

## Reti Logiche Appello del 5 giugno 2008

(D1) Un'interfaccia IF-REV riceve dalla CPU PD-32 l'indirizzo iniziale  $P$  e la lunghezza  $N > 0$  (in byte) di un buffer interamente contenuto in un blocco di *memoria condivisa* (*shared memory*) da 1024 longword. Su comando dalla CPU, l'interfaccia inverte la posizione dei byte all'interno del buffer ( $P[k] \leftrightarrow P[N - k - 1]$ ) e notifica alla CPU il completamento delle operazioni.

Si assuma che:

- il System Clock abbia frequenza di 100 MHz;
- il blocco di memoria condivisa sia allocato all'indirizzo 0FFFFFF00h;
- i moduli con cui è implementata la memoria condivisa abbiano tempo di lettura/scrittura pari a 10 nsec e tempo di hold pari a 2 nsec.

Progettare l'hardware dell'interfaccia e illustrarne le temporizzazioni.

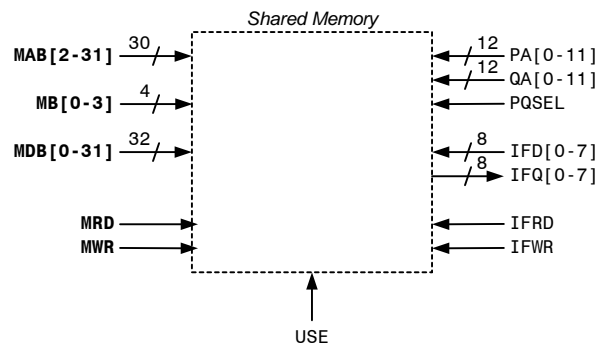
---

### Algoritmo

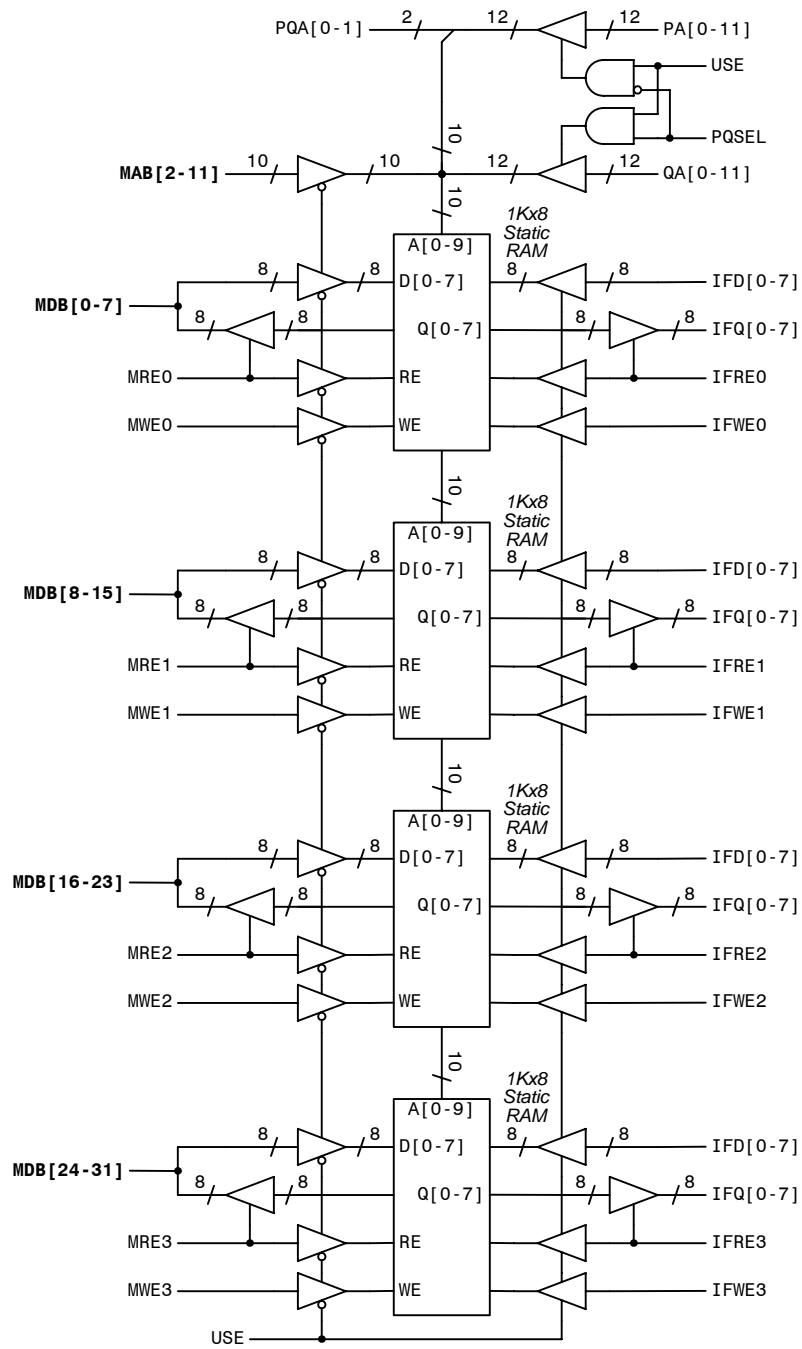
Inizialmente  $P$  contiene l'offset del primo byte del buffer all'interno del blocco di memoria condivisa, e  $N$  contiene la lunghezza del buffer in byte.

```
 $Q \leftarrow P + N$  (offset del primo byte non contenuto nel buffer)
while 1
     $Q \leftarrow Q - 1$ 
    if  $P \geq Q$  then stop
     $T1 \leftarrow \text{Memory}[P]$ 
     $T2 \leftarrow \text{Memory}[Q]$ 
     $\text{Memory}[Q] \leftarrow T1$ 
     $\text{Memory}[P] \leftarrow T2$ 
     $P \leftarrow P + 1$ 
end while
```

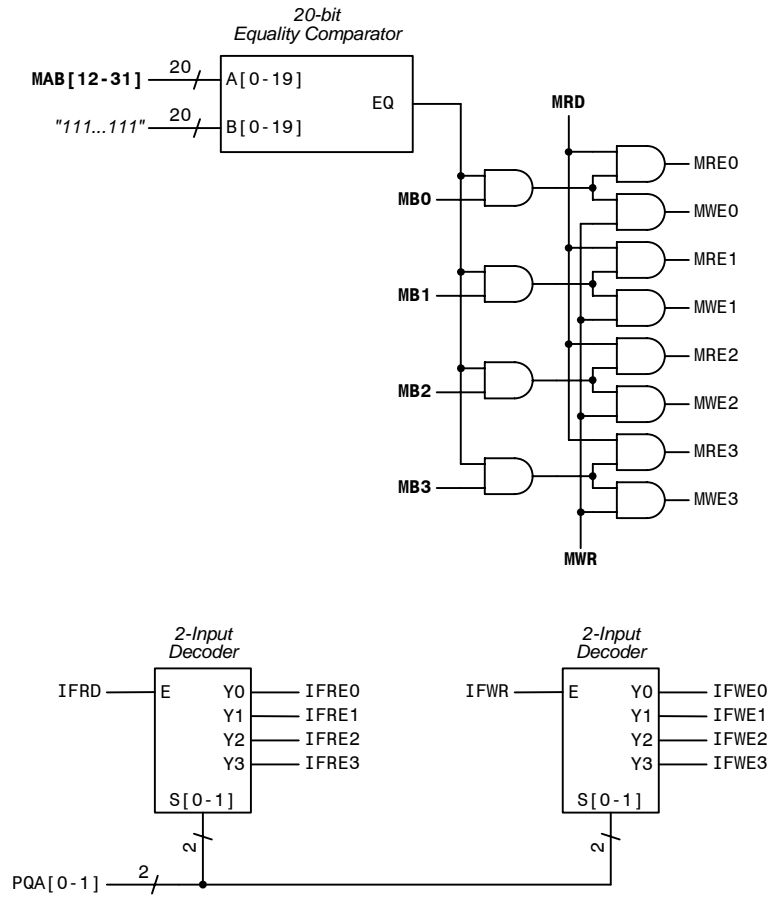
Per minimizzare il numero di stati di SCO, alcune delle operazioni indicate possono essere sovrapposte nel tempo. Inoltre, per semplificare i circuiti di lettura/scrittura nella memoria condivisa,  $T1$  e  $T2$  vengono connessi come un pipeline a due stadi tra l'uscita e l'ingresso della memoria.

**Memoria condivisa – Schema di principio**

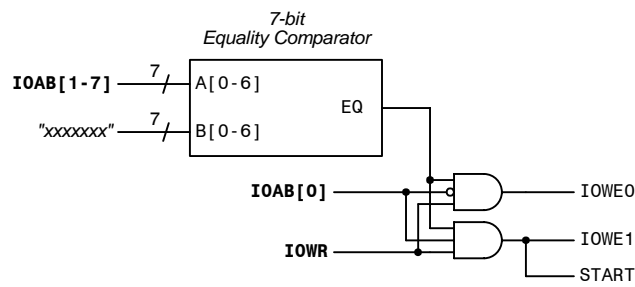
**Memoria condivisa – Realizzazione**



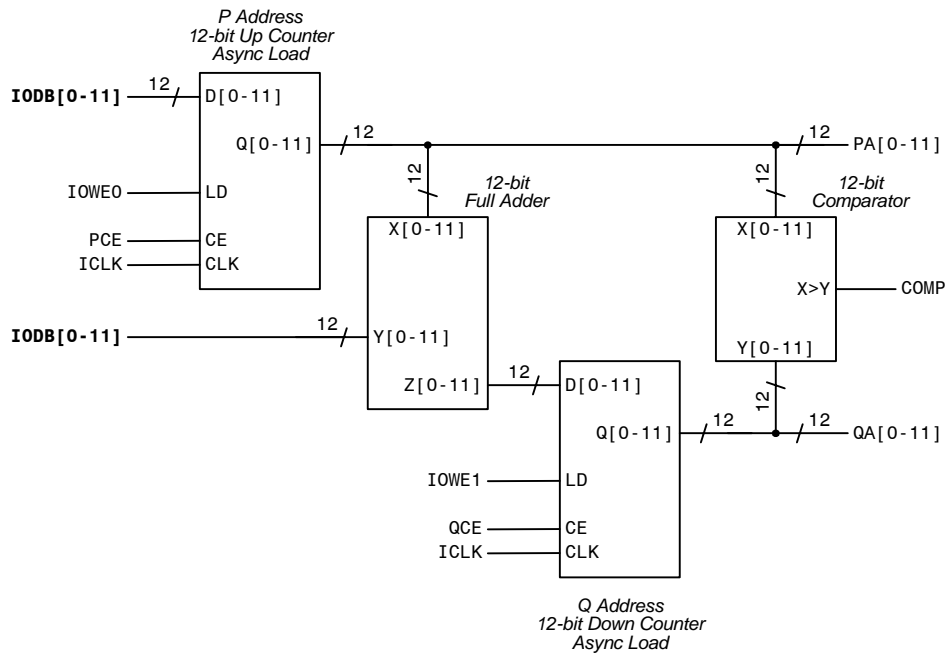
**Memoria condivisa – Circuiti supplementari**



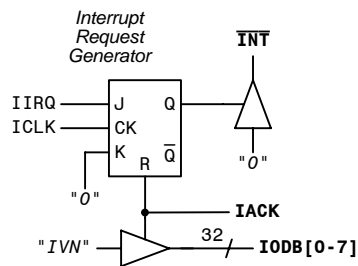
**Decoder degli indirizzi di I/O**



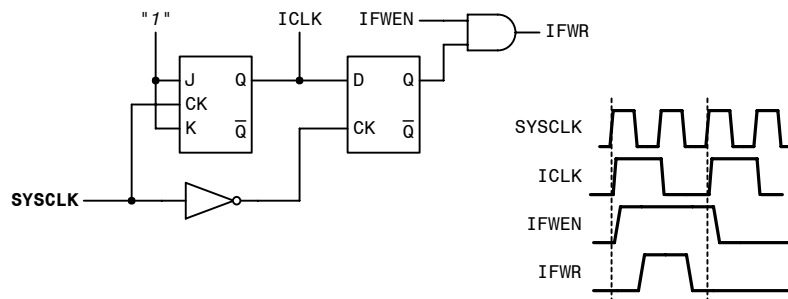
**Porte di I/O – Contatori di indirizzi e comparatore**



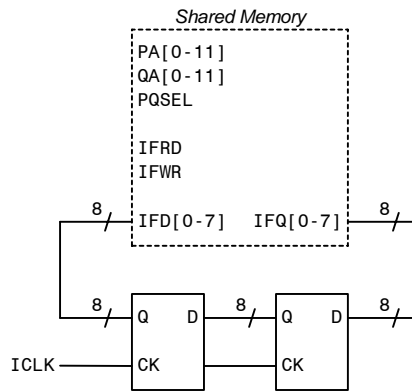
**Generatore di interrupt**



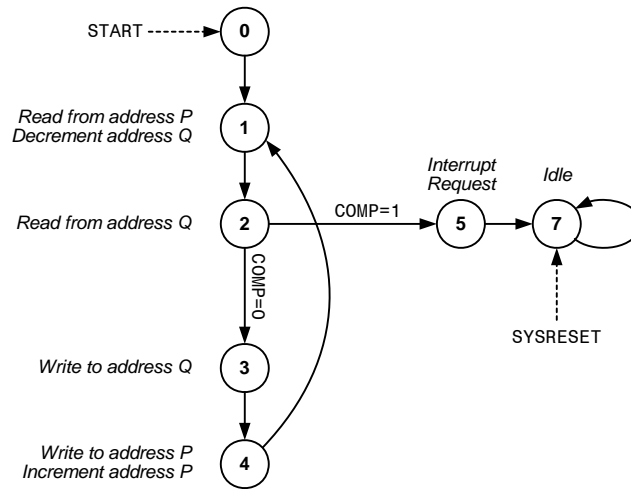
**Generatore di clock**



**Pipeline di lettura/scrittura**



**Sottosistema di Controllo (SCO) – Diagramma di stato**



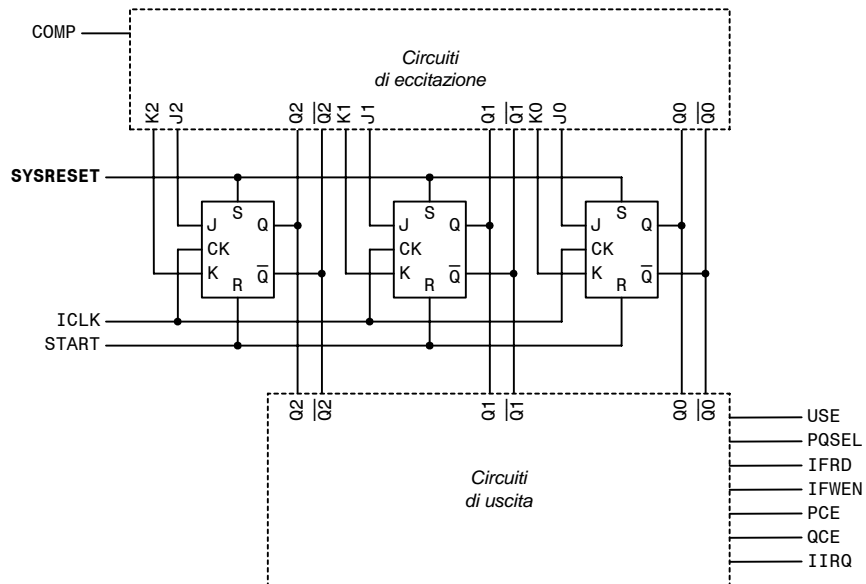
**Sottosistema di Controllo (SCO) – Tavola riassuntiva dei task**

State	USE	PQSEL	IFRD	IFWR	PCE	QCE	IIRQ
0	CPU	-	-	-	off	off	off
1	IF	P	on	off	off	on	off
2	IF	Q	on	off	off	off	off
3	IF	Q	off	on	off	off	off
4	IF	P	off	on	on	off	off
5	-	-	-	off	-	-	on
7	CPU	-	-	-	-	-	off

**Sottosistema di Controllo (SCO) – Realizzazione**

$Q_2Q_1Q_0$	COMP	$Q'_2Q'_1Q'_0$	$J_2K_2$	$J_1K_1$	$J_0K_0$
000	0	001	0-	0-	1-
000	1	001	0-	0-	1-
001	0	010	0-	1-	-1
001	1	010	0-	1-	-1
010	0	011	0-	-0	1-
010	1	101	1-	-1	1-
011	0	100	1-	-1	-1
011	1	100	1-	-1	-1
100	0	001	-1	0-	1-
100	1	001	-1	0-	1-
101	0	111	-0	1-	-0
101	1	111	-0	1-	-0
110	0	---	---	---	---
110	1	---	---	---	---
111	0	111	-0	-0	-0
111	1	111	-0	-0	-0

$Q_2Q_1Q_0$	USE	PQSEL	IFRD	IFWEN	PCE	QCE	IIRQ
000	0	-	-	-	0	0	0
001	1	0	1	0	0	1	0
010	1	1	1	0	0	0	0
011	1	1	0	1	0	0	0
100	1	0	0	1	1	0	0
101	-	-	-	0	-	-	1
110	-	-	-	-	-	-	-
111	0	-	-	-	-	-	0



(D2) Date le funzioni di commutazione

$$f(x, y, w, z) = x(w + \bar{y}z) + \bar{y}(\bar{x}\bar{w}\bar{z} + xz)$$

$$g(x, y, w, z) = w(x + \bar{y}z) + w(\bar{x}\bar{y}z + yz)$$

determinare tutte le funzioni  $h(x, y, w, z)$  tali che

$$f + \bar{h} = g$$

Non esiste alcuna funzione  $h$  che soddisfi il requisito. Si consideri infatti l'espressione  $f\bar{g}$ ; se  $g = f + \bar{h}$  per qualunque combinazione di variabili, si dovrà avere

$$f\bar{g} = \overline{f + \bar{h}} = f\bar{f}h = 0$$

Sviluppando  $f$  e  $\bar{g}$  in somme di prodotti si ottiene:

$$f = xw + \bar{x}\bar{y}\bar{w}\bar{z} + x\bar{y}z$$

$$\bar{g} = x\bar{y}w + x\bar{y}\bar{w} + \bar{x}\bar{y}z + y\bar{w}z$$

da cui segue

$$f\bar{g} = x\bar{y}\bar{w}z$$

Non essendo  $f\bar{g}$  identicamente uguale a 0, si deduce che non esiste alcuna funzione  $h$  che soddisfi  $g = f + \bar{h}$ .

Alla stessa conclusione si può pervenire mediante ispezione delle tavole di verità delle funzioni  $f, g, h$ :

$xywz$	$f$	$g$	$\bar{h}$
0000	1	1	x
0001	0	0	0
0010	0	1	1
0011	0	0	0
0100	0	0	0
0101	0	1	1
0110	0	0	0
0111	0	0	0
1000	0	0	0
1001	1	0	?
1010	1	1	x
1011	1	1	x
1100	0	0	0
1101	0	1	1
1110	1	1	x
1111	1	1	x

per la combinazione di variabili  $xywz = 1001$  non esiste alcun valore di  $\bar{h}$  (e dunque di  $h$ ) in grado di soddisfare  $g = f + \bar{h}$ , poiché  $g(1001) = 0$  e  $f(1001) = 1$ .



(D3) Sintetizzare in logica CMOS la funzione di commutazione

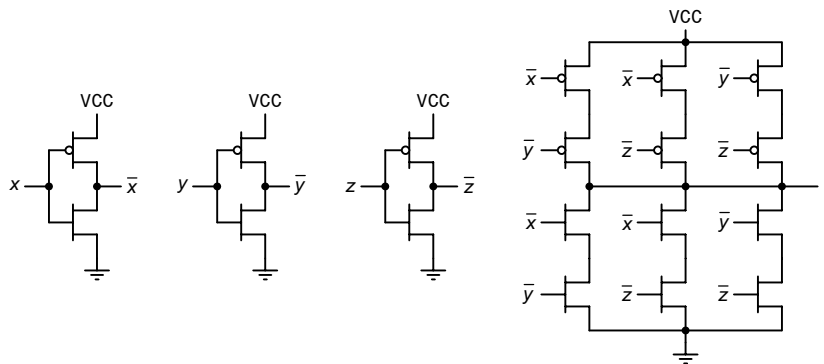
$$f = xy + xz + yz$$


---

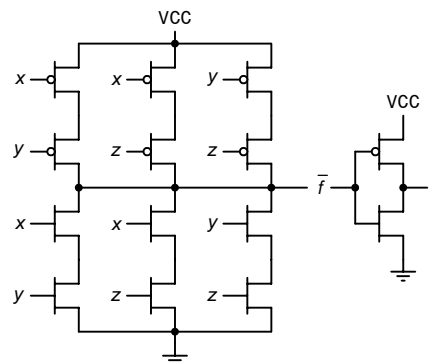
Si tenga presente che

$$\bar{f} = \overline{xy + xz + yz} = \bar{x}\bar{y} + \bar{x}\bar{z} + \bar{y}\bar{z}$$

**Realizzazione diretta:**



**Realizzazione ottimizzata:**



**(D4)** Progettare un circuito sequenziale sincrono che accetti in ingresso, a partire dal meno significativo, i bit  $x_0, x_1, x_2, \dots$  di un numero binario assoluto  $N$  e produca in uscita il valore di  $N \bmod 3$ , ossia il resto della divisione intera di  $N$  per 3.

Sia  $N_k$  il numero rappresentato dai  $k$  bit  $x_{k-1}x_{k-2}\dots x_1x_0$ , e  $\hat{N}_k = N_k \bmod 3$ . Allora

$$N_k = N_{k-1} + 2^k x_k$$

da cui

$$\begin{aligned} \hat{N}_k &= N_k \bmod 3 = \\ &= (N_{k-1} + 2^k x_k) \bmod 3 = \\ &= [N_{k-1} \bmod 3 + x_k (2^k \bmod 3)] \bmod 3 = \\ &= \begin{cases} (\hat{N}_{k-1} + x_k) \bmod 3 & \text{per } k \text{ pari} \\ (\hat{N}_{k-1} + 2x_k) \bmod 3 & \text{per } k \text{ dispari} \end{cases} \end{aligned}$$

In termini di tavola di transizione:

$\hat{N}_{k-1}$	$x_k$	
	0	1
0	0	1
1	1	2
2	2	0

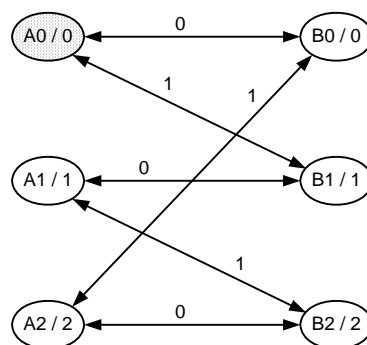
$\hat{N}_k$  ( $k$  pari)

$\hat{N}_{k-1}$	$x_k$	
	0	1
0	0	2
1	1	0
2	2	1

$\hat{N}_k$  ( $k$  dispari)

Poiché  $\hat{N}_k$  può valere soltanto 0, 1 oppure 2, la macchina sequenziale relativa può essere realizzata con due insiemi **A**, **B** da tre stati ciascuno, in modo che ad ogni bit di ingresso  $x_k$  lo stato della macchina evolve da uno stato di **A** ad uno stato di **B** o viceversa.

### Diagramma di stato



**Tavola di transizione**

stato presente	$Q_2$	$Q_1$	$X$	stato futuro	$Q_2'$	$Q_1'$	$Y_0$
A0	000	0		B0	100		00
	000	1		B1	101		01
A1	001	0		B1	101		01
	001	1		B2	110		10
A2	010	0		B2	110		10
	010	1		B0	100		00
	011	0		--	---		--
	011	1		--	---		--
B0	100	0		A0	000		00
	100	1		A2	010		10
B1	101	0		A1	001		01
	101	1		A0	000		00
B2	110	0		A2	010		10
	110	1		A1	001		01
	111	0		--	---		--
	111	1		--	---		--

**Realizzazione con flip-flop D – Mappe di Karnaugh**

	$Q_2 Q_1$			
$Q_0 X$	00	01	11	10
00	1	1		
01	1	1		
11	1	-	-	
10	1	-	-	

$$D_2 = \bar{Q}_2 \bar{Q}_1$$

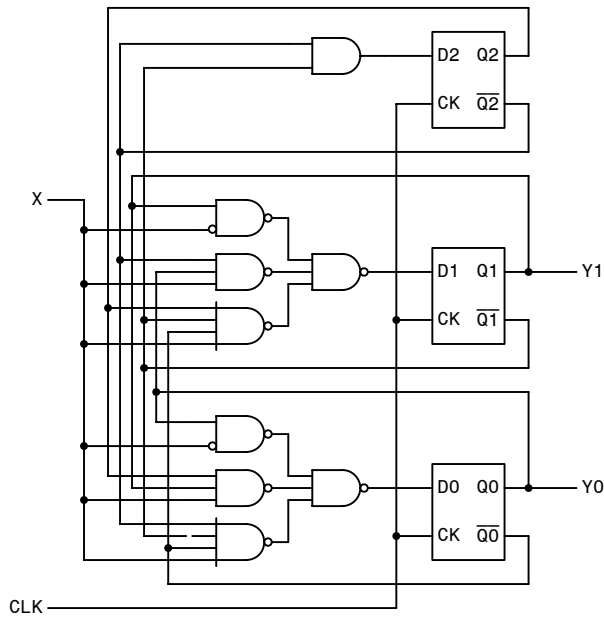
	$Q_2 Q_1$			
$Q_0 X$	00	01	11	10
00		1	1	
01				1
11	1	-	-	
10		-	-	

$$D_1 = Q_1 \bar{x} + \bar{Q}_2 Q_0 x + Q_2 \bar{Q}_1 \bar{Q}_0 x$$

	$Q_2 Q_1$			
$Q_0 X$	00	01	11	10
00				
01	1		1	
11		-	-	
10	1	-	-	1

$$D_0 = Q_0 \bar{x} + Q_2 Q_1 x + \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 x$$

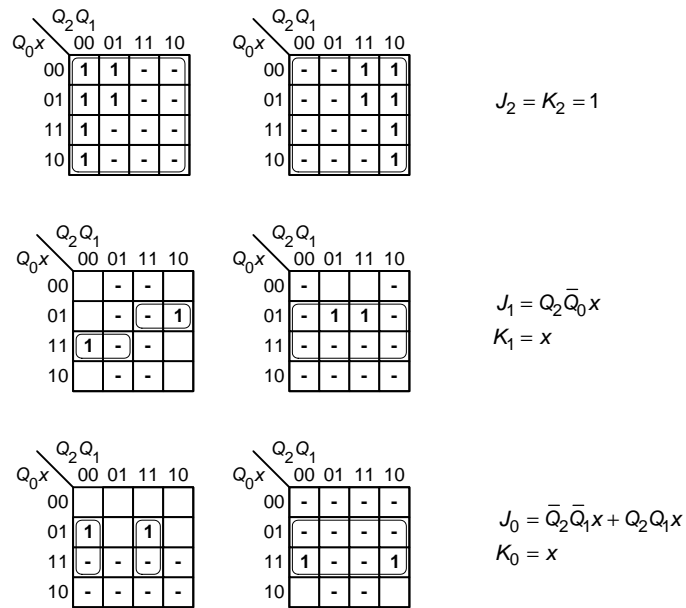
**Realizzazione con flip-flop D – Circuito finale**



**Realizzazione con flip-flop JK – Tavola di transizione-eccitazione**

stato presente	$Q_2 Q_1 Q_0$	X	stato futuro	$Q_2' Q_1' Q_0'$	$J_2 K_2 J_1 K_1 J_0 K_0$	$Y_{10}$
A0	000	0	B0	100	1- 0- 0-	00
	000	1	B1	101	1- 0- 1-	01
A1	001	0	B1	101	1- 0- -0	01
	001	1	B2	110	1- 1- -1	10
A2	010	0	B2	110	1- -0 0-	10
	010	1	B0	100	1- -1 0-	00
	011	0	--	---	-- -- --	--
	011	1	--	---	-- -- --	--
B0	100	0	A0	000	-1 0- 0-	00
	100	1	A2	010	-1 1- 0-	10
B1	101	0	A1	001	-1 0- -0	01
	101	1	A0	000	-1 0- -1	00
B2	110	0	A2	010	-1 -0 0-	10
	110	1	A1	001	-1 -1 1-	01
	111	0	--	---	-- -- --	--
	111	1	--	---	-- -- --	--

**Realizzazione con flip-flop JK – Mappe di Karnaugh**



**Realizzazione con flip-flop JK – Circuito finale**

