

# Logica e Reti Logiche

## Esercitazione

Francesco Pasquale

27 novembre 2025

**Esercizio 1.** Scrivere in binario i seguenti numeri espressi in decimale<sup>1</sup>

$11_{10}$ ,  $87_{10}$ ,  $118_{10}$ ,  $365_{10}$ ,  $512_{10}$

**Esercizio 2.** Scrivere in binario la vostra data di nascita.

Tipicamente il formato decimale di una data è  $gg/mm/aaaa$ . Come dovrebbe essere il formato di una data in binario? e in esadecimale?

**Esercizio 3.** Scrivere in decimale i seguenti numeri espressi in binario

$1010_2$ ,  $110110_2$ ,  $11001100_2$ ,  $11110000_2$ ,  $01011010_2$

**Esercizio 4.** Scrivere in esadecimale i numeri degli Esercizi 1 e 3.

**Esercizio 5.** Scrivere in binario e in decimale i seguenti numeri espressi in esadecimale

$A5_{16}$ ,  $35B_{16}$ ,  $CEE2_{16}$ ,  $6E42_{16}$ ,  $D0000000_{16}$

**Esercizio 6.** Scrivere in esadecimale il vostro numero di matricola.

**Esercizio 7.** Scrivere in *complemento a due* a dieci bit i numeri seguenti

$11_{10}$ ,  $-87_{10}$ ,  $118_{10}$ ,  $-365_{10}$ ,  $-512_{10}$

**Esercizio 8.** Che numeri codificano in complemento a due a sei bit le seguenti sequenze di bit?

$010101_2$ ,  $101010_2$ ,  $110101_2$ ,  $011111_2$ ,  $111111_2$

**Esercizio 9.** Scrivere in complemento a due a sei bit i seguenti numeri decimali ed eseguire le somme. Quali di loro vanno in *overflow*<sup>2</sup>?

$16_{10} + 9_{10}$ ,  $27_{10} + 31_{10}$ ,  $-4_{10} + 19_{10}$ ,  $3_{10} + (-32_{10})$ ,  $-27_{10} + (-31_{10})$

---

<sup>1</sup>Usiamo la notazione  $n_b$  per indicare che  $n$  è la rappresentazione del numero in base  $b$ . Per esempio,  $121_{10}$  è una rappresentazione del numero *centoventuno*, mentre  $121_3$  è una rappresentazione del numero *sedici*. Il numero  $b$ , che indica la base, si intende sempre espresso in decimale.

<sup>2</sup>Una somma di due numeri in complemento a due va in *overflow* quando sommando due numeri positivi si ottiene un numero negativo, oppure quando sommando due numeri negativi si ottiene un numero positivo

		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	`	p
0001	1	SOH	DC1		1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	10	LF	SUB	*	:	J	Z	j	z
1011	11	VT	ESC	+	;	K	[	k	{
1100	12	FF	FS	,	<	L	\	l	!
1101	13	CR	GS	-	=	M	]	m	}
1110	14	SO	RS	.	>	N	'	n	~
1111	15	SI	US	/	?	O	-	o	DEL

Figura 1: Codifica ASCII

Nella codifica *ASCII* (si veda la Figura 1) si usano 7 bit per codificare  $2^7 = 128$  caratteri.<sup>3</sup>

**Esercizio 10.** Scrivete il vostro nome in codifica *ASCII*. Scrivete poi in esadecimale la codifica di ogni lettera del vostro nome.

Nell'algebra Booleana, ossia quella in cui i valori delle variabili sono soltanto 0 (**False**) e 1 (**True**), con i simboli “.” e “+” indichiamo rispettivamente le operazioni di AND e OR, e con una barretta  $\bar{x}$  sopra una variabile  $x$  ne indichiamo la negazione, valgono le seguenti uguaglianze

$$\begin{array}{l}
 \text{Assorbimento} \\
 \text{Combinazione} \\
 \text{Consenso}
 \end{array}
 \left\| \begin{array}{l}
 x(x+y) = x \\
 xy + x\bar{y} = x \\
 xy + \bar{x}z + yz = xy + \bar{x}z
 \end{array} \right| \begin{array}{l}
 x + xy = x \\
 (x+y)(x+\bar{y}) = x \\
 (x+y)(\bar{x}+z)(y+z) = (x+y)(\bar{x}+z)
 \end{array}$$

**Esercizio 11.** Verificare le uguaglianze “Assorbimento”, “Combinazione” e “Consenso” usando le tabelle di verità.

<sup>3</sup>Nella tabella in Figura 1 i tre bit che indicizzano le colonne sono quelli più significativi. Per esempio, la codifica della lettera *F* è 1000110

**Esercizio 12.** Dimostrare le uguaglianze “Assorbimento”, “Combinazione” e “Consenso” usando le proprietà delle operazioni AND e OR

	AND	OR
Elemento neutro	$x \cdot 1 = x$	$x + 0 = x$
Annullamento	$x \cdot 0 = 0$	$x + 1 = 1$
Idempotenza	$x \cdot x = x$	$x + x = x$
Complementarità	$x \cdot \bar{x} = 0$	$x + \bar{x} = 1$
Commutatività	$xy = yx$	$x + y = y + x$
Associatività	$x(yz) = (xy)z$	$x + (y + z) = (x + y) + z$
Distributività	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$

Un *grafo*  $G$  è una coppia  $G = (V, E)$  in cui  $V$  è un insieme finito ed  $E$  è un insieme di coppie di elementi di  $V$ . Per esempio,  $G = (V, E)$  dove

$$V = \{a, b, c, d\}, \quad E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{c, d\}\}. \quad (1)$$

Gli elementi di  $V$  si chiamano *nod*i (o *vertici*) del grafo, gli elementi di  $E$  si chiamano *archi*. Ogni grafo può essere disegnato pensando ai nodi come punti e agli archi come linee che uniscono i punti. Per esempio, il grafo in (1) può essere disegnato come in Figura 2

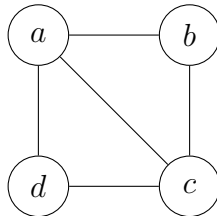


Figura 2: Disegno del grafo in (1)

Un grafo in cui  $V$  è  $\{0, 1\}^n$ , ossia l'insieme di tutte le stringhe di  $n$  bit, ed  $E$  è formato da tutte le coppie di stringhe che differiscono per un unico bit si chiama  $n$ -cubo.

**Esercizio 13.** Disegnare un 3-cubo e un 4-cubo.

**Esercizio 14.** Quanti nodi contiene un  $n$ -cubo? quanti archi?

Un *cammino* in un grafo è una sequenza di nodi  $(v_0, v_1, \dots, v_k)$  tali che  $\{v_i, v_{i+1}\}$  è un arco per ogni  $i = 0, 1, \dots, k - 1$ . Per esempio, nel grafo in (1) (si faccia riferimento anche alla Figura 2)  $(a, c, d)$  è un cammino, mentre  $(a, b, d)$  non è un cammino.

**Esercizio 15.** Evidenziare il cammino individuato dal codice Gray nel disegno di un 3-cubo e nel disegno di un 4-cubo.

**Esercizio 16.** Costruire un circuito che implementi la formula seguente

$$(p \rightarrow q \wedge r) \vee (\neg q \rightarrow \neg p)$$

**Esercizio 17.** Costruire un circuito che implementi la seguente tabella di verità

$p$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
$q$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
$r$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
$s$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$?$	0	0	0	1	1	0	1	1	0	0	0	0	1	0	1

**Esercizio 18.** Costruire un circuito che implementi la seguente funzione booleana<sup>4</sup>

$$f : \{0, 1\}^3 \rightarrow \{0, 1\}$$

$$f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$$

**Esercizio 19.** Costruire un circuito che implementi la seguente funzione booleana

$$f : \{0, 1\}^4 \rightarrow \{0, 1\}$$

$$f(x_1, x_2, x_3, x_4) = \begin{cases} 1 & \text{se } x_1 + x_2 + x_3 + x_4 \equiv 0 \pmod{3} \\ 0 & \text{altrimenti} \end{cases}$$

**Esercizio 20.** Scrivere la mappa di Karnaugh della tabella di verità dell'Esercizio 17. Qual è il numero di implicant che avete?

**Esercizio 21.** Sappiamo come costruire una formula in forma normale disgiuntiva (somma di prodotti) a partire da una mappa di Karnaugh. Come possiamo costruire una formula in forma normale congiuntiva (prodotto di somme) a partire dalla stessa mappa (senza costruire prima la disgiuntiva)?

**Esercizio 22.** Progettare un circuito che prenda in input due numeri espressi in binario a quattro bit,  $\mathbf{a} = a_3a_2a_1a_0$  e  $\mathbf{b} = b_3b_2b_1b_0$  e restituisca 1 se  $\mathbf{b}$  è il doppio di  $\mathbf{a}$  e 0 altrimenti (per esempio, se  $a_3a_2a_1a_0 = 0110$  e  $b_3b_2b_1b_0 = 1100$  il circuito deve restituire 1, perché  $\mathbf{a} = (6)_{10}$  e  $\mathbf{b} = (12)_{10}$ ).

**Esercizio 23.** Progettare un circuito che implementi il seguente algoritmo.

INPUT: Tre bit,  $x_1, x_2, x_3$ .

OUTPUT: Un bit,  $y$ .

**if**  $x_1 = 1$  **then**

$y = x_2$

**else**

$y = x_3$

**return**  $y$

**Esercizio 24.** Progettare un circuito con tre ingressi,  $x_0, x_1, x_2$ , e un'uscita,  $y$ , che implementi la formula seguente

$$y = \overline{x_0}x_2 + \overline{x_1}x_2 + x_0x_1\overline{x_2}$$

usando due *Half-Adder* e nessun'altra porta logica.

---

<sup>4</sup>Con in simbolo  $\oplus$  indichiamo l'operatore *XOR* (*OR* esclusivo), quindi in generale  $x_1 \oplus \dots \oplus x_n$  vale 1 se un numero dispari di variabili sono 1 e vale 0 altrimenti