



Sistemi di Elettronica Digitale, Sez.4

Alessandra Flammini
alessandra.flammini@unibs.it
Ufficio 24 Dip. Ingegneria dell'Informazione
030-3715627 Lunedì 16:30-18:30

Dispositivi MSI, Memorie



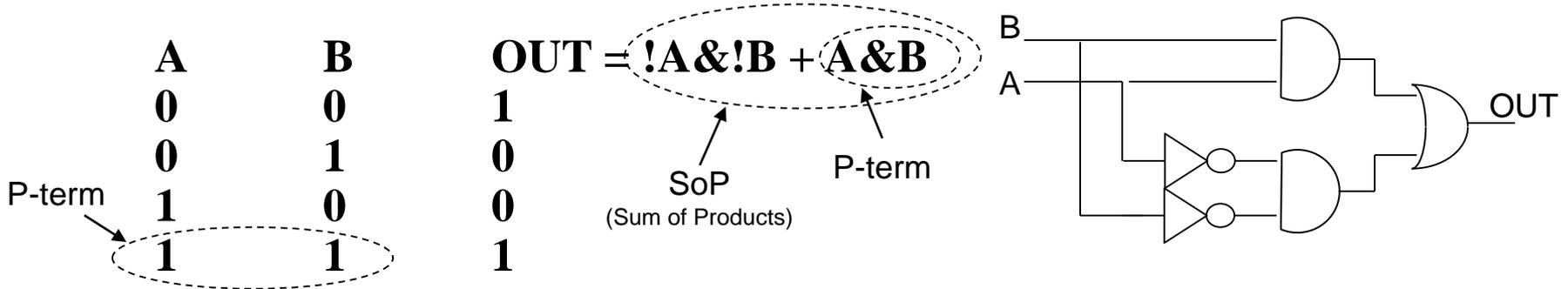
Dispositivi MSI (Medium Scale Integration)

- **Dispositivi SSI – "Gates"- (<200 transistors)**
 - Porte logiche (Buffer, NOT, AND, OR, NAND, NOR, XOR, AOI)
 - FAN-IN – numero di variabile tipicamente da 1 a 16 (max 32)
 - Assemblate in dispositivi a gruppi di 1-6
 - Inadatte per la descrizione/realizzazione di funzioni complesse
 - Oggi in disuso tranne che per i "Buffer"
- **Dispositivi MSI – "Registers"- (da 200 a 2000 transistors)**
 - Organizzati tipicamente a 4 bit
 - Dispositivi MSI combinatori
 - Convertitori di codice, Sommatore, Comparatori, Moltiplicatori,...
 - Dispositivi MSI sequenziali
 - Elementi di memoria, Contatori, Shift register, memorie
- **Modalità (alcune) di descrizione di un dispositivo digitale:**
 - **Layer:** strati di fabbricazione (processi fotolitografici)
 - **Transistor:** come sono connessi i transistor, ossia i source e i drain dei vari transistor o celle CMOS (nMOS+pMOS)
 - **Gate:** descrizione della funzione in termini di porte logiche (es. Tabella della verità, ossia funzioni in termini di somme di prodotti)
 - **RTL – Register Transfer Level-** descrizione mediante collegamento di blocchetti funzionali (tip. MSI)

Dispositivi MSI (Medium Scale Integration)

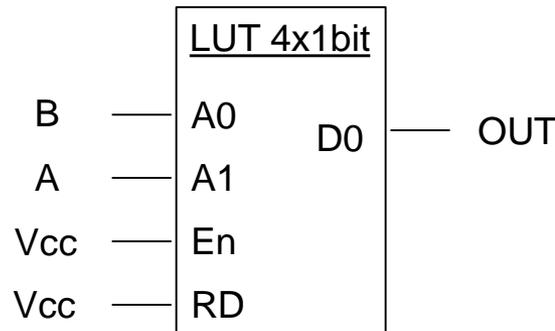
• Modalità di descrizione: Somma di Prodotti –SoP-

- Le funzionalità sono descritte mediante Tabella della verità (SoP)
- Una tabella della verità è una "Look-up Table" (LUT) realizzabile mediante una memoria



- La rappresentazione circuitale ha ritardi variabili tra ingressi e uscita
- La realizzazione a Look-up Table ha ritardo fisso, pari al tempo di lettura della memoria, per ogni funzione logica. N ingressi -> memoria a 2^N bit

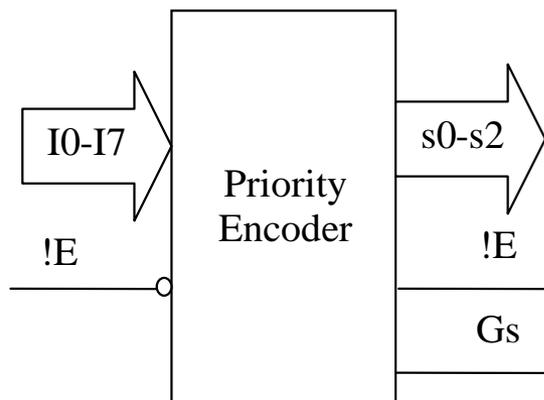
Indirizzo	Dato
0 (A=0, B=0)	1
1 (A=0, B=1)	0
2 (A=1, B=0)	0
3 (A=1, B=1)	1



(es. 32 IN -> 4Gbit!
-> FAN-IN limitato)

Dispositivi combinatori

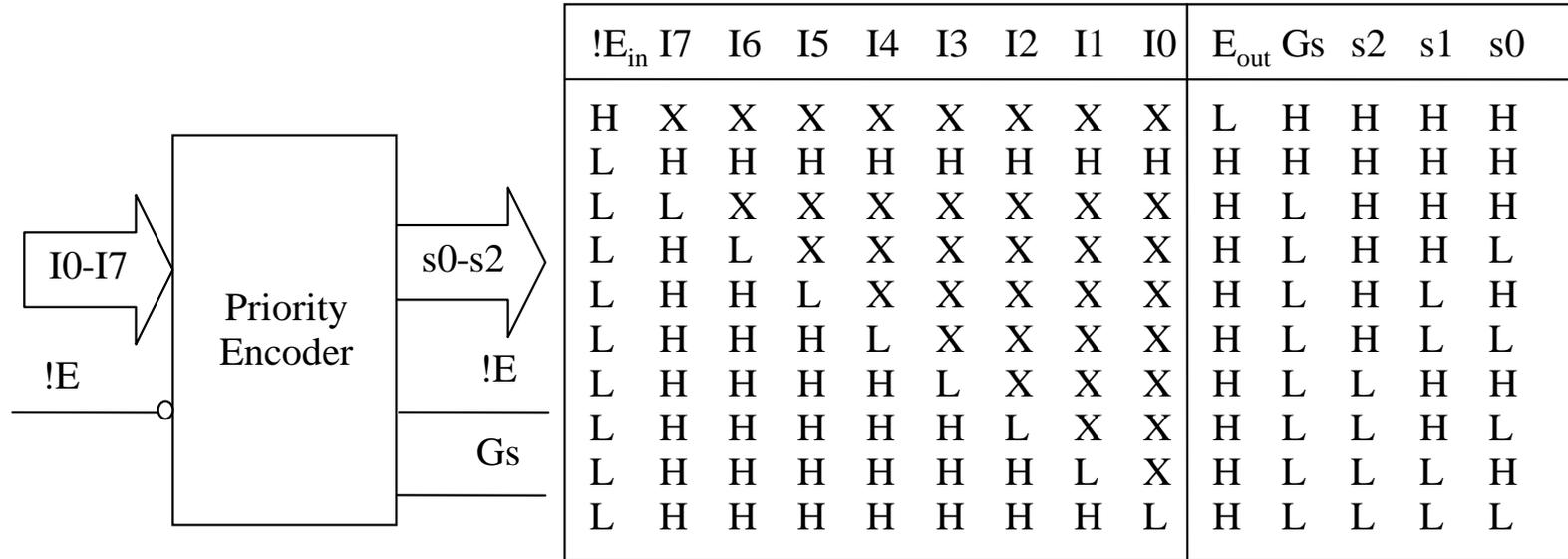
- Encoders/Decoders (convertitori di codice)
- **DECODERS: convertitori di codice da “compatto” a “espanso”**
 - I DEMUX 3 a 8 sono binary to octal decoders (74XX138)
 - Conversione BCD (4 linee) verso codice per led a 7 segmenti (74XX47)
- **ENCODERS: convertitori di codice da “espanso” a “compatto”**
 - I MUX possono essere considerati come convertitori di codice
 - Conversione da decimale a BCD (4 linee) (74XX147)
 - Convertendo da “espanso” a “compatto” devono gestire conflitti (priority encoders, 74XX148, tipicamente utilizzato per la gestione interrupt, vedi figura)



!E	I7	I6	I5	I4	I3	I2	I1	I0	E	Gs	s2	s1	s0
H	X	X	X	X	X	X	X	X	L	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	X	X	X	X	X	X	X	H	L	H	H	H
L	H	L	X	X	X	X	X	X	H	L	H	H	L
L	H	H	L	X	X	X	X	X	H	L	H	L	H
L	H	H	H	L	X	X	X	X	H	L	L	H	L
L	H	H	H	H	L	X	X	X	H	L	L	H	H
L	H	H	H	H	H	L	X	X	H	L	L	L	H
L	H	H	H	H	H	H	H	L	H	L	L	L	L

Dispositivi combinatori

• Priority Encoder, 74XX148 (Esempio: gestione interrupt)

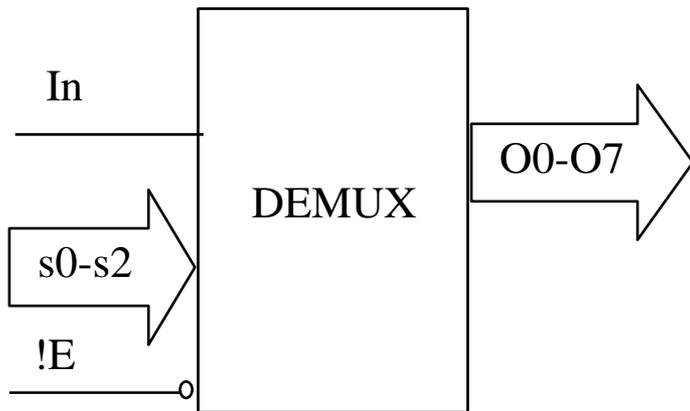


• La Tabella ha N ingressi, descritti in M righe, e K uscite

- Ciascun ingresso può assumere gli stati: 0, 1, X (indifferente) o essere funzione
- I termini prodotto (P-term) sono le uscite di porte AND con FAN-IN=N
- Il termine SoP che descrive ciascuna delle K uscite è l'uscita di una porta OR con FAN-IN=M (in genere il FAN-IN della porta OR è la principale criticità)
- $!E_{out} = (!E_{in})$ ($E_{out} = E_{in}$)
- $Gs = !E_{in} + E_{in} \& I7 \& I6 \& I5 \& I4 \& I3 \& I2 \& I1 \& I0$
- $!s2 = E_{in} \& I7 \& I6 \& I5 \& I4 \& !I3 + E_{in} \& I7 \& I6 \& I5 \& I4 \& I3 \& I2 \& I1 \& !I0 + E_{in} \& I7 \& I6 \& I5 \& I4 \& I3 \& !I2 + E_{in} \& I7 \& I6 \& I5 \& I4 \& I3 \& I2 \& !I1$

Dispositivi combinatori: demultiplexer

- Mediante le linee s_i si seleziona l'uscita O_j sulla quale, se $!E="0"$, viene convogliato l'ingresso I
 - L'ingresso può essere implicito (tip. "0") e le uscite non selezionate si trovano al livello opposto (tip. "1")
 - Possono essere da 3 a 8 linee (74XX138) o duali da 2 a 4 (74XX139)
 - For $i=1$ to 7
 - If $!E="0"$ AND $i=s$ then $O_i=I$ else $O_i="Z"$
 - Nota: l'uscita di una tabella della verità può assumere vari stadi: 0, 1, terzo stato (z), indeterminato, non inizializzato,..) o essere una funzione degli ingressi



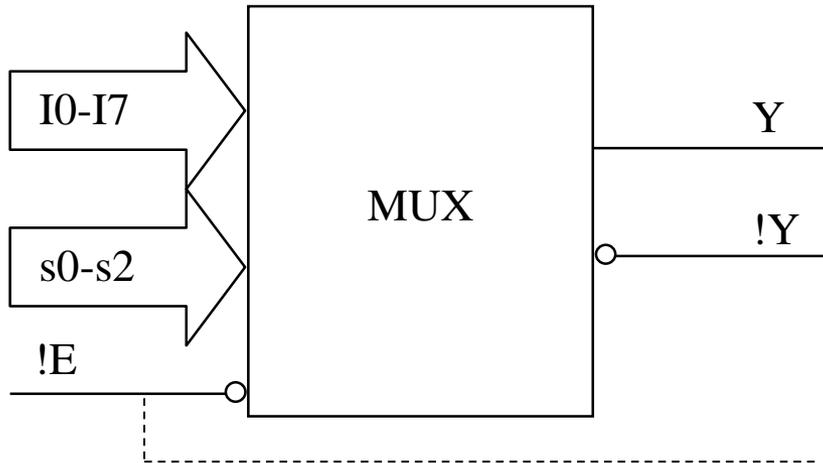
!E	s2	s1	s0	O0	O1	O2	O3	O4	O5	O6	O7
H	X	X	X	z	z	z	z	z	z	z	z
L	L	L	L	I	z	z	z	z	z	z	z
L	L	L	H	z	I	z	z	z	z	z	z
L	L	H	L	z	z	I	z	z	z	z	z
L	L	H	H	z	z	z	I	z	z	z	z
L	H	L	L	z	z	z	z	I	z	z	z
L	H	L	H	z	z	z	z	z	I	z	z
L	H	H	L	z	z	z	z	z	z	I	z
L	H	H	H	z	z	z	z	z	z	z	I

- Nota: se $In=1$ allora il Demux è un decoder da n a 2^n

Dispositivi combinatori

• Multiplexer

- Mediante le linee s_i si seleziona la linea I_j che, se $!E="0"$, va in Y
 - Se $!E="1"$ allora l'uscita può essere in 3-state o negata o identificata da $!y=y$
 - Utilizzati per l'utilizzo comune di risorse (ad esempio per codificare in seriale)
 - Si possono collegare ad albero Es. 74XX151
 - $Y = I_0 \& !s_0 \& !s_1 \& !s_2 + I_1 \& s_0 \& !s_1 \& !s_2 + I_2 \& !s_0 \& s_1 \& !s_2 + \dots$

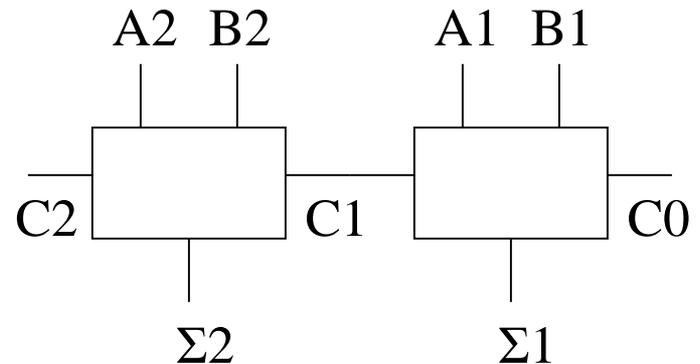
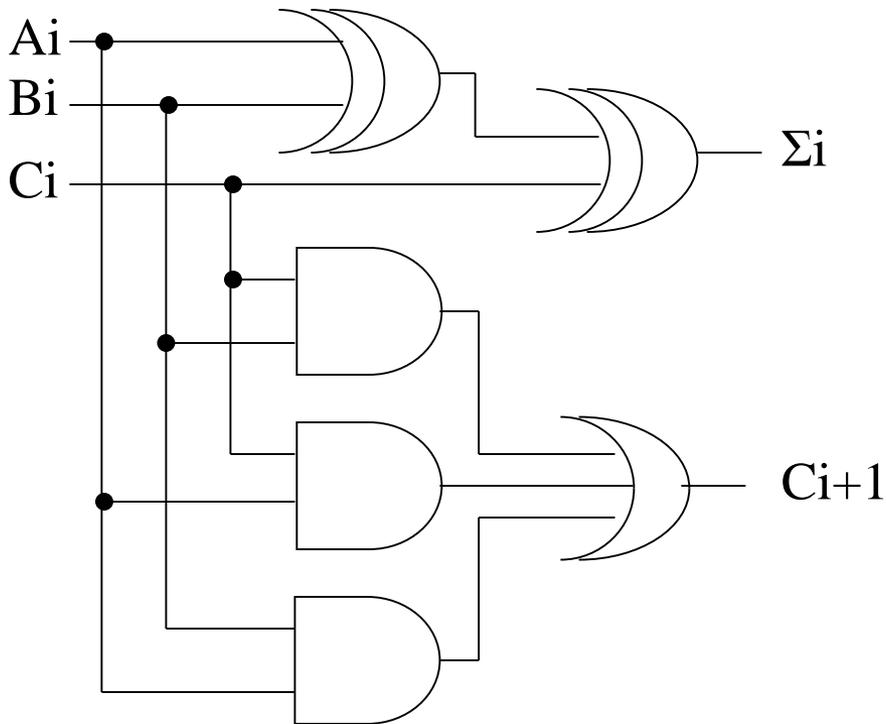


!E	s2	s1	s0	!y	y
H	X	X	X	H	L
L	L	L	L	!I0	I0
L	L	L	H	!I1	I1
L	L	H	L	!I2	I2
L	L	H	H	!I3	I3
L	H	L	L	!I4	I4
L	H	L	H	!I5	I5
L	H	H	L	!I6	I6
L	H	H	H	!I7	I7

- Nota: le uscite della tabella non sono codificate in termini di "0" e "1" ma di funzione che risulta in AND con il minterm indicato da quella riga. Es. $y = I_0 \& E \& !s_2 \& !s_1 \& !s_0 + I_1 \& E \& !s_2 \& s_1 \& s_0 + \dots + I_7 \& E \& s_2 \& s_1 \& s_0$

Dispositivi combinatori: sommatore

- **SOMMATORI** (es. 4-bit binary full-adder 74XX83 o 74XX283)
 - Tempo di propagazione max. da C_0 a $C_4 = 24\text{ns}$ (74LS83) Nota: $T_i + T_f + T_o$
 - Ingressi: A_{1-4}, B_{1-4}, C_0 Uscite: Σ_{1-4}, C_4
 - $\Sigma_i = A_i \cdot \text{XOR} \cdot B_i \cdot \text{XOR} \cdot C_{i-1}$ $C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$
 - E' possibile connettere in ripple-carry più moduli di questo tipo
 - Ritardo T_f sommatore a n bit = $2n \cdot T_o$ (T_o =ritardo T_f di 1 porta), a causa di C_i



Dispositivi combinatori: sommatori

• SOMMATORI CARRY LOOKAHEAD

– In figura c'è il classico schema in cascata, ma è possibile anticipare il calcolo del carry per un gruppo di bit (es. 3) diminuendo il ritardo complessivo

– $C_i = A_i B_i + C_{i-1} (A_i + B_i)$

Espressione più semplice

– $C_0 = A_0 B_0$

Ritardo $C_0 = T_0$ (sono pronti tutti i P_i e i G_i)

– $C_1 = G_0 + P_0 C_0$

Ritardo $C_1 = \text{Ritardo } C_0 + 2T_0$

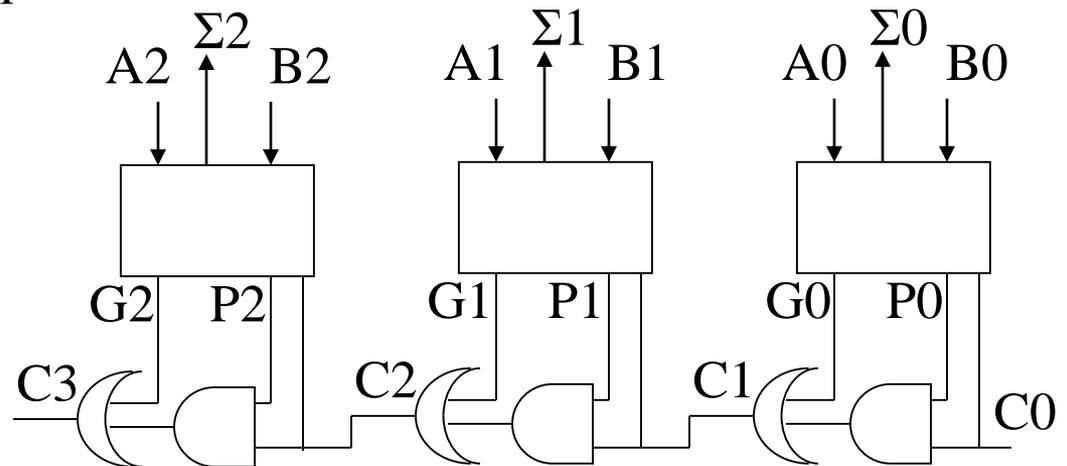
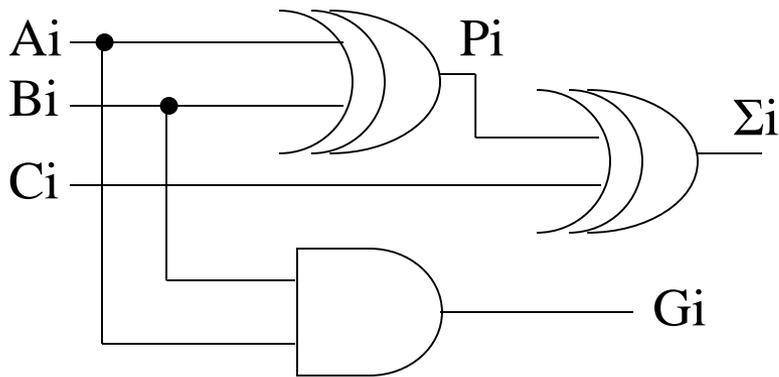
– $C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$

Ritardo $C_2 = \text{Ritardo } C_0 + 2T_0$

– $C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

Ritardo $C_3 = \text{Ritardo } C_2$

– Se uso porte con FAN-IN elevato, posso trattare gruppi di bit in SOP

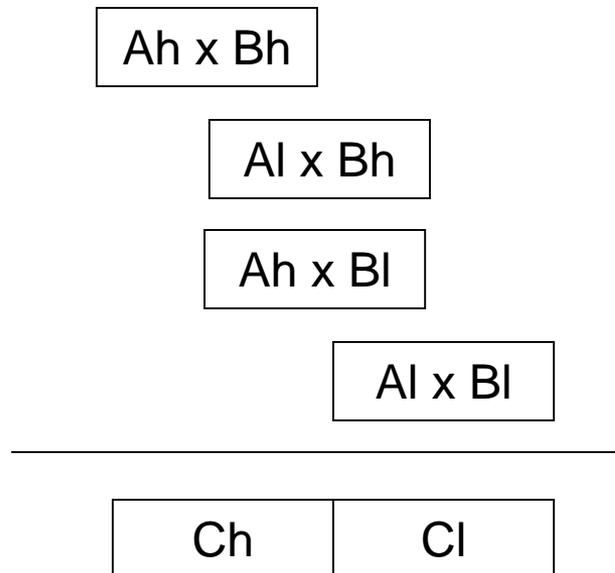


Dispositivi combinatori: sommatore

- **Dimensionamento forme AOI (SOP = Sum Of Product = circuiti con porte AND OR ed eventualmente negatori programmabili XOR)**
 - Realizzo la tabella della verità a n ingressi, 2^n righe (a meno che non riesca ad ottimizzare il numero di righe compattando righe per condizioni "indifferenti") e k uscite
 - Il FAN-IN richiesto alle porte AND è pari a n .
 - Il FAN-IN richiesto alle porte OR è pari alla metà del numero delle righe, ipotizzando di realizzare F se ci sono meno "1" che "0" e di realizzare $\neg F$ in caso contrario, quindi pari a 2^{n-1}
 - Es. sommatore a 4 bit: FAN-IN(AND) = 9, FAN-IN(OR) = 256!!!
 - Es. sommatore a 2 bit: FAN-IN(AND) = 5, FAN-IN(OR) = 16
- **Dimensionamento Look-up Table (memorie usate in lettura dove si suppone di collegare gli ingressi agli indirizzi e le uscite ai dati)**
 - La memoria deve disporre di n linee di indirizzi e quindi deve essere a 2^n celle
 - Le uscite determinano la capacità di ogni singola cella (es. 8 uscite -> memoria organizzata a byte)
 - Es. sommatore a 4 bit (9 ingressi: 4 bit per ciascun addendo più il CarryIN): memoria da 512 celle da 5 bit

Dispositivi combinatori: moltiplicatori

- **MOLTIPLICATORI** (es. 4*4 bit multiplier 74XX284)
 - Architettura sincrona o asincrona
 - Ingressi: A_{0-3} , B_{0-3} Uscite: C_{0-7} E' facile scrivere le espressioni di C_{0-7}
 - E' possibile connettere in sequenza con sommatore più moduli di questo tipo (realizzazione full-HW: 4 moltiplicatori 2x2bit, 1 sommatore a 4 bit ($R = A1Bh + AhB1$) con shifter, 1 sommatore a 8 bit ($R + AhBh$ "concatenato" $A1B1$))
 - I 74XX284 sono pochissimo utilizzati (meglio realizzazione mediante FPGA)



A1	A0	B1	B0	C3	C2	C1	C0
0	0	X	X	0	0	0	0
X	X	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Dispositivi combinatori: comparatori

- **COMPARATORI (es. 8-bit-comparator 74XX85)**

- Tempo di propagazione massimo = 45ns
- Connessione in cascata per effettuare comparazioni a byte (guardo i 4 bit meno significativi solo se i 4 più significativi sono uguali)
- Nelle configurazioni ambigue di $I_{A>B}$ $I_{A<B}$ e $I_{A=B}$ (informazione che viene dallo stadio meno significativo), si considera rotto o assente lo stadio precedente
- $A_i > B_i$ equivale a $A_i \& !B_i$, $A_i < B_i$ equivale a $!A_i \& B_i$, $A_i = B_i$ equivale a $A_i \cdot \text{XOR} \cdot B_i$

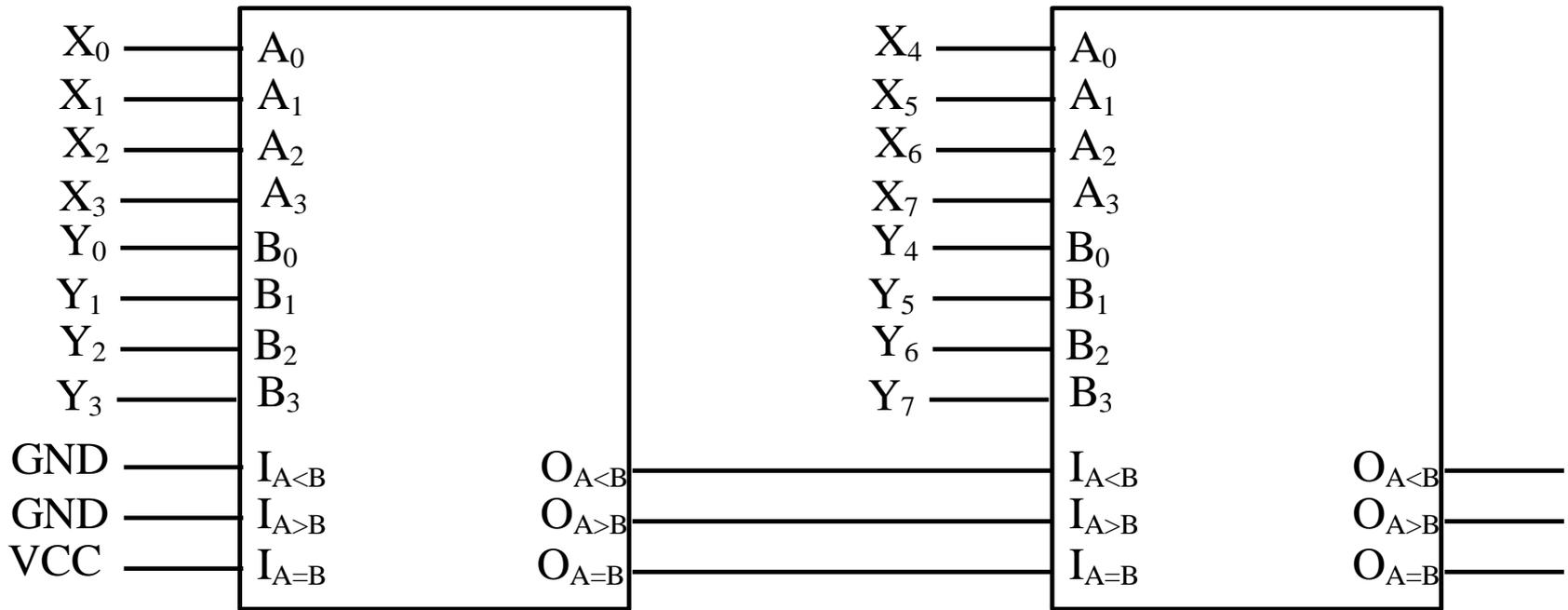
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$I_{A=B}$	$I_{A>B}$	$I_{A<B}$	$O_{A=B}$	$O_{A>B}$	$O_{A<B}$
$A_3 > B_3$	X	X	X	X	X	X	L	H	L
$A_3 < B_3$	X	X	X	X	X	X	L	L	H
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	H	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	X	X	H	L	L

Dispositivi combinatori

- Comparatori

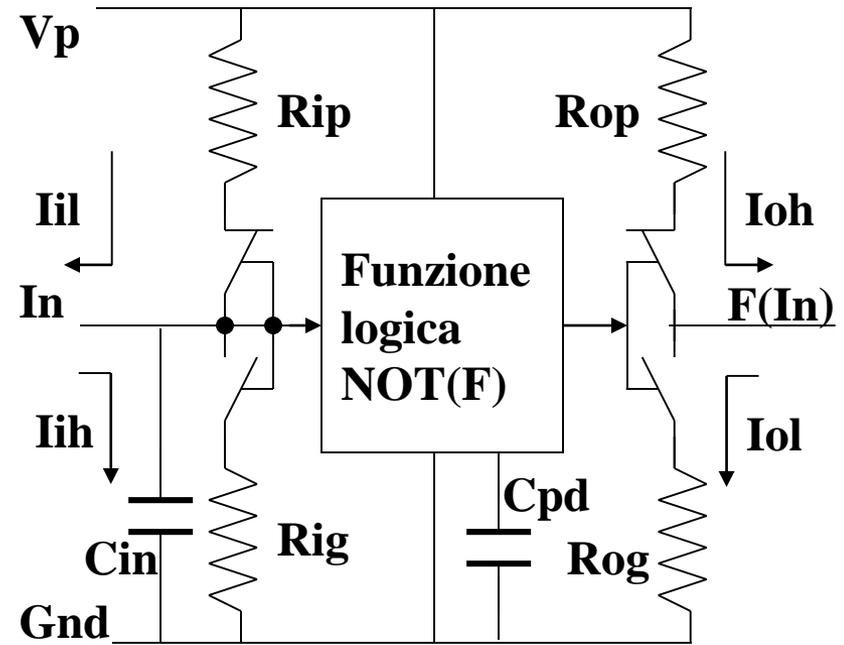
- Connessione in cascata per effettuare comparazioni a byte

- Gli ingressi $I_{A>B}$, $I_{A<B}$ e $I_{A=B}$ agiscono se $A_i=B_i$ per ogni valore di i

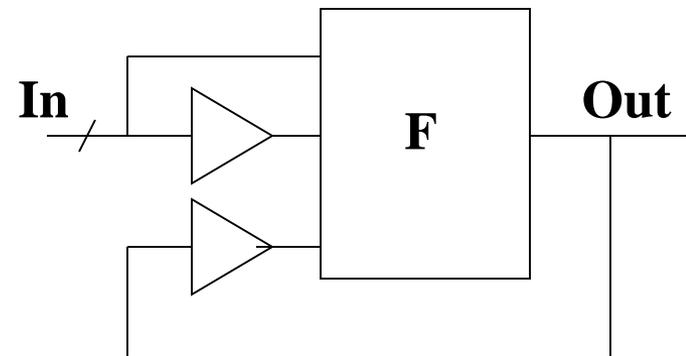


Modello, dispositivi combinatori e sequenziali

- **Funzione logica**
- **Funzione logica F combinatoria:**
 - Può essere descritta come tabella dove
$$\text{Out}(t) = F(\text{In}(t))$$
 - Ogni ingresso ha uguale interfaccia elettrica (tranne gli ingressi Schmidt trigger, dove $\text{In}(t) = S(\text{In}(t), \text{In}(t-\tau))$)
 - Ogni uscita ha uguale interfaccia elettrica
 - Si definiscono tempi di propagazione da ogni ingresso a ogni uscita



- **Funzione logica F sequenziale**
 - Può essere descritta come tabella dove
$$\text{Out}(t) = F(\text{In}(t), \text{In}(t-\tau), \text{Out}(t-\tau))$$
 - o più propriamente
$$\text{Out}(t+\tau) = F(\text{In}(t), \text{In}(t-\tau), \text{Out}(t))$$



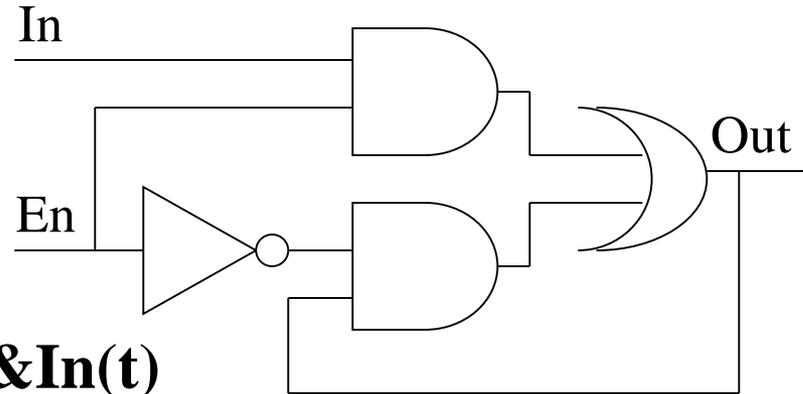
Dispositivi sequenziali: latch

- Latch di tipo “D”

$$\text{Out}(t) = F(\text{In}(t), \text{Out}(t-\tau))$$

$$\text{Out}(t) = \text{!En}(t) \& \text{Out}(t-\tau) + \text{En}(t) \& \text{In}(t)$$

$$\text{Out} = \text{!En} \& \text{Out} + \text{En} \& \text{In}$$



- Esistono anche altri latch

- RS
- JK
- T

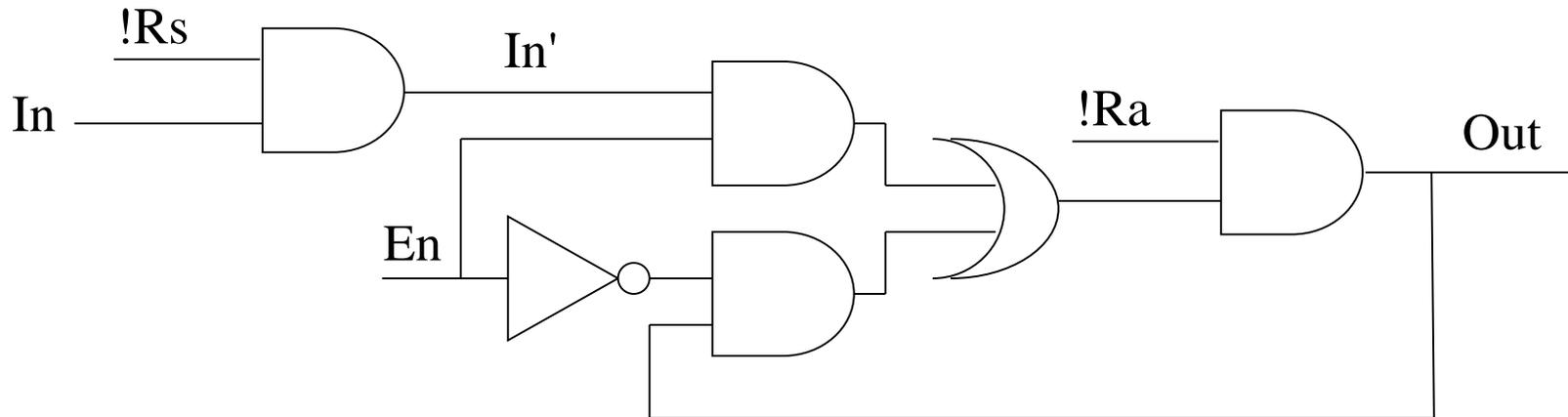
- **NOTA:**

– gli ingressi alle porte AND hanno tutti uno stadio con negatore programmabile in modo da avere ritardi di propagazione uniformi evitando la formazione di stati "spuri" di breve durata

En (t)	In (t)	Out(t-τ)	Out(t)
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

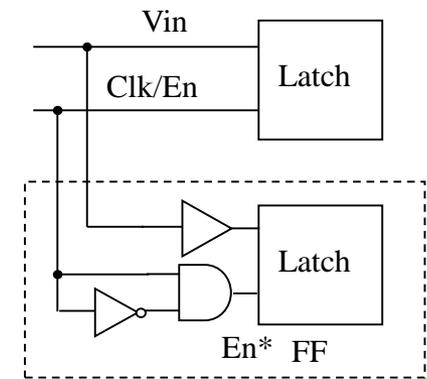
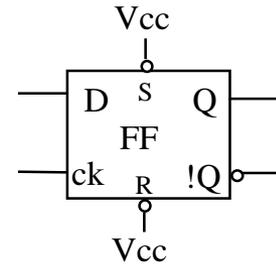
Dispositivi sequenziali

- **Reset sincrono e asincrono**
- **Reset sincrono R_s**
 - Agisce (attivo basso) sull'uscita quando permesso da En
- **Reset asincrono R_a**
 - Agisce (attivo basso) sull'uscita indipendentemente da En



Dispositivi sequenziali

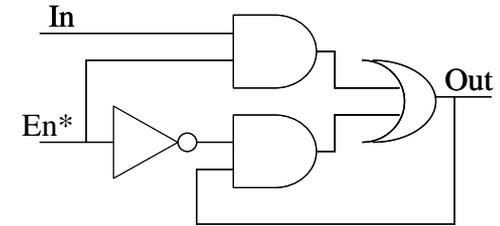
- Flip-flop (FF)
- Flip-flop di tipo “D”



$$\text{Out}(t) = F(\text{In}(t), \text{In}(t-\tau), \text{Out}(t-\tau))$$

$$\text{Out}(t) = (!\text{En}(t) + \text{En}(t) \& \text{En}(t-\tau)) \& \text{Out}(t-\tau) + \text{En}(t) \& !\text{En}(t-\tau) \& \text{In}(t)$$

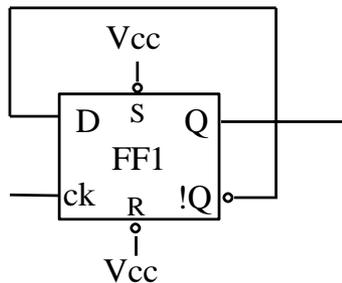
$$\text{Out}(t) = !\text{En}^*(t) \& \text{Out}(t-\tau) + \text{En}^*(t) \& \text{In}(t)$$



NOTA: Attenzione agli impulsi generati da disallineamento temporale tra i percorsi (es. En^*)

- Esistono anche altri FF

- RS
- JK
- T

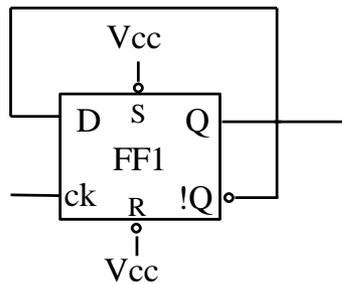
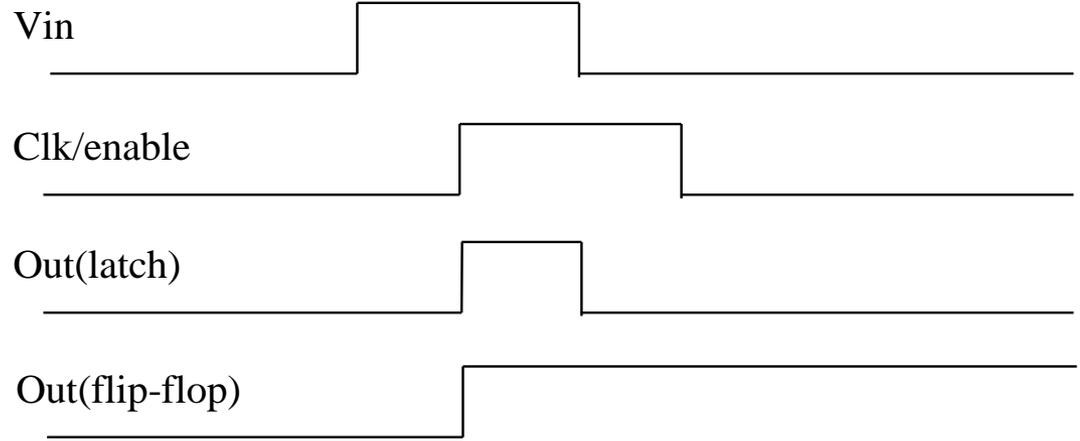
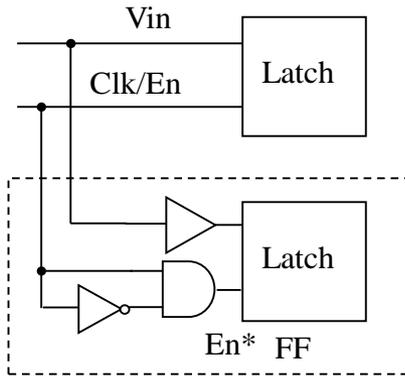


En (t)	En (t-τ)	In (t)	Out(t-τ)	Out(t)
0	X	X	0	0
0	X	X	1	1
1	1	X	0	0
1	1	X	1	1
1	0	0	X	0
1	0	1	X	1

Dispositivi sequenziali

- Latch e Flip-Flop

- Differenze



!S	!R	ck	J	!K	Q	!Q
L	H	X	X	X	H	L
X	L	X	X	X	L	H
H	H	stable	X	X	Q	!Q
H	H	edge+	L	L	L	H
H	H	edge+	L	H	Q	!Q
H	H	edge+	H	L	!Q	Q
H	H	edge+	H	H	H	L

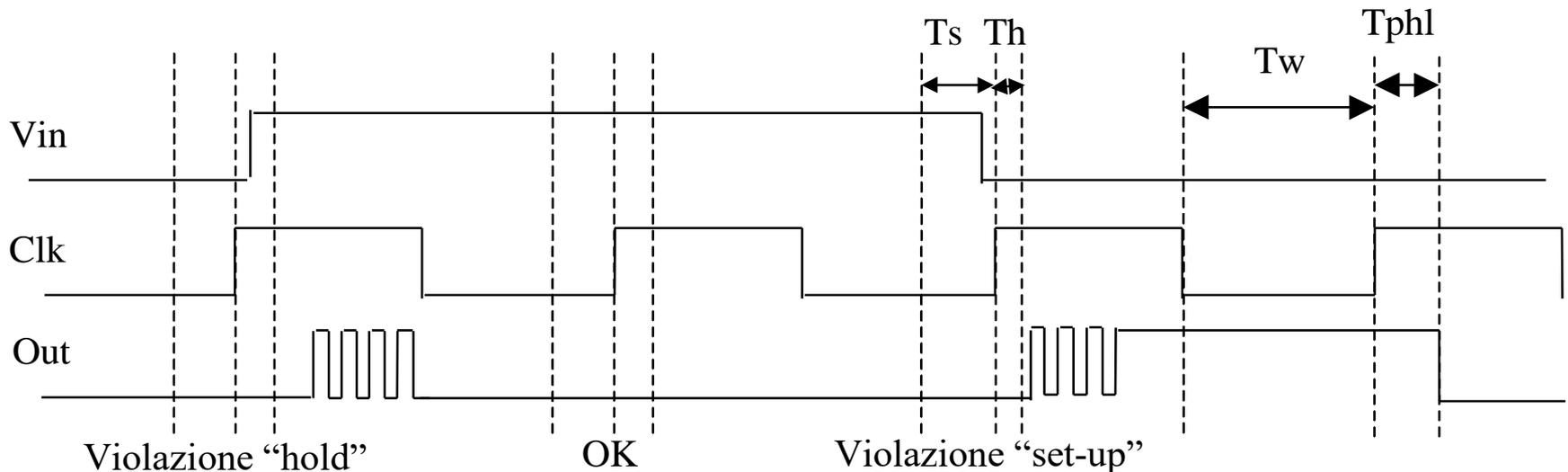
Dispositivi sequenziali

- **Caratteristiche dinamiche**

- **I dati (o le linee sincrone di set/reset) devono essere stabili un certo tempo prima del fronte attivo del clock (Tempo di set-up = T_s)**

- **I dati (o le linee sincrone di set/reset) devono essere stabili un certo tempo dopo il fronte attivo del clock (Tempo di hold = T_h)**

- **Si deve rispettare la massima frequenza F_{max} e la minima larghezza di impulso di clock e set/reset asincroni)**



Dispositivi sequenziali

- Caratteristiche dinamiche
- Flip-flop D 74LS74 (Motorola)

(Esempio)

	min	typ	max
T_{phl} (from ck to Q, !Q)		12ns	40ns
T_{plh} (from ck to Q, !Q)		10ns	25ns
T_{phl} (from S,R to Q, !Q)		12ns	40ns
T_{plh} (from S,R to Q, !Q)		11ns	25ns
T_w (ck, S, R)	25ns	12ns	
T_{su} (setup time)	20ns	18ns	
T_h (hold time)	5ns	-1ns	
F_{max}		25MHz	40MHz

Test Conditions (T_a = 25°C, V_{cc} = 5V, C_l = 15pF)

- **T_{set-up} + T_{hold} ≈ tempo di reazione**
 - La realizzazione di molte macchine sincrone richiede **T_{hold} < T_{p_{ck,Q}}**

Dispositivi sequenziali

- **Metastabilità**
- **La metastabilità è uno stato di funzionamento dei circuiti sequenziali nel quale l'uscita assume un valore imprevedibile (“0”, “1”, oscillante)**
- **La metastabilità evolve verso uno stato stabile**
 - lo stato stabile è equiprobabile ($P(\text{“0”}) \approx P(\text{“1”}) \approx 50\%$)
 - tempo in cui si raggiunge uno stato stabile al 99.9999% = “resolving time”
 - La probabilità P di sopravvivenza dello stato metastabile al tempo t segue una legge esponenziale $P(t) = e^{-kt}$

Es. 74HC:

$P(10\text{ns}) \approx 10\%$

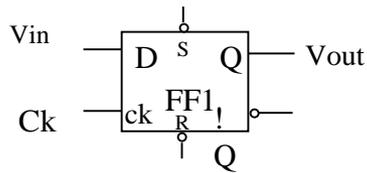
$P(20\text{ns}) \approx 1\%$

$P(100\text{ns}) \approx 10^{-6}$

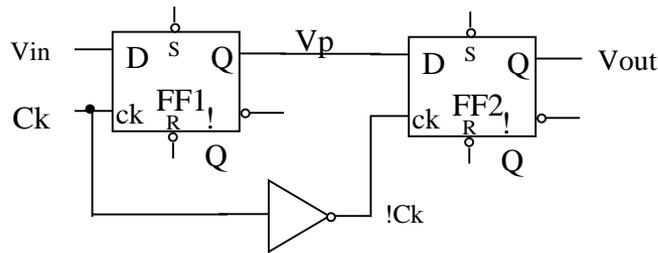
- **Si può avere metastabilità se:**
 - non si rispettano le condizioni di set-up
 - non si rispettano le condizioni di hold
 - non si rispettano le condizioni di minima larghezza d'impulso

Dispositivi sequenziali

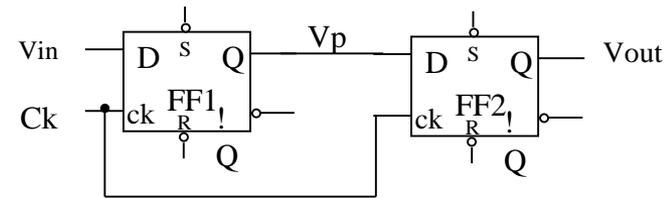
- Circuiti “robusti” alla metastabilità
- Il problema della sincronizzazione in ingresso di segnali asincroni



A



B

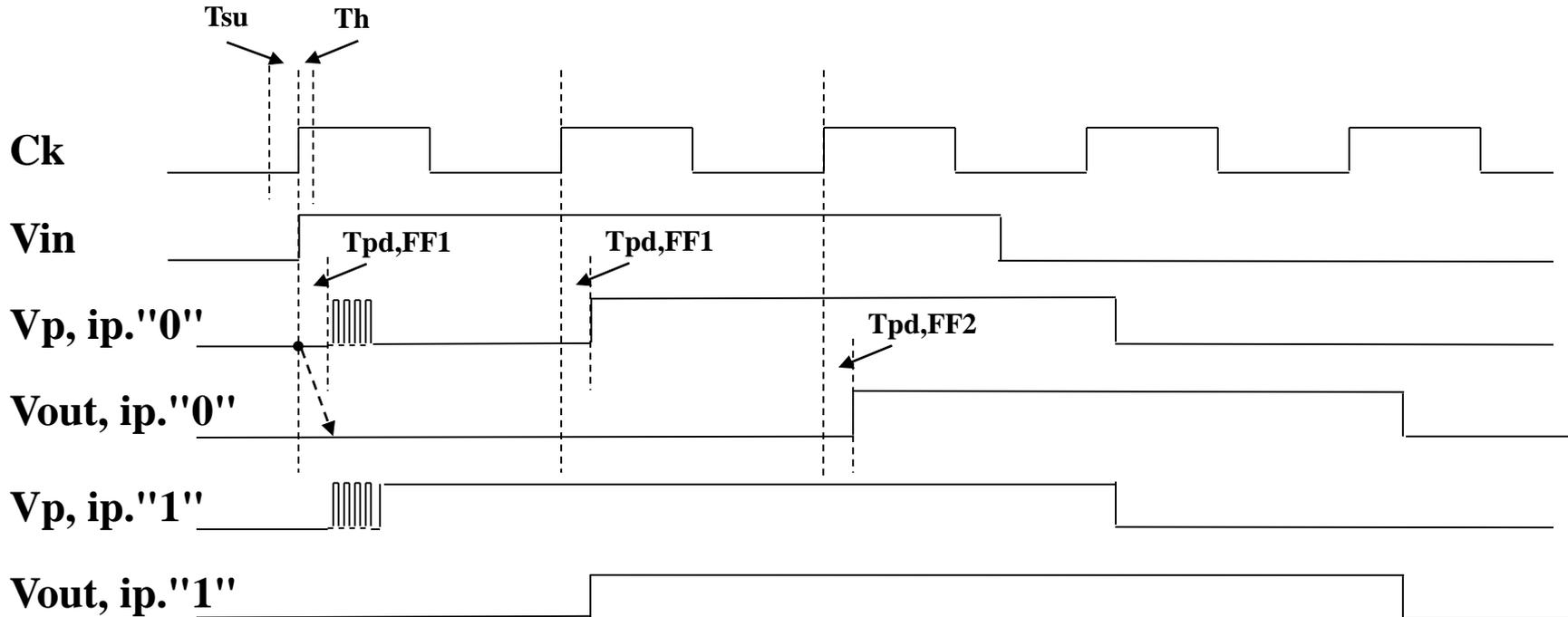
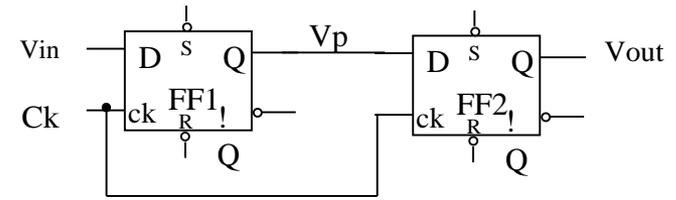


C

- **A:** V_{out} potrebbe essere metastabile per violazione di “set-up” o di “hold”. Il ritardo da V_{in} a V_{out} è variabile tra T_p e $T_p + T_{ck}$
- **B:** V_p potrebbe essere metastabile per violazione di “set-up” o di “hold”. Se $T_{ck}/2$ è superiore al resolving time allora V_{out} si considera esente da metastabilità. Il ritardo tra V_{in} e V_{out} è variabile tra $T_p + 0.5T_{ck}$ e $T_p + 1.5T_{ck}$
- **C:** V_p potrebbe essere metastabile per violazione di “set-up” o di “hold”. Se T_{ck} è superiore al resolving time allora V_{out} si considera esente da metastabilità. Il ritardo tra V_{in} e V_{out} è variabile tra $T_p + T_{ck}$ e $T_p + 2T_{ck}$

Dispositivi sequenziali

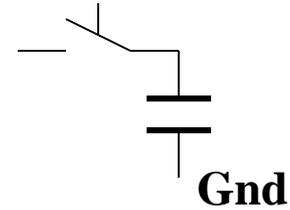
- Circuiti “robusti” alla metastabilità
- Analisi dei segnali



- Il tempo di propagazione di FF1 $T_{pd,FF1}$ fa sì che FF2 campioni un segnale stabile, purchè $T_{pd,FF1}$ sia maggiore del tempo di hold T_h di FF2
- Il ritardo di propagazione complessivo $Ck \rightarrow Vout$ varia tra $T_{ck} + T_{pd,FF2}$ (situazione che si avrebbe se V_{in} commutasse un attimo prima del clock rispettando il tempo di set-up T_{su}) e $2T_{ck} + T_{pd,FF2}$ (situazione che si avrebbe se V_{in} commutasse poco dopo del clock rispettando T_h)

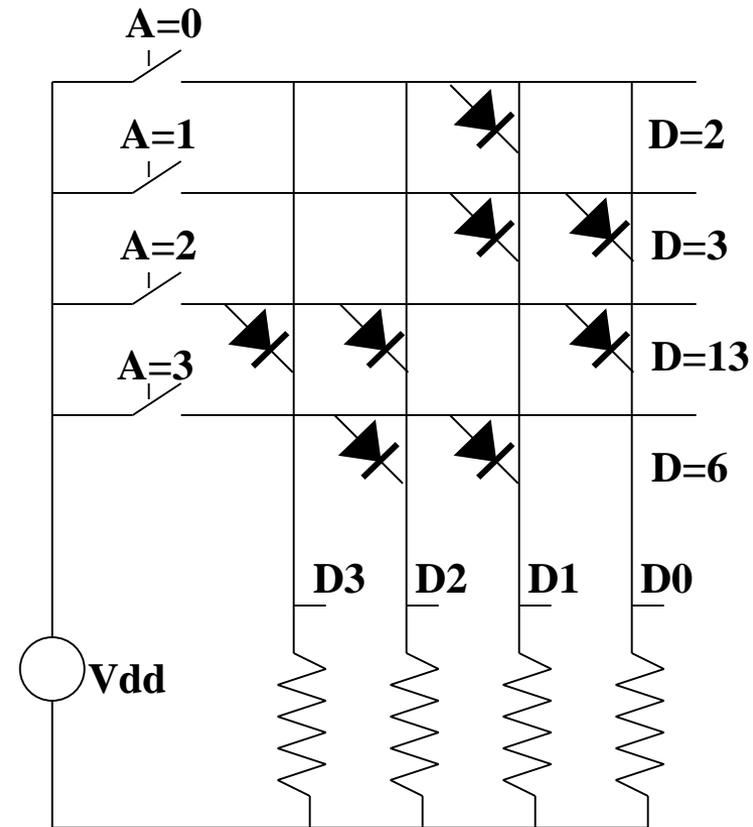
Memorie

- **Memorie RAM dinamiche (DRAM)**
- **Un condensatore intrappola la carica**
 - Necessità di rinfresco
 - Usate in schede video (veloci, elevata integrazione)
- **Synchronous DRAM (SDRAM)**
 - Le operazioni (scrittura e/o lettura) sono vincolate ad un clock
- **Double Data Rate DRAM (DDR SDRAM)**
 - Utilizza entrambi i fronti del segnale di sincronismo
- **DDR**
 - 2.5V, frequenze di clock variabili da 200MHz a 400MHz (DDR 400 o PC3200)
 - Tagli da 128, 256, 512 e 1024 MB. Banda passante pari a 3,2 GB/s
- **DDR2**
 - 1,8V, Frequenze di clock da 400MHz a 1GHz
- **DDR3**
 - 1,5V, Frequenze di clock da 800MHz a 2GHz



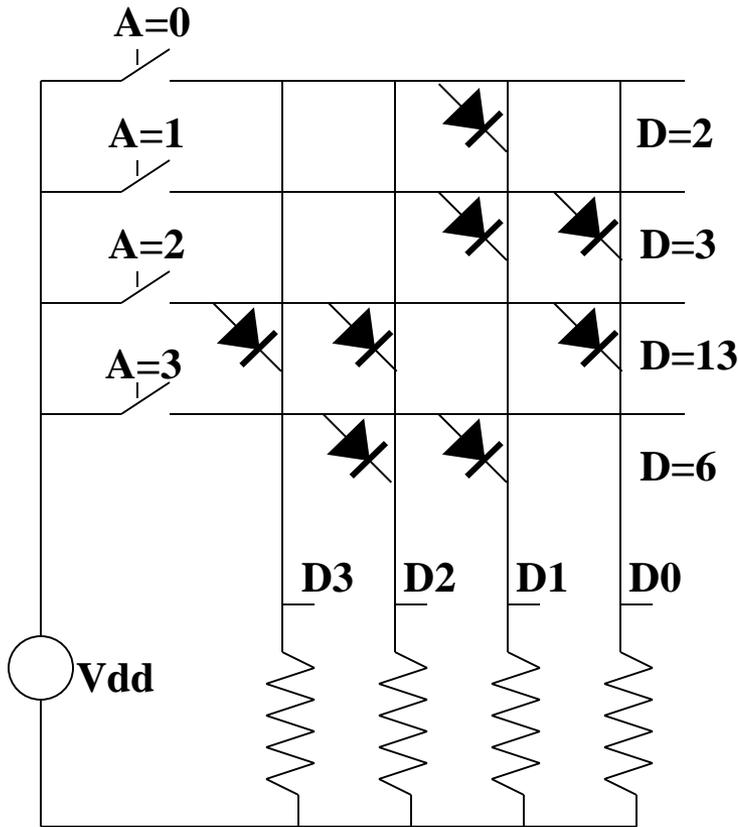
Memorie

- Memorie ROM
- Read Only Memory
- E' scritta dal costruttore
- Non può essere cancellata (memoria non volatile)
- Si può avere un percorso conduttivo (1) che si interrompe programmando (->0)
 - programmazione mediante corrente
 - logica “fuse”
- Si può avere un percorso isolante (0) che si interrompe programmando (->1)
 - programmazione mediante tensione
 - logica “antifuse”
 - (Es. condensatore al quale foro il dielettrico)

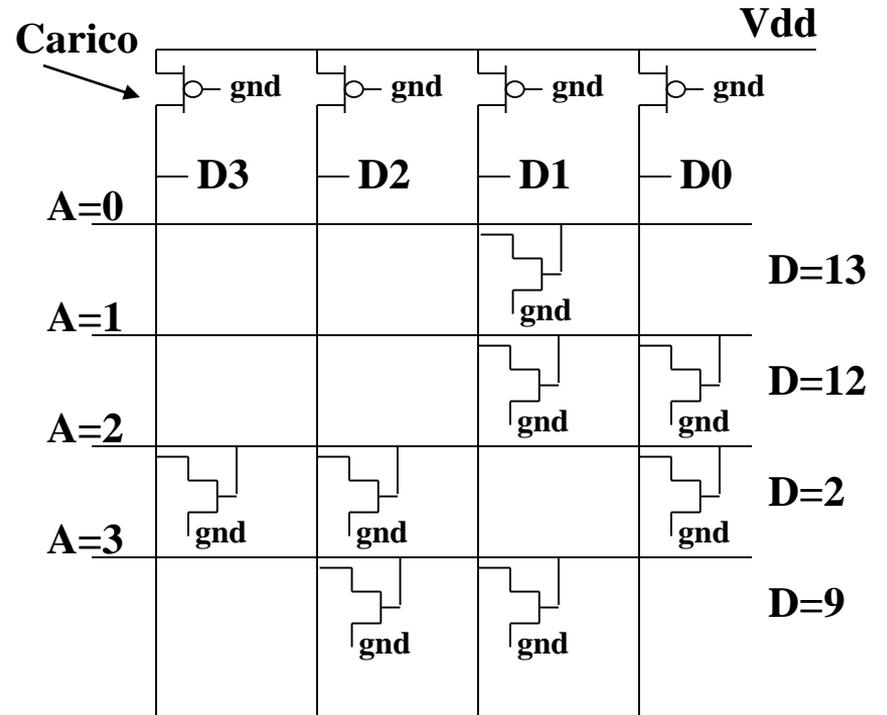


Memorie

- Memorie ROM
- Mask-programmed ROM

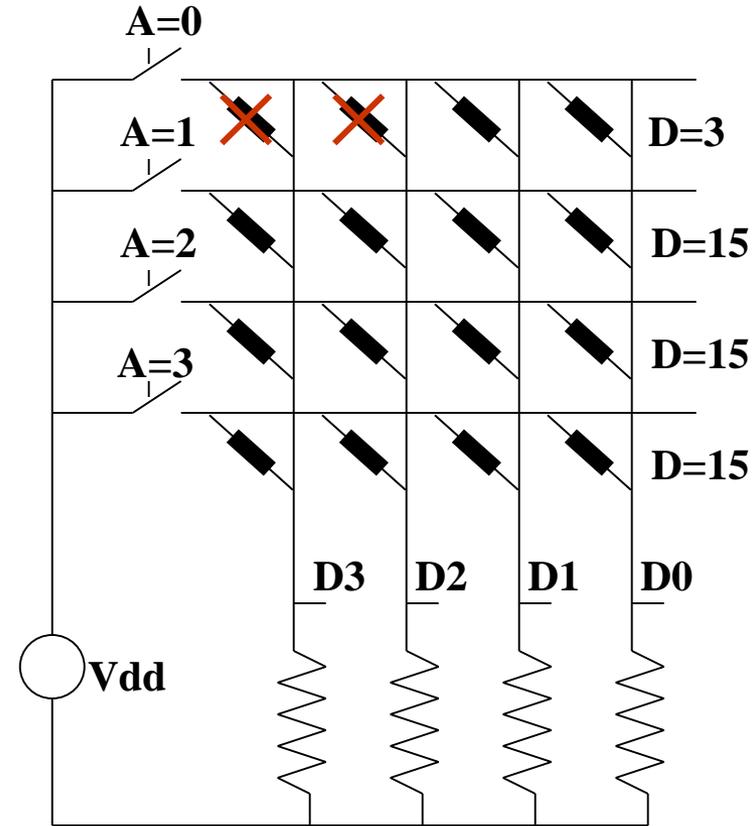


NMOS Mask-programmed ROM



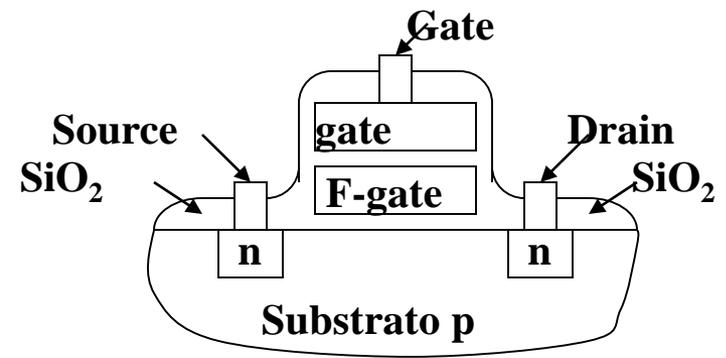
Memorie

- Memorie PROM
- Programmable Read Only Memory
- E' scritta dall'utente
- Non può essere cancellata (memoria non volatile)
- Si può avere un percorso conduttivo (1) che si interrompe programmando (->0)
 - programmazione mediante corrente
 - logica “fuse”
- Si può avere un percorso isolante (0) che si interrompe programmando (->1)
 - programmazione mediante tensione
 - logica “antifuse”
 - (Es. condensatore al quale foro il dielettrico)



Memorie

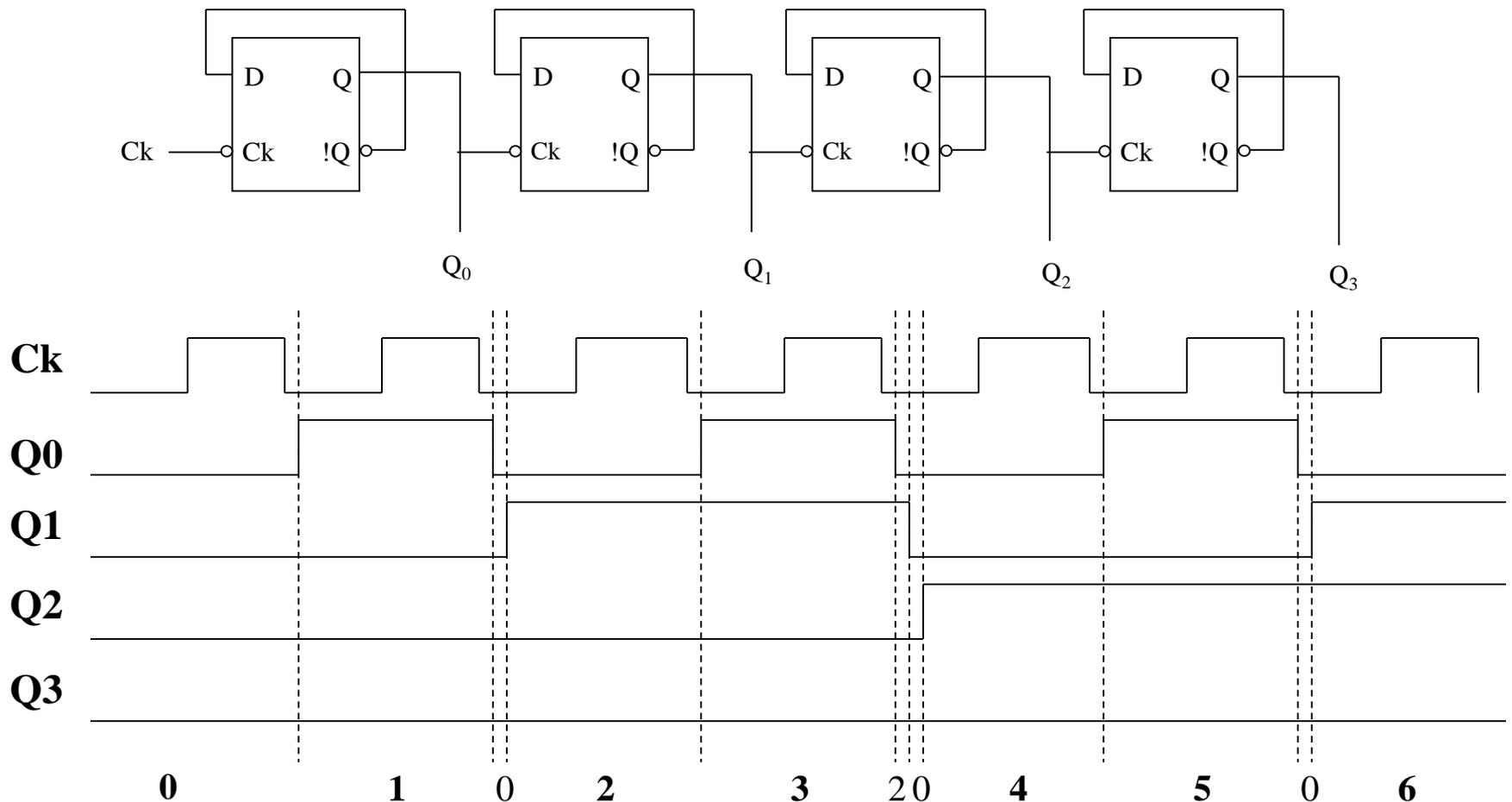
- Memorie EPROM
- Erasable PROM



- **Si utilizza un NMOS speciale con gate aggiuntivo flottante**
 - Applicando tensioni gate-source molto elevate si produce un breakdown a valanga (elettroni ad alta energia penetrano attraverso il SiO₂ e rimangono intrappolati nel gate flottante in polisilicio)
 - L'accumulazione della carica nel gate flottante impedisce la formazione del canale (l'NMOS è come se non ci fosse)
 - La luce UV dà agli elettroni intrappolati l'energia sufficiente per liberarsi
- **Electrically Erasable PROM (E²PROM)**
 - La carica viene accumulata o rimossa dal gate flottante per effetto tunnel grazie ad una riduzione dello strato di ossido tra gate flottante e drain
 - Controllo cella a cella, ma minore densità perché per evitare interferenza tra le celle, le celle sono più complesse (due o più transistor)
- **Flash Electrically Erasable PROM (Flash E²PROM)**
 - La carica viene accumulata per effetto valanga e rimossa per effetto tunnel
 - Cella a singolo transistor (migliore integrazione), ma cancellazione globale

Dispositivi sequenziali

- **Contatori Asincroni (poco usati –Es. 74LS93, $F_{max}=30\text{MHz}$, $T_{pd}=70\text{ns}$)**
 - **Necessitano di uno stadio di sincronizzazione a valle (4 Flip-flop con il clock in comune e i segnali Q_i in ingresso) per eliminare lo skew e gli stati spuri**
 - **Nel caso di clock attivo sul fronte di discesa si ha un contatore down**

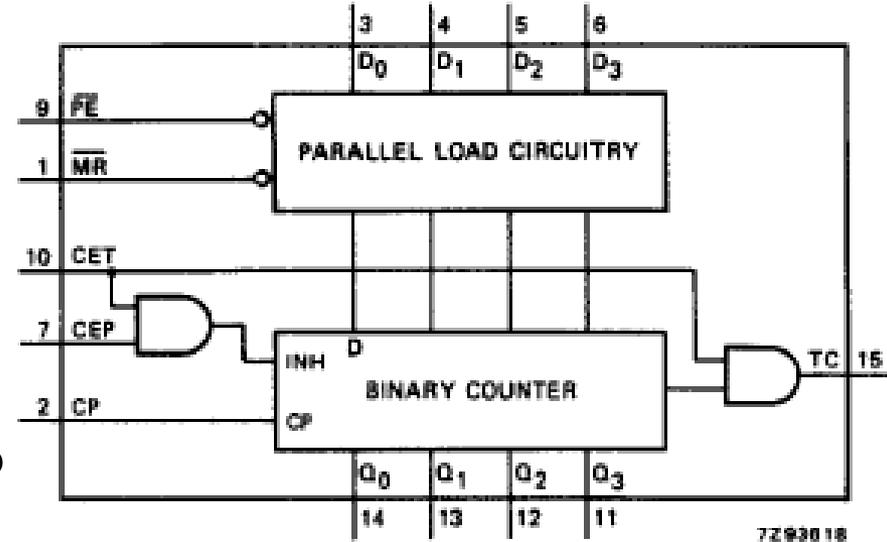


Dispositivi sequenziali

• Contatori

• Sincroni 4 bit (fronte di salita)

- Organizzazione (binaria, BCD);
- Capacità (numero di bit)
- Con reset sincrono/asincrono;
- Con preset (LOAD) sincrono/asincrono
- UP/DOWN (Carry/Borrow);
- Es. 74LS163, $F_{max}=25\text{MHz}$, $T_{pd}=28\text{ns}$



74HC_HCT163 from Philips Semiconductor

OPERATING MODE	INPUTS						OUTPUTS	
	$\overline{\text{MR}}$	CP	CEP	CET	$\overline{\text{PE}}$	D_n	Q_n	TC
reset (clear)	l	\uparrow	X	X	X	X	L	L
parallel load	h	\uparrow	X	X	l	l	L	L
	h	\uparrow	X	X	l	h	H	(1)
count	h	\uparrow	h	h	h	X	count	(1)
hold (do nothing)	h	X	l	X	h	X	q_n	(1)
	h	X	X	l	h	X	q_n	L

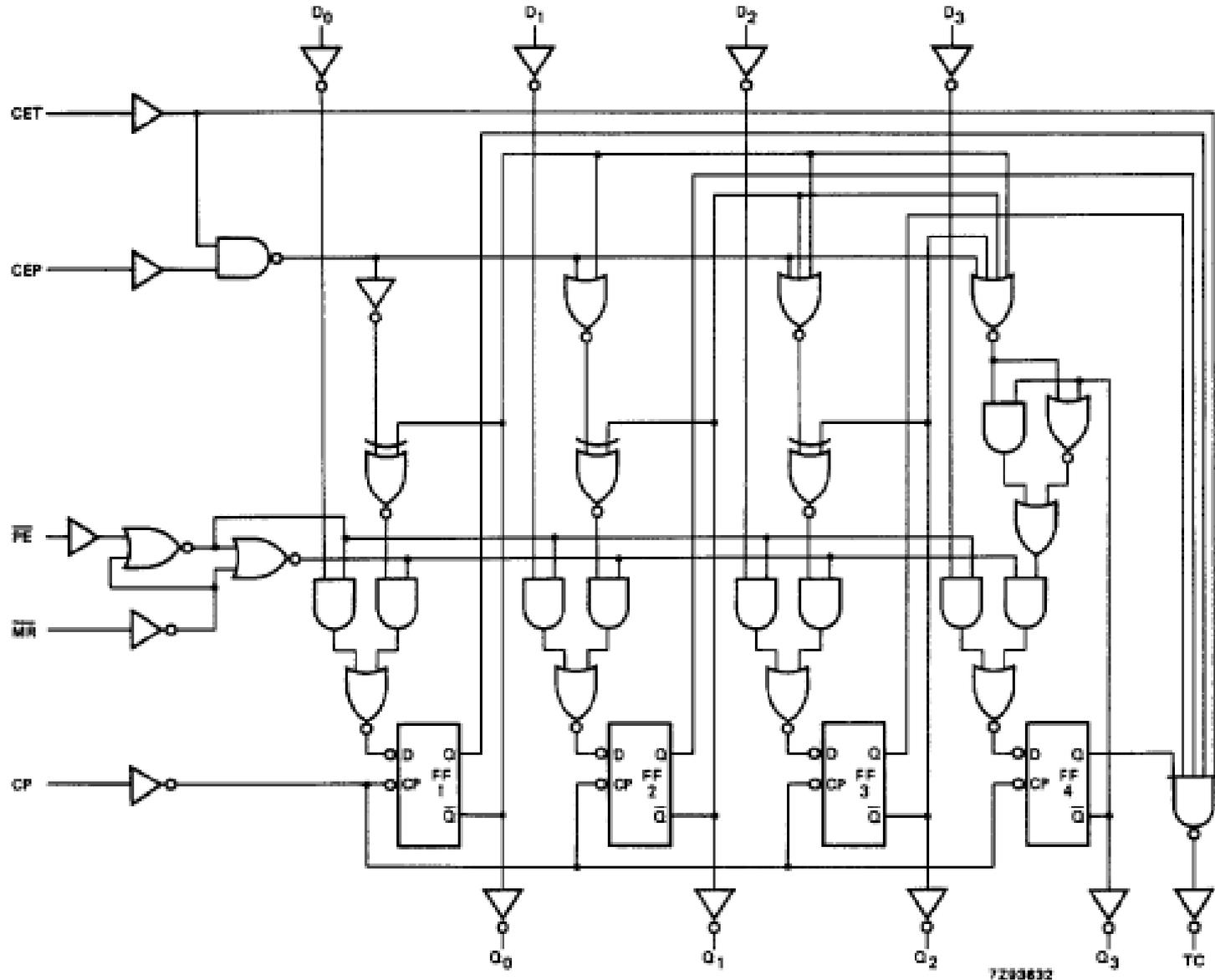
• Segnali

- $\overline{\text{MR}}$ Master Reset sincrono
- CEP Count Enable Input
- CET Count Enable Carry Input
- CP Clock (fronte di salita)
- $\overline{\text{PE}}$ Parallel Enable Input
- TC Terminal Count Output (1111)

Dispositivi sequenziali

74HC_HCT163 from Philips Semiconductor

- Contatore sincrono



Dispositivi sequenziali

74HC_HCT163 from Philips Semiconductor

- Contatore sincrono

Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

- Equazioni

$$Q0 = !Q0$$

$$Q1 = Q0 \& !Q1 + Q1 \& !Q0 = Q1.XOR.Q0$$

$$Q2 = !Q2 \& Q1 \& Q0 + Q2 \& !Q1 \& !Q0$$

$$+ Q2 \& !Q1 \& Q0 + Q2 \& Q1 \& !Q0 =$$

$$= Q2 \& (!Q1 + !Q0) + !Q2 \& Q1 \& Q0 =$$

$$= Q2 \& !(Q1 \& Q0) + !Q2 \& Q1 \& Q0 =$$

$$= Q2.XOR.Q1 \& Q0 =$$

$$= Q2.XOR.!(Q1 \& !Q0)$$

$$Q3 =$$

- “pseudolinguaggio”

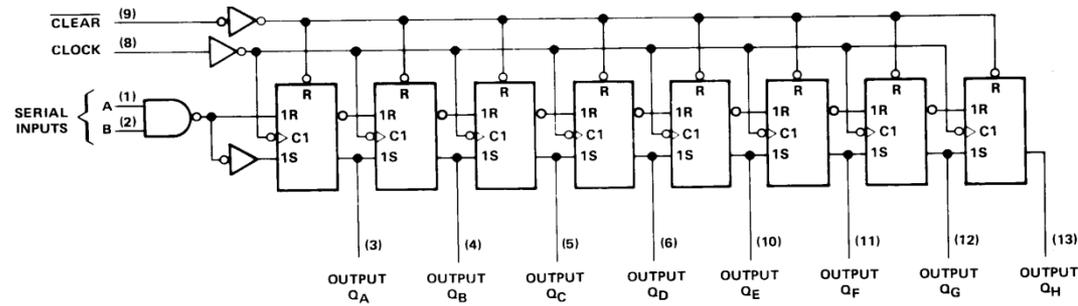
If Q0 = ‘1111’ then Q0 <= ‘0000’

else Q0 <= Q0+1;

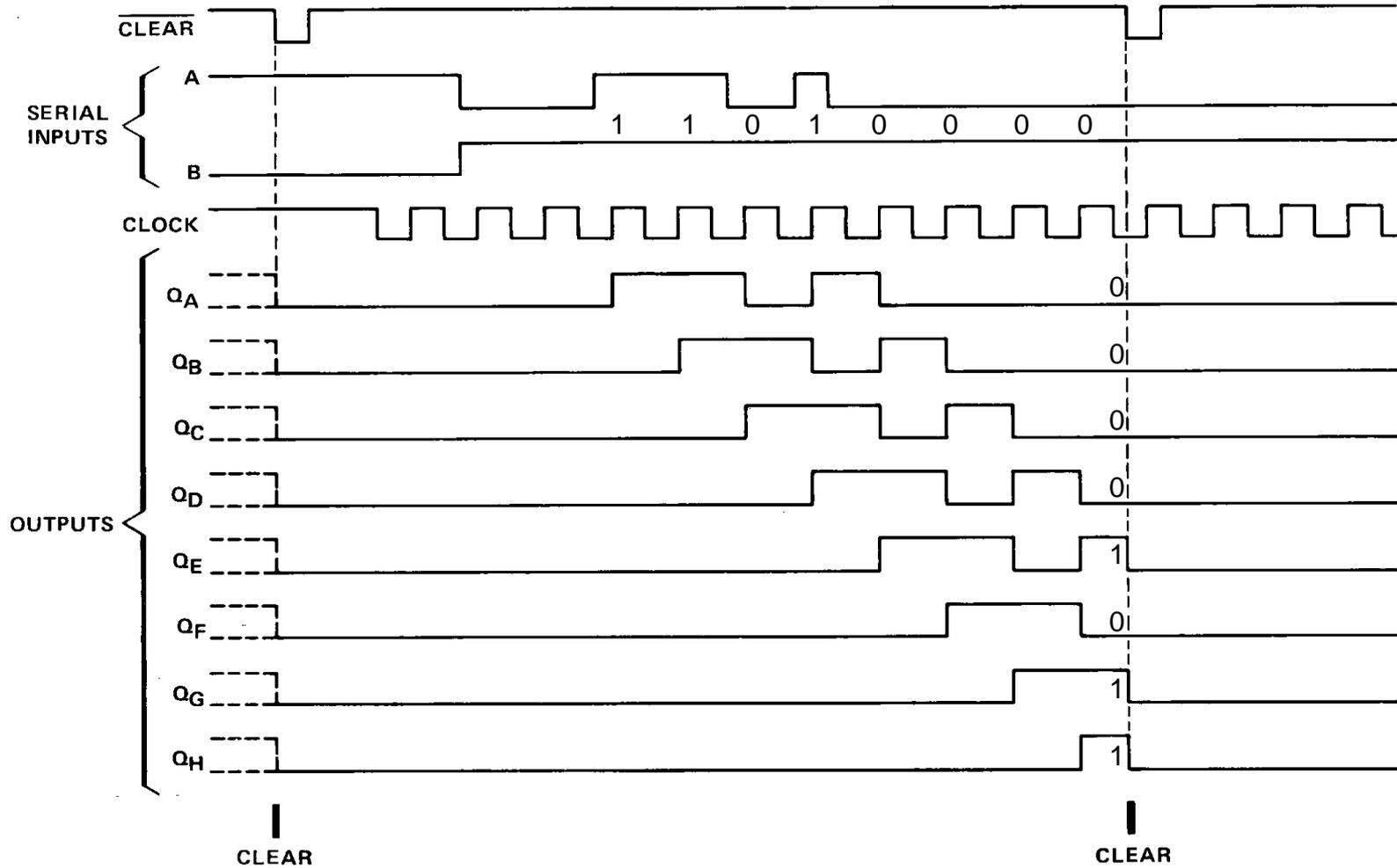
- Come capisco se è una funzione combinatoria o sequenziale?

Dispositivi sequenziali

• Shift Register –Es. 74XX164



Pin numbers shown are for D, J, N, and W packages.

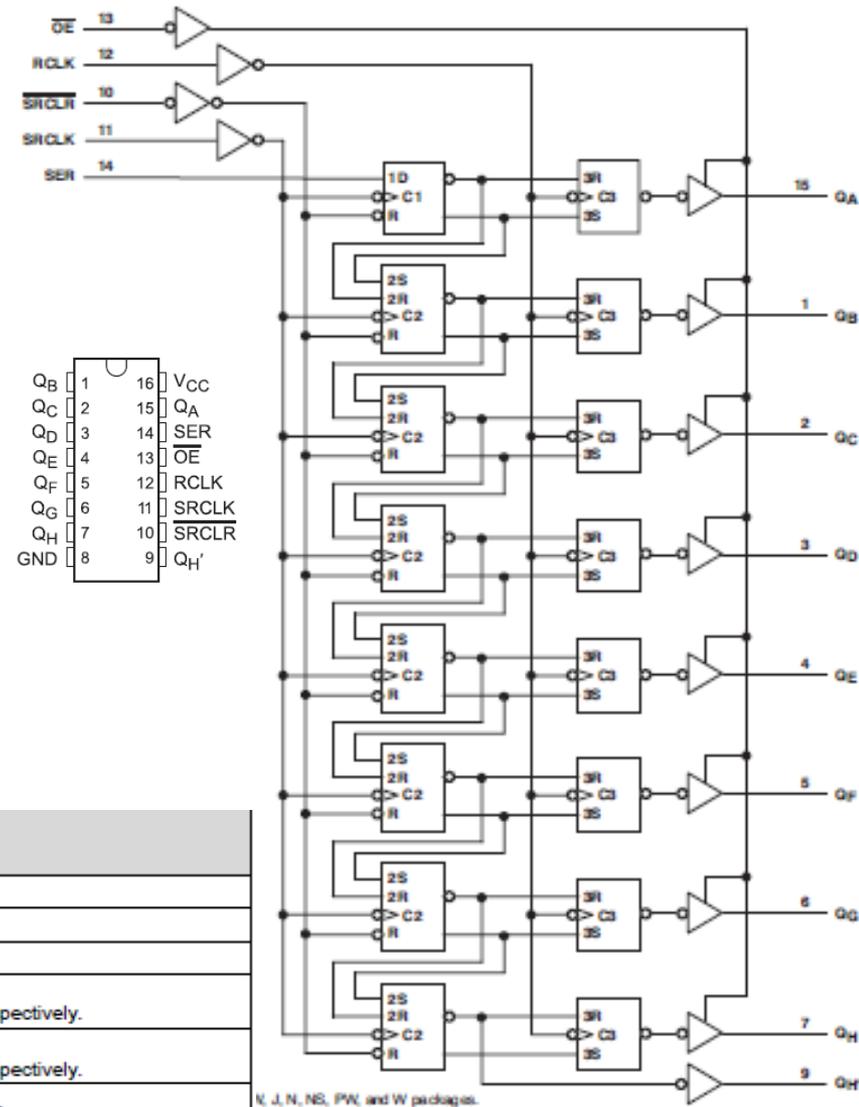
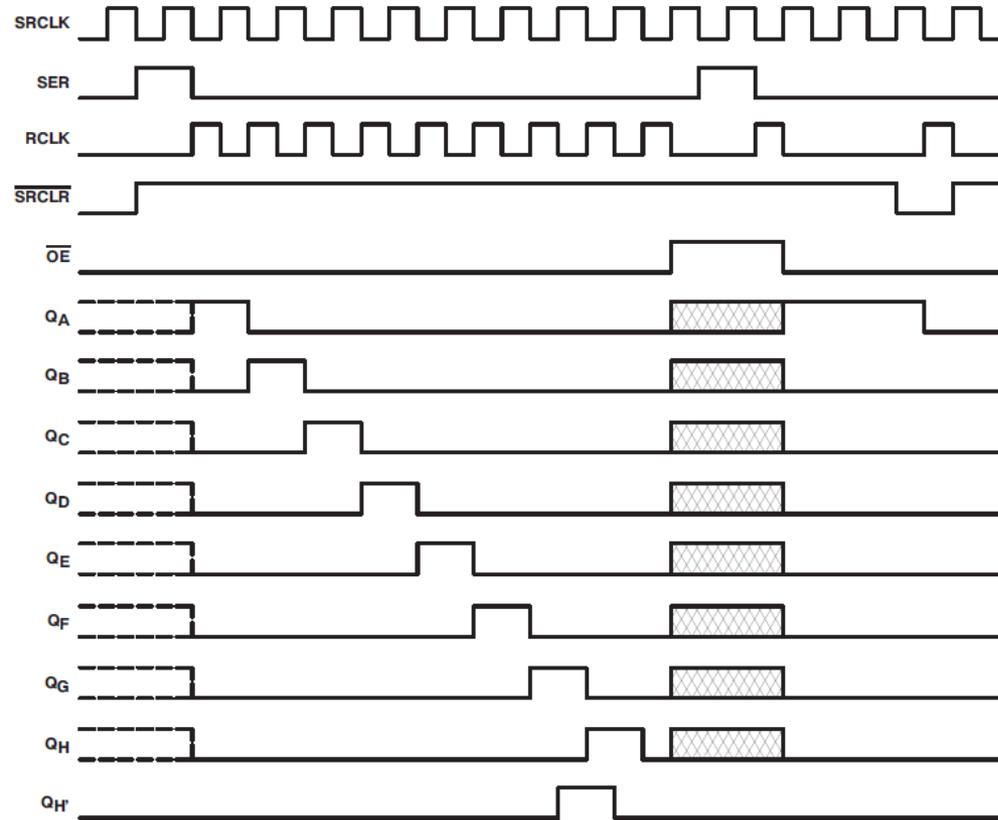


Dispositivi sequenziali

Shift Register – 74XX595, kit

• Utilizzato come "Port expander"

- SRCLK fa avanzare il dato SER nello shift register
- RCLK, dopo 8 SRCLK, memorizza il dato parallelo
- !OE abilita l'uscita del dato parallelo



QB	1	16	VCC
QC	2	15	QA
QD	3	14	SER
QE	4	13	OE
QF	5	12	RCLK
QG	6	11	SRCLK
QH	7	10	SRCLR
GND	8	9	QH'

INPUTS					FUNCTION
SER	SRCLK	SRCLR	RCLK	OE	
X	X	X	X	H	Outputs QA – QH are disabled.
X	X	X	X	L	Outputs QA – QH are enabled.
X	X	L	X	X	Shift register is cleared.
L	↑	H	X	X	First stage of the shift register goes low. Other stages store the data of previous stage, respectively.
H	↑	H	X	X	First stage of the shift register goes high. Other stages store the data of previous stage, respectively.
X	X	X	↑	X	Shift-register data is stored in the storage register.

V, J, N, NS, PW, and W packages.