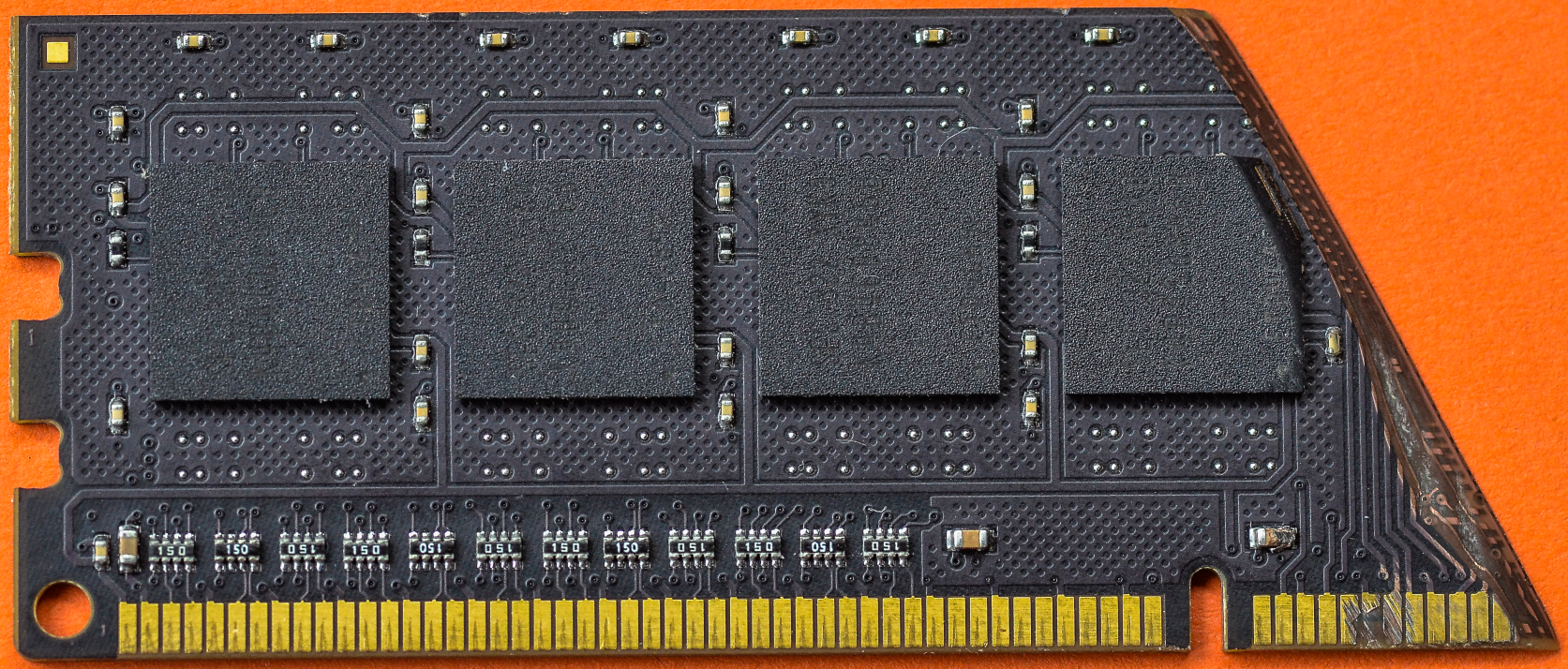
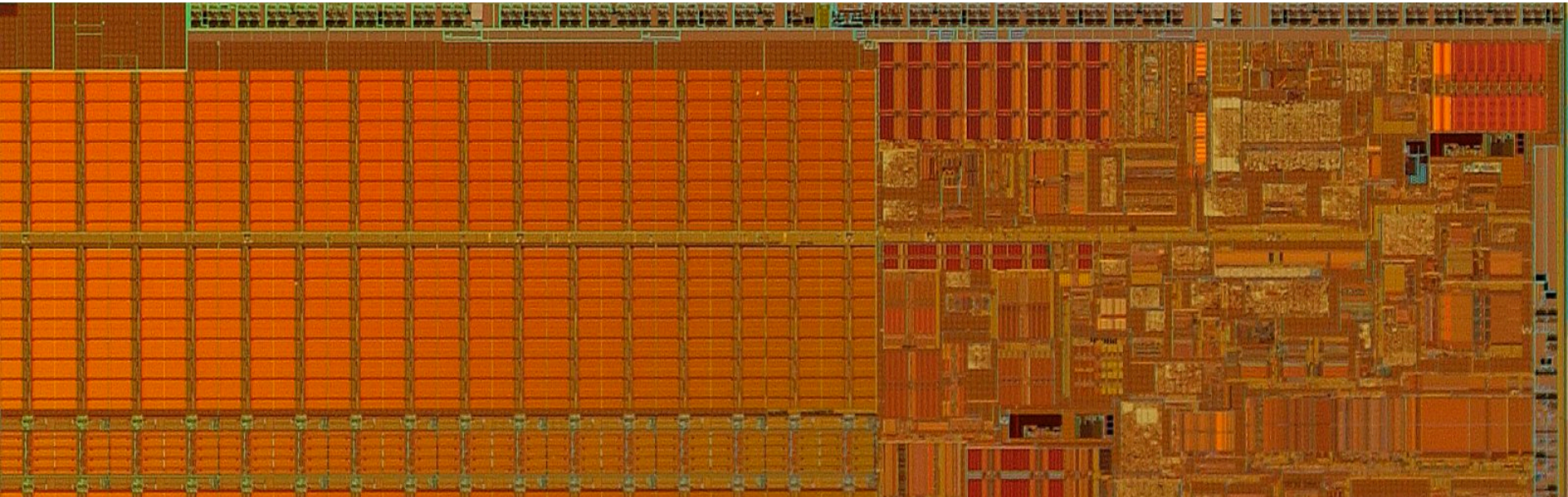
Memory



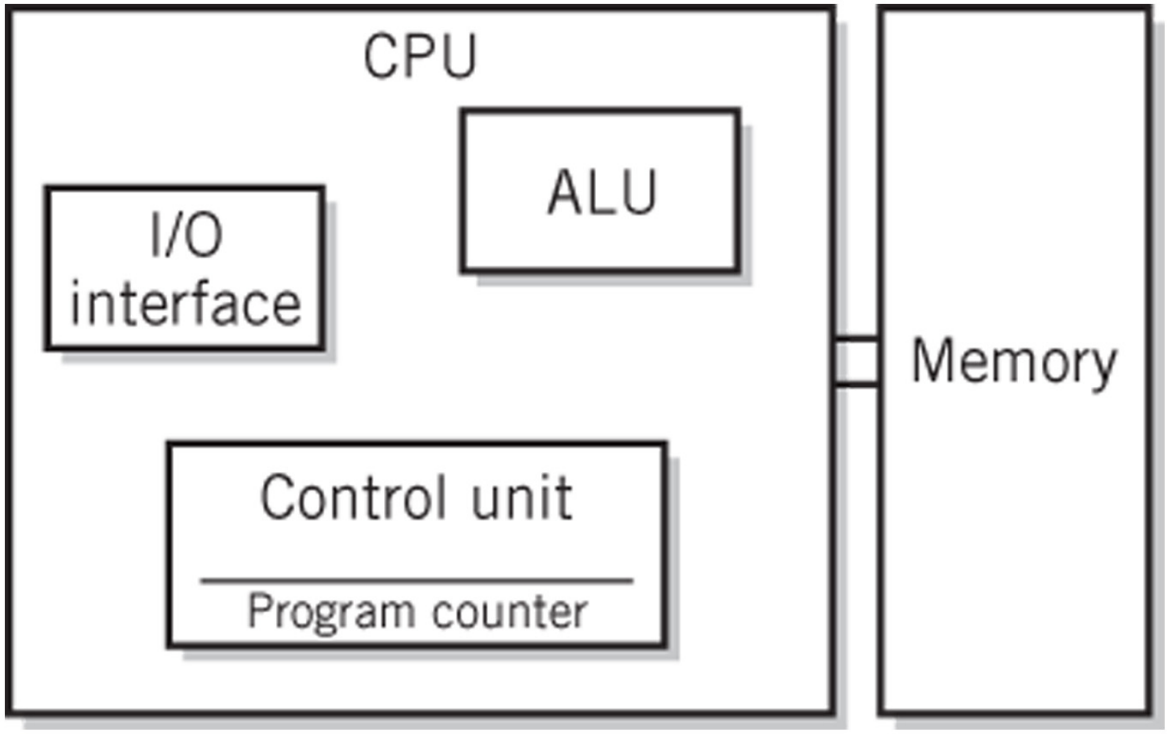
Memory

On completion of this lecture you will be able to:

- Explain the origins and consequences of the memory performance gap
- Explain the different levels in the **memory hierarchy**
- Explain the concept of **locality** as it pertains to memory access
- Quantify the impact of **cache hits and misses**



CPU & Primary Memory



Quantities

1 bit (b)		
1 nibble	= 4 b	= 2^2 b
1 Byte (B)	= 8 b	= 2^3 b

1 KiloByte (KB)	= 1,024 B	= 2^{10} B
1 MegaByte (MB)	= 1,048,576 B	= 2^{20} B
1 GigaByte (GB)	= 1,073,741,824	= 2^{30} B
1 TeraByte (TB)	= 1,099,511,627,776	= 2^{40} B

What comes next?

For the curious, read more about the origin of these prefixes:
https://en.wikipedia.org/wiki/International_System_of_Units

Memory Metrics

Cost

Capacity (MB)
Maximum number of bytes of data that can be stored.

Access time (ns).
Delay between request and first data.

Memory bandwidth (Gb/s).
Sustained data transfer rate between memory and CPU.

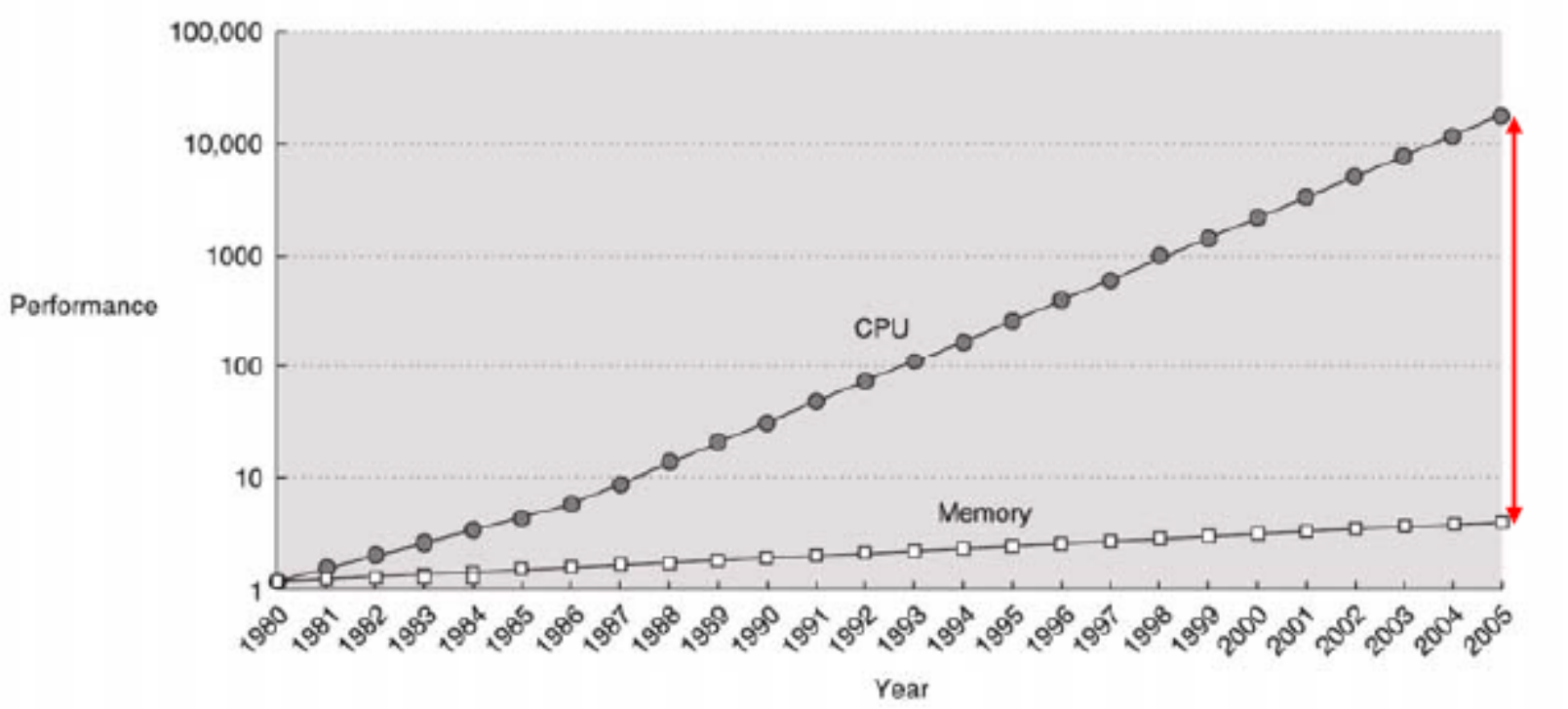
- Width of interface (bits)
- Frequency of interface (MHz)
- Number of channels

Volatile or non-volatile: Data lost when memory is not powered

Memory performance gap

CPU speeds increase 25%-30% per year

DRAM speeds increase 2%-11% per year
https://en.wikipedia.org/wiki/Dynamic_random-access_memory

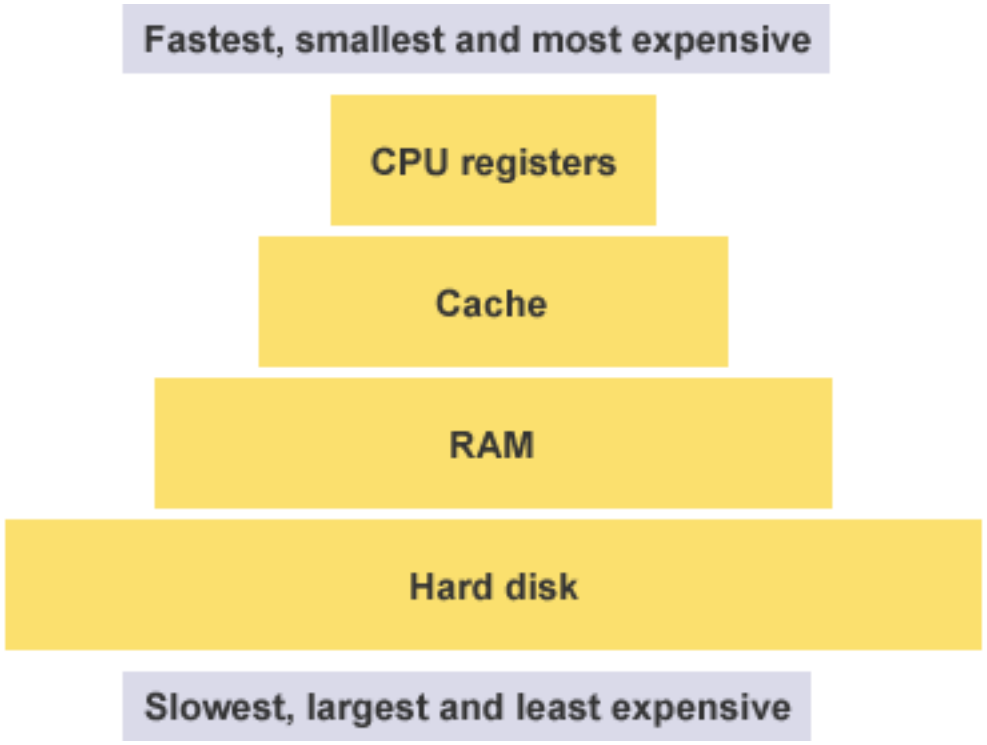


Memory Hierarchy

Ideally one would desire an indefinitely large memory capacity such that any particular … word would be immediately available. … We are … forced to recognize the possibility of constructing a hierarchy of memories, each of which has greater capacity than the preceding but which is less quickly accessible.

A. W. Burks, H. H. Goldstine, and J. von Neumann
Preliminary Discussion of the Logical Design of an Electronic Computing Instrument (1946)

4 types of primary memory



Typical Access Times

0.25 nsec

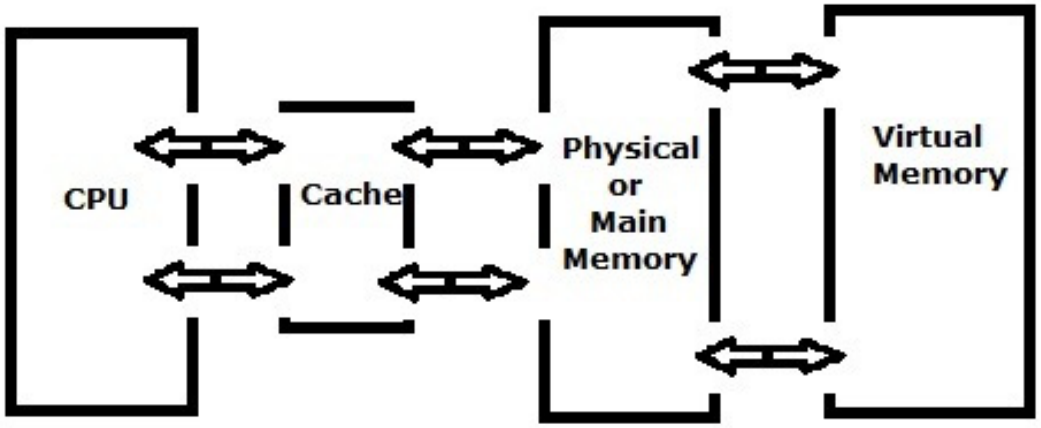
1-10 nsec

10-20 nsec

3-15 msec

Increasing access time

Aside: Virtual Memory



OS X and iOS include a fully-integrated virtual memory system.

Up to 4 GB of addressable space per 32-bit process.

OS X provides approximately 18 exabytes of addressable space for 64-bit processes.

Single process rarely gets this much RAM

- Even when it is available

Backing store (AKA Swap):

System ***pretends*** to a process that 4GB (or 18EB) is available

Data not in current use stored on disk (backing store)

As this data is needed it is swapped in to RAM.

This really belongs in the OS module but we need to mention it here.

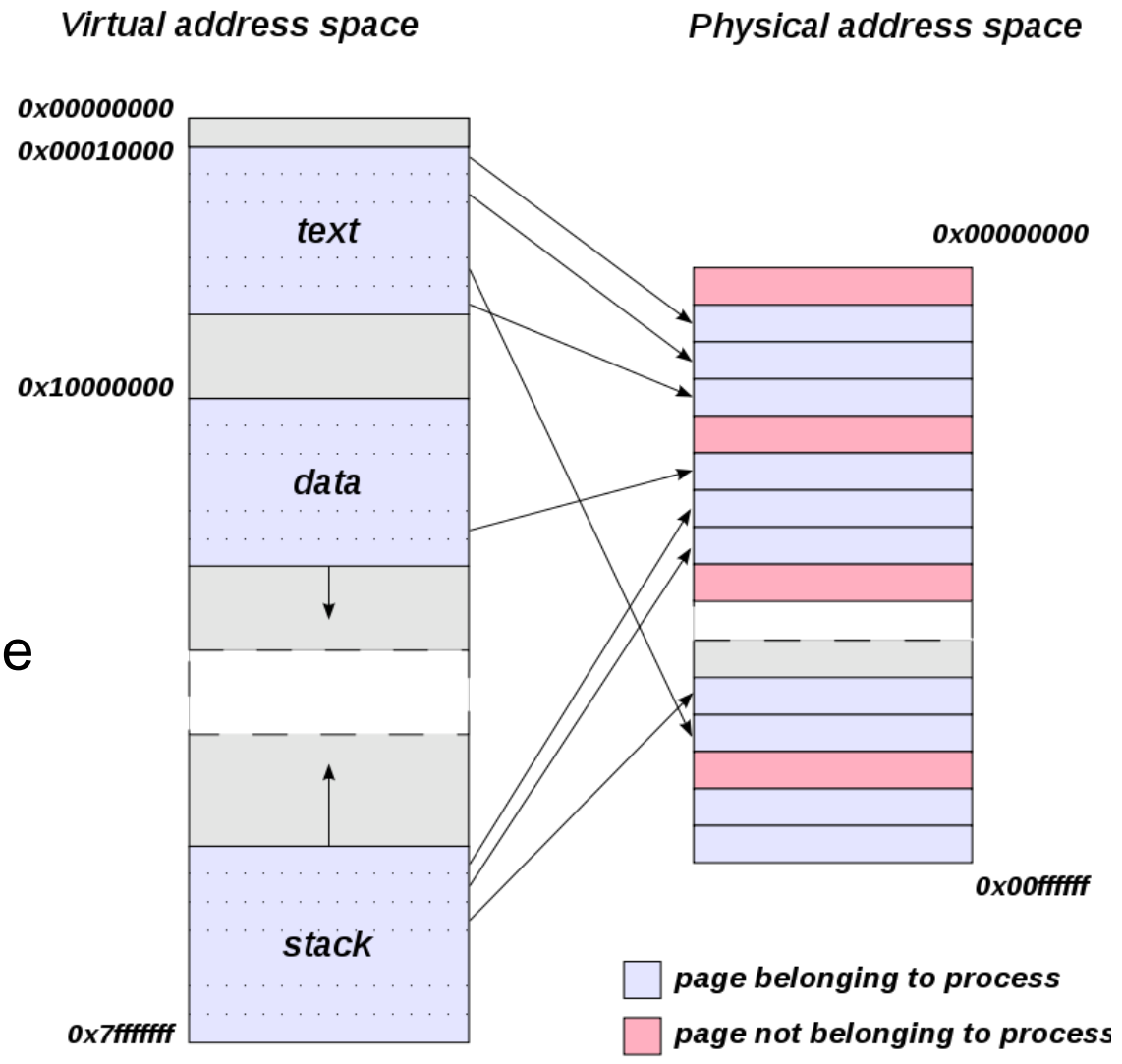
Aside: Virtual Memory

Text
- program code

Data

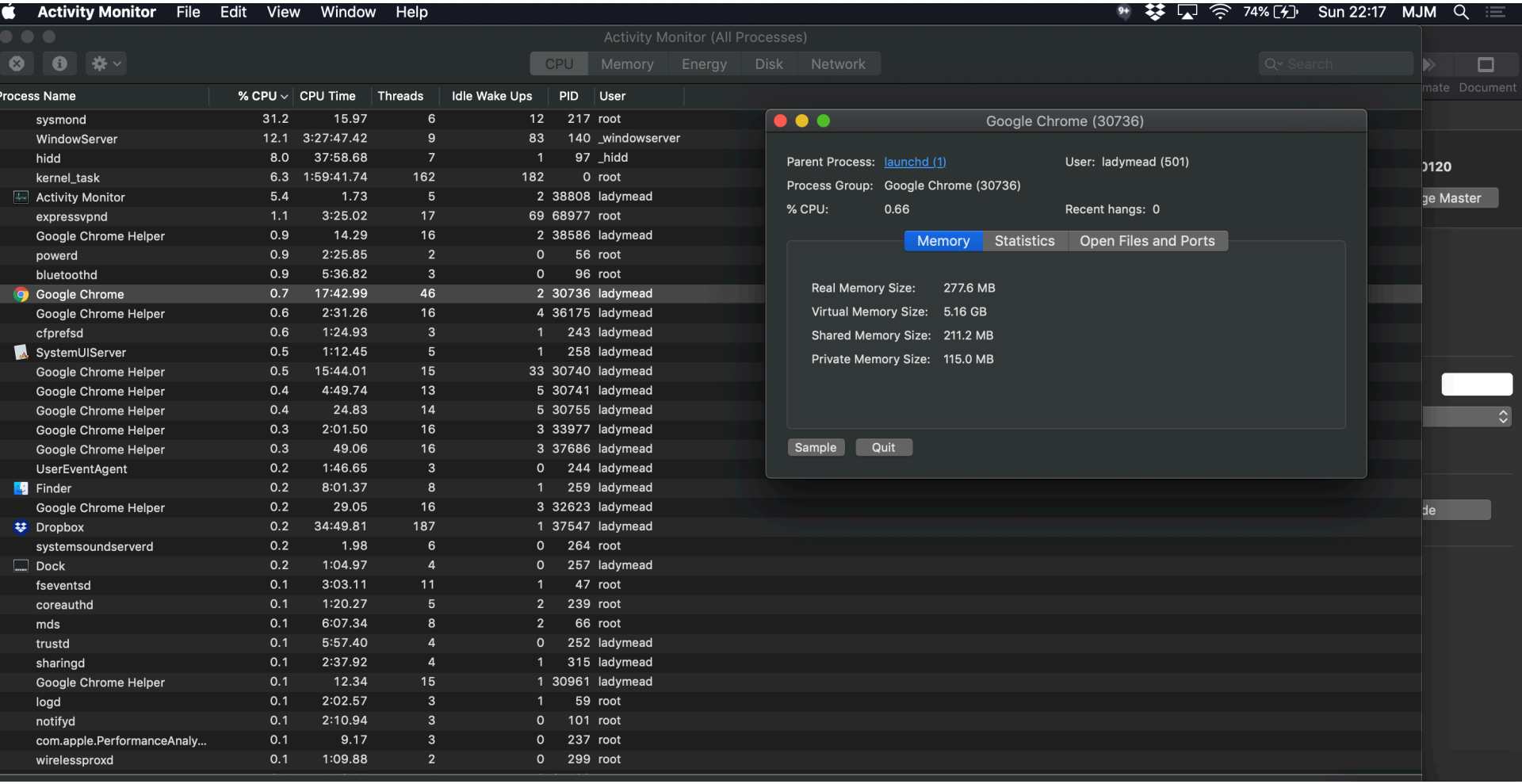
Stack
- remembering state
- returning from function calls

Not all address space in use
- need not be in RAM

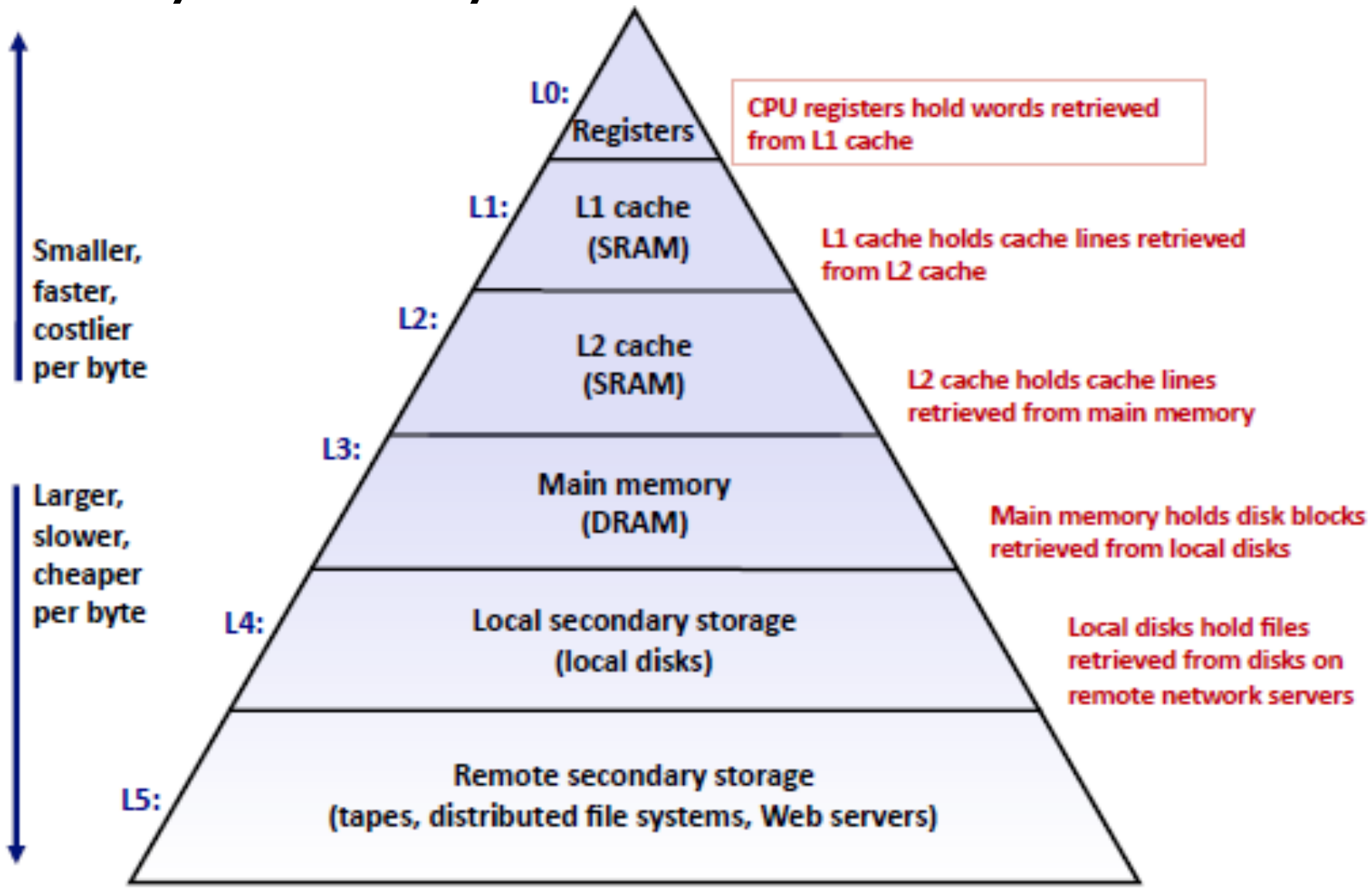


MMU (Memory Management Unit)

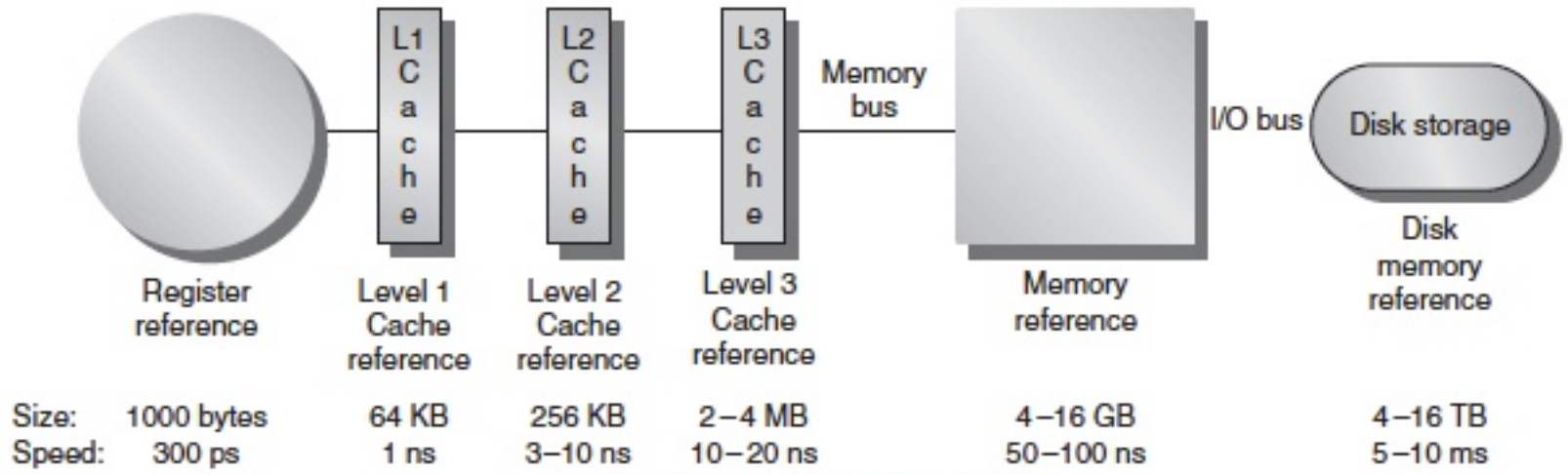
Mac OS example



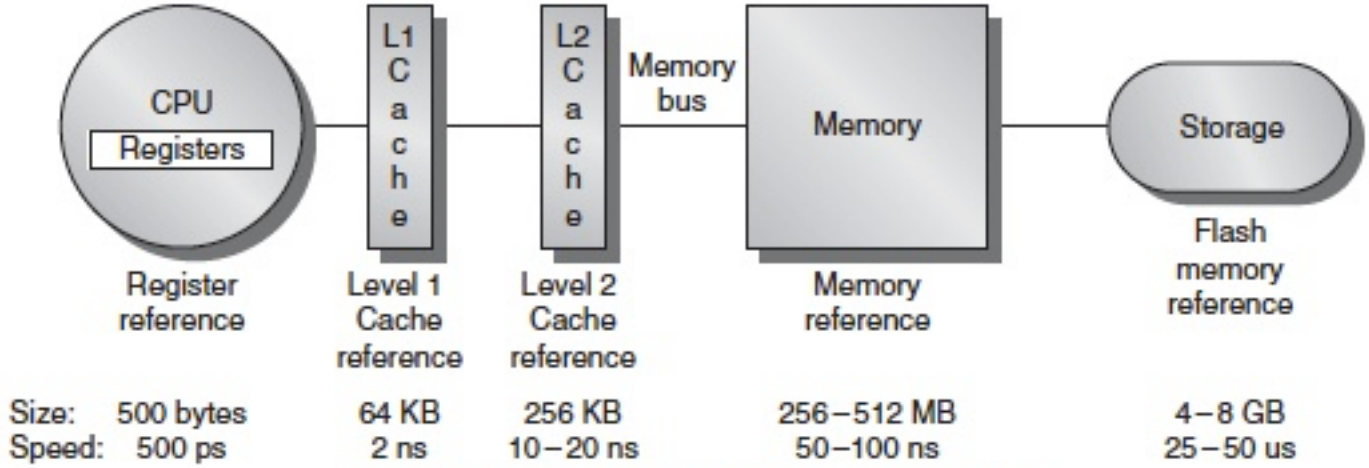
Memory Hierarchy



Memory Hierarchy



(a) Memory hierarchy for server



(b) Memory hierarchy for a personal mobile device

Computer Architectures: A Quantitative Approach, Hennessy and Patterson

Examples

Intel® Core™ i7-5600U Processor – 2 cores
- Level 1 cache: 2 x 32 KB instruction caches, 2 x 32 KB data caches
- Level 2 cache: 2 x 256 KB caches
- Level 3 cache: 4 MB shared cache

Apple A11 (iphone 8 and X) - 6 cores
- L1 - 32 KB data and 32 KB instructions per-core
- L2 – 8 MB shared
- L3 – none

https://en.wikipedia.org/wiki/Apple_A11

Memory Hierarchy

To access a particular piece of data, the CPU first sends a request to its nearest memory.

If the data is not in that memory, then the next level in the memory hierarchy is queried.

If the data is not in main memory, then the request goes to disk.

Once the data is located, then the data and a number of its nearby data elements (a Block) are fetched into cache memory.

Blocks of memory from RAM(Disk) are Mapped into Block areas in the Cache area.

Cache

Cache used by the CPU to reduce the average time to access data.

Smaller, faster memory which stores copies of the data from frequently used main memory locations.

Caching in general – storing a copy of data to make future access requests faster

e.g. web cache

Locality

Cache memories are effective by virtue of the Principle of Locality.

Temporal locality. Recently-accessed data elements tend to be accessed again.

Spatial locality. Accesses tend to cluster; Nearby locations tend to be accessed soon.

Sequential locality. Instructions tend to be accessed sequentially.

Locality Example

```
total = 0;
for (i = 0; I < n; i++)
   total += a[i];
return total;
```

Temporal: Variable total referenced on each iteration

Spatial: Array a[] accessed

Cache Memory Definitions (1 of 2)

A Hit is when data/instruction is found in Cache.

A Miss is when a reference to data/instruction is not in Cache.

The Hit-ratio/Hit-rate is the percentage of time data/instruction are found in Cache.

The Miss-ratio/Miss-rate is the percentage of time data/ instructions are not in Cache.

Miss-ratio = 100 – Hit-ratio

Cache Memory Definitions (2 of 2)

The Hit-time is the time required to access Cache.

The Miss-penalty is the time required to process a miss, including the time that it takes to replace a block of memory plus the time it takes to deliver the data to the processor.

Average Memory Access Time

$$\text{AMAT} = \begin{matrix}\text{time for}\\ \text{a hit}\end{matrix} + \text{miss rate} \times \text{miss penalty}$$

Calculate AMAT:

1GHz processor

time for a hit = (cache access time) 1 clock cycle

miss penalty = 20 clock cycles

miss rate = 5% (0.05)

Calculate AMAT if miss rate = 10%

What miss rate will give an AMAT of 1.5ns?

Dell Vostro 1540

Processor: Intel Core i3-370M, 2.4GHz

Memory: 3GB 1,333MHz DDR3 RAM

Graphics: Intel HD Graphics integrated

Hard disk: 320GB

Display: 15.6in 1,366x768, LED-backlit screen

Features: 1.3 megapixel webcam, microphone

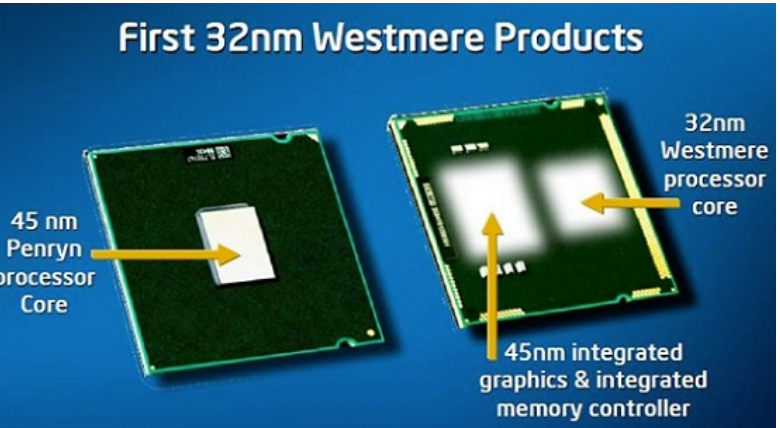
Connectivity: 802.11g/n Wi-Fi, Gigabit Ethernet, Bluetooth 3.0

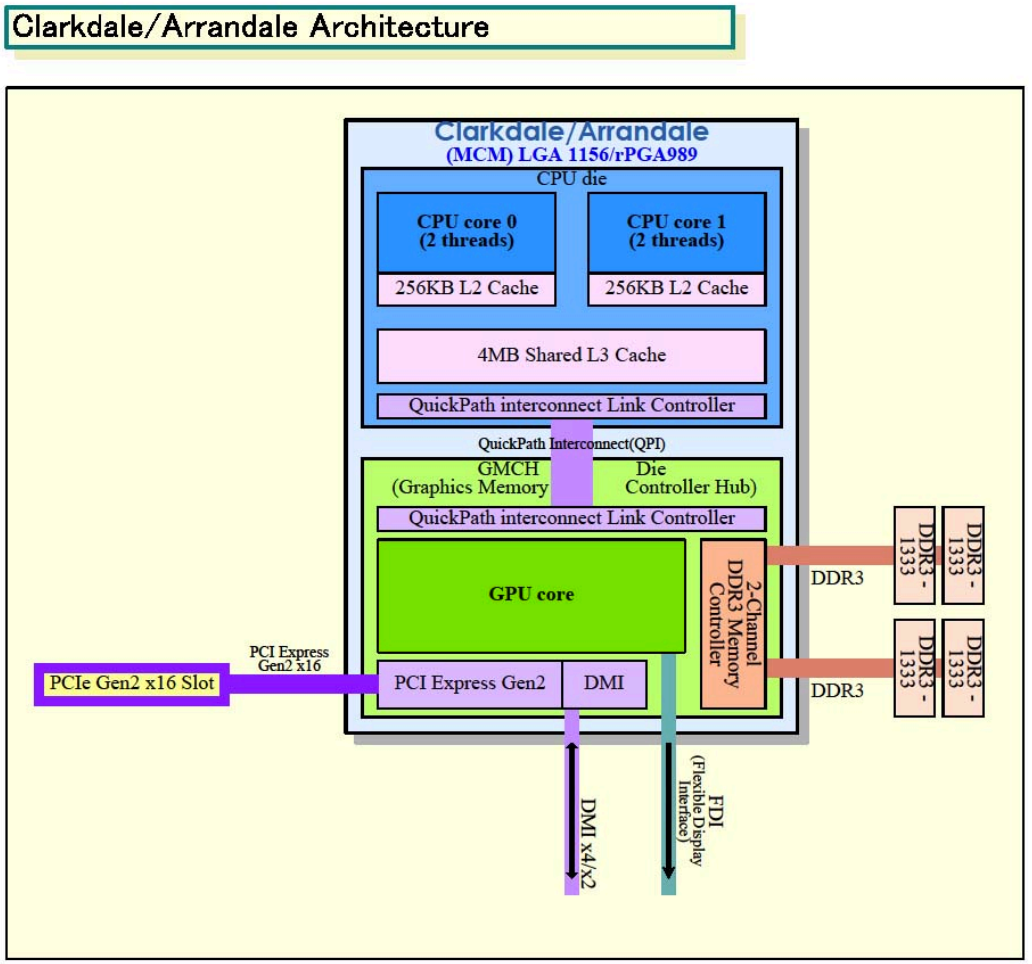
Ports: 3 x USB2, 1 x VGA, 1 x HDMI,

1x 3.5mm headphone output, 1x 3.5mm microphone input, SD/MMC/MS card reader

Dimensions: 376 x 260 x 33mm (WxDxH)

Weight: 2.4kg





Memory

On completion of this lecture you will be able to:

- Explain the origins and consequences of the memory performance gap
- Explain the different levels in the memory hierarchy
  - from registers and caches to virtual memory
- Explain the concept of **locality** as it pertains to memory access
- Quantify the impact of cache hits and misses