

## Registri: contatori

Prof. Daniele Gorla

## Contatori

Un *contatore* è un registro usato per contare il numero di occorrenze di un determinato evento, sempre modulo un certo numero naturale.

→ se formato da  $n$  FF, potrà contare fino a modulo  $2^n$

Tipicamente, gli eventi che può contare sono i colpi di clock o le occorrenze di alcuni valori (o sequenze) di input.

Si distinguono in:

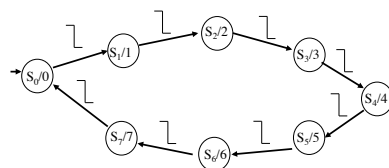
- sincroni (tutti i FF del contatore hanno lo stesso clock)
- asincroni (nello stesso contatore i vari FF hanno clock diversi)

Possono contare a salire o a scendere (o entrambe)

Possono essere settabili (forzare un valore che non rispetta la normale sequenza)

## Sintesi del contatore modulo 8 (1)

Il *contatore modulo 8* parte da 0 e ad ogni fronte d'onda discendente del clock incrementa di 1 il suo valore, fino ad arrivare a 7; poi torna a 0 e ricomincia.



Codifica dell'automa:

- Associa allo stato  $S_i$  la codifica binaria di  $i \rightarrow 3$  bit  $\rightarrow 3$  FF
- Non c'è alfabeto di input
- I caratteri di output vengono codificati con la loro normale codifica binaria

## Sintesi del contatore modulo 8 (2)

Stato(t)		Stato(t+1)	
$S_0$	$S_1$	$S_0$	$S_1$
$S_1$	$S_2$	$S_1$	$S_2$
$S_2$	$S_3$	$S_2$	$S_3$
$S_3$	$S_4$	$S_3$	$S_4$
$S_4$	$S_5$	$S_4$	$S_5$
$S_5$	$S_6$	$S_5$	$S_6$
$S_6$	$S_7$	$S_6$	$S_7$
$S_7$	$S_0$	$S_7$	$S_0$

$y_2$	$y_1$	$y_0$	$Y_2$	$Y_1$	$Y_0$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	1	0	-	0	-	1	-
0	0	1	0	1	0	0	-	1	-	-	1
0	1	0	0	1	1	0	-	-	0	1	-
0	1	1	1	0	0	1	-	-	1	-	1
1	0	0	1	0	1	-	0	0	-	1	-
1	0	1	1	1	0	-	0	1	-	-	1
1	1	0	1	1	1	-	0	-	0	1	-
1	1	1	0	0	0	-	1	-	1	-	1

$y_2$	$y_1$	$y_0$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
00	0	1	X	X	X	X	X	X
01	0	1	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X	X

$y_2$	$y_1$	$y_0$	$J_1$	$K_1$	$J_0$	$K_0$
00	0	1	X	X	X	X
01	0	1	X	X	X	X
11	0	1	X	X	X	X
10	X	X	X	X	X	X

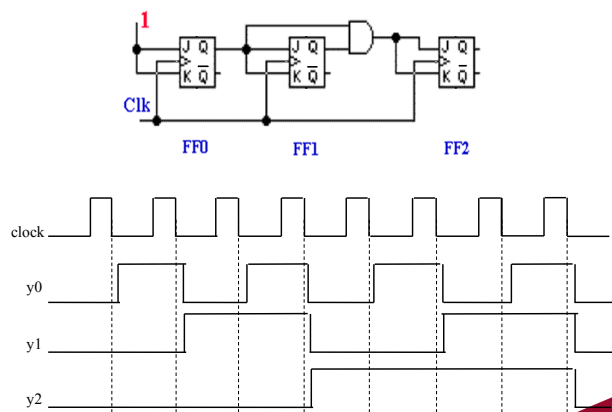
  

$y_2$	$y_1$	$y_0$	$J_0$	$K_0$
00	0	1	X	X
01	0	1	X	X
11	0	1	X	X
10	X	X	X	X

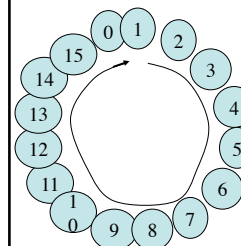
  

$J_0 = K_0 = 1$   
 $J_1 = K_1 = y_0$   
 $J_2 = K_2 = y_1 y_0$

## Il contatore modulo 8



## Sintesi del contatore modulo 16



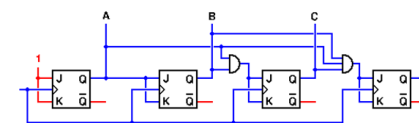
y3 y2 y1 y0	Y3 Y2 Y1 Y0	J3 K3	J2 K2	J1 K1	J0 K0
0 0 0 0	0 0 0 1	0 -	0 -	0 -	1 -
0 0 0 1	0 0 1 0	0 -	0 -	1 -	- 1
0 0 1 0	0 0 1 1	0 -	0 -	- 0	- 1
0 0 1 1	0 1 0 0	0 -	1 -	- 1	- 1
0 1 0 0	0 1 0 1	0 -	- 0	- 0	- 1
0 1 0 1	0 1 1 0	0 -	- 0	1 -	- 1
0 1 1 0	0 1 1 1	0 -	- 0	- 0	1 -
0 1 1 1	1 0 0 0	1 -	- 1	- 1	- 1
1 0 0 0	1 0 0 1	- 0	0 -	- 0	- 1
1 0 0 1	1 0 1 0	- 0	0 -	1 -	- 1
1 0 1 0	1 0 1 1	- 0	0 -	- 0	1 -
1 0 1 1	1 1 0 0	- 0	1 -	- 1	- 1
1 1 0 0	1 1 0 1	- 0	- 0	- 0	1 -
1 1 0 1	1 1 1 0	- 0	- 0	- 1	- 1
1 1 1 0	1 1 1 1	- 0	- 0	- 0	1 -
1 1 1 1	0 0 0 0	- 1	- 1	- 1	- 1

$$J_0 = K_0 = 1$$

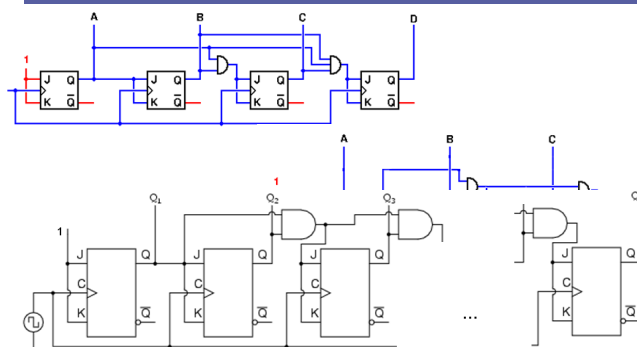
$$J_1 = K_1 = y_0$$

$$J_2 = K_2 = y_1 y_0$$

$$J_3 = K_3 = y_2 y_1 y_0$$

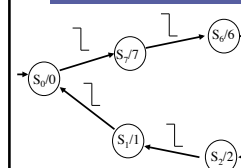


## Contatore modulo $2^n$



$$J_0 = K_0 = 1 \quad \text{e} \quad J_{i+1} = K_{i+1} = J_i \text{ AND } Q_i$$

## Sintesi del contatore alla rovescia modulo 8

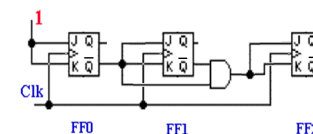


y2 y1 y0	Y2 Y1 Y0	J2 K2	J1 K1	J0 K0
0 0 0	1 1 1	1 -	1 -	1 -
0 0 1	0 0 0	0 -	0 -	- 1
0 1 0	0 0 1	0 -	- 1	1 -
0 1 1	0 1 0	0 -	- 0	- 1
1 0 0	0 1 1	- 1	1 -	1 -
1 0 1	1 0 0	- 0	0 -	- 1
1 1 0	1 0 1	- 0	- 1	1 -
1 1 1	1 1 0	- 0	- 0	- 1

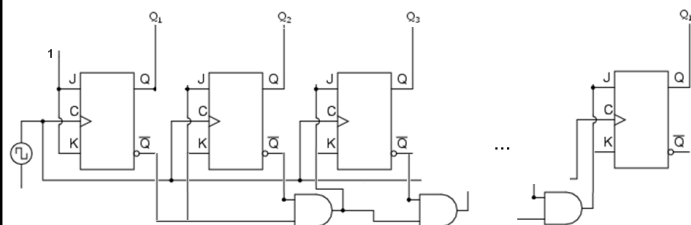
$$J_0 = K_0 = 1$$

$$J_1 = K_1 = \bar{y}_0$$

$$J_2 = K_2 = \bar{y}_1 \bar{y}_0$$



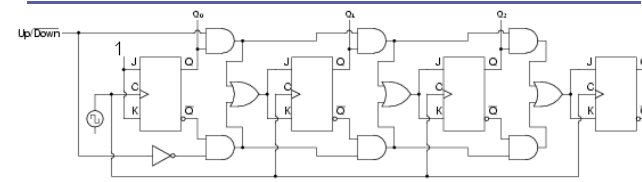
### Contatore alla rovescia modulo $2^n$



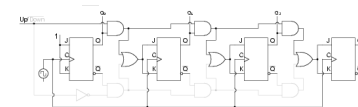
$$J_0 = K_0 = 1 \quad \text{e} \quad J_{i+1} = K_{i+1} = J_i \text{ AND } \bar{Q}_i$$

9

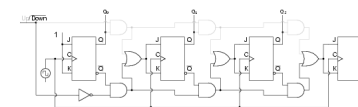
### Contatore bidirezionale modulo $2^n$



Contatore ascendente: Up = 1



Contatore discendente: Up = 0



10

### Contatore modulo $k$ ( $k \neq 2^n$ )



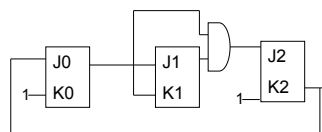
2 strade: 1. procedura di sintesi per ogni  $k$   
2. soluzione modulare, ma usando FF con ingressi asincroni (vedi dopo)

Es.: contatore modulo 5

$y_2 y_1 y_0$	$Y_2 Y_1 Y_0$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
0 0 0	0 0 1	0 -	0 -	1 -
0 0 1	0 1 0	0 -	1 -	- 1
0 1 0	0 1 1	0 -	- 0	1 -
0 1 1	1 0 0	1 -	- 1	- 1
1 0 0	0 0 0	- 1	0 -	0 -
1 0 1	- - -	- - - -	- - - -	- - - -
1 1 0	- - -	- - - -	- - - -	- - - -
1 1 1	- - -	- - - -	- - - -	- - - -

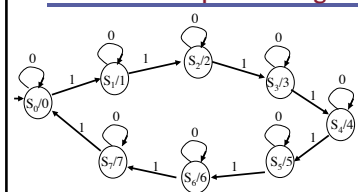
Stato(t)	Stato(t+1)
$S_0$	$S_1$
$S_1$	$S_2$
$S_2$	$S_3$
$S_3$	$S_4$
$S_4$	$S_0$

$$\begin{aligned} J_0 &= \bar{y}_2 \\ K_0 &= 1 \\ J_1 &= K_1 = y_0 \\ J_2 &= y_1 y_0 \\ K_2 &= 1 \end{aligned}$$



11

### Contatore di impulsi sull'ingresso



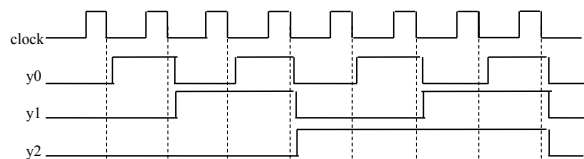
$x y_2 y_1 y_0$	$Y_2 Y_1 Y_0$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
0 0 0 0	0 0 0	0 -	0 -	0 -
0 0 0 1	0 0 1	0 -	0 -	0 -
0 0 1 0	0 1 0	0 -	- 0	0 -
0 0 1 1	0 1 1	0 -	- 0	0 -
0 1 0 0	1 0 0	- 0	- 0	- 0
0 1 0 1	1 0 1	- 0	- 0	- 0
0 1 1 0	1 1 0	- 0	- 0	- 0
0 1 1 1	1 1 1	- 0	- 0	- 0
1 0 0 0	0 0 0	0 -	0 -	1 -
1 0 0 1	0 0 1	0 -	1 -	- 1
1 0 1 0	0 1 1	0 -	- 0	1 -
1 0 1 1	1 0 0	1 -	- 1	- 1
1 1 0 0	1 0 1	- 0	0 -	1 -
1 1 0 1	1 1 0	- 0	1 -	- 1
1 1 1 0	1 1 1	- 0	- 0	1 -
1 1 1 1	0 0 0	- 1	- 1	- 1

$$\begin{aligned} J_0 &= K_0 = x1 \\ J_1 &= K_1 = x y_0 \\ J_2 &= K_2 = x y_1 y_0 \end{aligned}$$

La stessa del contatore modulo 8

12

### Contatore asincrono MOD 8



OSS.: il FF0 commuta ad ogni fronte d'onda discendente del clock;  
il FF1 commuta ad ogni fronte d'onda discendente del FF0;  
il FF2 commuta ad ogni fronte d'onda discendente del FF1.

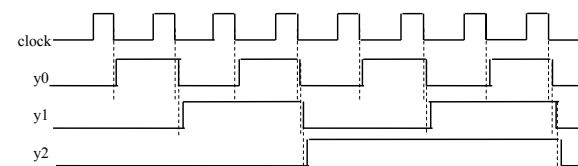
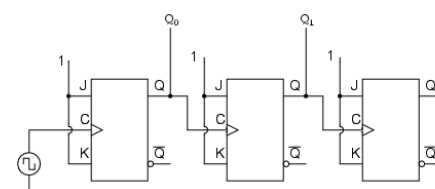
Possiamo progettare un diverso contatore MOD 8 in cui

- tutti i FF sono in uno stato di toggle ( $J = K = 1$ )
- il FF0 ha come clock il segnale *clock*;
- il FF1 ha come clock  $y_0$ ;
- il FF2 ha come clock  $y_1$ .

Chiameremo *asincrono* tale contatore poiché i FF non sono sincronizzati sullo stesso clock.

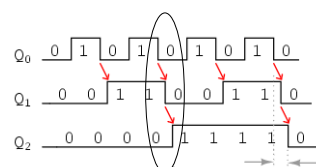
13

### Realizzazione e diagramma del contatore asincrono MOD 8



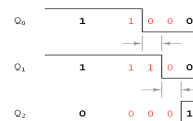
14

### Accumulo di ritardo



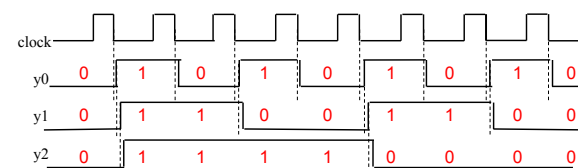
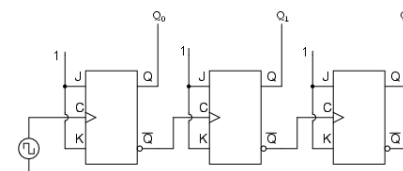
Poiché i FF non sono sincronizzati sullo stesso clock, i ritardi di commutazione si sommano e danno vita, per pochi istanti, a sequenze fuori dalla numerazione (*false counts*). Questo effetto si chiama *ripple* (= mormorio).

In quasi tutte le applicazioni, questo effetto è trascurabile, perché i ritardi sono piccolissimi.



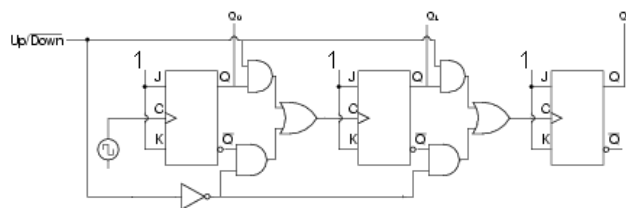
15

### Contatore asincrono MOD 8 alla rovescia



16

### Contatore asincrono bidirezionale

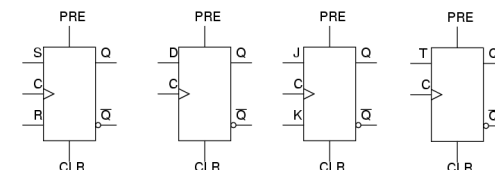


17

### FF con ingressi asincroni



Spesso, i FF sono equipaggiati con due ulteriori ingressi, chiamati PRESET e CLEAR, che funzionano in maniera *asincrona* rispetto al clock: essi cioè sono usati per settare o resettare il FF in modo istantaneo, cioè indipendentemente dalle entrate usuali e dal *clock*.



Funzionamento:

- PRESET = CLEAR = 0: normale funzionamento del FF;
- PRESET = 1, CLEAR = 0: set immediato del FF;
- PRESET = 0, CLEAR = 1: reset immediato del FF;
- PRESET = CLEAR = 1: non usata.

18

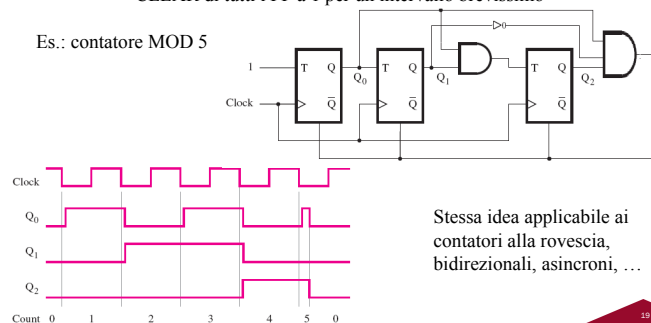
### Progetto modulare di un contatore MOD $k$ ( $\neq 2^n$ )



Idea: nel passare da  $k-1$  a  $k$ , forzo un reset del contatore tramite i segnali CLEAR

→ appena il contatore contiene (la codifica binaria di)  $k$ , si mette il CLEAR di tutti i FF a 1 per un intervallo brevissimo

Es.: contatore MOD 5



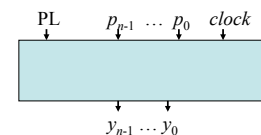
Stessa idea applicabile ai contatori alla rovescia, bidirezionali, asincroni, ...

19

### Contatori preselezionabili



Un secondo utilizzo dei FF con ingressi asincroni è la realizzazione di *contatori preselezionabili*, in cui cioè si può forzare un valore e mantenerlo, indipendentemente dai valori in ingresso e dal clock.



Funzionamento:

- Se il segnale PL (= *parallel load*) vale 0, allora si comporta come un normale contatore MOD  $2^n$ ;
- Se PL = 1, forza i valori presenti sulle linee  $p_{n-1}, \dots, p_0$  (*parallel data inputs*) nei rispettivi FF;
- Per mantenere memorizzato un valore, basta tenere tale valore sulle linee di input parallele e PL = 1.

20

