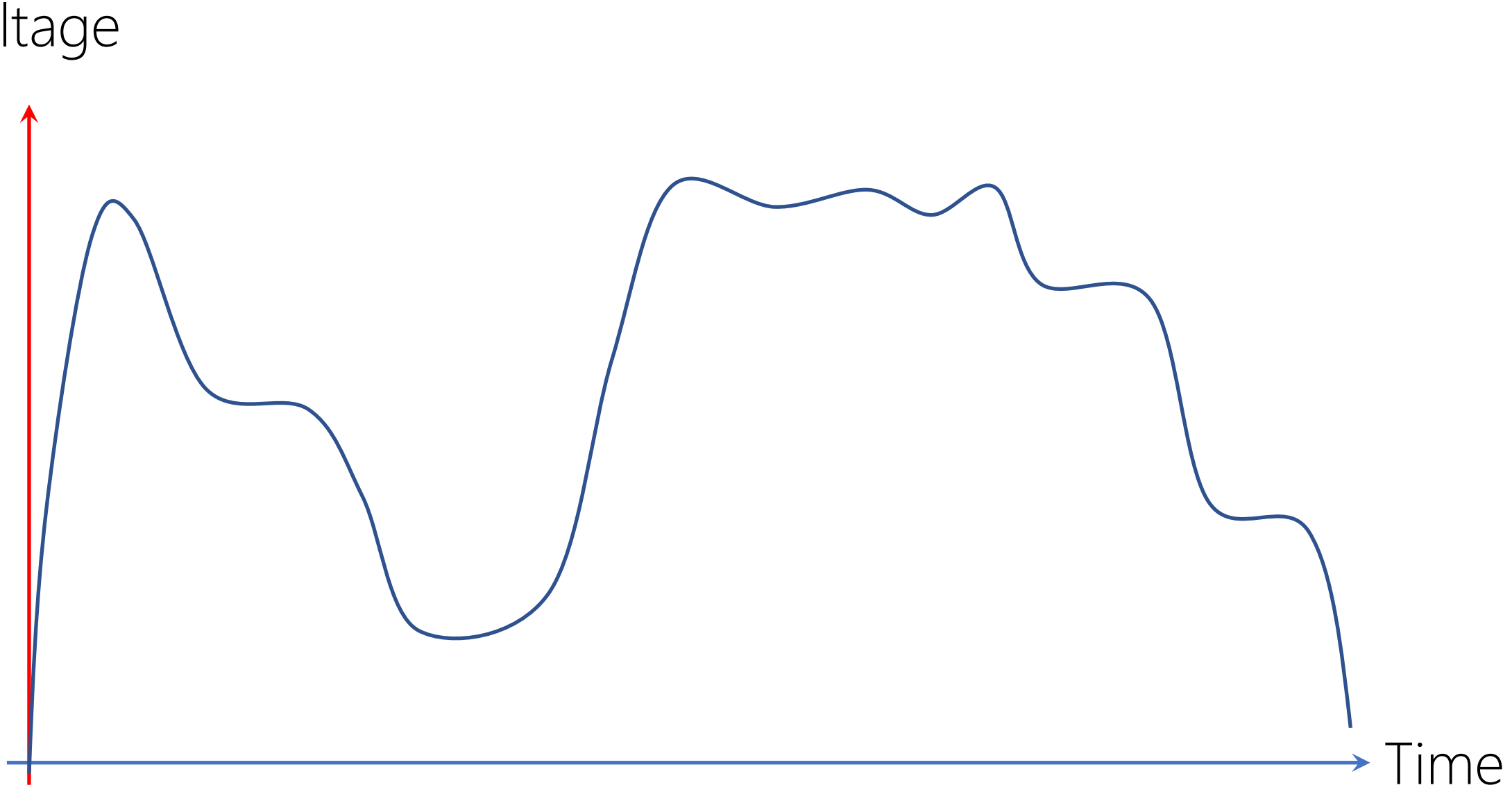
ANALOG SYSTEMS

Continuous


Voltage
Time

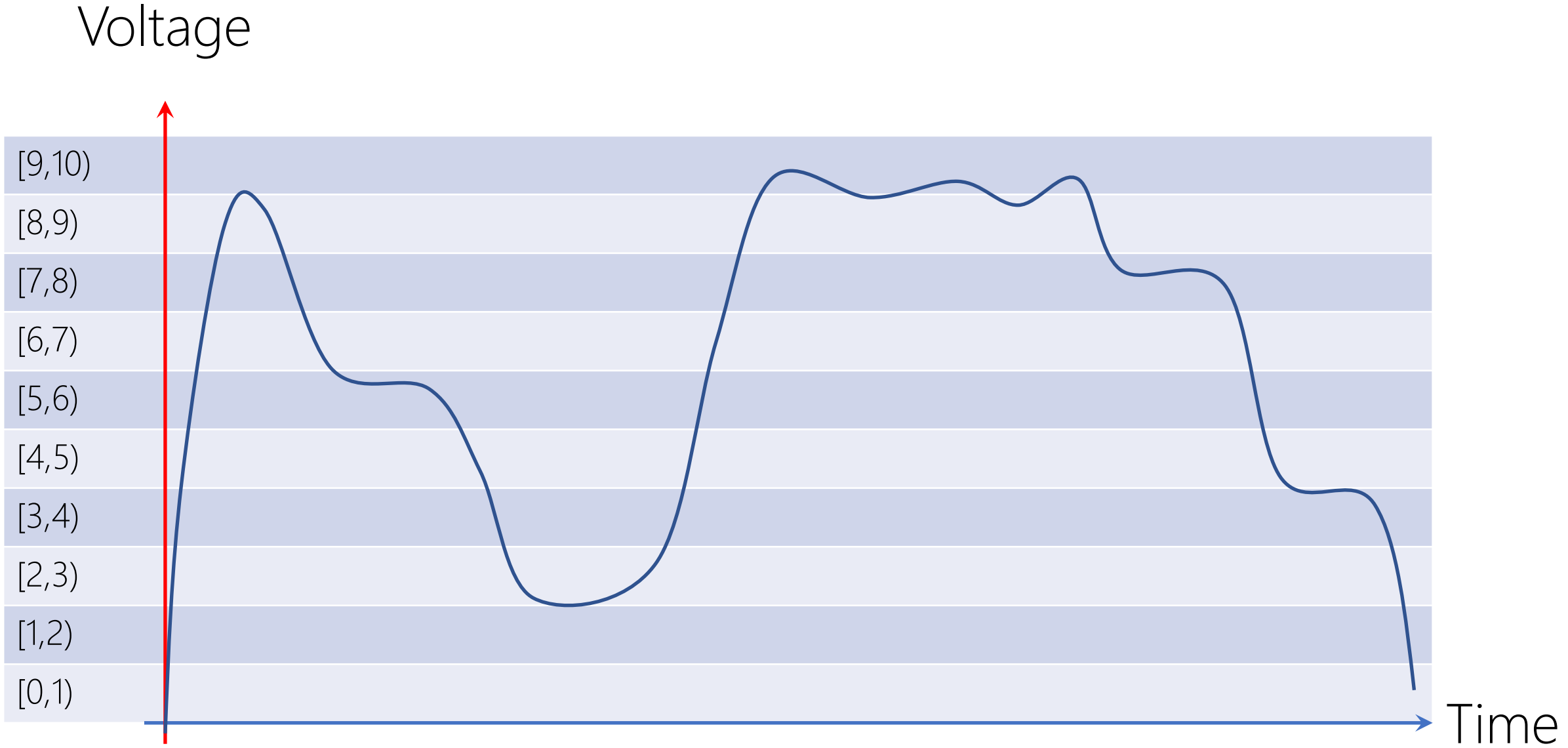
DIGITAL SYSTEMS

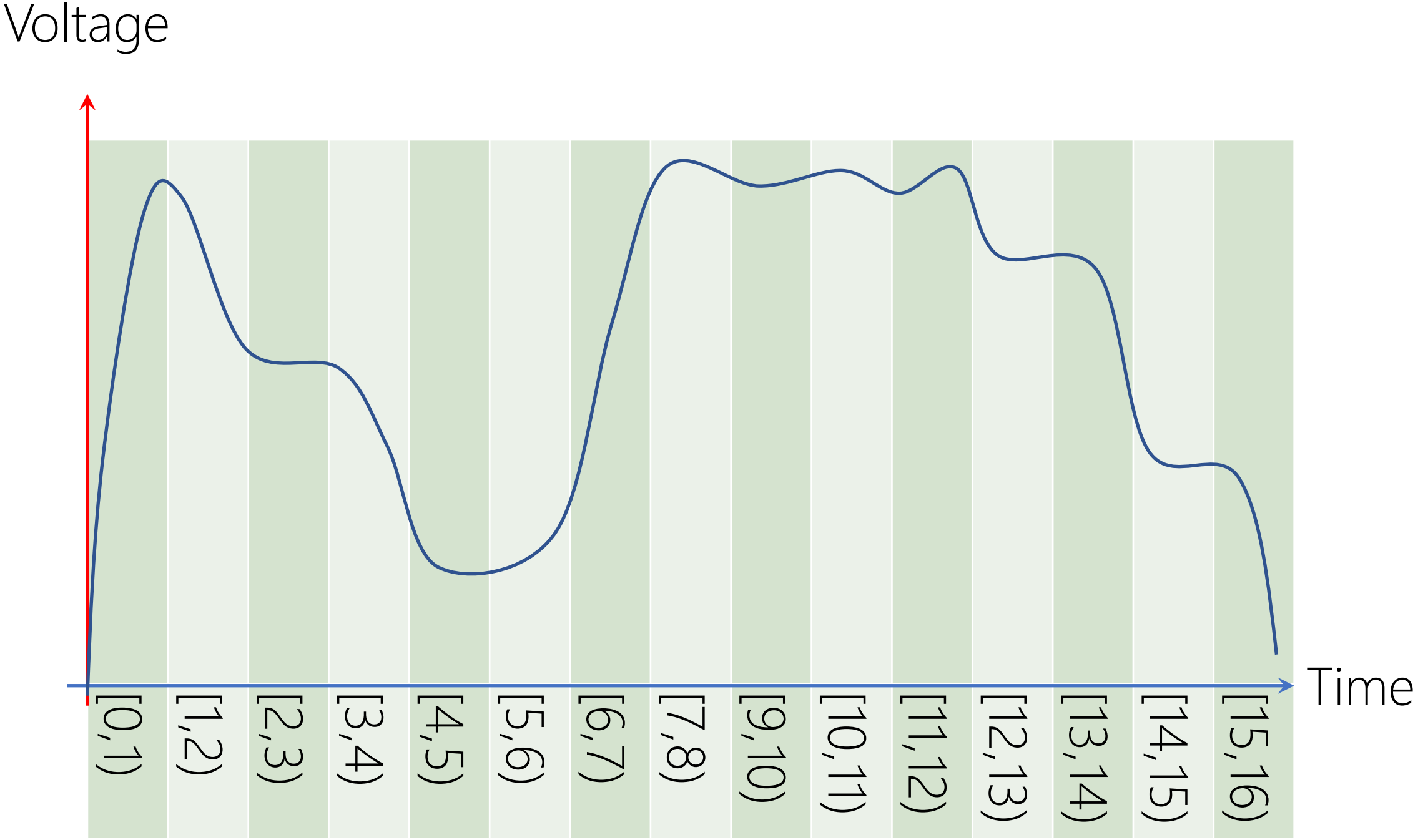
Discrete

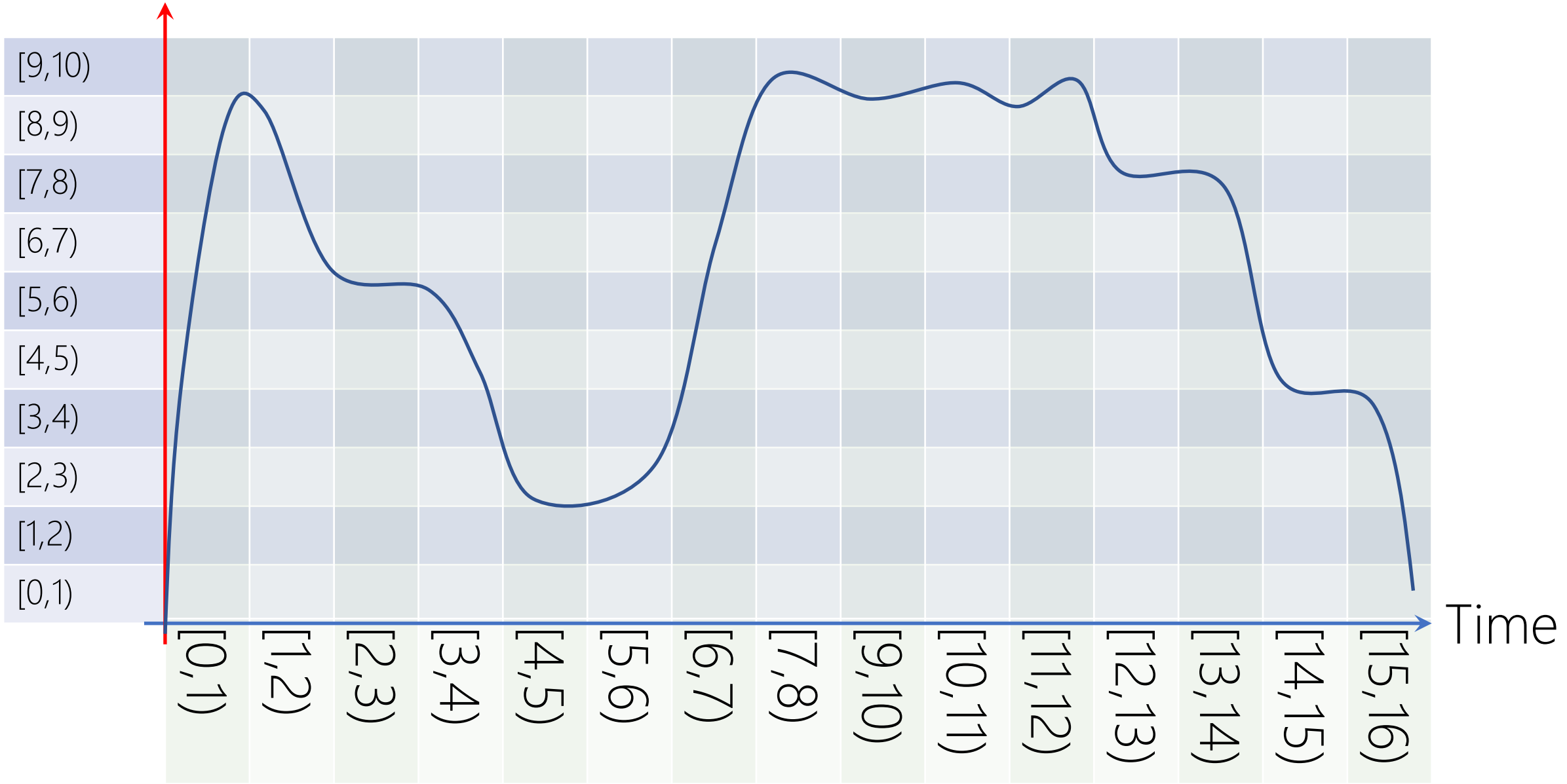
ELECTRICITY NUMBER SYSTEM

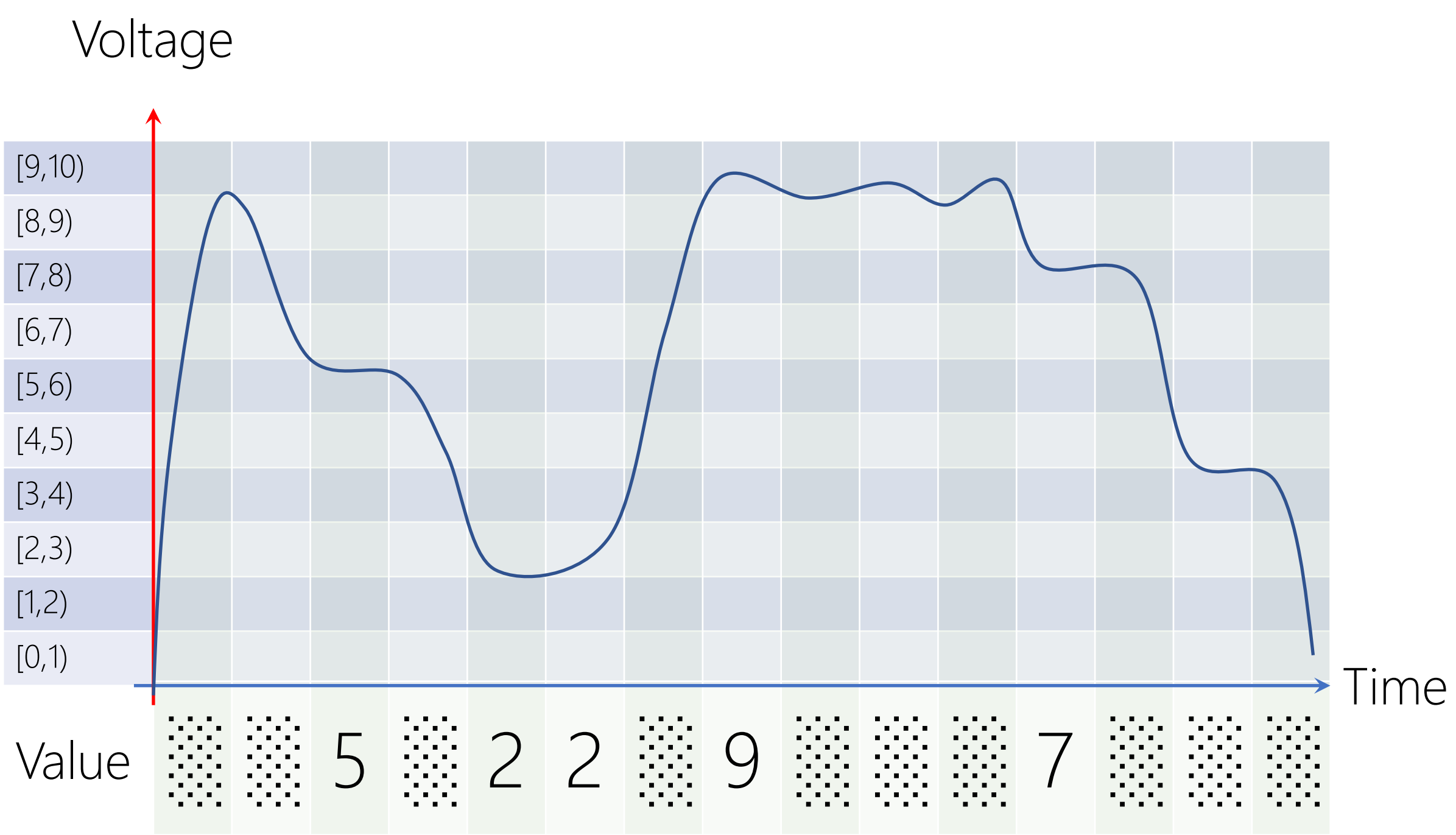
BASE-?

BASE-10

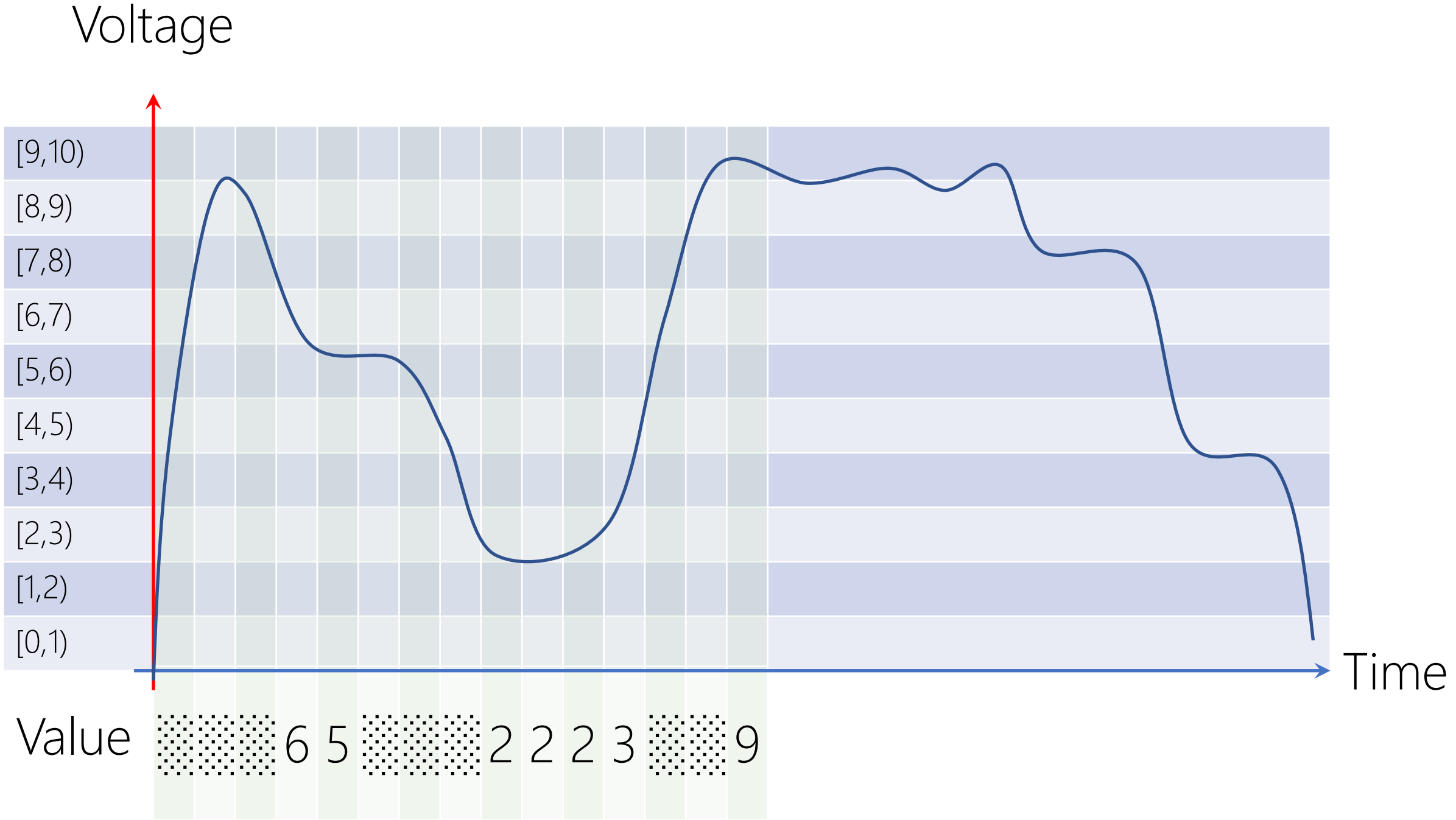

Voltage
[9,10)
[8,9)
[7,8)
[6,7)
[5,6)
[4,5)
[3,4)
[2,3)
[1,2)
[0,1)
Time

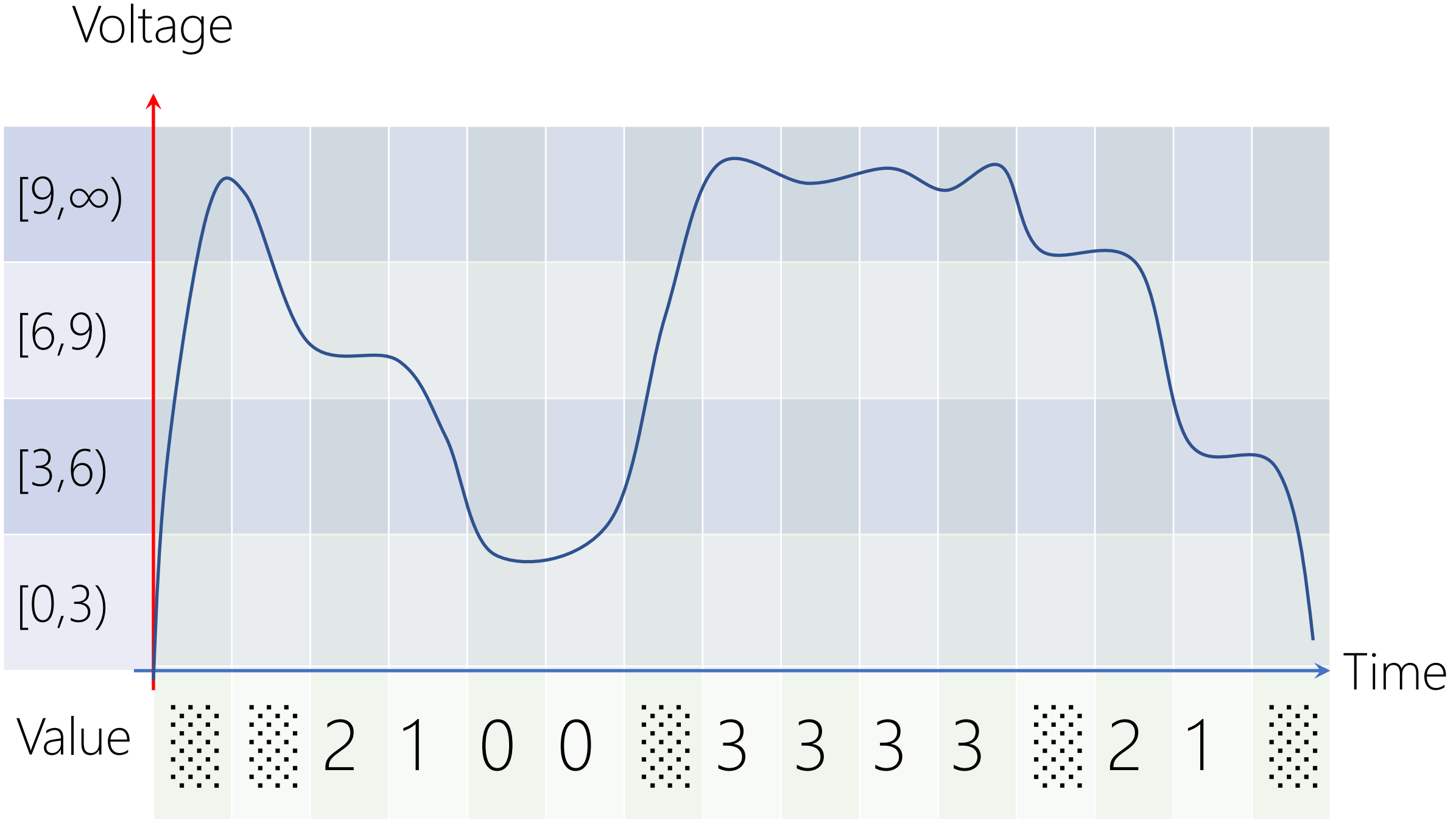

Voltage
Time
[0,1)
[1,2)
[2,3)
[3,4)
[4,5)
[5,6)
[6,7)
[7,8)
[9,10)
[10,11)
[11,12)
[12,13)
[13,14)
[14,15)
[15,16)

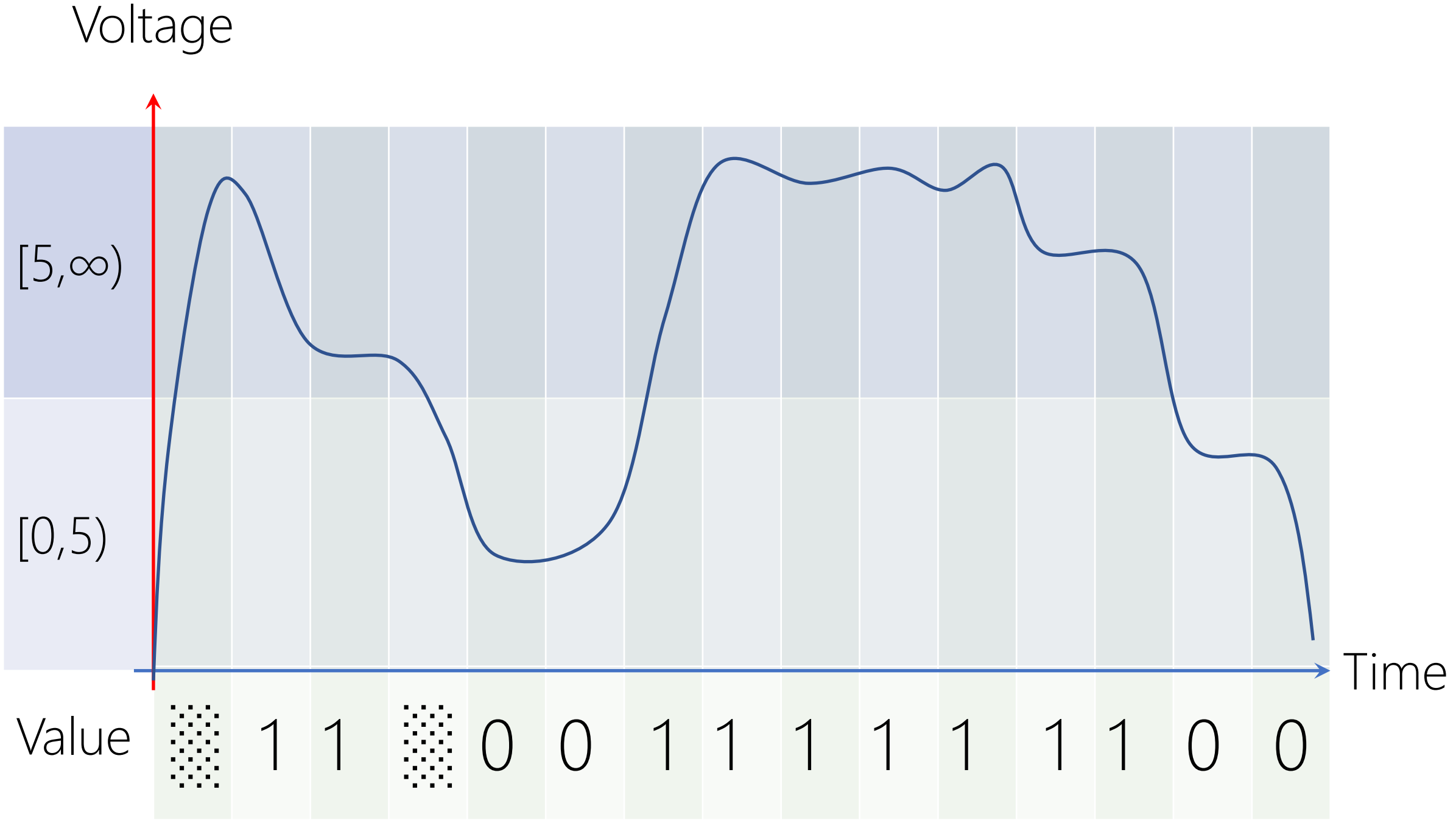

Voltage
[9,10)
[8,9)
[7,8)
[6,7)
[5,6)
[4,5)
[3,4)
[2,3)
[1,2)
[0,1)
Time
[0,1)
[1,2)
[2,3)
[3,4)
[4,5)
[5,6)
[6,7)
[7,8)
[9,10)
[10,11)
[11,12)
[12,13)
[13,14)
[14,15)
[15,16)


Voltage
[9,10)
[8,9)
[7,8)
[6,7)
[5,6)
[4,5)
[3,4)
[2,3)
[1,2)
[0,1)
Time
Value
5
2
2
9
7

GRANULARITY


Voltage
[9,10)
[8,9)
[7,8)
[6,7)
[5,6)
[4,5)
[3,4)
[2,3)
[1,2)
[0,1)
Time
Value
6 5
2 2 2 3
9


Voltage
[9,∞)
[6,9)
[3,6)
[0,3)
Time
Value
2 1 0 0 3 3 3 3 2 1


Voltage
[5,∞)
[0,5)
Time
Value
1 1 0 0 1 1 1 1 1 1 1 0 0

RELIABILITY

Robust to Noise

Fundamentally Hardware/Engineering Problem

TERNARY COMPUTER

https://en.wikipedia.org/wiki/Ternary_computer

Balanced Trinary {−1,0,1}

Entirely from *Wood*!

Thomas Fowler 1840

More History and Etymology → https://en.wikipedia.org/wiki/Computer

DECIMAL COMPUTER

https://en.wikipedia.org/wiki/Decimal_computer

They are not actually base-10! We'll cover them later.

TRUE VS. FALSE

George Boole (/bu:l/)

Mathematician
Philosopher
Logician

The Laws of Thought (1854)

Boolean Algebra!

Claude Elwood Shannon

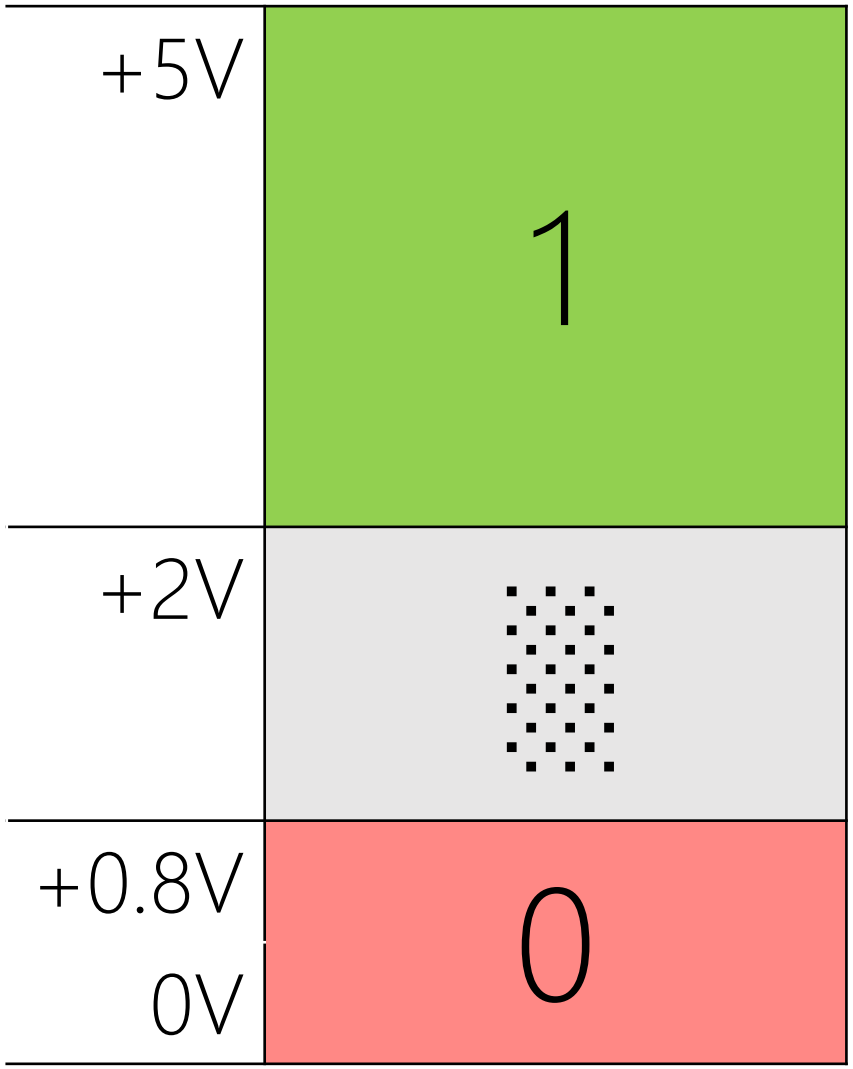
Mathematician
Electrical Engineer
Cryptographer

M.Sc. Thesis (1937)

A Symbolic Analysis of Relay and Switching Circuits
21 years old!

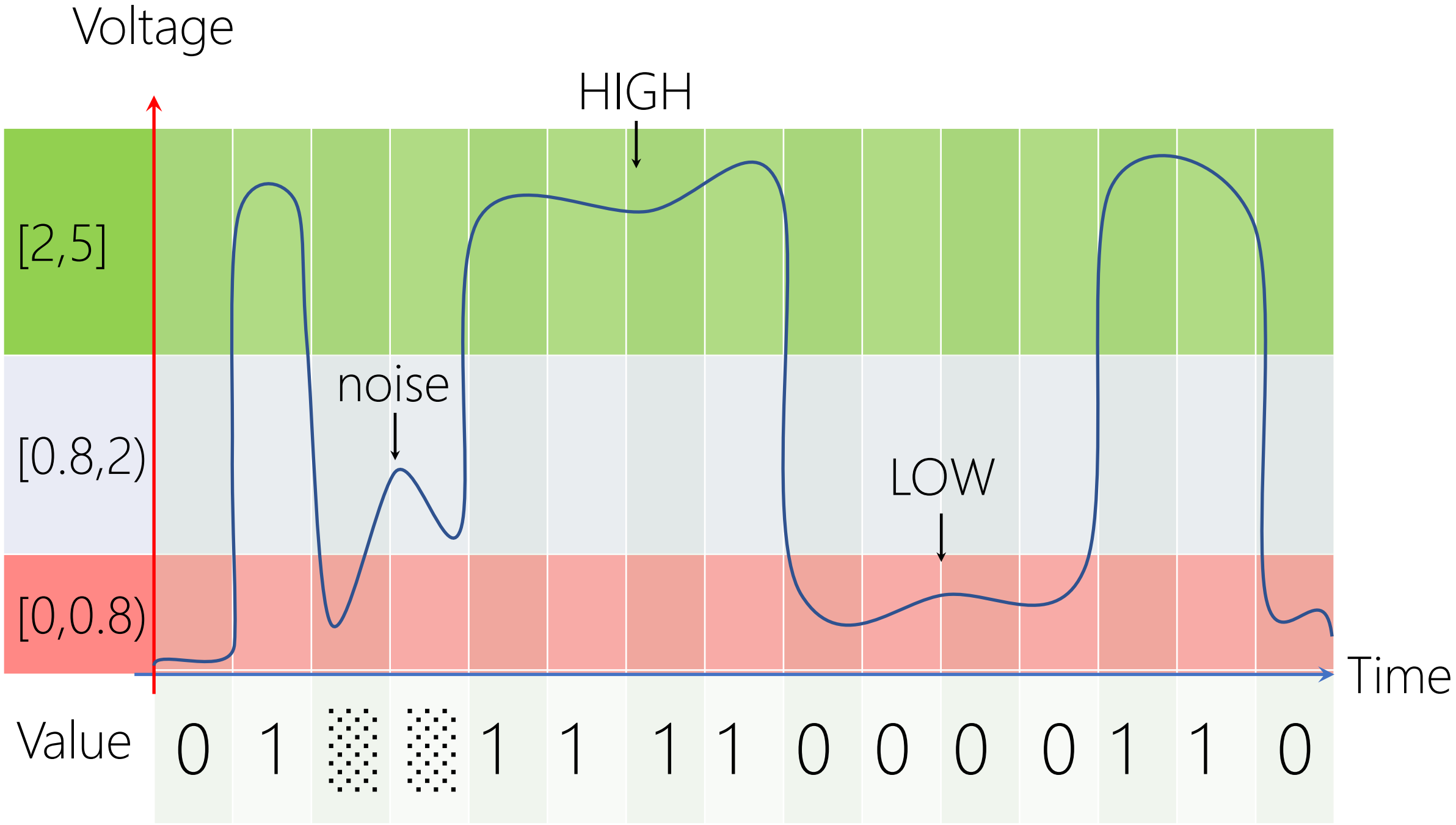
Switching Algebra!

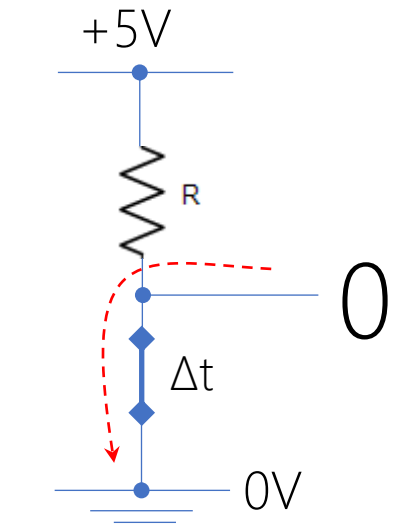
BINARY COMPUTER

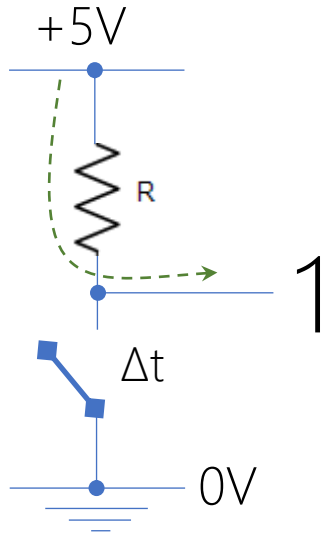


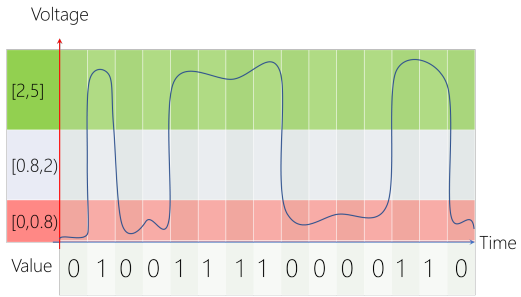
POSITIVE LOGIC

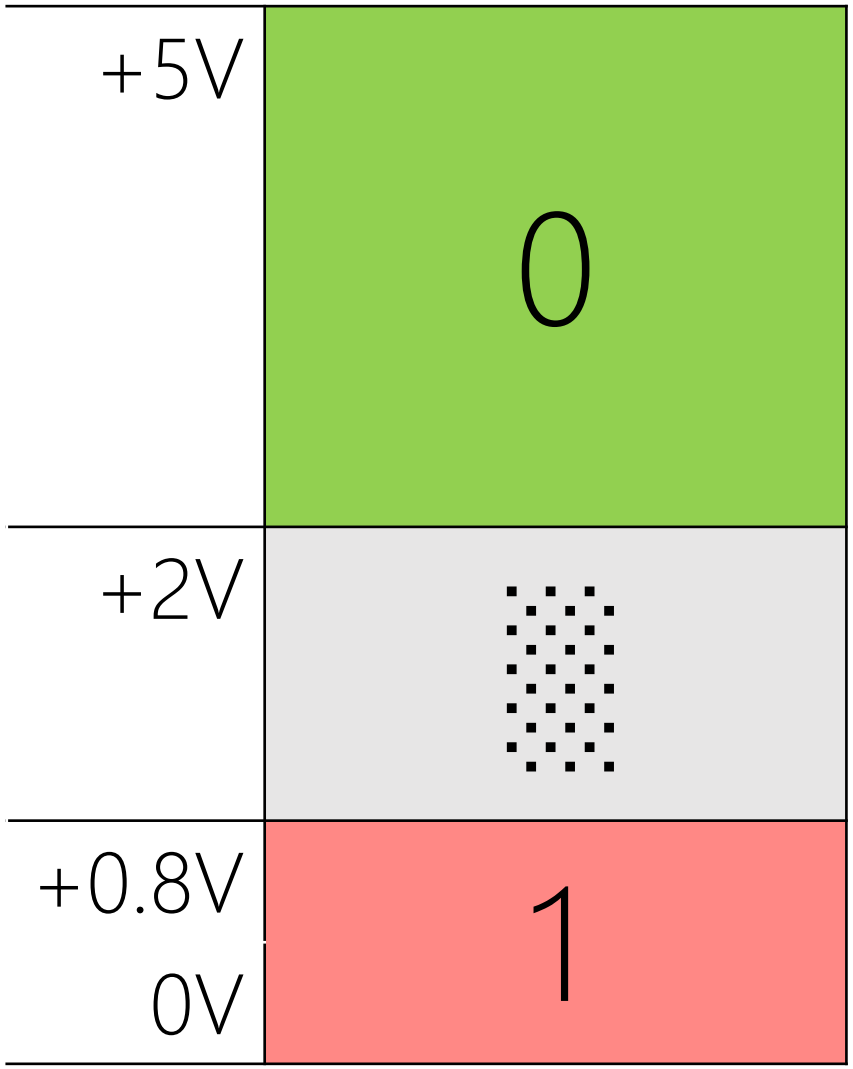
https://www.electronics-tutorials.ws/logic/logic_1.html


Voltage
HIGH
[2,5]
noise
[0.8,2)
LOW
[0,0.8)
Time
Value
0 1 1 1 1 0 0 0 0 1 1 0


+5V
R
0
Δt
0V
Earth (Ground)

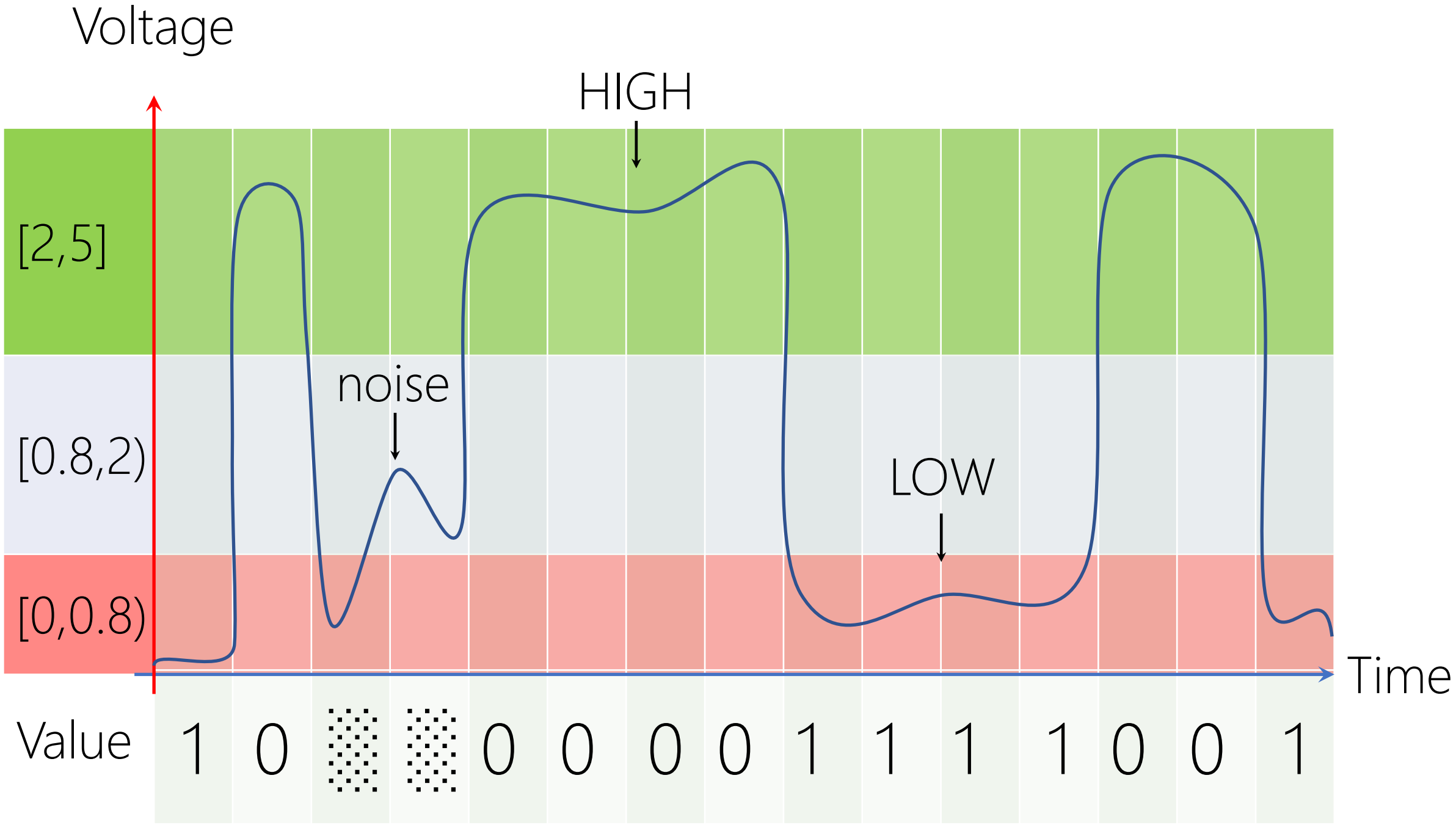

+5V
R
1
Δt
0V
Earth (Ground)

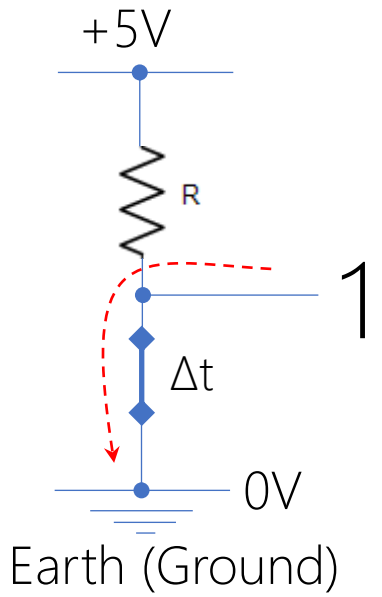

Voltage
[2,5]
[0.8,2)
[0,0.8)
Time
Value
0 1 0 0 1 1 1 1 0 0 0 0 1 1 0

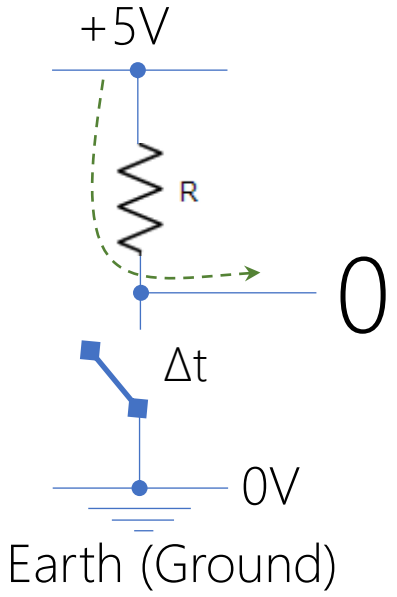


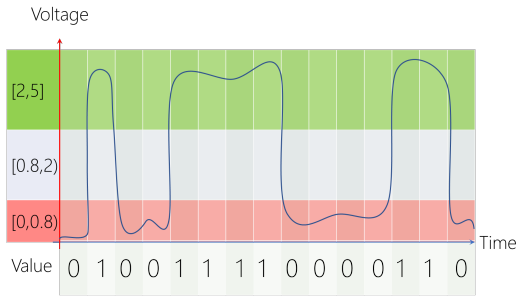
NEGATIVE LOGIC

https://www.electronics-tutorials.ws/logic/logic_1.html


Voltage
HIGH
[2,5]
noise
[0.8,2)
LOW
[0,0.8)
Time
Value
1 0 0 0 0 1 1 1 1 0 0 1


+5V
R
1
Δt
0V
Earth (Ground)


+5V
R
0
Δt
0V
Earth (Ground)


Voltage
[2,5]
[0.8,2)
[0,0.8)
Time
Value
0 1 0 0 1 1 1 1 0 0 0 0 0 1 1 0

DESIGN COMPUTER

Positive Logic
Button-Up Approach

DESIGN COMPUTER

Positive Logic
Button-Up Approach

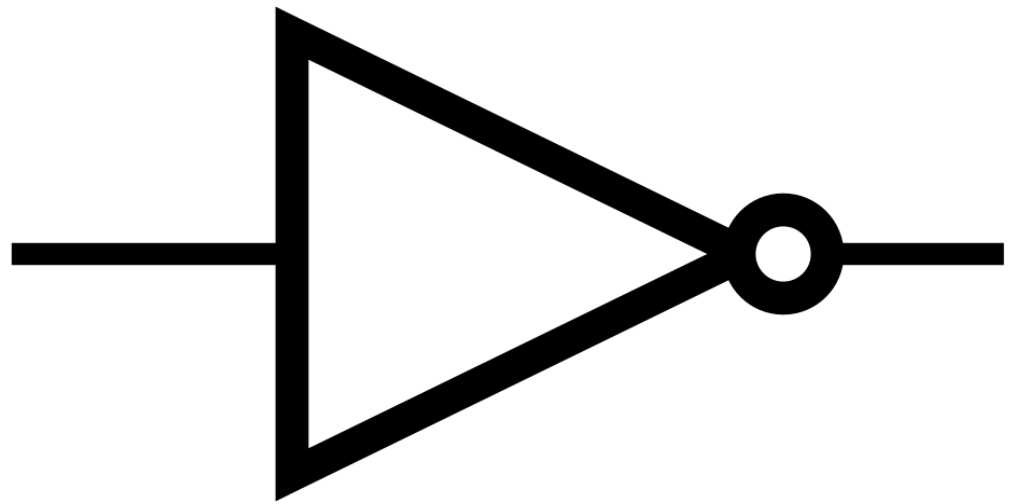
Finding simpler, but equivalent, computers reduces the overall cost!
Rely primarily on mathematical methods in Boolean algebra!

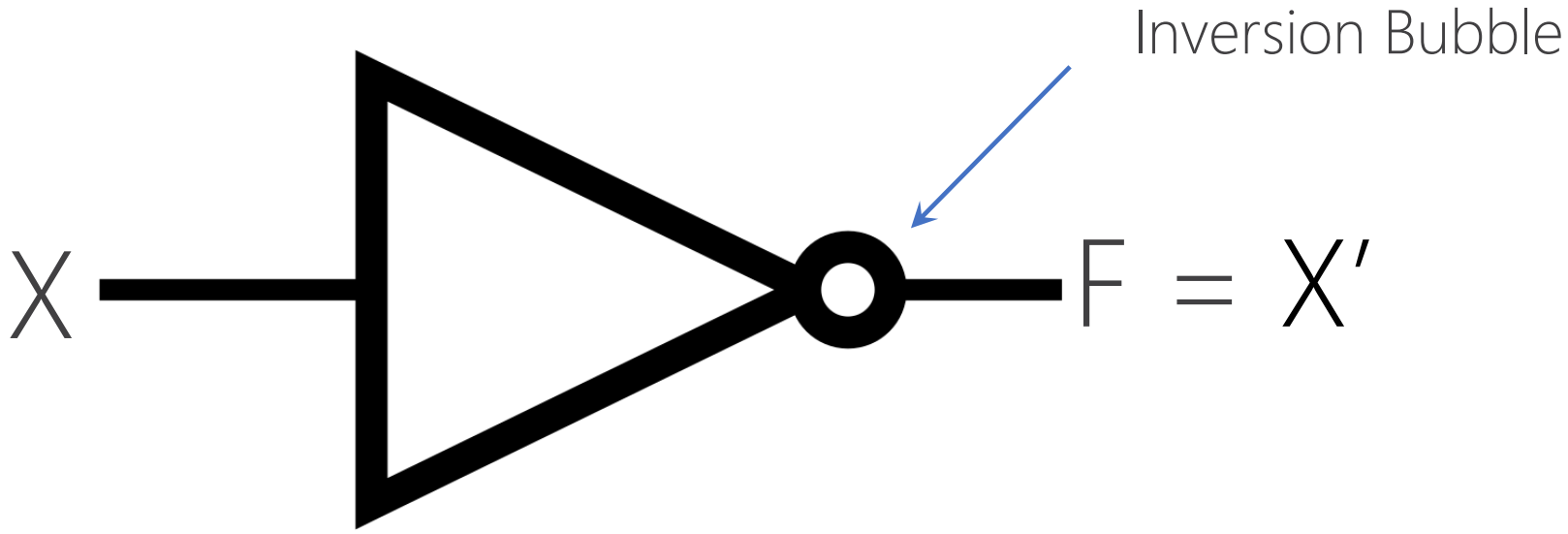
BUILD COMPUTER

Electrical and Computer Engineering

LOGIC GATES

BOOLEAN GATES

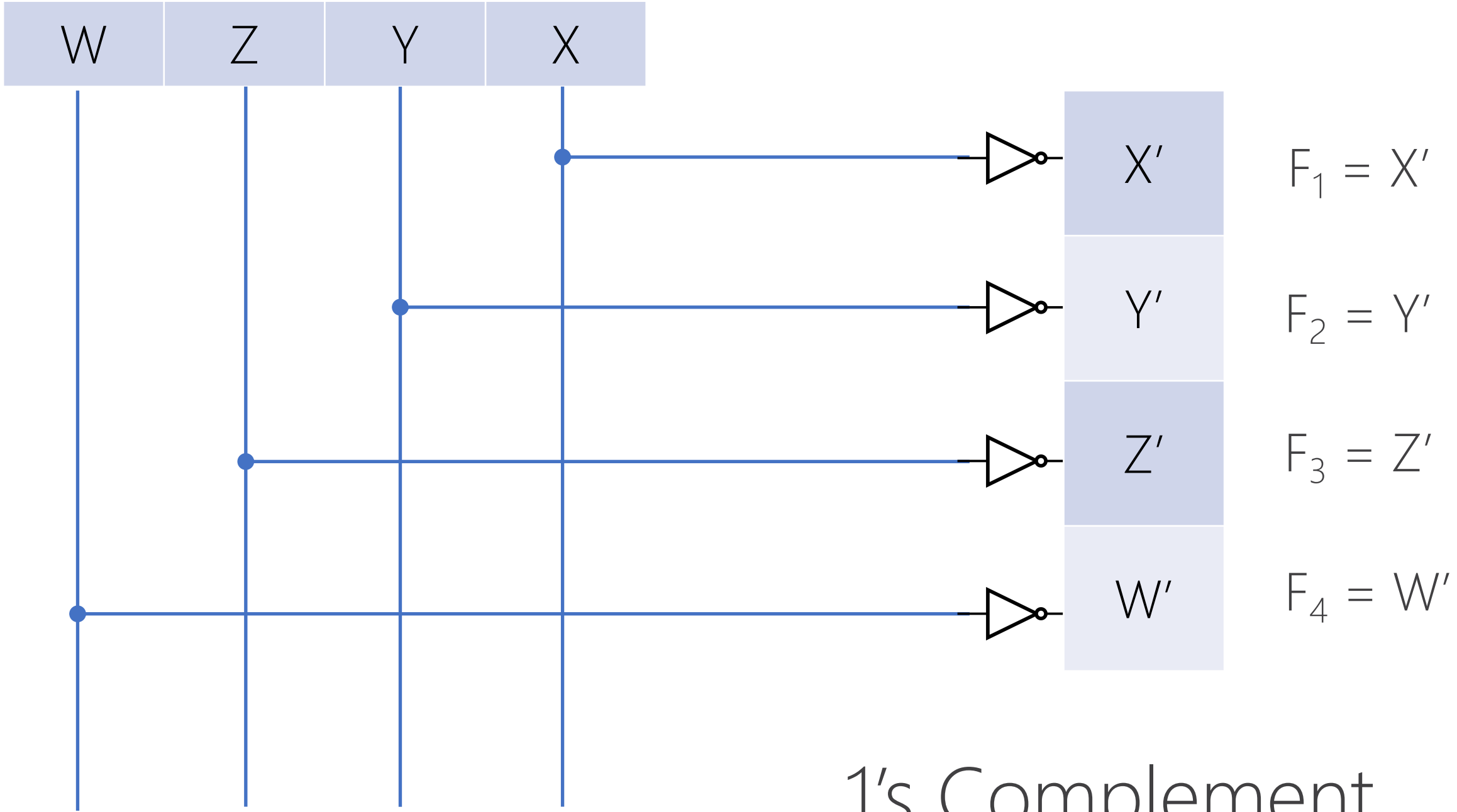




X	NOT X	Invertor X	X′	$\overline{X}$
0		1		
1		0		

Boolean Expression/Function: F = X′

> *inverse of X gives F* <



1's Complement

BINARY VARIABLE

aka. Boolean variable

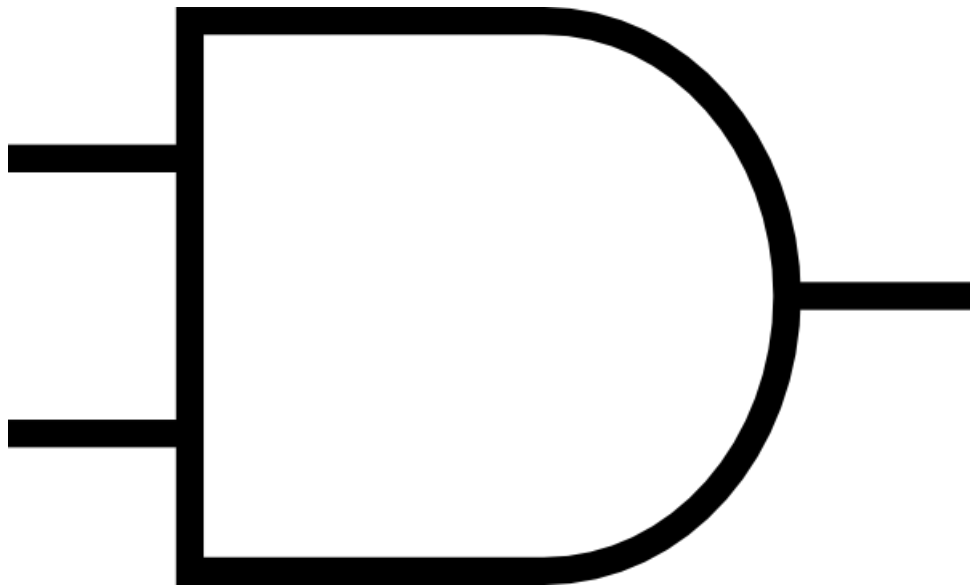
A variable that can have a value in {0,1}

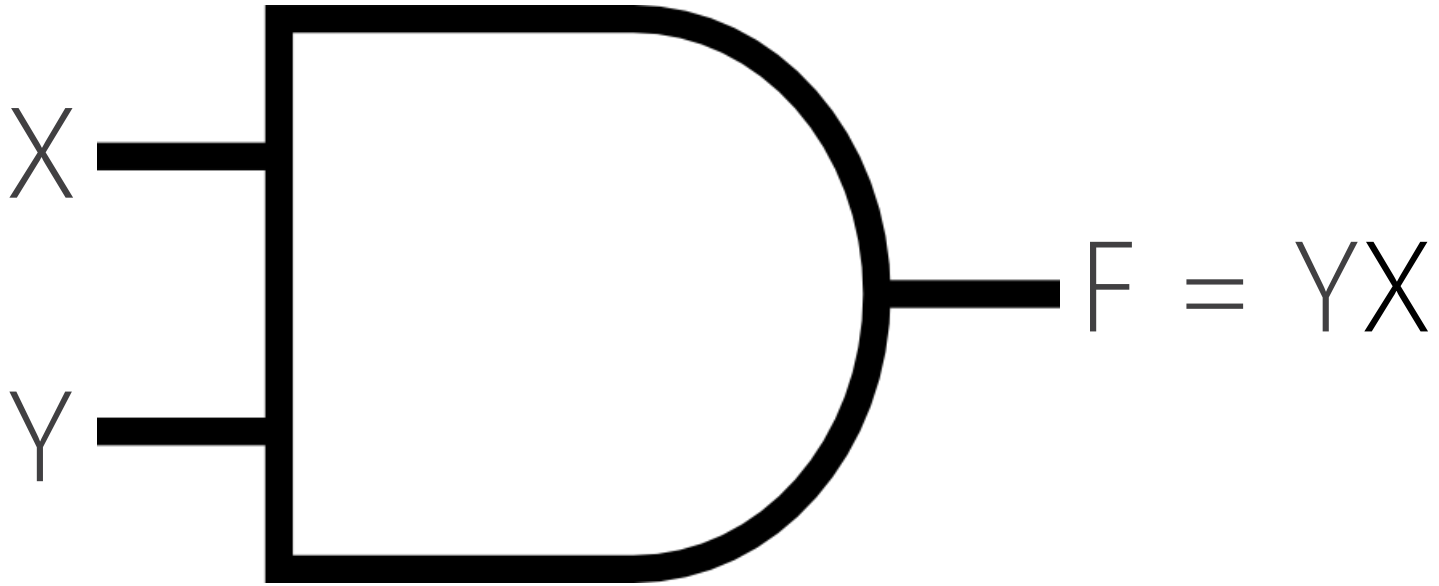
X, Y, Z, W, …

BOOLEAN FUNCTION

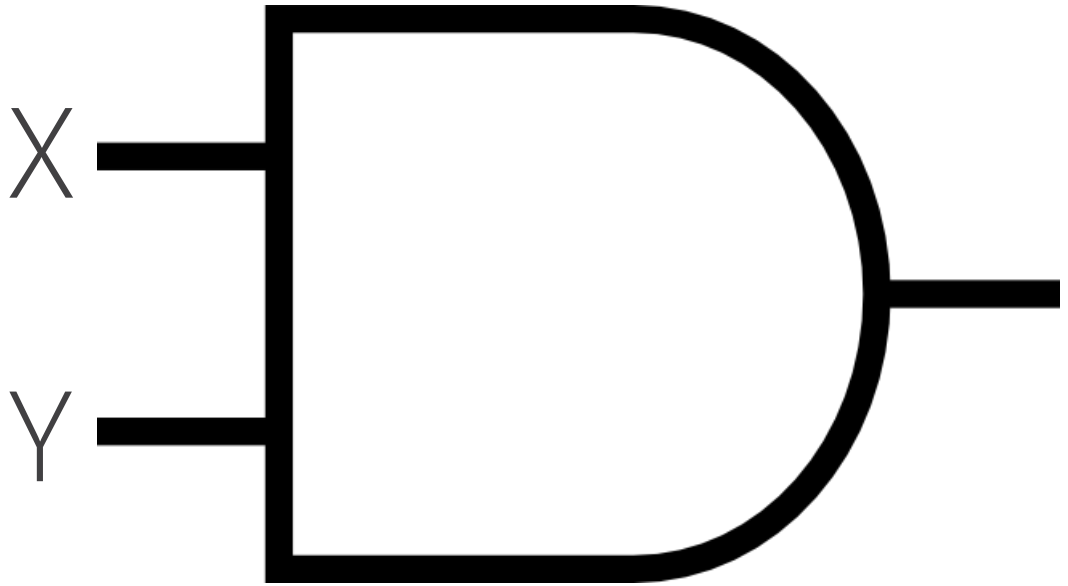
aka. Boolean Expression

A function that accept Boolean variable(s) and output a value in {0,1}, e.g., F(X) = X’





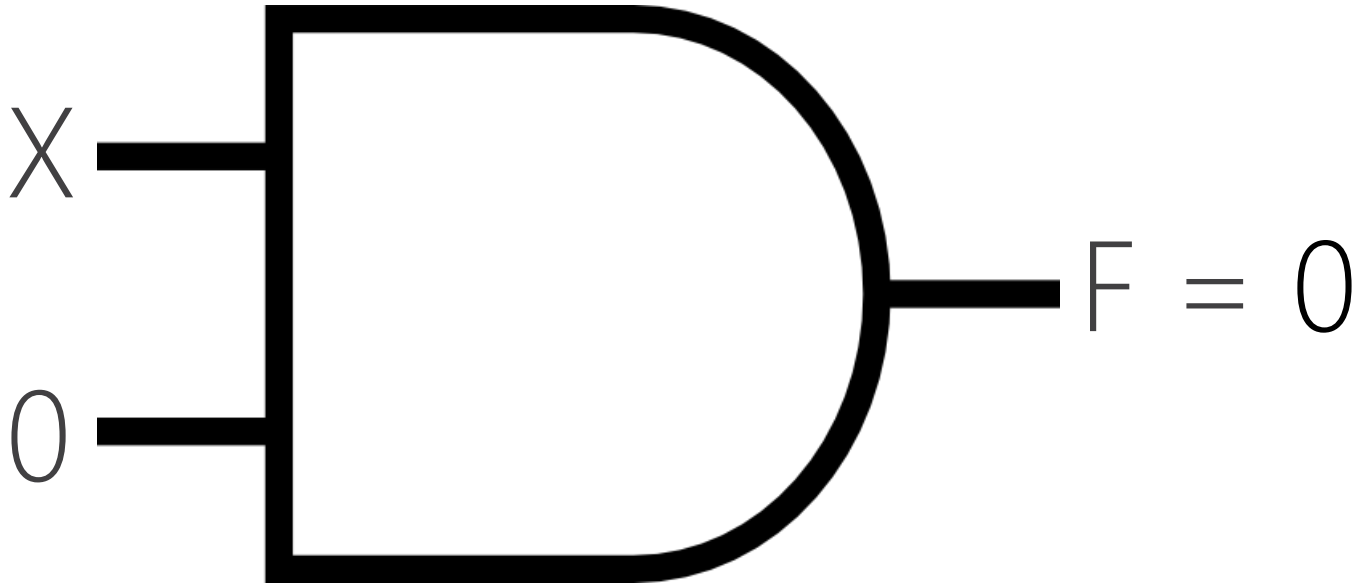
Y	X	X AND Y	Y · X	Y * X
0	0		0	
0	1		0	
1	0		0	
1	1		1	



F = YX = XY

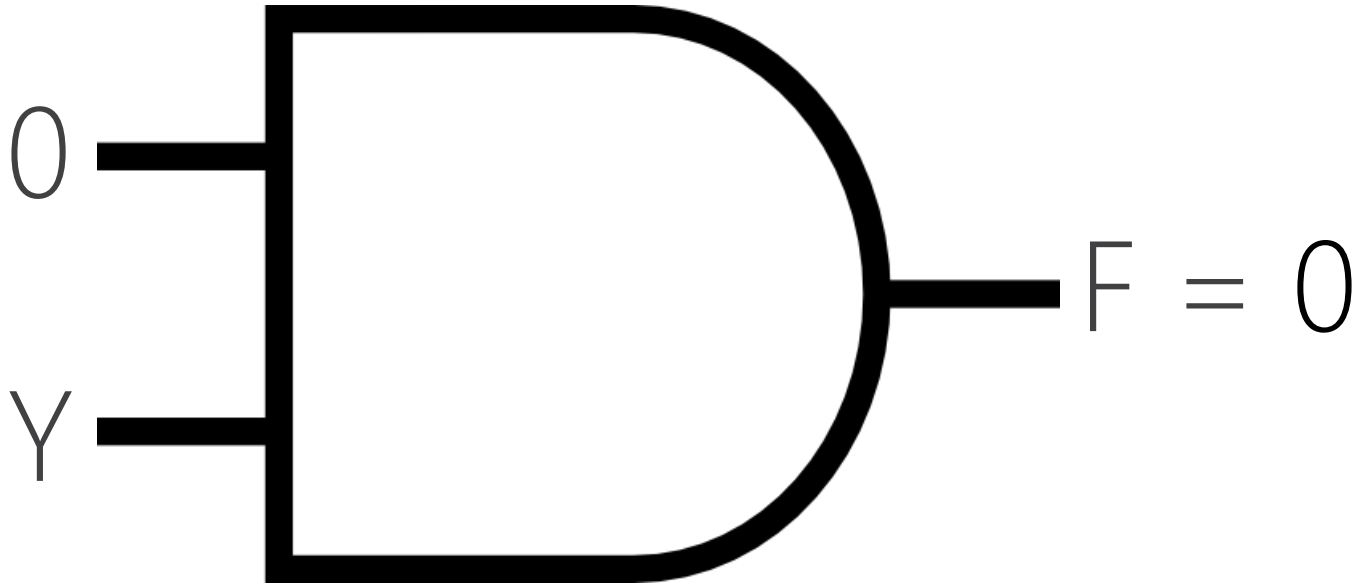
Commutative

X	Y	Y AND X	X · Y	X * Y
0	0		0	
0	1		0	
1	0		0	
1	1		1	



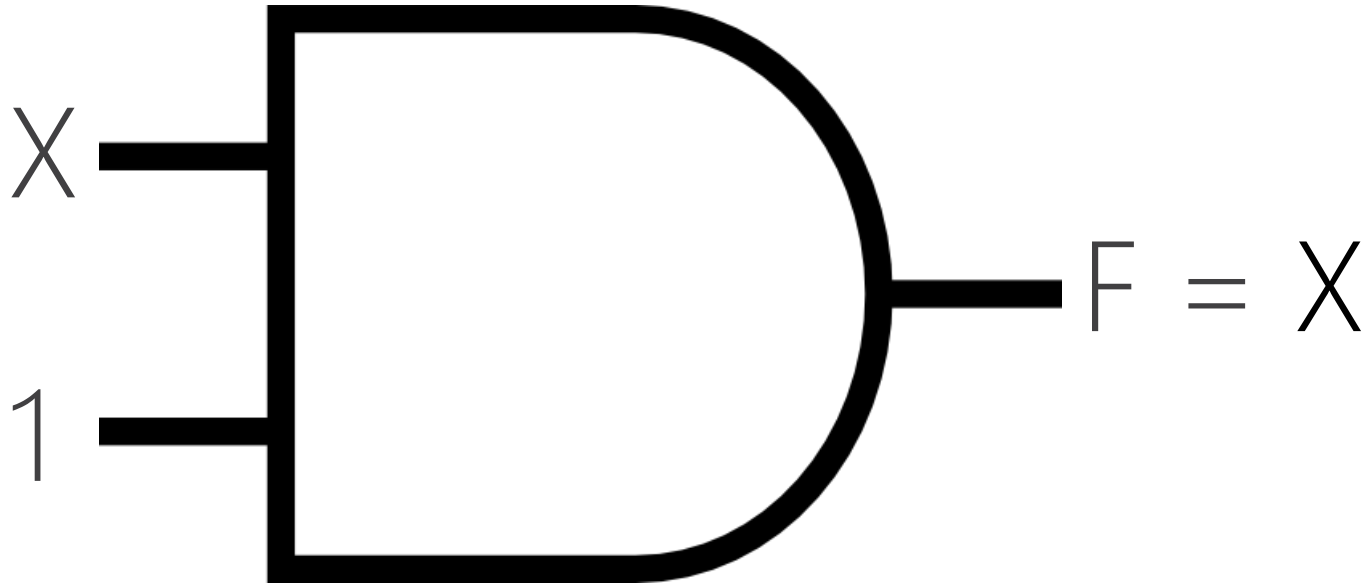
Y	X	YX
0	1	0
0	0	0

$$F = X0 = 0$$



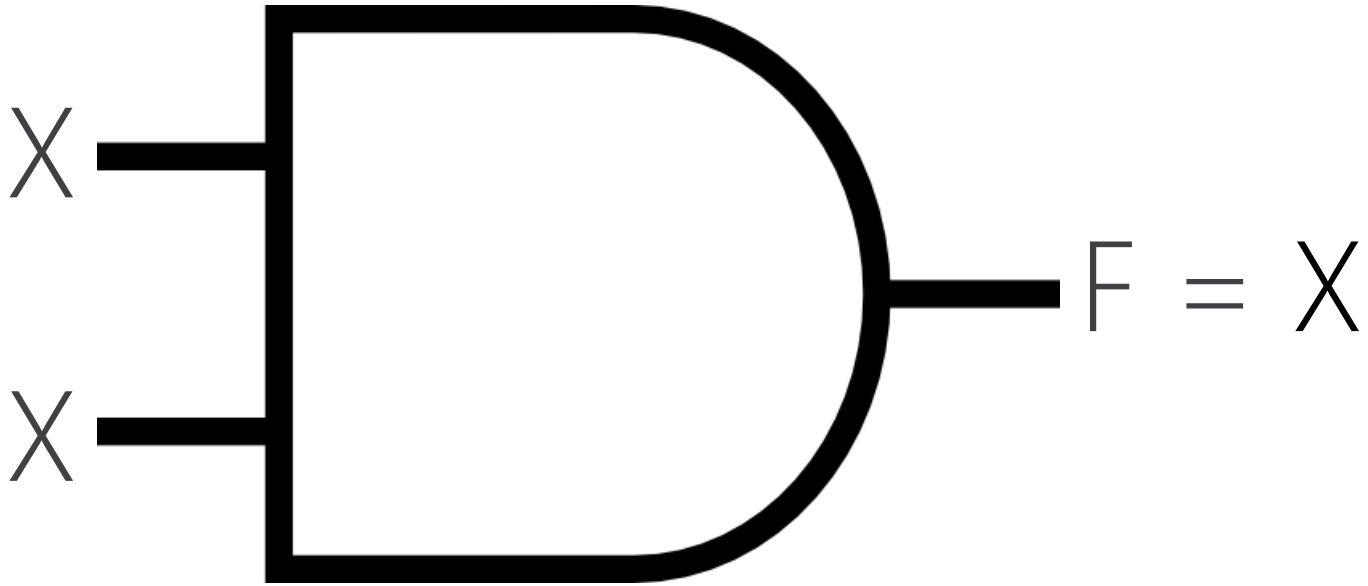
Y	X	YX
0	0	0
1	0	0

$$F = 0Y = 0$$



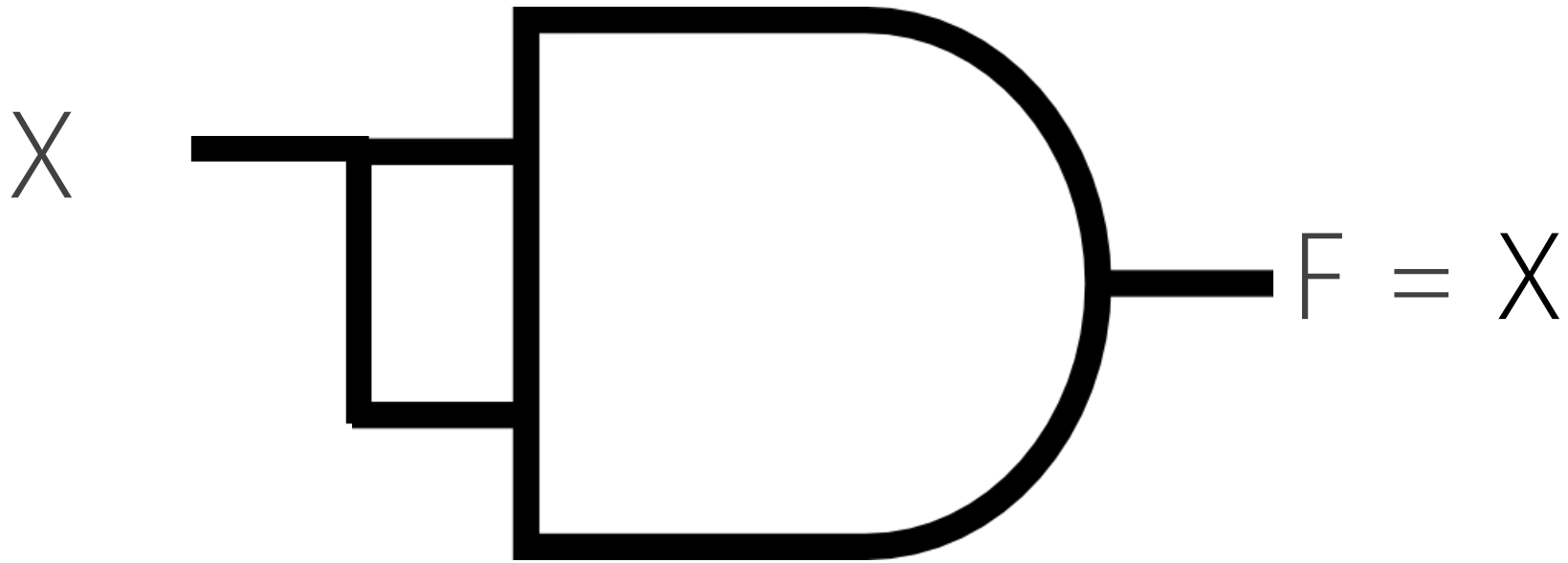
Y	X	YX
1	0	0
1	1	1

$$F = X1 = 1111X1111 = X$$



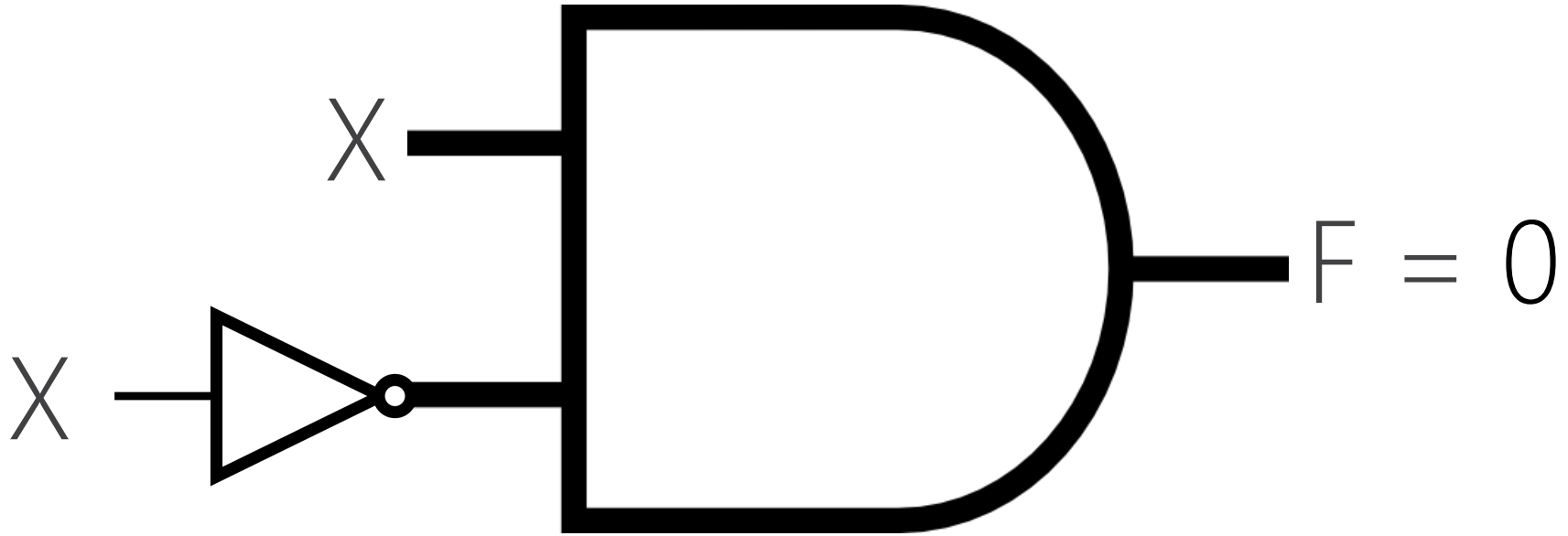
X	X	XX
0	0	0
1	1	1

$$F = XX = X$$



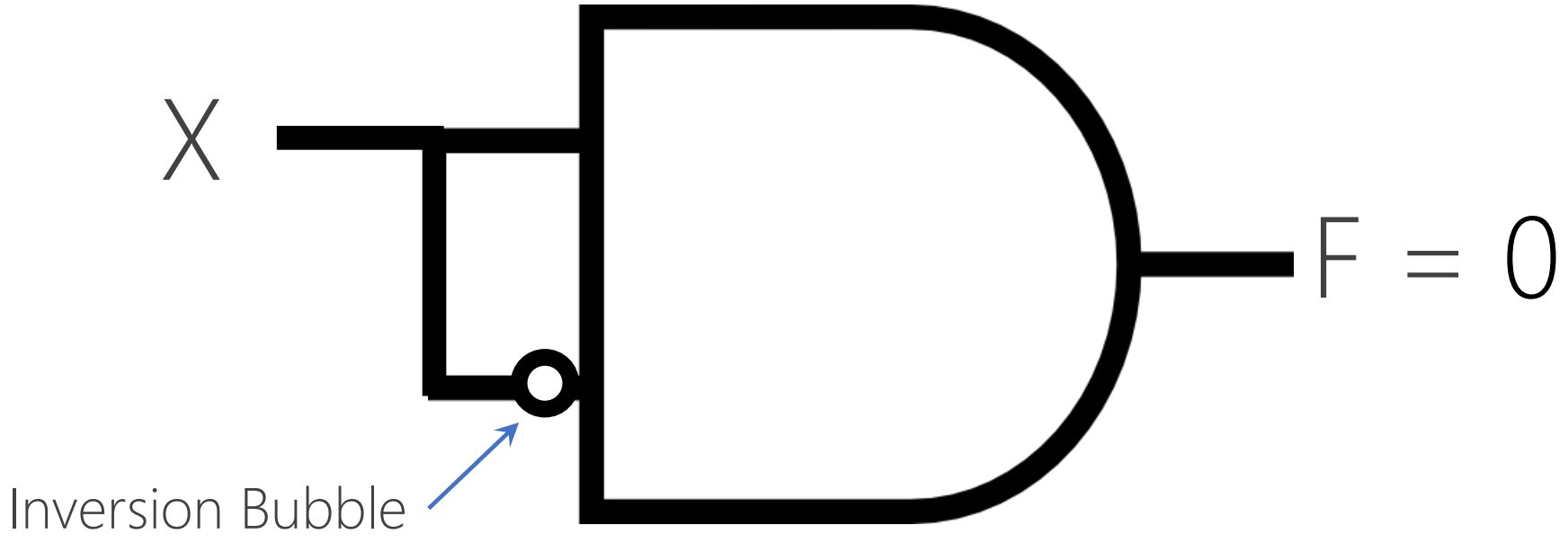
X	X	XX
0	0	0
1	1	1

$$F = XX = X$$



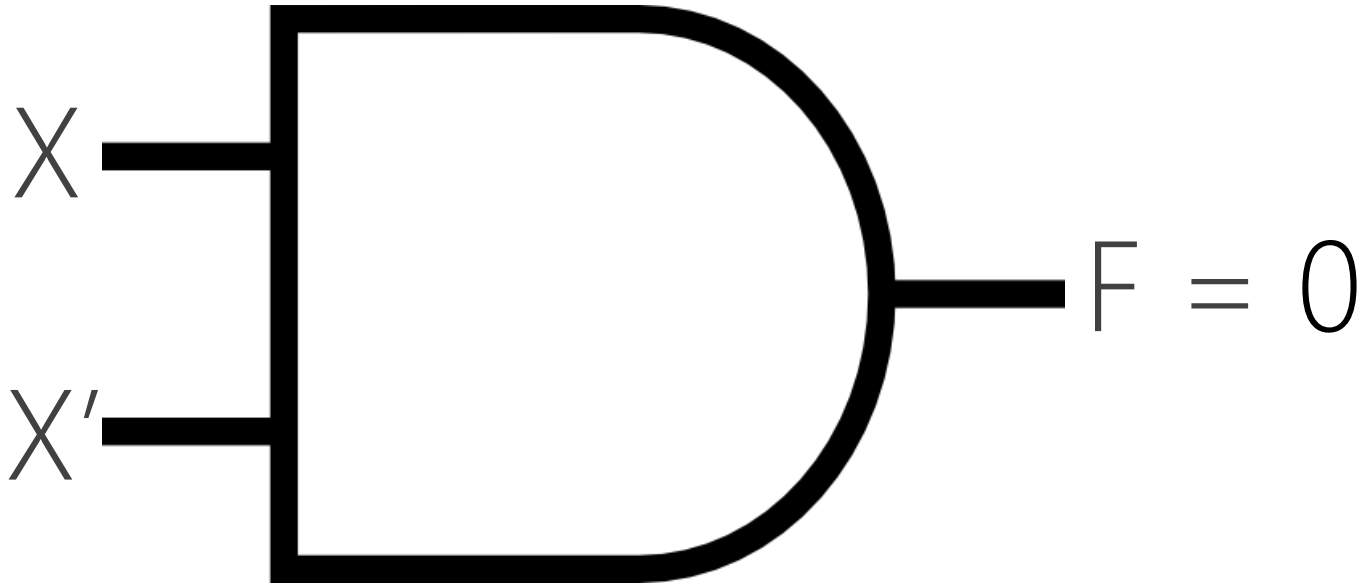
X′	X	X′X
1	0	0
0	1	0

$$F = XX' = 0$$



X′	X	X′X
1	0	0
0	1	0

$$F = XX' = 0$$

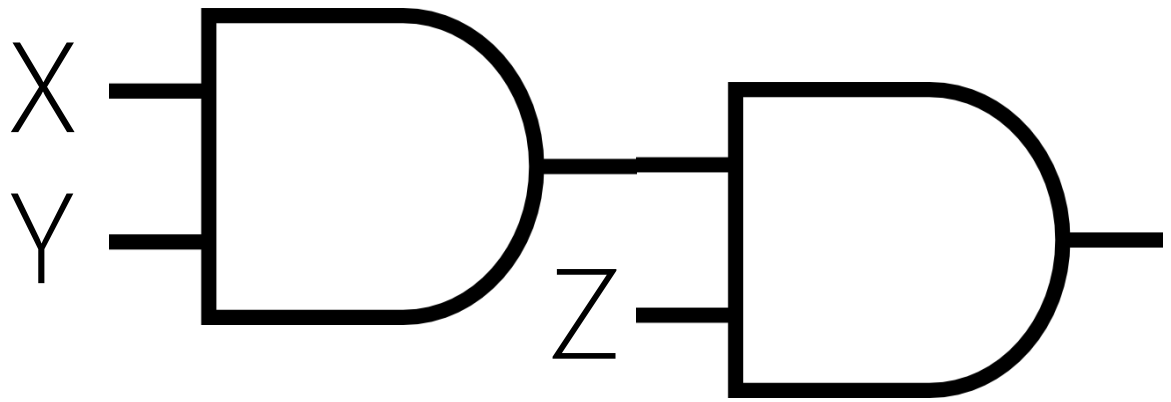


X′	X	X′X
1	0	0
0	1	0

$$F = XX' = 0$$

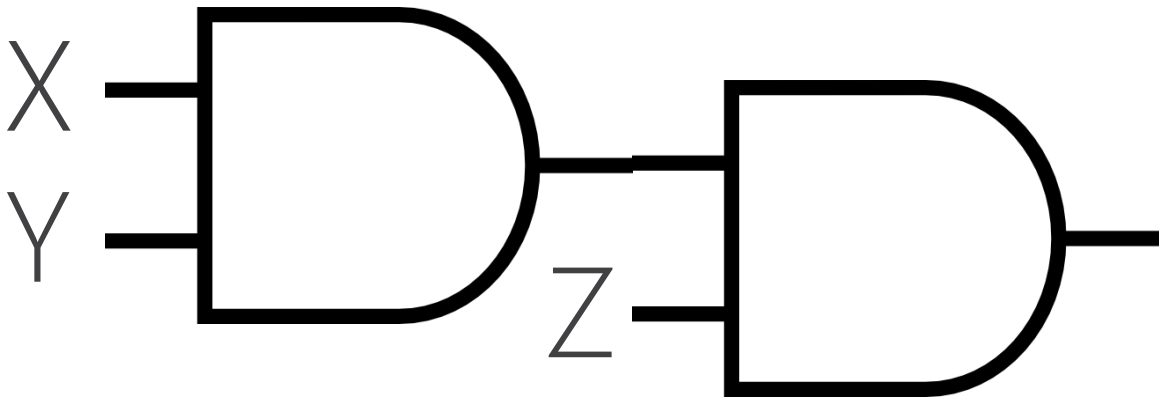
3-INPUT AND

Z	Y	X	ZYX
0	0	0	0
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	1



$$F = Z(YX)$$

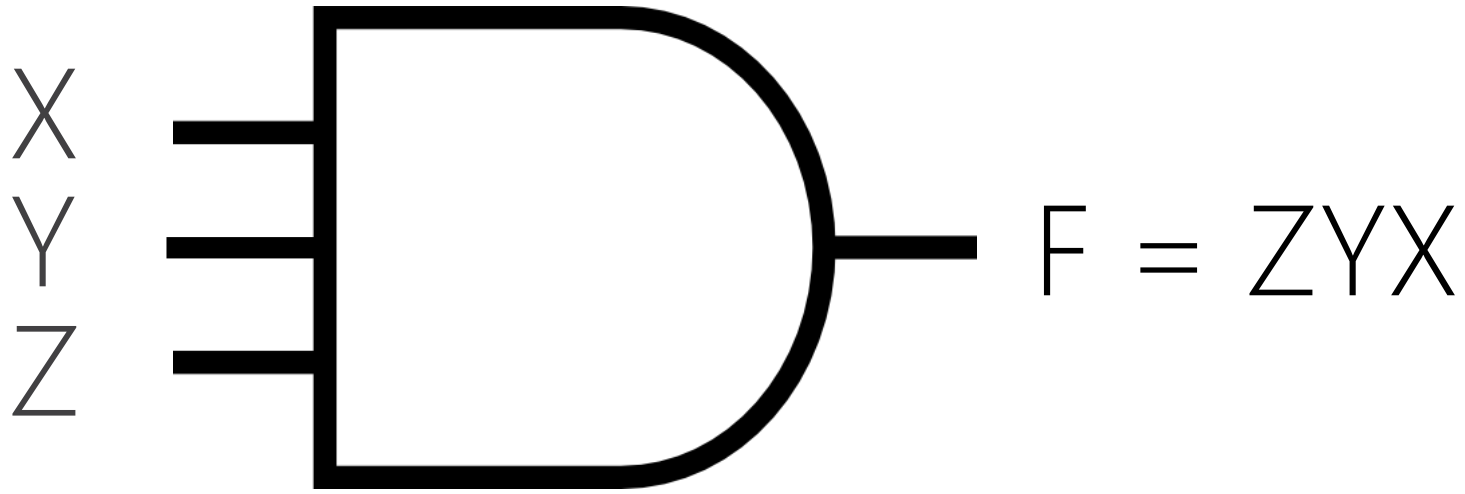
Z	Y	X	Z(YX)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



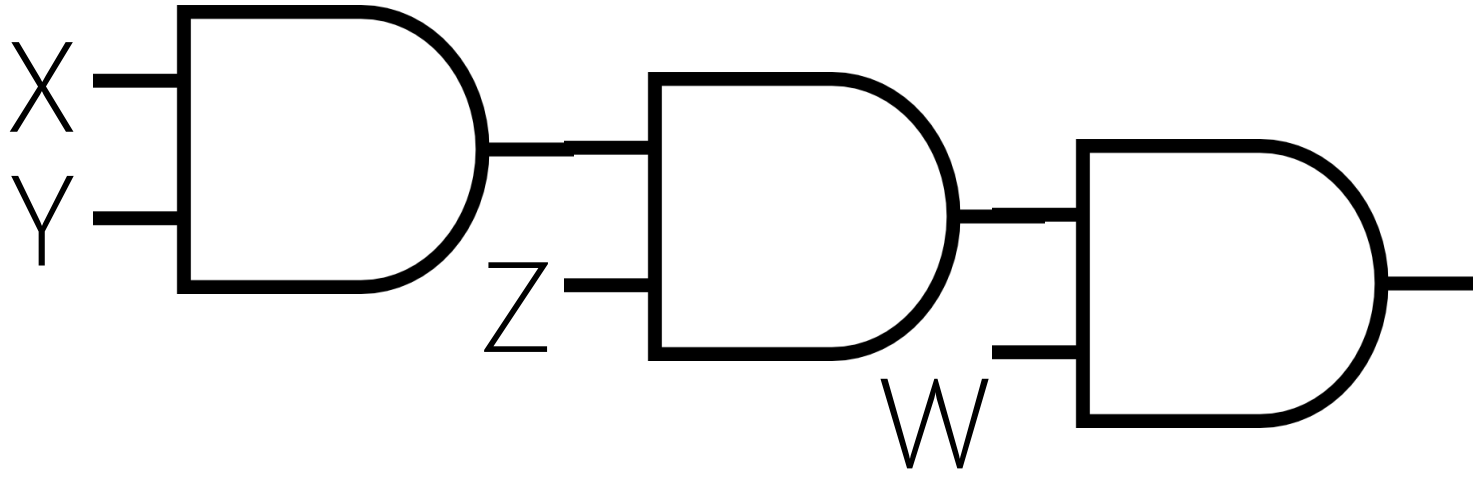
$$F = Z(YX) = Z(XY) = (ZX)Y = (XZ)Y = XZY$$

Associative

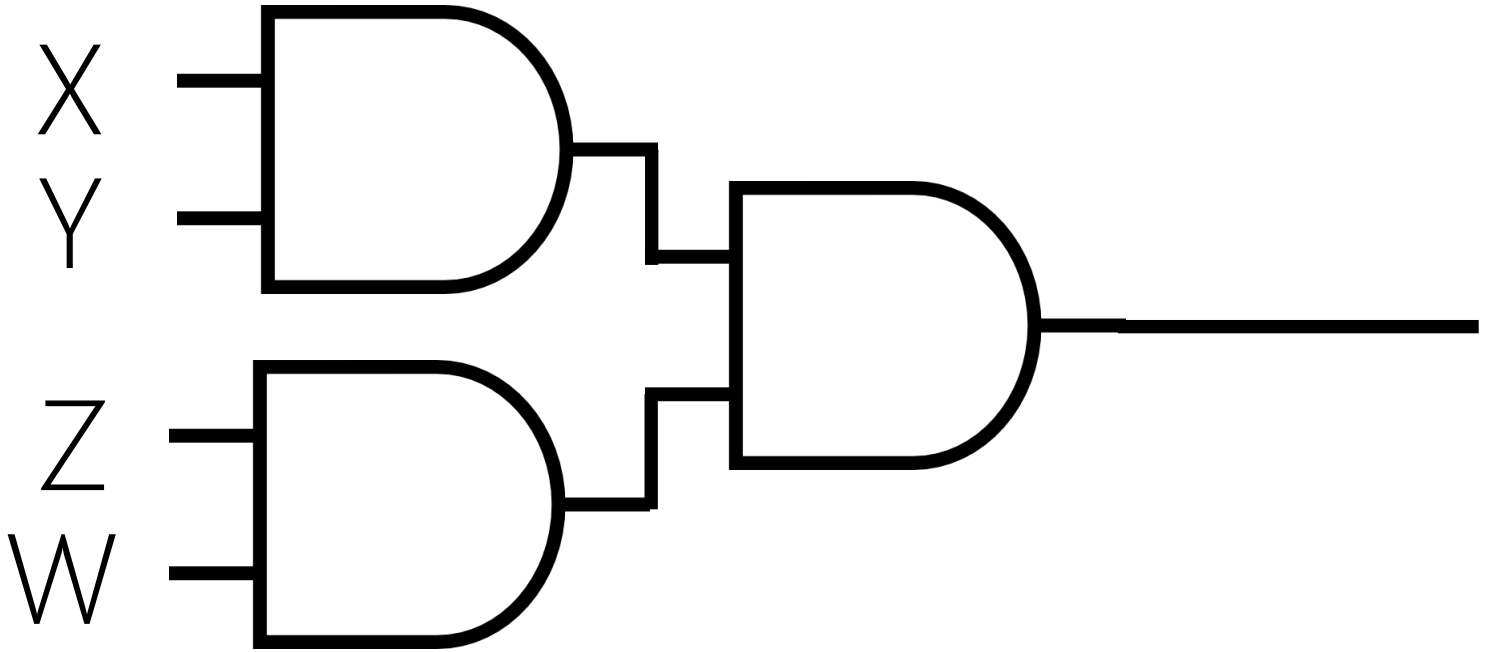
Z	Y	X	Z(YX)	Z(XY)	(ZX)Y	XZY
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1



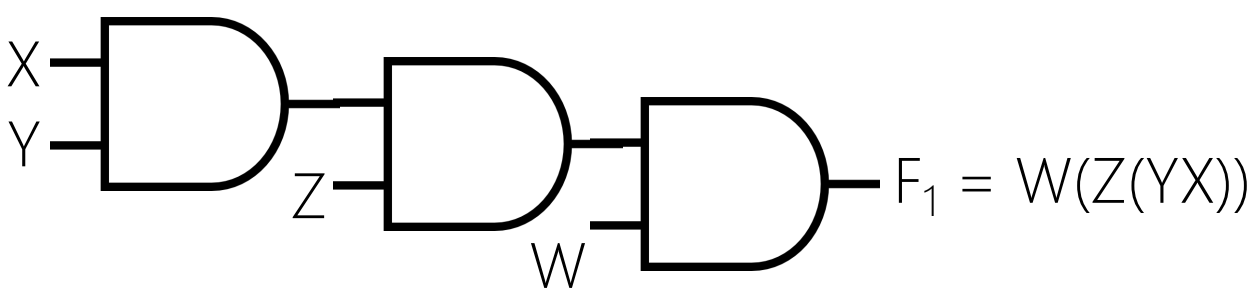
Z	Y	X	ZYX
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

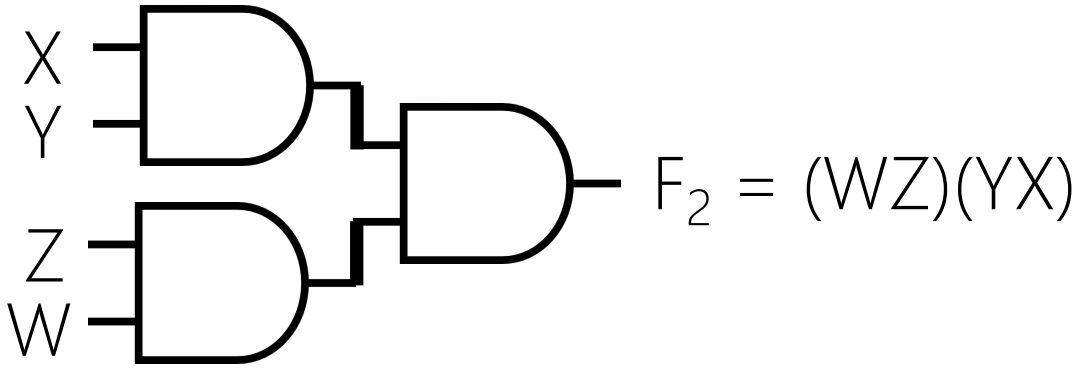


$F_1 = W(Z(YX))$

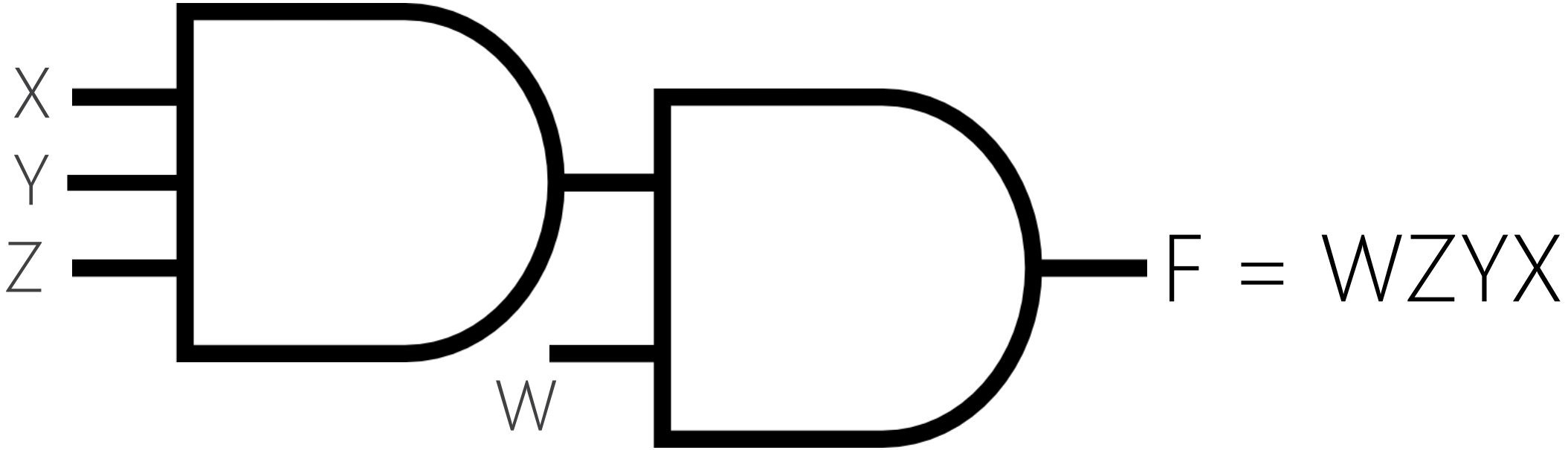


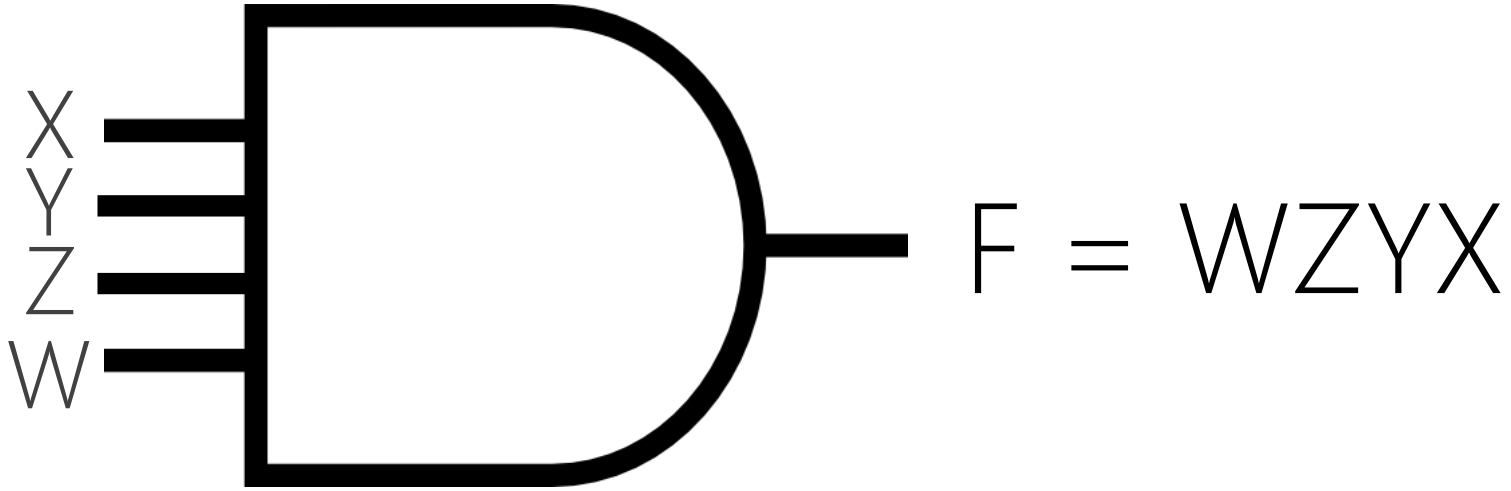
$F_2 = (WZ)(YX)$

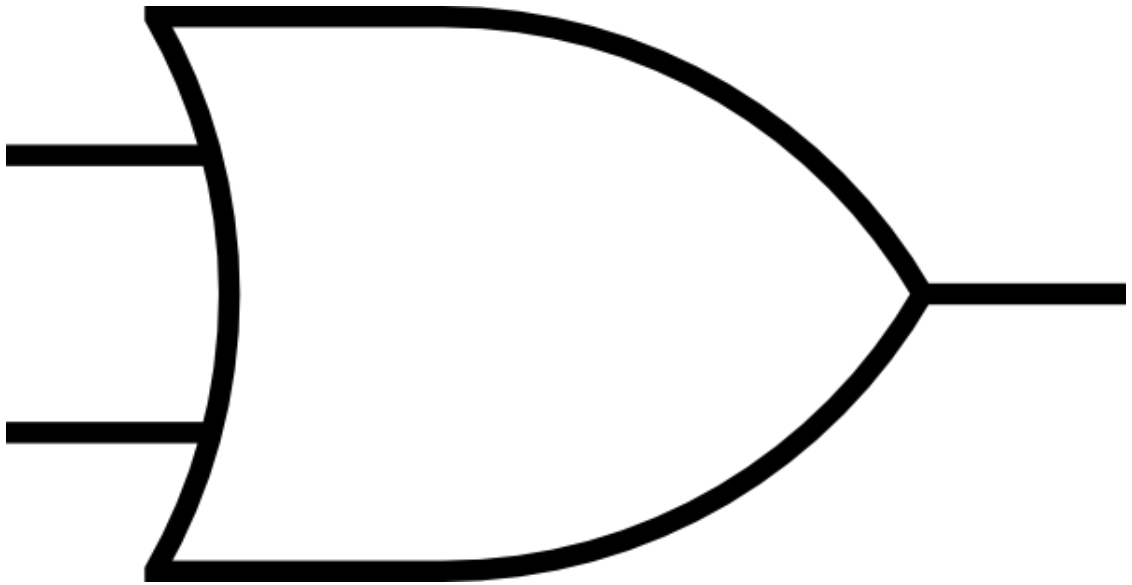




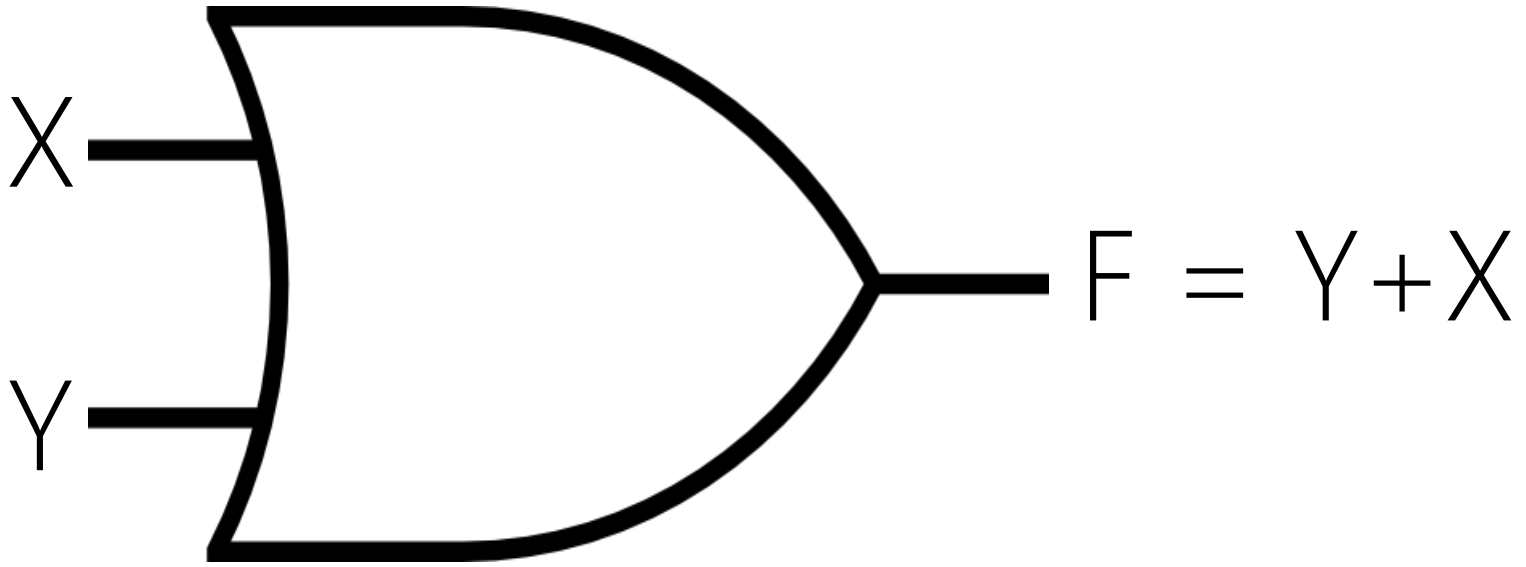
F = WZYX	F_1	F_2
Effective (True)	Yes	Yes
Efficient (Fast)	Hmm, 3 levels, No!	Yes! 2 levels
Min. Cost	3 gates, Yes	3 gates, Yes


X
Y
Z
W
F = WZYX

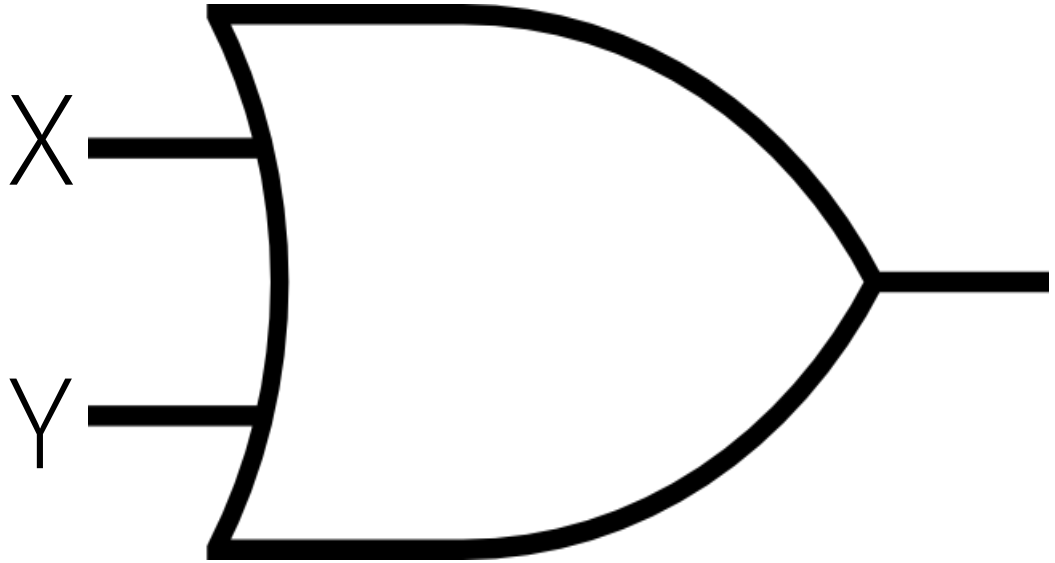

X
Y
Z
W
F = WZYX



OR



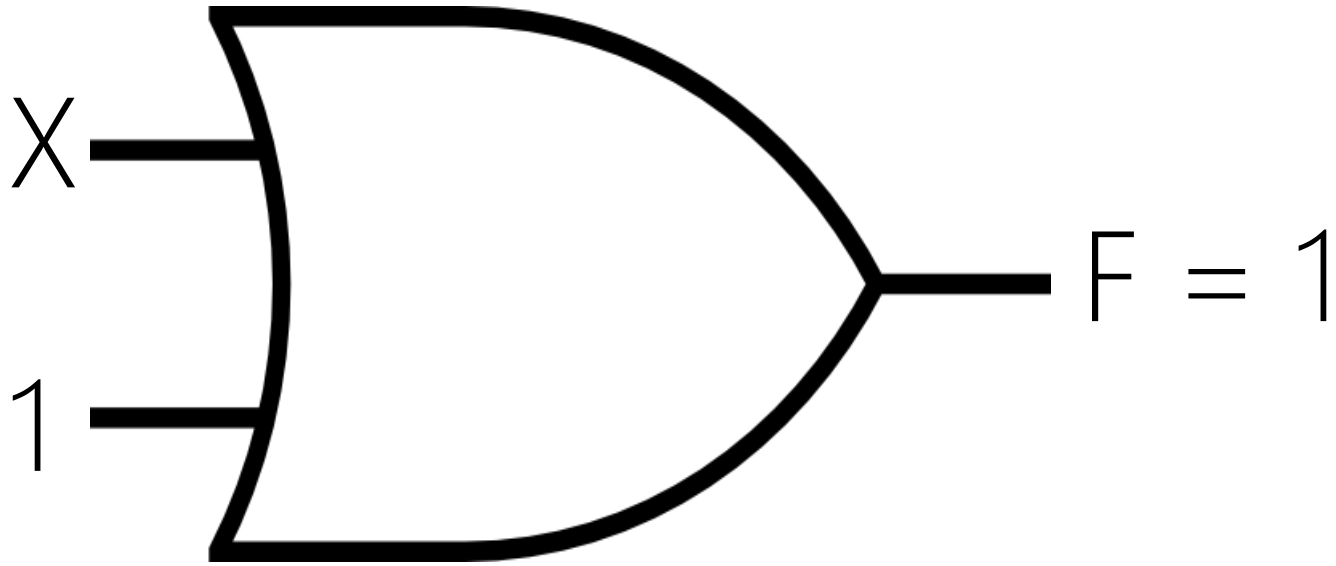
Y	X	Y OR X	Y+X
0	0	0	
0	1	1	
1	0	1	
1	1	1	



$F = Y + X = X + Y$

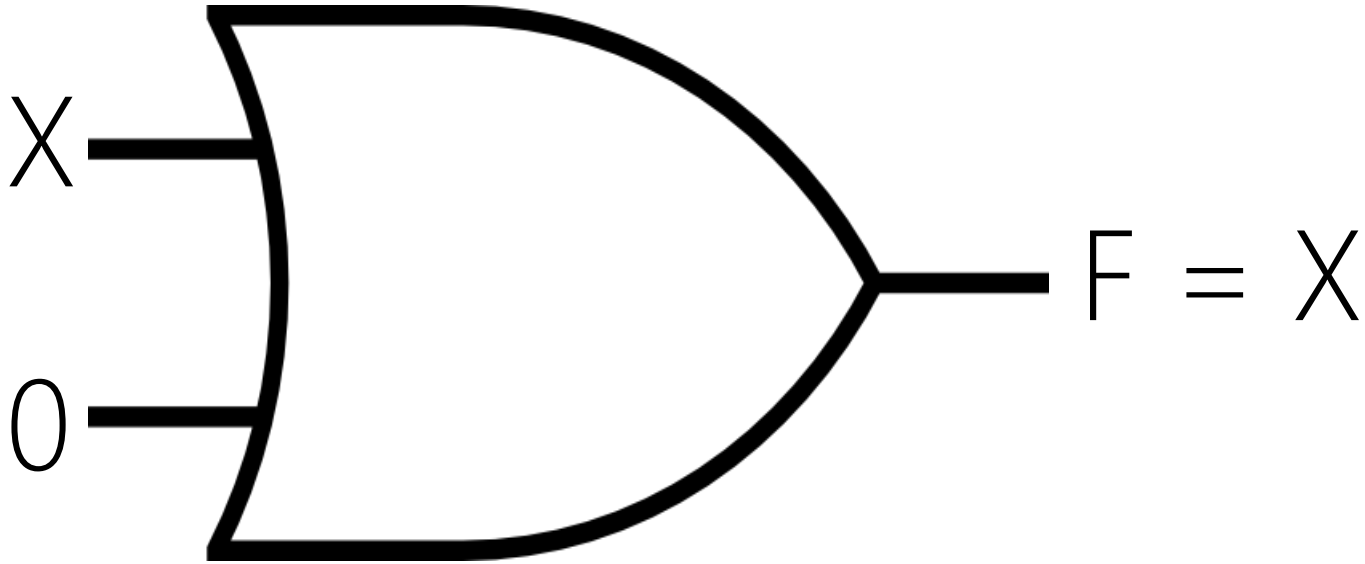
Commutative

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



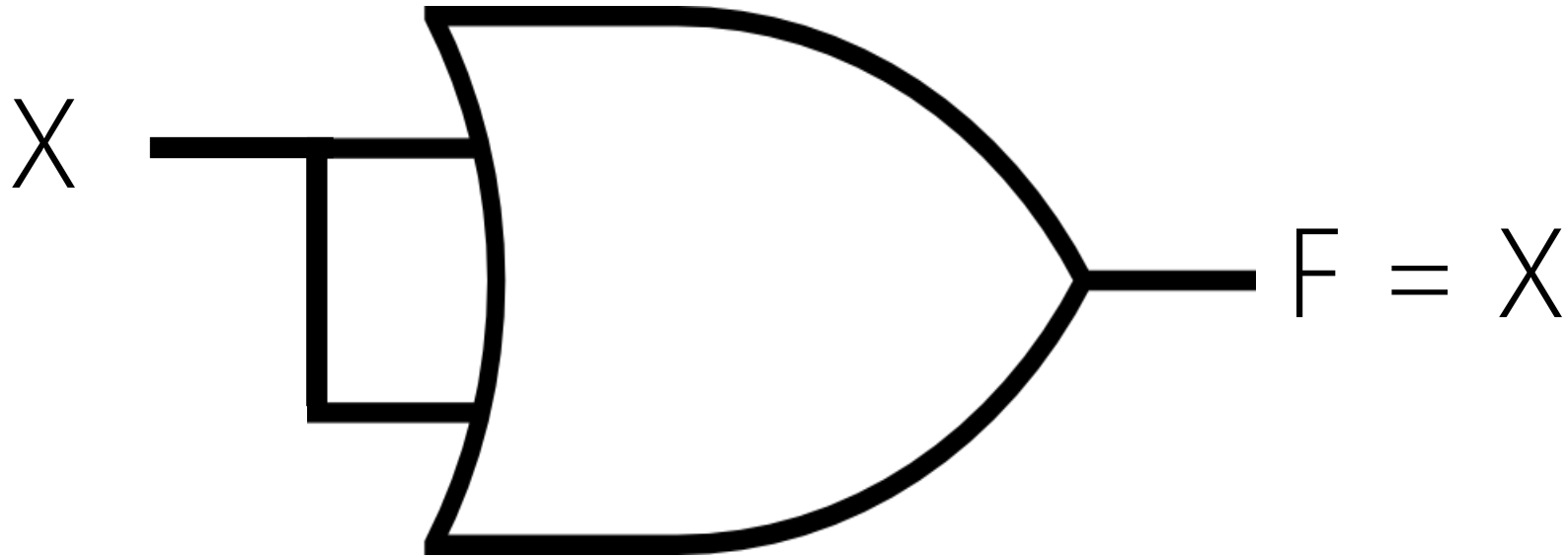
Y	X	Y+X
1	0	1
1	1	1

$$F = X+1 = 1$$



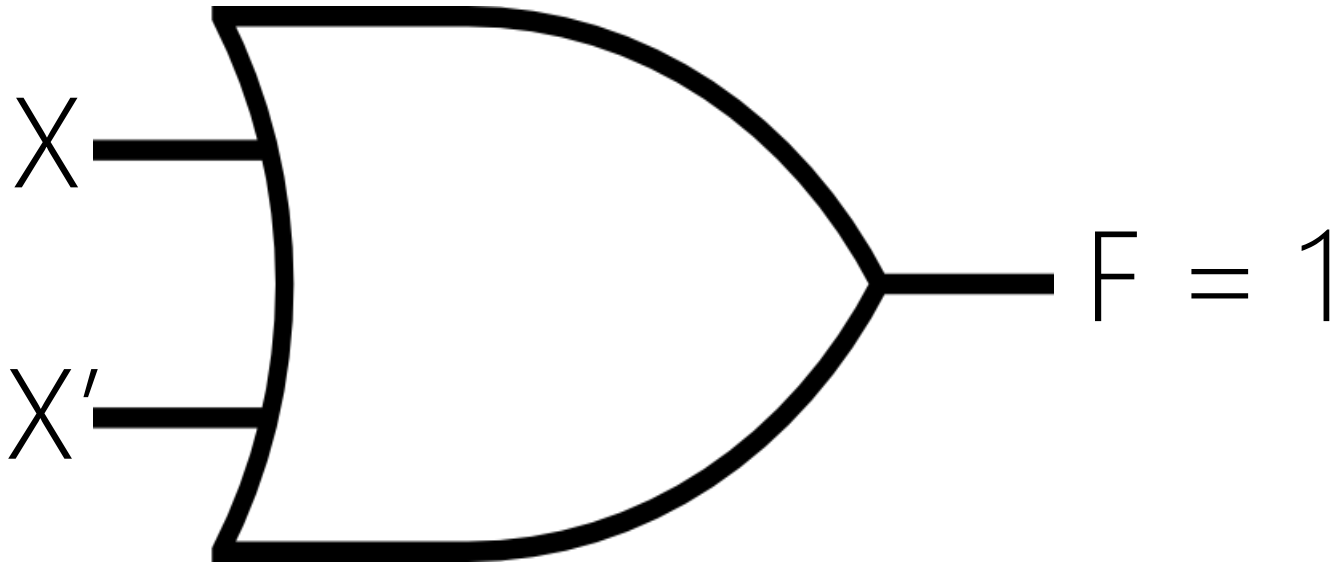
Y	X	Y+X
0	0	0
0	1	1

$$F = X + 0 = X$$



X	X	X+X
0	0	0
1	1	1

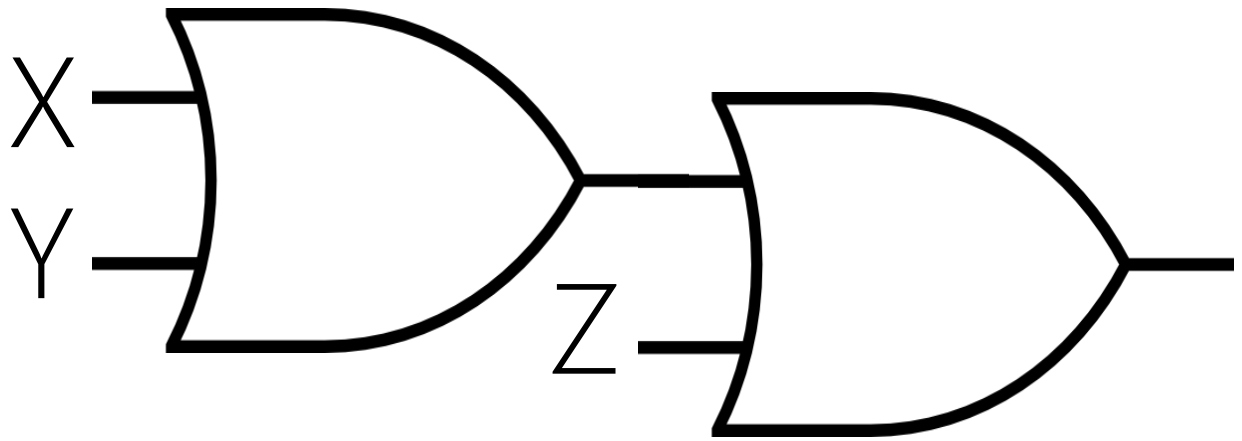
$$F = X+X = X$$



X′	X	X′+X
1	0	1
0	1	1

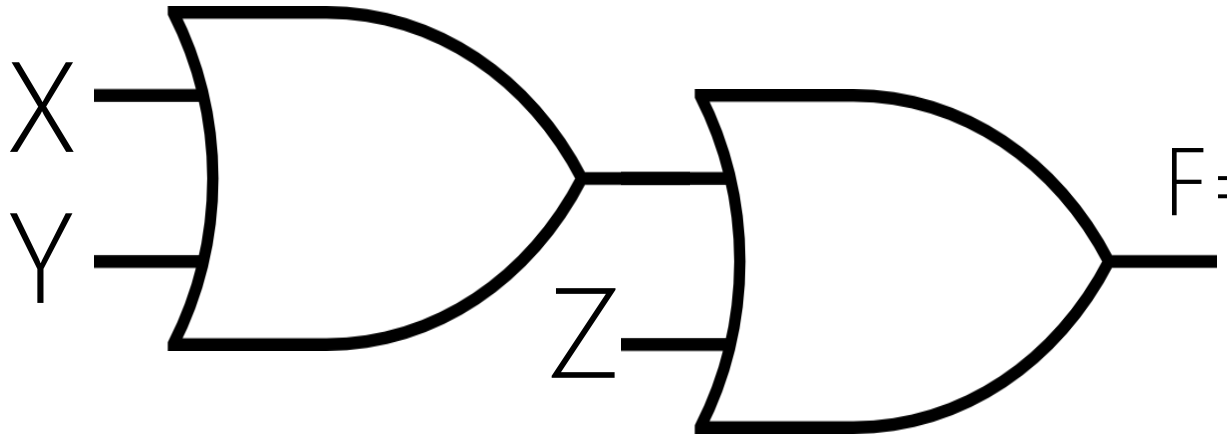
$$F = X + X' = 1$$

3-INPUT OR



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

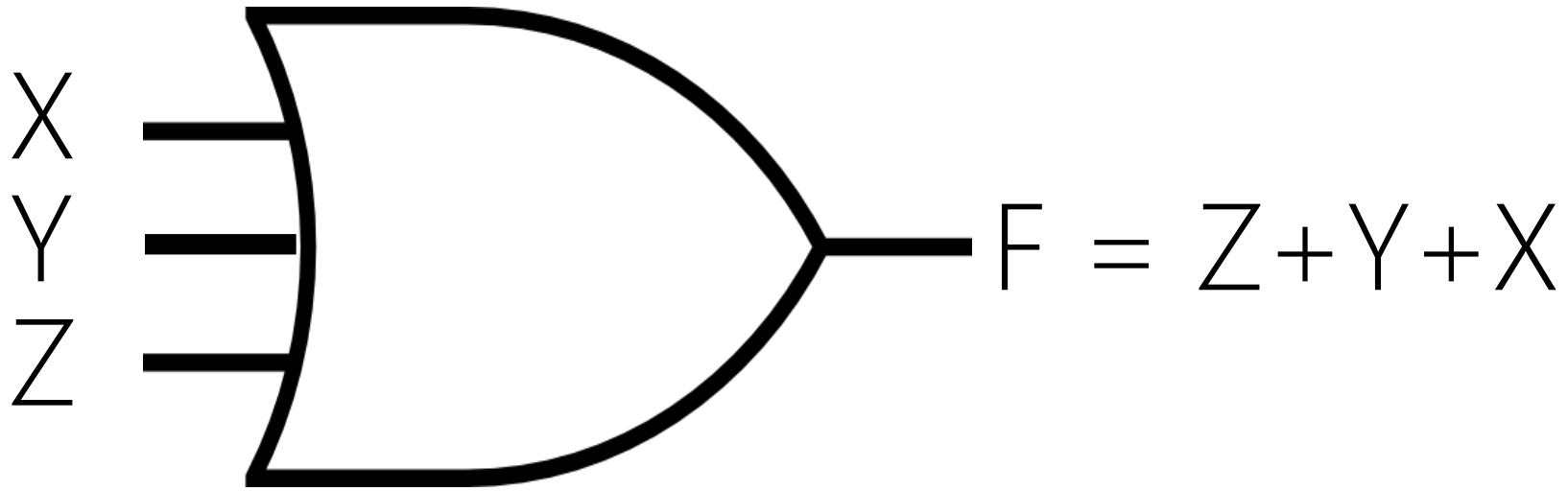
Z	Y	X	Z+(X+Y)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



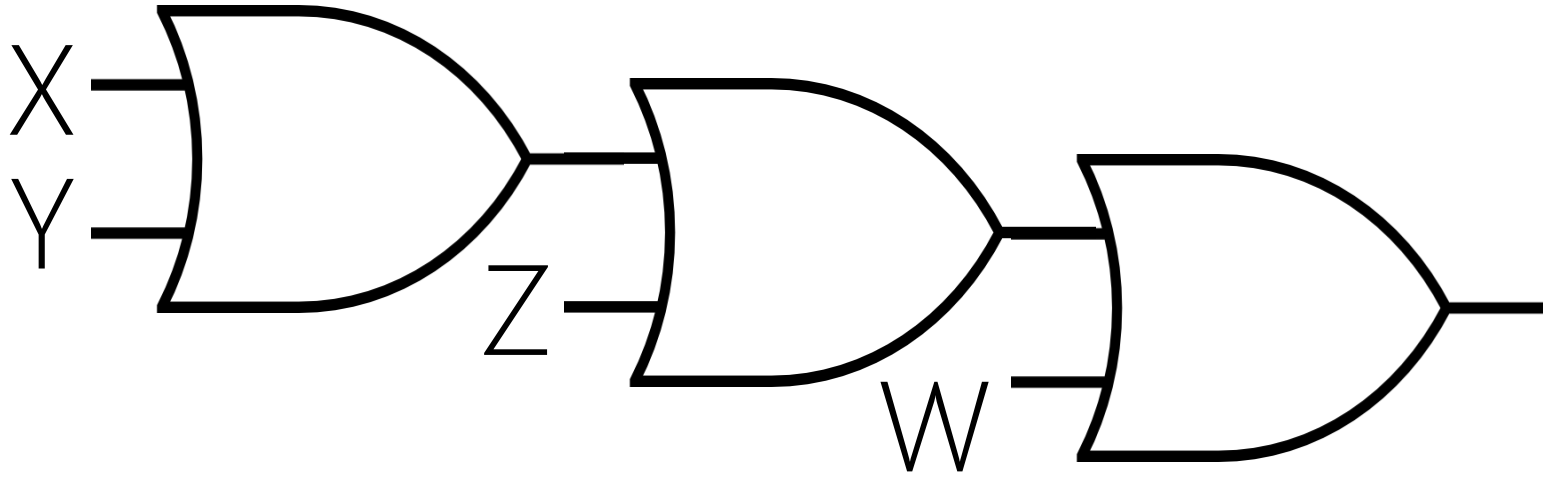
$$F = Z + (Y + X) = Z + (X + Y) = (Z + X) + Y$$
$$= Z + Y + X$$

Associative

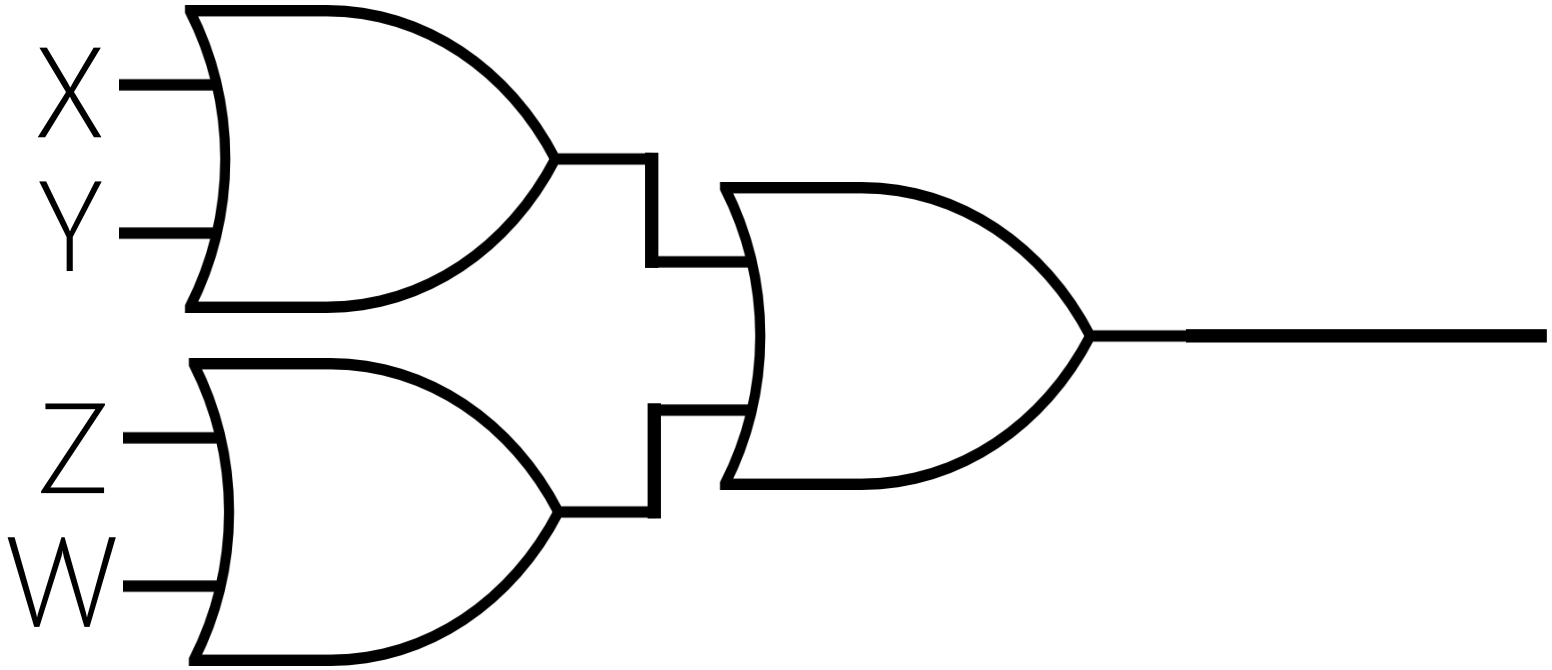
Z	Y	X	Z+(Y+X)	Z+(X+Y)	(Z+X)+Y	ZXY
0	0	0	0			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	1			
1	0	1	1			
1	1	0	1			
1	1	1	1			



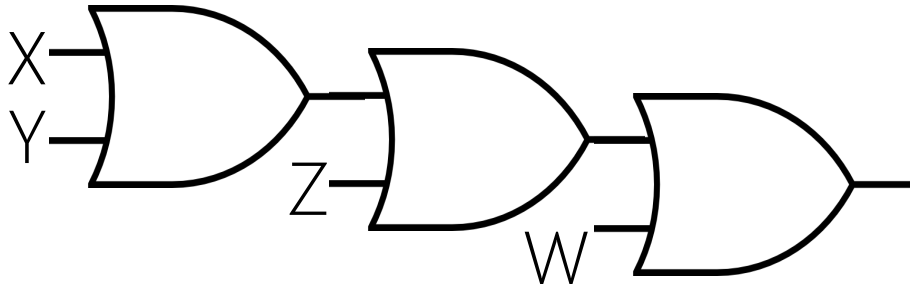
Z	Y	X	Z+(Y+X)	Z+(X+Y)	(Z+X)+Y	ZXY
0	0	0	0			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	1			
1	0	1	1			
1	1	0	1			
1	1	1	1			



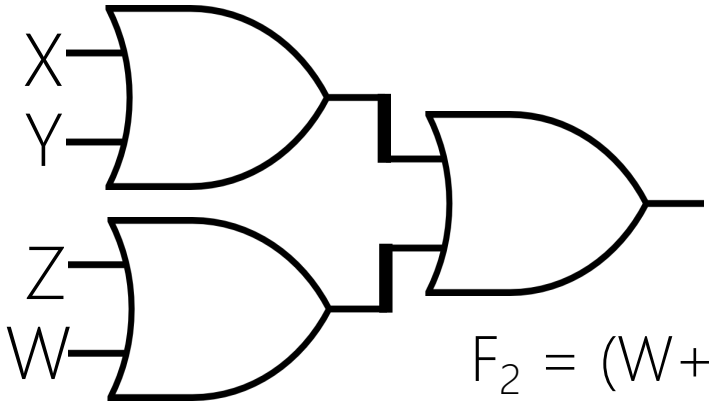
$F_1 = W+(Z+(Y+X))$



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



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



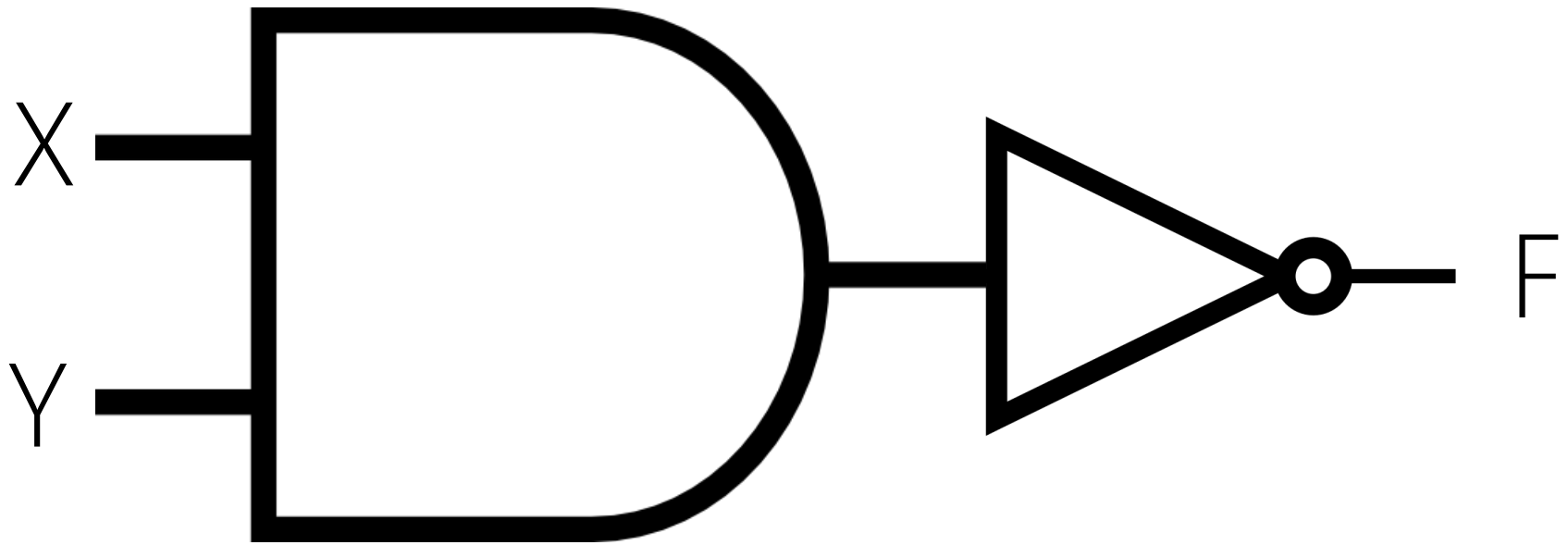
$F_2 = (W+Z)+(Y+X)$

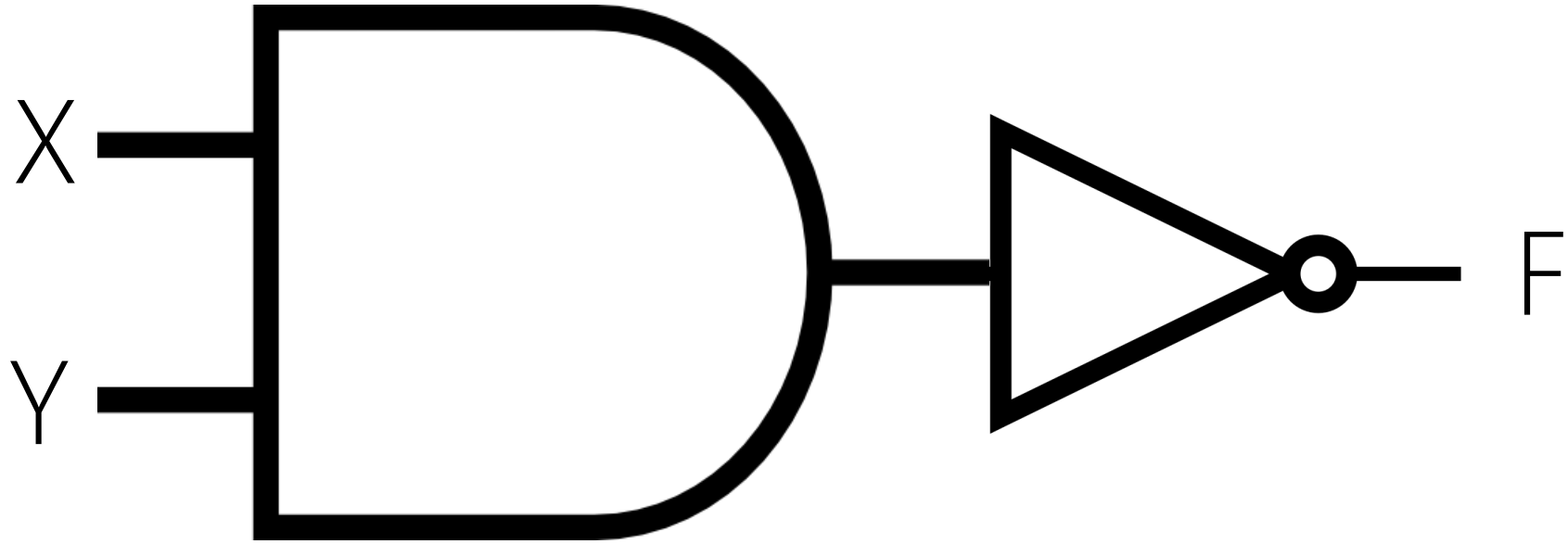
F = W+Z+Y+X	F_1	F_2
Effective (True)	Yes	Yes
Efficient (Fast)	Hmm, 3 levels, No!	Yes! 2 levels
Min. Cost	3 gates, Yes	3 gates, Yes

ANALYSIS I

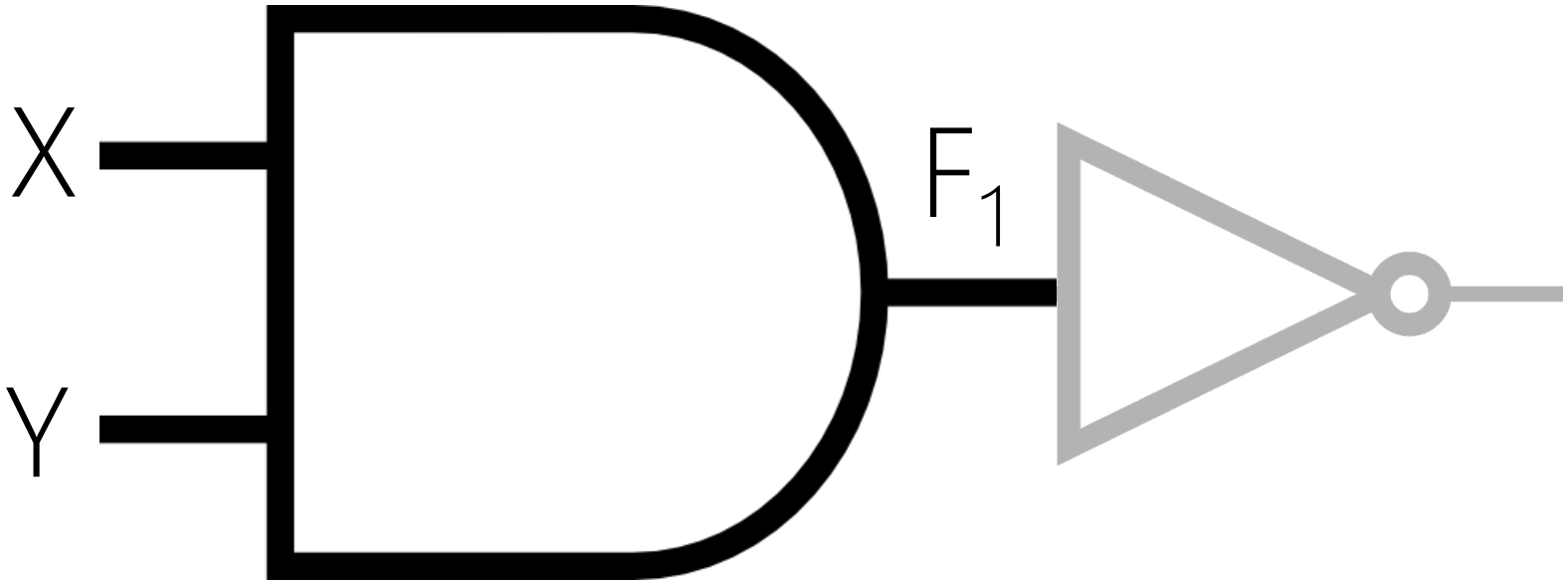
System analysis is given the structure of a system, find its functionality.

Determine the functionality exhibited by a structure.

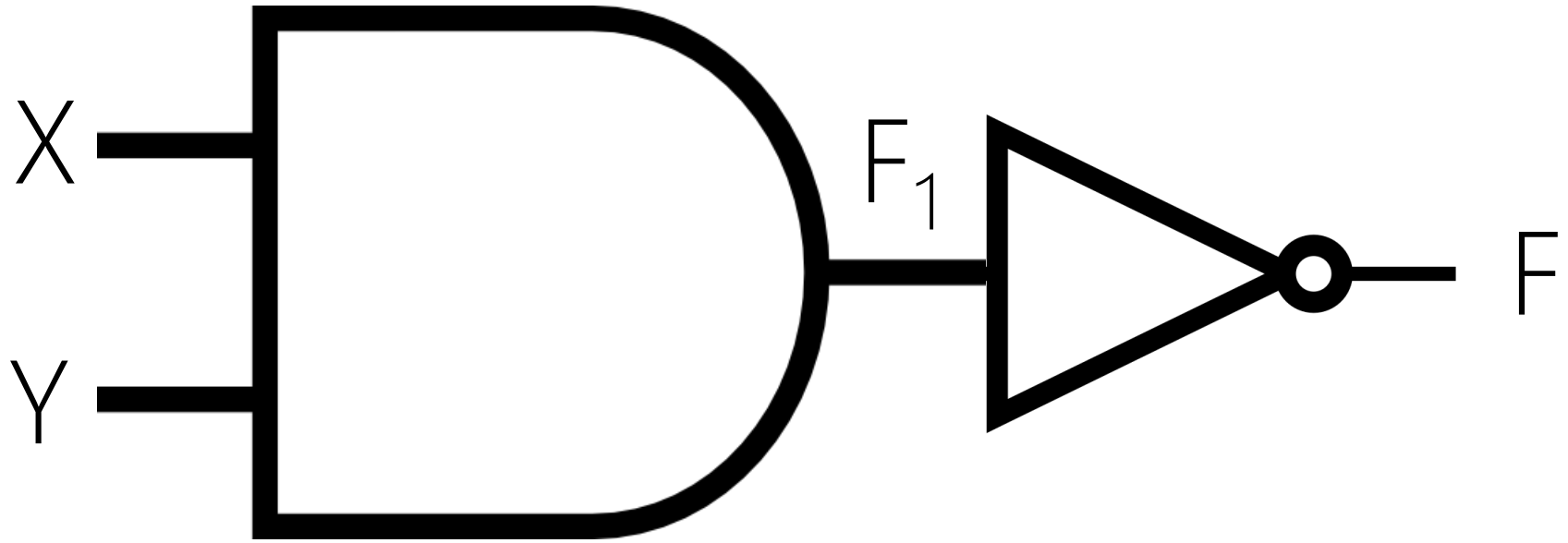

X
Y
F



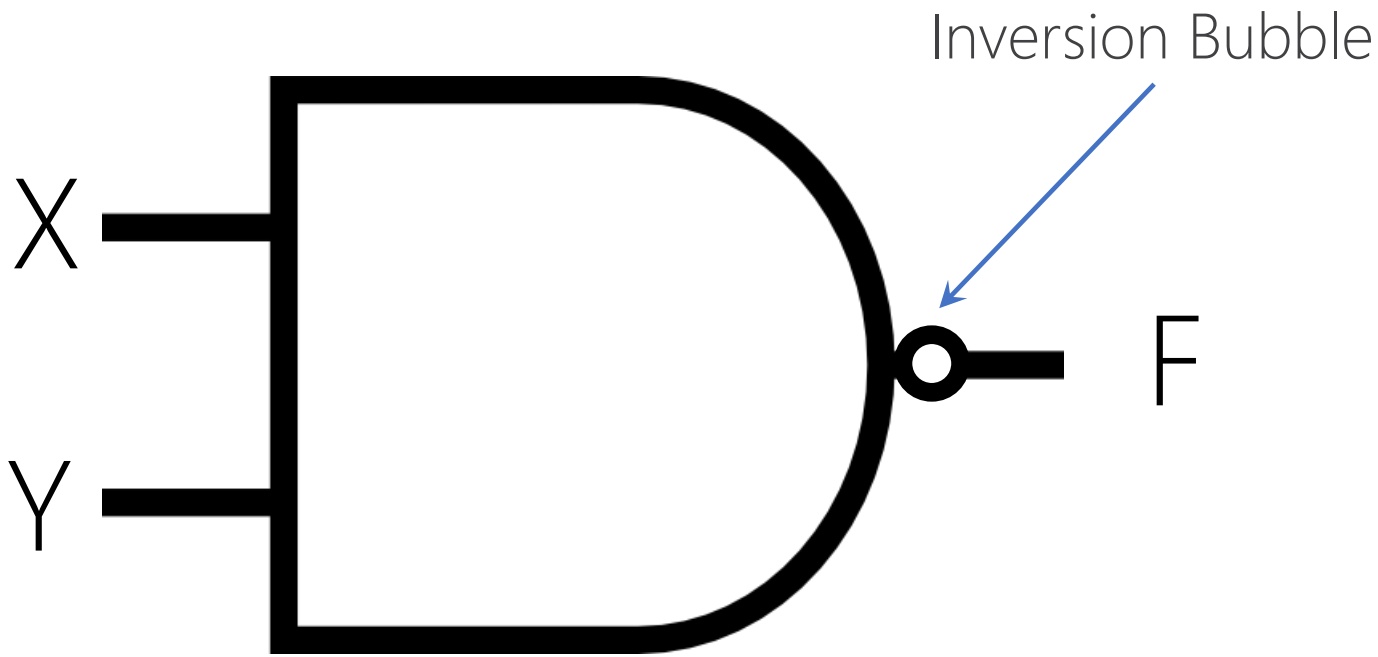
Y	X	F = ?
0	0	?
0	1	?
1	0	?
1	1	?



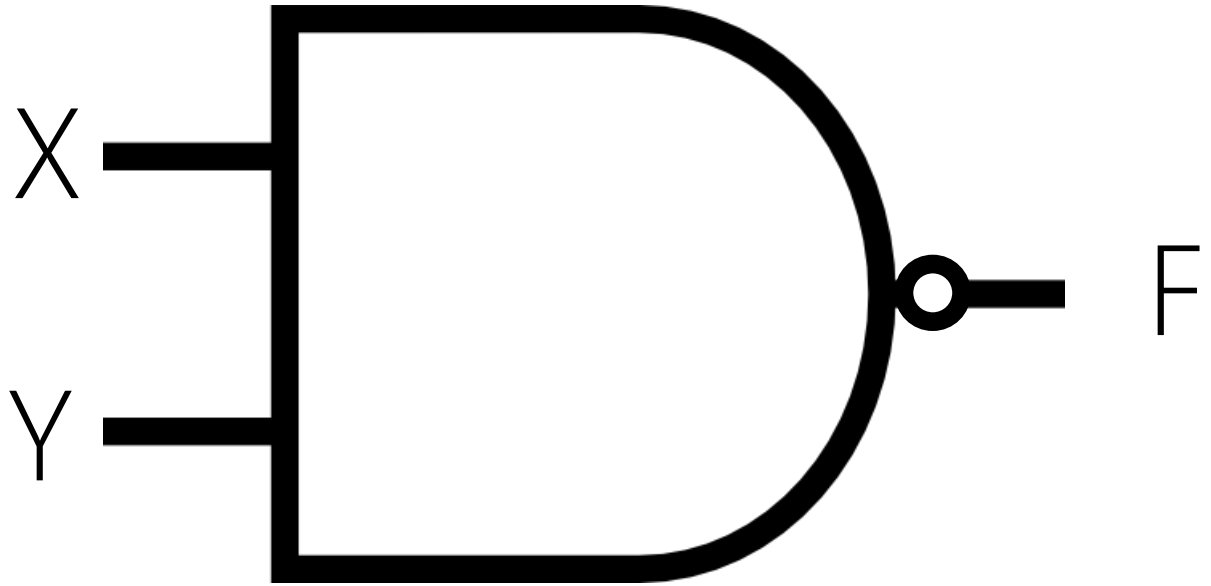
Y	X	$F_1 = YX$
0	0	0
0	1	0
1	0	0
1	1	1



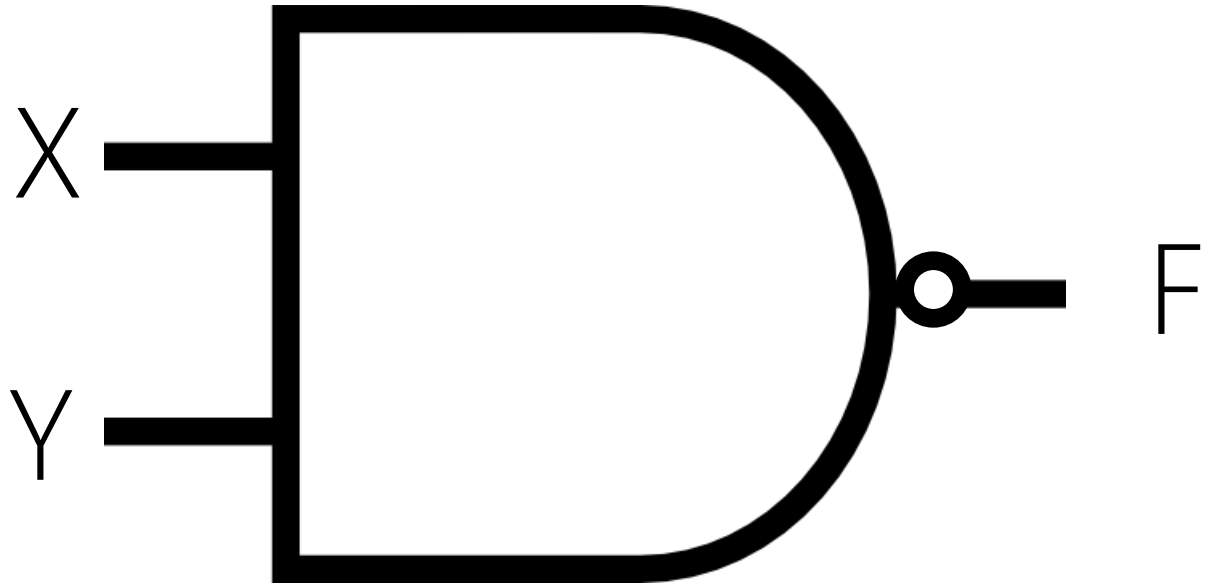
Y	X	$F_1 = YX$	$F = (YX)'$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



NAND (Not – AND)

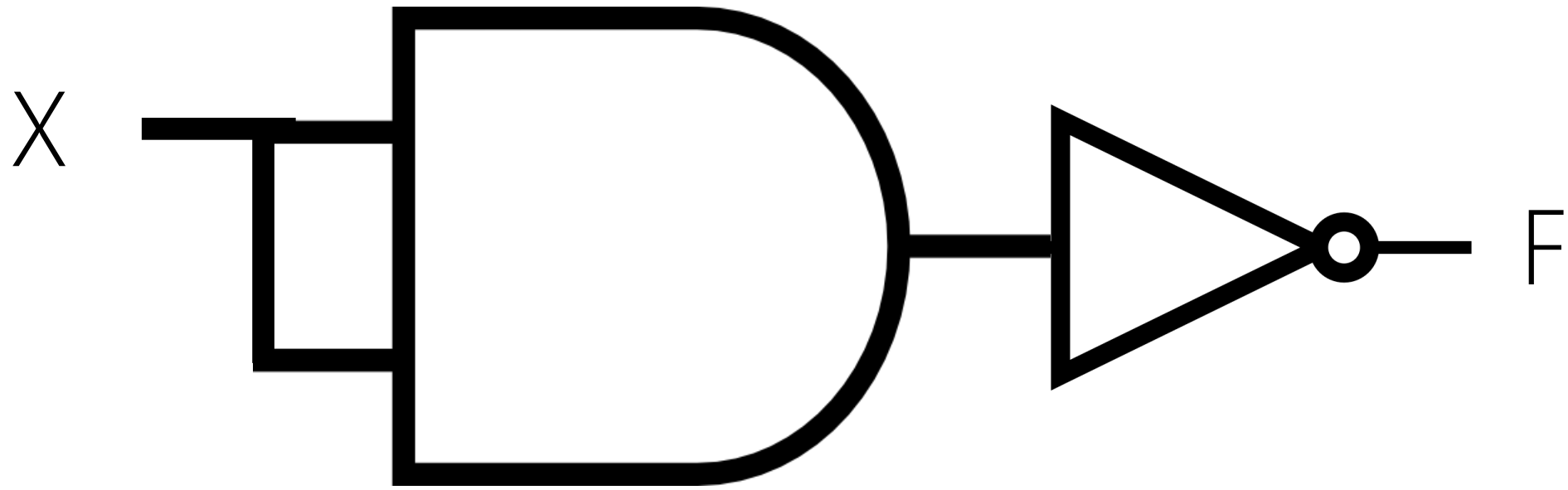


Y	X	F = (YX)′	F=Y ↑ X
0	0	1	
0	1	1	
1	0	1	
1	1	0	

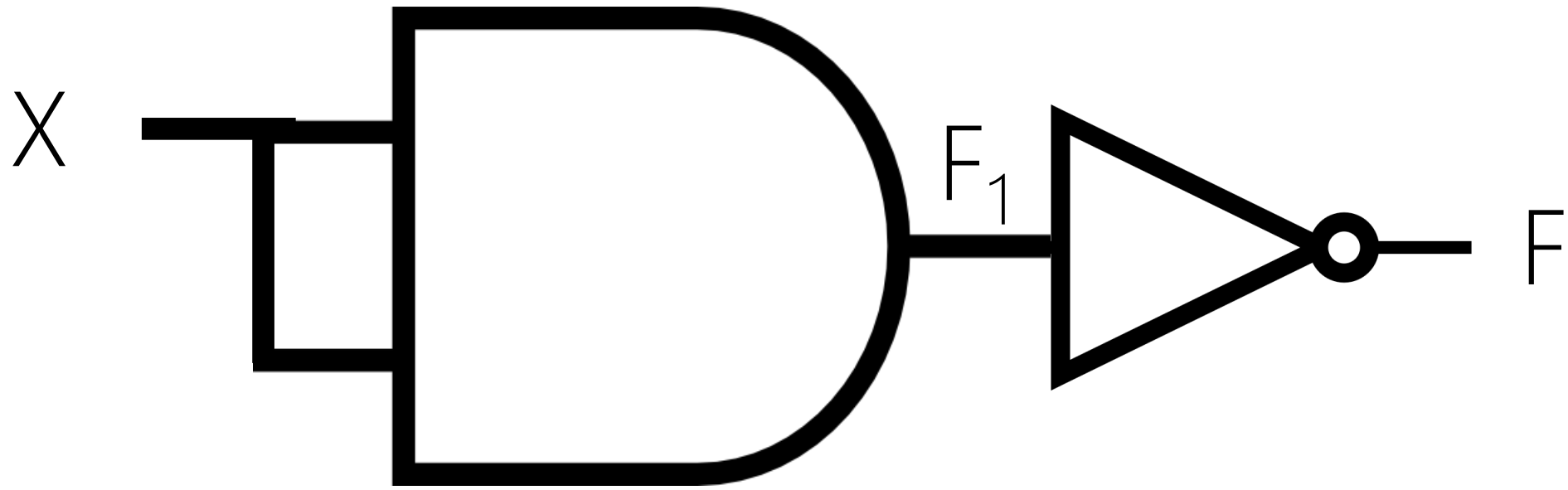


$$F = (YX)' = (XY)' = Y \uparrow X = X \uparrow Y$$

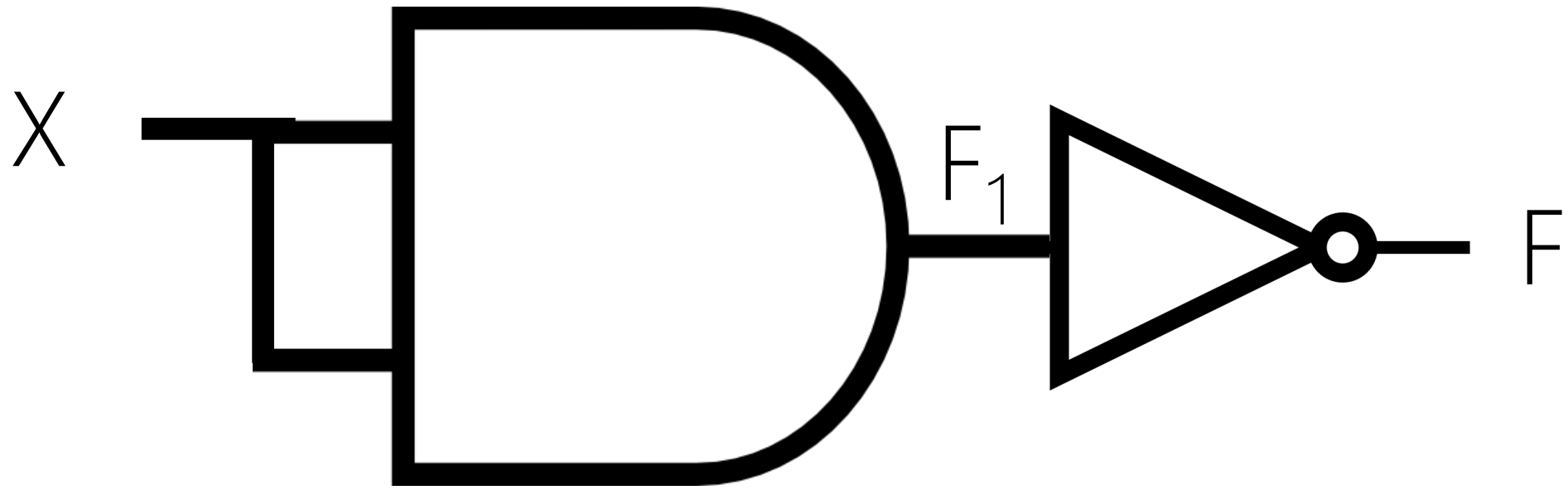
Commutative



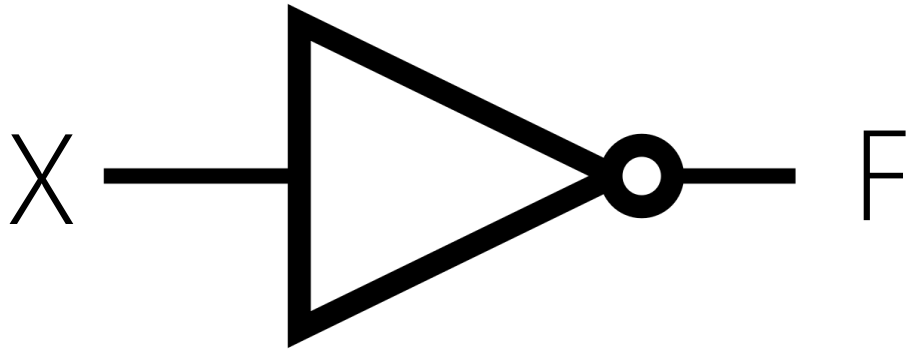
X	F = ?
0	
1	



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



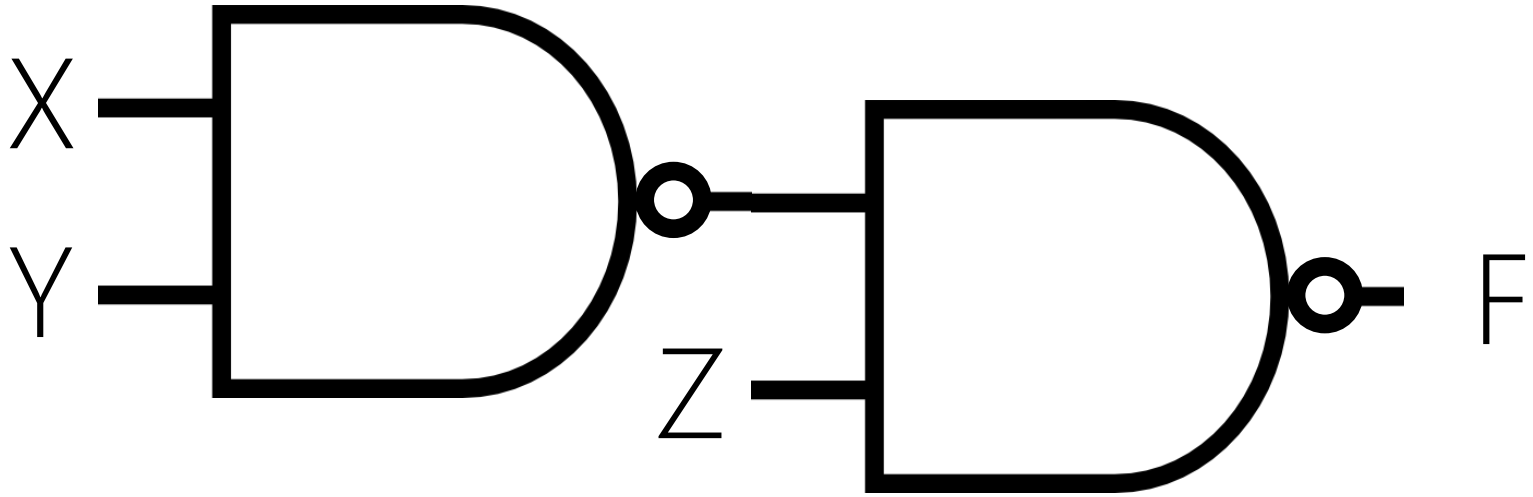
X	$F_1 = XX$	$F = (XX)'$
0	0	1
1	1	0



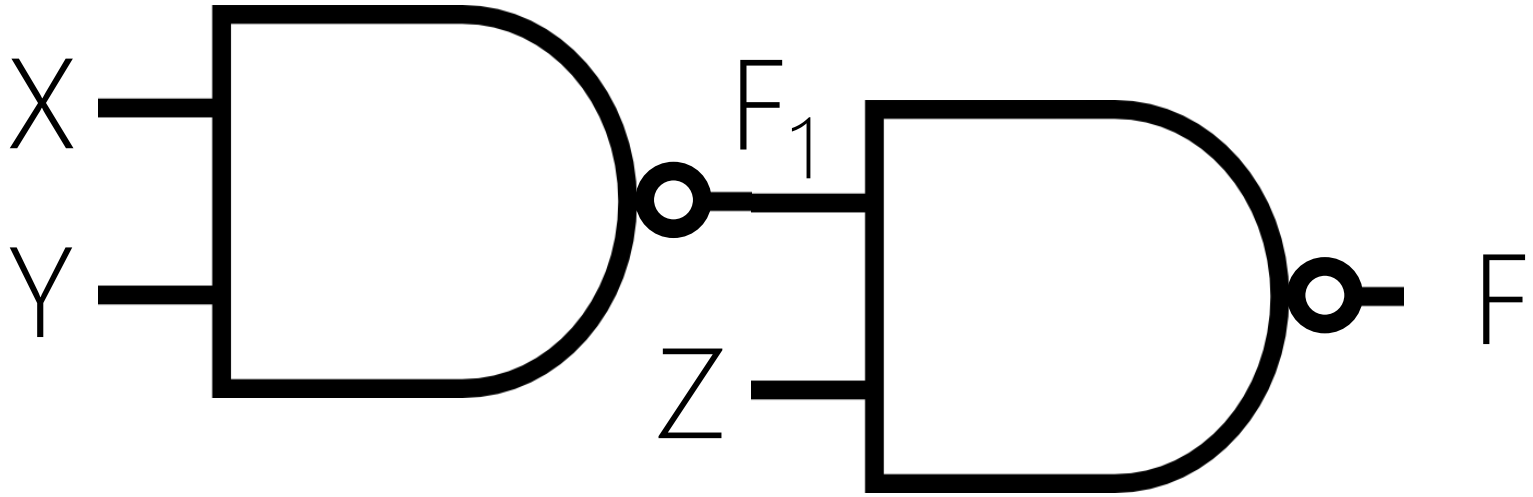
X	F = (XX)′=X↑X = X′
0	1
1	0

3-INPUT NAND

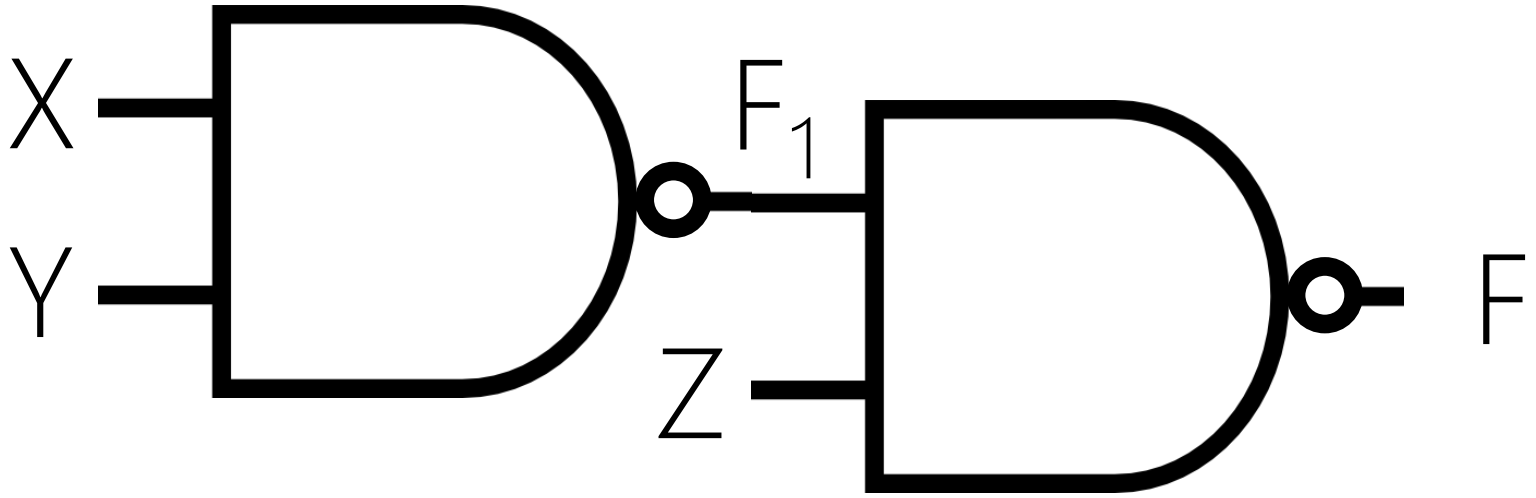
Z	Y	X	F=(ZYX)’
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



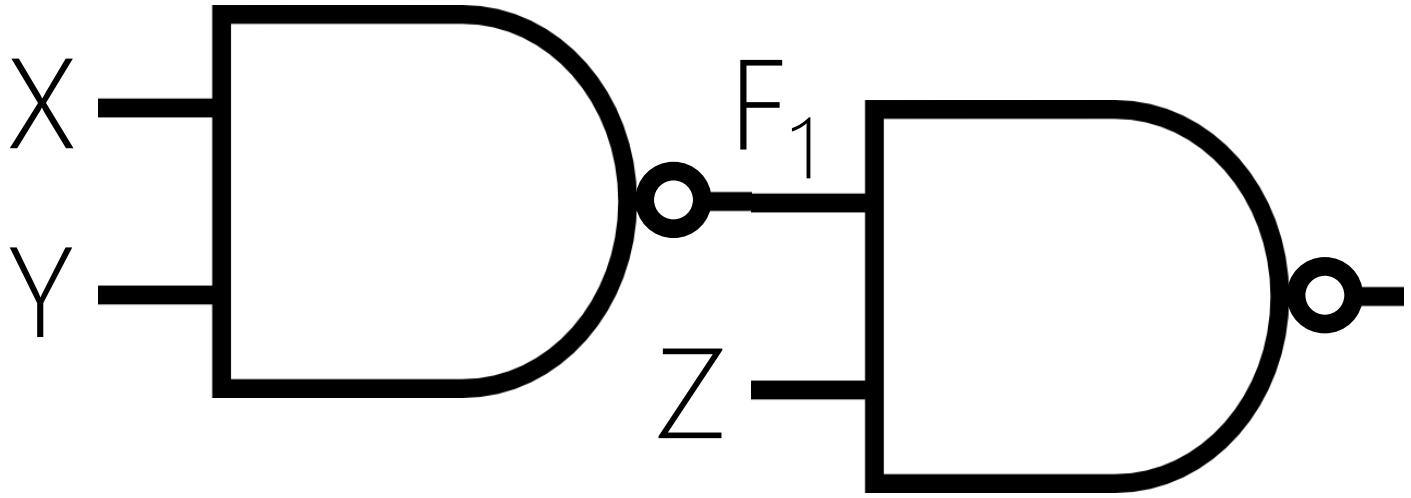
Z	Y	X	F = ?
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



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	



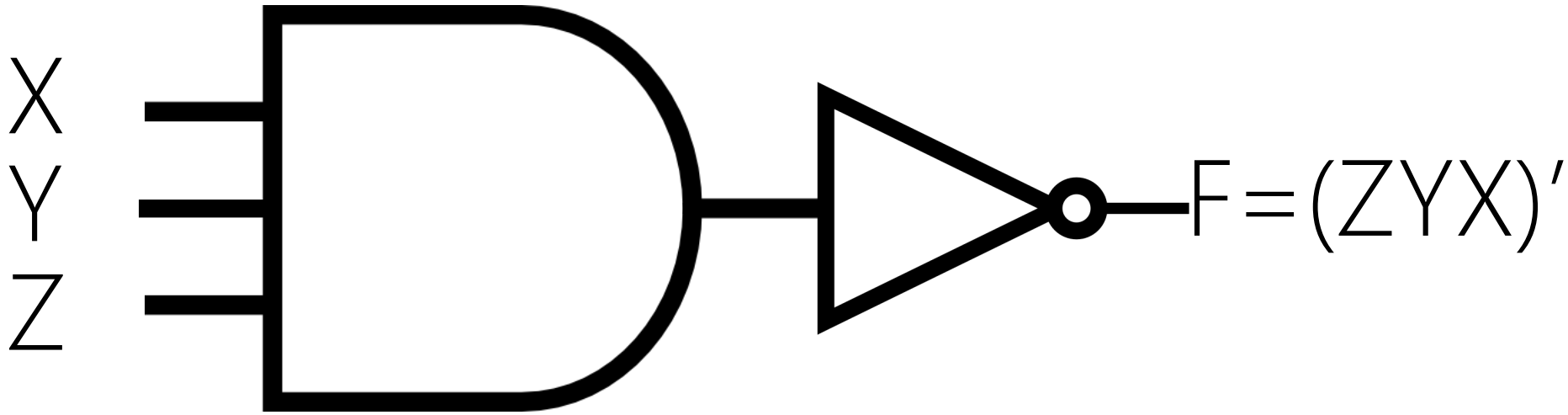
Z	Y	X	$F_1 = (YX)'$	$F = (ZF_1)' = (Z(YX)')'$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1



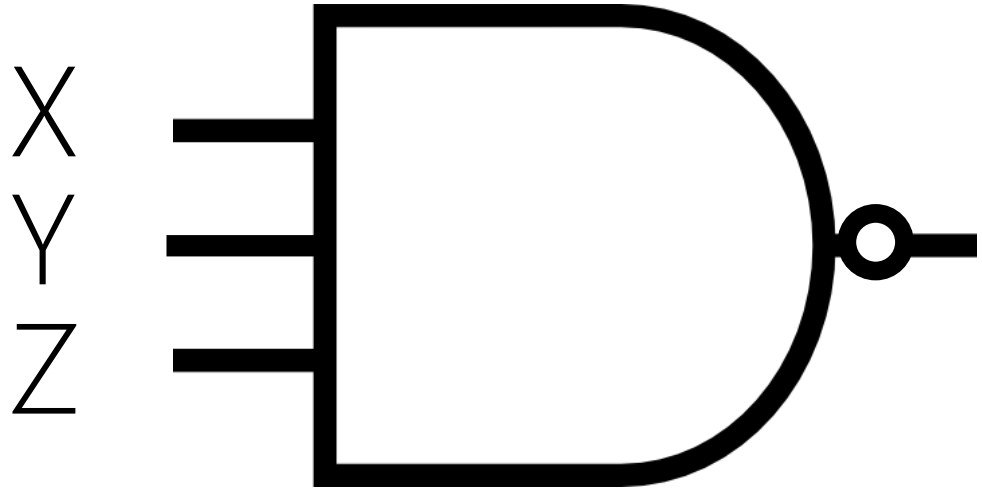
$$F = (Z(YX)')' \neq (ZYX)'$$

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

NOT (3-INPUT AND)



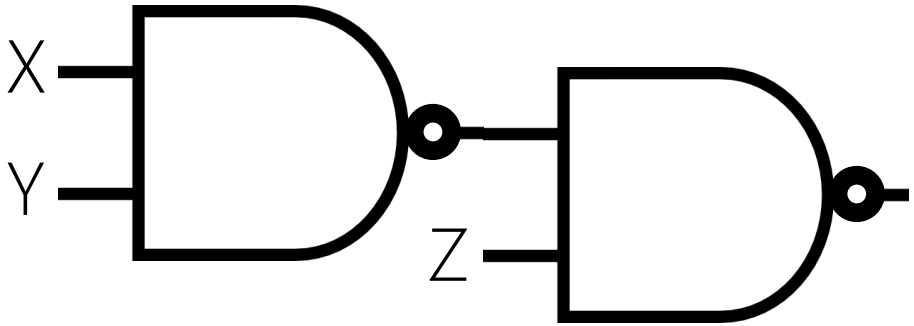
Z	Y	X	F=(ZYX)'	F=(ZYX)'
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



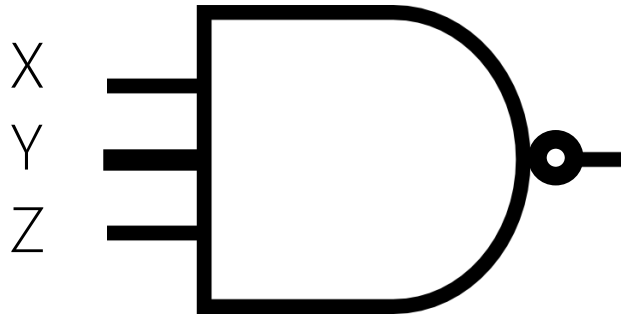
$F=(ZYX)'=(XZY)'$

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

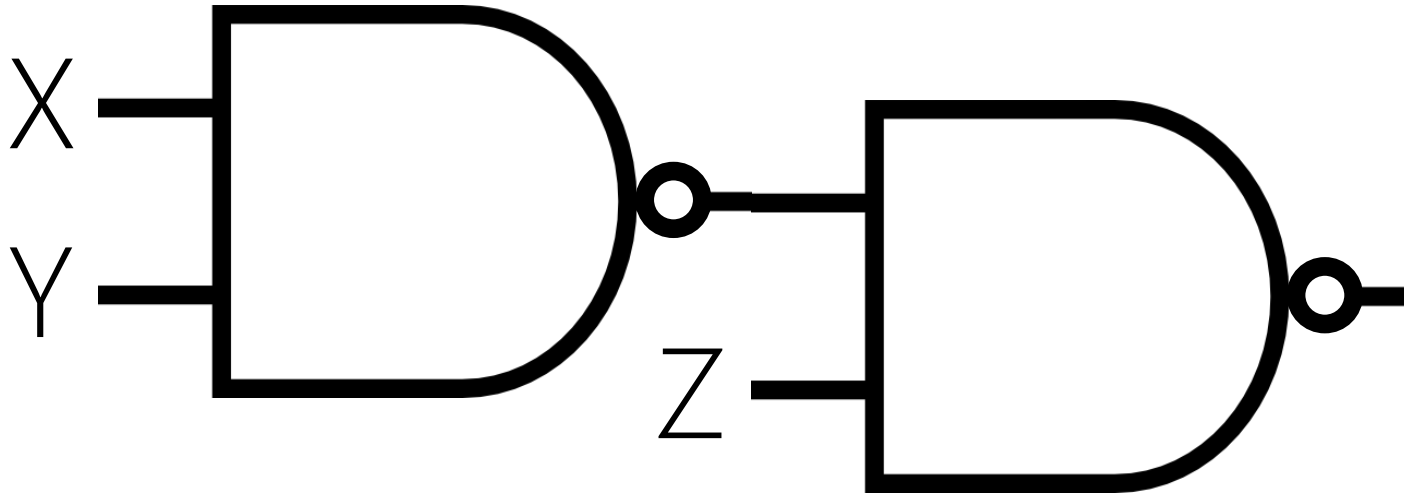


$F_1=(Z(YX)')'$



$F_2=(ZYX)'$

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



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

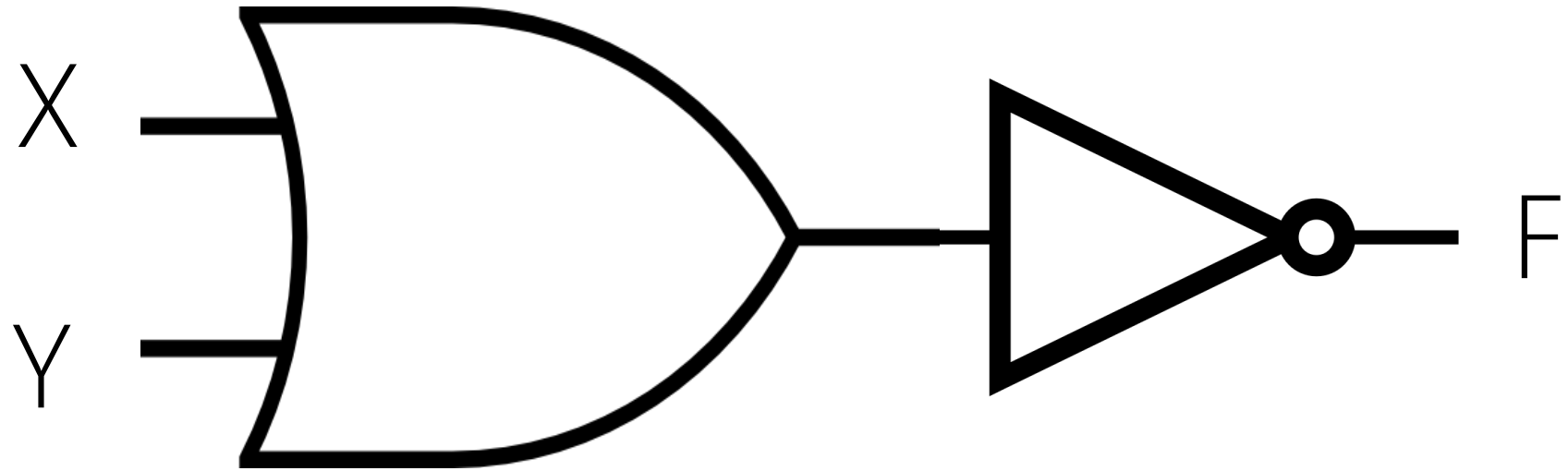
Associative?

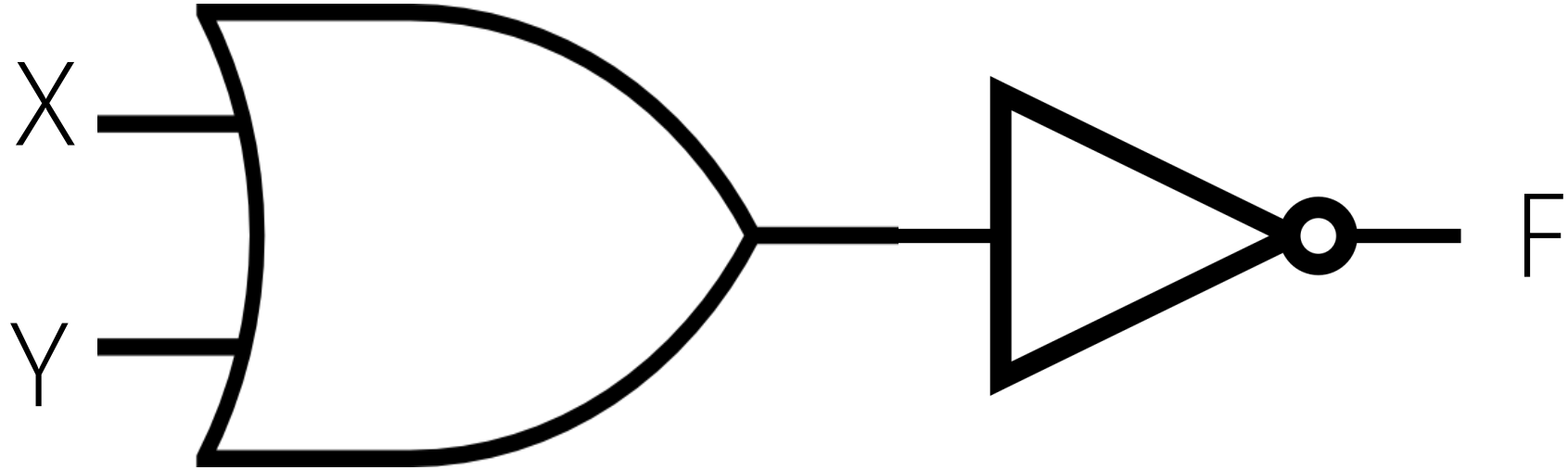
Z	Y	X	$F = (Z(YX)')' = Z \uparrow (Y \uparrow X)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

ANALYSIS II

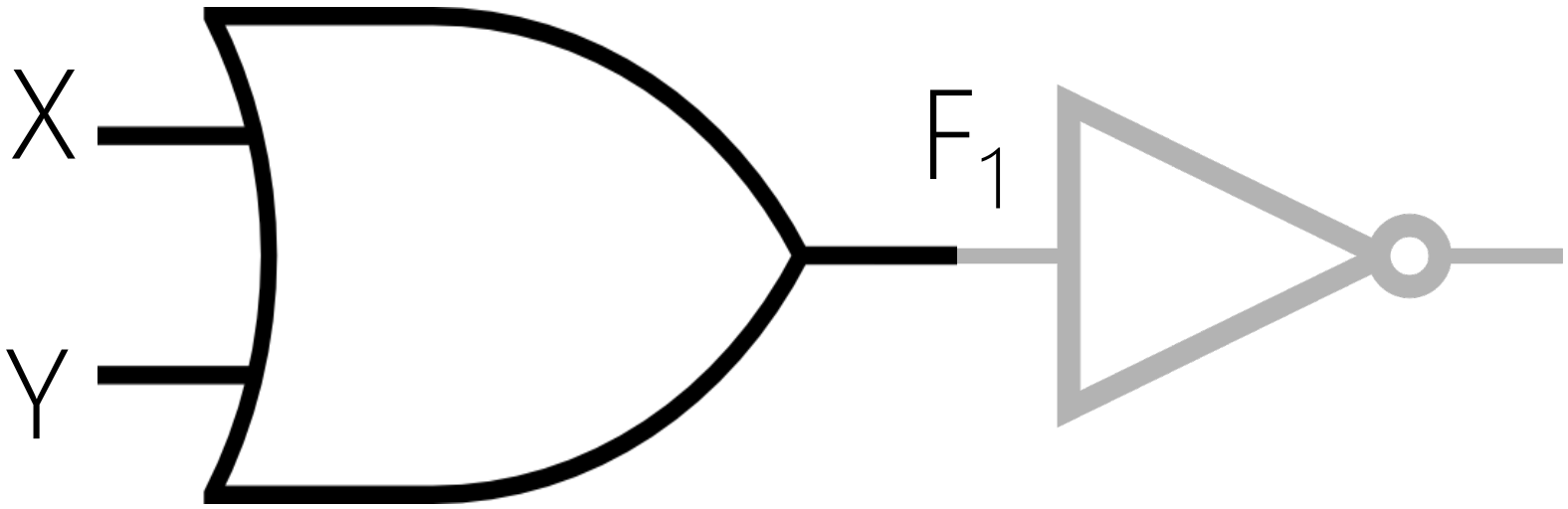
System analysis is given the structure of a system, find its functionality.

Determine the functionality exhibited by a structure.

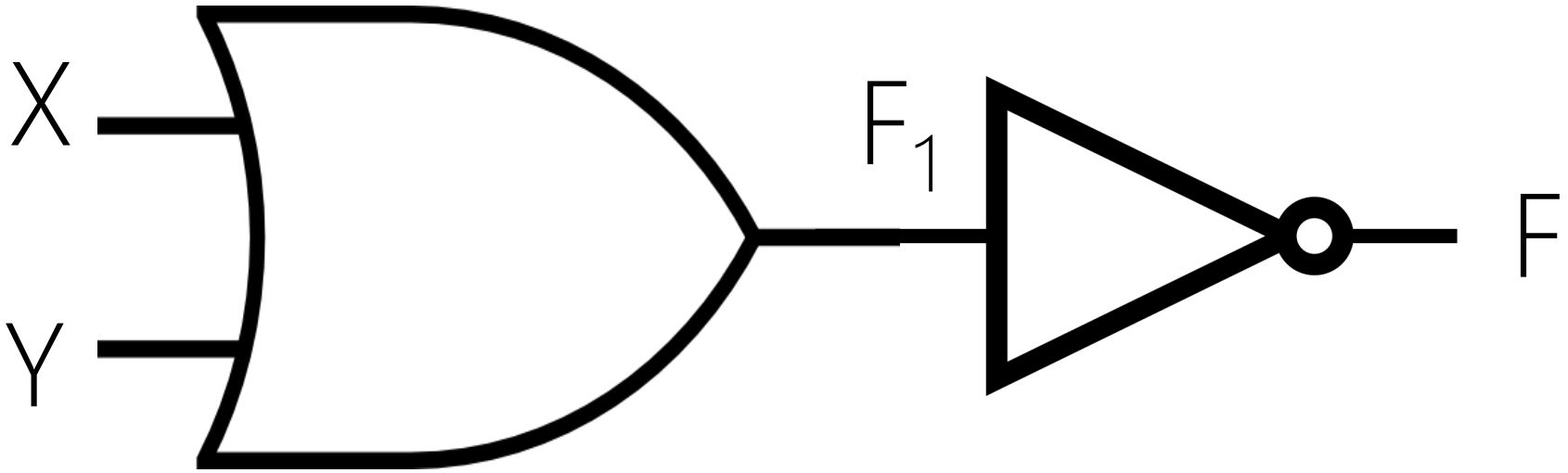

X
Y
F



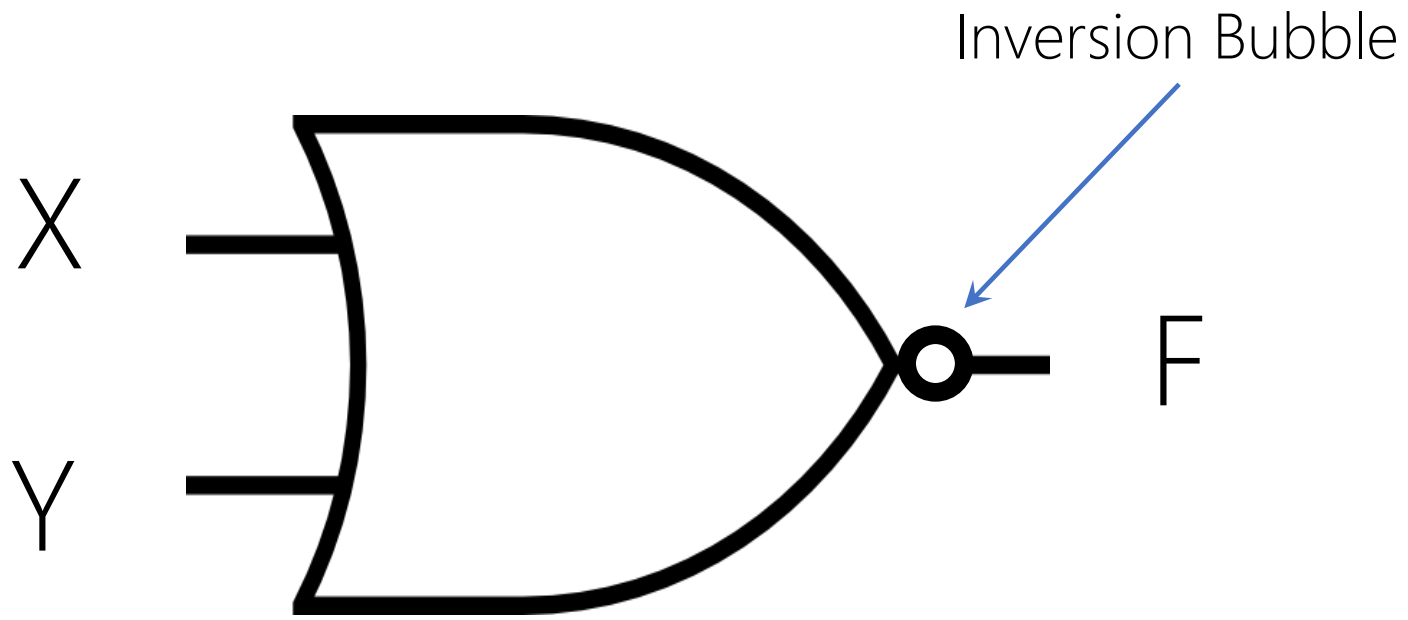
Y	X	F = ?
0	0	?
0	1	?
1	0	?
1	1	?



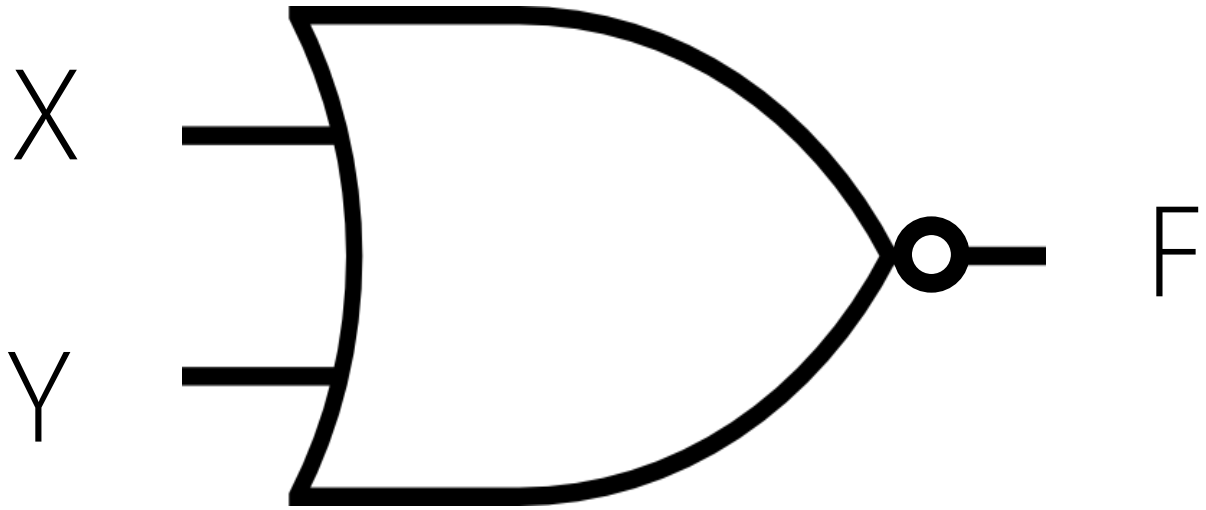
Y	X	$F_1 = Y+X$
0	0	0
0	1	1
1	0	1
1	1	1



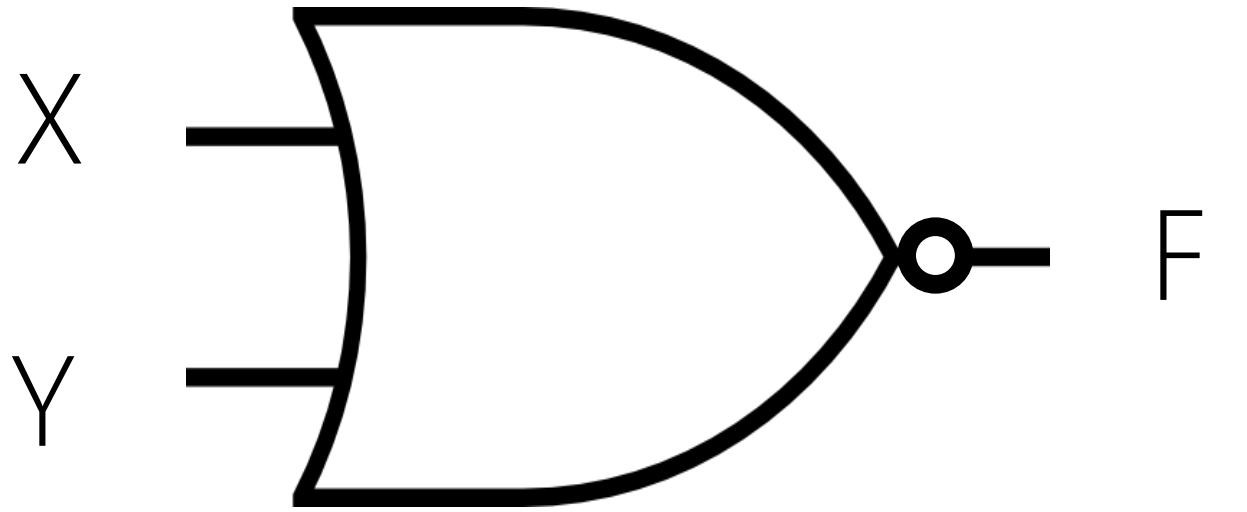
Y	X	$F_1 = Y+X$	$F = (Y+X)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



NOR (Not – OR)

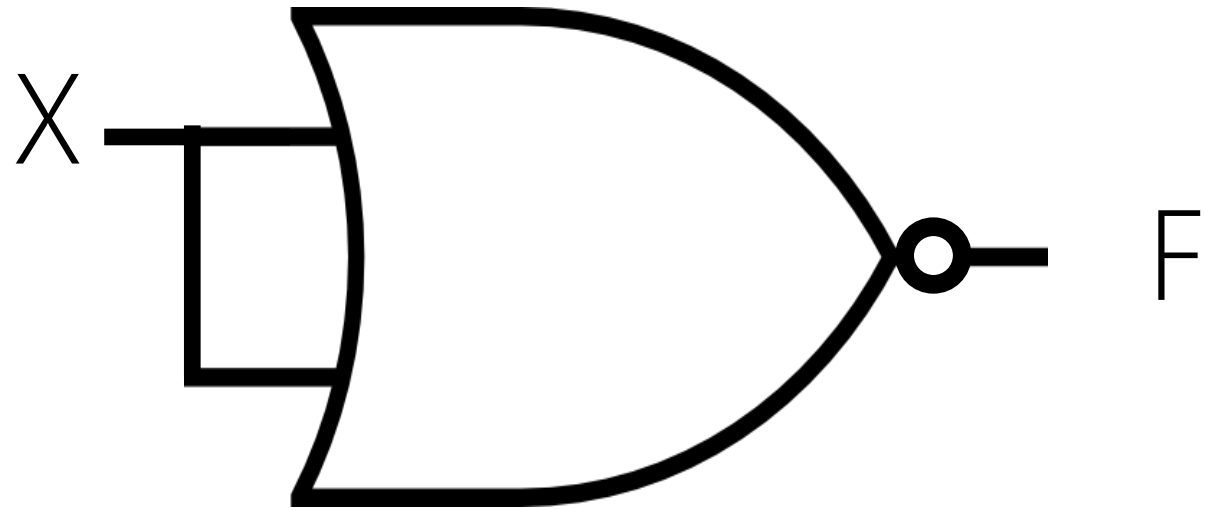


Y	X	F = (Y+X)'	F=Y ↓ X
0	0	1	
0	1	0	
1	0	0	
1	1	0	

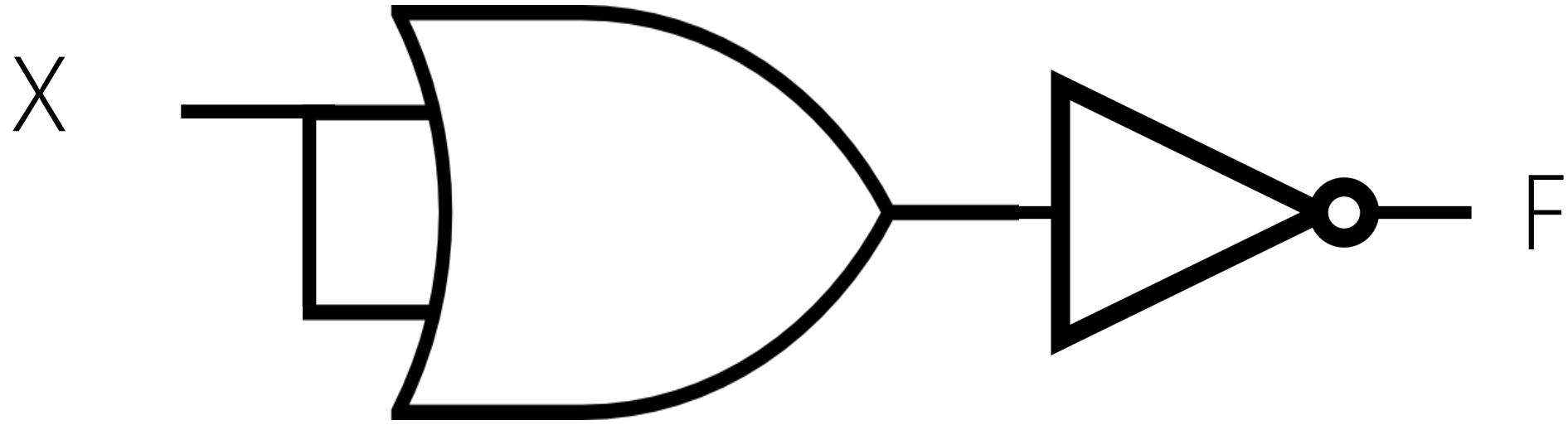


$$F = (Y + X)' = (X + Y)' = Y \downarrow X = X \downarrow Y$$

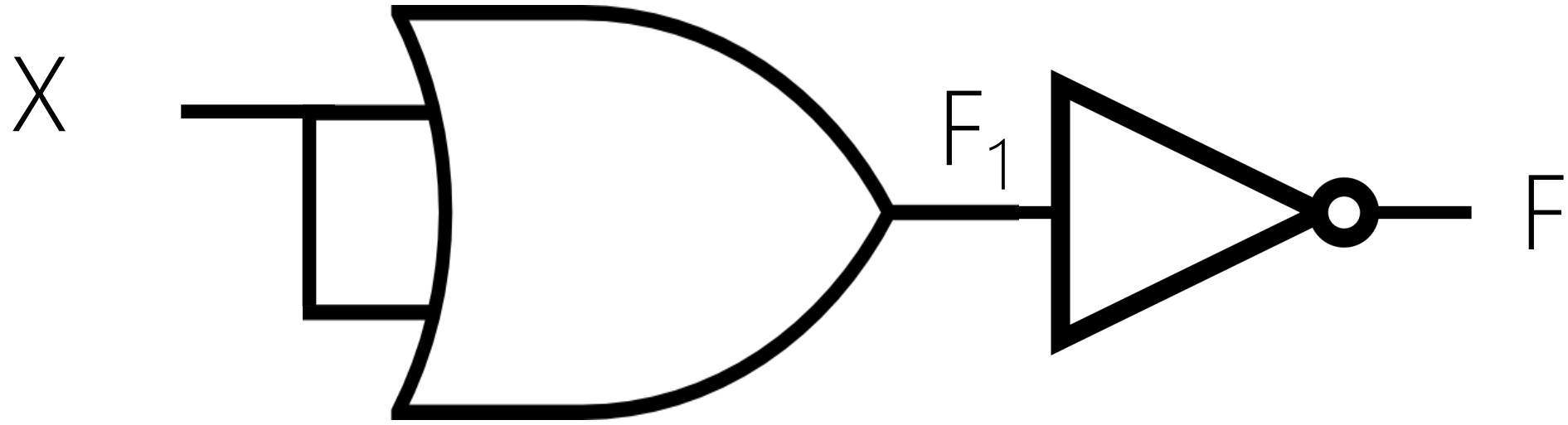
Commutative



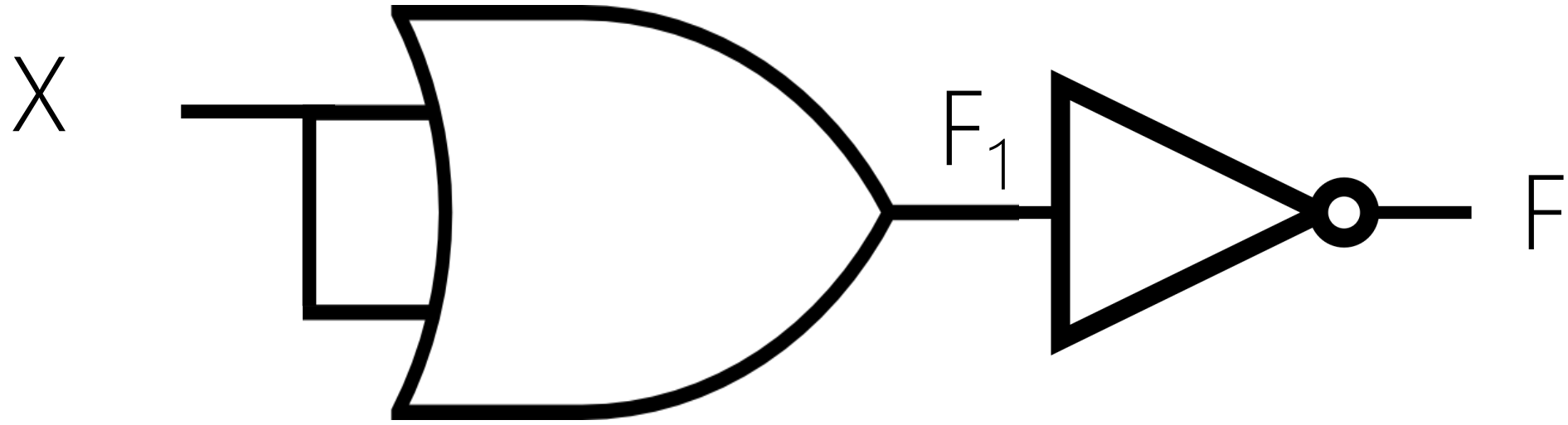
F=?



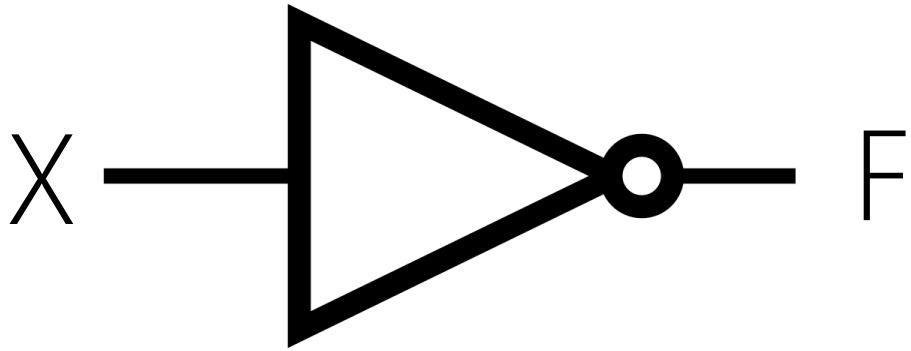
X	F = ?
0	
1	



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



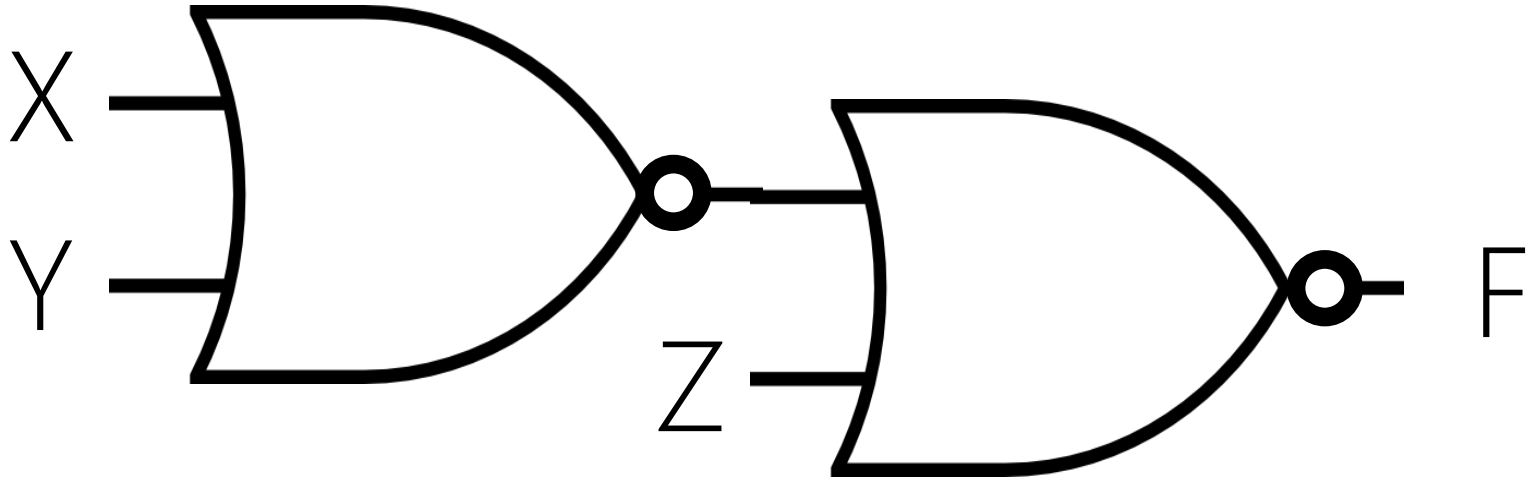
X	$F_1 = X+X$	$F = (X+X)'$
0	0	1
1	1	0



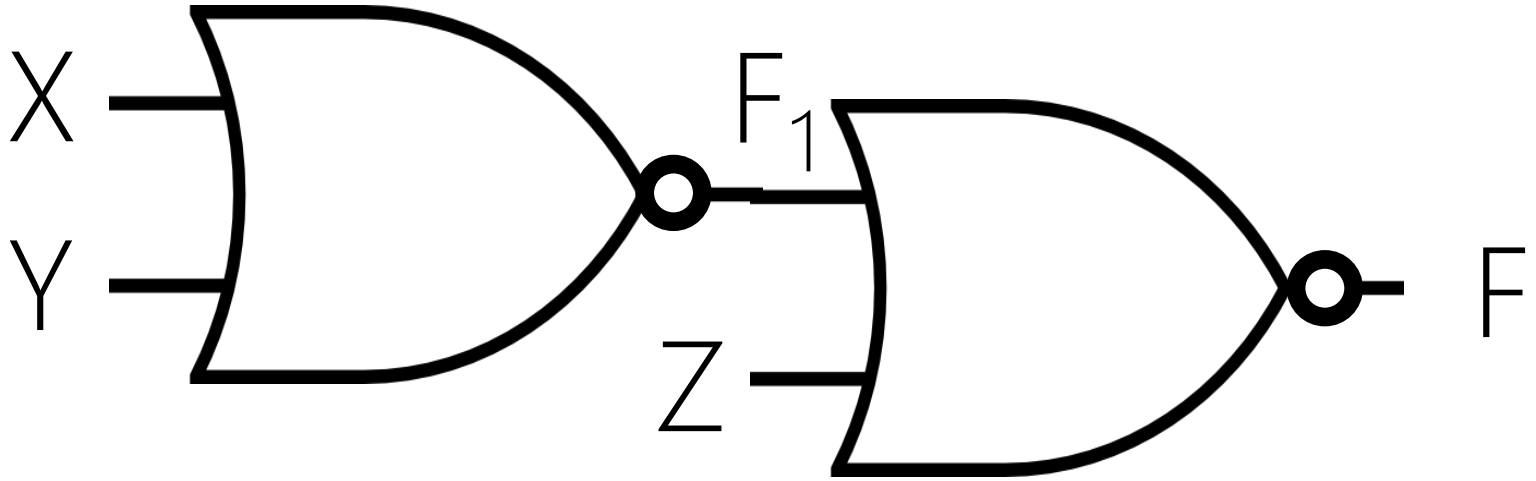
X	F = (X+X)′= X↓X =X′
0	1
1	0

3-INPUT NOR

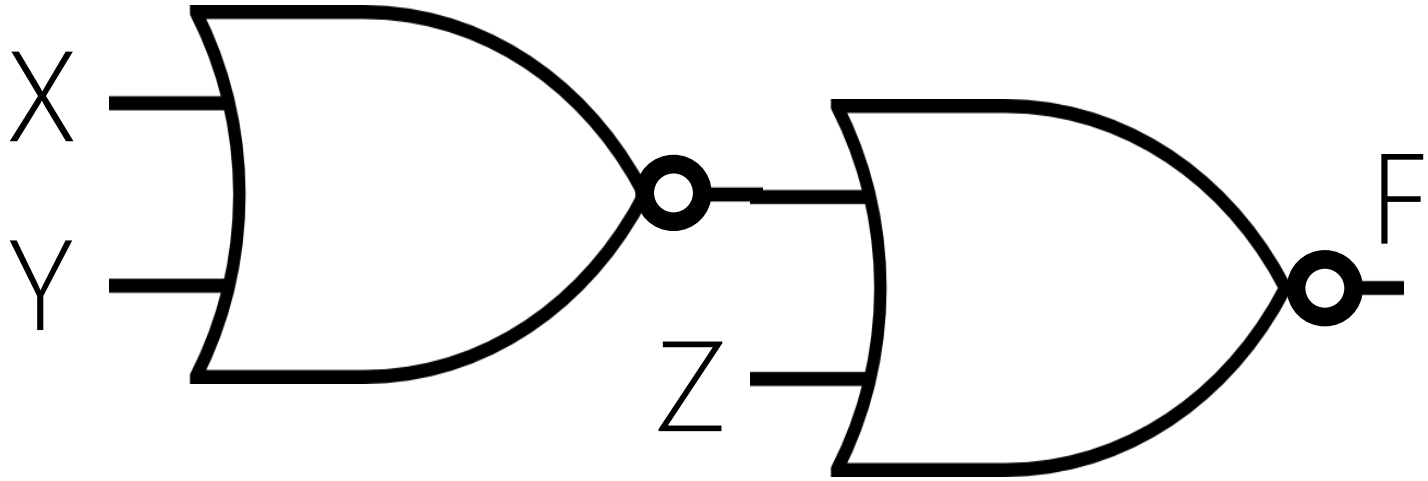
Z	Y	X	F=(Z+Y+X)’
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



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	



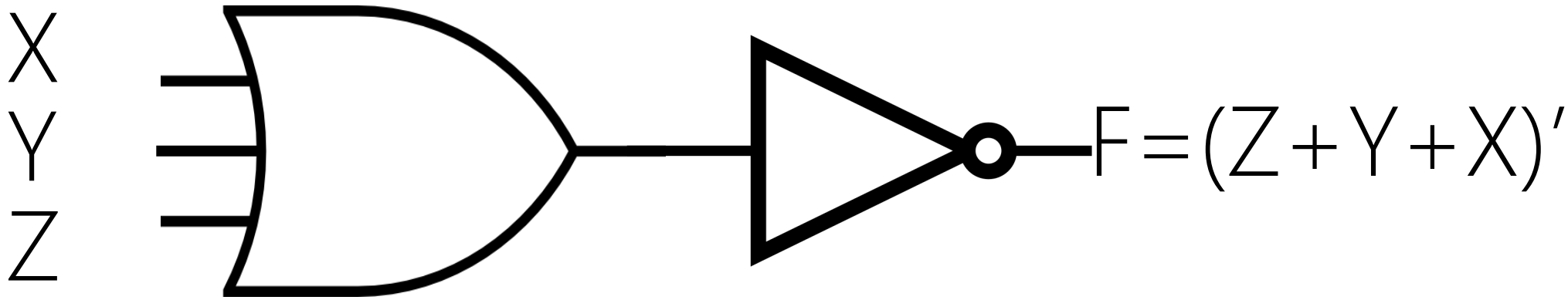
Z	Y	X	$F_1 = (Y+X)'$	$F = (Z+F_1)' = (Z+(Y+X)')'$
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0



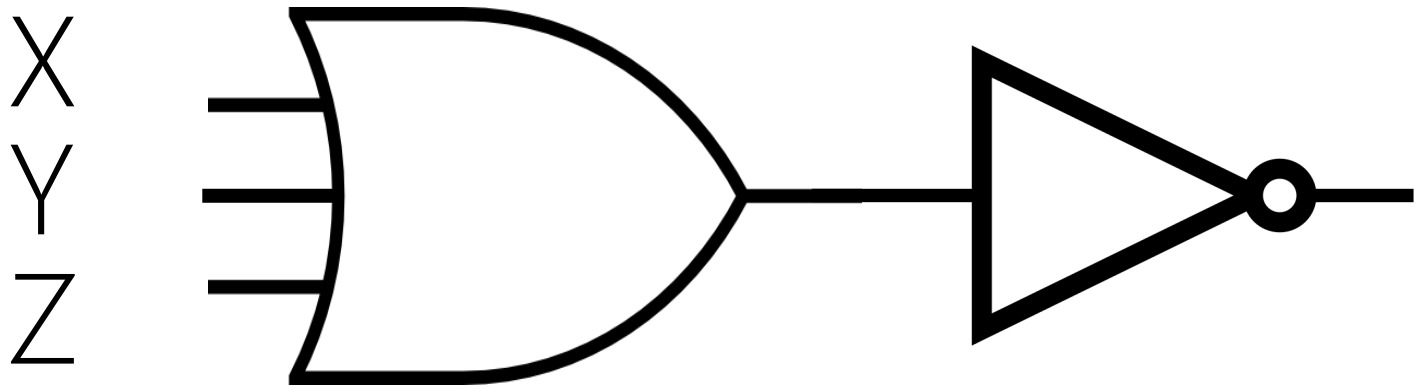
$$F = (Z + (Y + X)')'$$
$$\neq (Z + Y + X)'$$

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)



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

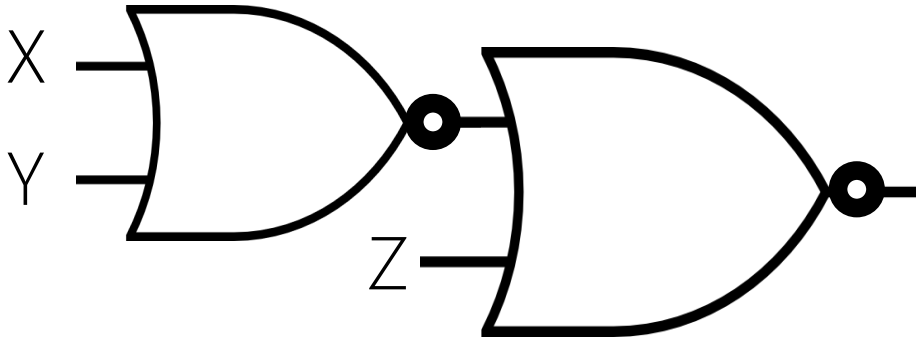


$F=(Z+Y+X)'$

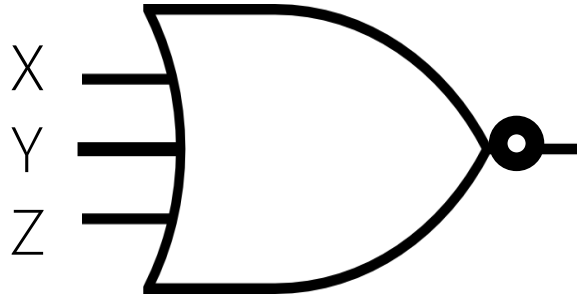
$F=(X+Y+Z)'$

Associative

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

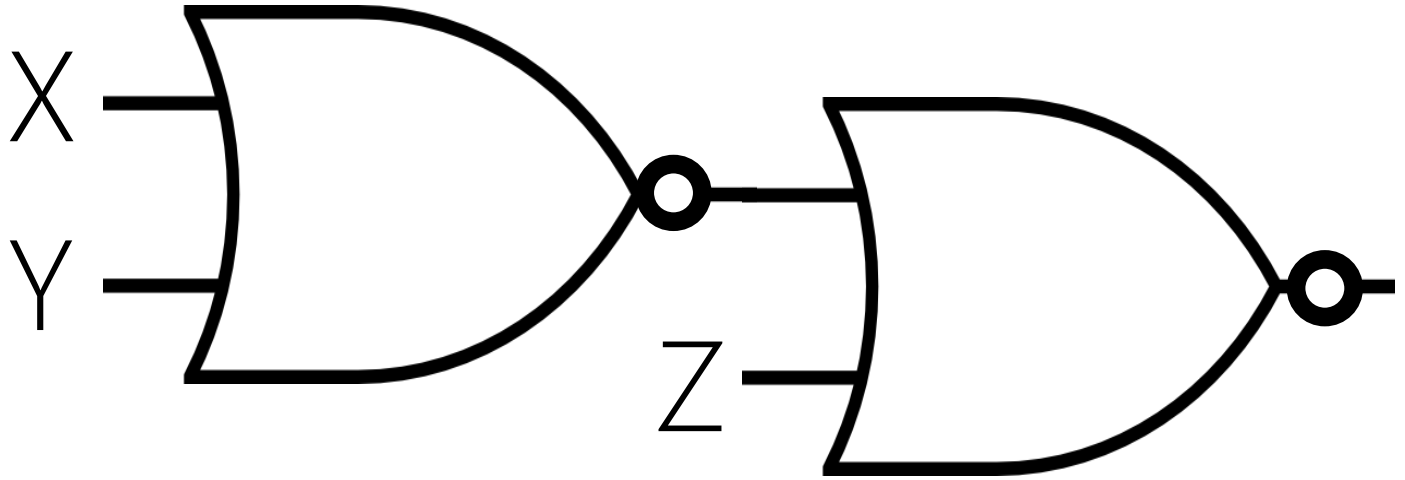


$F_1=(Z+(Y+X)')'$



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

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



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

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	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