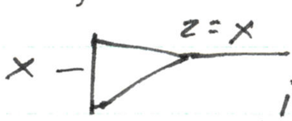
CSE241 Feb 13, 2017

Logic gates 1. name 2. graphic symbol 3. algebraic symbol 4. logic expression 5. Truth Table

I Inverter NOT ' x' $\bar{x}$

input	output
x	z
0	1
1	0

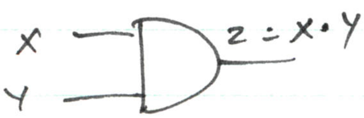
II Buffer



input	output
x	z
0	0
1	1

delay ; increases fan-out capability "boost"

III

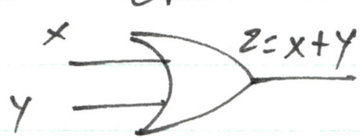


AND

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

$z = x \cdot y = xy$

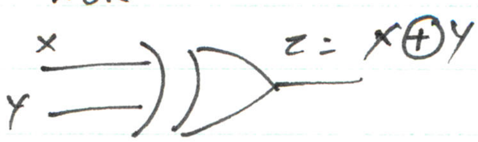
IV OR



input

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

V XOR

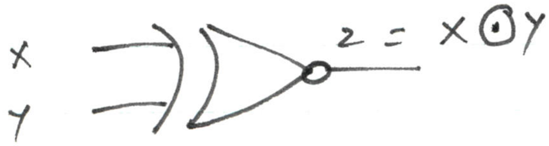


x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

$z = x \oplus y$
$= x' \cdot y + x \cdot y'$

Feb 13, 2017 (2)

VI XNOR

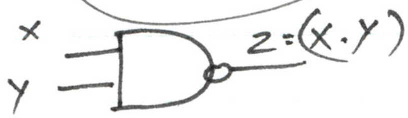


input		output
x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

$z = x \odot y$

$= x' \cdot y' + x \cdot y$

VII NAND



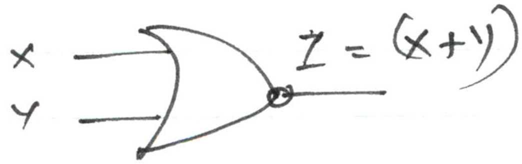
x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

$z = (x \cdot y)'$

$= x' + y'$

DeMorgan's law

VIII NOR



input		output
x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

$z = (x+y)'$

$= x' \cdot y'$

Problem $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$

Given this function, simplify to minimum # of terms + literals. Use algebraic simplification

$$\underset{1}{A'B'C} + \underset{4}{AB'C'} + \underset{5}{AB'C} + \underset{6}{ABC'} + \underset{7}{ABC}$$

001, 100, 101, 110, 111

$= A'B'C + AB'(C' + C) + AB(C' + C)$ ($C'+C = 1$)

$= A'B'C + AB' + AB = A'B'C + A(B' + B)$ ($B'+B = 1$)

$= A' \cdot B'C + A = (A' + A) \cdot (B'C + A)$ ($A'+A = 1$)

$= \underline{B'C + A}$

Assess: Simplification

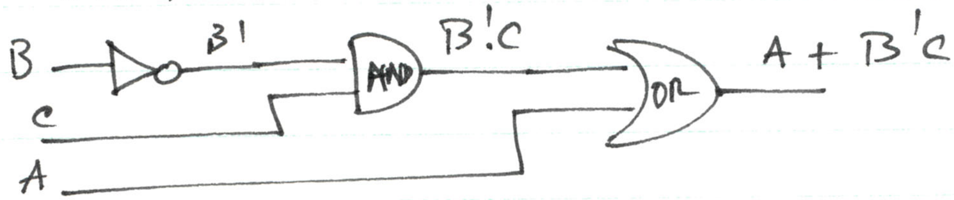
5 → 2 terms

15 literal → 3 literal

Feb 13, 2017 (3)

Implement the simplified expression using (i) {NOT, AND, OR} (ii) only NAND

$B'C + A.1$



Three diff. types of gates:

inventory	NOT	hex inverter	74LS06	NOT or inverter IC
	AND	quad AND	74LS08	AND IC
	OR	quad OR	74LS32	OR IC

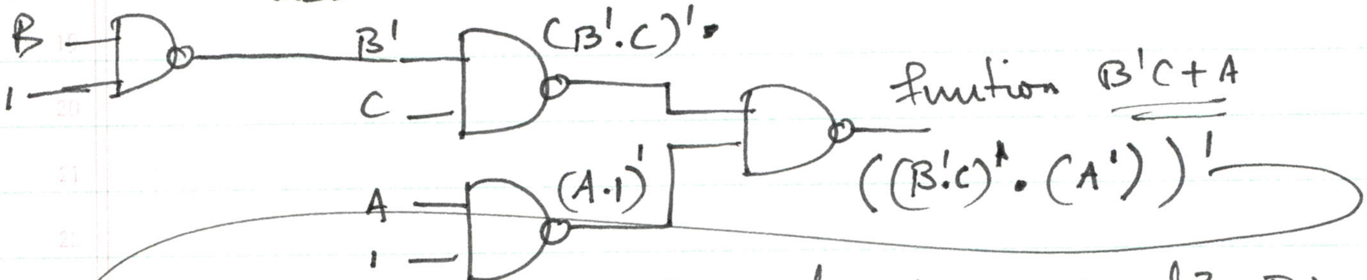
NAND implementation.

Implement each term with NAND gate

Sum up the terms (output of NAND) using another NAND gate.

$f(A, B, C) = B'C + A.1$

assume B' is available.



What IC chip do you need? I just need only 1 kind. NAND IC

quad NAND 74LS00

$((B'C)' \cdot A')' = (B'C) + A = B'C + A$

DeMorgan's law; Proceed equivalen

Feb 13, 2014

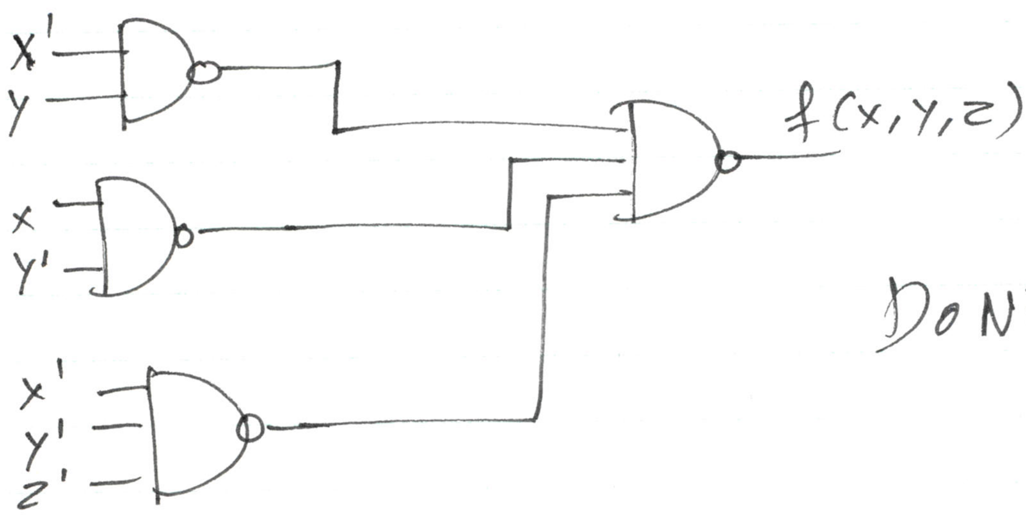
If the final simplified function is in sum of products form: how do you implement it using only NAND gates?

Step 1: Implement each term using a NAND

Step 2: Sum all the NAND outputs in Step1 using another NAND gate.

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

Assume: Prime of variables are available
3-input NAND gate is available



DONE