EITF20: Computer Architecture

Part 5.2.1: Storage System and IO

Liang Liu
liang.liu@eit.lth.se

Outline

- Reiteration
- I/O
- Storage Systems
- DMA
- RAID
- Summary

Virtual memory benifits

Using physical memory efficiently

- Allowing software to address more than physical memory
- Enables programs to begin before loading fully (some implementations)
- Programmers used to use overlays and manually control loading/unloading (if the program size is larger than mem size)

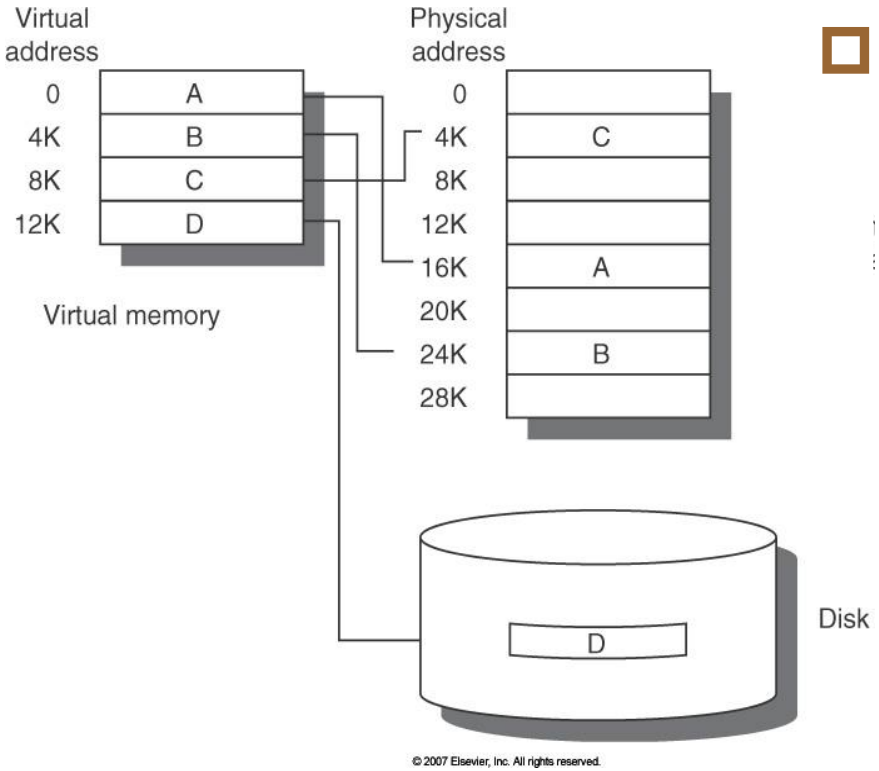
Using physical memory simply

- Virtual memory simplifies memory management
- Programmer can think in terms of a large, linear address space

Using physical memory safely

- Virtual memory protests process' address spaces
- Processes cannot interfere with each other, because they operate in different address space (or limited mem space)
- User processes cannot access priviledged information

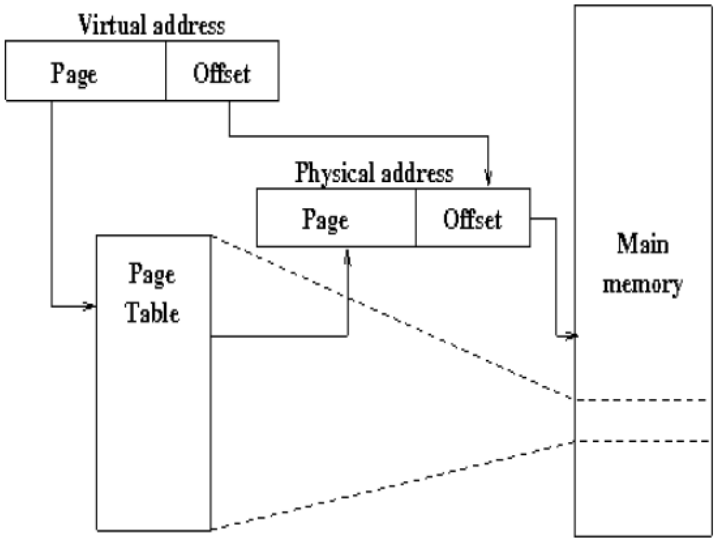
Virtual memory concept



- **Is part of memory hierarchy**
  - The virtual address space is divided into pages (blocks in Cache)
  - The physical address space is divided into page frames
  - A miss is called a page fault
  - Pages not in main memory are stored on disk

- **The CPU uses *virtual addresses***
- **We need an *address translation* (memory mapping) mechanism**

Page identification: address mapping



- **4Byte per page table entry**
- **Page table will have**

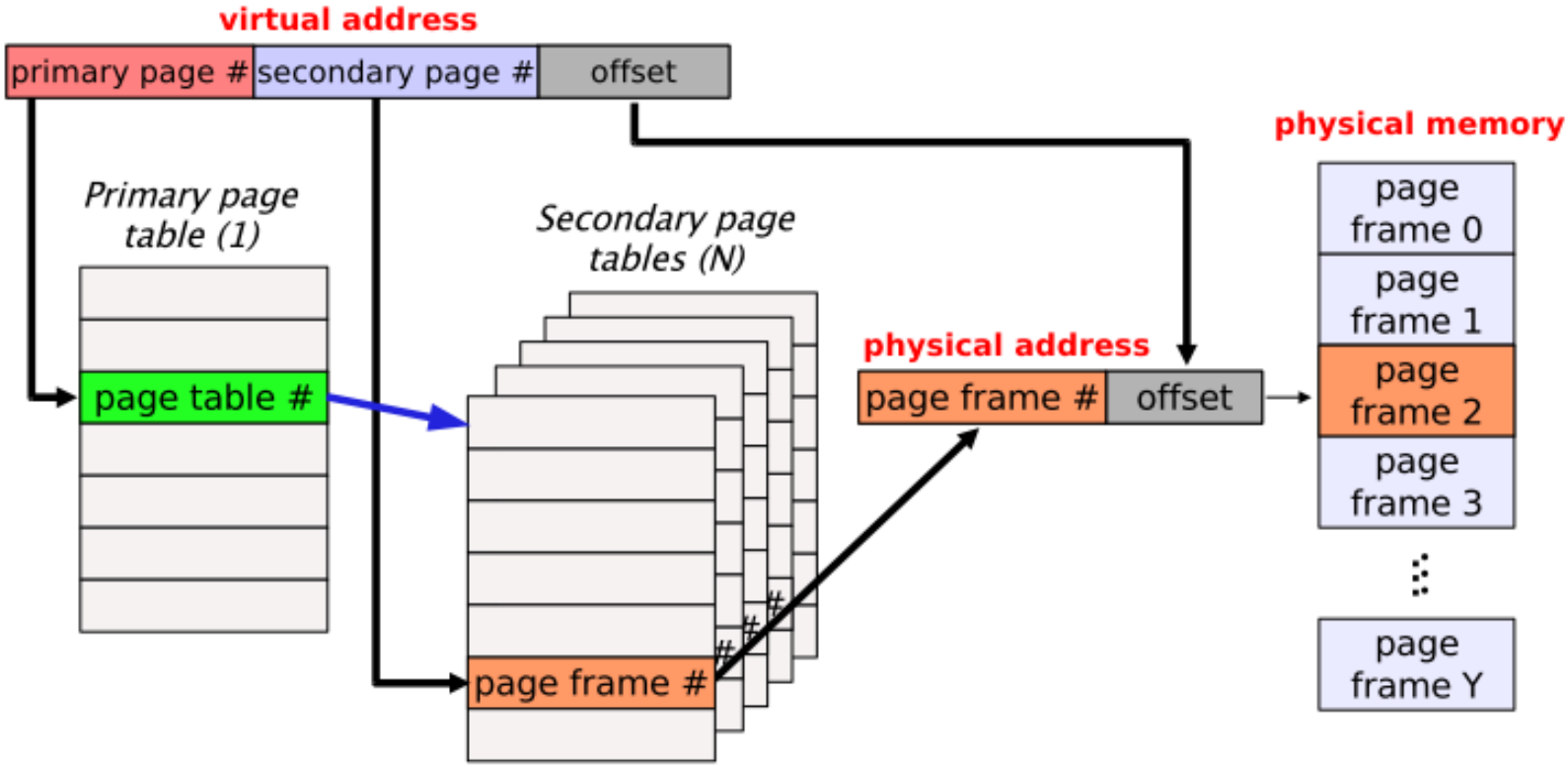
$2^{20}*4=2^{22}=4$MByte

- **64 bit virtual address,16 KB pages →**

$2^{64}/2^{14}*4=2^{52}=2^{12}$TByte

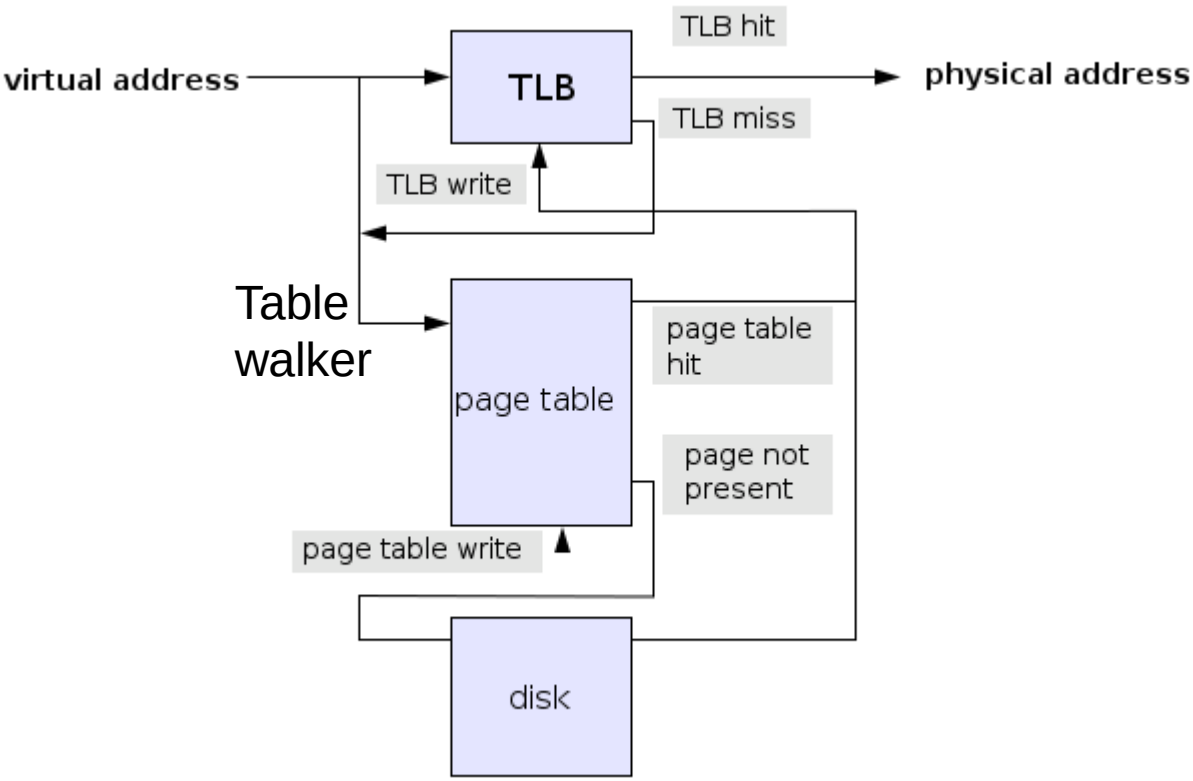
- **Solutions**
  - Multi–level page table
  - Inverted page table

Multi-level PT



Page identification

- **How do we avoid two (or more) memory references for each original memory reference?**
  - Cache address translations – Translation Look-aside Buffer (TLB)



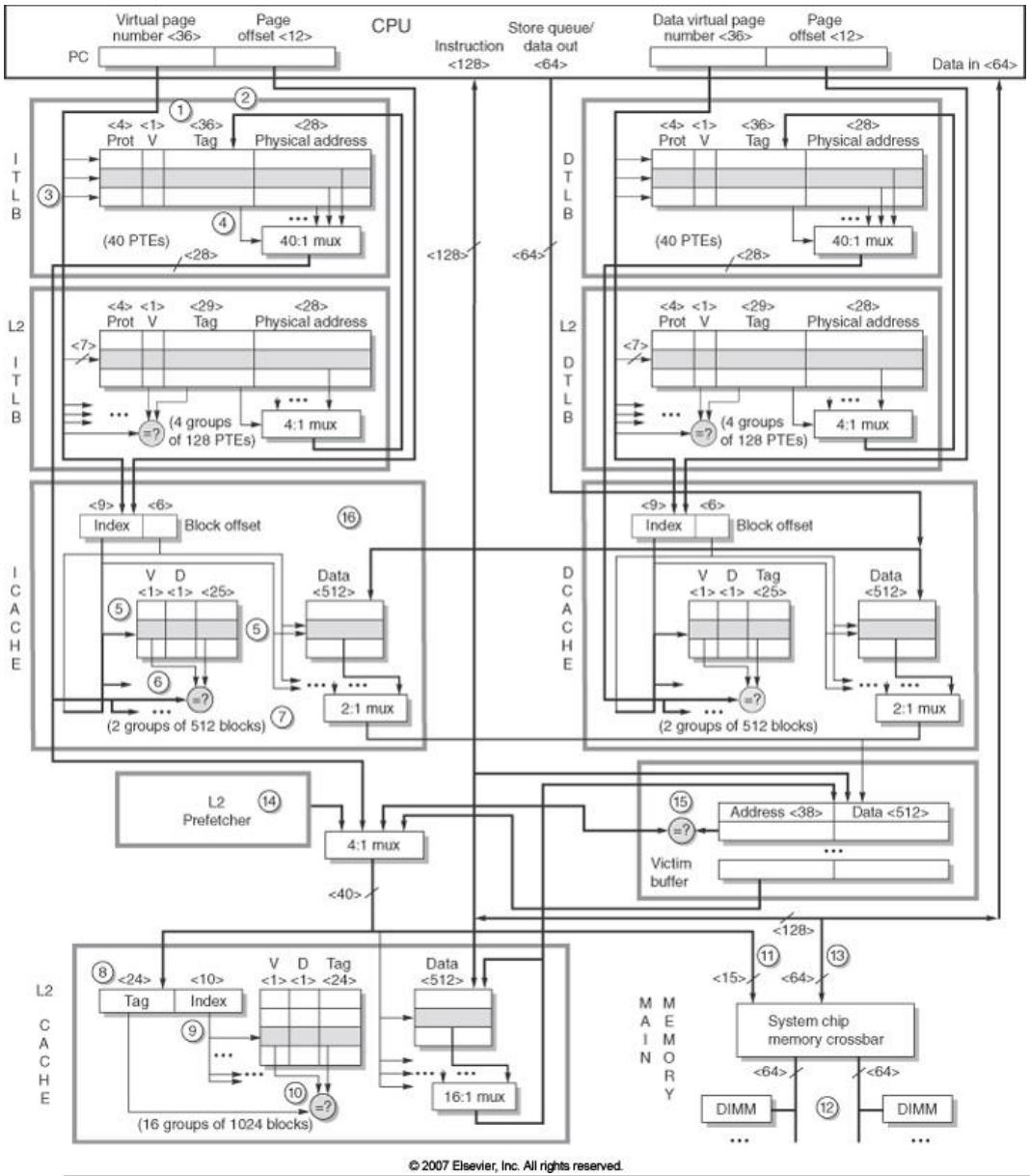
Summary memory hierarchy

Hide CPU - memory performance gap
Memory hierarchy with several levels
Principle of locality

Cache memories:	**Virtual memory:**
• Fast, small - Close to CPU	• Slow, big - Close to disk
• Hardware	• Software
• TLB	• TLB
• CPU performance equation	• Page-table
• Average memory access time	• Very high miss penalty $\Longrightarrow$ miss rate must be low
• Optimizations	• Also facilitates: relocation; memory protection; and multiprogramming

Same 4 design questions - Different answers

The memory hierarchy of AMD Opteron



- **Separate Instr & Data TLB and Caches**
- **2-level TLBs**
  - L1 TLBs fully associative
  - L2 TLBs 4 way set associative
- **Write buffer (and Victim cache)**
- **Way prediction**
- **Line prediction: prefetch**
- **hit under 10 misses**
- **1 MB L2 cache, shared, 16 way set associative, write back**

Take a step back

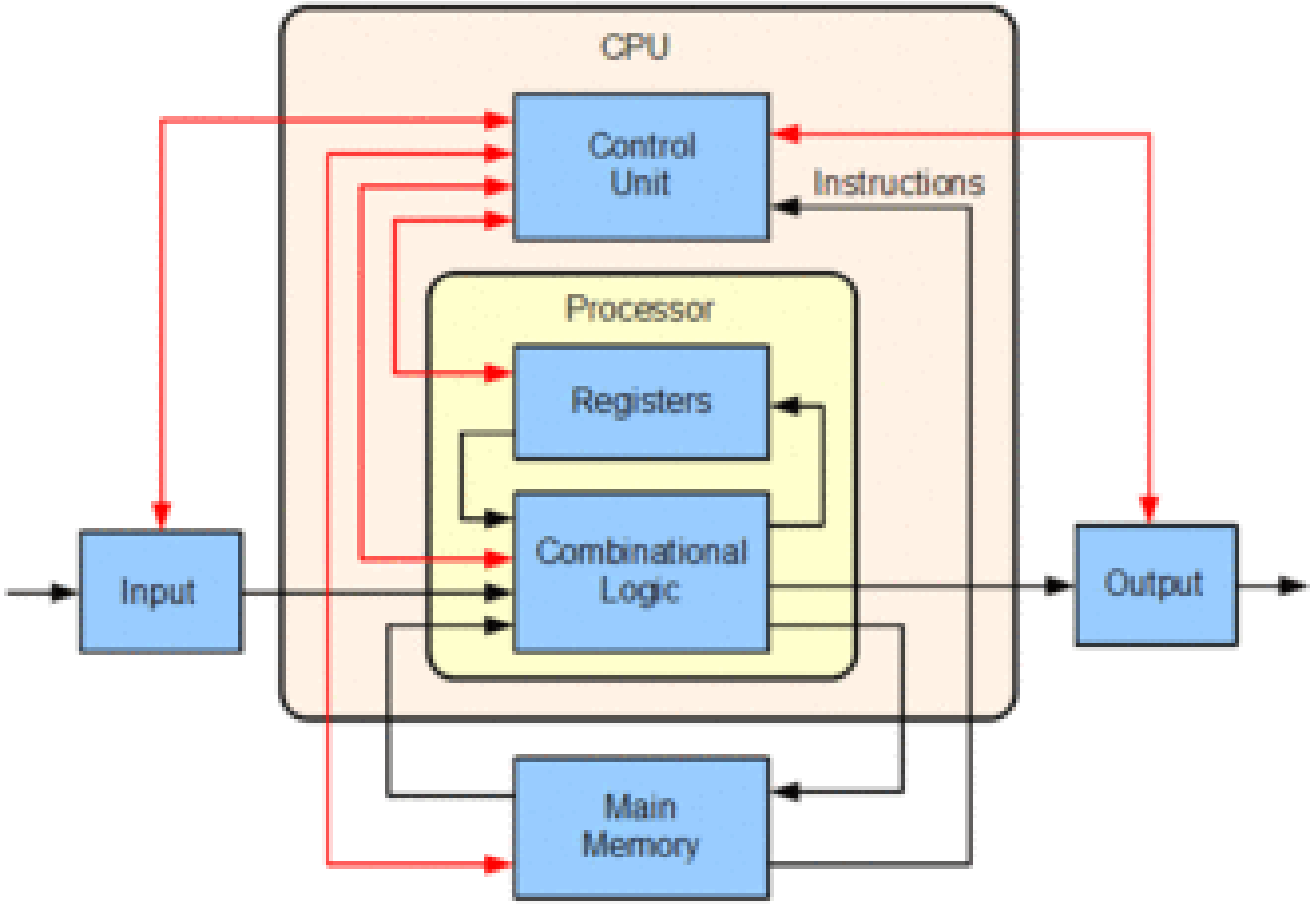
So far

- Performance, Quantitative principles
- Instruction set architectures, ISA
- Pipelining, ILP
- Memory systems, cache, virtual memory

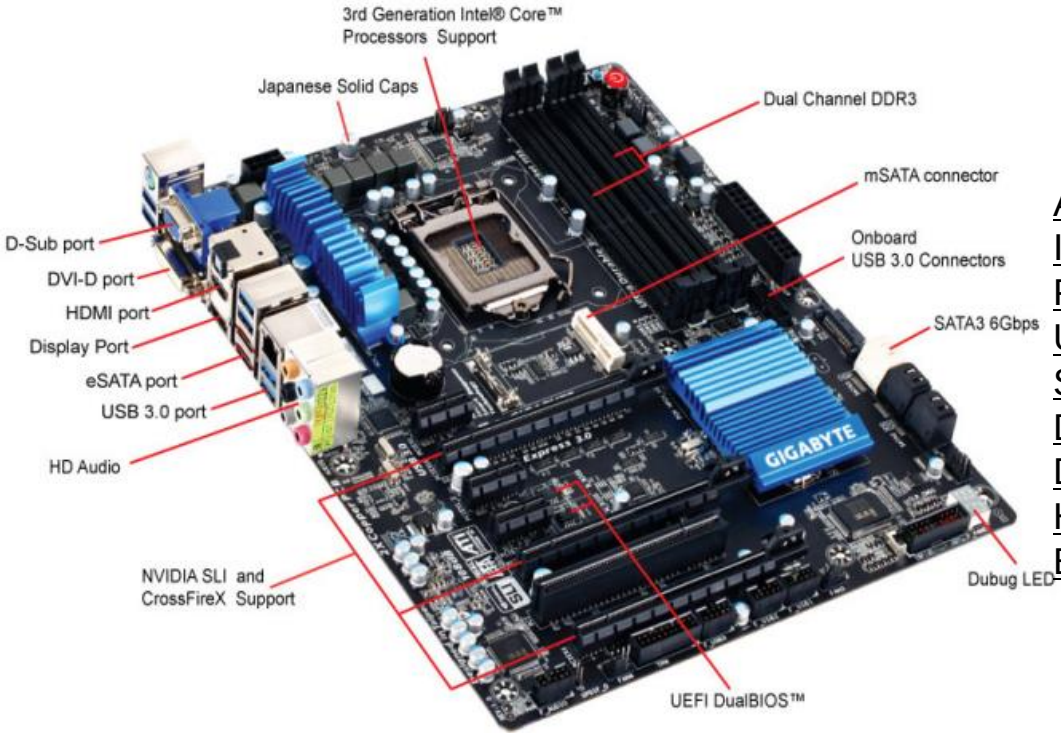
Coming

- Storage Systems, I/O
- Course summary

Computer function and component

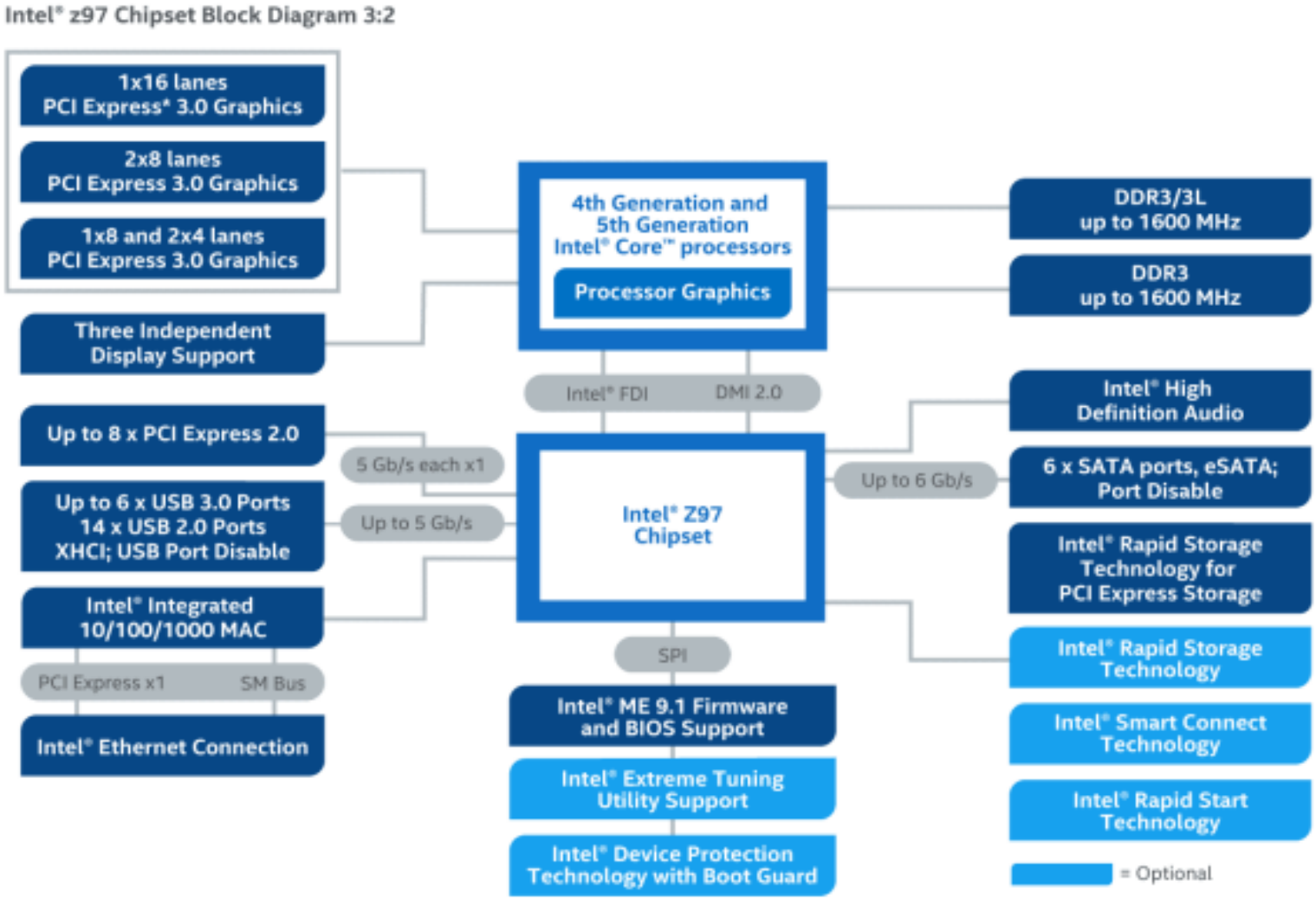


Motherboard



AGP
IDE
PCI
USB
SATA
DDR
DVI
HDMI
BIOS

Chip-set architecture



Outline

- Reiteration
- I/O
- Storage
- DMA
- RAID
- Summary

Outside the processor and memory

Storage Systems

I/O via BUS

I/O

Computers useless without I/O

Over time, literally thousands of forms of computer I/O: punch cards to brain interfaces

- **Broad categories:**
  - Secondary/Tertiary storage (flash/disk/tape)
  - Network (Ethernet, WiFi, Bluetooth, LTE)
  - Human-machine interfaces (keyboard, mouse, touchscreen, graphics, audio, video, neural,…)
  - Printers (line, laser, inkjet, photo, 3D, …)
  - Sensors (process control, GPS, heartrate, …)
  - Actuators (valves, robots, car brakes, …)

Mix of I/O devices is highly application-dependent

Who cares about I/O?

- **CPU performance increases dramatically**
- **I/O system performance limited by mechanical delays ⇒ *less than 10% performance improvement per year***
- **Amdahl's law: system speedup limited by the slowest component:**
  - Assume 10% I/O
  - CPU speedup = 10 ⇒ System speedup = 5
  - CPU speedup = 100 ⇒ System speedup = 10
- **I/O will more and more become a bottleneck!**

$$\text{Speedup}_{\text{overall}} = \frac{1}{\left(1 - \text{Fraction}_{\text{enhanced}}\right) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Synchronous/Asynchronous I/O

Synchronous I/O

- Request data
- Wait for data
- Use data

Asynchronous I/O

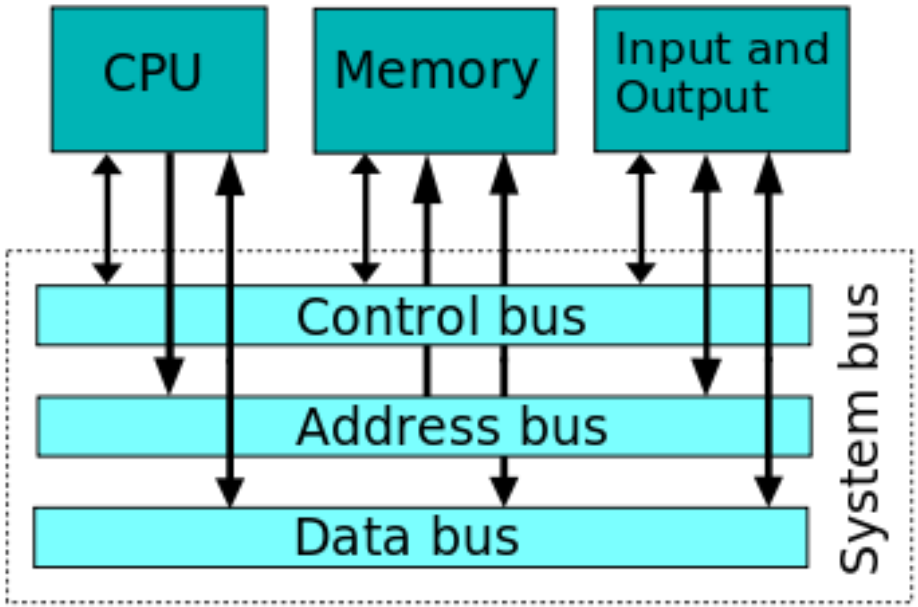
- Request data
- Continue with other things
- Block when trying to use data
- Compare non-blocking caches in out-of-order CPUs
- Multiple outstanding I/O requests

I/O technologies

- **The techniques for I/O have evolved (and sometimes unevolved):**
  - ***Direct control:*** CPU controls device by reading/writing data lines directly
  - ***Polled I/O:*** CPU communicates with hardware via built-in controller; busy-waits (sampling) for completion of commands
  - ***Driven I/O:*** CPU issues command to device, gets interrupt on completion
  - ***Direct memory access:*** CPU commands device, which transfers data directly to/from main memory (DMA controller may be separate module, or on device).
  - ***I/O channels:*** device has specialized processor, interpreting main CPU only when it is truly necessary. CPU asks device to execute entire I/O program

Bus-based interconnect

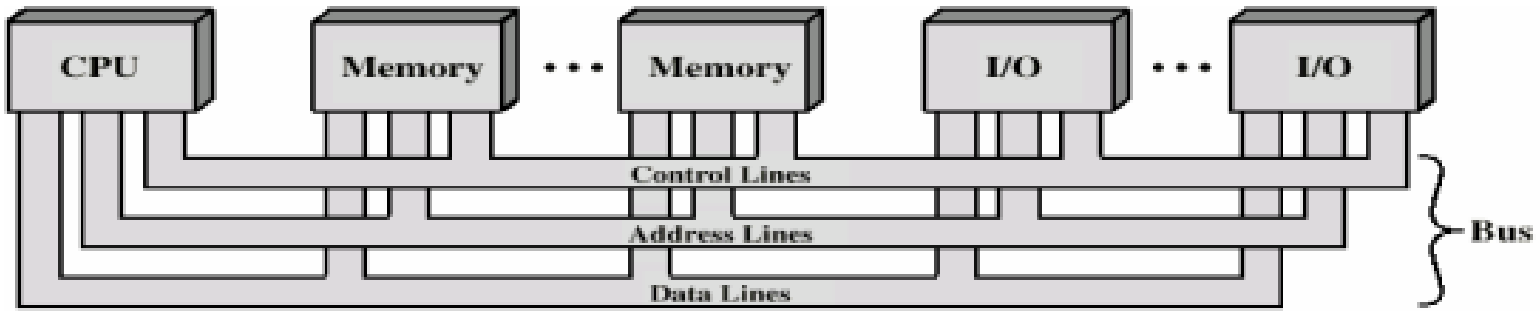
- **Buses are the number one technology to connect the CPU with memory and I/O subsystems**
  - ***Advantages:*** Low cost, shared medium to connect a variety of devices; flexible, expandable
  - ***Disadvantages*:** Inherent problem – limited bandwidth; Bandwidth is limited by bus length and number of devices



Single bus vs multiple bus

Single Bus

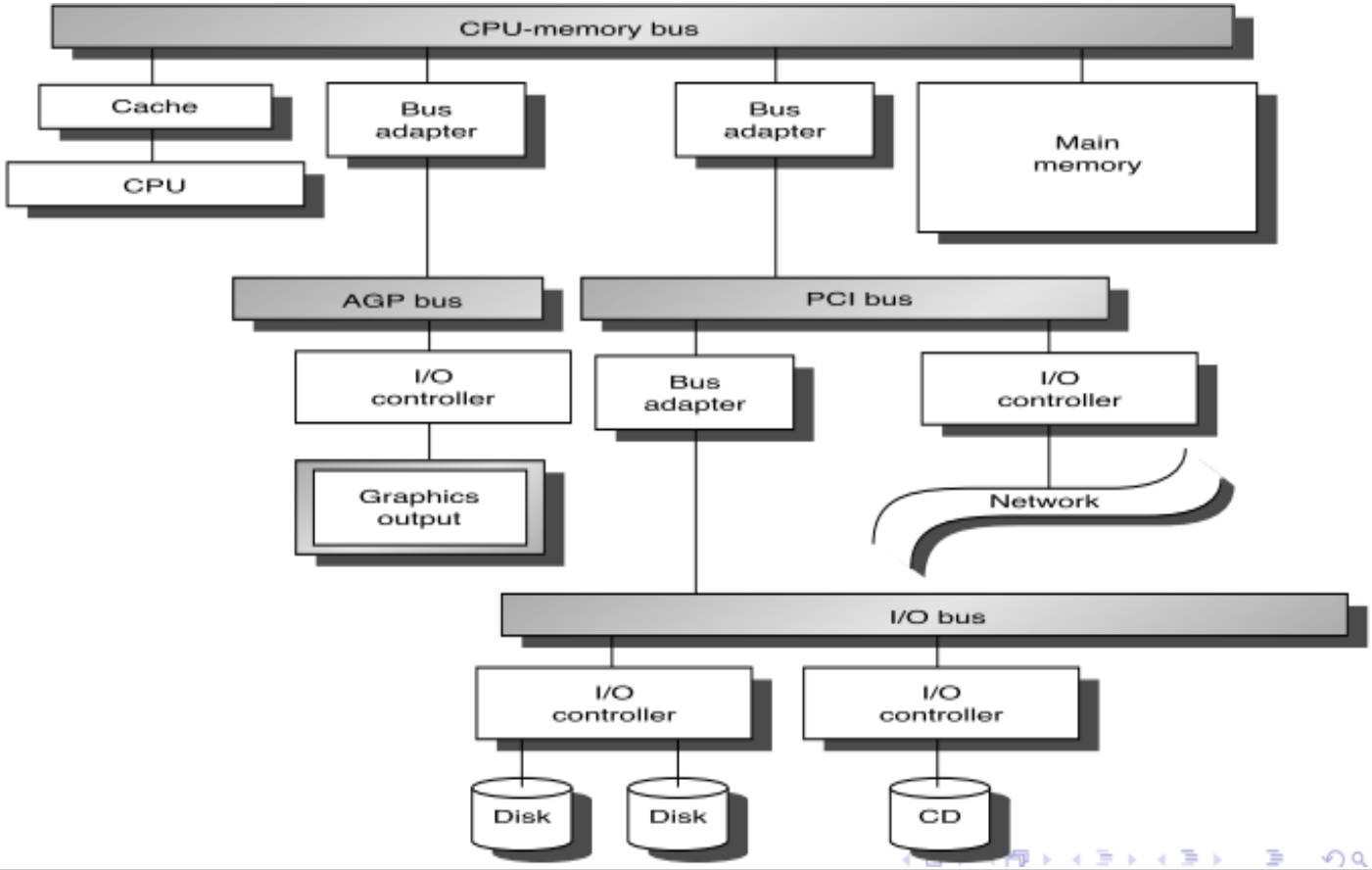
- **Lots of devices on one bus leads to:**
  - Propagation delays; clock skew (100MHz)
  - Long data paths mean that co-ordination of bus use can adversely affect performance
  - Bus may become bottleneck if aggregate data transfer approaches bus capacity
- **Most systems use multiple buses to overcome these problems**



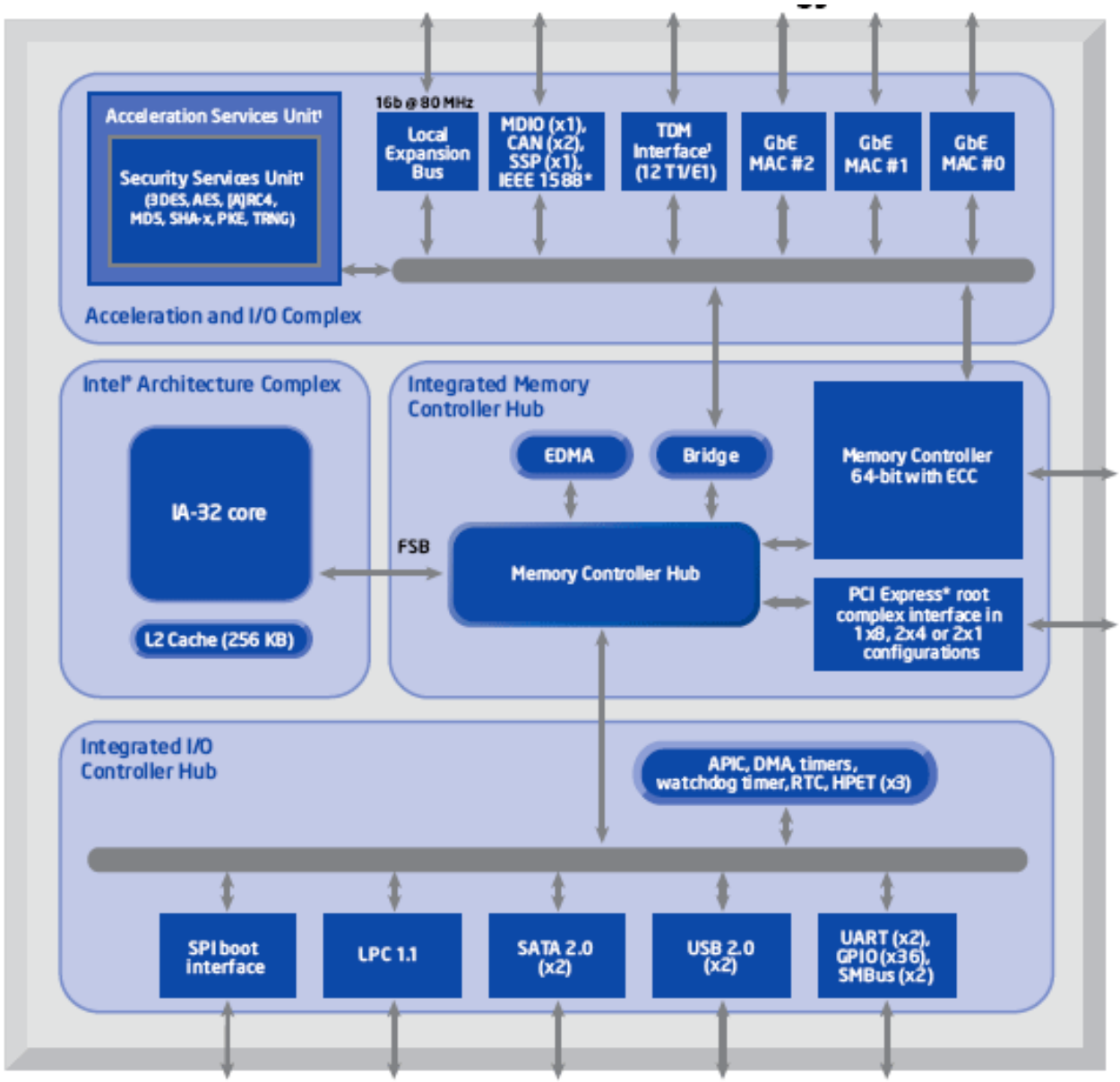
Single bus vs multiple bus

Multiple Bus

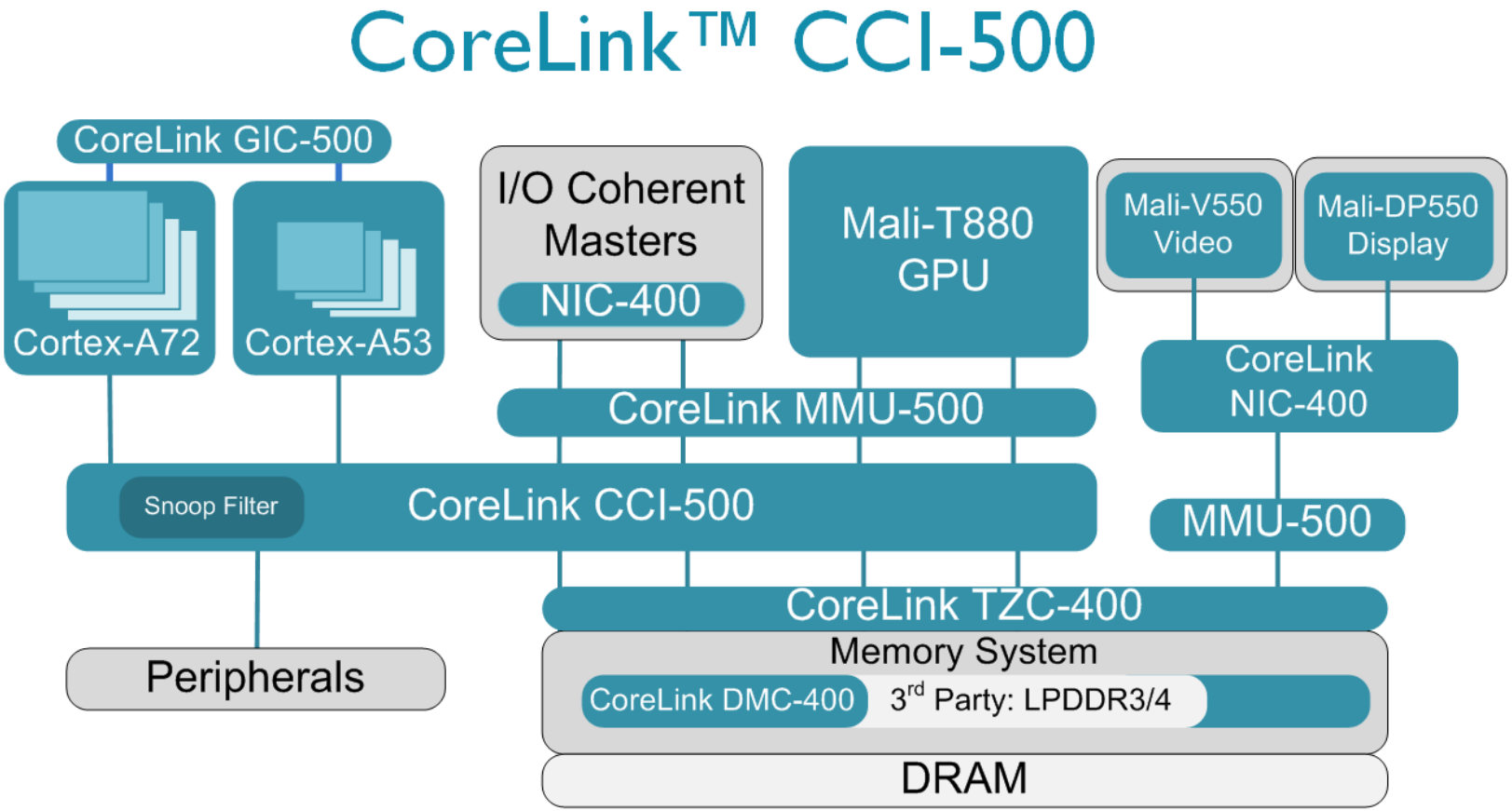
- Allows system to support wide variety of I/O devices
- Insulates memory-to-process traffic from I/O traffic



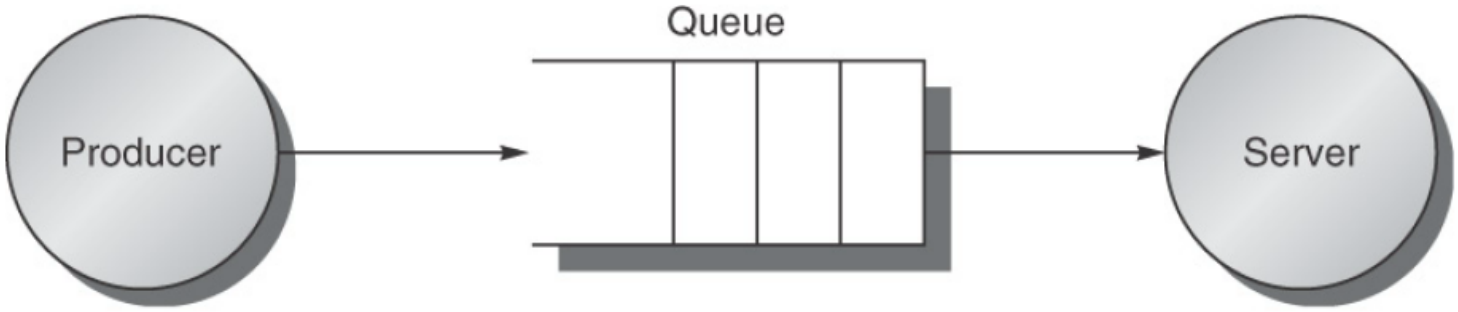
Example: Intel



Example: ARM


CoreLink™ CCI-500
CoreLink GIC-500
Cortex-A72
Cortex-A53
I/O Coherent Masters
NIC-400
Mali-T880 GPU
Mali-V550 Video
Mali-DP550 Display
CoreLink MMU-500
CoreLink NIC-400
Snoop Filter
CoreLink CCI-500
MMU-500
CoreLink TZC-400
Peripherals
Memory System
CoreLink DMC-400
3rd Party: LPDDR3/4
DRAM

Producer-server model



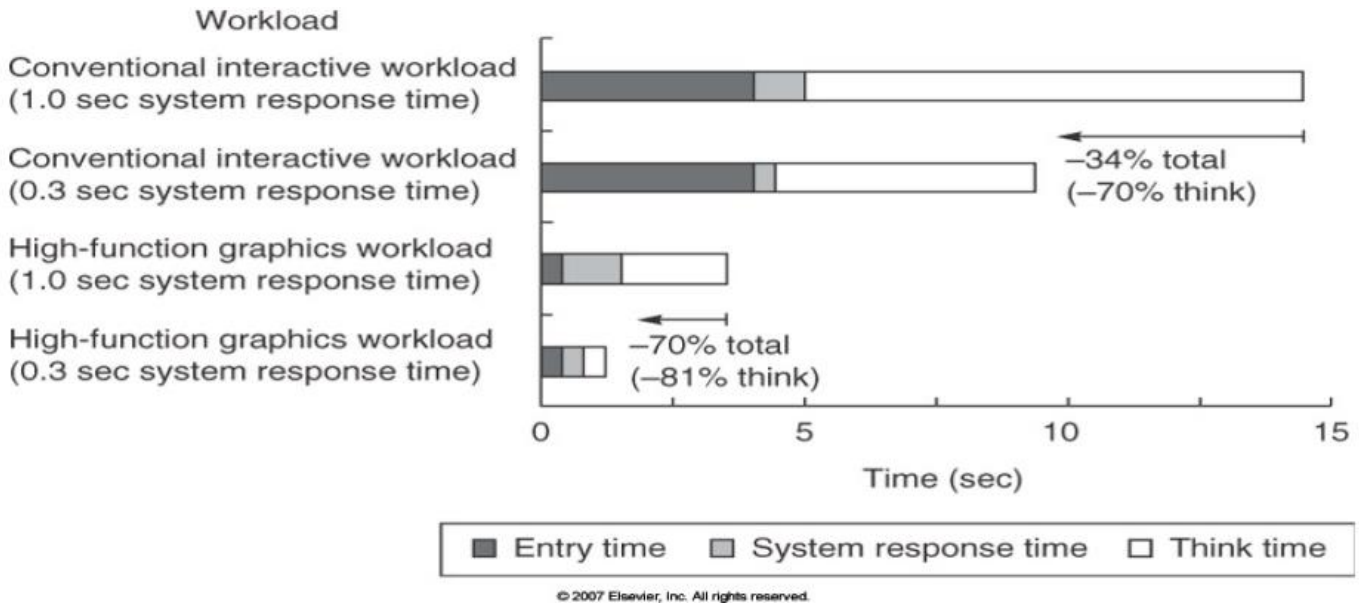
Response time: Time from placed in queue until server is finished

Throughput: Average no of tasks completed per time unit

System response time vs Think time

Interactive environments:

- **Each interaction or transaction has 3 parts:**
  - Entry Time: time for user to enter command
  - System Response Time: time between user entry & system replies
  - Think Time: Time from response until user begins next command



Buses

Standard	Width (bits)	Clock rate	MB/sec
(Parallel) ATA	8/16	133 MHz	133/266
Seri			0
Serial ATA	2	6 GHz	600
USB 2.0	1		35
USB 3.0			400
(USB 3.1)	1	7-10 Gbit/sec	?
SCSI	16	80 MHz	320
Serial Attach SCSI	2	(DDR)	375
PC			
P			
Ethernet	1	1 Gbit/s	<100

SATA revision 3.2 (16 Gbit/s, 1969 MB/s)

USB 3.1 Gen2 (10Gbit/s)

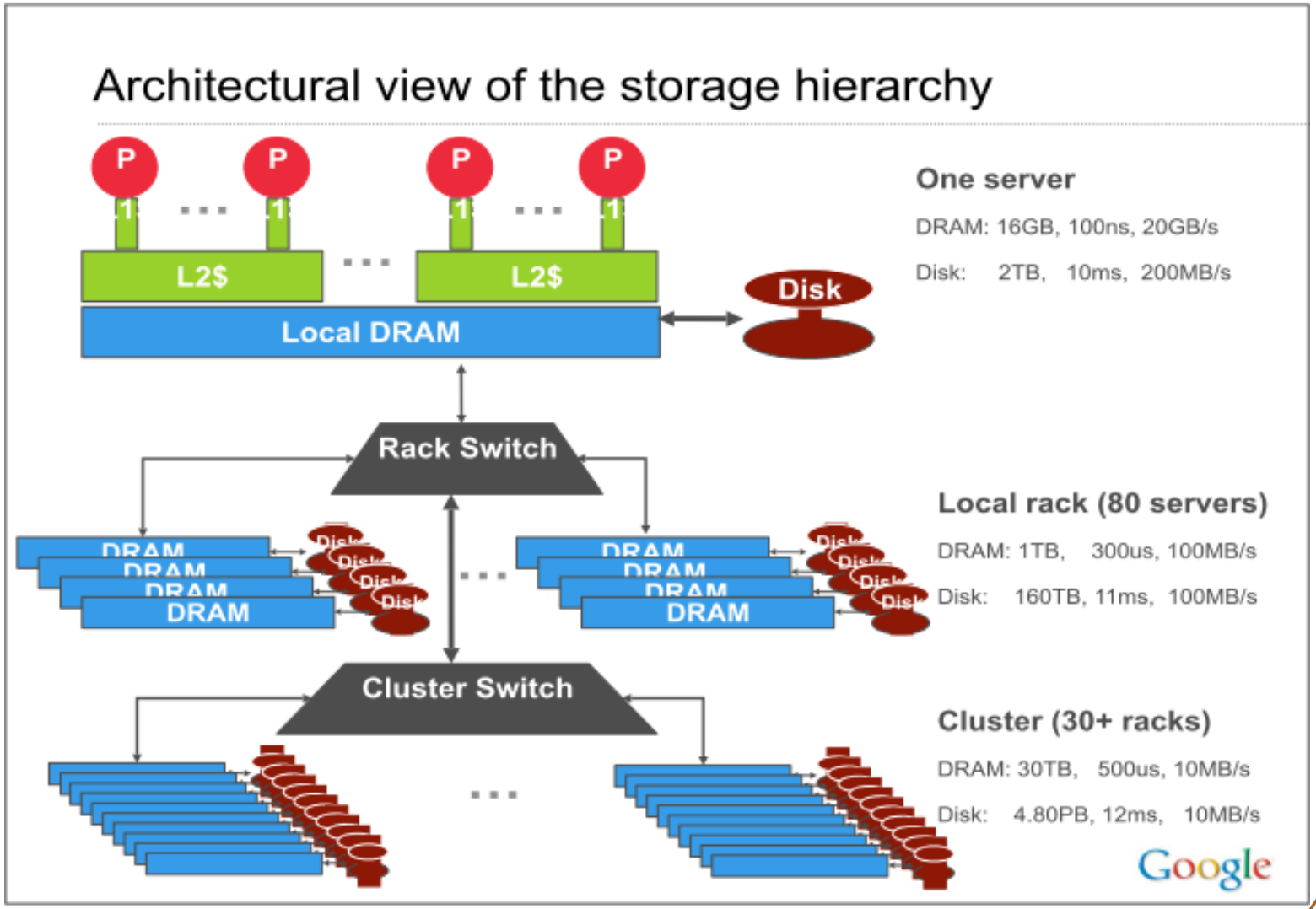
PCIe 4.0 (15.7Gbit/s/lane, 252Gbit/s for 16X)

10GBASE-PR 10 Gbit/s

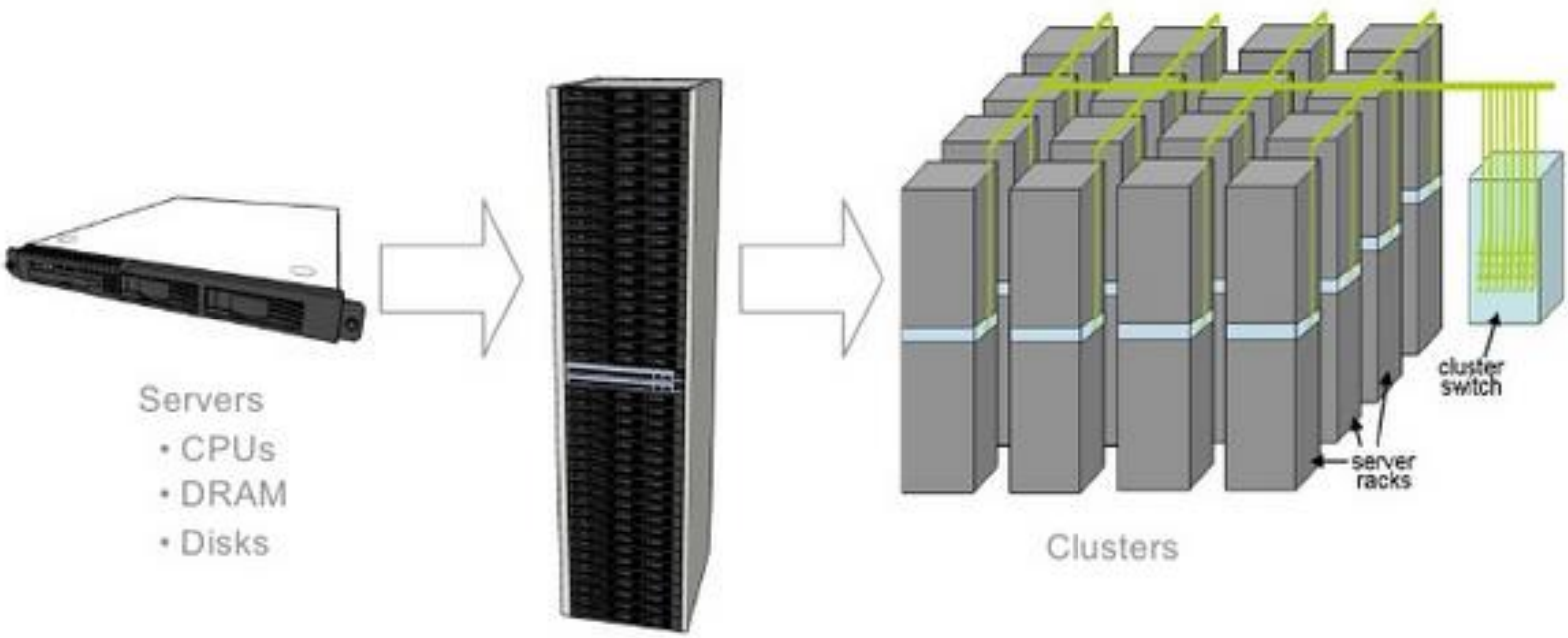
Outline

- Reiteration
- I/O
- **Storage system**
- DMA
- RAID
- Summary

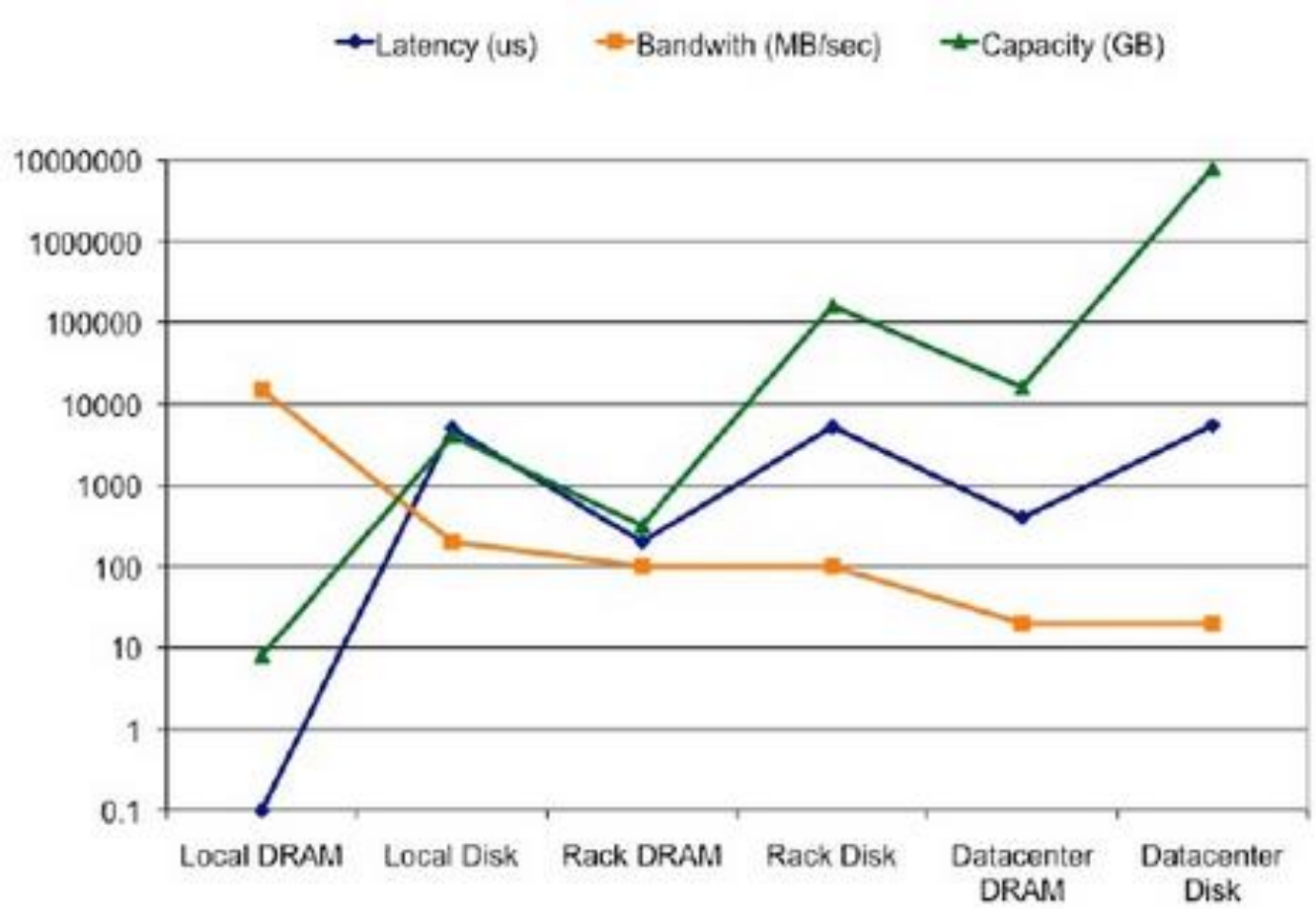
Google view of storage hierarchy



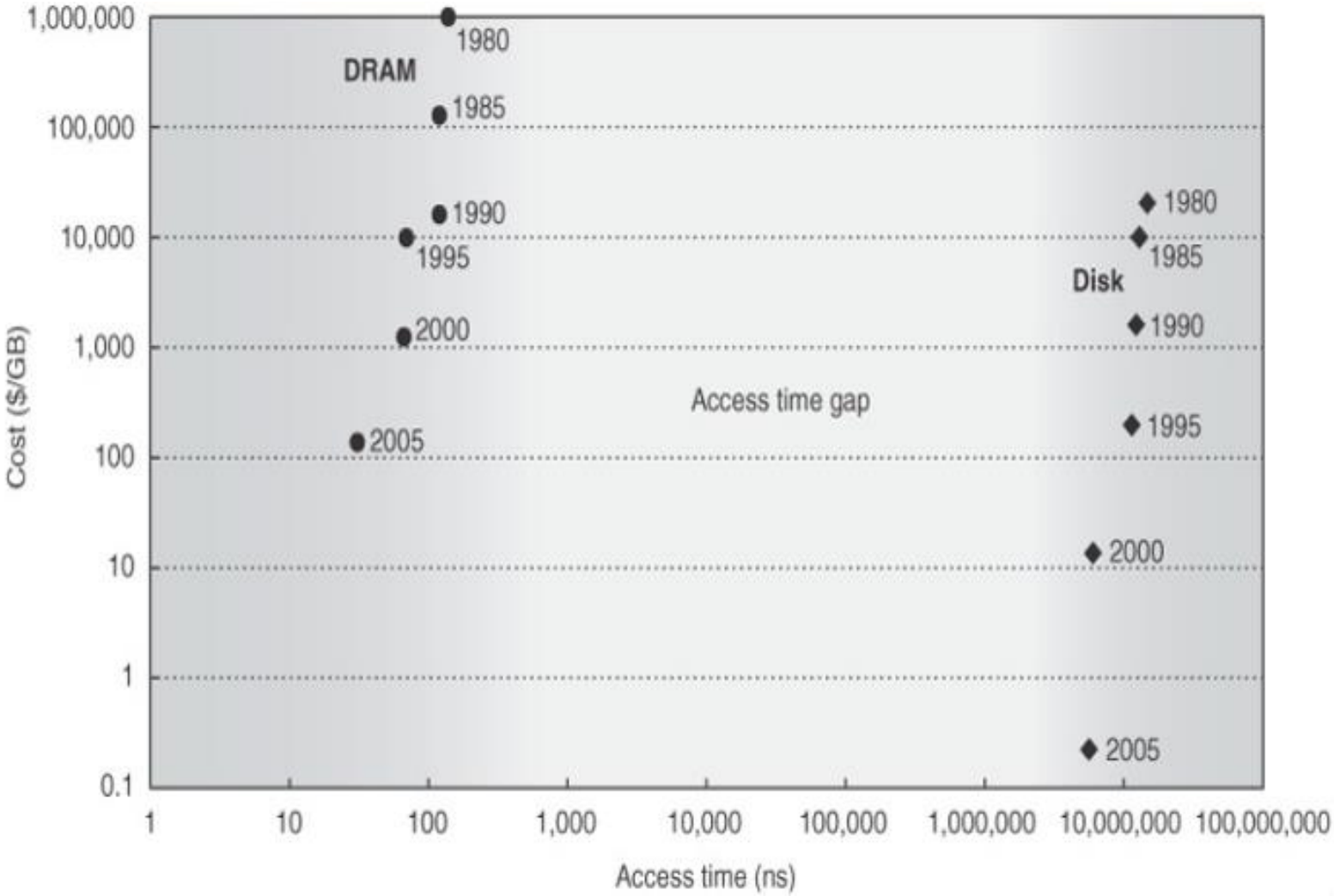
Google view of storage hierarchy



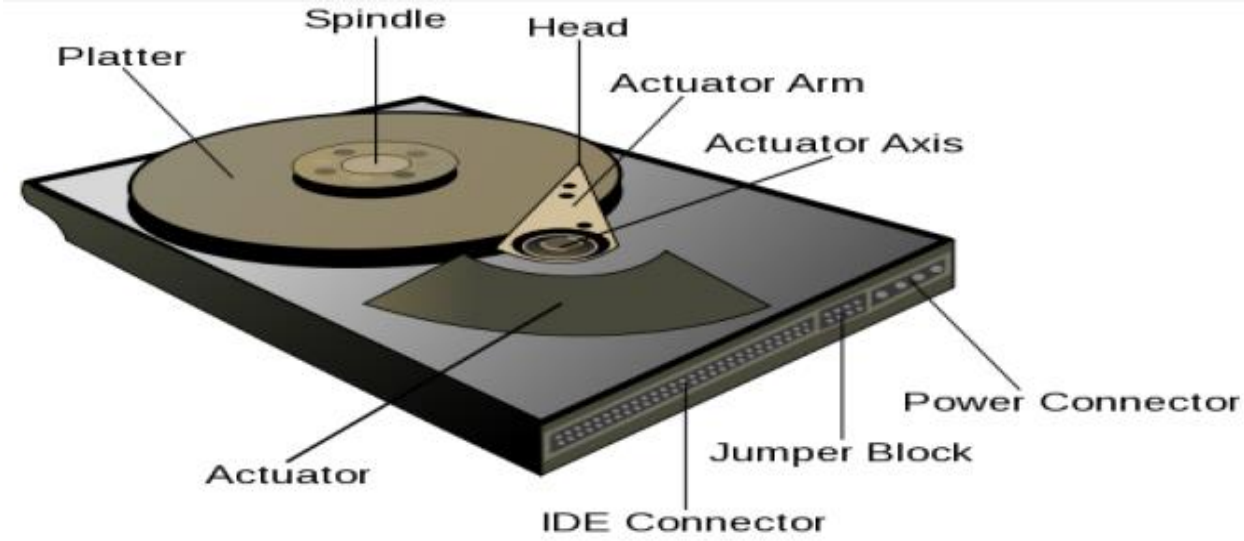
Google view of storage hierarchy

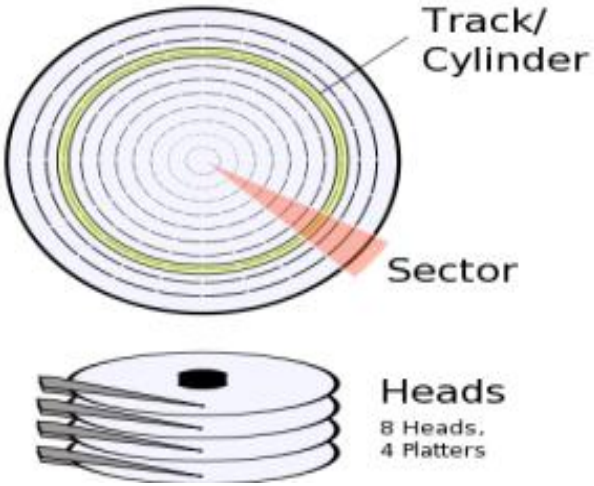


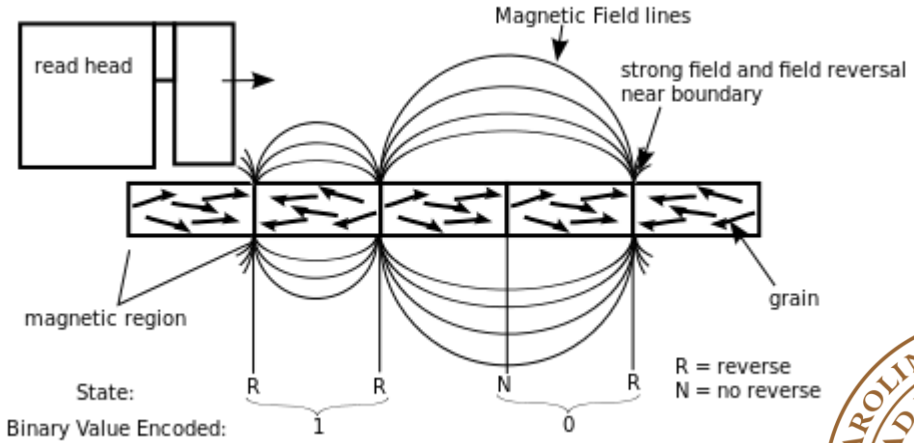
Cost vs access time



Hard disk revealed



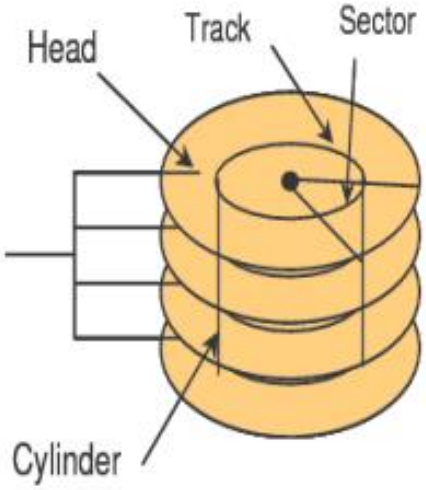




Hard disk revealed (video)

Hard disk anatomy

The organization of a disk



- **Purpose**
  - Long time, non-volatile storage
  - Large, inexpensive, slow level in the memory hierarchy
- **Characteristics:**
  - Seek time (3 - 8 -15 ms)
  - Rotational latency (2 - 4 - 8 ms)
- **Transfer rate**
  - 10 - 100 - 200 Mbyte/s
- **Capacity**
  - Terabytes
  - Quadruples every 3 years

$$T_{response} = T_{queue} + T_{service}$$

$$T_{service} = T_{controller} + T_{seek} + T_{rotation} + T_{transfer}$$

First hard disk

5 Mbyte storage
>1 ton

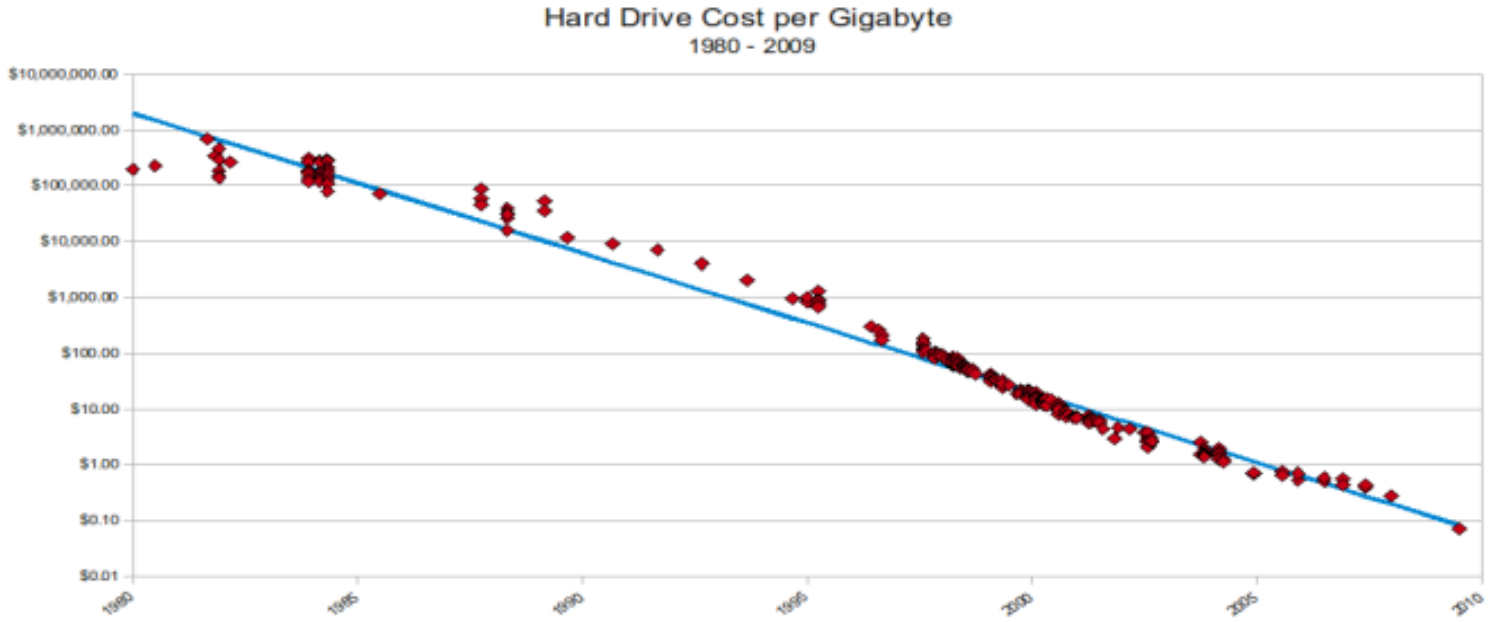
The world's first hard drive, first introduced in 1956 -- IBM's 5MB Random Access Memory Accounting: RAMAC®, magnetic-disk memory storage. It stored information on fifty disks, which spun at 1,200 rpm.

” These disks are mounted so as to rotate about a vertical axis, with a spacing of three tenths of an inch between disks. This spacing permits two magnetic heads to be positioned to any one of the 100 concentric tracks which are available on each side of each disk. Each track contains 500 alphanumeric characters. Total storage capacity: 5,000,000 characters. The two recording heads are mounted in a pair of arms which are moved, by a feed-back control system, in a radial direction to straddle a selected disk.”

Extreme hard disk

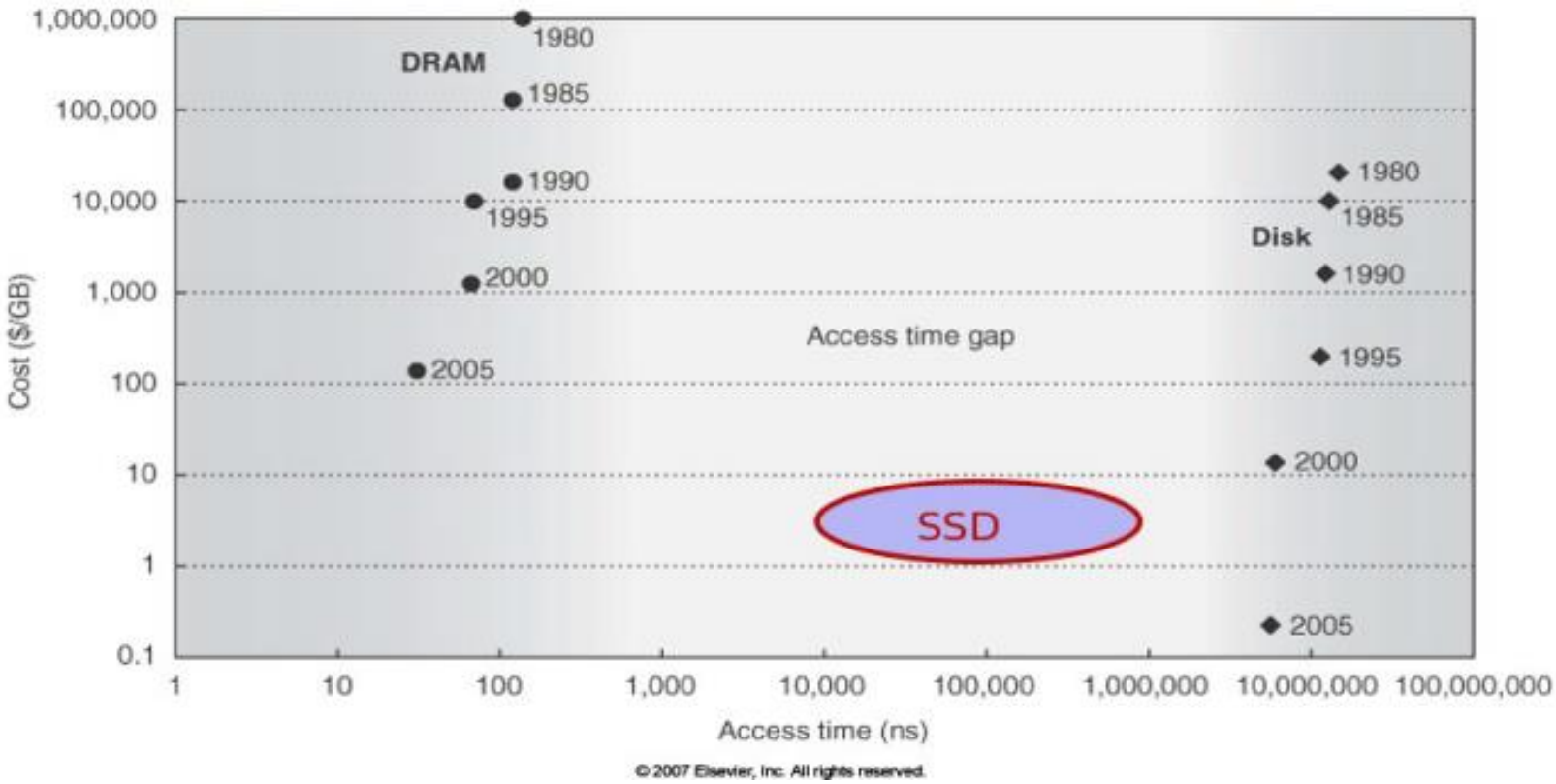
Toshiba introduced the first 0.85" hard drive and shipped 2GB and 4GB units in 2005. weigth < 10 grams

Disk technology trends



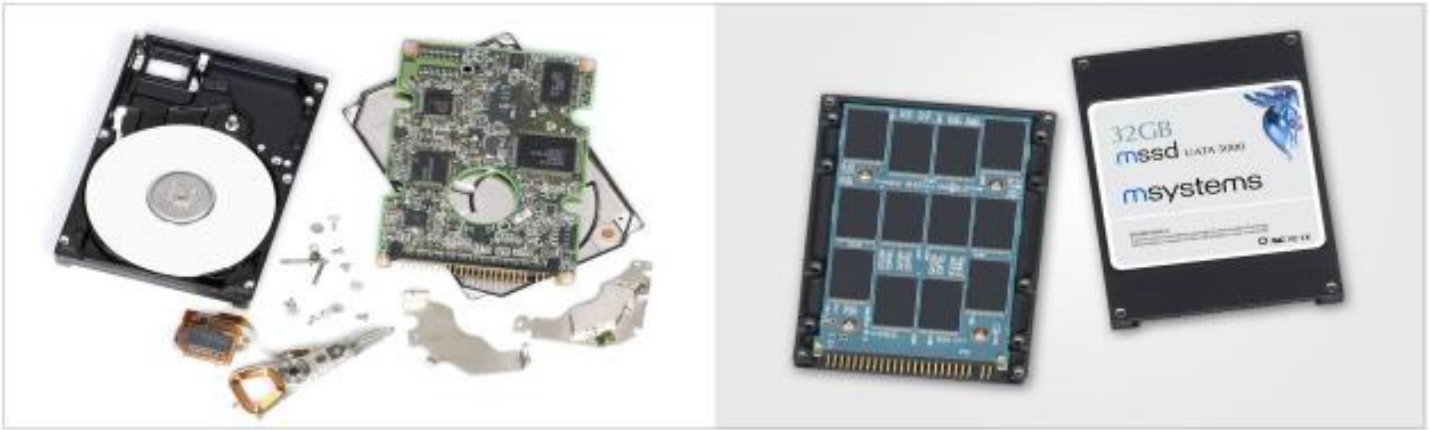
- Processing power doubles every 18 months
- Memory size doubles every 18 months
- Disk capacity doubles every 18 months
- ***Disk positioning rate (seek & rotate) doubles every ten years!***

DRAM disks?



- Can the access time gap be filled with other technologies?
- Cost is higher but SSD coming strong!

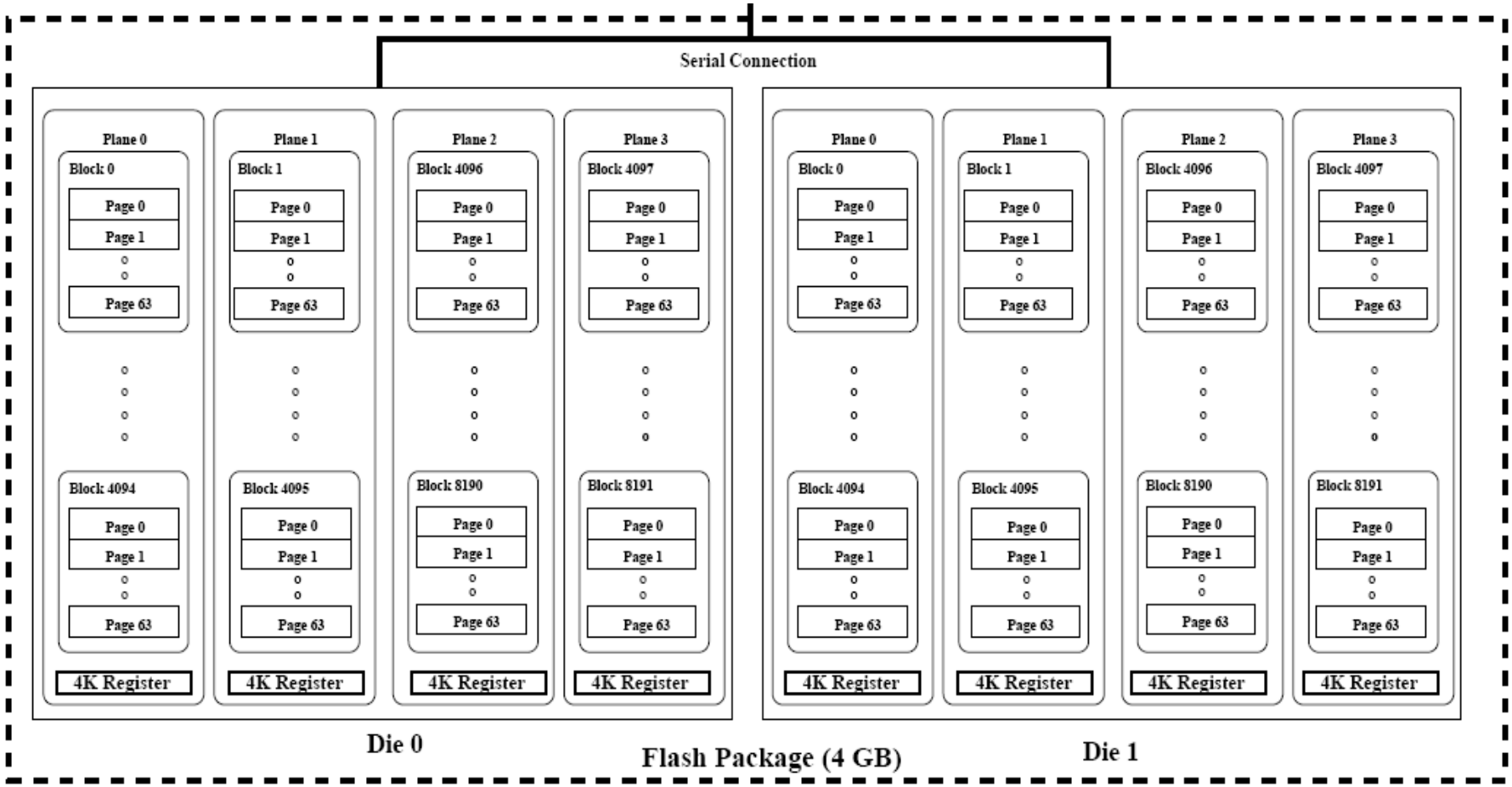
HDD vs. SSD



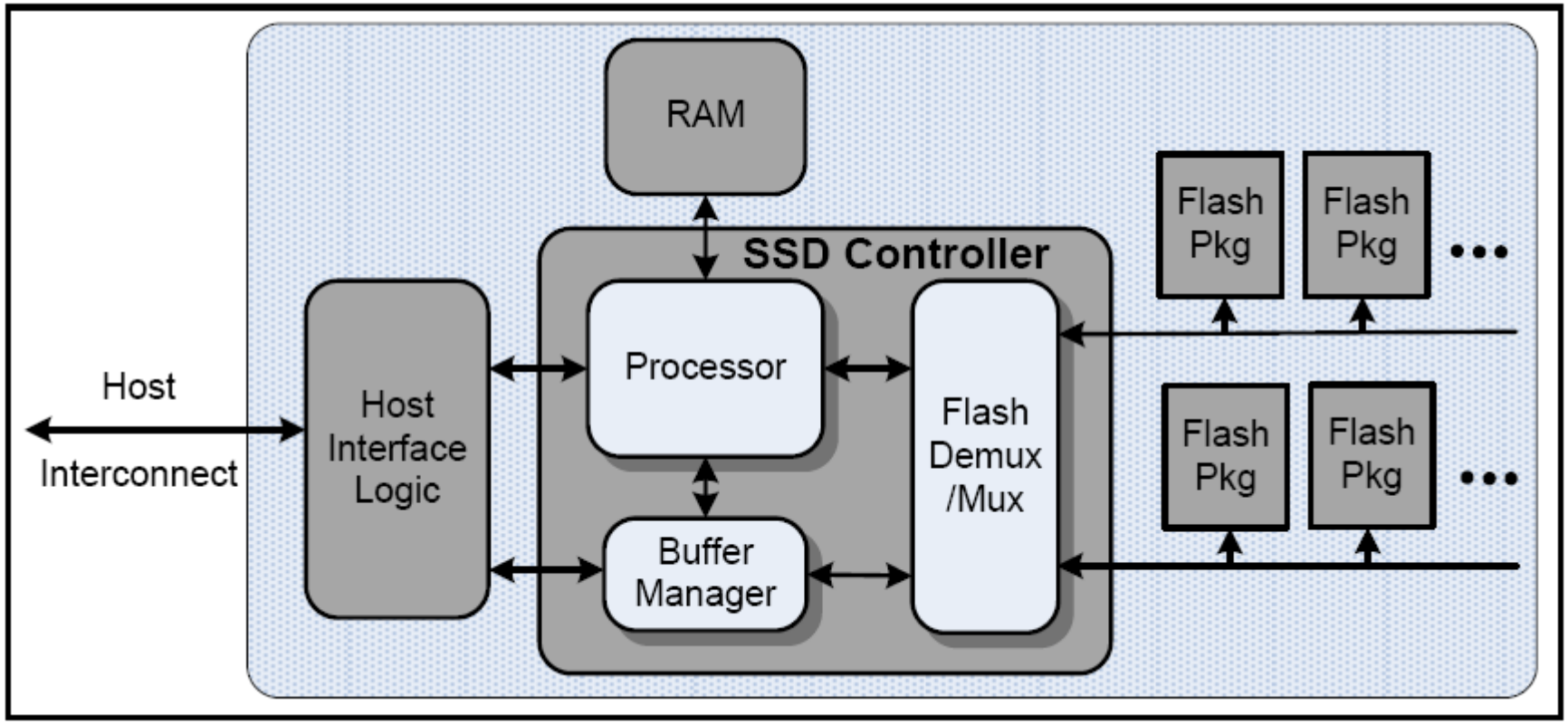
Typical read and write rates

	Drive Model	Description	Seek Time			Latency	Read XFR Rate		Write XFR Rate	
			Track to Track	Average	Full Stroke		Outer Tracks	Inner Tracks	Outer Tracks	Inner Tracks
Hard Drives	Western Digital WD7500AYYS	7200 RPM 3.5" SATA	0.6 ms	8.9 ms	12.0 ms	4.2 ms	85 MB/sec	60 MB/sec*	85 MB/sec	60 MB/sec*
	Seagate ST936751SS	15K RPM 2.5" SAS	0.2 ms	2.9 ms	5.0 ms*	2.0 ms	112 MB/sec	79 MB/sec	112 MB/sec	79 MB/sec
Flash SSDs	Transcend TS8GCF266	8GB 266x CF Card	0.09ms				40 MB/sec		32 MB/sec	
	Samsung MCAQE32G5APP	32G 2.5" PATA	0.14ms				51 MB/sec		28 MB/sec	
	Sandisk SATA5000	32G 2.5" SATA	0.125ms				68 MB/sec		40 MB/sec	

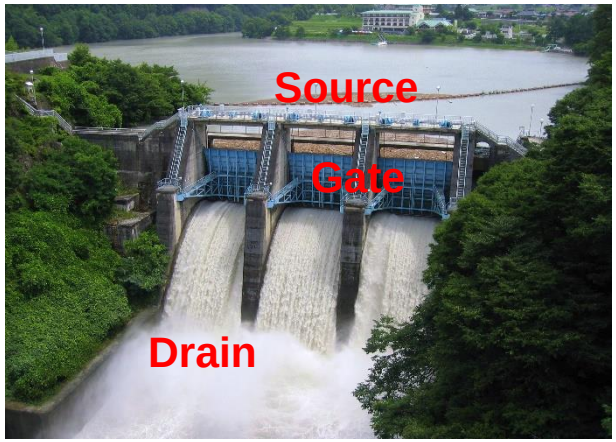
Samsung flash internals

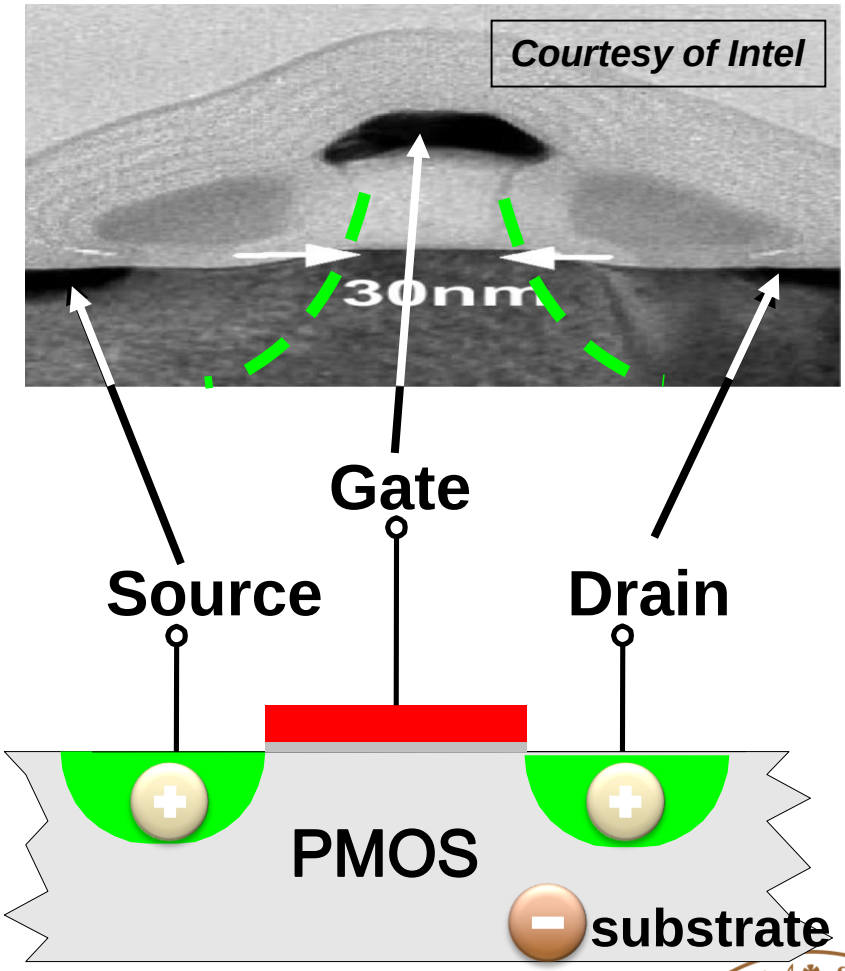


SSD Logic components



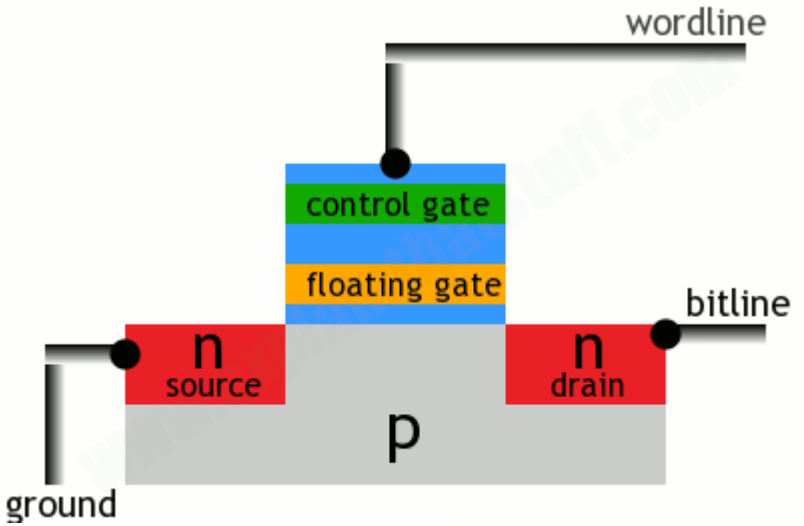
Transistor

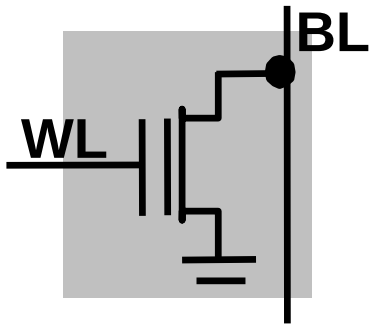


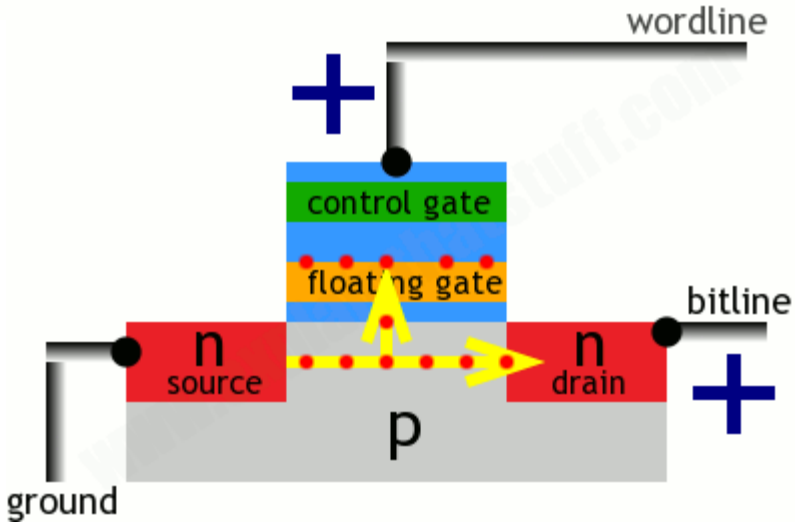


Principle:
By applying a voltage to the gate the current between the drain and the source can be controlled. Several different types of transistors exist with different properties.

Flash memory cell







EPROM, EEPROM and Flash has different ways of controlling the charge of the floating gate

Power consumption

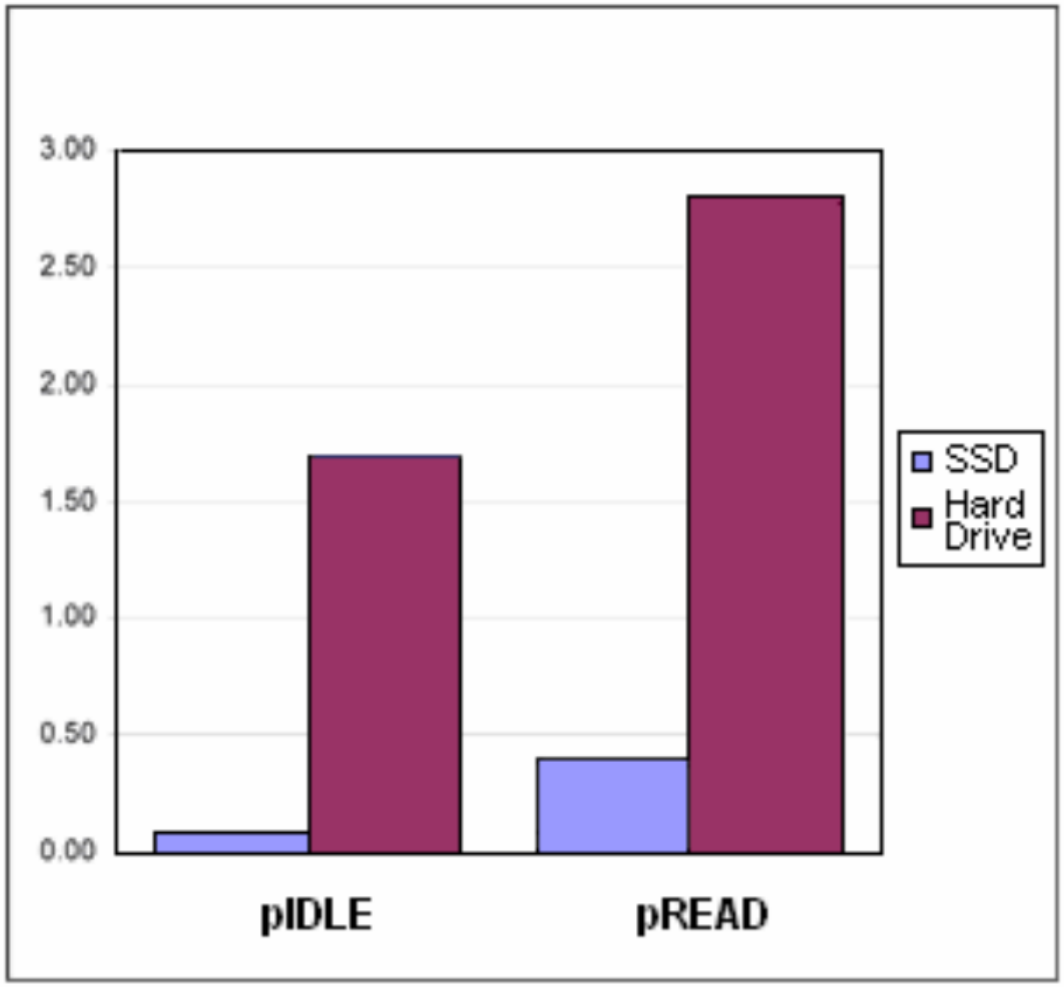


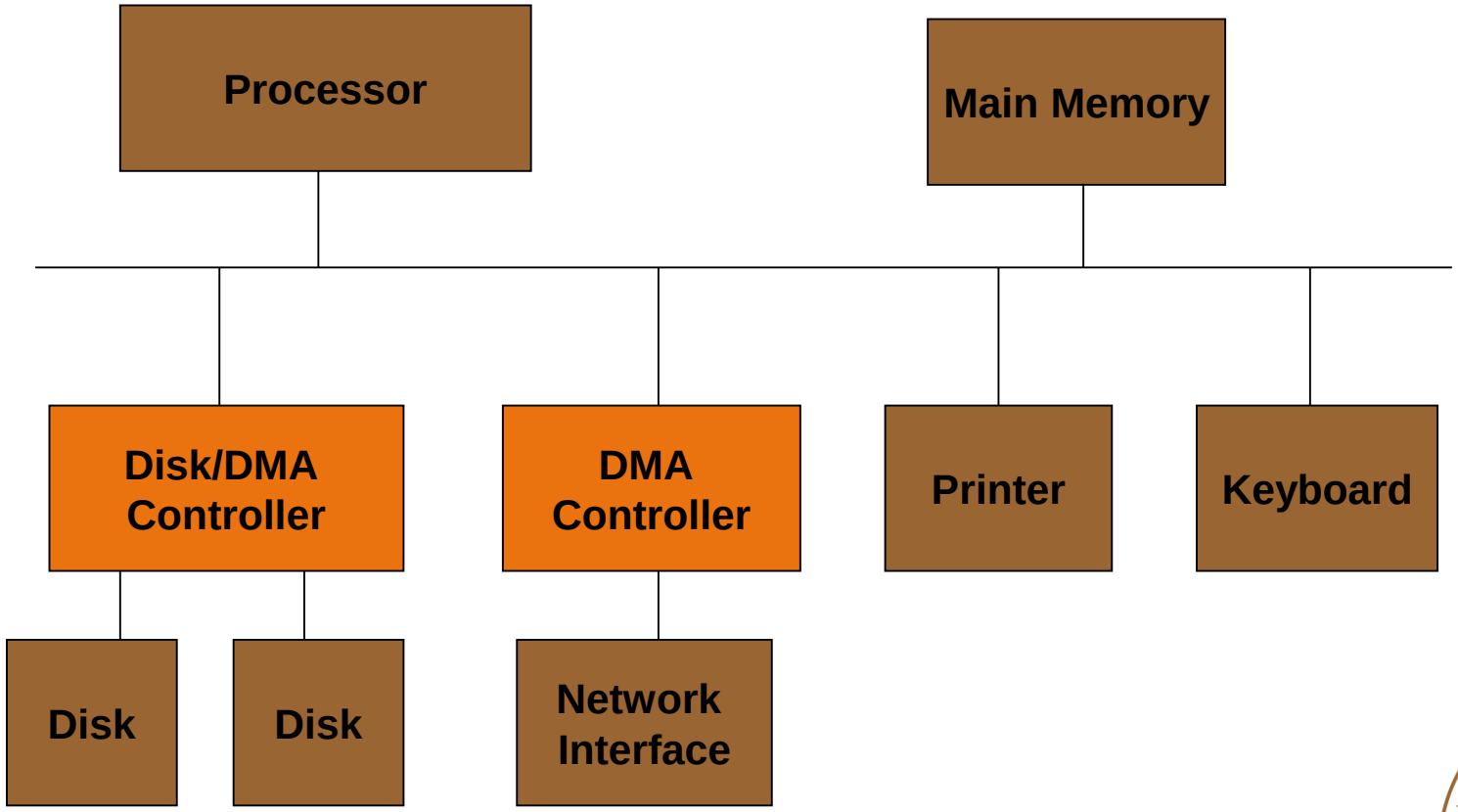
Figure 1: Typical Power Consumption in Watts

Outline

- Reiteration
- I/O
- Storage Systems
- **DMA**
- RAID
- Summary

Direct memory access (DMA)

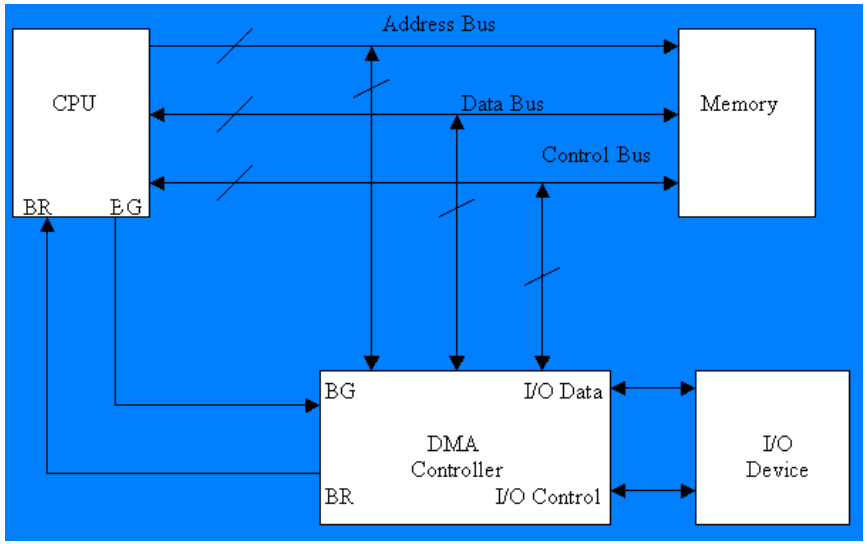
- DMA is a feature of computer systems that allows certain hardware subsystems to access main system (RAM) memory independently of the CPU



DMA: operation

Data transfer between I/O and memory

- Data transfer preparation
  - DMA Address Register contains the memory address, Word Count Register
  - Commands specify transfer options, DMA transfer mode, the direction
- Control grant
  - DMA sends a Bus Request (setting BR to 1)
  - When it is ready to grant this request, the CPU sets it's Bus grant signal, BG to 1
- Data transfer modes
  - Bust mode/Cycle stealing mode/Transparent mode



DMA: performance

Example: 1000 transfers of 1 byte

10 Mbyte/s transfer rate ⇒ 0.1 µs/byte

1000 bytes ⇒ 100 µs

- **Interrupt driven**
  - 1000 interrupts at 2 µs each
  - 1000 interrupt service routines at 98 µs each
  - Totals 0.1 CPU seconds
- **DMA**
  - 1 DMA set-up sequence at 50 µs
  - 1 interrupt at 2 µs
  - 1 interrupt service sequence at 98 µs
  - Totals 0.00015 CPU seconds

DMA: example on massive MIMO testbed

imec

KATHOLIEKE UNIVERSITEIT LEUVEN

SONY

ERICSSON

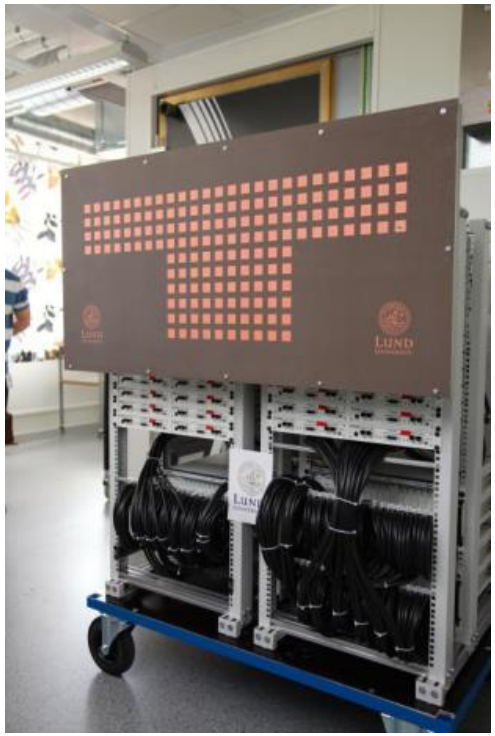
Linköpings universitet

Infineon

TECHNIKON

NATIONAL INSTRUMENTS

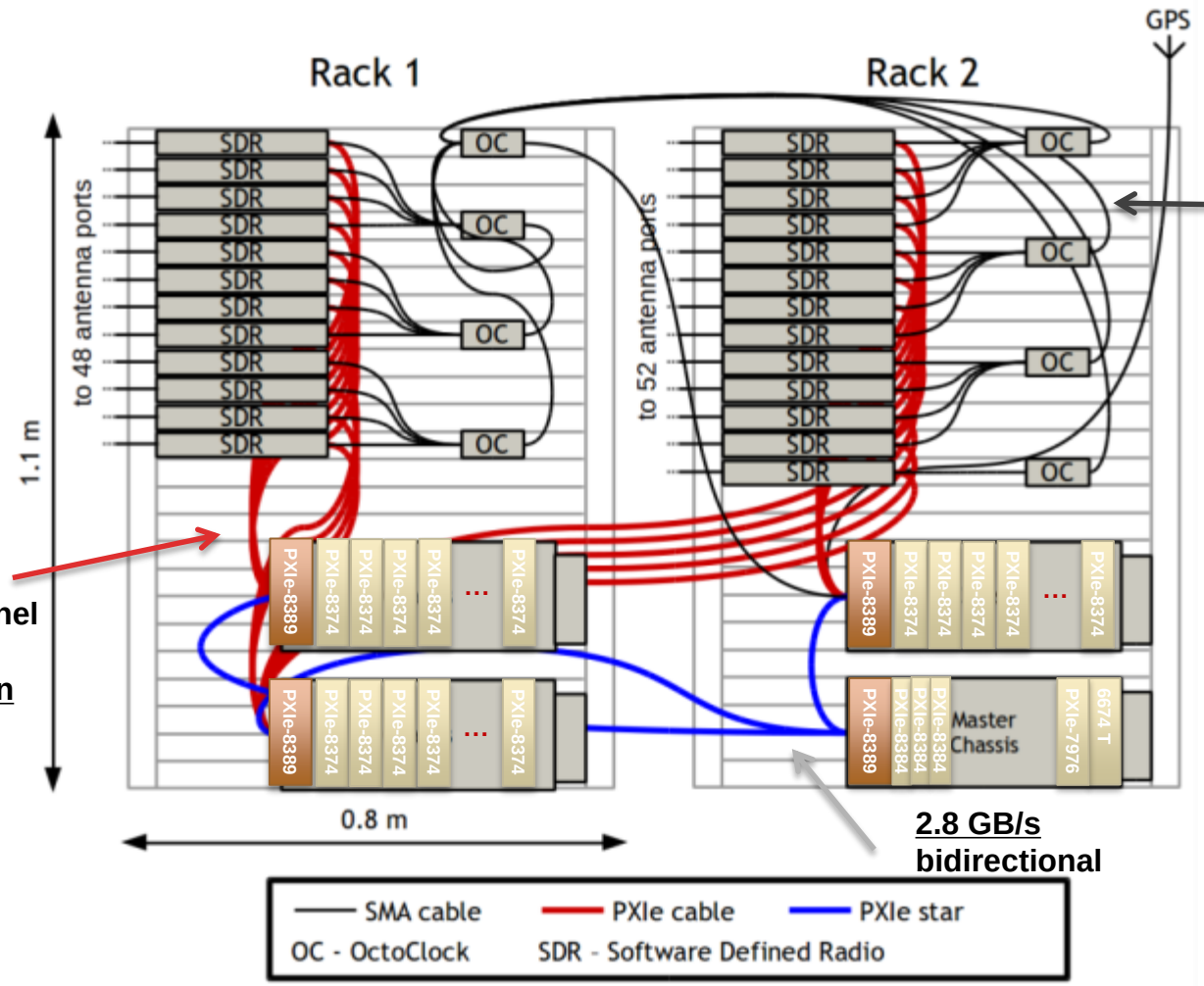
intel



100X

System component & architecture

System component & architecture



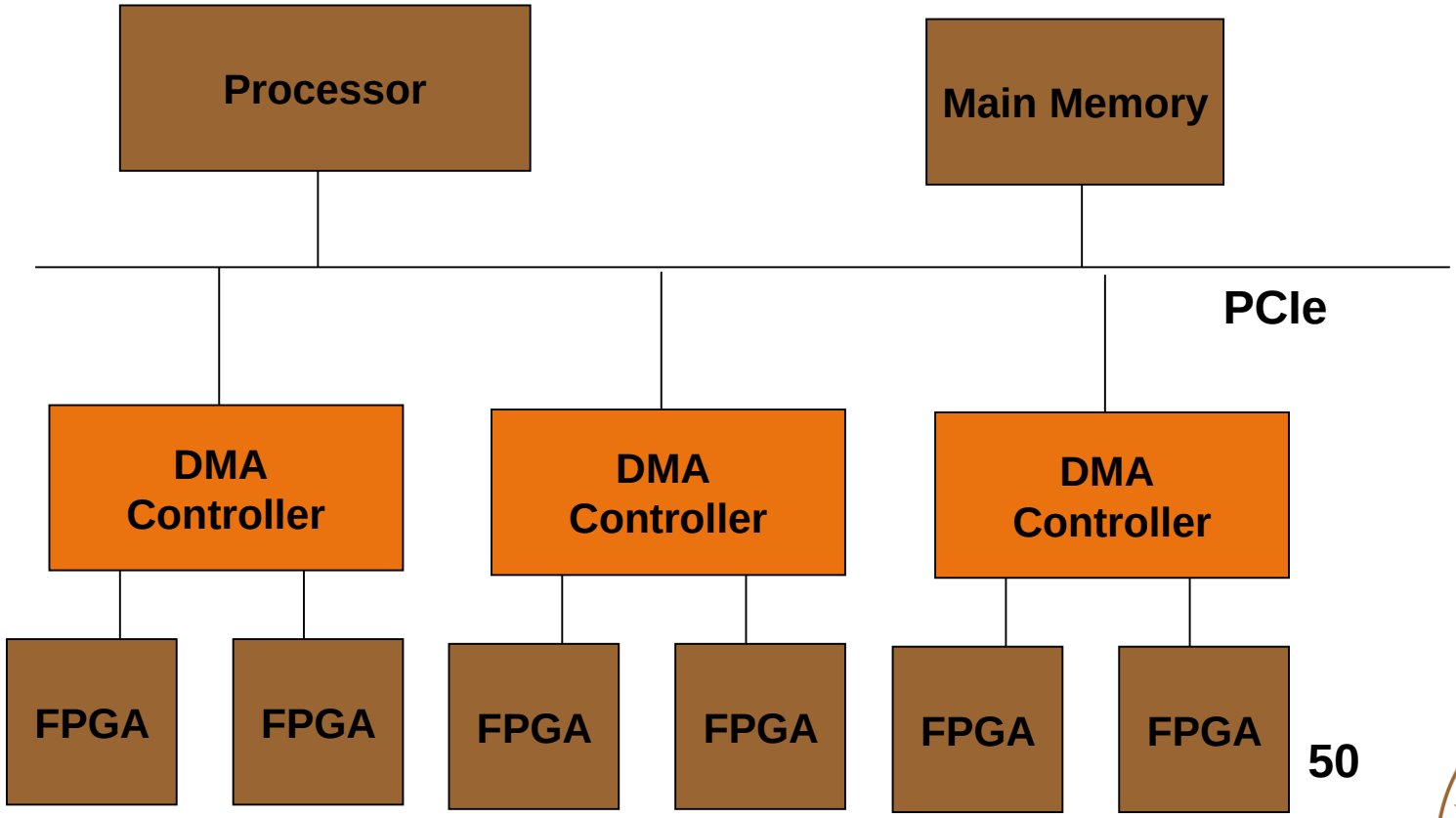
Clock/PPS Distribution network

10 DMA channel Up to 800 MB/s/direction

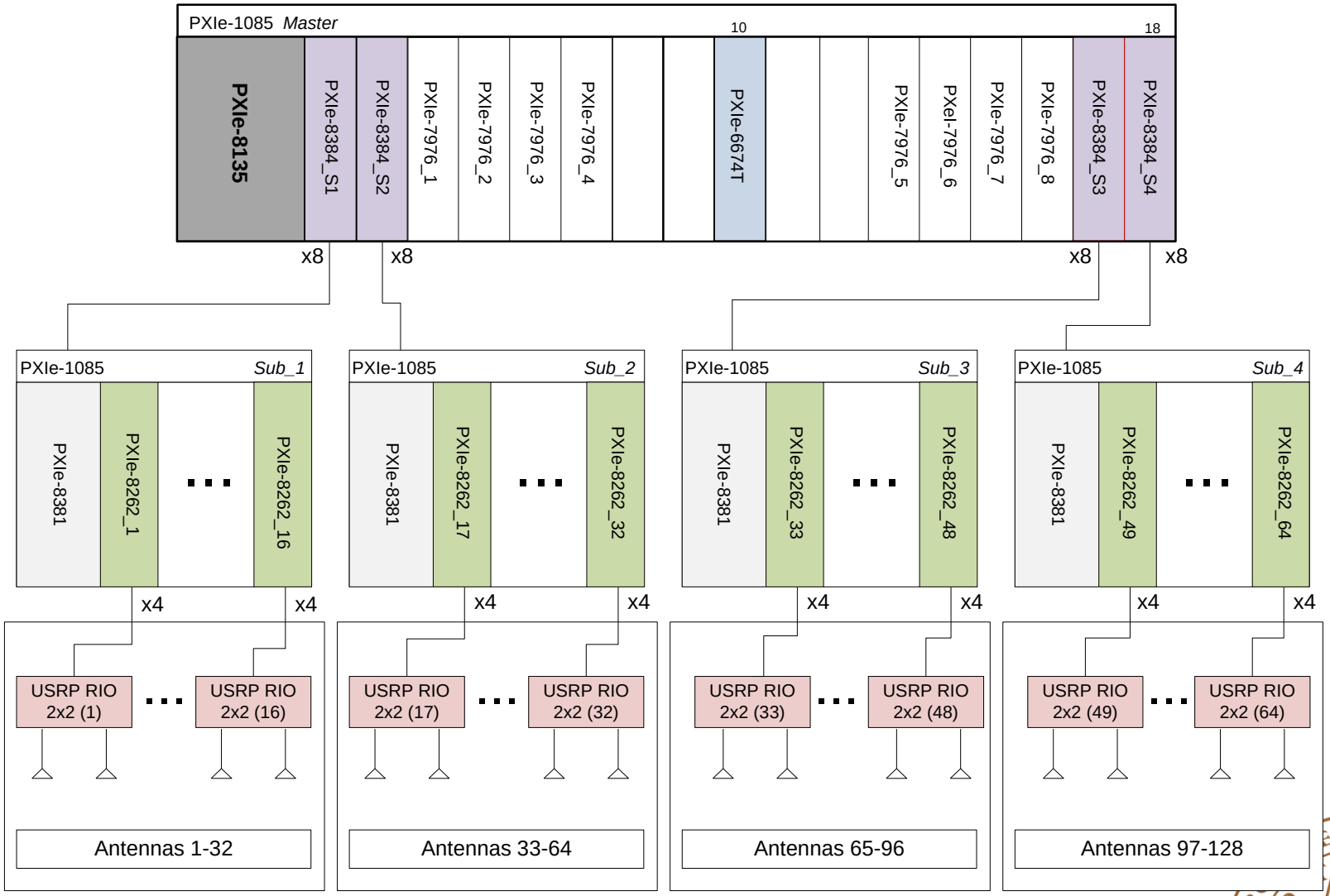
2.8 GB/s bidirectional

Computing platform

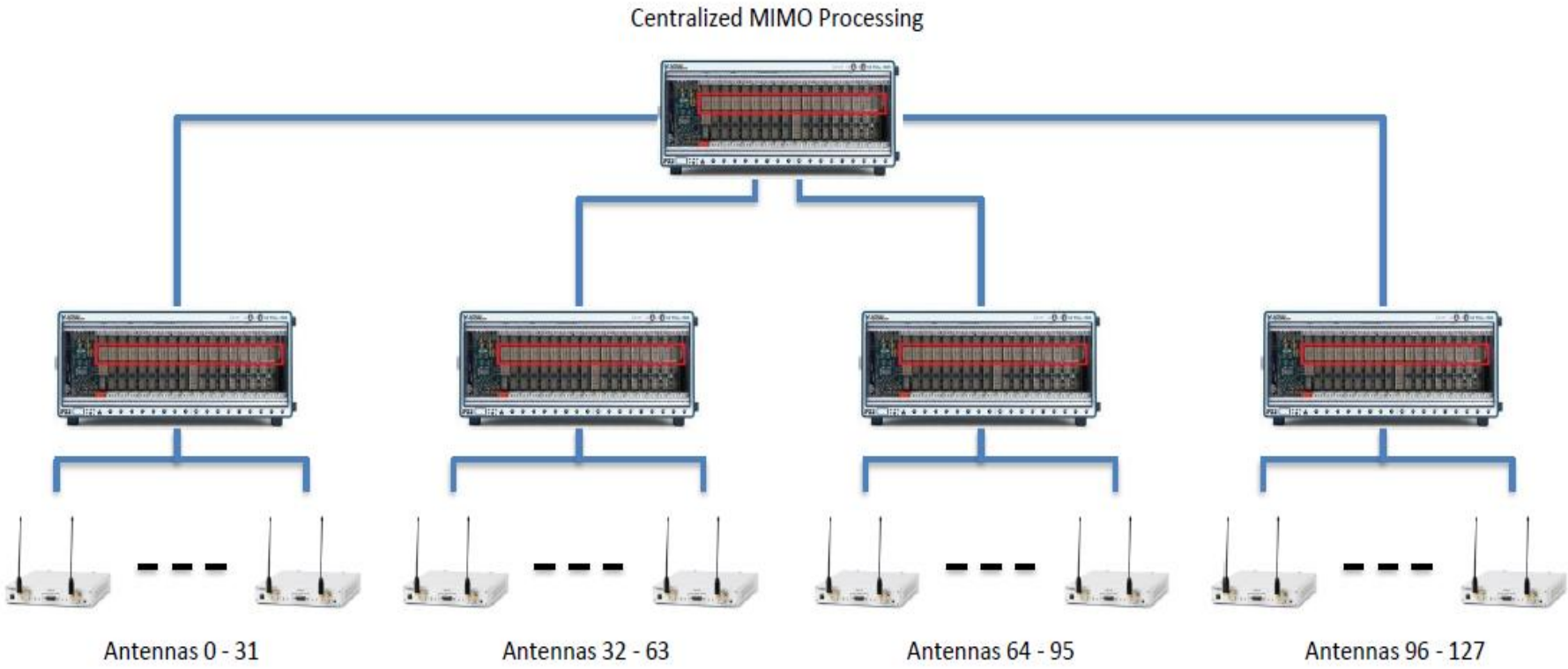
- I7 CPU as controller and 50 Xilinx FPGA as processing platform



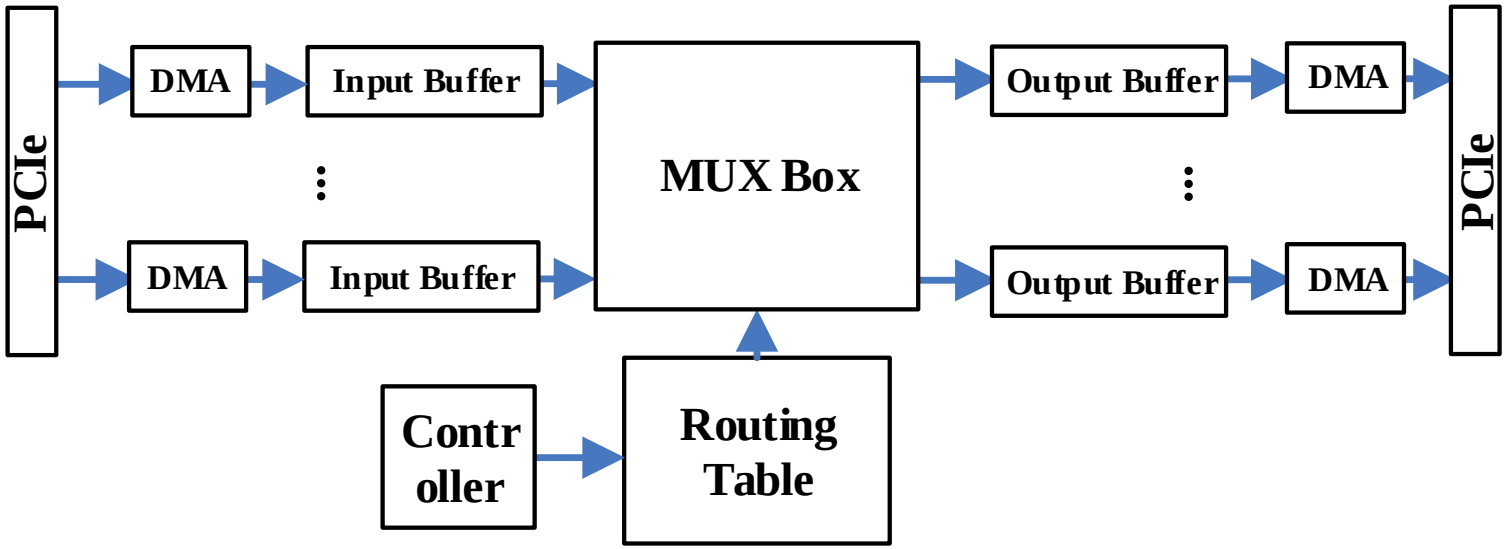
Connections



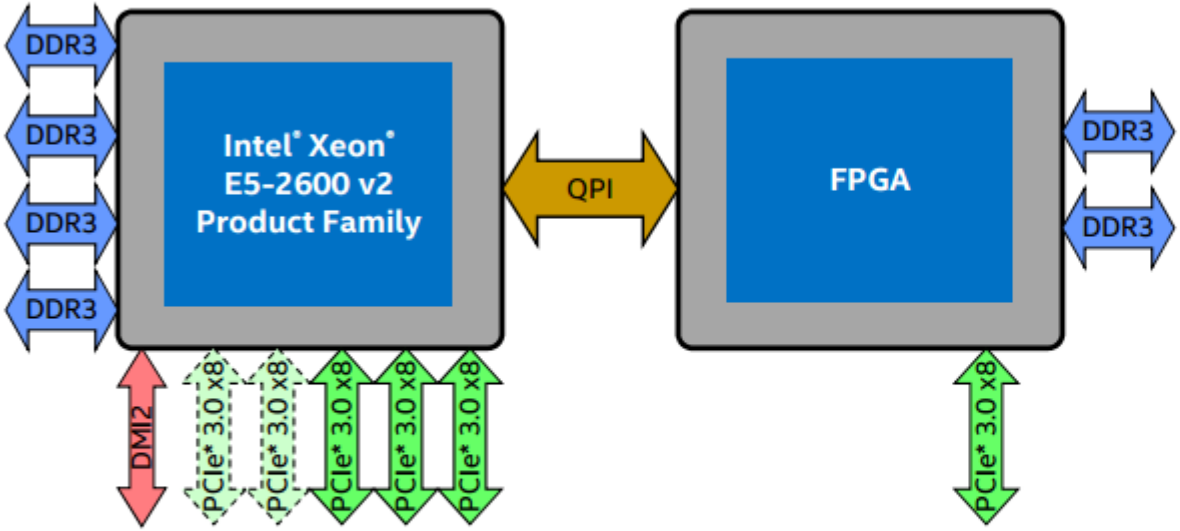
Connections



Connections



Intel's new computing platform



Processor	Intel® Xeon® E5-26xx v2 Processor
FPGA Module	Altera Stratix V
QPI Speed	6.4 GT/s full width (target 8.0 GT/s at full width)
Memory to FPGA Module	2 channels of DDR3 (up to 64 GB)
Expansion connector to FPGA Module	PCIe 3.0 x8 lanes - maybe used for direct I/O e.g. Ethernet
Features	Configuration Agent, Caching Agent,, (optional) Memory Controller
Software	Accelerator Abstraction Layer (AAL) runtime, drivers, sample applications

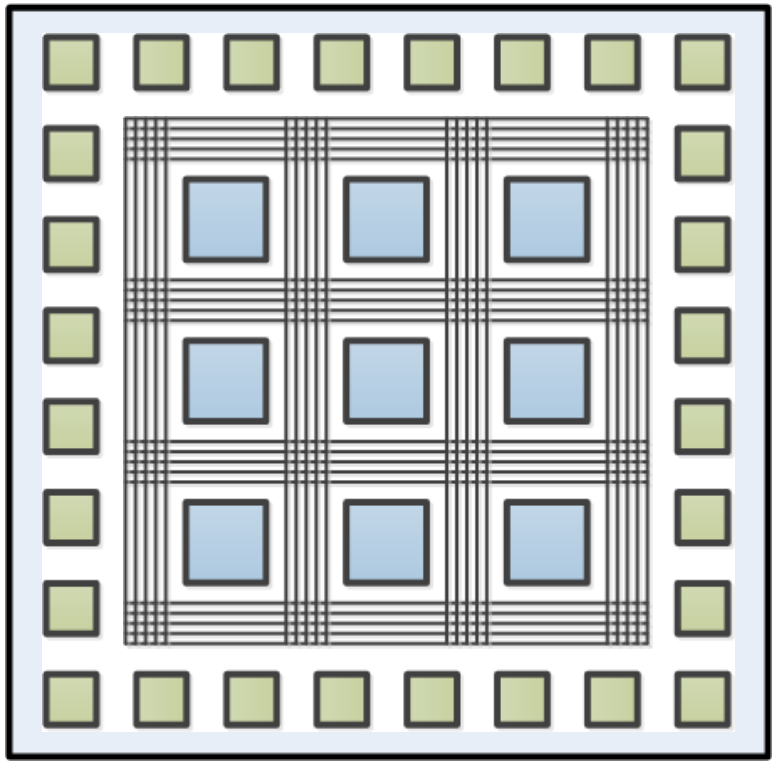
Heterogeneous architecture with homogenous platform support

intel | 14

FPGA

- What is FPGA?
  - Field Programmable Gate Array

Configurable logic blocks

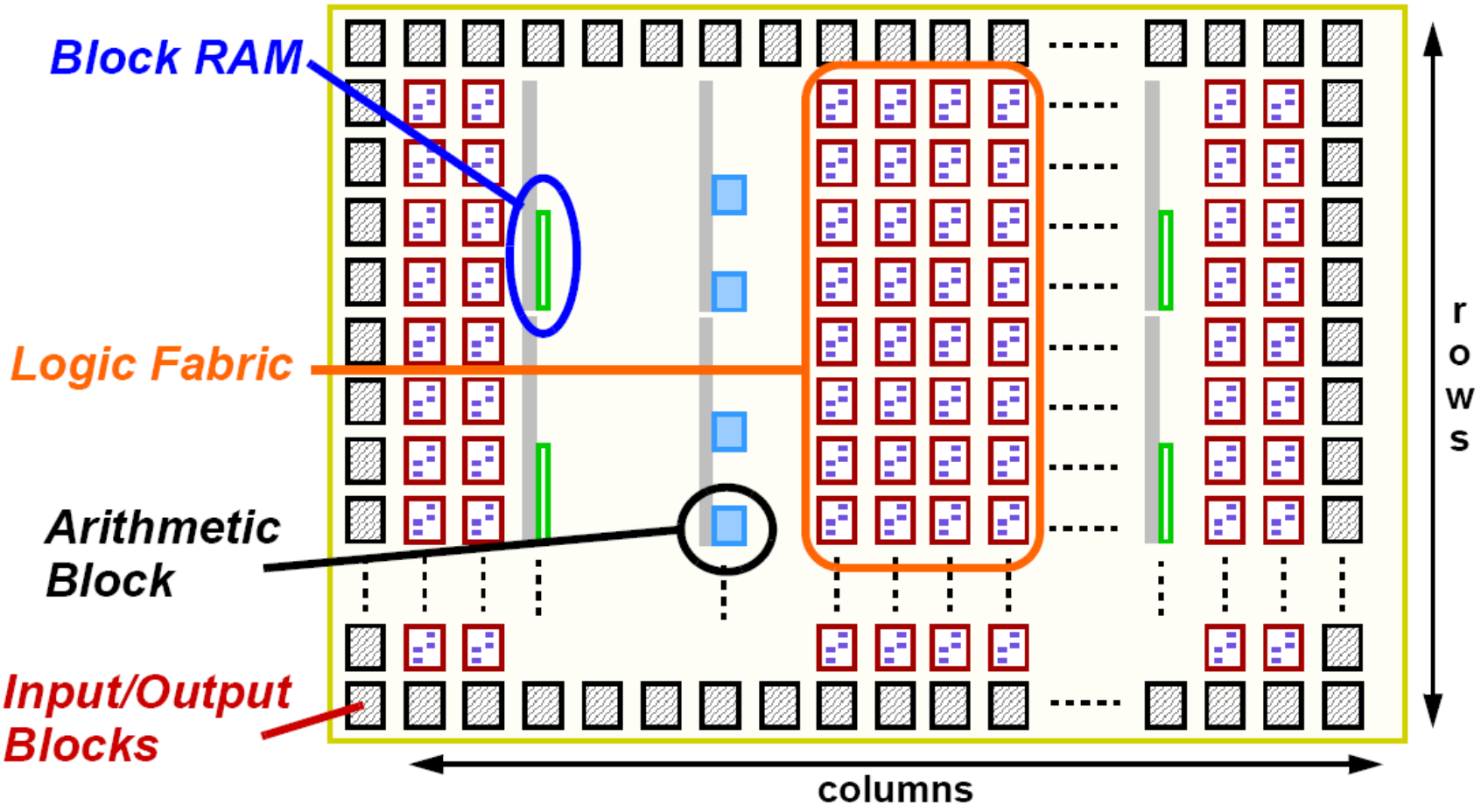


Interconnects

Configuration memory

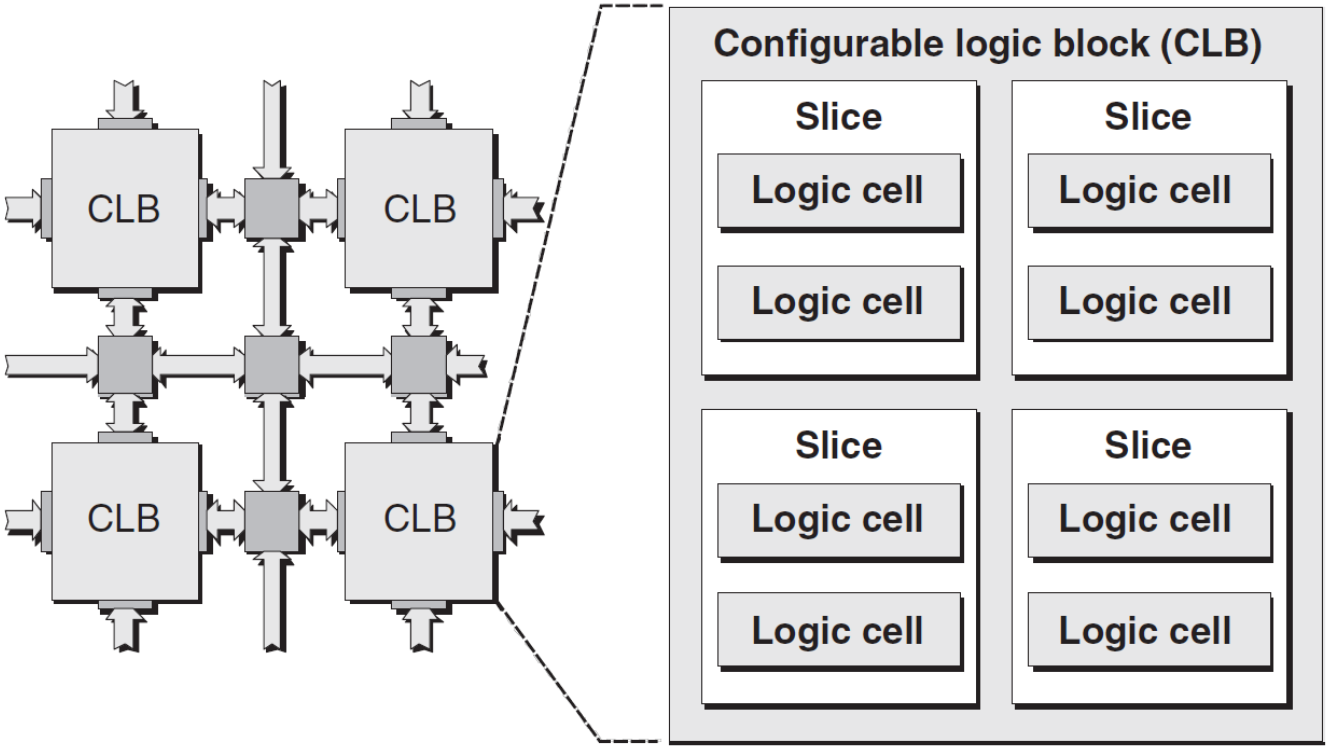
IO blocks

FPGA


Block RAM
Logic Fabric
Arithmetic Block
Input/Output Blocks
rows
columns

FPGA

Configurable logic block (CLB) contains several slices



FPGA vs CPU

	Intel Itanium 2	Xilinx Virtex-II Pro (XC2VP100)
Technology	0.13 µm	0.13 µm
Clock speed	1.6 GHz	180 MHz
Internal memory bandwidth	102 GBytes/S	7.5 TBytes/S
# Processing units	5 FPU (2 MACs+1 FPU) 6 MMU 6 Integer units	212 FPU or 300+Integer units or …
Power consumption	130 W	15 W
Peak performance	8 GFLOPs	38 GFLOPs
Sustained performance	~2GFLOPs	~19 GFLOPs
IO/External memory bandwidth	6.4 GBytes/S	67 GBytes/S

Outline

- Reiteration
- I/O
- Storage Systems
- DMA
- **RAID**
- Summary

Reliability / Availability – Dependability

- **Definitions:**
  - Reliability – Is anything broken?
  - Availability – Is the system available for the user?
  - Dependability – Is the system doing what it is supposed to do?
- **Why is this an issue?**
  - Small disks and large disks cost the same / byte
  - An array of N small disks can achieve higher bandwidth than one large disk
  - However, the reliability is 1/N of the reliability of a single disk

Interlude - Google experience - Jeff Dean

The Joys of Real Hardware

Typical first year for a new cluster:

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
~1 network rewiring (rolling ~5% of machines down over 2-day span)
~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
~5 racks go wonky (40-80 machines see 50% packetloss)
~8 network maintenances (4 might cause ~30-minute random connectivity losses)
~12 router reloads (takes out DNS and external vips for a couple minutes)
~3 router failures (have to immediately pull traffic for an hour)
~dozens of minor 30-second blips for dns
~1000 individual machine failures
~thousands of hard drive failures
slow disks, bad memory, misconfigured machines, flaky machines, etc.

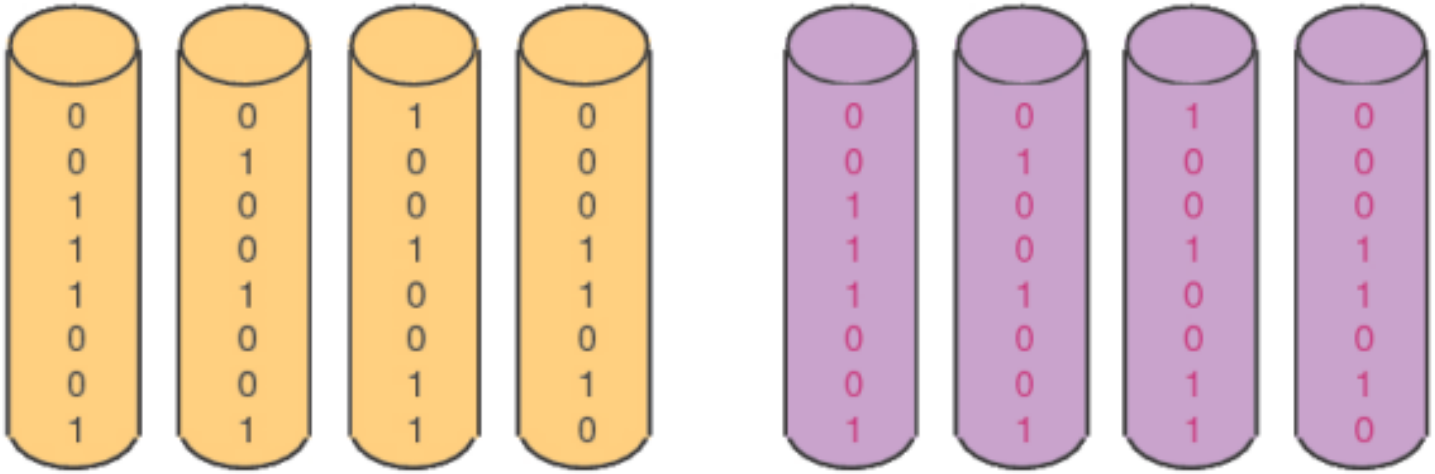
Long distance links: wild dogs, sharks, dead horses, drunken hunters, etc.

RAID

Redundant Array of Inexpensive (Independent) Disks

	RAID level	Failures tolerated	Overhead 8 data disks	comment
0	striped	0	0	JBOD, common
1	mirrored	1-8	8	high overhead
2	ECC	1	4	not used
3	bit parity	1	1	synchronized drives
4	block parity	1	1	
5	block parity distributed	1	1	common
6	row-diagonal dual parity	2	2	high availability
01	mirrored stripes	1-8	8	
10	striped mirrors	1-8	8	

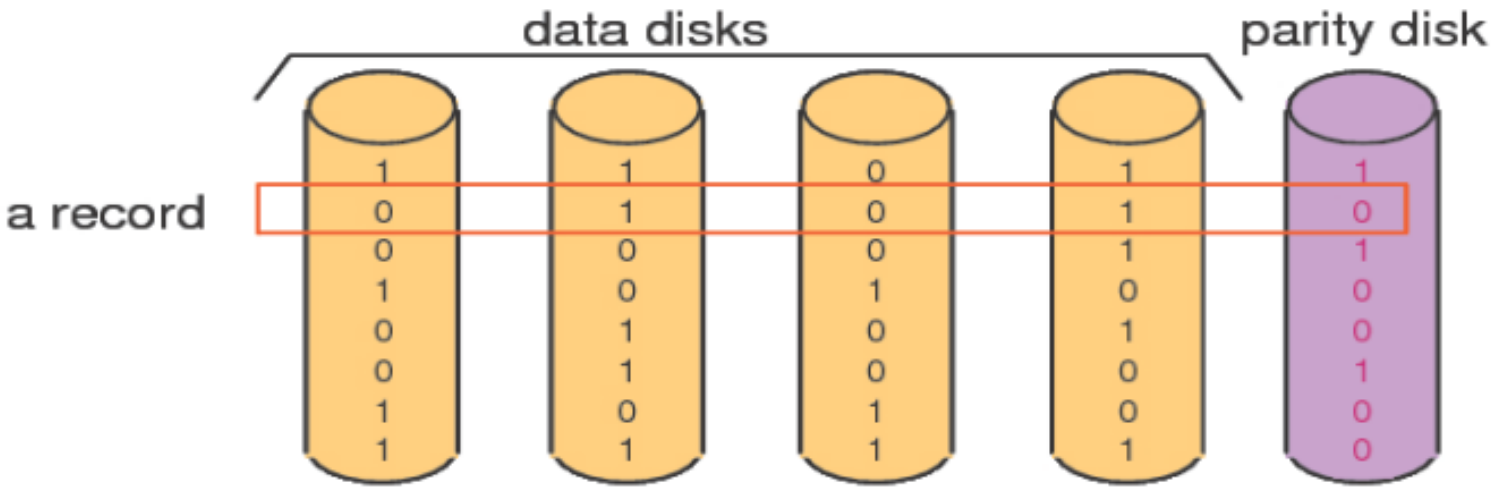
Disk mirroring – RAID-1



- Each disk is fully duplicated onto its "shadow"
- Logical write = two physical writes
- Easy recovery, just use the copy
- 100% capacity overhead

Targeted for high I/O rate, high availability systems

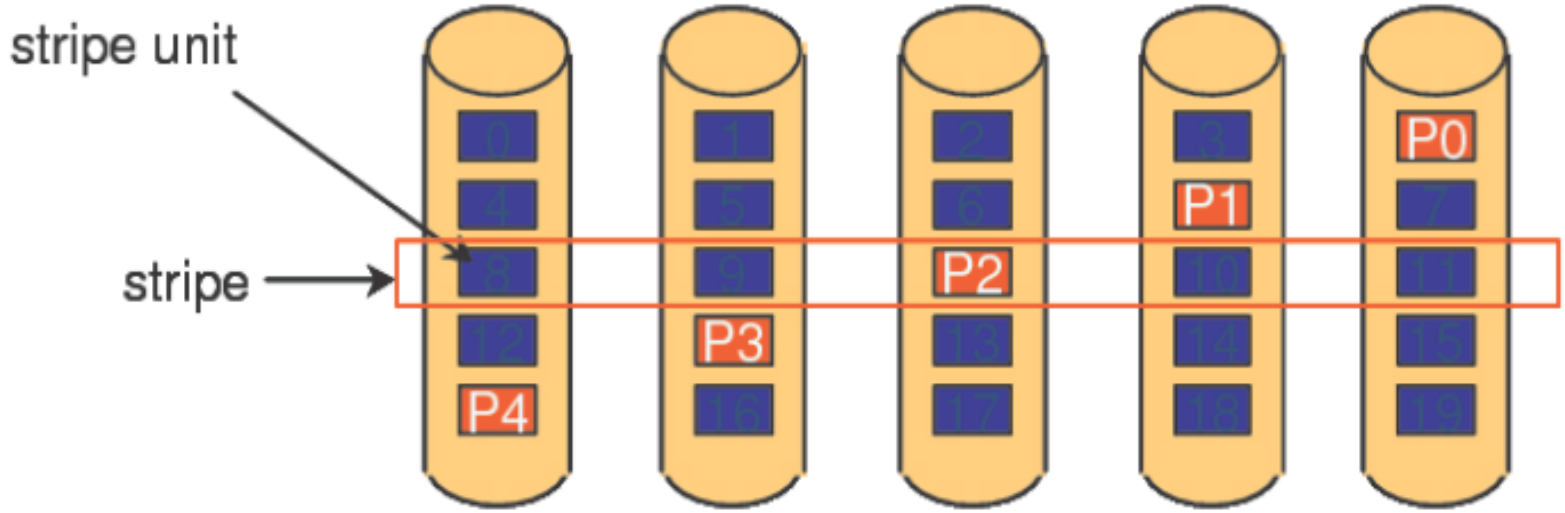
Bit Parity – RAID 3



- Parity is computed across recovery group
- 20% capacity cost (this config), wider arrays reduce capacity cost but also decreases expected availability
- Logically a single high capacity, high transfer rate disk

Targeted for high bandwidth applications: Scientific, image processing

Block-Interleaved Parity – RAID 5



- "Small" reads can occur in parallel
- A logical write becomes four physical I/O-ops
- Independent writes possible due to interleaved parity

Targeted for mixed applications with high-performance I/O needs

Summary I/O

I/O:

- **I/O performance is important!**
- **The task of the I/O system designer:**
  - meet performance needs
  - cost-effective
  - reliability, availability
- **I/O system parts**
  - CPU interface
  - Interconnect technology
  - Device performance

Disks:

- **Disks have moving parts leading to long service times**
- **RAID disk arrays provide high bandwidth, high capacity disk storage at a reasonable cost**
- **SSD is faster and more expensive**