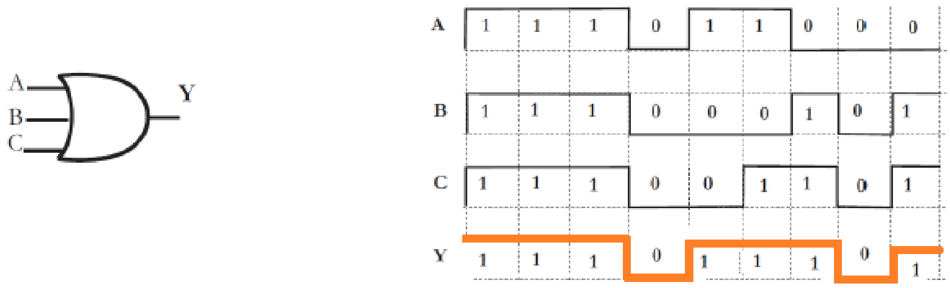
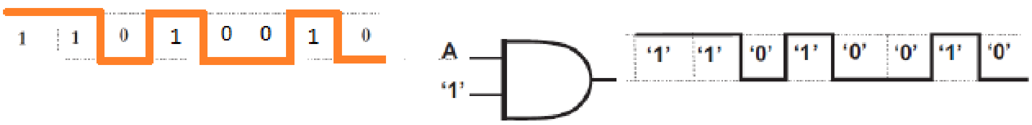
[2.1] Given the pulse train sent to the input A of an OR gate, draw its output pulse train.

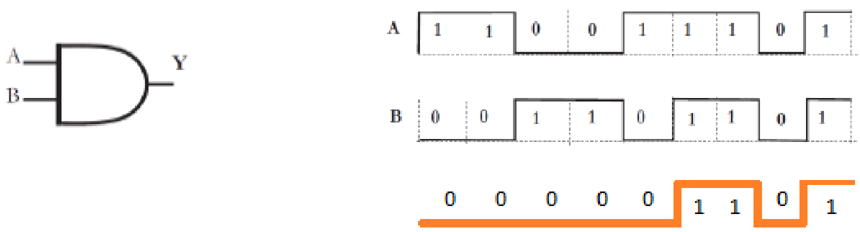
[2.2] Given the pulse train sent to the input A, B and C of an OR gate, draw its output pulse train.



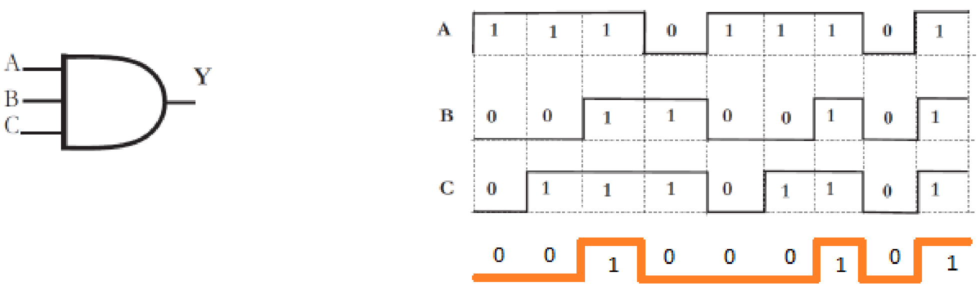
[2.3] The output of an AND gate shows the following pulse train. Draw the corresponding signal at input *A*.



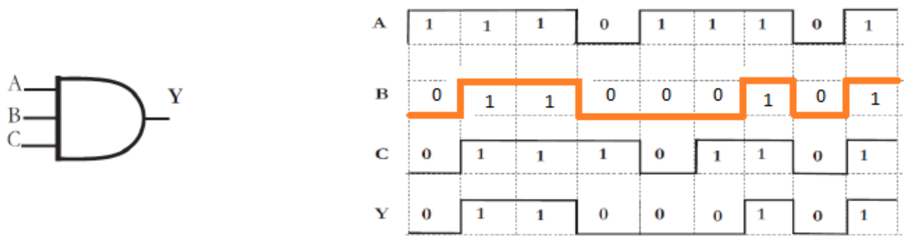
[2.4] Given the following pulse trains sent to the 2 inputs A and B of an AND gate, find the output pulse train.



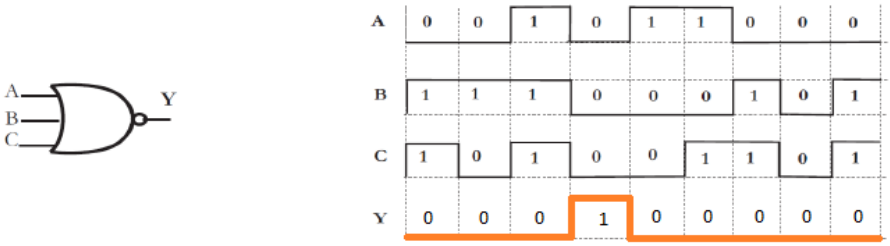
[2.5] Given the following pulse trains sent to the 3 inputs A, B and C of an AND gate, find the output pulse train.



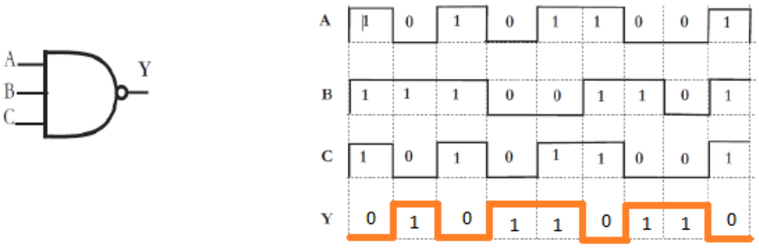
[2.6] The output Y and the inputs A and C of an AND gate have the following pulse trains. Draw the corresponding signal at input B.



[2.7] Given the following pulse trains sent to the 3 inputs A, B and C of a NOR gate, find the output pulse train.



[2.8] Given the following pulse trains sent to the 3 inputs A, B and C of a NAND gate, find the output pulse train.



[2.9] Demonstrate the validity of the following identities by means of truth tables :
(a) DeMorgan's theorem for three variables: $(x + y + z)' = x'y'z'$ and $(xyz)' = x' + y' + z'$

x y z	$x + y + z$	$(x + y + z)'$	x'	y'	z'	$x'y'z'$
0 0 0	0	1	1	1	1	1
0 0 1	1	0	1	1	0	0
0 1 0	1	0	1	0	1	0
0 1 1	1	0	1	0	0	0
1 0 0	1	0	0	1	1	0
1 0 1	1	0	0	1	0	0
1 1 0	1	0	0	0	1	0
1 1 1	1	0	0	0	0	0

x y z	(xyz)	$(xyz)'$	x'	y'	z'	$x' + y' + z'$
0 0 0	0	1	1	1	1	1
0 0 1	0	1	1	1	0	1
0 1 0	0	1	1	0	1	1
0 1 1	0	1	1	0	0	1
1 0 0	0	1	0	1	1	1
1 0 1	0	1	0	1	0	1
1 1 0	0	1	0	0	1	1
1 1 1	1	0	0	0	0	0

(b) The distributive law: $x + yz = (x + y)(x + z)$

x y z	x+yz	(x + y)	(x + z)	(x + y)(x + z)
0 0 0	0	0	0	0
0 0 1	0	0	1	0
0 1 0	0	1	0	0
0 1 1	1	1	1	1
1 0 0	1	1	1	1
1 0 1	1	1	1	1
1 1 0	1	1	1	1
1 1 1	1	1	1	1

(c) The distributive law: $x(y + z) = xy + xz$

x y z	x(y + z)	xy	xz	xy + xz
0 0 0	0	0	0	0
0 0 1	0	0	0	0
0 1 0	0	0	0	0
0 1 1	0	0	0	0
1 0 0	0	0	0	0
1 0 1	1	0	1	1
1 1 0	1	1	0	1
1 1 1	1	1	1	1

(d) The associative law: $x + (y + z) = (x + y) + z$

x y z	y + z	x + (y + z)	(x + y)	(x + y) + z
0 0 0	0	0	0	0
0 0 1	1	1	0	1
0 1 0	1	1	1	1
0 1 1	1	1	1	1
1 0 0	0	1	1	1
1 0 1	1	1	1	1
1 1 0	1	1	1	1
1 1 1	1	1	1	1

(e) The associative law and $x(yz) = (xy)z$

x y z	yz	x(yz)	xy	(xy)z
0 0 0	0	0	0	0
0 0 1	0	0	0	0
0 1 0	0	0	0	0
0 1 1	1	0	0	0
1 0 0	0	0	0	0
1 0 1	0	0	0	0
1 1 0	0	0	1	0
1 1 1	1	1	1	1

[2.10] Simplify the following Boolean expressions to a minimum number of literals :

(a) $xy + xy'$

$=xy + xy' = x(y + y') = x$

(b) $(x + y) (x + y')$

$=(x + y)(x + y') = x + yy' = x(x +y') + y(x + y') = xx + xy' + xy + yy' = x$

(c) $xyz + x'y + xyz'$

$=xyz + x'y + xyz' = xy(z + z') + x'y = xy + x'y = y$

(d) $(A + B)' (A' + B')'$

$=(A + B)'(A' + B') = (A'B')(A B) = (A'B')(BA) = A'(B'BA) = 0$

(e) $(a + b + c')(a' b' + c)$

= a a' b' + a c + b a' b' + b c + c' a' b' + c' c

= (a a') b' + a c + (b b') a' + b c + c' a 'b' + 0

=b'(0) + a c + a'(0) + b c + c' a' b'

= a c + b c + c' a' b'

[2.11] Simplify the following Boolean expressions to a minimum number of literals :

(a) $ABC + A'B + ABC'$

$ABC + A'B + ABC' = AB + A'B = B$

(b) $x'yz + xz$

$x'yz + xz = (x'y + x)z = z(x + x')(x + y) = z(x + y)$

(c) $(x + y)'(x' + y')$

$(x + y)'(x' + y') = x'y'(x' + y') = x'y'$

(d) $xy + x(wz + wz')$

$xy + x(wz + wz') = x(y + wz + wz') = x(w + y)$

[2.12] Reduce the following Boolean expressions to the indicated number of literals :

(a) $A'C' + ABC + AC'$ to three literals

$A'C' + ABC + AC' = C' + ABC = (C + C')(C' + AB) = AB + C'$

(b) $(x'y' + z)' + z + xy + wz$ to three literals

$(x'y' + z)' + z + xy + wz = (x'y')'z' + z + xy + wz = [(x + y)z' + z] + xy + wz =$
$= (z + z')(z + x + y) + xy + wz = z + wz + x + xy + y = z(1 + w) + x(1 + y) + y = x + y + z$

(c) $A'B(D' + C'D) + B(A + A'CD)$ to one literal

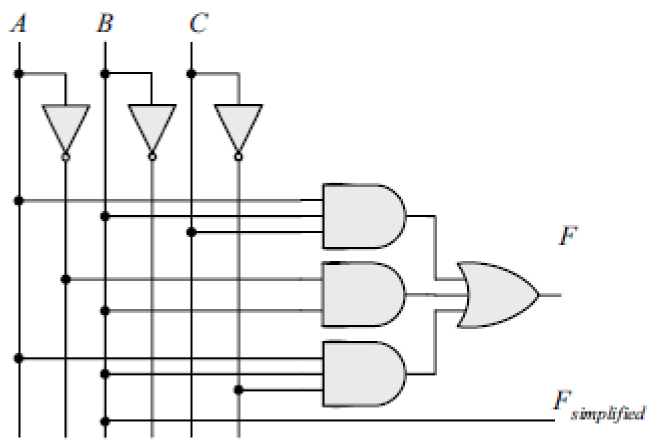
$A'B(D' + C'D) + B(A + A'CD) = B(A'D' + A'C'D + A + A'CD)$
$= B(A'D' + A + A'D(C + C') = B(A + A'(D' + D)) = B(A + A') = B$

(d) $(A' + C)(A' + C')(A + B + C'D)$ to four literals

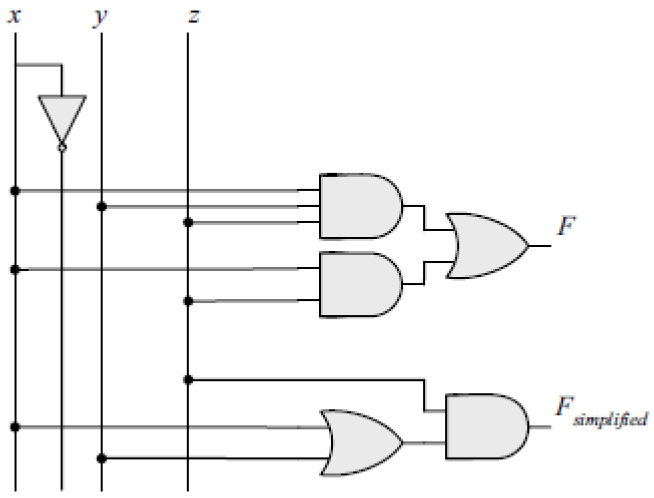
$(A' + C)(A' + C')(A + B + C'D) = (A' + CC')(A + B + C'D) = A'(A + B + C'D)$
$= AA' + A'B + A'C'D = A'(B + C'D)$

[2.13] Draw logic diagrams of the circuits that implement the original and simplified expressions in Problem 2.11.

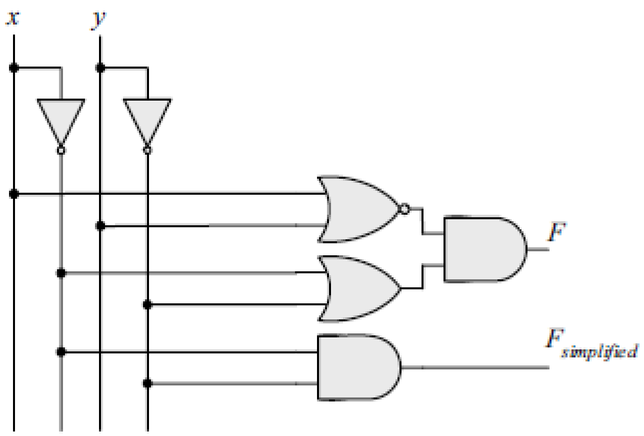
a)



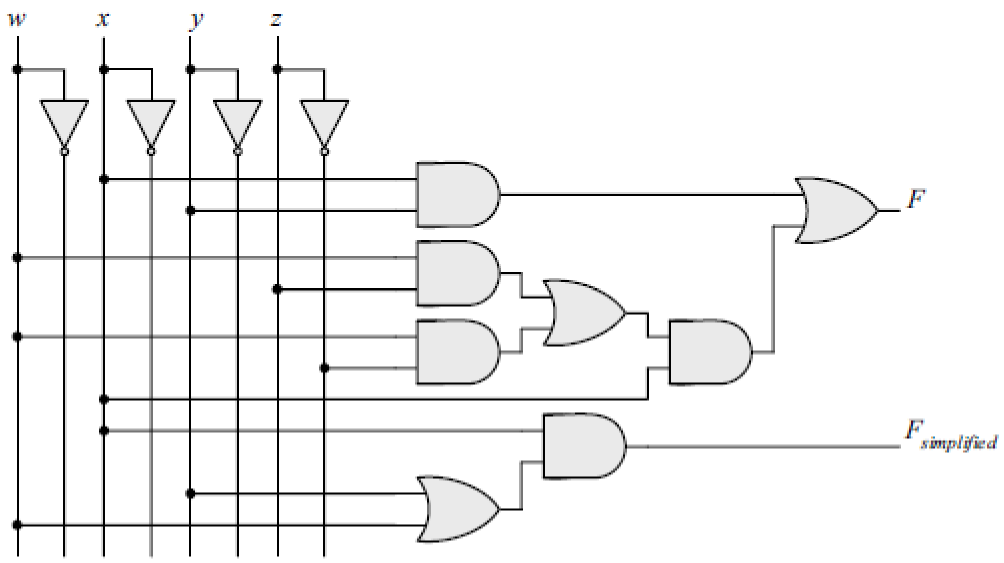
b)



c)



d)



[2.14] Find the complement of the following expressions :

(a) $xy' + x'y$

$F' = (xy' + x'y)' = (xy')'(x'y)' = (x' + y)(x + y') = xy + x'y'$

(b) $(a + c)\ (a + b')\ (a' + b + c')$

$F' = (a'c') + (a'b) + (\ ab'c\ )$

(c) $z + z'(v'w + xy)$

$F' = z'\ z + (\ v + w'\ )\ (x' + y'\ )$

[2.15] We can perform logical operations on strings of bits by considering each pair of corresponding bits separately (called bitwise operation). Given two eight – bit strings A =10110001 and B =10101100, evaluate the eight – bit result after the following logical operations :

(a) AND (b) OR (c) XOR

A = 1011_0001

B = 1010_1100

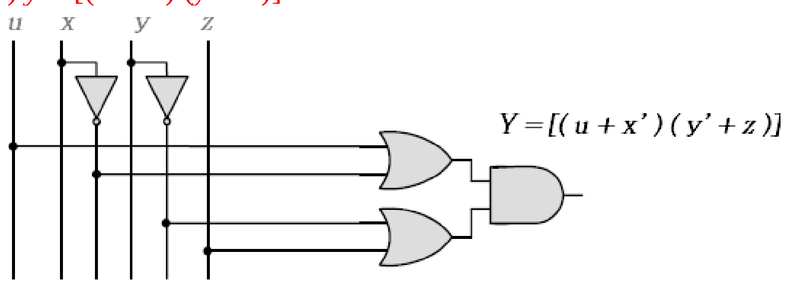
(a) *A AND B = 1010_0000*

(b) *A OR B = 1011_1101*

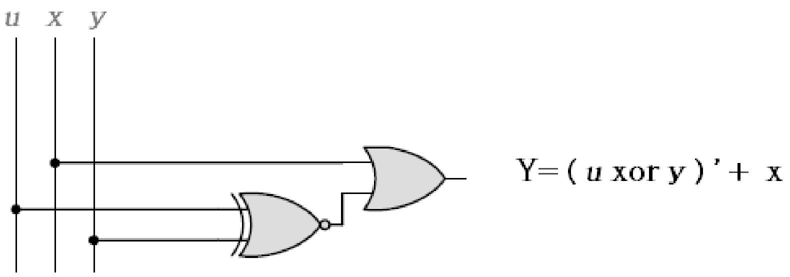
(c) *A XOR B = 0001_1101*

[2.16] Draw logic diagrams to implement the following Boolean expressions :

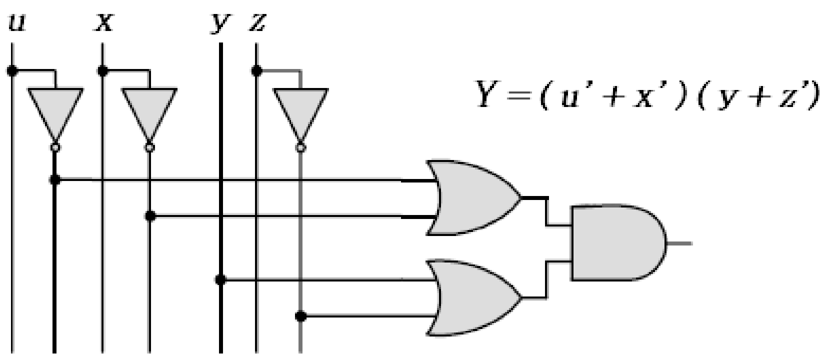
(a) $y = [(u + x')\ (y' + z)]$



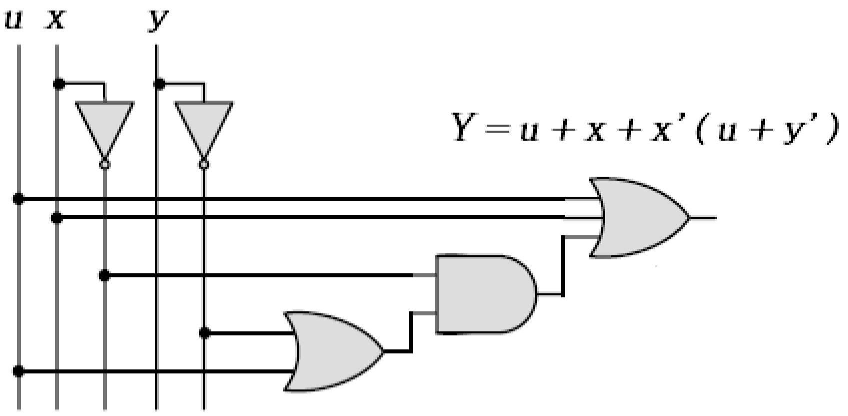
(b) $y = (u \otimes y)' + x$



(c) $y = (u' + x')\ (y + z')$



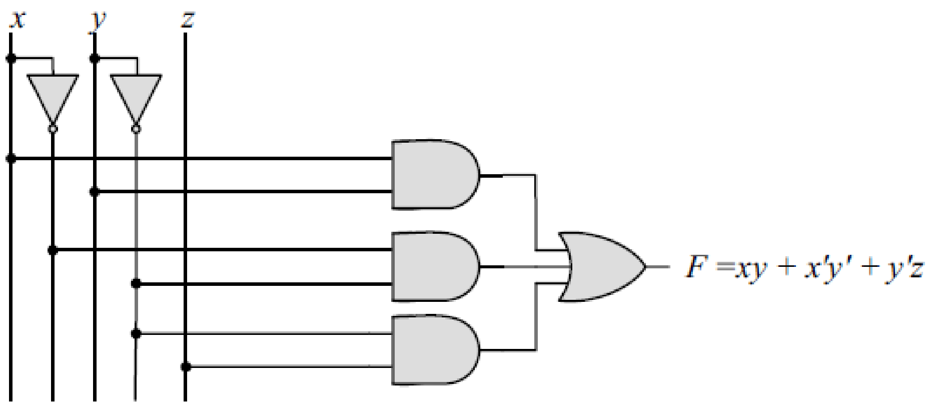
(f) $y = u + x + x'(u + y')$



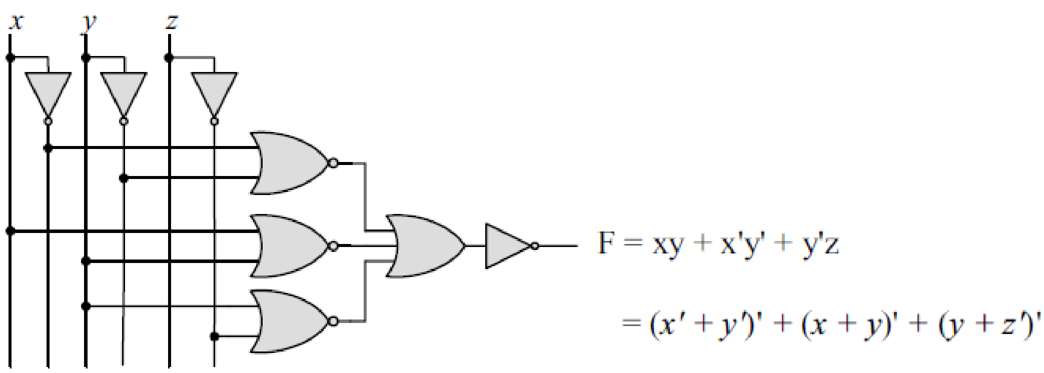
[2.17] Implement the Boolean function

$F = xy + x'y' + y'z$

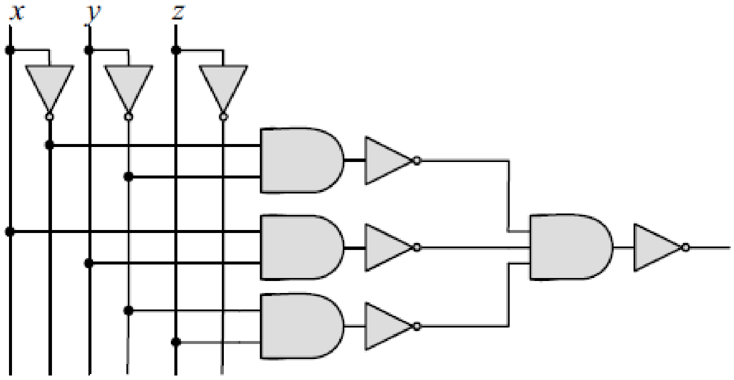
(a) With AND, OR, and inverter gates



(b) With OR and inverter gates



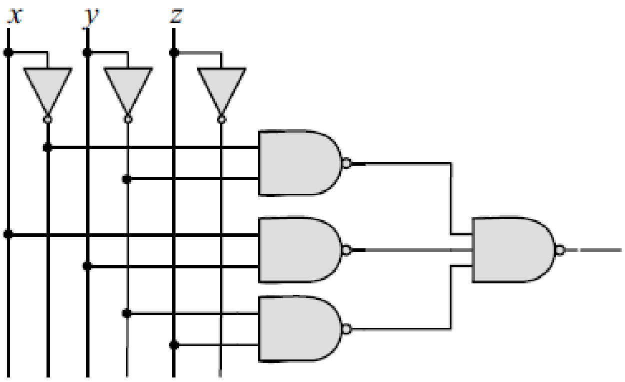
(c) With AND and inverter gates



$F = xy + x'y' + y'z$

$= [(xy)'(x'y')'(y'z)']'$

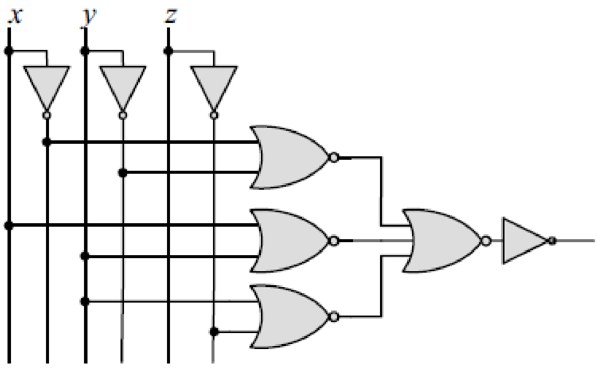
(d) With NAND and inverter gates



$F = xy + x'y' + y'z$

$= [(xy)'(x'y')'(y'z)']'$

(e) With NOR and inverter gates



$F = xy + x'y' + y'z$

$= (x' + y')' + (x + y)' + (y + z')'$

[2.18] Simplify the following Boolean functions T_1 and T_2 to a minimum number of literals :

A	B	C	T1	T2
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

(a) $T1 = A'B'C' + A'B'C + A'BC' = A'B'(C' + C) + A'C'(B' + B) = A'B' + A'C' = A'(B' + C')$
(b) $T2 = T1' = A'BC + AB'C' + AB'C + ABC' + ABC$
$= BC(A' + A) + AB'(C' + C) + AB(C' + C)$
$= BC + AB' + AB = BC + A(B' + B) = A + BC$

$\sum(3, 5, 6, 7) = \prod(0,1, 2, 4)$

$T1 = A'B'C' + A'B'C + A'BC'$

$A'B' \quad A'C'$

$T1 = A'B' A'C' = A'(B' + C')$

$T2 = A'BC + AB'C' + AB'C + ABC' + ABC$

$AC' \quad AC$

BC

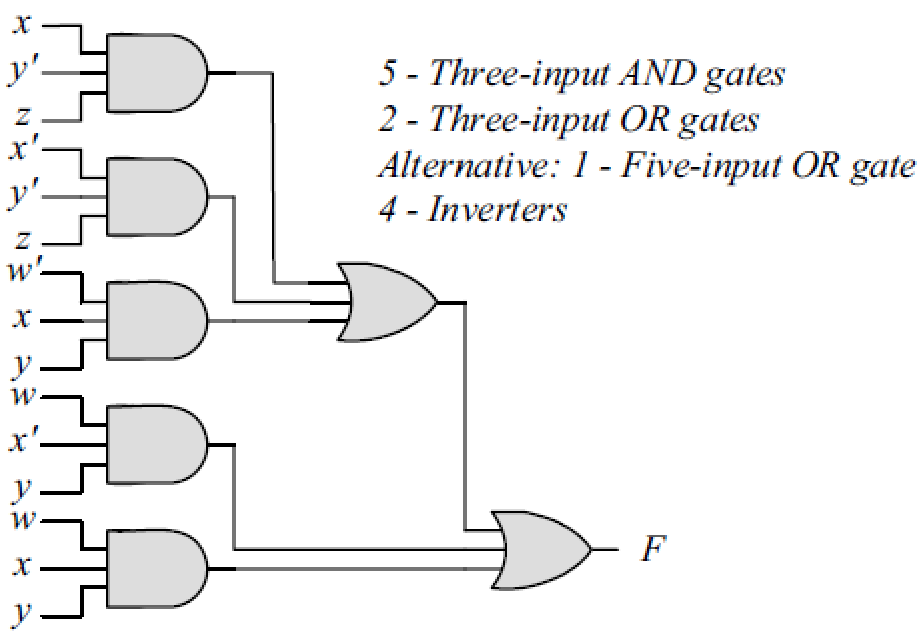
[2.19] Obtain the truth table of the following functions, and express each function in sum-of-minterms and product-of-maxterms form :
(a) $(b + cd)(c + bd)$
(a) $(b + cd)(c + bd) = cd + cb + cbd + bd = \sum(3, 5, 6, 7) = \prod(0, 1, 2, 4)$
(b) $(cd + b'c + bd')(b + d)$
$(cd + b'c + bd')(b+d) = bcd + bb'c + bbd' + cdd + b'cd + bd'd$
$= bcd + 0 + bd' + cd + b'cd + 0$
$= \sum(3,4,6,7) = \prod(0,1,2,5)$

[2.20] For the Boolean function
$F = xy'z + x'y'z + w'xy + wx'y + wxy$
(a) Obtain the truth table of F.
$F = xy'z + x'y'z + w'xy + wx'y + wxy$

w x y z	F
0 0 0 0	0
0 0 0 1	1
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	1
0 1 1 0	1
0 1 1 1	1
1 0 0 0	0
1 0 0 1	1
1 0 1 0	1
1 0 1 1	1
1 1 0 0	0
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1

$F = \sum(1, 5, 6, 7, 9, 10\ 11, 13, 14, 15)$

(b) Draw the logic diagram, using the original Boolean expression.



(c) Use Boolean algebra to simplify the function to a minimum number of literals.
$F = xy'z + x'y'z + w'xy + wx'y + wxy = y'z + xy + wy = y'z + y(w + x)$

[2.21] Express the following function as a sum of minterms and as a product of maxterms :
$F(A, B, C, D) = B'D + A'D + BD$

$F = B'D + A'D + BD$

ABCD	ABCD	ABCD
-B'-D	A'--D	-B-D
0001 = 1	0001 = 1	0101 = 5
0011 = 3	0011 = 3	0111 = 7
1001 = 9	0101 = 5	1101 = 13
1011 = 11	0111 = 7	1111 = 15

$F = \sum(1, 3, 5, 7, 9, 11,13, 15) = \prod(0, 2, 4, 6, 8, 10, 12, 14)$

[2.22] Express the complement of the following functions in sum-of-minterms form :
(a) $F(A,B ,C, D) = \sum(2, 4, 7, 10, 12, 14)$
$F(A,B ,C, D) = \sum(2, 4, 7, 10, 12, 14) = \prod(0,1,3,5,6,8,9,11,13,15)$
(b) $F (x, y, z) = \prod(3, 5, 7)$
$F (x, y, z) = \prod(3, 5, 7) = \sum (3,5,7)$

[2.23] Convert each of the following to the other canonical form :

(a) $F(x, y, z) = \sum(1, 3, 5)$
$F(x, y, z) = \sum(1, 3, 5) = \prod(0,2,4,6,7)$

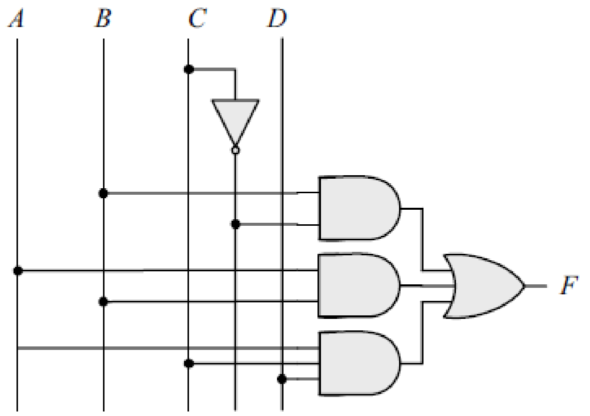
(b) $F(A, B, C, D) = \prod(3, 5, 8, 11)$
$F(A, B, C, D) = \prod(3, 5, 8, 11) = \sum (0,1,2,4,6,7,9,10,12,13,14,15)$

[2.24] Convert each of the following expressions into sum of products and product of sums :

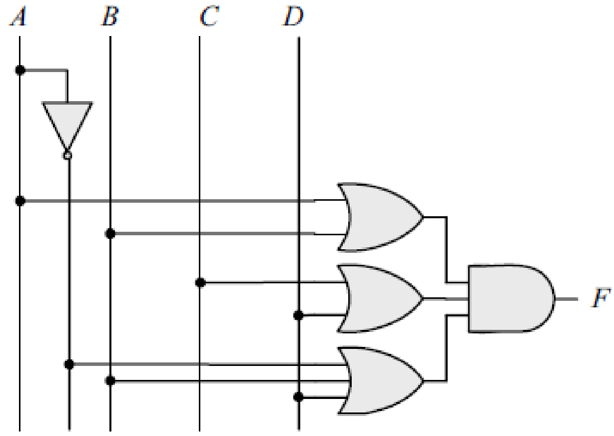
(a) (*u* + *xw*)(*x* + *u'v*)
= ux + uu'v + xxw + xwu'v
= ux + xw + xwu'v

(b) *x'*+ *x*(*x* + *y'*)(*y* + *z'*)
= *(x' + x)[x' + (x + y')(y + z')]*
= *(x' + x + y')(x' + y + z')*
= *x'* + *y* + *z'*

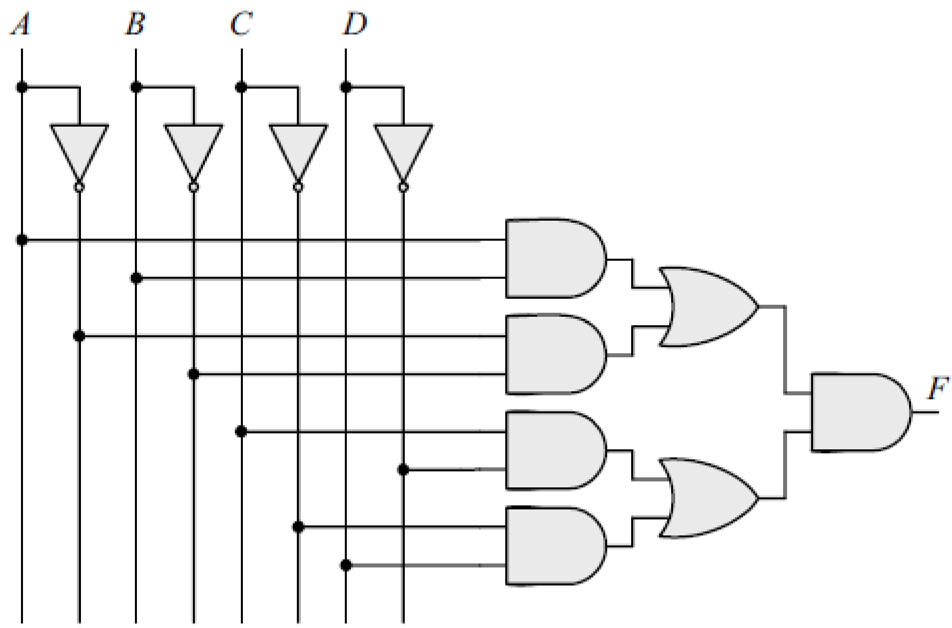
[2.25] Draw the logic diagram corresponding to the following Boolean expressions without simplifying
them :
(a) *BC'* + *AB* + *ACD*



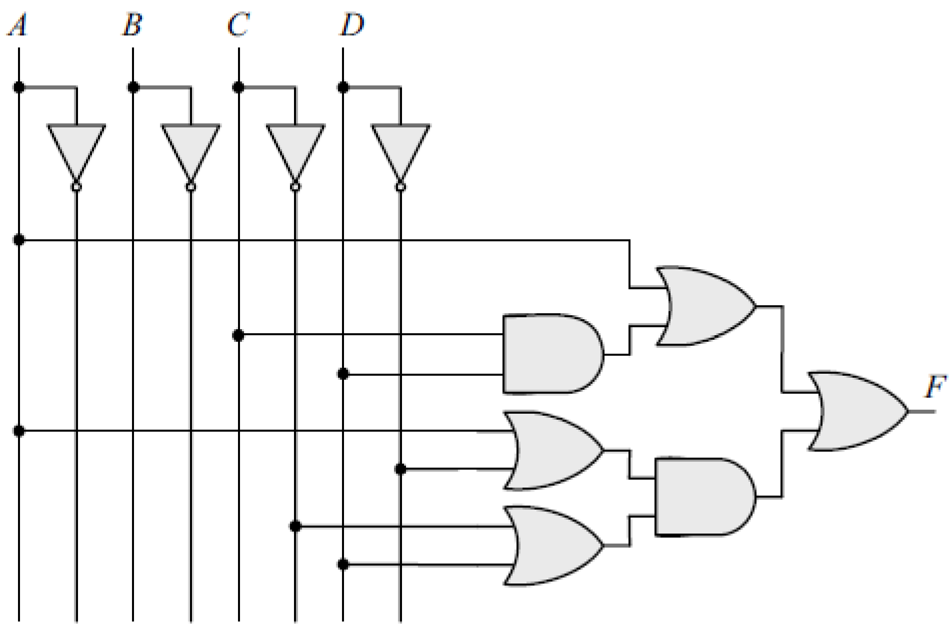
(b) (*A* + *B*)(*C* + *D*)(*A'* + *B* + *D*)



(c) (*AB* + *A'B'*)(*CD'* + *C'D*)



(d) *A + CD + (A + D')(C' + D)*



[2.26] Write the following Boolean expressions in sum of products form :
(b + d)(a' + b' + c)
a'b + b'b + bc + a'd + db' + dc
a'b + 0 + bc + a'd +db' + dc
a'b + bc + a'd + db' + dc

[2.27] Write the following Boolean expression in product of sums form:
a'b + a'c' + abc
=∑(0,2,3,7)
=∏(1,4,5,6)
(a+b+c') (a'+b+c) (a'+b+c') (a'+b'+c)