Implementazione delle porte logiche CMOS

▼ Creatore originale: @Stefano Alverino

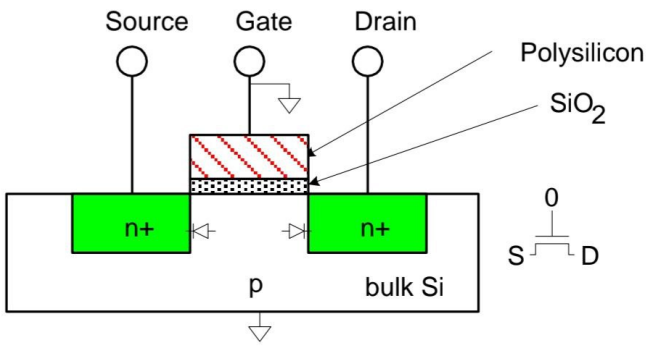
La tecnologia CMOS è fortemente utilizzata nella creazione di svariate porte logiche. Alla base di questa tecnologia vi è l'uso di due tipi di transistori: NMOS e PMOS.

Prima di spiegare l'implementazione delle porte logiche, è bene fare chiarezza sulla differenza tra NMOS e PMOS e sull'implementazione dei CMOS.

NMOS

Il transistore MOS a canale n (NMOS) presenta quattro terminali: Gate, Source, Drain e Body (quest'ultimo è a potenziale fisso, tipicamente a ground).

Gate e Body sono due conduttori separati da un isolante, e questa disposizione permette la creazione di un condensatore MOS (metallo-ossido-semiconduttore).



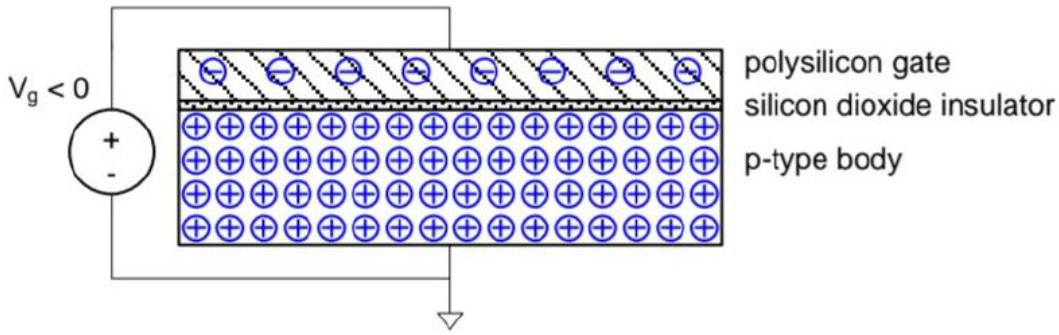
Visualizzazione di un transistore NMOS

Il transistore presenterà, quindi, diversi modi di funzionamento, a seconda della tensione VgV_gVg tra gate e body:

- accumulo: Vg<0V_{\text g} < 0Vg < 0;
- svuotamento: 0<Vg<Vt0 < V_\text g < V_\text t0 < Vg < Vt;
- inversione: Vg>VtV_\text g > V_\text t Vg > Vt.

Fase di accumulo

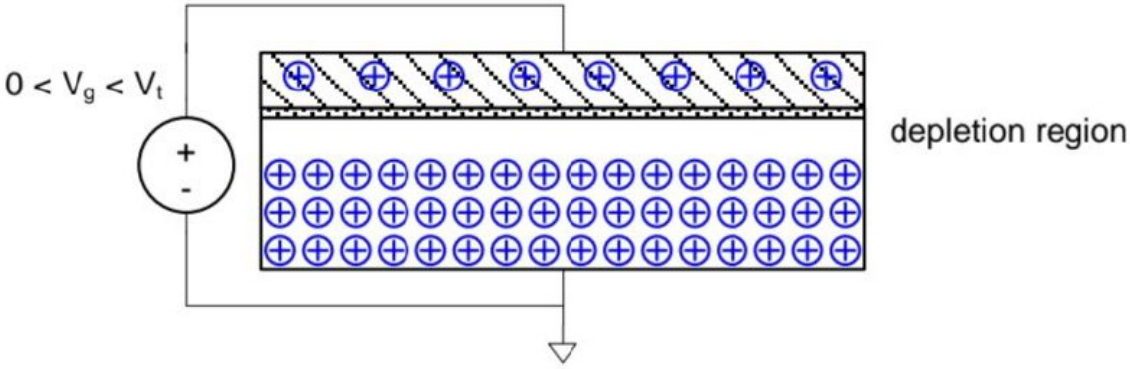
Nella fase di accumulo, le cariche positive presenti nel Body vengono attratte dalle negative poste sul gate, creando così un "muro" che blocca il passaggio di corrente.



Visualizzazione della fase di accumulo

Fase di svuotamento

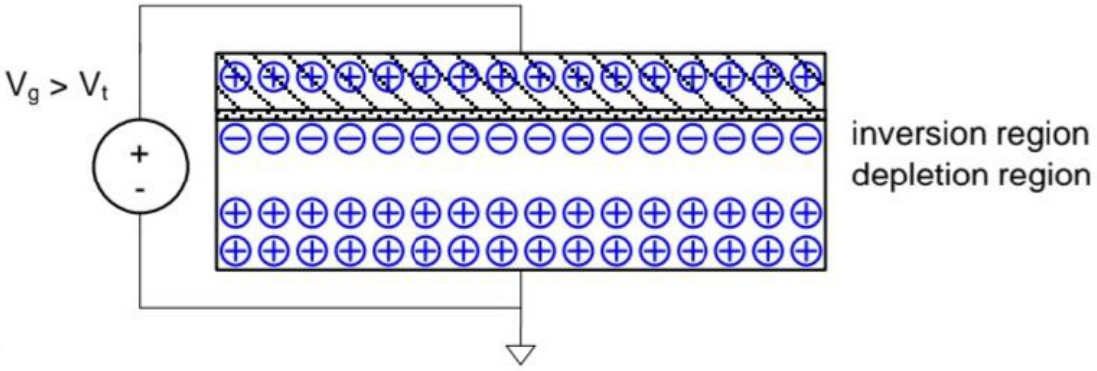
Quando VgV_\text gVg supera lo 0, si passa alla fase di svuotamento, in cui il potenziale positivo tende a portare cariche negative dal body al gate, ma dove la tensione non è abbastanza elevata.



Visualizzazione della fase di svuotamento

Fase di inversione

Quando VgV_\text gVg supera VtV_\text tVt (tensione di soglia o threshold), si passa alla fase di inversione, dove le cariche negative del body vengono effettivamente portate al gate, producendo un "canale" di passaggio per la corrente tra il terminale di drain e il terminale di source.



Visualizzazione della fase di inversione

Commento sul funzionamento

In un transistore NMOS, quando la tensione del gate è inferiore alla tensione di soglia, il passaggio di corrente tra source e drain è bloccato, producendo così un valore logico 0 (OFF).

Al contrario, quando la tensione del gate supera la tensione di soglia, il passaggio di corrente è abilitato, generando un valore logico 1 (ON).

{OFFse Vg<VtONse Vg>Vt\begin{cases}

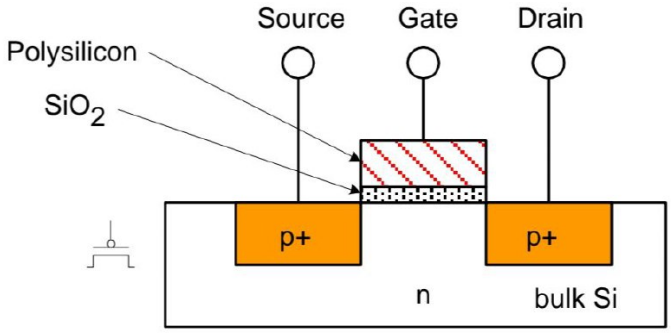
\text{OFF} & \text{se } V_\text g < V_\text t \\
\text{ON} & \text{se } V_\text g > V_\text t

\end{cases}{OFFONse Vg < Vtse Vg > Vt

PMOS

Il transistore MOS a canale p (PMOS) presenta quattro terminali: Gate, Source, Drain e Body (quest'ultimo è a potenziale fisso, tipicamente a tensione alta VDDV_{DD}VDD).

Gate e Body sono due conduttori separati da un isolante, questa disposizione permette la creazione di un condensatore MOS (metallo-ossido-semiconduttore).



Visualizzazione di un transistore PMOS

Commento sul funzionamento

Il comportamento è esattamente inverso all'NMOS.

Per una tensione di gate VgV_\text gVg inferiore alla tensione di soglia VtV_\text tVt avverrà un passaggio di corrente tra drain e source, producendo il valore logico 1 (ON).

Per una tensione di gate superiore alla tensione di soglia non verrà consentito il passaggio di corrente, producendo un valore logico 0 (OFF).

{ONse Vg<VtOFFse Vg>Vt\begin{cases}

\text{ON} & \text{se } V_\text g < V_\text t \\
\text{OFF} & \text{se } V_\text g > V_\text t

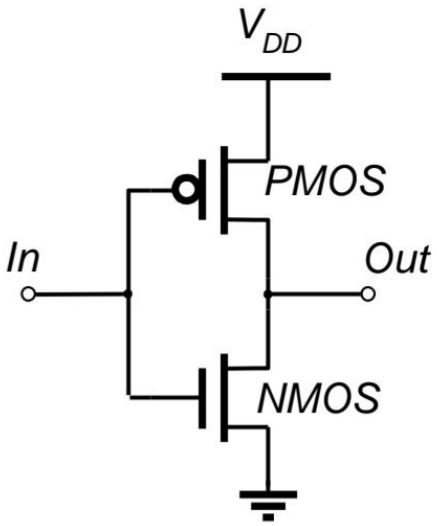
\end{cases}{ONOFFse Vg < Vtse Vg > Vt

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 VDDV_{\text{DD}}VDD 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:

{Vout=VDDse Vin<VtVout=0se Vin>Vt\begin{cases}

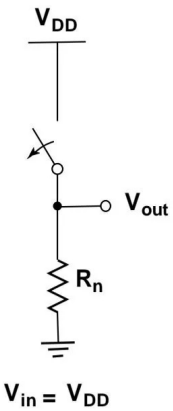
V_{\text{out}} = V_{\text{DD}} & \text{se } V_{\text{in}} < V_\text t \\
V_{\text{out}} = 0 & \text{se } V_{\text{in}} > V_\text t

\end{cases}{Vout = VDDVout = 0se Vin < Vtse Vin > Vt

Comportamento del CMOS

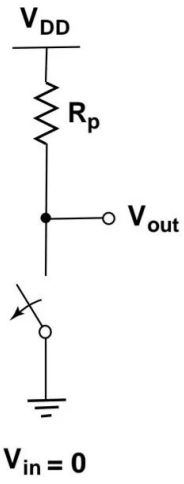
I risultati ottenuti possono essere facilmente compresi analizzando il comportamento dei due MOS complementari:

- quando la tensione di ingresso VinV_{\text{in}}Vin 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 VoutV_{\text{out}}Vout a massa.
- quando la tensione di ingresso VinV_{\text{in}}Vin 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 VoutV_{\text{out}}Vout a VDDV_{\text{DD}}VDD.



CMOS con tensione di

ingresso alta



CMOS con tensione di ingresso bassa

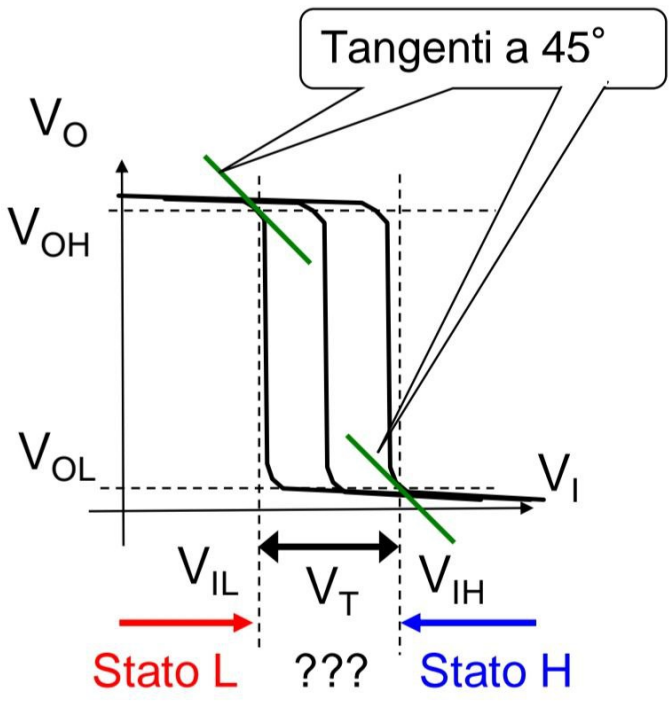
Definizione delle due soglie di ingresso e dell'uscita

La tensione di soglia VtV_\text tVt 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 VILV_{\text{IL}}VIL, e sotto questo valore l'ingresso è interpretato come 0 logico;
- tensione di soglia superiore VIHV_{\text{IH}}VIH, e sopra questo valore è interpretato come 1 logico.

Nel range [VIL,VIH][V_{\text{IL}}, V_{\text{IH}}][VIL, VIH], 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 VOV_\text{O}VO può essere:

- uscita alta VOHV_{\text{OH}}VOH quando l'ingresso è basso;
- uscita alta VOLV_{\text{OL}}VOL quando l'ingresso è alto.

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

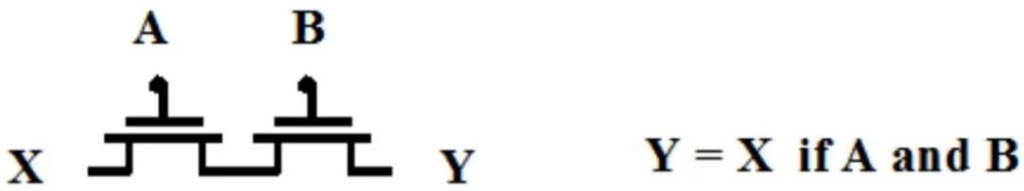
Implementazione di porte logiche con CMOS

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.

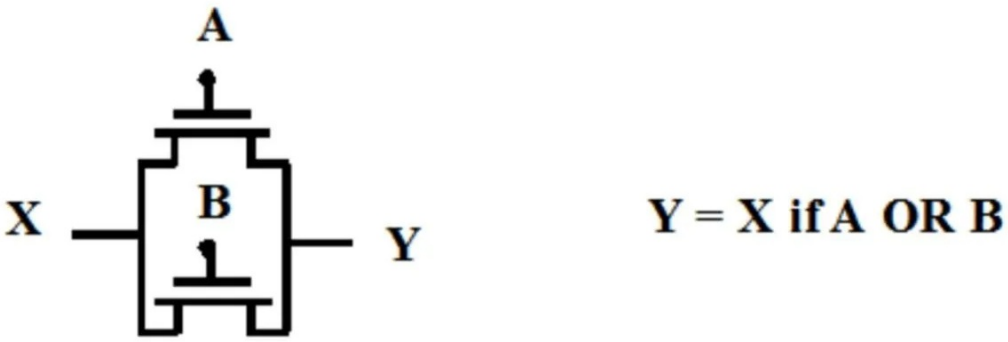
Per comprendere a pieno le implementazioni delle porte logiche, è bene analizzare il comportamento degli NMOS e dei PMOS in serie e in parallelo:

- due NMOS in serie lasciano arrivare il segnale XXX all'uscita YYY se e solo se sia AAA che BBB presentano valore logico alto, creando una porta logica AND;



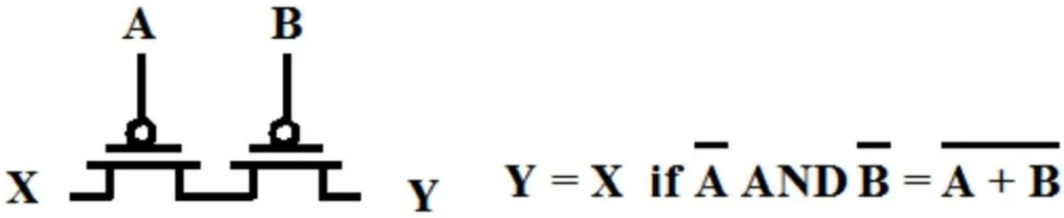
Implementazione di una porta logica AND con CMOS

- due NMOS in parallelo lasciano arrivare il segnale XXX all'uscita YYY se almeno uno tra AAA e BBB presenta un valore logico alto, creando una porta logica OR;



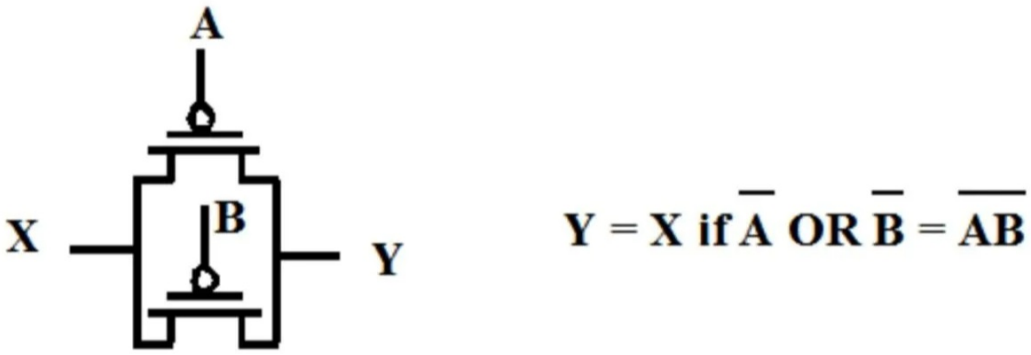
Implementazione di una porta logica OR con CMOS

- due PMOS in serie lasciano arrivare il segnale XXX all'uscita YYY se e solo se sia AAA che BBB 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 XXX all'uscita YYY se almeno uno tra AAA e BBB 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.

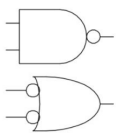
NAND a 2 ingressi

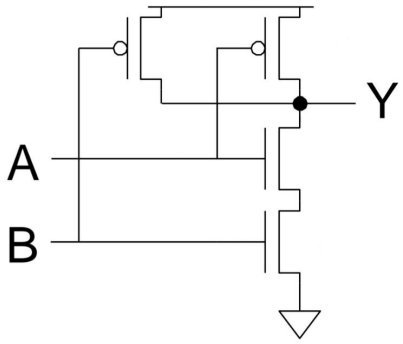
La rete di pull-up è composta da 2 PMOS in parallelo, mentre la rete di pull-down è composta da 2 NMOS in serie.

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

Y=A⋅B¯Y = \overline {A \cdot B}Y = A ⋅ B

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





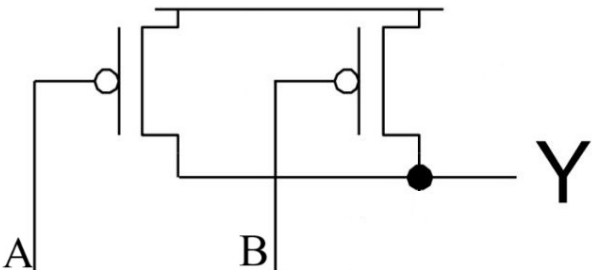
Implementazione della porta logica NAND a 2 ingressi con CMOS

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.

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

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

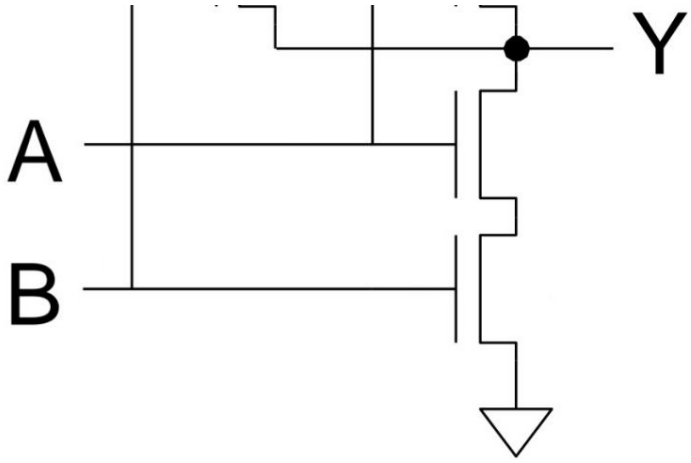


Rete di pull-up

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

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

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



Rete di pull-down

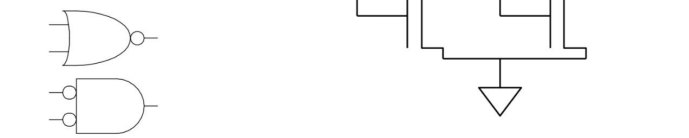
NOR a 2 ingressi

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.

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

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



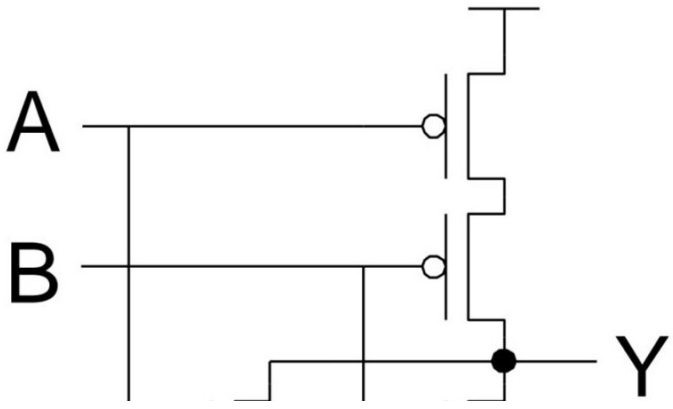
Implementazione della porta logica NOR a 2 ingressi con CMOS

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.

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

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



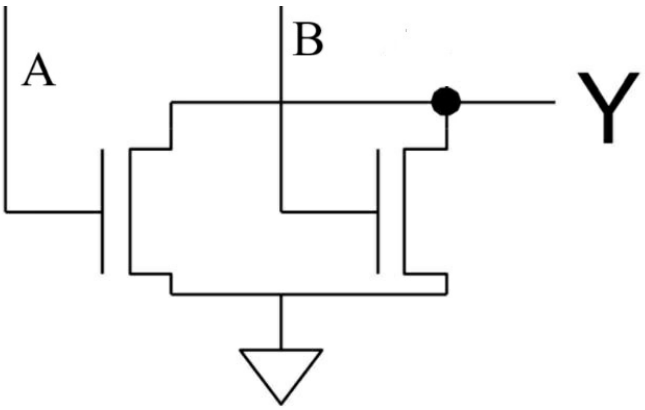
Rete di pull-up

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

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

   L'OR di due valori è rappresentato dal parallelo di due

NMOS, formando la rete di pull-down.



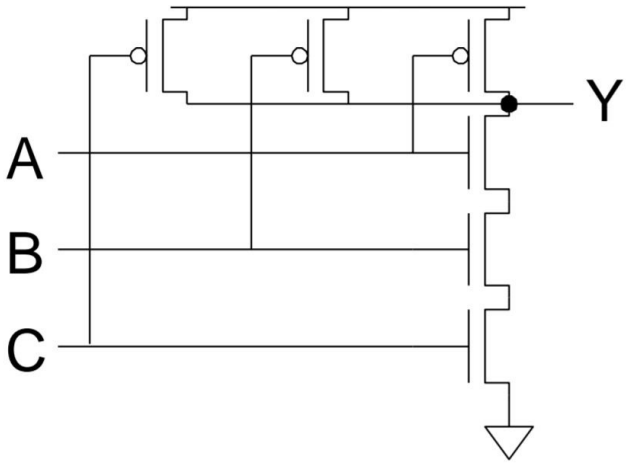
Rete di pull-down

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.

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



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 UUU.

$U = A + B$ $U = A + B$ $U = A + B$

Si implementi tale funzione logica tramite porte CMOS.

▼ Implementazione della funzione logica

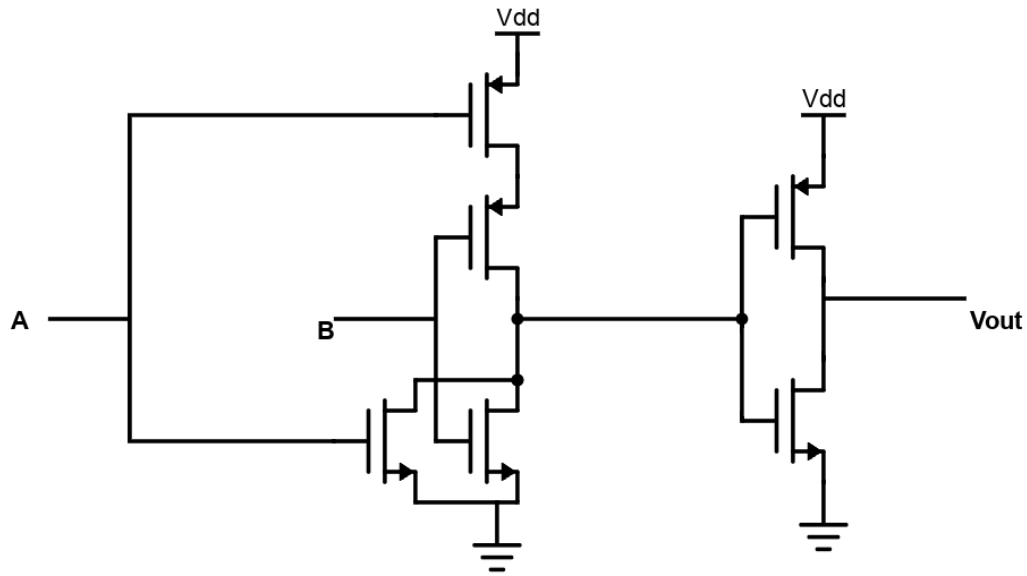
Dobbiamo ricondurci a una funzione logica del seguente tipo:

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

La prima cosa da fare è considerare:

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

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



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 UUU.

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

Si implementi tale funzione logica tramite porte CMOS.

▼ Implementazione della funzione logica

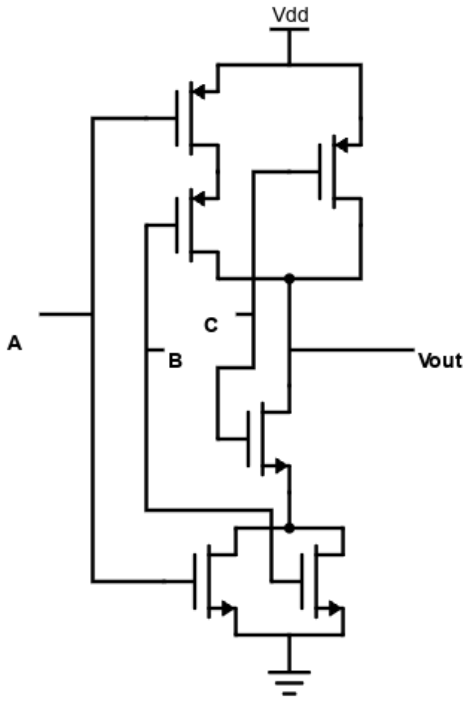
Applicando le leggi di De Morgan:

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

Riapplicando nuovamente le leggi di De Morgan, otteniamo:

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

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



Implementazione della porta logica richiesta con CMOS