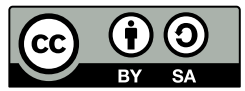
Paskaita bus įrašyta
This lecture will be recorded

Transistors and logic gates

Saulius Gražulis

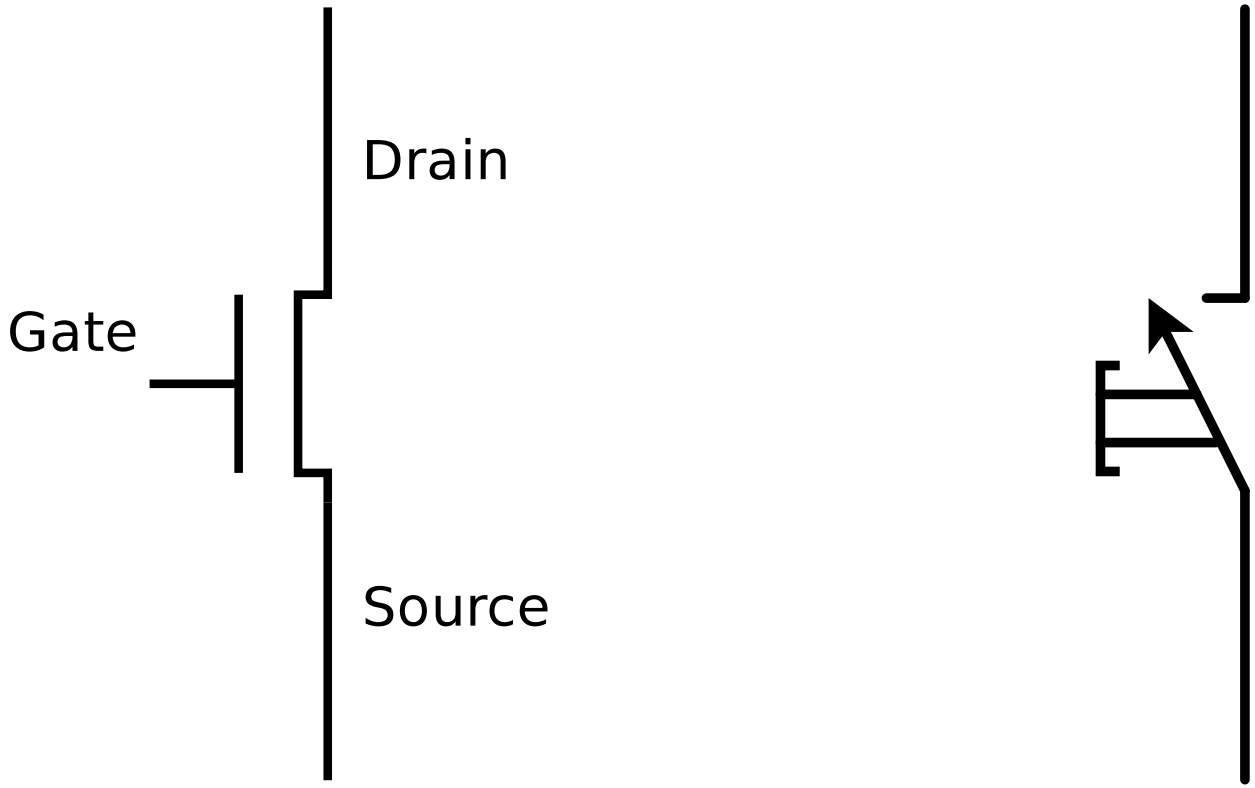
Vilnius, 2020

Vilnius University, Faculty of Mathematics and Informatics
Institute of Informatics



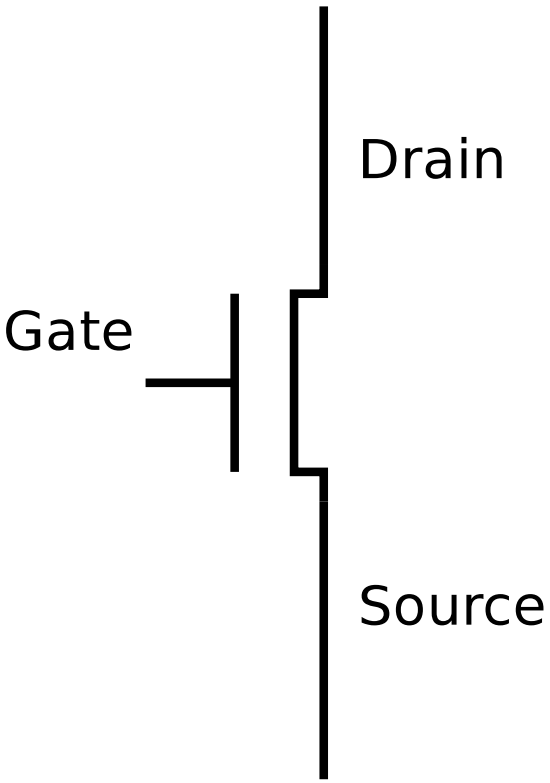
Transistor, a controlled electric switch

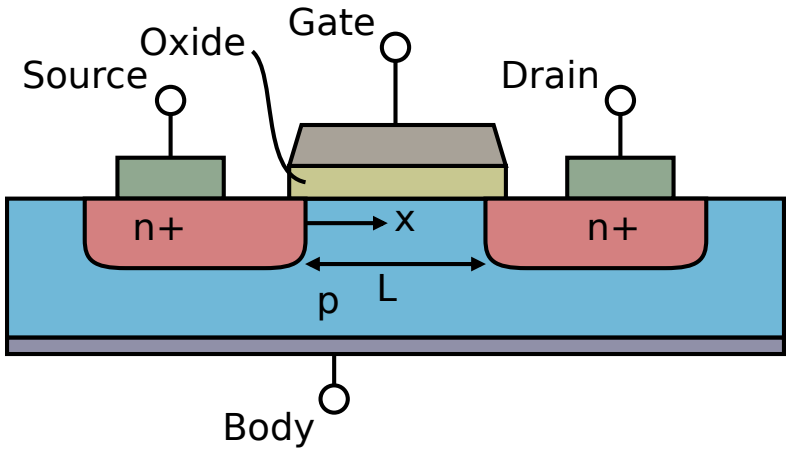
FET – Field-effect transistor
MOSFET – Metal–oxide–semiconductor field-effect transistor



Transistor, a controlled electric switch

FET – Field-effect transistor
MOSFET – Metal–oxide–semiconductor field-effect transistor

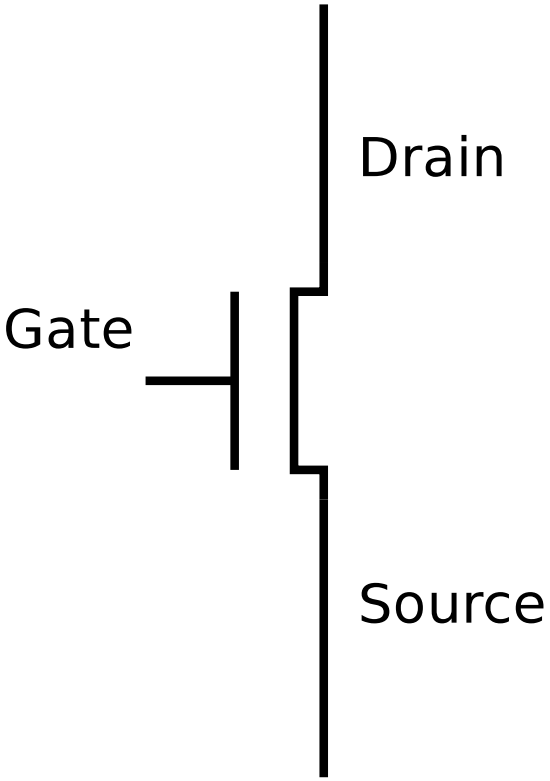


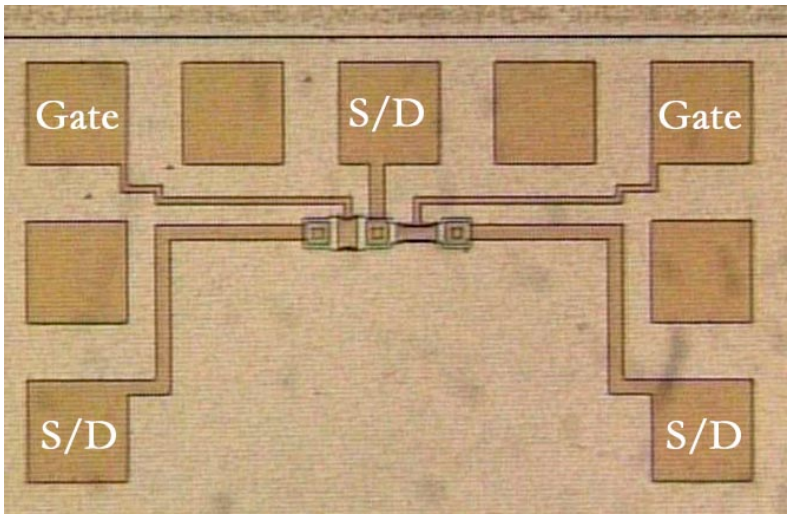


Cyril Buttay [CC BY-SA 3.0]

Transistor, a controlled electric switch

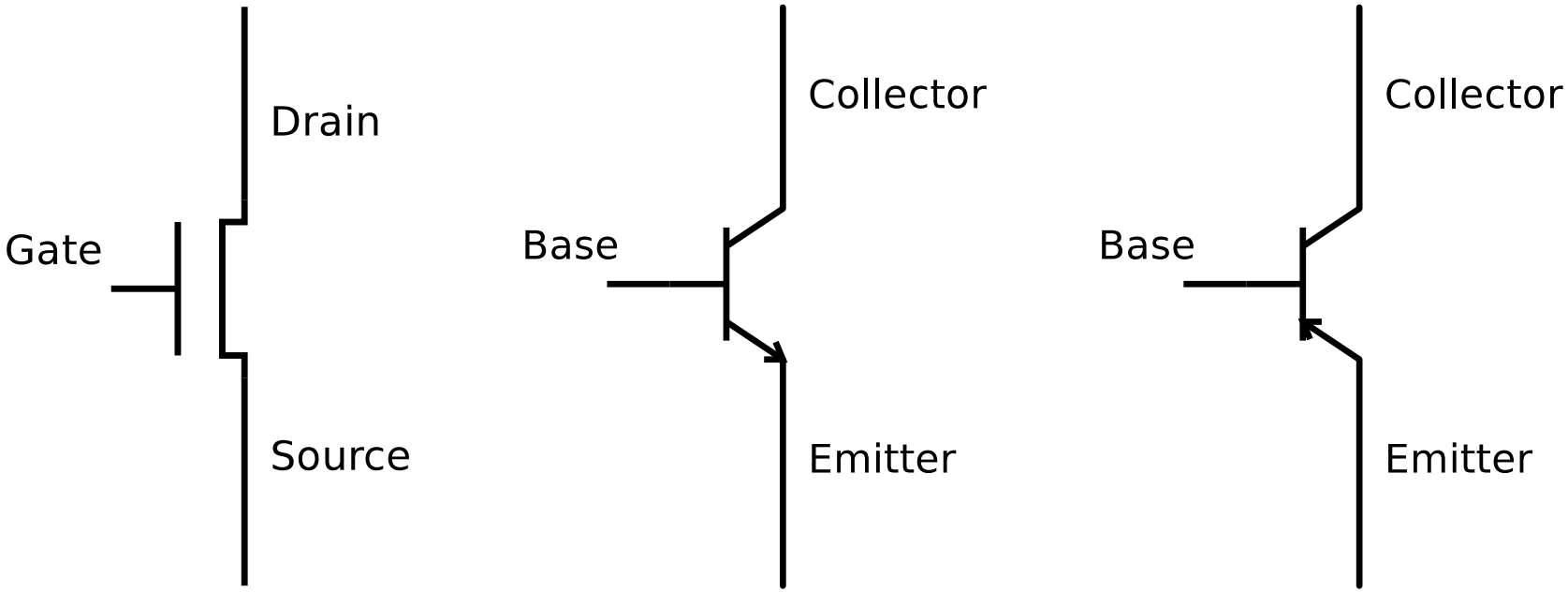
FET – Field-effect transistor
MOSFET – Metal–oxide–semiconductor field-effect transistor



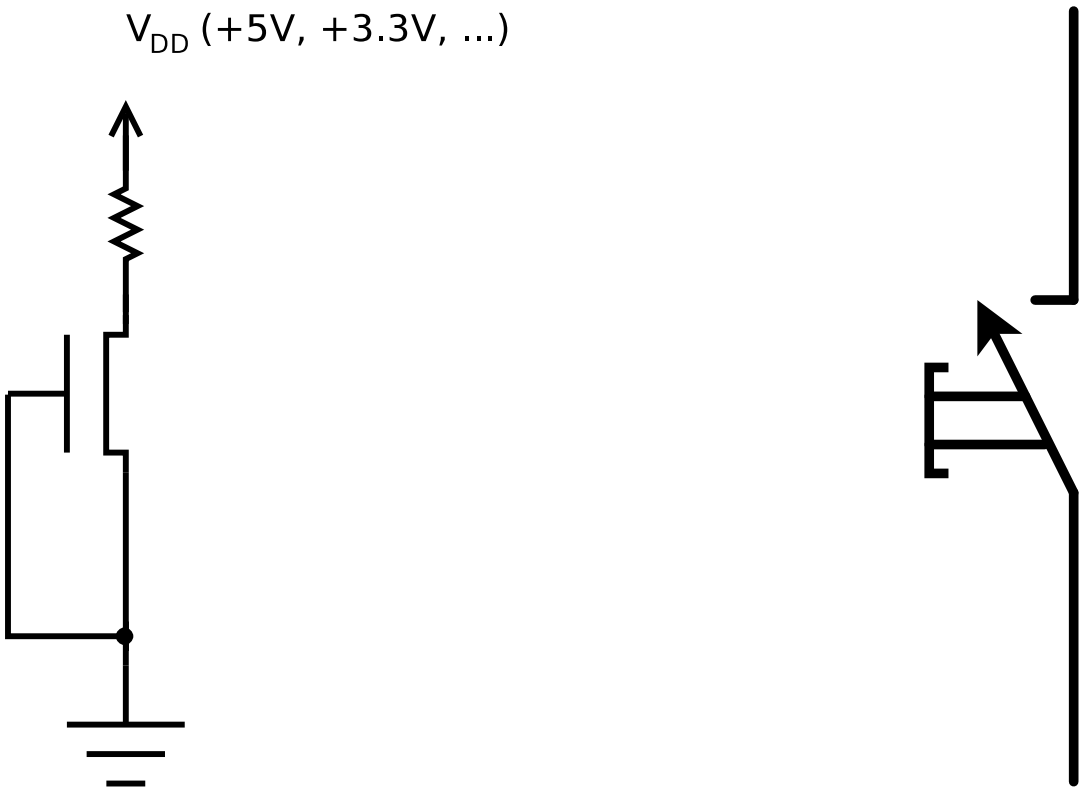


Dick Lyon [Public domain]

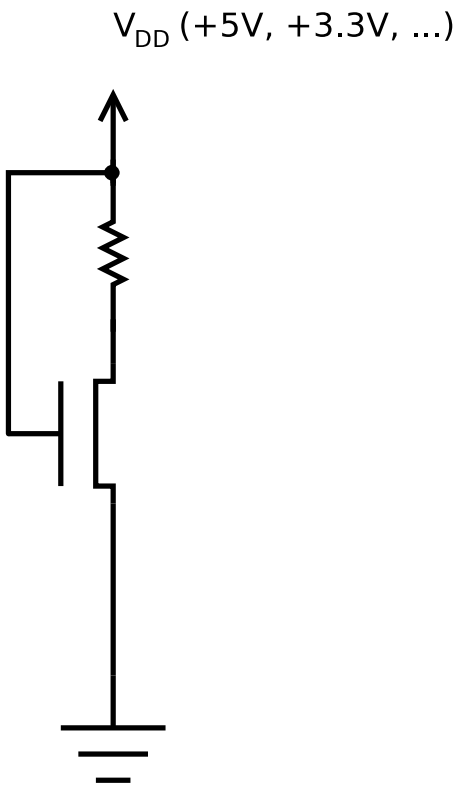
FET vs bipolar transistor

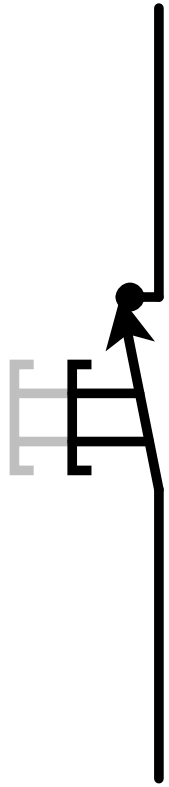


Properties of a transistor

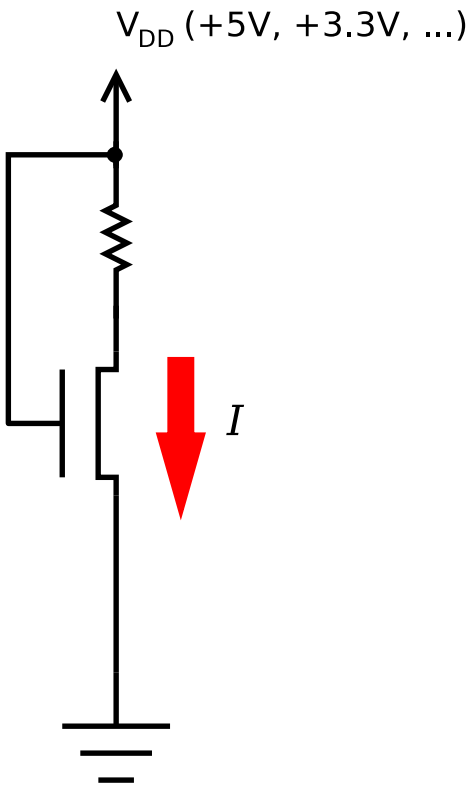


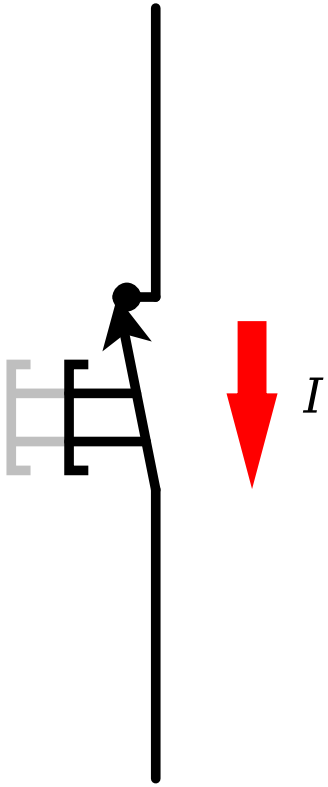
Properties of a transistor



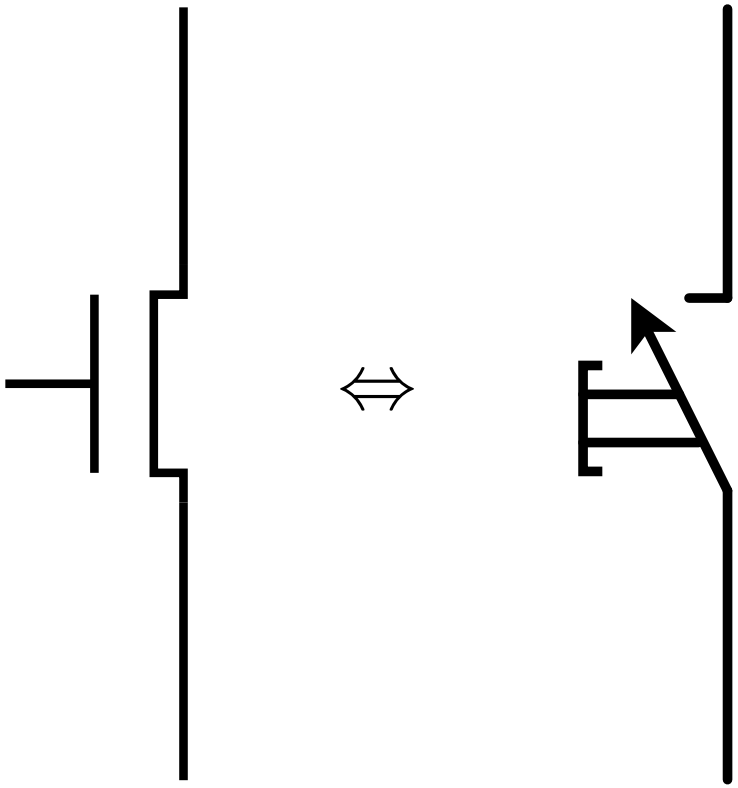


Properties of a transistor

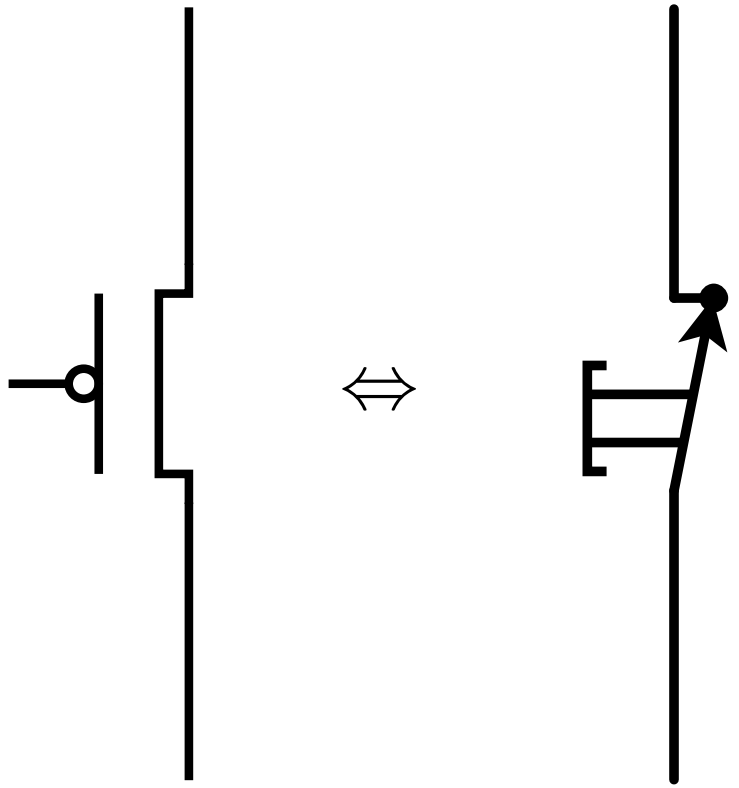




Two types of transistors

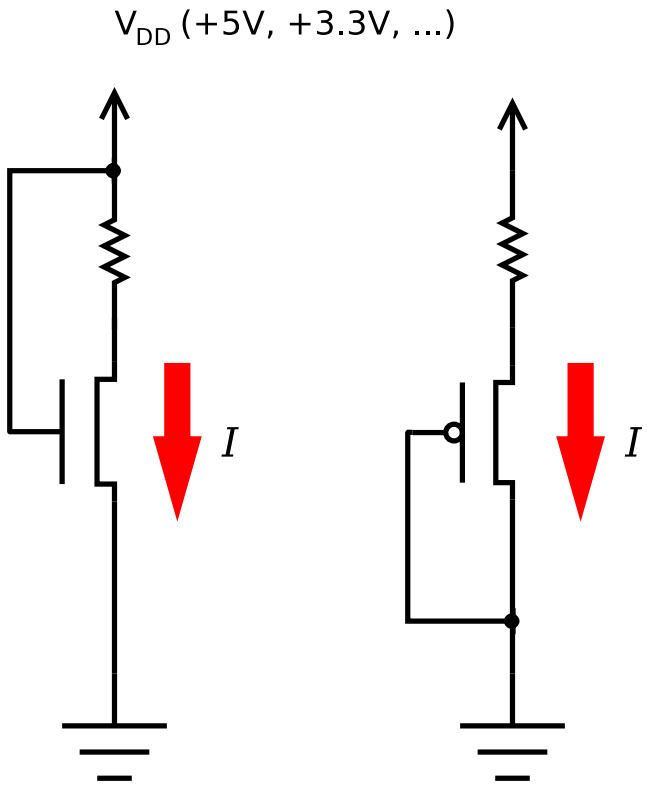


n-type transistor

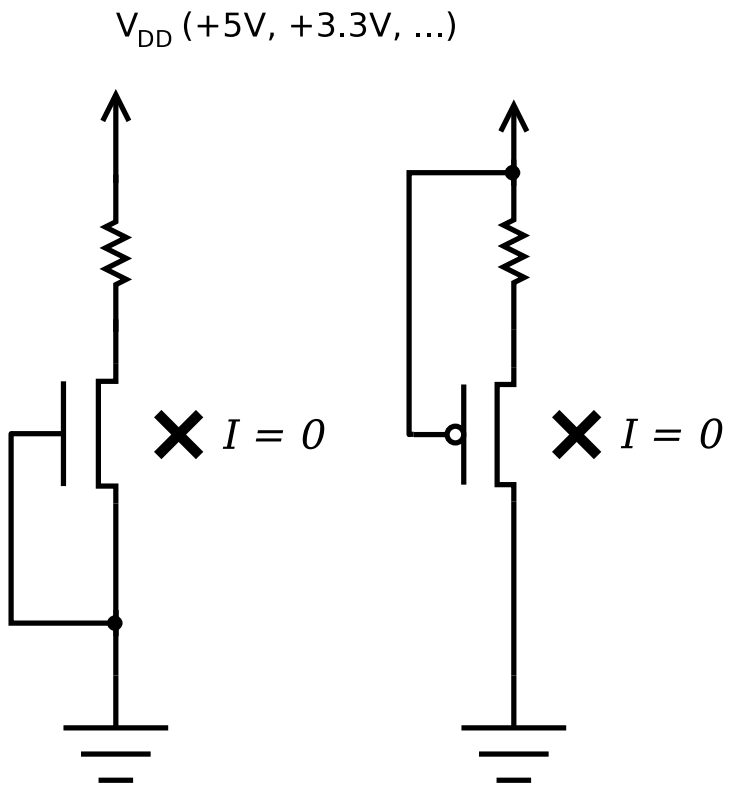


p-type transistor

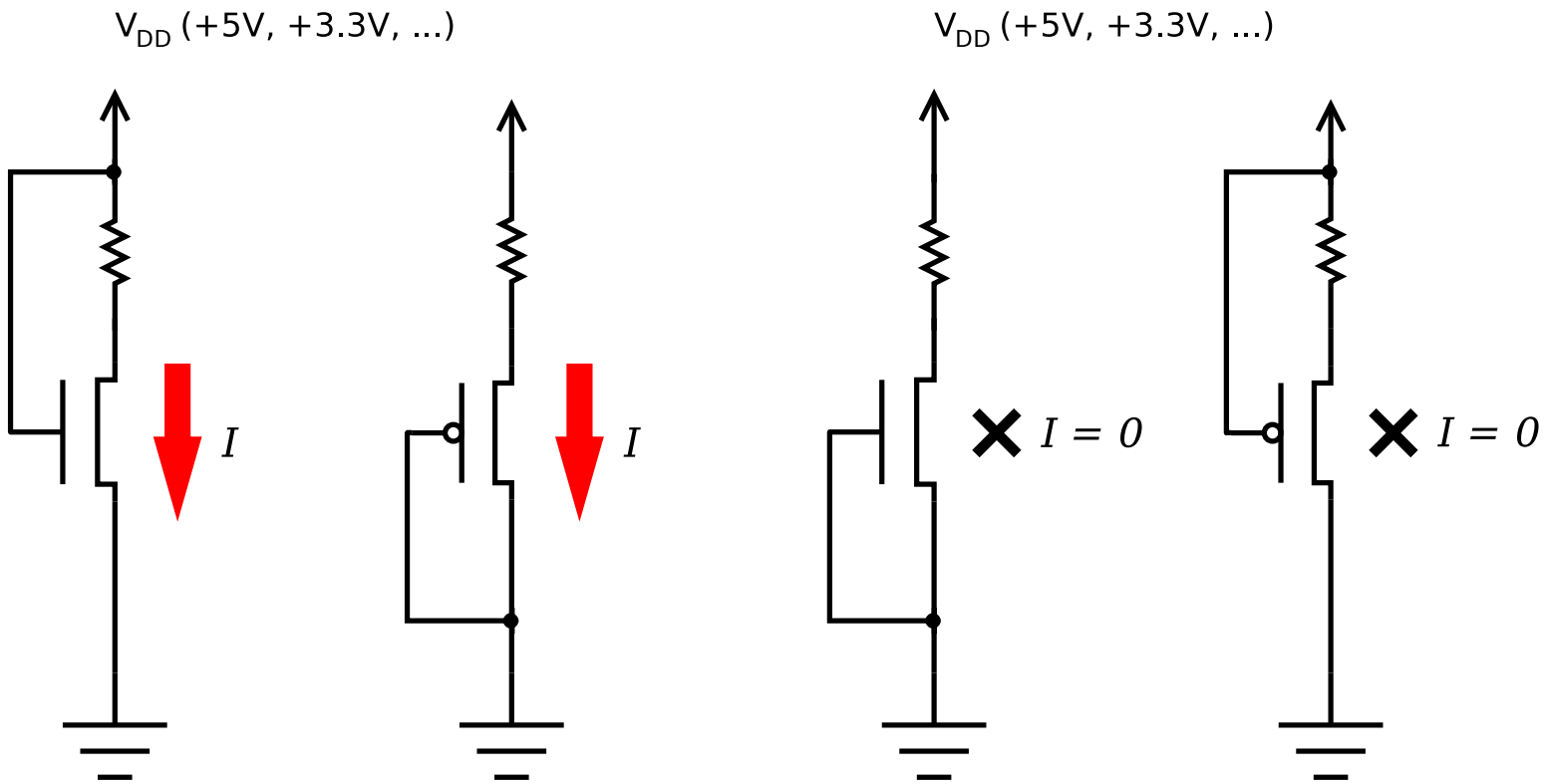
Comparison of n-type and p-type transistors



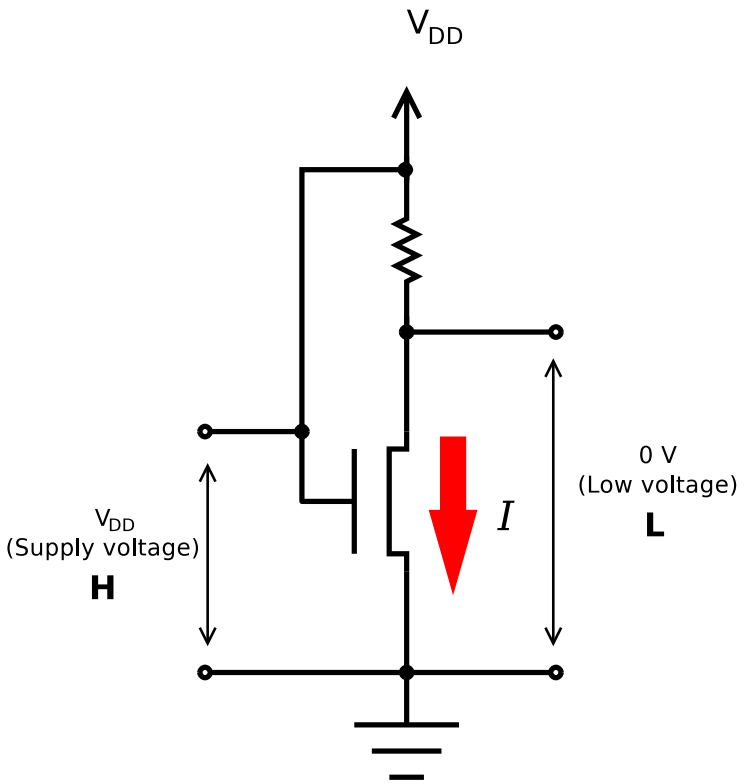
Comparison of n-type and p-type transistors

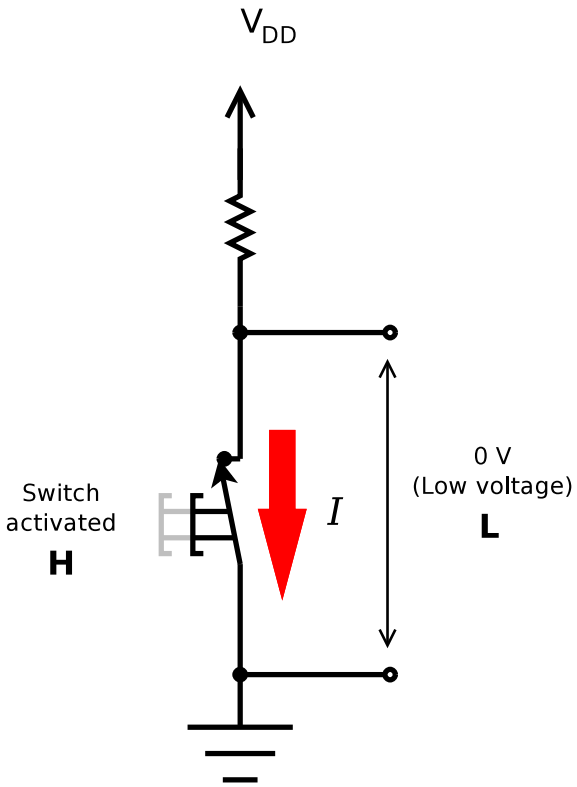


An overview n-type and p-type transistors

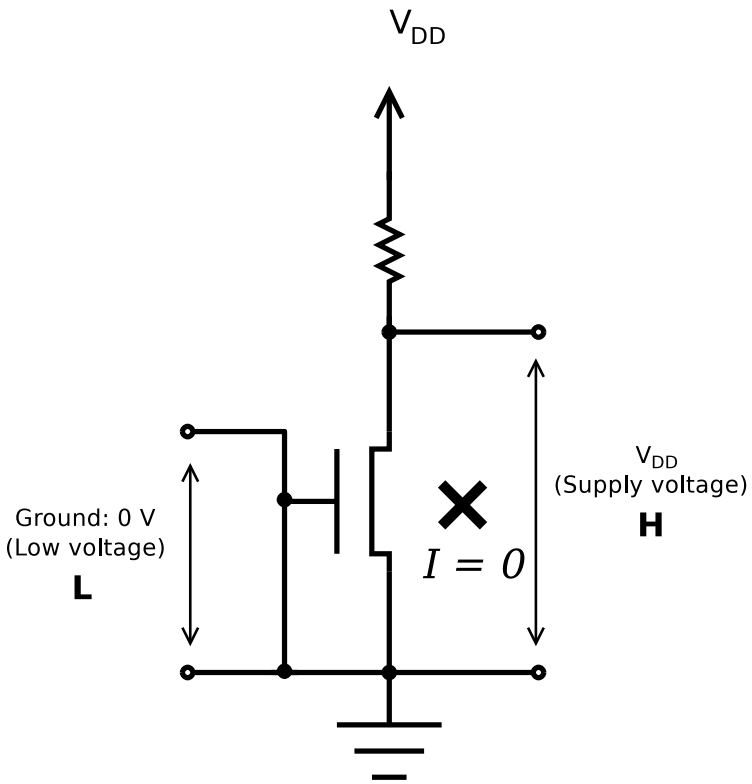


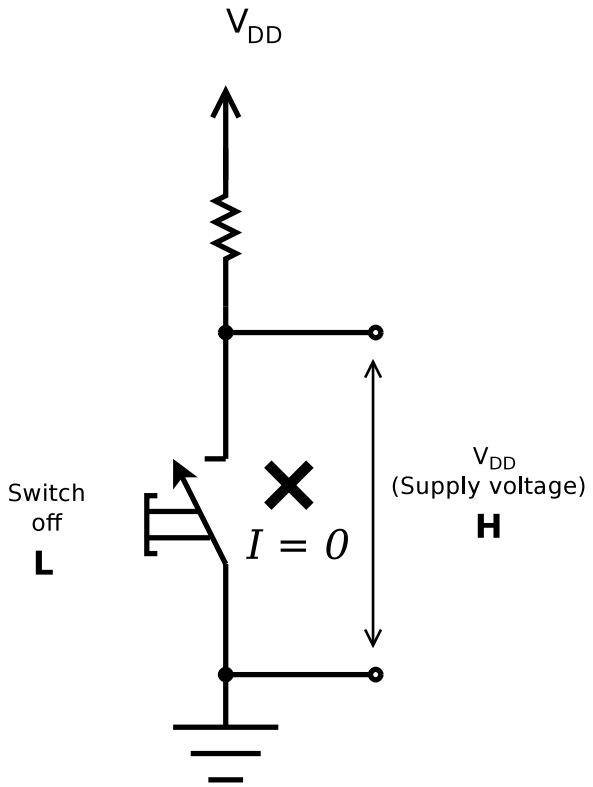
Transistor switch voltages



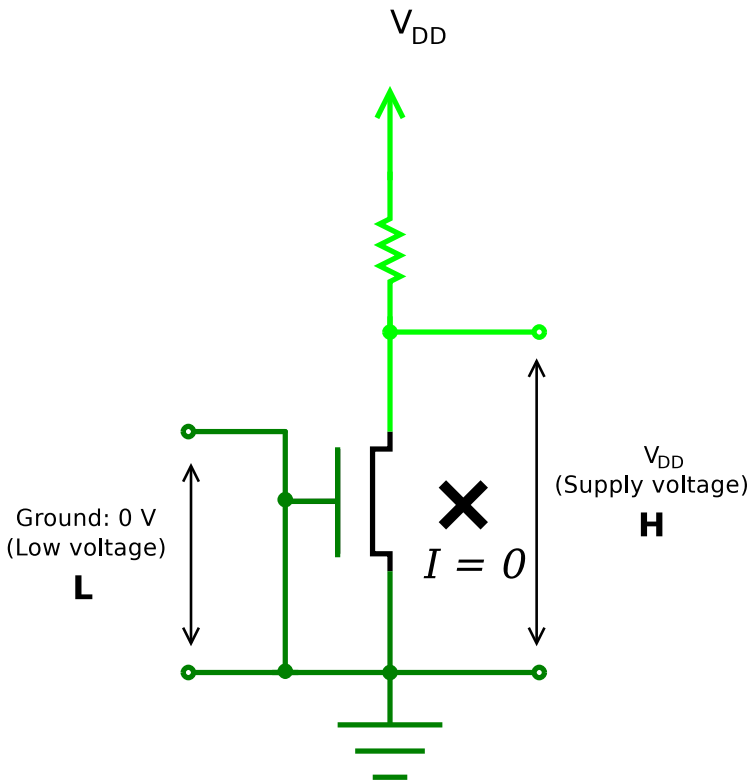


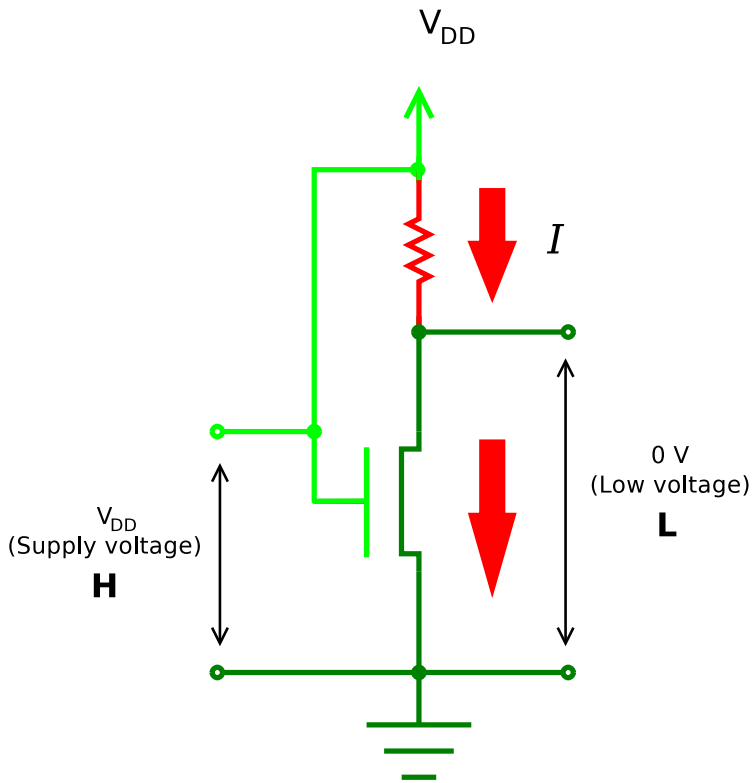
Transistor switch voltages





Opposite voltages of a transistor switch





Inverter (the logic function NOT)

L: Ground voltage ($\approx$ 0V)
H: Supply voltage (5V, 3.3V, ...)

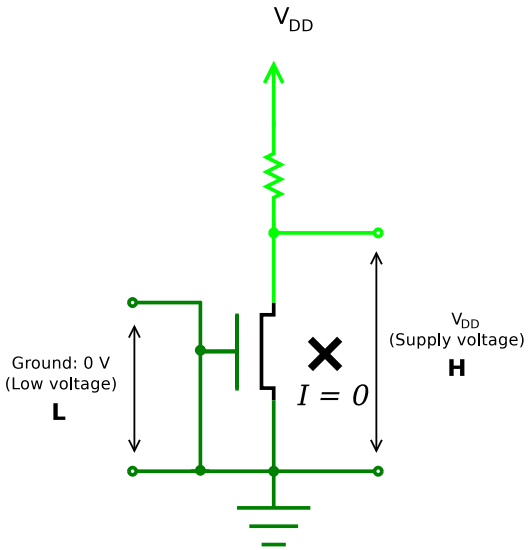
L: 0
H: 1

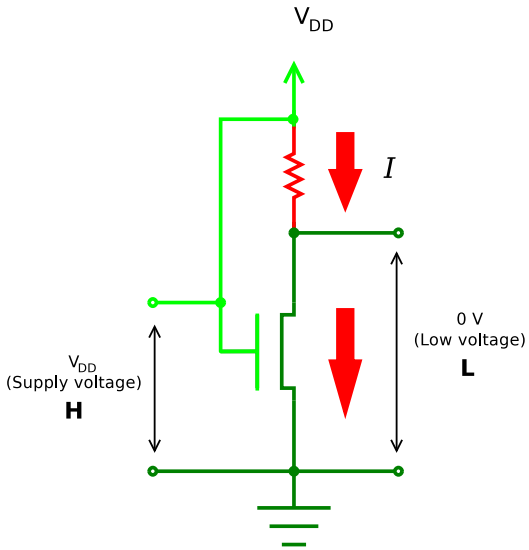
L: False
H: True

Input	Output
L	H
H	L

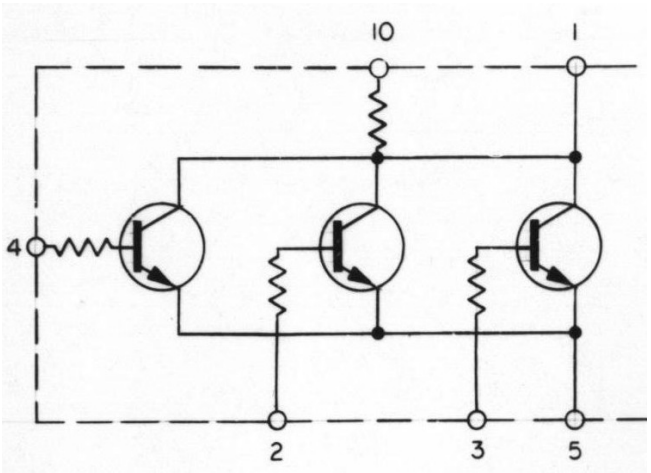
Input	Output
0	1
1	0

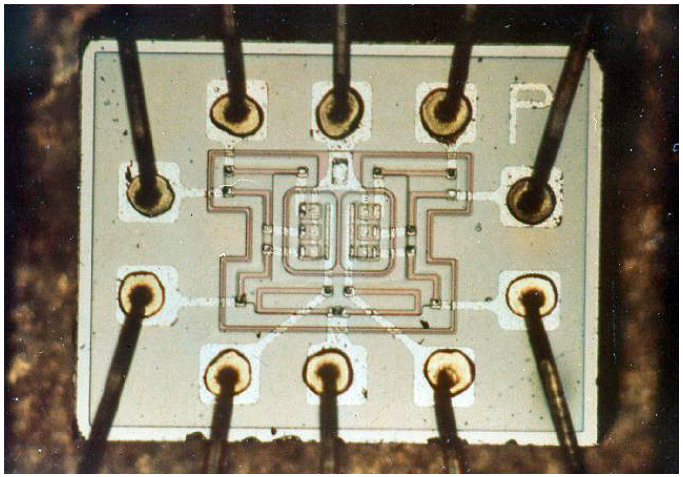
Input	Output
False	True
True	False

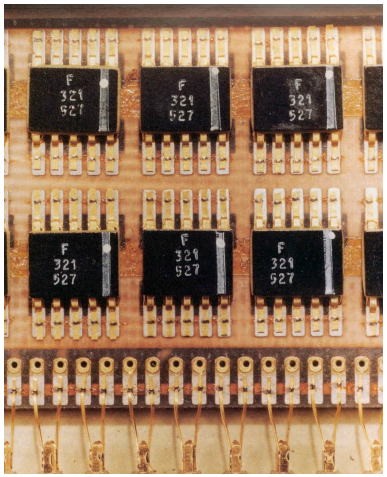


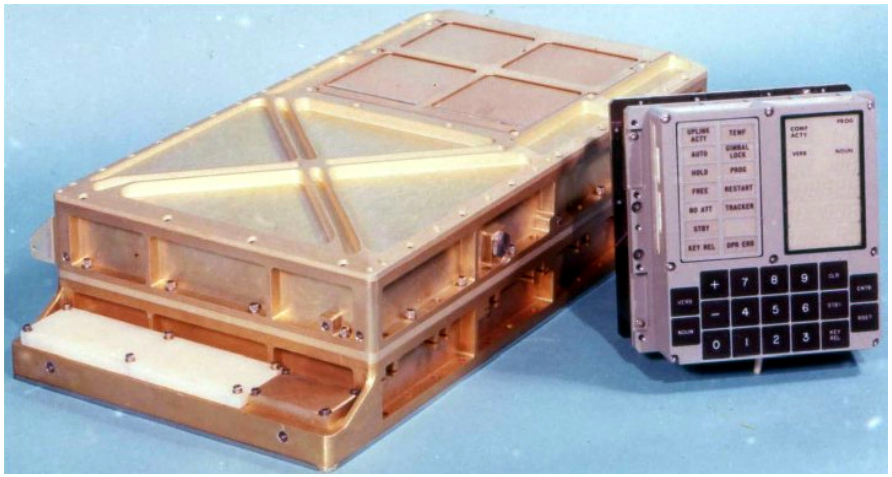


Historic NOR gates – Apollo guidance computer



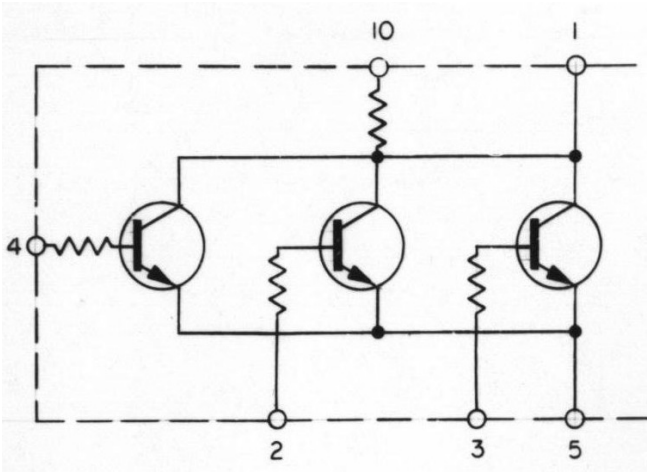


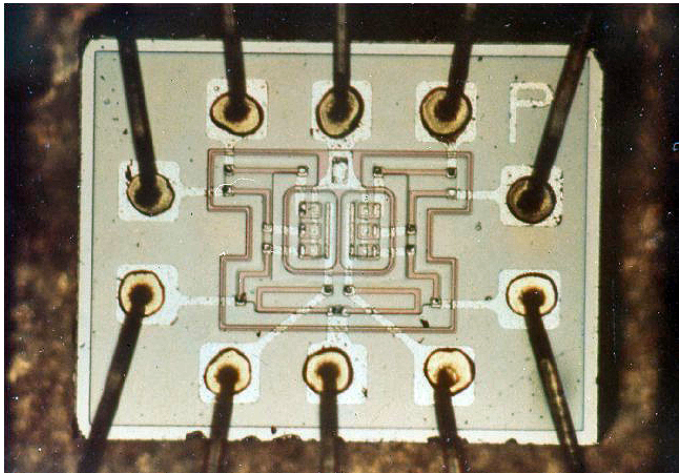


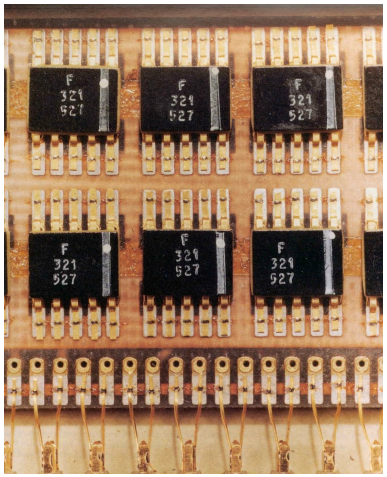


All AGC images: Wikipedia. Apollo guidance computer, (viewed 2020-08-10). All images by NASA, public domain.

Historic NOR gates – Apollo guidance computer

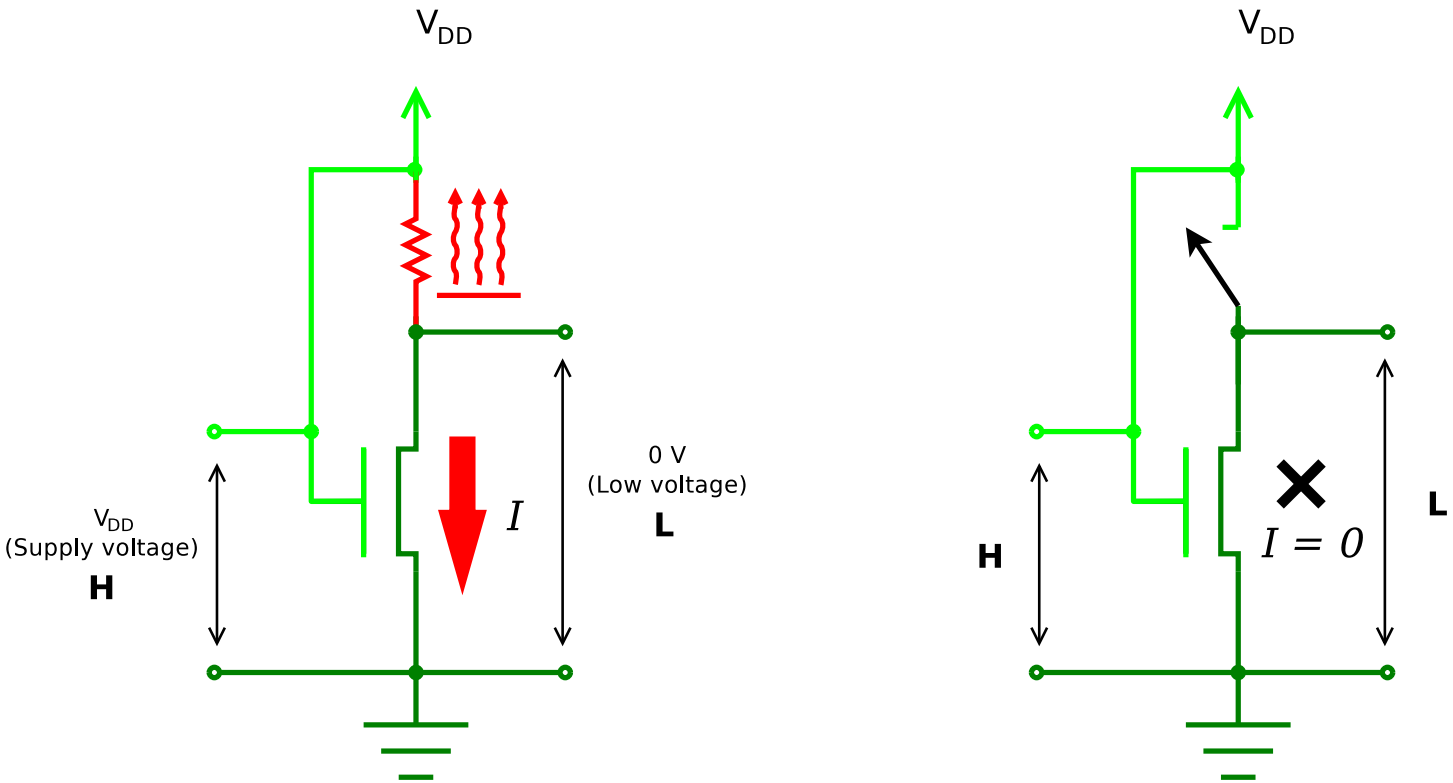




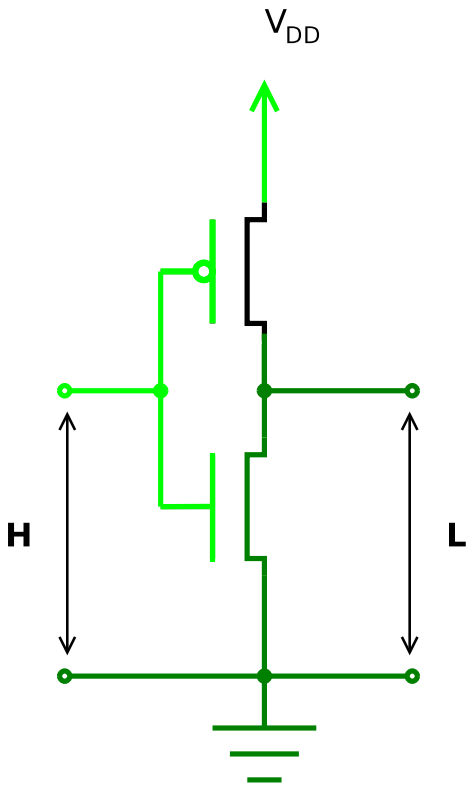


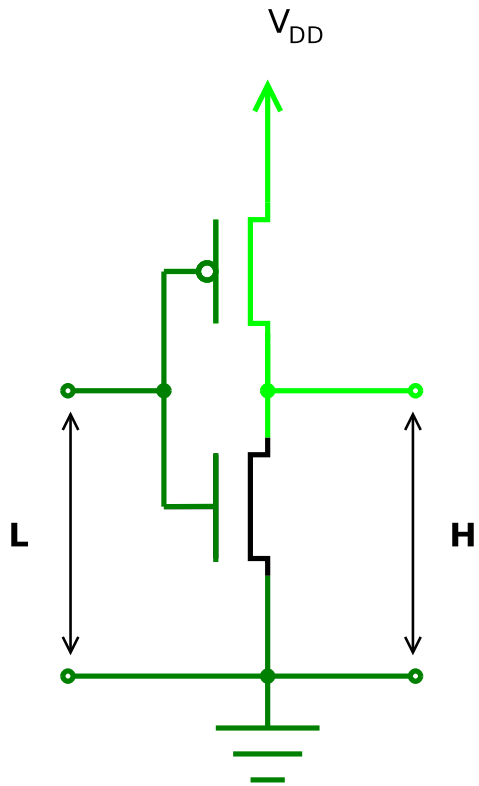
Lunar Landing Module image: by Neil Armstrong (viewed 2020-08-10). [Public domain]

Power dissipation



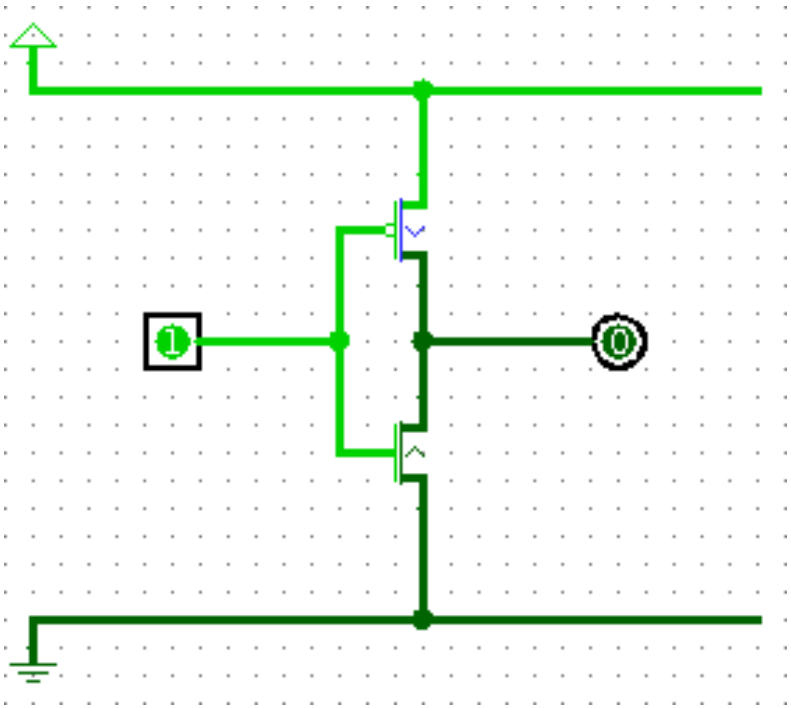
CMOS NOT gate

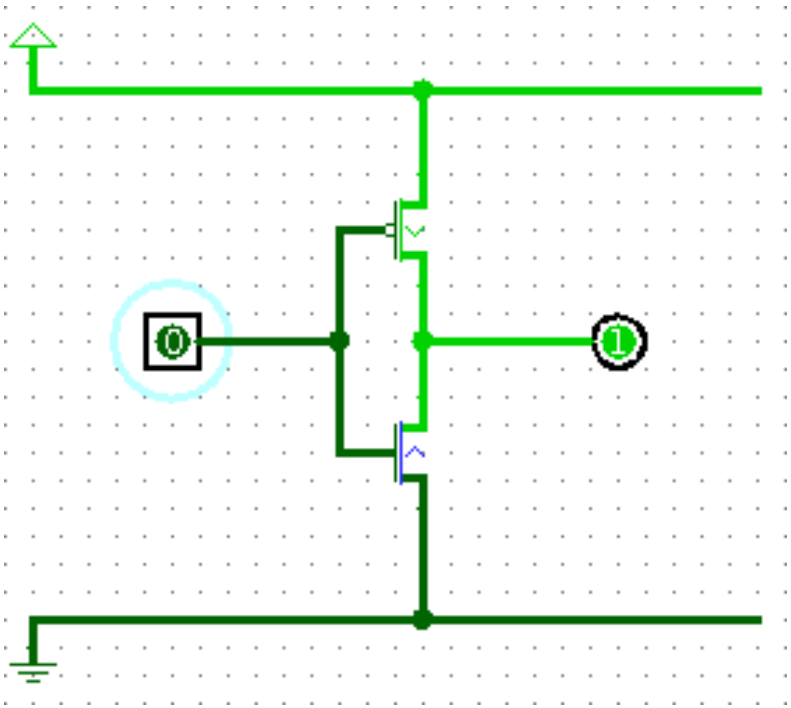




Logisim simulated NOT gate

Logisim is a very nice program for designing and simulating digital logic circuits.



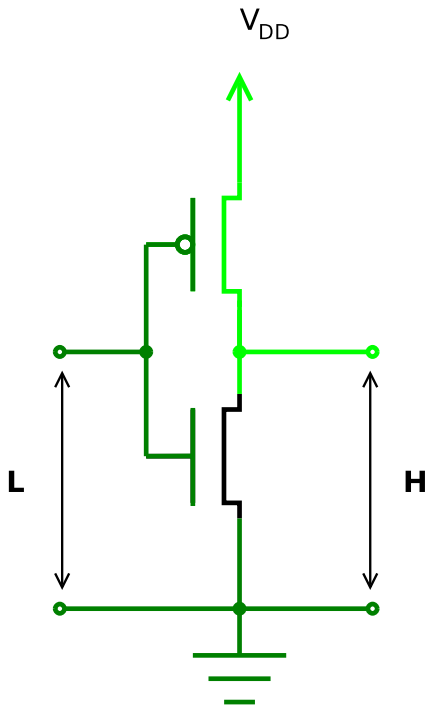


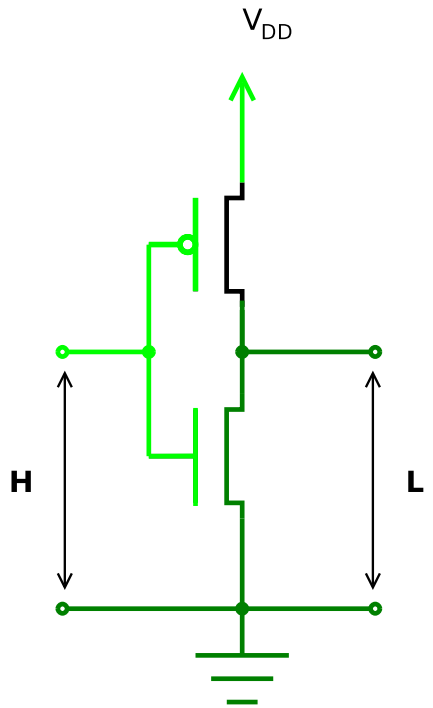
NOT gate revisited

Input	Output
L	H
H	L

Input	Output
0	1
1	0

Input	Output
False	True
True	False



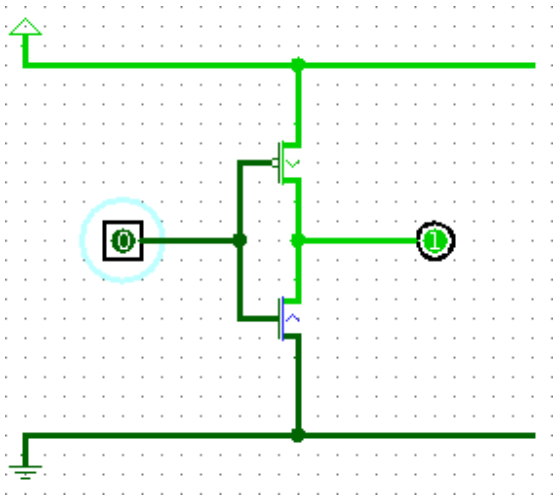


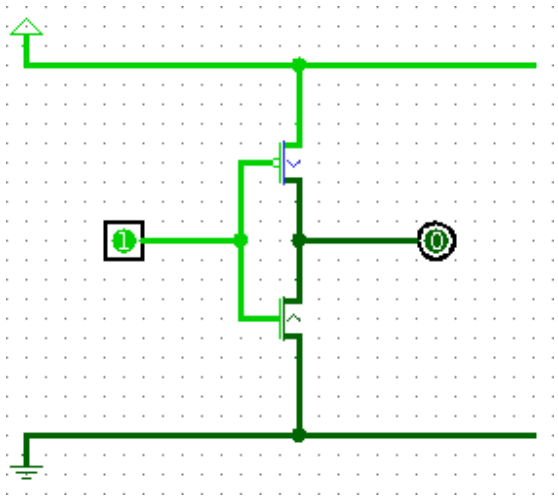
NOT gate revisited

Input	Output
L	H
H	L

Input	Output
0	1
1	0

Input	Output
False	True
True	False



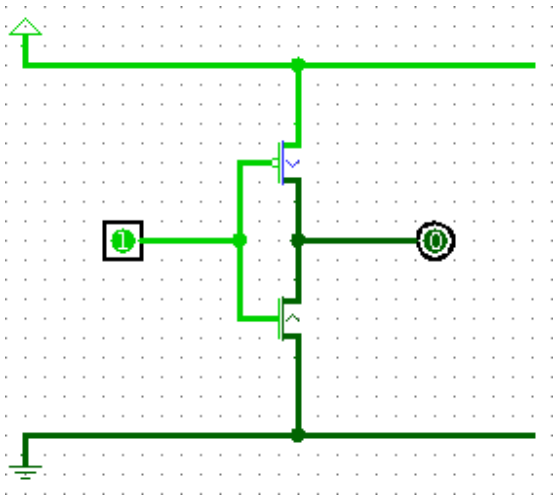


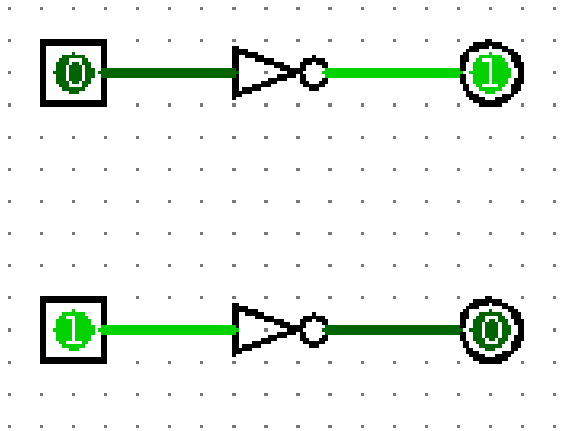
NOT gate revisited

Input	Output
L	H
H	L

Input	Output
0	1
1	0

Input	Output
False	True
True	False



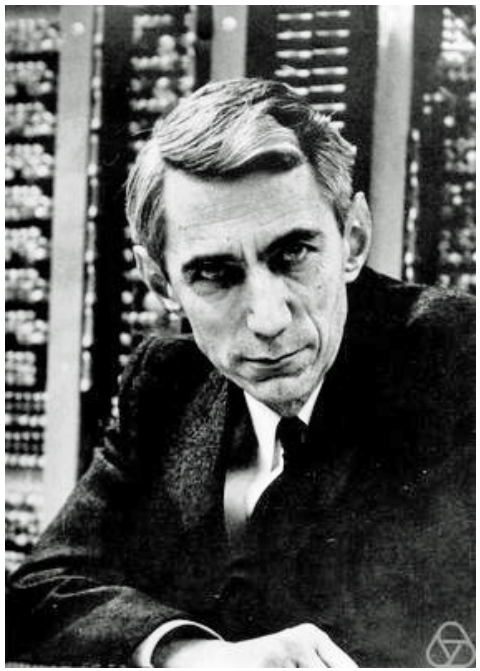


Boolean functions

Based on the two-element Boolean algebra, or switching algebra.

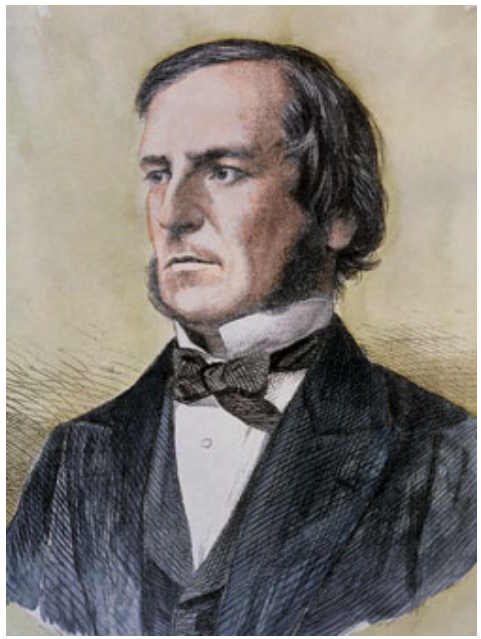
$$B = \{0, 1\}$$
$$B \times \cdots \times B \mapsto B$$

Augustus De Morgan



Claude Shannon

Photo by Konrad Jacobs, reused with permission



George Boole

Number of one-input functions

Input	Output
0	1
1	0

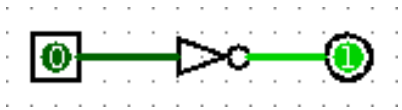
2 positions $\Rightarrow 2^2 = 4$ combinations

Input	Output
0	0
1	1

identity $f(x) = x$

Input	Output
0	1
1	0

NOT $f(x) = \overline{x}$

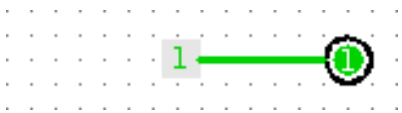


Input	Output
0	0
1	0

constant 0 $f(x) = 0$

Input	Output
0	1
1	1

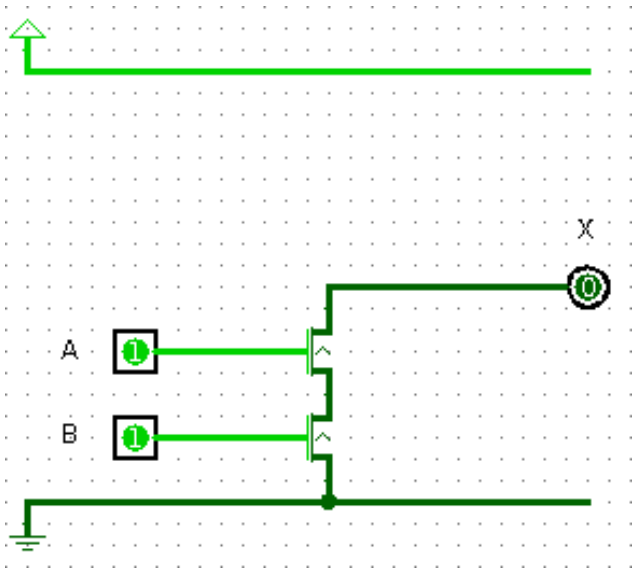
constant 1 $f(x) = 1$



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

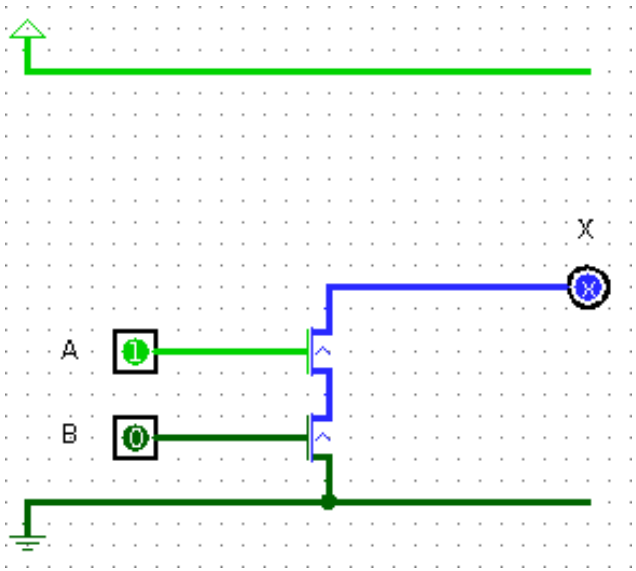
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

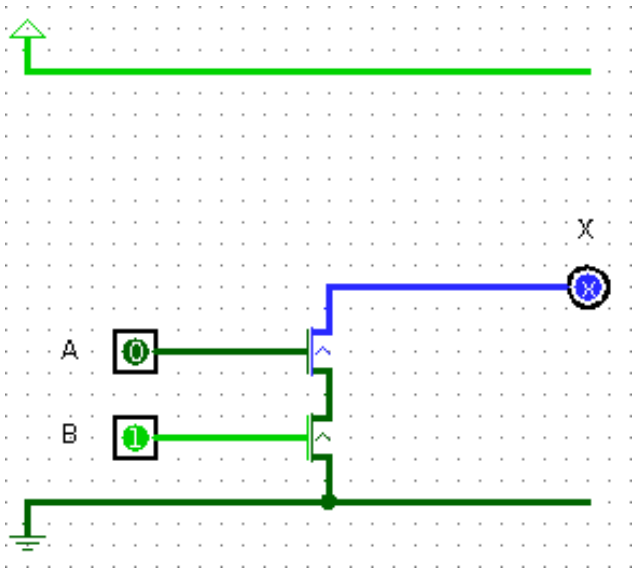
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

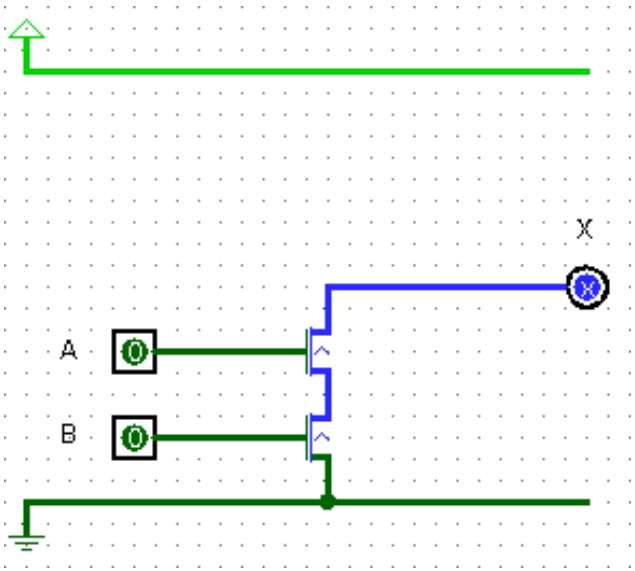
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

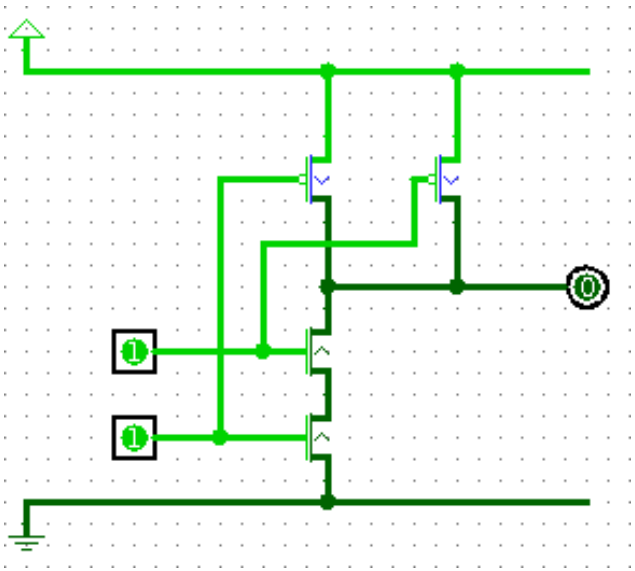
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

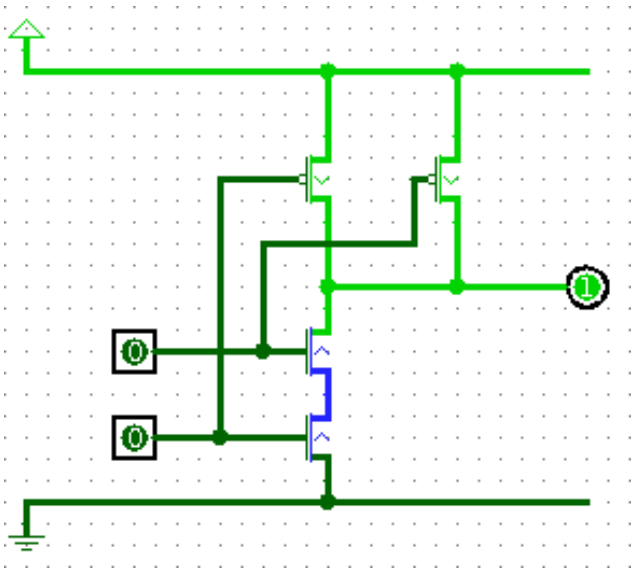
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

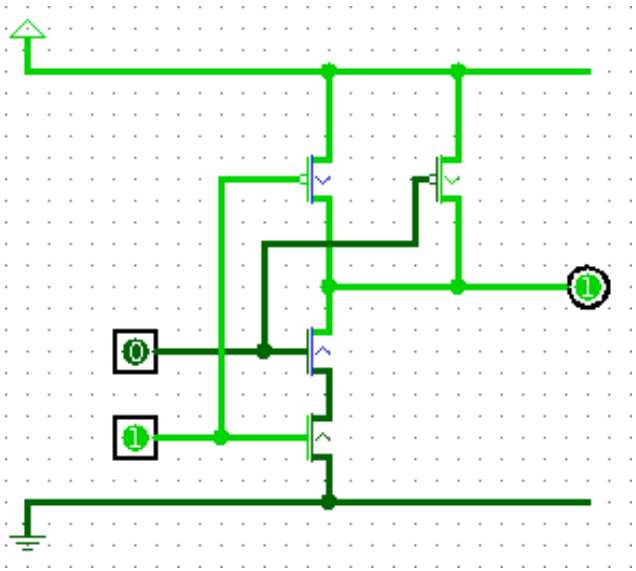
A	**B**	**X**
0	**0**	**1**
0	1	1
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

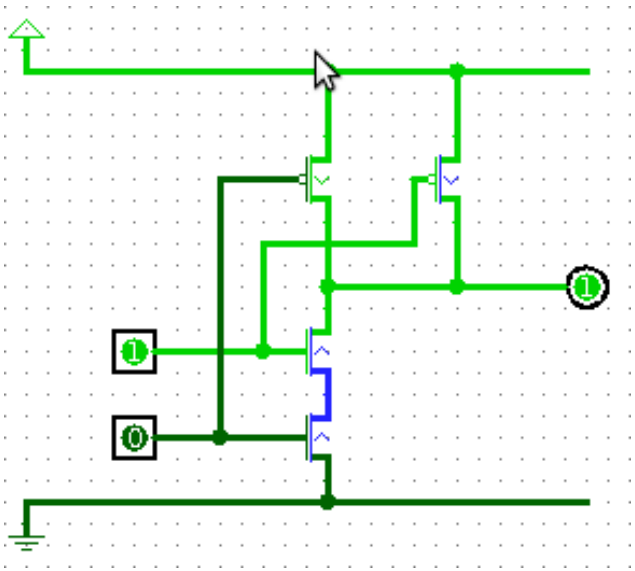
A	B	X
0	0	1
0	**1**	**1**
1	0	1
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

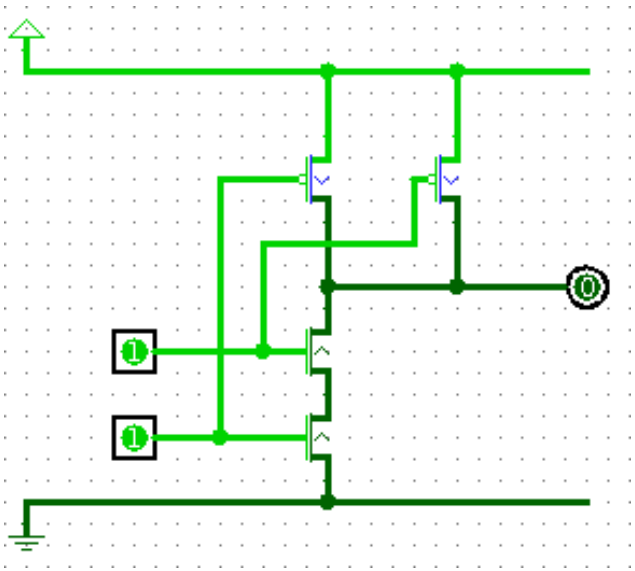
A	B	X
0	0	1
0	1	1
1	**0**	**1**
1	1	0



Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

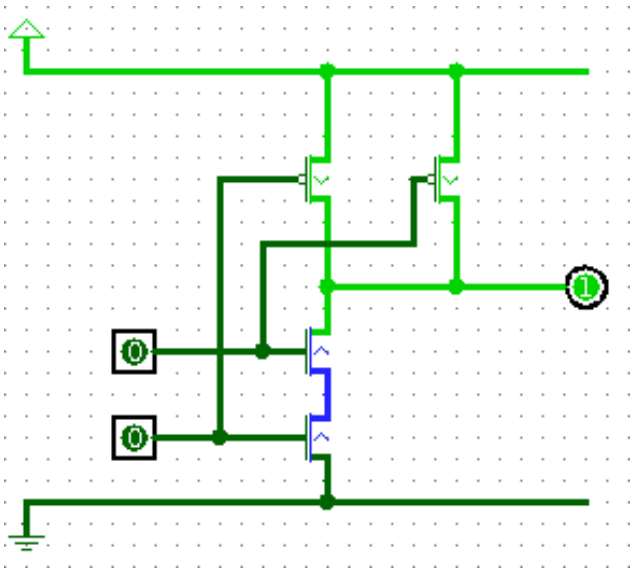
A	B	X
0	0	1
0	1	1
1	0	1
1	**1**	**0**

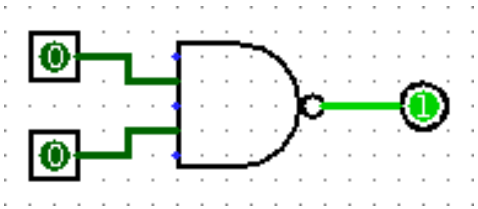


Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

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

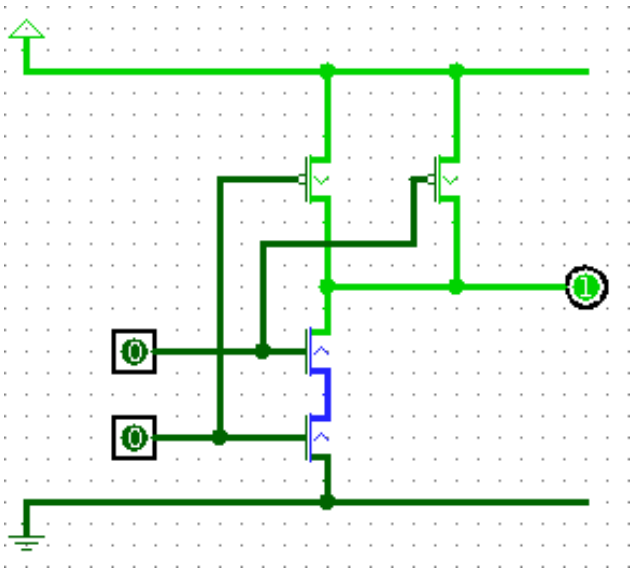


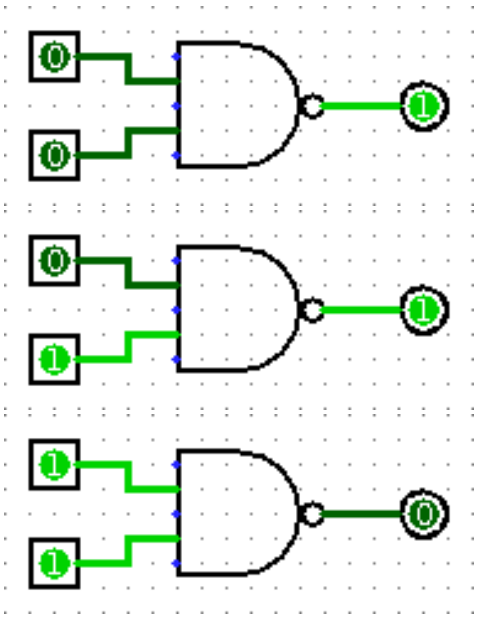


Logic NAND

$$X = f(A, B) = \overline{A \cdot B} = \overline{AB}$$

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

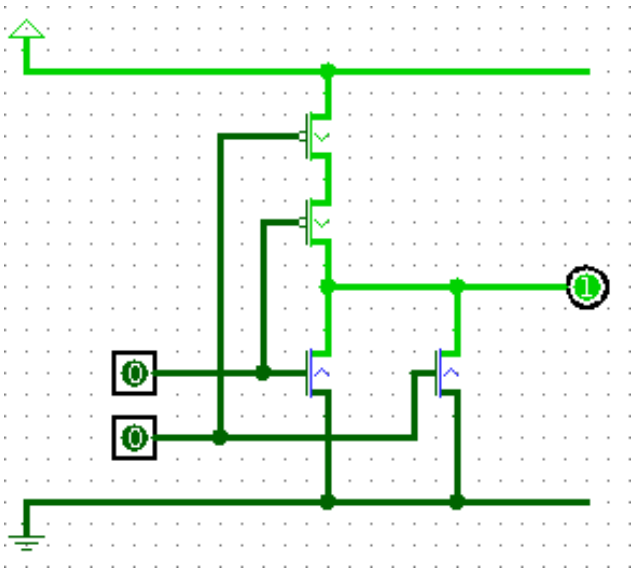




Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

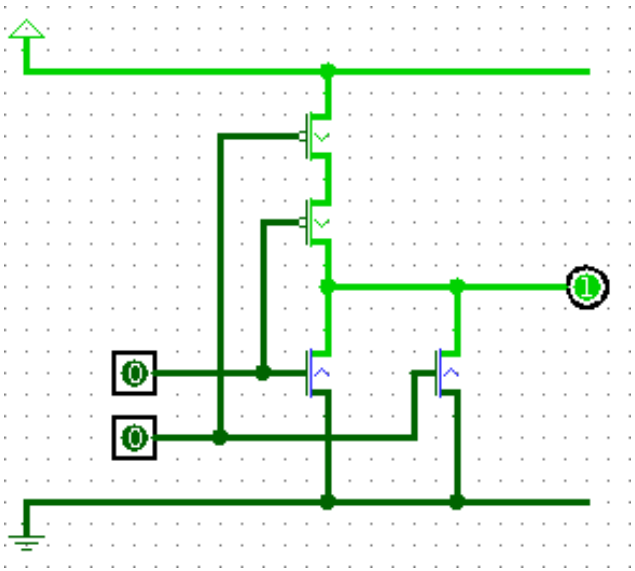
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

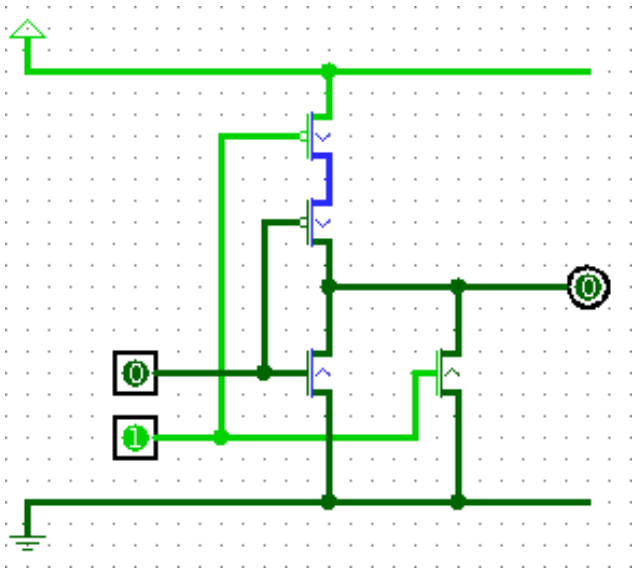
A	B	X
0	**0**	**1**
0	1	0
1	0	0
1	1	0



Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

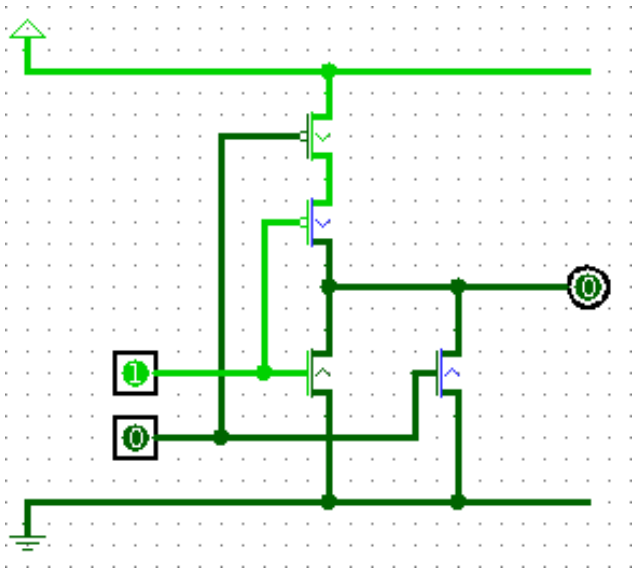
A	B	X
0	0	1
0	**1**	**0**
1	0	0
1	1	0



Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

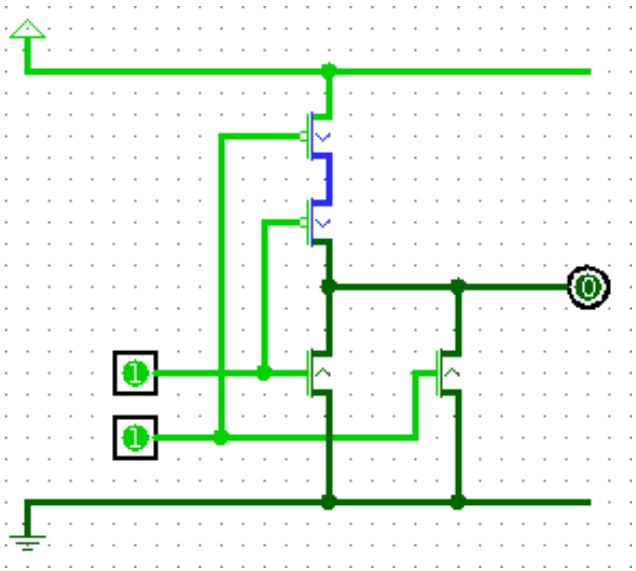
A	B	X
0	0	1
0	1	0
1	**0**	**0**
1	1	0



Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

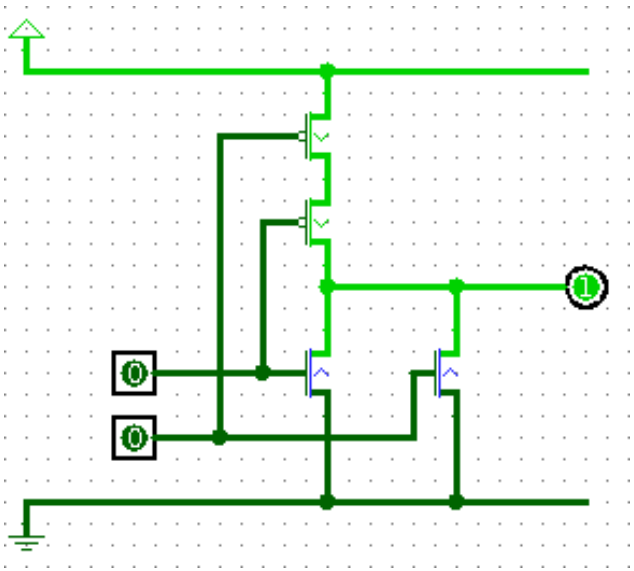
A	B	X
0	0	1
0	1	0
1	0	0
1	**1**	**0**

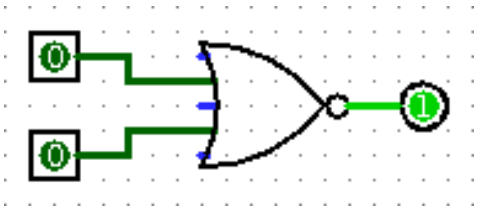


Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

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

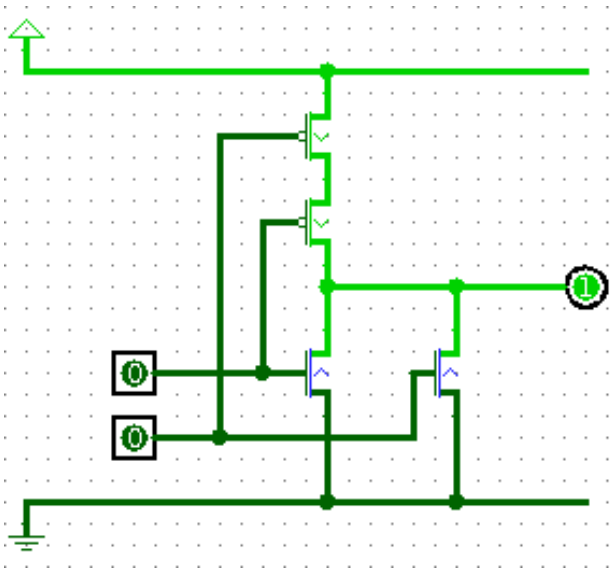


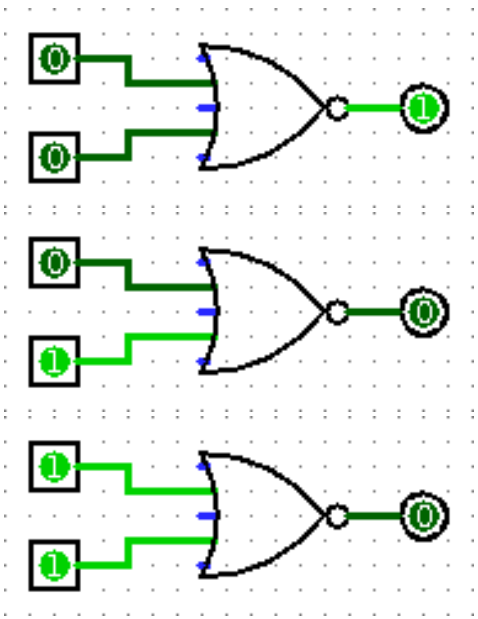


Logic NOR

$$X = f(A, B) = \overline{A \vee B}$$

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

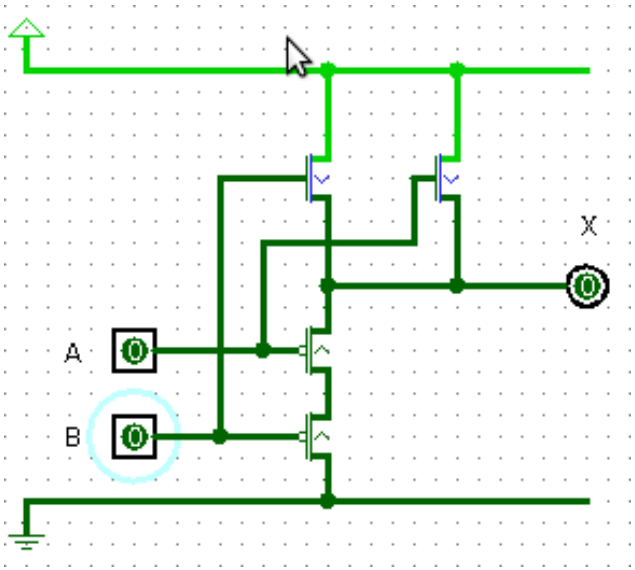




Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

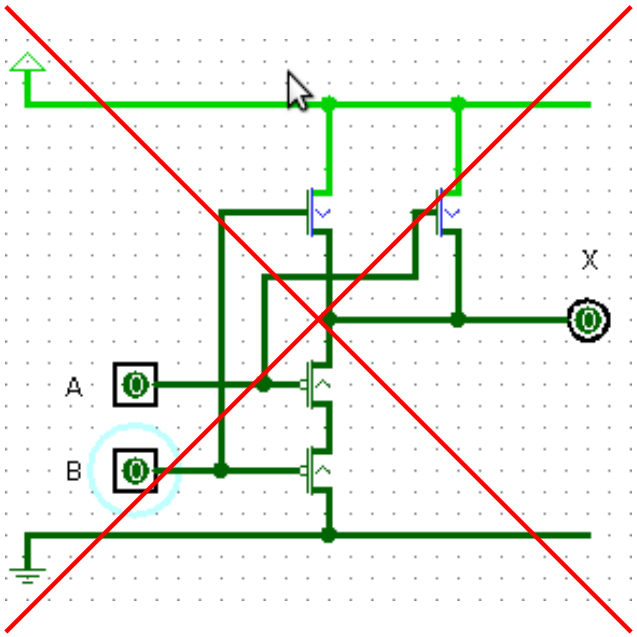
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

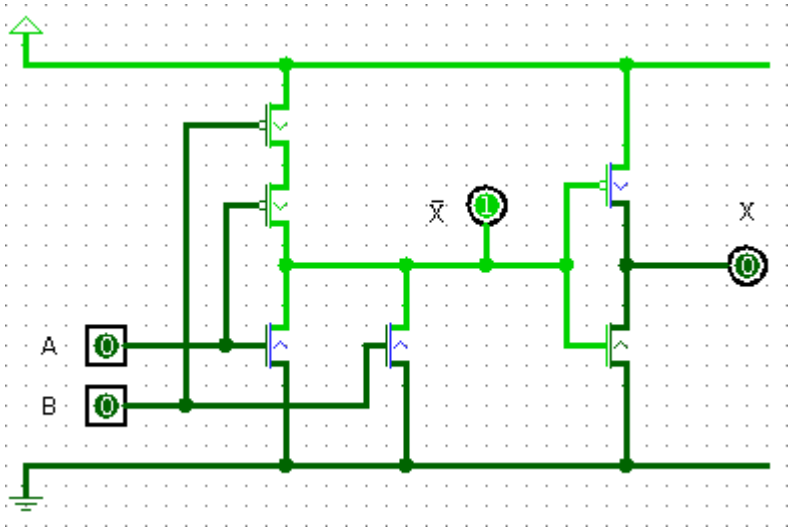
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

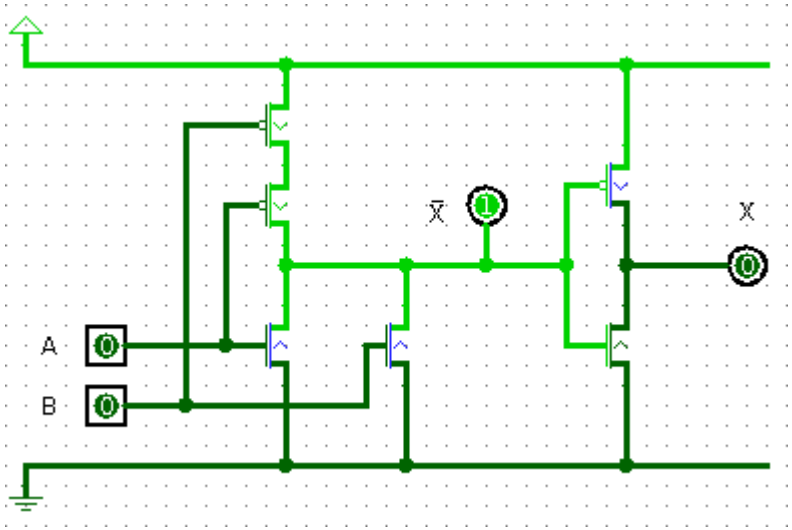
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

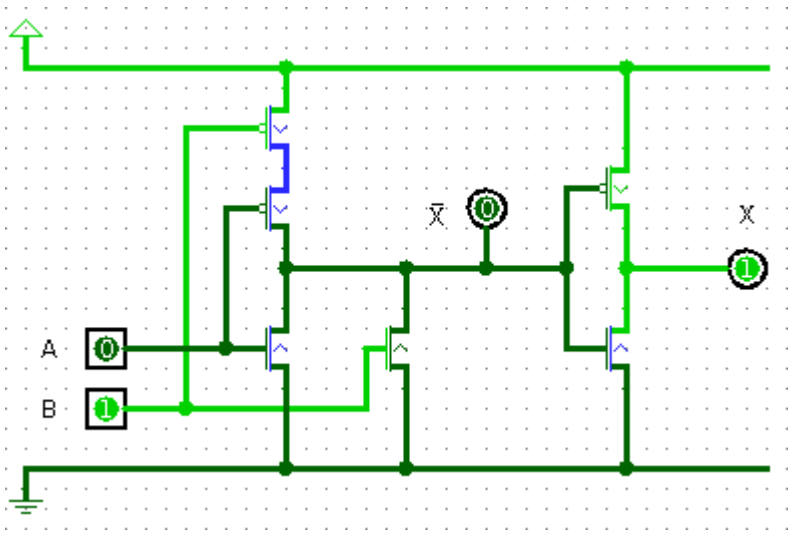
A	B	X
0	**0**	**0**
0	1	1
1	0	1
1	1	1



Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

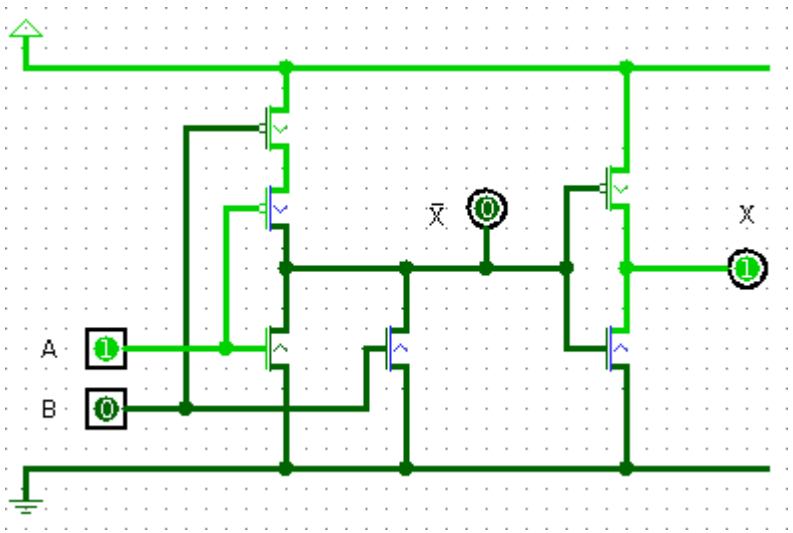
A	B	X
0	0	0
0	**1**	**1**
1	0	1
1	1	1



Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

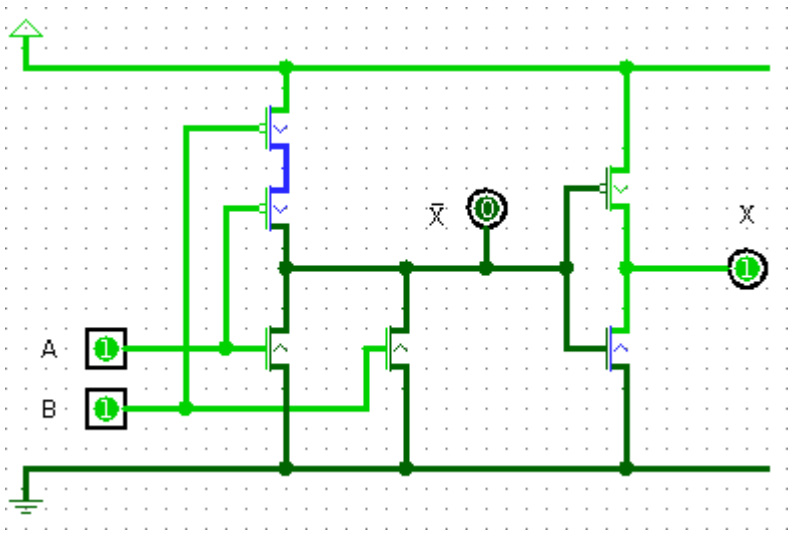
A	B	X
0	0	0
0	1	1
1	**0**	**1**
1	1	1



Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

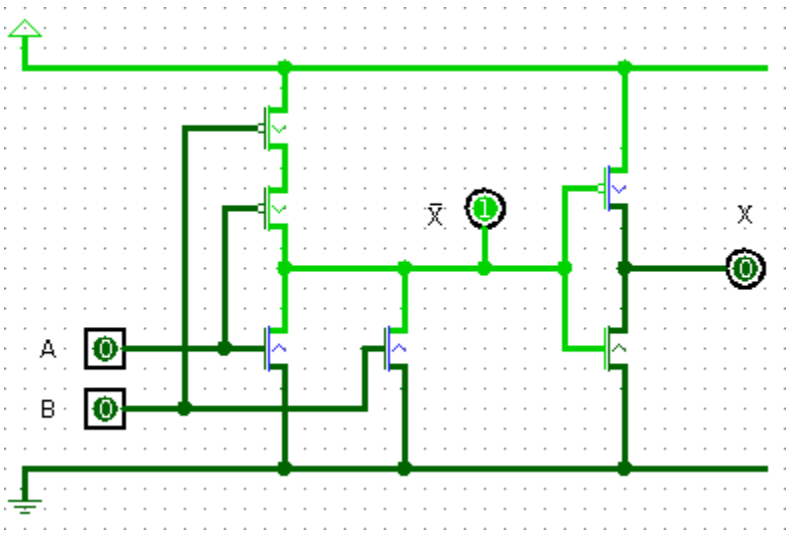
A	B	X
0	0	0
0	1	1
1	0	1
1	**1**	**1**

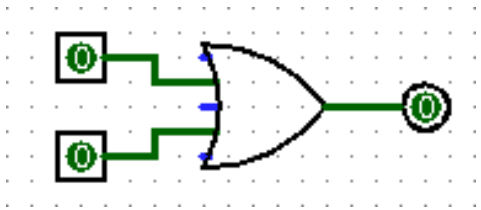


Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

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

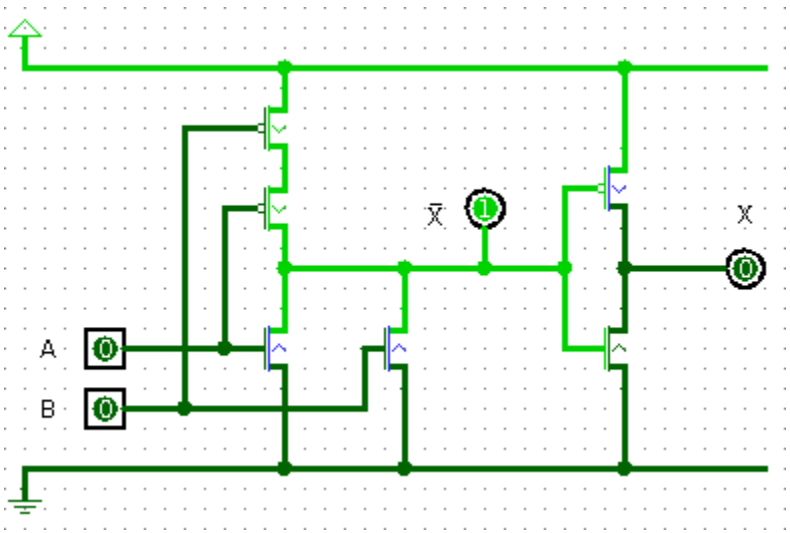


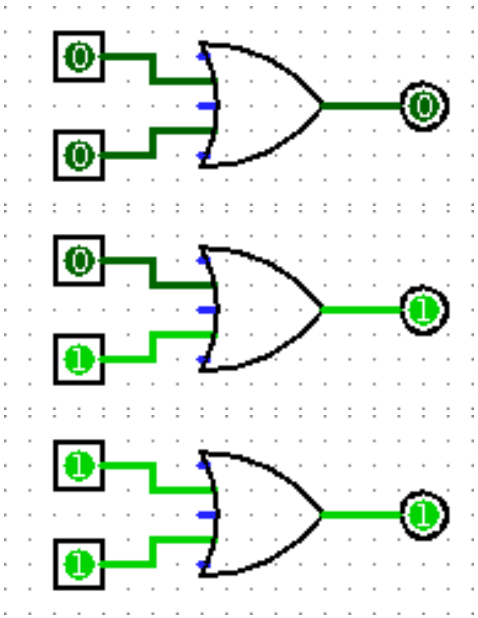


Logic OR (disjunction)

$$X = f(A, B) = A \vee B$$

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

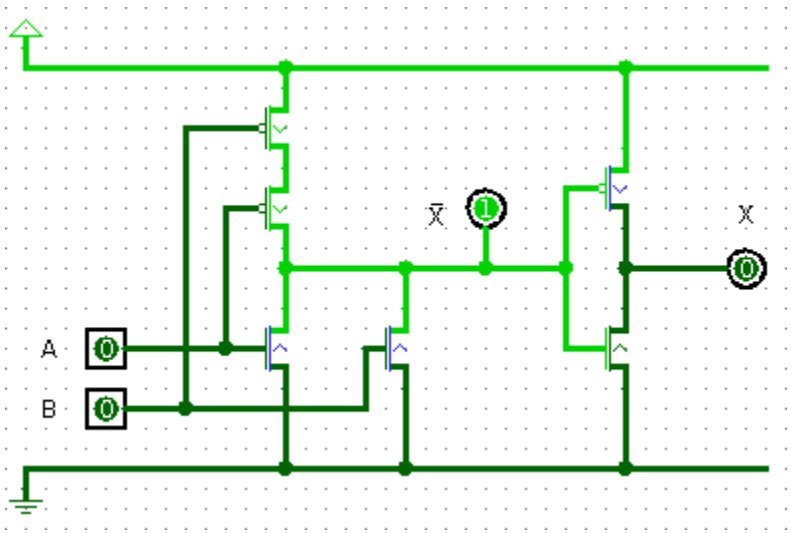


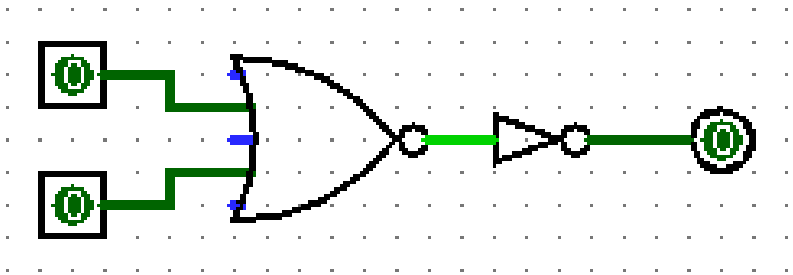


Two-argument Boolean functions

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

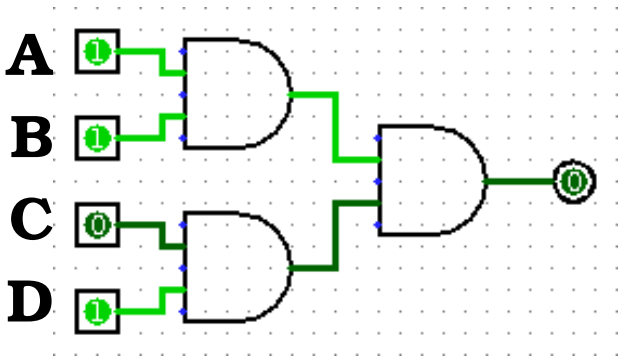
$\Big\}$ 4 positions $\Rightarrow 2^{2^2} = 16$ combinations

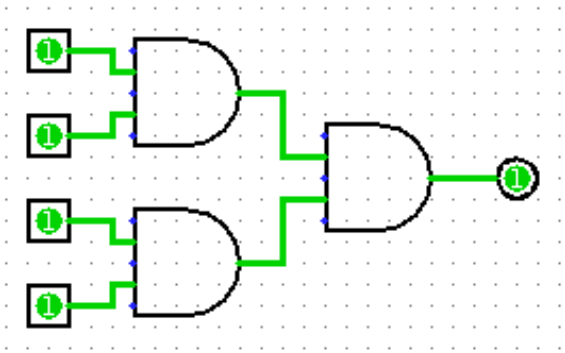




Multiple-input logic circuits

A	B	C	D	X
0	0	0	0	0
0	1	0	0	0
		...		
1	1	1	0	0
1	1	1	1	1





Computing arbitrary logic function

x_1	x_2	...	x_n	$f(x_1, x_2, \ldots, x_n)$	**conjunction**
0	0	...	0	1	$\overline{x_1} \cdot \overline{x_2} \cdot \ldots \cdot \overline{x_n}$
0	1	...	0	1	$\overline{x_1} \cdot x_2 \cdot \ldots \cdot \overline{x_n}$
0	1	...	0	0	0
0	1	...	0	0	0
		...			...
1	1	...	0	1	$x_1 \cdot x_2 \cdot \ldots \cdot \overline{x_n}$
1	1	...	1	0	0

Disjunctive Normal Form (DNF):

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) = \quad & \overline{x_1} \cdot \overline{x_2} \cdot \ldots \cdot \overline{x_n} \\ \vee \quad & \overline{x_1} \cdot x_2 \cdot \ldots \cdot \overline{x_n} \\ \vee \quad & \ldots \\ \vee \quad & x_1 \cdot x_2 \cdot \ldots \cdot \overline{x_n} \end{aligned}$$

Example: synthesis of the function XOR

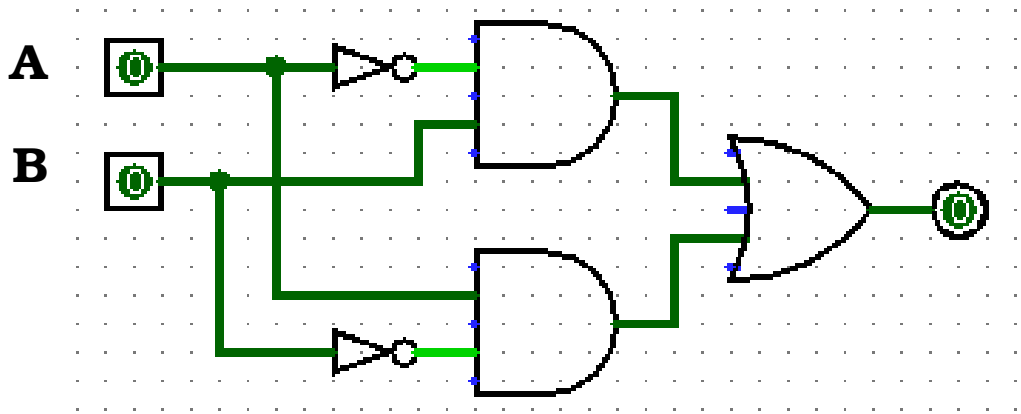
A	**B**	**X**	**conjunction**
0	0	0	
0	1	1	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	

DNF: $X = \overline{A}B \vee A\overline{B}$

Example: synthesis of the function XOR

A	B	X	conjunction
0	0	0	
0	1	1	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	

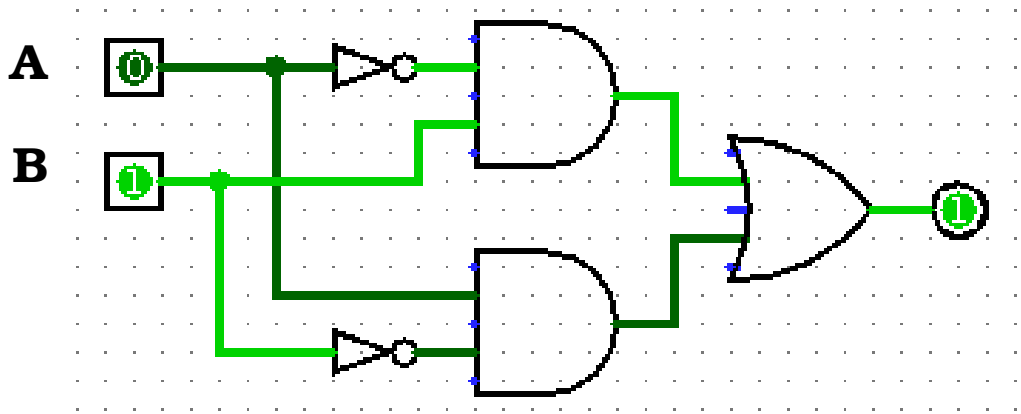
DNF: $X = \overline{A}B \vee A\overline{B}$



Example: synthesis of the function XOR

A	B	X	conjunction
0	0	0	
0	1	1	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	

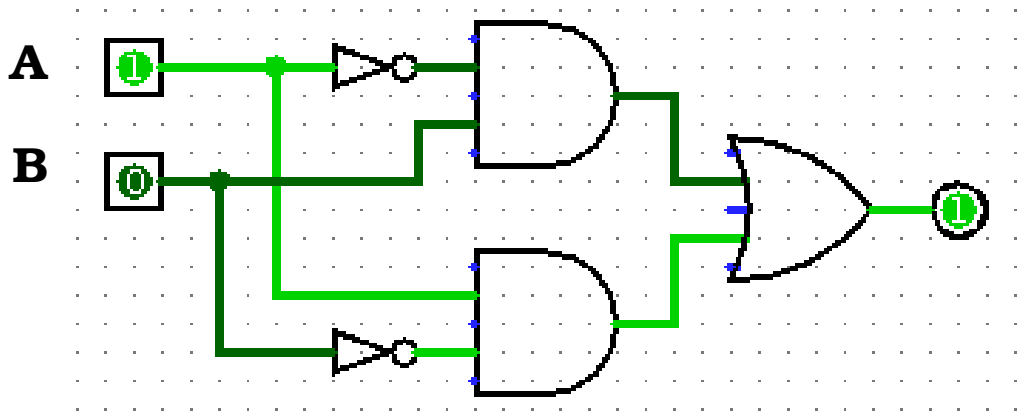
DNF: $X = \overline{A}B \vee A\overline{B}$



Example: synthesis of the function XOR

A	B	X	conjunction
0	0	0	
0	1	1	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	

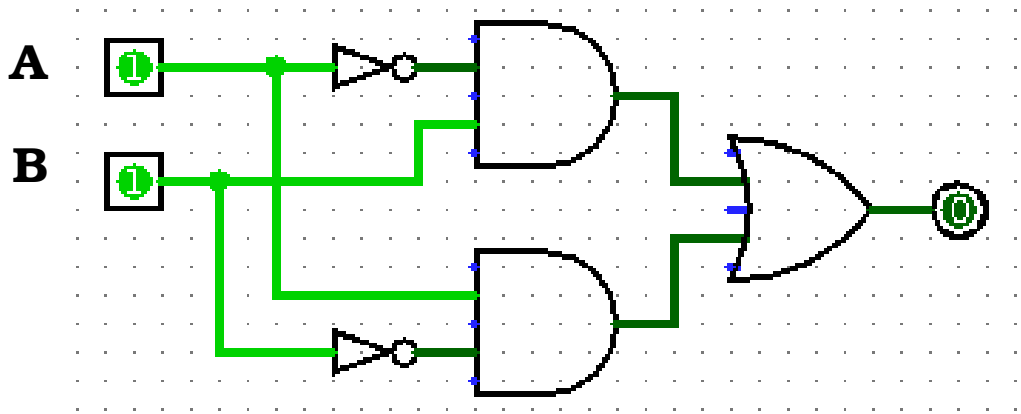
DNF: $X = \overline{A}B \vee A\overline{B}$



Example: synthesis of the function XOR

A	B	X	conjunction
0	0	0	
0	1	1	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	

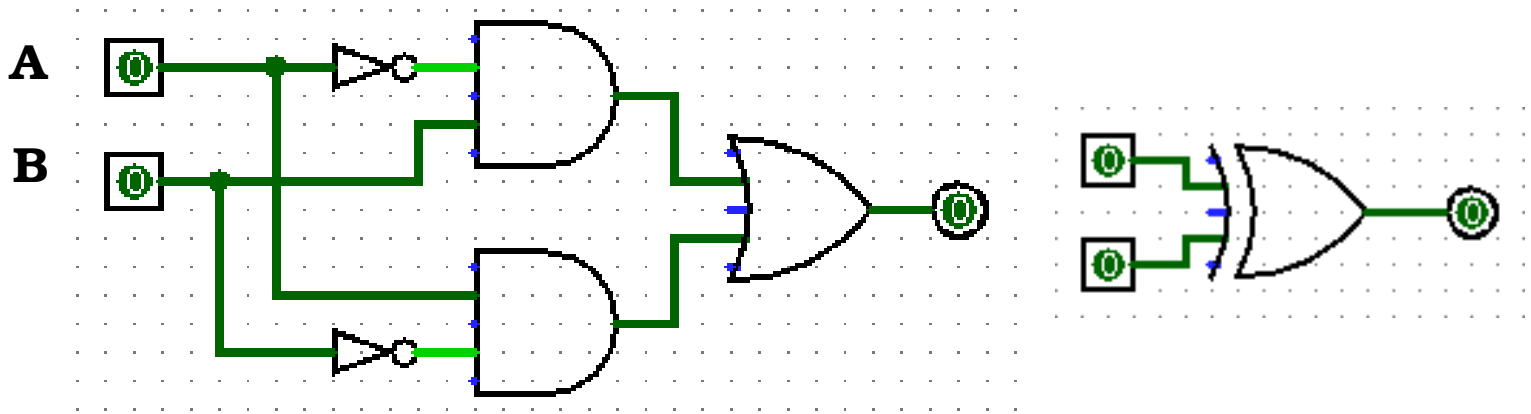
DNF: $X = \overline{A}B \vee A\overline{B}$



Example: synthesis of the function XOR

A	B	X	conjunction
0	0	0	
0	1	1	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	

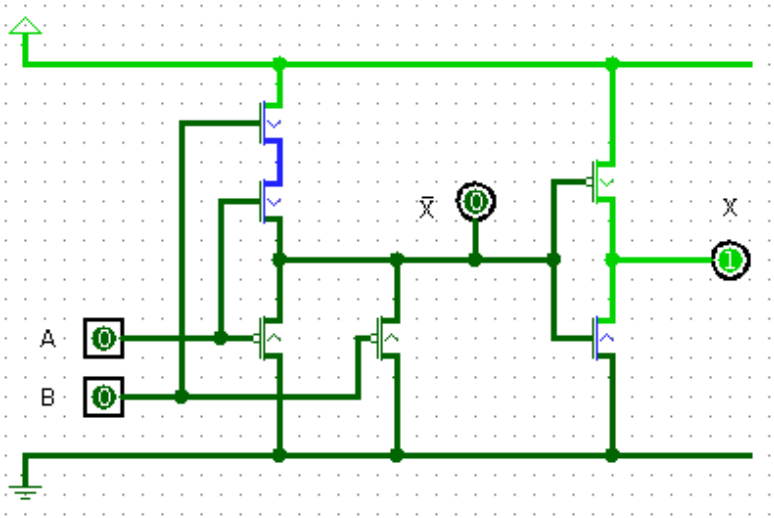
DNF: $X = \overline{A}B \vee A\overline{B}$

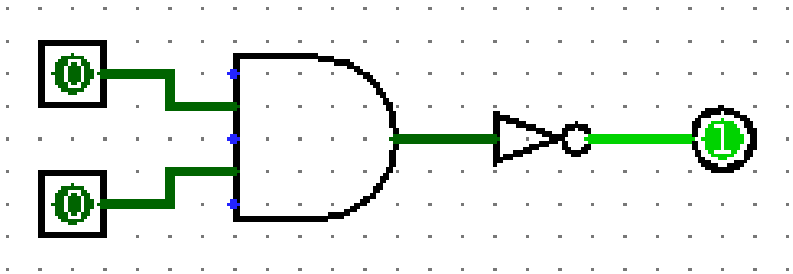


NAND gate (revisited)

$$X = f(A, B) = \overline{AB}$$

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

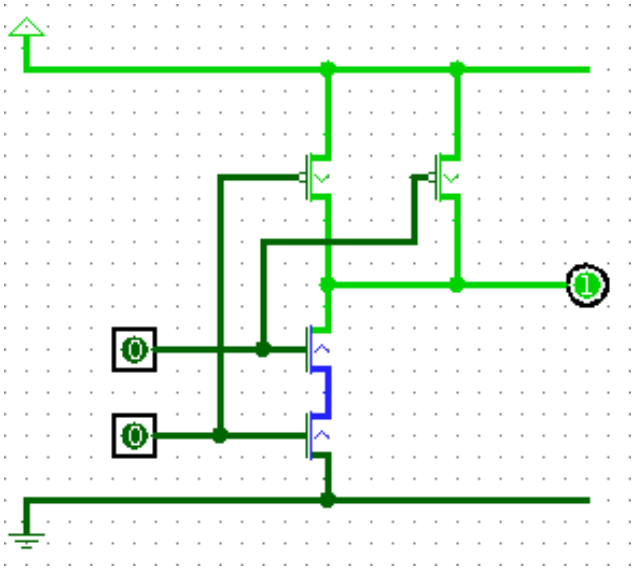


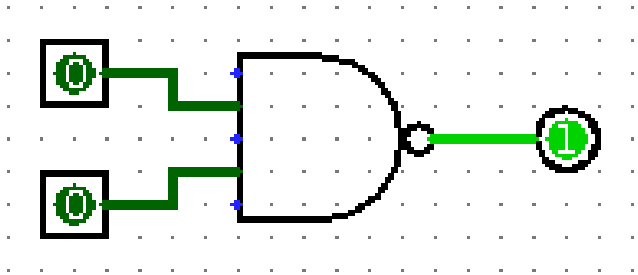


NAND gate (revisited)

$$X = f(A, B) = \overline{AB}$$

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

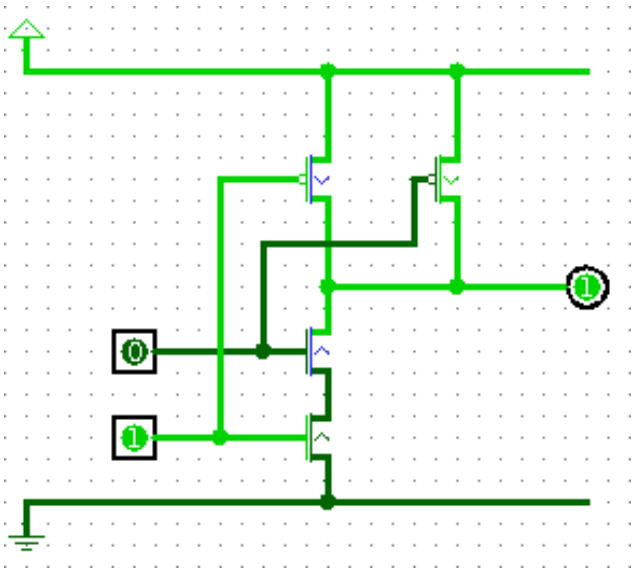


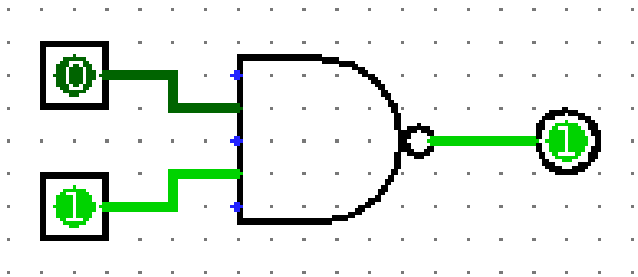


NAND gate (revisited)

$$X = f(A, B) = \overline{AB}$$

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

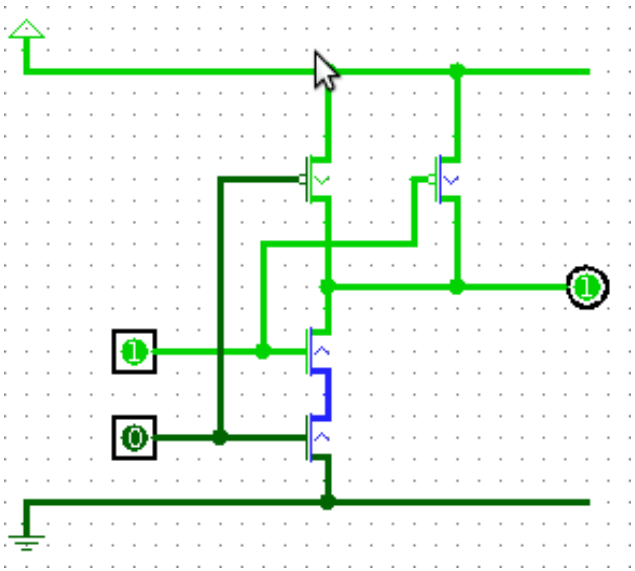


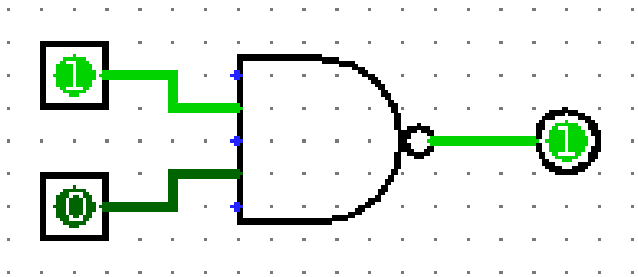


NAND gate (revisited)

$$X = f(A, B) = \overline{AB}$$

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

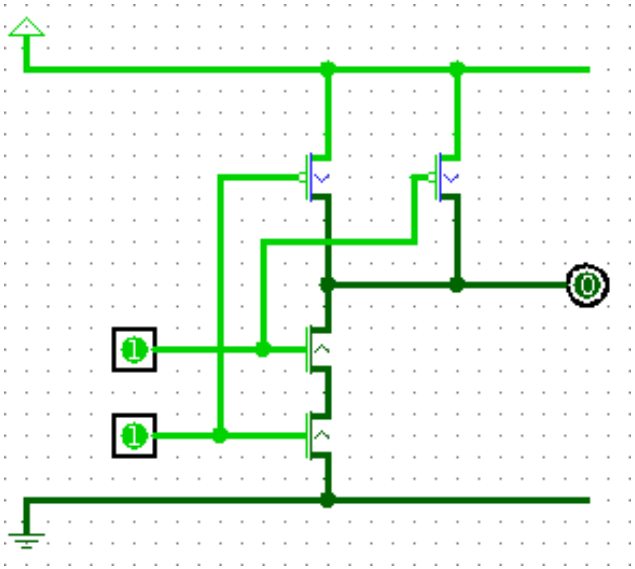


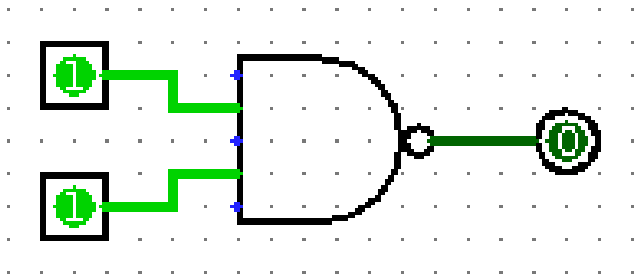


NAND gate (revisited)

$$X = f(A, B) = \overline{AB}$$

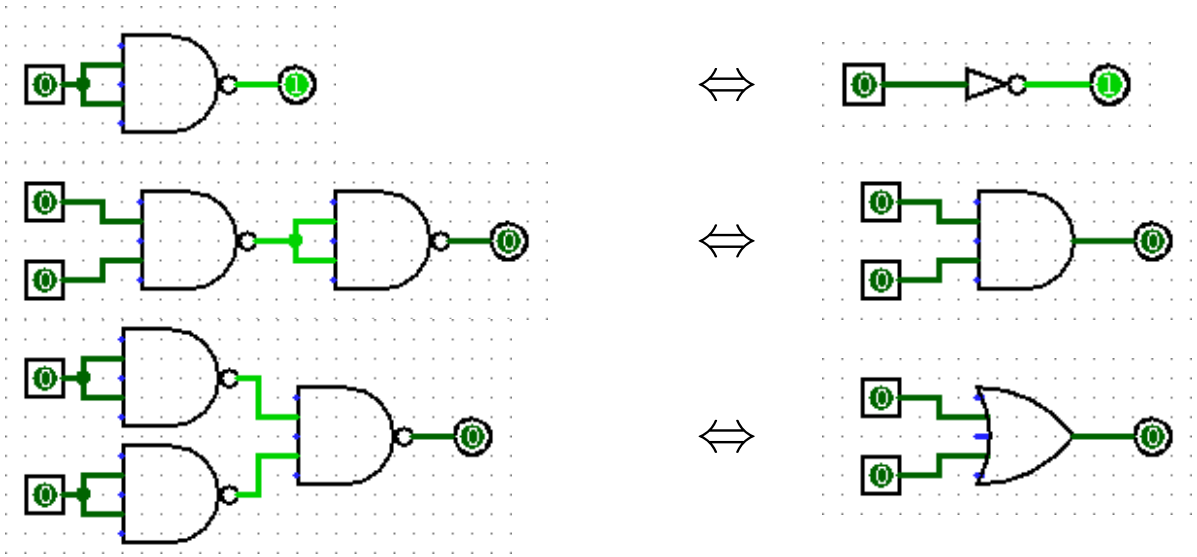
A	B	X
0	0	1
0	1	1
1	0	1
1	**1**	**0**





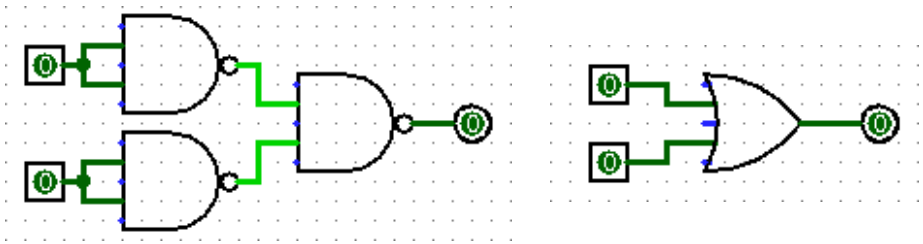
NAND – a Universal Gate

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



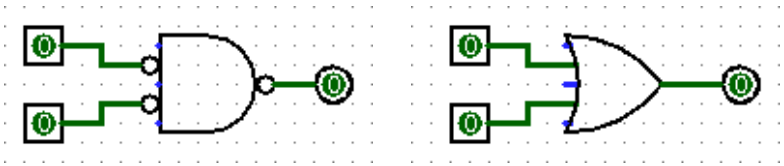
Dual gates

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A \vee B$
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1



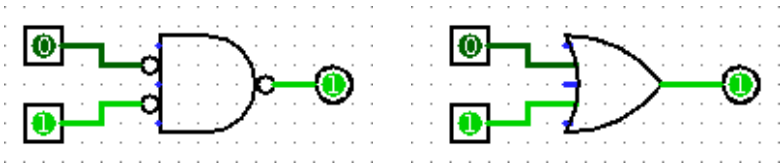
Dual gates

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A \vee B$
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1



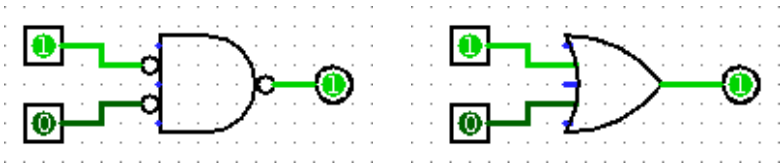
Dual gates

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A \vee B$
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1



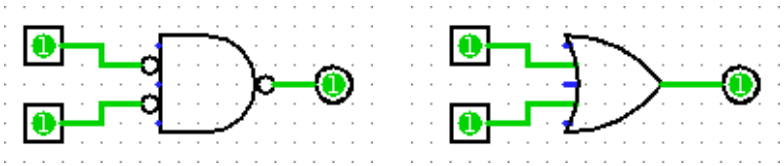
Dual gates

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A \vee B$
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1



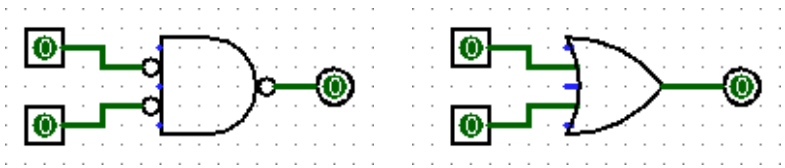
Dual gates

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A \vee B$
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1



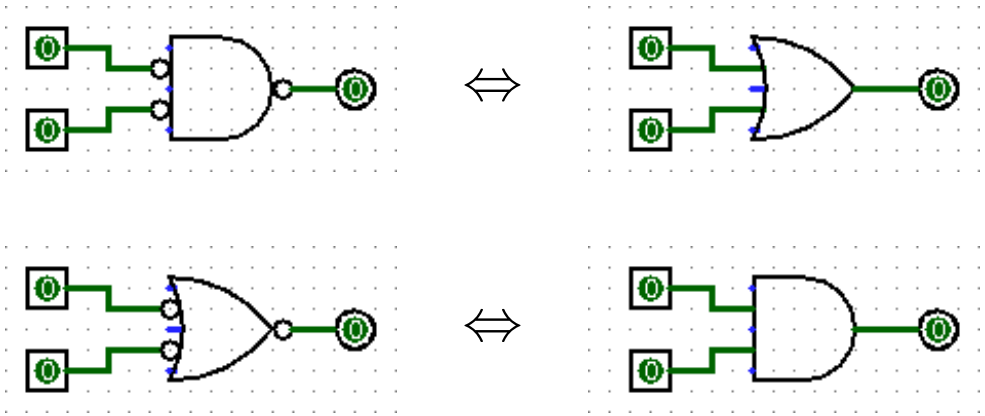
Dual gates

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A \vee B$
0	0	1	1	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	1	1



$$
\begin{aligned}
A \vee B &= \overline{\overline{A} \cdot \overline{B}} \\
\overline{\overline{A}} &= A \\
\overline{A \vee B} &= \overline{\overline{A} \cdot \overline{B}} = \overline{A} \cdot \overline{B}
\end{aligned}
$$

Dual gates, De Morgan laws

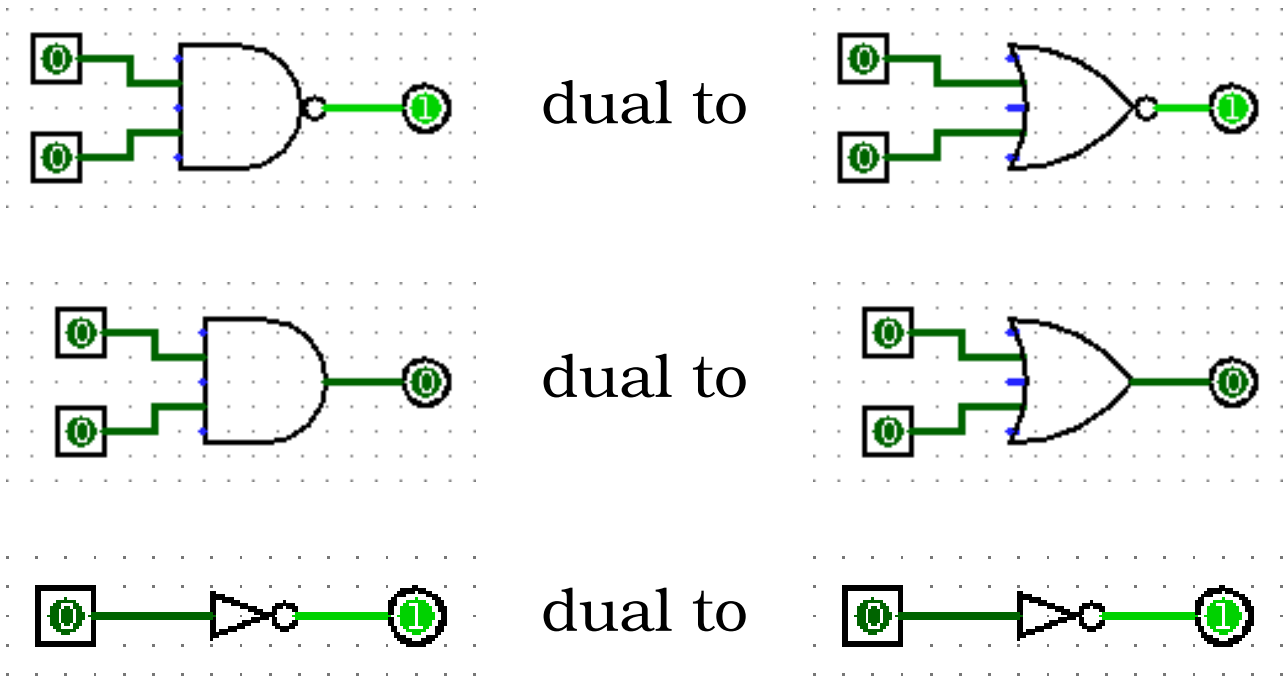


$$\left.\begin{array}{rcl} \overline{\overline{A}\cdot\overline{B}} & = & A\vee B \\ \overline{\overline{A}\vee\overline{B}} & = & A\cdot B \end{array}\right\} \text{ dual functions}$$

$$\left.\begin{array}{rcl} \overline{A\vee B} & = & \overline{A}\cdot\overline{B} \\ \overline{A\cdot B} & = & \overline{A}\vee\overline{B} \end{array}\right\} \text{ De Morgan laws}$$

$$h \text{ is dual of } f \underset{\text{def}}{\Leftrightarrow} h(x_1,x_2,\ldots,x_n)=\overline{f(\overline{x_1},\overline{x_2},\ldots,\overline{x_n})}$$

Another universal gate: NOR



Complete systems of logic functions

Superposition:

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

Example:

$$\mathrm{NOT}(\mathrm{AND}(x_1, \mathrm{NOT}(x_2)) \underset{\text{def}}{\equiv} \overline{x_1 \overline{x_2}} = \overline{x_1} \vee \overline{\overline{x_2}} = \overline{x_1} \vee x_2$$

Complete systems (examples):

$$\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\} = \{\cdot, \vee, \overline{X}\} \quad \text{(not minimal!)}$$
$$\{\mathrm{AND}, \mathrm{NOT}\} = \{\cdot, \overline{X}\}$$
$$\{\mathrm{OR}, \mathrm{NOT}\} = \{\vee, \overline{X}\}$$
$$\{\mathrm{AND}, \mathrm{XOR}, 1\} = \{\cdot, \oplus, 1\}$$
$$\{\mathrm{OR}, \mathrm{XOR}, 1\} = \{\vee, \oplus, 1\}$$
$$\{\mathrm{NAND}\} = \{\uparrow\} \quad \text{Sheffer stroke}$$
$$\{\mathrm{NOR}\} = \{\downarrow\} \quad \text{Peirce's arrow/Quine's dagger}$$

Closed Boolean function classes

Function classes closed under superposition:

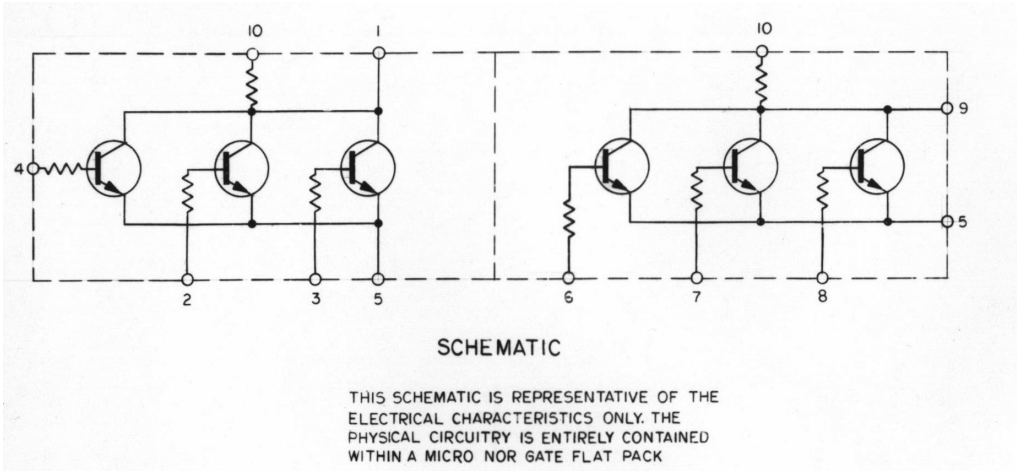
1. T_0: zero-preserving:
$$f(0,0,\ldots,0) = 0$$
2. T_1: one-preserving:
$$f(1,1,\ldots,1) = 1$$
3. S: self-dual:
$$f(x_1,x_2,\ldots,x_n) = \overline{f(\overline{x_1},\overline{x_2},\ldots,\overline{x_n})}$$
4. L: linear:
$$f(x_1,x_2,\ldots,x_n) = a_0 \oplus a_1 x_1 \oplus \cdots \oplus a_n x_n$$
5. M: monotonic:
$$\vec{a} \underset{\text{def}}{\equiv} (a_1,a_2,\ldots,a_n)$$
$$\vec{a} \le \vec{b} \underset{\text{def}}{\Leftrightarrow} \forall i \in \{1,\ldots,n\} : a_i \le b_i$$
$$f \in M \underset{\text{def}}{\Leftrightarrow} \vec{a} \le \vec{b} \Rightarrow f(\vec{a}) \le f(\vec{b})$$

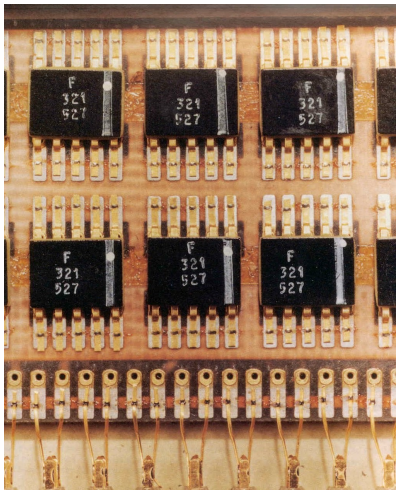
Post's theorem

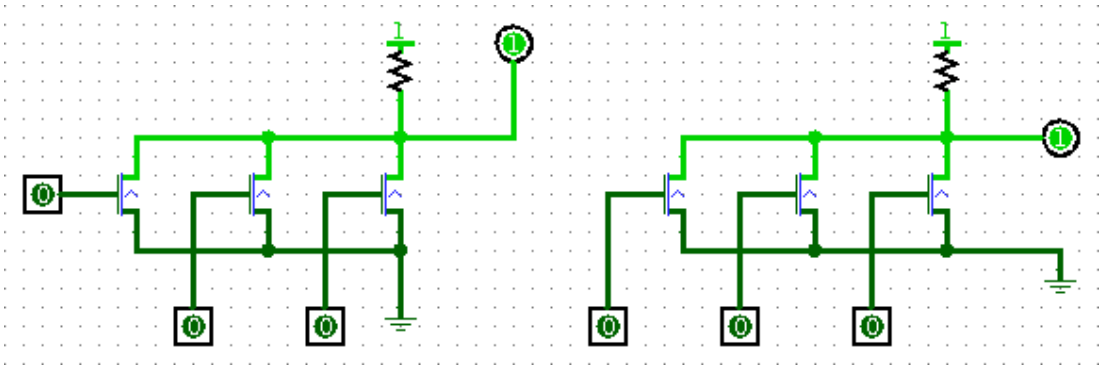
A Boolean function set $B = \{f_1, f_2, \ldots, f_n\}$ is complete under superposition **iff**:

1. it contains a function that does not preserve 0:
$$\exists f_i \in B : f_i \notin T_0$$
2. it contains a function that does not preserve 1:
$$\exists f_i \in B : f_i \notin T_1$$
3. it contains a function that is not self-dual:
$$\exists f_i \in B : f_i \notin S$$
4. it contains a function that is not linear:
$$\exists f_i \in B : f_i \notin L$$
5. it contains a function that is not monotonic:
$$\exists f_i \in B : f_i \notin M$$

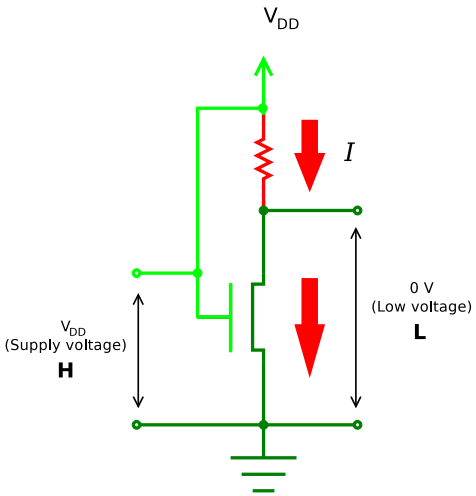
Apollo Guidance Computer gates

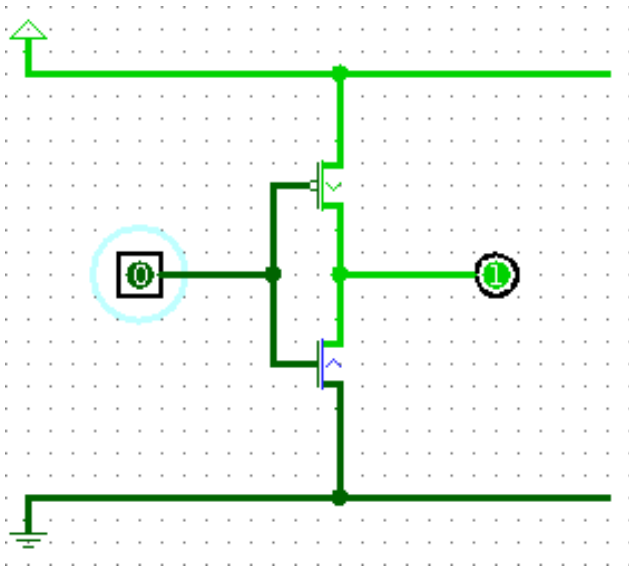


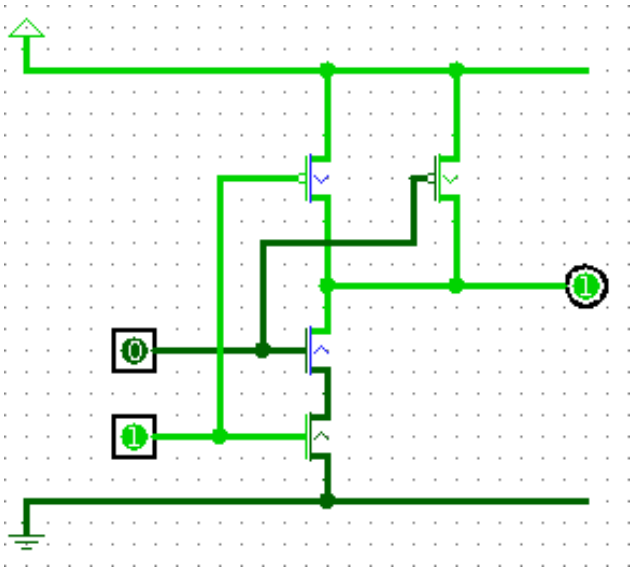


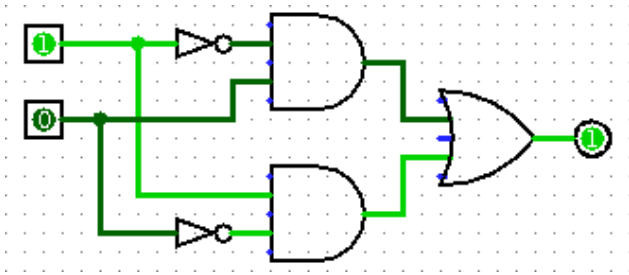


Achievements so far ...









Take home messages

- Digital circuits can be described using Boolean algebra;
- Any Boolean function can be implemented using a complete set of Boolean functions, as per Post's theorem; each of functions NAND and NOR alone form a complete set and any digital circuit can be build from them;
- All Boolean functions can be implemented as electric (e.g. CMOS FET) switches;
- All blocks of a computer can be built from controllable electric switches; modern computers use CMOS field effect transistors.