ECE/CS 552: Intro to Computer Architecture

Instructor: Mikko Lipasti

TA: Guangyu Shi

Input/Output

- Motivation
- Disks
- Networks
- Buses
- Interfaces
- Examples

Input/Output

- I/O necessary
  - To/from users (display, keyboard, mouse)
  - To/from non-volatile media (disk, tape)
  - To/from other computers (networks)
- Key questions
  - How fast?
  - Getting faster?

Typical Collection of I/O Devices

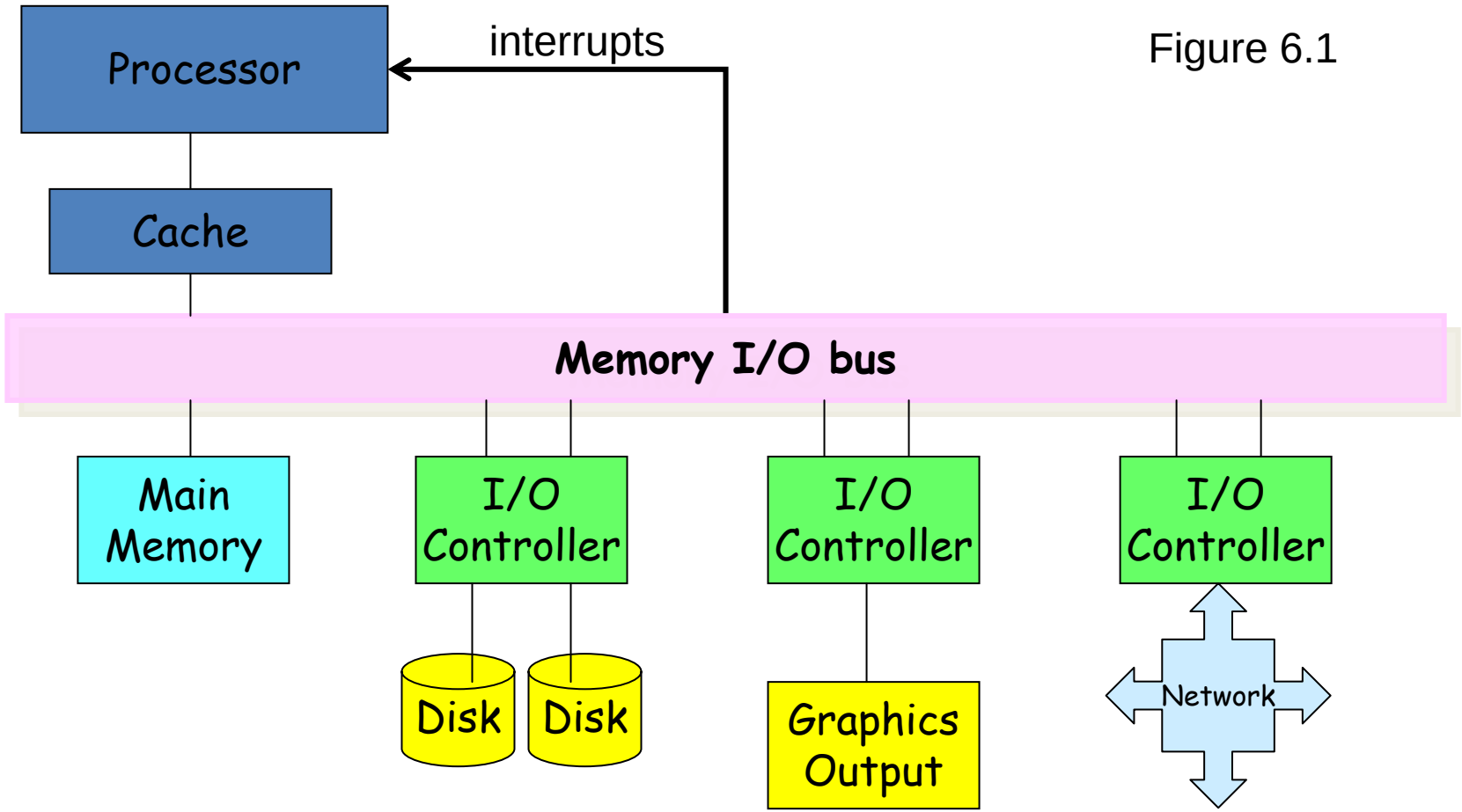


Figure 6.1

Examples

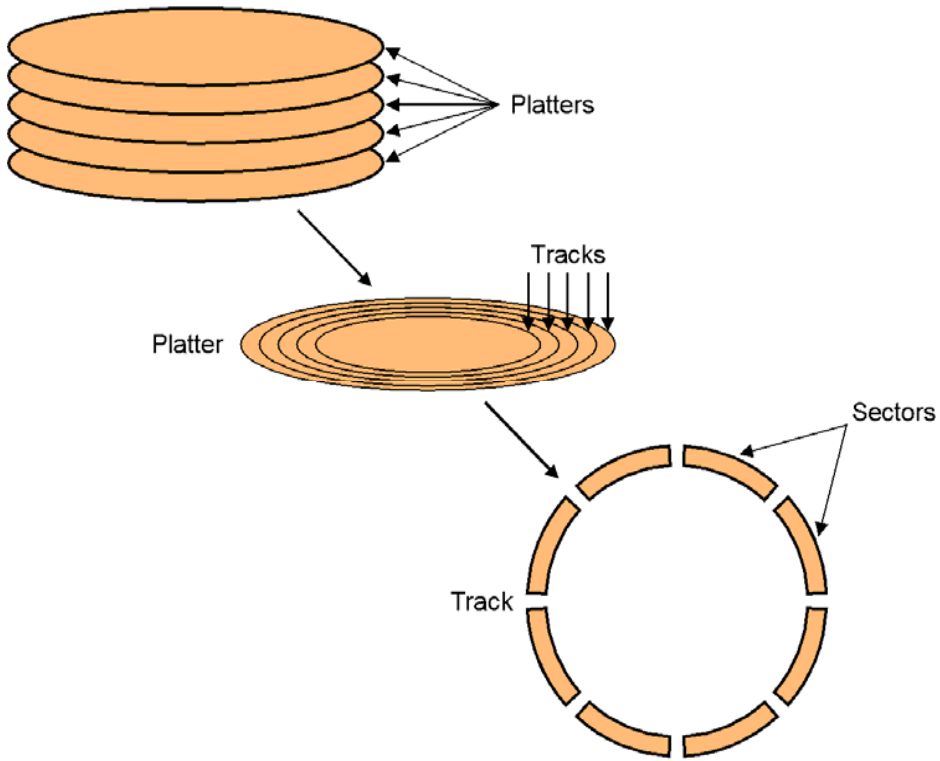
Device	I or O?	Partner	Data Rate KB/s
Mouse	I	Human	0.01
Display	O	Human	60,000
Modem	I/O	Machine	2-8
LAN	I/O	Machine	10000
Tape	Storage	Machine	2000
Disk	Storage	Machine	2000-100,000

I/O Performance

- What is performance?
- Supercomputers read/write 1GB of data
  - Want high bandwidth to vast data (bytes/sec)
- Transaction processing does many independent small I/Os
  - Want high I/O rates (I/Os per sec)
  - May want fast response times
- File systems
  - Want fast response time first
  - Lots of locality

Magnetic Disks

- Stack of platters
- Two surfaces per platter
- Tracks
- Heads move together
- Sectors
- Disk access
  - Queueing + seek
  - Rotation + transfer



Magnetic Disks

- How it works.

http://www.youtube.com/watch?v=9eMWG3fwiEU&feature=related

http://www.youtube.com/watch?v=_NLW4sU3DN0

Disk Trends

- Disk trends
  - $/MB down (well below $1/GB)
  - Disk diameter: 14” => 3.5” => 1.8” => 1”
  - Seek time down
  - Rotation speed increasing at high end
    - 5400rpm => 7200rpm => 10Krpm => 15Krpm
    - Slower when energy-constrained (laptop, IPod)
  - Transfer rates up
  - Capacity per platter way up (100%/year)
  - Hence, op/s/MB way down
    - High op/s demand forces excess capacity

Flash Storage

- Flash memory
  - A type of EEPROM
- possible substitute of disk
  - Nonvolatile
  - 100-1000 times faster than disks
  - Small, power efficient & shock resistant
- Popular in mobile devices

Flash Storage

- Disadvantage: wear out
  - Not so popular for desktop and servers
- Solution: wear leveling
  - one block with a specially extended life of 100,000+ cycles (regular: ~1000 cycles)
  - erasures and re-writes are distributed evenly across the medium

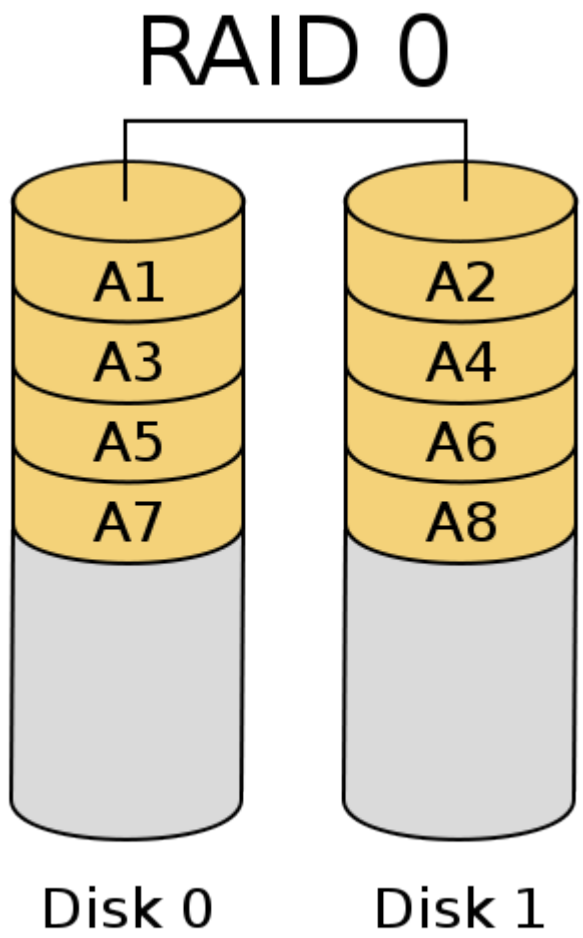
Types of Storage

Type	Size	Speed	Cost/bit
Register	< 1KB	< 1ns	$$$$
On-chip SRAM	8KB-6MB	< 10ns	$$$
Off-chip SRAM	1Mb – 16Mb	< 20ns	$$
DRAM	64MB – 1TB	< 100ns	$
Flash	64MB – 32GB	< 100us	~c
Disk	40GB – 1PB	< 20ms	~0

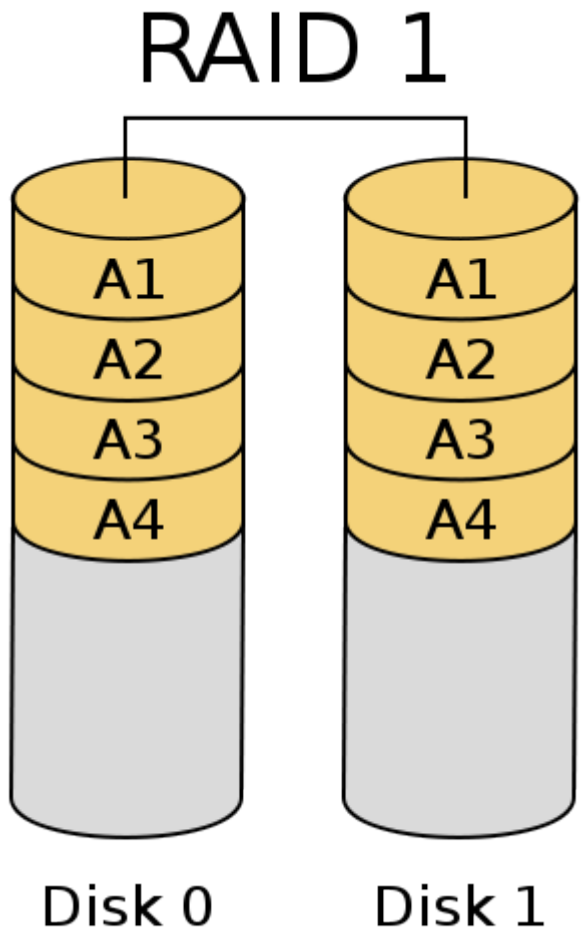
RAID:
Redundant Array of Inexpensive Disks

- Dependability
  - Reliability: Measured by MTTF (mean time to failure)
  - Service interruption: measured by MTTR (mean time to repair)
  - Availability = MTTF/(MTTF + MTTR)
- What if we need 100 disks for storage?
- MTTF = 5 years / 100 = 18 days!
- RAID 0
  - Data striped (spread over multiple disks), but no error protection
- RAID 1
  - Mirror = stored twice = 100% overhead (most expensive)
- RAID 5
  - Block-wise parity = small overhead and small writes

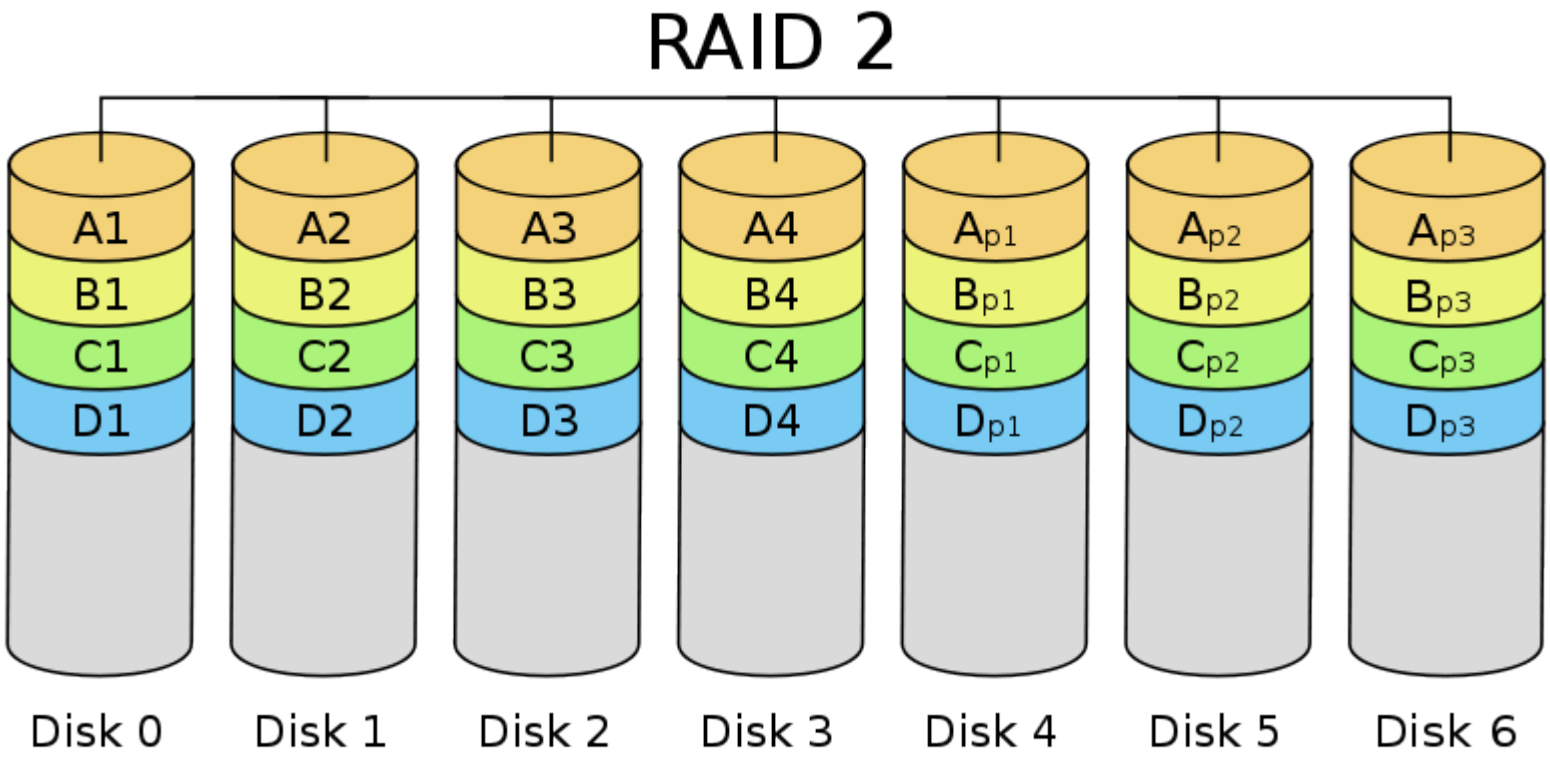
RAID 0: Block-level stripping



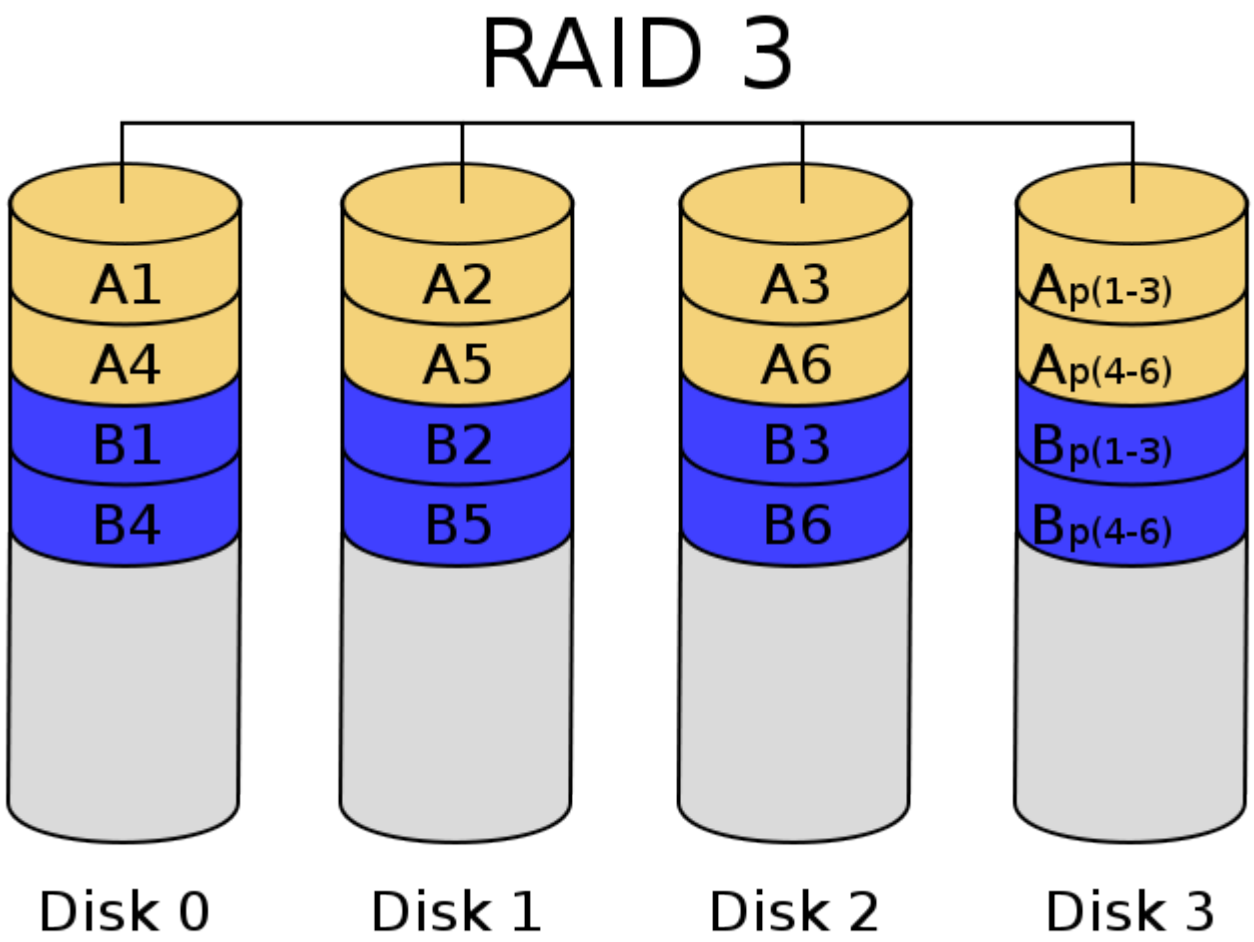
RAID 1: Mirroring



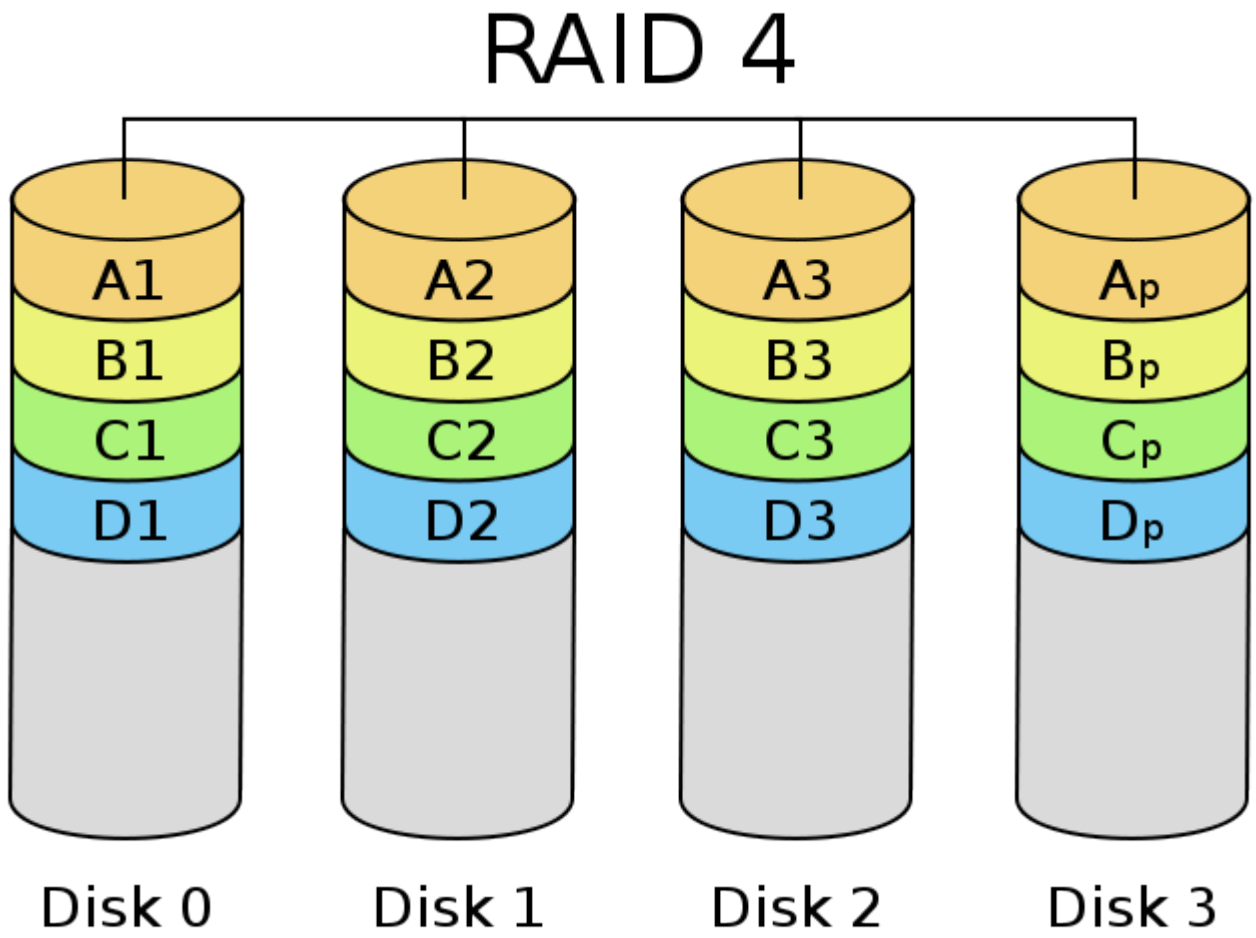
RAID 2: Bit-level Interleave with ECC



RAID 3: Byte-level Interleave

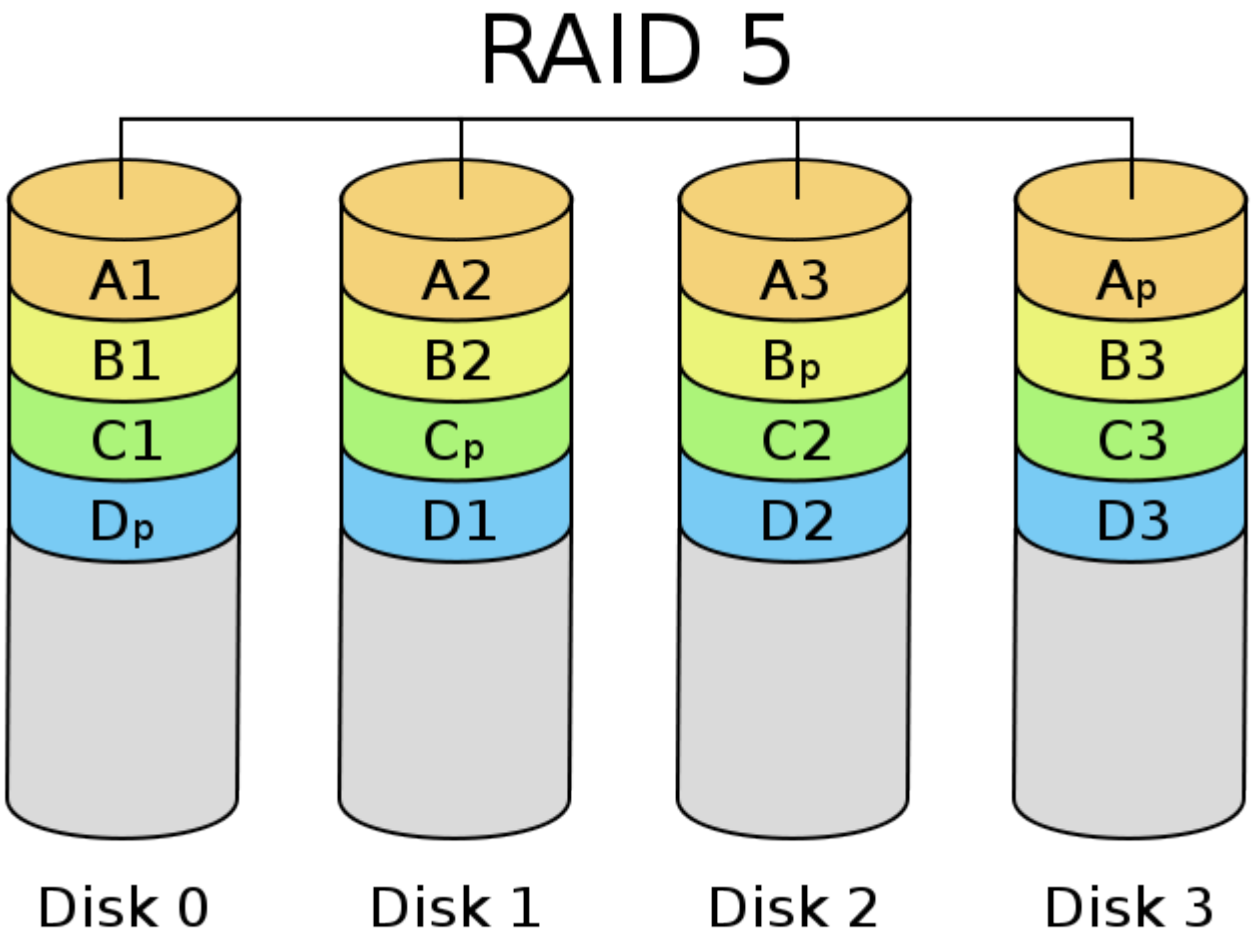


RAID 4: Block-level Interleave



http://en.wikipedia.org

RAID 5: Block-level interleaved distributed parity



RAID Illustrations

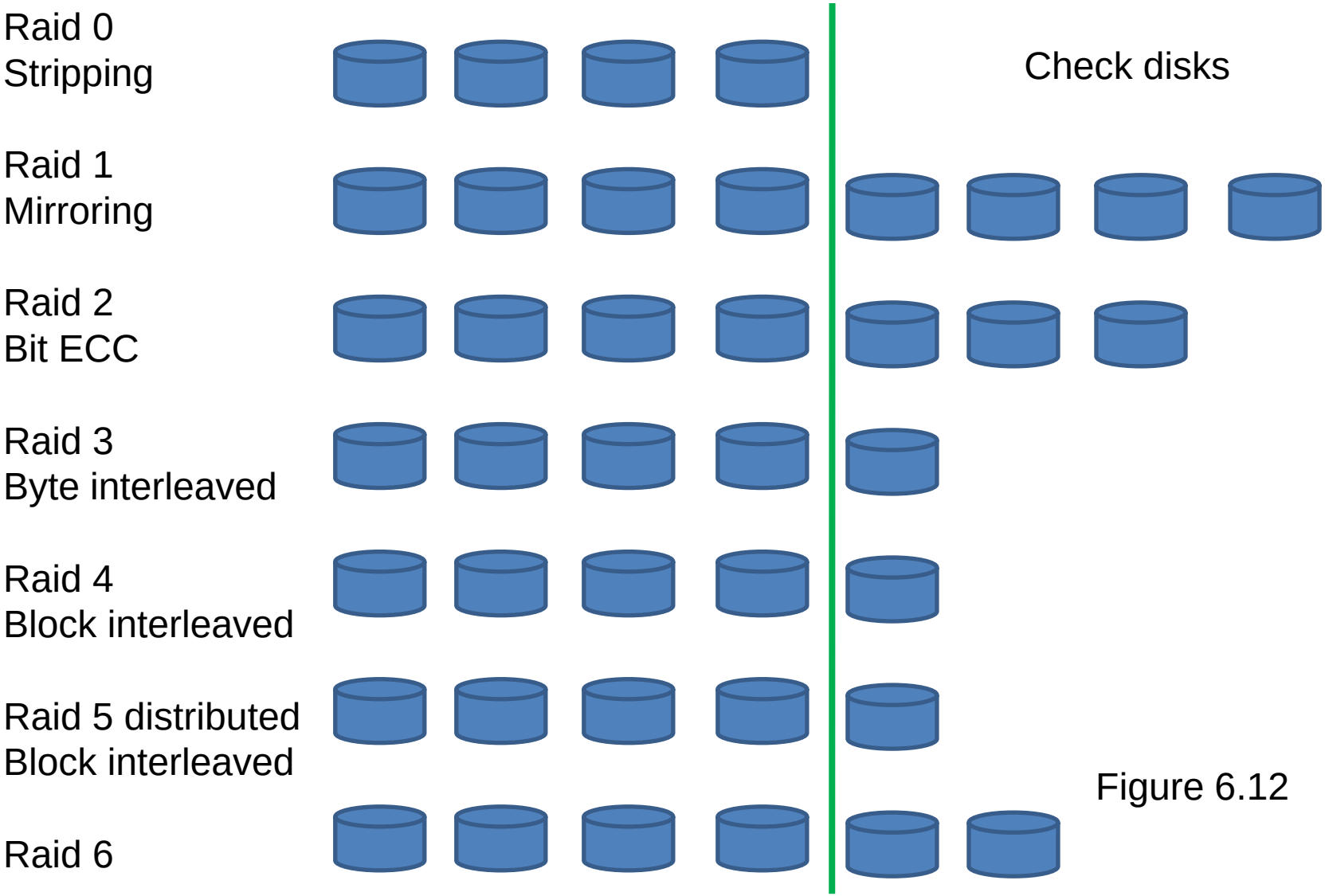


Figure 6.12

Tape

- Helical scan
  - 8mm video tape + ECC
  - 7GB/tape at $6/tape = <$1/GB
    - Note similar to cheap IDE hard drives!
  - Tape robots
- E.g. Library of Congress is 10TB text
  - 1500 tapes x $6 = $9000
  - Of course, not that simple

- Helica
  - 8mn
  - 7GB
    - N
  - Tape
- E.g. Li
- 1500 t
  - Of c

Frame Buffer

- Extreme bandwidth requirement
  - 1560x1280 pixels x 24bits/pixel = 5.7MB
  - Refresh whole screen 30 times/sec = 170MB/s > PCI
- On memory bus
  - Use 24 video DRAMs (dual ported)
    - Refresh display and allow image change by CPU
    - DRAM port
    - Serial port to video
- Also AGP (Accelerated Graphics Port)
  - Video card talks directly to DRAM (not thru PCI)

LAN = Ethernet

- Original Ethernet
  - One-write bus with collisions and exponential back-off
  - Within building
  - 10Mb/s (~= 1MB/s)
- Now Ethernet is
  - Point to point clients (switched network)
  - Client s/w, protocol unchanged
  - 100Mb/s => 1Gb/s

LAN

- Ethernet not technically optimal
  - 80x86 is not technically optimal either!
- Nevertheless, many efforts to displace it have failed (token ring, ATM)
- Emerging: System Area Network (SAN)
  - Reduce SW stack (TCP/IP processing)
  - Reduce HW stack (interface on memory bus)
  - Standard: Infiniband (http://www.infinibandta.org)

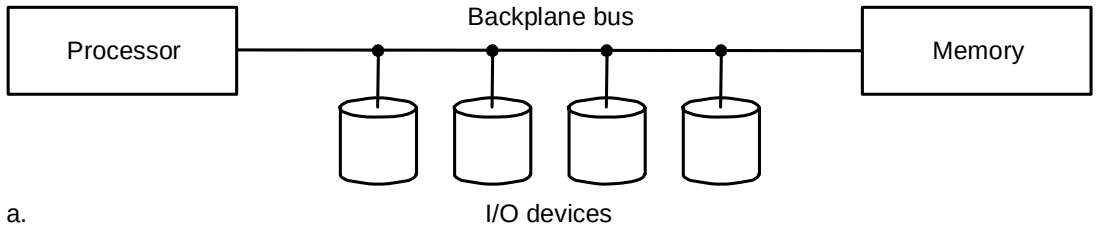
Bus

- A shared communication link between processor and memory, and I/O devices
- Consisting of control lines and data lines
  - Control: determine which device gets to access the bus, what type of information on data lines
  - Data: address, command, data to and from devices
- Function:
  - Access control, arbitration: right to use bus
  - Ensure safe transaction in asynchronous transfer using hand-shaking protocol, and/or ECC

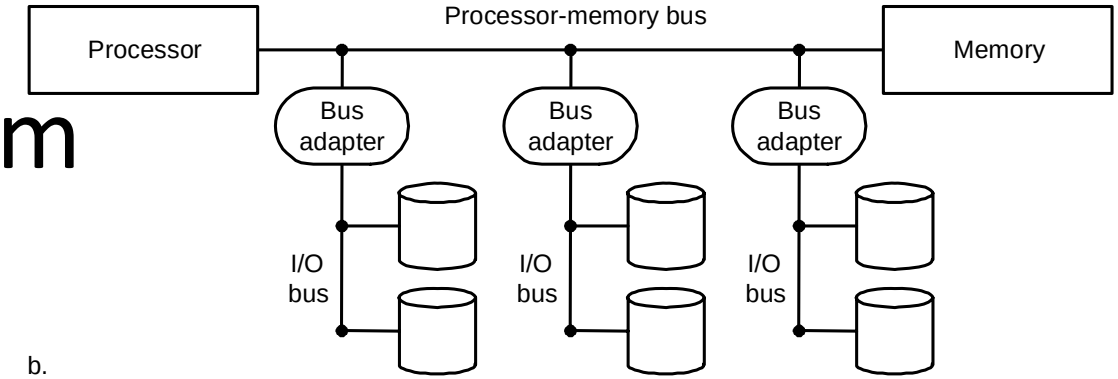
Buses

- Bunch of wires
  - Arbitration
  - Control/command
  - Data
  - Address
  - Flexible, low cost
  - Can be bandwidth bottleneck
- Types
  - Processor-memory
    - Short, fast, custom
  - I/O
    - Long, slow, standard
  - Backplane
    - Medium, medium, standard

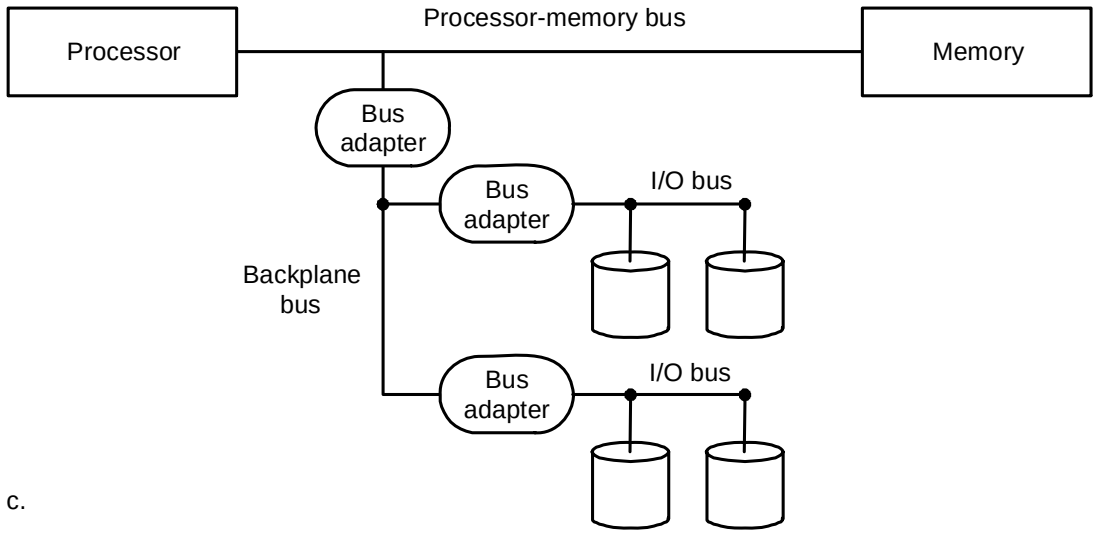
Buses in a Computer System



a.



b.

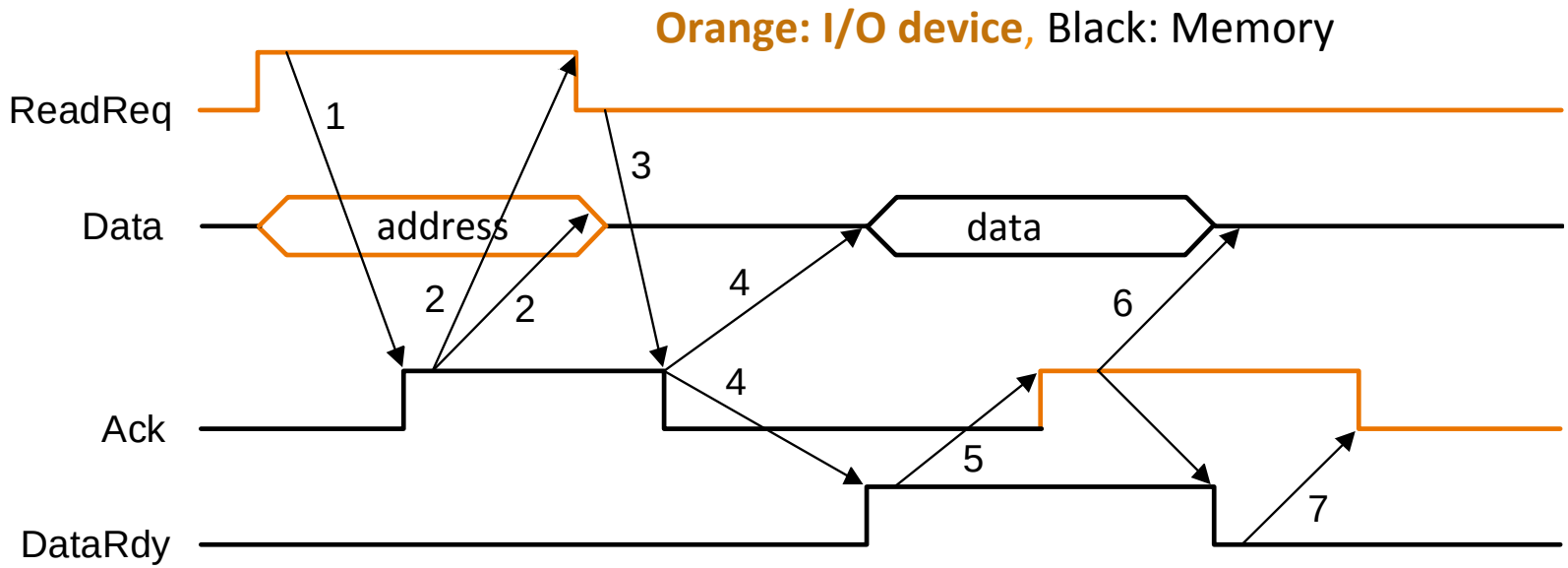


c.

Buses

- Synchronous – has clock
  - Everyone watches clock and latches at appropriate phase
  - Transactions take fixed or variable number of clocks
  - Faster but clock limits length
  - E.g. processor-memory
- Asynchronous – requires handshake
  - More flexible
  - I/O
  - Handshaking protocol: A series of steps used to coordinate asynchronous bus transfers in which the sender and receiver proceed to the next step only when both parties agree that the current step has been completed.

Async. Handshake (Fig. 8.10)



(1) Request made by I/O device & (2) ack send by Memory

(3) Request deasserted & (4) ack deasserted

Wait till memory has data ready, it raise data ready and place data on data bus

(5) Data sent & (6) Data rec'd by I/O device and raise ack line & (7) ack deasserted and data off line by memory

Buses

- Improving bandwidth
  - Wider bus
  - Separate/multiplexed address/data lines
  - Block transfer
    - Spatial locality

Bus Arbitration

- Resolving bus control conflicts and assigning priorities to the requests for control of the bus.
- One or more bus masters, others slaves
  - Bus request
  - Bus grant
  - Priority
  - Fairness
- Implementations
  - Centralized (Arbiter, can be part of CPU or separate device)
  - Distributed (e.g. Ethernet)

Bus Mastering

- Allows bus to communicate directly with other devices on the bus without going through CPU
- Devices capable to take control of the bus.
- Frees up the processor (to do other work simultaneously)

Buses

- Bus architecture / standards
  - ISA (Industry Standard Architecture)
  - MCA (Micro Channel Architecture)
  - EISA (Extended Industry Standard Architecture)
  - VLB (Vesa Local Bus)
  - PCI (Peripheral Communications Interconnect)
- PCI
  - 32 or 64 bit
  - Synchronous 33MHz or 66MHz clock
  - Multiple masters
  - 111 MB/s peak bandwidth

Example: The Pentium 4's Buses

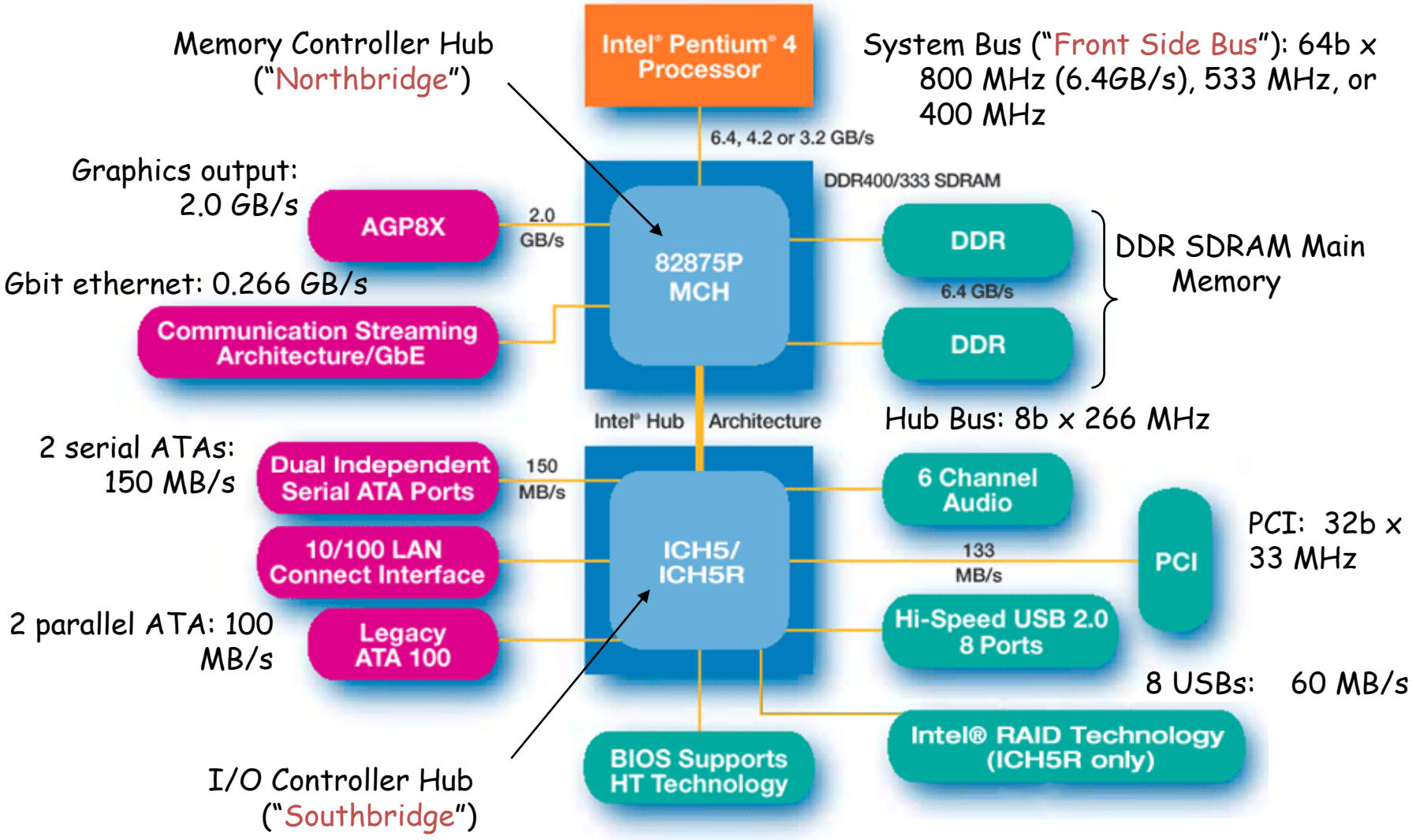
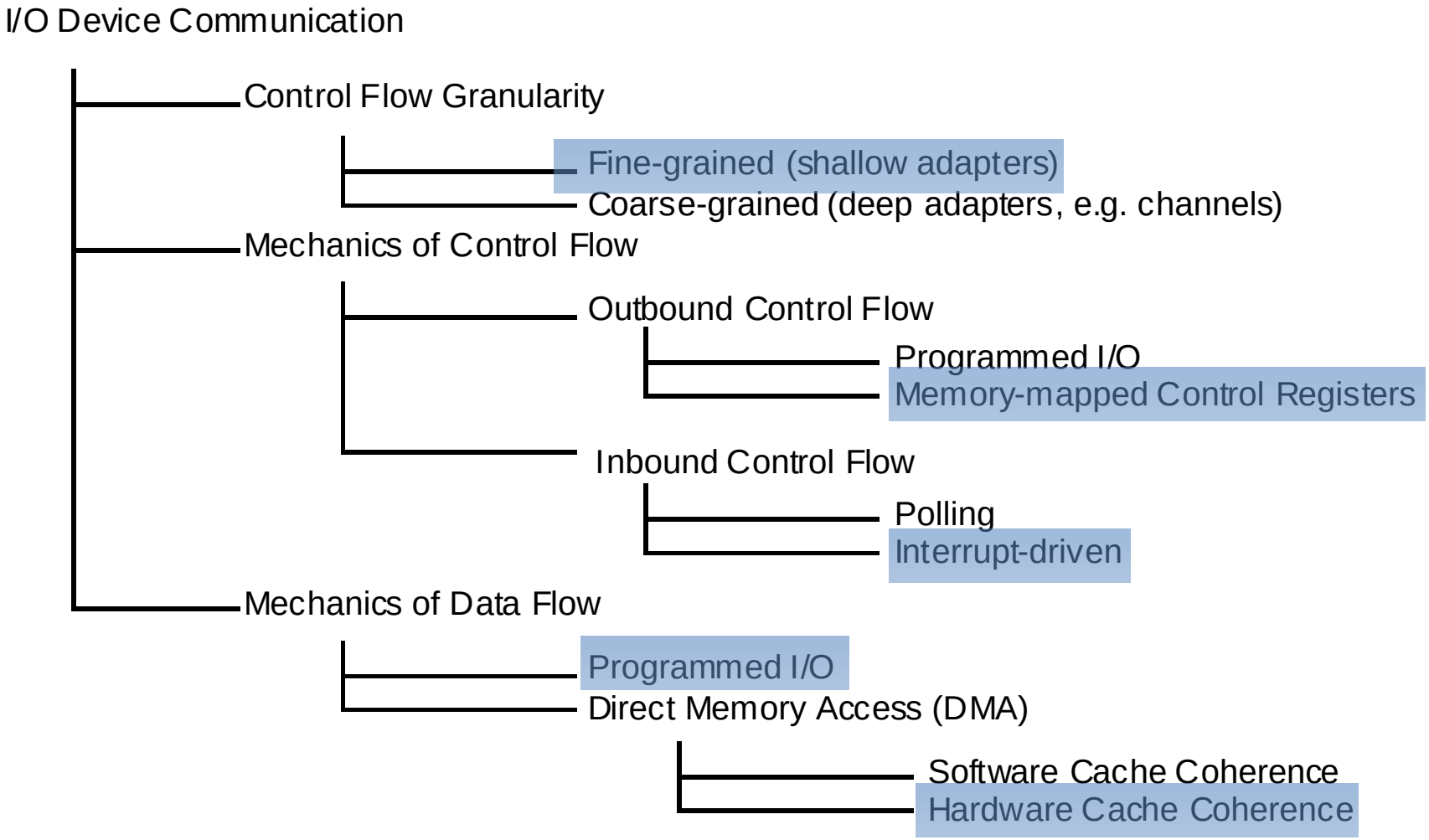


Fig 8.11

Interfacing

- Three key characteristics
  - Multiple users share I/O resource
  - Overhead of managing I/O can be high
  - Low-level details of I/O devices are complex
- Three key functions
  - Virtualize resources – protection, scheduling
  - Use interrupts (similar to exceptions)
  - Device drivers

Interfacing to I/O Devices



Interfacing

- How do you give I/O device a command?
  - Memory-mapped load/store
    - Special addresses not for memory
    - Send commands as data
    - Cacheable?
  - I/O commands
    - Special opcodes
    - Send over I/O bus

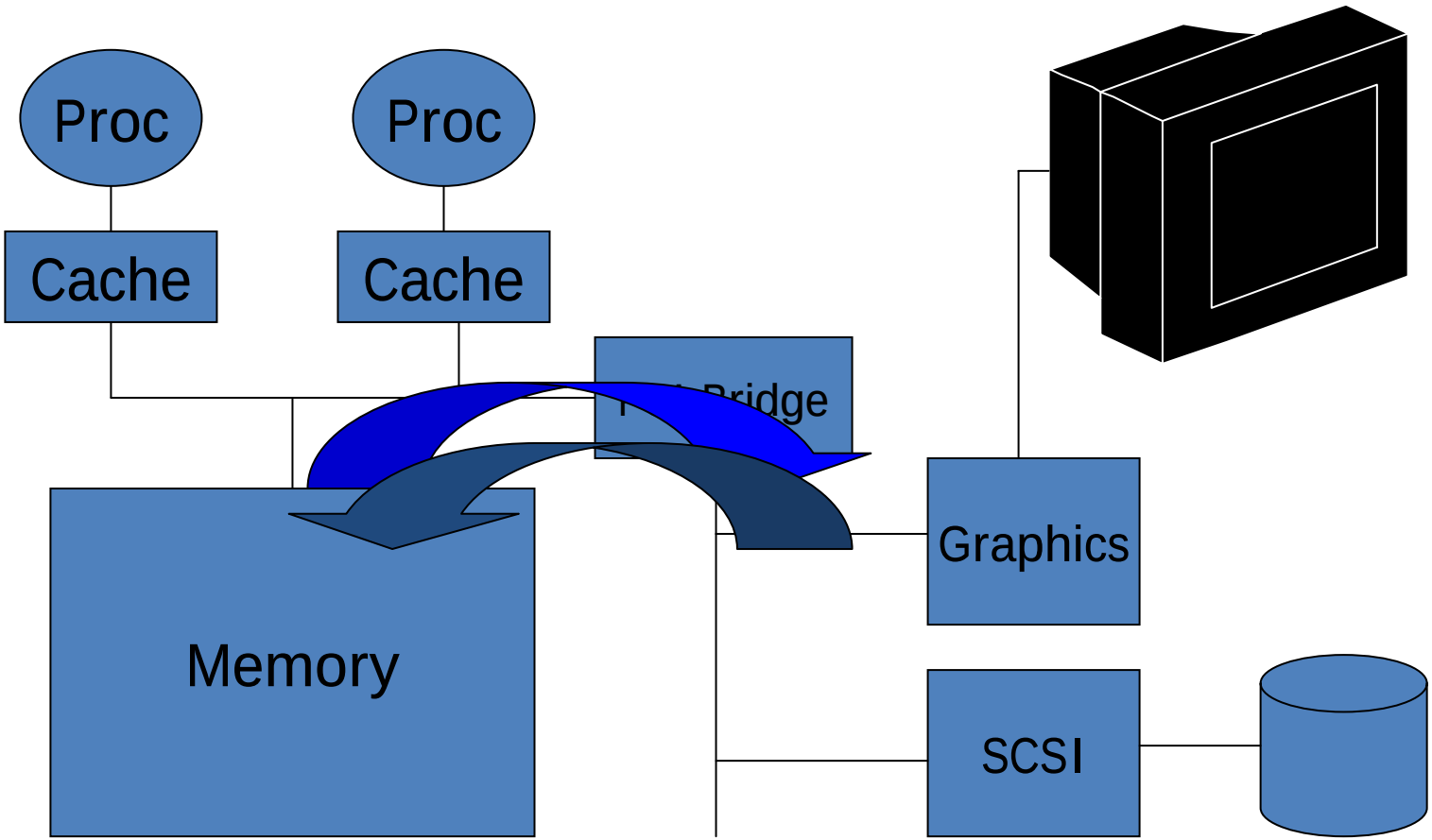
Interfacing

- How do I/O devices communicate w/ CPU?
  - Poll on devices
    - Waste CPU cycles
    - Poll only when device active?
  - Interrupts
    - Similar to exceptions, but asynchronous
    - Info in cause register
    - Possibly vectored interrupt handler

Interfacing

- Transfer data
  - Polling and interrupts – by CPU
  - OS transfers data
- Too many interrupts?
  - Use DMA so interrupt only when done
  - Use I/O channel – extra smart DMA engine
    - Offload I/O functions from CPU

Input/Output



Interfacing

- DMA
  - CPU sets up
    - Device ID, operation, memory address, # of bytes
  - DMA
    - Performs actual transfer (arb, buffers, etc.)
  - Interrupt CPU when done
- Typically I/O bus with devices use DMA
  - E.g. hard drive, NIC

Interfacing

- DMA virtual or physical addresses?
- Cross page boundaries within DMA?
  - Virtual
    - Page table entries, provided by OS
  - Physical
    - One page per transfer
    - OS chains the physical addresses
- No page faults in between – lock pages

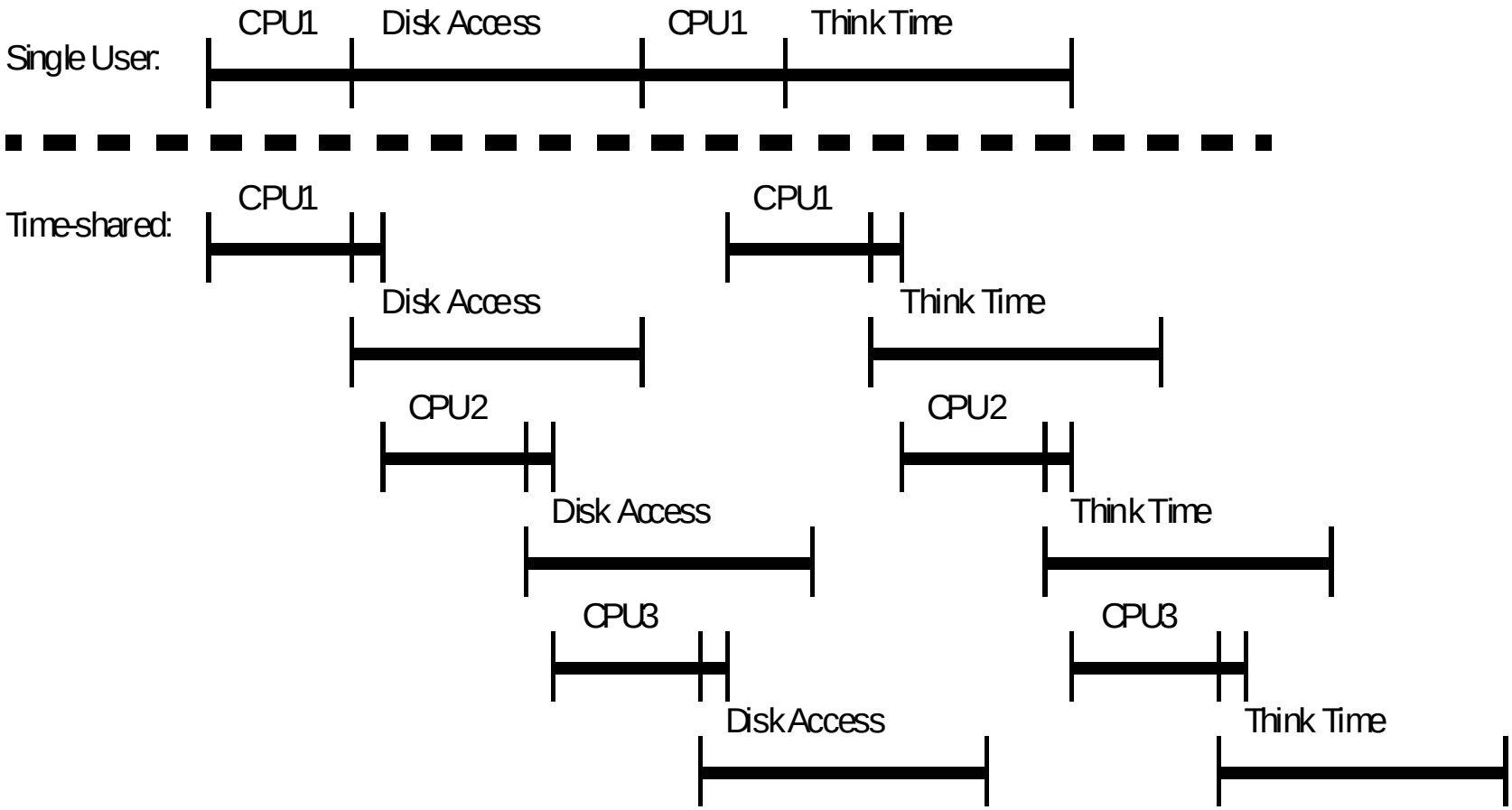
Interfacing

- Caches and I/O
  - I/O in front of cache – slows CPU
  - I/O behind cache – cache coherence?
  - OS must invalidate/flush cache first before I/O

Interfacing

- Multiprogramming
  - I/O through OS
  - Syscall interface between program and OS
  - OS checks protections, runs device drivers
  - Suspends current process, switches to other
  - I/O interrupt fielded by O/S
  - O/S completes I/O and makes process runnable
  - After interrupt, run next ready process

Multiprogramming



Summary – I/O

- I/O devices
  - Human interface – keyboard, mouse, display
  - Nonvolatile storage – hard drive, tape
  - Communication – LAN, modem
- Buses
  - Synchronous, asynchronous
  - Custom vs. standard
- Interfacing
  - O/S: protection, virtualization, multiprogramming
  - Interrupts, DMA, cache coherence