator indicates write by setting signal WR to high
   Initiator indicates that access is started by setting SEL signal to high
2. Target acknowledges write access by RDY signal

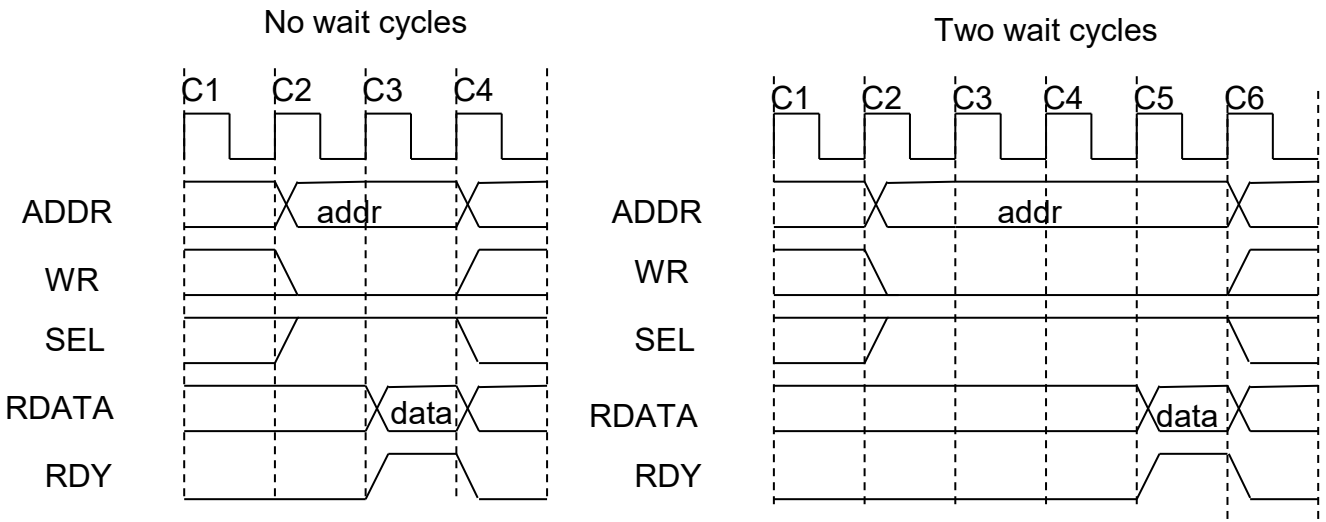




Simple Read Access

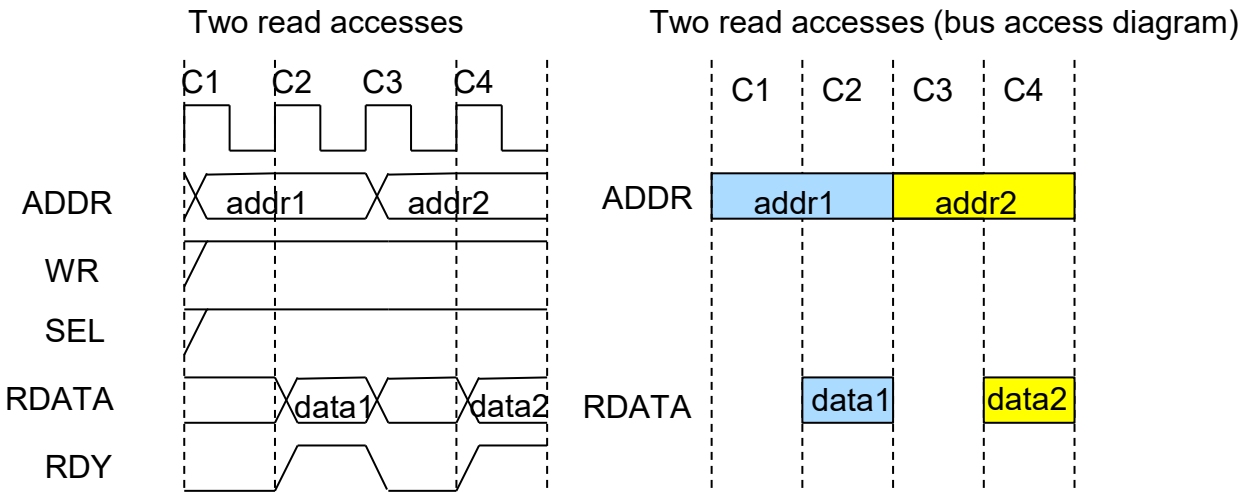
1. Initiator places address on the ADDR bus
   Initiator indicates read access by setting signal WR to low
   Initiator indicates that access is started by setting SEL signal to high
2. Target places data on RDATA bus
   Target acknowledges write access by RDY signal





Performance of Simple Accesses

- Each access takes minimally two cycles
- Maximal bus bandwidth is: $BW_{bus} = 0.5 \cdot buswidth \cdot f_{bus}$

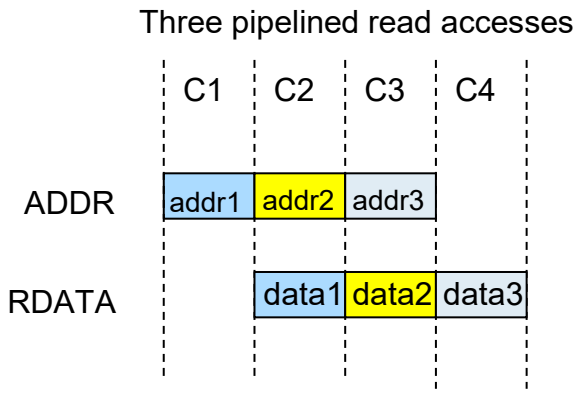


Pipelined Accesses

- The next address can be placed on the bus while the data is read
- Maximal bandwidth supported by bus is equal to:

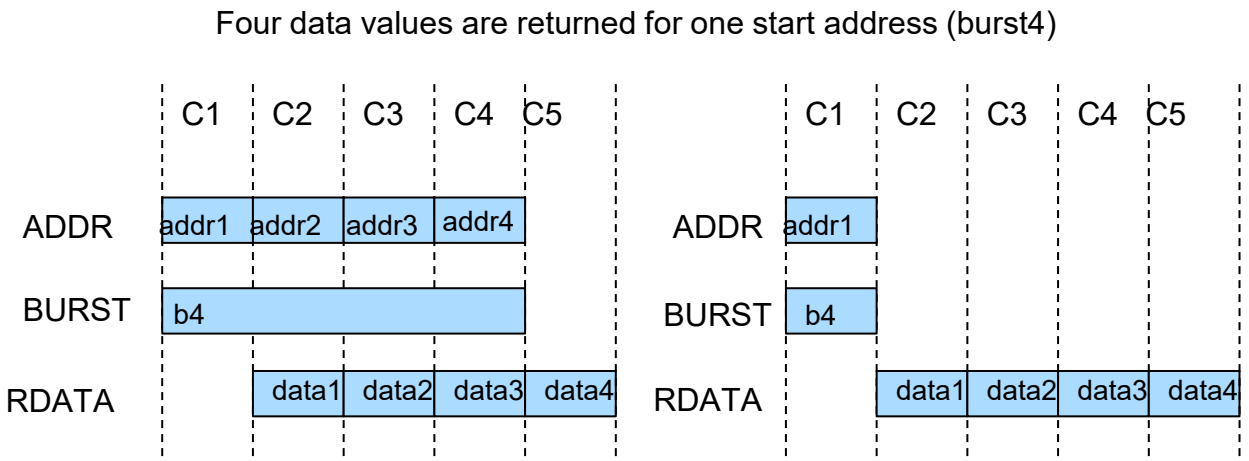
$$BW_{bus} = buswidth \cdot f_{bus}$$

- Additional control signals and logic required to support pipelined accesses.



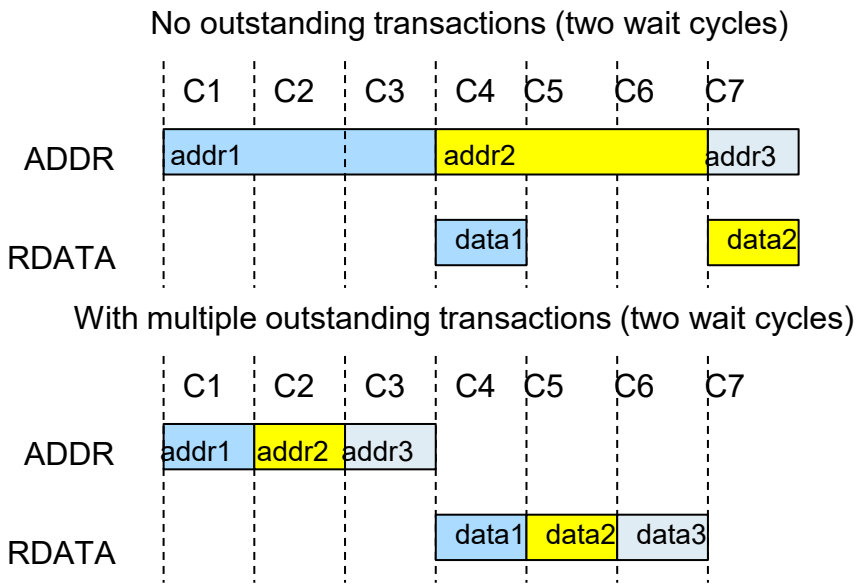
Burst Accesses

- A burst accesses a consecutive row of addresses
- Version 1: the addresses for all accesses must be given and a control signal that indicates that this is a burst access of a certain size
- Version 2: Only the start address must be given and a control signal that indicates that this is a burst access of a certain size



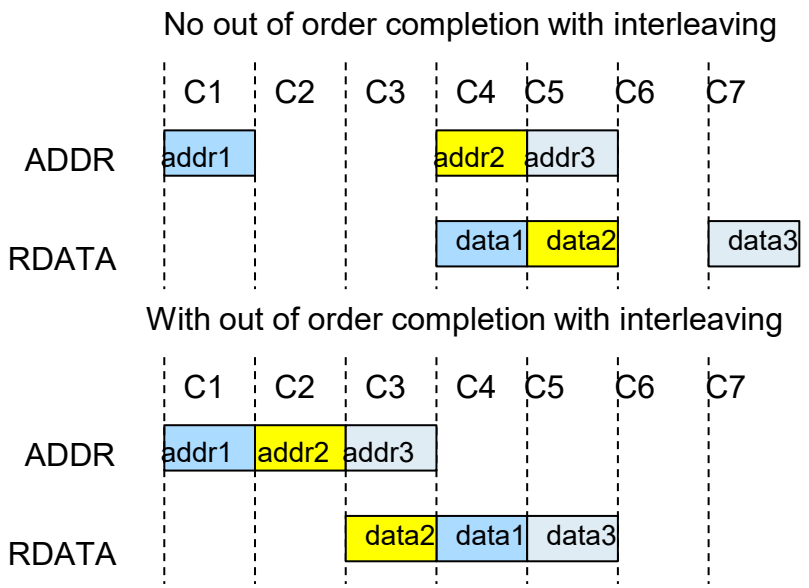
Multiple Outstanding Transactions

- A address may be placed on the bus before the data of the previous access has been read or be written
- This improves performance in case of wait cycles.



Out of order Completion with Interleaving

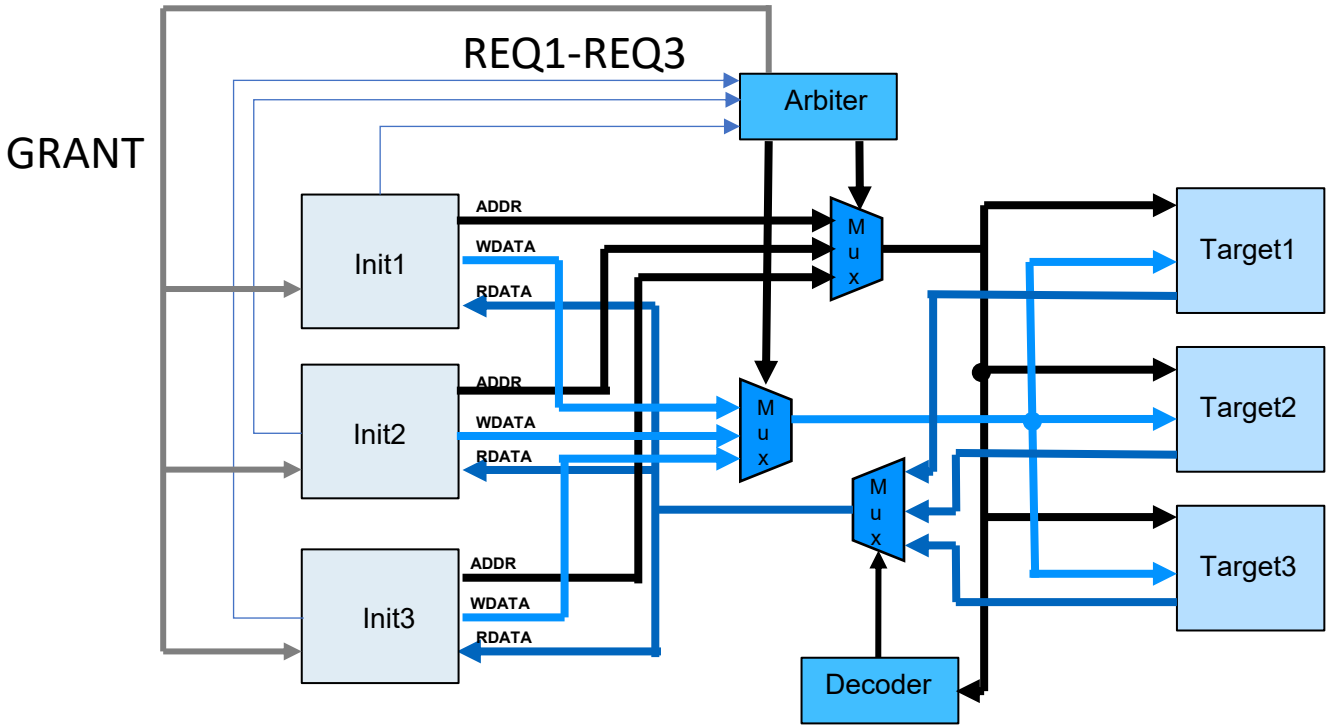
- A address may be placed on the bus before the data of the previous access has been read or be written
- In case of wait cycles, the order of data reads may be changed



F1.4 On-chip Buses – Multiple Initiators

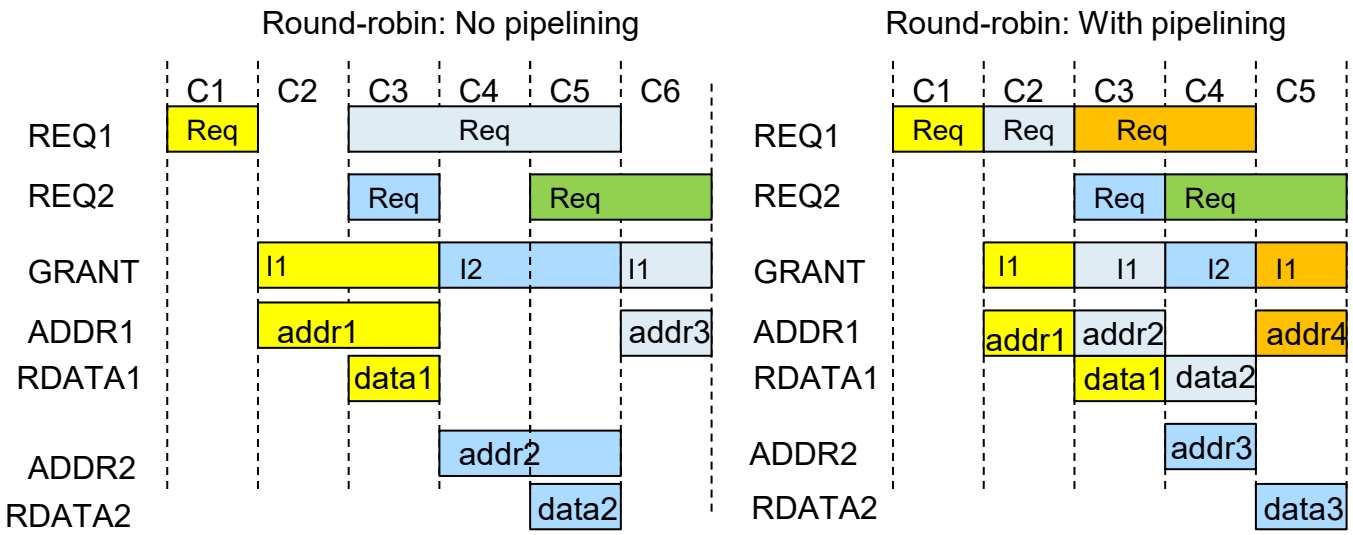
Shared Bus

- Arbiter grants access to the initiator:
- Only the address and data of one initiator is forwarded to the targets



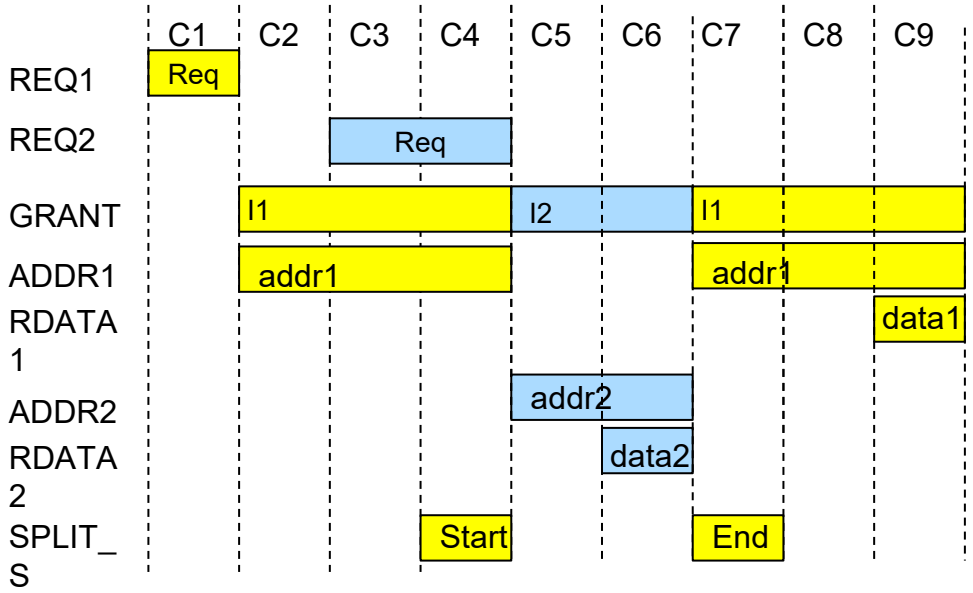
Bus Arbitration

- The arbiter grants access to initiator that request the bus
- Round-robin: Access granted to initiators in pre-defined order that is repeated
- FIFO: First initiator requesting the bus is granted access
- Priority: Initiator with highest priority is granted access to the bus



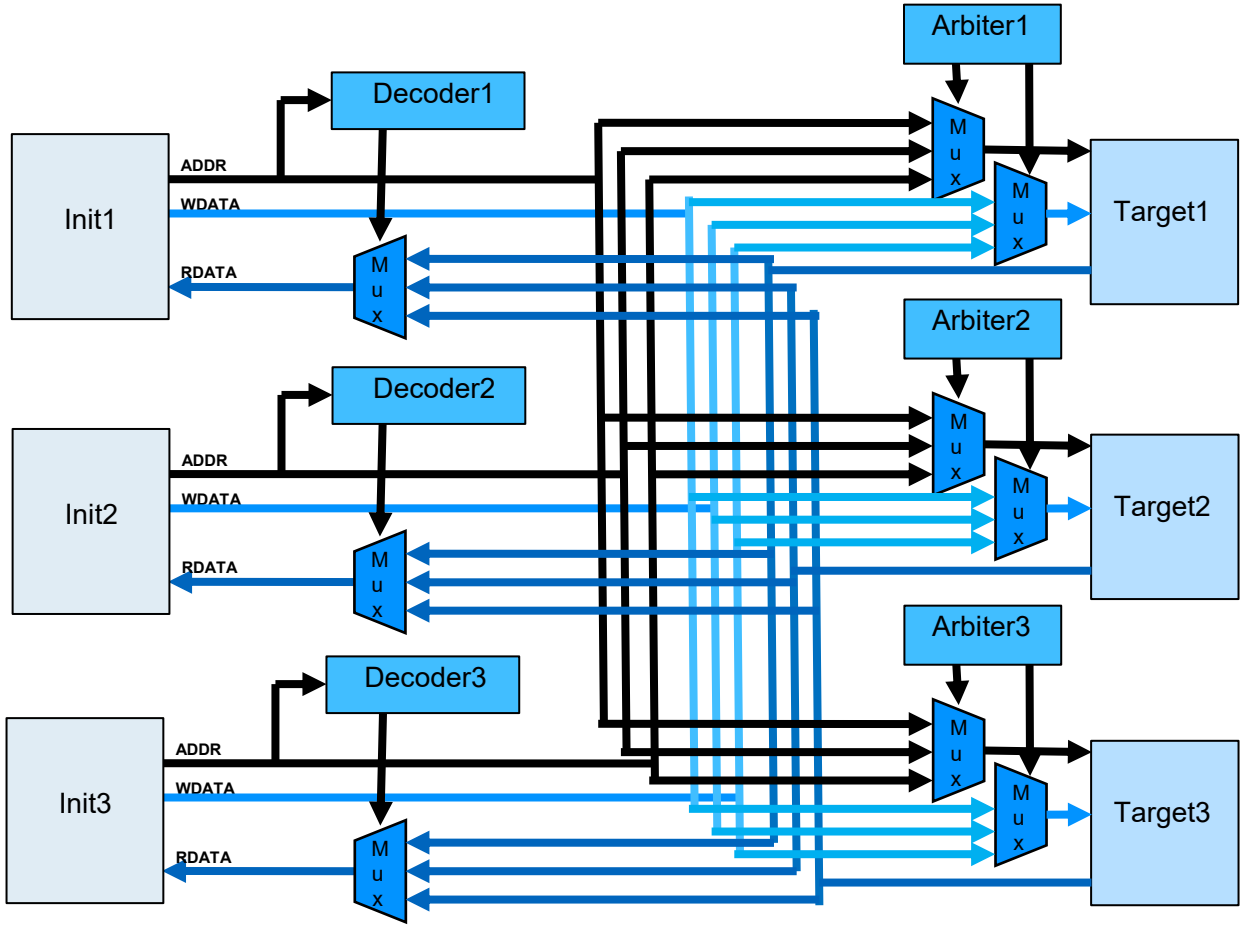
Split Accesses

- Slave can allow a split of an access if it was many wait cycles
- Access of initiator I1 is split by issuing a start of split by slave
- I2 is granted the bus and access of initiator I2 is performed
  Then access of initiator I1 is finished by issuing an end of split



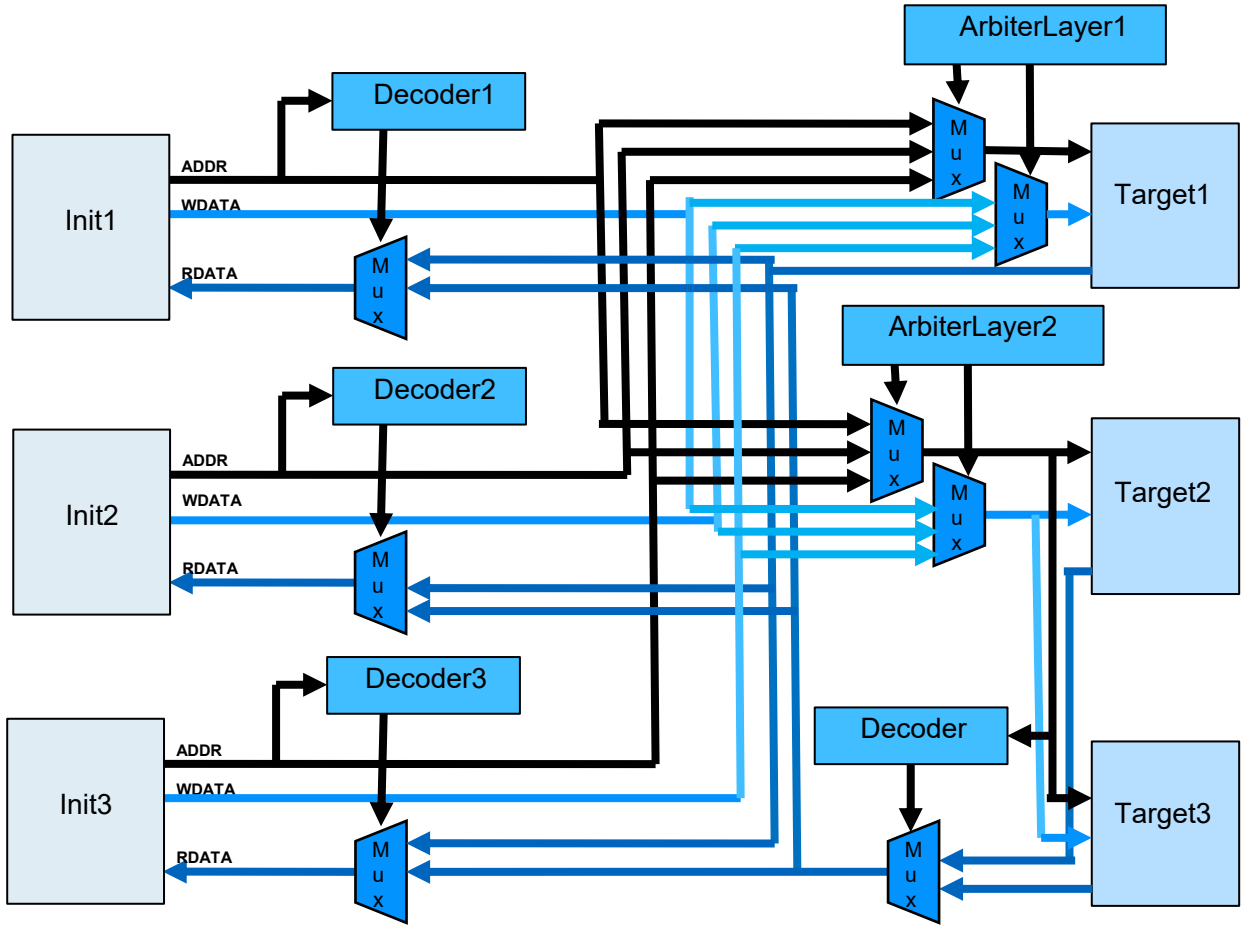
Crossbar / Bus Matrix

- All targets can be accessed individually
- Only conflict when two initiators access same target
- GRANT/REQ omitted.



Layered Bus

- Targets are on different layers
- Initiator can connect to targets on different layers simultaneously



Some Bus Standards

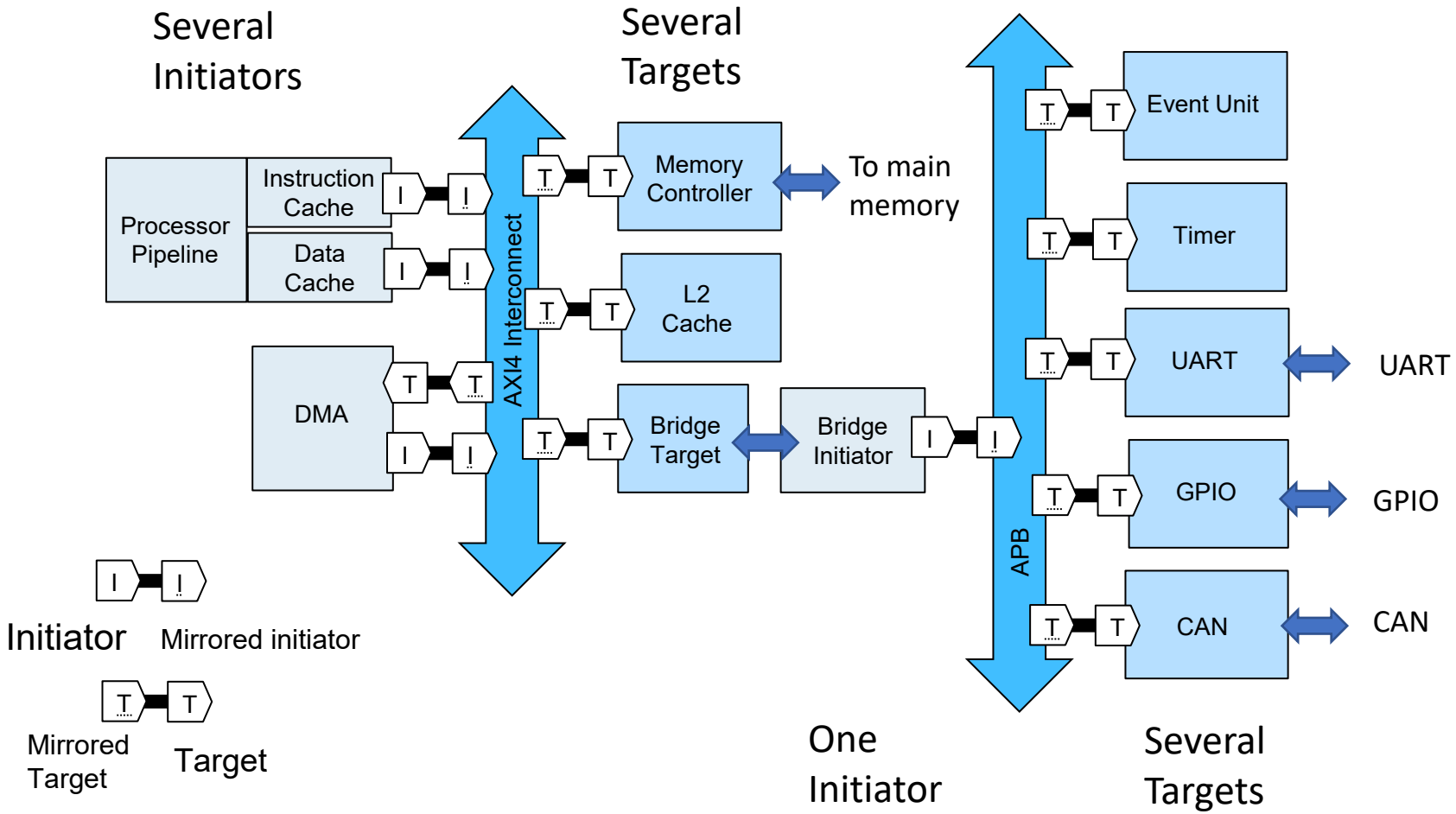
- AMBA Bus (ARM)
  - AHB: Advanced High Performance Bus
  - APB: Advanced Peripheral Bus
  - AXI: Advanced eXetendible Interface
- Wishbone (Open)
- TileLink (Open)

ARM AMBA Standard

- Different Versions e.g., AMBA 2,0, AMBA 3.0,…
- AHB: Advanced High Performance Bus
  - High performance
  - Pipelined operation
  - Multiple bus initiators
  - Burst transfers
  - Split transactions
- APB: Advanced Peripheral Bus
  - Low power
  - Simple Interface
  - Suitable for many peripherals
  - One initiator (APB Bridge)
- AXI: Advanced eXetendible Interface
  - Configurable channel-based specification

Typical On-Chip Interconnect for Smaller Embedded Devices

- High-performance near the processor cores, low-performance near the slow I/O devices



Summary

Conclusion

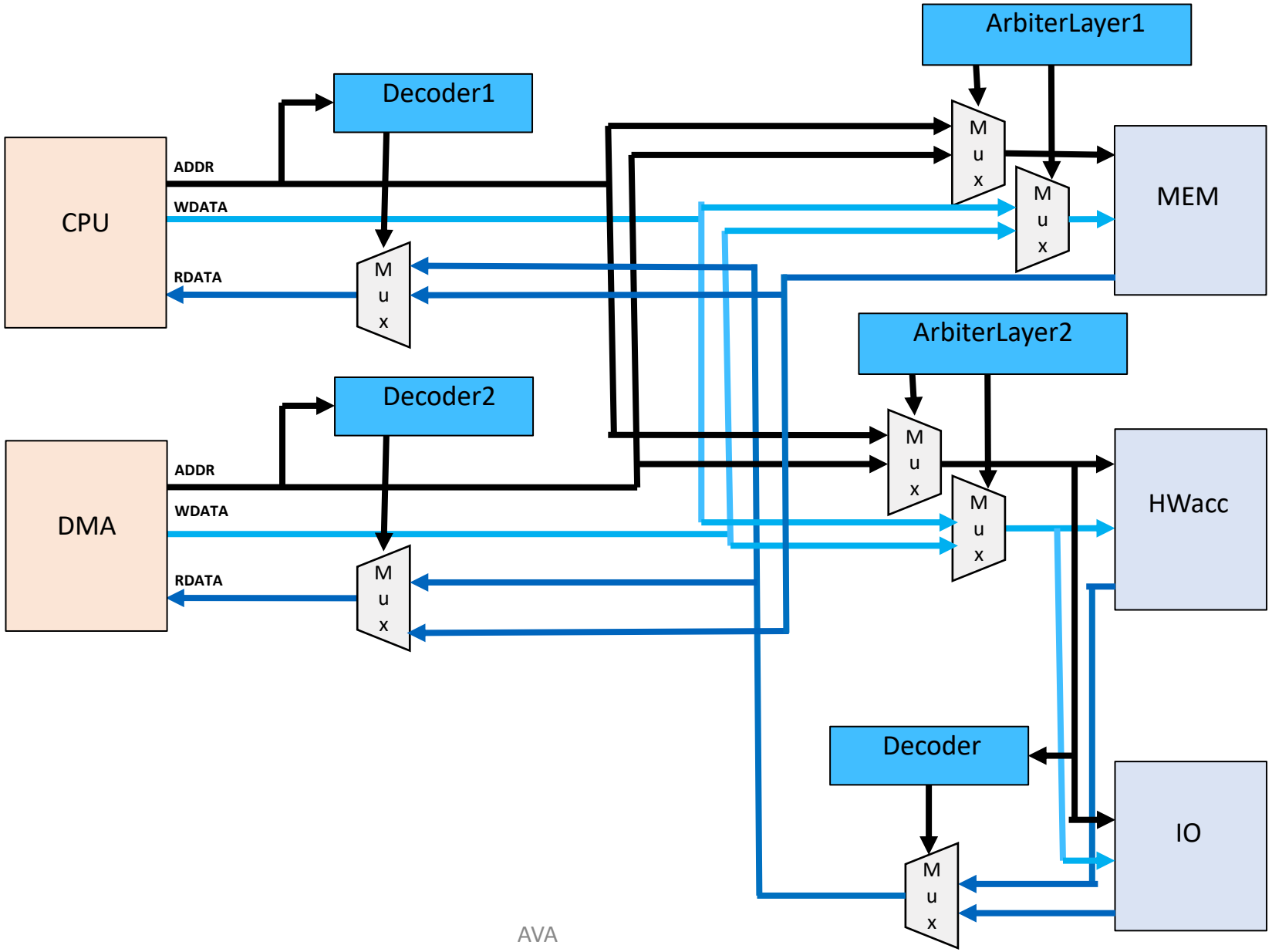
- Bus-based On-chip Interconnect
- Network on-Chip
- Next Sessions: Specialized Cores

Thank you for your attention

Example – Layered Bus

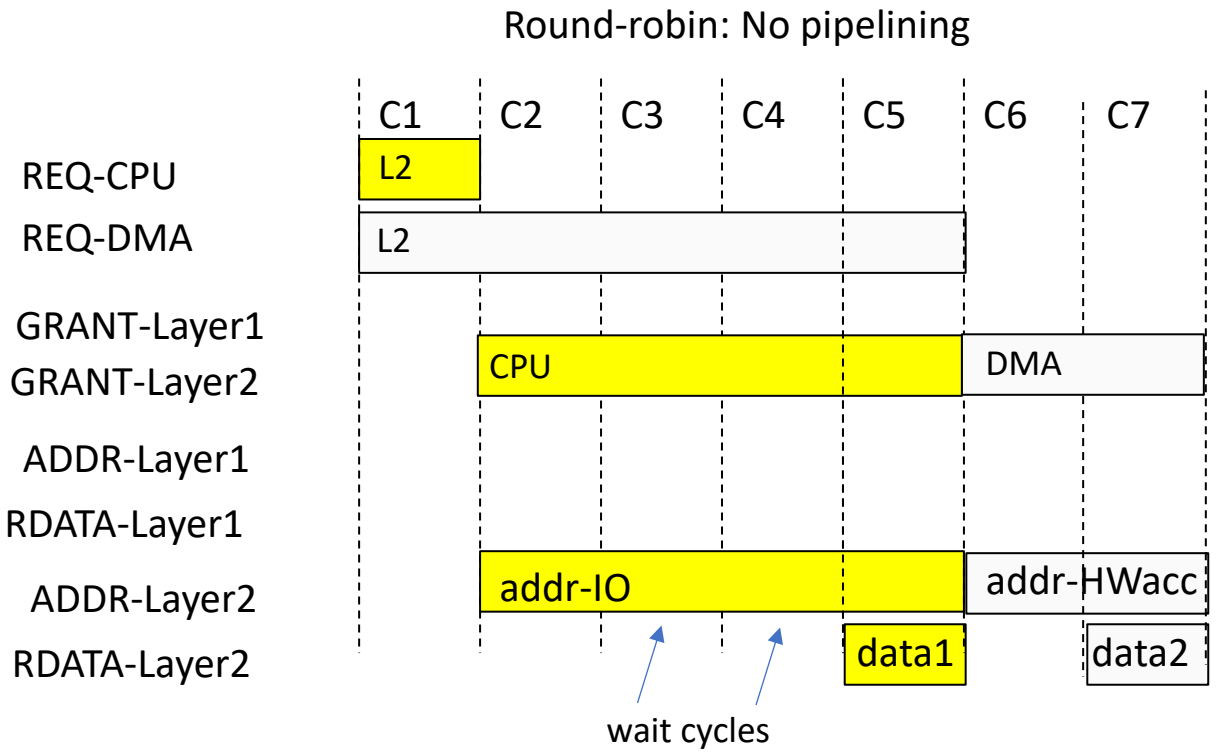
- Given is the following architecture for a shared layered bus:
  - There are two initiator components, CPU and DMA.
  - There are three target components, MEM, HWacc and IO.
    The MEM, is on layer 1, the Hwacc and IO component is on layer 2.

Example – Layered Bus



Example – Layered Bus - Access

- Assume that the CPU wants to read access the IO slave component in the bus cycle 1 and that the DMA wants to read access the HWacc in the same bus cycle 1. Draw the bus access diagram for the data and address bus of the two bus masters as well as the control request and grant signals for the two layers assuming that the bus does not support pipelining. The IO component inserts two wait cycles. The HWacc component inserts no wait cycles. The arbitration order is CPU first, then DMA. There is no pipelining.



Example – Layered Bus - Access

- Assume that the CPU wants to read access the IO slave component in the bus cycle 1 and that the DMA wants to read access the HWacc in the same bus cycle 1. Draw the bus access diagram for the data and address bus of the two bus masters as well as the control request and grant signals for the two layers assuming that the bus does not support pipelining. The IO component inserts two wait cycles. The HWacc component inserts no wait cycles. The arbitration order is CPU first, then DMA. There is no pipelining.