

Numeri e Logica Digitale: Teoria ed Esercizi

Corso di Sistemi e Reti

Contents

1	Numeri in base 2 (binario)	2
1.1	Rappresentazione dei numeri binari	2
1.2	Conversione tra decimale e binario	2
1.3	Conversione tra binario e decimale	2
1.4	Esercizi svolti	3
1.5	Esercizi da fare (finchè non si ha la nausea :))	3
2	Operazioni in binario	5
2.1	Addizione binaria	5
2.2	Sottrazione binaria	5
2.3	Esercizi svolti	6
2.4	Esercizi da fare	6
3	Numeri esadecimali	7
3.1	Concetti base	7
3.2	Conversione binario \leftrightarrow esadecimale	7
3.3	Conversione esadecimale \rightarrow decimale	7
3.4	Convertire decimale in esadecimale	7
3.5	Esercizi svolti	7
4	Floating Point (virgola mobile)	9
4.1	Concetti base	9
4.2	gray!50Standard IEEE 754 (32 bit)	9
4.3	gray!50Esempio	9
5	Porte logiche	10
5.1	Porte fondamentali	10
5.2	Tabelle di verità	10
5.2.1	AND (La uso se voglio verificare che TUTTE le condizioni sono vere)	10
5.2.2	OR (La uso se voglio verificare che ALMENO una condizione è vera)	10
5.2.3	NOT	10
5.2.4	XOR	10
5.3	Esercizi svolti	10
5.4	Esercizi avanzati: Circuiti combinatori	11
5.5	Esercizi da fare	12

1 Numeri in base 2 (binario)

Il sistema binario è la base dei calcolatori digitali, in quanto ogni bit può rappresentare solo due stati: 0 e 1.

1.1 Rappresentazione dei numeri binari

Ogni cifra in binario è detta **bit**. Ad esempio:

$$1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11_{10}$$

Il pedice ₂ indica che il numero è in base 2, mentre ₁₀ indica la base decimale. 2^n rappresenta il peso di ogni bit, partendo da destra con 2^0 . Ricorda che:

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024 = 1\text{K}(1\text{Kilo})$$

$$2^{20} = 1048576 = 1\text{M}(1\text{Mega})$$

$$2^{30} = 1073741824 = 1\text{G}(1\text{Giga})$$

1.2 Conversione tra decimale e binario

Metodo della divisione per 2: Per convertire un numero decimale in binario, si divide per 2 ripetutamente e si prendono i resti dal basso verso l'alto.

Esempio: Convertire 25_{10} in binario.

$$25 \div 2 = 12 \text{ resto } 1$$

$$12 \div 2 = 6 \text{ resto } 0$$

$$6 \div 2 = 3 \text{ resto } 0$$

$$3 \div 2 = 1 \text{ resto } 1$$

$$1 \div 2 = 0 \text{ resto } 1$$

Risultato: $25_{10} = 11001_2$

1.3 Conversione tra binario e decimale

Metodo della somma ponderata: Per convertire un numero binario in decimale, si sommano i pesi dei bit che sono 1. **Esempio:** Convertire 1101_2 in decimale.

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = 13_{10}$$

1.4 Esercizi svolti

1. Convertire 43_{10} in binario:

$$43 \div 2 = 21 \text{ resto } 1$$

$$21 \div 2 = 10 \text{ resto } 1$$

$$10 \div 2 = 5 \text{ resto } 0$$

$$5 \div 2 = 2 \text{ resto } 1$$

$$2 \div 2 = 1 \text{ resto } 0$$

$$1 \div 2 = 0 \text{ resto } 1$$

Risultato: $43_{10} = 101011_2$

2. Convertire 14_{10} in binario:

$$14 \div 2 = 7 \text{ resto } 0$$

$$7 \div 2 = 3 \text{ resto } 1$$

$$3 \div 2 = 1 \text{ resto } 1$$

$$1 \div 2 = 0 \text{ resto } 1$$

Risultato: $14_{10} = 1110_2$

3. Convertire 101011_2 in decimale:

$$\begin{aligned} 101011_2 &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 32 + 0 + 8 + 0 + 2 + 1 = 43_{10} \end{aligned}$$

4. Convertire 1110_2 in decimale:

$$\begin{aligned} 1110_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\ &= 8 + 4 + 2 + 0 = 14_{10} \end{aligned}$$

1.5 Esercizi da fare (finchè non si ha la nausea :))

1. Convertire 7_{10} in binario (111).
2. Convertire 5_{10} in binario (101).
3. Convertire 12_{10} in binario (1100).
4. Convertire 78_{10} in binario (1001110).
5. Convertire 255_{10} in binario (11111111).
6. Convertire 100_{10} in binario (1100100).
7. Convertire 200_{10} in binario (11001000).
8. Convertire 120_{10} in binario (1111000).
9. Convertire 111_2 in decimale (7).

10. Convertire 101_2 in decimale (5).
11. Convertire 1100_2 in decimale (12).
12. Convertire 1001110_2 in decimale (78).
13. Convertire 1111111_2 in decimale (255).
14. Convertire 1100100_2 in decimale (100).
15. Convertire 11001000_2 in decimale (200).
16. Convertire 1111000_2 in decimale (120).

2 Operazioni in binario

2.1 Addizione binaria

Le regole fondamentali sono:

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 10$$

Quando si somma $1 + 1$, si scrive 0 e si riporta 1 alla colonna successiva. Quando si somma $1 + 1 + 1$ (incluso il riporto), si scrive 1 e si riporta 1 alla colonna successiva.

Esempio: $1011_2 + 1101_2$

$$\begin{array}{r} 1 \ 0 \ 1 \ 1 \\ + \ 1 \ 1 \ 0 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 0 \end{array}$$

Risultato: 11000_2

Esempio: $111_2 + 101_2$

$$\begin{array}{r} 1 \ 1 \ 1 \\ + \ 1 \ 0 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \end{array}$$

Risultato: 1100_2

Esempio: $1111_2 + 1111_2$

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \\ + \ 1 \ 1 \ 1 \ 1 \\ \hline 1 \ 1 \ 1 \ 1 \ 0 \end{array}$$

Risultato: 11110_2 Nota come la somma di due 1 dove c'è un riporto produce un ulteriore riporto.

2.2 Sottrazione binaria

Si usa spesso il **complemento a 2**:

1. Invertire i bit del sottraendo (complemento a 1)
2. Sommare 1
3. Sommare il risultato al minuendo

Esempio: $1011_2 - 0101_2$

$$0101_2 \rightarrow \text{complemento a 1} = 1010_2$$

$$1010_2 + 1 = 1011_2 \text{ (complemento a 2)}$$

$$1011_2 + 1011_2 = 10110_2 \text{ (scartare il riporto finale)}$$

Risultato: $0110_2 = 6_{10}$

2.3 Esercizi svolti

1. $1101_2 + 1011_2$

$$\begin{array}{r} 1 \ 1 \ 0 \ 1 \\ + \ 1 \ 0 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 0 \end{array}$$

Risultato: 11000_2

2. $1001_2 - 0110_2$

$$0110_2 \rightarrow \text{complemento a 1} = 1001_2$$

Notabenecheil1chesommiinquestocasosarebbe0001

$$1001_2 + 1 = 1010_2 \text{ (complemento a 2)}$$

$$1001_2 + 1010_2 = 10011_2 \text{ (scartare il riporto finale)}$$

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \\ + \ 1 \ 0 \ 1 \ 0 \\ \hline 1 \ 0 \ 0 \ 1 \ 1 \end{array}$$

Risultato: $0011_2 = 3_{10}$

2.4 Esercizi da fare

1. $1010_2 + 1101_2$ (risultato: 10111_2)

2. $1110_2 + 1011_2$ (risultato: 11001_2)

3. $10000_2 + 1111_2$ (risultato: 11111_2)

4. $1101_2 - 0011_2$ (risultato: 0110_2)

5. $1010_2 - 0101_2$ (risultato: 101_2)

6. $1111_2 - 0110_2$ (risultato: 1001_2)

3 Numeri esadecimali

3.1 Concetti base

Il sistema esadecimale ha 16 simboli: $0, 1, 2, \dots, 9, A, B, C, D, E, F$. Ogni cifra esadecimale rappresenta 4 bit.

Le lettere A – F rappresentano i valori decimali 10–15, cioè:

$$A_{16} = 10_{10}, B_{16} = 11_{10}, C_{16} = 12_{10}, D_{16} = 13_{10}, E_{16} = 14_{10}, F_{16} = 15_{10}.$$

In binario:

$$A = 1010_2, B = 1011_2, C = 1100_2, D = 1101_2, E = 1110_2, F = 1111_2.$$

3.2 Conversione binario \leftrightarrow esadecimale

Esempio: 11011101_2 Dividere in gruppi di 4 bit: $1101\ 1101\ 1101_2 = D_{16}, 1101_2 = D_{16}$
Risultato: DD_{16}

Esempio: 10100011_2 Dividere in gruppi di 4 bit: $1010\ 0011\ 1010_2 = A_{16}, 0011_2 = 3_{16}$
Risultato: $A3_{16}$

3.3 Conversione esadecimale \rightarrow decimale

$$2F_{16} = 2 \cdot 16^1 + 15 \cdot 16^0 = 32 + 15 = 47_{10}$$

3.4 Convertire decimale in esadecimale

Metodo della divisione per 16: Per convertire un numero decimale in esadecimale, si divide per 16 ripetutamente e si prendono i resti dal basso verso l'alto. **Esempio:** Convertire 254_{10} in esadecimale.

$$\begin{aligned} 254 \div 16 &= 15 \text{ resto } 14(E) \\ 15 \div 16 &= 0 \text{ resto } 15(F) \end{aligned}$$

Risultato: $254_{10} = FE_{16}$

3.5 Esercizi svolti

1. Convertire $3B_{16}$ in binario:

$$3 = 0011, \quad B = 1011 \Rightarrow 3B_{16} = 00111011_2$$

2. Convertire $A7_{16}$ in binario:

$$A = 1010, \quad 7 = 0111 \Rightarrow A7_{16} = 10100111_2$$

3. Convertire $7E_{16}$ in decimale:

$$7E_{16} = 7 \cdot 16^1 + 14 \cdot 16^0 = 112 + 14 = 126_{10}$$

4. Convertire $1F_{16}$ in decimale:

$$1F_{16} = 1 \cdot 16^1 + 15 \cdot 16^0 = 16 + 15 = 31_{10}$$

5. Convertire 10110101_2 in esadecimale:

$$1011 \ 0101 \rightarrow B5_{16}$$

6. Convertire 11110000_2 in esadecimale:

$$1111 \ 0000 \rightarrow F0_{16}$$

7. Convertire 255_{10} in esadecimale:

$$255 \div 16 = 15 \text{ resto } 15(F)$$

$$15 \div 16 = 0 \text{ resto } 15(F)$$

Risultato: FF_{16}

8. Convertire 100_{10} in esadecimale:

$$100 \div 16 = 6 \text{ resto } 4$$

$$6 \div 16 = 0 \text{ resto } 6$$

Risultato: 64_{16}

4 Floating Point (virgola mobile)

4.1 Concetti base

I numeri reali si rappresentano in forma normalizzata:

$$N = (-1)^S \cdot 1.M \cdot 2^E$$

- S = segno (0=positivo, 1=negativo)
- M = mantissa (parte significativa)
- E = esponente

Attenzione che questa rappresentazione non memorizza il numero perfettamente, ma una sua approssimazione. Questo può portare a errori di arrotondamento (provare a stampare il risultato di $0.1 + 0.2$:)).

4.2 Standard IEEE 754 (32 bit)

- 1 bit per il segno
- 8 bit per l'esponente
- 23 bit per la mantissa

4.3 Esempio

Rappresentare -5.25 in floating point:

$$5.25_{10} = 101.01_2 = 1.0101_2 \times 2^2$$
$$S = 1, \quad E = 2 + 127 = 129 = 10000001_2, \quad M = 010100 \dots 0$$

5 Porte logiche

Le porte logiche sono blocchi base dei circuiti digitali. Elaborano segnali 0 e 1.

5.1 Porte fondamentali

Porta	Simbolo	Funzione
AND	\wedge	1 se tutti ingressi = 1
OR	\vee	1 se almeno un ingresso = 1
NOT	\neg	inversione $0 \leftrightarrow 1$
NAND	\uparrow	negazione AND
NOR	\downarrow	negazione OR
XOR	\oplus	1 se ingressi diversi

5.2 Tabelle di verità

5.2.1 AND (La uso se voglio verificare che TUTTE le condizioni sono vere)

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

5.2.2 OR (La uso se voglio verificare che ALMENO una condizione è vera)

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

5.2.3 NOT

A	NOT A
0	1
1	0

5.2.4 XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

5.3 Esercizi svolti

1. Costruire la tabella di verità della porta NAND a due ingressi:

$A \quad B \quad A \text{ NAND } B$

Soluzione:

0 0 1, 0 1 1, 1 0 1, 1 1 0

2. Costruire la tabella di verità della porta NOR a due ingressi:

$A \quad B \quad A \text{ NOR } B$

Soluzione:

0 0 1, 0 1 0, 1 0 0, 1 1 0

5.4 Esercizi avanzati: Circuiti combinatori

1. Dato il circuito, trovare la tabella di verità:

Circuito: $(A \wedge B) \vee (\neg C)$

A	B	C	$A \wedge B$	$\neg C$	Uscita
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

2. Dato il circuito, trovare la tabella di verità:

Circuito: $(A \oplus B) \wedge C$

A	B	C	$A \oplus B$	Uscita
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

3. Dato il circuito, trovare la tabella di verità:

Circuito: $\neg(A \wedge B) \vee (B \wedge C)$

A	B	C	$A \wedge B$	$\neg(A \wedge B)$	$B \wedge C$	Uscita
0	0	0	0	1	0	1
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	1	1	1
1	0	0	0	1	0	1
1	0	1	0	1	0	1
1	1	0	1	0	0	0
1	1	1	1	0	1	1

4. **Data la tabella di verità, trovare il circuito logico:**

A	B	Uscita
0	0	1
0	1	0
1	0	0
1	1	0

Soluzione: $\neg A \wedge \neg B$ oppure $\neg(A \vee B)$ (porta NOR)

5. **Data la tabella di verità, trovare il circuito logico:**

A	B	C	Uscita
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

Soluzione: $A \vee B \vee C$ (L'uscita è 1 se almeno uno degli ingressi è 1)

6. **Dato il circuito, trovare la tabella di verità:**

Circuito: $(A \wedge \neg B) \vee (\neg A \wedge B)$

Nota: questa è equivalente a $A \oplus B$

A	B	$\neg A$	$\neg B$	$A \wedge \neg B$	$\neg A \wedge B$	Uscita
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

5.5 Esercizi da fare

1. Costruire la tabella di verità per: $(A \vee B) \wedge \neg C$

Soluzione:

A	B	C	$A \vee B$	$\neg C$	Uscita
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

2. Costruire la tabella di verità per: $\neg(A \oplus B)$ (XNOR)

Soluzione:

A	B	$A \oplus B$	Uscita
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

3. Dato il circuito $\neg A \vee (B \wedge C)$, costruire la tabella di verità completa

Soluzione:

A	B	C	$\neg A$	$B \wedge C$	Uscita
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

4. Data la tabella: $A=0, B=0 \rightarrow 1$; $A=0, B=1 \rightarrow 1$; $A=1, B=0 \rightarrow 1$; $A=1, B=1 \rightarrow 0$.
Trovare il circuito.

Soluzione: $\neg(A \wedge B)$ oppure $A \text{ NAND } B$

5. Costruire la tabella di verità per: $(A \wedge B) \oplus C$

Soluzione:

A	B	C	$A \wedge B$	Uscita
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	0

6. Trovare il circuito per l'uscita che vale 1 solo quando esattamente due ingressi su A,B,C sono 1

Soluzione: $(A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C)$

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