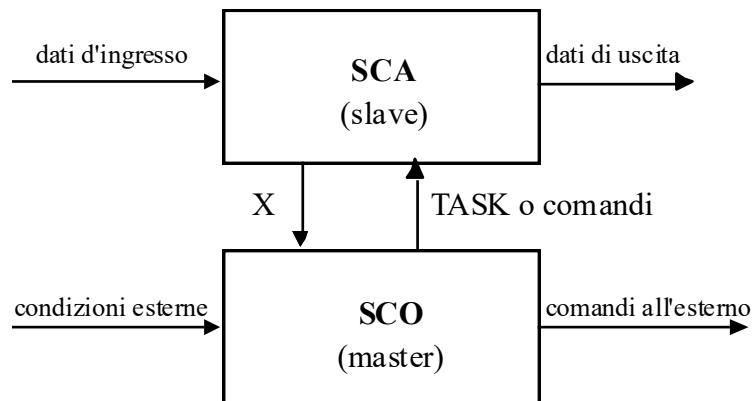


# SISTEMI DIGITALI COMPLESSI



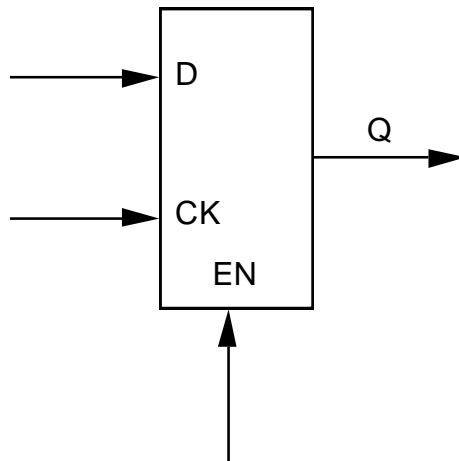
**Sistema digitale complesso suddiviso in SCO-SCA**

Il procedimento di sintesi di un sistema digitale, può essere suddiviso nei seguenti passi:

1. Specifica del problema.
  2. Individuazione di un algoritmo di soluzione.
  3. Progetto di un SCA atto a supportare l'algoritmo.
  4. Definizione di un SCO che implementa l'algoritmo.
  5. Valutazione del sistema: se le prestazioni rispondono alle specifiche del problema si passa al punto 6. Altrimenti si verifica se è possibile definire un altro SCA: in caso positivo si modifica il SCA e si torna al punto 4; se no, si passa al punto 2.
  6. Sintesi del sistema e verifica del corretto funzionamento.
-

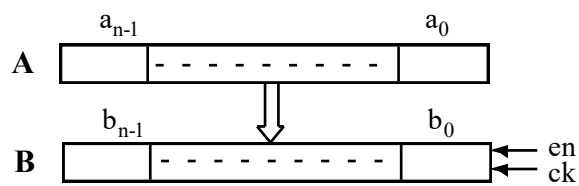
## Sottosistema di Calcolo (SCA)

*Registri, segnali e operazioni*



**Cella di un registro**

Un registro viene indicato con R, il suo contenuto con (R).

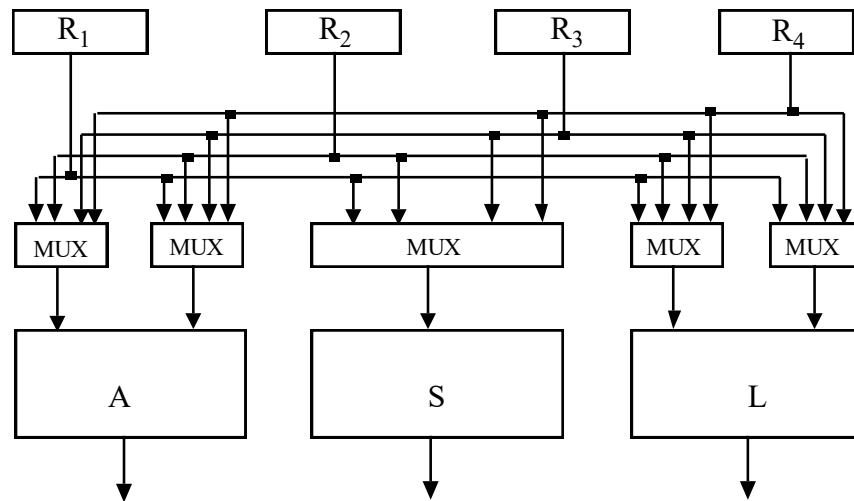


**Trasferimento dati tra due registri**

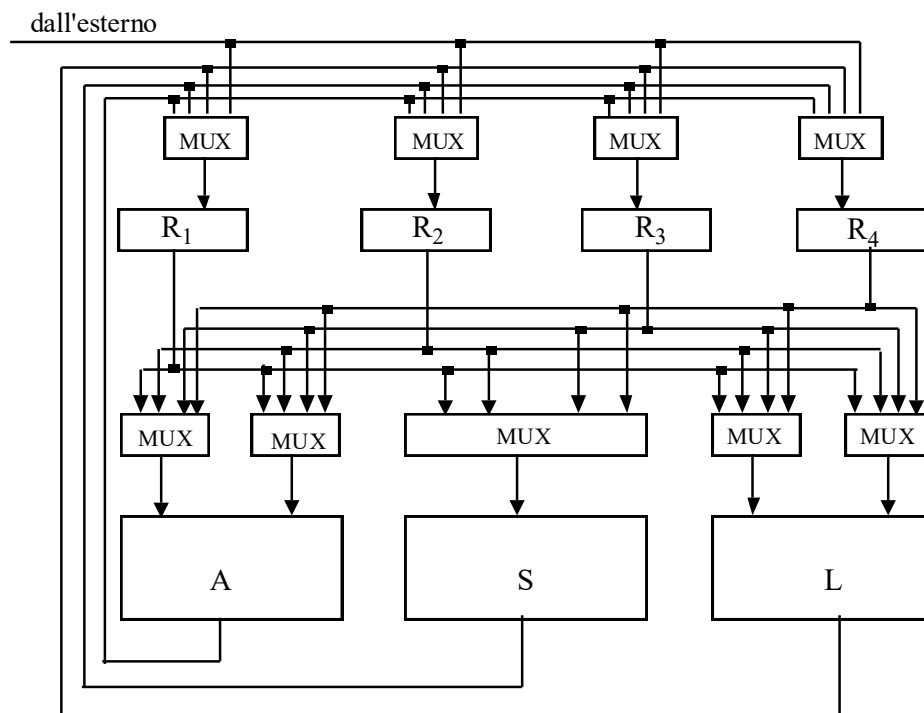
## Operazioni tra dati contenuti in registri

- OR A, B:  $(A) \cup (B) = Y$  somma logica bit a bit  
esempio:  $(0110) \cup (1100) = 1110$
- AND A, B:  $(A) \cap (B) = Y$  prodotto logico bit a bit  
esempio:  $(0110) \cap (1100) = 0100$
- NOT A:  $(A) = Y$  complementazione dei bit  
esempio:  $(0110) = 1001$
- EXOR A, B:  $(A) \oplus (B) = Y$  or esclusivo bit a bit  
esempio:  $(0110) \oplus (1100) = 1010$
- ADD A, B:  $(A) + (B) = Y$  somma aritmetica  
esempio:  $(0110) + (1100) = 0010$
- SUB A, B:  $(A) - (B) = Y$  sottrazione  
esempio:  $(0110) - (1100) = 1010$
- SR A, k: shift right (scalamento a destra) di k posizioni  
 $a_i \rightarrow a_{i-k} \quad i = n-1, \dots, k$   
 $0 \rightarrow a_h \quad h = n-1, \dots, k$
- SL A, k: shift left (scalamento a sinistra) di k posizioni  
 $a_i \rightarrow a_{i+k} \quad i = 0, 1, \dots, n-k$   
 $0 \rightarrow a_h \quad h = 0, 1, \dots, n-k$
- RR A, k: rotate right (rotazione a destra) di k posizioni  
 $a_i \rightarrow a_{i-k} \quad i = n-1, \dots, k$   
 $a_h \rightarrow a_{n-k+h} \quad h = 0, 1, \dots, k-1$
- RL A, k: rotate left (rotazione a sinistra) di k posizioni  
 $a_i \rightarrow a_{(i+k) \bmod n} \quad i = 0, 1, \dots, n$
- INC A, k:  $(A) + k \rightarrow A$
- DEC A, k:  $(A) - k \rightarrow A$
- EXCH A, B:  $(B) \rightarrow A \quad (A) \rightarrow B$

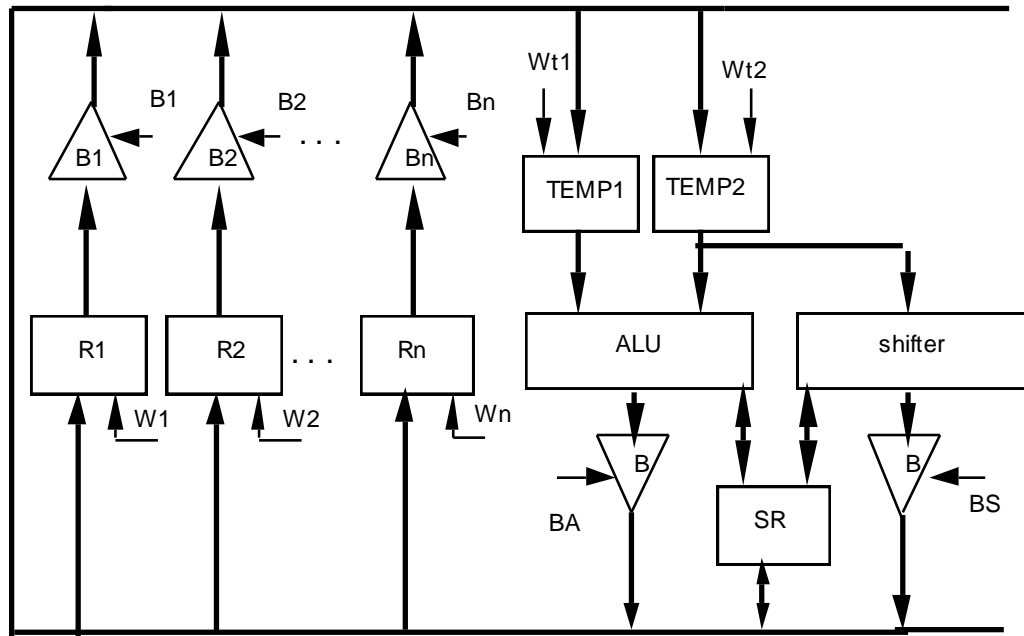
## Interconnessione registri-circuiti di calcolo



**Esempio di interconnessione tra registri e circuiti di calcolo**



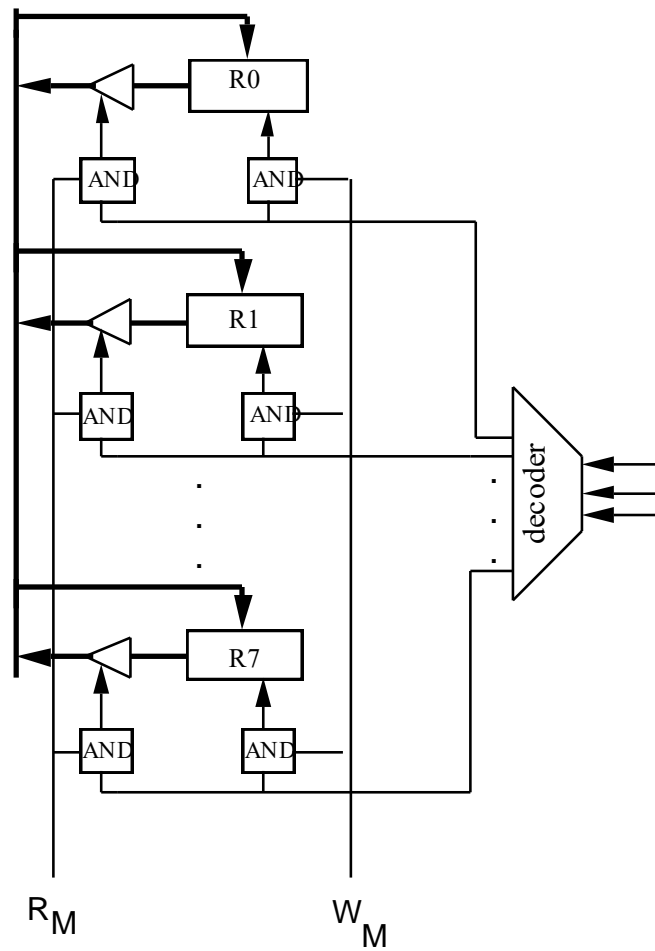
**Interconnessione tra registri e circuiti di calcolo**



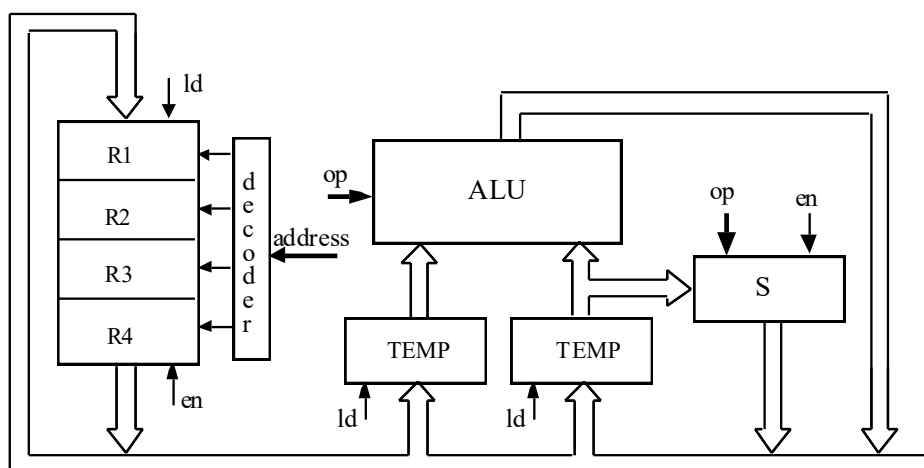
→ singola linea

→ linee multiple

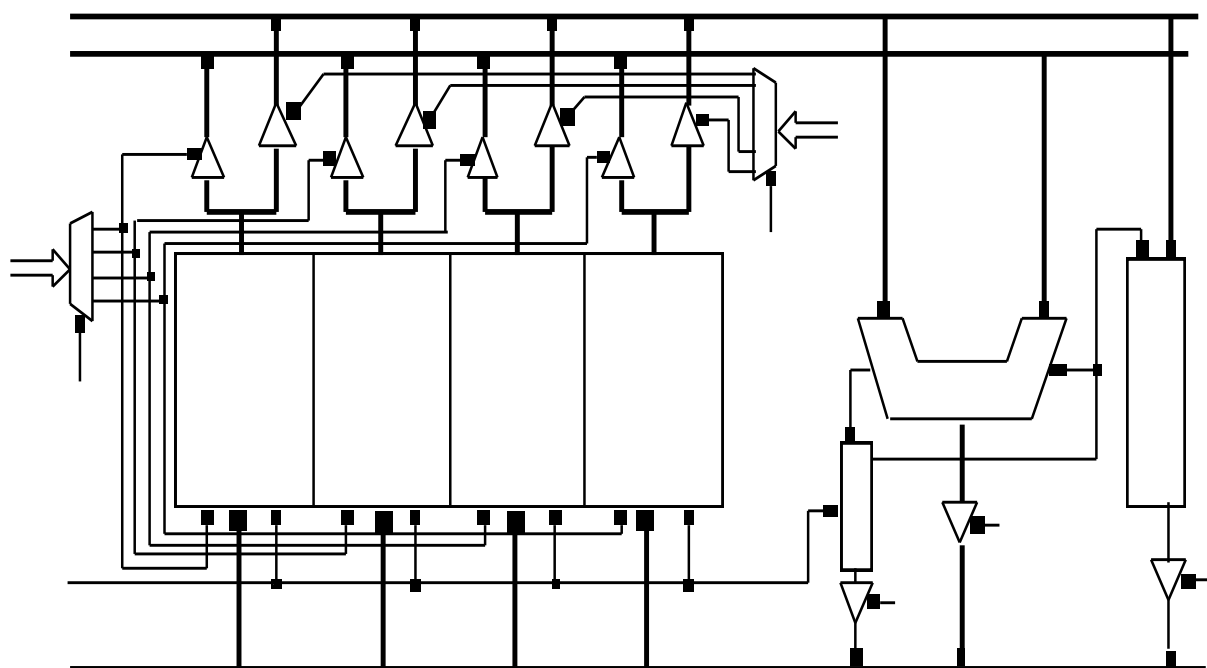
**Interconnessione tra registri e circuiti di calcolo tramite bus**



**Organizzazione vettoriale dei registri**



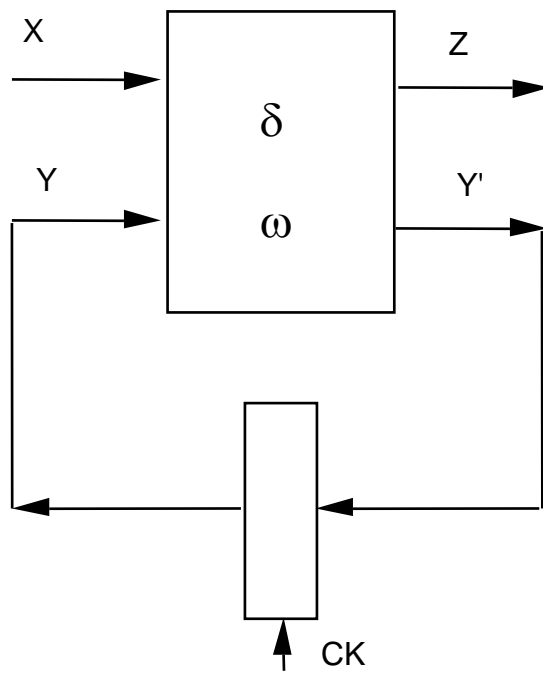
**Interconnessione a bus tra registri e circuiti di calcolo**



**Esempio di interconnessione con tre bus**

## Sottosistema di Controllo (SCO)

### *La microprogrammazione*



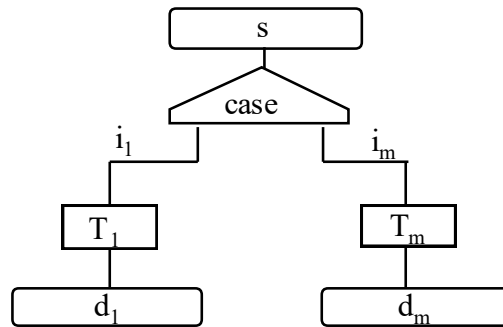
Rappresentazione Algorithm State Machine

Implementazione di  $\delta$  e  $\omega$  tramite ROM

} = microprogrammazione



## Modello di Mealy



### Modulo elementare della rappresentazione ASM per macchine di tipo Mealy

La corrispondente microistruzione ha un formato del tipo:

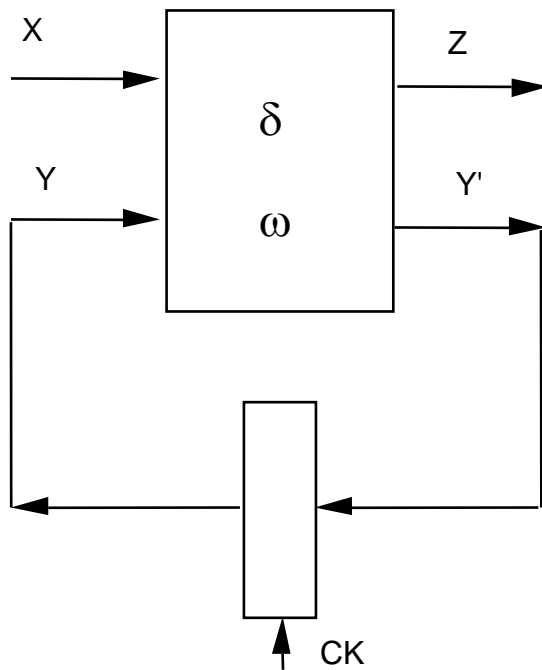
$\mu_i : C_1(T_1, \mu_{i1}), C_2(T_2, \mu_{i2}), \dots, C_m(T_m, \mu_{im})$

dove

$C_1, C_2, \dots, C_m$  ( $m \leq 2^k$ ,  $k = n$ ) sono le condizioni derivanti dalle variabili di decisione  $x_1, x_2, \dots, x_n$ ;

$T_1, T_2, \dots, T_m$  sono le corrispondenti azioni da effettuare;

$\mu_{i1}, \mu_{i2}, \dots, \mu_{im}$  le microistruzioni successive a  $\mu_i$ .



### Modello strutturale standard

SS	TASK
----	------

Struttura della parola di ROM

Se

- **n** è il numero delle variabili di ingresso
- **m** è il numero delle variabili di stato

allora

$$\text{numero di parole} = 2^{n+m}$$

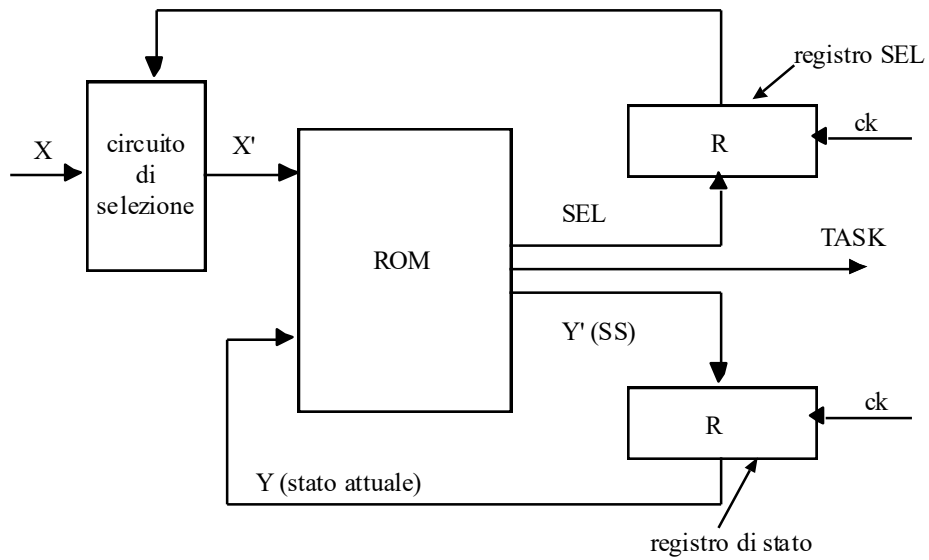
p.e., se

- **n** = 20
- **m** = 4

allora

$$\text{numero di parole} = 2^{24} \text{ (16 Mega)}$$

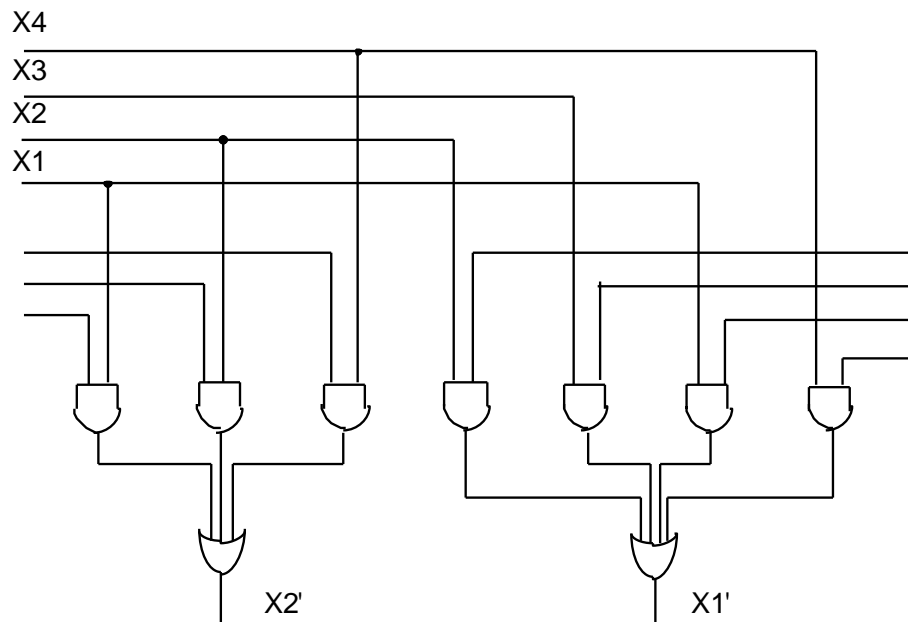
**ECCESSIVO!!!!**



**Struttura del SCO nel caso di modello di Mealy**

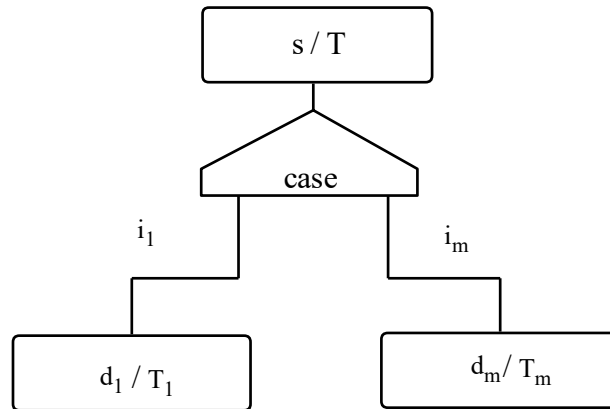
SEL	SS	TASK
-----	----	------

**Struttura della parola di ROM nel caso di modello di Mealy con selezione**



**Esempio di circuito di mascheramento non codificato**

## Modello di Moore



**Modulo elementare della rappresentazione ASM per macchine di tipo Moore**

La corrispondente microistruzione ha un formato del tipo:

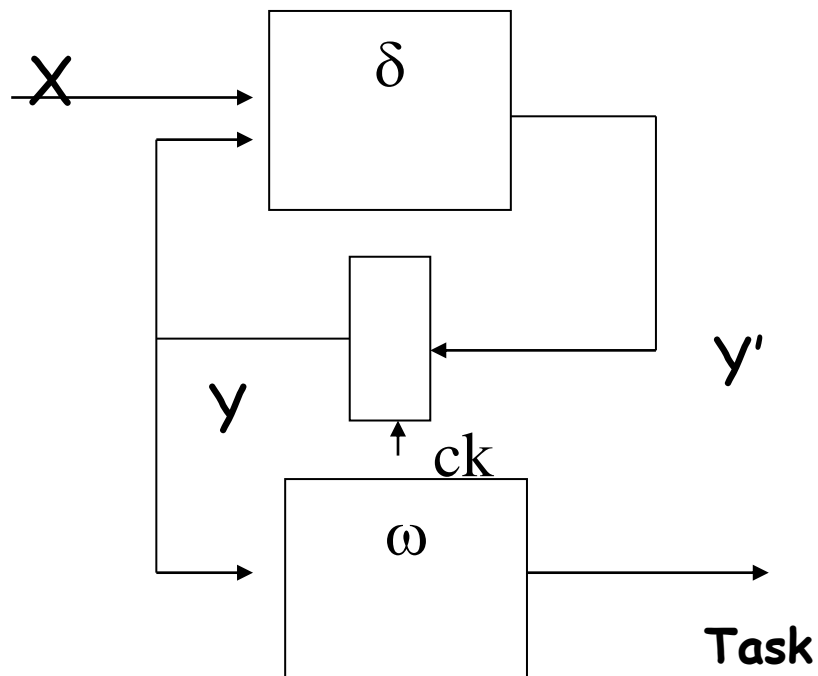
$\mu_i : T_i; C_1(\mu_{i1}), C_2(\mu_{i2}), \dots, C_m(\mu_{im})$

dove

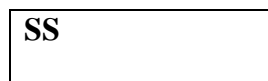
$C_1, C_2, \dots, C_m$  ( $m \leq 2^k$ ,  $k = n$ ) sono le condizioni derivanti dalle variabili di decisione  $x_1, \dots, x_n$ ;

$T_i$  è l'azione da effettuare;

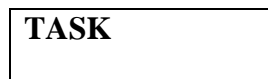
$\mu_{i1}, \dots, \mu_{im}$  le microistruzioni successive a  $\mu_i$ .



**Modello strutturale di Moore standard**



**Struttura della parola di ROM (che implementa la  $\delta$ )**



**Struttura della parola di ROM (che implementa la  $\omega$ )**

Se

- **n** è il numero delle variabili di ingresso
- **m** è il numero delle variabili di stato

allora

**numero delle righe della prima ROM (che impl. la  $\delta$ ) =  $2^{n+m}$**

p.e., se

- **n** = 20
- **m** = 4

allora

**numero delle righe =  $2^{24}$  (16 Mega)**

**ECCESSIVO!!!!!!**

## Soluzione che permette l'uso di una unica ROM

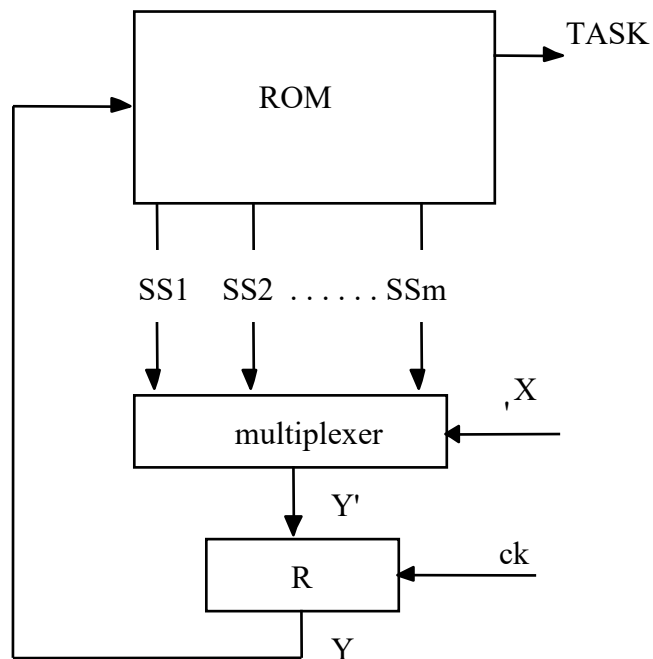
Stato attuale	input	Stato successivo
$Y_1 Y_0$	$X_2 X_1 X_0$	$Y'_1 Y'_0$
0 0	0 0 0	
0 0	0 0 1	
• • .....		
1 1	1 1 1	

Stato attuale	Uscita
$Y_1 Y_0$	$Z_1 Z_0$
0 0	
0 1	
1 0	
1 1	

## Rappresentazione tabellare, una per ROM

Stato attuale $Y_1 Y_0$	$X_2$ $X_1$ $X_0$ 000	$X_2$ $X_1$ $X_0$ 001	$X_2$ $X_1$ $X_0$ 010	$X_2$ $X_1$ $X_0$ 011	$X_2$ $X_1$ $X_0$ 100	$X_2$ $X_1$ $X_0$ 101	$X_2$ $X_1$ $X_0$ 110	$X_2$ $X_1$ $X_0$ 111	Uscita $Z_1 Z_0$
0 0									
0 1									
1 0									
1 1									

## Rappresentazione matriciale, un'unica ROM



**Dimensione della parola della ROM ancora eccessivo**

**Infatti**

**numero delle righe =  $2^m$**

**dove ogni riga ha una dimensione pari a  $m = 2^n + k$**

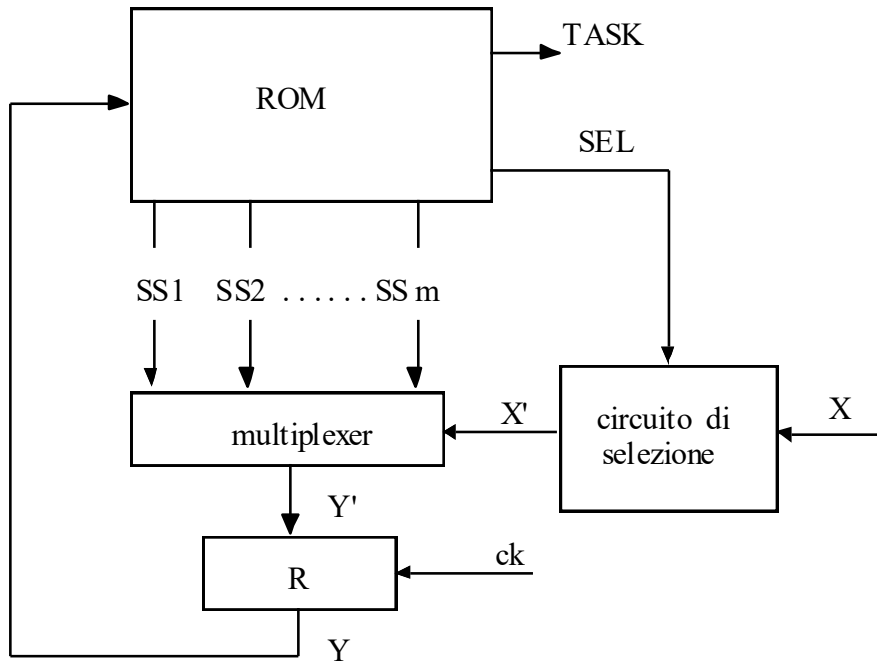
**p.e., se**

- $n = 20$
- $m = 4$
- $k = 10$

**allora**

**dimensione della parola di ROM =  $2^{20} + 10$  (1Mega)**

## Soluzione che riduce le dimensioni della ROM



**Struttura del SCO nel caso di modello di Moore con circuito di selezione**

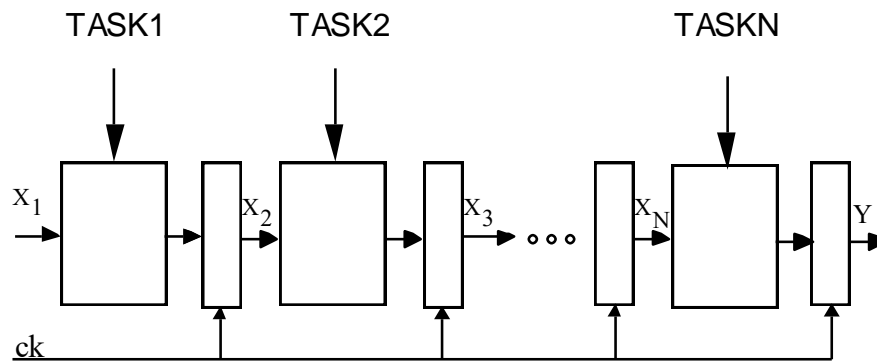
SS <sub>1</sub>	SS <sub>2</sub>	.....	SS <sub>m</sub>	SEL	TASK
-----------------	-----------------	-------	-----------------	-----	------

**Parola di ROM nel caso di modello di Moore con circuito di selezione**

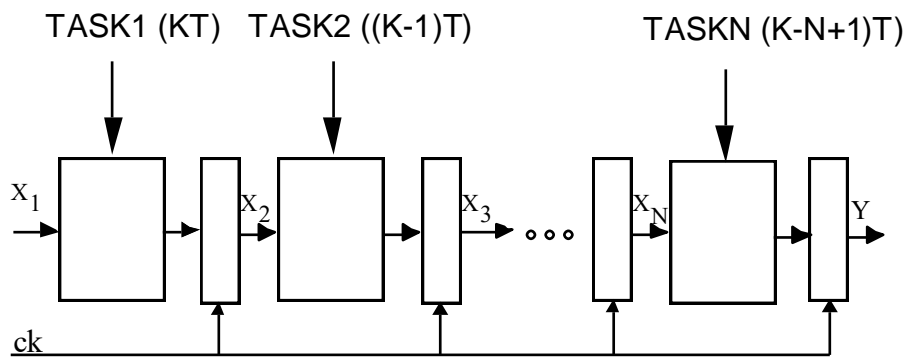
Dove **m** questa volta è pari alla cardinalità delle variabili selezionate **X'**, normalmente 1 o 2.



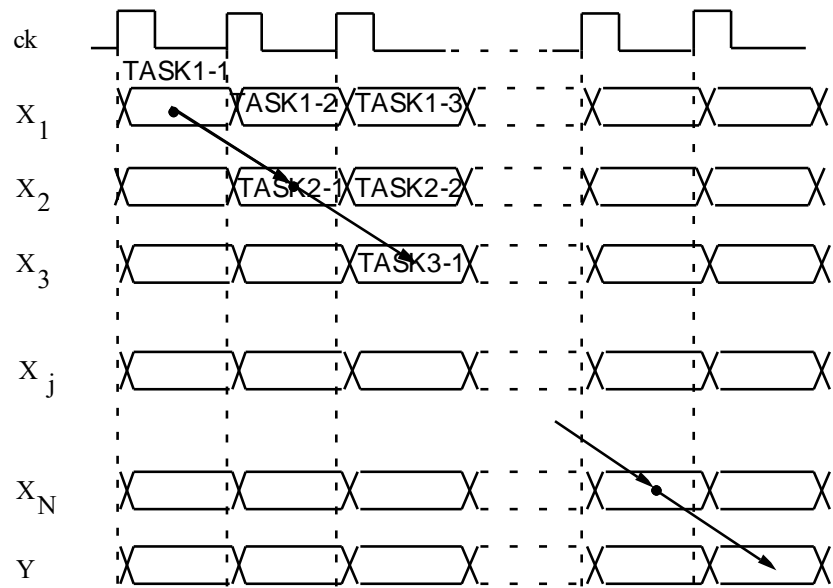
## Controllo per strutture Pipeline.



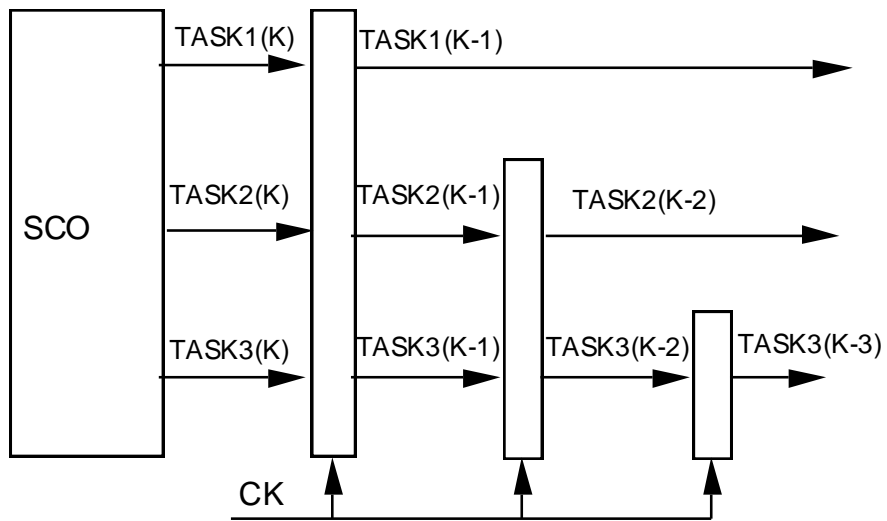
Sistemi di calcolo a task fisso nel tempo



Sistemi di calcolo a task variabile nel tempo

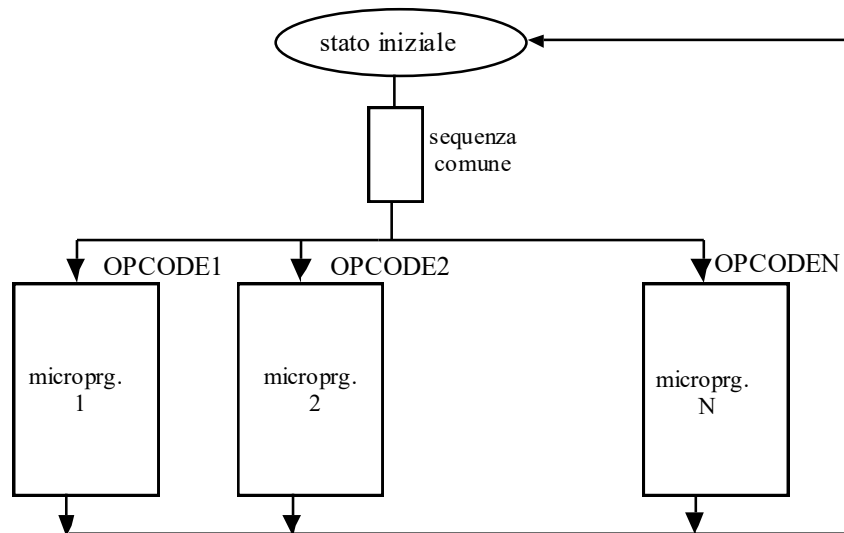


**Temporizzazione di una catena pipeline**

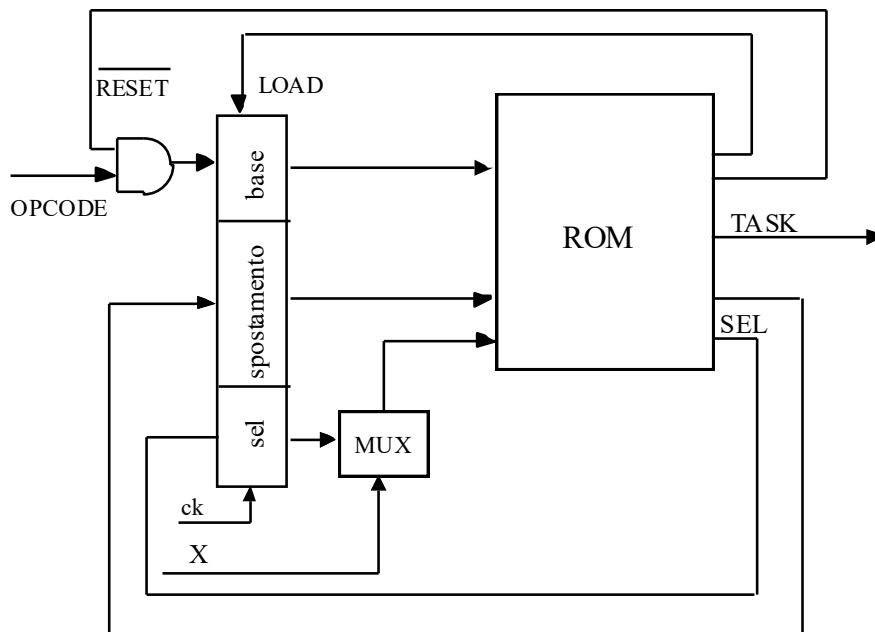


**Architettura SCO per generare TASK sfasati nel tempo**

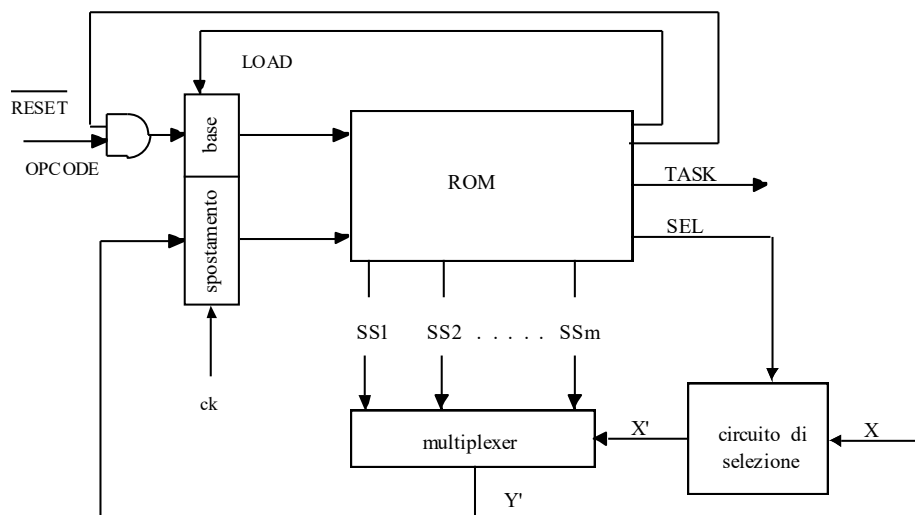
## Sistemi con molti microprogrammi



**Diagramma di flusso con più microprogrammi**

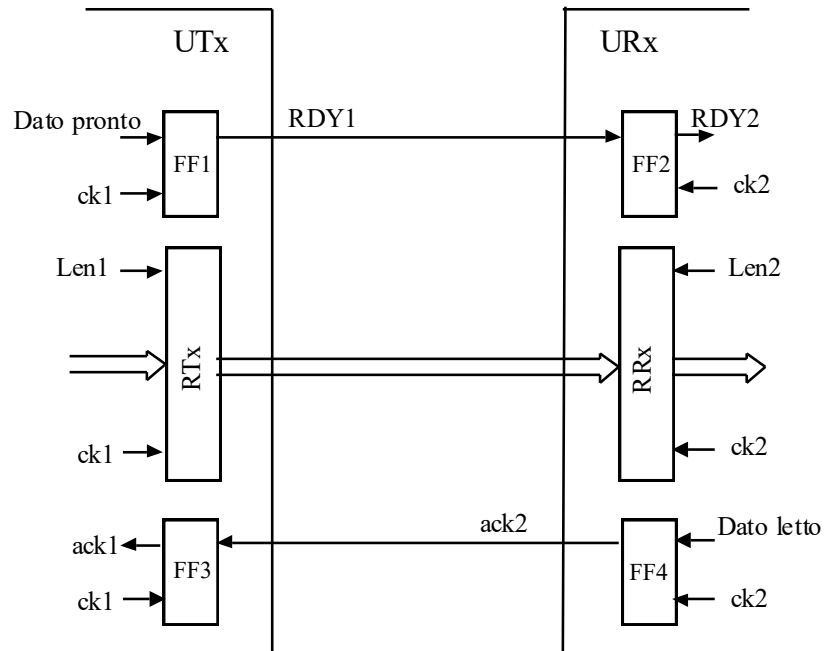


**Schema di SCO con molti microprogrammi (modello di Mealy)**



**Schema di SCO con molti microprogrammi (modello di Moore)**

# Comunicazione tra due sistemi digitali



## Sincronizzazione di due unità in comunicazione

$$\mathbf{UT_x}$$

- 1: dato  $\rightarrow$  RTx, 1  $\rightarrow$  FF1;
- 2: if ack1=0, then vai a 2;
- 3: 0  $\rightarrow$  FF1;
- 4: if ack1=1, then vai a 4;
- 5: i+1  $\rightarrow$  i;
- 6: if i  $\neq$  N, vai a 1

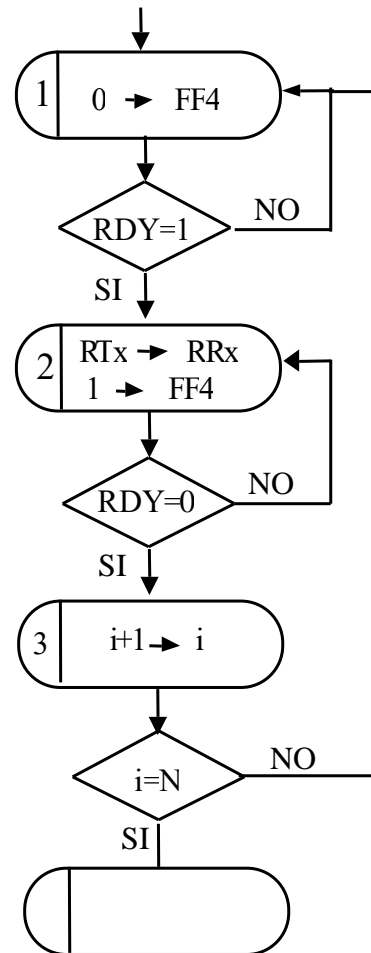
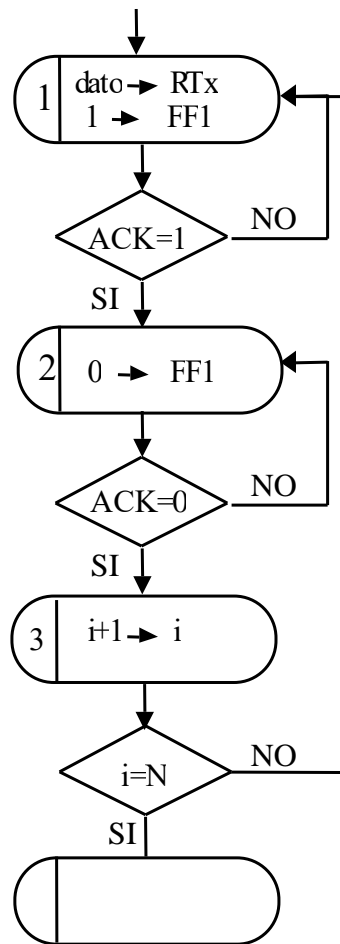
URx

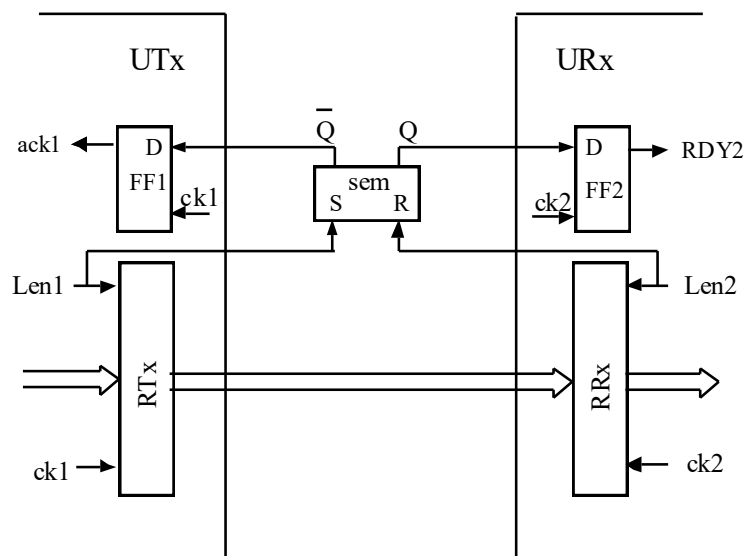
```

1: 0→FF4;
2: if RDY2=0, then vai a 2;
3: RTx→RRx, 1→FF4;
4: if RDY2=1, vai a 4 ;
5: i+1→i;
6: if i≠N, vai a 1

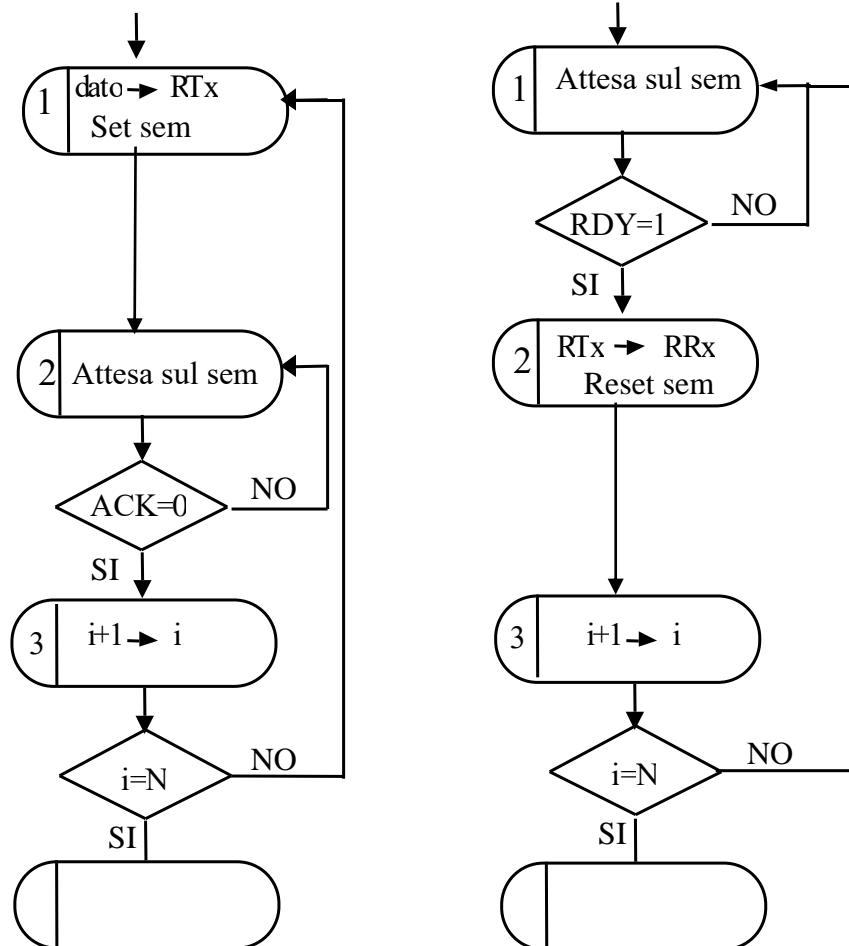
```

**Sequenze di microistruzioni eseguite da UTx e URx durante il protocollo di comunicazione**





**Sincronizzazione tra due unità in comunicazione mediante flip-flop di semaforo**



**Sequenze di microistruzioni eseguite da UTx e URx durante il  
protocollo di comunicazione**

**UTx**

- 1: dato  $\rightarrow$  RTx, set **sem**;
- 2: if ack1=0 vai a 2;
- 3:  $i+1 \rightarrow i$ ;
- 4: if  $i \neq N$ , vai a 1;

**URx**

- 1: if RDY2=0, vai a 1;
- 2: RTx  $\rightarrow$  RRx, reset **sem**;
- 3:  $i+1 \rightarrow i$ ;
- 4: if  $i \neq N$ , vai a 1;