

# Rappresentazione digitale dei dati

## Codifica delle Informazioni

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Hardware: architettura dei computer

- In un computer possiamo distinguere quattro unità funzionali:
  - il processore
  - la memoria principale
  - la memoria secondaria
  - i dispositivi di input (inserimento)/output (restituzione di risultati)
- Il processore e la memoria principale costituiscono l'unità centrale del computer

L. De Panfilis - G. Manuppella

---

---

---

---

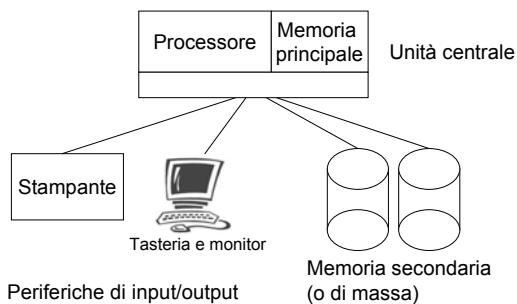
---

---

---

---

## Componenti principali di un computer



L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## ***L'uso dell'Informazione***

Un computer deve:

- memorizzare l'informazione
  - usando la memoria principale/secondaria
- elaborare l'informazione
  - usando il processore
- fare l'input/output dell'informazione
  - usando i dispositivi di input/output

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## ***Tipi di Informazione***

- Esistono vari tipi di informazione, di natura e forma diversa, così come rappresentazioni diverse della stessa informazione
  - La scelta della rappresentazione è in genere vincolata al tipo di utilizzo ed al tipo di operazioni che devono essere fatte sulle informazione stesse
- Il computer memorizza ed elabora informazioni che devono essere rappresentate in una forma gestibile
- Rappresentazione **digitale**

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## ***Rappresentazione digitale = rappresentazione binaria***

- L'entità minima di informazione che possiamo trovare all'interno di un elaboratore prende il nome di bit
  - binary digit – cifra binaria
  - un bit può assumere due valori
- Rappresentazione binaria
  - solo due simboli (0 e 1)

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Perché la rappresentazione binaria?

- Le informazioni rappresentate nel sistema binario possono essere elaborate secondo vari criteri e con vari strumenti
- I due simboli (0 e 1) possono essere rappresentati da:
  - Due stati di polarizzazione di una sostanza magnetizzabile
  - Due stati di carica elettrica di una sostanza
  - Al passaggio/non passaggio di corrente attraverso un cavo conduttore
  - Al passaggio/non passaggio di luce attraverso un cavo ottico

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dell'Informazione

- Per poter rappresentare un numero maggiore di informazioni si usano **sequenze** di bit
- Per esempio, per rappresentare quattro informazioni diverse possiamo utilizzare due bit che ci permettono di ottenere quattro configurazioni distinte

00 01 10 11

Il processo secondo cui si fa corrispondere ad un'informazione una sequenza di bit prende il nome **codifica dell'informazione**

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica binaria

- Esempio: un esame può avere quattro possibili esiti: ottimo, discreto, sufficiente, insufficiente
- Codifico
  - ottimo con 00
  - discreto con 01
  - sufficiente con 10
  - insufficiente con 11

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica binaria

- Con 2 bit si codificano 4 informazioni ( $2^2$ )
- Con 3 bit si codificano 8 informazioni ( $2^3$ )
- ...
- Con N bit si possono codificare  $2^N$  informazioni differenti

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica binaria

- Se il problema è quello di dover rappresentare M informazioni differenti si deve selezionare il numero di N bit in modo tale che

$$2^N \geq M$$

- Esempio: per rappresentare 40 informazioni differenti devo utilizzare 6 bit perché

$$2^6 = 64$$

– 5 bit non sono sufficienti perché  $2^5 = 32$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica binaria

- Esiste una particolare aggregazione di bit che è costituita da 8 bit ( $2^8 = 256$  informazioni) e prende il nome di **byte**
- Di solito si usano i multipli del byte

|      |    |                                              |
|------|----|----------------------------------------------|
| Kilo | KB | $2^{10}$ (~ un migliaio, 1024 byte)          |
| Mega | MB | $2^{20}$ (~ un milione, 1KB x 1024 byte)     |
| Giga | GB | $2^{30}$ (~ un miliardo, 1MB x 1024 byte)    |
| Tera | TB | $2^{40}$ (~ mille miliardi, 1GB x 1024 byte) |

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei caratteri

- Alfabeto anglosassone
    - Lettere maiuscole e minuscole
    - Cifre numeriche (0, 1, 2, ..., 9)
    - Simboli di punteggiatura (., ; : ! " ? ...)
    - Segni matematici (+, -, {, [, >, ...)
    - Caratteri nazionali (à, è, ì, ò, ù, ç, ñ, ö, ...)
- può essere codificato usando un byte (220 caratteri circa)
- Il metodo di codifica più diffuso tra i produttori di hardware e di software prende il nome ASCII (American Standard Code for Information Interchange)

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei caratteri

- Codice di riferimento:
  - ASCII: 8 bit per carattere
- Altri codici:
  - UNICODE, 16 bit per carattere (ASCII + caratteri etnici)
  - MSWindows usa un codice simile ad UNICODE

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei caratteri (ASCII)

| ASCII    | Simbolo             |
|----------|---------------------|
| 00000000 | NUL (spazio bianco) |
| ...      | ...                 |
| 00111110 | >                   |
| 00111111 | ?                   |
| 01000000 | @                   |
| 01000001 | A                   |
| 01000010 | B                   |
| 01000011 | C                   |
| ...      | ...                 |

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Tabella codice ascii standard

| HEX      | CHAR                  | DEC | CHAR | HEX      | CHAR | DEC | CHAR |
|----------|-----------------------|-----|------|----------|------|-----|------|
| 00000000 | Ctrl                  | 0   |      | 01000000 | A    | 65  | a    |
| 00000001 | 1 Start of heading    | 1   |      | 01000001 | B    | 66  | b    |
| 00000010 | 2 Start of text       | 2   |      | 01000010 | C    | 67  | c    |
| 00000011 | 3 End of text         | 3   |      | 01000011 | D    | 68  | d    |
| 00000100 | 4 End of transmit     | 4   |      | 01000100 | E    | 69  | e    |
| 00000101 | 5 Enquiry             | 5   |      | 01000101 | F    | 70  | f    |
| 00000110 | 6 Acknowledge         | 6   |      | 01000110 | G    | 71  | g    |
| 00000111 | 7 Audible bell        | 7   |      | 01000111 | H    | 72  | h    |
| 00001000 | 8 Backspace           | 8   |      | 01001000 | I    | 73  | i    |
| 00001001 | 9 Horizontal tab      | 9   |      | 01001001 | J    | 74  | j    |
| 00001010 | 10 Line feed          | 10  |      | 01001010 | K    | 75  | k    |
| 00001011 | 11 Vertical tab       | 11  |      | 01001011 | L    | 76  | l    |
| 00001100 | 12 Form Feed          | 12  |      | 01001100 | M    | 77  | m    |
| 00001101 | 13 Carriage return    | 13  |      | 01001101 | N    | 78  | n    |
| 00001110 | 14 Shift out          | 14  |      | 01001110 | O    | 79  | o    |
| 00001111 | 15 Shift in           | 15  |      | 01001111 | P    | 80  | p    |
| 00010000 | 16 Data link escape   | 16  |      | 01010000 | Q    | 81  | q    |
| 00010001 | 17 Device control 1   | 17  |      | 01010001 | R    | 82  | r    |
| 00010010 | 18 Device control 2   | 18  |      | 01010010 | S    | 83  | s    |
| 00010011 | 19 Device control 3   | 19  |      | 01010011 | T    | 84  | t    |
| 00010100 | 20 Device control 4   | 20  |      | 01010100 | U    | 85  | u    |
| 00010101 | 21 Neg. acknowledge   | 21  |      | 01010101 | V    | 86  | v    |
| 00010110 | 22 Synchronous idle   | 22  |      | 01010110 | W    | 87  | w    |
| 00010111 | 23 End trans. block   | 23  |      | 01010111 | X    | 88  | x    |
| 00011000 | 24 Cancel             | 24  |      | 01011000 | Y    | 89  | y    |
| 00011001 | 25 End of medium      | 25  |      | 01011001 | Z    | 90  | z    |
| 00011010 | 26 Substitution       | 26  |      | 01011010 | [    | 91  | [    |
| 00011011 | 27 Escape             | 27  |      | 01011011 | \    | 92  | \    |
| 00011100 | 28 File separator     | 28  |      | 01011100 | ]    | 93  | ]    |
| 00011101 | 29 Group separator    | 29  |      | 01011101 | ^    | 94  | ^    |
| 00011110 | 30 Directed separator | 30  |      | 01011110 | _    | 95  | _    |
| 00011111 | 31 Link separator     | 31  |      | 01011111 | Del  | 96  | Del  |

L. De Panfilis - G. Manuppella

## Come si inseriscono i caratteri ASCII da tastiera

- E' sufficiente tenere premuto il tasto ALT e contemporaneamente digitare sul tastierino numerico il codice decimale del carattere da inserire. Rilasciando il tasto ALT viene inserito il carattere desiderato.
- Es.: Se si deve inserire la tilde ~ bisogna tenere premuto il tasto ALT e digitare sul tastierino numerico il valore 126.

~ = ALT + 126

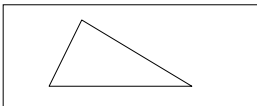
L. De Panfilis - G. Manuppella

## Codifica delle parole

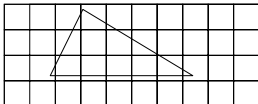
- Parole sono sequenze di caratteri
- Codifica della parole **cane**  
01100011 01100001 01101110 01100101  
c a n e
- Il problema inverso: data una sequenza di bit, il testo che essa codifica può essere ottenuto nel modo seguente:
  - si divide la sequenza in gruppi di otto bit (byte)
  - si determina il carattere corrispondente ad ogni byte

L. De Panfilis - G. Manuppella

## Codifica delle immagini



Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante



L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

---

---

## Codifica delle immagini

- Ogni quadratino derivante da tale suddivisione prende il nome di **pixel** (picture element) e può essere codificato in binario secondo la seguente convenzione:
  - Il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il bianco è predominante
  - Il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il nero è predominante

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

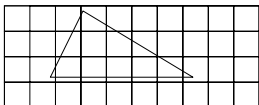
---

---

---

---

## Codifica delle immagini



|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

---

---

## Codifica delle immagini

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Poiché una sequenza di bit è lineare, è necessario definire convenzioni per ordinare la griglia dei pixel in una sequenza. Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra

0000000000 0011111000 0011100000 0001000000

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

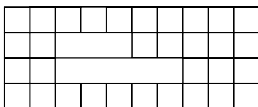
## Codifica delle immagini

Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria.

Se riconvertiamo la sequenza di stringhe

0000000000 0011111000 0011100000 0001000000

in immagine otteniamo



L. De Panfilis - G. Manuppella

---

---

---

---

---

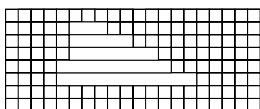
---

---

---

## Codifica delle immagini

La rappresentazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine



L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica delle immagini

- Assegnando un bit ad ogni pixel è possibile codificare solo immagini senza livelli di chiaroscuro
- Le immagini in bianco e nero hanno invece delle sfumature (diversi livelli di intensità di grigio)
- Per codificare le immagini con diversi livelli di grigio oppure a colori si usa la stessa tecnica
  - per ogni pixel viene assegnata una rappresentazione binaria con più di un bit

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica delle immagini (grigio e colore)

- Per memorizzare un pixel non è più sufficiente un solo bit
  - per esempio, se utilizziamo quattro bit possiamo rappresentare  $2^4 = 16$  livelli di grigio o 16 colori diversi
  - mentre con otto bit ne possiamo distinguere  $2^8 = 256$ , ecc.

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Risoluzione

- Il numero di pixel presenti sullo schermo (colonne x righe) prende il nome di **risoluzione**
- Risoluzione tipiche sono  
640 x 480   800 x 600   1024 x 768   1280 x 1024
- Esempio:
  - per distinguere 256 colori sono necessari otto bit per la codifica di ciascun pixel
  - la codifica di un'immagine formata da 640 x 480 pixel richiederà 2.457.600 bit (307.200 byte)

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## ***Grafica bitmap***

- Le immagini codificate pixel per pixel sono dette immagini in grafica **bitmap**
  - Le immagini bitmap occupano parecchio spazio
- Esistono delle tecniche di compressione che permettono di ridurre le dimensioni
  - Ad esempio, se più punti vicini di un'immagine assumono lo stesso colore, si può memorizzare la codifica del colore una sola volta e poi ricordare per quante volte deve essere ripetuta
- I formati come GIF e JPEG sono formati compressi

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## ***Grafica vettoriale***

- Crea le immagini manipolando linee e curve
  - La colorazione avviene attraverso la colorazione delle linee e delle aree chiuse
- Spesso occupano meno spazio rispetto alle immagini bitmap
- La grafica bitmap, invece, riesce a rendere con qualità maggiore immagini con un numero elevato di colori

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## ***Codifica delle immagini***

- Immagini complesse od irregolari: codifica bitmap (o raster)
- Immagini regolari: codifica vettoriale
- Codifiche ibride (raster/vettoriale)  
codifiche standard: Postscript, PDF

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica di immagini in movimento

- Un filmato è una sequenza di immagini statiche (dette fotogrammi o frame)
- Per codificare un filmato si digitalizzano i suoi fotogrammi
- Esempio:
  - 30 immagini ad alta risoluzione al secondo
  - $30 \text{ imm./sec} \times 2.457.600 \text{ bit/imm.} = 73.728.000 \text{ bit/sec}$
  - Un minuto richiederebbe  $60 \text{ sec} \times 73.728.000 = 4.423.680.000 \text{ bit}$  (5.529.600 byte)
- Sono necessarie delle tecniche per ottimizzare tale processo. Lo standard più diffuso: MPEG (molto efficiente)

L. De Panfilis - G. Manuppella

---

---

---

---

---

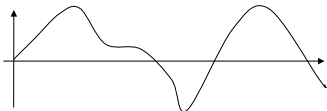
---

---

---

## Codifica dei suoni

- Fisicamente un suono è rappresentato come un'onda che descrive la variazione della pressione dell'aria nel tempo (onda sonora)



- Sull'asse delle ascisse viene rappresentato il tempo e sull'asse delle ordinate viene rappresentata la variazione di pressione corrispondente al suono stesso

L. De Panfilis - G. Manuppella

---

---

---

---

---

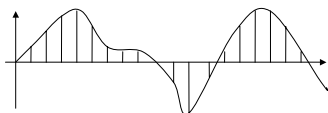
---

---

---

## Codifica dei suoni

- Si effettuano dei **campionamenti** sull'onda (cioè si misura il valore dell'onda a intervalli costanti di tempo) e si codificano in forma digitale le informazioni estratte



- Quanto più frequentemente il valore di intensità dell'onda viene campionato, tanto più precisa sarà la sua rappresentazione

L. De Panfilis - G. Manuppella

---

---

---

---

---

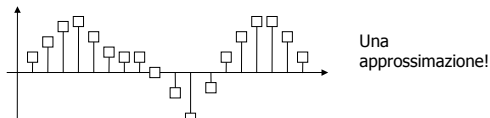
---

---

---

## Codifica dei suoni

- La sequenza dei valori numerici ottenuti dai campioni può essere facilmente codificata con sequenze di bit



- La rappresentazione è tanto più precisa quanto maggiore è il numero di bit utilizzati per codificare l'informazione estratta in fase di campionamento

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei suoni (esempio)

- Se volessimo codificare la musica di qualità CD dovremmo:
  - Usare due registrazioni corrispondenti a due microfoni distinti
  - Campionare il segnale musicale producendo 44100 campioni al secondo
  - Per ogni campione (che è un numero) si usano 16 bit
  - Per cui, il numero di bit che sarebbero necessari per codificare ogni secondo è pari a

$$2 \times 44.100 \text{ campioni} \times 16 \text{ bit/campione} = 1.414.200 \text{ bit}$$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei suoni

- Codifiche standard
  - WAV (MS-Windows)
  - MIDI
  - MP3
- MIDI
  - Codifica le note e gli strumenti che devono eseguirle
  - Efficiente, ma solo musica, non voce
- MP3
  - MPEG-3: variante MPEG per suoni
  - Grande diffusione, molto efficiente

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei numeri

- Il codice ASCII consente di codificare le cifre decimali da "0" a "9" fornendo in questo modo una rappresentazione dei numeri
- Per esempio: il numero 324 potrebbe essere rappresentato dalla sequenza di byte:

00110011 00110010 00110100  
3 2 4

- Ma questa rappresentazione non è efficiente e soprattutto non è adatta per eseguire le operazioni aritmetiche sui numeri

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei numeri (il sistema decimale)

- La rappresentazione dei numeri con il sistema decimale può essere utilizzata come spunto per definire un metodo di codifica dei numeri all'interno degli elaboratori
  - Esempio: la sequenza di cifre 324 viene interpretato come:
    - 3 centinaia + 2 decine + 4 unità
    - $324 = 3 \times 100 + 2 \times 10 + 4 \times 1$
    - $324 = 3 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei numeri (il sistema decimale)

- In generale la sequenza  $c_n c_{n-1} c_{n-2} \dots c_1 c_0$  (ogni " $c_i$ " è una cifra compresa tra "0" e "9") viene interpretata come:

La cifra meno significativa  $\rightarrow c_0 \times 10^0 + (c_0 \text{ unità})$   
 $c_1 \times 10^1 + (c_1 \text{ decine})$   
 $c_2 \times 10^2 + (c_2 \text{ centinaia})$   
 $\dots$   
 $c_{n-1} \times 10^{n-1} +$   
 La cifra più significativa  $\rightarrow c_n \times 10^n$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei numeri (il sistema binario)

- La numerazione decimale quindi utilizza una **notazione posizionale** basata sul numero 10 (numero delle cifre)
- La notazione posizionale può essere utilizzata in qualunque altro sistema di numerazione (con base diversa di 10)
- Nel sistema di numerazione binario i numeri vengono codificati utilizzando le due cifre "0" e "1"

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei numeri (il sistema binario)

- In analogia con il caso decimale la sequenza  $c_n c_{n-1} c_{n-2} \dots c_1 c_0$  (ogni " $c_i$ " è la cifra "0" o la cifra "1") rappresenterà il numero

$$\begin{array}{rcl}
 \text{La cifra meno} & \rightarrow & c_0 \times 2^0 \quad + \\
 \text{significativa} & & c_1 \times 2^1 \quad + \\
 & & c_2 \times 2^2 \quad + \\
 & & \dots \quad + \\
 & & c_{n-1} \times 2^{n-1} \quad + \\
 \text{La cifra più} & \rightarrow & c_n \times 2^n
 \end{array}$$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Codifica dei numeri (il sistema binario)

- Esempio: la sequenza "1011" rappresenta il numero

$$1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 11 \text{ (in base 10)}$$

- Per evitare ambiguità si usa la notazione

$$1011_2 = 11_{10}$$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Conversione dalla base 10 alla base 2

- Dato un numero  $N$  rappresentato in base dieci, la sua rappresentazione in base due sarà del tipo  $c_m c_{m-1} c_{m-2} \dots c_1 c_0$  (le " $c_i$ " sono cifre binarie)
- Per convertire un numero in base dieci nel corrispondente in base due si devono trovare i resti delle divisioni successive del numero  $N$  per due

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Conversione dalla base 10 alla base 2

- Esempio: il numero  $6_{10}$ :

$$6/2 = 3 \text{ resto } 0$$

$$3/2 = 1 \text{ resto } 1$$

$$1/2 = 0 \text{ resto } 1$$

- Leggendo i resti dal basso verso l'alto, si ha che la rappresentazione binaria del numero  $6_{10}$  è  $110_2$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Conversione dalla base 10 alla base 2

- Perché  $110_2 = 6_{10}$  ?

$$6/2 = 3 \text{ resto } 0 \quad 0 \times 2^0 +$$

$$3/2 = 1 \text{ resto } 1 \quad 1 \times 2^1 +$$

$$1/2 = 0 \text{ resto } 1 \quad \frac{1 \times 2^2}{6}$$

$$\frac{0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2}{2} = 1 \times 2^0 + 1 \times 2^1 \text{ con resto } 0$$

$$\frac{1 \times 2^0 + 1 \times 2^1}{2} = 1 \times 2^0 \text{ con resto } 1$$

$$\frac{1 \times 2^0}{2} = 0 \text{ con resto } 1$$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

## Conversione dalla base 10 alla base 2

- Esempio: il numero  $345_{10}$ :

$$\begin{array}{ll} 345/2 = 172 & \text{resto } 1 \\ 172/2 = 86 & \text{resto } 0 \\ 86/2 = 43 & \text{resto } 0 \\ 43/2 = 21 & \text{resto } 1 \\ 21/2 = 10 & \text{resto } 1 \\ 10/2 = 5 & \text{resto } 0 \\ 5/2 = 2 & \text{resto } 1 \\ 2/2 = 1 & \text{resto } 0 \\ 1/2 = 0 & \text{resto } 1 \end{array}$$

- Leggendo i resti dal basso verso l'alto, si ha che la rappresentazione binaria del numero  $345_{10}$  è  $101011001_2$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

---

---

## Conversione dalla base 2 alla base 10

- Sia  $c_m c_{m-1} c_{m-2} \dots c_1 c_0$  un numero rappresentato in base 2, usiamo:

$$c_0 \times 2^0 + c_1 \times 2^1 + c_2 \times 2^2 + \dots + c_{m-1} \times 2^{m-1} + c_m \times 2^m = N$$

- Esempio:  $101011001_2$

$$\begin{aligned} 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + \\ 1 \times 2^6 + 0 \times 2^7 + 1 \times 2^8 \\ = \\ 1 + 8 + 16 + 64 + 256 \\ = \\ 345 \end{aligned}$$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

---

---

## Altre basi: ottale, esadecimale

- Sistema ottale

- Utilizza una notazione posizionale basata su otto cifre (0,1,...,7) e sulle potenze di 8
- Esempio:  $103_8 = 1 \times 8^2 + 0 \times 8^1 + 3 \times 8^0 = 67$

- Sistema esadecimale

- Utilizza una notazione posizionale basata su sedici cifre (0,1,...,9,A,B,C,D,E,F) e sulle potenze di 16
- Esempio:  $103_{16} = 1 \times 16^2 + 0 \times 16^1 + 3 \times 16^0 = 259$
- Esempio:  $AC4_{16} = 10 \times 16^2 + 12 \times 16^1 + 4 \times 16^0 = 2756$

L. De Panfilis - G. Manuppella

---

---

---

---

---

---

---

---

---

---