Q1. What is a Logic Gate? Which are the three elementary (basic) gates?
       Which accepts one or more inputs and one output signal – NOT, AND, OR

Q2. Explain the following operators: not, and, or, xor, nand, nor, xnor (give Gate, Operator, T/Table, Circuit and name)
       Note: TT not given below, XNOR can also be represented by a dot inside a circle.
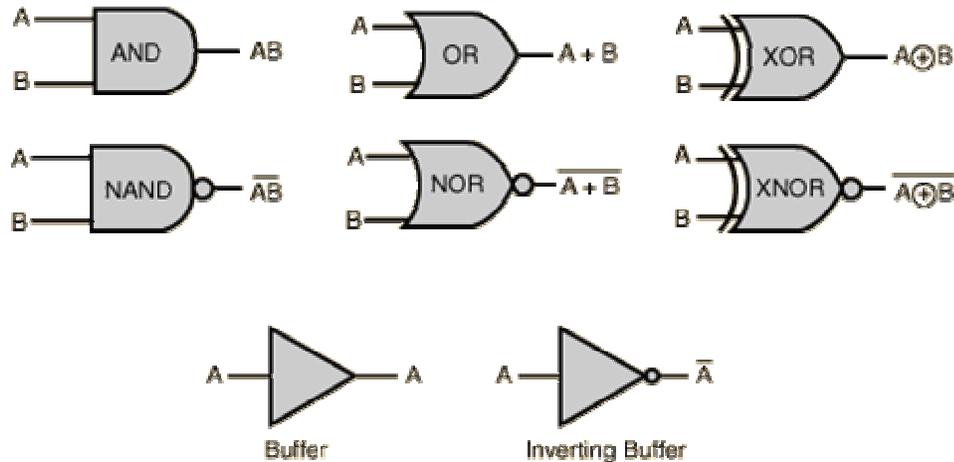
Notes: Inverting buffer simple means the NOT gate, Buffer is just a gate which keeps value of the variable intact, the symbol for XNOR is also $\odot$ (e.g. a$\odot$b)

Q3. Explain the following operators with their equivalents in basic gates? $\oplus$ $\ominus$ $\uparrow$ $\downarrow$.

$\oplus$	XOR	A'B + AB'	(A+B).(A'+B') (Derived after selecting 1/0 from truth tables)
$\ominus$	XNOR	A'B'+AB	(A'+B).(A+B')
$\uparrow$	NAND	derive SOP or POS forms from the truth tables	
$\downarrow$	NOR	derive SOP or POS forms from the truth tables	

Q4. Give three common errors in logic gate diagrams.
       1. Not inputting more than 2 variables in gates(Other than NOT)
       2. Not assuming NOT to be available as input
       3. Diagrammatically icreasing the size of gates to fit input lines

Q5. Show algebraically and diagrammatically how are NAND and NOR universal gates.
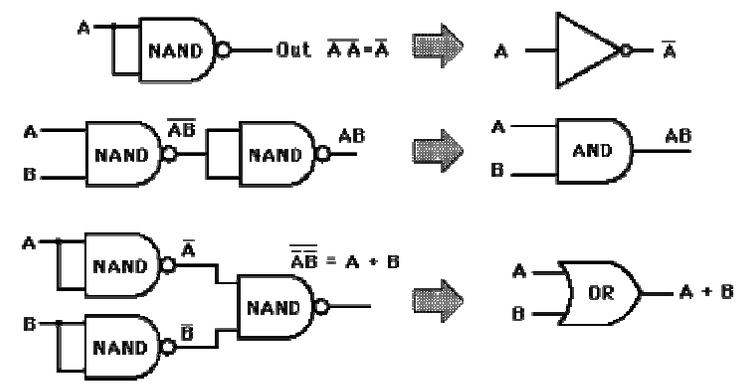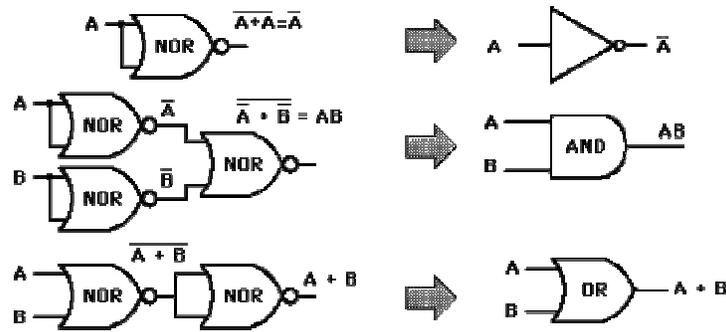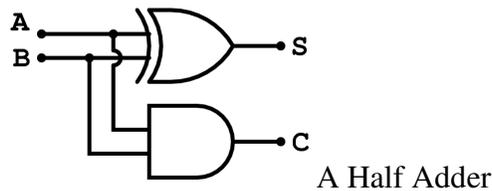
Q6. Give the design rule for NAND-TO-NAND and NOR-TO-NOR logic network.
1. Convert to SOP for NAND-TO-NAND and POS for NOR-TO-NOR (Canonical not required)
2. Replace all the gates with NAND (or NOR)

Q7. What are Adders and where are they applied? Name their two types and give their circuits.
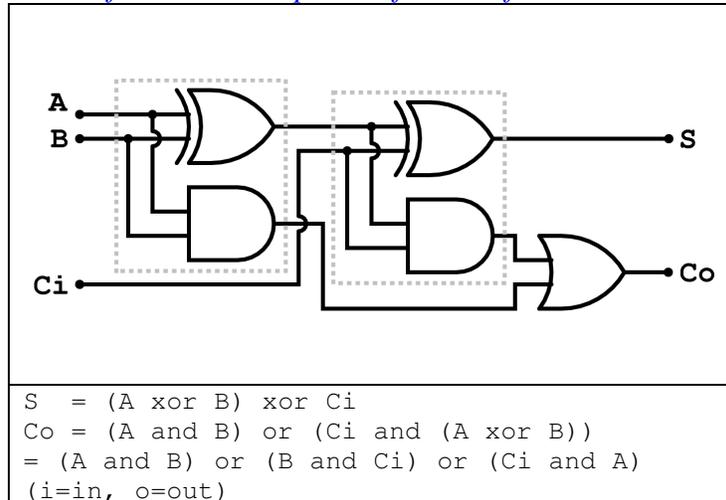Used to add – ALU – Half Adder & Full Adder.

A Half Adder

Please note that in the full adder circuit given below you only one XOR gate accepting all the three inputs will be correct. There is some extra labelling also which can be ignored.
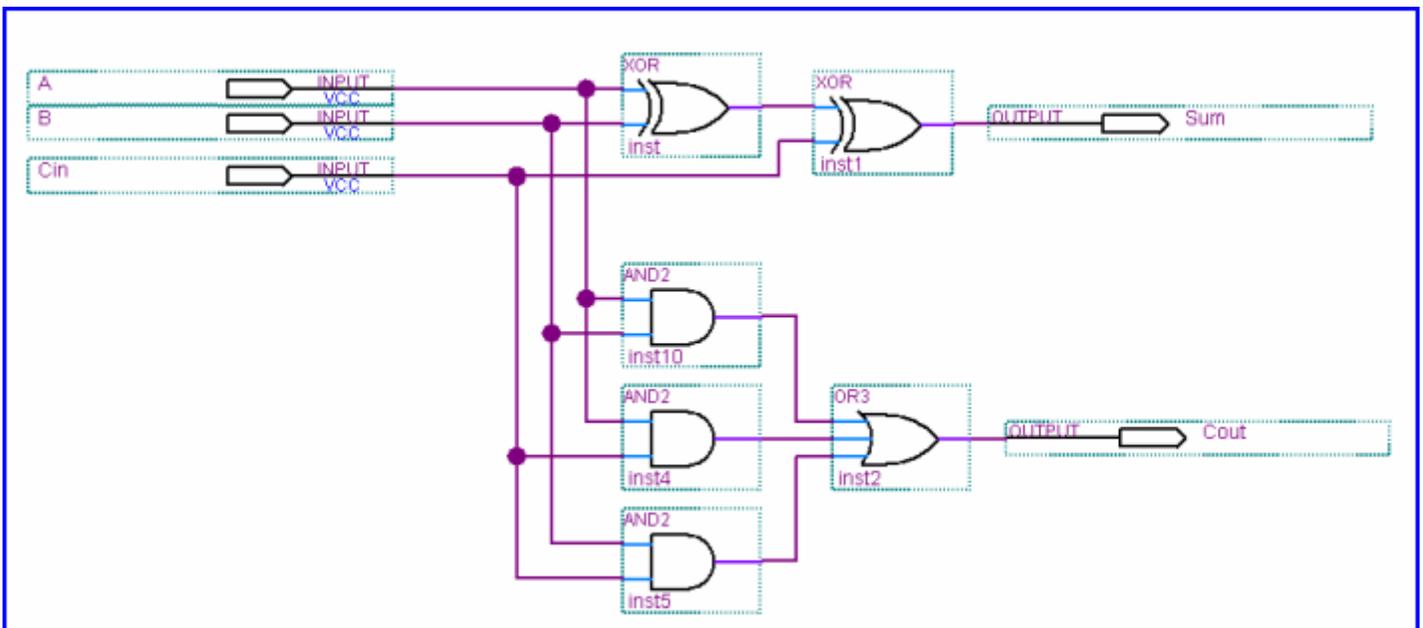
Alternate It can also be seen that a full adder comprises of two half-adders and an OR gate:


```
S  = (A xor B) xor Ci
Co = (A and B) or (Ci and (A xor B))
= (A and B) or (B and Ci) or (Ci and A)
(i=in, o=out)
```

Q8. What are Encoders and where are they applied? Name their three types and give their circuits.
   Which convert to binary – Keyboards & other input devices – Oct to Bin; Dec to Bin and Hex to Bin.
   (There are OR gates at the bottom)
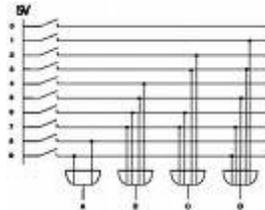
Q9. What are Decoders and where are they applied? Name their two types and give their circuits.
 Which convert binary to some other – Output devices – 2 to 4(Shown below- labelling might differ from book), 4 to 16.
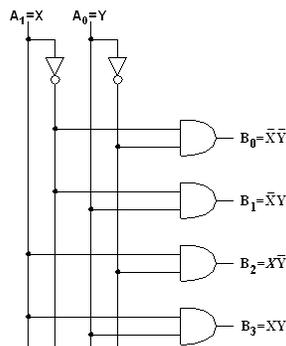
Q10. What is a Multiplexer? How is it abbreviated? What is its other name? What is multiplexing?
   Converts many lines to one – MUX – "Data selector" – Transmitting a large number of information units over a smaller channel. (Please see the book for the diagram).

Q11.   (a) Give three applications of a MUX.
             1. Routing several words to the same destination.
             2. Parallel to serial conversion.
             3. Directly implement any 2 level logic circuit on *n* variables. (as asked in Q13 & Q14).

(b) What is a De-multiplexer?
        That performs de-multiplexing i.e. directing single input into many output lines.

Q12. Implement a full adder circuit with a decoder.
Q13. Give the steps and an example of implementing a 3 variable Boolean function using a MUX.
Q14. Give the steps and an example of implementing a 4 variable Boolean function using a MUX.
(Ignore only for this test)

Q15. Give block diagrams of- Adders, Encoders, Decoder, MUX.
        Please refer the book.

Important: Solved problems 8,9,10 & 11.

Extra questions

Q1. Output $a \oplus b$ without using the exclusive OR gate.
        Hint: use XOR's sop or pos equivalent

Q2. Express using (i) NOR (ii) NAND :   A.(B+C)
      I.   (A+A)(B+C)   – now replace all gates
      II.   AB+AC   – now replace all gates

Q3. Design a decoder without using any NOT gate.
        Hint: use any universal gate

Q4. Design a logic gate circuit for an alarm which should hoot if a runner wins 2 consecutive races out of 3.

a	b	c	F	minterms
0	0	0		
0	0	1		
0	1	0		
0	1	1	1	a'bc
1	0	0		
1	0	1		
1	1	0	1	abc'
1	1	1	1	abc

        Now reduce the sop obtained from the minters and make a circuit diagram.

Q5. Design a circuit for a half subtractor (Input a and b, Output – Difference & Borrow)

a	b	diff	Borr	
0	0	0		
0	1	1	1	
1	0	1		
1	1	0		

        Diff= $a \oplus b$,   Borrow= a' .b, Now design a circuit with 2 outputs(as in adders) for difference and borrow.
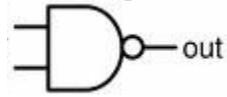
Q6. Implement a MUX in an encoder.
Hint: - Direct all the output signals at the bottom to a single AND gate.

Q7. Design the NOR gate using the least number of NAND gates.

$[(A{\uparrow}A) \uparrow (B{\uparrow}B)] \uparrow [(A{\uparrow}A) \uparrow (B{\uparrow}B)]$

Gates can further be reduced if NOT is assumed to be available as input.

Q8. What will the gate shown below give if A and 1 are passed through it.

$(a{\uparrow}1)=a'+1'=a'+0=a'$

Gives a NOT gate

Q9. Simplify $((A{\uparrow}A)\uparrow(B{\uparrow}B)) \uparrow ((A{\uparrow}B)\uparrow(A{\uparrow}B))$

$=(A+B) \uparrow (A.B)$

$=(A+B)' + (A.B)'$

$=(A'.B')+(A'+B')$

$=(A'+B')+(A'.B')$

$=(A'+B'+A') . (A'+B'+B')$

$=(A'+B')(A'+B')$

$=A'+B'$

Q10. Express the following using logic gates (a) De Morgan's first theorem (b) Implies

Hint: (a) Make a gate showing (a+b)' and outputting a'.b'

(b) Make a fate showing a'+b

[END]