

INTRODUZIONE AI CALCOLATORI ELETTRONICI

**CORSO DI LAUREA IN ECONOMIA AZIENDALE
ING. PAOLA LAPADULA - UNIVERSITÀ DEGLI STUDI
DELLA BASILICATA**

A.A. 2023/2024

SOMMARIO

- Elementi di Base dell'ICT
 - Rappresentazione delle informazioni
 - Rappresentazione dei caratteri
 - Rappresentazione dei numeri
 - Rappresentazione di immagini e video

RAPPRESENTAZIONE DELLE INFORMAZIONI

- In un calcolatore, le informazioni sono codificate in forma binaria – sistema binario
 - ossia in una sequenza finita di 0 e di 1
- La più piccola unità di informazione memorizzabile o elaborabile da un calcolatore è il bit (binary digit) → 8 bit = 1 Byte
- L'alfabeto con cui un calcolatore **codifica** ogni informazione consiste di due soli simboli 0 e 1

RAPPRESENTAZIONE DELLE INFORMAZIONI

- Oltre al sistema binario, in base 2, esistono altri sistemi
 - Sistema esadecimale è un sistema numerico posizionale in base 16
 - utilizza 16 simboli invece dei 10 del sistema numerico tradizionale
 - si usano simboli da 0 a 9 per le prime dieci cifre, e poi le lettere da A a F per le successive sei cifre, per un totale di 16 simboli

RAPPRESENTAZIONE DELLE INFORMAZIONI

- Manipolare bit significa manipolare valori vero/falso
 - bit 0: valore falso
 - bit 1: valore vero
- Per mettere in relazione le istruzioni contenute nei linguaggi di programmazione, si utilizzano i **connettivi logici** o **operatori booleani**

GLI OPERATORI BOOLEANI

- Le Operazioni booleane (mat. George Boole 1815-1864) e operatori combinano due valori di ingresso (input) per produrre un terzo valore in uscita (output)
- Ciascun operatore prende in input uno o due booleani e restituisce in output un altro booleano

GLI OPERATORI BOOLEANI

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

- Gli operatori più importanti sono: AND, OR e NOT
- **AND**, prende in input due operandi e produce in output un booleano, attenendosi al seguente comportamento:
 - Se entrambi gli operandi sono true allora l'output è true; in tutti gli altri casi l'output è uguale a false

GLI OPERATORI BOOLEANI

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

- (cont.) **OR**, prende in input due operandi e produce in output un booleano, attenendosi al seguente comportamento
 - Se almeno uno degli operandi è uguale a true, l'output è true;
 - altrimenti, se nessuno dei due operandi è uguale a true l'output sarà false

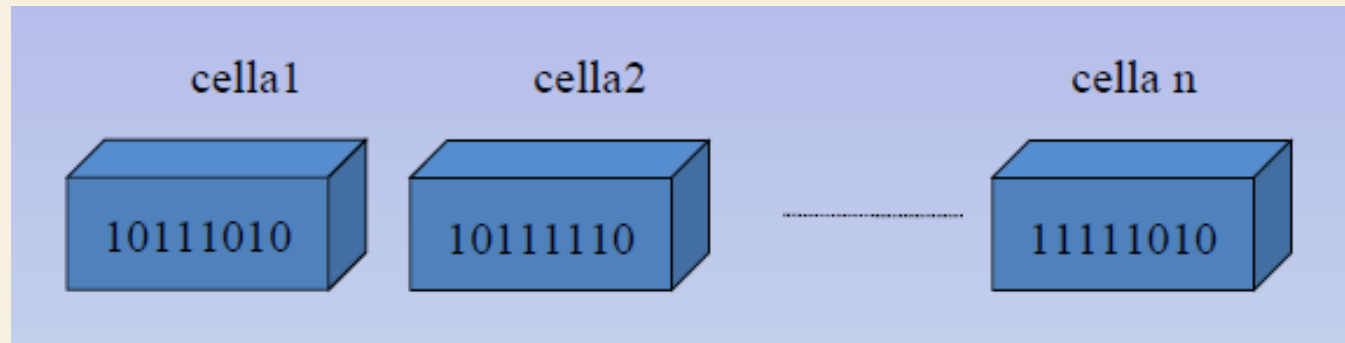
A	Not A
0	1
1	0

GLI OPERATORI BOOLEANI

- **NOT**, prende in input un solo operando e produce in output un booleano, attenendosi al seguente comportamento:
 - Se l'operando di input è true allora l'output sarà false
 - Se, invece l'operando di input è false, allora l'output sarà uguale a true
 - In altri termini, l'operatore di NOT prende un input e ne restituisce l'esatto contrario

MEMORIZZAZIONE E RAPPRESENTAZIONE DEI DATI

- Per archiviare i dati, un computer contiene un gran numero di circuiti flip-flop,
- ognuno in grado di memorizzare un singolo bit
- I circuiti di memoria sono organizzati in unità chiamate celle di memoria



MEMORIZZAZIONE E RAPPRESENTAZIONE DEI DATI

- Per identificare le singole celle di memoria ad ognuna di esse è assegnata una denominazione univoca chiamata **indirizzo**
- Poiché la memoria principale di un computer è organizzata sotto forma di singole celle dotate di indirizzo, è possibile accedere a tali celle a secondo della necessità

CODIFICA DELLE INFORMAZIONI

- Analizziamo il metodo con cui è possibile codificare le informazioni tramite bit: testi, immagini, suoni
- Le informazioni per poter essere interpretate dal calcolatore devono essere “trasformate” in sequenze di bit
 - La “trasformazione” viene detta CODIFICA
- Codificare: trasformare un messaggio di un alfabeto in un messaggio di un altro alfabeto mediante l'applicazione di opportune regole dette regole di codifica

CODIFICA DELLE INFORMAZIONI

- Esistono diverse convenzioni per codificare i diversi tipi di dati
- Nello specifico... processo secondo cui si fa corrispondere ad un'informazione una configurazione di cifre binarie
- Con un certo numero di bit possiamo codificare un certo numero di informazioni

CODIFICA DELLE INFORMAZIONI

- Con una sequenza di n bit si possono rappresentare 2^n informazioni diverse
 - $n = 1 \rightarrow 0, 1$
 - $n = 2 \rightarrow 00, 01, 10, 11$
 - $n = 3 \rightarrow 000, 001, 010, 011, 100, 101, 110, 111$
- Con 8 bit si possono rappresentare 256 informazioni:
 - $2^8 = 256$
 - Si usano tutte le combinazioni possibili tra 00000000 e 11111111

UNITÀ DI MISURA

- BIT (binary digit)
- BYTE = 8 bit
- 1 Kb (kilobit) 1024 bit
- 1 KB (kilobyte) 1024 byte (2^{10} byte)
- 1 Mb (megabit) 1024 Kb
- 1 MB (megabyte) 1024 KB (2^{20} byte)
- 1 GB (gigabyte) 1024 MB
- 1 TB (terabyte) 1024 GB
- 1 PB (petabyte) 1024 TB

RAPPRESENTAZIONE DEI CARATTERI

– Codice ASCII (American Standard Code for Information Interchange)

- codice di traduzione per i simboli alfanumerici
- ad ogni carattere è associato un numero
- la rappresentazione del carattere coincide con quella del numero
- vale anche per le cifre (0-9) ed i caratteri speciali (es: *, #, \$)

RAPPRESENTAZIONE DEI

CAR

Tabella dei Codici ASCII Estesi

000	NUL	033	!	066	B	099	c	132	ä	165	ñ	198	ã	231	þ
001	Start Of Header	034	"	067	C	100	d	133	å	166	ª	199	Ä	232	ß
002	Start Of Text	035	#	068	D	101	e	134	ä	167	º	200	ℒ	233	Û
003	End Of Text	036	\$	069	E	102	f	135	ç	168	¿	201	℞	234	Ü
004	End Of Transmission	037	%	070	F	103	g	136	è	169	®	202	ℓ	235	Ù
005	Enquiry	038	&	071	G	104	h	137	é	170	¬	203	℥	236	Ý
006	Acknowledge	039		072	H	105	i	138	è	171	½	204	℥	237	Ÿ
007	Bell	040	(073	I	106	j	139	ï	172	¼	205	=	238	–
008	Backspace	041)	074	J	107	k	140	î	173	½	206	℥	239	˙
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	¼	207	℥	240	-
010	Line Feed	043	+	076	L	109	m	142	Á	175	»	208	ð	241	±
011	Vertical Tab	044	,	077	M	110	n	143	À	176	∴	209	Ð	242	_
012	Form Feed	045	-	078	N	111	o	144	É	177	∴	210	È	243	¼
013	Carriage Return	046	.	079	O	112	p	145	æ	178	⊞	211	Ê	244	¶
014	Shift Out	047	/	080	P	113	q	146	Æ	179		212	Ë	245	§
015	Shift In	048	0	081	Q	114	r	147	ø	180	†	213	Ì	246	÷
016	Delete	049	1	082	R	115	s	148	ö	181	À	214	Í	247	˘
017	-- frei --	050	2	083	S	116	t	149	ò	182	Á	215	Î	248	°
018	-- frei --	051	3	084	T	117	u	150	û	183	À	216	Ï	249	˙
019	-- frei --	052	4	085	U	118	v	151	ù	184	©	217	Ĵ	250	.
020	-- frei --	053	5	086	V	119	w	152	ÿ	185	¶	218	Œ	251	ˆ
021	Negative Acknowledge	054	6	087	W	120	x	153	Ö	186		219	█	252	˚
022	Synchronous Idle	055	7	088	X	121	y	154	Ü	187	¶	220	■	253	˚
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	¶	221	¡	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	φ	222	í	255	
025	End Of Medium	058	:	091	[124		157	∅	190	¥	223	■		
026	Substitute	059	;	092	\	125	}	158	×	191	γ	224	Ó		
027	Escape	060	<	093]	126	~	159	f	192	ℒ	225	Ô		
028	File Separator	061	=	094	^	127	▯	160	á	193	⊥	226	Ö		
029	Group Separator	062	>	095	_	128	Ç	161	í	194	⊥	227	Ø		
030	Record Separator	063	?	096	`	129	ü	162	ó	195	⊥	228	ð		
031	Unit Separator	064	@	097	a	130	é	163	ú	196	–	229	Õ		
032		065	A	098	b	131	â	164	ñ	197	+	230	μ		

RAPPRESENTAZIONE DEI CARATTERI

- Il codice ASCII a 7 bit è limitato:
 - non include lettere accentate
- Codice ASCII a 8 bit (“ASCII esteso”):
 - al carattere è associato un num. tra 0 e 255 maggior numero di caratteri rappresentabili
- Al giorno d’oggi codice UNICODE (16 bit o superiore)

RAPPRESENTAZIONE DEI CARATTERI

- **Unicode** è un sistema per rappresentare i caratteri di tutti i differenti linguaggi del mondo
- Il codice assegnato al carattere viene rappresentato con U+, seguito dalle quattro (o sei) cifre esadecimali del numero che lo individua
- Es: Il carattere **a** ha codice Unicode **U+0061**

RAPPRESENTAZIONE DEI CARATTERI

Ricapitolando

- Codice ASCII (American Standard Code for Information Interchange)
 - ad ogni carattere è associato un numero
 - Codice ASCII a 7 bit
 - Codice ASCII a 8 bit (“ASCII esteso”)
- Unicode a 16 bit o superiore
 - viene supportato dai moderni standard della programmazione e del markup come XML, Java, JavaScript, LDAP, CORBA 3.0, e da vari sistemi operativi

I FILE

- Il file è un contenitore di informazione digitalizzata
- Le informazioni codificate al suo interno sono leggibili solo da software
- I dati codificati in un file sono organizzati come una sequenza di byte, immagazzinati come un solo elemento su una memoria di massa

I FILE

- Immaginate un file come un foglietto a quadretti nel quale scrivete i codici corrispondenti alle lettere che voi digitate dalla tastiera:
- Scrivo

0	1	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	1
0	1	0	0	1	1	1	1																

- Ad esempio per scrivere C scrive 01000011, per scrivere A scrive 01000001 ...

I FILE

- Ogni programma salva il file in un determinato formato identificato univocamente dall'estensione del programma
- Il nome del file è del tipo:
 - nome . estensione
 - Estensione e icona identificano il programma
 - Es: foto.gif, documento.doc, calcolo.xls

I FILE

- Alcune estensioni molto note:
 - ZIP/TAR -> Winzip
 - XLSX -> Excel
 - PPT -> Power Point -> Presentazioni multimediali
 - MDB -> Access -> DataBase
 - EXE -> File eseguibili
 - MP3 -> Winamp / Windows Media Player -> File musicale
 - HTML -> Browser -> File Internet Pagina web

RAPPRESENTAZIONE DEI NUMERI

- Numeri rappresentabili all'interno di un calcolatore sono di due tipi

Numeri Interi:

- utilizzando il sistema di numerazione binario,
- qualsiasi numero intero può essere rappresentato tramite una sequenza di bit di opportuna lunghezza

RAPPRESENTAZIONE DEI NUMERI

- Numeri rappresentabili all'interno di un calcolatore sono di due tipi (cont.)

Numeri Con la Virgola:

- Scomponendo il numero in parte intera e parte frazionaria
- e sempre utilizzando il sistema di numerazione binario,
- è possibile ottenere una rappresentazione anche per questo tipo di numeri

CONVERSIONI

- Per codificare i numeri esiste un sistema di numerazione simile a quello tradizionale (sistema decimale e posizionale)
 - Ogni cifra ha un valore che dipende dalla posizione che occupa all'interno del numero
 - Per esempio

2 1 0 Posizione

645

$$6 * (10^2) + 4 * (10^1) + 5 * (10^0) = 600 + 40 + 5 = 645$$

CONVERSIONI

- Conversione dal sistema binario al sistema decimale
 - Per esempio

3 2 1 0 Posizione

1 1 0 1

$$1 * (2^3) + 1 * (2^2) + 0 * (2^1) + 1 * (2^0) = 8+4+1 = 13_{10}$$

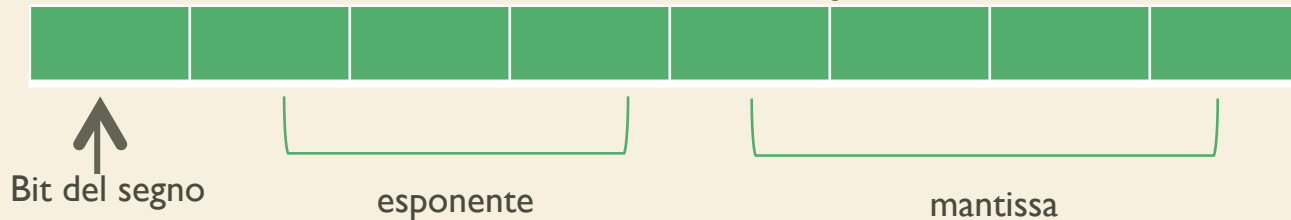
CONVERSIONI

- Per ottenere il valore binario di un numero intero si procede utilizzando successive divisioni per 2
 - Per esempio

$$\begin{array}{r} 26_{10} \\ 26:2 = 0 \\ 13:2 = 1 \\ 6:2 = 0 \\ 3:2 = 1 \\ 1:2 = 1 \end{array} \begin{array}{c} \uparrow \\ \text{Resto della} \\ \text{divisione per 2} \end{array} 11010_2$$

CONVERSIONI PER I NUMERI REALI

- Codifica in virgola mobile



Nel registro saranno prefissate zone diverse per la mantissa e per l'esponente

- Fissata la base B (es. 10 o 2)
- il numero viene rappresentato attraverso due altri numeri **mantissa** ed **esponente**
 - $N = m B^e$, con $B^{i-1} \leq m < B^i$ (i dipende dal numero di cifre da rappresentare)
 - es: $23.315 = 0.23315 \times 10^2$

$$10110.010 = 10.110010 \times 2^3$$

CONVERSIONI PER I NUMERI REALI

- Lo Standard per la presentazione di numeri reali in singola precisione
 - è utilizzato, ad esempio, per rappresentare il tipo `float` del C++ che occupa 4 byte (32 bit): 1 bit per il segno, 8 bit per l'esponente e 23 bit per la mantissa

CONVERSIONI

- Trasformare i seguenti numeri dal sistema decimale a quello binario:
 - 67
 - 124
 - 65
 - 79
- Verificare la conversione ritrasformando le stringhe di byte ottenute nuovamente nel sistema decimale