DiS Notes (divB):- ODD 2020-2021

(Author: Prof Ravindra Divekar)

Here is the list of rules used for the boolean expression simplifications. This is a fairly standard list you could find most anywhere, but we thought you needed an extra copy.

The Idempotent Laws	$AA = A$		$A+A = A$	
The Associative Laws	$(AB)C = A(BC)$		$(A+B)+C = A+(B+C)$	
The Commutative Laws	$AB = BA$		$A+B = B+A$	
The Distributive Laws	$A(B+C) = AB+AC$		$A+BC = (A+B)(A+C)$	
The Identity Laws	$AF = F$	$AT = A$	$A+F = A$	$A+T = T$
The Complement Laws	$A\overline{A} = F$	$A+\overline{A} = T$	$\overline{F} = T$	$\overline{T} = F$
The Involution Law	$\overline{\overline{A}} = A$			
DeMorgan's Law	$\overline{AB} = \overline{A}+\overline{B}$		$\overline{A+B} = \overline{A}\;\overline{B}$	

Simplify: C + BC:

Expression	Rule(s) Used
C + BC	Original Expression
C + (B + C)	DeMorgan's Law.
(C + C) + B	Commutative, Associative Laws.
T + B	Complement Law. (T means true or logic1)
T	Identity Law.(T=1)

Simplify: AB(A + B)(B + B):

Expression	Rule(s) Used
AB(A + B)(B + B)	Original Expression
AB(A + B)	Complement law, Identity law.
(A + B)(A + B)	DeMorgan's Law
A + BB	Distributive law. This step uses the fact that “or” distributes over “and”. Itcan look a bit strange since addition does not distribute Over multiplication.
A	Complement, Identity.

Simplify: (A + C)(AD + AD) + AC + C:

Expression	Rule(s) Used
(A + C)(AD + AD) + AC + C	Original Expression
(A + C)A(D + D) + AC + C	Distributive.
(A + C)A + AC + C	Complement, Identity.
A((A + C) + C) + C	Commutative, Distributive.
A(A + C) + C	Associative, Idempotent.
AA + AC + C	Distributive.
A + (A + *T*)C	Idempotent, Identity, Distributive.
A + C	Identity, twice.

UNIVERSAL GATES: NAND, NOR

Every other gate can be expressed or Symbolized as a combination of nand gates or nor gates.

NOR as a universal gate:-

NOR: Y = a' + b'   OR = ~NOR = (a' + b') '

NOT: (a + a)'

AND: Y = a.b
= ((a.b)')'
= (a' + b') '
= ((a+a)' + (b+b)')'
NAND: = (((a+a)' + (b+b)')')'

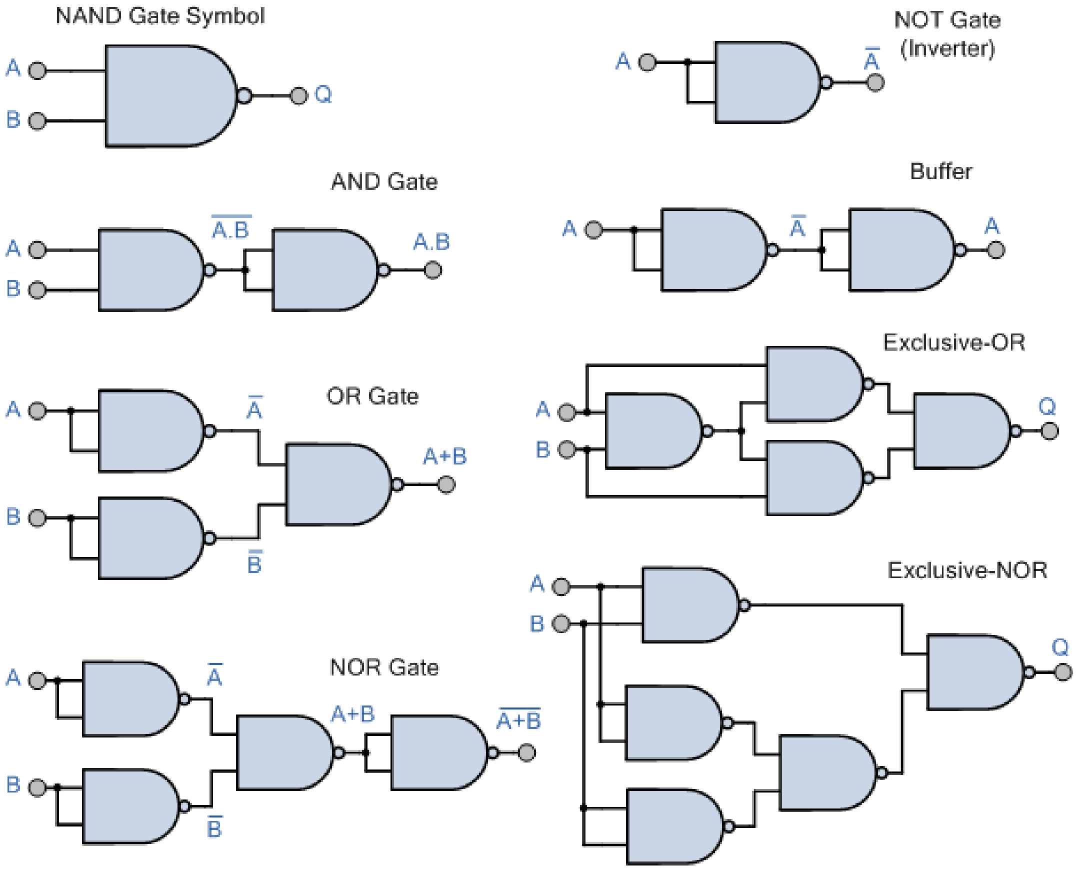
EXOR: Y = a xor b
= ab' + a'b
((ab' + a'b)')'
= ((ab')'. (a'b)')'   :::: (x'+y') = x'.y'
= ((a' + b). (a+b'))'   ::::: (xy)' = x' + y'
= let m = (a'+b) n = (a+b')
(m.n)' = m' + n'
= ((a'+b)' + (a+b')')' ← all NOR gates?
= (((a+a)' +b)' + (a + (b+b)')')'

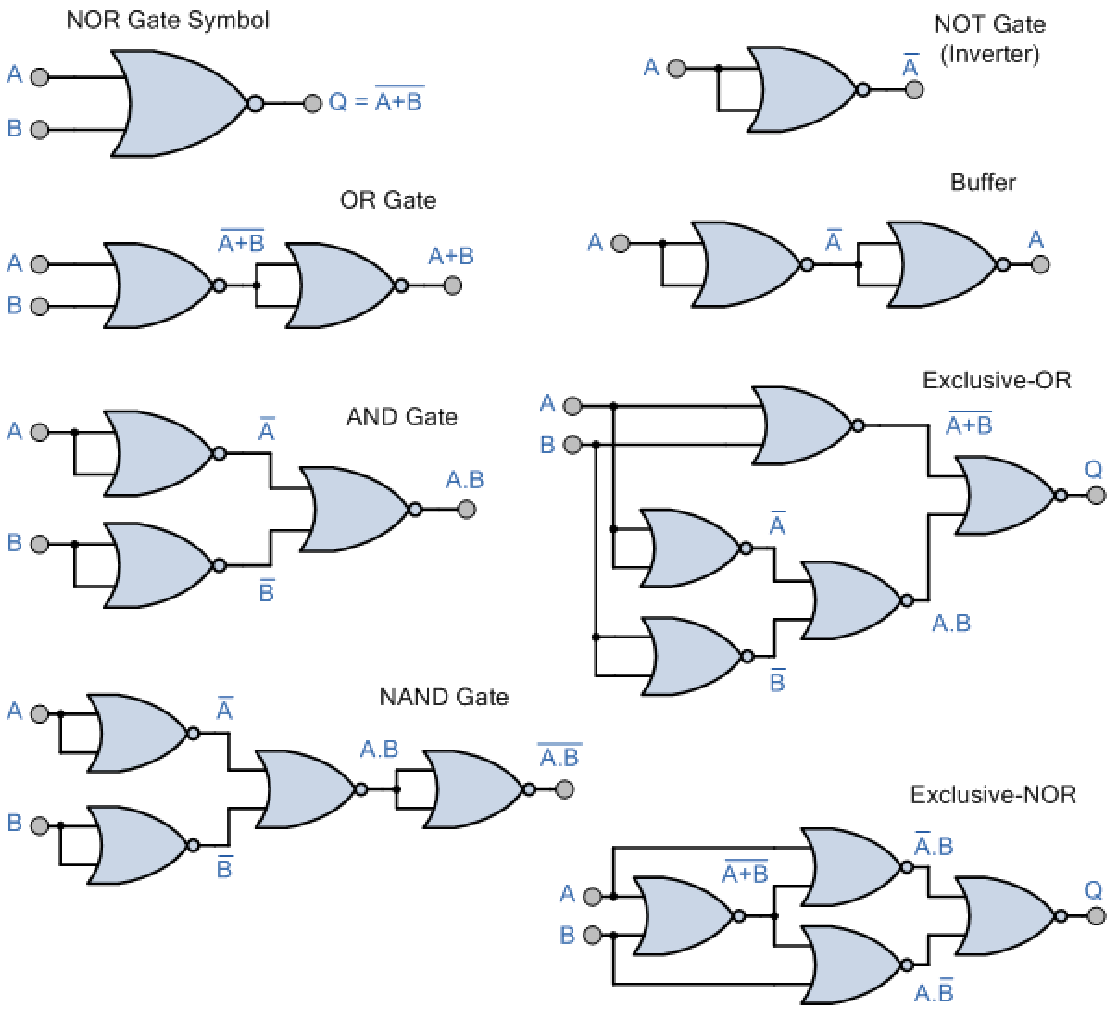
EX-NOR: (ab' + a'b)' = ab + a'b'

NAND as a universal gate:-

NAND: Y = (a.b)'
AND: ((ab)')' = ((ab)'.(ab)')'
NOT: (a.a)'
OR: a+b = ((a+b)')' = (a'.b')' = ((a'b').(a'b'))'
NOR: ((ab)'.(ab)')'
EXOR: (((a'b)'(ab')')'
EX_NOR: do it yourself

Logic Gates using only NAND Gates




NOR Gate Symbol
A
B
Q = $\overline{A+B}$
NOT Gate (Inverter)
A
$\overline{A}$
OR Gate
A
B
$\overline{A+B}$
A+B
Buffer
A
$\overline{A}$
A
AND Gate
A
B
$\overline{A}$
$\overline{B}$
A.B
Exclusive-OR
A
B
$\overline{A+B}$
$\overline{A}$
$\overline{B}$
A.B
Q
NAND Gate
A
B
$\overline{A}$
$\overline{B}$
A.B
$\overline{A.B}$
Exclusive-NOR
A
B
$\overline{A+B}$
$\overline{A}$.B
A.$\overline{B}$
Q

SOP = sum of products

Y = a’b + bc each term (a’b , bc) is called a **Minterm**

POS = product of sums

Y = (x’+z) . (y+z’) each term (x’+z) and (y+z’) is a **Maxterm**

Canonical or Standard Form

Y(a,b,c) = ab + a'bc not in canonical form

ab(c+c') + a'bc

abc + abc' + a'bc ← **is** a canonical SOP form

ex2: Y(a,b,c,d) = ab + c'd'

= ab + cd'
= ab(c+c')(d+d') + cd'(a+a')(b+b')
= ab(cd+c'd+cd'+c'd') + cd'(ab+a'b+ab'+a'b')
= abcd + abc'd + abcd'+ abc'd' + abcd' + a'bcd' + ab'cd' + a'b'cd' (term in red occurs more than once so only one copy is kept).
= [abcd + abc'd + abcd'+ abc'd' + a'bcd' + ab'cd' + a'b'cd']

(is canonical SOP or standard SOP form of the given expression: ab + cd')

Y(p,q,r) = (p+r') . (p+q+r)
= (p + qq' + r'). (p+q+r)
= (p+q+r') . (p+q'+r') . (p+q+r)

Is canonical POS form of the given expression.

P1) X(a,b,c) = a'+b'
= a'(b+b')(c+c') + b'(a+a')(c+c')
= a'bc + a'b'c + a'bc' + a'b'c' +
ab'c + a'b'c + ab'c' + a'b'c'

P2) Y(a,b,c)
= (a'+b')c
= (a'+b' + cc')(c + aa' + bb')
= (a'+b'+c)(a'+b'+c')(a+b+c)(a'+b+c)(a+b'+c)(a'+b'+c')
= (a'+b'+c)(a'+b'+c')(a+b+c)(a'+b+c)(a+b'+c)

Relation of Canonical Form or Standard form to Minterms or Maxterms:

EX1:

Y(a,b,c) = ab + a'bc ;not canonical form

= ab(c+c') + a'bc

= abc + abc' + a'bc 🡨 is a canonical SOP form

= m7 + m6 + m3

= $\sum$m(3,6,7)

Here Y(a,b,c) = Sum of products

= contains minterms

= each product is a minterm

a b c
0 0 0
0 0 1
0 1 0
0 1 1 = a'bc = m3
1 0 0
1 0 1
1 1 0 abc' = m6
1 1 1 abc = m7

P = (a'+b'+c).(a'+b'+c').(a+b+c).(a'+b+c).(a+b'+c) = Product of Sums

a b c
0 0 0 = (a+b+c) = M0
0 0 1
0 1 0 =.(a+b'+c) = M2
0 1 1
1 0 0 = (a'+b+c) = M4
1 0 1
1 1 0 = (a'+b'+c) = M6
1 1 1 =(a'+b'+c') = M7

P = M0. M2. M4. M6. M7 = πM(0,2,4,6,7)