MAKE UP CLASS

Tomorrow @ 10am

ER3123

LEC05 & LAB05

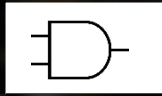
Lec03 & Lab03 Marks

Lec02 Keys

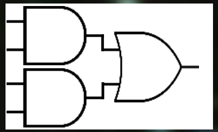
W2022: A Digital Odyssey

Number Systems | $(12)_{10} \rightarrow (1100)_2$

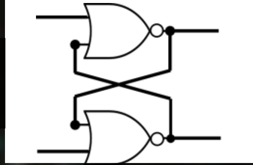
Logic Gates |



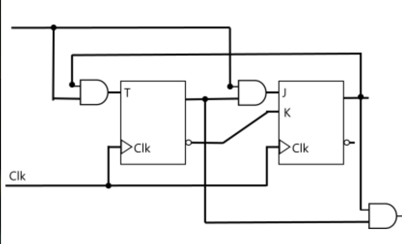
Combinational Logic |

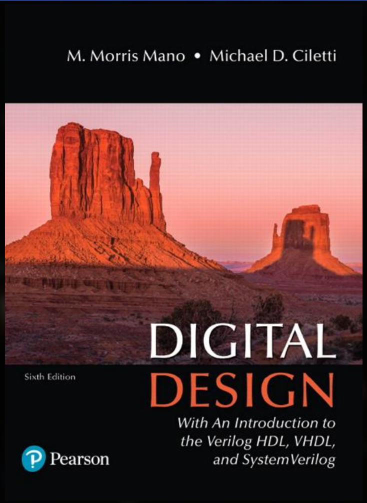


Flip-Flop |



Sequential Logic |





Chapter 2
Boolean Algebra and Logic Gates

BINARY COMPUTER

More Reliability in Engineering
Deep Logic Foundation

2001: A Space Odyssey (1968), Stanley Kubrick, Arthur C. Clarke

Ape Learning Scene: https://www.youtube.com/watch?v=VABNA_an2A0

Bone/Satellite Match Cut: https://www.youtube.com/watch?v=avjdKTqiVvQ

DESIGN COMPUTER

Positive Logic
Button-Up Approach

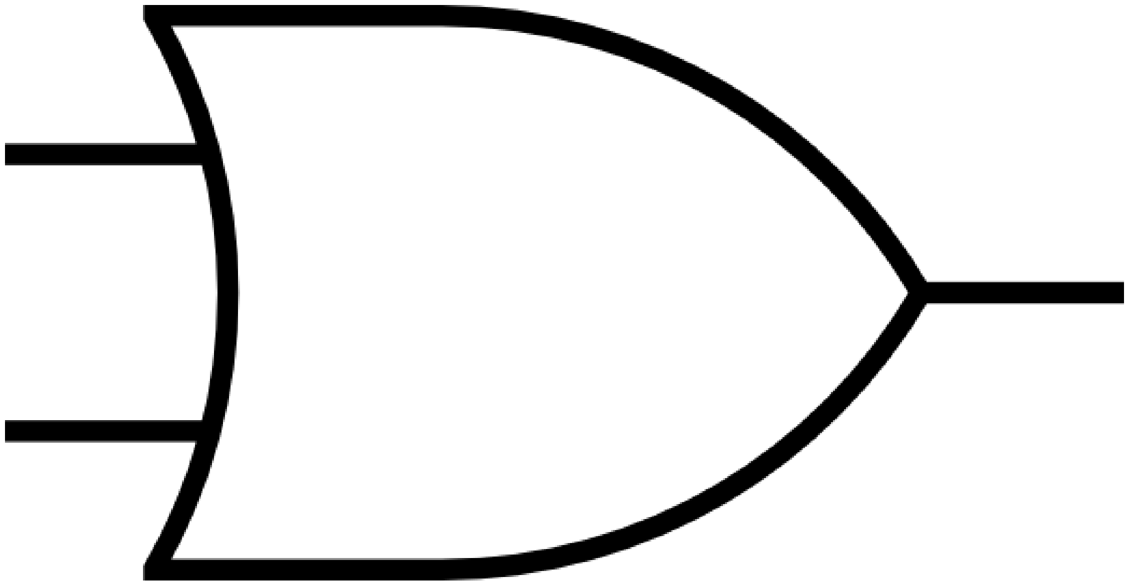
DESIGN COMPUTER

Positive Logic
Button-Up Approach

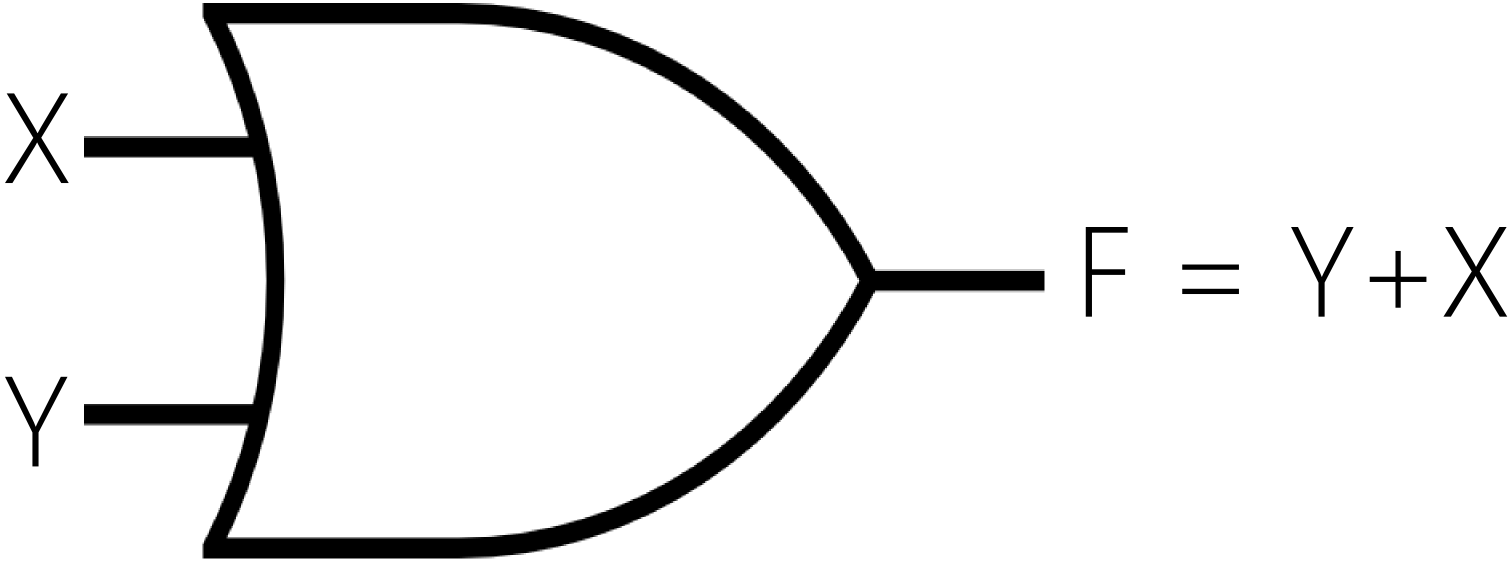
Finding simpler, but equivalent, computers reduces the overall cost!
Rely primarily on mathematical methods in Boolean algebra!

BUILD COMPUTER

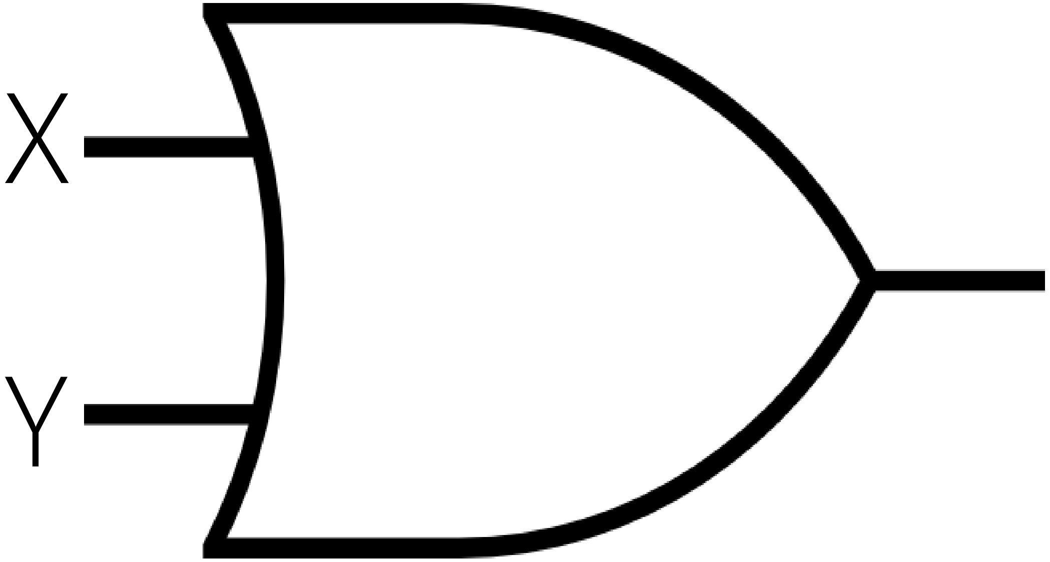
Electrical and Computer Engineering



OR



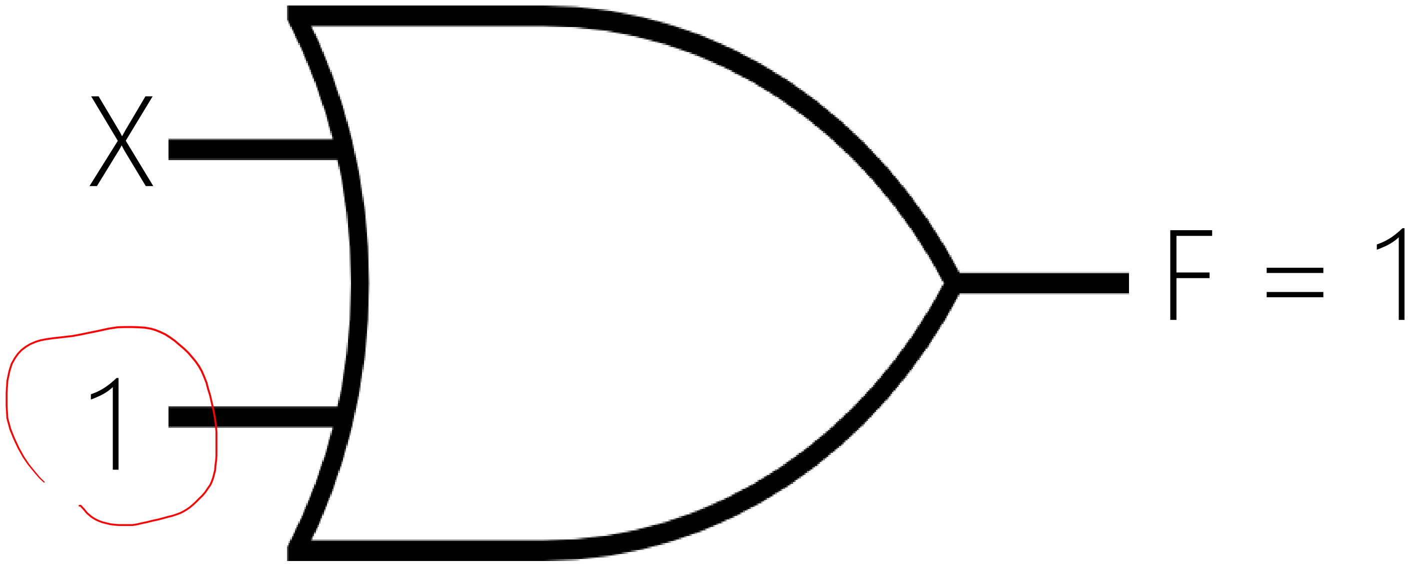
Y	X	Y OR X	Y+X
0	0	0	
0	1	1	
1	0	1	
1	1	1	



$F = Y + X = X + Y$

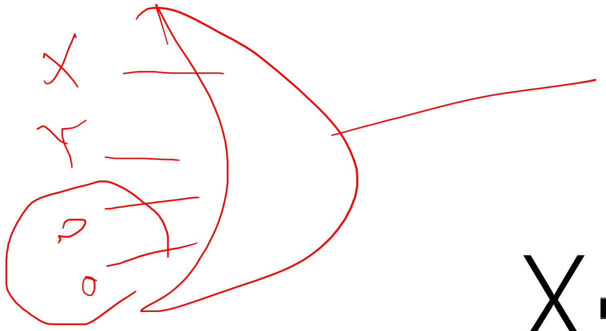
Commutative

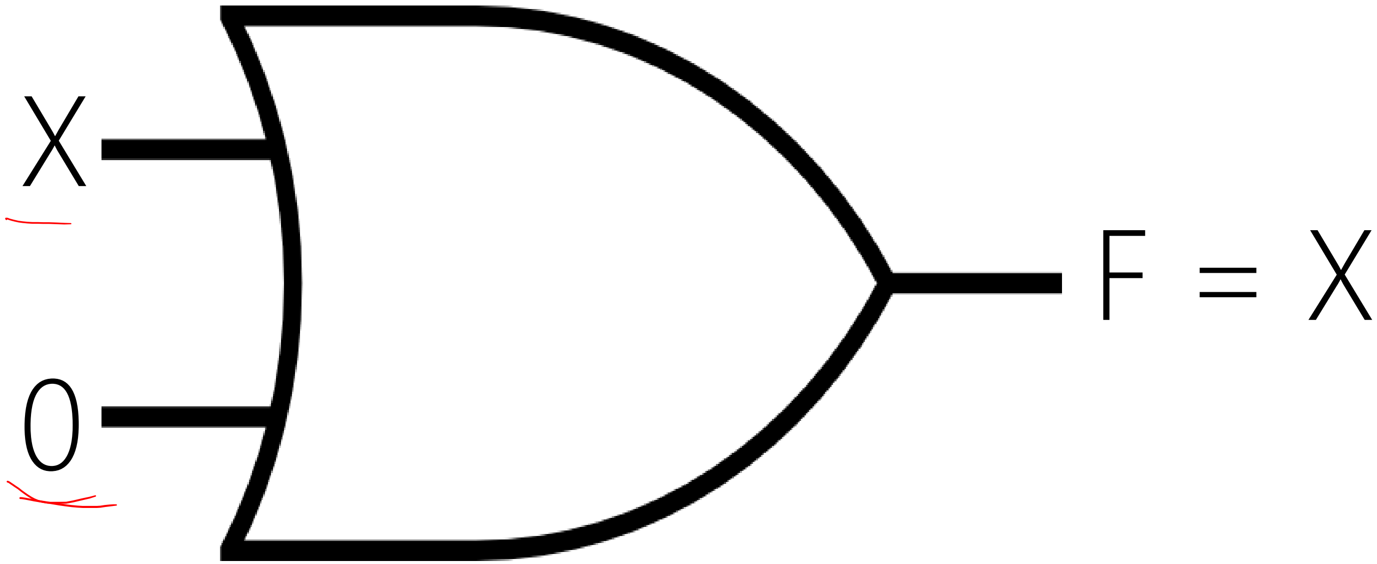
X	Y	X OR Y	X+Y
0	0	0	
0	1	1	
1	0	1	
1	1	1	



Y	X	Y+X
1	0	1
1	1	1

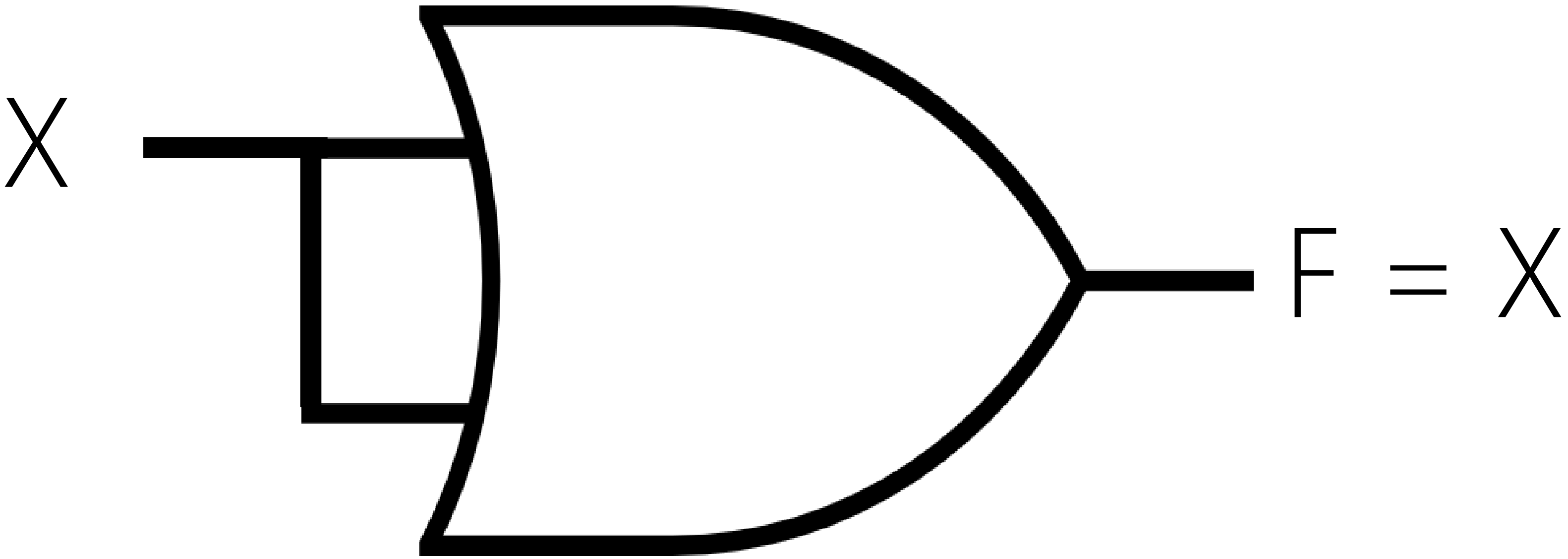
$$F = X+1 = 1$$





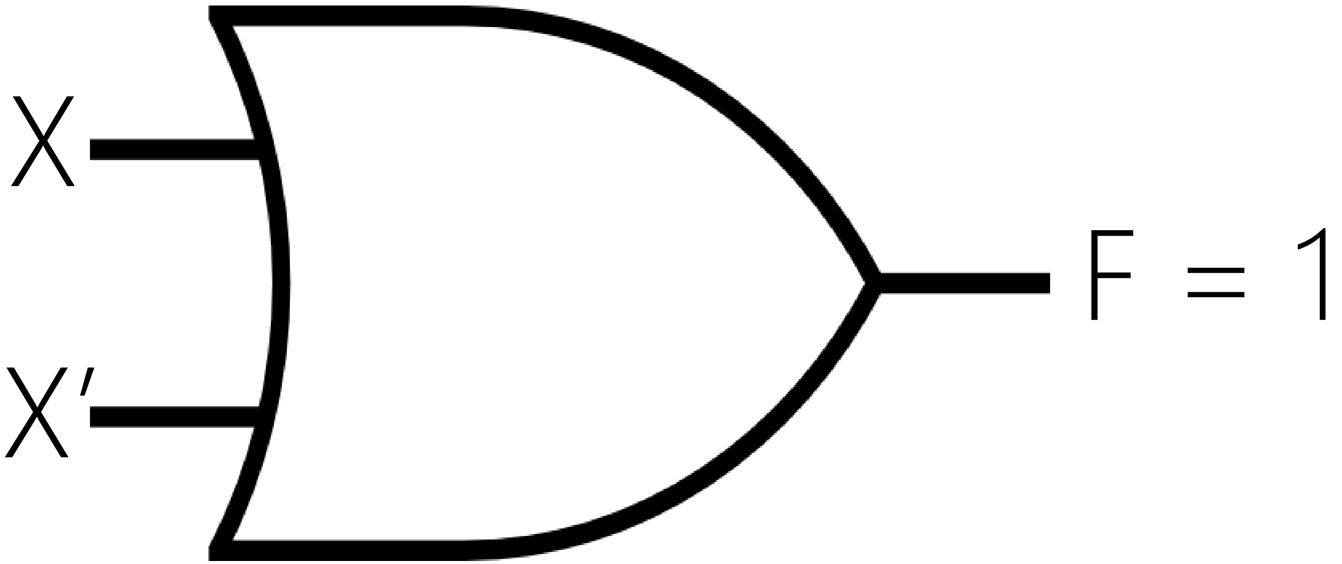
Y	X	Y+X
0	0	0
0	1	1

$$F = X + 0 = X$$



X	X	X+X
0	0	0
1	1	1

$$F = X + X = X$$



$$F = X + X' = 1$$

X′	X	X′+X
1	0	1
0	1	1

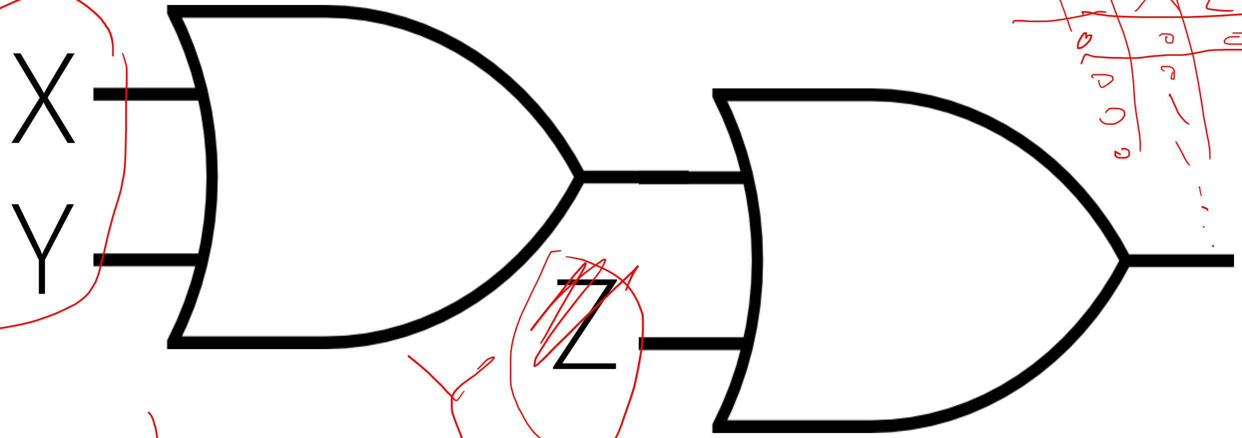
DESIGN

given the functionality, design the structure of a system

3-INPUT OR

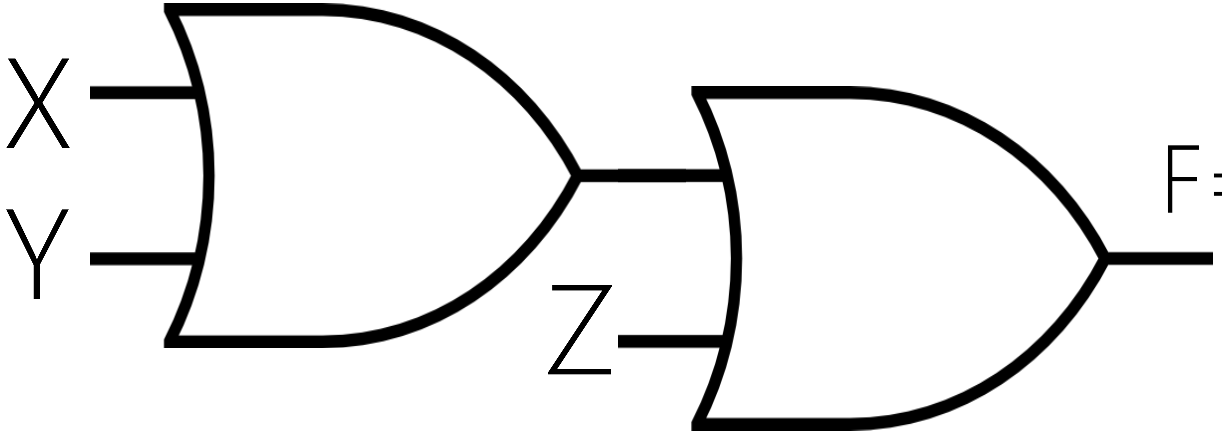
DESIGN PATTERNS

Using Same or Similar Previous Designs for New Designs



$F = Z+(X+Y)$

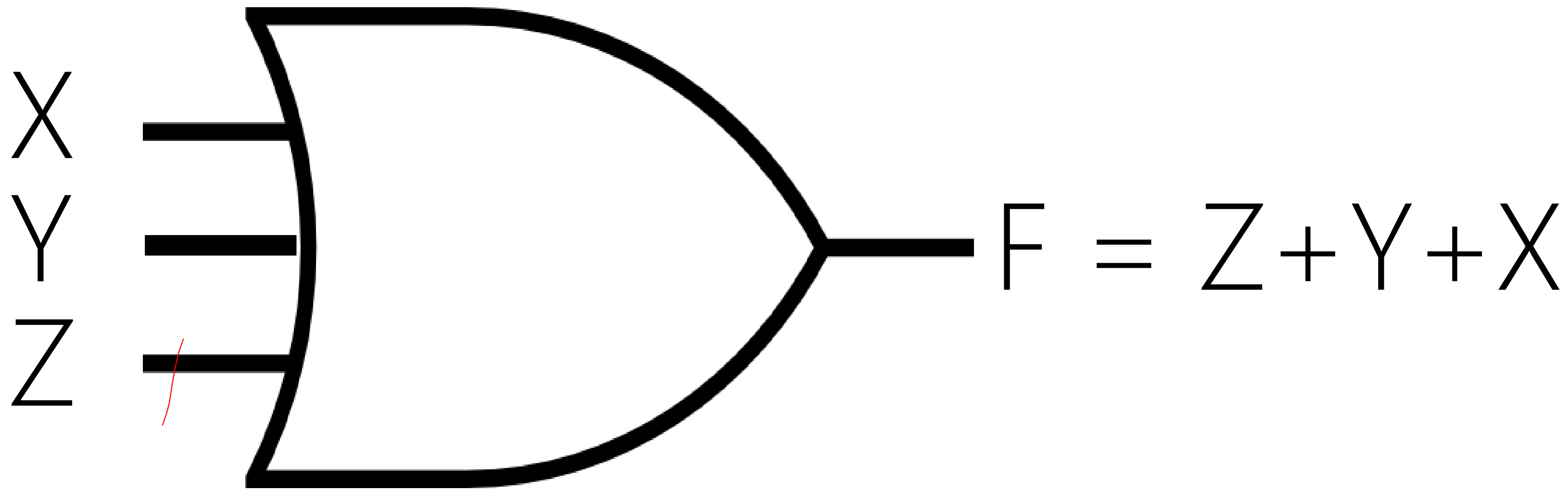
Z	Y	X	Z+(X+Y)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



$$F=Z+(Y+X)=Z+(X+Y)=(Z+X)+Y$$
$$= Z+Y+X$$

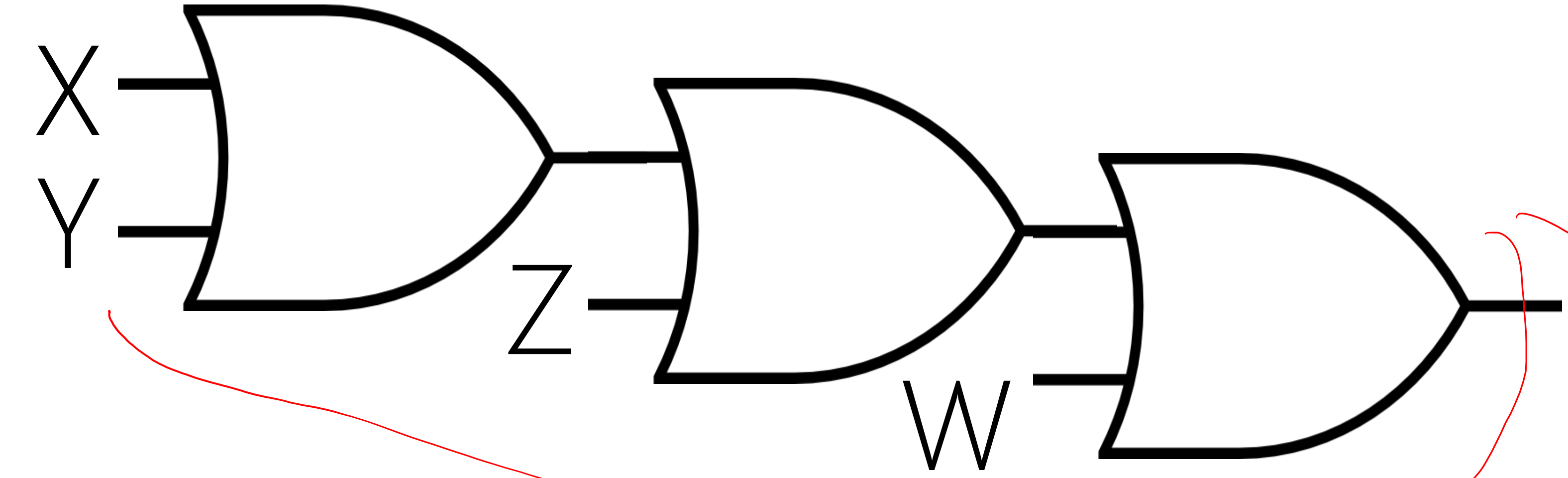
Associative

<table>
<tr><th>Z</th><th>Y</th><th>X</th><th>Z+(Y+X)</th><th>Z+(X+Y)</th><th>(Z+X)+Y</th><th>ZXY</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td colspan="4">0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="4">1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td colspan="4">1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="4">1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="4">1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="4">1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td colspan="4">1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="4">1</td></tr>
</table>

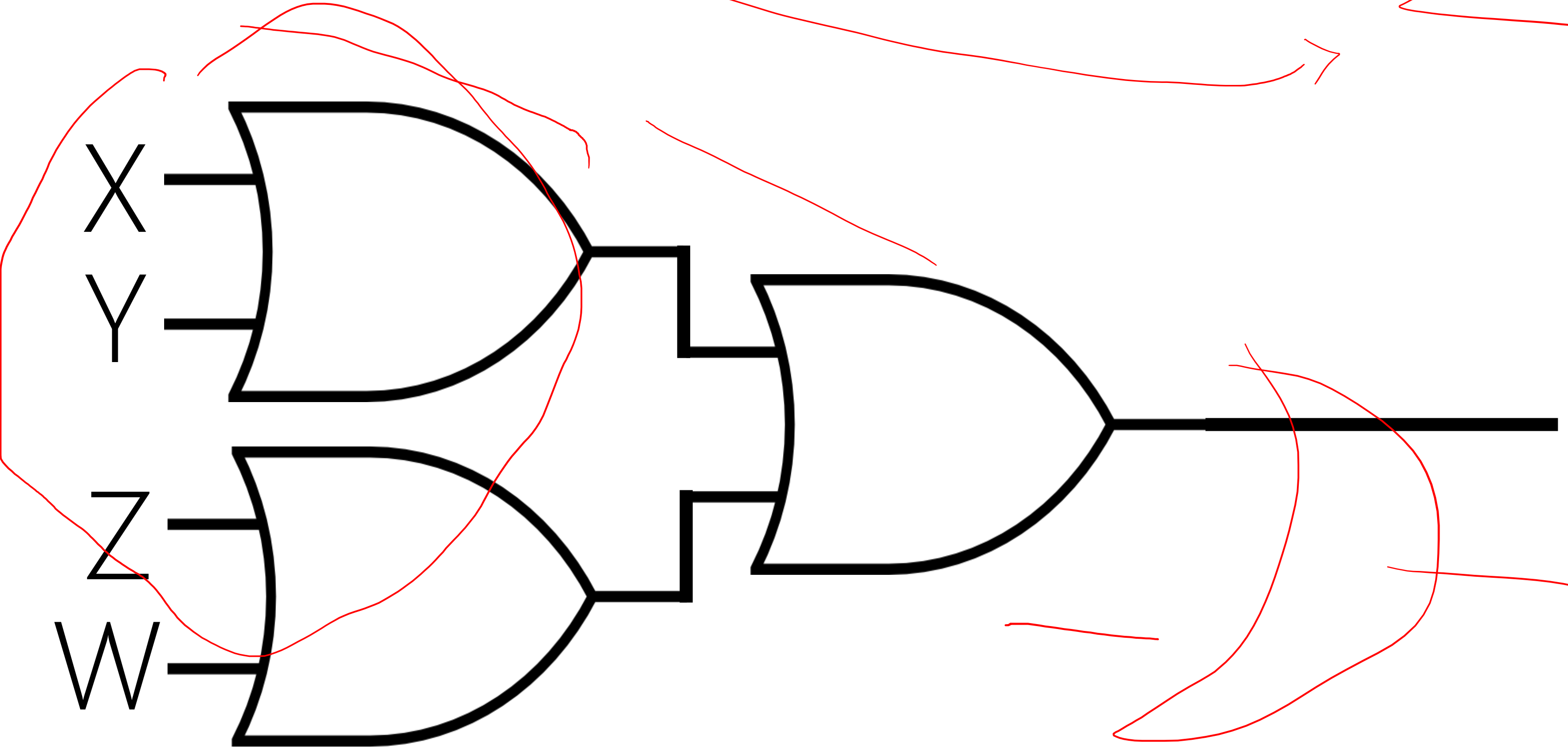


Z	Y	X	Z+(Y+X)	Z+(X+Y)	(Z+X)+Y	ZXY
0	0	0	0			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	1			
1	0	1	1			
1	1	0	1			
1	1	1	1			

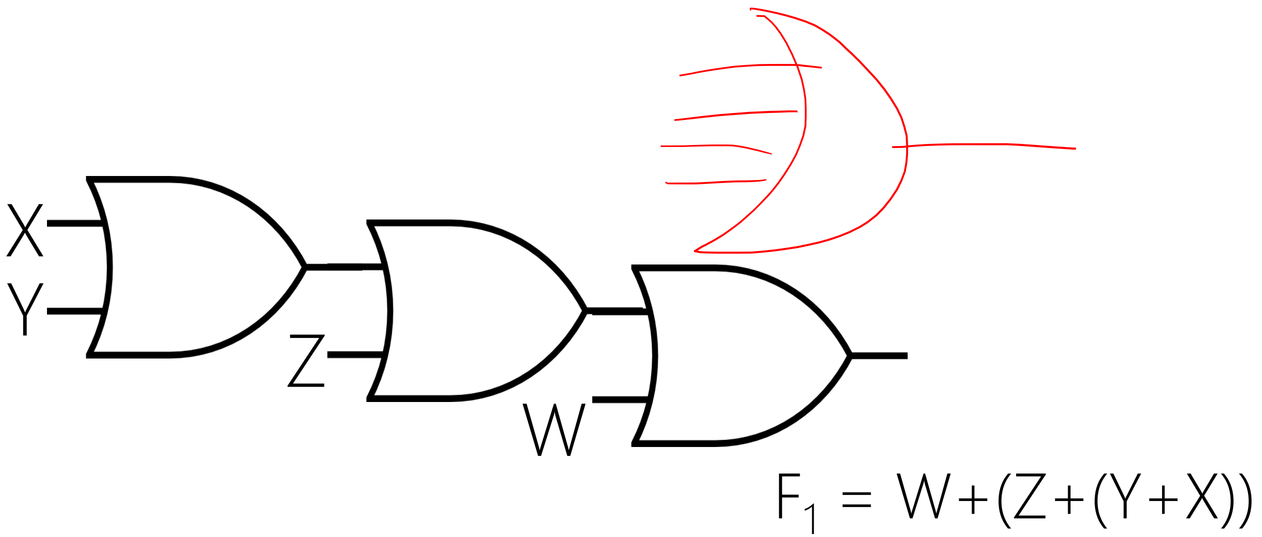
4-INPUT OR

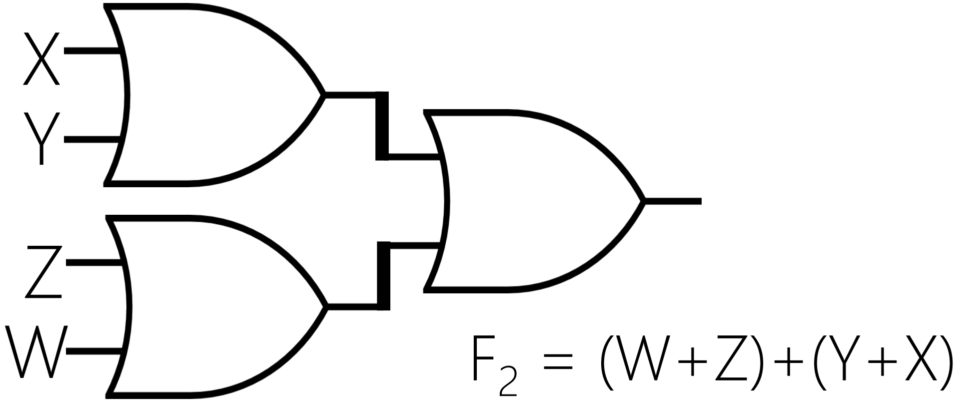


$F_1 = W+(Z+(Y+X))$



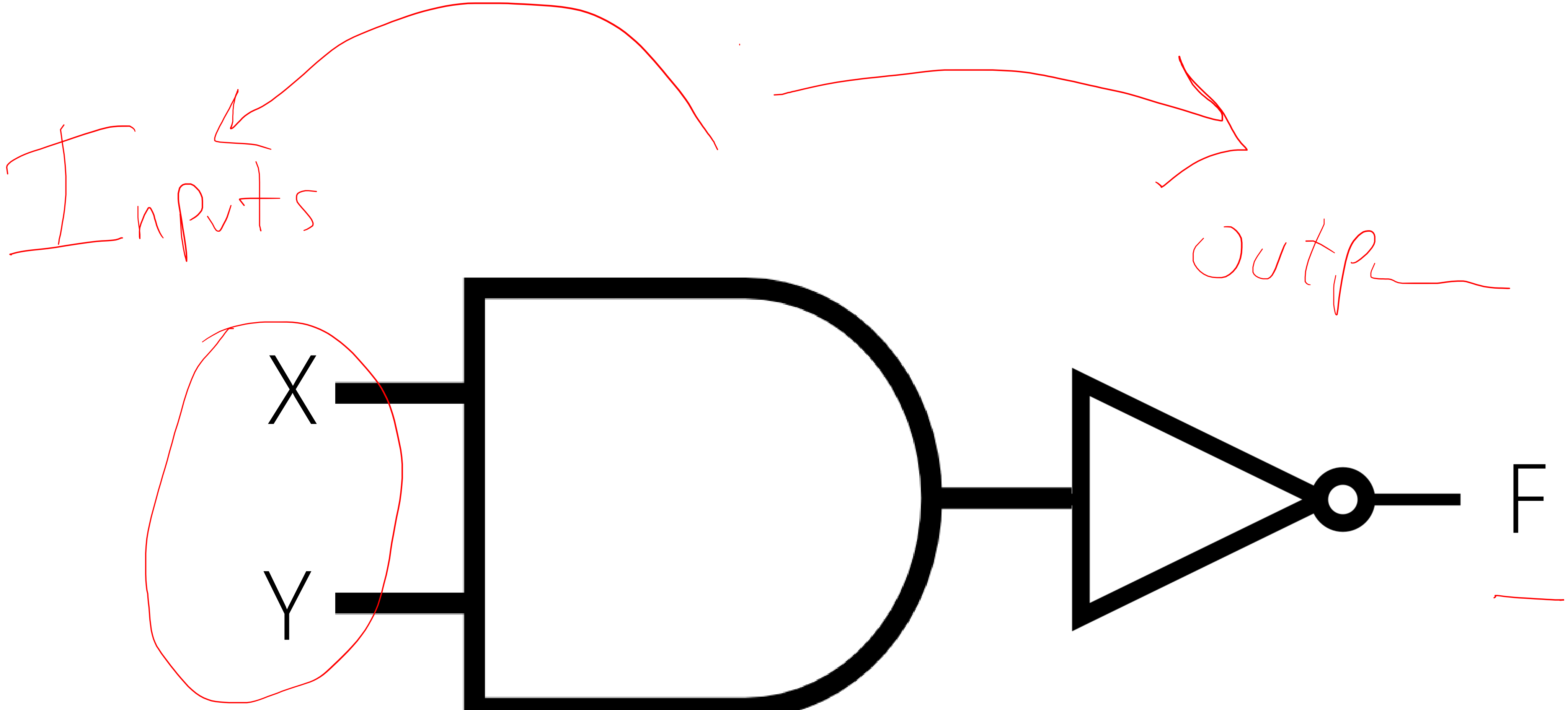
$F_2 = (W+Z)+(Y+X)$

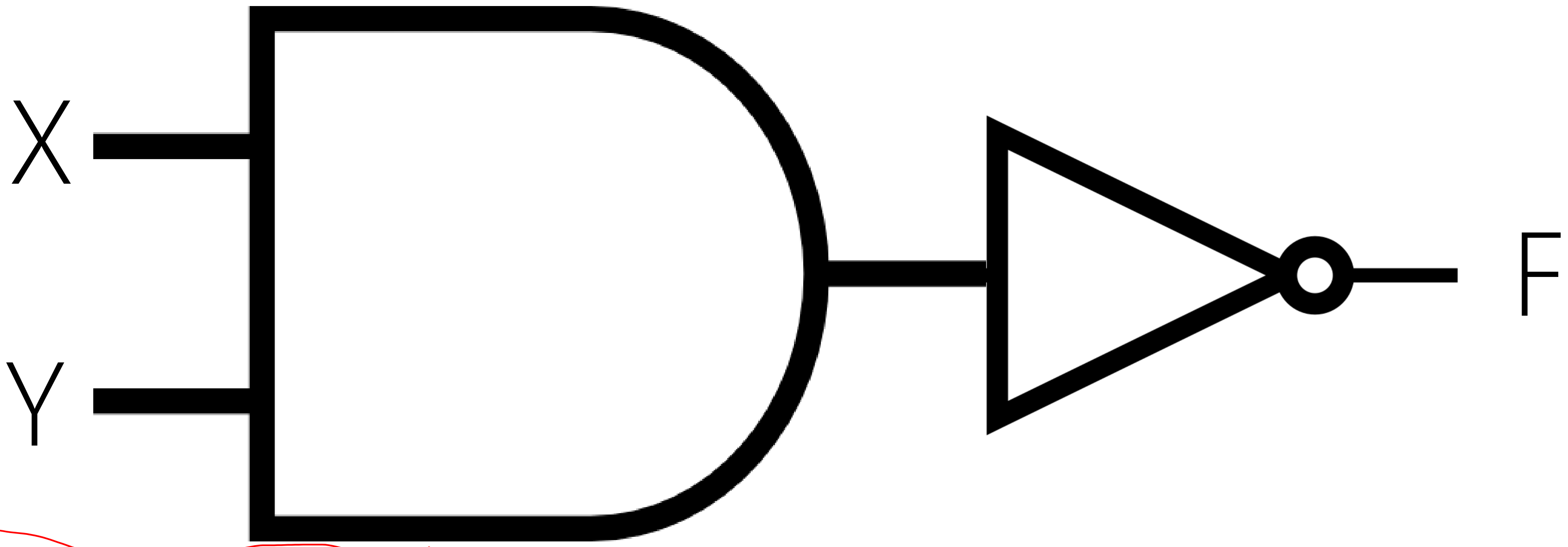




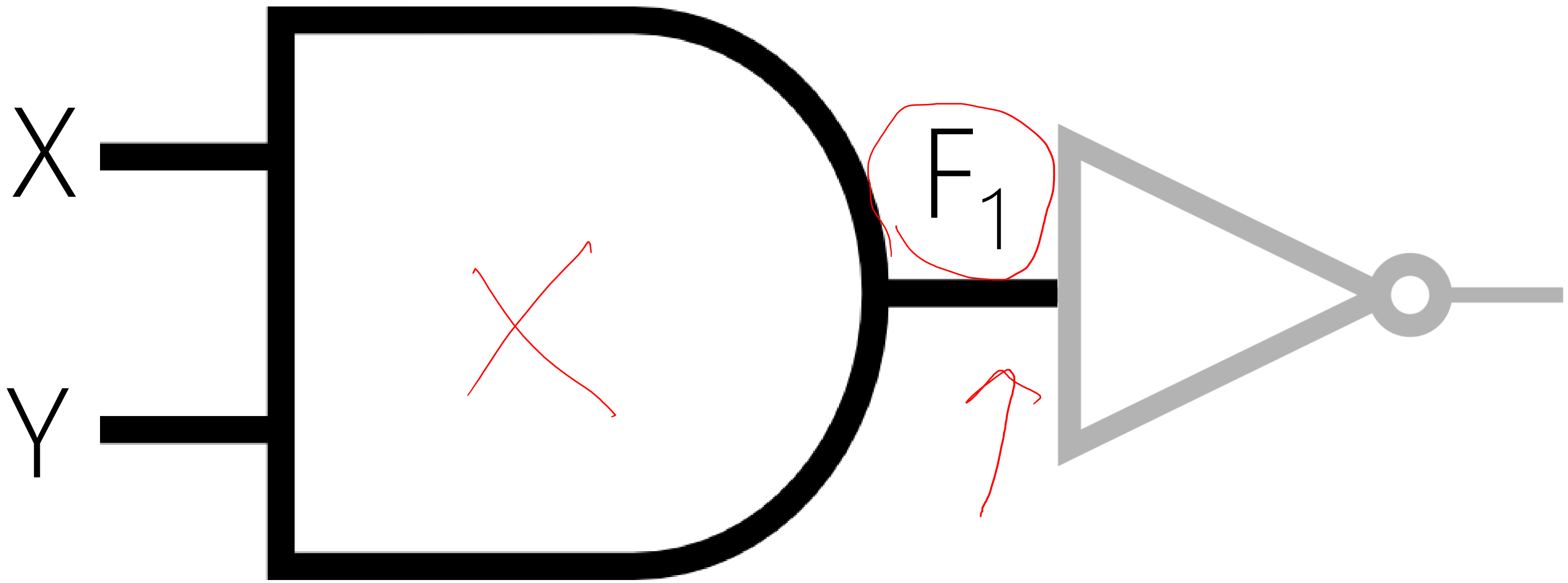
F = W+Z+Y+X	F_1	F_2
Effective (True)	Yes	Yes
Efficient (Fast)	Hmm, 3 levels, No!	Yes! 2 levels
Min. Cost	3 gates, Yes	3 gates, Yes

DESIGN vs. ANALYSIS

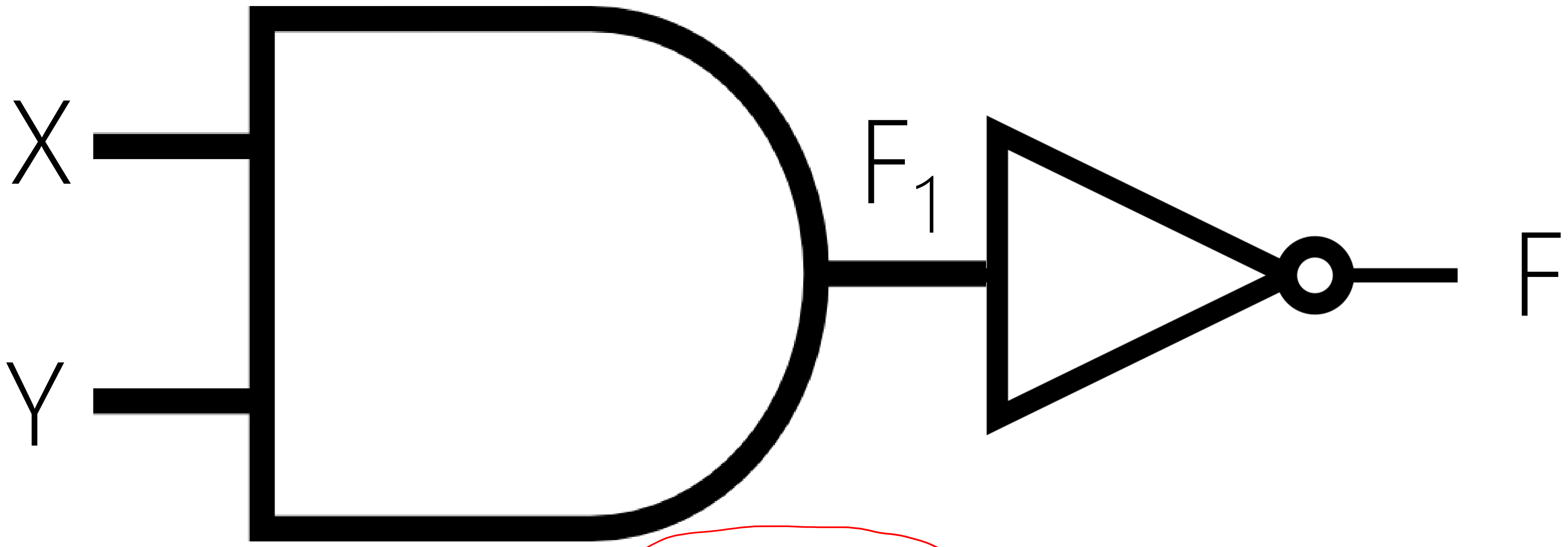

Inputs
Outp
X
Y
F



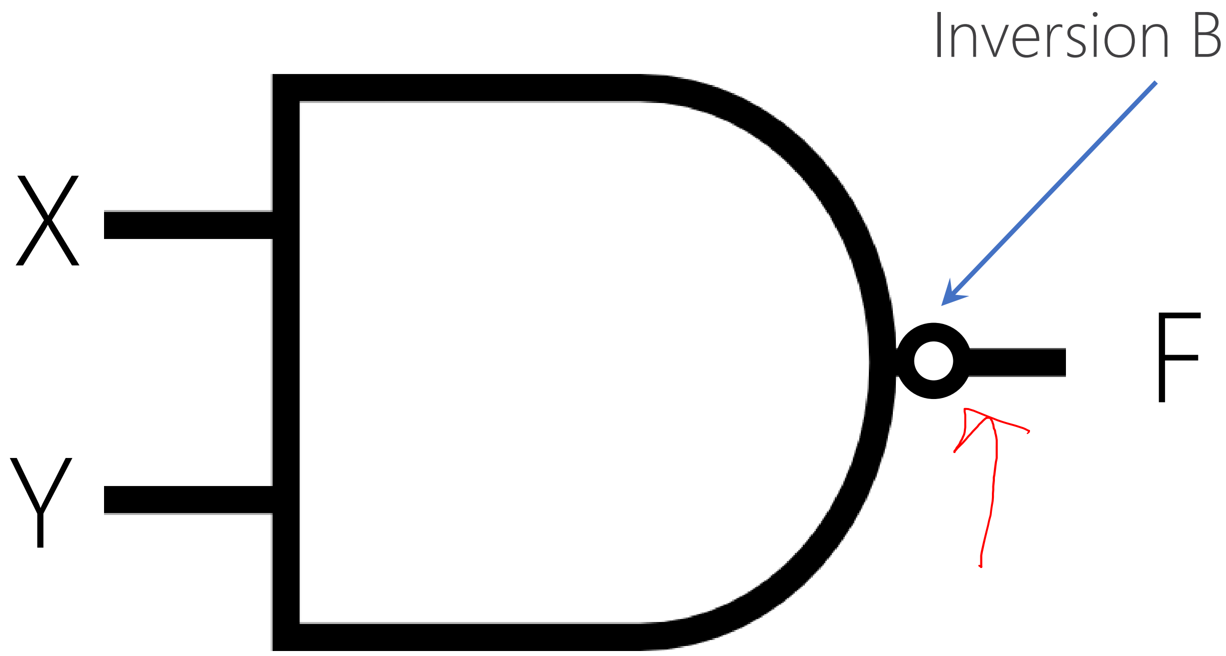
Y	X	F = ?
0	0	?
0	1	?
1	0	?
1	1	?



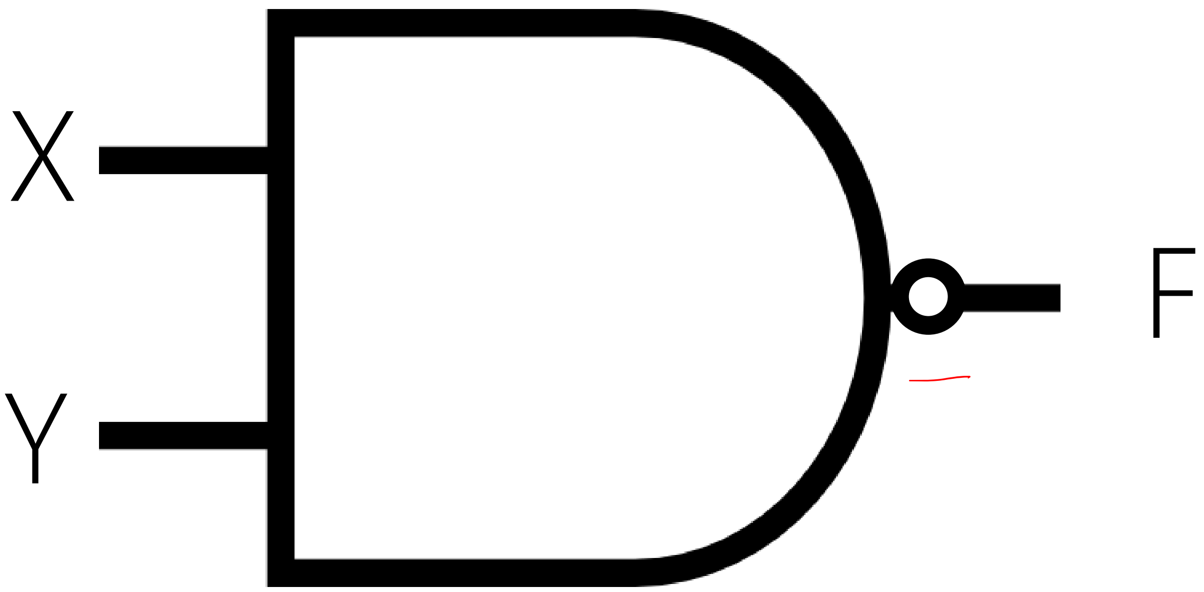
Y	X	$F_1 = YX$
0	0	0
0	1	0
1	0	0
1	1	1



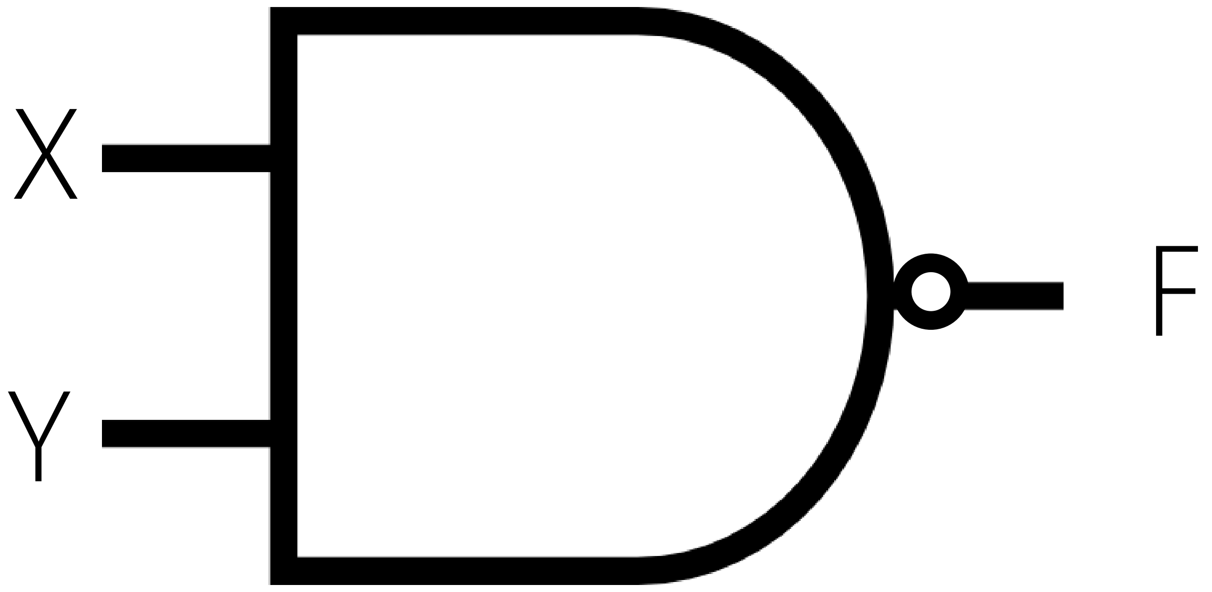
Y	X	$F_1 = YX$	$F = (YX)'$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



NAND (Not – AND)



Y	X	$F = (YX)'$	$F = Y \uparrow X$
0	0	1	
0	1	1	
1	0	1	
1	1	0	



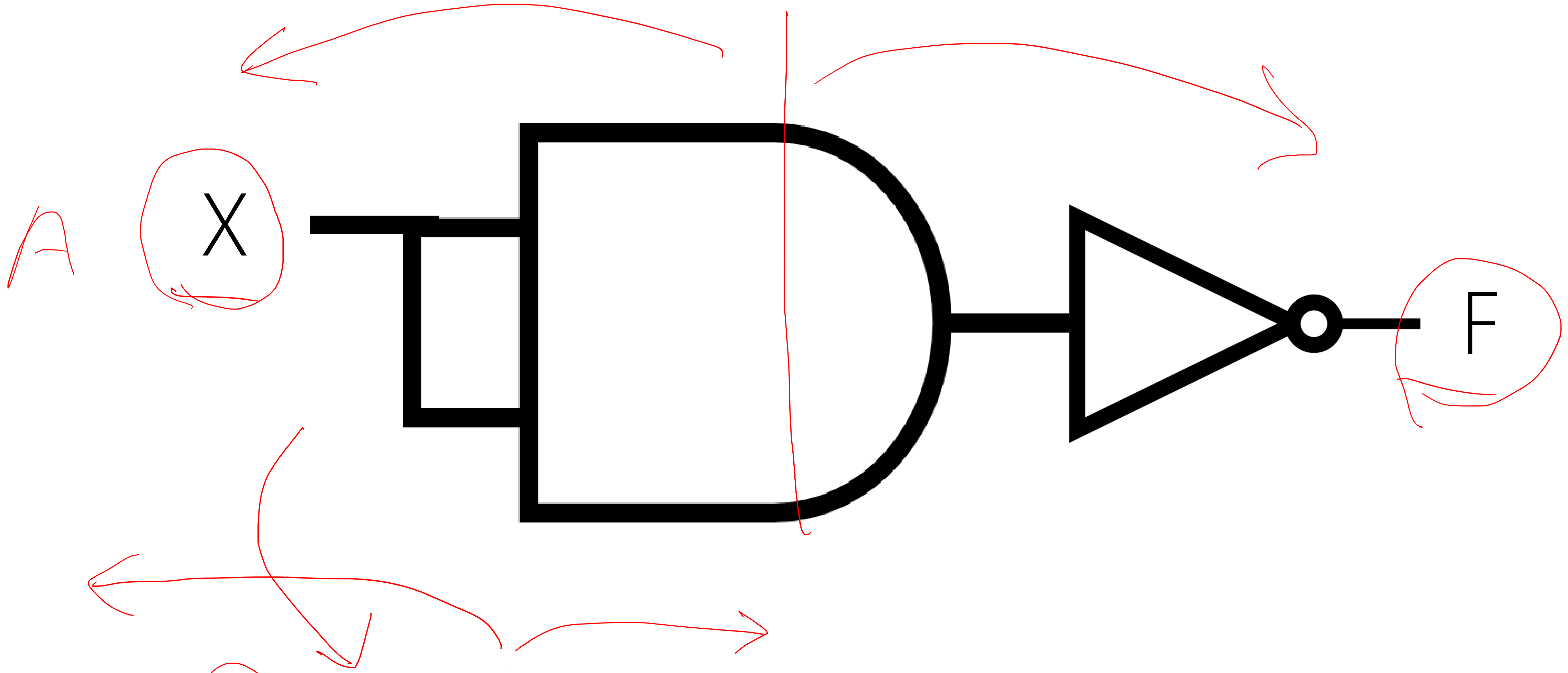
$$F = (YX)' = (XY)' = Y \uparrow X = X \uparrow Y$$

Commutative

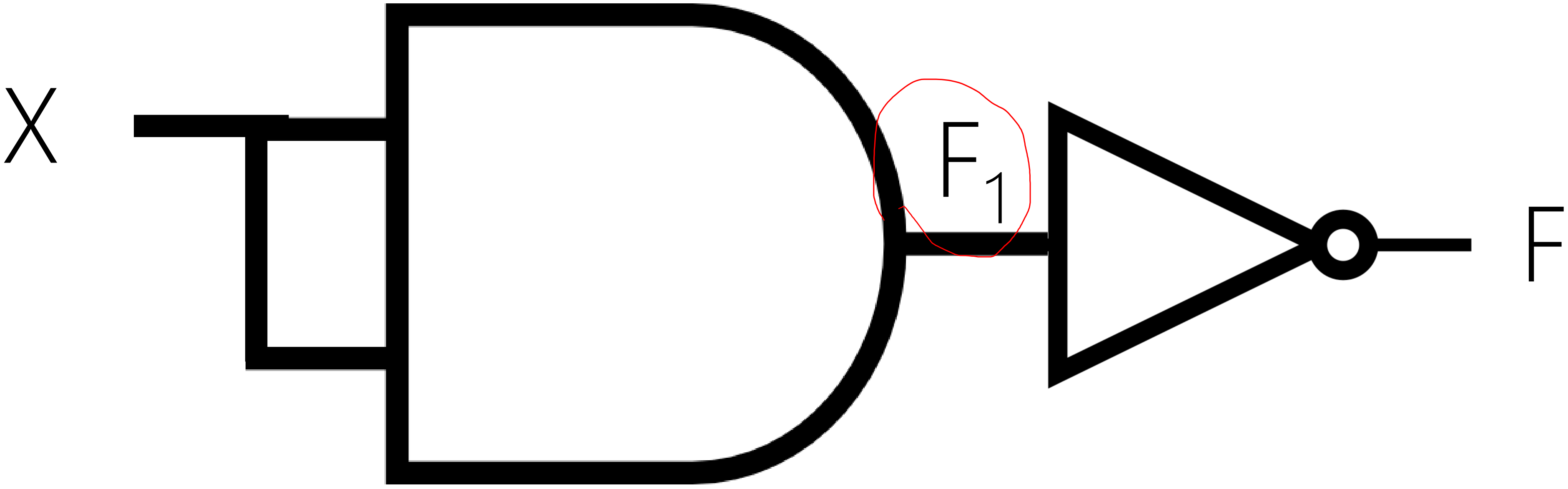
ANALYSIS

given the structure of a system, find its functionality.

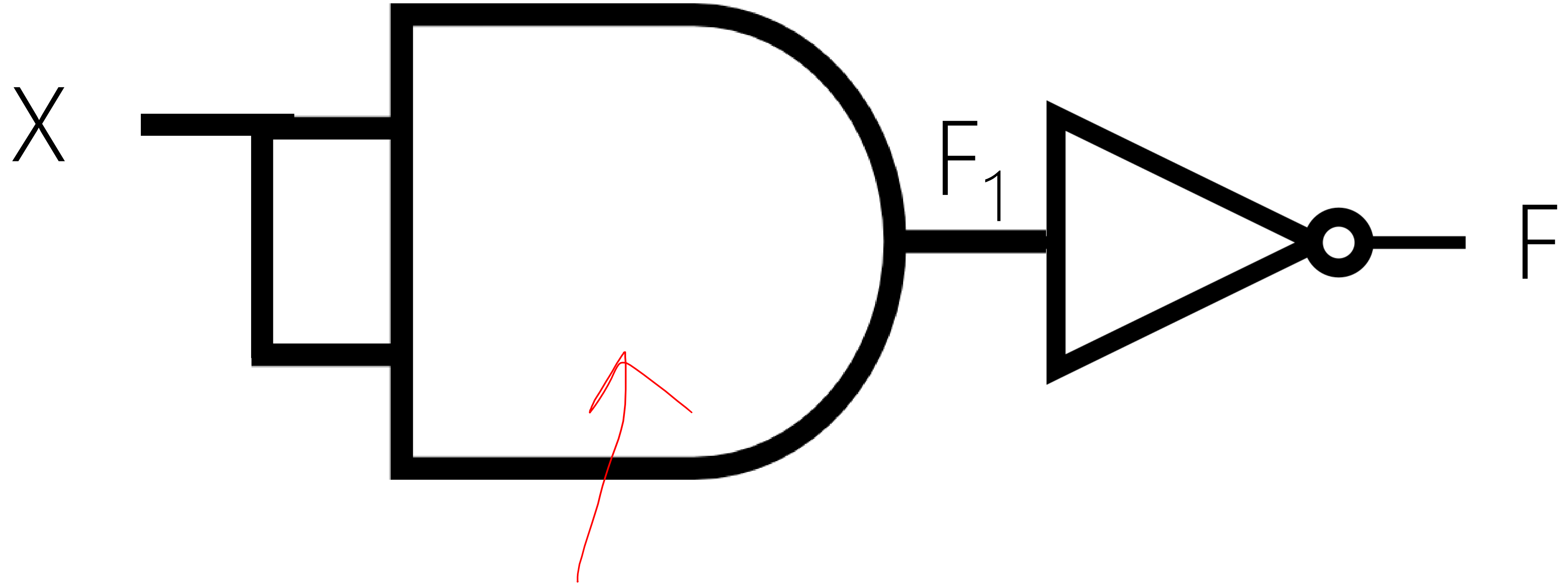
determine the functionality exhibited by a structure.



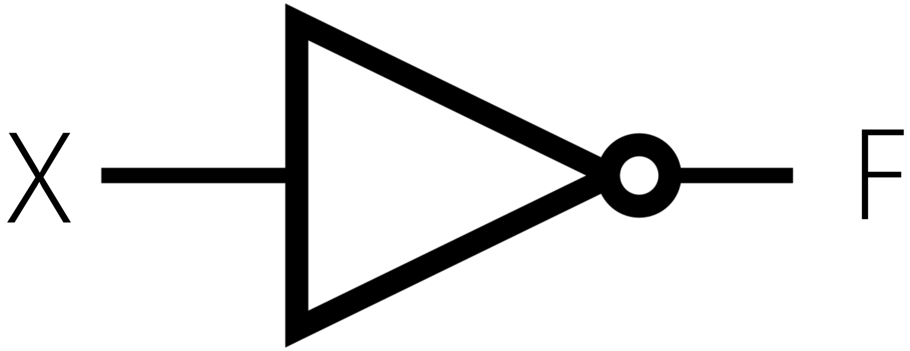
X	F = ?
0 →	
1 →	



X	$F_1 = XX$	$F = ?$
0	0	
1	1	



X	$F_1 = XX$	$F = (XX)'$
0	0	1
1	1	0

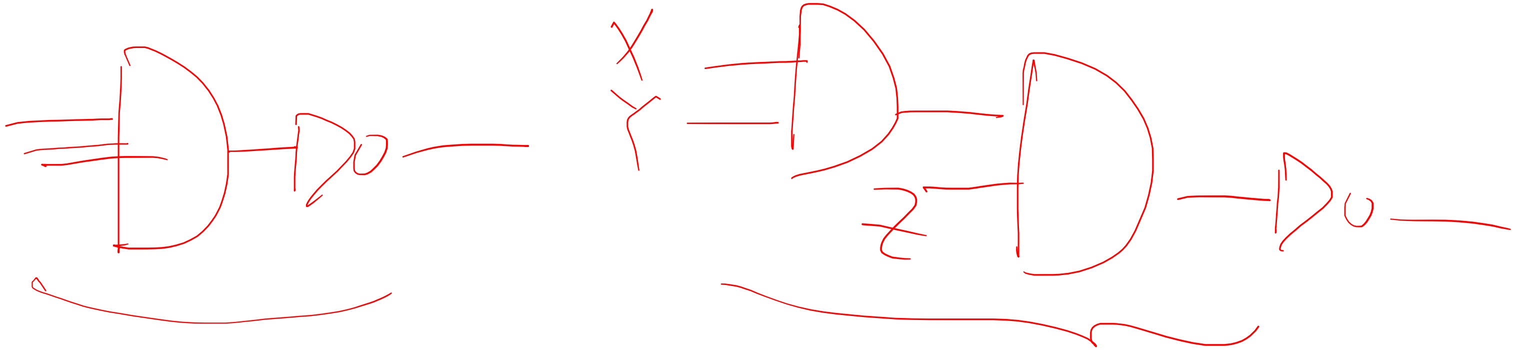


X	$F = (XX)' = X \uparrow X = X'$
0	1
1	0

DESIGN

given the functionality, design the structure of a system

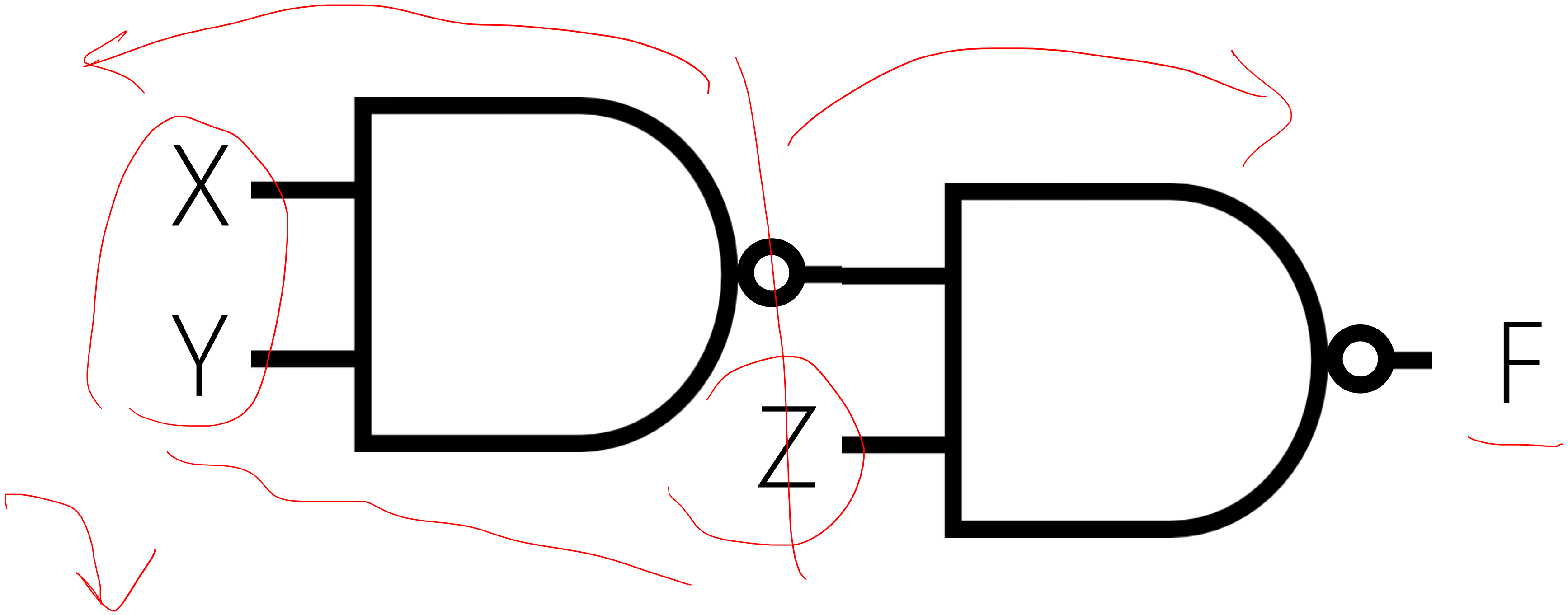
3-INPUT NAND



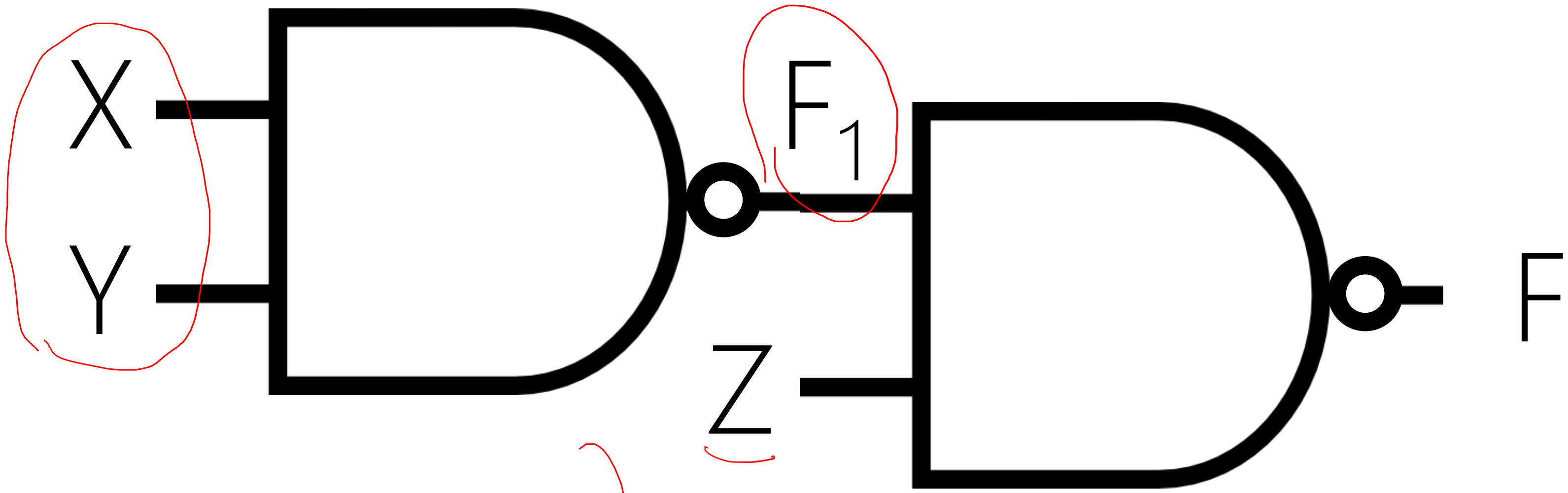
Z	Y	X	F=(ZYX)′
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

DESIGN PATTERNS

Using Same or Similar Previous Designs for New Designs

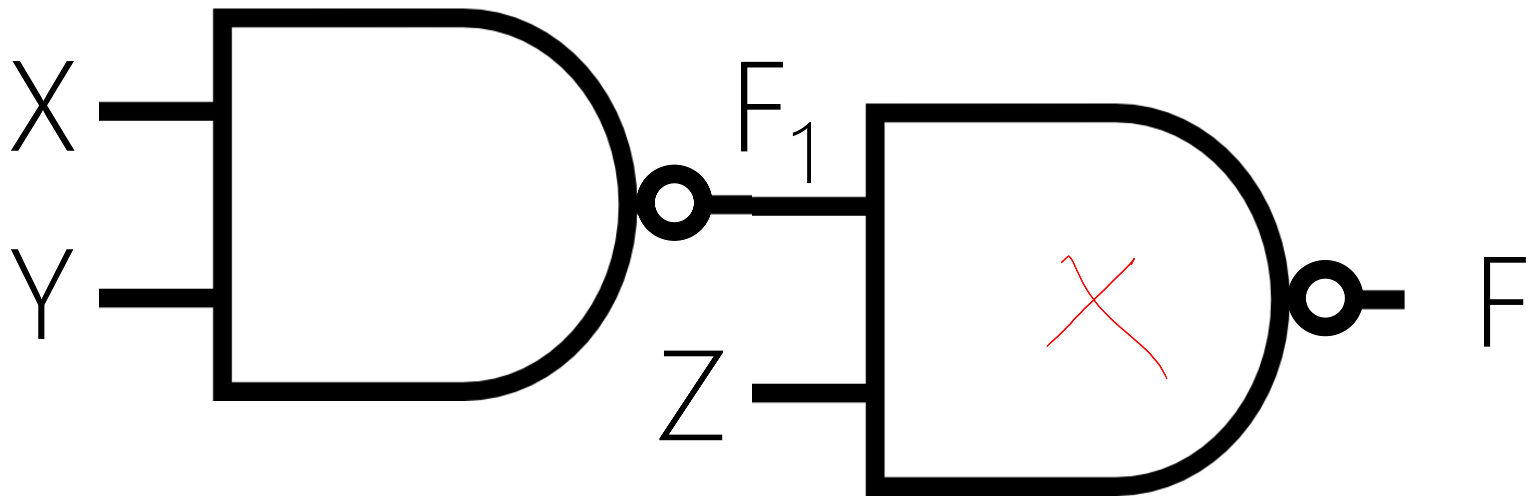


Z	Y	X	F = ?
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

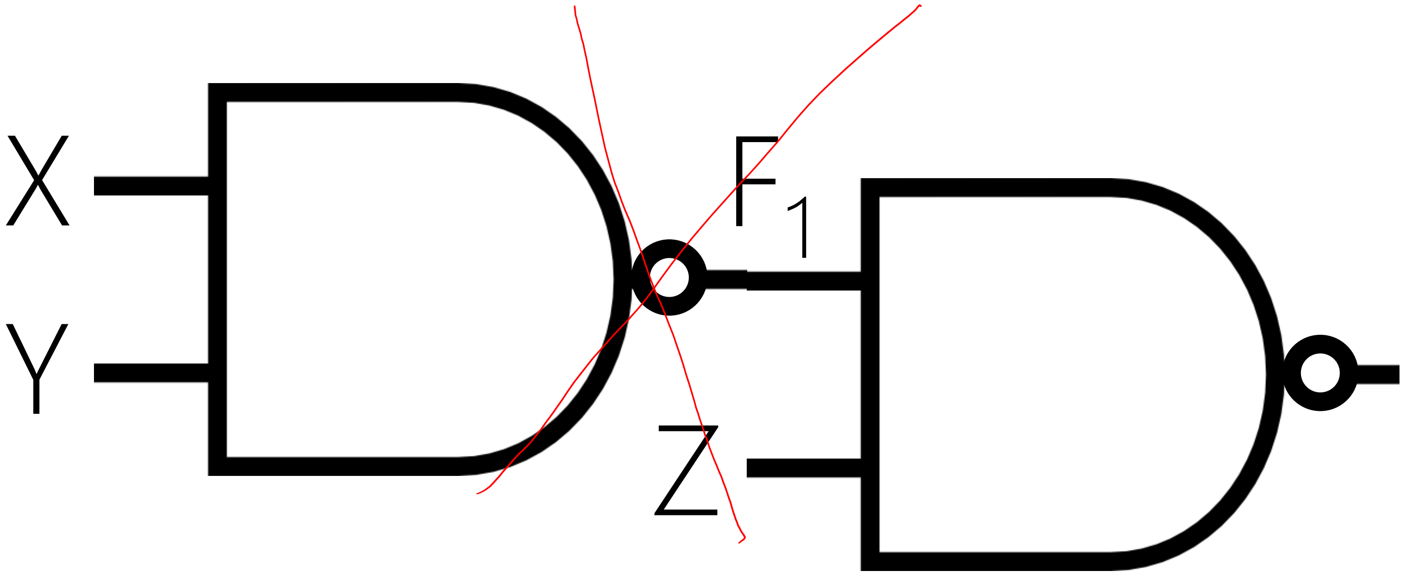


Z	Y	X	$F_1 = (YX)'$	F = ?
0	0	0	1	
0	0	1	1	
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	0	

0'



Z	Y	X	$F_1 = (YX)'$	$F = (ZF_1)' = (Z(YX)')'$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1



$$F = (Z(YX)')' \neq (ZYX)'$$

Z	Y	X	$F_1 = (YX)'$	$F = (ZF_1)' = (Z(YX)')'$	$F = (ZYX)'$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	0	1	0

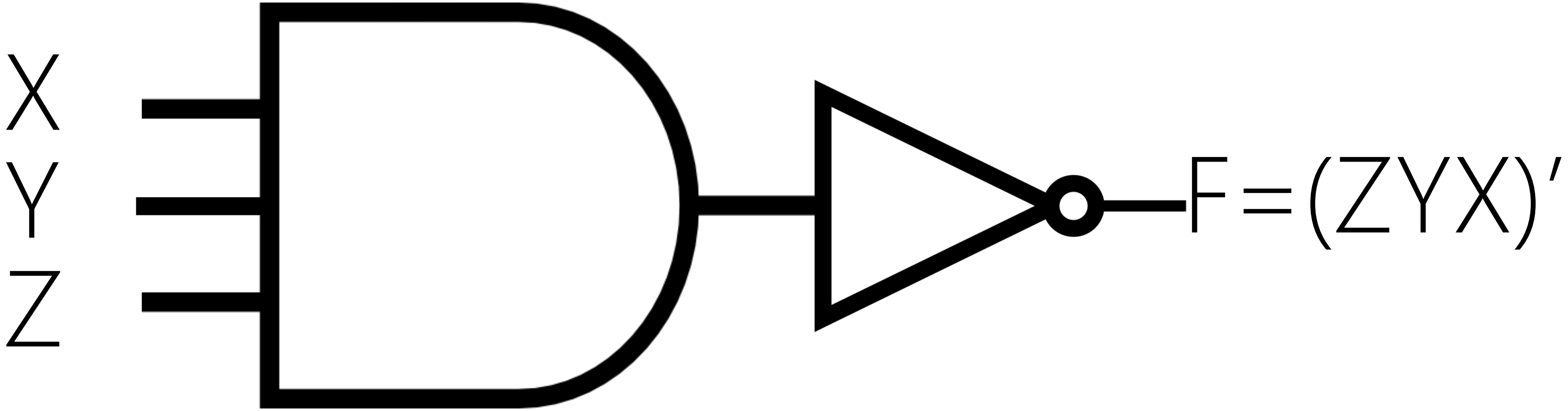
DESIGN PATTERNS

Using Same or Similar Previous Designs for New Designs

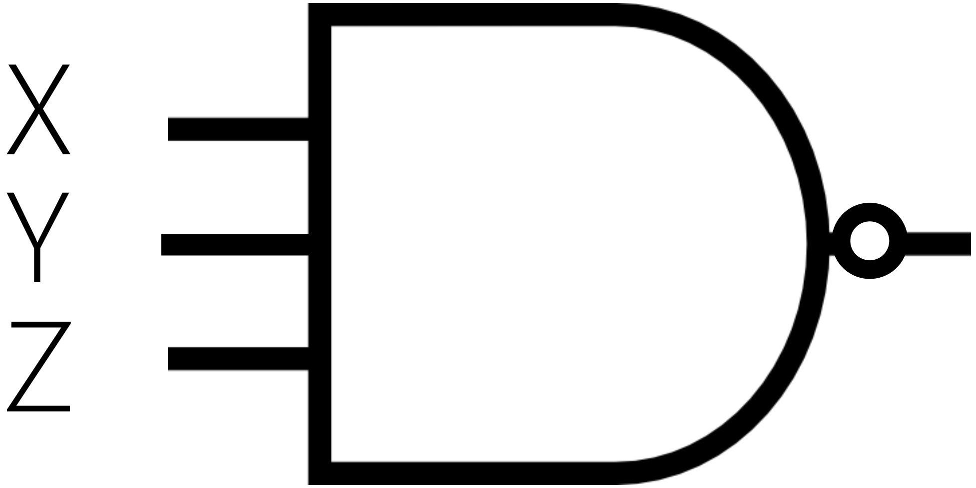
DESIGN → ANALYSIS → EVALUATION

Always Check

NOT (3-INPUT AND)



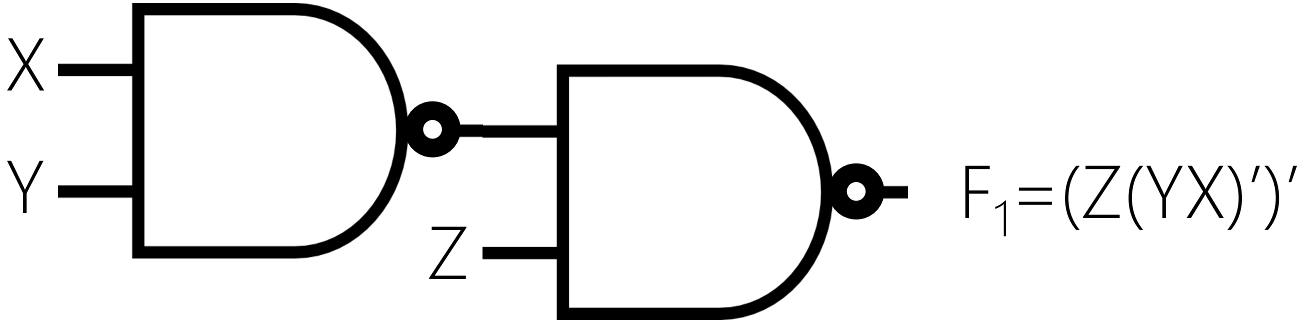
Z	Y	X	F=(ZYX)′	F=(ZYX)′
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



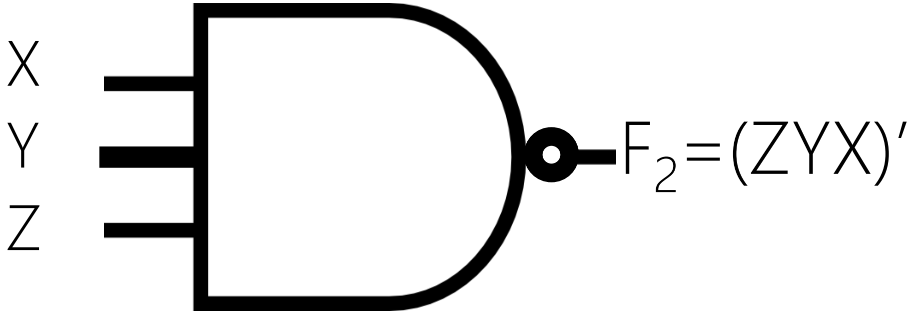
$F = (ZYX)' = (XZY)'$

Associative

Z	Y	X	F=(ZYX)′	F=(ZYX)′
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

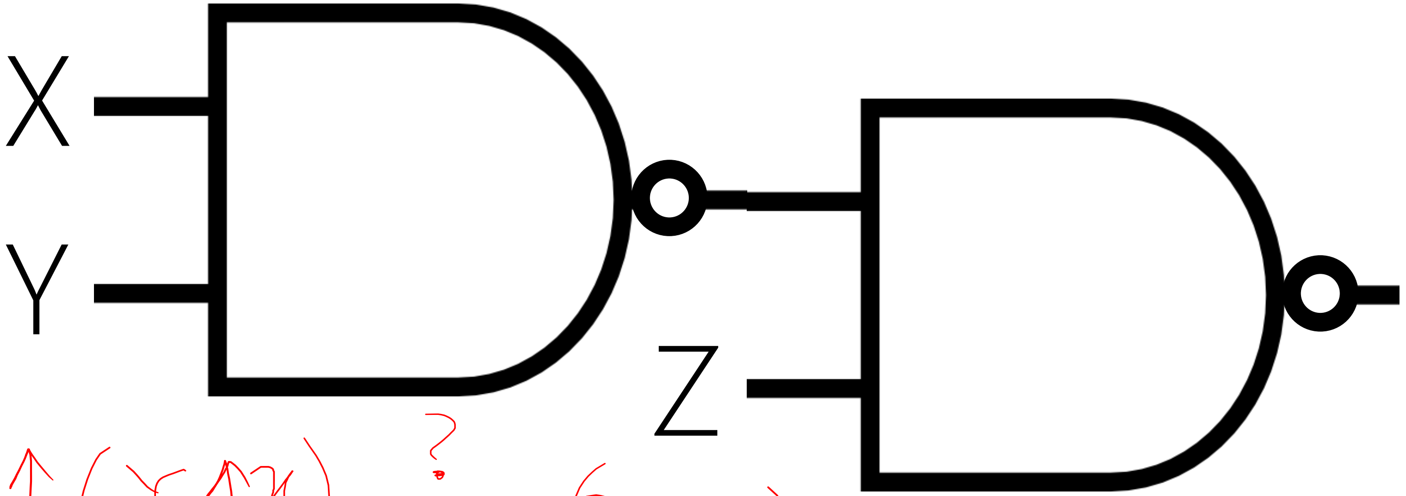


A



B

F = (ZYX)'	F_1	F_2
Effective (True)	No!	Yes
Efficient (Fast)		
Min. Cost		



$F = Z \uparrow (Y \uparrow X)$

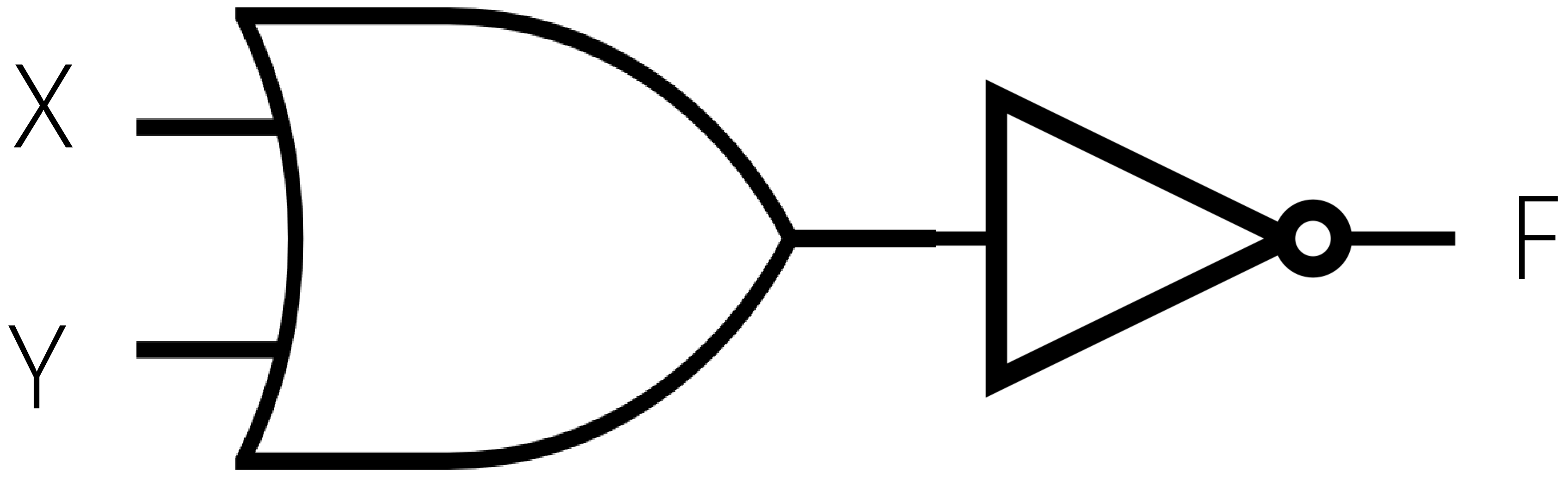
$(Z \uparrow Y) \uparrow X$

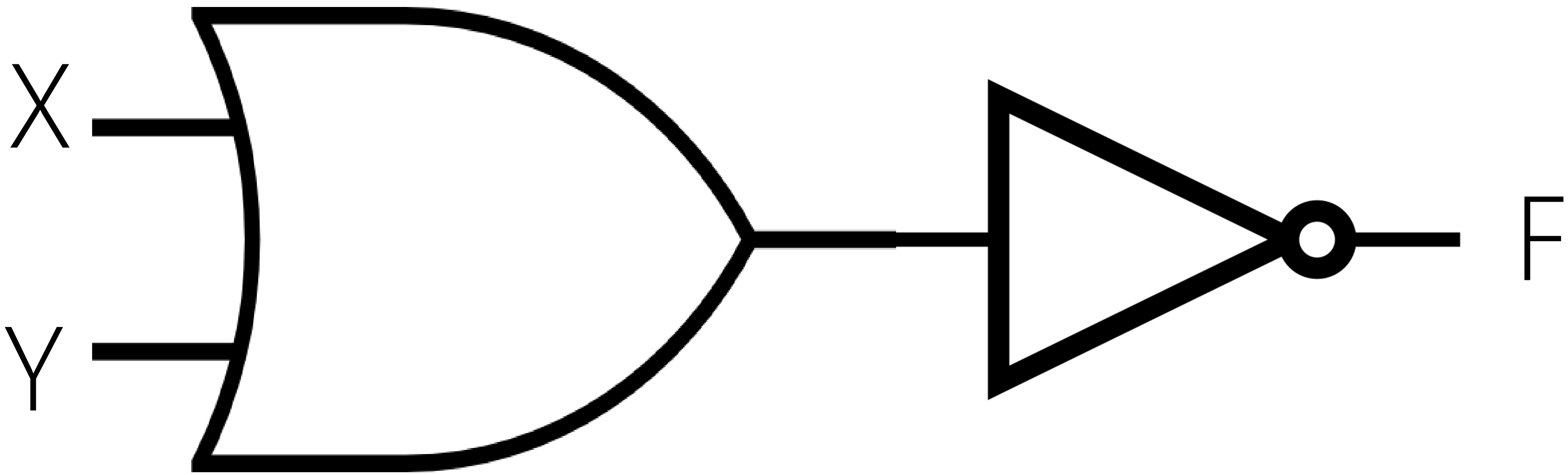
Associative?

$\neq \quad Z \uparrow (Y \uparrow X) \stackrel{?}{=} (Z \uparrow Y) \uparrow X$

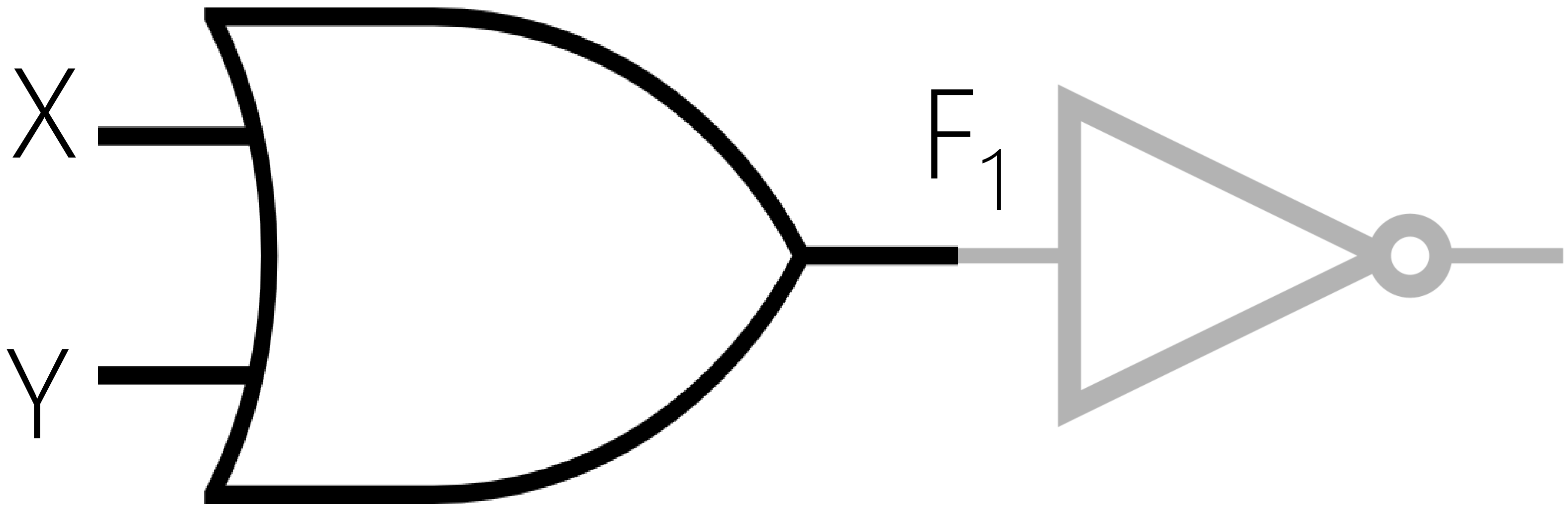
Z	Y	X	$F = (Z(YX)')' = Z \uparrow (Y \uparrow X)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

ANALYSIS

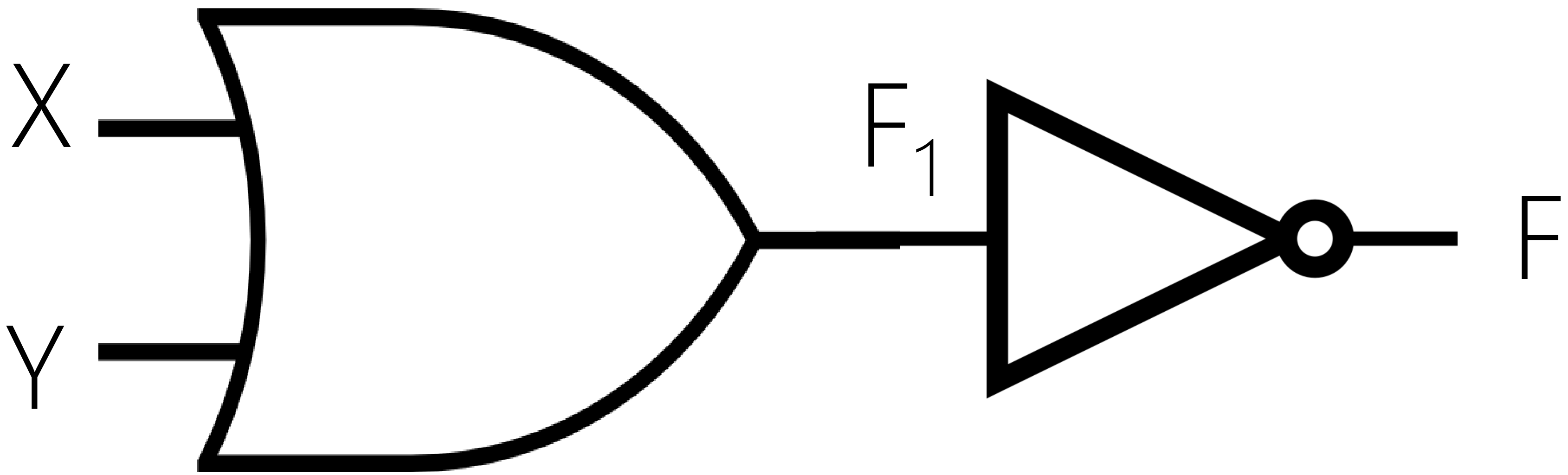

X
Y
F



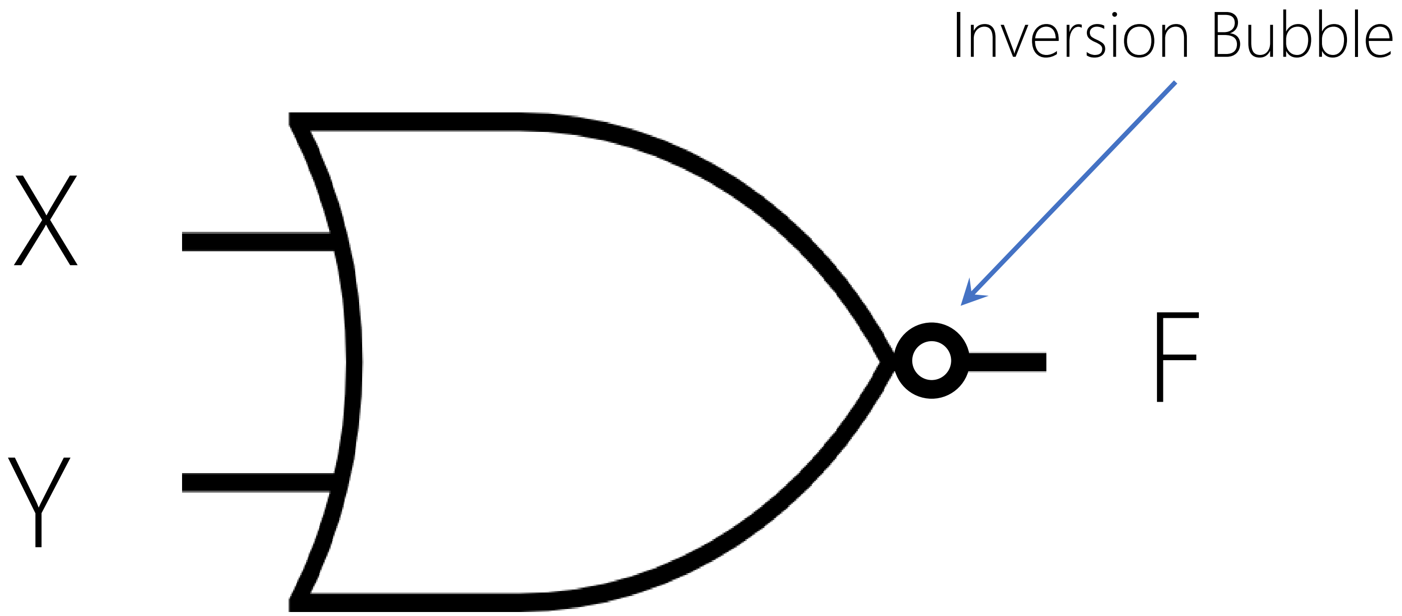
Y	X	F = ?
0	0	?
0	1	?
1	0	?
1	1	?



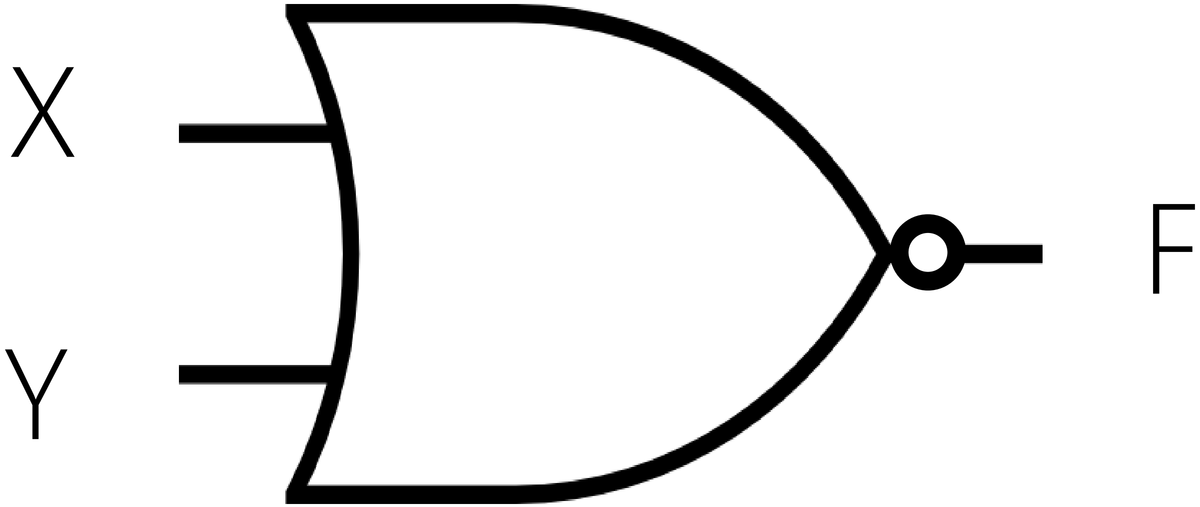
Y	X	$F_1 = Y+X$
0	0	0
0	1	1
1	0	1
1	1	1



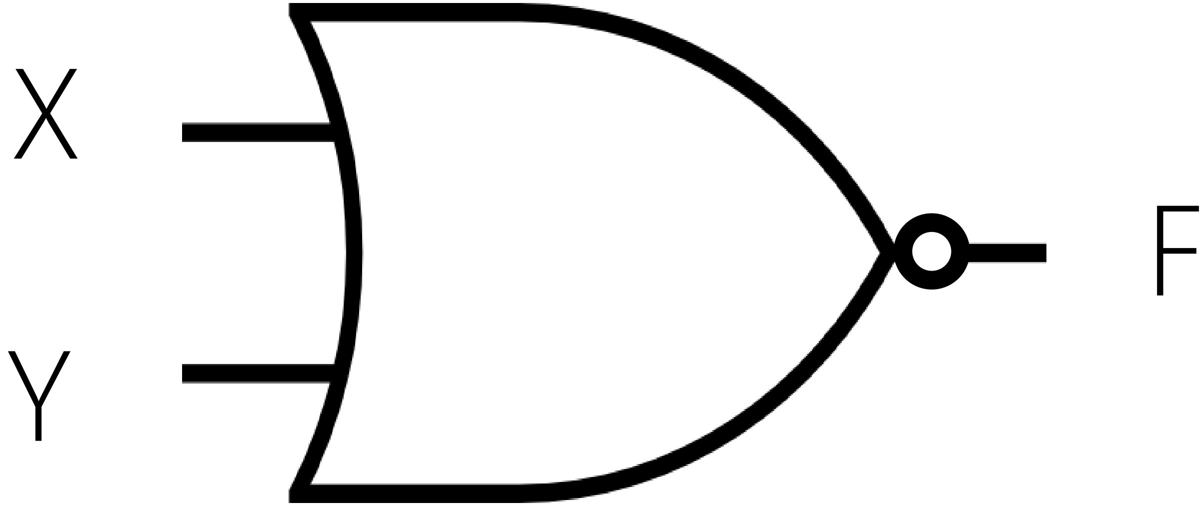
Y	X	$F_1 = Y+X$	$F = (Y+X)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



NOR (Not – OR)

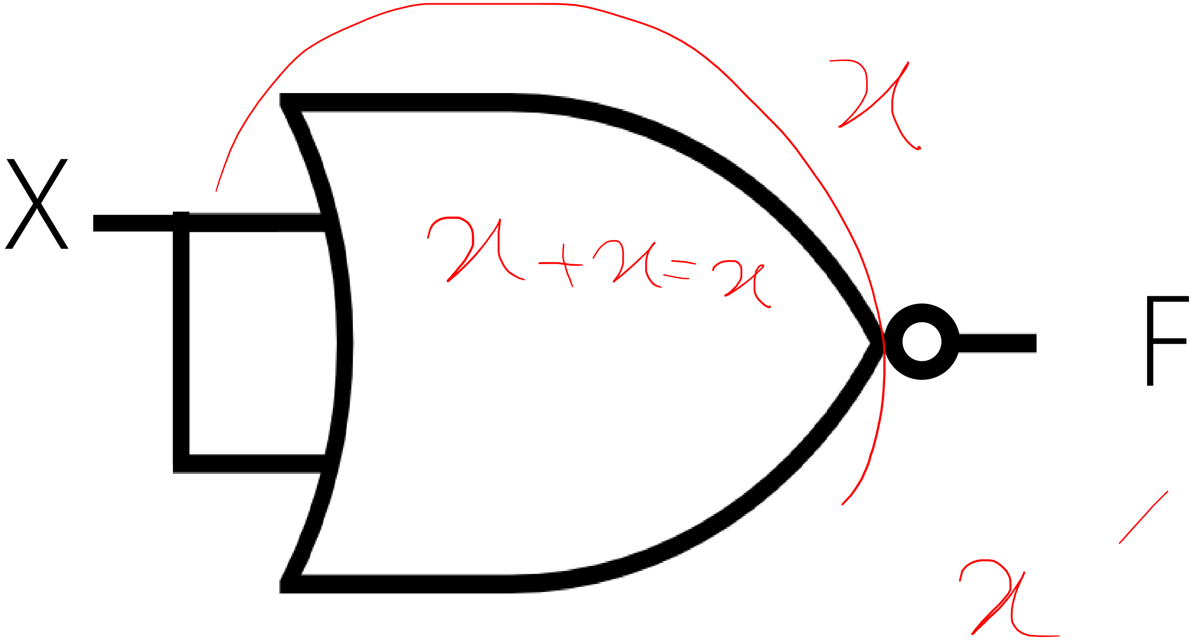


Y	X	$F = (Y+X)'$	$F = Y \downarrow X$
0	0	1	
0	1	0	
1	0	0	
1	1	0	

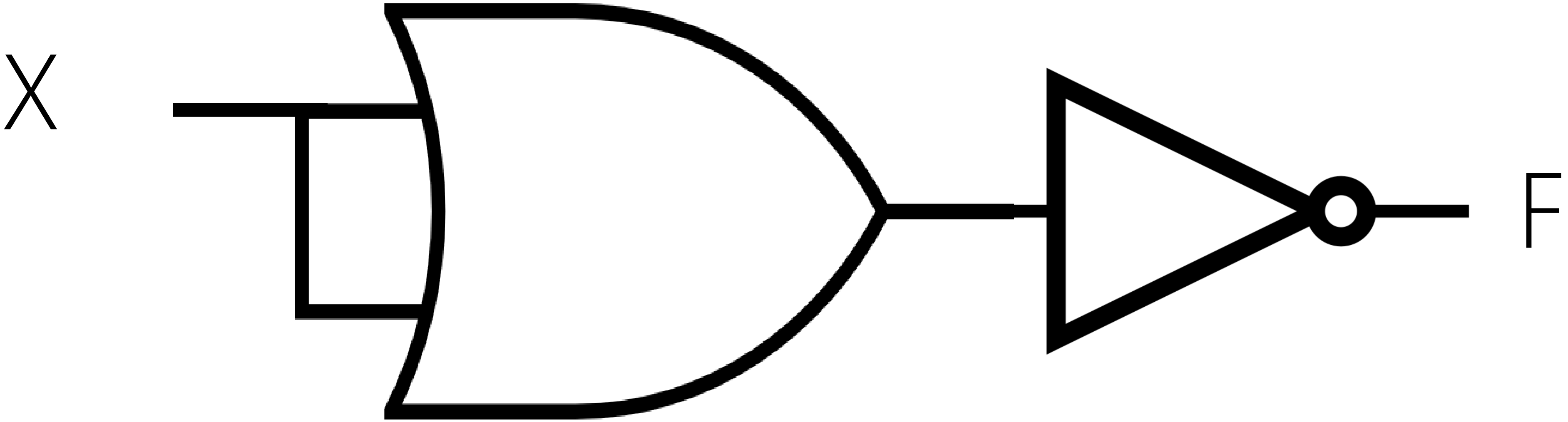


$$F = (Y+X)' = (X+Y)' = Y \downarrow X = X \downarrow Y$$

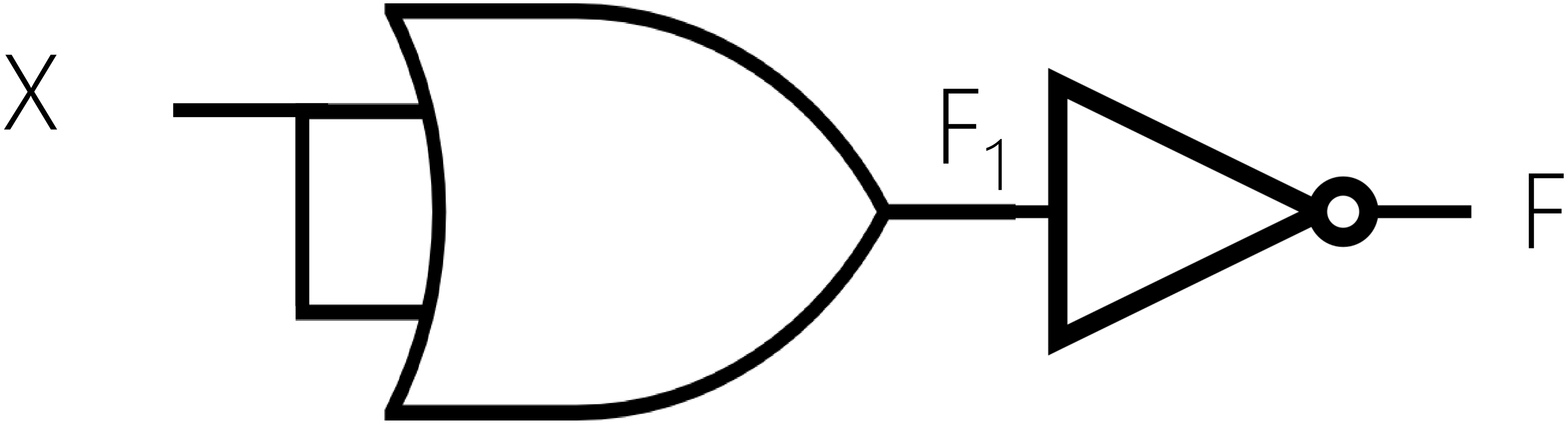
Commutative



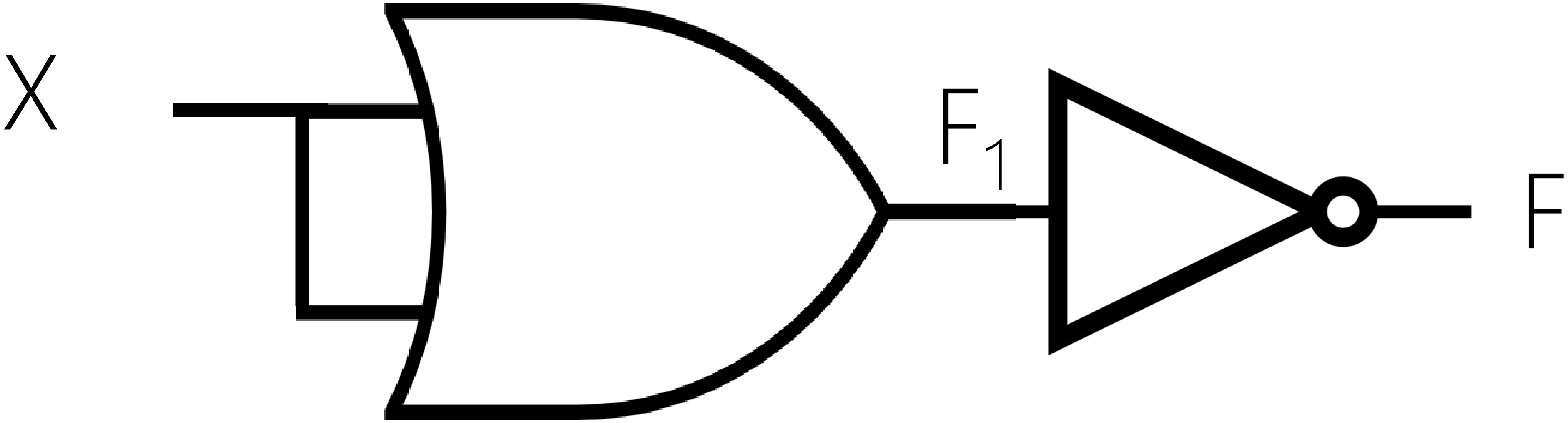
F=?



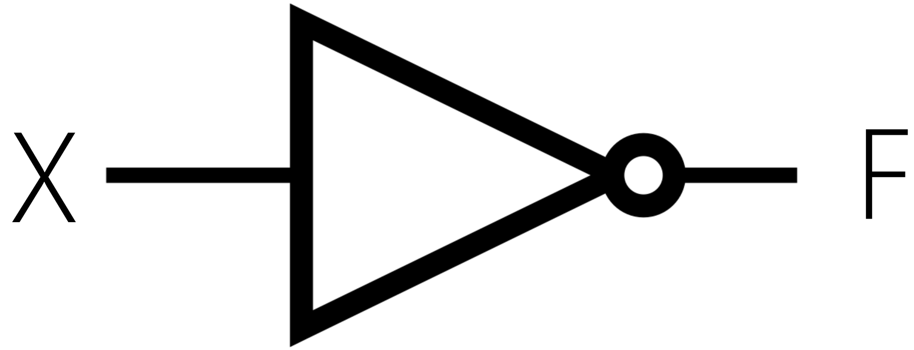
X	F = ?
0	
1	



X	$F_1 = X+X$	$F = ?$
0	0	
1	1	



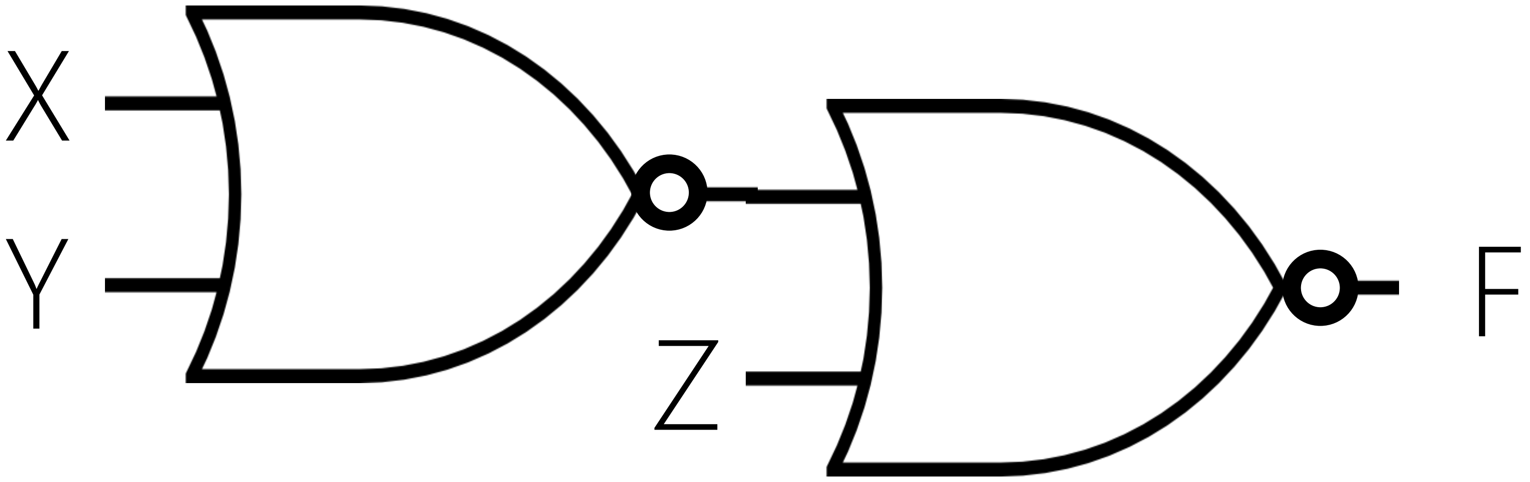
X	$F_1 = X+X$	$F = (X+X)'$
0	0	1
1	1	0



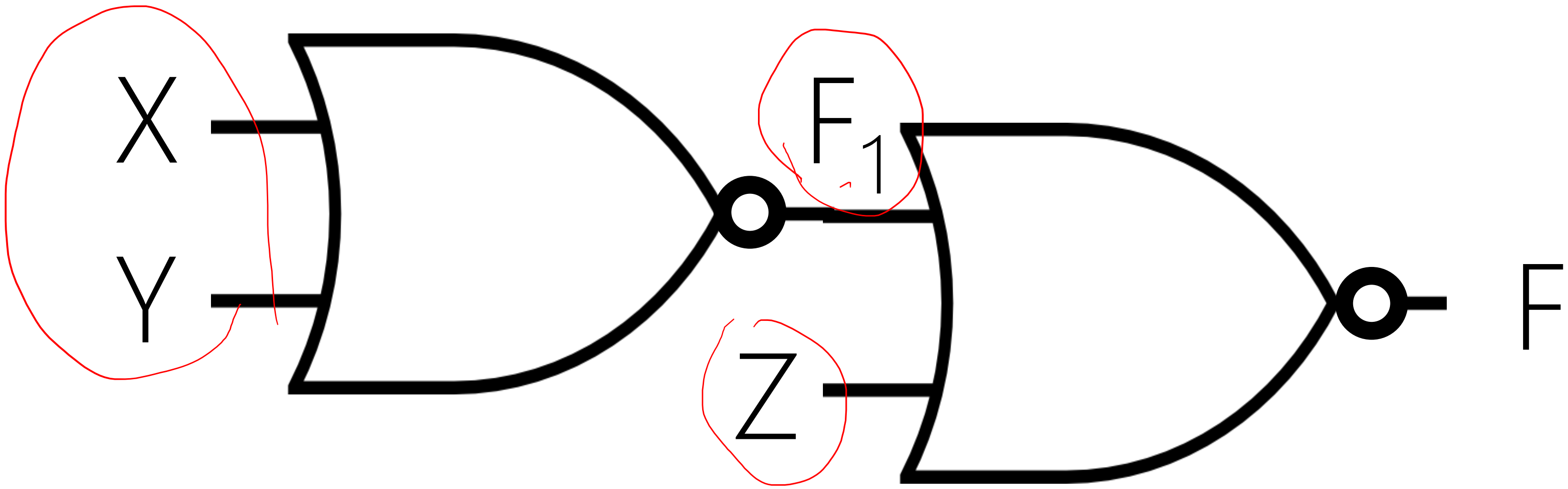
X	$F = (X+X)' = X \downarrow X = X'$
0	1
1	0

3-INPUT NOR

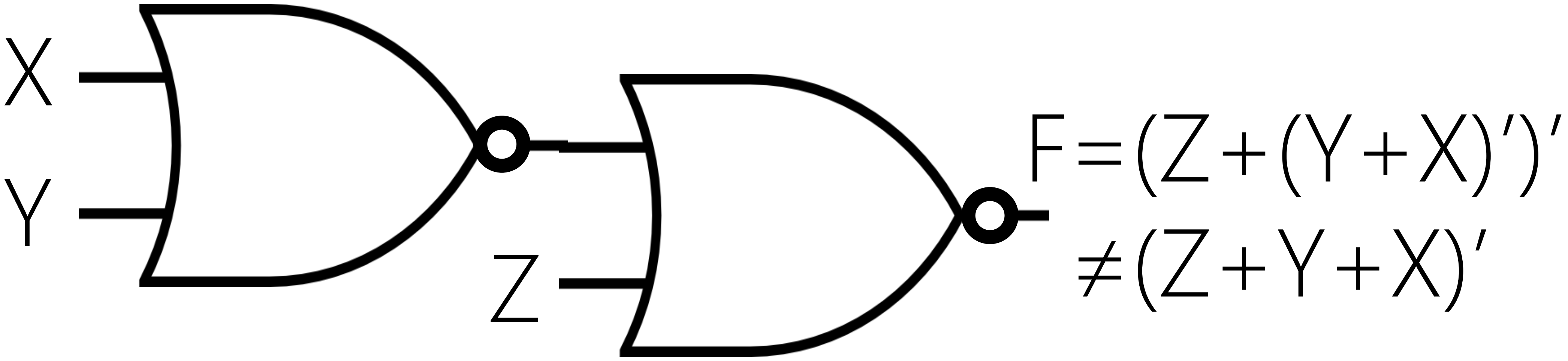
Z	Y	X	$F=(Z+Y+X)'$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



Z	Y	X	F = ?
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

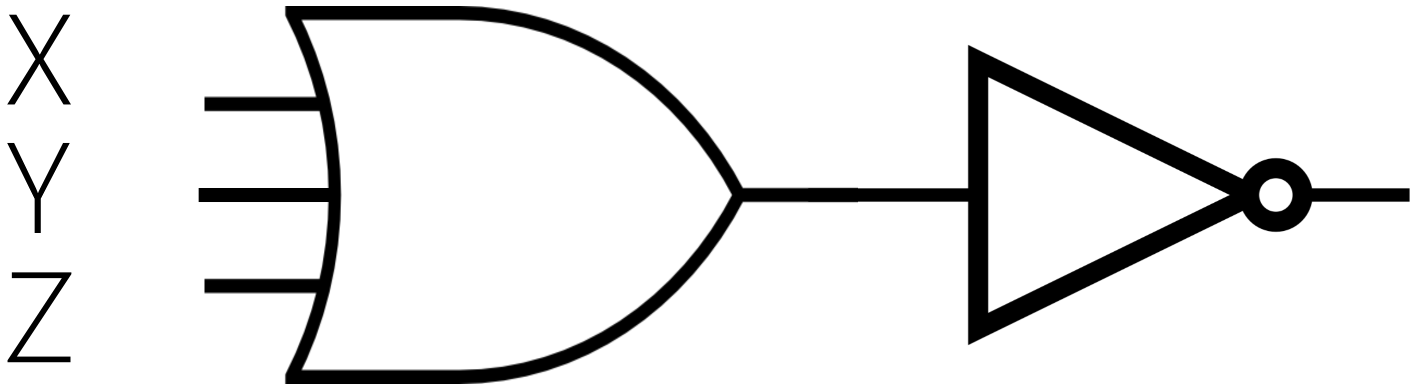


Z	Y	X	$F_1 = (Y+X)'$	$F = (Z+F_1)' = (Z+(Y+X)')'$
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0



Z	Y	X	$F = (Z+F_1)' = (Z+(Y+X)')'$	$F=(Z+Y+X)'$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

NOT (3-INPUT OR)

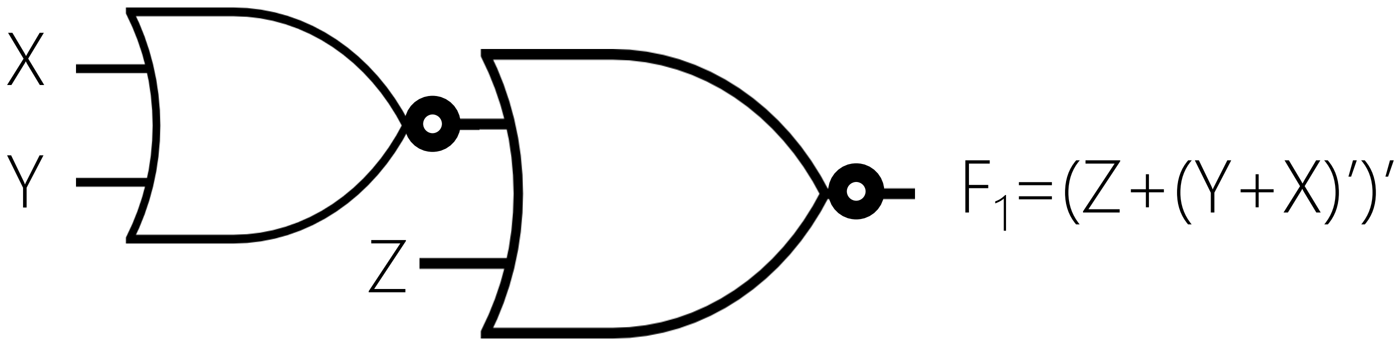


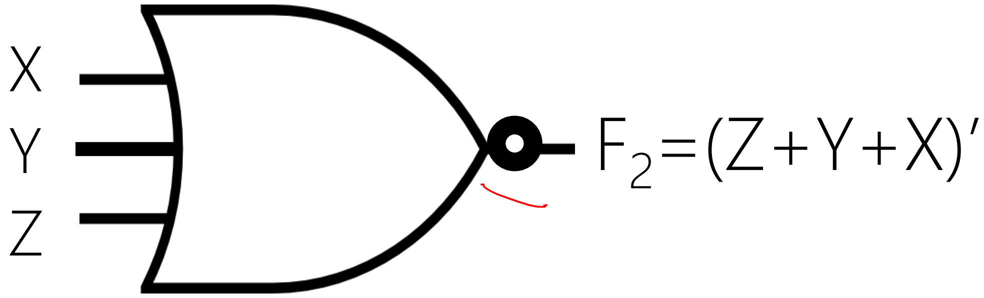
$F=(Z+Y+X)'$

$F=(X+Y+Z)'$

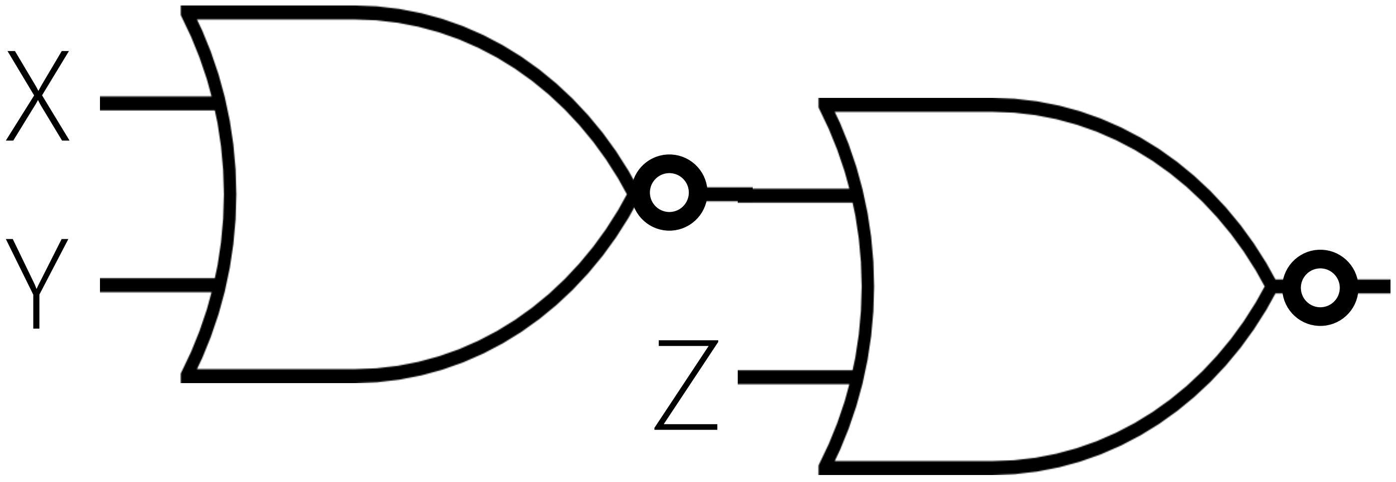
Associative

Z	Y	X	F=(Z+Y+X)′	F=(X+Y+Z)′
0	0	0	1	1
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	0





$F = (Z+Y+X)'$	F_1	F_2
Effective (True)	No!	Yes
Efficient (Fast)		
Min. Cost		



$F = Z \downarrow (Y \downarrow X)$

$(Z \downarrow Y) \downarrow X$

Associative?

Z	Y	X	$F = (Z(YX)')' = Z \downarrow (Y \downarrow X)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

RECAP

GATE	WHEN F=1
NOT	0
AND	
OR	
NAND	
NOR	

GATE	WHEN F=1
NOT	The input is 0
AND	All the inputs are 1
OR	At least one input is 1
NAND	At least one input is 0
NOR	All the inputs are 0

DESIGN

a design algorithm for any digital units (logic circuits), given truth table

1. minterm

aka. Standard Product

X′ vs. X

1 binary variable appear either:
- in its normal form X, or
- in its complement form X′

m_0	X'
m_1	X

$$YX \text{ vs. } YX' \text{ vs. } Y'X \text{ vs. } Y'X'$$

2 binary variables appear either in one of these forms:

m_0	00	$Y'X'$
m_1		$Y'X$
m_2	10	YX'
m_3	11	YX

ZYX vs. ZYX' vs. …

3 binary variables appear either in one of these forms: how many?

ZYX vs. ZYX' vs. …

3 binary variables appear either in one of these forms: how many?
Each variable can take 2 forms (normal and complement)
We have 3 variables, $2 \times 2 \times 2 = 2^3 = 8$

m_0	$Z'Y'X'$
m_1	$Z'Y'X$
m_2	$Z'YX'$
m_3	$Z'YX$
m_4	$ZY'X'$
m_5	$ZY'X$
m_6	ZYX'
m_7	ZYX

$A_{n-1} \cdots A_2 A_1 A_0$ vs. $A_{n-1} \cdots A_2 A_1 A'_0$ …

n binary variables appear either in one of these forms: how many?

Each variable can take 2 forms (normal and complement)

We have n variables, $2 \times 2 \times 2 \times \cdots \times 2 = 2^n$

m_0	$A'_{n-1} \cdots A'_2 A'_1 A'_0$
m_1	$A'_{n-1} \cdots A'_2 A'_1 A_0$
m_2	$A'_{n-1} \cdots A'_2 A_1 A'_0$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
m_{2^n-3}	$A_{n-1} \cdots A_2 A'_1 A_0$
m_{2^n-2}	$A_{n-1} \cdots A_2 A_1 A'_0$
m_{2^n-1}	$A_{n-1} \cdots A_2 A_1 A_0$

2. TRUTH TABLE

en.wikipedia.org/wiki/Truth_table

X	F = F(X) = ?
0	?
1	?

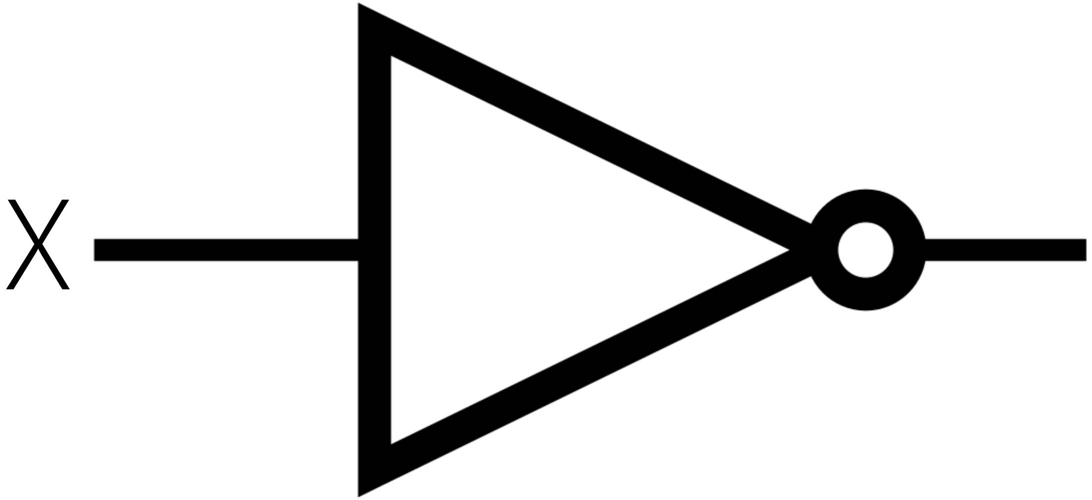
X	F = F(X) = 0
0	0
1	0

X	$F = F(X) = X'$
0	1
1	0

m_0	X'
m_1	X

X	$F = F(X) = X' = m_0$
0	1
1	0

X	$F = F(X) = X' = m_0$
0	1
1	0



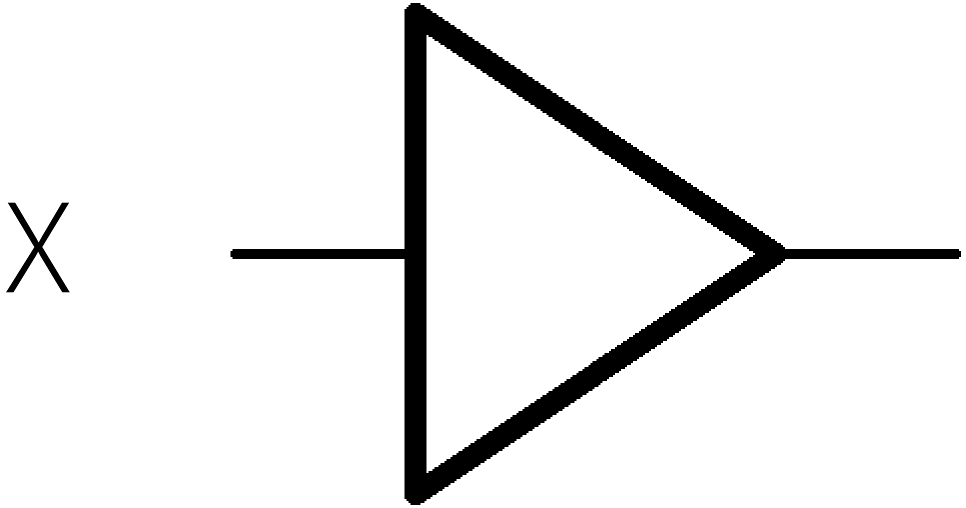
$F = X' = m_0$

X	F = F(X) = X
0	0
1	1

X	$F = F(X) = X = m_1$
0	0
1	1

X	F = F(X) = X = m1
0	0
1	1

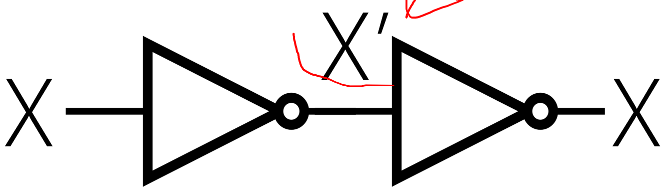
$f(x) = x$



$F = X = m_1$

Digital Buffer

$(x')' = x$

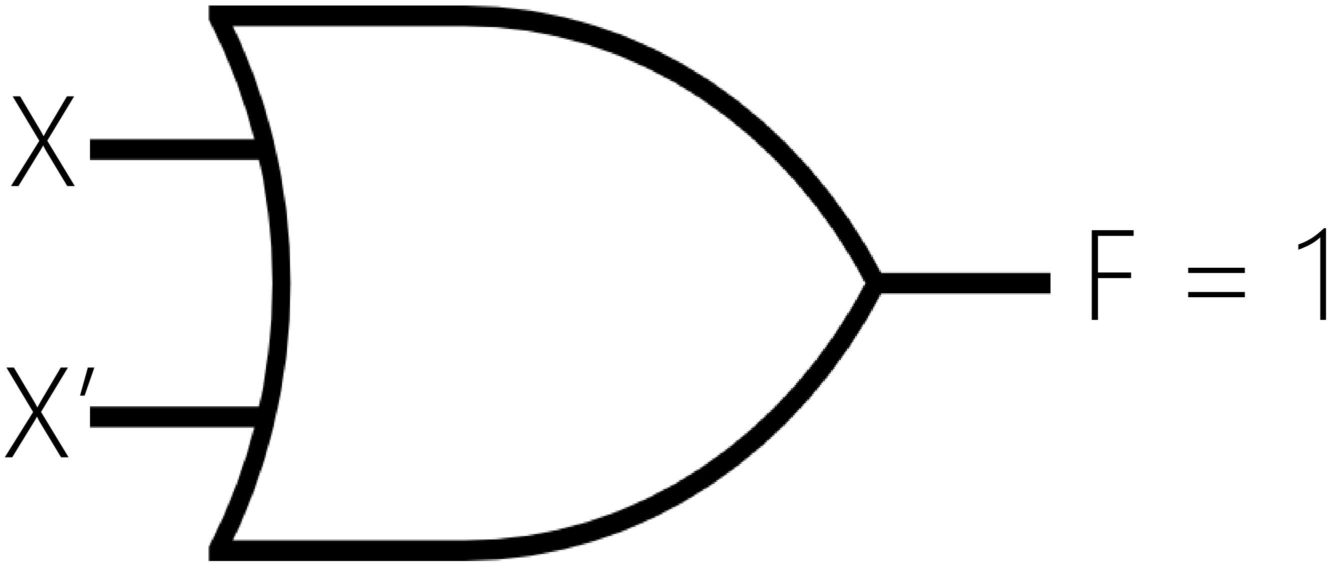


X	F = F(X)
0	1
1	1

X	$F = F(X) = X' = m_0$
0	1
1	1

X	$F = F(X) = X' + X = m_0 + m_1$
0	1
1	1

1



X′	X	X′+X
1	0	1
0	1	1

$$F = X + X' = 1$$

TRUTH TABLE $\leftrightarrow$ minterm