YASH SARANG

DGAD /47

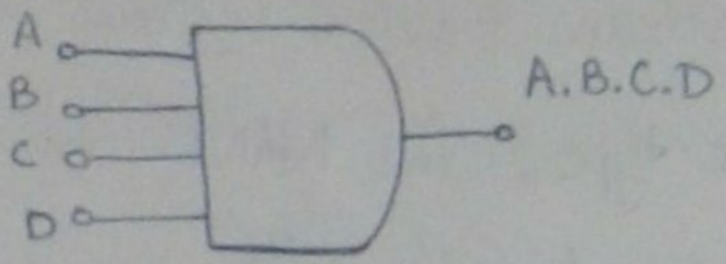
DLCOA

EXPERIMENT / ASSIGNMENT - 2

Sarangyash

Teacher's Sign :

Assignment 2

① 

A	B	C	D	A.B.C.D
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(2) A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND Gate & NOR Gate are universal gates. This is advantageous, since NAND & NOR gates are economical & easier to fabricate & are the basic gates using the in all IC digital log'c families.

(3) AND → Boolean expression of AND gate $Y = A.B$.

OR → Boolean express is $Y = A + B$.

NAND → Boolean expression is $Y = \overline{A.B}$

NOR → Boolean expression is $Y = \overline{A+B}$.

(4) De Morgan's theorem describes the equivalence between gates with inverted inputs & gates with inverted outputs. Simply put a NAND gate is equivalent to Negative-OR gate and NOR gate is equivalent to a Negative-AND gate.

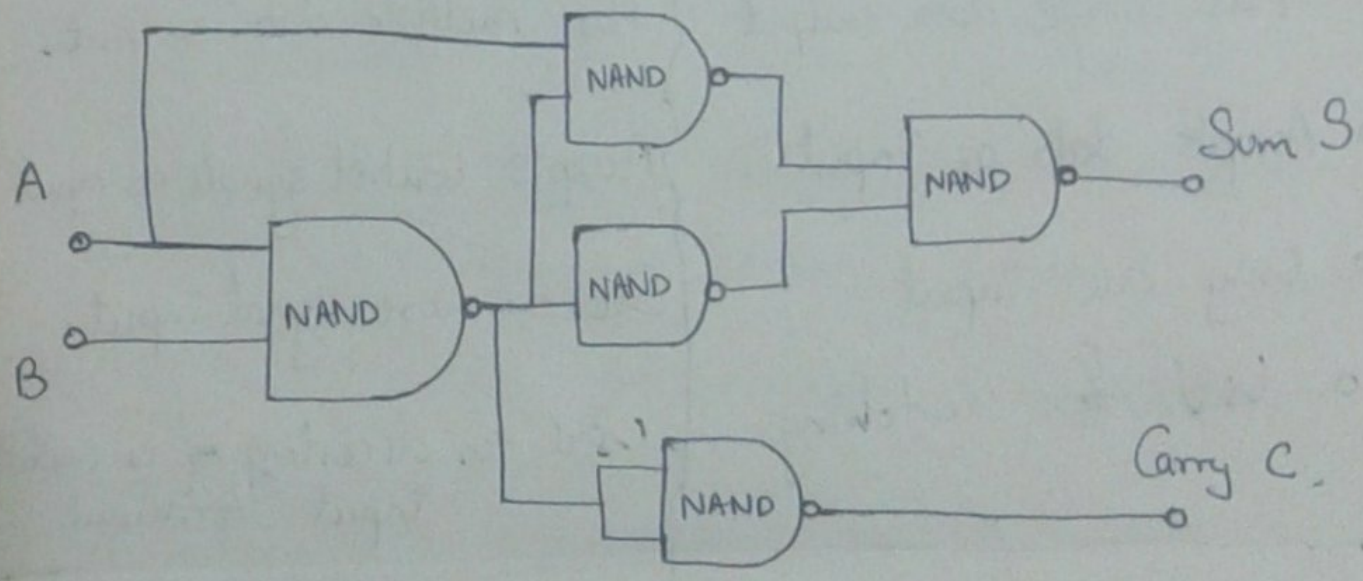
(5) NAND gate is considered as a universal gate, it can be designed of any digital circuitry.

The minimum number of NAND gates required to design a half adder is 5.

The first NAND gate takes the inputs which are the two 1-bit numbers. The resultant NAND operated inputs will be again given to as input to 3- NAND gates along with the original input.

Out of the 3 considered NAND gates, the third NAND gate will generate the carry bit.

The NAND operation can be understood more clearly with the given diagram.



⑥ Demultiplexer (DEMUX)

It is a combinational logic circuit designed to switch one common input line to one of several separate output line. The data distributor is commonly known as Demultiplexer.

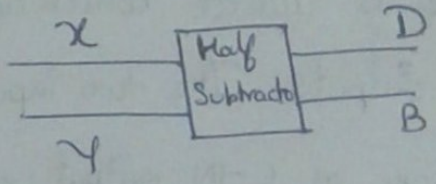
Multiplexer	Demultiplexer
• Multiplexer process the digital info from various source into a single source.	It recieves digital info from a single source and converts into several sources
• Known as data selector	Known as data distributor.
• It is a digital switch.	It is a digital circuit.
• Has multiple data input	Has single data input.
• Has single data output	Has multiple data output.
• Accepts data as input	Accepts control signals as input
• Only one input	One or more signal input.
• Used for switching	Used for decoding of encoded input terminal.

7) Define half subtractor & full subtractor.

(i) Half subtractor,

It is a combined circuit which is used to perform subtraction of two bits. It has two inputs, x (minhead) & Y (subtrahand) and two outputs D (difference) & B (borrow).

Diagram / Symbol:

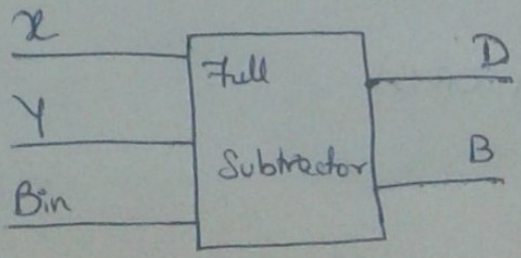


Truth table:

x	Y	D	B
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

(ii) Full subtractor

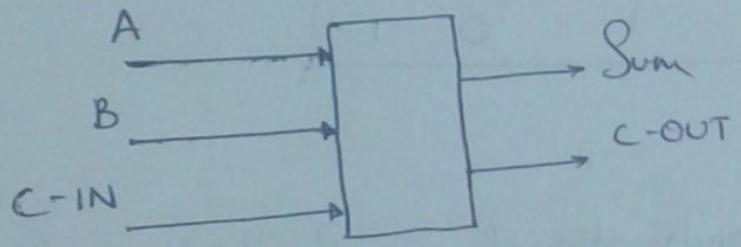
It is a combinational circuit that performs subtraction including 3 bits namely minuend, subtraend and borrowin. The logic symbol & truth table are shown.



Truth table:

X	Y	Bin	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

(8) Full adder is added which add 3 inputs & produces 2 outputs. The two inputs are A & B. and input carry as C-IN output ~~carry~~ carry is designed as C-OUT & the normal output is designated as S which is SUM.

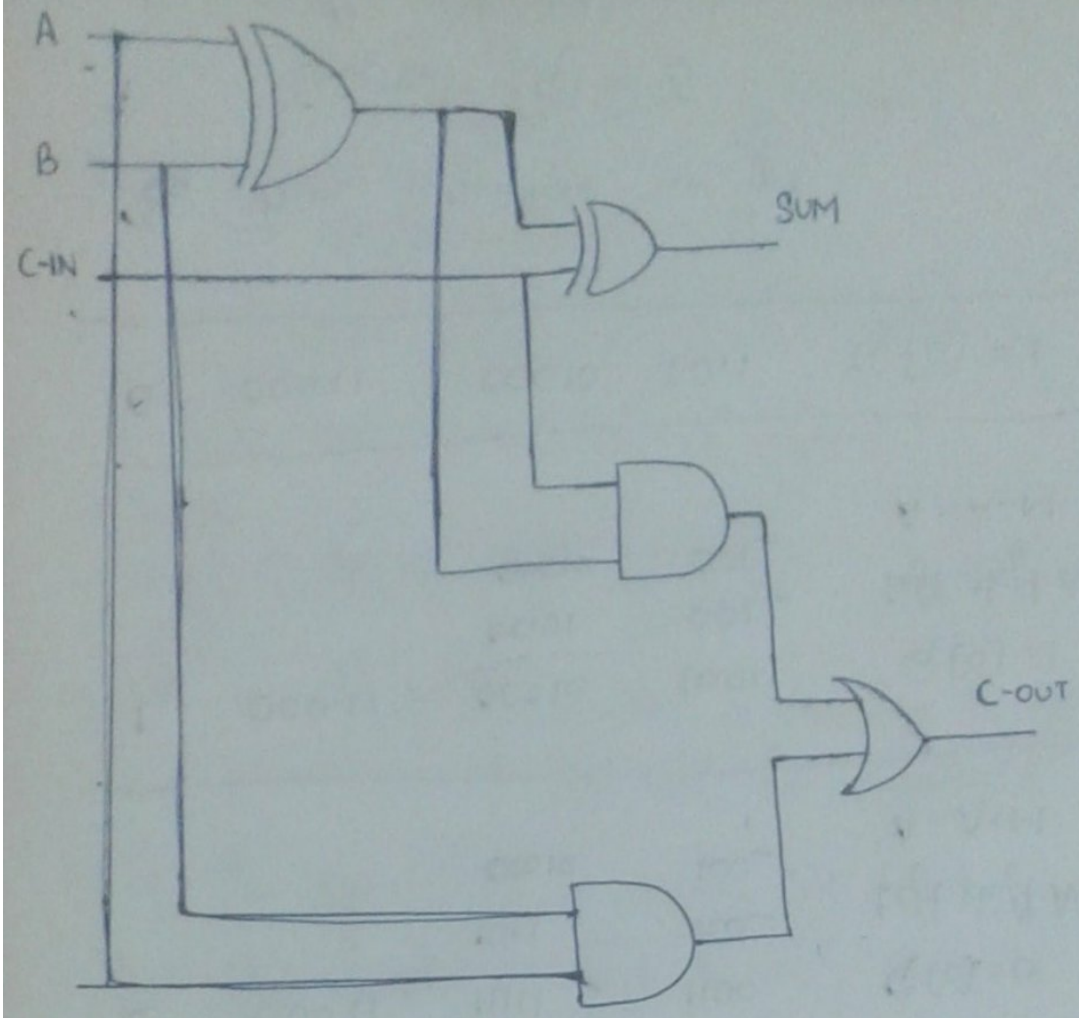


Logical Expression.

$A'B'C\text{-}IN + A'BC\text{-}IN' + AB'C\text{-}IN' + ABC\text{-}IN$

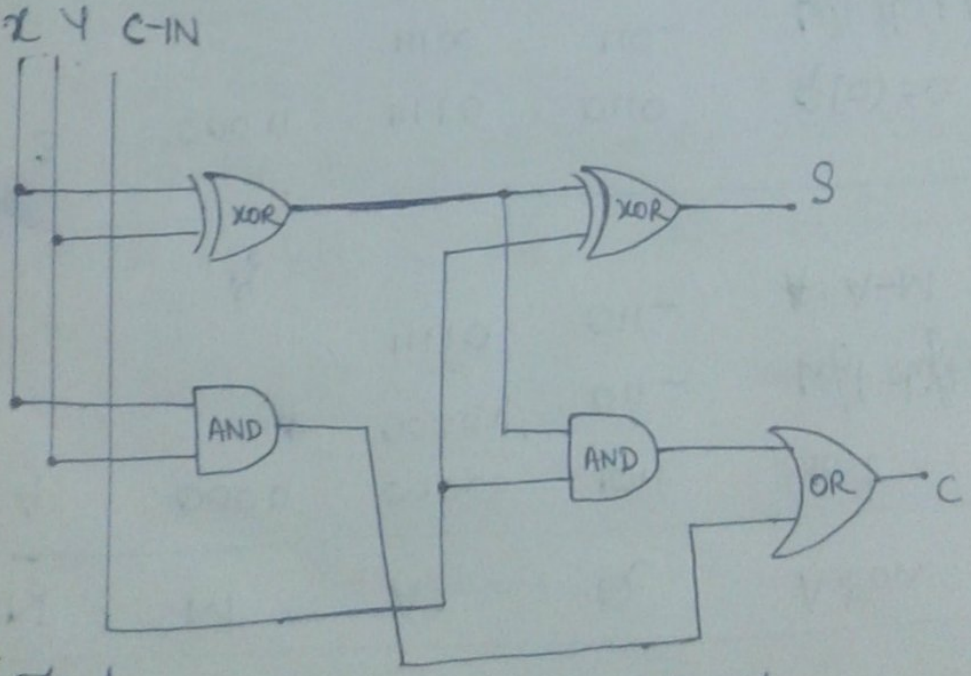
$= C\text{-}IN(A'B' + AB) + C\text{-}IN(A'B + AB')$

$= C\text{-}IN \text{ XOR } (A \text{ XOR } B)$.



Full adder logic logic circuit.

(9)



First Half adder

Second Half adder

(10) Algorithm:- Register, M ← Divisor

Register, Q ← Dividend.

Register, AC ← 0

Register, C ← 0 (1-bit carry)

Count ← n [size of register]

} Initial values

Restoring:

1. Shift C, AC, Q, left by 1 bit.

2) AC ← AC-M [subtraction & checking the sign can be used for comparison]

3) If the sign of AC is negative (i.e C=1) set Q_0 & add M back to AC, otherwise set Q_0 to 1.

Non Restoring:

1) Do the following n times.

If the sign of AC is positive (C=0) then shift C, AC & Q left one bit & subtract M from AC.

else

shift C, AC & Q left one bit position & add M to AC.

If the sign of AC is positive then

Set q_0 to 1

else

Set q_0 to 0

2) If the step of AC is positive, then add M to AC.

Non Restoring Division Algorithm

N	M	A	Q	Action
4	00011	00000	1011	Start
		00001	011_	Left shift AQ
		11110	011_	A = A-M
3	00011	11110	0110	Q[0] = 0
		11100	110_	Left Shift AQ
		11111	110_	A = A+M.
2	00011	11111	1100	Q[0] = 0
		11111	100_	Left shift AQ
		00010	100_	A = A+M
1	00011	00010	1001	Q[0] = 1
		00001	001_	Left shift AQ
		00010	001_	A = A-M
0	00011	00010	0011	Q[0] = 1

From calculations we get,

Quotient (Q) = 3

Remainder (A) = 2.

Examples: Perform Division Restoring Algorithm.

Dividend = 11 , Divisor = 3.

Restoring Division Algorithm

n	M	A	Q	Operation
4	00011	00000	1011	initialize
	00011	00001	011_	shift left AQ
	00011	11110	011_	A = A-M
	00011	00001	0110	Q[0] = 0 & restore A
3	00011	00010	110_	shift left AQ
	00011	11111	110_	A = A-M
	00011	00010	1100	Q[0] = 0
2	00011	00101	100_	Shift left AQ
	00011	00010	100_	A = A-M
	00011	00010	1001	Q[0] = 1.
1	00011	00101	001_	shift left AQ
	00011	00010	001_	A = A-M
	00011	00010	0011	Q[0] = 1

Register Q contains the quotient i.e 3. and register A contains remainder 2.