

Capitolo 3

Modelli

3.1 – Elaborazione di sequenze di simboli

3.2 – La macchina combinatoria

3.3 – La macchina asincrona

3.4 – La macchina sincrona

"ex-or"

L'interruttore "complessivo" è chiuso se sono alti o D1 = D2, ma non entrambi

Il gate "ex-or"

D1	D2	I
alto	alto	spenta
basso	alto	accesa
alto	basso	accesa
basso	basso	spenta

Due "nor" in retroazione

$V_2 = V_3 = L$
 $V_1 = V_2 = ?$
o H o L

Le due trascodifiche

ENCODER

trascod. da 1 su 4 a binario

x_1, x_2, x_3, x_4	y_1	y_2	y_3	y_4
0 0 0 0	0	0	0	0
0 0 0 1	0	0	1	0
0 0 1 0	0	1	0	0
0 1 0 0	1	0	0	0
1 0 0 0	1	0	0	0

DECODER

trascod. da binario a 1 su 4

y_1, y_2, y_3, y_4	x_1	x_2	x_3	x_4
0 0 0 0	0	0	0	0
0 0 0 1	0	0	0	1
0 0 1 0	0	0	1	0
0 1 0 0	0	1	0	0
1 0 0 0	1	0	0	0

La conversione P/S di un byte

Ingresso: $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$

Uscita: $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$

Stato: $000, 001, 010, 011, 100, 101, 110, 111$

Controller

comportamento

struttura

Il modello del "blocco" o "scatola nera"

Alfabeto d'ingresso

Alfabeto d'uscita

ingresso dei dati → **P** → uscita dei risultati

P ↔ relazione ingresso/uscita o di causa/effetto
↳ trasformazione
↳ temporizzazione

pochi modelli!

Regole "elementari" di composizione

a) in serie: $M_1 \rightarrow M_2 \rightarrow u$, $u = M_2(M_1(i))$, Funzione composta

b) in parallelo: $M_1 \rightarrow u_1$, $M_2 \rightarrow u_2$, $u = M_1(i)$, $u = M_2(i)$, Sistema di funzioni

c) in retroazione: $M_1 \rightarrow u$, $M_2 \rightarrow u$, Funzione ricorsiva

pochi componenti primitivi!

3.1 Elaborazione di sequenze di simboli

Digitale è sinonimo di discreto

La discretizzazione degli stimoli e delle risposte

$J(t)$: tutta l'informazione ricevuta fino a t

P

$u(t) = P(J(t))$

$i(t) \in I$
(alfabeto di ingresso)

$u(t) \in U$
(alfabeto di uscita)

Comunicazione: numero finito di segnali binari

0 0 0	0
1 0 0	0
0 1 0	0
1 1 0	0
0 0 1	0
0 0 1	1
0 1 1	1
1 1 1	1

2ⁿ config.

Insieme discreto

La discretizzazione del tempo

Evento che introduce informazione

- > modifica dell'ingresso
- > scadere di un intervallo di tempo

l'uscita attuale può dunque dipendere:

- dall'ingresso contemporaneo
- dagli ingressi precedenti
- dal trascorrere del tempo

Esempi

esempio 1: $u = \sqrt{i}$

dato: 9, 25, 49
risultato: 3, 5, 7

esempio 2: CPU (istruzione, operando) → risultato

Stato macchina: V (Verde), G (Giallo), R (Rosso)

La macchina a memoria finita

Indichiamo con $t_0, t_1, \dots, t_{n-1}, t_n$ una sequenza finita di istanti in cui si sono verificati degli eventi

l'uscita al generico istante t_n dipende

- > dalla sequenza di ingresso $i(t_0) \Rightarrow i(t_1) \Rightarrow \dots \Rightarrow i(t_{n-1}) \Rightarrow i(t_n)$
- $u(t_n) = P(\dots, i(t_0), i(t_1), \dots, i(t_{n-1}), i(t_n))$

e questo come si esprime??

e

- > dalla condizione iniziale della macchina $s(t_0)$.
- $u(t_n) = P(s(t_0), i(t_0), i(t_1), \dots, i(t_{n-1}), i(t_n))$

Stato iniziale di una macchina

$s(t_0) \in S$

Esempio: il percorso di un'auto dipende non solo dai comandi via dati con volante, freno, acceleratore, ma anche dalla benzina inizialmente nel serbatoio e dallo stato di usura delle gomme.

Esempio: Non basta caricare un orologio per avere l'ora esatta. L'ora indicata dipende infatti non solo dal n° di scatti che la molla ha dato alle lancette, ma anche dalla loro posizione iniziale.

Esempi: digitazione del PIN allo sportello Bancomat, posizione del bit in uscita dal convertitore P/S

Macchina sequenziale: memorie e funzione

Due soli tipi di componente!

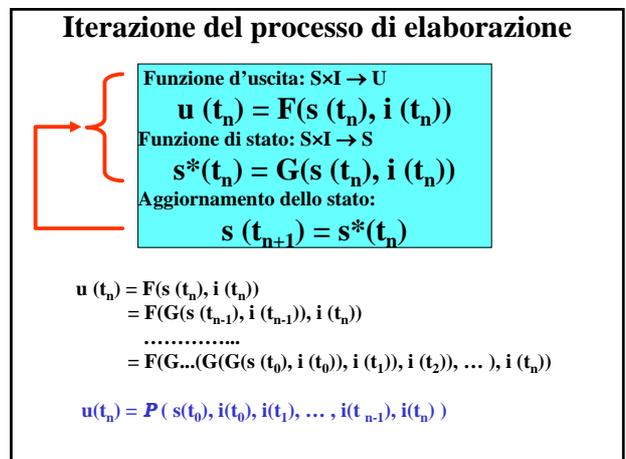
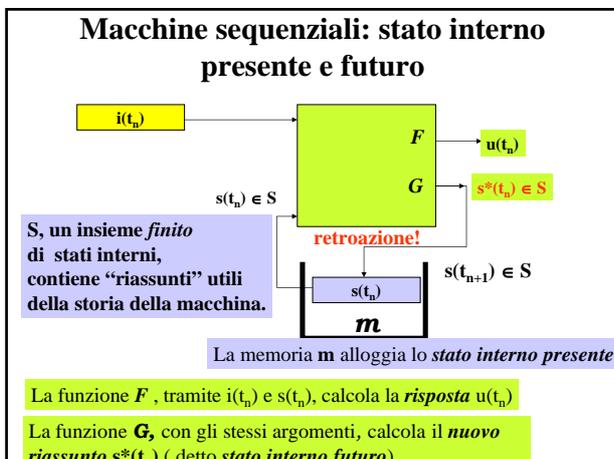
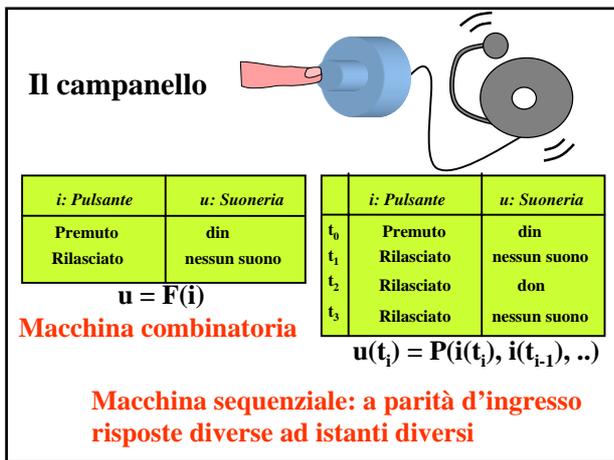
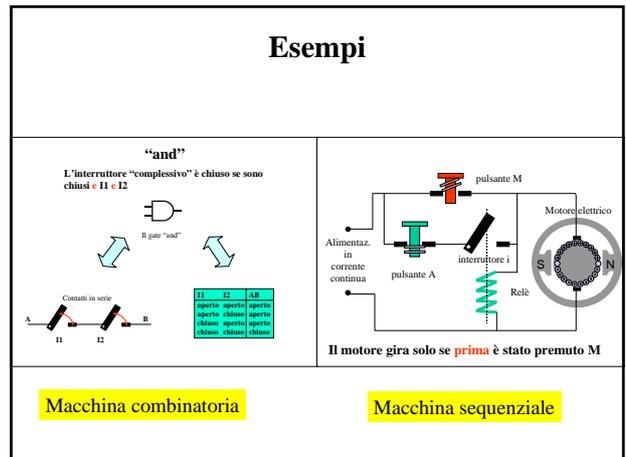
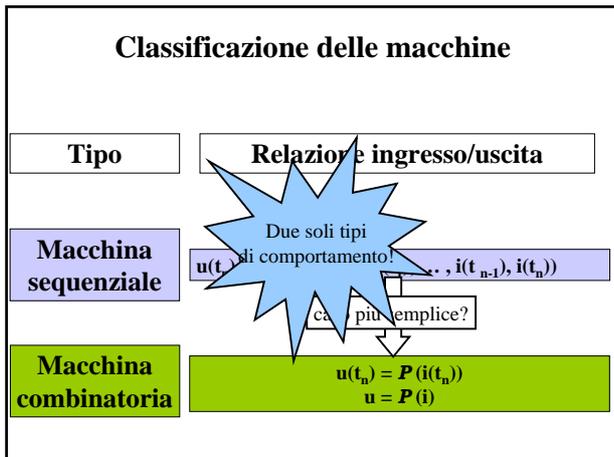
Il modello può essere utilmente impiegato solo quando l'uscita dipende da un numero limitato di eventi (capacità della **memoria** M, n° di ingressi della **funzione** P) e da uno stato iniziale prefissato.

Un altro esempio: il Bancomat

Compilazione e interpretazione di un linguaggio di alto livello

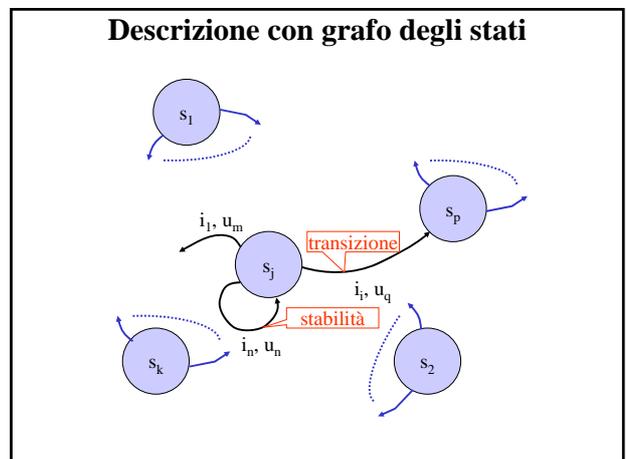
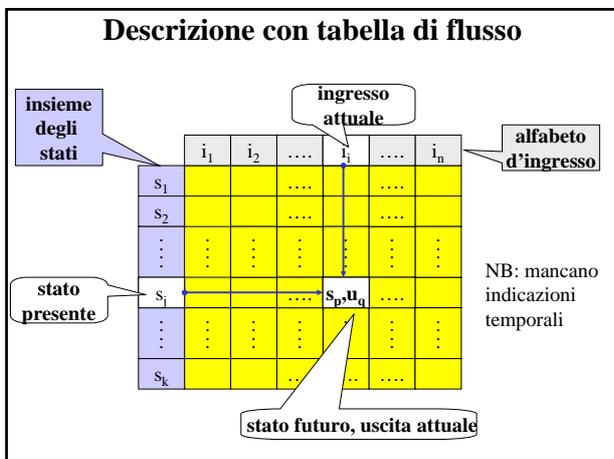
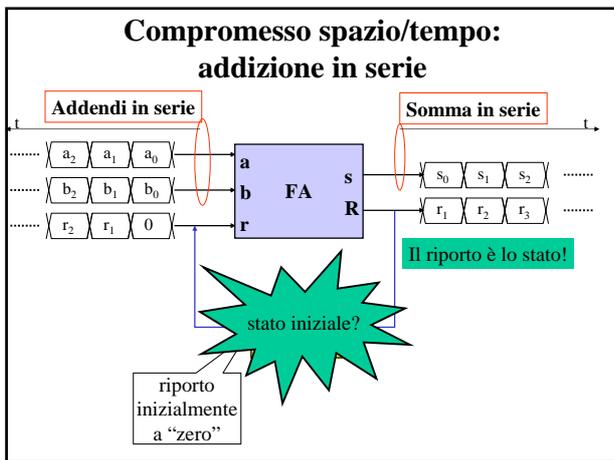
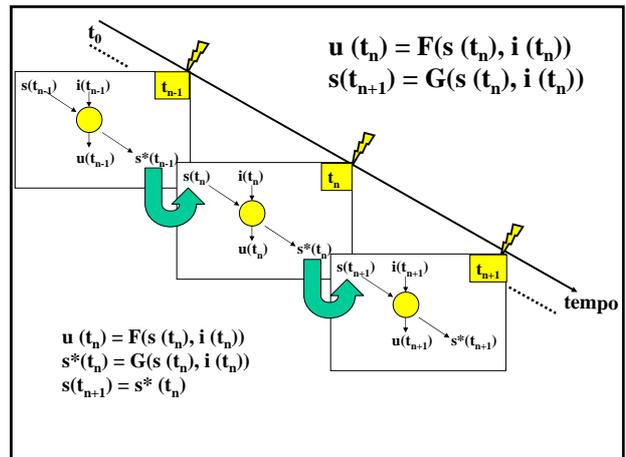
.. bla,bla ...

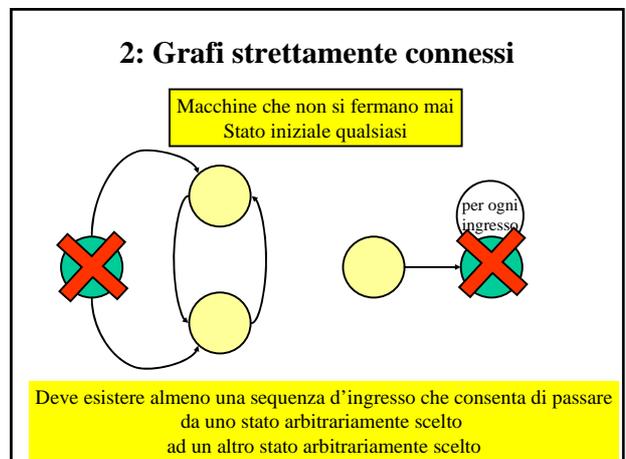
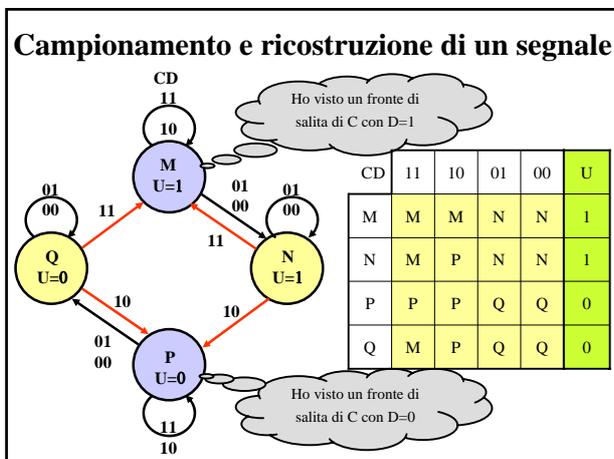
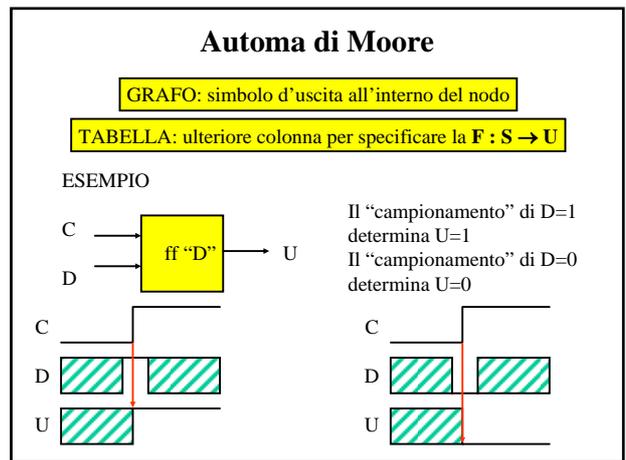
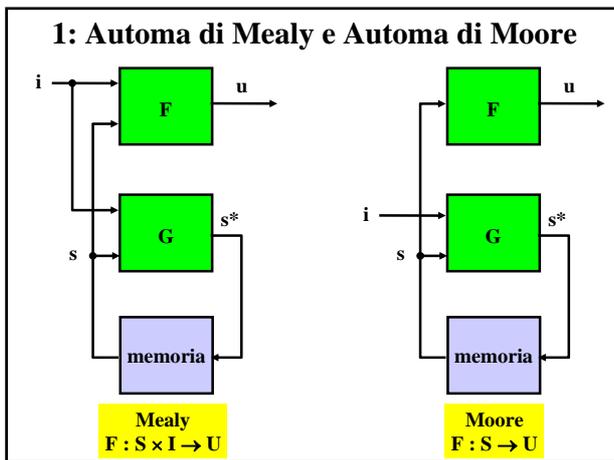
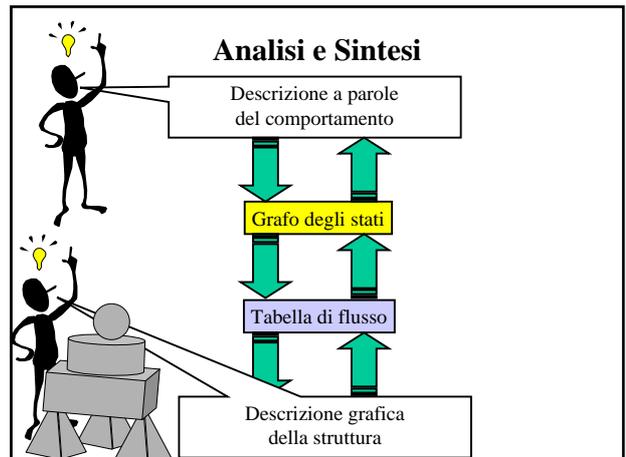
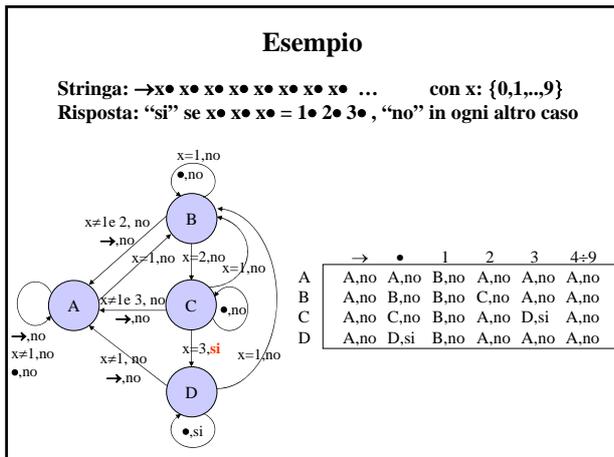
Per tradurre un testo da una lingua all'altra è necessario leggere e memorizzare non solo tutti i caratteri di una parola, ma anche quelli di parole precedenti e successive (un'intera frase). Stato iniziale necessario per il processo di traduzione: il foglio non deve essere bianco, l'interprete deve conoscere le due lingue e non deve essere cieco e afono.



La FSM (Finite State Machine)

Sistema matematico
 $M = \{I, U, S, F, G\}$
 formato da 3 INSIEMI
 $I: \{i_1, i_2, \dots, i_n\}$ alfabeto di ingresso
 $U: \{u_1, u_2, \dots, u_m\}$ alfabeto di uscita
 $S: \{s_1, s_2, \dots, s_k\}$ insieme degli stati
 da 2 FUNZIONI
 $F: S \times I \rightarrow U$ funzione di uscita
 $G: S \times I \rightarrow S$ funzione di aggiornamento dello stato interno
 e da una MEMORIA
 che mantiene il "vecchio stato" s fino a quando non è necessario sostituirlo con il "nuovo stato" s^*



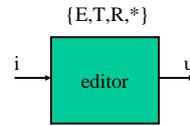


3: Stati indistinguibili

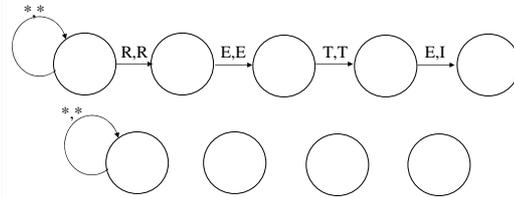
La descrizione con un automa di un comportamento sequenziale non è unica

- **Stati indistinguibili:** due o più stati a partire dai quali, per ogni possibile sequenza d'ingresso, si ottengono sequenze d'uscita identiche
- **Automi equivalenti:** automi che descrivono lo stesso comportamento con differente numero di stati interni
- **Automa minimo:** automa i cui stati interni sono tutti tra loro distinguibili

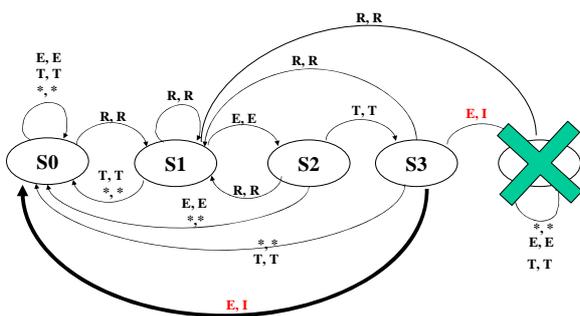
Esercitazione N.1



Trova: RETE
Sostituisci: RETI



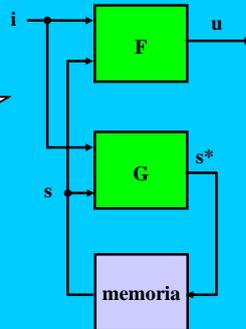
Soluzione: grafo con 5 e con 4 stati



Soluzione: Tabella di Flusso minima

		*	R	E	T
{S0,S4} = S0	S0	S0,*	S1,R	S0,E	S0,T
	S1	S0,*	S1,R	S2,E	S0,T
	S2	S0,*	S1,R	S0,E	S3,T
	S3	S0,*	S1,R	S0,I	S0,T
		*	R	E	T
	S0	S0,*	S1,R	S0,E	S0,T
	S1	S0,*	S1,R	S2,E	S0,T
	S2	S0,*	S1,R	S0,E	S3,T
	S3	S0,*	S1,R	S4,I	S0,T
	S4	S4,*	S1,R	S4,E	S4,T

Macchine
•combinatorie
•asincrone
•sincrone



Classificazione delle FSM

N° di stati interni:

0 o 1 **Macchina combinatoria**



2 o più **Macchina sequenziale**

N° di simboli d'uscita per simbolo d'ingresso:

1 **Macchina sequenziale asincrona**



2 o più **Macchina sequenziale sincrona**



Evento che può modificare l'uscita

▶ modifica dell'ingresso

⌚ modifica dello stato ad ogni

ma come misurararlo?

Esempi

Il relè ad autoritenuta è una macchina **asincrona**:
 MARCIA, finché è premuto, produce passaggio di corrente
 ARRESTO, finché è premuto, impedisce il passaggio di corrente
 MARCIA e ARRESTO, finché non sono premuti, determinano
 o passaggio o assenza di corrente.
**N.B. due effetti per una sola causa, quindi è una macchina sequenziale;
 la durata dell'ingresso non influisce, quindi è una macchina asincrona**

Il semaforo è una **macchina sincrona**:
 Il GIALLO sostituisce il VERDE solo dopo che è trascorsa una
 prefissata quantità di tempo.
 Il ROSSO sostituisce il GIALLO solo dopo che è trascorsa una
 prefissata quantità di tempo.
 Il VERDE sostituisce il ROSSO solo dopo che è trascorsa una
 prefissata quantità di tempo.
**N.B. effetti diversi a istanti successivi e senza modifica
 dell'ingresso, quindi è una macchina sequenziale sincrona**

Macchine asincrone e sincrone

Macchina asincrona - Lo stato e l'uscita possono cambiare solo se cambia l'ingresso.

La "durata" dell'ingresso **non produce informazione**.

Ogni stato è "stabile" per l'ingresso che lo ha causato

"se $s^* = G(s, i)$ allora anche $s^* = G(s^*, i)$ "

Macchina sincrona - Lo stato e l'uscita possono cambiare solo allo scadere di un prefissato intervallo di tempo T_0
 (istanti di sincronismo $t = T_0, 2T_0, 3T_0, \dots$).

Ipotesi: durante l'intervallo l'ingresso è costante

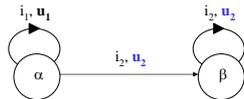
$$u^n = F(s^n, i^n)$$

$$s^{n+1} = G(s^n, i^n)$$

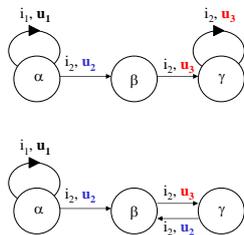
L'intervallo compreso tra due successivi istanti di sincronismo è l'**unità di misura del tempo**.

Grafo di comportamenti asincroni e sincroni

Macchina asincrona: ogni nuovo ingresso produce subito una stabilità e genera quindi un solo nuovo simbolo d'uscita

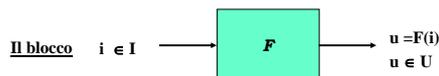


Macchina sincrona: un nuovo ingresso produce una sequenza, finita o periodica, di transizioni di stato e di simboli d'uscita



Macchina combinatoria "ideale": la funzione

Elaborazione combinatoria: per ogni $i \in I$ esiste un solo $u \in U$ che gli corrisponde. **NON c'è MEMORIA, NON c'è RETROAZIONE**



Encoder e Decoder

Encoder: 3 bits input (x2, x1, x0) to 3 bits output (y2, y1, y0). Truth table:

x2	x1	x0	y2	y1	y0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

Decoder: 3 bits input (y2, y1, y0) to 3 bits output (x2, x1, x0). Truth table:

y2	y1	y0	x2	x1	x0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

Full Adder

Inputs: a, b, c. Outputs: s, r.

a	b	c	s	r
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Subtractor

Inputs: a, b, c. Outputs: d, p.

a	b	c	d	p
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Il Multiplexer a due vie

Inputs: i0, i1, s. Output: u.

s	i0	i1	u
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

se a=0 allora u=i0
altrimenti u=i1

Descrizione del comportamento

La tabella

i: var. indipendente
u: var. dipendente

i	u = F(i)
a ₁	b ₂
a ₂	b ₃
a ₃	b ₂
a ₄	b ₃
a ₅	b ₁

$B^n \rightarrow B^m$

L'espressione

ADDER: $u = i_1 + i_2$

SELETORE: $u = i_A$

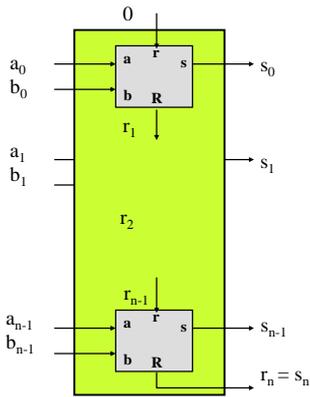


Struttura: composizione e decomposizione

La composizione in serie e/o in parallelo di macchine combinatorie è ancora una macchina combinatoria

Ogni macchina combinatoria può essere decomposta fino ad individuare una disposizione in serie/parallelo di gate

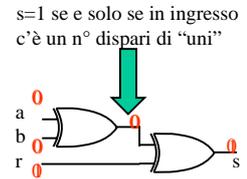
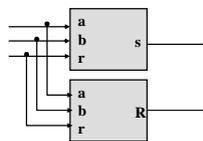
Composizione in serie di Full Adder



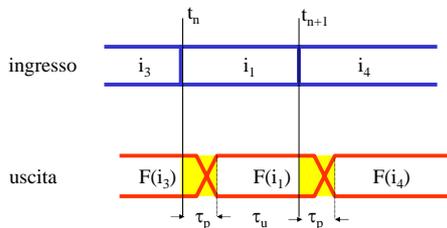
Decomposizione di un Full Adder



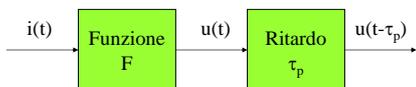
r _i	a _i	b _i	r _{i+1}	s _i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Macchina combinatoria "reale": throughput



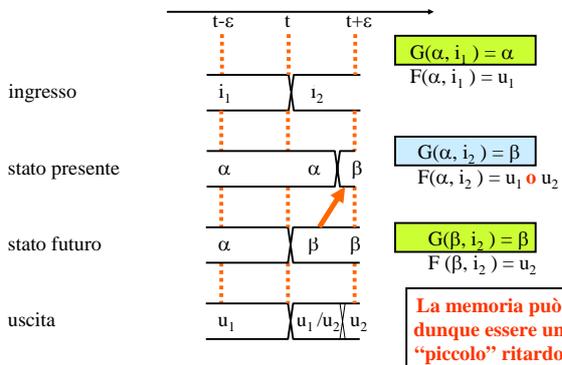
τ_p : tempo di calcolo della F τ_u : tempo di acquisizione del risultato



throughput: $(\tau_p + \tau_u)$ o anche $1/(\tau_p + \tau_u)$



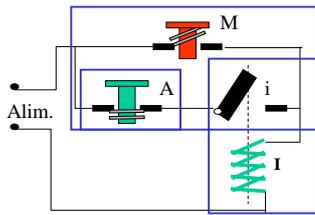
La macchina asincrona (comportamento)



La macchina asincrona (struttura)



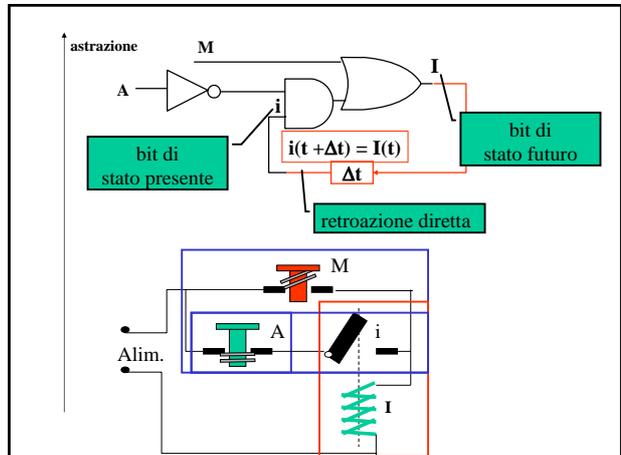
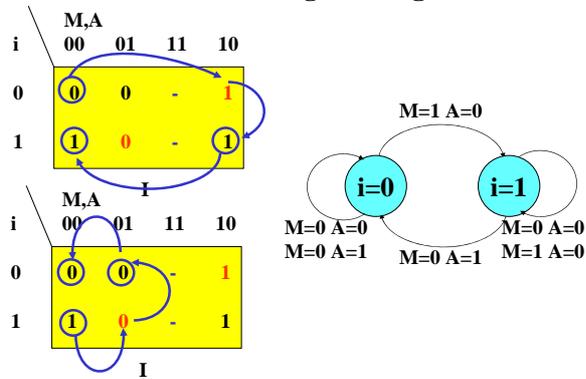
Analisi del relè ad autoritenuta



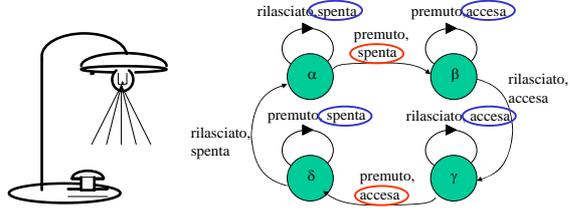
Tabulazione degli esperimenti

		stato presente		stato futuro		
Pulsante M	Alte A	Interruttore i	Corrente I	Situazione		
rilasciato	rilasciato	aperto	00	stabile		
rilasciato	rilasciato	chiuso	11	stabile		
preluto	rilasciato	aperto	10	instabile		
preluto	rilasciato	chiuso	11	stabile		
rilasciato	preluto	aperto	00	stabile		
rilasciato	preluto	chiuso	00	instabile		
preluto	preluto	aperto	11	inutile		
preluto	preluto	chiuso	11	inutile		

Relè con autoritenuta: tabella di flusso e grafo degli stati



Un esempio di macchina asincrona: la lampada da tavolo



pulsante $i \in I: \{\text{rilasciato, premuto}\}$
lampadina $u \in U: \{\text{spenta, accesa}\}$

N.B: durata di una transizione uscita durante una transizione

	rilasciato	premuto
α	α , spenta	β , spenta
β	γ , accesa	β , accesa
γ	δ , accesa	δ , accesa
δ	α , spenta	δ , spenta

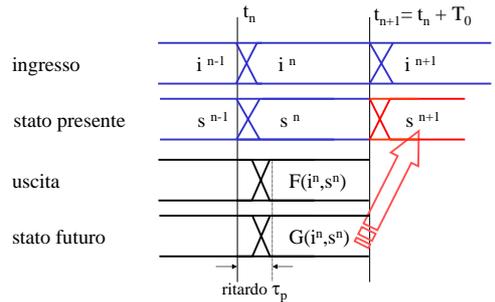
3.3 La macchina sincrona

Segnali sincroni

Per ottenere un'esatta misura del tempo la modifica dei segnali di ingresso/uscita/stato deve verificarsi solo in corrispondenza di **istanti di sincronismo** distanziati uno dall'altro di una quantità prefissata T_0

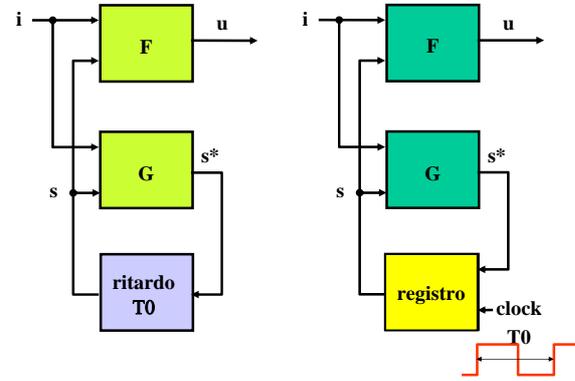
La macchina sincrona

T_0 : intervallo di tempo in cui la macchina non modifica il suo stato



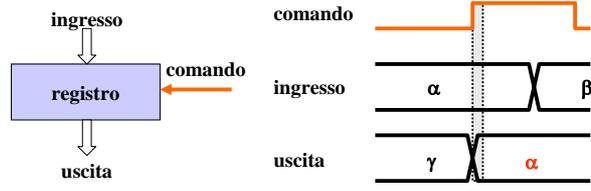
τ_p : intervallo di tempo impiegato dal calcolo di F e di G

La macchina sincrona (struttura)

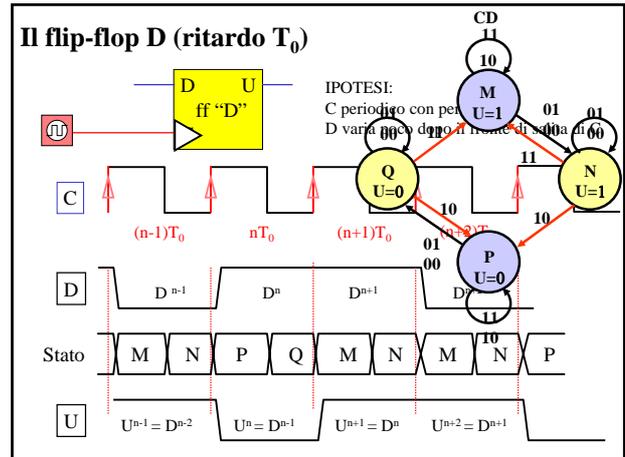


Il registro

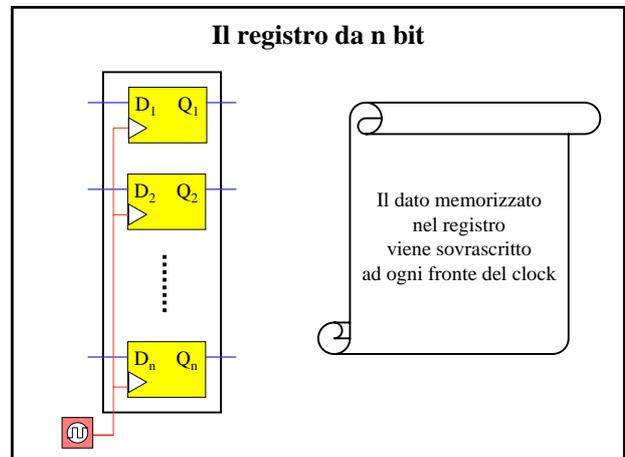
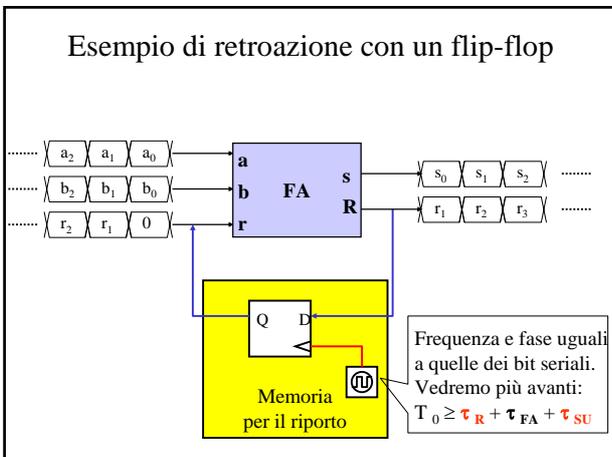
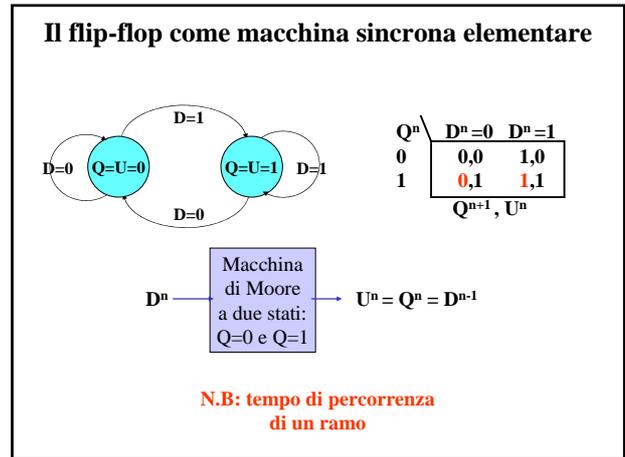
Registro - Macchina sequenziale che memorizza e rende disponibile in uscita un "dato" che in precedenza le è stato fornito in ingresso. La scrittura di un nuovo "dato" è stabilita da un comando esterno detto "clock".

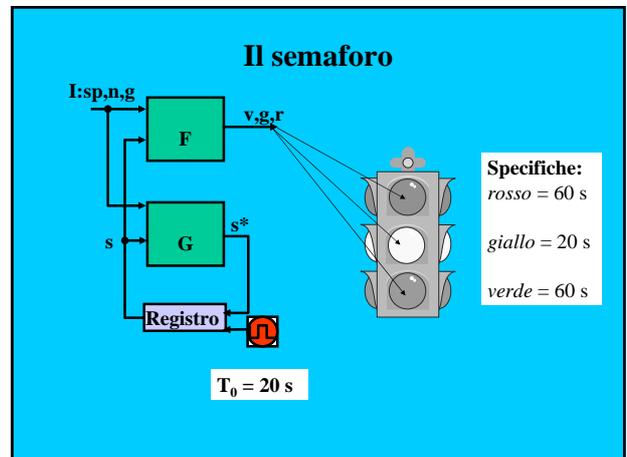
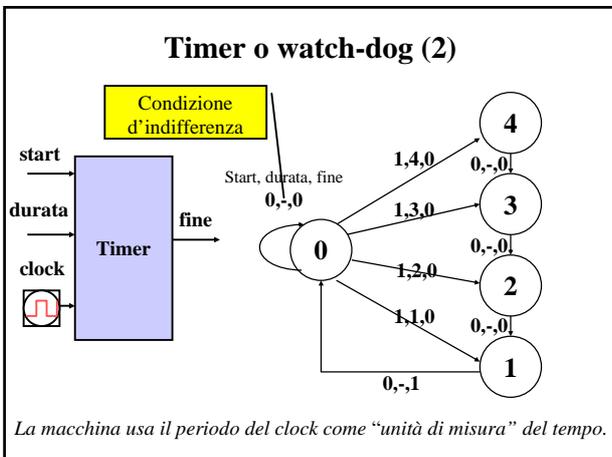
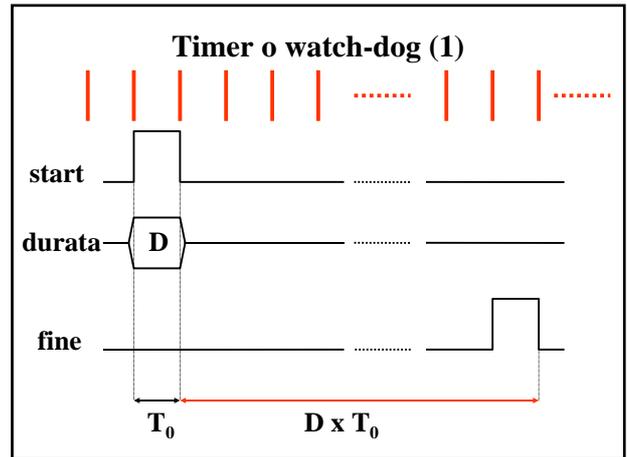
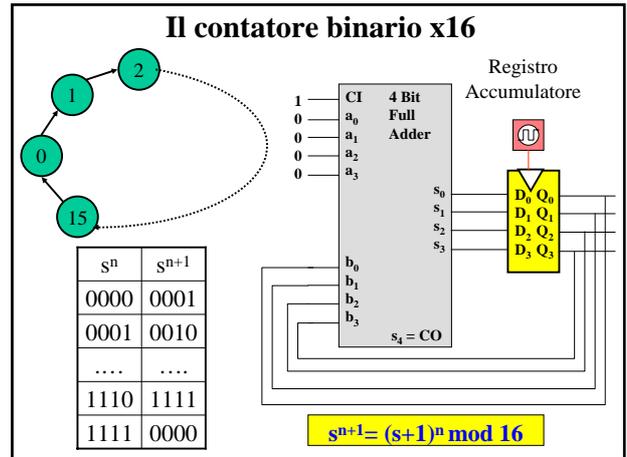
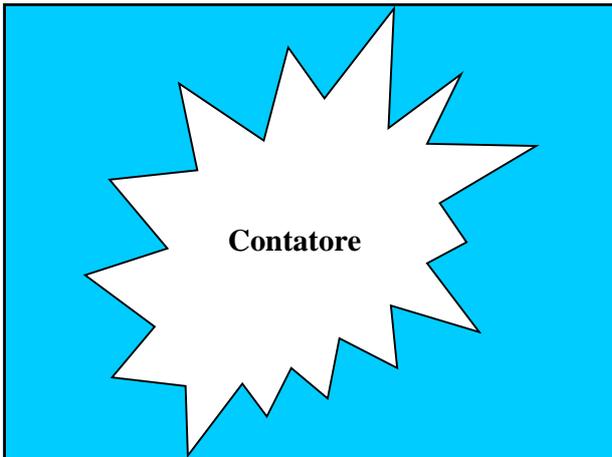


Il registro da un bit

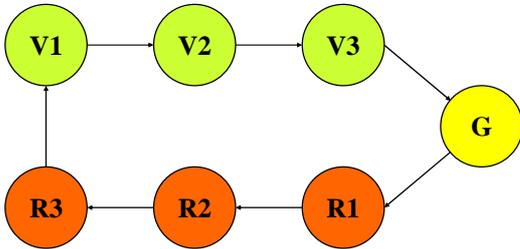


Il flip-flop genera un segnale sincrono anche se le variazioni di D non sono allineate con gli istanti di sincronismo. Basta che D sia costante al momento del campionamento





Il grafo degli stati



La tabella di flusso

stato presente	stato futuro	lampada		
		verde	giallo	rosso
V1	V2	accesa	spenta	spenta
V2	V3	accesa	spenta	spenta
V3	G	accesa	spenta	spenta
G	R1	spenta	accesa	spenta
R1	R2	spenta	spenta	accesa
R2	R3	spenta	spenta	accesa
R3	V1	spenta	spenta	accesa

La macchina sequenziale per il semaforo

Stato interno

$s = y_2 y_1 y_0$ (7 stati)

Uscita

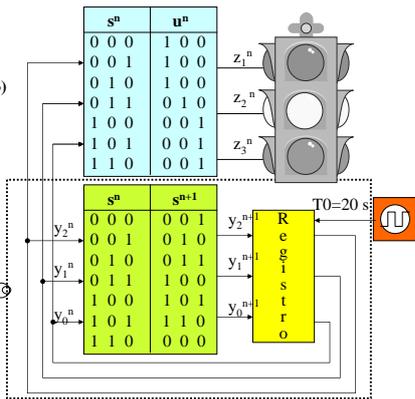
$u = z_1 z_2 z_3$ (codice 1 su 3)

Comportamento:

$s_2 \leftarrow (s+1)_2 \text{ mod } 7$

$u \leftarrow F(s)$

Contatore da "zero" a "sei"



..e la seconda direttrice di marcia?

stato presente	stato futuro	lampade					
		V ₁	G ₁	R ₁	V ₂	G ₂	R ₂
0	1	1	0	0	0	0	1
1	2	1	0	0	0	0	1
2	3	1	0	0	0	0	1
3	4	0	1	0	0	0	1
4	5	0	0	1	1	0	0
5	6	0	0	1	1	0	0
6	0	0	0	1	0	1	0