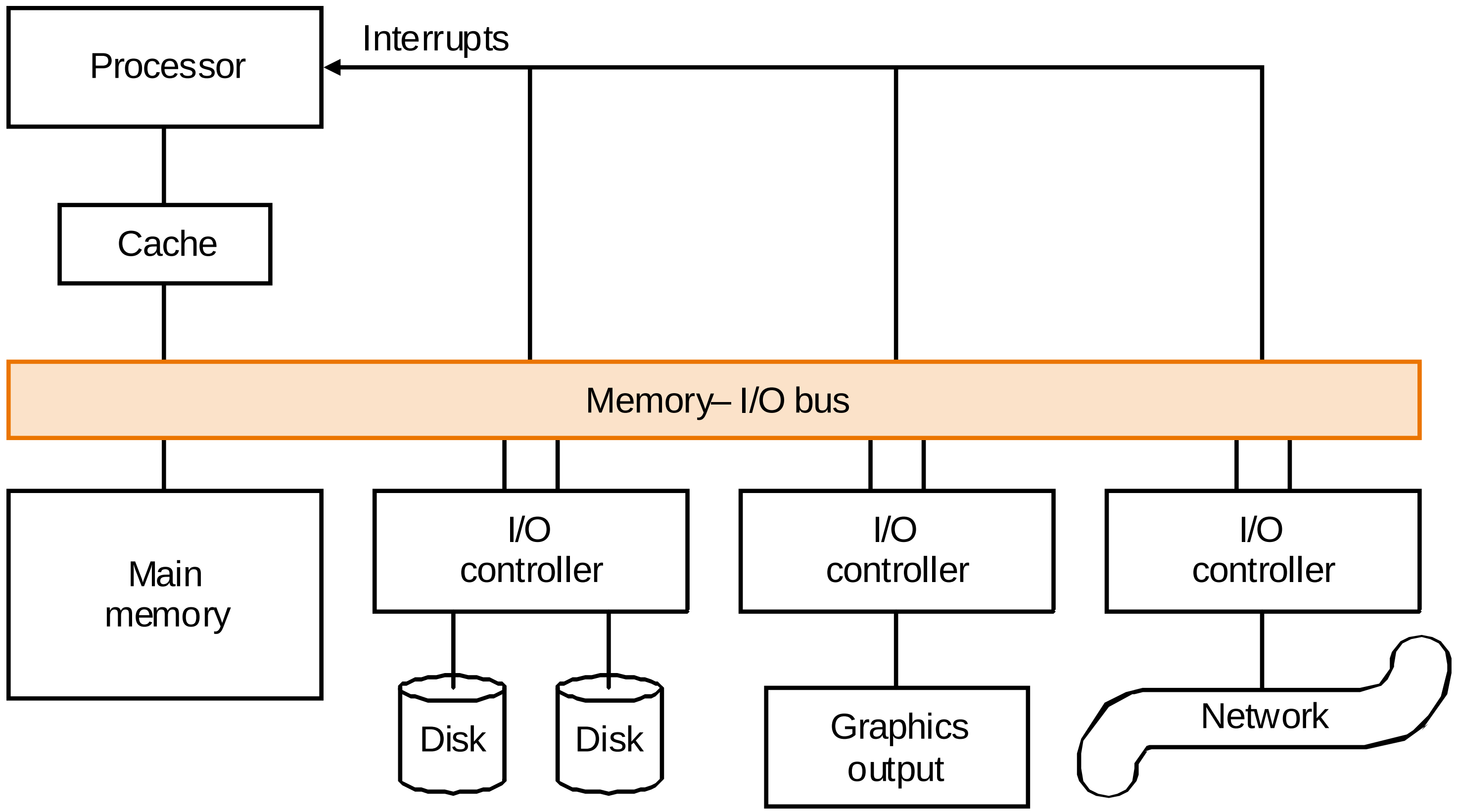
Interfacing Processor and Peripheral

EL3010 Arsitektur Sistem Komputer

Sekolah Teknik Elektro dan Informatika – ITB 2010

I/O System Design Issues

- Performance
- Expandability
- Resilience in the face of failure



Motivation for Input/Output

- I/O is how humans interact with computers
- I/O lets computers do amazing things:
  - Read pressure, temperature, etc
  - Control propellers, motors, communicate
  - Read bar codes of items in refrigerator
- Computer without I/O like a car without wheels; great technology, but won’t get you anywhere

I/O Device Examples and Speeds

- I/O Speed: bytes transferred per second (from mouse to display: million-to-1)

Device	Behavior	Partner	Data Rate (Kbytes/sec)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Line Printer	Output	Human	1
Floppy disk	Storage	Machine	50
Laser Printer	Output	Human	100
Magnetic Disk	Storage	Machine	10000
Network-LAN	I or O	Machine	10000
Graphics Display	Output	Human	30000

I/O System Performance

- I/O System performance depends on many aspects of the system ("limited by weakest link in the chain"):
  - The CPU
  - The memory system:
    - Internal and external caches
    - Main Memory
  - The underlying interconnection (buses)
  - The I/O controller
  - The I/O device
  - The speed of the I/O software (Operating System)
  - The efficiency of the software's use of the I/O devices
- Two common performance metrics:
  - Throughput: I/O bandwidth
  - Response time: Latency

Impact of I/O on System Performance

- Suppose a benchmark executes in 100 seconds, where 90 seconds is CPU time and the rest is I/O.
- If CPU time improves by 50% per year for the next five years, but I/O time does not improve, how much faster will the program run at the end of five years.

Elapsed time = CPU time + I/O time

I/O time = 100 - 90 = 10 seconds

After n Years	CPU time (seconds)	I/O time (seconds)	Elapsed time (seconds)	% I/O time
0	90	10	100	0.1
1	90/1.5 = 60	10	70	0.142857
2	60/1.5 = 40	10	50	0.2
3	40/1.5 = 27	10	37	0.27027
4	27/1.5 = 18	10	28	0.357143
5	18/1.5 = 12	10	22	0.454545

I/O Architecture

- Two methods are used to address the device:
  - Special I/O instructions
  - Memory-mapped I/O
- Special I/O instructions specify:
  - Both the device number and the command word
  - Device number: the processor communicates this via a set of wires normally included as part of the I/O bus
  - Command word: this is usually send on the bus’s data lines

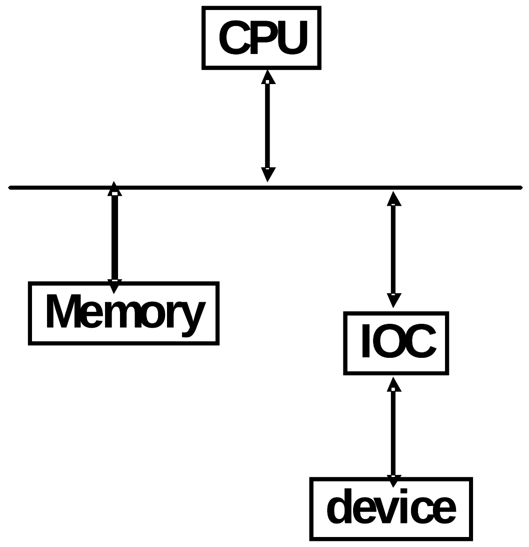
I/O Architecture

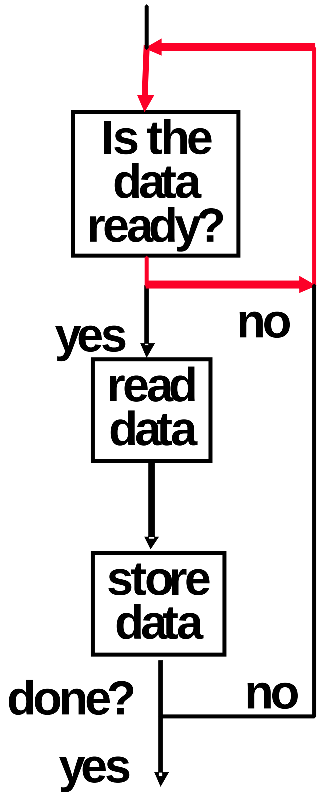
- Memory-mapped I/O:
  - Portions of the address space are assigned to I/O device
  - Read and writes to those addresses are interpreted s commands to the I/O devices
  - User programs are prevented from issuing I/O operations directly:
    - The I/O address space is protected by the address translation

I/O Device Notifying the OS

- The OS needs to know when:
  - The I/O device has completed an operation
  - The I/O operation has encountered an error
- This can be accomplished in two different ways:
  - Polling:
    - The I/O device put information in a status register
    - The OS periodically check the status register
  - I/O Interrupt:
    - Whenever an I/O device needs attention from the processor, it interrupts the processor from what it is currently doing.

Polling: Programmed I/O





busy wait loop not an efficient way to use the CPU unless the device is very fast!

but checks for I/O completion can be dispersed among computation intensive code

- Advantage:
  - Simple: the processor is totally in control and does all the work
- Disadvantage:
  - Polling overhead can consume a lot of CPU time

Cost of Polling?

- Assume for a processor with a 1000-MHz clock it takes 800 clock cycles for a polling operation (call polling routine, accessing the device, and returning). Determine % of processor time for polling
  - Mouse: polled 30 times/sec so as not to miss user movement
  - Floppy disk: transfers data in 2-byte units and has a data rate of 50 KB/second. No data transfer can be missed.
  - Hard disk: transfers data in 16-byte chunks and can transfer at 8 MB/second. Again, no transfer can be missed.

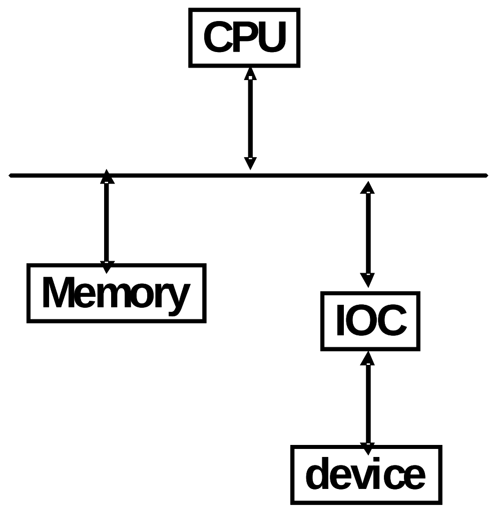
% Processor time to poll mouse, floppy

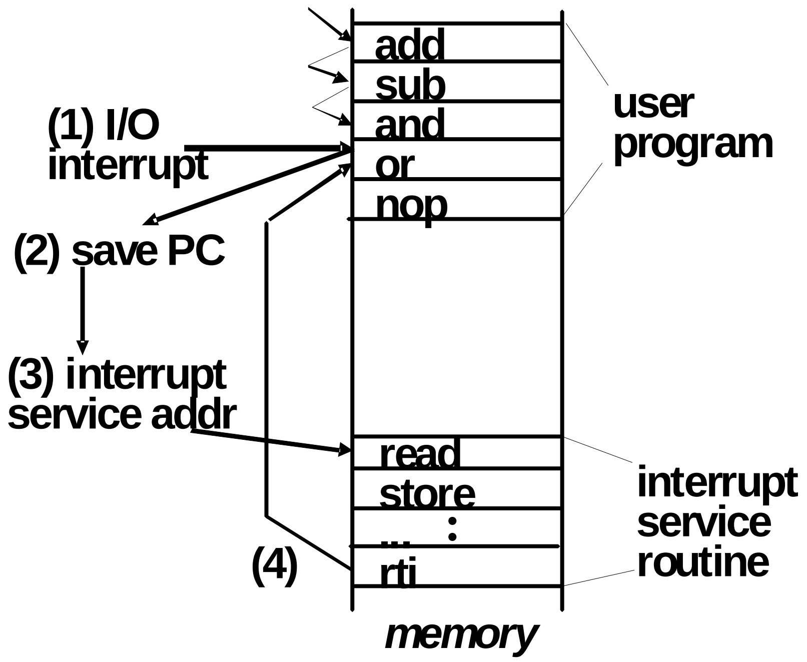
- Mouse Polling Clocks/sec
  - = 30 * 800 = 24000 clocks/sec
  - % Processor for polling:
    - 24*103/(1000*106) = 0.002%
  - Polling mouse little impact on processor
- Times Polling Floppy/sec
  - = 50 KB/s /2B = 25K polls/sec
  - Floppy Polling Clocks/sec
    - = 25K * 800 = 20,000,000 clocks/sec
  - % Processor for polling:
    - 20*106/(1000*106) = 2%
  - OK if not too many I/O devices

% Processor time to hard disk

- Times Polling Disk/sec
  - = 8 MB/s /16B = 500K polls/sec
- Disk Polling Clocks/sec
  - = 500K * 800 = 400,000,000 clocks/sec
- % Processor for polling:
  - 400*106/1000*106 = 40%
- Unacceptable

Interrupt Driven Data Transfer





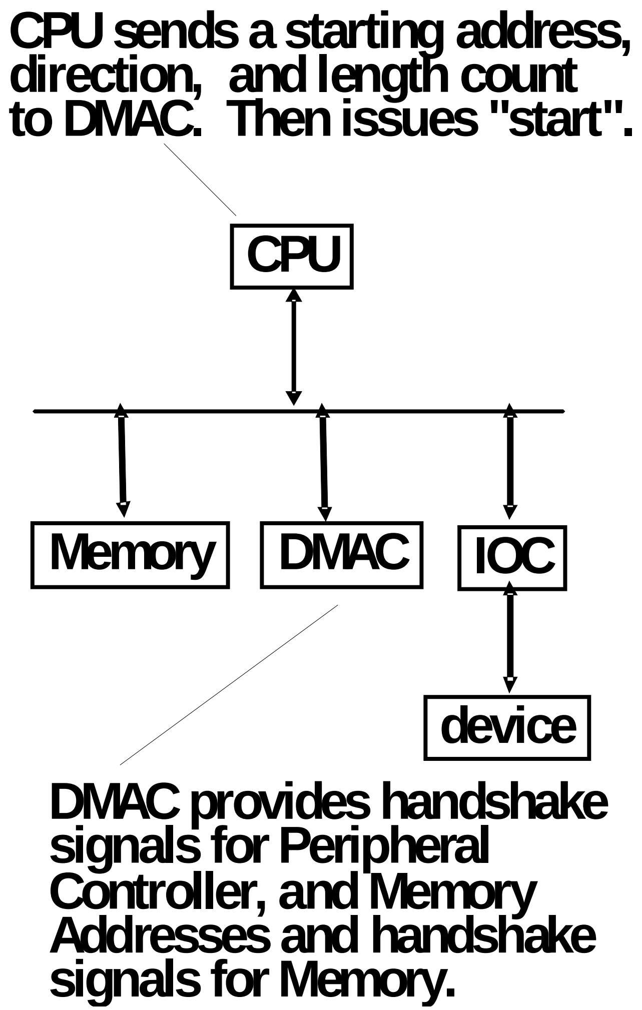
- Advantage:
  - User program progress is only halted during actual transfer
- Disadvantage, special hardware is needed to:
  - Cause an interrupt (I/O device)
  - Detect an interrupt (processor)
  - Save the proper states to resume after the interrupt (processor)

I/O Interrupt

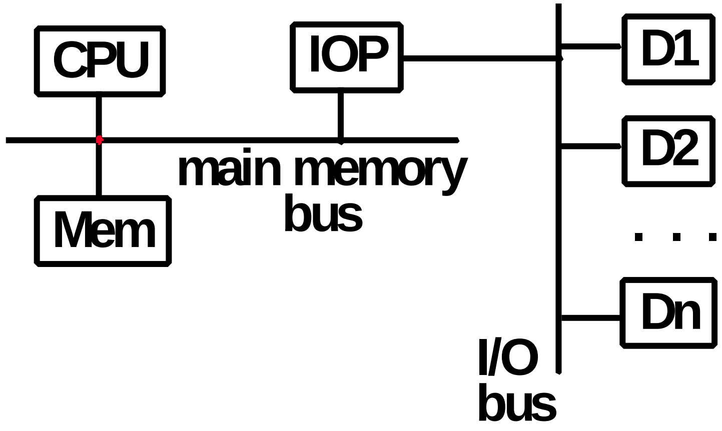
- An I/O interrupt is just like the exceptions except:
  - An I/O interrupt is asynchronous
  - Further information needs to be conveyed
- An I/O interrupt is asynchronous with respect to instruction execution:
  - I/O interrupt is not associated with any instruction
  - I/O interrupt does not prevent any instruction from completion
  - You can pick your own convenient point to take an interrupt
- I/O interrupt is more complicated than exception:
  - Needs to convey the identity of the device generating the interrupt
  - Interrupt requests can have different urgencies:
  - Interrupt request needs to be prioritized

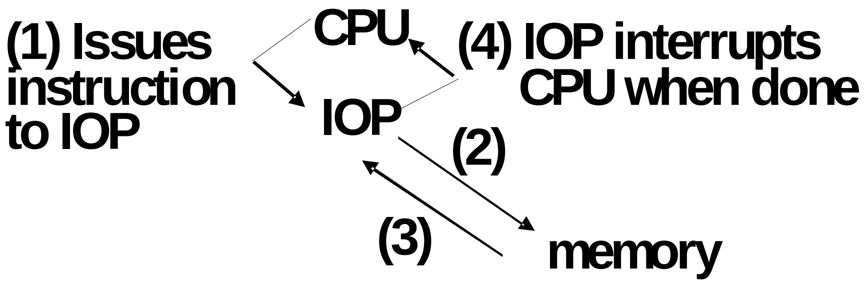
Delegating I/O Responsibility from the CPU: DMA

- Direct Memory Access (DMA):
  - External to the CPU
  - Act as a master on the 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".
CPU
Memory
DMAC
IOC
device
DMAC provides handshake signals for Peripheral Controller, and Memory Addresses and handshake signals for Memory.

Delegating I/O Responsibility from the CPU: IOP

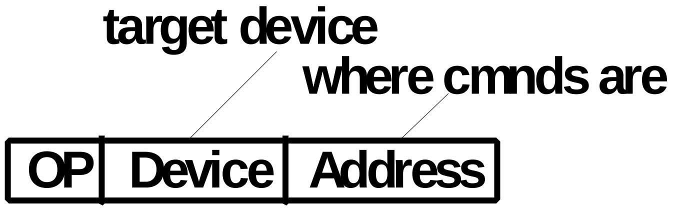




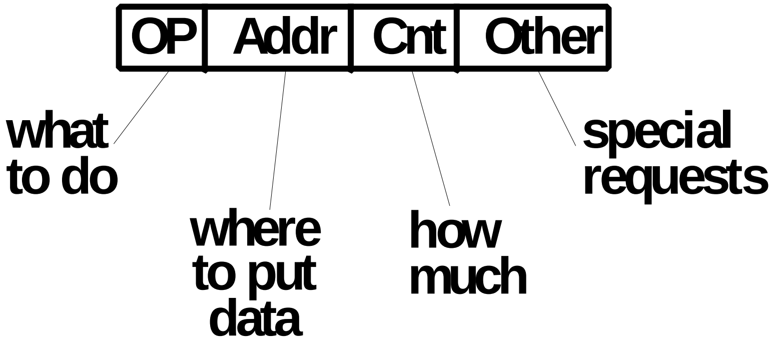
(1) Issues instruction to IOP
CPU
(4) IOP interrupts CPU when done
IOP
(2)
(3)
memory

Device to/from memory transfers are controlled by the IOP directly.

IOP steals memory cycles.



IOP looks in memory for commands



Responsibilities of the Operating System

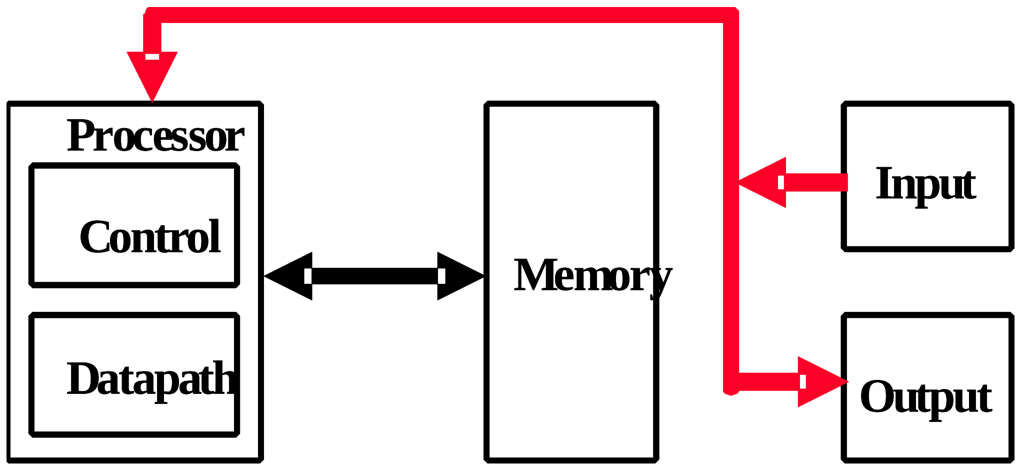
- The operating system acts as the interface between:
  - The I/O hardware and the program that requests I/O
- Three characteristics of the I/O systems:
  - The I/O system is shared by multiple program using the processor
  - I/O systems often use interrupts (external generated exceptions) to communicate information about I/O operations.
    - Interrupts must be handled by the OS because they cause a transfer to supervisor mode
  - The low-level control of an I/O device is complex:
    - Managing a set of concurrent events
    - The requirements for correct device control are very detailed

Operating System Requirements

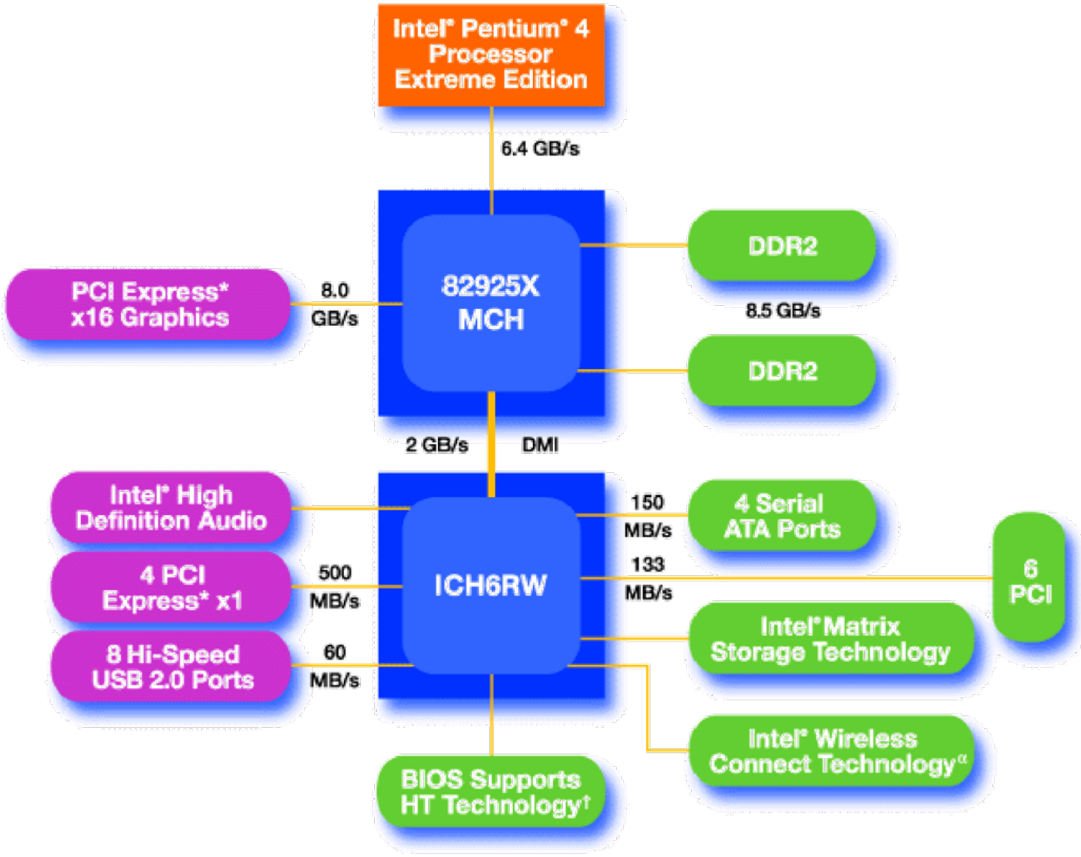
- Provide protection to shared I/O resources
  - Guarantees that a user's program can only access the portions of an I/O device to which the user has rights
- Provides abstraction for accessing devices:
  - Supply routines that handle low-level device operation
- Handles the interrupts generated by I/O devices
- Provide equitable access to the shared I/O resources
  - All user programs must have equal access to the I/O resources
- Schedule accesses in order to enhance system throughput

Bus

- Shared communication link
- Single set of wires used to connect multiple subsystems
- A Bus is also a fundamental tool for composing large, complex systems
  - systematic means of abstraction



Example: Pentium System Organization

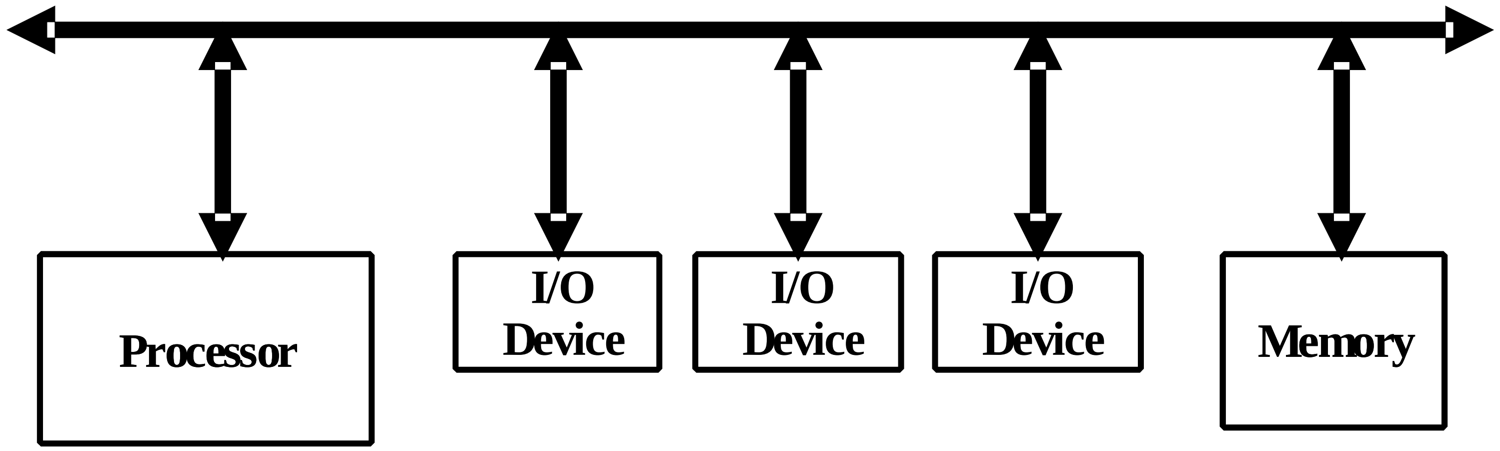


Intel® Pentium® 4 Processor Extreme Edition
6.4 GB/s
82925X MCH
PCI Express* x16 Graphics
8.0 GB/s
DDR2
8.5 GB/s
DDR2
2 GB/s
DMI
ICH6RW
Intel® High Definition Audio
4 PCI Express* x1
500 MB/s
8 Hi-Speed USB 2.0 Ports
60 MB/s
150 MB/s
4 Serial ATA Ports
133 MB/s
6 PCI
Intel® Matrix Storage Technology
Intel® Wireless Connect Technology[α]
BIOS Supports HT Technology[†]

[†] Hyper-Threading (HT) Technology requires a computer system with an Intel® Pentium® 4 processor supporting Hyper-Threading Technology and an HT Technology enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. See http://www.intel.com/info/hyperthreading/ for more information including details on which processors support HT Technology.

[α] Intel® Wireless Connect Technology requires an Intel® 9xx Express chipset with Intel® 82801FW or RW (ICH6W or RW) I/O Controller Hub and a separate Intel® PRO/Wireless 2225BG Network Connection solution. Availability may be limited. Consult with your system vendor for more information, including whether Intel Wireless Connect Technology has been enabled on your system.

Advantages of Buses

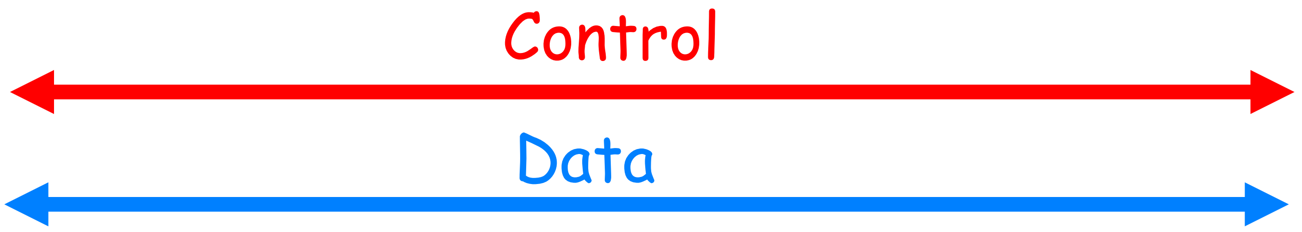


- Versatility:
  - New devices can be added easily
  - Peripherals can be moved between computer systems that use the same bus standard
- Low Cost:
  - A single set of wires is shared in multiple ways
- Manage complexity by partitioning the design

Disadvantage of Buses

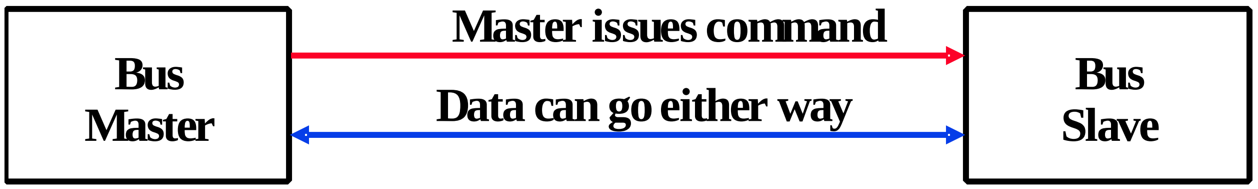
- It creates a communication bottleneck
  - The bandwidth of that bus can limit the maximum I/O throughput
- The maximum bus speed is largely limited by:
  - The length of the bus
  - The number of devices on the bus
  - The need to support a range of devices with:
    - Widely varying latencies
    - Widely varying data transfer rates

The General Organization of a Bus



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

Master versus Slave



- A bus transaction includes two parts:
  - Issuing the command (and address) – request
  - Transferring the data – action
- Master is the one who starts the bus transaction by:
  - issuing the command (and address)
- Slave is the one who responds to the address by:
  - Sending data to the master if the master ask for data
  - Receiving data from the master if the master wants to send data

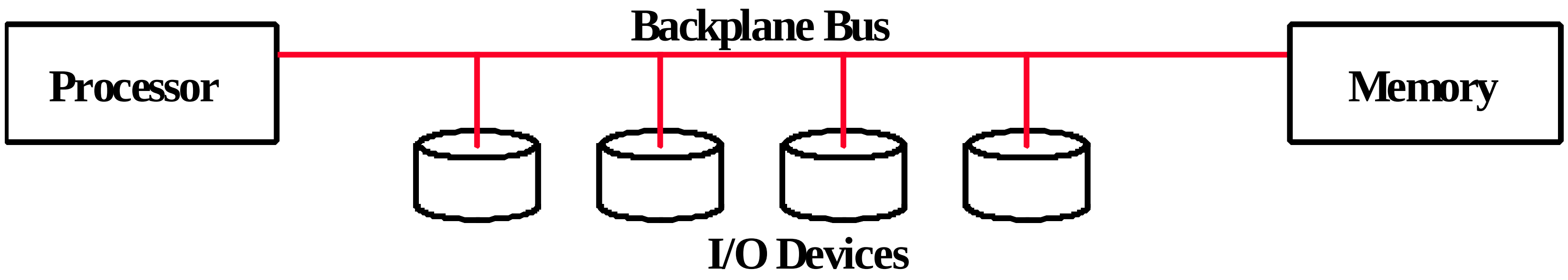
Types of Buses

- Processor-Memory Bus (design specific)
  - Short and high speed
  - Only need to match the memory system
  - Maximize memory-to-processor bandwidth
  - Connects directly to the processor
  - Optimized for cache block transfers
- I/O Bus (industry standard)
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus

Types of Buses

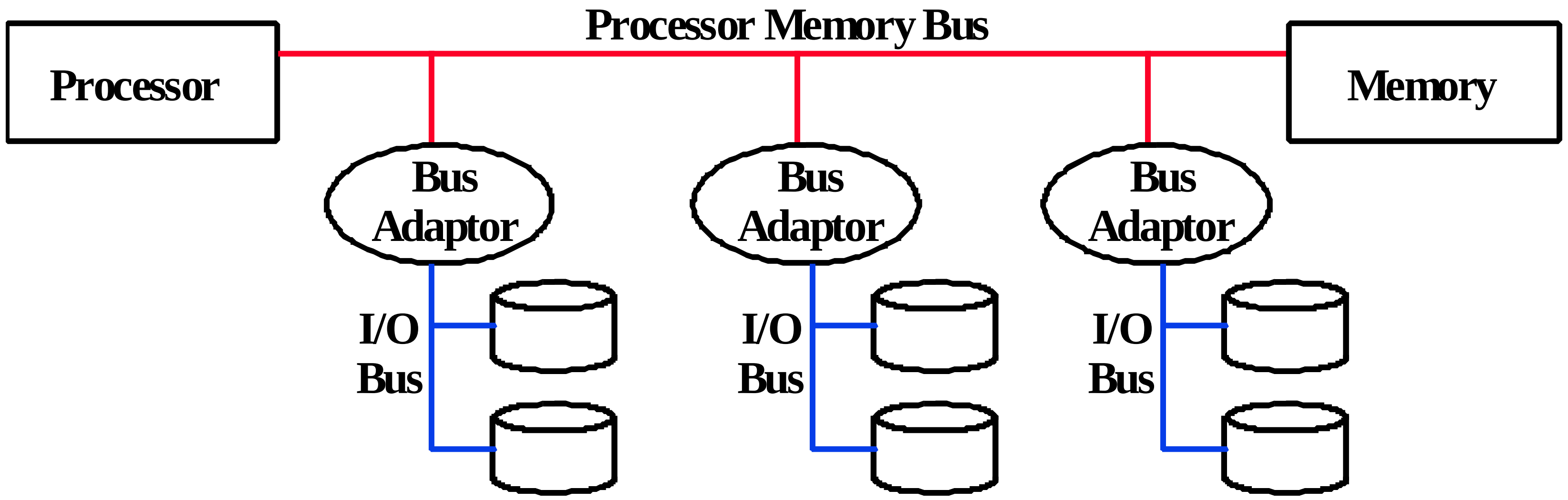
- Backplane Bus (standard or proprietary)
  - Backplane: an interconnection structure within the chassis
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one bus for all components

A Computer System with One Bus: Backplane Bus



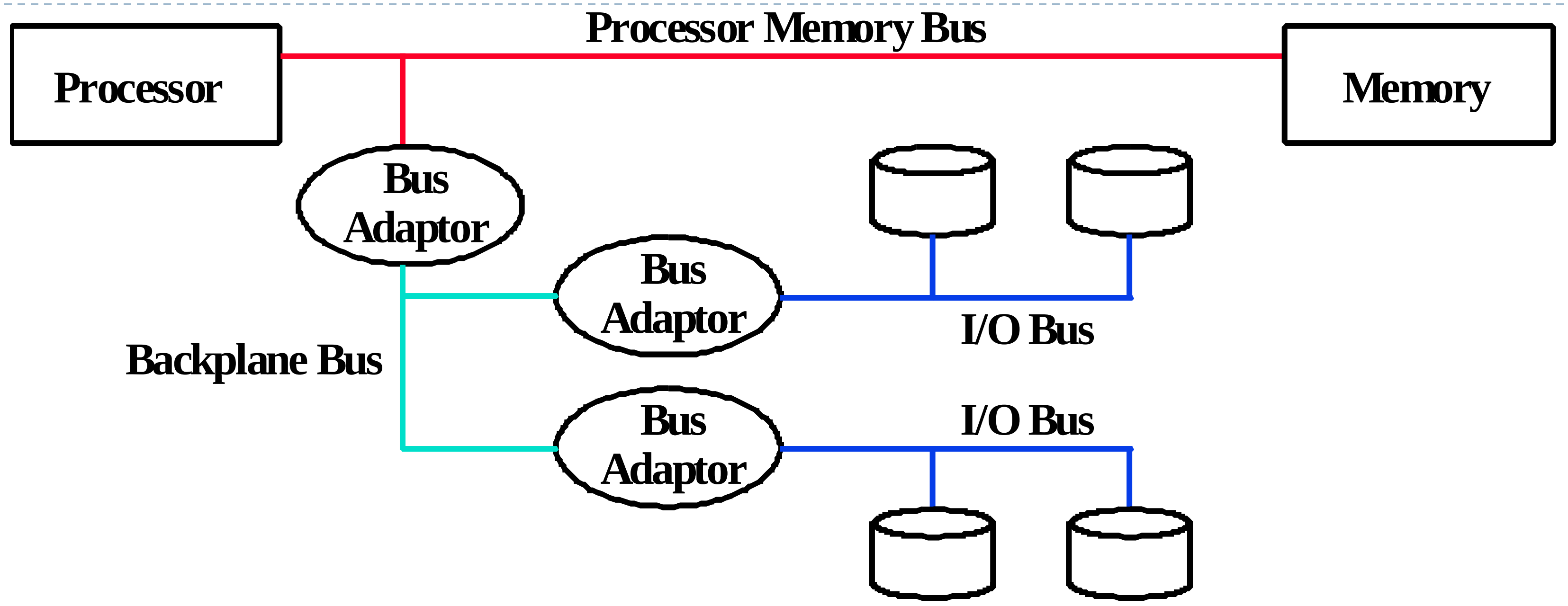
- A single bus (the backplane bus) is used for:
  - Processor to memory communication
  - Communication between I/O devices and memory
- Advantages: Simple and low cost
- Disadvantages: slow and the bus can become a major bottleneck
- Example: IBM PC - AT

A Two-Bus System



- I/O buses tap into the processor-memory bus via bus adaptors:
  - Processor-memory bus: mainly for processor-memory traffic
  - I/O buses: provide expansion slots for I/O devices
- Apple Macintosh-II
  - NuBus: Processor, memory, and a few selected I/O devices
  - SCCI Bus: the rest of the I/O devices

A Three-Bus System

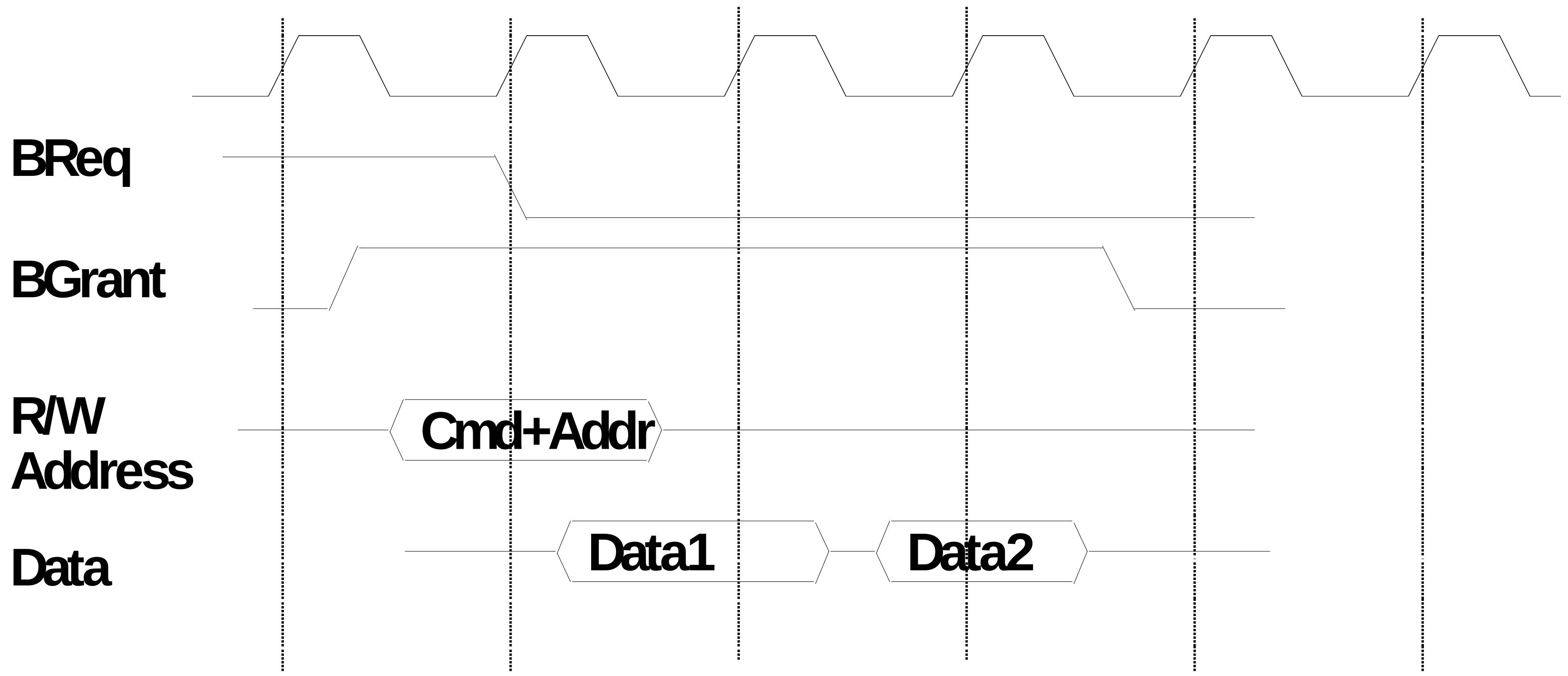


- A small number of backplane buses tap into the processor-memory bus
  - Processor-memory bus is used for processor memory traffic
  - I/O buses are connected to the backplane bus
- Advantage: loading on the processor bus is greatly reduced

Synchronous and Asynchronous Bus

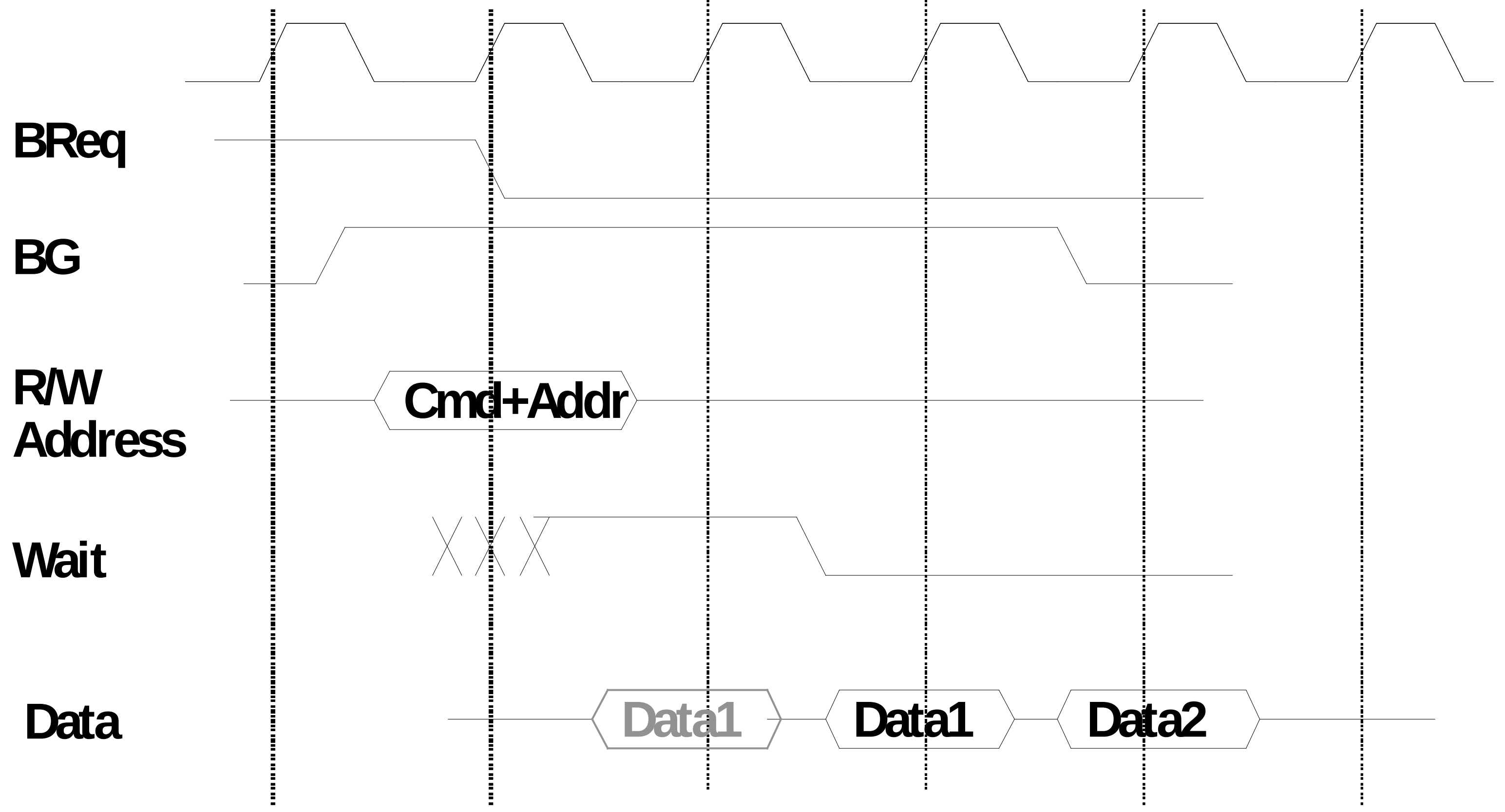
- Synchronous Bus:
  - Includes a clock in the control lines
  - A fixed protocol for communication that is relative to the clock
  - Advantage: involves very little logic and can run very fast
  - Disadvantages:
    - Every device on the bus must run at the same clock rate
    - To avoid clock skew,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

Simple Synchronous Protocol



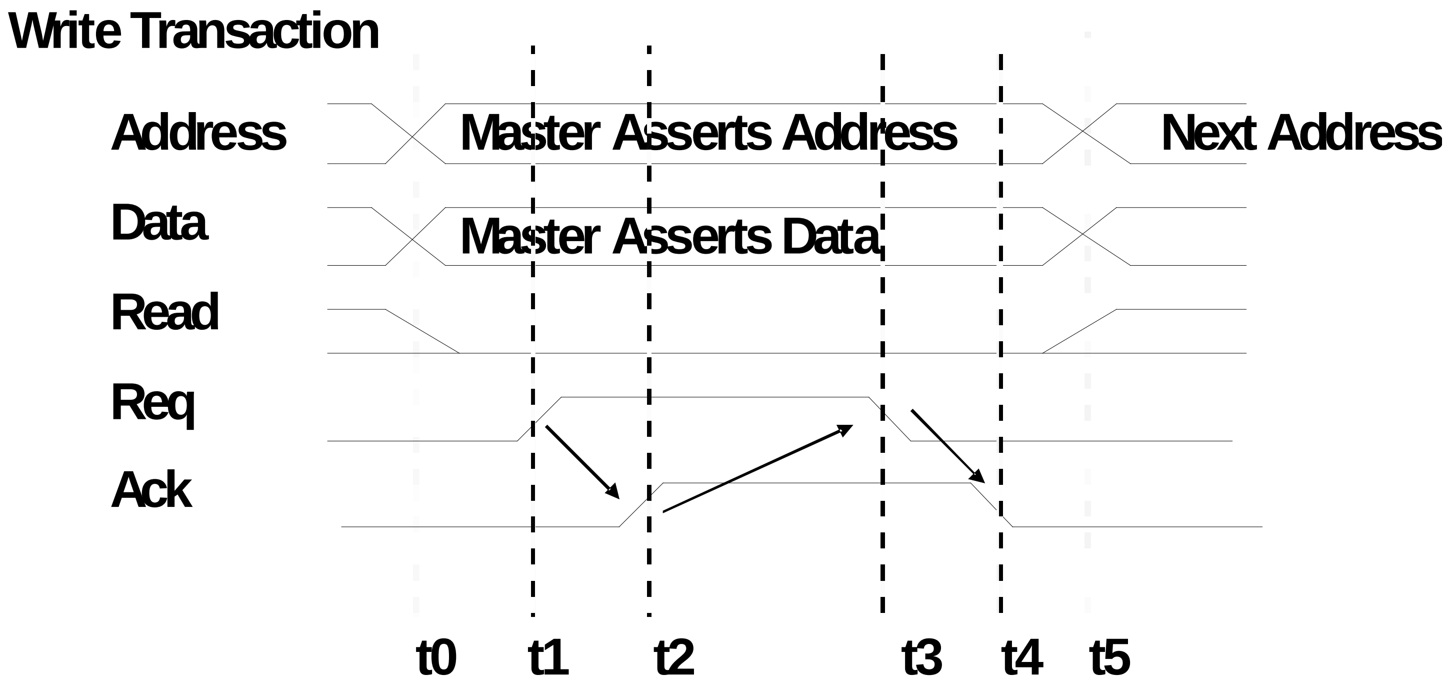
- Even memory busses are more complex than this
  - memory (slave) may take time to respond
  - need to control data rate

Typical Synchronous Protocol



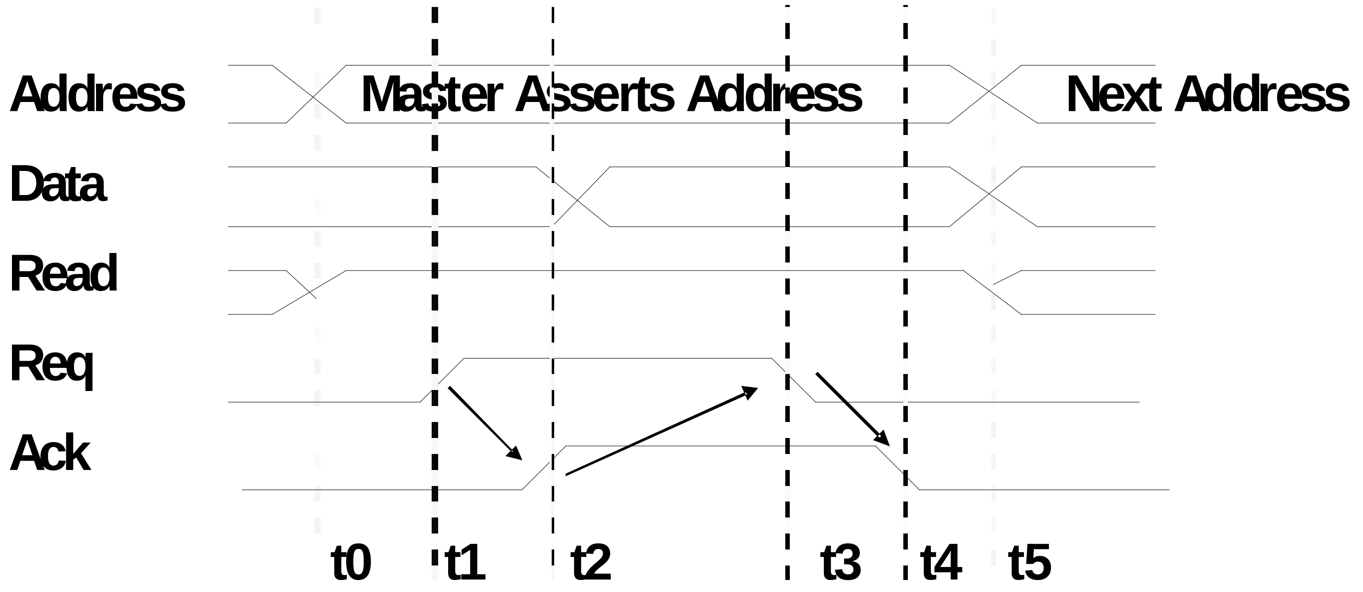
- Slave indicates when it is prepared for data transfer
- Actual transfer goes at bus rate

Asynchronous Handshake



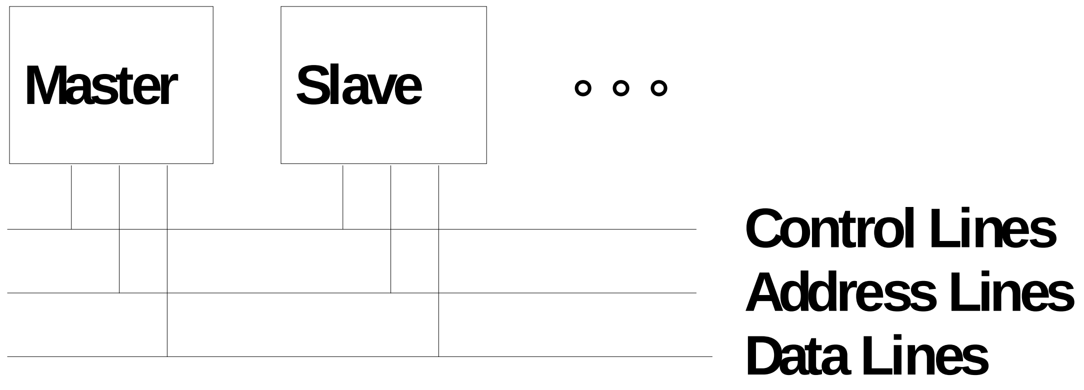
- t0 : Master has obtained control and asserts address, direction, data
  - Waits a specified amount of time for slaves to decode target
- t1: Master asserts request line
- t2: Slave asserts ack, indicating data received
- t3: Master releases req
- t4: Slave releases ack

Read Transaction



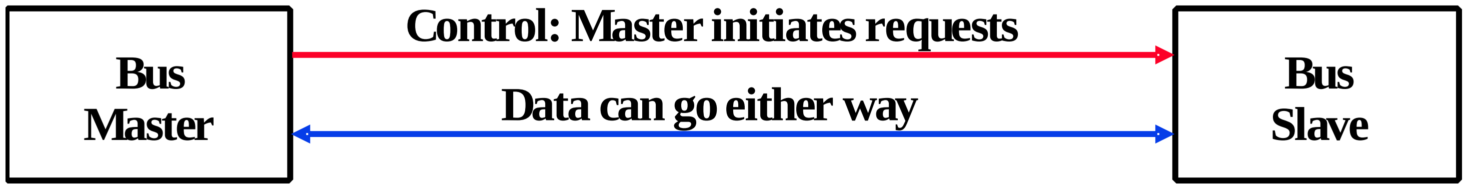
- t0 : Master has obtained control and asserts address, direction, data
  - Waits a specified amount of time for slaves to decode target\
- t1: Master asserts request line
- t2: Slave asserts ack, indicating ready to transmit data
- t3: Master releases req, data received
- t4: Slave releases ack

Busses so far



- Bus Master: has ability to control the bus, initiates transaction
- Bus Slave: module activated by the transaction
- Bus Communication Protocol: specification of sequence of events and timing requirements in transferring information.
- Asynchronous Bus Transfers: control lines (req, ack) serve to orchestrate sequencing.
- Synchronous Bus Transfers: sequence relative to common clock

Arbitration: Obtaining Access to the Bus



- One of the most important issues in bus design:
  - How is the bus reserved by a devices that wishes to use it?
- Chaos is avoided by a master-slave arrangement:
  - Only the bus master can control access to the bus:
    - It initiates and controls all bus requests
    - A slave responds to read and write requests
- The simplest system:
    - Processor is the only bus master
    - All bus requests must be controlled by the processor
    - Major drawback: the processor is involved in every transaction

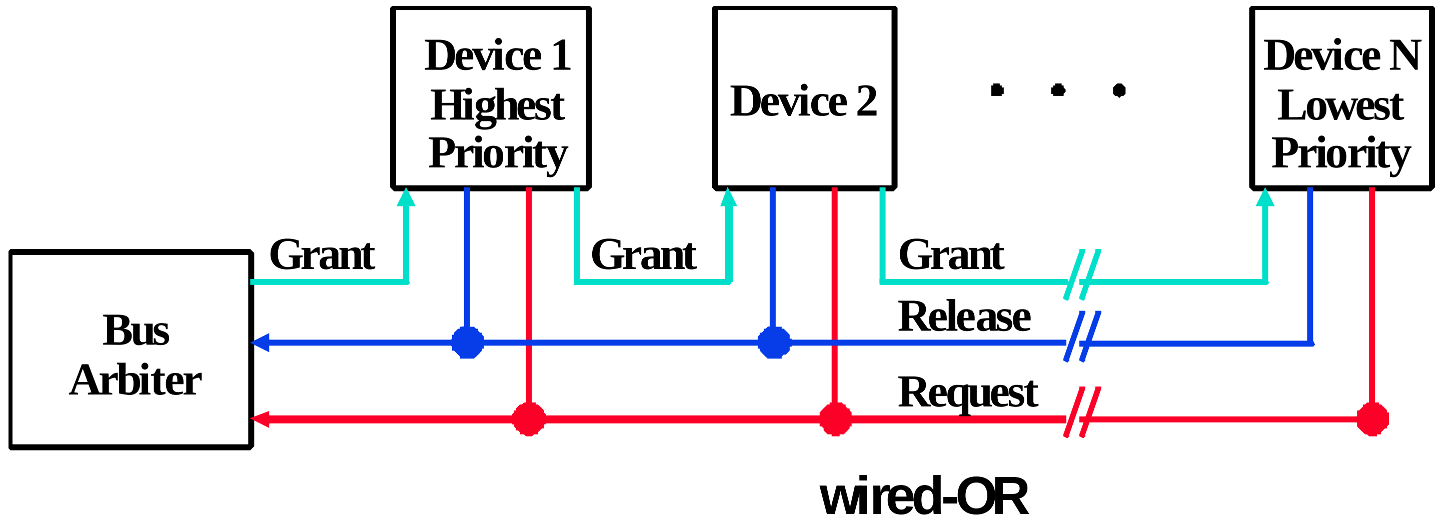
Multiple Potential Bus Masters: the Need for Arbitration

- Bus arbitration scheme:
  - A bus master wanting 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
- Bus arbitration schemes usually try to 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 the bus

Multiple Potential Bus Masters: the Need for Arbitration

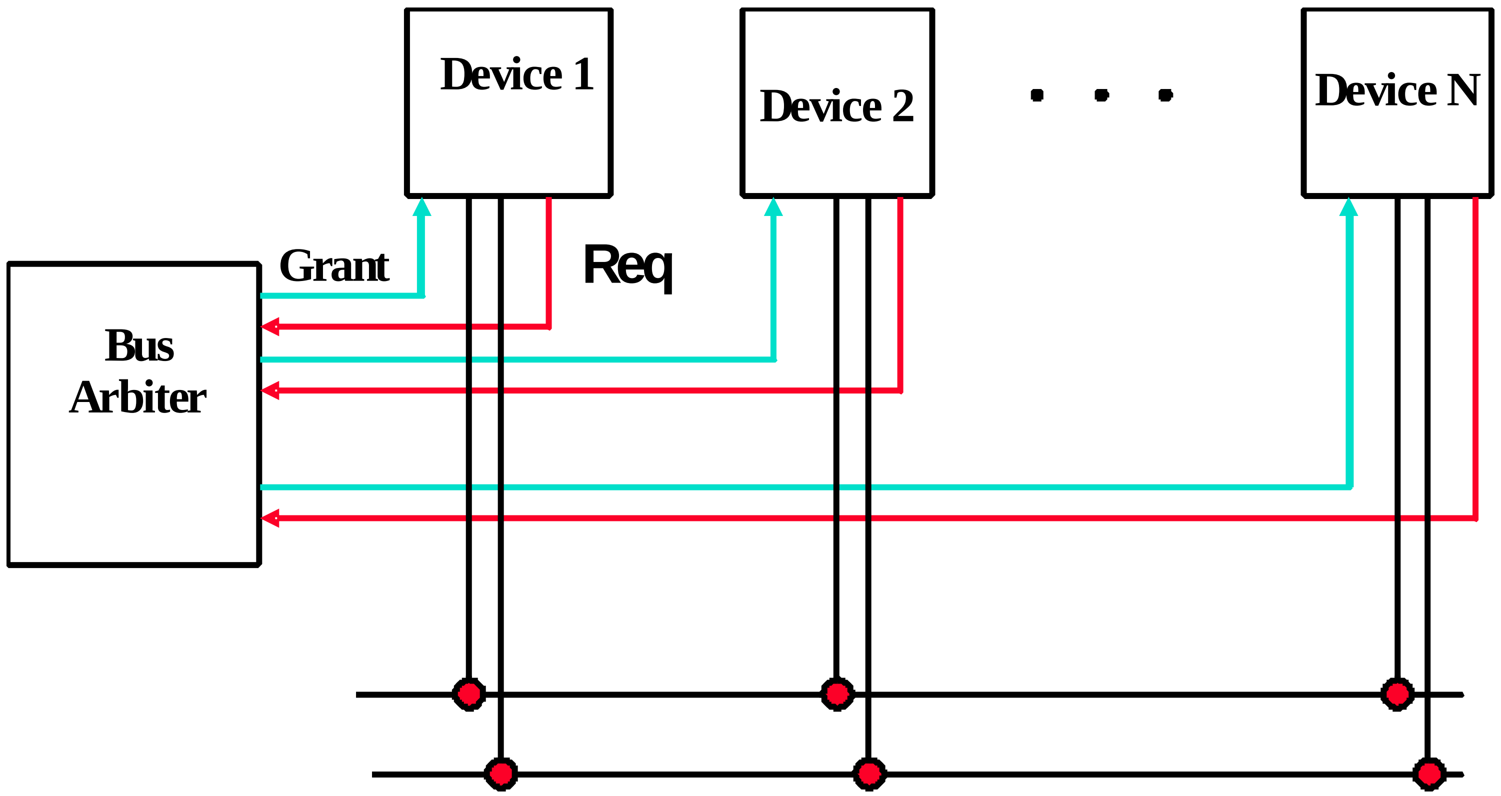
- Bus arbitration schemes can be divided into four broad classes:
  - Daisy chain arbitration: single device with all request lines.
  - Centralized, parallel arbitration: see next-next slide
  - Distributed arbitration by self-selection: each device wanting the bus places a code indicating its identity on the bus.
  - Distributed arbitration by collision detection: Ethernet uses this.

The Daisy Chain Bus Arbitration Scheme



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

Centralized Parallel Arbitration



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

Increasing the Bus Bandwidth

- Separate versus multiplexed address and data lines:
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available
  - Cost: (a) more bus lines, (b) increased complexity
- Data bus width:
  - By increasing the width of the 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 the Bus Bandwidth

- Block transfers:
  - Allow the bus to transfer multiple words in back-to- back bus cycles
  - Only one address needs to be sent at the beginning
  - The bus is not released until the last word is transferred
  - Cost: (a) increased complexity

    (b) decreased response time for request