ECE/CS 552: Input/Output

© Prof. Mikko Lipasti

Lecture notes based in part on slides created by Mark Hill, David Wood, Guri Sohi, John Shen and Jim Smith

Input/Output

- Motivation
- I/O Devices
- Buses
- Interfacing
- Examples

Motivation

- 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?

Examples

Device	I or O?	Partner	Data Rate KB/s
Mouse	I	Human	0.01
Display	O	Human	373,000
Modem	I/O	Machine	2-8
LAN	I/O	Machine	100,000
Tape	Storage	Machine	2000
Disk	Storage	Machine	2000-100,000

Humans are asymmetric!

I/O Performance

- What is performance?

- Supercomputers read/write 1GB of data
  - Want high bandwidth to vast data (bytes/sec)

- Transaction processing: 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

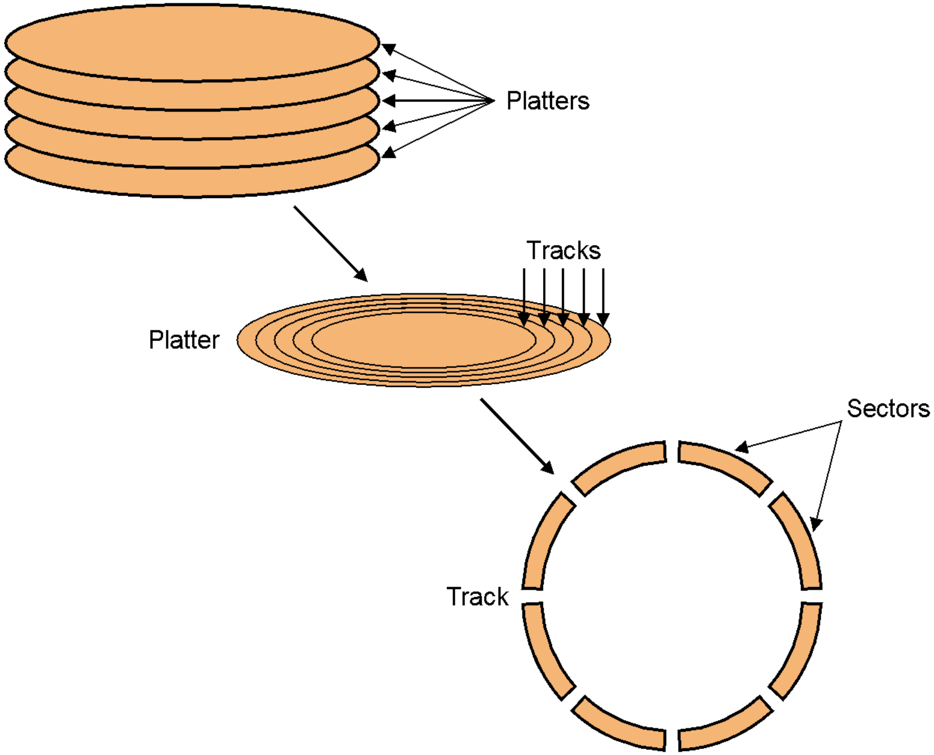
Tracks

Heads move together

Sectors

Disk access

- Queueing + seek
- Rotation + transfer



Magnetic Disks

- Seek = 10-20ms but smaller with locality
- Rotation = ½ rotation/3600rpm = 8.3ms
- Transfer = x / 2-4MB/s
  - E.g. 4kB/4MB/s = 1ms
- Remember: mechanical => ms

Disk Trends

- Disk trends
  - $/MB down (well below $.10/GB)
  - Disk diameter: 14” => 3.5” => 2.5” => 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

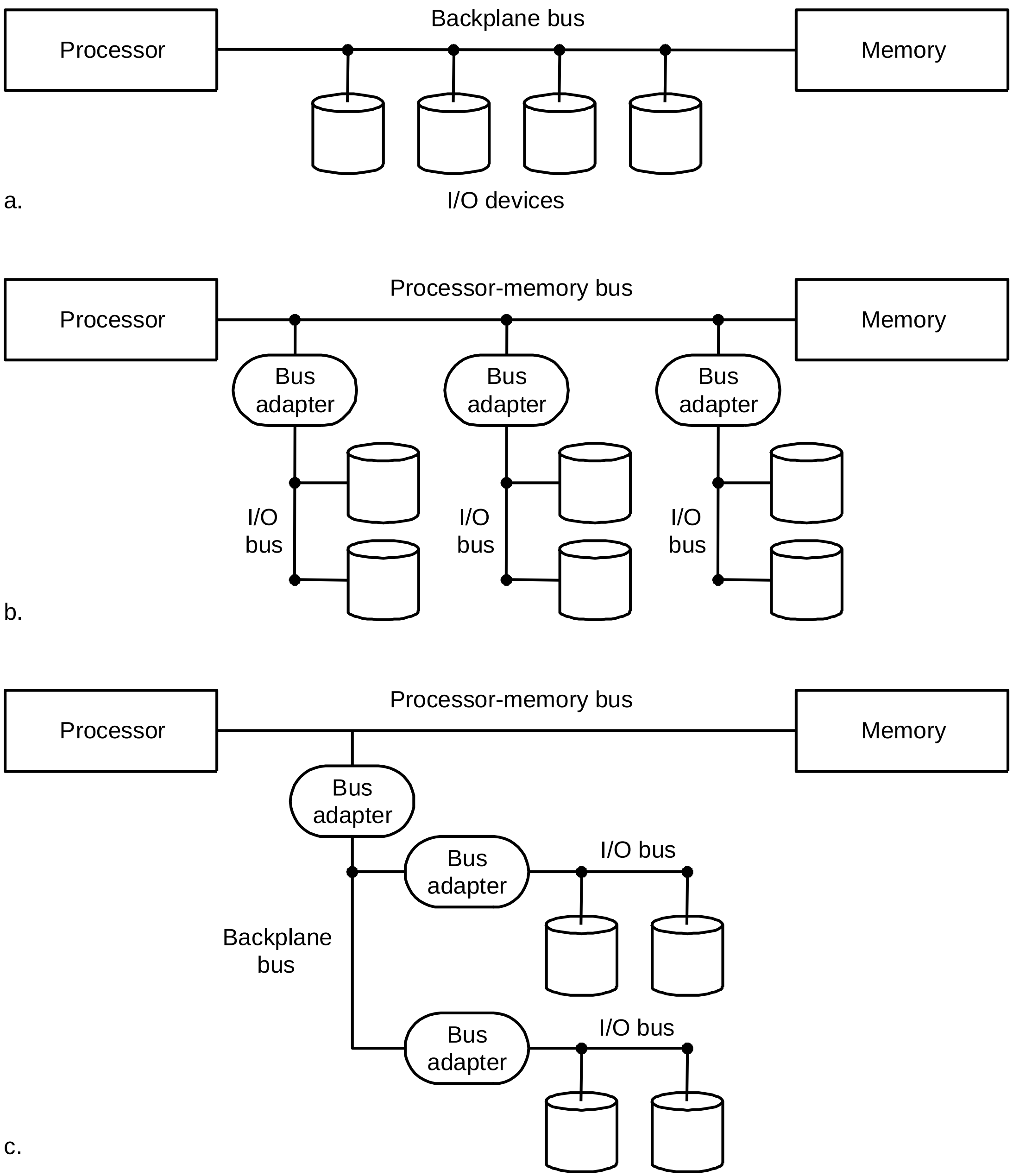
RAID

- What if we need 100 disks for storage?
- MTTF = 5 years / 100 = 18 days!
- RAID 0
  - Data striped, but no error protection
- RAID 1
  - Mirror = stored twice = 100% overhead
- RAID 5
  - Block-wise parity = small overhead and small writes
    - Need (n+1) disks for (n) capacity
  - Know which disk failed => know which bit is wrong

GPU/Video Card

- Extreme bandwidth requirement just for frame buffer
  - 1920x1080 pixels x 24bits/pixel = 6.2MB
  - Refresh whole screen 60 times/sec = 373MB/s !
- 3D rendering amplifies bandwidth demand
  - Texture memory access, etc.
- GPUs use specialized, dedicated memory (GDDRx)
  - APUs share DDRx memory, can't keep up
- Connected via PCIe x16 to system memory

Buses in a Computer System



Buses

- Bunch of wires
  - Arbitration
  - Control
  - Data
  - Address
  - Flexible, low cost
  - Can be bandwidth bottleneck

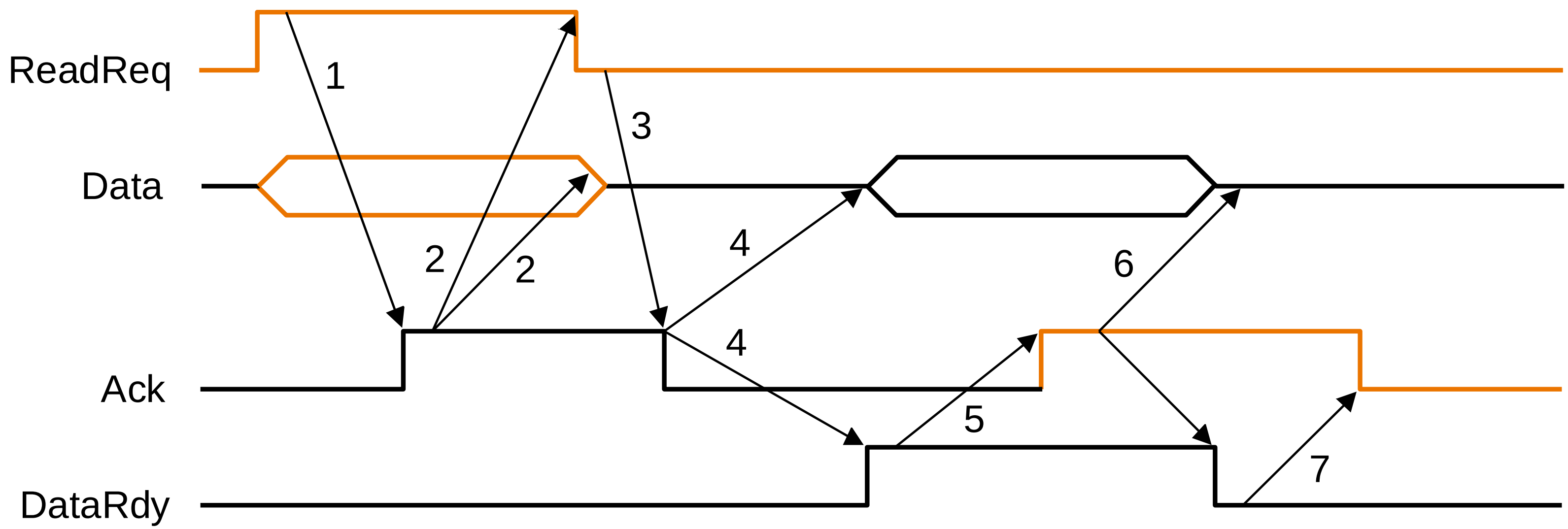
Buses

- Types
  - Processor-memory
    - Short, fast, custom
  - I/O
    - Long, slow, standard
  - Backplane
    - Medium, medium, standard

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

Async. Handshake Example



(1) Request made & (2) request send

(3) Request deasserted & (4) ack deasserted

(5) Data sent & (6) Data rec'd & (7) ack deasserted

Buses

- Synchronous vs. asynchronous
  - Must distribute clock and deal with skew
  - Simple handshake
  - Backward compatibility difficult, esp. with slow devices
  - No metastability problems (FSD)

Buses

- Improving bandwidth
  - Wider bus
  - Block transfer to exploit spatial locality
  - Separate address/data lines
  - Split transactions (multiple concurrent requests)
  - Pipelined in-order responses
  - Out-of-order responses

Bus Arbitration

- One or more bus masters, others slaves
  - Bus request
  - Bus grant
  - Priority
  - Fairness

- Implementations
  - Centralized vs. distributed

Buses

- Bus standards: ISA, PCI, PCI-X, AGP, …
- Currently PCIe 2.x
  - Serial, point-to-point topology
  - Bidirectional differential lanes (4 wires each)
  - 5GHz signaling rate per lane
  - 8b/10b encoding for DC balance, clock recovery
  - 5Gbit/sec x 10bit/byte = 500 MB/s per lane per direction
  - x1-x16 lanes per slot
- PCIe 3.0: 8GHz, 128/130b encoding

Interfacing

- Three key characteristics
  - Multiple users/programs 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

- 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

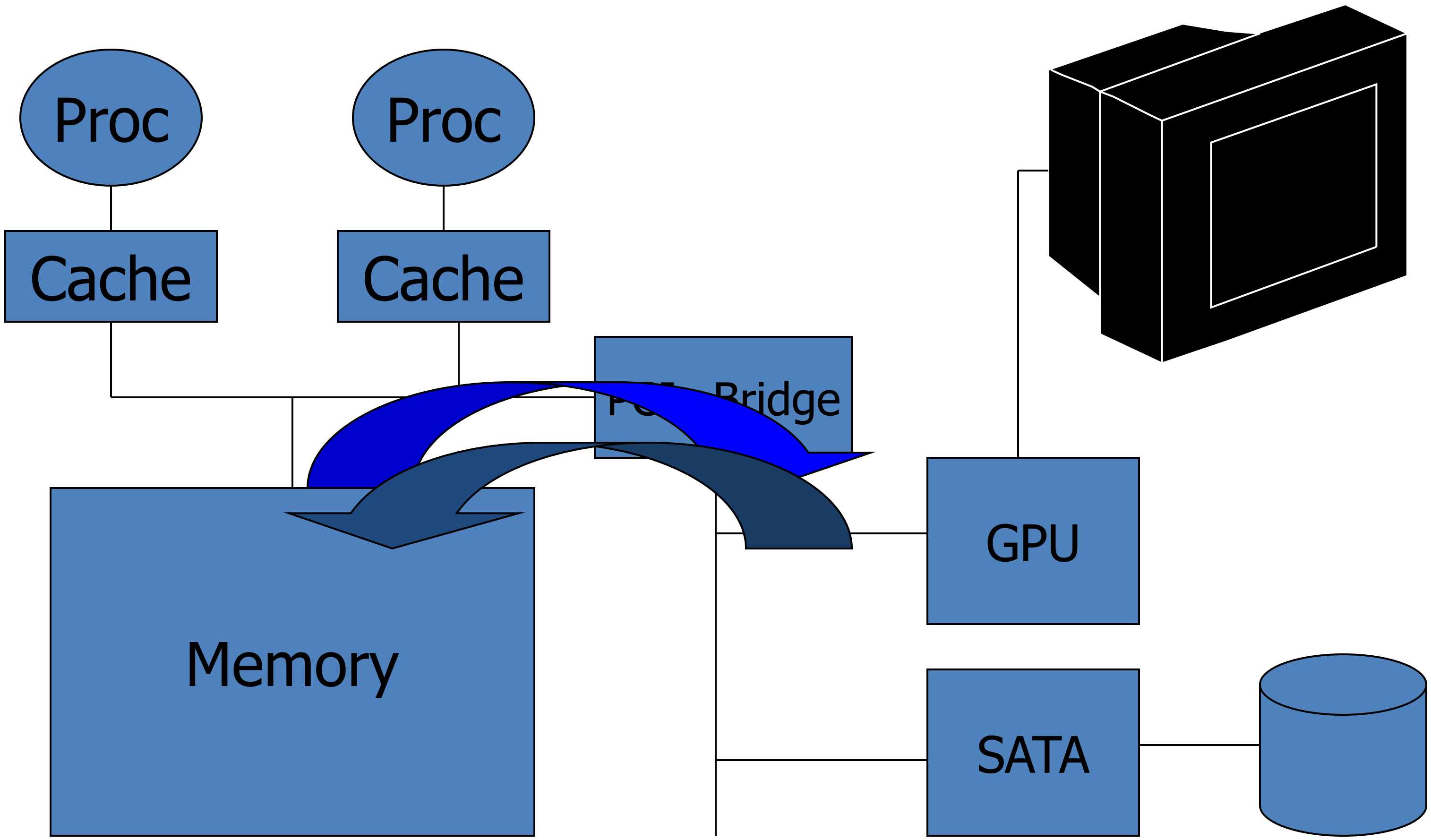
Interfacing

- 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

Direct Memory Access (DMA)



DMA (cont’d)

- DMA
  - CPU sets up
    - Device ID, operation, memory address, # of bytes
  - DMA
    - Performs actual transfer (arb, buffers, etc.)
  - Interrupt CPU when done

- Typical I/O devices that use DMA
  - Hard drive, SSD, NIC, GPU

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 Summary

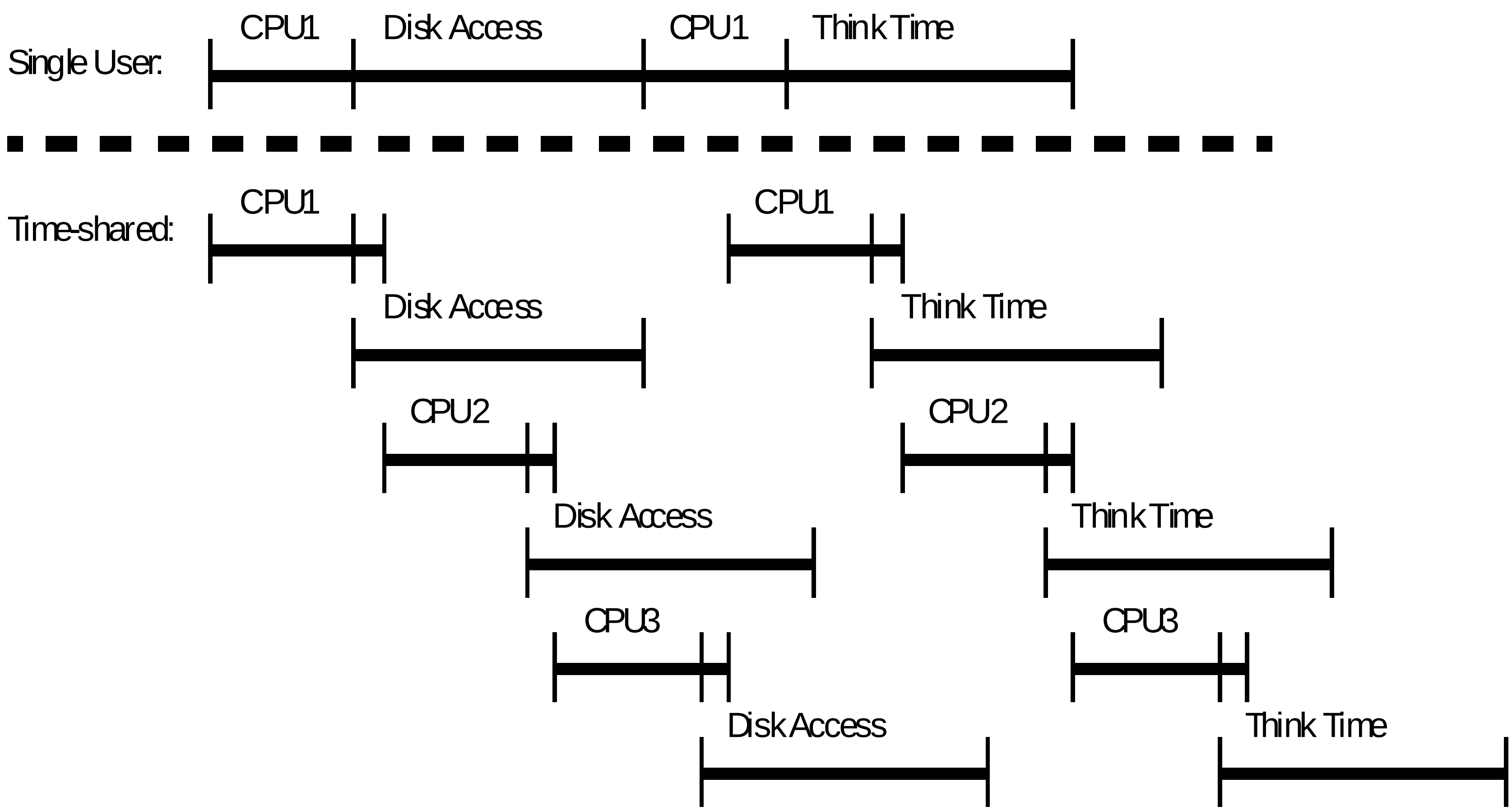
I/O Device Communication

- Control Flow Granularity
  - Fine-grained (shallow adapters)
  - Coarse-grained (deep adapters, e.g. channels)
- Mechanics of Control Flow
  - Outbound Control Flow
    - Programmed I/O
    - Memory-mapped Control Registers
  - Inbound Control Flow
    - Polling
    - Interrupt-driven
- Mechanics of Data Flow
  - Programmed I/O
  - Direct Memory Access (DMA)
    - Software Cache Coherence
    - Hardware Cache Coherence

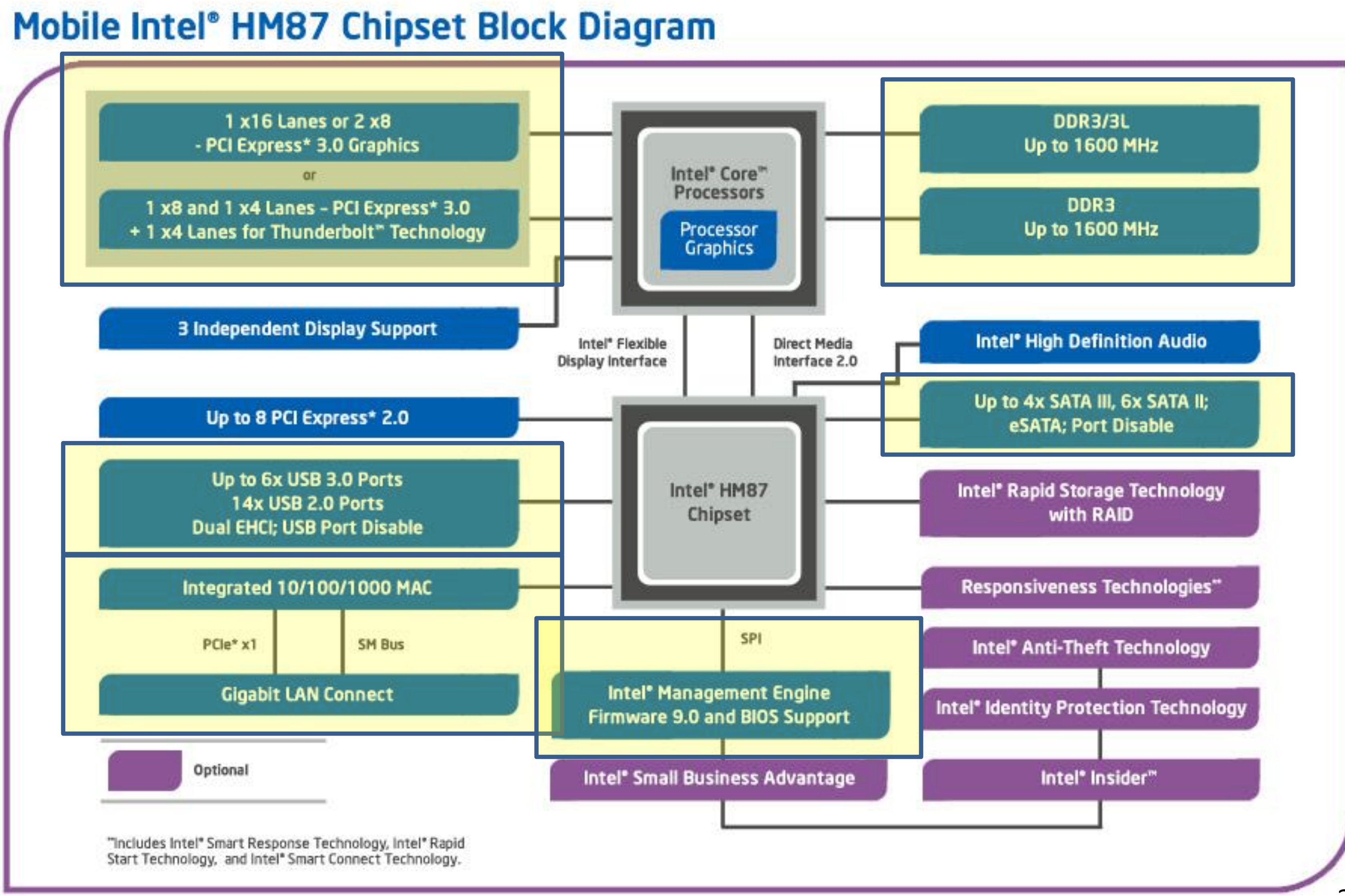
Software Interfacing

- I/O access provided by 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

Multiprogramming



I/O System Example



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
  - Interrupts, DMA, cache coherence
  - O/S: protection, virtualization, multiprogramming