| **SL.NO** | **Static Web Page** | **Dynamic Web Page** |
| --- | --- | --- |
| 1. | In static web pages, Pages will remain same until someone changes it manually. | In dynamic web pages, Content of pages are different for different visitors. |
| 2. | Static Web Pages are simple in terms of complexity. | Dynamic web pages are complicated. |
| 3. | In static web pages, Information are change rarely. | In dynamic web page, Information are change frequently. |
| 4. | Static Web Page takes less time for loading than dynamic web page. | Dynamic web page takes more time for loading. |
| 5. | In Static Web Pages, database is not used. | In dynamic web pages, database is used. |
| 6. | Static web pages are written in languages such as: HTML, JavaScript, CSS, etc. | Dynamic web pages are written in languages such as: CGI, AJAX, ASP, ASP.NET, etc. |
| 7. | Static web pages does not contain any application program . | Dynamic web pages contains application program for different services. |
| 8. | Static web pages require less work and cost in designing them. | Dynamic web pages require comparatively more work and cost in designing them. |

1. **algebra** is concerned with binary variables and logic operationsBoolean Algebra is fundamental in the development of digital electronics systems as tall
2. **Internal**Internal memory, also known as primary memory, stores small amounts of data that the computer can access while you're actively using it. Internal memory consists of chips connected to the motherboard and needs to be connected directly to the device in order to use it. There are two basic types of internal memory called RAM and ROM, and those have their own subsets of memory. **RAM**Random access memory (RAM) is the primary internal memory of the central processing unit (CPU). Your electronic device uses it for storing temporary data. It does this by providing applications with a place to store data you're actively using so that it can quickly access the data. The amount of RAM on your device controls its performance and speed. If you don't have enough RAM, it may process programs slowly which can affect the output and speed with which you can use the computer.RAM also has "volatile memory" because it loses the data it was storing if you turn off the device. For example, if you're on your laptop using the internet browser and your computer gets turned off, it may not have saved the web pages you were previously using because RAM only stores that information temporarily.**DRAM**Dynamic random access memory (DRAM) is one of the two specific types of RAM seen in modern devices, such as laptops, desktops, portable devices and gaming systems. It's the more affordable of the two types of RAM and produces high-capacity memory.It's made of two components, transistors and capacitors, which require a recharge every few seconds to retain its data. Like RAM, it also loses data when it loses power and has volatile memory. **SRAM**Static random access memory (SRAM) is the second type of RAM and stores data as long as there is power in the system, unlike DRAM which is refreshed much more frequently. Because it holds power longer, it's more expensive than DRAM, which typically makes it not as widely used. Users generally use SRAM for cache memory, which makes it a faster form of memory than DRAM.**. ROM**Read-only memory (ROM) is another type of primary internal memory, but unlike RAM, ROM is non-volatile and stores data permanently. It does not depend on the device to be turned on in order to save data.Instead, a programmer writes the data into individual cells using binary code, which represents text using the "1" and "0" two-symbol system. Because you cannot alter the data on ROM, you can use this type of memory for aspects that don't change, such as the boot-up of software or firmware instructions, which help a device function properly.  **PROM**Programmable read-only memory (PROM) is a type of ROM that begins as memory with no data on it. A user can write data onto the chip using a special device called a PROM programmer.Like ROM, data is permanent once a user has written it onto the chip. This type of memory may be useful to programmers who would like to create specific firmware for a chip and use it to alter the typical function of a system.**EPROM**Erasable programmable read-only memory (EPROM) is another type of ROM chip that users can write data on as well as erase old data and reprogram it. You can erase the current data by using ultraviolet (UV) light in the form of a quartz crystal window at the top of the chip.After you've erased the data, you can use the PROM programmer to reprogram it. You can only erase data from an EPROM chip a certain number of times because excessive erasing can damage the chip and make it unreliable for future use.**EEPROM**Electrically erasable programmable read-only memory (EEPROM) is the final type of non-volatile ROM chip that usually replaces the need for PROM or EPROM chips. This type of memory also allows users to erase and reprogram data onto the chip, but does so using an electric field and is much quicker at erasing data than the EPROM. In addition, you can conveniently erase data while the chip is still inside the computer, whereas EPROM chips need to be taken out of the computer to erase it.**Cache**Cache memory is an internal high-speed semiconductor memory that stores instances of data frequently used by the CPU. It provides access to the CPU, so when the CPU requests data or programs the cache memory can transfer them to the CPU almost instantly. Cache memory usually sits between the CPU and RAM and serves as a buffer between the two.**. External**External memory, also known as secondary memory, is memory not directly connected to the CPU that you can attach or remove as needed. There are many types of external memory that individuals use in their devices.Examples include external hard drives, flash drives, memory cards and compact discs (CDs). You can store data from a computer onto external memory, remove it from the device and connect it with another compatible device to transfer data. **Optical drive**Optical drive memory is an external memory that can both store and read data using light. The most common types are CDs, DVDs and Blu-ray discs. To access the contents on an optical drive, you place the disc inside the computer and the computer spins the disc. laser beam inside the system scans it, receives the data on the optical drive and uploads it onto the computer. This type of memory can be useful because it's generally inexpensive, easily accessible and stores a lot of data.**Magnetic storage**Magnetic storage devices have a coating of magnetic material where it encodes data as an electric current. This type of memory uses magnetic fields to magnetize small sections of a metal spinning disk.Each section represents a "1" or a "0" and contains an extensive amount of data, often many terabytes. Users like this type of memory because it's affordable, durable and can store a lot of data. Common forms of magnetic storage devices are magnetic tape, hard disk drives and floppy disks.
3. 1.What is an intrupt?
4. The interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process. In I/O devices one of the bus control lines is dedicated for this purpose and is called the *Interrupt Service Routine (ISR)*. When a device raises an interrupt at let’s say process i, the processor first completes the execution of instruction i. Then it loads the Program Counter (PC) with the address of the first instruction of the ISR. Before loading the Program Counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after handling the interrupt the processor can continue with process i+1. While the processor is handling the interrupts, it must inform the device that its request has been recognized so that it stops sending the interrupt request signal. Also, saving the registers so that the interrupted process can be restored in the future, increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called Interrupt Latency
5. 2.All generation compuer?
6. **First Generation Computers**The technology behind the primary generation computers was a fragile glass device, which was called a vacuum tube. These computers were very heavy and really large. These weren’t very reliable and programming on them was a tedious task as they used low-level programming language and used no OS. First-generation computers were used for calculation, storage, and control purpose. They were too bulky and large that they needed a full room and consume a lot of electricity.Examples of some main first-generation computers are mentioned below.

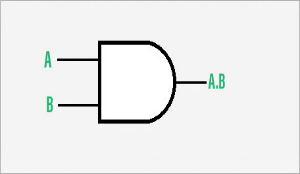
**ENIAC:** Electronic Numerical Integrator and Computer, built by J. Presper Eckert and John V. Mauchly was a general-purpose computer. It had been cumbersome, and large, and contained 18,000 vacuum tubes.**EDVAC:** Electronic Discrete Variable Automatic Computer was designed by von Neumann. It could store data also as instruction and thus the speed was enhanced.**UNIVAC:** Universal Automatic Computer was developed in 1952 by Eckert and Mauchly.**Second Generation Computers**Second-generation computers used the technology of transistors rather than bulky vacuum tubes. Another feature was the core storage. A transistor may be a device composed of semiconductor material that amplifies a sign or opens or closes a circuit.Transistors were invented in Bell Labs. The use of transistors made it possible to perform powerfully and with due speed. It reduced the dimensions and price and thankfully the warmth too, which was generated by vacuum tubes. Central Processing Unit (CPU), memory, programming language, and input, and output units also came into the force within the second generation.The programming language was shifted from high level to programming language and made programming comparatively a simple task for programmers. Languages used for programming during this era were FORTRAN (1956), ALGOL (1958), and COBOL (1959)**Third Generation Computers**

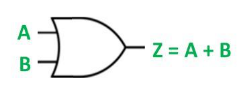
During the third generation, technology envisaged a shift from huge transistors to integrated circuits, also referred to as IC. Here a variety of transistors were placed on silicon chips, called semiconductors. The most feature of this era’s computer was speed and reliability. IC was made from silicon and also called silicon chips.A single IC has many transistors, registers, and capacitors built on one thin slice of silicon. The value size was reduced and memory space and dealing efficiency were increased during this generation. Programming was now wiped out Higher level languages like BASIC (Beginners All-purpose Symbolic Instruction Code). Minicomputers find their shape during this era. **Fourth Generation Computers**In 1971 First microprocessors were used, the large-scale of integration LSI circuits built on one chip called microprocessors. The advantage of this technology is that one microprocessor can contain all the circuits required to perform arithmetic, logic, and control functions on one chip.The computers using microchips were called microcomputers. This generation provided even smaller size of computers, with larger capacities. That’s not enough, then Very Large Scale Integrated (VLSI) circuits replaced LSI circuits. The Intel 4004 chip, developed in 1971, located all the components of the pc from the central processing unit and memory to input/ output controls on one chip and allowed the dimensions to reduce drastically.Technologies like multiprocessing, multiprogramming, time-sharing, operating speed, and virtual memory made it a more user-friendly and customary device. The concept of private computers and computer networks came into being within the fourth generation. **Fifth Generation Computers**The technology behind the fifth generation of computers is AI. It allows computers to behave like humans. It is often seen in programs like voice recognition, area of medicine, and entertainment. Within the field of game playing also it’s shown remarkable performance where computers are capable of beating human competitors.The speed is the highest, size is the smallest and area of use has remarkably increased within the fifth generation computers. Though not a hundred percent AI has been achieved to date but keeping in sight the present developments, it is often said that this dream also will become a reality very soon.To summarize the features of varied generations of computers, it is often said that a big improvement has been seen so far because of the speed and accuracy of functioning care, but if we mention the dimensions, it’s been small over the years. The value is additionally diminishing and reliability is increasing.

3.What is logic gates?

A **semiconductor** material’s electrical conductivity is somewhere between that of a conductor, such as metallic copper, and that of an insulator, such as glass. As the temperature rises, its resistivity reduces, whereas metals have the opposite effect. The conductivity of a crystal structure can be modified in a favorable way by introducing impurities (doping) to it. When two separate doped regions in the same crystal exist, a semiconductor junction is generated. The behavior of charge carriers such as electrons, ions, and electron holes at these junctions is the foundation of diodes, transistors, and most modern electronics.

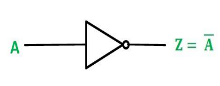
Semiconductors include silicon, germanium, gallium arsenide, and elements on the periodic table’s so-called metalloid staircase. After silicon, gallium arsenide is the second most common semiconductor, and it’s used in laser diodes, solar cells, microwave-frequency integrated circuits, and other things. Silicon is a critical component in the manufacture of nearly all electrical circuits.**Logic Gates***A logic gate is a simple switching circuit that determines whether an input pulse can pass through to the output in digital circuits.*The building blocks of a digital circuit are logic gates, which execute numerous logical operations that are required by any digital circuit. These can take two or more inputs but only produce one output. The mix of inputs applied across a logic gate determines its output. Logic gates use Boolean algebra to execute logical processes. Logic gates are found in nearly every digital gadget we use on a regular basis. Logic gates are used in the architecture of our telephones, laptops, tablets, and memory devices.**Types of Logic Gates**A logic gate is a digital gate that allows data to be transferred. Logic gates,uselogictodetermine whether or not to pass a signal. Logic gates, on the other hand, govern the flow of information based on a set of rules. The following

types of logic gates are commonly used:AND OR NOT NOR NAND XOR XNOR **Basic Logic Gates** **AND Gate**An AND gate has a single output and two or more inputs.When all of the inputs are 1, the output of this gate is 1.The AND gate’s Boolean logic is**Y=A.B** if there are two inputs A and B.An AND gate’s symbol and truth table are as follows:*Symbol of AND gate*Therefore, in And gate, the output is high when all the inputs are high.**OR Gate**Two or more inputs and one output can be used in an OR gate.The logic of this gate is that if at least one of the inputs is 1, the output will be 1.The OR gate’s output will be given by the following mathematical procedure if there are two inputs A and B: Y=A+B

, in the OR gate, the output is high when any of the inputs is high.

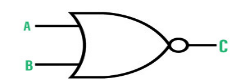
**NOT Gate**

The NOT gate is a basic one-input, one-output gate.When the input is 1, the output is 0, and vice versa. A NOT gate is sometimes called an inverter because of its feature.If there is only one input A, the output may be calculated using the Boolean equation Y=A’.

A NOT gate, as its truth table shows, reverses the input signal.**Universal Logic Gates**

**NOR Gate**

A NOR gate, sometimes known as a “NOT-OR” gate, consists of an OR gate followed by a NOT gate.This gate’s output is 1 only when all of its inputs are 0. Alternatively, when all of the inputs are low, the output is high.The Boolean statement for the NOR gate is Y=(A+B)’ if there are two inputs A and B.

By comparing the truth tables, we can observe that the outputs of the NOR gate are the polar opposite of those of an OR gate. The NOR gate is sometimes known as a universal gate since it may be used to implement the OR, AND, and NOT gates.**NAND Gate**A NAND gate, sometimes known as a ‘NOT-AND’ gate, is essentially a Not gate followed by ANDgate.This gate’s output is 0 only if none of the inputs is 0. Alternatively, when all of the inputs are not high and at least one is low, the output is high.If there are two inputs A and B, the Boolean expression for the NAND gate is Y=(A.B)’By comparing their truth tables, we can observe that their outputs are the polar opposite of an AND gate. The NAND gate is known as a universal gate because it may be used to implement the AND, OR, and NOT gates.**Other Logic GatesXOR Gate**

The Exclusive-OR or ‘Ex-OR’ gate is a digital logic gate that accepts more than two inputs but only outputs one value.If any of the inputs is ‘High,’ the output of the XOR Gate is ‘High.’ If both inputs are ‘High,’ the output is ‘Low.’ If both inputs are ‘Low,’ the output is ‘Low.’The Boolean equation for the XOR gate is Y=A’.B+A.B’ if there are two inputs A and B.Its outputs are based on OR gate logic, as we can see from the truth table.**XNOR Gate**

The Exclusive-NOR or ‘EX-NOR’ gate is a digital logic gate that accepts more than two inputs but only outputs one.If both inputs are ‘High,’ the output of the XNOR Gate is ‘High.’ If both inputs are ‘Low,’ the output is ‘High.’ If one of the inputs is ‘Low,’ the output is ‘Low.’If there are two inputs A and B, then the XNOR gate’Boolean equation is: Y=A.B+A’B’.The truth table shows that its outputs are based on NOR gate logic.

**4.What is flip-fop?**

## In [electronics](https://en.wikipedia.org/wiki/Electronics), flip-flops and latches are [circuits](https://en.wikipedia.org/wiki/Electronic_circuit) that have two stable states that can store state information – a [bistable multivibrator](https://en.wikipedia.org/wiki/Bistable_multivibrator). The circuit can be made to change state by [signals](https://en.wikipedia.org/wiki/Signal) applied to one or more control inputs and will output its state (often along with its [logical complement](https://en.wikipedia.org/wiki/Logical_complement) too). It is the basic storage element in [sequential logic](https://en.wikipedia.org/wiki/Sequential_logic). Flip-flops and latches are fundamental building blocks of [digital electronics](https://en.wikipedia.org/wiki/Digital_electronics) systems used in computers, communications, and many other types of systems.Flip-flops and latches are used as data storage elements to store a single [*bit*](https://en.wikipedia.org/wiki/Bit) (binary digit) of data; one of its two states represents a "one" and the other represents a "zero". Such data storage can be used for storage of [*state*](https://en.wikipedia.org/wiki/State_(computer_science)), and such a circuit is described as [sequential logic](https://en.wikipedia.org/wiki/Sequential_logic) in electronics. When used in a [finite-state machine](https://en.wikipedia.org/wiki/Finite-state_machine), the output and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be used for counting of pulses, and for synchronizing variably-timed input signals to some reference timing signal.The term flip-flop has historically referred generically to both level-triggered (asynchronous, transparent, or opaque) and edge-triggered ([synchronous](https://en.wikipedia.org/wiki/Synchronous), or [clocked](https://en.wikipedia.org/wiki/Clock_signal)) circuits that store a single bit of data using [gates](https://en.wikipedia.org/wiki/Logic_gate).[[1]](https://en.wikipedia.org/wiki/Flip-flop_(electronics)#cite_note-1) Modern authors reserve the term *flip-flop* exclusively for edge-triggered storage elements and *latches* for level-triggered ones.[[2]](https://en.wikipedia.org/wiki/Flip-flop_(electronics)#cite_note-pedroni-2)[[3]](https://en.wikipedia.org/wiki/Flip-flop_(electronics)#cite_note-ee42-3) The terms "edge-triggered", and "level-triggered" may be used to avoid ambiguity.[[4]](https://en.wikipedia.org/wiki/Flip-flop_(electronics)#cite_note-Roth,_Charles_H_1995-4)When a level-triggered latch is enabled it becomes transparent, but an edge-triggered flip-flop's output only changes on a clock edge (either positive going or negative going). Types[Flip-flops and latches can be divided into common types: the SR ("set-reset"), D ("data" or "delay"[[14]](https://en.wikipedia.org/wiki/Flip-flop_(electronics)#cite_note-14)), T ("toggle"), and JK. The behavior of a particular type can be described by what is termed the characteristic equation, which derives the "next" (i.e., after the next clock pulse) output, Qnext in termof the input signal(s) and/or the current output, �Simple set-reset latches[[edit](https://en.wikipedia.org/w/index.php?title=Flip-flop_(electronics)&action=edit&section=4)]When using static gates as building blocks, the most fundamental latch is the simple *SR latch*, where S and R stand for *set* and *reset*. It can be constructed from a pair of cross-coupled [NOR](https://en.wikipedia.org/wiki/NOR_gate) or [NAND](https://en.wikipedia.org/wiki/NAND_gate) [logic gates](https://en.wikipedia.org/wiki/Logic_gate). The stored bit is present on the output marked Q.SR NOR latch[[edit](https://en.wikipedia.org/w/index.php?title=Flip-flop_(electronics)&action=edit&section=5)]An animation of a SR latch, constructed from a pair of cross-coupled [NOR gates](https://en.wikipedia.org/wiki/NOR_gate). Red and black mean logical '1' and '0', respectively.An animated SR latch. Black and white mean logical '1' and '0', respectively.S = 1, R = 0: SetS = 0, R = 0: HoldS = 0, R = 1: ResetS = 1, R = 1: Not allowed

Transitioning from the restricted combination (D) to (A) leads to an unstable state.While the R and S inputs are both low, [feedback](https://en.wikipedia.org/wiki/Feedback) maintains the Q and Q outputs in a constant state, with Q the complement of Q. If S (*Set*) is pulsed high while R (*Reset*) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low.

*Note: X means*don't care*, that is, either 0 or 1 is a valid value.*

The R = S = 1 combination is called a **restricted combination** or a **forbidden state** because, as both NOR gates then output zeros, it breaks the logical equation Q = **not** Q. The combination is also inappropriate in circuits where *both* inputs may go low *simultaneously* (i.e. a transition from *restricted* to *keep*). The output would lock at either 1 or 0 depending on the propagation time relations between the gates (a [race condition](https://en.wikipedia.org/wiki/Race_condition)).How an SR NOR latch worksTo overcome the restricted combination, one can add gates to the inputs that would convert (S, R) = (1, 1) to one of the non-restricted combinations. That can be:Q = 1 (1, 0) – referred to as an *S (dominated)-latch*Q = 0 (0, 1) – referred to as an *R (dominated)-latch*This is done in nearly every [programmable logic controller](https://en.wikipedia.org/wiki/Programmable_logic_controller).Keep state (0, 0) – referred to as an *E-latch*Alternatively, the restricted combination can be made to *toggle* the output. The result is the [JK latch](https://en.wikipedia.org/wiki/Flip-flop_(electronics)#JK_latch).The characteristic equation for the SR latch is:

5.what is boolen algebra?**Boolean algebra** is a type of [algebra](https://www.geeksforgeeks.org/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**. **Boolen algebra** is concerned with binary variables and logic operationsBoolean Algebra is fundamental in the development of digital electronics systems as tall use the concept of **Boolean Algebra** to execute commands. Apart from digital electronics this algebra also finds its application 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.Boolean Algebra OperationsThere are various operations that 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

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 OperationUsing 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)’ = 0If 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 = 0Conjunction 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 = TrueIf A = True, B = False, Or A = false, B = True, then A . B = FalseIf A = False, B = False, then A . B = FalseDisjunction (OR) Operation

Using the OR operation satisfies the condition if any value of the individual variables are 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 = TrueIf A = True, B = False, Or A = false, B = True, then A + B = TrueIf A = False, B = False, then A + B = FalseBoolean 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 **Boolen 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 = TrueA.B = True(A)’ = FalseBoolean Algebra TerminologiesThere are various terminologies related to Boolean Algebra, which are used to explain various parameters of **Boolen Algebra**. That includes,Boolean AlgebraBoolean VariablesBoolean FunctionLiteralComplementTruth TableNow, we will discuss the important terminologies of Boolean algebra in the article belowBoolean AlgebraThe branch of algebra that deals with binary operations or logical operations is called Boolean Algebra.Boolean VariablesVariables 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 FunctionA function of the Boolean Algebra that is formed by the use of Boolean variables and Boolean operators is called the Boolean function.LiteralA variable or the complement of the variable in Boolean Algebra is called the Literal.

### ComplementThe 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. Itrepresented by ‘ over the variable.Truth TableTable 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,*where “****n”****is the number of boolean variables used.*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.

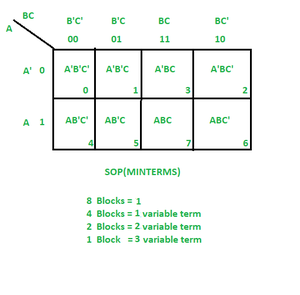
**Example: Draw the truth table of the conditions A + B and A.B where A and b are boolean variables.**

**Solution:**Laws for Boolean AlgebraThe basic laws of the Boolean Algebra are added in the table added belowLet’s learn about these laws in detail.Identity LawIn 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 = AA.1 = ACommutative LawBinary 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. AA + B = B + AAssociative LawAssociative 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 LawBoolean Variables also follow the distributive law and the expression for Distributive law is given as:

A . ( B + C) = (A . B) + (A . C)Inversion LawInversion 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’)’ = AApart from these other laws are mentioned below:AND LawAND law of the Boolean algebra uses AND operator and the AND law is,A . 0 = 0A . 1 = AA . A = AOR LawOR law of the Boolean algebra uses OR operator and the OR law is,A + 0 = AA + 1 = 1A + A = ADe Morgan’s Laws are also called Demorgan’s Theorem. They are the most important laws in **Boolen Algebra** and these are added below under the heading Boolean Algebra TheoremBoolean Algebra TheoremsThere 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 lawsDe Morgan’s Law states that,**Statement:**The 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: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 twoBoolean 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:

5.define? A hard disk drive (HDD), hard disk, hard drive, or fixed disk, is**an electro-mechanical data storage device that stores and retrieves digital data using magnetic storage with one or more rigid rapidly rotating platters coated with magnetic material.** A **floppy disk** or **floppy diskette** (casually referred to as a **floppy** or a **diskette**) is a type of [disk storage](https://en.wikipedia.org/wiki/Disk_storage) composed of a thin and flexible disk of a [magnetic storage](https://en.wikipedia.org/wiki/Magnetic_storage) medium in a square or nearly square plastic enclosure lined with a fabric that removes dust particles from the spinning disk. Floppy disks store [digital data](https://en.wikipedia.org/wiki/Digital_data) which can be read and written when the disk is inserted into a **floppy disk drive** (**FDD**) connected to or inside a [computer](https://en.wikipedia.org/wiki/Computer) or other device. CD-ROM is a read-only [optical media](https://techterms.com/definition/opticalmedia) format that uses [CDs](https://techterms.com/definition/cd) to store data. A computer with an [optical drive](https://techterms.com/definition/opticaldrive) can read, but not write, data on a CD-ROM. A single disc can store up to 700 MB of data. Some CD-ROM discs, known as "Enhanced CDs," store a combination of audio tracks and a data track. CD-ROM discs were often used to distribute [software](https://techterms.com/definition/software) and video games in the 1990s and 2000s before eventually being replaced by higher-capacity [DVDs](https://techterms.com/definition/dvd). A register is a temporary storage area built into a [CPU](https://techterms.com/definition/cpu). Some registers are used internally and cannot be accessed outside the [processor](https://techterms.com/definition/processor), while others are user-accessible. Most modern CPU [architectures](https://techterms.com/definition/architecture) include both types of registers.Internal registers include the instruction register (IR), memory buffer register (MBR), memory data register (MDR), and memory address register (MAR). The instruction register fetches instructions from the program counter (PC) and holds each instruction as it is executed by the processor. The memory registers are used to pass data from [memory](https://techterms.com/definition/memory) to the processor. The storage time of internal registers is extremely temporary, as they often hold data for less than a millisecond. **Associative memory** is also known as content addressable memory (CAM) or associative storage or associative array. It is a special type of memory that is optimized for performing searches through data, as opposed to providing a simple direct access to the data based on the address.

| **A** | **B** | **A + B** | **(A + B)’** | **A’** | **B’** | **A’. B’** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

it can store the set of patterns as memories when the associative memory is being presented with a key pattern, it responds by producing one of the stored pattern which closely resembles or relates to the key pattern.it can be viewed as **data correlation** here. input data is correlated with that of stored data in the CAM. Shift register is used as Parallel to serial converter, which converts the parallel data into serial data. It is utilized at the transmitter section after Analog to Digital Converter ADC��� block.Shift register is used as Serial to parallel converter, which converts the serial data into parallel data. It is utilized at the receiver section before Digital to Analog Converter DAC��� block.Shift register along with some additional gates� generate the sequence of zeros and ones. Hence, it is used as sequence generator.Shift registers are also used as counters. There are two types of counters based on the type of output from right most D flip-flop is connected to the serial input. Those are Ring counter and Johnson Ring counter. Sequential circuits are digital circuits that store and use the previous state information to determine their next state. Unlike combinational circuits, which only depend on the current input values to produce outputs, sequential circuits depend on both the current inputs and the previous state stored in memory elements.Sequential circuits are commonly used in digital systems to implement state machines, timers, counters, and memory elements. The memory elements in sequential circuits can be implemented using flip-flops, which are circuits that store binary values and maintain their state even when the inputs change.There are two types of sequential circuits: finite state machines (FSMs) and synchronous sequential circuits. FSMs are designed to have a limited number of states and are typically used to implement state machines and control systems. Synchronous sequential circuits, on the other hand, are designed to have an infinite numare typically used to implement timers, counters, and memory eleSynchronous sequential circuit: These circuits **uses clock signal** and level inputs (or pulsed) (with restrictions on pulse width and circuit propagation). The output pulse is the same duration as the clock pulse for the clocked sequential circuits. Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are bit **slower** compared to asynchronous. Level output changes state at the start of an input pulse and remains in that until the next input or clock pulse. **Combinatioal circuit**These circuits are developed using AND, OR, NOT, NAND, and NOR logic gates. These logic gates are building blocks of combinational circuits. A combinational circuit consists of input variables and output variables. Since these circuits are not dependent upon previous input to generate any output, so are combinational logic circuits. A combinational circuit can have an n number of inputs and m number of outputs. In combinational circuits, the output at any time is a direct function of the applied external inputs. A **universal gate** is a logic gate which can implement any Boolean function without the need to use any other type of logic gate. The [NOR gate](https://www.electrical4u.com/nor-gate/) and [NAND gate](https://www.electrical4u.com/nand-gate/) are universal gates. This means that you can create any logical Boolean expression using only NOR gates or only NAND gates.In practice, this is advantageous since NOR and NAND gates are economical and easier to fabricate than other logic gates. So much so that an [AND gate](https://www.electrical4u.com/logical-and-gate/) is typically implemented as a NAND gate followed by an inverter (not the other way around)! Similarly, an [OR gate](https://www.electrical4u.com/logical-or-gate/) is typically realised as a NOR gate followed by an inverter.Race Around Condition in JK Flip-flopFor J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then output Q will toggle as long as CLK remains high which makes the output unstable or uncertain.This is called a race around condition in J-K flip-flop.We can overcome this problem by making the clock =1 for very less duration.The circuit used to overcome race around conditions is called the   The truth table for first De Morgan’s Law is given as follows:

## ****Proof Using Algebra of Sets****

Second De Morgan’s law: (A ∩ B)’ = A’ ∪ B’

Let X = (A ∩ B)’ and Y = A’ ∪ B’

| **A** | **B** | **A . B** | **(A. B)’** | **A’** | **B’** | **A’ + B’** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |

Let p be any element of X, then p ∈ X ⇒ p ∈ (A ∩ B)’

⇒ p ∉ (A ∩ B)

⇒ p ∉ A and p ∉ B

⇒ p ∈ A’ or p ∈ B’

⇒ p ∈ A’ ∪ B’

⇒ p ∈ Y

∴ X ⊂ Y ————–(i) Again, let q be any element of Y, then q ∈ Y ⇒ q ∈ A’ ∪ B’

⇒ q ∈ A’ or q ∈ B’

⇒ q ∉ A and q ∉ B

⇒ q ∉ (A ∩ B)

⇒ q ∈ (A ∩ B)’

⇒ q ∈ X

∴ Y ⊂ X ————–(ii)

From (i) and (ii) X = Y

(A ∩ B)’ = A’ ∪ B’

We need to prove, (A ∪ B)’ = A’ ∩ B’

Let X = (A ∪ B)’ and Y = A’ ∩ B’

Let p be any element of X, then p ∈ X ⇒ p ∈ (A ∪ B)’

⇒ p ∉ (A ∪ B)

⇒ p ∉ A or p ∉ B

⇒ p ∈ A’ and p ∈ B’

⇒ p ∈ A’ ∩ B’

⇒ p ∈ Y

∴ X ⊂ Y              . . . (i)

Again, let q be any element of Y, then q ∈ Y ⇒ q ∈ A’ ∩ B’

⇒ q ∈ A’ and q ∈ B’⇒ q ∉ A or q ∉ B⇒ q ∉ (A ∪ B)⇒ q ∈ (A ∪ B)’⇒ q ∈ X∴ Y ⊂ X              . . . (ii)From (i) and (ii) X = Y(A ∪ B)’ = A’ ∩ B’