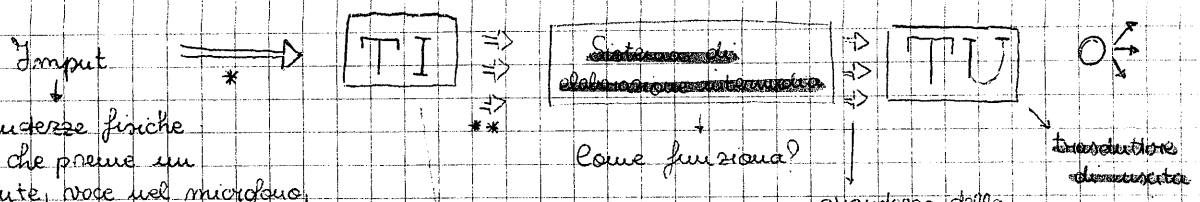
21/03/2015

DOMANDA GUIDA: come funziona un sistema che elabora informazioni? In che forma sono le (**)?

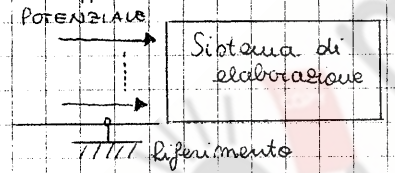


Grandezze fisiche (dito che preme un pulsante, voce nel microfono, segnale elettrico, banda frequenza, ampiezza...)

trasduttore d'ingresso: trasforma la prima grandezza fisica (*) in altre intermedie (es: segnali elettrici, posizioni meccaniche, pressioni, valvole)

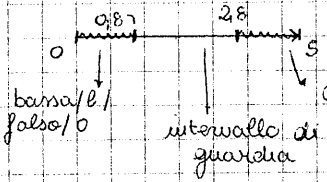
Grandezze fisiche < **discrete** → valori distinti assumibili
continue → gamma infinita di valori assumibili (es: tra 15° e 35°)

Es 1: grandezza intermedia = tensione del filo è una grandezza fisica continua. Problema: non siamo in grado di avere una misura esatta, precisione assoluta; tuttavia se applico diverse misure in entrata, avrò risultati diversi in uscita



Es 2: un led si illumina se ho luce < 250 lumen. "Misurando" la stessa luce in modi diversi, avrò risultati diversi.

Soluzione: scompongo il problema in 2 parti, usando come ingresso al sist. di elaborazione grandezze discrete, avendo solo alcuni valori «assumibili»
Es: Intervallo 0-5 volt. Definiamo 2 regioni e facciamo sì che il sistema di elab. funzioni solo nella fascia alta e bassa (L e H / f. e v. / 0 e 1)
In questo caso, se compio un errore piccolo (es: 0,1 invece che 0) rimango «tutelata», ho più margine di errore.



Per rappresentare le g.f **discrete** continue mediante g.f discrete, «raggruppo» diverse g.f continue e le associo ad una unica discreta.
Es: intervallo -20°C – 100°C, ±1% errore; allora avrò 50 intervalli da 2,4 °C a cui corrispondono circa 50 g.f discrete.

Se voglio 50 **configurazioni** distinte, come deve costruire il trasduttore?
1 filo → 2 config (A B)
2 fili → 4 " (AA BB AB BA)
3 " → 8 "
~~fili~~ → 2^n " "
} nel caso in cui abbia solo 2 intervalli (A, B) con opportuno intervallo di guardia

Se avessi 5 intervalli (con opportuni int. di guardia) avrei 5^n configurazioni

Per avere m configurazioni 2^n, $n = \lceil \log_2 m \rceil$ → Dato $2^{n-1} \leq m \leq 2^n$

$\lceil\ \rceil$ = intero superiore. $n-1 \leq \log_2 m \leq n$.

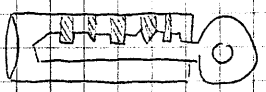
2 = # minimo di configurazioni; se una variabile ha 2 config possibili ha 1 bit

Quantità di info $B = \log_2$ (# configurazioni)

Comportamento della «scatola nera»

- **SISTEMA COMBINATORIO**: indipendentemente da ciò che è successo prima, un certo ingresso darà sempre la stessa uscita
- **SEQUENZIALE**: ricorda la "storia" degli ingressi; in base a ciò che è avvenuto prima, a parità di ingresso cambia l'uscita

Es 1) Chiave sistema combinatorio. Il cilindro ha una serie di cilindretti di diverse lunghezze. Inserendo la chiave giusta, ovvero con le scanalature adatte in cui si inseriscono i cilindretti, allineandoli, permettendo al cilindro principale di ruotare. Altrimenti la rotazione è ostacolata dagli stessi cilindretti.





Se avessi 5 cilindretti e ognuno potesse assumere 10 diverse posizioni, posso avere 10^5 combinazioni, ma solo una di esse «funziona» } Ingressi = varie posizioni
Uscite = aprio / non apro

«Tabella di verità»

Ingressi	Uscite
0 0 0 0 1	non apre
0 0 0 0 2	" "
...	...
3-5-2-1-3	apre!

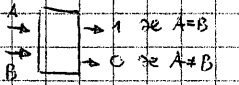
Es 2)

Ingressi	Uscita
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	0



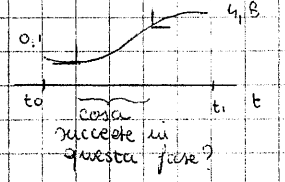
Es 3)

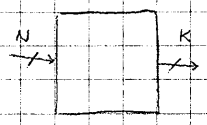
A	B	Uscita
0	0	1
0	1	0
1	0	0
1	1	1



Es 4) Esistono anche reti combinatorie con uscita unica per tutti i diversi ingressi, non molto utili

Ogni ingresso può avere solo 2 comportamenti. Se nel tempo varia la tensione...





Neutrate
K uscite

Tabella di verità

Ingressi	Uscite
2^n	N | K

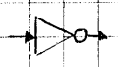
Posso scomporre la rete combinatoria in reti più semplici
Ciò deriva dal fatto che ogni uscita è indipendente, autonoma dalle altre
Quindi costruirò K reti distinte con un'uscita.



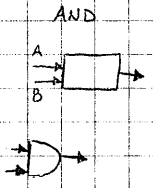
3 PORTE LOGICHE

NOT

i	u
0	1
1	0



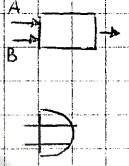
AND



(i)

A	B	u
0	0	0
0	1	0
1	0	0
1	1	1

OR



A	B	u
0	0	0
0	1	1
1	0	1
1	1	1

Es: AND_3

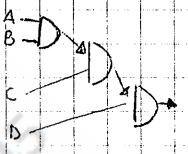
A	B	C	u
1	1	1	1
others			ϕ

AND_N

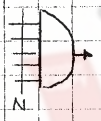
A_0	...	A_{n-1}	u
1	...	1	1
others			ϕ

Infatti

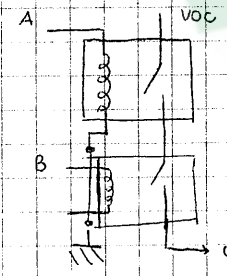




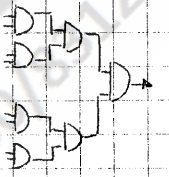
Posso semplificare in



Nella realtà: circuiti elettronici (porte and); il circuito in genere risponde in un tempo finito con tempi di ritardo (anche picosecondi). Una porta a n ingressi ha ritardo proporzionale a n ingressi.



Per ridurre il ritardo, costruisco in un altro modo:



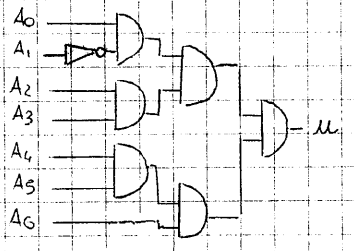
La profondità è logaritmica (non più lineare) rispetto al # ingressi

In questo caso, 8 ingressi e 7 porte and. Rispetto al caso lineare, ho ugual # di porte and e di ingressi, ma questo è più veloce)

Se avessimo...

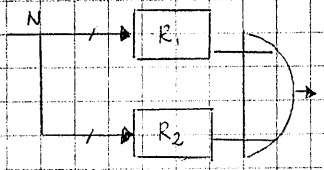
A_0	...	A_{n-1}	u
1	0	1 1 1 1 1 1	1
others			ϕ

dovrei impiegare un invertitore



Le uscite

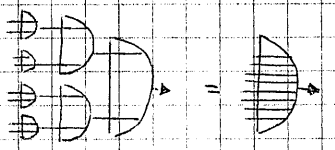
	A_0						A_{n-1}	u
R_1	1	1	0	1	1	1	1	1
R_2	1	0	0	0	0	0	0	1
			others					0



Ricorro alla «OR»

A	B	u
0	0	0
0	1	1
1	0	1
1	1	1

A	B	C	u
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



Altra notazione: **Funzioni booleane**

Date A, B {
- not(A) $\triangleq \bar{A}$ /A !A
- (A or B) $\triangleq$ A+B A|B
- (A and B) $\triangleq$ A·B A &B

Quindi...

A	B	A+B	$\overline{A+B}$	A·B	$\overline{A \cdot B}$
0	0	0	1	0	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	1	0

$\overline{A+B} = \bar{A}\cdot\bar{B}$ $\qquad$ $\overline{A \cdot B} = \bar{A}+\bar{B}$

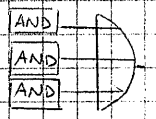
in generale, $\overline{A_0 \cdots A_{n-1}} = \bar{A}_0 + A_1 + \dots + A_{n-1}$

Formule di De Morgan

NAND (= and negato)



Somma di prodotti



1101111
1000000
} $u = A_0 A_1 \bar{A_2} A_3 A_4 A_5 A_6 + A_0 \bar{A_1} \bar{A_2} \bar{A_3} \bar{A_4} \bar{A_5} \bar{A_6}$

*

A	A'	NAND
0	0	1
1	1	0

→ NOT

5/03/2015

Proprietà funz booleane

$A(B+C) = AB + AC$

OR

$A + A = ?$

OR	A_1	A_2	u
A=0 →	0	0	0
* →	0	1	1
* →	1	0	1
A=1 →	1	1	1

$A + A = A$

(considero i casi in cui $A_1 = A_2$)

$A + \bar{A} = ?$ * **$A + \bar{A} = 1$**

AND

$A \cdot A = ?$

AND			u
A=0 →	0	0	0
* →	0	1	0
* →	1	0	0
A=1 →	1	1	1

$A \cdot A = A$

$A \cdot \bar{A} = ?$ → = 0

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

5/03/2015



A	B	C	y
0	0	0	1
0	0	1	1
1	0	0	1
others			0

$y = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot C + A\cdot\bar{B}\cdot\bar{C}$

Semplifichiamo l'espressione

- $\bar{A}\bar{B}(\bar{C}+C) + A\cdot\bar{B}\cdot\bar{C} = \bar{A}\cdot\bar{B}\cdot 1 + A\bar{B}\bar{C} = y$

- Oppure... (ricordando che $x+x=x$)

$y = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot C + A\cdot\bar{B}\cdot\bar{C}$

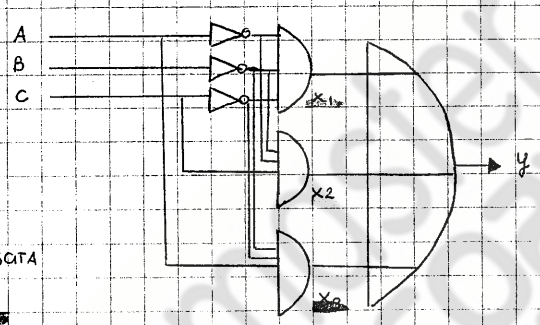
$= \bar{A}\bar{B}(\bar{C}+C) + \bar{B}\bar{C}(\bar{A}+A) = \bar{A}\bar{B} + \bar{B}\bar{C}$

($\bar{C}+C = 1$, $\bar{A}+A = 1$)

Glitch = Segnale non corretto in uscita da un circuito logico, causato anche da un ritardo nella propagazione dei segnali elettrici.

	A	B	C	y
x_1	0	0	0	1
x_2	0	0	1	1
x_3	1	0	0	1
others				0

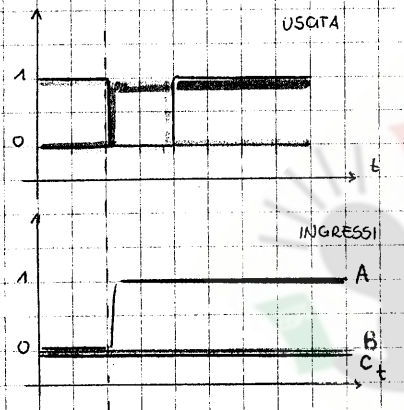
$y = \bar{A}\bar{B}\bar{C} + \bar{A}\cdot\bar{B}\cdot C + A\cdot\bar{B}\cdot\bar{C}$



Circuito che implementa la rete combinatoria; costruendo sulla base di essa un sistema fisico posso avere un ritardo a rispondere ai cambiamenti

↓

ciò avviene se un blocco reagisce prima dell'altro (sullo stesso livello)



Hp: A varia da $A_0 = 0$ ad $A_1 = 1$. Quindi, $\begin{cases} A_0 = 0 \\ B_0 = 0 \\ C_0 = 0 \end{cases} \Rightarrow \begin{cases} A_1 = 1 \\ B_1 = 0 \\ C_1 = 0 \end{cases}$

↓

La risposta mi aspetto che sia sempre 1, tuttavia se per esempio x_1 risponderà con un ritardo al cambiamento, ai "$t_{0,5}$" si avrebbe $y=0$ (causata da $x_{1\,0} = x_{1\,0,5} = 1$, $x_{2\,1} = x_{2\,0} = x_{2\,0,5} = 0$ e $x_{3\,1} = x_{3\,0,5} = 1$)

$\begin{cases} x_{1\,0} = 1 \\ x_{2\,0} = 0 \\ x_{3\,0} = 0 \end{cases}$ ↓ $y = 1$; $\begin{cases} x_{1\,1} = 0 \\ x_{2\,1} = 0 \\ x_{3\,1} = 1 \end{cases}$ ↓ $y = 1$

Affinché nonostante le diverse velocità di reazione la transizione non si verifichi è necessario un valore fisso a 1.

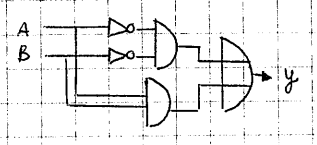
- Spesso è più facile implementare la espressione inversa rispetto a quella diretta ovvero quella per cui $u = 0$

Esempio XNOR

- XNOR = «NOR esclusivo» → Si comporta come NOR quando entrambe le variabili sono uguali:

A	B	y
0	0	1
0	1	0
1	0	0
1	1	1

$y = \bar{A}\cdot\bar{B} + A\cdot B$



$\overline{A \oplus B}$ = XNOR →

