

SISTEMI DI NUMERAZIONE

La modalità di rappresentare elettronicamente l'informazione sono due:

- digitale;
- analogica.

Un **sistema analogico** è un sistema in cui, le informazioni sono rappresentate da un insieme continuo di valori; poiché esiste una analogia tra i valori della grandezza adottata e le informazioni da rappresentare, questa modalità viene detta, appunto, analogica.

In un **sistema digitale**, invece, le informazioni sono rappresentate in modo discreto da entità distinte detti livelli e ogni livello viene codificato in modo opportuno tramite i simboli 0 e 1.

il termine **digitale** deriva dal latino **digitus**, che significa “dito”, e fa riferimento al fatto che, in origine, il conteggio veniva svolto utilizzando le dita delle mani.

I calcolatori utilizzano i sistemi digitali binari dove i simboli usati sono solamente lo 0 e l'1. A questa unità dell'informazione si è attribuito il nome **bit**, che deriva da **binary digit** (cifra binaria), e il raggruppamento di **8 bit**, che costituisce l'unità di misura della capacità di memoria, è il **byte**.

1	0	0	1	1	1	0	0	
							Bit	
Byte (8 bit)								

SISTEMI DI NUMERAZIONE A CODICI PESATI

Un **sistema di numerazione** è un modo di esprimere e rappresentare i numeri attraverso un insieme di simboli e delle regole. I simboli di un sistema di numerazione prendono il nome di **cifre**.

Nei **sistemi posizionali (decimale, binario, ottale e esadecimale)** le cifre costituenti un qualsiasi numero hanno un **peso** dipendente dalla **posizione** della cifra nel numero stesso. In ogni numero del sistema posizionale, la cifra più a destra ha il peso minore (**cifra meno significativa**), quella più a sinistra il peso maggiore (**cifra più significativa**).

In generale, in un sistema posizionale di base **b** la scrittura:

$$a^k a^{k-1} \dots a^1 a^0, a^{-1} a^{-2} \dots a^{-h} a^{-h}$$

Equivale a:

$$a^k \cdot b^k + a^{k-1} \cdot b^{k-1} + \dots + a^1 \cdot b^1 + a^0 \cdot b^0 + a^{-1} \cdot b^{-1} + a^{-2} \cdot b^{-2} + \dots + a^{-h} \cdot b^{-h}$$

Sistema decimale: Il sistema di numerazione posizionale più noto è il sistema **decimale** che si avvale dei dieci simboli (o cifre) 0,1,2,3,4,5,6,7,8,9 ed è un sistema a base 10.

Esempio: Vediamo come si può scrivere il numero **821,32**

Esso si ottiene sommando:

8 centinaia +

2 decine +

1 unità +

3 decimi +

2 centesimi =

821,32

Pertanto il suo valore nel sistema decimale si può ottenere come:

$$8 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 + 3 \times 10^{-1} + 2 \times 10^{-2}$$

Notiamo che la notazione con il pedice 10 indica che il numero si riferisce ad un valore espresso in base 10.

Nei sistemi informatici si utilizza prevalentemente la numerazione:

- binaria;
- ottale;
- esadecimale.

REGOLA DI CONVERSIONE DI BASE

base decimale base b

- ☐ Divisioni ripetute del numero N per il valore della base b
- ☐ Il **resto** i-esimo rappresenta la cifra in base b (dalla cifra meno significativa)
- ☐ Il **quoziente** diventa il nuovo dividendo della divisione successiva per b
- ☐ Stop quando l'ultimo quoziente diventa zero

CONVERSIONE DA BASE b A BASE 10

Per convertire un numero $a^k a^{k-1} \dots a^1 a^0$ da una qualunque base b alla base 10 è sufficiente rappresentarlo esplicitamente:

$$a^k \cdot b^k + a^{k-1} \cdot b^{k-1} + \dots + a^1 \cdot b^1 + a^0 \cdot b^0, a^{-1} \cdot b^{-1} + \dots + a^{-h} \cdot b^{-h}$$

SISTEMA DI NUMERAZIONE BINARIA

La base del sistema binario è 2. I simboli del sistema binario sono: 0 e 1

Il sistema di conteggio avviene in maniera analoga a quello del sistema decimale.

Il numero **1001,101** può essere corrisponde al numero decimale:

$$1001,101_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 9,625_{10}$$

SISTEMA DI NUMERAZIONE OTTALE

Il sistema di numerazione *ottale* utilizza una notazione posizionale basata su 8 cifre (da 0 a 7) e sulle potenze di 8.

Il numero **534** corrisponde al numero decimale: $534_8 = 5 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = 348_{10}$

SISTEMA DI NUMERAZIONE ESADECIMALE

La numerazione *esadecimale* utilizza una notazione posizionale basata su 16 cifre (da 0 a 9 ed i caratteri A, B, C, D, E, F) e sulle potenze di 16. [A=10; B=11; C=12; D=13; E=14; F=15].

Il numero **B7FC16** corrisponde al numero decimale :

$$B7FC_{16} = (11) \times 16^3 + 7 \times 16^2 + (15) \times 16^1 + (12) \times 16^0 = 47100_{10}$$

CONVERSIONE DA BASE 10 A BASE b Per convertire un numero decimale ad una base b qualsiasi occorre trovare tutti i resti delle successive divisioni del numero per la base **b** fino a ottenere un quoziente uguale a zero.

CONVERSIONE DA BASE 10 A BASE 2

Come esempio si vuole trovare il valore binario del numero 210

Per una verifica di correttezza basta riconvertire il risultato alla base 10:

$$11010010_2 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 210_{10}$$

210	2	resto	0	LSB	↑ > La sequenza dei resti dal basso verso l'alto, fornisce il numero: 11010010₂ LSB: Least Significant Bit, bit meno significativo MSB: Most Significant Bit, bit più significativo
105	2	1			
52	2	0			
26	2	0			
13	2	1			
6	2	0			
3	2	1			
1	2	1	MSB		
0					

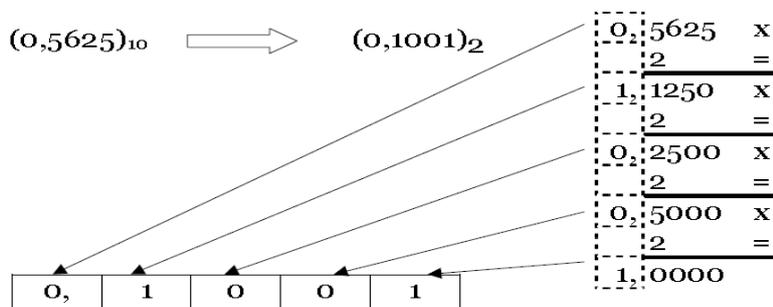
CONVERSIONE DECIMALE-BINARIO DI NUMERI CON PARTE FRAZIONARIA

Per la conversione di numeri decimali con parte frazionaria, occorre distinguere fra parte intera e quella frazionaria. Alla prima si applica il metodo delle successive divisione per 2.

Alla seconda quello dei successivi prodotti per 2.

Nella parte frazionaria, il procedimento ha termine quando la parte decimale del risultato si riduce a zero. A questo punto le parti intere dei prodotti svolti costituiscono, nell'ordine in cui sono state ricavate, il numero binario frazionario cercato.

Esempio



Nota: **Non** è detto che il procedimento di conversione da un numero decimale frazionario ad un numero binario abbia sempre termine.

In alcuni casi il risultato della moltiplicazione per 2 non arriverà mai ad avere parte decimale nulla.

In questi casi bisognerà arrestarsi arbitrariamente ad un certo numero di cifre dopo la virgola. Questo introdurrà un inevitabile errore di conversione dovuto al troncamento di cifre effettuato.

OPERAZIONI BINARIE

Le operazioni binarie fondamentali utilizzati nei circuiti digitali sono: l'addizione, la sottrazione, la moltiplicazione e la divisione.

Addizione aritmetica binaria

Questa operazione rispetta le stesse regole di quella decimale e soddisfa l'algoritmo:

$$\begin{array}{l}
 0 + 0 = 0 \quad 1 + 0 = 1 \\
 0 + 1 = 1 \quad 1 + 1 = 0 \quad \text{con riporto di 1}
 \end{array}$$

Poiché in un sistema binario la cifra più alta è 1, tutte le volte che la somma supera tale valore si ha un riporto, così come avviene nell'addizione decimale ogni volta che si supera 9.

ESERCIZI SVOLTI

1. calcola la somma dei numeri binari $(110)_2$ e $(101)_2$.

1					Riporto
	1	1	0		1° numero $(6)_{10}$
	1	0	1		2° numero $(5)_{10}$
	1	0	1	1	Risultato $(11)_{10}$

Nel caso in cui siano da sommare due 1 e un riporto di 1, l'algoritmo precedente deve essere completato nel modo seguente: $1 + 1 + 1 = 1$ con riporto di 1.

2. Somma il numero binario $(101101)_2$.

1	1	1	1	1	1	← Riporto
	1	0	1	1	0	1° numero $(45)_{10}$
	1	1	0	1	1	2° numero $(55)_{10}$
1	1	0	0	1	0	Risultato $(100)_{10}$

IL BIT DI SEGNO

La maggior parte dei sistemi numerici trattano i numeri positivi e negativi distinguendoli mediante i due segni aggiuntivi + e -.

Poiché i dispositivi digitali accettano soltanto i due simboli 0 e 1, per tenere conto del valore positivo o negativo dei numeri binari si è introdotto un bit supplementare: di valore 0 se il numero è positivo, di valore 1 se il numero è negativo. Tale **bit di segno** è posto alla sinistra del bit più significativo (MSB) del numero binario. Così ad esempio, i numeri decimali +15 e -15 sono espressi in forma binaria da:

$$(+15)_{10} = \mathbf{0} \ 1111 \qquad (-15)_{10} = \mathbf{1} \ 1111$$

ADDIZIONE BINARIA DI NUMERI INTERI CON BIT DI SEGNO

Nell'addizione di numeri binari provvisti di bit di segno possono presentarsi tre casi:

1. Addizione di due numeri positivi;
2. Addizione di due numeri di segno opposto;
3. Addizione di due numeri negativi.

Il metodo per compiere tale operazione consiste nello scrivere normalmente i numeri positivi e sostituire i numeri **negativi** con il loro **complemento a 2** prima di sommarli.

Una volta eseguita l'operazione, e trascurato l'eventuale riporto, se il risultato trovato è negativo occorre complementarlo a 2.

Nell'eseguire le operazioni è inoltre necessario che il numero dei bit di ciascun numero sia lo stesso; questo numero di bit è dato dalla relazione:

$$N_{\text{bit}} = n + 1$$

N_{bit} : numero di bit da utilizzare per rappresentare i numeri (detto campo di bit);

n : numero di bit del numero più grande comprensivo del bit di segno.

ESERCIZIO SVOLTO Addizione in binario i due numeri di segno opposto (+17)₁₀ e (-12)₁₀.

Il numero $(+17)_{10}$ è costituito da 6 bit $(\mathbf{0} \ 10001)_2$, mentre il numero $(-12)_{10}$ è costituito da 5 bit $(\mathbf{1} \ 1100)_2$. Dalla relazione $N_{\text{bit}} = n + 1$ deriva che:

$$N_{\text{bit}} = 6 + 1 = 7 \text{ bit}$$

Dunque $(+17)_{10}$ e $(-12)_{10}$ vanno scritti ciascuno con 7 bit. Ciò si ottiene aggiungendo al primo numero uno 0 e al secondo due 0. E vengono inseriti a sinistra del bit MSB e a destra del bit di segno.:

$$\begin{array}{l} (+17)_{10} \longrightarrow (\mathbf{0} \ 010001)_2 \\ (-12)_{10} \longrightarrow (\mathbf{1} \ 001100)_2 \end{array}$$

Per l'addizione dei due numeri, occorre sostituire il numero negativo $(-12)_{10}$, con il suo **complemento a 2**.

Numero binario negativo	Complemento a 1	Complemento a 2
$\mathbf{1} \ 001100$	$\mathbf{1} \ 110011$	$\mathbf{1} \ 110100$

Complemento a 1 di un numero binario si ottiene molto semplicemente sostituendo nel numero dato gli 1 con gli 0 e gli 0 con 1.

Complemento a 2 di un numero binario si può facilmente ricavare sommando 1 al complemento a 1 dello stesso numero.

$$\begin{array}{r} \mathbf{0} \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\ \mathbf{1} \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \\ \hline \mathbf{1} \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \end{array} \qquad \begin{array}{l} (+17)_{10} \\ (-12)_{10} \\ \text{Risultato } (+5)_{10} \end{array}$$

Il riporto è da trascurare perché, abbiamo supposto che la rappresentazione è a 7 bit, $N_{\text{bit}} = 7$.

ESERCIZI SVOLTI

a) Sottrai in binario $(+13)_{10}$ e $(+11)_{10}$ con il metodo del complemento a 2

Sottrarre i numeri $(+13)_{10}$ e $(+11)_{10}$ equivale ad aggiungere $(+13)_{10}$ e $(-11)_{10}$.

Utilizzando il metodo del complemento a 2 (con $N_{\text{bit}} = 7$), si ha:

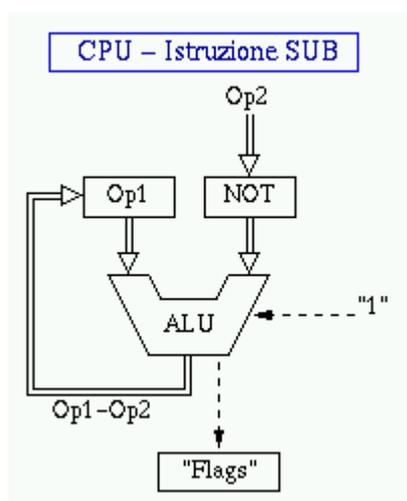
$$(+13)_{10} = (\mathbf{0} \ 001101)_2 \qquad (-11)_{10} = (\mathbf{1} \ 001011)_2$$

Per l'addizione dei due numeri, occorre sostituire il numero negativo $(-11)_{10}$, con il suo **complemento a 2**.

Numero binario negativo	Complemento a 1	Complemento a 2
1 001011	1 110100	1 110101

Effettuando quindi l'addizione tra il primo numero e il secondo, comprensivi dei bit di segno, trascurato l'ultimo eventuale riporto:

$$\begin{array}{r}
 0\ 001101 \quad (+13)_{10} \\
 1\ 110101 \quad (-11)_{10} \\
 \hline
 1\ 000010 \quad \text{Risultato } (+2)_{10} \\
 \uparrow \uparrow \\
 \text{Bit di segno} \\
 \text{Riporto da trascurare } (N_{\text{bit}}=7)
 \end{array}$$



Il vantaggio del complemento a 2 e' quello di velocizzare i calcoli effettuati dall'ALU (**A**ritmetico **L**ogico **U**nit) posta all'interno della CPU (**C**entral **P**rocessing **U**nit) nel calcolo delle differenze fra valori.

La rete di ingresso dell'ALU e' molto veloce ad effettuare il complemento a 1 dell'operando da sottrarre (sottraendo) eseguendo una negazione di esso, dopodich  l'ALU effettua l'operazione di somma tra

- il primo operando (minuendo)
- la negazione del sottraendo
- il valore '1' preventivamente inserito nell'ingresso del riporto (CI - Carry Input)

SOTTRAZIONE BINARIA

La sottrazione in modulo e segno si esegue in maniera simile alla sottrazione per i numeri naturali (decimali). Si esegue la sottrazione bit a bit da destra verso sinistra, tenendo conto di eventuali prestiti dalle cifre a sinistra quando necessario. **Il metodo non funziona se il risultato della sottrazione   negativo**, cio  quando il primo termine (Minuendo)   pi  piccolo del secondo (Sottraendo). In questo caso occorre invertire i termini prima della sottrazione e aggiustare il segno alla fine del calcolo.

Nella sottrazione binaria tra due numeri senza segno occorre tener presente che:

$$\begin{array}{l}
 0 - 0 = 0 \\
 \mathbf{0 - 1 = 1 \text{ con prestito di 1 dalla cifra precedente}} \\
 1 - 0 = 1 \\
 1 - 1 = 0
 \end{array}$$

ESERCIZI SVOLTI

a) Sottrai il numero binario $(01111)_2$ dal numero binario $(10000)_2$.

$$\begin{array}{r}
 1\ 1\ 1 \quad \text{Cifra in prestito} \\
 1\ 0\ 0\ 0 \quad \text{Minuendo } (16)_{10} \\
 0\ 1\ 1\ 1 \quad \text{Sottraendo } (15)_{10} \\
 \hline
 0\ 0\ 0\ 1 \quad \text{Risultato } (1)_{10}
 \end{array}$$

La sottrazione fra numeri binari pu  essere vista anche come una somma del (minuendo) A e del (sottraendo) B.

$$A - B = A + (-B)$$

In questo caso occorre utilizzare la notazione del complemento a 1 oppure a 2.

LA SOTTRAZIONE IN COMPLEMENTO A 2**1) CASO IN CUI IL MINUENDO SIA MAGGIORE O UGUALE DEL SOTTRAENDO (M >= S)**

Le regole da seguire sono:

- Determinare il complemento a 2 del Sottraendo (il numero più piccolo).
- Sommare il complemento a 2 ottenuto al Minuendo (il numero più grande).
- La somma genera sempre un riporto che va scartato; la somma senza il riporto rappresenta il risultato della sottrazione data.

$$\begin{array}{r}
 27 - \quad 11011 - \\
 18 = \quad 10010 = \text{--- C a 2} \rightarrow \quad 01110 = \\
 \hline
 9 \quad \quad 01001
 \end{array}
 \quad
 \begin{array}{r}
 \quad \quad \quad 1 \ 111 \\
 \quad \quad \quad 11011 + \\
 \hline
 \quad \quad \quad 01110 = \\
 \text{Riporto} \quad \times \quad 01001 + \\
 \quad \quad \quad \downarrow \\
 \text{Risultato} \quad \quad 01001
 \end{array}$$

2) CASO IN CUI IL MINUENDO SIA MINORE DEL SOTTRAENDO (M <= S)

Le regole da seguire sono:

- Determinare il complemento a 2 del Sottraendo (il numero più grande).
- Sommare il complemento a 2 ottenuto al Minuendo (il numero più piccolo).
- La somma del punto b) non genera alcun riporto. Per ottenere il risultato della sottrazione occorre complementare a 2 la somma ottenuta. Infine, la mancanza di riporto nella somma indica che il risultato sarà negativo, quindi aggiungere il segno.

$$\begin{array}{r}
 18 - \quad 10010 - \\
 27 = \quad 11011 = \text{--- C a 2} \rightarrow \quad 00101 = \\
 \hline
 -9 \quad \quad -01001
 \end{array}
 \quad
 \begin{array}{r}
 \quad \quad \quad 11011 \quad 10010 + \\
 \quad \quad \quad 00101 = \\
 \hline
 \quad \quad \quad 10111 \\
 \quad \quad \quad \downarrow \text{C a 2} \\
 \text{Risultato} \quad -01001
 \end{array}$$

LA SOTTRAZIONE IN COMPLEMENTO A 1

La sottrazione tra due numeri binari può essere effettuata con le regole della sottrazione stessa viste in precedenza oppure ricorrendo al metodo del complemento a 1. Con questo metodo si esegue soltanto l'operazione di addizione.

1) CASO IN CUI IL MINUENDO SIA MAGGIORE O UGUALE DEL SOTTRAENDO (M >= S)

Le regole da seguire per una corretta esecuzione dell'operazione sono:

- Determinare il complemento a 1 del Sottraendo (il numero più piccolo).
- Sommare il complemento a 1 ottenuto al Minuendo (il numero più grande).
- La somma del punto b) genera un riporto che va sommato alla somma stessa per ottenere il risultato della sottrazione data.

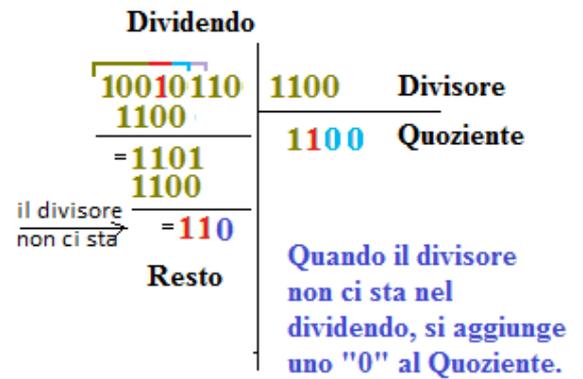
Esempio.

$$\begin{array}{r}
 27 - \quad 11011 - \\
 18 = \quad 10010 = \text{--- C a 1} \rightarrow \quad 01101 = \\
 \hline
 9 \quad \quad 01001
 \end{array}
 \quad
 \begin{array}{r}
 \quad \quad \quad 1 \ 1111 \\
 \quad \quad \quad 11011 + \\
 \hline
 \quad \quad \quad 01101 = \\
 \text{Riporto} \quad 1 \ 01000 + \\
 \quad \quad \quad \downarrow \\
 \quad \quad \quad 1 = \\
 \text{Risultato} \quad \quad 01001
 \end{array}$$

ESEMPIO: Vogliamo dividere A:B
 $A = (10010110)_2$ corrisponde a $(150)_{10}$
 $B = (1100)_2 \rightarrow (12)_{10}$

Procedimento dell'operazione:

- 1- Si cerca la prima parte del dividendo che sia maggiore del divisore. Tale prima parte è nel nostro caso 10010;
- 2- Scrivere 1 al quoziente;
- 3- Calcolare il resto: la differenza $10010 - 1100$. Si ottiene 110.
- 4- Si abbassa, la cifra successiva del dividendo, cioè 1, ottenendo 1101;
- 5- Il divisore 1100 "sta" nel 1101, ovviamente una volta e con resto 1. Il quoziente diviene 11;
- 6- Abbassando la cifra successiva "1" si ha 11. Questa volta il divisore "non sta" in questa parte del dividendo e quindi si aggiunge uno "0" al quoziente, abbassando la cifra successiva. Questo è l'ultimo 0, che dà 110 nel dividendo;
- 7- Di nuovo il 1100 "non sta" nel 110 e perciò si aggiunge un altro 0 al quoziente;
- 8- Non essendoci più cifre da calare ciò significa che l'operazione è finita;
- 9- Quindi il quoziente è 1100 in base 2 = 12 in base 10,
- 10- Il resto è 110 in base 2 = 6 in base 10.

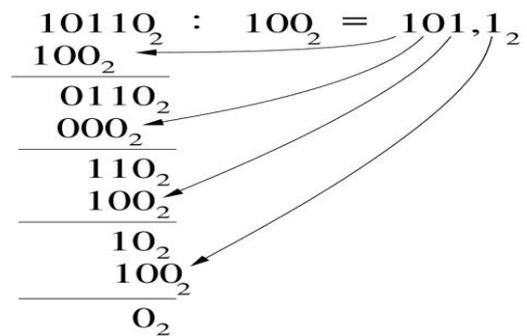


Risultato 150 in base 10 :12 in base 10 =12 in base 10 con resto 6 in base 10

Mentre in base 2 è:

$(10010110)_2 : (1100)_2 = (1100)_2$ con resto $(110)_2$

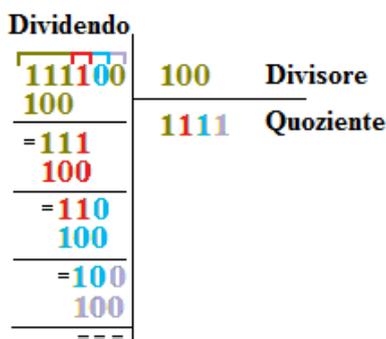
Esempio: $A = (10110)_2 \rightarrow (22)_{10}$ e $B = (100)_2 \rightarrow (4)_{10}$
 In questo caso il risultato è 101_2 (5_{10}), con il resto di 10_2 (2_{10});
 ovvero $101,1_2$ ($5,5_{10}$).



ESERCIZIO SVOLTO

vogliamo dividere i numeri $(111100)_2$ con $(100)_2$ che, rispettivamente, corrispondono $(60)_{10}$ e $(4)_{10}$.

Si procede, nel modo seguente:



Prendiamo le prime tre cifre del dividendo **111** e dividiamo per **100**. Il risultato è **1**. Moltiplichiamo **1** per **100** e otteniamo **100**. Eseguiamo $111 - 100 = 11$.

Abbassiamo la quarta cifra del dividendo **1** e dividiamo **111** per **100**. Il risultato è **1**. Moltiplichiamo **1** per **100** e otteniamo **100**. Eseguiamo $111 - 100 = 11$.

Abbassiamo la quinta cifra del dividendo **0** e dividiamo **110** per **100**. Il risultato è **1**. Moltiplichiamo **1** per **100** e otteniamo **100**. Eseguiamo $110 - 100 = 10$.

Abbassiamo l'ultima cifra del dividendo **0** e dividiamo **100** per **100**. Il risultato è **1**. Moltiplichiamo **1** per **100** e otteniamo **100**. Eseguiamo $100 - 100 = 0$.

Quindi la divisione tra 111100 e 100 è uguale a 1111 .

Verifichiamo con il **SISTEMA DECIMALE**:

NUMERO BINARIO	NUMERO DECIMALE
111100 :	60 :
100 =	4 =
1111	15

Infatti: $1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 = 8 + 4 + 2 + 1 = 15$.

Facciamo un altro esempio

vogliamo dividere il numero binario **11001** con il numero binario **101** che indicano, rispettivamente, i numeri decimali **25** e **5**.

Procediamo nei modi consueti:

$\begin{array}{r} \overline{11001} \\ 101 \\ \hline = 101 \\ 101 \\ \hline = = = \end{array}$	$\begin{array}{r} 101 \\ 101 \end{array}$
---	---

Abbassiamo le prime tre cifre del dividendo **110** e dividiamo per **101**. Il risultato è **1**. Moltiplichiamo **1** per **101** e otteniamo **101**. Eseguiamo **110 - 101** ricordando che quando occorre **CHIEDERE IN PRESTITO una unità alla cifra di ordine immediatamente superiore essa vale 2 UNITA' dell'ordine immediatamente inferiore** quindi il risultato è **1**.

Abbassiamo la quarta cifra del dividendo **0** e dividiamo **10** per **101**. Il risultato è **0**.

Abbassiamo la quinta cifra del dividendo **1** e dividiamo **101** per **101**. Il risultato è **1**. Moltiplichiamo **1** per **101** e otteniamo **101**. Eseguiamo **101 - 101 = 0**.

Quindi la divisione tra **11001** e **101** è uguale a **101**.

Verifichiamo con il **SISTEMA DECIMALE**:

NUMERO BINARIO	NUMERO DECIMALE
11001 :	25 :
101 =	5 =
101	5

Infatti: $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 4 + 0 \times 2 + 1 \times 1 = 4 + 0 + 1 = 5$.

ESERCIZI DA SVOLGERE

Es.1 Convertire in base 10 i seguenti numeri rappresentati nelle basi indicate:

$(40F)_{16}$	$(5778)_{16}$	$(781)_{16}$	$(3B8)_{16}$	$(10010)_8$	$(2EA)_{16}$
$(1000101)_2$	$(126)_8$	$(477)_8$	$(1357)_8$	$(1110)_2$	$(1011011,011)_2$

Es.2 Convertire i seguenti numeri decimali nelle basi specificate:

345 in base 2	345 in base 8	345 in base 16	989 in base 5	417 in base 7	615 in base 9
426 in base 2	811 in base 16	1101 in base 8	1101 in base 16	1101 in base 2	111 in base 2

Es. 3 Eseguire le seguenti addizioni in binario:

34 + 77	225 + 63	229 + 111	153 + 197
---------	----------	-----------	-----------

Es. 4 Eseguire le seguenti addizioni in ottale:

$34 + 77$	$225 + 63$	$229 + 111$	$153 + 197$
-----------	------------	-------------	-------------

Es. 5 Eseguire le seguenti addizioni in esadecimale:

$34 + 77$	$225 + 63$	$229 + 111$	$153 + 197$
-----------	------------	-------------	-------------

Es. 6 Eseguire le seguenti sottrazioni in binario:

$84 - 37$	$225 - 63$	$229 - 111$	$321 - 123$	$-123 - 321$
-----------	------------	-------------	-------------	--------------

Es. 7 Eseguire le seguenti sottrazioni in ottale:

$84 - 37$	$225 - 63$	$229 - 111$	$321 - 123$	$123 - 321$
-----------	------------	-------------	-------------	-------------

Es. 8 Eseguire le seguenti sottrazioni in esadecimale:

$84 - 37$	$225 - 63$	$229 - 111$	$321 - 123$	$123 - 321$
-----------	------------	-------------	-------------	-------------

Es.9 Esegui la **moltiplicazione binaria** tra **63** e **30** e tra **59** e **31**, convertendo il risultato in decimale.

Es.10 Calcola i numeri rappresentati in modulo e segno, dalla parola binaria:

$(1\ 1110001)_2$	$(1\ 1100101)_2$	$(1\ 1101001)_2$	$(1\ 1100011)_2$	$(0\ 1100011)_2$
------------------	------------------	------------------	------------------	------------------

Es. 11 Rappresenta con 8 bit in modulo e segno, del numero.

-105	-113	+99	-5	+10
------	------	-----	----	-----

Es. 12 Scrivere in forma polinomiale i seguenti numeri:

$(1110001)_2$	$(1010)_{10}$	$(123)_8$	$(891)_{16}$
---------------	---------------	-----------	--------------

Es.13 Scrivi sotto forma di sommatoria l'espressione seguente:

$5 \cdot 8^2 + 6 \cdot 8^1 + 1 \cdot 8^0$	$4 \cdot 2^{-2} + 6 \cdot 2^{-1} + 1 \cdot 2^0 + 5 \cdot 2^1 + 9 \cdot 2^2 + 1 \cdot 2^3$
---	---