Unit - 4

Design of Control Unit

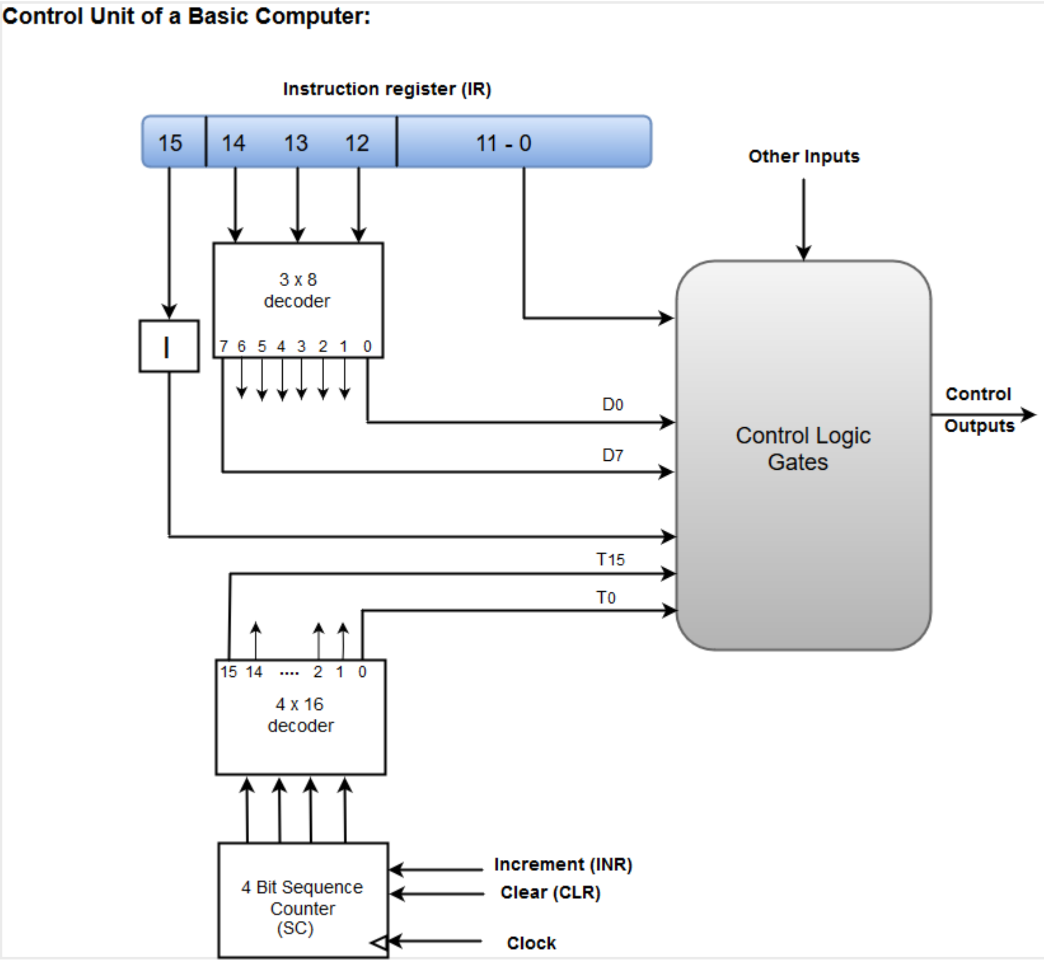
The Control Unit is classified into two major categories:

1. Hardwired Control
2. Microprogrammed Control

Hardwired Control

The Hardwired Control organization involves the control logic to be implemented with gates, flip-flops, decoders, and other digital circuits.

The following image shows the block diagram of a Hardwired Control organization.



- A Hard-wired Control consists of two decoders, a sequence counter, and a number of logic gates.
- An instruction fetched from the memory unit is placed in the instruction register (IR).
- The component of an instruction register includes; I bit, the operation code, and bits 0 through 11.
- The operation code in bits 12 through 14 are coded with a 3 x 8 decoder.

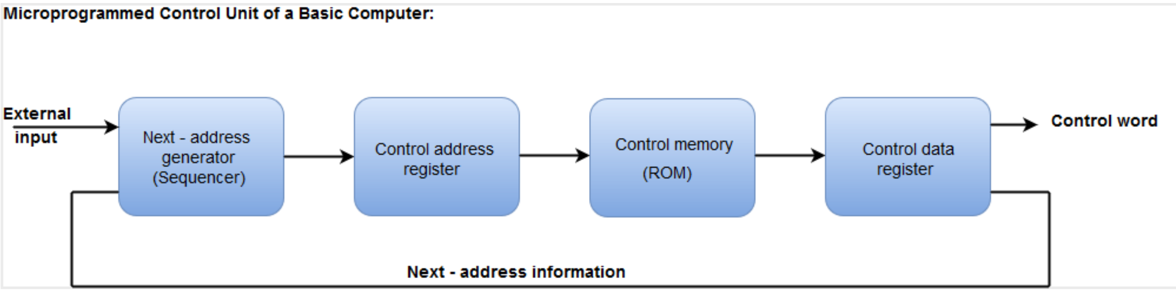
- The outputs of the decoder are designated by the symbols D0 through D7.
- The operation code at bit 15 is transferred to a flip-flop designated by the symbol I.
- The operation codes from Bits 0 through 11 are applied to the control logic gates.
- The Sequence counter (SC) can count in binary from 0 through 15.

Micro-programmed Control

The Microprogrammed Control organization is implemented by using the programming approach.

In Microprogrammed Control, the micro-operations are performed by executing a program consisting of micro-instructions.

The following image shows the block diagram of a Microprogrammed Control organization.



- The Control memory address register specifies the address of the micro-instruction.
- The Control memory is assumed to be a ROM, within which all control information is permanently stored.
- The control register holds the microinstruction fetched from the memory.
- The micro-instruction contains a control word that specifies one or more micro-operations for the data processor.
- While the micro-operations are being executed, the next address is computed in the next address generator circuit and then transferred into the control address register to read the next microinstruction.
- The next address generator is often referred to as a micro-program sequencer, as it determines the address sequence that is read from control memory.

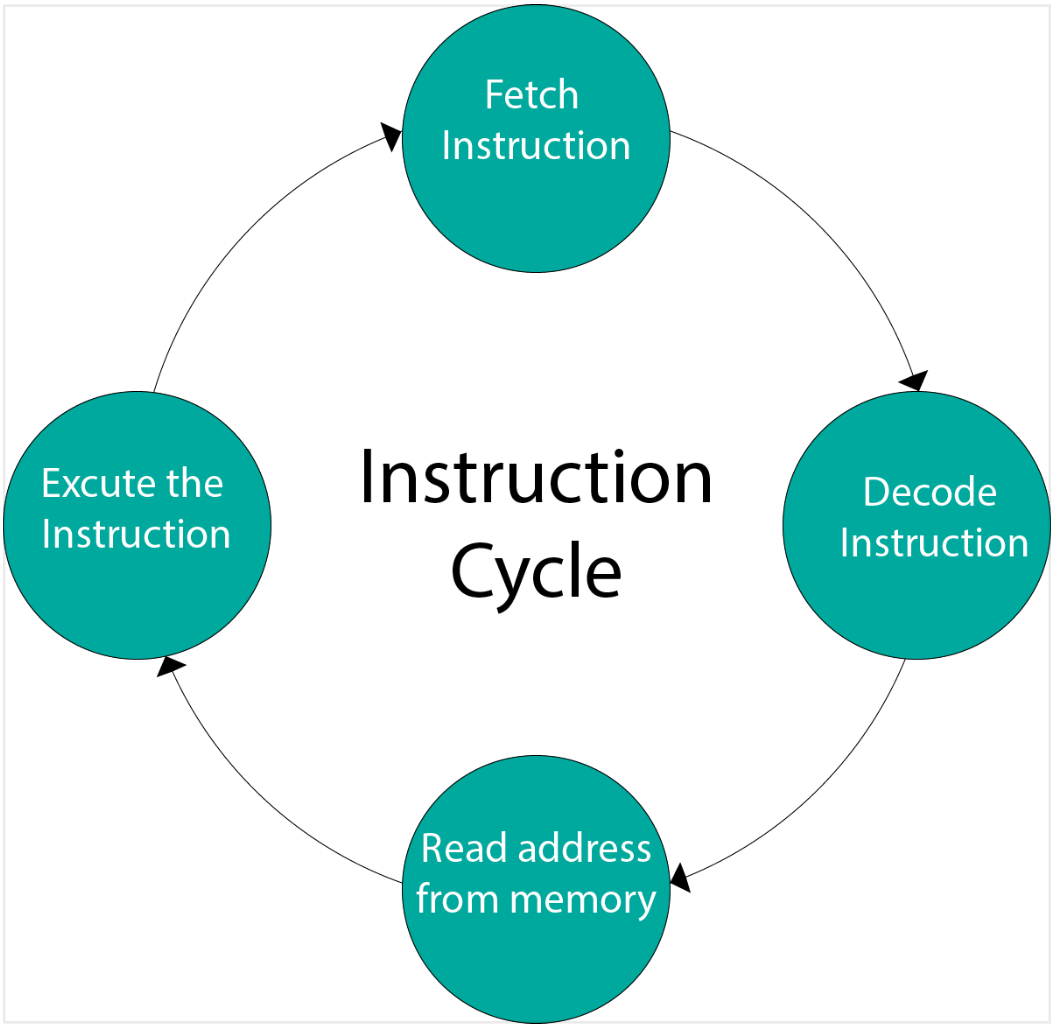
Next TopicInstruction Cycle

Instruction Cycle

A program residing in the memory unit of a computer consists of a sequence of instructions. These instructions are executed by the processor by going through a cycle for each instruction.

In a basic computer, each instruction cycle consists of the following phases:

1. Fetch instruction from memory.
2. Decode the instruction.
3. Read the effective address from memory.
4. Execute the instruction.

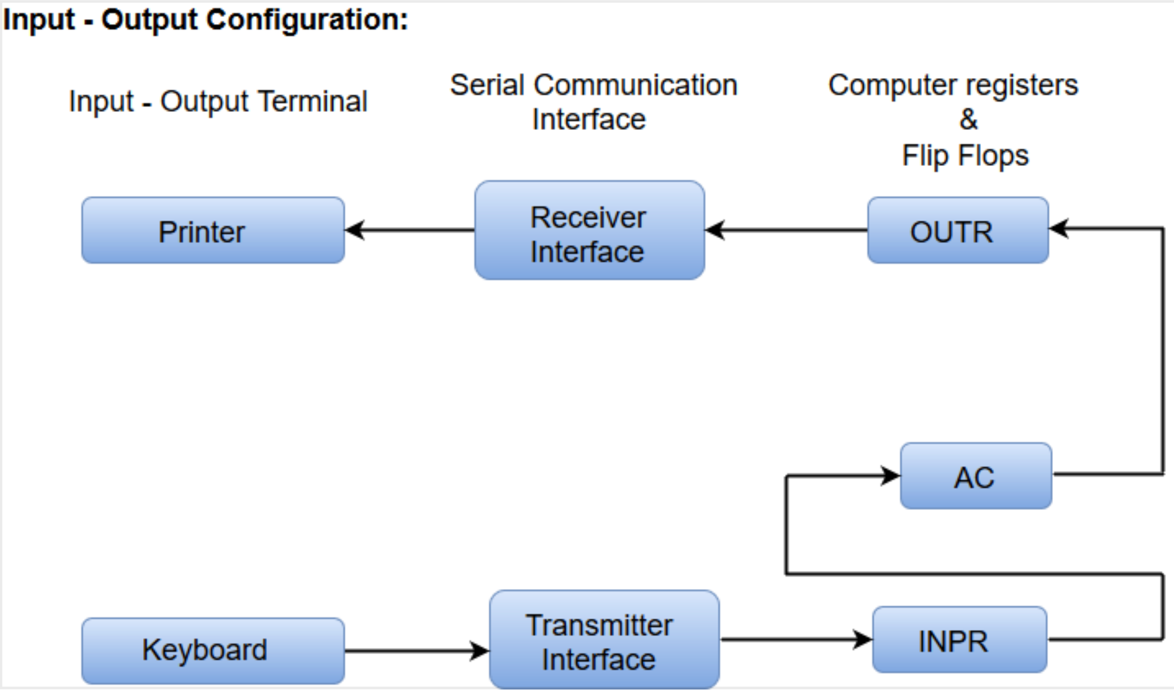


Input-Output Configuration

In computer architecture, input-output devices act as an interface between the machine and the user.

Instructions and data stored in the memory must come from some input device. The results are displayed to the user through some output device.

The following block diagram shows the input-output configuration for a basic computer.



- The input-output terminals send and receive information.
- The amount of information transferred will always have eight bits of an alphanumeric code.
- The information generated through the keyboard is shifted into an input register 'INPR'.
- The information for the printer is stored in the output register 'OUTR'.
- Registers INPR and OUTR communicate with a communication interface serially and with the AC in parallel.
- The transmitter interface receives information from the keyboard and transmits it to INPR.
- The receiver interface receives information from OUTR and sends it to the printer serially.

Design of a Basic Computer

A basic computer consists of the following hardware components.

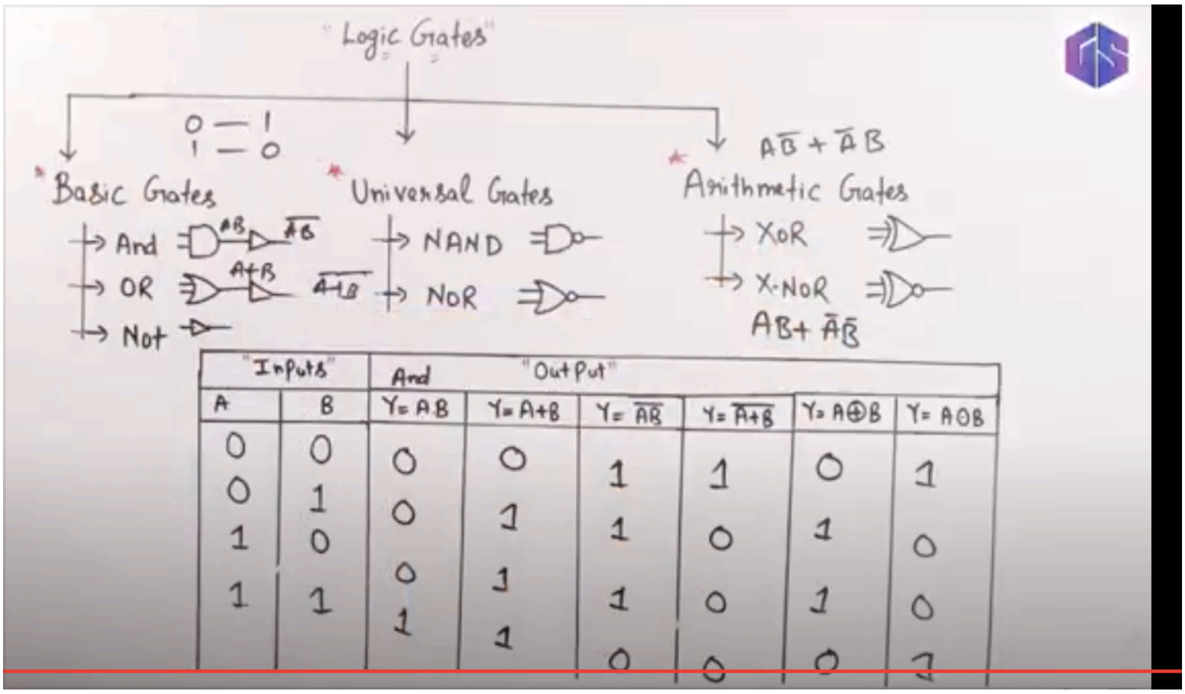
1. A memory unit with 4096 words of 16 bits each
2. Registers: AC (Accumulator), DR (Data register), AR (Address register), IR (Instruction register), PC (Program counter), TR (Temporary register), SC (Sequence Counter), INPR (Input register), and OUTR (Output register).
3. Flip-Flops: I, S, E, R, IEN, FGI and FGO

Note: FGI and FGO are corresponding input and output flags which are considered as control flip-flops.

1. Two decoders: a 3 x 8 operation decoder and 4 x 16 timing

decoder
2. A 16-bit common bus
3. Control Logic Gates
4. The Logic and Adder circuits connected to the input of AC.

Logic Gates at a glance



Classification of Memory

In computers, **memory** is the most essential component of the normal functioning of any system. The computer system categorizes the memory for different purposes and uses. In this section, we have discussed the **classification of memory** in detail. Also, we will discuss **types of memory, features of memory, RAM, ROM, SRAM, DRAM,** and its advantages and disadvantages.

What is computer memory?

Computer memory is any physical device, used to store data, information or instruction temporarily or permanently. It is the collection of storage units that stores binary information in the form of bits. The memory block is split into a small number of components, called cells. Each cell has a unique address to store the data in memory, ranging from zero to memory size minus one. For example, if the size of computer memory is 64k words, the memory units have

64 * 1024 = 65536 locations or cells. The address of the memory's cells varies from 0 to 65535.

Why do we need a computer memory?
In the computer system, we need computer memory to store various types of data like text, images, video, audio, documents, etc. We can retrieve it when the data is required. For example, when we write and execute any computer program, it is initially stored in primary memory. If the processor does not need particular items for a longer time, the program or data is automatically saved into the permanent or secondary memory. Then the data is called from secondary memory to main memory and performs the execution of codes.

Features of Memory

Following are the different features of the memory system that includes:

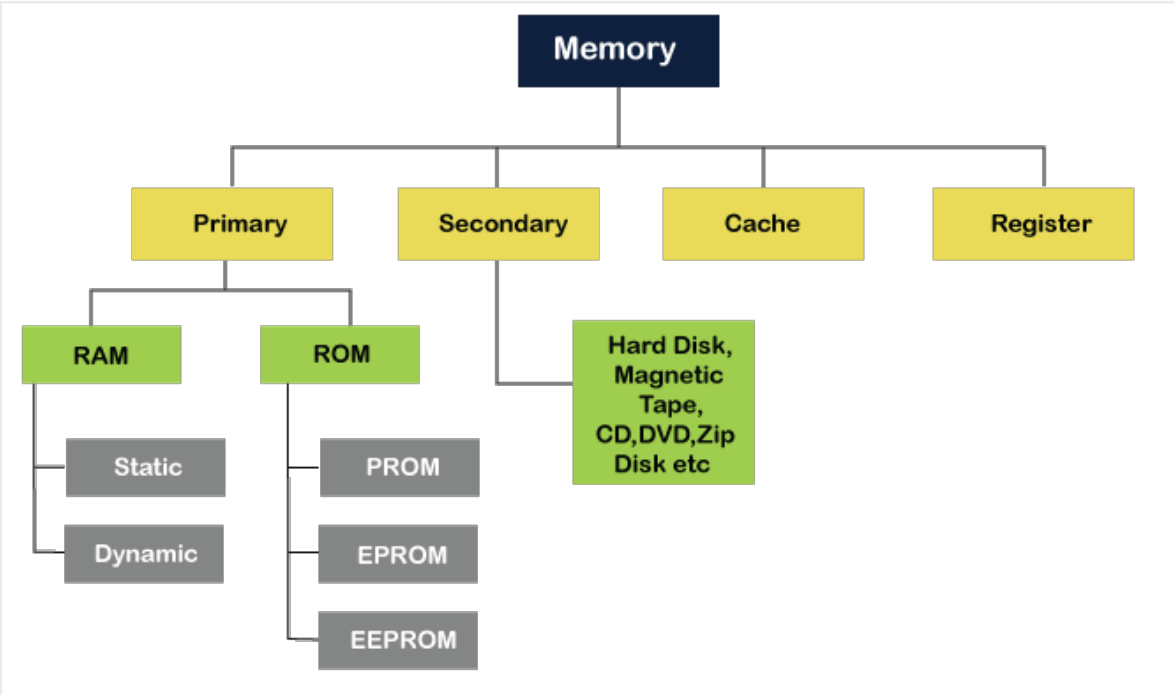
1. **Location:** It represents the internal or external location of the memory in a computer. The internal memory is inbuilt in computer memory. It is also known as primary memory. the example of primary memory are registers, cache and main memory. Whereas, external memory is the separate storage device from the computer, such as disk, tape, USB pen drive.
2. **Capacity:** It is the most important feature of computer memory. Storage capacity can vary in external and internal memory. External devices' storage capacity is measured in terms of bytes, whereas the internal memory is measured with bytes or words. The storage word length can vary in bits, such as 8, 16 or 32 bits.
3. **Access Methods:** Memory can be accessed through four modes of memory.
    - **DMA:** As the name specifies, Direct Memory Address (DMA) is a method that allows input/output (I/O) devices to access or retrieve data directly or from the main memory.
    - **Sequential Access Method:** The sequential access method is used in a data storage device to read stored data sequentially from the computer memory. Whereas, the data received from random access memory (RAM) can be in any order.
    - **Random Access Method:** It is a method used to randomly access data from memory. This method is the

opposite of SAM. For example, to go from A to Z in random access, we can directly jump to any specified location. In the Sequential method, we have to follow all intervening from A to Z to reach at the particular memory location.
    - **Associative Access Method:** It is a special type of memory that optimizes search performance through defined data to directly access the stored information based on a memory address.
4. **Unit of transfer:** As the name suggests, a unit of transfer measures the transfer rate of bits that can be read or write in or out of the memory devices. The transfer rate of data can be different in external and internal memory.
    - **Internal memory:** The transfer rate of bits is mostly equal to the word size.
    - **External memory:** The transfer rate of bit or unit is not equal to the word length. It is always greater than a word or may be referred to as **blocks**.
5. **Performance:** The performance of memory is majorly divided into three parts.
    - **Access Time:** In random access memory, it represents the total time taken by memory devices to perform a read or write operation that an address is sent to memory.
    - **Memory Cycle Time:** Total time required to access memory block and additional required time before starting second access.
    - **Transfer rate:** It describes the transfer rate of data used to transmit memory to or from an external or internal memory device. Bit transfer can be different for different external and internal devices.
6. **Physical types:** It defines the physical type of memory used in a computer such as magnetic, semiconductor, magneto-optical and optical.
7. **Organization:** It defines the physical structure of the bits used in memory.
8. **Physical characteristics:** It specifies the physical behavior of the memory like volatile, non-volatile or non-erasable memory. Volatile memory is known as RAM, which requires power to retain stored information, and if any power loss has occurred, stored data will be lost. Non-volatile memory is a

permanent storage memory that is used to obtain any stored information, even when the power is off. Non-erasable memory is a type of memory that cannot be erased after the manufactured like ROM because at the time of manufactured ROM are programmed.

Classification of Memory

The following figure represents the classification of memory:



Primary or Main Memory

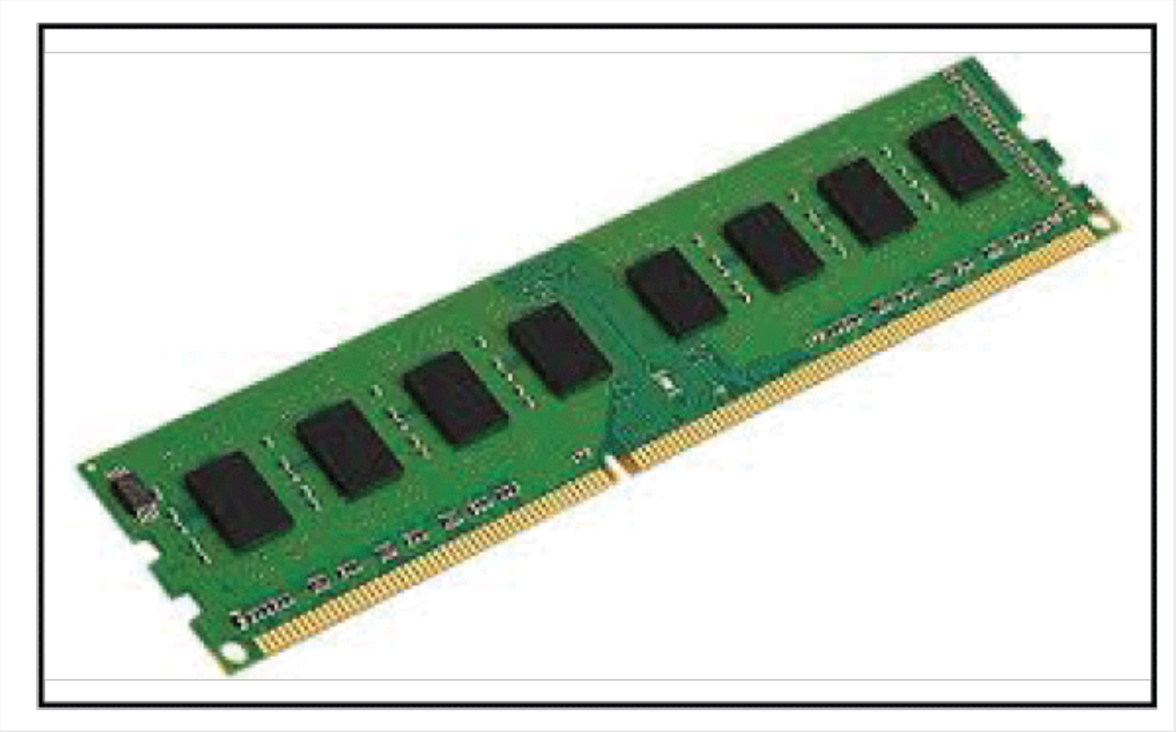
Primary memory is also known as the computer system's main memory that communicates directly within the CPU, Auxiliary memory and the Cache memory. Main memory is used to kept programs or data when the processor is active to use them. When a program or data is activated to execute, the processor first loads instructions or programs from secondary memory into main memory, and then the processor starts execution. Accessing or executing of data from primary memory is faster because it has a cache or register memory that provides faster response, and it is located closer to the CPU. The primary memory is volatile, which means the data in memory can be lost if it is not saved when a power failure occurs. It is costlier than secondary memory, and the main memory capacity is limited as compared to secondary memory.

The primary memory is further divided into two parts:

1. RAM (Random Access Memory)
2. ROM (Read Only Memory)

Random Access Memory (RAM)

Random Access Memory (RAM) is one of the faster types of main memory accessed directly by the CPU. It is the hardware in a computer device to temporarily store data, programs or program results. It is used to read/write data in memory until the machine is working. It is volatile, which means if a power failure occurs or the computer is turned off, the information stored in RAM will be lost. All data stored in computer memory can be read or accessed randomly at any time.



There are two types of RAM:

- SRAM
- DRAM

DRAM: DRAM **(Dynamic Random-Access Memory)** is a type of RAM that is used for the dynamic storage of data in RAM. In DRAM, each cell carries one-bit information. The cell is made up of two parts: a **capacitor** and a **transistor**. The size of the capacitor and the transistor is so small, requiring millions of them to store on a single chip. Hence, a DRAM chip can hold more data than an SRAM chip of the same size. However, the capacitor needs to be continuously refreshed to retain information because DRAM is volatile. If the power is switched off, the data store in memory is lost.

Characteristics of DRAM

1. It requires continuously refreshed to retain the data.
2. It is slower than SRAM
3. It holds a large amount of data
4. It is the combination of capacitor and transistor
5. It is less expensive as compared to SRAM

6. Less power consumption

SRAM: SRMA **(Static Random-Access Memory)** is a type of RAM used to store static data in the memory. It means to store data in SRAM remains active as long as the computer system has a power supply. However, data is lost in SRAM when power failures have occurred.

Characteristics of Static Ram

1. It does not require to refresh.
2. It is faster than DRAM
3. It is expensive.
4. High power consumption
5. Longer life
6. Large size
7. Uses as a cache memory

SRAM Vs. DRAM

SRAM	DRAM
It is a Static Random-Access Memory.	It is a Dynamic Random Access Memory.
The access time of SRAM is low.	The access time of DRAM is high.
It uses flip-flops to store each bit of information.	It uses a capacitor to store each bit of information.
It does not require periodic refreshing to preserve the information.	It requires periodically refreshing to preserve the information.
It uses in cache memory.	It is used in the main memory.
The cost of SRAM is expensive.	The cost of DRAM is less expensive.
It has a complex structure.	Its structure is simple.
It requires low power consumption.	It requires more power consumption.

Advantages of RAM

- It is a faster type of memory in a computer.
- It requires less power to operate.
- Program loads much faster
- More RAM increases the performance of a system and can multitask.
- Perform read and write operations.
- The processor can read information faster than a hard disc,

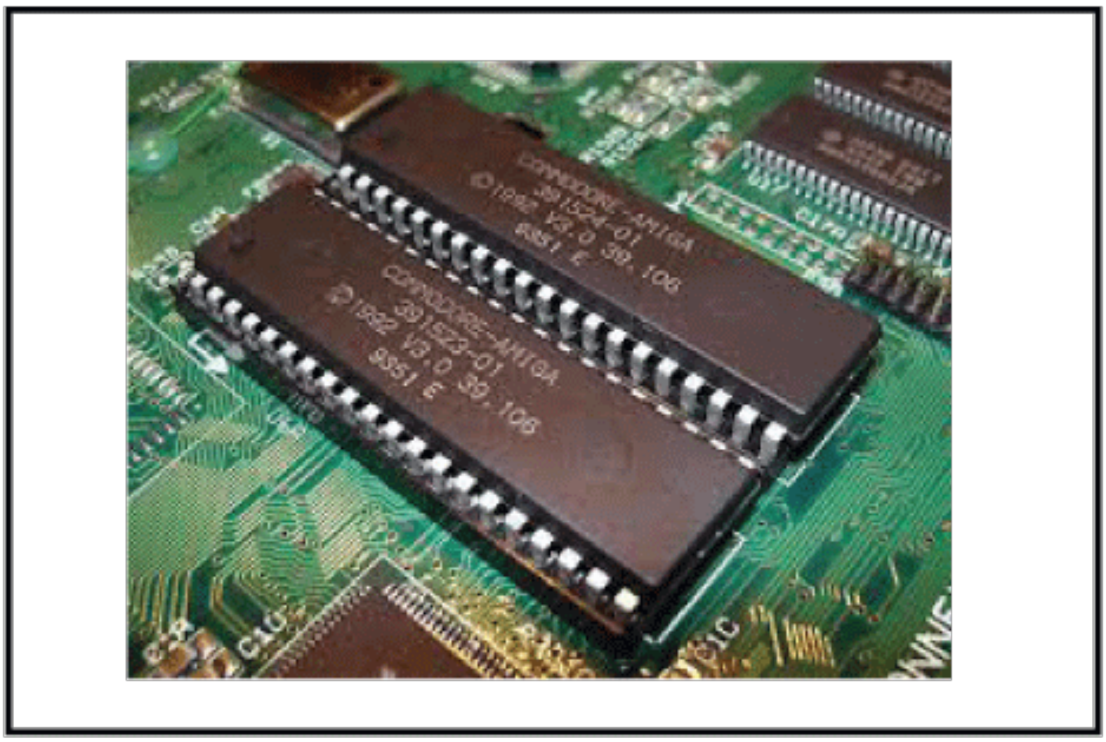
floppy, USB, etc.

Disadvantages of RAM

- Less RAM reduces the speed and performance of a computer.
- Due to volatile, it requires electricity to preserve the data.
- It is expensive than ROM
- It is unreliable as compared to ROM
- The Size of RAM is limited.

Read-Only Memory (ROM)

ROM is a memory device or storage medium that is used to **permanently** store information inside a chip. It is a read-only memory that can only read stored information, data or programs, but we cannot write or modify anything. A ROM contains some important instructions or program data that are required to start or boot a computer. It is a **non-volatile** memory; it means that the stored information cannot be lost even when the power is turned off or the system is shut down.



Types of ROM

There are five types of Read Only Memory:

1. **MROM (Masked Read Only Memory):**
   MROM is the oldest type of read-only memory whose program or data is pre-configured by the integrated circuit manufacture at the time of manufacturing. Therefore, a

program or instruction stored within the MROM chip cannot be changed by the user.

2. **PROM (Programmable Read Only Memory):**
   It is a type of digital read-only memory, in which the user can write any type of information or program only once. It means it is the empty PROM chip in which the user can write the desired content or program only once using the special PROM programmer or PROM burner device; after that, the data or instruction cannot be changed or erased.
3. **EPROM (Erasable and Programmable Read Only Memory):**
   It is the type of read only memory in which stored data can be erased and re-programmed only once in the EPROM memory. It is a non-volatile memory chip that holds data when there is no power supply and can also store data for a minimum of 10 to 20 years. In EPROM, if we want to erase any stored data and re-programmed it, first, we need to pass the ultraviolet light for 40 minutes to erase the data; after that, the data is re-created in EPROM.
4. **EEPROM (Electrically Erasable and Programmable Read Only Memory):**
   The EEROM is an electrically erasable and programmable read only memory used to erase stored data using a high voltage electrical charge and re-programmed it. It is also a non-volatile memory whose data cannot be erased or lost; even the power is turned off. In EEPROM, the stored data can be erased and reprogrammed up to 10 thousand times, and the data erase one byte at a time.
5. **Flash ROM:**
   Flash memory is a non-volatile storage memory chip that can be written or programmed in small units called Block or Sector. Flash Memory is an EEPROM form of computer memory, and the contents or data cannot be lost when the power source is turned off. It is also used to transfer data between the computer and digital devices.

Advantages of ROM

1. It is a non-volatile memory in which stored information can be lost even power is turned off.
2. It is static, so it does not require refreshing the content every time.

3. Data can be stored permanently.
4. It is easy to test and store large data as compared to RAM.
5. These cannot be changed accidently
6. It is cheaper than RAM.
7. It is simple and reliable as compared to RAM.
8. It helps to start the computer and loads the OS.

Disadvantages of ROM

1. Store data cannot be updated or modify except to read the existing data.
2. It is a slower memory than RAM to access the stored data.
3. It takes around 40 minutes to destroy the existing data using the high charge of ultraviolet light.
4.

RAM Vs. ROM

RAM	ROM
It is a Random-Access Memory.	It is a Read Only Memory.
Read and write operations can be performed.	Only Read operation can be performed.
Data can be lost in volatile memory when the power supply is turned off.	Data cannot be lost in non-volatile memory when the power supply is turned off.
It is a faster and expensive memory.	It is a slower and less expensive memory.
Storage data requires to be refreshed in RAM.	Storage data does not need to be refreshed in ROM.
The size of the chip is bigger than the ROM chip to store the data.	The size of the chip is smaller than the RAM chip to store the same amount of data.
Types of RAM: DRAM and SRAM	Types of ROM: MROM, PROM, EPROM, EEPROM

Secondary Memory

Secondary memory is a **permanent storage** space to hold a large amount of data. Secondary memory is also known as external memory that representing the various storage media (hard drives, USB, CDs, flash drives and DVDs) on which the computer data and program can be saved on a long term basis. However, it is cheaper and slower than the main memory. Unlike primary memory, secondary memory cannot be accessed directly by the CPU. Instead of that, secondary memory data is first loaded into the RAM (Random Access Memory) and then sent to the processor to read and update the data. Secondary memory devices also include magnetic disks like hard disk and floppy disks, an optical disk such as CDs and CDROMs, and magnetic tapes.

Features of Secondary Memory

- Its speed is slower than the primary/ main memory.
- Store data cannot be lost due to non-volatile nature.
- It can store large collections of different types, such as audio, video, pictures, text, software, etc.
- All the stored data in a secondary memory cannot be lost because it is a permanent storage area; even the power is turned off.
- It has various optical and magnetic memories to store data.

Types of Secondary Memory

The following are the types of secondary memory devices:

Hard Disk

A hard disk is a computer's permanent storage device. It is a non-volatile disk that permanently stores data, programs, and files, and cannot lose store data when the computer's power source is switched off. Typically, it is located internally on computer's motherboard that stores and retrieves data using one or more rigid fast rotating disk platters inside an air-sealed casing. It is a large storage device, found on every computer or laptop for permanently storing installed software, music, text documentation, videos, operating system, and data until the user did not delete.

Floppy Disk

A floppy disk is a secondary storage system that consisting of thin, flexible magnetic coating disks for holding electronic data such as computer files. It is also known as Floppy Diskette that comes in three sizes like 8 inches, 5.5 inches and 3.5 inches. The stored data of a floppy disk can be accessed through the floppy disk drive. Furthermore, it is the only way through a new program installed on a computer or backup of the information. However, it is the oldest type of portable storage device, which can store data up to 1.44 MB. Since most programs were larger, that required multiple floppy diskettes to store large amounts of data. Therefore, it is not used due to very low memory storage.

CD (Compact Disc)

A CD is an optical disk storage device, stands for Compact Disc. It is a storage device used to store various data types like audio, videos, files, OS, Back-Up file, and any other information useful to a computer. The CD has a width of 1.2 mm and 12 cm in height, which can store approximately 783 MB of data size. It uses laser light to read and write data from the CDs.

Types of CDs

1. **CD-ROM (Compact Disc Read Only Memory):** It is mainly used for bulk size mass like audio CDs, software and computer games at the time of manufacture. Users can only read data, text, music, videos from the disc, but they cannot modify or burnt it.
2. **CD-R (Compact Disc Recordable):** The type of Compact Disc used to write once by the user; after that, it cannot be modified or erased.
3. **CD-RW (Compact Disc Rewritable):** It is a rewritable CD disc, often used to write or delete the stored data.

DVD Drive/Disc

DVD is an optical disc storage device, stands for **Digital Video Display or Digital Versatile Disc**. It has the same size as a CD but can store a larger amount of data than a compact disc. It was developed in **1995** by Sony, Panasonic, Toshiba and Philips four electronics companies. DVD drives are divided into three types, such as DVD ROM (Read Only Memory), **DVD R** (Recordable) and **DVD RW** (Rewritable or Erasable). It can store multiple data formats like audio, videos, images, software, operating system, etc. The storing capacity of data in DVD is 4.7 GB to 17 GB.

Blu Ray Disc (BD)

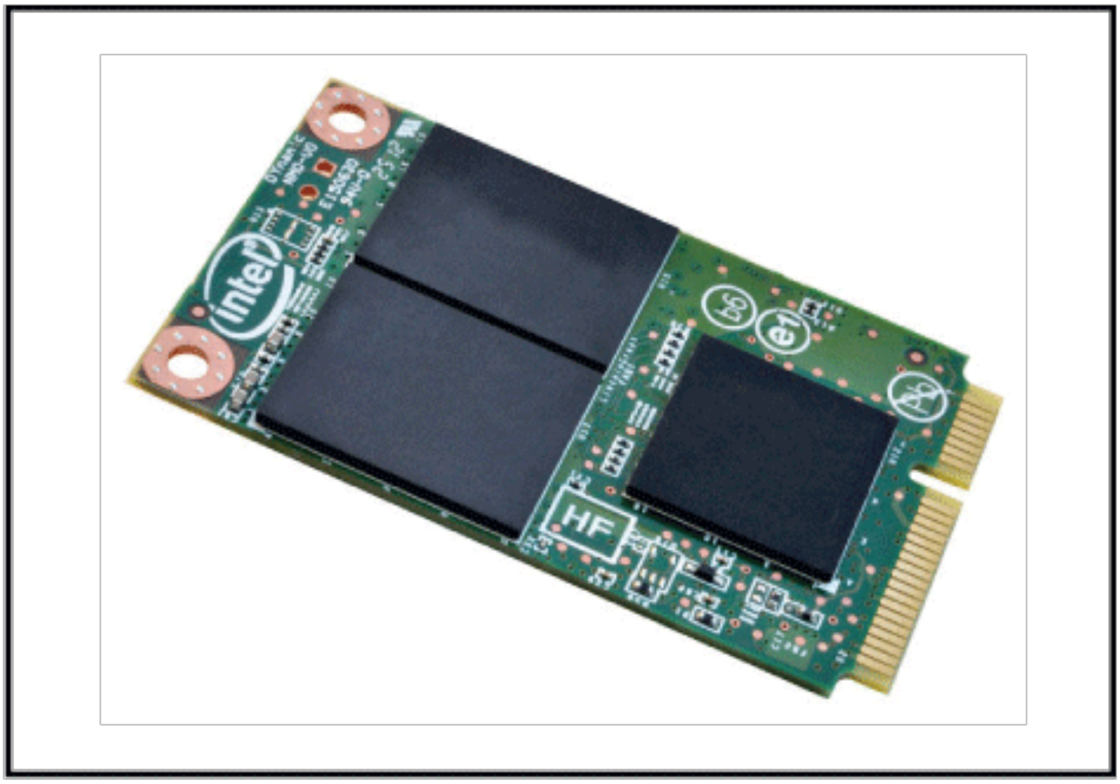
Blu Ray is an Optical disc storage device used to store a large amount of data or high definition of video recording and playing other media files. It uses laser technology to read the stored data of the Blu-ray Disk. It can store more data at a greater density as compared to CD/DVD. For example, compact discs allow us to store 700 MB of data, and in DVDs, it provides up to 8 GB of storage capacity, while Blu-ray Discs provide 28 GB of space to store data.

Pen Drive

A pen drive is a portable device used to permanently store data and is also known as a USB flash drive. It is commonly used to store and transfer the data connected to a computer using a USB port. It does not have any moveable part to store the data; it uses an integrated circuit chip that stores the data. It allows the users to store and transfer data like audio, videos, images, etc. from one computer to any USB pen drive. The storing capacity of pen drives from 64 MB to 128 GB or more.

Cache Memory

It is a small-sized chip-based computer memory that lies between the CPU and the main memory. It is a faster, high performance and temporary memory to enhance the performance of the CPU. It stores all the data and instructions that are often used by computer CPUs. It also reduces the access time of data from the main memory. It is faster than the main memory, and sometimes, it is also called CPU memory because it is very close to the CPU chip. The following are the levels of cache memory.



1. **L1 Cache:** The L1 cache is also known as the onboard, internal, or primary cache. It is built with the help of the CPU. Its speed is very high, and the size of the L1 cache varies from 8 KB to 128 KB.
2. **L2 Cache:** It is also known as external or secondary cache, which requires fast access time to store temporary data. It is built into a separate chip in a motherboard, not built into the CPU like the L1 level. The size of the L2 cache may be 128 KB to 1 MB.
3. **L3 Cache:** L3 cache levels are generally used with high performance and capacity of the computer. It is built into a motherboard. Its speed is very slow, and the maximum size up to 8 MB.

Advantages of Cache Memory

1. Cache memory is the faster memory as compared to the main memory.
2. It stores all data and instructions that are repeatedly used by the CPU for improving the performance of a computer.
3. The access time of data is less than the main memory.

Disadvantage of Cache Memory

1. It is very costly as compared to the Main memory and the Secondary memory.
2. It has limited storage capacity.

Register Memory
The register memory is a temporary storage area for storing and transferring the data and the instructions to a computer. It is the smallest and fastest memory of a computer. It is a part of computer memory located in the CPU as the form of registers. The register memory is 16, 32 and 64 bits in size. It temporarily stores data instructions and the address of the memory that is repeatedly used to provide faster response to the CPU.
Primary Vs. Secondary Memory

Primary Memory	Secondary Memory
It is also known as temporary memory.	It is also known as a permanent memory.
Data can be access directly by the processor or CPU.	Data cannot be accessed directly by the I/O processor or CPU.
Stored data can be a volatile or non-volatile memory.	The nature of secondary memory is always non-volatile.
It is more costly than secondary memory.	It is less costly than primary memory.
It is a faster memory.	It is a slower memory.
It has limited storage capacity.	It has a large storage capacity.
It required the power to retain the data in primary memory.	It does not require power to retain the data in secondary memory.
Examples of primary memory are RAM, ROM, Registers, EPROM, PROM and cache memory.	Examples of secondary memory are CD, DVD, HDD, magnetic tapes, flash disks, pen drive, etc.

FAQs on the classification of memory
Here are a few frequently requested questions (FAQs) on the type of memory:

Q1: What are the main types of memory in a computer system?
A1: There are frequently two kinds of memory in a computer system: **Primary memory** (RAM) and **Secondary memory** (storage devices like hard drives and SSDs).

Q2: How is the primary memory categorized?
A2: Primary memory, or RAM (*Random Access Memory*), is assessed

into two predominant types: volatile memory (loses information while electricity is off, e.g., RAM) and non-volatile memory (maintains records although power is off, e.g., ROM).

Q3: What is the distinction between RAM and ROM?
A3: RAM (Random Access Memory) is volatile and used for transient record storage, even as ROM (Read-Only Memory) is non-volatile and normally stores firmware or everlasting commands.

Q4: How is secondary memory categorized?
A4: Secondary memory is categorized based totally on the generation used, together with HDDs (Hard Disk Drives), SSDs (Solid State Drives), and visual storage (CDs, DVDs).

Q5: What is the reason for cache memory?
A5: Cache memory is a small-sized kind of volatile PC memory that provides high-speed information access to a processor and stores often used computer programs, applications, and facts.

Q6: How is memory categorized primarily based on access to time?
A6: Memory can be categorized into two classes primarily based on getting proper access to time: Random Access Memory (RAM), which affords speedy access to any storage place, and Sequential Access Memory (SAM), which calls for having access to data in a difficult and fast, linear collection.

Q7: What is virtual memory?
A7: Virtual memory is a memory manipulation functionality of an OS that makes use of hardware and software to permit a computer to make amends for physical memory shortages by briefly shifting data from RAM to disk memory.

Q8: What is the function of registers in computer memory?
A8: Registers are small, excessive-velocity storage locations within the CPU that hold data for short-term access via the processor. They are used to keep intermediate data for the duration of processing.

Q9: How is memory categorized based totally on the data storage unit?
A9: Memory may be categorized into **bits, bytes, kilobytes, megabytes, gigabytes, terabytes,** and so forth, based totally on the

amount of information it can keep.

Q10: What is the concept of cache levels in processors?
A10: Processors regularly have more than one tier of cache (L1, L2, and sometimes L3), each with various sizes and speeds. L1 is the smallest but quickest, and as you pass to better stages, the scale increases while speed decreases.
These FAQs cover diverse elements of memory type, from sorts of memory to precise info about primary and secondary memory.

Conclusion
In short, the type of memory in a PC device is a comprehensive categorization that encompasses numerous kinds and factors. Access time and data storage devices are also used for sophistication. Cache memory plays a crucial role in providing excessive-tempo access to frequently used records. Virtual memory is utilized to manipulate short-term storage shortages.

Boolean Algebra

Last Updated : 20 Aug, 2024
Boolean algebra is a type of algebra that is created by operating the binary system. In the year 1854, George Boole, an English mathematician, proposed this algebra. This is a variant of Aristotle's propositional logic that uses the symbols 0 and 1, or True and False. Boolean algebra is concerned with binary variables and logic operations.
Boolean Algebra is fundamental in the development of digital electronics systems as they all use the concept of **Boolean Algebra** to execute commands. Apart from digital electronics this algebra also finds its application in Set Theory, Statistics, and other branches of mathematics.
In this article, we will learn about, **basic Boolean operations, Boolean expressions, Truth Tables, Boolean laws, and others in detail.**
Table of Content

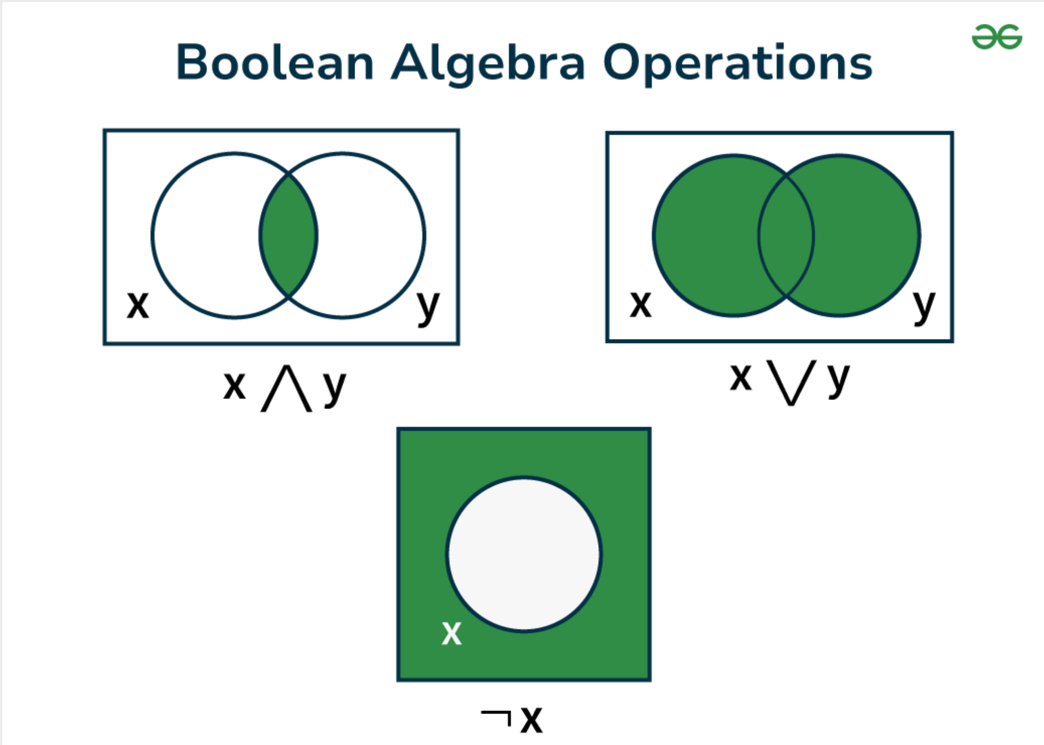
What is Boolean Algebra?

Boolean Algebra is a branch of algebra that deals with boolean values —true and false. It is fundamental to digital logic design and computer science, providing a mathematical framework for describing logical operations and expressions

Boolean Algebra Operations

Various operations are used in Boolean algebra but the basic operations that form the base of Boolean Algebra are.

- ***Negation*** *or NOT Operation*
- ***Conjunction*** *or AND Operation*
- ***Disjunction*** *or OR Operation*



Boolean Algebra Expression

Check: **Basics of Boolean Algebra in Digital Electronics**

These operations have their own symbols and precedence and the table added below shows the symbol and the precedence of these operators.

Operator	Symbol	Precedence
NOT	' (or) ¬	First
AND	. (or) ∧	Second
OR	+ (or) ∨	Third

We can easily define these operations using two boolean variables. Let's take two boolean variables A and B that can have any of the two values 0 or 1, i.e. they can be either OFF or ON. Then these operations are explained as,

Negation or NOT Operation

Using the NOT operation reverse the value of the Boolean variable from 0 to 1 or vice-versa. This can be understood as:

- *If A = 1, then using NOT operation we have (A)' = 0*
- *If A = 0, then using the NOT operation we have (A)' = 1*
- *We also represent the negation operation as ~A, i.e if A = 1, ~A = 0*

Check: **Properties of Boolean Algebra**

Conjunction or AND Operation

Using the AND operation satisfies the condition if both the value of the individual variables are true and if any of the value is false then this operation gives the negative result. This can be understood as,

- *If A = True, B = True, then A . B = True*
- *If A = True, B = False, Or A = false, B = True, then A . B = False*
- *If A = False, B = False, then A . B = False*

Check: Boolean Algebraic Theorems

Disjunction (OR) Operation

Using the OR operation satisfies the condition if any value of the individual variables is true, it only gives a negative result if both the values are false. This can be understood as,

- *If A = True, B = True, then A + B = True*
- *If A = True, B = False, Or A = false, B = True, then A + B = True*
- *If A = False, B = False, then A + B = Falses*

Boolean Algebra Table

Given Below is the Expression for the Boolean Algebra

Operation	Symbol	Definition
AND Operation	· or ∧	Returns true only if both inputs are true.
OR Operation	+ or ∨	Returns true if at least one input is true.
NOT Operation	¬ or ~	Reverses the input.
XOR Operation	⊕	Returns true if exactly one input is true.
NAND Operation	↓	Returns false only if both inputs are true.
NOR Operation	↑	Returns false if at least one input is true.
XNOR Operation	↔	Returns true if both inputs are equal.

Boolean Expression and Variables

Boolean expression is an expression that produces a Boolean value when evaluated, i.e. it produces either a true value or a false value. Whereas boolean variables are variables that store Boolean numbers.

P + Q = R is a Boolean phrase in which P, Q, and R are Boolean variables that can only store two values: 0 and 1. The 0 and 1 are the synonyms for false and True and are used in Boolean Algebra, sometimes we also use "Yes" in place of True and "No" in place of False.

Thus, we can say that statements using Boolean variables and operating on Boolean operations are Boolean Expressions. Some examples of Boolean expressions are,

- *A + B = True*
- *A.B = True*
- *(A)′ = False*

Check: **Axioms of Boolean Algebra**

Boolean Algebra Terminologies

There are various terminologies related to Boolean Algebra, which are used to explain various parameters of **Boolean Algebra**. That includes,

- *Boolean Variables*
- *Boolean Function*
- *Literal*
- *Complement*
- *Truth Table*

Now, we will discuss the important terminologies of Boolean algebra in the article below,

Boolean Variables

Variables used in Boolean algebra that store the logical value of 0 and 1 are called the boolean variables. They are used to store either true or false values. Boolean variables are fundamental in representing logical states or propositions in Boolean expressions and functions.

Boolean Function

A function of the Boolean Algebra that is formed by the use of Boolean variables and Boolean operators is called the Boolean function. It is formed by combining Boolean variables and logical expressions such as AND, OR, and NOT. It is used to model logical relationships, conditions, or operations.

Literal

A variable or the complement of the variable in Boolean Algebra is called the Literal. Literals are the basic building blocks of the boolean expressions and functions. They represent the operands in logical operations.

Complement

The inverse of the Boolean variable is called the complement of the variable. The complement of 0 is 1 and the complement of 1 is 0. It is represented by ' or (¬) over the variable. Complements are used to represent logical negations in Boolean expressions and functions.

Truth Table

Table containing all the possible values of the logical variables and the combination of the variable along with the given operation is called the truth table. The number of rows in the truth table depends on the total Boolean variables used in that function. It is given by using the formula,

Number of Rows in Truth Table = 2^n
where "n" is the number of Boolean variables used.
Check:

Truth Tables in Boolean Algebra

A truth table represents all the combinations of input values and outputs in a tabular manner. All the possibilities of the input and output are shown in it and hence the name truth table. In logic problems, truth tables are commonly used to represent various cases. T or 1 denotes 'True' & F or 0 denotes 'False' in the truth table.
Example: Draw the truth table of the conditions A + B and A.B where A and b are boolean variables.
Solution:
The required Truth Table is,

A	B	X = A + B	Y = A.B
T	T	T	T
T	F	T	F
F	T	T	F
F	F	F	F

Boolean Algebra Rules
In Boolean Algebra there are different fundamental rules for logical expression.

- **Binary Representation:** In Boolean Algebra the variables can have only two values either 0 or 1 where 0 represents Low and 1 represents high. These variables represents logical states of the system.
- **Complement Representation:** The complement of the variables is represented by (¬) or (') over the variable. This indicates logical negation or inversion of the variable's value. So Complement of variable A can be represented by *A*,if the value of A=0 then its complement is 1.

- **OR Operation:** The OR operation is represented by (+) between the Variables. OR operation returns true if at least one of the operands is true. For Examples let us take three variables A,B,C the OR operation can be represented as A + B + C.
- **AND Operation:** The AND Operation is denoted by (.) between the Variables. AND operation returns true only if all the operands are true. For Examples let us take three variables A,B,C the AND operation can be represented A.B.C or ABC.

Laws for Boolean Algebra

The basic laws of the Boolean Algebra are added in the table added below,

Law	OR form	AND form
Identity Law	P + 0 = P	P.1 = P
Idempotent Law	P + P = P	P.P = P
Commutative Law	P + Q = Q + P	P.Q = Q.P
Associative Law	P + (Q + R) = (P + Q) + R	P.(Q.R) = (P.Q).R
Distributive Law	P + QR = (P + Q).(P + R)	P.(Q + R) = P.Q + P.R
Inversion Law	(A′)′ = A	(A′)′ = A
De Morgan's Law	(P + Q)′ = (P)′.(Q)′	(P.Q)′ = (P)′ + (Q)′

Let's learn about these laws in detail.

Identity Law

In the Boolean Algebra, we have identity elements for both AND(.) and OR(+) operations. The identity law state that in boolean algebra we have such variables that on operating with AND and OR operation we get the same result, i.e.

- *A + 0 = A*
- *A.1 = A*

Commutative Law

Binary variables in Boolean Algebra follow the commutative law. This law states that operating boolean variables A and B is similar to operating boolean variables B and A. That is,

- *A. B = B. A*
- *A + B = B + A*

Associative Law

Associative law state that the order of performing Boolean operator is illogical as their result is always the same. This can be understood as,

- *(A × B) × C = A . (B . C)*
- *(A + B) + C = A + (B + C)*

Distributive Law

Boolean Variables also follow the distributive law and the expression for Distributive law is given as:

- *A × (B + C) = (A . B) + (A . C)*

Inversion Law

Inversion law is the unique law of Boolean algebra this law states that, the complement of the complement of any number is the number itself.

- *(A’)’ = A*

Apart from these other laws are mentioned below:

AND Law

AND law of the Boolean algebra uses AND operator and the AND law is,

- *A . 0 = 0*
- *A × 1 = A*
- *A × A = A*

OR Law

OR law of the Boolean algebra uses OR operator and the OR law is,

- *A + 0 = A*
- *A + 1 = 1*
- *A + A = A*

De Morgan’s Laws are also called **De morgan’s Theorem**. They are the most important laws in **Boolean Algebra** and these are added below under the heading Boolean Algebra Theorem

Boolean Algebra Theorems

There are two basic theorems of great importance in Boolean Algebra, which are De Morgan’s First Laws, and De Morgan’s Second Laws. These are also called De Morgan’s Theorems. Now let’s learn about both in detail.

De Morgan’s First laws

De Morgan’s Law states that **t**he complement of the product (AND) of two Boolean variables (or expressions) is equal to the sum (OR) of the complement of each Boolean variable (or expression).

(P.Q)’ = (P)’ + (Q)’

The truth table for the same is given below:

P	Q	(P)′	(Q)′	(P.Q)′	(P)′ + (Q)′
T	T	F	F	F	F
T	F	F	T	T	T
F	T	T	F	T	T
F	F	T	T	T	T

We can clearly see that truth values for (P.Q)′ are equal to truth values for (P)′ + (Q)′, corresponding to the same input. Thus, De Morgan's First Law is true.

De Morgan's Second laws

Statement: The Complement of the sum (OR) of two Boolean variables (or expressions) is equal to the product(AND) of the complement of each Boolean variable (or expression).

(P + Q)′ = (P)′.(Q)′

Proof:

The truth table for the same is given below:

P	Q	(P)′	(Q)′	(P + Q)′	(P)′.(Q)′
T	T	F	F	F	F
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

We can clearly see that truth values for (P + Q)′ are equal to truth values for (P)′.(Q)′, corresponding to the same input. Thus, De Morgan's Second Law is true.

Articles related to Boolean Algebra:

- *Properties of Boolean Algebra*
- *Principle of Mathematical Induction*
- *Logic Gates*

Solved Examples on Boolean Algebra

Draw Truth Table for P + P.Q = P

Solution:

The truth table for P + P.Q = P

P	*Q*	*P.Q*	*P + P.Q*
T	*T*	*T*	*T*
T	*F*	*F*	*T*
F	*T*	*F*	*F*
F	*F*	*F*	*F*

In the truth table, we can see that the truth values for P + P.Q is *exactly the same as P.*

Draw Truth Table for P.Q + P + Q

Solution:

The truth table for P.Q + P + Q

P	***Q***	***P.Q***	***P.Q + P + Q***
T	*T*	*T*	*T*
T	*F*	*F*	*T*
F	*T*	*F*	*T*
F	*F*	*F*	*F*

Solve $A+B \cdot C$

Solution:

Using De Morgan's Law

$A+B \times C=A.(B+C)$

Using Distributive Law

$A \times (B+C)=A.B+A.C$

So, the simplified expression for the given equation

$A \times (B+C)=A.B+A.C$

Modes of Transfer (DMA, I/o)

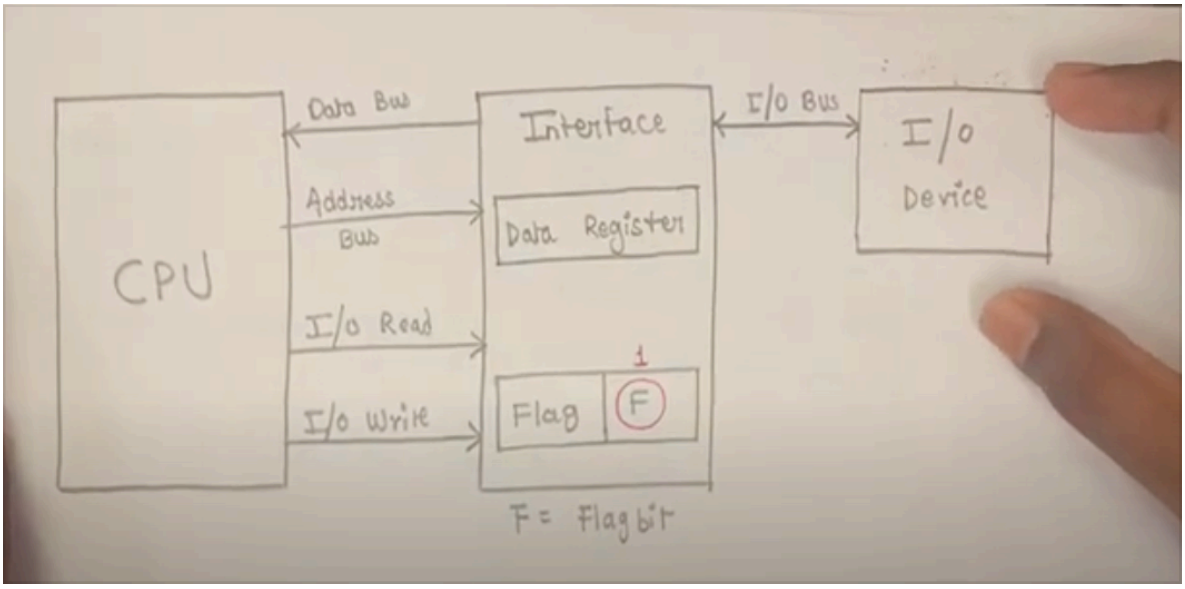
- Before delving into the modes of transfer, it's essential to understand that when data is transferred between the CPU and I/O devices, the CPU primarily handles execution and stores some data in its registers. However, the actual data transfer to I/O devices is facilitated through the assistance of memory. Therefore, memory plays a crucial role, serving as an intermediate step when performing data transfers from I/O devices. In this process, the CPU acts as an intermediary.
- Various modes exist for transferring data between the central computer and I/O devices, and understanding these modes is integral to comprehending the intricacies of data transfer.
- There are three primary modes through which data is transferred from peripheral devices to the CPU:
    1. **Programmed I/O:** In this mode, the CPU serves as an intermediate link in the data transfer process.
    2. **Interrupt-Initiated I/O:** Similar to Programmed I/O, this mode involves the CPU as an intermediary in the data transfer.

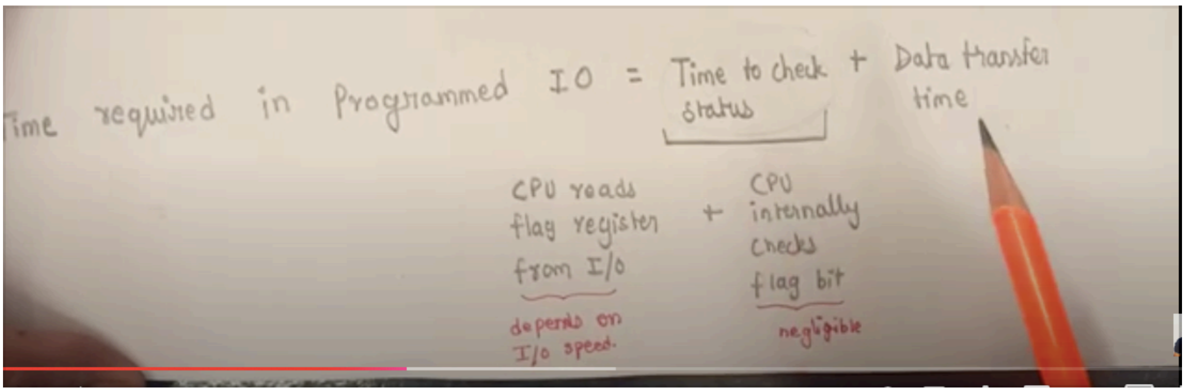
3. **Direct Memory Access (DMA):** In DMA mode, the CPU is not an intermediate participant, allowing for more efficient data transfer between peripheral devices and memory.

- While Programmed I/O and Interrupt-Initiated I/O involve the CPU as an intermediate step, Direct Memory Access (DMA) bypasses the CPU, streamlining the data transfer process.

Programmed I/O

- **There is no provision or rule that io devices can inform cpu about data transfer.**
- **Io devices set its own status and waits for cpu to read it.**
- **Cpu runs program periodically and checks the status of each devices one by one.**
- **If any devices have status to transfer the data cpu will starts transferring their data one by one**
- **It works on pricipal of polling**

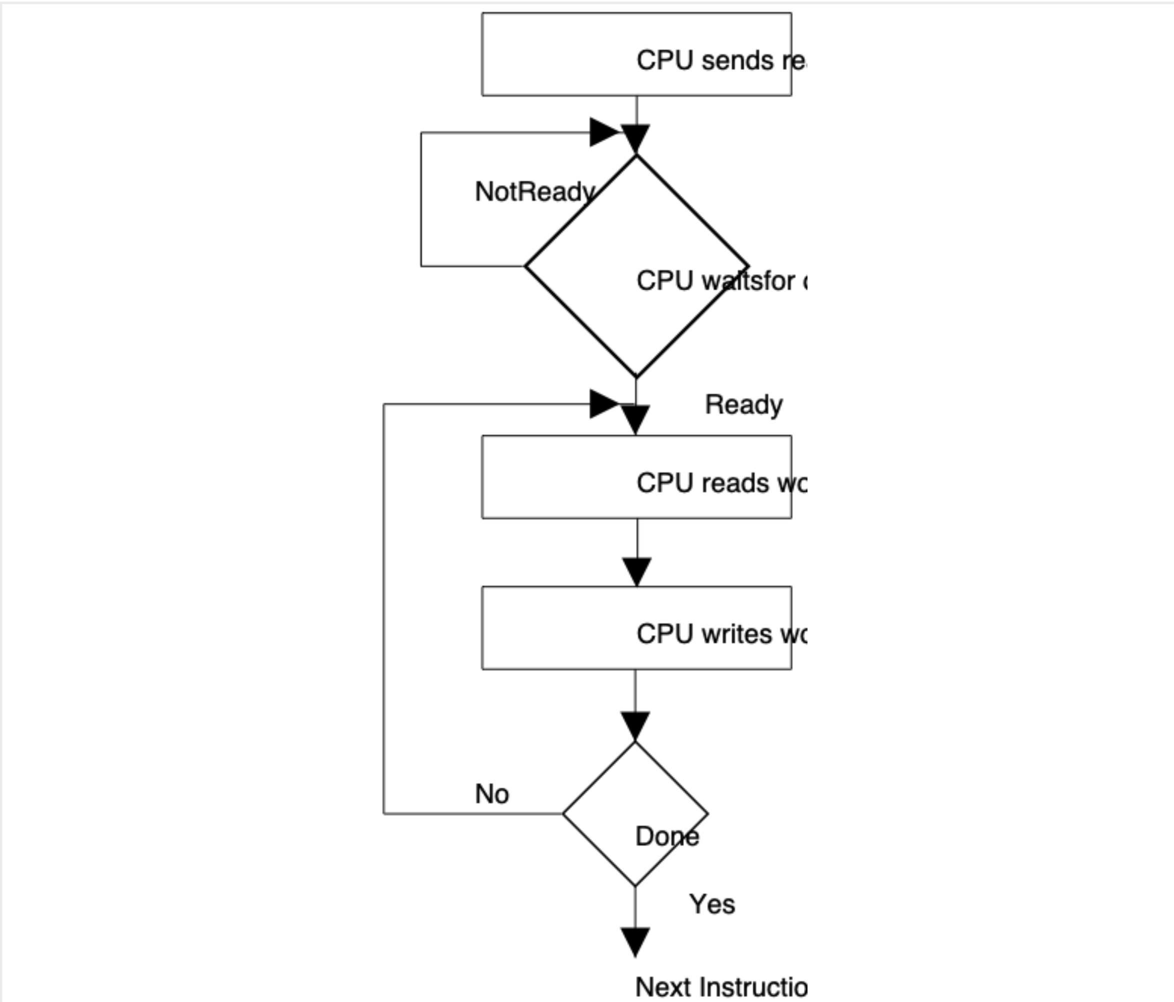




- Programmed I/O is utilized when there is a need to transfer data between the CPU and I/O, and the data transfer is managed through program instructions.

- In the context of computer programming languages such as C++, Programmed I/O operations result from I/O instructions embedded in the computer program. Examples of such instructions include those for input (e.g., cin) and output (e.g., cout).
- Each data transfer in Programmed I/O is initiated by an I/O instruction within the program, typically to access registers or memory on a specific device.
- Executing data transfers under program control demands continuous monitoring of I/O devices by the CPU.

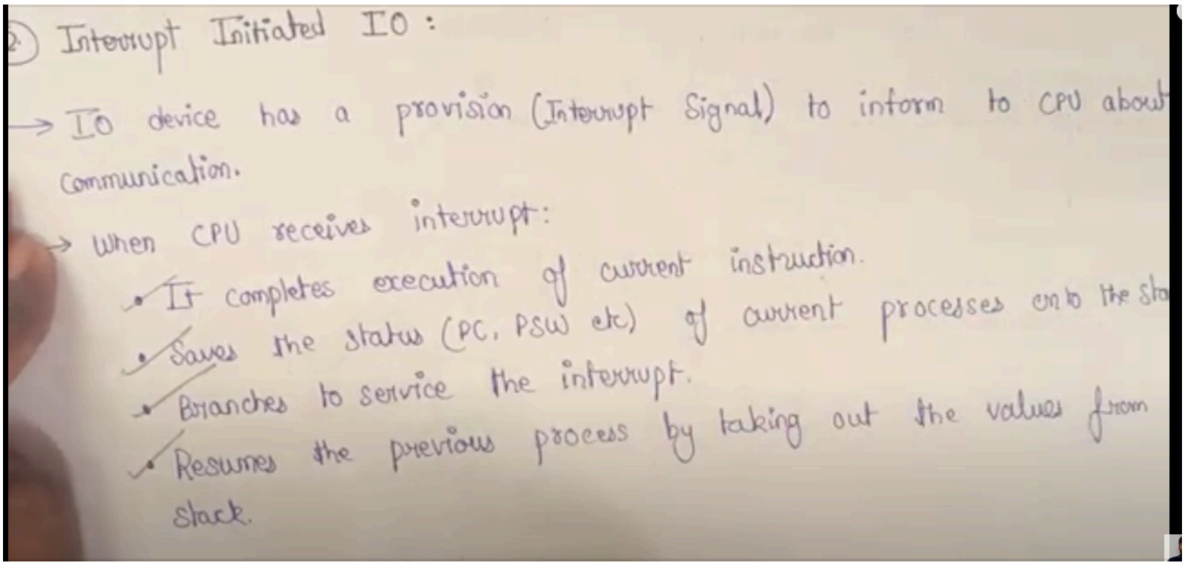
The diagram below illustrates the process of Programmed I/O:



In Programmed I/O, the CPU initiates a request and then remains in a program loop (polling) until the I/O device signals its readiness for data transfer. Importantly, the I/O device takes no further action to interrupt the CPU; it does not independently interrupt the CPU.

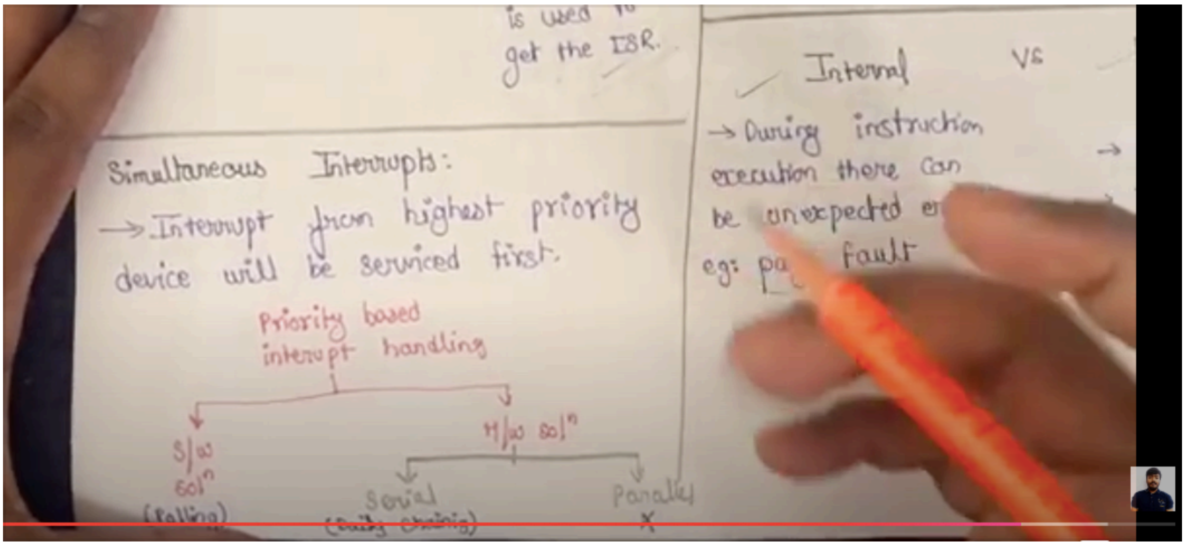
- Disadvantages:
    - Programmed I/O can be time-consuming as it keeps the CPU unnecessarily busy. To address this issue, interrupt facilities are often employed to enhance efficiency.

Interrupt Initiated I/O



There are bunch of types in here :
Vector Interrupt -> Here Io device sends the interrupt signal with proper address from where to transfer the data
Non Vector Interrupt -> Here IO device only sends interrupt signal without address, cpu figure out the address using ISR routine.

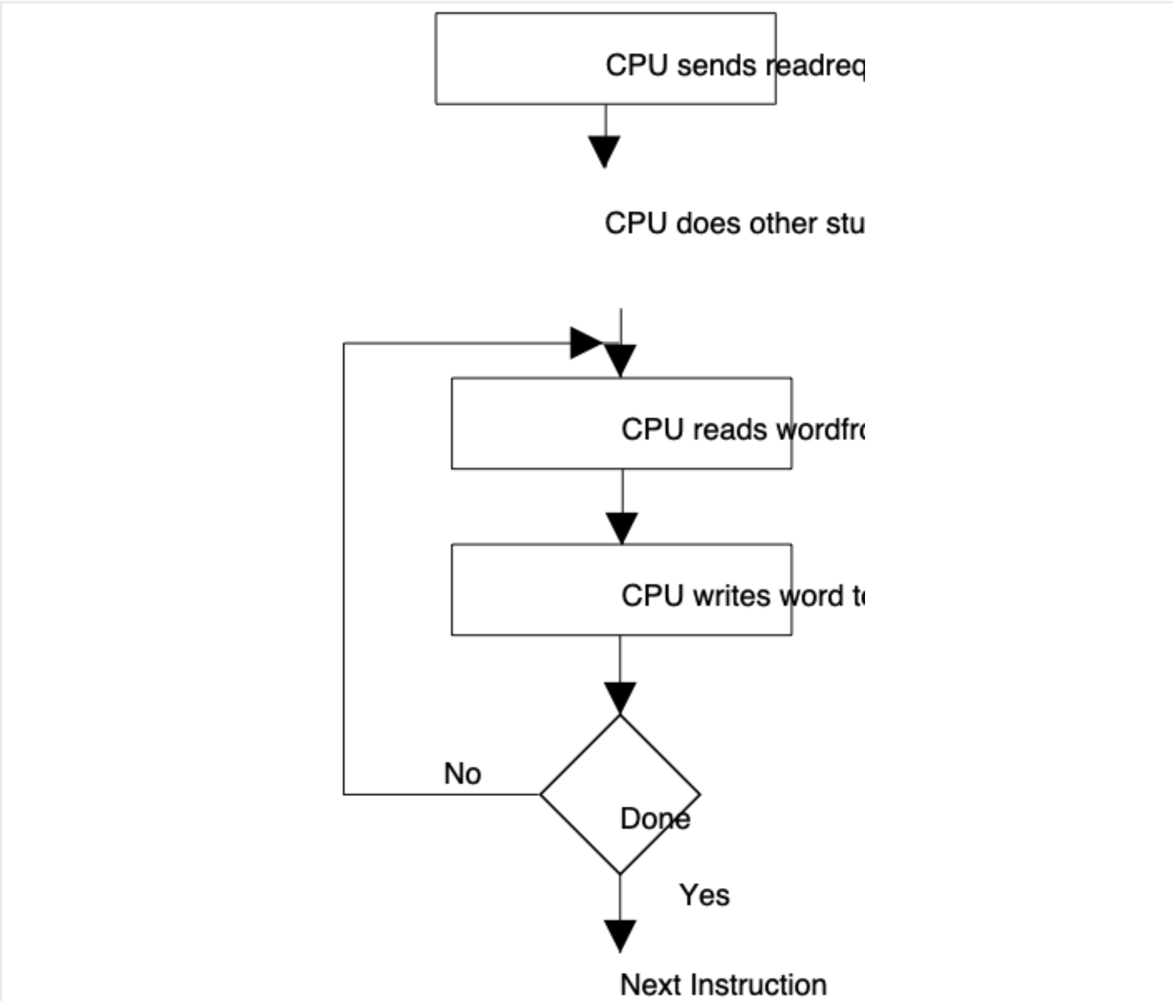
Maskable -> Cpu can either reject or accept these type of interrupt signals
Non - Maskable -> Always accepts



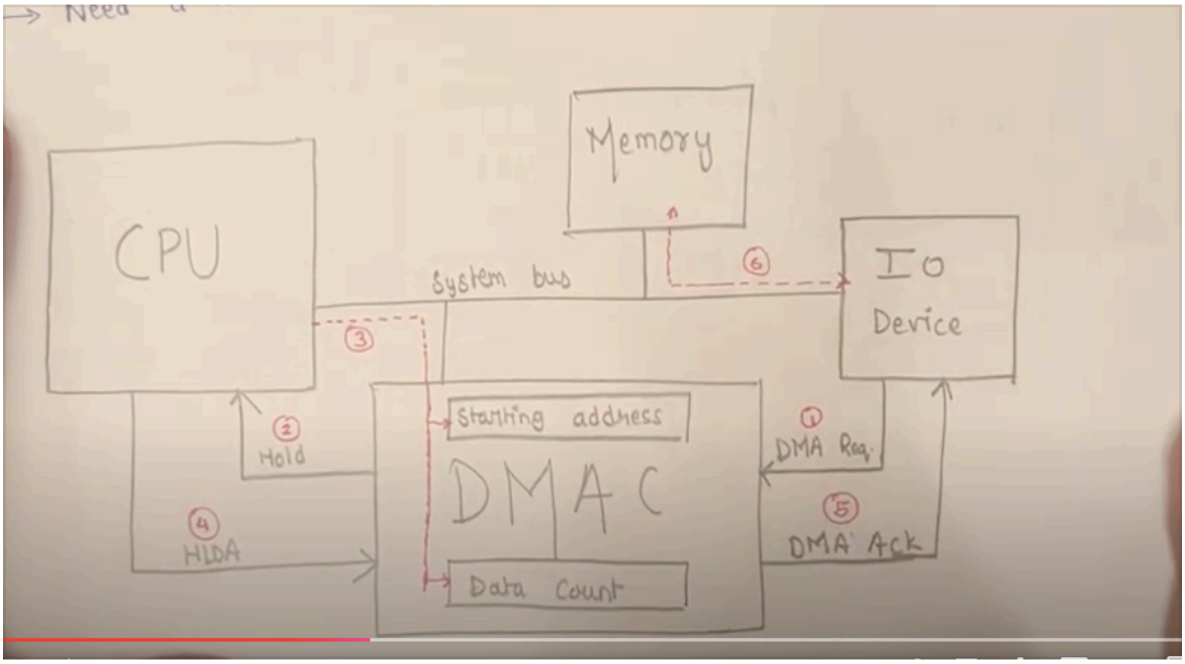
- Interrupt Initiated I/O was introduced to address the polling-related issue present in Programmed I/O.
- An interrupt is a high-priority signal, generated either by an external device or some software, designed to immediately capture the CPU's attention. The use of interrupts aims to

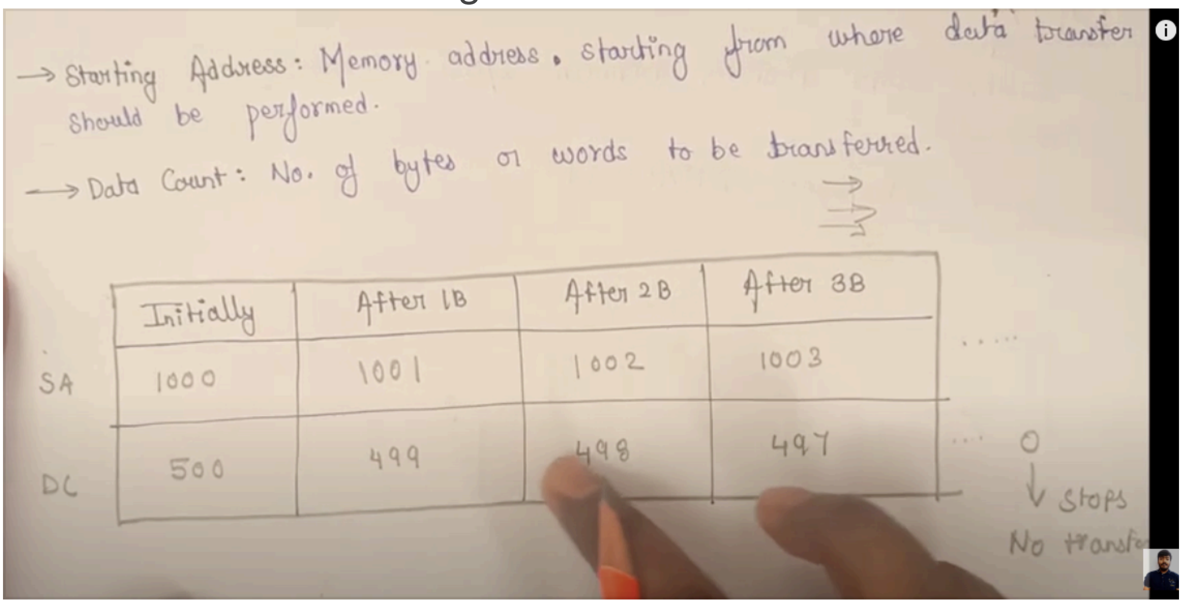
eliminate the waiting period inherent in Programmed I/O.

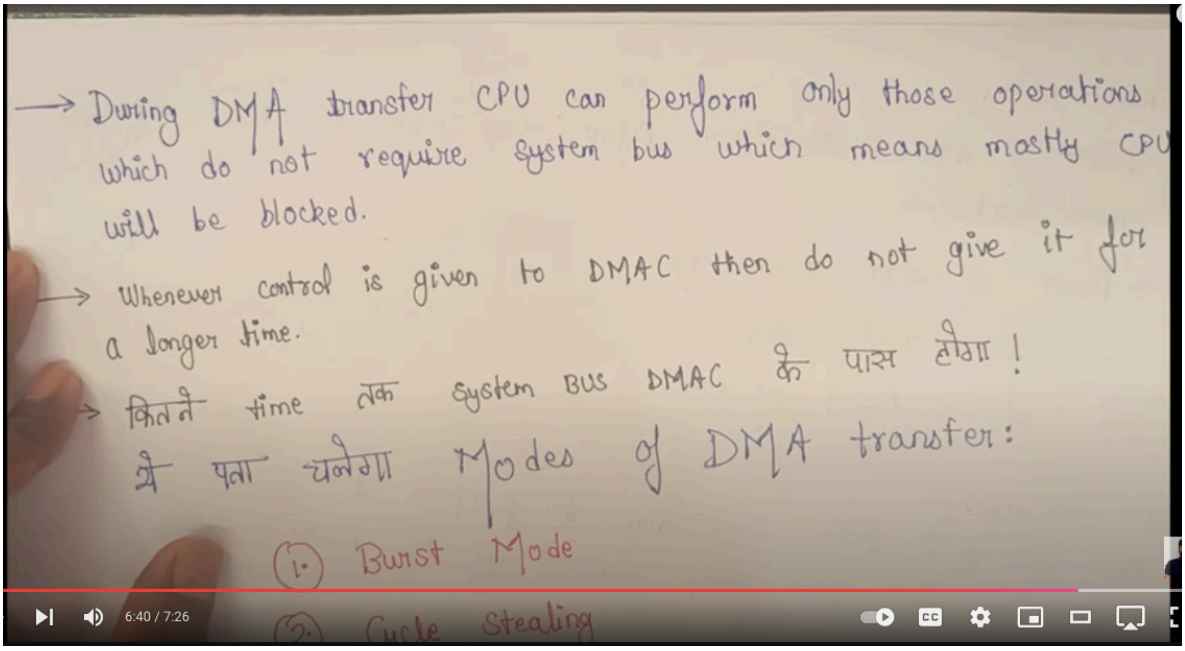
- In Interrupt Initiated I/O, instead of the CPU continuously monitoring, the interface is informed to issue an interrupt request signal when data becomes available from the device.
- Meanwhile, the CPU proceeds to execute other programs while the interface keeps monitoring the device.
- When the device is ready for data transfer, it generates an interrupt request.
- Upon detecting the external interrupt signal, the CPU interrupts its current task, processes the I/O data transfer, and then resumes the original task it was performing.



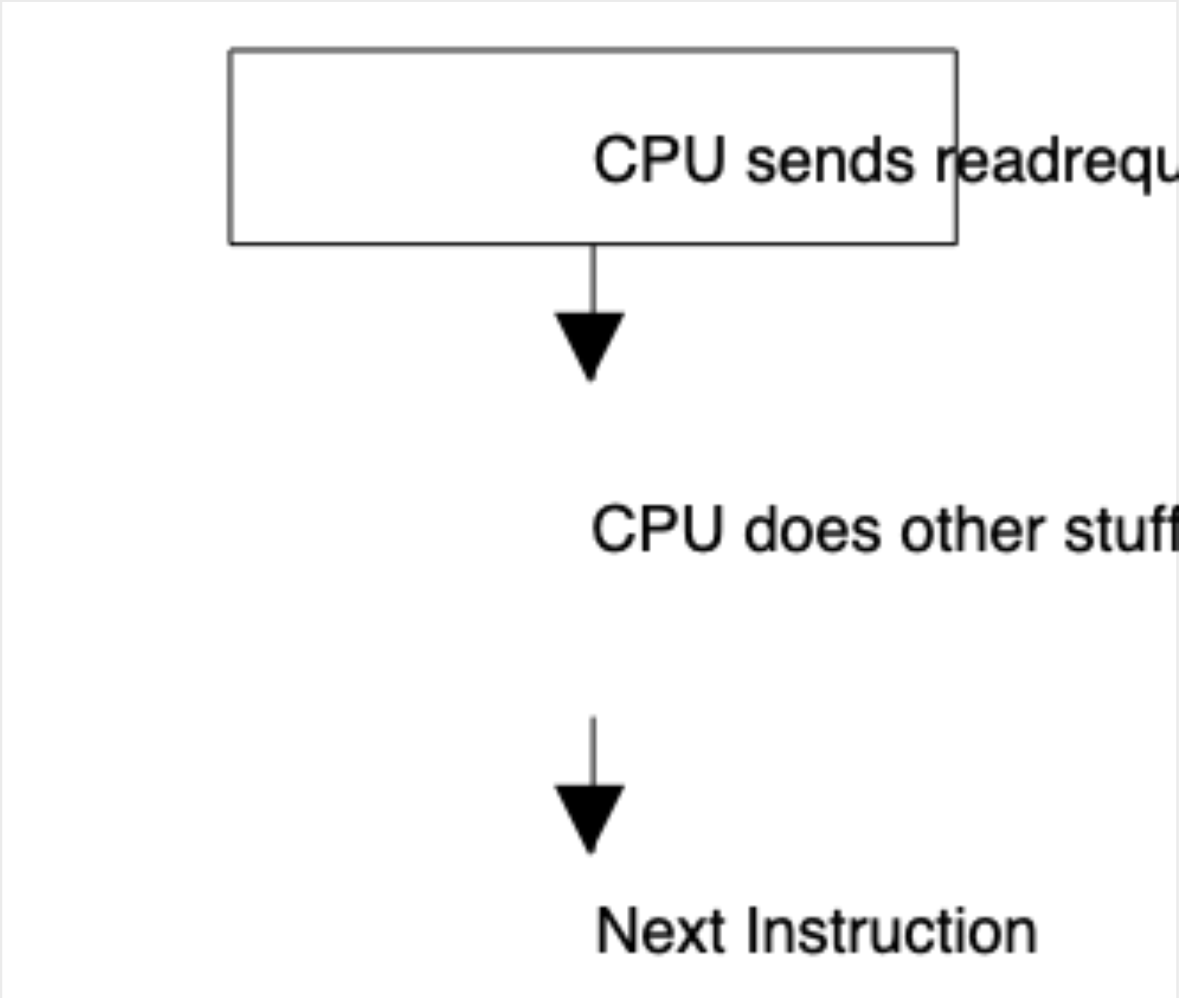
Direct Memory Access (DMA)



1. IO device sends request to DMAC (controller) device
2. DMAC requests to the cpu for system bus access
3. If cpu accepts the req it will send starting address & data count to the DMAC.
4. Now cpu send the success acknowledgement to the DMAC.
5. DMAC sends the ack to the IO device
6. Io device starts tranferring the data
7. 



- DMA is employed when large blocks of data need to be transferred between the CPU and I/O devices, rendering Programmed or Interrupt Initiated I/O less efficient.
- For high-speed transfers of substantial data blocks between external devices and main memory, the DMA approach is often utilized.
- In other transfer modes, memory is accessed indirectly through the CPU. However, when transferring significant data blocks and frequently utilizing the CPU before memory access, this process becomes time-consuming. DMA addresses this by allowing direct communication between I/O devices and memory, minimizing CPU intervention.
- DMA permits data transfer directly between the I/O device and main memory with minimal CPU involvement.
- In DMA, the CPU grants the I/O interface the authority to read from or write to memory without direct CPU intervention.
- The DMA controller autonomously manages data transfer between main memory and the I/O device.
- The CPU is only involved at the beginning and end of the transfer and is interrupted only after the entire block has been successfully transferred.



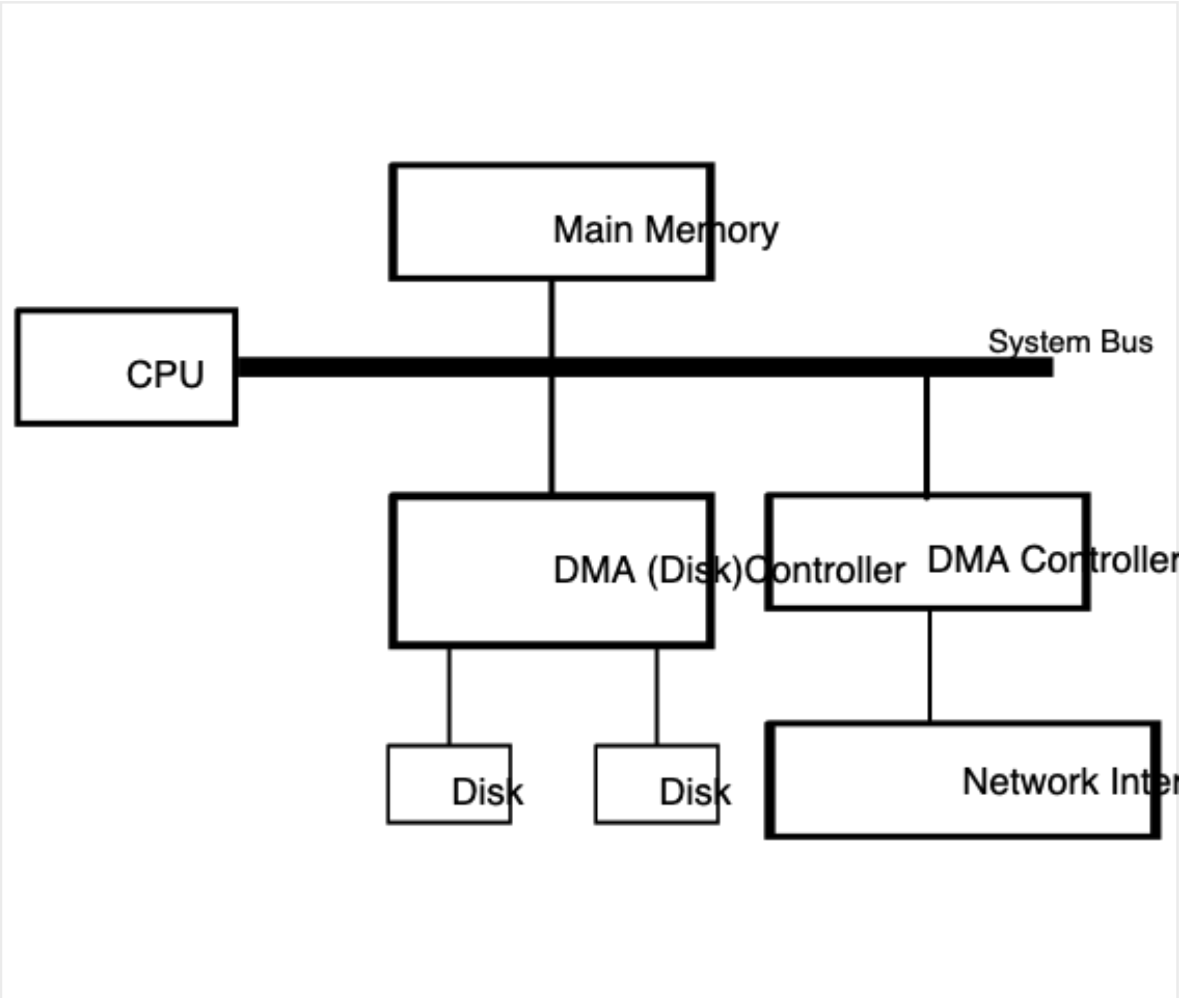
- The process involves the CPU initiating the DMA controller to transfer data between a device and main memory, allowing the CPU to proceed with other tasks.
- The DMA controller issues a request to the relevant I/O device, manages the data transfer between the device and main memory, and waits for its completion.
- Upon the conclusion of the data transfer, the DMA controller interrupts the CPU.

DMA Controller

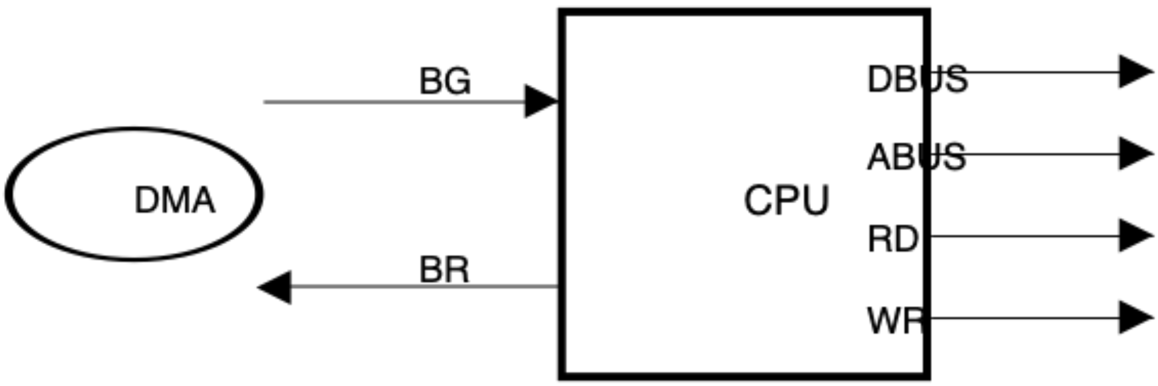
- DMA enables I/O devices to transfer data directly to or from main memory without requiring CPU intervention, effectively bypassing the CPU.
- Between I/O devices and the CPU, there exists an interface, but between memory and I/O devices, the DMA controller (or DMA channel) acts as the intermediary, creating a channel between main memory and I/O devices. Devices such as magnetic disks, USB drives, network cards, graphics cards, and sound cards, when connected via DMA, can achieve

faster data transfer rates.

- DMA finds applications in systems utilizing multicore architectures, particularly in scenarios where intrachip data transfers are required.
- Within the system bus, comprising address lines, data lines, and control lines connected to the CPU, memory, and DMA controller, during DMA transfers, the CPU is temporarily disabled. While the CPU typically controls the system bus, in DMA transfers, the DMA controller temporarily borrows control of the system bus from the CPU to facilitate efficient data transfer between I/O devices and memory.



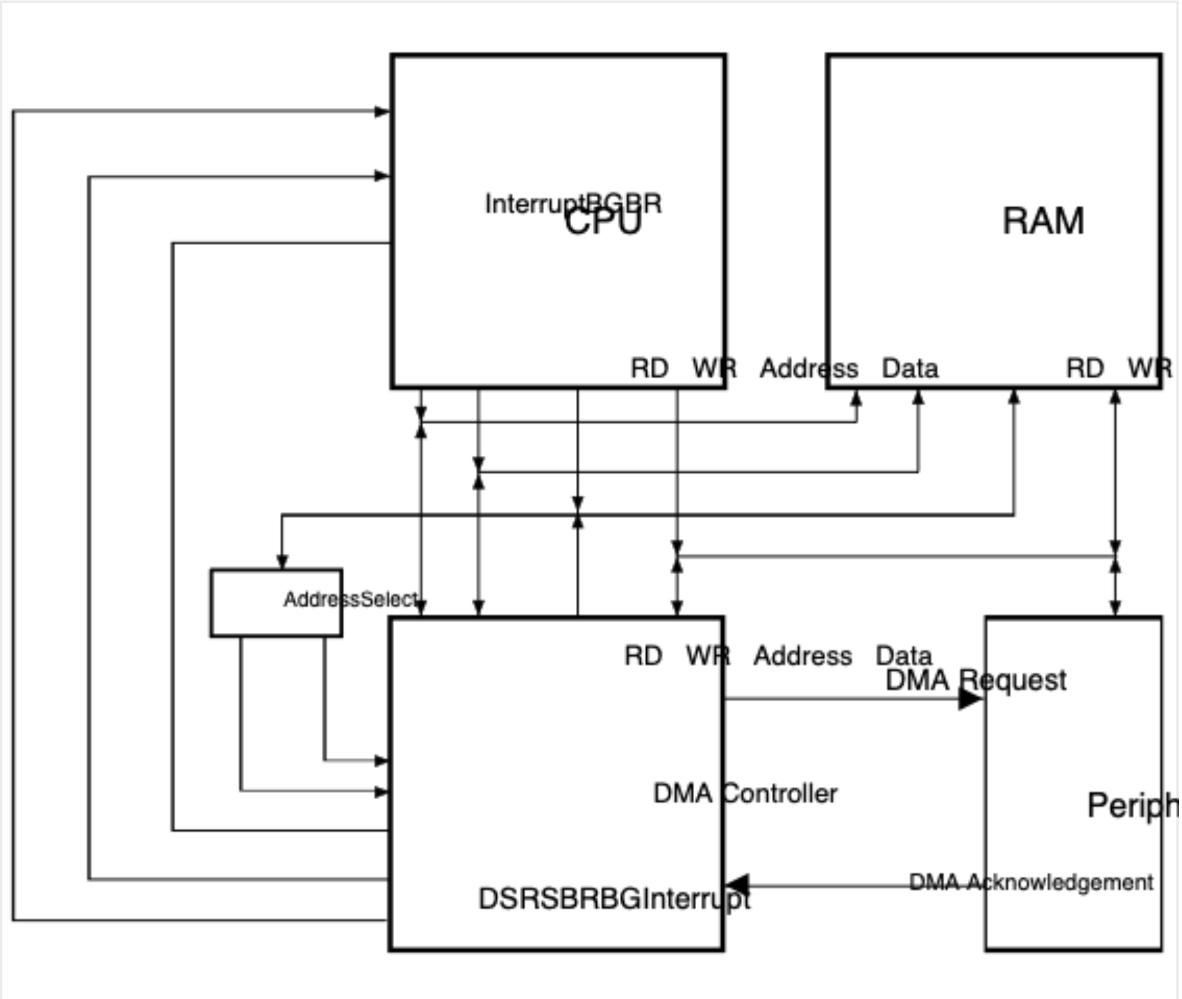
How DMA makes CPU to go in idle state?



- To facilitate DMA's control over the bus system from the CPU, two signals are employed: Bus Request (BR) and Bus Grant (BG).
- When DMA desires full control of the bus system, it initiates the process by sending a Bus Request (BR) signal through the bus request line.
- Upon receiving the Bus Request (BR) signal, the CPU interrupts its ongoing tasks, relinquishes control of all three components—data lines, address lines, and control lines—and enters a high-impedance state. In this state, the bus behaves like an open circuit, disabling all signals and buses.
- To signal to the DMA that control has been transferred, the CPU sends a Bus Grant (BG) signal through the bus grant line, indicating that the DMA now has authority over the buses. This communication allows the DMA to use the buses to transfer data directly to memory.
- Essentially, the two main signals used in this process are Bus Request (BR) and Bus Grant (BG).

DMA Working

When an I/O device wishes to transfer data with main memory, the following steps outline the DMA process:



In the diagram above, we have a detailed breakdown of the components involved in the Direct Memory Access (DMA) process:

1. **DS (DMA Select):** The processor sets DS = 1 to activate DMA, initiating the DMA process.
2. **RS (Register Select):** The CPU uses this signal to select DMA registers for storing values, such as the starting address and the number of words to be transferred.
3. **RD (Read) & WR (Write):** These signals, Iri is used for reading and writing purposes during the DMA operation.
4. **BR (Bus Request):** DMA employs this line to send a request to the processor to release the BUS system, indicating its need for control over the system bus.
5. **BG (Bus Grant):** When the processor relinquishes control of the bus to DMA, it sets BG = 1, signifying that the bus is now granted to the DMA controller.
6. **Interrupt:** DMA uses this line to send a signal when data transfer is completed. The processor can also use this line to

check whether data transfer has been successfully completed.
7. **DMA Request:** I/O devices utilize this line to send a request to the DMA controller, signaling the need for data transfer.
8. **DMA Acknowledgement:** The DMA controller responds to I/O devices through this line, acknowledging the receipt of the request and preparing for data transfer.
9. **Address Register:** The processor stores the starting address of data in this register, providing the necessary information for the DMA controller to locate the data.
10. **Word Count Register:** The processor stores the total number of words to be transferred in this register, allowing the DMA controller to determine the extent of the data transfer operation.
11. **Control Register:** The processor stores control signals in this register, dictating various aspects of the DMA operation, such as the transfer mode and direction.
12. **Data Bus Buffer:** This component is employed to temporarily store data during the DMA transfer process, ensuring efficient and synchronized data movement.
13. **Data Bus:** DMA utilizes this bus for the actual transfer of data between memory and peripheral devices, facilitating high-speed and direct communication.

Now, let's delve into the operational sequence of the Direct Memory Access (DMA) process:

- **Initiation of Data Transfer Request:** When an I/O device wishes to transfer data, it sends a request to the DMA through the DMA request line.
- **Bus Request to Processor:** DMA, in response to the request, sends a bus request (BR = 1) to the processor, requesting it to release control of the bus system (utilizing the BR line).
- **Processor's Response:** Upon receiving the bus request, the processor stores its current work, and then transmits essential information, such as the starting address and the number of words to be transferred, to DMA. Subsequently, the processor relinquishes control of the bus system and notifies DMA by setting the BG line to 1.
- **DMA Acknowledgement and Data Transfer:** Upon receiving the BG signal, DMA activates the DMA acknowledgement line, informing the I/O device that it can commence data transfer.

The DMA controller initiates the actual data transfer process.

- **Decrementing Word Count:** With each data transfer, the value of the Word Count (WC) register is decremented by 1, keeping track of the progress of the data transfer operation.
- **Data Transfer Completion:** When the WC register reaches 0, DMA sets BR = 0 and sends an interrupt signal to the CPU, signaling the completion of the data transfer.
- **CPU's Post-Transfer Actions:** The CPU, upon receiving the interrupt, checks the WC register. Since it is now 0, the CPU sets BG = 0, reclaiming control of the bus system for its own operations.