



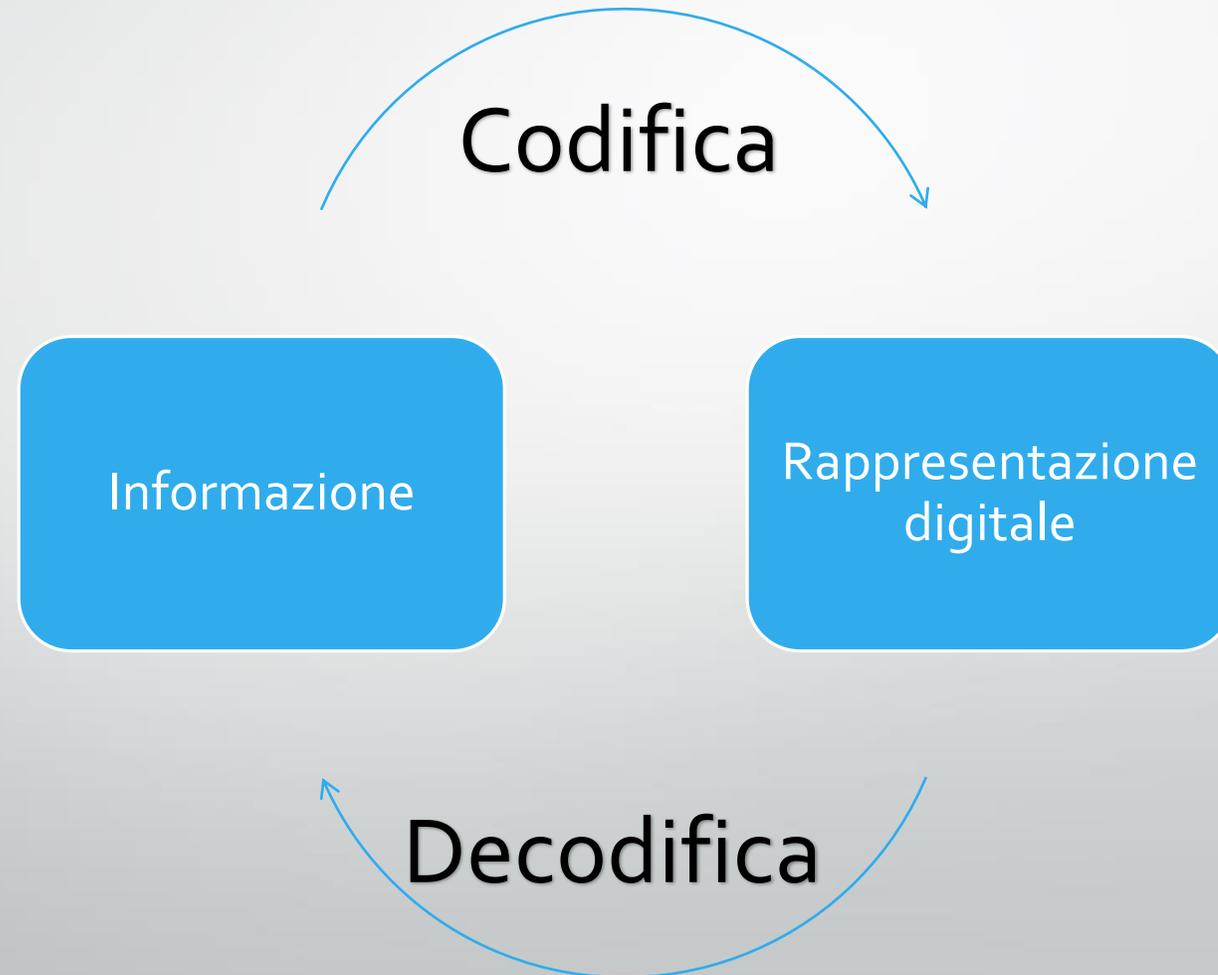
# CODIFICA DELLE INFORMAZIONI

MODULO 5

# INFORMAZIONI: tipi

- Le informazioni sono concetti astratti che esistono indipendentemente dalla loro rappresentazione
- Tutto ciò che ci circonda è informazione
- Qualche esempio:
  - Numeri
  - Testi
  - Immagini
  - Suoni
  - Video

# INFORMAZIONI



# CODIFICA BINARIA: BIT

- Usiamo una rappresentazione binaria:
  - Presenza/assenza di carica elettrica
  - Passaggio/non passaggio di corrente
- BIT: BInary digiT
  - La più piccola informazione che possiamo memorizzare
  - Sì/no – 0/1 – on/off – vero/falso

# CODIFICA BINARIA: BIT

- BIT: Binary digiT
  - La più piccola informazione che possiamo memorizzare
  - Si/no – 0/1 – on/off – vero/falso
- Quante informazioni possiamo rappresentare con 1 bit?
  - Solo 2, come abbiamo appena visto!
  - 0 / 1

# CODIFICA BINARIA: BIT

- Possiamo combinare più bit per rappresentare più informazioni.
- Quante informazioni possiamo rappresentare con 2 bit?
  - 4: 00 / 01 / 10 / 11
- E con 3 bit?
  - 8: 000 / 001 / 010 / 011 / 100 / 101 / 110 / 111
- Regola generale: con N bit possiamo rappresentare  $2^N$  informazioni

# CODIFICA BINARIA: BYTE

- In informatica ha assunto particolare importanza il concetto di BYTE
- 1 byte = 8 bit =  $2^8$  informazioni = 256 informazioni
- BYTE = BinarY ocTEt
- Perché proprio 8 bit? Semplicemente perché per l'indirizzamento della memoria si utilizza la notazione esadecimale...che è compatibile con la notazione binaria

# CODIFICA BINARIA: BYTE

## MULTIPLI DEL BYTE:

- Nel Sistema Internazionale
  - kB = kilobyte =  $10^3$  byte = 1000 byte
  - MB = megabyte =  $10^6$  byte = 1000 kB
  - GB = gibabyte =  $10^9$  byte = 1000 MB
  - TB = terabyte =  $10^{12}$  byte = 1000 GB
  - PB = petabyte =  $10^{15}$  byte = 1000 TB

# CODIFICA BINARIA: BYTE

## MULTIPLI DEL BYTE:

- Utilizzando i prefissi binari

- kiB = kibibyte =  $2^{10}$  byte = 1024 byte
- MiB = mebibyte =  $2^{20}$  byte =  $2^{10}$  kB =  $1024 * 1024$  byte
- GiB = gibibyte =  $2^{30}$  byte =  $2^{10}$  MB =  $1024 * 1024 * 1024$  byte
- TiB = tebibyte =  $2^{40}$  byte =  $2^{10}$  GB =  $1024 * 1024 * 1024 * 1024$  byte
- PiB = pebibyte =  $2^{50}$  byte =  $2^{10}$  TB =  $1024 * 1024 * 1024 * 1024 * 1024$  byte

# CODIFICA BINARIA: Numeri

- Come codifichiamo un numero in binario?
- Effettuando la conversione tra il sistema decimale e quello binario:

$$12_{10} = 1100_2 = C_{16}$$

- Da dove esce la lettera C?

# CODIFICA BINARIA: Numeri

decimal	hexadecimal	binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

# CODIFICA BINARIA: Testi

- Come codifichiamo una parola utilizzando un sistema binario?
- Si utilizza la tabella ASCII :
  - American Standard Code for Information Interchange

# CODIFICA BINARIA: tabela ASCII

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	00000000	NUL	32	0x20	040	01000000	space	64	0x40	100	10000000	@	96	0x60	140	11000000	`
1	0x01	001	00000001	SOH	33	0x21	041	01000001	!	65	0x41	101	10000001	A	97	0x61	141	11000001	a
2	0x02	002	00000010	STX	34	0x22	042	01000010	"	66	0x42	102	10000010	B	98	0x62	142	11000010	b
3	0x03	003	00000011	ETX	35	0x23	043	01000011	#	67	0x43	103	10000011	C	99	0x63	143	11000011	c
4	0x04	004	00000100	EOT	36	0x24	044	01000100	\$	68	0x44	104	10000100	D	100	0x64	144	11000100	d
5	0x05	005	00000101	ENQ	37	0x25	045	01000101	%	69	0x45	105	10000101	E	101	0x65	145	11000101	e
6	0x06	006	00000110	ACK	38	0x26	046	01000110	&	70	0x46	106	10000110	F	102	0x66	146	11000110	f
7	0x07	007	00000111	BEL	39	0x27	047	01000111	'	71	0x47	107	10000111	G	103	0x67	147	11000111	g
8	0x08	010	00010000	BS	40	0x28	050	01010000	(	72	0x48	110	10010000	H	104	0x68	150	11010000	h
9	0x09	011	00010001	TAB	41	0x29	051	01010001	)	73	0x49	111	10010001	I	105	0x69	151	11010001	i
10	0x0A	012	00010010	LF	42	0x2A	052	01010010	*	74	0x4A	112	10010010	J	106	0x6A	152	11010010	j
11	0x0B	013	00010011	VT	43	0x2B	053	01010011	+	75	0x4B	113	10010011	K	107	0x6B	153	11010011	k
12	0x0C	014	00011000	FF	44	0x2C	054	01011000	,	76	0x4C	114	10011000	L	108	0x6C	154	11011000	l
13	0x0D	015	00011001	CR	45	0x2D	055	01011001	-	77	0x4D	115	10011001	M	109	0x6D	155	11011001	m
14	0x0E	016	00011010	SO	46	0x2E	056	01011010	.	78	0x4E	116	10011010	N	110	0x6E	156	11011010	n
15	0x0F	017	00011011	SI	47	0x2F	057	01011011	/	79	0x4F	117	10011011	O	111	0x6F	157	11011011	o
16	0x10	020	00100000	DLE	48	0x30	060	01100000	0	80	0x50	120	10100000	P	112	0x70	160	11100000	p
17	0x11	021	00100001	DC1	49	0x31	061	01100001	1	81	0x51	121	10100001	Q	113	0x71	161	11100001	q
18	0x12	022	00100010	DC2	50	0x32	062	01100010	2	82	0x52	122	10100010	R	114	0x72	162	11100010	r
19	0x13	023	00100011	DC3	51	0x33	063	01100011	3	83	0x53	123	10100011	S	115	0x73	163	11100011	s
20	0x14	024	00100100	DC4	52	0x34	064	01100100	4	84	0x54	124	10100100	T	116	0x74	164	11100100	t
21	0x15	025	00100101	NAK	53	0x35	065	01100101	5	85	0x55	125	10100101	U	117	0x75	165	11100101	u
22	0x16	026	00100110	SYN	54	0x36	066	01100110	6	86	0x56	126	10100110	V	118	0x76	166	11100110	v
23	0x17	027	00100111	ETB	55	0x37	067	01100111	7	87	0x57	127	10100111	W	119	0x77	167	11100111	w
24	0x18	030	00110000	CAN	56	0x38	070	01110000	8	88	0x58	130	10110000	X	120	0x78	170	11110000	x
25	0x19	031	00110001	EM	57	0x39	071	01110001	9	89	0x59	131	10110001	Y	121	0x79	171	11110001	y
26	0x1A	032	00110010	SUB	58	0x3A	072	01110010	:	90	0x5A	132	10110010	Z	122	0x7A	172	11110010	z
27	0x1B	033	00110011	ESC	59	0x3B	073	01110011	;	91	0x5B	133	10110011	[	123	0x7B	173	11110011	{
28	0x1C	034	00111000	FS	60	0x3C	074	01111000	<	92	0x5C	134	10111000	\	124	0x7C	174	11111000	
29	0x1D	035	00111001	GS	61	0x3D	075	01111001	=	93	0x5D	135	10111001	]	125	0x7D	175	11111001	}
30	0x1E	036	00111010	RS	62	0x3E	076	01111010	>	94	0x5E	136	10111010	^	126	0x7E	176	11111010	~
31	0x1F	037	00111011	US	63	0x3F	077	01111011	?	95	0x5F	137	10111011	_	127	0x7F	177	11111011	DEL

# CODIFICA BINARIA: tabella ASCII

- Ad esempio, la parola INFORMATICA si traduce in:

	I	N	F	O	R	M	A
Hx	49	4E	46	4F	52	4D	41
Bn	01001001	01001110	01000110	01001111	01010010	01001110	01000001
	T	I	C	A			
Hx	54	49	43	41			
Bn	01010100	01001001	01000011	01000001			





# IMMAGINI DIGITALI

Nozioni di base

# IMMAGINI DIGITALI

- Un'immagine è una rappresentazione visiva, non solida, della realtà.
- Può rappresentare la realtà fisica (in modo più o meno realistico) oppure una realtà fittizia o astratta.
- Un'immagine GRAFICA è un'immagine fissata su un supporto fisico oppure visualizzata (in modo più o meno prolungato nel tempo) su uno schermo di visualizzazione.

# IMMAGINI DIGITALI

- Perché le immagini siano elaborate al computer occorre trasformarle in una rappresentazione numerica/digitale, attraverso un processo chiamato **DIGITALIZZAZIONE**.
- Un'immagine digitale è la rappresentazione numerica di una immagine bidimensionale.
- La rappresentazione può essere di tipo **RASTER** o **VETTORIALE**.
  - **RASTER**: l'immagine è composta da una matrice di punti, detti pixel, la cui colorazione è definita (codificata) tramite uno o più valori numerici (bit).
  - **VETTORIALE**: sono descritti degli elementi primitivi, quali linee o poligoni, che vanno a comporre l'immagine

# IMMAGINI DIGITALI

## IMMAGINE RASTER o BITMAP:

- nelle immagini a COLORI, viene memorizzato solitamente il livello di intensità dei colori fondamentali (nel modello di colore RGB, uno dei più usati, sono tre: rosso, verde e blu. Un altro esempio è CMYK, usato per la stampa, basato su quattro colori fondamentali: ciano, magenta, giallo e nero.)
- nelle immagini MONOCROMATICHE in scala di grigio (dette impropriamente bianco e nero) il valore indica l'intensità del grigio, che varia dal nero al bianco.

# IMMAGINI DIGITALI

IMMAGINE RASTER o BITMAP:



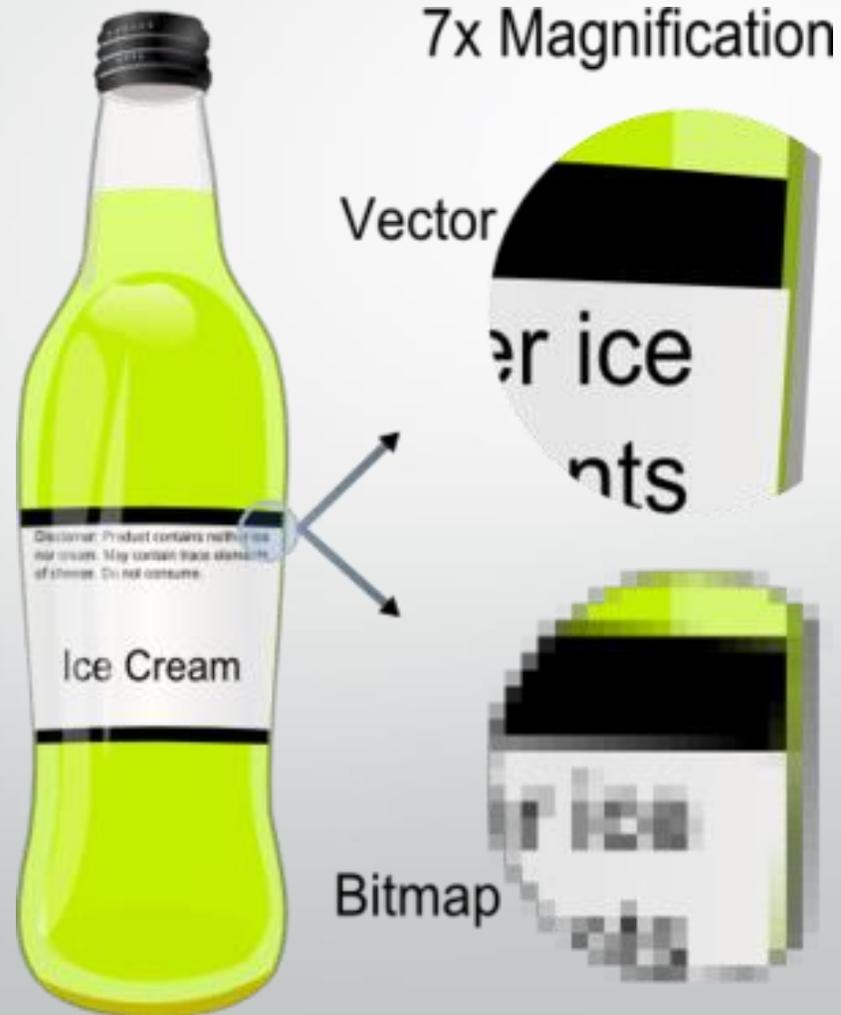
# IMMAGINI DIGITALI

## IMMAGINE RASTER o BITMAP:

- Le immagini BITMAP possono essere memorizzate in vari formati, basati su algoritmi di compressione, che possono essere:
  - LOSSY – con perdita
    - Bitmap – Jpeg
  - LOSSLESS – senza perdita
    - GIF – PNG

# IMMAGINI DIGITALI

IMMAGINE VETTORIALE:



# IMMAGINI DIGITALI

## IMMAGINE VETTORIALE:

- Le immagini vettoriali sono un tipo di immagine facilmente scalabile e ruotabile, ottenuto dall'unione di un certo numero di punti o nodi, che formano linee e poligoni, a loro volta uniti in strutture più complesse, fino a formare l'immagine voluta.
- Questo tipo di immagine è utilizzato nel disegno tecnico per la progettazione architettonica ed industriale, nella rappresentazione di certi font, nella grafica per la creazione di loghi e marchi o altri oggetti, eccetera.

# IMMAGINI DIGITALI

## IMMAGINE VETTORIALE:

- Vantaggi:
  - Possibilità di esprimere i dati in una forma direttamente comprensibile ad un essere umano
  - Possibilità di esprimere i dati in un formato che occupi (molto) meno spazio rispetto all'equivalente raster
  - Possibilità di ingrandire l'immagine arbitrariamente, senza che si verifichi una perdita di risoluzione dell'immagine stessa
- Svantaggi:
  - Difficoltà di utilizzo dei software

# IMMAGINI DIGITALI

## PRINCIPALI FORMATI:

- Analogici
  - 333×480 : VHS
  - 580×480 : Super VHS, LaserDisc
  - 768×576 : Trasmissioni televisive (PAL)
- Digitali
  - 720×480 : DVD
  - 1280×720 : Blu-ray
  - 1920×1080 : FullHD, Blu-ray
  - 3840×2160 : 4K UHD TV



# ALGEBRA BOOLEANA

# ALGEBRA BOOLEANA

- Il calcolatore può essere visto come una rete logica cioè come un insieme di dispositivi chiamati porte logiche opportunamente connessi.
- Le porte logiche sono dispositivi capaci di eseguire operazioni logiche su segnali binari.
- I segnali binari sono livelli di tensione.
- Questi livelli sono identificati tramite una coppia di simboli:
  - 0, 1
  - False, True

# ALGEBRA BOOLEANA

- Le tecniche di composizione delle porte logiche in una rete sono derivate da una particolare algebra operante su variabili binarie e chiamata Algebra Booleana.
- Prende il nome dal matematico inglese George Boole (1815-1864).
- Nel 1938 Shannon ha dimostrato come l'algebra booleana potesse essere presa a fondamento per la progettazione di circuiti logici digitali.
- Vengono definiti i seguenti concetti:
  - variabili booleane
  - operatori booleani
  - funzioni booleane
  - ecc

# VARIABILI BOOLEANE

- Una variabile booleana è una variabile binaria che può assumere esclusivamente due valori logici che saranno denotati con **0** e **1**.
- Se  $x$  è una variabile booleana, vale quindi la seguente definizione formale:

$$x = 0 \text{ se } x \neq 1$$

$$x = 1 \text{ se } x \neq 0$$

# OPERATORI BOOLEANI

- Si definiscono gli operatori booleani o fondamentali:

**AND:** prodotto logico

**OR:** somma logica

**NOT:** negazione logica

# OPERATORI BOOLEANI: AND

- È una operazione binaria (o n-aria) che restituisce 1 solo nel caso in cui tutti i valori delle variabili assumono valore logico 1.
- Per rappresentare il prodotto logico di due variabili  $x$  e  $y$  si usa la notazione:

$x \text{ and } y$

$x \cdot y$

$x y$

# OPERATORI BOOLEANI: AND

- Tabella della verità:

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

# OPERATORI BOOLEANI: AND

- Proprietà:

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

$$x \cdot x = x$$

$$x \cdot \bar{x} = 0$$

# OPERATORI BOOLEANI: OR

- È una operazione binaria (o n-aria) che restituisce 1 se e solo se almeno una delle variabili assume valore logico 1.
- Per rappresentare il prodotto logico di due variabili  $x$  e  $y$  si usa la notazione:

$x \text{ or } y$

$x + y$

# OPERATORI BOOLEANI: OR

- Tabella della verità:

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

# OPERATORI BOOLEANI: OR

- Proprietà:

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + \bar{x} = 1$$

# OPERATORI BOOLEANI: NOT

- È una operazione unaria che restituisce il valore logico opposto a quello della variabile di ingresso.
- Per rappresentare il complemento di una variabile  $x$  vengono usate varie notazioni:

$not(x)$

$\neg x$

$\bar{x}$

# OPERATORI BOOLEANI: NOT

- Tabella della verità:

x	not(x)
0	1
1	0

- Proprietà:

$$\text{not}(\text{not}(x)) = x$$

# ALGEBRA BOOLEANA: PROPRIETA'

- Idempotenza

$$x + x = x$$

$$x \cdot x = x$$

- Elemento nullo

$$x + 1 = 1$$

$$x \cdot 0 = 0$$

- Proprietà Commutativa

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

- Proprietà Associativa

$$x + (y + z) = (x + y) + z = x + y + z$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z = x \cdot y \cdot z$$

# ALGEBRA BOOLEANA: PROPRIETA'

- Proprietà Distributiva

$$x \cdot y + x \cdot z = x \cdot (y + z)$$
$$(x + y) \cdot (x + z) = x + y \cdot z$$

- Assorbimento

$$x + (x \cdot y) = x$$
$$x \cdot (x + y) = x$$
$$(x + y) \cdot y = x \cdot y$$
$$x \cdot y + y = x + y$$

- Teoremi di De Morgan

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$
$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

# OPERATORI BOOLEANI: NAND

- L'operazione di nand logico è l'operazione negata dell'operazione and.
- Corrisponde alla contrazione di not and.
- Per rappresentare il prodotto logico di due variabili  $x$  e  $y$  si usa la notazione:

*$x \text{ nand } y$*

# OPERATORI BOOLEANI: NAND

- Tabella della verità:

x	y	x nand y
0	0	1
0	1	1
1	0	1
1	1	0

# OPERATORI BOOLEANI: NAND

- Proprietà:

$$x \text{ nand } 0 = 1$$

$$x \text{ nand } 1 = \bar{x}$$

$$x \text{ nand } x = \bar{x}$$

$$x \text{ nand } \bar{x} = 1$$

$$x \text{ nand } y = \overline{x \cdot y}$$

# OPERATORI BOOLEANI: NOR

- L'operazione di nand logico è l'operazione negata dell'operazione or.
- Corrisponde alla contrazione di not od.
- Per rappresentare il prodotto logico di due variabili  $x$  e  $y$  si usa la notazione:

*$x \text{ nor } y$*

# OPERATORI BOOLEANI: NOR

- Tabella della verità:

x	y	x nor y
0	0	1
0	1	0
1	0	0
1	1	0

# OPERATORI BOOLEANI: NOR

- Proprietà:

$$x \text{ nor } y = \overline{x + y}$$

$$x \text{ nor } \bar{x} = 0$$

$$x \text{ nor } x = \bar{x}$$

$$x \text{ nor } 1 = 0$$

$$x \text{ nor } 0 = \bar{x}$$

# FORMA CANONICA SP

- La forma canonica Somma di Prodotti (SP) di una funzione logica si ottiene sommando i **minterm** in corrispondenza dei quali la funzione vale 1.
- Esempio:

x	y	z	F	Minterm
0	0	0	1	$\bar{x}\bar{y}\bar{z}$
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{x}y z$
1	0	0	1	$x\bar{y}\bar{z}$
1	0	1	0	
1	1	0	1	$x y \bar{z}$
1	1	1	0	

$$F(x, y, z) = \bar{x}\bar{y}\bar{z} + \bar{x}y z + x\bar{y}\bar{z} + x y \bar{z}$$