



Università degli Studi di Milano
CdL in Informatica per la comunicazione digitale
AA 2025/2026

Architettura degli elaboratori - Lez 5:

Porte logiche e circuiti combinatori

Marco Tarini
marco.tarini@unimi.it

1



Segnali e informazioni

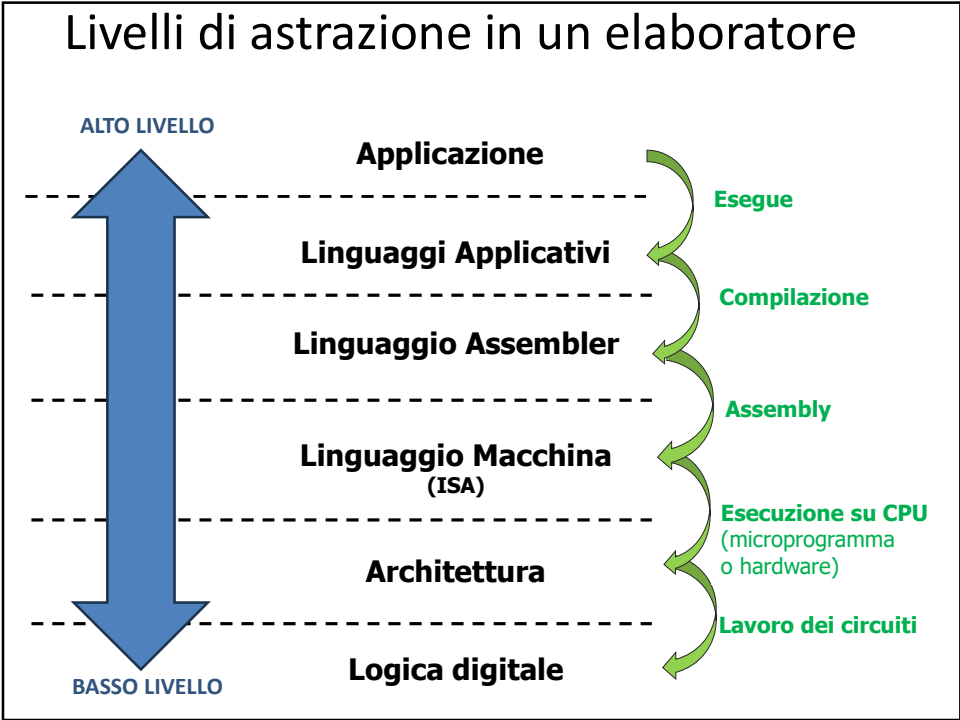
- Per elaborare informazioni, occorre rappresentarle (o codificarle)
- Per rappresentare (o codificare) le informazioni si usano segnali
- I segnali devono essere elaborati, nei modi opportuni, tramite dispositivi di elaborazione.
- Questi dispositivi sono il soggetto di questa parte del corso

Architettura degli elaboratori

- 3 -

Porte logiche

3



4



5



Il segnale binario e circuiti digitali

- **Segnale binario:** una grandezza che può assumere due valori distinti, convenzionalmente indicati con 0 e 1
 - ▶ $s \in \{0, 1\}$
 - ▶ Come abbiamo visto, qualsiasi informazione è rappresentabile (o codificabile) tramite un insieme o una sequenza di segnali binari (per esempio i caratteri del codice ASCII)
- **Circuiti digitali**
 - ▶ L'elaborazione di segnali (o informazioni) binarie è oggi svolta principalmente tramite tecnologie microelettroniche (e in parte anche ottiche)
 - ▶ I circuiti microelettronici che elaborano segnali (o informazioni) binari si chiamano circuiti digitali (o circuiti numerici, o circuiti logici)

Architettura degli elaboratori - 6 - Porte logiche

6



Il segnale binario

- Il segnale binario è adottato per convenienza tecnica
 - ▶ in linea di principio si potrebbe usare un segnale ternario o a n valori
- Implementazione fisica del segnale binario: si usano svariate grandezze fisiche come
 - ▶ corrente elettrica
 - ▶ luminosità
 - ▶ tensione elettrica ← **la più usata!**
cioè potenziale elettrico;
si misura in Volt
tipicamente:
tensione alta = segnale logico 1
tensione bassa = segnale logico 0
(nota: *passa comunque una corrente!*)
 - ▶ altre grandezze fisiche ancora (suono, etc)

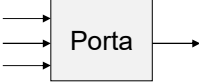
Architettura degli elaboratori - 7 - Porte logiche

7



Porte logiche (*logic gates*)

- Minuscoli dispositivi sono dotati di alcuni cavi («wire») di ingresso, e uno di uscita



- Funzionamento:
 1. dai cavi di ingresso viene immesso un certo segnale binario (di *input*)...
... e, dopo un certo tempo «di commutazione»
(che è brevissimo: si parla di 10^{-10} sec...)
 2. dai cavi di uscita esce un certo altro segnale (elaborato, di *output*)
 - ▶ sia gli input e gli output sono codificati nello stesso modo fisico (esempio con una tensione)
 - ▶ finché il segnale di input resta invariato, neanche l'output cambia
 - ▶ quando il segnale di input cambia, (dopo il tempo di commutazione) il segnale di output cambia (se deve)

Architettura degli elaboratori- 9 -Porte logiche

9



Tipi di porte logiche

- Classificazione delle porte logiche:
per numero di ingressi:
 - ▶ porte a 1 ingresso, (dette anche unarie)
 - ▶ porte a 2 ingressi, (dette anche binarie)
 - ▶ porte a 3 ingressi, (dette anche ternarie)
 - ▶ e così via ...

} casi più comuni

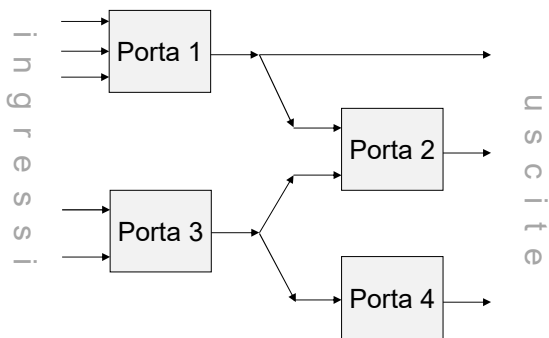
Architettura degli elaboratori- 10 -Porte logiche

10



Circuiti digitali (o reti digitali)

- Collegando gli output di una porta logica agli input di un'altra, e così via, costruiremo dei circuiti logici (circuiti digitali, o reti) che *implementano* funzioni a molti input e molti output ottenendo così elaborazioni via via più complesse



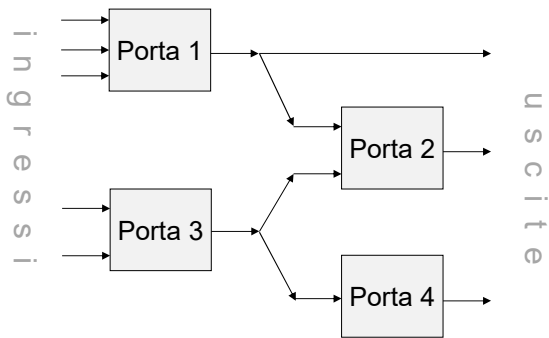
Architettura degli elaboratori - 11 - Porte logiche

11



Circuiti digitali (o reti digitali)

- Un circuito digitale:
 - è dotato di $n \geq 1$ ingressi e di un'uscita
 - è formato da porte logiche interconnesse da cavi
 - l'uscita di una porta è connessa con entrata in una porta o con l'output del circuito

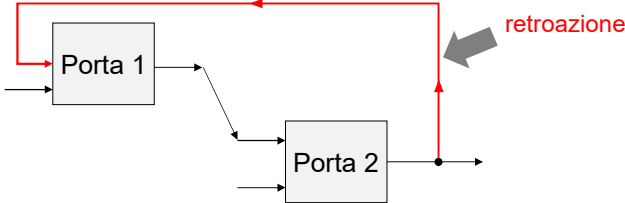


Architettura degli elaboratori - 12 - Porte logiche

12



Circuiti digitali: combinatori VS sequenziali

- Due tipi di circuiti:
 - ▶ **combinatori** ← *per ora, ci concentriamo solo su questi*
senza retroazioni
 - il segnale viaggia dall'input all'output «a senso unico»
 - c'è una gerarchia (un ordinamento parziale) fra le porte
 - niente cicli (aciclico)!
 - ▶ **sequenziali**
con retroazioni
 - **retroazione**: quando il segnale che esce da una porta torna indietro alla stessa porta (anche dopo essere passato da altre porte)
 - esempio:

Architettura degli elaboratori - 14 - Funzioni e circuiti combinatori

14



Vari tipi di porte logiche e loro comportamento

- Useremo porte logiche di vario tipo
 - ▶ (porte «AND», porte «OR», porte «NOT»...)
- Per ogni tipo, stiamo per vedere:
 - ▶ un *simbolo grafico* con cui rappresentarla nei nostri schemi (seguendo in questo delle tradizioni ben consolidate)
 - ▶ come si comporta, cioè cosa produce in output per ogni possibile input
- Come descrivere il comportamento di una porta (a N ingressi)? Useremo una tabella, detta «**tabella delle verità**», che ha:
 - ▶ Una colonna per ciascuno degli N ingressi
 - ▶ Una riga per ogni possibile combinazione di ingressi (sono 2^N righe)
 - ▶ Una colonna con il valore assunto dell'uscita per quegli ingressi

Architettura degli elaboratori - 15 - Porte logiche

15



Quali tipi di porte logiche usare?

Le porte logiche «classiche»

- Useremo (soprattutto) tre tipi di porte logiche:
NOT , AND , OR
- Vantaggi di questa scelta:
 - ▶ Corrispondono a concetti logici molto intuitivi
 - ▶ E' un set con una lunghissima storia di uso nella logica (da Aristotele in poi!). Sono quindi gli operatori «classici»
 - ▶ Come vedremo, è possibile realizzare qualsiasi circuito usando solo porte di questi tre tipi
 - ▶ Sono solo tre tipi: usare pochi tipi di porta logica abbatte i costi
 - ▶ Ma, va detto, non è l'unico «set» di porte (né il più piccolo possibile) con questa proprietà: ne esistono di più piccoli ma ancora sufficienti, come: { NOT, OR } , { NOT, AND } , { NAND } , { NOR }
 - ▶ (vedremo queste porte in seguito) *molto usato in pratica*

Architettura degli elaboratori

- 17 -

Porte logiche

17



Porta AND (a due ingressi)

Simbolo

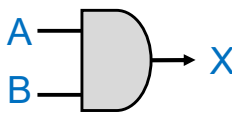


Tabella delle verità

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

L'uscita vale 1 se e solo se entrambi gli ingressi valgono 1

«Se in A viene immesso 1 e in B viene immesso 0, l'uscita assume 0»

Architettura degli elaboratori

- 18 -

Porte logiche

18



Porta NOT (detto invertitore, negatore)

Simbolo

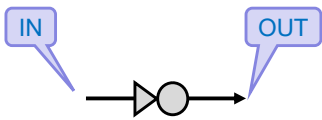


Tabella delle verità

IN	OUT
0	1
1	0

L'uscita vale 1 quando l'ingresso vale 0

L'uscita vale 0 quando l'ingresso vale 1

Architettura degli elaboratori

- 19 -

Porte logiche

19



Porta OR (a due ingressi)

Simbolo

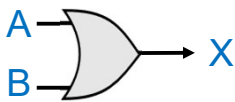


Tabella delle verità

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

L'uscita vale 1 se e solo se almeno un ingresso vale 1

NB: è un "or" inclusivo

Architettura degli elaboratori

- 20 -

Porte logiche

20



Porte a più ingressi

- Le porta AND e la porta OR (e altre) si possono generalizzare a 3, 4, ecc. Ingressi.
- Per es:
 - L'uscita X della porta AND a 3 ingressi vale 1 se e soltanto se tutti e tre gli ingressi A, B e C valgono 1
 - L'uscita X della porta OR a 3 ingressi vale 1 se e soltanto se almeno uno tra gli ingressi A, B e C vale 1
- Tipicamente si usano AND (o OR) a 2, 4 o 8 ingressi (raramente più di 8)

Architettura degli elaboratori

- 21 -

Porte logiche

21



Porta AND a 3 ingressi

Simbolo funzionale

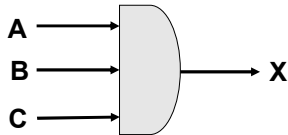


Tabella delle verità

A	B	C	X
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

L'uscita vale 1 se e solo se **tutti** gli ingressi valgono 1

Architettura degli elaboratori

- 22 -

Porte logiche

22



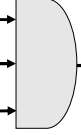
Porte AND e OR a più di 2 ingressi

- La porta AND a 3 ingressi si può realizzare come albero di porte AND a 2 ingressi

A

B

C



X

≡

A

B

C





X

- ma non è l'unico modo, es  (circuito funzionalmente equivalente)
(= esegue la stessa funzione degli input)

A

B

C





X

Architettura degli elaboratori

- 23 -

Porte logiche

23



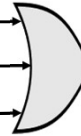
Porta OR a 3 ingressi

Simbolo funzionale

A

B

C



X

Tabella delle verità

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

L'uscita vale 1 se e solo se **almeno uno** degli ingressi vale 1

Architettura degli elaboratori

- 24 -

Porte logiche

24



Come si realizza una porta logica

- Questo tipo di domanda pertiene al **livello dei dispositivi**
 - esula da questo corso (noi partiamo dal **livello logico**)
 - (ci limitiamo qui ad alcune considerazioni di massima)

Livello logico

Livello della elettronica

- Una porta logica è realizzata con un certo numero di **transistor** e **diodi**
- Il numero di questi componenti determina il costo della porta
- Dipende da: tecnologia utilizzata, funzione logica implementata, etc
- Come naturale, maggiore il numero di ingressi, maggiore il costo
 - Linea di massima:
da 2 a 5 componenti per le nostre porte fino a 2 input

Architettura degli elaboratori

- 25 -

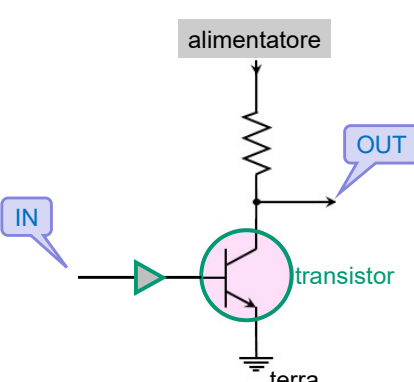
Porte logiche

25

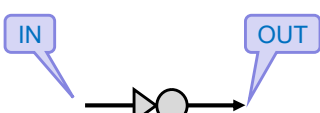


Esempio: porta NOT

- A basso livello:



- A livello più alto:
(quello che useremo noi)



IN	OUT
0	1
1	0

Architettura degli elaboratori

- 29 -

Porte logiche

29

Marco Tarini
Università degli Studi di Milano

11

Esempio: porta NAND – Not AND
(una porta che non abbiamo ancora visto)

- A basso livello:
- A livello più alto:
(quello che useremo noi)

A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	0

Architettura degli elaboratori - 30 - Porte logiche

30

Circuit Manufacturing:
una tecnologia molto avanzata

...ma ora torniamo su alcuni livelli di astrazione
(al livello delle porte logiche e sopra)

Architettura degli elaboratori - 35 - Porte logiche

35



Nota lessicale: «Algebra»

- In termini generali, un'**algebra** è una struttura matematica definita da:
 - ▶ Uno o più insiemi di valori possibili
 - ▶ Operazioni possibili che posso fare sugli elementi di questo insieme (che godono di alcune proprietà, eccetera)
- Un'**espressione**, in una data algebra, è una combinazione di **operatori** che operano su **variabili** (simboli che possono assumere qualsiasi valore) e **costanti** (valori predeterminati e fissi) e parentesi per determinare l'ordine di esecuzioni degli operatori
- Per esempio, nella familiare algebra dei numeri reali, ho espressioni come

$$(5x + 12) \cdot y$$

- Dove x e y sono le variabili, 5 e 12 le costanti, + e $\cdot$ gli operatori



Algebra di Boole (o Booleana)

- Insieme = $\{0, 1\}$ (cioè $\{\text{Falso}, \text{Vero}\}$)
- Operatori = AND (prodotto logico),
OR (somma logica),
NOT (inversione logica)
- Una **espressione booleana** contiene una o più
 - ▶ variabili booleane,
 - ▶ operatori booleani prodotto (AND), somma (OR) e negazione (NOT):
 - ▶ Costanti (0 o 1)
- esempio: $A \cdot B + C$
cioè: $((\text{NOT } A) \text{ AND } B) \text{ OR } C$
- Useremo le espressioni dell'algebra di Boole per descrivere matematicamente i circuiti combinatori
- Ogni **espressione booleana** (espressioni nell'algebra di Boole) corrisponde ad un **circuito combinatorio**, e viceversa!



Operatori booleani classici: sintassi alternative comunemente usate

OR	AND	NOT
$A + B$	$A \times B$	$/ A$
$A B$	$A \&\& B$	$! A$
$A B$	$A \& B$	$\sim A$
$A \text{ or } B$	$A \text{ and } B$	$\text{not } A$
$A \vee B$	$A \wedge B$	$\neg A$
	$A \cdot B$	$\overline{A}$
	$A * B$	
	$A B \leftarrow \text{nota!}$	

dal latino «vel»

Architettura degli elaboratori- 38 -Porte logiche

38



Operatori booleani classici: analogia con gli operatori numerici

OR	AND	NOT
(op.binario)	(op.binario)	(op.unario)
$0 + 0 = 0$	$0 \cdot 0 = 0$	$/0 = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$/1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	



- OR somiglia molto ad una somma (fra numeri 0 e 1) (*eccetto quel caso...*)
- AND si comporta proprio come un prodotto (fra numeri 0 e 1)
- per questo sono chiamati anche **Somme** e **Prodotti** logici, denotati da + e ·
- NOT si comporta invece come il complemento: $\text{NOT}(X) = 1 - X$

Architettura degli elaboratori- 39 -Porte logiche

39



Precedenza degli operatori nelle espressioni (analoghe a quelli numerici)

Il simbolo del prodotto logico si può omettere:

Cioè, se scrivo $A B$ si intende $A \cdot B$

Il prodotto «ha precedenza» sulla somma,
e l'inversione «ha precedenza» su somma e prodotto

Cioè, se scrivo: $A + /B C$ si intende $A + ((/B) C)$
prima si esegue l'inversione di B,
poi il suo prodotto con C,
e infine la somma con A

E' possibile imporre un ordine diverso da quello determinato
dalla precedenza degli operatori, usando le parentesi:

Per es, scrivendo: $(A + /B) C$
si intende che prima avviene l'inversione di B, poi la somma con A, e
infine il prodotto con C

Architettura degli elaboratori

- 41 -

Porte logiche

41



Precedenza degli operatori nelle espressioni (analoghe a quelli numerici)

Mini esercizi:

se A e B valgono entrambi 0
quanto valgono le espressioni $/AB$ e $/(AB)$?

se A e B valgono entrambi 1
quanto valgono le espressioni $/A+B$ e $/(A+B)$?

Architettura degli elaboratori

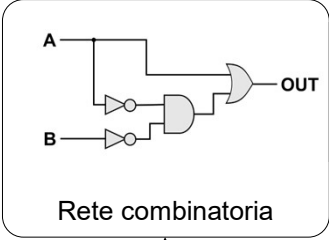
- 42 -

Porte logiche

42



Ogni espressione booleana
è una rete combinatoria, e viceversa



Rete combinatoria

$$A + A/B$$

Espressione booleana

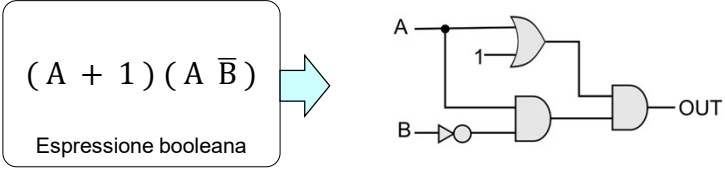
- Gli *ingressi* della rete sono le *variabili* dell'espressione
- Ogni *operatore* della espressione è una *porta logica* della rete
- L'*ingresso* della rete sono i valori assegnati alle *variabili* dell'espressione
- L'*uscita* della rete è il valore assunto dall'espressione (per quell'assegnamento delle variabili)
- Ogni wire della rete contiene il valore di una *sottoespressione* (per es, quale cavo contiene la sottoespressione *A/B* ?)

Architettura degli elaboratori - 43 - Funzioni e circuiti combinatori

43



Esempio ed Esercizi



Espressione booleana

- Disegna il circuito che corrisponde alle espressione logiche:

$(A+B) (B+C)$
 $A + (B C)$
 $(1 + A) \cdot C + (D \cdot B)$

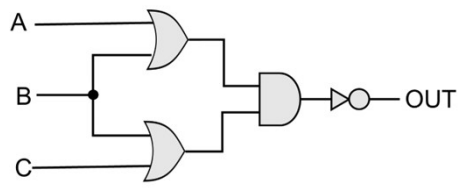
Architettura degli elaboratori - 44 - Porte logiche

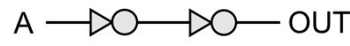
44



Esercizi

- Riporta le espressioni booleane corrispondenti ai circuiti:





Architettura degli elaboratori
- 45 -
Porte logiche

45



Nota lessicale: funzione logica o funzione booleana

- Come si sa, una funzione **numerica** è una funzione che, dato uno (o più) numeri, restituisce uno (o più) numeri
 - ▶ Per es, una funzione da naturali a naturali $f: \mathbb{N} \rightarrow \mathbb{N}$
 - ▶ Come la funzione «doppio», $f(x) = 2x$, dove per esempio $f(4) = 8$ e $f(15) = 30$
 - ▶ Una funzione può avere ha più di un argomento (è allora «multivariata») Come la funzione «somma», $f(x, y) = x + y$, dove $f(4, 6) = 10$ che «va» da due naturali ad un naturale, quindi $f: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$
 - ▶ Una funzione anche può restituire più oggetti in uscita, Come la funzione «quoziente e resto»
 $f: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$
 definita da $f(x, y) = (x/y, x \% y)$.
 per esempio, $f(44, 6) = (7, 2)$

Prodotto cartesiano, cioè insieme delle coppie

- Una funzione **logica** (o **booleana**) è esattamente lo stesso concetto, ma lavora, sia in ingresso che in uscita, su valori «booleani» cioè l'insieme $\{0, 1\}$ o $\{\text{falso}, \text{vero}\}$, invece che su numeri

Architettura degli elaboratori
- 46 -
Porte logiche

46



Tabella (o tavola) di verità

- La **tabella di verità** è un modo per descrivere una **funzione booleana**
 - Nota che non sarebbe altrettanto facile usare questo metodo per le funzioni numeriche, perché ci sarebbero infiniti numeri da «tabellare»
- La tabella di verità di una funzione a N variabili è costituita da
 - N colonne degli ingressi,
 - la colonna di uscita (o più di una, se la funzione restituisce più valori) che riporta i valori assunti dalla funzione
 - 2^N righe, una delle per ciascuna delle possibili combinazioni di valori degli ingressi
- Nota: se due funzioni hanno la stessa tabella di verità, allora sono la **stessa** funzione!
 - Ogni **tabella di verità** corrisponde ad una **funzione booleana**
 - E viceversa
 - Tabella di verità** è in pratica *sinonimo* di **funzione booleana**

Architettura degli elaboratori

- 47 -

Funzioni e circuiti combinatori

47



Ogni espressione booleana esprime una funzione booleana

- Attribuendo valori in $\{0,1\}$ a ciascuna delle variabili booleane di un'espressione, e calcolando il suo valore, si ottiene un valore risultante (0 oppure 1)
- Una data espressione booleana (ad N variabili) corrisponde una **funzione booleana** con un'uscita
 - cioè una funzione che va
 - da: N booleani (gli argomenti della funzione)
 - a: un booleano (il valore restituito della funzione)

Architettura degli elaboratori

- 48 -

Funzioni e circuiti combinatori

48



Trovare la funzione booleana di una espressione logica

- Esempio: espressione logica: $(A + B) (C + \neg B)$
- Tre variabili (A, B, C), quindi $2^3 = 8$ righe
- Passo 1: elenchiamo tutte le combinazioni di input possibili

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

Nota: è in pratica la lista dei numeri in binario a N cifre, elencati, in ordine, dal primo (0...0) all'ultimo (1...1)

Architettura degli elaboratori

- 49 -

Porte logiche

49



Trovare la funzione booleana di una espressione logica

- Esempio: espressione logica: $(A + B) (C + \neg B)$
- Passo 2: calcoliamo una a una tutte le sotto espressioni...

A	B	C	A+B	$\neg B$	$(C + \neg B)$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	0	1

Architettura degli elaboratori

- 50 -

Porte logiche

50



Trovare la funzione booleana di una espressione logica

- Esempio: espressione logica: $(A + B) (C + \neg B)$
- Passo 3: fino a quella finale...

A	B	C	$(A+B)$	$\neg B$	$(C+\neg B)$	$(A+B) (C+\neg B)$
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	1	0	0	0
0	1	1	1	0	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	0	0	0
1	1	1	1	0	1	1

Architettura degli elaboratori

- 51 -

Porte logiche

51



Trovare la funzione booleana di una espressione logica

- La funzione booleana calcolata dall'espressione logica $(A + B) (C + \neg B)$ è dunque:

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- E' l'unica espressione logica ad avere questa tabella di verità?
- (e dunque questa funzione booleana)?
- Certamente NO!
- Espressioni booleane diverse possono avere la stessa funzione logica!
- Si dicono allora «espressioni equivalenti»

Architettura degli elaboratori

- 52 -

Porte logiche

52



Dal rete combinatoria alla tabella delle verità

- Data una rete combinatoria, qual è la funzione calcolata?
 - ▶ Possiamo tradurre la rete in una espressione, e poi calcolare la tabella della verità.
 - ▶ Oppure, la tabella delle verità può essere ricavata per simulazione del funzionamento circuitale della rete combinatoria stessa
- Per simulare il funzionamento di una rete combinatoria, si applicano dei valori agli ingressi, e li si propaga lungo la rete fino all'uscita
 - ▶ si applica ogni possibile combinazioni di 0 e 1 in ingresso
 - ▶ in ciascun caso, si computa l'uscita

Rete combinatoria

→

Tabella verità della funzione

Simulazione (circuitale)

Architettura degli elaboratori - 53 - Funzioni e circuiti combinatori

53



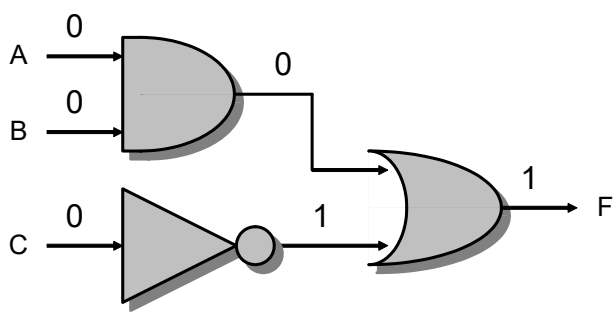
Simulazione circuitale (esempio)

Passo 1/8: input 0,0,0

A 0

B 0

C 0



F

Risultato della simulazione: $F(0, 0, 0) = 1$

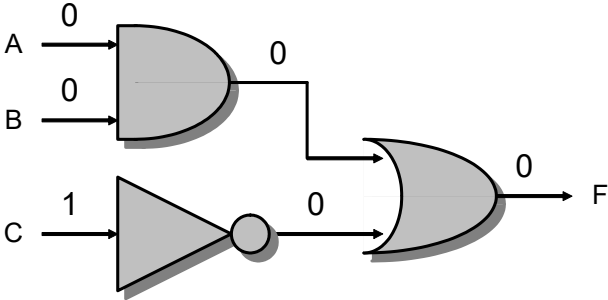
Architettura degli elaboratori - 54 - Funzioni e circuiti combinatori

54



Simulazione circuitale (esempio)

Passo 2/8: input 0,0,1



Risultato della simulazione: $F(0, 0, 1) = 0$

Architettura degli elaboratori

- 55 -

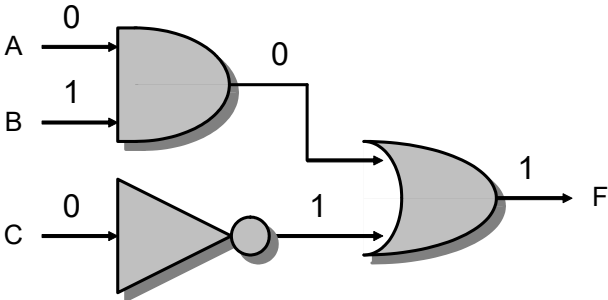
Funzioni e circuiti combinatori

55



Simulazione circuitale (esempio)

Passo 3/8: input 0,1,0



Risultato della simulazione: $F(0, 1, 0) = 1$

Architettura degli elaboratori

- 56 -

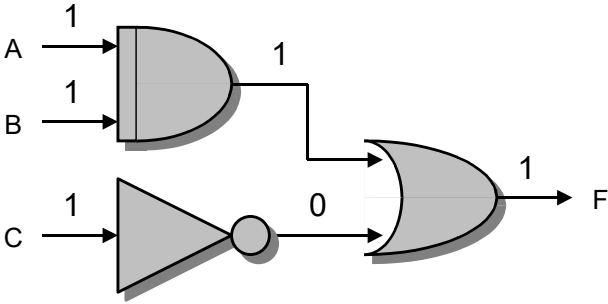
Funzioni e circuiti combinatori

56



Simulazione circuitale (esempio)

Passo 8/8: input 1,1,1



Risultato della simulazione: $F(1, 1, 1) = 1$

Architettura degli elaboratori

- 57 -

Funzioni e circuiti combinatori

57



Simulazione circuitale: risultato (dopo tutte le simulazioni)

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

Architettura degli elaboratori

- 58 -

Funzioni e circuiti combinatori

58



Reti combinatorie, espressioni booleane, e funzioni booleane

- Ogni espressione booleana corrisponde ad un circuito combinatorio
- E viceversa
- Ogni espressione booleana *calcola* una funzione booleana
- Ogni circuito combinatorio *implementa* una funzione booleana
- Una stessa funzione booleana può essere calcolata da infiniti circuiti combinatori diversi (equivalenti), corrispondenti ad altrettante espressioni booleane (equivalenti)
- Questi circuiti, pur equivalenti fra loro (stesso comportamento in termini di input e output) possono avere numero di porte, costi, velocità, consumi, molto diversi!
- Idea per ottimizzare un circuito: usando l'algebra delle espressioni booleane, *riscriviamo un'espressione booleana in una equivalente* ma che corrisponde (si spera) ad un circuito di costo minore
- Vediamo dunque alcune regole per riscrivere una espressione in una equivalente

Architettura degli elaboratori

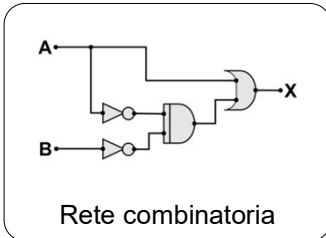
- 59 -

Funzioni e circuiti combinatori

59



Ricapitolando



Rete combinatoria

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

Espressione booleana

A	B	X
0	0	1
0	1	0
1	0	1
1	1	1

Tabella verità
cioè
funzione booleana

$\infty:1$

$1:1$

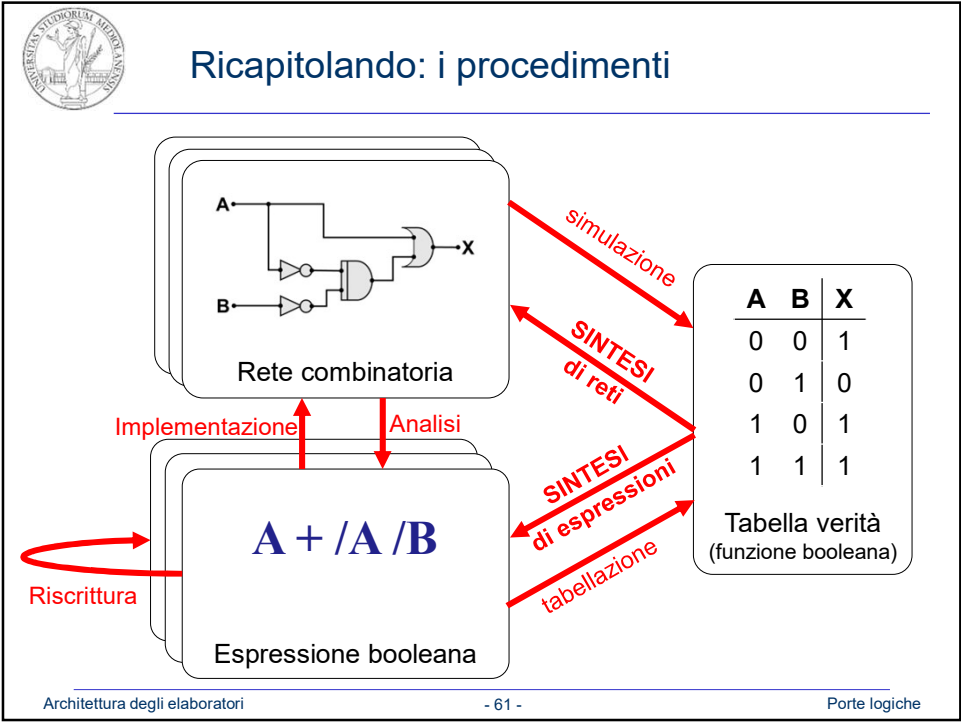
$\infty:1$

Architettura degli elaboratori

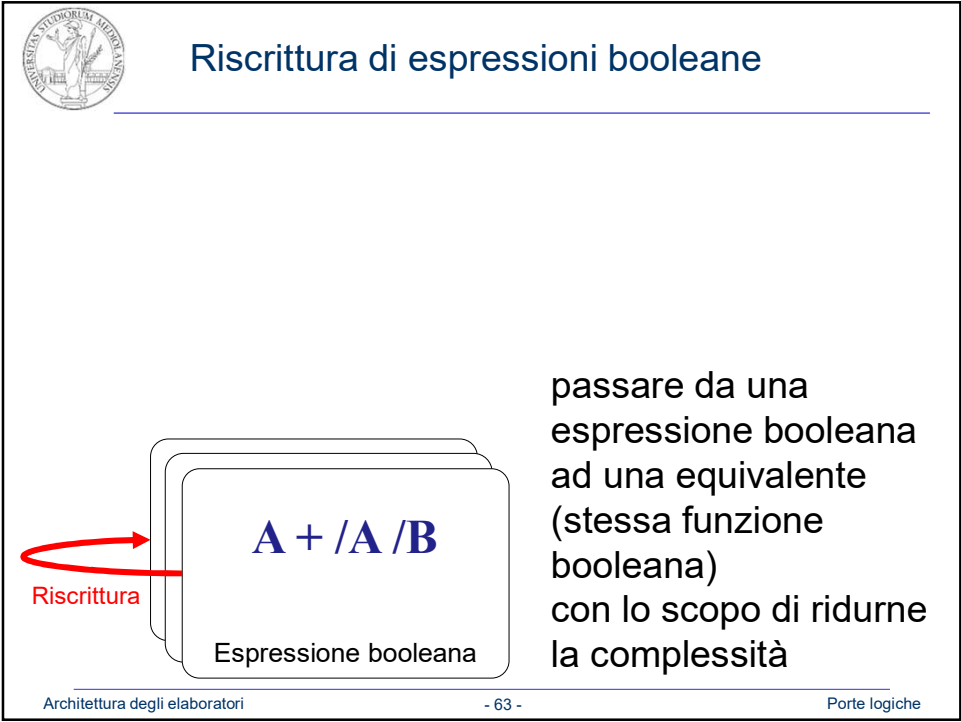
- 60 -

Porte logiche

60



61



63



Regole di trasformazione delle espressioni booleane

Legge	con AND	con OR (duale)
Identità	$1 \cdot A = A$	$0 + A = A$
Elemento nullo	$0 \cdot A = 0$	$1 + A = 1$
Idempotenza	$A \cdot A = A$	$A + A = A$
Inverso	$A \cdot \neg A = 0$	$A + \neg A = 1$
Commutativa	$A \cdot B = B \cdot A$	$A + B = B + A$
Associativa	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$
Distributiva	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
Assorbimento	$A \cdot (A + B) = A$	$A + A \cdot B = A$
De Morgan	$\neg(A \cdot B) = \neg A + \neg B$	$\neg(A + B) = \neg A \cdot \neg B$
Tertium non datur	$\neg \neg A = A$	

Architettura degli elaboratori- 64 -Porte logiche

64



Regole di trasformazione: note

- Le regole di trasformazione (o «di riscrittura») ci consentono di passare da una espressione ad un'altra, **equivalente**.
 - lobiettivo delle riscritture: ottimizzare l'espressione di partenza
 - cioè: rendere il circuito associato più economico, o più veloce, etc
- Gli A, B nelle regole rappresentano sotto-espressioni qualsiasi
 - non necessariamente variabili. es:
Idempotenza: $(A(B+C) + A(B+C)) = A(B+C)$
- Le regole sono a coppie (una per l'AND una per l'OR)
 - una è la regola **DUALE** dell'altra (principio di dualità)
 - cioè una è ottenuta dall'altra scambiando fra di loro:
 $AND \iff OR \ \& \ 0 \iff 1$
- Ciascuna regola si può usare in un verso, o nel verso opposto
 - $XXX = YYY \implies$ posso passare da XXX a YYY... oppure viceversa
- Solo alcune delle regole sono equivalenti a quelle familiari dell'algebra numerica (per esempio, la commutatività, o una delle due distributive)

Architettura degli elaboratori- 65 -Porte logiche

65



De Morgan explained

- Nel primale:

$$\neg(A \cdot B) = \neg A + \neg B$$

 - ▶ negare che «sia A che B sono entrambi veri»
significa
 affermare che o uno o l'altro (o entrambi) sia falso
- Nel duale:

$$\neg(A + B) = \neg A \cdot \neg B$$

 - ▶ negare che «o vale A o vale B (o entrambi)»
significa
 affermare che siano entrambi falsi

Architettura degli elaboratori
- 66 -
Porte logiche

66



Ulteriori esercizi

1. Costruisci la tabella di verità per l'espressione
 $AB + ABC$
2. Determinare se l'espressione al punto 1 sia o no equivalente all'espressione
 $AB + C$ (compara le due tabelle di verità!)
3. Dimostra, usando la tabella di verità, la validità della seguente identità
 (cioè l'equivalenza delle due espressioni):
 $\neg(ABC) = \neg A + \neg B + \neg C$
4. Vero o falso: queste sono riscritture valide dell'equivalenza al punto 3

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

$$\sim (A \wedge B \wedge C) = \sim A \vee \sim B \vee \sim C$$

$$\text{NOT}(A \text{ AND } B \text{ AND } C) = (\text{NOT } A) \text{ OR } (\text{NOT } B) \text{ OR } (\text{NOT } C)$$

$$\neg(A \ \&\& \ B \ \&\& \ C) == (\neg A \ || \ \neg B \ || \ \neg C)$$

(l'ultima è nel linguaggio di programmazione C)
5. Prova (a fantasia) a definire espressioni, con due variabili A e B, che valgano sempre 0, o sempre 1, con qualsiasi valore assegnato ad A o B
6. Prova a semplificare questa espressione booleana, usando le regole viste:

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

Architettura degli elaboratori
- 67 -
Porte logiche

67