

Circuiti Sequenziali



Corso di Architettura degli Elaboratori 1

Fulvio Valenza

A.A. 2018-2019

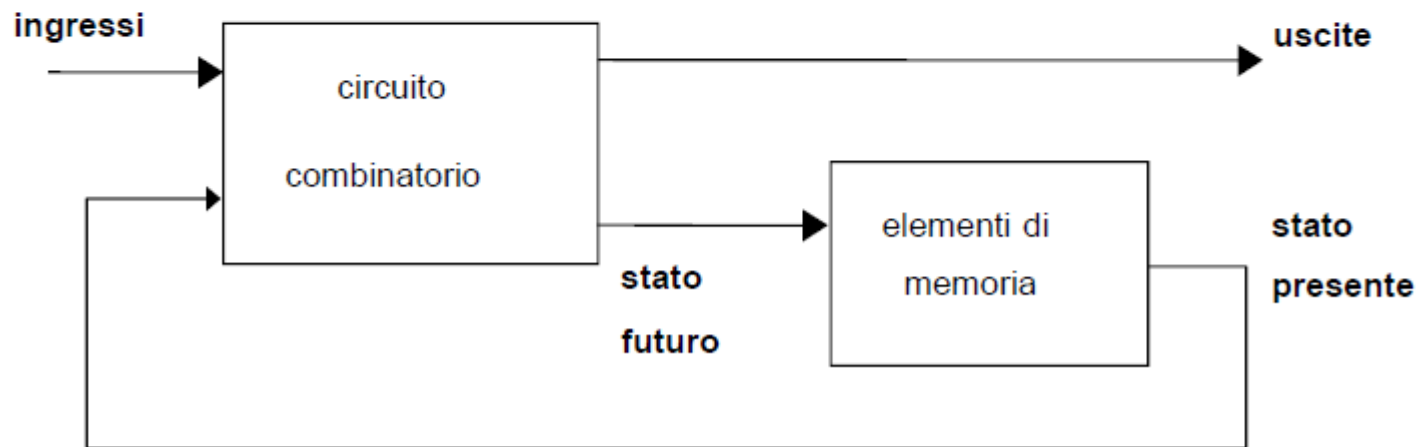
Polo di Vercelli

Circuito Sequenziale

- **Il circuito digitale è di tipo sequenziale se le uscite dipendono non solo dai valori correnti degli ingressi, ma anche da (alcuni di) quelli passati:**
 - *una stessa configurazione di ingresso applicata in due istanti di tempo successivi può produrre due valori di uscita differenti.*
- **Il circuito digitale sequenziale (o rete sequenziale) è pertanto dotato, in ogni istante di tempo, di uno stato che, insieme ai valori degli ingressi, ne determina il comportamento futuro:**
 - *lo stato del circuito sequenziale rappresenta una forma di memoria e contiene una sorta di descrizione della storia passata del circuito stesso.*
- L'elemento funzionale elementare per la realizzazione del circuito sequenziale è il bistabile (elemento di memoria), che è in grado di memorizzare un bit di informazione.

Circuito Sequenziale

- Il circuito sequenziale è costituito dagli elementi seguenti:
 - *bistabile*, che ha la funzione di memorizzare un bit,
 - *rete combinatoria*, che ha funzione di elaborare le informazioni.
- *In ogni istante il circuito sequenziale ha uno stato: il valore dei bit memorizzati nei bistabili facenti parte del circuito.*



Reti sequenziali

- ❑ Variazioni delle configurazioni di ingresso modificano, oltre che le uscite, anche lo stato interno. Lo stato interno attuale si dice **stato presente**.
- ❑ In seguito alla variazione degli ingressi il sistema può calcolare in ogni istante quello che sarà lo **stato futuro**.

Elementi di Memoria

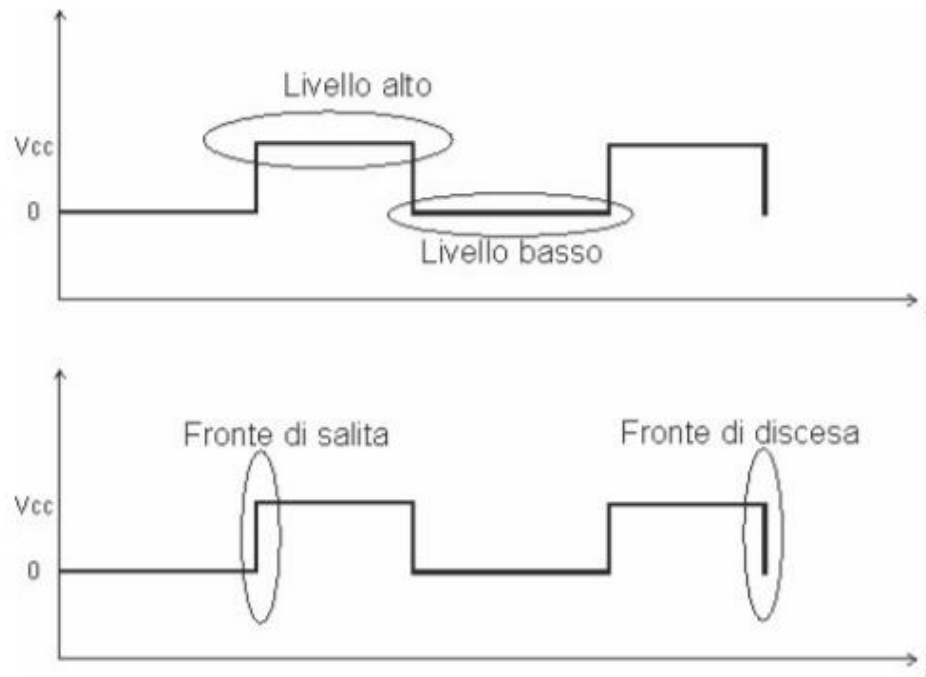
- L'elemento di memoria fondamentale, o bistabile, è caratterizzato da due stati (0 e 1) stabili.
 - Mantiene lo stato finché uno o più segnali di ingresso forzano il cambiamento di stato.
- I bistabili sono classificati come segue:
 - *secondo il numero di ingressi previsti per comandare il bistabile, e*
 - *secondo come tali ingressi determinano il cambiamento di stato.*

Clock

- Il segnale di *clock* è generato da un circuito (realizzato con un opportuno cristallo) che emette un segnale impulsivo periodico con una precisa durata (*pulse width*) e con un preciso intervallo tra due impulsi consecutivi.
- Il clock è un segnale free-running ossia che continua indefinitamente (almeno finché il sistema è alimentato), di tipo periodico, con un periodo detto **tempo di clock** T_{ck} (clock cycle time); il suo reciproco è la **frequenza di clock** f_{ck} o f . Una rete che ha la frequenza di 100MHz ha un ciclo di clock di 10ns.

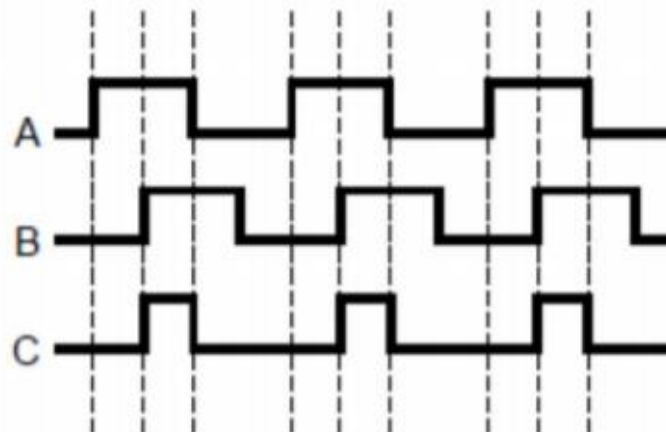
Clock

- Si definiscono livelli (alto e basso) e fronti o edge (di salita e di discesa) le quattro parti della forma d'onda riportata in figura



Clock

- Nei calcolatori il segnale di clock sequenzializza tutti gli eventi. Spesso nel calcolatore si usa oltre al clock primario dei clock secondari che sono sincroni ma che sono di dimensione minore (la metà) per eseguire più azioni nello stesso clock o maggiori (il doppio, il quadruplo) se alcune reti non sono sufficientemente veloci. Per questo si parla del clock della **CPU**, del **clock di sistema**, o di **clock multipli** (di frequenza).



Clock

- Nelle reti logiche ogni evento elementare si verifica in un ciclo di clock. Se l'evento si verifica mentre il clock è attivo (di solito alto) si dice che la logica lavora “a **livello**”, se ogni evento, ogni transizione di stato e di uscite si verifica al cambiamento del clock si dice che l'evento è “**edge-triggered**” o “a **fronte**”. Di solito si usa il fronte di salita, ma in alcuni casi si usano entrambi
- Le reti logiche che studieremo sono di tipo sincrone e normalmente di tipo *edge-triggered*.

Classificazione dei Bistabili

- bistabile **asincrono**: è privo di segnale di sincronizzazione e **cambia stato rispondendo direttamente a eventi** (fronti di transizione) **sui segnali di ingresso**
- bistabile **sincrono**: è **sensibile a un segnale di sincronizzazione** (o di controllo) e **la transizione di stato può avvenire solo in corrispondenza di eventi** (fronti) **sul segnale di controllo**:
 - si può dire che il comportamento di un circuito sincrono viene osservato in istanti discreti di tempo,
 - il segnale di sincronizzazione tipicamente utilizzato è il clock.

Classificazione dei Bistabili

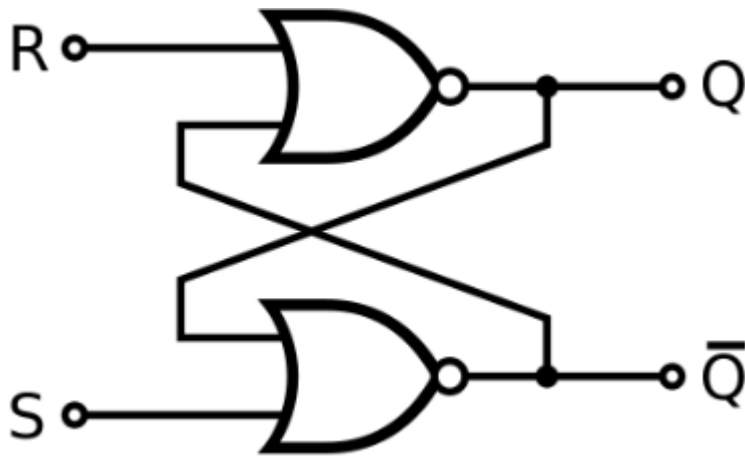
- bistabile trasparente (latch)
- flip-flop (bistabile non trasparente)
 - flip flop master-slave (a livello)
 - flip flop edge-triggered (a fronte)

Latch

- ❑ Elementi di memorizzazione molto semplici
- ❑ Vengono principalmente usati come componenti nella costruzione dei **Filp Flop**
- ❑ Essi si distinguono in base al numero dei loro ingressi e in base al modo in cui cambiano stato
- ❑ Analizzeremo
 - ❑ Latch SR e \overline{SR}
 - ❑ Latch D

Latch SR

- Il Latch di tipo set-reset (SR) è un circuito composto da due port NOR connesse tramite collegamenti incrociati
- È caratterizzato da due ingressi S e R, da una uscita Q e dal suo completo \bar{Q}

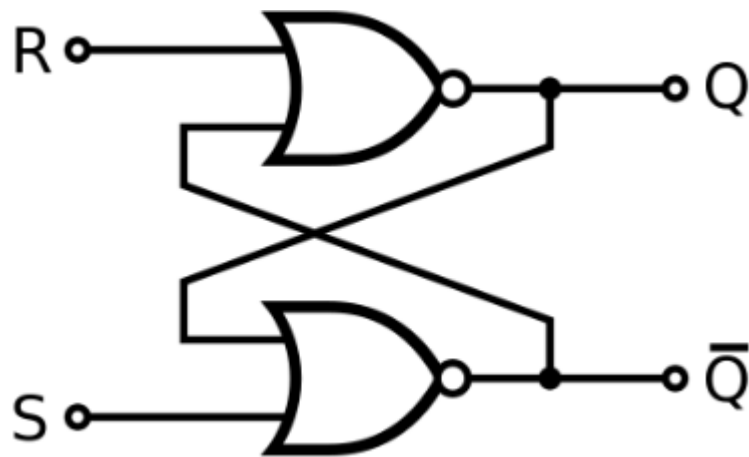


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Latch SR

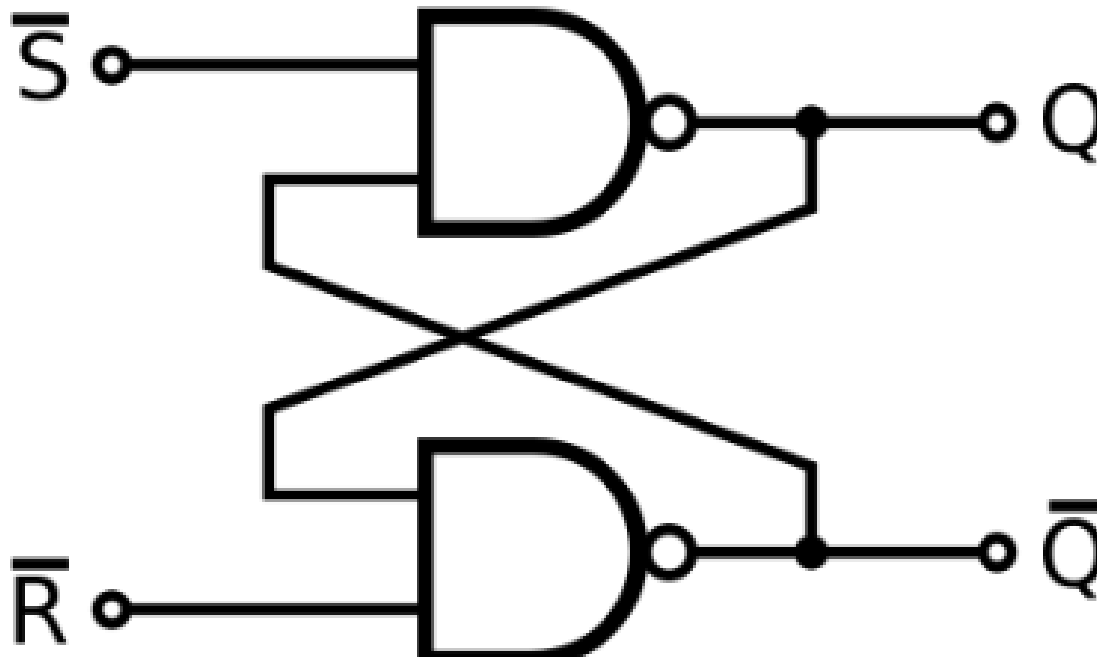
- ▶ In tutti gli stati stabili che ci interessano, le due uscite del latch valgono una l'opposto dell'altra. In particolare avremo:
 - ▶ Lo stato in cui $S = 0$ e $R = 0$. In questo stato, Q può avere qualunque valore.
 - ▶ Lo stato in cui $S = 1$ e $R = 0$. In questo caso, Q deve necessariamente avere valore 1 (e quindi, \overline{Q} deve valere 0).
 - ▶ Lo stato in cui $S = 0$ e $R = 1$. In questo caso, Q deve necessariamente avere valore 0 (e quindi, \overline{Q} deve valere 1).
- ▶ In altre parole, lo **stato interno** del latch è rappresentato dal singolo valore binario Q .
- ▶ Normalmente, entrambi gli ingressi valgono 0 e questa situazione non fa cambiare Q .
- ▶ Se occorre settare Q a 1, il valore di S va modificato a 1 (lasciando R a 0). Se, invece, occorre resettare Q a 0, basta modificare R a 1 (lasciando S a 0).

Latch SR



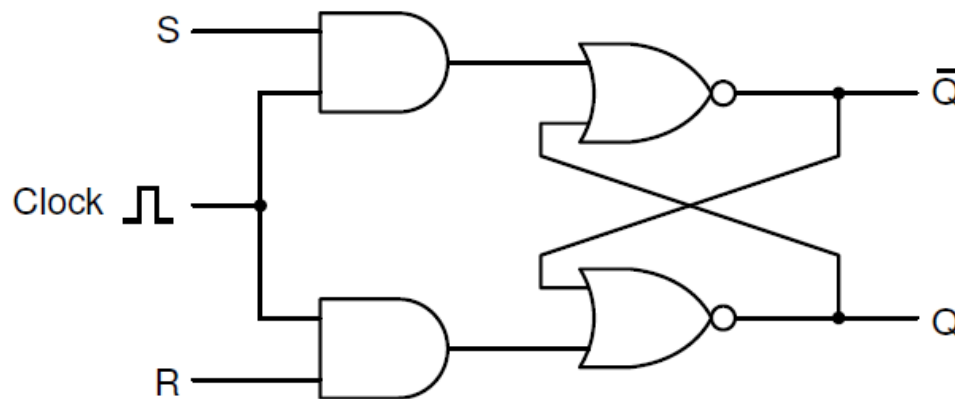
s	r	q	
1	0	1	<i>Set</i>
0	1	0	<i>Reset</i>
0	0	$q'=q$	<u><i>Conservazione</i></u>
1	1	-	<i>Non consentito</i>

Latch \overline{SR}



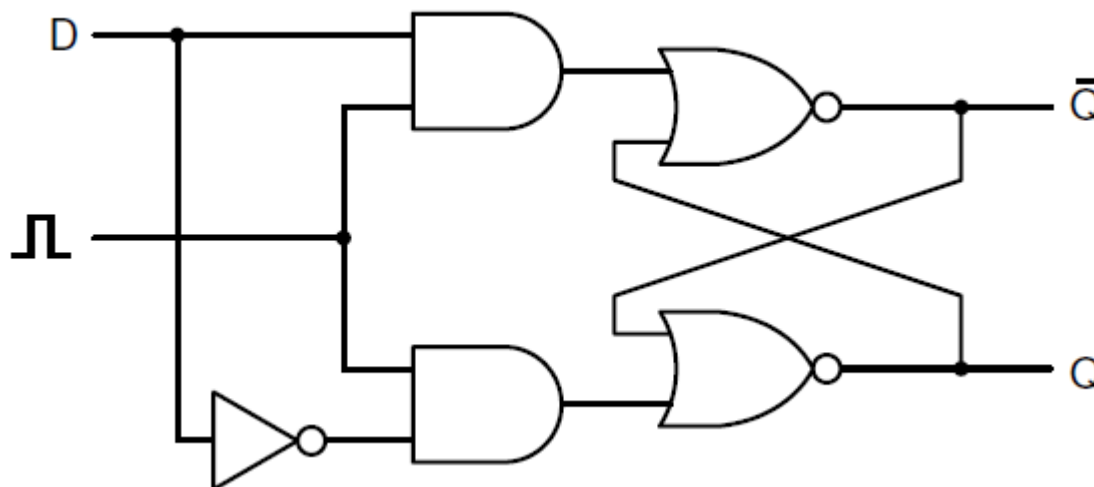
Latch SR con ingresso di controllo

- ▶ Spesso è utile **impedire** che un latch cambi stato, se non in specifici momenti.
- ▶ Vi sarà quindi un ulteriore ingresso C , in genere collegato ad un clock,
 - ▶ Quando $C = 0$, lo stato interno del latch non cambia. In particolare, il latch non è sensibile alle variazioni degli ingressi S e R .
 - ▶ Quando, invece, $C = 1$, lo stato interno del latch SR può cambiare liberamente, seguendo le regole che conosciamo già.

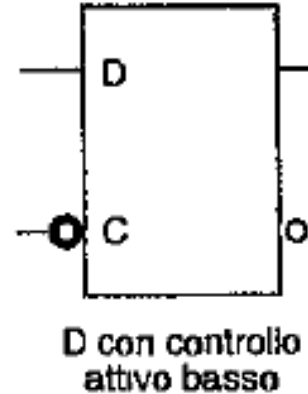
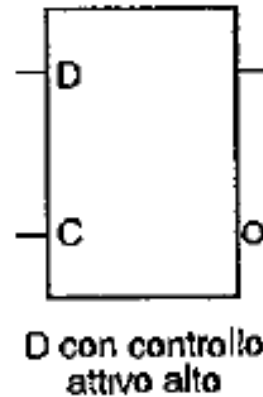
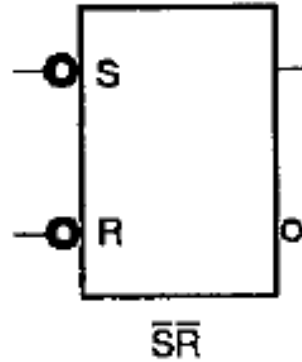
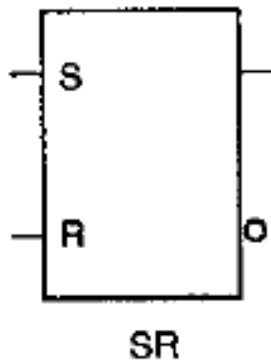


Latch D

- ▶ Abbiamo finora volutamente evitato di parlare degli stati in cui $S = R = 1$.
 - ▶ Solo uno tra essi (ovvero quello per cui $Q = \bar{Q} = 0$) è stabile, ma non è facile passare da esso a qualunque altro stato stabile.
- ▶ Per fare in modo che la situazione $S = R = 1$ non si presenti mai, basta sostituire i due ingressi S e R con un unico ingresso D e fare in modo che il valore in D determini lo stato interno del circuito.
 - ▶ Tale circuito sequenziale è chiamato **latch D**.



Latch simboli grafici

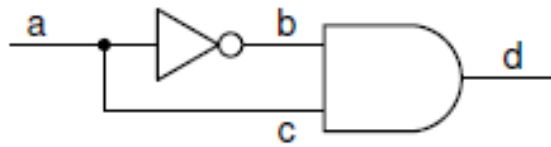


(a) Latch

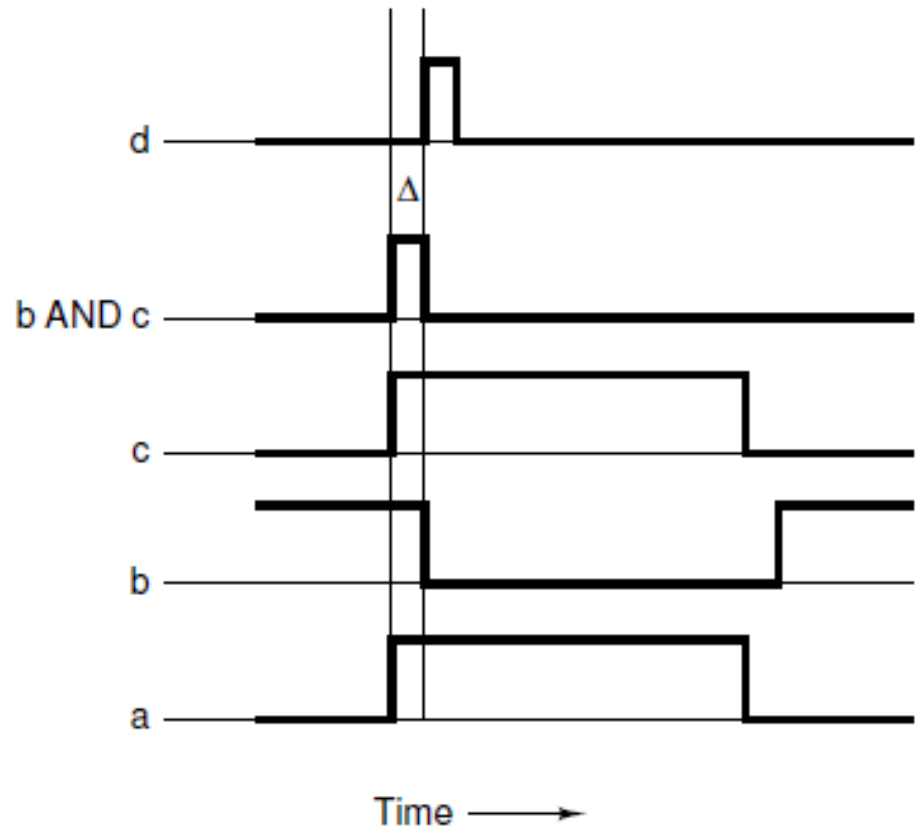
Flip-Flops

- ▶ In molte situazioni, è preferibile avere a disposizione un circuito che **memorizzi** (o **campioni**) il valore binario in input in un particolare istante, piuttosto che in un intervallo.
- ▶ In tali circuiti, chiamati **flip-flop**, lo stato interno cambia quando il clock cambia il suo valore da 0 a 1 oppure da 1 a 0.
 - ▶ Nel primo caso parliamo di **fronte di salita** o **positive edge** e di flip-flop **positive-edge triggered**.
 - ▶ Nel secondo caso parliamo invece di **fronte di discesa** o **negative edge** e di flip-flop **negative-edge triggered**.
- ▶ In questo senso, si dice che i latch (temporizzati) sono circuiti sequenziali a **commutazione di livello**, mentre i flip-flop sono circuiti sequenziali a **commutazione di fronte**.
- ▶ Esistono vari modi per progettare un flip-flop. Un modo semplice è quello di generare un impulso molto breve in corrispondenza del fronte di salita di un clock.
 - ▶ Basterà poi impiegare un latch D.

Generatore d'Impulso

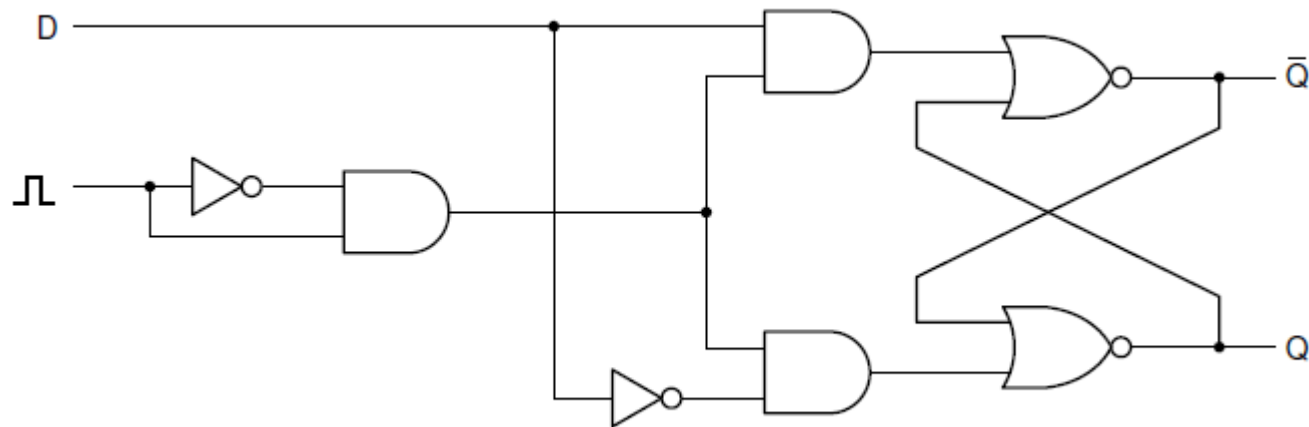


(a)

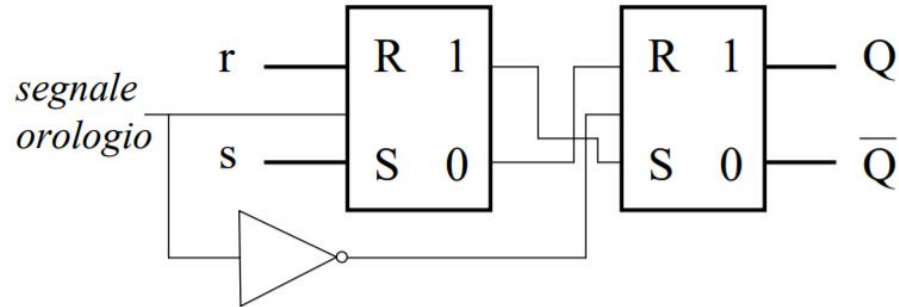


(b)

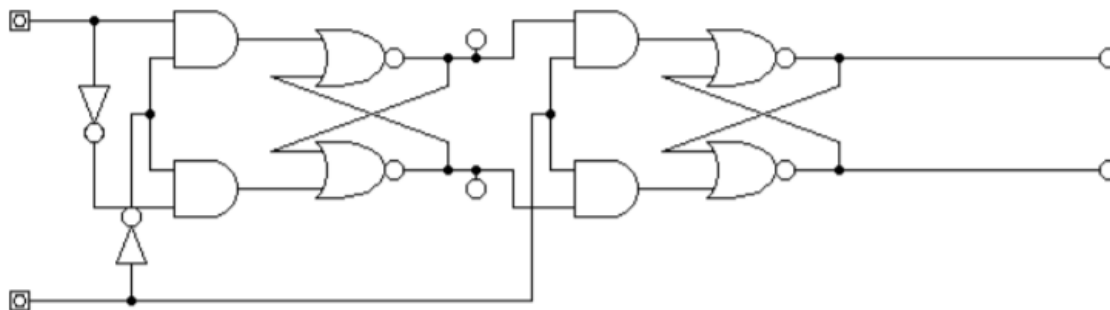
Filp Flop D Positive-edge Triggered



Flip-flop master-slave



- Collegamento di 2 latch in serie: il latch master riceve gli ingressi e il latch slave produce le uscite
- Si sfruttano i 2 livelli del ciclo di clock per effettuare in tempi diversi i cambiamenti di stato dei due latch
- Durante il livello attivo è possibile la lettura, dato che è nel livello passivo che il latch slave cambia stato.



Registri

- Un registro è un elemento di memoria in cui n flip-flop vengono controllati dallo stesso clock, formando sostanzialmente una unità in grado di memorizzare parole composte da n bit.
- Tipicamente sono presenti un segnale di *Input Enable* (o *Chip Select CS*), cioè una linea che consente di attivare la fase di memorizzazione e un segnale di *Output Enable* che rende visibile in uscita la parola memorizzata.

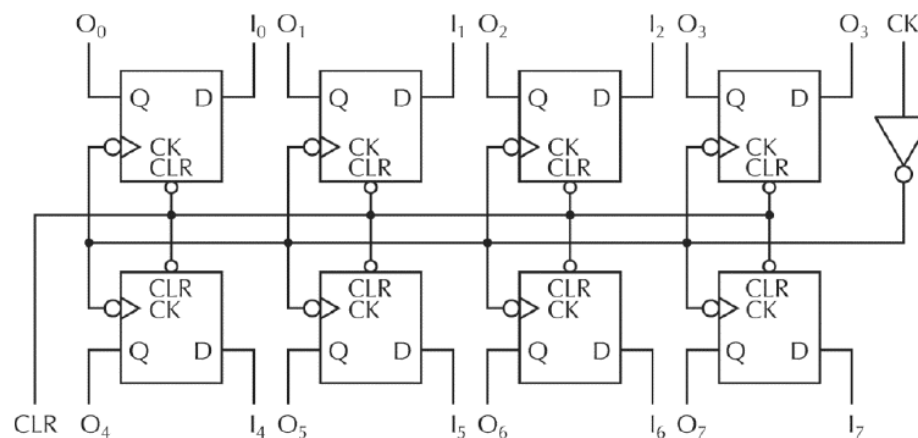


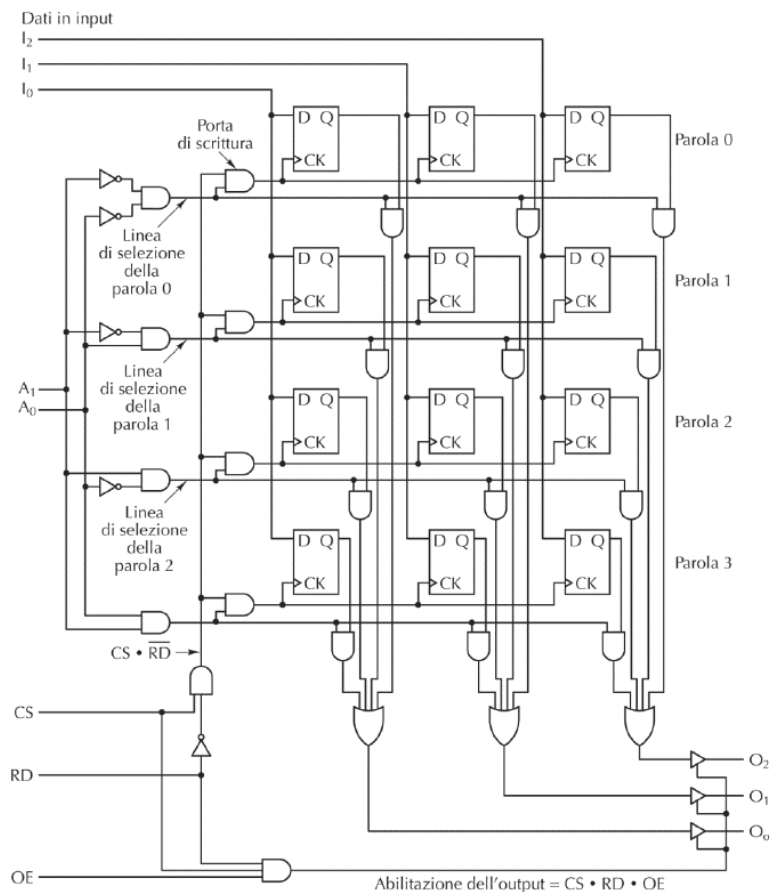
Figura 3.27 T Un registro a 8 bit costruito a partire da flip-flop a 1 bit.

Memorie

- Le memorie sono dispositivi di memorizzazione logicamente assimilabili a banchi di registri, anche se dal punto di vista architetturale se ne discostano profondamente.
- Ogni unità di memorizzazione viene detta **cella** di memoria.
- La presenza di più di un registro introduce la ovvia necessità di *selezionare* a quale registro vogliamo accedere. Dal momento che stiamo lavorando con circuiti binari, la scelta più ovvia è quella di codificare in n bit il numero e utilizzare un decoder per produrre i segnali di abilitazione della cella in questione.
- Il numero così codificato viene detto **indirizzo** della cella e il numero di bit per l'indirizzamento verrà indicato con n_a dove la a indica la parola *address* (indirizzo). Il numero di bit contenuti in ogni cella viene indicato con n_d dove d indica la parola *data* (dati).

Memorie

Una memoria composta da 4 parole lunghe 3 bit. Lettura e scrittura agiscono su una intera parola



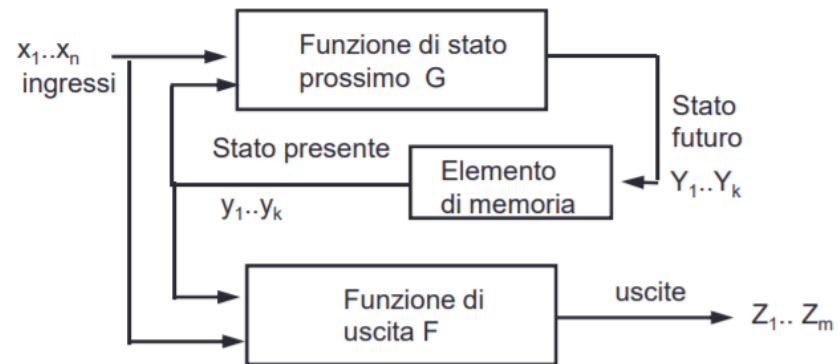
Al **Figura 3.28** Diagramma logico di una memoria 4×3 . Ogni riga è una delle quattro parole a 3 bit. Lettura e scrittura riguardano sempre parole complete.

Automa a stati finiti

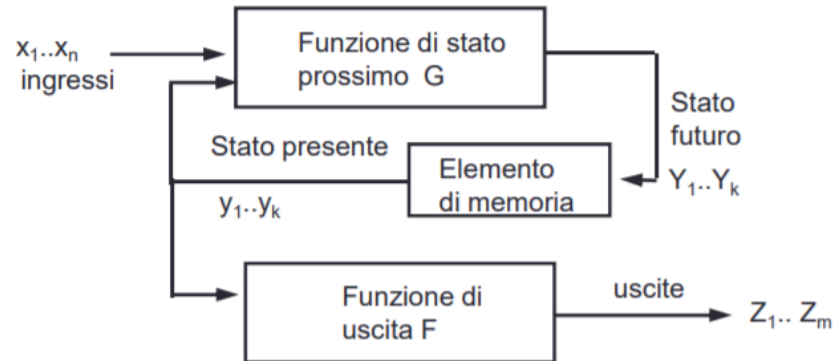
- Una rete sequenziale memorizza le informazioni sulle configurazioni di ingresso che si verificano nel tempo; la memorizzazione avviene in *stati interni*
- Le variabili di stato che definiscono lo stato interno in cui si trova la rete sono memorizzate in elementi di retroazione
- Tra le reti sequenziali, importanza fondamentale hanno le macchine a stati finiti (**FSM**, **Finite state machine**) in cui gli elementi di retroazione sono Flip Flop con un unico segnale di clock
- L'insieme dei FF è detto **registro di stato** e memorizza lo stato futuro presentando a valle lo stato presente

Automa a stati finiti

Modello generale (**automa di Mealy**): il valore delle uscite dipende dallo stato presente e dagli ingressi in quell'istante



Modello equivalente (**automa di Moore**): il valore delle uscite dipende solo dallo stato presente e non dagli ingressi in quell'istante



E' sempre possibile passare da un modello all'altro

Il modello di Moore ha più stati ma funzioni di uscita più semplici

Automa a stati finiti

- A seconda del progetto e della descrizione a parole si può decidere di realizzare un automa di **Moore** o di **Mealy**. Di solito l'automa di **Mealy** ha meno stati (quindi meno elementi di memoria) ma ha le reti combinatorie ed in particolare la rete combinatoria delle uscite più complessa e quindi potenzialmente più lenta. Spesso si realizza l'automa di **Moore** perché è concettualmente più semplice.

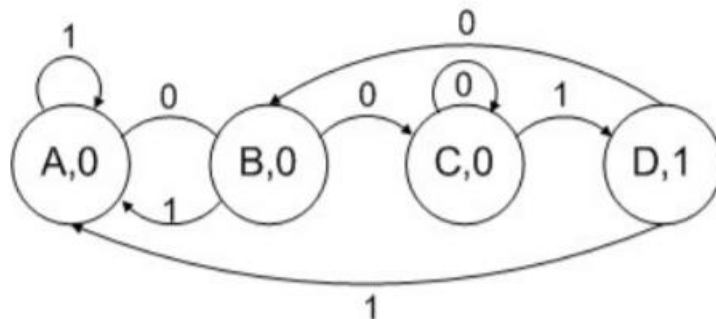


Diagramma di Moore

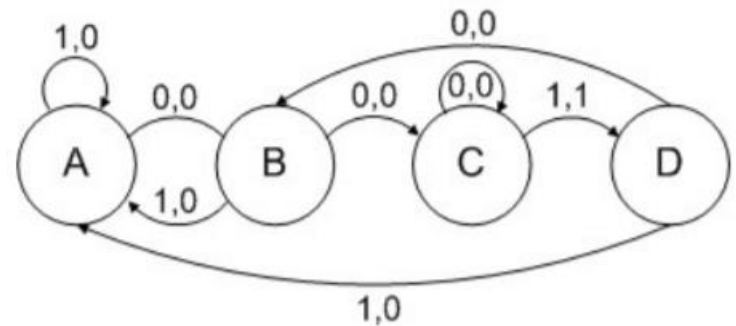


Diagramma di Mealy

Analisi di Reti Sequenziali Sincrone

- ▶ Data una rete sequenziale sincrona, vogliamo darne una descrizione esauriente tramite:
 - ▶ **Equazioni booleane,**
 - ▶ **Tabelle di verità.**
 - ▶ **Diagrammi di Stato.**
- ▶ Sappiamo che ad ogni uscita di una rete combinatoria corrisponde un'espressione booleana.

Analisi di Reti Sequenziali Sincrone

- ▶ Dato un circuito sequenziale sincrono, potremo prima di tutto costruire le **equazioni di ingresso ai flip flop**:
 - ▶ Associamo un nome ad ogni flip-flop incluso nella rete.
 - ▶ Associamo una variabile booleana ad ogni ingresso e ad ogni uscita di ogni flip-flop incluso nella rete (per esempio, J_A , K_A , D_A , etc.).
 - ▶ Ad ogni ingresso di ogni flip-flop, potremo costruire un'equazione $V = E$ dove V è la variabile associata all'ingresso ed E è l'espressione relativa a V costruita a partire dalla rete combinatoria.
- ▶ In modo analogo, possiamo costruire **equazioni di uscita** per ogni uscita della rete sequenziale.

Tabella di Stato

- ▶ A partire dalle equazioni di ingresso relative ad un circuito sequenziale, è possibile costruire una tabella, detta **tabella di stato**, che metta in relazione ingressi, uscite e stato.
- ▶ Ogni tabella di stato è formata da quattro sezioni.
- ▶ Nella sezione **Stato presente** sono elencati tutti i possibili stati per i flip-flop del circuito.
- ▶ Nella sezione **Ingressi** sono riportati i valori degli ingressi per ogni possibile stato presente.

Tabella di Stato

- ▶ Nella sezione **Stato futuro** sono elencati i valori dello stato futuro per ogni combinazione dei valori dello stato presente e degli ingressi.
- ▶ Nella sezione **Uscite** sono elencati i valori delle uscite per ogni possibile combinazione dei valori dello stato presente e degli ingressi.
- ▶ È possibile costruire la tabella di stato monodimensionalmente o bidimensionalmente.

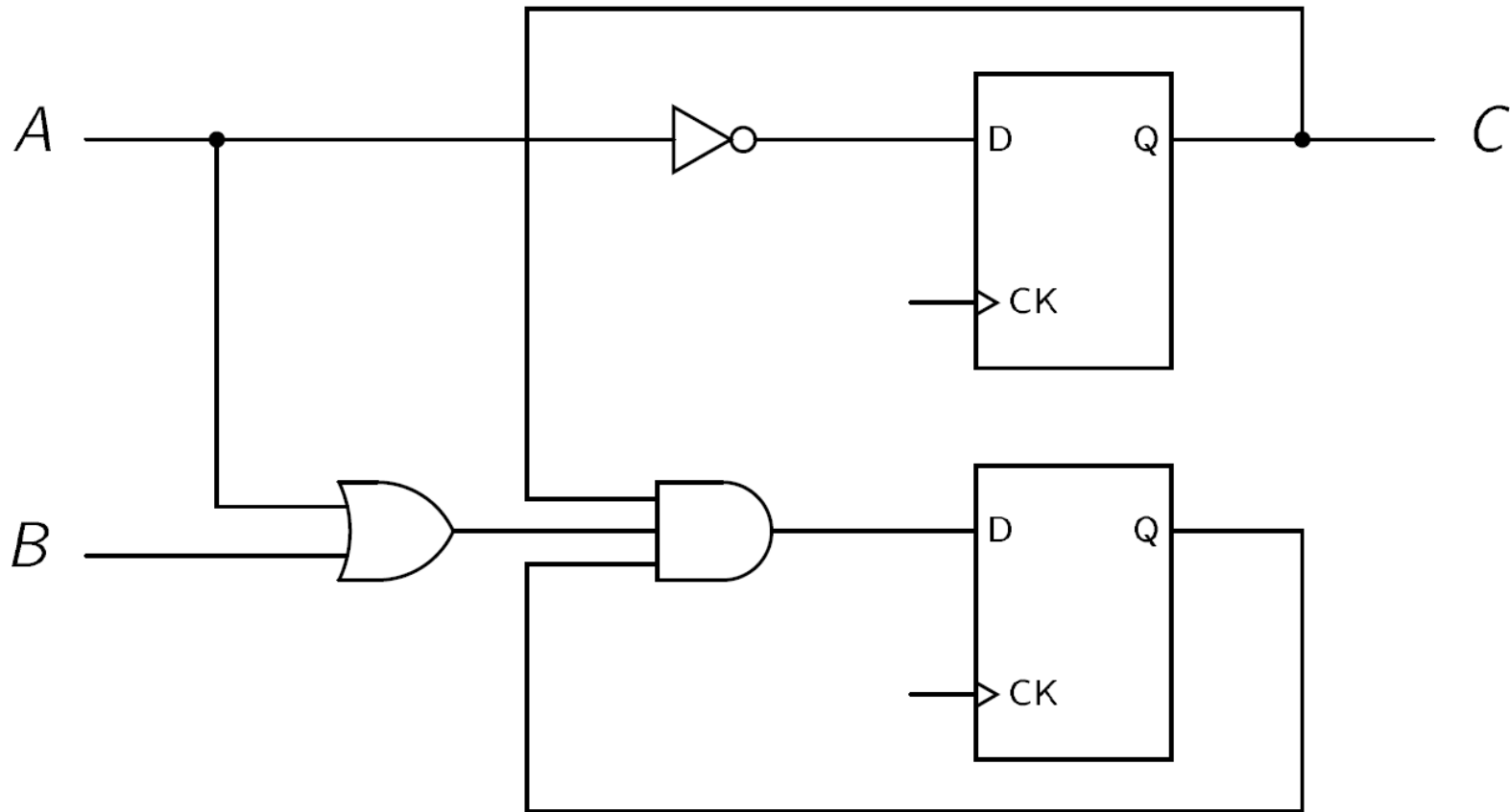
Diagramma di Stato

- ▶ Le informazioni presenti nella tabella di stato possono poi essere rappresentate graficamente nel cosiddetto **diagramma di stato**.
- ▶ Nel diagramma di stato:
 - ▶ Ogni possibile valore dello stato interno è rappresentato da un **cerchio**.
 - ▶ Se in corrispondenza di uno stato interno S (e di un valore V per gli ingressi) lo stato futuro è T , allora esisterà una freccia (etichettata con V) che collega S a T .

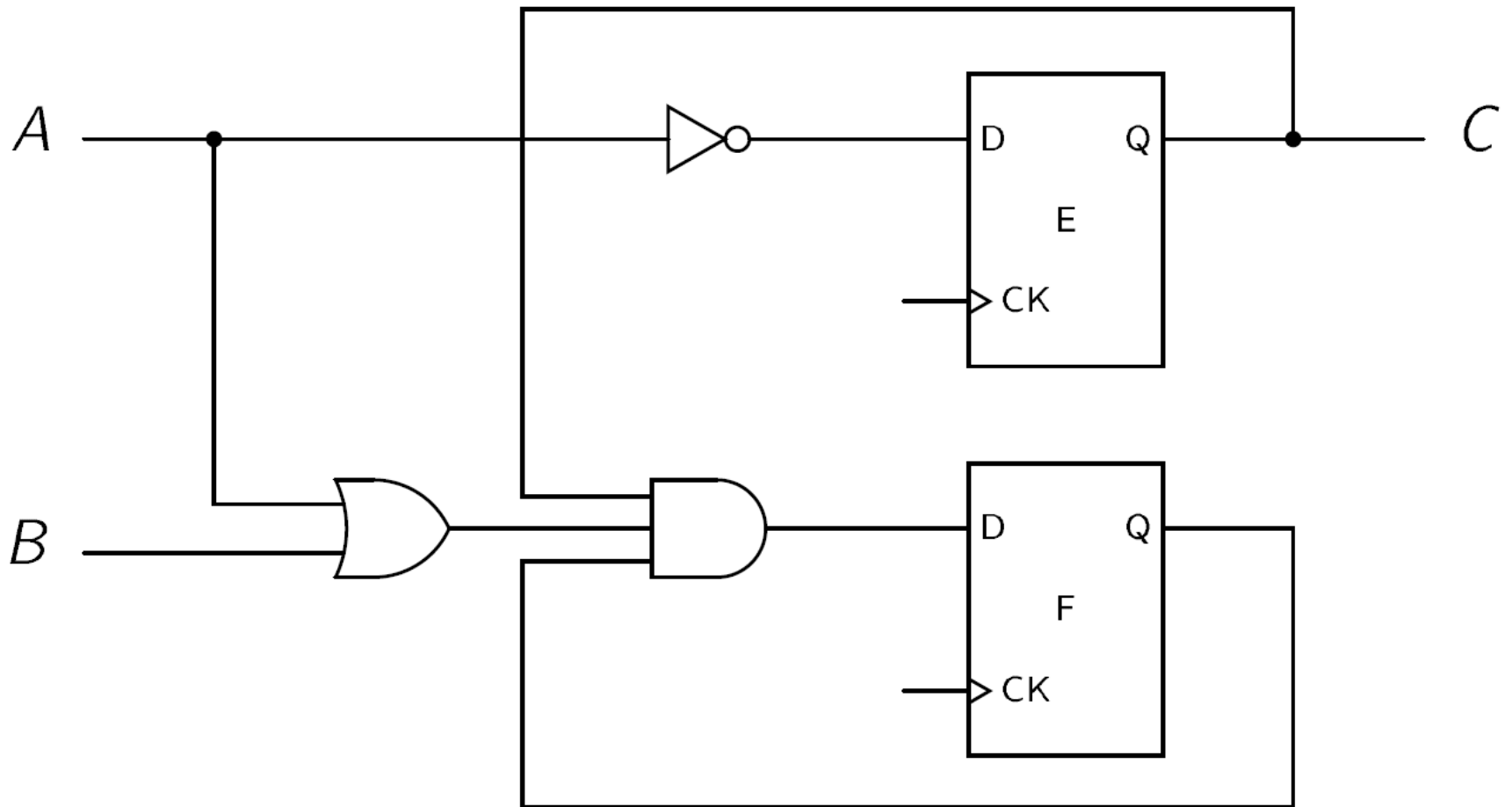
Diagramma di Stato

- ▶ Nei circuiti di tipo Mealy le uscite dipendono sia dallo stato che dagli ingressi. Di conseguenza, occorrerà specificare il loro valore sulle **freccie**.
- ▶ Nei circuiti di tipo Moore, invece, le uscite dipendono solo dallo stato. Di conseguenza, basterà specificare il loro valore sugli **stati**.

Esempio di Analisi di Reti Sequenziali Sincrone



Esempio di Analisi di Reti Sequenziali Sincrone



Esempio di Analisi di Reti Sequenziali Sincrone

**Equazioni di Ingresso ai Flip-Flop
e Equazioni di Uscita**

$$D_E = \overline{A}$$

$$D_F = E(A + B)F$$

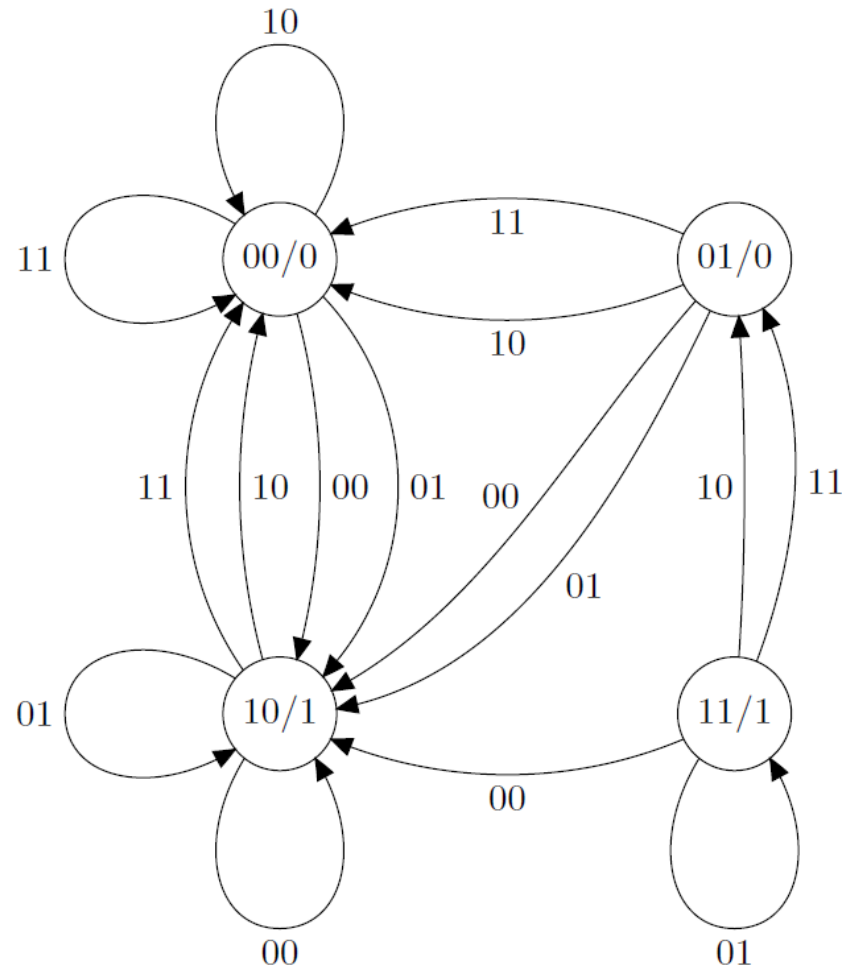
$$C = E$$

Esempio di Analisi di Reti Sequenziali Sincrone

Tabella di Stato

Stato presente		Ingressi		Stato futuro		Uscite
E	F	A	B	D_E	D_F	C
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	0	1	1	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	0	1	1	0	0	1
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	0	1	1
1	1	1	1	0	1	1

Esempio di Analisi di Reti Sequenziali Sincrone



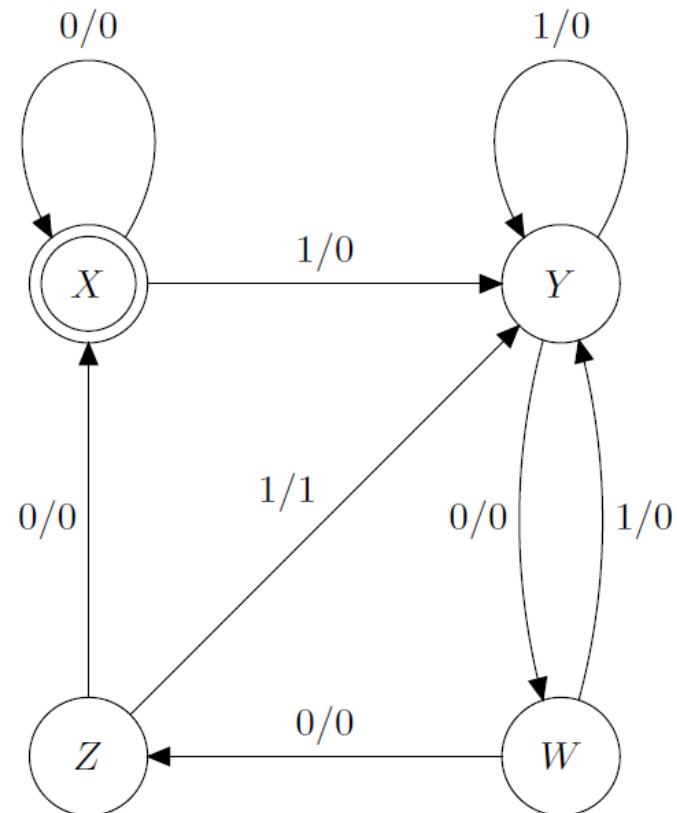
Sintesi di reti sequenziali

- ❑ Si parte da una descrizione comportamentale
- ❑ Si definisce il diagramma degli stati per definire le transizioni che si traduce nella tabella di flusso. Questa è la fase più importante in quanto definisco gli stati interni e le transizioni
- ❑ Si impiegano metodi manuali o automatici per la minimizzazione degli stati. Spesso il diagramma degli stati può essere minimizzato con un numero minore di stati (esistono algoritmi appositi).
- ❑ Dal diagramma minimizzato e tabella di flusso corrispondente si crea la tabella delle transizioni e delle uscite con l'assegnamento degli stati (indicando quale numero binario corrisponde ad ogni stato, date le variabili di stato presente e futuro).
- ❑ Infine si ottiene la implementazione (avendo scelto i componenti bistabili elementari e i gate elementari per le reti combinatorie).

Esempio di sintesi

- ▶ Vogliamo costruire una rete sequenziale che riconosca la presenza di una certa sequenza di bit fissata, anche se inclusa in una sequenza più lunga.
- ▶ La sequenza di bit che vogliamo riconoscere è 1001.
- ▶ La rete dovrà avere un ingresso A e un'uscita B e dovrà riconoscere la sequenza di bit 1001 applicata all'ingresso A .
- ▶ Più precisamente, l'uscita dovrà valere 1 se e solo se:
 - ▶ In corrispondenza ai 3 tre precedenti fronti di salita del clock, i valori letti in A erano, rispettivamente, 1, 0 e 0.
 - ▶ Il valore attuale dell'ingresso A è 1.

Esempio di sintesi



Esempio di sintesi

Tabella di Stato Implicita

Stato presente	Ingressi	Stato futuro	Uscite
	<i>A</i>		<i>B</i>
<i>X</i>	0	<i>X</i>	0
<i>X</i>	1	<i>Y</i>	0
<i>Y</i>	0	<i>W</i>	0
<i>Y</i>	1	<i>Y</i>	0
<i>W</i>	0	<i>Z</i>	0
<i>W</i>	1	<i>Y</i>	0
<i>Z</i>	0	<i>X</i>	0
<i>Z</i>	1	<i>Y</i>	1

Esempio di sintesi

- ▶ Per catturare quattro stati diversi abbiamo bisogno di sequenze di bit lunghe 2.
- ▶ Introduciamo le due variabili booleane E e F . Facciamo corrispondere gli stati agli assegnamenti di verità alle due variabili E e F come segue:

Stato	E	F
X	0	0
Y	0	1
W	1	0
Z	1	1

- ▶ Abbiamo bisogno, quindi, di due flip-flop, che chiameremo E e F .

Esempio di sintesi

Tabella di Stato Esplicita

Stato presente		Ingressi	Stato futuro		Uscite
E	F	A	D_E	D_F	B
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1

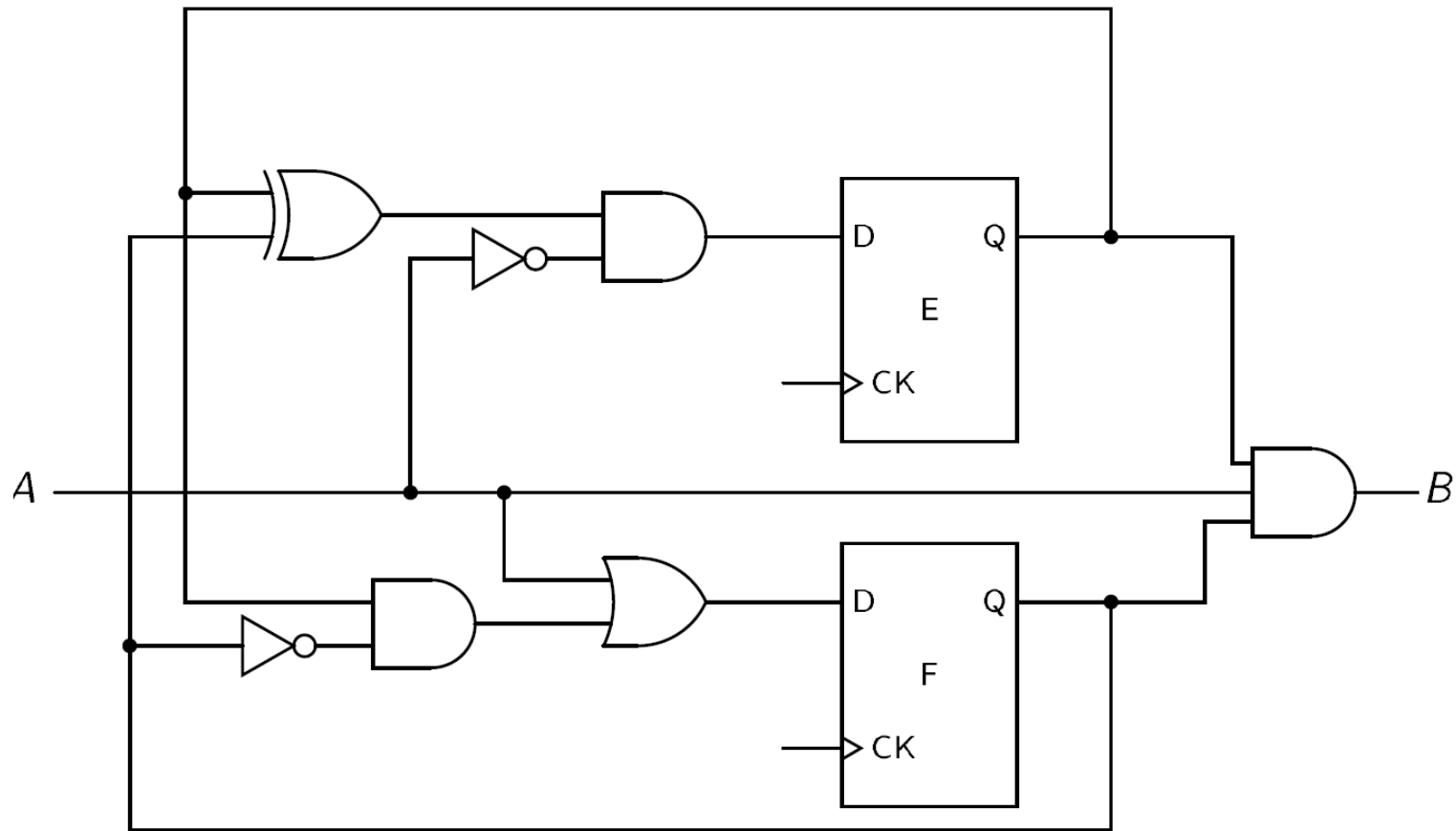
Esempio di sintesi

$$B = EFA$$

$$\begin{aligned} D_E &= \overline{E}F\overline{A} + E\overline{F}\overline{A} = \\ &= (\overline{E}F + E\overline{F})\overline{A} \\ &= (E \oplus F)\overline{A} \end{aligned}$$

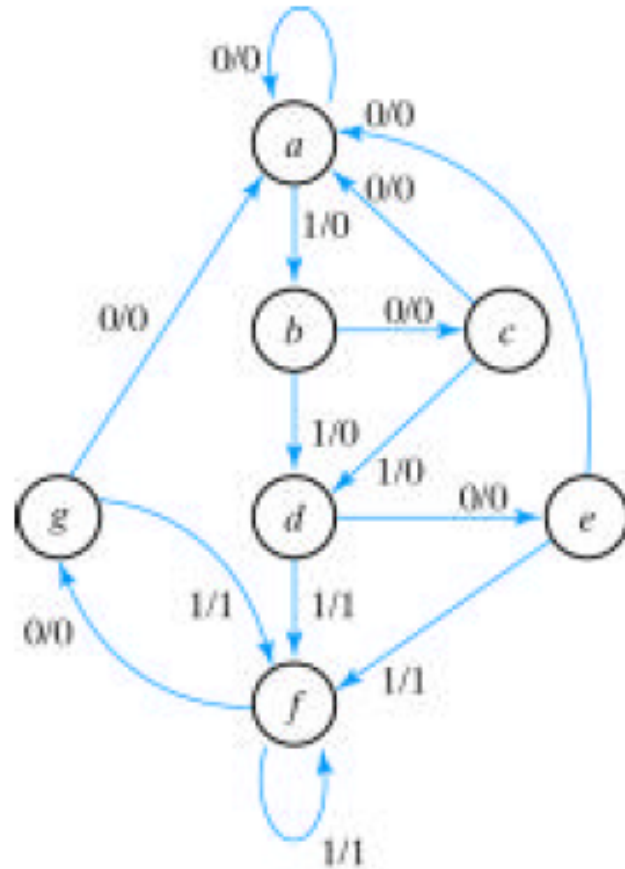
$$\begin{aligned} D_F &= \overline{E}\overline{F}A + \overline{E}FA + E\overline{F}\overline{A} + E\overline{F}A + EFA \\ &= \overline{E}\overline{F}A + \overline{E}FA + E\overline{F}A + EFA + E\overline{F}\overline{A} \\ &= (\overline{E}\overline{F} + \overline{E}F + E\overline{F} + EF)A + E\overline{F}\overline{A} \\ &= A + E\overline{F}\overline{A} \\ &= (A + E\overline{F})(A + \overline{A}) \\ &= A + E\overline{F} \end{aligned}$$

Esempio di sintesi



Riduzione degli stati

- La riduzione del numero di FF in una rete sequenziale (RS) prende il nome di *riduzione degli stati*.
- Infatti a ciascun FF è associata una variabile di stato, quindi una coppia di stati possibili.
- Ad m FF sono associati 2^m stati possibili
- Spesso, una riduzione di stati comporta un incremento di complessità della rete combinatoria (RC).
- Si consideri la RS specificata dal diagramma di stato (DS) in figura.



Riduzione degli stati (esempio)

- I 7 stati sono indicati con una etichetta: a, b, c, d, e, f, g .
 - La rete presenta un ingresso (X) e una uscita (Y). Si tratta di una RS di tipo Mealy.
 - Occorre prestare attenzione alle sequenze di ingresso-uscita. Esse sono ovviamente infinite.
 - Per es., se a partire dallo stato a si presentasse in ingresso la sequenza $X=01010110100$, l'analisi del DS consente la determinazione della uscita corrispondente:
- | | | | | | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $X:$ | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| stato: | a | a | b | c | d | e | f | f | g | f | g |
| $Y:$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

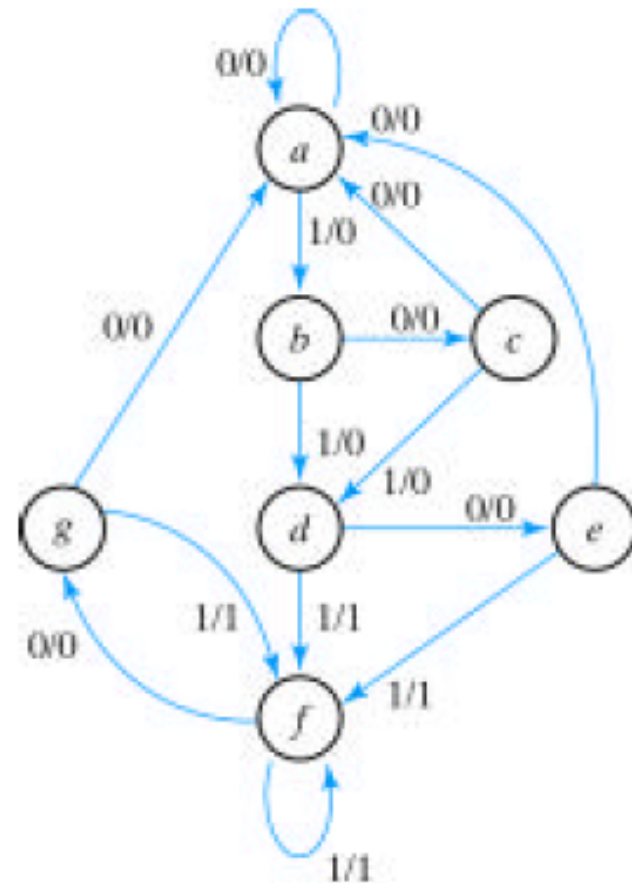


Tabella di Transazione di Stato

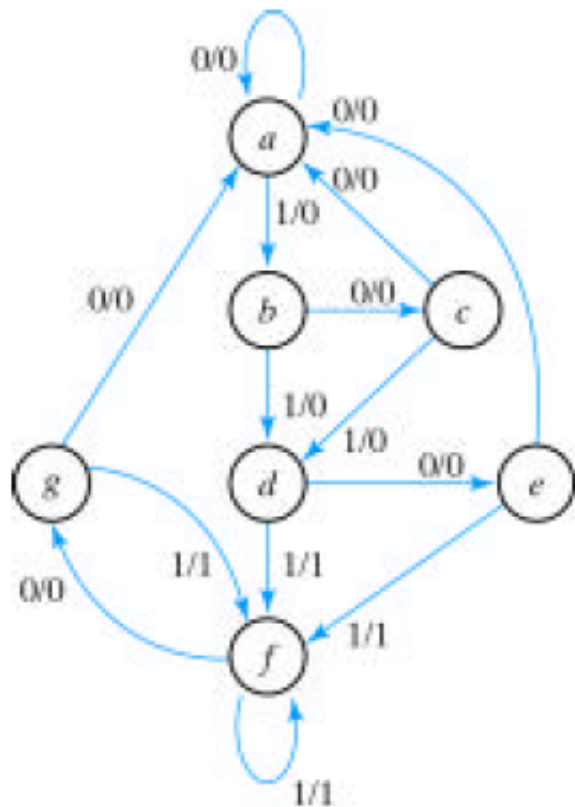


Fig. 5-22 State Diagram

Stato Presente	Stato Futuro		Uscita (Y)	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

- A partire dal DS si determina la tabella di transizione di stato.
- La riduzione degli stati ha come obiettivo la trasformazione di un DS in uno equivalente, per il quale la sequenza ingresso-uscita rimane immutata
- La riduzione degli stati si basa sul concetto di *stati equivalenti*.

Stati equivalenti

- E' possibile dare la definizione seguente di stati equivalenti per tabelle di transizione di stato **completamente specificate**.
 - Due stati si dicono equivalenti se per ciascuna variabile di ingresso essi producono esattamente le stesse uscite ed evolvono nello stesso stato o in stati equivalenti.
- In altre parole, perché due stati siano equivalenti occorre verificare che, per ciascuna configurazione degli ingressi (ovvero arco che origina nello stato in esame), si produca la stessa uscita e si arrivi al medesimo stato o a stati equivalenti.
- Quando due stati sono equivalenti, è possibile rimuovere uno dei due senza alterare il comportamento esterno della RS.

Stati equivalenti

- E' possibile identificare gli stati equivalenti direttamente sul DS.
- Gli stati g ed e sono equivalenti perché per entrambi terminano in a con uscita $Y=0$ e per entrambi terminano in f con uscita $Y=1$.

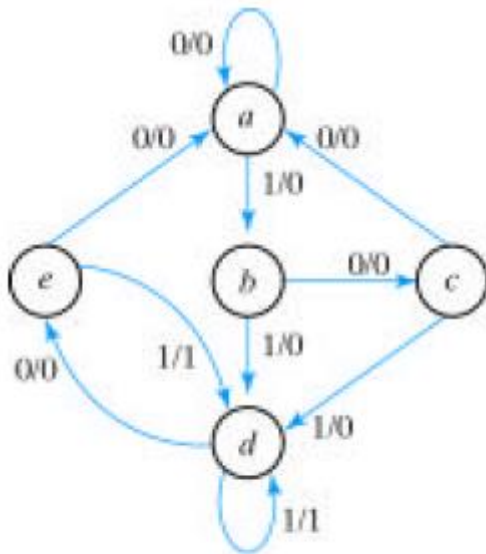


Fig. 5-23 Reduced State Diagram

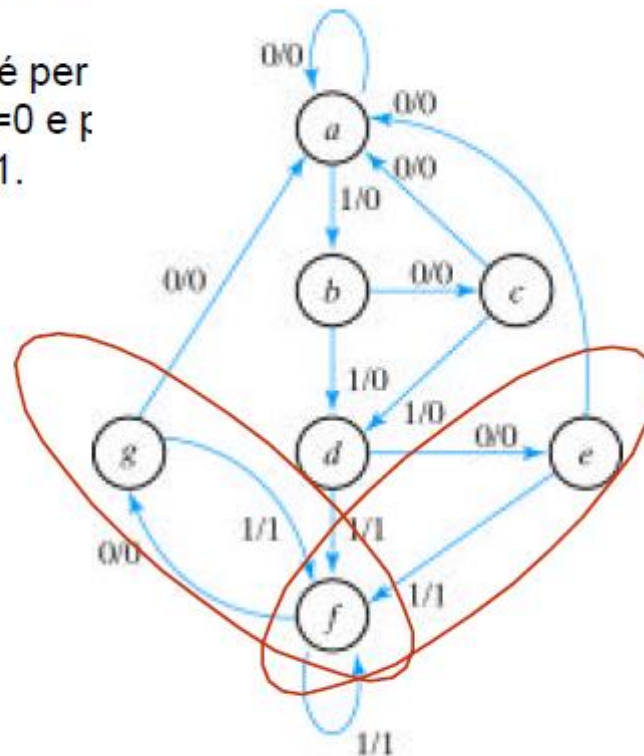


Fig. 5-22 State Diagram

Riduzione degli stati

- Un algoritmo sistematico di riduzione di stati si basa sulla tabella di transizione di stato.
- Si analizzano tutte le coppie di stati, S_i ed S_j per i quali coincidono lo stato futuro e l'uscita per ogni configurazione degli ingressi.
- Se tale coppia di stati esiste, essi sono equivalenti e uno può essere rimosso.
- Nella tabella, ad ogni istanza dello stato rimosso viene sostituito lo stato equivalente.
- La procedura viene ripetuta sino a quando non esistono più stati equivalenti.

Stato Presente	Stato Futuro		Uscita (Y)	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Stato Presente	Stato Futuro		Uscita (Y)	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Tabella degli Implicanti

- In alcuni casi, l'equivalenza tra due stati dipende dalla eventuale equivalenza di un'altra coppia di stati, come nel caso di d ed f
- La tabella di implicazione estende il metodo di riduzione visto, esplorando sistematicamente tutte le possibili equivalenze
- Si costruisce una tabella triangolare di tutte le possibili coppie di stati, prese una sola volta
- Ciascuna cella conterrà
 - *X* se gli stati non sono equivalenti
 - *~* se i due stati sono equivalenti
 - *L'*elenco di stati da cui eventualmente dipende l'equivalenza

B	A,C B,D					
C	B,D	A,C				
D	X	X	X			
E	X	X	X	A,E		
F	X	X	X	E,G	A,G	
G	X	X	X	A,E	~	A,G
	A	B	C	D	E	F

Tabella degli Implicanti

- Nella seconda passata, si analizzano le dipendenze indicate
 - *A e B non sono equivalenti perché non lo sono B e D*
 - *A e C non sono equivalenti perché non lo sono B e D*
 - *D ed E non sono equivalenti perché non lo sono A ed E*
 - *D ed F sono equivalenti essendolo E e G*
 - *D e G non sono equivalenti perché non lo sono A ed E*
 - *F e G non sono equivalenti perché non lo sono A e G*
- Gli stati sono quindi:
 - *a, b, c, d-f, e-g,*

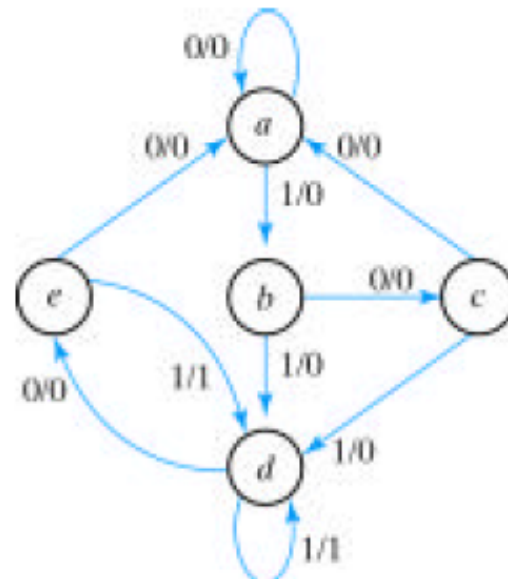
B	X					
C	X	X				
D	X	X	X			
E	X	X	X	X		
F	X	X	X	~	X	
G	X	X	X	X	~	X
	A	B	C	D	E	F

Tabella e diagrammi Ridotti

- La tabella ridotta con la eliminazione degli stati *f* e *g* è illustrata con il DS ridotto corrispondente.
- La verifica con la stessa sequenza di ingresso mostra che i due DS sono equivalenti.

X: 0 1 0 1 0 1 1 0 1 0 0
 stato: *a a b c d e d d e d e a*
 Y: 0 0 0 0 0 1 1 0 1 0 0

Stato Presente	Stato Futuro		Uscita (Y)	
	X=0	X=1	X=0	X=1
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>d</i>	0	1
<i>e</i>	<i>a</i>	<i>d</i>	0	1



Relazione di equivalenza circolari

- Si consideri la seguente tabella degli stati e la corrispondente tabella delle implicazioni

Stato Presente	Stato Futuro	
	X=0	X=1
A	G/00	C/01
B	G/00	D/01
C	D/10	A/11
D	C/10	B/11
E	G/00	F/01
F	F/10	E/11
G	A/01	F/11

B	C,D					
C	X	X				
D	X	X	A,B			
E	C,F	D,F	X	X		
F	X	X	A,E D,F	B,E C,F	X	
G	X	X	X	X	X	X
	A	B	C	D	E	F

Relazione di equivalenza circolari

- In alcuni casi, l'equivalenza tra due stati dà luogo a relazioni circolari come nel caso dell'esempio
 - $A \sim B$ se $C \sim D$ e $C \sim D$ se $A \sim B$
 - Si può imporre quindi che $A \sim B$ e $C \sim D$, e procedere con la riduzione
- In modo analogo risultano equivalenti
 - $A \sim E$ ed $C \sim F$

B	~					
C	X	X				
D	X	X	~			
E	C,F	D,F	X	X		
F	X	X	A,E D,F	B,E C,F	X	
G	X	X	X	X	X	X
	A	B	C	D	E	F

Relazione di equivalenza circolari

- La tabella delle implicazioni finali evidenzia quindi i soli stati
 - $A \sim B \sim E$
 - $C \sim D \sim F$
 - G
- Cui corrisponde la tabella ridotta seguente

Stato Presente	Stato Futuro/Uscite	
	X=0	X=1
a	c/00	b/01
b	b/10	a/11
c	a/01	b/11

B	~					
C	X	X				
D	X	X	~			
E	~	~	X	X		
F	X	X	~	~	X	
G	X	X	X	X	X	X
	A	B	C	D	E	F