DATA STORAGE TECHNOLOGIES & NETWORKS

(CS C446, CS F446 & IS C446)

LECTURE 06 – STORAGE

Memory Requirements

- Data can be “per-computation” or “persistent”
  - Separate Memory/Storage for both forms
    - Technology driven:
      - Volatile vs. Non-volatile
    - Cost driven:
      - Faster and Costlier vs. Slower and cheaper

Per-Computation Data

- Memory Bandwidth Requirement
  - **How much data is needed to keep the processor busy?**

	DEC VAX 11/780 (circa '80)	Early pipelines (circa '90)	Superscalars (circa '00)	Hyperthreaded Multi-cores (circa '08)
Clock cycle	250ns	25ns	1ns	0.4ns

Per-Computation Data

- Memory Bandwidth Requirement
  - **How much data is needed to keep the processor busy?**

	DEC VAX 11/780 (circa '80)	Early pipelines (circa '90)	Superscalars (circa '00)	Hyperthreaded Multi-cores (circa '08)
Clock cycle	250ns	25ns	1ns	0.4ns
Instructions per cycle	0.1	1	2 (4-way)	8 (quad-core, 2 threads / core)

Per-Computation Data

- Memory Bandwidth Requirement
  - **How much data is needed to keep the processor busy?**

	DEC VAX 11/780 (circa '80)	Early pipelines (circa '90)	Superscalars (circa '00)	Hyperthreaded Multi-cores (circa '08)
Clock cycle	250ns	25ns	1ns	0.4ns
Instructions per cycle	0.1	1	2 (4-way)	8 (quad-core, 2 threads / core)
Instructions per second = Cycles per second * Instructions per cycle				
Instructions per second	$4 * 10^5$	$40 * 10^6$	$2 * 10^9$	$20 * 10^{10}$

Per-Computation Data

- Memory Bandwidth Requirement
  - **How much data is needed to keep the processor busy?**

	DEC VAX 11/780 (circa '80)	Early pipelines (circa '90)	Superscalars (circa '00)	Hyperthreaded Multi-cores (circa '08)
Instructions per second	$4 * 10^5$	$40 * 10^6$	$2 * 10^9$	$20 * 10^{10}$
Instruction size	3.8B	4B	4B	4B
Operands in memory per instruction	1.8 *4B	0.3*4B	0.25*4B	0.25*4B

Per-Computation Data

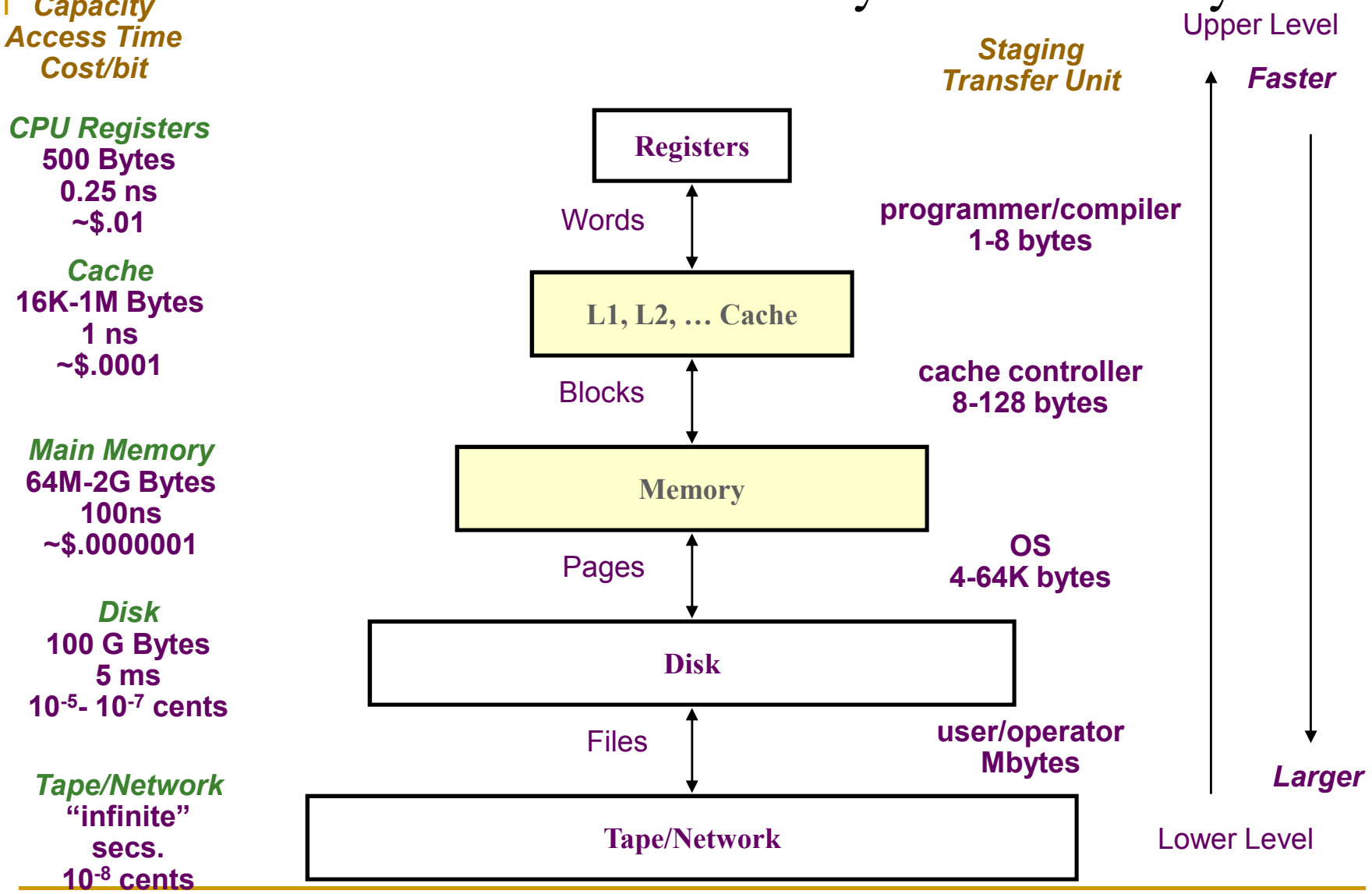
- Memory Bandwidth Requirement
  - **How much data is needed to keep the processor busy?**

	DEC VAX 11/780 (circa '80)	Early pipelines (circa '90)	Superscalars (circa '00)	Hyperthreaded Multi-cores (circa '08)
Instructions per second	$4 * 10^5$	$40 * 10^6$	$2 * 10^9$	$20 * 10^{10}$
Instruction size	3.8B	4B	4B	4B
Operands in memory per instruction	1.8 *4B	0.3*4B	0.25*4B	0.25*4B
BW Demand = Instructions-per-second * (Instruction Size + Operand-size)				
BW Demand	4.4 MBps	208 MBps	10 GBps	100 GBps

Memory Organisation

- Memory Hierarchy
  - How do we meet memory BW requirements?
  - Multiple Levels
    - early days - register set, primary, secondary, and archival
    - present day - register set, L1 cache, L2 cache, DRAM, direct attached storage, networked storage, and archival storage
  - Motivation:
    - Amortization of cost:
      - As we move down the hierarchy cost decreases and speed decreases.
        - Cost decrease implies increased capacity.

Levels of the Memory Hierarchy



Comparison Chart

Level	1	2	3	4
Name	Register	Cache	Main memory	Disk storage
Typical size	< 1 KB	< 16 MB	< 16 GB	> 100 GB
Implementation technology	Custom memory with multiple ports, CMOS	On-chip or off-chip CMOS SRAM	CMOS DRAM	Magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	50,00,000
Bandwidth (MB/sec)	20,000 – 1,00,000	5,000 – 10,000	1,000 – 5,000	20 – 150
Managed by	Compiler	Hardware	Operating system	Operating system / operator
Backed by	Cache	Main memory	Disk	CD or Type

Memory Hierarchy

- Multi-Level Inclusion Principle
  - All the data in level h is included in level h+1
- Reasons?
  - Level h+1 is typically more persistent than level h.
  - Level h+1 is order(s) of magnitude larger.
  - When level h data has to be replaced (Why?),
    - only written data needs to be copied.
    - Why is this good savings?
      - typically 15% of all memory activity is made of “write”s i.e. about 85% of data can be discarded

Memory Hierarchy

- Performance Issue:
  - What is the amortized access time?
- For a 2-level hierarchy
  - Effective Access time at level h

$T_{eff} = (1-P_{miss}) * T_h + P_{miss} * T_{h+1}$

$T_{eff} = T_h + P_{miss} (T_{h+1} - T_h)$

where P_{miss} is “probability of not finding the data in level h”

Memory Hierarchy

- Memory efficiency (M.E)
  - Highest if we can access the hierarchy as fast as the fastest level
    - i.e. at level h in a hierarchy with levels h, h+1, ...
- Define $M.E = 100 * (T_h / T_{eff})$

  $M.E = 100 / (1 + P_{miss} (R-1))$
  - Where $R = T_{h+1} / T_h$
  - M.E. = 100 when R=1 or Pmiss=0
    - When will R be 1?
  - Consider
    - R=10 (CPU/SRAM),
    - R= 50(CPU/DRAM)
    - R= 10000 (CPU/disk)
    - Calculate the Pmiss for good M.E (say 95) for each of these.

Memory Technologies – Computational

- Cache between CPU registers and main memory
  - Static RAM (6 transistors per cell)
  - Typical Access Time ~10ns
- Main Memory
  - Dynamic RAM (1 transistor + 1 capacitor)
  - Capacitive leakage results in loss of data
    - Needs to be refreshed periodically – hence the term “dynamic”
  - Typical Access Time ~50ns
  - Typical Refresh Cycle ~100ms.

Memory Technologies - Persistent

- Hard Disks
  - Used for persistent online storage
  - Typical access time: 10 to 15ms
  - Semi-random or semi-sequential access:
    - Access in blocks – typically – of 512 bytes.
  - Cost Per GByte – Rs 5
- Flash Devices (Solid State Drive)
  - Electrically Erasable Programmable ROM
  - Used for persistent online storage
  - Limit on Erases – currently 100,000 to 500,000
  - Read Access Time: 50ns
  - Write Access Time: 10 micro seconds
  - Semi-random or semi-sequential write:
    - Blocks – say 512 bits.
  - Cost Ber GByte – U.S. $5.00 (circa 2007)

Memory Technologies - Archival

- Magnetic Tapes
  - Access Time – (Initial) 10 sec.; 60Mbps data transfer
  - Cost per Gbyte – about 10c to 20c
  - Density – upto 6.67 billion bits per square inch