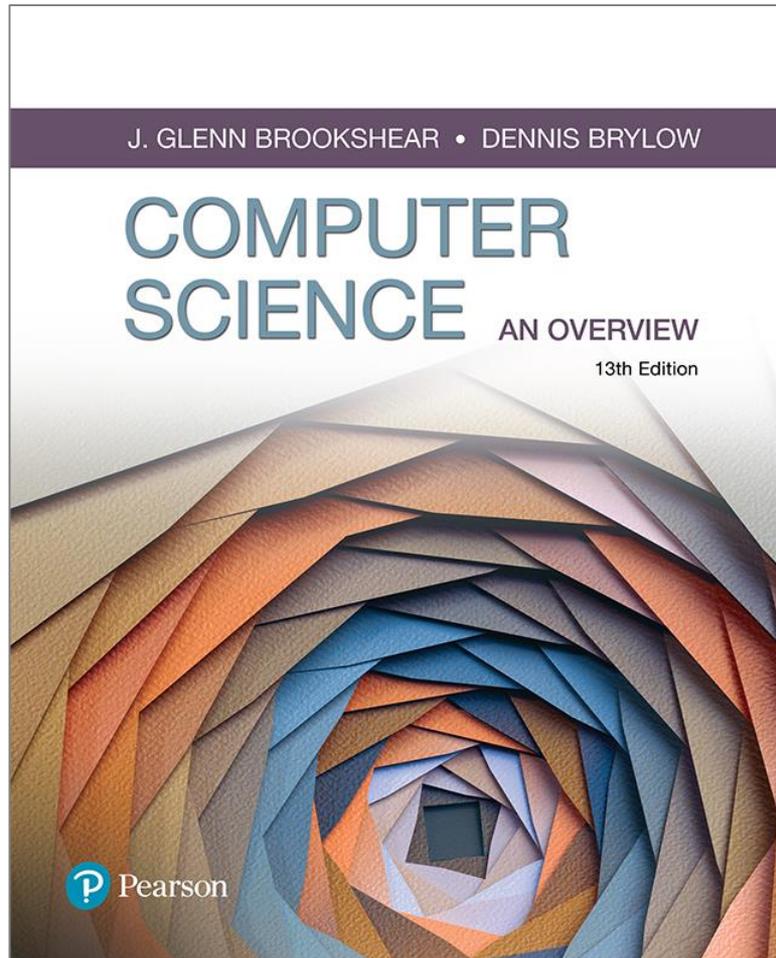


Computer Science An Overview

13th Edition



Chapter 1 Data Storage

Bits

- **Bit:** Binary Digit (0 o 1)
- Bit Patterns sono usati per rappresentare l'informazione
 - Numeri
 - Caratteri
 - Immagini
 - Suoni

Operazioni booleane

- **Operazioni booleane:** un'operazione che manipola uno o più valori V/F
- Operazioni specifiche
 - AND
 - OR
 - XOR (exclusive or)
 - NOT

Notazione esadecimale

- **Esadecimale: 0,1,....,9,A,B,....,F**
- Notazione più corta: 8 cifre decimali → 2 cifre esadecimali
 - Sequenza divisa in gruppi di 4 bit
 - Ogni gruppo è rappresentato da una cifra esadecimale
- Ex.: 10110101 diventa 0xB5

1011 0101

1011=B e 0101=5

Tabella di conversione

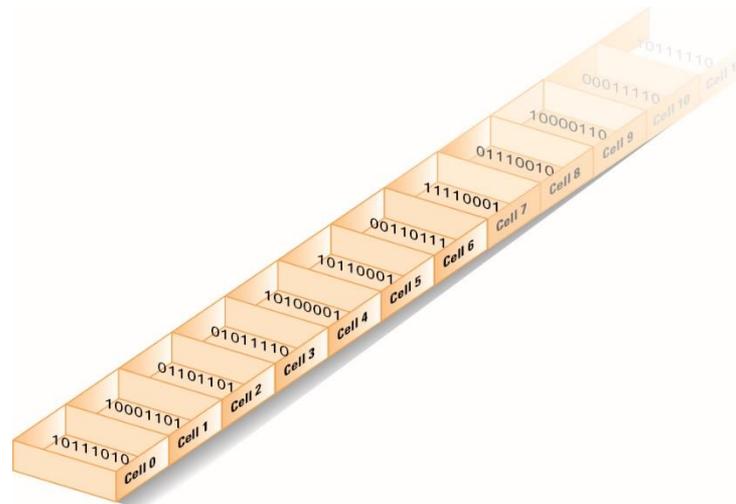
Bit pattern	Hexadecimal representation
0000	0x0
0001	0x1
0010	0x2
0011	0x3
0100	0x4
0101	0x5
0110	0x6
0111	0x7
1000	0x8
1001	0x9
1010	0xA
1011	0xB
1100	0xC
1101	0xD
1110	0xE
1111	0xF

Memoria principale

- **Cella:** una unità di memoria principale (tipicamente 8 bits = 1 **byte**)
 - **Most significant bit**
 - **Least significant bit**

Indirizzo di memoria

- **Address:** identifica univocamente una cella nella memoria
 - Numeri assegnati consecutivamente



Terminologia per le memorie

- **Random Access Memory (RAM):** detta anche ad accesso diretto
- **Dynamic Memory (DRAM):** RAM composta da memoria volatile

Capacità di memoria

- **Kilobyte:** 2^{10} bytes = 1024 bytes
 - Example: 3 KB = 3 times 1024 bytes
- **Megabyte:** 2^{20} bytes = 1,048,576 bytes
 - Example: 3 MB = 3 times 1,048,576 bytes
- **Gigabyte:** 2^{30} bytes = 1,073,741,824 bytes
 - Example: 3 GB = 3 times 1,073,741,824 bytes

Perchè 1byte=8bit?

Rappresentazione del testo

- **Problema della codifica dei caratteri della tastiera**
 - **ASCII**: Usa sequenze di 7-bits per rappresentare la maggiorparte dei simboli usati nella lingua inglese
7 bit → 128 simboli
 - **ASCII esteso** 8 bit → 256 simboli
 - **Unicode**: fino a 21-bits; 16-bits per le lingue più comuni

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

Codifica ASCII del messaggio “Hello.”

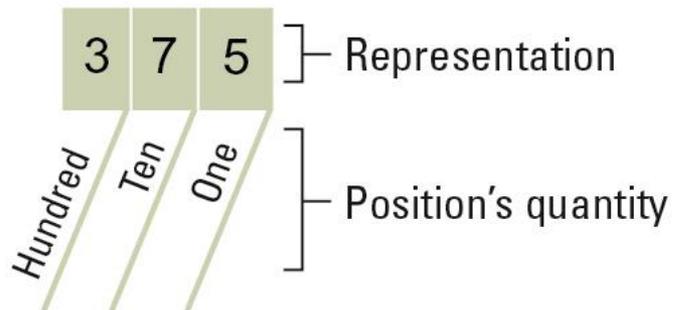
01001000	01100101	01101100	01101100	01101111	00101110
H	e	l	l	o	.

Notazione binaria

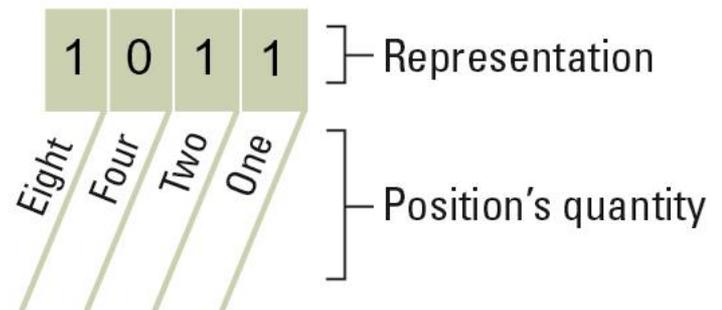
- **Notazione binaria:** usa i bit per rappresentare un numero in base due
 - Nota che $10^3=1000$ (decimale)
 - $2^9=512$ e $2^{10}=1024$
(in base 2 non si arriva mai esattamente a mille)

Base 10 e base 2

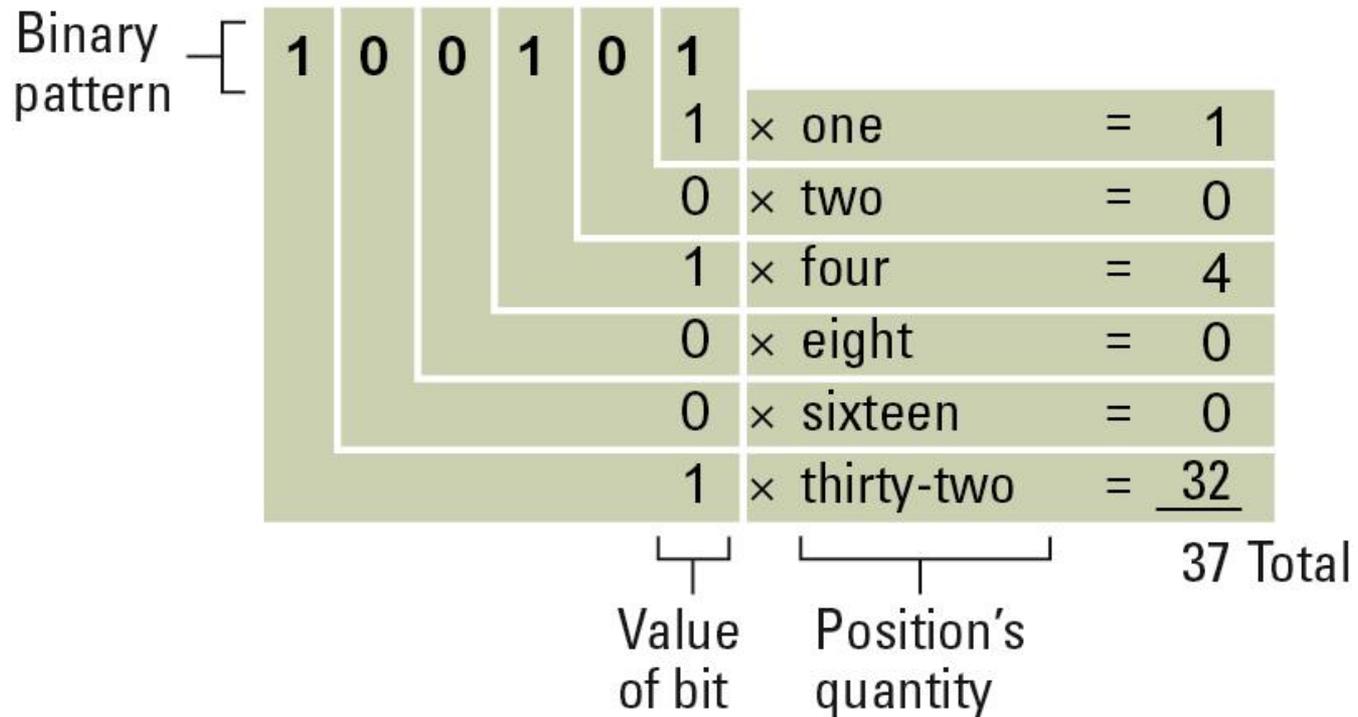
a. Base 10 system



b. Base two system



Decodifica di 100101



Overflow

- Esiste un limite alla dimensione dei valori che possono essere rappresentati in qualunque sistema
- Overflow
 - Avviene quando una computazione produce un valore che va oltre il range dei valori rappresentabili dalla macchina



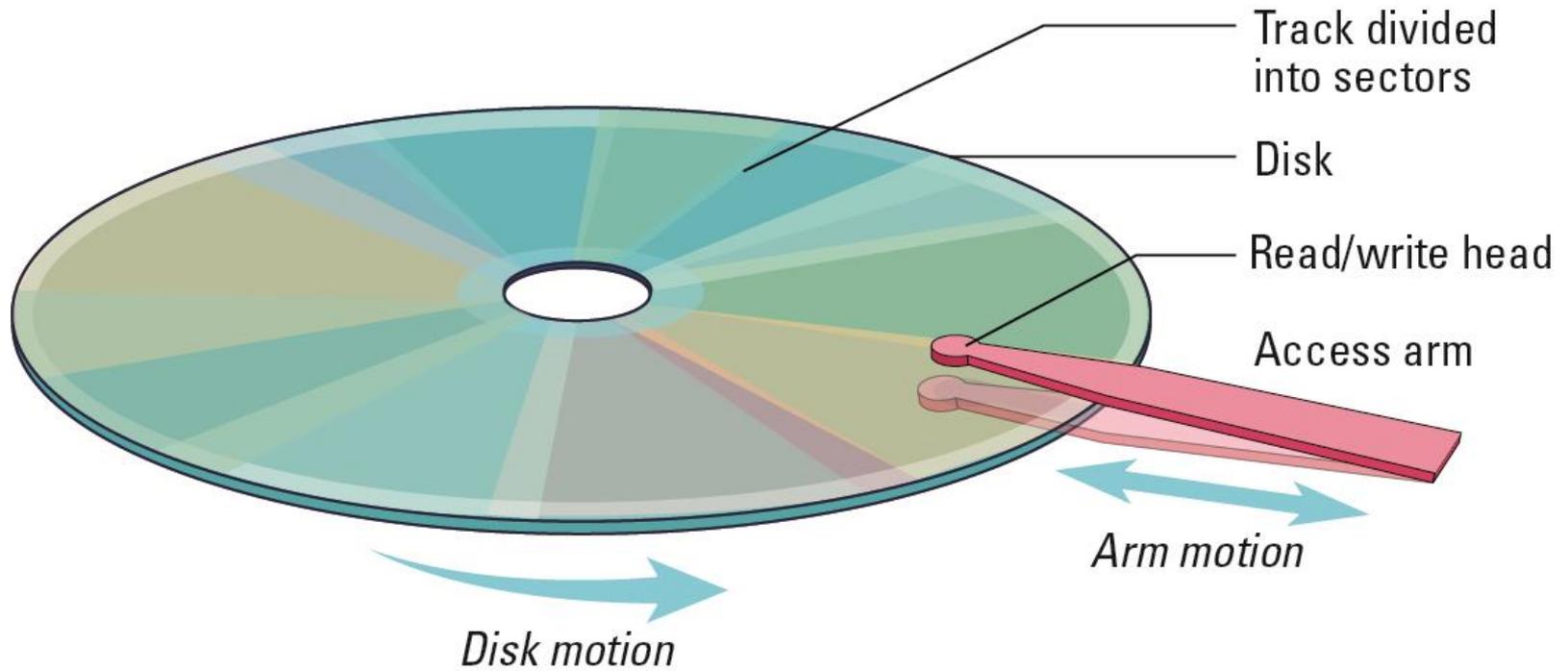
Memoria di massa

- Dispositivi aggiuntivi:
 - Magnetic disks
 - CDs
 - DVDs
 - Nastri magnetici
 - Flash drives
 - Solid-state drives
- Vantaggi rispetto alla memoria principale
 - Minore volatilità
 - Grandi capacità di memorizzazione
 - Basso costo
 - In molti casi possono essere rimosse

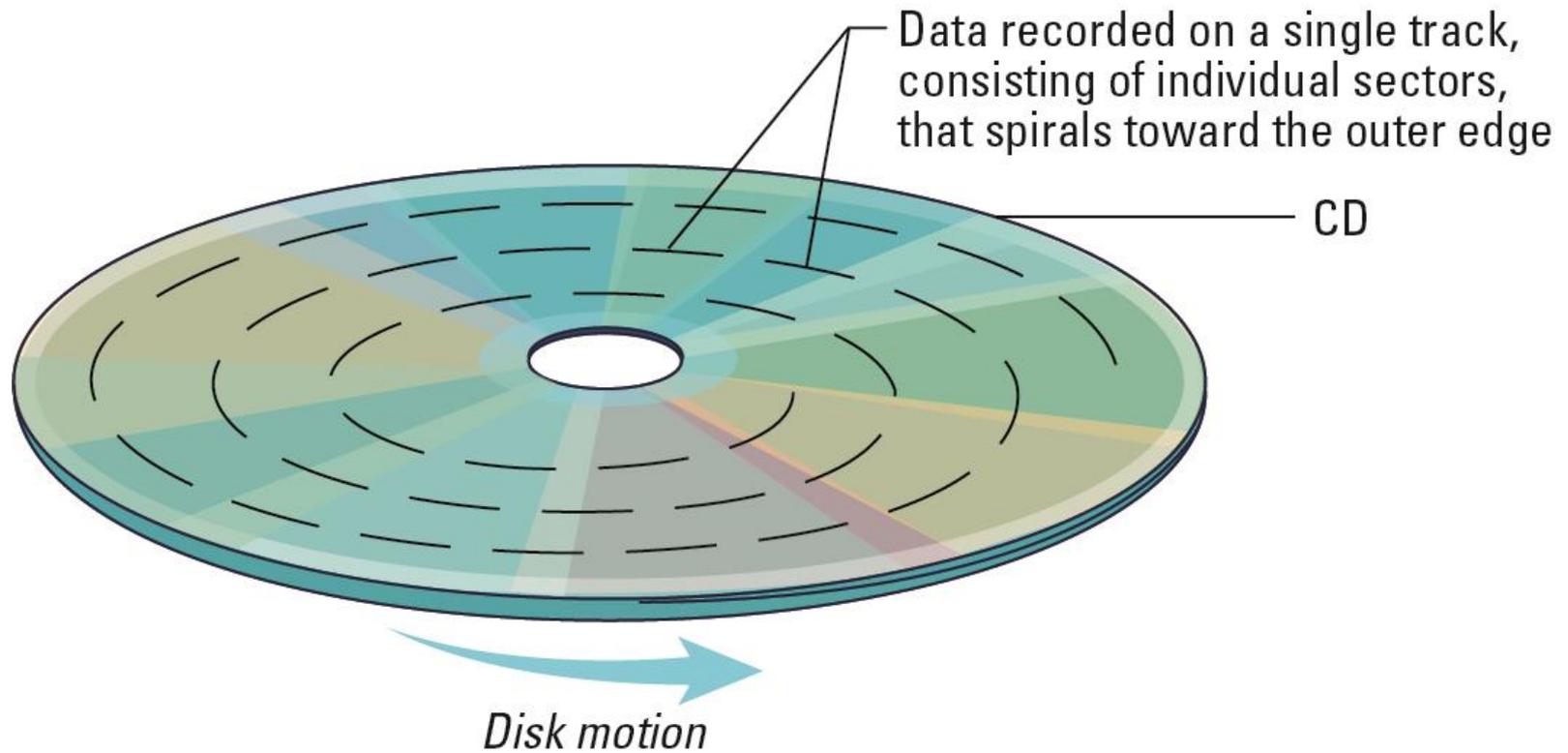
Mass Storage Performance

- **Bandwidth:** quantità totale di bit trasferibili in una unità di tempo
- **Latency:** il tempo totale tra la richiesta di un trasferimento dati ed il suo arrivo a destinazione.

Disco magnetico



Formato di memorizzazione su CD

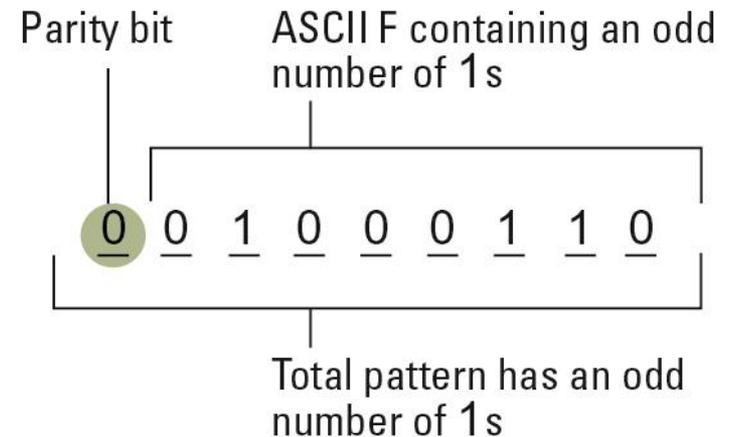
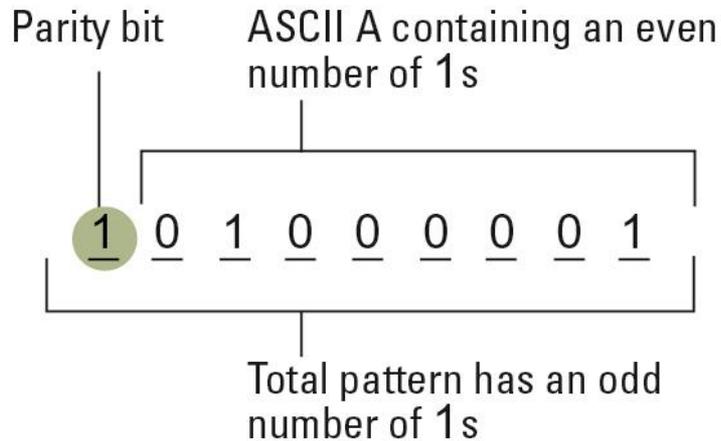


Flash Drives

- **Flash Memory** – circuiti che intrappolano elettroni in piccole celle di biossido di silicio
- Cancellazioni ripetute danneggiano le celle
- Utilizzate per:
 - Digital cameras
 - Smartphones
- **SD Cards** forniscono GBs of spazio

Errori di Comunicazione

- Goal: ridurre gli errori
- Parity bits (even versus odd)
- Controllo byte



Distanza di Hamming

- Distanza di Hamming: numero di bit in cui differiscono due pattern

Esempio:

0 1 1 0 1 0

0 1 0 0 0 0

Distanza di Hamming uguale a 2, cioè:

$$D(011010,010000)=2$$

Codice correttore d'errore

Symbol	Code
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

Distanza di Hamming pari almeno a 3 ($D \geq 3$)

Esempio:

- $D(A,B)=4$
- $D(B,C)=3$

Funzionamento del codice (I)

Il codice rileva l'errore di un bit e lo corregge.

- Il pattern modificato sarà ad una distanza pari a 1 rispetto all'originale ed almeno a 2 dagli altri.

Esempio:

- Originale 0 1 1 1 0 0
- Modificato (1 bit di errore) 0 1 **0** 1 0 0

Funzionamento del codice (II)

- Per decodificare un messaggio:
 - confrontare ogni pattern ricevuto con quelli del codice fino a che $D=1$.
- Codice che rileva al massimo 2 errori e ne corregge 1

Decodificare 010100 usando il codice della tabella precedente

Character	Code	Pattern received	Distance between received pattern and code
A	0 0 0 0 0 0	0 1 0 1 0 0	2
B	0 0 1 1 1 1	0 1 0 1 0 0	4
C	0 1 0 0 1 1	0 1 0 1 0 0	3
D	0 1 1 1 0 0	0 1 0 1 0 0	1
E	1 0 0 1 1 0	0 1 0 1 0 0	3
F	1 0 1 0 0 1	0 1 0 1 0 0	5
G	1 1 0 1 0 1	0 1 0 1 0 0	2
H	1 1 1 0 1 0	0 1 0 1 0 0	4

Smallest distance