



IGNITED MINDS
Journals

*International Journal of
Information Technology
and Management*

*Vol. VI, Issue No. II,
May-2014, ISSN 2249-4510*

**A CRITICAL STUDY ON COMPUTER MEMORY
AND ITS MANAGEMENT**

AN
INTERNATIONALLY
INDEXED PEER
REVIEWED &
REFEREED JOURNAL

A Critical Study on Computer Memory and Its Management

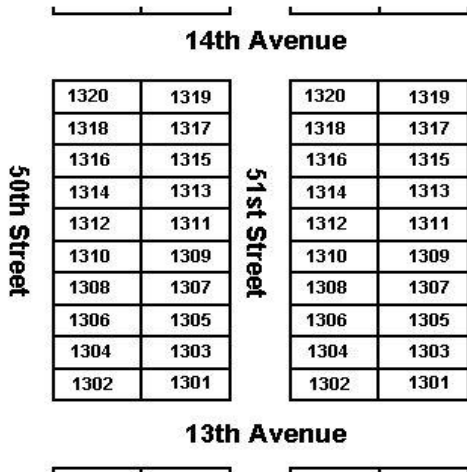
Rishi Mathur¹ Dr. D. D. Aggarawal²

¹Research Scholar, Bundelkhand University, Jhansi (U.P.)

Abstract – Memory of a computer is organized into a regular pattern of containers for information. These containers for information are called "words". Each one has a numeric address and each one is the same size as each of the others. For most applications, it is inconvenient to refer to portions of memory by their numeric addresses, so programming languages allow us to allocate portions of memory by name. When we store information in the memory of a computer we need to decide on how much we need for various purposes and on how it will be organized. Programming languages provide mechanism for "types" of information in memory. They also provide mechanisms to identify repetitive arrays of items of the same type and to aggregate possibly heterogeneous types under a common name.

INTRODUCTION

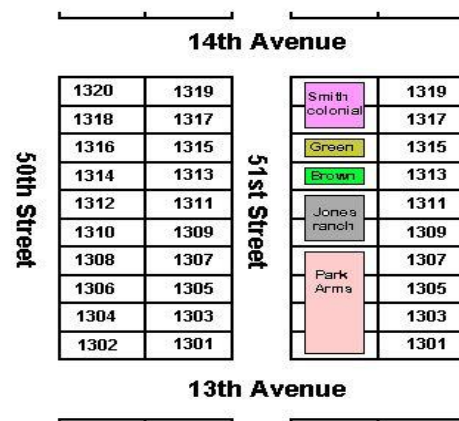
Conceptually the memory in a computer is very much like building lots laid out by lot number on a map of the streets of a city before any houses have been built. Each lot on the map has a sequential number along some street with the streets themselves numbered sequentially. Each lot is the same size, say 20' by 100'.



Now suppose people buy some of the lots and build houses. Suppose a developer takes the first 4 lots (1302-1308 51st Street) and builds an apartment building, calling it the "Park Arms". Suppose Mr. Jones and Ms. Smith each decide to build big houses, a ranch and a colonial, respectively, which span two lots each (1310-1312 and 1318-1320), and that Mr. Green and Mrs. Brown each decide to build a similar modest

one-lot houses (1314 and 1316). The lots still have their numbers, so that the post-office can deliver mail, but people know the buildings by the appropriate names, "The Park Arms", the "Jones" ranch, the "Brown" house, the "Green" house and the "Smith" colonial.

In working with the memory of a computer, we do very much the same thing, we assemble the words that we need to hold the information we are working with into aggregates of the appropriate "type". Sometimes we need several words, sometimes just a couple, and sometimes just one. Different styles of houses, such as colonials and ranches, may use the space of a building lot differently. In a computer, we may decide to use portions of the spaces provided by the words differently. In a computer, the combination of the size of the space we reserve and the way in which we use that space defines a "type" of storage.



Just as we make the distinction between two-lot colonial houses as a type of house and the particular two-lot colonial that Ms. Smith built in our hypothetical city, we distinguish the abstract types of storage allocation in a computer from the particular layouts of data we establish using those types. A type in a computer language is like a blueprint for building houses.

It may be convenient to refer to buildings by names that help us remember who lives there. In using computer memory, can be helpful to refer to allocated portions of computer memory by names that help us remember the uses to which we are putting those portions of storage. In computing we call such names "variable" names.

A house has rooms into which we may cram only a certain amount of furniture. The words in a computer have a certain capacity, given in "bits", "nibbles", "characters", "bytes", "octets" "half-words" and other terms intended to convey a sense of smaller or larger portions of a word.

REVIEW LITERATURE

Historically, early computers used delay lines, Williams’s tubes or rotating magnetic drums as primary storage. By 1954, those unreliable methods were mostly replaced by magnetic core memory. Core memory remained dominant until the 1970s, when advances in integrated circuit technology allowed semiconductor memory to become economically competitive.

This led to modern random-access memory (RAM). It is small-sized, light, but quite expensive at the same time. (The particular types of RAM used for primary storage are also volatile, i.e. they lose the information when not powered).

As shown in the diagram, traditionally there are two more sub-layers of the primary storage, besides main large-capacity RAM:

- Processor registers are located inside the processor. Each register typically holds a word of data (often 32 or 64 bits). CPU instructions instruct the arithmetic and logic unit to perform various calculations or other operations on this data (or with the help of it). Registers are the fastest of all forms of computer data storage.
- Processor cache is an intermediate stage between ultra-fast registers and much slower main memory. It was introduced solely to improve the performance of computers. Most actively used information in the main memory is just duplicated in the cache memory, which is faster, but of much lesser capacity. On the other hand, main memory is much slower, but has a much greater storage capacity than processor registers. Multi-level hierarchical

cache setup is also commonly used—*primary cache* being smallest, fastest and located inside the processor; *secondary cache* being somewhat larger and slower.

Main memory is directly or indirectly connected to the central processing unit via a *memory bus*. It is actually two buses (not on the diagram): an address bus and a data bus. The CPU firstly sends a number through an address bus, a number called memory address that indicates the desired location of data. Then it reads or writes the data in the memory cells using the data bus. Additionally, a memory management unit (MMU) is a small device between CPU and RAM recalculating the actual memory address, for example to provide an abstraction of virtual memory or other tasks.

As the RAM types used for primary storage are volatile (cleared at start up), a computer containing only such storage would not have a source to read instructions from, in order to start the computer. Hence, non-volatile primary storage containing a small startup program (BIOS) is used to bootstrap the computer, that is, to read a larger program from non-volatile *secondary* storage to RAM and start to execute it. A non-volatile technology used for this purpose is called ROM, for read-only memory (the terminology may be somewhat confusing as most ROM types are also capable of *random access*).

Many types of "ROM" are not literally *read only*, as updates are possible; however it is slow and memory must be erased in large portions before it can be re-written. Some embedded systems run programs directly from ROM (or similar), because such programs are rarely changed. Standard computers do not store non-rudimentary programs in ROM, and rather, use large capacities of secondary storage, which is non-volatile as well and not as costly.

MEMORY STORAGE

On modern computers, all the units of storage are effectively given in terms of "bits". A **bit** is a portion of memory which can hold either a "0" or a "1". The other terms mentioned above gone through historic changes and are likely to change again. Different computer manufacturers favored different numbers of bits in, say, a **byte**. At present, for the most popular desk-top computers and workstations, there is considerable agreement on the following sizes:

Portion of word	Number of bits
nibble	4
character	8 or 16
byte	8
octet	8

half-word	16
word	32

The ambiguity on the size of characters is because we are in a time of transition to richer and more complex character sets for which the limit of 256 characters imposed by an 8-bit character size is inappropriate. We may also be starting on a transition to 64-bit words. Time will tell.

Now let us return to our analogy to street maps. The apartment house, the "Park Arms", has one address (1302-51st Street), which refers to many apartments, each of which has an address (an apartment number) within the apartment house. Similar concepts are available in organizing computer memory. An "array" is a sequence of allocated locations in memory, each of which is of the same type. We locate items within an array by their "indices". An "index" serves a similar purpose to the number of an apartment along a floor, or the floor number, or the building number within an apartment complex. In a computer an array may be organized around one, two, three or more indices, which of which is allowed to vary within a predetermined range called a "dimension".

MEMORY STRUCTURE

Again returning to our analogy to street maps, consider the possibility that someone might decide to buy several blocks of a neighborhood and make a subdivision. Within that subdivision, we could have a new street number system, and either a uniform type of house or a variety of types of houses. In organizing computer memory we can do much the same thing with "structures". A "structure" is a heterogeneous combination of variable types and arrays of types composing one aggregate type. Items within the structure are identified by the name of the structure and the name of a particular component.

Generally, the lower a storage is in the hierarchy, the lesser its bandwidth and the greater its access latency is from the CPU. This traditional division of storage to primary, secondary, tertiary and off-line storage is also guided by cost per bit. In contemporary usage, "memory" is usually semiconductor storage read-write random-access memory, typically DRAM (Dynamic-RAM) or other forms of fast but temporary storage. "Storage" consists of storage devices and their media not directly accessible by the CPU (secondary or tertiary storage), typically hard disk drives, optical disc drives, and other devices slower than RAM but non-volatile (retaining contents when powered down). Historically, *memory* has been called *core*, *main memory*, *real storage* or *internal memory* while storage devices have been referred to as *secondary storage*, *external memory* or *auxiliary/peripheral storage*.

Primary storage (also known as *main memory* or *internal memory*), often referred to simply as *memory*, is the only one directly accessible to the CPU. The CPU continuously reads instructions stored there and executes them as required. Any data actively operated on is also stored there in uniform manner.

REFERENCES

1. *Storage* as defined in Microsoft Computing Dictionary, 4th Ed. (c)1999 or in The Authoritative Dictionary of IEEE Standard Terms, 7th Ed., (c) 2000.
2. "Primary Storage or Storage Hardware" (shows usage of term "primary storage" meaning "hard disk storage"). Searchstorage.techtarget.com (2011-06-13). Retrieved on 2011-06-18.
3. J. S. Vitter, *Algorithms and Data Structures for External Memory*, Series on Foundations and Trends in Theoretical Computer Science, now Publishers, Hanover, MA, 2008, ISBN 978-1-60198-106-6.
4. A thesis on Tertiary storage. (PDF) . Retrieved on 2011-06-18.
5. Pearson, Tony (2010). "Correct use of the term Nearline.". *IBM Developerworks, Inside System Storage*. Retrieved 2015-08-16.
6. National Communications System (1996). "Federal Standard 1037C – Telecommunications: Glossary of Telecommunication Terms". General Services Administration. FS-1037C. Retrieved 2007-10-08. See also article Federal Standard 1037C.
7. Mike Chin (8 March 2004). "IS the Silent PC Future 2.5-inches wide?". Retrieved 2008-08-02.
8. Mike Chin (2002-09-18). "Recommended Hard Drives". Retrieved 2008-08-02.
9. Super Talent's 2.5" IDE Flash hard drive – The Tech Report – Page 13. The Tech Report. Retrieved on 2011-06-18.