Capitolo 7

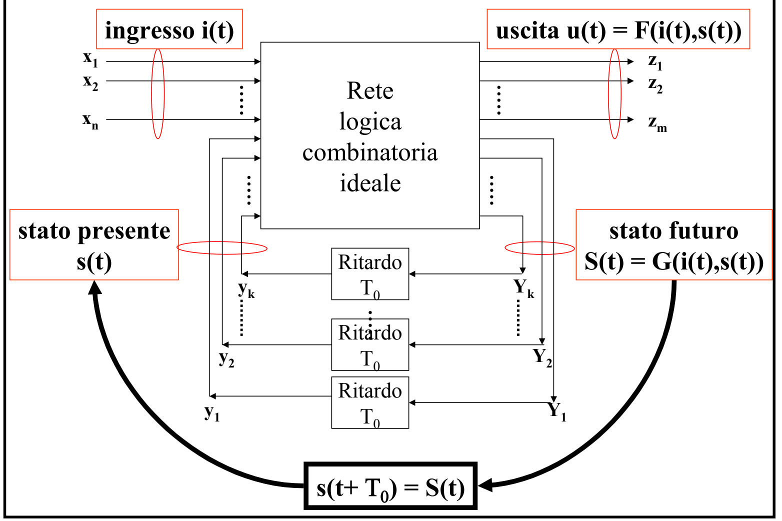
Reti sincrone

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

7.1 Elaborazione sincrona

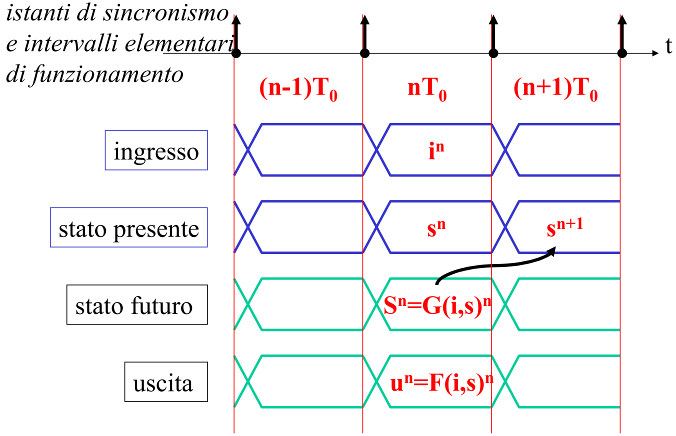
Esigenze e vincoli

Struttura

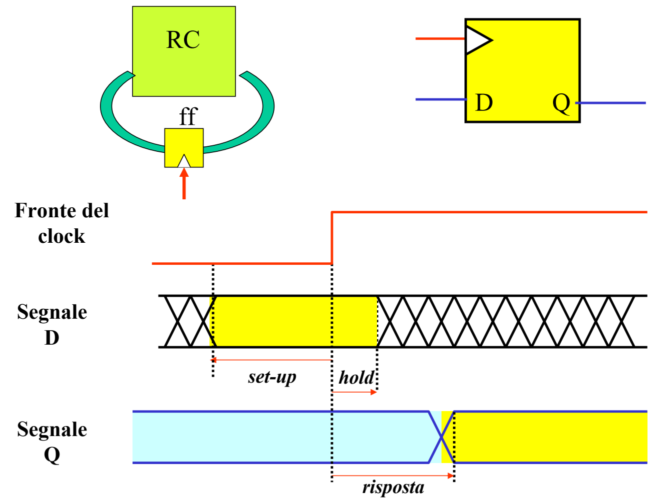


Comportamento

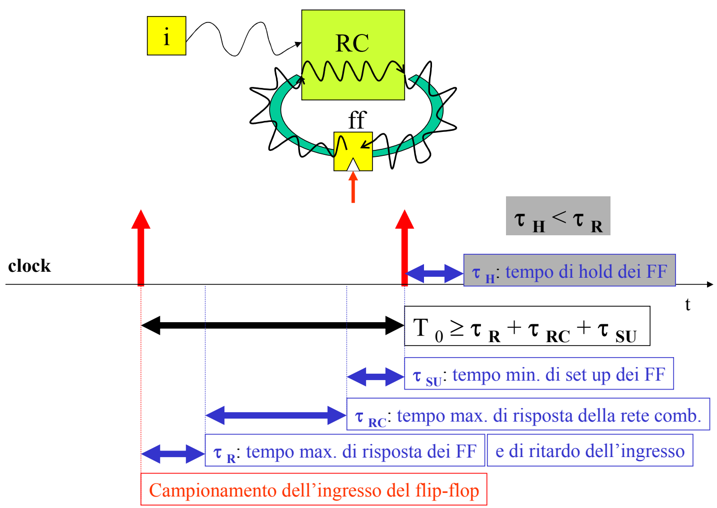
IPOTESI: SEGNALI D'INGRESSO SINCRONI



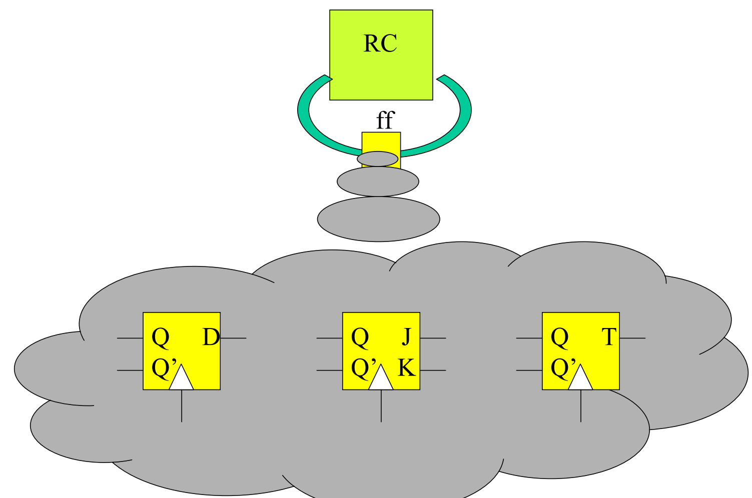
Il campionamento con un fronte



Vincolo per il corretto funzionamento

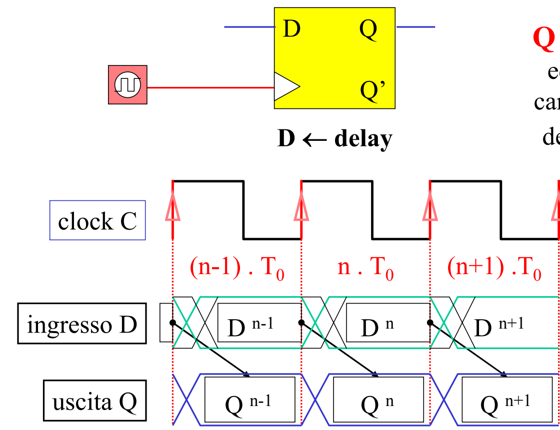


Tipi di flip-flop



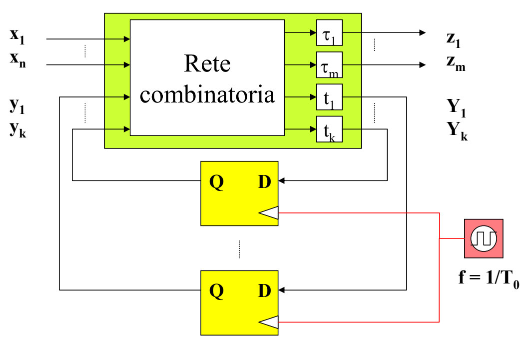
Il flip-flop D

Il flip-flop D come “ritardo” di durata T_0



$Q^{n+1} = D^{n}$
equazione caratteristica del FF “D”

Rete sequenziale sincrona a flip-flop D

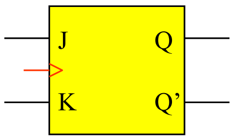


$z_i^{\,n} = F_i(x_1,\ldots, x_n, y_1,\ldots, y_k)^n$ per i = 1, .. , m

$y_i^{\,n+1} = D_i^{\,n} = Y_i^{\,n} = G_i(x_1,\ldots, x_n, y_1,\ldots, y_k)^n$ per i = 1, .. , k

Il flip-flop JK

Il flip-flop JK



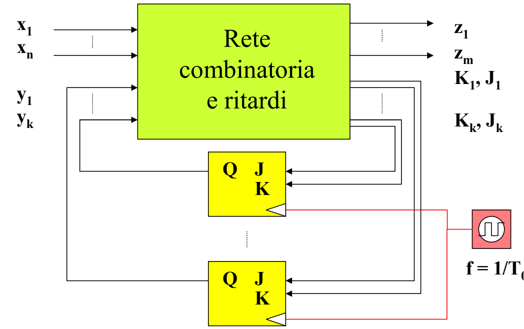
	J^n	K^n	Q^n	Q^{n+1}
hold	0	0	0	0
	0	0	1	1
set	1	0	0	1
	1	0	1	1
reset	0	1	0	0
	0	1	1	0
toggle	1	1	0	1
	1	1	1	0

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

Q^n \ $J^n K^n$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Q^{n+1}

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, .. , m

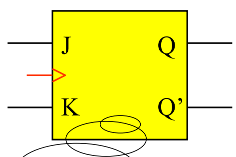
$y_i^{\,n+1} = (J_i.\, y_i' + K_i'.\, y_i)^n$ per i = 1, .. , k

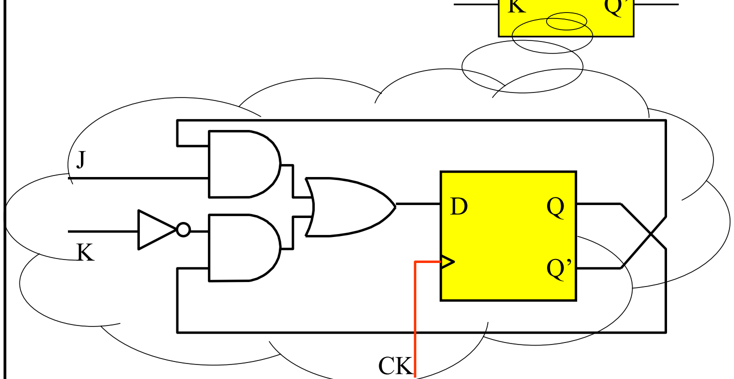
con $J_i^{\,n} = J_i(x_1,\dots, x_n, y_1,\dots, y_k)^n$

$K_i^{\,n} = K_i(x_1,\dots, x_n, y_1,\dots, y_k)^n$

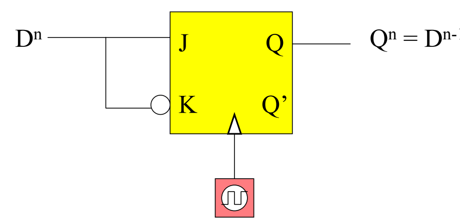
Il flip-flop JK (struttura con ff D)

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





Dal ff JK al ff D



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

Pongo J=D e K=D’

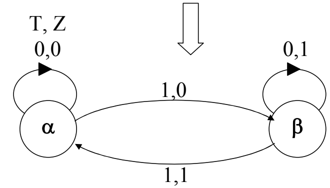
$Q^{n+1} = (D.Q' + D.Q)^n$

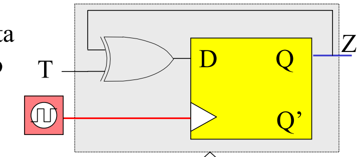
$Q^{n+1} = D^n$

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.





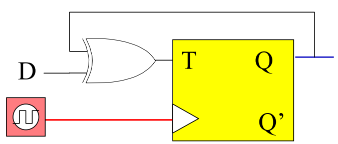
Equazione caratteristica:
$\mathbf{Q^{n+1} = (T \oplus Q)^n}$

s^n \ T^n	0	1
α	α,0	β,0
β	β,1	α,1

Errori nel testo

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

Dal flip-flop T al flip-flop D

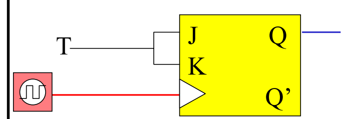


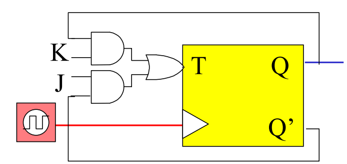
$\mathbf{Q^{n+1} = (T \oplus Q)^n}$
Pongo T = $\mathbf{D \oplus Q}$
$\mathbf{Q^{n+1} = ((D \oplus Q) \oplus Q)^n}$
$\mathbf{Q^{n+1} = D^n}$

Dal flip-flop T al JK e viceversa

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

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





7.2
Analisi e Sintesi

Il procedimento di sintesi

Il procedimento di sintesi

Il procedimento di sintesi di una rete sequenziale sincrona è formato da 6 passi e consente di dedurne lo schema logico dal comportamento:

1: individuazione del grafo degli stati,

2: definizione della tabella di flusso,

3: codifica degli stati e definizione della tabella delle transizioni,

4: scelta dei flip-flop,

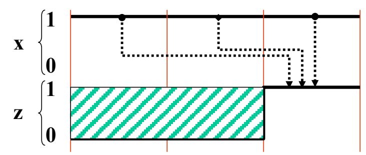
5: sintesi della parte combinatoria,

6: disegno dello schema logico

Esempio: il riconoscitore di sequenza

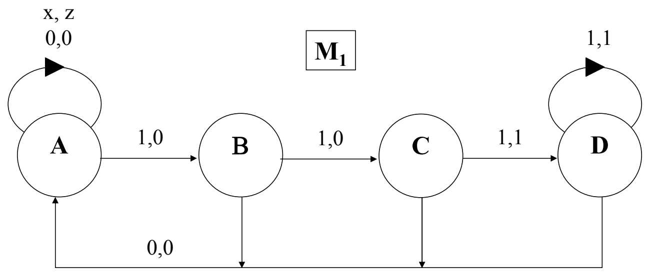
Una rete sequenziale sincrona ha un ingresso x ed una uscita z. La relazione ingresso/uscita è descritta dalla seguente frase:

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



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

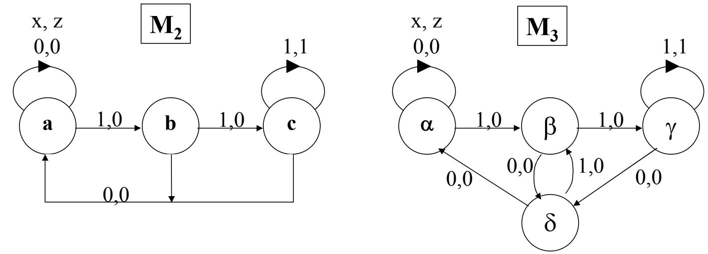
① Il grafo degli stati



•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



<u>Macchine equivalenti</u> - Sono dette equivalenti macchine sequenziali che presentano uno stesso comportamento impiegando un diverso numero di stati. (Esempio: $M_1 = M_2 = M_3$)
<u>Macchina minima</u> : macchina che, per un dato comportamento, ha il più piccolo insieme di stati. (Esempio: M_2)

Tabelle di flusso di macchine equivalenti

M1

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

M3

	0	1
α	α,0	β,0
β	δ,0	γ,0
γ	δ,0	γ,1
δ	α,0	β,0

M2

	0	1
a	a,0	b,0
b	a,0	c,0
c	a,0	c,1

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 = {α,δ}, b = β e c = γ

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 per la variazione di ingresso. E' proprio la assenza di questo vincolo che consente di specificare comportamenti di tipo 2 o di tipo 3.

Codifica degli stati

$y_1^n y_2^n$ \ x^n	0	1
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.

④&⑤ Sintesi di M_1 con flip-flop D

Ipotesi: si cercano reti minime di tipo SP

y_1^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	1	0	0	1

y_2^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	0	1	1	1

z^n

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	0	1	1	0

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

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

Sintesi di M_1 con flip-flop JK

y_1^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	1	0	0	1

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

y_1^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	1	0	0	1

x \ y_1y_2	00	01	11	10
0	0	0	-	-
1	1	0	-	-

$J_1 = x.y_2'$

x \ y_1y_2	00	01	11	10
0	-	-	1	1
1	-	-	1	0

$K_1 = x' + y_2$

y_2^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	0	1	1	1

x \ y_1y_2	00	01	11	10
0	0	-	-	0
1	0	-	-	1

$J_2 = x.y_1$

x \ y_1y_2	00	01	11	10
0	-	1	1	-
1	-	0	0	-

$K_2 = x'$

Sintesi di M_1 con flip-flop T

	Q^{n+1}	Q^n	T^n
0 ⇒	0	0	0
1 ⇒	1	1	0
1 ⇒	1	0	1
0 ⇒	0	1	1

y_1^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	1	0	0	1

x \ y_1y_2	00	01	11	10
0	0	0	1	1
1	1	0	1	0

$T_1 = x.\, y_2'.\, y_1' + y_1.y_2 + x'.y_1$

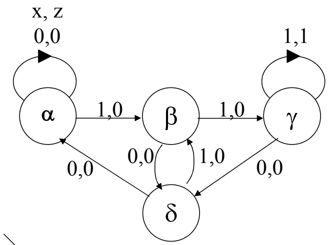
y_2^{n+1}

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	0	1	1	1

x \ y_1y_2	00	01	11	10
0	0	1	1	0
1	0	0	0	1

$T_2 = x.\, y_2'.\, y_1 + x'.y_2$

③&④&⑤ Grafo e tabella di flusso di M_3



stato \ x	0	1
α	α,0	β,0
β	δ,0	γ,0
γ	δ,0	γ,1
δ	α,0	β,0

Codifica e tabella delle transizioni di M_3

$Q_1^n Q_2^n$ \ x^n	0	1
α:00	00,0	10,0
β:10	01,0	11,0
γ:11	01,0	11,1
δ:01	00,0	10,0

$Q_1^{n+1} Q_2^{n+1}, z^n$

Sintesi di M_3 con flip-flop D

Ipotesi: reti minime di tipo SP

x \ Q_1Q_2	00	01	11	10
0	0	0	0	0
1	1	1	1	1

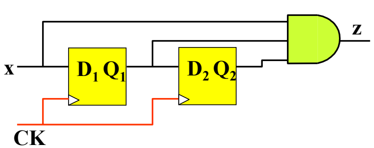
$D_1^n = Q_1^{n+1} = x^n$

x \ Q_1Q_2	00	01	11	10
0	0	0	1	1
1	0	0	1	1

$D_2^n = Q_2^{n+1} = Q_1^n$

x \ Q_1Q_2	00	01	11	10
0	0	0	0	0
1	0	0	1	0

$z^n = x^n . Q_1^n . Q_2^n$



Verifica del comportamento:

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

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

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

Esercitazione N. 15 (il semaforo)

	y_1	y_2	y_3
T0	0	0	0
T1	0	0	1
T2	0	1	0
T3	0	1	1
T4	1	0	0
T5	1	0	1
T6	1	1	0

s^n	s^{n+1}
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	1 0 1
1 0 1	1 1 0
1 1 0	0 0 0

Funzione di stato

s^n	u^n
0 0 0	1 0 0
0 0 1	1 0 0
0 1 0	1 0 0
0 1 1	0 1 0
1 0 0	0 0 1
1 0 1	0 0 1
1 1 0	0 0 1

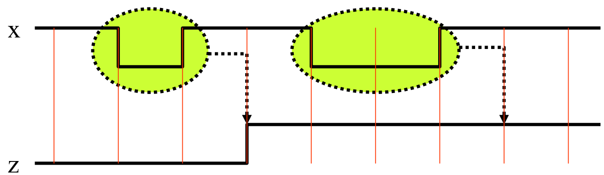
Funzione di uscita

	z_1	z_2	z_3
verde	1	0	0
giallo	0	1	0
rosso	0	0	1

Realizzare con AND, OR, NOT e FF “D”

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

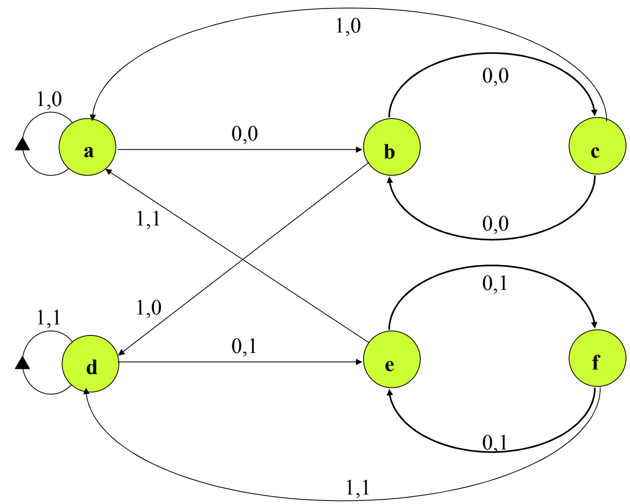


Tabelle di flusso

stato \ x	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}

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

<u>Stati indistinguibili</u> - 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.

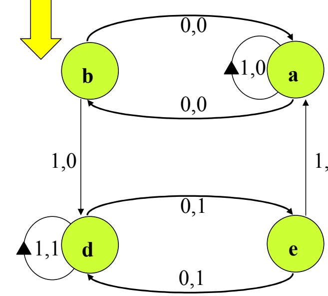
Macchina minima

stato \ x	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}

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



y1 \ y2	0	1
0	b	a
1	d	e

y_1y_2 \ x	0	1
01	00,0	01,0
00	01,0	10,0
10	11,1	10,1
11	10,1	01,1

Sintesi con ff JK

y_1y_2 \ x	0	1
01	00	01
00	01	10
10	11	10
11	10	01

J_1: y_1y_2 \ x	0	1
01	0	0
00	0	1
10	-	-
11	-	-

K_1: y_1y_2 \ x	0	1
01	-	-
00	-	-
10	0	0
11	0	1

$J_1 = x.y_2'$

$K_1 = x.y_2$

J_2: y_1y_2 \ x	0	1
01	-	-
00	1	0
10	1	0
11	-	-

K_2: y_1y_2 \ x	0	1
01	1	0
00	-	-
10	-	-
11	1	0

$J_2 = x'$

$K_2 = x'$

Esercitazione N. 16

Una RSS ha il compito di riprodurre sulla sua uscita z, con un ritardo di due intervalli di clock, il valore presente sul suo ingresso x, a condizione però che tale valore perduri per più di due intervalli.
Se il valore di x è presente solo per uno o per due intervalli, l'uscita z deve ignorarlo e mantenere il valore che aveva prima della variazione di x.

Il procedimento di analisi

Il procedimento di analisi

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

1: analisi dei segnali d'ingresso di ciascun flip-flop,

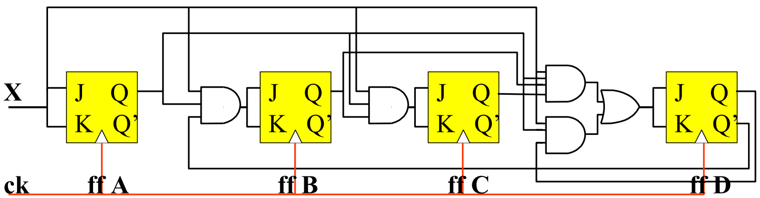
2: deduzione delle variabili di stato futuro,

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:

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

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

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

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

Espressioni di stato

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

$Q^{n+1} = (J.Q' + K'.Q)^n$
Nel caso J=K=T si ha
$Q^{n+1} = (T \oplus Q)^n$

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

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

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

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

Tabella delle transizioni

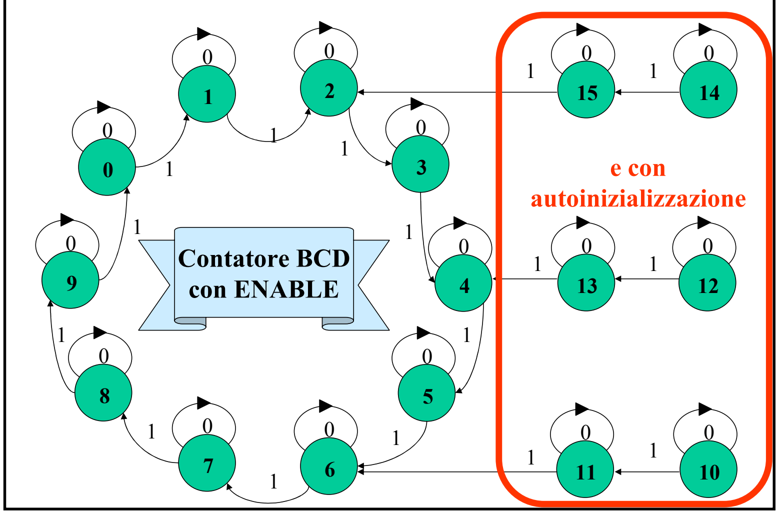
$Q_D\ Q_C\ Q_B\ Q_A$	X = 0	X = 1
0 0 0 0	0 0 0 0	0 0 0 1
0 0 0 1	0 0 0 1	0 0 1 0
0 0 1 0	0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 1	0 1 0 0
0 1 0 0	0 1 0 0	0 1 0 1
0 1 0 1	0 1 0 1	0 1 1 0
0 1 1 0	0 1 1 0	0 1 1 1
0 1 1 1	0 1 1 1	1 0 0 0
1 0 0 0	1 0 0 0	1 0 0 1
1 0 0 1	1 0 0 1	0 0 0 0
1 0 1 0	1 0 1 0	1 0 1 1
1 0 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

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

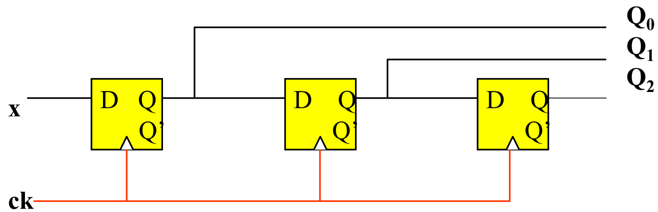
Per X = 1 e
S = 0000, 0001, .., 1001
si ha:
$(S)_2^{n+1} = (S+1)_2^n \bmod 10$

Per X = 0 si ha
$(S)_2^{n+1} = (S)_2^n$
Un ingresso di questo tipo è denominato **comando di ENABLE.**

Grafo degli stati



Registro a scorrimento



$Q_0^{n+1} = x^n$

$Q_1^{n+1} = Q_0^n$

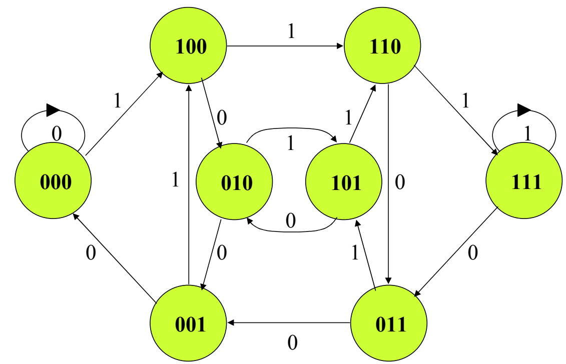
$Q_2^{n+1} = Q_1^n$

$(Q_2\,Q_1\,Q_0)^n$	$x^n = 0$	$x^n = 1$
0 0 0	0 0 0	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	1 0 0	1 0 1
0 1 1	1 1 0	1 1 1
1 0 0	0 0 0	0 0 1
1 0 1	0 1 0	0 1 1
1 1 0	1 0 0	1 0 1
1 1 1	1 1 0	1 1 1

$(Q_2\,Q_1\,Q_0)^{n+1}$

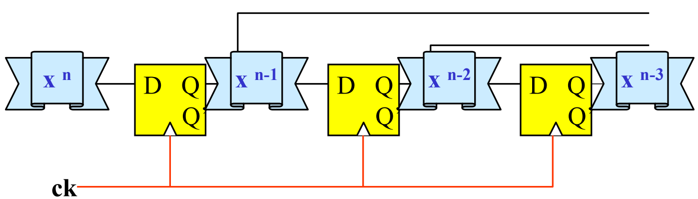
Grafo

Q_0, Q_1, Q_2



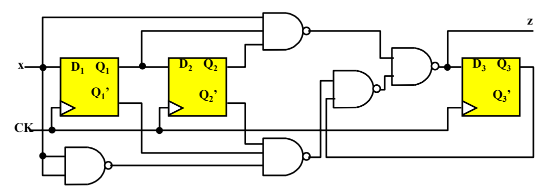
Ogni stato è codificato dalla configurazione corrispondente agli ultimi tre valori dell'ingresso. Ad ogni colpo di clock il bit in ingresso entra in Q_0, il bit in Q_0 "scorre" o "trasla" su Q_1, il bit in Q_1 su Q_2

Comportamento



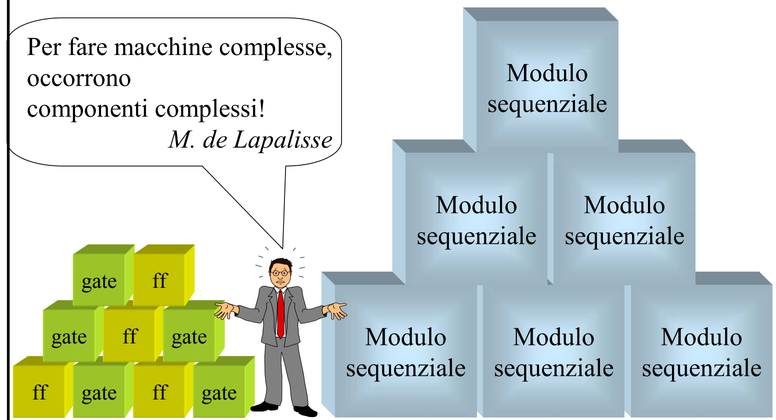
APPLICAZIONI
- **linea di ritardo**
- **riconoscitore di sequenze**
- **convertitore serie/parallelo**

Esercitazione N.17

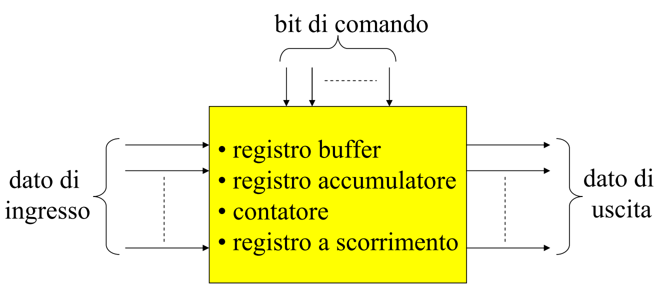


7.3
Registri e Contatori

Progetto logico e famiglie logiche

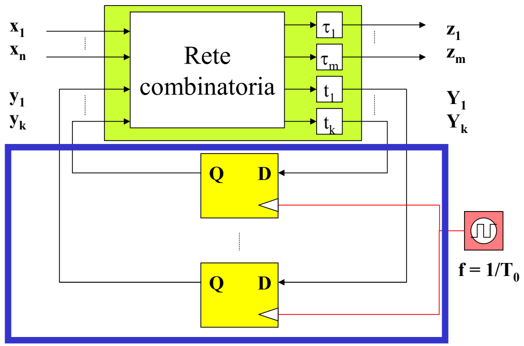


Moduli sequenziali programmabili



Il registro accumulatore

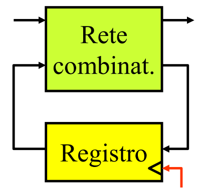
Il registro per lo stato interno



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

L'applicazione tipica di questo circuito integrato è la realizzazione delle retroazioni di una rete logica sequenziale sincrona.

I comandi dell'accumulatore



<u>Tre esigenze del registro accumulatore</u> :
1- inizializzazione,
2- aggiornamento,
3- memorizzazione.

<u>Una soluzione molto comoda</u>: avere nel registro anche una parte combinatoria in grado di interpretare due comandi.

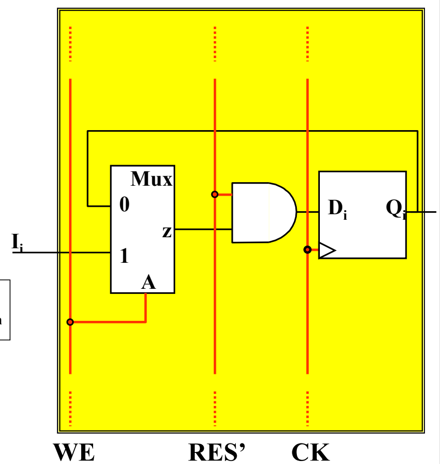
WE	**RES**	***esigenza***	***descrizione RTL***
-	**1**	**1**	**R ← 0**
1	**0**	**2**	**R ← G(R,I)**
0	**0**	**3**	**R ← R**

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

Q^{n+1}

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



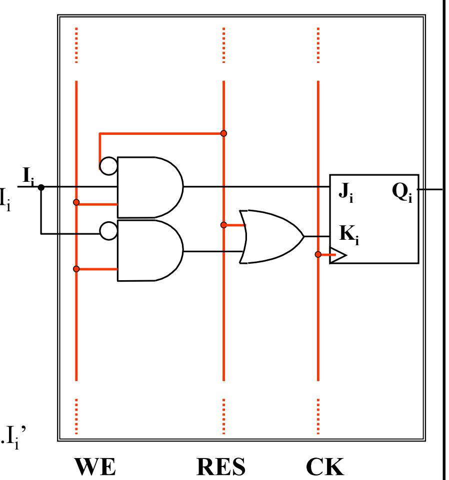
I comandi WE e RES (Registro con ff JK)

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

$J^n = RES'.WE.I_i$

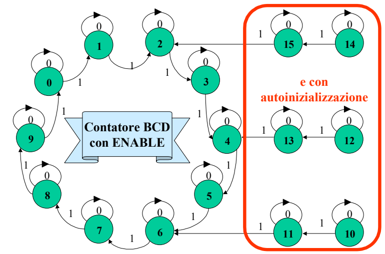
RES^n,Q^n \ WE^n, I^n	00	01	11	10
00	-	-	0	-
01	0	0	0	1
11	1	1	1	1
10	-	-	-	-

$K^n = RES+WE.I_i'$



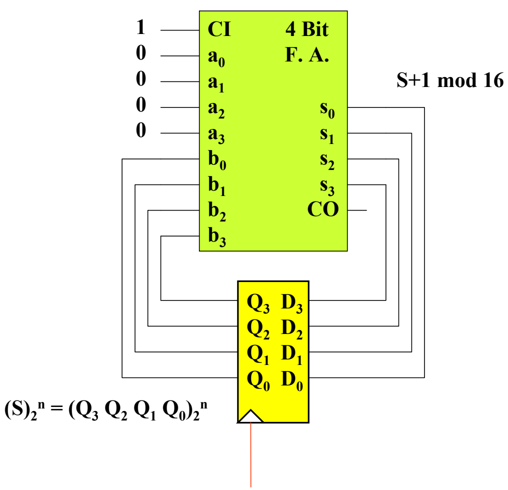
I contatori

Grafo degli stati



- Ciclo
- Base di conteggio
- Codifica degli stati

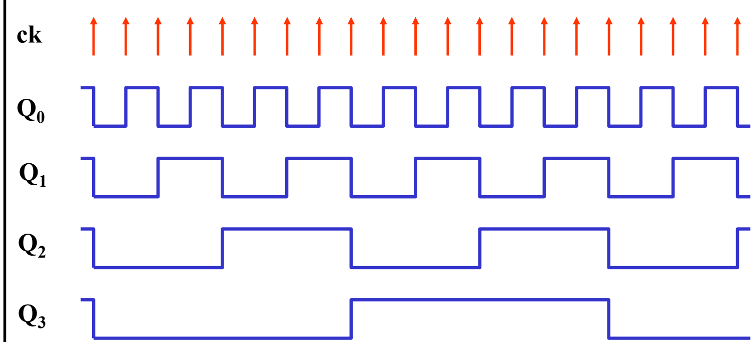
Il contatore binario x16



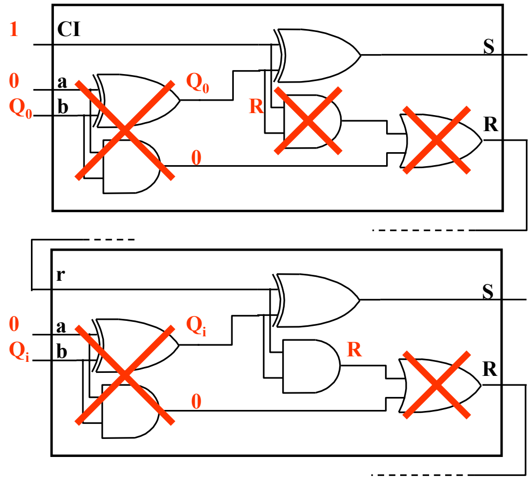
s^n	s^{n+1}
0000	0001
0001	0010
0010	0011
0011	0100
0100	0101
0101	0110
0110	0111
0111	1000
1000	1001
1001	1010
1010	1011
1011	1100
1100	1101
1101	1110
1110	1111
1111	0000

Forme d’onda

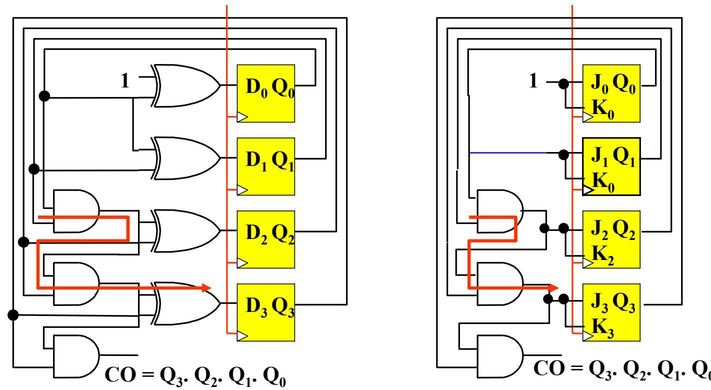
In un contatore binario, l’uscita Q del flip-flop che memorizza il bit di peso 2^i è un’onda quadra con periodo doppio di quella presente sull’uscita Q del flip-flop che memorizza il bit di peso 2^{i-1}. Il primo flip-flop divide per due la frequenza del clock.



Una rete più semplice per l'incremento



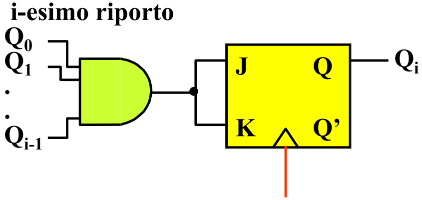
Periodo del clock e complessità della rete



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

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

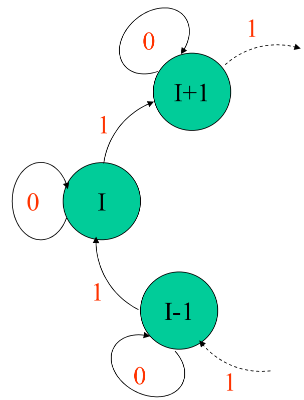
Un contatore ancora più veloce

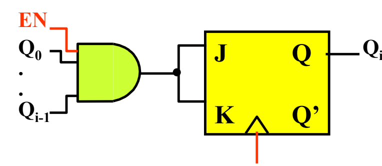


$T_0 \geq \tau_r + \tau_g + \tau_{su}$

N.B. - In realtà il ritardo di propagazione τ_g aumenta al crescere del fan-in del gate.

Il comando di ENABLE

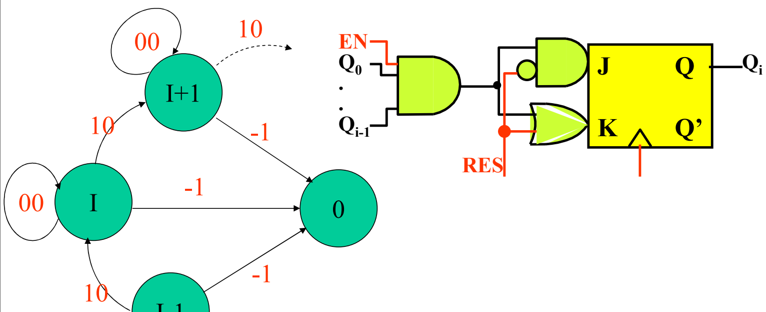




EN = 0: $S^{n+1} = S^n$

EN = 1: $S^{n+1} = (S+1)^n \bmod 2^n$

ENABLE & RESET



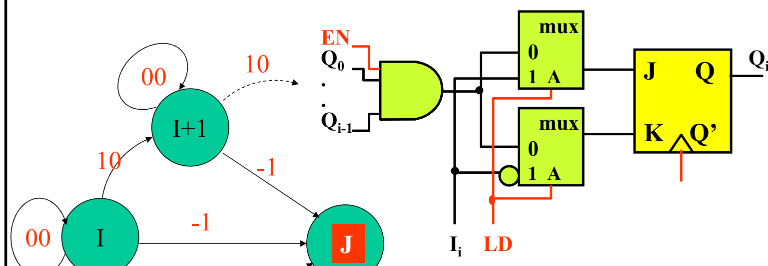
RES = 1: $S^{n+1} = 0$

RES = 0 e

EN = 0: $S^{n+1} = S^{n}$

EN = 1: $S^{n+1} = (S+1)^{n} \bmod 2^{n}$

ENABLE & LOAD



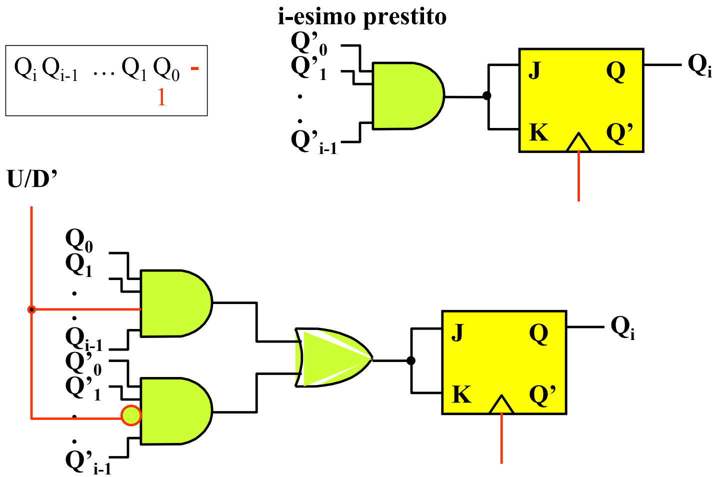
LD = 1: $S^{n+1} = J$

LD = 0 e

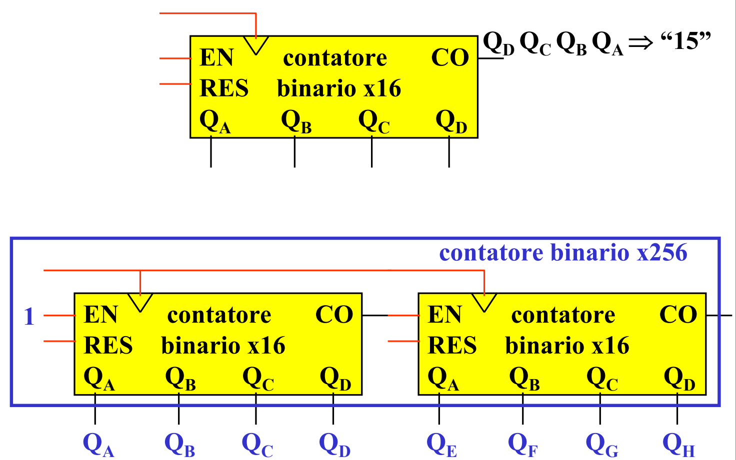
EN = 0: $S^{n+1} = S^{n}$

EN = 1: $S^{n+1} = (S+1)^{n} \bmod 2^{n}$

Il contatore Up/Down (U/D')



Aumento della base di conteggio

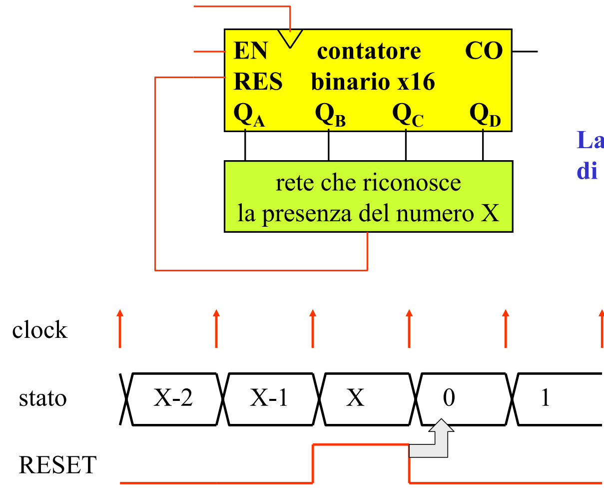


Disposizione in cascata di contatori

$EN_i = \textcolor{red}{EN} . CO_{i-1}$

La disposizione in cascata di **n** moduli di conteggio,
rispettivamente con base **B1**, **B2**,..,**Bn**,
fornisce un contatore con base
B = B1 x B2 x ……Bn

Diminuzione della base di conteggio



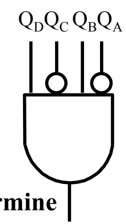
La nuova base di conteggio è X + 1

Esempio: sintesi del RESET per ottenere base 11

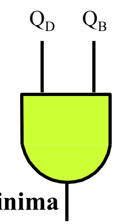
Q_B^n, Q_A^n

Q_D^n,Q_C^n	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	-	-	-	-
10	0	0	-	1

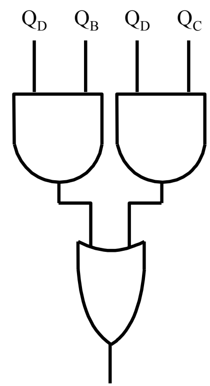
RES ^{n}



mintermine



rete minima

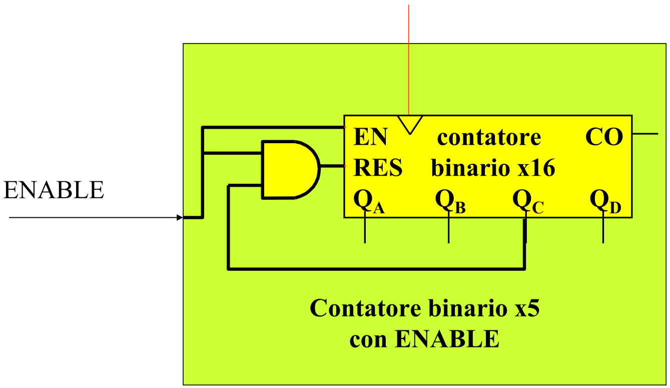


autoinizializzazione in un solo clock

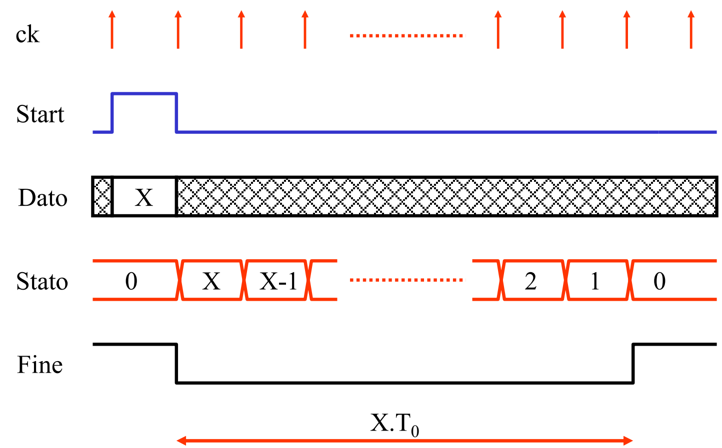
Rete minima per base X+1

Si connette al RESET un AND
avente in ingresso i soli bit
che hanno valore 1 in X

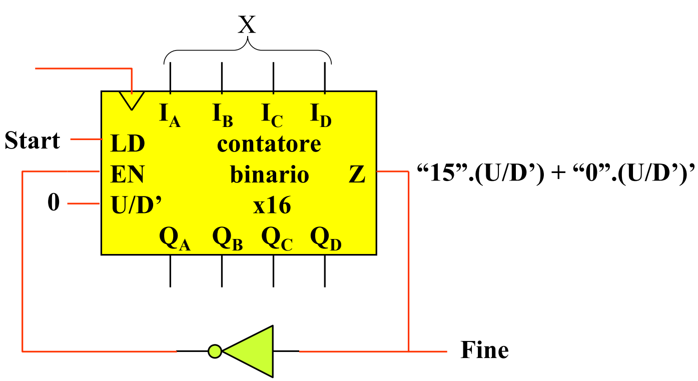
Esercizio



Watch dog: forme d'onda

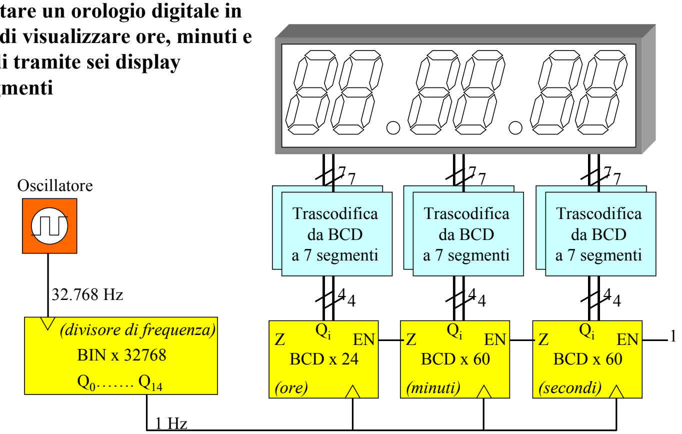


Watch dog: schema logico

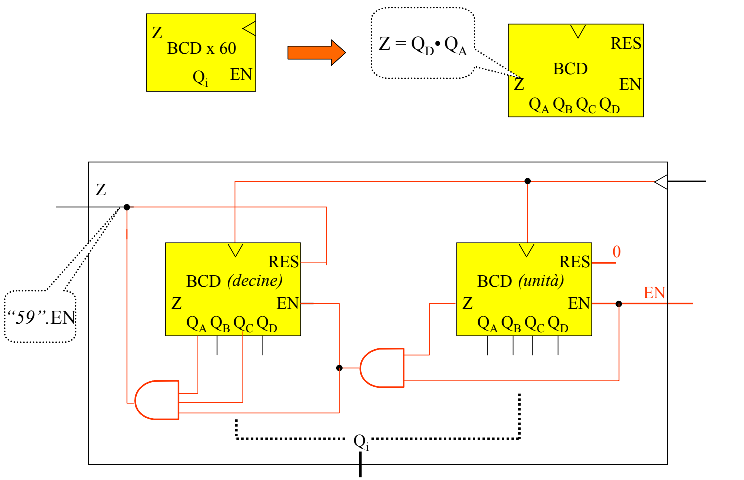


Orologio digitale

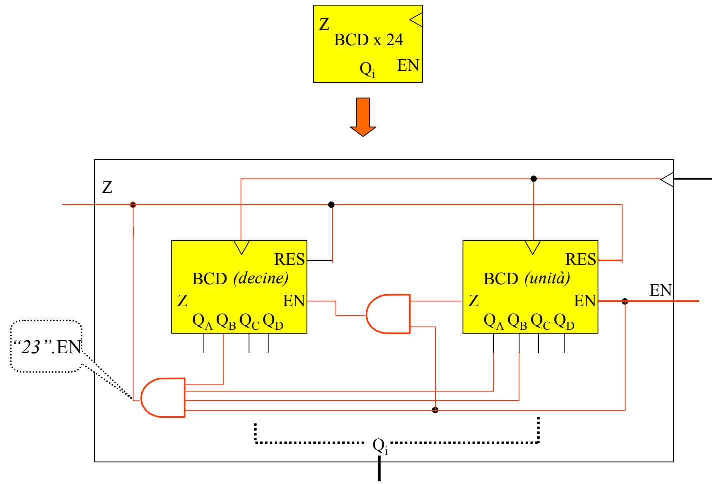
A partire da un oscillatore a 32.768 Hz progettare un orologio digitale in grado di visualizzare ore, minuti e secondi tramite sei display a 7 segmenti



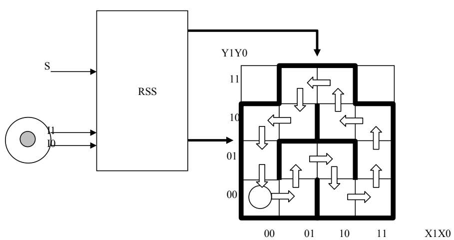
Il contatore dei minuti/secondi



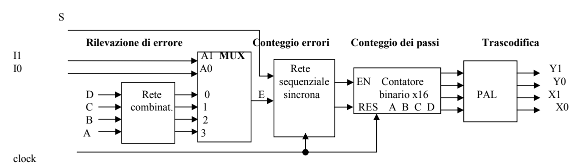
Il contatore delle ore



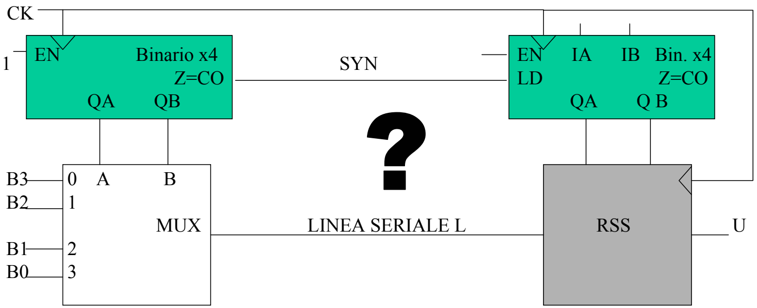
Esercitazione N. 18



Esercitazione N. 18



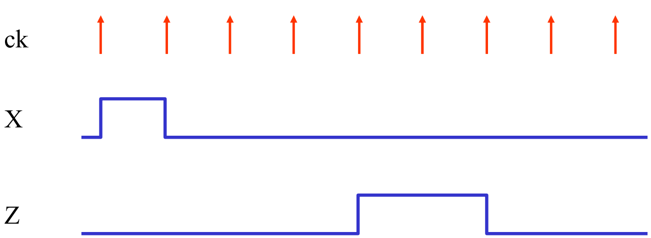
Esercitazione N. 19



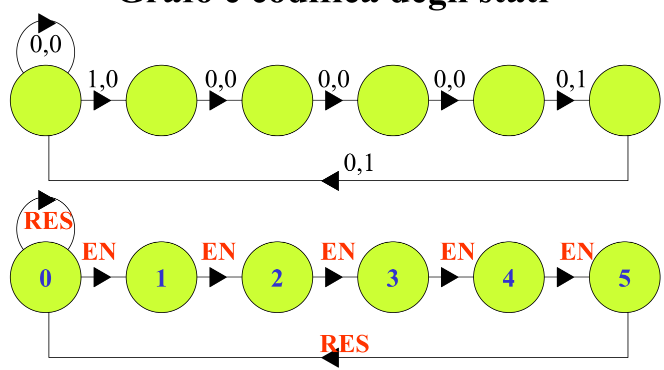
Sintesi con contatori

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
riposo	000	000	001	0
	001	010	---	0
	010	011	---	0
	011	100	---	0
	100	101	---	1
	101	000	---	1
	110	---	---	-
	111	---	---	-

→

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

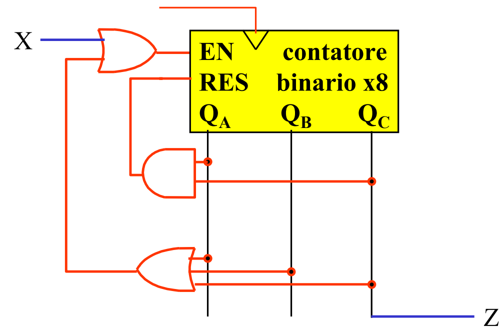
EN, RES, Z

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

$RES = Q_C\, Q_A$

$Z = Q_C$

I° esempio: schema logico



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

$RES = Q_C\, Q_A$

$Z = Q_C$

Esercitazione N. 20

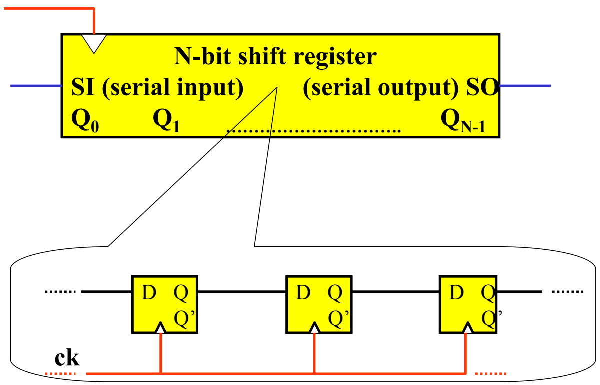
	$x_1^n x_2^n$			
Q^n	00	01	10	11
A	A,00	C,01	B,10	B,10
B	D,00	D,00	B,10	D,00
C	A,00	C,01	A,00	A,00
D	D,00	C,01	B,10	C,01

Q^{n+1}, $\mathbf{s_1^n s_2^n}$

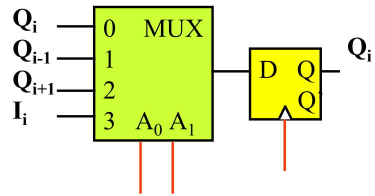
I registri a scorrimento

Shift register

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



Universal shift register

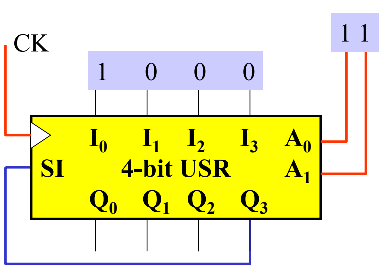


	A_0^n	A_1^n	Q_i^{n+1}
hold	0	0	Q_i^n
right	1	0	Q_{i-1}^n
left	0	1	Q_{i+1}^n
load	1	1	I_i^n

APPLICAZIONI

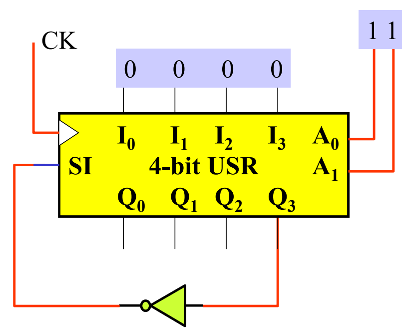
- **linea di ritardo**
- **convertitore S/P e P/S**
- **conteggio**
- **memoria a circolazione**
- **rotazione verso destra/sinistra**
- **moltiplicazione/divisione per 2^i**

Contatore ad anello



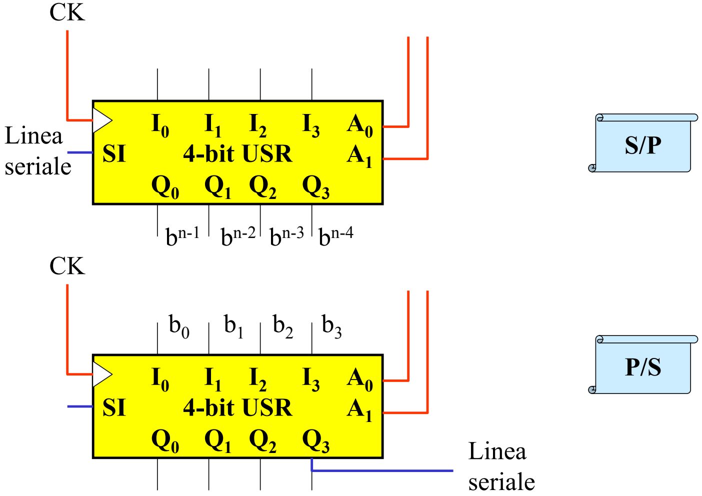
Base: 4 Codice: 1 su 4
1000
0100
0010
0001

Contatore a "riempimento/svuotamento"



Base: 8 Codice: Johnson
0000
1000
1100
1110
1111
0111
0011
0001

Esempio: conversioni S/P e P/S



Esercitazione N. 21