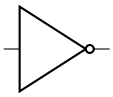
Logic Gates

NOT Gate

Implements the NOT(X) *unary* logic function. Also known as the invertion function.

$\overline{X}$

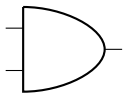


	X
true	*false*
`false`	`true`

AND Gate

Implements the AND(X, Y) *binary* logic function. Returns `true` if both X and Y are `true`.

$X \cdot Y$

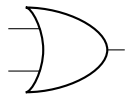


		X
		true	*false*
Y	*true*	`true`	`false`
	false	`false`	`false`

OR Gate

Implements the OR(X, Y) *binary* logic function. Returns `true` if any of X and Y are `true`.

$X + Y$

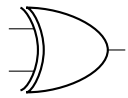


		X
		true	*false*
Y	*true*	`true`	`true`
	false	`true`	`false`

XOR Gate

Implements the exclusive OR or XOR(X, Y) *binary* logic function. Returns `true` if exacty one of X and Y are `true`.

$X \oplus Y$

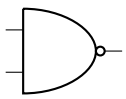


		X
		true	*false*
Y	*true*	`false`	`true`
	false	`true`	`false`

NAND Gate

Implements the negated AND or NAND(X, Y) *binary* logic function. Can be constructed as NOT(AND(X, Y)). Returns `true` if at most one of X and Y are `true`.

$\overline{X \cdot Y}$

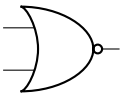


		X
		true	*false*
Y	*false*	`true`	`true`
	true	`true`	`false`

NOR Gate

Implements the negated OR or NOR(X, Y) *binary* logic function. Can be constructed as NOT(OR(X, Y)). Returns `true` if neither X or Y are `true`.

$\overline{X + Y}$



Y \ X	*true*	*false*
false	`false`	`true`
true	`true`	`true`