

Compito A

Esercizio 1 (15 punti)

Progettare un circuito il cui output è 1 quando viene riconosciuta una delle seguenti stringhe:

00111

00100

00000

00011

l'output è zero altrimenti.

Il primo bit che viene letto è il bit **più a sinistra**.

Le stringhe sono *sovrapponibili*, nel senso chiarito a lezione..

Pur non essendo richiesta l'applicazione di un criterio formale di minimizzazione dell' automa, sarà elemento di valutazione il numero degli stati complessivi utilizzati.

Esercizio 2 (15 punti)

Disegnare il circuito sequenziale il cui funzionamento è descritto dalle seguenti espressioni booleane:

$$J0 = \overline{Q1} \wedge \overline{Q2}$$

$$K0 = \overline{Q2}$$

$$J1 = K1 = 1$$

$$J2 = Q0 \vee \overline{Q1}$$

$$K2 = \overline{Q1}$$

e sintetizzare il relativo automa (procedere come visto a lezione per la derivazione della tabella degli stati futuri).

Compito B

Esercizio 1 (15 punti)

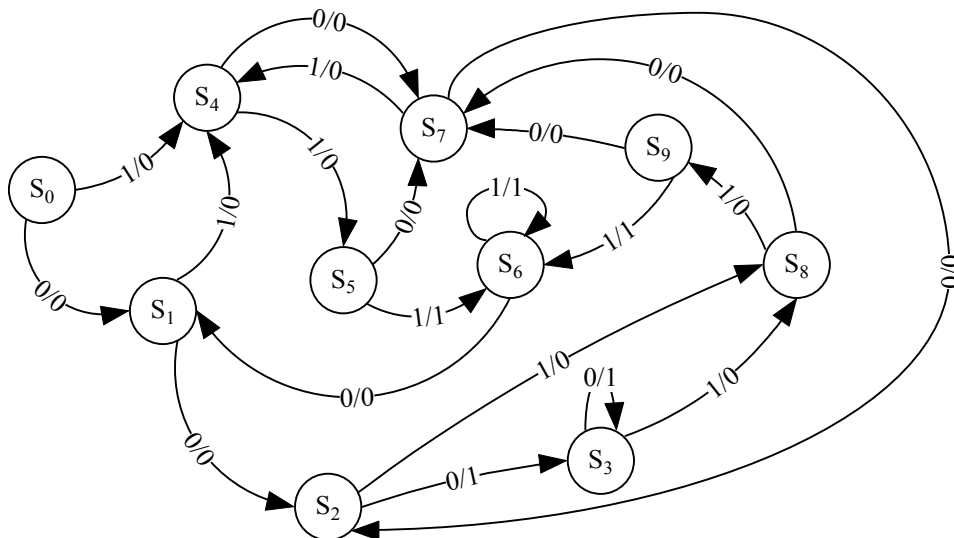
Progettare un registro a 3 bit realizzato con FF di tipo JK, che riceve in ingresso un input sequenziale I, un segnale READ (attivo alto), un segnale SHIFT_LEFT, un segnale SHIFT_RIGHT, ed un clock.

1. Fintanto che il segnale READ è "1" il registro deve leggere i valori di input e memorizzarli, secondo la temporizzazione del clock. (Questo significa che vengono letti tanti bit di input quanti sono i fronti di clock ricevuti nel periodo in cui READ=1) In questa fase il registro si deve comportare come un registro a slittamento a destra (SISO).
2. Fintanto che READ=1 i valori di SHIFT_LEFT e SHIFT_RIGHT non modificano il comportamento del registro, descritto al punto 1.
3. Quando READ=0, se (e fintanto che) SHIFT_LEFT=1 l'informazione memorizzata nel registro viene fatta slittare a sinistra, secondo la temporizzazione del clock.
4. Quando READ=0, se (e fintanto che) SHIFT_RIGHT=1 l'informazione memorizzata nel registro viene fatta slittare a destra, secondo la temporizzazione del clock.
5. SHIFT_LEFT e SHIFT_RIGHT non sono **mai** contemporaneamente ad "1".
6. Poiché durante lo slittamento a destra (o sinistra) il valore dell'input I non viene letto, il bit più a destra (sinistra) viene perso, mentre quello più a sinistra (destra) diventa uno zero.

Rappresentare il funzionamento del circuito con un automa e progettare il circuito.

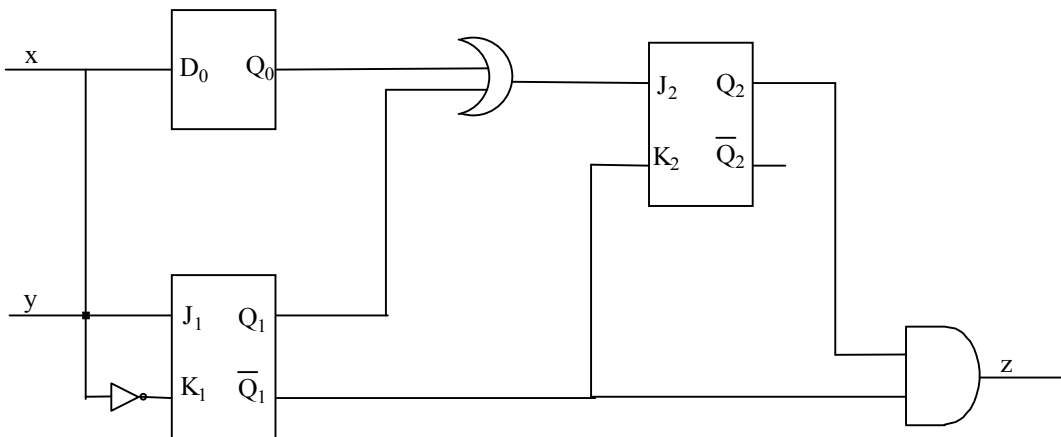
Suggerimento: Idealmente, sarebbe preferibile non rappresentare tutti gli stati dell'automata, ma, a partire da un generico stato in cui il valore memorizzato è $b_1b_2b_3$, esaminare le possibili transizioni in funzione di READ, SHIFT_LEFT e SHIFT_RIGHT, ed I. Se, ad esempio, la stringa viene fatta slittare a sinistra, la nuova stringa sarà b_2b_30 .

Esercizio 2 (15 punti). Minimizzare il seguente automa usando il metodo di minimizzazione (tabella triangolare) visto a lezione. Quali sequenze riconosce l'automata?



Compito C

Esercizio 1 (15 punti). Analizzare il seguente circuito sequenziale, ricavando il diagramma di stato.



Esercizio 2 (15 punti). Sia n il numero memorizzato in un contatore a un generico passo t . Al passo $t+1$ il contatore si comporta come segue:

- se riceve in input 0, memorizza il numero $(n*2) \bmod 8$ ($*$ = moltiplicazione)
- se riceve in input 1, memorizza il numero $((n*2)+1) \bmod 8$

I numeri memorizzati vengono emessi in output.

Si progetti il circuito del contatore (indicate con S_x lo stato in cui $n=x$).

Ad esempio, partendo da S1 e ricevendo in input la sequenza 01001 (primo bit a sinistra), il contatore emette in output la sequenza 2, 5, 2, 4, 1 opportunamente codificata. Si utilizzino FF di tipo JK.

Si ricorda che $N \bmod M$ è definito come il resto della divisione intera tra N e M (ad es., $8 \bmod 5 = 3$ poiché $8 / 5 = 1$ con resto 3 o ancora $6 \bmod 10 = 6$ poiché $6 / 10 = 0$ con resto 6).

Compito D

Esercizio 1 (13 punti)

Progettare un divisore di frequenza, ovvero un circuito che riceve in ingresso un segnale di clock di frequenza f e produce in uscita un segnale di clock di frequenza $\frac{f}{8}$. Ovviamente, la durata dell'impulso è T in ingresso, $T \times 8$ in uscita.

Esercizio 2 (17 punti). La pompa automatica di un distributore di benzina accetta banconote da 5, 10 e 20 € ed eroga per ogni utente benzina fino a 20 € (eventuali banconote in eccesso vengono restituite automaticamente, e monete in eccedenza al valore di 20 € non vengono accettate). Se sono stati inseriti almeno 5 € l'utente può procedere al rifornimento. Il circuito riceve in input anche l'informazione che indica se l'utente ha effettuato il rifornimento. In tal caso il contatore di banconote viene azzerato. Progettare il circuito della pompa di benzina, emettendo in output le banconote da restituire e la possibilità o meno per l'utente di rifornirsi. Si tenga conto che la macchina restituisce il resto utilizzando i tagli massimi di banconote (ad es, un resto di 15 € consiste in un biglietto da 5 € e uno da 10 € e non tre biglietti da 5 €). Si utilizzino FF di tipo JK. Suggerimento: modellare l'input come combinazioni delle possibili informazioni sulle banconote immesse e sull'avvenuto rifornimento, e l'output come possibili resti che devono essere erogati. Chiarite per scritto eventuali ipotesi sulla sequenza degli eventi, per facilitare la correzione del compito.

SOLUZIONI

Compito A

Esercizio 1

Progettare un circuito il cui output è 1 quando viene riconosciuta una delle seguenti stringhe:

00111

00100

00000

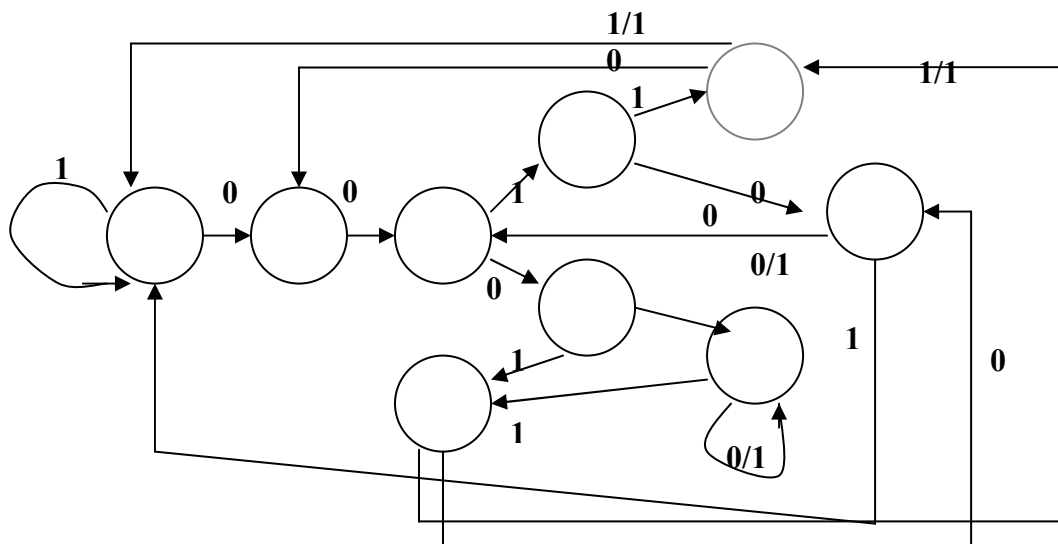
00011

l'output è zero altrimenti.

Il primo bit che viene letto è il bit **più a sinistra**.

Le stringhe sono sovrapponibili –nel senso illustrato a lezione–.

Pur non essendo richiesta l'applicazione di un criterio formale di minimizzazione dell'automa, sarà elemento di valutazione il numero degli stati complessivi utilizzati.



L'automa di Mealy è rappresentato in figura. In figura, per semplicità, l'output è mostrato solo quando assume il valore "1".

L'automa ha 9 stati e dunque occorrono 4FF. L'esercizio va proseguito completando la tabella degli stati futuri (ma la cosa essenziale era il disegno dell'automa).

S0=stato iniziale (nessun inizio stringa rilevato)

S1= stato in cui si è rilevato "0"

S2= stato in cui si è rilevato "00"

S3= stato in cui si è rilevato "001"

S4= stato in cui si è rilevato "000"

S5= stato in cui si è rilevato "0011"

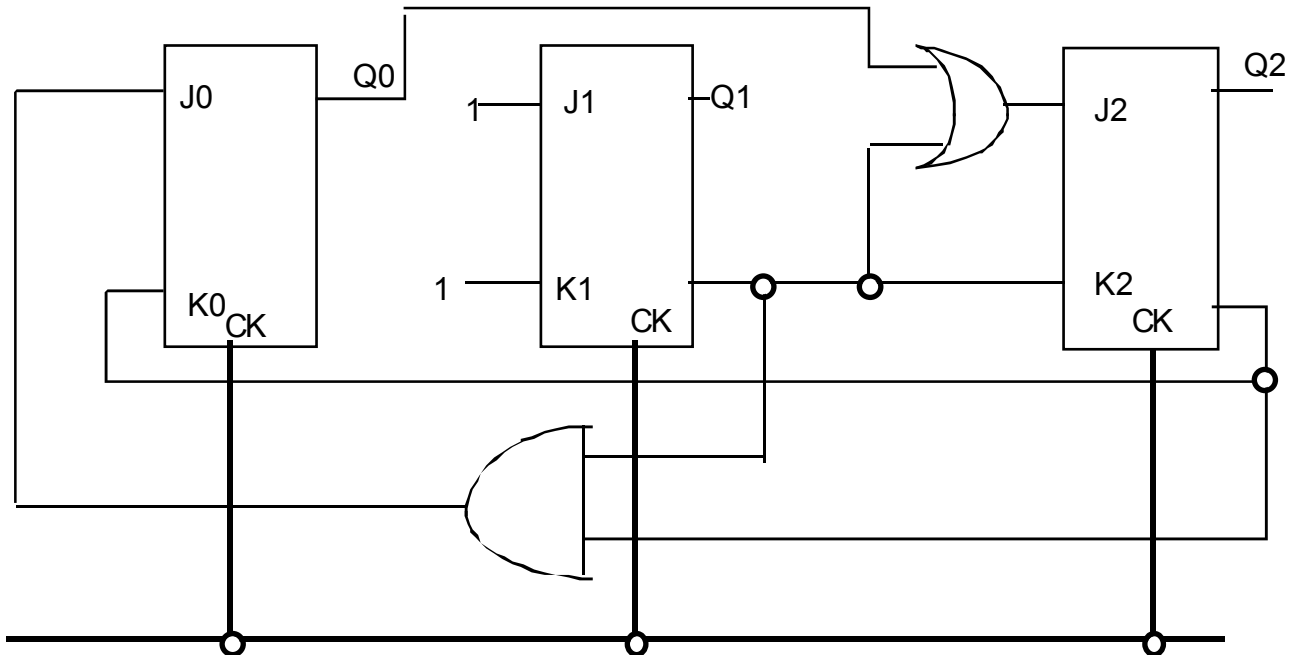
S6 = stato in cui si è rilevato "0010"

S7 = stato in cui si è rilevato "0000"

S8 = stato in cui si è rilevato "0001"

Notate che, ad esempio, da S8 ricevendo uno 0 si va in S6, infatti **00010**

Esercizio 2



S_i ($Q_2Q_1Q_0$) stato di partenza	J_2K_2	J_1K_1	J_0K_0	S_j stato di arrivo
000	11	11	11	111
001	11	11	11	110
010	00	11	01	000
011	10	11	01	100
100	11	11	00	010
101	11	11	00	011
110	00	11	00	100
111	10	11	00	101

La sequenza contata è 000 111 101 011 100 010 000...

Si noti che, qualora il sistema erroneamente ricada in uno stato non incluso nella sequenza, comunque "ricade" successivamente nel ciclo previsto.

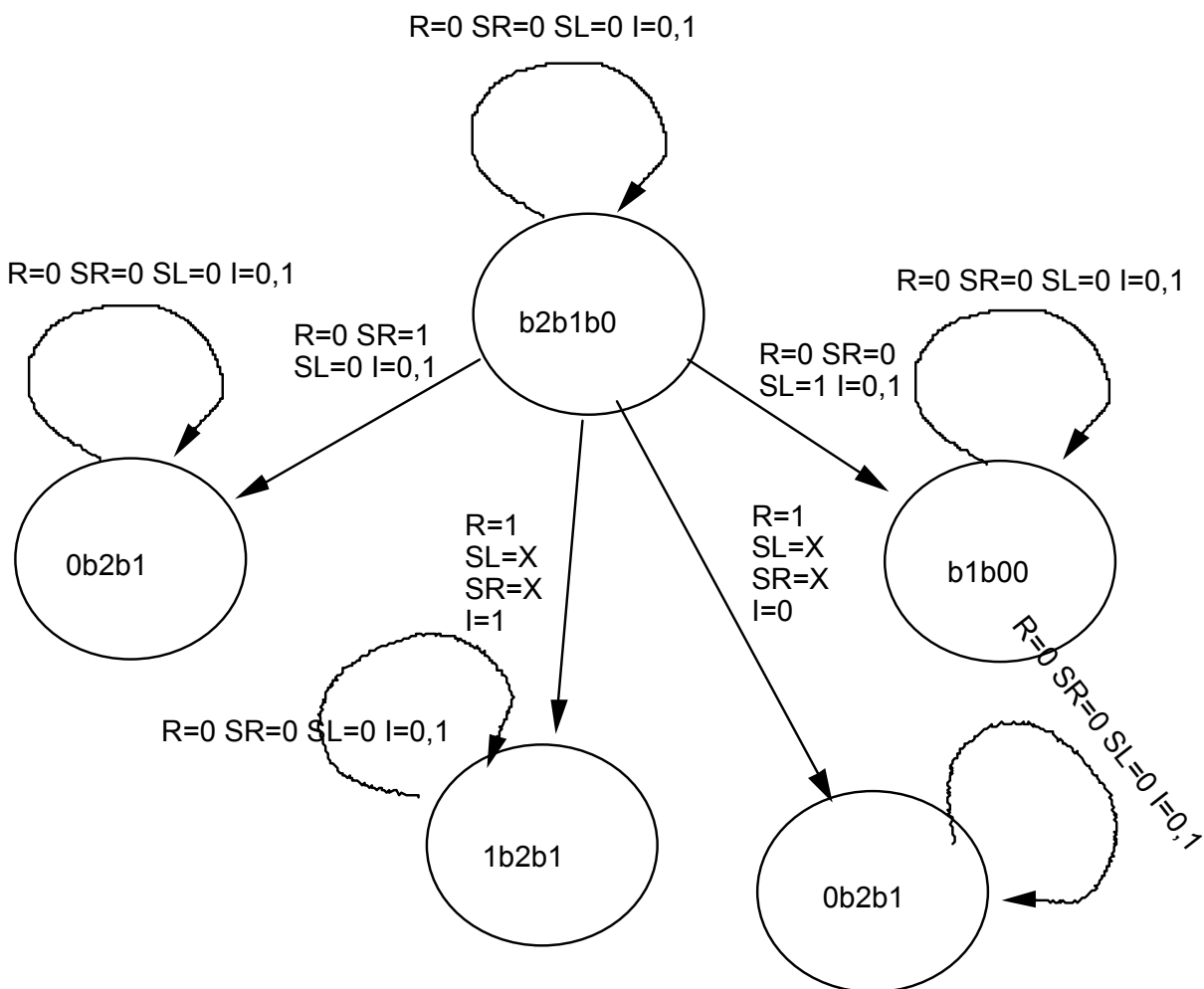
Compito B

Esercizio 1

Il registro shift left shift right è stato descritto a lezione. Una figura molto simile a quella che vedete è la 3.11 dei miei appunti (parte terza).

Considerando ad es. un registro a 3 bit, e considerando un generico stato del registro (poiché l'output di un registro coincide con lo stato della memoria conviene usare Moore), le transizioni da tale stato generico sono illustrate nel seguito:

La figura mostra le transizioni dell'automa a partire da un generico stato in cui i FF memorizzano i bit $b_1b_2b_0$ (gli input sono: SL =shift left, SR= shift right, R= read, I=input, X indica una condizione di indifferenza)



R=read SL=SHIFT_LEFT SR=SHIFT_RIGHT I=Input

Lo schema circuitale può essere ricavato nel solito modo, ma è possibile anche ricavare le espressioni booleane di J_iK_i scrivendo in forma di espressione booleana la specifica verbale.

Ad esempio, l'input J_0 del FF più a sinistra è pari ad I (input) se $Read=1$, è pari a Q_1 se $SL=1$, ed è pari a zero se $SR=1$, dunque:

$$J_0 = R \cdot I + \text{not}(R) \cdot (SL \cdot Q1 + SR \cdot 0)$$

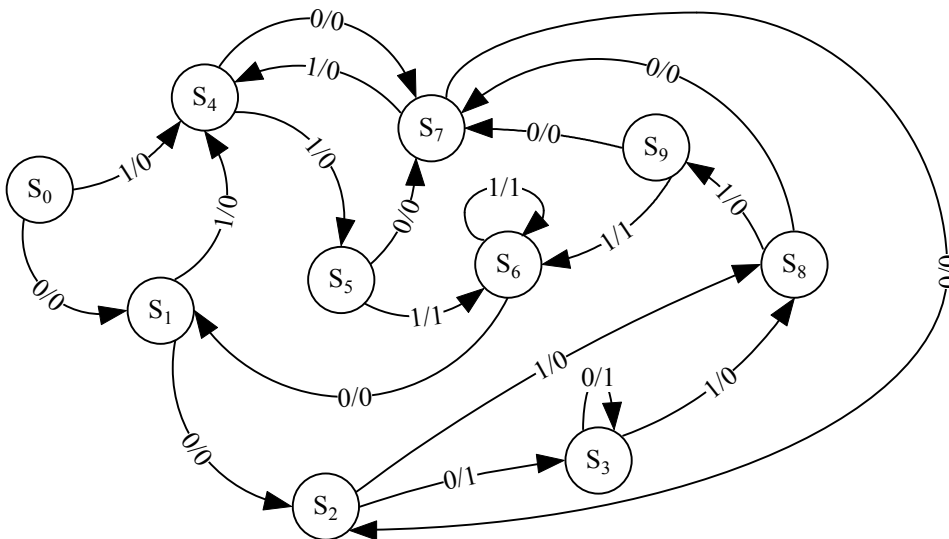
$$J_i = R \cdot Q_{i-1} + \text{not}(R) \cdot (SL \cdot Q_{i+1} + SR \cdot Q_{i-1})$$

$$J_n = R \cdot Q_{i-1} + \text{not}(R) \cdot (SL \cdot 0 + SR \cdot Q_{i-1})$$

Notate che, se $R=SL=SR=0$, lo stato dei FF non cambia ($J=0, K=0$).

Esercizio 2

Minimizzare il seguente automa usando il metodo di minimizzazione (tabella triangolare) visto a lezione. Quali sequenze riconosce l'automata?



Soluzione

Impostando la tabella triangolare risultano indistinguibili i seguenti insiemi di stati:

[1, 7], [2, 3], [4, 8], [5, 6, 9]

Le indistinguibilità vengono individuate nel seguente ordine:

[1, 7], [2, 3] e [5, 9] a fronte degli stessi input transitano negli stessi stati con gli stessi output

[5, 6] a seguito dell'indistinguibilità di [1, 7]

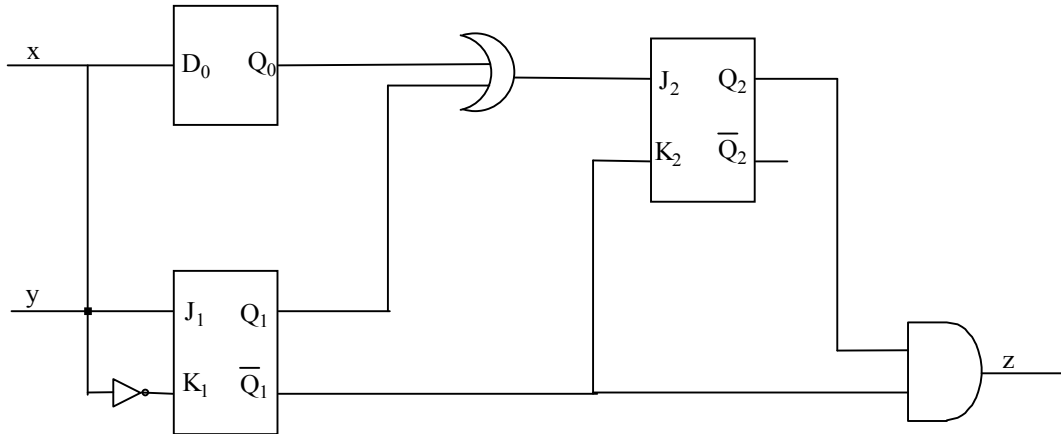
[6, 9] a seguito dell'indistinguibilità di [1, 7]

[4, 8] a seguito dell'indistinguibilità di [5, 9]

L'automata riconosce stringhe binarie che terminano in 000 oppure in 111.

Compito C

Esercizio 1. Analizzare il seguente circuito sequenziale, ricavando il diagramma di stato.



Soluzione Es. 1

Si ha:

$$D_0 = x$$

$$J_1 = y$$

$$K_1 = \bar{x}$$

e quindi:

$$J_2 = Q_0 + Q_1$$

$$K_2 = \bar{Q}_1$$

$$z = Q_2 \cdot \bar{Q}_1$$

Il diagramma degli stati è il seguente:

x	y	Q ₂ (t)	Q ₁ (t)	Q ₀ (t)	J ₂	K ₂	J ₁	K ₁	D ₀	Q ₂ (t+1)	Q ₁ (t+1)	Q ₀ (t+1)	Z
0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	1	1	1	0	1	0	1	0	0	0
0	0	0	1	0	1	0	0	1	0	1	0	0	0
0	0	0	1	1	1	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	1	0	0	0	0	1
0	0	1	0	1	1	1	0	1	0	0	0	0	1
0	0	1	1	0	1	0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	0	1	0	1	0	0	0
0	1	0	0	0	0	1	1	1	0	0	1	0	0
0	1	0	0	1	1	1	1	1	0	1	1	0	0
0	1	0	1	0	1	0	1	1	0	1	0	0	0
0	1	0	1	1	1	0	1	1	0	1	0	0	0
0	1	1	0	0	0	1	1	1	0	0	1	0	1

0	1	1	0	1	1	1	1	1	0	0	1	0	1
0	1	1	1	0	1	0	1	1	0	1	0	0	0
0	1	1	1	1	1	0	1	1	0	1	0	0	0
1	0	0	0	0	0	1	0	0	1	0	0	1	0
1	0	0	0	1	1	1	0	0	1	1	0	1	0
1	0	0	1	0	1	0	0	0	1	1	1	1	0
1	0	0	1	1	1	0	0	0	1	1	1	1	0
1	0	1	0	0	0	1	0	0	1	0	0	1	1
1	0	1	0	1	1	1	0	0	1	0	0	1	1
1	0	1	1	0	1	0	0	0	1	1	1	1	0
1	0	1	1	1	1	1	0	0	1	1	1	1	0
1	1	0	0	0	0	1	1	0	1	0	1	1	0
1	1	0	0	1	1	1	1	0	1	1	1	1	0
1	1	0	1	0	1	0	1	0	1	1	1	1	0
1	1	0	1	1	1	1	0	1	0	1	1	1	0
1	1	1	0	0	0	1	1	0	1	0	1	1	1
1	1	1	0	1	1	1	1	0	1	0	1	1	1
1	1	1	1	0	1	0	1	0	1	1	1	1	0
1	1	1	1	1	1	0	1	0	1	1	1	1	0

Soluzione Es. 2

Esercizio 2. Sia n il numero memorizzato in un contatore a un generico passo t . Al passo $t+1$ il contatore si comporta come segue:

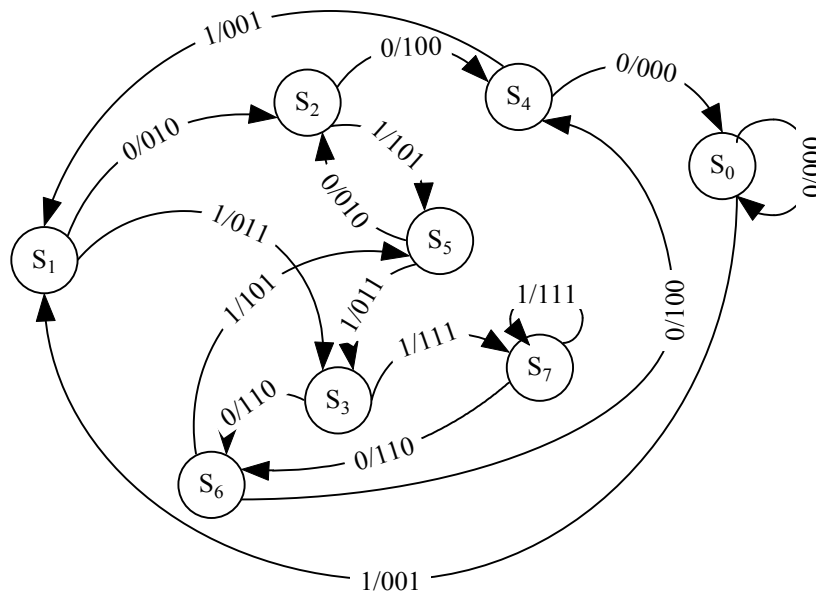
- c) se riceve in input 0, memorizza il numero $(n*2) \bmod 8$
- d) se riceve in input 1, memorizza il numero $((n*2)+1) \bmod 8$

I numeri memorizzati vengono emessi in output. Si progetti il circuito del contatore. Ad esempio, ricevendo in input la sequenza 01001, il contatore emette in output la sequenza 2, 5, 2, 4, 1 opportunamente codificata. Si utilizzino FF di tipo JK.

Si ricorda che $N \bmod M$ è definito come il resto della divisione intera tra N e M (ad es., $8 \bmod 5 = 3$ poiché $8 / 5 = 1$ con resto 3 o ancora $6 \bmod 10 = 6$ poiché $6 / 10 = 0$ con resto 6).

Soluzione Es. 2

Si potrebbe optare per l'automa di Moore, considerando il fatto che gli stati codificherebbero le uscite, ma poi non potremmo minimizzare l'automa. L'automa di Mealy è il seguente:



L'automata può essere minimizzato. Si ha:

Stato	Input 0	Input 1
S ₀	S ₀ /0	S ₁ /1
S ₁	S ₂ /2	S ₃ /3
S ₂	S ₄ /4	S ₅ /5
S ₃	S ₆ /6	S ₇ /7
S ₄	S ₀ /0	S ₁ /1
S ₅	S ₂ /2	S ₃ /3
S ₆	S ₄ /4	S ₅ /5
S ₇	S ₆ /6	S ₇ /7

Applicando il metodo della tabella triangolare si ottiene l'automata minimizzato con i seguenti stati:
 $S_0' = [S_0, S_4]$, $S_1' = [S_1, S_5]$, $S_2' = [S_2, S_6]$, $S_3' = [S_3, S_7]$.

Stato	Input 0	Input 1
S ₀ '	S ₀ '/0	S ₁ '/1
S ₁ '	S ₂ '/2	S ₃ '/3
S ₂ '	S ₀ '/4	S ₁ '/5
S ₃ '	S ₂ '/6	S ₃ '/7

Sono sufficienti due FF per la codifica degli stati. Il diagramma degli stati futuri è il seguente:

Stato	Q ₁ (t)	Q ₀ (t)	x	J ₁	K ₁	J ₀	K ₀	Q ₁ (t+1)	Q ₀ (t+1)	z ₂	z ₁	z ₀
S ₀ '	0	0	0	0	X	0	X	0	0	0	0	0
	0	0	1	0	X	1	X	0	1	0	0	1
S ₁ '	0	1	0	1	X	X	1	1	0	0	1	0
	0	1	1	1	X	X	0	1	1	0	1	1
S ₂ '	1	0	0	X	1	0	X	0	0	1	0	0
	1	0	1	X	1	1	X	0	1	1	0	1
S ₃ '	1	1	0	X	0	X	1	1	0	1	1	0
	1	1	1	X	0	X	0	1	1	1	1	1

Applicando le mappe di Karnaugh:

$x \backslash Q_1Q_0$	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$$J_1 = Q_0$$

$x \backslash Q_1Q_0$	00	01	11	10
0	X	X	0	1
1	X	X	0	1

$$K_1 = \bar{Q}_0$$

$x \backslash Q_1Q_0$	00	01	11	10
0	0	X	X	0
1	1	X	X	1

$$J_0 = x$$

$x \backslash Q_1Q_0$	00	01	11	10
0	X	1	1	X
1	X	0	0	X

$$K_0 = \bar{x}$$

e banalmente:

$$z_2 = Q_1$$

$$z_1 = Q_0$$

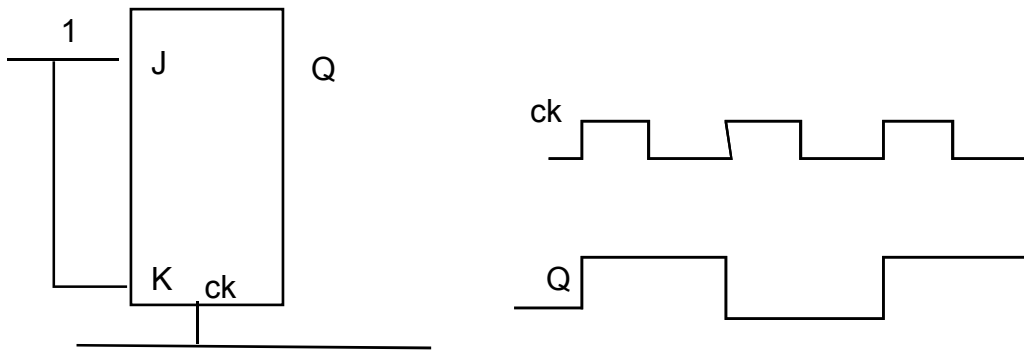
$$z_0 = x$$

Il corrispondente circuito si ricava facilmente.

Compito D

Soluzione Es. 1

Un singolo FF JK connesso come in figura funziona come divisore di frequenza $f/2$:



Banalmente, è possibile realizzare un divisore di frequenza f/n usando $\log_2 n$ FF di tipo JK con $J=K=1$ (oppure Toggle con $T=1$), collegando l'output Q_i del i -esimo FF con il clock del $(i+1)$ -esimo FF.

Tale circuito rappresenta un contatore ASINCRONO (infatti i FF non hanno lo stesso segnale di clock).

Volendo utilizzare un circuito sincrono (ricordatevi che la metodologia di sintesi illustrata a lezione si applica ai soli circuiti sincroni) possiamo ancora usare un contatore (sincrono) con $\log_2 n$ FF ($\log_2 8=3$), sincronizzati sul clock la cui frequenza si desidera dividere.

Formalmente, la tabella degli stati futuri è la seguente (nella tabella sono anche evidenziate le transizioni):

Fronti di clock	Transizioni Dei FF (Q2Q1Q0)	Si → Sj (stato di partenza stato di arrivo)	J2K2	J1K1	J0K0
↑	00↑	000→001	0X	0X	1X
↑	0↑↓	001→010	0X	1X	X1
↑	01↑	010→011	0X	eccetera!!	
↑	↑↓↓	011→110	1X		
↑	10↑	110→101	X0		
↑	1↑↓	101→110	X0		
↑	11↑	110→111	X0		
↑	↓↓↓	111→000	X1		

Da cui ricaviamo:

$$J_0=K_0= 1$$

$$J_1= Q_0$$

$$J_2= Q_0Q_1$$

(in generale, per un contatore sincrono ad n bit, $J_i= \prod Q_j$ ($j=1..i-1$))

Si osservi in tabella (colonne 1 e 2) che Q_2 ha un fronte di salita ogni 8 fronti di salita del clock, come richiesto.

E' banale ricavare lo schema circuitale.

Soluzione Es. 2

La difficoltà principale dell'esercizio consiste nell'individuare l'input e l'output del circuito. Si codifica l'input nel modo seguente:

00 inserito biglietto da 5 euro
01 inserito biglietto da 10 euro
10 inserito biglietto da 20 euro
11 rifornimento effettuato dall'utente

(si potevano anche utilizzare 3 bit: uno per il rifornimento e due per le banconote inserite, ma si sarebbe sprecato un bit, perché l'utente non può inserire banconote mentre completa il rifornimento, e non si sarebbe potuto procedere con le mappe di Karnaugh)

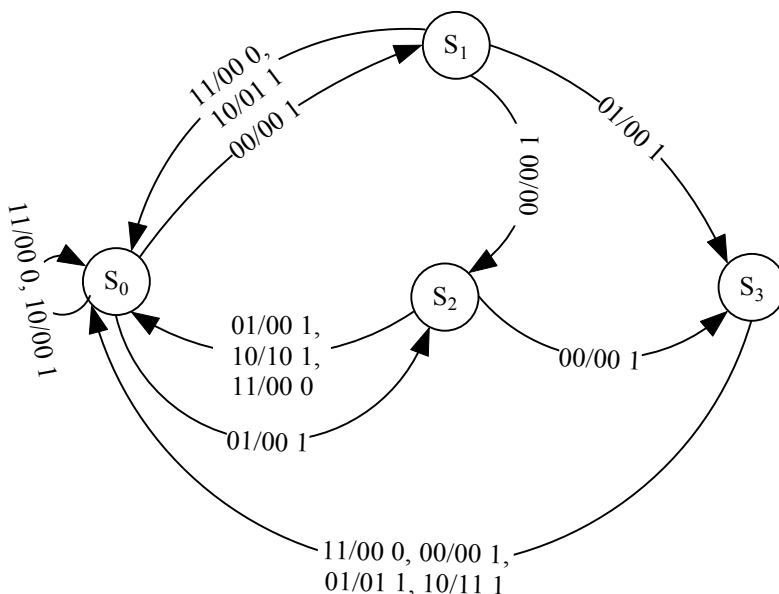
Inoltre il resto (in output) è codificato come segue:

00 nessun resto
01 5 euro
10 10 euro
11 15 euro

e la possibilità di effettuare il rifornimento è segnalata da un terzo bit in output. Notate che quest'ultimo bit è sempre =1 tranne che per gli archi che vanno in S_0 con input 11 (rifornimento effettuato). Gli stati dell'automa sono:

S_0 = 0 oppure 20 euro inseriti
 S_1 = 5 euro inseriti
 S_2 = 10 euro inseriti
 S_3 = 15 euro inseriti

e richiedono, per la loro memorizzazione, 2 bit: $S_0 \rightarrow 00$, $S_1 \rightarrow 01$, $S_2 \rightarrow 10$, $S_3 \rightarrow 11$.
L'automa di Mealy che descrive il circuito è il seguente:



Il fatto che S_0 memorizzi sia lo stato in cui sono stati inseriti 20 euro sia quello in cui non sono state ancora inserite banconote non costituisce un problema, poiché la macchina, una volta che raggiunto l'importo di 20 euro, non accetta più banconote e si mette in attesa dell'avvenuta erogazione (il cappio etichettato con 11/00 0 sullo stato S_0).

La tabella degli stati futuri è la seguente:

$Q_1(t)$	$Q_0(t)$	x_1	x_0	J_1	K_1	J_0	K_0	$Q_1(t+1)$	$Q_0(t+1)$	z_2	z_1	z_0
0	0	0	0	0	X	1	X	0	1	0	0	1
0	0	0	1	1	X	0	X	1	0	0	0	1
0	0	1	0	0	X	0	X	0	0	0	0	1
0	0	1	1	0	X	0	X	0	0	0	0	1
0	1	0	0	1	X	X	1	1	0	0	0	1
0	1	0	1	1	X	X	0	1	1	0	0	1
0	1	1	0	0	X	X	1	0	0	0	1	1
0	1	1	1	0	X	X	1	0	0	0	0	0
1	0	0	0	X	0	1	X	1	1	0	0	1
1	0	0	1	X	1	0	X	0	0	0	0	1
1	0	1	0	X	1	0	X	0	0	1	0	1
1	0	1	1	X	1	0	X	0	0	0	0	0
1	1	0	0	X	1	X	1	0	0	0	0	1
1	1	0	1	X	1	X	1	0	0	0	1	1
1	1	1	0	X	1	X	1	0	0	1	1	1
1	1	1	1	X	1	X	1	0	0	0	0	0

Applicando le mappe di Karnaugh:

$x_1 x_0$	00	01	11	10
$Q_1 Q_0$				
00	0	1	0	0
01	1	1	0	0
11	X	X	X	X
10	X	X	X	X

$$J_1 = \overline{x_1}x_0 + x_1\overline{Q_0} = \overline{x_1}(x_0 + Q_0)$$

$x_1 x_0$	00	01	11	10
$Q_1 Q_0$				
00	X	X	X	X
01	X	X	X	X
11	1	1	1	1
10	0	1	1	1

$$K_1 = Q_0 + x_1 + x_0$$

$x_1 x_0$	00	01	11	10
$Q_1 Q_0$				
00	1	0	0	0
01	X	X	X	X

11	X	X	X	X
10	1	0	0	0

$$J_0 = \overline{x_1 x_0}$$

	$x_1 x_0$	00	01	11	10
$Q_1 Q_0$					
00		X	X	X	X
01		1	0	1	1
11		1	1	1	1
10		X	X	X	X

$$K_0 = Q_1 + x_1 + \overline{x_0}$$