Algebra di Boole e circuiti

dalle funzioni logiche
ai circuiti digitali

A. Borghese, F. Pedersini

Dip. Informatica
Università degli Studi di Milano

George Boole, 1854:
"An Investigation of the Laws of Thought on which to found the Mathematical Theories of Logic and Probabilities"

Algebra di Boole

Algebra basata su:

- Variabili binarie: **FALSE** (=**0**); **TRUE** (=**1**)
- Operatori logici sulle variabili: **NOT**, **AND**, **OR**

Applicazioni:

- Rappresentazione delle informazioni (in un elaboratore)
- Analisi / Sintesi dei circuiti digitali
  - ANALISI: Dal sistema alla descrizione formale del suo funzionamento.
  - SINTESI: dalla descrizione (funzione logica) al progetto del sistema che la implementa.

Funzione logica / circuito logico

Funzione logica $\quad f: \mathbf{B}^N \rightarrow \mathbf{B}\ , \quad \mathbf{B} = \{\mathbf{V}, \mathbf{F}\} \cong \{0, 1\}$

funzione booleana di *N* variabili booleane

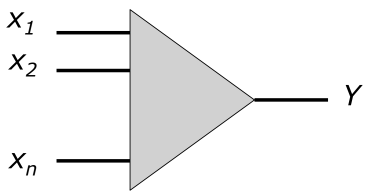
- Può essere rappresentata mediante combinazione di **variabili** e **operatori** logici elementari (**NOT**, **AND**, **OR**)
- Definita per tutte le 2^N **combinazioni** delle variabili (ingressi)

Può essere rappresentata in **tre** diversi modi:

1. Espressione booleana: $\quad \mathbf{Y} = f(x_1, x_2, \ldots, x_n)$

2. Circuito logico

- **1 uscita Y,** funzione booleana di
- **N ingressi** $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ variabili booleane



x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

3. Tabella di verità (Truth Table, TT)

- Definizione della funzione come **elenco** dei valori della funzione per tutte le possibili combinazioni di valori delle variabili ($\mathbf{2}^N$).

Operatori logici elementari

Operatore logico NOT

$$\mathrm{NOT}: \quad \mathrm{B} \rightarrow \mathrm{B}$$

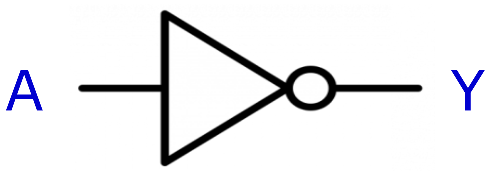
- Operazione logica di **negazione**
  - Se **A** è vera, **NOT(A)** è falsa

$$Y = \mathrm{NOT}(A) = \bar{A} = \sim A$$

- Operazione definita dalla **tabella di verità**
  - Funzione definita per tutte le combinazioni di variabili

Tabella di verità

A	Y
0	1
1	0



Negazione logica
("Inverter")

Operatore AND

Operatore logico AND

$$\text{AND}: \quad B^2 \rightarrow B$$

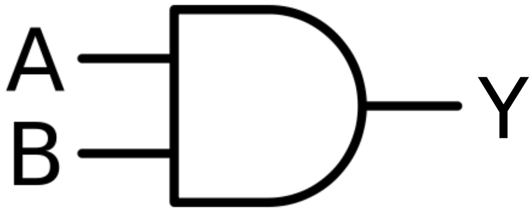
Operazione di **prodotto logico**

- Solo se sia A che B sono veri, A AND B è vera.

$$\mathbf{Y} = \mathbf{A}\ \text{AND}\ \mathbf{B} = \mathbf{A} \wedge \mathbf{B} = \mathbf{A} \cdot \mathbf{B} = \mathbf{AB}$$

Tabella di verità

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



Prodotto logico
(porta **AND**)

Operatore OR

Operatore logico OR

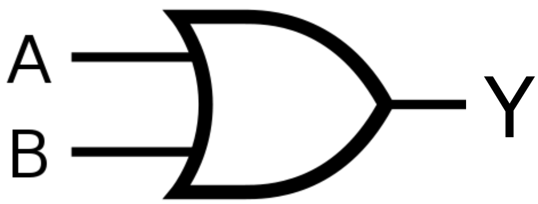
$$OR: \quad B^2 \rightarrow B$$

- Operazione di **somma logica**
  - Se A o B sono veri, che A OR B è vera.

$$\mathbf{Y} = \mathbf{A}\ \text{OR}\ \mathbf{B} = \mathbf{A} \vee \mathbf{B} = \mathbf{A} + \mathbf{B}$$

Tabella di verità

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



Somma logica
(porta **OR**)

Priorità degli operatori

Regola di **priorità** degli operatori

In assenza di parentesi:

- AND ha la priorità sull'OR,
- NOT su entrambi:

NOT → AND → OR

- Esempi:

$$A \text{ OR } B \text{ AND } C = A + B \cdot C = A + (B \cdot C)$$

$$\text{NOT } A \text{ AND } C = \text{NOT } A \cdot C = (\text{NOT } A) \cdot C = \overline{A} \cdot C$$

Dualità e Postulati

Principio di **dualità**:

se un'espressione booleana è vera, è vera anche la sua duale

il DUALE di un'espressione booleana si ottiene:

- scambiando **AND** con **OR** (OR→AND , AND→OR)
- scambiando **TRUE (1)** con **FALSE (0)** (0→1 , 1→0)

Esempio: $\boxed{A \cdot \overline{A} = 0 \quad \leftrightarrow \quad A + \overline{A} = 1}$

Postulati dell'algebra di Boole:

- *Le proprietà: commutativa, distributiva, identità, elemento inverso sono postulate: assunte vere per definizione.*
- *Le altre proprietà sono teoremi che si ottengono per dimostrazione a partire dai postulati.*

Proprietà degli operatori logici

Proprietà	AND	OR (duale)
• Identità	$1 \cdot x = x$	$0 + x = x$
• Elem. zero	$0 \cdot x = 0$	$1 + x = 1$
• Idempotenza	$x \cdot x = x$	$x + x = x$
• El. inverso	$x \cdot \sim x = 0$	$x + \sim x = 1$
• Commutativa	$x \cdot y = y \cdot x$	$x + y = y + x$
• Associativa	$(x \cdot y) \cdot z = x \cdot (y \cdot z) = xyz$	$(x+y) + z = x + (y+z) = x+y+z$
	AND rispetto a OR	**OR rispetto a AND**
Distributiva	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + y \cdot z = (x+y) \cdot (x+z)$
I assorbimento	$x \cdot (x + y) = x$	$x + x \cdot y = x$
II assorbimento	$x \cdot (\sim x + y) = x \cdot y$	$x + \sim x \cdot y = x + y$
De Morgan	$\overline{(xy)} = \overline{x} + \overline{y}$	$\overline{(x + y)} = \overline{x} \cdot \overline{y}$

Operatori logici composti

Operatore logico XOR

$$\text{XOR}: \quad B^2 \rightarrow B$$

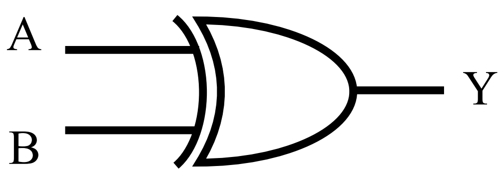
- Operazione di **mutua esclusione:**
  Y è vera se e solo se **o A o B sono veri**, ma **non entrambi**

$$Y = A \text{ XOR } B = A \oplus B$$

Tabella di verità

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

Porta logica XOR



XOR espresso con gli operatori fondamentali:

$$A \oplus B = A\overline{B} + \overline{A}B = (A + B)(\overline{A} + \overline{B})$$

Proprietà: A XOR B è **VERA** quando A e B sono **DIVERSI**

Operatore NAND

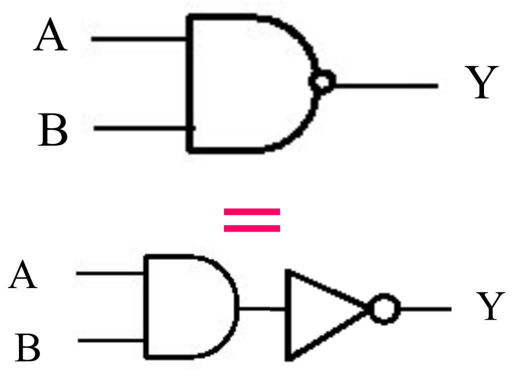
Operatore logico NAND (Not-AND)

- Operatore **AND negato**

A **NAND** B = **NOT**(A **AND** B)

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

operatore "NAND"



Operatore NOR

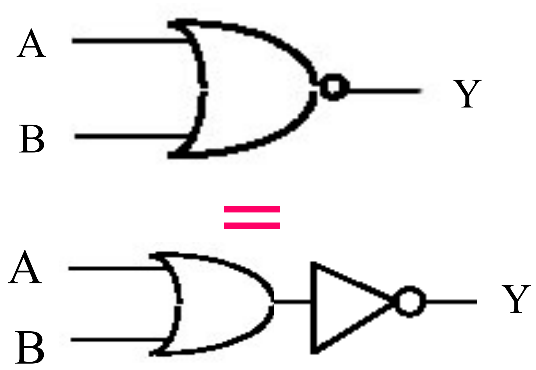
Operatore logico NOR (Not-OR)

- Operatore **OR negato**

A **NOR** B = **NOT**(A **OR** B)

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

operatore "NOR"



Porte universali

Quale è il numero minimo di porte con cui è possibile implementare tutte le altre?

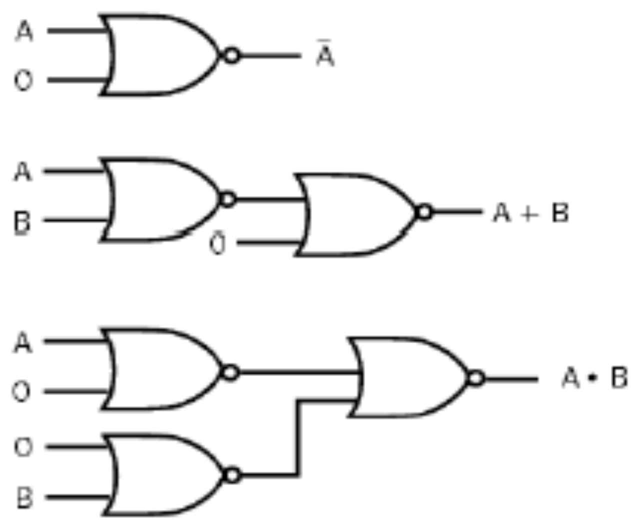
- Con la legge di De-Morgan riusciamo a passare da 3 a 2:
  - con NOT e AND si ottiene OR:

NOT(NOT(**A**) AND NOT(**B**)) = **A** OR **B**

- E' possibile usarne una sola?
  - Sì, ad esempio la porta **NAND** e la **NOR** che sono chiamate **porte universali**

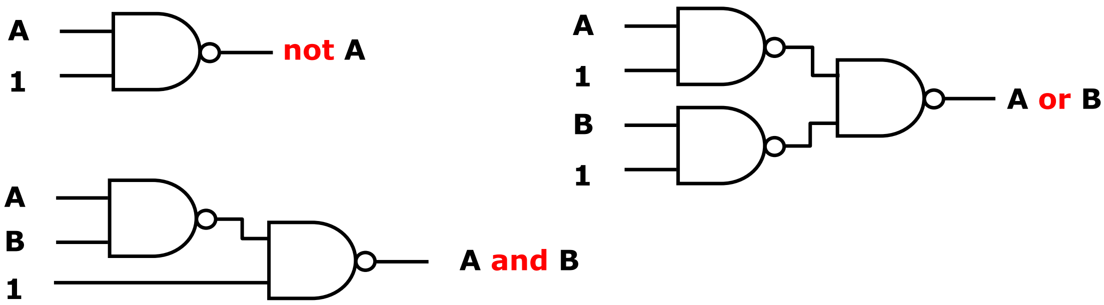
Porta Universale NOR

NOT A = 0 NOR A = A NOR A
A OR B = (A NOR B) NOR 0
A AND B = (A NOR 0) NOR (B NOR 0)



Porta Universale NAND

NOT **A**	=	**1** NAND **A** **=** **A** NAND **A**
A AND **B**	=	(**A** NAND **B**) NAND **1**
A OR **B**	=	(**A** NAND **1**) NAND (**B** NAND **1**)



Circuiti digitali

Ricordando che:

- Un oggetto di **materiale conduttore** si trova tutto allo stesso potenziale elettrico (**equipotenziale**)
- Un **generatore di tensione** (batteria, alimentatore) genera una **differenza di potenziale** tra due conduttori detti **POLI**: **positivo (+)** e **negativo (–)**

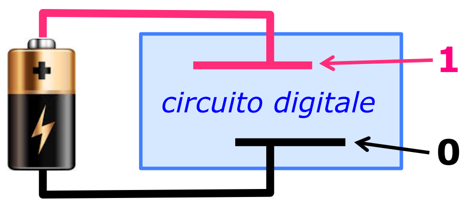
Definiamo:

- **TENSIONE** su un conduttore: **differenza di potenziale** tra il conduttore ed un conduttore di riferimento ➔ **polo negativo**

In un circuito digitale ho **2 TENSIONI possibili** per ogni conduttore:

- **Tensione MASSIMA** (potenziale del polo **+**) ➔ **"1"**
- **Tensione MINIMA**: 0 Volt (potenziale del polo **–**) ➔ **"0"**

"1": collegamento elettrico a **"+"**
"0": collegamento elettrico a **"–"**

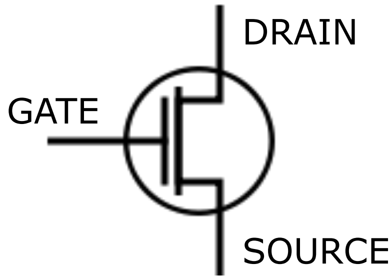


Il transistore **MOSFET**

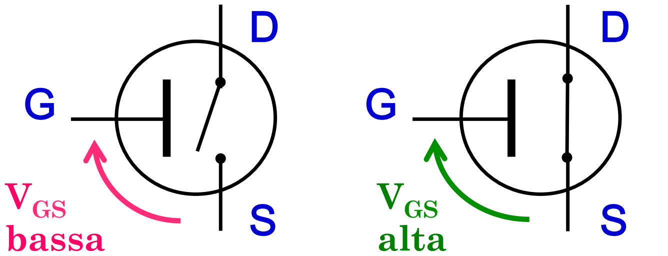
MOSFET: **M**etal-**O**xide-**S**emiconductor **F**ield **E**ffect **T**ransistor

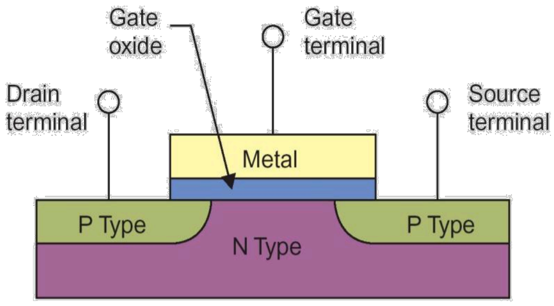
Modello di funzionamento del MOSFET:

collegamento elettrico tra **DRAIN** e **SOURCE** comandato dalla tensione su **GATE:**



- Tensione **V_{GS} bassa → D, S isolati**
  - MOSFET in stato di INTERDIZIONE
- Tensione **V_{GS} alta → D, S collegati**
  - MOSFET in stato di SATURAZIONE

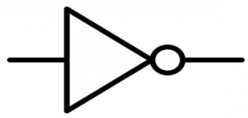


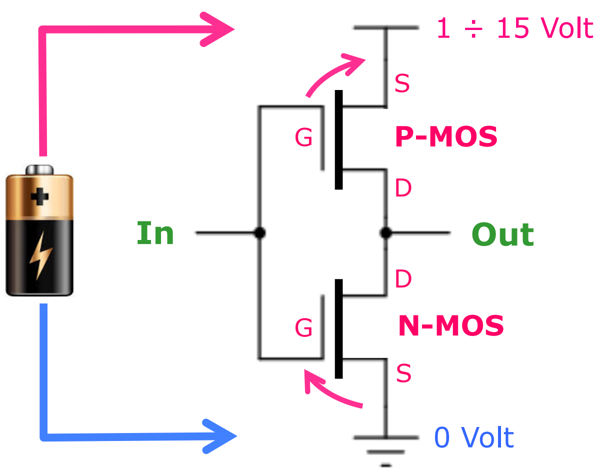


La tecnologia CMOS (1980 – oggi)

- **CMOS: Complementary–MOS**
  MOSFET a coppie complementari (**N-MOS** + **P-MOS**) che lavorano in "contrapposizione"
  - Se un **N-MOS conduce**
    ➔ il corrispondente **P-MOS è isolato**
    e viceversa

Inverter CMOS

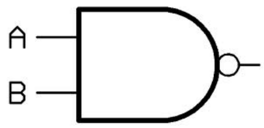




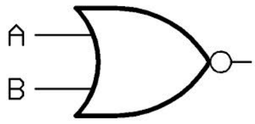
- Vantaggi tecnologia CMOS:
  - Tensione di alimentazione "flessibile":
    - V_{CC} = 1 ÷ 15 Volt
    - V_{LOW} = 0 ÷ $V_{CC}/2$
    - V_{HIGH} = $V_{CC}/2$ ÷ V_{CC}
  - Consumo bassissimo:
    - Consuma solo nella transizione
    - In condizioni statiche, consumo nullo!

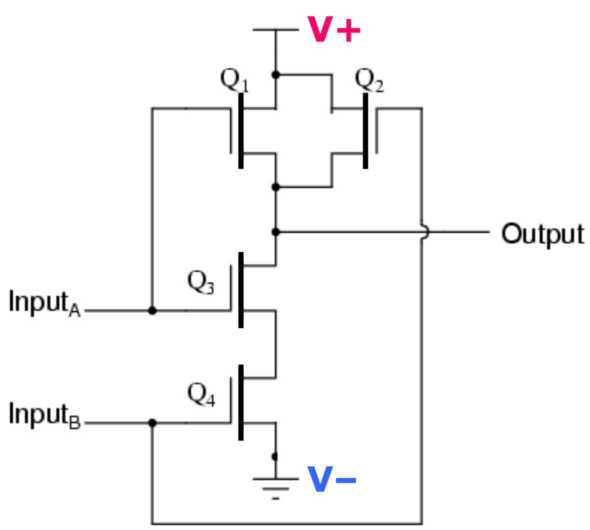
Porte CMOS

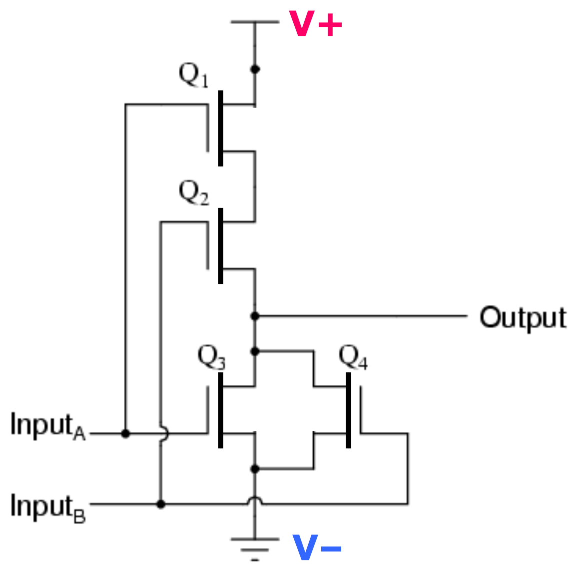
Porta **NAND**



Porta **NOR**







Funzione logica / circuito logico

Funzione logica: $f: \mathbf{B}^N \rightarrow \mathbf{B}$

funzione booleana di *N* variabili booleane

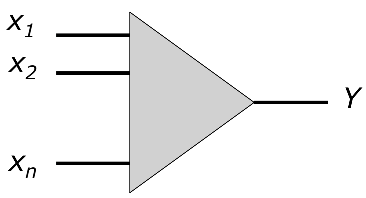
- Può essere rappresentata mediante combinazione di variabili e operatori elementari (NOT,AND,OR)
- Definita per tutte le 2^N combinazioni delle variabili (ingressi)

Può essere rappresentata in tre diversi modi:

1. Espressione: $\mathbf{Y} = f(x_1, x_2, \ldots, x_n)$

2. Circuito logico

- **1 uscita Y,** funzione booleana di
- **N ingressi ($x_1, \ldots, x_n$)** variabili booleane



x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

3. Tabella di verità (Truth Table, TT)

- Definizione della funzione come **elenco** dei valori della funzione per tutte le possibili combinazioni di valori delle variabili (2^N).

Funzione / circuito / tab. verità

Esempio: $$F\left(A,B,C\right)=A\cdot B+B\cdot\overline{C}$$

3 variabili: $F = f\,(A,B,C)$ → 2^3 = 8 combinazioni possibili delle variabili

Circuito logico

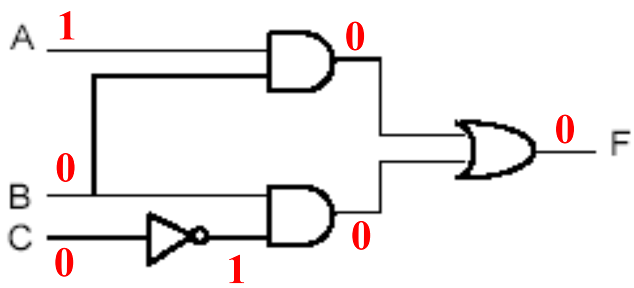


Tabella di verità

A	B	C	A•B	B•not C	F
0	0	0	0	0	**0**
0	0	1	0	0	**0**
0	1	0	0	1	**1**
0	1	1	0	0	**0**
1	0	0	0	0	**0**
1	0	1	0	0	**0**
1	1	0	1	1	**1**
1	1	1	1	0	**1**

*Data una funzione **F**,*
*esistono **infinite espressioni** e **infiniti circuiti**,*
*ma **una sola tabella di verità** che la rappresenta.*

Sintesi di circuiti combinatori

*Dal circuito/espressione alla tabella di verità → **ANALISI***

*Problema della **SINTESI (progetto) di circuiti** combinatori:*
*Come passare da: **tabella di verità***
*a: **espressione logica / circuito digitale?***

Data la tabella di verità:

A	B	C	F
0	0	0	0
0	0	1	0
0	**1**	**0**	**1**
0	1	1	0
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**1**

F = 1 ***se e solo se:***

A = 0 AND B = 1 AND C = 0
OR
A = 1 AND B = 1 AND C = 0
OR
A = 1 AND B = 1 AND C = 1

Funzione: espressione / tabella di verità

F = 1 ***se e solo se:***

A = 0 AND B = 1 AND C = 0

OR

A = 1 AND B = 1 AND C = 0

OR

A = 1 AND B = 1 AND C = 1

$F = 1$ se e solo se:

$\left(\overline{A} = 1 \text{ and } B = 1 \text{ and } \overline{C} = 1\right)$ or

$\left(A = 1 \text{ and } B = 1 \text{ and } \overline{C} = 1\right)$ or

$\left(A = 1 \text{ and } B = 1 \text{ and } C = 1\right)$

$F = 1$ se e solo se: $\overline{A}B\overline{C} = 1$ or $AB\overline{C} = 1$ or $ABC = 1$

$F = 1$ se e solo se: $\overline{A}B\overline{C} + AB\overline{C} + ABC = 1$

$$\boxed{F = \overline{A}B\overline{C} + AB\overline{C} + ABC}$$

La prima forma canonica

$$F = A \cdot B + B \cdot \overline{C} = \boxed{\overline{A}B\overline{C} + AB\overline{C} + ABC}$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Implicante:
Prodotto delle variabili (in forma naturale o negata) per le quali **la funzione vale 1**

Mintermine m_j:
implicante che contiene **tutte le n variabili** della funzione (e.g. ABC).

$$\text{Prima forma canonica (SoP)}: \quad F = \sum_{j=1}^{Q} m_j \quad , \quad Q \leq 2^n$$

Prima forma canonica (SoP) di una **funzione logica F**:
la **somma logica** dei suoi **mintermini**

Somma di Prodotti

Regola di costruzione:

considero i MINTERMINI (prodotti delle var. per cui: **F = 1**)

MINTERMINI: prodotti di **tutte** le variabili, con le variabili **NEGATE** se nella tabella di verità sono **0**, **NATURALI** se sono **1**

$$\text{Prima forma canonica (SoP)}: \quad F = \sum_{j=1}^{Q} m_j \quad , \quad Q \leq 2^n$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$F = \overline{A}B\overline{C} + AB\overline{C} + ABC$$

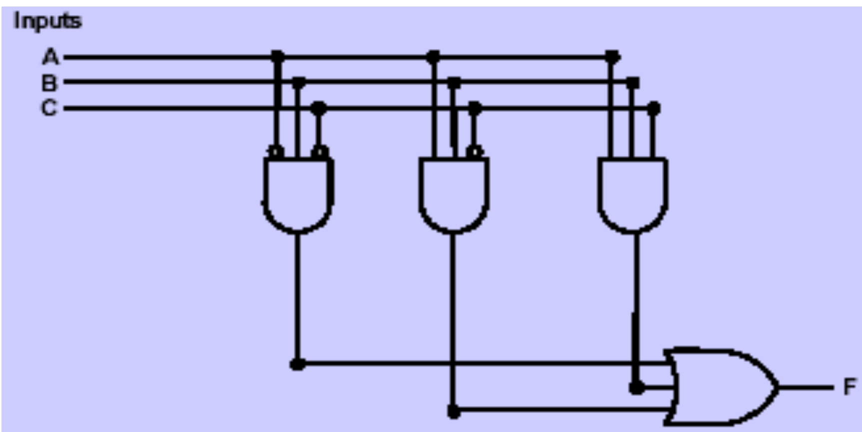
I forma canonica: dall'espressione al circuito

Sintesi del circuito:

$$F = \overline{A}B\overline{C} + AB\overline{C} + ABC$$

Circuito a **due stadi**:

1. Stadio **AND**: **Q porte AND a n ingressi**, una per ogni mintermine
2. Stadio **OR**: **1 porta OR a Q ingressi**



Esercizio: *funzione maggioranza*

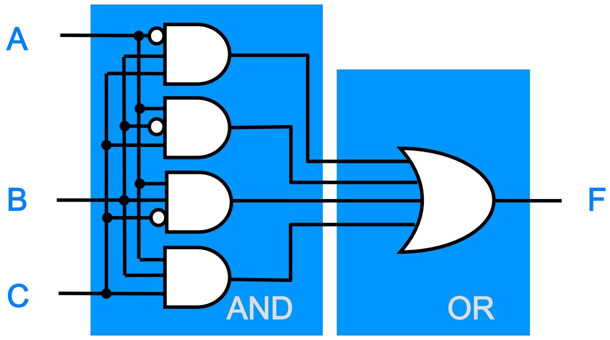
Funzione logica di 3 variabili ➔ 3 ingressi, 1 uscita

1. Costruzione tabella di verità o espressione logica
2. Trasformazione a forma SOP
3. Eventuale semplificazione

$$F(A,B,C) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

I forma canonica



Seconda forma canonica

Seconda forma canonica di ***F(A,B,C)***:

- Approccio DUALE rispetto alla I forma canonica: considero i casi in cui: **F = 0**

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

F = 0 *se e solo se:*

A=0 and B=0 and C=0
OR
A=0 and B=0 and C=1
OR
A=0 and B=1 and C=1
OR
A=1 and B=0 and C=0
OR
A=1 and B=0 and C=1

Funzione: espressione / tabella di verità

F = 0 *se e solo se:*

A=0 and B=0 and C=0 OR
A=0 and B=0 and C=1 OR
A=0 and B=1 and C=1 OR
A=1 and B=0 and C=0 OR
A=1 and B=0 and C=1

$F = 0$ se e solo se:

$\left(\overline{A} = 1 \text{ and } \overline{B} = 1 \text{ and } \overline{C} = 1\right)$ or

$\left(\overline{A} = 1 \text{ and } \overline{B} = 1 \text{ and } C = 1\right)$ or

$\left(\overline{A} = 1 \text{ and } B = 1 \text{ and } C = 1\right)$ or

$\left(A = 1 \text{ and } \overline{B} = 1 \text{ and } \overline{C} = 1\right)$ or

$\left(A = 1 \text{ and } \overline{B} = 1 \text{ and } C = 1\right)$

$$F = 0 \ \text{ se e solo se: } \ \overline{A}\,\overline{B}\,\overline{C} = 1 \text{ or } \overline{A}\,\overline{B}C = 1 \text{ or } \overline{A}BC = 1 \text{ or } A\overline{B}\,\overline{C} = 1 \text{ or } A\overline{B}C = 1$$

$$F = 0 \ \text{ se e solo se: } \ \left(\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C\right) = 1$$

$$\boxed{\overline{F} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C}$$

Seconda Forma Canonica: POS

$$\overline{F} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C$$

- Negando entrambi i membri
  ed applicando il II teorema di De Morgan si ottiene:

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

II Forma Canonica – PoS (Product of Sums):
Prodotto delle somme rappresentanti i **MAXtermini**

In generale:

$$\overline{F} = \sum_{i=1}^{W} M_i \quad , \quad W \le 2^N$$

$$\boxed{\overline{\overline{F}} = F = \left(\overline{\sum_{i=1}^{W} \overline{M_i}}\right) = (2^\circ \text{ Th. De Morgan}) = \prod_{i=1}^{W} M_i}$$

$$\overline{M_i} = a \cdot b \cdot c \longrightarrow M_i = \overline{a} + \overline{b} + \overline{c}$$

MAXtermine M_i: somma logica di tutte le variabili della funzione: $M_i = 0 \rightarrow F = 0$

Somma di Prodotti

Costruzione della II forma canonica:

❖ **Prodotto dei maxtermini**

$$\overline{M_i} = 0 \longrightarrow F = 0, \quad \forall i = 1..N$$

MAXtermine M_i : somma logica di tutte le variabili della funzione:

$M_i = 0 \rightarrow F = 0$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

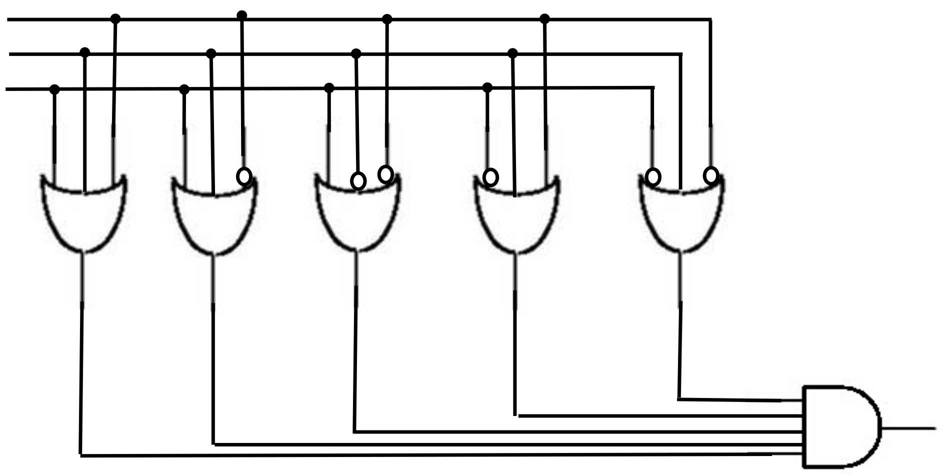
$$F =$$
$$= (A+B+C)\cdot$$
$$\cdot(A+B+\overline{C})\cdot$$
$$\cdot(A+\overline{B}+\overline{C})\cdot$$
$$\cdot(\overline{A}+B+C)\cdot$$
$$\cdot(\overline{A}+B+\overline{C})$$

Circuito 2ª forma canonica: POS

Circuito a **due stadi**:

1. Stadio **OR**: **W porte OR a n ingressi**, una per ogni MAXtermine
2. Stadio **AND**: **1 porta AND a W ingressi**

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



Forme canoniche – riepilogo

RIEPILOGO:

I Forma Canonica

- Mintermini m_i: i 2^N prodotti di tutte le variabili di F.
- Prendo i m_i per i quali $F(x,y,z)=1$ e li metto in **OR**

Minterms for Three Variables

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

II Forma Canonica

- Maxtermini M_i: le 2^N somme di tutte le variabili di F.
- Prendo tutti i M_i per i quai $F(x,y,z)=0$ e li metto in **AND**

Maxterms for Three Variables

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\overline{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\overline{Y}+\overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X}+Y+\overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X}+\overline{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	M_7	1	1	1	1	1	1	1	0

(da: Mano, Kime – *Reti logiche* – Pearson)

Circuiti logici – Aspetti implementativi

Le porte logiche sono realizzate nella pratica mediante dispositivi elettronici

Porte logiche reali ≠ operatori logici ideali

Aspetti di diversità:

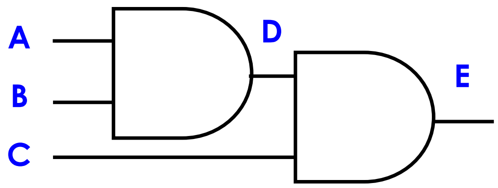
- **Velocità di commutazione (*progagation delay*)**
  - Le transizioni (0→1, 1→0) dell'uscita di una porta logica, a causa di una modifica degli ingressi, **non** sono istantanee, ma avvengono con un ritardo, detto tempo di propagazione della porta t_p
- **Limiti di pilotaggio di ingressi (*fan-out*)**
  - Un'uscita può essere collegata ad un numero limitato di ingressi.
  - Più ingressi sono collegati a una uscita, più lente saranno le transizioni 0→1 / 1→0.

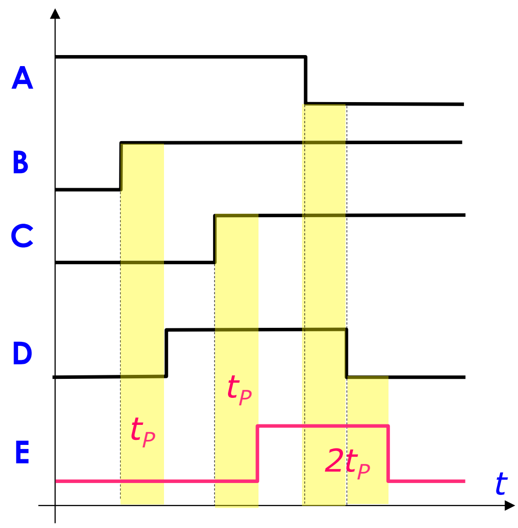
Velocità di un circuito

Velocità di commutazione:
ogni circuito logico è caratterizzato da un tempo di commutazione

CAMMINO CRITICO: **massimo** numero di porte da attraversare da un qualsiasi ingresso a una qualsiasi uscita

- Non si contano gli inverter (inclusi nelle porte)

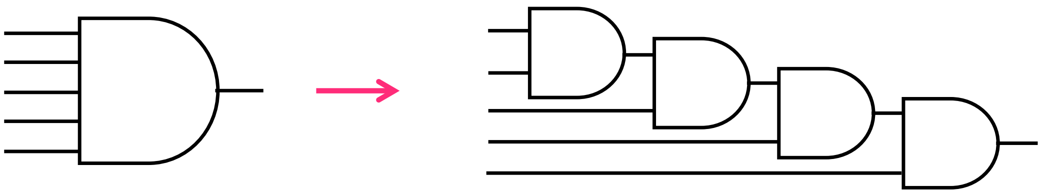




Implementazione con porte a 2 ingressi

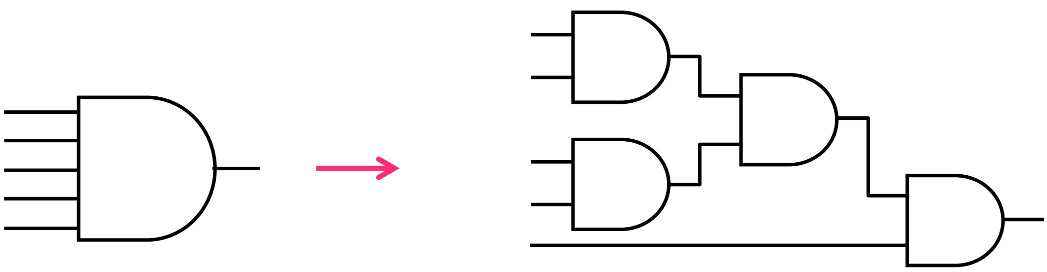
Spesso gli elementi costruttivi di base tipici sono porte a 2 ingressi

- **Porta a N ingressi → N–1 porte a 2 ingressi**



Porta a **N** ingressi → Cammino Critico: **N-1**

Ottimizzazione del cammino critico:



Porta a **N** ingressi → Cammino Critico: $\log_2(N)$

Criteri di valutazione delle prestazioni:

Semplicità (area)

- numero di porte in totale

Velocità (tempo di commutazione)

- numero di porte attraversate

Soddisfazione di **altri vincoli**

- potenza dissipata,
- facilità di debug...

A causa del cammino critico, un circuito più semplice è spesso (ma non sempre!) anche più veloce.

➔ ***SEMPLIFICAZIONE di circuiti***

Funzione logica di 3 variabili ➔ 3 ingressi, 1 uscita

1. Costruzione tabella di verità
2. Trasformazione a forma canonica (ad es. SOP)
3. Eventuale semplificazione

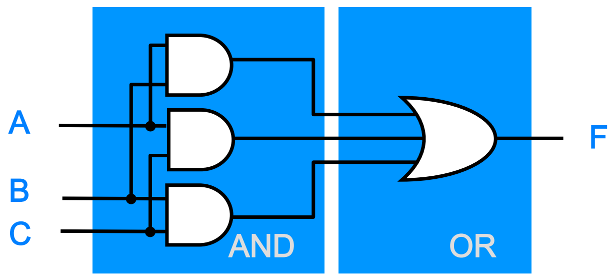
$$F(A,B,C) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC =$$

$$= \overline{A}BC + A\overline{B}C + AB\overline{C} + \underline{ABC + ABC + ABC} =$$

$$= AB(C+\overline{C}) + AC(B+\overline{B}) + BC(A+\overline{A}) =$$

$$= AB + AC + BC$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Esempio di semplificazione algebrica

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

- raccolgo: $B\cdot\bar{C}$

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

- inverso: $\bar{A}+A=1$

$$=1\cdot B\cdot\bar{C} + A\cdot B\cdot C =$$

- identità: $(1\cdot B = B)$

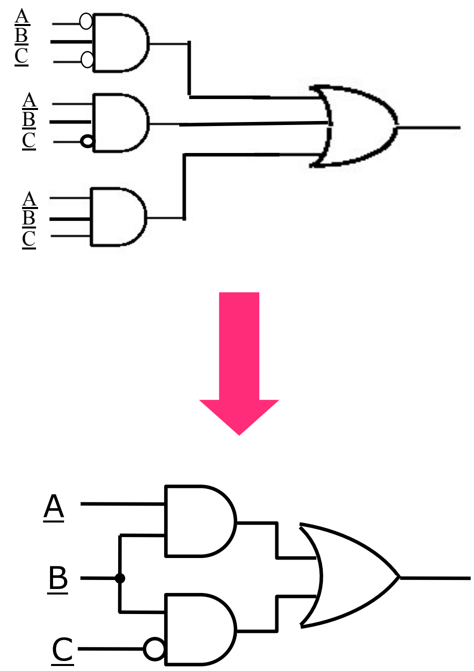
$$=B\cdot\bar{C} + A\cdot B\cdot C$$

- raccolgo: B

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

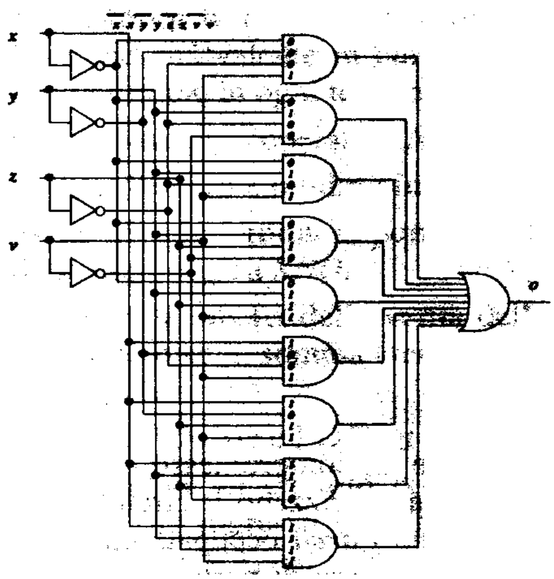
- II legge assorb.: $(A+\bar{A}B = A+B)$

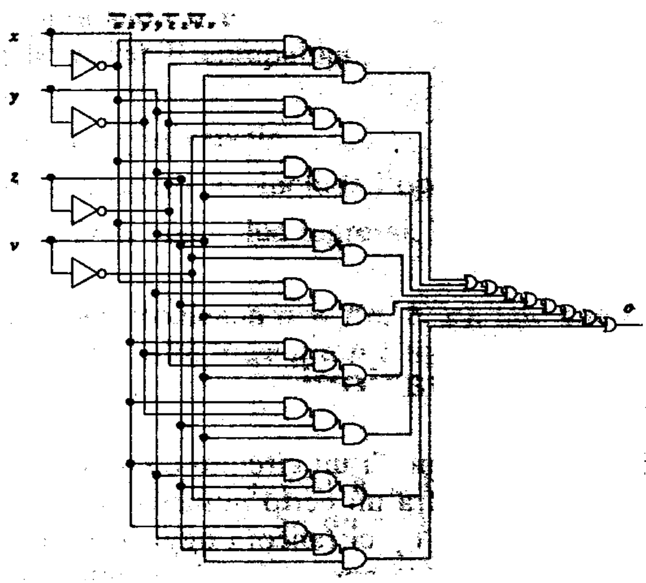
$$=B\cdot\left(\bar{C}+A\right) = AB + B\bar{C}$$



Riduzione a circuiti con porte a 2 ingressi

$$O = \bar{x}\bar{y}\bar{z}\bar{v} + \boxed{\bar{x}y\bar{z}\bar{v} + \bar{x}y\bar{z}v} + \bar{x}yz\bar{v} + \bar{x}yzv + \bar{x}\bar{y}zv + \bar{x}yzv + xyz\bar{v} + xyzv$$





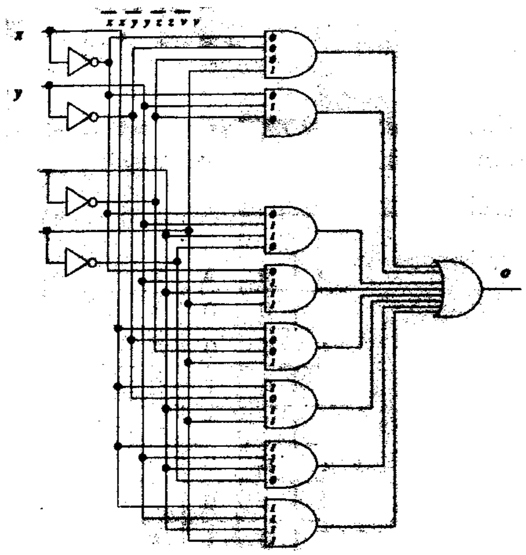
Cammino critico = 2 , N. porte = 10

Cammino critico = 11 , N. porte = 35

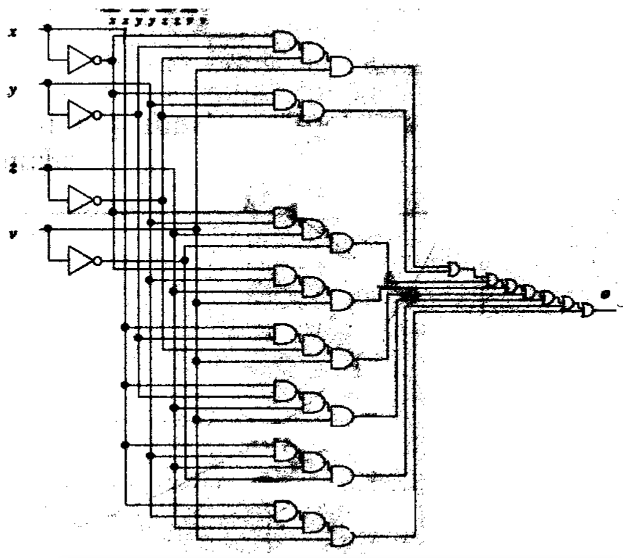
Semplificazione

- Semplificando la prima parte dell'espressione logica...

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



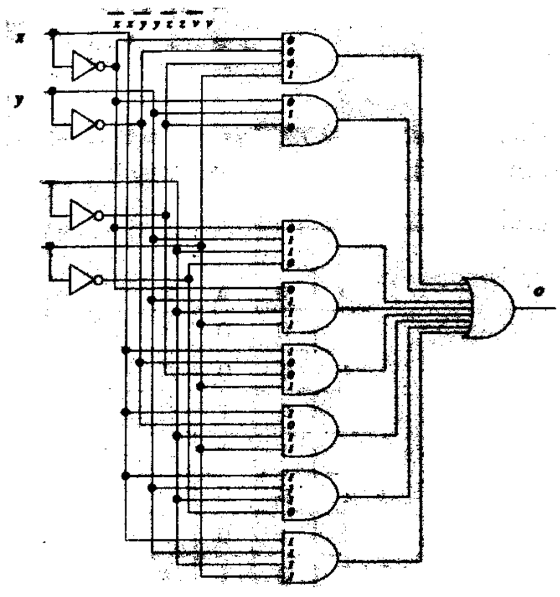
Cammino critico = 2 N. porte = 9

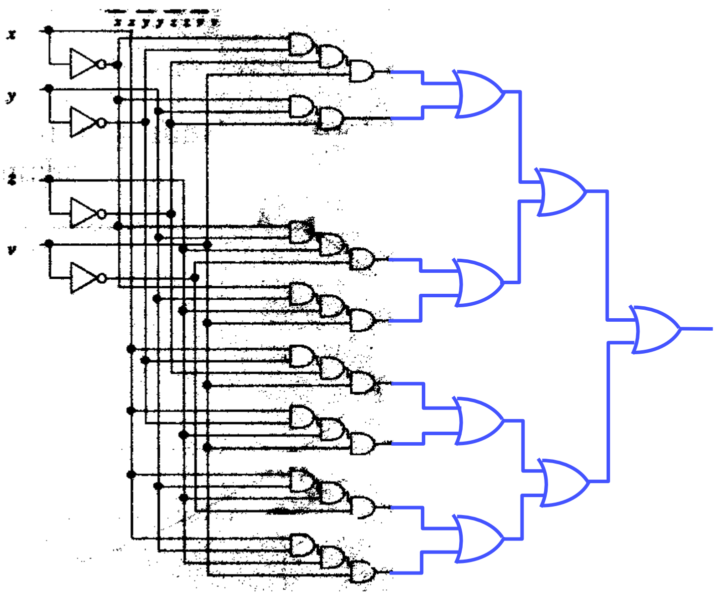


Cammino critico = 10 N. porte = 30

Ottimizzazione del cammino critico

- Collegando le porte in modo ottimizzato, si riduce significativamente il cammino critico...





Cammino critico = 6 N. porte = 30

Semplificazione

$$O = \bar{x}\bar{y}\bar{z}\bar{v} + \bar{x}y\bar{z}\bar{v} + \bar{x}y\bar{z}v + \bar{x}yz\bar{v} + \bar{x}yzv + \bar{x}\bar{y}zv + \bar{x}yzv + xyz\bar{v} + xyzv$$

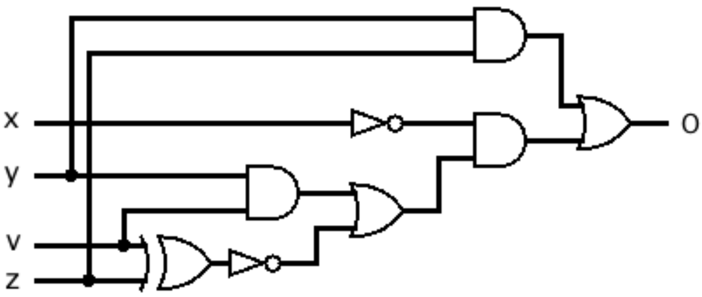
$$= \bar{x}\bar{z}\bar{v}\left(y + \bar{y}\right) + \bar{x}y\bar{z}v + \bar{x}yz\left(\bar{v} + v\right) + \bar{x}zv\left(\bar{y} + y\right) + xyz\left(\bar{v} + v\right) =$$

$$= \bar{x}\left(\bar{z}\bar{v} + zv + y\bar{z}v\right) + yz = \bar{x}\left(\bar{z}\bar{v} + v\left(z + \bar{z}y\right)\right) + yz =$$

$$= \bar{x}\left(\bar{z}\bar{v} + v\left(z + y\right)\right) + yz =$$

$$= \bar{x}\left(\bar{z}\bar{v} + vz + vy\right) + yz =$$

$$= \bar{x}\left(\overline{\left(v \oplus z\right)} + vy\right) + yz$$



Cammino critico = 5 N. porte = 8

Semplificazione: mappe di Karnaugh

Mappe di Karnaugh: tecnica di semplificazione di espressione (circuito) **a partire dalla tabella di verità**

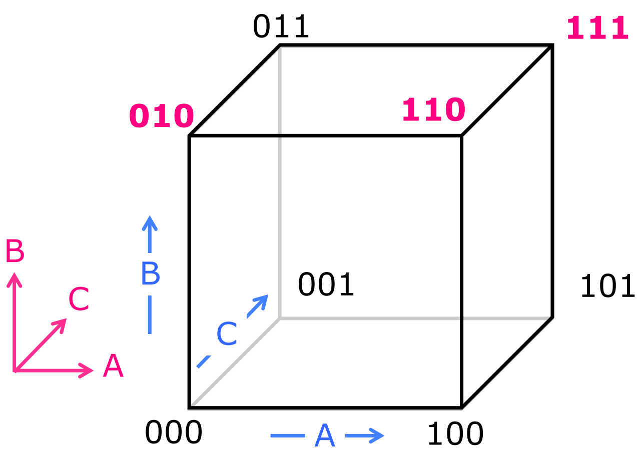
Consideriamo una funzione di **3 variabili**

- Rappresentazione cubica di funzioni logiche a 3 variabili: F = *f*(a,b,c)
- Muovendosi sugli spigoli, la configurazione di variabili cambia di un solo bit
- **Distanza di HAMMING: *d(v1, v2)* = n. di bit diversi tra le sequenze**

$Esempio: \quad F = A \cdot B + B \cdot \overline{C}$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

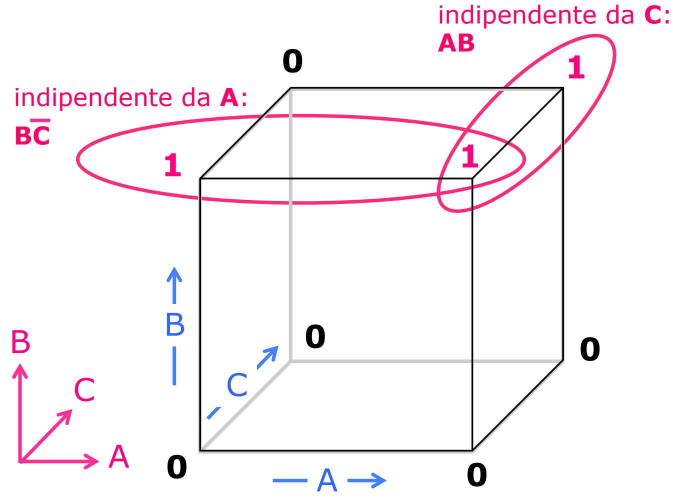
rappres. cubica



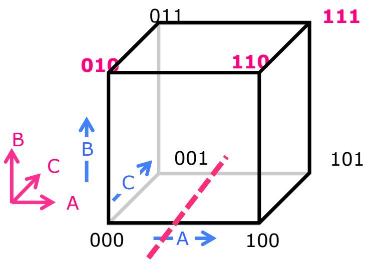
Copertura: ricerca di tutti gli implicanti

- Se i vertici di un lato sono entrambi 1, essi insieme rappresentano un implicante
- L'implicante è indipendente dalla variabile corrispondente al lato

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	**1**
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	**1**
1	1	1	**1**



- Rappresentazione **piana** della funzione:
  - "srotolo" il cubo
  - codifica di Gray (codice riflesso) lungo ogni direzione



AB / C	00	01	11	10
0	000	010	110	100
1	001	011	111	101

indipendente da A: **B$\overline{C}$**

indipendente da C: **AB**

AB / C	00	01	11	10
0	0	1	1	0
1	0	0	1	0

F = AB + B$\overline{C}$

Semplificazione: mappe di Karnaugh

Per **N>3** variabili, la rappresentazione diviene complessa...

- Rappresentazione piana utilizzabile per: **2 ≤ N ≤ 4**

Codifica di **Gray**:
config. **adiacenti** hanno **d_H=1**

B \ A	0	1
0	1	0
1	1	0

$F = \overline{A}$

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	1	1	1	1
10	0	0	1	0

$F = AB + B\overline{C}\,\overline{D} + CD$

Semplificazione: mappe di Karnaugh

Mappa di Karnaugh: rappresentazione piana e ciclica

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

mappa Karnaugh

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	1	0	1	1
10	0	1	1	0

$F = AB + B\overline{D} + \overline{B}CD$

Uscite indifferenti di una funzione logica

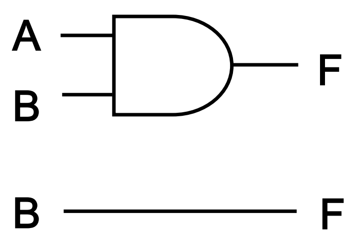
Situazione tipica in sintesi (progetto) di funzioni/circuiti logici:

- Per alcune combinazioni degli ingressi,
  il valore assunto dall'uscita è INDIFFERENTE
- Si parla in tal caso di:
  funzione logica (o tabella di verità) ***non completamente specificata***
  - **Simbolo: X**
- Come si risolve?
  - **Si sceglie il caso che rende il circuito più semplice**

A	B	F
0	0	**0**
0	1	**X**
1	0	**0**
1	1	**1**

X=0 → F=AB

X=1 → F=B



Esercizi

Da un tema d'esame:

Si progetti un circuito caratterizzato da un ingresso a 4 bit rappresentante un numero binario intero senza segno A, e un'uscita che vale '1' se e solo se:

(A<4 ed è divisibile per 2) oppure
(4≤A<8) oppure
(A≥8 ed è divisibile per 4).

a) Determinare la tabella di verità della funzione logica di uscita;
b) scrivere la funzione nella forma canonica più adatta;
c) semplificarla mediante mappa di Karnaugh.

Generatore di parità dispari su 3 bit:

Si progetti un circuito caratterizzato da 3 ingressi (a,b,c) e da un'uscita P tale che:

P = 1 se e solo se il n. di "1" sugli ingressi è dispari

a) Determinare la tabella di verità della funzione logica di uscita;
b) semplificarla mediante mappa di Karnaugh;
c) semplificarla ulteriormente, se possibile, mediante trasformazioni algebriche;
d) disegnarne il corrispondente circuito digitale.

Progetto con tabella di verità non completamente specificata:

Si progetti un circuito caratterizzato da 4 linee di ingresso A,B,C,D, e un'uscita Y che vale '1' se e solo se su almeno 2 delle 4 linee è presente un '1'. Si supponga che, per vincoli circuitali, al più 2 linee di ingresso possano valere '1'.

Determinare ... a) b) c) d).