

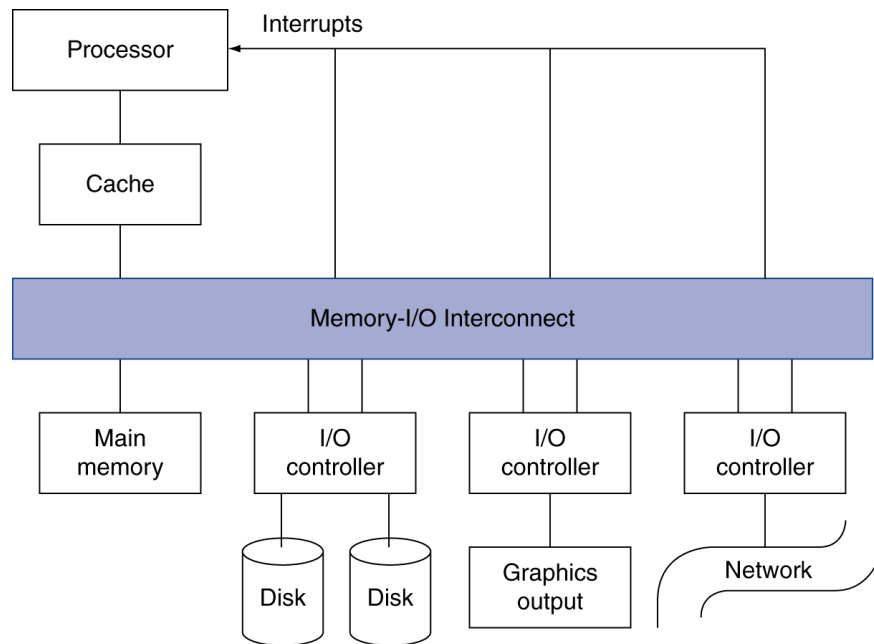


Topic 15

I/Os and Their Interfaces

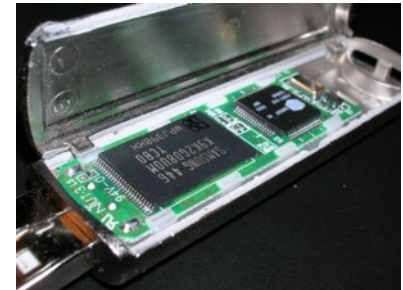
Introduction

- I/O devices can be characterized by
 - Behavior: input, output, storage
 - Partner: human or machine
 - Data rate: bytes/sec, transfers/sec
- I/O bus connections



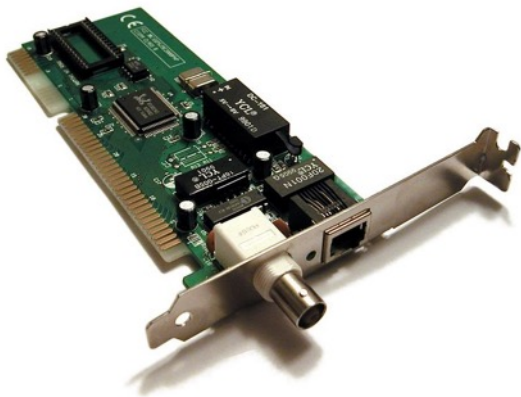
Peripheral – Memory

- Volatile main memory
 - Loses instructions and data when power off
- Non-volatile secondary memory
 - Magnetic disk
 - Flash memory
 - Optical disk (CDROM, DVD)

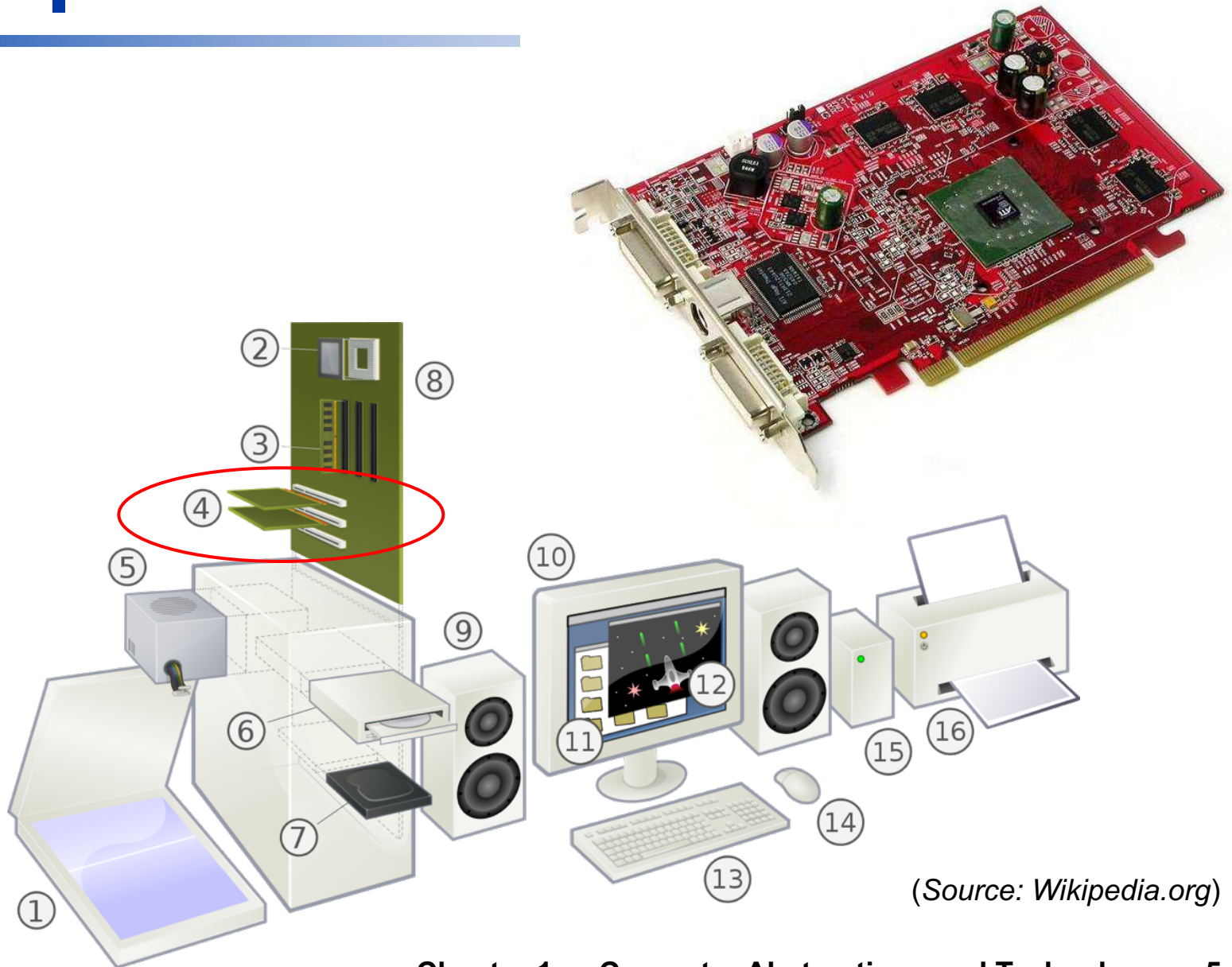


Peripheral – Networks

- Communication and resource sharing
- Local area network (LAN): Ethernet
 - Within a building
- Wide area network (WAN): the Internet
- Wireless network: WiFi, Bluetooth



Peripheral – others



(Source: Wikipedia.org)

I/O System Characteristics

- Dependability
 - Very important
 - Particularly for storage devices
- Performance measures
 - Latency (response time)
 - Desktops & embedded systems – mainly interested in response time & diversity of devices
 - Throughput (bandwidth)
 - Servers – mainly interested in throughput & expandability of devices

I/O Management

- I/O is managed by the Operating System (OS)
 - Multiple programs share I/O resources
 - Need protection and scheduling
 - Done by OS in supervisor mode
 - I/O causes asynchronous interrupts to communicate operation information with CPU
 - Same mechanism as exceptions
 - Interrupt service routine – part of OS
 - I/O programming is non-trivial and sophisticated
 - OS provides abstractions (interfaces) to programs
 - API – Application Programming Interface

I/O Control Register

- I/O devices are controlled by a set of registers
 - Command, Status, Data
- Command registers
 - Cause device to do something
- Status registers
 - Indicate what the device is doing or has done and occurrence of errors
- Data registers
 - Write: transfer data to an I/O device
 - Read: transfer data from an I/O device

OS (sw) & I/O (hw) Interface

- Memory mapped I/O
 - I/O registers are connected to memory locations
 - I/Os are accessed as regular memory locations
 - Accessible from software by virtual memory addresses
 - OS writes/reads memory to operate I/O devices
 - OS uses address translation mechanism to make them only accessible in kernel mode
 - Virtual address translation only accessible to OS
- I/O instructions
 - Separate instructions to access I/O registers
 - Can only be executed in kernel mode (by OS)

I/O & Processor Communication – Polling

- Periodically check I/O status register
 - If device ready, do operation
 - If error, take action
- Common in small or low-performance real-time embedded systems
 - Predictable timing
 - Low hardware cost
- In other systems, wastes CPU time

I/O & Processor Communication – Interrupt

- When a device is ready or error occurs
 - Controller interrupts CPU – by hardware
 - Will invoke handler between instructions
 - Not synchronized to instruction execution
- Interrupt Priority
 - Devices needing more urgent attention get higher priority
 - Higher priority interrupt can interrupt execution of a lower priority interrupt

Hardware Support for Interrupt in MIPS

- Status register enables interrupts and configures interrupt priority hierarchy
- Cause register used to identify and properly handle the interrupting device
- Configuration and control registers for interrupts
- Memory support – interrupt vector
 - Entrance of interrupt service routine

I/O Data Transfer

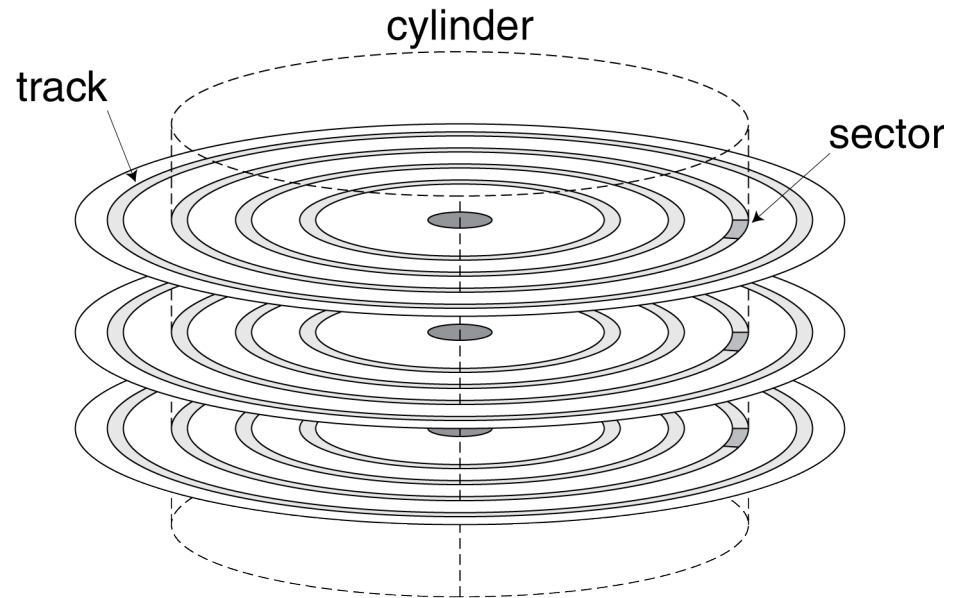
- I/O only transfers data to/from memory
 - Supervised by processor (OS), or
 - Initiated and validated by processor
- Polling and interrupt-driven I/O
 - CPU transfers data between memory and I/O data registers
 - Consumes CPU time
- Direct memory access (DMA)
 - Processor sets up DMA controller by providing device ID, starting address in memory, number of bytes to transfer, triggering events
 - I/O controller interrupt or CPU (software) requests data transfer to start
 - DMA controller transfers between memory & I/O device autonomously without supervision of CPU
 - DMA Controller interrupts to call CPU attention on completion or critical events or error

DMA/Cache Interaction

- DMA problem – stale data problem
 - If DMA writes to a memory block that is also in cache, cached copy becomes obsolete
 - If DMA reads memory block while corresponding cache is updated (due to write-back)
- Solutions:
 - Route I/O devices to memory through cache
 - Use cache for infrequently used I/O is expensive
 - Assistance from OS
 - Intentionally avoid reading cache for I/O memory locations
 - Force write-back for I/O write

Disk Storage

- Nonvolatile, rotating magnetic storage



Disk Sectors and Access

- Each sector records
 - Sector ID
 - Data (512 bytes)
 - Error correcting code (ECC)
 - Used to hide defects and recording errors
 - Synchronization fields and gaps
- Access to a sector involves
 - Queuing delay if other accesses are pending
 - Seek: move the heads
 - Rotational latency
 - Data transfer
 - Controller overhead

Disk Performance Issues

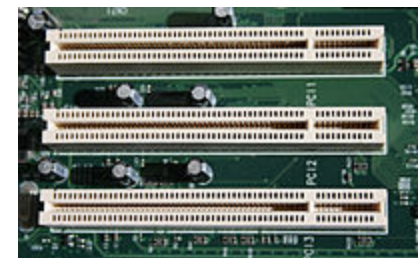
- Has smart disk controller to create simpler interface
 - With microprocessor inside
 - Present logical sector interface to host
 - Various controller interfaces:
 - SCSI – small computer system interface
 - ATA – AT attachment
 - SATA – Serial ATA
 - PCI – peripheral controller interface
 - PCI x – eXtended, PCI Express
 - LPC – low pin count bus
- Disk controllers include caches
 - Write through
 - Pre-fetch sectors in anticipation of access
 - Avoid seek and rotational delay



SCSI connector



ATA connector



PCI socket

Interconnecting Components

- Need interconnections between
 - CPU, memory, and I/O controllers
 - Using buses
- Bus: shared communication channel
- Parallel set of wires for data and synchronization of data transfer
 - Advantages:
 - Versatility – various functions, easy to be added or removed
 - Low cost
 - Concerns: performance limited by physical factors
 - Bus speed – can become a communication bottleneck
 - Wire length, number of connections
- More recent alternative: high-speed serial connections

Bus Types

- Processor-Memory buses (North connection)
 - Short, high speed
 - Designed to match memory organization
- I/O-Memory buses (South connection)
 - Longer, allowing multiple connections
 - Specified by standards for interoperability
 - Connected through a north bridge then to memory

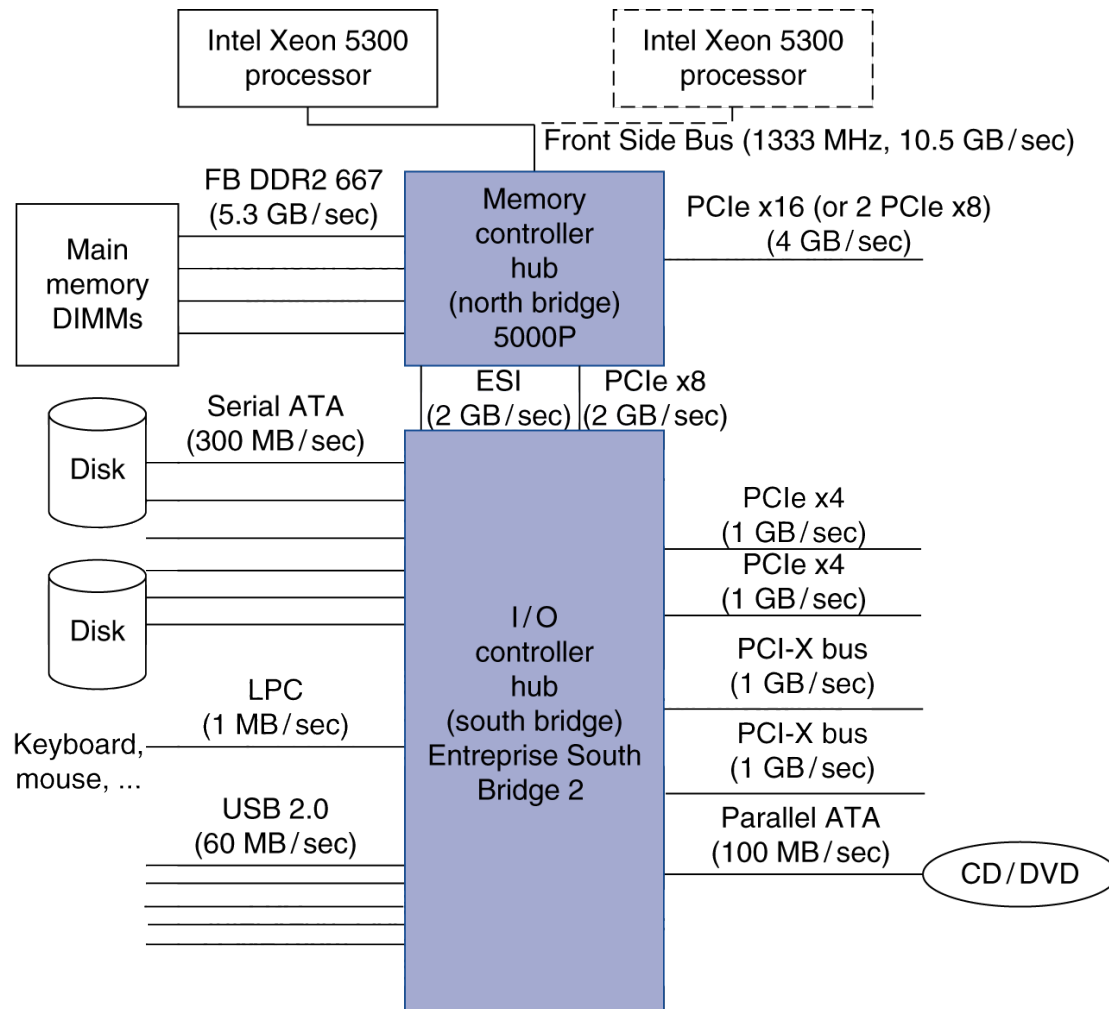
Bus Types

- Data/Address bus
 - Carry data/address, respectively
 - Multiplexed or separate
- Control bus
 - Indicate data type, synchronize transactions
 - Synchronous – uses a separate clock line
 - Asynchronous – synchronization integrated in data
- Communication standard
 - Coordinate communications
 - Ensure compatibility
 - E.g. RS232, 802.11, 802.15.1, 802.15.4, USB...

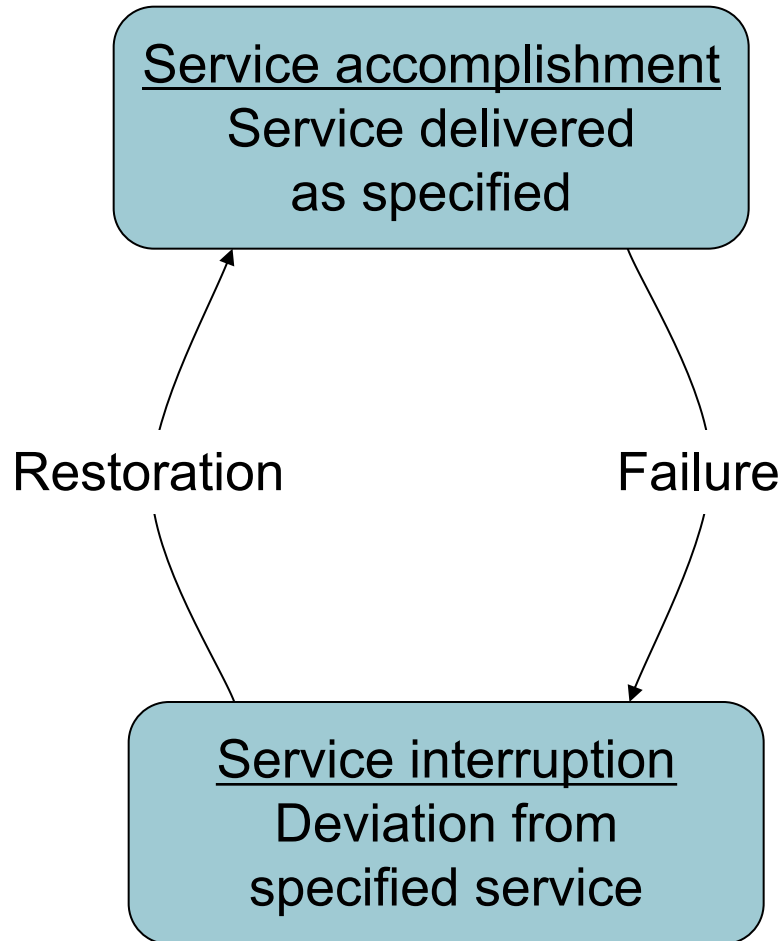
I/O Bus Examples

	Firewire	USB 2.0	USB 3.1	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	External	Internal	Internal	External
Devices per channel	63	127	127	1	1	4
Data width	4	2	2	2/lane	4	4
Peak bandwidth	50MB/s or 100MB/s	0.2MB/s, 1.5MB/s, or 60MB/s	1GB/s	250MB/s/lane 1×, 2×, 4×, 8×, 16×, 32×	300MB/s	300MB/s
Hot pluggable	Yes	Yes	Yes	Depends	Yes	Yes
Max length	4.5m	5m	3m	0.5m	1m	8m
Standard	IEEE 1394	USB Implementers Forum	USB Imp. Forum	PCI-SIG	SATA-IO	INCITS TC T10

Typical x86 PC I/O System



Two System States



- Failure of a component
 - May or may not lead to system failure

Dependability Measures

- Dependability
 - Reliability
 - Availability
- Measures
 - Reliability: Mean time to failure (MTTF)
 - Time spent in “service accomplishment”
 - Mean time to repair (MTTR)
 - Time spent in “service interruption”
 - Mean time between failures: $MTBF = MTTF + MTTR$
 - Availability = $MTTF / MTBF$
- Improving Availability
 - Increase MTTF: fault avoidance, fault tolerance, fault forecasting
 - Reduce MTTR: improved tools and processes for diagnosis and repair