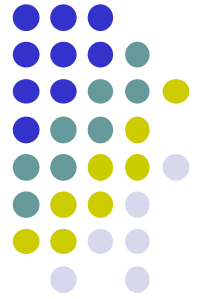
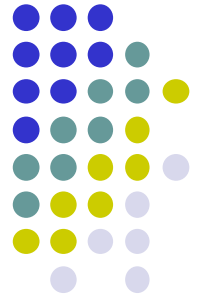
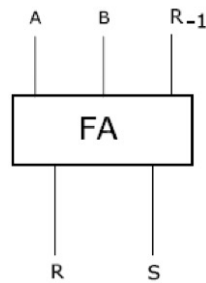


Aggiunta di operazioni logiche



- Se si vogliono sfruttare le porte AND e OR dell' ALU già presenti nei sommatatori ed in particolare quelle usate per calcolare le somme S_i , è necessario annullare l' effetto dei riporti
- Ciò si può ottenere mediante l' aggiunta di altra logica e altre linee di controllo
- Prima però modifichiamo opportunamente le celle sommatrici per poter poi includere più agevolmente tali operazioni

A_i	B_i	R_{i-1}	S_i	R_i
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



- Vediamo le mappe di Karnaugh corrispondenti alle somme S_i e ai riporti R_i

		$B_i R_{i-1}$			
		00	01	11	10
A_i	0		1		1
	1	1		1	

S_i

		$B_i R_{i-1}$			
		00	01	11	10
A_i	0			1	
	1		1	1	1

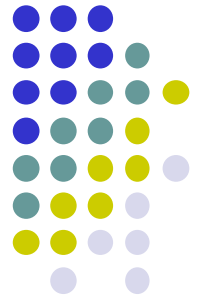
R_i

Figura 3.52 Mappe Karnaugh di S_i e R_i .

- Da esse ricaviamo

$$S_i = \neg A_i \neg B_i R_{i-1} + \neg A_i B_i \neg R_{i-1} + A_i \neg B_i \neg R_{i-1} + A_i B_i R_{i-1}$$

$$R_i = \boxed{A_i R_{i-1}} + \boxed{A_i B_i} + \boxed{B_i R_{i-1}}$$



- Esprimiamo S_i in funzione di R_i

		$B_i R_{i-1}$			
		00	01	11	10
A_i	0	1	1		1
	1	1			

a)

$$\overline{R_i}$$

		$B_i R_{i-1}$			
		00	01	11	10
A_i	0		1		1
	1	1			

b)

$$\overline{R_i}(A_i + B_i + R_{i-1})$$

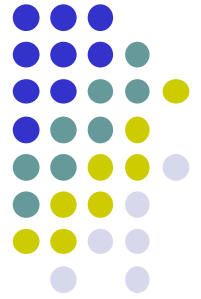
		$B_i R_{i-1}$			
		00	01	11	10
A_i	0		1		1
	1	1		1	

c)

$$\overline{R_i}(A_i + B_i + R_{i-1}) + A_i B_i R_{i-1}$$

Figura 3.53 Metodo pratico per pervenire a una espressione di S_i in funzione di A_i , B_i , R_i e R_{i-1} .

- Risulta $S_i = A_i \neg R_i + B_i \neg R_i + \neg R_i R_{i-1} + A_i B_i R_{i-1}$



- Ecco il circuito corrispondente

$$S_i = A_i \neg R_i + B_i \neg R_i + \neg R_i R_{i-1} + A_i B_i R_{i-1}$$

$$R_i = A_i R_{i-1} + A_i B_i + B_i R_{i-1}$$

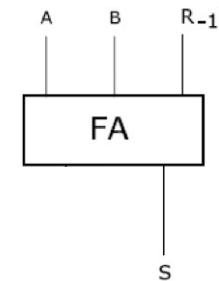
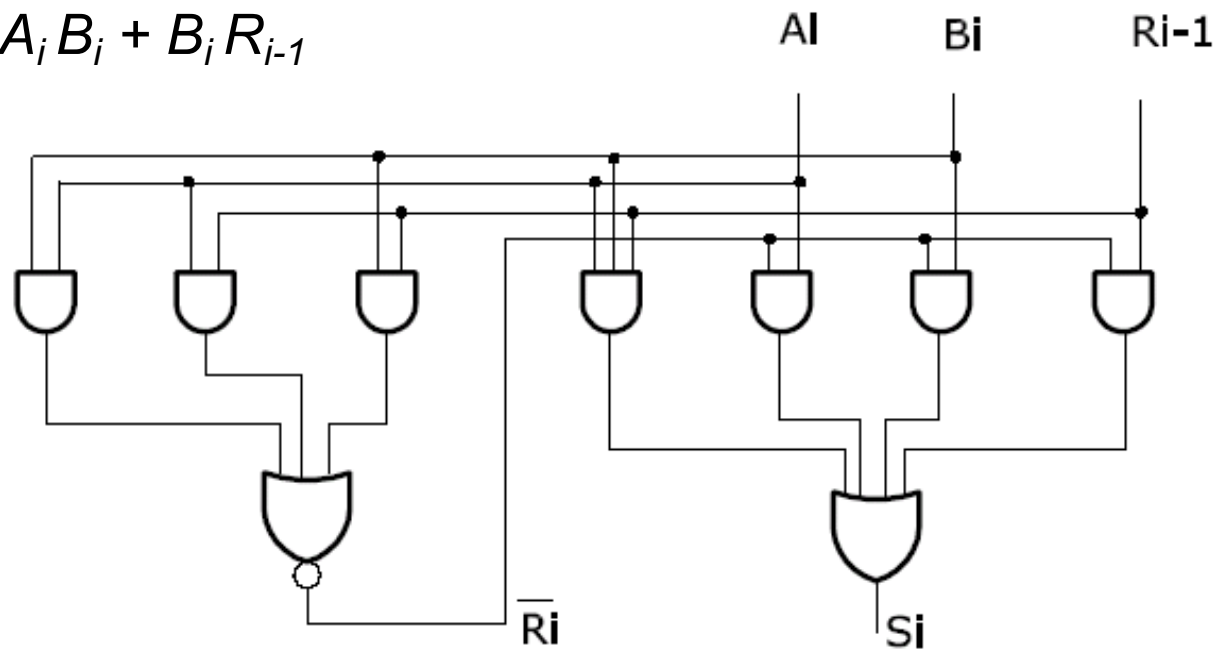
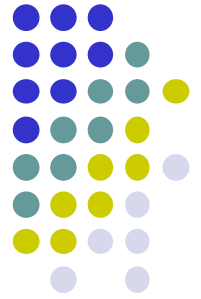


Figura 3.54 Schema per il sommatore completo con uscita S_i in funzione di A_i , B_i , R_i e R_{i-1} . La rete corrisponde – a meno di differenze irrilevanti – a metà del contenuto del componente 7482.



- Se si vogliono sfruttare le porte AND e OR dell' ALU già presenti nei sommatatori ed in particolare quelle usate per calcolare le somme S_i , è necessario annullare l' effetto dei riporti
- Ciò si può ottenere mediante l' aggiunta di altre due linee di controllo c_2 e c_3
- Ecco lo schema circuitale della ALU così estesa

$$R_i = A_i R_{i-1} + A_i B_i + B_i R_{i-1}$$

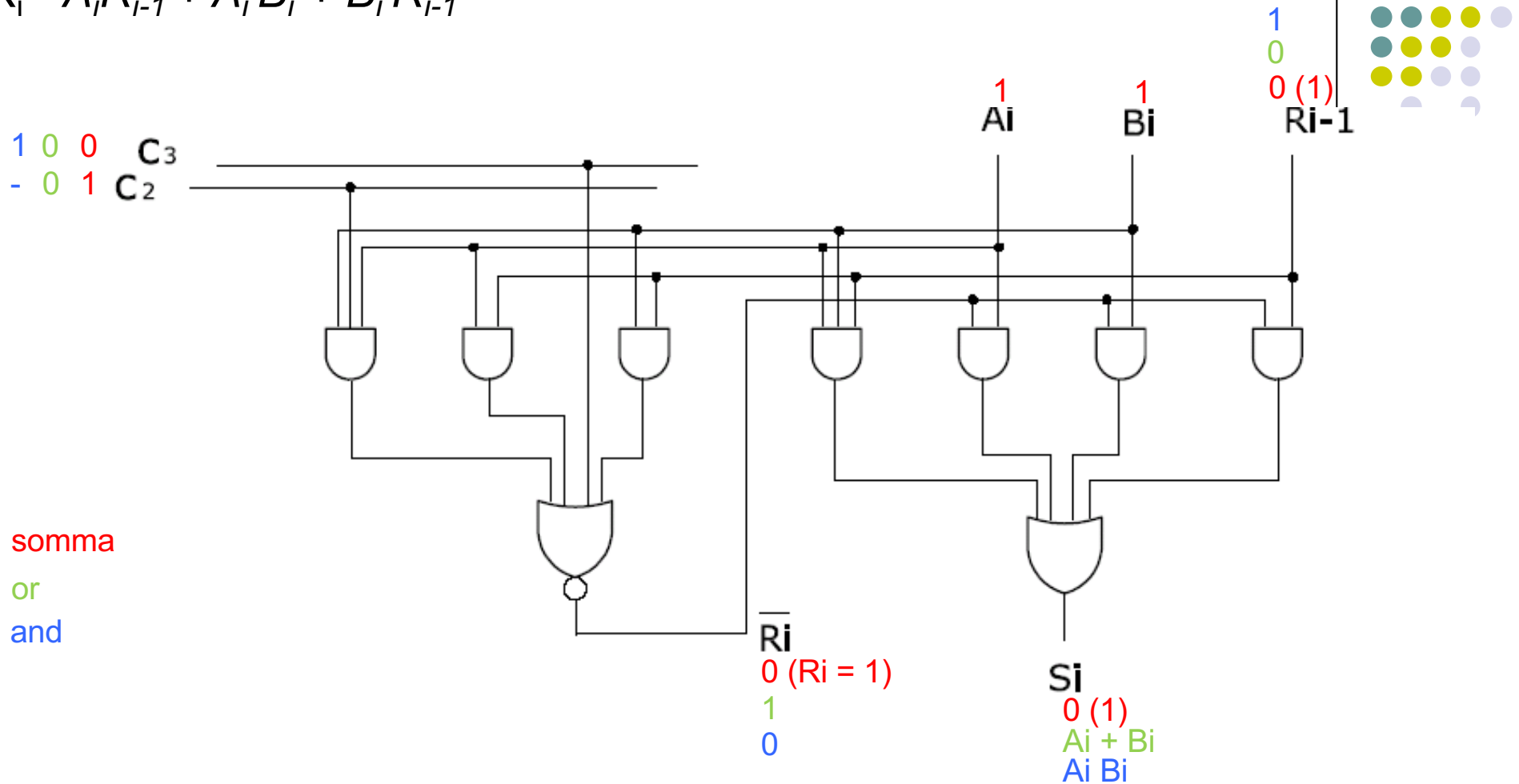
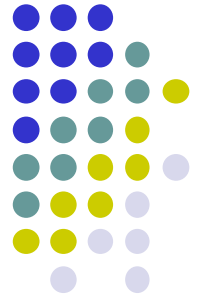
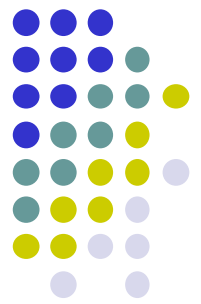


Figura 3.58 Disattivazione degli effetti dei riporti nel sommatore completo attraverso l'aggiunta di due linee di controllo che rendono disponibile l'AND e l'OR dei due ingressi



- Per $c_3=0$ e $c_2=1$ il valore R_i calcolato dalla rete risulta invariato, per cui si comporta esattamente come la precedente
- Se $c_3=1$ risulta $\neg R_i=0$ per ogni i , per cui il contributo a S_i delle 3 porte AND con $\neg R_i$ in ingresso è nullo, mentre la porta AND a 3 ingressi fornisce $A_i B_i \cdot 1$; la rete quindi calcola $A_i B_i$
- Se $c_3=0$, $c_2=0$ e $R_{-1}=0$ risulta $\neg R_i=1$ per ogni i , per cui la rete calcola $S_i = A_i \neg R_i + B_i \neg R_i + \neg R_i R_{i-1} + A_i B_i R_{i-1} = A_i \cdot 1 + B_i \cdot 1 + 1 \cdot 0 + A_i B_i \cdot 0 = A_i + B_i$
- Una sintesi del comportamento è mostrato nella tabella che segue
- La rete effettua altre operazioni oltre a quelle riportate; ad esempio con la configurazione di controllo “1-11-” calcola $A \neg B$

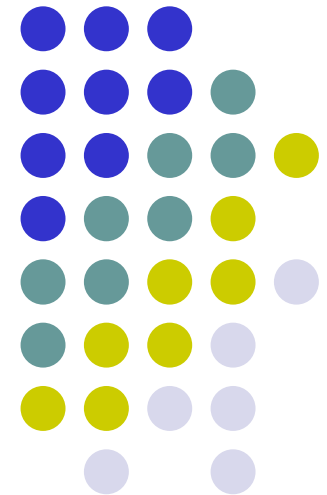


c_3	c_2	c_1	c_0	R_{-1}	Risultato	Commento
0	1	0	0	0	$S = 0 + B = B$	Selezione di B
0	1	0	0	1	$S = 0 + B + 1 = B + 1$	Incremento di B
0	1	0	1	0	$S = 0 + \overline{B} = \overline{B}$	Complementazione di B
0	1	0	1	1	$S = 0 + \overline{B} + 1 = -B$	Cambio segno di B
0	1	1	0	0	$S = A + B$	Somma $A+B$
0	1	1	1	1	$S = A + \overline{B} + 1 = A - B$	Differenza $A-B$
0	0	1	0	0	$S = A \text{ OR } B$	Somma logica di A e B
1	–	1	0	–	$S = A \text{ AND } B$	Prodotto logico di A e B

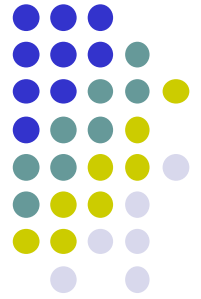
Tabella 3.5 Principali operazioni effettuate dalla ALU estesa.

Reti logiche

- Reti combinatorie
- Reti sequenziali
 - flip-flop
 - reti sequenziali sincrone ed asincrone
 - registri e contatori
 - bus
 - sintesi reti sequenziali sincrone

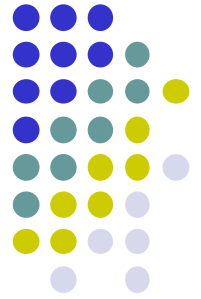


Reti sequenziali

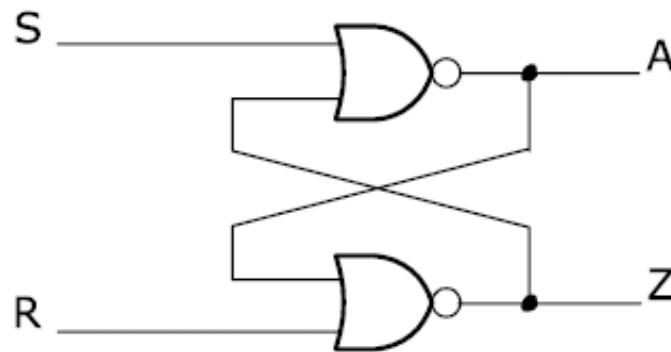


- In una rete sequenziale l'uscita (output) è funzione non solo degli ingressi (input) ma anche dello stato corrente
- Le reti sequenziali possono essere sincrone o asincrone
- Nel contesto dei calcolatori maggior rilevanza è assunta dalle sincrone

Latch

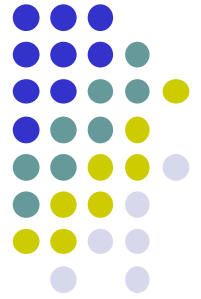


- E' l' elemento fondamentale di una rete sequenziale
- In tale elemento le uscite vengono riportate ingresso creando una "retroazione" che le influenza direttamente
- Latch di NOR:



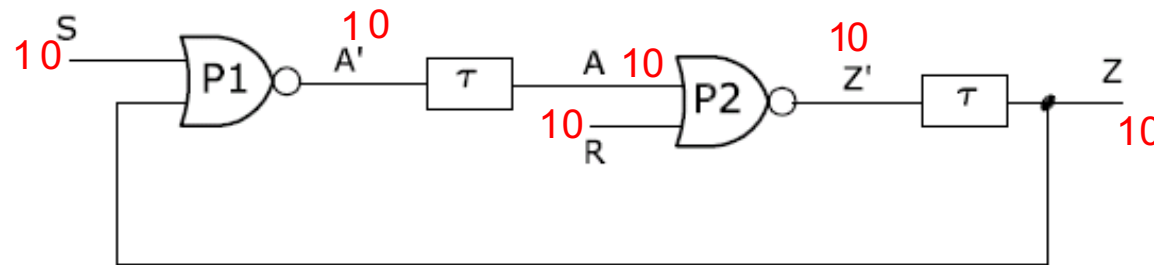
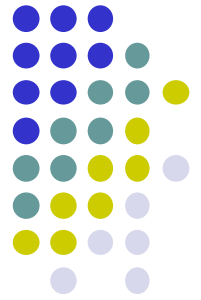
S	R	Z	A
0	0	0	1
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	0

Figura 4.1 Il Latch di NOR e relative relazioni di ingresso/uscita. Il termine anglosassone *latch* significa lucchetto, chiavistello ed esprime, in modo figurato, il comportamento della rete.



- Si noti che:
 - con input $S=0$ e $R=0$ sono possibili due uscite diverse, per cui la tabella precedente tale ingresso è riportato due volte
 - ad eccezione della configurazione in ingresso $S=1$ e $R=1$, l'uscita Z è sempre uguale al complemento di A
- Vediamo ora più in dettaglio il comportamento della rete
- A tal fine modelliamo una porta reale come una porta ideale con tempo di commutazione nullo, seguita da un elemento di ritardo che rappresenta il tempo di commutazione τ

Latch NOR con porte ideali

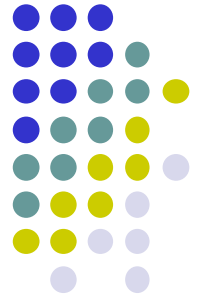


stabile
instabile

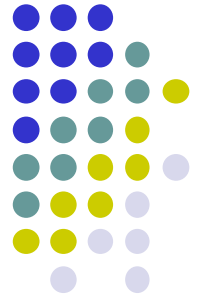
*2 alternative per
ritornare stabile*

Figura 4.2 Il latch di NOR ridisegnato con gli elementi di ritardo.

- Supponiamo una configurazione iniziale in cui $S=0$, $R=0$, $Z=1$, $A=0$
- Se al tempo t R diventa 1 , istantaneamente Z' diventa $0 \neq Z$
- In tal caso si parla di **stato instabile**
- All'istante $t+\tau$, Z passa a 0 e quindi A' a 1 ottenendo un altro stato instabile dovuto a $A' \neq A$ durante l'intervallo di tempo $[t+\tau, t+2\tau]$
- All'istante $t+2\tau$, A passa a 1 e, se l'input non cambia, neanche l'output cambia, ottenendo quindi un **stato stabile**

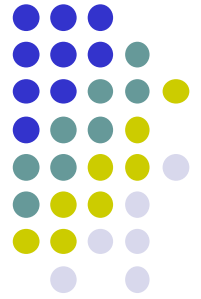


- Sempre in riferimento al latch della figura precedente, supponiamo che R torni a 0 (e rimanga $S=0$)
- E' facile verificare che nulla accade alla porta $P1$ e quindi che la rete rimane nello stato (valori delle uscite Z e A) precedente, finché l'ingresso non varia
- Se a questo punto diventa $S=1$, in modo analogo a quanto visto nel lucido precedente, dopo un transitorio lungo tempo 2τ , in uscita si avrà $A=0, Z=1$
- Si noti che in ogni caso, con $S=R=1$, lo stato stabile raggiunto sarà $A=Z=0$
- Se in tale situazione diventa $S=R=0$, allora a seconda di quale porta commuterà prima si andrà nello stato stabile $A=0, Z=1$ o $A=1, Z=0$, ossia non è possibile prevedere il comportamento della rete
- Per questo motivo la configurazione in ingresso $S=R=1$ non è ammessa
- Si noti infine che finora abbiamo assunto che gli ingressi varino solo quando la rete ha raggiunto uno stato stabile



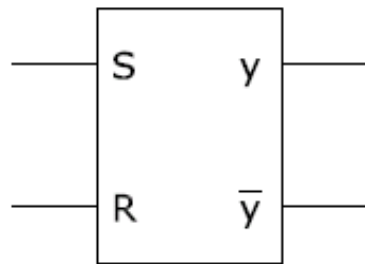
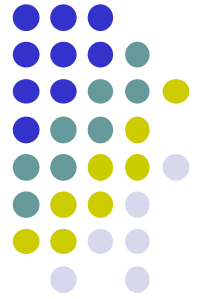
Concludendo:

- Se la configurazione in ingresso è $S=1, R=0$, allora in uscita si avrà $Z=1$
- Se $S=0, R=1$, allora in uscita si avrà $Z=0$
- Se $S=0, R=0$, l'uscita non varia, ossia sarà $Z=1$ se S ha avuto per ultimo valore 1, mentre $Z=0$ se l'ultimo ad avere valore 1 è stato R (NB: non posso avere entrambi valore 1)
- In altre parole, il latch di NOR ha memoria di quale dei due ingressi ha avuto per ultimo valore 1
- Quindi
 - portare S a 1 e poi a 0 corrisponde a memorizzare 1
 - portare R a 1 e poi a 0 corrisponde a memorizzare 0
- Un dispositivo con tali caratteristiche è un elemento binario di memoria chiamato *flip-flop*
- Nel caso specifico si parla di flip-flop *Set-Reset* o *SR* asincrono



- Come già accennato, l'uscita Z è sempre il complemento di A , ossia $Z = \neg A$
- L'uscita Z è considerata lo *stato* del flip-flop e indicata con y
- L'uscita A è indicata con $\neg y$
- Lo stato futuro Y può essere espresso in funzione degli ingressi e dello stato presente y :

- Lo stato futuro Y può essere espresso in funzione degli ingressi e dello stato presente y :



a)

S	R	y	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	-	-

b)

SR		00	01	11	10
y	0	0	0	—	1
	1	1	0	—	1

c)

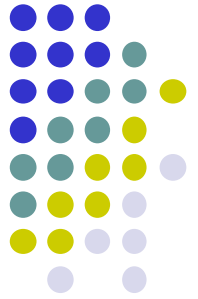
Y

Figura 4.4 Il flip-flop SR asincrono: a) schema convenzionale; b) tabella di verità; c) mappa di Karnaugh dello stato futuro.

- Da questo si deduce l'equazione di stato

$$Y = S + \neg R y$$
con il vincolo $SR = 0$

FUNZIONE DI ECCITAZIONE



Y	Z	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

\nearrow

S	R
0	0
0	1

\nearrow

S	R
0	0
1	0

F_u e F_{lop} S/R

Dal latch di NOR al flip-flop SR sincrono (Reti Sincrone)

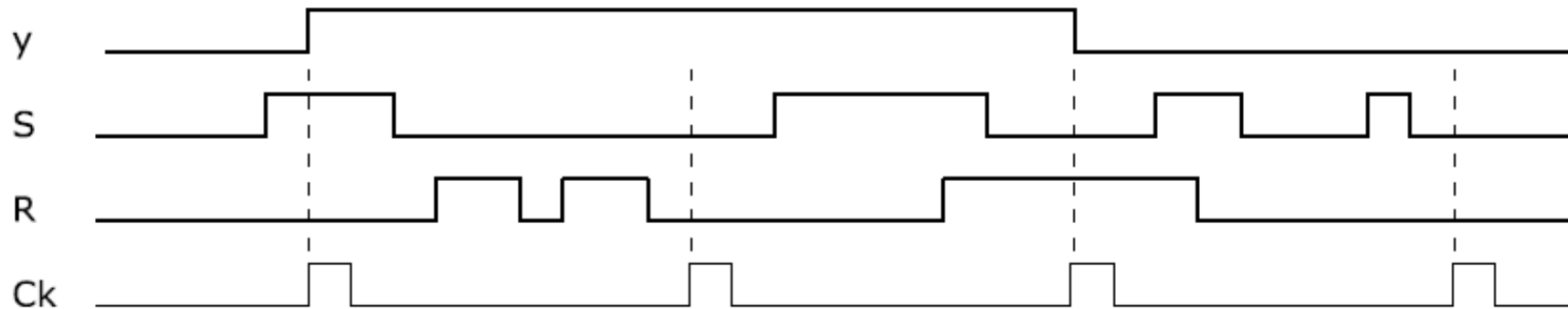
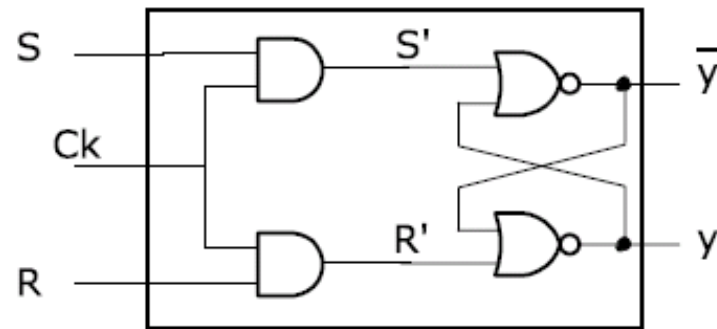
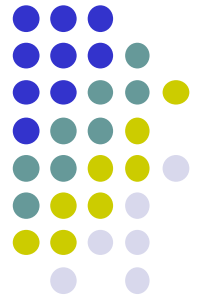
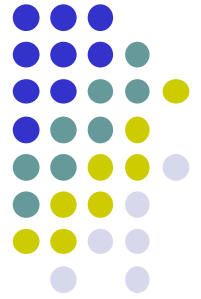


Figura 4.9 Trasformazione del latch di NOR in flip-flop sincrono e possibili andamenti temporali dei segnali. Si assume che il tempo di commutazione delle porte sia nullo.



- D' ora in poi faremo riferimento ad un segnale Ck periodico, anche se non è essenziale
- Mentre $Ck=0$, $S'=0$ e $R'=0$, per cui il latch non cambia stato
- Quando $Ck=1$, le porte AND sono *trasparenti* rispetto a S e R e il latch si comporta come quello asincrono
- Sia T il tempo di Ck , Δ_1 frazione di T in cui $Ck=1$, Δ_2 frazione di T in cui $Ck=0$
- Per un corretto funzionamento
 - S e R non devono variare durante Δ_1
 - Δ_1 deve essere sufficientemente ampio da far effettuare l' eventuale transizione di stato del latch
- Sotto queste ipotesi il latch si accorge degli ingressi S e R solo quando $Ck=1$, ossia la sua attività si *sincronizza* rispetto al segnale Ck

Flip-Flop SR (set,reset) sincrono – FFSR

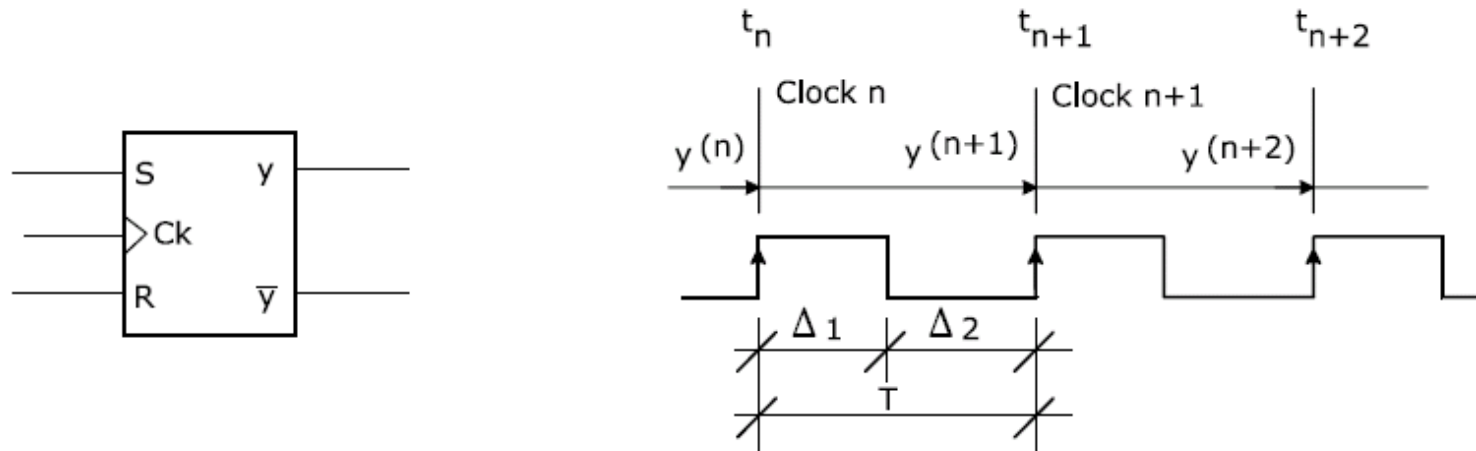
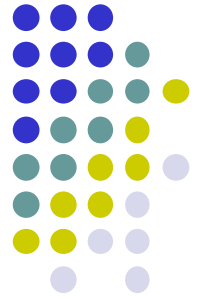


Figura 4.10 Schema convenzionale del flip-flop SR sincrono e interpretazione dello stato presente e dello stato futuro rispetto a successivi impulsi di clock.

- Stessa funzione di transizione di stato di quello asincrono
- Lo stato in cui si troverà il flip-flop all' impulso $(n+1)$ -esimo dipende dallo stato in cui si trova all' impulso n -esimo e dai valori di S e R in tale impulso
- In formule $y^{(n+1)} = S + \neg R y^{(n)}$ o semplicemente $Y = S + \neg R y$

Flip-Flop JK - FFJK

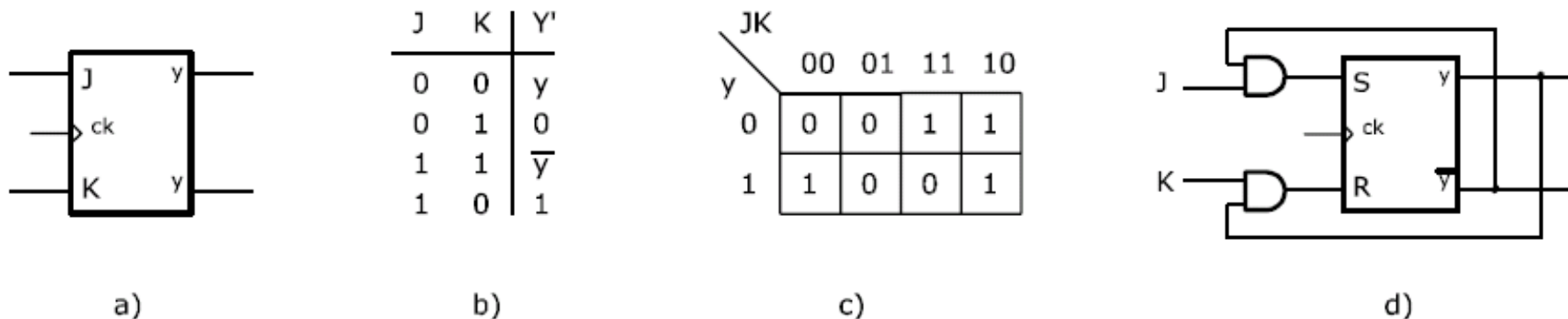
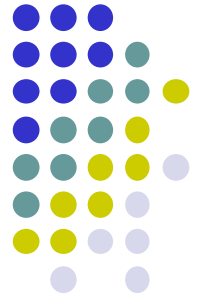
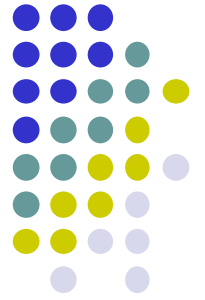


Figura 4.11 Il flip-flop JK: a) simbolo schematico; b) tabella dello stato futuro; c) mappa dello stato futuro; d) JK ottenuto da SR.

- Il comportamento è lo stesso del flip-flop SR tranne che per l'ingresso 11 che nel flip-flop JK è ammessa
- In tal caso il flip-flop cambia sempre stato all'impulso successivo, ossia passa da 0 a 1 o da 1 a 0
- In formule $Y = \neg y J + \neg K y$



FUNZ. DI ECCITAZIONE

FLIP FLOP J/K

y	\bar{y}	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

J	K
1	0
1	1

J	K
0	1
1	1

Arrows indicate the mapping from the first table's rows to the second and third tables:

- Row 1 (0, 0) maps to (1, 0)
- Row 2 (0, 1) maps to (1, 1)
- Row 3 (1, 0) maps to (0, 1)
- Row 4 (1, 1) maps to (1, 1)

Flip-Flop D (Delay) - FFD

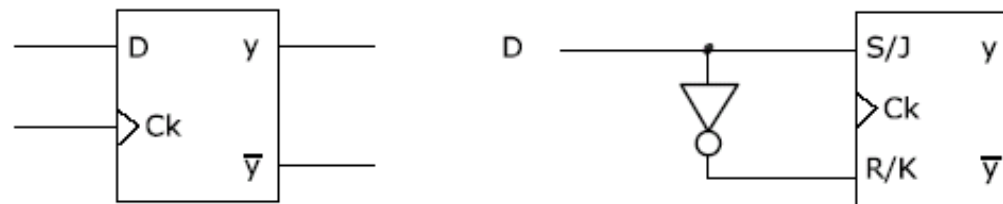
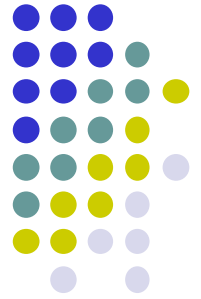


Figura 4.12 Schema convenzionale del flip-flop FFD e suo ottenimento da FFJK e FFJK.

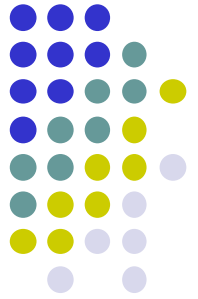
- Realizza un blocco di ritardo
- Lo stato al prossimo impulso sarà pari all'ingresso attuale
- In formule banalmente $Y=D$

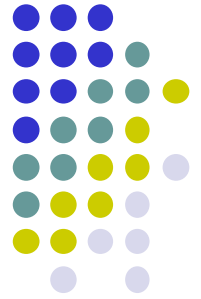
FUNZ. DI ECCITAZIONE

FLIP FLOP D



y	\bar{y}	d
0	0	0
0	1	1
1	0	0
1	1	1





Flip-Flop T (Toggle) - FFT

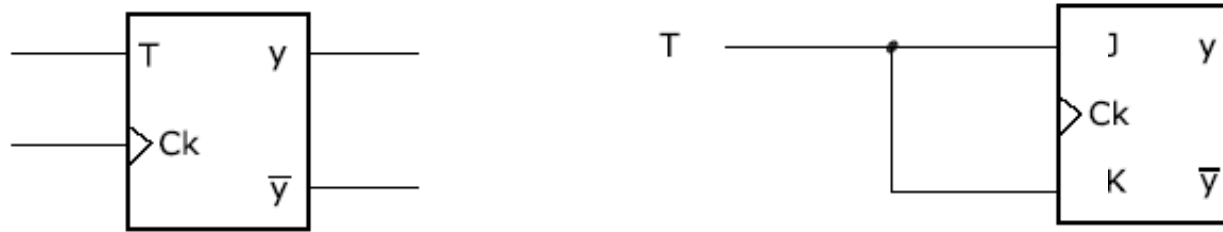
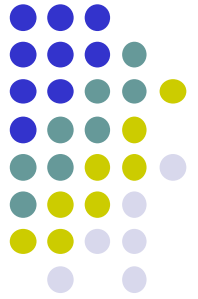


Figura 4.13 Schema convenzionale del flip-flop FFT e suo ottenimento da FFSR e FFJK.

- Il flip-flop cambia sempre stato all' impulso successivo quando l'ingresso $T=1$, resta nello stesso se $T=0$
- In formule $Y = \neg T y + T \neg y = T \oplus y$

PUNZ. DI ECCITAZIONE

FLIP FLOP T



y	y	t
0	0	0
0	1	1
1	0	1
1	1	0

Reti sequenziali sincrone: modello generale

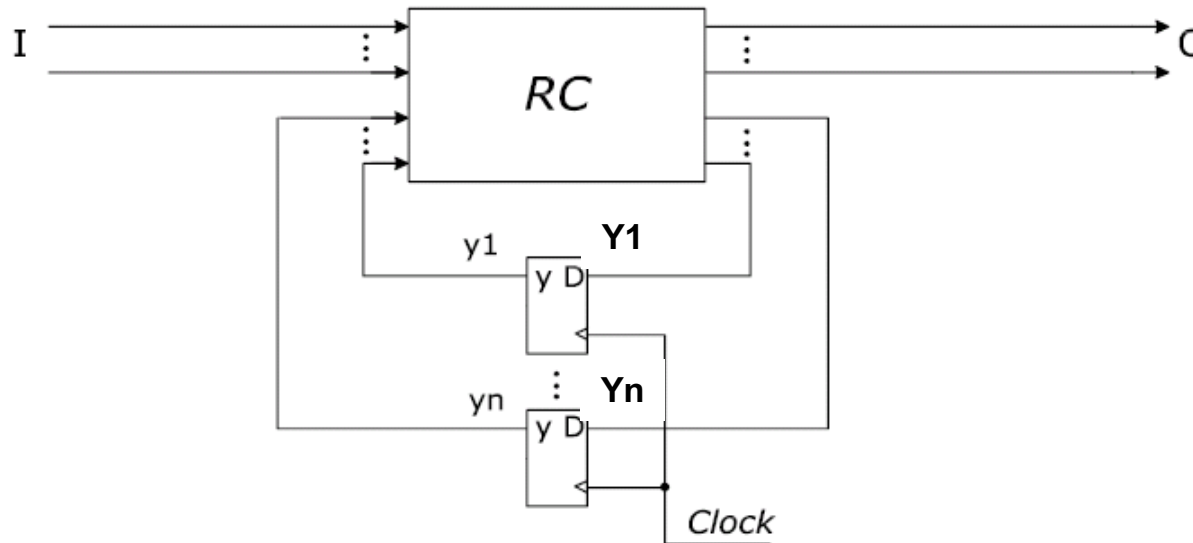
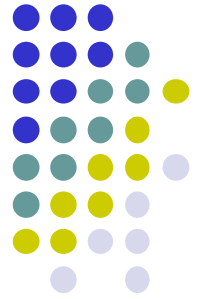
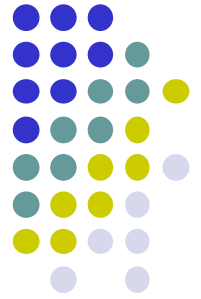


Figura 4.14 Modello di rete sequenziale sincrona.

- Procedono per passi scanditi dal clock
- L'ingresso e lo stato nel passo attuale determinano
 1. l'uscita attuale
 2. lo stato del prossimo passo
- Uscita attuale e stato futuro calcolati dalla parte combinatoria RC



- Chiaramente lo stato, cioè la memoria della rete, può essere mantenuto tramite qualunque tipo di flip-flop
- Si ottiene così un modello composto da
 - rete combinatoria in cui l'uscita è funzione dell'ingresso e dello stato presente
 - insieme di flip-flop che rappresentano lo stato

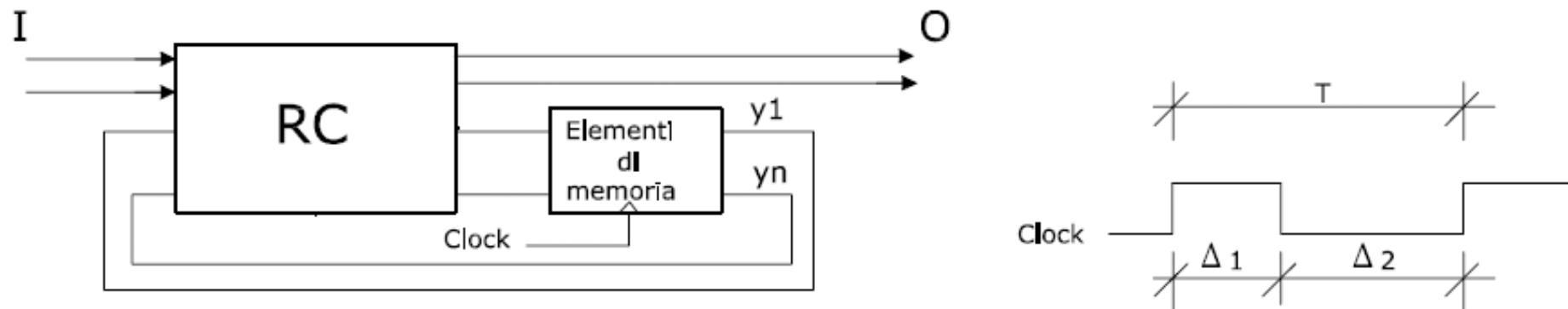
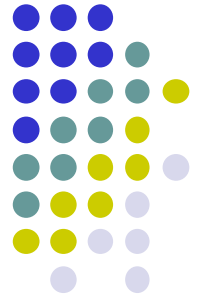


Figura 4.15 Evidenziazione degli elementi di memoria in una rete sincrona.



Indichiamo con

- x_1, \dots, x_n : variabili di ingresso
- z_1, \dots, z_m : variabili d'uscita
- y_1, \dots, y_l : variabili di stato presente
- Y_1, \dots, Y_l : variabili di stato futuro

con x_i, z_i, y_i, Y_i appartenenti a $\{0, 1\} \quad \forall i$

L'insieme delle $N=2^n$ n -uple (x_1, \dots, x_n) in ingresso definisce l'alfabeto di ingresso $I = \{I_1, \dots, I_N\}$

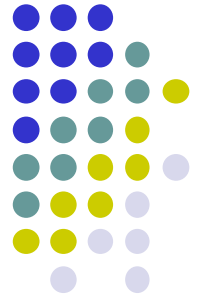
Analogamente

$O = \{O_1, \dots, O_M\}$ con $M=2^m$

:alfabeto di uscita

$S = \{S_1, \dots, S_L\}$ con $L=2^l$

:alfabeto di stato



Macchina sequenziale: quintupla $M=(I,O,S,f,g)$ dove

- I, O, S sono rispettivamente gli alfabeti di ingresso, di uscita e di stato
- f e g sono due funzioni rispettivamente:
$$f : S \times I \rightarrow O \quad (\text{uscita attuale})$$
$$g : S \times I \rightarrow S \quad (\text{stato futuro})$$

Poiché l'alfabeto di stato è finito, la macchina ha memoria finita

In questo caso si parla anche di *macchine* ovvero *automi a stati finiti*

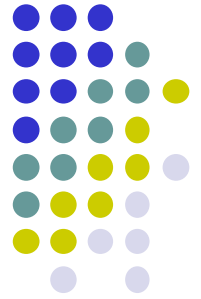
Modello di *Mealy*:

- è quello appena visto
- l'uscita dipende sia dall'ingresso che dallo stato corrente, ossia $f : S \times I \rightarrow O$

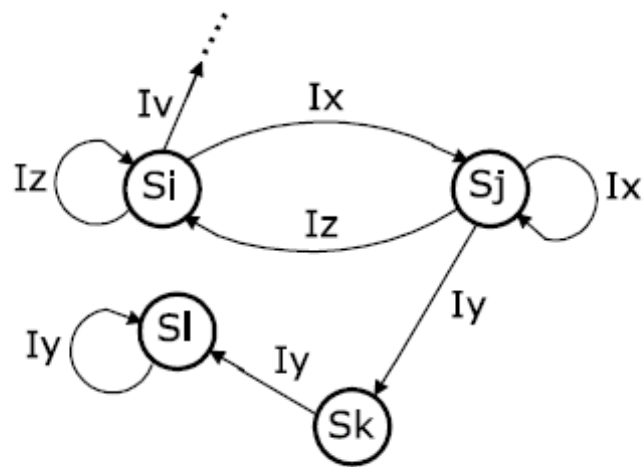
Modello di *Moore*:

- l'uscita dipende solo dallo stato corrente, ossia $f : S \rightarrow O$

Rappresentazione funzioni di stato e uscita



- Due modi per descrivere una macchina a stati finiti:
 1. diagrammi di stato
 2. tabelle di flusso
- Entrambi descrivono le funzioni f e g che definiscono l'automa
- Tenendo conto delle definizioni del lucido precedente, scriviamo sinteticamente
 - $O=f(I, S_p)$, ossia O è l'output che si ha con l'ingresso I e lo stato presente S_p
 - $S_f=g(I, S_p)$, ossia S_f è lo stato futuro in cui si andrà con l'ingresso I e lo stato presente S_p
- Vediamo graficamente i due formalismi



a)

	I1	...	Ix	Iy	Iz	...	IN
S1							
::							
Si			Sj		Ⓢi		
Sj			Ⓢj	Sk	Si		
Sk				Sl			
Sl				Sl			
SL							

b)

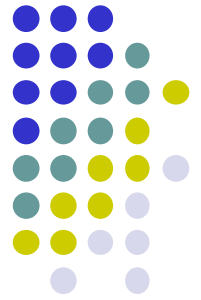


Figura 4.6 Diagrammi di stato e tabelle di flusso: a) esempio di (parte di) diagramma di stato; b) tabella di flusso corrispondente.

- Diagramma di stato: intuitivo ma non formalmente elaborabile
- Tabelle di flusso: facilmente riconducibili a funzioni booleane in forma tabellare
- In entrambi i formalismi vengono riportate le uscite, ma bisogna distinguere tra modello di Mealy ($O_{jz}=f(I_z, S_j)$) e modello di Moore ($O_j=f(S_j)$)

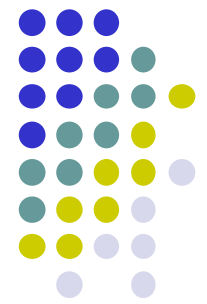
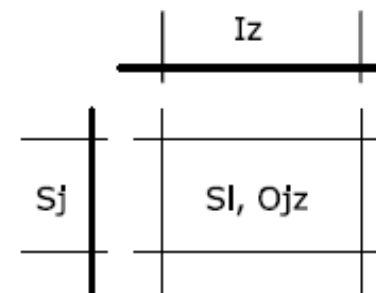
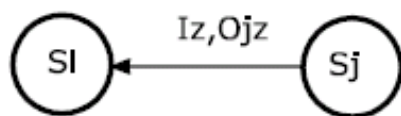


Figura 4.7 Rappresentazione delle uscite col modello di Mealy.

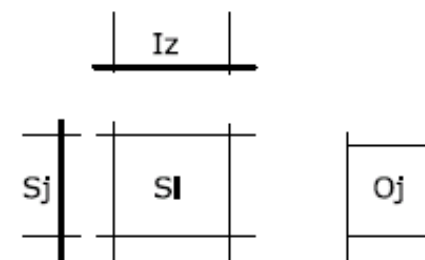
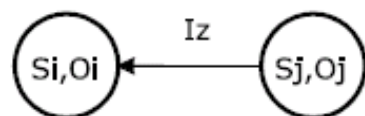
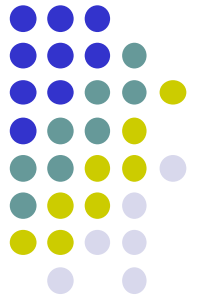


Figura 4.8 Rappresentazione delle uscite col modello di Moore.

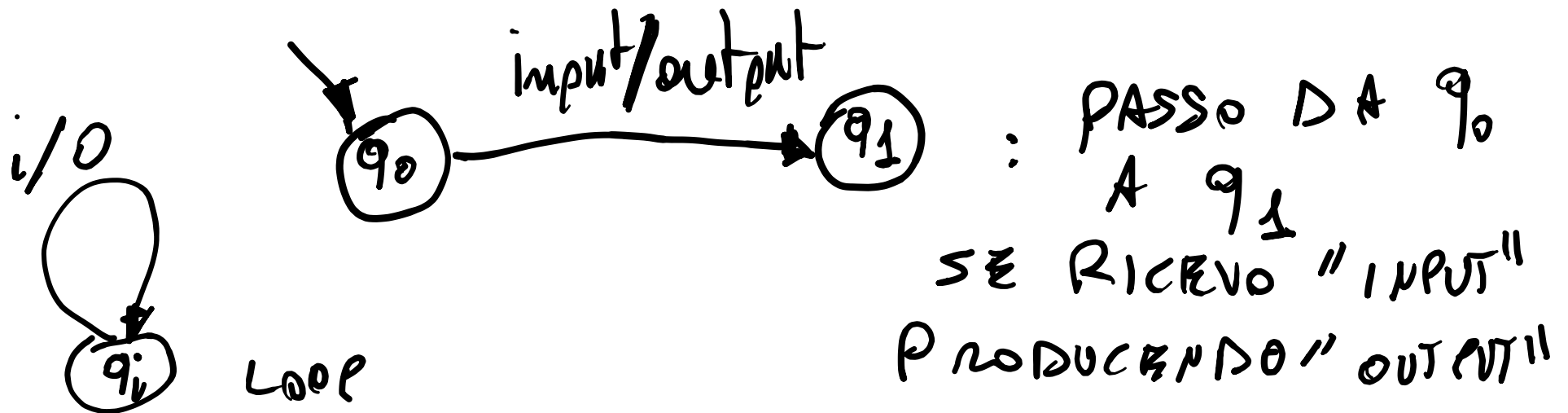
NOTAZIONE GRAFICA DEL DIAGRAMMA DI STATO DI UNA RETE

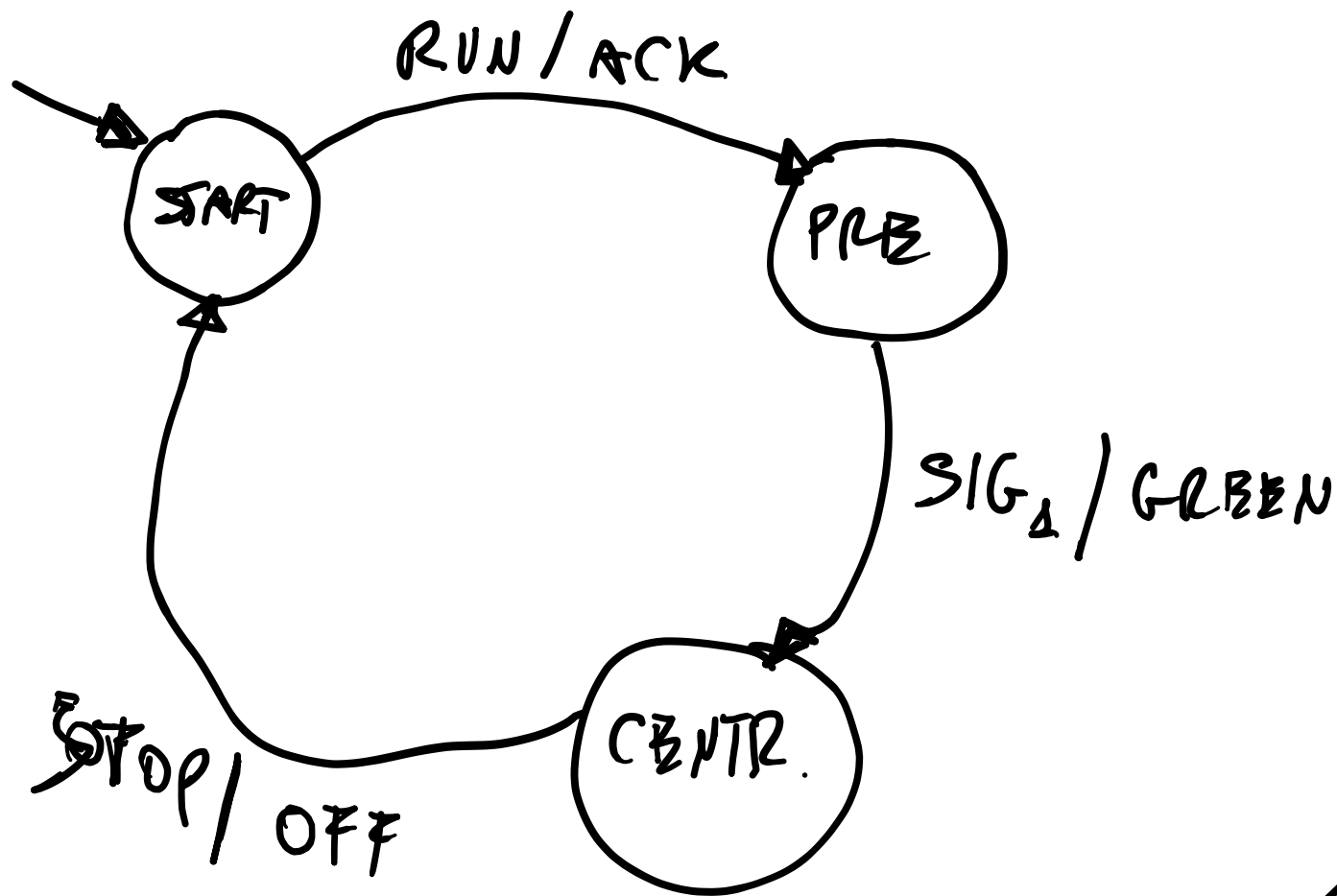
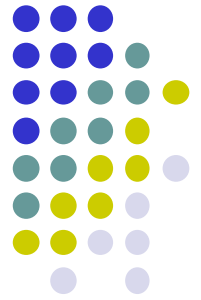


○ : STATI

→ ○ : STATO INIZIALE

→ : TRANSIZIONI





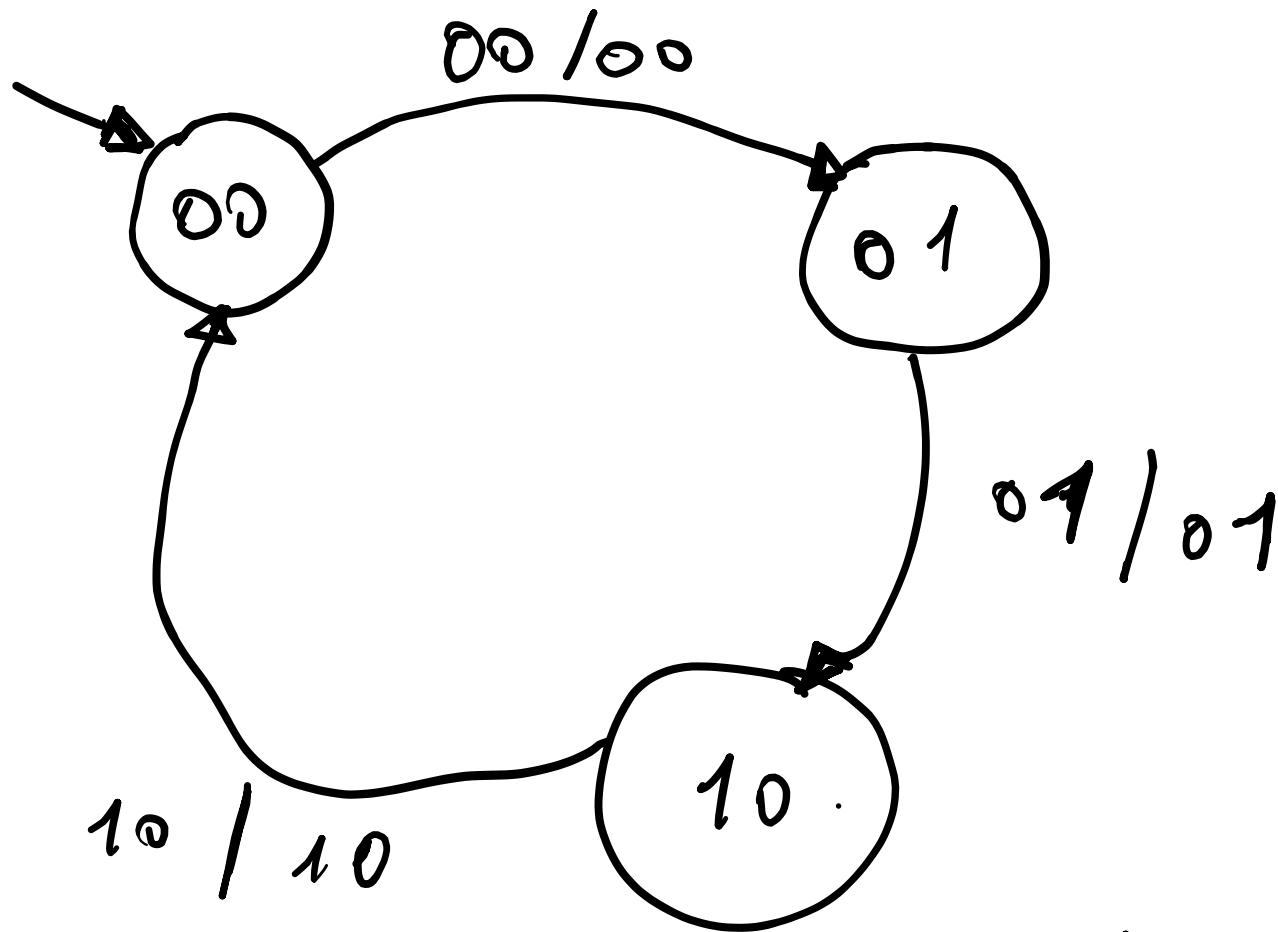
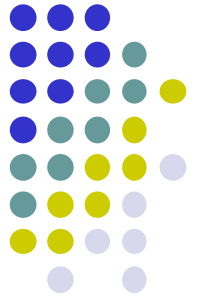
3 STATI : { START, PRB, CENTR }

I = { RUN, SIG_A, STOP }

O = { ACK, GREEN, OFF }

2 bit

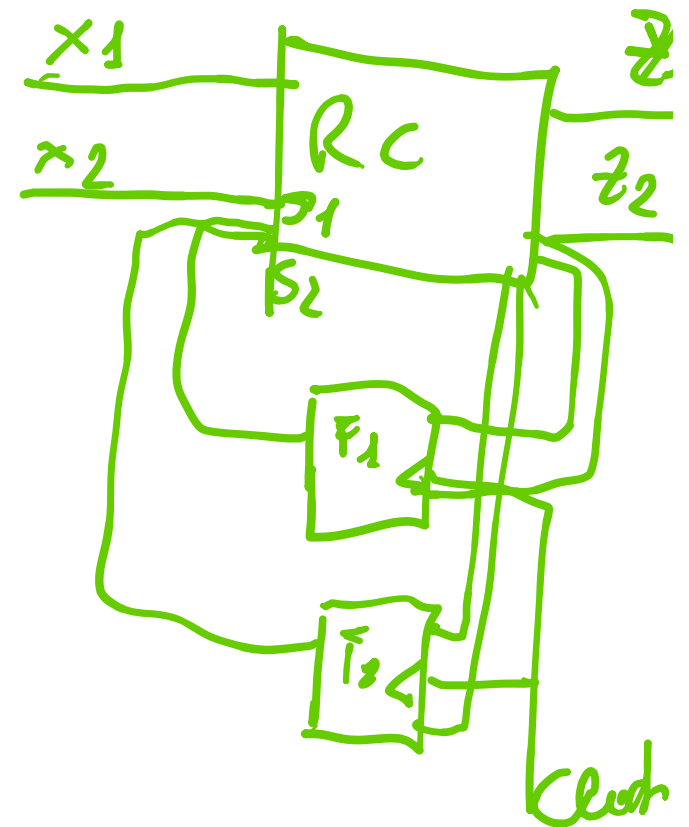
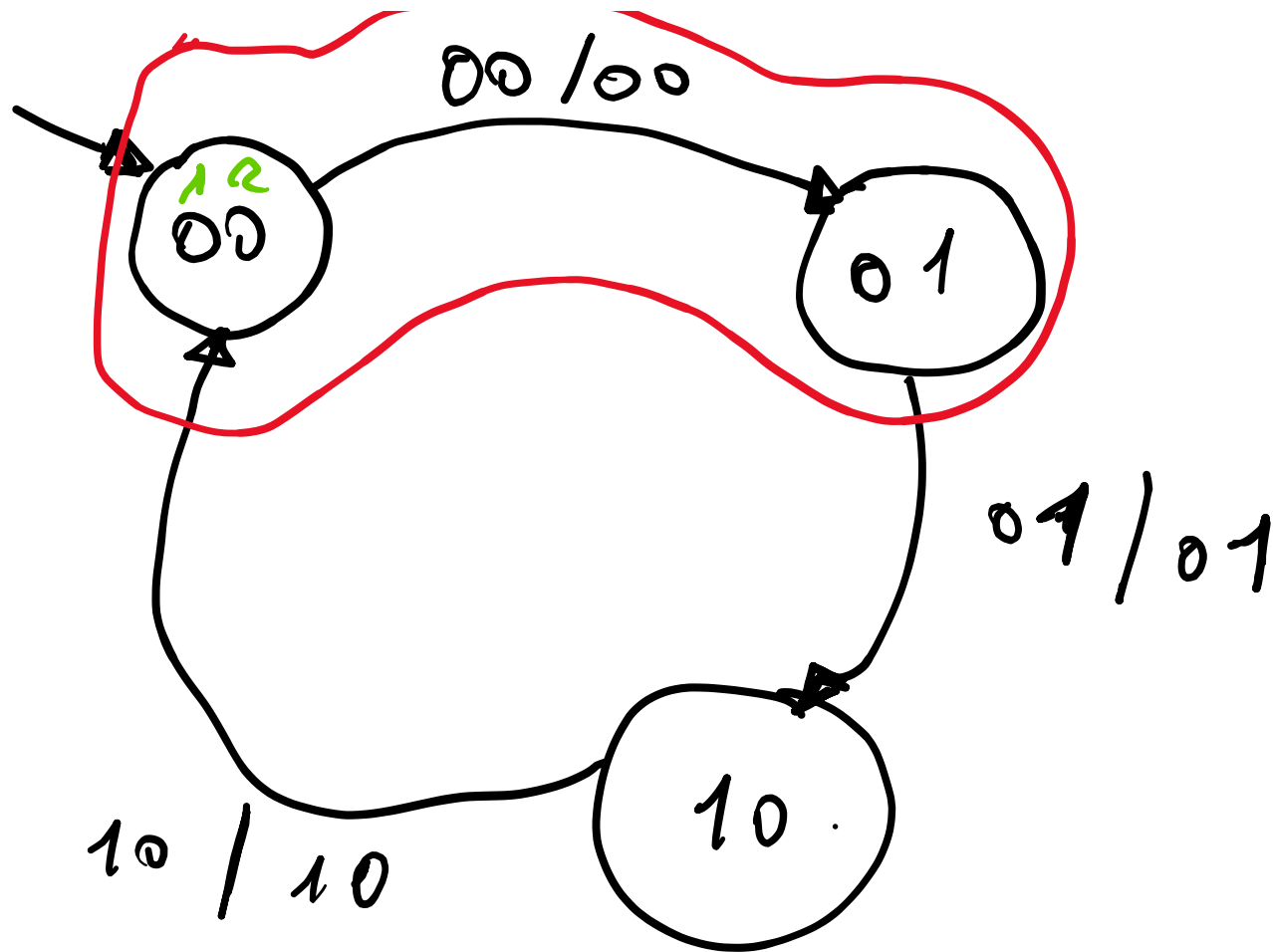
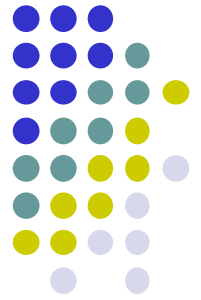




3 STATI : $\begin{matrix} 00 & 01 & 10 \\ \text{START}, & \text{PRE}, & \text{CENTR} \end{matrix}$

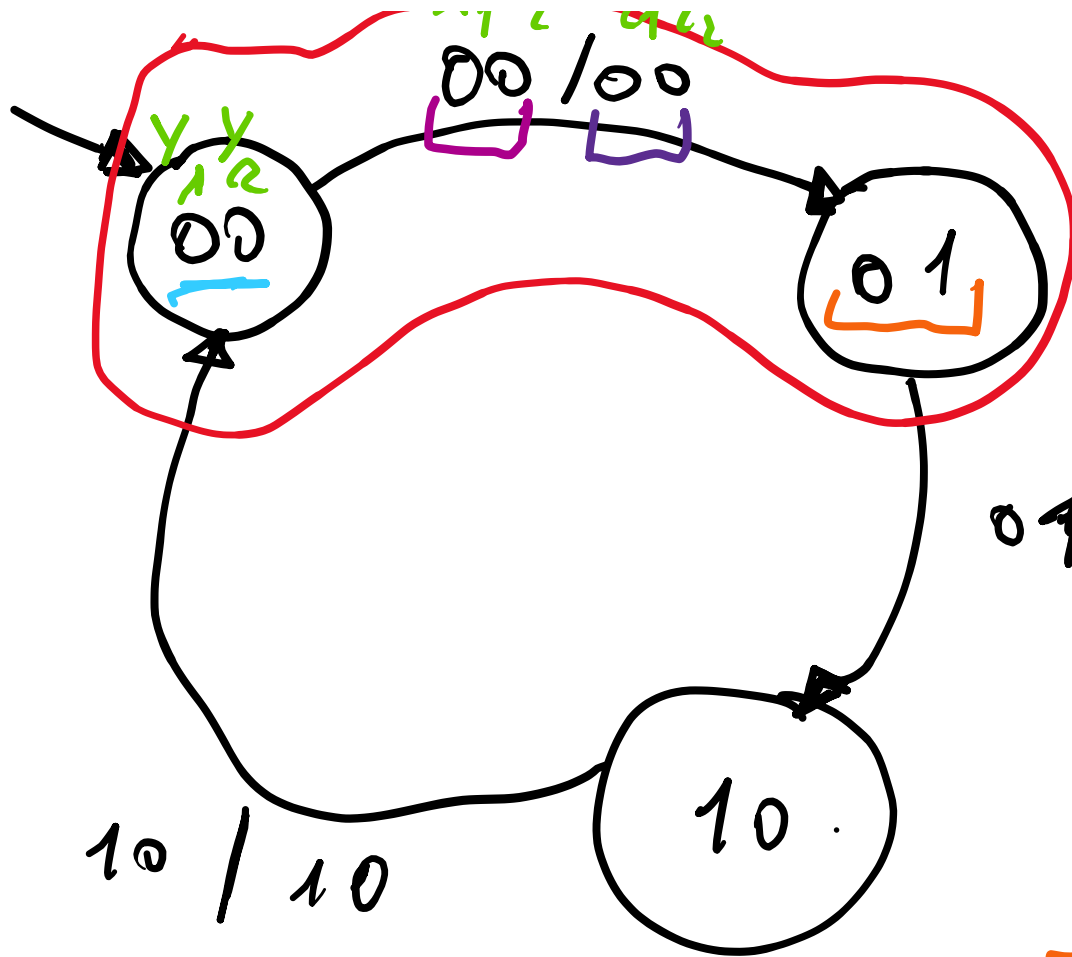
$\bar{I} = \begin{matrix} 00 & 01 & 10 \\ \text{RUN}, & \text{SIG}, & \text{STOP} \end{matrix}$

$O = \begin{matrix} 00 & 01 & 10 \\ \text{ACK}, & \text{GRZBN}, & \text{OFF} \end{matrix}$



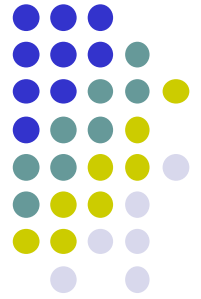
$$z_1 = f(x_1, x_2, s_1, s_2)$$

$$z_2 = f(x_1, x_2, s_1, s_2)$$



$$z_1 = f\left(\begin{matrix} x_1 x_2, \\ y_1 y_2 \end{matrix}\right)$$

[illegible]



Esercizio

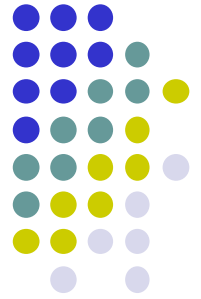
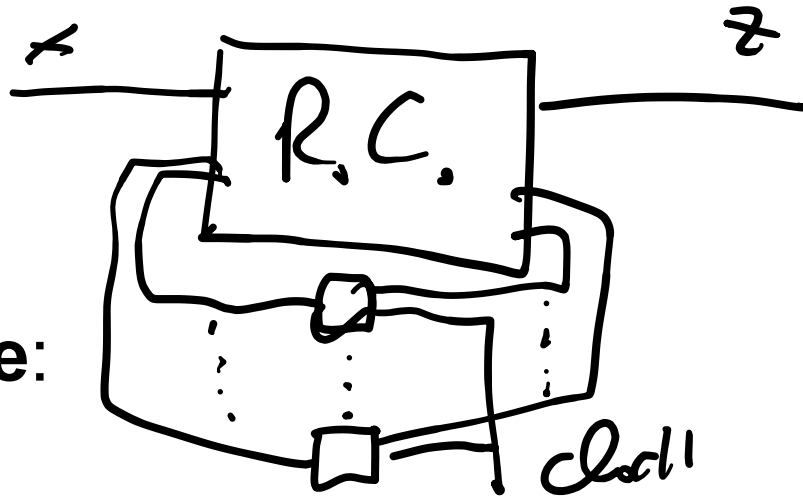
Descrizione verbale:

Progettare una rete sequenziale a singolo ingresso (x) e singola uscita (z) che realizzi la logica di controllo della gettoniera di una macchinetta del caffè, considerando che

1. il costo del caffè è di 40€
2. è possibile introdurre monete da 10€ e 20€ .
3. la macchina non dà resto ma ricorda le somme in eccesso

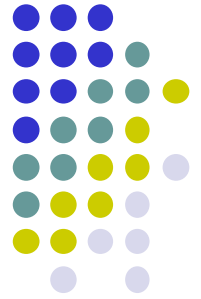
Esercizio

Descrizione verbale:



Progettare una rete sequenziale a singolo ingresso (x) e singola uscita (z) che realizzi la logica di controllo della gettoniera di una macchinetta del caffè, considerando che

1. il costo del caffè è di 40€c
2. è possibile introdurre monete da 10€c e 20€c.
3. la macchina non dà resto ma ricorda le somme in eccesso



Esercizio

Descrizione verbale:

Progettare una rete sequenziale a singolo ingresso (x) e singola uscita (z) che realizzi la logica di controllo della gettoniera di una macchinetta del caffè, considerando che

1. il costo del caffè è di 40€
2. è possibile introdurre monete da 10€ e 20€ .
3. la macchina non dà resto ma ricorda le somme in eccesso

