





Algebra booleana e Reti logiche

Reti sequenziali
(seconda parte)

Analisi e sintesi

- *Analisi*  dato lo schema logico della rete, se ne identifica il comportamento
- *Sintesi*  a partire dalla descrizione del comportamento della rete, si realizza lo schema logico

Procedimento di sintesi

- Si tracciano il diagramma di stato e la corrispondente tabella di flusso (in forma minima).
- Si codificano gli stati attraverso variabili di stato booleane:

sostituendo agli stati simbolici della tabella di flusso le rispettive codifiche si ottengono le mappe delle funzioni di uscita e di stato della rete

Procedimento di sintesi

- A partire dalle mappe delle funzioni di stato, si determinano le funzioni di eccitazione dei flip-flop usati come elementi di memoria:

in tal modo risulta determinata la parte combinatoria che porta gli ingressi primari e le variabili di stato agli ingressi dei flip-flop

- Si determina la parte combinatoria rimanente ovvero quella corrispondente alle funzioni di uscita.

Esempio di sintesi

- Sintesi di una rete sequenziale sincrona avente un ingresso x e un'uscita z:

l'uscita z assume il valore 1 quando in ingresso si presenta la sequenza 1 1 1 1
(*riconoscitore di sequenza*)



data la sequenza d'ingresso:

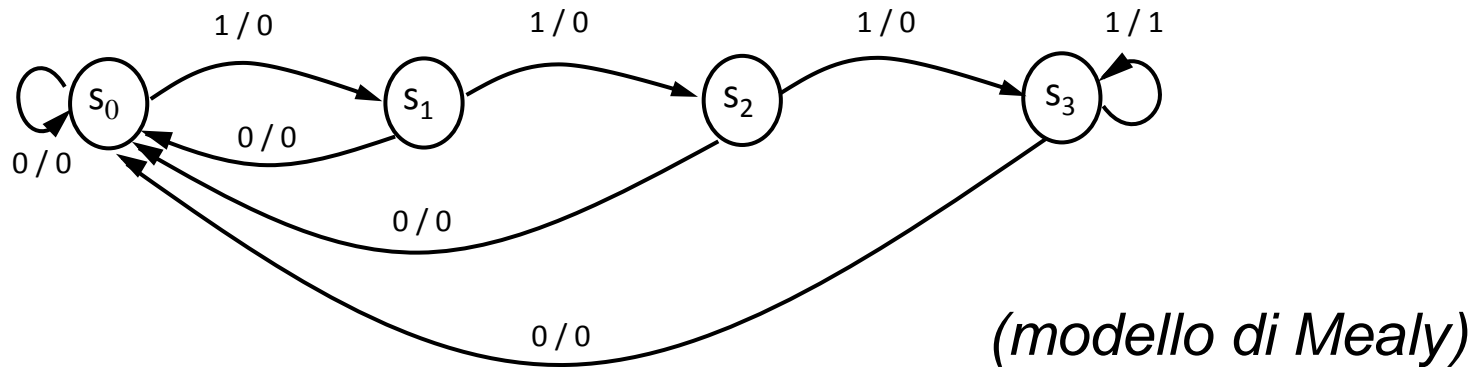
1 1 0 1 1 1 1 1 1 1 0 1 0

si avrà in uscita:

0 0 0 0 0 0 1 1 1 1 0 0 0

Esempio di sintesi

■ Diagramma di stato:



S_0 : nessuna sotto-sequenza riconosciuta

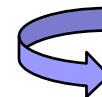
S_1 : sotto-sequenza 1 S_2 : sotto-sequenza 11

S_3 : sotto-sequenza 111

Esempio di sintesi

- Tabella di flusso:

<i>stati</i> \ <i>ingressi</i>	0	1
S_0	$S_0 / 0$	$S_1 / 0$
S_1	$S_0 / 0$	$S_2 / 0$
S_2	$S_0 / 0$	$S_3 / 0$
S_3	$S_0 / 0$	$S_3 / 1$



codifica
degli stati

Esempio di sintesi

- I quattro stati simbolici S_0 , S_1 , S_2 e S_3 possono essere codificati usando due variabili di stato y_1 e y_2 :

stato	y_1	y_2
S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1



Per ogni variabile di stato
ci occorrerà un elemento
di memoria (flip-flop)

Esempio di sintesi


- Nella tabella di flusso sostituiamo agli stati simbolici le rispettive codifiche:

$y_1 \ y_2 \backslash x$	x	
	0	1
0 0	0 0 / 0	0 1 / 0
0 1	0 0 / 0	1 0 / 0
1 0	0 0 / 0	1 1 / 0
1 1	0 0 / 0	1 1 / 1



mappe dell'uscita
e degli stati futuri

Esempio di sintesi



$y_1 \ y_2 \backslash x$	0	1
0 0	0 0 / 0	0 1 / 0
0 1	0 0 / 0	1 0 / 0
1 0	0 0 / 0	1 1 / 0
1 1	0 0 / 0	1 1 / 1



Mappa dell'uscita z:

$y_1 \ y_2 \backslash x$	0	1
0 0	0	0
0 1	0	0
1 1	0	1
1 0	0	0

$$z = xy_1y_2$$

Esempio di sintesi

y ₁ y ₂ \ x		0	1
		0	1
0 0		0 0 / 0	0 1 / 0
0 1		0 0 / 0	1 0 / 0
1 0		0 0 / 0	1 1 / 0
1 1		0 0 / 0	1 1 / 1



Mappa di y_1' :

y ₁ y ₂ \ x		0	1
		0	1
0 0		0	0
0 1		0	1
1 1		0	1
1 0		0	1

$$y_1' = xy_1 + xy_2$$

Esempio di sintesi

x \ y ₁ y ₂		x	
		0	1
0	0	0 0 / 0	0 1 / 0
0	1	0 0 / 0	1 0 / 0
1	0	0 0 / 0	1 1 / 0
1	1	0 0 / 0	1 1 / 1



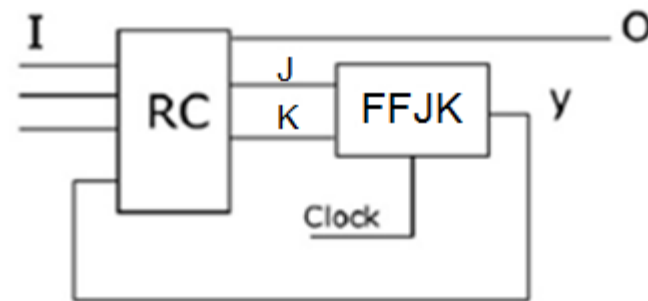
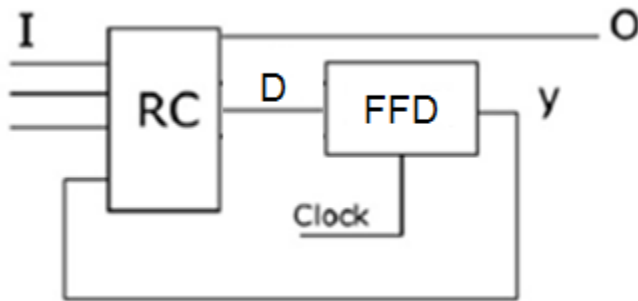
Mappa di y_2' :

y ₁ y ₂ \ x		x	
		0	1
0	0	0	1
0	1	0	0
1	1	0	1
1	0	0	1

$$y_2' = xy_1 + x\bar{y}_2$$

Esempio di sintesi

- Non ci resta ora che determinare le funzioni di eccitazione dei flip-flop:



un flip-flop per ogni
variabile di stato

Esempio di sintesi

- Usando due FF-D :

$$D_1 = xy_1 + xy_2$$

$$D_2 = xy_1 + x\bar{y}_2$$

- Usando due FF-JK :

$$J_1 = xy_2$$

$$J_2 = x$$

$$K_1 = \bar{x}$$

$$K_2 = \bar{y}_1 + \bar{x}$$

Esempio di sintesi

- Ricordiamo che le mappe di J_1 e K_1 si ricavano dalla mappa di y_1' :

x \ y ₁ y ₂		0	1
0	0	0	0
0	1	0	1
1	1	0	1
1	0	0	1

y_1'



x \ y ₁ y ₂		0	1
0	0	0	0
0	1	0	1
1	1	-	-
1	0	-	-

$J_1 = xy_2$

x \ y ₁ y ₂		0	1
0	0	-	-
0	1	-	-
1	1	1	0
1	0	1	0

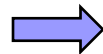
$K_1 = \bar{x}$

Esempio di sintesi

- Analogamente, le mappe di J_2 e K_2 si ricavano dalla mappa di y_2' :

		x	
		0	1
y ₁	y ₂		
0	0	0	1
0	1	0	0
1	1	0	1
1	0	0	1

$$y_2'$$

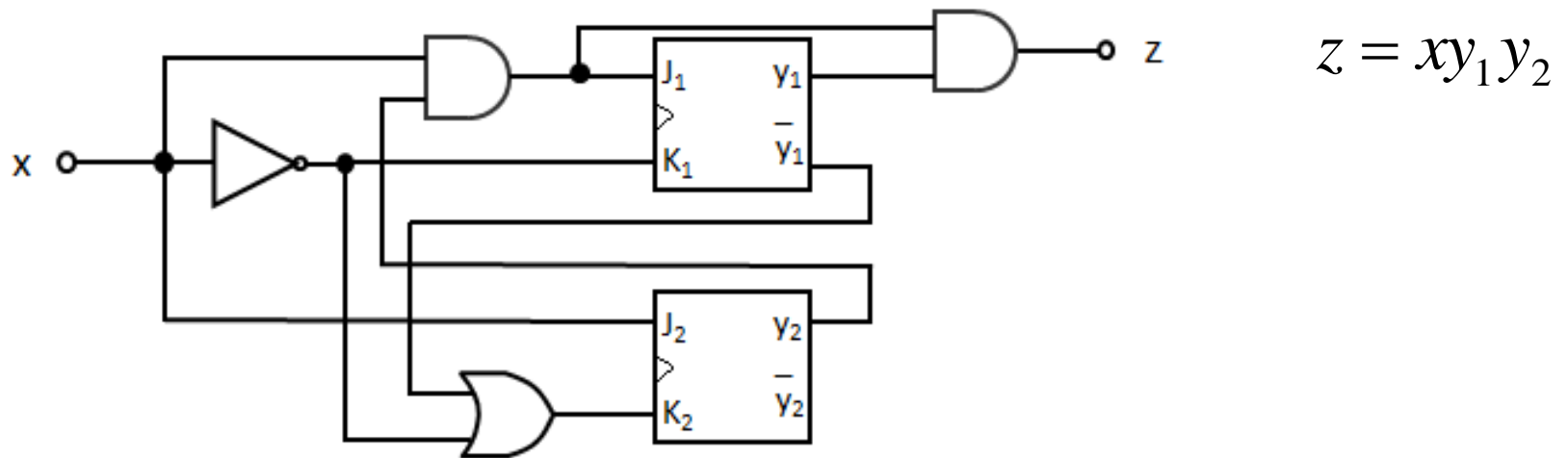


		x	
		0	1
y ₁	y ₂		
0	0	0	1
0	1	-	-
1	1	-	-
1	0	0	1

$$J_2 = x$$

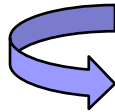
		x	
		0	1
y ₁	y ₂		
0	0	-	-
0	1	1	1
1	1	1	0
1	0	-	-

$$K_2 = \bar{y}_1 + \bar{x}$$



Ancora sugli elementi di memoria

- Abbiamo visto che un flip-flop è un circuito sequenziale molto semplice che si comporta come un elemento di memoria in grado di contenere un bit di informazione.

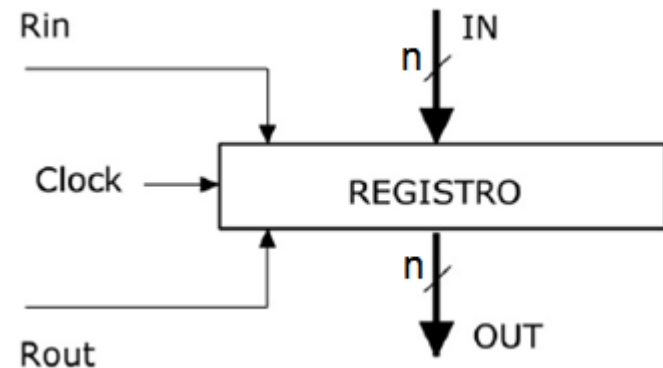
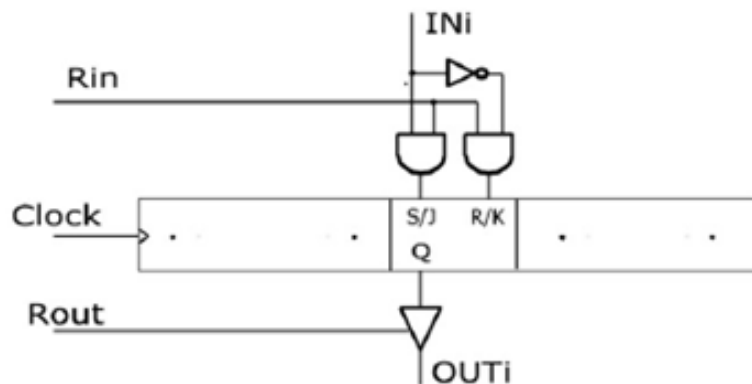


Più flip-flop possono essere organizzati in strutture denominate *registri*.

Registro

- Insieme di n elementi di memoria (flip-flop) identici, sincronizzati tramite un unico clock.
- Permette di memorizzare sequenze di n bit (parole).
- I bit delle parole possono essere registrati o letti in parallelo, in serie o in modo misto.

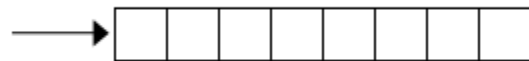
Registro in parallelo



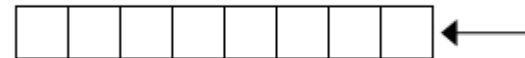
- n linee binarie di ingresso/uscita (dimensione/parallelismo del registro)
- segnali di controllo per l'abilitazione degli ingressi e delle uscite

Registro a scorrimento

- Un unico ingresso seriale che va dal bit meno significativo a quello più significativo (*left shift register*) o viceversa (*right shift register*):



right shift



left shift

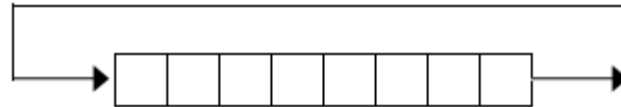
- ➡ Ad ogni impulso di clock il contenuto di ciascun flip-flop viene trasferito al successivo (sono quindi necessari n cicli di clock per memorizzare una nuova parola nel registro)

Registro a scorrimento

- Una volta caricato il registro, l'output può essere prelevato
 - in modo seriale, cioè un bit per volta dal terminale di output (registro *seriale-seriale*)
 - in modo parallelo, leggendo contemporaneamente gli n bit (registro *seriale-parallelo*)

Registro ad anello

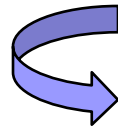
- Collegando l'output di un registro seriale-seriale con l'input, si ottiene un registro a *scorrimento circolare*:



Si può così spostare l'ordine dei bit senza distruggere l'informazione immagazzinata

Registro misto

- Esistono anche registri che, in base ai valori di alcune variabili di comando, possono accettare e rendere disponibili le informazioni sia in modo seriale che in modo parallelo.



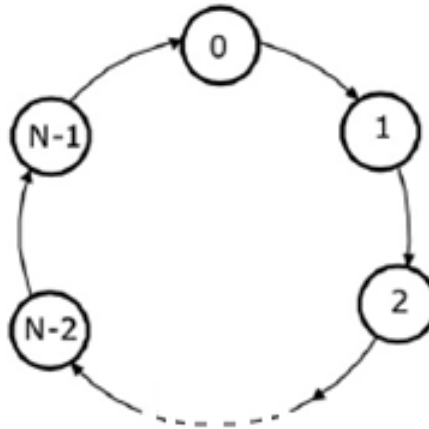
“registro universale”

Contatore

- Rete sequenziale con un solo ingresso costituito da un segnale, normalmente periodico (il clock), di cui si vogliono contare gli impulsi.
- Il numero di impulsi ricevuti è memorizzato come stato interno della rete:
 - ad ogni impulso di clock
la rete cambia stato
(avanza il conteggio)

Contatore

- Il numero N di stati interni è detto *modulo* del contatore:

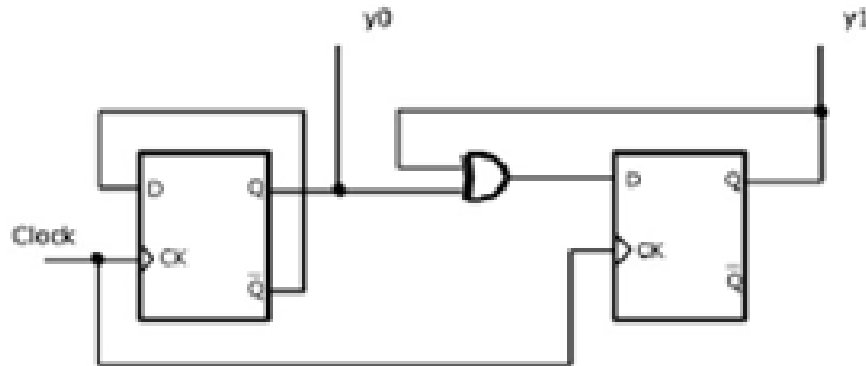


Stato presente	Stato futuro
0	1
1	2
2	3
...	...
$N - 2$	$N - 1$
$N - 1$	0

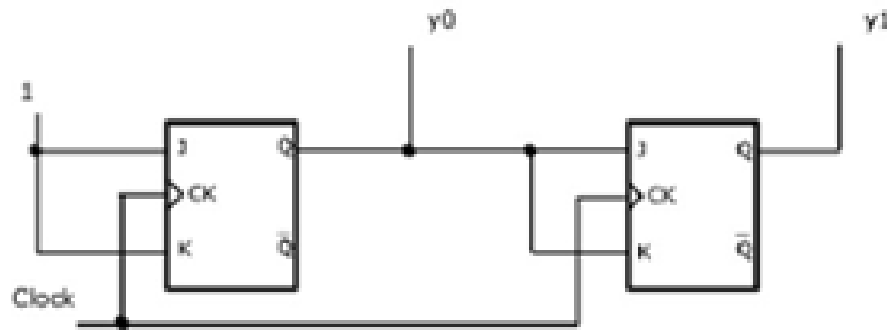


Con una rete formata da n flip-flop si possono rappresentare $N = 2^n$ stati

Esempio di contatore modulo 4



(realizzazione
con FF-D)



(realizzazione
con FF-JK)

(Figure parzialmente tratte da G. Bucci. *Calcolatori elettronici. Architettura e organizzazione*.
Copyright © 2009 - The McGraw-Hill Companies)