Computer Architecture:
I/O and Handshaking

Hossein Asadi (asadi@sharif.edu)
Department of Computer Engineering
Sharif University of Technology
Spring 2024

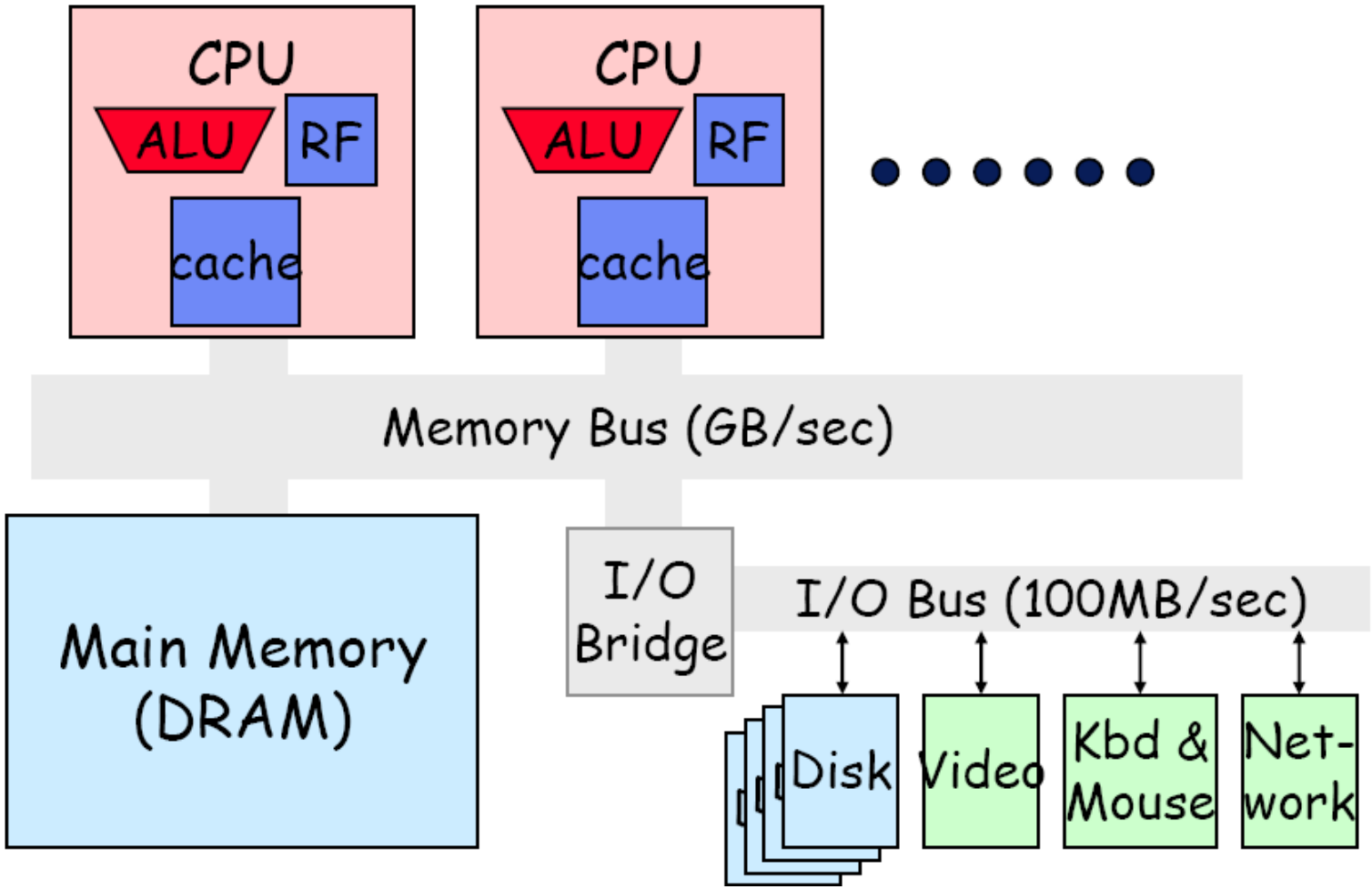
Copyright Notice

- Some Parts (text & figures) of this Lecture adopted from following:
  - D.A. Patterson and J.L. Hennessy, "Computer Organization and Design: the Hardware/Software Interface" (MIPS), 6th Edition, 2020.
  - J.L. Hennessy and D.A. Patterson, "Computer Architecture: A Quantitative Approach", 6th Edition, Nov. 2017.
  - "Intro to Computer Architecture" handouts, by Prof. Hoe, CMU, Spring 2009.
  - "Computer Architecture & Engineering" handouts, by Prof. Kubiatowicz, UC Berkeley, Spring 2004.
  - "Intro to Computer Architecture" handouts, by Prof. Hoe, UWisc, Spring 2021.
  - "Computer Arch I" handouts, by Prof. Garzarán, UIUC, Spring 2009.

Topics Covered in This Lecture

- **I/O & Mass Storage**
  - **Busses**
  - **I/O Handshaking**

Computer Organization



Input/Output

- Fact
  - Input/outputs very slow devices
    - Average response time: few mili-seceonds
  - CPUs very fast devices
    - Average running time: nano-seconds
- Question:
  - How Input/Output devices are connected to CPUs?
    - I/O controller (also called I/O bridge)

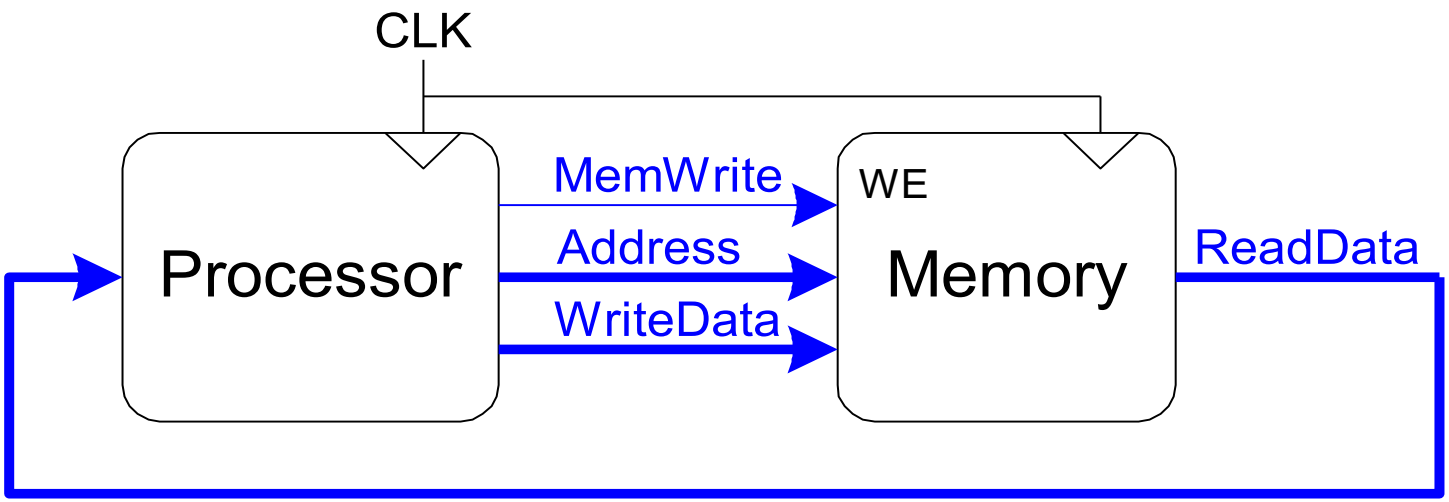
Input/Output (cont.)

- I/O Controller
  - Input/Output handshaking
    - Interrupt
    - Polling

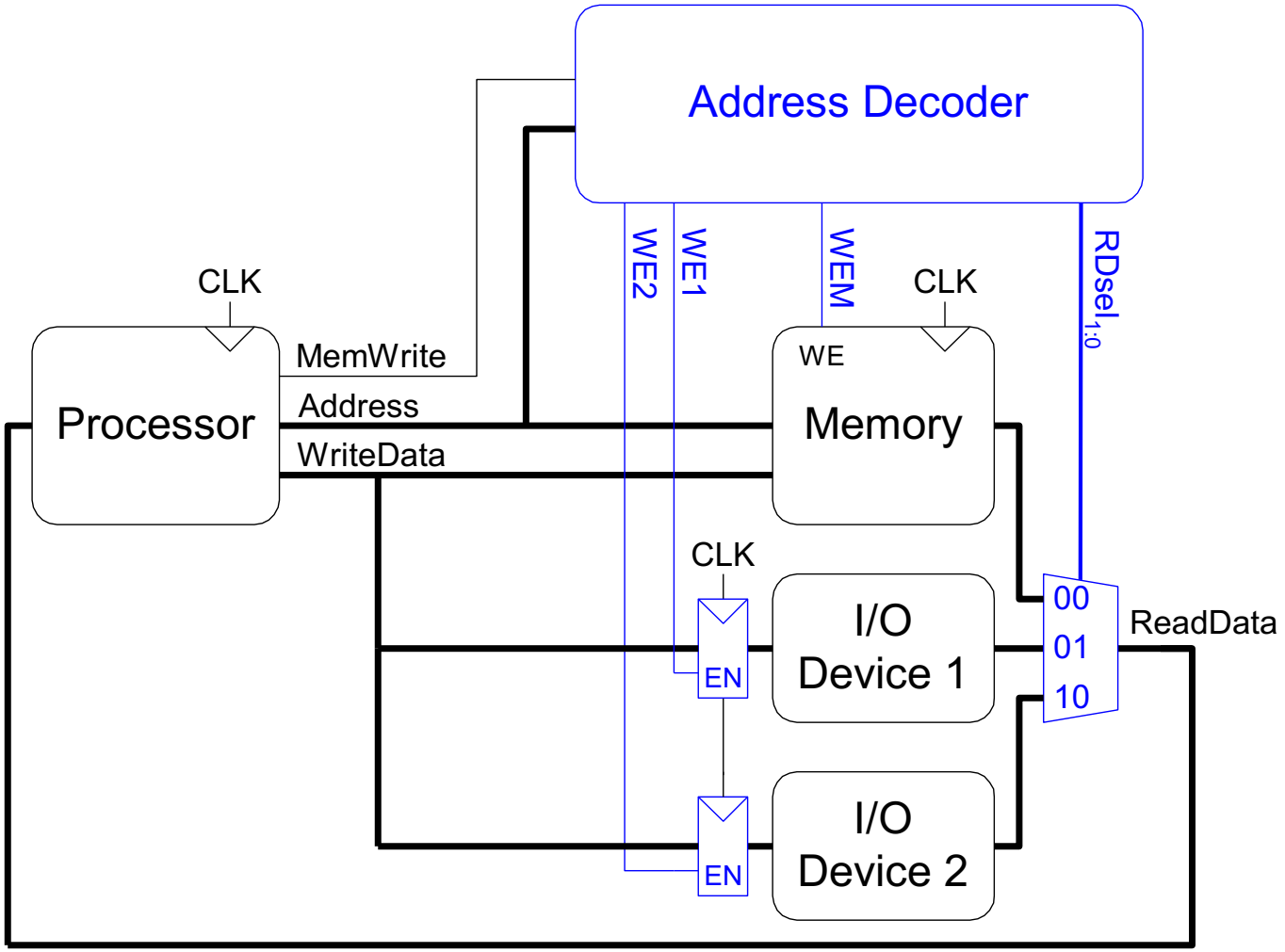
Input/Output (cont.)

- How to Access I/O?
  - From ISA perspective
- I/O Configuration
  - I/O-Mapped I/O
  - Memory-Mapped I/O

Memory Interface



Memory-Mapped I/O Hardware



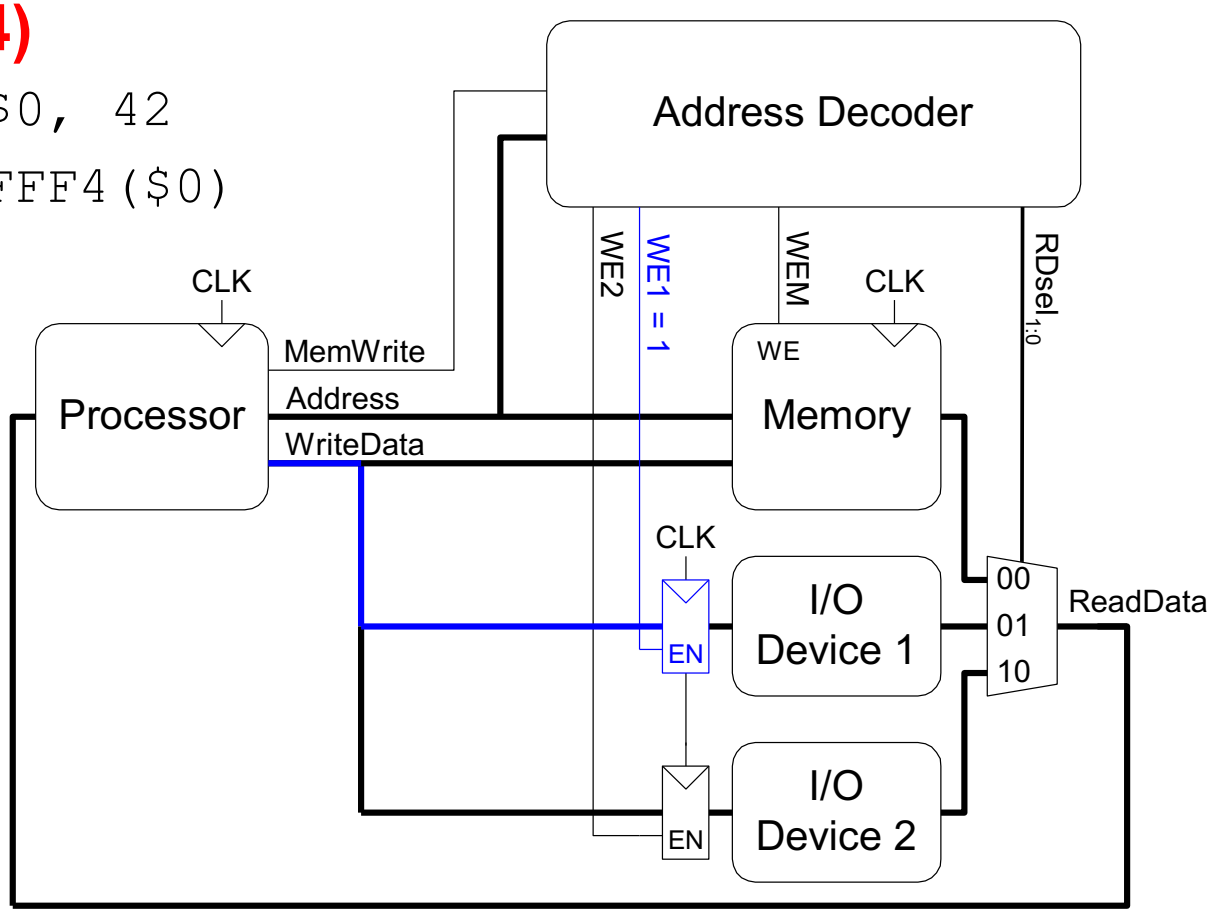
Memory-Mapped I/O Hardware (cont.)

- Suppose I/O Device 1 is assigned the address 0xFFFFFFF4
  - **Write the value 42 to I/O Device 1**
  - **Read value from I/O Device 1 and place in `r1`**

Memory-Mapped I/O Hardware (cont.)

- **Write the value 42 to I/O Device 1 (0xFFFFFFF4)**

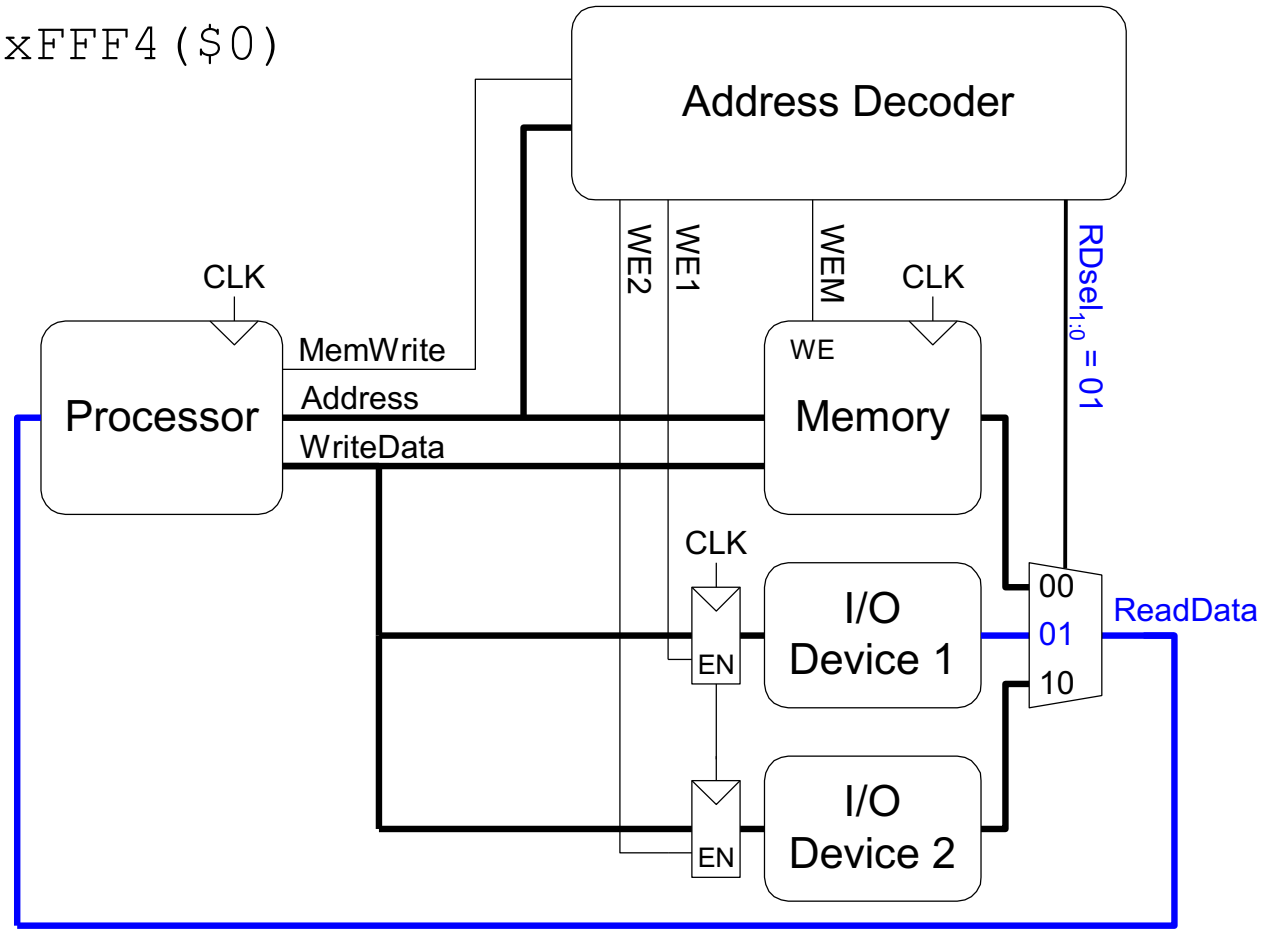
```
addi r1, $0, 42
sw r1, 0xFFF4($0)
```



Memory-Mapped I/O Hardware (cont.)

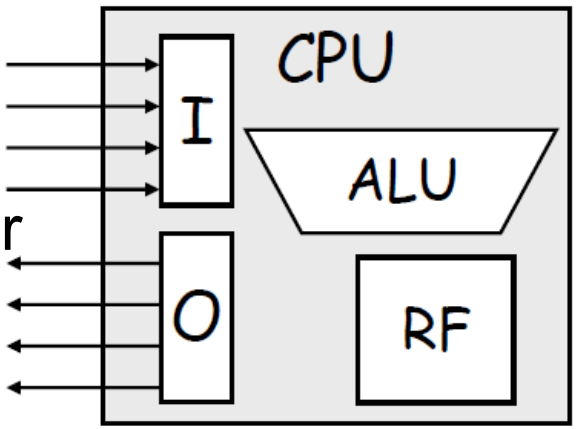
- **Read value from I/O Device 1 and place in `r1`**

```
lw r1, 0xFFF4($0)
```



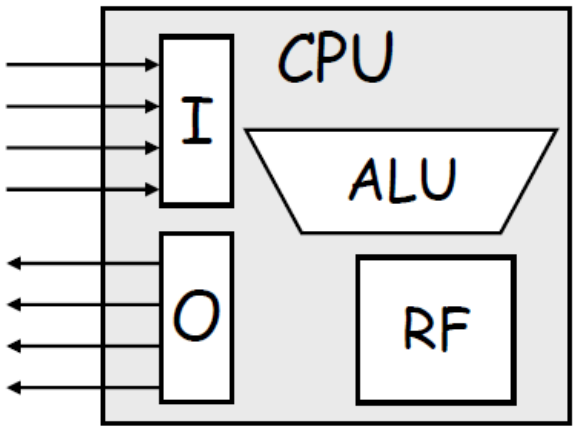
Input/Output (cont.)

- I/O Mapped I/O
  - Dedicated I/O instructions
    - Part of ISA
  - Output
    - Values written to an output register
    - On output pins of an external port
  - Input
    - Values read from an input register
    - At input pins of an external port
  - Example: Intel microprocessors: IN / OUT



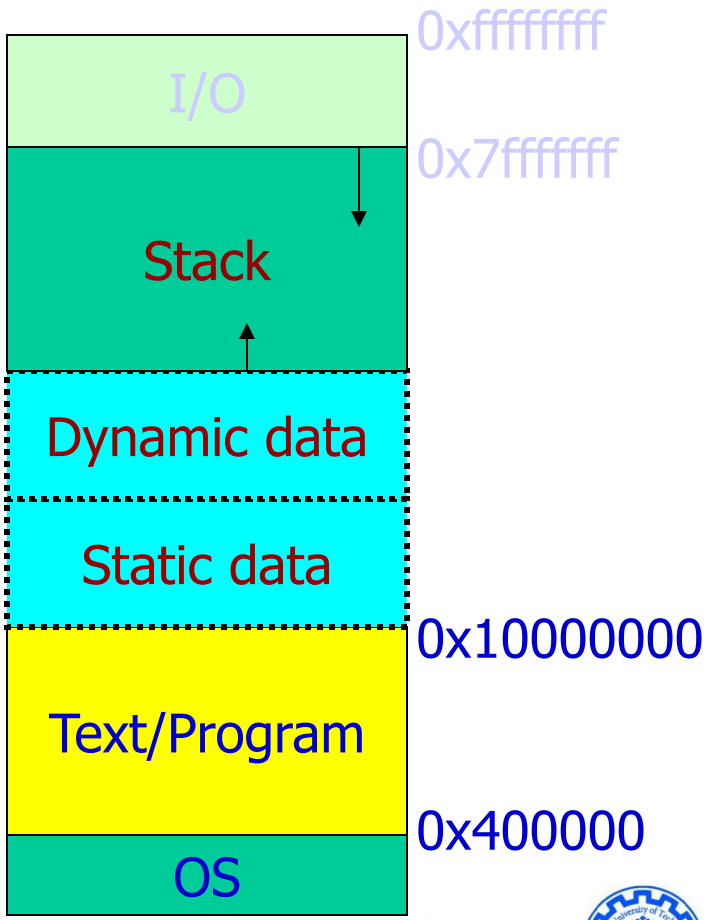
I/O-Mapped I/O

- Pros
  - Low latency
    - Simpler decoding
- Cons
  - Not implemented by all CPUs
  - Limited addressing
    - Predetermined # of I/O signals



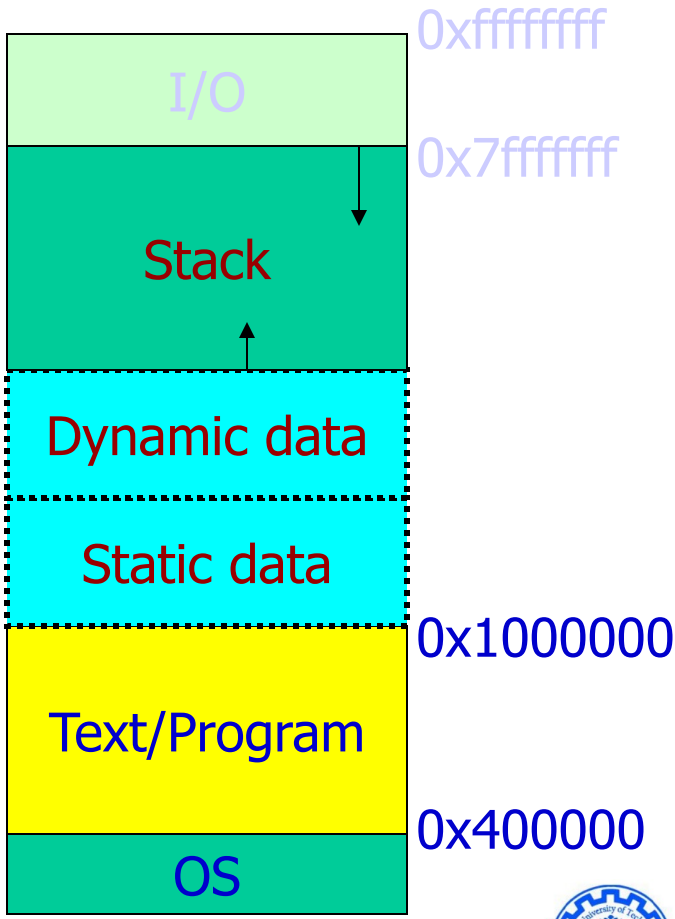
Input/Output (cont.)

- Memory-Mapped I/O
  - Performed by simple load/store instructions
  - A subset of unused memory addresses mapped to registers of external devices
  - Memory & devices on bus programmed to respond only to their own address ranges



Memory-Mapped I/O

- Pros
  - Easy to handle memory locations (SW perspective)
  - Wide range of addresses
- Cons
  - Caching may cause to memory incoherency issue
  - Slow & consumes CPU cycles
    - ➔ Use DMA

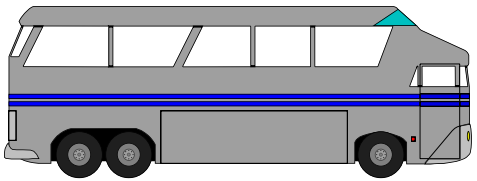


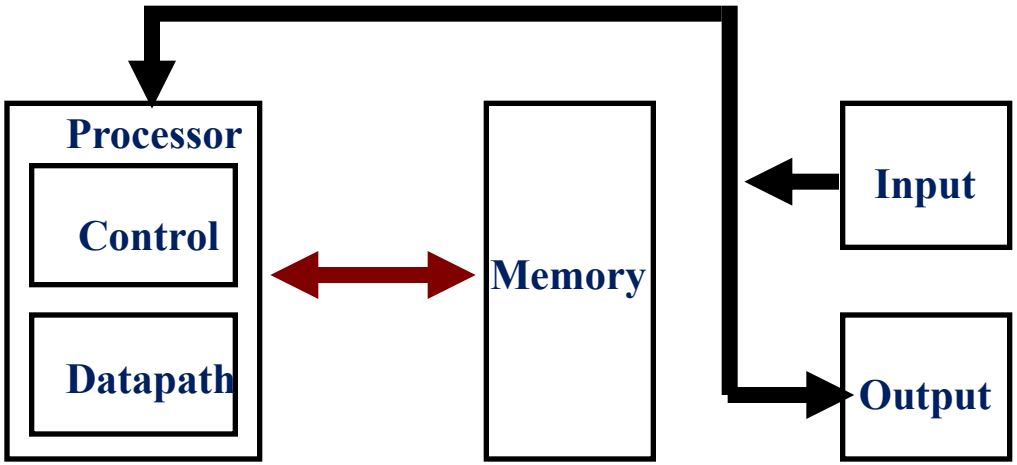
Input/Output (cont.)

- Standard I/O Protocols
  - PCIe
  - Serial ATA (SATA)
  - Serial Attached SCSI (SAS)
  - USB 2.0
  - Parallel ATA (PATA)

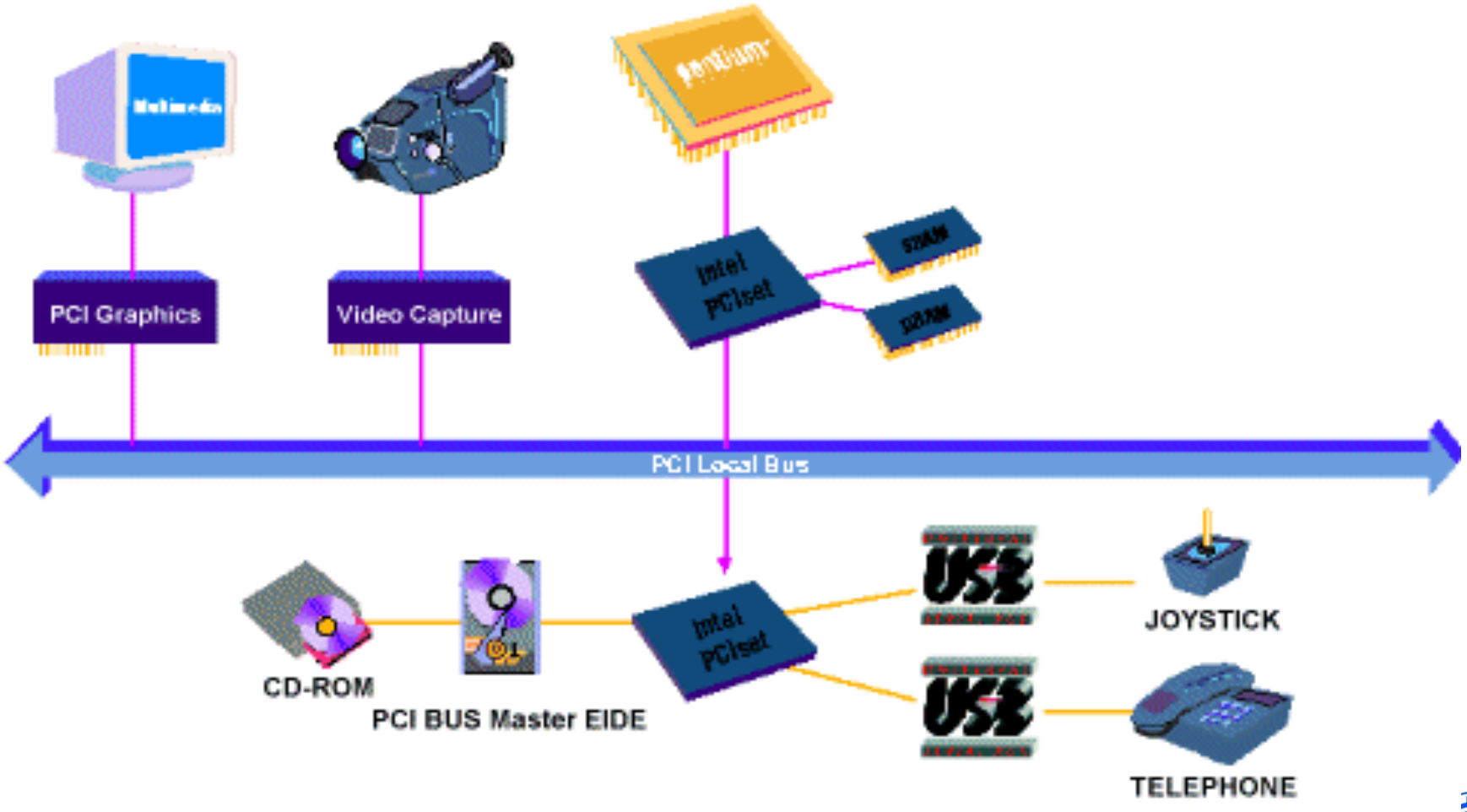
What is a Bus?

- Bus
  - Shared communication link
  - Single set of wires used to connect multiple subsystems

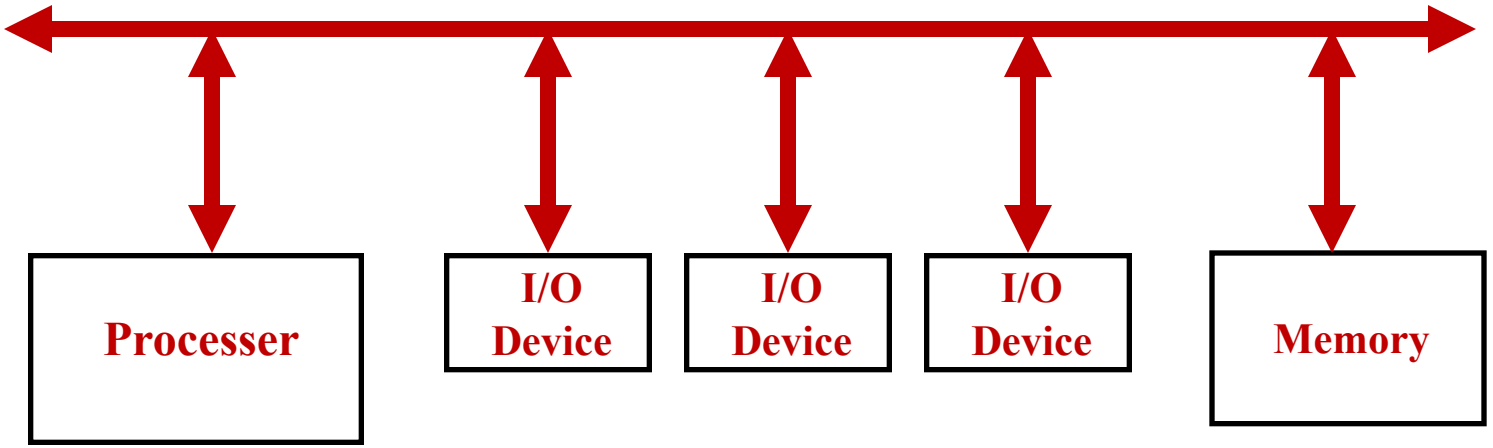




Buses

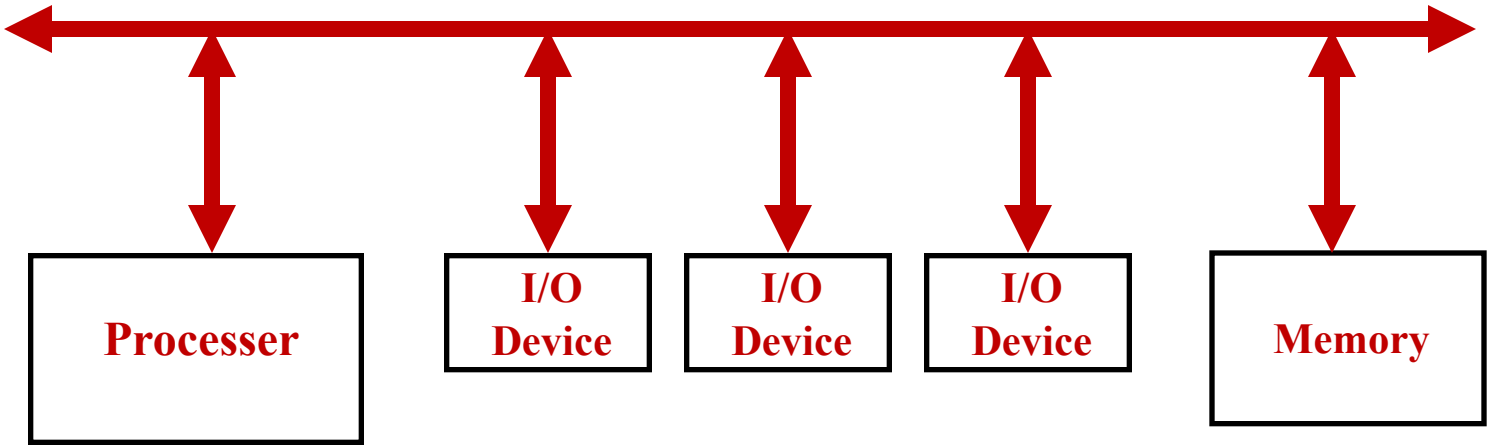

PCI Graphics
Video Capture
Intel PCIset
PCI Local Bus
CD-ROM
PCI BUS Master EIDE
Intel PCIset
JOYSTICK
TELEPHONE

Advantages of Buses



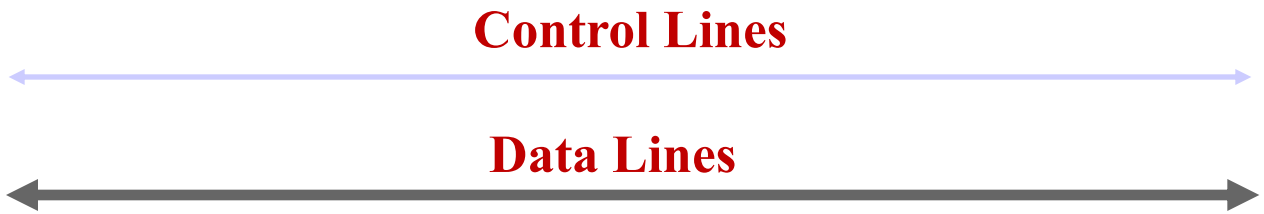
- Versatility
  - New devices can be added easily
  - Peripherals can be moved between computer systems that use same bus standard
- Low Cost
  - A single set of wires shared in multiple ways

Disadvantage of Buses



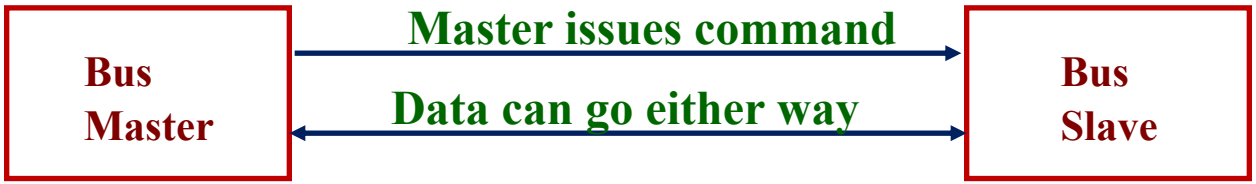
- Creates a communication bottleneck
  - Bus bandwidth limits maximum I/O throughput
- Maximum bus speed largely limited by:
  - Length of bus
  - Number of devices on bus
  - Slowest device on bus

General Organization of Buses



- Control Lines
  - Signal requests and acknowledgments
  - Indicate what type of information is on data lines
- Data Lines
  - Carry information between source and destination
  - Data and Addresses
  - Complex commands

Master versus Slave



- A Bus Transaction Includes Two Parts:
  - Issuing command (and address) – request
  - Transferring data – action
- Master: one who starts bus transaction by:
  - Issuing command (and address)
- Slave: one who responds to address by:
  - Sending data to master (M) if M asks for data
  - Receiving data from M if M wants to send data

Types of Buses

- Processor-Memory Bus
- I/O Bus
- Backplane Bus

Types of Buses

- Processor-Memory Bus
  - Design specific
  - Short and high speed
  - Only need to match memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to processor
  - Optimized for cache block transfers

Types of Buses

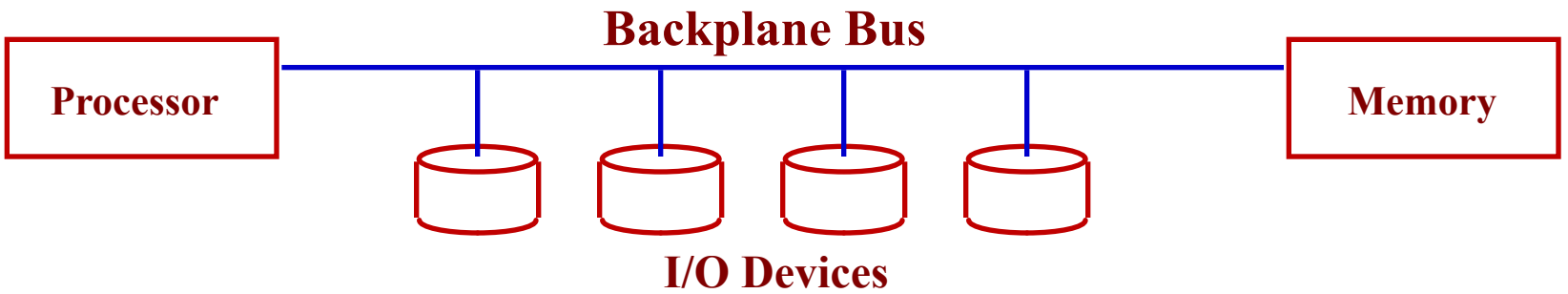
- I/O Bus
  - Industry standard
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to processor-memory bus or backplane bus

Types of Buses

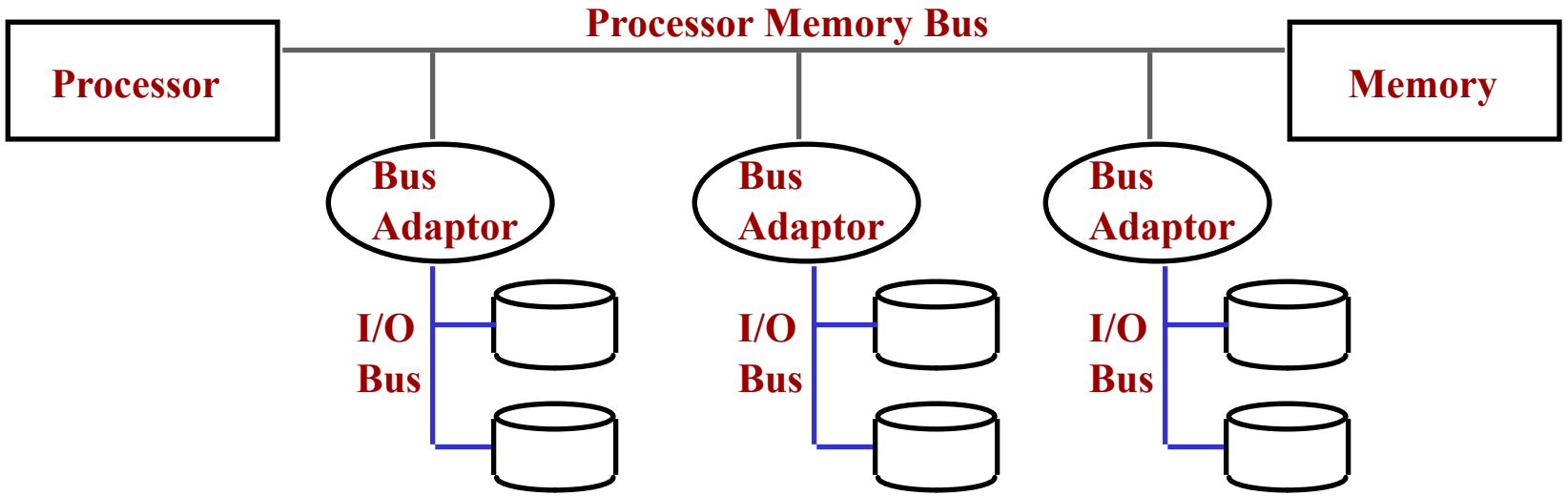
- Backplane Bus
  - Standard or proprietary
  - A single bus used for:
    - Processor to memory communication
    - Communication between I/O devices and memory
  - Backplane is an interconnection structure within chassis
  - Allow processors, memory, and I/O devices to coexist

Types of Buses

- Backplane Bus
  - Advantages: Simple and low cost
  - Disadvantages: slow and bus can become a major bottleneck
  - Example: IBM PC - AT



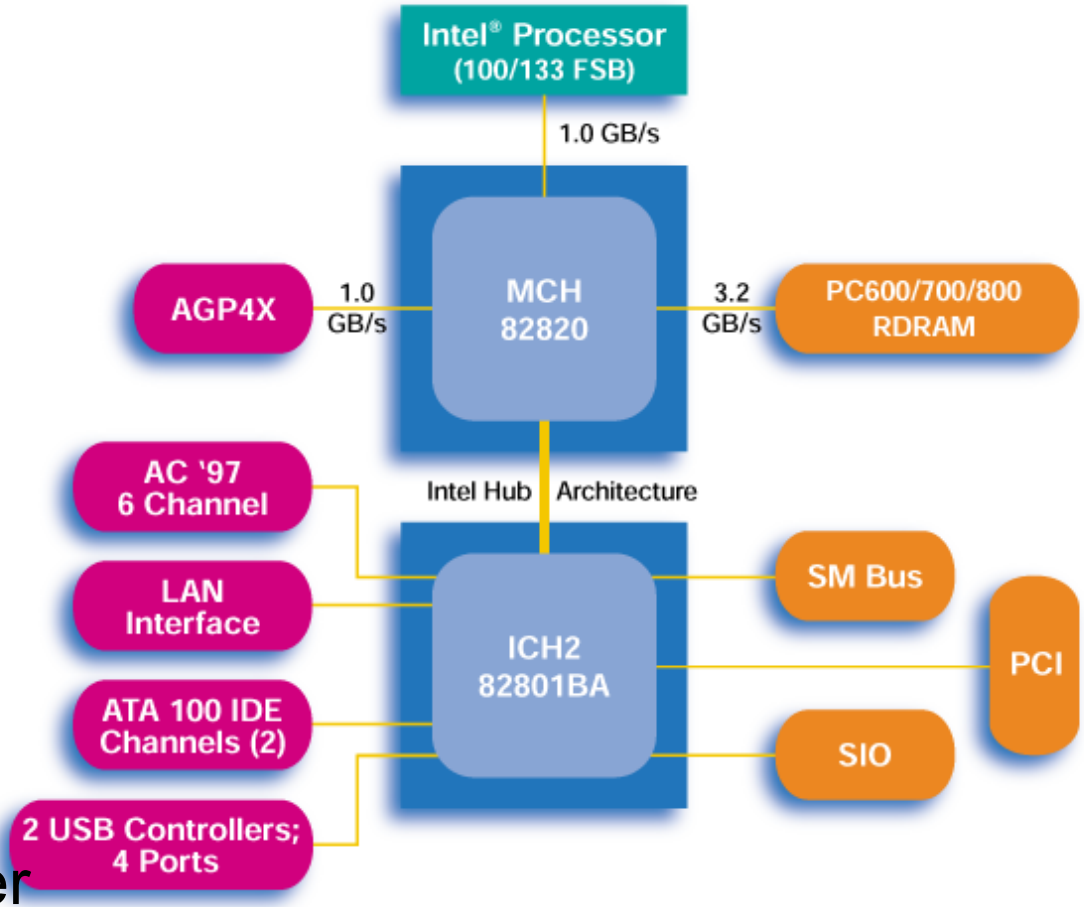
Two-Bus System



- I/O buses tap into processor-memory bus via bus adaptors:
- I/O buses provide expansion slots for I/O devices
- Example: Apple Macintosh-II

Main components of Intel Chipset: Pentium II/III

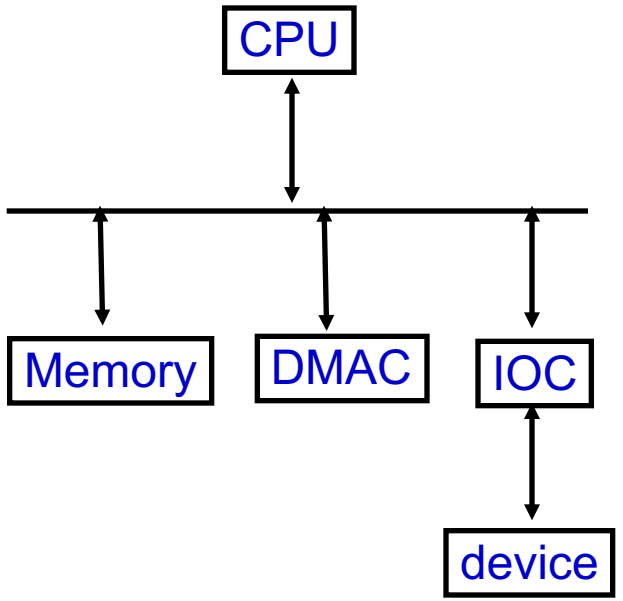
- Northbridge:
  - Handles memory
  - Graphics
- Southbridge: I/O
  - PCI bus
  - Disk controllers
  - USB controllers
  - Audio
  - Serial I/O
  - Interrupt controller
  - Timers



Delegating I/O Responsibility from CPU: DMA

- Direct Memory Access (DMA):
  - External to CPU
  - Act as a maser on bus
  - Transfer blocks of data to or from memory without CPU intervention

CPU sends a starting address, direction, and length count to DMAC. Then issues "start".



DMAC provides handshake signals for Peripheral Controller, and Memory Addresses and handshake signals for Memory.

Bus Transactions

- Transaction Phases
  - Master requests ownership from arbiter
  - Arbiter grants ownership to master
  - Master drives address for all to see
  - A slave claims transaction
  - Master (or slave) drives data (depending on read or write) for all to see
  - Master terminates transaction and bus ownership

Basic Bus Signals

- CLK
  - All devices synchronized by a clock signal
- Private Signals to/from Arbiter per Master
  - **REQ** (output): assert to request ownership; de-assert to signal end of transaction
  - **GNT** (input): ownership is granted

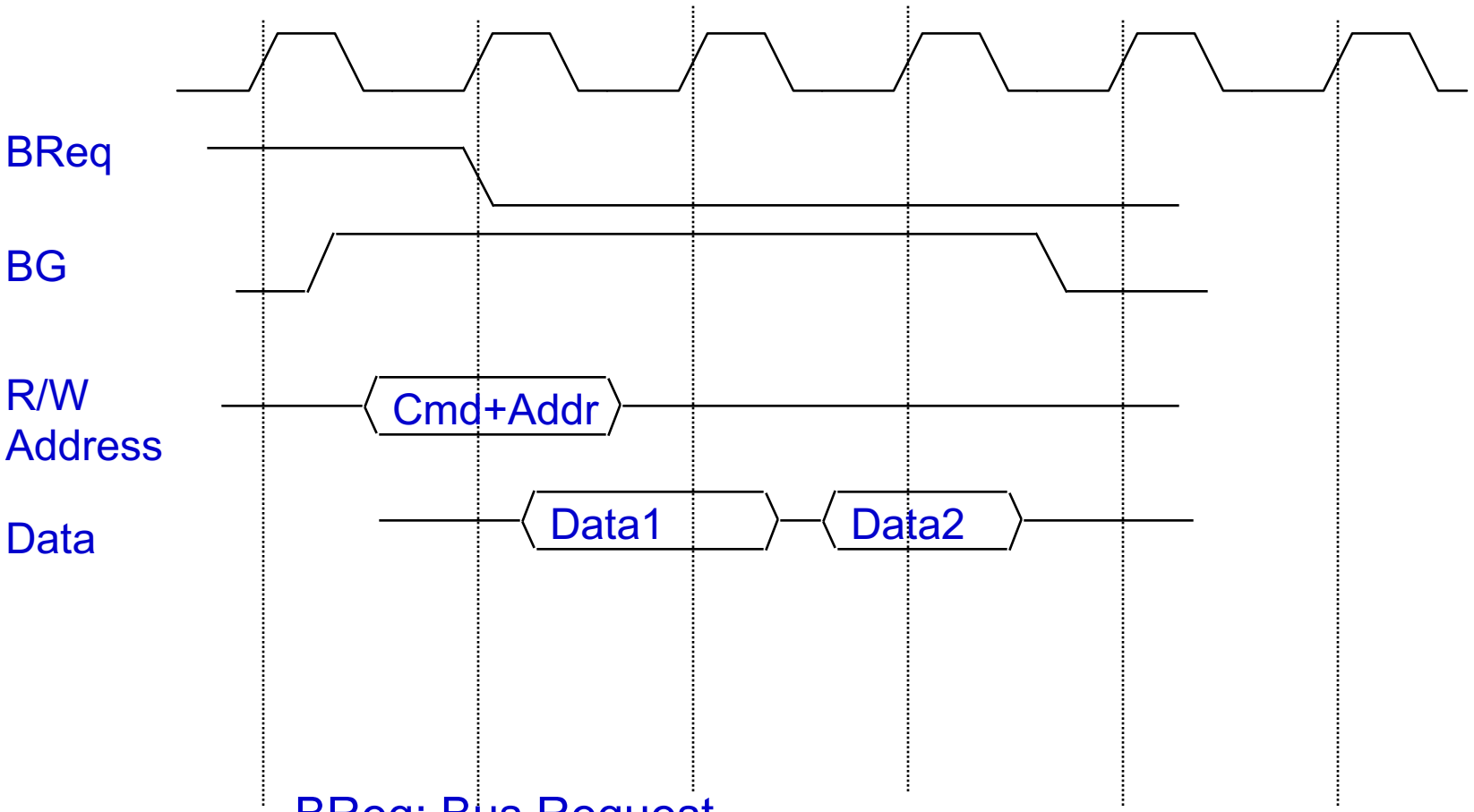
Basic Bus Signals

- "Broadcast" signals shared by all devices
  - **AD** (address/data bus, bi-directional): master drives address during the address phase, master/slave drives data during the data phase
  - Why not have separate address bus and data bus?
- R/W (bi-directional):
  - Bus commands, e.g. read vs. write

Synchronous Bus

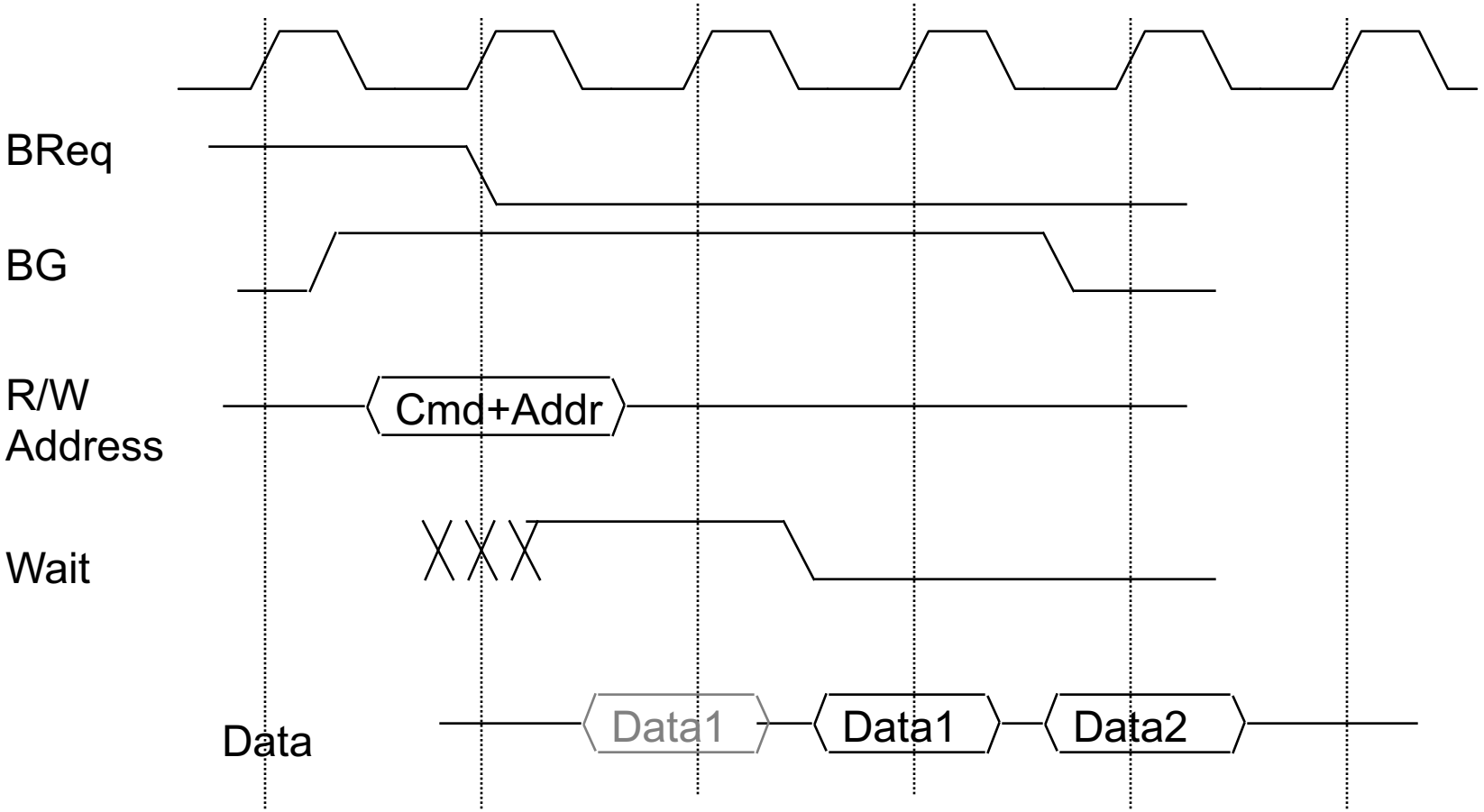
- **Synchronous** Bus:
  - Includes a clock in control lines
  - A fixed protocol relative to the clock
  - Advantage: little logic and very fast
  - Disadvantages:
    - Every device on the bus must run at the same clock rate
    - To avoid clock skew, they cannot be long if they are fast

Simple Synchronous Protocol



BReq: Bus Request
BG: Bus Grant

Typical Synchronous Protocol

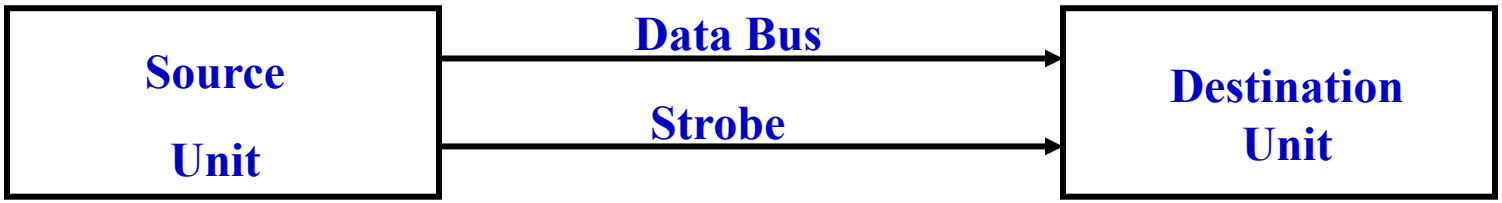


- Slave indicates when it is prepared for data xfer
- Actual transfer goes at bus rate

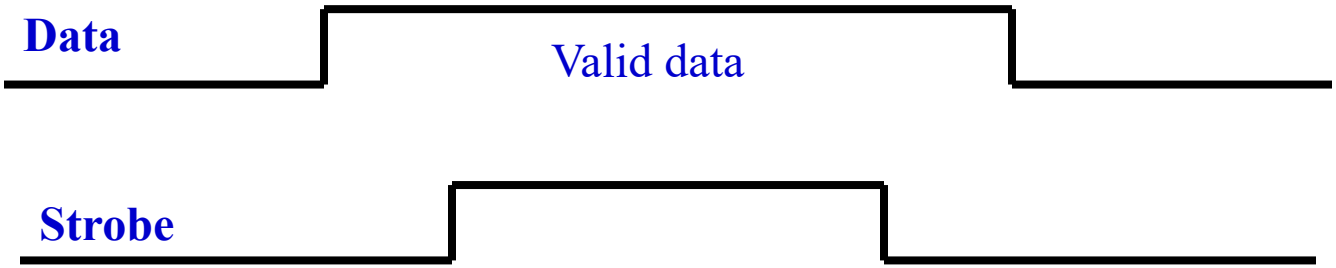
Asynchronous Bus

- **Asynchronous** Bus:
  - It is not clocked
  - It can accommodate a wide range of devices
  - It can be lengthened without worrying about clock skew
  - It requires a handshaking protocol

Data Transfer Initiated by Source Unit



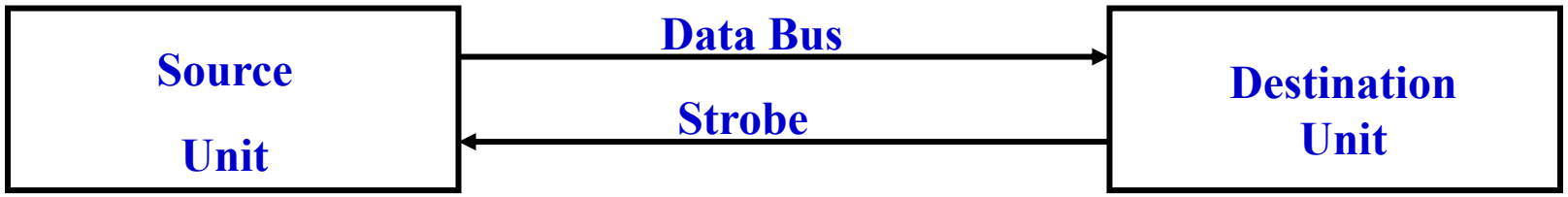
(a) **Block Diagram**



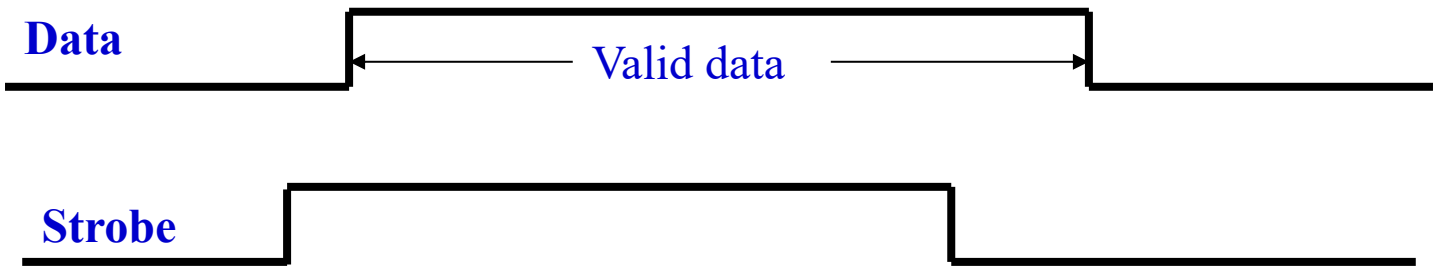
(b) **Timing Diagram**

Source-Initiated strobe for Data Transfer

Data Transfer Initiated by Destination Unit



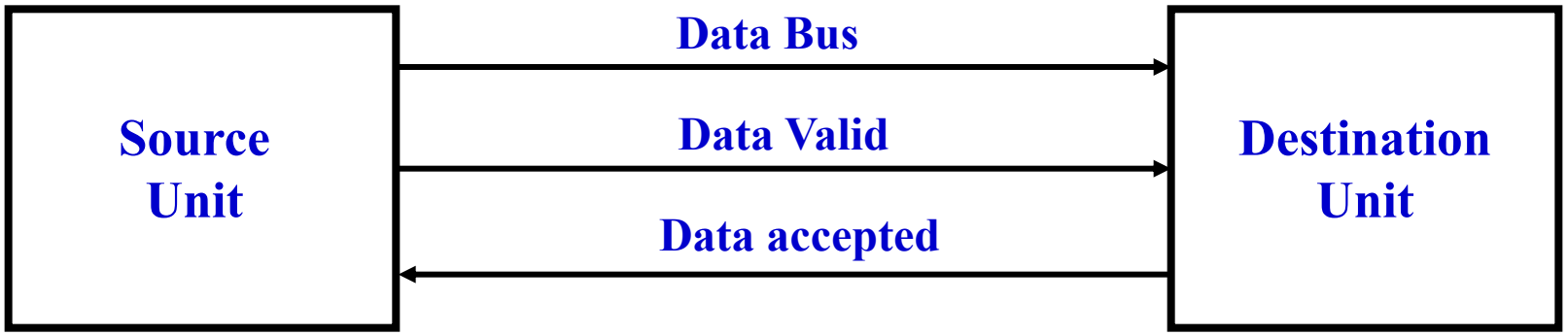
(a) **Block Diagram**

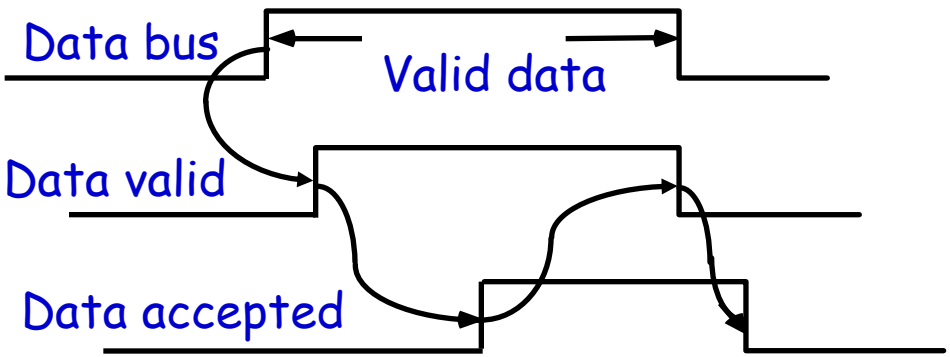


(b) **Timing Diagram**

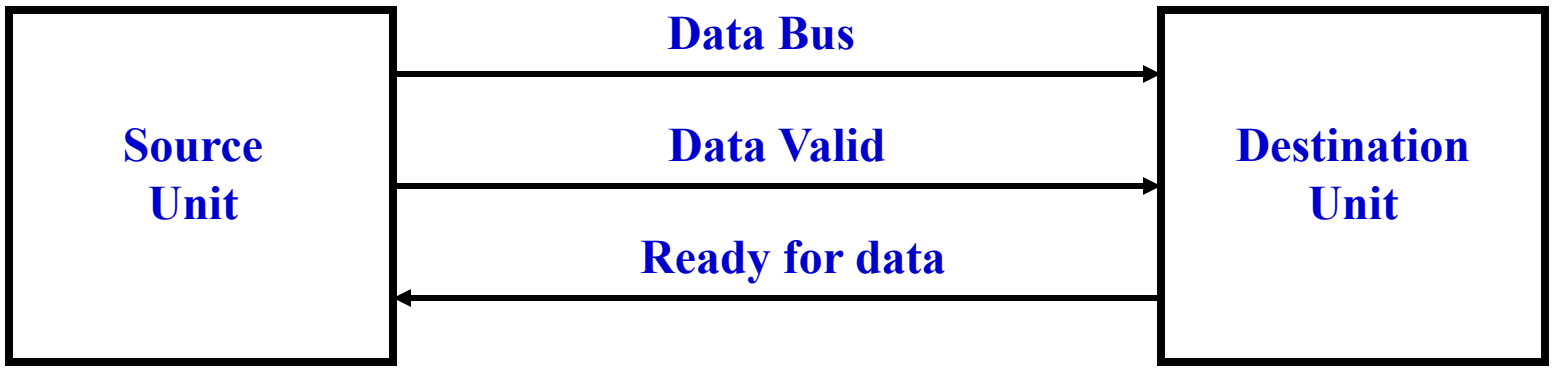
Destination-Initiated strobe for Data Transfer

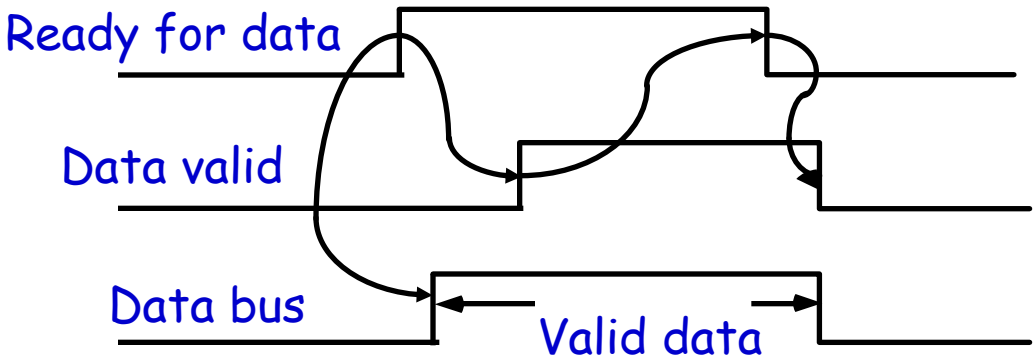
Source-Initiated Transfer Using Handshake



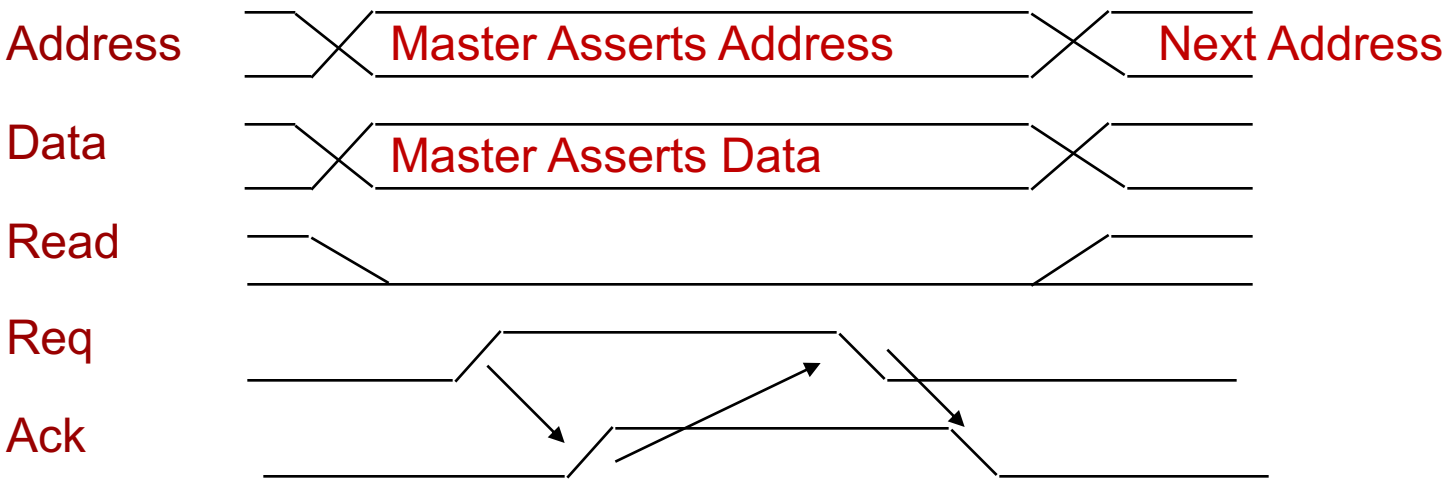


Destination-Initiated Transfer Using Handshake





Asynchronous Write Transaction



t0 : Master has obtained control and asserts address, direction, data
Waits a specified amount of time for slaves to decode target.

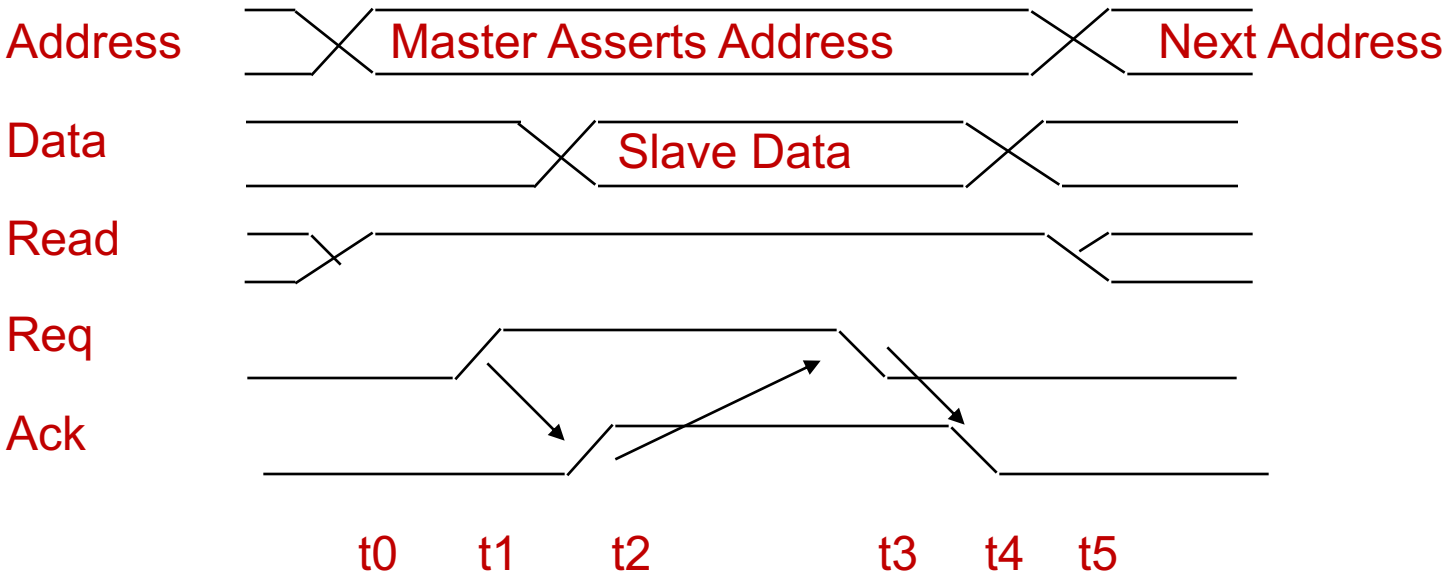
t1: Master asserts request line

t2: Slave asserts ack, indicating data received

t3: Master releases req

t4: Slave releases ack

Asynchronous Read Transaction



t0 : Master has obtained control and asserts address, direction, data
Waits a specified amount of time for slaves to decode target.

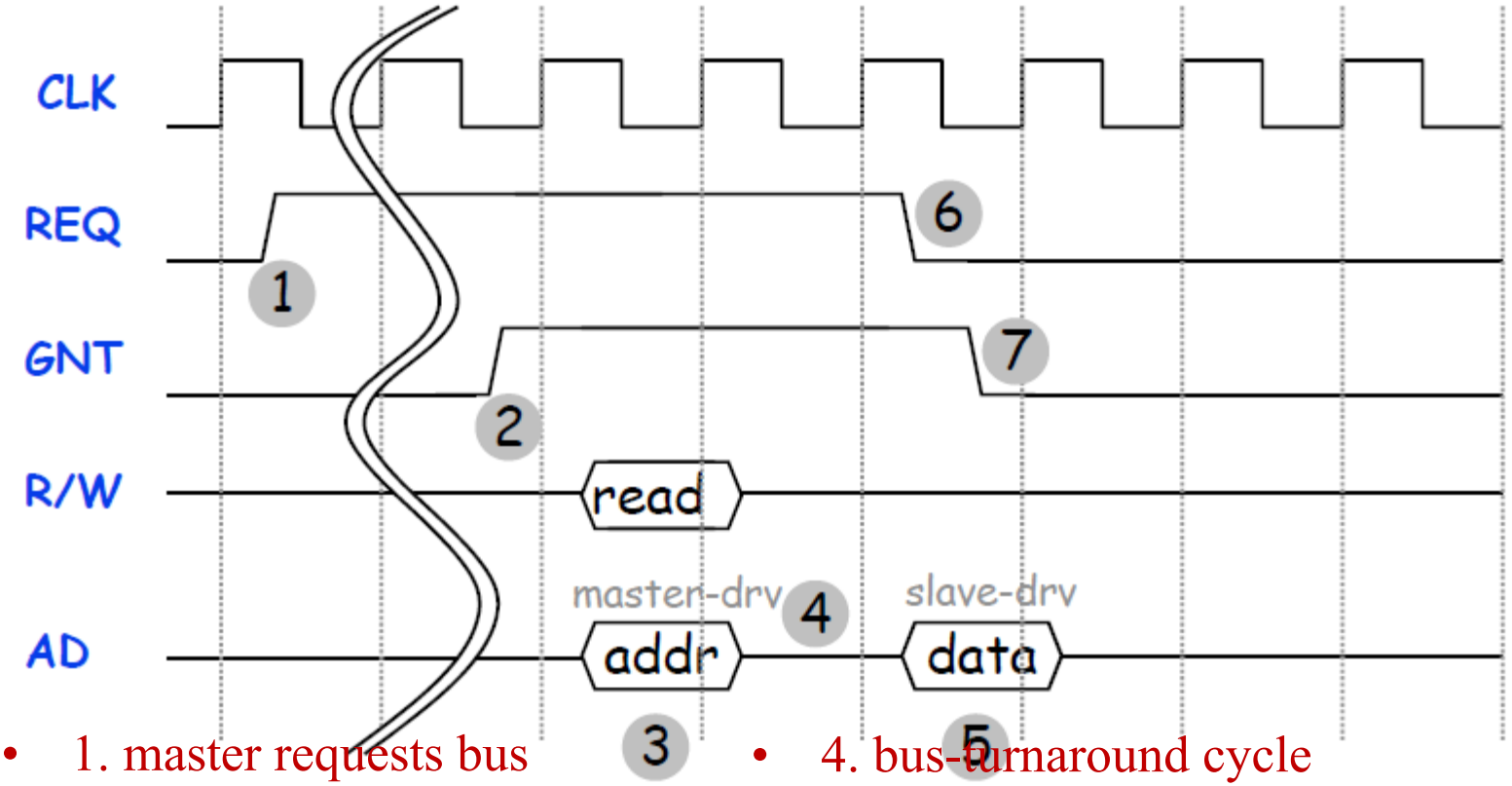
t1: Master asserts request line

t2: Slave asserts ack, indicating ready to transmit data

t3: Master releases req, data received

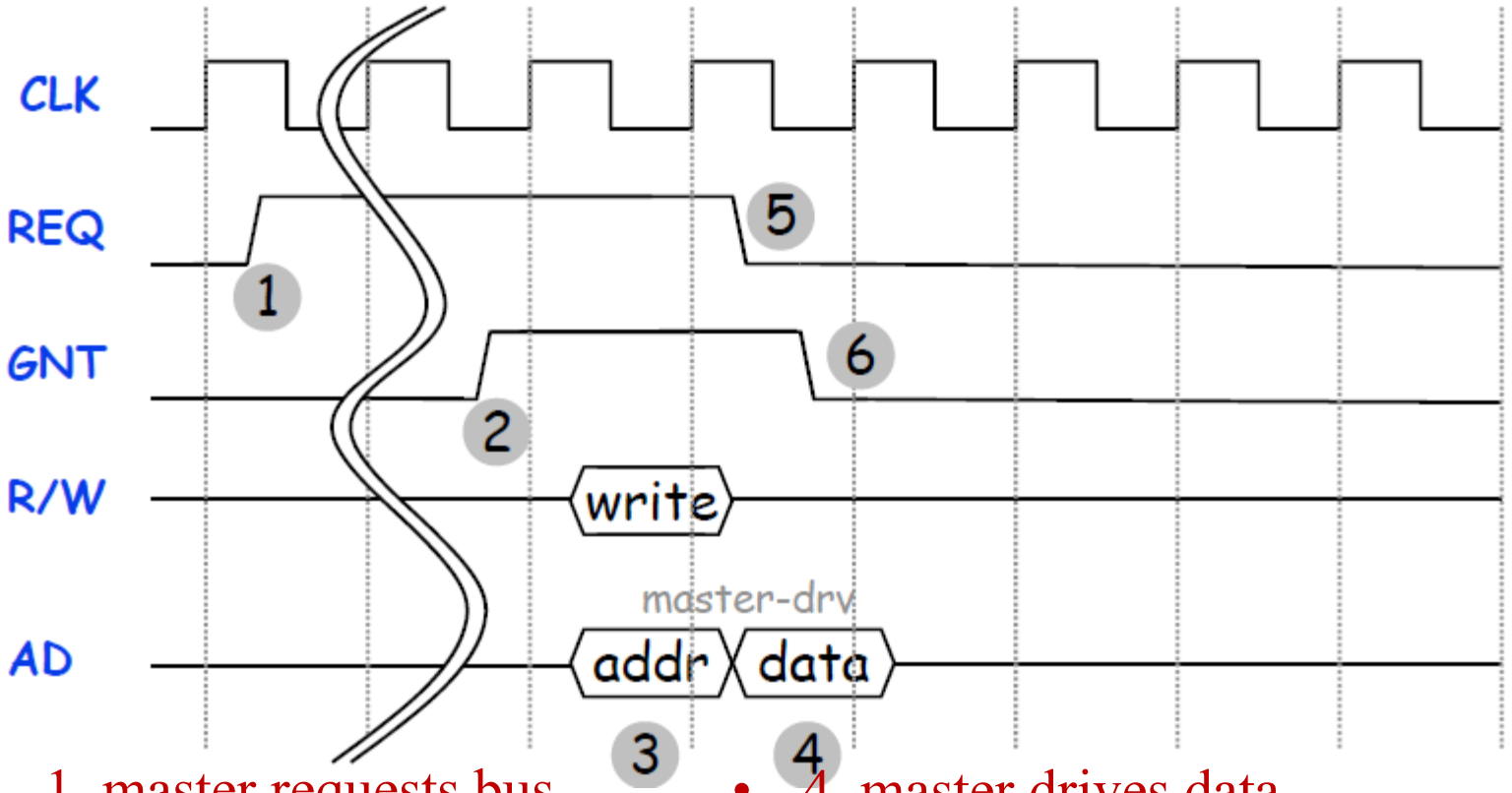
t4: Slave releases ack

Simple Read Transaction



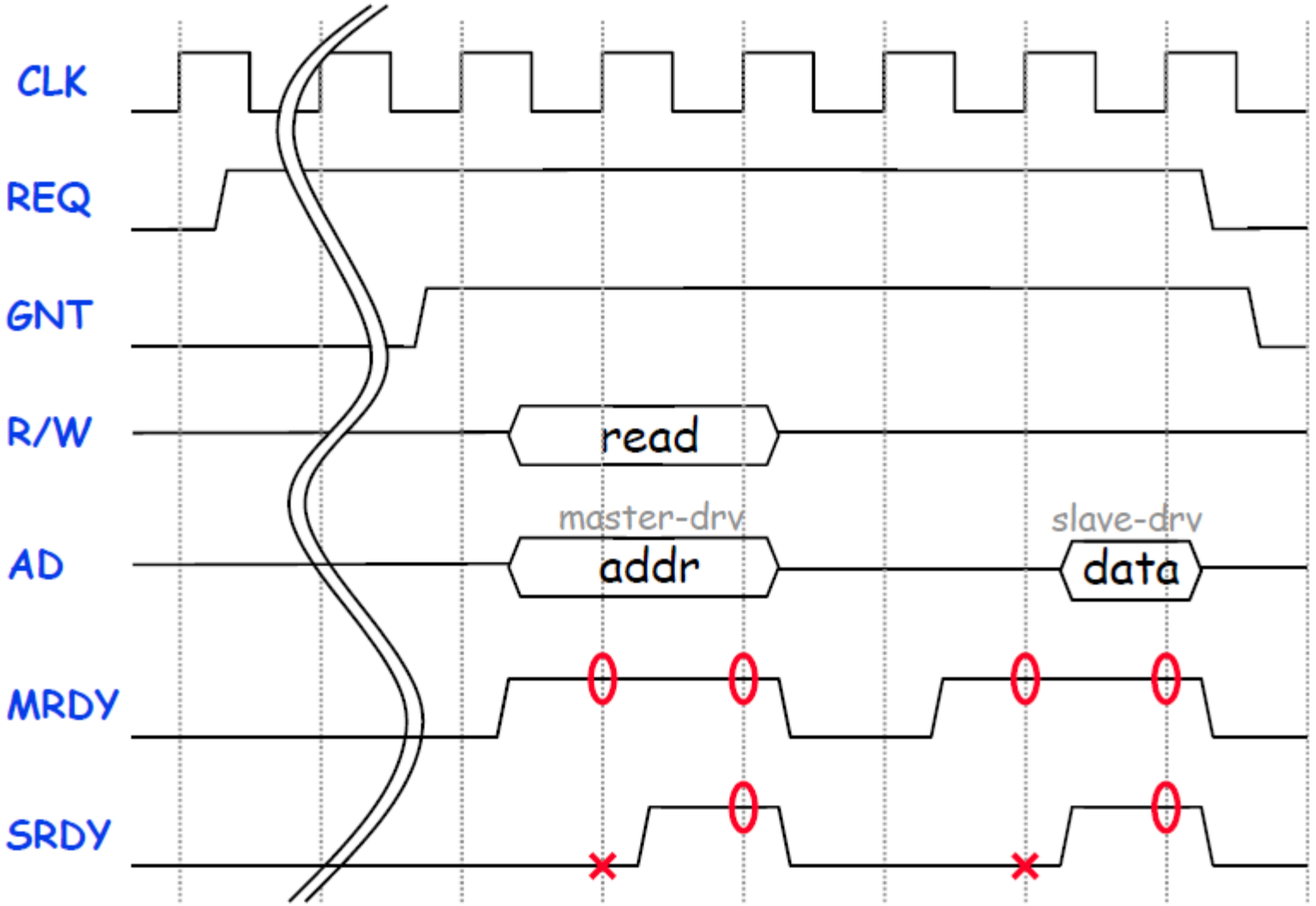
- 1. master requests bus
- 2. arbiter grants bus
- 3. master drives address /command
- 4. bus-turnaround cycle
- 5. slave drives data
- 6. master signals final cycle
- 7. arbiter acknowledges

Simple Write Transaction

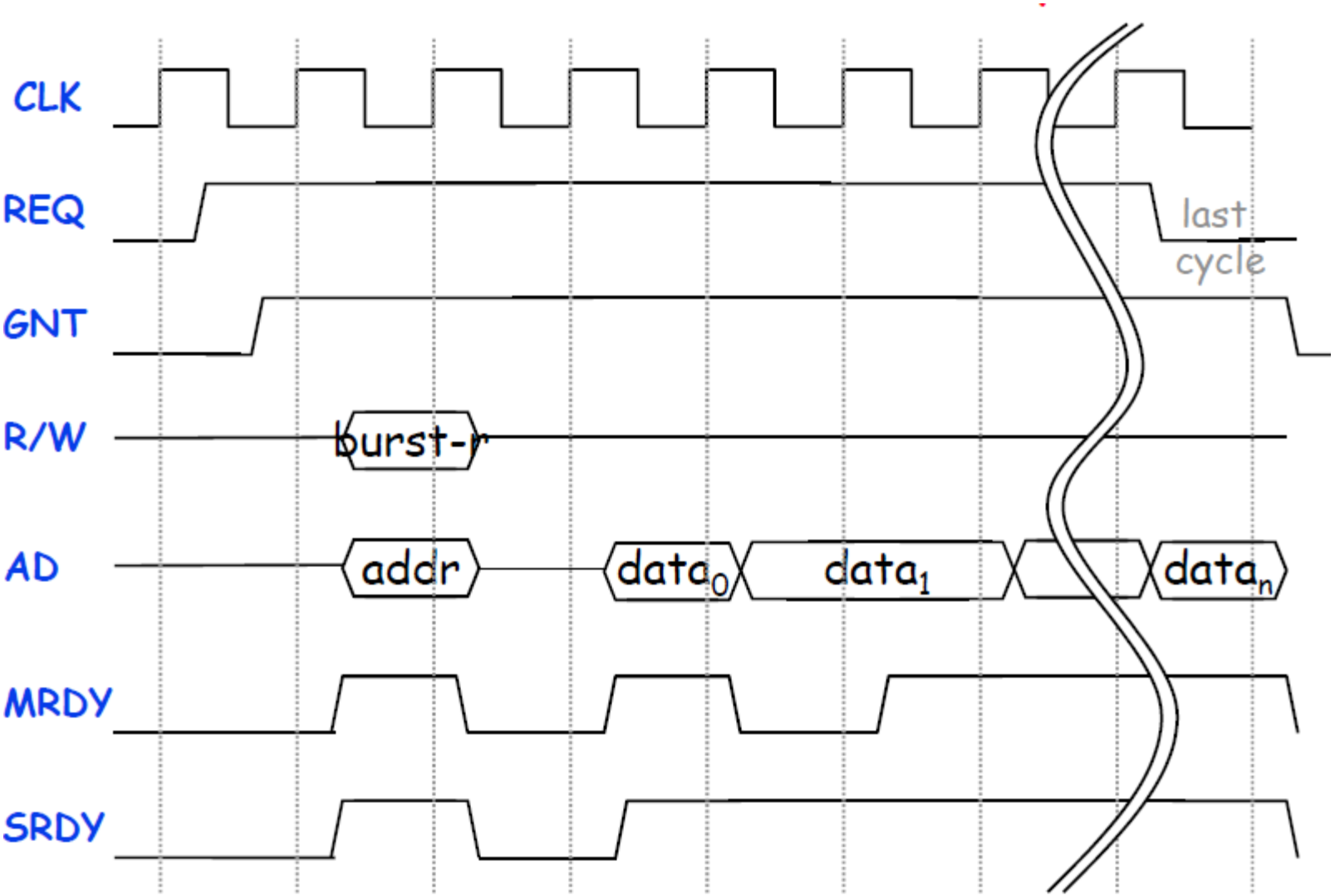


- 1. master requests bus
- 2. arbiter grants bus
- 3. master drives address /command
- 4. master drives data
- 5. master signals final cycle
- 6. arbiter acknowledges

Asynch Read Transaction



Burst Read Transaction

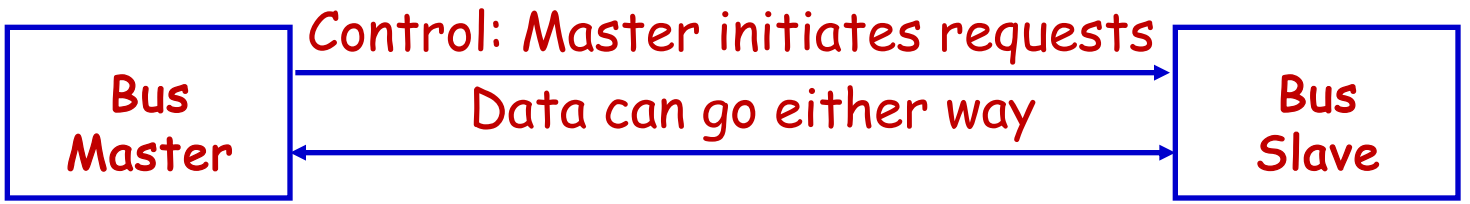


Bus Arbitration

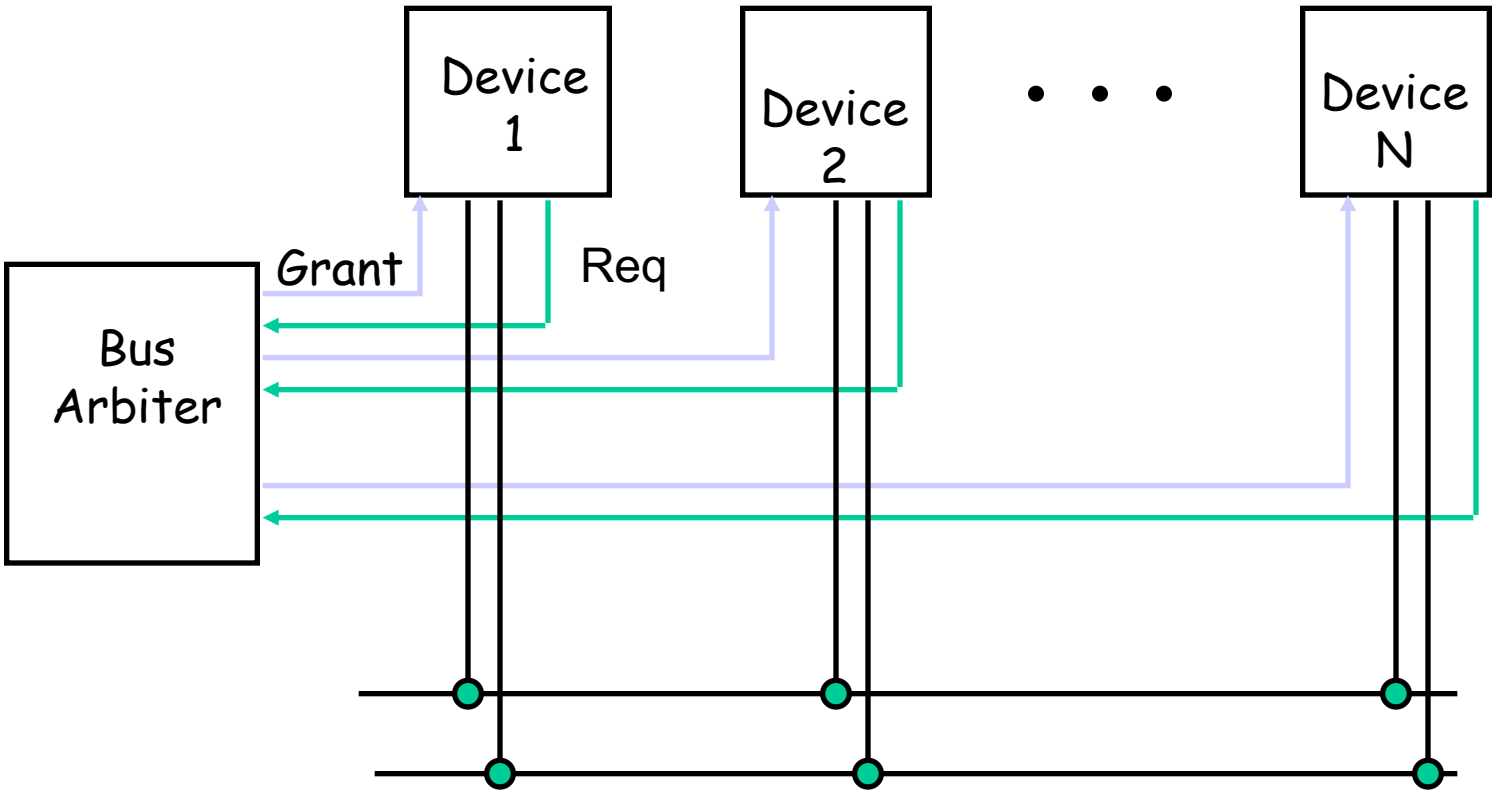
- Arbitration scheme:
  - A bus master wants to use the bus; asserts the bus request
  - A bus master cannot use the bus until its request is granted
  - A bus master must signal to the arbiter after finish using the bus
- Arbitration schemes balance two factors:
  - Bus priority: the highest priority device should be serviced first
  - Fairness: Even the lowest priority device should never be completely locked out from bus

Simple Bus Arbitration

- Master-slave arrangement:
  - Only bus master can control access to bus
    - It initiates and controls all bus requests
  - Slave responds to read/write requests
  - Example:
    - Processor is only bus master
    - All bus requests controlled by processor
    - Major drawback ?



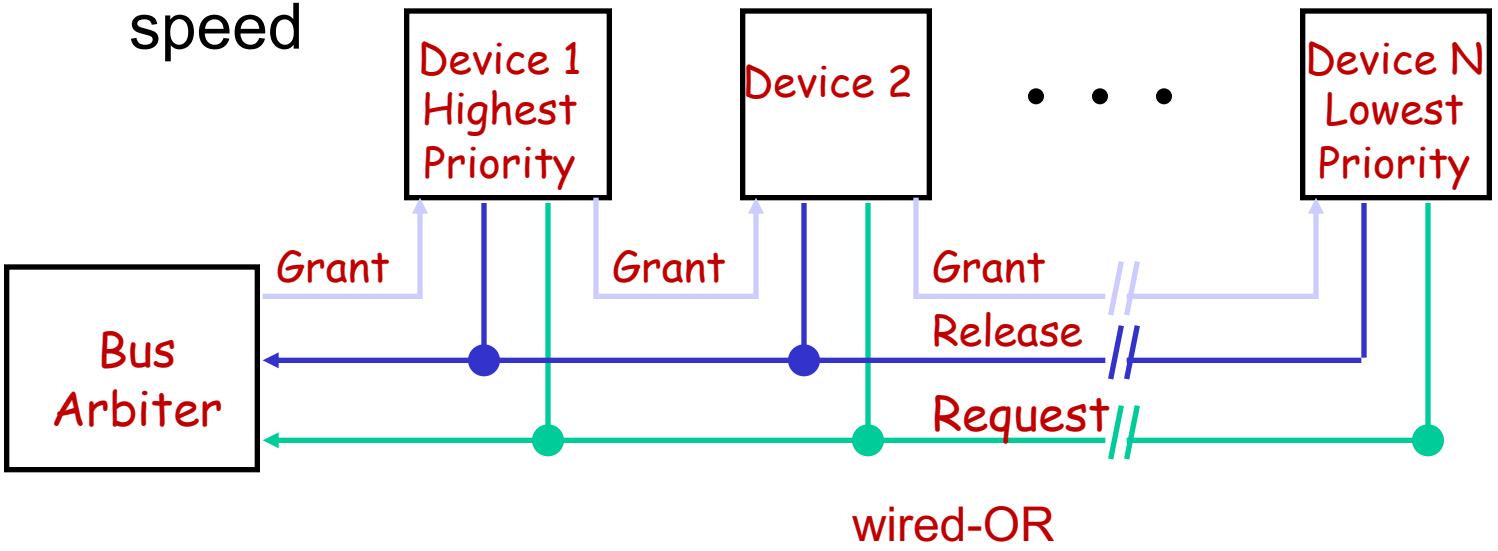
Centralized Parallel Arbitration



- Used in essentially all processor-memory busses and in high-speed I/O busses

Daisy Chain Bus Arbitrations Scheme

- Advantage: simple
- Disadvantages:
  - Cannot assure fairness:
    - A low-priority device may be locked out indefinitely
  - Use of daisy chain grant signal also limits bus speed



Increasing Bus Bandwidth

- Separate Address and Data Lines
- Data Bus Width
- Block Transfers

Increasing Bus Bandwidth

- Separate Address and Data Lines
  - vs. multiplexed address/data lines
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available
  - Cost:
    - More bus lines
    - Increased complexity

Increasing Bus Bandwidth

- Data Bus Width
  - By increasing width of data bus, transfers of multiple words require fewer bus cycles
  - Example: SPARCstation 20's memory bus is 128 bit wide
  - Cost: more bus lines

Increasing Bus Bandwidth

- Block Transfers (Bursts Transfer)
  - Allow bus to transfer multiple words in back-to-back bus cycles
  - Only one address sent at beginning
  - Bus is not released until the last word is transferred
  - Cost:
    - Increased complexity
    - Decreased response time for request

I/O System Characteristics

- Dependability
  - Particularly for storage devices
- Performance Measures
  - Latency (response time)
  - Throughput (bandwidth)
  - Desktops & embedded systems
    - Mainly interested in response time & diversity of devices
  - Servers
    - Mainly interested in throughput & expandability of devices

I/O Device Examples

Device	Behavior	Partner	Data Rate: KB/sec
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Line Printer	Output	Human	1.00
Floppy disk	Storage	Machine	50.00
Laser Printer	Output	Human	100.00
Optical Disk	Storage	Machine	500.00
Magnetic Disk	Storage	Machine	5,000.00
Network-LAN	Input/Output	Machine	20 – 1,000.00
Graphics Display	Output	Human	30,000.00

Thanks for Your Attention!