Codifica e rappresentazione dell’informazione: numeri interi, numeri reali e testo

Esercitazione 1
Architettura degli Elaboratori

Esercizio 1

Convertire i numeri 0x52, 0x1010101, 0xF3B4C (esadecimali) in binario

Esercizio 1 – soluzione

Convertiamo i numeri 0x52, 0x1010101 (esadecimali) in binario

Ricorda che:

$1_{10} = 1_{16} = 0001_2$
$2_{10} = 2_{16} = 0010_2$
……
$9_{10} = 9_{16} = 1001_2$
$10_{10} = A_{16} = 1010_2$
$11_{10} = B_{16} = 1011_2$
$12_{10} = C_{16} = 1100_2$
$13_{10} = D_{16} = 1101_2$
$14_{10} = E_{16} = 1110_2$
$15_{10} = F_{16} = 1111_2$

0x52 $= (0101\ 0010)_2$

Esercizio 1 – soluzione

Convertiamo i numeri 0x52, 0x1010101 (esadecimali) in binario

Ricorda che:

$1_{10} = 1_{16} = 0001_2$
$2_{10} = 2_{16} = 0010_2$
......
$9_{10} = 9_{16} = 1001_2$
$10_{10} = A_{16} = 1010_2$
$11_{10} = B_{16} = 1011_2$
$12_{10} = C_{16} = 1100_2$
$13_{10} = D_{16} = 1101_2$
$14_{10} = E_{16} = 1110_2$
$15_{10} = F_{16} = 1111_2$

0x1010101 $= (0001\ 0000\ 0001\ 0000\ 0001\ 0000\ 0001)_2$

Esercizio 1 – soluzione

Convertiamo i numeri 0x52, 0x1010101 (esadecimali) in binario

Ricorda che:

$1_{10} = 1_{16} = 0001_2$
$2_{10} = 2_{16} = 0010_2$
......
$9_{10} = 9_{16} = 1001_2$
$10_{10} = A_{16} = 1010_2$
$11_{10} = B_{16} = 1011_2$
$12_{10} = C_{16} = 1100_2$
$13_{10} = D_{16} = 1101_2$
$14_{10} = E_{16} = 1110_2$
$15_{10} = F_{16} = 1111_2$

0xF3B4C = (1111 0011 1011 0100 1100)

Esercizio 2

Convertire un numero decimale (es. 98_{10}) in esadecimale

Esercizio 2

Convertire un numero decimale in esadecimale

→

Tre metodi per risolvere l'esercizio:

Esercizio 2 – soluzione (metodo 1)

Convertire il numero decimale in esadecimale, scrivendo la sequenza di resti (a partire dall'ultimo) della divisione per 16

98 : 16 = 6 con resto 2
6 : 16 = 0 con resto 6

e quindi $98_{10} = 62_{16}$

Esercizio 2 – soluzione (metodo 2)

Convertire il numero decimale in binario:

- ripetendo la divisione per 2 fino a che il risultato è 0
- scrivendo i resti a partire dall'ultimo

e, poi, convertire in cifra esadecimale i gruppi di 4 cifre binarie

98:2=49	resto 0
49:2=24	resto 1
24:2=12	resto 0
12:2=6	resto 0
6:2=3	resto 0
3:2=1	resto 1
1:2=0	resto 1

$98_{10} = 1100010_2$

$= 01100010_2$

$= 0110\ 0010_2$

$= 0x62$

Esercizio 2 – soluzione (metodo 3)

Convertire il numero decimale, prima, in binario:

- scrivendolo come somma di potenze di 2
- scrivendo 1 in corrispondenza della potenza di 2 presente

e, poi, convertire in cifra esadecimale i gruppi di 4 cifre binarie (a partire da destra e aggiungendo eventuali 0 all'inizio)

$$99_{10} = 64 + 32 + 2 + 1$$

$$= 1100011_2$$

$$= 01100011_2$$

$$= (\ 6 \quad 3\ )_{16}$$

$$= 63_{16} = \text{0x63}$$

Esercizio 3

Se il numero decimale 100 fosse rappresentato in binario, quale sarebbe il valore dei suoi bit in posizione 3 e in posizione 5?

Esercizio 3 – soluzione

Se il numero decimale 100 fosse rappresentato in binario, quale sarebbe il valore dei suoi bit in posizione 3 e in posizione 5?

Poichè il bit in prima posizione è quello che corrisponde a 2^0, i bit in posizione 3 e 5 sono quelli che corrispondono a 2^2 e 2^4

Essendo

$$\begin{aligned} 100_{10} &= 64 \quad + 32 \quad + 4 \\ &= 2^6 \quad +2^5 \quad +2^2 \\ &= 1100100_2 \end{aligned}$$

i bit in posizione 3 e in posizione 5 hanno rispettivamente valore 1 e 0

Esercizio 4

Effettuare le conversioni

$(110011101111011)_2 = ?_8 = ?_{16}$ da binario a ottale e esadecimale

$(0x71A9)_{16} = ?_8 = ?_2$ da esadecimale a ottale e binario

$(736)_8 = ?_{16} = ?_2$ da ottale a esadecimale e binario

Esercizio 4 – soluzione

Effettuare le conversioni

$(110011101111011)_2 = ?_8 = ?_{16}$ — da binario a ottale e esadecimale

$(110\ 011\ 101\ 111\ 011)_2 = (63573)_8$ — - consideriamo gruppi di 3 cifre (a partire dal bit più a destra), eventualmente completando a sinistra con 0, e convertiamo da binario a ottale i singoli gruppi)

$(0110\ 0111\ 0111\ 1011)_2 = (677B)_{16}$ — - consideriamo gruppi di 4 cifre (a partire dal bit più a destra), eventualmente completando a sinistra con 0, e convertiamo da binario a esadecimale i singoli gruppi)

$(0x71A9)_{16} = ?_8 = ?_2$ — da esadecimale a ottale e binario

$(0x71A9)_{16} = (111\ 0001\ 1010\ 1001)_2$ — - prima, convertiamo da esadecimale a binario (convertendo le singole cifre esadecimali nelle 4 cifre binarie corrispodnenti)

$(111\ 000\ 110\ 101\ 001)_2 = (70651)_8$ — - quindi, convertiamo gruppi di 3 cifre binarie in corrispondenti cifre ottali

$(736)_8 = ?_{16} = ?_2$ — da ottale a esadecimale e binario

$(736)_8 = (111\ 011\ 110)_2$ — - prima, da ottale a binario …

$(0001\ 1101\ 1110)_2 = (1DE)_{16}$ — - quindi, da binario a esadecimale …

Esercizio 5

Convertire i numeri decimali 33, 10, 59, -59, -20 e -114 in binario modulo e segno su 8 bit

Esercizio 5 – soluzione

Convertire i numeri decimali 33, 10, 59, -59, -20 e -114 in binario modulo e segno su 8 bit

$33 = (32+1)_{10} = 00100001_{MS}$

$10 = (8+2)_{10} = 00001010_{MS}$

$59 = (32+16+8+2+1)_{10} = 00111011_{MS}$

$-59 = -(32+16+8+2+1)_{10} = 10111011_{MS}$

$-20 = -(16+4)_{10} = 10010100_{MS}$

$-114 = -(64+32+16+2)_{10} = 11110010_{MS}$

Prima cifra è 1 se il numero è negativo; le restanti 7 cifre per la codifica binaria

Esercizio 6

Convertire i numeri decimali 33, 10, 59, -59, -20 e -114 in binario complemento a 2 su 8 bit

Esercizio 6 – soluzione

Convertire i numeri decimali 33, 10, 59, -59, -20 e -114 in binario complemento a 2 su 8 bit

33 $= 00100001_{MS} = 00100001_{CA2}$

10 $= 00001010_{MS} = 00001010_{CA2}$

59 $= 00111011_{MS} = 00111011_{CA2}$

Codifica MS e CA2 di numeri positivi sono uguali!

\- 59 = inversione bit a bit e + 1 di 59 (00111011) $\rightarrow$ $11000100 + 1 = 11000101_{CA2}$

\- 20 = inversione bit a bit e + 1 di 20 ($20_{10} = 16+4 = 00010100_2$) $\rightarrow$ $11101011 + 1 = 11101100_{CA2}$

\- 114 = inversione bit a bit e + 1 di 114 ($114_{10} = 64+32+16+2$) $= 10001101 + 1 = 10001110_{CA2}$

Esercizio 7

Quale numero decimale rappresentano i numeri binari in complemento a 2, 11000101_{CA2} , 1101_{CA2} , 101_{CA2} e 111101_{CA2} ?

Esercizio 7 – soluzione

Quale numero decimale rappresentano i numeri binari in complemento a 2, 11000101_{CA2} , 1101_{CA2} , 101_{CA2} e 111101_{CA2} ?

$1100\ 0101_{CA2}$ = $-2^7 + 2^6 + 2^2 + 2^0$ =
= $-128 + 64 + 4 + 1$ = -59_{10}

1101_{CA2} = $-2^3 + 2^2 + 2^0$ =
= $-8 + 4 + 1$ = -3_{10}

101_{CA2} = - [(inversione bit a bit e +1) → 010 + 1] = $-(011)_2 = -3_{10}$

111101_{CA2} = $-2^5 + 2^4 + 2^3 + 2^2 + 2^0$ = -3_{10}
= $-(000010 + 1)_2 = -(000011)_2$ = -3_{10}

Metodi alternativi

Esercizio 8

Date le seguenti coppie di numeri binari, considerarli prima in MS e poi in CA2; per ciascuna coppia si determini qual è il maggiore tra i valori rappresentati (o se rappresentano il medesimo valore)

		MS	CA2
a)	01001 ? 10001	?	?
b)	10110 ? 11010	?	?
c)	101 ? 11101	?	?
d)	11111 ? 10001	?	?

Esercizio 8 – soluzione

Date le seguenti coppie di rappresentazioni, considerarle prima in MS e poi in CA2; per ciascuna coppia si determini qual è il maggiore tra i valori rappresentati

					MS	CA2
a)	01001	?	10001		>	>
b)	10110	?	11010		>	<
c)	101	?	11101		?	?
d)	11111	?	10001		?	?

a) Sia per la rappresentazione MS sia per quella CA2, il primo numero è positivo e il secondo è negativo
b) Entrambi i numeri sono negativi ma considerando la rappresentazione MS il modulo del primo numero è minore di quello del secondo (seconda cifra rispettivamente, 0 e 1) e quindi il primo dei due numeri negativi è maggiore del secondo; considerando invece la rappresentazione CA2, operando l'inversione dei bit avremo la situazione opposta

Esercizio 8 – soluzione

Date le seguenti coppie di rappresentazioni, considerarle prima in MS e poi in CA2; per ciascuna coppia si determini qual è il maggiore tra i valori rappresentati

				MS	CA2
a)	01001	?	10001	>	>
b)	10110	?	11010	>	<
c)	101	?	11101	>	=
d)	11111	?	10001	<	>

c) Nel caso MS, abbiamo come nel caso precedente, due numeri negativi in cui quello più a sinistra ha modulo minore e quindi è maggiore di quello più a destra. Nella rappresentazione CA2, per poter confrontare i due numeri, aggiungiamo due 1 → otteniamo, quindi, due numeri uguali.

d) Entrambi i numeri sono negativi ma considerando la rappresentazione MS il modulo del primo numero è maggiore di quello del secondo (seconda cifra rispettivamente, 1 e 0) e quindi il primo dei due numeri negativi è minore del secondo; considerando invece la rappresentazione CA2, operando l'inversione dei bit avremo la situazione opposta ovvero un numero con modulo che inizia per 0 a sinistra e quindi minore (in modulo) del numero a destra che ha modulo che inizia con 1

Esercizio 9

Eseguire $53_{10} – 35_{10}$ in complemento a 2 su 8 bit

Esercizio 9 – soluzione

Eseguire $53_{10} - 35_{10}$ in complemento a 2 su 8 bit

$53_{10} - 35_{10} = 53_{10} + (-35_{10})$

$35_{10} = 00100011_2$ *complementando*: $-35_{10} = 11011101_2$

riporti

$$
\begin{array}{rcrcr}
 & & & & 11111101 \\
53_{10}\ - & & 53_{10}\ + & & 00110101_2\ + \\
35_{10}\ = & \Rightarrow & (-35)_{10}\ = & \Rightarrow & 11011101_2\ = \\
\hline
18_{10} & & 18_{10} & & (100010010_2) \bmod 2^8
\end{array}
$$

$00010010_2 = 18_{10}$

Esercizio 10

Eseguire $15_{10} - 38_{10}$ in complemento a 2 su 8 bit

Esercizio 10 – soluzione

Eseguire $15_{10} - 38_{10}$ in complemento a 2 su 8 bit

$38_{10} = 00100110_2$ *complementando*: $-38_{10} = 11011010_2$

$$\begin{array}{r} 15_{10} \;\; - \\ 38_{10} \;\; = \\ \hline -23_{10} \end{array} \Rightarrow \begin{array}{r} 15_{10} \;\; + \\ (-38)_{10} \;\; = \\ \hline -23_{10} \end{array} \Rightarrow \begin{array}{r} 00011110 \\ 00001111_2 \; + \\ 11011010_2 \; = \\ \hline (011101001_2) \bmod 2^8 \end{array}$$

per verificare il risultato, convertiamo in decimale
$= -2^7 + 2^6 + 2^5 + 2^3 + 2^0 = -23_{10}$

Esercizio 11

Quale numero decimale rappresenta il numero binario in complemento a 2

$$1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 0000\ 1100_2$$

Esercizio 11 – soluzione

Quale numero decimale rappresenta il numero binario in complemento a 2

$$1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 0000\ 1100_2$$

$$\downarrow$$

$$0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 1111\ 0100_2 =$$

$$2^2 + 2^4 + 2^5 + 2^6 + 2^7 + 2^8 = 500_{10}$$

Soluzione: il numero è -500_{10}

oppure $= - 2^9 + 2^3 + 2^2 = - 512 + 8 + 4 = - 500$

ultimo 1 della sequenza di 1 con cui inizia il numero

Esercizio 12

Calcolare 0xA789 + 0x987 (somma tra numeri esadecimali)

Esercizio 12 – soluzione

Calcolare 0xA789 + 0x987 (somma tra numeri esadecimali)

Partendo dalle cifre più a destra, considerando i riporti e che le cifre esadecimali sono 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

9+7	= 16	→ 0 (=16 - 16) con riporto 1
8+8+ 1 (riporto)	= 17	→ 1 (=17 – 16) con riporto 1
7+9+ 1 (riporto)	= 17	→ 1 (=17 – 16) con riporto 1
A+ 1 (riporto)	= 11	→ B

0xA789 + 0x987 = 0xB110

Esercizio 13

- Convertire in esadecimale il numero $\mathbf{101111101101_2}$ e sommarlo a $\mathbf{3C9_{16}}$

- Convertire in binario il numero $\mathbf{3C9_{16}}$ e sommarlo a $\mathbf{101111101101_2}$

- Verificare che i due risultati sono uguali

Esercizio 13 – soluzione

- Convertire in esadecimale il numero **101111101101_2** e sommarlo a **$3C9_{16}$**

→ BED +

3C9 =

FB6 $= (15*16^2 + 11*16^1 + 6*16^0)_{10}$

- Convertire in binario il numero **$3C9_{16}$** e sommarlo a **101111101101_2**

→ 001111001001 +

101111101101 =

111110110110 $= (2^{11} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^5 + 2^4 + 2^2 + 2^1)_{10}$

- Senza fare calcoli, possiamo verificare che i due risultati sono uguali dal fatto che 1111_2=0xF; 1011_2=0xB; 0110_2=0x6

Esercizio 14

- Effettuare le seguenti conversioni:

$432_{10} = ?_{5}$

$12_{5} = ?_{10}$

$1024_{10} = ?_{16}$

$123_{8} = ?_{5}$

Esercizio 14 - soluzione

$432_{10} = ?_5$
432 : 5 = 86 con resto 2
86 : 5 = 17 con resto 1
17 : 5 = 3 con resto 2
3 : 5 = 0 con resto 3 → $432_{10} = 3212_5$

$12_5 = ?_{10}$
$12_5 = 1*5^1 + 2*5^0$ → $12_5 = 7_{10}$

$1024_{10} = ?_{16}$
1024 : 16 = 64 con resto 0
64 : 16 = 4 con resto 0
4 : 16 = 0 con resto 4 → $1024_{10} = 400_{16}$

$123_8 = ?_5$
prima si converte il numero in base 10 …..
$123_8 = 1*8^2 + 2*8^1 + 3*8^0 = 64 + 16 + 3 = 83_{10}$
… e poi si converte il risultato in base 5
83 : 5 = 16 con resto 3
16 : 5 = 3 con resto 1
3 : 5 = 0 con resto 3

→ $123_8 = 313_5$

Esercizio 15

Quali sono i numeri decimali rappresentati in eccesso 128 dai numeri binari 00000000 e 00011100 ?

Esercizio 15 – soluzione

Quali sono i numeri decimali rappresentati in eccesso 128 dai numeri binari 00000000 e 00011100 ?

00000000 = 0 -128 = -128

00011100 = 16 + 8 + 4 -128 = -100

Esercizio 16

Rappresentare i numeri -20 e 20 in eccesso 128

Esercizio 16 – soluzione

Rappresentare i numeri -20 e 20 in eccesso 128

-20 = -20 + 128 = 108 = riscriviamo 108 come somma di potenze di 2
= 64 + 32 + 8 + 4
= 01101100

20 = 20 + 128 = 148 = 128 + 16 + 4 = 1001 0100

Esercizio 17

Su un disco di 20 GB quante immagini da 4 MB possono essere memorizzate?

Esercizio 17 - soluzione

Nel mondo dell'informatica, per indicare le **dimensioni delle memorie**, si utilizzano le potenze di 2

Kbyte = 2^{10} byte = 1024 byte
Mbyte = 2^{20} byte = 1.048.576 byte
Gbyte = 2^{30} byte = 1.073.741.824 byte
Tbyte = 2^{40} byte
Pbyte = 2^{50} byte

Su un disco di 20 GB quante immagini da 4 MB possono essere memorizzate?

20 GB = 20 * 1024 MB = $5 * 2^2 * 2^{10}$ MB

num. immagini = $5 * 2^2 * 2^{10}$ MB / 2^2 MB

= $5 * 2^{10}$

Esercizio 18

In quali dei seguenti casi si può ottenere overflow?

- somma di due numeri con segno concorde?
- somma di due numeri con segno discorde?
- sottrazione di due numeri con segno concorde?
- sottrazione di due numeri con segno discorde?

Esercizio 18 – soluzione

In quali dei seguenti casi si può ottenere overflow?

- somma di due numeri con segno concorde? → **Sì**
- somma di due numeri con segno discorde? → **NO**
- sottrazione di due numeri con segno concorde? → **NO**
- sottrazione di due numeri con segno discorde? → **Sì**

Esercizio 19

Codificare in numeri esadecimali e decimali secondo lo standard ASCII le seguenti stringhe di caratteri

Bit

2012

x=7*(y)

Facendo riferimento ad una delle tabelle di codifica ASCII (sotto sono riportate una tabella di codifica in decimale, esadecimale, ottale e binario; e, sulla destra, una seconda tabella con codifica in esadecimale), cerco i codici dei singoli caratteri che compongono la stringa da convertire e scrivo la codifica della stringa concatenando i codici

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	0000000	NUL	32	0x20	040	0100000	space	64	0x40	100	1000000	@	96	0x60	140	1100000	`
1	0x01	001	0000001	SOH	33	0x21	041	0100001	!	65	0x41	101	1000001	A	97	0x61	141	1100001	a
2	0x02	002	0000010	STX	34	0x22	042	0100010	"	66	0x42	102	1000010	B	98	0x62	142	1100010	b
3	0x03	003	0000011	ETX	35	0x23	043	0100011	#	67	0x43	103	1000011	C	99	0x63	143	1100011	c
4	0x04	004	0000100	EOT	36	0x24	044	0100100	$	68	0x44	104	1000100	D	100	0x64	144	1100100	d
5	0x05	005	0000101	ENQ	37	0x25	045	0100101	%	69	0x45	105	1000101	E	101	0x65	145	1100101	e
6	0x06	006	0000110	ACK	38	0x26	046	0100110	&	70	0x46	106	1000110	F	102	0x66	146	1100110	f
7	0x07	007	0000111	BEL	39	0x27	047	0100111	'	71	0x47	107	1000111	G	103	0x67	147	1100111	g
8	0x08	010	0001000	BS	40	0x28	050	0101000	(	72	0x48	110	1001000	H	104	0x68	150	1101000	h
9	0x09	011	0001001	TAB	41	0x29	051	0101001	)	73	0x49	111	1001001	I	105	0x69	151	1101001	i
10	0x0A	012	0001010	LF	42	0x2A	052	0101010	*	74	0x4A	112	1001010	J	106	0x6A	152	1101010	j
11	0x0B	013	0001011	VT	43	0x2B	053	0101011	+	75	0x4B	113	1001011	K	107	0x6B	153	1101011	k
12	0x0C	014	0001100	FF	44	0x2C	054	0101100	,	76	0x4C	114	1001100	L	108	0x6C	154	1101100	l
13	0x0D	015	0001101	CR	45	0x2D	055	0101101	-	77	0x4D	115	1001101	M	109	0x6D	155	1101101	m
14	0x0E	016	0001110	SO	46	0x2E	056	0101110	.	78	0x4E	116	1001110	N	110	0x6E	156	1101110	n
15	0x0F	017	0001111	SI	47	0x2F	057	0101111	/	79	0x4F	117	1001111	O	111	0x6F	157	1101111	o
16	0x10	020	0010000	DLE	48	0x30	060	0110000	0	80	0x50	120	1010000	P	112	0x70	160	1110000	p
17	0x11	021	0010001	DC1	49	0x31	061	0110001	1	81	0x51	121	1010001	Q	113	0x71	161	1110001	q
18	0x12	022	0010010	DC2	50	0x32	062	0110010	2	82	0x52	122	1010010	R	114	0x72	162	1110010	r
19	0x13	023	0010011	DC3	51	0x33	063	0110011	3	83	0x53	123	1010011	S	115	0x73	163	1110011	s
20	0x14	024	0010100	DC4	52	0x34	064	0110100	4	84	0x54	124	1010100	T	116	0x74	164	1110100	t
21	0x15	025	0010101	NAK	53	0x35	065	0110101	5	85	0x55	125	1010101	U	117	0x75	165	1110101	u
22	0x16	026	0010110	SYN	54	0x36	066	0110110	6	86	0x56	126	1010110	V	118	0x76	166	1110110	v
23	0x17	027	0010111	ETB	55	0x37	067	0110111	7	87	0x57	127	1010111	W	119	0x77	167	1110111	w
24	0x18	030	0011000	CAN	56	0x38	070	0111000	8	88	0x58	130	1011000	X	120	0x78	170	1111000	x
25	0x19	031	0011001	EM	57	0x39	071	0111001	9	89	0x59	131	1011001	Y	121	0x79	171	1111001	y
26	0x1A	032	0011010	SUB	58	0x3A	072	0111010	:	90	0x5A	132	1011010	Z	122	0x7A	172	1111010	z
27	0x1B	033	0011011	ESC	59	0x3B	073	0111011	;	91	0x5B	133	1011011	[	123	0x7B	173	1111011	{
28	0x1C	034	0011100	FS	60	0x3C	074	0111100	<	92	0x5C	134	1011100	\	124	0x7C	174	1111100	|
29	0x1D	035	0011101	GS	61	0x3D	075	0111101	=	93	0x5D	135	1011101	]	125	0x7D	175	1111101	}
30	0x1E	036	0011110	RS	62	0x3E	076	0111110	>	94	0x5E	136	1011110	^	126	0x7E	176	1111110	~
31	0x1F	037	0011111	US	63	0x3F	077	0111111	?	95	0x5F	137	1011111	_	127	0x7F	177	1111111	DEL

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80			‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ			
90		‘	’	“	”	•	–	—	˜	™	š	›	œ			Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Esercizio 19 – soluzione

Codificare in numeri esadecimali e decimali secondo lo standard ASCII le seguenti stringhe di caratteri

	Esadecimale	Decimale
Bit	42 69 74	066 105 116
2012	32 30 31 32	050 048 049 050
x=7*(y)	78 3D 37 2A 28 79 29	120 061 055 042 040 121 041

Esercizio 20

- Qual è la rappresentazione in **virgola fissa** (con 8 bit per la parte intera e 8 bit per la parte decimale) del numero 23.625_{10}?

Esercizio 20 – soluzione

- Qual è la rappresentazione in virgola fissa (con 8 bit per la parte intera e 8 bit per la parte decimale) del numero 23.625_{10}?
- Segno: + = 0
- Parte intera:

$$23_{10} = 16 + 4 + 2 + 1 = 0010111$$

- Parte frazionaria:
  - 0.625 * 2 = 1.25 → 1 (prima cifra dopo la virgola)
  - 0.25 * 2 = 0.50 → 0 (seconda cifra dopo la virgola)
  - 0.50 * 2 = 1.00 → 1 (terza cifra dopo la virgola)

- Rappresentazione in virgola fissa (con 8 bit per parte intera e 8 bit per parte frazionaria) di 23.625_{10} = 00010111.10100000

Esercizio 21

- Si supponga di rappresentare i numeri compresi tra 5 e 17,8 con una rappresentazione in virgola fissa di 7 bit. Qual è l’errore di approssimazione (numero – approssimante) che si commetterebbe nel rappresentare il numero 14,888?

Esercizio 21 – soluzione

- Con 7 bit possiamo rappresentare 128 numeri distinti
- Nell'intervallo tra 5 e 17,8 abbiamo un numero infinito di numeri reali ma dividendo l'ampiezza dell'intervallo che vogliamo rappresentare (12,8) per il numero di numeri distinti rappresentabili, otteniamo 0,1 ovvero possiamo rappresentare solo la prima delle cifre decimali
- Il numero approssimante sarà il numero reale con una cifra decimale più vicino a 14,888 e quindi 14,9
- L'errore di approssimazione (numero – approssimante) è quindi 14,888 - 14,9 = - 0,012

Esercizio 22

- Quale è la rappresentazione in **virgola mobile (standard IEEE-754)** del numero decimale 13,25 (con 32 bit)?

Esercizio 22 – soluzione

- Quale è la rappresentazione in virgola mobile (standard IEEE-754) per il numero decimale 13,25 (con 32 bit)?
- La rappresentazione in virgola mobile (standard IEEE-754) con 32 bit prevede:
  - **1 bit per il segno** (in questo caso 0 perché il numero da rappresentare è positivo)
  - **8 bit per l'esponente** (rappresentato in binario eccesso 127 e dopo che il numero è stato normalizzato)
  - **23 bit per la "mantissa"** (in questo caso "mantissa" indica la parte a destra della virgola del numero codificato in binario e normalizzato)

Esercizio 22 – soluzione

Spostamento di 3 posizioni a sinistra della virgola (*shift left*), di un numero binario, corrisponde a dividere il numero per 2^3

- Rappresentazione in virgola mobile (standard IEEE-754) con 32 bit di 13,25:
  - Codifica della parte a sinistra della virgola: 13 → 1101
  - Codifica della parte decimale, a destra della virgola: moltiplico per 2 e scrivo 0 se risultato è minore di 1; e 1 altrimenti. Continuo a ripetere la moltiplicazione della parte decimale del risultato fino a che ottengo 0

    0.25 * 2 = **0**.5

    0.5 * 2 = **1**.0

    0.0 * 2 = 0.0
  - Abbiamo quindi ottenuto: 1101.010
  - Normalizziamo, cioè spostiamo la virgola a sinistra fino a prima dell'ultimo 1 e moltiplichiamo per 2^x dove x è il numero di spostamenti a sinistra della virgola otteniamo: $1.101010 * 2^3$
- La codifica risultante sarà:

  0 10000010 10101000000000000000000
  - N.B. nella mantissa si rappresenta SOLO la parte a destra della virgola; "1." resta sottointeso!
  - N.B la codifica dell'esponente è eccesso 127! Quindi 3_{10} →10000010 (in binario eccesso 127)

Esercizio 23

- Quale valore è rappresentato dalla seguente sequenza di bit interpretati come un numero in virgola mobile?

  1 10000000 11000000000000000000000

Esercizio 23 – soluzione

- Quale valore è rappresentato dalla seguente sequenza di bit interpretati come un numero in virgola mobile?

  1 10000000 11000000000000000000000

  – segno: MSB = 1 $\Rightarrow$ numero negativo

  – esponente: $1000\ 0000_2 = 128_{10} - 127 = 1$

  – "mantissa": ricordiamo che è la parte frazionaria del numero ("1." del numero normalizzato è sottointesa nella rappresentazione!)

  $11000000000000000000000_2 = 2^{-1} + 2^{-2} = 0.5_{10} + 0.25_{10} = 0.75_{10}$

- Risultato: $(-1)*(1+0.75)*2^1 = -\ 1.75*2 = -3.5$

Esercizio 24

- Convertire in binario i seguenti numeri (espressi in base 16) e interpretarli come rappresentazione formato floating point standard IEEE-754. Quale numero razionale rappresentano?

  - 0x41340000
  - 0xC1140000

Esercizio 24 – soluzione

0x41340000

- Numero in floating point= 0100 0001 0011 0100 0000 0000 0000 0000
- Segno = + Esponente (130-127) = 3_{10}
- Numero = $1011.01_2 = 11{,}25_{10}$

0xC1140000

- Numero in floating point = 1100 0001 0001 0100 0000 0000 0000 0000
- Segno = - Esponente (130-127) = 3_{10}
- Numero = $- 1001{,}01_2 = - 9{,}25_{10}$

Esercizio 24

0x41340000

- Numero in floating point= 0100 0001 0011 0100 0000 0000 0000 0000
- Segno = + Esponente (130-127) = 3_{10}
- Numero = $1011,01_2 = 11,25_{10}$

0xC1140000

- Numero in floating point = 1100 0001 0001 0100 0000 0000 0000 0000
- Segno = - Esponente (130-127) = 3_{10}
- Numero = $- 1001,01_2 = - 9,25_{10}$

Esercizio 25

- Rappresentare i seguenti numeri, espressi in base 10, nel formato floating point (standard IEEE-754) e farne la differenza (X - Y)
- X = -5.828125
- Y = 3758.125

Esercizio 25 – soluzione (conversione di X)

X = -5.828125

- Parte intera: $5_{10} = 101_2$
- Parte frazionaria:

0,828125	x 2 =	1,65625	Parte intera	1
0,65625	x 2 =	1,3125	Parte intera	1
0,3125	x 2 =	0,625	Parte intera	0
0,625	x 2 =	1,25	Parte intera	1
0,25x 2 =		0,5	Parte intera	0
0,5	x 2 =	1	Parte intera	1
0	x 2 =	0	Parte intera	0

- Abbiamo ottenuto (da normalizzare): 101.1101010
- Dopo aver normalizzato: $1.01110101 * 2^2$
- X = $1\ \underbrace{10000001}_{=(2+127)_{10}}\ 01110101000000000000000$

Esercizio 25 – soluzione (conversione di Y)

Y = 3758.125

- Parte intera: 3758_{10} = 2048 + 1024 + 512 + 128 + 32 + 8 + 4 + 2 = 111010101110_2
- Parte frazionaria:

0,125 x 2 =	0,25	Parte intera	0
0,25 x 2 =	0,5	Parte intera	0
0,5 x 2 =	1	Parte intera	1
0 x 2 =	0	Parte intera	0

- Numero in binario è 111010101110.0010 che normalizzato equivale a 1.11010101110001$0^*$ 2^{11}
- Y = 0 10001010 11010101110001000000000

= $(11+127)_{10}$ =
128 + 8 + 2

Esercizio 25 – soluzione (differenza)

- X – (Y) = X + (-Y)
- Se

  X = -5.828125 = 1 10000001 01110101000000000000000

  e Y = 3758.125 = 0 10001010 11010101110001000000000

- - Y = 1 10001010 11010101110001000000000 (cambia SOLO bit corrispondente al segno!)

Nell'uso della mantissa per lo svolgimento delle operazioni di somma/sottrazione, dovremmo tener conto che se rappresenta il modulo di un numero negativo, deve prima essere complementata.

Essendo in questo caso, entrambi i numeri negativi, possiamo svolgere l'operazione richiesta negando la somma dei moduli delle mantisse (senza necessità di interpretarle in CA2)

- **Per poter sommare, è SEMPRE necessario allineare le mantisse**: Shift Smaller Right (shift a dx del numero con esponente più piccolo) →

  1.01110101000000000000000 * 2^2 sposto virgola a sx 9 (=11-2) posizioni 0.00000000101110101000000 * 2^11

  1.11010101110001000000000 * 2^11 1.11010101110001000000000 * 2^11

- X – Y = - 5.828125 - 3758.125 = - (5.828125 + 3758.125) e quindi

  0.00000000101110101000000 +

  1.11010101110001000000000 =

  1.11010110011111101000000

Rappresentazione FP IEEE-754, 1 10001010 11010110011111101000000