

Architettura degli Elaboratori

Rappresentazione
dell'Informazione



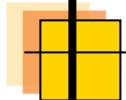
Barbara Masucci
UNIVERSITA DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Cosa studiamo oggi

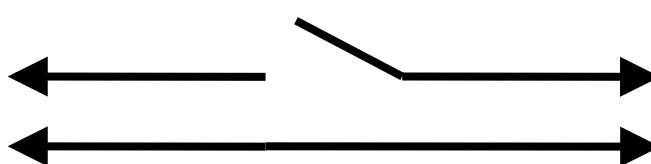
- Un moderno elaboratore è un sistema elettronico digitale **programmabile**
- Il suo comportamento è flessibile e specificato mediante un **programma**
 - Ossia un insieme di ordini, che lavorano su un insieme di dati
- Che lingua parla un elaboratore?
 - Come **codificare** e **memorizzare** opportunamente **dati** e **informazioni**?
 - Come **impartire** le giuste **istruzioni** per risolvere un problema?



Il linguaggio dei computer



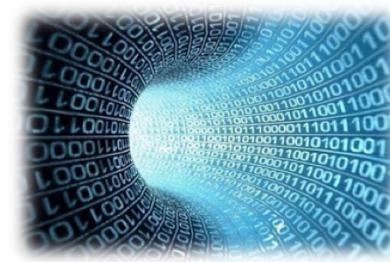
- I computer «parlano» in **binario**
- Alfabeto costituito da due soli simboli, 0 e 1, corrispondenti a
 - Spento (OFF)
 - Acceso (ON)
- Questi due simboli, detti **bit = binary digit** corrispondono ai due stati possibili di un circuito elettrico



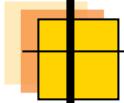
Rappresentazione Binaria



- Cosa possiamo rappresentare in binario?
 - Numeri (interi, col segno, con la virgola)
 - Parole
 - Istruzioni
 - Programma
- Idea fondamentale su cui sono costruiti i calcolatori:
 - Programmi e dati rappresentati da numeri
 - Oggi ci occupiamo della rappresentazione degli interi



Rappresentazione con n bit



Quanti interi possiamo rappresentare
con una sequenza n bit?

E quali sono?



Rappresentazione con n bit



n = 1

n = 2

n = 3

n = 4

0
1

0 0 =0
0 1 =1
1 0 =2
1 1 =3

0 00=0
0 01=1
0 10=2
0 11=3
1 00=4
1 01=5
1 10=6
1 11=7

0 000	=0
0 001	=1
0 010	=2
0 011	=3
0 100	=4
0 101	=5
0 110	=6
0 111	=7
1 000	=8
1 001	=9
1 010	=10
1 011	=11
1 100	=12
1 101	=13
1 110	=14
1 111	=15

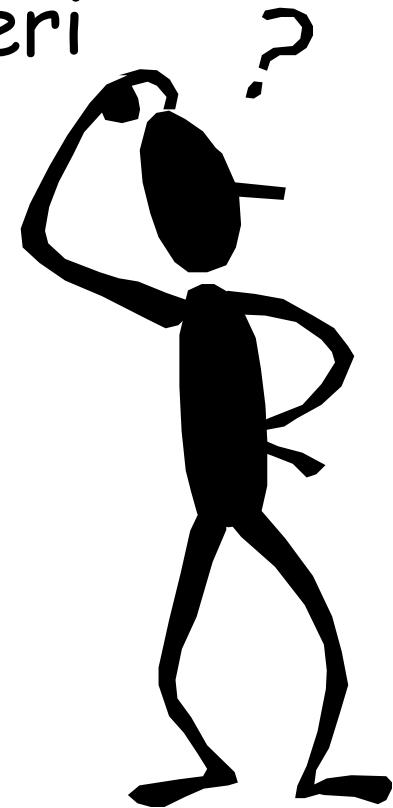


Rappresentazione con n bit



Con n bit è possibile rappresentare tutti i 2^n interi compresi fra 0 e $2^n - 1$

Perché?



Risposta

Se con k bit posso rappresentare p sequenze distinte,
con $k+1$ bit posso rappresentare $2p$ sequenze distinte

1 bit 2 sequenze distinte

2 bit $4=2\times 2$ sequenze distinte

3 bit $8=2\times 4$ sequenze distinte

4 bit $16=2\times 8$ sequenze distinte

.

.

.

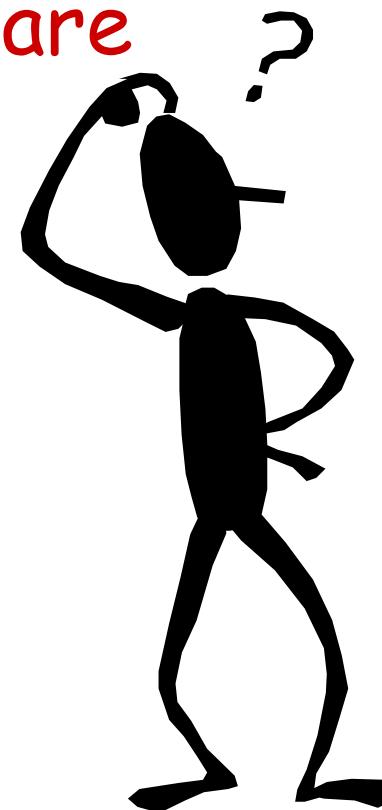
k bit $2^k = 2 \times 2^{k-1}$ sequenze distinte



Si formalizza con una prova per induzione

Rappresentazione Binaria

E' possibile prevedere il numero di bit
necessari per rappresentare
 N in binario?



Numero di bit per rappresentare N

- E' possibile rappresentare 152 con **7 bit**?
 - **No**, perché con 7 bit possiamo rappresentare tutti e soli gli interi da 0 a $2^7 - 1 = 127$
- E' possibile rappresentare 152 con **8 bit**?
 - **Si**, perché con 8 bit possiamo rappresentare tutti gli interi da 0 a $2^8 - 1 = 255$

In effetti: $2^7 - 1 < 152 < 2^8 - 1$

- Il minimo numero n di bit necessari a rappresentare N in binario è tale che $N < 2^n$, da cui

$$n > \log_2 N$$

Poiché n deve essere un intero, si considera il più piccolo intero successivo al valore ottenuto



Rappresentazione con n bit



	$n = 1$	$n = 2$	$n = 3$	$n = 4$
0	0	0 0 =0	0 00 =0	0 000 =0
1	1	0 1 =1 1 0 =2 1 1 =3	0 10 =2 0 11 =1 0 110 =4 0 111 =5 1 01 =6 1 10 =7 1 11 =8	1 01 =5 1 10 =6 1 11 =7 1 00 =8 1 011 =9 1 100 =10 1 101 =11 1 110 =12 1 111 =13 1 000 =14 1 111 =15

Perché $101 = 5$
e $1101 = 13$?



Notazione Decimale

- Anche detta **notazione araba**
- E' una **notazione posizionale**
 - Ogni cifra ha un **peso** diverso secondo la **posizione** che occupa
 - $7 \text{ } 5 \text{ } 2 = 7 \times 10^2 + 5 \times 10^1 + 2 \times 10^0$
 - 2 unità
 - 5 decine
 - 7 centinaia



Notazione Binaria

- Anche la notazione binaria è **posizionale**
 - Ogni cifra ha un **peso** diverso secondo la **posizione** che occupa
 - $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
= 5
 - $1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
= 13



Evitare ambiguità

- Per distinguere le due rappresentazioni, indichiamo la **base** (numero di simboli dell'alfabeto) come pedice del numero
 - Binario: base **2**
 - Decimale: base **10**
- Esempi
 - $(101)_2 = (5)_{10}$
 - $(1101)_2 = (13)_{10}$
 - $(1011)_2 = (11)_{10}$



Conversioni

Come convertire una sequenza
binaria in un intero decimale?



Da Binario a Decimale

$$N = (10110101)_2$$

$$\begin{aligned} N &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 2^7 + 2^5 + 2^4 + 2^2 + 2^0 \\ &= 128 + 32 + 16 + 4 + 1 \\ &= 181 \end{aligned}$$

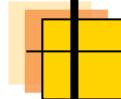


Conversioni

Come convertire un intero decimale
in una sequenza binaria?



Da Decimale a Binario



Dato $N = (181)_{10}$, cerco la più grande potenza di 2 contenuta in 181:

$$181 = 128 + 53 = 2^7 + 53$$

$$53 = 32 + 21 = 2^5 + 21$$

$$21 = 16 + 5 = 2^4 + 5$$

$$5 = 4+1 = 2^2+1$$

$$1 = 2^0$$

$$181 = 2^7 + 2^5 + 2^4 + 2^2 + 2^0$$

$$= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= (10110101)_2$$

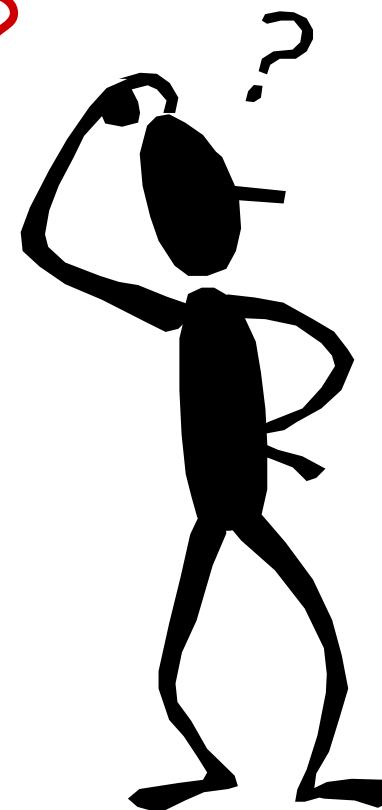
$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$



Nota: esiste un unico modo di esprimere un intero come somma di potenze distinte di 2

Notazione posizionale

La notazione posizionale è definita
solo per le basi 10 e 2?



Base $b=10$

La sequenza di **cifre decimali**

$$a_{n-1} a_{n-2} \dots a_1 a_0$$

rappresenta l'intero

$$\begin{aligned} N &= a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \dots + a_1 \times 10^1 + a_0 \times 10^0 \\ &= \sum_{i=0}^{n-1} a_i 10^i \end{aligned}$$



Base b=2



La sequenza binaria

$$a_{n-1} a_{n-2} \dots a_1 a_0$$

rappresenta l'intero

$$N = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_1 \times 2^1 + a_0 \times 2^0$$

$$= \sum_{i=0}^{n-1} a_i 2^i$$



Base b generica



La sequenza $a_{n-1} a_{n-2} \dots a_1 a_0$

di simboli in $\{0, 1, \dots, b-1\}$

rappresenta l'intero

$$N = a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} + \dots + a_1 \times b^1 + a_0 \times b^0$$

$$= \sum_{i=0}^{n-1} a_i b^i$$



Esempi

$$(234)_{10} \quad 2 \times 10^2 + 3 \times 10 + 4 = 234_{10}$$

$$(234)_8 \quad 2 \times 8^2 + 3 \times 8 + 4 = 156_{10}$$

$$(234)_{16} \quad 2 \times 16^2 + 3 \times 16 + 4 = 564_{10}$$

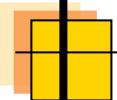
$$(101)_{10} \quad 1 \times 10^2 + 0 \times 10 + 1 = 101_{10}$$

$$(101)_8 \quad 1 \times 8^2 + 0 \times 8 + 1 = 65_{10}$$

$$(101)_2 \quad 1 \times 2^2 + 0 \times 2 + 1 = 5_{10}$$



Esercizio



- Da cosa sono caratterizzati i **numeri pari** in binario?
 - $(11010)_2$ è pari?
- Suggerimento: guardate l'ultimo bit (il meno significativo)
 - $(11010)_2$ è caratterizzato da...



Esercizio



Da cosa sono caratterizzati i numeri pari in binario?

0
1

0 0 =0
0 1 =1
1 0 =2
1 1 =3

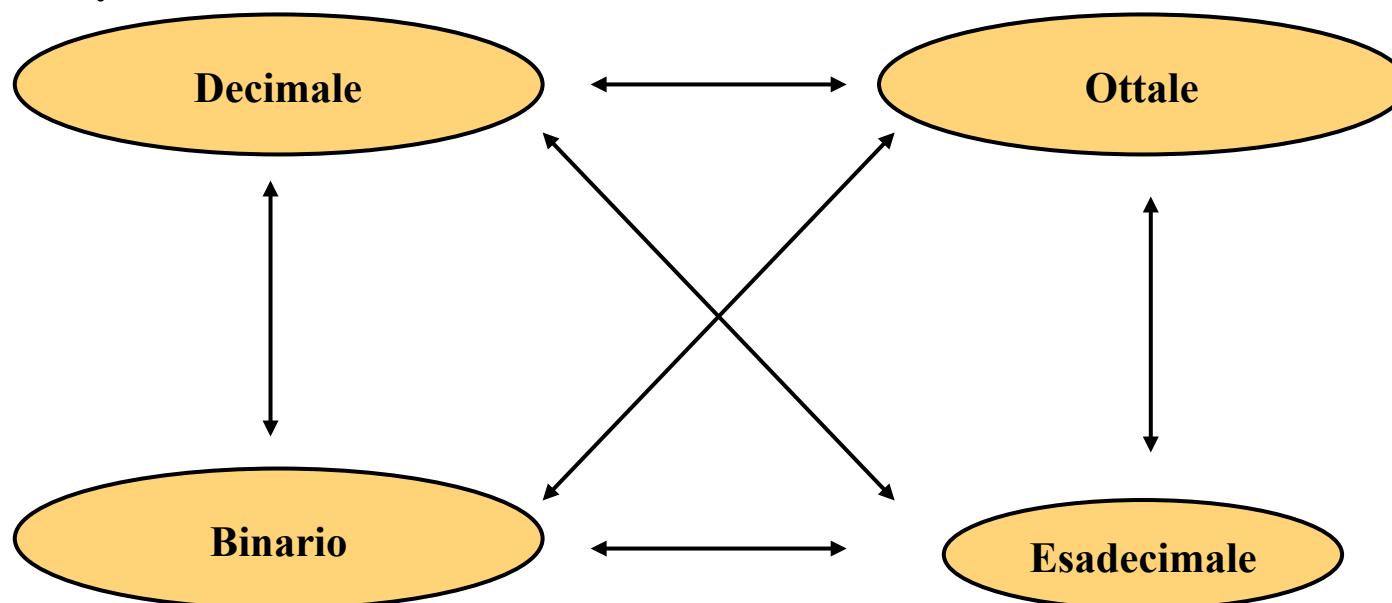
0 00=0
0 01=1
0 10=2
0 11=3
1 00=4
1 01=5
1 10=6
1 11=7

0 000 =0
0 001 =1
0 010 =2
0 011 =3
0 100 =4
0 101 =5
0 110 =6
0 111 =7
1 000 =8
1 001 =9
1 010 =10
1 011 =11
1 100 =12
1 101 =13
1 110 =14
1 111 =15



Conversioni tra basi (più diffuse)

- Le possibilità



Algoritmi di conversione

Binario \longleftrightarrow Decimale



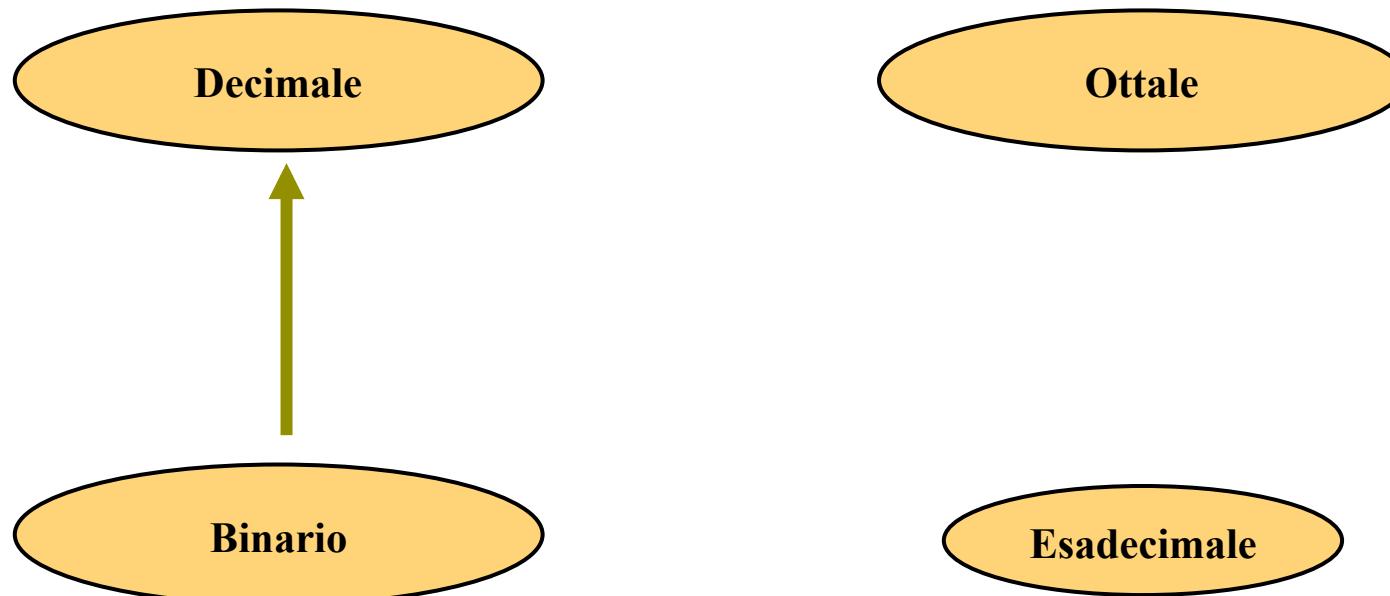
- **Da Binario a Decimale**
 - Moltiplico ogni cifra binaria per l'opportuna potenza di 2 e poi faccio la somma

- **Da Decimale a Binario**
 - Esprimo il numero come somma di potenze di 2, partendo dalla più grande potenza di 2 minore del numero

Esistono altri algoritmi, più efficienti, per convertire un numero da binario a decimale e viceversa



Da Binario a Decimale



Da Binario a Decimale

- Sia $N = (a_{n-1} a_{n-2} \dots a_1 a_0)_2$ un intero in binario
- Il valore di N è

$$N = a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \dots + a_1 2^1 + a_0$$

- Definiamo i valori

$$\begin{aligned}S_{n-1} &= a_{n-1} \\S_{n-2} &= a_{n-2} + 2S_{n-1} \\S_{n-3} &= a_{n-3} + 2S_{n-2}\end{aligned}$$

.....

$$S_i = a_i + 2S_{i+1}$$

.....

$$S_0 = N$$



Esempio 1

$(10110101)_2$

$n=8$

$$S_7 = a_7 = 1$$

$$S_6 = a_6 + 2S_7 = 0 + 2 = 2$$

$$S_5 = a_5 + 2S_6 = 1 + 4 = 5$$

$$S_4 = a_4 + 2S_5 = 1 + 10 = 11$$

$$S_3 = a_3 + 2S_4 = 0 + 22 = 22$$

$$S_2 = a_2 + 2S_3 = 1 + 44 = 45$$

$$S_1 = a_1 + 2S_2 = 0 + 90 = 90$$

$$S_0 = a_0 + 2S_1 = 1 + 180 = 181$$

$$S_0 = N = 181$$



Esempio 2

$(11101001)_2$

$n=8$

$$S_7 = a_7 = 1$$

$$S_6 = a_6 + 2S_7 = 1 + 2 = 3$$

$$S_5 = a_5 + 2S_6 = 1 + 6 = 7$$

$$S_4 = a_4 + 2S_5 = 0 + 14 = 14$$

$$S_3 = a_3 + 2S_4 = 1 + 28 = 29$$

$$S_2 = a_2 + 2S_3 = 0 + 58 = 58$$

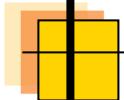
$$S_1 = a_1 + 2S_2 = 0 + 116 = 116$$

$$S_0 = a_0 + 2S_1 = 1 + 232 = 233$$

$$S_0 = N = 233$$



Esempio 3



$(11111111)_2$

$n=8$

$$S_7 = a_7 = 1$$

$$S_6 = a_6 + 2S_7 = 1 + 2 = 3$$

$$S_5 = a_5 + 2S_6 = 1 + 6 = 7$$

$$S_4 = a_4 + 2S_5 = 1 + 14 = 15$$

$$S_3 = a_3 + 2S_4 = 1 + 30 = 31$$

$$S_2 = a_2 + 2S_3 = 1 + 62 = 63$$

$$S_1 = a_1 + 2S_2 = 1 + 126 = 127$$

$$S_0 = a_0 + 2S_1 = 1 + 254 = 255$$

$$S_0 = N = 255$$



Perché funziona?



$$N = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_12 + a_0$$

$$= (a_{n-1}2^{n-2} + a_{n-2}2^{n-3} + \dots + a_22 + a_1)2 + a_0$$

$$= ((a_{n-1}2^{n-3} + a_{n-2}2^{n-4} + \dots + a_2)2 + a_1)2 + a_0$$

.....

.....

$$= (((\dots (a_{n-1}2 + a_{n-2})2 + \dots + a_2)2 + a_1)2 + a_0)$$

$$= (((\dots ((a_{n-1})2 + a_{n-2})2 + \dots + a_2)2 + a_1)2 + a_0)$$

$$S_0 = S_1 2 + a_0$$

$$S_1 = S_2 2 + a_1$$

$$S_2 = S_3 2 + a_2$$

.....

$$S_{n-2} = S_{n-1} 2 + a_{n-2}$$

$$S_{n-1} = a_{n-1}$$



Algoritmo di conversione: da Binario a Decimale

MSD=cifra **più** significativa

$$N = (a_{n-1} a_{n-2} \dots a_1 a_0)_2$$

LSD=cifra **meno** significativa

$$N = (((\dots ((a_{n-1}) 2 + a_{n-2}) 2 + \dots + a_2) 2 + a_1) 2 + a_0$$

$$N = S_0 = S_1 2 + a_0$$

$$S_1 = S_2 2 + a_1$$

$$S_2 = S_3 2 + a_2$$

.....

$$S_{n-2} = S_{n-1} 2 + a_{n-2}$$

$$S_{n-1} = a_{n-1}$$

Dal basso verso l'alto:
da a_{n-1}
a a_0 ;
da MSD
a LSD

$$S_{n-1} = a_{n-1}$$

$$S_{n-2} = a_{n-2} + 2S_{n-1}$$

$$S_{n-3} = a_{n-3} + 2S_{n-2}$$

.....

$$S_i = a_i + 2S_{i+1}$$

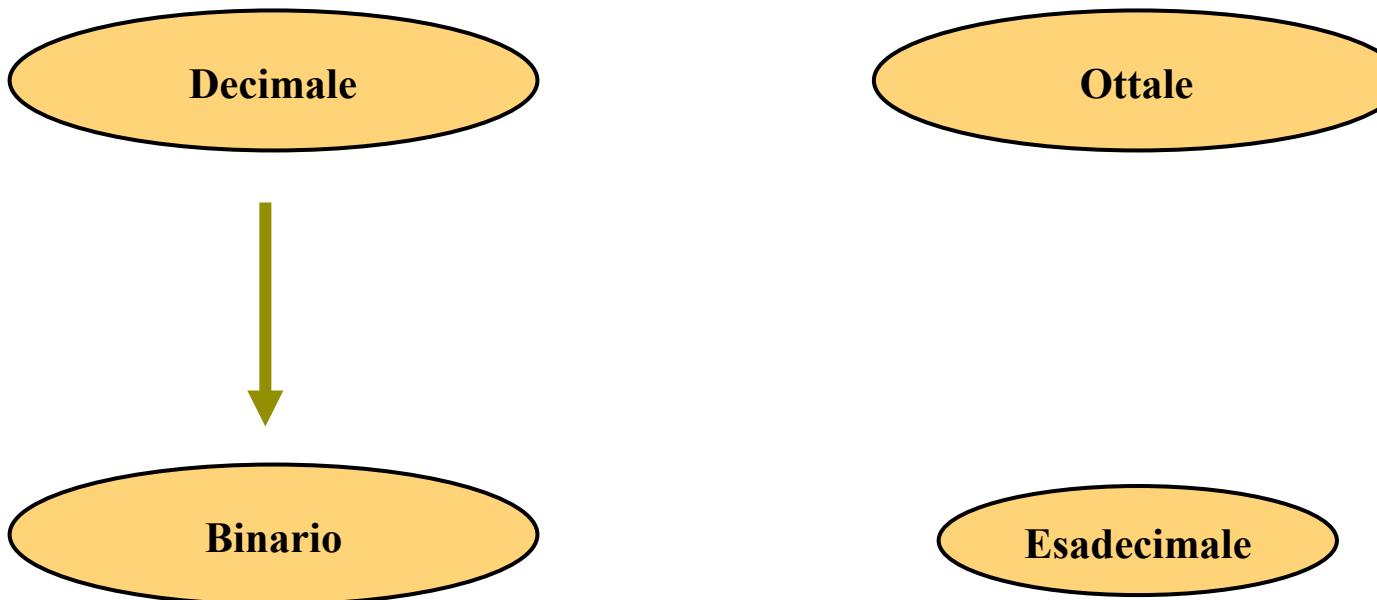
.....

$$S_0 = a_0 + 2S_1$$

$$S_0 = N$$



Da Decimale a Binario



Algoritmo di conversione: da Decimale a Binario

Dato: N Risultato: $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ con $a_i = 0$ o 1

Procedura inversa: dall'alto verso il basso (da a_0 ad a_{n-1})

$$N = S_0 = S_1 2 + a_0$$

$$S_1 = S_2 2 + a_1$$

$$S_2 = S_3 2 + a_2$$

.....

$$S_i = S_{i+1} 2 + a_i$$

.....

$$S_{n-2} = S_{n-1} 2 + a_{n-2}$$

$$S_{n-1} = a_{n-1}$$



a_0 ed S_1 sono il resto e il quoziente della divisione di $N = S_0$ per 2

....

a_i ed S_{i+1} sono il resto e il quoziente della divisione di S_i per 2

....

Fino ad ottenere un $S_i = 0$
Algoritmo delle divisioni successive

Esempio: N=152

$$N = (a_{n-1} a_{n-2} \dots a_1 a_0)_2 ???$$

$$S_0 = a_0 + 2S_1 = 0 + 2 \times 76 = 152$$

$152 : 2 = 76$ con resto 0

$$S_1 = a_1 + 2S_2 = 0 + 2 \times 38 = 76$$

$76 : 2 = 38$ con resto 0

$$S_2 = a_2 + 2S_3 = 0 + 2 \times 19 = 38$$

$38 : 2 = 19$ con resto 0

$$S_3 = a_3 + 2S_4 = 1 + 2 \times 9 = 19$$

$19 : 2 = 9$ con resto 1

$$S_4 = a_4 + 2S_5 = 1 + 2 \times 4 = 9$$

$9 : 2 = 4$ con resto 1

$$S_5 = a_5 + 2S_6 = 0 + 2 \times 2 = 4$$

$4 : 2 = 2$ con resto 0

$$S_6 = a_6 + 2S_7 = 0 + 2 \times 1 = 2$$

$2 : 2 = 1$ con resto 0

$$S_{n-1} = a_{n-1}$$

$$S_7 = a_7 + 2S_8 = 1 + 2 \times 0 = 1$$

$1 : 2 = 0$ con resto 1

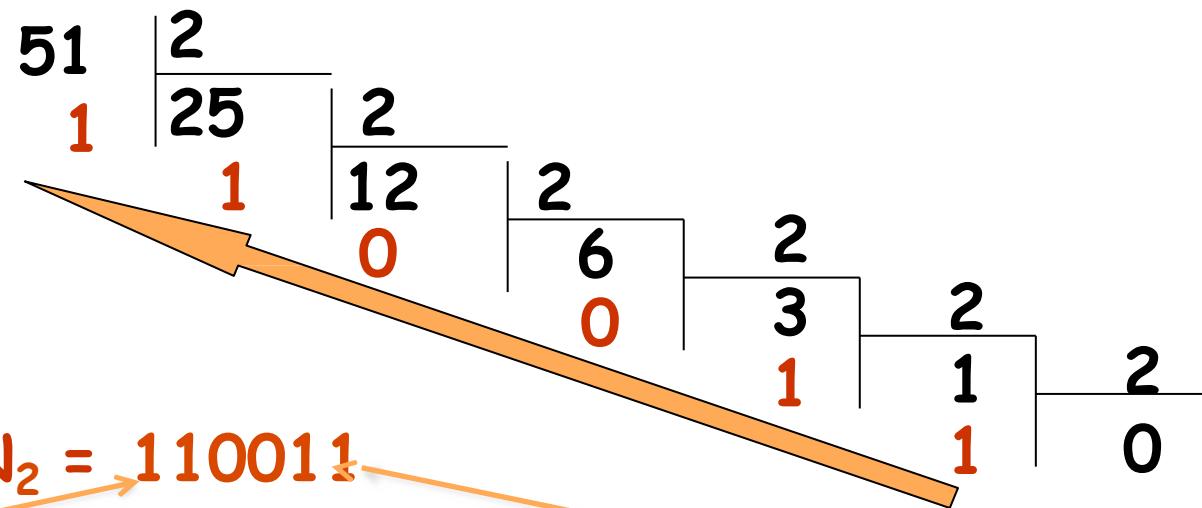
STOP

$$N = (10011000)_2$$



Esempio: N=51

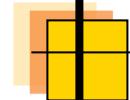
$$N = (a_{n-1} a_{n-2} \dots a_1 a_0)_2 ???$$



$$51 = 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0$$



Esercizi



- Convertire in base 2
 - 175_{10}
 - 98_{10}
- Scrivere in binario su 7 bit il numero 13_{10}
 - Suggerimento: effettuare la conversione e aggiungere i bit mancanti per arrivare a 7 bit
- Convertire in base 10
 - 10010101101_2
 - 0000000001_2



Indovinello: come conta ET?



- ET viene sulla Terra e ci dice che i re di Roma sono 13
- Quante dita ha ET?



Indovinello: come conta ET?



- ET viene sulla Terra e ci dice che i re di Roma sono 13
- Quante dita ha ET?
 - Il 13 deve essere interpretato come una stringa di simboli
 - Non conosciamo la base della numerazione
 - Sappiamo che il sistema di numerazione usato è POSIZIONALE
 - Sappiamo che i re di Roma sono 7_{10}



Riepilogo e riferimenti

- Rappresentazione binaria con n bit
 - Algoritmi di conversione per interi
 - bin->dec e dec->bin
- [P] parr. 1.1, 1.2, 1.3.1

