

# Architetture dei Sistemi Elettronici

## 11. Reti sequenziali

Roberto Roncella



# Reti sequenziali sincrone

## Un approccio strutturato

(7.1-7.3, 7.5-7.6)

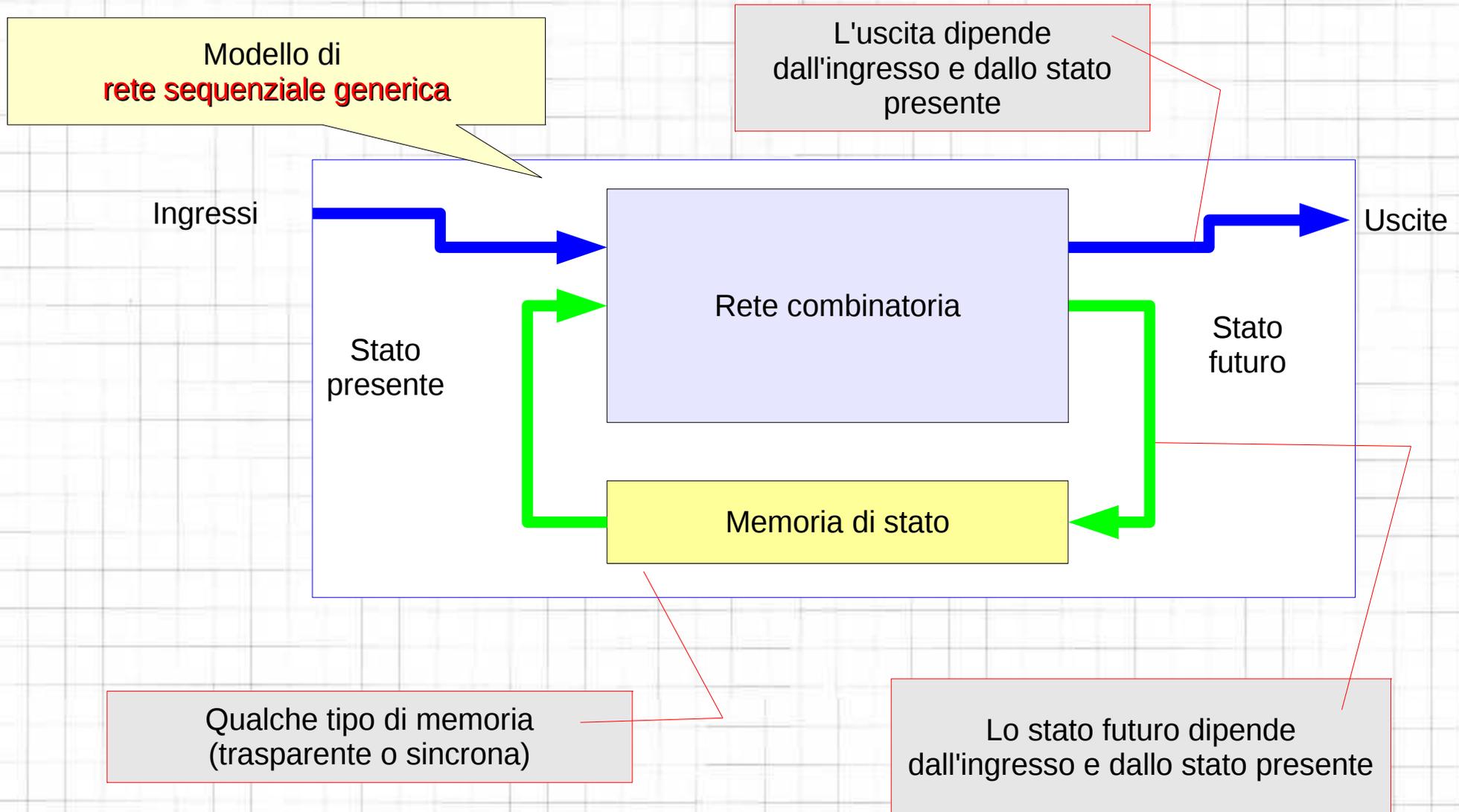
Modelli di reti sincrone

Analisi di reti sincrone

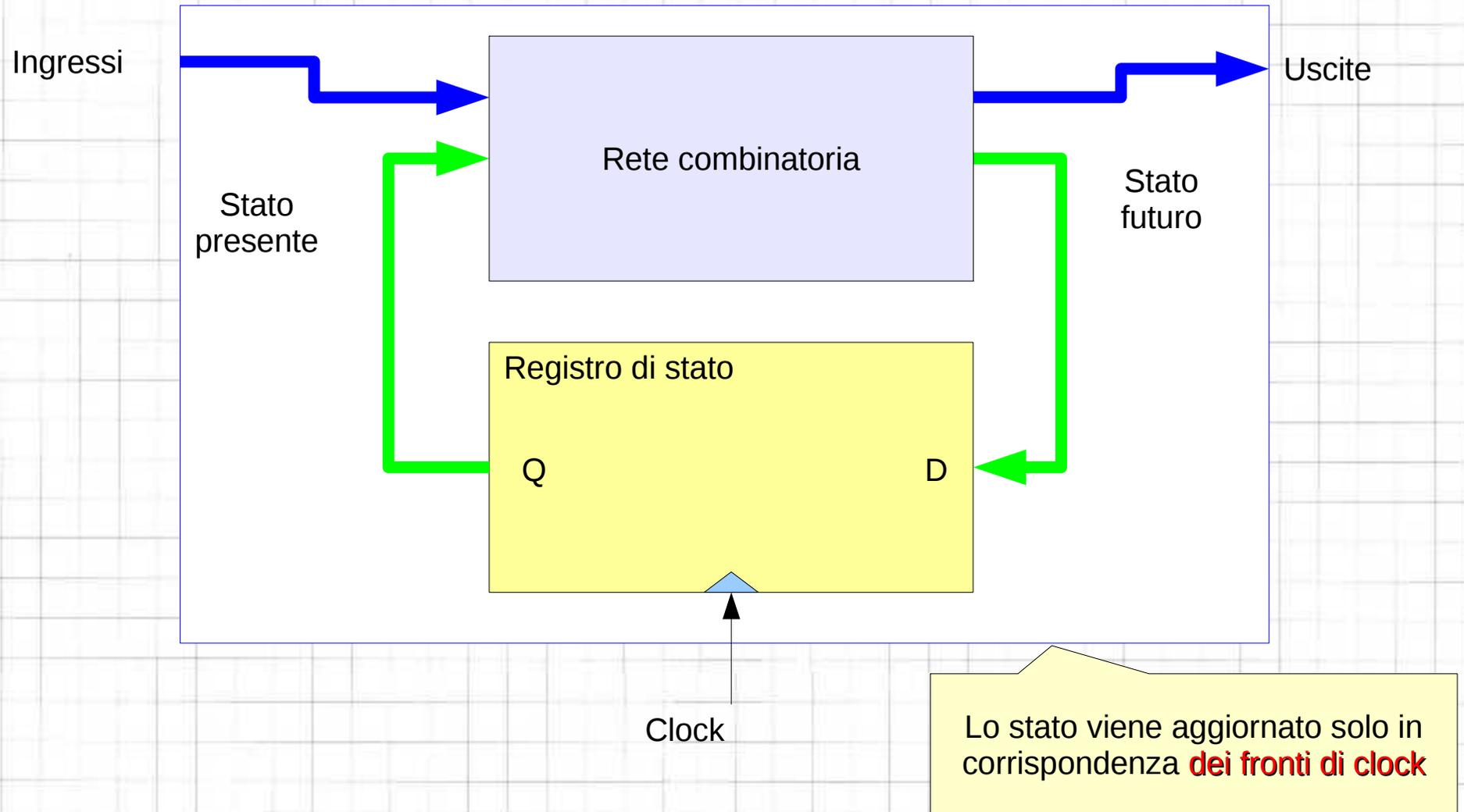
Descrizioni e sintesi di reti sequenziali sincrone

Sintesi con flip-flop  $D$ ,  $DE$ ,  $T$  o  $JK$

# Reti sequenziali

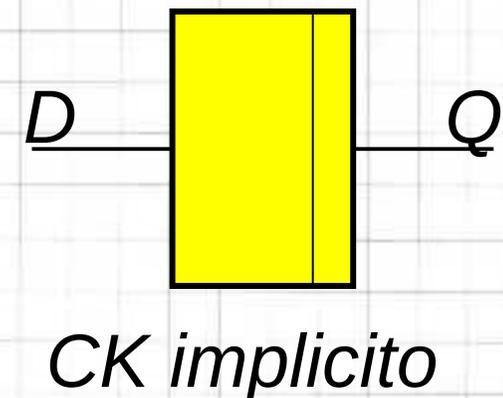
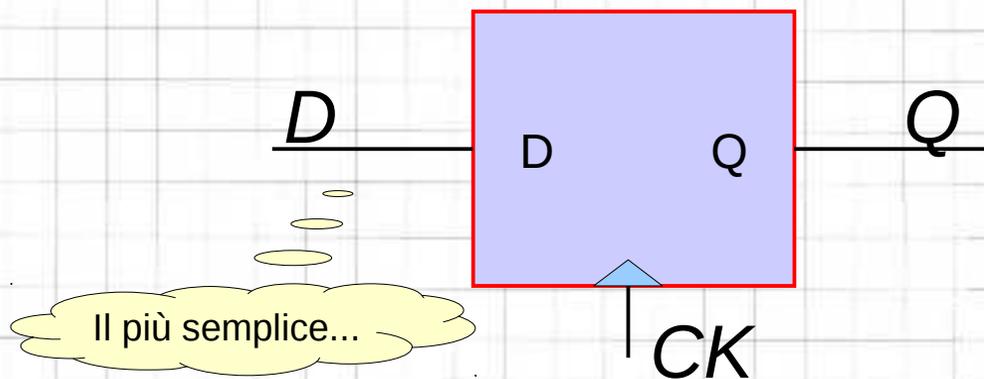


# Reti sequenziali sincrone



# Registro di stato

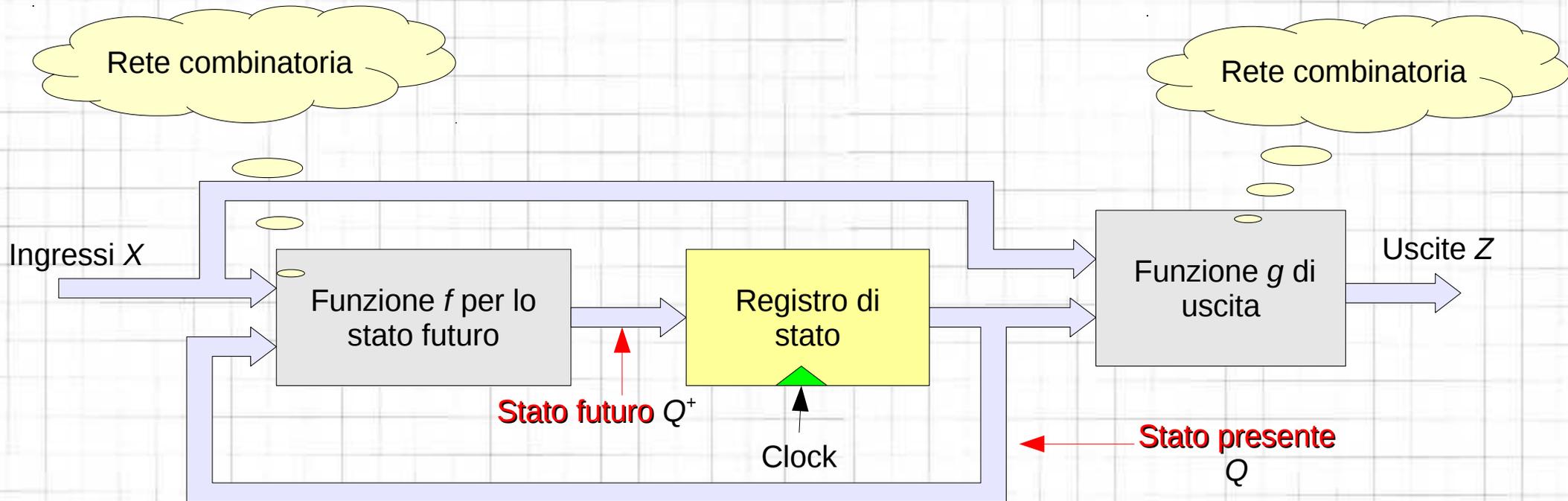
- Elemento di memoria (variabile di stato)
  - Possiamo usare qualsiasi flip-flop non trasparente, in grado di mantenere nel tempo la sua uscita ( $Q$ )
    - Determinata dal valore degli ingressi ( $D$ ,  $DE$ ,  $JK$ ,  $T$ ) assunti in corrispondenza dei fronti di clock ( $CK$ )
    - Gli ingressi sono stabili a cavallo del fronte ( $t_{su}$  e  $t_h$ )
  - L'uscita è aggiornata solo in un secondo momento
    - Dopo un tempo  $t_{co}$



# Modelli di macchina sincrona

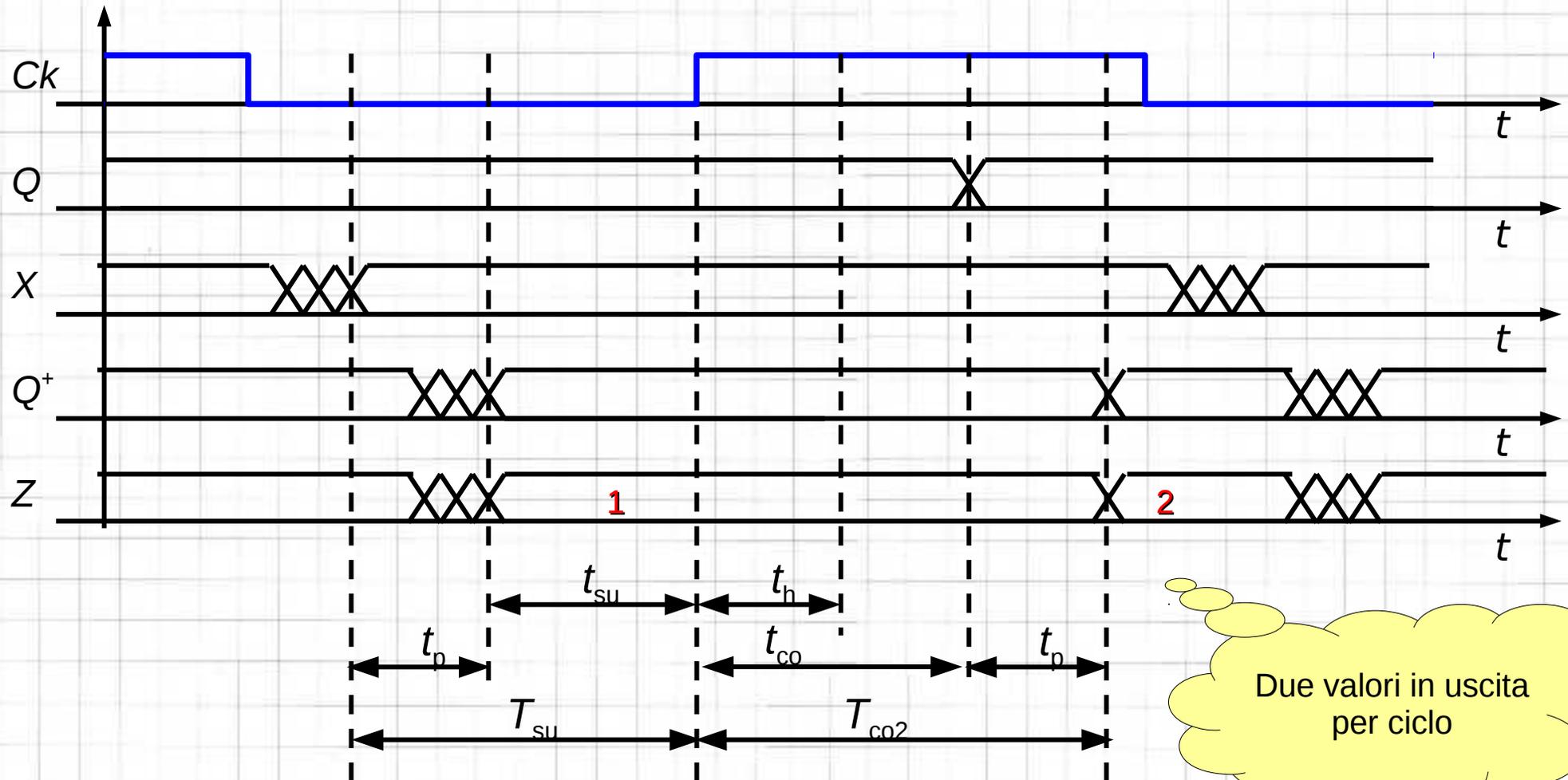
- Abbiamo  $n$  ingressi esterni e uno stato interno rappresentato da  $k$  variabili di stato
  - Esistono  $2^k$  possibili stati diversi
  - Le  $m$  uscite della macchina dipendono, in generale, dal valore dello stato e da quello degli ingressi
- I principali modelli si differenziano per la gestione delle uscite
  - Nel modello di **Mealy** l'uscita dipende dallo stato presente (oppure futuro...) e dagli ingressi
  - Nel modello di **Moore** l'uscita dipende solo dallo stato presente

# Macchina di Mealy



Le uscite dipendono dalle variabili di stato e dagli ingressi e, a differenza dello stato, sono aggiornate in modo **asincrono**

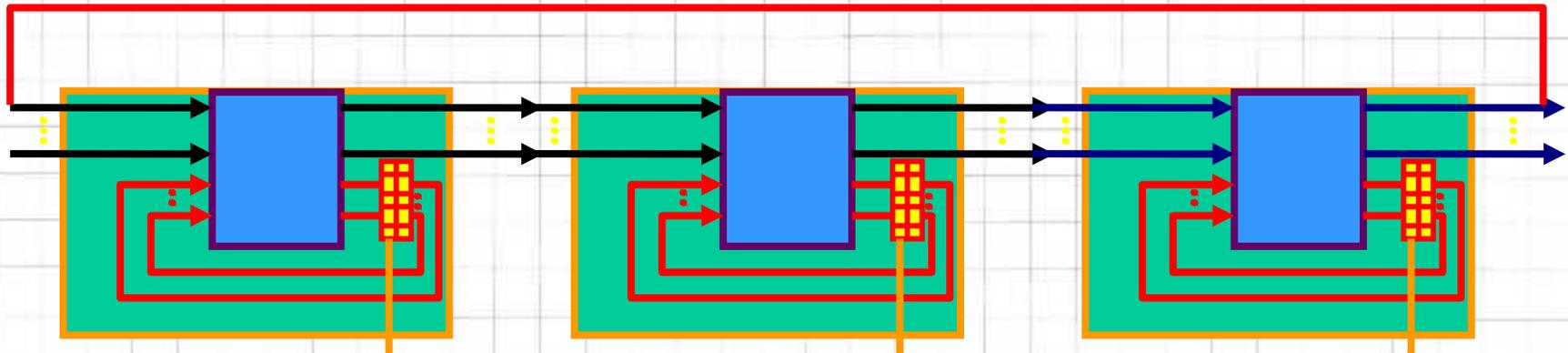
# Temporizzazione (Mealy)



Due valori in uscita per ciclo

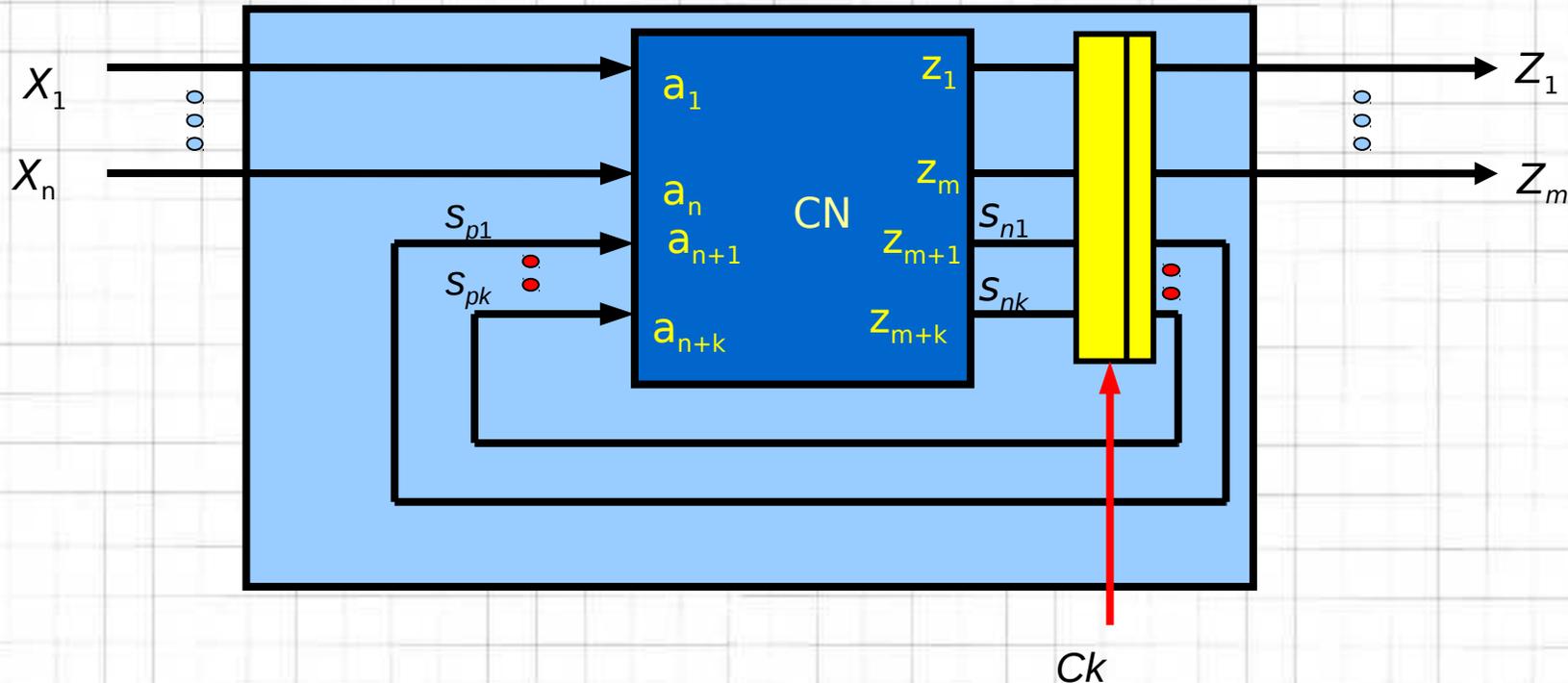
# Osservazioni

- Le uscite sono **asincrone**
  - Variano anche quando il clock è stabile
- È rischioso usare più reti fra loro connesse
  - Si possono ottenere relazioni iterative
    - Possono dare origine a nodi di memoria od oscillazioni



# Macchina di Mealy sincronizzata

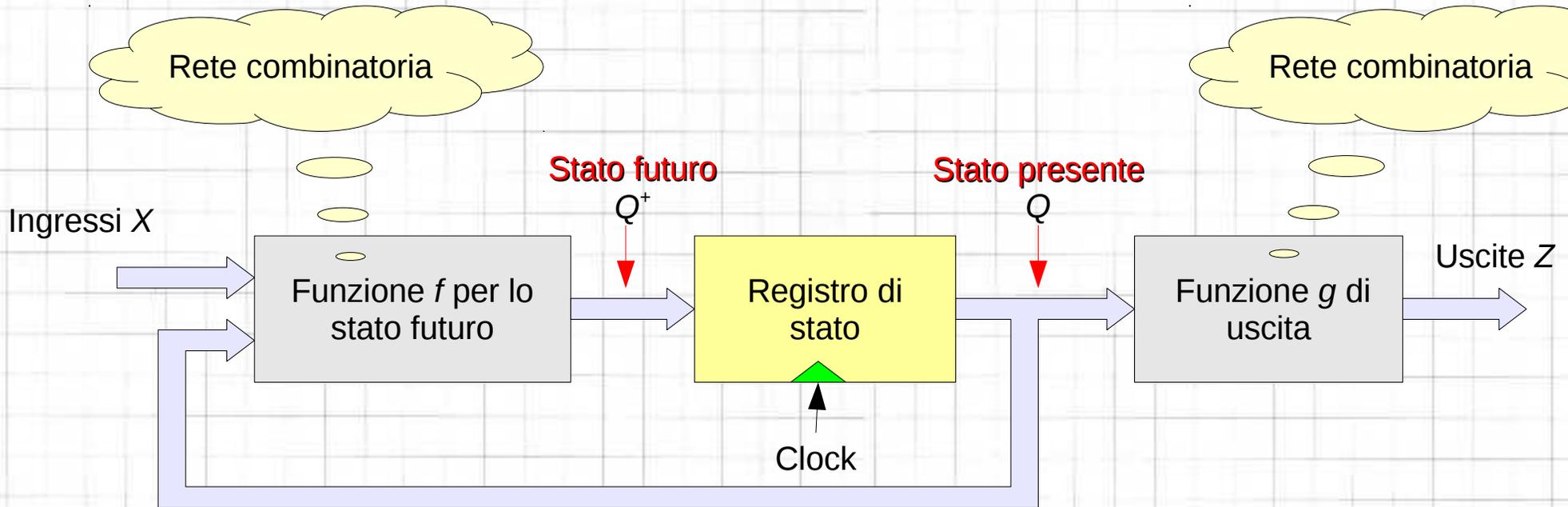
- Si aggiunge un **registro di uscita**
  - Le uscite sono funzioni delle variabili di stato e degli ingressi, ma risultano sincronizzate
  - Viene congelato il valore **1** della temporizzazione



# Osservazioni su Mealy ritardata

- Le uscite sono sincrone
  - Si presentano solamente dopo il fronte di clock
  - Si perde il valore **2** generato in uscita in conseguenza al cambiamento dello stato
- È possibile usare più reti fra loro connesse senza il pericolo di creare anelli di reazione che possano dare luogo a reti sequenziali asincrone

# Macchina di Moore



Le uscite dipendono solo dalle variabili di stato e, come lo stato, sono aggiornate in modo sincrono

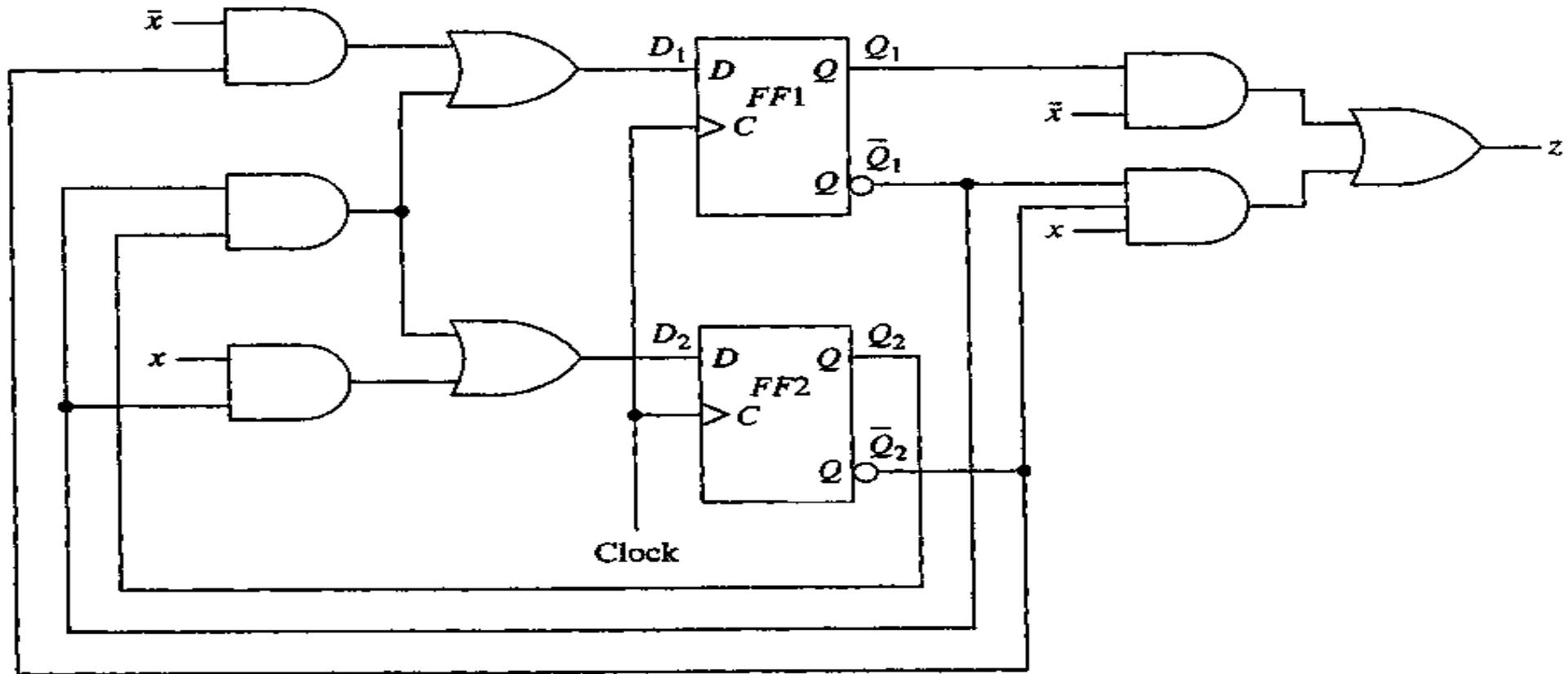
# Equivalenza dei modelli

- Equivalenza logica tra Mealy ritardata e Moore
  - La macchina di Mealy ritardata è una macchina di Moore in cui la rete di uscita è un corto circuito
  - Una macchina di Moore si può realizzare con una macchina di Mealy ritardata in cui la rete per le uscite è la composizione delle due reti di Moore
- Differente temporizzazione delle due reti
  - L'analisi dei tempi dei due sistemi evidenzia una diversa gestione dei tempi
    - Cambiano i tempi di rispetto esterni  $T_{su}$  e  $T_h$
    - Cambia il tempo di valutazione complessivo  $T_{co}$

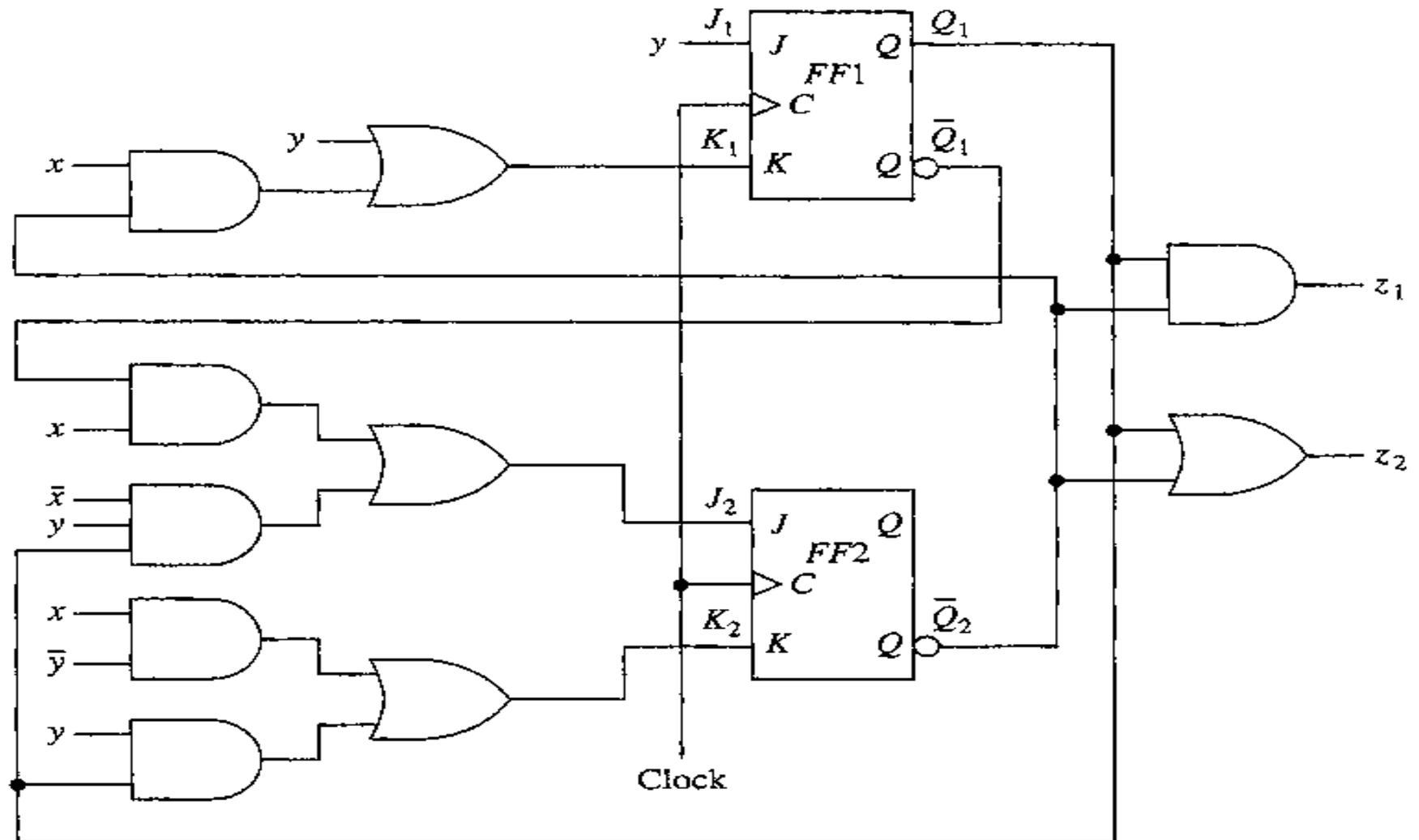
# Analisi di rete sincrona

- Dallo **schema** a una **descrizione formale**
  - Individuare l'architettura e i suoi elementi
  - Determinare il comportamento della rete
    - Insieme delle funzioni combinatorie per il calcolo dello stato futuro e dell'uscita
    - Estrarre le informazioni essenziali che descrivono il comportamento ed esprimerle in modo sintetico e rigoroso
- Dall'**analisi** alla **sintesi**
  - Cioè dalla descrizione formale allo schema
  - Ripercorrendo la procedura in senso inverso si ottiene una modalità strutturata di progetto
    - Risolvendo diversi gradi di libertà: architettura, codifica degli stati, tipologia di reti combinatorie, ecc.

# Una rete di Mealy



# Una rete di Moore



# Tabelle di uscita ed eccitazione

- Individuare e definire con una variabile le uscite di tutti i flip-flop
  - Questo individua le variabili che codificano lo stato presente
- Determinare la relazione booleana [stato presente, (e ingressi se Mealy)] → [uscita]
- Determinare la relazione booleana [stato presente, ingressi] → [ingressi dei flip-flop]
  - Si definisce tabella di eccitazione
  - Per ogni possibile stato si esaminano le uscite al variare di ogni ingresso

# Equazioni di transizione

- Determinazione dello stato futuro
  - Combinando la tabella di eccitazione con quella del funzionamento del flip-flop presente nel registro di stato si ottiene la relazione tra lo stato attuale, l'ingresso e lo stato futuro
  - 
  - Si può usare una forma algebrica, come espressione, definita equazione di transizione
  - Oppure una forma tabellare stato presente-stato futuro, ovviamente al variare di tutte le combinazioni degli ingressi

# Equazioni di transizione (es.)

$$Q_1^+ = y\bar{Q}_1 + \overline{(y + x\bar{Q}_2)}Q_1$$

$$= y\bar{Q}_1 + \bar{y}(\bar{x} + Q_2)Q_1$$

$$= y\bar{Q}_1 + \bar{x}\bar{y}Q_1 + \bar{y}Q_1Q_2$$

$$Q_2^+ = (x\bar{Q}_1 + \bar{x}yQ_1)\bar{Q}_2 + \overline{(x\bar{y} + yQ_1)}Q_2$$

$$= (x\bar{Q}_1 + \bar{x}yQ_1)\bar{Q}_2 + (\bar{x} + y)(\bar{y} + \bar{Q}_1)Q_2$$

$$= x\bar{Q}_1\bar{Q}_2 + \bar{x}yQ_1\bar{Q}_2 + \bar{x}\bar{y}Q_2 + \bar{x}\bar{Q}_1Q_2 + y\bar{Q}_1Q_2$$

# Tabella di transizione (Mealy)

**Table 7.1** Transition table for Example 7.1

| Present state<br>( $Q_1Q_2$ ) | Next state<br>( $Q_1^+Q_2^+$ ) |    | Output<br>( $z$ ) |   |
|-------------------------------|--------------------------------|----|-------------------|---|
|                               | Input ( $x$ )                  |    | Input ( $x$ )     |   |
|                               | 0                              | 1  | 0                 | 1 |
| 00                            | 10                             | 01 | 0                 | 1 |
| 01                            | 11                             | 11 | 0                 | 0 |
| 10                            | 10                             | 00 | 1                 | 0 |
| 11                            | 00                             | 00 | 1                 | 0 |

# Tabella di transizione (Moore)

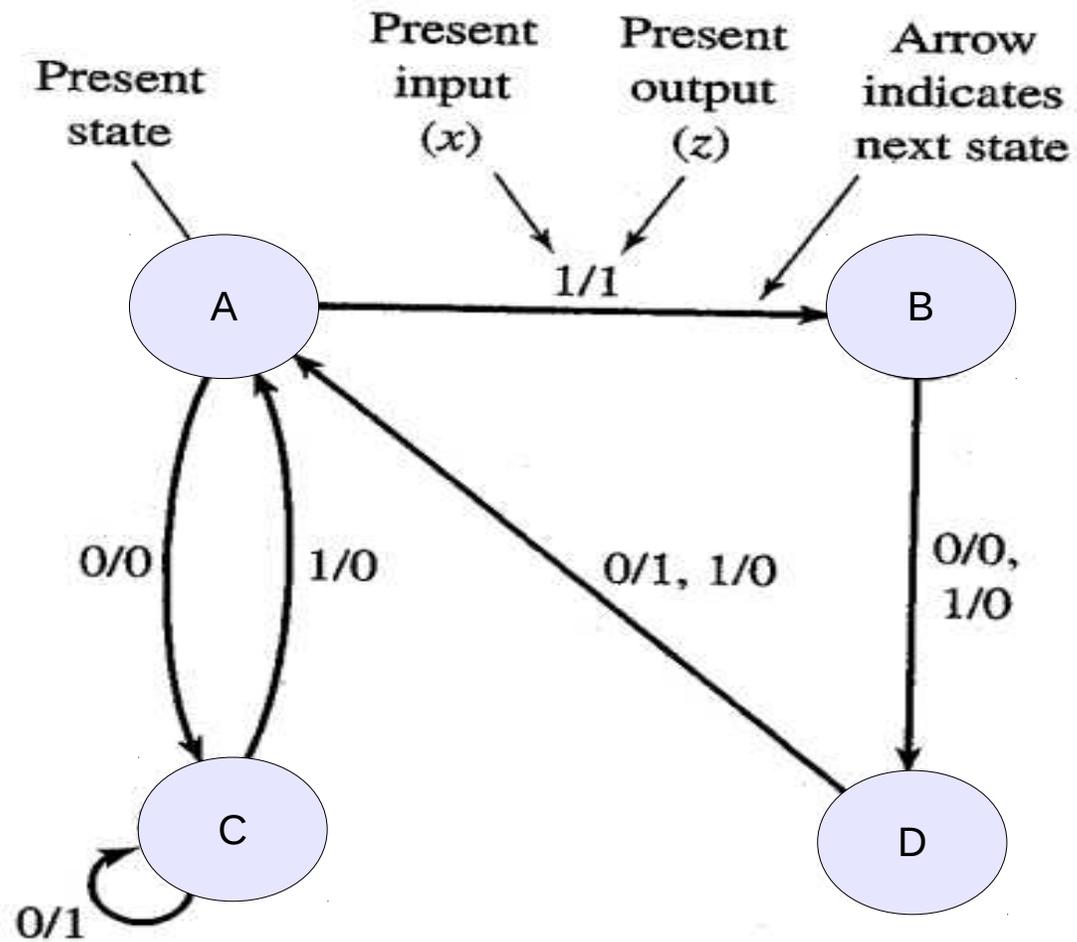
**Table 7.2** Transition table for Example 7.2

| Present state<br>( $Q_1Q_0$ ) | Next state<br>( $Q_1^+Q_0^+$ ) |    |    |    | Output<br>( $Z_1Z_0$ ) |
|-------------------------------|--------------------------------|----|----|----|------------------------|
|                               | Inputs ( $x_1x_0$ )            |    |    |    |                        |
|                               | 00                             | 01 | 10 | 11 |                        |
| 00                            | 00                             | 10 | 01 | 11 | 01                     |
| 01                            | 01                             | 11 | 00 | 11 | 00                     |
| 10                            | 10                             | 01 | 00 | 00 | 11                     |
| 11                            | 11                             | 00 | 10 | 00 | 01                     |

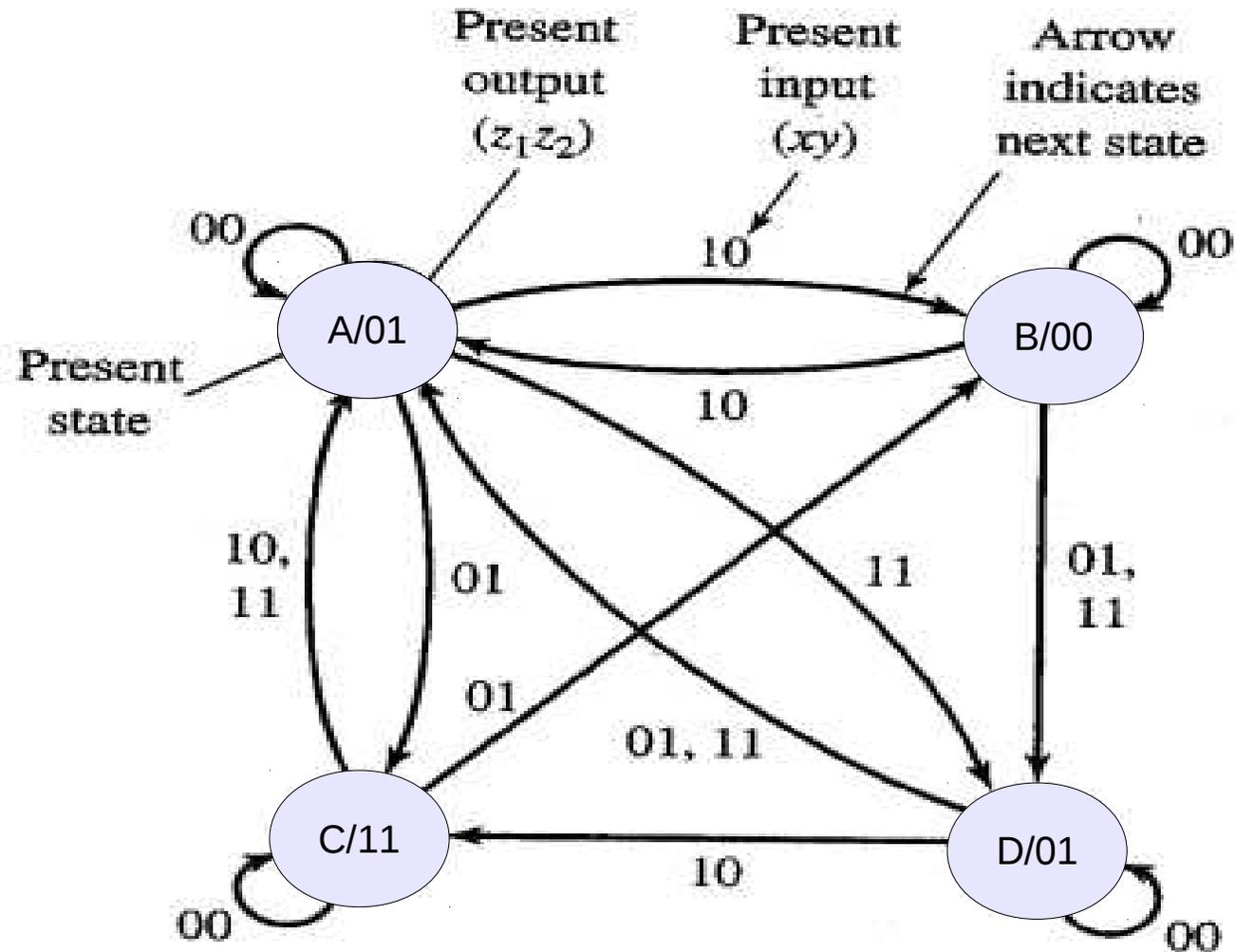
# Tabella degli stati

- Il codice assunto da uno stato non ha alcun rilievo sul funzionamento della rete
  - Si può attribuire a ciascuno degli stati della macchina un nome simbolico e ricostruire la tabella delle transizioni evitando di fare riferimento ai valori specifici binari usati dalla rete
  - Quindi si può fare la tabella [simbolo dello stato presente, ingresso] → [simbolo dello stato futuro]

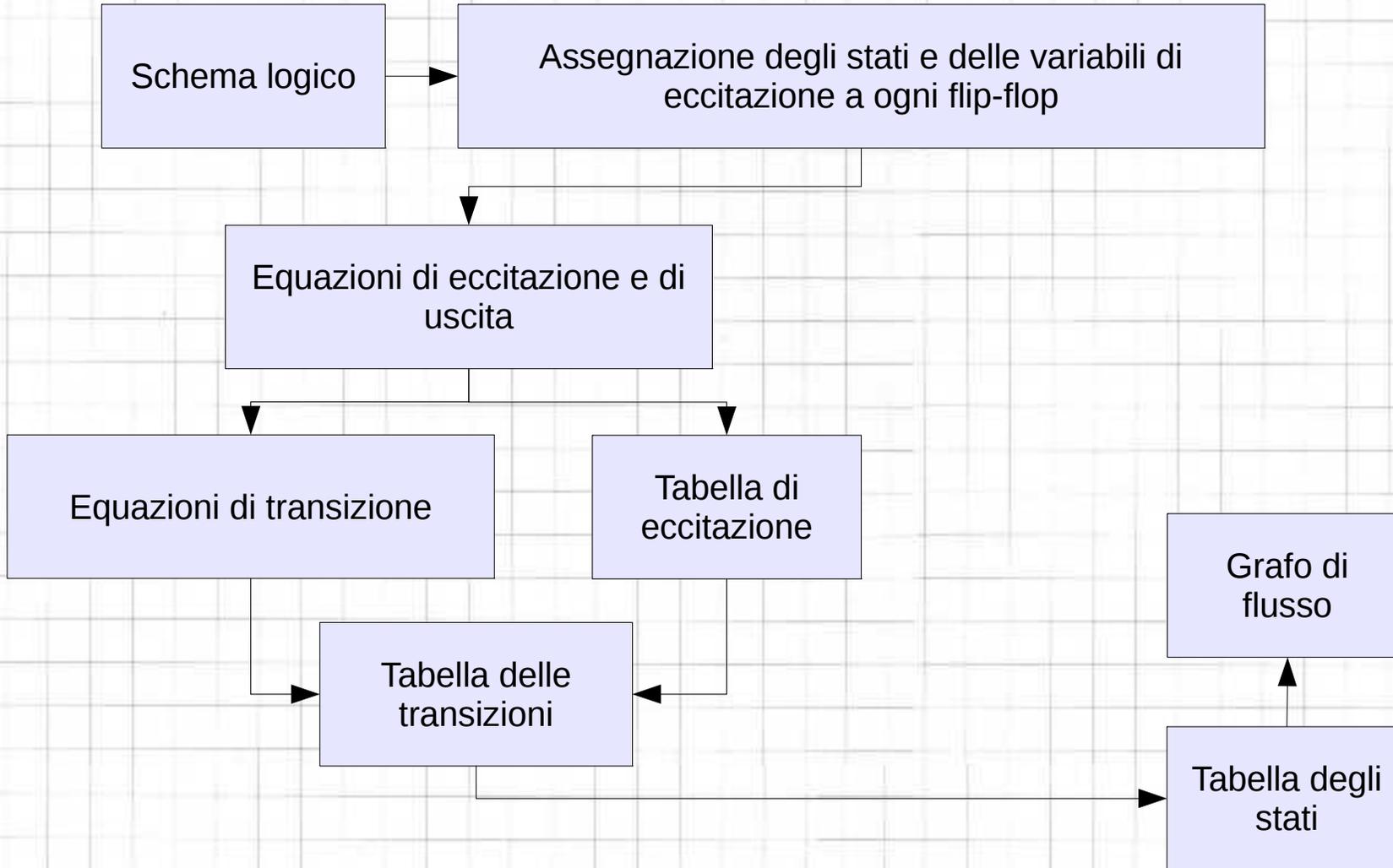
# Diagramma degli stati (Mealy)



# Diagramma degli stati (Moore)



# Procedura di analisi



# Sintesi di reti sincrone

- Definire una descrizione formale
  - Grafo di flusso o tabella delle transizioni
- Codificare gli stati
- Determinare la funzione stato presente -stato futuro
- Eccitare correttamente i flip-flop per ottenere la transizione prevista
- Sintetizzare le reti per lo stato futuro e per l'uscita

# Descrizione di reti sincrone

- Descrizioni “intuitive”
  - Linguaggio naturale
  - Tabella operativa
  - Forme d’onda
- Descrizioni formali
  - Diagramma di flusso
  - Grafo di flusso
  - Tabella di flusso
    - Permette la realizzazione della sintesi
  - HDL

# Il linguaggio naturale

- Esempio: una rete *JK* sincrona
  - Rete logica con due ingressi, due stati e una uscita
    - Gli ingressi *J* e *K*
    - Lo stato Spento (*S0*) e lo stato Acceso (*S1*)
    - L'uscita *Q*
  - Descrizione dell'effetto degli ingressi
    - L'attivazione del solo *J* porta la rete nello stato *S1*
    - L'attivazione del solo *K* porta la rete nello stato *S0*
    - Se nessun ingresso è attivo, la rete resta dove è
    - Se entrambi gli ingressi sono attivi, si cambia stato
  - Definizione dell'uscita (secondo Moore)
    - Nello stato *S0* *Q* è basso, in *S1* invece è alto

# Altre descrizioni “intuitive”

- Descrizione tabellare
  - La più diffusa nei data sheet
  - Riepiloga le affermazioni della descrizione in linguaggio naturale per tutte le combinazioni di ingressi
  - Nella tabella appaiono le variabili di stato contemporaneamente come ingressi ( $Q_n$ )
  - ... e come uscite ( $Q_{n+1}$ , o stato futuro)
- Mediante forme d'onda
  - Fornisce indicazione dell'andamento nel tempo
  - Utile per capire le specifiche sui tempi, ma difficile per apprezzare il comportamento logico

# Descrizione tabellare

- Simile alla tabella di verità delle reti combinatorie
  - Ma nella colonna delle uscite si presentano i simboli  $Q'$  e  $\bar{Q}$  invece della stessa uscita
  - Viene fatto riferimento al valore assunto nello stato precedente

| CK | J | K | Q'        |
|----|---|---|-----------|
| 0  | - | - | Q         |
| 1  | - | - | Q         |
| ↓  | - | - | Q         |
| ↑  | 0 | 0 | Q         |
| ↑  | 0 | 1 | 0         |
| ↑  | 1 | 0 | 1         |
| ↑  | 1 | 1 | $\bar{Q}$ |

# Descrizione formali

- Tecniche grafiche
  - Mediante diagramma di flusso
    - Intuitivo, di facile interpretazione
  - Mediante grafo
    - Molto compatto, evidenzia la memorizzazione
- Tecniche analitiche o procedurali
  - Tabella di flusso
    - Molto compatta, può essere utilizzata per la sintesi
  - Mediante linguaggio di programmazione (HDL)
    - Consente la verifica e sintesi automatica

# Diagramma degli stati

- Detto anche **grafo di flusso**
- Per rappresentare una rete si usa una forma grafica costituita da nodi uniti da archi orientati
  - I nodi corrispondono agli stati
    - Internamente è indicato il valore dello stato (e delle variabili d'uscita, in Moore)
    - Da ogni nodo partano tanti archi quante le combinazioni di ingresso
  - Gli archi orientati corrispondono alle transizioni degli ingressi
    - Sopra gli archi è riportata la configurazione degli ingressi corrispondente (e dell'uscita, in Mealy ritardata)
    - Le configurazioni degli ingressi che danno luogo a stati non specificati comportano archi interrotti

# Sintesi di una rete sincrona (1)

- Scegliere e indicare l'architettura
- Disegnare il grafo di flusso
- Dal grafo alla tabella di stato
- Codificare gli stati
  - Valutare il numero di stati e decidere quanti flip-flop sono necessari
  - Assegnare a ogni stato un valore binario
    - L'assegnazione è arbitraria
    - Si possono seguire alcune linee guida per facilitare la sintesi successiva
- Tabella delle transizioni

# Sintesi di una rete sincrona (2)

- Dalla tabella delle transizioni si può produrre direttamente una sintesi ottima SP o PS per le due reti  $f$  e  $g$
- Occorrono alcuni passaggi
  - Scelta del tipo di flip-flop
  - Tabella di eccitazione (per FF-D coincide con la tabella di transizione)
    - La tabella viene riassemblelata in forma di mappa
  - Sintesi della rete per lo stato futuro  $f$
  - Sintesi della rete per l'uscita  $g$ 
    - La descrizione si costruisce mappando la funzione combinatoria che dà l'uscita in funzione dello stato (e dell'ingresso in Mealy)

# Sintesi con altri FF

- In alcuni casi può essere conveniente usare per le variabili di stato altri tipi di FF
  - FF-DE, FF-JK, FF-T
  - La tabella delle eccitazioni non coincide con la tabella delle transizioni, come nel caso del FF-D
  - Occorre modificare la rete combinatoria principale per il calcolo dello stato futuro
  - Occorre calcolare il valore degli ingressi dei flip-flop a partire da quello di  $D (Q_{n+1})$
- Occorre fare uso della tabella di applicazione dei diversi flip-flop

# Sintesi con FF-DE

- Devo sintetizzare la rete considerando i due ingressi  $D$  ed  $E$ 
  - Se si pone  $E = 1$ , la sintesi rimane la stessa
  - Si semplificano i casi in cui  $Q_n$  e  $Q_{n+1}$  sono uguali
    - Esistono più combinazioni di ingressi che permettono di garantire il mantenimento dello stato

| $Q_n$ | $Q_{n+1}$ | $D$ | $E$ |
|-------|-----------|-----|-----|
| 0     | 0         | -/0 | 0/1 |
| 0     | 1         | 1   | 1   |
| 1     | 0         | 0   | 1   |
| 1     | 1         | -/1 | 0/1 |

# Sintesi con FF-JK

- Devo sintetizzare la rete considerando i due ingressi J ed K
  - Si ha la massima versatilità nella sintesi delle nuove variabili: sono presenti “non specificati” nel 50% delle caselle

| $Q_n$ | $Q_{n+1}$ | J | K |
|-------|-----------|---|---|
| 0     | 0         | 0 | - |
| 0     | 1         | 1 | - |
| 1     | 0         | - | 1 |
| 1     | 1         | - | 0 |

# Sintesi con FF-T

- L'uso della variabile T altera la distribuzione dei mintermini della funzione per lo stato futuro
  - Conveniente quando nella sintesi con i FF-D sono presenti operazioni XOR, come nei contatori

| $Q_n$ | $Q_{n+1}$ | T |
|-------|-----------|---|
| 0     | 0         | 0 |
| 0     | 1         | 1 |
| 1     | 0         | 1 |
| 1     | 1         | 0 |