

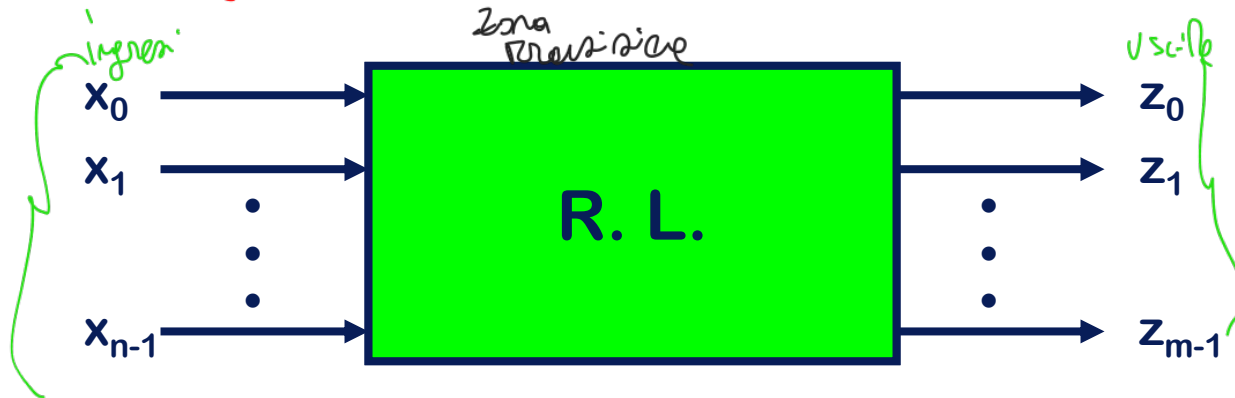
Lezione 2: Dalle Porte Elementari alle Reti Logiche

- Una Rete Logica è tipicamente un sistema elettronico: *è una scatola con un certo numero di ingressi e uscite*
 - **Unidirezionale**, cioè in cui i segnali viaggiano dagli ingressi verso le uscite e non viceversa (ovvero il sistema non è reversibile)
 - I cui segnali di ingresso e di uscita sono digitali, cioè ogni filo assume valori zero, uno, indeterminato o alta-impedenza *X = segnale inverter, Z = dipende alta impedenza, filo scollegato*
valori in un momento di tempo 0 o 1
 - **Descrivibile con equazioni dell'algebra booleana**, se si assume una situazione ideale (in cui i valori dei segnali sono solo zero o uno)

DIRET: ingresso $\rightarrow$ uscita

Una rete logica è descrivibile con eq. booleane.

Segnali sono DIGITALI, valgono 0 e 1, X e Z



Reti si dividono in due classi fondamentali:

Reti Combinatorie e Reti Sequenziali

RETI LOGICHE POSSONO ESSERE

- **Reti COMBINATORIE** (RC o CN == Combinatorial Network) *rete logica in cui l'uscita dipende solo dall'ingresso*
 - In qualunque istante *Come prima* le uscite sono funzione del valore che gli ingressi hanno in quell'istante
 - Il comportamento (uscite in funzione degli ingressi) *rete combinatorio descritto completamente da una tabella I/O* è descritto da una tabella
 - NON CONTENGONO ANELLI DI RICHIUSURA *RETE COMBINATORIO, NO Anelli di richiusura*



• Reti SEQUENZIALI

- In un determinato istante *quello provoca un effetto memoria, uscita dipende da ingressi e uno stato interno che ne descrivono lo stato precedente* le uscite sono funzione del valore che gli ingressi hanno in quell'istante e dei valori che gli ingressi hanno assunto precedentemente
- Presentano Stati Interni (sono quindi reti dotate di MEMORIA)
→ La descrizione è più complessa di una semplice tabella

PRESENTANO ANELLI DI RICHIUSURA

NON SI DESCRIVONO CON UNA TABELLA

RETI Seq.

LE RAPPRESENTO CON MACCHINE A STATO E MODE

Porte Logiche Base

Rete logica più semplice, una porta logica

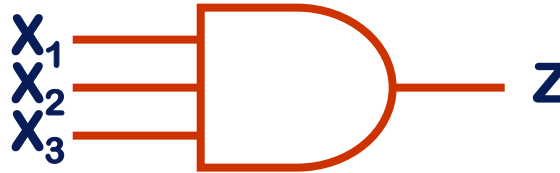
• Rappresentazione circuitale delle funzioni logiche

oggetto fisico ben definito

• AND o prodotto

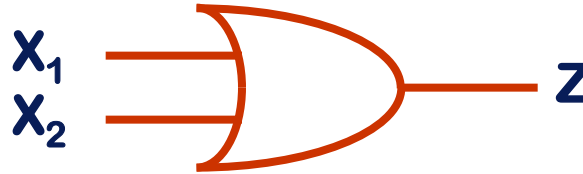
$$Z = X_1 \cdot X_2 \cdot X_3$$

AND tra tre variabili; AND è un prodotto



• OR o somma *cf. è una somma*

$$Z = X_1 + X_2$$



• NOT o negazione

$$Z = \bar{X}$$

possibili notazione per NOT

$\bar{X}$ /X X_{-} $X_{\neg}$ LOW



• Con le sole porte AND, OR, NOT si può realizzare una Rete Logica qualsiasi

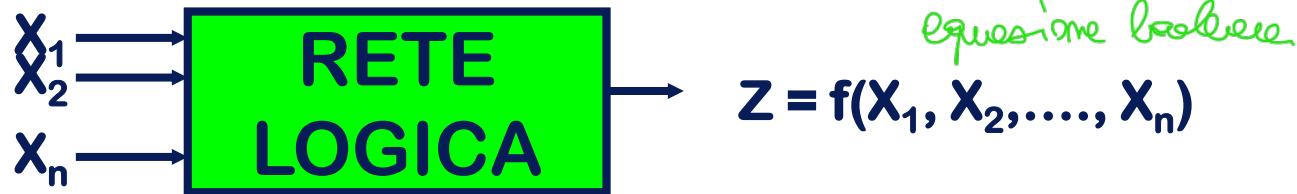
- Verrà dimostrato nel seguito riducendo una rete logica arbitraria a forme «standard» (cf. "somma di prodotti" o "prodotto di somme")

possibile realizzare qualunque espressione booleana

Esempio di rete logica

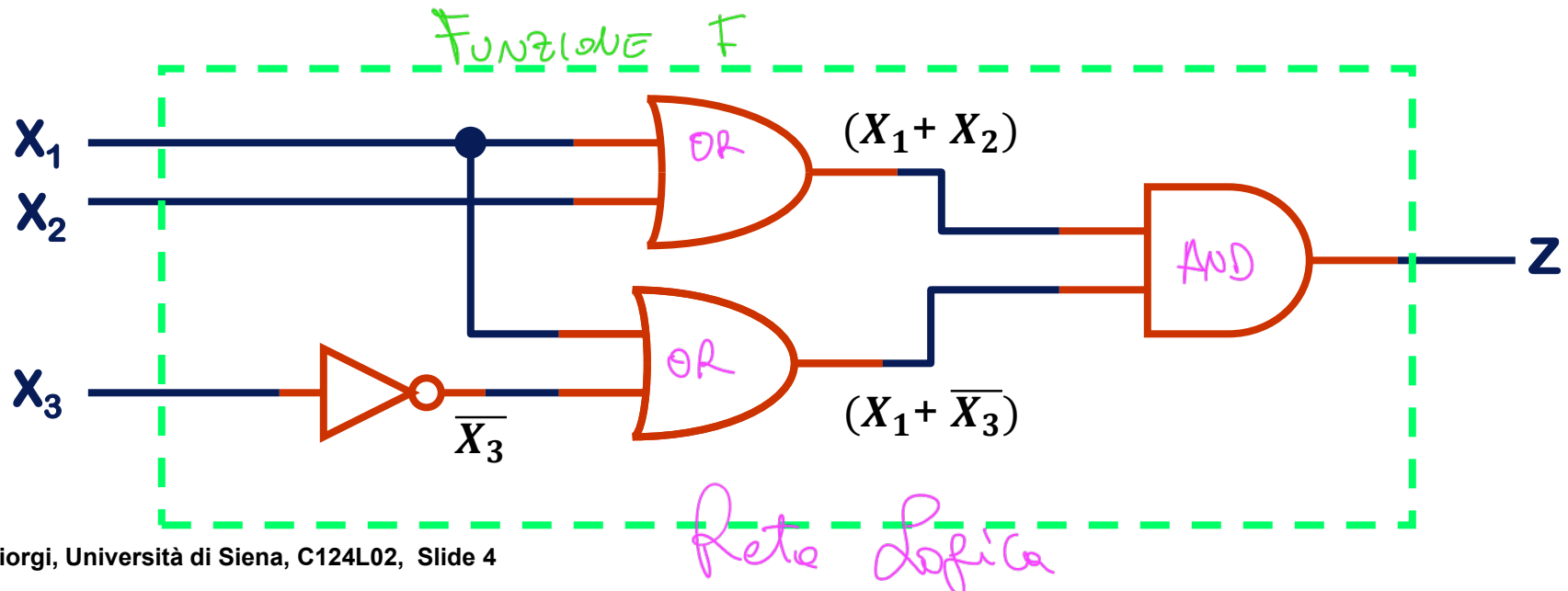
rete logica, è la $\bar{f}$

- Data una RETE LOGICA, possiamo descrivere la relazione fra ingresso X e uscita Z tramite una funzione 'f'



- Ad esempio, potremmo avere: $Z = f(X_1, X_2, \dots, X_n) = (X_1 + X_2) \cdot (X_1 + \bar{X}_3)$

equazione Bool, allora anche definito il circuito fisico
scrivendo f , ho anche determinato il circuito



Porta NAND e NOR

Cascata AND con porta NOT = NAND

oppure Cascata NOT e OR ottengo NOR

• NAND

representa
tutta verità

X_1	X_2	Z
0	0	1
0	1	1
1	0	1
1	1	0

l'uscita è 0 $\Leftrightarrow$ entrambi
input
1

Relazione tra NOT



$$Z = \overline{X_1 \cdot X_2}$$

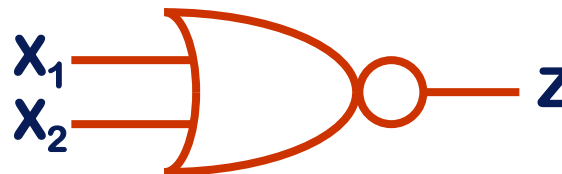


• NOR

X_1	X_2	Z
0	0	1
0	1	0
1	0	0
1	1	0

NOR
uscita
1 $\Leftrightarrow$ entrambi
input sono 0

$$Z = \overline{X_1 + X_2}$$



4 transistor, ha una sua vapp. circuibile

Proprietà della porta **NAND** (NOR)

NAND posso farlo con 4 transistor

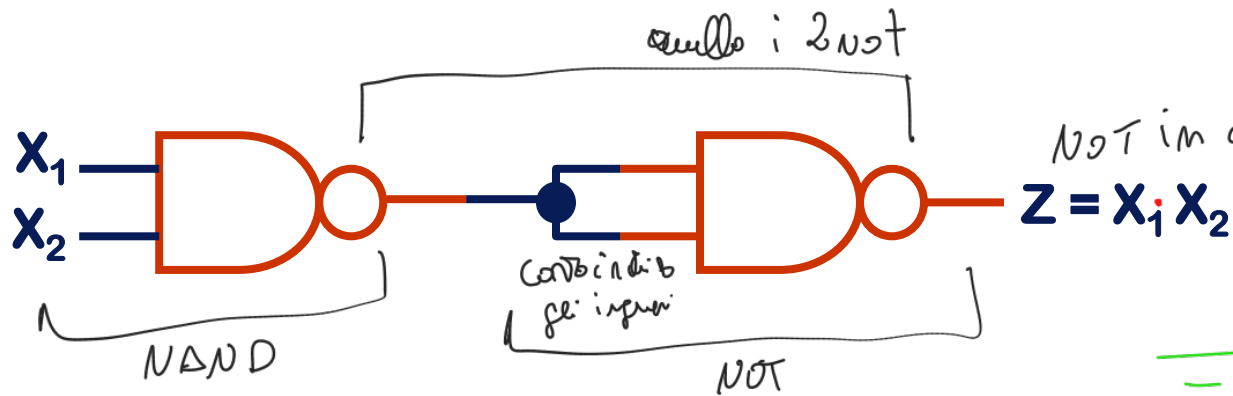
- Utilizzando solamente porte solo NAND (o solo NOR) è possibile realizzare qualunque rete logica; infatti

• NOT



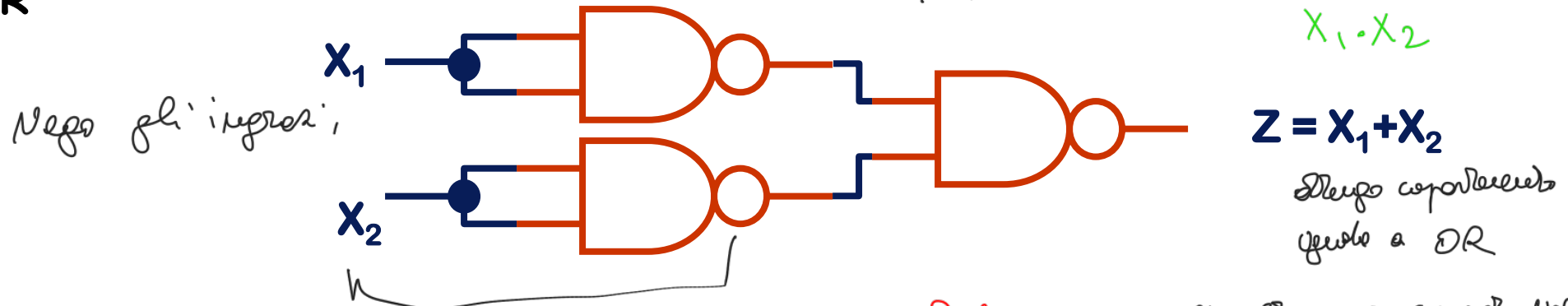
Se entrai gli ingressi da 1, $\Rightarrow 0$
... lo invertito

• AND



NOT in cascata = AND

• OR



$$\overline{\overline{X_1}} \cdot \overline{\overline{X_2}}$$

OR Esclusivo (XOR)

date replicate notand

Exclusive OR

• Realizzazione dell'OR Esclusivo

X_1	X_2	Z
0	0	0
0	1	1
1	0	1
1	1	0

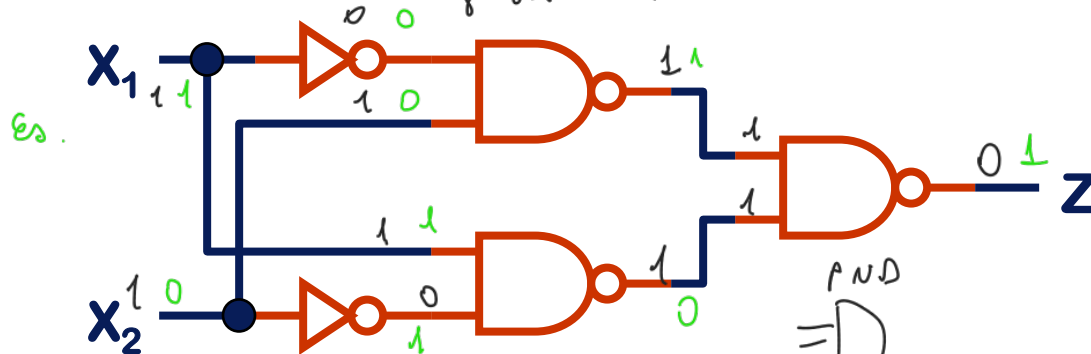
l'uscita vale 1 se gli ingressi sono diversi, 0 se sono uguali, utile es. NEL COMPARATORE

$$Z = X_1 \oplus X_2 = \overline{X_1}X_2 + X_1\overline{X_2} = (X_1 + \overline{X_2})(\overline{X_1} + X_2) = \overline{(\overline{X_1}X_2)}(\overline{X_1\overline{X_2}})$$

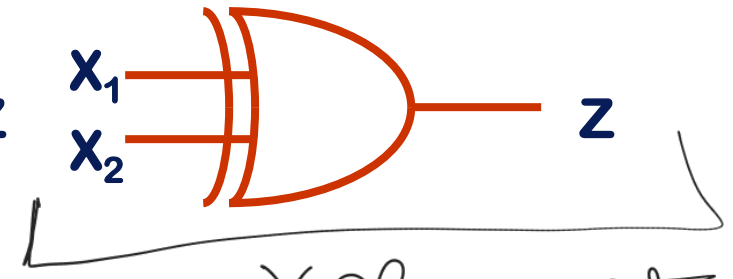
or esclusivo

Realizzazione in due porte

ingressi uguali => uscita 0
ingressi diversi => uscita 1



Simbolo OR con arcobaleno



XOR, comparatore

OR Esclusivo



AND
=> OR
=> XOR
=> NOR
=> NAND

Algebra della Logica

2 soli valori NO 7 0 X

- **George Boole**

- Matematico inglese

(1815 - 1864)

- **Algebra della Logica, Algebra di Boole, Algebra Booleana**

- Sistema matematico formale che descrive funzioni logiche

- Funzioni e variabili che possono assumere solo due valori

- Falso

Vero

- 0 (logico)

1 (logico)

- Livello logico Basso

Livello logico Alto

- 0 V

5 V

Modi eq. per scrivere sistemi digitali

- **Sistema matematico formale**

- insieme di elementi

- insieme di operazioni

- insieme di postulati

- TEOREMI

Algebra il cui dominio è
costituito da 2 valori
0, 1 / Vero, Falso

- **Perché è importante:**

- Nel 1938, Shannon dimostrò che l'Algebra booleana può descrivere le operazioni di circuiti elettrici che commutano fra due valori

La uso per descrivere sistemi digitali

Tabella di Verità

Retta logica, derivata da un
funzione logica, relas. tra in e out

- Una funzione logica può sempre essere espressa da una tabella detta:

TABELLA DI VERITÀ (TRUTH TABLE)

1° parte: Retta per i l'i' calcol. variab.
2° parte: VScolla

Osservazioni

- Una funzione logica di "n" variabili ammette 2^n possibili valori
 - Tale funzione 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. Tale tabella è detta **tabella di verità**
- Esempio (funzione di tre variabili) $z = f(x_1, x_2, x_3)$

Ingresso Tabella,
elenco TUTTI i possibili
ingressi
Tutte i possibili
n variab., 2^n righe

ingressi			uscita
x_1	x_2	x_3	z
0	0	0	$f(0,0,0)$
0	0	1	$f(0,0,1)$
0	1	0	$f(0,1,0)$
0	1	1	$f(0,1,1)$
1	0	0	$f(1,0,0)$
1	0	1	$f(1,0,1)$
1	1	0	$f(1,1,0)$
1	1	1	$f(1,1,1)$

uscita, funzione
booleana valutata
in questo
ingresso

0 e 1

Postulati di HUNTINGTON

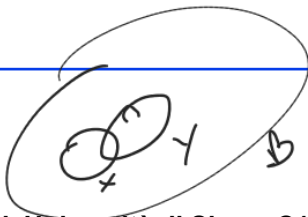
Ho solo 2 ops.

Proprietà posso esprimere per somma o prodotto

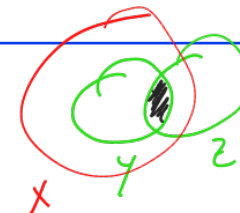
Qualità

Qualità principi, Somma o Prodotto

Postulato	Espressione primale (rispetto alla somma)	Espressione duale (rispetto al prodotto)
Esistenza di almeno due elementi distinti	$\exists x, y \in B: x \neq y$	
Esistenza degli elementi identità <i>unione insieme con insieme vuoto</i>	$x + 0 = x$ <i>el. identità</i>	$x \cdot 1 = x$ <i>ident. prodotto</i> <i>interos. infimo con insieme universo</i>
Proprietà commutativa	$x + y = y + x$	$x \cdot y = y \cdot x$ <i>NOVITA' distributivo rispetto alla somma</i>
Proprietà distributiva	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$	$x + (y \cdot z) = (x + y) \cdot (x + z)$
Esistenza del complemento <i>unione tra insieme e complemento</i>	$x + \bar{x} = 1$ <i>ness. capo insi.</i>	$x \cdot \bar{x} = 0$ <i>interos. tra 2 insiemi</i>
Chiusura	$\forall x, y \in B, (x + y) \in B$	$\forall x, y \in B, (x \cdot y) \in B$



$x + \bar{x} = 1$, l'insieme



interos. tra $x \in y \cap z$

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 nell'algebra tradizionale:
 - Proprietà distributiva della somma logica
- L'operazione complemento logico esiste solo nell'algebra booleana
- La sottrazione e la divisione non esistono nell'algebra booleana
- Vale un PRINCIPIO DI DUALITÀ *qui invece Booleana può essere trasformata nella duale*
 - I postulati duali si ottengono da quelli primali operando le seguenti sostituzioni:

- { Scambiando i due operatori binari fra loro, $(+)$ con $(\cdot)$ e $(\cdot)$ con $(+)$
 - E
 - { Scambiando fra loro i due elementi identità, 1 con 0 e 0 con 1

Duali
Posso trasformare
nella duale
qual'ora esista

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 $n=0$ (nel caso booleano sempre) e assumendo vera l'ipotesi con k variabili risulta vera anche per $k+1$ variabili allora è vera per qualunque n)

• Dimostrazione per induzione perfetta

- 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 in ingresso
- Tale metodo prende il nome di **Metodo dell'INDUZIONE PERFETTA**

Tutte le possibili ipotesi sono veri ripete a quella che vale di m, allora ...

Vera perché verifico la tesi con tutte le possibili ipotesi

Tutte possibili combinazioni di ingresso

TEOREMI

TEOREMA	Espressione primale (rispetto alla somma)	Espressione duale (rispetto al prodotto)
Annullamento	<i>intero + parte = intero</i> $x + 1 = 1$	$x \cdot 0 = 0$
Involuzione		$\overline{(\bar{x})} = x$
Idempotenza	$x + x = x$	$x \cdot x = x$
Assorbimento	$x + xy = x$	$x(x + y) = x$
Somma con il complemento	$x + \bar{x}y = x + y$	$x(\bar{x} + y) = xy$
Associatività	$x + (y + z) = (x + y) + z$ $= x + y + z$	$x(yz) = (xy)z$ $= xyz$
Teorema del consenso	$xy + \bar{x}z + yz$ $= xy + \bar{x}z$ <i>intersezione comune non serve</i>	$(x + y)(\bar{x} + z)(y + z)$ $= (x + y)(\bar{x} + z)$
Teorema di De Morgan	$\overline{(x + y)} = \bar{x} \cdot \bar{y}$	$\overline{(x \cdot y)} = \bar{x} + \bar{y}$

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

$$\begin{array}{c|c|c} x & y & x+y \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{array} \quad \begin{array}{c|c|c} \bar{x} & \bar{y} & \bar{x} \cdot \bar{y} \\ \hline 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}$$

Esempio di dimostrazione

- Teorema di De Morgan *Dim. indus. Perfetta*

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

Wolfe pos. coerenza

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

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.

Teorema del Consenso

$$xy + /xz + yz = xy + /xz$$

x	y	z	$/x$	xy	$/xz$	yz	$xy + /xz + yz$	$xy + /xz$
0	0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	1	1
0	1	0	1	0	0	0	0	0
0	1	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1	1
1	1	1	0	1	0	1	1	1

c.v.d.

Dalle tabelle di verità alla sintesi della rete logica

- Una funzione booleana può sempre essere definita tramite **tabella di verità**

• Esempio:

a	b	c	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

- Posso scrivere, data tabella, l'espressione bool
 corrisp. con mintermi o maxtermi
- È possibile derivare una espressione algebrica a partire dalla tabella di verità in (almeno) due modi:

$$0 \quad 1 = 1$$

$$\bar{a} \cdot \bar{b} \cdot \bar{c}$$

- Identificando tutte le righe che producono un 1 e sommando (OR) i corrispondenti termini (prodotti ovvero AND di variabili) che producono tale 1. Tali «termini prodotto» prendono il nome di **mintermini**. L'espressione risultante è detta **somma di prodotti**. Esempio:

$$z = \underbrace{\bar{a} \cdot \bar{b} \cdot \bar{c}}_{1^{\circ} \Delta} + \underbrace{\bar{a} \cdot b \cdot \bar{c}}_{2^{\circ} \Delta} + \underbrace{a \cdot \bar{b} \cdot \bar{c}}_{3^{\circ} \Delta} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$

Somma di prodotti

- Identificando tutte le righe che producono uno 0 e facendo il prodotto (AND) dei corrispondenti termini (somme ovvero OR di variabili) che producono tale 0. Tali «termini somma» prendono il nome di **maxtermini**. L'espressione risultante è detta **prodotto di somme**. Esempio:

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

Prodotto di somme

0 | 0 | 1 = 0

e i termini li moltiplico

Somma, Scrivo eq corrisp. alla Tabella verità

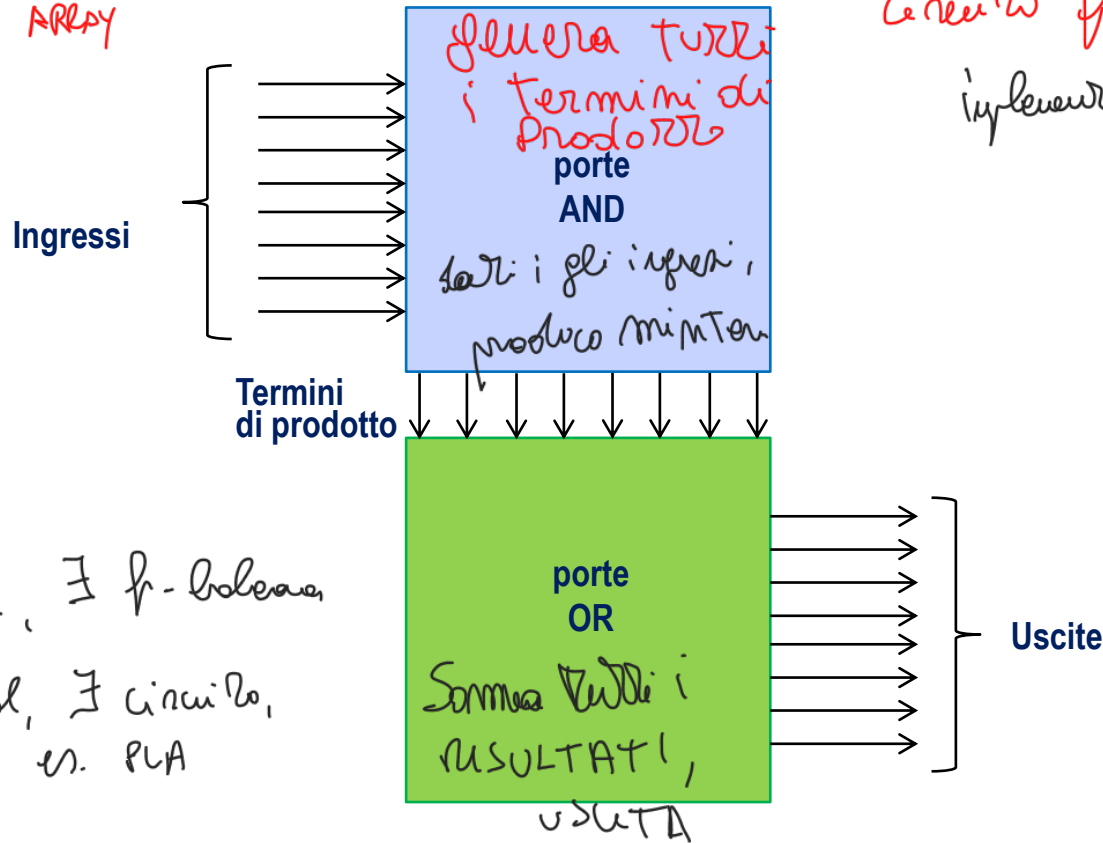
Tabella, posso implementare fisicamente, Vh. Bool.

 C'è l'implementazione fisica e possibile

Programmable Logic Array - PLA

- Dato che si può sempre rappresentare una funzione logica come tabella di verità, si può sempre rappresentare anche come somma di prodotti (o prodotto di somme)
- Questo conduce ad una tecnica di implementazione universale nota come PLA

Programmable Logic Array



circuito fisico modo per implementare fisicamente la funzione booleana tramite somme prodotti o prodotto somme

Le due matrici AND e OR possono essere anche rese "programmabili" (vedere lezioni successive)

dato Tabella, $\exists$ f. booleana
 dato f. Bool, $\exists$ circuito, es. PLA

se lo si allunga, allora i mintermi ecc.

Mappa di Karnaugh *Karnaugh* *strutturato di quello che abbiamo visto*

- Può essere comodo semplificare l'espressione booleana

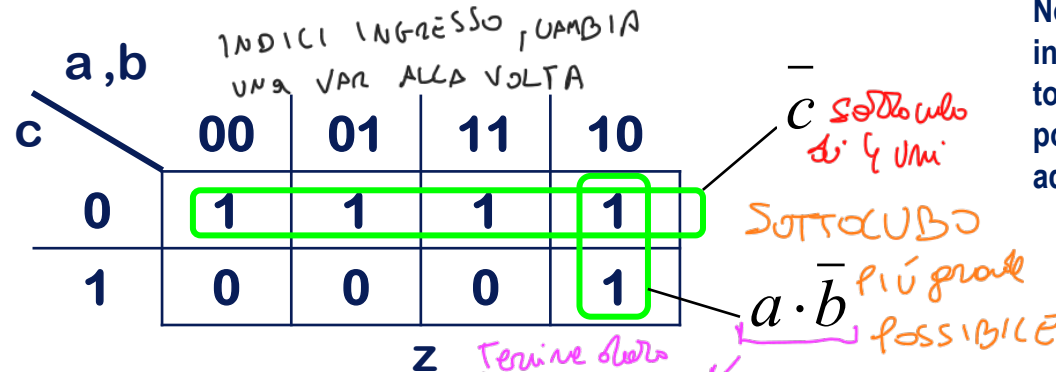
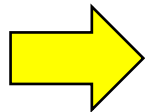
- Per ottenere espressioni booleane più semplici e facili da verificare *Stema EQ in MANIERA PIÙ semplice*
- Per sintetizzare una rete logica con un minor numero di porte (minor area)

- La Mappa di Karnaugh è una rappresentazione alternativa della tabella di verità che, sfruttando la proprietà di assorbimento*, permette rapidamente di identificare termini più semplici

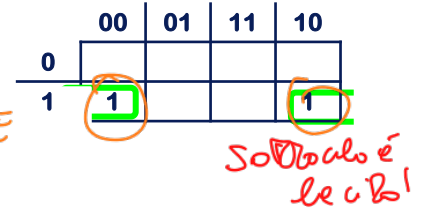
- La Mappa di Karnaugh si ottiene:

- disponendo gli "indici" generati dalle variabili di ingresso in modo che due indici adiacenti differiscano per una sola cifra. Esempio:

a	b	c	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



Nota: la mappa si svolge in realtà su una superficie toroidale → I sottocubi possono continuare su celle adiacenti la parte opposta



- Localizzando i gruppi adiacenti di 1 con un numero di 1 pari a una potenza di 2 (detti **sottocubi di ordine k** se contengono 2^k uni)

- Un termine è identificato dagli indici che rimangono "fissi" nel sottocubo. Esempio:

vedo cosa varia e cosa resta fisso

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

interseco xy e $\bar{x}y$ e sommo...

da 2 termini ne ottengo 1 solo

*** $xy + \bar{x}y = y$**

deno minore sottocubo più grande possibile

Mappe di Karnaugh a 2, 3, 4, 5 variabili

	0	1
0	1	1
1		

2 variabili

	00	01	11	10
0				
1		1	1	

3 variabili

	00	01	11	10
00				
01		1	1	
11		1	1	
10				

4 variabili
1 uscita

	00	01	11	10
00	1			
01				
11				
10				

5 variabili (richiede una visione "multidimensionale")

*2 tabelle da 4, e lo da 16 sono
adiacenti nella 3ª dim*

Note sulle Mappe di Karnaugh

- Se la mappa contiene dei ^{può andar bene sia 0 che 1} non-specificato (X o -)
 - Si può interpretare come è più conveniente per ottenere sottocubi più grandi
- Se ci sono più possibilità con sottocubi più piccoli e più grandi
 - Si preferiscono sempre coperture con sottocubi più grandi (hanno meno ingressi)
- Si prendono solo i termini strettamente necessari per ottenere una lista di copertura "a costo minimo"
 - Esiste molta letteratura in merito alla ottimizzazione delle liste a costo minimo (ma non fa parte di questo corso)
- Esercizio:
 - Scaricare la app per Android: "Karnaugh Kmap Solver" (simili esistono anche per iPhone)
 - Provare a verificare la sintesi con alcune funzioni logiche usando tale app

Karnaugh, M. "A Map method for Synthesis of Combinational Logic", Transactions of AIEE, Communication and Electronics, 72, part I, Nov. 1953, pp. 593-99