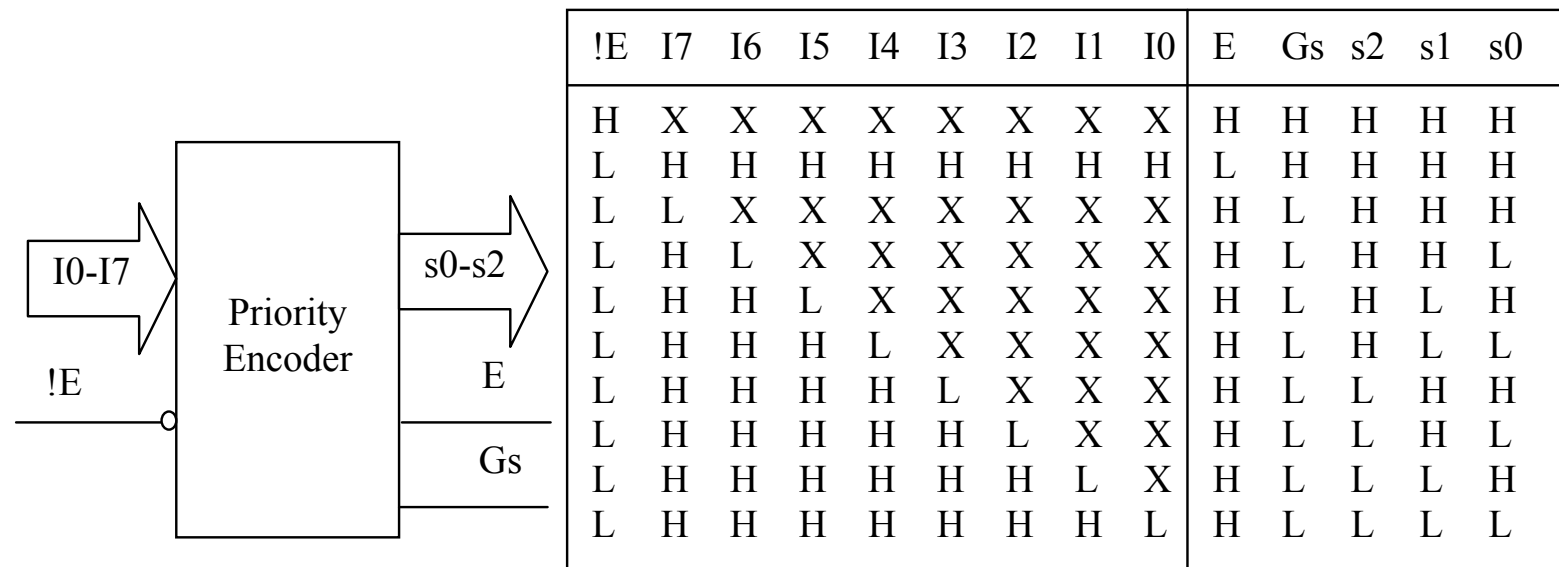




Dispositivi combinatori e sequenziali

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, vedi figura)

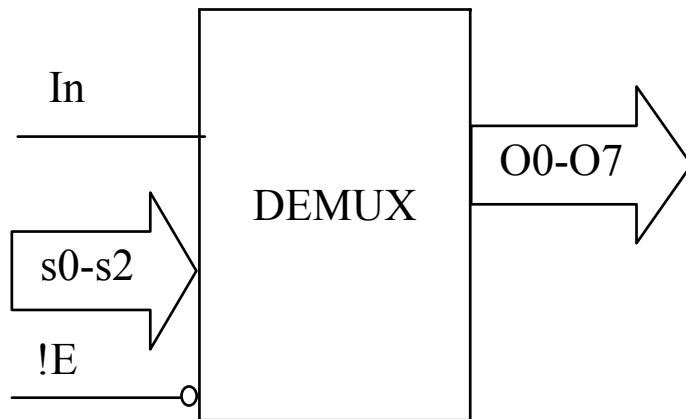


Dispositivi combinatori

- 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 $O_i="Z"$



!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

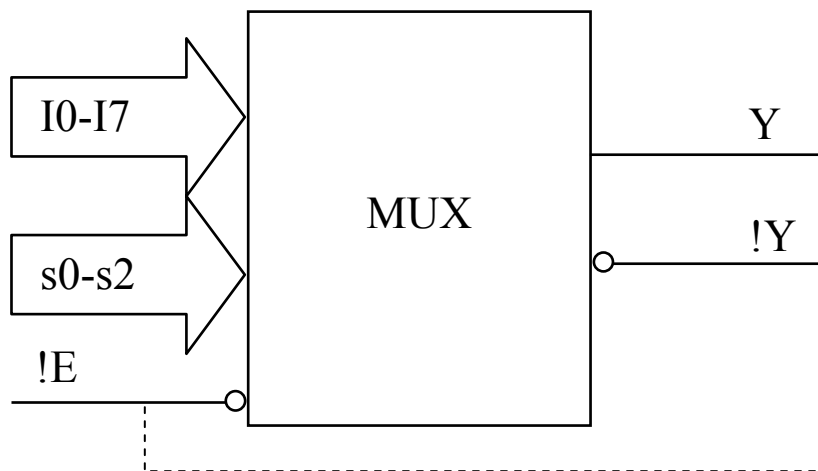
- **Nota:** se $In=1$ allora il Demux è un decoder da n a 2^n

Dispositivi combinatori

- **Multiplexer**

- **Mediante le linee s_i si seleziona la linea I_j che, se $!E="0"$, viene portata all'uscita 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

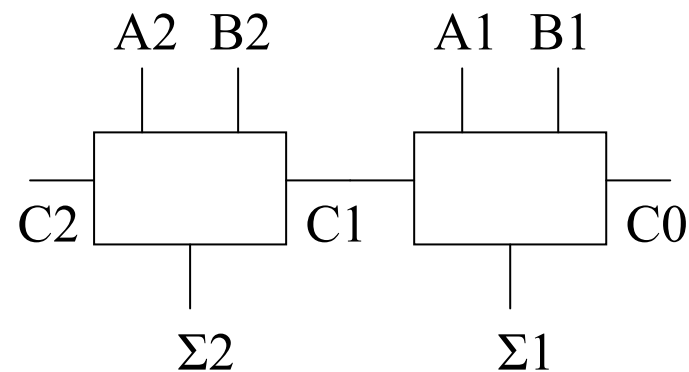
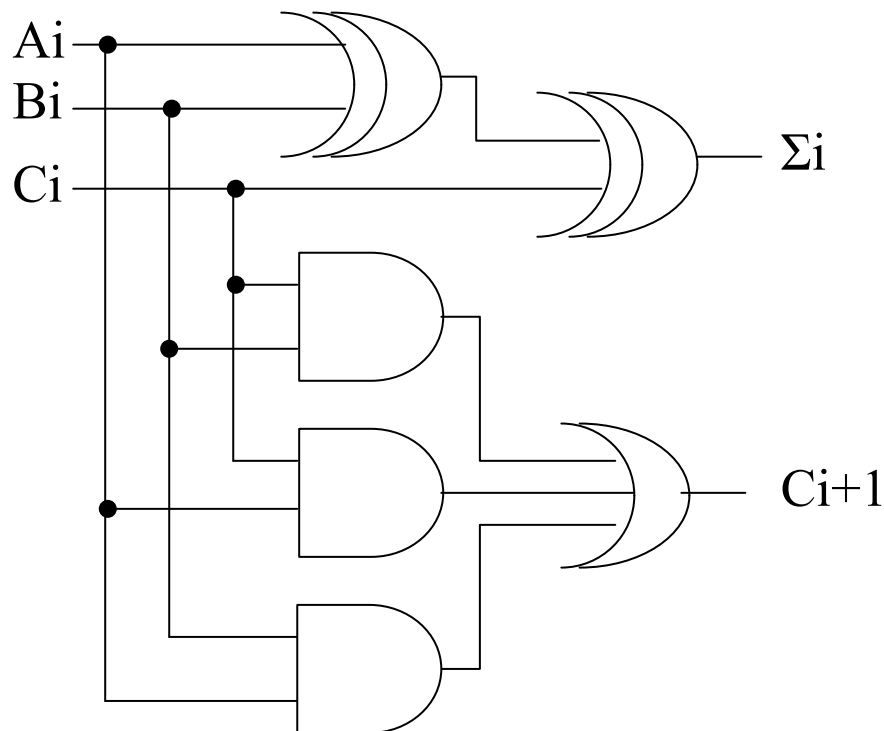
- **Nota:** l'indicazione $E=H$ poteva essere codificata in $!y=L$ $y=L$ oppure mandare le uscite in alta impedenza: : If $!E="0"$ then $Y=I(S)$ else $Y="Z"$

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)
- Ingressi: A_{1-4}, B_{1-4}, C_0 Uscite: Σ_{1-4}, C_4
- $\Sigma_i = A_i \cdot \text{XOR} \cdot B_i \cdot \text{XOR} \cdot C_{i-1}$ $C_i = A_i \cdot B_i + A_i \cdot C_{i-1} + B_i \cdot C_{i-1}$
- E' possibile connettere in ripple-carry più moduli di questo tipo
- Ritardo di un sommatore a n bit = $2n$



Dispositivi combinatori

- Sommatori

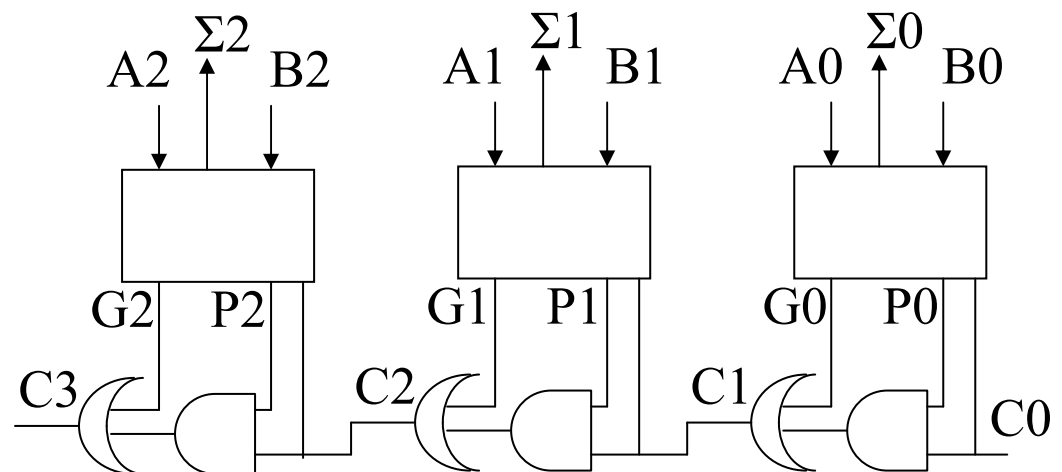
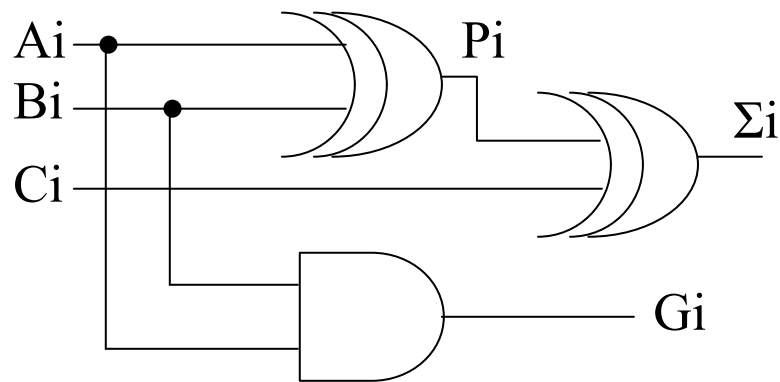
- **SOMMATORI CARRY LOOKAHEAD**

- In figura c'è il classico schema in cascata, ma possibile anticipare il calcolo del carry

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

- $C_0 = A_0 B_0$

- $C_1 = A_1 B_1 + A_0 B_0(A_1 + B_1)$



Dispositivi combinatori

- **Moltiplicatori**

- **MOLTIPLICATORI (es. 4*4 bit multiplier 74XX284)**

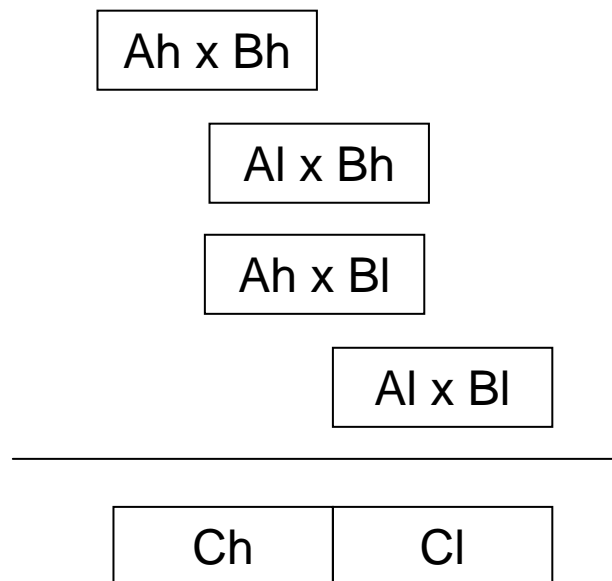
- Architettura sincrona o asincrona

- Ingressi: A_{0-3} , B_{0-3}

Uscite: C_{0-15}

- E' possibile connettere in sequenza con sommatori più moduli di questo tipo

- Praticamente 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

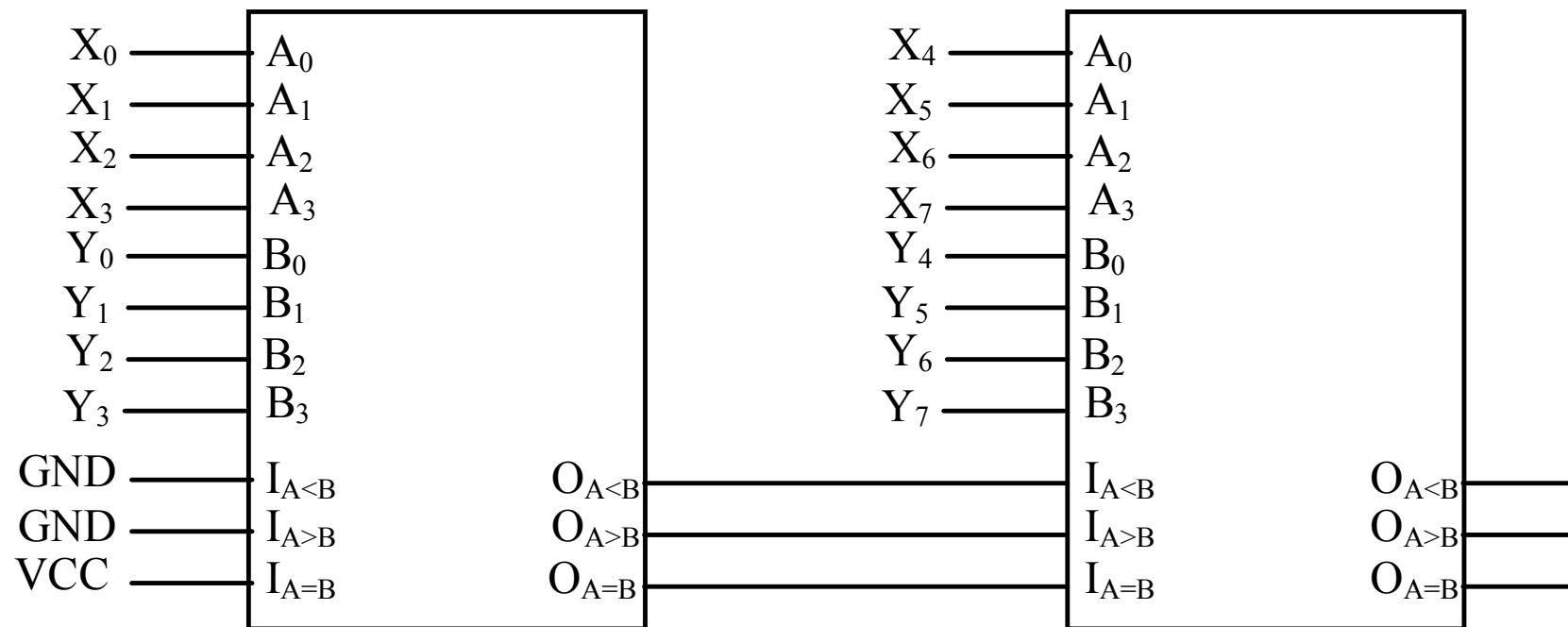
- **COMPARATORI (es. 8-bit-comparator 74XX85)**

- Tempo di propagazione massimo = 45ns
- Estensione a 5 bit connettendo LSB(A) a $I_{A>B}$ e LSB(B) a $I_{A<B}$ e $I_{A=B}$ a massa
- Connessione in cascata per effettuare comparazioni a byte

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	L	H	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	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

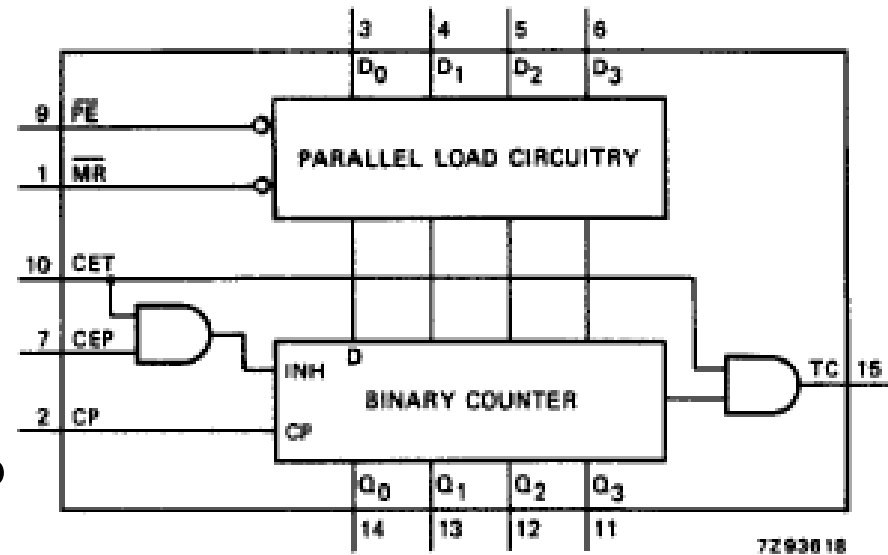
Dispositivi combinatori

- 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



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, $F_{max}=25MHz$, $T_{pd}=28ns$



74HC_HCT163 from Philips Semiconductor

OPERATING MODE	INPUTS						OUTPUTS	
	\overline{MR}	CP	CEP	CET	\overline{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

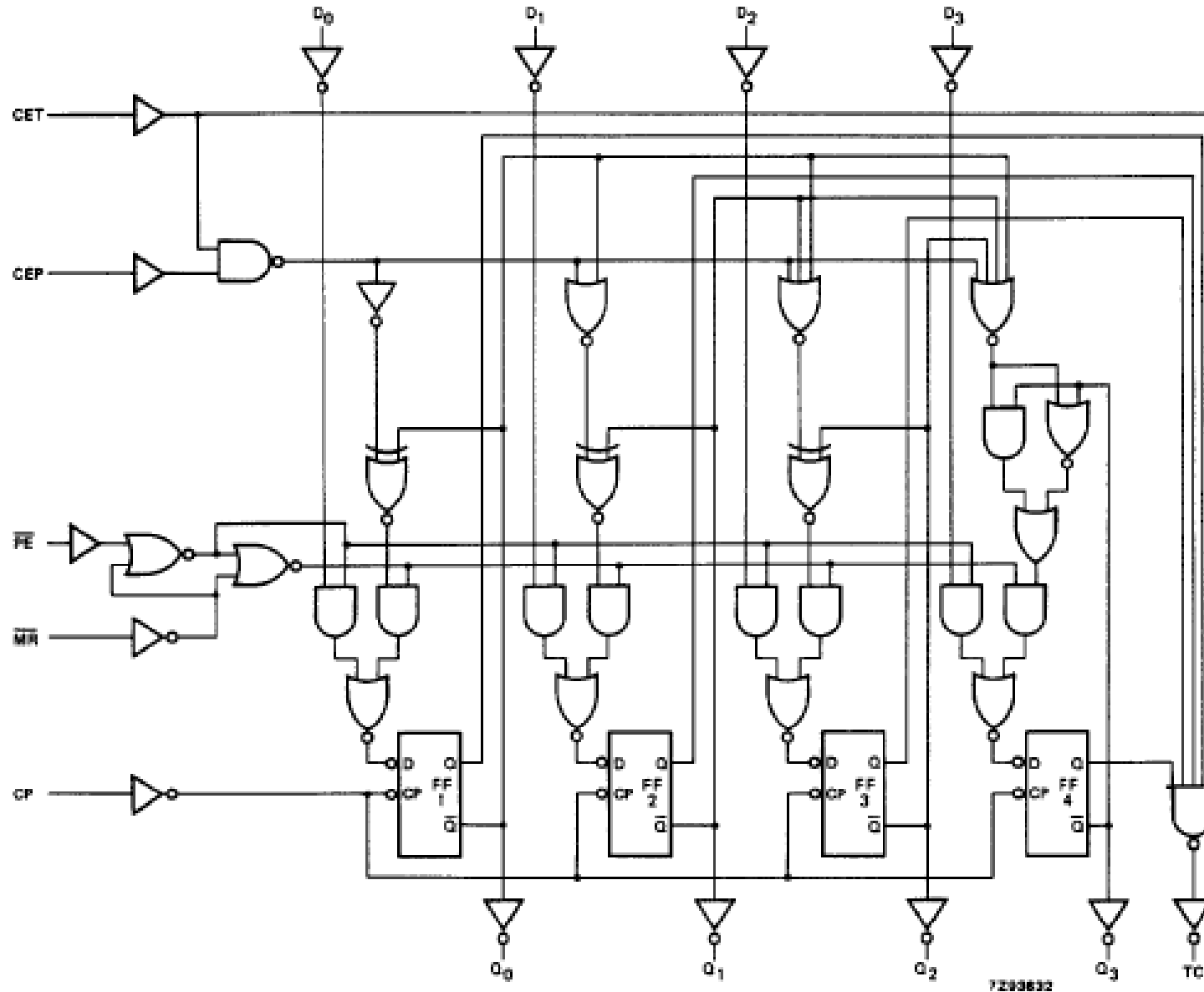
• Segnali

- **!MR** Master Reset sincrono
- **CEP** Count Enable Input
- **CET** Count Enable Carry Input
- **CP** Clock (fronte di salita)
- **!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

$$Q0 = !Q0$$

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

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

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

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

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

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

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

$$Q3 = \dots$$

• “pseudolinguaggio”

If $Q0 = '1111'$ then $Q0 \leq '0000'$
 else $Q0 \leq Q0+1;$