FORMULARIO: SISTEMI DI NUMERAZIONE

In ogni riquadro: la regola con le sue condizioni e un esempio con i numeri.

1 Basi e conversioni

1.1 Notazione posizionale

In base b, le cifre $c_n \dots c_1 c_0$ valgono

$$c_n b^n + \cdots + c_1 b^1 + c_0 b^0$$

Cifre ammesse: da 0 a $b-1$. La base si scrive a pedice.

ESEMPIO

$$(1011)_2 = 8 + 0 + 2 + 1 = 11 \qquad (3407)_{10} = 3 \cdot 10^3 + 4 \cdot 10^2 + 7$$

1.2 Da base b a base 10

Si moltiplica ogni cifra per la sua potenza e si somma. In binario basta **sommare le colonne dove c'è un 1**.

ESEMPIO

$$(11010110)_2 = 128 + 64 + 16 + 4 + 2 = 214$$

$$(3A7)_{16} = 3 \cdot 256 + 10 \cdot 16 + 7 = 935$$

1.3 Da base 10 a base b: divisioni successive

Si divide per b tenendo i **resti**, finché il quoziente è 0. I resti si leggono **dal basso verso l'alto**.

ESEMPIO

$37 : 2$ dà resti $1, 0, 1, 0, 0, 1$ (in quest'ordine), quindi

$$37 = (100101)_2 \qquad \text{controllo: } 32 + 4 + 1 = 37$$

1.4 Da una base all'altra

Quando nessuna delle due è 10, si passa **dalla base 10**: prima $b \to 10$, poi $10 \to d$. Se invece una è potenza dell'altra si raggruppano le cifre (scheda seguente).

ESEMPIO

$$(2301)_4 \to 177 \to (261)_8$$

1.5 Cambi rapidi: basi potenza di due

Verso la base 2^k si raggruppano le cifre binarie a k per volta, **partendo da destra**, completando l'ultimo gruppo con zeri **a sinistra**.

$$4 = 2^2 \to 2 \text{ bit} \qquad 8 = 2^3 \to 3 \text{ bit} \qquad 16 = 2^4 \to 4 \text{ bit}$$

ESEMPIO

$$\underbrace{110}_{6}\,\underbrace{101}_{5}\,\underbrace{110}_{6} \Rightarrow (656)_8 \qquad \underbrace{0001}_{1}\,\underbrace{1010}_{A}\,\underbrace{1110}_{E} \Rightarrow (1AE)_{16}$$

1.6 Le cifre esadecimali

hex	A	B	C	D	E	F
dec	10	11	12	13	14	15

Maiuscole. Ogni cifra vale quattro bit, quindi **un byte sta in due cifre**.

ESEMPIO

$$(2F)_{16} = 47 \qquad (FF)_{16} = 255 = (11111111)_2$$

1.7 Contare in binario

Si conta come in base 10, ma i simboli finiscono subito: dopo 1 si riparte da 0 e si aggiunge una colonna.

$$0,\ 1,\ 10,\ 11,\ 100,\ 101,\ 110,\ 111,\ 1000, \ldots$$

ESEMPIO

Ogni volta che si supera una potenza di due si aggiunge una cifra: $(1000)_2 = 8$ è il primo numero a quattro cifre, come 1000 è il primo a quattro cifre in base 10.

1.8 Quanti numeri stanno in n bit

Con n bit si scrivono 2^n numeri diversi, da 0 a $2^n - 1$:

$$8 \text{ bit} \to 256 \text{ valori, da } 0 \text{ a } 255$$

Serve a sapere subito se un numero *ci sta*.

ESEMPIO

300 non sta in 8 bit (max $= 255$): ne servono 9. In complemento a due il campo si dimezza da ogni parte: da -128 a $+127$.

La tavola da tenere a mente

dec	bin	oct	hex
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2^n	valore
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

2 Le quattro operazioni in binario

2.1 Addizione

$$0+0=0 \qquad 0+1=1 \qquad 1+0=1 \qquad 1+1=10$$

$1+1$: scrivo 0 e riporto 1. $\quad 1+1+1=11$: scrivo 1 e riporto 1.

ESEMPIO

$$\begin{array}{r} \text{rip. } 1\ 1\ 1 \\ 1\,0\,1\,1 \\ +\ 0\,1\,1\,0 \\ \hline 1\,0\,0\,0\,1 \end{array} \qquad 11+6=17$$

2.2 Sottrazione

$$0-0=0 \qquad 1-0=1 \qquad 1-1=0 \qquad 0-1=1 \text{ col prestito}$$

Il prestito vale **due**: $10-1=1$.

ESEMPIO

$$\begin{array}{r} 1010 \\ -\,0011 \\ \hline 0111 \end{array} \qquad 10 - 3 = 7$$

2.3 Moltiplicazione

$$1 \cdot 1 = 1, \quad \text{tutto il resto } 0$$

Ogni prodotto parziale è **zero** o una **copia** del primo numero, spostata di una posizione ogni volta.

ESEMPIO

$$\begin{array}{r} 1101 \\ \times \quad 101 \\ \hline 1101 \\ 0000 \\ 1101 \\ \hline 1000001 \end{array} \qquad 13 \times 5 = 65$$

2.4 Moltiplicare e dividere per potenze di due

Per 2 si aggiunge **uno zero a destra**; per 4 due, per 8 tre. Dividere per 2 è togliere l'ultima cifra (che diventa il resto).

ESEMPIO

$$(1101)_2 \times (100)_2 = (110100)_2 \qquad 13 \times 4 = 52$$

2.5 Divisione

Divisione lunga: a ogni passo la cifra del quoziente è 1 se il divisore ci sta, 0 se non ci sta. Si sottrae e si abbassa la cifra dopo.

ESEMPIO

$$(110110)_2 : (101)_2 = (1010)_2 \text{ resto } (100)_2$$

Prova: $5 \cdot 10 + 4 = 54$.

3 Numeri negativi e controlli

3.1 Complemento a due

Per scrivere $-x$ su n bit:

$$2^n - x \qquad \text{ovvero} \qquad \text{inverti tutti i bit e aggiungi 1}$$

Il primo bit è il **segno**: 0 positivo, 1 negativo. Su 8 bit si va da -128 a $+127$.

ESEMPIO

$37 = 00100101 \;\to$ inverto $11011010 \;\to +1$:

$$-37 = (11011011)_2 \qquad \text{controllo: } 256 - 37 = 219$$

3.2 Sottrarre sommando

$a - b$ si fa come $a +$ (complemento a due di b), e il riporto che esce dagli n bit **si butta via**.

ESEMPIO

$$50 - 37: \quad 00110010 + 11011011 = 1\;00001101 \;\to\; (00001101)_2 = 13$$

3.3 Il controllo, sempre lo stesso

Si converte tutto in base 10 e si rifà l'operazione lì. Nelle divisioni:

$$\text{divisore} \times \text{quoziente} + \text{resto} = \text{dividendo}$$

ESEMPIO

Un risultato binario con dentro un 2 è sbagliato di sicuro: in base b le cifre arrivano a $b - 1$.

Quale strada conviene

Da	A	Come
base b	10	somma dei valori posizionali
10	base b	divisioni successive, resti dal basso
2	4, 8, 16	gruppi di 2, 3, 4 bit da destra
4, 8, 16	2	ogni cifra $\to$ 2, 3, 4 bit
8	16	passando dal binario, mai diretto
b	d (nessuna è 10)	$b \to 10 \to d$

I sei errori che costano di più

- Leggere i resti dall'alto invece che dal basso.
- Usare cifre che nella base non esistono ($(129)_8$, un 2 in binario).
- Scrivere $1 + 1 = 2$ invece di 0 con riporto.
- Nel prestito, calcolare $10 - 1 = 9$: in binario fa 1.
- Raggruppare da sinistra nei cambi rapidi, o mettere gli zeri di completamento dopo invece che davanti.
- Leggere $(10000000)_2$ come 128 quando si lavora in complemento a due: lì vale -128.