Implementazione delle porte logiche CMOS

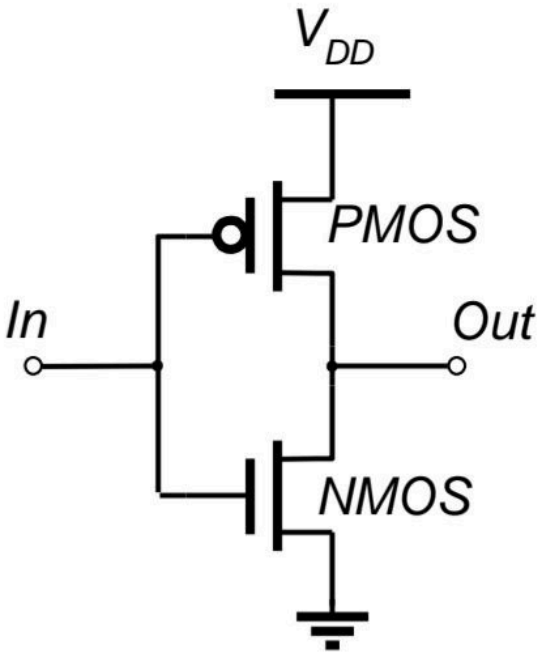
▼ Creatore originale: @Stefano Alverino

La tecnologia CMOS è fortemente utilizzata nella creazione di svariate porte logiche. Alla base di questa tecnologia vi è l'uso di due tipi di transistori: NMOS e PMOS.

CMOS

Il componente CMOS (Complementary MOS), detto anche inverter, è composto da due transistori complementari:

- un PMOS, avente il terminale di gate connesso all'ingresso, il terminale di source alla tensione V_{DD} e il terminale di drain all'uscita. Prende il nome di rete di pull-up;
- un NMOS avente il terminale di gate connesso all'ingresso, il terminale di source a ground e il terminale di drain all'uscita. Prende il nome di rete di pull-down;



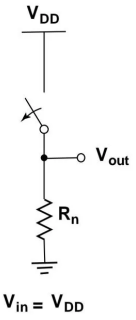
E' bene notare che il nome inverter è dovuto alla relazione ingresso-uscita di questo MOS:

$$\begin{cases} V_{\text{out}} = V_{\text{DD}} & \text{se } V_{\text{in}} < V_{\text{t}} \\ V_{\text{out}} = 0 & \text{se } V_{\text{in}} > V_{\text{t}} \end{cases}$$

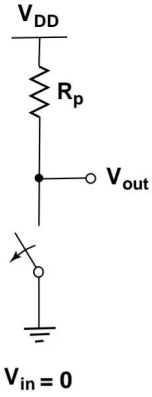
Comportamento del CMOS

I risultati ottenuti possono essere facilmente compresi analizzando il comportamento dei due MOS complementari:

- quando la tensione di ingresso V_{in} è alta (1), cioè supera il valore di soglia, e l'uscita assume un valore logico basso (0);
  - il PMOS si comporta come un circuito aperto;
  - l'NMOS si chiude e si comporta come una resistenza, collegando l'uscita V_{out} a massa.
- quando la tensione di ingresso V_{in} è bassa (0), quindi inferiore al valore di soglia, e l'uscita assume un valore logico alto (1);
  - l'NMOS si comporta come un circuito aperto;
  - il PMOS si chiude, collegando V_{out} a V_{DD}.



CMOS con tensione di ingresso alta



CMOS con tensione di ingresso bassa

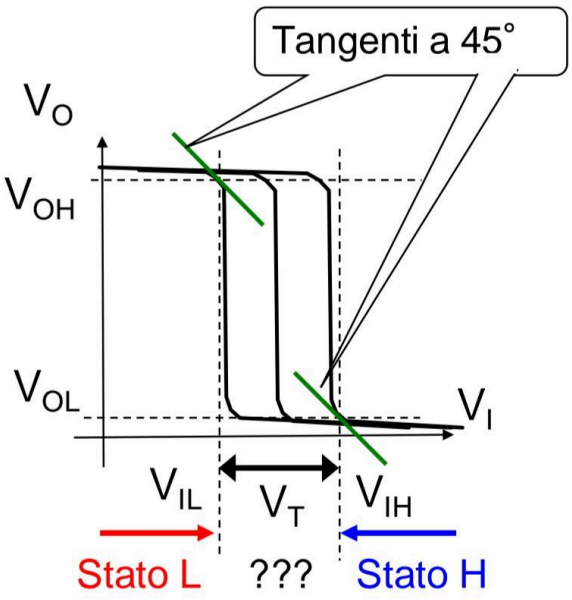
Definizione delle due soglie di ingresso e dell'uscita

La tensione di soglia V_{t} di un inverter CMOS può variare con l'alimentazione, la temperatura e altri tipi di disturbo, quindi è difficile determinarne un valore preciso.

Per gestire questa incertezza, si definiscono due soglie:

- tensione di soglia inferiore V_{IL}, e sotto questo valore l'ingresso è interpretato come 0 logico;
- tensione di soglia superiore V_{IH}, e sopra questo valore è interpretato come 1 logico.

Nel range $[V_{\mathrm{IL}}, V_{\mathrm{IH}}]$, lo stato logico non è definito, ed è quindi una zona instabile da evitare.



Visualizzazione delle tensioni di soglia per ingresso e uscita

L'uscita V_{O} può essere:

- uscita alta V_{OH} quando l'ingresso è basso;
- uscita alta V_{OL} quando l'ingresso è alto.

V_{IL} e V_{IH} sono, dal punto di vista grafico, i punti dove la curva ha una pendenza di -1, e quindi tangenti a 45°.

Calcolo della resistenza equivalente

Il calcolo della resistenza equivalente di un circuito basato su CMOS può essere effettuato tramite due approcci differenti:

▼ Approccio combinatorio:

In questo approccio si prevede di considerare tutti i possibili cammini che collegano V_{DD} (in caso di fronte di salita) o GND (in caso di fronte di discesa) all'uscita. Ogni MOS attraversato introduce una certa resistenza, sommando le

resistenze lungo ogni cammino e applicando le regole per serie e paralleli, si ottiene una resistenza equivalente per ciascun percorso.
Una volta calcolate tutte, si può selezionare quella minima o massima a seconda dello scenario da modellare.

Questo approccio è efficacie e preciso però diventa impegnativo in caso di grande numero di cammini

▼ Approccio euristico:

Questo approccio sfrutta le proprietà base delle reti resistive:
Le resistenze in parallelo riducono la resistenza totale, mentre quelle in serie la aumentano.
Quindi, per stimare la resistenza minima si considera il cammino che presenta il maggior numero di resistenze in parallelo e il minor numero di serie, mentre per la massima si fa l'opposto.

Pur essendo meno accurato, questo metodo è estremamente utile per un'analisi rapida e per avere un ordine di grandezza della resistenza complessiva.

> 🚨 Vi rimando ad un esercizio in cui viene utilizzato questo metodo per rendervi più chiare le idee.

Calcolo della capacità equivalente

Per capacità equivalente si intende la capacità totale vista dal nodo di uscita del circuito CMOS analizzato.
Per calcolarla si devono sommare tutti i contributi delle capacità di gate dei transistor MOS collegati direttamente all'uscita del circuito in esame.
Se l'uscita è collegata a
N porte logiche, e ognuna di queste possiede un numero M_i di gate MOS collegati a tale uscita, la capacità equivalente di carico può essere espressa come:

$$C_{eq} = \sum_{i=1}^{N} M_i * C_g$$

🚨 Vi rimando ad un esercizio in cui viene utilizzato questo metodo per rendervi più chiara e semplice la comprensione.

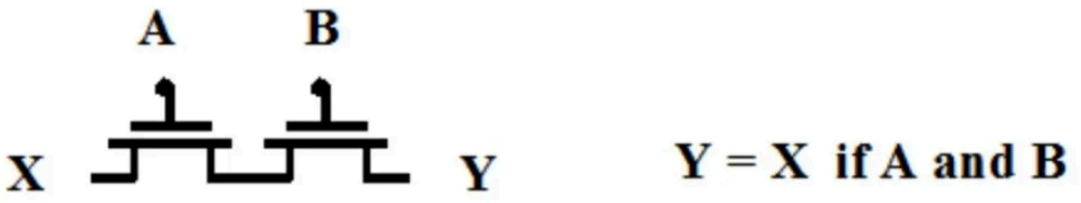
Implementazione di porte logiche con CMOS

Utilizzando due reti logiche di pull-up e pull-down, al posto dei singoli transistor (come nell'inverter), si possono creare diverse porte logiche.

⚠️ Quando la rete di pull-up risulta accesa, la rete di pull-down deve risultare spenta, e viceversa.

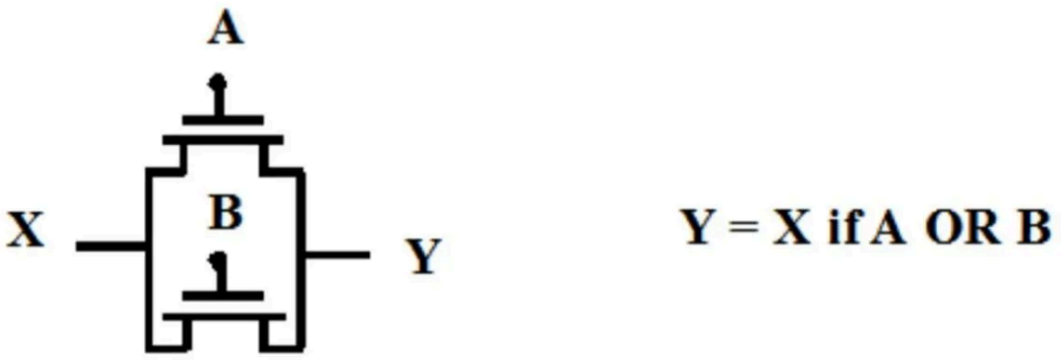
Per comprendere a pieno le implementazioni delle porte logiche, è bene analizzare il comportamento degli NMOS e dei PMOS in serie e in parallelo:

- due NMOS in serie lasciano arrivare il segnale X all'uscita Y se e solo se sia A che B presentano valore logico alto, creando una porta logica AND;



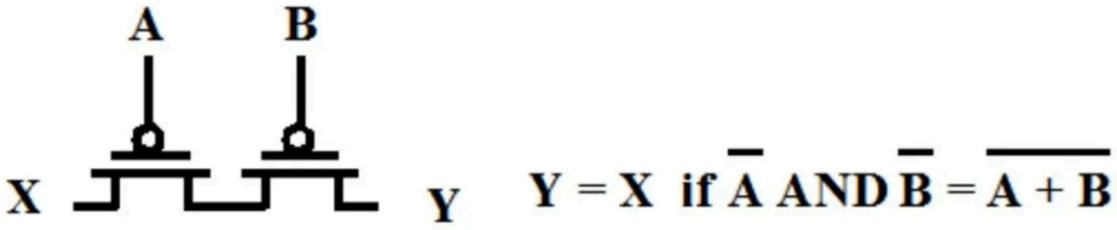
Implementazione di una porta logica AND con CMOS

- due NMOS in parallelo lasciano arrivare il segnale X all'uscita Y se almeno uno tra A e B presenta un valore logico alto, creando una porta logica OR;



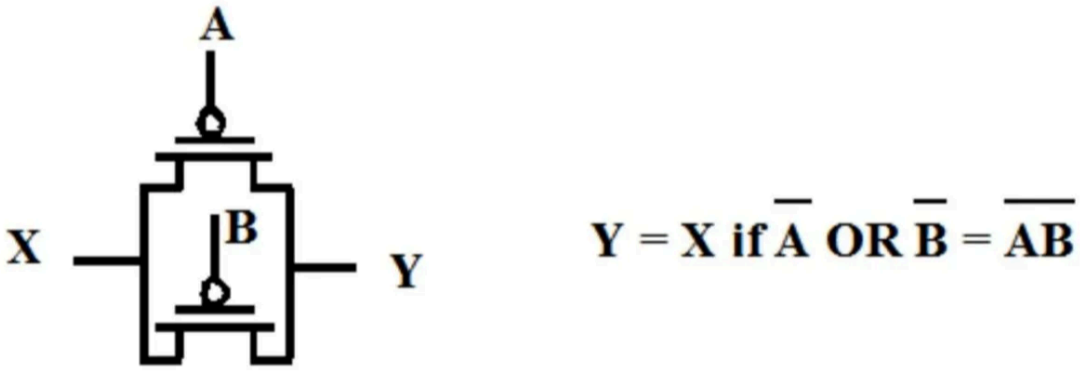
Implementazione di una porta logica OR con CMOS

- due PMOS in serie lasciano arrivare il segnale X all'uscita Y se e solo se sia A che B presentano valore logico basso, creando, tramite le leggi di De Morgan, una porta logica NOR;



Implementazione di una porta logica NOR con CMOS

- due PMOS in parallelo lasciano arrivare il segnale X all'uscita Y se almeno uno tra A e B presenta un valore logico basso, creando, tramite le leggi di De Morgan, una porta logica NAND.



Implementazione di una porta logica NAND con CMOS

Correlazione tra rete di pull-down e rete di pull-up

Per costruire la rete di pull-down a partire da quella di pull-up, bisogna:

1. scrivere la funzione logica implementata dalla rete di pull-up;
2. negare questa funzione logica usando le leggi di De Morgan;
3. utilizzare la funzione risultante per costruire la rete di pull-down con transistori NMOS.

In pratica, si realizza la funzione complementare di quella della rete di pull-up, ma con struttura opposta:

- il PMOS in serie diventa NMOS in parallelo;
- il PMOS in parallelo diventa NMOS in serie.

Per costruire la rete di pull-up da quella di pull-down si può seguire il medesimo procedimento.

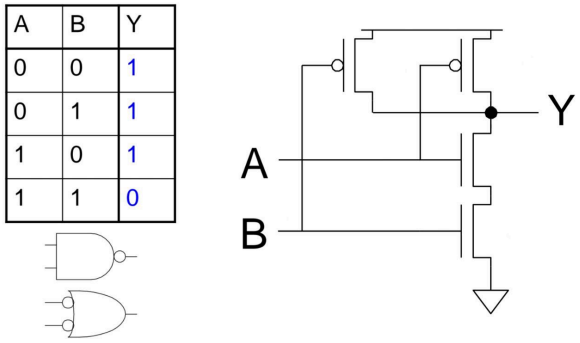
NAND a 2 ingressi

La rete di pull-up è composta da 2 PMOS in parallelo, mentre la rete di pull-down è composta da 2 NMOS in serie.

Vogliamo che la nostra porta logica restituisca il valore logico 0 solo quando entrambi gli ingressi presentano valore logico 1.

$$Y = \overline{A \cdot B}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



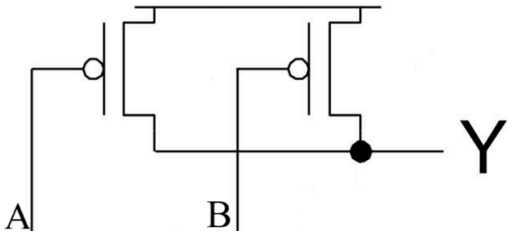
Implementazione della porta logica NAND a 2 ingressi con CMOS

Procediamo alla costruzione della porta logica NAND, ricordando che la rete di pull-up è quella che produce i valori logici alti sull'uscita:

1. partendo dal NAND, applichiamo De Morgan.

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

L'OR dei due valori negati è rappresentato dal parallelo di due PMOS, formando la rete di pull-up;

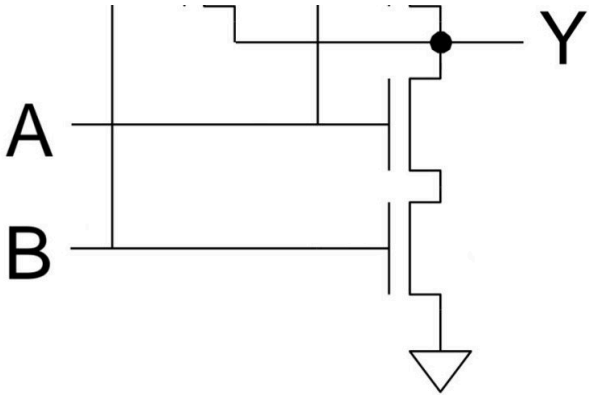


Rete di pull-up

2. partendo dalla rete di pull-up, possiamo iniziare negando la sua funzione logica.

$$\overline{\overline{A \cdot B}} = A \cdot B$$

L'AND di due valori è rappresentato dalla serie di due NMOS, formando la rete di pull-down.



Rete di pull-down

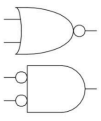
NOR a 2 ingressi

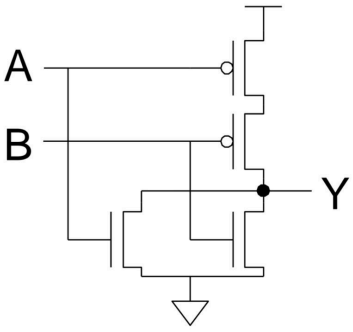
La rete di pull-up è composta da 2 PMOS in serie, mentre la rete di pull-down è composta da 2 NMOS in parallelo.

Vogliamo che la nostra porta logica restituisca il valore logico 1 solo quando entrambi gli ingressi presentano valore logico 0.

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

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0





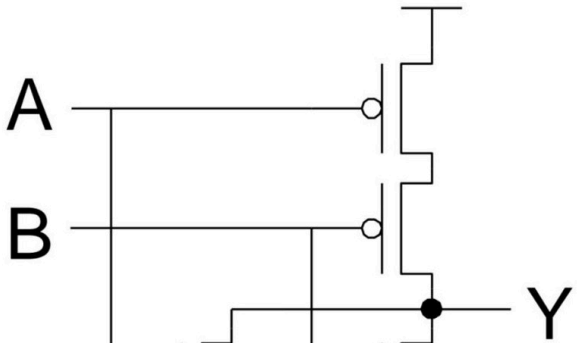
Implementazione della porta logica NOR a 2 ingressi con CMOS

Procediamo alla costruzione della porta logica NOR, ricordando che la rete di pull-up è quella che produce i valori logici alti sull'uscita:

1. partendo dal NOR, applichiamo De Morgan.

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

L'AND dei due valori negati è rappresentato dalla serie di due PMOS, formando la rete di pull-up;

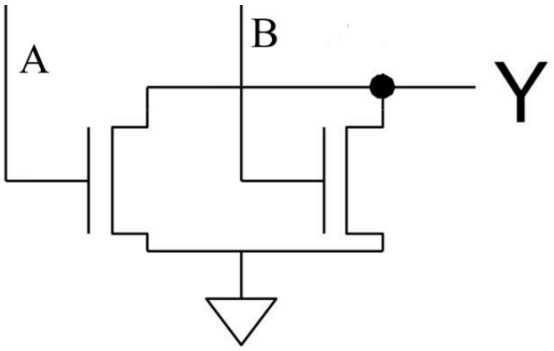


Rete di pull-up

2. partendo dalla rete di pull-up, possiamo iniziare negando la sua funzione logica.

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

L'OR di due valori è rappresentato dal parallelo di due NMOS, formando la rete di pull-down.



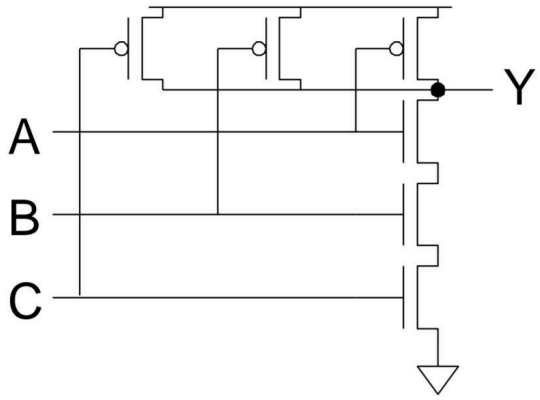
Rete di pull-down

NAND a 3 ingressi

La rete di pull-up è composta da 3 PMOS in parallelo, mentre la rete di pull-down è composta da 3 NMOS in serie.

Vogliamo che la nostra porta logica restituisca il valore logico 0 solo quando tutti gli ingressi presentano valore logico 1.

$$Y = \overline{A \cdot B \cdot C}$$



Implementazione della porta logica NAND a 3 ingressi con CMOS

Il processo di costruzione della rete di pull-up e di quella di pull-down è il medesimo del NAND a 2 ingressi, ma inserendo un MOS aggiuntivo per ogni rete, in modo tale da gestire il terzo ingresso.

Esempio - costruzione di una funzione logica

Obiettivo

Si consideri la funzione logica U.

$$U = A + B$$

Si implementi tale funzione logica tramite porte CMOS.

▼ Implementazione della funzione logica

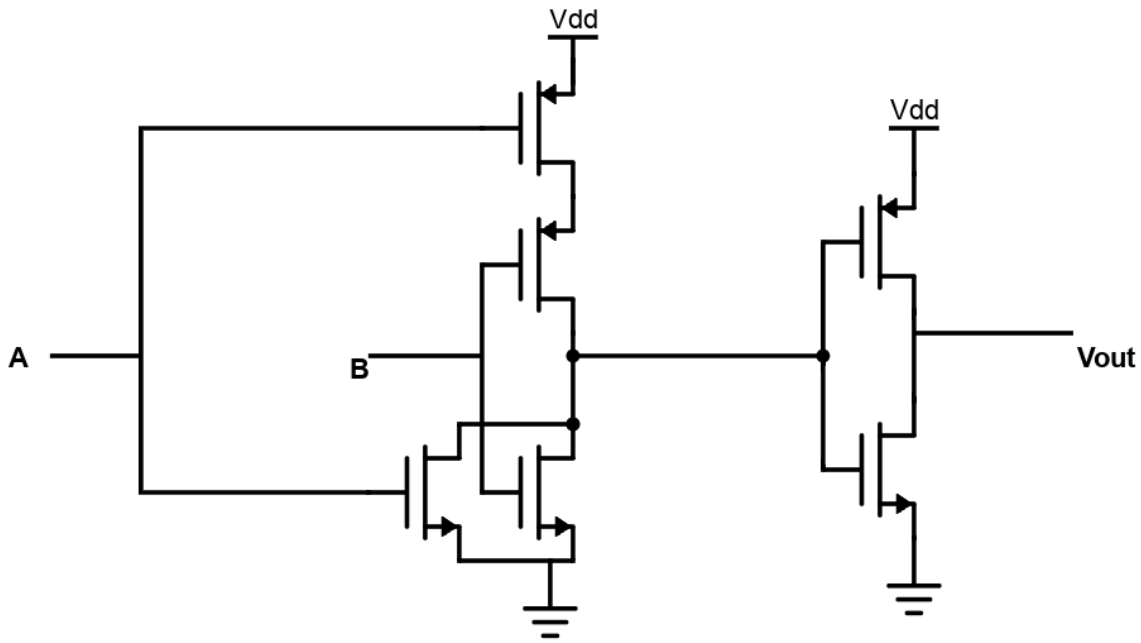
Dobbiamo ricondurci a una funzione logica del seguente tipo:

$$f = \overline{A <\text{operazione}> B}$$

La prima cosa da fare è considerare:

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

In questo caso particolarmente fortunato, possiamo utilizzare due porte logiche note per implementare tale circuito, poiché $\overline{A + B}$ rappresenta una porta NOR a cui, però, dobbiamo aggiungere un inverter, definito dall'altra negazione, in modo da raggiungere la funzione $\overline{\overline{A + B}}$.



Il primo blocco a sinistra è la realizzazione della porta NOR, mentre il secondo blocco rappresenta l'inverter

Esempio - costruzione di una funzione logica

Obiettivo

Si consideri la funzione logica U.

$$U = \overline{A} \cdot \overline{B} + \overline{C}$$

Si implementi tale funzione logica tramite porte CMOS.

▼ Implementazione della funzione logica

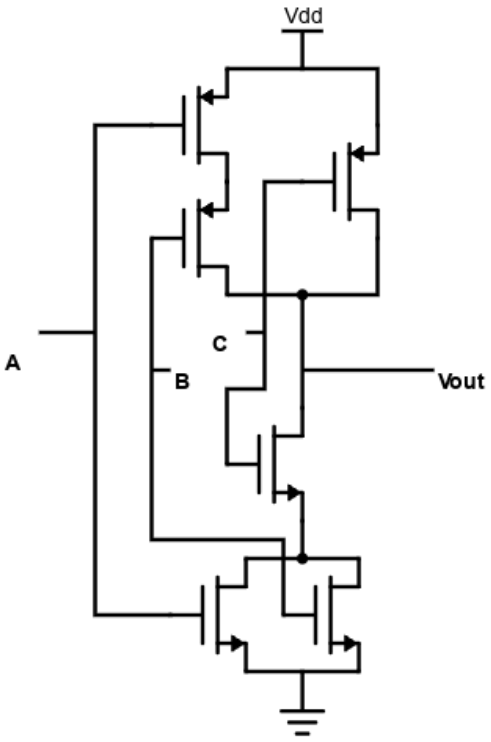
Applicando le leggi di De Morgan:

$$\overline{A} \cdot \overline{B} + \overline{C} = \overline{A + B} + \bar{C}$$

Riapplicando nuovamente le leggi di De Morgan, otteniamo:

$$\overline{A + B} + \bar{C} = \overline{(A + B) \cdot C}$$

Tale funzione logica è ora pienamente realizzabile attraverso porte logiche con implementazione CMOS.



Implementazione della porta logica richiesta con CMOS