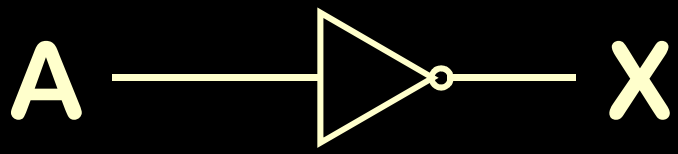
Digital Logic

How to do something useful with all those 1s and 0s

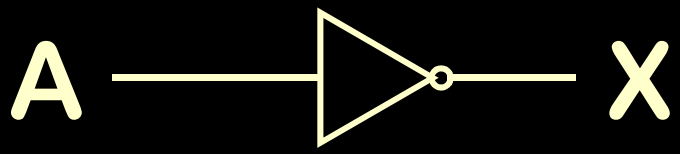
Processing 1s and 0s

- Logic “Gates” (electronics)
- Truth Tables (logic)
- Boolean Expressions (algebra)

NOT Gate
aka "Inverter"



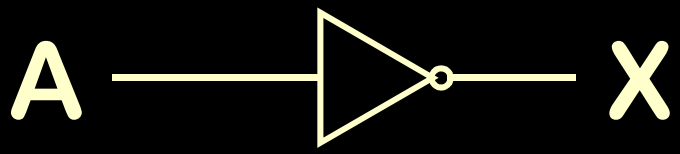
NOT Gate
aka "Inverter"



A	X
0	1
1	0

Truth Table

NOT Gate
aka "Inverter"



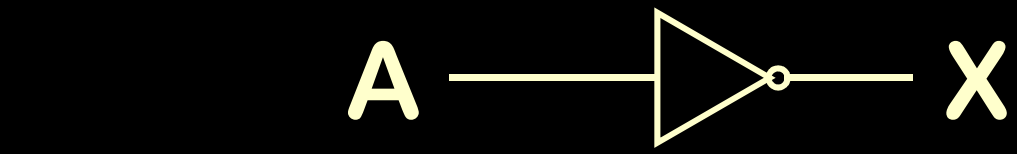
$X = \overline{A}$

A	X
0	1
1	0

Boolean Algebra

Truth Table

NOT Gate
aka "Inverter"



$X = \overline{A}$

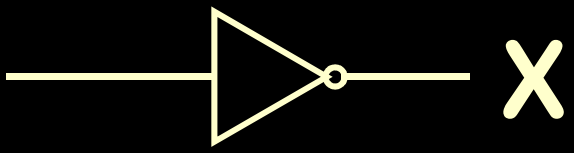
$X = A'$

Boolean Algebra

A	X
0	1
1	0

Truth Table

NOT Gate
aka "Inverter"



$X = \overline{A}$

X = A'

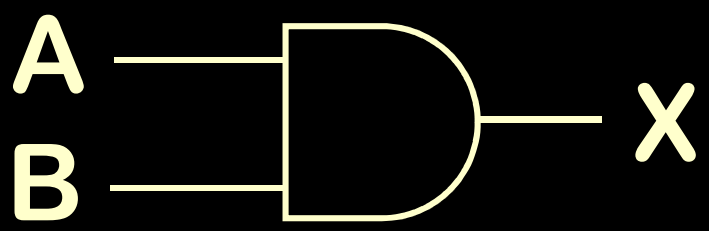
X = !A

Boolean Algebra

A	X
0	1
1	0

Truth Table

AND Gate

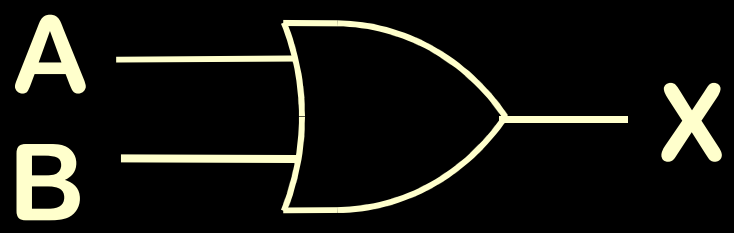


$X = A \cdot B$

$X = AB$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

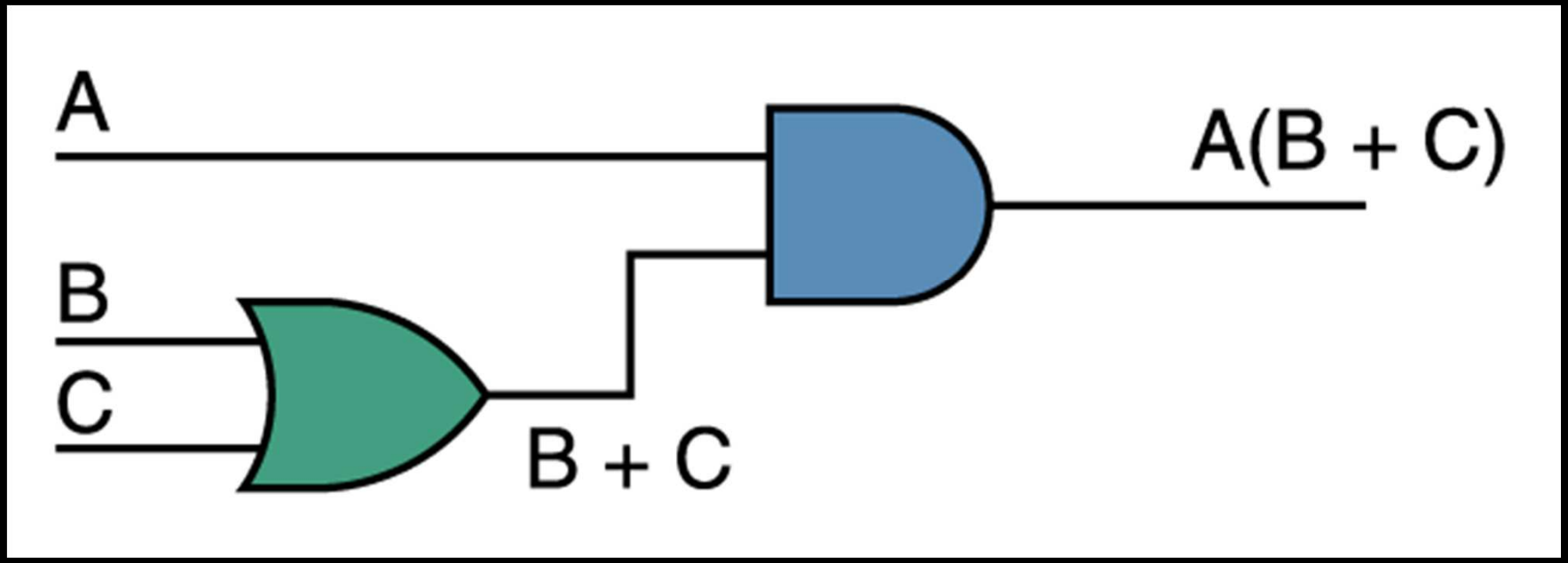


X = A+B

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Combinational Circuits

A(B + C)



Combinational Circuits

A(B + C)

A	B	C	B+C	A(B+C)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Combinational Circuits

A(B + C)

A	B	C	B+C	A(B+C)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Combinational Circuits

A(B + C)

A	B	C	B+C	A(B+C)
0	0	0	0	
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

Combinational Circuits

A(B + C)

A	B	C	B+C	A(B+C)
0	0	0	0	
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

Combinational Circuits

A(B + C)

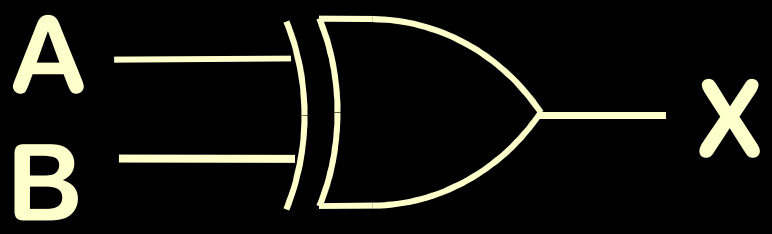
A	B	C	B+C	A(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Combinational Circuits

A(B + C)

A	B	C	B+C	A(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

XOR Gate



$X = A \oplus B$

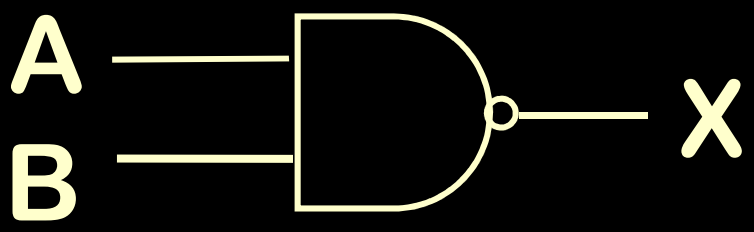
MS Office: Insert Symbol
Google: Insert →Special Characters

$X = A \wedge B$

On a standard keyboard

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

NAND Gate



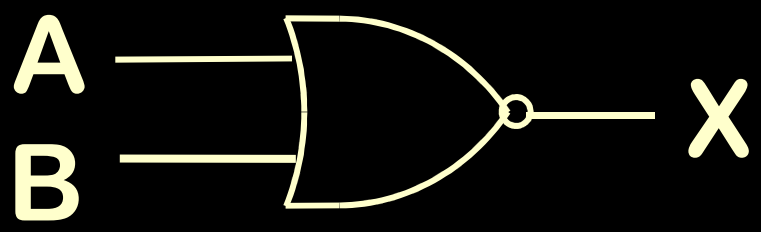
$$X = \overline{AB}$$

X = (AB)'

X = !(AB)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate



$$X = \overline{A+B}$$

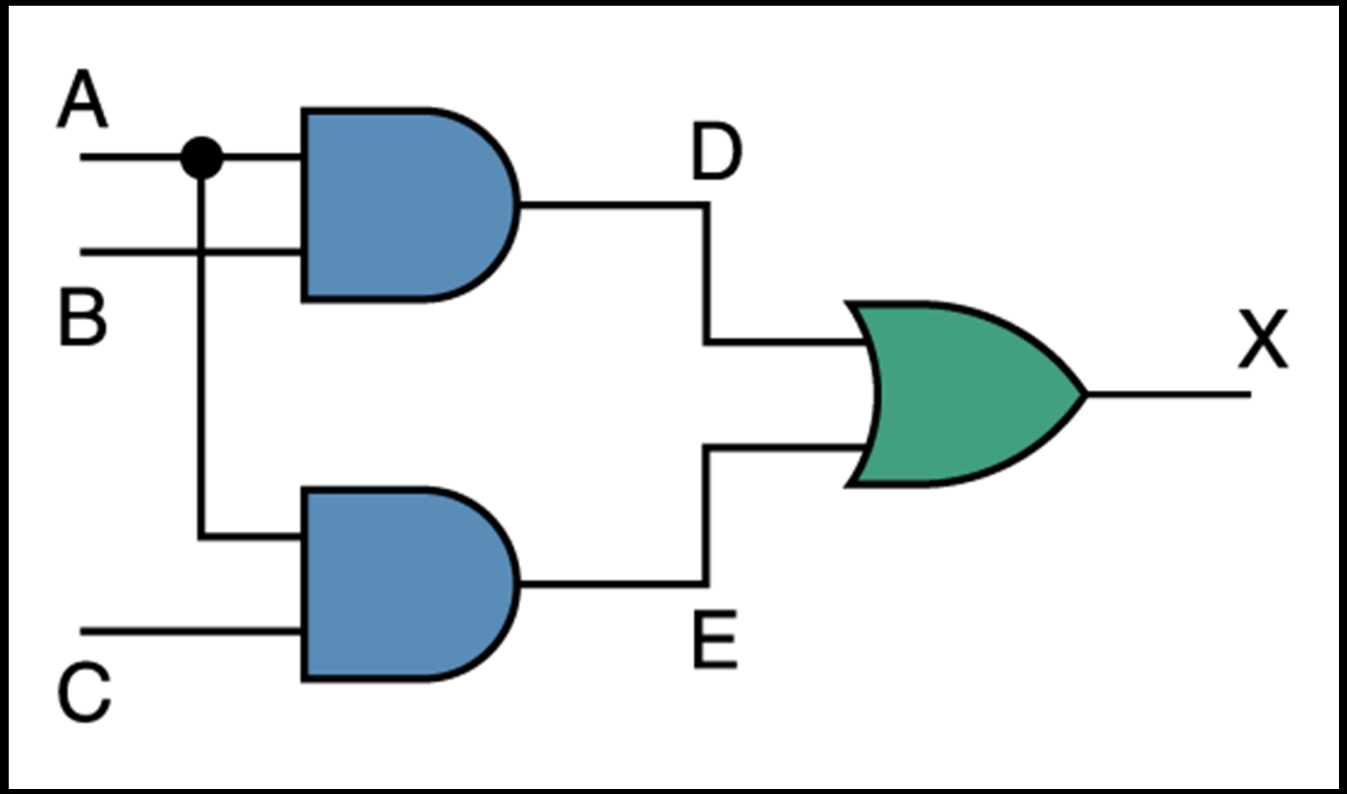
X = (A+B)'

X = !(A+B)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Combinational Circuits

(AB + AC)



Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
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	1		
1	1	1	1		

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
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	1		
1	1	1	1		

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
0	0	0	0	0	
0	0	1	0	0	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	0	0	
1	0	1	0	1	
1	1	0	1	0	
1	1	1	1	1	

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
0	0	0	0	0	
0	0	1	0	0	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	0	0	
1	0	1	0	1	
1	1	0	1	0	
1	1	1	1	1	

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Combinational Circuits

(AB + AC)

A	B	C	AB	AC	AB+AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

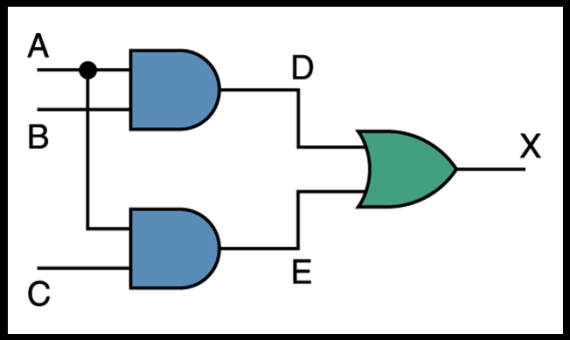
Combinational Circuits

(AB + AC) = A(B + C)

A	B	C	AB	AC	AB+AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

Equivalent Circuits

(AB + AC) = A(B + C)



=