




**Sintesi di reti sequenziali**  
 Prof. Daniele Gorla




**Lo scopo della sintesi**

Come nel caso combinatorio, il procedimento di sintesi ha lo scopo di creare un circuito digitale partendo da una specifica astratta.

	Combinatorio	Sequenziale
<i>Specifica formale</i>	TV	Automa
<i>Rappresentazione intermedia</i>	EB	Tabella degli stati futuri e funzioni di eccitazione  EB
<i>Circuito finale</i>	Porte logiche + interconnessioni (acicliche)	Porte logiche + interconnessioni (anche cicliche) + FF

2



**Il procedimento di sintesi tramite un esempio**

Si progetti un automa che, presa in input una stringa di bit, dà in output 1 se e solo se il numero di "1" ricevuti fino a quel momento è un multiplo di 3.

*Passo 1. Dalla specifica all'automa*  
 Anzitutto, dalla specifica verbale bisogna definire l'automa e minimizzarlo.

Nel nostro esempio, un numero è multiplo di 3 se è del tipo  $3 \cdot k$ , dove  $k$  è un numero naturale (0, 3, 6, 9, 12, ...). L'automa richiesto deve quindi vedere se, detto  $n$  il numero di "1" ricevuti fino a quel momento, si ha:

$$n = 3 \cdot k \qquad n = 3 \cdot k + 1 \qquad n = 3 \cdot k + 2$$


e solo nel primo caso deve dare in output 1.

Associamo quindi la prima condizione allo stato  $q_0$ , la seconda allo stato  $q_1$  e la terza allo stato  $q_2$ . Lo stato iniziale è  $q_0$ .

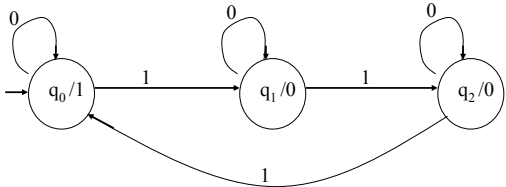
C'è una transizione da  $q_i$  a  $q_{(i+1) \text{ MOD } 3}$  ogni volta che arriva un "1"; con "0" resto nello stato corrente.

$q_0$  è l'unico stato ad emettere 1 (Moore).

3



**Automa dell'esempio**



O, equivalentemente (sarà utile tra un po'):

	0	1
$q_0$	$q_0 / 1$	$q_1 / 0$
$q_1$	$q_1 / 0$	$q_2 / 0$
$q_2$	$q_2 / 0$	$q_0 / 1$

Si vede banalmente che l'automa è già minimo (tutti gli stati sono distinguibili)

4

## 2. Codifica in binario dell'automa



Bisogna rappresentare i 3 insiemi  $Q, \Sigma$  e  $\Delta$  in binario.

N.B.: un insieme con  $n$  elementi richiede  $\lceil \log_2 n \rceil$  bit per essere rappresentato.

- ogni bit necessario per rappresentare lo stato è memorizzato in un FF (e quindi è associato all'uscita  $y$  di un FF);
- ogni bit necessario per rappresentare l'input è un ingresso del circuito (e quindi è associato ad una variabile di input  $x$ );
- ogni bit necessario per rappresentare l'output è una uscita del circuito (e quindi è associato ad una variabile di output  $z$ ).

Nel nostro esempio, l'alfabeto di input e di output è già codificato in binario (è proprio l'insieme  $\{0,1\}$ ).

Codifichiamo lo stato  $q_0$  con la configurazione  $y_1 y_0 = 00$  dei FF,  $q_1$  con  $y_1 y_0 = 01$  e  $q_2$  con  $y_1 y_0 = 10$  (la configurazione  $y_1 y_0 = 11$  non è usata).

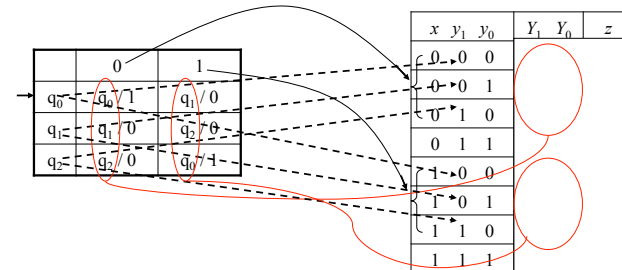
5

## 3. Tabella degli stati futuri



Passiamo dall'automa a una tabella (chiamata *degli stati futuri*) in cui esprimiamo stato futuro (cioè, all'istante  $t+1$ ) e output (all'istante  $t$ ) in funzione dello stato corrente e dell'input (all'istante  $t$ ).

Questo può esser fatto in maniera molto semplice partendo dalla rappresentazione tabellare dell'automa (ma anche dal disegno...)



6

## 4. Funzioni di eccitazione dei FF



Alla tabella degli stati futuri, aggiungiamo una colonna per ogni entrata di ogni FF necessario per realizzare il circuito (uno per ogni  $y_i$ ).

Riempiamo ogni colonna usando le funzioni di eccitazione dei FF, in base allo stato corrente ( $y$ ) e stato futuro ( $Y$ ).

Il tipo dei FF può essere o specificato nel progetto o sennò si può scegliere quello che genera il circuito finale più semplice.

$x$	$y_1$	$y_0$	$Y_1$	$Y_0$	$z$	$s_1$	$r_1$	$s_0$	$r_0$	$j_1$	$k_1$	$j_0$	$k_0$	$d_1$	$d_0$	$t_1$	$t_0$
0	0	0	0	0	1												
0	0	1	0	1	0												
0	1	0	1	0	0												
0	1	1	-	-	-												
1	0	0	0	1	0												
1	0	1	1	0	0												
1	1	0	0	0	1												
1	1	1	-	-	-												

7

## 5. Espressioni booleane minime



Da tale tabella bisogna ricavare le EB (minime) per gli ingressi dei FF e per le uscite del circuito

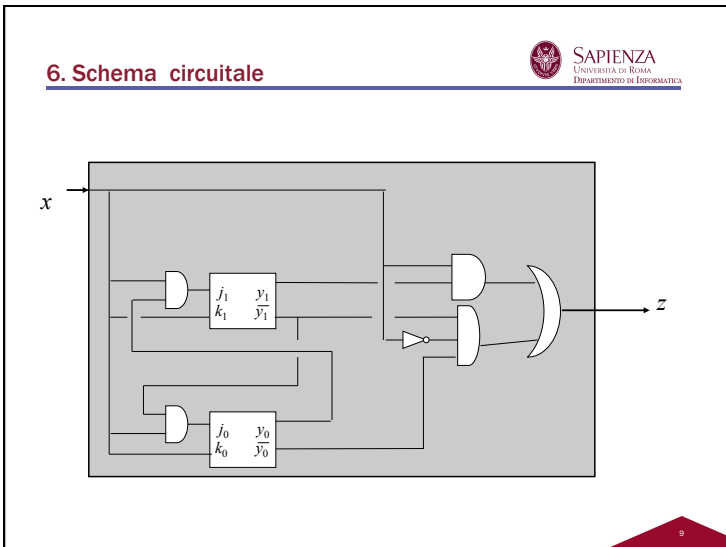
N.B.: per gli stati futuri ( $Y$ ) non bisogna calcolare una EB visto che essi vengono calcolati dal FF in base allo stato corrente (memorizzato nel FF) e all'input (di cui andiamo a calcolare la EB).

In base alle EB minime per le entrate dei FF si sceglierà il modello di FF da usare per ogni bit da memorizzare (non è necessario che i FF siano tutti dello stesso tipo!), se tale modello non era specificato tra le specifiche del problema

$$z = \bar{x}y_1\bar{y}_0 + xy_1$$

$s_1 = xy_0$	$j_1 = xy_0$	$d_1 = \bar{x}y_1 + xy_0$	$t_1 = x(y_1 + y_0)$
$r_1 = xy_1$	$k_1 = x$	$d_0 = \bar{x}y_0 + xy_1\bar{y}_0$	$t_0 = xy_1$
$s_0 = x\bar{y}_1\bar{y}_0$	$j_0 = x\bar{y}_1$	$\underbrace{\hspace{10em}}_{8 \text{ porte}}$	$\underbrace{\hspace{10em}}_{3 \text{ porte}}$
$r_0 = xy_0$	$k_0 = x$		
$\underbrace{\hspace{10em}}_{4 \text{ porte}}$	$\underbrace{\hspace{10em}}_{2 \text{ porte}}$		

8



### Codifica e ottimizzazioni

OSS.: Al passo 2 abbiamo dovuto codificare stati e alfabeti in binario; tale codifica ha un impatto sulla dimensione finale del circuito!!

Es.: Avevamo usato la codifica per gli stati  $q_0 \rightarrow 00, q_1 \rightarrow 01, q_2 \rightarrow 10$ .

Proviamo a vedere cosa succede con la codifica  $q_0 \rightarrow 11, q_1 \rightarrow 01, q_2 \rightarrow 10$ . Per semplicità, ignoro l'output (che non è influenzato dalla codifica degli stati) e considero solo FF di tipo D:

x	y <sub>1</sub>	y <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>	d <sub>1</sub>	d <sub>0</sub>
0	0	0	-	-	-	-
0	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	1	1	1	1
1	0	0	-	-	-	-
1	0	1	1	0	1	0
1	1	0	1	1	1	1
1	1	1	0	1	0	1

$d_1 = \overline{y_0} + \overline{x}y_1 + xy_1 = \overline{y_0} + (x \oplus y_1)$   
 $d_0 = \overline{x}y_0 + xy_1$

6 porte  
(contro le 8 della codifica precedente)

### Rete sequenziale ottima

Per ottenere una rete sequenziale ottima (nel senso che ha il minor numero di porte logiche possibile) bisogna:

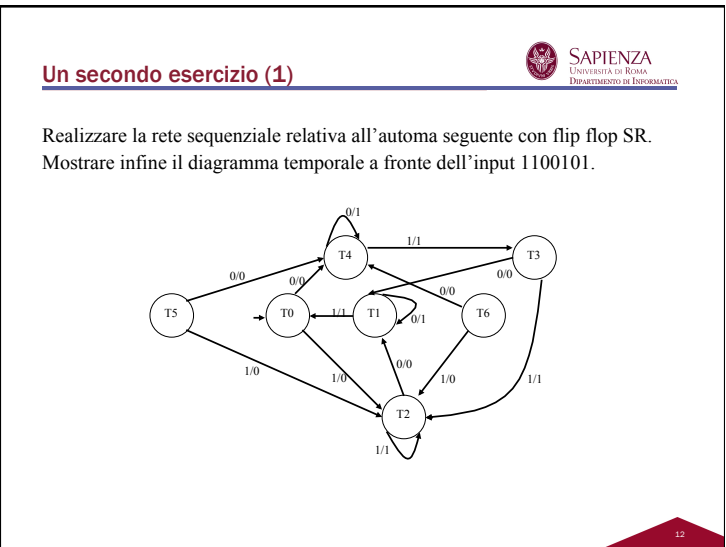
1. utilizzare l'automa minimo;
2. considerare tutte le possibili codifiche binarie dell'insieme di stati e degli alfabeti di input e output;
3. per ogni possibile codifica del punto 2, considerare tutti i possibili tipi di FF.

Chiaramente, fare questo è pesantissimo già per automi piccolissimi:

Nel nostro esempio, ho 24 possibili codifiche degli stati e 4 tipi di FF: totale = 96 colonne per le entrate dei FF; avendo 2 FF, 2 con 2 ingressi e 2 con 1 ingresso, dovrei calcolare  $96 \times 12 = 1152$  EB!!!

È impraticabile in casi reali!!

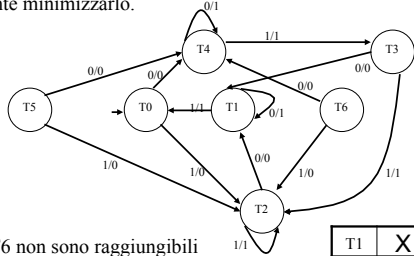
→ per i costi e i tempi di attraversamento odierni, non è quasi mai necessario avere il circuito migliore in assoluto



**Un secondo esercizio (2)**



Passo 1: l'automata è già dato; bisogna solo vedere se è minimo ed eventualmente minimizzarlo.



OSS: T5 e T6 non sono raggiungibili

Automa minimo:

	0	1
<b>S0</b> (T0)	S3/0	S2/0
<b>S1</b> (T1)	S1/1	S0/1
<b>S2</b> (T2+T3)	S1/0	S2/1
<b>S3</b> (T4)	S3/1	S2/1

T1	X			
T2	X	X		
T3	X	X	O	
T4	X	X	X	X
T0		T1	T2	T3

13

**Un secondo esercizio (3)**



Codifichiamo gli stati nel modo seguente:

S0 → 00 , S1 → 01 , S2 → 10 , S3 → 11

Con tale codifica, la tabella degli stati futuri è:

	0	1
<b>S0</b> (T0)	S3/0	S2/0
<b>S1</b> (T1)	S1/1	S0/1
<b>S2</b> (T2+T3)	S1/0	S2/1
<b>S3</b> (T4)	S3/1	S2/1

Considerando i valori dello stato agli istanti t e t+1, calcoliamo le EB da dare in input ai FF SR, usando la funzione di eccitazione di tali FF:

x	y1	y0	Y1	Y0	z	s1	r1	s0	r0
0	0	0	1	1	0	1	0	1	0
0	0	1	0	1	1	0	-	-	0
0	1	0	0	1	0	0	1	1	0
0	1	1	1	1	1	-	0	-	0
1	0	0	1	0	0	1	0	0	-
1	0	1	0	0	1	0	-	0	1
1	1	0	1	0	1	-	0	0	-
1	1	1	1	0	1	-	0	0	1

14

**Un secondo esercizio (4)**



x	y1	y0	z	s1	r1	s0	r0
0	0	0	0	1	0	1	0
0	0	1	1	0	-	-	0
0	1	0	0	0	1	1	0
0	1	1	1	-	0	-	0
1	0	0	0	1	0	0	-
1	0	1	1	0	-	0	1
1	1	0	1	-	0	0	-
1	1	1	1	-	0	0	1

Ricaviamo da tale tabella le EB minime per z, s1, r1, s0 ed r0:

$$z = y_0 + xy_1$$

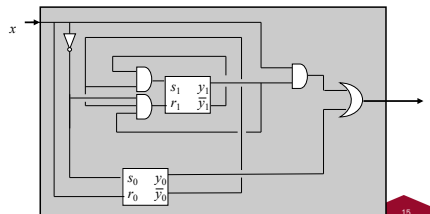
$$s_1 = \bar{y}_1 \bar{y}_0$$

$$r_1 = \bar{x} y_1 \bar{y}_0$$

$$s_0 = \bar{x}$$

$$r_0 = x$$

Da queste EB, ricavo il circuito:



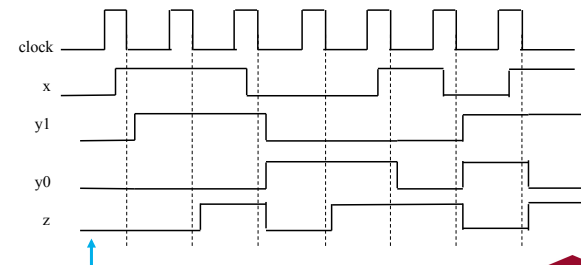
15

**Un secondo esercizio (4)**



	0	1
<b>S0</b> (00)	S3/0	S2/0
<b>S1</b> (01)	S1/1	S0/1
<b>S2</b> (10)	S1/0	S2/1
<b>S3</b> (11)	S3/1	S2/1

Diagramma temporale per l'input 1100101:



16