AP00028

Fondamenti di Informatica: Teoria

Maurizio Rebaudengo, Alfredo Benso

http://www.appunti.net/
Il sito italiano di appunti universitari

Informazioni sul Copyright

Questo documento è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi al documento (comprensivo, ma non limitatamente, di ogni immagine, fotografia e testo) sono di proprietà dell'autore o degli autori indicati in copertina.
Il documento può essere utilizzato liberamente dal legittimo acquirente (ivi compresi acquisti promozionali o scaricamenti gratuiti dalla rete Internet) per gli usi consentiti dalla legge. Ogni legittimo acquirente è autorizzato a stampare al più una copia del documento.
Ogni altro utilizzo o riproduzione (ivi incluse, ma non limitatamente, le riproduzioni su supporti magnetici, su reti di calcolatori e stampe) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte degli autori.
L'informazione contenuta in questo documento è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, ecc. In ogni caso essa è soggetta a cambiamenti ed aggiornamenti senza preavviso. Gli autori non assumono alcuna responsabilità per il contenuto di questo documento (ivi incluse, ma non limitatamente, la correttezza, completezza, applicabilità, aggiornamento dell'informazione).
In ogni caso non può essere dichiarata conformità all'informazione contenuta in questo documento.
In ogni caso questa nota di copyright e il suo richiamo in calce ad ogni pagina non devono mai essere rimossi e devono essere riportati anche in utilizzi parziali.

Sistemi di Numerazione

Sistemi di Numerazione

I sistemi di numerazione sono abitualmente *posizionali*.

Gli elementi costitutivi di un sistema posizionale sono:

- *base* o *radice*
- *insieme di cifre*.

2

Sistema posizionale decimale

Il sistema di numerazione usato normalmente è posizionale e decimale.

- La base è *10*
- L'insieme di cifre è costituito da *{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}*.
- La cifra più alta utilizzabile per rappresentare un numero in base N è (N-1).

3

Altri sistemi posizionali

Sistema binario:

- r = 2
- cifre = {*0, 1*}

Sistema ottale:

- r = 8
- cifre = {*0, 1, 2, 3, 4, 5, 6, 7*}

Sistema esadecimale:

- r = 16
- cifre = {*0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F*}

4

Sistema posizionale

Un numero è rappresentato da una sequenza di cifre.

$N = d_n d_{n-1} .... d_2 d_1 d_0$.

Il valore assunto dal numero N è calcolato come:

$N = d_n*10^n + d_{n-1}*10^{n-1} + ... + d_2*10^2 + d_1*10^1 + d_0*10^0$.

Partendo dalla cifra più significativa, si moltiplica la cifra per il valore della base elevata alla potenza corrispondente alla posizione.

5

Base 10 Esempi

$208 = 2*10^2 + 0*10^1 + 8*10^0$ = 2 centinaia + 0 decine + 8 unità

$1232 = 1*10^3 + 2*10^2 + 3*10^1 + 2*10^0$ =
1 migliaio + 2 centinaia + 3 decine + 2 unità.

6

© 2000 M. Rebaudengo, A. Benso

Sistema non posizionale

Il sistema romano è basato su regole diverse da un sistema posizionale.

Il valore associato a ciascun simbolo non dipende dalla sua posizione: il sistema di numerazione romano non è posizionale.

Esempio:

I due numeri XII e XIX hanno la seconda cifra che pur avendo la stessa posizione assume nei due numeri due pesi diversi.

XII = 10 *+ 1* + 1

XIX = 10 + 10 *- 1*

7

Conversione da base N a base 10

Regola:

Dato un numero $M = (d_n d_{n-1} .... d_2 d_1 d_0)_N$ rappresentato in un sistema posizionale in base *N*, per convertire il numero in base *10* occorre effettuare la somma dei singoli termini pesati opportunamente in base alla loro posizione:

$$d_n * N^n + d_{n-1} * N^{n-1} + ... + d_2 * N^2 + d_1 * N^1 + d_0 * N^0$$

8

Conversione da base N a base 10 Esempi

$(302)_7 = (3*7^2 + 0*7^1 + 2*7^0)_{10} = (3*49 + 0*7 + 2*1)_{10} =$

$= (147 + 0 + 2)_{10} = (149)_{10}$

$(1001101)_2 = 1*2^6+0*2^5+0*2^4+1*2^3+1*2^2+0*2^1+1*2^0 = (77)_{10}$

$(477)_8 = 4*8^2+7*8^1+7*8^0 = (319)_{10}$

$(40F)_{16} = 4*16^2+7*16^1+7*16^0 = (1039)_{10}$

9

Conversione da base N a base 10 Esercizi

Convertire in base 10 i seguenti numeri espressi nelle basi indicate:

- $(5778)_9$
- $(3074)_5$
- $(3B8)_{16}$
- $(3B8)_{11}$
- $(303)_4$

10

Conversione da base N a base 10 Soluzioni

Convertire in base 10 i seguenti numeri espressi nelle basi indicate:

- $(5778)_9$ = $(4283)_{10}$
- $(3074)_5$ = Impossibile !
- $(3B8)_{16}$ = $(952)_{10}$
- $(3B8)_{11}$ = Impossibile !
- $(303)_4$ = $(51)_{10}$

11

Conversione da base N a base 10 Esercizi proposti

Convertire in base 10 i seguenti numeri espressi nelle basi indicate:

- $(10010)_8$ [R. 4104]
- $(10010)_2$ [R. 18]
- $(5669)_{11}$ [R. 7456]
- $(889)_{12}$ [R. 1257]
- $(1110)_3$ [R. 1065]

12

© 2000 M. Rebaudengo, A. Benso

Conversione da base 10 a base N

$(A)_{10} = d_n{*}N^n + d_{n-1}{*}N^{n-1} + ... + d_2{*}N^2 + d_1{*}N^1 + d_0{*}N^0 =$

$= N{*}(d_n{*}N^{n-1} + d_{n-1}{*}N^{n-2} + ... + d_2{*}N^1 + d_1) + d_0 =$

$= N{*}Q_0 + d_0$

d_0 è ottenibile come resto tra il numero e la base.

Gli altri termini si ottengono iterando l'algoritmo:

$Q_0 = d_n{*}N^{n-1} + d_{n-1}{*}N^{n-2} + ... + d_3{*}N^2 + d_2{*}N^1 + d_1 =$

$= N{*}(d_n{*}N^{n-2} + d_{n-1}{*}N^{n-3} + ... + d_3{*}N^1 + d_2) + d_1 =$

$= N{*}Q_1 + d_1$

13

Conversione da base 10 a base N

Regola:

Si effettuano divisioni successive del numero dato per la base N. I resti delle singole divisioni, presi alla rovescia e scritti in base N, rappresentano le cifre del numero nella base N.

14

Conversione da base 10 a base N Esempi

Convertire il numero decimale 109 in base 2, 5 e 16:

base 2

109 : 2 = 54 + 1
54 : 2 = 27 + 0
27 : 2 = 13 + 1
13 : 2 = 6 + 1
6 : 2 = 3 + 0
3 : 2 = 1 + 1
1 : 2 = 0 + 1

109	54	27	13	6	3	1	0	← quozienti
	1	0	1	1	0	1	1	← resti

$(109)_{10} = (1101101)_2$

15

Conversione da base 10 a base N Esempi

base 5

109 : 5 = 21 + 4
21 : 5 = 4 + 1
4 : 5 = 0 + 4

109	21	4	0
	4	1	4

$(109)_{10} = (414)_5$

base 16

109 : 16 = 6 + 13
6 : 16 = 0 + 6

109	6	0
	13	6

$(109)_{10} = (6D)_{16}$

16

Conversione da base 10 a base N Esercizi

Convertire i seguenti numeri decimali nelle base specificate:

- **345 in base 2**
- **345 in base 8**
- **345 in base 16**
- **6666 in base 16**

17

Conversione da base 10 a base N Soluzioni (1)

Convertire i seguenti numeri decimali nelle base specificate:

- **345 in base 2**

345	172	86	43	21	10	5	2	1	0
	1	0	0	1	1	0	1	0	1

$(345)_{10} = (101011001)_2$

- **345 in base 8**

345	43	5	0
	1	3	5

$(345)_{10} = (531)_8$

18

© 2000 M. Rebaudengo, A. Benso

Conversione da base 10 a base N Soluzioni (2)

Convertire i seguenti numeri decimali nelle base specificate:

- **345 in base 16**

345	21	1	0
	9	5	1

$(345)_{10} = (159)_{16}$

- **6666 in base 16**

6666	416	26	1	0
	10	0	10	1

$(6666)_{10} = (1A0A)_{16}$

Conversione da base 10 a base N Esercizi proposti

Convertire i seguenti numeri decimali nelle base specificate:

- **9787 in base 16** [R. 263B]
- **417 in base 7** [R. 1134]
- **615 in base 9** [R. 753]
- **426 in base 2** [R. 110101010]
- **4596 in base 4** [R. 1013310]
- **111 in base 2** [R. 1101111]
- **1101 in base 8** [R. 2115]

Tabella delle potenze di 2

Conversione da base 10 a base 2

Un metodo alternativo di conversione da base 10 a base 2 fa uso della tabella delle potenze di 2.

***Regola*:**

Ad ogni passo si prende la più alta potenza di 2 inferiore al numero decimale; si assegna la cifra 1 alla posizione corrispondente all'esponente della massima potenza di 2.

Si sottrae il numero di partenza e si ripete il procedimento finchè il numero è diverso da 0.

Conversione da base 10 a base 2 Esempio

Si converta in base 2 il numero decimale 2453.

Soluzione:

2^{11} = 2048 < 2453 => cifra posizione 11 = 1 ; 2453 - 2048 = 405

2^{8} = 256 < 405 => cifra posizione 8 = 1 ; 405 - 256 = 149

2^{7} = 128 < 149 => cifra posizone 7 = 1 ; 149 - 128 = 21

2^{4} = 16 < 21 => cifra posizione 4 = 1 ; 21-16 = 5

2^{2} = 4 < 5 => cifra posizione 2 =1 ; 5 - 4 = 1

2^{0} = 1 = 1 => cifra posizione 0 = 1 ; 1 - 1 = 0

In tutte le altre posizioni ho uno 0.

11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	1	0	1	0	1

Conversione da base N a base M

In generale conviene fare la conversione da base N a base 10, seguita dalla conversione da base 10 a base M.

Esempio:

Convertire da base 5 a base 2 il numero $(104)_5$.

Il numero $(104)_5$ = 1*25 + 0*5 + 4 = $(29)_{10}$

Per convertire in binario si esegue l'algoritmo delle divisioni:

29	14	7	3	1	0
	1	0	1	1	1

Il numero $(104)_5$ equivale a $(11101)_2$.

© 2000 M. Rebaudengo, A. Benso

Conversione da base 2 a base 8 o base 16

Un caso particolare di conversione da base N a base M si ha quando si deve passare da base 2 alle basi 8 o 16 (o viceversa).

Il calcolo è semplificato perchè:

- ogni cifra ottale (0, 1, ... 7) è esprimibile nella corrispondente codifica binaria (000, 001, ... 111) su 3 cifre binarie
- ogni cifra esadecimale (0, 1, ... F) è esprimibile nella corrispondente codifica binaria (0000, 0001, ... 1111) su 4 cifre binarie.

25

Conversione da base 2 a base 8 e viceversa

Regola:

Per convertire un numero da base 2 a base 8, si parte dalla cifra meno significativa e si considerano le cifre binarie a gruppi di 3.

Regola:

Per convertire un numero da base 8 a base 2, si converte ogni cifra del numero in base 8 nella corrispondente codifica su 3 cifre in base 2.

26

Conversione da base 2 a base 8 e viceversa: esempio

- Convertire in ottale il numero binario $(1001010100010110)_2$:

$$(1\ \ 001\ \ 010\ \ 100\ \ 010\ \ 110)_2$$
$$(1\ \ 1\ \ 2\ \ 4\ \ 2\ \ 6)_8$$

- Convertire in binario il numero ottale $(2435)_8$:

$$(2\ \ 4\ \ 3\ \ 5)_8$$
$$(010\ \ 100\ \ 011\ \ 101)_2$$

27

Conversione da base 2 a base 16 e viceversa

Regola:

Per convertire un numero da base 2 a base 16, si parte dalla cifra meno significativa e si considerano le cifre binarie a gruppi di 4.

Regola:

Per convertire un numero da base 16 a base 2, si converte ogni cifra del numero in base 8 nella corrispondente codifica su 4 cifre in base 2.

28

Conversione da base 2 a base 16 e viceversa: esempio

- Convertire in esadecimale il numero binario $(1001010100010110)_2$:

$$(1001\ \ 0101\ \ 0001\ \ 0110)_2$$
$$(9\ \ 5\ \ 1\ \ 6)_{16}$$

- Convertire in binario il numero esadecimale $(A3D)_{16}$:

$$(A\ \ 3\ \ D)_8$$
$$(1010\ \ 0011\ \ 1101)_2$$

29

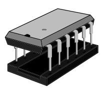
Conversione da base M a base N Esercizi proposti

Convertire i seguenti numeri nelle basi indicate:

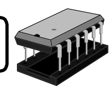
- $(10010101001010)_2$ in base 8 [R. 22512]
- $(11110101101000)_2$ in base 16 [R. 3D68]
- $(13277)_8$ in base 2 [R. 1011010111111]
- $(80E9)_{16}$ in base 2 [R. 1000000011101001]
- $(213)_5$ in base 2 [R. 111010]
- $(354)_7$ in base 8 [R. 272]
- $(AC29B)_{16}$ in base 8 [R. 2541233]
- $(34772)_8$ in base 16 [R. 39FA]

30

© 2000 M. Rebaudengo, A. Benso



Algebra di Boole



31

Variabili booleane

Sono variabili che possono assumere solo il valore 0 oppure 1.

32

Operatori booleani

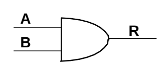
Sono operatori applicabili a variabili booleane.

E' possibile associare ad ogni operatore una corrispondente *porta logica*.

33

Operatore AND

A	B	R
0	0	0
0	1	0
1	0	0
1	1	1



R = AB

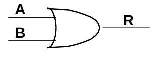
↓

R = 1 se TUTTE le variabili sono = 1

34

Operatore OR

A	B	R
0	0	0
0	1	1
1	0	1
1	1	1



R = A + B

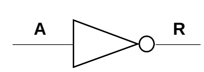
↓

R = 1 se ALMENO UNA variabile = 1

35

Operatore NOT

A	R
0	1
1	0



$R = \overline{A}$

↓

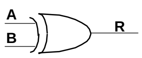
R = 1 se A = 0

36

© 2000 M. Rebaudengo, A. Benso

Operatore EXOR

A	B	R
0	0	0
0	1	1
1	0	1
1	1	0

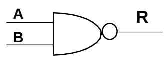


R = A ⊕ B
↓
R = 1 se A ≠ B

37

Operatore NAND

A	B	R
0	0	1
0	1	1
1	0	1
1	1	0

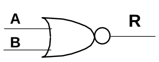


R = 1 se ALMENO UNA variabile = 0

38

Operatore NOR

A	B	R
0	0	1
0	1	0
1	0	0
1	1	0

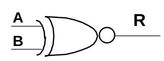


R = 1 se TUTTE le variabili = 0

39

Operatore EXNOR

A	B	R
0	0	1
0	1	0
1	0	0
1	1	1



R = 1 se A = B

40

Funzioni booleane

Una funzione booleana è una funzione di variabili booleane che può assumere soltanto i valori 0 od 1.

Es.:

$$F = a + \overline{bc} + c\overline{e}$$

41

Teoremi dell'algebra di Boole

Associatività della somma	**(A + B) + C = A + (B + C)**
Associatività del prodotto	**(A * B) * C = A * (B * C)**
Doppia Negazione	$\overline{\overline{A}} = A$
Distributività del prodotto	**A+(B*C) = (A + B)* (A+C)**
Distributività della somma	**A*(B+C) = (A*B)+(A*C)**
Assorbimento	**A*A = A e A+A = A**

42

© 2000 M. Rebaudengo, A. Benso

Teoremi dell’algebra di Boole
(segue)

$X * 0 = 0$	$X + 1 = 1$
$X * 1 = X$	$X + 0 = X$
$X * \overline{X} = 0$	$\overline{X} + X = 1$
$X + X*Y = X$	$X * (X + Y) = X$
$X + \overline{X}*Y = X + Y$	$\overline{X} * (X + Y) = X * Y$
$X * Y + X * \overline{Y} = X$	$(X + Y) * (X + \overline{Y}) = X$

Leggi di De Morgan

$\overline{A+B} = \overline{A} * \overline{B}$

$\overline{A*B} = \overline{A} + \overline{B}$

Sono molto utili nel caso in cui si debba negare una funzione booleana.

Attenzione :

$\overline{ab}$ è DIVERSO da $\overline{a}\,\overline{b}$

a	b	$\overline{ab}$
0	0	1
0	1	1
1	0	1
1	1	0

a	b	$\overline{a}\,\overline{b}$
0	0	1
0	1	0
1	0	0
1	1	0

Semplificazione di espressioni booleane

Semplificare la seguente espressione e poi ricavare la funzione $\overline{f}$

f (x, y, z) $= \overline{x}y\overline{z} + \overline{x}yz + yz =$
$= \overline{x}y(\overline{z} + z) + yz =$
$= \overline{x}y + yz =$
$= y(\overline{x} + z)$

$\overline{f(x, y, z)} = \overline{y(x + z)} = \overline{y} + \overline{(x + z)} = \overline{y} + \overline{x}\,\overline{z}$

Esercizi proposti

Semplificare le seguenti espressioni:

1 $\overline{ac}d + bcd + acd + ab + bd$

2 $(\overline{a} + c)a\overline{b}\,\overline{c}$

Esercizi proposti (soluzione)

F (a, b, c) $= \overline{ac}d + bcd + acd + ab + bd =$
$= d(\overline{ac} + ac) + bd(c + 1) + ab =$
$= d + bd + ab =$
$= d(b + 1) + ab =$
$= d + ab$

F (a, b, c) $= (\overline{a} + c)a\overline{b}\,\overline{c} =$
$= \overline{a}a\overline{b}\,\overline{c} + ca\overline{b}\,\overline{c} = 0 + 0 = 0$

© 2000 M. Rebaudengo, A. Benso

Rappresentazione di funzioni booleane

Le funzioni booleane possono essere definite in due modi:

- **tramite *un'espressione* di variabili booleane;**
- **tramite *tabelle di verità*, nelle quali si indicano il valore assunto dalla funzione per ogni possibile combinazione delle sue variabili.**

49

Tabelle della verità Esempio

$F(x_1,x_2) = x_1 + x_2$ è una funzione booleana rappresentata da un'espressione con due variabili booleane.

Può essere definita dalla seguente tabella della verità:

x_1	x_2	F
0	0	0
0	1	1
1	0	1
1	1	1

50

Tabelle della verità Esercizio

Ricavare la tabella della verità della seguente espressione booleana:

F = ab + ac + bc

Si deve scrivere il valore della funzione per ogni possibile combinazione delle sue variabili.

51

Tabelle della verità Esercizio (soluzione)

F = ab + ac + bc

a	b	c	ab	ac	bc	F
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

52

Tabelle della verità Esercizi

Determinare le tabelle delle verità delle seguenti funzioni:

- **$f1 = a (a + b)$**
- **$f2 = (ab + \bar{b})\bar{c}$**
- **$f3 = a\bar{b}c + bc + ac$**

53

Tabelle della verità Esercizi (soluzione)

- **f1 = a (a + b) = a + ab = a (1 + b) = a**

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

54

© 2000 M. Rebaudengo, A. Benso

Tabelle della verità Esercizi (soluzione)

- $f2 = (ab + \bar{b})\bar{c} = ab\bar{c} + \bar{b}\bar{c}$

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

55

Tabelle della verità Esercizi (soluzione)

- $f3 = a\bar{b}c + bc + ac = ac + bc$

a	b	c	ac	bc	F
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	0	1	1

56

Tabella della verità

Data una tabella della verità è possibile ricavare l'espressione booleana che specifica la relativa funzione booleana sotto forma di *somma di prodotti*.

57

Tabella della verità

x_1	x_2	F
0	0	0
0	1	1
1	0	1
1	1	0

$\overline{x_1}\,x_2 + x_1\,\overline{x_2}$

$F(x_1,x_2) = \overline{x_1}\,x_2 + x_1\,\overline{x_2}$

58

Tabella della verità Esercizio

Ricavare la funzione espressa dalla seguente tabella della verità:

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

59

Tabella della verità Esercizio (soluzione)

a	b	c	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	→ $\bar{a}\,b\,c$
1	0	0	0	
1	0	1	1	→ $a\,\bar{b}\,c$
1	1	0	1	→ $a\,b\,\bar{c}$
1	1	1	1	→ $a\,b\,c$

$F = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc =$
$ab + ac + bc$

60

© 2000 M. Rebaudengo, A. Benso

Esercizio 1

Data la seguente proposizione:

vado in vacanza ad agosto se ho già dato tutti gli esami oppure se ho degli esami arretrati e gli appelli di tali esami non sono ai primi di settembre.

identificare le variabili booleane e scrivere la funzione che identifica se uno studente può andare in vacanza ad agosto.

61

Esercizio 1 (soluzione)

1) Individuo le variabili booleane:

vado in vacanza ad agosto se ho già dato tutti gli esami oppure se ho degli esami arretrati e non ci sono appelli di tali esami ai primi di settembre.

Posso individuare due variabili:

- **variabile a: situazione esami**
  - **0: ho esami arretrati**
  - **1: ho dato tutti gli esami.**
- **variabile b: situazione appelli**
  - **0: gli appelli non sono ai primi di settembre**
  - **1: gli appelli sono ai primi di settembre.**

62

Esercizio 1 (soluzione)

2) Scrivo l'espressione booleana che mi identifica la funzione:

vado in vacanza ad agosto se

(ho già dato tutti gli esami)

OR

(NON ho dato tutti gli esami

AND

NON ci sono appelli di tali esami ai primi di settembre).

a

+

$\bar{a}\ \bar{b}$

$$F = a + \bar{a}\ \bar{b}$$

63

Esercizio 2

Data la seguente proposizione:

Un impianto chimico è dotato di un sistema di allarme automatico che si attiva quando la temperatura della caldaia supera i 170°C e la pressione è superiore a due atmosfere, oppure non affluisce combustibile e la temperatura della caldaia è inferiore a 170°C.

identificare le variabili booleane e scrivere la funzione che identifica quando l'allarme è in funzione.

64

Esercizio 2 (soluzione)

1) Individuo le variabili booleane:

Un impianto chimico è dotato di un sistema di allarme automatico che si attiva quando la temperatura della caldaia supera i 170°C e la pressione è superiore a due atmosfere, oppure non affluisce combustibile e la temperatura della caldaia è inferiore a 170°C.

Posso individuare tre variabili:

- **variabile a: situazione temperatura**
  - **0: < 170°C**
  - **1: > 170°C**

65

Esercizio 2 (soluzione)

- **variabile b: situazione pressione**
  - **0: < 2 atm**
  - **1: > 2 atm**
- **variabile c: situazione combustibile**
  - **0: affluisce**
  - **1: non affluisce**

66

© 2000 M. Rebaudengo, A. Benso

Esercizio 2 (soluzione)

2) Scrivo l'espressione booleana che mi identifica la funzione:

l'allarme è attivo se:

(temp > 170°C AND press < 2 atm) → **ab**

OR → **+**

(NON affluisce combustibile AND temp < 170°C). → $\bar{a}\,\bar{c}$

F = $a\,b + \bar{a}\,\bar{c}$

Circuiti Combinatori

Funzioni booleane e circuiti combinatori

In un circuito logico combinatorio per ogni istante, il valore delle uscite è funzione del valore degli ingressi in quello stesso istante.

I circuiti combinatori sono *circuiti senza memoria*.

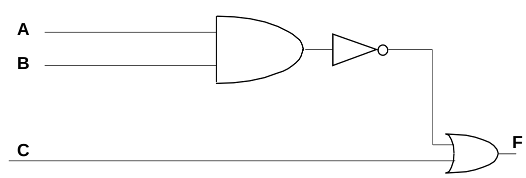
A partire da una funzione booleana F è possibile disegnare un circuito combinatorio in cui:

- **gli ingressi sono le variabili booleane**
- **le uscite sono la funzione booleana**

Esempio

Disegnare il circuito logico che rappresenta la seguente funzione:

F = $\overline{ab} + c$



Ritardi di un circuito combinatorio

Si definisce ritardo di una porta logica, il tempo che la porta impiega, dati gli ingressi, a valutare l'uscita.

A ogni porta di un circuito combinatorio è possibile assegnare un ritardo. Il ritardo massimo globale del circuito si calcola come somma dei ritardi delle porte che stanno sul percorso più lungo, ossia quel percorso che unisce uno degli ingressi con l'uscita e che passa attraverso il maggior numero di porte.

Esercizi

Disegnare i circuiti logici che rappresentano le seguenti funzioni:

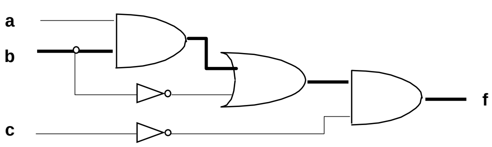
- **f1 =** $(ab + \bar{b})\,\bar{c}$
- **f2 =** $\overline{a\,b}\,\bar{c} + bc + ac$

Supponendo di assegnare ad ogni porta un ritardo pari a 10ns, calcolare i ritardi globali dei circuiti ottenuti.

© 2000 M. Rebaudengo, A. Benso

Esercizi (soluzione)

- $f1 = (ab + \overline{b})\,\overline{c}$

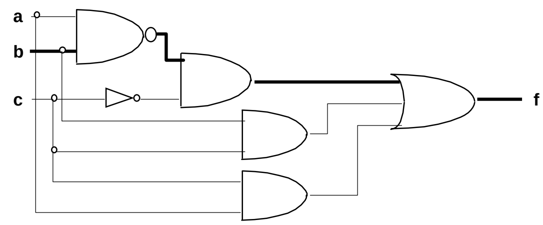


Ritardo globale : 10ns * 3 = 30 ns

73

Esercizi (soluzione)

- $f2 = \overline{ab}\,\overline{c} + bc + ac$



Ritardo globale : 10ns * 3 = 30 ns

74

Aritmetica degli elaboratori

Aritmetica degli elaboratori

Numeri binari

Addizione in binario

La addizione tra due numeri binari segue le stesse regole dell'addizione tra due numeri decimali:

- i due numeri sono incolonnati uno sopra l'altro.
- si effettua la somma delle cifre di pari peso.

77

Tabella di addizione

A_i	B_i	S_i	R_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$A_i + B_i = S_i$

R_i = *Carry* o *Riporto*

78

© 2000 M. Rebaudengo, A. Benso

Addizione in binario Esempio

	1	1	1				
		1	1	1	1	0	+
		1	0	1	0	0	
carry	1	1	0	0	1	0	

Sottrazione in binario

Anche la sottrazione segue le stesse regole della sottrazione tra due numeri decimali:

- **i due numeri sono incolonnati uno sopra l'altro.**
- **si effettua la sottrazione delle cifre di pari peso.**

Tabella di sottrazione

A_i	B_i	S_i	B_i
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$A_i - B_i = S_i$

B_i = ***Borrow*** **o** ***Prestito***

Sottrazione in binario Esempio

		1			
borrow	1	0	1	1	+
	0	1	1	0	
	0	1	0	1	

Somma e sottrazione in binario Esercizi

Effettuare le seguenti somme e sottrazioni in binario puro:

- $(34)_{10} + (77)_{10}$
- $(225)_{10} + (63)_{10}$
- $(84)_{10} - (37)_{10}$
- $(25)_{10} - (15)_{10}$

Somma e sottrazione in binario Soluzioni (1)

- $(34)_{10} + (77)_{10}$

0	1	0	0	0	1	0	+
1	0	0	1	1	0	1	=
1	1	0	1	1	1	1	

- $(225)_{10} + (63)_{10}$

	1	1	1	1	1	1	1		
0	1	1	1	0	0	0	0	1	+
0	0	0	1	1	1	1	1	1	=
1	0	0	1	0	0	0	0	0	

© 2000 M. Rebaudengo, A. Benso

Somma e sottrazione in binario Soluzioni (2)

- $(84)_{10}$ - $(37)_{10}$

1		1	1	1	1		
1	0	1	0	1	0	0	+
0	1	0	0	1	0	1	=
0	1	0	1	1	1	1	

- $(25)_{10}$ - $(15)_{10}$

1	1	1			
1	1	0	0	1	+
0	1	1	1	1	=
0	1	0	1	0	

Somma e sottrazione in binario Esercizi proposti

Effettuare le seguenti somme e sottrazioni in binario puro:

- $(15)_{10}$ + $(1)_{10}$ [R. 10000]
- $(229)_{10}$ + $(111)_{10}$ [R. 101010100]
- $(256)_{10}$ + $(64)_{10}$ [R. 101000000]
- $(1500)_{10}$ + $(547)_{10}$ [R. 11111111111]
- $(10)_{10}$ - $(6)_{10}$ [R. 100]
- $(39)_{10}$ - $(14)_{10}$ [R. 1101]
- $(18)_{10}$ - $(7)_{10}$ [R. 1011]
- $(64)_{10}$ - $(13)_{10}$ [R. 110011]

Condizione di Overflow

Nel caso in cui si abbia un numero limitato di bit a disposizione (come avviene nella realtà), si possono avere due casi particolari di *errore*:

- carry sul bit più significativo (MSB)
- borrow dal bit più significativo (MSB).

In entrambi i casi il numero di bit fissato non è sufficiente per rappresentare il risultato.

Tale condizione si dice condizione di *overflow*.

Esempio di Overflow

Considerando i numeri binari di 4 bit, effettuare la somma 9 + 7.

$(9)_{10} = (1001)_2$

$(7)_{10} = (0111)_2$

1	1	1	1		
	1	0	0	1	+
	0	1	1	1	=
1	0	0	0	0	

Il risultato non è rappresentabile su 4 bit, quindi si ha overflow

Esempio di Overflow

Considerando i numeri binari di 4 bit, effettuare la differenza 5 - 7.

$(5)_{10} = (0101)_2$

$(7)_{10} = (0111)_2$

1	1	1			
	0	1	0	1	-
	0	1	1	1	=
	1	1	1	0	

Il risultato non è rappresentabile su 4 bit, quindi si ha overflow.

Condizione di Overflow Esercizi proposti

Effettuare le seguenti operazioni aritmetiche tra numeri binari su 4 bit, e dire se si verifica overflow.

- $(4)_{10}$ + $(9)_{10}$ [R. 1101]
- $(4)_{10}$ - $(9)_{10}$ [R. overflow]
- $(10)_{10}$ + $(9)_{10}$ [R. overflow]
- $(9)_{10}$ - $(10)_{10}$ [R. overflow]

© 2000 M. Rebaudengo, A. Benso

Moltiplicazione tra due numeri binari

La moltiplicazione tra due numeri binari si esegue con lo stesso algoritmo adottato per i numeri decimali, detto di *somma e scorrimento*:

- si moltiplica il primo fattore (moltiplicando) per la cifra meno significativa del secondo fattore (moltiplicatore);
- si moltiplica il moltiplicando per la seconda cifra del moltiplicatore e si incolonna questo secondo risultato parziale sotto il primo, dopo averlo scalato di una posizione verso sinistra;
- dopo aver ripetuto questo procedimento sino all'ultima cifra del moltiplicatore, si esegue la somma di tutti i risultati parziali.

91

Esempio di moltiplicazione

Si esegua la seguente operazione:

- 11011 X 101

```
     1 1 0 1 1   X
         1 0 1   =
     ---------
     1 1 0 1 1
   0 0 0 0 0 -
 1 1 0 1 1 - -
 -------------
1 0 0 0 0 1 1 1
```

92

Esempio di moltiplicazione

Si esegua la seguente moltiplicazione:

- 1011 X 1011

```
      1 0 1 1   X
      1 0 1 1   =
      -------
      1 0 1 1
    1 0 1 1 -
  1 0 1 1 0 - -
  -------------
  1 1 1 1 0 0 1
```

93

Moltiplicazione tra numeri binari
Esercizi proposti

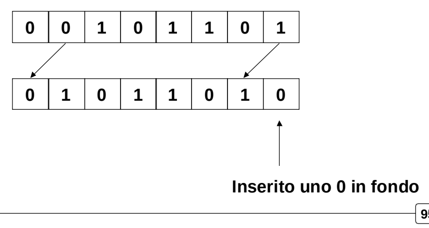
Si eseguano le seguenti moltiplicazioni:

- 00001 X 01101 [R. 1101]
- 10011 X 00010 [R. 100110]
- 11011 X 10110 [R. 1001010010]
- 01100 X 10101 [R. 11111100]
- 01011 X 11100 [R. 100110100]

94

Operazione di scalamento (shift) a sinistra

Uno scalamento a sinistra di un numero binario equivale ad una moltiplicazione per 2.



95

Scalamento a sinistra

Uno scalamento di N posizioni a sinistra equivale a moltiplicare il numero binario per 2^N.

96

© 2000 M. Rebaudengo, A. Benso

Scalamento a sinistra
Esercizi proposti

- Avendo una rappresentazione dei numeri binari su 6 bit, dire se effettuando una moltiplicazione per 8 del numero 12 si ha o no overflow. [R. Sì]
- Dire a quale moltiplicazione corrisponde lo scalamento di 5 posizioni a sinistra di un numero binario. [R. x 32]
- Dire se esiste una analogia tra l'operazione di scalamento a sinistra in un numero binario ed una operazione di scalamento a sinistra in un numero decimale. [R. Sì. In entrambi i casi si esegue una moltiplicazione per la base]
- Dire a quale divisione corrisponde lo scalamento di 2 posizioni a destra di un numero in base 3. [R. x 9]

97

Scalamento a destra

Uno scalamento a destra di un numero binario equivale ad una divisione per 2.

0	0	1	0	1	1	0	1

0	0	0	1	0	1	1	0

l'ultimo bit a sinistra esce dalla rappresentazione

Inserito uno 0 in cima

98

Scalamento a destra

Uno scalamento di N posizioni a sinistra equivale a dividere il numero binario per 2^N.

99

Scalamento a destra
Esercizi proposti

- Calcolare il risultato di uno shift a destra di 3 posizioni della rappresentazione binaria del numero decimale 15. [R. 1]
- Dire a quale divisione corrisponde lo scalamento di 4 posizioni a destra di un numero binario. [R. / 16]
- Dire se esiste una analogia tra l'operazione di scalamento a destra in un numero binario ed una operazione di scalamento a destra in un numero decimale. [R. Sì. In entrambi i casi si esegue una divisione per la base]
- Dire a quale divisione corrisponde lo scalamento di 2 posizioni a destra di un numero in base 8. [R. / 64]

100

Aritmetica degli elaboratori

Codifica BCD

101

Codifica BCD

Esiste una particolare rappresentazione dei numeri decimali, detta ***Binary Coded Decimal*** (***BCD***), in cui ciascuna cifra decimale è separatamente codificata in binario.

Per rappresentare l'insieme dei numeri da 0 a 9 ho bisogno di 4 bit.

Ogni cifra decimale è rappresentata con 4 cifre binarie.

102

© 2000 M. Rebaudengo, A. Benso

Tabella di conversione da decimale a BCD

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Tutte le altre configurazioni non sono ammesse.

103

Codifica BCD Esempi

- Convertire il numero decimale 106 in codifica BCD:

(1 0 6)

$(0001\ 0000\ 0110)_{BCD}$

- Ricavare il numero decimale corrispondente al numero $(0101100000000100111)_{BCD}$

$(0101\ 1000\ 0000\ 0100\ 0111)_{BCD}$

$(\ 5\ \ 8\ \ 0\ \ 4\ \ 7\ )_{10}$

104

Codifica BCD Esercizi proposti

Convertire in BCD ed in binario puro i seguenti numeri decimali e dire quale delle due rappresentazioni è più compatta :

- 298 [R. $(0010\ 1001\ 1000)_{BCD}$; $(100101010)_2$; BCD]
- 125 [R. $(0001\ 0010\ 0101)_{BCD}$ $(1111101)_2$; binaria]

Convertire in numero decimale i seguenti numeri, letti come codificati in binario e BCD:

- 0001010010011001

  [R. BCD: 1499; binario: 5273]
- 001110001101

  [R. BCD: impossibile; binario: 909]

105

Aritmetica degli elaboratori

Numeri relativi

106

Rappresentazione dei numeri relativi

Esistono tre diverse rappresentazione dei numeri relativi:

- rappresentazione in modulo e segno
- rappresentazione in complemento alla base
- rappresentazione in complemento alla base diminuita.

107

Aritmetica degli elaboratori

Numeri relativi

Modulo e segno

108

© 2000 M. Rebaudengo, A. Benso

Rappresentazione modulo e segno

Dati N bit, il bit più significativo indica il segno, ed i restanti N-1 indicano il valore assoluto del numero in binario puro.

S	Modulo

Il segno è codificato nel seguente modo:

0 : segno +

1 : segno -

109

Rappresentazione dello 0 in modulo e segno

Il numero 0 può avere, in modulo e segno, le seguenti due rappresentazioni:

0	0..........0
1	0..........0

110

Modulo e segno Esempio

Rappresentare in modulo e segno su 5 bit i numeri +12 e -5.

- +12:

0 : bit di segno (+)

1100: valore assoluto

01100: rappresentazione M&S

- -5:

1: bit di segno (-)

0101: valore assoluto

10101: rappresentazione M&S

111

Modulo e segno Esercizi proposti

- Rappresentare il massimo e il minimo valore esprimibile su 4 bit in modulo e segno. [R. ±7]
- Rappresentare, in modulo e segno i numeri:
  - $(+128)_{10}$ [R. 010000000]
  - $(-65)_{10}$ [R. 11000001]
  - $(+14)_{16}$ [R. 010100]
  - $(-1)_{10}$ [R. 11]

112

Somma in modulo e segno

Dati due numeri X e Y in modulo e segno su N bit, la somma X+Y si calcola:

- se X ed Y hanno lo stesso segno, il risultato ha lo stesso segno e modulo pari alla somma dei moduli;
- se X ed Y hanno segni discordi si valuta il numero di modulo maggiore. Il risultato ha segno concorde al numero di modulo maggiore. Il modulo è pari alla differenza tra il modulo maggiore ed il modulo minore.

113

Somma in modulo e segno Esempio 1

Calcolare la seguente somma in modulo e segno su 4 bit:

- $(+3)_{10}+(+2)_{10}$

+3 = 0011

+2 = 0010

I due numeri hanno lo stesso segno, per cui si sommano i moduli:

011 +

010 =

101

Il risultato è dunque uguale a 0101 (+5)

114

© 2000 M. Rebaudengo, A. Benso

Somma in modulo e segno Esempio 2

Calcolare la seguente somma in modulo e segno su 4 bit:

- $(-1)_{10} + (+5)_{10}$

-1 = 1001

+5 = 0101

I due numeri sono discordi. 0101 è il numero avente modulo maggiore, il risultato ha dunque segno positivo.

Il modulo è ricavato effettuando la differenza dei moduli.

101 -
001 =
100

Il risultato è uguale a 0100 (+4).

115

Somma in modulo e segno Esempio 3

Calcolare la seguente somma in modulo e segno su 4 bit:

- $(-7)_{10} + (+4)_{10}$

-7 = 1111

+4 = 0100

I due numeri sono discordi. 1111 è il numero avente modulo maggiore, il risultato ha dunque segno negativo.

Il modulo è ricavato effettuando la differenza dei moduli.

111 -
100 =
011

Il risultato è uguale a 1011 (-3).

116

Overflow in modulo e segno

Nel caso in cui i segni degli addendi siano concordi, si può verificare un overflow.

Tale overflow si verifica quando si ha un carry sul bit più significativo (MSB) effettuando la somma dei moduli.

117

Overflow in modulo e segno Esempio

Calcolare la seguente somma su 4 bit:

- -7 + (-1)

-7 = 1111

-1 = 1001

I numeri sono concordi, per cui si sommano i moduli.

1
111 +
001 =
1000

Si ha un carry sul MSB del modulo e dunque si ha overflow. Il risultato (-8) non è rappresentabile in modulo e segno su 4 bit.

118

Sottrazione in modulo e segno

Si converte l'operazione in somma operando una inversione del minuendo.

A - B = A + (-B).

L'inversione del minuendo si effettua cambiando il segno a B.

119

Sottrazione in modulo e segno Esempio

Si effettui la seguente sottrazione:

- +5 - (+15).

+5 = 00101

+15 = 01111

La sottrazione si converte in somma invertendo il segno del minuendo che diventa 11111: +5 + (-15)

I due numeri sono discordi. 1111 è il modulo maggiore, dunque il risultato ha segno negativo:

1111 -
0101 =
1010

Il risultato, con il segno corretto, vale dunque 11010 = -10.

120

© 2000 M. Rebaudengo, A. Benso

Somma e sottrazione in modulo e segno Esercizi proposti

Effettuare le seguenti somme e sottrazioni in modulo e segno su 6 bit:

- **14 + (-7)** **[R. 0111]**
- **(-24) + 15** **[R. 11001]**
- **4 - (12)** **[R. 11000]**
- **-2 - (-13)** **[R. 01001]**
- **-2 - (+31)** **[R. overflow]**

121

Moltiplicazione tra numeri in modulo e segno

Dati due numeri X e Y rappresentati in modulo e segno su N bit per effettuare il prodotto tra X e Y occorre:

- **separare il modulo dal segno di ciascun fattore;**
- **il modulo del risultato è ottenuto operando il prodotto dei moduli;**
- **il segno è ottenuto mediante l'applicazione della regola elementare del prodotto dei segni:**

X	Y	Ris
0	0	0
0	1	1
1	0	1
1	1	0

122

Moltiplicazione in modulo e segno Esempio

Eseguire il seguente prodotto tra numeri in modulo e segno:

10011 x 00101

Il segno è negativo.

Il prodotto dei moduli vale:

```
  11 x
 101 =
  11 +
1100 =
1111
```

Il risultato vale 11111 = -15.

123

Aritmetica degli elaboratori

Numeri relativi

Complemento alla base

124

Rappresentazione in complemento alla base

Nella rappresentazione in complemento alla base si dà una diversa attribuzione dei pesi associati alle cifre che codificano il numero:

- **alla cifra più alta è associato un peso negativo**
- **le cifre più basse hanno un peso positivo.**

125

Rappresentazione in complemento alla base (segue)

Data una rappresentazione in complemento alla base N di un numero M su n cifre

$(M)_N = (d_n d_{n-1} \dots d_2 d_1 d_0)_N$

la formula di conversione è la seguente:

$$(M)_{10} = - d_n * N^n + d_{n-1} * N^{n-1} + \dots + d_2 * N^2 + d_1 * N^1 + d_0 * N^0$$

126

© 2000 M. Rebaudengo, A. Benso

Conversione da numero decimale a complemento a 2

Occorre distinguere i due casi:

- **il numero è positivo: la rappresentazione è identica alla rappresentazione in modulo e segno**
- **il numero è negativo:**
  - **si scrive il numero binario positivo corrispondente;**
  - **si effettua l'operazione di complemento a due del numero binario.**

127

Operazione di complemento a 2 di un numero binario

Esistono due modi di effettuare il complemento a 2 di un numero binario:

- **Metodo 1:**
  - **si complementano tutti i bit;**
  - **si somma 1.**
- **Metodo 2:**

considerando il numero a partire dal LSB:

- **copio tutti gli zeri fino a trovare il primo uno;**
- **copio l'uno;**
- **complemento i restanti bit.**

128

Conversione da numero decimale a complemento a 2

Osservazioni:

- **i numeri negativi hanno sempre il MSB = 1**
- **esiste una sola rappresentazione dello zero: (0...0)**

129

Conversione da numero decimale a complemento a 2 Esempi

Rappresentare in complemento a 2 su 6 cifre i numeri

- **+3**
- **-6**
- **-20**

+3 è positivo: la rappresentazione è uguale a quella in modulo e segno: (000011)

-6 è negativo; uso il metodo 1 :

- **il numero corrispondente positivo (+6) vale 000110**
- **complemento bit a bit ed ottengo 111001**
- **sommo 1 ed ottengo 111010**

130

Conversione da numero decimale a complemento a 2 Esempi (segue)

-20 è negativo; uso il metodo 2:

- **il numero corrispondente positivo (+20) vale 10100;**
- **partendo dal LSB ricopio tutti gli zeri fino a trovare il primo uno;**
- **copio l'uno;**
- **complemento i restanti bit.**

0 1 0 1 0 0

1 0 1 1 0 0

131

Conversione da numero decimale a complemento a 2 Esercizi proposti

Rappresentare in complemento a 2 su 5 bit i seguenti numeri decimali:

- **+11** **[R. 01011]**
- **-8** **[R. 11000]**
- **+1** **[R. 00001]**
- **-1** **[R. 11111]**
- **0** **[R. 00000]**
- **+16** **[R. imposs.]**
- **-16** **[R. 10000]**

132

© 2000 M. Rebaudengo, A. Benso

Conversione da complemento a 2 a decimale

Occorre distinguere i due casi:

- il numero è positivo (MSB=0): l'operazione è identica alla conversione da modulo e segno
- il numero è negativo (MSB=1): esistono tre metodi diversi.

133

Conversione da complemento a 2 a decimale

- Metodo 1:

  dato un numero M in complemento a 2 negativo (MSB=1) su n bit si ha:

  $(M)_{10}=-2^n+$(numero reppresentato dalle restanti n-1 cifre)
- Metodo 2:

  –si rieffettua l'operazione di complemento a due del numero. Infatti $((\text{numero})_{CA2})_{CA2}$ = numero
- Metodo 3:

  –si sottrae 1

  –si complementano tutti i bit ottenendo il modulo del numero decimale corrispondente.

134

Conversione da complemento a 2 a decimale Esempi

Convertire in numero decimale i seguenti numeri in complemento a 2:

- 0110

  $0110 = - 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = +4 + 2 = +6$

- 1001

Metodo 1:

$1001 = - 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = -8 +1 = -7$

135

Conversione da complemento a 2 a decimale Esempi *(segue)*

Metodo 2:

- Effettuo il complemento a 2 di 1001 ottenendo 0111 che corrisponde a $(7)_{10}$.
- Quindi il numero $(1001)_{CA2}= (-7)_{10}$

Metodo 3:

- Sottraggo 1, ottenendo 1000.
- Complemento i bit e ottengo 0111 che equivale a $(7)_{10}$.
- Quindi il numero $(1001)_{CA2}= (-7)_{10}$

136

Conversione da complemento a 2 a decimale Esercizi proposti

Convertire in numero decimale i seguenti numeri espressi in complemento a 2:

- 10011 [R. -13]
- 01101 [R. +13]
- 11111 [R. -1]
- 10000 [R. -16]
- 00000 [R. 0]

137

Somma in complemento a due

- Dati due numeri X e Y in complemento a due su N bit, la somma X+Y si calcola sommando aritmeticamente tutti i bit degli addendi, ***compreso quello di segno***.
- L'eventuale carry oltre il bit di segno viene tralasciato.

138

© 2000 M. Rebaudengo, A. Benso

Esempio

Calcolare (-7) + 10 in CA2 su 5 bit.

-7 = 11001

10 = 01010

```
 1
  1 1 0 0 1 +
  0 1 0 1 0 =
  0 0 0 1 1
```

Tralasciando il carry, il risultato vale 00011 = +3.

139

Differenza in complemento a due

Ci si deve ricondurre al caso della somma trasformando la differenza nella somma del primo numero con l'inverso del secondo.

A-B = A+(-B)

140

Differenza in complemento a due Esempio

Calcolare (-7) - 4 in complemento a due su 5 bit.

Si opera come se l'operazione fosse (-7) + (-4).

-7 = 11001

-4 = 11100

```
 1
  1 1 0 0 1 +
  1 1 1 0 0 =
  1 0 1 0 1
```

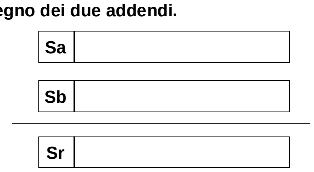
Tralasciando il carry, il risultato vale 10101 = -11.

141

Overflow in complemento a 2

Si può verificare un overflow nell'operazione tra due numeri in complemento a due quando si effettua la somma di due numeri concordi.

Si ha overflow quando il segno del risultato è diverso dal segno dei due addendi.



Sa = Sb ≠ Sr

142

Overflow in CA2 Esempio

Calcolare (-12) + (-6) in complemento a due su 5 bit:

(-12) = 10100

(-6) = 11010

```
 1
  1 0 1 0 0 +
  1 1 0 1 0 =
  0 1 1 1 0
```

Si ha carry oltre il bit di segno, che viene ignorato.

Il bit di segno è diverso da quello degli addendi, dunque la somma ha prodotto overflow.

143

Esempio

Eseguire la seguente operazione in complemento a due su 9 bit:

- 255 + 2 + (-127).

L'ordine con cui eseguo le due operazioni è aritmeticamente equivalente.

Provo ad eseguire prima 255 + 2.

Ottengo:

```
 01111111 +
 00000010 =
 10000001
```

Overflow!!

144

© 2000 M. Rebaudengo, A. Benso

Esempio
(segue)

Provo ad eseguire prima la somma +2 + (-127)

0000000010
<u>1100000001</u>
1100000011 +
<u>0111111111</u>
0100000010

Il risultato finale vale +130.

145

Esercizi proposti

Eseguire le seguenti operazioni

- 1101 + 1110
- 1011 + 0011
- 1001 + 1100

con i numeri rappresentati

① In complemento a due;
② in modulo e segno.

146

Esercizi proposti

Soluzioni:

- 1101 + 1110 [R. M&S: overflow; CA2: 1011]
- 1011 + 0011 [R. M&S: 0; CA2: 1110]
- 1001 + 1100 [R. M&S: 1101; CA2: overflow]

147

Esercizi proposti

Eseguire le seguenti operazioni

- 11101 - 01010
- 00011 - 00101
- 10010 - 11100

con i numeri rappresentati

① In complemento a due;
② in modulo e segno.

148

Esercizi proposti

Soluzioni:

- 11101 - 01010 [R. M&S: overflow; CA2: 10011]
- 00011 - 00101 [R. M&S: 10010; CA2: 11110]
- 10010 - 11100 [R. M&S: 01010; CA2: 10110]

149

Moltiplicazione tra numeri in complemento a due

Dati due numeri X e Y rappresentati in complemento a due su N bit per effettuare il prodotto tra X e Y occorre ricondursi al caso della moltiplicazione tra numeri in modulo e segno:

- occorre ricavarsi il modulo di ciascun fattore (per i numeri negativi occorre effettuare la complementazione);
- il modulo del risultato è ottenuto operando il prodotto dei moduli;
- il segno è ottenuto mediante l'applicazione della regola elementare del prodotto dei segni;
- se il risultato è negativo occorre effettuare la complementazione del modulo per ottenere il valore negativo.

150

© 2000 M. Rebaudengo, A. Benso

Moltiplicazione in CA2: Esempio

Eseguire il seguente prodotto tra numeri in complemento a 2:

00011 x 11101

I due moduli valgono: 00011 e 00011

Il segno è negativo, il prodotto dei moduli vale:

```
   11 x
   11
   11 +
  110
 1001
```

Devo complementare il valore 01001. Ottengo come risultato il numero 10111 = -9.

151

Moltiplicazione in CA2 Esercizi proposti

Eseguire le seguenti operazioni

- 11101 x 00110
- 11110 x 00101
- 00010 x 11100

con i numeri rappresentati

① In complemento a due;

② in modulo e segno.

152

Operazioni in CA2 Esercizi

Calcolare la seguente espressione:

1001 x (1010 - 01011 x 111) + 01011 x 01100

considerando i numeri espressi in complemento a due.

Risolvere le seguenti espressioni:

- $(-1B)_{16} + (27)_{10} \times (21)_3 + (13)_{10} \times (-C)_{16}$
- $(122)_{10} - (A7)_{16} + (456)_7 \times ((-14)_8 + (10001)_2)$

153

Aritmetica degli elaboratori
Numeri relativi
Complemento alla base diminuita

154

Complemento alla base diminuita

Data una rappresentazione in complemento alla base N di un numero M su n cifre

$(M)_N = (d_n d_{n-1} .... d_2 d_1 d_0)_N$

la formula di conversione è la seguente:

$(M)_{10} = - d_n*(N^{n-1}-1) + d_{n-1}*N^{n-1} + ... + d_2*N^2 + d_1*N^1 + d_0*N^0$

Nel caso di sistema binario la formula di conversione è la seguente:

$(M)_{10} = - d_n*(2^{n-1}-1) + d_{n-1}*2^{n-1} + ... + d_2*2^2 + d_1*2^1 + d_0*2^0$

155

Conversione da numero decimale a complemento ad 1

- Numeri positivi: la rappresentazione è identica alla rappresentazione modulo e segno.
- Numeri negativi: la rappresentazione è ottenuto invertendo bit a bit il numero positivo corrispondente.

156

© 2000 M. Rebaudengo, A. Benso

Complemento ad 1: Esempi

Il numero in complemento ad 1 011011 vale:

1+2+8+16 = 27

Il numero in complemento ad 1 101011 vale:

1+2+8-31 = -20.

157

Complemento ad 1 Esempi

Rappresentazione in complemento ad 1 dei numeri 12 e -19 su 6 bit:

12 è un numero positivo e dunque la sua rappresentazione in complemento ad 1 è equivalente ad una rappresentazione modulo e segno:

001100

-19 è un numero positivo dunque rappresento il corrispondente numero positivo su 6 bit: 010011 e lo complemento:

101100

158

Complemento ad 1 Esercizi proposti

Si calcoli il valore decimale dei seguenti numeri in complemento ad 1:

- 000000
- 011100
- 111100
- 100000
- 111111
- 000001
- 011111

159

Complemento ad 1 Esercizi proposti

Si calcoli la rappresentazione in complemento ad 1 su 7 bit dei seguenti valori decimali:

- il massimo valore positivo
- il massimo valore negativo
- lo zero
- i numeri +60, -25, +12, -55.

160

Aritmetica degli elaboratori

Limiti delle rappresentazioni

161

Limiti delle rappresentazioni

Fissato un numero N di bit disponibili, i limiti delle rappresentazioni fin'ora trattate sono:

- Binario puro $0...+(2^{N}-1)$
- Modulo e segno $-(2^{N-1}-1)...-0+0...+(2^{N-1}-1)$
- Complemento a due $-(2^{N-1})...0...+(2^{N-1}-1)$

162

© 2000 M. Rebaudengo, A. Benso

Limiti delle rappresentazioni

Ad esempio, fissato un numero di 5 bit disponibili, i limiti delle rappresentazioni fin'ora trattate sono:

- **Binario puro** **0...31**
- **Modulo e segno** **-15...-0+0...+15**
- **Complemento a due** **-16...0...+15**

163

Rappresentazione di oggetti

Per rappresentare K oggetti con numeri binari, occorrono almeno N bit, dove N è pari a:

$$N = \lceil \log_2 K \rceil$$

dove per $\lceil X \rceil$ si intende il numero intero immediatamente superiore od uguale ad X.

Ad esempio, per rappresentare 77 oggetti, sono necessari almeno 7 bit:

$$N = \lceil \log_2 K \rceil = \lceil \log_2 77 \rceil = \lceil 6.267 \rceil = 7$$

Con 6 bit, il massimo numero rappresentabile sarebbe stato 2^6=64.

164

Aritmetica degli elaboratori

Numeri frazionari

165

Rappresentazione dei numeri decimali frazionari

Nel sistema decimale per separare parte intera e parte frazionaria di un numero si usa *la virgola* (in notazione latina) oppure *il punto* (in notazione anglosassone).

Le cifre che compongono la parte frazionaria hanno un peso pari ad una potenza negativa della base 10.

166

Rappresentazione dei numeri decimali frazionari

Supponendo di avere un numero decimale M frazionario composto da *n* cifre di parte intera e da *m* cifre di parte frazionaria:

$$(M)_{10} = (d_{n-1} \ldots d_2 d_1 d_0 \,.\, d_{-1} d_{-2} \ldots d_{-(m-2)} d_{-(m-1)} d_{-m})_{10}$$

la formula di conversione è la seguente:

$$(M)_{10} = d_{n-1} * 10^{n-1} + \ldots + d_1 * 10^1 + d_0 + d_{-1} * 10^{-1} + \ldots + d_{-(m-1)} * 10^{-(m-1)} + d_{-m} * 10^{-m}$$

167

Rappresentazione dei numeri frazionari

Esempio:

- **823.456**

$$8 * 10^2 + 2 * 10^1 + 3 + 4 * \underbrace{10^{-1}}_{\frac{1}{10^1}} + 5 * \underbrace{10^{-2}}_{\frac{1}{10^2}} + 6 * \underbrace{10^{-3}}_{\frac{1}{10^3}}$$

168

© 2000 M. Rebaudengo, A. Benso

Rappresentazione dei numeri binari frazionari

In base 2 la formula di conversione è la seguente:

$(M)_{10} = (d_{n-1}....d_2d_1d_0\ .\ d_{-1}d_{-2}....d_{-(m-2)}d_{-(m-1)}d_{-m})_{10}$

$(M)_{10}=d_{n-1}*2^{n-1}+...+d_1*2^1+d_0+d_{-1}*2^{-1}+...+d_{-(m-1)}*2^{-(m-1)}+d_{-m}*2^{-m}$

Per esempio considerando il numero in base 2:

10101.1101

Esso è pari a:

$1*2^4+1*2^2+1*2^0+1*\underbrace{2^{-1}}_{\frac{1}{2^1}}+1*\underbrace{2^{-2}}_{\frac{1}{2^2}}+1*\underbrace{2^{-4}}_{\frac{1}{2^4}} = 21.8125$

Numeri binari frazionari Esercizi

Calcolare a quale numero decimale corrispondono i seguenti numeri binari frazionari:

- **0.1101001**
- **1011. 0101**
- **11.10011**

Numeri binari frazionari Esercizi

Soluzioni:

- **0.1101001** **[R. 0.8203125]**
- **1011. 0101** **[R. 11.3125]**
- **11.10011** **[R. 3.59375]**

Metodi di rappresentazione dei numeri frazionari

Esistono due metodi utilizzati per rappresentare i numeri frazionari su elaboratore:

- **la rappresentazione in virgola fissa (*fixed point*);**
- **la rappresentazione in virgola mobile (*floating point*).**

Rappresentazione in virgola fissa

Data una sequenza di bit si assume che la posizione della virgola sia fissata in un preciso punto al'interno della sequenza.

parte intera	.	parte frazionaria
M bit		F bit

Rappresentazione in virgola fissa

Ad esempio si supponga di rappresentare i numeri con 8 bit. La virgola sia posizionata in modo da utilizzare 6 bit per la codifica della parte intera, 2 per quella frazionaria.

- **La sequenza 01101111 rappresenta il numero frazionario binario 011011.11 = 27.75.**
- **Il numero binario frazionario 0.11 viene rappresentatto in virgola fissa con 00000011.**

© 2000 M. Rebaudengo, A. Benso

Tabella delle potenze negative di 2

Conversione da decimale a binario per numeri in virgola fissa

La conversione da sistema decimale a sistema binario in virgola fissa, in cui si impiegano *m* bit per la parte intera ed *n* bit per quella frazionaria si esegue con il seguente algoritmo:

- **si considerano separatamente parte intera e parte frazionaria;**
- **la parte intera viene convertita in un numero binario a m bit;**

Conversione da decimale a binario per numeri in virgola fissa
(segue)

- **la parte frazionaria viene convertita in un numero binario a n bit effettuando moltiplicazioni successive per 2 separando la parte intera (0 od 1) da quella frazionaria così ottenuta, finchè:**
  - **si ottiene parte frazionaria nulla**
  - **oppure si sono ricavati tutti gli n bit disponibili per la rappresentazione in virgola fissa. In questo caso la rappresentazione del sistema binario è una approssimazione di quella nel sistema decimale.**

Esempio di conversione

Convertire in binario il numero 6.25 su 6 bit di cui 3 per la parte intera e 3 per la parte frazionaria:

- **separo 6 da 0.25;**
- **la parte intera la rappresento come 110**
- **per quanto riguarda la parte frazionaria eseguo iterative moltiplicazioni per 2:**

parte frazionaria	0.25	.5	.0
parte intera		0	1

Il resto è 0 quindi mi fermo

Risultato
6.25=110.01

Approssimazione dei numeri frazionari

In generale ad un numero frazionario decimale con un numero limitato di cifre può corrispondere un numero frazionario binario con un numero infinito di cifre.

In tal caso ci si ferma quando si raggiunge la precisione desiderata oppure quando non si hanno più bit a disposizione.

Approssimazione dei numeri frazionari

Se il numero N di bit a disposizione per la parte frazionaria è fissato, la massima precisione raggiungibile è pari a:

$$\frac{1}{2^N}$$

Nel caso in cui tale precisione non sia sufficiente, il numero binario ottenuto è affetto da un errore di underflow.

© 2000 M. Rebaudengo, A. Benso

Esempio di approssimazione

Convertire in binario il numero 0.3 con la precisione dell'1%.

Eseguo le moltiplicazioni iterative:

0.3	.6	.2	.4	.8	.6	.2	.4
	0	1	0	0	1	1	0
	.5	.25	.125	.625	.031	.015	.0078

0.0078 è < 0.01 e quindi ho raggiunto la precisione richiesta.

Il numero in virgola fissa vale 0.0100110.

181

Esercizi

Convertire in binario i seguenti numeri:

- 0.625
- 0.03125
- 0.828125

182

Esercizi (soluzioni)

Convertire in binario i seguenti numeri:

- 0.625 [R. 0.101]
- 0.03125 [R. 0.00001]
- 0.828125 [R. 0.110101]

183

Esercizi

Convertire nel sistema di numerazione decimale i seguenti numeri binari espressi in virgola fissa su 8 bit di cui 4 sono utilizzati per rappresentare la parte intera e 4 per la parte frazionaria:

- 01110000
- 01101010
- 10110101
- 10101111
- 00001011

184

Esercizi (soluzioni)

Convertire nel sistema di numerazione decimale i seguenti numeri binari espressi in virgola fissa su 8 bit di cui 4 sono utilizzati per rappresentare la parte intera e 4 per la parte frazionaria:

- 01110000 [R. 7.0]
- 01101010 [R. 6.625]
- 10110101 [R. 11.3125]
- 10101111 [R. 10.9375]
- 00001011 [R. 0.6875]

185

Esercizi

Convertire nella notazione in virgola fissa con 5 bit di parte intera e 5 bit di parte frazionaria i seguenti numeri assoluti espressi nel sistema decimale:

- 26.375
- 25.346
- 1.225
- 1.234621
- 31.1025
- 1/3

186

© 2000 M. Rebaudengo, A. Benso

Esercizi (soluzioni)

Convertire nella notazione in virgola fissa con 5 bit di parte intera e 5 bit di parte frazionaria i seguenti numeri assoluti espressi nel sistema decimale:

- 26.375 [R. 1101001100 = 26.375]
- 25.346 [R. 1100101011 = 25.34375]
- 1.225 [R. 0000100111 = 1.21875]
- 1.234621 [R. 0000100111 = 12.21875]
- 31.1025 [R. 1111100011 = 31.09375]
- 1/3 = 0.333333 [R. 0000001010 = 0.3125]

187

Esercizi (soluzioni)

Tutti i risultati, escluso il primo, sono affetti da un errore di underflow.

I numeri 1.225 e 1.234621 hanno, in binario, la stessa rappresentazione.

Infatti:

1.234621 - 1.225 = .009621 < .03125 ($1/2^5$) che è il minimo scarto rappresentabile su 5 bit.

188

Rappresentazione dei numeri relativi con notazione in virgola fissa

Per rappresentare i numeri con segno nella notazione in virgola fissa, si può utilizzare una delle due rappresentazioni per i numeri relativi:

- modulo e segno;
- complemento a due.

189

Numeri relativi frazionari rappresentati in modulo e segno

- Viene codificato il modulo del numero frazionario su N-1 bit mediante l'algoritmo dei numeri frazionari positivi;
- il bit più significativo codifica il segno.

190

Esempio

Rappresentazione del numero -3.375 in modulo e segno in virgola fissa con 2 bit di parte intera e 5 bit di parte frazionaria:

- la parte intera si rappresenta come 11;
- per la parte frazionaria si eseguono le moltiplicazioni iterative:

  0.375 x 2 = 0.75 => bit 0

  0.75 x 2 = 1.5 => bit 1

  0.5 x 2 = 1.0 => bit 1

  la parte frazionaria si rappresenta come 01100
- il segno vale 1.
- Il numero -3.375 si rappresenta come 11101100

191

Esercizi

Convertire nel sistema decimale i seguenti numeri rappresentati in modulo e segno e virgola fissa su 8 bit con parte frazionaria rappresentata su 5 bit:

- 10110101
- 01101100
- 11111001
- 11111111
- 10000001

192

© 2000 M. Rebaudengo, A. Benso

Esercizi (soluzioni)

Convertire nel sistema decimale i seguenti numeri rappresentati in modulo e segno e virgola fissa su 8 bit con parte frazionaria rappresentata su 5 bit:

- **10110101 [R. -1.65625]**
- **01101100 [R. +3.375]**
- **11111001 [R. -3.78125]**
- **11111111 [R. -3.96875]**
- **10000001 [R. -0.03125]**

193

Esercizi

Convertire in base binaria su 11 bit con la notazione modulo e segno e virgola fissa con 5 bit di parte frazionaria i seguenti numeri:

- **+2.03125**
- **-10.25**
- **+25.35**
- **-24.001**

194

Esercizi (soluzioni)

Convertire in base binaria su 11 bit con la notazione modulo e segno e virgola fissa con 5 bit di parte frazionaria i seguenti numeri:

- **+2.03125 [R. 0 00010 00001]**
- **-10.25 [R. 1 01010 01000]**
- **+25.35 [R. 0 11001 01011*]**
- **-24.001 [R. 1 11000 00000*]**

*** = underflow**

195

Numeri relativi frazionari rappresentati in complemento a due

- **Viene codificato il modulo del numero frazionario su N bit mediante l'algoritmo dei numeri frazionari positivi;**
- **Nel caso di numero negativo si effettua l'operazione di complemento a due.**

196

Esempio

Rappresentazione del numero -3.875 in complemento a due in virgola fissa con 2 bit di parte intera e 5 bit di parte frazionaria:

- **si rappresenta il corrispondente numero positivo 3.875;**
- **la parte intera si rappresenta come 011;**
- **per la parte frazionaria si eseguono le moltiplicazioni iterative:**

  0.875 x 2 = 1.75 => bit 1

  0.75 x 2 = 1.5 => bit 1

  0.5 x 2 = 1.0 => bit 1

  la parte frazionaria si rappresenta come 11100
- **Il numero +3.875 si rappresenta come 01111100**
- **il complemento a due di 01111100 vale 10000100**

$\boxed{\textbf{-3.875 = 10000100}}$

197

Esempio

Convertire in decimale il numero 11100100 rappresentato in complemento a due e virgola fissa con 5 bit di parte frazionaria:

- **il numero è negativo e si converte nel corrispondente positivo effettuando l'operazione di complemento a due ottenendo:**

  000 11100

  modulo=0 parte frazionaria

  0.5 + 0.25 + 0.125 = 0.875.

Il numero decimale corrispondente vale -0.875.

198

© 2000 M. Rebaudengo, A. Benso

Esercizi

Convertire nel sistema decimale i seguenti numeri rappresentati in complemento a due e virgola fissa su 8 bit con parte frazionaria rappresentata su 5 bit:

- 10110101
- 01101100
- 11111001
- 11111111
- 10000001

199

Esercizi (soluzioni)

Convertire nel sistema decimale i seguenti numeri rappresentati in complemento a due e virgola fissa su 8 bit con parte frazionaria rappresentata su 5 bit:

- 10110101 [R. -2.34375]
- 01101100 [R. +3.375]
- 11111001 [R. -0.21875]
- 11111111 [R. -0.03125]
- 10000001 [R. -3.96875]

200

Esercizi

Convertire in base binaria su 11 bit con la notazione in complemento a due e virgola fissa con 5 bit di parte frazionaria i seguenti numeri:

- +2.84575
- -10.2823
- +25.3672
- -24.52462

201

Esercizio

Calcolare l'intervallo di numeri rappresentabili in virgola fissa utilizzando *m* bit per la parte intera e *n* per la parte frazionaria nei seguenti casi:

- rappresentazione dei numeri assoluti;
- rappresentazione dei numeri relativi in modulo e segno;
- rappresentazione dei numeri relativi in complemento a due.

202

Soluzione (1)

Il numero avente il maggior valore assoluto rappresentabile è la sequenza di $n+m$ bit di valore 1, di cui n rappresentano la parte intera e m la parte frazionaria.

Esse assumono i seguenti valori:

- parte intera: $2^n - 1$
- parte frazionaria: $1-2^{-m}$.
- Il massimo valore rappresentabile vale dunque: 2^n-2^{-m}.

Il minimo valore rappresentabile vale 0.

L'intervallo di rappresentazione vale dunque $[0, 2^n-2^{-m}]$.

203

Soluzione (2)

Il numero avente il massimo valore positivo rappresentabile è la sequenza di $n+m-1$ bit di valore 1, di cui $n-1$ rappresentano la parte intera e m la parte frazionaria, preceduti dal bit di segno 0.

Tale numero assume il seguente valore:

- parte intera: $2^{n-1} - 1$
- parte frazionaria: $1-2^{-m}$.
- Il massimo valore rappresentabile vale dunque: $2^{n-1}-2^{-m}$.

204

© 2000 M. Rebaudengo, A. Benso

Soluzione (2) cont.

Il minimo valore rappresentabile è dato dalla sequenza di $n+m$ bit di valore 1:

Tale numero assume il seguente valore:

- parte intera: $-(2^{n-1} - 1)$
- parte frazionaria: $1-2^{-m}$.
- Il minimo valore rappresentabile vale dunque: $-(2^{n-1}-2^{-m})$.

L'intervallo di rappresentazione è dunque: $[-(2^{n-1}-2^{-m}), 2^{n-1}-2^{-m}]$.

205

Soluzione (3)

Il numero avente il massimo valore positivo rappresentabile è la sequenza di $n+m-1$ bit di valore 1, di cui $n-1$ rappresentano la parte intera e m la parte frazionaria, preceduti dal bit di segno 0, come per la rappresentazione modulo e segno.

Tale numero assume il valore $+(2^{n-1}-2^{-m})$.

Il minimo valore rappresentabile è la sequenza di $m+n-1$ bit di valore 0 preceduta da un bit di segno di valore 1.

Tale numero assume il valore $-2^{(n-1)}$.

L'intervallo di rappresentazione è dunque: $[-2^{n-1}, 2^{n-1}-2^{-m}]$.

206

Rappresentazione in virgola mobile

La notazione in virgola mobile (floating point) è basata sulla rappresentazione dei numeri in *notazione scientifica*.

Nella notazione scientifica un numero N viene espresso nella seguente forma:

$$N = \pm \text{ mantissa } * \text{ base }^{\text{esponente}}$$

207

Floating point Esempio

$(-213.0465)_{10}$ si può scrivere nelle seguenti notazioni scientifiche:

- $-2.130465*10^2$
- $-0.2130465*10^3$.

$(+1011.011)_2$ si può rappresentare nelle seguenti forme:

- $+1.011011*2^3$
- $+0.1011011*2^4$
- $+1011011*2^{-3}$.

208

Standard IEEE 754

In binario, per evitare ambiguità di rappresentazione di uno stesso numero si utilizza una *forma normalizzata* (standard IEEE 754) su 32 bit suddivisi nel seguente modo:

	S	Esponente	Mantissa (modulo)
singola precisione (SP):		8 bit	23 bit
doppia precisione (DP):		11 bit	52 bit

209

Standard IEEE 754

S	Esponente	Mantissa (modulo)

Mantissa: E' sempre normalizzata nella forma 1.XXXXX; l'1 viene sottinteso.

Esponente: E' rappresentato come numero senza segno su 8 bit in eccesso 127, cioè i valori da -126 a 127 sono messi in corrispondenza con i valori da 1 a 254 per non dover gestire il segno dell'esponente.

S: 0 = positivo
1 = negativo

210

© 2000 M. Rebaudengo, A. Benso

Standard IEEE 754

La rappresentazione floating point IEEE 754 è quindi nella forma:

$$numero = \pm\, 1.XXXX * 2^{(YYYY)_2}$$

1/0	YYYY	XXXX

211

Conversione da decimale a IEEE 754

I passi da eseguire per convertire un numero decimale N nella sua rappresentazione floating point è:

- rappresentarlo in binario utilizzando il numero necessario di bit;
- trasformarlo nella forma normalizzata $\pm 1.XXXX*2^{(YYYY)_2}$ spostando il punto decimale verso sinistra o verso destra di n posizioni;
- l'esponente è pari a 127+n dove n è il numero di posizioni di cui è stata spostata la virgola per passare nella rappresentazione normalizzata;
- la parte frazionaria della mantissa normalizzata (XXXX) viene memorizzata nei 23 bit meno significativi.

212

Conversione da decimale a IEEE 754 Esempio

Convertire in formato IEEE 754 SP il numero decimale 13.25

- il numero è positivo per cui il segno è 0;
- il corrispondente numero binario è 1101.01;
- scrivo la forma normalizzata spostando la virgola verso sinistra di 3 posizioni

1 1 0 1 . 0 1 = 1 . 1 0 1 0 1

mantissa

- l'esponente in eccesso 127 vale :

127 + 3 = 130 = 10000010

213

Conversione da decimale a IEEE 754 Esempio *(segue)*

- Il numero, scritto in notazione esponenziale normalizzata è dunque:

$$13.25 = +\, 1.10101 * 2^{(10000010)_2}$$

0	10000010	10101 000000000000000000

- La rappresentazione richiesta è quindi:

0 10000010 10101000000000000000000

214

Conversione da decimale a IEEE 754 Esercizi

Rappresentare in formato IEEE 754 i seguenti numeri decimali:

- 5.0
- -9.25
- 12.8125

215

Conversione da decimale a IEEE 754 Esercizi *(soluzioni)*

Rappresentare in formato IEEE 754 i seguenti numeri decimali:

- 5.0 [R. 0 10000001 0100...0]
- -9.25 [R. 1 10000010 0010100...0]
- 12.8125 [R. 0 10000010 100110100...0]

216

© 2000 M. Rebaudengo, A. Benso

Standard IEEE 754
Intervallo di rappresentazione

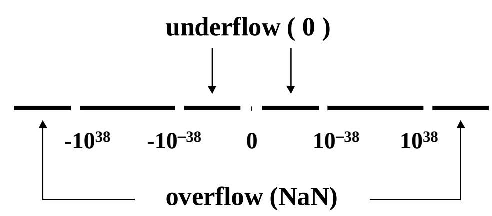
$M_{min} = 1$ $M_{max} < 2$
$E_{min} = -126$ $E_{max} = +127$

Valori rappresentabili:

- zero
- **$(2^{-126} \ldots 2^{127}) \sim (10^{-38} \ldots 10^{38})$**
- **$(-2^{127} \ldots -2^{-126}) \sim (-10^{38} \ldots -10^{-38})$**

217

Standard IEEE 754
Intervallo di rappresentazione



218

Conversione da IEEE 754 a decimale

E' necessario distinguere diversi casi. Chiamando s il segno, e l'esponente, ed m la mantissa si può avere:

E	M	valore
0	0	± 0
0	$\neq 0$	non normalizzato
max	0	$\pm\infty$
max	$\neq 0$	NaN (Not a Number)

219

Conversione da IEEE 754 a decimale
(metodo veloce)

Sia dato un numero in formato IEEE 754, ad esempio il numero: 1 10000001 0110...0

- **scrivo 1.m**

  1. 0110...0
- **calcolo k=e-127**

  e=10000001=129 - 127 = 2 = k
- **sposto la virgola verso destra (se k>0) o verso sinistra (se k<0) di k posizioni**

  1. 0110...0 $\longrightarrow$ 101.10...0
- **converto il numero binario frazionario ottenuto in decimale, ricordandomi di tener conto del bit di segno (1): 101.10...0 = -5.5**

220

Conversione da IEEE 754 a decimale
Esercizi

Rappresentare in decimale i seguenti numeri in formato IEEE 754:

- **0 10000001 010010...0**
- **1 10001101 011100...0**
- **0 01111101 010100...0**

221

Conversione da decimale a IEEE 754
Esercizi (soluzioni)

Rappresentare in decimale i seguenti numeri in formato IEEE 754:

- **0 10000001 010010...0** **[R. 5.125]**
- **1 10001101 011100...0** **[R. -23552.0]**
- **0 01111101 010100...0** **[R. 0.328125]**

222

© 2000 M. Rebaudengo, A. Benso