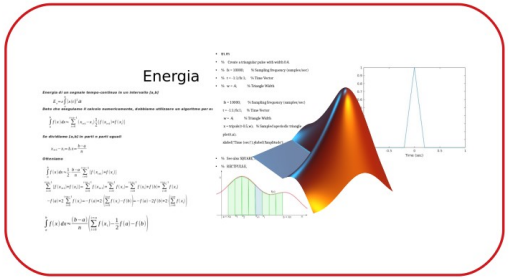
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Programmazione Informatica

Ingegneria Civile e Ambientale
Ingegneria Navale

Lezione Teoria 3. Algebra booleana, introduzione e applicazioni nella programmazione



Antonio Fiumara
antonio.fiumara@dia.units.it

A. A. 2025-26

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Struttura del Corso

Gli argomenti trattati nel corso saranno:

- Lezione Teoria 1 – Introduzione, presentazione del corso, codifica dell'informazione
- Lezione Teoria 2 – Rappresentazione dell'informazione in un computer
- **Lezione Teoria 3 – Algebra booleana, introduzione e applicazioni nella programmazione**
- Lezione Teoria 4 – Architettura dei calcolatori. HW, SW, Sistemi Operativi
- Lezione Teoria 5 – Algoritmi e strutture dati
- Lezione Teoria 6 – Implementazione di algoritmi notevoli (ordinamento , ricerca, ecc)
- Lezione Programmazione - Matlab 1 – Introduzione, installazione, comandi di base, matrici
- Lezione Programmazione - Matlab 2 – Input/Output (I/O), strutture di controllo e funzioni
- Lezione Programmazione - Matlab 3 – Sistemi Lineari e Algebra lineare
- Lezione Programmazione - Matlab 4 – Analisi matematica
- Lezione Programmazione - Matlab 5 – Funzioni nel dominio del tempo e della frequenza
- Lezione Programmazione - Matlab 6 – Analisi ed elaborazione dei segnali
- Lezione Programmazione - Matlab 7 – Ripasso e approfondimenti
- Esonero esame Programmazione Matlab (3/12/25)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

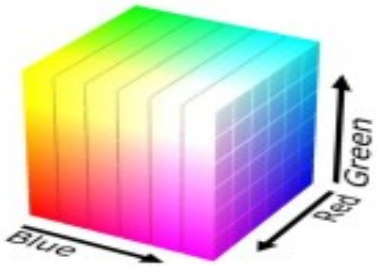
Introduzione

- Immagini a colori
- modelli di colore RGB, CMYK, HSV, HSL, LAB
- Immagini vettoriali
- Algebra booleana

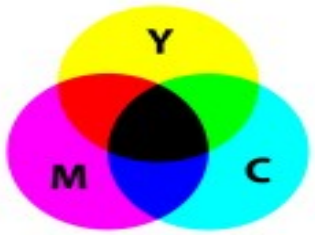
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Immagini a colori

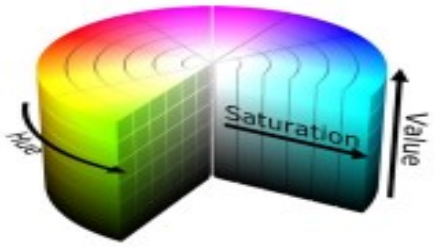
RGB (Red Green Blue): modello *additivo* dove le emissioni luminose dei tre colori principali rosso, verde e blu vengono sommate. Il bianco è quindi ottenuto dalla somma di tutti e tre i colori R, G e B. Utilizzato nei display di dispositivi elettronici come TV, computer, telefoni cellulari, ecc.



CMY[K] (Cyan, Magenta, Yellow, [Black]): modello *sottrattivo* dove i tre colori principali ciano, magenta e giallo vengono miscelati (pensando a vernici) o sovrapposti (pensando a lucidi colorati). È sottrattivo perchè ogni colore aggiunto sottrae una parte di spettro alla luce biance incidente: il nero è ottenuto dalla composizione di tutti e tre i colori C, M e Y. L'eventuale quarto parametro K (nero) è utilizzato per motivi di convenienza pratica in fase di stampa.



HSV (Hue, Saturation, Value) o l'equivalente HSL (Hue, Saturation, Lightness): modello nel quale un colore è definito da tonalità (hue), saturazione (saturation) ed un terzo parametro che può essere brillanza (brightness, value) oppure tono (lightness). Non è nè additivo nè sottrattivo.



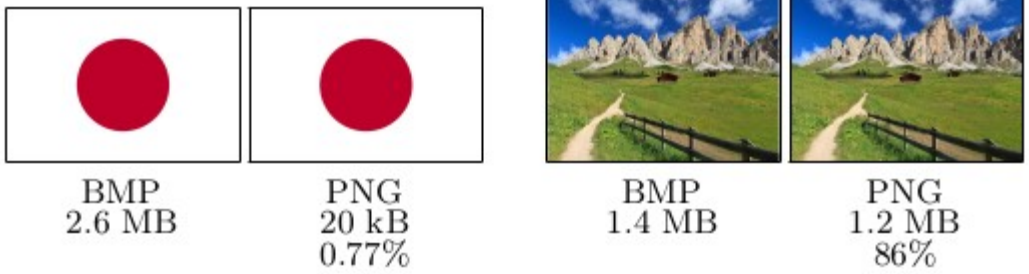
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Immagini a colori

- Il modello RGB è quello più utilizzato per la rappresentazione digitale delle immagini a colori.
- Indipendentemente dal modello di colore impiegato, è sempre necessario fornire una terna di numeri per definire un colore: (R, G, B), (C, M, Y) oppure (H, S, V), o una quaterna di numeri nel caso di modello di colore CMYK.
- E' possibile passare da un modello di colore ad un altro mediante opportune funzioni di trasformazione.
- L'informazione di un'immagine a colori può essere quindi rappresentata da tre (o quattro) immagini, o canali, in scala di grigi.
- Ogni canale ha la medesima risoluzione ed la medesima "profondità", cioè il numero k di bit utilizzati per ogni pixel per ogni canale è lo stesso.
- Un'immagine $m \times n$ con 2^k livelli per ciascun canale è quindi rappresentata da una sequenza di $3 \cdot m \cdot n \cdot k$ bit.
- La profondità del colore, cioè il numero di bit utilizzati per il colore di ogni pixel, è quindi $3 \cdot k$. La scelta più popolare, nota come profondità Truecolor, è quella di usare 8 bit per canale, per un totale di 24 bit (3 byte): si hanno a disposizione 2^{24} = 16'777'216 colori diversi.

Immagini a colori

- La memorizzazione diretta di immagini ad alta risoluzione richiede molto spazio: ad es. un'immagine con risoluzione 3840 × 2160 (4K UHD) conm24 bit di profondità di colore occuperà 3840 · 2160 · 24 bit ≈ 25 MB.
- Esistono varie tecniche di compressione per ridurre lo spazio richiesto per la memorizzazione, con e senza perdita d'informazioni.
- Compressione senza perdita di informazioni o lossless: l'informazione originale può essere completamente ricostruita a partire dai dati compressi. Nelle immagini in formato PNG, per esempio, viene utilizzata una tecnica di compressione lossless che si basa sulla ricerca di sequenze ripetute di dati. Funziona quindi bene su immagini con ampie aree dello stesso colore, meno bene su immagini generiche, ad esempio fotografie.



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Immagini a colori

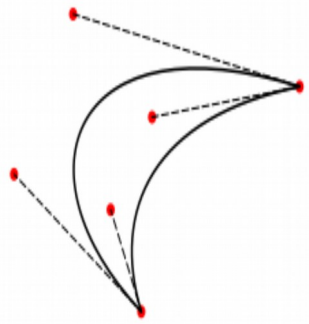
- Compressione con perdita di informazioni o lossy: parte dell'informazione originale viene persa irreversibilmente nella compressione. Nelle immagini in formato JPG, per esempio, viene utilizzata una tecnica di compressione lossy che si basa su una ri-quantizzazione delle componenti in frequenza di ciascun canale, preventivamente trasformato secondo un particolare modello di colore (luminanza e crominanza). L'accuratezza della quantizzazione, dalla quale dipende l'errore irreversibile di quantizzazione, può essere variata in funzione di un parametro di qualità Q, solitamente compreso tra 0 e 100.
- Risorse su internet per approfondimenti

https://www.gianlucatramontana.it/2017/05/29/teoria-del-colore-i-modelli-rgb-cmyk-hsb-e-lab/

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Immagini vettoriali

- Immagine vettoriale: è definita dalle proprietà di entità geometriche definite a priori (linee, segmenti, poligoni, cerchi, curve, caratteri con determinati font, ecc.).
- Un'immagine vettoriale può essere ridotta e/o ingrandita a piacere senza alcuna perdita di dettaglio. Viene tipicamente usata da chi si occupa di grafica.
- Formati di file più utilizzati:
  - Portable Document Format (PDF),
  - Encapsulated PostScript (EPS),
  - Scalable Vector Graphics (SVG)
  - AutoCAD Drawing Exchange Format (DXF), ecc.



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Algebra di Boole – curiosità storiche

George Boole fu matematico e logico inglese, nato nel 1815. Egli sviluppò l'algebra booleana nel suo libro "An Investigation of the Laws of Thought" (1854).

Sebbene l'obiettivo principale di Boole fosse sviluppare un sistema che modellasse i processi del ragionamento umano attraverso un formalismo matematico, l’algebra booleana si rivelò essenziale per la nascita dell'informatica moderna e della progettazione dei circuiti digitali.

L’algebra booleane oggi viene usata per progettare e ottimizzare circuiti logici, basi dati e sistemi informatici. Ogni componente di un computer, dal processore alla memoria, utilizza operazioni booleane per funzionare.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori logici

Operatori booleani, detti anche operatori logici, NOT, AND, OR, XOR

NOT: negazione logica, “inverte” il valore $\bar{a}$

not(1) = 0, *not*(0)=1

AND: a *and* b = 1 solo se a=1 e b=1 $a \cdot b$

OR: a *or* b = 1 se a=1 oppure b=1 $a + b$

XOR: a x*or* b = 1 se a è diverso da b $a \oplus b$

Se il numero è costituito da più bit, gli operatori booleani si applicano a ciascun bit

Esempio:

*not(*0100 1100)=1011 0011

(1110 0110) *and (*0111 1111)=0110 0110

(1110 0110) *or (*0111 1111)=0111 1111

(1110 0110) x*or (*0111 1111)=1001 1001

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Variabili booleane o logiche

Una **Variabile** Booleana $x \in B = \{0, 1\}$ può assumere solo due valori:

- 0 (Falso)
- 1 (Vero)

A differenza dell'algebra tradizionale, dove le variabili possono assumere una gamma infinita di valori, nell'algebra booleana tutto è binario, limitato a due possibili stati.

Se x è una variabile booleana, vale la seguente definizione formale:

$$x = 0 \Rightarrow x \neq 1$$
$$x = 1 \Rightarrow x \neq 0$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori logici fondamentali – NOT, AND, OR

NOT, negazione logica

Sia a una variabile booleana

$$not(a) = \bar{a}$$

$$a = 0 \Rightarrow \bar{a} = 1$$
$$a = 1 \Rightarrow \bar{a} = 0$$

a	$\bar{a}$
0	1
1	0

Tabella della verità

Proprietà

not(not(x)) = x

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori logici fondamentali – NOT, AND, OR

AND, prodotto logico

$a \text{ and } b = a \wedge b = a \cdot b$

a *and* b = 1 solo se a=1 e b=1

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

Tabella della verità

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori logici fondamentali – NOT, AND, OR

OR, somma logica

$a \text{ or } b = a \vee b = a + b$

a *or* b = 0 solo se a=0 e b=0

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	1

Tabella della verità

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori logici – bit level

Posso applicare un operatore booleano ad un numero costituito da più bit. In tal caso, gli operatori booleani si applicano a ciascun bit

Esempio:

*not(*0100 1100)=1011 0011

(1110 0110) *and (*0111 1111)=0110 0110

(1110 0110) *or (*0111 1111)=0111 1111

(1110 0110) x*or (*0111 1111)=1001 1001

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Algebra booleana - Proprietà

Proprietà commutativa:

$$x \cdot y = y \cdot x$$
$$x + y = y + x$$

Proprietà associativa:

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$
$$x + (y + z) = (x + y) + z$$

Proprietà distributiva:

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$
$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Algebra booleana - Proprietà

Proprietà di identità:

0 è elemento neutro rispetto a OR

$$x + 0 = x$$

1 è elemento neutro rispetto a AND

$$x \cdot 1 = x$$

Proprietà complementare:

$$x + \overline{x} = 1$$
$$x \cdot \overline{x} = 0$$

Legge di assorbimento:

$$x + x \cdot y = x$$
$$x \cdot x + y = x$$

Leggi di de Morgan:

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$
$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Algebra booleana - Teoremi

Doppia negazione:

$$\overline{\overline{x}} = x$$

Idempotenza:

$$x \cdot x = x$$

$$x + x = x$$

Annichilimento:

$$x + 1 = 1$$

$$x \cdot 0 = 0$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XOR

XOR, OR esclusivo

È un operaore logico di largo impiego

$a \ xor \ b = a \oplus b$

$a \ xor \ b = 1 \quad solo\ se \quad a \neq b$

$a \ xor \ b = 0 \quad solo\ se \quad a = b$

$a \ xor \ b = a \cdot \overline{b} + \overline{a} \cdot b$

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Tabella della verità

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XOR

Osservazioni:

L'operatore xor può essere utilizzato per calcolare la somma tra numeri binari (verificare per esercizio).

Può essere visto anche come operatore di uguaglianza, o disuguaglianza.

$$a \ xor \ b = 1 \quad solo\ se \quad a \neq b$$
$$a \ xor \ b = 0 \quad solo\ se \quad a = b$$

Inoltre è interessante notare che, l'operatore xor può essere visto come un operatore not condizionato

$$se \quad a = 0 \quad \Rightarrow \quad a \ xor \ b = b$$
$$se \quad a = 1 \quad \Rightarrow \quad a \ xor \ b = \overline{b}$$

In altre parole, l'operatore xor può essere visto come un operatore unario (agisce sulla variabile b, nel nostro esempio) che restituisce il complemento della variabile d'ingresso b se la variabile di controllo a = 1, invece restituisce il valore inalterato della variabile d'ingesso b se la variabile di controllo a = 0.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

a	b	$a \oplus b$
0	b	b
1	b	$\overline{b}$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Funzioni Booleane

Funzione Booleana f di n variabili Booleane $x_1, \ldots, x_n$:

$$f : B^n \rightarrow B, \qquad B = \{0,1\}$$

$$(x_1, \ldots, x_n) \rightarrow f(x_1, \ldots, x_n)$$

Le funzioni Booleane si possono esprimere algebricamente tramite operatori logici (or, and, not) applicati alle variabili Booleane ed alle costanti 0 e 1.
Ad esempio f (x, y, z) = $\bar{x} \cdot y + z$.

Essendo *B* un insieme finito, è possibile definire una funzione booleana anche per mezzo di una tabella di verità che riporta il valore di f per ogni combinazione dei valori binari delle variabili $x_1, \ldots, x_n$:

x_1	$\cdots$	x_n	$f(x_1, \ldots, x_n)$
0	0	0	$f(0, \ldots, 0)$
0	0	1	$f(0, \ldots, 1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	1	$f(1, \ldots, 1)$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Funzioni Booleane

Ad esempio $f(x, y, z) = \bar{x} \cdot y + z$.

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Funzioni Booleane

Oltre alle funzioni OR e AND già viste, assumono particolare interesse pratico le seguenti funzioni, simmetriche rispetto ai due ingressi x e y:

NAND (AND negato): $\overline{x \cdot y} = \overline{x} + \overline{y}$

NOR (OR negato): $\overline{x + y} = \overline{x} \cdot \overline{y}$

XNOR (XOR negato): $\overline{x \oplus y} = (x + \overline{y}) \cdot (\overline{x} + y)$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Verificare le seguenti identità

NAND (AND negato): $\overline{x \cdot y} = \overline{x} + \overline{y}$

NOR (OR negato): $\overline{x + y} = \overline{x} \cdot \overline{y}$

XNOR (XOR negato): $\overline{x \oplus y} = (x + \overline{y}) \cdot (\overline{x} + y)$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

NAND (AND negato): $\overline{x \cdot y} = \overline{x} + \overline{y}$

x	y	$\overline{x \cdot y}$	$\overline{x} + \overline{y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

NOR (OR negato): $\overline{x+y}=\overline{x}\cdot\overline{y}$

x	y	$\overline{x+y}$	$\overline{x}\cdot\overline{y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

XNOR (XOR negato):

$$\overline{x \oplus y} = \overline{\overline{x} \cdot y + x \cdot \overline{y}}$$

$$\overline{x \oplus y} = \overline{\overline{x} \cdot y + x \cdot \overline{y}} = \overline{\overline{x} \cdot y} \cdot \overline{x \cdot \overline{y}}$$

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

x	y	$\overline{x \oplus y}$	$(x + \overline{y}) \cdot (\overline{x} + y)$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

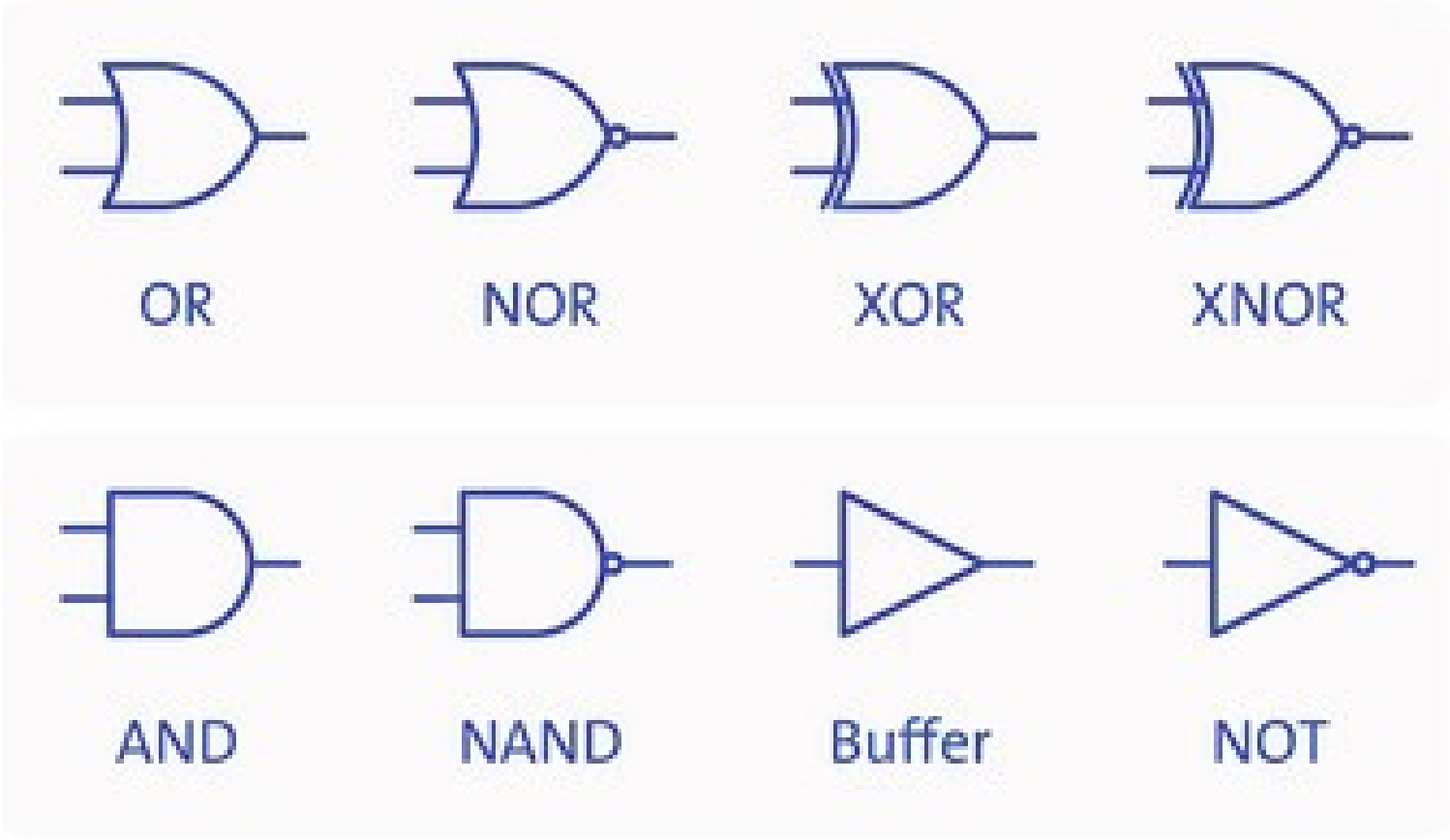
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare le funzioni somma s e riporto r tra due numeri binari da un bit x e y

x	y	s	r
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Simboli – porte logiche

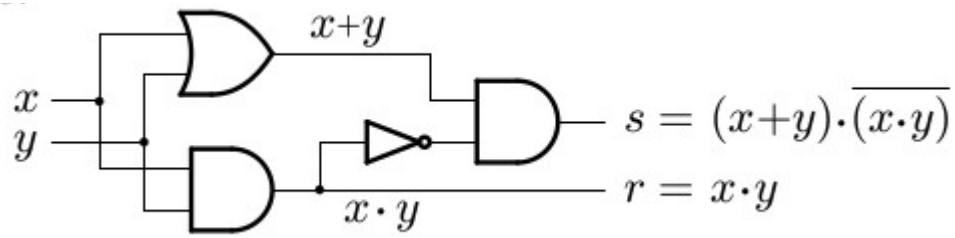


Reti logiche

Le porte logiche vengono realizzate materialmente mediante dispositivi elettronici come diodi, transistor, resistenze e MOSFET, ecc.

Funzioni Booleane più articolate si ottengono combinando gli ingressi e le uscite delle porte logiche fondamentali (OR, AND, NOT, NAND, NOR, ecc), dando origine alle reti logiche.

Esempio di rete logica che realizza l'addizione di due cifre binarie x e y con riporto, utilizzando la forma $s = x \oplus y = (x + y) \cdot (\overline{x \cdot y})$ per la somma (verificare per esercizio).



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare la tabella della verità della funzione $f(a,b,c)=\bar{a}\,b\,c$

a	b	d	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare la tabella della verità della funzione $f(a,b,c)=\bar{a}\,b\,c+a\,\bar{b}\,c$

a	b	d	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare la tabella della verità della funzione $f(a,b,c)=a\,b\,c+\bar{a}\,b\,c+a\,\bar{b}\,c+a\,\bar{b}\,\bar{c}$

a	b	d	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare la tabella della verità della funzione $f(a,b,c)=ab+b\bar{c}+a\bar{b}c+a\bar{b}\bar{c}$

a	b	d	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare la tabella della verità della funzione $f(a,b,c)=ab+b\bar{c}$

a	b	d	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Determinare la l'espressione della funzione booleana rappresentata nella seguente tabella della verità

a	b	d	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Funzioni Booleane di variabili reali

Nelle applicazioni pratiche è utile definire delle funzioni che date due variabili reali restituiscano dei valori logici.

$$f:\mathbb{R}^2 \rightarrow B, \qquad B=\{0,1\}$$

$$(a,b) \rightarrow f(a,b)$$

Un esempio notevole è costituito dagli operatori di confronto

a==b
a~=b
a>b
a>=b
a<b
a<=b

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori di confronto in Matlab

Supponiamo di aver definito la variabile x:
x=3;

Le operazione logiche restituiscono i valori come riportato qui di seguito:

y=x==0	y=0
y=x~=0	y=1
y=x>0	y=1
y=a>=0	y=1
y=x<0	y=0
y=x<=0	y=0

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori di confronto in Matlab

Supponiamo di aver definito il seguente vettore:
X=[1 2 -3 3 -2 1 4 0 -1];

L'operazione logica:

y=x>0;

fornisce un vettore in cui ogni elemento vale 1 (VERO) solo se i corrispondenti elementi del vettore sono maggiori di zero, altrimenti restituisce 0 (FALSO).
In questo caso:
y=[1 1 0 1 0 1 1 0 0];
Allo stesso modo il comando
y=(x==0);
darà in uscita un vettore in cui ogni elemento vale 1 (VERO) laddove gli elementi del vettore sono pari a zero.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operatori di confronto in Matlab

A questo punto il vettore y può essere utilizzato come una maschera per ottenere un vettore z in cui sono stati annullati tutti gli elementi minori di zero nel seguente modo:

```
y=x>0;

z=x.*y;
```

Si può usare anche la funzione **find**, per determinare la lista degli indici in un vettore per cui una determinata condizione è verificata. Per esempio

```
k=find(x>0);
```

restituisce la lista degli indici dove il vettore x è maggiore di zero.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Energia

Energia di un segnale tempo-continuo in un intervallo [a,b]

$$E_x = \varepsilon \int_a^b |x(t)|^2 dt$$

Dato che eseguiamo il calcolo numericamente, dobbiamo utilizzare un algoritmo per e:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{i=n-1} (x_{i+1}-x_i)\frac{1}{2}[f(x_{i+1})+f(x_i)]$$

Se dividiamo [a,b] in parti n parti uguali

$$x_{i+1}-x_i = \Delta x = \frac{b-a}{n}$$

Otteniamo

$$\int_a^b f(x)dx \approx \frac{1}{2}\cdot\frac{b-a}{n}\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)]$$

$$\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)] = \sum_{i=0}^{i=n-1} f(x_{i+1}) + \sum_{i=0}^{i=n-1} f(x_i) = \sum_{i=0}^{i=n-1} f(x_i) + f(b) + \sum_{i=0}^{i=n-1} f(x_i)$$

$$-f(a)+2\sum_{i=0}^{i=n-1} f(x_i) = -f(a)+2\left(\sum_{i=0}^{i=n} f(x_i) - f(b)\right) = -f(a)-2f(b)+2\left(\sum_{i=0}^{i=n} f(x_i)\right)$$

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n}\left(\sum_{i=0}^{i=n} f(x_i) - \frac{1}{2}f(a) - f(b)\right)$$

- tri.m
- % Create a triangular pulse with width 0.4.
- % fs = 10000; % Sampling frequency (samples/sec)
- % t = -1:1/fs:1; % Time Vector
- % w = .4; % Triangle Width

```
fs = 10000;      % Sampling frequency (samples/sec)
t = -1:1/fs:1;   % Time Vector
w = .4;          % Triangle Width
x = tripuls(t-0.5,w);  % Sampled aperiodic triangle
plot(t,x);
xlabel('Time (sec)');ylabel('Amplitude')
```

- % See also SQUARE, S
- % RECTPULSE,