

# Realizzazione di FSM sincrone

M. Favalli

Engineering Department in Ferrara



## Sommario

- 1 Realizzazione (sincrona) del STG
- 2 Il problema della codifica dello stato
- 3 Temporizzazioni

## Sommario

- 1 Realizzazione (sincrona) del STG
- 2 Il problema della codifica dello stato
- 3 Temporizzazioni

## Introduzione

- Una volta ottenuto l'automa minimo o comunque un automa soddisfacente si passa alla realizzazione dell'automa come rete sequenziale sincrona
- I simboli di ingresso ( $\mathcal{X}$ ), di uscita ( $\mathcal{Z}$ ) e gli stati ( $\mathcal{S}$ ) devono essere codificati con un codice binario
- Le funzioni  $\lambda$  e  $\delta$  diventano funzioni dell'algebra di commutazione ( $\Lambda$  e  $\Delta$ ) che vengono poi realizzate tramite reti combinatorie
- La memoria in retroazione che garantisce la trasformazione dello stato futuro in stato presente a ogni istante di sincronizzazione viene realizzata tramite flip-flop o latch controllati dal segnale di clock

## Obbiettivi della codifica

- Costo (costo della rete combinatoria e dei flip-flop o latch)
- Prestazioni
- Consumo di potenza
- Affidabilità

Questi obiettivi possono essere raggiunti mediante opportune tecniche euristiche

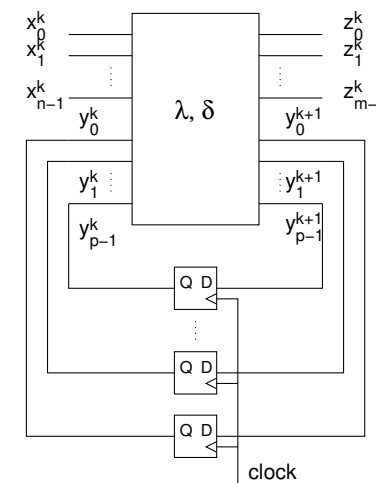
## Codifica

- Codifica dei simboli di ingresso:  $f_{in} : \mathcal{X} \rightarrow \{0, 1\}^n$ ,  $2^n \geq |\mathcal{X}|$
- Codifica dei simboli di uscita:  $f_{out} : \mathcal{Z} \rightarrow \{0, 1\}^m$ ,  $2^m \geq |\mathcal{Z}|$
- Codifica degli stati:  $f_{state} : \mathcal{S} \rightarrow \{0, 1\}^p$ ,  $2^p \geq |\mathcal{S}|$
- Variabili di ingresso:  $\{x_0, x_1, \dots, x_{n-1}\}$
- Variabili di uscita:  $\{z_0, z_1, \dots, z_{m-1}\}$
- Variabili di stato:  $\{y_0, y_1, \dots, y_{p-1}\}$

## Ipotesi

- Supponiamo che la codifica binaria dei simboli di ingresso e di uscita sia assegnata
- Per motivi di tempo supponiamo di utilizzare una codifica casuale per lo stato (si rimanda al testo per una trattazione del problema della scelta di un codice binario che riduca il costo della rete)
- Per la retroazione si utilizzano flip-flop di tipo D

## Modello di riferimento (Huffman)



## Rete sequenziale sincrona

- Sostituendo i simboli di ingresso, uscita e stato che compaiono nella tabella di transizione dello stato con le rispettive codifiche si ottiene la tabella delle transizioni codificata
- Tale tabella costituisce la tabella di verità per le funzioni  $\Lambda$  e  $\Delta$  (si considera un automa di Mealy):

- $\lambda \Rightarrow \Lambda : \{0, 1\}^{n+p} \rightarrow \{0, 1\}^m$ , che per la  $i$ -ma variabile di uscita all'istante  $k$  fornisce:

$$z_i^k = \Lambda_i(x_0^k, x_1^k, \dots, x_{n-1}^k, y_0^k, y_1^k, \dots, y_{p-1}^k)$$

- $\delta \Rightarrow \Delta : \{0, 1\}^{n+p} \rightarrow \{0, 1\}^p$ , che per l' $i$ -ma variabile di stato futuro all'istante  $k$  fornisce:

$$y_i^{k+1} = \Delta_i(x_0^k, x_1^k, \dots, x_{n-1}^k, y_0^k, y_1^k, \dots, y_{p-1}^k)$$

## Sommario

- 1 Realizzazione (sincrona) del STG
- 2 Il problema della codifica dello stato
- 3 Temporizzazioni

## Retroazione

- Lo stato corrente é memorizzato all'interno dei FF messi in retroazione
- Le uscite della rete combinatoria dipendono dal tipo di FF usati (ne esistono di diverso tipo oltre ai D)
- Nel caso di FF di tipo D, l'equazione caratteristica é:  $Q^{k+1} = D^k$  e quindi  $\Delta$  fornisce direttamente gli ingressi dei FF

## Codifica casuale

- Se si fa l'ipotesi che il costo della rete sia proporzionale al numero di variabili di stato (questa ipotesi in diversi casi non é verificata), conviene usarne il minimo numero:  $p = \lceil \log_2 |S| \rceil$
- Nella codifica casuale si può semplicemente assegnare una configurazione binaria scelta a caso a ciascuno degli stati
- Esistono codifiche ottimizzate e codifiche in cui si usa un numero di variabili di stato maggiore di quello minimo
- Un esempio é dato dalla codifica 1-out-of-n in cui si usa un FF per ogni stato

## Esempio - I

$s^k$	$(x_1 x_0)^k$			
	00	01	11	10
A	A, 0	B, 1	C, 0	A, 0
B	B, 0	D, 0	C, 0	B, 1
C	C, 0	A, 1	C, 0	D, 0
D	A, 0	B, 1	B, 0	D, 0

Tabella di transizione dello stato

$(y_1 y_0)^k$	$(x_1 x_0)^k$			
	00	01	11	10
00	00, 0	01, 1	11, 0	00, 0
01	01, 0	10, 0	11, 0	01, 1
11	11, 0	00, 1	11, 0	10, 0
10	00, 0	01, 1	01, 0	10, 0

Tabella delle transizioni

stato	$y_1 y_0$
A	00
B	01
C	11
D	10

Codifica

## Esempio - III

Le funzioni di eccitazione e di uscita possono essere sintetizzate utilizzando uno dei metodi noti per la sintesi di reti combinatorie

Ad esempio considerando forme normali di tipo SP di costo minimo:

$$y_1^{k+1} = (x_1 x_0 y_1' + y_1' y_0 x_0 + x_1 x_0 y_0 + y_1 y_0 x_0' + y_1 x_1 x_0')^k$$

$$y_0^{k+1} = (x_1 x_0 + y_0' x_0 + y_0 x_1' x_0' + y_1' y_0 x_0')^k$$

$$z_0^k = (y_0' x_1' x_0 + y_1 x_1' x_0 + y_1' y_0 x_1 x_0')^k$$

## Esempio - II

Dalla tabella delle transizioni si ottengono quindi le funzioni eccitazione per le 2 variabili di stato e la funzione per l'uscita

$(y_1 y_0)^k$	$(x_1 x_0)^k$				$y_1^{k+1}$
	00	01	11	10	
00	0	0	1	0	
01	0	1	1	0	
11	1	0	1	1	
10	0	0	0	1	

$(y_1 y_0)^k$	$(x_1 x_0)^k$				$y_0^{k+1}$
	00	01	11	10	
00	0	1	1	0	
01	1	0	1	1	
11	1	0	1	0	
10	0	1	1	0	

$(y_1 y_0)^k$	$(x_1 x_0)^k$				$z_0^k$
	00	01	11	10	
00	0	1	0	0	
01	0	0	0	1	
11	0	1	0	0	
10	0	1	0	0	

## Ottimizzazione della codifica

- Se nell'esempio precedente si fosse scelta una codifica diversa, probabilmente si sarebbe ottenuta una rete combinatoria con un costo diverso
- Nasce quindi il problema di selezionare la codifica che da luogo alla rete con il costo minore
- Dato il numero molto grande di possibili codifiche, esistono delle tecniche euristiche per la scelta della codifica (che non verranno esaminate)

## Problemi

- Se  $2^p > |\mathcal{S}|$  si avranno alcuni stati che esistono nell'automa codificato, ma non in quello di partenza
- Nella tabella di transizione dello stato saranno presenti delle indifferenze su stato futuro
- Tali indifferenze possono poi essere assegnate durante il processo di sintesi
- L'automa risultante risulterà diverso da quello di partenza
- É necessaria un operazione di analisi per capire se esistono problemi (stati irraggiungibili, componenti non connesse)

## Esempi di problemi dovuti alla codifica dello stato

## Analisi di una rete sequenziale sincrona

- Passaggio da una descrizione livello gate al STG
- Nel caso di reti molto semplici il problema può essere risolto per via grafica
  - 1 determinare le espressioni SP di uscite e variabili di stato futuro
  - 2 tracciare tali espressioni su mappe di Karnaugh
  - 3 riunire tali mappe in unica mappa che rappresenta la tabella delle transizioni codificata
  - 4 associare un nome simbolico a ciascuno stato e ottenere quindi la tabella di transizione dello stato
  - 5 disegnare il STG

## Sommario

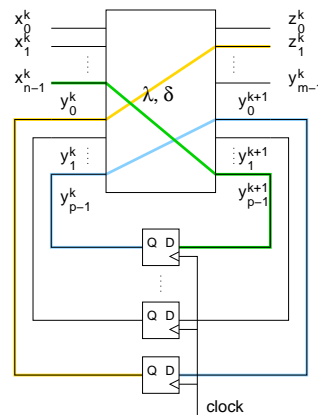
- 1 Realizzazione (sincrona) del STG
- 2 Il problema della codifica dello stato
- 3 Temporizzazioni

## Condizioni per il campionamento

- Una rete sequenziale (Huffman) si comporta come sincrona e in maniera aderente alle specifiche se le uscite della rete combinatoria vengono campionate quando tutti i transistori si sono esauriti
- Questo comportamento deve essere garantito dalla scelta di un valore opportuno del periodo di clock ( $T$ ) che dipende dal ritardo della rete combinatoria e dai parametri dei FF
- Si utilizza l'ipotesi che il cambiamento degli ingressi sia sincrono con il periodo di clock e che le uscite siano anch'esse campionate dal segnale di clock

## Ritardo massimo della rete combinatoria

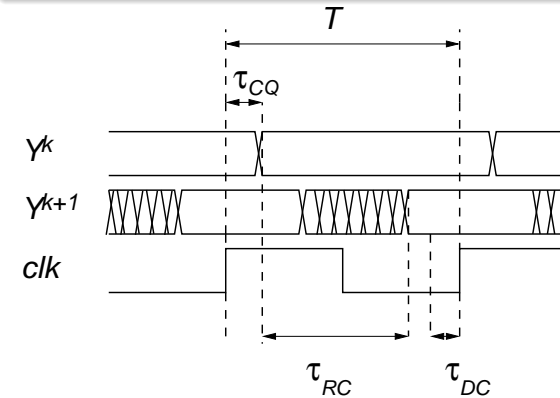
- Per determinare  $\tau_{RC}$  si devono considerare i diversi possibili cammini per la propagazione di una transizione da un ingresso della RC ( $x_i^k$  o  $y_i^k$ ) a una sua uscita ( $z_i^k$  o  $y_i^{k+1}$ )
- Questi cammini si possono attivare dipendentemente dalle configurazioni in ingresso alla RC
- $\tau_{RC}$  deve essere determinato in condizioni di caso peggiore



## Scelta del periodo di clock

$T$  minimo

$$T > \tau_{CQ} + \tau_{RC} + \tau_{DC}$$



$\tau_{RC}$  é il ritardo massimo della rete combinatoria  
 $\tau_{CQ}$  e  $\tau_{DC}$  sono il tempo di risposta e di setup dei FF

## Sommario

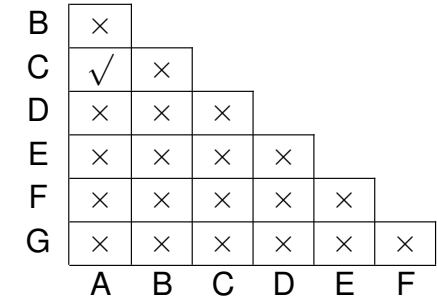
### Passi nella sintesi di una FSM

- 1 Interpretazione delle specifiche
- 2 Progetto del STG
- 3 Ottimizzazione del numero degli stati
- 4 Codifica dello stato (con eventuali euristici per la scelta del codice)
- 5 Sintesi della rete combinatoria che realizza  $\lambda$  e  $\delta$
- 6 Calcolo della frequenza di clock

# Esempio completo

Si realizzi una rete sequenziale sincrona con 2 ingressi ( $x, y$ ) e due uscite ( $w_1, w_0$ ). Quando  $y = 1$  compito della rete é contare modulo 3 il numero di transizioni da 0 a 1 ricevute su  $x$  e di produrre la codifica binaria di tale conteggio in uscita, se  $y = 0$  l'uscita vale 0 (nota le transizioni di  $x$  contemporanee a quella di  $y$  non vanno contate).

state	xy			
	00	01	11	10
A	A,00	B,00	C,00	A,00
B	A,00	B,00	D,01	A,00
C	A,00	B,00	C,00	A,00
D	A,00	E,01	D,01	A,00
E	A,00	E,01	F,10	A,00
F	A,00	G,10	F,10	A,00
G	A,00	G,10	C,00	A,00



Classi massime di indistinguibilità:  $\{A, C\}$

state	xy			
	00	01	11	10
$\alpha$	$\alpha,00$	B,00	$\alpha,00$	$\alpha,00$
B	$\alpha,00$	B,00	D,01	$\alpha,00$
D	$\alpha,00$	E,01	D,01	$\alpha,00$
E	$\alpha,00$	E,01	F,10	$\alpha,00$
F	$\alpha,00$	G,10	F,10	$\alpha,00$
G	$\alpha,00$	G,10	$\alpha,00$	$\alpha,00$

state	xy			
	00	01	11	10
000	000,00	001,00	000,00	000,00
001	000,00	001,00	011,01	000,00
011	000,00	001,01	011,01	000,00
010	000,00	010,01	100,10	000,00
100	000,00	101,10	100,10	000,00
101	000,00	101,10	000,00	000,00