ARCHITETTURA DEGLI ELABORATORI

A.A. 2020-2021

Università di Napoli Federico II

Corso di Laurea in Informatica

Docenti

Proff. Luigi Sauro gruppo 1 (A-G)

Silvia Rossi gruppo 2 (H-Z)

ALGEBRA DI BOOLE E RETI COMBINATORIE

Porte logiche

- Come accennato, un calcolatore può essere visto come un complesso sistema digitale che manipola e memorizza informazioni rappresentate in codice *binario*.
- I componenti digitali che costituiscono i mattoni fondamentali di un calcolatore sono le *porte logiche*
- Le porte logiche realizzano delle semplici operazioni che prendono uno o più input e producono un output
- Gli input sono indicati generalmente con le prime lettere dell'alfabeto A,B,C,D,… mentre gli output con le ultime X,Y,…
- Poiché gli input e gli output possono assumere in generale sia il valore 0 che 1 allora essi costituiscono delle variabili (dette variabili booleane)

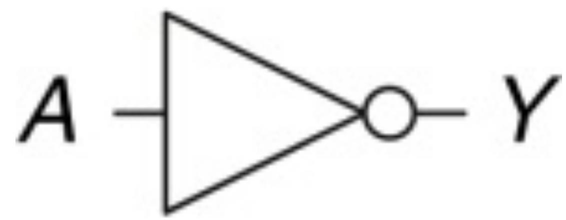
Logic Gates

- **Perform logic functions:**
  - inversion (NOT), AND, OR, NAND, NOR, etc.
- **Single-input:**
  - NOT gate, buffer
- **Two-input:**
  - AND, OR, XOR, NAND, NOR, XNOR
- **Multiple-input**

Porta NOT

- La porta NOT restituisce in output il *complemento* dell’input:
- Y è uguale a 1 se e solo se A non è uguale a 1: Y=A-1
- Altri simboli per NOT sono ¬A o ~A (usati soprattutto dai logici)

NOT



$$Y = \overline{A}$$

A	Y
0	1
1	0

Porta AND

- La porta AND restituisce in output la *congiunzione* degli input:

AND

$$Y = AB$$

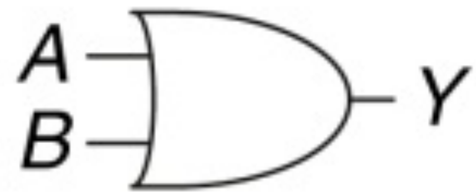
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

- Y è uguale a 1 se e solo se A e B sono entrambi uguali a 1
- L'operatore AND è anche rappresentato con $A \wedge B$

Porta OR

- La porta OR restituisce in output la *disgiunzione* degli input:

OR



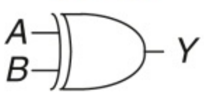
$$Y = A + B$$

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

- Y è uguale a 1 se e solo se A o B è uguale a 1
- L'operatore OR è anche rappresentato con A ∨ B

Altre porte logiche

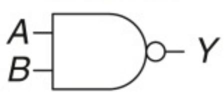
XOR



$Y = A \oplus B$

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

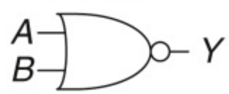
NAND



$Y = \overline{AB}$

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

NOR



$Y = \overline{A + B}$

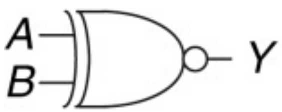
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

- XOR: Y = 1 se e solo se A oppure B è uguale a 1
- NAND: Y è uguale a 1 se e solo se A o B non sono uguali a 1
- NOR: Y è uguale a 1 se e solo se A e B non sono uguali a 1

XNOR

- Problema: quale è la tabella di verità della porta XNOR?

XNOR



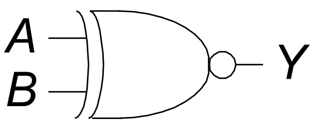
$$Y = \overline{A \oplus B}$$

A	B	Y
0	0	
0	1	
1	0	
1	1	

XNOR

- Problema: quale è la tabella di verità della porta XNOR?

XNOR



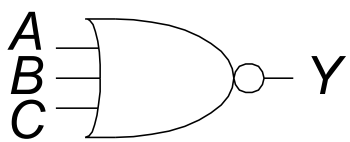
$$Y = \overline{A \oplus B}$$

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

Porte logiche con più linee di input

- Le porte logiche AND, OR, NAND,… possono avere anche più di 2 linee di ingresso.

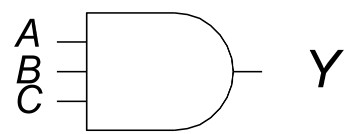
NOR3



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

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

NAND3



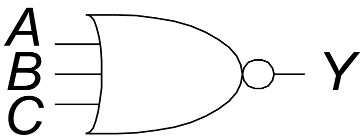
$$Y = ABC$$

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

Porte logiche con più linee di input

- Le porte logiche AND, OR, NAND,... possono avere anche più di 2 linee di ingresso.

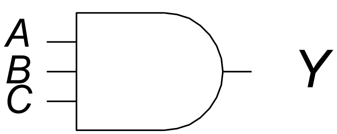
NOR3



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

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

AND3



$Y = ABC$

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

Logic Levels

- Discrete voltages represent 1 and 0
- For example:
  - 0 = *ground* (GND) or 0 volts
  - 1 = V_{DD} or 5 volts
- What about 4.99 volts? Is that a 0 or a 1?
- What about 3.2 volts?

Logic Levels

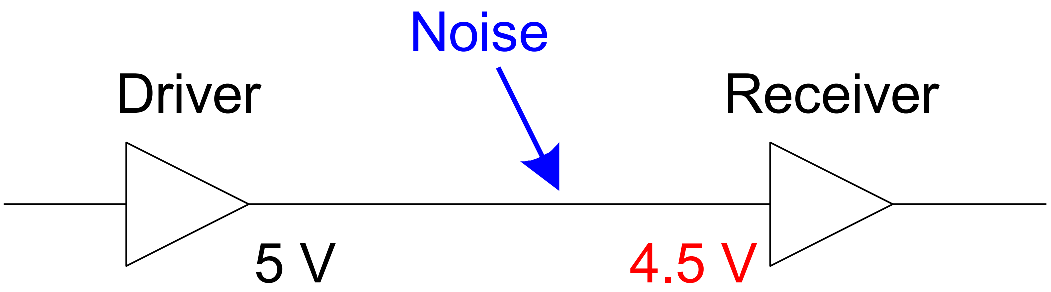
- *Range* of voltages for 1 and 0
- Different ranges for inputs and outputs to allow for *noise*

Cosa c’è sotto?

- Finora abbiamo descritto le porte logiche in termini astratti in cui gli ingressi e le uscite assumono i valori binari 0 o 1.
- Chiaramente, per realizzare queste porte in concreto, questi valori devono corrispondere a una certa grandezza fisica.
- Nei calcolatori elettronici (questo laptop per intenderci) la grandezza fisica che *reifica* i valori logici 0 e 1 è il potenziale elettrico.
- Daremo per scontato che abbiate le nozioni minime di base su cosa sia il potenziale elettrico (o che almeno non mettiate le dita in una presa da 220V).
- Il valore logico 0 è rappresentato dal valore di potenziale di 0V (GRD)
- Il valore logico 1 è rappresentato da un valore di potenziale V_{DD} fornito dal generatore.
  - Fino agli anni ottanta V_{DD} = 5V, l’avvento di portatili, tablet e smarthphone ha reso necessario operare con valori di potenziale più bassi $V_{DD} \leq 1{,}5V$

What is Noise?

- **Anything that degrades the signal**
  - E.g., resistance, power supply noise, coupling to neighboring wires, etc.
- **Example:** a gate (driver) outputs 5 V but, because of resistance in a long wire, receiver gets 4.5 V

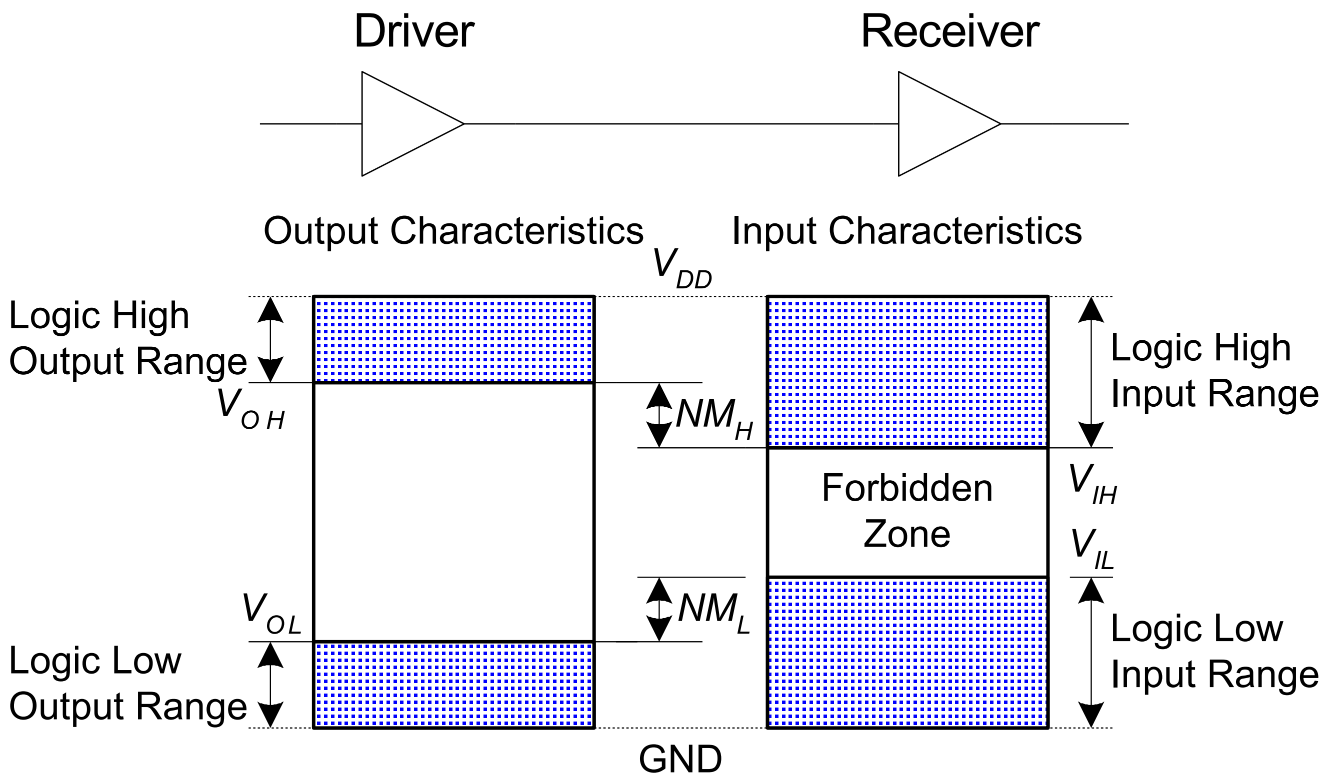


The Static Discipline

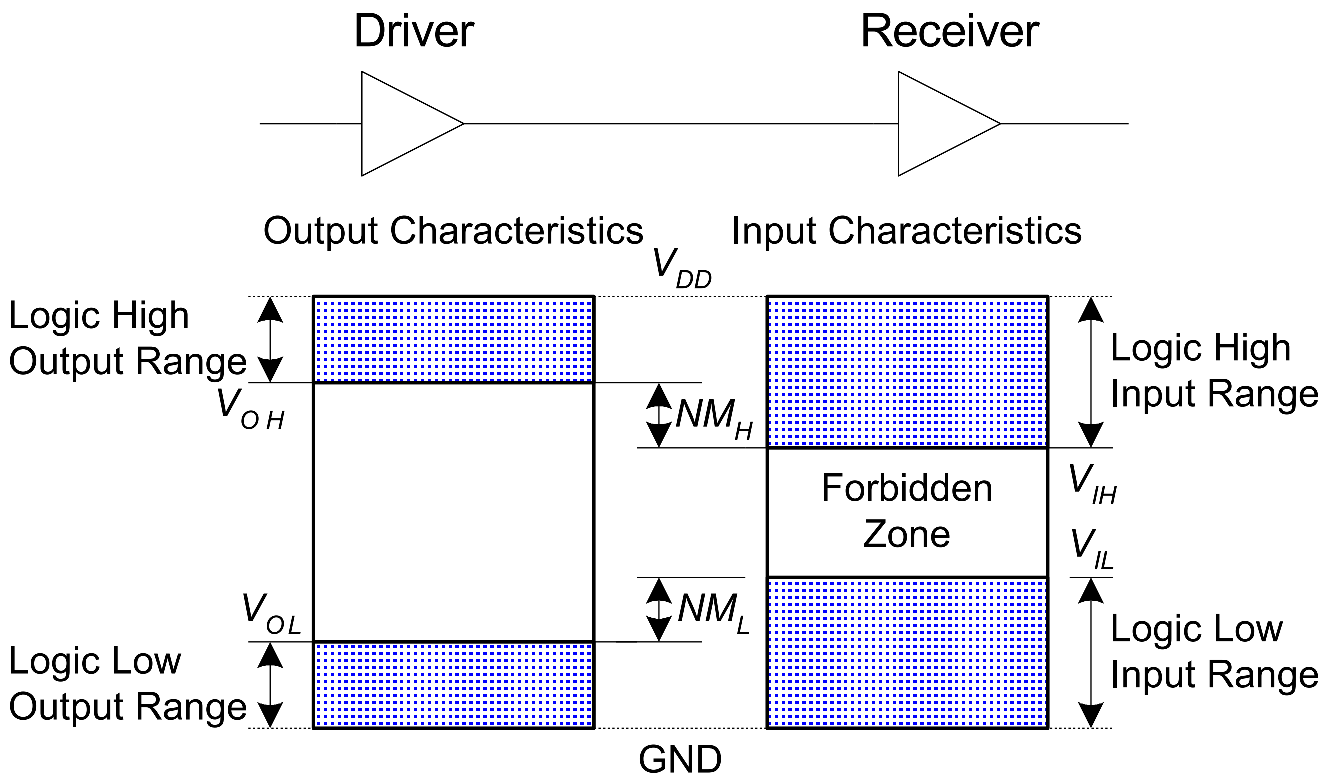
- With logically valid inputs, every circuit element must produce logically valid outputs

- Use limited ranges of voltages to represent discrete values

Noise Margins



Noise Margins



High Noise Margin: $NM_H = V_{OH} - V_{IH}$

Low Noise Margin: $NM_L = V_{IL} - V_{OL}$

V_{DD} Scaling

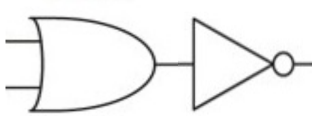
- In 1970's and 1980's, V_{DD} = 5 V
- V_{DD} has dropped
  - Avoid frying tiny transistors
  - Save power
- 3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V, 1.0 V, …
  - Be careful connecting chips with different supply voltages

Altre porte logiche

- La porta NAND è un elemento circuitale che viene prodotto direttamente, tuttavia può essere ottenuto complementando una porta AND, ovvero mettendo *in serie* una porta AND e una NOT:

A	B	$A \cdot B$	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

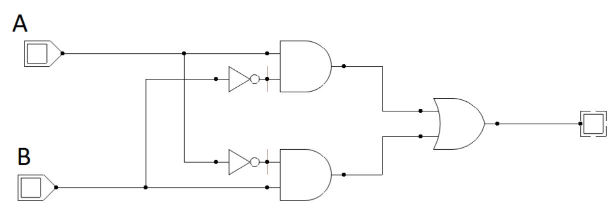
- Analogamente la porta NOR si può ottenere mettendo *in serie* una porta OR e una NOT:



Altre porte logiche

- Anche la porta XOR può essere ottenuta mediante porte AND, OR e NOT.
  - XOR: Y = 1 sse A≠B
  - A≠B sse (A=1 e B=0) o (A=0 e B=1)
  - $Y=(A\cdot\overline{B})+(\overline{A}\cdot B)$

A	B	$A\cdot\overline{B}$	$\overline{A}\cdot B$	+
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0



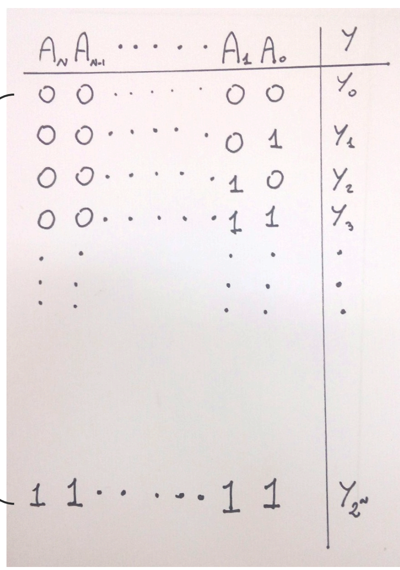
Funzioni booleane

- Le porte logiche esaminate finora costituiscono delle specifiche *funzioni booleane*

- Domanda: quante funzioni booleane di N variabili esistono?

$$f:\{0,1\}^N \rightarrow \{0,1\}$$

2^N righe



A_N	A_{N-1}	…	A_1	A_0	Y
0	0	…	0	0	Y_0
0	0	…	0	1	Y_1
0	0	…	1	0	Y_2
0	0	…	1	1	Y_3
⋮	⋮		⋮	⋮	⋮
1	1	…	1	1	Y_{2^N}

Una generica f assegna «liberamente» valori 0 o 1 alle Y_i
Quindi il numero di funzioni booleane di N variabili è pari al numero di parole binarie di lunghezza 2^N, ovvero:

$$2^{2^N}$$

Funzioni booleane di 2 variabili

Se le variabili sono 2 allora ottengo $2^4 = 16$ possibili funzioni booleane

A B

x y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		AND					XOR	OR	NOR	NXOR					NAND	

Funzioni booleane di 2 variabili

Se le variabili sono 2 allora ottengo $2^4 = 16$ possibili funzioni booleane

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Funzione costante
Y=0

Funzione costante
Y=1

Funzioni booleane di 2 variabili

Se le variabili sono 2 allora ottengo $2^4 = 16$ possibili funzioni booleane

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y=A

Y=B

Y=$\bar{B}$

Y=$\bar{A}$

Funzioni booleane di 2 variabili

Se le variabili sono 2 allora ottengo $2^4 = 16$ possibili funzioni booleane

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y=$A\bar{B}$

Y=$\bar{A}$B

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

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

Funzioni booleane di 2 variabili

Se le variabili sono 2 allora ottengo $2^4 = 16$ possibili funzioni booleane

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y= $\bar{B}$

Y=A

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

Funzioni booleane di 2 variabili

Se le variabili sono 2 allora ottengo $2^4 = 16$ possibili funzioni booleane

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y= $\bar{A}$

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

Y=B

Funzioni booleane di 2 variabili

- Se le variabili sono 2 allora ottengo 16 possibili funzioni booleane

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

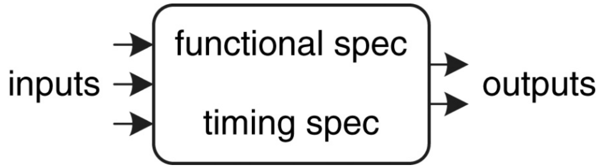
$Y = (A \leq B) \cdot (B \leq A)$ $A \equiv B$

$Y = A \leq B$

$Y = B \leq A$

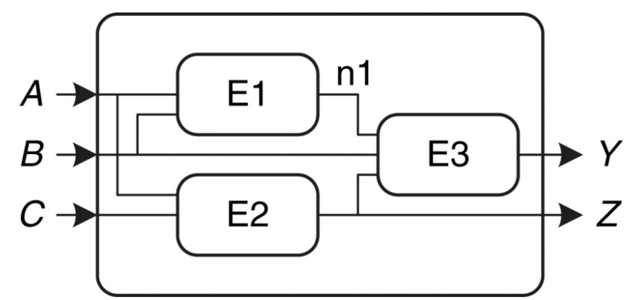
Circuiti digitali

- In generale un circuito digitale è una rete che elabora segnali discreti (rappresentati da variabili booleane). Prescidendo dalla sua configurazione interna, un circuito può essere visto come una *black-box* con
  - Uno o più input
  - Uno o più output
  - Una *specifica funzionale* che rappresenta la relazione fra input e output
  - Una *specifica temporale* che descrive il ritardo che intercorre affinché I segnali di input si propaghino nel circuito fino agli output.



Circuiti digitali

- La struttura interna di un circuito è composta da *elementi* e *nodi.*
- Un elemento è esso stesso un circuito digitale.
- Un nodo è una connessione che trasporta il segnale (e.g. filo elettrico). I nodi si distinguono in nodi di input, output e interni
  - input: riceve il segnale dal mondo esterno
  - output: riporta il segnale al mondo esterno
  - Nodo interno: connette due elementi

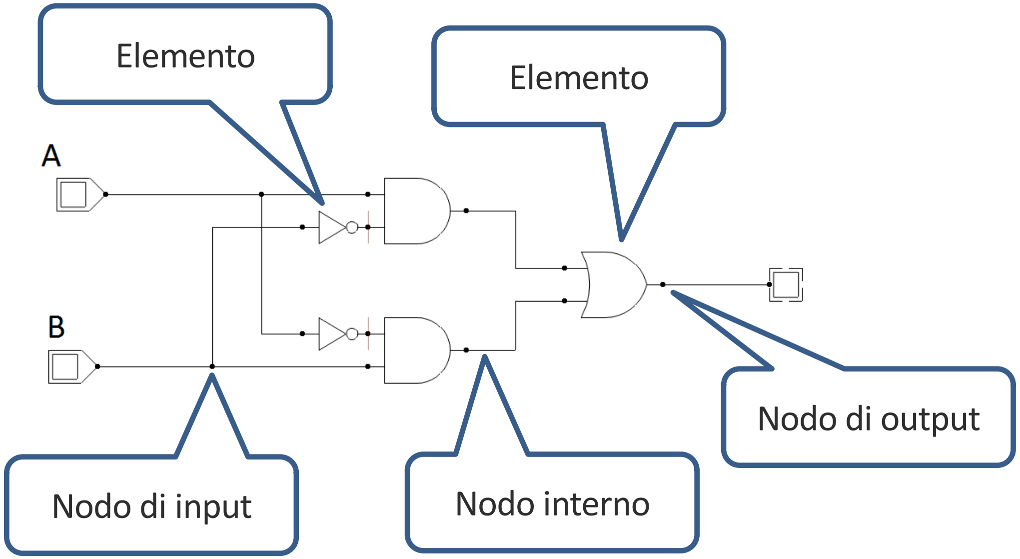


Reti combinatorie e sequenziali

- Vi sono due grandi categorie di circuiti digitali: le reti combinatorie e le reti sequenziali.
- In una rete combinatoria, i valori degli output dipendono esclusivamente dal valore corrente degli input (al netto dei ritardi di propagazione). In tal senso, le reti combinatorie si dicono memoryless, ovvero non hanno memoria della *“storia”* precedente del circuito
- In un rete sequenziale, invece, gli output dipendono non solo dal valore corrente degli input, ma anche dai valori precedenti. Si dice quindi che il circuito ha memoria.

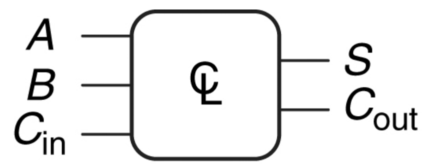
Reti combinatorie

- Le porte logiche viste finora sono un esempio di rete combinatoria.
- Un'altro esempio di rete combinatoria l'abbiamo visto in precedenza



Full adder

- Quando non siamo interessati alla struttura interna allora una rete combinatoria è descritta come segue

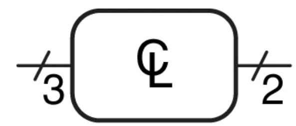


$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + AC_{in} + BC_{in}$$

- Questa rete combinatoria rappresenta un full adder di cui parleremo in seguito.
- Come avrete già capito, vi sono molti modi differenti per realizzare le funzioni che corrispondono al output S e C_{out}

Input e output multipli

- Per semplificare la rappresentazione grafica, si usa la notazione qui di fianco per indicare ingressi ed uscite multiple.
- In figura (a) un generico circuito combinatorio con 3 input e 2 output
- Il numero di input e output può essere omesso quando chiaro dal contesto o non rilevante. In figura (b) due generiche reti combinatorie in sequenza. Notate che il numero di output della prima rete deve essere uguale al numero di input della seconda.



(a)

(b)

Regole di composizione di reti combinatorie

- Come abbiamo visto reti combinatorie possono essere composte per formare reti più grandi. Cosa ci garantisce che il risultato sia ancora una rete combinatoria? Ecco un insieme di regole che costituiscono delle condizioni sufficienti (ma non necessarie)
  - Ogni elemento è esso stesso una rete combinatoria
  - Ogni nodo che non è un input connette esattamente un output di un elemento
  - Il circuito non contiene cicli, ogni cammino interno alla rete visita un nodo al più una volta

Regole di composizione di reti combinatorie

- Quali dei seguenti circuiti costituisce, secondo le regole di composizione, una rete combinatoria?

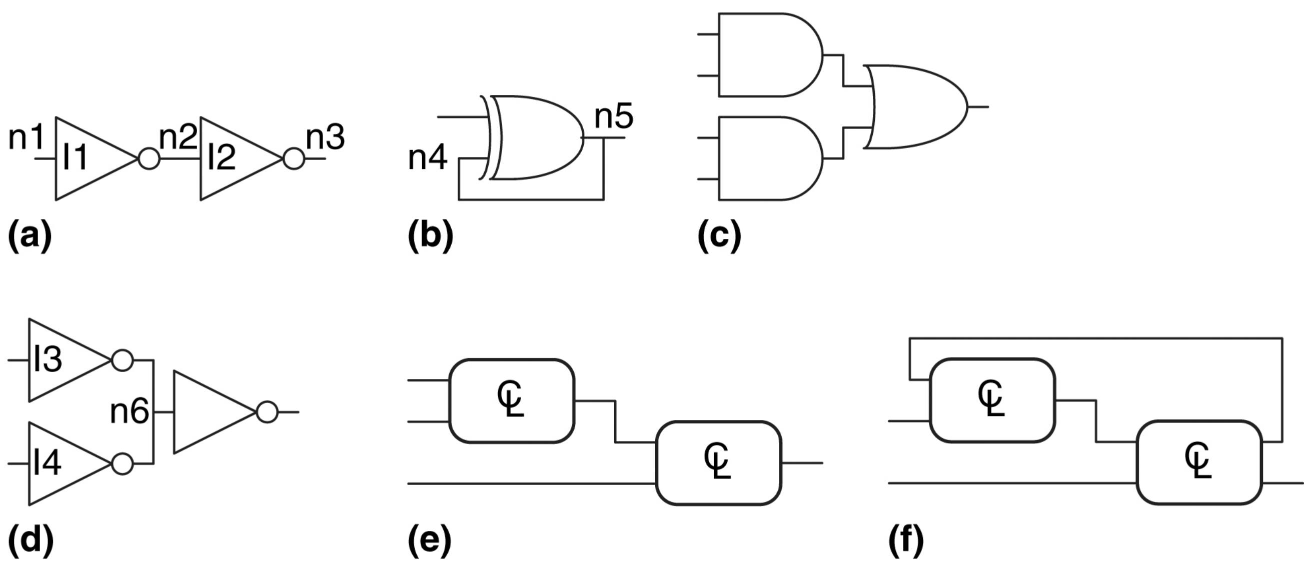


Tabella di verità e espressioni booleane

- Le specifiche funzionali di un circuito combinatorio sono descritte tramite tabelle di verità o espressioni booleane
- In generale, più espressioni booleane corrispondo alla stessa tabella di verità
  - Per esempio abbiamo visto che $\overline{A\bar{B}}$ e $\bar{A} + B$ corrispondono alla stessa tabella di verità, ovvero $\overline{A\bar{B}} = \bar{A} + B$.
- Come si ricava la tabella di verità di una data espressione?
- Viceversa, a partire da una tabella di verità, come si ricava una espressione corrispondente?

Dalle espressioni alle tabelle

- Il calcolo di una tabella di verità a partire una espressione booleana non è dissimile concettualmente dal calcolo di una espressione aritmetica. Si calcolano a ritroso le tabelle di verità delle sue sotto espressioni.
- Consideriamo l'espressione ($\bar{A}$B)+(B$\oplus$C)

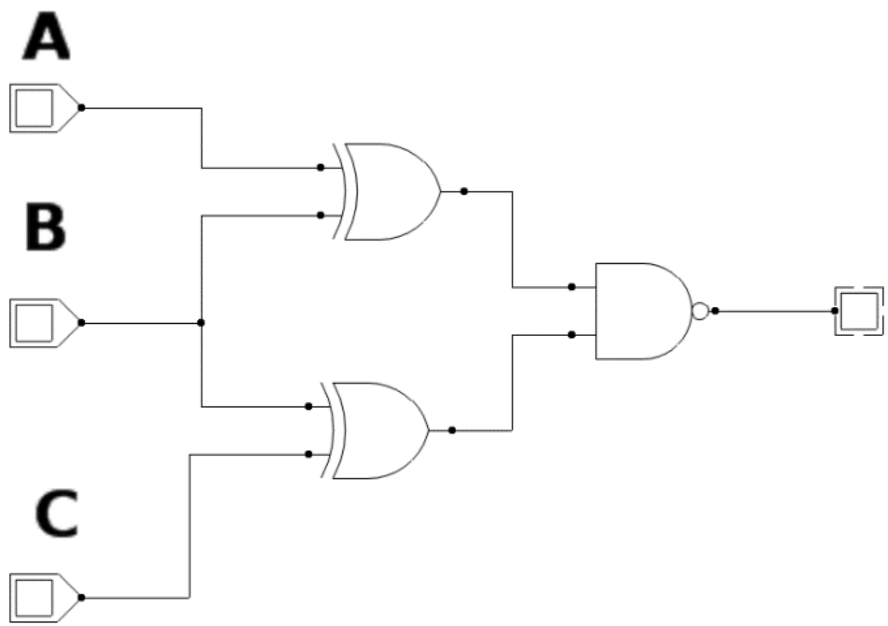
Dalle espressioni alle tabelle

- Il calcolo di una tabella di verità a partire una espressione booleana non è dissimile concettualmente dal calcolo di una espressione aritmetica. Si calcolano a ritroso le tabelle di verità delle sue sotto espressioni.
- Consideriamo l'espressione $(\bar{A}B)$+(B$\oplus$C)

A B C	$\bar{A}\cdot$	$\bar{A}\cdot B$	$B \oplus C$	$(\bar{A}B)+(B\oplus C)$
0 0 0	1	0	0	0
0 0 1	1	0	1	1
0 1 0	1	1	1	1
0 1 1	1	1	0	1
1 0 0	0	0	0	0
1 0 1	0	0	1	1
1 1 0	0	0	1	1
1 1 1	0	0	0	0

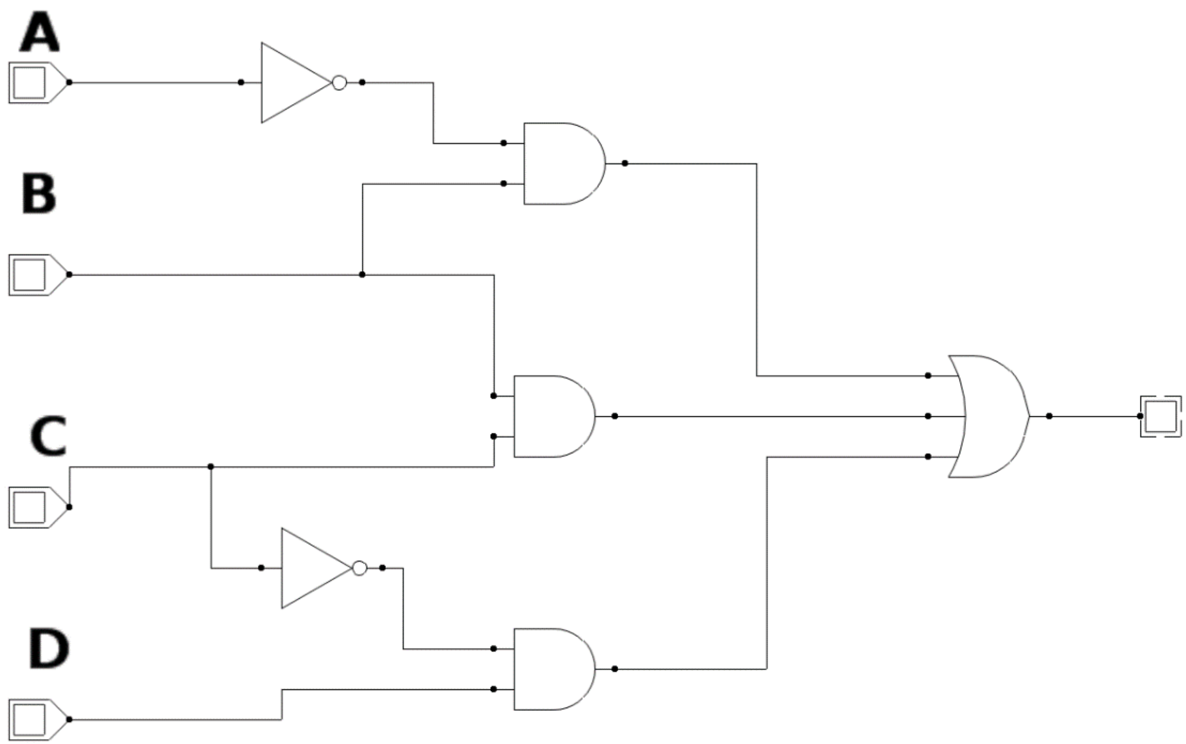
Esercizi

- Dimostrare che $(A \cdot \overline{B}) + (\overline{A} \cdot B)$ è equivalente alla porta XOR calcolandone la tabella di verità
- Esercizi del libro "Digital Design and Computer Architecture – ARM edition"
  - 1.87
- Scrivere l'espressione booleana e la relativa tabella di verità della rete:



Esercizi

- Scrivere l’espressione booleana e la relativa tabella di verità della rete:



Dalle tabelle alle espressioni

- ricavare da una tabella di verità un espressione booleana
- forma sintattica ben precisa: SOP (somma di prodotti)
- Definizioni preliminari:
  - Una variabile booleana A o la sua negata $\bar{A}$ sono detti litterali
  - Un prodotto (AND) di litterali è detto implicante: $\bar{A}$B , $\overline{A}$B$\bar{C}$, B sono tutti implicanti per una funzione booleana di almeno 3 variabili.
  - Dato un insieme K di variabili booleane, un mintermine di K è un implicante che comprende (positive o negate) tutte le variabili in K .
    - Ad esempio, se $\mathrm{K} = \{A, B, C\}$ allora $\overline{A}$B$\bar{C}$ è un mintermine per K. Invece, $\bar{A}$B è un implicante ma non un mintermine.
  - Analogamente, un maxtermine di K è una somma di litterali in cui occorrono tutte le variabili in K.
    - Ad esempio $\bar{A} + \bar{B} + \bar{C}$ è un maxtermine di $\mathrm{K} = \{A, B, C\}$.
- ordine di precedenza NOT →AND→OR
  - $\overline{A}$B$\bar{C}$ $+$ CB = ($\overline{A}$B$\bar{C}$) + (CB)

Some Definitions

- **Complement**: variable with a bar over it

  $\bar{A}, \bar{B}, \bar{C}$
- **Literal**: variable or its complement

  $A, \bar{A}, B, \bar{B}, C, \bar{C}$
- **Implicant**: product of literals

  $AB\bar{C}, \bar{A}C, BC$
- **Minterm**: product that includes all input variables

  $AB\bar{C}, A\bar{B}\bar{C}, ABC$
- **Maxterm**: sum that includes all input variables

  $(A+\bar{B}+C), (\bar{A}+B+\bar{C}), (\bar{A}+\bar{B}+C)$

La forma SOP (Sum-Of-Products)

- Ognuna delle 2^N righe di una tabella di verità è caratterizzata da un mintermine

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y=AB

Y=$A\bar{B}$

Y=$\bar{A}$B

Y= $\bar{A}\bar{B}$

La forma SOP

- I mintermini sono enumerati rigo dopo rigo a partire da 0,1 e così via. Quindi ogni mintermine è denotato dal numero binario della configurazione di input corrispondente.

A	B	Y	minterm	minterm name
0	0	0	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	0	$A\ B$	m_3

La forma SOP

- Quindi ad ogni tabella di verità corrisponde una espressione booleana ottenuta sommando tutti i mintermini per cui il valore dell'output Y è pari a 1

A	B	Y	minterm	minterm name
0	0	0	$\bar{A}\ \bar{B}$	m_0
0	1	1	$\bar{A}\ B$	m_1
1	0	0	$A\ \bar{B}$	m_2
1	1	1	$A\ B$	m_3

Y= $\bar{A}$B + AB

Y= $\Sigma(1,3)$

La forma SOP

- Consideriamo un esempio a tre variabili

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

La forma SOP

- Consideriamo un esempio a tre variabili

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

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

Y= $\Sigma(0,4,5)$

La forma POS (Product-of-Sums)

- Una forma duale per rappresentare una funzione booleana è in forma POS (prodotto di somme).
- Ad ogni riga di una tabella di verità corrisponde un maxtermine che è uguale a 0 solo per quella riga

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y=$A+B$

Y=$A+\overline{B}$

Y=$\overline{A}+$ B

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

La forma POS

- Anche i maxtermini sono enumerati come i mintermini.

A	B	Y	maxterm	maxterm name
0	0	0	$A + B$	M_0
0	1	1	$A + \overline{B}$	M_1
1	0	0	$\overline{A} + B$	M_2
1	1	1	$\overline{A} + \overline{B}$	M_3

- La forma normale POS di una funzione booleana si ottiene come prodotto dei maxtermini per cui la funzione ritorna 0
  - Y=$(A + B) \cdot (\bar{A} + \mathrm{B})$
  - Y=$\prod(0,2)$

La forma POS

- Y=$(A + B) \cdot (\bar{A} + \mathrm{B})$

A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Y=$(A + B) \cdot (\bar{A} + \mathrm{B})$

Y=$A + B$

Y=$\bar{A}$ +
B

Boolean Equations Example

- You are going to the cafeteria for lunch
  - You won't eat lunch ($\overline{E}$)
  - If it's not open ($\overline{O}$) or
  - If they only serve corndogs (C)
- Write a truth table for determining if you will eat lunch (E).

O	C	E
0	0	
0	1	
1	0	
1	1	

Boolean Equations Example

- You are going to the cafeteria for lunch
  - You won't eat lunch ($\overline{E}$)
  - If it's not open ($\overline{O}$) or
  - If they only serve corndogs (C)
- Write a truth table for determining if you will eat lunch (E).

O	C	E
0	0	0
0	1	0
1	0	1
1	1	0

SOP & POS Form

SOP – sum-of-products

O	*C*	*E*	minterm
0	0		$\overline{O}\ \overline{C}$
0	1		$\overline{O}\ C$
1	0		$O\ \overline{C}$
1	1		$O\ C$

POS – product-of-sums

O	*C*	*E*	maxterm
0	0		$O + C$
0	1		$O + \overline{C}$
1	0		$\overline{O} + C$
1	1		$\overline{O} + \overline{C}$

SOP & POS Form

SOP – sum-of-products

O	C	E	minterm
0	0	0	$\overline{O}\ \overline{C}$
0	1	0	$\overline{O}\ C$
1	0	1	$O\ \overline{C}$
1	1	0	$O\ C$

$$E = O\overline{C}$$
$$= \Sigma(2)$$

POS – product-of-sums

O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$E = (O + C)(O + \overline{C})(\overline{O} + \overline{C})$$
$$= \Pi(0, 1, 3)$$

Forme SOP & POS

Se la tabella di verità ha pochi 1 allora la forma SOP è più succinta della forma POS

Se la tabella di verità ha pochi 0 allora la forma POS è più succinta della forma SOP

Nel caso in cui il numero di 0 e 1 è pressappoco lo stesso le due forme si equivalgono