**Solution SET-A**

**Subject: -CA**

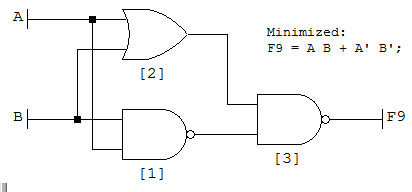
**I Year MCA. Ist Semester Ist Mid Term Examination, Oct. – 2018**

Ans 1 a) A logic gate is an elementary building block of a [digital](https://whatis.techtarget.com/definition/digital) [circuit](https://whatis.techtarget.com/definition/circuit). Most logic gates have two inputs and one output.

b) The **encoder** is a device or a transducer or a circuit.The **encoder** will convert the information from one format to another format i.e like electrical signals to counters .The **decoder** is a circuit used to change the code into a set of signals.

c) **DeMorgan's Theorems** are two additional simplification techniques that can be used to simplify Boolean expressions. Again, the simpler the Boolean expression the simpler the resulting the Boolean expression, the simpler the resulting logic.

d)



e) Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer.

Ans 2 a) The *XNOR (exclusive-NOR) gate* is a combination XOR gate followed by an inverter. Its output is "true" if the inputs are the same, and"false" if the inputs are different.

/WhatIs/images/xnor.gif (278 bytes)

**XNOR gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  |  | 1 |
|  | 1 |  |
| 1 |  |  |
| 1 | 1 | 1 |

Using combinations of logic gates, complex operations can be performed.

b) (54.6875)10=( )2

First we convert the integer 54 to a binary number in the normal way using successive division from above.

54 (divide by 2)  =  27  remainder **0**  (LSB)

27 (divide by 2)  =  13  remainder **1**  (↑)

13 (divide by 2)  =  6  remainder **1**  (↑)

6 (divide by 2)  =  3  remainder **0**  (↑)

3 (divide by 2)  =  1  remainder **1**  (↑)

1 (divide by 2)  =  0  remainder **1**  (MSB)

Thus the binary equivalent of 5410 is therefore: 1101102

Next we convert the decimal fraction 0.6875 to a binary fraction using successive multiplication.

0.6875 (multiply by 2)  =  **1**.375  =  0.375 carry **1**  (MSB)

0.375 (multiply by 2)  =  **0**.75  =  0.75 carry **0**  (↓)

0.75 (multiply by 2)  =  **1**.50  =  0.5 carry **1**  (↓)

0.5 (multiply by 2)  =  **1**.00    =  0.0 carry **1**  (LSB)

Thus the binary equivalent of 0.687510 is therefore: 0.10112  ← (LSB)

Hence the binary equivalent of the decimal number: 54.687510 is 110110.10112

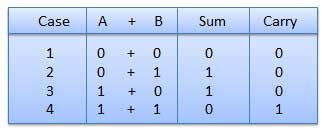
Ans 3 a) **8-1 Multiplexer Circuit**

For the combination of selection input, the data line is connected to the output line. The circuit shown below is an 8\*1 multiplexer. The 8-to-1 multiplexer requires 8 AND gates, one OR gate and 3 selection lines. As an input, the combination of selection inputs are giving to the AND gate with the corresponding input data lines.

In a similar fashion, all the AND gates are given connection. In this 8\*1 multiplexer, for any selection line input, one AND gate gives a value of 1 and the remaining all AND gates give 0. And, finally, by using OR gate, all the AND gates are added; and, this will be equal to the selected value.

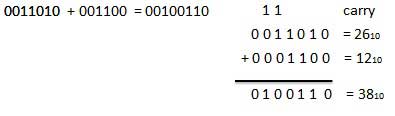
## b) Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.



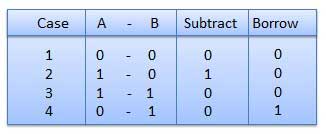
In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

### Example − Addition



## Binary Subtraction

**Subtraction and Borrow**, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

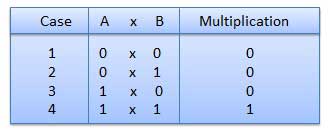


### Example − Subtraction

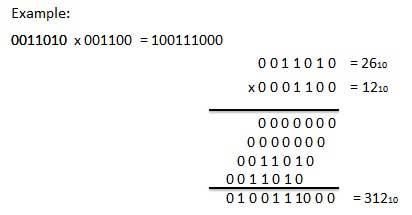


## Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.



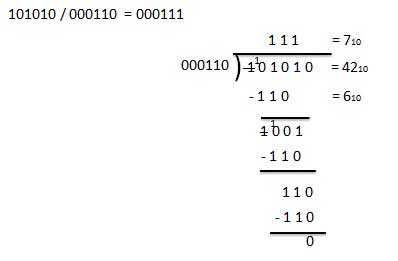
### Example − Multiplication



## Binary Division

Binary division is similar to decimal division. It is called as the long division procedure.

### Example − Division

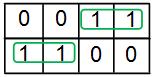


Ans 4 **Minterm Solution of K Map**

The following are the steps to obtain simplified minterm solution using [K-map](https://www.electrical4u.com/k-map/). **Step 1: Initiate** Express the given expression in its canonical form

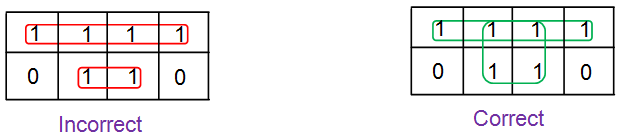
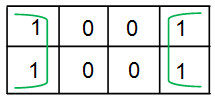
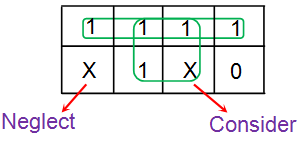
**Step 2: Populate the K-map** Enter the value of 'one' for each product-term into the K-map cell, while filling others with zeros.

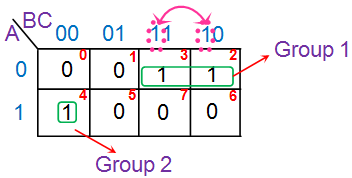
**Step 3: Form Groups**

1. Consider the consecutive 'ones' in the K-map cells and group them (green boxes). 

2 Each group should contain the largest number of 'ones' and no blank cell. 

3 The number of 'ones' in a group must be a power of 2 i.e. a group can contain https://www.electrical4u.com/images/march16/1460725662.GIF

1. Grouping has to be carried-on in decreasing order meaning, one has to try to group for 8 (octet) first, then for 4 (quad), followed by 2 and lastly for 1 (isolated 'ones'). 
2. Grouping is to done either horizontally or vertically or interms of squares or rectangles. Diagonal grouping of 'ones' is not permitted. 
3. The same element(s) may repeat in multiple groups only if this increases the size of the group. 
4. The elements around the edges of the table are considered to be adjacent and can be grouped together. 
5. Don’t care conditions are to be considered only if they aid in increasing the group-size (else neglected). 

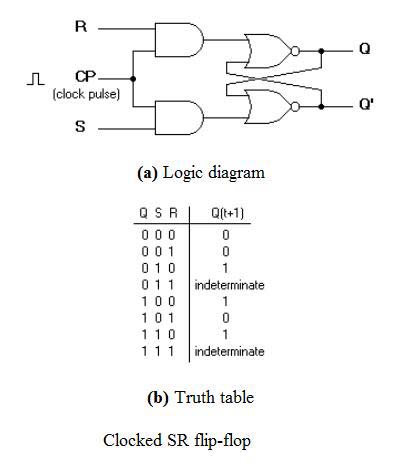
**Step 4: Obtain Boolean Expression for Each Group** Express each group interms of input variables by looking at the common variables seen in cell-labelling. For example in the figure shown below there are two groups with two and one number of 'ones' in them (Group 1 and Group 2, respectively). All the 'ones' in the Group 1 of the [K-map](https://www.electrical4u.com/k-map/) are present in the row for which A = 0. Thus they contain the variable A̅. Further these two 'ones' are present in adjacent columns which have only B term in common as indicated by the pink arrow in the figure.Hence the next term is B. This yields the product term corresponding to this group as A̅B. Similarly the 'one' in the Group 2 of the K-map is present in the row for which A = 1. Further the variables corresponding to its column are B̅C̅. Thus one gets the overall product-term for this group as AB̅C̅. **Step 5: Obtain Boolean Expression for the Output** The product-terms obtained for individual groups are to be combined to form sum-of-product (SOP) form which yields the overall **simplified Boolean expression**.

OR

### Ans **S-R Flip Flop**

It is also called a Gated S-R flip flop.

The problems with S-R flip flops using NOR and NAND gate is the invalid state. This problem can be overcome by using a bistable SR flip-flop that can change outputs when certain invalid states are met, regardless of the condition of either the Set or the Reset inputs. For this, a clocked S-R flip flop is designed by adding two AND gates to a basic NOR Gate flip flop. The circuit diagram and truth table is shown below.



Clocked S-R Flip Flop

A clock pulse [CP] is given to the inputs of the AND Gate. When the value of the clock pulse is ‘0’, the outputs of both the AND Gates remain ‘0’. As soon as a pulse is given the value of CP turns ‘1’. This makes the values at S and R to pass through the NOR Gate flip flop. But when the values of both S and R values turn ‘1’, the HIGH value of CP causes both of them to turn to ‘0’ for a short moment. As soon as the pulse is removed, the flip flop state becomes intermediate. Thus either of the two states may be caused, and it depends on whether the set or reset input of the flip-flop remains a ‘1’ longer than the transition to ‘0’ at the end of the pulse. Thus the invalid states can be eliminated.

SET-B

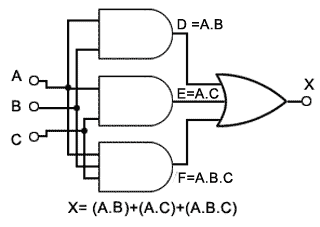
Ans 1 a) **A Boolean expression** always produces a Boolean value. A Boolean expression is composed of a combination of the Boolean constants (True or False), Boolean variables and logical connectives. Each Boolean expression represents a Boolean function.

b) A truth table is a tabular representation of all the combinations of values for inputs and their corresponding outputs. It is a mathematical table that shows all possible outcomes that would occur from all possible scenarios that are considered factual, hence the name.

c) Commutative Laws:-The "Commutative Laws" say we can **swap numbers** over and still get the same answer .**add**: a + b**=**b + a & **multiply**:a × b**=**b × a

Associative Laws :-The "Associative Laws" say that it doesn't matter how we group the numbers (a + b) + c**=**a + (b + c) &**multiply**: (a × b) × c**=**a × (b × c)

d)



e)Half adder- An adder is a digital circuit that performs addition of numbers. The half adder adds two binary digits called as augend and addend and produces two outputs as sum and carry; XOR is applied to both inputs to produce sum and AND gate is applied to both inputs to produce carry.

Full adder-This adder is difficult to implement than a half-adder. The difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs, whereas half adder has only two inputs and two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. When a full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

Ans 2 a) The *NAND gate* operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "and" followed by negation. The output is "false" if both inputs are "true." Otherwise, the output is "true."

/WhatIs/images/nand.gif (240 bytes)

**NAND gate**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
|  |  | 1 |
|  | 1 | 1 |
| 1 |  | 1 |
| 1 | 1 |  |

b) Hexadecimal Number: 19FDE16 =( )10

|  |  |  |
| --- | --- | --- |
| **Step** | **Binary Number** | **Decimal Number** |
| Step 1 | 19FDE16 | ((1 x 164) + (9 x 163) + (F x 162) + (D x 161) + (E x 160))10 |
| Step 2 | 19FDE16 | ((1 x 164) + (9 x 163) + (15 x 162) + (13 x 161) + (14 x 160))10 |
| Step 3 | 19FDE16 | (65536+ 36864 + 3840 + 208 + 14)10 |
| Step 4 | 19FDE16 | 10646210 |

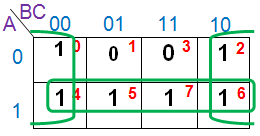
Octal Number: 125708 =( )10

|  |  |  |
| --- | --- | --- |
| **Step** | **Octal Number** | **Decimal Number** |
| Step 1 | 125708 | ((1 x 84) + (2 x 83) + (5 x 82) + (7 x 81) + (0 x 80))10 |
| Step 2 | 125708 | (4096 + 1024 + 320 + 56 + 0)10 |
| Step 3 | 125708 | 549610 |

Ans 3 a) **1-8 De-multiplexers**

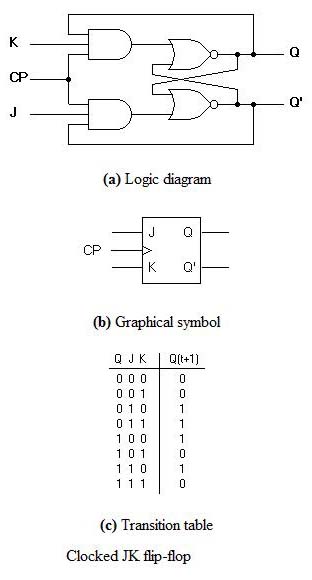
The demultiplexer is also called as data distributors as it requires one input, 3 selected lines and 8 outputs. De-multiplexer takes one single input data line, and then switches it to any one of the output line. 1-to-8 demultiplexer circuit diagram is shown below; it uses 8 AND gates for achieving the operation. The input bit is considered as data D and it is transmitted to the output lines. This depends on the control input value of the AB. When AB = 01, the upper second gate F1 is enabled, while the remaining AND gates are disabled, and the data bit is transmitted to the output giving F1= data. If D is low, the F1 is low, and if D is high, the F1 is high. So the value of the F1 depends on the value of D, and the remaining outputs are in low state.

b)

https://www.electrical4u.com/images/march16/1460728146.GIF

## Ans 4 **J-K Flip Flop**

The circuit diagram and truth-table of a J-K flip flop is shown below.



J-K Flip Flop

A J-K flip flop can also be defined as a modification of the S-R flip flop. The only difference is that the intermediate state is more refined and precise  than that of  a S-R flip flop.

The behavior of  inputs J and K is same as the S and R inputs of the S-R flip flop. The letter J stands for SET and the letter K stands for CLEAR.

When both the inputs J and K have a HIGH state, the flip-flop switch to the complement state. So, for a value of Q = 1, it switches to Q=0 and for a value of Q = 0, it switches to Q=1.

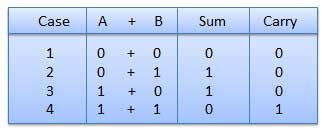
The circuit includes two 3-input AND gates. The output Q of the flip flop is returned back as a feedback to the input of the AND along with other inputs like K and clock pulse [CP]. So,  if the value of CP is ‘1’, the flip flop gets a CLEAR signal and with the condition that the value of Q was earlier 1. Similarly output Q’ of the flip flop is given as a feedback to the input of the AND along with other inputs like J and clock pulse [CP]. So the output becomes SET when the value of CP is 1 only if the value of Q’ was earlier 1.

The output may be repeated in transitions once they have been complimented for J=K=1 because of the feedback connection in the JK flip-flop. This can be avoided by setting a time duration lesser than the propagation delay through the flip-flop. The restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction.

OR

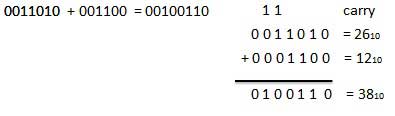
## ) Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.



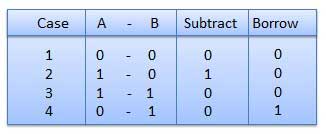
In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

### Example − Addition



## Binary Subtraction

**Subtraction and Borrow**, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

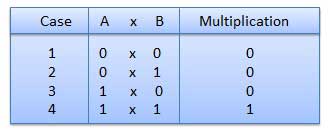


### Example − Subtraction

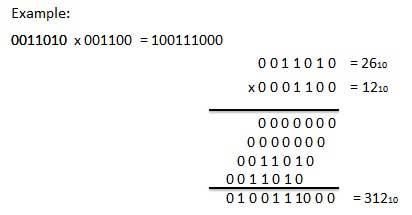


## Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.



### Example − Multiplication



## Binary Division

Binary division is similar to decimal division. It is called as the long division procedure.

### Example − Division

