William Stallings
Computer Organization and Architecture
8th Edition

Chapter 3
Elements of Bus Design

What is a Bus?

- A communication pathway connecting two or more devices
- Usually broadcast
- Often grouped
  - A number of channels in one bus
  - 32 bit data bus is 32 separate single bit channels
- What do buses look like?
  - Parallel lines on circuit boards
  - Ribbon cables
  - Strip connectors on mother boards (e.g. PCI)
  - Sets of wires

Buses

- There are a # of possible interconnection systems
- Single and multiple BUS structures are most common
- e.g. Control/Address/Data bus (PC)
- e.g. Unibus (DEC-PDP)

Connecting

- All the units must be connected
- Different type of connection for different type of units
  - Memory
  - Input/Output
  - CPU

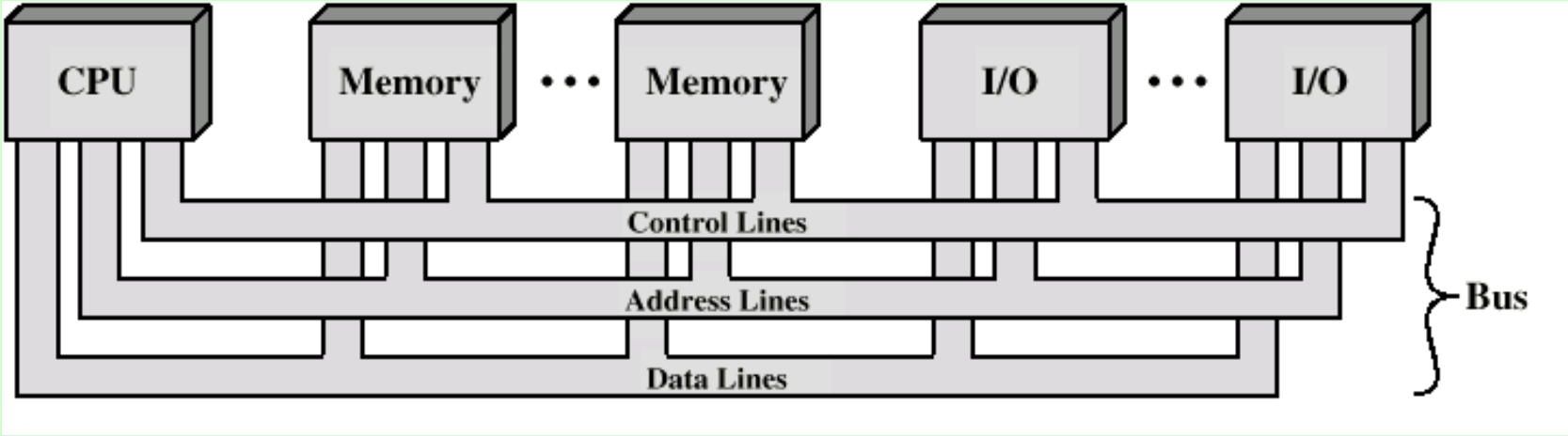
Bus Interconnection Scheme

Definition of a Bus

- Digital interconnection mechanism
- A set of parallel wires with rules for putting and retrieving information on the wires.
- A digital communication mechanism that allows two or more functional units to transfer control signals or data.
- The connection medium allowing the CPU, memory and I/O controllers to communicate

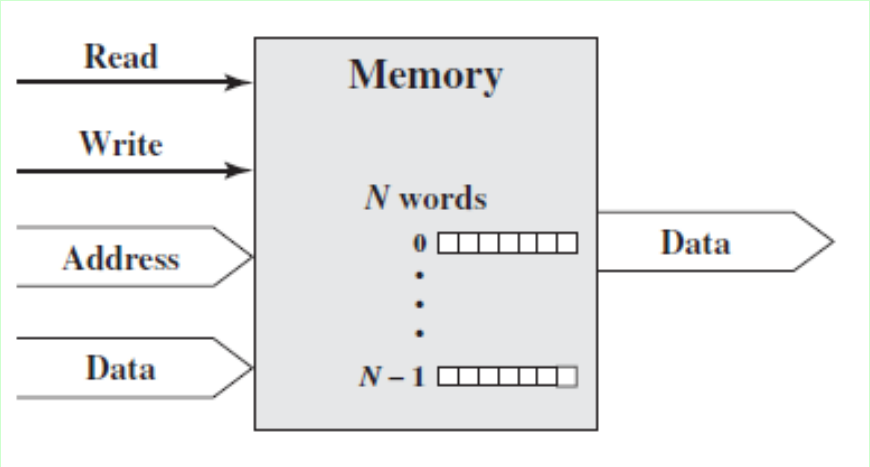
Bus lines

- Transfer of data
- Address information
- Control of the bus
  - Memory fetch or store
  - Ready
  - Bus Request and Bus Grant
  - Interrupt and Interrupt Acknowledge
  - Clock



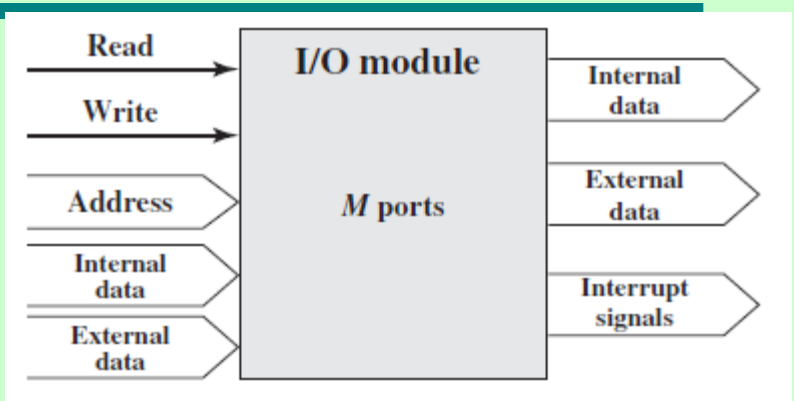
Memory Connection

- Receives and sends data
- Receives addresses (of locations)
- Receives control signals
  - —Read
  - —Write
  - —Timing



N words of = length. Each word is assigned a unique numerical address. A word can be read from or written into memory. The nature of operation is indicated by read and write control signals. The location for the operation is specified by an address

Input/Output Connection

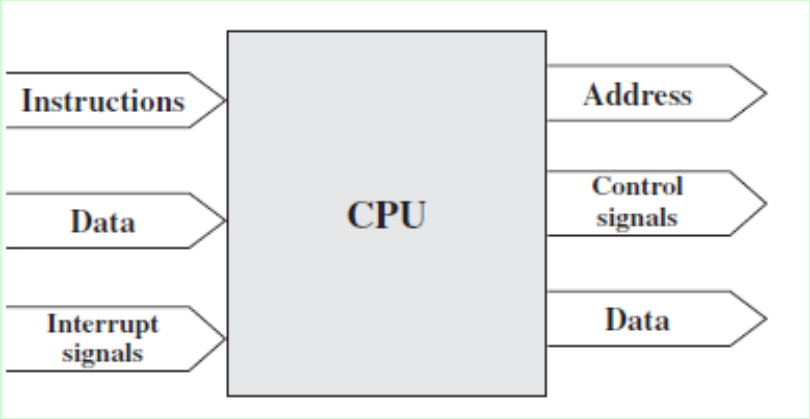


- Output
  - Receive data from computer
  - Send data to peripheral
- Input
  - Receive data from peripheral or send data to computer
- Receive control signals from computer
- Send control signals to peripherals (spin disk)
- Receive addresses from computer
  - —e.g. port number to identify peripheral
- Send interrupt signals (control)

I/O module may control more than one external device. There is an external data paths for the input and output of data with an external device. It is also able to send interrupt to the processor

CPU Connection

- Reads instruction and data
- Writes out data (after processing)
- Sends control signals to other units to control overall operation of the system
- Receives (& acts on) interrupts



The data to be exchanged are: memory to processor, processor to memory, I/O to processor, Processor to I/O, I/O to or from memory.

Interconnection structure

The interconnection structure must support the following types of transfers:

- **Memory to processor:** The processor reads an instruction or a unit of data from memory.
- **Processor to memory:** The processor writes a unit of data to memory.
- **I/O to processor:**The processor reads data from an I/O device via an I/O module.
- **Processor to I/O:** The processor sends data to the I/O device.
- **I/O to or from memory:** For these two cases, an I/O module is allowed to exchange data directly with memory, without going through the processor, using direct memory access (DMA).

Bus Types

- Data Bus
  - Carries data (no difference between “data” and “instruction”)
  - Width is a key determinant of performance(16, 32 bit)
- Address bus
  - Identify the source or destination of data (CPU needs to read an instruction (data) from a given location in memory)
  - Bus width determines maximum memory capacity of system
- Control Bus
  - Control and timing information
    - —Memory read/write signal
    - —Interrupt request
    - —Clock signals

Bus Width

- The width of a bus is the number of lines.
- The more data lines, the more data that can be transferred simultaneously.
- A “32 bit bus” has 32 data lines.
- The more address lines, the larger the maximum amount of memory that can be accessed.
- The greater the width, the more hardware required to implement the bus.

Examples

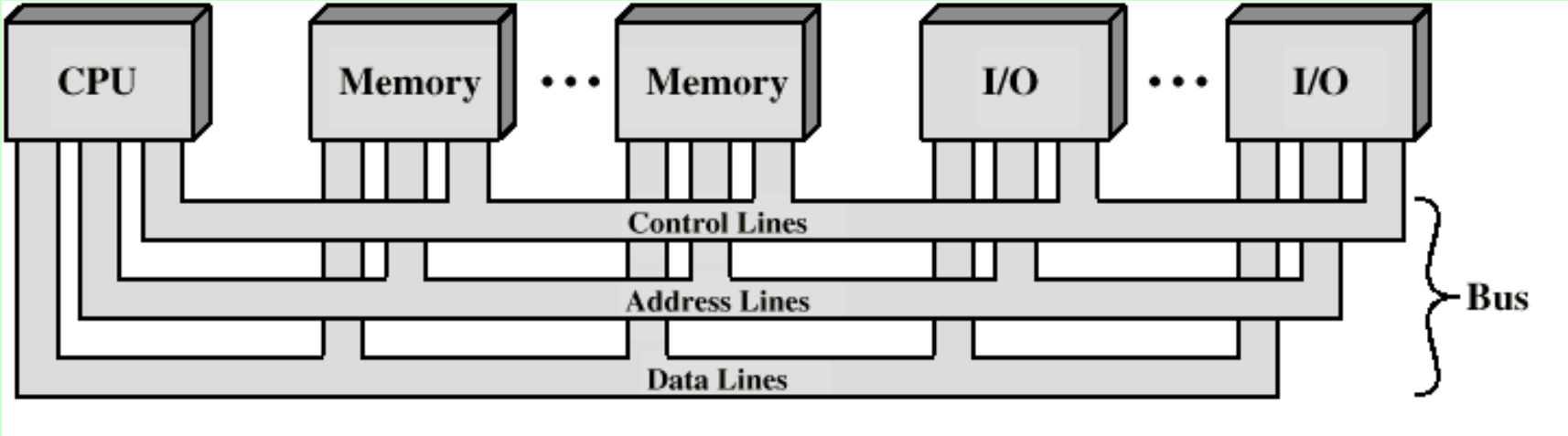
Pentium is a 32-bit processor with a 64-bit data bus
Itanium is a 64-bit processor with a 128-bit data bus

Address bus width

– Determines the system addressing capacity
– *N* address lines directly address 2^N memory locations
  8086: 20 address lines Can address 1 MB of memory
  Pentium: 32 address lines Can address 4 GB of memory
  Itanium: 64 address lines Can address 2^{64} bytes of memory
  AMD Athlon™ 64: 40 address lines Can address 1 TB of memory

Bus Interconnection Scheme

Data line or bus provides a path for moving data among systems modules. It may consist of 32, 64, or more separate lines. The # of lines is known as the width of data bus and determine how many bits can be transferred at a time. Each line carry one bit.

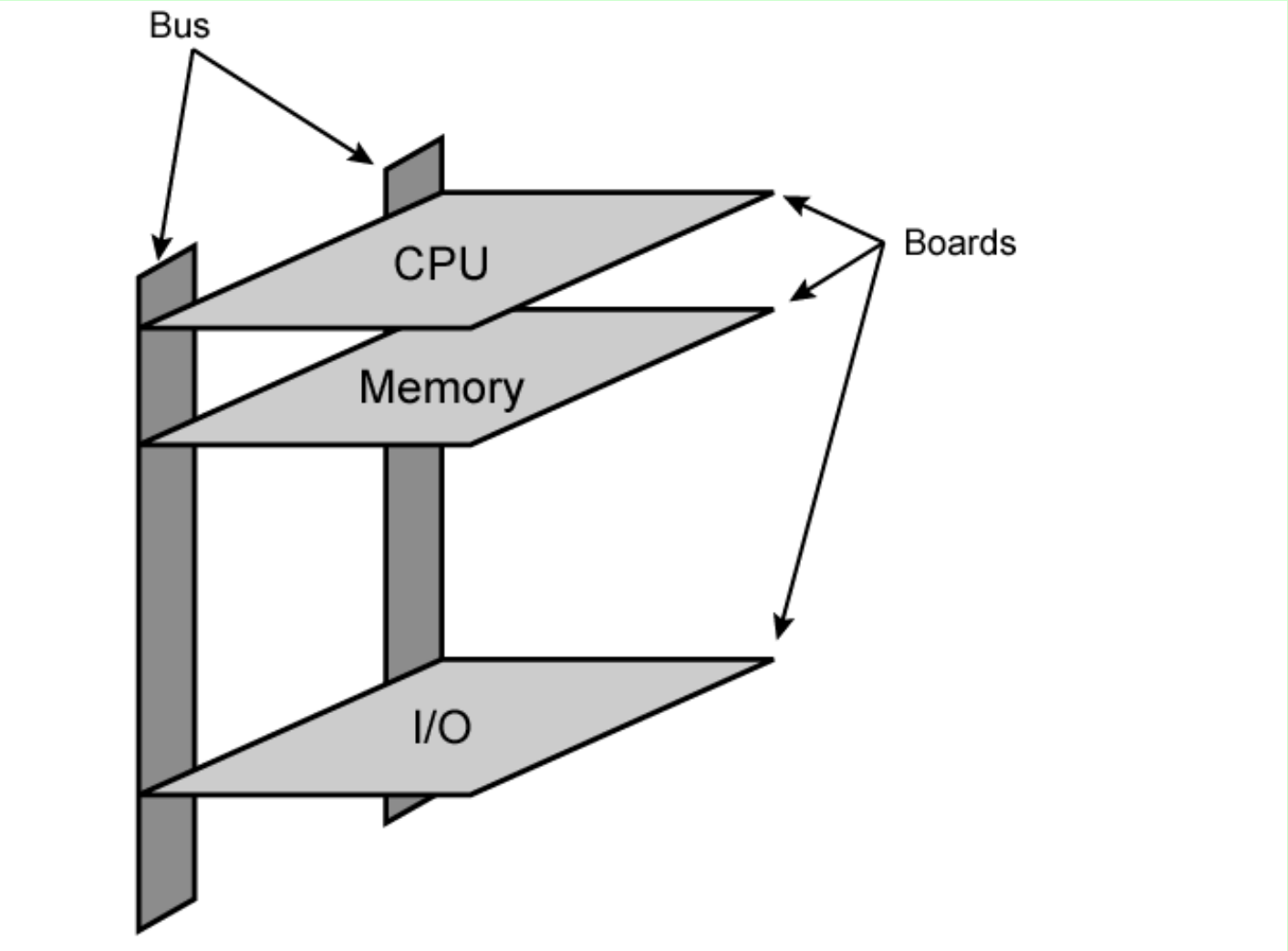


Address lines designate the source or destination on the data on the data bus. The width of the address bus determines the max possible memory capacity of the system. The address lines are also used to address I/O ports. Data and address lines are shared by all components.

Control signal transmit both command and timing info among system modules. They are used to control the access to and the use of data and address lines. Typical control lines include: memory write and read, I/O write, I/O read, transfer ACK, bus request, bus grant, interrupt request, interrupt ACK, Clock, reset.
Timing signal indicate the validity of data and address information.

Physical Realization of Bus Architecture

The operation of the bus is: 1- obtain the use of the bus and
2- transfer data to the bus

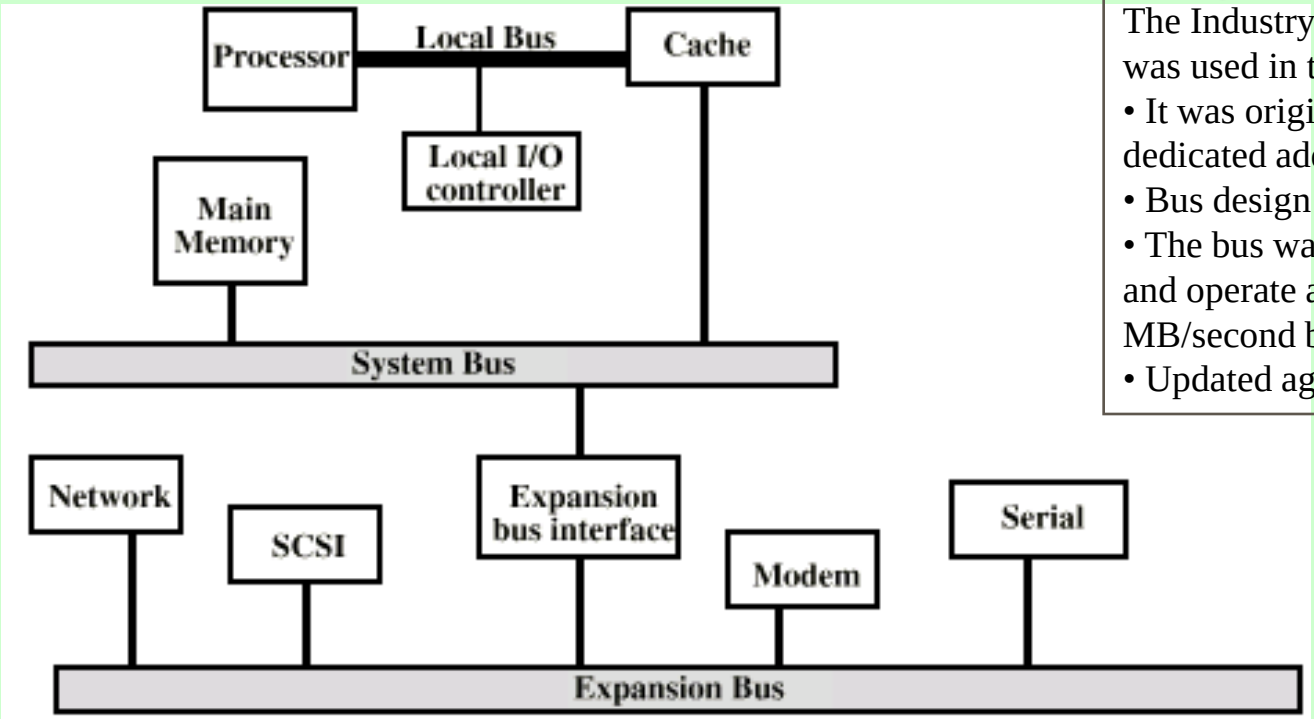


Single Bus Problems

- Lots of devices on one bus leads to:
  - Propagation delays
    - Long data paths mean that co-ordination of bus use can adversely affect performance
    - If aggregate data transfer approaches bus capacity
- Most systems use multiple buses to overcome these problems

Traditional (ISA) (with cache)

- Local bus connects processor to cache. Cache controller connects cache to system bus.
- System bus attach all memory modules.
- I/O transfers to and from main memory across system bus and do not interfere with processor activity.



History of ISA Bus

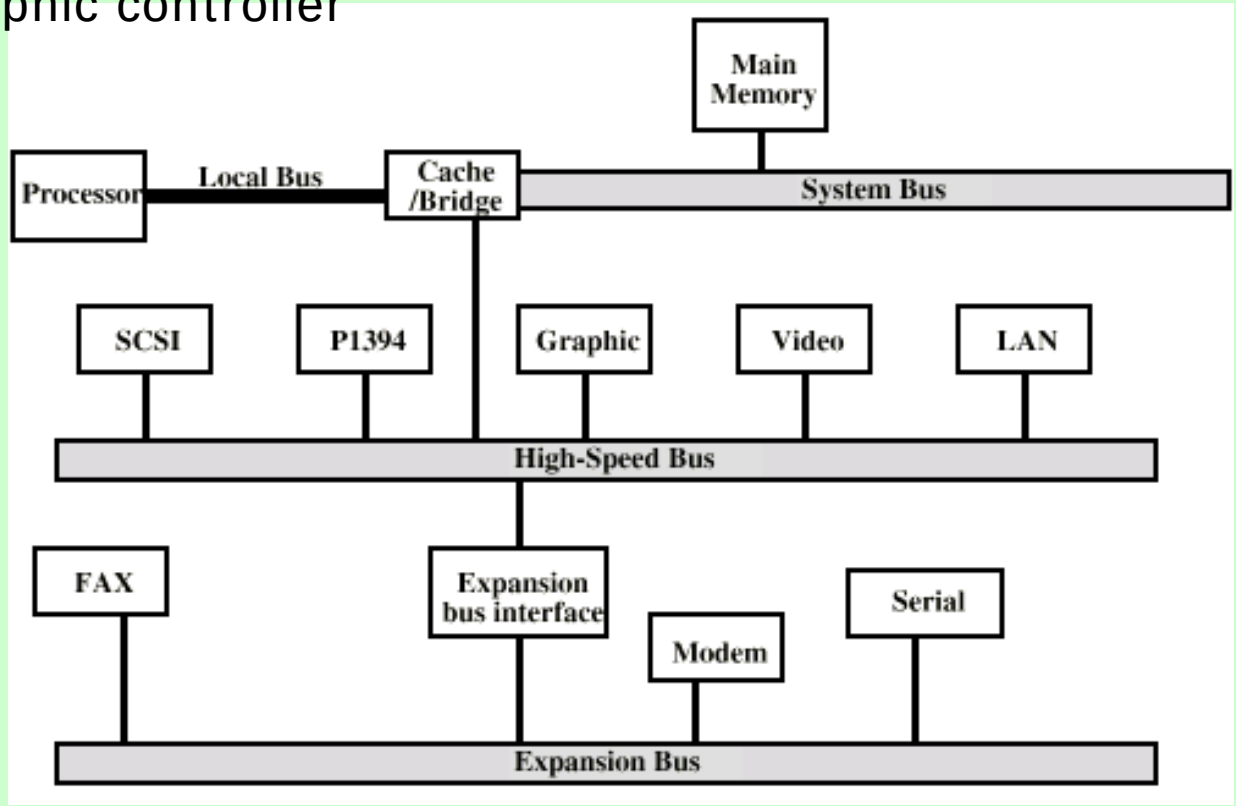
The Industry Standard Architecture (ISA) bus was used in the first 8088 PCs.

- It was originally a 8 bit data bus with 20 dedicated address lines.
- Bus design similar to 8088 local bus.
- The bus was updated to have 16 data lines and operate at 8.33 MHz providing 8 MB/second bandwidth.
- Updated again to the Extended ISA (EISA)

NOTE: SCSI or Small Computer System Interface, is a set of standards for physically connecting and transferring data between computers and peripheral devices. The SCSI standards define commands, protocols, and electrical and optical interfaces. SCSI is most commonly used for hard disks and tape drives, but it can connect a wide range of other devices, including scanners and CD drives.

High Performance Bus

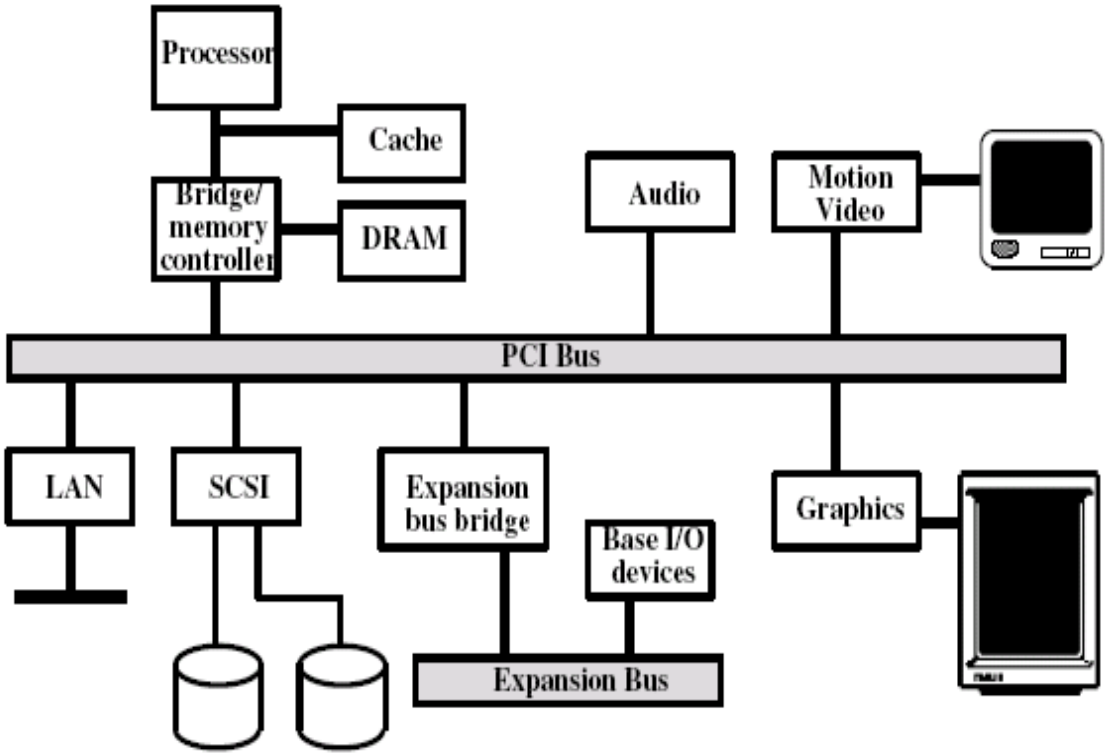
Local bus connects processor to cache controller which is connected to a bus system that supports main memory. CC is integrated into a bridge that connects to a high speed bus. HSB supports fast Ethernet, video & graphic controller



Advantages: HSB brings high demand devices into closer integration with processor and at the same time is independent from processor. Changes in processor architecture do not affect HSB

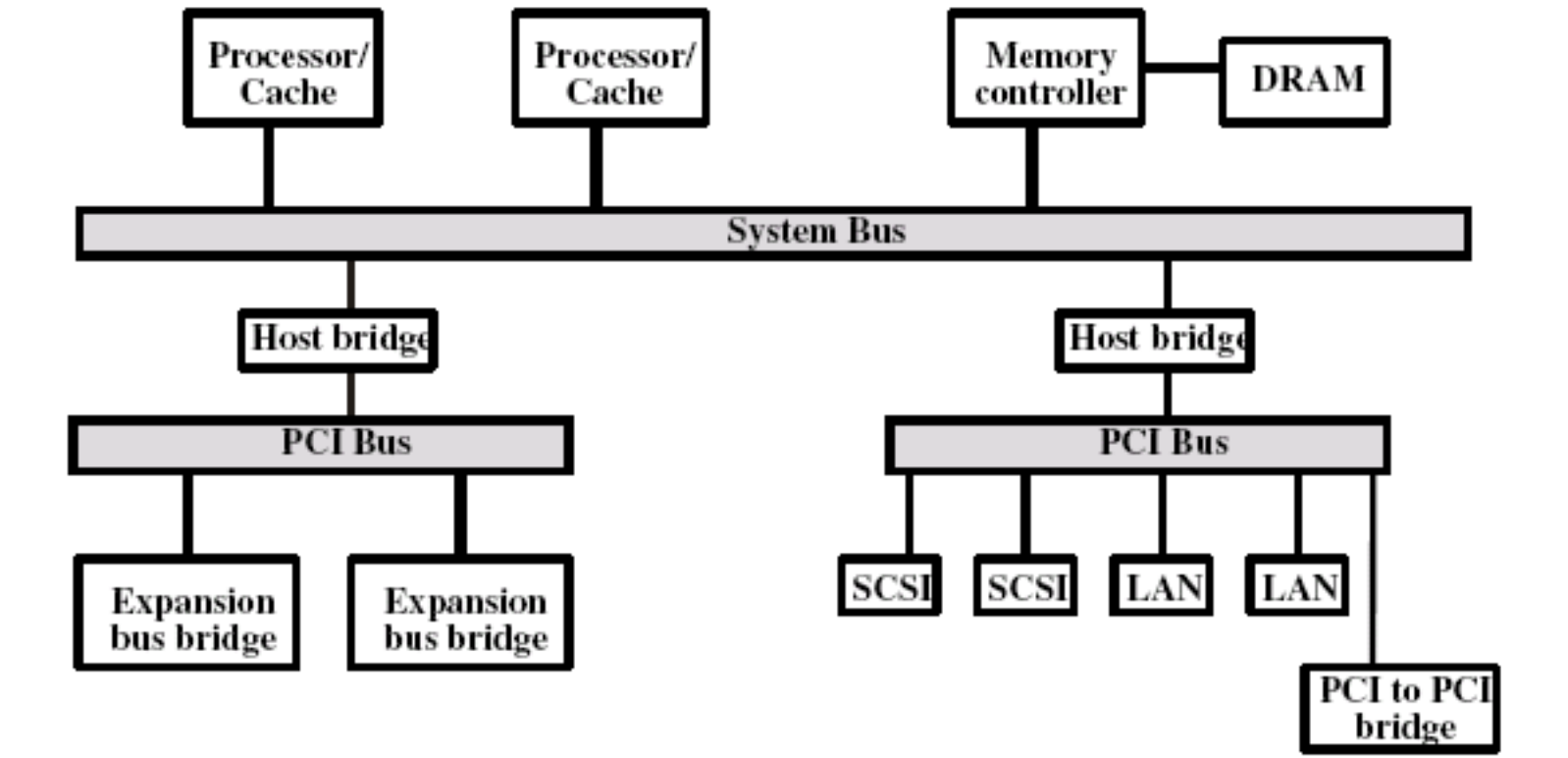
Desktop (PCI) Bus

- The Peripheral Component Interconnection (PCI) bus was developed by Intel in the early 1990s.
-PCI has 64 data lines running at 66 MHz providing up to 528 MB/sec bandwidth.
-Data and address lines are multiplexed.
- Centralized arbitration.



(a) Typical Desktop System

Server Bus



(b) Typical Server System

Bus Types

- Dedicated
  - Permanently assigned to one function or to a physical subset of computer components (Ex. Separate data & address lines)

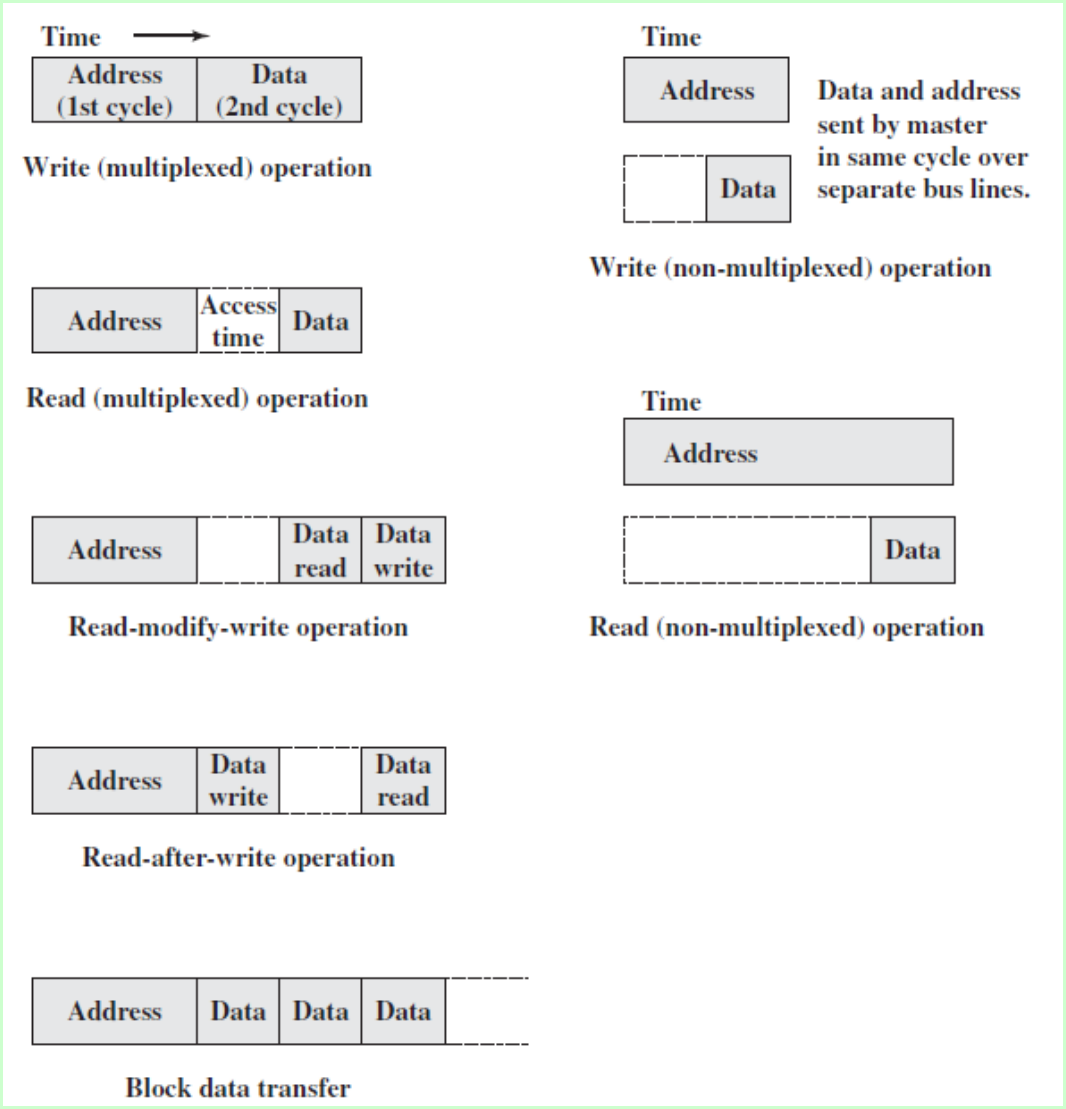
- Multiplexed: same line used for multiple purpose
  - Shared lines
  - Address valid or data valid control line
  - Advantage - fewer lines: save space & cost.
  - Disadvantages
    - More complex control: complex circuitry in each module
    - Ultimate performance: potential reduction in performance because certain events that share the same line can not take place in parallel

Multiplexed Bus Operations

In the case of a multiplexed address/data bus, the bus is first used for specifying the address and then for transferring the data.

For a read operation, there is typically a wait while the data are being fetched from the slave to be put on the bus.

For either a read or a write, there may be a delay if it is necessary to go through arbitration to gain control of the bus for the remainder of the operation



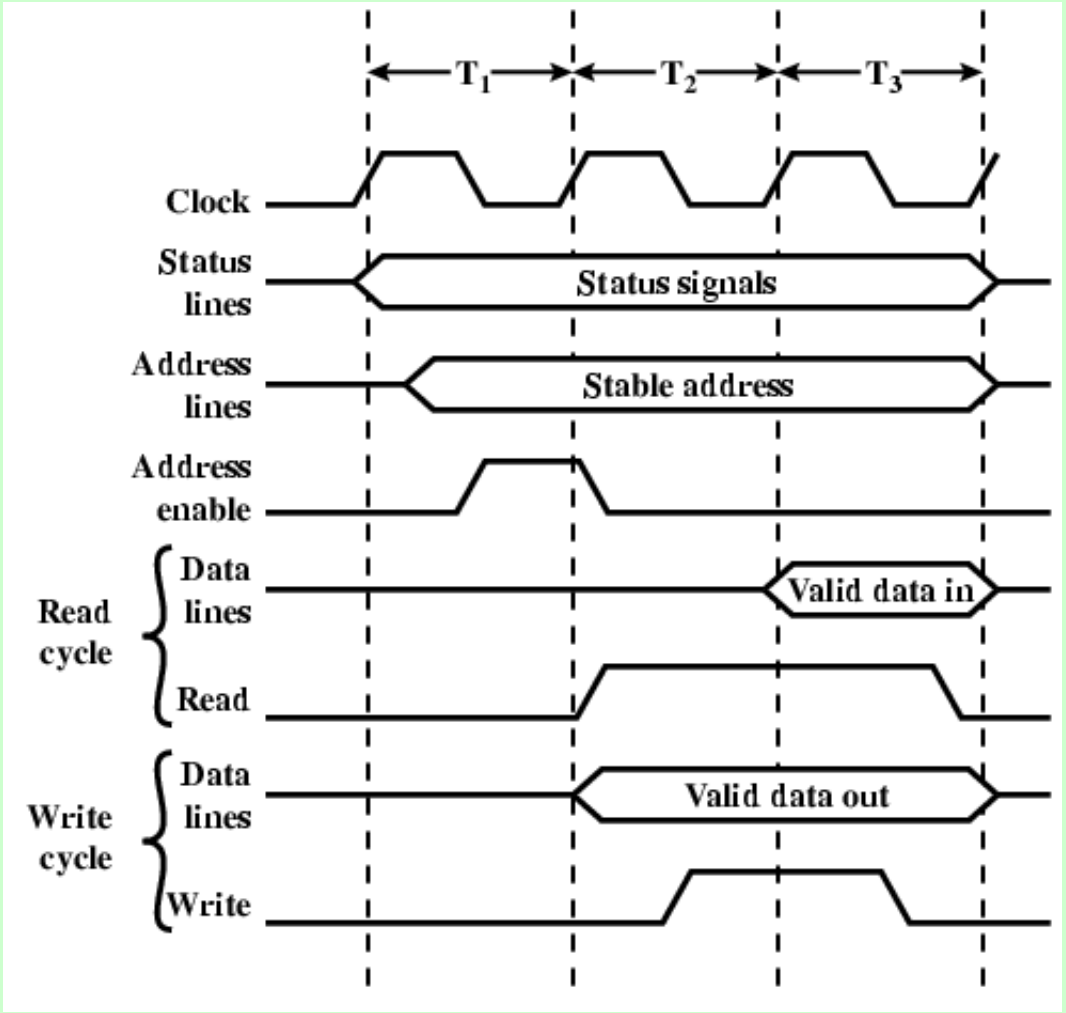
Timing

- Co-ordination of events on bus
- Synchronous
  - Events determined by clock signals
  - Control Bus includes clock line
  - A single 1-0 is a bus cycle
  - All devices can read clock line
  - Usually sync on leading edge
  - Usually a single cycle for an event

Synchronous Timing Diagram

Occurrence of events on a bus is determined by a clock

1. Processor place a MA on the AL during the first clock cycle.
2. Processor issues an address enable signal after AL is stabilized
3. for read operation: Processor issues read command at the start of 2nd cycle.
4. A MM recognize address and place data on the DL after a delay of 1 cycle. Processor reads from DL and drop the read signal
5. for write operation Processor puts data on DL at the start of 2nd cycle and issue a write command after DL have stabilized. MM copy info from DL during 3rd clock cycle

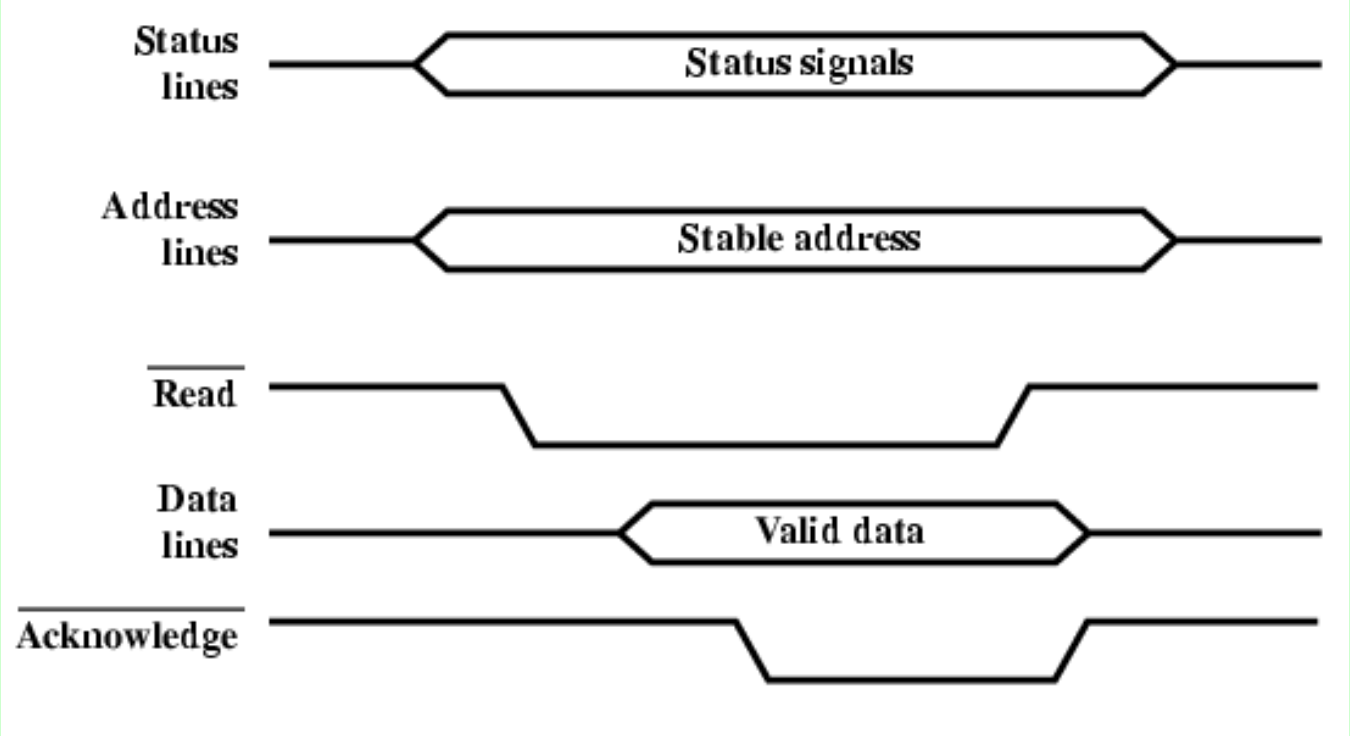


MA: memory address
AL: address line

MM: memory module
DL: data line

Asynchronous Timing – Read Diagram

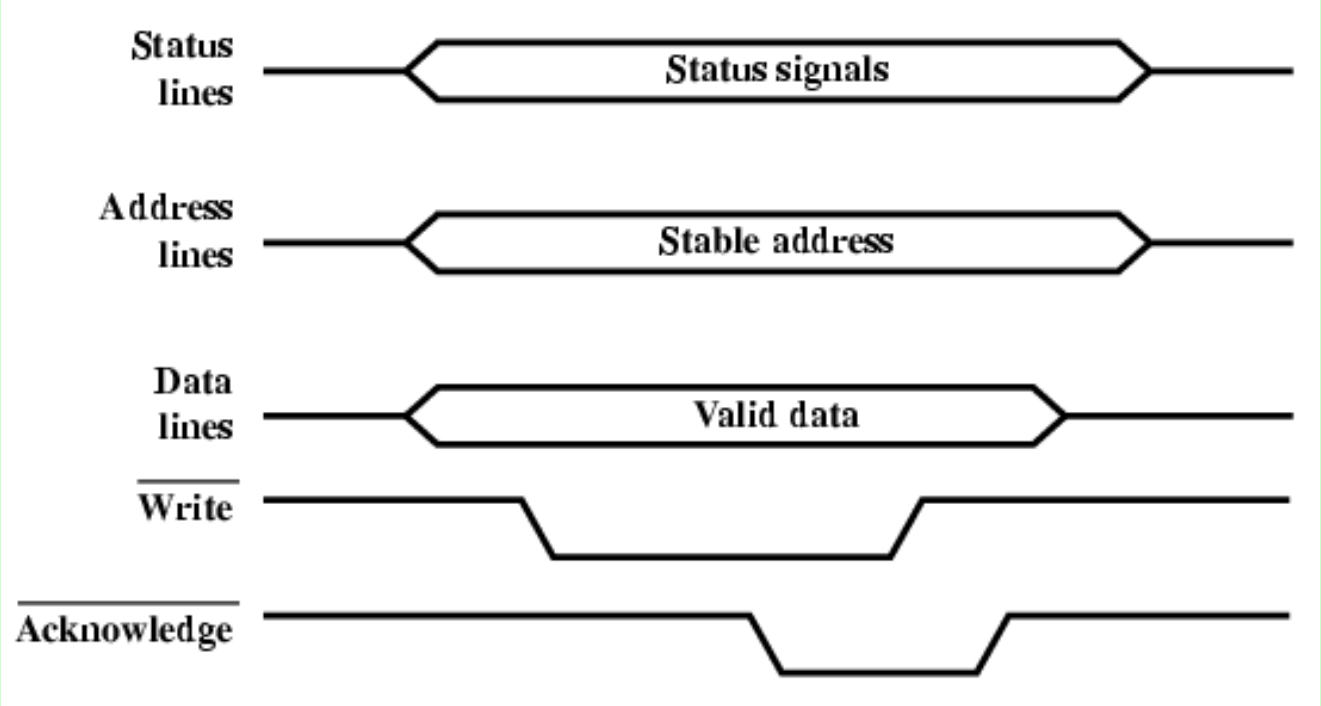
Occurrence of one event on a bus follows and depends on the occurrence of previous event



Processor places address and status signal on the bus. It issues a read command indicating the presence of valid address and control signals. Memory decodes address and place data on the DL. Memory module asserts Ac.L. to signal the processor that data are available. Once the master has read data from DL, it deasserts read signal. This causes memory module to drop data and Ac.L. Once Ac.L. is dropped, the master removes the address information

Asynchronous Timing – Write Diagram

The master places data on DL at the same time that is puts signals on the status and address lines. The memory module responds to the write command by copying data from DL and then asserting the Ac.L. Then the master drops the write signal & memory module drops Ac.L. signal



ST is simpler to implement and test. However, it is less flexible than AT. Because all devices on a synchronous bus are tied to a fixed clock rate, the system cannot take advantage of advances in device performance. With AT, a mixture of slow and fast devices, using older and newer technology, can share a bus.

PCI Bus

- Peripheral Component Interconnection
  - High bandwidth processor independent bus.
  - Deliver better system performance for high speed I/O subsystems.
  - Designed to meet I/O requirements for modern systems.
  - Designed to support a variety of microprocessor based configurations (single and multiprocessor systems.
  - Provide general purpose set of functions
- Intel released to public domain
- 32 or 64 bit
- 50 lines

PCI Commands

- Activities in the form of transaction between initiator (master) and target
- Master claims bus (acquire control)
- Determine type of transaction
  - e.g. I/O read/write
- Address phase to signal transaction type
- One or more data phases

PCI Commands

The commands are:

- Interrupt Acknowledge
- Special Cycle
- I/O Read
- Memory Read
- Memory Read Line
- Memory Read Multiple
- Memory Write
- Memory Write and Invalidate
- Configuration Read
- Configuration Write
- Dual Address Cycle

PCI Operation

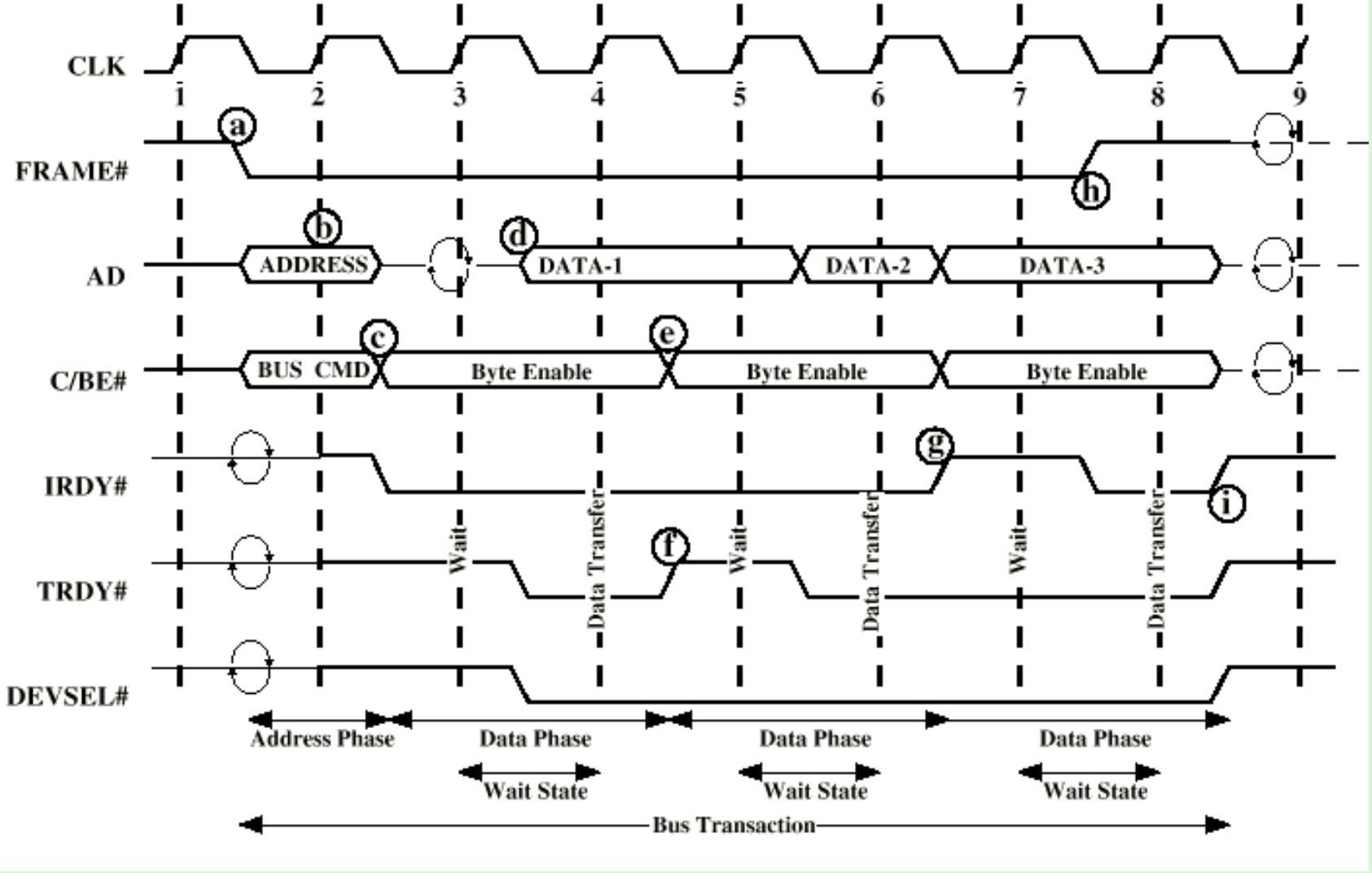
Optional Bus lines:

- Interrupt lines
- Cache support
- 64-bit Bus Extension
  - Additional 32 multiplexed lines
  - 2 lines to enable devices to agree to use 64-bit transfer

Transaction:

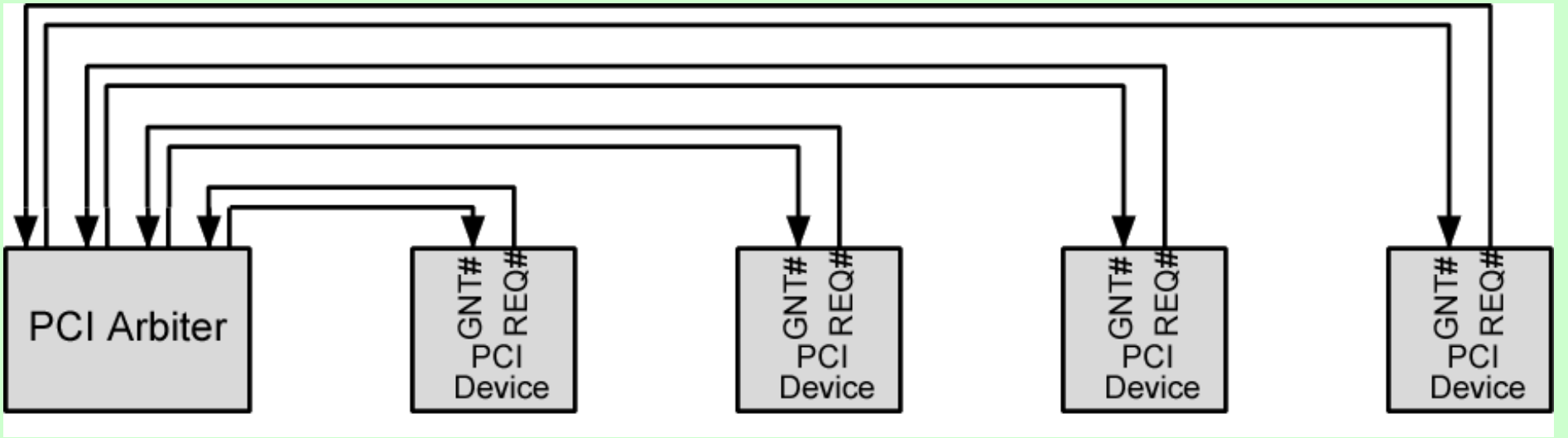
- All signal transitions are synchronized with F.E. of clock
- PCI devices sample the bus lines on the R.E. of clock
- Master claims bus & determines type of transaction:
  - e.g. I/O read/write
- Transaction
  - Starts with an address phase
  - Is followed by one or more data phases

PCI Read Timing Diagram



PCI Bus Arbiter

- Each device has a unique request REQ and grant GNT signal.
- These signals lines are attached to a central arbiter.
- Request-grant handshake is used to grant access to the bus.



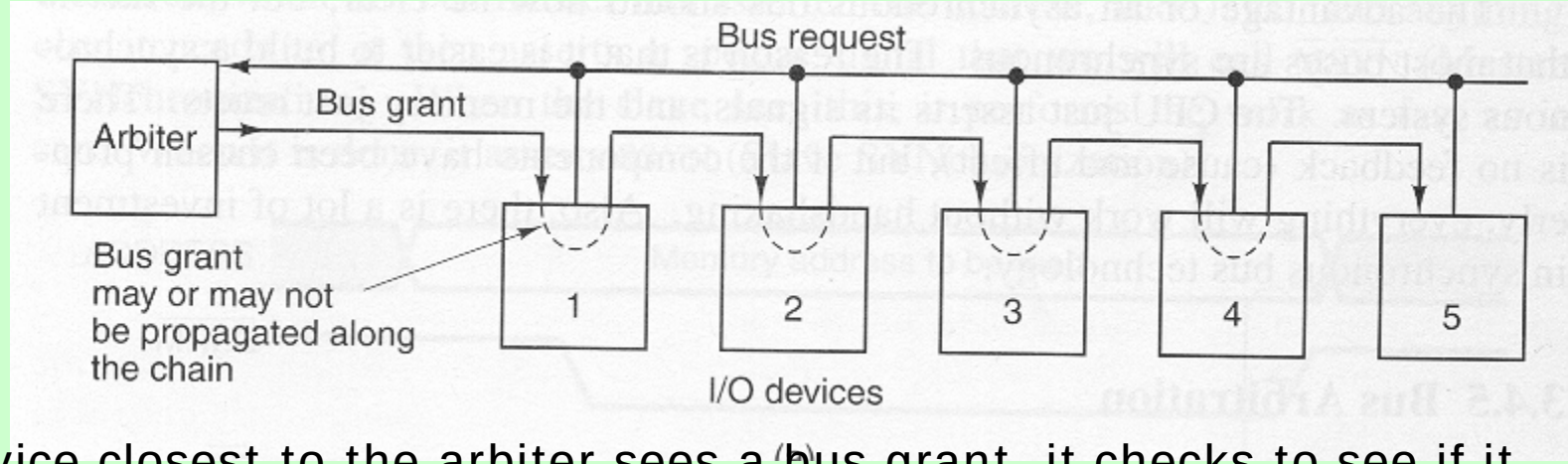
- PCI specifications does not dictate a particular arbitration algorithm.
- Arbiter can use first come first served approach or round robin approach.
- PCI master must arbitrate for each transaction that it wishes to perform.
- Single transaction consists of an address phase followed by one or more contiguous data phases.

Bus Arbitration

- More than one module controlling the bus
- e.g. CPU and DMA controller
- Only one module may control bus at 1 time
- Bus arbitration:
  - —Process by which another device is selected to become a bus master
  - — Arbitration may be :
    - -Centralized: single device (controller or arbiter) responsible for allocating time on the bus control bus access
      - – Bus Controller
      - – Arbiter
    - -May be part of CPU or separate

Centralized Arbitration

- Bus contains a single wired-OR request line
- Request line can be asserted by one or more devices
- When the arbiter sees a request, it issues a bus grant signal
- Bus grant is wired through devices using Daisy-Chaining

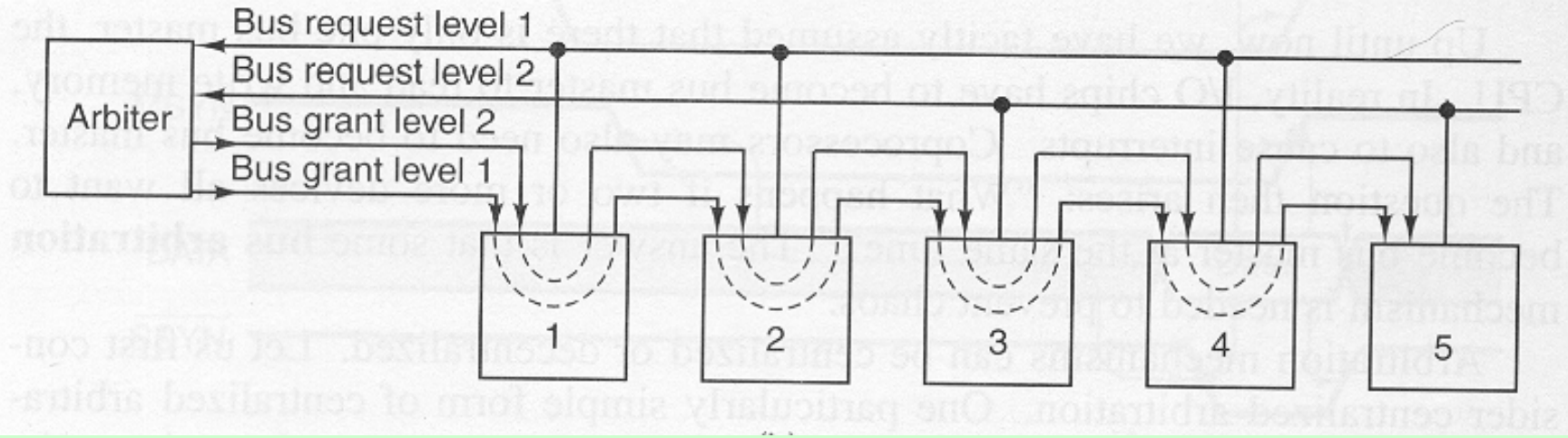


Device closest to the arbiter sees a bus grant, it checks to see if it made a request

- If yes:
  - It takes over the bus
  - It does not propagate bus grant further down the line
- If not:
  - It propagates the bus grant signal to the next device down the line
  - Here devices are assigned priorities depending on how close to the arbiter they are

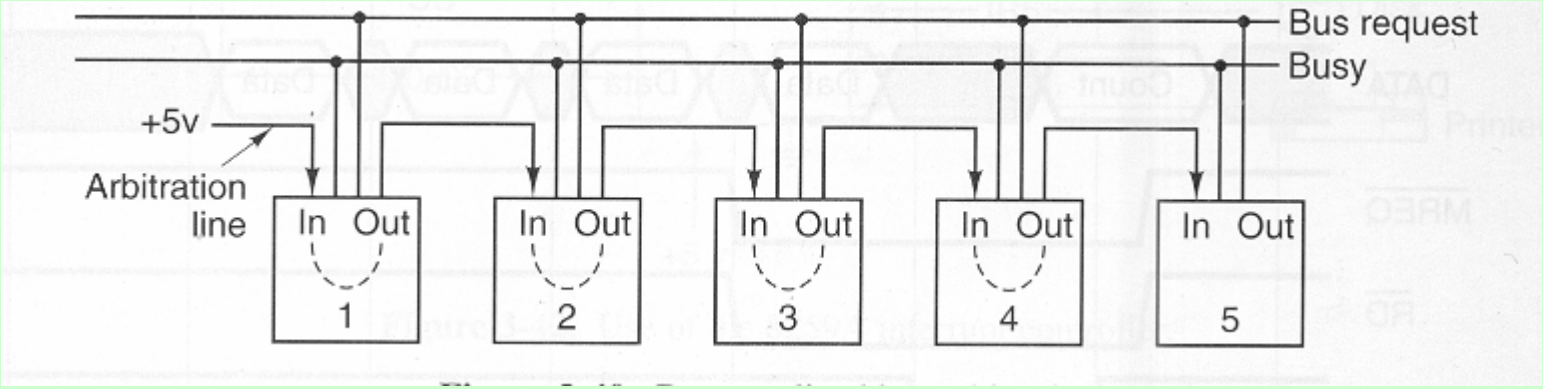
Centralized Arbitration 2nd Scenario

- Here, the bus has multiple priority levels
- More time-critical devices attach to higher priority bus request/grant lines
- If multiple requests are issued at the same time by devices of varying priority levels, arbiter issues a grant on the higher priority line
- If multiple requests are issued at the same time by devices of same priority level, daisy chaining is used



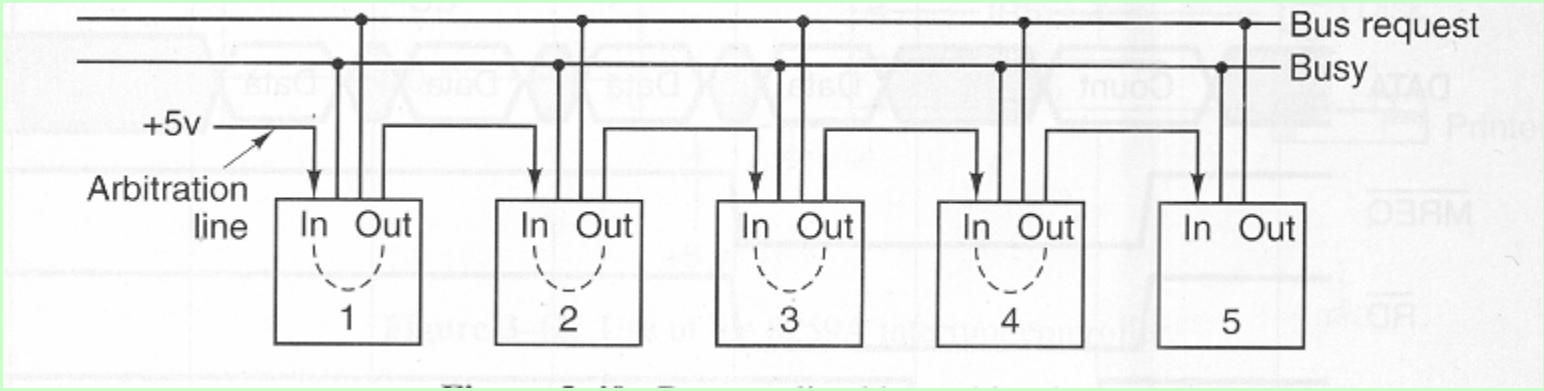
Distributed Arbitration

- No central controller. Each module contain access control logic & modules acts together to share the bus
  - Each module may claim the bus
  - Control logic on all modules
  - Bus request line: Wired-OR
  - Busy: asserted by current master
  - Arbitration line



Decentralized Arbitration

- When nobody wants the bus, asserted signal is propagated to all
- To acquire the bus, a device checks the busy line
  - If idle, it checks its arbitration signal (IN):
  - If not asserted:
    - –It may not become a bus master
    - –Then OUT is negated
  - If asserted:
    - –It negates its OUT
    - –Only one device has its IN asserted & OUT negated
    - –All downstream devices will have IN & OUT negated



PCI Bus Arbitration between Two Masters