

Esercizio 1 (15)

Disegnare l'automata che riceve in ingresso due stringhe di bit x_2 x_1 e produce in uscita $z=0$ se il numero di "zero" ricevuti su x_2 e x_1 nell'intervallo fra t_0 e t è pari (altrimenti produce 1) come illustrato nel seguente esempio:

x_2 : 01110100
 x_1 : 00011101
 z : 01001110

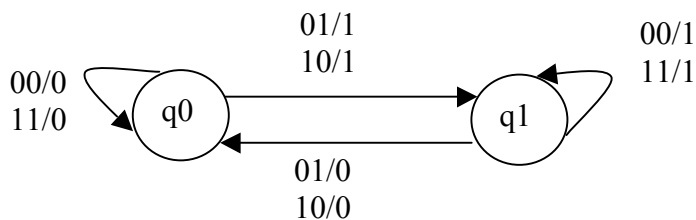
Disegnare poi il diagramma temporale rispetto alle sequenze di ingresso date nell'esempio. Non serve disegnare il circuito, nè la tabella degli stati futuri. Basta l'automata (minimizzato) e la forma d'onda.

Esercizio a1 (NOTA: se $z=1$ in caso di un numero di zero pari, l'automata è lo stesso ma gli output sono complementati)

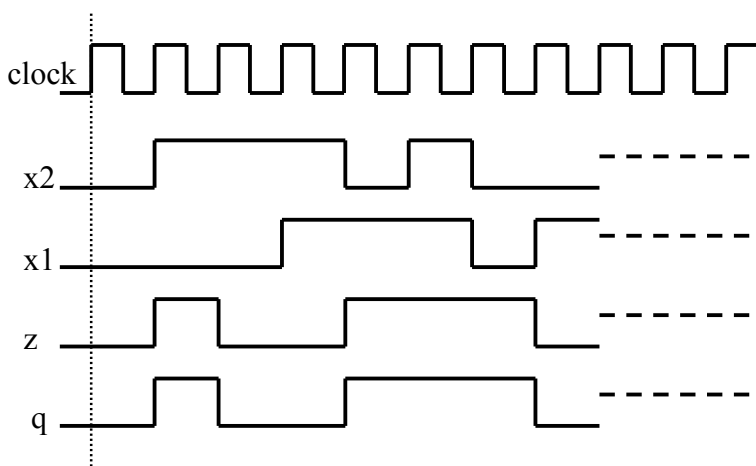
L'automata è costituito da due stati:

q_0 – stato in cui si sono ricevuti un numero pari di zero sulle linee x_2 e x_1

q_1 – stato in cui si sono ricevuti un numero dispari di zero sulle linee x_2 e x_1



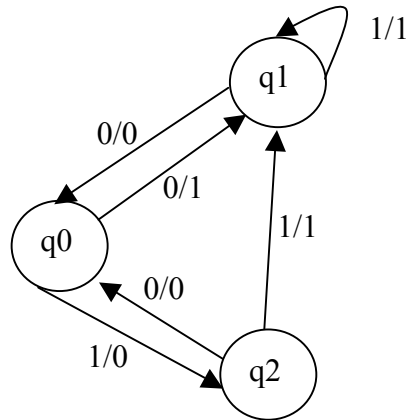
Poiché l'automata è costituito da due stati, basta un bit per la codifica; assegnando agli stati la codifica $q_0=0$ e $q_1=1$, si ottiene il seguente diagramma temporale in cui sono riportati, nell'ordine, il clock, la sequenza ricevuta su x_2 , la sequenza ricevuta su x_1 , la sequenza prodotta su z e la codifica degli stati attraversati a fronte delle sequenze ricevute:



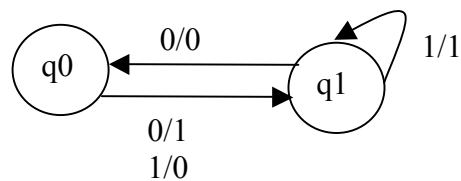
Supponendo di usare un FF di tipo D, e associando agli stati la stessa codifica dell'output z (cioè, se $Q=z$) il diagramma temporale si può disegnare solo sulla base dell'automata, senza necessità di calcolare le espressioni booleane di D e degli output.

Esercizio 2 (15)

Dato il seguente automa, minimizzare e progettare la rete sequenziale secondo il procedimento di sintesi illustrato a lezione, utilizzando FF di tipo JK e realizzando la parte combinatoria con una ROM.

**Esercizio a2**

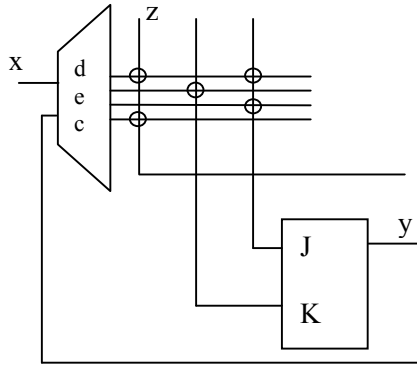
Osservando la tabella corrispondente all'automata si vede subito che l'automata può essere minimizzato essendo q1 e q2 equivalenti; l'automata minimo corrispondente è:



La tabella degli stati futuri è:

x	y	Y	z	J	K
0	0	1	1	1	x
0	1	0	0	x	1
1	0	1	0	1	x
1	1	1	1	x	0

Dalla tabella si ricava la realizzazione con ROM:

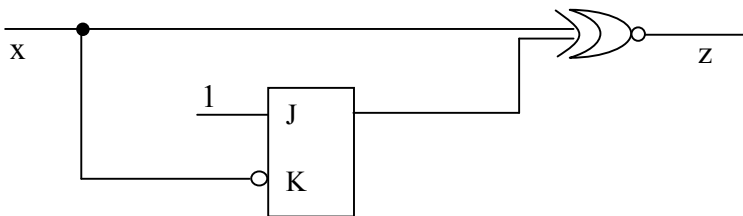


L'espressione di J si ricava dalla tabella: $J = 1$

Minimizzando con Karnaugh si ottiene $K = \underline{x}$

Infine, z non si minimizza e la sua espressione è: $z = \underline{x} q + x q$

La realizzazione circuitale è quindi:



Senza minimizzazione dell'automa, servono due bit per codificare gli stati, quindi due FF. Dopo avere steso la tabella di verità si può realizzare la rete con ROM; minimizzando con Karnaugh si ottengono le espressioni seguenti e si ricava la realizzazione con porte logiche.

$$z = \underline{x} y_2 y_1 + x y_1 + x y_2$$

$$J_2 = x y_1$$

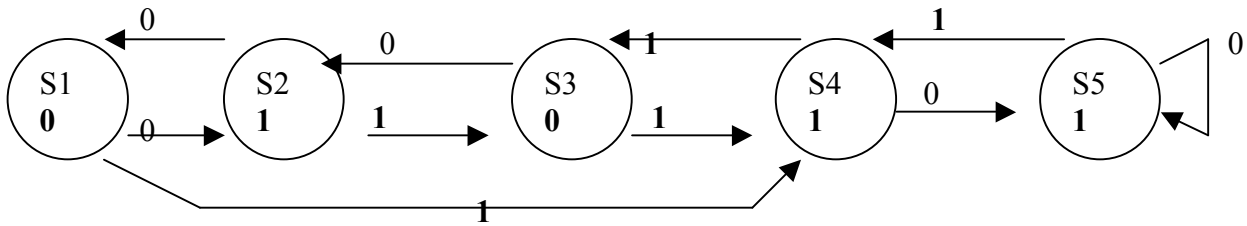
$$K_2 = 1$$

$$J_1 = \underline{x} y_2 + x y_2$$

$$K_1 = \underline{x}$$

Esercizio 1 (15)

Minimizzare il numero di stati dell'automa qui rappresentato. Disegnare l'automa minimo.



Non si richiede di progettare il circuito.

Esercizio 2 (15)

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

00111

00100

00000

00011

l'output è zero altrimenti.

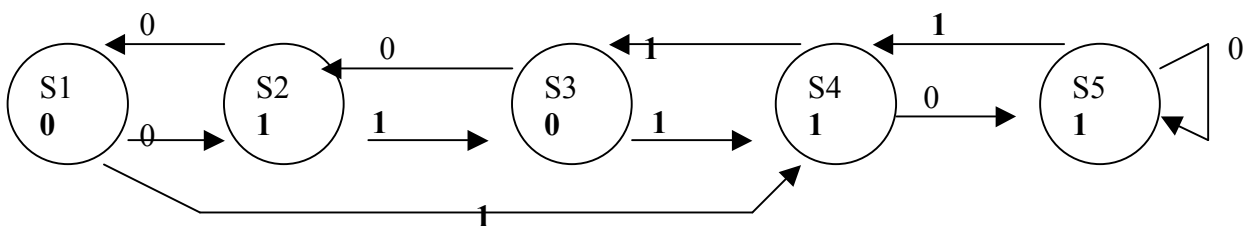
Il primo bit che viene letto è il bit **più a sinistra**.

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

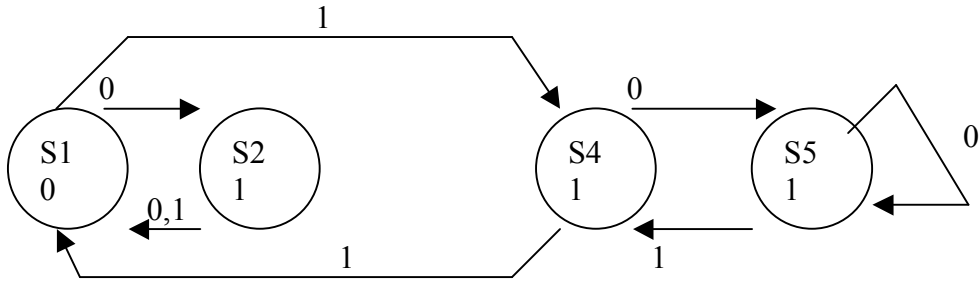
Sarà elemento di valutazione il numero degli stati complessivi utilizzati, perciò guardate bene le stringhe.

Soluzioni b**Esercizio 1**

Minimizzare il numero di stati dell'automa qui rappresentato. Disegnare l'automa minimo.

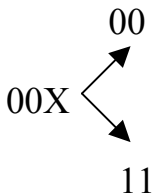


S2	X			
S3		X		
S4	X	X	X	
S5	X	X	X	X
	S1	S2	S3	S4



Esercizio 2

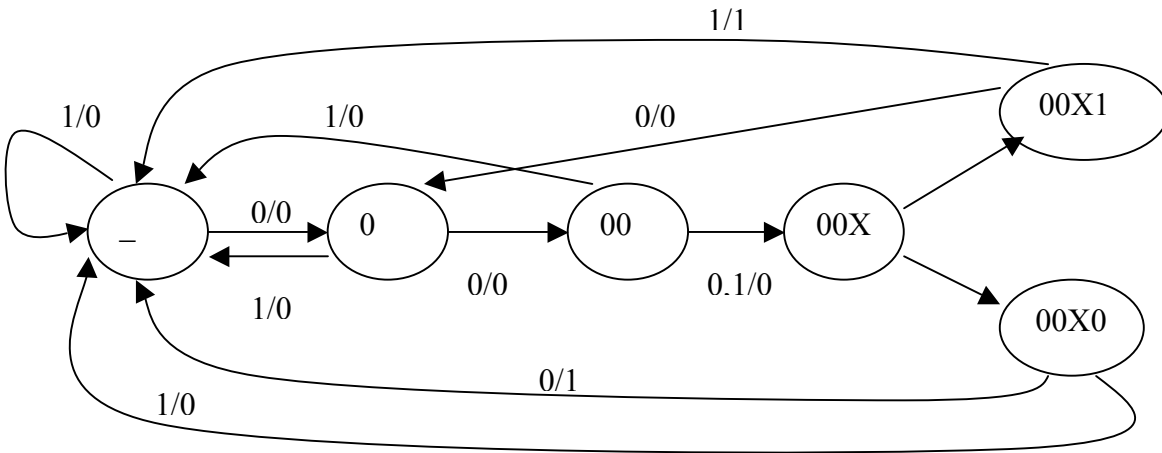
Osservate che le stringhe da riconoscere possono essere così sovrapposte (X = 0 oppure 1):



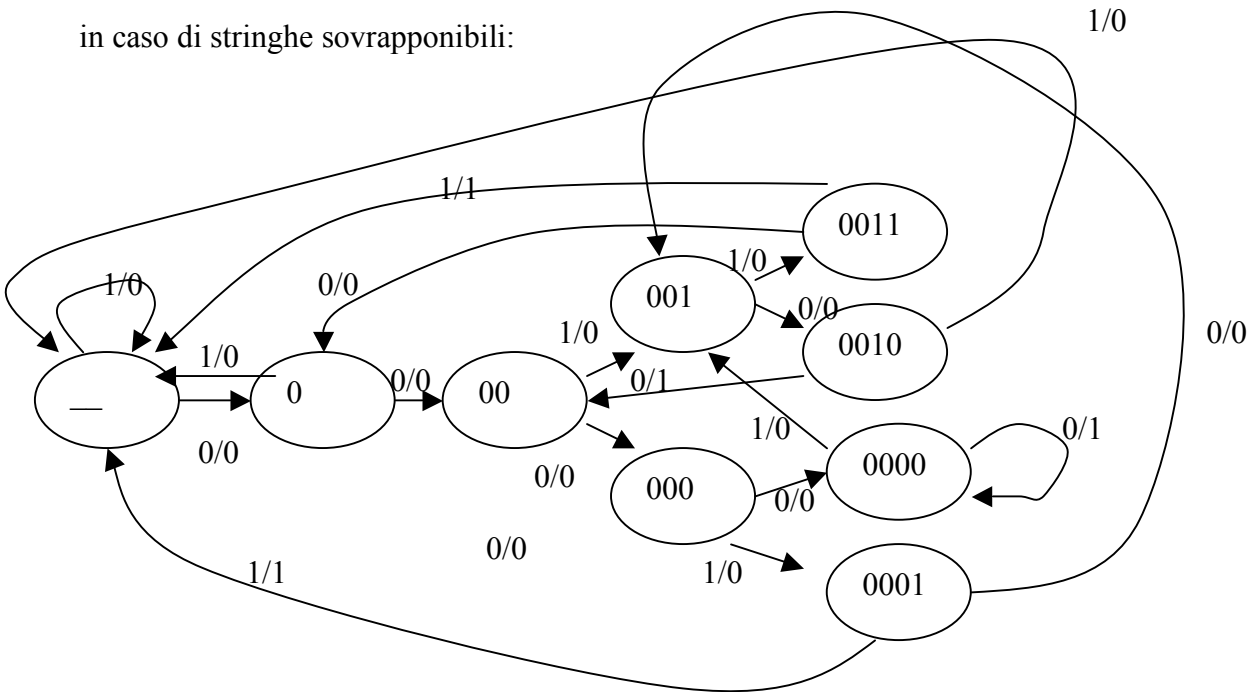
Ovvero, per come sono fatte le stringhe, non ha importanza che il terzo bit sia 0 o uno. In ogni caso, se i successivi 2 bit sono 00 o 11, si deve emettere un 1.

L'automa di mealy è in figura.

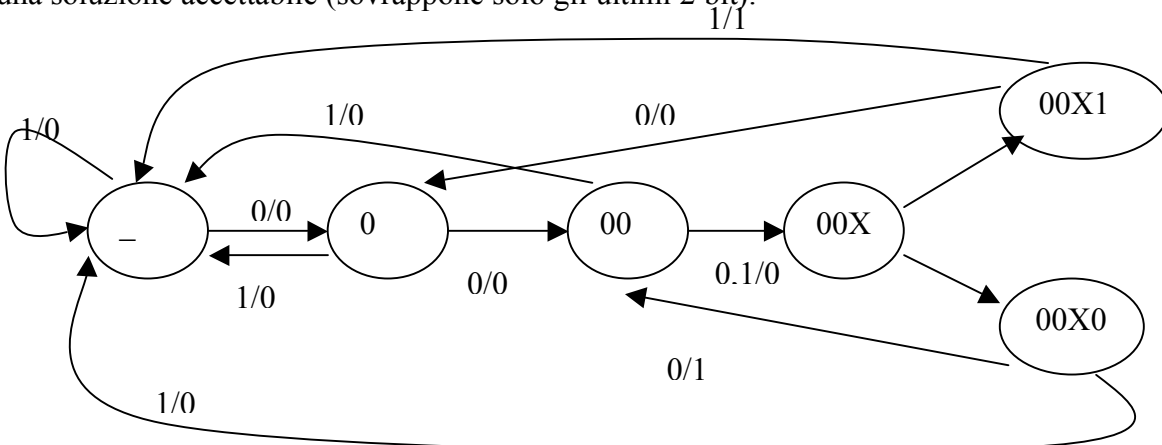
(gli stati sono “contrassegnati” sulla base della porzione di stringa riconosciuta, lo stato iniziale è contrassegnato con “_” perchè, finchè si riceve 1, nessuna porzione utile delle stringhe è stata riconosciuta.



in caso di stringhe sovrapponibili:



una soluzione accettabile (sovrappone solo gli ultimi 2 bit):



Esercizio 1 (20 punti)

Progettare una rete sequenziale con tre uscite S, C e D, ciascuna delle quali comanda l'accensione di tre lampadine L1 L2 ed L3 (ad es. se S=1 L1 è accesa).

Il circuito è scatenato da un segnale di clock.

Il circuito riceve un input binario I, tale per cui:

finchè I=1, le lampadine devono accendersi in sequenza ed una alla volta, cioè le uscite dell'automa devono ciclare come segue: 100, 010, 001, 100, 010 ecc.

finchè I=0, le lampadine devono accendersi due alla volta, secondo lo schema: 110, 011, 101, 110

Un cambiamento del valore di I mentre si sta visualizzando una sequenza porta (al clock successivo) allo stato in cui si produce il primo input dell'altra sequenza (rispettivamente, 100 e 110).

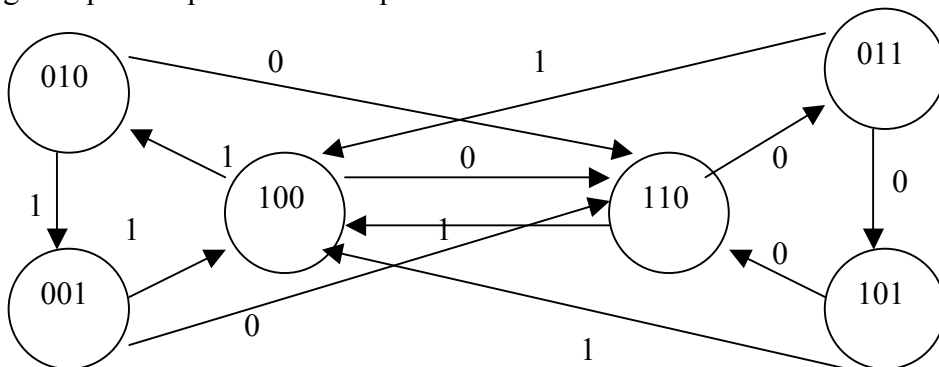
Si utilizzino FF di tipo D e si realizzi un automa di Moore.

Esercizio 2 (10 punti)

Illustrare lo schema ed il funzionamento di un sommatore aritmetico (a piacere, sequenziale o parallelo).

Soluzioni c

L'automa è visualizzato in figura. Si tratta di una macchina di Moore in cui agli stati è associato il valore degli output che pilotano le lampadine.



Dato l'automa, si può facilmente ricavare la tabella degli stati futuri e le espressioni booleane per D0, D1 e D2 e per le tre uscite S C e D.

Esercizio 2: Il sommatore parallelo è una concatenazione di full-adder (vista a lezione). Anche il sommatore sequenziale è stato svolto a lezione. Tutto è sui lucidi.

Esercizio1 (22 punti)

Si progetti un circuito che, presi in input due stringhe binarie, dà in output 1 se e solo se la somma dei valori binari associati alle due stringhe ricevuti fino a quel momento è un multiplo di 4. Si assuma che la prima cifra ricevuta sia la cifra più significativa del valore binario associato alla stringa stessa (cioè i bit si concatenano a destra). Per esempio

Input: 100100...

100010...

somma: 000110

Ouput: 011000...

Si segua lo schema di sintesi visto a lezione, inclusa la minimizzazione dell'automa e la semplificazione delle espressioni booleane ottenute dalle mappe di Karnaugh tramite porte NAND, NOR, XOR e NXOR (se possibile). Si usino flip-flop di tipo JK e non si utilizzi alcun modulo di somma predefinito (non serve!).

Suggerimento: sono i due bit meno significativi della stringa che rappresenta la somma a determinare l'output.

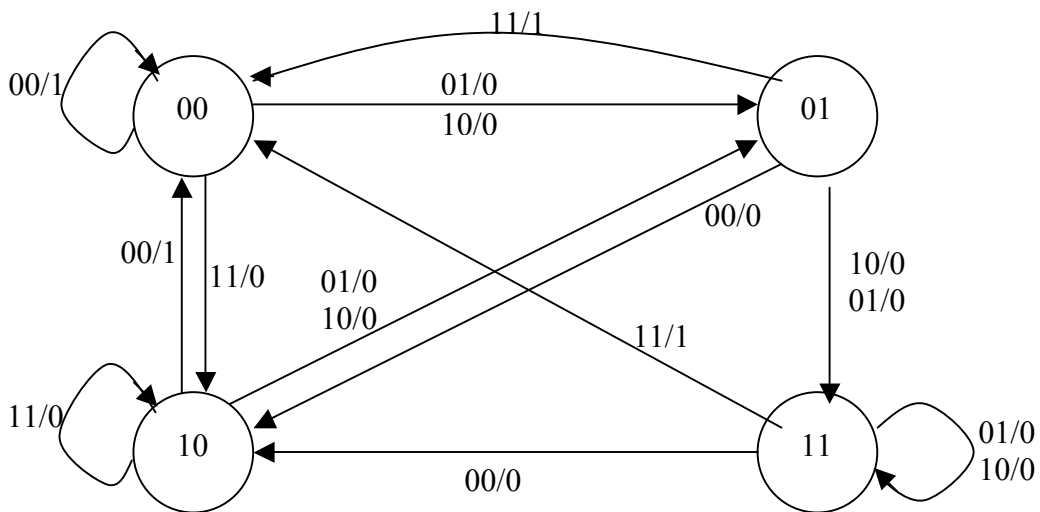
Non occorre disegnare il circuito, ma solo ottenere le espressioni booleane della parte combinatoria (uscite e JK).

Esercizio 2 (8 punti)

Nell'esercizio precedente, si implementino le funzioni booleane della parte combinatoria dl circuito con PLA e multiplexer.

Soluzioni d

Vale la pena realizzare l'automa associando agli stati gli ultimi due bit della rappresentazione binaria della somma dei due valori di input ricevuti fino a quel momento. Pertanto si avrà



In rappresentazione tabellare l'automa è

	00	01	10	11
S0	S0/1	S1/0	S1/0	S2/0
S1	S2/0	S3/0	S3/0	S0/1
S2	S0/1	S1/0	S1/0	S2/0
S3	S2/0	S3/0	S3/0	S0/1

Chiaramente l'automa è minimizzabile in

	00	01	10	11
T0	T0/1	T1/0	T1/0	T0/0
T1	T0/0	T1/0	T1/0	T0/1

La tabella degli stati futuri è

x	y	Q(t)	Q(t+1)	z	J	K
0	0	0	0	1	0	-

0	0	1	0	0	-	1
0	1	0	1	0	1	-
0	1	1	1	0	-	0
1	0	0	1	0	1	-
1	0	1	1	0	-	0
1	1	0	0	0	0	-
1	1	1	0	1	-	1

Usando le mappe di Karnaugh si ottiene

$$z = \underline{x} \underline{y} Q + x y Q$$

$$J = \underline{x} y + x \underline{y}$$

$$K = x y + \underline{x} \underline{y}$$

Le ultime due espressioni possono essere scritte come

$$J = x \oplus y$$

$$K = x \otimes y$$

Da esse si ricava facilmente il circuito risultante.

NOTA: il simbolo “-“ sta per don’t care (come il simbolo X)