

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 o D2, ma non entrambi

D1	D2	L
alto	alto	spenta
alto	basso	accesa
basso	basso	spenta

Due “nor” in retroazione

$V_2 = V_3 = L$
 $V_u = V_l = ?$
 o H o L

Le due trascodifiche

ENCODER
trascod. da 1 su 4 a binario

Inputs: x_4, x_3, x_2, x_1
Outputs: y_2, y_1, y_0

x_4, x_3, x_2, x_1	y_2, y_1, y_0
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 1 0
0 0 1 1	0 1 1
1 0 0 0	1 0 0

DECODER
trascod. da binario a 1 su 8

Inputs: y_2, y_1, y_0
Outputs: x_4, x_3, x_2, x_1, x_0

y_2, y_1, y_0	x_4, x_3, x_2, x_1, x_0
0 0 0	0 0 0 0 0
0 0 1	0 0 0 0 1
0 1 0	0 0 0 1 0
0 1 1	0 0 0 1 1
1 0 0	0 0 1 0 0
1 0 1	0 0 1 0 1
1 1 0	0 0 1 1 0
1 1 1	0 0 1 1 1

La conversione P/S di un byte

Ingresso: $b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7$
 Uscita: $b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7$
 Stato: $000, 001, 010, 011, 100, 101, 110, 111$
 Controller

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: $u = M_2(M_1(i))$ **Funzione composta**
 Deve operare prima il blocco a sinistra, poi quello a destra.

b) in parallelo: $\begin{cases} u_1 = M_1(i) \\ u_2 = M_2(i) \end{cases}$ **Sistema di funzioni**
 I due blocchi operano contemporaneamente.

c) in retroazione: $\begin{cases} u = M_1(i, s) \\ s = M_2(u) \\ u = M_1(i, M_2(u)) \end{cases}$ **Funzione ricorsiva**
 È necessario che l'anello completi un calcolo prima di avviarne uno nuovo.

pochi componenti primitivi!

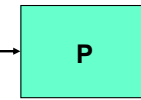
3.1

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

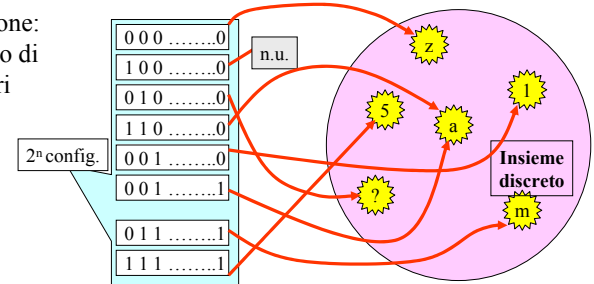


$u(t) = P(J(t))$

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

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

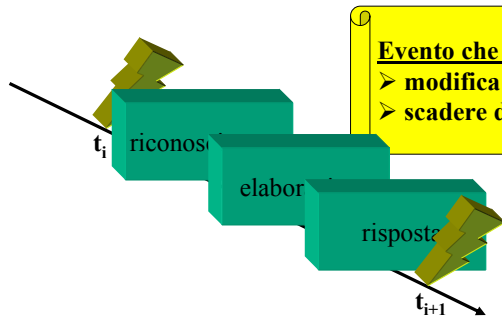
Comunicazione:
numero finito di
segnali binari



La discretizzazione del tempo

Evento che introduce informazione

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

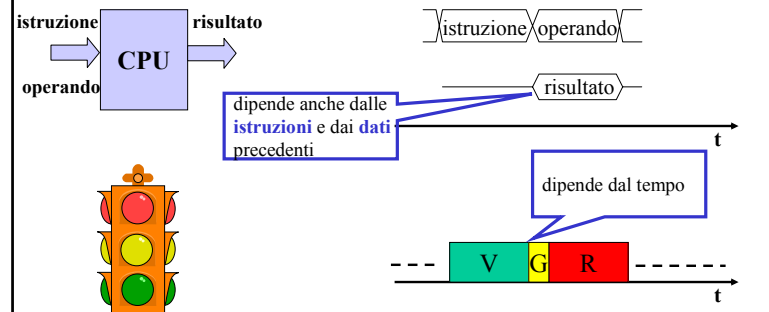
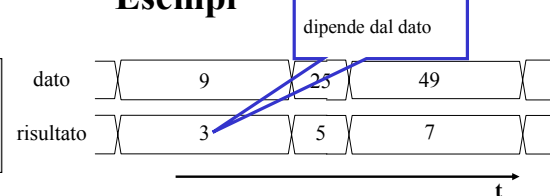


l'uscita può dunque dipendere:

- dall'ingresso contemporaneo
- dalla sequenza d'ingresso
- dal trascorrere del tempo

Esempi

$$u = \sqrt{i}$$



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 $s(t_0)$.

$$u(t_n) = P(s(t_0), i(t_0), i(t_1), \dots, i(t_{n-1}), i(t_n))$$

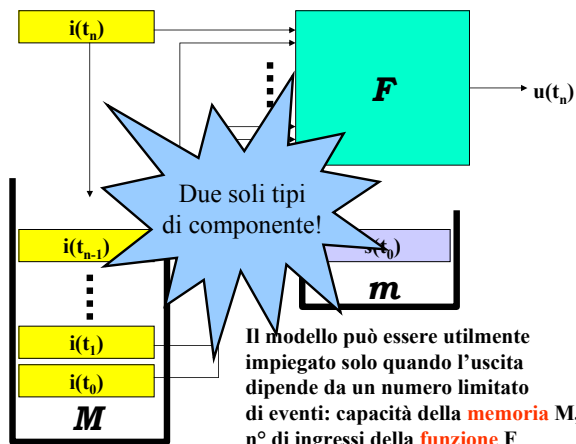
Stato iniziale di una macchina

$$s(t_0) \in S$$

Esempio : il percorso di un'auto dipende non solo dai comandi via 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.

Macchina sequenziale: memorie e funzione



Un esempio: l'interprete

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.

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.

Classificazione delle macchine

Tipo

Relazione ingresso/uscita



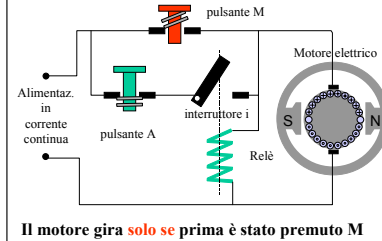
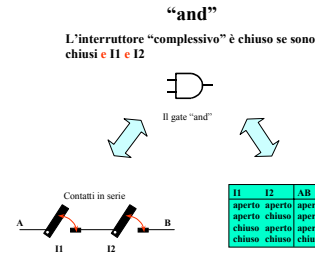
Macchina sequenziale

$u(t_n) = P(i(t_{n-1}), i(t_n))$

Macchina combinatoria

$u(t_n) = P(i(t_n))$
 $u = P(i)$

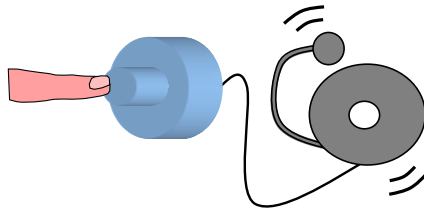
Esempi



Macchina combinatoria

Macchina sequenziale

Il campanello



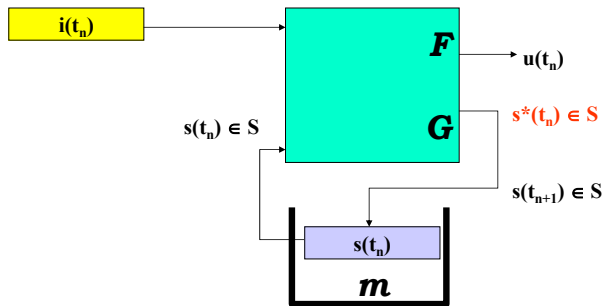
<i>i: Pulsante</i>	<i>u: Suoneria</i>		<i>i: Pulsante</i>	<i>u: Suoneria</i>
Premuto	din	t_0	Premuto	din
Rilasciato	nessun suono	t_1	Rilasciato	nessun suono
		t_2	Rilasciato	don
		t_3	Rilasciato	nessun suono

$u = F(i)$

$u(t_i) = P(i(t_i), i(t_{i-1}), ..)$

La macchina a stati finiti

Macchine sequenziali: stato interno presente e futuro



S , un insieme *finito* di stati interni, riassume tutte le possibili storie passate.
 La funzione F , tramite $i(t_n)$ e $s(t_n)$ (detto *stato interno presente*), calcola la *risposta* $u(t_n)$
 La funzione G , con gli stessi argomenti, calcola il *nuovo riassunto* $s^*(t_n)$ (detto *stato interno futuro*)

Evoluzione del processo di elaborazione

Funzione d'uscita: $S \times I \rightarrow U$

$$u(t_n) = F(s(t_n), i(t_n))$$

Funzione di stato: $S \times I \rightarrow S$

$$s^*(t_n) = G(s(t_n), i(t_n))$$

Aggiornamento dello stato:

$$s(t_{n+1}) = s^*(t_n)$$

$$\begin{aligned} u(t_n) &= F(s(t_n), i(t_n)) \\ &= F(G(s(t_{n-1}), i(t_{n-1})), i(t_n)) \\ &\dots\dots\dots \\ &= F(G\dots(G(G(s(t_0), i(t_0)), i(t_1)), i(t_2)), \dots), i(t_n)) \end{aligned}$$

$$u(t_n) = P(s(t_0), i(t_0), i(t_1), \dots, i(t_{n-1}), i(t_n))$$

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

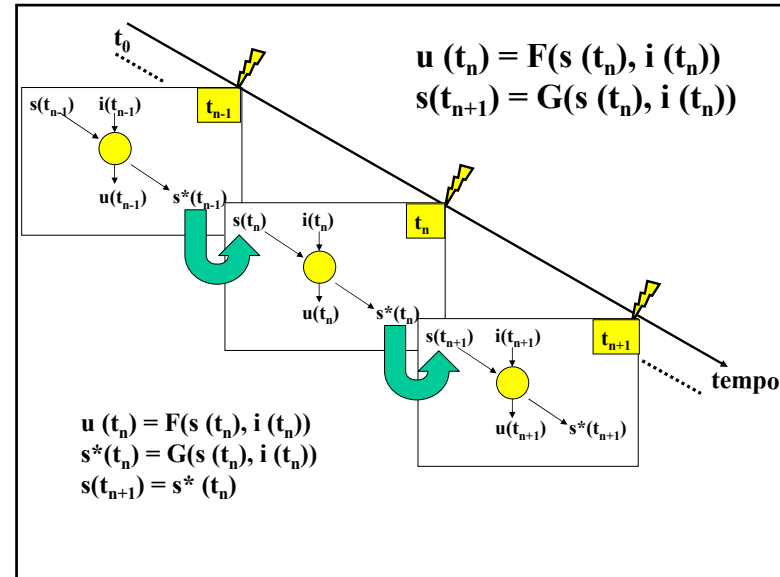
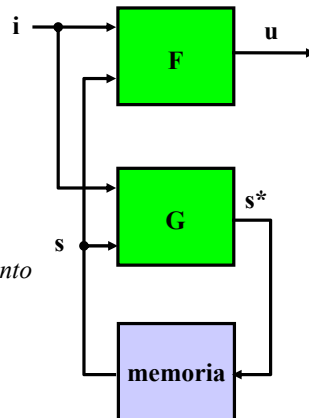
$G: S \times I \rightarrow S$ funzione di aggiornamento dello stato interno

e da una MEMORIA

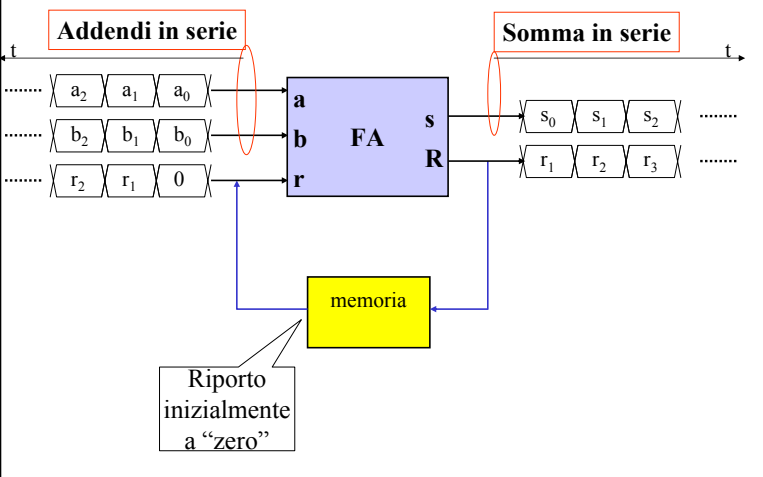
che mantiene il "vecchio stato" s

fino a quando non è necessario

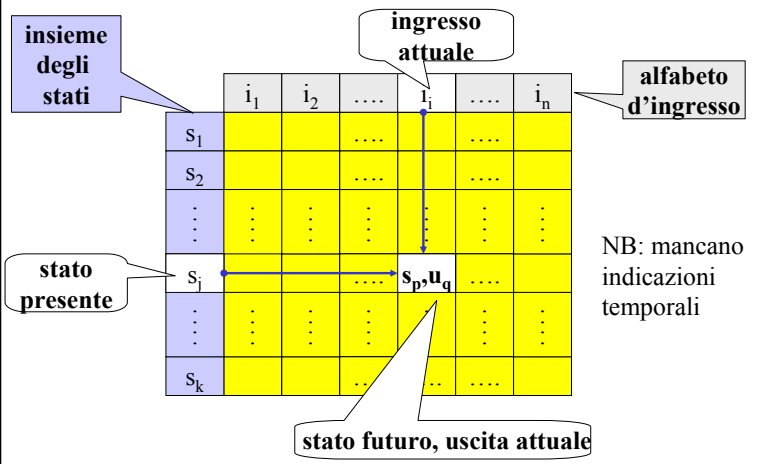
sostituirlo con il "nuovo stato" s^*



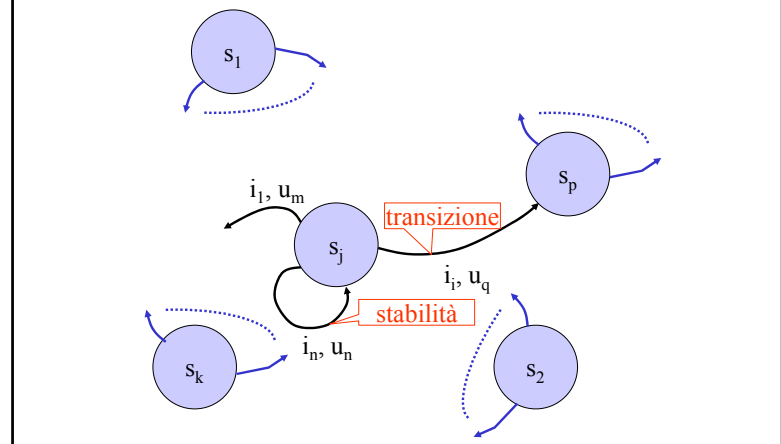
Compromesso spazio/tempo: addizione in serie



Descrizione con tabella di flusso

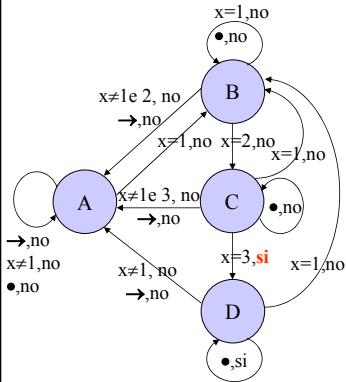


Descrizione con grafo degli stati



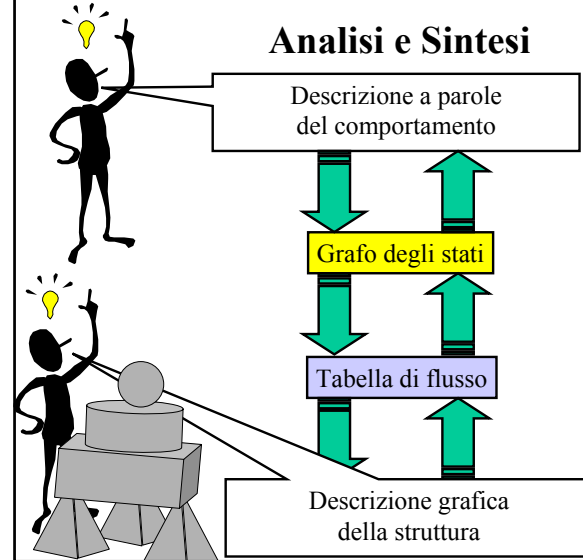
Esempio

Stringa: $\rightarrow x \bullet x \bullet x \bullet x \bullet x \bullet x \bullet x \bullet x \dots$ con $x: \{0,1,\dots,9\}$
 Risposta: "si" se $x \bullet x \bullet x = 1 \bullet 2 \bullet 3$, "no" in ogni altro caso

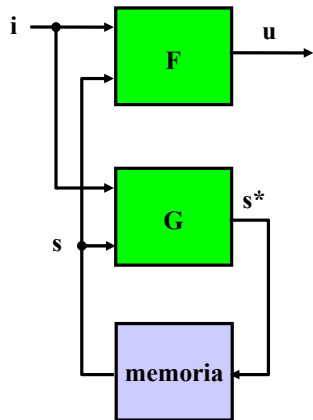


	\rightarrow	\bullet	1	2	3	4÷9
A	A,no	A,no	B,no	A,no	A,no	A,no
B	A,no	B,no	B,no	C,no	A,no	A,no
C	A,no	C,no	B,no	A,no	D,si	A,no
D	A,no	D,si	B,no	A,no	A,no	A,no

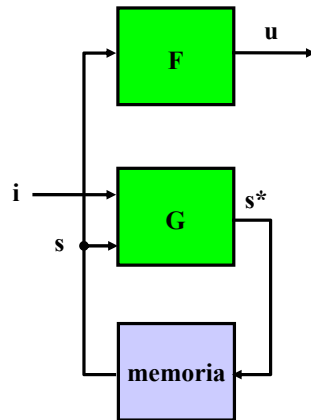
Analisi e Sintesi



Automa di Mealy e Automa di Moore



Mealy
 $F : S \times I \rightarrow U$



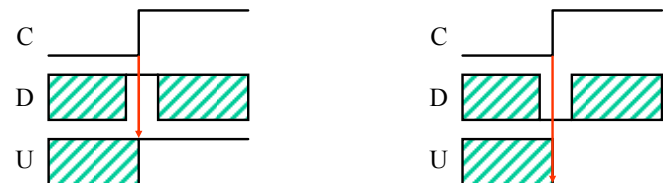
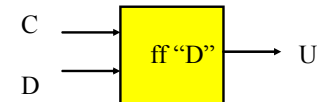
Moore
 $F : S \rightarrow U$

Automa di Moore

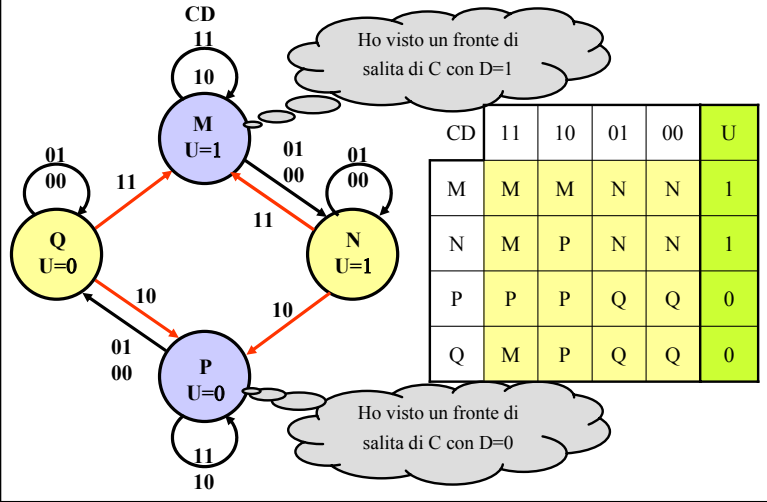
GRAFO: simbolo d'uscita all'interno del nodo

TABELLA: ulteriore colonna per specificare la $F : S \rightarrow U$

ESEMPIO

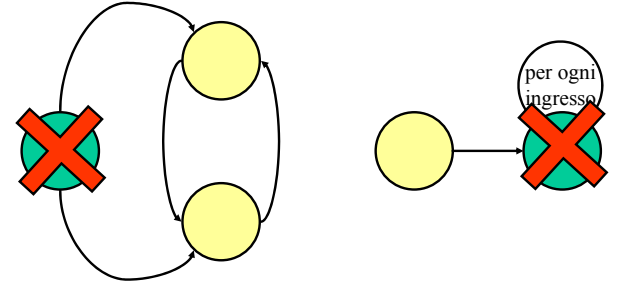


Campionamento e ricostruzione di un segnale



Grafi strettamente connessi

Macchine che non si fermano mai
Stato iniziale qualsiasi

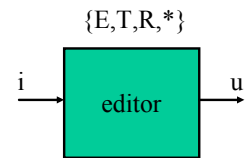


Deve esistere almeno una sequenza d'ingresso che consenta di passare da uno stato arbitrariamente scelto ad un altro stato arbitrariamente scelto

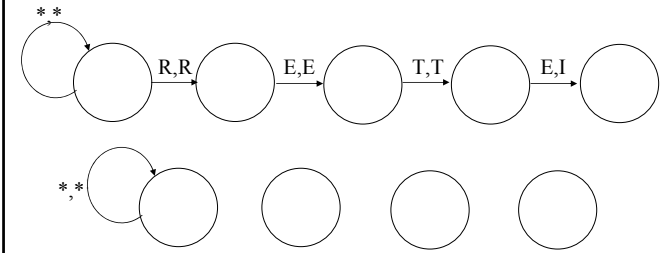
Stati equivalenti

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

Esercitazione



Trova: RETE
Sostituisci: RETI



Macchine asincrone e Macchine sincrone

Ulteriore classificazione delle macchine sequenziali

Evento che introduce informazione

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

Macchina
asincrona

Macchina
sincrona

Classificazione delle macchine sequenziali

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

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 (*istante di sincronismo*).

Durante l'intervallo l'ingresso è costante.

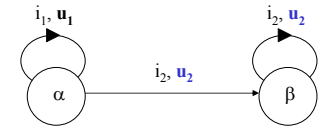
$$u^n = F(s^n, i^n)$$

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

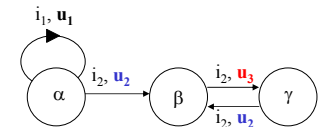
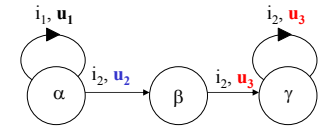
L'intervallo compreso tra due successivi istanti di sincronismo è **l'unità di misura del tempo**.

Grafo di comportamenti asincroni e sincrioni

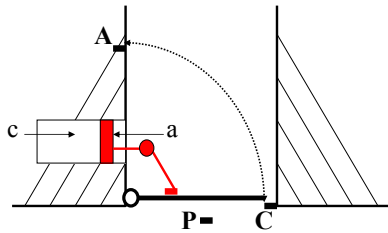
Macchina asincrona: ogni nuovo ingresso produce subito una stabilità e genera quindi **un solo nuovo simbolo d'uscita**



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



Esercitazione N. 2



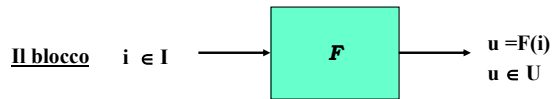
Macchina combinatoria? **NO!**

Macchina sequenziale asincrona? **SI!**

3.2 - La macchina combinatoria

Macchina combinatoria "ideale": la funzione

Elaborazione combinatoria: per ogni $i \in I$ esiste un solo $u \in U$ che gli corrisponde.



La tabella
 i : var. indipendente
 u : var. dipendente

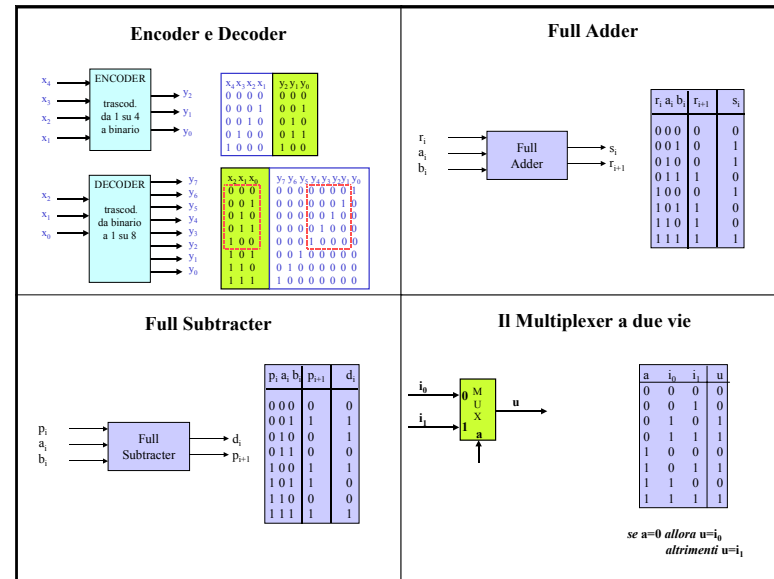
i	$u = F(i)$
a_1	b_2
a_2	b_3
a_3	b_2
a_4	b_3
a_5	b_1

$B^n \rightarrow B^m$



L'espressione **ADDER:** $u = i_1 + i_2$

SELETTORE: $u = i_A$

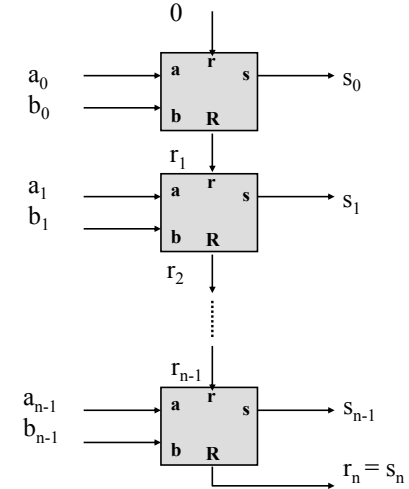


Composizione e Decomposizione

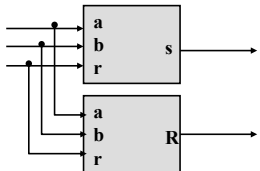
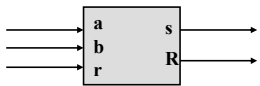
La composizione in serie e/o in parallelo di macchine combinatorie è ancora una macchina combinatoria

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

Composizione in serie di Full Adder



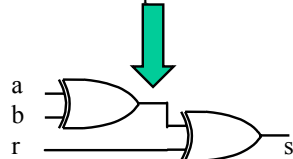
Decomposizione di un Full Adder



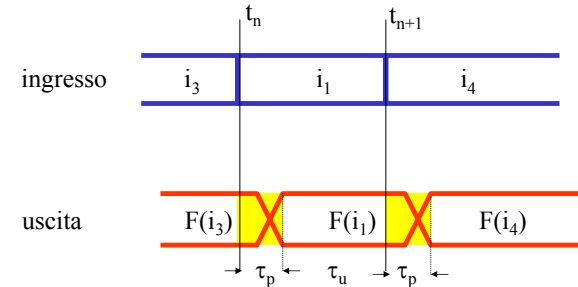
r_i	a_i	b_i	r_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$s=1$ se e solo se in ingresso c'è un n° dispari di "uni"

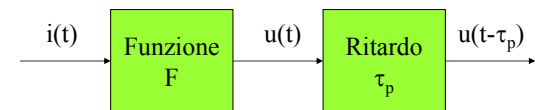
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0



Macchina combinatoria "reale": throughput



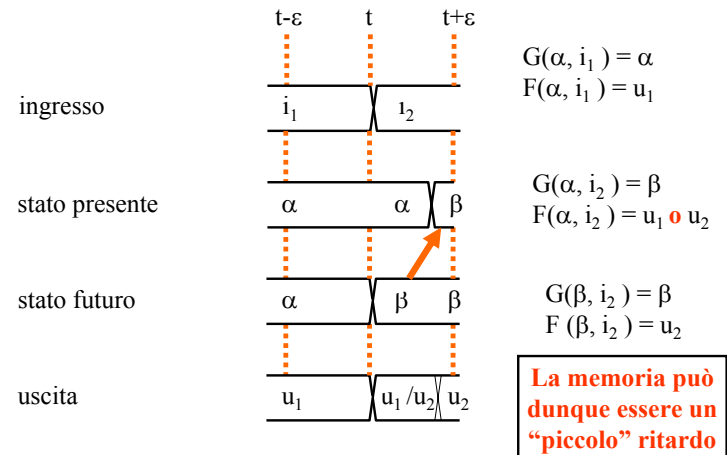
τ_p : tempo di calcolo della F τ_u : tempo di acquisizione del risultato



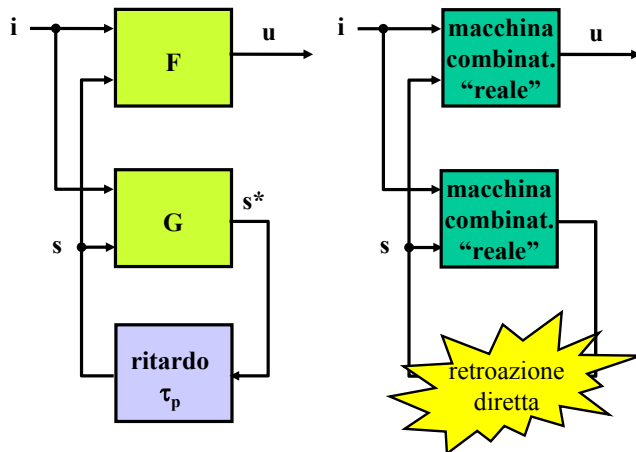
throughput: $(\tau_p + \tau_u)$ o anche $1/(\tau_p + \tau_u)$

3.2 La macchina asincrona

La macchina asincrona (comportamento)



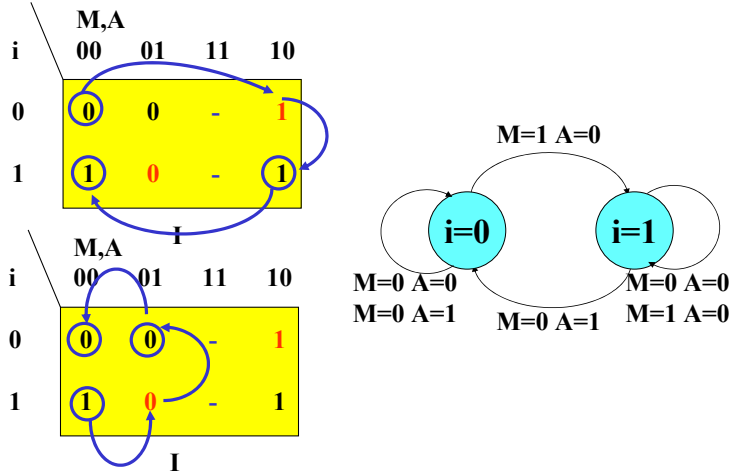
La macchina asincrona (struttura)



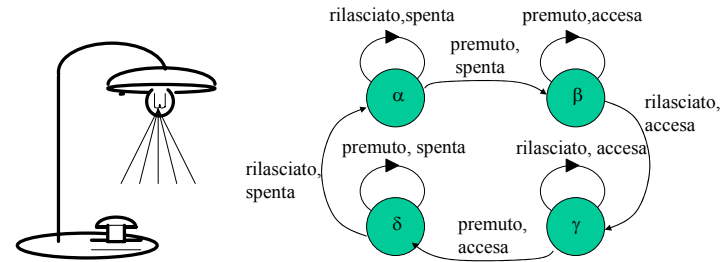
Relè ad "autoritenuta": tabulazione degli esperimenti

Pulsante M	Pulsante A	Interruttore i	Corrente I	Situazione
rilasciato	rilasciato	aperto	00	<i>stabile</i>
rilasciato	rilasciato	chiuso	11	<i>stabile</i>
premutato	rilasciato	aperto	11	<i>instabile</i>
premutato	rilasciato	chiuso	11	<i>stabile</i>
rilasciato	premutato	aperto	00	<i>stabile</i>
rilasciato	premutato	chiuso	00	<i>instabile</i>
premutato	premutato	aperto	11	<i>inutile</i>
premutato	premutato	chiuso	11	<i>inutile</i>

Relè con autoritenuta: tabella di flusso e grafo degli stati



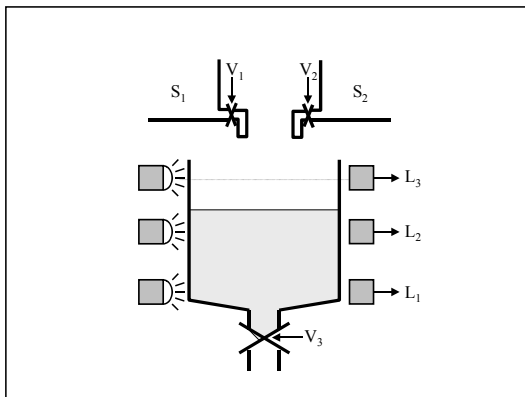
Un esempio di macchina asincrona: la lampada da tavolo



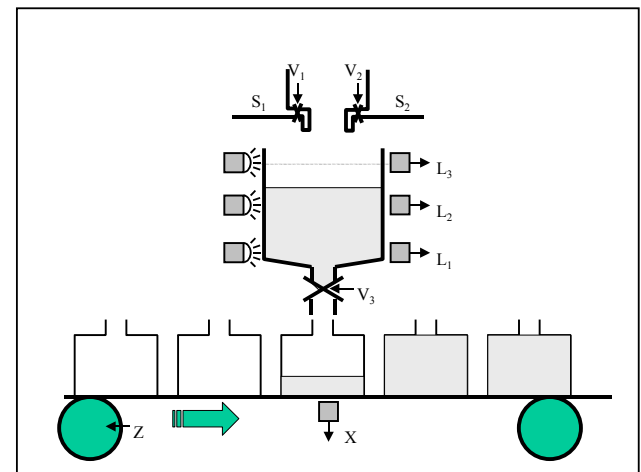
pulsante $i \in I: \{\text{rilasciato,premutato}\}$
lampadina $u \in U: \{\text{spenta, accesa}\}$

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

Esercitazione N.3 (prima parte)



Esercitazione N.3 (seconda parte)



3.3 La macchina sincrona

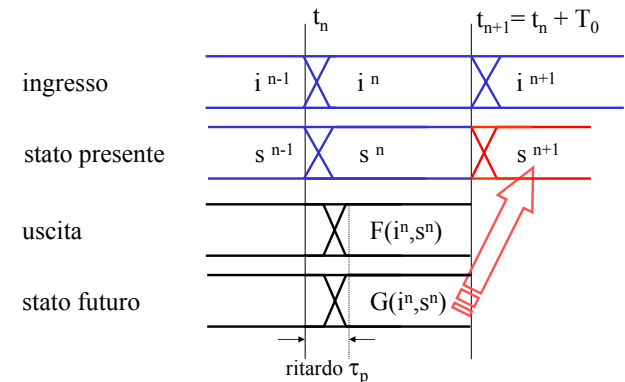
Segnali sincroni

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

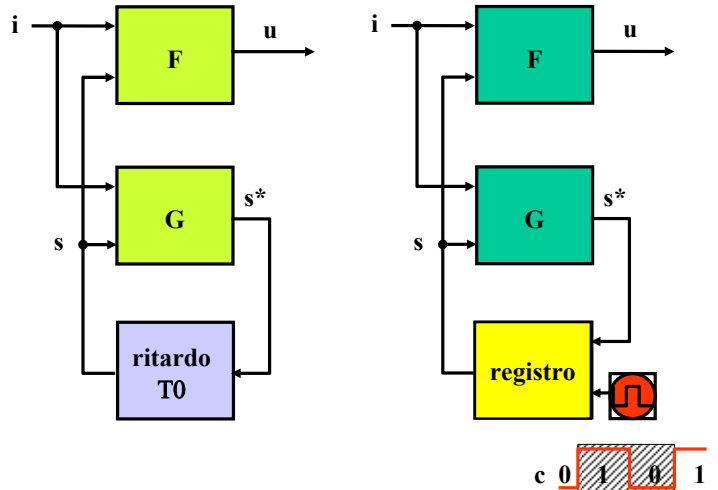
La macchina sincrona

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)



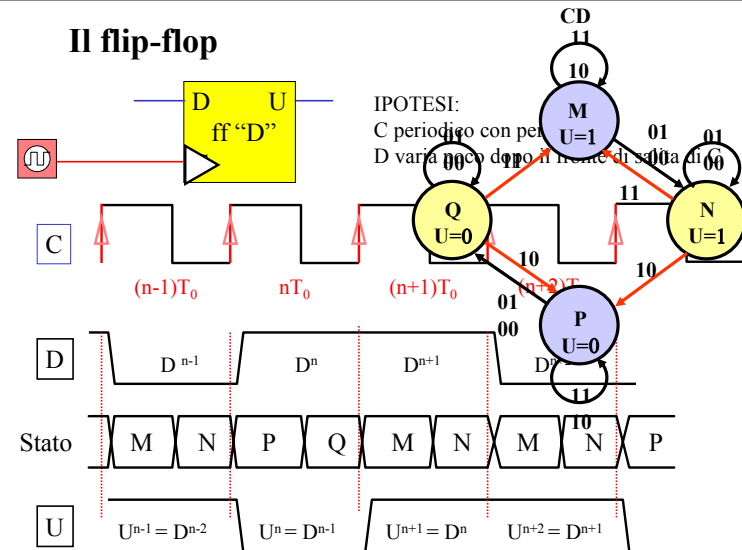
Retroazioni con flip-flop e registro

Il registro

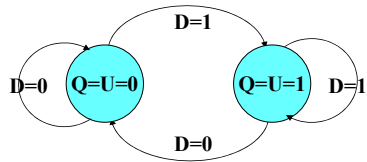
Registro - Macchina sequenziale che memorizza e rende disponibile in uscita un "dato" che in precedenza le è stato fornito in ingresso. La scrittura di un nuovo "dato" è stabilita da un comando che proviene dall'esterno.



Il flip-flop

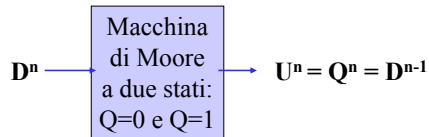


Il flip-flop come macchina sincrona elementare

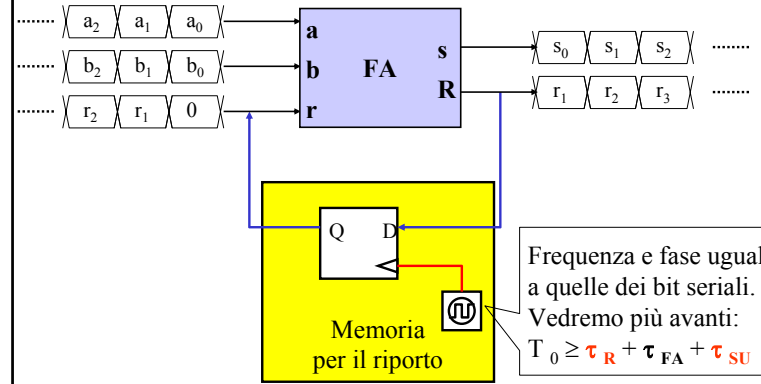


Q^n \ D^n	$D^n=0$	$D^n=1$
0	0,0	1,0
1	0,1	1,1

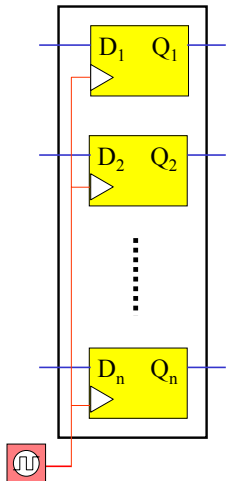
Q^{n+1}, U^n



Addizione in serie (ff D)



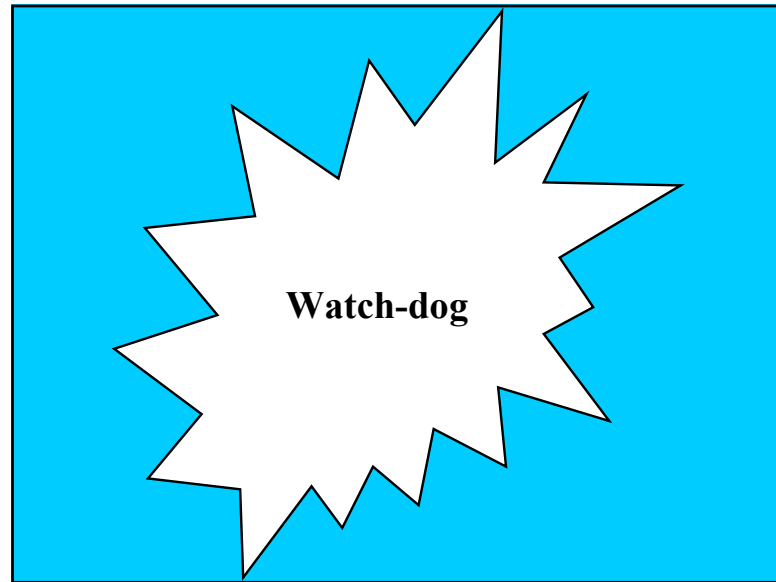
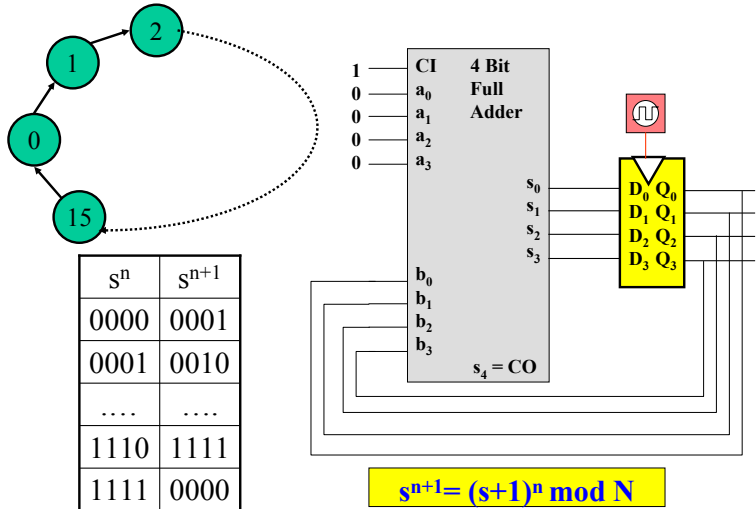
Il registro da n bit



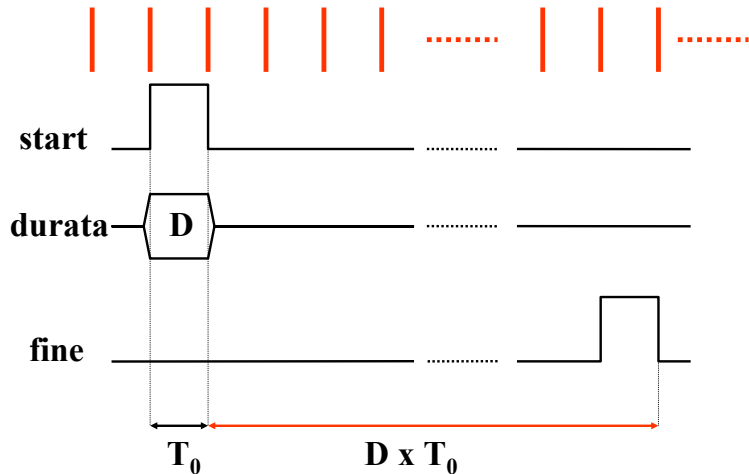
Il dato memorizzato nel registro viene sovrascritto ad ogni fronte del clock

Contatore

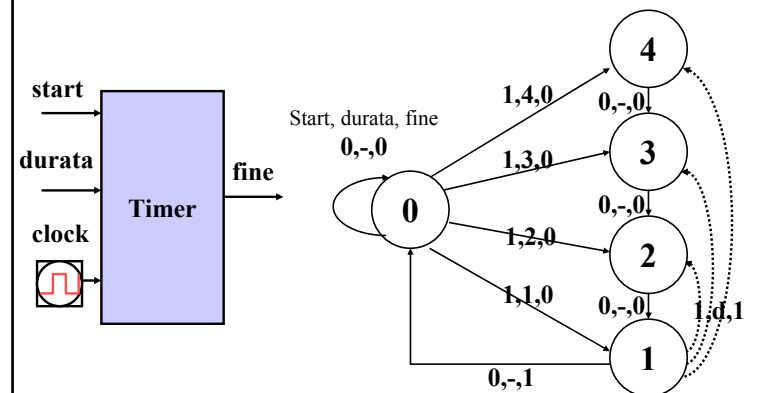
Il contatore binario x16



Timer o watch-dog (1)

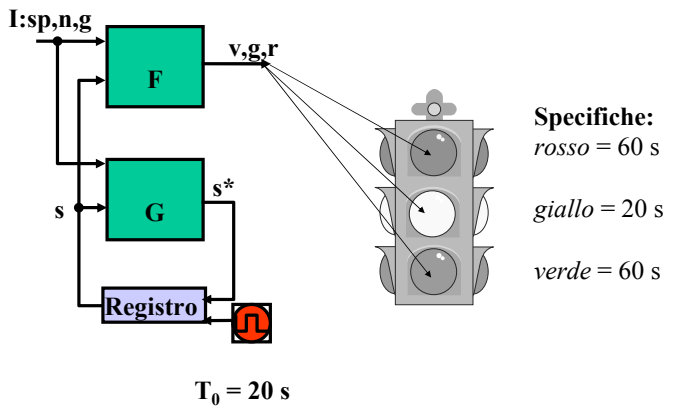


Timer o watch-dog (2)

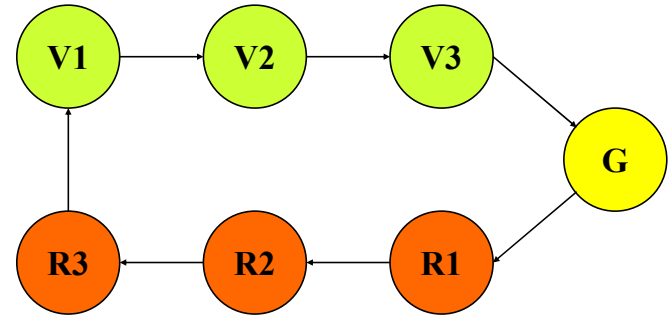


La macchina usa il periodo del clock come "unità di misura" del tempo.

Il semaforo



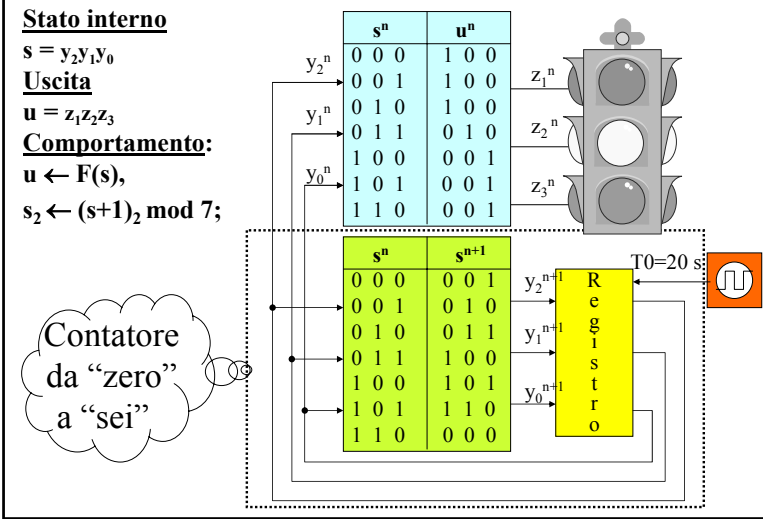
Il grafo degli stati



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



Esercitazione

Modificare la tabella di flusso del semaforo per poter gestire **due direzioni** di marcia e per attribuire a ciascuna: “verde” per 4 T0, “giallo” per T0 e “rosso” per 4T0.

Esercitazione N. 4

