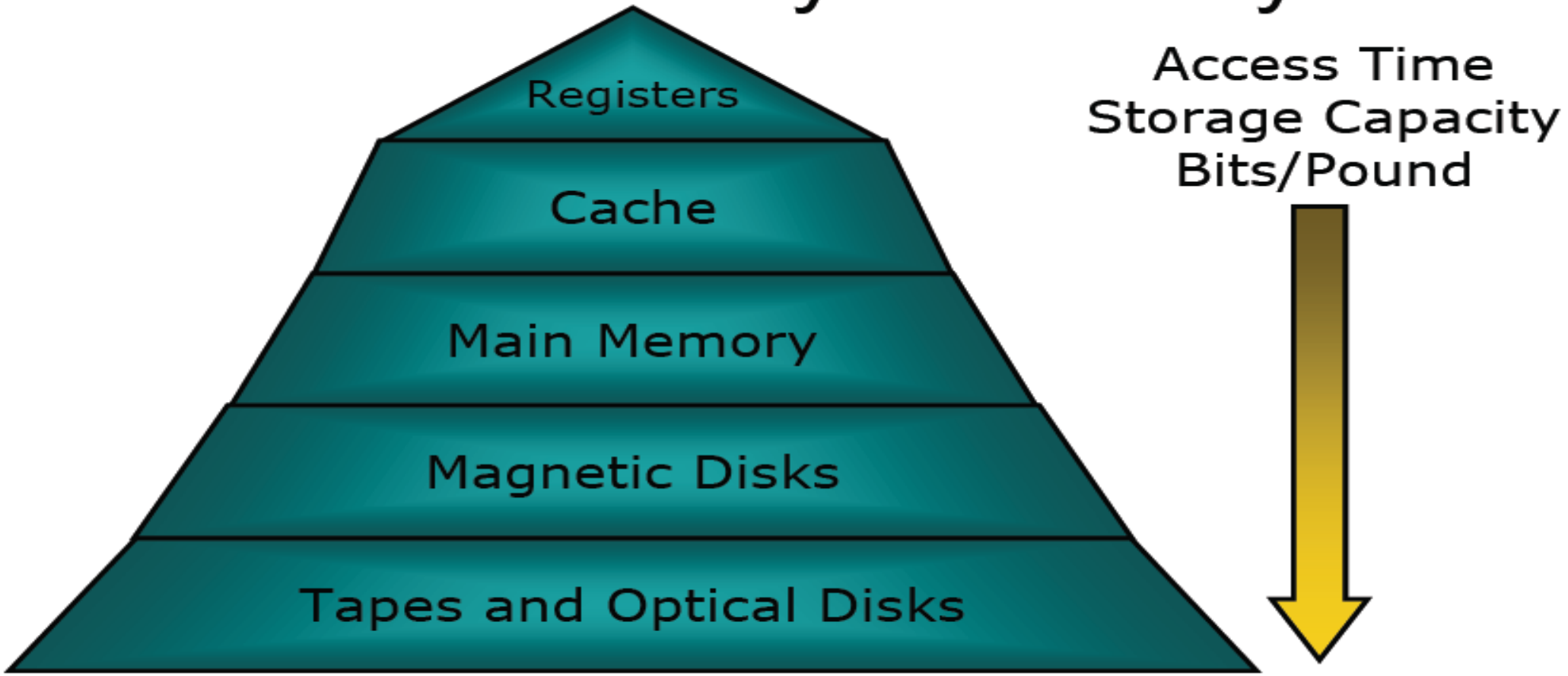
MEMORY

BY

MIKE DALEY

The Memory Hierarchy



- CPU registers, accessed at full CPU speed
- Cache, 32KB to several Mb
- Main memory, 256Mb to 10’s of Gb at high level
- Disks, main work horse for storage
- Tapes and optical disks mainly used for backup
- As we move down three parameters increase

Access Time

- Registers a few nanoseconds
- Cache a small multiple of CPU registers
- Main memory few 10s of nanoseconds
- Big gap - disk access at least 10 milliseconds
- Tape access measured in seconds if storage is offline

Nanosecond (ns)

1 Billionth of a second.

Light travels very fast! In a vaccum, light travels at:

- 186,000 miles per second
- 300,000 kilometres per second

Distance light to travels in 1ns is 186,000 feet.

Capacity

Registers ~128 bytes

Cache a few Mb

Memory 10 to 1000s of Mb

Magnetic disks 100G to 1Tb+

Tapes usually offline so limited only by budget

Comparisons

Bit: Computers deal with binary digits, or bits for short. A bit can be 0 or 1, equivalent or off or on.
Byte: One byte is eight binary digits, such as 1111001.
Kilobyte (KB): A kilobyte is 1024 bytes, a. Therefore 1KB is the same as 1024 x 8 = 8192 binary digits.
Megabyte (MB): 1024KB equals one megabyte (MB),
Gigabyte (GB): There are 1024MB in one gigabyte.
Terabyte (TB): There are 1024GB in one terabyte (TB)

Hard drive manufacturers have long eschewed this system in favour of rounding down to make things easier (and also provide less storage space).

This means that 1000 bytes = 1 kilobyte and 1000 kilobytes = 1MB. Again, 1000MB = 1GB and 1000GB = 1TB.

Windows, however, sticks to the 1024 rule, which means it sees a 250GB hard drive as 232GB, and a 1TB drive as 953GB.

This explains why hard drives appear to have a lower capacity than advertised. A 1TB hard drive has the capacity to store 1,000,000,000,000 bytes (1 Trillion). Divide this by 1024 and you get 976,562,500KB. Divide by 1024 again and you get 976,562,500MB. Finally, divide by 1024 to get gigabytes and you end up with 953,674GB.

Bits per pound

Number of bits per pound increases as we move down

Prices change but in March 2018 memory (1GByte approx 18 pounds) ~1.8p/Mb,

Magnetic disk (4TByte approx 112 pounds)

28 pounds/TB, ~0.27p/Gb

Memory Addresses

Computer memory consists of *cells*, each with a unique *address*

- A cell is the smallest addressable unit
- Each cell usually consists of 8 bits (1 Byte)

Bytes are grouped into **words**

Many instructions operate on entire words

32-bit computers have 32-bit words, made from 4 x 8-bit bytes

64-bit computers have 64-bit words made from 8 x 8-bit bytes

32-bit machine will have 32-bit registers and instructions for 32-bit words

64-bit machine will have 64-bit registers and instructions for 64-bit words

Byte Ordering

We have mentioned before.

The bytes within a word can be numbered differently

- left-to-right (*big endian*)
- right-to-left (*little endian*)

As long as we are consistent, this doesn't matter to the computer

But it can cause problems when transferring data from one computer to another

Especially with character strings

Mainframes and Sparc machines often Big endian.

Intel and PCs normally Little Endian

Byte-Ordering and Numbers

Problem: CPU Fast, Memory Slow

After a memory request, the CPU will not get the word for several cycles

Two simple solutions:

- Continue execution, but stall CPU if an instruction references the word before it has arrived (hardware)
- Require compiler to fetch words before they are needed (software)

May need to insert NOP instructions.

Very difficult to write compilers to do this effectively.

Memory speed always playing catch-up to processor speed.

Memory capacity increased rather than speed.

The Root of the Problem:

Economics

- Fast memory is possible, but to run at full speed, it needs to be located on the same chip as the CPU
- Very expensive
- Limits the size of the memory

Do we choose:

- A small amount of fast memory?
- A large amount of slow memory?

Problem increased by designers concentrating on making CPU's faster and memories bigger

Memory accessed on a bus is slow.

Limits on how big CPUs can be made.

Limits on chip memory.

The Best of Both Worlds: *Cache Memory*

Combine a small amount of fast memory (the *cache*) with a large amount of slow memory

When a word is referenced, put it and its neighbours into the cache

Programs do not access memory randomly

(**Locality Principle)**

- *Temporal Locality*: recently accessed items are likely to be used again
- *Spatial Locality*: the next access is likely to be near the last one

Early microprocessors had no cache;

INMOS transputer was the 1st processor with on-chip cache;

Acorn's ARM (Acorn RISC Machine) was 1st processor to have dual level cache.

Pentium II had:

- 32KB L1 cache (16K for instructions; 16K for data);
- 512KB L2 cache (not on processor chip, but on separate chip within SEC cartridge, connected via dedicated bus).

Example stored program in memory, instructions likely to be sequential and in consecutive memory locations (spatial)

Most execution time spent in loops where the same instructions are executed over and over (temporal)

CACHE LEVELS

Sometimes systems are built with more than one cache.

The cache closest to the CPU is called level 1 cache, the next is level 2 etc.

Level 1 cache will be faster than level 2, but smaller in capacity.

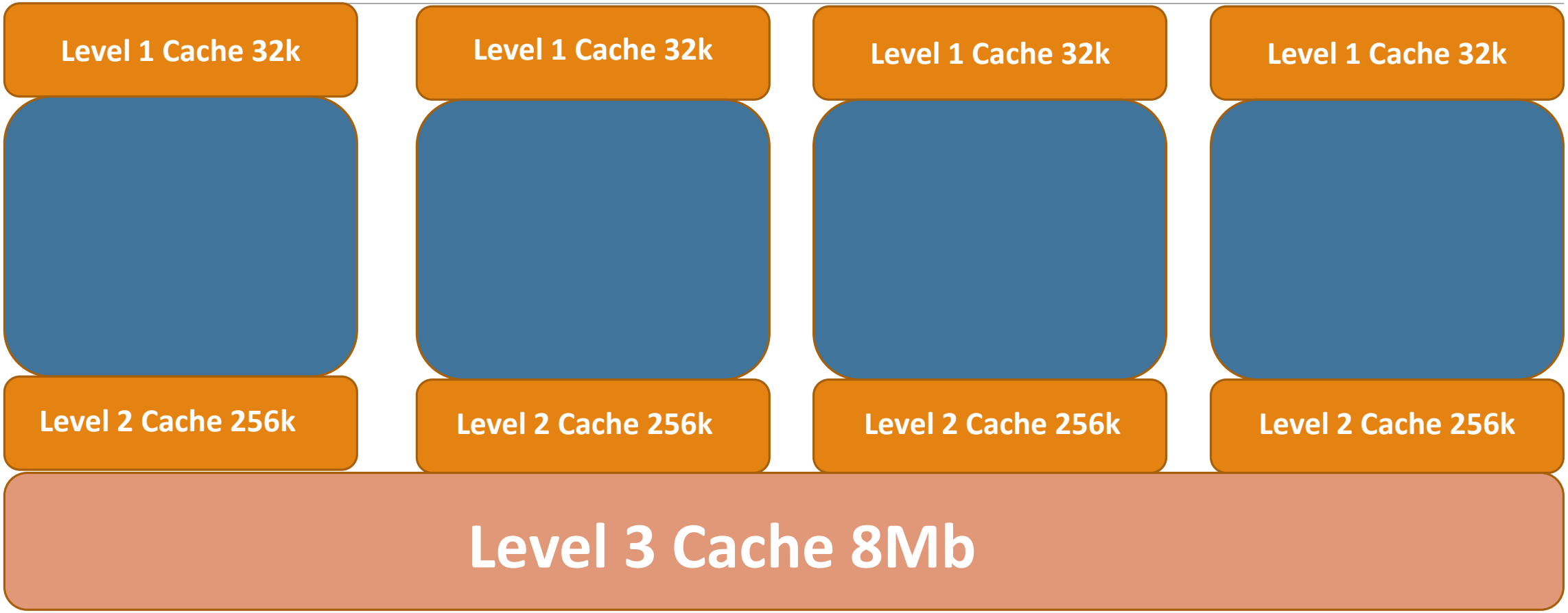
The CPU will always look for data in level 1 first. If it gets a cache miss, it looks in level 2 if it also gets a miss it goes to main memory.

On currently available desktop processors, i5 CPUs have 3MB to 6MB of L3 cache, while i7 processors have 8MB to 15MB L3 cache.

i7 first gen to 3rd gen all have 256kb L2 cache (11 clocks latency) per core.

32 KB data + 32 KB instruction L1 cache (4 clocks latency) per core

i7


Level 1 Cache 32k
Level 1 Cache 32k
Level 1 Cache 32k
Level 1 Cache 32k
Level 2 Cache 256k
Level 2 Cache 256k
Level 2 Cache 256k
Level 2 Cache 256k
Level 3 Cache 8Mb

Intel i9 (Released Q2 2017)

INTEL CORE I9 7900X (£800)

10 cores

32K L1 cache

1Mb L2 cache

14Mb L3 cache

INTEL CORE I9 7980X (£1,925)

18 cores

32K l1 CACHE

1Mb L2 cache

25Mb L3 cache