

# Il Livello Logico Digitale (ii)

A cura di:

Luca Breveglieri      Giuseppe Pozzi

DEI, Politecnico di Milano  
luca.breveglieri/giuseppe.pozzi@polimi.it  
- versione del 4 aprile 2003 -

4-04.-03

Informatica II - Livello logico (2)

1

# Mappe di Karnaugh

4-04.-03

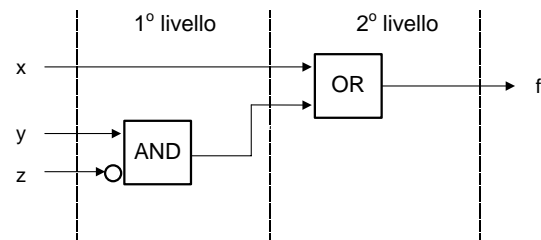
Informatica II - Livello logico (2)

2

## Rete a due livelli

- Rete a due livelli: contiene solo due livelli di operatori annidati (trascurando la negazione).

$$f(x, y, z) = x + y / z$$



4-04.-03

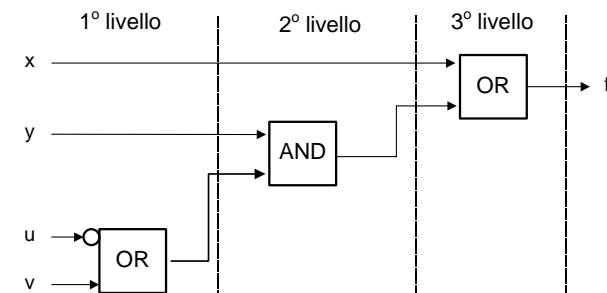
Informatica II - Livello logico (2)

3

## Rete multilivello

- Rete multilivello: contiene più livelli di operatori annidati (trascurando la negazione).

$$f(x, y, u, v) = x + y (/u + v) \quad (\text{a 3 livelli})$$



4-04.-03

Informatica II - Livello logico (2)

4

## Costo di una rete

- Criterio di costo "dei letterali" di una rete combinatoria a due livelli: numero delle variabili occorrenti nella funzione.

$$\text{costo}(x + y (/u + v)) = 4$$

$$\text{costo}(v + y (/u + v)) = 3$$

- Costo( $f(\bullet)$ ) = # di ingressi nel 1° livello della rete combinatoria associata alla funzione  $f()$ .
- Il criterio di costo dei letterali vale solo per le reti a due livelli, non per quelle multilivello!

4-04.-03

Informatica II - Livello logico (2)

5

## Equivalenza

- Due funzioni booleane  $f$  e  $g$ , a  $n \geq 1$  variabili, sono equivalenti se e solo se ammettono la stessa tabella delle verità.
- Le funzioni  $f$ ,  $g$  (a 3 variabili) sono equivalenti.

$$f(x, y, z) = x + y \cdot z, \quad g(x, y, z) = x + /x \cdot y \cdot z$$

x	y	z	f	g
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

4-04.-03

Informatica II - Livello logico (2)

6

## Sintesi minima

- Le reti combinatorie corrispondenti alle funzioni equivalenti  $f$  e  $g$  hanno pertanto lo stesso comportamento.
- Ma le reti combinatorie corrispondenti alle funzioni  $f$  e  $g$  sono diverse, e hanno costi diversi.
- Data una funzione booleana, esistono più (infinite) reti combinatorie che la realizzano.
- Sintetizzare la rete comb. di costo minimo!

4-04.-03

Informatica II - Livello logico (2)

7

## Forma canonica SOP

- Mintermine di ordine  $i$ : prodotto di variabili, che vale 1 solo nella  $j$ esima configurazione degli ingressi:

$$\bullet m_0 = /x \cdot /y \cdot /z \quad 0 = 000$$

$$\bullet m_3 = /x \cdot y \cdot z \quad 3 = 011$$

$$\bullet m_7 = x \cdot y \cdot z \quad 7 = 111$$

$$m_0(0, 0, 0) = /0 \cdot /0 \cdot /0 = 1 \cdot 1 \cdot 1 = 1$$

$$m_0(0, 1, 0) = /0 \cdot /1 \cdot /0 = 1 \cdot 0 \cdot 1 = 0$$

4-04.-03

Informatica II - Livello logico (2)

8

## Forma canonica SOP

$$f(x, y, z) = x \cdot /z + /x \cdot y \cdot /z + y \cdot z$$

N	x	y	z	f	mintermine
0	0	0	0	0	$/x \cdot /y \cdot /z$
1	0	0	1	0	$/x \cdot /y \cdot z$
2	0	1	0	1	$/x \cdot y \cdot /z$
3	0	1	1	1	$/x \cdot y \cdot z$
4	1	0	0	1	$x \cdot /y \cdot /z$
5	1	0	1	0	$x \cdot /y \cdot z$
6	1	1	0	1	$x \cdot y \cdot /z$
7	1	1	1	1	$x \cdot y \cdot z$

Tabella delle verità di f()

## Forma canonica SOP

- Forma canonica a 2 livelli SOP (Sum Of Products)

- $f(x, y, z) = m_2 + m_3 + m_4 + m_6 + m_7$

- $f(x, y, z) = /x \cdot y \cdot /z + /x \cdot y \cdot z + x \cdot /y \cdot /z + x \cdot y \cdot /z + x \cdot y \cdot z$

- $f(x, y, z) = \Sigma (2, 3, 4, 6, 7)$

- costo = 3 + 3 + 3 + 3 + 3 = 15

## Minimizzazione algebrica del costo

- Sia data la funzione seguente

$$f(x, y) = xy + /y \quad \text{costo} = 2 + 1 = 3$$

- Forma canonica SOP

$$f(x, y) = /x/y + x/y + xy \quad \text{costo} = 2+2+2 = 6$$

- Si vede che la forma canonica non è certamente quella di costo minimo.

## Minimizzazione algebrica del costo

### Equazione

### Proprietà

$$/x/y + x/y + xy =$$

idempotenza

$$= /x/y + x/y + x/y + xy =$$

distrib. prod. vs. som.

$$= (/x + x)/y + x(/y + y) =$$

elementi inversi

$$= 1/y + x1 =$$

elementi neutri

$$= /y + x =$$

commutativa

$$= x + /y$$

costo = 1 + 1 = 2

## Mappa di Karnaugh

- Sia data la funzione seguente

$$f(x, y) = xy + /y \quad \text{costo} = 2 + 1 = 3$$

m	x	y	f
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

Tabella delle verità

	y	
x	0	1
0	1	
1	1	1

Mappa di Karnaugh SOP

$$f(x, y) = x + /y \quad \text{costo} = 1 + 1 = 2$$

4-04.-03

Informatica II - Livello logico (2)

13

## Mappa di Karnaugh

$$f(x, y, z) = xy + x/z + /x/y/z$$

m	x	y	z	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Tabella delle verità

4-04.-03

Informatica II - Livello logico (2)

14

## Mappa di Karnaugh

Mappa di Karnaugh SOP

	yz			
x	00	01	11	10
0	1			
1	1		1	1

$$f(x, y, z) = xy + x/z + /x/y/z = xy + /y/z$$

$$\text{costo} = 2 + 2 = 4$$

4-04.-03

Informatica II - Livello logico (2)

15

## Implicanti primi

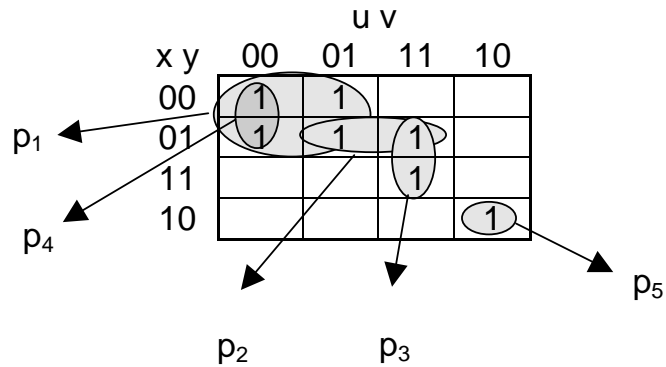
- Implicante:** gruppo di 1 "ben formato":
  - Numero di 1 pari a una potenza di 2 (1, 2, 4, 8, ecc).
  - Gli 1 sono disposti in modo opportuno (vedere esempi).
- Implicante primo:** implicante massimale, non contenuto in nessun altro implicante.

4-04.-03

Informatica II - Livello logico (2)

16

## Implicanti primi



## Implicanti primi

- $p_1 = /x./u$  è implicante primo
- $p_2 = /x.y.v$  è implicante primo
- $p_3 = y.u.v$  è implicante primo
- $p_4 = /x./u./v$  è implicante (ma non primo, perché  $\subset$  in  $p_1$ )
- $p_5 = x./y.u./v$  è implicante primo (ed è anche mintermine)

4-04.-03

Informatica II - Livello logico (2)

17

4-04.-03

Informatica II - Livello logico (2)

18

## Algoritmo

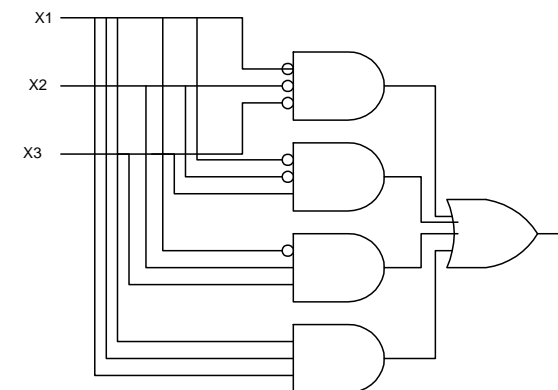
- **Raccolta degli implicanti primi:** Determinare tutti e soli gli implicanti primi della funzione combinatoria, tramite le mappe di Karnaugh.
- **Copertura della funzione:** Effettuare una scelta di implicanti primi che:
  - coprano tutti i mintermini della funzione
  - siano di costo minimo (secondo il criterio di costo dei letterali)

4-04.-03

Informatica II - Livello logico (2)

19

## Esempio di rete combinatoria



4-04.-03

Informatica II - Livello logico (2)

20

## Circuiti integrati e blocchi funzionali combinatori

4-04.-03

Informatica II - Livello logico (2)

21

## Circuiti integrati

- Le porte logiche non vengono prodotte isolatamente, ma sono realizzate su circuiti integrati (IC, Integrated Circuits)
- Un circuito integrato è una piastrina di silicio (o chip), quadrata o rettangolare, sulla cui superficie vengono realizzati e collegati transistor e dunque porte logiche, che complessivamente realizzano uno o più circuiti digitali

4-04.-03

Informatica II - Livello logico (2)

22

## Famiglie di circuiti integrati

- I circuiti integrati sono classificati in base alle loro dimensioni, cioè al numero di porte logiche contenute:
  - SSI (Small Scale Integrated): IC a scala di integrazione piccola, da 1 a 10 porte
  - MSI (Medium Scale Integrated): IC a scala di integrazione media, da 10 a 100 porte
  - LSI (Large Scale Integrated): IC a scala di integrazione grande, da 100 a 100.000 porte

4-04.-03

Informatica II - Livello logico (2)

23

## Famiglie di circuiti integrati

- VLSI (Very Large Scale Integrated): IC a scala di integrazione molto grande, > 100.000 porte
- Ogni famiglia ha degli usi caratteristici nei calcolatori e in generale nei dispositivi elettronici, che dipendono dalle sue dimensioni, ovvero dalla quantità di porte presenti sul circuito integrato stesso

4-04.-03

Informatica II - Livello logico (2)

24

## Usi caratteristici

- SSI: piccoli circuiti digitali di contorno a circuiti più grandi e impegnativi (glue logic, logica di "incollamento")
- MSI: circuiti digitali semplici, dotati di un'unica funzione ben definita, p. es.:
  - scambio di due o più segnali
  - confronto di due numeri
  - addizione o sottrazione di due numeri
  - e così via ...

4-04.-03

Informatica II - Livello logico (2)

25

## Usi caratteristici

- LSI: circuiti digitali complessi, dotati di funzionalità multiple, eventualmente programmabili, p. es.:
  - un intero insieme di op. aritmetiche: addizione, sottrazione, moltiplicazione, divisione, ecc (ALU)
  - memorie di piccole dimensioni
  - processori semplici
  - e così via ...

4-04.-03

Informatica II - Livello logico (2)

26

## Usi caratteristici

- VLSI: circuiti digitali molto complessi o di grandi dimensioni, spesso programmabili, p. es.:
  - Processori, a partire dai più semplici microcontrollori fino ai processori di uso generale (Pentium, SPARC, ecc)
  - Memorie, da qualche Kbyte in su
  - Unità di controllo delle periferiche
  - e così via ...
- Oggi (2000): max  $10^7$  transistor per IC

4-04.-03

Informatica II - Livello logico (2)

27

## Metodi di progetto logico

- Nessuno progetta un circuito integrato contenente 10 milioni di transistor, equivalenti a circa 2 milioni di porte logiche AND a 2 ingressi, trattandolo come un'unica rete combinatoria (o sequenziale) di dimensioni enormi!
- Per progettare circuiti digitali di tali dimensioni, si usano tecniche modulari, per scomporre il problema

4-04.-03

Informatica II - Livello logico (2)

28

## Matrici logiche programmabili

- Esistono circuiti integrati programmabili, chiamati PLA (Programmable Logic Array, matrici logiche programmabili), che permettono di realizzare qualsiasi rete combinatoria
- Il solo limite della PLA è dato da:
  - Il numero di piedini (che è fisso)
  - Il numero massimo di porte logiche disponibili sulla PLA

4-04.-03

Informatica II - Livello logico (2)

29

## Come si usa la PLA

- Il produttore produce e commercializza PLA "vergini": esse non hanno a bordo alcun circuito digitale definito
- Spetta al compratore programmare la PLA "vergine", installandole a bordo una o più reti combinatorie (o anche sequenziali), secondo le esigenze
- Per programmare la PLA occorre un apposito apparato programmatore

4-04.-03

Informatica II - Livello logico (2)

30

## Come funziona la PLA

- La PLA "vergine" contiene già un numero fissato di porte logiche AND, OR e NOT (o anche NAND, ecc)
- Nella PLA "vergine" i collegamenti tra queste porte logiche sono però indefiniti
- La PLA "vergine" contiene delle matrici di microinterruttori a transistor: bruciandoli o lasciandoli intatti si realizzano collegamenti tra le porte

4-04.-03

Informatica II - Livello logico (2)

31

## Come funziona la PLA

- L'apparato programmatore di PLA è in grado di bruciare selettivamente i microinterruttori presenti sulla PLA
- Per farlo, esso applica ad alcuni piedini della PLA (piedini di programmazione), speciali valori di tensione elettrica, che agiscono sui microinterruttori
- L'operazione è del tutto automatica e relativamente veloce, e irreversibile

4-04.-03

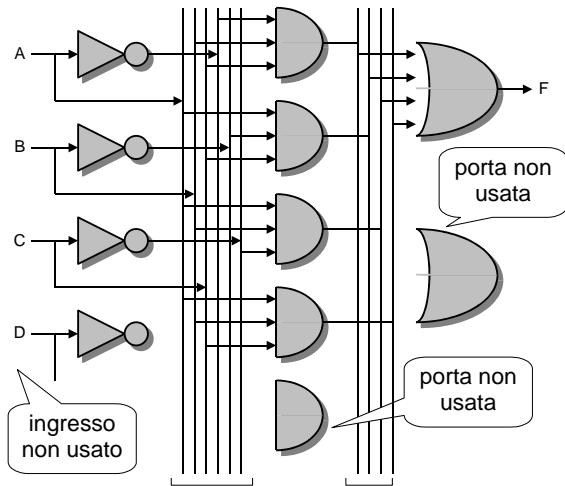
Informatica II - Livello logico (2)

32



## Esempio di PLA

PLA programmata: funzione maggioranza a 3 ingressi



4-04.-03

Informatica II - Livello logico (2)

33

Animazione

Fine

## Blocchi funzionali combinatori

- Esiste una ben nota e ormai stabilizzata libreria di blocchi funzionali predefiniti di tipo combinatorio
- Essa contiene blocchi funzionali per tutte le funzioni combinatorie di base
- Questi blocchi appartengono alle famiglie MSI e (alcuni di essi) LSI
- La libreria contiene anche blocchi funzionali di tipo sequenziale

4-04.-03

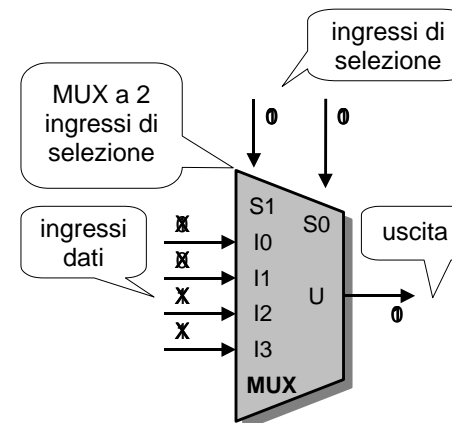
Informatica II - Livello logico (2)

34

## Multiplicatore (multiplexer)

- Il blocco funzionale multiplicatore ha:
  - $n \geq 1$  ingressi di selezione
  - $2^n \geq 2$  ingressi dati
  - un'uscita
- Gli ingressi dati sono numerati a partire da 0:  $k = 0, 1, 2, \dots, 2^n - 1$
- Se sugli ingressi di selezione è presente il numero binario  $k$ , il  $k^{\text{esimo}}$  ingresso dati viene inviato in uscita

## Multiplicatore



Multiplicatore a 2 ingressi di selezione

## Tabella delle verità

# riga	S1	S0	I0	I1	I2	I3	U
0	0	0	0	X	X	X	0
1	0	0	1	X	X	X	1
2	0	1	X	0	X	X	0
3	0	1	X	1	X	X	1
4	1	0	X	X	0	X	0
5	1	0	X	X	1	X	1
6	1	1	X	X	X	0	0
7	1	1	X	X	X	1	1

4-04.-03

Informatica II - Livello logico (2)

35

Animazione

Fine

4-04.-03

Informatica II - Livello logico (2)

36

# Decodificatore (decoder)

- Il blocco funzionale decodificatore ha:
  - $n \geq 1$  ingressi
  - $2^n \geq 2$  uscite
- Le uscite sono numerate a partire da 0:  $k = 0, 1, 2, \dots, 2^n - 1$
- Se sugli ingressi è presente il numero binario  $k$ , la  $k^{\text{esima}}$  uscita assume il valore 1 e le restanti uscite assumono il valore 0

# Decodificatore

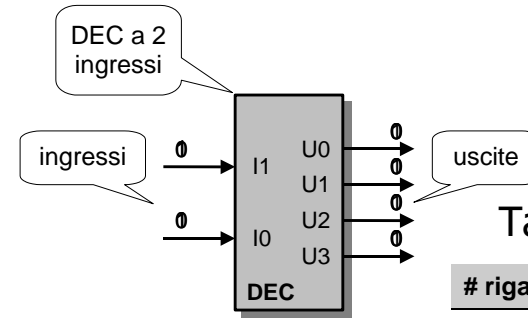


Tabella delle verità

# riga	I1	I0	U0	U1	U2	U3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

# Confrontatore (comparator)

- Il blocco funzionale confrontatore ha:
  - due gruppi A e B di ingressi da  $n \geq 1$  bit ciascuno
  - tre uscite: minoranza  $A < B$ , uguaglianza  $A = B$  e maggioranza  $A > B$
- Il blocco confronta i due numeri binari A e B da n bit presenti sui due gruppi di ingressi, e attiva (a 1) l'uscita corrispondente all'esito del confronto

# Confrontatore

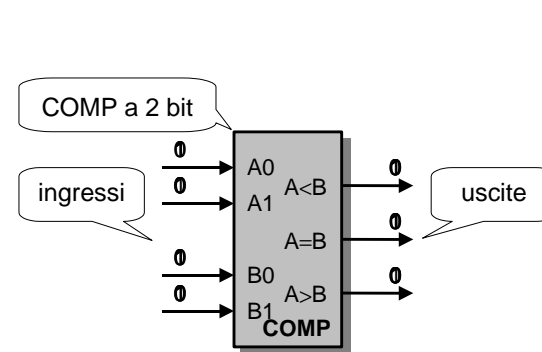


Tabella delle verità

# riga	A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	1
10	1	0	1	0	0	1	0
11	1	0	1	1	1	0	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	0	1
14	1	1	1	0	0	0	1
15	1	1	1	1	0	1	0

## Sommatore intero binario naturale a n bit (n-bit adder)

- È la generalizzazione del sommatore completo: addizione di numeri interi binari naturali (positivi) a n bit
- Ha in ingresso due numeri interi binari naturali A e B da  $n \geq 1$  bit ciascuno
- In uscita presenta la somma a n bit dei due numeri interi A e B
- Può avere un riporto in ingresso e un riporto in uscita, non sempre usati

4-04.-03

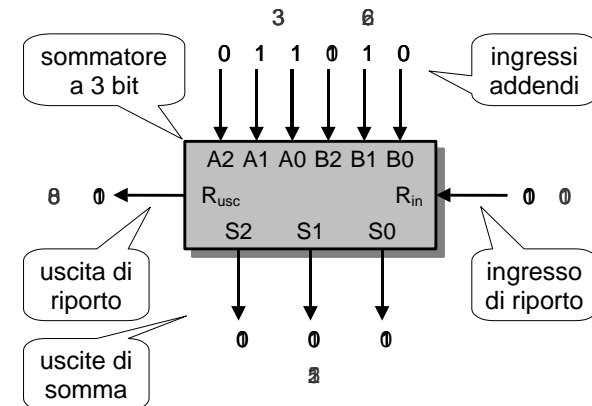
Informatica II - Livello logico (2)

41

Animazione

Fine

## Sommatore intero a n bit



Sommatore intero binario naturale a 3 bit

4-04.-03

Informatica II - Livello logico (2)

42

## Sottrattore intero binario naturale a n bit (n-bit subtractor)

- È il blocco funzionale inverso del sommatore intero a n bit
- Ha in ingresso due numeri interi binari naturali A e B da  $n \geq 1$  bit ciascuno
- In uscita presenta la differenza  $A - B$  a n bit dei due numeri interi A e B
- Può avere un prestito in ingresso e un prestito in uscita, non sempre usati

4-04.-03

Informatica II - Livello logico (2)

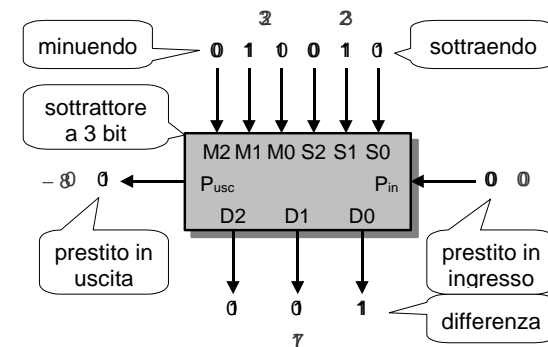
43

Animazione

Fine

## Sottrattore intero a n bit

Sottrattore intero binario naturale a 3 bit



$$M = M_2 M_1 M_0 = 100, S = S_2 S_1 S_0 = 010, P_{in} = P_{in} = 0$$

$$D = D_2 D_1 D_0 = 001, P_{usc} = P_{usc} = 0$$

4-04.-03

Informatica II - Livello logico (2)

44

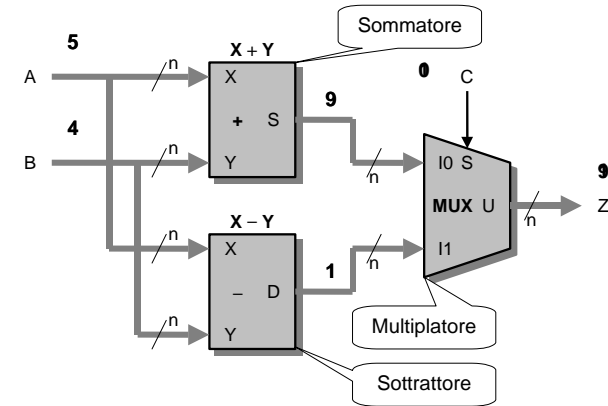
# Semplice esempio di progetto in stile funzionale

- Si chiede di progettare un circuito digitale combinatorio, che abbia:
  - in ingresso due numeri interi binari naturali (positivi) A e B da  $n \geq 1$  bit ciascuno
  - in ingresso un segnale di comando C
  - in uscita un numero intero binario naturale Z da  $n \geq 1$  bit
- Su Z deve uscire la somma  $A + B$  se  $C = 0$ , la differenza  $A - B$  se  $C = 1$

# Schema logico della soluzione

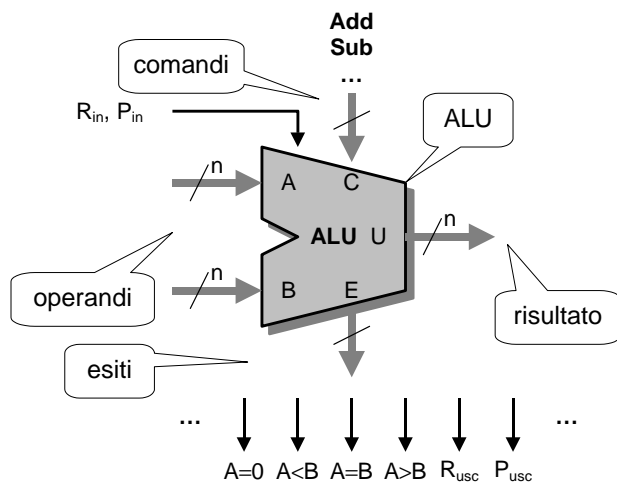
Animazione

Fine



(si usa un moltiplicatore a 2 gruppi di ingressi dati; ciascun gruppo è da n bit)

# Unità Aritmetico-Logica



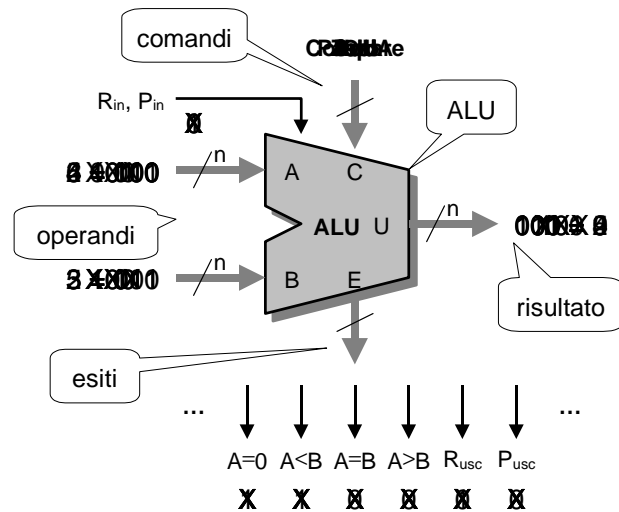
# Unità Aritmetico-Logica

# riga	Comando	Operazione	R	Esito
0	Add	somma A e B	$A + B + R_{in}$	riporto in uscita $R_{usc}$
1	Sub	sottrae B da A	$A - B - P_{in}$	prestito in uscita $P_{usc}$
2	Pass A	A passa in uscita	A	-
3	Pass B	B passa in uscita	B	-
4	Zero	annulla uscita	0	-
5	Shift Left A	A scorre a SX	$2A$	bit più significativo di A
6	Shift Right A	A scorre a DX	$A / 2$	bit meno significativo di A
7	Null	Confronta A con 0	-	$A = 0$
8	Compare	Confronta A con B	-	$A < B, A = B, A > B$
9	Multiply	prodotto di A e B	$A \times B$	riporto in uscita
10	Divide	divisione A / B	$A / B$	divisione per 0 ?
...	...	...	...	...

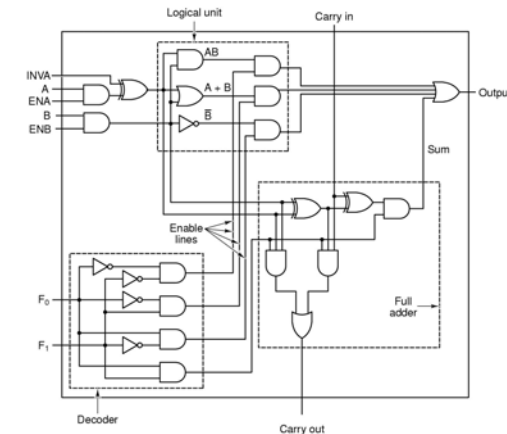
# Esempio di calcolo

Animazione

Fine



# Unità Aritmetico-Logica



Schema logico di una ALU da 1 bit

## ALU: numeri relativi

- I numeri relativi sono rappresentabili tramite sequenze di bit, proprio come i numeri interi naturali (sempre positivi)
- La tecnica più usata per rappresentare i numeri interi relativi è il complemento a due (two's complement)
- Le ALU sono normalmente in grado di operare sia con numeri interi naturali sia con numeri interi relativi rappresentati in complemento a due

## ALU: numeri reali

- I numeri reali sono rappresentabili tramite sequenze di bit, proprio come i numeri interi
- Esiste uno standard internazionale per la rappresentazione binaria di numeri reali: lo standard IEEE 754
- Esistono ALU in grado di effettuare i calcoli aritmetici con i numeri reali, oltre che con i numeri interi: addizione, ecc

## Il clock e i bistabili

4-04.-03

Informatica II - Livello logico (2)

53

## Il segnale di clock

- In molti circuiti digitali l'ordine temporale con cui si verificano gli eventi è una caratteristica significativa
- Dati due eventi E1 ed E2, potrebbe essere necessario fare in modo che:
  - E1 preceda E2
  - E1 sia simultaneo a E2
  - E1 segua E2

4-04.-03

Informatica II - Livello logico (2)

54

## Il segnale di clock

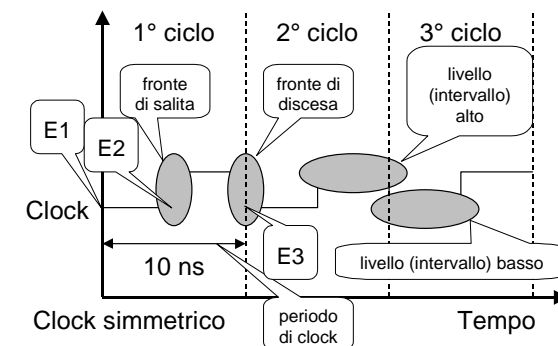
- Per definire e scandire il tempo, nei circuiti digitali si deve introdurre il segnale di clock (o di temporizzazione)
- Il clock è un segnale binario, con andamento periodico nel tempo
- Il segnale di clock è una successione di impulsi
- Ogni impulso ha una larghezza costante e due impulsi consecutivi stanno a una distanza costante

4-04.-03

Informatica II - Livello logico (2)

55

## Struttura e nomenclatura



$$\text{Frequenza di clock} = 1 / \text{periodo di clock} = 1 / 10 \text{ ns} = 100 \text{ MHz}$$

Il ciclo di clock contiene 3 eventi (E1, E2, E3)

4-04.-03

Informatica II - Livello logico (2)

56

## Memoria

- Le reti combinatorie sono prive di memoria: l'uscita di una rete combinatoria dipende solo dai valori logici applicati correntemente agli ingressi della rete
- Per eseguire certe elaborazioni, anche molto semplici, occorre conservare memoria di eventi (cioè di segnali binari) passati

4-04.-03

Informatica II - Livello logico (2)

57

## Esempio

- Progettare un circuito digitale dotato di due ingressi S e R, e di un'uscita Q. Il circuito:
  - manda a 1 l'uscita se  $S = 1$  (S è attivo)
  - manda a 0 l'uscita se  $R = 1$  (R è attivo)
  - altrimenti, l'uscita Q non subisce variazioni
  - nota: non accade mai che  $S = R = 1$
- Domanda: cosa vale l'uscita quanto  $S = R = 0$ , cioè quando entrambi gli ingressi S e R sono inattivi?

4-04.-03

Informatica II - Livello logico (2)

58

## Risposta alla domanda

- L'uscita Q potrebbe valere sia 1 che 0: il valore dell'uscita Q dipende dalla successione di eventi verificatisi su S e R in precedenza! Infatti:
  - se  $R = 0$  e  $S = 1$ , certamente  $Q = 1$ ; se poi S va a 0 l'uscita Q resta a 1
  - se  $R = 1$  e  $S = 0$ , certamente  $Q = 0$ ; se poi R va a 0 l'uscita Q resta a 0
- Conclusione: quando  $S = R = 0$  l'uscita Q potrebbe valere 1 oppure 0

4-04.-03

Informatica II - Livello logico (2)

59

## Il diagramma temporale

- Un buon modo per visualizzare comportamenti di circuiti digitali che dipendono dal tempo e da eventi passati è il diagramma temporale
- Diagramma temporale: sistema di assi cartesiani, con
  - in ascissa il tempo (in istanti discreti)
  - in ordinata i vari segnali i cui valori logici si succedono al trascorrere del tempo

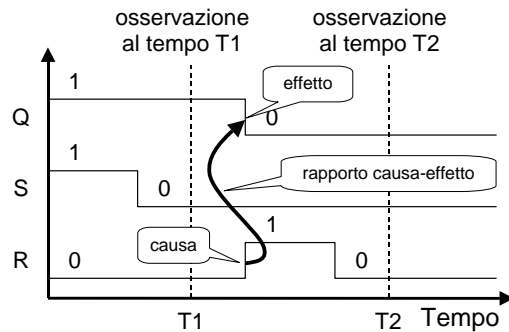
4-04.-03

Informatica II - Livello logico (2)

60

## Ancora l'esempio

Diagramma temporale del circuito precedente



- Al tempo T1 si ha:  $S = R = 0$  e  $Q = 1$
- Al tempo T2 si ha:  $S = R = 0$  e  $Q = 0$

## Commento

- Il diagramma temporale mostra che sapere che in ingresso si ha  $S, R = 0, 0$  non basta a determinare il valore dell'uscita  $Q$
- Il circuito digitale che risolve il problema di progetto appena descritto non può dunque essere di tipo combinatorio!

S	R	Q
0	0	0, 1
0	1	...
1	0	...
1	1	...

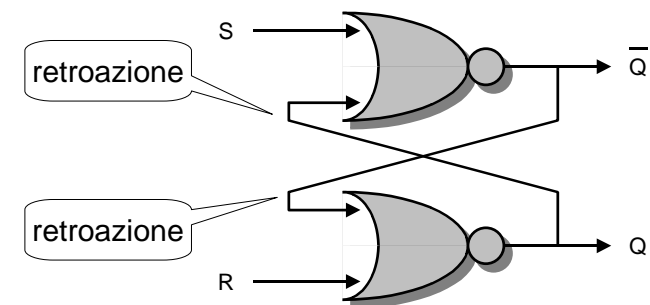
vietato!

## Come memorizzare un bit

- L'elemento funzionale elementare dotato di memoria è in grado di memorizzare il valore di un singolo bit
- Un modo (ma non l'unico) per realizzare tale elemento funzionale consiste nel collegare due porte logiche di tipo NOR (porta OR con uscita negata) in modo retroazionato (le reti combinatorie non contengono collegamenti retroazionati!)

## Come memorizzare un bit

Bistabile SR (SR latch, o SC latch)



- Il circuito ha due ingressi, S e R, e due uscite: Q e  $\bar{Q}$  (la versione negata di Q)



## Come funziona il bistabile SR

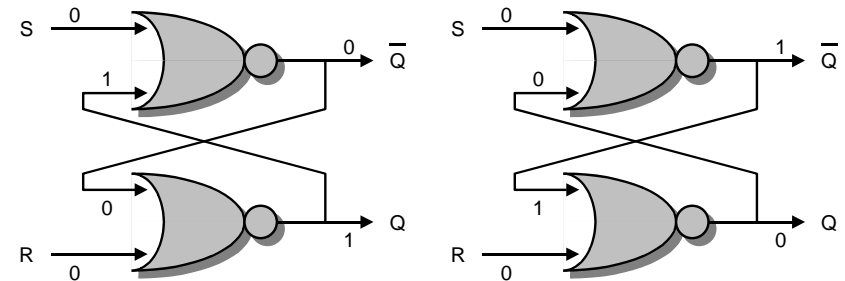
- Il bistabile SR ha un comportamento ben diverso da una rete combinatoria! Infatti:
  - se  $S = R = 0$ , le uscite  $Q$  e  $\bar{Q}$  possono valere 1 e 0, rispettivamente, ma ...
  - se  $S = R = 0$ , le uscite  $Q$  e  $\bar{Q}$  possono anche valere 0 e 1, rispettivamente
- Dunque, a parità di ingressi (cioè  $S = R = 0$ ) l'uscita  $Q$  ammette due possibili valori. Non è un comportamento da rete combinatoria!

4-04.-03

Informatica II - Livello logico (2)

65

## Come funziona il bistabile SR



$S = R = 0$  e  $Q = 1$   
memorizza il bit 1

$S = R = 0$  e  $Q = 0$   
memorizza il bit 0

Il circuito ha due stati di equilibrio (bistabile)

4-04.-03

Informatica II - Livello logico (2)

66

## Stati di memorizzazione

- Il bistabile SR è in grado di memorizzare due distinti valori logici:
  - se  $Q = 1$  il bistabile memorizza 1
  - se  $Q = 0$  il bistabile memorizza 0
- La capacità di memoria del bistabile SR è dunque di un bit (ovvero, esso memorizza l'uno o l'altro dei due possibili valori assumibili da un bit)

4-04.-03

Informatica II - Livello logico (2)

67

## Transizioni di stato

- Se il bistabile SR non potesse cambiare stato di memorizzazione, sarebbe un circuito digitale del tutto inutile per scopi pratici di memorizzazione
- È possibile fare transitare il bistabile SR dallo stato di memorizzazione 1 allo stato di memorizzazione 0, e viceversa, manovrando gli ingressi  $S$  e  $R$  in modo opportuno

4-04.-03

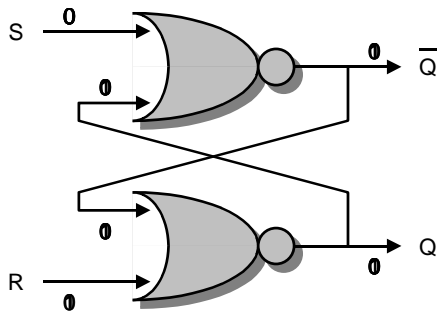
Informatica II - Livello logico (2)

68

## Transizione da 1 a 0

NOR

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

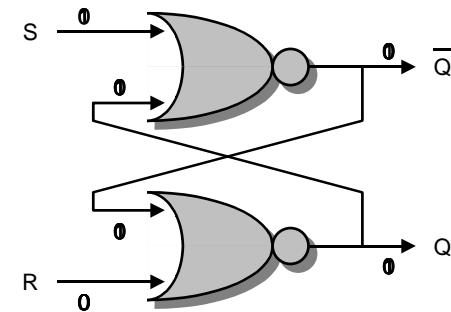


$S = R = 0$  e  $Q = 0$ ; ha memorizzato il bit 0

## Transizione da 0 a 1

NOR

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



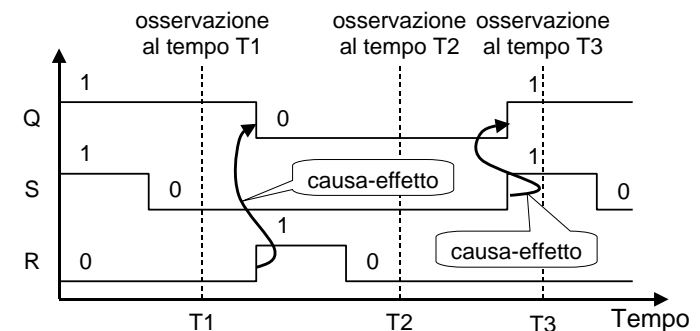
$S = R = 0$  e  $Q = 1$ ; ha memorizzato il bit 1

## Riassumendo

- Funzionamento del bistabile SR:
  - se  $S = R = 0$ , l'uscita Q mantiene memorizzato il valore logico di un bit (0 oppure 1)
  - se  $S = 1$  e  $R = 0$ , l'uscita Q assume il valore logico 1
  - se  $S = 0$  e  $R = 1$ , l'uscita Q assume il valore logico 0
  - è vietato applicare la configurazione di ingresso  $S = R = 1$  (in questa circostanza il comportamento del bistabile SR non è definito)

## Diagramma temporale

### Diagramma temporale del bistabile SR



Le frecce indicano un rapporto tra i fronti di tipo causa-effetto

## Dettagli

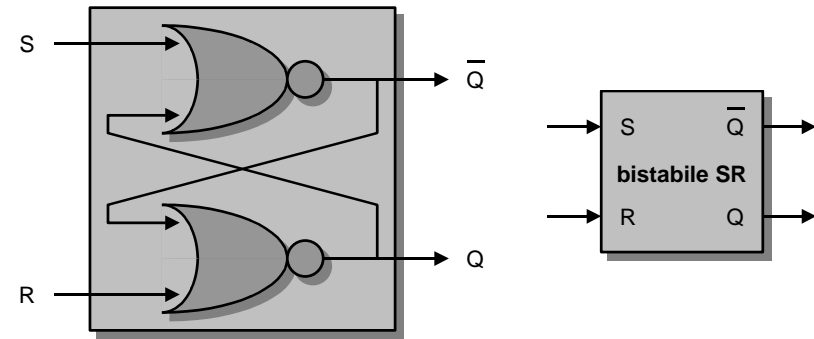
- Qualora si applicasse la configurazione (vietata) di ingresso  $S = R = 1$ , l'evoluzione successiva del bistabile diventerebbe imprevedibile:
  - se R tornasse a 0 prima (anche poco) di S, il bistabile transiterebbe nello stato 1
  - se S tornasse a 0 prima (anche poco) di R, il bistabile transiterebbe nello stato 0
  - ed è sostanzialmente impossibile fare commutare S e R simultaneamente

4-04.-03

Informatica II - Livello logico (2)

73

## Rappresentazione



Il bistabile SR (set-reset) come blocco funzionale SEQUENZIALE

4-04.-03

Informatica II - Livello logico (2)

74

## Usi caratteristici

- Il bistabile SR è un circuito digitale (sequenziale) fondamentale. Ha svariati usi caratteristici:
  - è l'elemento funzionale elementare per la costruzione di altri tipi di bistabili, più raffinati o specializzati
  - viene usato per memorizzare segnali binari
  - viene usato come adattatore, o interfaccia

4-04.-03

Informatica II - Livello logico (2)

75

## Esempio come adattatore

- Si supponga di avere una periferica che deve mandare un segnale di richiesta (p. es. di interruzione, interrupt) a un processore
- Ma la periferica genera solo un breve impulso di richiesta
- Mentre il processore potrebbe essere occupato e non in grado di rispondere subito alla richiesta, onorandola

4-04.-03

Informatica II - Livello logico (2)

76

## Funzionamento

- È dunque necessario interporre tra periferica e processore un circuito digitale adattatore (interfaccia), che:
  - riceva l'impulso di richiesta proveniente dalla periferica, lo memorizzi, stabilizzandolo, e lo mandi al processore
  - mantenga pendente la richiesta fintantoché il processore non sia disponibile a onorarla
  - cancelli la richiesta, quando il processore segnalasse di averla acquisita e di essere pronto a onorarla

4-04.-03

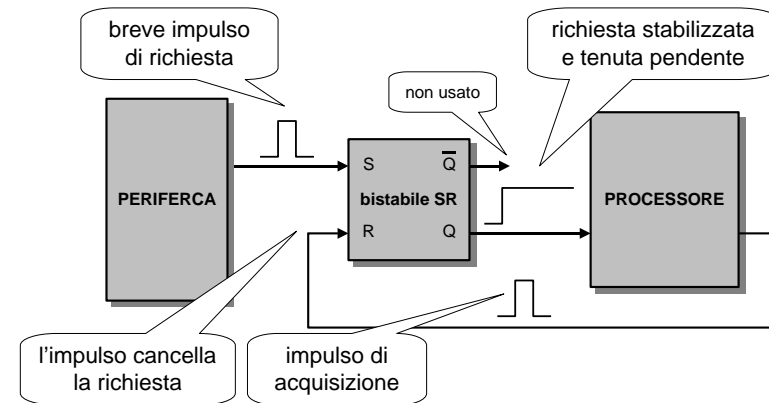
Informatica II - Livello logico (2)

77

## Schema logico

Animazione

Fine



Il bistabile SR funziona come adattatore, o interfaccia, tra periferica e processore

4-04.-03

Informatica II - Livello logico (2)

78

## Sincronizzazione

- In molte situazioni, è necessario impedire che lo stato di un bistabile possa cambiare, tranne in determinati istanti di tempo o intervalli di tempo
- Per ottenere questo scopo occorre:
  - disporre di un segnale di clock che scandisca gli istanti o intervalli di tempo in cui le transizioni di stato possono avvenire
  - sincronizzare il bistabile con il clock

4-04.-03

Informatica II - Livello logico (2)

79

## Bistabile D sincronizzato

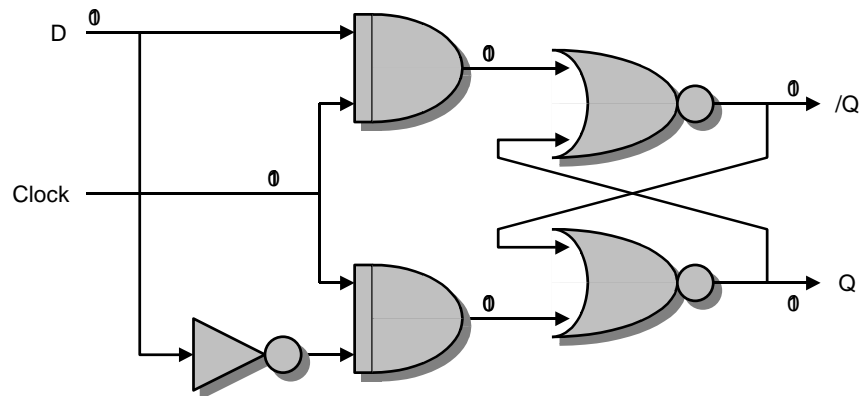
- Il bistabile SR ha due ingressi: S per portare a 1 lo stato, R per portarlo a 0
- Complessivamente comunque il bistabile SR memorizza un bit
- Molto spesso occorre un bistabile dotato di un solo ingresso, che si limiti a memorizzare il bit presente sul suo unico ingresso: il bistabile D sincronizzato (o D flip-flop)

4-04.-03

Informatica II - Livello logico (2)

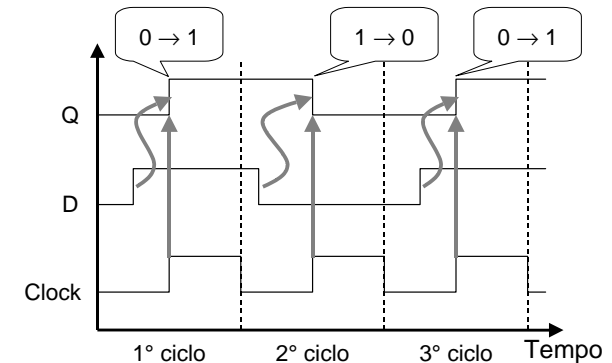
80

## Bistabile D sincronizzato



Se clock = 1 il bistabile memorizza l'ingresso D

## Diagramma temporale

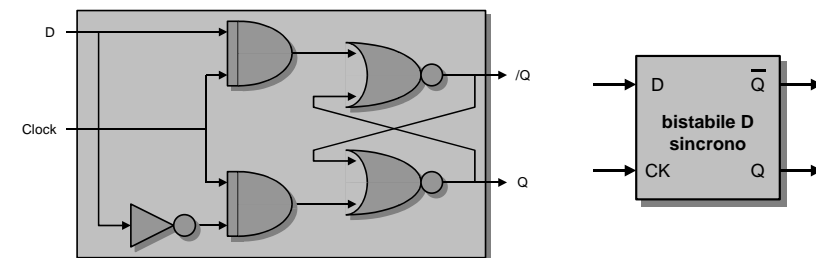


Le variazioni dell'uscita sono sincronizzate con il clock

## Bistabile D sincronizzato

- Nel bistabile D sincronizzato:
  - Se il clock vale 0, l'ingresso D non ha alcun effetto, e il bistabile mantiene memorizzato il suo stato corrente
  - Se il clock vale 1, l'ingresso D è efficace, e il bistabile memorizza il valore logico (0 oppure 1) presente sull'ingresso D
- Lo stato del bistabile può cambiare solo nell'intervallo alto del clock

## Rappresentazione



Il bistabile D sincrono come blocco funzionale SEQUENZIALE sincrono

## Comando di ripristino

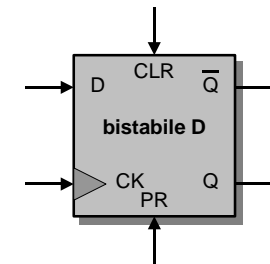
- Tutti i tipi di bistabili dispongono di varianti dotate di un comando di ripristino CLR (clear o reset), che forza lo stato del bistabile a 0
- Il comando di ripristino è molto utile per (re)inizializzare lo stato dei bistabili
- Alcuni bistabili dispongono anche del comando di precarica PR (preset), che forza lo stato del bistabile a 1

4-04.-03

Informatica II - Livello logico (2)

85

## Rappresentazione



Bistabile di tipo D, dotato di comandi di ripristino e di precarica. Di norma questi comandi sono asincroni, cioè agiscono immediatamente, non appena vengono attivati, senza attendere il clock

4-04.-03

Informatica II - Livello logico (2)

86

## Tempo

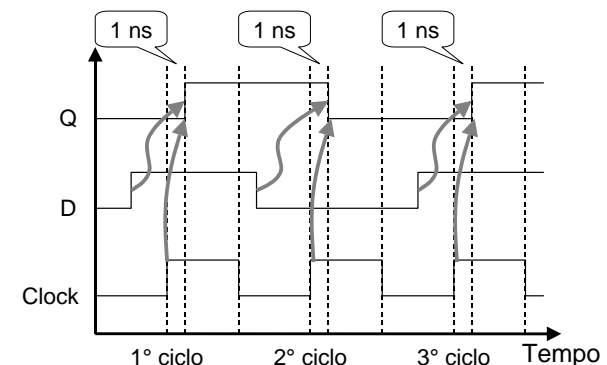
- Così come le porte logiche, anche i bistabili (sincronizzati o no) presentano un ritardo di commutazione dell'uscita
- La commutazione dell'uscita del bistabile avviene con un certo ritardo rispetto alla variazione degli ingressi o rispetto al fronte di clock che hanno indotto la transizione di stato
- Il ritardo di commutazione dipende dalla tecnologia microelettronica

4-04.-03

Informatica II - Livello logico (2)

87

## Diagramma temporale



Il bistabile sincrono di tipo D ha un ritardo di commutazione di 1 ns dell'uscita rispetto al fronte del clock

4-04.-03

Informatica II - Livello logico (2)

88

## Blocchi funzionali sequenziali

4-04.-03

Informatica II - Livello logico (2)

89

## Circuiti sequenziali

- Un circuito digitale è di tipo sequenziale se le sue uscite dipendono non solo dai valori correnti degli ingressi, ma anche da (alcuni di) quelli passati
- Un circuito digitale sequenziale (o rete sequenziale) è pertanto dotato, in ogni istante di tempo, di uno stato che, insieme ai valori degli ingressi, ne determina il comportamento futuro

4-04.-03

Informatica II - Livello logico (2)

90

## Circuiti sequenziali

- Lo stato di un circuito sequenziale rappresenta, in un certo senso, una forma di memoria: esso contiene una sorta di descrizione della storia passata del circuito stesso
- L'elemento funzionale elementare per la realizzazione di circuiti sequenziali è pertanto il bistabile, che è in grado di memorizzare un bit di informazione

4-04.-03

Informatica II - Livello logico (2)

91

## Circuiti sequenziali: struttura

- I circuiti sequenziali sono formati da:
  - bistabili, di vario tipo, che hanno funzioni di memorizzazione di informazioni, ovvero hanno la funzione di memorizzare dei bit
  - porte logiche, organizzate in reti combinatorie, che hanno funzioni di elaborazione di informazioni
- Il circuito sequenziale ha, in ogni istante, uno stato: i bit memorizzati nei bistabili facenti parte del circuito

4-04.-03

Informatica II - Livello logico (2)

92

## Circuiti sequenziali: famiglie

- Esistono due famiglie di circuiti digitali sequenziali:
  - asincroni, che non fanno uso di clock
  - sincroni, che necessitano di clock
- Un esempio di circuito sequenziale asincrono è il bistabile SR (non sincronizzato)
- I circuiti sequenziali asincroni, a parte pochi casi speciali, sono poco usati, perché difficili da usare e controllare

4-04.-03

Informatica II - Livello logico (2)

93

## Circuiti sequenziali sincroni

- I circuiti sequenziali sincroni sono invece universalmente diffusi
- Un esempio di circuito sequenziale sincrono è il bistabile di tipo D
- Il bistabile di tipo D è il costituente base dei circuiti sequenziali di tipo sincrono
- Circuito sequenziale (sincrono): bistabili di tipo D sincroni + reti combinatorie

4-04.-03

Informatica II - Livello logico (2)

94

## Blocchi funzionali sequenziali

- Come per le reti combinatorie, esistono delle procedure di sintesi anche per i circuiti sequenziali sincroni
- Tuttavia, la maggior parte dei circuiti sequenziali sincroni fondamentali è già disponibile sotto forma di blocchi funzionali precostituiti
- Tali blocchi funzionali fanno parte della libreria standard MSI (e in parte LSI)

4-04.-03

Informatica II - Livello logico (2)

95

## Il fenomeno di trasparenza

- Il bistabile D sincronizzato ha un comportamento sgradevole:
  - nell'intervallo alto del clock, l'ingresso D è efficace: le variazioni di D si propagano immediatamente (o quasi) all'uscita Q, la quale può dunque variare più volte; solo quando il clock torna a zero Q si stabilizza
  - è come se, nell'intervallo alto del clock, il bistabile non esercitasse alcuna funzione effettiva di memorizzazione
- Questo fenomeno si chiama *trasparenza*

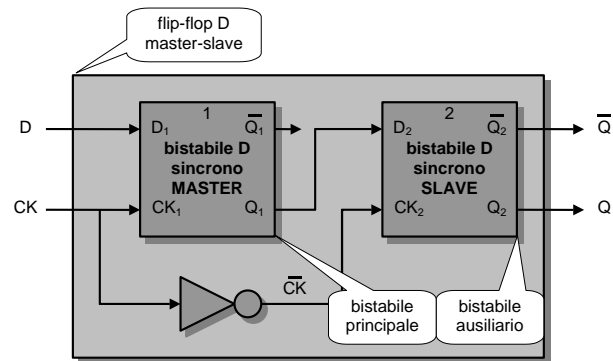
4-04.-03

Informatica II - Livello logico (2)

96



## Flip-flop D a memoria ausiliaria



Coppia di bistabili sincroni D trasparenti in cascata con clock invertiti; l'insieme dei due non presenta il fenomeno della trasparenza

4-04.-03

Informatica II - Livello logico (2)

97

## Funzionamento normale

- Il bistabile principale memorizza l'ingresso  $D = D_1$  durante l'intervallo alto del clock, lo emette sull'uscita  $Q_1$  e lo manda all'ingresso  $D_2$  del bistabile ausiliario
- Il bistabile ausiliario memorizza l'ingresso  $D_2$  durante l'intervallo basso del clock e lo emette sull'uscita  $Q_2 = Q$
- L'uscita generale  $Q$  può variare solo nell'istante del fronte di discesa del clock

4-04.-03

Informatica II - Livello logico (2)

98

## Come si evita la trasparenza

- Nell'intervallo basso del clock, il bistabile SLAVE è in stato di trasparenza
- Nell'intervallo alto del clock, il bistabile MASTER è in stato di trasparenza
- Se l'ingresso  $D$  varia durante l'intervallo alto del clock, il bistabile MASTER si comporta in modo trasparente
- Ma il bistabile SLAVE no, perché il suo clock si trova nell'intervallo basso

4-04.-03

Informatica II - Livello logico (2)

99

## Libreria di blocchi sequenziali

- Ecco una breve sintesi dei principali componenti sequenziali di libreria:
  - Registro parallelo
  - Registro a scorrimento
  - Banco di registri
  - Memoria
- Ognuno di questi blocchi ammette numerose versioni e varianti

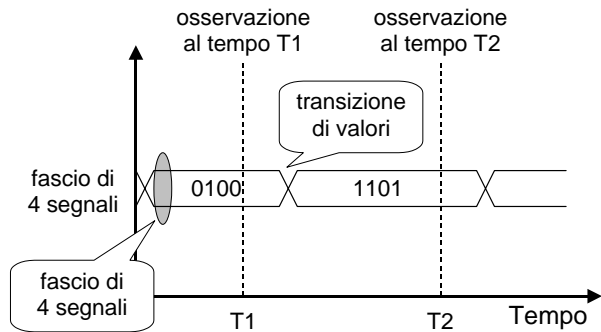
4-04.-03

Informatica II - Livello logico (2)

100

# Ancora diagramma temporale

Come rappresentare un fascio di segnali



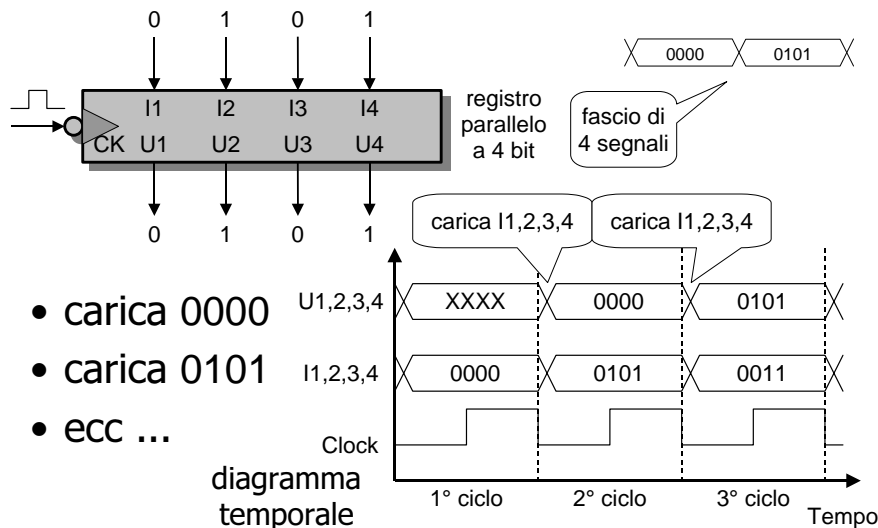
- Al tempo T1 i 4 segnali valgono 0100
- Al tempo T2 i 4 segnali valgono 1101

# Registro parallelo

- Il registro parallelo è un vettore di  $n \geq 1$  flip-flop di tipo D. Ha:
  - $n \geq 1$  ingressi  $I1, \dots, In$
  - $n \geq 1$  uscite  $U1, \dots, Un$
  - e naturalmente l'ingresso di clock CK
- A ogni ciclo di clock, il registro legge e memorizza nel suo stato la parola di  $n$  bit presente in ingresso, e la presenta sulle  $n$  uscite nel ciclo successivo

# Simbolo e funzionamento

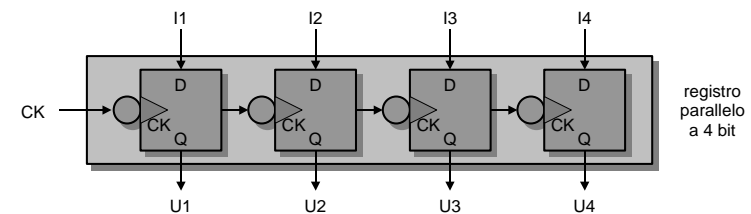
Animazione  
Fine



- carica 0000
- carica 0101
- ecc ...

diagramma temporale

# Progetto in stile funzionale



Registro parallelo progettato in stile funzionale, usando 4 bistabili D **sincroni sul fronte** (di discesa)

Nota bene: se si usassero dei bistabili D trasparenti (sincronizzati sul livello), durante il livello alto del clock il registro sarebbe esso stesso del tutto trasparente, e dunque non si comporterebbe come un registro ...

## Registro parallelo con comando di caricamento

- Funziona come il registro parallelo, ma ha in aggiunta un ingresso di comandi di caricamento (L, ingresso di Load):
  - Se il comando L è attivo (p. es. L = 1), la parola in ingresso al registro viene memorizzata nel registro stesso e presentata in uscita nel ciclo successivo
  - Altrimenti (cioè L = 0), il registro mantiene il suo stato corrente di memorizzazione

4-04.-03

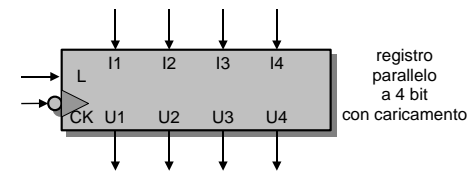
Informatica II - Livello logico (2)

105

## Simbolo e funzionamento

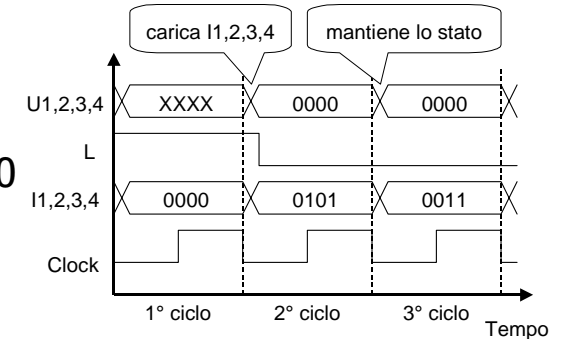
Animazione

Fine



registro parallelo a 4 bit con caricamento

- carica 0000
- mantiene 0000
- ecc ...

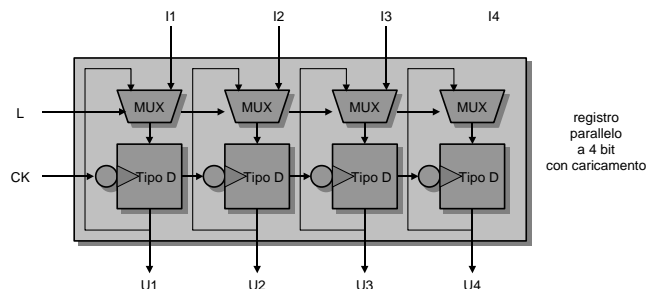


4-04.-03

Informatica II - Livello logico (2)

106

## Progetto in stile funzionale



Registro parallelo progettato in stile funzionale, usando 4 bistabili D **sincroni sul fronte** (di discesa) e 4 multiplatori a un ingresso di selezione e due ingressi dati

4-04.-03

Informatica II - Livello logico (2)

107

## Varianti e integrazioni

- Registro parallelo con comando di ripristino, per azzerare il contenuto
- Registro parallelo con comando di ripristino e di precarica
- Registro parallelo universale, riunisce le funzioni di tutti i registri precedenti: comando di caricamento, comando di ripristino e comando di precarica

4-04.-03

Informatica II - Livello logico (2)

108

## Registro a scorrimento

- Il registro a scorrimento è una successione di  $n \geq 1$  flip-flop di tipo D collegati in cascata. Ha:
  - un ingresso seriale S
  - $n \geq 1$  uscite parallele  $U_1, \dots, U_n$
  - e naturalmente l'ingresso di clock
- A ogni ciclo di clock, fa scorrere di un bit verso DX la parola memorizzata, perdendo il bit più a DX e aggiungendo il bit presente sull'ingresso seriale

4-04.-03

Informatica II - Livello logico (2)

109

## Simbolo e funzionamento

Animazione

Fine

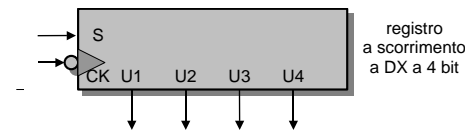
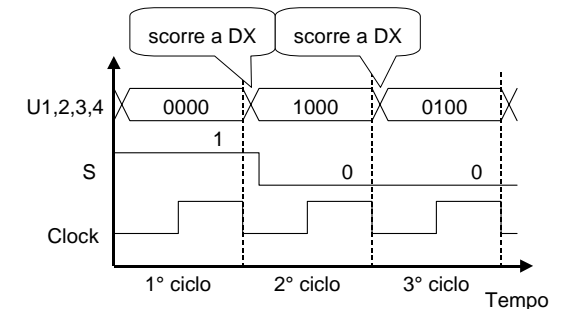


diagramma temporale



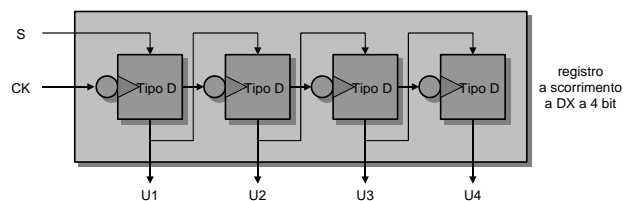
- scorre a DX
- scorre a DX
- ecc ...

4-04.-03

Informatica II - Livello logico (2)

110

## Progetto in stile funzionale



Registro a scorrimento a DX progettato in stile funzionale, usando 4 bistabili D sincroni **sul fronte** (di discesa) collegati in cascata

Nota bene: se si usassero dei bistabili D trasparenti (sincronizzati sul livello), durante il livello alto del clock un bit potrebbe propagarsi lungo l'intera catena di bistabili ... non sarebbe un comportamento accettabile!

4-04.-03

Informatica II - Livello logico (2)

111

## Varianti e integrazioni

- Registro a scorrimento a SX
- Registro a scorrimento universale: DX e SX (è dotato di un comando di scelta del verso di scorrimento)
- Registro a scorrimento (DX o SX) con funzione di caricamento parallelo
- Registro parallelo / a scorrimento universale: riunisce le funzioni dei registri parallelo e a scorrimento universali

4-04.-03

Informatica II - Livello logico (2)

112

## Semplice esempio di progetto in stile funzionale

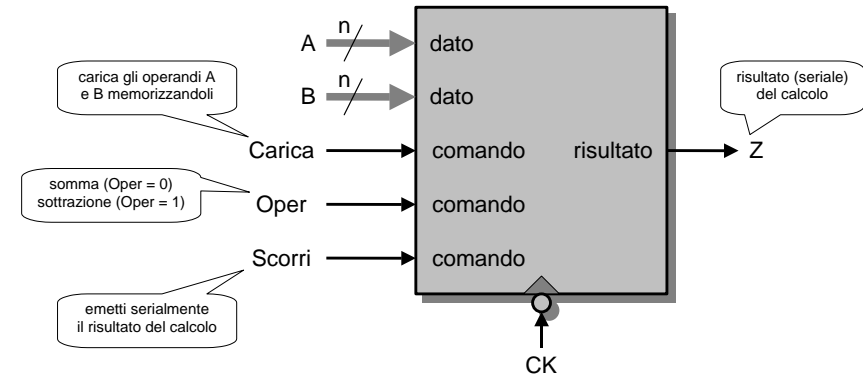
- Si chiede di progettare un circuito digitale sequenziale, che abbia:
  - in ingresso due numeri interi binari naturali (positivi) A e B da  $n \geq 1$  bit ciascuno
  - in ingresso 3 segnali di comando (vedere)
  - un'uscita seriale (da 1 bit) su cui emettere un numero intero binario naturale Z
- Naturalmente, il circuito ha in ingresso anche il segnale di clock

4-04.-03

Informatica II - Livello logico (2)

113

## Interfaccia del circuito



Gli ingressi di controllo Carica e Scorri sono attivi alti

4-04.-03

Informatica II - Livello logico (2)

114

## Specifiche di funzionamento

- Nel 1° ciclo di clock, i due numeri A e B vengono acquisiti e memorizzati
- Nel 2° ciclo di clock, i due numeri memorizzati vengono:
  - sommati se Oper = 0
  - sottratti se Oper = 1e il risultato viene memorizzato
- A partire dal 3° ciclo di clock, il risultato viene emesso serialmente (cioè un bit alla volta) in n cicli di clock (dal 3 a 3 + n - 1)

4-04.-03

Informatica II - Livello logico (2)

115

## Elenco dei componenti

- 2 registri paralleli a n bit, con comando di caricamento
- 1 addizionatore e 1 sottrattore intero binario naturale, a n bit
- 1 moltiplicatore a 1 ingresso di selezione e 2 gruppi di ingressi a n bit ciascuno
- 1 registro a scorrimento a n bit (unidirezionale), con comando di caricamento / scorrimento
- 1 porta logica NOT

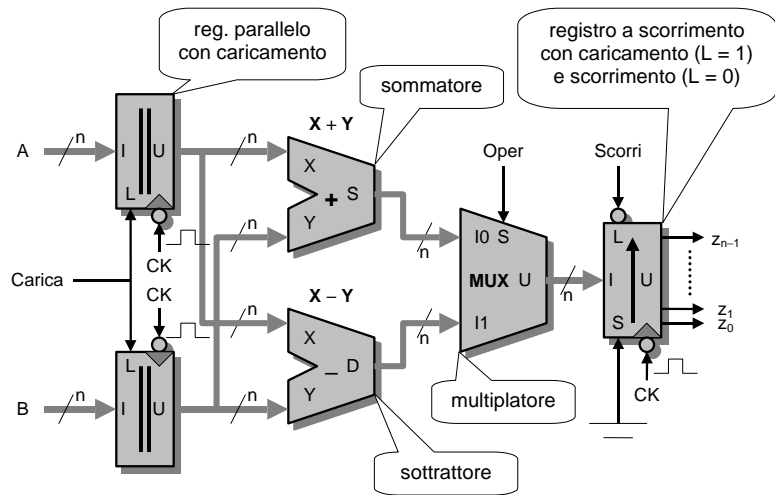
4-04.-03

Informatica II - Livello logico (2)

116

# Schema logico della soluzione

Animazione  
Fine



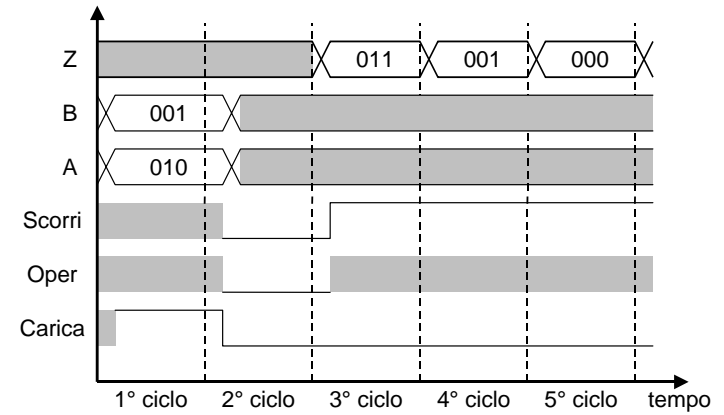
4-04.-03

Informatica II - Livello logico (2)

117

Ultima slide  
prima della fine  
della lezione

# Diagramma temporale



operazione di addizione di numeri a 3 bit

4-04.-03

Informatica II - Livello logico (2)

118

4-04.-03

Informatica II - Livello logico (2)

119

4-04.-03

Informatica II - Livello logico (2)

120