University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3202
Computer Organization & Architecture

The Buses

Buses

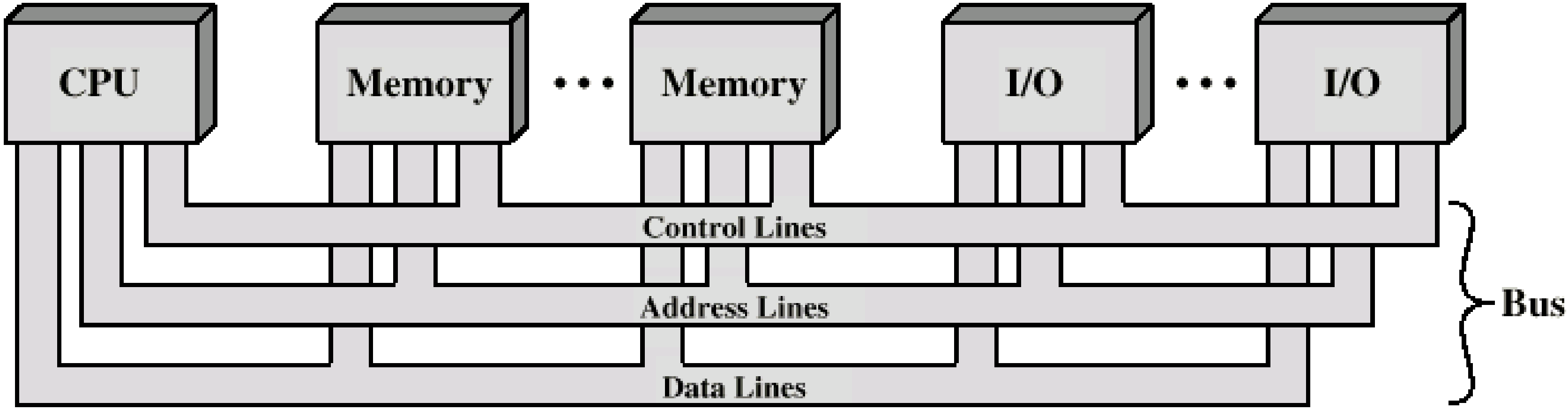
- CPU, Memory, and I/Os need **interconnection for communication**
- There are a number of possible interconnection systems
- Single and multiple **BUS** structures are most common
  - e.g. Control/Address/Data bus (PC)
  - e.g. Unibus (DEC-PDP)

What is a Bus?

- A **communication pathway** connecting multiple devices
- **Shared transmission medium**
- Only one device must use it at a time
- Consists of multiple lines that are often **grouped**
  - A number of **channels** in one bus
    - e.g. 32-bit bus = 32 single bit channels
  - Each line/channel can transmit signals representing 0 or 1

Bus Interconnection Scheme

- Called a **system bus** when used to connect major computer components (CPU, memory, I/O)
- **Three functional groups of communication lines:**
  - *Data bus, Address bus, Control bus*
- **Power lines** may not be shown



Data Bus

- **Moves data** around system modules
  - Remember that there is no difference between "*data*" and "*instruction*" at this level
- **Bus width** is a key determinant of **performance**
  - 8, 16, 32, 64-bit bus
  - Since each line can carry only one bit at a time, the **number of lines** determines how many bits can be transferred at a time.

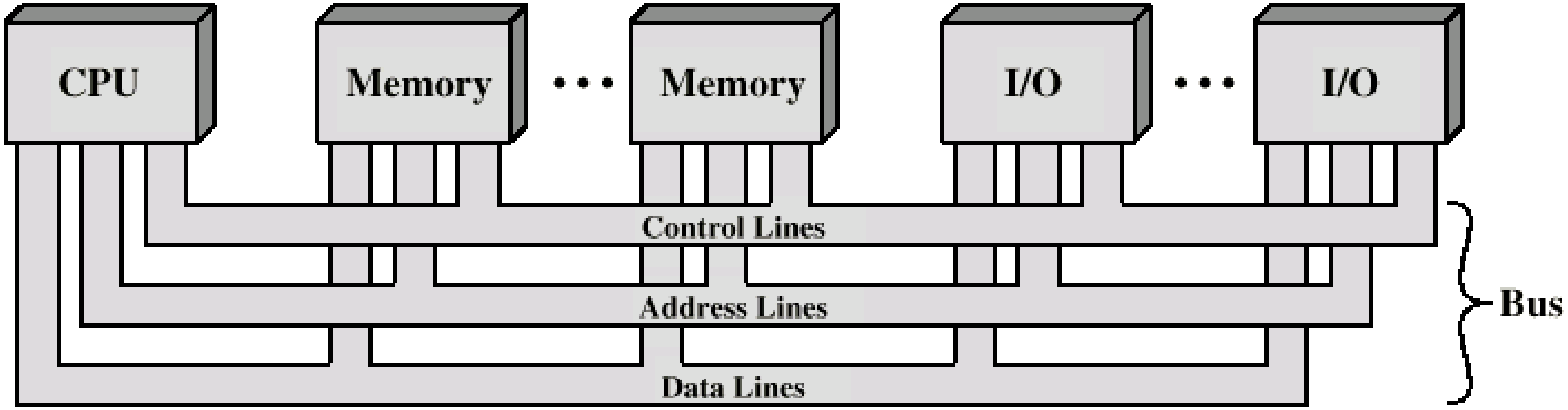
Address bus

- **Identifies the source or destination of data** on the data bus
  - e.g. CPU needs to read an instruction (*data*) from a given location in memory
- **Bus width** determines **maximum memory capacity** of system
  - 2^k memory locations, *k* address lines
    - 4 address lines: $2^4 = 16$ memory locations
  - e.g. 8080 has 16-bit address bus: 64k address space
- Also used to address I/O ports.

Control Bus

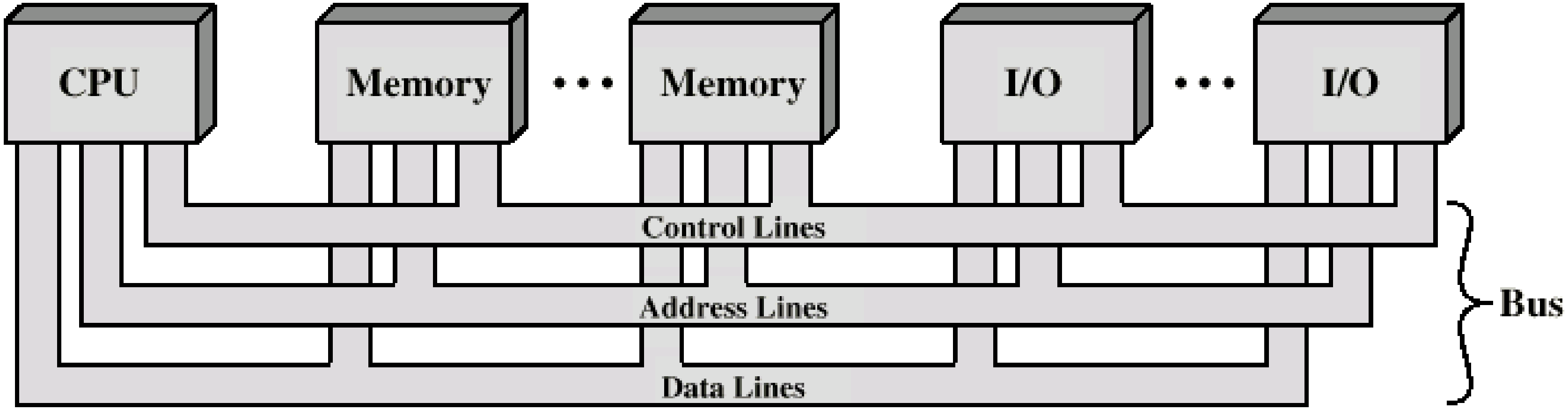
- **Controls access to** and **use of the data and address lines**
- Contains **control and timing signals**
  - **Control signals** contain **command and timing** information
  - **Timing signals** indicate the **validity** of data and address information
- Some typical control lines:
  - Memory read/write signal
  - Interrupt request
  - Clock and reset signals

Bus Operations



1. If one module wishes to **send data** to another, it must:
   - Obtain use of the bus
   - Transfer data via the bus

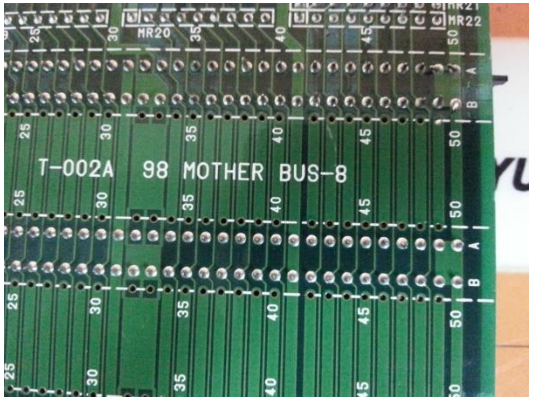
Bus Operations



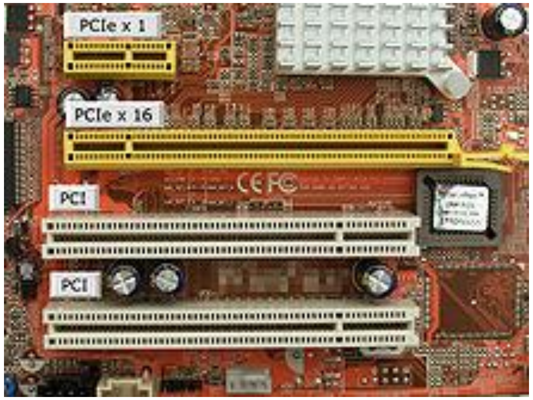
2. If one module wishes to **request data** from another, it must:
   - Obtain use of the bus
   - Transfer a request to the other module over control and address lines
   - Wait for second module to send data

What do buses look like?

- Parallel lines on circuit boards
- Ribbon cables
- Strip connectors on motherboards (e.g. PCI)
- Sets of wires



https://www.yuyiplc.com/products/26912/image/T-002A%2098%20MOTHER%20BUS-8%20PCB%20BOARD_3.jpg



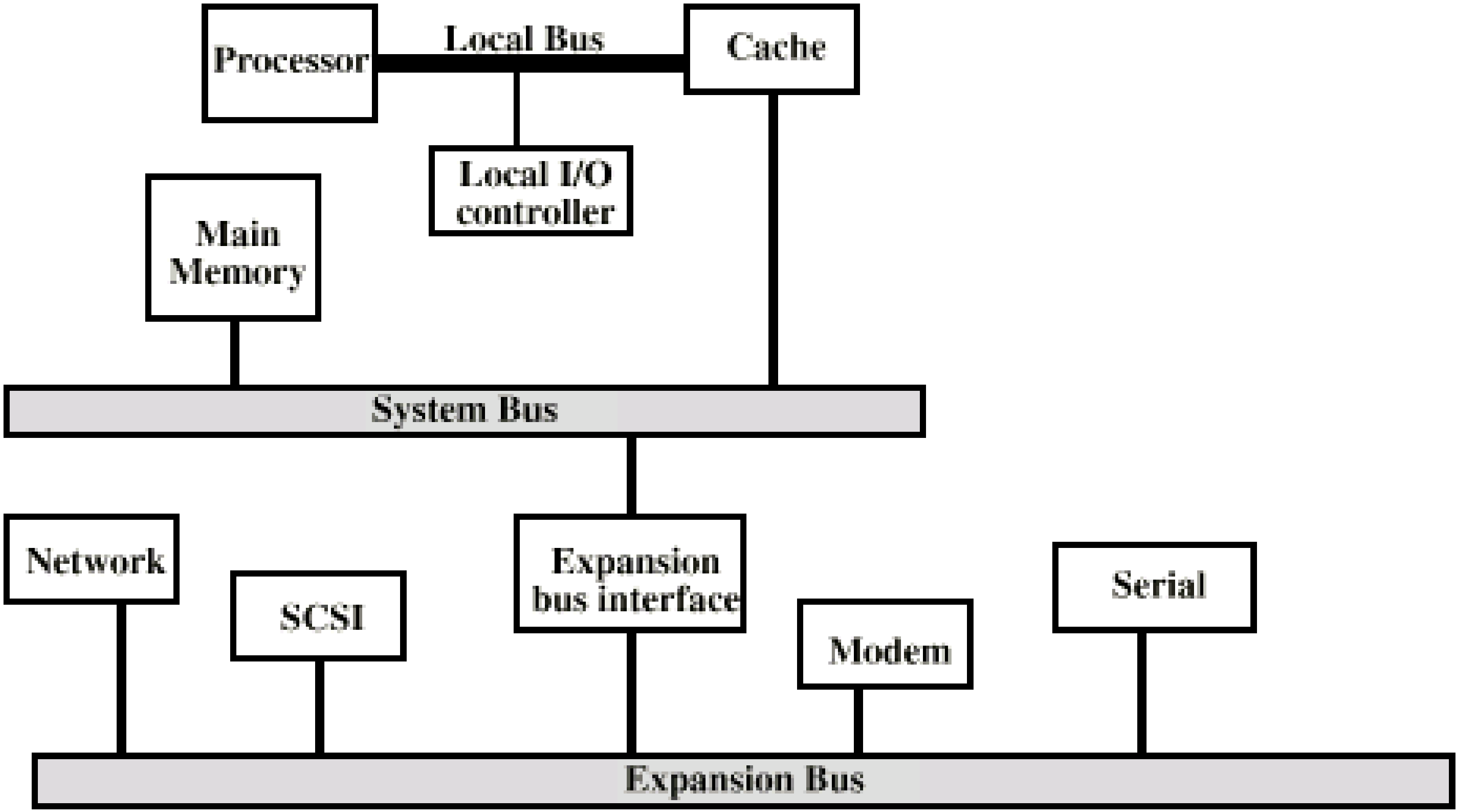
https://en.wikipedia.org/wiki/File:PCI_und_PCIe_Slots.jpg

https://thumbs.dreamstime.com/z/computer-bus-11062661.jpg

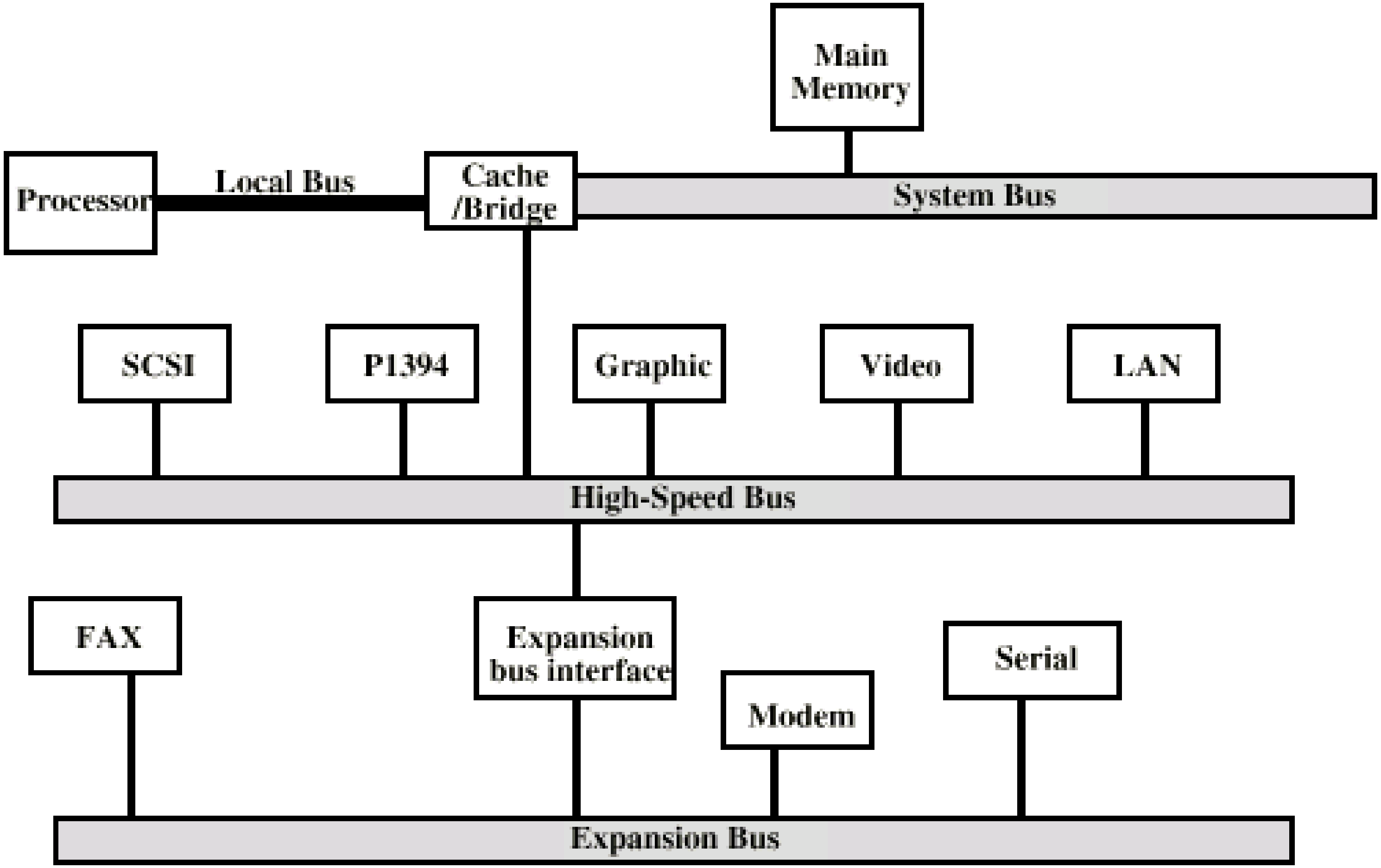
Single Bus Problems

- Lots of devices on one bus leads to **poor performance**
- **Main causes:**
  - **Propagation delays**
    - *Time it takes for devices to coordinate the use of the bus*
    - Long data paths mean that coordination of bus use can adversely affect performance
  - The bus may become a **bottleneck** as the aggregate data transfer demand approaches the capacity of the bus (*in available transfer cycles/second*)
- Most systems use **multiple buses** to overcome these problems

Traditional Bus Architecture (with Cache)



High-Performance Bus Architecture



Bus Types

- Dedicated
  - Separate data & address lines
- Multiplexed
  - Shared lines
  - Address valid or data valid control lines
  - Advantage
    - Fewer lines *(saves space and cost)*
  - Disadvantages
    - More complex control circuitry for each module
    - Potential reduction in performance

Bus Arbitration

- More than one module controlling the bus
  - e.g. CPU and DMA controller
- **Only one module** may control bus at one time
  - Because only one unit at a time can successfully transmit over the bus, some **method of arbitration** is needed.
- Arbitration
  - Determines who can use the bus at a particular time
  - May be **centralized** or **distributed**

Centralized Arbitration

- Single hardware device **controlling bus access**
  - Bus Controller
  - Arbiter
- Responsible for **allocating time** on the bus
- May be part of CPU or a separate module

Distributed Arbitration

- No central controller
- Each module may claim the bus
- Control logic on all modules
- Modules act together to share the bus

Master and Slave

- Both arbitration methods **designate one device** (either CPU or an I/O module) on the bus as **master**
  - May initiate a data transfer with some other device, which acts as a **slave**

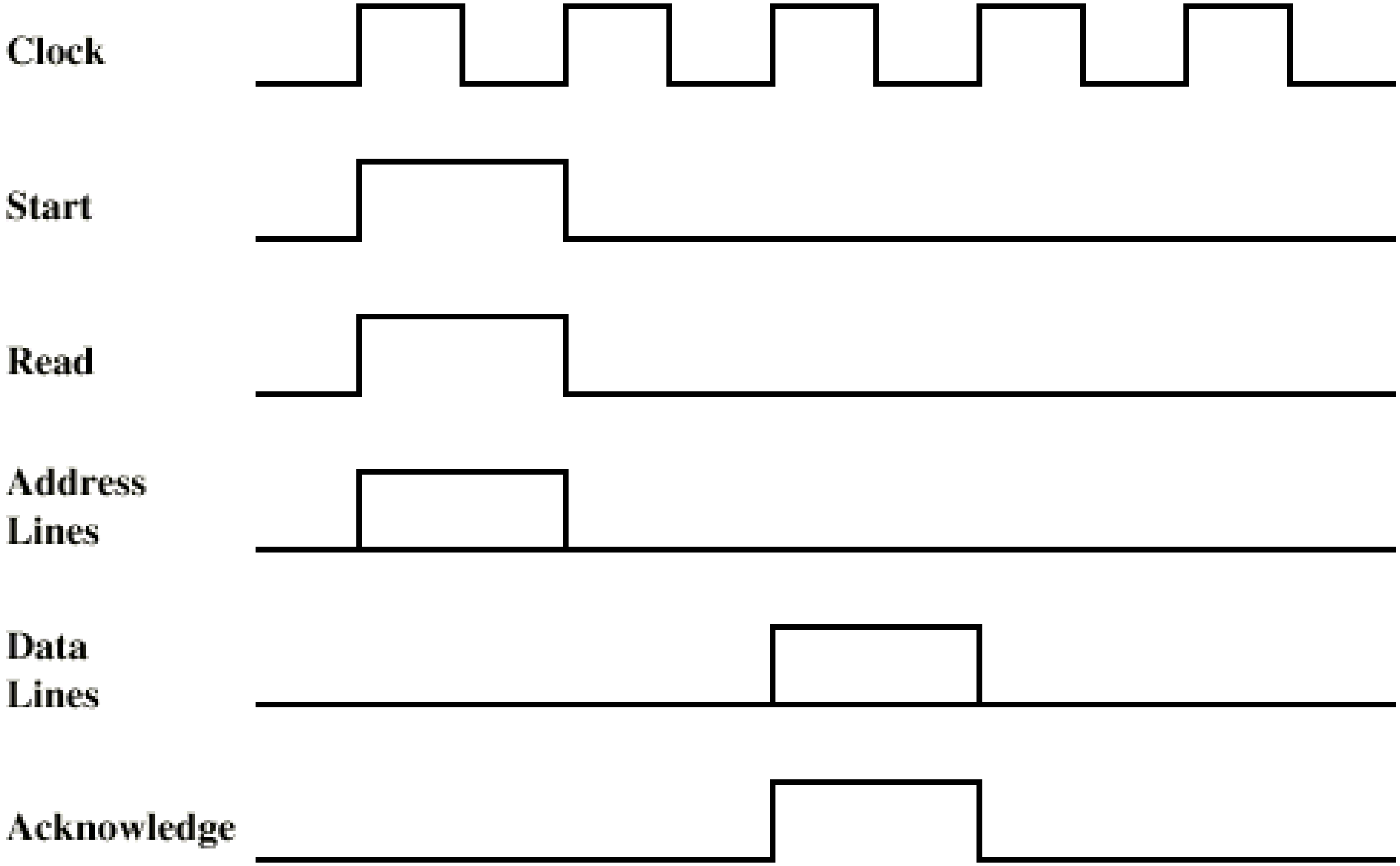
Timing

- **Coordination of events** on bus
- Buses use either **synchronous** or **asynchronous** timing

Synchronous Timing

- Events are determined by a **clock**
- Control Bus includes **clock line**
  - Transmits a **regular sequence** of alternating 1s and 0s of **equal duration**
- A single 1-0 is a ***clock cycle*** or a ***bus cycle***
- All devices **can read clock line**
- Usually **sync on leading edge**
- Usually a **single cycle for an event**

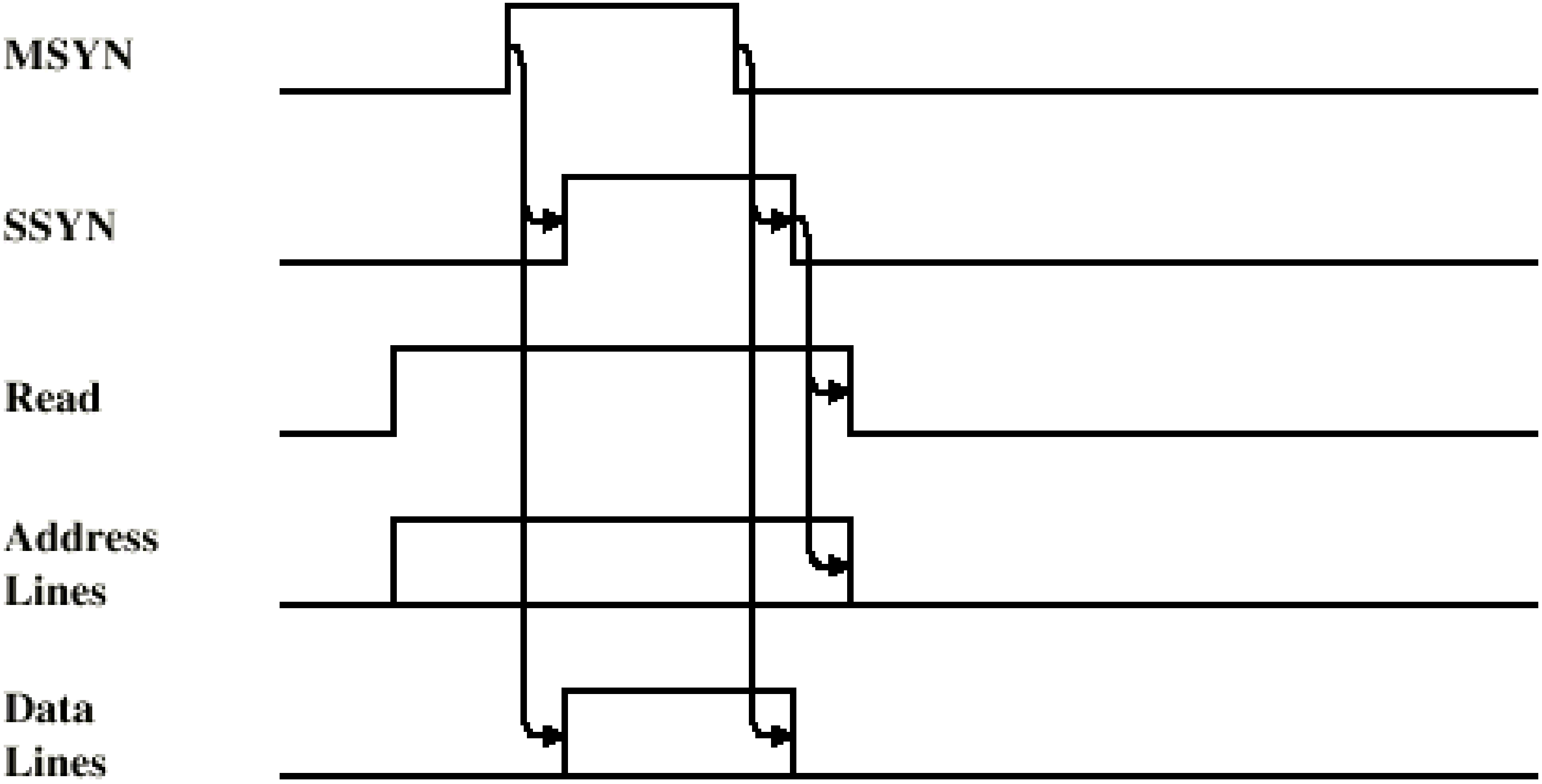
Synchronous Timing Diagram (Read)



Asynchronous Timing

- No clock line, no fixed time cycles
- An event on a bus **follows and depends on a previous event**
  - Uses *a cause and effect basis* to decide which signal gets changed next
- **Could be faster** than synchronous buses
  - Devices do not have to wait until a time unit ends to respond
  - Could share slow and fast devices (older and newer technology) on the same bus
- **But harder to implement and test**

Asynchronous Timing Diagram (Read)



PCI Bus

- Peripheral Component Interconnection
  - High-bandwidth
  - Processor-independent
  - Can function as a **mezzanine** or **peripheral bus**
- Delivers **better system performance** for **high-speed I/O subsystems**
  - e.g., graphic display adapters, network interface controllers, and disk controllers

PCI Bus

- Intel released to **public domain**
- Requires few chips to implement
- Supports other buses attached to PCI bus
- Supports a variety of microprocessor-based configurations, including multiple processors
- Uses **synchronous timing** and **centralized arbitration**
- 32 or 64-bit
- 50 lines

PCI Bus Lines (Required)

- Systems lines
  - Including clock and reset
- Address & Data
  - 32 time mux lines for address/data
  - Interrupt & valid lines
- Interface Control
- Arbitration
  - Not shared
  - Direct connection to PCI bus arbiter
- Error lines

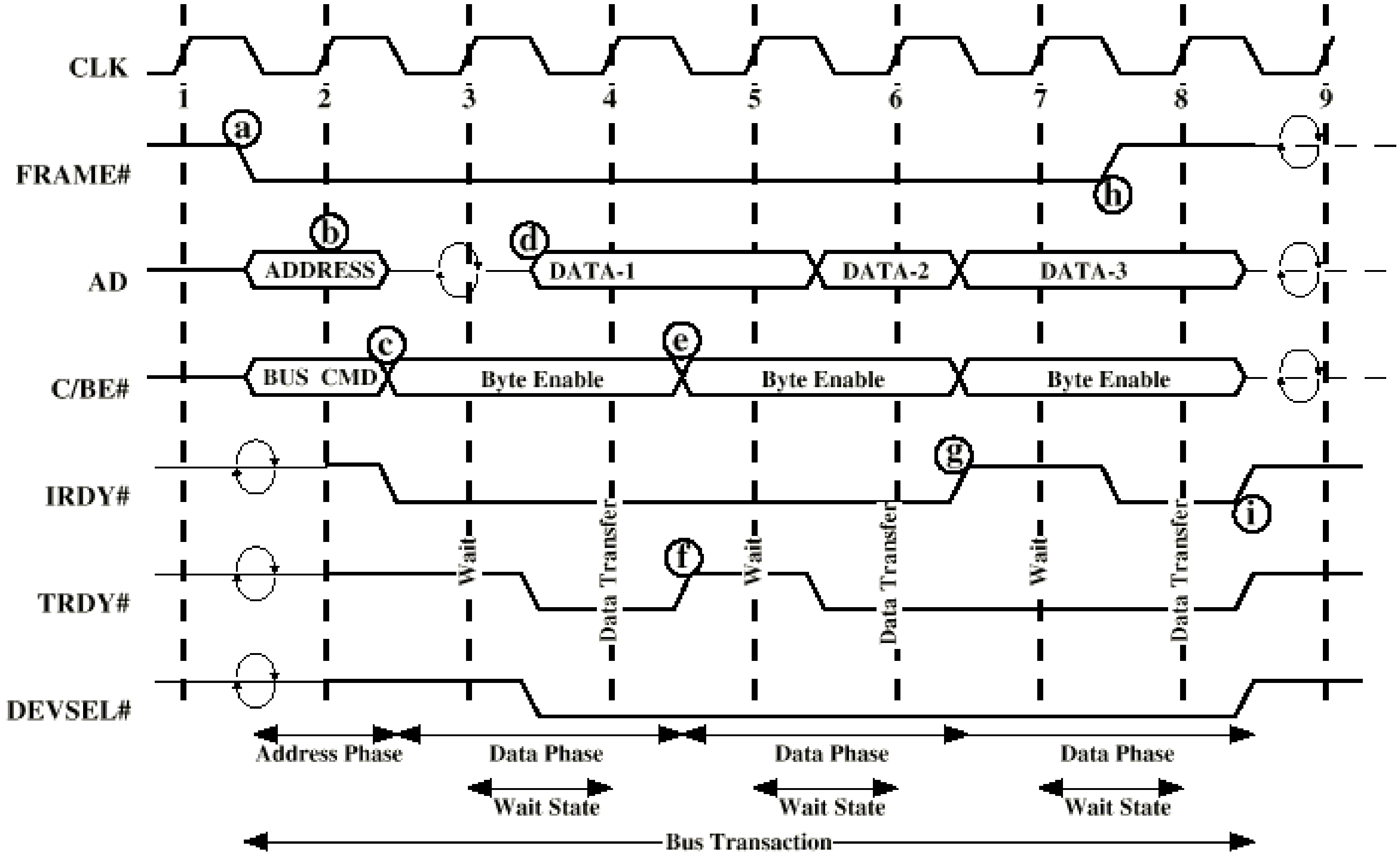
PCI Bus Lines (Optional)

- Interrupt lines
  - Not shared
- Cache support
- 64-bit Bus Extension
  - Additional 32 lines
  - Time multiplexed
  - 2 lines to enable devices to agree to use 64-bit transfer
- JTAG/Boundary Scan
  - For testing procedures

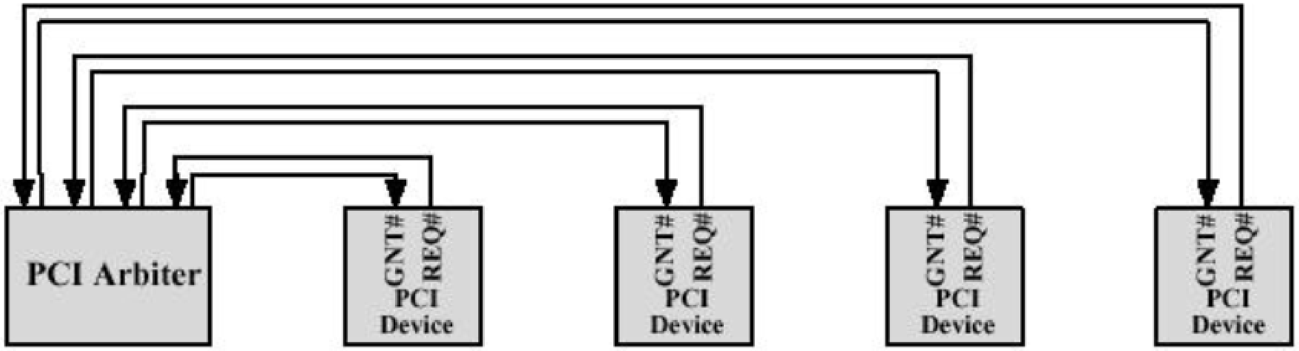
PCI Commands

- **Transaction** between initiator (master) and target (slave)
- Master claims bus
- Determine type of transaction
  - e.g. I/O read/write
- Address phase
- One or more data phases

PCI Timing Diagram (Read)

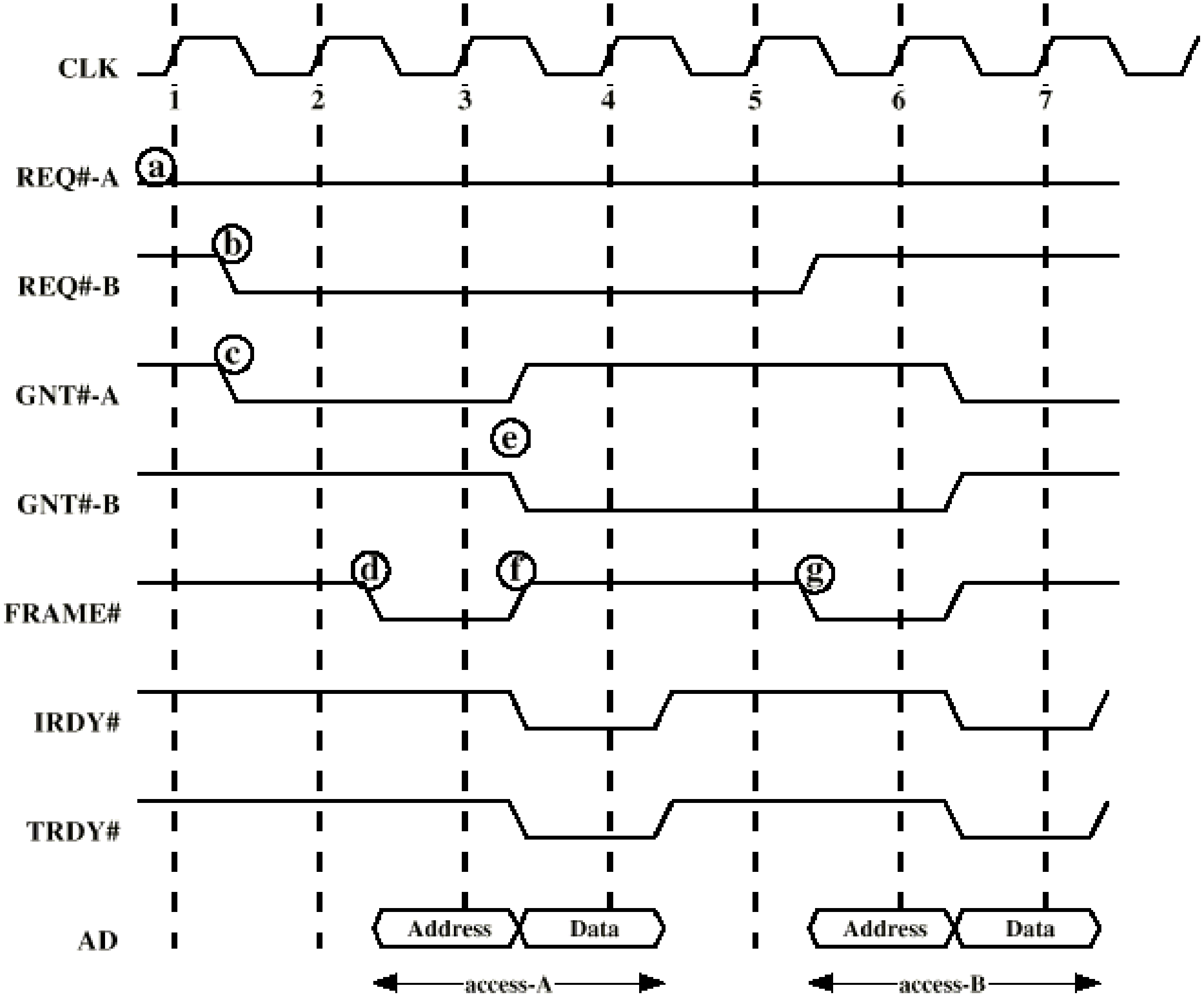


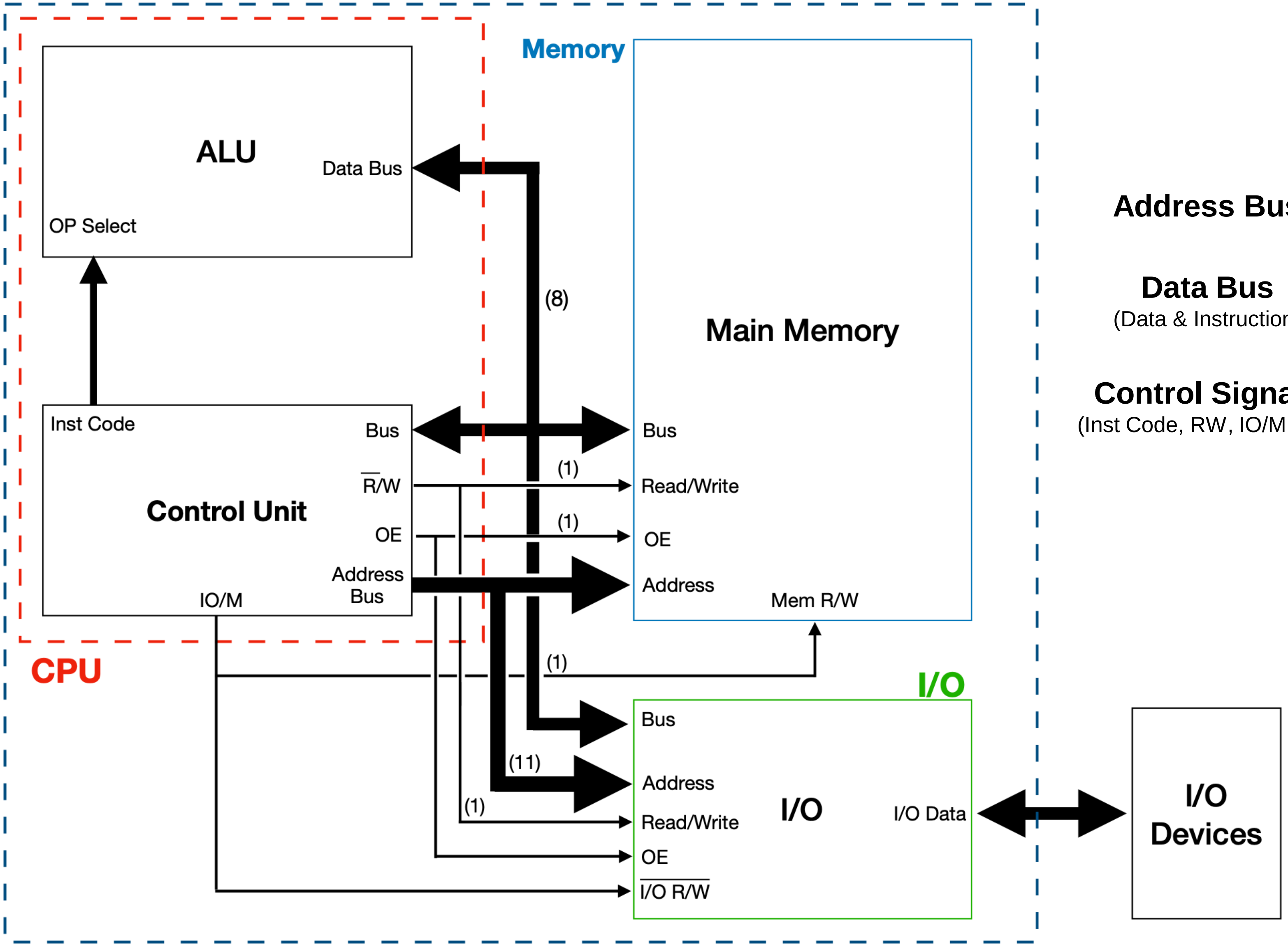
PCI Bus Arbitration



- **Centralized** and **synchronous**
- Each master has a **unique request (REQ)** and **grant (GNT) signal**, attached to a **central arbiter**
- **Arbitration algorithm** can be programmed into the arbiter
- Uses **hidden arbitration**
  - Arbitration can take place while other bus transactions are occurring on other bus lines

PCI Bus Arbitration Timing Diagram





Address Bus

Data Bus
(Data & Instruction)

Control Signals
(Inst Code, RW, IO/M, OE)

TRACS Buses

Further Reading

- Read about the following and legacy current bus technologies:
  - PCI Express
  - USB
  - Thunderbolt
  - Firewire

University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3202
Computer Organization & Architecture

End of Lecture

Note: The lecture slide was based on the accompanying lecture material from the Computer Organization and Architecture textbook by William Stallings. Some contents are contributed by Van B. Patiluna and Antoniette Mondigo-Cañete (USC). All contents contributed are copyright to the respective authors. For instructional use only. **Do not distribute.**

References:

- Brey, Barry B. *The Intel microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit extensions: architecture, programming, and interfacing* / Barry B. Brey—8th ed.
- Stallings, W. Computer Organization and Architecture, 6th edition, Pearson Education, Inc. (2003).