GROUP 2

A close up of a sign

Description automatically generated

COMPUTER ARCHITECTURE (COSC 403)

COMPUTER SCIENCE (GROUP B)

GROUP PROJECT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ASSOCIATIVE & INTERLEAVED MEMORY SYSTEM ORGANIZATION

SUBMITTED TO: DR. EMMANUEL OGU

DEPARTMENT OF COMPUTER SCIENCE

BABCOCK UNIVERSITY, ILISHAN - REMO

DECEMBER 2019

GROUP LIST

|  |  |  |
| --- | --- | --- |
| S/N | NAME | MATRIC NO |
| 1 | OBAJEMU OLUWATOMISIN JOHN | 16/0435 |
| 2 | NDUBUISI OKECHUKWU C | 17/1739 |
| 3 | NWACHUKWU EBUKA M | 16/0533 |
| 4 | NWABIANI PETER | 16/1740 |
| 5 | NDUMELE CHINONSO | 16/0624 |
| 6 | NDUPU ADAEZE | 16/0516 |
| 7 |  |  |
| 8 |  |  |

**CHAPTER ONE: INTRODUCTION**

A memory is a physical device with locations where data can be stored or retrieved for processing in a computer. Memory devices utilize integrated circuits and are used by hardware and software.A circuit board

Description automatically generated

**Figure 1.0. A memory stick module**

Memory units can be categorized into two: Online or Offline memory. The online memory is the type of memory that doesn’t not need direct physical connection to access its data. The device can be accessed from any location via network. An example includes cloud storages i.e. iCloud, OneDrive etc. The offline memory involves direct physical connection of mass storage device to access its data. It includes volatile storage devices, internal storage devices and external storage devices.

Computer memory can be classified into primary memory and secondary memory. Primary memory is the computer memory that is accessed directly by the CPU. It includes the processor cache, RAM and system ROM. The RAM and cache stores data temporarily while the ROM stored data permanently. Secondary memory is where programs and data are stored on a long-term permanent basis, it gives a slower access to the computer program’s content with a larger capacity and at a cheaper cost. They include hard disks, floppy disks, Solid-state drives and optical disks.

**History of The Computer Memory**

* Punch card – Invented in the 19th century by Herman Hollerith. It is a primitive form of storing and accessing data
* Magnetic Tapes – This was a huge leap in data storage which started in the 1970s. They could store vast amounts of data over a long period of time on an oxide coated piece of tape. A single reel could hold data equivalent to 10,000 punch cards Magnetic storage is still used effectively in this way; reels are now available with capacity of up to 1TB, and they are widely used for archiving purposes, as well as creating back-ups of a high volume.
* DRAM (Dynamic RAM) – This is a chip which was developed in 1968 by Lee Boysel. DRAM began to overtake magnetic core memory in the latter half of the 1970s which led to people having the opportunity to store large amounts of information on a single chip.
* Floppy disks – They were popular form of PC storage solution from the 1970s onwards and they constantly reduced in size. Similar disks are used as storage methods today.
* Hard disk drives/ Solid state drives – The utilize resultant discoveries from the inventions of the punch card and floppy disks. Hard disk drives use magnetic platters to store and retrieve data. Solid state drives use NAND flash memory which is much faster than the magnetic storage.

**CHAPTER TWO: MEMORY HIERARCHY DESIGN**

The memory hierarchy is a pyramid-based arrangement of the various types of computer memory which is ranked based on speed, cost and capacity. The CPU register is at the top of the hierarchy while the tape drive is at the bottom. The memory hierarchy is very important as it greatly affects the performance of the C.P.U. The memory hierarch was created as a source of reference for programmers in order to make the right memory choice.

A close up of text on a white surface

Description automatically generated

**Figure 2.0 A Computer Memory Hierarchy** (Geeks for Geeks, 2019)

**Memory Locality**

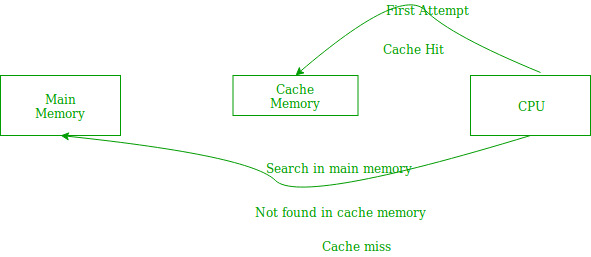
This relates to the memory access pattern where related storage locations or places having the same or similar values are frequently accessed to reduce data latency. There are two types which include:

* Temporal Locality – This includes storing a specific location where at a point, a particular memory is referenced and highly possible for that same address to be referenced in the near future. It increases access speed.
* Spatial Locality - This works in a case where there is always a high probability of referencing the nearby location of recently accessed data.

Memory Locality helps with linear data, cache efficiency, computer program structuring and linear data.

Scenario – The cache uses the spatial locality concepts to hold recently used data. It aims to receive its capacity to increase performance. Elements close by the cache are bought in one cache at a line, with one element being referenced, a few adjacent elements are brought in following up from the lower pyramid since the cache cannot directly access the random-access memory. A good program locality works with the higher hierarchy levels to access data often. A poor program locality makes use of the lower hierarchy.

The property of memory locality is shown mainly in loops, recursive functions and subroutine calls in a program. In case of loops, the CPU repeatedly refers to the set of instructions inside the loop and in case of subroutine calls, it happens every time the instructions are fetched from memory. References to data times also get localized.



**Figure 2.1 Cache Hit/Miss** (GeeksforGeeks, 2019)

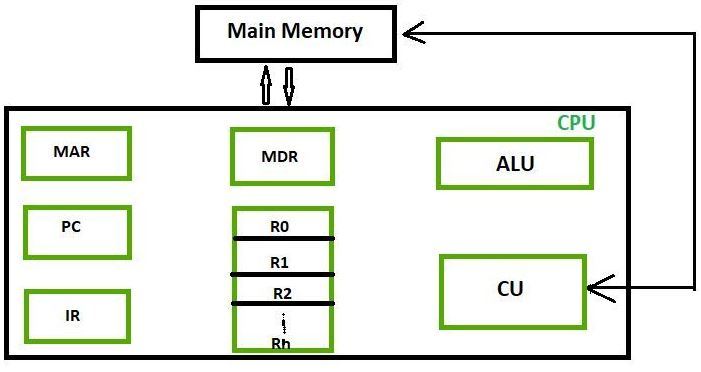
The above figure shows that in a data/instruction fetch. The CPU checks the cache memory as it is very fast and close to it. If found, it is known as a cache hit. If not found, it is known as a cache miss.

In the case of a cache miss, the CPU either fetches the instruction and uses it or it fetches the instruction and stores it in the cache for faster access when fetched next.

**CHAPTER THREE: COMPUTER MEMORY TYPES**

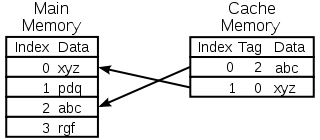
This chapter talks about the various types of memory with respect to the computer memory hierarchy pyramid.

1. Registers – These are a memory location which is quickly accessible in the computer architecture. It is the fastest, smallest and most expensive device in the memory hierarchy and it is closest to the CPU core. As not having a large amount of memory, the CPU register is involved in loading the larger memory sets into registers where it is variably used for arithmetic operations or execute immediate instructions. They can be classified into the Accumulator, General purpose register and Special purpose registers.



**Figure 3.0 Register classes** (Geeks for Geeks, 2019)

1. Cache – This contains the most frequently used data stored in its memory locations and acts as an intermediary between the CPU and RAM. It drastically reduces the cost in accessing data from the main memory. The cache is an associative memory (discussed in later chapters) where the content value is looked up in the memory by its address stored in the cache through a hash function. The hash function basically maps a key to the value of the content in the computer primary memory. Data is transferred between the memory and cache in blocks of fixed size while data is also transferred back to the main CPU from the cache in words transfer.

[](https://en.wikipedia.org/wiki/File:Cache,basic.svg)

**Figure 3.1 CPU Memory Cache Operation** (Wikipedia, 2019)

1. Main Memory (RAM) – This is the segment of the computer memory which is volatile and program codes are kept for execution.
2. Flash Memory - Solid-state drive – This is a storage device that stored data permanently on a solid-state flash memory which is made up of a flash controller and NAND flash memory chips. It is optimized to deliver high read and write performance for sequential and random data requests.



**Figure 3.3 A Solid-state drive** (PC World, 2012)

1. Hard disk drive – This is the largest storage medium compared to those discussed prior to this. It spans terabytes but it is slow in retrieving blocks of data. They use magnetic mechanisms to manipulate data non-sequentially. Data is stored in Logical blocks.



**Figure 3.3 A Hard Disk drive** (PC World, 2012)

Tape Storage – This is the cheapest, oldest and slowest memory device on a memory hierarchy pyramid. Its capacity is unlimited, and it represents one of the most iconic archival storage medium where most historical data is stored. Modern magnetic tapes are in cassette and cartridge form.

[](https://www.em360tech.com/wp-content/uploads/2017/08/ibm-tapes-PH2401.jpg)

**Figure 3.4 A Tape Drive** (Enterprise Management 360, 2017)

**CHAPTER FOUR: MEMORY SYSTEM ORGANIZATION**

**INTERLEAVED MEMORY SYSTEM ORGANIZATION**

This is a design that compensates for the relatively slow speed of the main memory by spreading memory addresses evenly across memory banks which increases memory throughput because waiting for memory banks to become ready is reduced. The individual access to memory banks is possible without dependency on any other one. This causes an increase in memory speed.

Memory addressed are allocated to each memory bank in turn. For example, in an interleaved system with two memory banks, logical address 32 belongs to bank 0, 33 will belong to bank 1, 34 to 0 and so on. Interleaved memory results in contiguous read and write operations.

For example: If we have 4 memory banks with each containing 256 bytes, the block-oriented scheme will assign addressed 0 – 255 to the first bank, 256 to 511 to the second bank. In an interleaved memory, the virtual address 0 will be with the first bank, 1 with the second memory bank, 2 with the third bank, 3 with the fourth and 4 with the first bank.

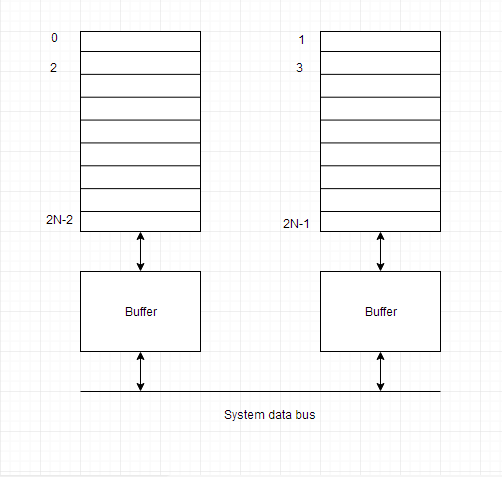
This results in the CPU being able to access alternate sections immediately without waiting for the memory to be cached. The multiple memory banks take turns for data supply.

From the above scenario of 4 memory banks, data with virtual addresses 0 – 3 can be accessed simultaneously as they reside in separate memory banks, hence we do not have to wait for the completion of a data fetch to begin with the next.

An interleaved memory with n banks is said to be n-way interleaved. In an interleave system, there are still two banks of DRAM but logically, the system sees one bank of memory that is twice as large.

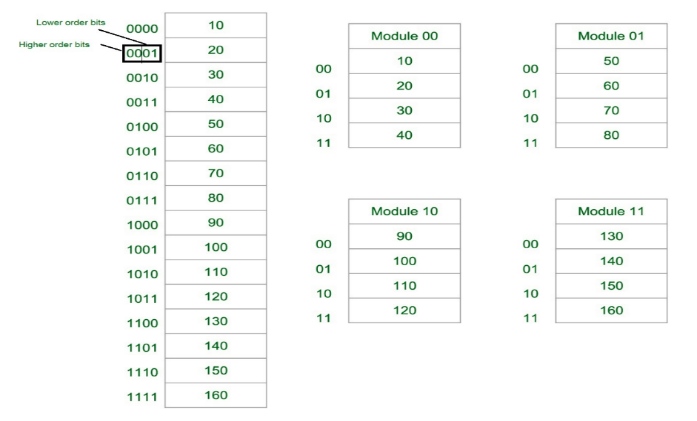
In the interleaved bank representation below with 2 memory banks, the first word of bank 0 is followed by that of bank 1, which is followed by the second-long word of bank 0, which is followed by the second-long word of bank 1 and so on.

The following figure shows the organization of two physical banks of n long words. All even long words of logical bank are located in physical bank 0 and all odd long words are located in physical bank 1.



**Figure 4.0 Organization of Two Physical Banks of n Long words** (Study Tonight, 2019)

Interleaved Memory system organization can also be seen as an abstraction technique. It divides memory into a number of modules such that successive words in the address space are placed in a different module.



**Figure 4.1 Consecutive words in a module** (Geeks for Geeks, 2019)

Let us assume 16 Data’s to be transferred to the Four module. Where module 00 be module 1, module 01 be module 2, module 10 be module 3 & module 11 be module 4. Also 10,20,30….130 are the data to be transferred.

From the figure above in Module 1, 10 [Data] is transferred then 20, 30 and 40. That means the data is added consecutively in the module till it becomes full.

The MSB provides the address of the module and the LSB provides the address of the data in the module.

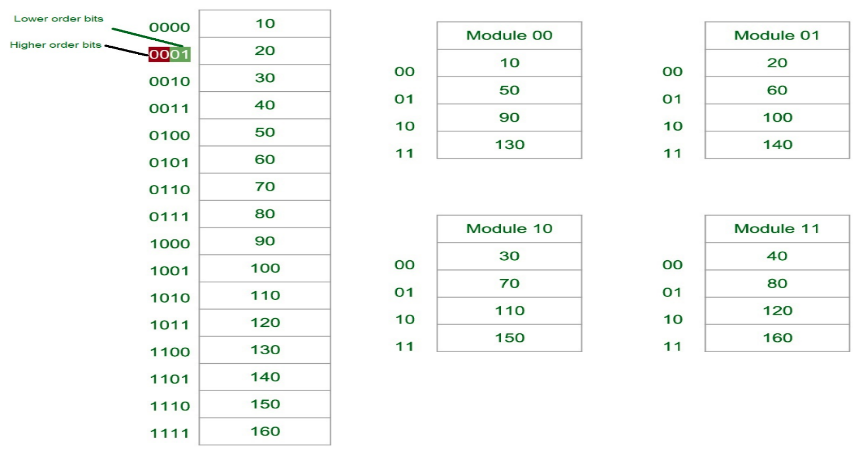
For example: To get 90 [Data], 1000 will be provided by the processor. In this, 10 will indicate that the data is in module 10 [module 3] & 00 is the address of 90 in Module 10 [module 3]. That is,

Module 1 🡪 10, 20, 30, 40

Module 2 🡪 50, 60, 70, 80

Module 3 🡪 90, 100, 110, 120

Module 4 🡪 130, 140, 150, 160



**Figure 4.2 Consecutive words in a consecutive module** (Geeks for Geeks, 2019)

Again, lets assume 16 data’s to be transferred to the four module. But for now, the consecutive data are added in the consecutive module. That is, 10 [Data] is added in Module 1, 20[Data] in Module 2 and so on.

The LSB provides the address of the module and the MSB provides the address of the data in the module.

We can take an example which is as follows: To get 90 (Data), 1000 will be provided by the processor. In this, 10 will indicate that the data is in module 10 (module 3) & 00 is the address of 90 in module 10 (module 3). So,

Module 1 🡪 10, 50, 90, 130

Module 2 🡪 20, 60, 100, 140

Module 3 🡪 30, 70, 110, 120

Module 4 🡪 40, 80, 150, 160

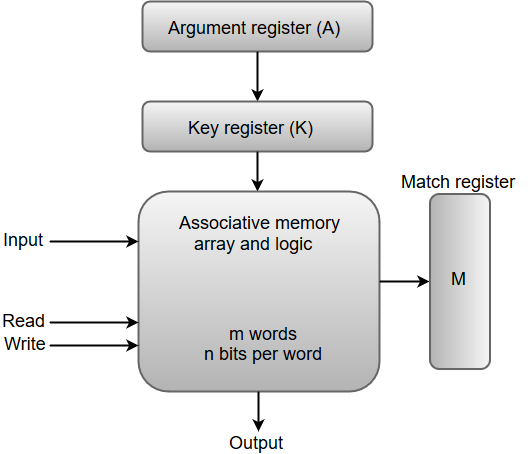
A major advantage of interleaving is to improve the access time of the main memory (reduce latency). As mentioned before, interleaving allows access to multiple modules at the same time which results in parallelism. From Figure 4.2, the data can be acquired from the Module using the Higher bits. This method Uses memory effectively.

**ASSOCIATIVE MEMORY SYSTEM ORGANIZATION**

This is a design that involves having its stored data identified for access by the content of the data itself rather by an address or memory location. When a write operation is performed, no address or memory location is given to the word. The memory is capable of finding an empty unused location to store the word and when read operations are performed, the content of the word is specified and words that match the content are located and marked for reading.

This is also known as content addressable memory. It is a memory chip where each bit position can be compared. In this, the content is compared in each bit cell which allows a very fast table lookup. Since the entire chip can be compared, content is randomly stored without considering addressing scheme. These chips have less storage capacity than regular memory chips.

The following diagram shows the block representation of an associative memory .

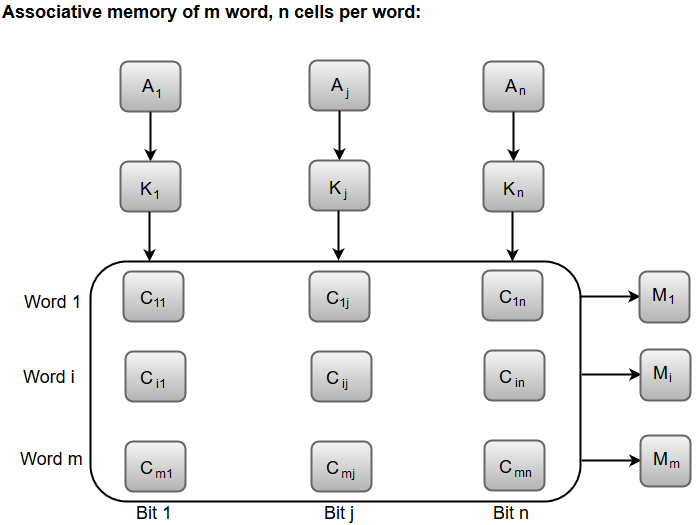


**Figure 4.3 Block diagram of an associative memory** (JavaTpoint, 2019)

An associative memory consists of a memory array and logic for ‘m’ words with ‘n’ bits per word. The functional registers like the argument register **A** and key register **K** each have **n**  bits, one for each word’s bit. The match register **M** consists of **m**  bits, one for each memory word.

The words which are kept in the memory are compared in parallel with the content of the argument register. The key register (K) provides a mask for choosing a particular field or key in the argument word. If the key register contains a binary value of all 1s, then the entire argument is compared with each memory word. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus, the key provides a mask for identifying a piece of information which specifies how the reference to memory is made.

The following diagram can represent the relation between the memory array and the external registers in an associative memory.



**Figure 4.4 Relation between the memory array and the external registers in an associative memory.** (JavaTpoint, 2019)

The cells present inside the memory array are marked by the letter C with the first subscript being the word number and the second being the bit position.

A bit Aj in the argument register is compared with all the bits in column j of the array provided that Kj=1. This process is done for all columns on j for 1 to n

If a match occurs between all the unmasked bits of the argument and the bits in word I, the corresponding bit Mi in the match register Is set to 1. If one or more unmasked bits of the argument and the words do not match, Mi is cleared to 0 which means content is not found.

Associative memory is a memory that is accessed through content rather than through a specific address. This is found on a computer hard drive and it is used only in specific high-speed searching applications. Associative memory works through the computer user providing a data word and then searching throughout the entire computer memory to see if the word is there. If the computer finds the data word then it offers a list of all the storage addresses where the word was found for the user. Associative memory is faster than Random-access memory in almost every search application, the reason it isn’t the preferred choice for most computer users is that the speed comes with a spike in cost. The reason for this price increase is that the associative memory computers need each cell to have the full storage capacity and logic circuits that can match content with external argument. Associative memory computers are best for users that require searches to take place quickly and whose searches are critical for job performance on the machine.

# References

Dakota Digital Ltd. (2013). *A Brief History of Computer Memory & Storage*. Retrieved from Cision: https://news.cision.com/dakota-digitalltd/r/a-brief-history-of-computer-memory---storage,c9397271

Enterprise Management 360. (2017). *IBM magnetic tape storage system breaks world record*. Retrieved from Enterprise Management 360: https://www.em360tech.com/tech-news/ibm-claims-new-world-record-magnetic-tape-storage/

Geeks for Geeks. (2019). *Cache Memory in Computer Organization*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/cache-memory-in-computer-organization/

Geeks for Geeks. (2019). *Different Classes of CPU Registers*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/different-classes-of-cpu-registers/

Geeks for Geeks. (2019). *Memory Hierarchy Design and its Characteristics*. Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/memory-hierarchy-design-and-its-characteristics/

Geeks for Geeks. (2019). *Memory Interleaving*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/memory-interleaving/

GeeksforGeeks. (2019, December). *Locality of Reference and Cache Operation in Cache Memory*. Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/locality-of-reference-and-cache-operation-in-cache-memory/

JavaTpoint. (2019). *Associative Memory*. Retrieved from JavaTpoint: https://www.javatpoint.com/coa-associative-memory

PC World. (2012). *How to partition and format your hard drive in Windows*. Retrieved from PC World: https://www.pcworld.com/article/248980/how-to-partition-and-format-your-hard-drive-in-windows.html

Rouse, M. (2019). *SSD (solid-state drive)* . Retrieved from Tech Target: https://searchstorage.techtarget.com/definition/SSD-solid-state-drive

Study Tonight. (2019). *What is Interleaved Memory?* Retrieved from Study Tonight: https://www.studytonight.com/computer-architecture/interleaved-memory

Tech Terms. (2019). *Primary Memory*. Retrieved from Tech Terms: https://techterms.com/definition/primary\_memory

*What is Associative Memory in Computer Organization?* (2019). Retrieved from Reference: https://www.reference.com/technology/associative-memory-computer-organization-c4d0fcbe6248554

Wikipedia. (2019, December 1). *Cache (computing)*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Cache\_(computing)