BOOLEAN ALGEBRA

AND LAWS

OR + LAWS

DE MORGAN’S THEOREM

NOTE THIS

PROPERTIES OF BOOLEAN ALGEBRA

Commutative Law:

Associative Law

Distributive Law

QUESTIONS

1.

Solution

2.

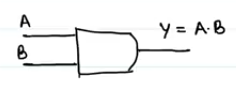
That’s the end of that question

3.

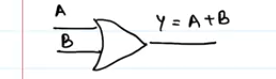
LOGIC GATES

Basic Gates → And, Or, Not

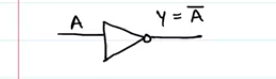
AND.



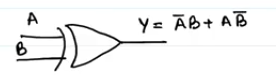
OR



NOT



EXOR / XOR → Exclusive Or

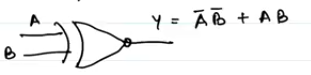


You’ll notice that when you make the truth table of an exor gate, you’ll notice that

If inputs are same, output = 0

If inputs are different, output = 1

EXNOR, XNOR. This is the opposite of the exor gate



On solving with De Morgan’s theorem, you’ll get

For the truth that if both inputs are the same, the output will be 1

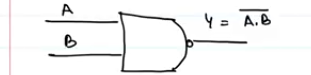
If the inputs are not the same, output is 0.

UNIVERSAL GATES

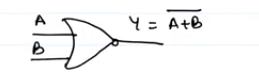
These are the NAND and NOR gates.

These are called universal gates because they can be used to construct any other gate as will be seen below

NAND GATE

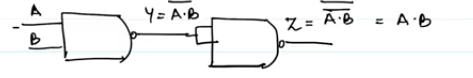


NOR GATE

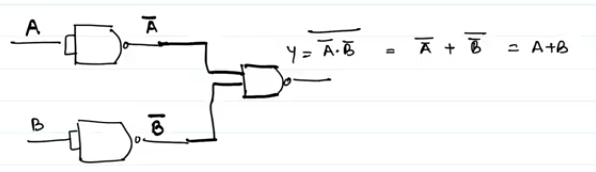


CONSTRUCTION OF GATES USING NAND GATE

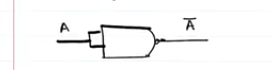
AND GATE



OR GATE BY NAND GATE

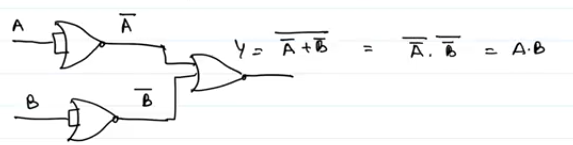


NOT GATE BY NAND GATE

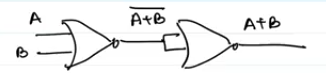


CONSTRUCTING THE GATES USING THE NOR GATE

AND GATE BY NOR GATE



OR GATE WITH NOR GATE



NOT GATE WITH NOR GATE



So those are the gates and solvings

SOME EXTRA NOTES

LOGIC CIRCUITS

Signals can be represented as ON or OFF, 1 or 0 as well. Therefore,

1 means ON or TRUE (T)

0 means OFF or FALSE (F)

A large number of electronic circuits (in computers, control units, and so on) are made up of logic gates. These process signals which represent true or false

Logic circuit is a circuit to perform complex functions defined in terms of elementary functions of mathematical logic. An electronic circuit used in computers to perform a logical operation on its two or more input signals. There are six basic circuits which are the AND, NOT, NAND, OR, NOR and exclusive OR circuits which can be combined into more complex circuits.

Logic gates:

Digital systems are said to be constructed by using logic gates. These gates are the BUFFER, AND, OR, NOT, NAND, NOR, EXOR, EXNOR gates. The basic operations are described below with the aid of a truth table.

Truth tables are used to help show the functions of logic gates

AND Gate





The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. Bear in mind that this dot is sometimes omitted i.e. AB

OR gate





The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.

NOT Gate





The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an *inverter*. If the input variable is “A”, the inverted output is known as NOT A.

This is also shown as:

A', or

A with a bar over the top .

The diagrams below show two ways that the NAND logic gate can be configured to produce a NOT gate. It can also be done using NOR logic gates in the same way.



To use a nand gate as a nor gate, divide the input A into two then you get two inputs A and A. Then on the NAND operation, you get (A.A)’ which will equal A’ which is not A.

NAND GATE





This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion

NOR gate:





This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high.

The symbol is an OR gate with a small circle on the output. The small circle represents inversion

EXOR or XOR





The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both, of its two inputs are high. An encircled plus sign () is used to show the EOR operation

EXNOR Gate





The 'Exclusive-NOR' gate circuit does the opposite to the EOR gate. It will give a low output if either, but not both, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion

Note:

The NAND and NOR gates are called *universal functions* since with either one, the “AND” “OR” and “NOT” functions can be generated.

A function in *sum of products* form can be implemented using NAND gates by replacing all AND and OR gates by NAND gates.

A function in *product of sums* form can be implemented using NOR gates by replacing all AND and OR gates by NOR gates.

In general, the gate symbols are:



The above table is a summary truth table of the input/output combinations for the NOT gate together with all possible input/output combinations for the other gate functions. Also note that a truth tables with 'n' inputs has rows. You can compare the outputs of different gates.

Take note of the following diagrams





In logic gates, we have to learn to interpret circuits and draw given truth tables e.g.

This logic circuit gives the following truth table and output expression.

TRUTH TABLES

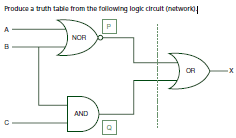
Truth tables are used to show logic gate functions

The not gate has only one input, but all others have two inputs

When constructing a truth table, the binary values 1 and 0 are used. Every possible combination, depending on the number of inputs is produced. Basically, the number of possible combinations of ones an zeros is where n = number of inputs. For example, 2 inputs will have 4 combinations, 3 inputs will have 8 combinations

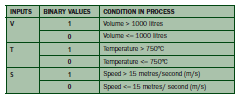


Try the following examples



A system used 3 switches A, B and C; a combination of switches determines whether an alarm, X, sounds: If switch A or switch B are in the ON position and if switch C is in the OFF position then a signal to sound an alarm, X is produced.

A manufacturing process is controlled by a built in logic circuit which is made up of AND, OR and NOT gates only. The process receives a STOP signal (i.e. X = 1) depending on certain conditions, shown in the following table:

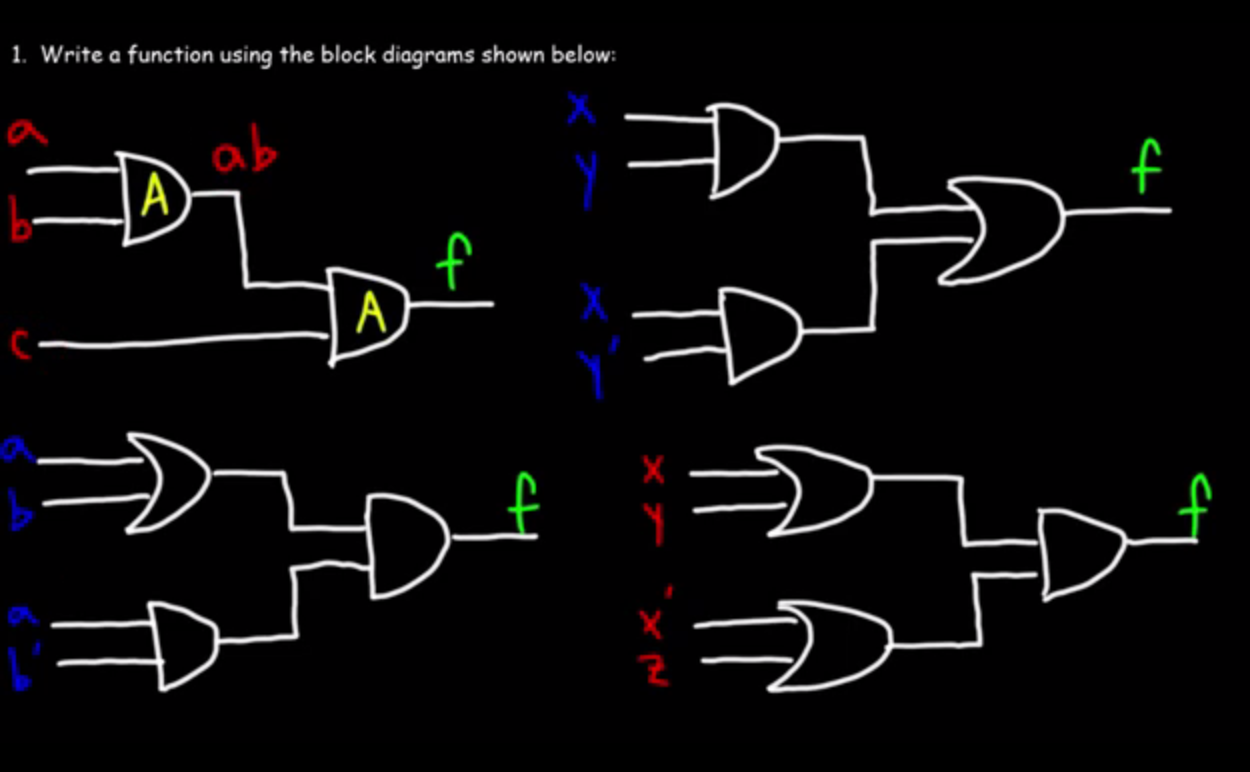
‘

A stop signal (X = 1) occurs when:

Volume, V > 1000 litres and Speed, S <= 15 m/s or

Temperature, T <= 750ºC and Speed, S > 15 m/s

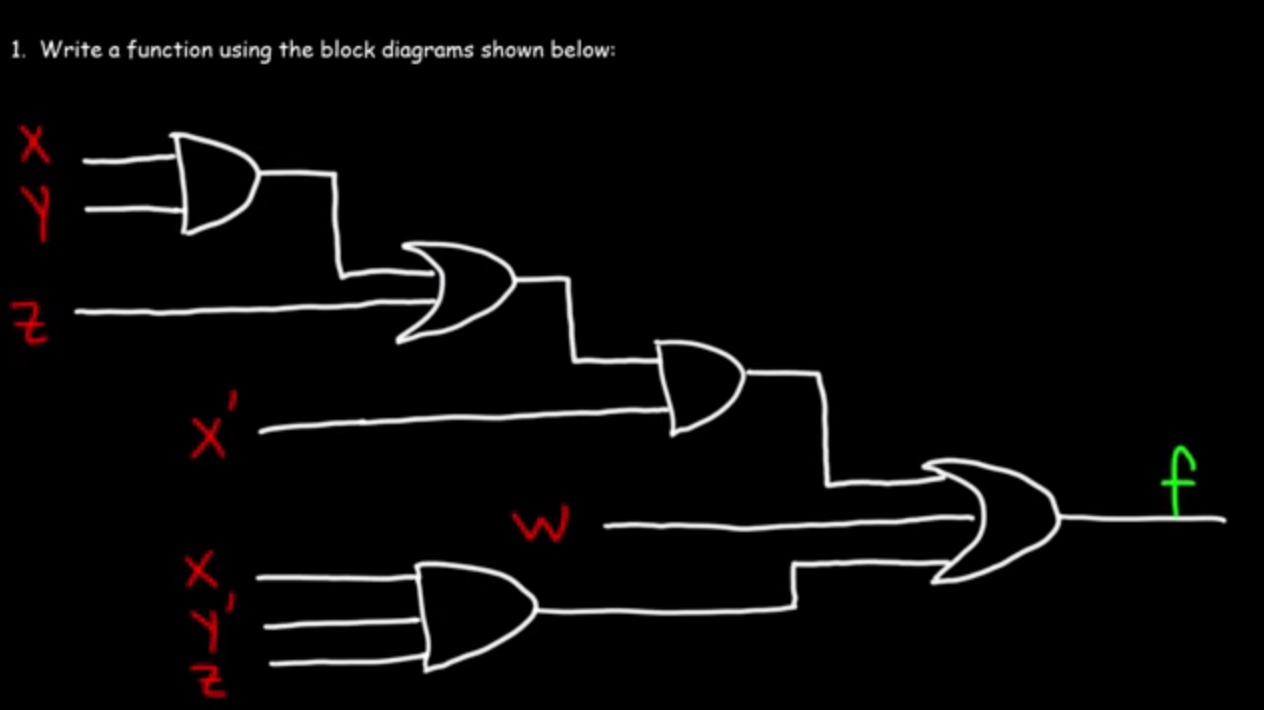
Draw the logic circuit and truth table to show all the possible situations when the stop signal could be received.

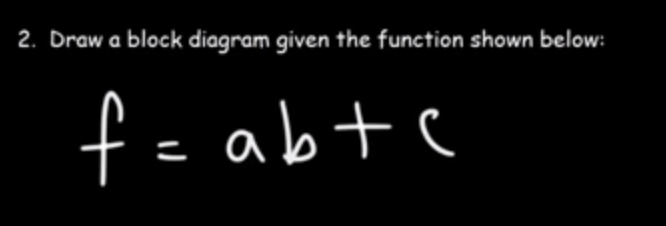


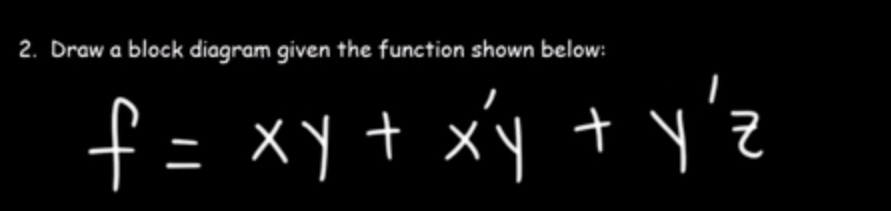
For the first one, the answer is f = abc

The second one is f = (xy)+(xy’)

For the second one, the answer is f = (a + b)(ab’)

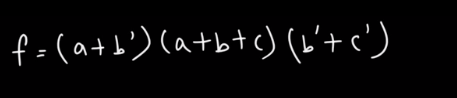






The above is said to be a sum of products (SOP)

Literals -> inputs



This is a product of sums (POS)

SWITCHING CIRCUITS

A Switch is a device used to control the current in an electric circuit.

IF the switch is open (i.e. it is off) current doesn’t flow through the circuit and vice-versa

SWITCHING CIRCUIT DESIGN

This means two points of electric circuit connected by wires and consisting of a finite number of switches

WAYS OF CONNECTING SWITCHES

Parallel arrangement:

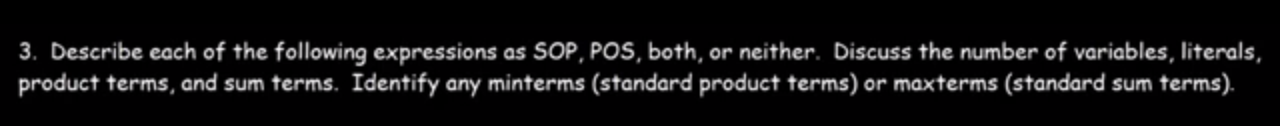
In parallel connection, current will be passed if either one of the switches is closed. That is to say that there is an “OR” operation in parallel arrangement

Series Arrangement:

For switches in series, all the switches have to be closed for current to pass therefore making the operation in series arrangement an “AND” operation

EQUIVALENT CIRCUITS

These are two switching circuits if in all positions the closure conditions are the same. If two circuits A and B are equivalent, we can write it as





This is a sum of products

Four product terms

We have four variables, x, y, z, w

We have 12 literals

A minterm (standard product terms) is a term that contains all variables x’yzw is the minterm



Product of sums

3 variables

Max term (standard sum term) is (x + y + z’)



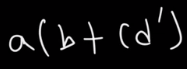
This is both an SOP (Sum of one product term) and a POS (Product of four sum terms)

We have four variables (w, x, y, z)

We also have four literals (w, x, y’, z)



This can be called an SOP or a POS



This is neither “a sum of products” nor a product of sums

BASIC LOGIC IN BOOLEAN ALGEBRA

Commutative property

A+B = B+A

AB = AB

Associative property

A+(B+C)=(A+B)+C

A(BC)=(AB)C

Identity rule

A+0=A

A X 1= A

Null property

A+1 = 1

A X 0

A + A’ = 1

A X A’ = 0