

Capitolo 7

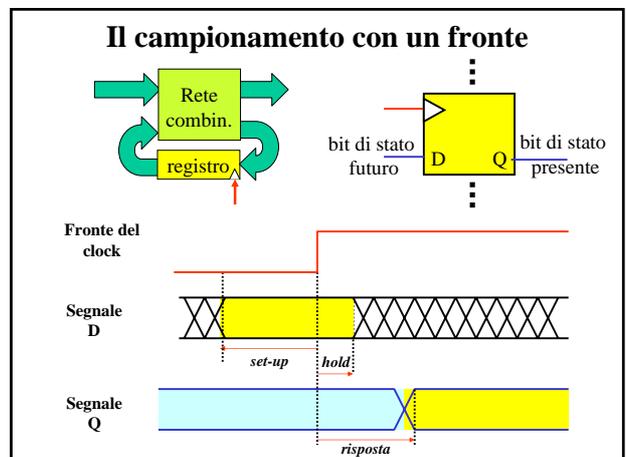
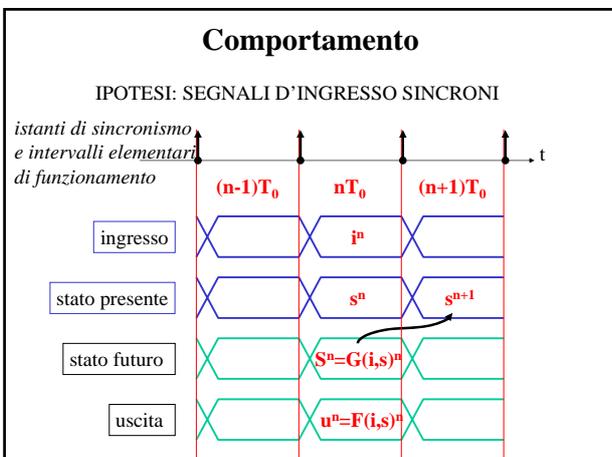
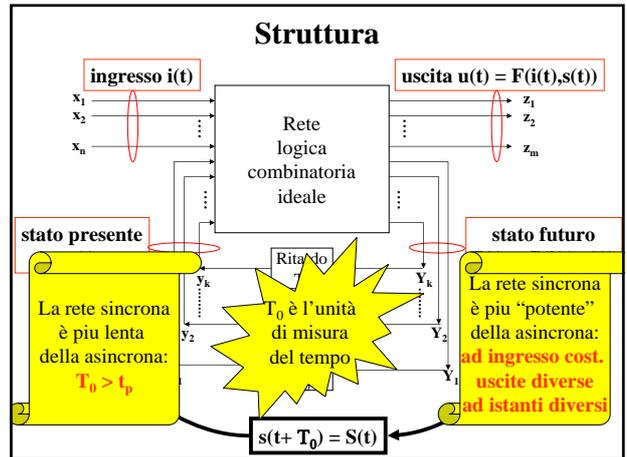
Reti sincrone

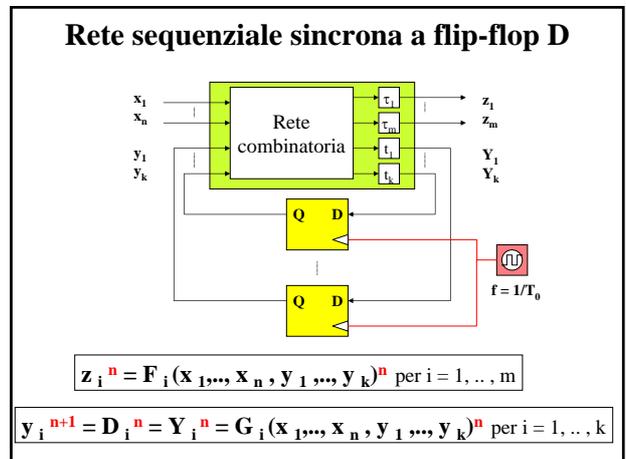
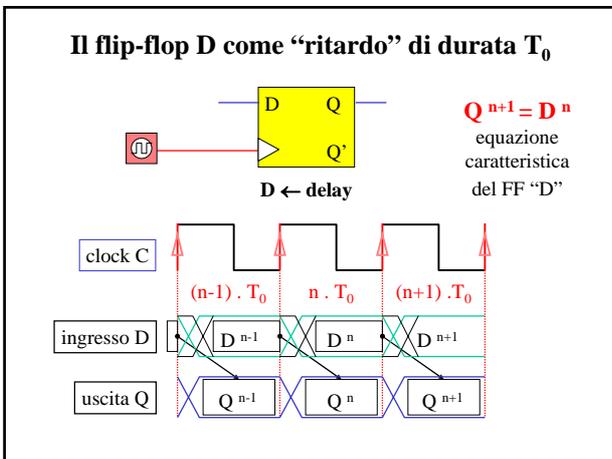
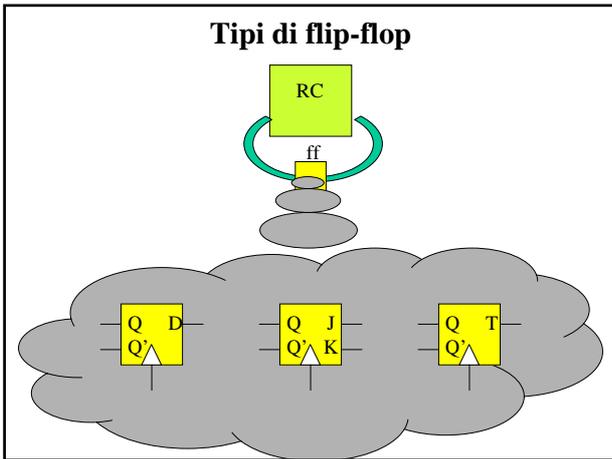
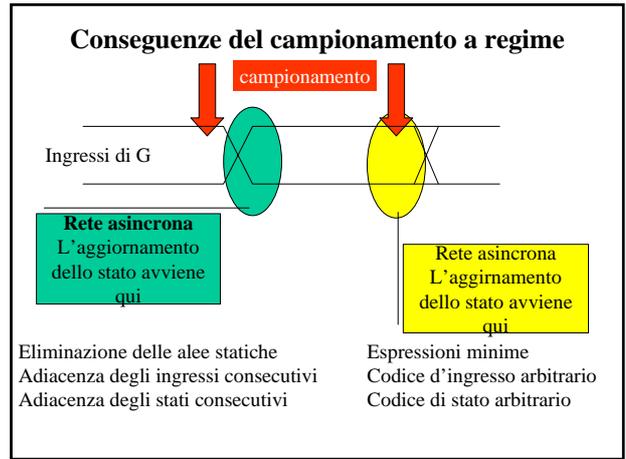
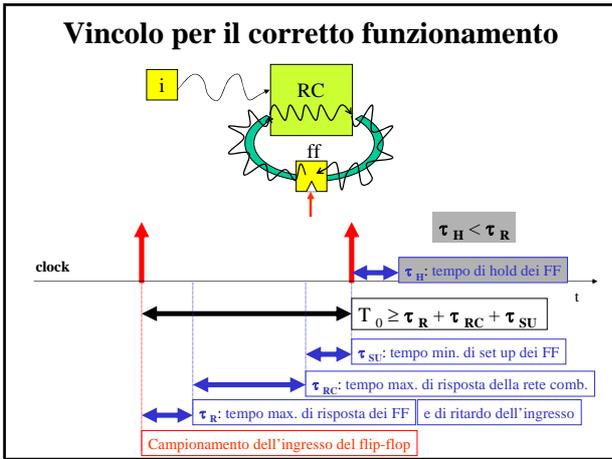
7.1 – Elaborazione sincrona
 7.2 - Analisi e Sintesi
 7.3 – Registri e Contatori

7.1

Elaborazione sincrona

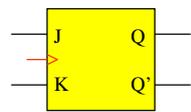
Esigenze e vincoli





Il flip-flop JK

Il flip-flop JK

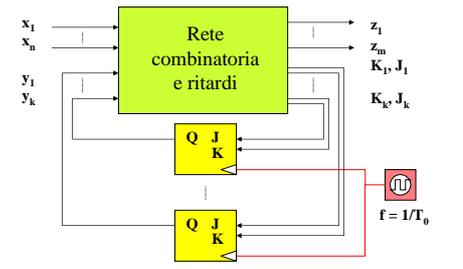


	J ⁿ	K ⁿ	Q ⁿ	Q ⁿ⁺¹
hold	0	0	0	0
	0	0	1	1
set	1	0	0	1
	1	0	1	1
reset	0	1	0	0
	0	1	1	0
toggle	1	1	0	1
	1	1	1	0

$$Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$$

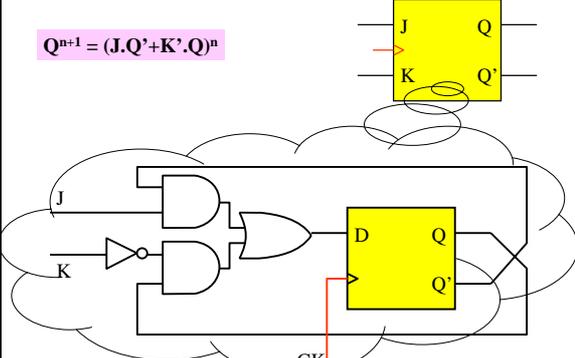
Q ⁿ	J ⁿ	K ⁿ	Q ⁿ⁺¹
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0

Rete sequenziale sincrona a flip-flop JK



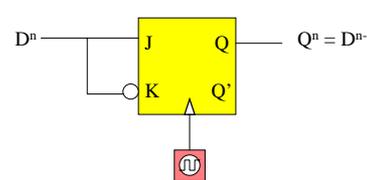
$z_i^n = F_i(x_1, \dots, x_n, y_1, \dots, y_k)^n$ per $i = 1, \dots, m$
 $y_i^{n+1} = (J_i \cdot y_i' + K_i' \cdot y_i)^n$ per $i = 1, \dots, k$
 con $J_i^n = J_i(x_1, \dots, x_n, y_1, \dots, y_k)^n$
 $K_i^n = K_i(x_1, \dots, x_n, y_1, \dots, y_k)^n$

Il flip-flop JK (struttura con ff D)



$$Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$$

Dal ff JK al ff D



$$Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$$

Pongo $J=D$ e $K=D'$

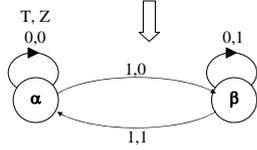
$$Q^{n+1} = (D \cdot Q' + D \cdot Q)^n$$

$$Q^{n+1} = D^n$$

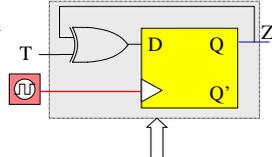
Il flip-flop T

Sintesi del flip-flop di tipo T (con ff D)

Comportamento: l'uscita Z commuta di valore al termine di ogni intervallo in cui si verifica $T = 1$.



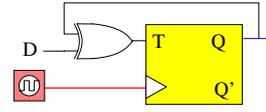
s^n	T^n	0	1
α		$\alpha, 0$	$\beta, 0$
β		$\beta, 1$	$\alpha, 1$



Equazione caratteristica:
 $Q^{n+1} = (T \oplus Q)^n$

Q^n	T^n	0	1
0		0,0	1,0
1		1,1	0,1

Dal flip-flop T al flip-flop D

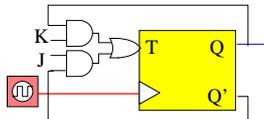
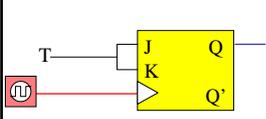


$Q^{n+1} = (T \oplus Q)^n$
Pongo $T = D \oplus Q$
 $Q^{n+1} = (D \oplus Q) \oplus Q^n$
 $Q^{n+1} = D^n$

Dal flip-flop T al JK e viceversa

Equazione caratteristica:
 $Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$
Pongo $J = K = T$
 $Q^{n+1} = (T \oplus Q)^n$

Equazione caratteristica:
 $Q^{n+1} = (T \cdot Q' + T' \cdot Q)^n$
Pongo $T = J \cdot Q' + K \cdot Q$
 $Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$



7.2 Analisi e Sintesi

Il procedimento di sintesi

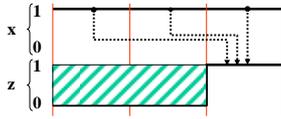
Il procedimento di sintesi di una rete sequenziale sincrona è formato da 5 passi e consente di dedurre lo schema logico dalle specifiche di comportamento:

- 1: comprensione delle specifiche
- 2: individuazione del grafo degli stati,
- 3: definizione della tabella di flusso,
- 4: codifica degli stati e definizione della tabella delle transizioni,
- 5: scelta dei flip-flop e sintesi della parte combinatoria,

Esempio: il riconoscitore di sequenza

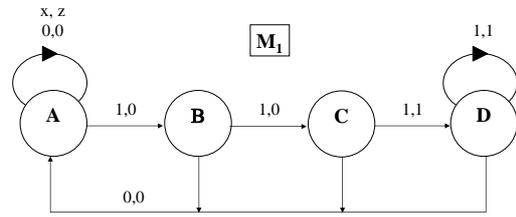
Una rete sequenziale sincrona ha un ingresso x ed una uscita z .
La relazione ingresso/uscita è descritta dalla seguente frase:

“ $z^n = 1$ quando $x^n = 1$ e solo se $x^{n-1} = x^{n-2} = 1$ ”



Riconoscitore di 3 “uni” consecutivi $z^n = x^n \cdot x^{n-1} \cdot x^{n-2}$

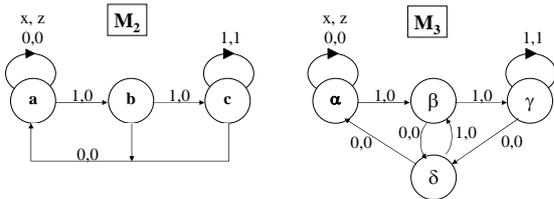
① Il grafo degli stati



• Si traccia la parte di grafo che riconosce la sequenza assegnata, specificando su ogni ramo il valore d'uscita (durata T_0).

• Si completa il grafo, prendendo in considerazione tutte le altre possibili situazioni e curando di renderlo “strettamente connesso” (ogni stato deve poter essere raggiunto da ogni altro).

② Macchine equivalenti



Macchine equivalenti - Sono dette equivalenti macchine sequenziali che presentano uno stesso comportamento impiegando un diverso numero di stati. (Esempio: $M_1 = M_2 = M_3$)

Macchina minima : macchina che, per un dato comportamento, ha il più piccolo insieme di stati. (Esempio: M_2)

③ Tabelle di flusso di macchine equivalenti

		M1		M3		M2		
		0	1	0	1	0	1	
A	A,0	B,0	α	α ,0	β ,0	a	a,0	b,0
B	A,0	C,0	β	δ ,0	γ ,0	b	a,0	c,0
C	A,0	D,1	γ	δ ,0	γ ,1	c	a,0	c,1
D	A,0	D,1	δ	α ,0	β ,0			

Le righe C e D sono identiche

Le righe α e δ sono identiche

M2 si ottiene da M1 ponendo $a = A$, $b = B$ e $c = \{C,D\}$
M2 si ottiene da M3 ponendo $a = \{\alpha,\delta\}$, $b = \beta$ e $c = \gamma$

③ Tabella di flusso di M_1

stato \ x	0	1
A	A,0	B,0
B	A,0	C,0
C	A,0	D,1
D	A,0	D,1

N.B. In una rete sequenziale sincrona ogni stato resta presente per almeno un periodo di clock, ogni cambiamento di ingresso avviene all'inizio di tali intervalli ed ogni transizione si verifica al termine. La stabilità dello stato presente non è una condizione necessaria dopo una variazione di ingresso. E' proprio la assenza di questo vincolo che consente di specificare comportamenti di tipo 2 o di tipo 3.

4- codifica degli stati e t.d.t.

		x^n	
		0	1
M1	$y_1^n y_2^n$		
	A: 00	00,0	10,0
	B: 10	00,0	11,0
	C: 11	00,0	01,1
D: 01	00,0	01,1	

$y_1^{n+1} y_2^{n+1}, z^n$

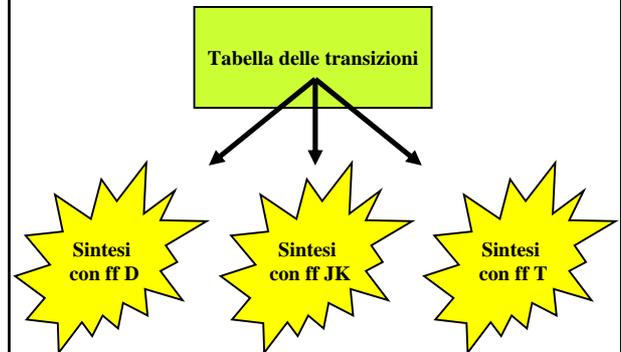
Codifica degli stati - In una rete sequenziale sincrona la codifica degli stati è **arbitraria** ($2^n \geq M$, naturalmente!). Il campionamento a regime dei segnali di stato elimina infatti a priori il problema di errate interpretazioni causate dal loro iniziale disallineamento.

Tabelle delle transizioni di M1, M3, M2

M1			M3			M2		
$y_1^n y_2^n$	0	1	$y_1^n y_2^n$	0	1	$y_1^n y_2^n$	0	1
A:00	00,0	10,0	α :00	00,0	10,0	a:00	00,0	10,0
B:10	00,0	11,0	β :10	01,0	11,0	b:10	00,0	11,0
C:11	00,0	01,1	γ :11	01,0	11,1	c:11	00,0	01,1
D:01	00,0	01,1	δ :01	00,0	10,0	01	--,-	--,-

$y_1^{n+1} y_2^{n+1} z^n$ $y_1^{n+1} y_2^{n+1} z^n$ $y_1^{n+1} y_2^{n+1} z^n$

5- Scelta del flip-flop



Sintesi con ff D

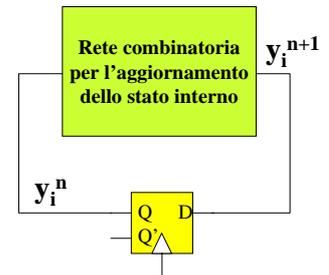
Funzioni di eccitazione: ff D

Flip-flop D

$$Q^{n+1} = D^n$$

Q^{n+1}	Q^n	D^n
0	0	0
1	1	1
1	0	1
0	1	0

$$D^n = Q^{n+1}$$



④&⑤ M1: sintesi con flip-flop D

Ipotesi: si cercano reti minime di tipo SP

$y_1 y_2$	00	01	11	10	$y_1 y_2$	00	01	11	10	$y_1 y_2$	00	01	11	10
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	1	1	1	1	0	1	1	0

y_1^{n+1} y_2^{n+1} z^n

$D_1 = y_1^{n+1} = x \cdot y_2'$ $D_2 = y_2^{n+1} = x \cdot y_2 + x \cdot y_1$ $z = x \cdot y_2$

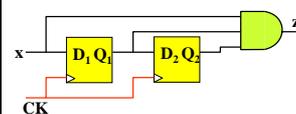
Copertura delle funzioni di eccitazione - Il campionamento a regime dei segnali di stato elimina a priori il pericolo di alee statiche e dinamiche.

M3: sintesi con flip-flop D

Ipotesi: reti minime di tipo SP

$Q_1 Q_2$	00	01	11	10	$Q_1 Q_2$	00	01	11	10	$Q_1 Q_2$	00	01	11	10
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
1	1	1	1	1	1	0	0	1	1	1	0	0	1	0

$D_1^n = Q_1^{n+1} = x^n$ $D_2^n = Q_2^{n+1} = Q_1^n$ $z^n = x^n \cdot Q_1^n \cdot Q_2^n$



Verifica del comportamento:

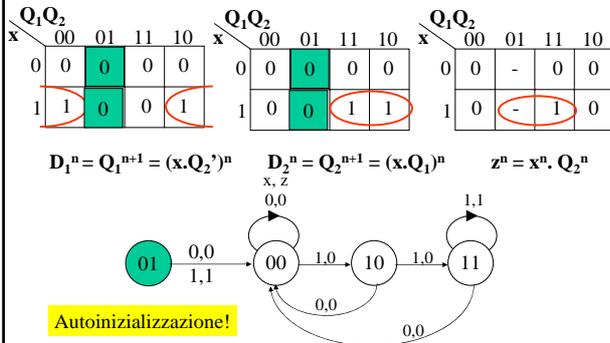
$$Q_1^{n+1} = x^n$$

$$Q_2^{n+1} = Q_1^n = x^{n-1}$$

$$z^n = (x \cdot Q_1 \cdot Q_2)^n = x^n \cdot x^{n-1} \cdot x^{n-2}$$

M2: sintesi con flip-flop D

Ipotesi: reti minime di tipo SP



Funzioni di eccitazione: ff JK

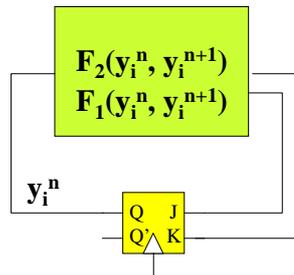
Flip-flop JK

$$Q^{n+1} = (JQ' + K'Q)^n$$

Q^{n+1}	Q^n	J^n	K^n
0	0	0	-
1	1	-	0
1	0	1	-
0	1	-	1

$$J^n = F_1(Q^{n+1}, Q^n)$$

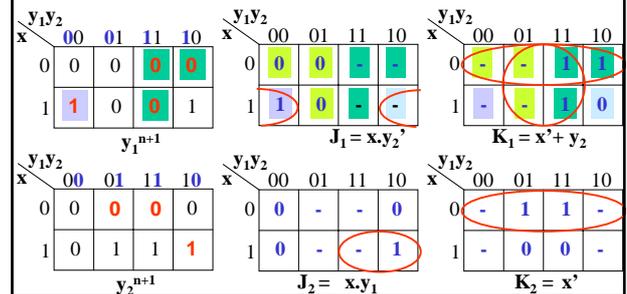
$$K^n = F_2(Q^{n+1}, Q^n)$$



M1: sintesi con flip-flop JK

Q^{n+1}	Q^n	J^n	K^n
0	0	0	-
1	1	-	0
1	0	1	-
0	1	-	1

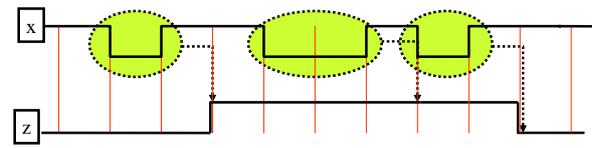
Equazione caratteristica:
 $Q^{n+1} = (J.Q' + K'.Q)^n$
Funzioni di eccitazione JK



Caso di studio: conteggio di eventi



La rete sequenziale sincrona di figura deve continuamente contare modulo 2 gli intervalli di tempo in cui si verifica $x = 0$. Il risultato del conteggio appare su z e viene aggiornato solo al termine di ogni intervallo in cui non si è contato ($x = 1$). I valori $z = 0$ e $z = 1$ indicano rispettivamente che la rete ha visto un numero "pari" ed un numero "dispari" di intervalli con $x = 0$.



Grafo degli stati

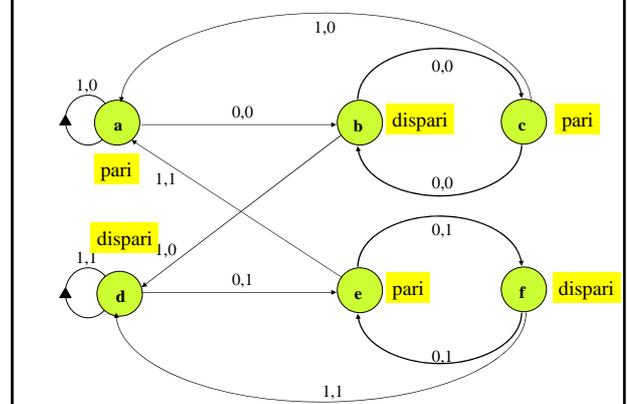


Tabelle di flusso

		x	
stato		0	1
a	b,0	a,0	
b	c,0	d,0	
c	b,0	a,0	
d	e,1	d,1	
e	f,1	a,1	
f	e,1	d,1	

$a = \{a, c\}$

		x	
stato		0	1
a	b,0	a,0	
b	a,0	d,0	
d	e,1	d,1	
e	d,1	a,1	

$d = \{d, f\}$

Stati indistinguibili - Sono detti indistinguibili stati a partire dai quali il comportamento della macchina è identico per qualsiasi sequenza di ingresso (esempio: $a = c$, $d = f$).

Sostituendo una "classe" di stati indistinguibili con un unico stato si ottiene una macchina equivalente a quella considerata.

Macchina minima

		x	
stato		0	1
a	b,0	a,0	
b	c,0	d,0	
c	b,0	a,0	
d	e,1	d,1	
e	f,1	a,1	
f	e,1	d,1	

$a = \{a, c\}$

$d = \{d, f\}$

		x	
stato		0	1
a	b,0	a,0	
b	a,0	d,0	
d	e,1	d,1	
e	d,1	a,1	

		x	
y ₁ y ₂		0	1
01	00,0	01,0	
00	01,0	10,0	
10	11,1	10,1	
11	10,1	01,1	

$z = y_1$

Sintesi con ff JK

		x	
y ₁ y ₂		0	1
01	00	01	
00	01	10	
10	11	10	
11	10	01	

		x	
y ₁ y ₂		0	1
01	0	0	
00	0	1	
10	-	-	
11	-	-	

$J_1 = x \cdot y_2'$
 $K_1 = x \cdot y_2$

		x	
y ₁ y ₂		0	1
01	-	-	
00	1	0	
10	1	0	
11	-	-	

$J_2 = x'$
 $K_2 = x'$

Sintesi con ff T

Funzioni di eccitazione

Flip-flop T

$$Q^{n+1} = (TQ' + T'Q)^n$$

		Q ⁿ		T ⁿ	
Q ⁿ⁺¹		0	1	0	1
0	0	0	0	0	0
1	1	1	0	1	0
1	0	1	1	0	1
0	1	0	1	1	1

$T^n = F_3(Q^{n+1}, Q^n)$

M1: sintesi con flip-flop T

		Q ⁿ⁺¹		Q ⁿ		T ⁿ	
0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	0
1	0	1	0	1	1	0	1
0	1	0	1	1	1	1	1

Equazione caratteristica:
 $Q^{n+1} = (T \cdot Q' + T' \cdot Q)^n$
Funzione di eccitazione T

		y ₁ y ₂			
x		00	01	11	10
0	0	0	0	1	1
1	1	0	0	1	0

$T_1 = x \cdot y_2' \cdot y_1' + y_1 \cdot y_2 + x' \cdot y_1$

		y ₁ y ₂			
x		00	01	11	10
0	0	0	1	1	0
1	0	0	0	0	1

$T_2 = x \cdot y_2' \cdot y_1 + x' \cdot y_2$

M1: sintesi con flip-flop T

$x \backslash y_1 y_2$	00	01	11	10
0	0	0	0	0
1	1	0	0	1

y_1^{n+1}

$x \backslash y_1 y_2$	00	01	11	10
0	0	0	1	1
1	1	0	1	0

$T_1^n = x \cdot y_2' \cdot y_1' + y_1 \cdot y_2 + x' \cdot y_1$

$x \backslash y_1 y_2$	00	01	11	10
0	0	0	0	0
1	0	1	1	1

y_2^{n+1}

$x \backslash y_1 y_2$	00	01	11	10
0	0	1	1	0
1	0	0	0	1

$T_2^n = x \cdot y_2' \cdot y_1 + x' \cdot y_2$

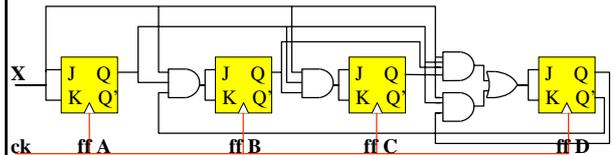


Il procedimento di analisi

Il procedimento di analisi di una rete sequenziale sincrona è formato da 5 passi e consente di dedurre il comportamento dallo schema logico:

- 1: analisi dei segnali d'ingresso di ciascun flip-flop,
- 2: deduzione delle variabili di stato futuro,
- 3: individuazione della tabella delle transizioni,
- 4: deduzione e studio della tabella di flusso,
- 5: tracciamento e studio del grafo degli stati.

Il contatore BCD



Si denominano i flip-flop e si scrivono le espressioni dei loro comandi:

$$J_A^n = K_A^n = X^n$$

$$J_B^n = K_B^n = (X \cdot Q_A \cdot Q_D)'^n$$

$$J_C^n = K_C^n = (X \cdot Q_A \cdot Q_B)'^n$$

$$J_D^n = K_D^n = (X \cdot Q_A \cdot Q_B \cdot Q_C + X \cdot Q_A \cdot Q_D)'^n$$

Espressioni di stato

Tramite l'equazione caratteristica si passa dalle espressioni delle funzioni di eccitazione a quelle delle variabili di stato futuro.

$$Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$$

Nel caso $J=K=T$ si ha

$$Q^{n+1} = (T \oplus Q)^n$$

$$Q_A^{n+1} = (X \oplus Q_A)^n$$

$$Q_B^{n+1} = ((X \cdot Q_A \cdot Q_D)' \oplus Q_B)^n$$

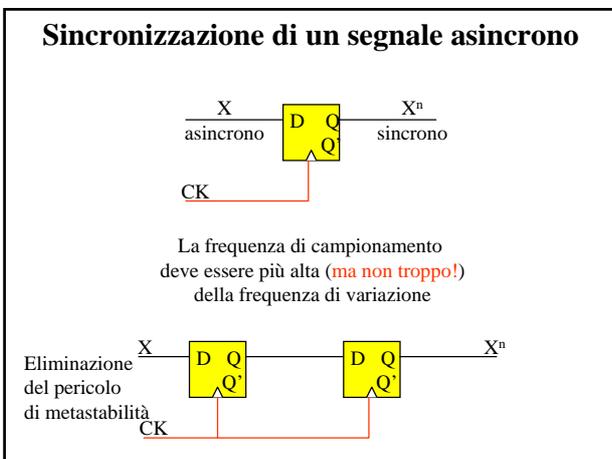
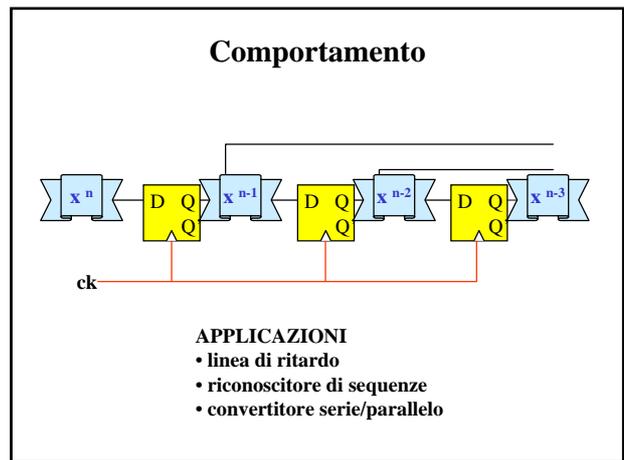
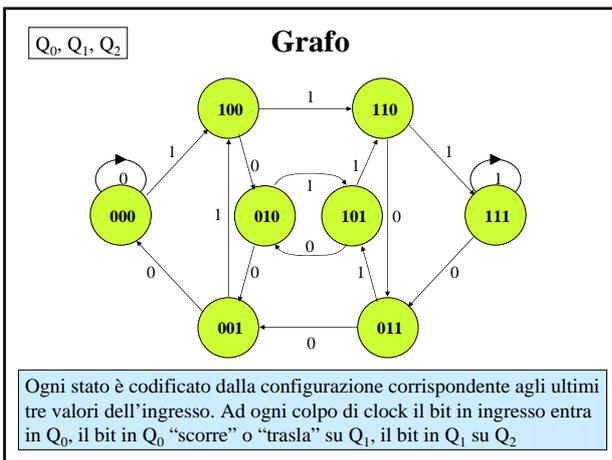
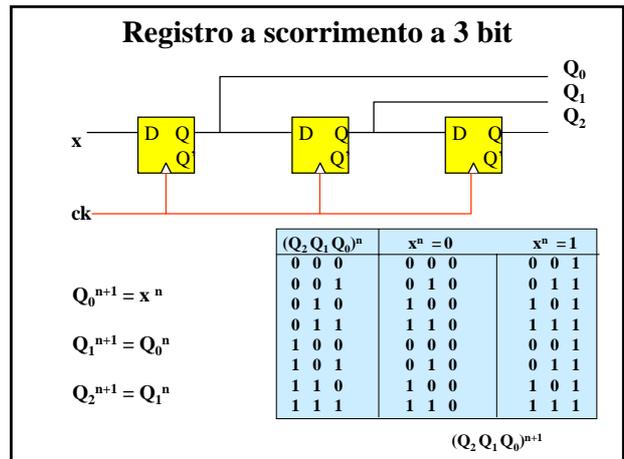
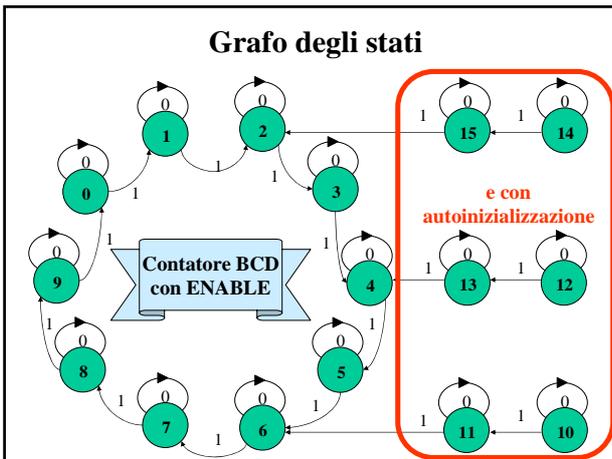
$$Q_C^{n+1} = ((X \cdot Q_A \cdot Q_B)' \oplus Q_C)^n$$

$$Q_D^{n+1} = ((X \cdot Q_A \cdot Q_B \cdot Q_C + X \cdot Q_A \cdot Q_D)' \oplus Q_D)^n$$

Tabella delle transizioni

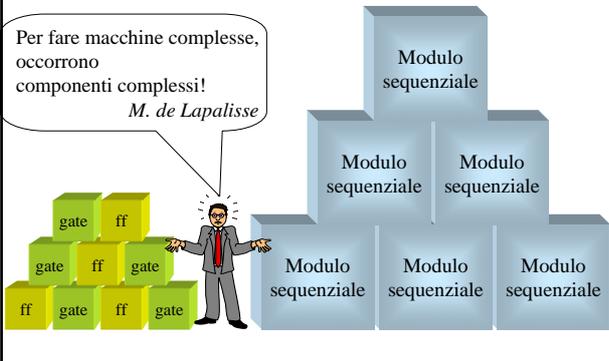
		X	
$Q_D^n Q_C^n Q_B^n Q_A^n$	0	1	
0 0 0 0	0 0 0 0	0 0 0 1	<div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> Per $X = 1$ e $S = 0000, 0001, \dots, 1001$ si ha: $(S)_2^{n+1} = (S+1)_2^n \text{ mod } 10$ </div> <div style="border: 1px solid green; padding: 5px;"> Per $X = 0$ si ha $(S)_2^{n+1} = (S)_2^n$ Un ingresso di questo tipo è denominato comando di ENABLE. </div>
0 0 0 1	0 0 0 1	0 0 1 0	
0 0 1 0	0 0 1 0	0 0 1 1	
0 0 1 1	0 0 1 1	0 1 0 0	
0 1 0 0	0 1 0 0	0 1 0 1	
0 1 0 1	0 1 0 1	0 1 1 0	
0 1 1 0	0 1 1 0	0 1 1 1	
0 1 1 1	0 1 1 1	1 0 0 0	
1 0 0 0	1 0 0 0	1 0 0 1	
1 0 0 1	1 0 0 1	0 0 0 0	
1 0 1 0	1 0 1 0	1 0 1 1	
1 0 1 1	1 0 1 1	0 1 1 0	
1 1 0 0	1 1 0 0	1 1 0 1	
1 1 0 1	1 1 0 1	0 1 0 0	
1 1 1 0	1 1 1 0	1 1 1 1	
1 1 1 1	1 1 1 1	0 0 1 0	

$Q_D^{n+1} Q_C^{n+1} Q_B^{n+1} Q_A^{n+1}$

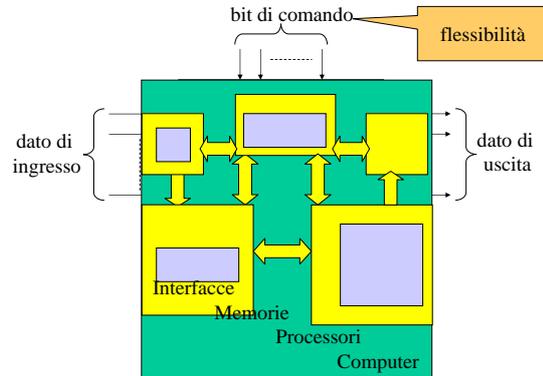


Progetto logico e famiglie logiche

Per fare macchine complesse, occorrono componenti complessi!
M. de Lapalisse



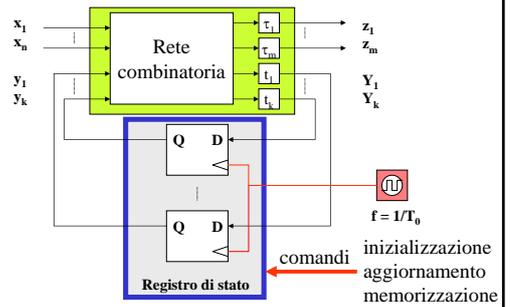
Moduli sequenziali programmabili



Il registro di stato

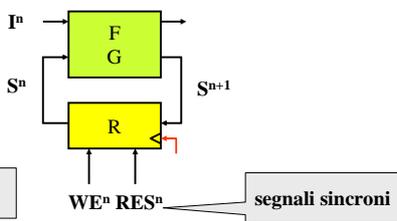


Il registro per lo stato interno



Registro di stato o registro accumulatore - Insieme di flip-flop, azionati dallo stesso clock e singolarmente dotati di un ingresso e di una uscita.

I comandi WE e RES

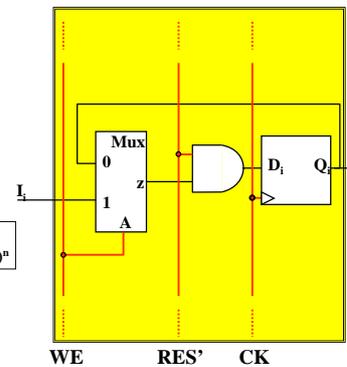


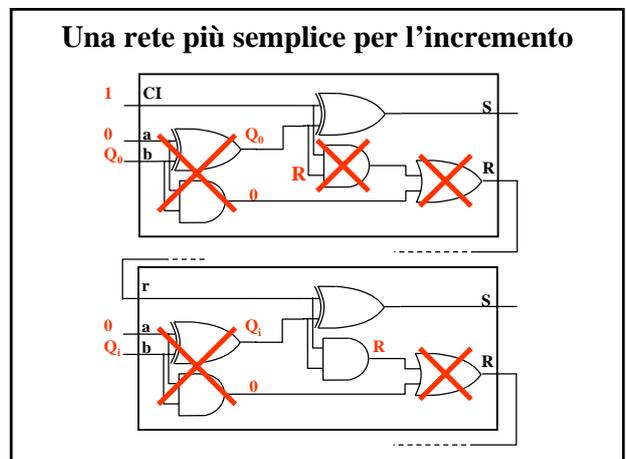
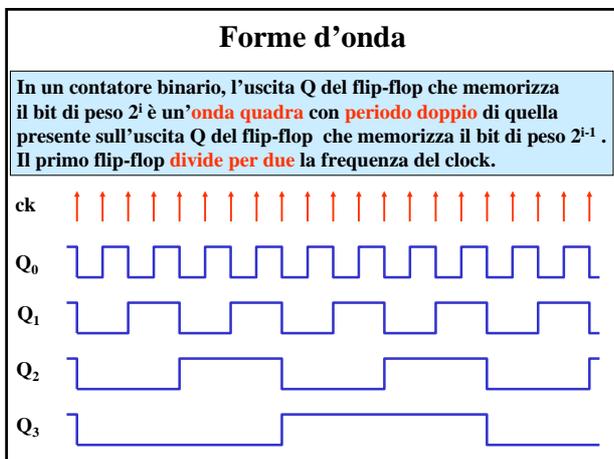
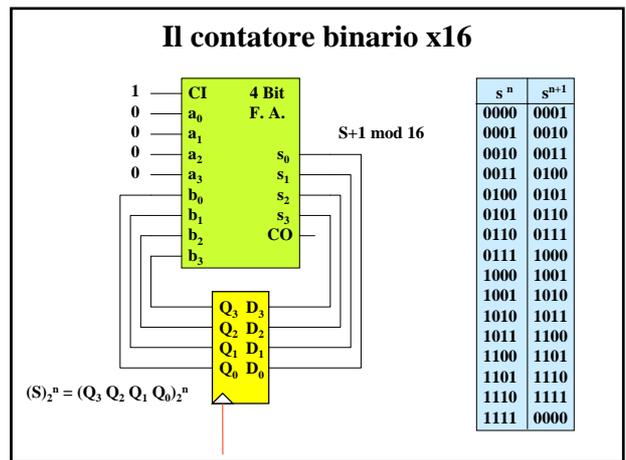
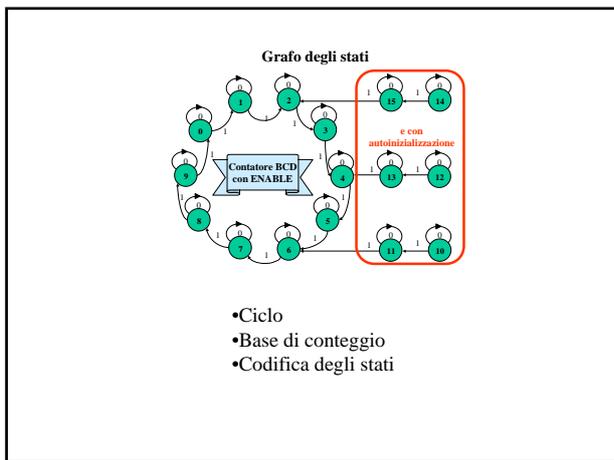
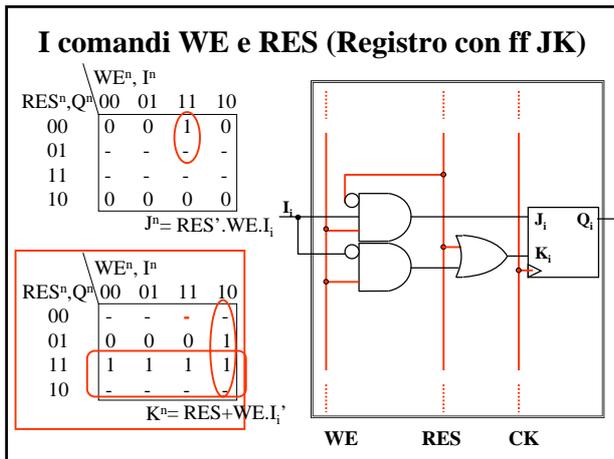
WE	RES^n	fase	comportamento
-	1	inizializzazione	$S^{n+1} = 0$
1	0	aggiornamento	$S^{n+1} = G(S^n, I^n)$
0	0	memorizzazione	$S^{n+1} = S^n$

I comandi WE e RES (Registro con ff D)

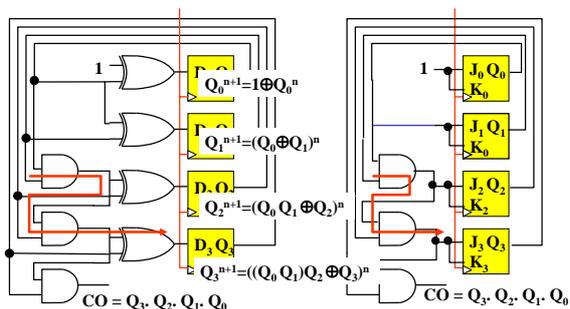
RES^n, Q^n	WE^n, I^n			
	00	01	11	10
00	0	0	1	0
01	1	1	1	0
11	0	0	0	0
10	0	0	0	0

Per $i = 0, 1, \dots, N-1$
 $Q_i^{n+1} = (RES' \cdot I_i \cdot WE + Q_i \cdot WE')$





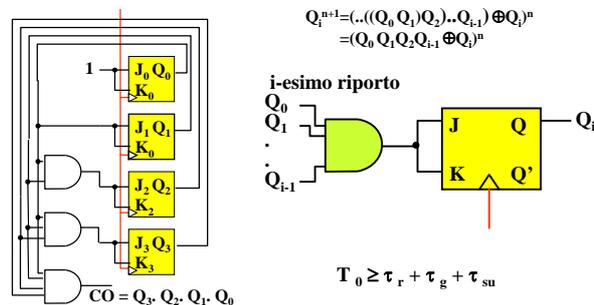
Periodo del clock e complessità della rete



$$T_0 \geq \tau_r + 3 \cdot \tau_g + \tau_{su}$$

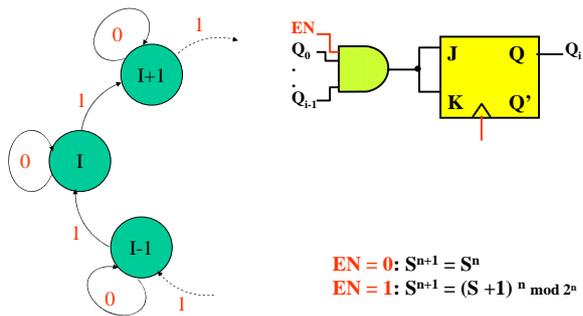
$$T_0 \geq \tau_r + 2 \cdot \tau_g + \tau_{su}$$

Un contatore ancora più veloce



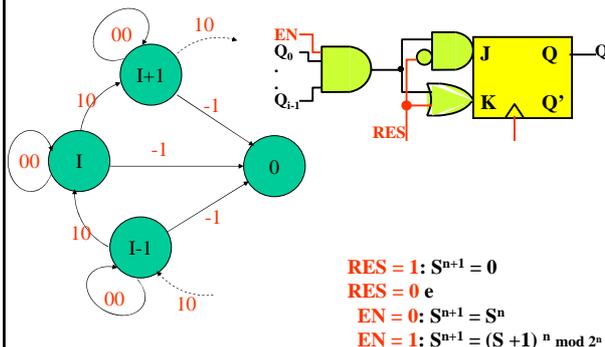
$$T_0 \geq \tau_r + \tau_g + \tau_{su}$$

Il comando di ENABLE



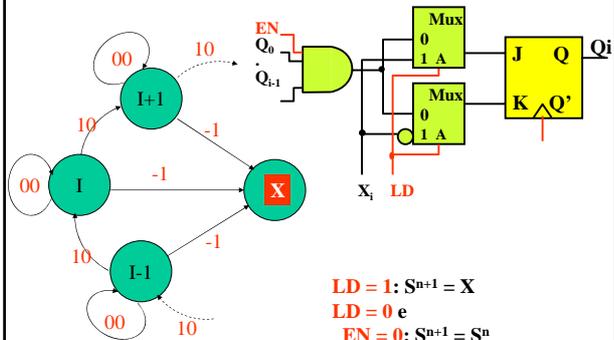
$$\begin{aligned} \text{EN} = 0: S^{n+1} &= S^n \\ \text{EN} = 1: S^{n+1} &= (S + 1)^n \bmod 2^n \end{aligned}$$

ENABLE & RESET



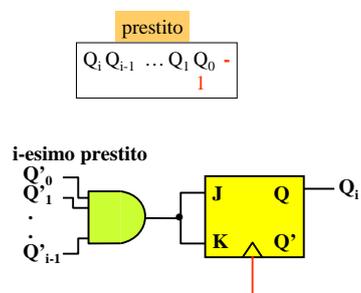
$$\begin{aligned} \text{RES} = 1: S^{n+1} &= 0 \\ \text{RES} = 0 \text{ e} \\ \text{EN} = 0: S^{n+1} &= S^n \\ \text{EN} = 1: S^{n+1} &= (S + 1)^n \bmod 2^n \end{aligned}$$

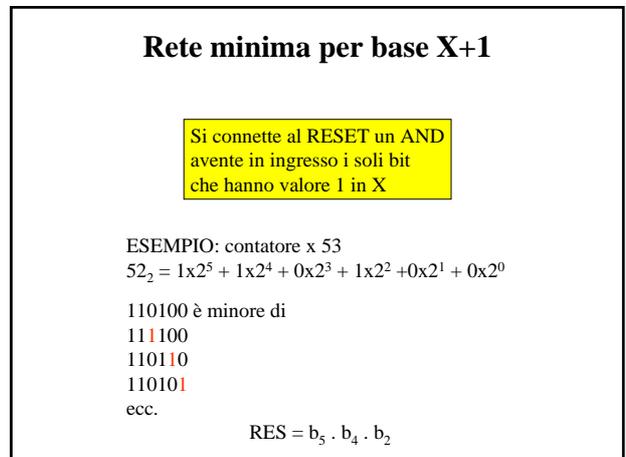
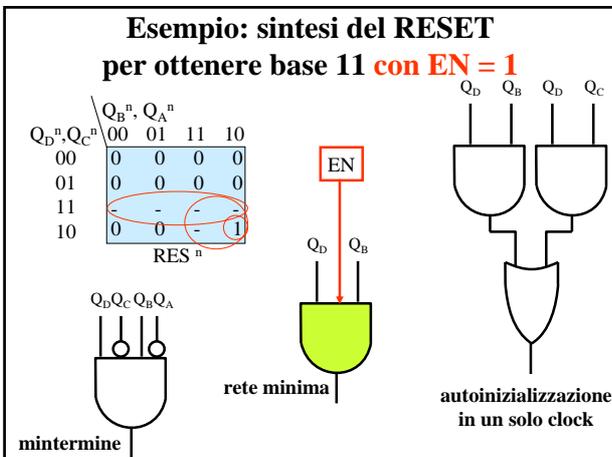
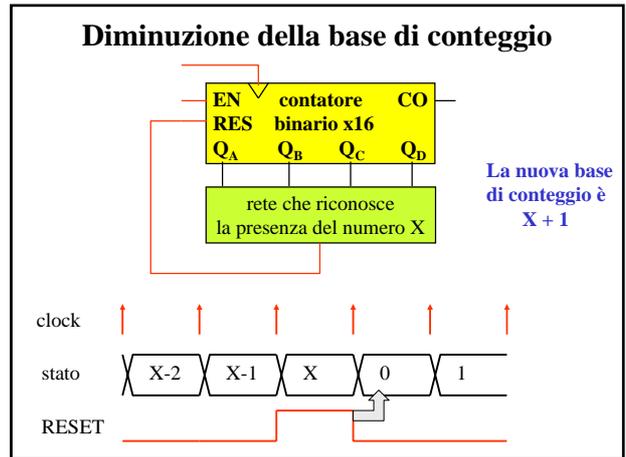
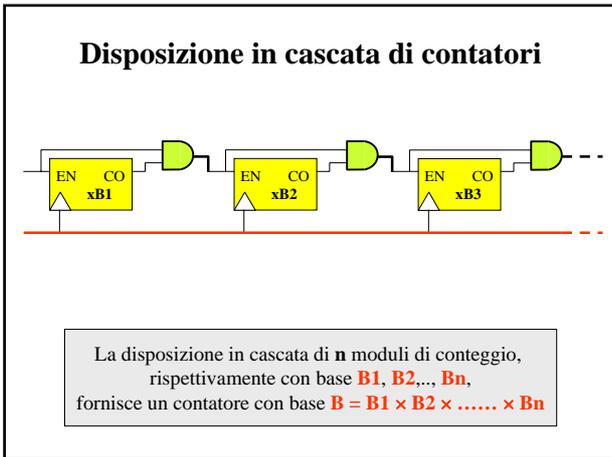
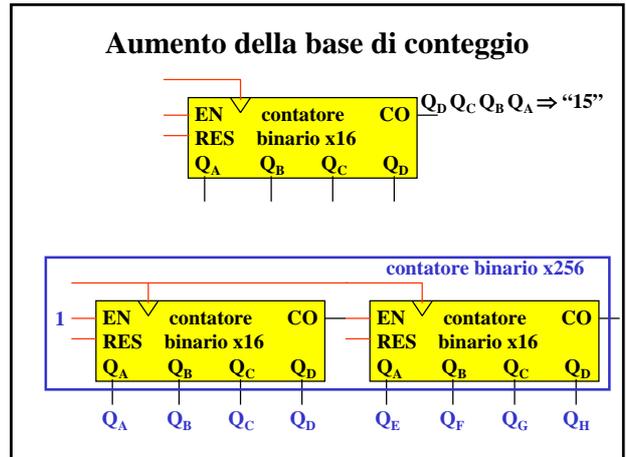
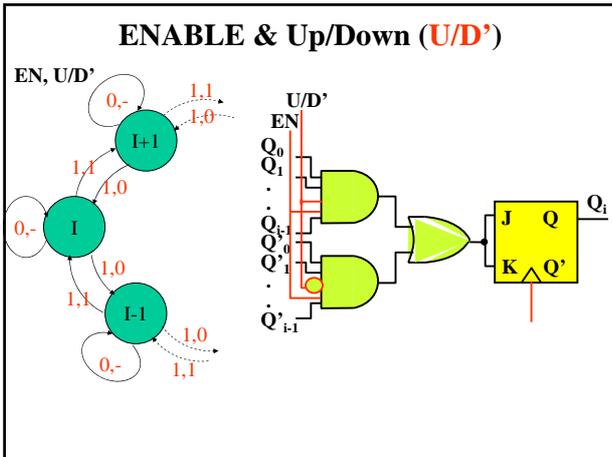
ENABLE & LOAD



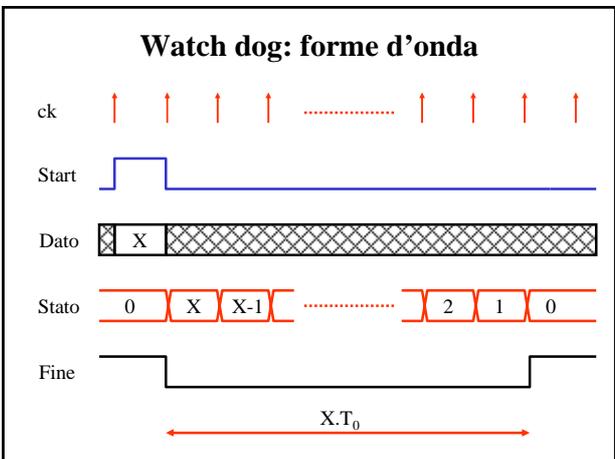
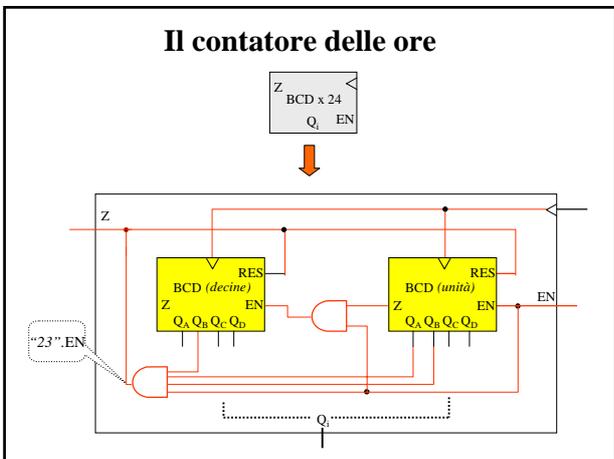
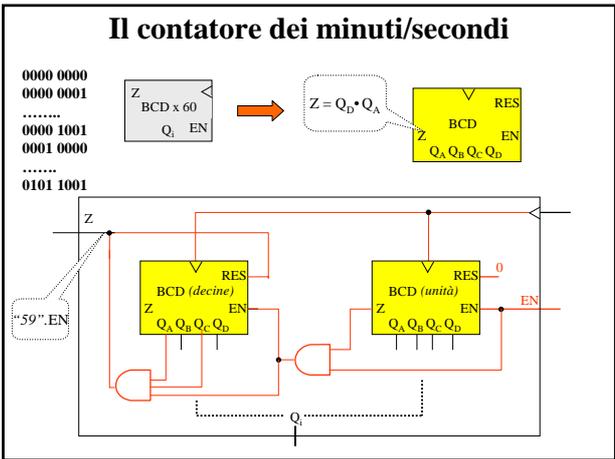
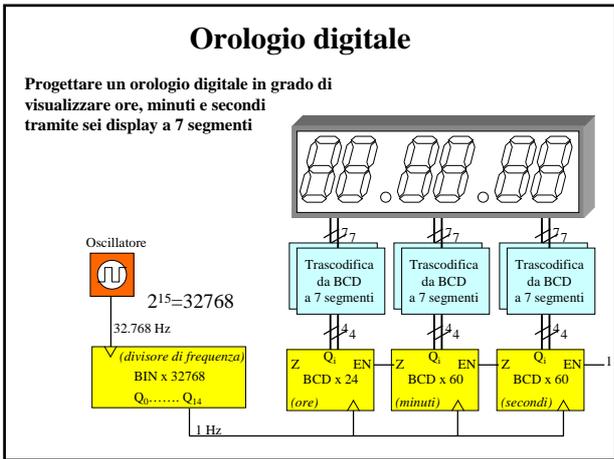
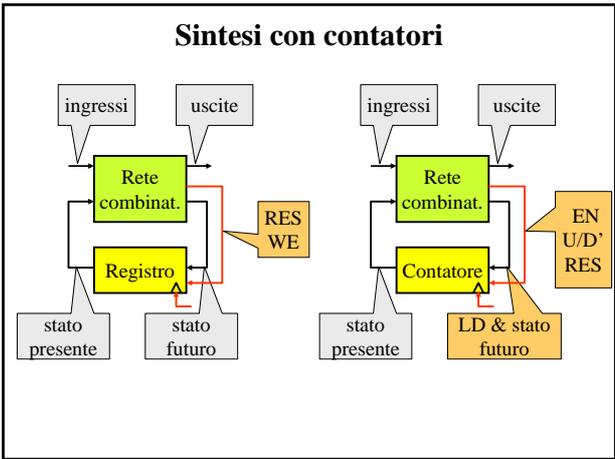
$$\begin{aligned} \text{LD} = 1: S^{n+1} &= X \\ \text{LD} = 0 \text{ e} \\ \text{EN} = 0: S^{n+1} &= S^n \\ \text{EN} = 1: S^{n+1} &= (S + 1)^n \bmod 2^n \end{aligned}$$

Il contatore "all'indietro"

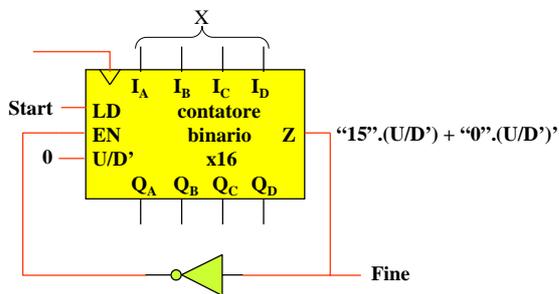




Sintesi con contatori

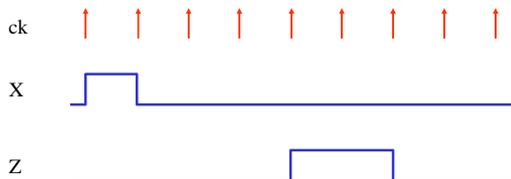


Watch dog: schema logico

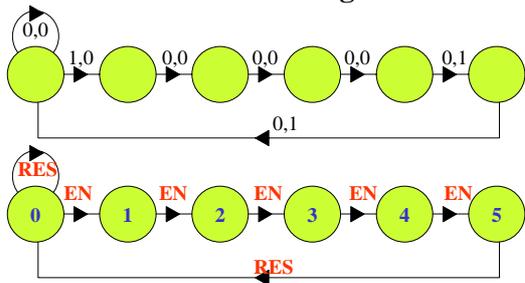


Esempio di sintesi con contatori

Una rete sequenziale sincrona ha un ingresso X che assume il valore 1 molto di rado e comunque sempre per un solo periodo di clock. L'uscita Z deve sia ritardare l'impulso di ingresso di quattro unità di tempo, sia raddoppiarne la durata.



Grafo e codifica degli stati



Il comportamento è quello di un contatore con base 6 che compie un intero ciclo per ogni evento X = 1
Per realizzarlo si può impiegare un contatore binario x8 dotato di comandi di ENABLE e di RESET.

Progetto di EN, RES, Z

stato	X=0	X=1	Z	stato	X=0	X=1
000	000	001	0	000	0,0,0	1,0,0
001	010	---	0	001	1,0,0	---
010	011	---	0	010	1,0,0	---
011	100	---	0	011	1,0,0	---
100	101	---	1	100	1,0,1	---
101	000	---	1	101	-,1,1	---
110	---	---	-	110	---	---
111	---	---	-	111	---	---

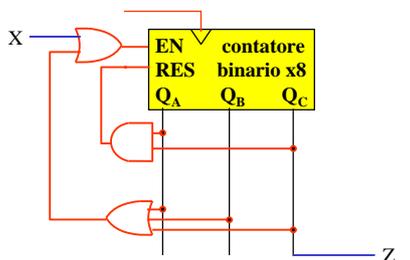
EN, RES, Z

$$EN = X + (Q_C + Q_B + Q_A) = X + (Q_C' Q_B' Q_A')$$

$$RES = Q_C Q_A$$

$$Z = Q_C$$

Schema logico



$$EN = X + (Q_C + Q_B + Q_A) = X + (Q_C' Q_B' Q_A')$$

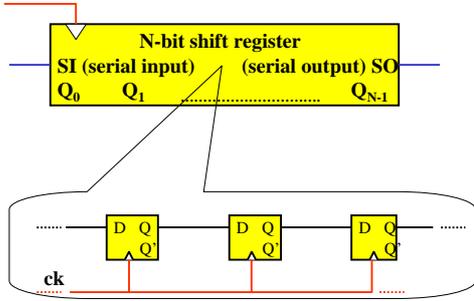
$$RES = Q_C Q_A$$

$$Z = Q_C$$

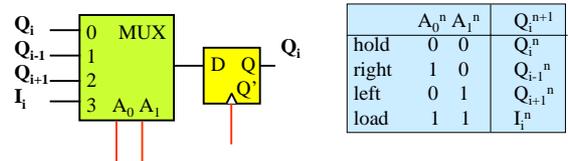
Registri
a scorrimento

Shift register

Shift register o registro a scorrimento - Rete sequenziale sincrona formata da N flip-flop D disposti in cascata.



Universal shift register

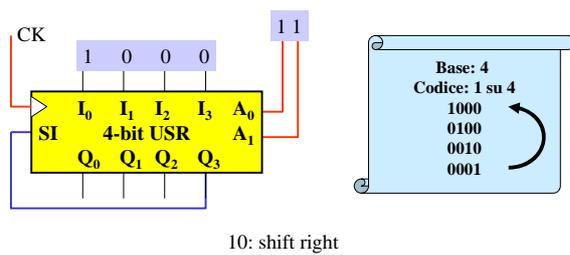


APPLICAZIONI

- linea di ritardo
- convertitore S/P e P/S
- conteggio
- memoria a circolazione
- rotazione verso destra/sinistra
- moltiplicazione/divisione per 2^i

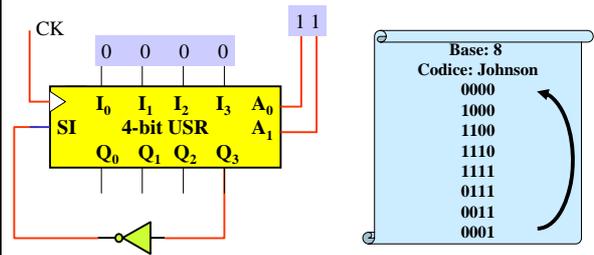
Contatore ad anello

...e se sta circolando una delle 12 configurazioni non utilizzate?
11: load



N flip-flop → Base N

Contatore a "riempimento/svuotamento"



N flip-flop → Base $2 \times N$

Esempio: conversioni S/P e P/S

