Compito A

Esercizio 1 (13 punti) Minimizzare l'automa descritto dalla seguente tabella degli stati

stato/input	x=0	x=1
A	B/0	A/0
B	C/0	A/0
C	B/0	D/1
D	B/0	E/0
E	B/0	D/1

Esercizio 2. (17 punti) Realizzare un distributore che accetta monete da 10 e 20 centesimi e che eroghi il prodotto al prezzo di 50 cent, dando un eventuale resto di 10. Disegnare il diagramma di una macchina di Mealy avente cinque stati, una variabile binaria di ingresso e due variabili binarie di uscita (una per il prodotto e l'altra per il resto), che corrisponde a tale distributore. Derivare poi la tabella e le funzioni booleane in forma *minima disgiuntiva* del corrispondente circuito sequenziale. Utilizzare flip-flop di tipo D.

Compito B

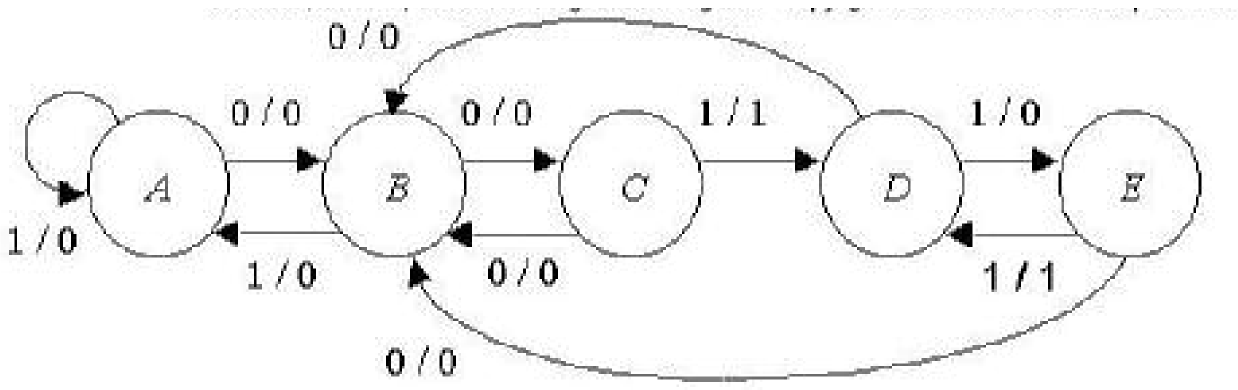
Esercizio 1 (14 punti) Minimizzare l'automa descritto dalla seguente tabella degli stati

stato/input	x=0	x=1
A	G/00	C/01
B	G/00	D/01
C	D/10	A/11
D	C/10	B/11
E	G/00	F/01
F	F/10	E/11
G	A/01	F/11

Esercizio 2 (16 punti) Progettare un circuito il cui output è 1 quando viene riconosciuta una delle seguenti stringhe: 001, 101 oppure 000. 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.

Compito C

Esercizio 1 (13 punti) Minimizzare l’automa in figura:



Esercizio 2 (17 punti) <u>Definizione</u>: In una stringa di bit di lunghezza *n* il *bit di parità* è l’(n+1)esimo bit e assume valore 1 se la stringa di n bit contiene un numero pari di 1, altrimenti assume valore 0.

Progettare un circuito sequenziale che legga in ingresso due stringhe sequenziali di bit, $X = x_1x_2x_3 \ldots x_i$ e $Y = y_1y_2y_3 \ldots y_i$ (nell’istante t_i vengono letti i bit x_i e y_i).

Il circuito deve comportarsi come segue: Se $y_i = 0$, il circuito emette 0 in uscita.. Se $y_i = 1$, l’uscita z é 1 se il bit x_i verifica la definizione di *bit di parità* di cui sopra, rispetto alla stringa $x_1 \ldots x_{i-1}$, e z=0 se x_i non la verifica.
In pratica, la stringa $y_1y_2y_3 \ldots y_i$ funziona come un “clock”, determinando quando effettuare il controllo di parità.
Ad esempio, osservate la seguente sequenza di input e output:

x (stringa di bit)	0	0	1	1	**0**	1	0	1	1	**0**	1	1	0	0	1	0	0	0	0	…
y (temporizzazione)	1	0	0	0	**1**	0	0	0	0	**1**	0	1	1	0	1	0	0	0	1	…
z (uscita)	0	0	0	0	**0**	0	0	0	0	**1**	0	0	0	0	1	0	0	0	0	…

Osservate ad esempio che in corrispondenza al secondo valore di y uguale ad uno (da sinistra, primo valore in grassetto), la stringa di valori $x_1 \ldots x_{i-1}$ letti é 00110, il bit xi=0, e dunque <u>non</u> rispetta la condizione di parità, perché la stringa che lo precede (0011) contiene un numero pari di “1”. Quindi, z=0.
La successiva occorrenza di y=1 (secondo valore in grassetto) produce z=1, perché xi=0, e la stringa che lo precede, 001101011, include 5 “1” (dispari, dunque xi ha il valore che dovrebbe avere un bit di parità).
Usare flip flop di tipo JK per la progettazione della Macchina di Mealy, che ha 2 soli stati!!!

Soluzione compito A

Esercizio 1

B	**B,C**			
C	**X**	**X**		
D	**A,E**	**A,E C,B**	**X**	
E	**X**	**X**		**X**
	A	**B**	**C**	**D**

A’=A B’=B C’=C,E D’=D

stato/input	x=0	x=1
A’	B’/0	A’/0
B’	C’/0	A’/0
C’	B’/0	D’/1
D’	B’/0	C’/0

Esercizio 2

L’input corrispondente a 10 centesimi lo rappresentiamo con x=0, mentre 20 c. con x=1. Le variabili di uscita, z e r, rappresentano rispettivamente l’erogazione di una bibita e l’erogazione del resto (es: 11 significa erogazione della bibita e del resto).

Diagramma degli stati macchina di Mealy:

	10 c. (x=0)	**20 c. (x=1)**
S0	S1/00	S2/00
S1	S2/00	S3/00
S2	S3/00	S4/00
S3	S4/00	S0/10
S4	S0/10	S0/11

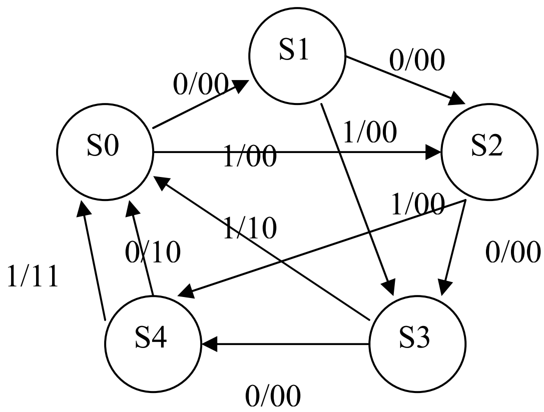


tabella con funzioni di eccitazione per flip-flop di tipo D (le colonne Qi(t+1) coincidono con le colonne Di(t))

Q1Q2Q3x	**z**	**r**	**D1**	**D2**	**D3 (= Q1Q2Q3 (t+1))**
0 0 0 0	0	0	0	0	1
0 0 0 1	0	0	0	1	0
0 0 1 0	0	0	0	1	0
0 0 1 1	0	0	0	1	1
0 1 0 0	0	0	0	1	0
0 1 0 1	0	0	1	0	0
0 1 1 0	0	0	1	0	0
0 1 1 1	1	0	0	0	0

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

mappa di **z** :

		Q3, x			
		00	01	11	10
Q1, Q2	00	0	0	0	0
	01	0	0	0	0
	11	-------------------			
	10	1	1	----------	

funzione minima: z = Q1

mappa di r :

		Q3, x			
		00	01	11	10
Q1, Q2	00	0	0	0	0
	01	0	0	0	0
	11	-------------------			
	10	0	1	----------	

funzione minima: r = Q1x

mappa di D1 :

		Q3, x			
		00	01	11	10
Q1, Q2	00	0	0	0	0

01	0	1	0	1
11	-------------------			
10	0	0	----------	

funzione minima di t1: t1 = s2not(s3)x + s2s3not(x)

mappa di D2 :

	Q3, x			
	00	01	11	10
Q1, Q2 00	0	1	1	1
01	1	0	0	0
11	-------------------			
01	0	0	----------	

funzione minima: D2 = Q2not(Q3)not(x) + not(Q1)not(Q2)x + Q2Q3

mappa di t3 :

	Q3, x			
	00	01	11	10
Q1, Q2 00	1	0	1	0
01	0	0	0	0
11	-------------------			
10	0	0	----------	

funzione minima di D3: D3 = not(Q1)not(Q2)not(Q3)not(x) + not(Q2)Q3x

Soluzioni compito B

Esercizio 1

B	C,D					
C	X	X				
D	X	X	A,B			
E	C,F	D,F	X	X		
F	X	X	A,E D,F	B,E C,F	X	
G	X	X	X	X	X	X
	A	**B**	**C**	**D**	**E**	**F**

classi di equivalenza

A’=A,B,E

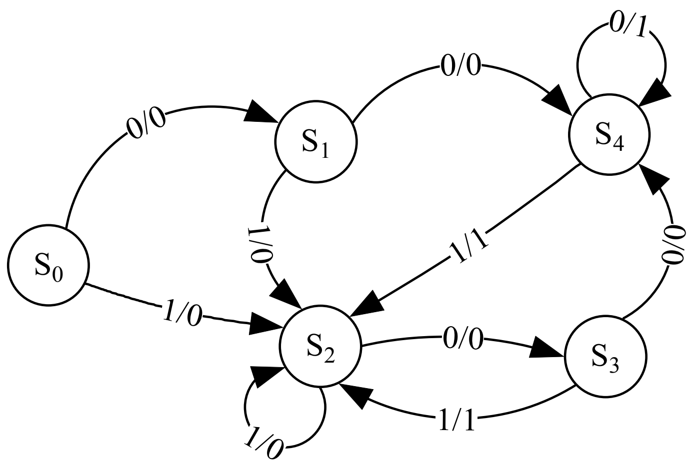
C’=C,D,F

G’=G

	x=0	x=1
A’	G’/00	C’/01
C’	C’/10	A’/11
G’	A’/01	C’/11

Esercizio 2

L’automa di Mealy è il seguente:



dove:

- S_0 indica lo stato iniziale
- S_1 lo stato in cui è stato riconosciuto “0”
- S_4 lo stato in cui è stata riconosciuta una stringa “00…0” di lunghezza almeno 2
- S_2 lo stato in cui è stato riconosciuto un “1”
- S_3 lo stato in cui è stato riconosciuto “10”

Si noti che non è necessario tenere uno stato per la stringa “101” perché è sufficiente tenere traccia dell’ultimo “1”. Infatti concatenando un bit alla stringa “101” non si ottiene nessuna sequenza valida. Lo stesso ragionamento si può applicare alla stringa “001” e così via.
Si poteva anche realizzare l’automa con tutti i possibili stati e poi minimizzarlo, ma sarebbe stato più laborioso.

Stato	**Input 0**	**Input 1**
S_0	$S_1/0$	$S_2/0$
S_1	$S_4/0$	$S_2/0$
S_2	$S_3/0$	$S_2/0$
S_3	$S_4/0$	$S_2/1$
S_4	$S_4/1$	$S_2/1$

Gli stati si codificano con 3 FF di tipo JK. La tabella degli stati futuri è la seguente:

Stato	**Q_2(t)**	**Q_1(t)**	**Q_0(t)**	**x**	**J_2**	**K_2**	**J_1**	**K_1**	**J_0**	**K_0**	**Q_2(t+1)**	**Q_1(t+1)**	**Q_0(t+1)**	**z**
S_0	0	0	0	0	0	X	0	X	1	X	0	0	1	0
	0	0	0	1	0	X	1	X	0	X	0	1	0	0
S_1	0	0	1	0	1	X	0	X	X	1	1	0	0	0
	0	0	1	1	0	X	1	X	X	1	0	1	0	0
S_2	0	1	0	0	0	X	X	0	1	X	0	1	1	0
	0	1	0	1	0	X	X	0	0	X	0	1	0	0
S_3	0	1	1	0	1	X	X	1	X	1	1	0	0	0
	0	1	1	1	0	X	X	0	X	1	0	1	0	1
S_4	1	0	0	0	X	0	0	X	0	X	1	0	0	1
	1	0	0	1	X	1	1	X	0	X	0	1	0	1
	1	0	1	0	X	X	X	X	X	X	X	X	X	X
	1	0	1	1	X	X	X	X	X	X	X	X	X	X
	1	1	0	0	X	X	X	X	X	X	X	X	X	X
	1	1	0	1	X	X	X	X	X	X	X	X	X	X
	1	1	1	0	X	X	X	X	X	X	X	X	X	X
	1	1	1	1	X	X	X	X	X	X	X	X	X	X

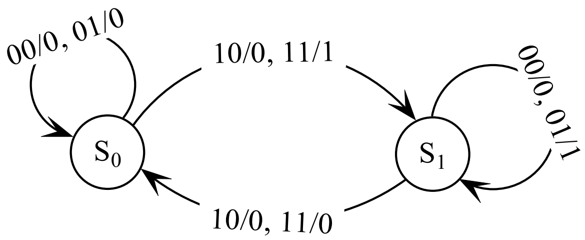
Applicando le mappe di Karnaugh si ottiene:

$J_1 = x$, $K_1 = J_2 = Q_0 \cdot \overline{x}$, $K_2 = x$, $J_0 = \overline{Q_2 x}$, $K_0 = 1$, $z = Q_2 + Q_1 \cdot Q_0 \cdot x$

Soluzioni compito C

Esercizio 1
Es.1: si tratta dello stesso automa dell’esercizio 1 del compito A

Esercizio 2
La macchina di Mealy è composta da due stati, uno per memorizzare lo stato in cui un numero di bit pari sono presenti nella stringa x (Q_0) e un altro per memorizzare lo stato in cui invece un numero di bit dispari sono presenti in x (Q_1). L’input è costituito da due bit, x_i e y_i.



La tabella degli stati futuri è la seguente:

Q(t)	00	01	10	11
Q_0	Q_0/0	Q_0/0	Q_1/0	Q_1/1
Q_1	Q_1/0	Q_1/1	Q_1/0	Q_0/0

L'automa può essere implementato con un solo flip flop JK, secondo la seguente tabella:

Q(t)	x_i	y_i	Q(t+1)	z	J	K
0	0	0	0	0	0	X
0	0	1	0	0	0	X
0	1	0	1	0	1	X
0	1	1	1	1	1	X
1	0	0	1	0	X	0
1	0	1	1	1	X	0
1	1	0	0	0	X	1
1	1	1	0	0	X	1

L'espressione booleana per z, l'uscita del circuito, può essere scritta direttamente:

$$z = y_i (Q \oplus x_i)$$

Applichiamo le mappe di Karnaugh per minimizzare l'espressione booleana di J e K:

Q \ $x_i y_i$	00	01	11	10
0	0	0	1	1
1	X	X	X	X

Da cui: $J = x_i$

Q \ $x_i y_i$	00	01	11	10
0	X	X	X	X
1	0	0	1	1

$$K = x_i$$

Il disegno del circuito è immediato.