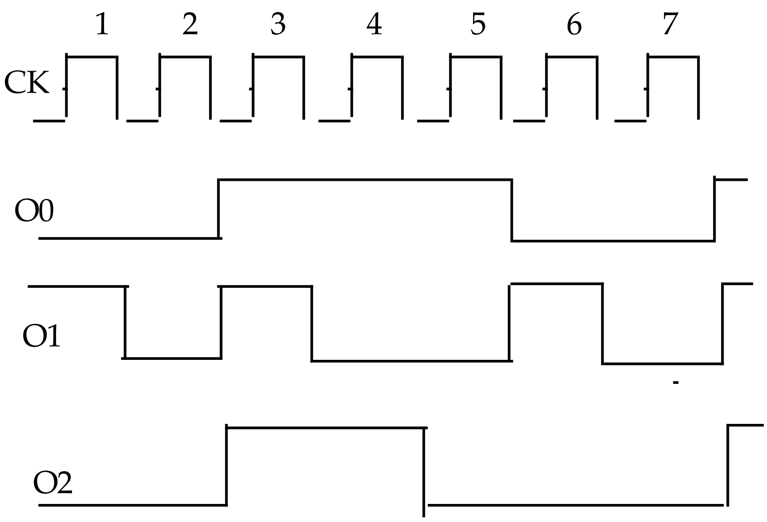
Esercizio 1.

Sintetizzare un circuito sequenziale sincrono in base alle specifiche temporali riportate nel seguito. Il circuito riceve in input solo il segnale di temporizzazione (CK) e produce tre uscite, O_0 O_1 ed O_2.



Utilizzare FF di tipo D (come ovvio dalla figura, sensibili al fronte di discesa del clock). Progettare il circuito con un PLA.

Esercizio

Seguendo lo schema di sintesi, progettare e disegnare la rete sequenziale con flip-flop di tipo D in grado di riconoscere le sequenze 100 e 111

a) **con** eventuali sovrapposizioni,
b) **senza** sovrapposizioni.

Disegnare il circuito con PLA.

Disegnare il diagramma temporale dei FF Qi e dell'uscita a fronte della sequenza di ingresso: 0010001111100 nel caso di sovrapposizioni.

Esercizio

Un codice Gray è una sequenza di numeri binari con la proprietà che tra due numeri consecutivi della sequenza può cambiare un solo bit. Per un codice Gray a tre bit la sequenza è: 000 001 011 010 110 111 101 100 .
Progettare un circuito che emetta (ciclicamente) in uscita un Codice Gray a 3 bit. Usare flip flop JK ed un PLA per la parte combinatoria. Il circuito deve ricevere in ingresso un segnale di controllo "inc". Il codice in uscita cambia sul fronte di salita del segnale inc. Fornire dapprima le specifiche in termini di automa a stati finiti.

Esercizio

Progettare col metodo visto a lezione un contatore che produca in uscita la seguente sequenza ciclica: 0011, 1010, 1111,0000. Usare FF di tipo SR, tracciando prima, come parte del compito, la tabella delle transizioni di un FF di tipo SR. Realizzare il circuito utilizzando per la parte combinatoria un PLA.

Soluzione a (parziale)

Soluzione. Si tratta di un contatore di sequenze, le cui commutazioni avvengono sul fronte di discesa del clock. La sequenza contata é 000 $\rightarrow$111$\rightarrow$101 $\rightarrow$001$\rightarrow$ 010 $\rightarrow$000
La soluzione più semplice consiste nel far coincidere la codifica degli stati con il valore della sequenza contata (e quindi dell'output), utilizzando una macchina di Moore le cui transizioni avvengono in corrispondenza di ogni fronte $\overline{\ }$.
Si costruisce la tabella degli stati futuri nel seguente modo (ad esempio utilizzando FF di tipo D)
La prima e terza colonna della tabella rappresentano una tabella di verità da cui ricavare, col metodo di Karnaugh, le funzioni trasferimento per $D_2D_1D_0$, e quindi il circuito.
Le configurazioni non incluse nella sequenza possono essere associate a transizioni in stati qualsiaisi (X) ed in tal modo é possibile ottenere espressioni semplificate per $D_2D_1D_0$.
Alternativamente, si possono "forzare" transizioni dagli stati non consentiti ad uno stato della sequenza, ad esempio 000.

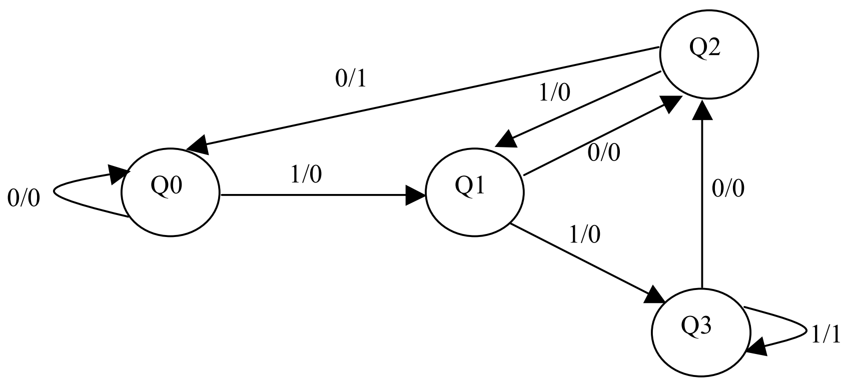
S(t) $Q_2Q_1Q_0$	S(t+1) $Q_2Q_1Q_0$	D(t) $D_2D_1D_0$
000	111	111
001	010	010
010	000	000
011	XXX	XXX
100	XXX	XXX
101	001	001
110	XXX	XXX
111	101	101

Risolvendo con le mappe di Karnaugh e sfruttando le condizioni di indifferenza, si ottiene:

$$D0 = Q2 + \overline{Q}1\overline{Q}0$$
$$D1 = \overline{Q}2$$
$$D2 = \overline{Q}1\overline{Q}0 + Q2Q1$$

Soluzione b

a) L'automa nel caso di riconoscimento di sequenze con sovrapposizioni è:



La tabella di verità è:

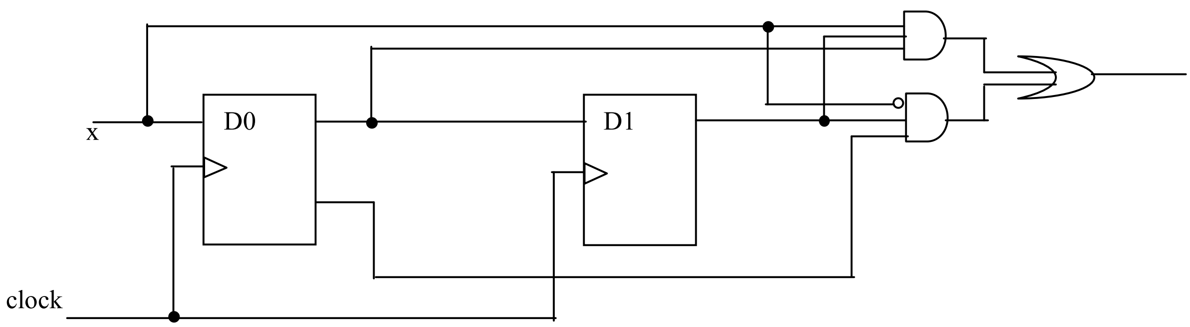
x	q1	q0	Q1	Q0	z	D1	D0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0
1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1
1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	1

Usando le mappe di Karnaugh (o semplicemente osservando la tabella) si ottengono le espressioni minimizzate:

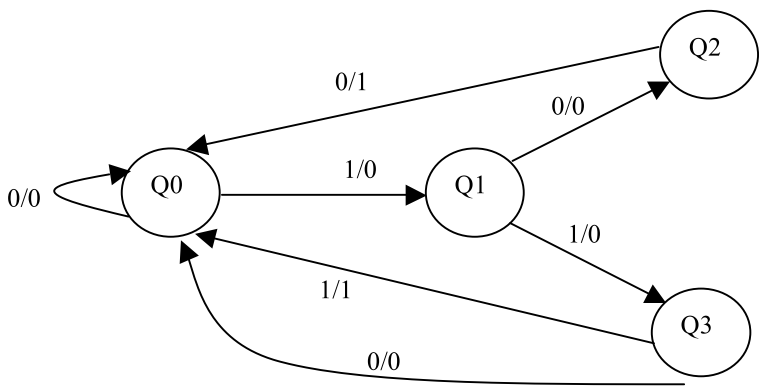
D1 = q0
D2 = x

La funzione di uscita è: $z = \underline{x}\ q1\ \underline{q0} + x\ q1\ q0$

Il circuito è:



b) Nel caso di riconoscimento di sequenze senza sovrapposizioni, l'automa è:



La tabella di verità è:

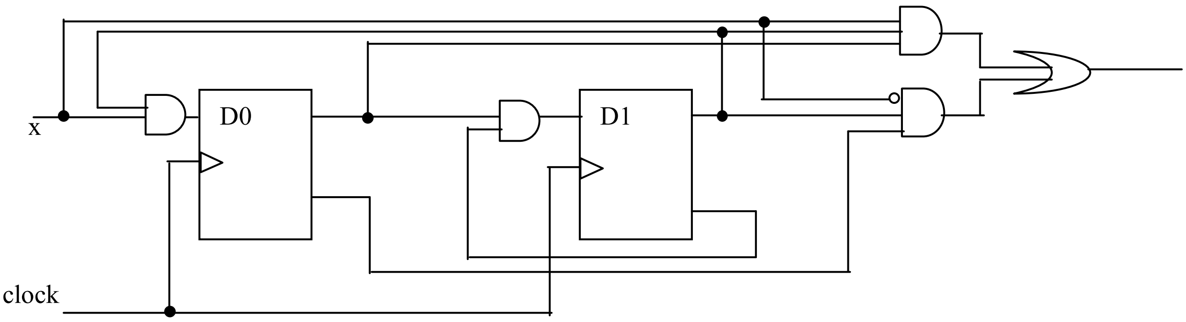
x	q1	q0	Q1	Q0	z	D1	D0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1
1	1	0	0	0	0	0	0
1	1	1	0	0	1	0	0

Usando le mappe di Karnaugh si ottengono le espressioni minimizzate:

$D1 = \underline{q1}\ q0$
$D2 = x\ q1$

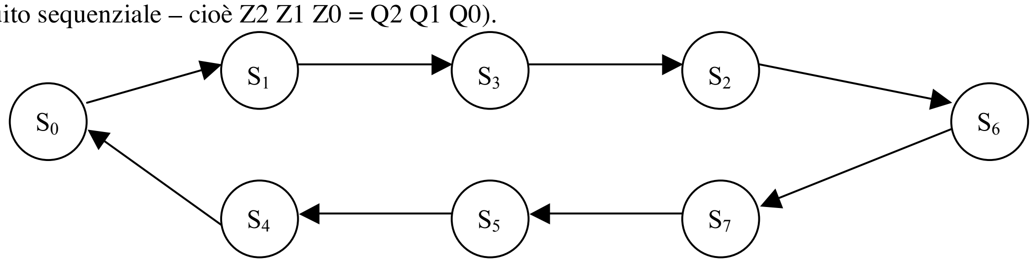
La funzione di uscita è: $z = \underline{x}\ q1\ \underline{q0} + x\ q1\ q0$

Il circuito è:



soluzione c

L'esercizio in realtà è semplicissimo: si deve realizzare un contatore di sequenza non ordinata. Poiché la sequenza è di 8 stringhe da 3 bit, serviranno 3 FF (di tipo JK). L'automa è semplicemente realizzabile con output associati agli stati (Moore) e in cui le variabili di uscita coincidono con le uscite dei FF (a patto di far coincidere stato dei FF con le uscite del circuito sequenziale – cioè Z2 Z1 Z0 = Q2 Q1 Q0).



con la solita associazione S0 → Q2 Q1 Q0 = 000 , S1 → Q2 Q1 Q0 = 001 , …

La tabella degli stati futuri avrà la seguente struttura:

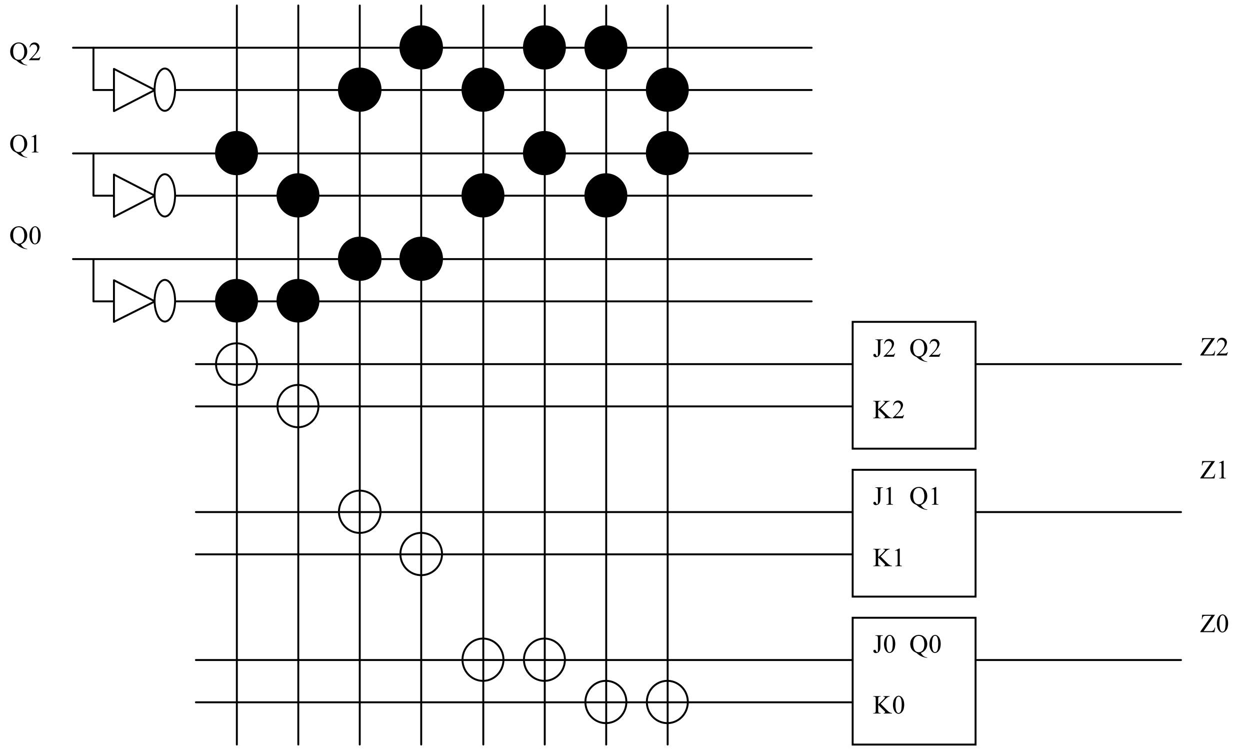
Q2 Q1 Q0 (S(t))	Q2 Q1 Q0 (S(t+1))	J2 K2 (t)	J1 K1 (t)	J0 K0 (t)
0 0 0	0 0 1	0 X	0 X	1 X
0 0 1	0 1 1	0 X	1 X	X 0
0 1 0	1 1 0	1 X	X 0	0 X
0 1 1	0 1 0	0 X	X 0	X 1
1 0 0	0 0 0	X 1	0 X	0 X
1 0 1	1 0 0	X 0	0 X	X 1
1 1 0	1 1 1	X 0	X 0	1 X
1 1 1	1 0 1	X 0	X 1	X 0

da cui (usando le mappe di Karnaugh)

$J2 = Q1\,\overline{Q0}$, $K2 = \overline{Q1}\,\overline{Q0}$, $J1 = \overline{Q2}\,Q0$, $K1 = Q2\,Q0$

$J0 = Q2\,Q1 + \overline{Q2}\,\overline{Q1}$, $K0 = \overline{Q2}\,Q1 + \overline{Q2}\,Q1$

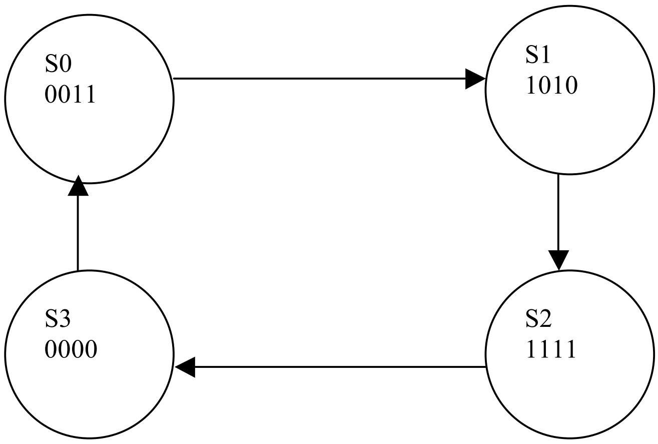
e quindi il PLA è (dove ogni FF ha come ingresso di clock il segnale INC):



Esercizio d

Progettare col metodo visto a lezione un contatore che produca in uscita la seguente sequenza ciclica: 0011, 1010, 1111,0000. Usare FF di tipo SR, tracciando, come parte del compito, la tabella delle transizioni di un FF di tipo SR.

Si tratta di un contatore di sequenza non ordinata. Il contatore "conta" 4 numeri, dunque l'automa ha quattro stati (sia con Mealy che con Moore: qui mostriamo un Moore). Le transizioni avvengono sui fronti di clock.



Le transizioni di un FF SR, tenendo conto che l'input S=R=1 non è ammesso, saranno attivate dai seguenti impulsi SR (X = condizione di indifferenza):

Q(t)→Q(t+1)	S(t) R(t)
0→0	0 X
0→1	1 0
1→0	0 1
1→1	X 0

La tabella degli stati futuri è:

Q1Q0 (t)	S1R1 S0R0	Q1Q0 (t+1)	y3y2y1y0 (t)
S0: 00	0X 10	01	0011
S1: 01	10 01	10	1010
S2: 10	X0 10	11	1111
S3: 11	01 01	00	0000

Da cui:

$$\begin{cases} S1 = Q_0\overline{Q_1} \\ R1 = Q_1Q0 \\ S0 = \overline{Q0} \\ R0 = Q0 \\ y0 = \overline{Q0} \\ y1 = \overline{Q0} + \overline{Q1} \\ y2 = Q1\overline{Q0} \\ y3 = Q1\overline{Q0} + \overline{Q1}Q0 \end{cases}$$

j