3. Porte logiche

Le porte logiche

Le porte logiche sono i componenti base di un circuito logico combinatorio, e permettono di ottenere specifici segnali di uscita da un certo numero di segnali di ingresso: con le espressioni dell'algebra di boole, stiamo effettivamente definendo un segnale di uscita prodotto tramite il passaggio di più segnali attraverso delle porte logiche.

Porte logiche di base

AND, OR, NOT ed XOR

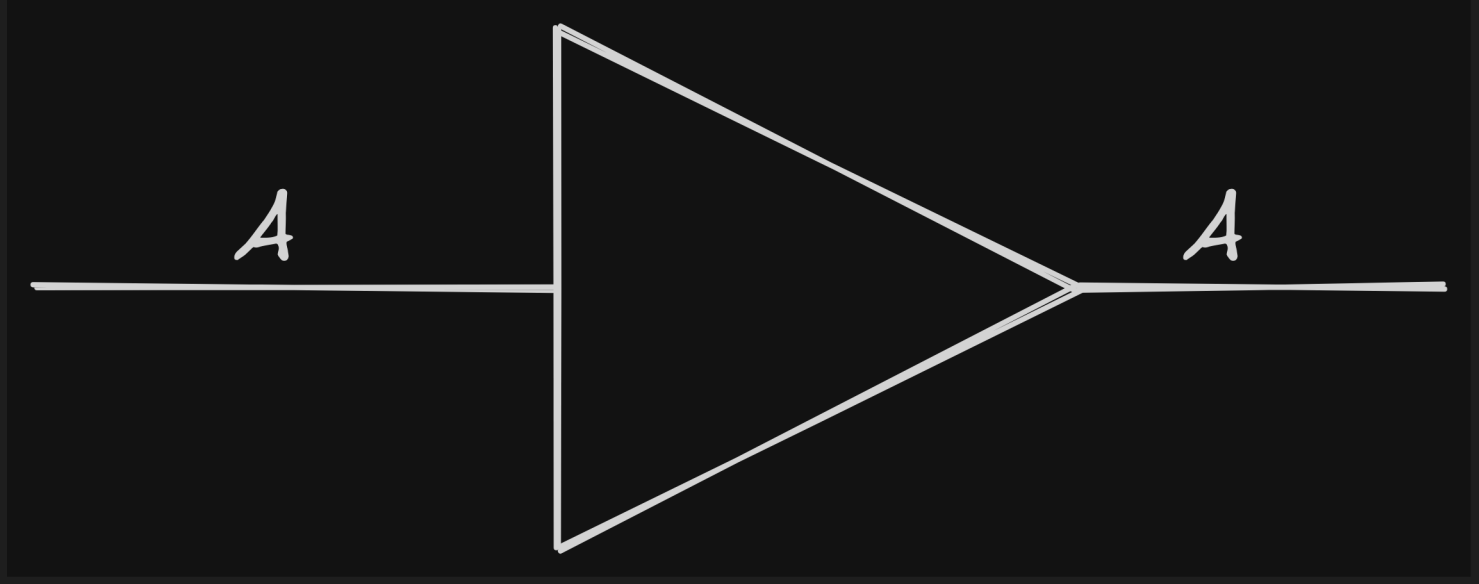
Mentre parlavamo dell'algebra di Boole, abbiamo già visto queste quattro porte ed il loro comportamento:

- AND
- OR
- NOT
- XOR

Buffer

La porta Buffer è il complemento della porta NOT e quindi restituisce in output il valore di input così com'è.

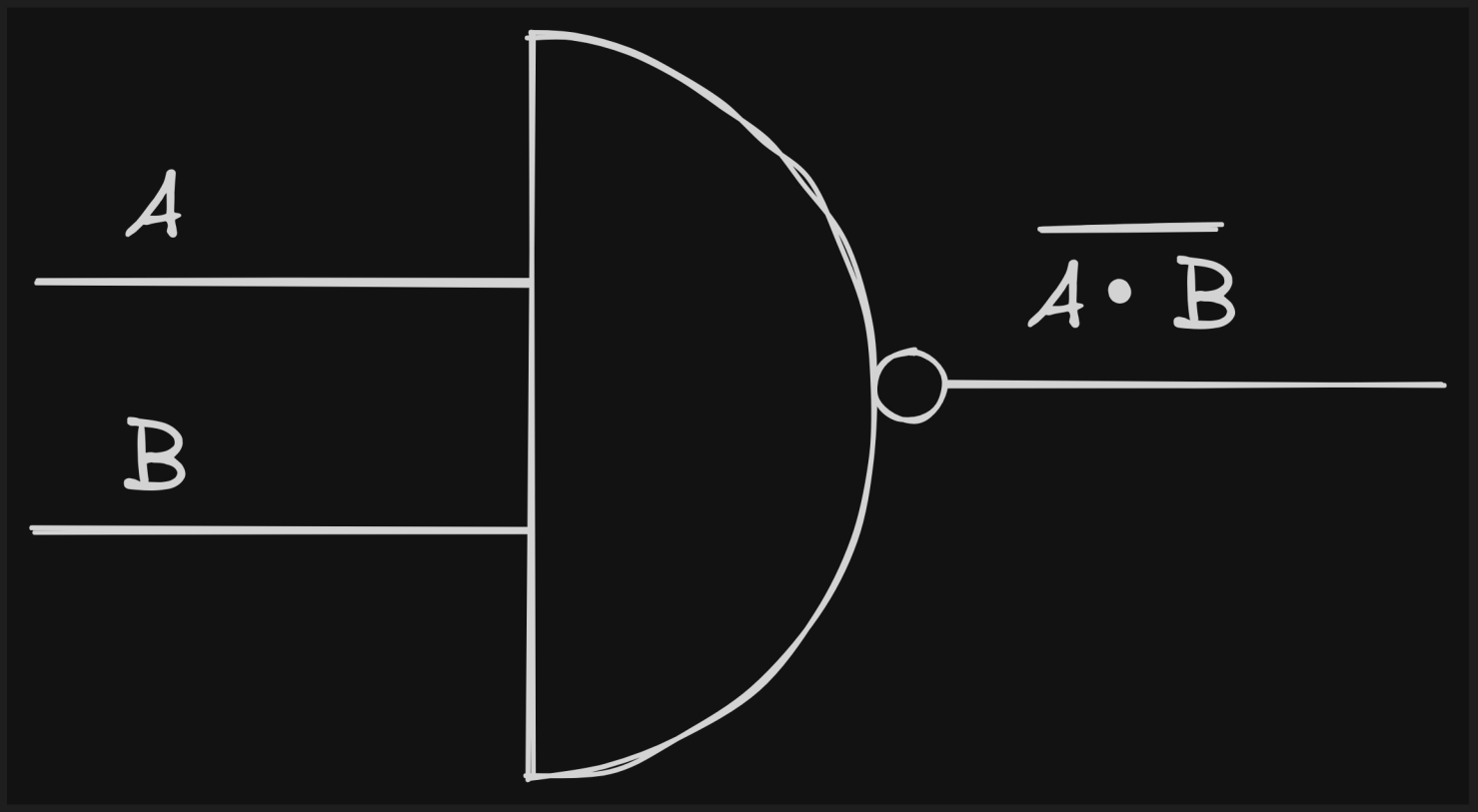
A	A
0	0
1	1



NAND

La porta NAND è il complemento della porta AND e quindi restituisce un valore di 1 quando almeno uno dei due fattori è 0.

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

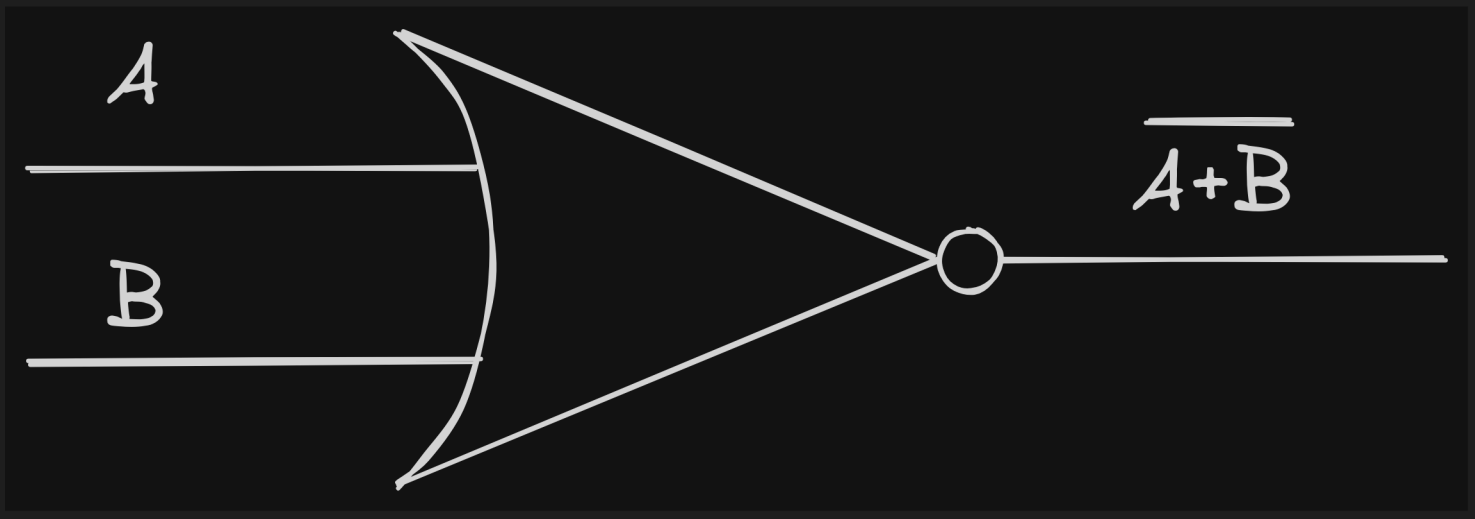


Si noti che viene utilizzata la simbologia alternativa della porta NOT, mettendo un "pallino" sull'uscita della porta per indicare la complementazione.

NOR

La porta NOR è il complemento della porta OR e quindi restituisce un valore di 1 esclusivamente quando ambi gli addendi hanno il valore 0.

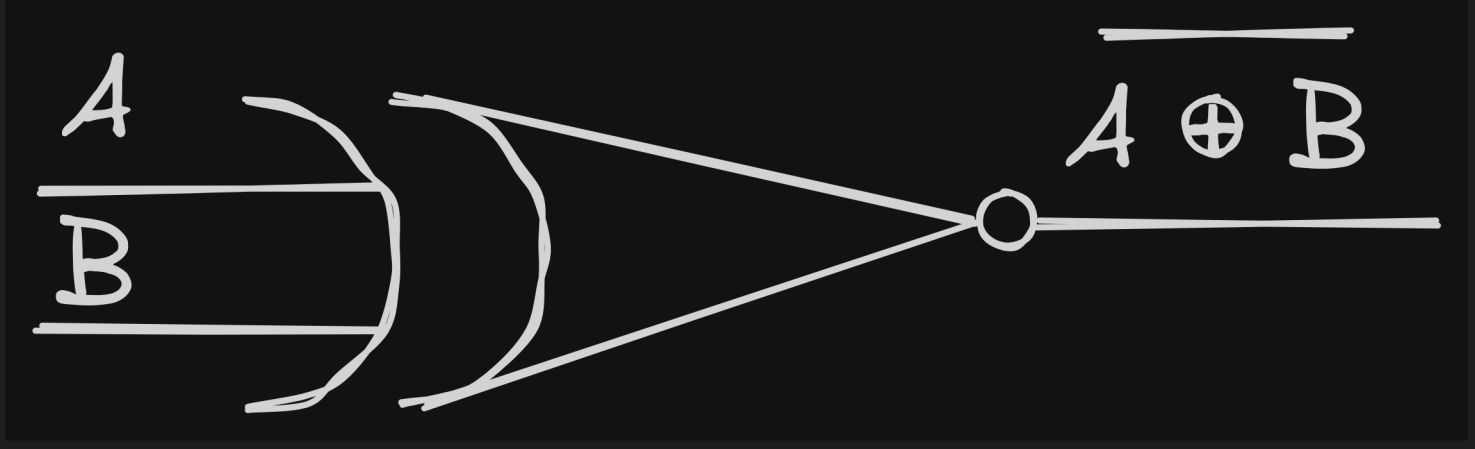
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



XNOR

L'XNOR è il complemento della porta XOR, e quindi restituisce un valore di 1 esclusivamente quando i due operandi hanno valore uguale.

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1



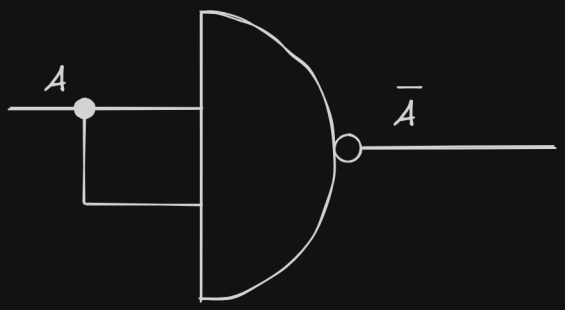
Universalità

Le porte NAND e NOR godono di una proprietà detta universalità: cioè, è possibile utilizzare solo porte NAND o solo porte NOR per reimplementare la funzione delle altre porte logiche, e dunque, permettono di realizzare qualsiasi circuito combinatorio con solo porte NAND o solo porte NOR.

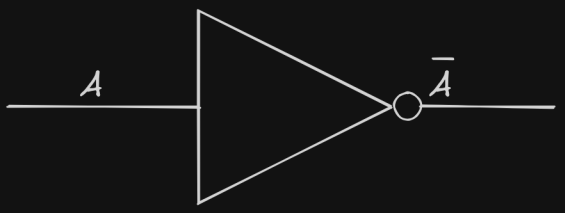
Universalità porte NAND

Le porte NAND permettono di definire le altre porte logiche di base. Nei circuiti realizzati con solo NAND, se specificato si possono rappresentare le operazioni usando i simboli grafici alternativi. Nei grafici della AND e della OR elencati qui sotto, il simbolo della NOT è il simbolo grafico alternativo, quindi va inteso come una NOT implementata tramite porta NAND.

NOT

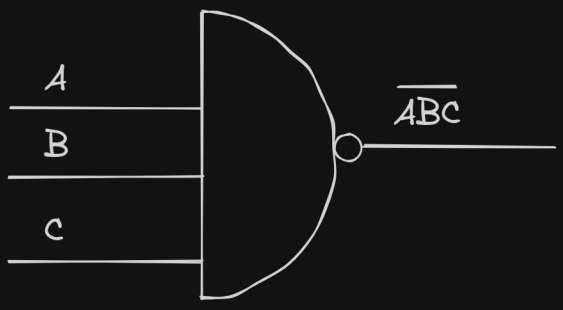


Simbolo grafico alternativo: NOT

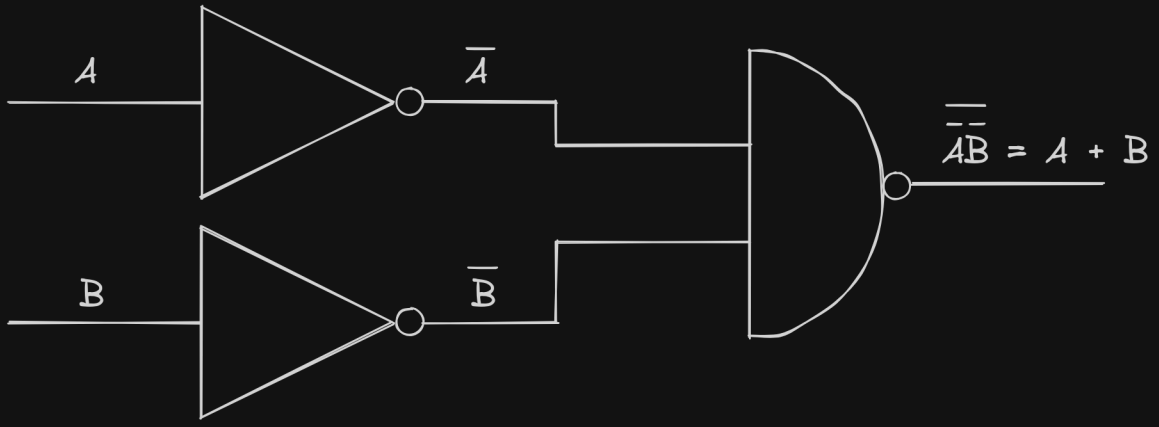


AND

Simbolo grafico alternativo: AND-NOT



OR



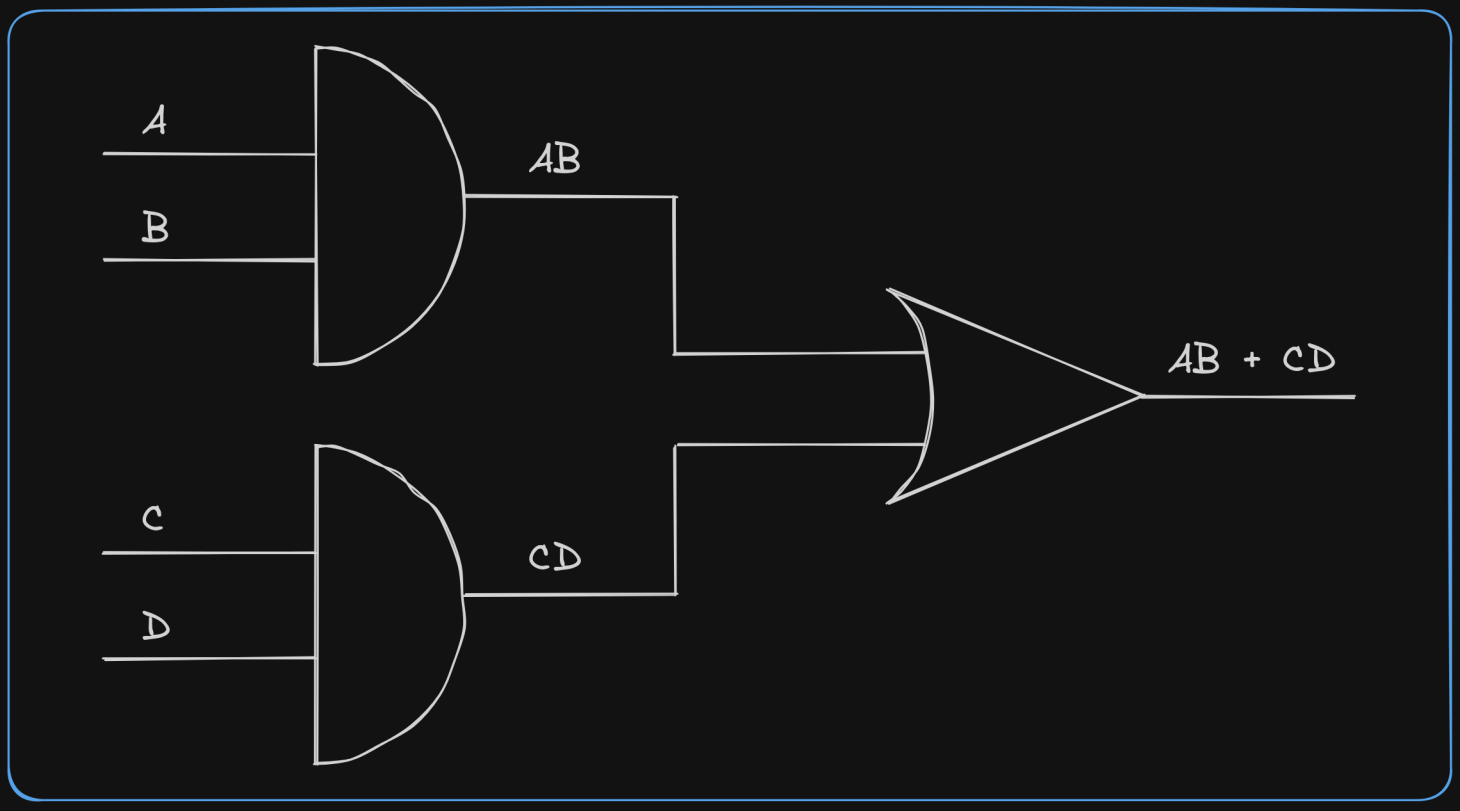
Simbolo grafico alternativo: NOT-OR

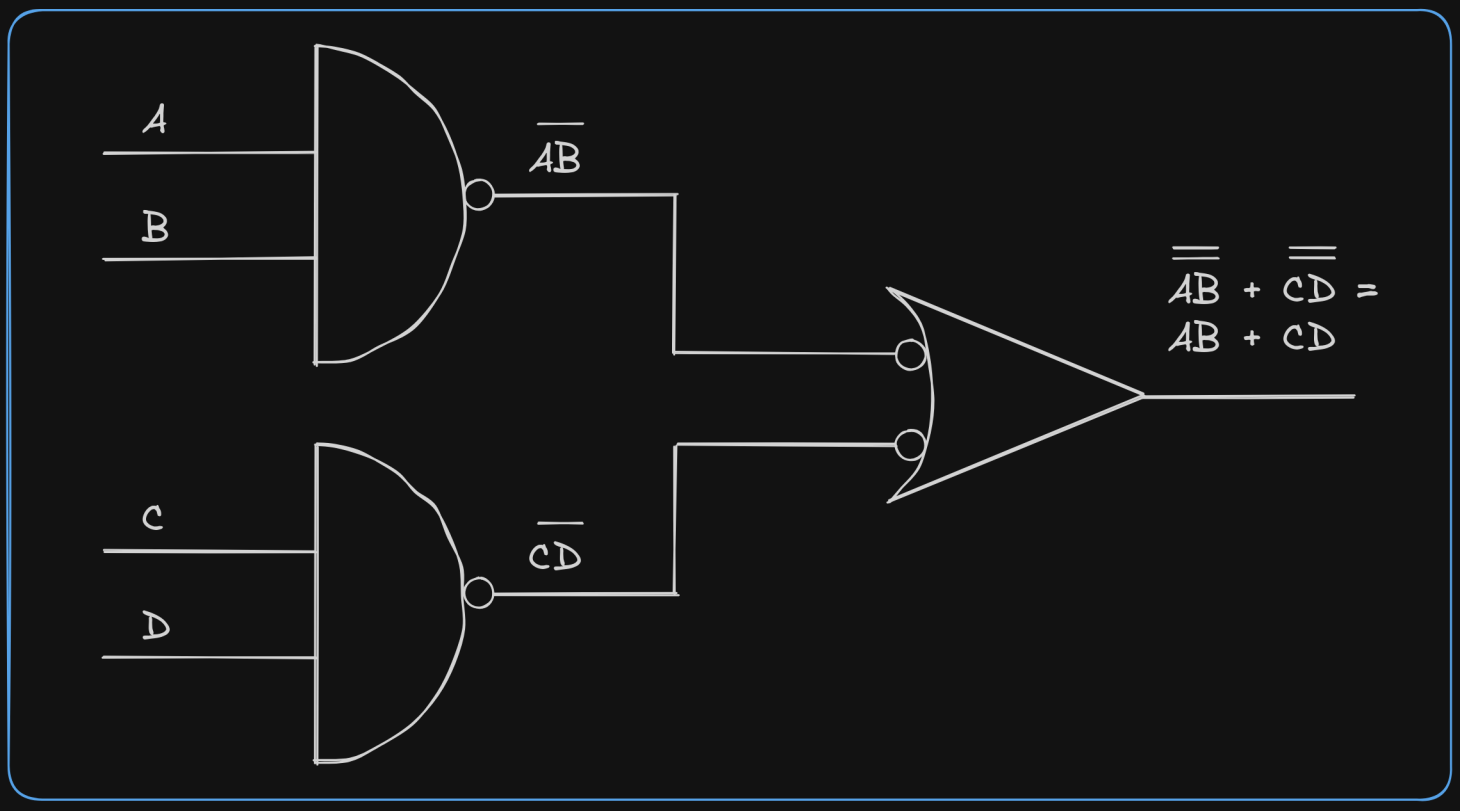
Si possono applicare le relative sostituzioni in una espressione booleana per scriverla in forma All-NAND.

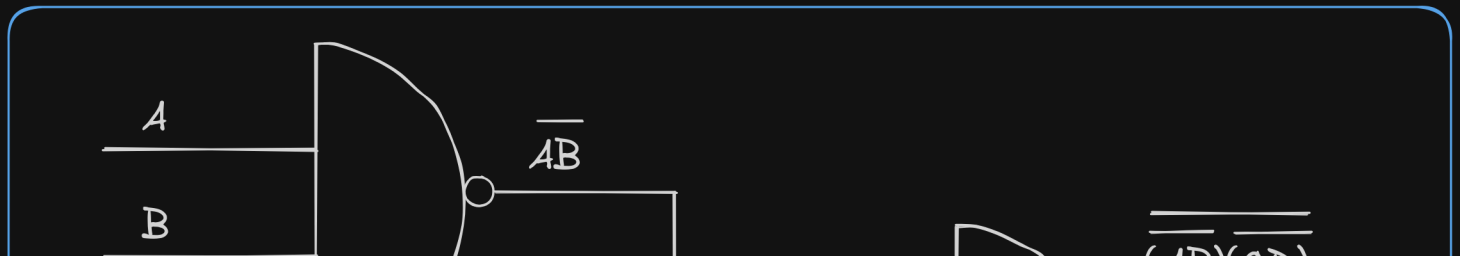
Implementazione a due livelli

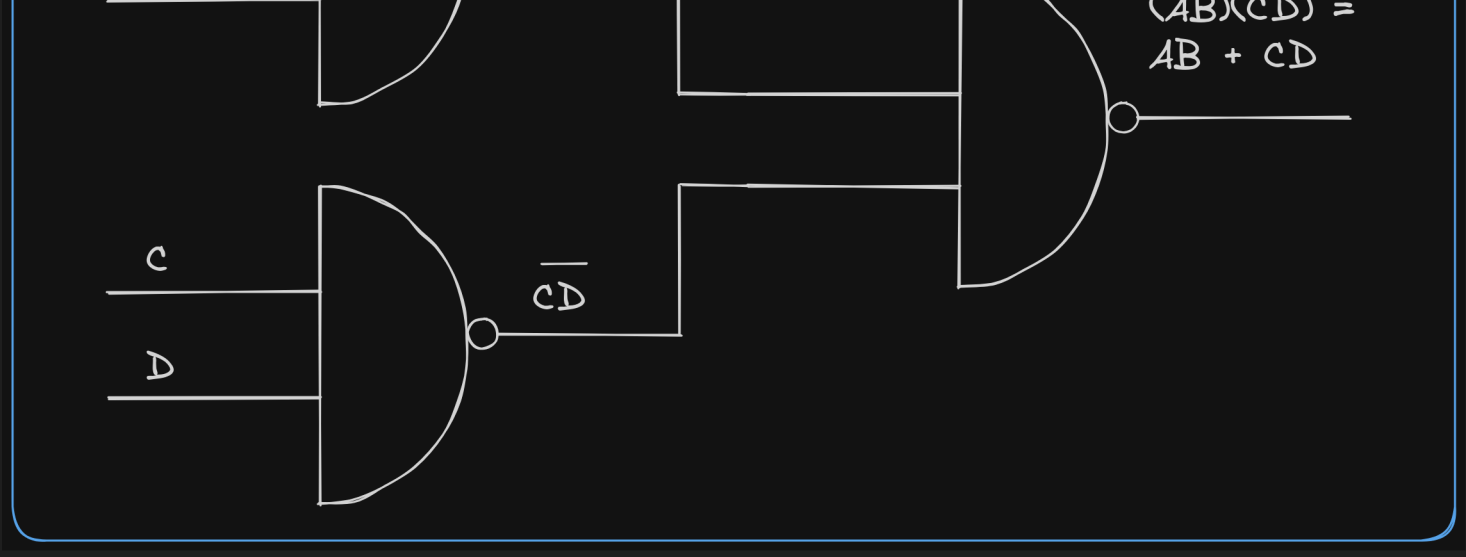
L'implementazione di una funzione booleana con porte NAND risulta semplice se la funzione è espressa in forma SOP, che genera quindi un circuito a due livelli: due porte AND (dette di primo livello) sono connesse ad una porta OR (detta di secondo livello). Eventuali invertitori sugli ingressi degli AND e/o sull'uscita dell'OR non sono considerati livelli aggiuntivi. Una volta progettato il circuito normalmente a due livelli, risulta più semplice ricavarsi il circuito con sole porte NAND.

3 Modi per implementare a due livelli
$AB + CD$









Circuiti NAND multilivello

Esistono vari casi in cui il progetto di un sistema digitale richieda strutture con tre o più livelli. In questi casi, in genere si esprime la funzione booleana in termini di AND, OR e NOT, per poi implementare direttamente il circuito con tali porte, o convertendo prima in logica NAND.

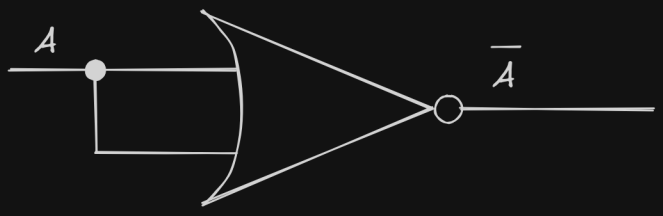
La procedura generale per la conversione di diagrammi AND-OR multilivello in diagrammi in logica NAND, utilizzando i simboli alternativi, è tipicamente:

1) Converti tutte le porte AND in porte NAND con simbolo AND-NOT
2) Converti tutte le porte OR in porte NAND con simbolo NOT OR
3) Controlla tutte le complementazioni nel diagramma, inserendo una porta NOT o una complementazione in ingresso per ogni "pallino" che non è bilanciato da un altro sulla stessa connessione.

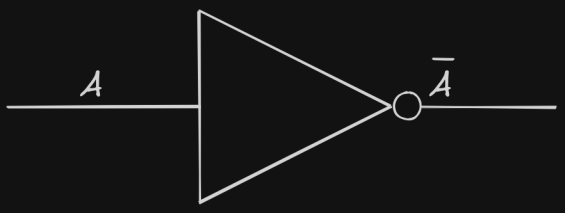
Universalità porte NOR

L'operazione NOR è la duale dell'operazione NAND, pertanto è anche essa universale, e tutte le procedure e regole per la logica NOR sono il corrispondente duale delle procedure viste per la logica NAND. Anche qui, le porte NOT rappresentate nei grafici di OR ed AND vanno intese come implementate con solo la porta NOR.

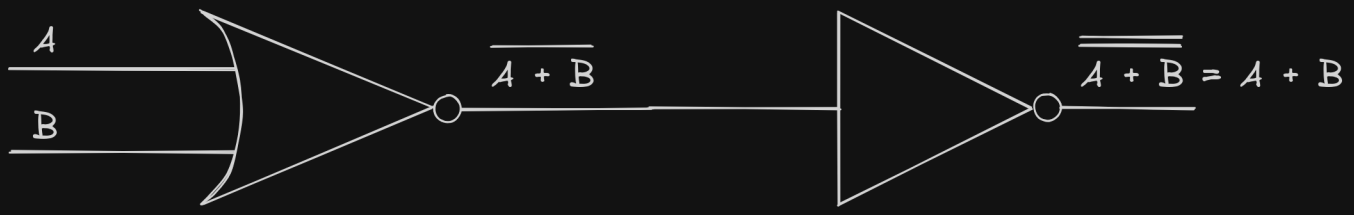
NOT



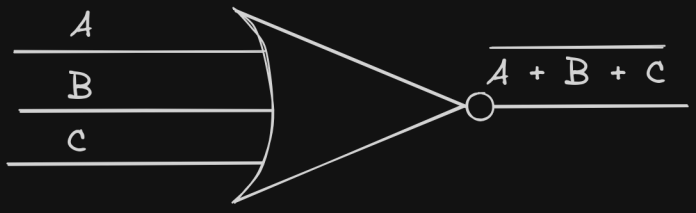
Simbolo grafico alternativo: NOT



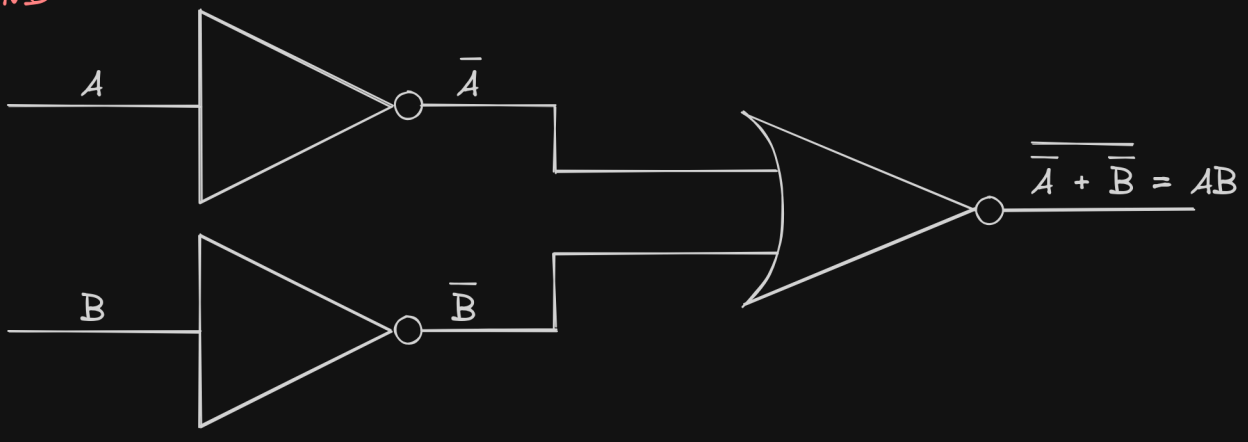
OR



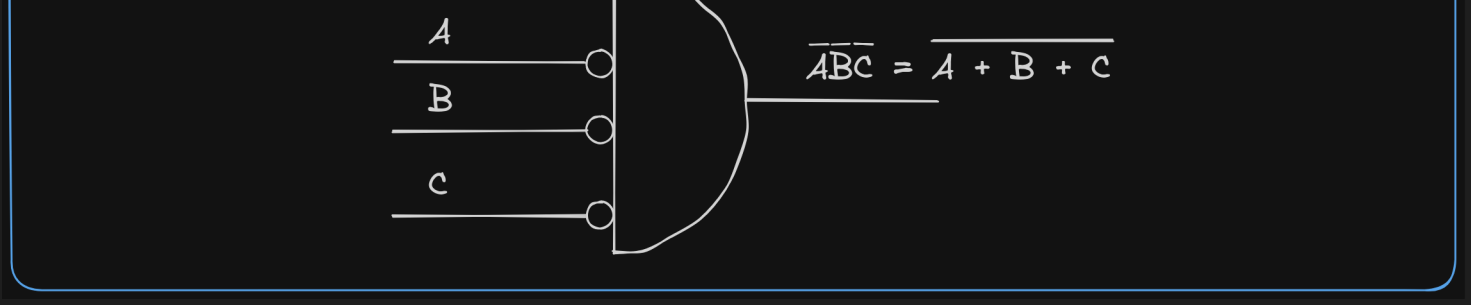
Simbolo grafico alternativo: OR-NOT



AND



Simbolo grafico alternativo: NOT-AND



Si possono applicare le relative sostituzioni in una espressione booleana per scriverla in forma All-NOR.

Implementazione a due livelli

L'implementazione di una funzione booleana con porte NOR risulta semplice se la funzione è espressa in forma POS, che genera quindi un circuito a due livelli: due porte OR (dette di primo livello) sono connesse ad una porta AND (detta di secondo livello). Eventuali invertitori sugli ingressi degli AND e/o sull'uscita dell'OR non sono considerati livelli aggiuntivi. Una volta progettato il circuito normalmente a due livelli, risulta più semplice ricavarsi il circuito con sole porte NOR.

Circuiti NOR multilivello

Esistono vari casi in cui il progetto di un sistema digitale richieda strutture con tre o più livelli. In questi casi, in genere si esprime la funzione booleana in termini di AND, OR e NOT, per poi implementare direttamente il circuito con tali porte, o convertendo prima in logica NOR.

La procedura generale per la conversione di diagrammi AND-OR multilivello in diagrammi in logica NOR, utilizzando i simboli alternativi, è tipicamente:

1) Converti tutte le porte AND in porte NOR con simbolo NOT-AND
2) Converti tutte le porte OR in porte NOR con simbolo OR-NOT
3) Controlla tutte le complementazioni nel diagramma, inserendo una porta NOT o una complementazione in ingresso per ogni "pallino" che non è bilanciato da un altro sulla stessa connessione.