Architettura degli elaboratori

Lezione 2 - teoria

Prof.ssa Valentina Ciriani
Università degli Studi di Milano
www.di.unimi.it/ciriani

1

Contenuto

1. Rappresentazione degli interi [PH 2.4] [MKM 3.10, 3.11]:
   1. Complemento a 2
   2. Complemento a 1
2. Rappresentazione di numeri reali [PH 3.5]
3. Caratteri e codice ASCII [MKM 1.6]

-MKM = M. Morris Mano, C.R. Kime, T. Martin, Reti logiche, Pearson
-PH = D.A. Patterson, J.L. Hennessy, Struttura e Progetto dei Calcolatori, Zanichelli

2

2

Numeri interi con segno

- Rappresentazione a **complemento a 2 (N=32):**

$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 0_{10}$
$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_2 = 1_{10}$
$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_2 = 2_{10}$
. . . . . .
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_2 = 2.147.483.645_{10}$
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_2 = 2.147.483.646_{10}$
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2 = 2.147.483.647_{10}$
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = -2.147.483.648_{10}$
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_2 = -2.147.483.647_{10}$
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_2 = -2.147.483.646_{10}$
. . . . . .
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_2 = -3_{10}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_2 = -2_{10}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2 = -1_{10}$

Complemento a 2 (C2)

- Intervallo di rappresentazione: $[-2^{N-1}, +2^{N-1}-1]$
- Es: N=8 intervallo $[-2^7, 2^7-1]=[-128, 127]$
- Con N=32 i numeri da 0 a $2.147.483.647_{10}$ ($= 2^{31} - 1$) sono rappresentati come degli interi positivi
- $1000...000_2$ rappresenta il numero negativo di valore assoluto maggiore $-2.147.483.648_{10}$ ($= -2^{31}$)
- $1111...1111_2$ rappresenta -1
- $-2.147.483.648_{10}$ non ha un corrispondente numero positivo
- I numeri negativi iniziano con 1 (bit di segno)

C2: da decimale a binario

- Sia x intero decimale in $[-2^{N-1}, +2^{N-1}-1]$
- la rappresentazione in complemento a 2 di x è:

1. se $x \geq 0$ allora è come la rappresentazione di x senza segno
2. se $x<0$ è la rappresentazione, come intero senza segno, di 2^N+x
(attenzione: x è negativo)

5

5

C2: da decimale a binario esempio

- Es: N=8 intervallo [-128, 127]
  - 3_{10} si rappresenta come $0000\ 0011_2$
  - -3_{10} si rappresenta come $2^8 - 3 = 256 - 3 = 253 = 1111\ 1101_2$

6

6

C2: cambio di segno

- ricopio le cifre da destra a sinistra fino al primo 1 (compreso) e poi invertendo le rimanenti (da 0 a 1 e da 1 a 0)
- Esempio:

$(0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1100)_2 = 12_{10}$
$(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 0100)_2 = -12_{10}$

7

7

C2: da decimale a binario

Algoritmo semplificato:

- Sia x intero decimale in $[-2^{N-1}, +2^{N-1}-1]$
- la rappresentazione in complemento a 2 di x è:

1. se $x \geq 0$ allora è come la rappresentazione di x senza segno
2. se $x<0$ parto dalla rappresentazione di -x (-x è positivo) e poi ricopio le cifre da destra a sinistra fino al primo 1 (compreso) e poi invertendo le rimanenti (da 0 a 1 e da 1 a 0)

8

8

C2: da binario a decimale

Regola (sia per $x \geq 0$ che per $x<0$):
$(x^{N-1}x^{N-2}\dots x^1x^0)_2 =$
$(x^{N-1}(-2^{N-1}) + x^{N-2}2^{N-2} + \dots + x^1 2^1 + x^0 2^0)_{10}$

Esempio:
$(1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001)_2$
$= (-2^{31} + 2^{30} + 2^{29} + 2^{28} + 2^0)_{10} = (-2.147.483.648 + 1.073.741.824 + 536.870.912 + 268.435.456 + 1)_{10}$
$= (-268.435.455)_{10}$

9

9

C2: da binario a decimale

Notiamo che: $-2^N+2^{N-1} = -2^{N-1}$ Es: $-2^4+ 2^3 = -16+8 = -8 = -2^3$
$-2^{31}+2^{30} = -2^{30}$
$-2^{30}+2^{29} = -2^{29}$
ecc…

- Sia x negativo e sia x^i l'ultimo 1 da sinistra, quindi:

$(x^N x^{N-1} x^{N-2}\dots x^1 x^0)_2 =$
$(-2^i + x^{i-1}2^{i-1} + x^{i-2}2^{i-2} + \dots + x^1 2^1 + x^0 2^0)_{10}$

Esempio:
$(1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001)_2$
$= (-2^{31} + 2^{30} + 2^{29} + 2^{28} + 2^0)_{10} = (-2^{30} + 2^{29} + 2^{28} + 2^0)_{10} = (-2^{29} + 2^{28} + 2^0)_{10}$
$= (-2^{28} + 2^0)_{10}$

$=(-2.147.483.648 + 1.073.741.824 + 536.870.912 + 268.435.456 + 1)_{10}$
$=(-1.073.741.824 + 536.870.912 + 268.435.456 + 1)_{10}$
$=(-536.870.912 + 268.435.456 + 1)_{10}$
$=(-268.435.456 + 1)_{10}$
$=(-268.435.455)_{10}$

10

10

C2: da binario a decimale

Quindi:

$(1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001)_2 = (-2^{28} + 2^0)_{10} = (-268.435.456 + 1)_{10} = (-268.435.455)_{10}$

Esempio:

$(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 1001)_2$
$= (-2^5 + 2^3 + 2^0)_{10} = (-32 + 8 + 1)_{10} = (-23)_{10}$

11

11

Esempio N=32

positivi:

$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 0_{10}$
$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_2 = 1_{10}$
$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_2 = 2_{10}$
. . . . . .
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_2 = 2.147.483.645_{10}$
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_2 = 2.147.483.646_{10}$
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2 = 2.147.483.647_{10}$

negativi:

$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = -2.147.483.648_{10}$ $(-2^{31})_{10}$
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_2 = -2.147.483.647_{10}$ $(-2^{31}+1)_{10}$
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_2 = -2.147.483.646_{10}$ $(-2^{31}+2)_{10}$
. . . . . .
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_2 = -3_{10}$ $(-2^2+1)_{10}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_2 = -2_{10}$ $(-2^1)_{10}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2 = -1_{10}$ $(-2^0)_{10}$

12

12

C2: Somma/sottrazione

- La somma e la sottrazione sono identiche a quelle con i numeri senza segno
- Si usa lo stesso metodo per somma e sottrazione x - y = x + (-y)
- Esempio N=8 calcolare 14 -17 = 14 + (-17):
  - 14_{10}= $0000\ 1110_2$
  - 17_{10}= $0001\ 0001_2$ -17_{10}= $1110\ 1111_2$

```
   1  11
0000 1110 +
1110 1111=
1111 1101
```

$= (-2^2 + 2^0)_{10} = -3_{10}$

13

Overflow

- L'addizione genera overflow quando la somma non è nell'intervallo di rappresentazione e quindi non è rappresentabile

1. la somma fra un numero positivo e un numero negativo **non** genera overflow
   - se c'è un riporto finale, viene **ignorato**
2. la somma tra due addendi positivi (o due negativi) **può dare** overflow
   - se il **segno** del risultato è diverso da quello degli operandi si ha overflow
   - se c'è un riporto finale, viene **ignorato**

14

C2: Esempi overflow N=4

```
 111
  1110 +
  0111 =
 -------
 10101
```

Corretto sempre (mai overflow)

$(-2+7)_{10} = 5_{10}$

```
  0010 +
  0001 =
 -------
  0011
```

Corretto perché segni uguali!

$(2+1)_{10} = 3_{10}$

```
 11
  1110 +
  1100 =
 -------
 11010
```

Corretto perché segni uguali!

$(-2-4)_{10} = -6_{10}$

15

15

C2: Esempi overflow N=4

```
  1
  0111 +
  0100 =
 -------
  1011
```

Overflow perché segni diversi!

$(7+4)_{10} = 11_{10}$ (overflow perché>7)

```
 1
  1000 +
  1000 =
 -------
 10000
```

Overflow perché segni diversi!

$(-8-8)_{10} = -16_{10}$(overflow perché<8)

16

16

Complemento a 1

- È una terza possibile rappresentazione dei numeri dotati di segno
- Il numero negativo di un numero in complemento a 1 si ottiene invertendo ciascun bit, da 0 a 1 e da 1 a 0
- Esempio

$(0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0100)_2 = (4)_{10}$

$(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1011)_2 = (-4)_{10}$

- Contro:
  - due rappresentazioni per lo 0: 000…00 e 111…11
  - i sommatori in complemento a 1 richiedono un passo in più per eseguire la sottrazione
- Non è utilizzato

17

17

Rappresentazione dei reali

18

18

Rappresentazione di numeri reali

- Come si rappresenta la parte frazionaria?
  - Rappresentazione in **virgola fissa** (non si possono rappresentare numeri molto grandi o molto piccoli). Si fissa un numero fisso di bit per la parte intera e un numero fisso per la parte frazionaria. Non utilizzato.
  - Rappresentazione in **virgola mobile** che corrisponde alla rappresentazione scientifica

19

19

Virgola mobile

- In base 10 la notazione scientifica rappresenta un numero con le potenze di 10
- Ad esempio $+278{,}34 = +2{,}7834 \times 10^2$
- Nel sistema binario abbiamo:
  $-10110{,}101 = -1{,}0110101 \times 2^4$
  $+0{,}001101 = +1{,}101 \times 2^{-3}$

20

20

Virgola mobile

- numero binario in forma **normalizzata**:

$$+/-\ (1,\ldots)_2 \times 2^y$$

(per semplicità rappresentiamo y in decimale)

- Si può rappresentare nella forma:

$$(-1)^S \times (1,M)_2 \times 2^y$$

M = mantissa (parte dopo la virgola)
y = esponente
S = segno (S=1 numero negativo, S=0 positivo)

- è una rappresentazione del tipo **modulo e segno**:
  il complemento a 2 si usa per gli interi ma non per i reali

21

21

Precisione

- Singola precisione (float) 32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	esponente								mantissa																						
1 bit	8 bit								23 bit																						

- Doppia precisione (double) 64 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	esponente											mantissa																			
1 bit	11 bit											20 bit																			
mantissa (continuazione)																															
32 bit																															

22

22

Esempio

- Rappresentare in virgola mobile -37,75
- Conversione in binario di 37,75:
  100101,11
- Forma normalizzata: $-1{,}0010111 \times 2^{5}$
- M = 0010111
- y = 101 (5)
- S=1

23

23

Lo standard IEEE 754

- Standard utilizzato dagli anni '80 per la rappresentazione dei numeri in virgola mobile
- Precisione singola (32 bit):
  - Si utilizza la codifica **polarizzata** a **127** (in eccesso di 127) dell'esponente (E è sempre un positivo)
  - $(-1)^{S} \times (1{,}M) \times 2^{E-127}$
  - Quindi per il numero normalizzato +/- $(1,\dots)_2 \times 2^{y}$
    y = E-127 ovvero **E = 127+y**
  - $-126 \leq y \leq 127$
  - **$1 \leq E \leq 254$**

24

24

Lo standard IEEE 754

- Precisione doppia (64 bit):
  - Si utilizza la codifica **polarizzata** a **1023** dell'esponente (E è sempre un positivo)
  - $(-1)^S \times (1,M) \times 2^{E-1023}$
  - Quindi per il numero normalizzato $+/-(1,\ldots)_2 \times 2^y$
    $y = E-1023$ ovvero **$E = 1023+y$**
  - $-1022 \leq y \leq 1023$
  - **$1 \leq E \leq 2046$**

25

25

Lo standard IEEE 754

- Come faccio a rappresentare lo **0** con $(-1)^S \times (1,M)_2 \times 2^{E-127}$?
- Per lo 0 metto a 0 sia mantissa che esponente (convenzione)

Singola precisione		Doppia precisione		Situazioni rappresentate
Esponente	Mantissa	Esponente	Mantissa	
0	0	0	0	0
1-254	Qualsiasi numero	1-2046	Qualsiasi numero	± numero in virgola mobile
255	0	2047	0	± infinito
255	Diverso da 0	2047	Diverso da 0	NaN (Not a Number)

26

26

Esempio da decimale a binario

- $-0{,}75_{10}$ in singola precisione standard IEEE 754 (polarizzazione 127)
- $0{,}75_{10} = 0{,}11_2$
- $-1{,}1 \times 2^{-1}$ (normalizzazione) y= -1 M=1 e S=1
- E= $(127-1)_{10}$ quindi E = $126_{10} = 01111110_2$
- $(-1)^1 \times (1{,}10000000000000000000000) \times 2^{126-127}$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1 bit 8 bit 23 bit

27

27

Esempio da decimale a binario

- $-0{,}75_{10}$ in doppia precisione standard IEEE 754 (polarizzazione 1023)
- $0{,}75_{10} = 0{,}11_2$
- $-1{,}1 \times 2^{-1}$
- E = 1023 -1 quindi E = $1022_{10} = 01111111110_2$
- $(-1)^1 \times (1{,}100000...00000) \times 2^{1022-1023}$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1 bit 11 bit 20 bit

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32 bit

28

28

Esempio da decimale a binario

- $12{,}25_{10}$ in doppia precisione standard IEEE 754 (polarizzazione 1023)
- $12{,}25_{10} = 1100{,}01_2$
- $1{,}10001 \times 2^3$ (normalizzazione)
- $E=1023 + 3$ quindi $E = 1026_{10} = 10000000010_2$
- $(-1)^0 \times (1{,}1000100...00000) \times 2^{1026-1023}$

$$\underbrace{0}_{1\text{bit}}\ \underbrace{10000000010}_{11\text{bit}}\ \underbrace{1000100...00000}_{52\text{bit}}$$

29

29

Esempio: da binario a decimale

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	.

Rappresentare in decimale
(singola precisione: polarizzazione 127)

- $S = 1$
  $E = 10000001_2 = 129_{10}$
  e quindi $y = (129-127)_{10} = 2_{10}$
  $M = 01$ quindi ho $1{,}01_2$ ovvero $1{,}25_{10}$
- $(-1)^1 \times (1{,}25) \times 2^{129-127} = -1{,}25 \times 2^2 = -5_{10}$

30

30

Codice ASCII

- American Standard Code for Information Interchange
- 7 bit per la codifica di 128 caratteri

caratteri di controllo

	$B_7B_6B_5$							
$B_4B_3B_2B_1$	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

31

31

Esercizi da fare a casa 1

1. Si consideri N=8. Rappresentare i numeri 26_{10} e -26_{10} in binario:
   1. modulo e segno
   2. complemento a 2
   3. complemento a 1
2. Si consideri N=8. Convertire in decimali, i seguenti numeri in C2
   1. 1110 0101
   2. 0011 0110
3. Eseguire la somma e la sottrazione in C2 con N=8 tra le seguenti coppie di numeri decimali (ci sono casi di overflow?):
   1. -126_{10} e 25_{10}
   2. -62_{10} e 25_{10}
   3. 115_{10} e -24_{10}

32

32

Esercizi da fare a casa 2

4. Rappresentare in virgola mobile $-43{,}25_{10}$ e $43{,}25_{10}$
5. Rappresentare i numeri $-12{,}75_{10}$, 0_{10} e 1_{10}
    1. in singola precisione standard IEEE 754
    2. in doppia precisione standard IEEE 754
6. Rappresentare in decimale il seguente numero in singola precisione standard IEEE 754

    11010110110101010000000000000000

33

33

Architettura degli elaboratori

Lezione 2 - esercizi

Prof.ssa Valentina Ciriani
Università degli Studi di Milano
www.di.unimi.it/ciriani

34

Esercizio 1.1

Convertire da base r a base 10:

r=2 $11010{,}101_2$

Soluzione:

$11010{,}101_2 =$

$(2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-3})_{10} =$

$(16+8+2+0{,}5+0{,}125)_{10}=$

$26{,}625_{10}$

35

35

Esercizio 1.2

Convertire da base r a base 10:

r=8 $4156{,}27_8$

Soluzione:

$4156{,}27_8 =$

$(4\times8^3+8^2+5\times8^1+ 6 + 2\times8^{-1}+7\times8^{-2})_{10}=$

$(2048+64+40+6+0{,}25+0{,}109375)_{10}=$

$2158{,}359375_{10}$

36

36

Esercizio 1.3

Convertire da base r a base 10:

r=16 $A8F,B_{16}$

Soluzione:

$A8F,B_{16} =$

$(10\times16^2 + 8\times16 + 15 + 11\times16^{-1})_{10}=$

$(2560 + 128 + 15 + 0,6875)_{10}=$

$2703,6875_{10}$

37

37

Esercizio 2.1

Convertire da base 10 a base 2 il numero 4526,76 con precisione p=5

Soluzione:

4526/2 = 2263	4526%2 = 0	$0,76 \times 2 = 1,52$	1	
2263/2 = 1131	2263%2 = 1	$0,52 \times 2 = 1,04$	1	
1131/2 = 565	1131%2 = 1	$0,04 \times 2 = 0,08$	0	
565/2 = 282	565%2 = 1	$0,08 \times 2 = 0,16$	0	
282/2 = 141	282%2 = 0	$0,16 \times 2 = 0,32$	0	p=5
141/2 = 70	141%2 = 1			
70/2 = 35	70%2 = 0			
35/2 = 17	35%2 = 1			
17/2 = 8	17%2 = 1			
8/2= 4	8%2 = 0			
4/2= 2	4%2 = 0			
2/2= 1	2%2 = 0			
1/2= 0	1%2 = 1			

$(1000110101110,11000)_2$

38

38

Esercizio 2.2

Convertire da base 10 a base 2 il numero 4526,75 con precisione p=5

Soluzione:

4526/2 = 2263	4526%2 = 0	$0{,}75 \times 2 = 1{,}50$	1
2263/2 = 1131	2263%2 = 1	$0{,}50 \times 2 = 1{,}00$	1
1131/2 = 565	1131%2 = 1	0,00 mi fermo	
565/2 = 282	565%2 = 1		
282/2 = 141	282%2 = 0		
141/2 = 70	141%2 = 1		
70/2 = 35	70%2 = 0		
35/2 = 17	35%2 = 1		
17/2 = 8	17%2 = 1		
8/2= 4	8%2 = 0		
4/2= 2	4%2 = 0		
2/2= 1	2%2 = 0		
1/2= 0	1%2 = 1		

$(1000110101110{,}11000)_2$ **stessa rappresentazione di 4526,76!**

39

39

Esercizio 2.3

Convertire da base 10 a base 8 il numero 23,2 con precisione p=3

23/8 = 2	23%8 = 7	$0{,}2 \times 8 = 1{,}6$	1
2/8 = 0	2%8 = 2	$0{,}6 \times 8 = 4{,}8$	4
		$0{,}8 \times 8 = 6{,}4$	6

$(27{,}146)_8$

40

40

Esercizio 2.4

Convertire da base 10 a base 16 il numero 270,12 con precisione p=3

270/16 = 16	270%16 = 14 (E)	$0,12 \times 16 = 1,92$	1
16/16 = 1	16%16 = 0	$0,92 \times 16 = 14,72$	E
1/16 = 0	1%16 = 1	$0,72 \times 16 = 11,52$	B

$10E,1EB_{16}$

41

41

Esercizio 3.1

Convertire da base 2 a base 8 e a base 16 e viceversa (da 8 a 2 e da 16 a 2) il binario 1010111001_2

base 8: $001\ 010\ 111\ 001_2$
1 2 7 1 1271_8

base 16: $0010\ 1011\ 1001_2$
2 B 9 $2B9_{16}$

42

42

Esercizio 3.2

Convertire da base 2 a base 8 e a base 16 e viceversa (da 8 a 2 e da 16 a 2) il binario 11101110000_2

base 8: $011\ 101\ 110\ 000_2$

$3 \quad 5 \quad 6 \quad 0_8 \qquad 3560_8$

base 16: $0111\ 0111\ 0000_2$

$7 \quad 7 \quad 0_{16} \qquad 770_{16}$

43

43

Esercizio 4.1

Eseguire la somma, la sottrazione e la moltiplicazione tra le seguenti coppie di numeri binari senza segno: 11010_2 e 101_2

```
11010+        101          11010×
  101=      11010−           101=
-----         101=         -----
11111       -----          11010
            10101         00000
                         11010
                        --------
                        10000010
```

44

44

Esercizio 4.2

Eseguire la somma, la sottrazione e la moltiplicazione tra le seguenti coppie di numeri binari senza segno: 11011_2 e 10101_2

```
 1111          1
  11011+       11011-        11011×
  10101=       10101=        10101=
 110000        00110         11011
                            00000
                           11011
                          00000
                         11011
                         1000110111
```


Esercizio 4.3

1. Eseguire la somma, la sottrazione e la moltiplicazione tra le seguenti coppie di numeri binari senza segno: 101001_2 e 11111_2

```
 11111         1111
 101001+       101001-        101001×
  11111=        11111=         11111=
1001000        001010         101001
                             101001
                            101001
                           101001
                          101001
                          10011110111
```

Esercizio 5

Quale è l'intervallo di rappresentazione di binari senza segno con 10 cifre?

$[0, 2^{10}-1] = [0,1023]$

47

47

Esercizio 6

Si consideri N=8. Rappresentare i numeri 85_{10} e -85_{10} in binario:

1. modulo e segno
2. complemento a 2

85/2 = 42 85%2=1
42/2 = 21 42%2=0
21/2 = 10 21%2=1
10/2 = 5 10%2=0
5/2 = 2 5%2=1
2/2 = 1 2%2=0
1/2 = 0 1%2=1

$85_{10}=1010101_2$

1. 85: 01010101_2 -85: 11010101_2
2. 85: 01010101_2 -85: 10101011_2

48

48

Esercizio 7

Quanti bit mi servono per rappresentare le 21 lettere dell'alfabeto italiano?

- Per rappresentare 21 valori mi servono $\lceil \log_2 21 \rceil = \lceil 4{,}39 \rceil = 5$ bit
- Ad esempio, per rappresentare i 21 caratteri dell'alfabeto italiano possiamo usare la codifica:

A = 00000	H= 00111	Q=01110
B = 00001	I = 01000	R=01111
C = 00010	L= 01001	S=10000
D = 00011	M= 01010	T=10001
E = 00100	N= 01011	U=10010
F = 00101	O= 01100	V=10011
G = 00110	P= 01101	Z=10100

49

49