LOGICA PROPOSIZIONALE E ALGEBRA DI BOOLE

Def.: La logica proposizionale è un "MODELLO MATEMATICO" che ci consente di ragionare sulla verità o falsità di espressioni logiche.

Espressione logica: è una combinazione di variabili proposizionali e costanti logiche, che possono così assumere valore .TRUE. o .FALSE., e opportuni operatori logici. L'espressione logica a sua volta può restituire un solo valore T o F.

Un insieme di base di espressioni logiche può essere definito in maniera INDUTTIVA:

BASE: le variabili proposizionali e le costanti logiche sono espressioni logiche. Sono operandi atomici.

INDUZIONE: Se E e F sono espressioni logiche, lo sono anche le seguenti:

a) E .AND. F, Assume valore .True. se sia E che F sono .TRUE., .False. altrimenti.

b) E .OR. F. Assume valore .True. se almeno uno tra E ed F sono .True. altrimenti è .False.

c) .NOT. E. Assume valore .True. se E è .False. e .False. se E è .True.

Le espressioni logiche sono quindi ottenute combinando gli operatori binari .AND. e .OR. e l'operatore unario prefisso .NOT.

Questo insieme di regole costituisce un'ALGEBRA, l'algebra di BOOLE

Analogia con: Computer

BIT : 1 → .True.

0 → .False.

Il computer prende decisioni, quindi compie certe operazioni se una espressione logica è .True.

SE (ESPR. LOGICA) FAI....

SE $(A<B)$ .OR. $((A \geq B)$ .AND. $(C=D))$ FAI....

↓ posso semplificarla in

SE $(A<B)$ .OR. $(C=D)$ FAI...

Per semplificare ci basta osservare che $A<B$ e $A \geq B$ sono complementari, cioè se è vero uno, l'altro è falso. Possiamo sostituire a ciascuna espressione una variabile proposizionale

$A<B \Rightarrow p$ $\quad$ $A \geq B \Rightarrow$ .NOT. p

$C=D \Rightarrow q$

$$p \text{ .OR. } ((\text{.NOT. } p) \text{ .AND. } q) \equiv p \text{ .OR. } q$$

A questo punto p, q potrebbero essere proposizioni qualsiasi:

p = C'è il sole

q = Francesco prende l'ombrello.

1) C'è il sole oppure non c'è il sole e Francesco prende l'ombrello

2) C'è il sole oppure Francesco prende l'ombrello

PRECEDENZA DEGLI OPERATORI

1) .NOT. } RAGGRUPPATO DA SINISTRA
2) .AND. } RAGGRUPPATI DA DESTRA
3) .OR.

.NOT. .NOT. p .OR. q $\Leftrightarrow$ (.NOT. (.NOT.) p) . OR . q

.NOT. p . OR . q . AND . r $\Leftrightarrow$ (.NOT. p) . OR . (q . AND . r)

Analogia con + (.OR.), × (. AND.) e $^{-}$ (.NOT.)

$$\bar{\bar{p}} + q \Leftrightarrow p + q$$

$$\bar{p} + qr$$

TABELLE DI VERITÀ

p	q	p . and . q	p . OR . q	.NOT. p
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Utili per comprendere le espressioni logiche il loro uso però è limitato ad espressioni con poche variabili poiché il numero di righe è pari a 2^K con K numero variabili.

ALTRI OPERATORI LOGICI

IMPLICAZIONE : $\rightarrow$ $p \rightarrow q$ « Se p è vero allora q è vero »

Assume valore .true. se p è vero e q è vero oppure se p è falso

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

può essere rappresentata da .NOT. p . AND. q

EQUIVALENZA: $p \equiv q$ è vero se sia p che q sono vere o sono entrambe false

p	q	$p \equiv q$
0	0	1
0	1	0
1	0	0
1	1	1

si rappresenta come p.AND.q .OR. (NOT.p.AND.NOT.q)

NAND: p.NAND.q $\Leftrightarrow$ .NOT.(p.AND.q) $\Leftrightarrow$ (.NOT.p).OR.(.NOT.q)

NOR: p.NOR.q $\Leftrightarrow$ .NOT.(p.OR.q) $\Leftrightarrow$ (.NOT.p).AND.(.NOT.q)

ESEMPIO:

p.AND.q $\longrightarrow$ p.OR.r

Costruire la TABELLA DI VERITÀ

p	q	r	p.AND.q	p.OR.r	ESPR
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	1	1	1

In generale una ESPRESSIONE LOGICA e LA SUA TABELLA DI VERITÀ rappresentano UNA FUNZIONE BOOLEANA.
Cioè una funzione che ha per argomenti variabili proposizionali e restituisce una variabile proposizionale.
Il problema generale è costruire una ESPRESSIONE LOGICA A PARTIRE DA UNA TABELLA DI VERITÀ

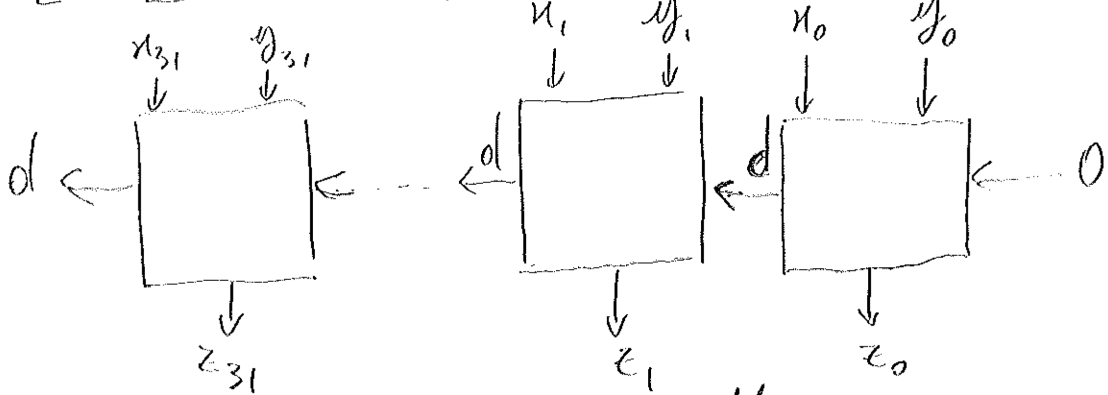
ESEMPIO: SOMMA DI INTERI IN BASE 2

0 1 0 0 1 1	+	x
0 1 1 0 0 1		y
1 0 0 1 1 0	RIPORTO	
1 0 1 1 0 0		z

$x = \boxed{x_{31}}\,\boxed{\;}\,\boxed{x_2}\,\boxed{x_1}\,\boxed{x_0}$

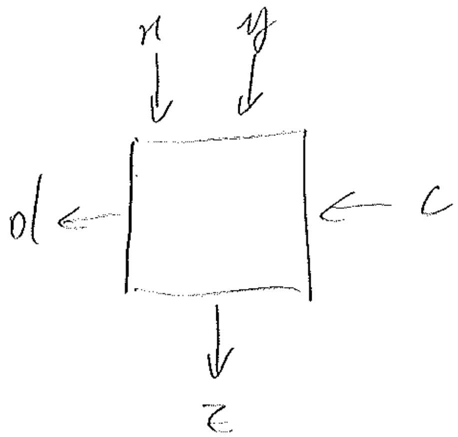
$y = \boxed{y_{31}}\,\cdots\,\boxed{y_2}\,\boxed{y_1}\,\boxed{y_0}$

$z = \boxed{z_{31}}\,\cdots\,\boxed{z_2}\,\boxed{z_1}\,\boxed{z_0}$



Se $d \neq 0$ allora ho un overflow

Macchina elementare è la seguente.



	x	y	c	d	z
0.	0	0	0	0	0
1.	0	0	1	0	1
2.	0	1	0	0	1
3.	0	1	1	1	0
4.	1	0	0	0	1
5.	1	0	1	1	0
6.	1	1	0	1	0
7.	1	1	1	1	1

COSTRUIAMO L'ESPRESSIONE LOGICA DEL RIPORTO d

1) Righe 3 e 7, d è 1 se y e c sono 1
2) Righe 5 e 7, d è 1 se x e c sono 1
3) Righe 6 e 7, d è 1 se x e y sono 1

$$\left.\begin{array}{ll} 1) & y\ .AND.\ c \\ 2) & x\ .AND.\ c \\ 3) & x\ .AND.\ y \end{array}\right\} \Rightarrow \underbrace{(y.AND.c).OR.(x.AND.c).OR.(x.AND.y)}_{d = yc + xc + x.y}$$

$p.AND.q.OR.r \Rightarrow pq + r$

$p.AND..NOT.q.OR..NOT.r \Rightarrow p\bar{q} + \bar{r}$

COSTRUZIONE DI ESPR. LOGICA

In generale è molto difficile costruire una espressione logica con un numero minimo di termini. Tuttavia si può passare da tabella a espressione in modo automatico.

Tecnica:

$m_1 .OR. m_2 .OR. m_3 \ldots$

Dove m_i è un termine che corrisponde ad un 1 della mia tabella. m_i è detto MINTERMINE.

LETTERALE è una espressione che consiste di sola variabile.

Costruiamo un mintermine con tanti letterali collegati da .AND. quante sono le colonne della tabella.

Se la variabile p in tabella vale 1 il letterale è p se è 0 è .NOT.p

$$d = \bar{x}yc + x\bar{y}c + xy\bar{c} + xyc = (\bar{x}+x)yc + x\bar{y}c + xy\bar{c} = yc + xc + xy$$

$$z = \bar{x}\bar{y}c + \bar{x}y\bar{c} + x\bar{y}\bar{c} + xyc$$

Circuiti Logici

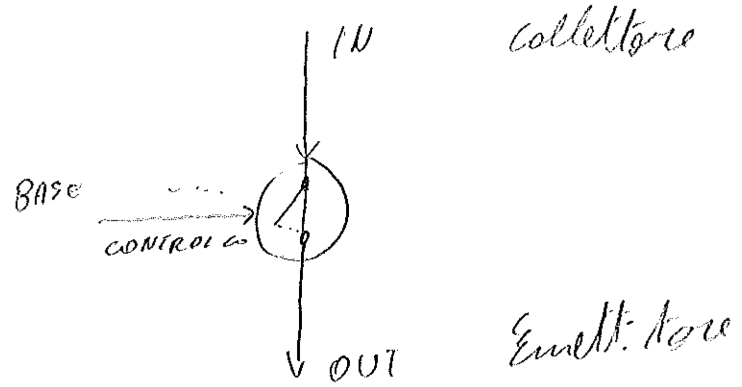
Computer sono composti da transistor:

- 2 stati: conduzione, interdizione (acceso/spento)
- Altissima densità di integrazione 3-10 M trans/cm^2
- Altissima frequenza di commutazione

1000 - 3000 milioni di cicli al secondo (Hertz)

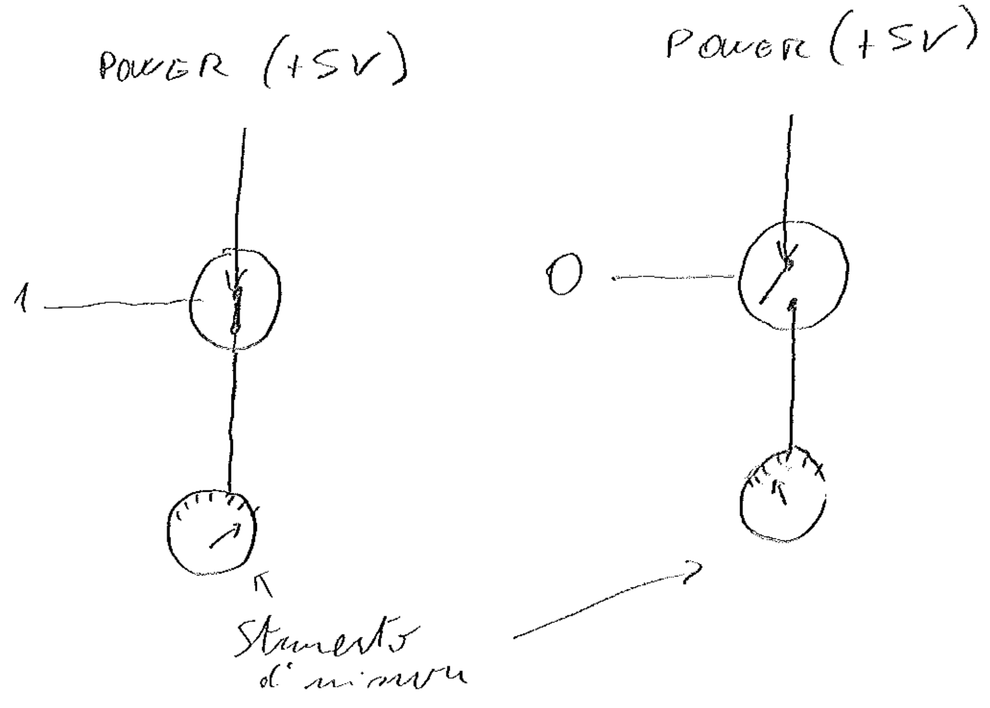
0,3 ÷ 1 miliardesimi di secondo

SCHEMA



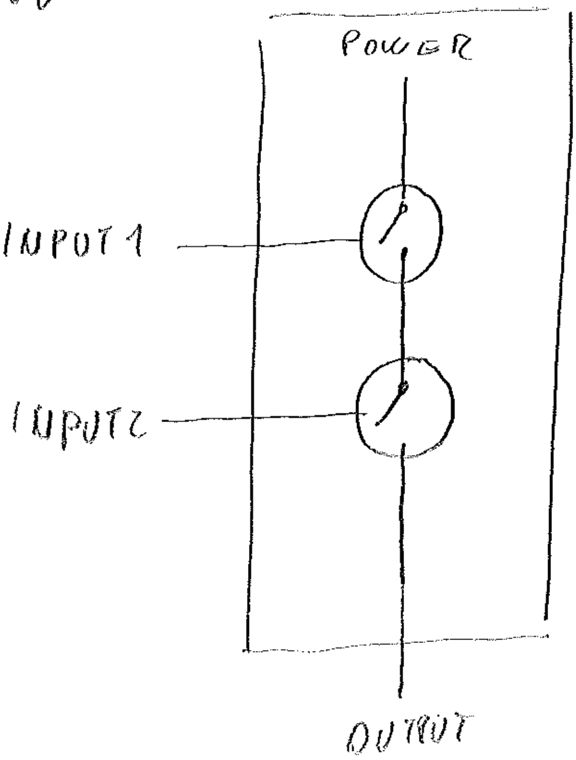
La base in presenza di tensione chiude lo switch e permette il passaggio di corrente dal collettore all'emettitore

→ LOGICA POSITIVA: Associo valore 1 al passaggio di corrente
LOGICA NEGATIVA: " " 0 " " " "

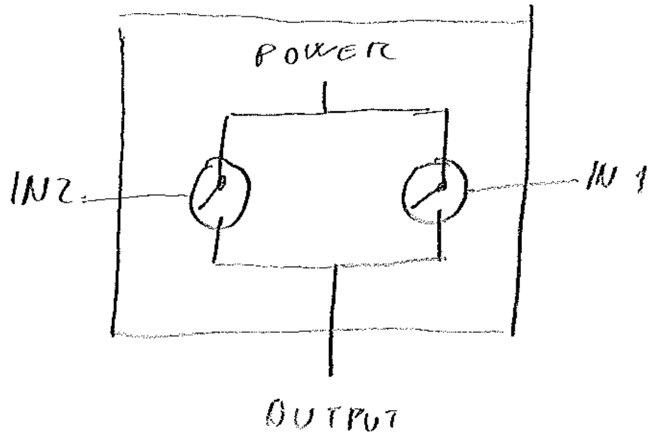


Costruzione degli Operatori Logici

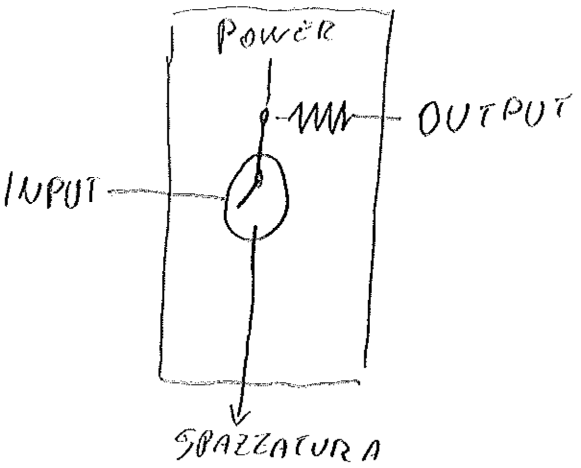
AND



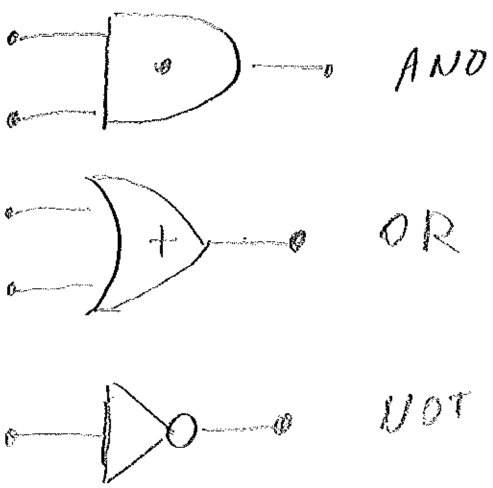
OR



NOT



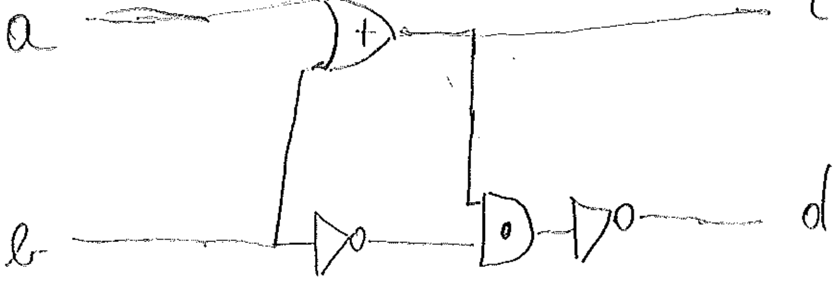
SIMBOLI



ESEMPIO
IN (a, b)
OUT(c, d) con

$c = a \text{ .OR. } b$

$d = \text{.NOT.}((a \text{ .OR. } b) \text{ .AND. } (\text{NOT. } b))$



CIRCUITO CE (Compare for equality)

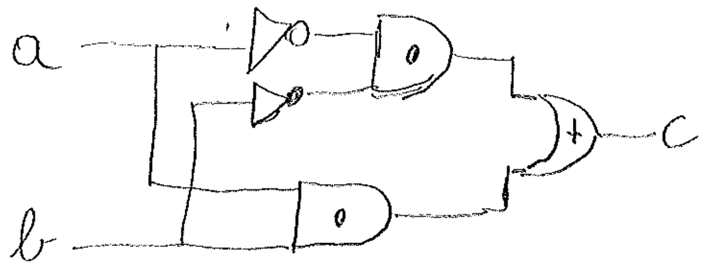
$a \equiv b$

Tabella di verità:

a	b	$a \equiv b$
0	0	1
0	1	0
1	0	0
1	1	1

$\Rightarrow \bar{a}\,\bar{b} + a\,b =$

$= (.NOT.\ a\ .AND.\ .NOT.\ b)\ .OR.\ (a\ .AND.\ b)$



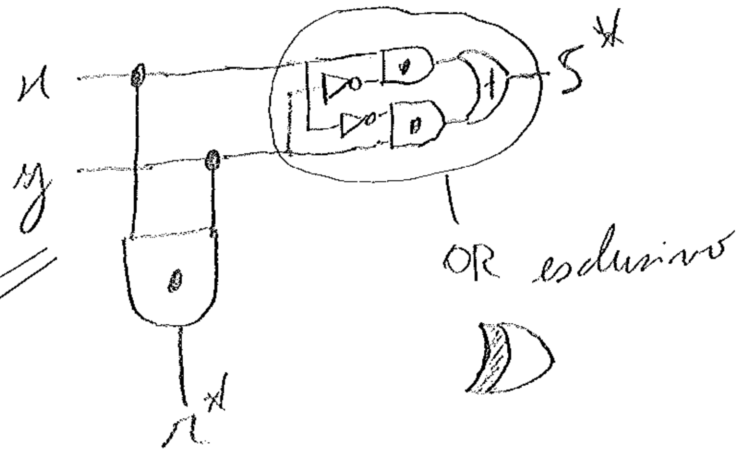
CIRCUITO SOMMATORE

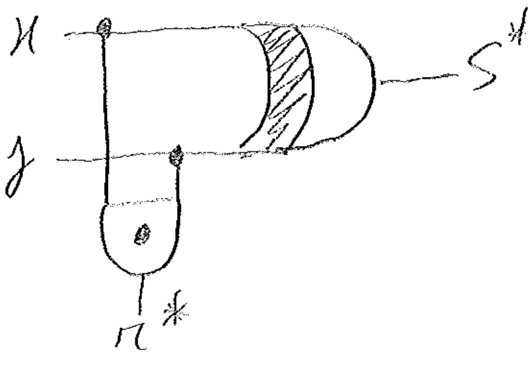
Caso $c = 0$

x	y	Somma*	Riporto*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S^* = \bar{x}\,y + x\,\bar{y}$

$r^* = x\,y$



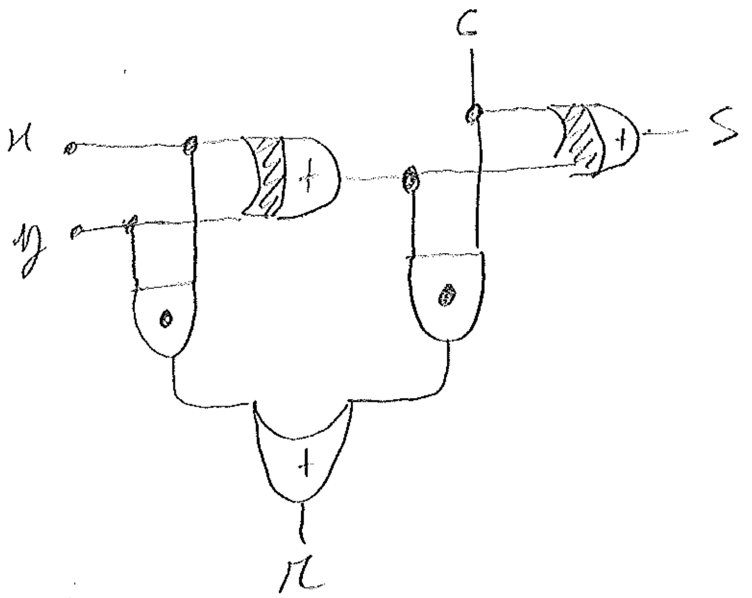


Se abbiamo c come si modificano somma e riporto

S^*	C	S
0	0	0
0	1	1
1	0	1
1	1	0

OR esclusivo

$$S^*C + r^* = r$$



Circuito Decodificatore