DEI
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sistemi Digitali

Porte logiche
Algebra di Boole

Marta Bagatin, marta.bagatin@unipd.it

Corso di Laurea in Ingegneria dell'Informazione

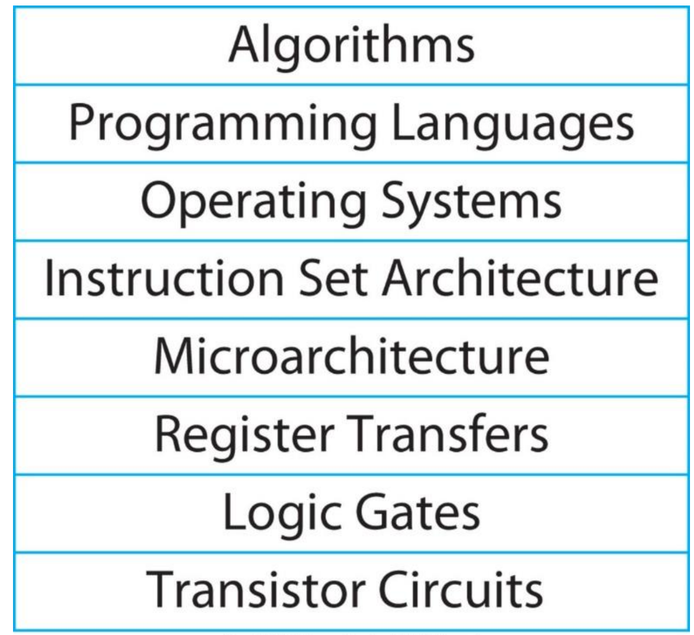
Anno accademico 2022-2023

Scopo della lezione

- Studiare le **porte logiche** e come usarle per realizzare circuiti digitali con determinate relazioni tra ingressi e uscite
- Introdurre l'**algebra booleana** e identificare le regole per manipolare le funzioni logiche ai fini di renderle adatte alla progettazione di un circuito digitale

Circuiti digitali e porte logiche

- I **circuiti digitali** sono componenti hardware che processano informazioni binarie
- I blocchi di base di questi circuiti sono le **porte logiche** (logic gates)
- Una porta logica ha il compito di **svolgere determinate operazioni logiche tra i suoi ingressi e produrre un'uscita**



A loro volta, le porte logiche sono realizzate con un insieme di transistor e interconnessioni

Logica binaria

- Le porte logiche operano in **logica binaria** (= logica booleana), cioè con variabili che possono assumere solo 2 valori discreti, per convenzione ‘**0**’ e ‘**1**’
- Le **tre operazioni fondamentali della logica binaria** sono **AND**, **OR**, **NOT**: tutte le espressioni booleane sono esprimibili in termini di queste operazioni logiche
- Descriveremo una funzione logica tramite la sua **tabella di verità** che elenca i valori dell’uscita di tale funzione in corrispondenza a tutte le possibili combinazioni dei suoi ingressi

Funzione logica AND

- Funzione con due o più variabili in ingresso
- Notazioni: $Z = X \cdot Y$ $Z = X\,Y$ $Z = X \wedge Y$ Z = X **AND** Y
- <u>Z = 1 se e solo se X = 1 e Y = 1</u>

  <u>altrimenti Z = 0</u>
- **Tabella di verità**

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

Funzione logica OR

- Funzione con due o più variabili in ingresso
- Notazioni: Z = X **+** Y   Z = X **V** Y   Z = X **OR** Y
- Z = 1 se almeno uno tra X e Y è 1

  altrimenti Z = 0
- **Tabella di verità**

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

Funzione logica NOT

- Funzione con una variabile in ingresso, è anche detta funzione complemento (nega la variabile in ingresso)
- Notazioni: $Z = \overline{X}$ $Z = \sim X$ Z = **NOT** X
- Z = 1 se X = 0

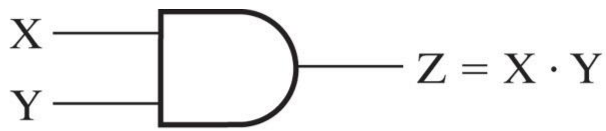
  Z = 0 se X = 1
- **Tabella di verità**

X	NOT X
0	1
1	0

Porte logiche

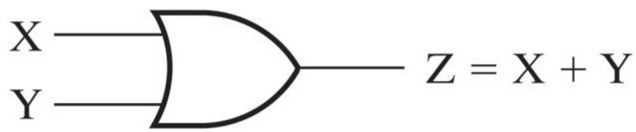
- Le porte logiche AND, OR, NOT sono circuiti che producono in uscita il valore previsto dalla rispettiva tabella di verità

- Porta **AND**



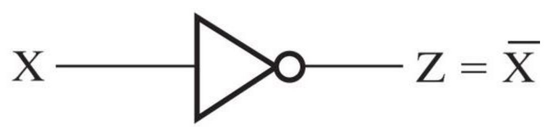
AND gate

- Porta **OR**



OR gate

- Porta **NOT** (inverter)



NOT gate or inverter

Porte logiche

AND

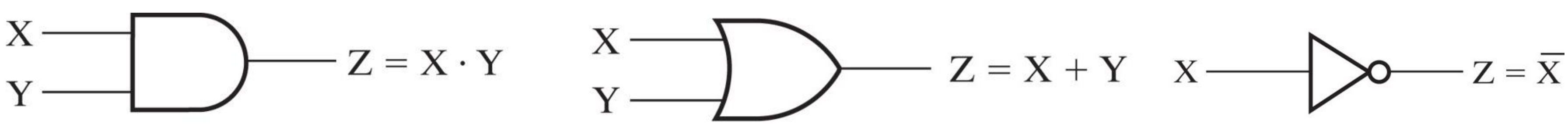
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

X	Y	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

X	$Z = \overline{X}$
0	1
1	0



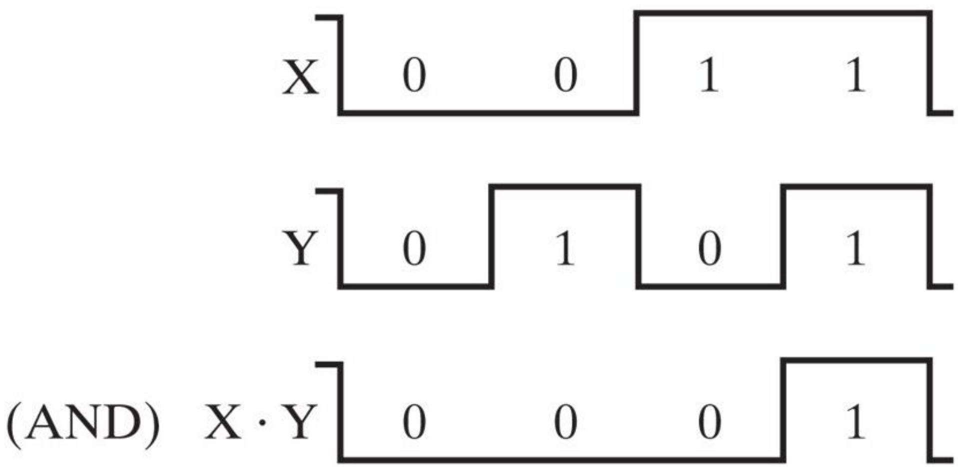
AND gate

OR gate

NOT gate or inverter

Diagrammi temporali

- Un diagramma temporale descrive l'andamento dei **segnali in ingresso** in funzione del tempo e la corrispondente **evoluzione dei segnali in uscita**
- E' un grafico che ha in **asse X il tempo e in asse Y un segnale binario** che può assumere solo i valori '0' o '1'
  - Es: diagramma temporale di una porta AND

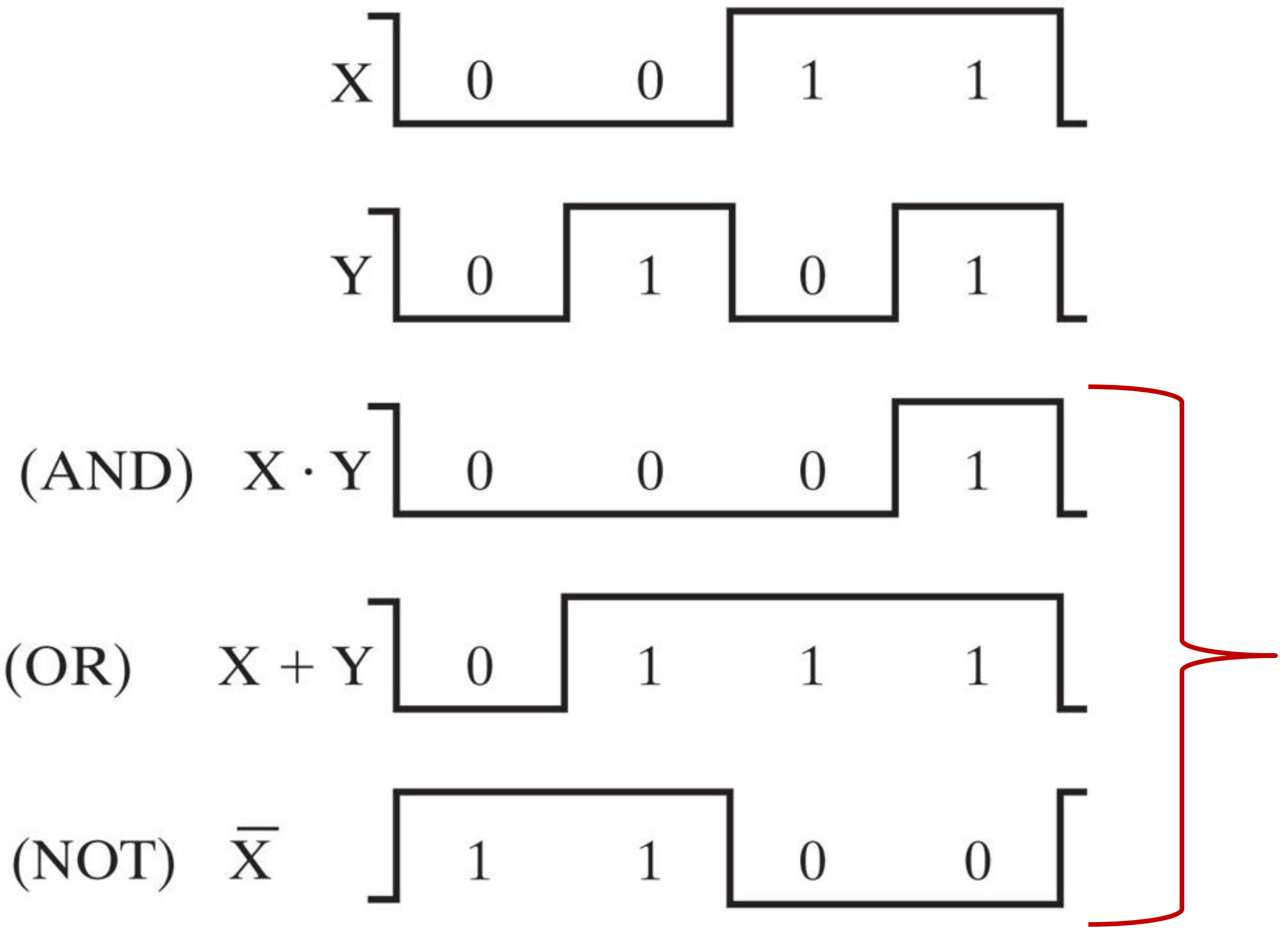


Questo è un diagramma temporale ideale: la transizione dell'uscita avviene istantaneamente al cambiamento degli ingressi

Tempo di ritardo di una porta logica

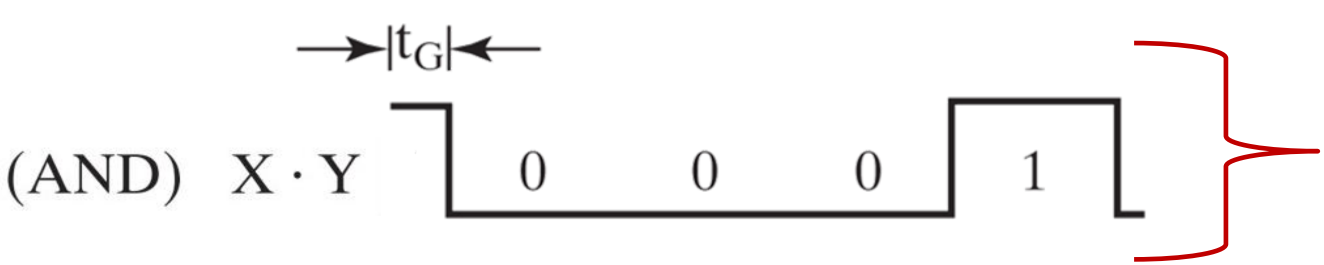
- Una **porta logica ideale** ha un tempo di ritardo nullo (una variazione in ingresso produce una variazione istantanea nell'uscita)
- Una **porta logica reale** è caratterizzata da un certo **tempo di ritardo** ($\boldsymbol{t_G}$: gate delay)
- t_G è il tempo che trascorre **dal momento in cui cambiano gli ingressi della porta al momento in cui tale cambiamento produce un cambiamento nelle uscite**
- Il tempo di ritardo dipende dalla tecnologia con cui la porta logica è realizzata
- Il tempo per commutare dal valore logico basso a quello alto non necessariamente è lo stesso di quello impiegato per la commutazione inversa

Diagrammi temporali di porte ideali



Diagrammi temporali di **porte ideali** (ritardo nullo)

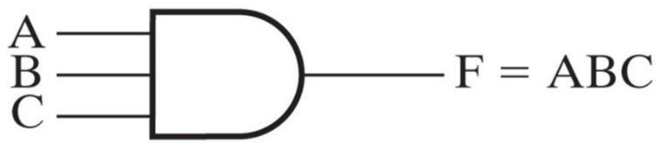
Diagrammi temporali di porta con ritardo



Porta AND con **ritardo t_G** ($t_{G,0->1}$ = $t_{G,1->0}$)

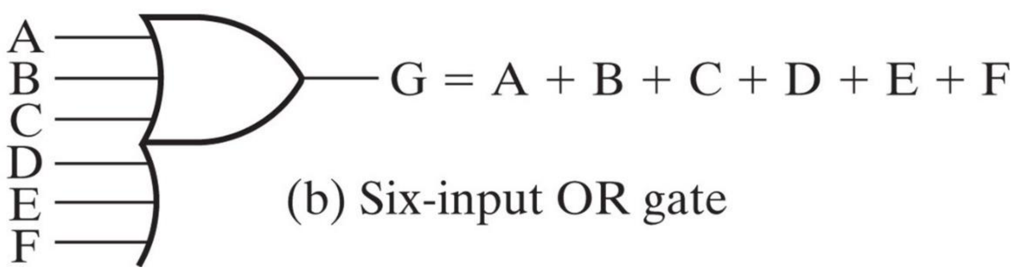
Porte logiche con più di due ingressi

- Una porta logica può avere anche **più di due ingressi**
  - Esempio 1: porta AND a 3 ingressi



(a) Three-input AND gate

  - Esempio 2 : porta OR a 6 ingressi



(b) Six-input OR gate

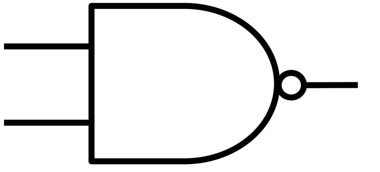
Altre porte logiche

- Ci sono altre porte logiche usate di frequente, oltre alle porte elementari (AND, OR, NOT):
  - **NAND**: complemento della porta AND
  - **NOR**: complemento della porta OR
  - **XOR**: OR esclusivo
  - **XNOR**: NOR esclusivo (complemento della porta XOR)

- Notazione: il **pallino** (bubble) all’uscita (o ingresso) di una porta logica rappresenta una **negazione**

Porta NAND

- Porta con due o più variabili in ingresso
- Notazioni: $Z = \overline{X \cdot Y}$ $Z = \overline{X\,Y}$ Z = X **NAND** Y
- Z = 1 se almeno uno tra X e Y vale 0
  altrimenti Z = 0



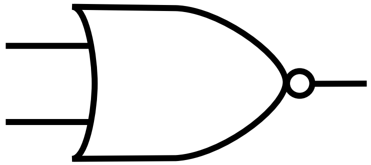
- **Tabella di verità**

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

Porta NOR

- Porta con due o più variabili in ingresso
- Notazioni: $Z = \overline{X \mathbf{+} Y}$ $Z = \overline{X \mathbf{V} Y}$ Z = X **NOR** Y
- <u>Z = 1 solo se entrambi X e Y valgono 0</u>

  <u>altrimenti Z = 0</u>

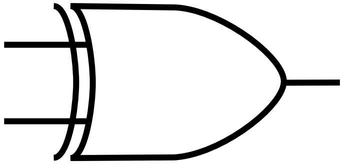


- **Tabella di verità**

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

Porta XOR

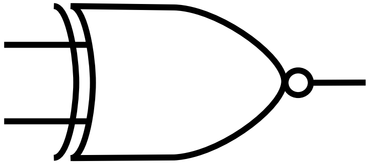
- Porta con due o più variabili in ingresso
- Notazioni: $Z = X \oplus Y$ $Z = X$ **XOR** Y
- Z = 1 se solo uno tra X e Y vale 1 (se X e Y sono diversi) altrimenti Z = 0
- **Tabella di verità**



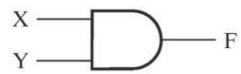
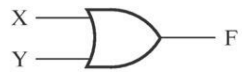
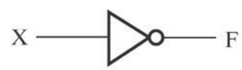
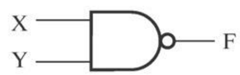
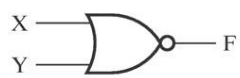
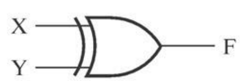
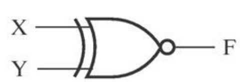
X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

Porta XNOR

- Porta con due o più variabili in ingresso
- Notazioni: $Z = \overline{X \oplus Y}$ $\qquad$ Z = X **XNOR** Y
- <u>Z = 1 se entrambi X e Y valgono 0 o 1</u> (se X e Y sono uguali) <u>altrimenti Z = 0</u>
- **Tabella di verità**

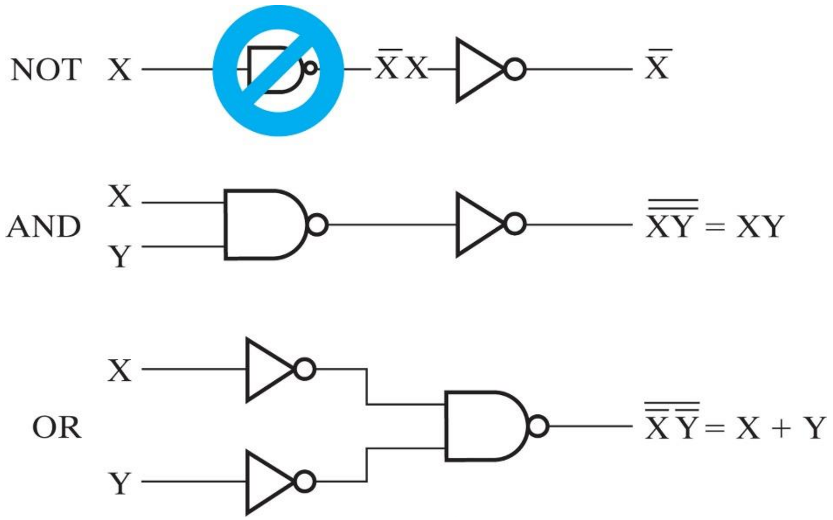


X	Y	X XNOR Y
0	0	1
0	1	0
1	0	0
1	1	1

Name	Distinctive-Shape Graphics Symbol	Algebraic Equation	Truth Table
AND		$F = XY$	X Y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = X + Y$	X Y F 0 0 0 0 1 1 1 0 1 1 1 1
NOT (inverter)		$F = \overline{X}$	X F 0 1 1 0
NAND		$F = \overline{X \cdot Y}$	X Y F 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$F = \overline{X + Y}$	X Y F 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive-OR (XOR)		$F = X\overline{Y} + \overline{X}Y$ $= X \oplus Y$	X Y F 0 0 0 0 1 1 1 0 1 1 1 0
Exclusive-NOR (XNOR)		$F = XY + \overline{X}\,\overline{Y}$ $= \overline{X \oplus Y}$	X Y F 0 0 1 0 1 0 1 0 0 1 1 1

Porte logiche universali

- Tutte le funzioni booleane si possono rappresentare con **sole porte NAND** (considerando l'inverter come caso particolare di porta NAND con un solo ingresso)



L'equivalenza di queste funzioni verrà dimostrata a breve (Teorema di De Morgan)

- Lo stesso vale per le **porte NOR**

=> NAND e NOR sono dette **porte universali** (complete)

Rappresentazione VHDL delle porte logiche

Operatori predefiniti per rappresentare le porte logiche in VHDL:

VHDL logic operator	Example
`not`	`F <= not X;`
`and`	`F <= X and Y;`
`or`	`F <= X or Y;`
`nand`	`F <= X nand Y;`
`nor`	`F <= X nor Y;`
`xor`	`F <= X xor Y;`
`xnor`	`F <= X xnor Y;`

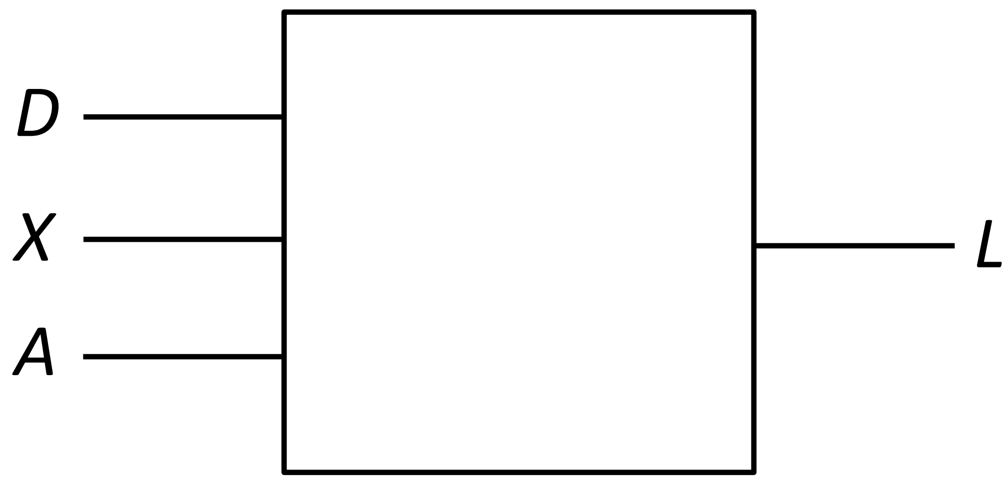
Funzioni booleane

Espressione booleana

- Un’espressione booleana è un’espressione algebrica composta da **variabili binarie** e **costanti binarie** (‘0’ e ‘1’), legate dalle **operazioni logiche di base AND, OR e NOT**

Funzione booleana: esempio

- Funzione booleana a tre variabili per l'abbassamento di un finestrino elettrico*:*



$$\boldsymbol{L = D\overline{X} + A}$$

Funzione booleana: es.

- Funzione booleana a tre variabili per controllare un alzacristalli elettrico: $\boldsymbol{L = D\overline{X} + A}$
  - L'uscita *L* comanda un motore che abbassa il finestrino: se *L* = '1' il motore viene azionato, se *L* = '0' il motore non è attivo
  - L'ingresso *D* = '1' se il guidatore preme il pulsante di abbassamento del finestrino
  - L'ingresso *X* è collegato all'uscita di un interruttore di fine corsa: *X* = '1' quando il finestrino è completamente abbassato
  - L'ingresso *A* (abbassamento automatico del finestrino) è un segnale generato dalla logica di temporizzazione tra *D* e *X*: se *D* rimane a '1' per più di 0.5 s, *A* diventa '1' e lo rimane finché *X* = '1'

➢ Il finestrino viene abbassato se vale '1' almeno una tra:

- *D* **AND NOT**(*X*): il guidatore preme il pulsante e il finestrino non è già completamente abbassato
- *A*: il guidatore preme il pulsante per abbassare il finestrino per più di 0.5 s

Funzione booleana: esempio

- Tabella di verità

Truth Table for the Function $L = D\overline{X} + A$

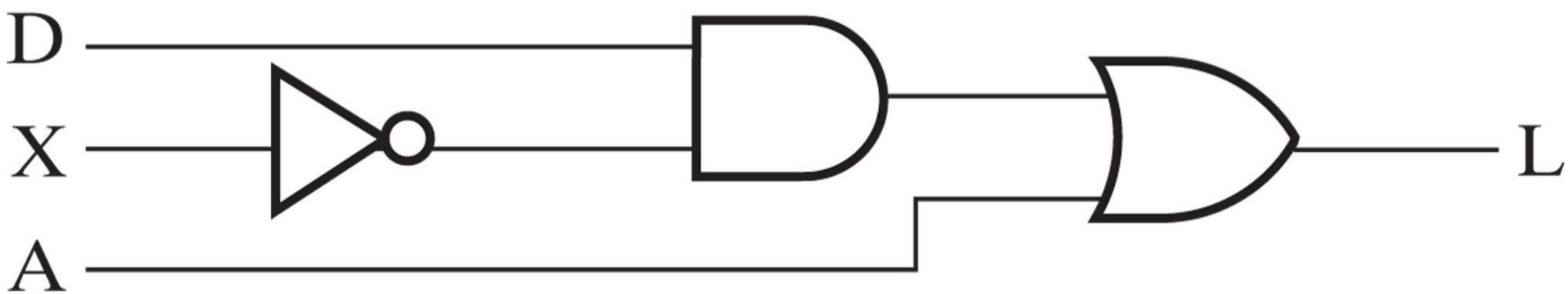
D	X	A	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- La tabella di verità di una funzione a n variabili ha 2^n righe (ognuna corrispondente ad una combinazione di ‘0’ e ‘1’ assegnati alle variabili) e mostra il valore della funzione in corrispondenza a quei valori delle variabili

Funzione booleana: esempio

- La funzione booleana si traduce in un circuito costituito da porte logiche che operano sulle variabili di ingresso per produrre le variabili di uscita

- Circuito logico per il l’abbassamento di un finestrino elettrico:



$$\boldsymbol{L = D\overline{X} + A}$$

Algebra booleana

Algebra di Boole

- L’algebra booleana (George Boole, 1854) comprende una serie di **postulati, proprietà e teoremi per manipolare le funzioni booleane**, che possono essere utilizzate per semplificare le espressioni
- Il nostro scopo è **minimizzare** il numero delle porte logiche del circuito e il numero di ingressi
- **Semplificare** (ottenere espressioni più semplici) significa tipicamente usare meno hardware, avere circuiti più veloci e usare meno area di silicio

Dualità nell'algebra di Boole

- L'**espressione duale** di un'espressione logica si ottiene
  - Sostituendo le porte **AND** con porte **OR**, e viceversa
  - Sostituendo **'0'** con **'1'**, e viceversa
- Nell'algebra di Boole vale la dualità: **se un'equazione è vera, lo è anche la sua duale** (pur essendo diversa da quella originaria, è ancora vera)

$$\cdot \rightleftarrows +$$

$$0 \rightleftarrows 1$$

Identità di base dell'algebra booleana

1. $X + 0 = X$	2. $X \cdot 1 = X$
3. $X + 1 = 1$	4. $X \cdot 0 = 0$
5. $X + X = X$	6. $X \cdot X = X$
7. $X + \overline{X} = 1$	8. $X \cdot \overline{X} = 0$
9. $\overline{\overline{X}} = X$	

- Relazioni tra una variabile, la stessa variabile negata e le costanti '0' e '1'
- NOTA: le due colonne contengono **equazioni duali**

Identità di base dell'algebra booleana

1. $X + 0 = X$	2. $X \cdot 1 = X$
3. $X + 1 = 1$	4. $X \cdot 0 = 0$
5. $X + X = X$	6. $X \cdot X = X$
7. $X + \overline{X} = 1$	8. $X \cdot \overline{X} = 0$
9. $\overline{\overline{X}} = X$	

- Possiamo dimostrarle con il principio di induzione perfetta, cioè dimostriamo che sono valide in **tutti i casi possibili**:

Identità di base dell'algebra booleana

1. $X + 0 = X$	2. $X \cdot 1 = X$
3. $X + 1 = 1$	4. $X \cdot 0 = 0$
5. $X + X = X$	6. $X \cdot X = X$
7. $X + \overline{X} = 1$	8. $X \cdot \overline{X} = 0$
9. $\overline{\overline{X}} = X$	

- Possiamo dimostrarle con il principio di induzione perfetta, cioè dimostriamo che sono valide in **tutti i casi possibili**:

$\mathbf{X}$	$\mathbf{X + 0}$	$\mathbf{X + 1}$	$\mathbf{X + X}$	$\mathbf{X + \overline{X}}$	$\mathbf{\overline{\overline{X}}}$
0	0	1	0	1	0
1	1	1	1	1	1

Proprietà dell’algebra booleana

10.	$X + Y = Y + X$	11.	$XY = YX$	Commutative
12.	$X + (Y + Z) = (X + Y) + Z$	13.	$X(YZ) = (XY)Z$	Associative
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	Distributive
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan’s

- In assenza di parentesi, **AND ha la precedenza su OR** (analogamente all’algebra ordinaria, dove la moltiplicazione ha la precedenza sull’addizione)
- E’ consigliabile usare sempre le parentesi, anche se sottintese!
- Alcune di queste proprietà valgono anche nell’algebra ordinaria (10-14), altre no (15-17)

Teorema di De Morgan

$$\overline{X+Y} = \overline{X} \cdot \overline{Y} \qquad \overline{X \cdot Y} = \overline{X} + \overline{Y}$$

- E' un teorema molto importante nella logica booleana, si usa **per ottenere il complemento di un'espressione**
- Può essere esteso a più variabili:

$$\overline{X_1 + X_2 + \ \dots + X_n} = \overline{X_1} \cdot \overline{X_2} \cdots \overline{X_n}$$

$$\overline{X_1 \cdot X_2 \cdot \cdots \cdot X_n} = \overline{X_1} + \overline{X_2} + \cdots + \overline{X_n}$$

- Forma **generalizzata**: si può ottenere il complemento di un'espressione logica sostituendo AND con OR e viceversa e negando tutte le variabili e le costanti

Dimostrazione del teorema di De Morgan

$$\overline{\boldsymbol{X}+\boldsymbol{Y}}=\overline{\boldsymbol{X}}\cdot\overline{\boldsymbol{Y}} \qquad \overline{\boldsymbol{X}\cdot\boldsymbol{Y}}=\overline{\boldsymbol{X}}+\overline{\boldsymbol{Y}}$$

- Si può dimostrare **sostituendo alle variabili tutte le combinazioni di valori possibili** (dimostrazione per **induzione**)

Dimostrazione del teorema di De Morgan

$$\overline{X+Y} = \bar{X} \cdot \bar{Y} \qquad \overline{X \cdot Y} = \bar{X} + \bar{Y}$$

- Si può dimostrare **sostituendo alle variabili tutte le combinazioni di valori possibili** (dimostrazione per **induzione**)

(a) X	Y	X + Y	$\overline{X+Y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

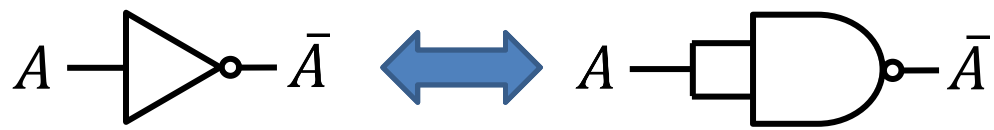
(b) X	Y	$\bar{X}$	$\bar{Y}$	$\bar{X} \cdot \bar{Y}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

NAND e NOR sono operatori universali

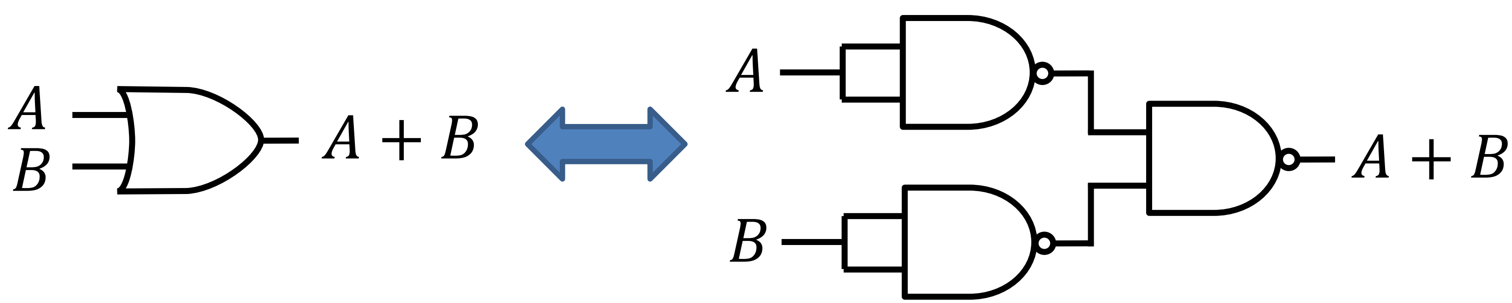
$$\overline{\boldsymbol{X}+\boldsymbol{Y}} = \overline{\boldsymbol{X}}\cdot\overline{\boldsymbol{Y}} \qquad\qquad \overline{\boldsymbol{X}\cdot\boldsymbol{Y}} = \overline{\boldsymbol{X}}+\overline{\boldsymbol{Y}}$$

- Ogni funzione logica è rappresentabile con sole funzioni NAND o sole funzioni NOR

NOT:



OR: $A + B = \overline{\overline{A+B}} = \overline{\bar{A}\cdot\bar{B}}$ (De Morgan)



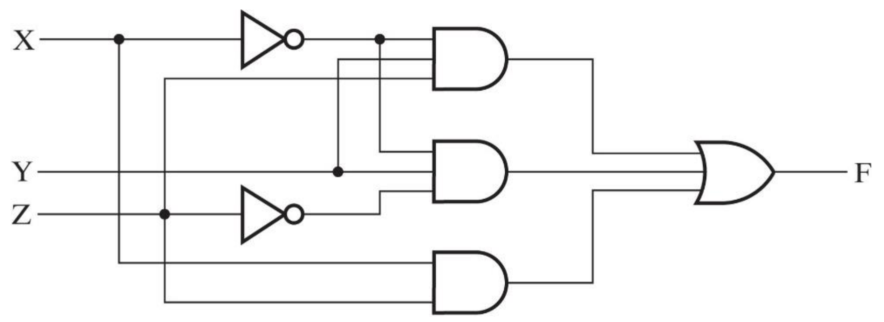
Esempio: applicazione delle proprietà

$$F = \boxed{\bar{X}YZ + \bar{X}Y\bar{Z} + XZ}$$

$= \bar{X}Y(Z + \bar{Z}) + XZ$ → proprietà distributiva

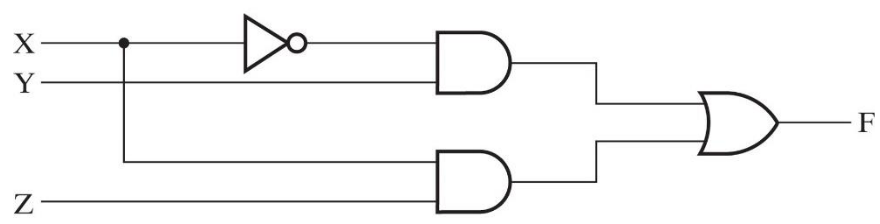
$= \bar{X}Y \cdot 1 + XZ$ → identità # 7: $X + \bar{X} = 1$

$= \boxed{\bar{X}Y + XZ}$ → identità # 2: $X \cdot 1 = X$



(a) $F = \bar{X}YZ + \bar{X}Y\bar{Z} + XZ$

=> Le due espressioni sono equivalenti, (b) risparmia in termini di numero di porte!



(b) $F = \bar{X}Y + XZ$

Porte logiche e letterali

- Consideriamo l'implementazione di un'equazione booleana con porte logiche in forma di somma di prodotti
  - Ogni **termine prodotto** nell'equazione rappresenta una porta logica di tipo AND
  - Ogni **variabile** nel termine prodotto rappresenta un ingresso della porta AND
- Si definisce **letterale** ogni **occorrenza separata di una variabile, diretta o negata**
  - Esempi:

(a) $F = \bar{X}YZ + \bar{X}Y\bar{Z} + XZ$ → 3 termini, 8 letterali

(b) $G = \bar{X}Y + XZ$ → 2 termini, 4 letterali

Teorema di assorbimento

$$\boldsymbol{X + XY = X}$$

(il secondo termine si può omettere perché ridondante)

- Dimostrare il teorema
  1. Usando il principio di induzione perfetta
  2. Usando le proprietà dell'algebra booleana

Teorema di assorbimento

$$X + XY = X$$

(il secondo termine si può omettere perché ridondante)

1. Usando il principio di induzione perfetta

X	Y	XY	X + XY
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Teorema di assorbimento

$$\boldsymbol{X + XY = X}$$

(il secondo termine si può omettere perché ridondante)

2. Usando le proprietà dell'algebra booleana

$X + XY = X(1 + Y)$ → proprietà distributiva

$= X \cdot 1$ → identità # 3: $X + 1 = 1$

$= X$ → identità # 1: $X \cdot 1 = \mathrm{X}$

Teorema del consenso

$$\boldsymbol{XY + \bar{X}Z + YZ = XY + \bar{X}Z}$$

- Il terzo termine si può omettere perché ridondante: Y e Z compaiono in AND con X e X negato, rispettivamente nel primo e nel secondo termine
- <u>Dimostrazione:</u>

$$XY + \bar{X}Z + YZ = XY + \bar{X}Z + YZ(X + \bar{X})$$ → identità # 7: $X + \bar{X} = 1$

$$= XY + \bar{X}Z + XYZ + \bar{X}YZ$$ → proprietà distributiva

$$= XY + XYZ + \bar{X}Z + \bar{X}YZ$$ → proprietà commutativa

$$= XY(1 + Z) + \bar{X}Z(1 + Y)$$ → proprietà distributiva

$$= XY + \bar{X}Z$$ → identità # 3: $X + 1 = 1$

Complemento di una funzione

- Per ottenere il complemento di una funzione, si **scambiano '0' e '1' nella colonna dell'uscita della tabella di verità**
- Ci sono due modi alternativi per ottenere l'espressione algebrica del complemento di una funzione
  a) Usare il teorema di De Morgan
  b) Trovare l'espressione duale e negare ciascun letterale (sia le variabili che le costanti). Infatti il teorema di De Morgan generalizzato afferma che il complemento di una funzione si può ottenere scambiando AND con OR e facendo il complemento di ciascun letterale

Esempio: complemento di una funzione

- Esercizio: trovare il complemento delle seguenti funzioni nei due modi descritti nella slide precedente

$$F_1 = \bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z$$
$$F_2 = X(\bar{Y}\bar{Z} + YZ)$$

Esempio: complemento di una funzione

- Esercizio: trovare il complemento delle seguenti funzioni nei due modi descritti nella slide precedente

$$F_1 = \bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z$$

a) Trovare il complementare usando il teorema di De Morgan

$$\overline{F_1} = \overline{\bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z}$$ → Negazione di F_1

$$= \overline{\bar{X}Y\bar{Z}} \cdot \overline{\bar{X}\bar{Y}Z}$$ → Teorema di De Morgan

$$= (\bar{\bar{X}} + \bar{Y} + \bar{\bar{Z}}) \cdot (\bar{\bar{X}} + \bar{\bar{Y}} + \bar{Z})$$ → Teorema di De Morgan

$$= (X + \bar{Y} + Z) \cdot (X + Y + \bar{Z})$$ → Identità # 9: $\bar{\bar{X}} = \mathrm{X}$

Esempio: complemento di una funzione

- Esercizio: trovare il complemento delle seguenti funzioni nei due modi descritti nella slide precedente

$$F_2 = X(\bar{Y}\bar{Z} + YZ)$$

a) Trovare il complementare usando il teorema di De Morgan

$$\begin{aligned}
\overline{F_2} &= \overline{X(\bar{Y}\bar{Z} + YZ)} && \rightarrow \text{Negazione di } F_2 \\
&= \bar{X} + (\overline{\bar{Y}\bar{Z} + YZ}) && \rightarrow \text{Teorema di De Morgan} \\
&= \bar{X} + \left(\overline{\bar{Y}\bar{Z}}\right) \cdot (\overline{YZ}) && \rightarrow \text{Teorema di De Morgan} \\
&= \bar{X} + (Y + Z) \cdot (\bar{Y} + \bar{Z}) && \rightarrow \text{Teorema di De Morgan}
\end{aligned}$$

Esempio: complemento di una funzione

- Esercizio: trovare il complemento delle seguenti funzioni nei due modi descritti nella slide precedente

$$F_1 = \bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z$$

b) Trovare l'espressione duale e negare ciascun letterale

Duale di F_1: $(\bar{X} + Y + \bar{Z}) \cdot (\bar{X} + \bar{Y} + Z)$

$$\Rightarrow \overline{F_1} = (X + \bar{Y} + Z) \cdot (X + Y + \bar{Z})$$

Attenzione alle parentesi: è bene metterle prima di sostituire un'espressione con la duale!

Esempio: complemento di una funzione

- Esercizio: trovare il complemento delle seguenti funzioni nei due modi descritti nella slide precedente

$$F_2 = X(\bar{Y}\bar{Z} + YZ)$$

b) Trovare l'espressione duale e negare ciascun letterale

Duale di F_2: $X + [(\bar{Y} + \bar{Z}) \cdot (Y + Z)]$

$$\Rightarrow \overline{F_2} = \bar{X} + (Y + Z) \cdot (\bar{Y} + \bar{Z})$$

Riepilogo

- Con le porte logiche fondamentali AND, OR e NOT è possibile realizzare qualsiasi funzione logica
- Esistono altre porte logiche NAND, NOR, XOR derivate da quelle fondamentali: NOR e NAND sono porte universali
- L'algebra booleana opera su funzioni logiche
  - Principio di dualità
  - Identità di base e proprietà (teorema di De Morgan)
- Lo scopo è generalmente quello di semplificare le espressioni, per poter realizzare circuiti logici più semplici possibile (meno porte, meno ingressi, meno connessioni)
- Nella prossima lezione vedremo delle tecniche per la minimizzazione delle funzioni logiche

Disclaimer

Figures from *Logic and Computer Design Fundamentals*, Fifth Edition, GE Mano | Kime | Martin