Informatica e Laboratorio di Programmazione

Sistema Binario - Esercizi

Alberto Ferrari

numeri naturali - conversione

- convertire in formato ***decimale*** i seguenti numeri ***binari***
  - 11, 101011, 111111, 10101010
- convertire in formato ***decimale*** i seguenti numeri ***ottali***
  - 12, 23, 345, 333, 560
- convertire in formato ***decimale*** i seguenti numeri ***esadecimali***
  - 12, DAB, 15D, FFFF, 51A
- convertire in ***binario*** i seguenti numeri ***decimali***
  - 45, 234, 67, 83, 972
- convertire in ***ottale*** e in ***esadecimale*** i numeri binari ottenuti dalla conversione dei numeri decimali di cui al punto precedente

soluzioni (1)

$\mathbf{11}_{due} = (1x2^1 + 1x2^0)_{dieci} = (2 + 1)_{dieci} = 3_{dieci}$

$\mathbf{101011}_{due} = (1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0)_{dieci} = (32 + 8 + 2 + 1)_{dieci} = 43_{dieci}$

$\mathbf{1100}_{due} = (1x2^3 + 1x2^2 + 0x2^1 + 0x2^0)_{dieci} = (8 + 4 + 0 + 0)_{dieci} = 12_{dieci}$

$\mathbf{111111}_{due} = (1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0)_{dieci} = (32 + 16 + 8 + 4 + 2 + 1)_{dieci} = 63_{dieci}$

$\mathbf{10101010}_{due} = (1x2^7 + 0x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 1x2^1 + 0x2^0)_{dieci} = (128 + 32 + 8 + 2)_{dieci} = 170_{dieci}$

$\mathbf{12}_{otto} = (1x8^1 + 2x8^0)_{dieci} = (8 + 2)_{dieci} = 10_{dieci}$

$\mathbf{23}_{otto} = (2x8^1 + 3x8^0)_{dieci} = (16 + 3)_{dieci} = 19_{dieci}$

$\mathbf{345}_{otto} = (3x8^2 + 4x8^1 + 5x8^0)_{dieci} = (3x64 + 32 + 5)_{dieci} = 229_{dieci}$

$\mathbf{333}_{otto} = (3x8^2 + 3x8^1 + 3x8^0)_{dieci} = (3x64 + 24 + 3)_{dieci} = 219_{dieci}$

$\mathbf{560}_{otto} = (5x8^2 + 6x8^1 + 0x8^0)_{dieci} = (5x64 + 48 + 0)_{dieci} = 368_{dieci}$

$\mathbf{12}_{sedici} = (1x16^1 + 2x16^0)_{dieci} = (16 + 2)_{dieci} = 18_{dieci}$

$\mathbf{DAB}_{sedici} = (13x16^2 + 10x16^1 + 11x16^0)_{dieci} = (13x256 + 160 + 11)_{dieci} = 3499_{dieci}$

$\mathbf{15D}_{sedici} = (1x16^2 + 5x16^1 + 13x16^0)_{dieci} = (256 + 80 + 13)_{dieci} = 349_{dieci}$

$\mathbf{FFFF}_{sedici} = (15x16^3 + 15x16^2 + 15x16^1 + 15x16^0)_{dieci} = (15x4096 + 15x256 + 15x16 + 15)_{dieci} = (61440 + 3840 + 240 + 15)_{dieci} = 65535_{dieci}$

$\mathbf{51A}_{sedici} = (5x16^2 + 1x16^1 + 10x16^0)_{dieci} = (5x256 + 16 + 10)_{dieci} = 1306_{dieci}$

soluzioni (2)

45_{dieci}

45/2 = 22 con resto 1
22/2 = 11 con resto 0
11/2 = 5 con resto 1
5/2 = 2 con resto 1
2/2 = 1 con resto 0
1/2 = 0 con resto 1

$45_{dieci} = 101101_{due}$

234_{dieci}

234/2 = 117 con resto 0
117/2 = 58 con resto 1
58/2 = 29 con resto 0
29/2 = 14 con resto 1
14/2 = 7 con resto 0
7/2 = 3 con resto 1
3/2 = 1 con resto 1
1/2 = 0 con resto 1

$234_{dieci} = 11101010_{due}$

67_{dieci}

67/2 = 33 con resto 1
33/2 = 16 con resto 1
16/2 = 8 con resto 0
8/2 = 4 con resto 0
4/2 = 2 con resto 0
2/2 = 1 con resto 0
1/2 = 0 con resto 1

$67_{dieci} = 1000011_{due}$

83_{dieci}

83/2 = 41 con resto 1
41/2 = 20 con resto 1
20/2 = 10 con resto 0
10/2 = 5 con resto 0
5/2 = 2 con resto 1
2/2 = 1 con resto 0
1/2 = 0 con resto 1

$83_{dieci} = 1010011_{due}$

soluzioni (3)

972_{dieci}

$972/2 = 486$ con resto 0
$486/2 = 243$ con resto 0
$243/2 = 121$ con resto 1
$121/2 = 60$ con resto 1
$60/2 = 30$ con resto 0
$30/2 = 15$ con resto 0
$15/2 = 7$ con resto 1
$7/2 = 3$ con resto 1
$3/2 = 1$ con resto 1
$1/2 = 0$ con resto 1

$972_{dieci} = 1111001100_{due}$

$101101_{due} \rightarrow 101\ 101 = 55_{otto}$
$11101010_{due} \rightarrow 11\ 101\ 010 = 352_{otto}$
$1000011_{due} \rightarrow 1\ 000\ 011 = 103_{otto}$
$1010011_{due} \rightarrow 1\ 010\ 011 = 123_{otto}$
$1111001100_{due} \rightarrow 1\ 111\ 001\ 100 = 1714_{otto}$

$101101_{due} \rightarrow 10\ 1101 = 2D_{sedici}$
$11101010_{due} \rightarrow 1110\ 1010 = EA_{sedici}$
$1000011_{due} \rightarrow 100\ 0011 = 43_{sedici}$
$1010011_{due} \rightarrow 101\ 0011 = 53_{sedici}$
$1111001100_{due} \rightarrow 11\ 1100\ 1100 = 3CC_{sedici}$

numeri interi - esercizi

- dati i seguenti numeri ***decimali*** interi positivi:
  - 55, 121, 16, 42
- rappresentarli come numeri ***binari*** su ***8 bit***
- determinare i numeri ***negativi*** corrispondenti in binario con le seguenti rappresentazioni:
  - ***modulo e segno***
  - in ***complemento*** a ***1***
  - in ***complemento*** a ***2***

soluzioni (1)

55_{dieci}

55/2 = 27 con resto 1
27/2 = 13 con resto 1
13/2 = 6 con resto 1
6/2 = 3 con resto 0
3/2 = 1 con resto 1
1/2 = 0 con resto 1

$55_{dieci} = 00110111_{due}$

121_{dieci}

121/2 = 60 con resto 1
60/2 = 30 con resto 0
30/2 = 15 con resto 0
15/2 = 7 con resto 1
7/2 = 3 con resto 1
3/2 = 1 con resto 1
1/2 = 0 con resto 1

$121_{dieci} = 01111001_{due}$

16_{dieci}

16/2 = 8 con resto 0
8/2 = 4 con resto 0
4/2 = 2 con resto 0
2/2 = 1 con resto 0
1/2 = 0 con resto 1

$16_{dieci} = 00010000_{due}$

42_{dieci}

42/2 = 21 con resto 0
21/2 = 10 con resto 1
10/2 = 5 con resto 0
5/2 = 2 con resto 1
2/2 = 1 con resto 0
1/2 = 0 con resto 1

$42_{dieci} = 00101010_{due}$

soluzioni (2)

CODIFICA IN MODULO E SEGNO

$55_{dieci} = 0\,0\,1\,1\,0\,1\,1\,1_{due}$ $-55_{dieci} = 1\,0\,1\,1\,0\,1\,1\,1_{due}$	$121_{dieci} = 0\,1\,1\,1\,1\,0\,0\,1_{due}$ $-121_{dieci} = 1\,1\,1\,1\,1\,0\,0\,1_{due}$
$16_{dieci} = 0\,0\,0\,1\,0\,0\,0\,0_{due}$ $-16_{dieci} = 1\,0\,0\,1\,0\,0\,0\,0_{due}$	$42_{dieci} = 0\,0\,1\,0\,1\,0\,1\,0_{due}$ $-42_{dieci} = 1\,0\,1\,0\,1\,0\,1\,0_{due}$

CODIFICA IN COMPLEMENTO A 1

$55_{dieci} = 0\,0\,1\,1\,0\,1\,1\,1_{due}$ $-55_{dieci} = 1\,1\,0\,0\,1\,0\,0\,0_{due}$	$121_{dieci} = 0\,1\,1\,1\,1\,0\,0\,1_{due}$ $-121_{dieci} = 1\,0\,0\,0\,0\,1\,1\,0$
$16_{dieci} = 0\,0\,0\,1\,0\,0\,0\,0_{due}$ $-16_{dieci} = 1\,1\,1\,0\,1\,1\,1\,1_{due}$	$42_{dieci} = 0\,0\,1\,0\,1\,0\,1\,0_{due}$ $-42_{dieci} = 1\,1\,0\,1\,0\,1\,0\,1_{due}$

CODIFICA IN COMPLEMENTO A 2

$55_{dieci} = 0\,0\,1\,1\,0\,1\,1\,1_{due}$ $-55_{dieci} = 1\,1\,0\,0\,1\,0\,0\,1_{due}$	$121_{dieci} = 0\,1\,1\,1\,1\,0\,0\,1_{due}$ $-121_{dieci} = 1\,0\,0\,0\,0\,1\,1\,1_{due}$
$16_{dieci} = 0\,0\,0\,1\,0\,0\,0\,0_{due}$ $-16_{dieci} = 1\,1\,1\,1\,0\,0\,0\,0_{due}$	$42_{dieci} = 0\,0\,1\,0\,1\,0\,1\,0_{due}$ $-42_{dieci} = 1\,1\,0\,1\,0\,1\,1\,0_{due}$

operazioni

- eseguire la ***somma*** dei numeri binari in complemento a 2 codificati su 8 bit
  che corrispondono ai numeri 16_{dieci} e $–42_{dieci}$
- eseguire la ***somma*** dei numeri binari in complemento a 2 codificati su 6 bit
  che corrispondono ai numeri -5_{dieci} e $–28_{dieci}$

soluzioni (1)

Somma di numeri in complemento a 2

[16] **0 0 0 1 0 0 0 0 +**

[-42] **1 1 0 1 0 1 1 0**

1 1 1 0 0 1 1 0

1 1 1 0 0 1 1 0

Segno: negativo (1)

Modulo del numero: si ottiene facendo il complemento a 2 di

1 1 0 0 1 1 0 ➔ 0 0 1 1 0 1 0 ➔ 16 + 8 + 2 = 26

Quindi **–26**$_{\text{dieci}}$

Somma di numeri in complemento a 2

5$_{\text{dieci}}$ **= 000101**$_{\text{due}}$ **su n = 6 bit**

28$_{\text{dieci}}$ **= 011100**$_{\text{due}}$ **su n = 6 bit**

-5$_{\text{dieci}}$ **= 111011**$_{\text{due}}$ **su n = 6 bit**

-28$_{\text{dieci}}$ **= 100100**$_{\text{due}}$ **su n = 6 bit**

[-5] **1 1 1 0 1 1 +**

[-28] **1 0 0 1 0 0**

(1) 0 1 1 1 1 1

rip 1 0 ➔ overflow

0 1 1 1 1 1

Segno: positivo (0)

Modulo del numero: si ottiene facendo il complemento a 2 di

1 1 1 1 1 ➔ 0 0 0 0 1 ➔ 1

Quindi: risultato = +1$_{\text{dieci}}$ che non è ciò che ci aspettavamo

numeri frazionari

- convertire in ***decimale*** i seguenti numeri ***frazionari*** binari:
  - 0.111_{due}, 0.0101_{due}, 0.00011_{due}
- convertire in ***binario*** (su 6 bit – cifre dopo la virgola) i seguenti numeri frazionari decimali:
  - 0.226_{dieci}, 0.349_{dieci}, 0.712_{dieci}
- esprimere i numeri
  - 13.25_{dieci} e 189.8123_{dieci}
    in forma normalizzata in base 10 e in base 2

soluzioni (1)

$\mathbf{0.111}_{due} = (1x2^{-1} + 1x2^{-2} + 1x2^{-3})_{dieci} = (0.5 + 0.25 + 0.125)_{dieci} = 0.875_{dieci}$

$\mathbf{0.0101}_{due} = (0x2^{-1} + 1x2^{-2} + 0x2^{-3} + 1x2^{-4})_{dieci} = (0.25 + 0.0625)_{dieci} = 0.3125_{dieci}$

$\mathbf{0.00011}_{due} = (0x2^{-1} + 0x2^{-2} + 0x2^{-3} + 1x2^{-4} + 1x2^{-5})_{dieci} = (0.0625 + 0.03125)_{dieci} = 0.09375_{dieci}$

$\mathbf{0.226}_{dieci}$

0.226x2 = 0.452 p.f. 0.452 p.i. 0

0.452x2 = 0.904 p.f. 0.904 p.i. 0

0.904x2 = 1.808 p.f. 0.808 p.i. 1

0.808x2 = 1.616 p.f. 0.616 p.i 1

0.616x2 = 1.232 p.f 0.232 p.i. 1

0.232x2 = 0.464 p.f. 0.464 p.i. 0

$\Rightarrow$ $0.226_{dieci} = 0.001110_{due}$

soluzioni (2)

0.349_{dieci}

0.349x2 = 0.698	p.f. 0.698 p.i. 0
0.698x2 = 1.396	p.f. 0.396 p.i. 1
0.396x2 = 0.792	p.f. 0.792 p.i. 0
0.792x2 = 1.584	p.f. 0.584 p.i 1
0.584x2 = 1.168	p.f 0.168 p.i. 1
0.168x2 = 0.336	p.f. 0.336 p.i. 0

$\Rightarrow$ $0.349_{dieci} = 0.010110_{due}$

0.712_{dieci}

0.712x2 = 1.424	p.f. 0.424 p.i. 1
0.424x2 = 0.848	p.f. 0.848 p.i. 0
0.848x2 = 1.696	p.f. 0.696 p.i. 1
0.696x2 = 1.392	p.f. 0.392 p.i 1
0.392x2 = 0.784	p.f 0.784 p.i. 0
0.784x2 = 1.568	p.f. 0.568 p.i. 1

$\Rightarrow$ $0.712_{dieci} = 0.101101_{due}$

soluzioni (3)

13.25 ➔ in forma normalizzata 0.1325×10^{2}
189.8123 ➔ in forma normalizzata 0.1898123×10^{3}

$13_{dieci} = 1101_{due}$
$0.25_{dieci} = 0.01_{due}$ infatti:
$0.25 \times 2 = 0.50$ p.i. 0
$0.50 \times 2 = 1.0$ p.i. 1
da cui
$13.25_{dieci} = 1101.01_{due}$ che, in forma normalizzata, diventa 0.110101×10^{100}
(dove $m = 0.110101_{due}$, $e = 100_{due}$ e la base $b = 10_{due}$)

soluzioni (4)

189.8123_{dieci} → lo trasformiamo in base due usando 8 cifre binarie per la parte intera e 4 bit per la parte frazionaria

189_{dieci}

189/2 = 94	con resto 1
94/2 = 47	con resto 0
47/2 = 23	con resto 1
23/2 = 11	con resto 1
11/2 = 5	con resto 1
5/2 = 2	con resto 1
2/2 = 1	con resto 0
1/2 = 0	con resto 1

$189_{dieci} = 1011\ 1101_{due}$

0.8123_{dieci}

0.8123x2 = 1.6246	p.f. 0.6246 p.i. 1
0.6246x2 = 1.2492	p.f. 0.2492 p.i. 1
0.2492x2 = 0.4984	p.f. 0.4984 p.i. 0
0.4984x2 = 0.9968	p.f. 0.9968 p.i 0

$0.8123_{dieci} = 0.1100_{due}$

Da cui $189.8123_{dieci} = 10111101.1100_{due}$ che, in forma normalizzata, diventa $0.101111011 \times 10^{1000}$

numeri reali (1)

- convertire il numero ***decimale*** `-30,375` in formato a virgola mobile IEEE 754 (precisione singola)
  - *IEEE 754 single precision 32 bit (1 x segno, 8 x esponente, 23 x frazione)*

soluzione

$-30.375 = (-11110.011)_{binario} = (-1.1110011)binario \times 2^4$

$= (-1)1 \times (1 + 0.1110011) \times 2^{(131-127)}$

- *il formato IEEE 754 utilizza lo schema di rappresentazione*
  $(-1)segno \times (1 + significando) \times 2^{(esponente-127)}$
- *Abbiamo:*

  segno = 1

  esponente = 131 = (10000011)binario

  signif icando = (11100110000000000000000)binario

- $\Rightarrow (-30.375)_{10} = (1\ 10000011\ 11100110000000000000000)_{binario}$

numeri reali (2)

- che numero rappresenta **01000110010001100000000000000000** in forma to IEEE 754 (*single precision*)?
- ***soluzione***
- ***segno*** **0** $\Rightarrow$ segno **+**
- ***esponente*** **10001100** $\rightarrow$ **140**$_{\text{decimale}}$
  - (bisogna sottrarre la polarizzazione (**127**) per ottenere il vero esponente, cioè **13**
- ***significando*** **10001100000000000000000**
  - $1+2^{-1}+2^{-5}+2^{-6}=1,546875$
- il numero è $+1 \times 1,546875 \times 2^{13} = 12672,0$

half-precision IEEE 754

- determinare la rappresentazione binaria di **92.375**
- *codifica: 1 bit per il segno, 5 per l'esponente, 10 per la parte frazionaria*
- parte intera **92** ⇒ **1011100**
- parte frazionaria **0.375** ⇒ **0.011**
- **1011100.011** normalizzato ⇒ **1.011100011 * 2^6** (esponente **110**)
- agg. esponente **01111** +
- **00110** =
- **10101**
- segno **0** esponente(5 bit) **10101** mantissa (10 bit) **0111000110**
- **0101010111000110**
- **0101 0101 1100 0110** binario
- **5 5 C 6** esadecimanle

Unicode – UTF-8

- riferimenti
  - https://www.utf8-chartable.de/unicode-utf8-table.pl
  - http://www.endmemo.com/unicode/unicodeconverter.php

Intervallo Unicode esadecimale	UTF-16	UTF-8 binario	Note
0x000000-0x00007F	00000000 0XXXXXXX	0XXXXXXX	Caratteri equivalenti al codice ASCII; i byte iniziano con 0 e da soli indicano un carattere
0x000080-0x0007FF	00000XXX XXXXXXXX	110XXXXX 10XXXXXX	il primo byte inizia per 110 o 1110, il successivo(i) con 10 e devono essere concatenati per formare un carattere
0x000800-0x00FFFF	XXXXXXXX XXXXXXXX	1110XXXX 10XXXXXX 10XXXXXX	
0x010000-0x10FFFF	110110XX XXXXXXXX 110111XX XXXXXXXX	11110XXX 10XXXXXX 10XXXXXX 10XXXXXX	Confronto tra UTF-16 e UTF-8: UTF-16 richiede l'uso di coppie surrogate: viene sottratto il valore esadecimale 0x10000, in modo che la sequenza dei bit non coincida con quella usata da UTF-8

esercizi

ALGEBRA DI BOOLE

tavole di verità

- costruire la tavola di verità della funzione
  F(a,b) = ab+¬c = a and b or not c

a	b	c
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

a	b	c	ab
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a	b	c	ab	$\bar{c}$
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

a	b	c	ab	$\bar{c}$	ab+$\bar{c}$
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

forme canoniche

- data la tabella di verità di ***F***:
  - ricavare l'espressione logica in *prima forma canonica* ***somma di prodotti***
  - ricavare l'espressione logica in *seconda forma canonica* ***prodotti di somme***
- *soluzione*
  - ***somma di prodotti (SP): si considerano le righe a 1***
    - **F(a,b,c) = (¬a¬bc)+(¬ab¬c)+(¬abc)+(abc)**
  - ***prodotti di somme (PS): si considerano le righe a 0***
    - **F(a,b,c) = (a+b+c)(¬a+b+c)(¬a+b+¬c)(¬a+¬b+c)**

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1