Computer Architecture
The Memory Heirarchy

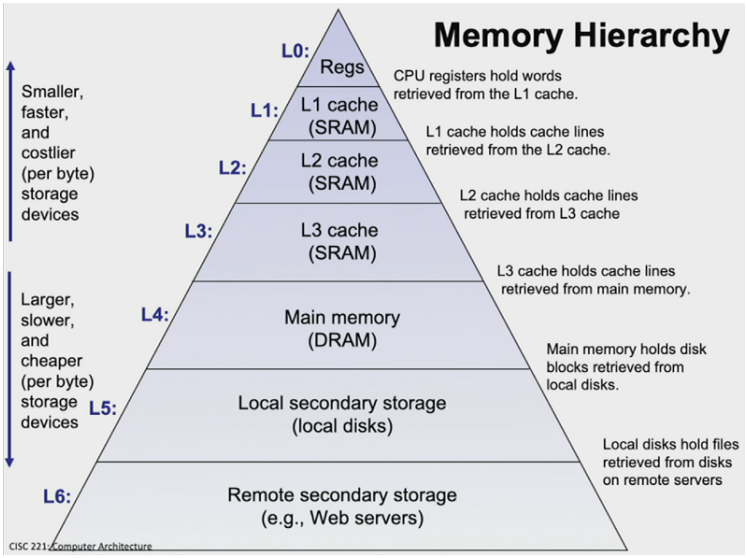
Cain Susko

Queen's University
School of Computing

March 22, 2022

The Hierarchy

different classes of memory are segregated by their speed of access, their relative speeds can be visualized on a graph like so:

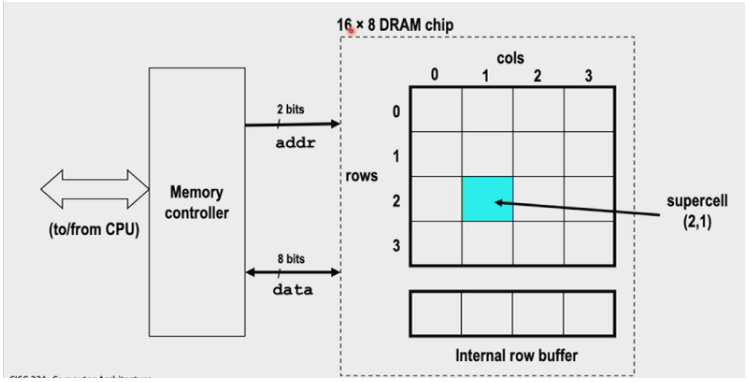


Faster Memory classes are more expensive in resources, and thus are mostly used in smaller quantities. (in addition to them being physically smaller than the lower classes of chips)

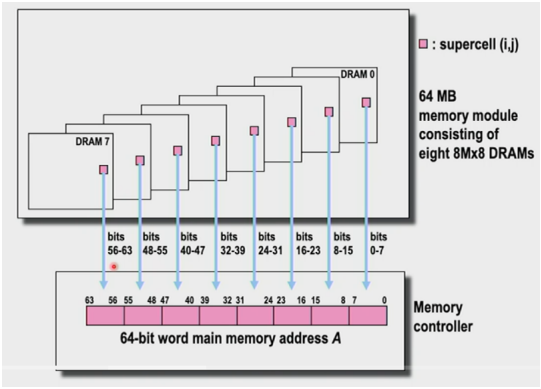
Random-Access Memory

RAM are traditionally chips who's basic storage unit was a cell. Multiple cells of the word form a super cell, which id id'd by a single address. Many RAM chips then make up the Computers memory. This is in contrast to Sequential Access Memory which can only be accessed by iterating a pointer along an array.

RAM Organization the chip is normally organized with a 2D array like so; where the array has the dimensions of $d \times w$, where dw are the total bits organized as supercells of size w bits.



There are 4 bits for the access address because there is needed 2 bits for the row address and 2 bits for the column address. When a row is addressed, it is then saved into the RAM chip's internal row buffer. Then, then the column is addresses, the specified column within the row is taken out and returned. When writing this is the exact same thing but backwards. This is very fast as the row can be copied with parallel circuits. This Process can also be scaled up with many (8) 8×8 RAM chips. if we had 64 bit word across this **memory module**, we would copy into the memory controller buffer.



SRAM v. DRAM there are 2 types of RAM, SRAM and DRAM. this first is Static RAM while the second is Dynamic RAM. Static RAM's are

fast and costly while Dynamic RAM's are slow and less costly, with only 1 transistor.

	Trans. per bit	Access time	Needs refresh?	Needs EDC?	Cost	Applications
SRAM	4 or 6	1X	No	Maybe	100x	Cache memories
DRAM	1	10X	Yes	Yes	1X	Main memories, frame buffers

All of this memory is what is known as Volatile Memory, which means that if the power were to be fully shut off on the computer, the data stored in these chips would be lost, as, they require power to retain information. This requires what is known as **non-volatile memory**

Non-Volatile Memory

There are a few types of memory that are non-volatile, meaning they do not lose their data once power is cut to them.

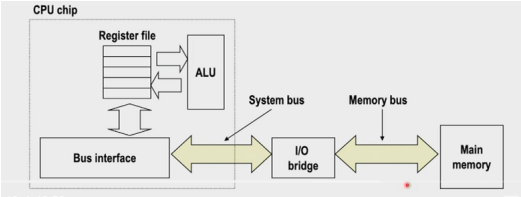
- Read-Only Memory (ROM): normally programmed during production of chip
- Programmable ROM (PROM): can be programmed once
- Erasable PROM (EPROM): can be bulk erased (commonly with UV or x-rays)
- Electric EPROM (EEPROM): electronically erasable EPROM
- Flash Memory: EEPROM(s) with partial (block level) erase capability. These wear out after around 100,000 erasures.

These are used for Firmware programs stored in a ROM (BIOS, Controllers for Disks , network cards, graphics accelerators, security subsystems, ...) or also Solid State Disks.

Traditional Bus Structure

A bus is a collection of parallel wires that carry addresses, data, and control signals. A bus is shared between multiple devices, connecting them. The

most important use of a bus is connecting the CPU to Memory in order to read and write data for executing programs.



Reading When the CPU is **reading** memory the transaction goes as follows:

1. The CPU places an address A onto the memory bus -

2. The Main memory reads A from the memory bi=us, and retrieves the word x at address A, and places x on the bus

3. Lastly, the CPU reads the word x from the bus and copies it into register `%rax`

Note: the assembly operation for this would be: `movq A, %rax`

Writing When a CPU is **writing** to memory, the operation goes as follows:

1. The CPU places the address A on a bus, the Main memory then reads it and waits for the corresponding data word to arrive.

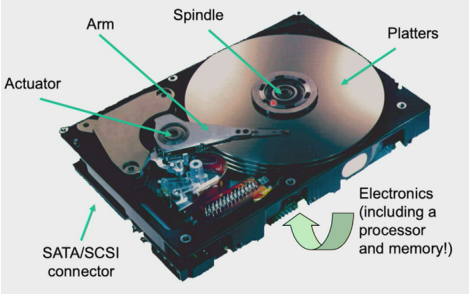
2. The CPU then places the data word y on the bus

3. Finally, the Main memory reads the data word y from the bus and stores it at address A

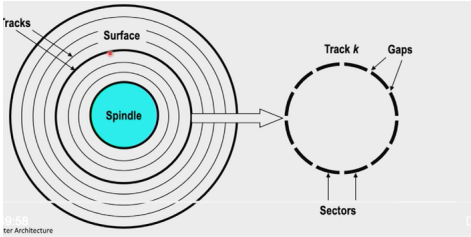
Note: the assembly operation for this would be: `movq %rax, A`

Hard Disk Drive

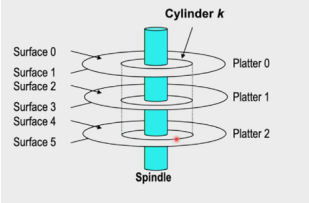
If one were to open up a disk drive in their computer, it would look like this: The spindle, in combination with the rotating disk, finds parts (or sectors)



of memory. This can be seen like so



Many sectors make a Track, and many tracks make a cylinder, which can be visualized as many disks stacked on top of each other:

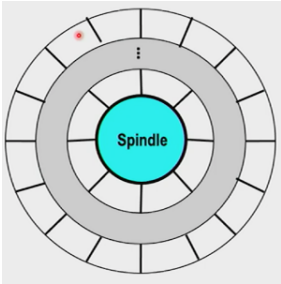


Disk Memory The memory that can be stored in a disk can be using:

- track density (bits/in)
- recording density (tracks/in)

- areal density (bits/in^2)

Note: it should be considered that because the disk is well, a disk, the diameter gets smaller as the the radius gets smaller, meaning less data can be stored closer to the middle of the disk. Therefore, disks are partitioned as such:



Therefore, the calculation for the capacity for a disk would be:

$$C = (bytes/sector) \times (avg.sectors/track) \times (tracks/surface) \times (surfaces/platter) \times (platters/disk)$$

Disk Operations Disk Reading and writing are carried out by the rotating disk and arm. when there are multiple disks, there are multiple arms that all work in unison, and all the disks are on one rotating spindle.