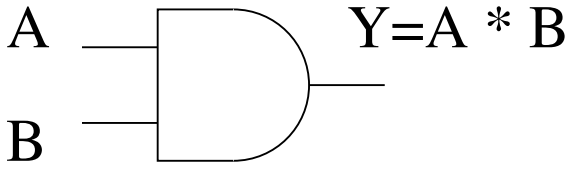
ECE 238L
Digital Computers and Number Systems
Lecture lab 1

August 25, 2006

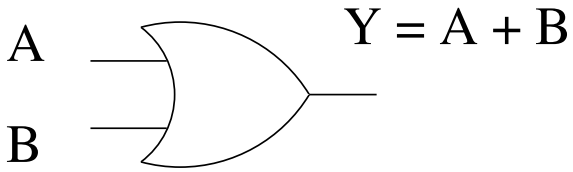
Basic Logic Gates

And Gate



A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

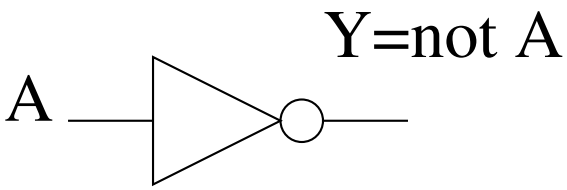
Or Gate



A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

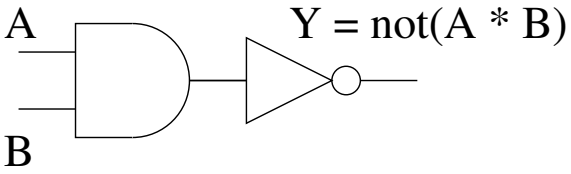
Basic Logic Gates

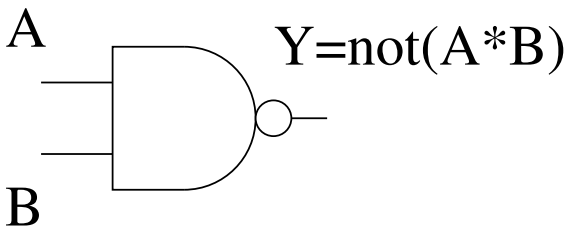
Inverter



A	Y = not A
0	1
1	0

Nand Gate



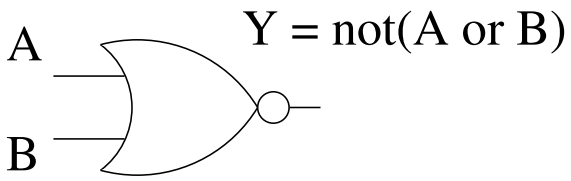


Elementary Theorem - Identity

$X \cdot 1 = X; X \cdot 0 = 0$ $\quad$ $X + 1 = 1; X + 0 = X$

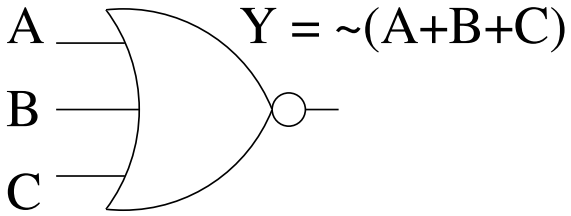
Basic Logic Gates

Nor Gate



A	B	A + B	Y = not(A + B)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

How would you create a two-input nor from a three input nor?



Not(A+B+C) = Not(A + B)

C = ?

Example 1

Design a circuit for the following function:

$$F3 = (Y + Z')X + X'YZ$$

X	Y	Z	X'	Z'	Y + Z'	(Y+Z')X	X'YZ	F3
0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	1	1	0	0	0
0	1	1	1	0	1	0	1	1
1	0	0	0	1	1	1	0	1
1	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	1
1	1	1	0	0	1	1	0	1

$$F3 = XY + XZ' + X'YZ$$

FE equals 1 when XY or XZ' or X'YZ equals 1

K-maps

F3 = (Y + Z')X + X'YZ

X	Y	Z	X'	Z'	Y + Z'	(Y+Z')X	X'YZ	F3
0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	1	1	0	0	0
0	1	1	1	0	1	0	1	1
1	0	0	0	1	1	1	0	1
1	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	1
1	1	1	0	0	1	1	0	1

1- Map TT to K-map

		Y Z			
		00	01	11	10
X	0	0	0	1	0
	1	1	0	1	1

2- Simplify the boolean function

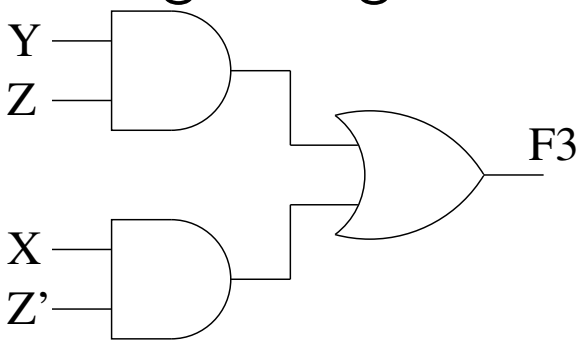
		Y Z			
		00	01	11	10
X	0	0	0	1	0
	1	1	0	1	1

F3 = YZ + XZ'

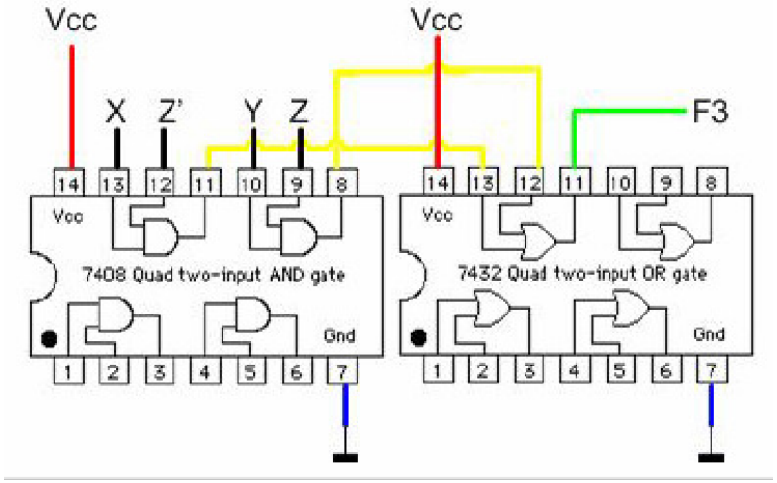
Design an AND/OR circuit

$$F3 = YZ + XZ'$$

Logic Diagram



Layout diagram - position on the breadboard



Example 2

Consider the following 4 variable function:

$$F4 = B'CE' + A'B'E' + AC'E + ABC'E$$

Map to K-map.

Simplify.