



Sintesi Sequenziale Sincrona

Sintesi Comportamentale di reti Sequenziali Sincrone di Macchine Senza
Processo di Ottimizzate a Livello Comportamentale

Sintesi comportamentale e architettura generale

Diagramma degli stati

Tabella degli stati

Tabella delle Transizioni: Codifica dello Stato

Tabella delle Eccitazioni: Scelta degli elementi di memoria

Esempi

Appendice: Trasformazioni tra bistabili

versione 19/11/03



Introduzione

- Le uscite di un **circuito sequenziale** in un dato istante di tempo dipendono:
 - Dallo condizione iniziale del circuito;
 - Dalla sequenza di ingressi, applicata in un arco temporale finito, fino all'istante considerato
- Questo aspetto implica che il **dispositivo ha memoria degli eventi passati**
- In un generico istante t l'informazione relativa al “contenuto” di questa memoria è rappresentata nel **concetto di stato**
 - Nota: le reti combinatorie possono essere considerate un caso particolare di sistema sequenziale dove lo stato è unico



Progetto e strumenti

□ Progetto di reti combinatorie

- I metodi sono noti e ben assestati
- Si ricercano nuove soluzioni che aumentino l'efficienza:
 - Computazionale degli strumenti automatici
 - Nell'uso delle risorse (es. BDD)
 - Metodologica (es. *Signature Cube* - '92)

□ Progetto di reti sequenziali

- L'ottimizzazione di circuiti sequenziali è in costante evoluzione
 - Esistono buoni metodi ma non di uso generale
- Lo sviluppo di software efficienti necessita di ulteriori sforzi



Modello del circuito sequenziale

Il modello di un circuito sincrono può essere

- **Comportamentale** (descrive l'evoluzione degli stati e delle uscite del dispositivo)
 - La transizione degli stati è descritta in termini di tabelle o diagrammi
 - Le informazioni sugli stati sono esplicite
 - Le informazioni sull'area e sui ritardi sono implicite

- **Strutturale**
 - Il modello del circuito è una *netlist* ovvero un insieme di componenti, registri e logica combinatoria, collegati tra loro
 - Le informazioni sugli stati sono implicite
 - Le informazioni sull'area e ritardi sono esplicite



Modello comportamentale

- Il modello generale delle macchine sequenziali cui si fa riferimento è quello delle **Macchine a Stati Finiti Deterministiche** (*Finite State Machine - FSM*)
 - con questo modello le macchine sequenziali vengono descritte tramite la teoria degli automi
 - su questo modello si basano la metodologia e le tecniche di sintesi esposte
- **Macchine a stati finiti deterministiche**
 - **fisica realizzabilità**: il numero di stati è finito e il comportamento della macchina in un istante t non dipende da eventi futuri
 - dato uno stato ed una configurazione di ingresso il **nuovo stato** è **identificato univocamente**
- Si considera la **sintesi di FSM sincrone**



Sintesi comportamentale di FSM (1)

- Una **macchina sequenziale** è definita dalla quintupla $(I, U, S, \delta, \lambda)$
 - I - Alfabeto di Ingresso
 - E' costituito dall'insieme *finito* dei *simboli* di ingresso
 - Con n linee di ingresso si hanno 2^n simboli
 - U - Alfabeto d'Uscita
 - E' costituito dall'insieme *finito* dei *simboli* d'uscita
 - Con m linee d'uscita si hanno 2^m simboli
 - S - Insieme degli Stati
 - Insieme *finito* e *non vuoto* degli *stati*.
 - Spesso viene definito anche uno stato iniziale o **stato di reset**, in cui la macchina deve portarsi all'accensione o all'applicazione del segnale di reset
 - δ - Funzione stato prossimo
 - λ - Funzione d'uscita
-



Sintesi comportamentale di FSM (2)

□ Funzione stato prossimo δ

- Ad ogni stato presente e per ogni simbolo di ingresso la funzione δ associa uno stato prossimo:

$$\delta : S \times I \rightarrow S$$

- Ad ogni coppia $\{stato, simbolo\}$ di ingresso è associato, se specificato, **uno ed uno** solo stato prossimo.
- Definisce l'evoluzione della macchina nel tempo, in risposta agli eventi in ingresso

□ Funzione d'uscita λ

- Genera il simbolo d'uscita
- **Macchine di Mealy**: l'uscita dipende sia dallo stato presente sia dall'ingresso:

$$\lambda : S \times I \rightarrow U$$

- **Macchine di Moore**: l'uscita dipende solamente dallo stato presente:

$$\lambda : S \rightarrow U$$



Macchine di Mealy e Macchine di Moore

□ Macchine di Mealy

- la funzione di uscita costituisce la risposta della macchina quando, trovandosi in un certo stato presente, riceve un simbolo di ingresso
- nelle macchine di Mealy, l'uscita va "letta" mentre la macchina subisce una transizione di stato

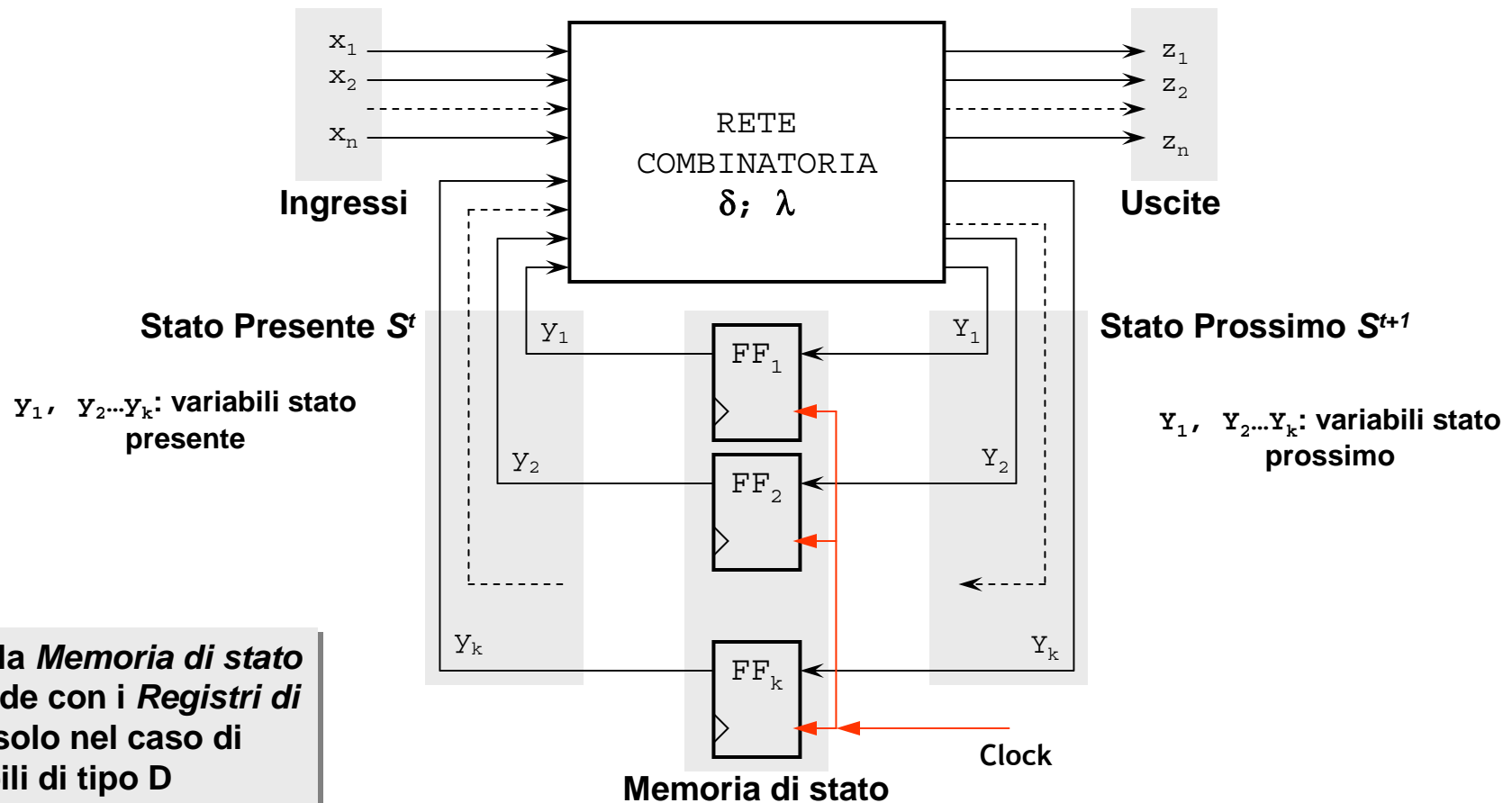
□ Macchine di Moore

- la funzione di uscita costituisce la risposta della macchina associata allo stato in cui si trova
 - nelle macchine di Moore, l'uscita viene letta mentre la macchina si trova in un determinato stato
-
- E' possibile trasformare una macchina di Mealy in una macchina equivalente di Moore, e viceversa



Architettura generale

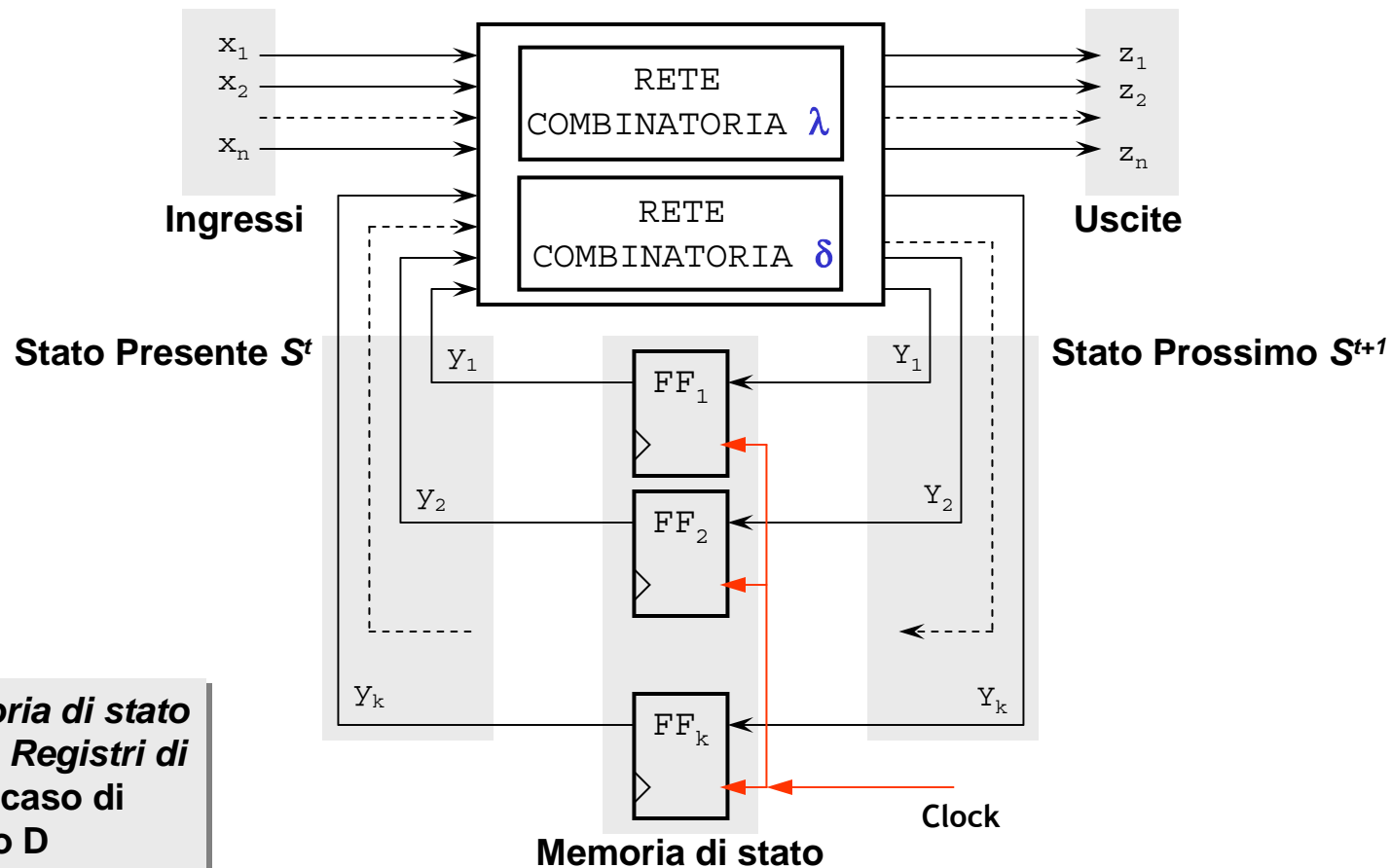
- Struttura generale di una macchina sequenziale (Huffman):





Architettura generale: macchina di Mealy

- Struttura generale di una macchina di Mealy:

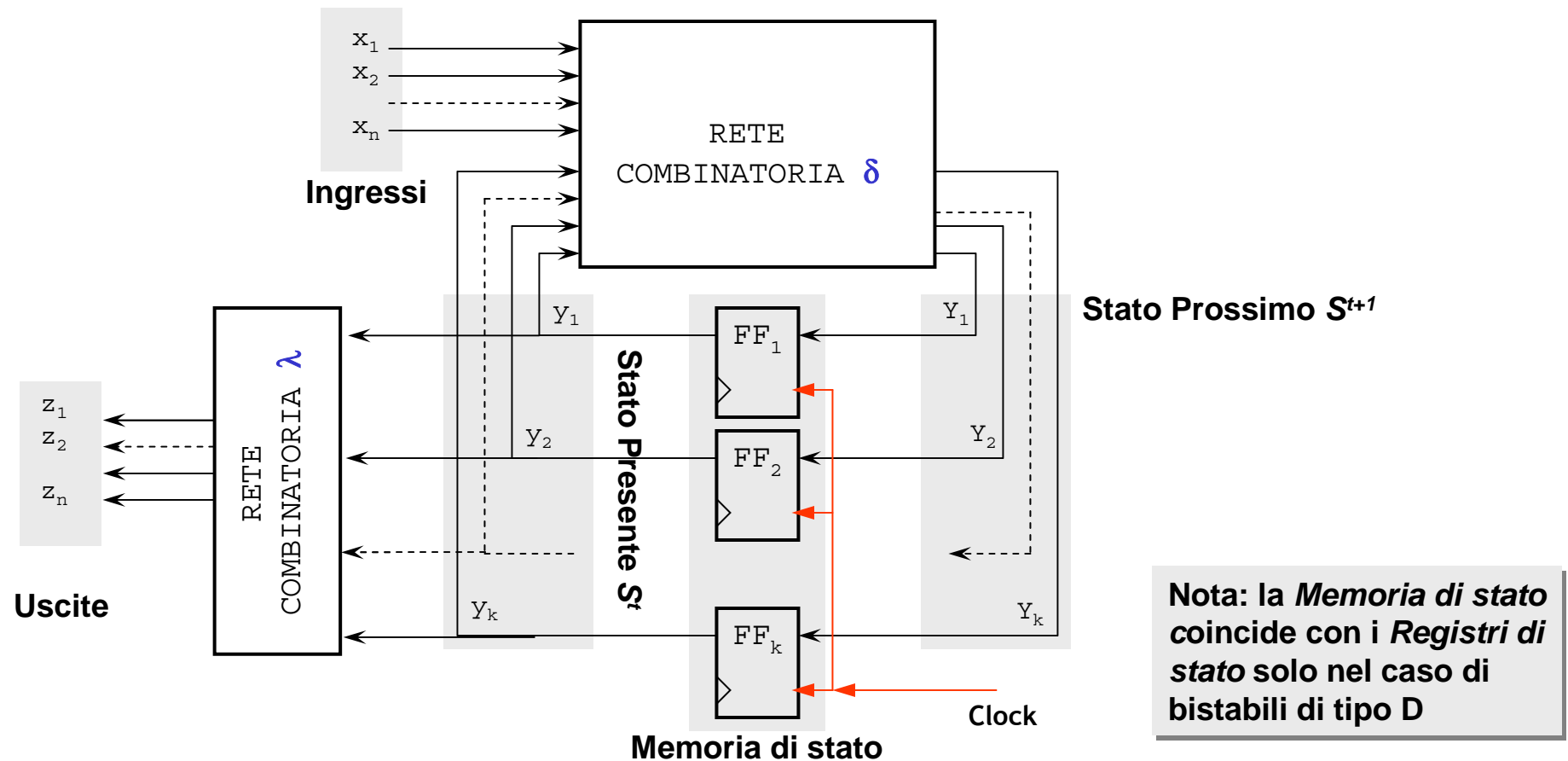


Nota: la Memoria di stato coincide con i Registri di stato solo nel caso di bistabili di tipo D



Architettura generale: macchina di Moore

- Struttura generale di una macchina di Moore:





Architettura generale

- La **sintesi comportamentale** di una rete sequenziale consiste nella:
 - Identificazione delle le funzioni δ e λ
 - Sintesi della rete combinatoria che le realizza
- Gli elementi di memoria sono costituiti da Flip-Flop
 - I flip-flop di tipo D sono quelli usati più comunemente
- La funzione di stato prossimo δ dipende dal tipo di bistabili utilizzati.
- La funzione di uscita λ non dipende dal tipo di bistabili utilizzati.



Architettura generale

- La funzione δ dipende dai bistabili utilizzati:

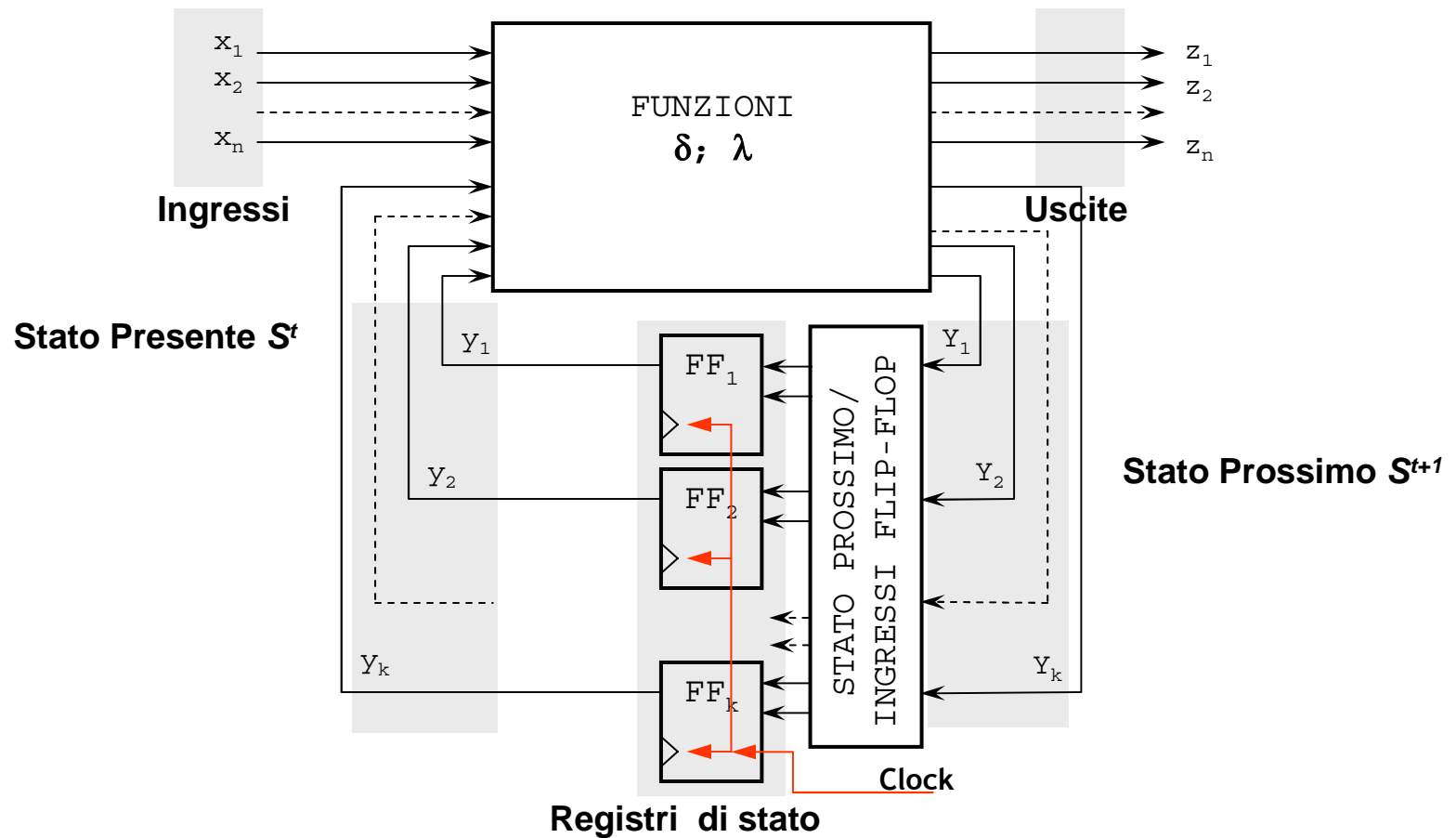




Tabella degli stati

Il comportamento di una FSM può essere descritto mediante la **Tabella degli stati**

- Gli indici di **colonna** sono i **simboli di ingresso** $i_\alpha \in I$
- Gli indici di **riga** sono i simboli di stato $s_j \in S$ che indicano lo **stato presente**

□ Gli elementi della tabella sono:

- **Macchine di Mealy**
- la **coppia** $\{u_\beta, s_j\}$
- $u_\beta = \lambda(i_\alpha, s_j)$ è il simbolo di uscita
- $s_j = \delta(i_\alpha, s_j)$ è il simbolo stato prossimo

	i_1	i_2	...
s_1^t	s_j^{t+1}/u_j	s_k^{t+1}/u_k	...
s_2^t	s_m^{t+1}/u_m	s_1^{t+1}/u_1	...
...

- **Macchine di Moore**
- Il simbolo **stato prossimo** s_j
- $s_j = \delta(i_\alpha, s_j)$ è il simbolo stato prossimo
- i simboli d'uscita sono associati allo stato presente

	i_1	i_2	...	
s_1^t	s_j^{t+1}	s_k^{t+1}	...	u_1
s_2^t	s_m^{t+1}	s_1^{t+1}	...	u_2
...



Diagramma degli stati

- Spesso, la stesura della *Tabella degli stati* è preceduta da una rappresentazione grafica ad essa equivalente, denominata *Diagramma degli stati*

- Il diagramma degli stati è un *grafo orientato* $G(V, E, L)$
 - V - Insieme dei *nodi*
 - Ogni nodo rappresenta uno stato
 - Ad ogni nodo è associato un simbolo d'uscita (macchine di Moore)
 - E - Insieme degli *archi*
 - Ogni arco rappresenta le transizioni di stato
 - Ad ogni arco è associato un simbolo di uscita (macchina di Mealy)
 - L - Insieme degli:
 - **Ingressi e Uscite** (macchina di Mealy)
 - **Ingressi** (macchina di Moore)



Macchina di Mealy: Esempio

- Equivalenza delle due rappresentazioni nel caso di una macchina di Mealy

Diagramma degli stati

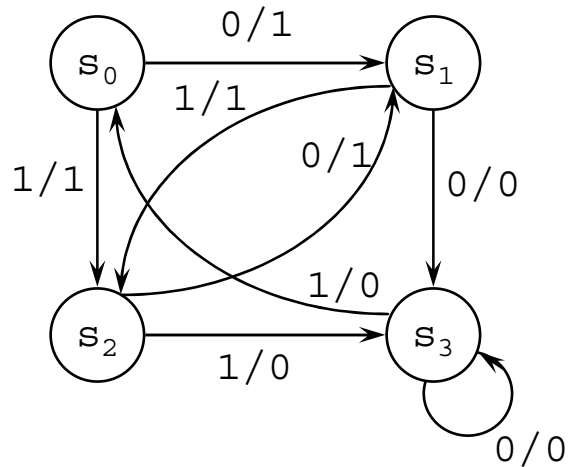


Tabella degli stati

	0	1
s ₀	s ₁ /1	s ₂ /1
s ₁	s ₃ /0	s ₂ /1
s ₂	s ₁ /1	s ₃ /0
s ₃	s ₃ /0	s ₀ /0



Macchina di Moore: *Esempio*

- Equivalenza delle due rappresentazioni nel caso di una macchina di Moore

Diagramma degli stati

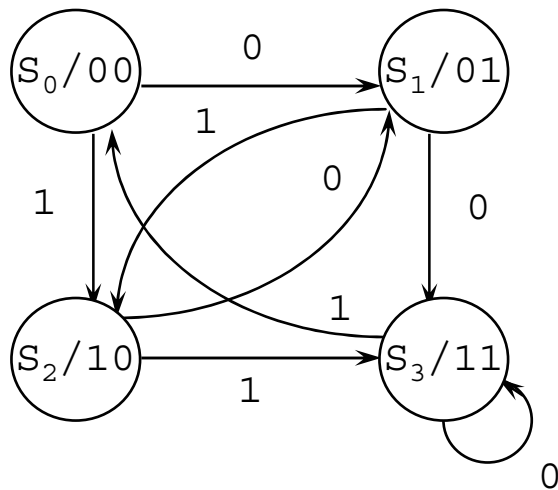


Tabella degli stati

	0	1	U
S ₀	S ₁	S ₂	00
S ₁	S ₃	S ₂	01
S ₂	S ₁	S ₃	10
S ₃	S ₃	S ₀	11



Passi della Sintesi di una FSM (i)

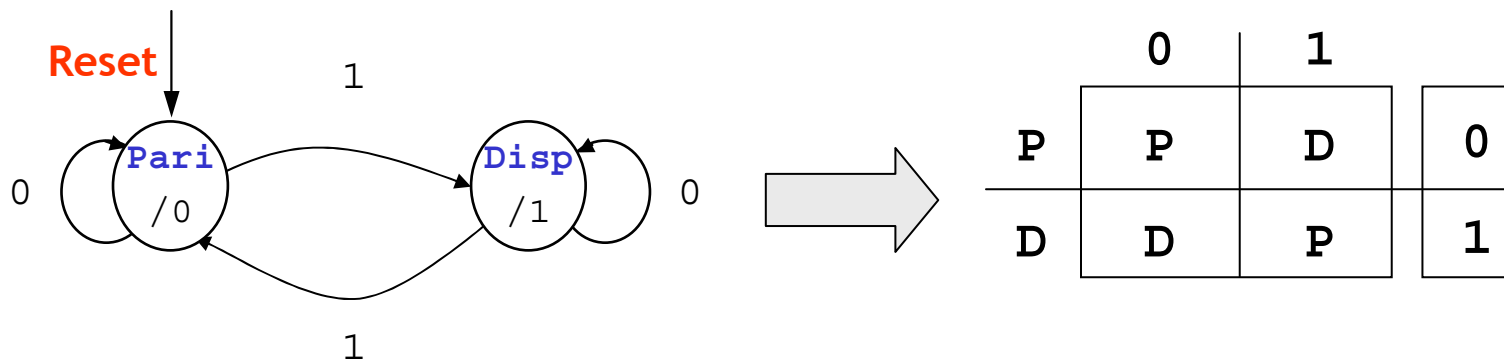
1. Realizzazione del *diagramma degli stati* a partire dalle specifiche funzionali (informali) del comportamento del sistema
 - e' il passo che richiede maggior intuito, anche se è più semplice rispetto al costruire direttamente la tabella degli stati
 - una volta identificato lo **stato iniziale**, si applicano a tale stato tutte le possibili configurazioni di ingresso
 - **stato iniziale**: identificazione univoca e non ambigua di uno stato da cui iniziare la stesura del diagramma degli stati
 - **stato di reset**: se questo è definito esplicitamente nella specifica funzionale
 - ogni configurazione di ingresso può portare a uno stato già esistente oppure a un **nuovo stato** che viene aggiunto al diagramma
 - per ogni nuovo stato introdotto, si applicano tutte le sequenze di ingresso
 - Il procedimento termina quando **non vengono più introdotti nuovi stati**
-



Passi della Sintesi di una FSM (i)

Controllore di parità (dispari)

- Una macchina sequenziale sincrona ha un ingresso x e un'uscita z . L'uscita z assume il valore 1 se e solo se sull'ingresso si sono presentati un numero dispari di 1. In ogni altro caso è z uguale a 0. All'accensione la macchina riconosce parità dispari non verificata.
- Considerazioni:
 - specifiche funzionali analitiche: non è necessario ulteriore raffinamento
 - dalle specifiche, c'è uno stato di RESET esplicito e la macchina da sintetizzare è una macchina di Moore





Passi della Sintesi di una FSM (ii)

2. Costruzione della *tabella degli stati* a partire da diagramma degli stati

- non introduce alcuna informazione aggiuntiva. Definisce la cardinalità iniziale degli stati e le funzioni δ e λ in forma astratta

	0	1	U
S_0	S_1	S_2	00
S_1	S_3	S_2	01
S_2	S_1	S_3	10
S_3	S_3	S_0	11

3. Riduzione del numero degli stati: *ottimizzazione*

- Identificazione di una *macchina equivalente* (oppure di una *macchina compatibile*) *minima* a quella rappresentata dalla tabella degli stati
- I *criteri di riduzione* del numero degli stati per equivalenza e compatibilità verranno affrontati in seguito



Passi della Sintesi di una FSM (iii)

4. Costruzione della *tabella delle transizioni* della FSM

- In modo informale, si può dire che questo passo traduce la tabella degli stati in una *tabella rappresentata tramite funzioni di commutazione*.
 - Nota: ad una tabella degli stati corrispondono più tabelle delle transizioni

Definisce *l'assegnamento degli stati* e cioè

- il *numero di variabili di stato necessarie* (y_i : stato presente e Y_i : stato prossimo) a rappresentare la cardinalità degli stati
 - In questo modo viene determinato il *numero di flip-flop necessari* a realizzare la macchina
- assegna una *codifica* (configurazione tra quelle disponibili nel codice) ad *ogni stato*.
 - La scelta della codifica influenza in modo significativo la realizzazione e complessità circuitale della funzione stato prossimo δ (anche in funzione dei bistabili utilizzati)
- I criteri di assegnamento verranno affrontati in seguito



Passi della Sintesi di una FSM (iii)

- Costruzione della *tabella delle transizioni* della FSM

Tabella degli stati

	0	1	U
S ₀	S ₁	S ₂	00
S ₁	S ₃	S ₂	01
S ₂	S ₁	S ₃	10
S ₃	S ₃	S ₀	11

Assegnamento degli stati

- 2 variabili di stato Y_0Y_1

(quindi 2 bistabili)

- Assegnamento banale

$$S_0 = 00$$

$$S_1 = 01$$

$$S_2 = 11$$

$$S_3 = 10$$

Tabella delle transizioni

Y_0Y_1 \ I	0	1	U
00	01	11	00
01	10	11	01
11	01	10	10
10	10	00	11

$$Y_0Y_1 =$$

stato prossimo



Passi della Sintesi di una FSM (iv)

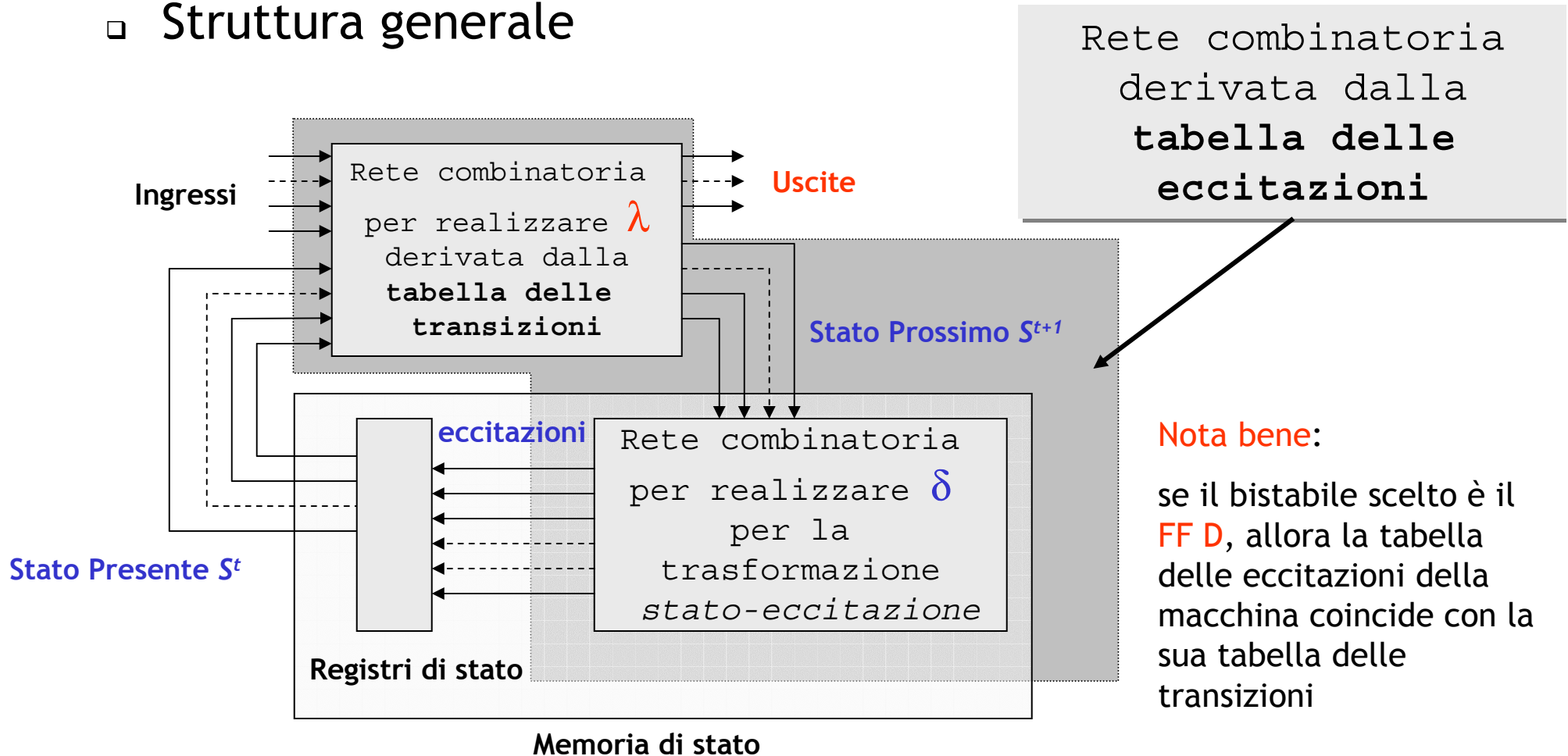
5. Costruzione della *tabella delle eccitazioni* della FSM
 - Scelta degli elementi di memoria.
 - Una volta scelti gli elementi di memoria, la tabella delle eccitazioni della macchina è **ottenuta dalla tabella delle transizioni della macchina e da quella delle eccitazioni del bistabile scelto**
 - Al termine di questo passo, per ogni bistabile (e cioè per ogni variabile di stato) si hanno le funzioni di commutazione relative ai suoi ingressi che consentono le transizione stato presente - stato prossimo

6. *Sintesi ottimizzata* sia della **rete combinatoria che realizza la funzione stato prossimo** sia della **rete combinatoria che realizza la funzione d'uscita**



Passi della Sintesi di una FSM (iv)

□ Struttura generale





Passi della Sintesi di una FSM (iv)

- La tabella delle transizioni descrive la relazione tra i bit di stato presente e quelli di stato futuro.
 - La configurazione binaria dello stato presente corrisponde all'uscita dei bistabili relativi
 - La configurazione binaria dello stato prossimo precisa quello che si desidera ottenere
- A seconda del tipo di bistabile scelto, variano i segnali che devono essere generati per realizzare la transizione *stato presente - stato prossimo* desiderata
 - I segnali di ingresso di un bistabile prendono il nome di *eccitazioni*
- La *tabella delle eccitazioni* di un bistabile rappresenta il mezzo di collegamento tra la **tabella delle transizioni** e la **tabella delle eccitazioni** di una specifica macchina a stati.



Passi della Sintesi di una FSM (iv)

- Si scelgono i bistabili SR

Tabella delle transizioni

$Y_0Y_1 \backslash I$	0	1	U
00	01	11	00
01	10	11	01
11	01	10	10
10	10	00	11

$Y_0Y_1 =$

stato prossimo

Tabella delle eccitazioni del FF SR

Q	Q*	C	S	R
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	0
1	0	1	0	1
1	1	1	-	0

Tabella delle eccitazioni con FF SR

$Y_0Y_1 \backslash I$	0	1	U
00	0-, 10	10, 10	00
01	10, 01	10, -0	01
11	01, -0	-0, 01	10
10	-0, 0-	01, 0-	11

$S_0R_0 \ S_1R_1$

Dalla tabella delle eccitazioni posso sintetizzare le reti combinatorie (mappe di Karnaugh) che realizzano $S_0R_0 \ S_1R_1$ in funzione di y_0, y_1 e I



Diagramma degli stati - Esempio 1: specifiche

Specifiche

- Una macchina sequenziale sincrona ha un ingresso x e un'uscita z . L'uscita z assume il valore 1 se e solo se sull'ingresso si sono presentati almeno due 0 seguiti esattamente da due 1 (z va a 1 in corrispondenza del secondo 1 su x). In ogni altro caso è z uguale a 0.
- Considerazioni:
 - specifiche funzionali analitiche: non è necessario ulteriore raffinamento delle specifiche
 - dalle specifiche, la macchina da sintetizzare è una macchina di Mealy
 - la macchina è un riconoscitore di sequenze nella forma

$$\begin{array}{l} \mathbf{x} = \dots 0011\dots \\ \mathbf{z} = \dots 00010\dots \end{array}$$

The diagram shows two binary sequences, x and z. The sequence x is "...0011..." and the sequence z is "...00010...". A red arrow points from the first '1' in x to the second '1' in x. A blue arrow points from the second '0' in z to the '1' in z.



Diagramma degli stati - Esempio 1: stato iniziale - caso (a)

(a) Scelta dello **stato iniziale** per la stesura del diagramma degli stati

- dalle specifiche: “ z assume il valore 1 se e solo se sull’ingresso si sono presentati almeno due 0 seguiti esattamente da **due 1**.....”
- una **sequenza di tre o più 1 su x**, indipendentemente dalla successione di valori di x ricevuti precedentemente, porta la macchina in uno stato in cui “si aspetta” una sequenza da riconoscere. Inoltre, sicuramente al terzo 1 l’uscita vale 0. Chiamiamo questa sequenza di tre o più 1 “**non utile**” a fini del riconoscimento
- la sequenza di esattamente tre 1 su x è la minima sequenza non utile
 - Nota: una sequenza di soli due 1 non sarebbe univoca per l’uscita, in quanto in corrispondenza del secondo 1 l’uscita potrebbe valere 1 (sequenza precedente riconosciuta) oppure 0 (sequenza precedente non riconosciuta)
- **stato iniziale** = stato derivante da tre 1 su x, indipendentemente dai valori precedenti



Diagramma degli stati - Esempio 1 - caso (a)

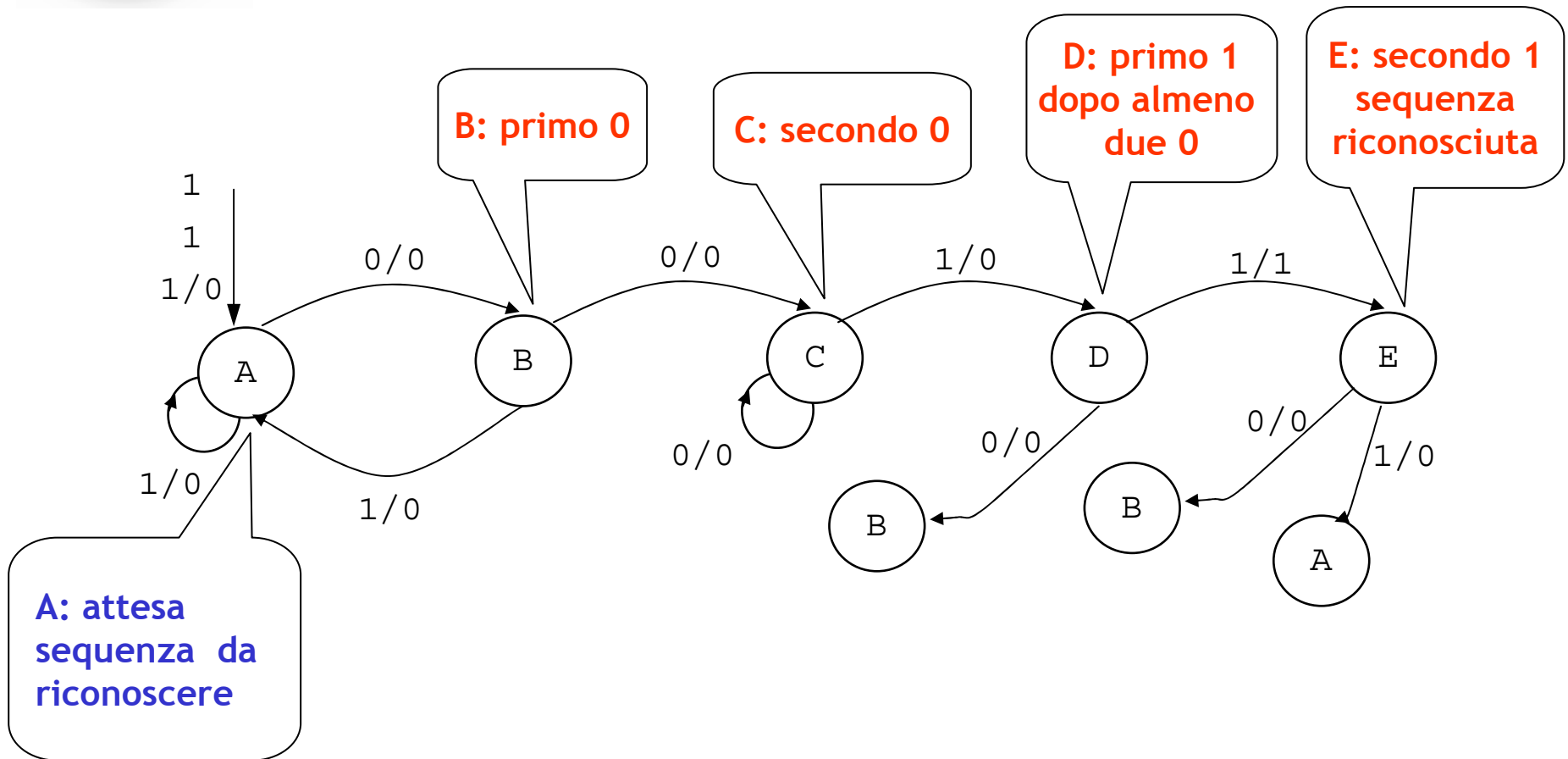




Tabella degli stati - Esempio 1 - caso (a)

Tabella degli stati

	0	1
A	B, 0	A, 0
B	C, 0	A, 0
C	C, 0	D, 0
D	B, 0	E, 1
E	B, 0	A, 0

Riduzione della tabella degli stati (banale!!)

	0	1
A	B, 0	A, 0
B	C, 0	A, 0
C	C, 0	D, 0
D	B, 0	E, 1
A = E	B, 0	A, 0

	0	1
A	B, 0	A, 0
B	C, 0	A, 0
C	C, 0	D, 0
D	B, 0	A, 1



Diagramma degli stati - Esempio 1: stato iniziale - caso (b)

(b) Scelta dello **stato iniziale** per la stesura del diagramma degli stati

- dalle specifiche: “ z assume il valore 1 se e solo se sull’ingresso si sono presentati **almeno due 0** seguiti esattamente da due 1.....”
- una **sequenza di due o più 0 su x**, indipendentemente dalla successione di valori di x ricevuti precedentemente, porta la macchina in uno stato in cui “si è presentata la parte iniziale, indispensabile, della sequenza da riconoscere”. Inoltre, sicuramente l’uscita vale 0. Chiamiamo questa sequenza di due o più 0 “**utile**” a fini del riconoscimento
- la sequenza di esattamente due 0 su x è la minima sequenza utile
- **stato iniziale** = stato derivante da due 0 su x, indipendentemente dai valori precedenti



Diagramma degli stati - Esempio 1 - caso (b)

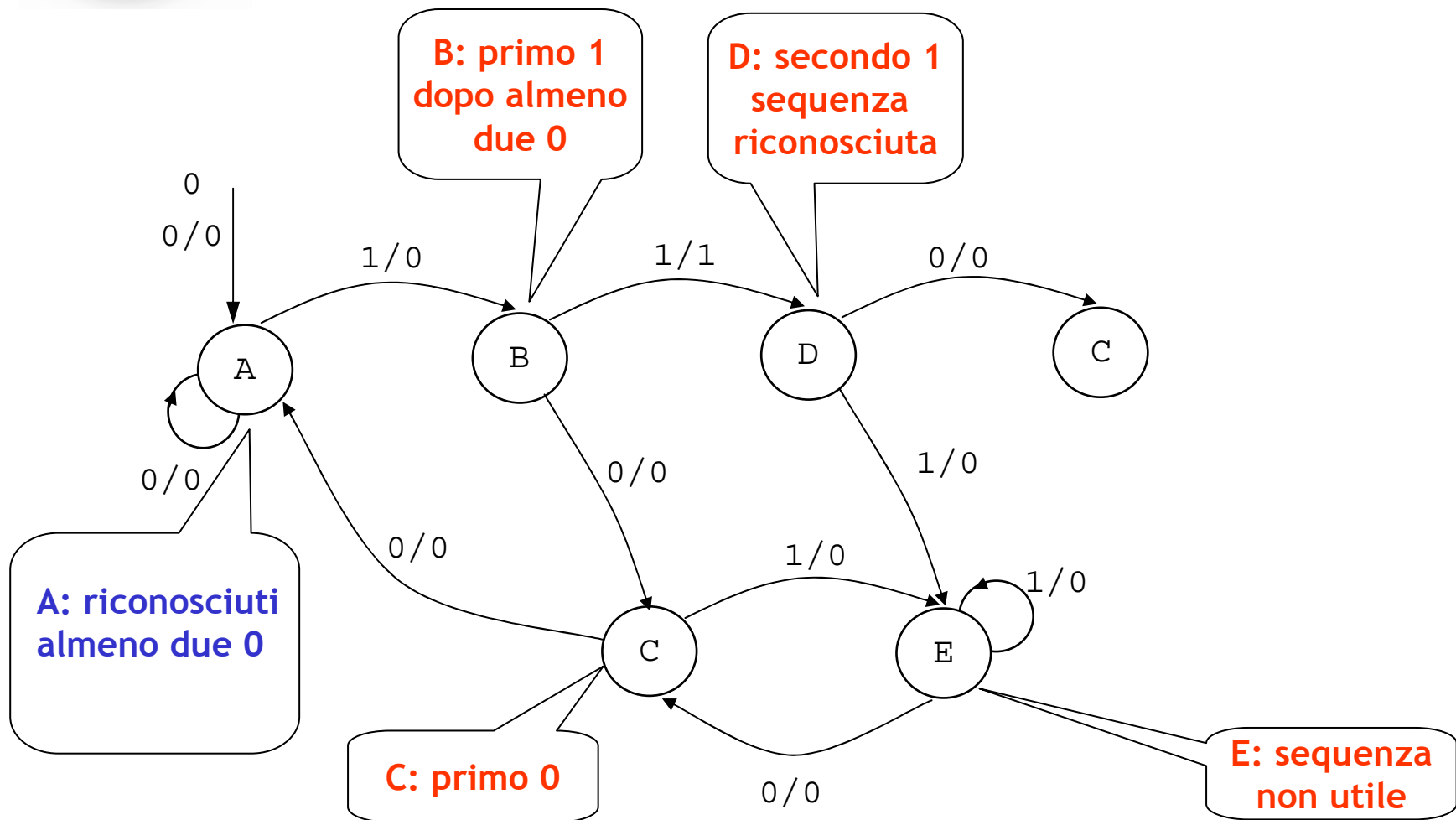




Diagramma degli stati - Esempio 2: specifiche (i)

Specifiche

- Si vuole realizzare un controllore di semaforo all'incrocio tra via Mazzini e via Garibaldi tramite una macchina sequenziale sincrona. La macchina riceve un segnale di sincronismo con periodo di un minuto. Esiste un pulsante P per attraversamento pedonale.
- Normalmente il semaforo alterna un minuto VERDE su via Mazzini e ROSSO su via Garibaldi, poi un minuto VERDE su via Garibaldi e ROSSO su via Mazzini, e così via
- Se si preme il pulsante P, alla scadenza del minuto si porta il ROSSO su entrambe le strade e lo si mantiene per due minuti indipendentemente da P
- al termine dei due minuti riparte il funzionamento normale con la configurazione VERDE-ROSSO per la via in cui precedentemente ai due minuti era ROSSO e successivamente, dopo una nuova alternanza, si prende in considerazione P



Diagramma degli stati - Esempio 2: specifiche (ii)

- Considerazioni:
 - le specifiche funzionali non sono adatte allo scopo: è utile un ulteriore raffinamento
 - dalle specifiche, la macchina da sintetizzare è una macchina di Moore: infatti le uscite devono mantenere il loro valore stabile nell'intervallo tra due impulsi di sincronismo

 - Riscrittura delle specifiche
 - **due uscite** G e M, l'uscita vale 0 se semaforo rosso, 1 se semaforo verde
 - **ingresso primario**: P, 1 se premuto, 0 altrimenti

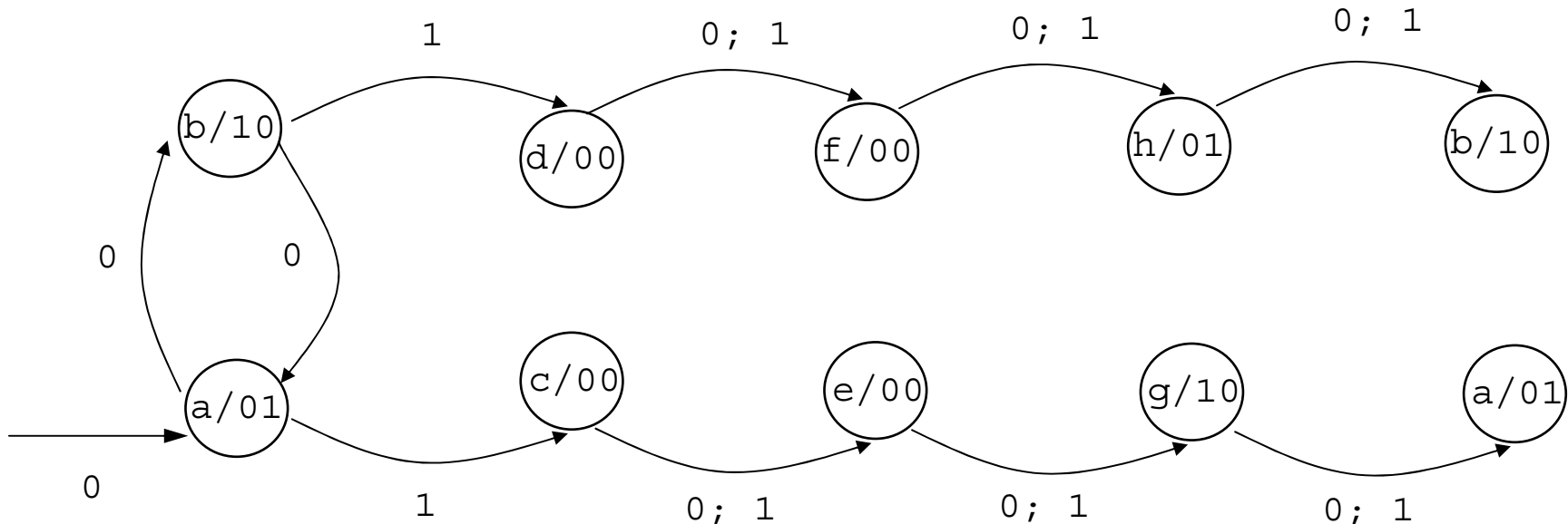
 - Una macchina sequenziale sincrona ha un ingresso P e due uscite G e M. Se $P=0$, le due uscite si alternano a 1 ad ogni impulso di sincronismo. Se $P=1$, le due uscite vanno a 0 per due impulsi di sincronismo. Successivamente, ritornano ad alternarsi con un 1 su quella che precedentemente era 0. Solo dopo una nuova alternanza, P viene preso in considerazione
-



Diagramma degli stati - Esempio 2: stato iniziale

Scelta dello **stato iniziale** per la stesura del diagramma degli stati

- stato in cui non è richiesto attraversamento pedonale: ad esempio, stato con uscite 01 e **ingresso 0**





Sintesi: Esempio 3

- Si sintetizzi la funzione di stato prossimo della seguente FSM nell'ipotesi di utilizzare bistabili di tipo SR

Tabella degli stati

	00	01	11	10	Z
S0	S0	S0	S2	S1	1
S1	S1	S1	S0	S1	0
S2	S2	S3	S0	S2	1
S3	S3	S3	S2	S3	0

Codifica

S0	00
S1	01
S2	11
S3	10

Tabella delle transizioni

	00	01	11	10	Z
00	00	00	11	01	1
01	01	01	00	01	0
11	11	10	00	11	1
10	10	10	11	10	0



Sintesi: Esempio 3

Tabella delle eccitazioni di un bistabile di tipo SR

Q	Q'	SR
0	0	0-
0	1	10
1	0	01
1	1	-0

Tabella delle transizioni

	00	01	11	10
00	00	00	11	01
01	01	01	00	01
11	11	10	00	11
10	10	10	11	10

Tabella delle eccitazioni (con SR)

		$I_1 I_0$			
Q_1	Q_0	00	01	11	10
00	0- 0-	0- 0-	10 10	0- 10	
01	0- -0	0- -1	0- 0-	0- -0	
11	-0 -0	-0 01	01 01	-0 -0	
10	-0 0-	-0 0-	-0 10	-0 0-	



Sintesi: Esempio 3

- Le quattro mappe di Karnaugh che si ottengono sono quindi:

Q ₁ Q ₀		I ₁ I ₀			
		00	01	11	10
00	0	0	1	1	
01	-	-	0	-	
11	-	0	0	-	
10	0	0	1	0	

$$\text{Set}_0 = Q_1' Q_0' I_1 + I_1 I_0 Q_0' = Q_1' Q_0' I_1 + \text{Set}_1$$

Q ₁ Q ₀		I ₁ I ₀			
		00	01	11	10
00	-	-	0	0	
01	0	1	-	0	
11	0	1	1	0	
10	-	-	0	-	

$$\text{Res}_0 = Q_0 I_0$$

Q ₁ Q ₀		I ₁ I ₀			
		00	01	11	10
00	0	0	1	0	
01	0	0	0	0	
11	-	-	0	-	
10	-	-	-	-	

$$\text{Set}_1 = I_1 I_0 Q_0'$$

Q ₁ Q ₀		I ₁ I ₀			
		00	01	11	10
00	-	-	0	-	
01	-	-	-	-	
11	0	0	1	0	
10	0	0	0	0	

$$\text{Res}_1 = I_1 I_0 Q_0 = I_1 \text{Res}_0$$



Appendice

Trasformazioni tra bistabili



Sintesi del bistabile JK tramite altri bistabili

- Specifica: realizzare una FSM sincrona con due ingressi ed una uscita che abbia il seguente comportamento (è il bistabile JK):
 - Configurazione di ingressi 00: l'uscita non cambia valore
 - Configurazione di ingressi 01: l'uscita assume valore 0
 - Configurazione di ingressi 10: l'uscita assume valore 1
 - Configurazione di ingressi 11: l'uscita cambia valore dopo ogni fronte di salita del clock

Diagramma degli stati

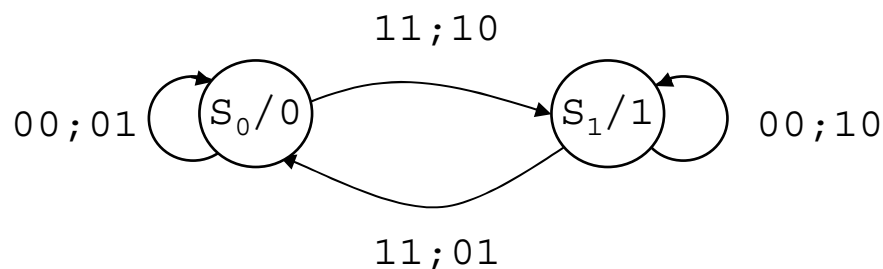


Tabella degli stati

	00	01	11	10	U
S ₀	S ₀	S ₀	S ₁	S ₁	0
S ₁	S ₁	S ₀	S ₀	S ₁	1



Codifica degli stati

Tabella degli stati

	00	01	11	10	U
S_0	S_0	S_0	S_1	S_1	0
S_1	S_1	S_0	S_0	S_1	1

□ Codifica Naturale

$S_0=0; S_1=1$

oppure

$S_0=1; S_1=0$

Tabella delle transizioni

Q \ $i_H i_L$	00	01	11	10	U
0	0	0	1	1	0
1	1	0	0	1	1

Q \ $i_H i_L$	00	01	11	10	U
1	1	1	0	0	0
0	0	1	1	0	1



Tabelle delle transizioni e eccitazioni dei bistabili

□ Tabelle delle Transizioni:

C	S	R	Q*
0	-	-	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	-

C	J	K	Q*
0	-	-	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	Q'

C	D	Q*
0	-	Q
1	0	0
1	1	1

C	T	Q*
0	-	Q
1	0	Q
1	1	Q'

□ Tabelle delle Eccitazioni:

Q	Q*	C	S	R
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	0
1	0	1	0	1
1	1	1	-	0

Q	Q*	C	J	K
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	-
1	0	1	-	1
1	1	1	-	0

Q	Q*	C	D
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Q	Q*	C	T
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0



Scelta del Bistabile - FF SR

- Codifica Naturale ($S_0=0$; $S_1=1$)

Tabella delle transizioni della FSM

	$i_H i_L$	00	01	11	10	U
Q	0	0	0	1	1	0
	1	1	0	0	1	1

Tabella delle eccitazioni della FSM (con FF SR)

	$i_H i_L$	00	01	11	10	U
Q	0	0-	0-	10	10	0
	1	-0	01	01	-0	1

Tabella delle eccitazioni FF SR

Q	Q*	C	S	R
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	0
1	0	1	0	1
1	1	1	-	0



Realizzazione delle reti combinatorie (FF SR)

- Codifica Naturale ($S_0=0$; $S_1=1$) cont.

Tabella delle eccitazioni della FSM (con FF SR)

Q \ $i_H i_L$	00	01	11	10	U
0	0 -	0 -	1 0	1 0	0
1	- 0	0 1	0 1	- 0	1

Nota: L'uscita corrisponde al bit di stato. La funzione d'uscita è $U=Q$.

Mappa di Karnaugh S

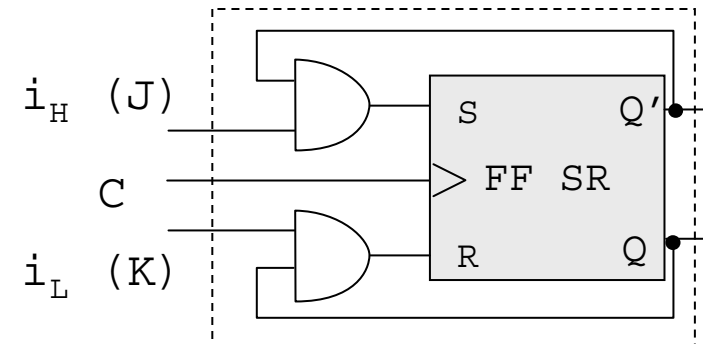
Q \ $i_H i_L$	00	01	11	10
0	0	0	1	1
1	-	0	0	-

$$S = !Q * i_H$$

Mappa di Karnaugh R

Q \ $i_H i_L$	00	01	11	10
0	-	-	0	0
1	0	1	1	0

$$R = Q * i_L$$





Scelta del Bistabile - FF JK

- Codifica Naturale ($S_0=0$; $S_1=1$)

Tabella delle transizioni della FSM

Q	$i_H i_L$				U
	00	01	11	10	
0	0	0	1	1	0
1	1	0	0	1	1

Tabella delle eccitazioni della FSM (con FF JK)

Q	$i_H i_L$				U
	00	01	11	10	
0	0-	0-	1-	1-	0
1	-0	-1	-1	-0	1

Tabella delle eccitazioni FF JK

Q	Q*	C	J	K
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	-
1	0	1	-	1
1	1	1	-	0



Realizzazione delle reti combinatorie (FF JK)

- Codifica Naturale ($S_0=0$; $S_1=1$) cont.

Tabella delle eccitazioni della FSM (con FF JK)

Q	$i_H i_L$	00	01	11	10	U
0		0 -	0 -	1 -	1 -	0
1		- 0	- 1	- 1	- 0	1

Nota: L'uscita corrisponde al bit di stato. La funzione d'uscita è $U=Q$.



Mappa di Karnaugh J

Q	$i_H i_L$	00	01	11	10
0		0	0	1	1
1		-	-	-	-

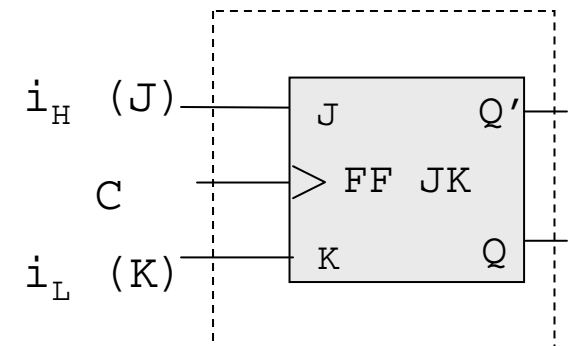
$$J = i_H$$



Mappa di Karnaugh K

Q	$i_H i_L$	00	01	11	10
0		-	-	-	-
1		0	1	1	0

$$K = i_L$$





Scelta del Bistabile - FF D

- Codifica Naturale ($S_0=0$; $S_1=1$)

Tabella delle transizioni della FSM

Q	$i_H i_L$	00	01	11	10	U
0	0	0	0	1	1	0
1	1	1	0	0	1	1

Tabella delle eccitazioni della FSM
(con FF D)

Q	$i_H i_L$	00	01	11	10	U
0	0	0	0	1	1	0
1	1	1	0	0	1	1

Tabella delle eccitazioni FF D

Q	Q*	C	D
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Nota: Nel caso si faccia uso di FF D, la tabella delle eccitazioni coincide esattamente con la tabella delle transizioni. Questo aspetto rende più semplice prevedere l'effetto della codifica sulla realizzazione e ben si presta per l'applicazione di euristiche.



Realizzazione delle reti combinatorie (FF D)

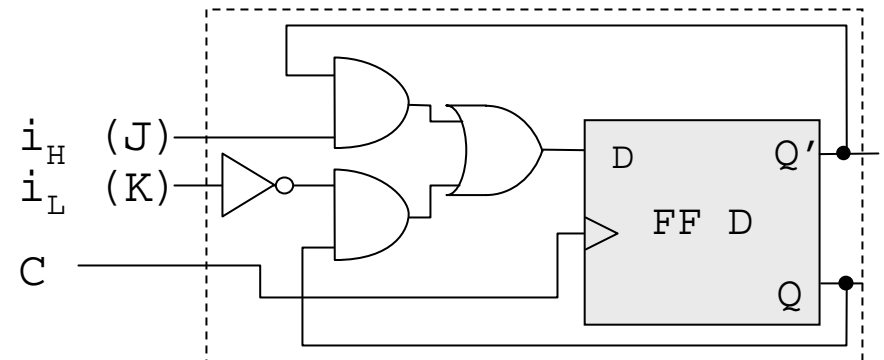
- Codifica Naturale ($S_0=0$; $S_1=1$) cont.

Tabella delle eccitazioni della FSM (con FF D)

Mappa di Karnaugh di D

Q	$i_H i_L$		U			
	00	01	11	10		
0	0	0	1	1	0	
1	1	0	0	1	1	

$$D = !Q * i_H + Q * !i_L$$





Trasformazioni tra bistabili

- L'esempio svolto è relativo alla trasformazione di un elemento di memoria in un altro.
 - Ottenere un JK utilizzando SR, JK, D e T.
- Le trasformazioni tra bistabili sono molto utili quando è richiesto uno specifico elemento di memoria ma non è disponibile
- Un modo differente per raggiungere lo stesso obiettivo è quello di utilizzare le equazioni caratteristiche

Flip Flop SR

$$Q^* = C' Q + C (S + R' Q)$$

Flip Flop JK

$$Q^* = C' Q + C (JQ' + K' Q)$$

Flip Flop D

$$Q^* = C' Q + CD$$

Flip Flop T

$$Q^* = C' Q + C (TQ' + T' Q)$$



Trasformazioni

□ T utilizzando un SR

Flip Flop SR

$$Q^* = C'Q + C(S + R'Q)$$

Flip Flop T

$$Q^* = C'Q + C(TQ' + T'Q)$$

$$Q=0: S + R'0 = T1 + T'0 \rightarrow S = T$$

$$Q=1: S + R'1 = T0 + T'1 \rightarrow S + R' = T' \rightarrow S'R = T$$

Quindi quando $T=1$ deve essere $S=0$ e $R=1$ mentre quando $T=0$ deve essere $S=1$; $R=0$ oppure $S=0$; $R=1$ ma non $S=1$; $R=0$



$$S = TQ' + T'Q$$

$$R = TQ$$

□ SR utilizzando un T

Flip Flop T

$$Q^* = C'Q + C(TQ' + T'Q)$$

Flip Flop SR

$$Q^* = C'Q + C(S + R'Q)$$

$$Q=0: T1 + T'0 = S + R'0 \rightarrow T = S$$

$$Q=1: T0 + T'1 = S + R'1 \rightarrow T' = S + R' \rightarrow T = S'R$$



$$T = SQ' + (S'R)Q$$