
Lezione 90

Elementi di Progettazione Digitale

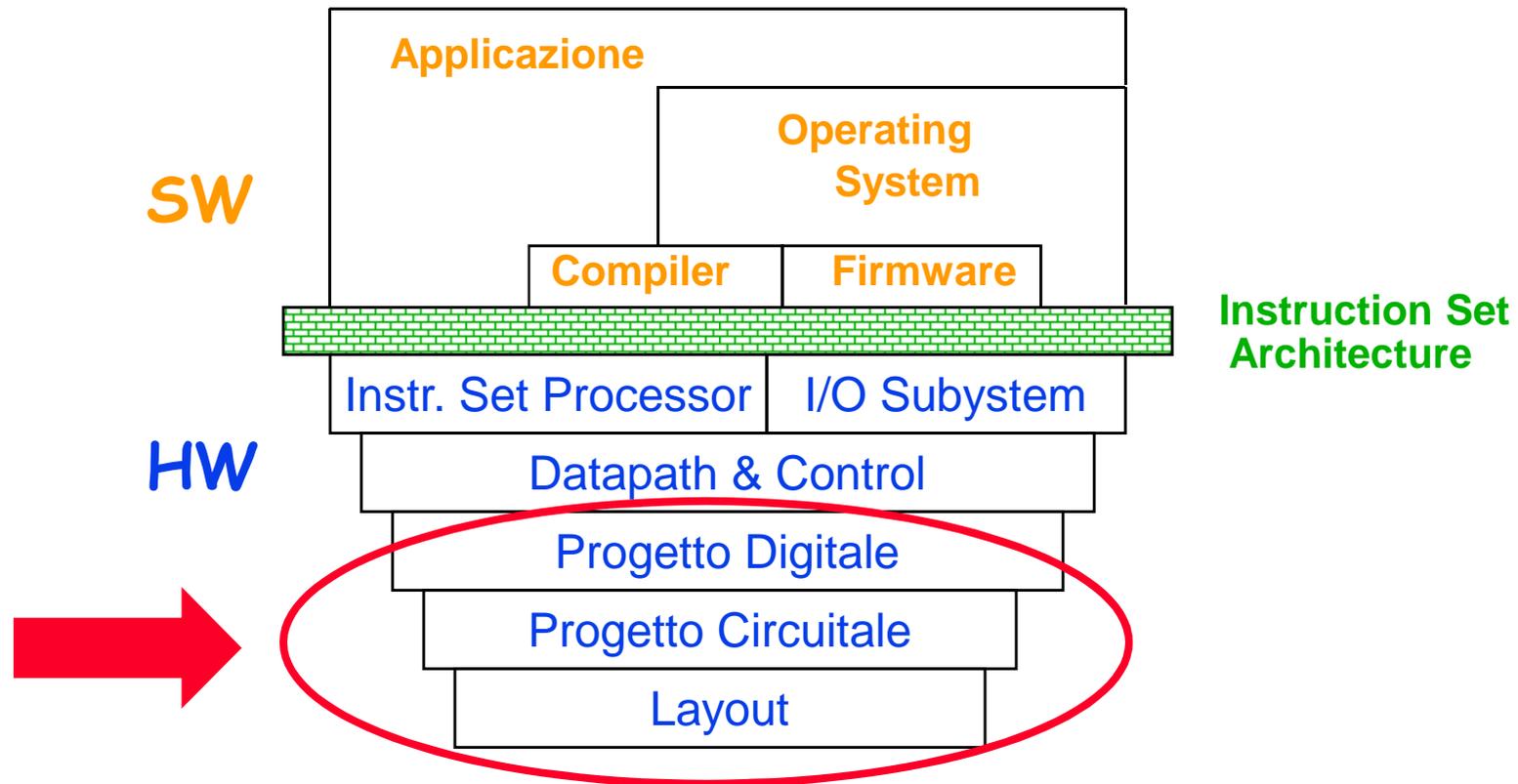
<http://www.dii.unisi.it/~giorgi/didattica/arc1>

All figures from Computer Organization and Design: The Hardware/Software Approach, Second Edition, by David Patterson and John Hennessy, are copyrighted material. (COPYRIGHT 1998 MORGAN KAUFMANN PUBLISHES, INC. ALL RIGHTS RESERVED.) Figures may be reproduced only for classroom or personal educational use in conjunction with the book and only when the above copyright line is included. They may not be otherwise reproduced, distributed, or incorporated into other works without the prior written consent of the publisher.

Other material is adapted from CS61C, CS152 Copyright (C) 2000 UCB

Progettazione Logica

Livelli di Astrazione di un Calcolatore



- Come si passa dal progetto digitale al layout ?
- Quali possibilita' offre l'attuale tecnologia ?

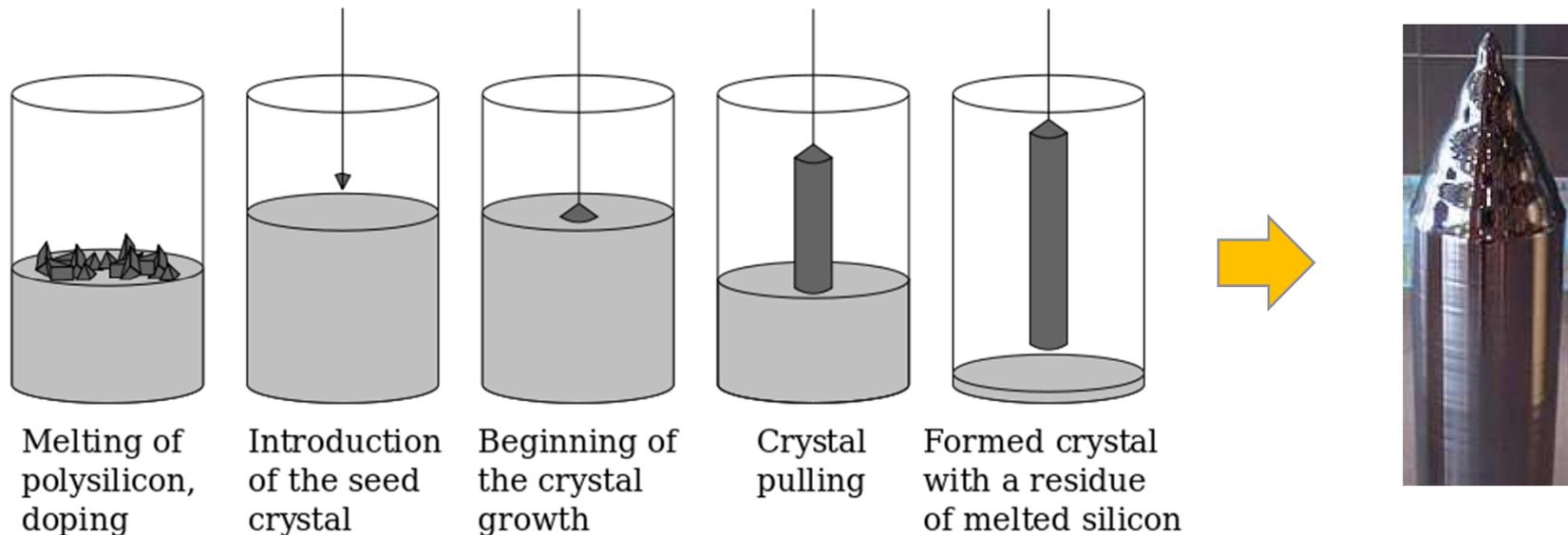
Approccio Bottom-Up: dal livello fisico all'architettura

- Preparazione del wafer di silicio
- Generazione dei dispositivi sul wafer
- Caratteristiche dell'inverter CMOS
- Altre porte: NAND, NOR, Transmission Gate

Il processo e' legato alla tecnologia:
nel periodo 1980-2015 questa e'
la metodologia consolidata

Preparazione del wafer di silicio

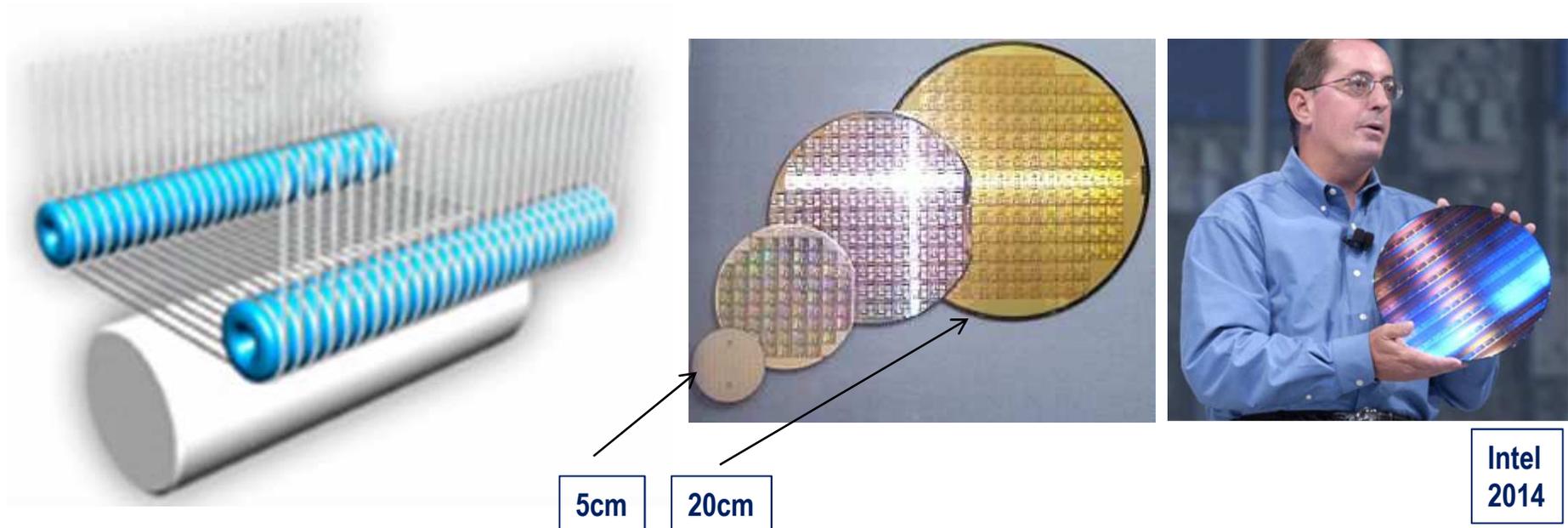
- Produzione di un lingotto di silicio monocristallino (es. Metodo di crescita di Czochralski) da silicio grezzo (quarzite)
- $\text{SiO}_2 + \text{C} \rightarrow \text{Si} + \text{CO}_2$ (in forno a 1500 C, purezza 98%)
- $\text{SiHCl}_3 + \text{H}_2 \rightarrow \text{Si} + 3\text{HCl}$ (silicio purissimo 99.9999999%)



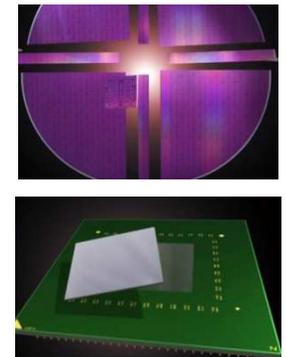
- Il diametro del cilindro così generato variano fra 10 cm e 45 cm circa

Dal lingotto al wafer

- Il lingotto viene sezionato in cosiddetti "wafer" ovvero dischi di spessore tra i $270\ \mu\text{m}$ a $920\ \mu\text{m}$ circa

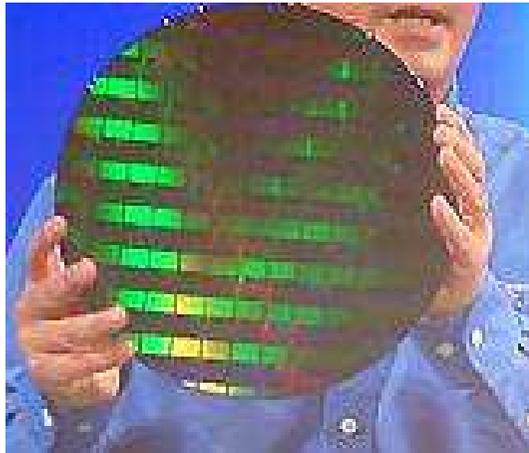


- Il wafer viene successivamente sottoposto a lappatura (etching), lucidatura e attacco con acido per ridurre imperfezioni e rugosità
- Il wafer viene successivamente sottoposto a lunghi processi per generare i dispositivi (transistor CMOS) e successivamente i chip vengono tagliati, verificati e inseriti nei package

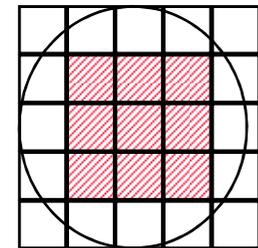
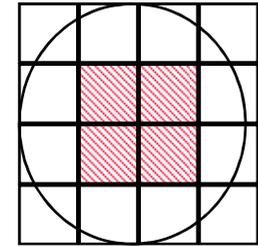


Perche' non si fanno chip "grossi"

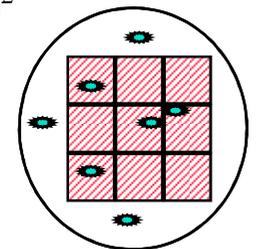
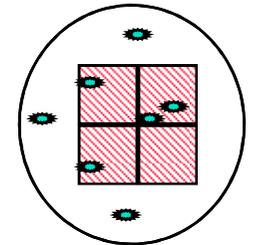
$$C_{DIE} = \frac{C_{WAFER}}{N_{DIE} \cdot Y_{WAFER}}$$



- C_{DIE} = Costo del 'die' (del 'chip')
- C_{WAFER} = Costo del wafer
- N_{DIE} = Numero di 'die' in un wafer
- Y_{WAFER} = 'Yield' o Resa del wafer
(numero 'die' per unita' di sup.)
- D_{WAFER} = Diametro del wafer
- A_{DIE} = Area del 'die'
- A_{WAFER} = Area del wafer = $\pi \cdot (D_{WAFER}/2)^2$
- N_{TEST} = Numero di 'die' usati per test
- F = Difetti per unita' di superficie



- N. di chip "teorici" sul wafer
- N. di chip sul bordo del wafer
- N. di chip usati per test "distruttivi"



$$N_{DIE} = \frac{\pi \cdot \left(\frac{D_{WAFER}}{2}\right)^2}{A_{DIE}} - \frac{\pi \cdot D_{WAFER}}{\sqrt{2} \cdot A_{DIE}} - N_{TEST} \cong \frac{A_{WAFER}}{A_{DIE}}$$

Inoltre: $Y_{WAFER} = \frac{1}{1 + (F \cdot A_{DIE}/2)^2}$

$$\rightarrow C_{DIE} = \frac{C_{WAFER} \cdot A_{DIE} \cdot \left[1 + (F \cdot A_{DIE}/2)^2\right]}{A_{WAFER}} \propto \frac{A_{DIE}^3}{A_{WAFER}}$$

Il costo di un chip e' proporzionale a circa il cubo della sua area!

Generazione dei dispositivi sul wafer

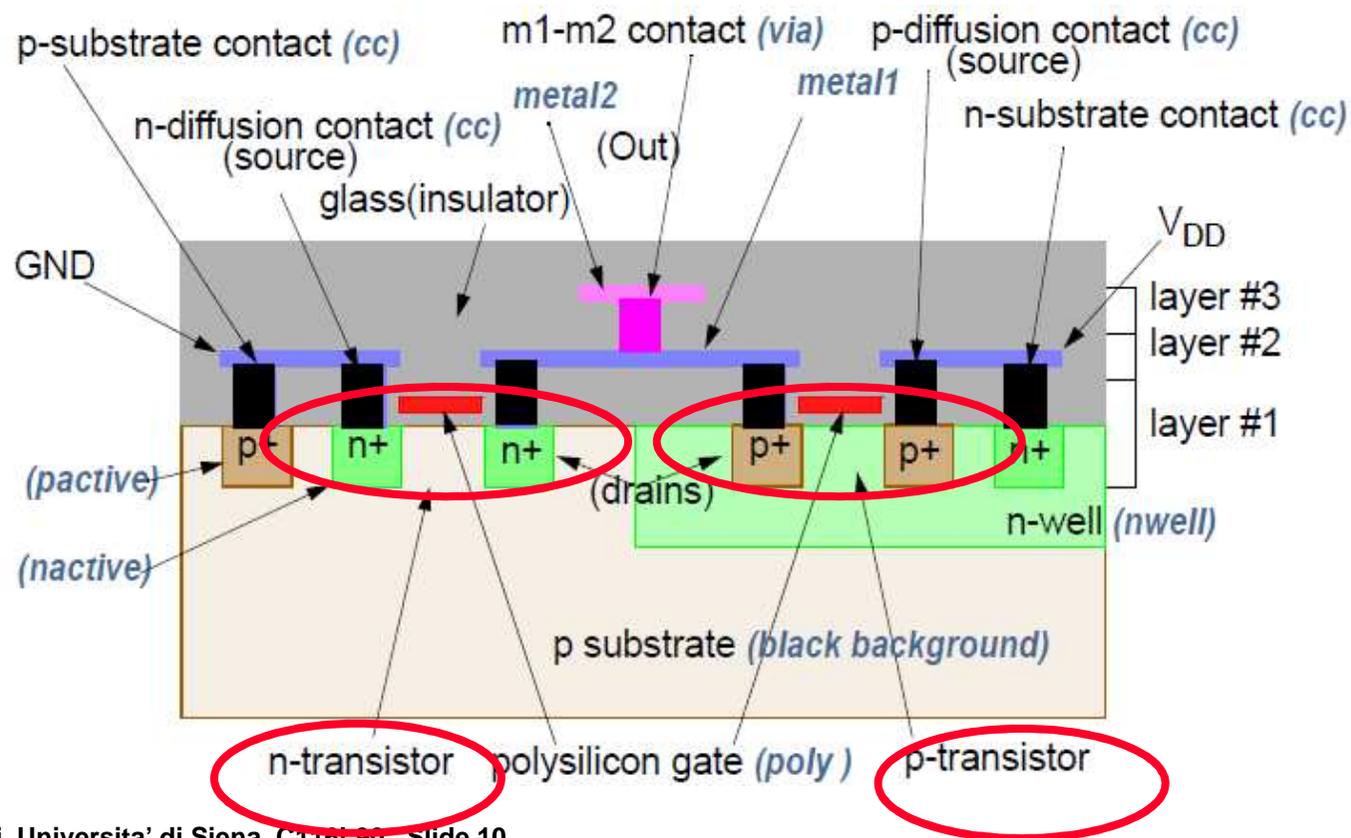
- Nel caso di chip usati per sistemi digitali (es. CPU), i dispositivi generati sul wafer consistono quasi esclusivamente in transistor e collegamenti
 - I transistor vengono realizzati con la tecnologia "MOS" (Metal Oxide Semiconductor)
 - I collegamenti possono essere realizzati sia con metallo che con polisilicio
- I transistor sono principalmente di due categorie:
 - Transistor di logica
 - Transistor di passo
- Inoltre per rendere il funzionamento piu' efficiente viene adottata la tecnologia CMOS (Complementary MOS) che sfrutta la possibilita' di abbinare transistor "di tipo N" (NMOS) a transistor "di tipo P" (PMOS)

MOS: Metal Oxide Semiconductor

- I transistor sono generati sul substrato di Silicio (semiconduttore)
- Il silicio puro non ha portatori di carica liberi ed e' un cattivo conduttore
- Il processo di drogaggio aumenta la conducibilita':
 - Con diffusioni di tipo 'n' (es. Fosforo) si dispone di elettroni in eccesso (n-well)
 - Con diffusioni di tipo 'p' (es. Boro) si dispone di lacune in eccesso (p-well)

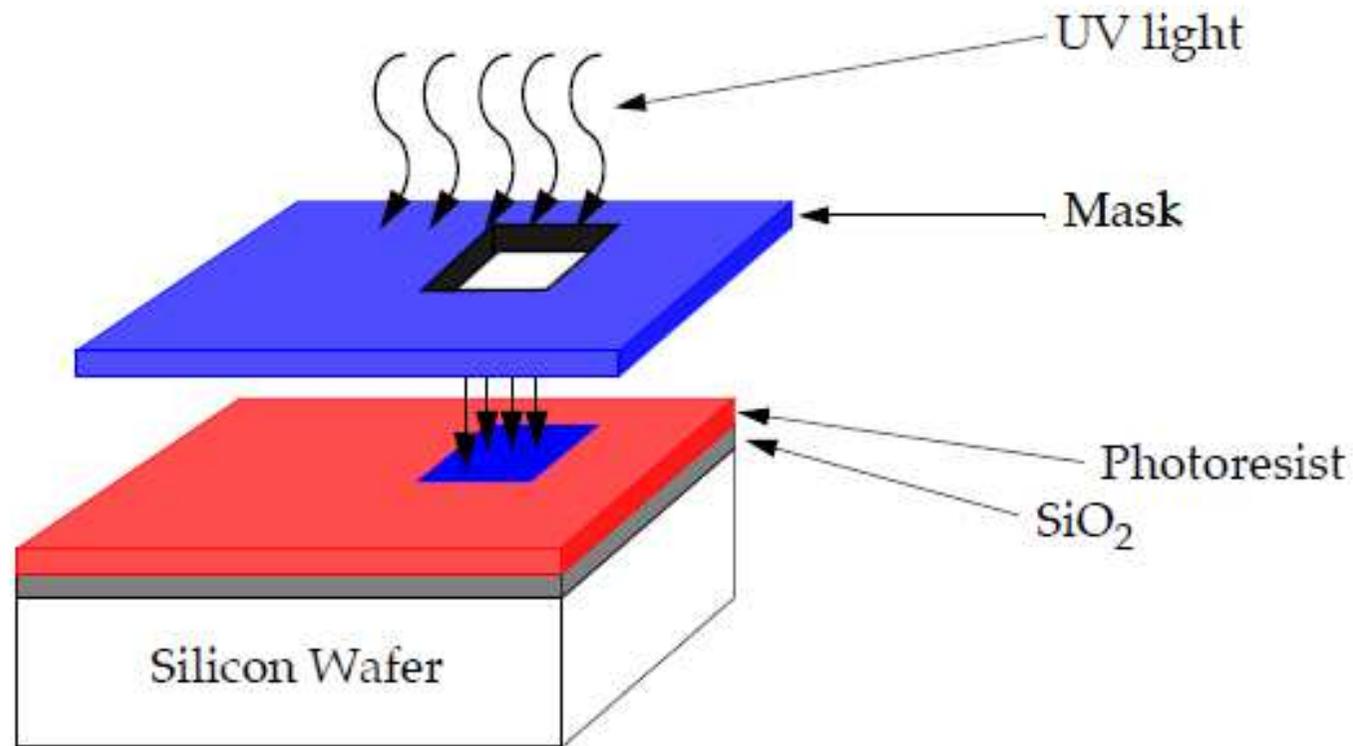
Tecnologia CMOS (brevetto Fairchild, 1963)

- La tecnologia CMOS (Complementary MOS) prevede l'utilizzazione di transistor sia di tipo N (NMOS) che di tipo P (PMOS) sullo stesso substrato, sfruttando la possibilità di poter effettuare diffusioni sia di tipo N che di tipo P per realizzare sia i contatti del transistor (source e drain) che il canale di collegamento fra di essi controllabile attraverso il segnale di gate

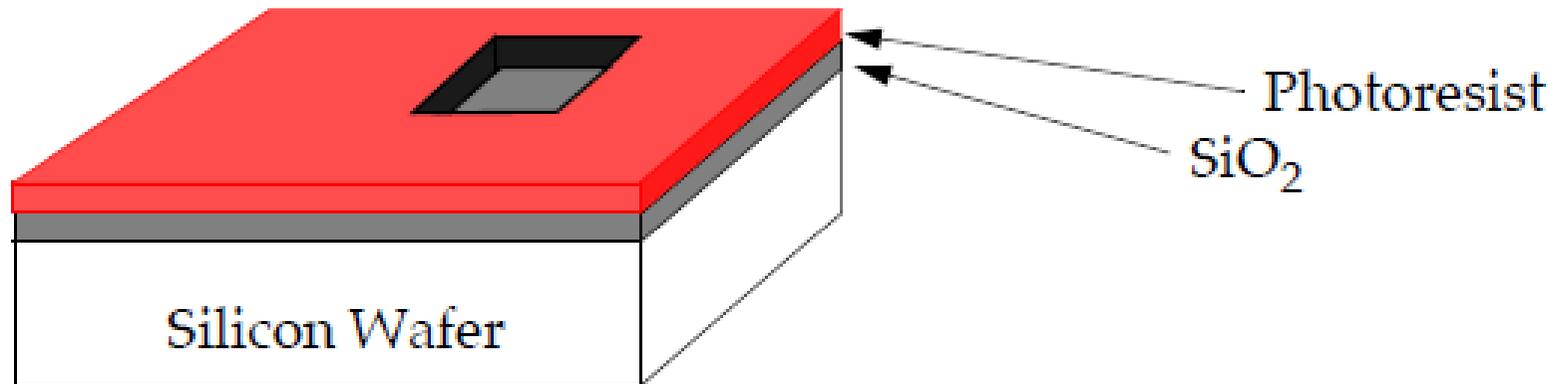


Generazione dei transistor MOS

- **Mascheratura.**
L'ossido e' gia' stato generato e ricoperto di photoresist.

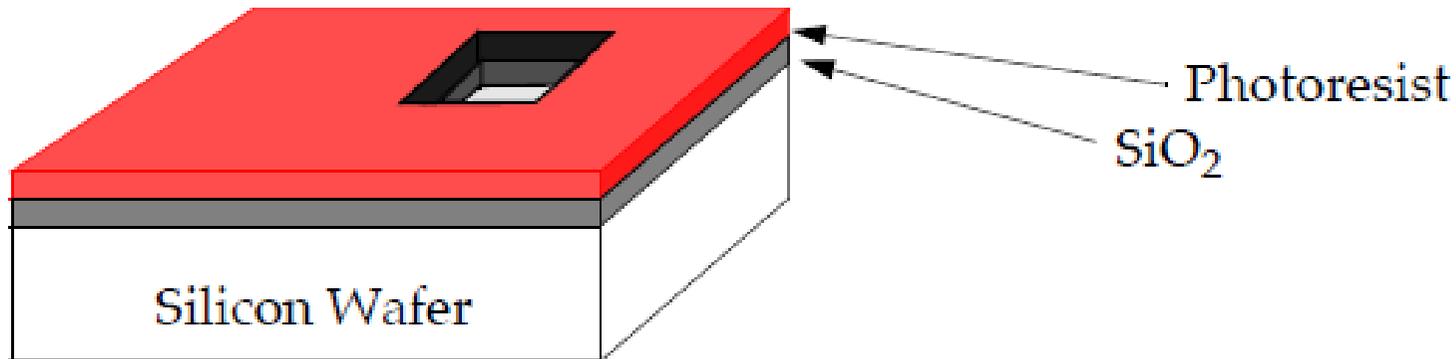


Rimozione photoresist polimerizzato dagli UV

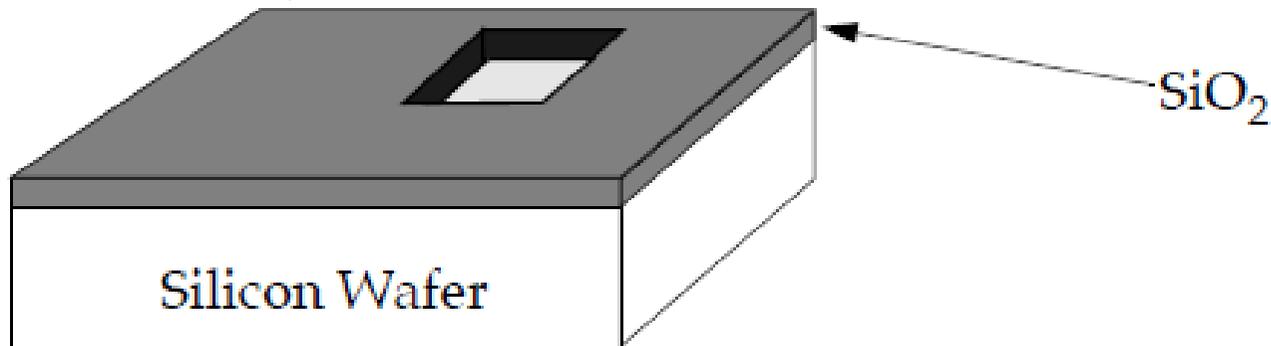


Attacco con l'acido e rimozione del photoresist

- Attacco es. con Acido Fluoridrico (HF) e creazione della finestra per la diffusione

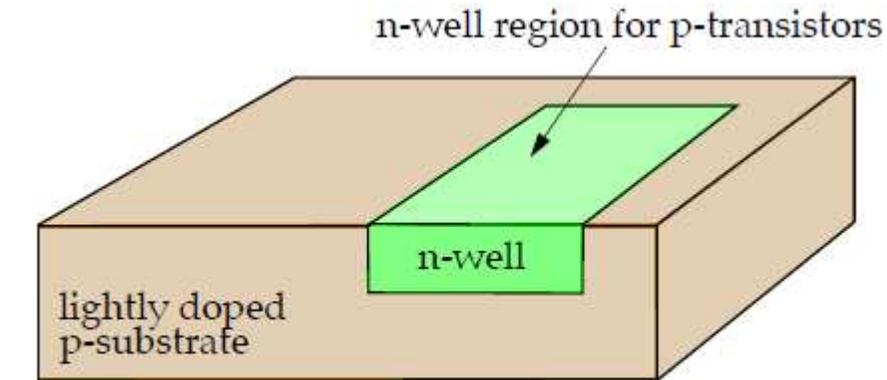


- Pulizia con acqua deionizzata e asciugatura con azoto
- Rimozione photoresist in eccesso



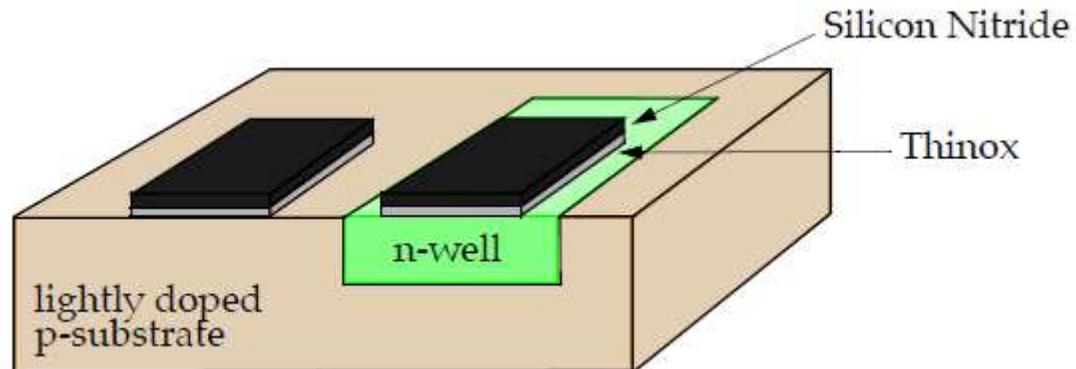
Processo CMOS semplificato

- Maschera N-WELL

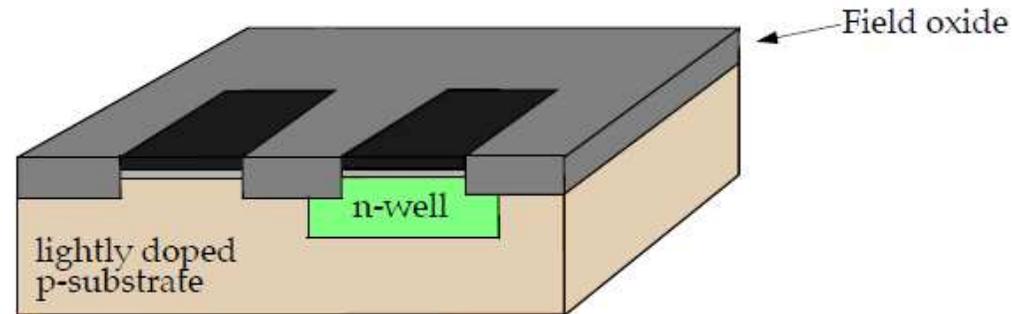


SiO₂ can be patterned as the mask for this step.

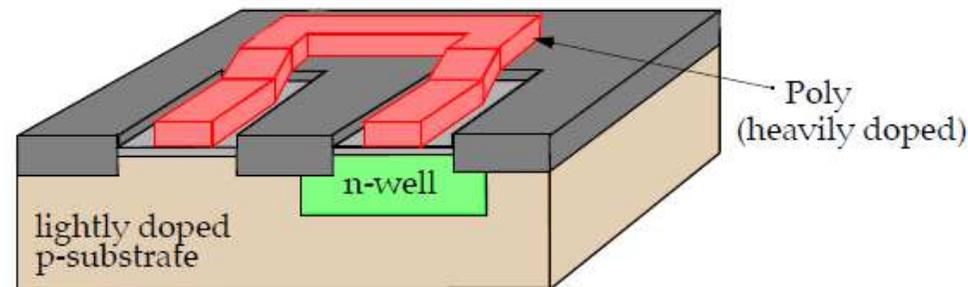
- Deposizione dell'OSSIDO di gate



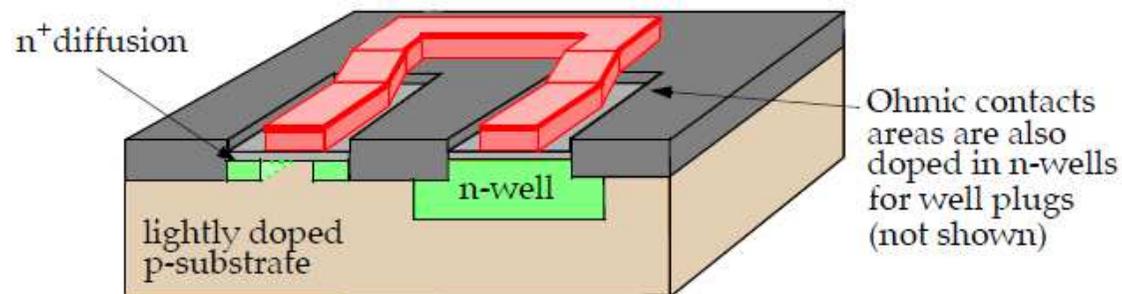
- Deposizione dell'ossido di isolamento



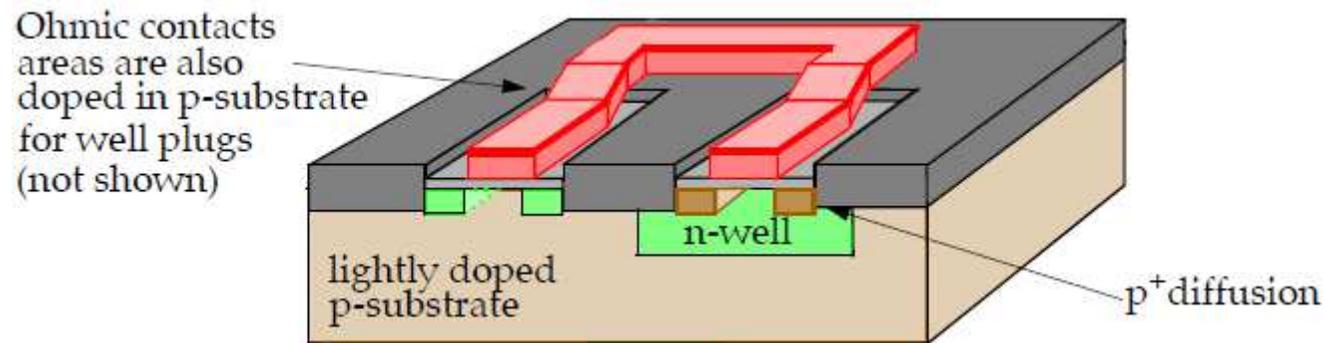
- Mascheratura per il polisilicio



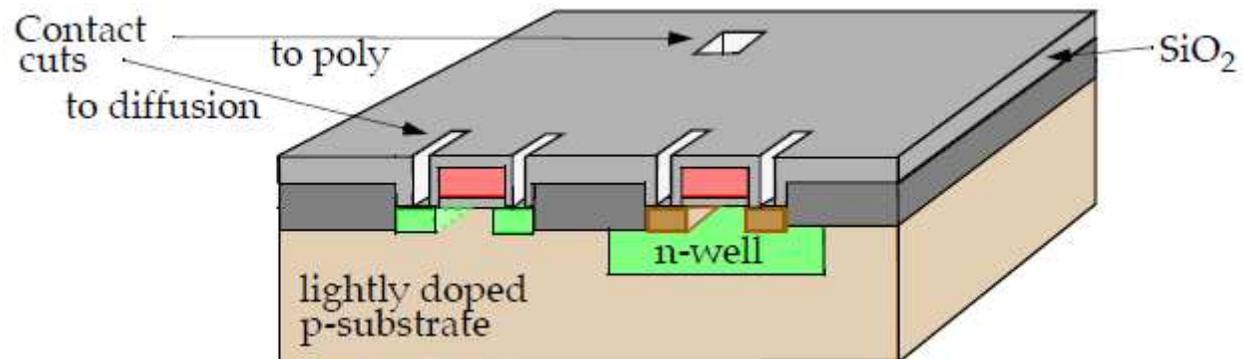
- Mascheratura per i contatti del transistor NMOS (sinistra)



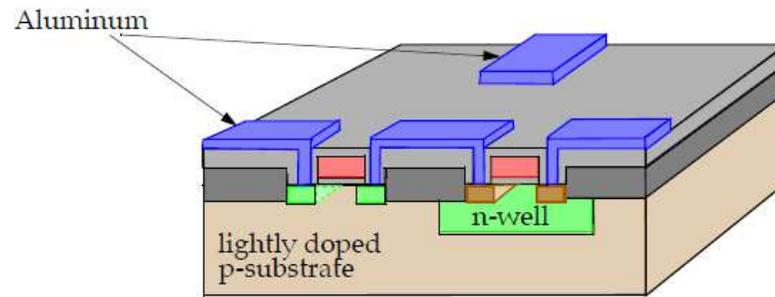
- Mascheratura per i contatti del transistor PMOS (destra)



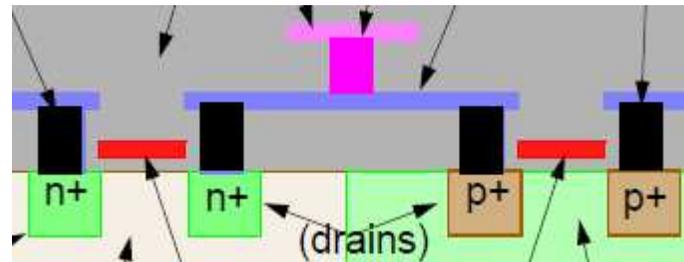
- Isolamento con ossido e fori per i contatti



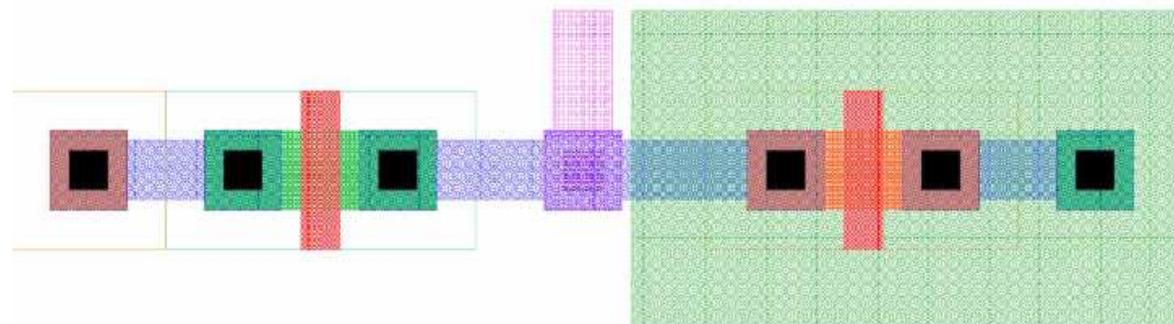
-
- Mascheratura per generare le piste di metal



- Con questo abbiamo generato la parte fondamentale dello schema iniziale

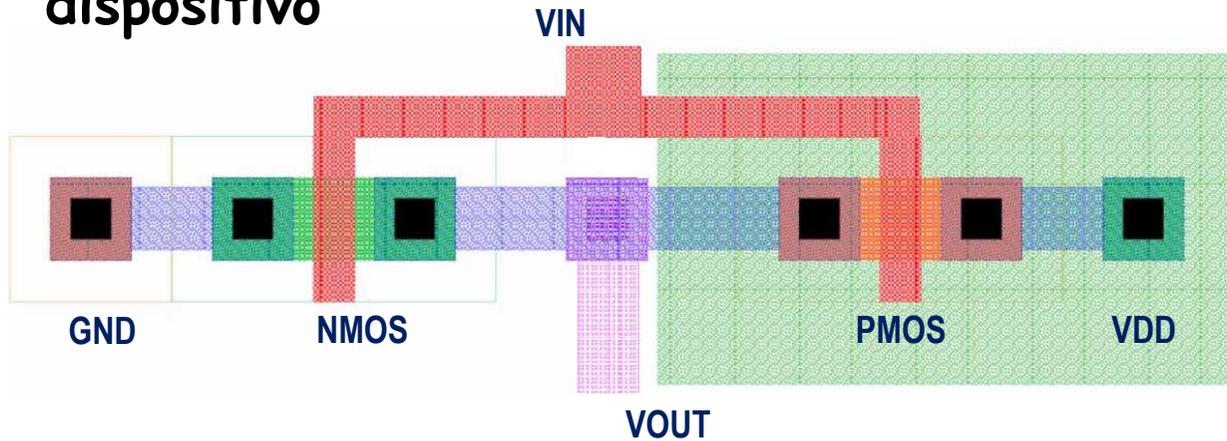


- Visto dall'alto il disegno puo' essere cosi' schematizzato (layout fisico)

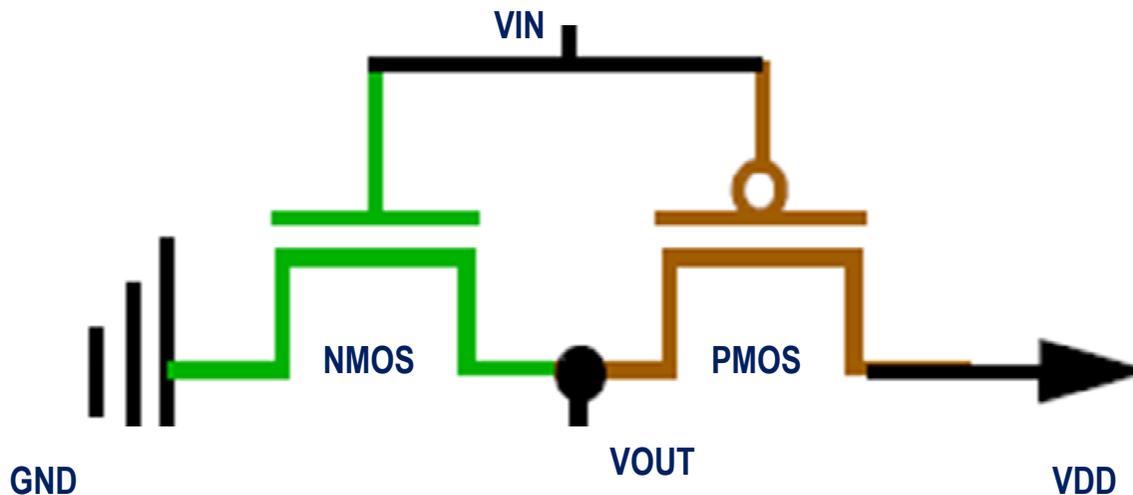


Layout fisico, elettrico e logico

- Questo "disegno" (layout fisico) fornisce quindi gli elementi essenziali per effettuare il progetto fisico del dispositivo



- Il corrispondente layout elettrico:



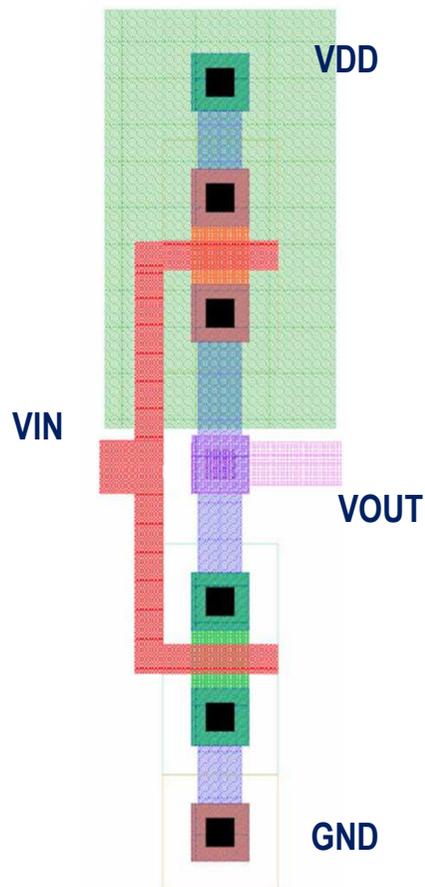
- e logico:



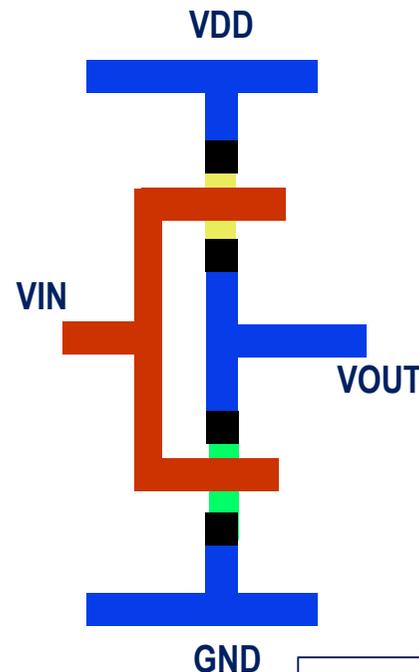
Layout fisico / Layout Simbolico / Stick diagram

- Il precedente layout fisico deve essere generato rispettando le caratteristiche fisiche della "tecnologia CMOS" specifica utilizzata (es. 14nm) → "layout rules"

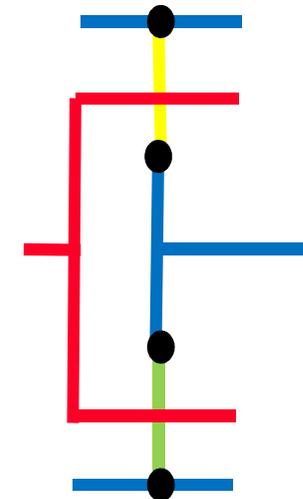
Layout fisico con design rules



Layout simbolico con design rules



Stick diagram



Le regole di layout impongono ad es. di rispettare certe distanze minime fra le piste, i contatti, etc

Nota: questa e' solo una delle possibili implementazioni fisiche di un inverter !

Vantaggi dell'inverter CMOS

- “Fully restored”: i livelli alto e basso (VDD e GND) sono completamente ristabiliti dopo il passaggio attraverso la porta
- “Ratioless”: I livelli logici non sono dipendenti dalle dimensioni relativi dei dispositivi
- Bassa impedenza di uscita: a regime l'uscita e' connessa o a VDD o a GND, aumentando la robustezza al rumore
- Alta impedenza di ingresso: il fanout (quanti dispositivi posso collegare sull'uscita) e' teoricamente illimitato in condizioni statiche, anche se il numero di dispositivi in uscita influenza il transitorio
- Bassa dissipazione di potenza statica: non c'e' mai un cammino diretto fra alimentazione e terra

Note importanti

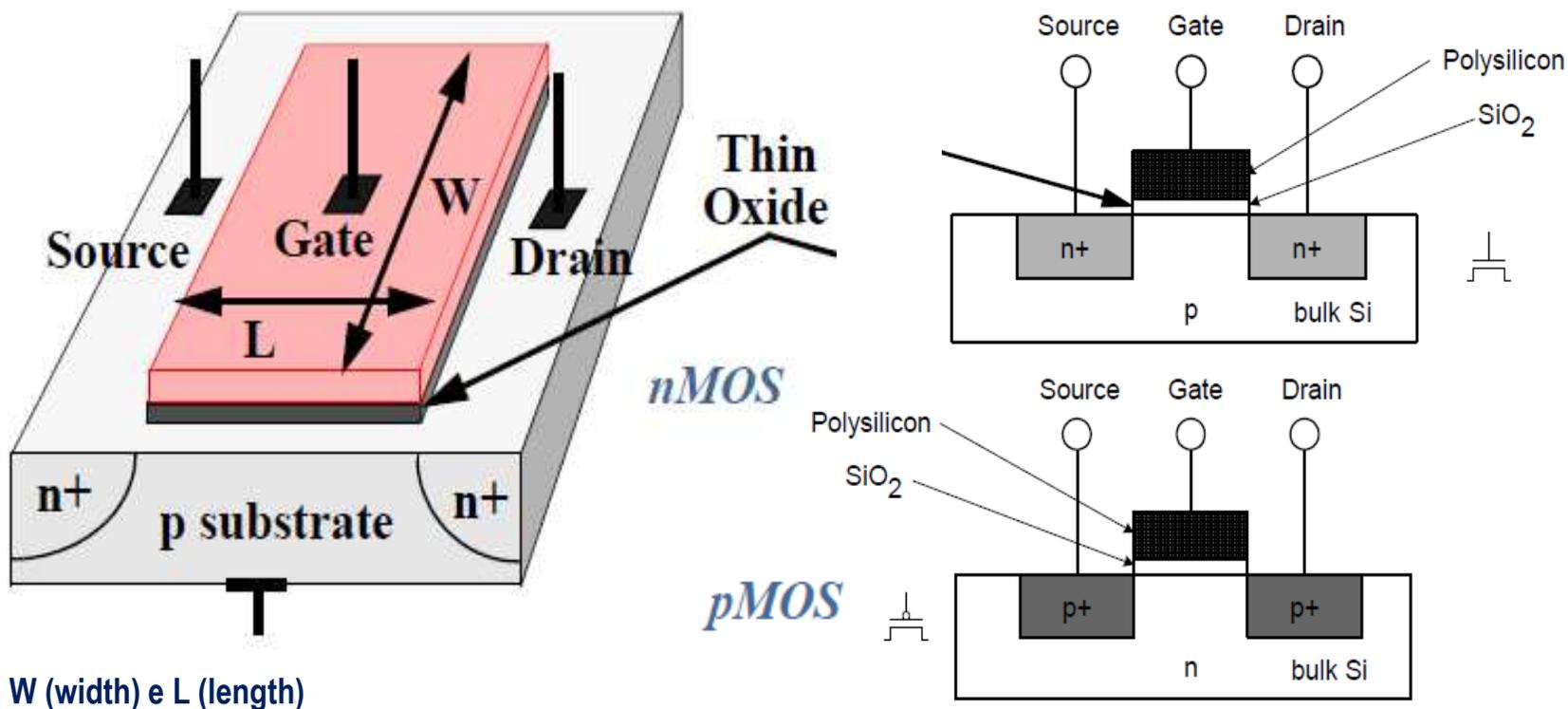
- La progettazione fisica descritta e' stata negli anni estremamente perfezionata e si e' recentemente (2011) passati alla produzione commerciale di dispositivi (es. FinFET) che si sviluppano non solo planarmente (2D) ma anche verticalmente (3D)
- La progettazione fisica permette di sfruttare al massimo ogni grado di liberta' (progettazione "full-custom") ma nel caso di chip contenenti vari miliardi di dispositivi non e' umanamente gestibile se non con l'aiuto di strumenti CAD (Computer Aided Design)
 - Tipicamente, si ricorre per la grande maggioranza dei casi a librerie gia' predisposte ("standard cells") e si usa il design full-custom solo in casi di necessita'
 - Il design fisico puo' quindi essere gestito con appositi strumenti in cui il progettista disegna "blocchi colorati" e automaticamente vengono verificate e imposte le regole di layout (es. L-Edit, Cadence tools, Synopsys tools, ...)
- Il passo puo' essere automatizzato (similmente a un compilatore per il software) generando il layout fisico a partire dal circuito elettrico
 - E' importante anche il passo inverso: estrarre il circuito elettrico (compreso gli effetti "parassiti" quali capacita', cross-talking) a partire dal layout fisico

Osservazioni importanti

- La fabbricazione dei chip e' appannaggio di pochi in quanto il costo di un "silicon foundry" supera oggi i 10 miliardi di euro (es. Intel, Samsung, Global Foundries)
- Viceversa la *progettazione* puo' essere effettuata anche con pochissime risorse
 - Grandi ditte come ARM, AMD, Xilinx hanno un business model "fabless" ovvero dopo aver generato il layout fisico lo passano a una silicon foundry che crea il chip per loro o addirittura vendono il design (es. ARM)

Caratteristiche elettriche del transistor MOS

- Abbiamo visto come si puo' realizzare un transistor MOS



- Dimensionando opportunamente W/L si possono variare i tempi di risposta (o di commutazione)

$$I_D = k_n' \frac{W}{L} \left[(V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

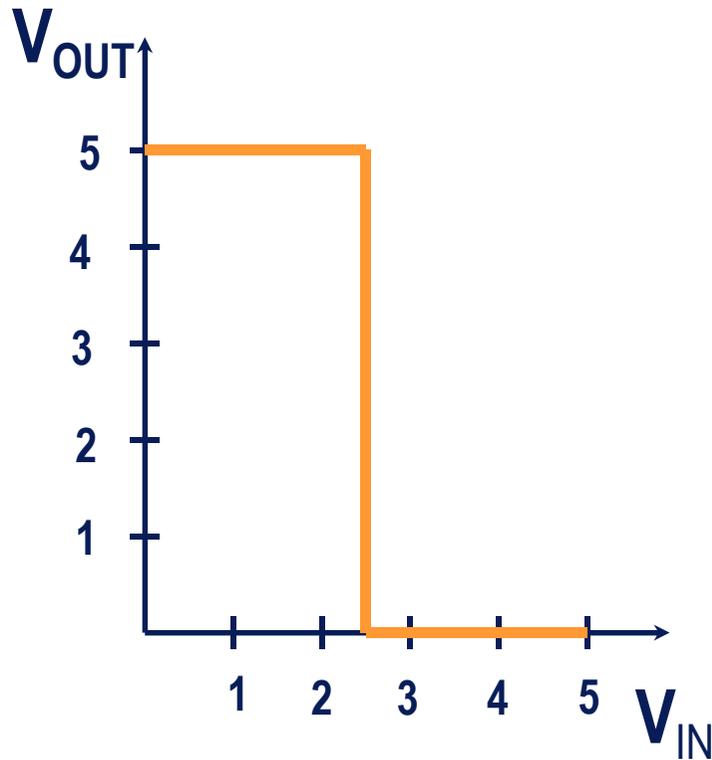
$$k_n' = \mu_n C_{ox} = \frac{\mu_n \epsilon_{ox}}{t_{ox}}$$

V_T Tensione di soglia

CARATTERISTICA DI TRASFERIMENTO dell'INVERTER

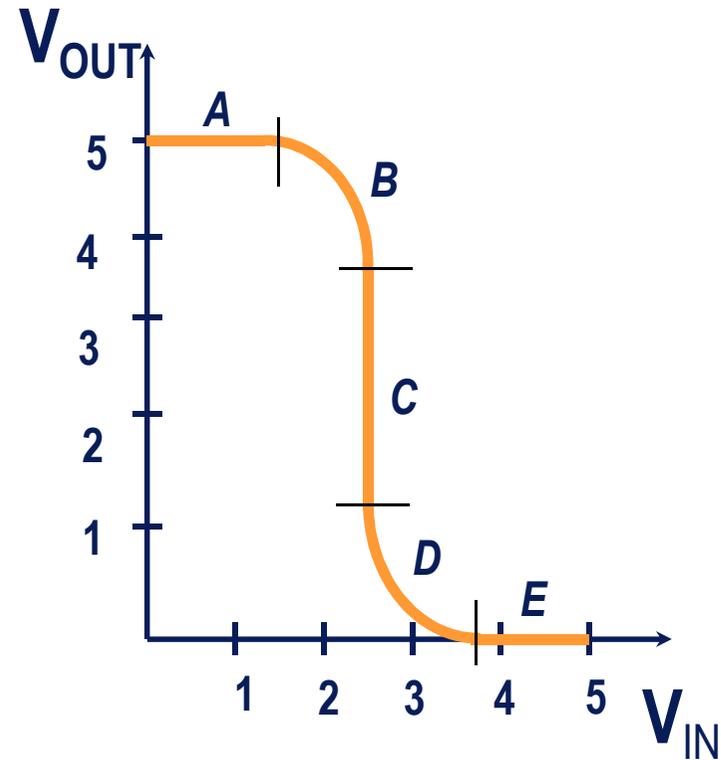
- Caratteristica

IDEALE



Caratteristica

REALE



Tratto di curva	Mn	Mp
A	interdizione	zona triodo
B	saturazione	zona triodo
C	saturazione	saturazione
D	zona triodo	saturazione
E	zona triodo	Interdizione

Osservazioni

- La tecnologia CMOS consente di realizzare la funzione logica piu' semplice (inverter o porta NOT) in modo tale che la sua caratteristica di trasferimento sia vicina alla caratteristica ideale
 - Altre tecnologie (es. NMOS, BJT) hanno alcune limitazioni (NMOS: livelli logici non pieni, BJT: alto consumo) ma possono offrire alcuni vantaggi (NMOS: maggiore compattezza, BJT: maggiore velocita')
 - Attualmente la tecnologia CMOS e' la tecnologia di gran lunga piu' utilizzata per realizzare sistemi digitali soprattutto per la sua efficienza energetica
- Per realizzare funzioni logiche piu' complesse ci viene in aiuto l'algebra booleana (v. lezioni successive)
 - In teoria oltre all'inverter ci servirebbero le funzioni elementari AND e OR
 - In pratica, le porte piu' semplici da realizzare a partire dall'inverter sono le porte NAND e NOR (v. slide successive)
 - Fortunatamente si puo' facilmente dimostrare che e' possibile costruire qualsiasi funzione logica a partire anche dalle sole porte NAND (o NOR)
 - E' immediato altresì vedere che NAND seguito da NOT equivale a AND e NOR seguito da NOR quivale a OR.

Porta NAND CMOS

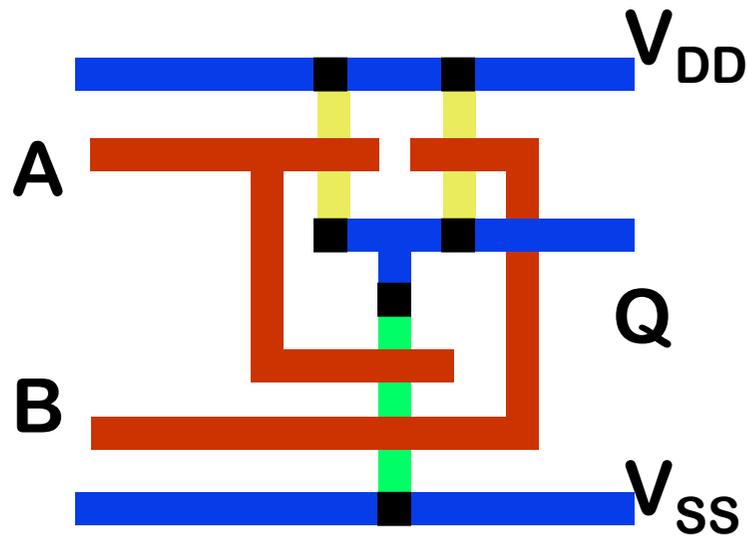
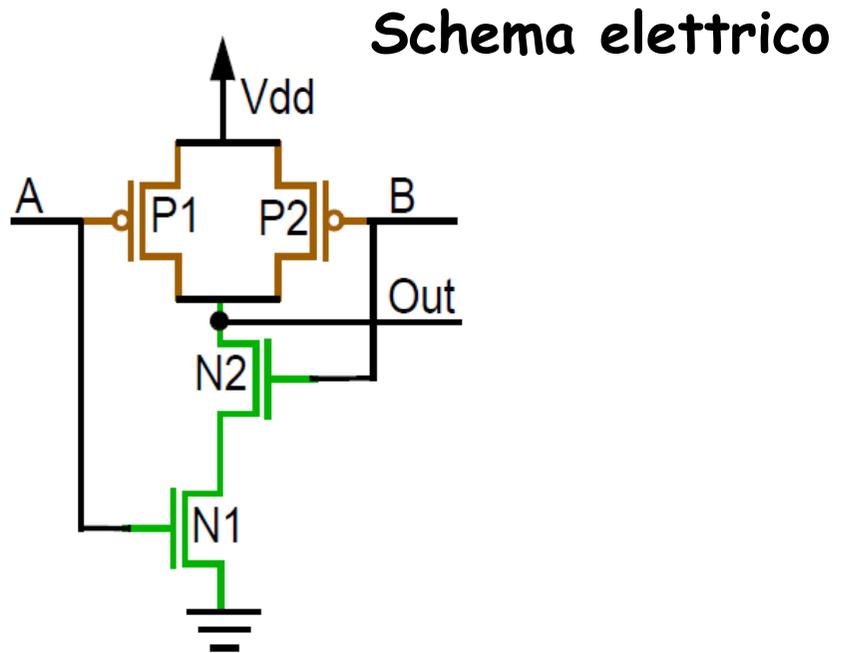
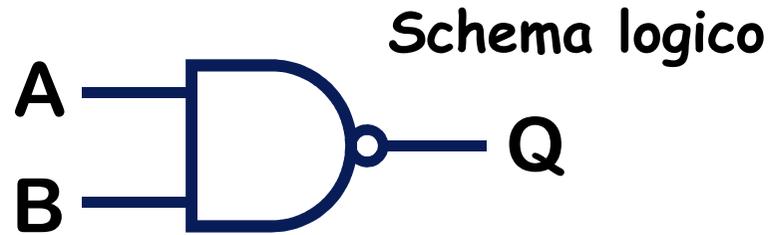
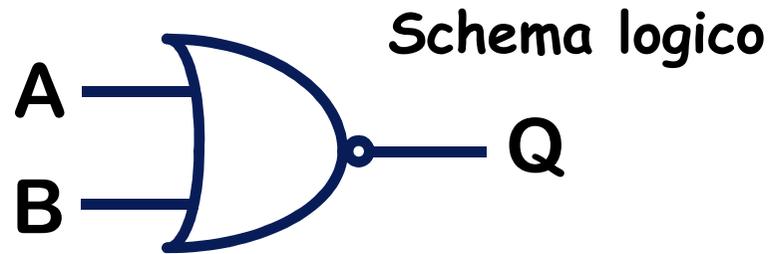
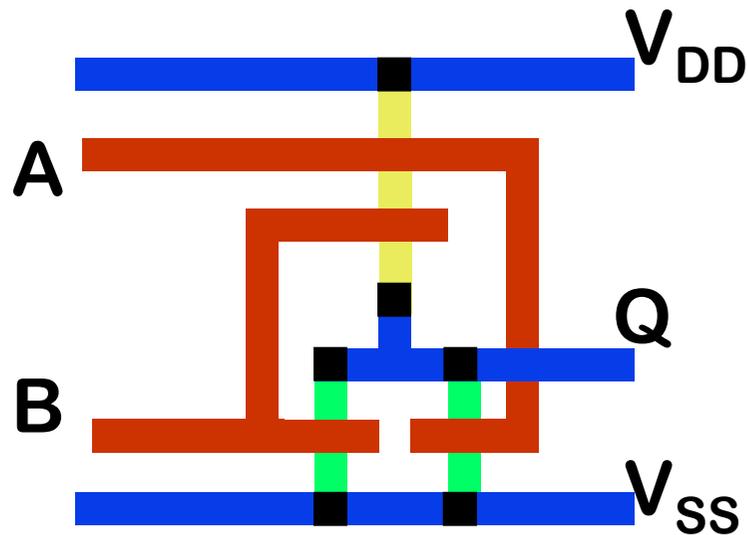


Tabella di Verità

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



Porta NOR CMOS



Schema elettrico

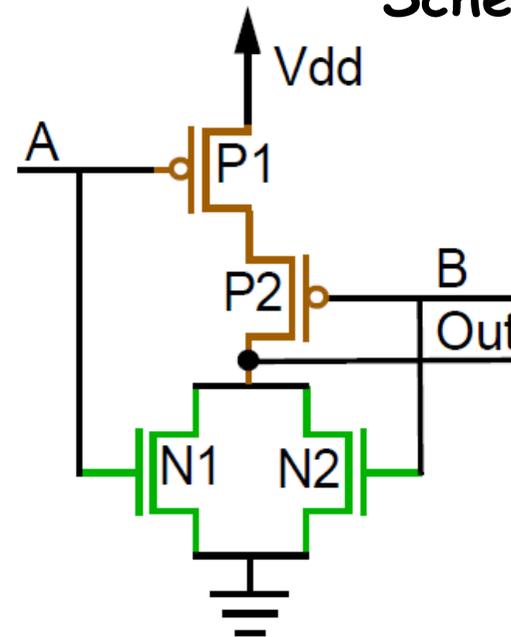


Tabella di Verità

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Porta di transito CMOS (transmission gate)

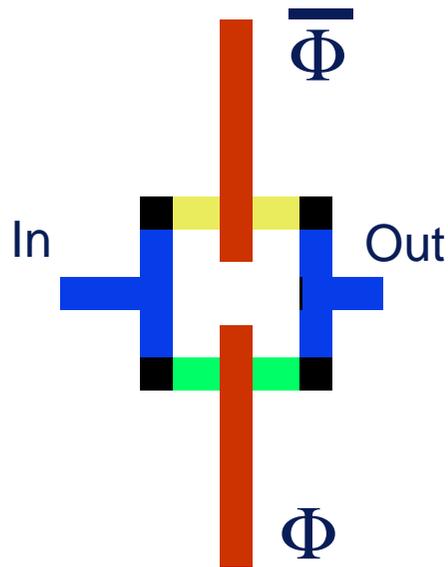
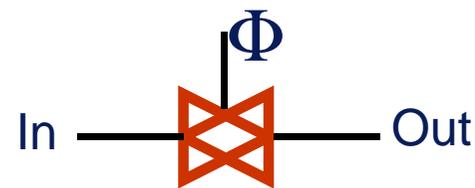


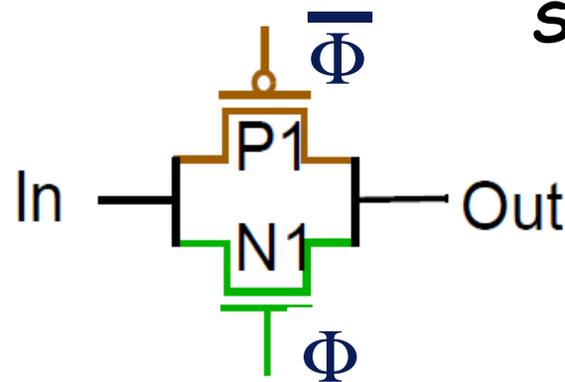
Tabella di Verità

Φ	In	Out
0	X	Z
1	0	0
1	1	1



Simbolo

Nota: $\overline{\Phi}$ e' sottinteso



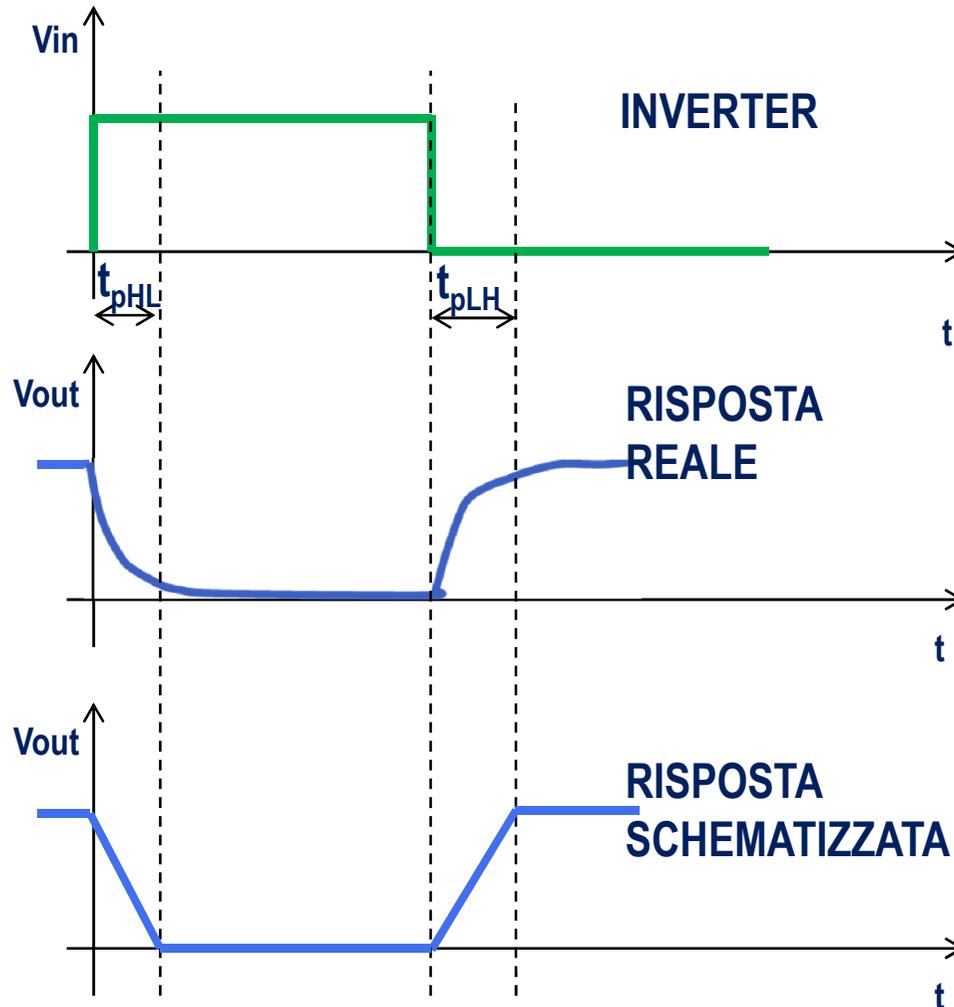
Schema elettrico

Nota: non e' un "tri-state" buffer in quanto quest'ultimo rigenera anche il segnale, mentre in questo caso il segnale si degrada un poco nel passaggio.

Nota2: viceversa si realizza un buffer mettendo due inverter in cascata

Ritardo di propagazione dell'inverter CMOS

- Le porte fondamentali e di trasmissione presentano un certo ritardo di propagazione t_p



Ad esempio:

applicando un segnale a gradino all'ingresso dell'inverter la forma d'onda di uscita raggiunge un livello accettabile (es. 90%) con un ritardo t_p

Il ritardo necessario per la transizione dal livello alto a livello basso t_{pHL} può essere diverso da quello della transizione opposta t_{pLH}

$$t_{pHL} = \ln(2)R_{eqn}C_L = 0.69R_{eqn}C_L$$

$$t_{pLH} = \ln(2)R_{eqp}C_L = 0.69R_{eqp}C_L$$

$$R_{eq} \approx \frac{3}{4} \frac{V_{DD}}{I_{DSAT}} \left(1 - \frac{7}{9} \lambda V_{DD} \right) \quad \lambda: \text{empirical parameter called channel-length modul} \quad \lambda \propto \frac{1}{L}$$

$$I_{DSAT} = k \frac{W}{L} \left((V_{DD} - V_T)V_{DSAT} - \frac{V_{DSAT}^2}{2} \right)$$

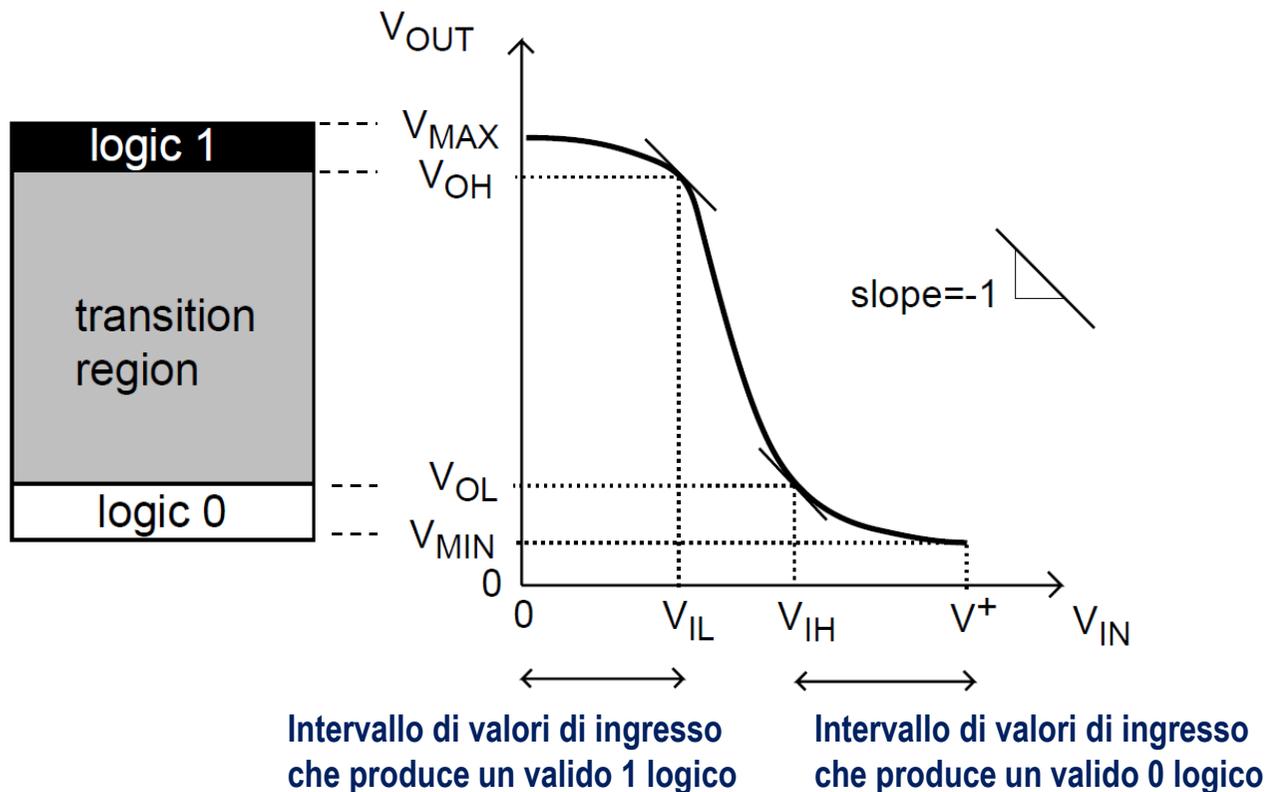
$$V_{DSAT} = \kappa(V_{GS} - V_T)(V_{GS} - V_T)$$

$$V_{DD} \uparrow \Rightarrow t_p \downarrow$$

$$L \downarrow \Rightarrow t_p \downarrow$$

Approfondimenti ulteriori possono essere fatti nei corsi di Elettronica. In questo contesto schematizzeremo il ritardo con una transizione lineare con durata finita

Definizione dei livelli logici 1 e 0



- **0 logico:**

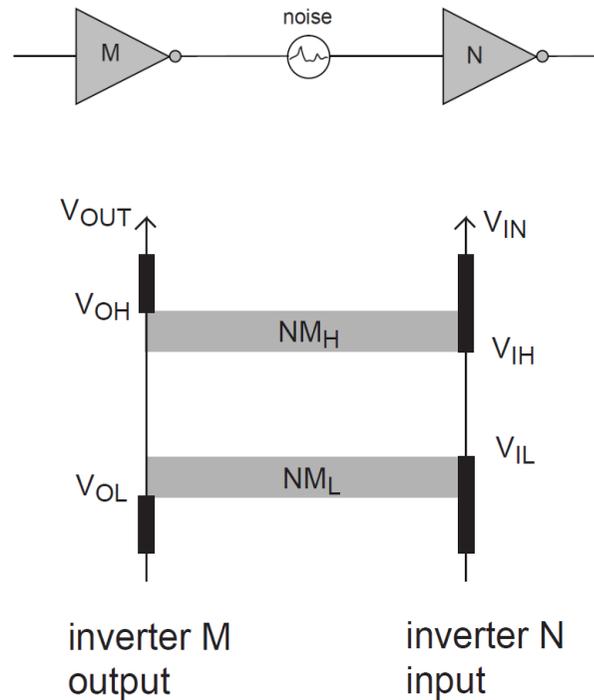
- V_{min} = tensione di uscita per la quale $V_{IN}=V^+$
- V_{OL} = la piu' piccola tensione di uscita per la quale la pendenza $e' = -1$

- **1 logico:**

- V_{OH} = la piu' grande tensione di uscita per la quale la pendenza $e' = -1$
- V_{MAX} = tensione di uscita per la quale $V_{IN}=0$

Margine di rumore

- Se $(V_{OH}-V_{OL}) > (V_{IH}-V_{IL})$ l'inverter ha una certa immunita' al rumore



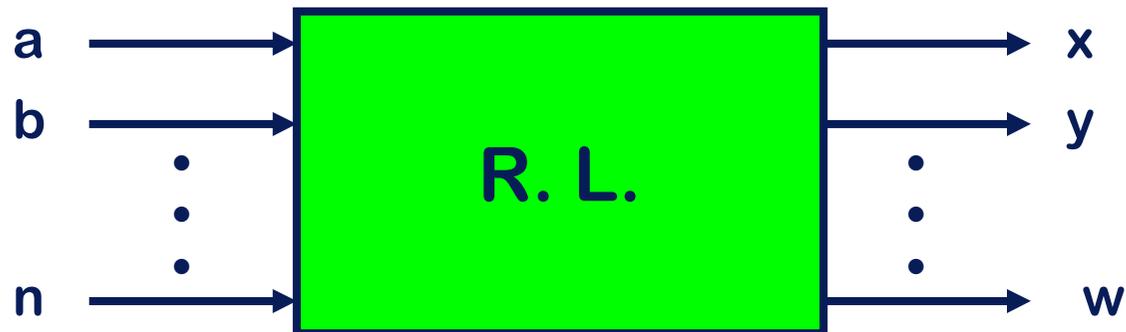
- $NM_H = (V_{OH}-V_{IH})$
- $NM_L = (V_{IL}-V_{OL})$

margine di rumore sul livello alto
margine di rumore sul livello basso

Dalle porte elementari
alle reti logiche

Reti Logiche

- Sistema elettronico che ha in ingresso segnali digitali e fornisce in uscita segnali digitali secondo leggi descrivibili con l'algebra Booleana



- R.L. è unidirezionale

Tipi di reti

- Reti **COMBINATORIE**

- In qualunque istante le uscite sono funzione del valore che gli ingressi hanno in quell'istante
- Il comportamento (uscite in funzione degli ingressi) è descritto da una tabella

- Reti **SEQUENZIALI**

- In un determinato istante le uscite sono funzione del valore che gli ingressi hanno in quell'istante e i valori che hanno assunto precedentemente
- La descrizione è più complessa
- Stati Interni
- Reti dotate di **MEMORIA**

Simboli

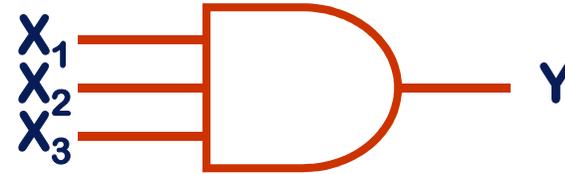
- Rete Logica => scomponibile in blocchi
- Blocchi base = simboli degli operatori elementari
- Rappresentazione delle funzioni logiche mediante schemi
- RAPPRESENTAZIONE SCHEMATICA

Porte Logiche Base

- Rappresentazione circuitale delle funzioni logiche

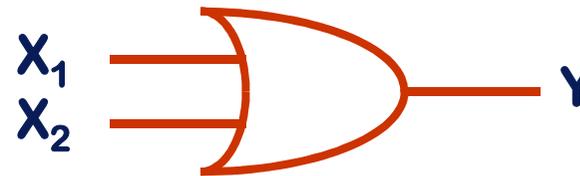
- AND

$$Y = X_1 \bullet X_2 \bullet X_3$$



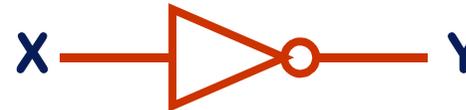
- OR

$$Y = X_1 + X_2$$



- NOT

$$Y = \overline{X}$$

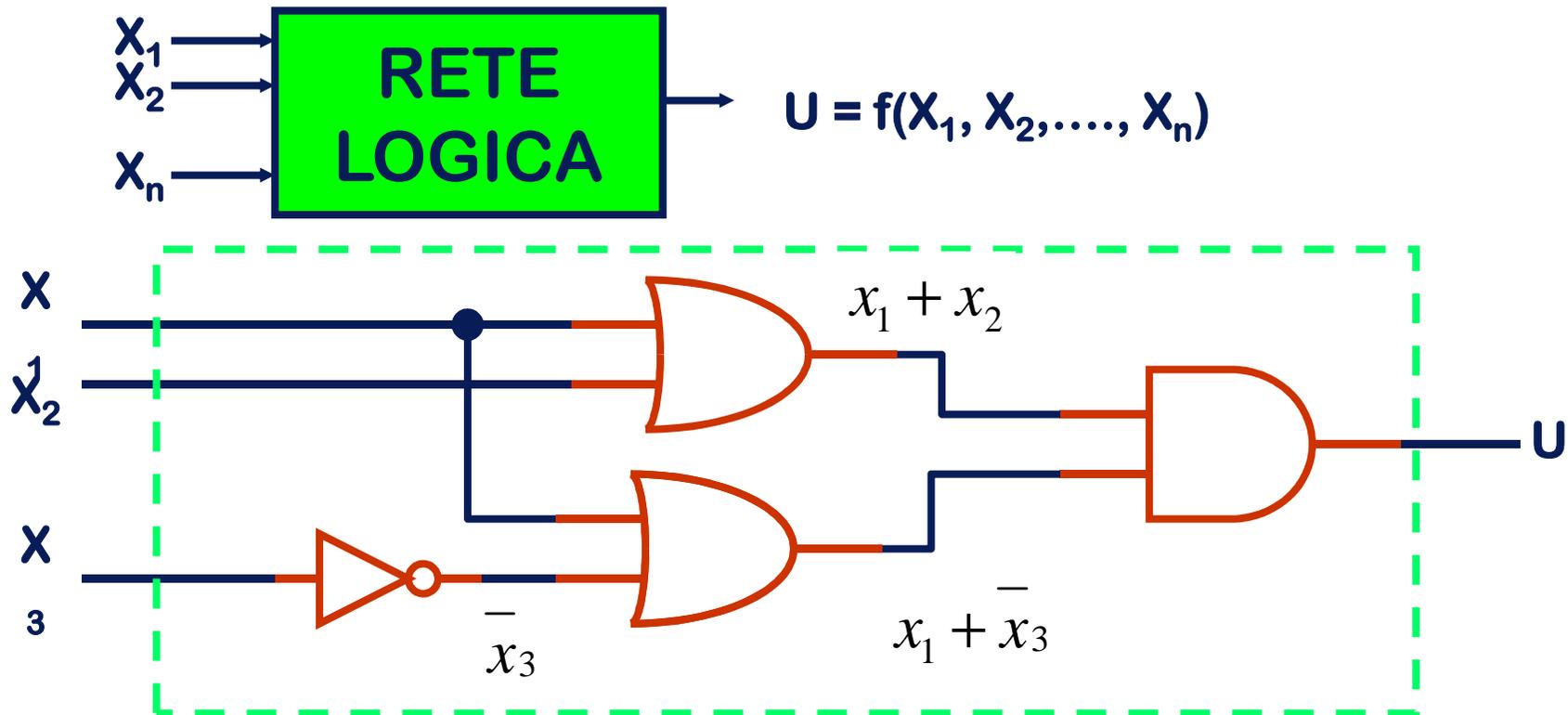


Esempio

- Schema simbolico della funzione

$$U = f(X_1, X_2, \dots, X_n) = (X_1 + X_2) \bullet (X_1 + \overline{X_3})$$

- RETE LOGICA

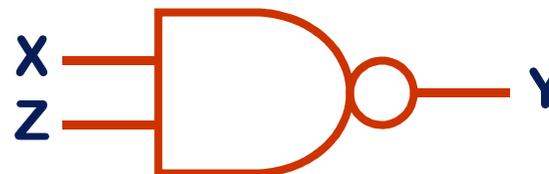


Altre porte logiche

- NAND

X	Z	Y
0	0	1
0	1	1
1	0	1
1	1	0

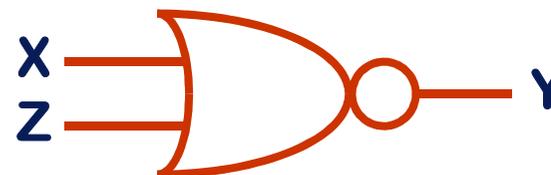
$$Y = \overline{X \cdot Z}$$



- NOR

X	Z	Y
0	0	1
0	1	0
1	0	0
1	1	0

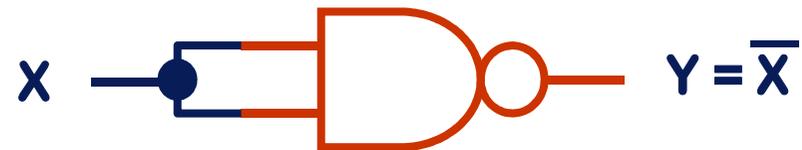
$$Y = \overline{X + Z}$$



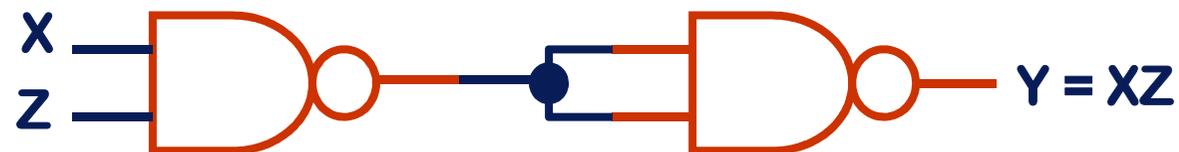
Proprietà della porta NAND (NOR)

- Utilizzando solamente porte NAND (NOR) è possibile realizzare qualunque rete logica

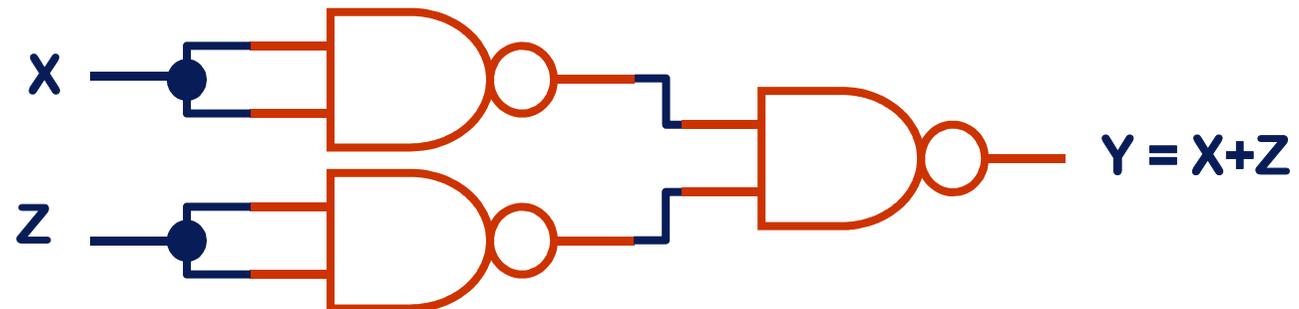
- NOT



- AND



- OR

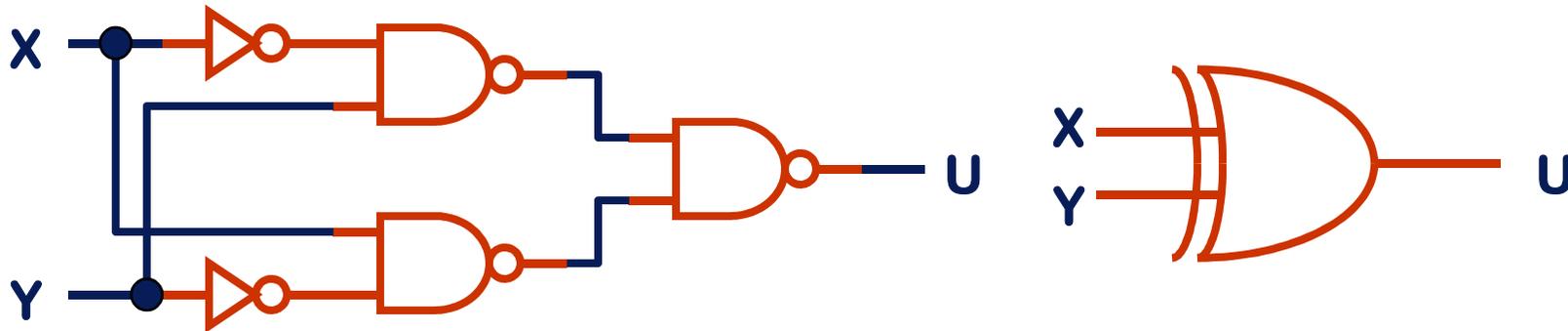


OR Esclusivo

- Realizzazione dell'OR Esclusivo

X	Y	U
0	0	0
0	1	1
1	0	1
1	1	0

$$U = X \oplus Y = \overline{X}Y + X\overline{Y}$$



Algebra Booleana

Algebra della Logica

- **George Boole**
 - **Matematico inglese** (1815 - 1864)
 - **Algebra della Logica, Algebra di Boole, Algebra Booleana**
 - **Sistema matematico formale che descrive funzioni logiche**
 - **Funzioni che possono assumere al minimo (solo) due valori**
 - **Vero** **Falso**
 - **Le variabili di funzioni logiche possono assumere solo due valori**
 - **Sistema matematico formale**
 - **Insieme di elementi**
 - **insieme di operazioni**
 - **insieme di postulati**
- » **TEOREMI**

Definizione di "AND"

- **Operazione**

- AND o **PRODOTTO LOGICO**

$$x \bullet y \quad xy$$

- **Postulato**

- l'operazione AND è definita dalla tabella

x		y		$x \bullet y$
0	•	0	=	0
0	•	1	=	0
1	•	0	=	0
1	•	1	=	1

Osservazioni

1. $x \cdot y$ è uguale a "1" se e solo se x e y sono uguali a "1", altrimenti $x \cdot y$ è uguale a "0"
 2. Si può estendere a "n" variabili:
 $x_1 \cdot x_2 \cdot \dots \cdot x_n$ è uguale "1" se e solo se x_1, x_2, \dots, x_n sono uguali a "1"
- La funzione AND corrisponde al concetto:
un evento si verifica se e solo se tutte le condizioni sono verificate

Definizione di "OR"

- Operazione
 - OR o SOMMA LOGICA

$$x + y$$

- Postulato
 - l'operazione OR è definita dalla tabella

x		y		$x + y$
0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	1

Osservazioni

1. $x + y$ è uguale a "0" se e solo se x e y sono uguali a "0", altrimenti $x + y$ è uguale a "1"
 2. Si può estendere a "n" variabili:
 $x_1 + x_2 + \dots + x_n$ è uguale "0" se e solo se x_1, x_2, \dots, x_n sono uguali a "0"
- La funzione OR corrisponde al concetto:
perché un evento si verifica è sufficiente che una sola condizione sia verificata

Definizione di "NOT"

- **Operazione**

- NOT o Complemento Logico , o Negazione, o Inversione

$$\overline{x}$$

- **Postulato**

- l'operazione NOT è definita dalla tabella

x	\overline{x}
0	1
1	0

Osservazioni

1. se x è uguale a "0" allora x negato è uguale a "1", se x è uguale a "1" allora x negato è uguale a "0"

2. Ovvero $\bar{X} = 1$ se $X = 0$

$\bar{X} = 0$ se $X = 1$

$\bar{0} = 1$ e $\bar{1} = 0$

- La funzione NOT corrisponde al concetto:
negazione della condizione

Funzione logica (o Boleana)

- Una funzione

$$u = f(x_1, \dots, x_n)$$

è una legge che fa corrispondere un valore logico (0 o 1) di u ad ogni combinazione di valori x_1, \dots, x_n .

- La funzione f è costituita da variabili logiche, costanti e le tre operazioni logiche fondamentali

$$u = \overline{(x_1 + x_2)} \bullet (x_3 + x_1 \bullet \overline{x_2}) + \overline{x_3}$$

Osservazioni

- Nelle funzioni logiche le parentesi indicano una gerarchia di esecuzione uguale a quella comunemente usata nelle espressioni aritmetiche note
- Fra le operazioni logiche AND, OR e NOT esiste la gerarchia: 1) NOT, 2) AND, 3) OR
- La gerarchia prima descritta consente di ridurre l'uso di parentesi nelle funzioni logiche

Tabella di Verità (1)

- Una funzione logica può sempre essere espressa da una tabella che prende il nome di:

TABELLA DI VERITÀ (TRUTH TABLE)

- Osservazione
- Una funzione di "n" variabili ammette 2^n possibili configurazioni
- Una funzione di "n" variabili è completamente descritta da una tabella che ha sulla sinistra le 2^n possibili configurazioni degli ingressi e a destra i valori (0 o 1) a secondo del valore della funzione

Tabella di verità (2)

- Funzione di tre variabili

$$u = f(x, y, z)$$

<i>x</i>	<i>y</i>	<i>z</i>	<i>u</i>
0	0	0	<i>f</i> (0,0,0)
0	0	1	<i>f</i> (0,0,1)
0	1	0	<i>f</i> (0,1,0)
0	1	1	<i>f</i> (0,1,1)
1	0	0	<i>f</i> (1,0,0)
1	0	1	<i>f</i> (1,0,1)
1	1	0	<i>f</i> (1,1,0)
1	1	1	<i>f</i> (1,1,1)

Esempio

$$u = f(x, y, z) = (x + \bar{y}) \cdot (\bar{x} + z) + yz$$

$$f(0,1,1) = (0 + \bar{1}) \cdot (\bar{0} + 1) + 1 \cdot 1 = (0 + 0) \cdot (1 + 1) + 1 = 0 \cdot 1 + 1 = 0 + 1 = 1$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Passo 1

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Passo 2

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Passo 3

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Passo 4

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Passo 5

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Passo 6

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Fine

$$u = f(x, y, z) = (x + \bar{y})(\bar{x} + z) + yz$$

x	y	z	\bar{x}	\bar{y}	$x + \bar{y}$	$\bar{x} + z$	$(x + \bar{y})(\bar{x} + z)$	yz	u
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Richiami

- Insieme di elementi
- Variabili, costanti
- Insieme di operazioni
- Insieme di postulati
- Espressioni algebriche
- Tabella di verità

Postulati di HUNTINGTON

Algebra Booleana

Almeno due elementi distinti

1a Somma logica (+)

1b Prodotto logico (\bullet)

Elementi Identità

2a $x + 0 = x$

2b $x \bullet 1 = x$

Proprietà Commutativa

3a $x + y = y + x$

3b $x \bullet y = y \bullet x$

Proprietà Distributiva

4a $x \bullet (y + z) = (x \bullet y) + (x \bullet z)$

4b $x + (y \bullet z) = (x + y) \bullet (x + z)$

Complementazione

5a $x + \bar{x} = 1$

5b $x \bullet \bar{x} = 0$

Osservazioni

- Alcune proprietà dell'algebra booleana sono vere anche nell'algebra normalmente usata:
 - Proprietà commutativa
 - Proprietà distributiva del prodotto logico
- Altre proprietà non sono vere :
 - Proprietà distributiva della somma logica
- L'operazione complemento logico esiste solo nell'algebra booleana
- La sottrazione e la divisione non esistono nell'algebra booleana

Principio di DUALITÀ

- Da un'osservazione dei postulati precedenti si osserva che quelli "b" si ottengono da "a"
 - Scambiando i due operatori binari fra loro, (+) con (·) e (·) con (+)
 - Scambiando fra loro i due elementi identità, 1 con 0 e 0 con 1

TEOREMI FONDAMENTALI

- **Tecniche di dimostrazione dei teoremi**
 - **Impiego dei postulati fondamentali**
 - **Uso di teoremi precedentemente dimostrati**
 - **Dimostrazione per assurdo**
 - (si ipotizza verificata l'ipotesi opposta a quella desiderata e si conclude che non è possibile che sia vera)
 - **Dimostrazione per induzione**
 - (se una ipotesi è vera per k variabili e per $k+1$ variabili allora è vera per qualunque n)

Osservazione

- La tabella di verità consente di provare la veridicità di una relazione logica, poiché verifica se la relazione è vera per TUTTE le possibili combinazioni dei valori delle variabili
- Tale metodo prende il nome di
- *Metodo dell'INDUZIONE PERFETTA*

TEOREMI

$$1a \quad x + 1 = 1$$

$$1b \quad x \bullet 0 = 0$$

$$2a \quad \overline{\overline{x}} = x$$

$$3a \quad x + x = x$$

$$3b \quad x \bullet x = x$$

$$4a \quad x + xy = x$$

$$4b \quad x(x + y) = x$$

$$5a \quad x + \overline{xy} = x + y$$

$$5b \quad x(\overline{x + y}) = xy$$

$$6a \quad x + (y + z) = (x + y) + z = x + y + z$$

$$6b \quad x(yz) = (xy)z = xyz$$

$$7a \quad xy + \overline{xz} + yz = xy + \overline{xz}$$

$$7b \quad (x + y)(\overline{x + z})(y + z) = (x + y)(\overline{x + z})$$

$$8a \quad \overline{(x + y)} = \overline{x} \cdot \overline{y}$$

$$8b \quad \overline{(x \cdot y)} = \overline{x} + \overline{y}$$

Esempio di dimostrazione

- Teorema di De Morgan (8a e 8b)

8a $\overline{(x + y)} = \bar{x} \cdot \bar{y}$ 8b $\overline{(x \cdot y)} = \bar{x} + \bar{y}$

x	y	\bar{x}	\bar{y}	$x + y$	$\overline{(x + y)}$	$\bar{x} \cdot \bar{y}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

c.v.d.

Osservazioni

1. I teoremi di destra si possono ottenere da quelli di sinistra scambiando OR con AND e "0" con "1"
2. Principio di dualità
3. Molti dei teoremi visti sono veri anche nell'algebra che conosciamo
4. Particolarmente significativi sono i teoremi di De Morgan e la proprietà distributiva
5. Molti teoremi, in particolare quelli di De Morgan, sono veri anche per "n" variabili

Esempio 1

- Semplificare la seguente espressione:

$$(x + z) \cdot (x + \bar{z}) \cdot (\bar{y} + z)$$

- In base ai teoremi visti si ha:

$$\begin{aligned}(x + z) \cdot (x + \bar{z}) \cdot (\bar{y} + z) &= (x + z \cdot \bar{z}) \cdot (\bar{y} + z) && \text{P 4b} \\ &= (x + 0) \cdot (\bar{y} + z) && \text{P 5b} \\ &= x \cdot (\bar{y} + z) && \text{P 2a}\end{aligned}$$

1a $x+1=1$

1b $x \cdot 0 = 0$

2a $\overline{\overline{x}} = x$

1a Somma logica (+)

1b Prodotto logico (\bullet)

2a $x+0=x$

2b $x \bullet 1 = x$

3a $x+x=x$

3b $x \bullet x = x$

3a $x+y=y+x$

3b $x \bullet y = y \bullet x$

4a $x+xy=x$

4b $x(x+y)=x$

4a $x \bullet (y+z) = (x \bullet y) + (x \bullet z)$

4b $x + (y \bullet z) = (x+y) \bullet (x+z)$

5a $x + \bar{x}y = x + y$

5b $x(\bar{x} + y) = xy$

5a $x + \bar{x} = 1$

5b $x \bullet \bar{x} = 0$

6a $x + (y+z) = x + y + z$

6b $x(yz) = (xy)z = xyz$

7a $xy + \bar{x}z + yz = xy + \bar{x}z$

7b $(x+y)(\bar{x}+z)(y+z) = (x+y)(\bar{x}+z)$

8a $\overline{(x+y)} = \bar{x} \cdot \bar{y}$

8b $\overline{(x \cdot y)} = \bar{x} + \bar{y}$

Esempio 1'

- Per altra via; posto:
- si ha:

$$T = (x + \bar{z})$$

$$\begin{aligned}
 (x + z) \cdot (x + \bar{z}) \cdot (\bar{y} + z) &= (x + z) \cdot T \cdot (\bar{y} + z) \\
 &= (T \cdot x + T \cdot z) \cdot (\bar{y} + z) && \text{P 4b} \\
 &= (xx + xz + \bar{z}x + \bar{z}z) \cdot (\bar{y} + z) && \text{P 4a} \\
 &= (x + xz + \bar{z}x + \bar{z}z) \cdot (\bar{y} + z) && \text{P 3b} \\
 &= (x + 0) \cdot (\bar{y} + z) \\
 &= x \cdot (\bar{y} + z)
 \end{aligned}$$

1a $x+1=1$

1b $x \bullet 0 = 0$

2a $(\bar{\bar{x}}) = x$

3a $x+x=x$

3b $x \bullet x = x$

4a $x+xy=x$

4b $x(x+y)=x$

5a $x+\bar{x}y=x+y$

5b $x(\bar{x}+y)=xy$

6a $x+(y+z)=x+y+z$

6b $x(yz)=(xy)z=xyz$

7a $xy+\bar{x}z+y\bar{z}=xy+\bar{x}z$

7b $(x+y)(\bar{x}+z)(y+z)=(x+y)(\bar{x}+z)$

8a $(\overline{x+y}) = \bar{x} \cdot \bar{y}$

8b $(\overline{x \cdot y}) = \bar{x} + \bar{y}$

1a Somma logica (+)

1b Prodotto logico (\bullet)

2a $x+0=x$

2b $x \bullet 1 = x$

3a $x+y=y+x$

3b $x \bullet y = y \bullet x$

4a $x \bullet (y+z) = (x \bullet y) + (x \bullet z)$

4b $x + (y \bullet z) = (x+y) \bullet (x+z)$

5a $x+\bar{x}=1$

5b $x \bullet \bar{x} = 0$

Premessa 1

- Osservazioni

- le funzioni AND, OR e NOT costituiscono un insieme *funzionalmente completo* di operatori logici

- In base al teorema di De Morgan si ha:

$$x + y = \overline{\overline{x} \cdot \overline{y}}$$

- ovvero la funzione OR si può realizzare con le funzioni AND e NOT quindi:
- le funzioni **AND e NOT** costituiscono un insieme *funzionalmente completo* di operatori logici

Premessa 2

- Osservazioni

- Sempre in base al teorema di De Morgan si ha:

$$x \cdot y = \overline{\overline{x} + \overline{y}}$$

- ovvero la funzione AND si può realizzare con le funzioni OR e NOT quindi
- le funzioni **OR e NOT** costituiscono un insieme *funzionalmente completo* di operatori logici
- le funzioni OR e AND **non** costituiscono un insieme *funzionalmente completo* di operatori logici perché non è possibile realizzare la funzione NOT

Definizione

- Le funzioni NAND e NOR sono definite dalle seguenti tabelle di verità

x	y	u
0	0	1
0	1	1
1	0	1
1	1	0

$$\text{NAND} \quad u = \overline{(x \cdot y)}$$

x	y	u
0	0	1
0	1	0
1	0	0
1	1	0

$$\text{NOR} \quad u = \overline{(x + y)}$$

Osservazioni

- NAND e NOR sono contrazioni di NOT-AND e NOT-OR
- la funzione **NAND** costituisce un insieme *funzionalmente completo* di operatori logici

$$\overline{(x \cdot x)} = \bar{x} \quad \overline{\overline{(x \cdot y)}} = x \cdot y$$

- la funzione **NOR** costituisce un insieme *funzionalmente completo* di operatori logici

$$\overline{(x + x)} = \bar{x} \quad \overline{\overline{(x + y)}} = x + y$$

Funzioni "complesse" 1

- L'operatore "XOR", OR ESCLUSIVO è:

$$x \oplus y$$

- Definizione

x	y	u
0	0	0
0	1	1
1	0	1
1	1	0

$$x \oplus y = x \cdot \bar{y} + \bar{x} \cdot y = (x + y) \cdot (\bar{x} + \bar{y}) = (x + y) \cdot \overline{(x \cdot y)}$$

Funzioni "complesse" 2

- L'operatore "XNOR", NOR ESCLUSIVO è:

$$\overline{x \oplus y}$$

- Definizione

x	y	u
0	0	1
0	1	0
1	0	0
1	1	1

$$\overline{x \oplus y} = x \cdot y + \bar{x} \cdot \bar{y} = (x + \bar{y}) \cdot (\bar{x} + y)$$

Sintesi ad occhio di Reti Combinatorie

Ciclo

- **Definizione**

- **Ciclo: Percorso chiuso che attraversa k blocchi ($k \geq 1$) tutti nella loro direzione di funzionamento**

- **Osservazioni**

- **Tutte le reti viste sono prive di cicli**
- **I blocchi base combinatori sono privi di cicli**
- **Le funzioni descrivibili dalle tabelle di verità sono tutte prive di cicli (le uscite sono funzione dei solo ingressi)**

- **Conclusione**

- **Tutte le reti logiche composte di blocchi combinatori e prive di cicli sono reti combinatorie**

Sintesi di reti combinatorie

- **Sintesi**

- data la descrizione ai terminali di una rete combinatoria
- ottenere la struttura in blocchi logici e le relative interconnessioni

- **Osservazioni**

- il funzionamento della rete deve essere possibile descriverlo mediante una tabella di verità
- non esiste una sola realizzazione
- per poter scegliere fra le varie soluzioni è necessario definire il parametro da ottimizzare
- Funzione *COSTO*
- (numero di blocchi base, ritardo ingresso uscita, uso di particolari blocchi,)

- **VEDERE ESEMPI SUCCESSIVI**

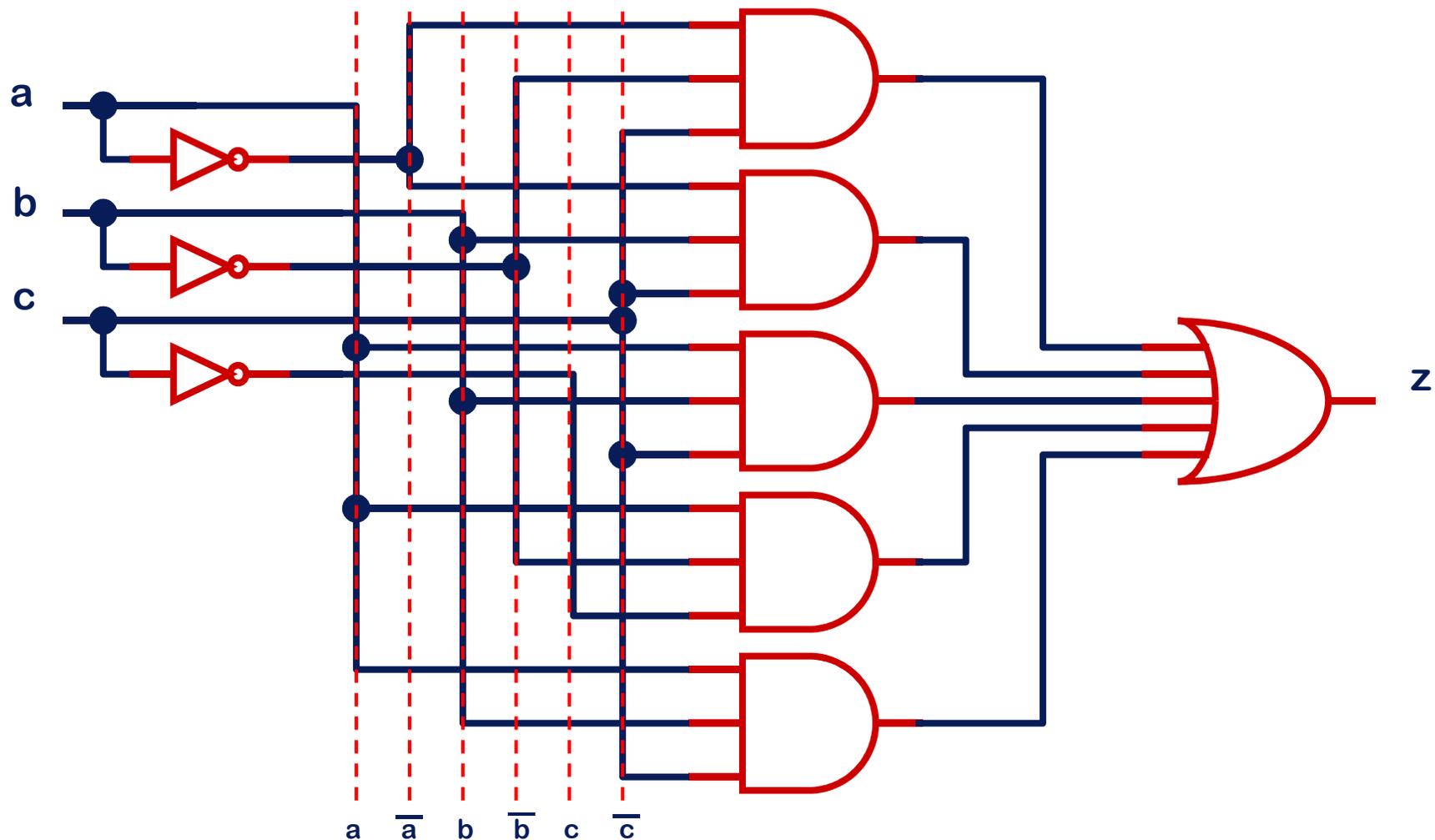
Esempio di funzione

- Data la funzione definita dalla Tabella di Verità:

a	b	c	z	
0	0	0	1	$z = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$
0	0	1	0	$= \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$
0	1	0	1	$= \bar{c} + a \cdot \bar{b} = c \cdot \overline{(a \cdot \bar{b})} = c \cdot \overline{(a + b)} = \overline{c \cdot a} + c \cdot b$
0	1	1	0	$= \overline{(a + c)} + \overline{(c + b)}$
1	0	0	1	$z = (a + b + \bar{c}) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$
1	0	1	1	
1	1	0	1	
1	1	1	0	

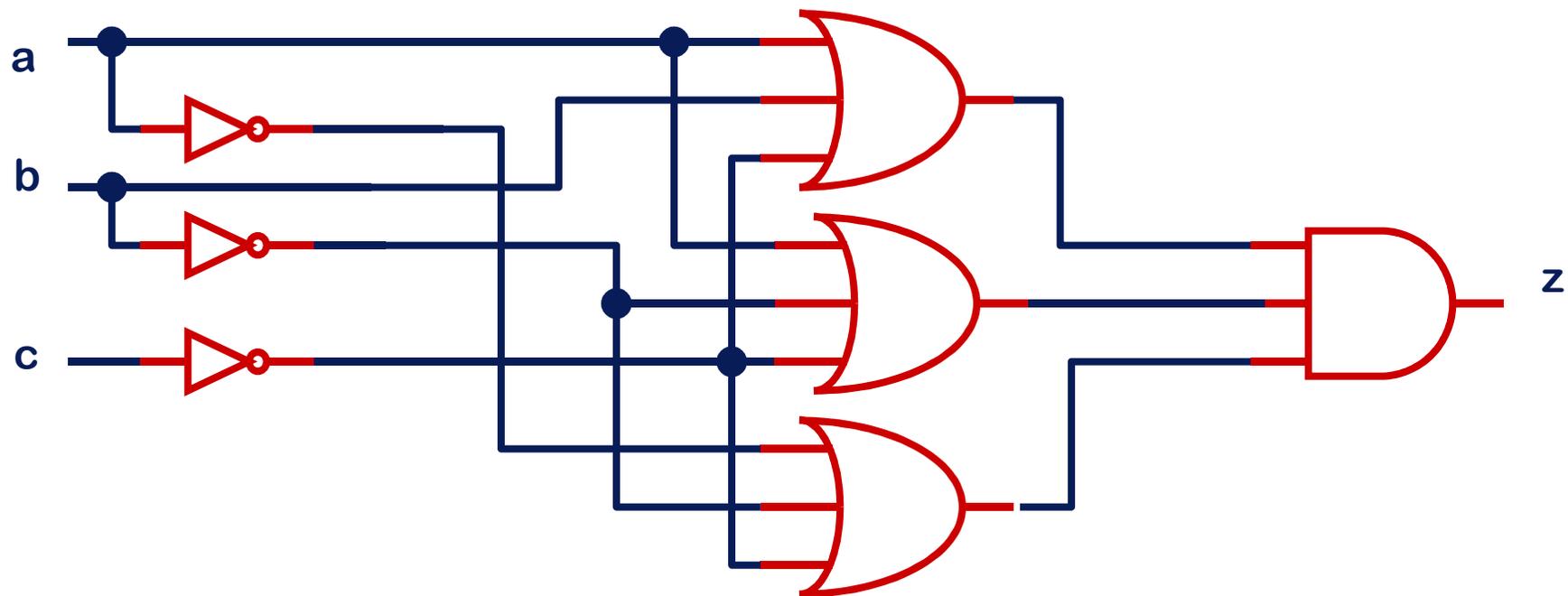
Schemi relativi 1

$$z = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$



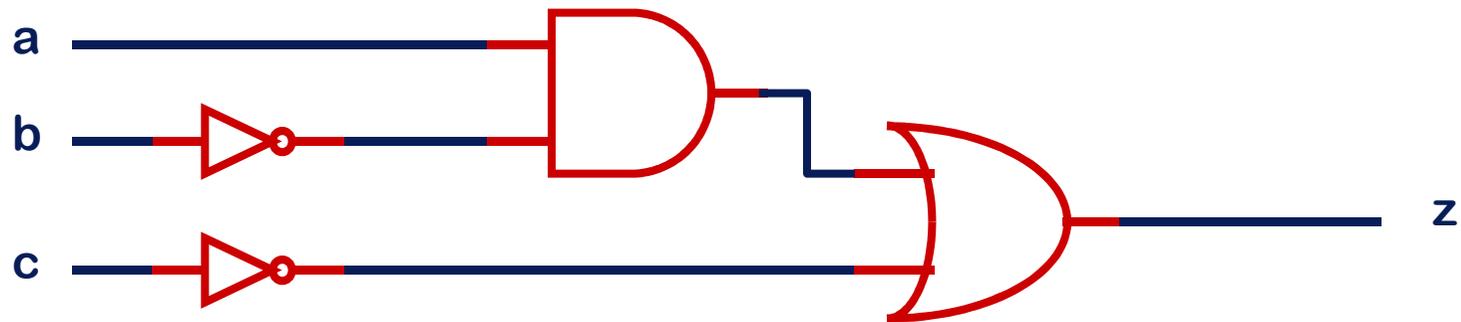
Schemi relativi 2

$$z = (a + b + \bar{c}) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$$



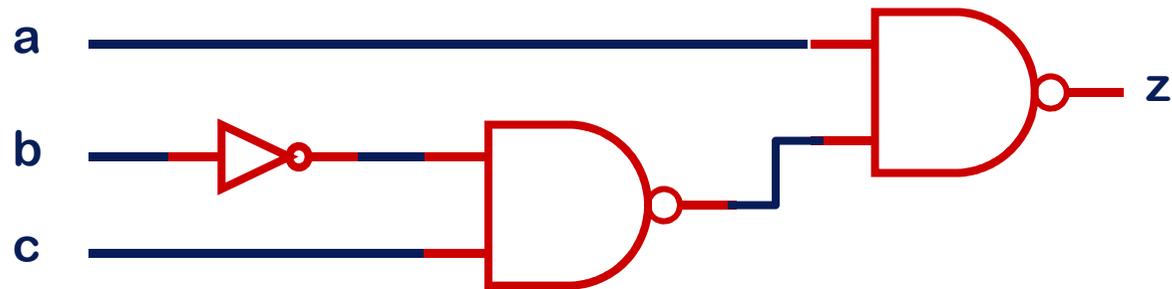
Schemi relativi 3

$$z = \bar{c} + a \cdot \bar{b}$$

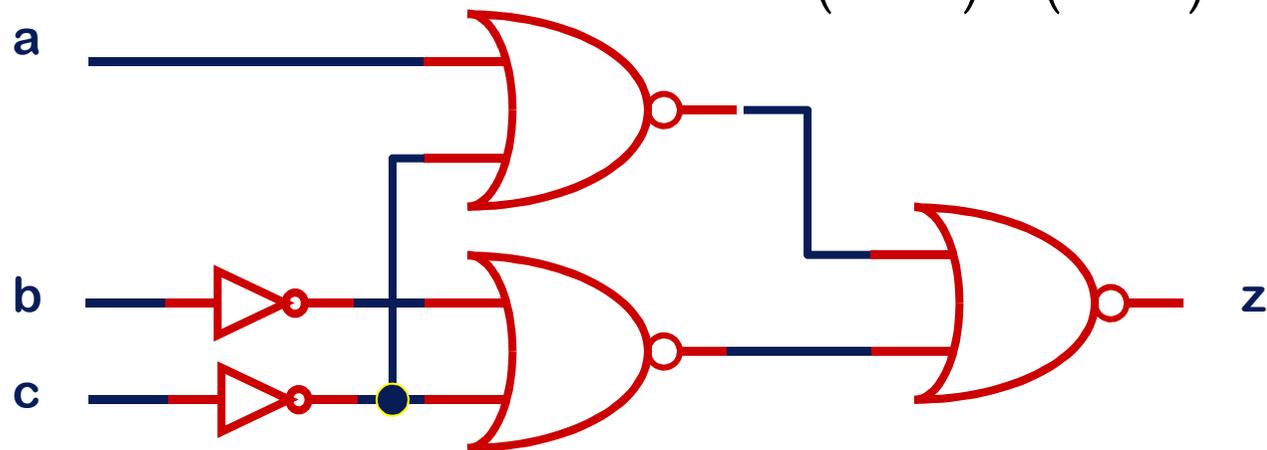


Schemi relativi 4

$$z = \overline{c \cdot (a \cdot \bar{b})}$$



$$z = \overline{(a + c) + (c + \bar{b})}$$

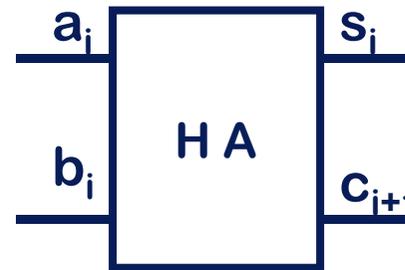
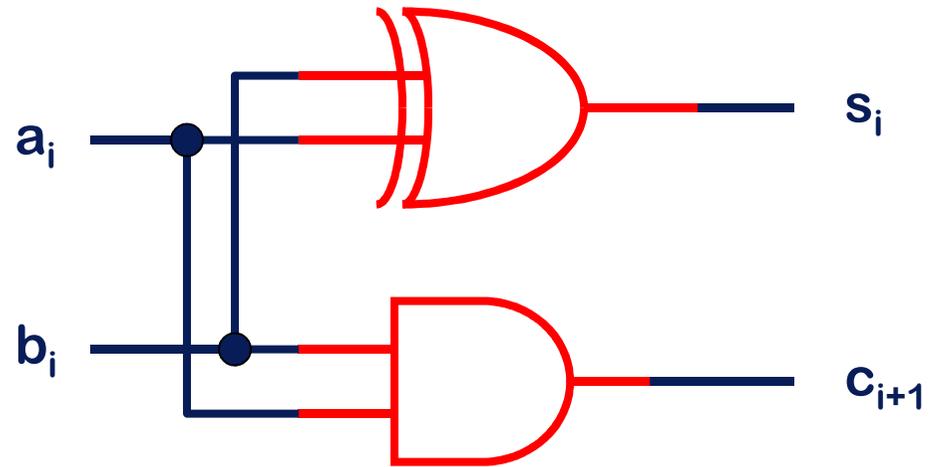


Half Adder

- Somma di due bit (senza riporto in ingresso)

a_i	b_i	s_i	c_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$s_i = a_i \oplus b_i$$
$$c_{i+1} = a_i \cdot b_i$$



Full Adder 1

- Somma di due bit compreso il Carry

c_i	a_i	b_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

s_i

$c_i \backslash a_i, b_i$	00	01	11	10
0		1		1
1	1		1	

c_{i+1}

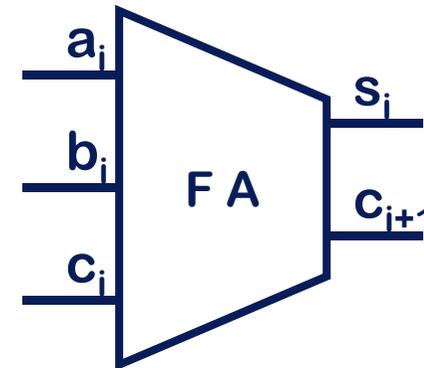
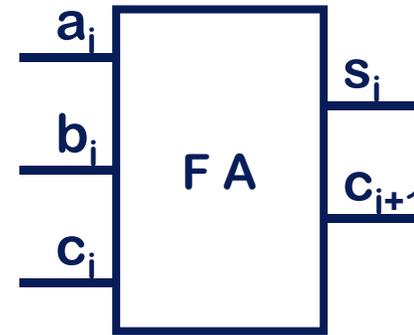
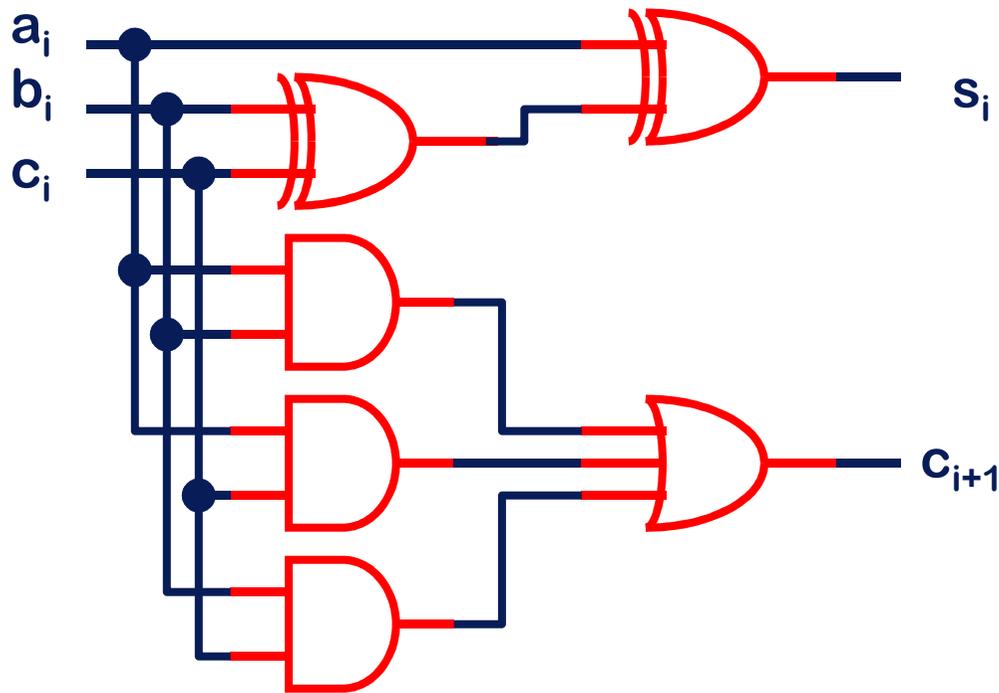
$c_i \backslash a_i, b_i$	00	01	11	10
0			1	
1		1	1	1

Full Adder 2

- Lo schema risulta

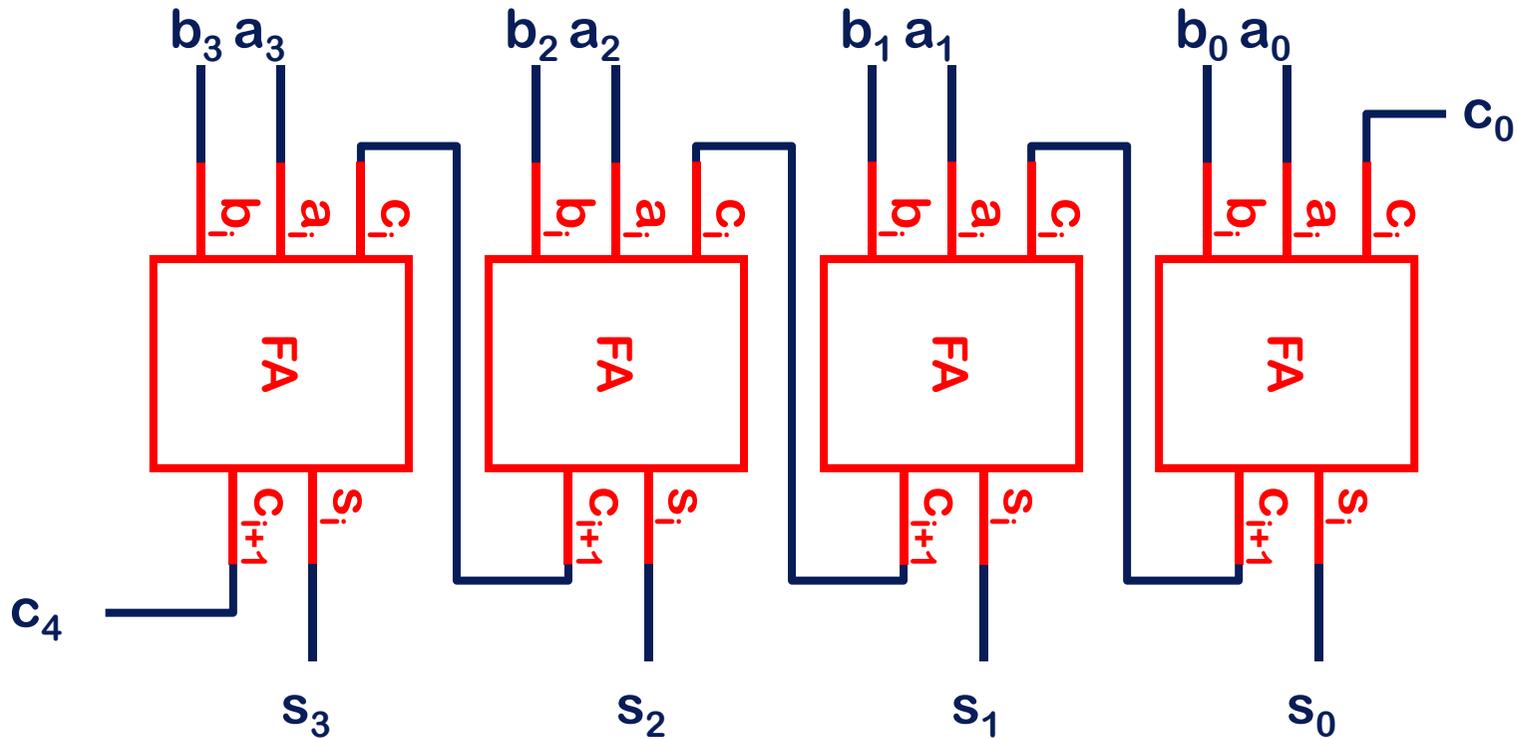
$$s_i = a_i \oplus b_i \oplus c_i = a_i \oplus (b_i \oplus c_i)$$

$$c_{i+1} = a_i \cdot b_i + a_i \cdot c_i + b_i \cdot c_i$$



Sommatore a riporto seriale (Ripple-Carry Adder)

- Somma di due parole di 4 bit in $C. 2$



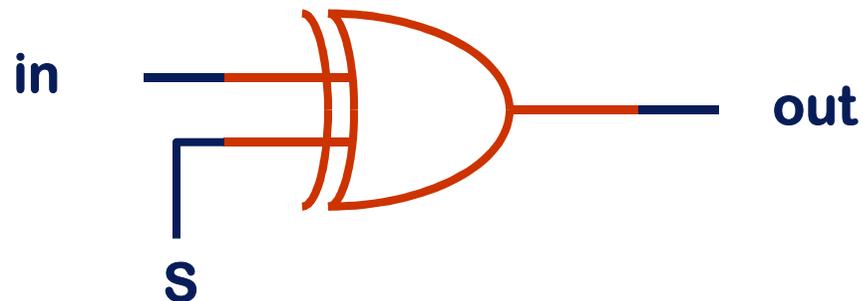
Proprietà dello XOR

- Lo XOR può essere visto come un inverter "programmabile"

S	in	out
0	0	0
0	1	1
1	0	1
1	1	0

per $S = 0$ è $out = in$

per $S = 1$ è $out = \overline{in}$



Considerazioni sulla sottrazione

- Si ricorda che

$$W = A - B = A + (-B)$$

- Operando in complemento a 2 si ha

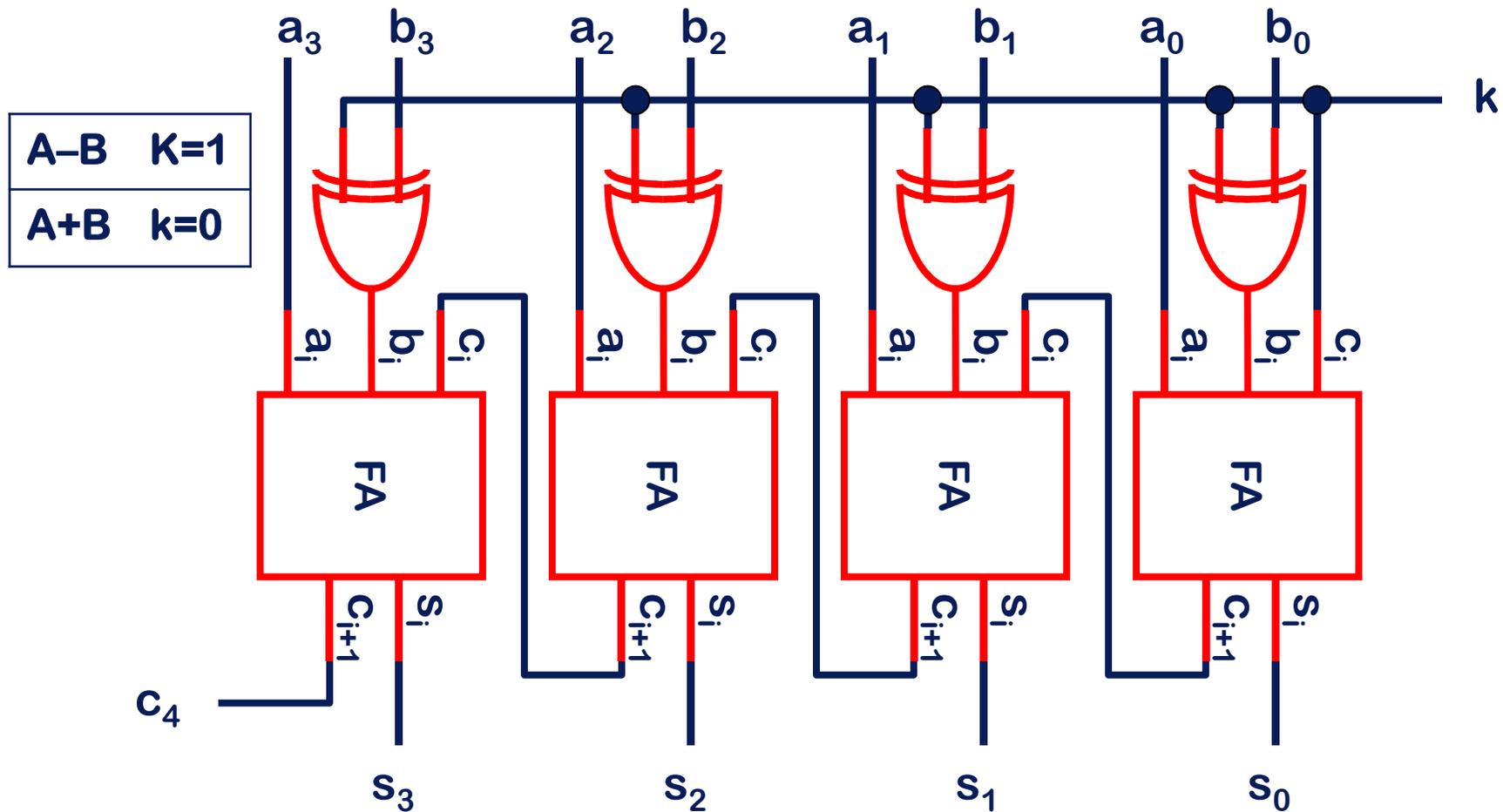
$$-B = (\overline{B}) + 1$$

- Quindi

$$W = A - B = A + (\overline{B}) + 1$$

Sommatore/Sottrattore

- In base alle proprietà dello XOR e come si può eseguire la differenza $(A - B)$ in C . 2 si ha:



Conclusioni

- Postulati
- Principio di dualità
- Teoremi fondamentali
- insieme *funzionalmente completo* NAND e NOR
- Funzione XOR
- Reti logiche combinatorie e sequenziali
- Simboli
- Concetto di minimizzazione (funzione costo)
- Realizzazioni diverse della stessa funzione
- Half Adder e Full Adder
- Sommatore di due word di n bit