

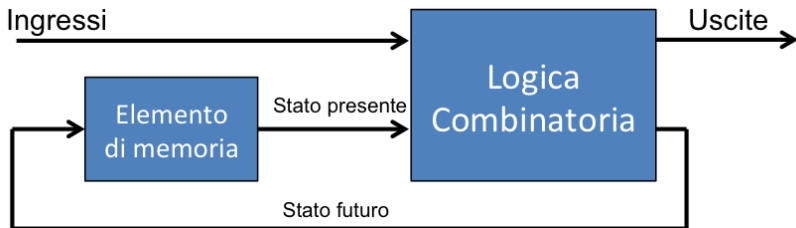
# LSS 2016-17

## Reti Logiche: circuiti sequenziali

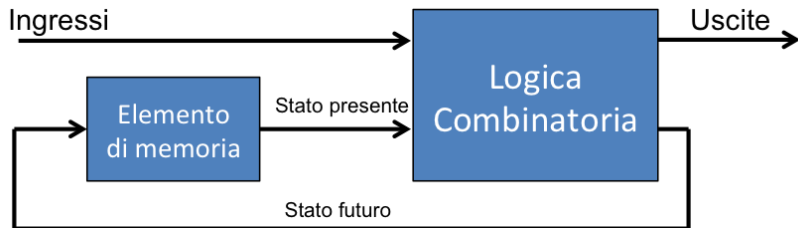
Piero Vicini

A.A. 2017-2018

- L'output di un circuito combinatorio e' *solo* funzione del valore combinatorio degli ingressi
- L'output di un circuito *sequenziale* dipende anche *dalla storia passata* del circuito i.e. da stati precedenti
- In generale possiamo dire che la funzione calcolata a tempo  $T$  dipende dalla sequenza temporale degli stati precedenti del circuito



- L'output di un circuito combinatorio e' *solo* funzione del valore combinatorio degli ingressi
- L'output di un circuito *sequenziale* dipende anche *dalla storia passata* del circuito i.e. da stati precedenti
- In generale possiamo dire che la funzione calcolata a tempo  $T$  dipende dalla sequenza temporale degli stati precedenti del circuito

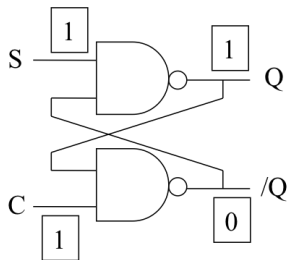
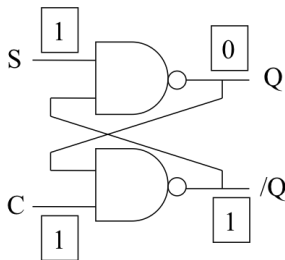


- E' necessario quindi un *elemento di memoria*
- Per cominciare vedremo elementi di memoria costruiti con porte logiche: *latch*

- Un *Latch S-R* dove (S e R hanno il significato di *Set* e *Reset*) e' **IL** circuito elettronico base degli elementi di memoria.
- La caratteristica principale e' quella di avere *2 stati stabili* per la stessa combinazione degli ingressi *condizione di riposo* con  $S = 1$  e  $C = 1$  (da cui il nome *flip-flop*)

NAND

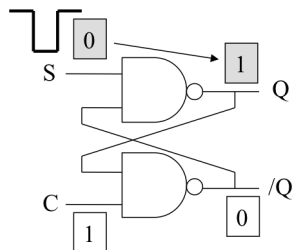
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0



- Di conseguenza possiamo dire che il latch *memorizza un valore* pari alla sua uscita.

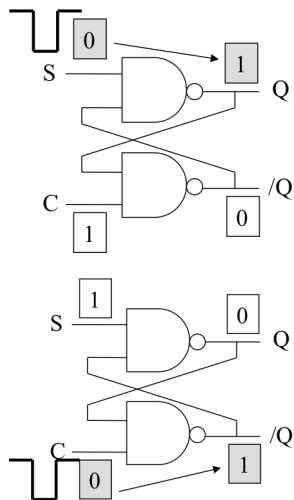
- Con quale meccanismo posso *scrivere* un valore nel Latch S-R?

- Con quale meccanismo posso *scrivere* un valore nel Latch S-R?



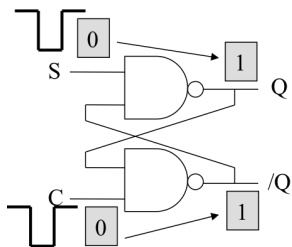
- **Set:** dato  $C = 1$ , se  $S = 0$  allora  $Q = 1$  (e  $\bar{Q} = 0$ ) qualunque valore fosse stato scritto in precedenza nel latch

- Con quale meccanismo posso *scrivere* un valore nel Latch S-R?



- **Set:** dato  $C = 1$ , se  $S = 0$  allora  $Q = 1$  (e  $\overline{Q} = 0$ ) qualunque valore fosse stato scritto in precedenza nel latch
- **Reset:** dato  $S = 1$ , se  $C = 0$  allora  $Q = 0$  (e  $\overline{Q} = 1$ ) qualunque valore fosse stato scritto in precedenza nel latch

- Che succede se  $S = 0$  e  $C = 0$  *contemporaneamente*?



- Condizione vietata!!!!
- $Q$  e  $\overline{Q}$  sono entrambi a 0 e lo stato e' *metastabile*
- Lo stato finale in condizione di riposo e' imprevedibile
- Dipende dal timing relativo di deasserzione degli ingressi e/o dalla struttura interna



## Clocked Latch (D-Latch)

- I segnali S e R devono essere stabili per memorizzare correttamente il valore nel Latch (i.e. non voglio la condizione di imprevedibilita' dello stato finale)
- In generale pero' S e R sono uscite di circuiti combinatori con un loro tempo di propagazione  $\neq 0$  e presenza di *glitch*.

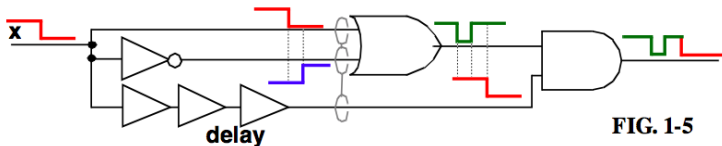


FIG. 1-5

## Clocked Latch (D-Latch)

- I segnali S e R devono essere stabili per memorizzare correttamente il valore nel Latch (i.e. non voglio la condizione di imprevedibilita' dello stato finale)
- In generale pero' S e R sono uscite di circuiti combinatori con un loro tempo di propagazione  $\neq 0$  e presenza di *glitch*.

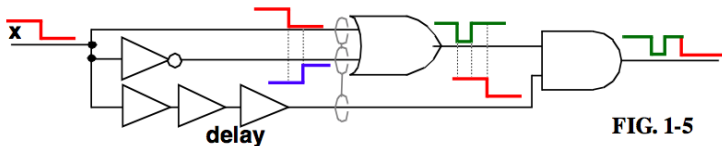
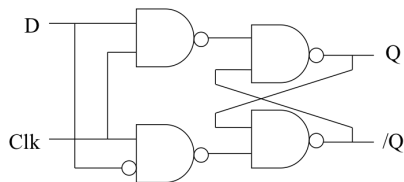


FIG. 1-5

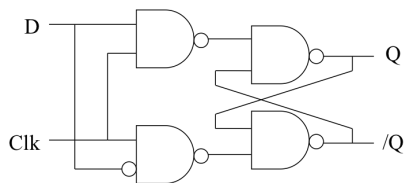
- In questo intervallo di tempo dobbiamo evitare di presentare i segnali S e R in ingresso al latch per evitare di scrivere valori non corretti
- Possiamo usare un *segnale di clock* per abilitare la scrittura nel latch.  $T_{clock} >$  del massimo tempo di propagazione dei circuiti combinatori che generano S e R

- Come possiamo correlare un segnale di clock con un latch?  
Realizziamo un D-Latch



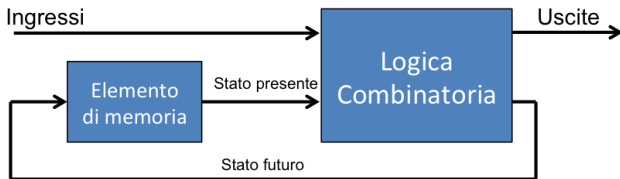
- $D = 1$  corrisponde al setting;
- $D = 0$  corrisponde al resetting
- Viene memorizzato il valore presente in  $D$  solo quando il clock e' asserito
- Non si puo' mai verificare la condizione di impredicibilita'

- Come possiamo correlare un segnale di clock con un latch?  
Realizziamo un D-Latch



- $D = 1$  corrisponde al setting;
- $D = 0$  corrisponde al resetting
- Viene memorizzato il valore presente in  $D$  solo quando il clock e' asserito
- Non si puo' mai verificare la condizione di impredicibilita'

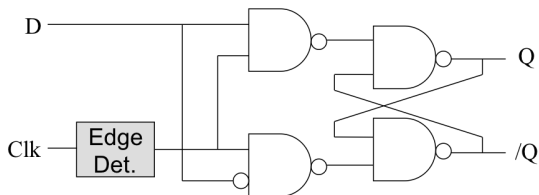
- Siamo soddisfatti? Non totalmente
- Problema della *trasparenza* del Latch:
- Durante il semiperiodo positivo del clock qualsiasi variazione di  $D$  si propaga in  $Q$  i.e. il latch non esercita nessuna funzione di memorizzazione....



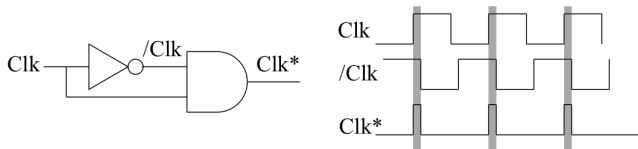
Durante ogni periodo di clock:

- Il circuito combinatorio di sopra dovrebbe calcolare una funzione sulla base dell'attuale valore dell'elemento di memoria (stato del circuito)
- l'output calcolato dovrebbe diventare il nuovo valore da memorizzare nell'elemento di memoria (nuovo stato del circuito)
- il nuovo valore memorizzato dovrebbe essere usato come input del circuito durante il ciclo di clock successivo
- D-Latch non lo permette....

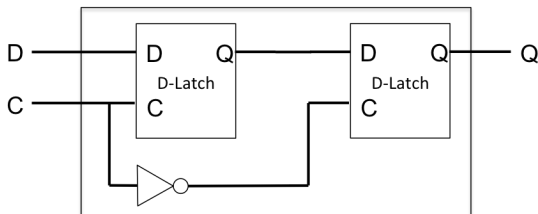
- Modifichiamo il latch per farlo commutare solo in presenza di *fronti* del clock
- Si distinguono in *positive edge triggered* e *negative edge triggered*



- La differenza principale con il D-Latch e' la presenza di un circuito di *edge detection*

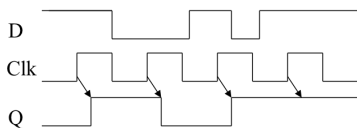


- Una struttura completa prevede un sistema a due stadi (*master-slave*)
- Ogni D-Latch e' attivo su un differente semiperiodo di clock -> risolto il problema della trasparenza



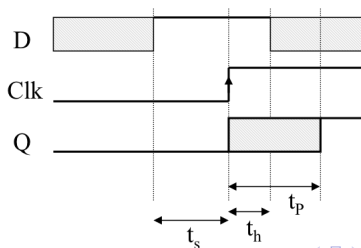
- Rispetto al flip-flop precedente, questo e' di tipo *falling triggered* in quanto l'uscita cambia sul fronte negativo di clock

# D-FlipFlop Timing



D	Clk	$Q_{n+1}$
0	↑	0
1	↑	1
X	↓	$Q_n$

- Esistono parametri elettrici caratteristici del componente
- $t_s$  *setup time*: periodo di tempo precedente la transizione del clock in cui l'ingresso D deve essere stabile.
- $t_h$  *hold time*: periodo di tempo seguente la transizione del clock in cui l'ingresso D deve essere stabile.
- $t_p$  *propagation time*: ritardo combinatorio del registro *clk to output*

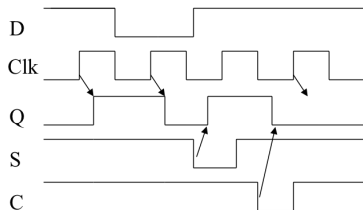




## DFF con Set e Reset asincroni

- Come garantiamo il comportamento corretto del DFF all'accensione?
- Due segnali *set* e *clear* che agiscono in maniera indipendente dal (e con priorit  sul) clock.
- Set attivo  $\rightarrow Q = 1$ , Clear attivo  $\rightarrow Q = 0$

D	Clk	S	C	$Q_{n+1}$
0	$\uparrow$	1	1	0
1	$\uparrow$	1	1	1
X	$\downarrow$	1	1	$Q_n$
X	X	0	1	1
X	X	1	0	0



## SN74LS74A

### Dual D-Type Positive Edge-Triggered Flip-Flop

The SN74LS74A dual edge-triggered flip-flop utilizes Schottky TTL circuitry to produce high speed D-type flip-flops. Each flip-flop has individual clear and set inputs, and also complementary Q and  $\bar{Q}$  outputs.

Information at input D is transferred to the Q output on the positive-going edge of the clock pulse. Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive-going pulse. When the clock input is at either the HIGH or the LOW level, the D input signal has no effect.

#### MODE SELECT - TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	$\bar{S}_D$	$S_D$	D	Q	$\bar{Q}$
Set	L	H	X	H	L
Reset (Clear)	H	L	X	L	H
*Undetermined	L	L	X	H	H
Load "1" (Set)	H	H	h	H	L
Load "0" (Reset)	H	H	l	L	H

\* Both outputs will be HIGH while both  $\bar{S}_D$  and  $\bar{C}_D$  are LOW, but the output states are unpredictable if  $\bar{S}_D$  and  $\bar{C}_D$  go HIGH simultaneously. If the levels at the set and clear are near  $V_{IL}$  maximum then we cannot guarantee to meet the minimum level for  $V_{OH}$ .

H, h = HIGH Voltage Level

L, l = LOW Voltage Level

X = Don't Care

l, h (q) = Lower case letters indicate the state of the referenced input

$t_{SU}(q)$  = (or output) one set-up time prior to the HIGH to LOW clock transition.



**ON Semiconductor**  
Formerly a Division of Motorola

<http://onsemi.com>

**LOW  
POWER  
SCHOTTKY**



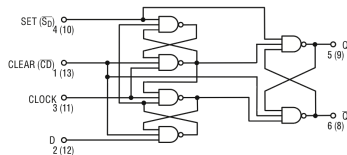
PLASTIC  
N SUFFIX  
CASE 646



SOIC  
D SUFFIX  
CASE 751A

## SN74LS74A

#### LOGIC DIAGRAM (Each Flip-Flop)



## DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$V_{IH}$	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
$V_{IL}$	Input LOW Voltage			0.8	V	Guaranteed Input LOW Voltage for All Inputs
$V_{IK}$	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN}$ , $I_{IN} = -18 \text{ mA}$
$V_{OH}$	Output HIGH Voltage	2.7	3.5		V	$V_{CC} = \text{MIN}$ , $I_{OH} = \text{MAX}$ , $V_{IN} = V_{IH}$ or $V_{IL}$ per Truth Table
$V_{OL}$	Output LOW Voltage		0.25	0.4	V	$I_{OL} = 4.0 \text{ mA}$
			0.35	0.5	V	$I_{OL} = 8.0 \text{ mA}$
$I_{IH}$	Input High Current Data, Clock Set, Clear			20 40	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{IN} = 2.7 \text{ V}$
	Data, Clock Set, Clear			0.1 0.2	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 7.0 \text{ V}$
$I_{IL}$	Input LOW Current Data, Clock Set, Clear			-0.4 -0.8	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 0.4 \text{ V}$
$I_{OS}$	Output Short Circuit Current (Note 1)	-20		-100	mA	$V_{CC} = \text{MAX}$
$I_{CC}$	Power Supply Current			8.0	mA	$V_{CC} = \text{MAX}$

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

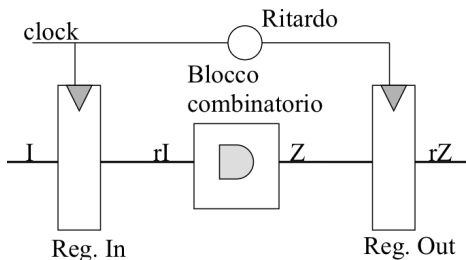
## AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ , $V_{CC} = 5.0 \text{ V}$ )

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$f_{MAX}$	Maximum Clock Frequency	25	33		MHz	Figure 1
$t_{PLH}$ $t_{PHL}$	Clock, Clear, Set to Output		13 25	25 40	ns	Figure 1
						$V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$

## AC SETUP REQUIREMENTS ( $T_A = 25^\circ\text{C}$ )

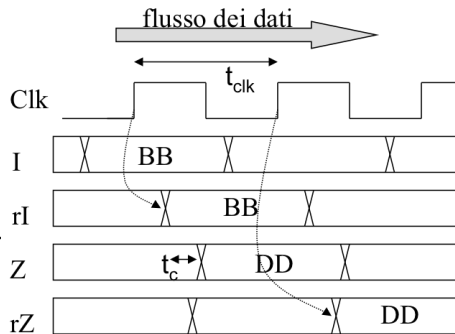
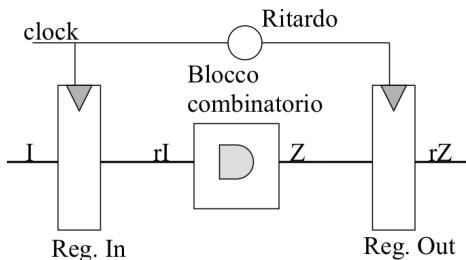
Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$t_{W(H)}$	Clock	25			ns	Figure 1
$t_{W(L)}$	Clear, Set	25			ns	Figure 2
$t_s$	Data Setup Time — HIGH	20			ns	Figure 1
	Clear, Set, Clock — LOW	20			ns	
$t_h$	Hold Time	5.0			ns	Figure 1

- I DFF sono normalmente raggruppati in *banchi di registri* dove il clock e' comune a tutto l'array di DFF.
- In questa configurazione sono fondamentali per le operazioni di *trasferimento parallelo sincrono*

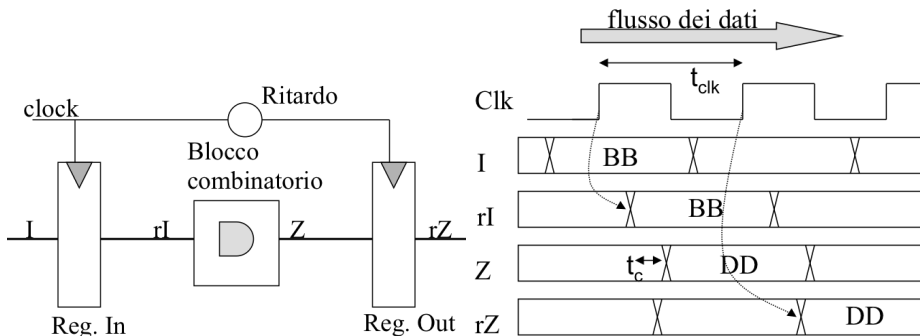


# Trasferimento parallelo

- I DFF sono normalmente raggruppati in *banchi di registri* dove il clock e' comune a tutto l'array di DFF.
- In questa configurazione sono fondamentali per le operazioni di *trasferimento parallelo sincrono*



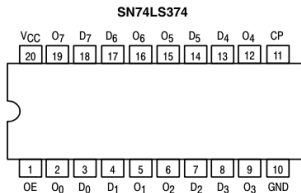
- I DFF sono normalmente raggruppati in *banchi di registri* dove il clock e' comune a tutto l'array di DFF.
- In questa configurazione sono fondamentali per le operazioni di *trasferimento parallelo sincrono*



- Warning: nell'ipotesi  $t_{skew} = 0$  il sistema funziona se:

$$t_h + t_p + t_c + t_s < t_{clk}$$

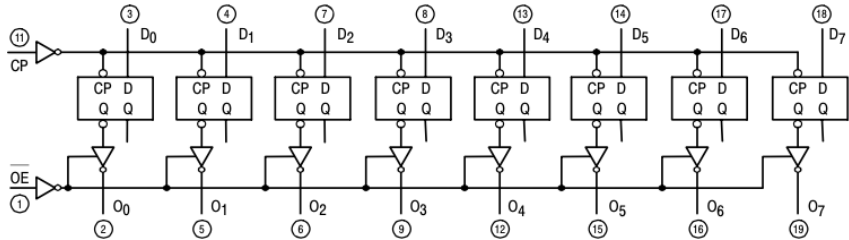
# Octal D-FlipFlop 74LS374



**LS374**

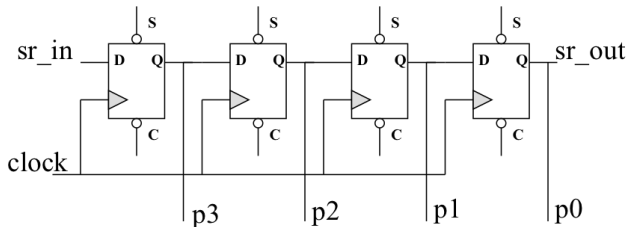
D <sub>n</sub>	LE	OE	O <sub>n</sub>
H		L	H
L		L	L
X	X	H	Z*

## SN74LS374



# Trasferimento seriale

- Lo *shift register* e' un circuito che realizza un trasferimento seriale sincrono
- Lo shift register puo' essere scritto serialmente (modo sincrono) o inizializzato con una opportuna combinazione di Set e Clear (modo asincrono)



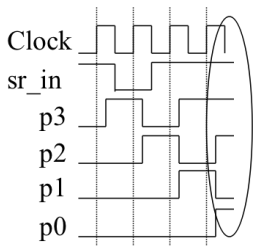
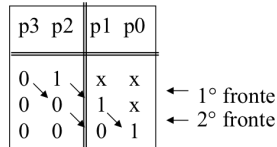
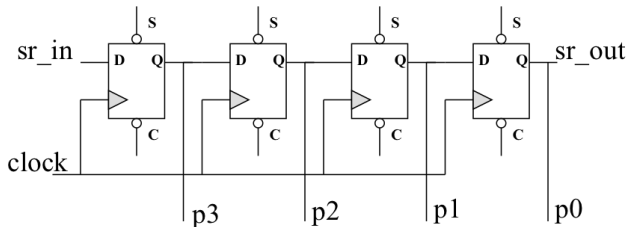
p3	p2	p1	p0
0	1	x	x
0	0	1	x
0	0	0	1

← 1° fronte  
← 2° fronte



# Trasferimento seriale

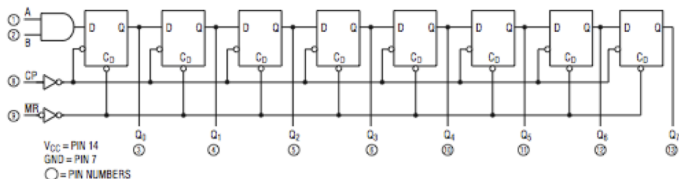
- Lo *shift register* e' un circuito che realizza un trasferimento seriale sincrono
- Lo shift register puo' essere scritto serialmente (modo sincrono) o inizializzato con una opportuna combinazione di Set e Clear (modo asincrono)



- Uno shift register implementa una conversione seriale-parallelo (e viceversa)

## SN74LS164

### LOGIC DIAGRAM



### FUNCTIONAL DESCRIPTION

The LS164 is an edge-triggered 8-bit shift register with serial data entry and an output from each of the eight stages. Data is entered serially through one of two inputs (A or B); either of these inputs can be used as an active HIGH Enable for data entry through the other input. An unused input must be tied HIGH, or both inputs connected together.

Each LOW-to-HIGH transition on the Clock (CP) input shifts data one place to the right and enters into  $Q_0$  the logical AND of the two data inputs (A•B) that existed before the rising clock edge. A LOW level on the Master Reset (MR) input overrides all other inputs and clears the register asynchronously, forcing all Q outputs LOW.

### MODE SELECT – TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	MR	A	B	$Q_0$	$Q_1-Q_7$
Reset (Clear)	L	X	X	L	L - L
Shift	H	l	l	L	$q_0 - q_7$
	H	l	h	L	$q_0 - q_6$
	H	h	l	L	$q_0 - q_6$
	H	h	h	H	$q_0 - q_6$

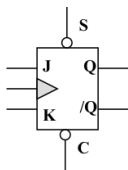
L (l) = LOW Voltage Levels

H (h) = HIGH Voltage Levels

X = Don't Care

$q_n$  = Lower case letters indicate the state of the referenced input or output one

- Un contatore e' un circuito sequenziale in grado di "contare" impulsi.
- Elemento essenziale di un contatore e' un FF che commuta il suo stato d'uscita ad ogni colpo di clock *toggle Flip-flop (TFF)*
- Il TFF si realizza a partire da un semplice JK (derivato da un Set-Reset standard)



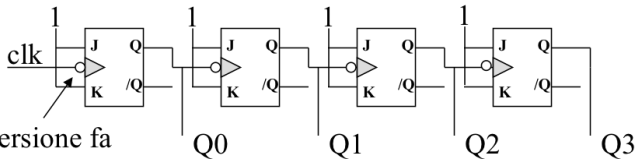
J	K	Q ↑
0	0	Q <sub>old</sub>
0	1	0
1	0	1
1	1	/Q <sub>old</sub>

Equivalgono ad  
un set e clear  
sincroni!!!

- Due classi di contatori
  - Contatori **asincroni** (*ripple counter*)
  - Contatori **sincroni** (*parallel counter*)

# Contatori asincroni (ripple)

- Il JK-FF con  $J = K = 1$  realizza un TFF
- Esempio di contatore a 4 bit:

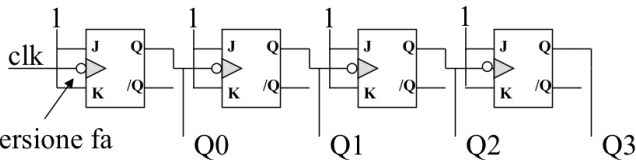


L' inversione fa  
commutare FF  
su fronte negativo

n°C	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
.....				
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

# Contatori asincroni (ripple)

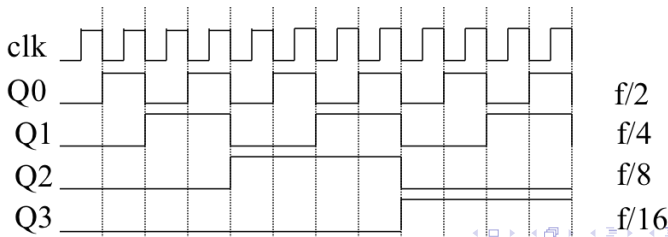
- Il JK-FF con  $J = K = 1$  realizza un TFF
- Esempio di contatore a 4 bit:



L'inversione fa commutare FF su fronte negativo

n°C	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
.....				
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

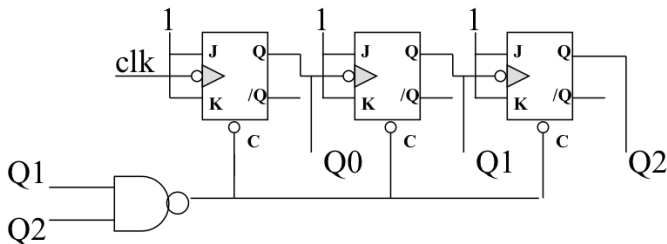
- Un contatore ripple agisce (anche) da divisore di frequenza...



# Contatori asincroni (ripple)

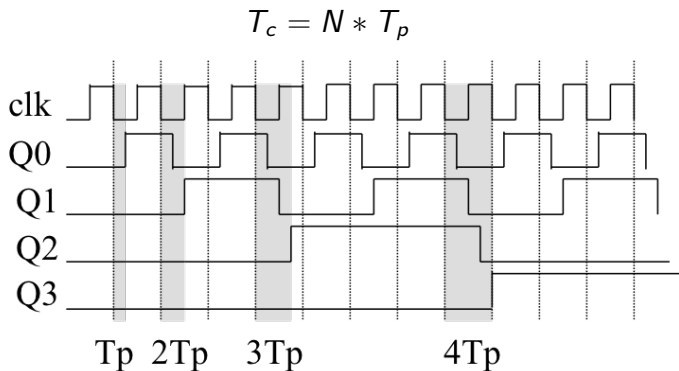
- Progetto di un contatore asincrono *modulo 6* a partire da un contatore *modulo 8*
- Il contatore deve essere *resettato* in maniera asincrona quando il suo output e' uguale a "6" (ovvero appena le sue uscite assumono il valore 110)

n°C	Q2	Q1	Q0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	0	0	0



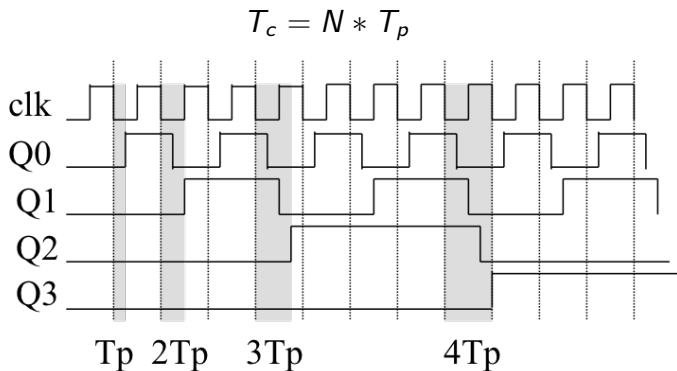
## Contatori asincroni (ripple)

- Quanto vale il *tempo di propagazione* di un contatore asincrono modulo  $2N$ ?
- Poiche' il clock dello stadio successivo e' il data output dello stadio precedente, dato  $T_p$  il tempo di propagazione del singolo stadio, il tempo di commutazione dal clock in ingresso al conteggio in uscita e'



## Contatori asincroni (ripple)

- Quanto vale il *tempo di propagazione* di un contatore asincrono modulo  $2N$ ?
- Poiche' il clock dello stadio successivo e' il data output dello stadio precedente, dato  $T_p$  il tempo di propagazione del singolo stadio, il tempo di commutazione dal clock in ingresso al conteggio in uscita e'



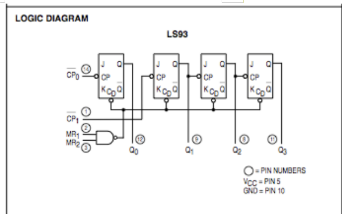
- Attenzione: notate che se  $T_c < N * T_p$  il contatore non funziona.....



# Contatori asincroni: 7493 Data sheet

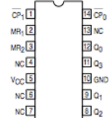
## 4-BIT BINARY COUNTER

SN54/74LS93



### CONNECTION DIAGRAM

DIP (TOP VIEW)



## LS92 AND LS93 MODE SELECTION

RESET INPUTS		OUTPUTS			
MR <sub>1</sub>	MR <sub>2</sub>	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
H	H	L	L	L	L
L	H		Count		
H	L		Count		
L	L		Count		

H = HIGH Voltage Level

L = LOW Voltage Level

X = Don't Care

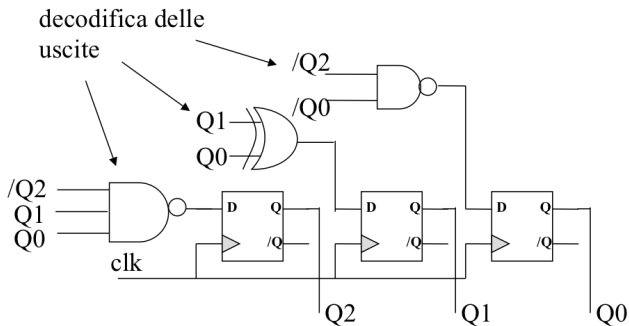
### AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ , $V_{CC} = 5.0\text{ V}$ , $C_L = 15\text{ pF}$ )

Symbol	Parameter	Limits									Unit
		LS90			LS92			LS93			
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
$f_{MAX}$	CP <sub>0</sub> Input Clock Frequency	32			32			32			MHz
$f_{MAX}$	CP <sub>1</sub> Input Clock Frequency	16			16			16			MHz
$t_{PLH}$ $t_{PHL}$	Propagation Delay, CP <sub>0</sub> Input to Q <sub>0</sub> Output		10 12	16 18		10 12	16 18		10 12	16 18	ns
$t_{PLH}$ $t_{PHL}$	CP <sub>0</sub> Input to Q <sub>3</sub> Output		32 34	48 50		32 34	48 50		46 46	70 70	ns
$t_{PLH}$ $t_{PHL}$	CP <sub>1</sub> Input to Q <sub>1</sub> Output		10 14	16 21		10 14	16 21		10 14	16 21	ns
$t_{PLH}$ $t_{PHL}$	CP <sub>1</sub> Input to Q <sub>2</sub> Output		21 23	32 35		10 14	16 21		21 23	32 35	ns
$t_{PLH}$ $t_{PHL}$	CP <sub>1</sub> Input to Q <sub>3</sub> Output		21 23	32 35		21 23	32 35		34 34	51 51	ns

## Contatori sincroni (parallel)

- I problemi di velocità connessi con l'accumularsi dei tempi di propagazione dei TFF della catena asincrona viene risolto con i contatori *sincroni*.
- Un contatore è detto sincrono quando tutti gli output commutano in maniera **sincrona con il clock in ingresso** i.e. il clock è distribuito **in parallelo** a tutti i FF del contatore
- Esempio: contatore modulo 5

Q2	Q1	Q0	Q2	Q1	Q0↑
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	0	0	0



- Per approfondire....
  - Tocci, R. J. *Digital System: principles and application*, Prentice-Hall
  - Horowitz, P. Hill, W. *The art of Electronics*, Cambridge University Press
  - Wakerly, J *Digital Design*, Prentice-Hall