

# Esercitazioni di Reti Logiche

## **Sintesi di Reti Sequenziali**

Zeynep KIZILTAN

Dipartimento di Scienze dell'Informazione

Universita' degli Studi di Bologna

Anno Accademico 2007/2008

# Sintesi dei circuiti sequenziali

- Come nel caso dei circuiti combinatori, la progettazione dei circuiti sequenziali:
  - parte dalla descrizione verbale del problema;
  - si conclude con un diagramma logico o un elenco di funzioni booleane da cui il diagramma può essere ottenuto.
- Abbiamo visto che un circuito sequenziale è definito da una tabella di stato.
- Quindi, il primo passo della procedura di progettazione ha lo scopo di ottenere **la tabella o il diagramma di stato** del circuito voluto.

# Tabella di Stato

- La tabella di stato elenca tutte le combinazioni dello stato presente e degli ingressi.
- Per ogni combinazione, sono riportati:
  - lo stato futuro (ottenuto dalle equazioni d'ingresso e la tabella caratteristica);
  - le uscite del circuito (ottenute dalle equazioni di uscita).

Present State		Input	Next State		Output
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

# Tabella di Stato

- Una rappresentazione alternativa della tabella separa gli stati futuri, a seconda del valore dell'ingresso.

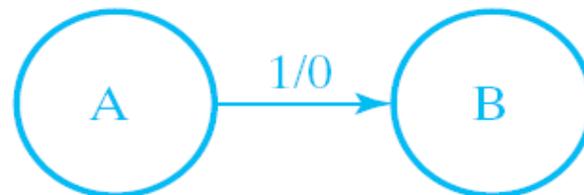
Present state		Next state				Output	
		X = 0		X = 1		X = 0	X = 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

# Esercitazione 1

- Vediamo adesso come si puo' costruire il diagramma di stato e poi la tabella di stato di un circuito che:
  - riconosca la presenza di una certa sequenza di bit fissata, anche se inclusa in una sequenza piu' lunga.
- La sequenza di bit che vogliamo riconoscere e' 1101.
- La rete deve avere un ingresso X e un'uscita Z, e deve riconoscere la sequenza di bit 1101 applicata all'ingresso X.
- Piu' precisamente, l'uscita Z deve valere 1 se e solo se:
  - in corrispondenza dei 3 precedenti fronti di salita del clock, i valori letti in X erano, rispettivamente, 1, 1, 0.
  - il valore attuale dell'ingresso X e' 1.

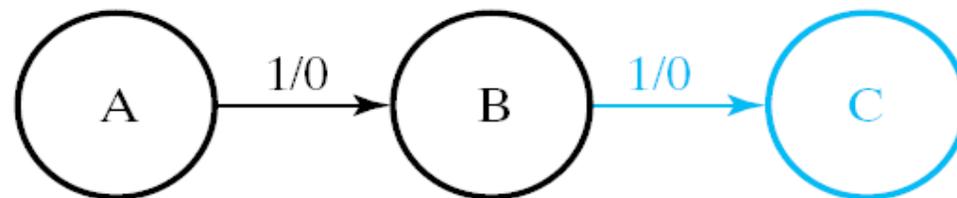
# Esercitazione 1

- Ricordate che in un circuito sequenziale, gli stati sono utilizzati per ricordare la storia degli ingressi precedenti.
- Definiamo uno stato A nel quale si trova il circuito quando non e' stato ancora rilevato 1 (il primo bit della sequenza da riconoscere).
- Se a questo punto e' applicato un 1 all'ingresso, l'evento deve essere ricordato e lo stato seguente non puo' essere ancora A.
- Pertanto, si definisce un secondo stato B per ricordare che e' stato riconosciuto il primo 1 della sequenza.
- La transizione da A a B e' etichettata con 1/0.



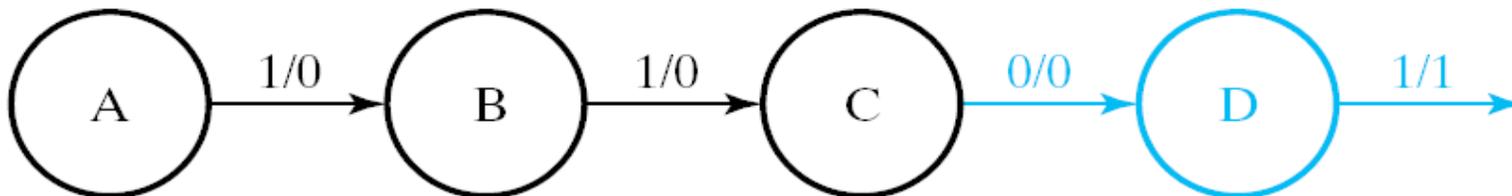
# Esercitazione 1

- Il bit successivo della sequenza e' 1.
- Partendo dallo stato B, se si presenta un 1 all'ingresso, e' necessario un altro stato C per ricordare il verificarsi di due 1 consecutivi.
- La transizione da B a C e' etichettata con 1/0.



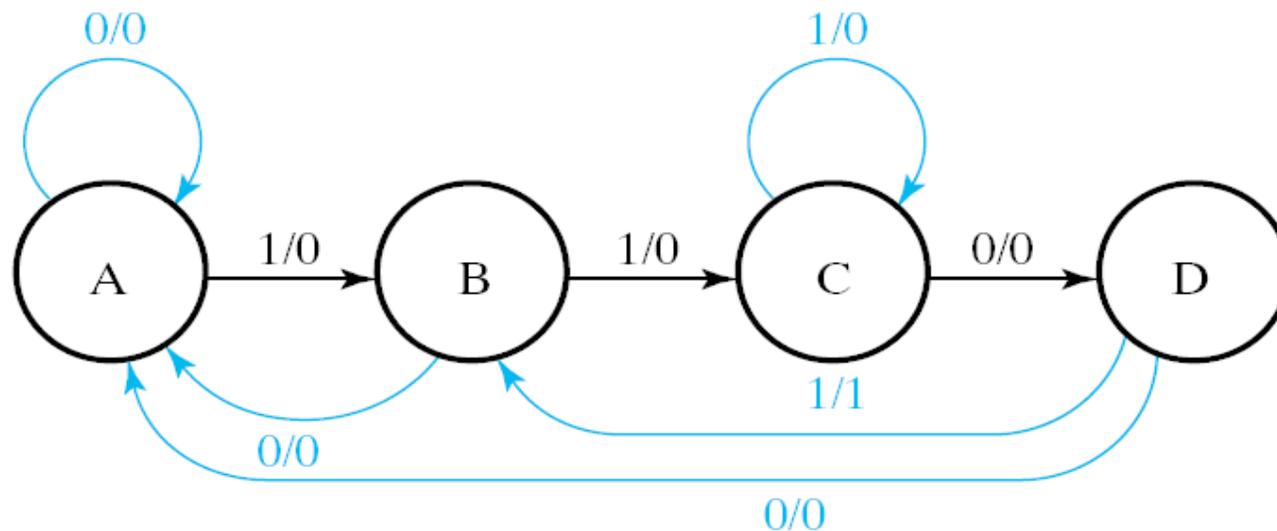
# Esercitazione 1

- Il bit successivo della sequenza e' 0.
- Partendo dallo stato C, se si presenta uno 0 all'ingresso, e' necessario uno stato ulteriore D per rappresentare il verificarsi di due 1 seguiti da uno 0.
- La transizione da C a D e' etichettata con 0/0.
- Siccome si arriva a D se i primi 3 bit di X sono 110, il verificarsi di un 1 nello stato D completa la sequenza da riconoscere.
- Quindi, la transizione dallo stato D e' etichettata con 1/1.



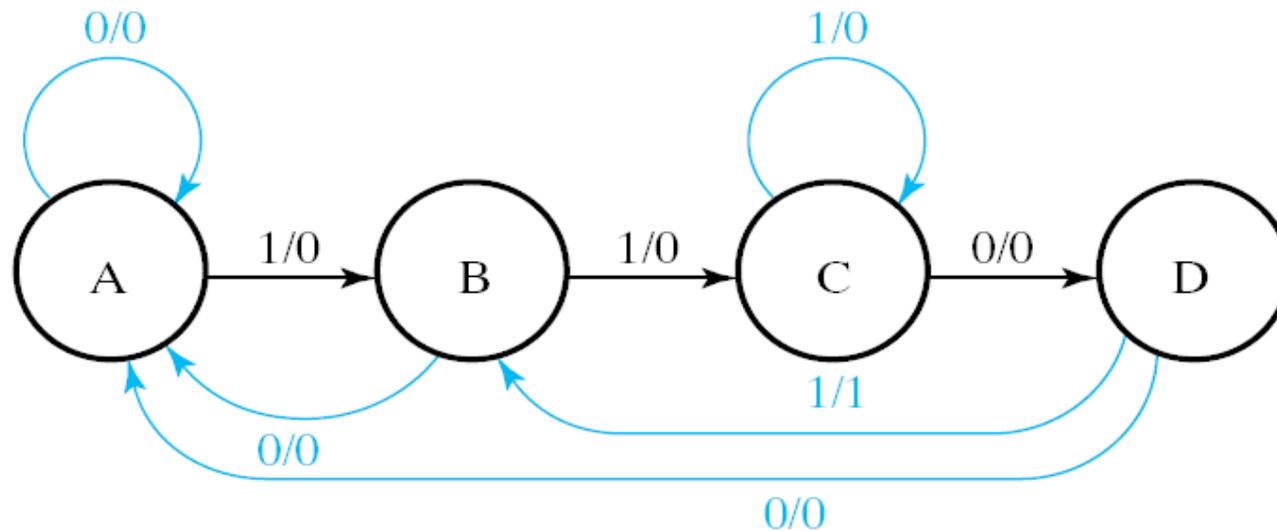
# Esercitazione 1

- I 4 stati bastano per riconoscere una sequenza di 4 bit.
- Ma dove si arriva a partire dallo stato D? All'inizio?
- Dobbiamo tenere conto che il circuito puo' riconoscere tante sequenze di 1101 in una sequenza lunga.
- L'ultimo 1 che abbiamo gia' riconosciuto puo essere il primo 1 di un'altra sequenza. E.g., 111**1**10**1**10**1**111
- Quindi, con l'ingresso 1 lo stato dopo lo stato D e' lo stato B.

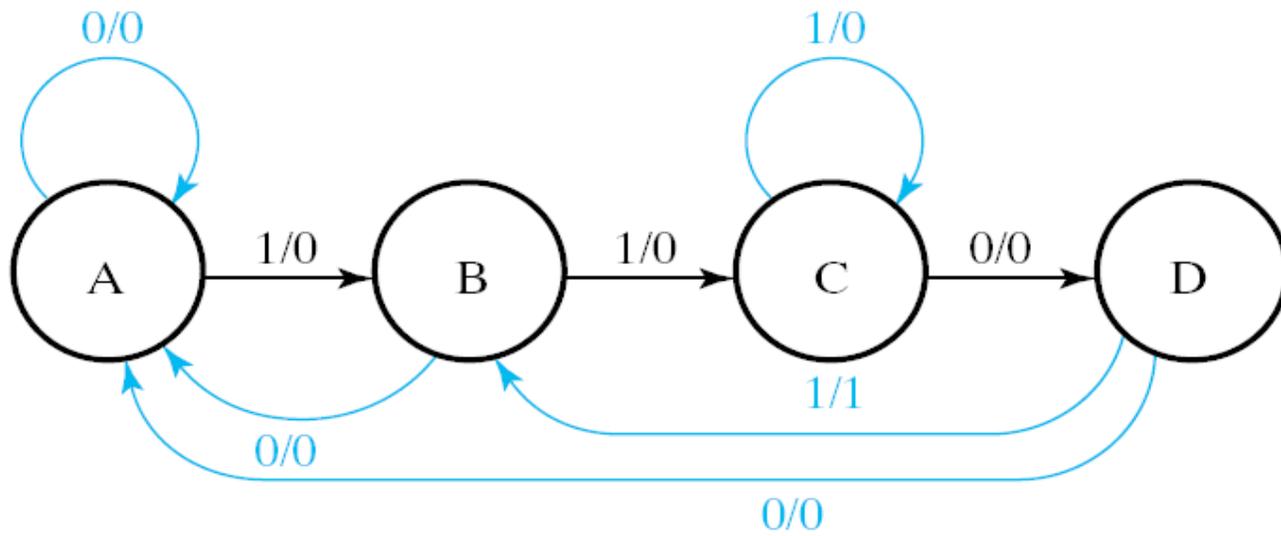


# Esercitazione 1

- Esaminiamo adesso le altre transizioni.
- Uno 0 all'ingresso dopo un 1 deve essere ricordato solo se l'1 e' il secondo 1.
- Negli altri casi, il verificarsi di uno 0 ci porta all'inizio (lo stato A).
- Nello stato C, il verificarsi di un ingresso 1 corrisponde al secondo 1 della sequenza.
  - E.g., 11111111011
  - Quindi, lo stato non cambia con l'ingresso 1 nello stato C.



# Esercitazione 1



Present State	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

# La Procedura di Progettazione (segue)

Una volta ottenuta la tabella di stato, la procedura di progettazione e' effettuata come segue:

1. Si assegnano codici binari agli stati denominati da lettere.
2. Si derivano le equazioni d'ingresso ai flip-flop a partire dalla tabella di stato.
3. Si derivano le equazioni di uscita a partire dalla tabella di stato.
4. Si semplificano le equazioni d'ingresso e le equazioni di uscita tramite K-mappe.
5. Si costruisce le rete sequenziale utilizzando flip-flop.

# Esercitazione 2

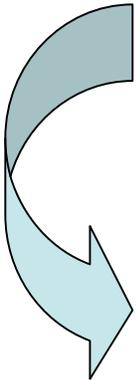
- Progettare un circuito sequenziale che operi secondo la tabella di stato della esercitazione 1. Utilizzare flip-flop D.

Present State	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

# Esercitazione 2

1. Assegnamo codici binari agli stati denominati da lettere.

Present State	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

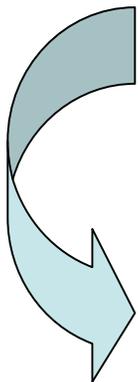


Present State AB	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
00	00	01	0	0
01	00	11	0	0
11	10	11	0	0
10	00	01	0	1

# Esercitazione 2

2. Deriviamo le equazioni d'ingresso ai flip-flop a partire dalla tabella di stato.
  - Si noti che 4 stati richiedono 2 flip-flop!
3. Deriviamo le equazioni di uscita dalla stessa tabella.

Present State AB	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
00	00	01	0	0
01	00	11	0	0
11	10	11	0	0
10	00	01	0	1



$$A(t+1) = D_A (A,B,X) = \Sigma m(3,6,7)$$

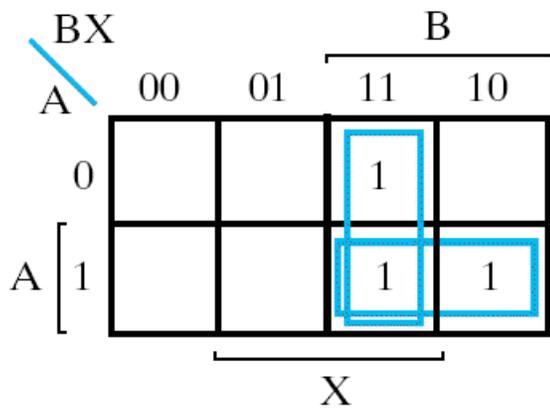
$$B(t+1) = D_B (A,B,X) = \Sigma m(1,3,5,7)$$

$$Z(A,B,X) = \Sigma m(5)$$

# Esercitazione 2

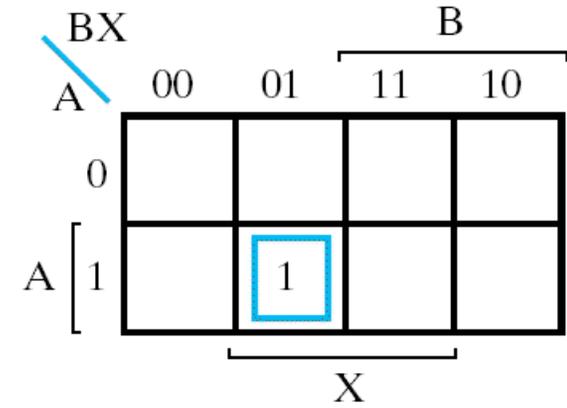
4. Semplifichiamo le equazioni d'ingresso e le equazioni di uscita tramite K-mappe.

$$D_A(A,B,X) = \sum m(3,6,7)$$



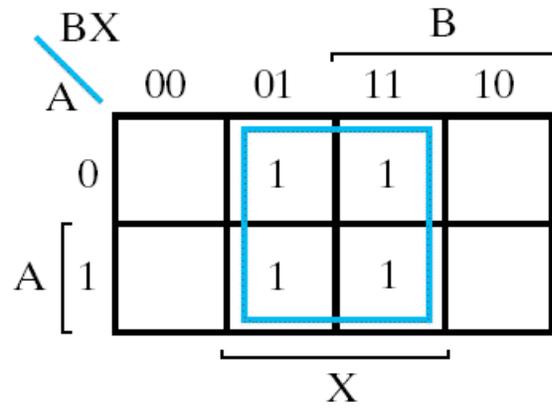
$$D_A = AB + BX$$

$$Z(A,B,X) = \sum m(5)$$



$$Z = A\bar{B}X$$

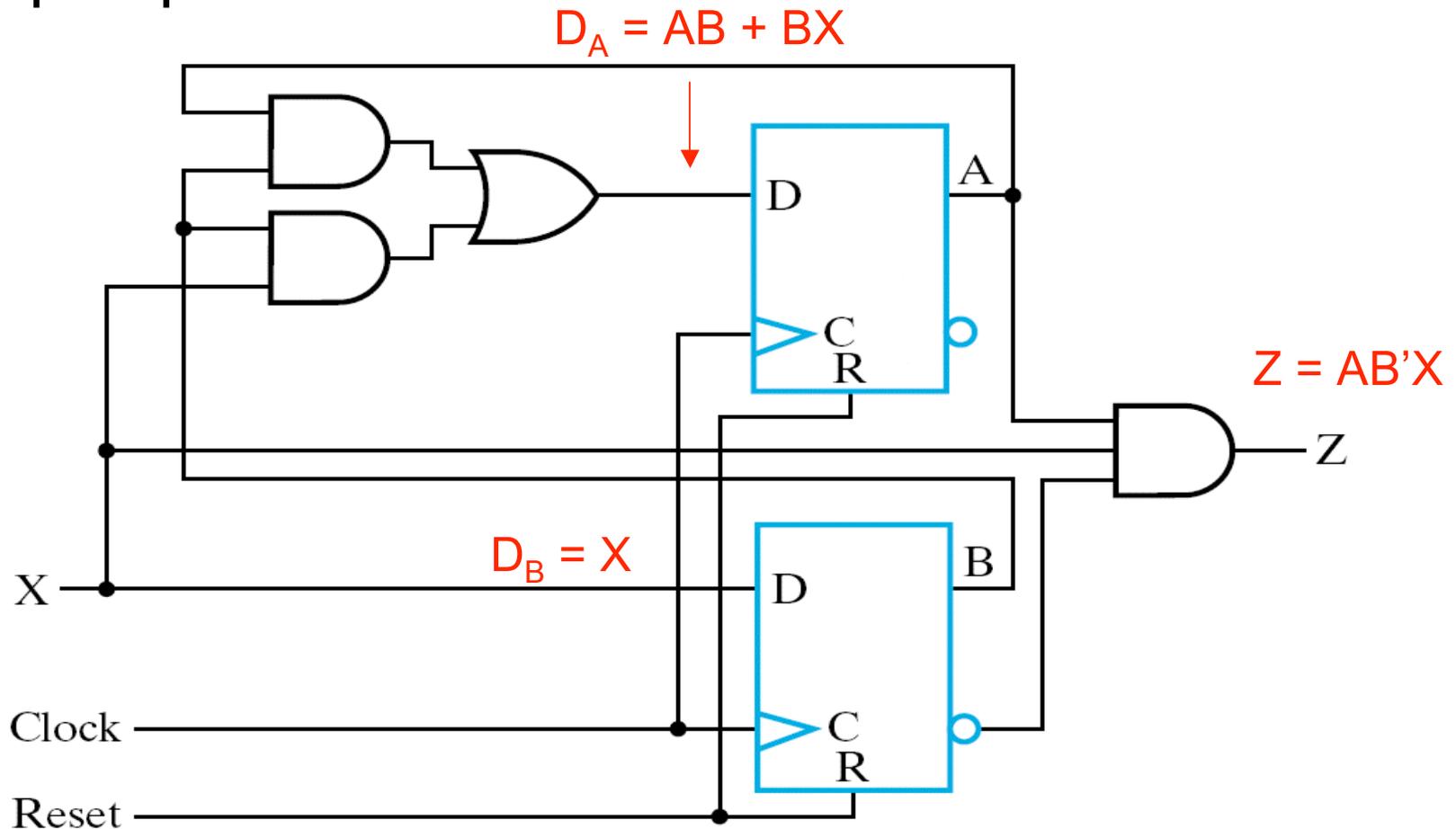
$$D_B(A,B,X) = \sum m(1,3,5,7)$$



$$D_B = X$$

# Esercitazione 2

5. Infine, disegniamo il diagramma logico utilizzando flip-flop.



# Progettazione con Stati Non Utilizzati

- Un circuito con  $n$  flip-flop può avere fino a  $2^n$  stati binari distinti.
- Una tabella di stato può pertanto avere  $m$  stati dove  $m \leq 2^n$ .
- Gli stati non utilizzati ( $2^n - m$ ) non sono riportati nella tabella, ma possono essere trattati come condizioni di non-specificazione durante la semplificazione delle equazioni.

# Esercitazione 3

- Progettare un circuito sequenziale che opera secondo la tabella di stato seguente, utilizzando flip-flop D.

Present State			Input	Next State		
A	B	C	X	A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0

# Esercitazione 3

1. Gli stati sono già indicati con codici binari.
2. Deriviamo le equazioni d'ingresso ai flip-flop:  
 $D_A = \Sigma m(5,7,8,9,11)$ ,  $D_B = \Sigma m(3,4)$ ,  $D_C = \Sigma m(2,4,6,8,10)$
3. Non ci sono uscite.

Present State			Input	Next State		
A	B	C	X	A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0

# Esercitazione 3

## 4. Semplifichiamo le equazioni d'ingresso tramite K-mappe.

- Tre stati (000), (010), (111) non sono utilizzati e non sono inclusi nella tabella.
- Considerando l'ingresso X, ci sono 6 possibili combinazioni non utilizzate per lo stato presente e gli ingressi:

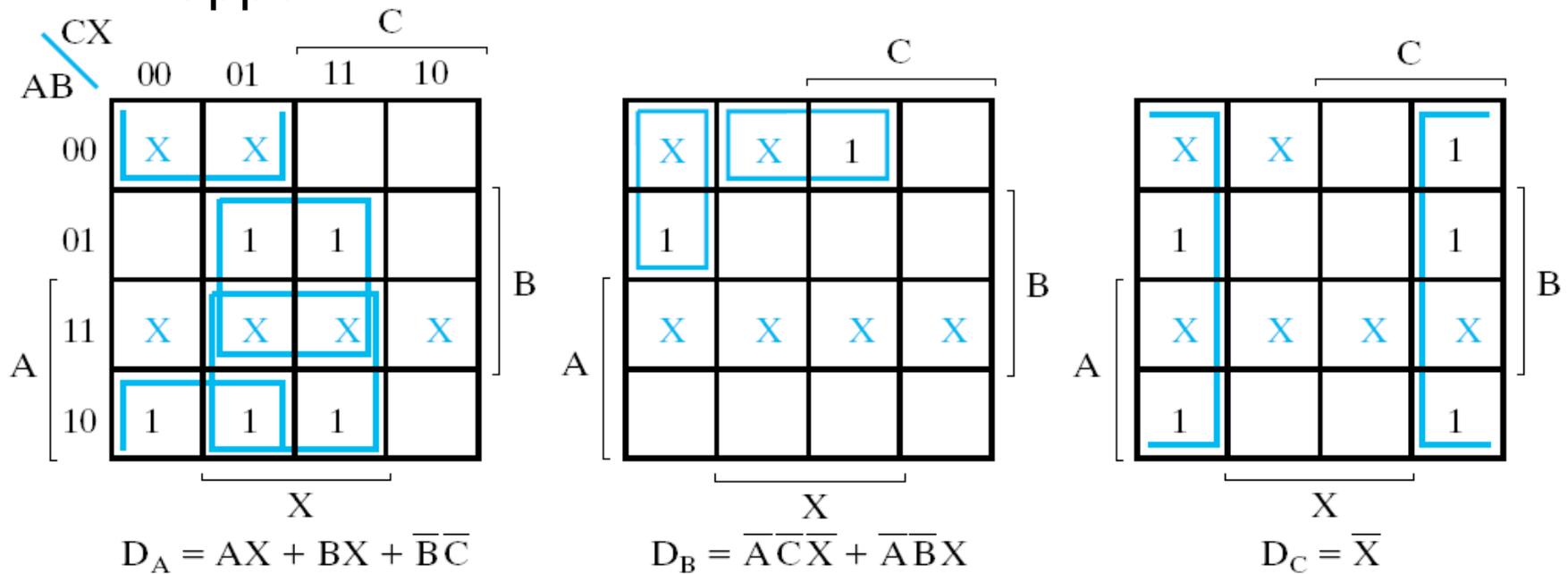
0000, 0001, 1100, 1101, 1110, 1111

Present State			Input	Next State		
A	B	C	X	A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0

possono essere trattate  
come mintermini non-  
specificati

# Esercitazione 3

4. Semplifichiamo le equazioni d'ingresso tramite K-mappe.
- Includendo le condizioni di non-specificazione nelle mappe:



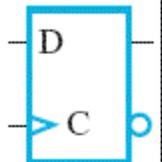
# Progettazione con flip-flop JK

- Nel caso del flip-flop D, le equazioni d'ingresso sono ottenute direttamente dalle colonne relative allo stato futuro.
- La procedura di progettazione con flip-flop JK invece richiede piu' lavoro:
  - nello specifico, le equazioni d'ingresso sono ricavate utilizzando tabelle di eccitazione.

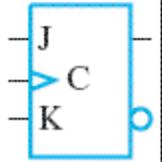
# Tabelle di eccitazione

- Le tabelle caratteristiche forniscono i valori dello stato futuro, dati i valori degli ingressi e dello stato presente.
- In una tabella di stato, sono note le transizioni dagli stati presenti agli stati futuri, ma non sono presenti le condizioni d'ingresso ai flip-flop che danno luogo a tali transizioni.
- Le tabelle di eccitazione però forniscono tali informazioni:
  - indicando quali ingressi sono necessari da applicare per generare una determinata transizione.

# Tabelle di eccitazione

Type	Symbol	Characteristic Table			Characteristic Equation	Excitation Table		
D		D	Q(t+1)	Operation	$Q(t+1) = D(t)$	Q(t+1)	D	Operation
		0	0	Reset		0	0	Reset
		1	1	Set		1	1	Set

Lo stato non cambia e rimane 0. O J=0 e K=0 oppure J=0 e K=1.

JK		J	K	Q(t+1)	Operation	$Q(t+1) = J(t)\bar{Q}(t) + \bar{K}(t)Q(t)$	Q(t)	Q(t+1)	J	K	Operation
		0	0	$Q(t)$	No change		0	0	0	X	No change
		0	1	0	Reset		0	1	1	X	Set
		1	0	1	Set		1	0	X	1	Reset
		1	1	$\bar{Q}(t)$	Complement		1	1	X	0	No Change

Lo stato cambia da 0 a 1. O J=1 e K=0 oppure J=1 e K=1.

# Progettazione con Flip-Flop JK (segue)

- Quindi, la procedura di progettazione dei circuiti sequenziali con flip-flop JK e' la stessa per i circuiti con flip-flop D, tranne il secondo passaggio:
  - i valori degli ingressi ai flip-flop sono determinati secondo la tabella di eccitazione;
  - le equazioni d'ingresso sono ricavate dalle colonne degli ingressi ai flip-flop.

# Esercitazione 4

- Progettare un circuito sequenziale che opera secondo la tabella di stato seguente, utilizzando flip-flop JK.

<u>Present state</u>		<u>Input</u>	<u>Future state</u>	
A	B	X	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

# Esercitazione 4

1. Gli stati sono già indicati con codici binari.
2. a) Prima, deriviamo gli ingressi ai flip-flop.
2. b) Poi, deriviamo le equazioni d'ingresso ai flip-flop:

$$J_A = \Sigma m(2,4,5,6,7) \quad K_A = \Sigma m(0,1,2,3,7) \quad J_B = \Sigma m(1,2,3,5,6,7) \quad K_B = \Sigma m(0,1,2,4,5,7)$$

<u>Present state</u>		<u>Input</u>	<u>Future state</u>		<u>Flip-flop inputs</u>			
A	B	X	A	B	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

# Esercitazione 4

3. Non ci sono uscite.

<u>Present state</u>		<u>Input</u>	<u>Future state</u>		<u>Flip-flop inputs</u>			
A	B	X	A	B	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

# Esercitazione 4

4. Semplifichiamo le equazioni d'ingresso tramite K-mappe.

	<b>BX</b>			
	00	01	11	10
<b>A</b>				
0				1
1	X	X	X	X

$$J_A = B X'$$

	<b>BX</b>			
	00	01	11	10
<b>A</b>				
0	X	X	X	X
1			1	

$$K_A = B X$$

	<b>BX</b>			
	00	01	11	10
<b>A</b>				
0		1	X	X
1		1	X	X

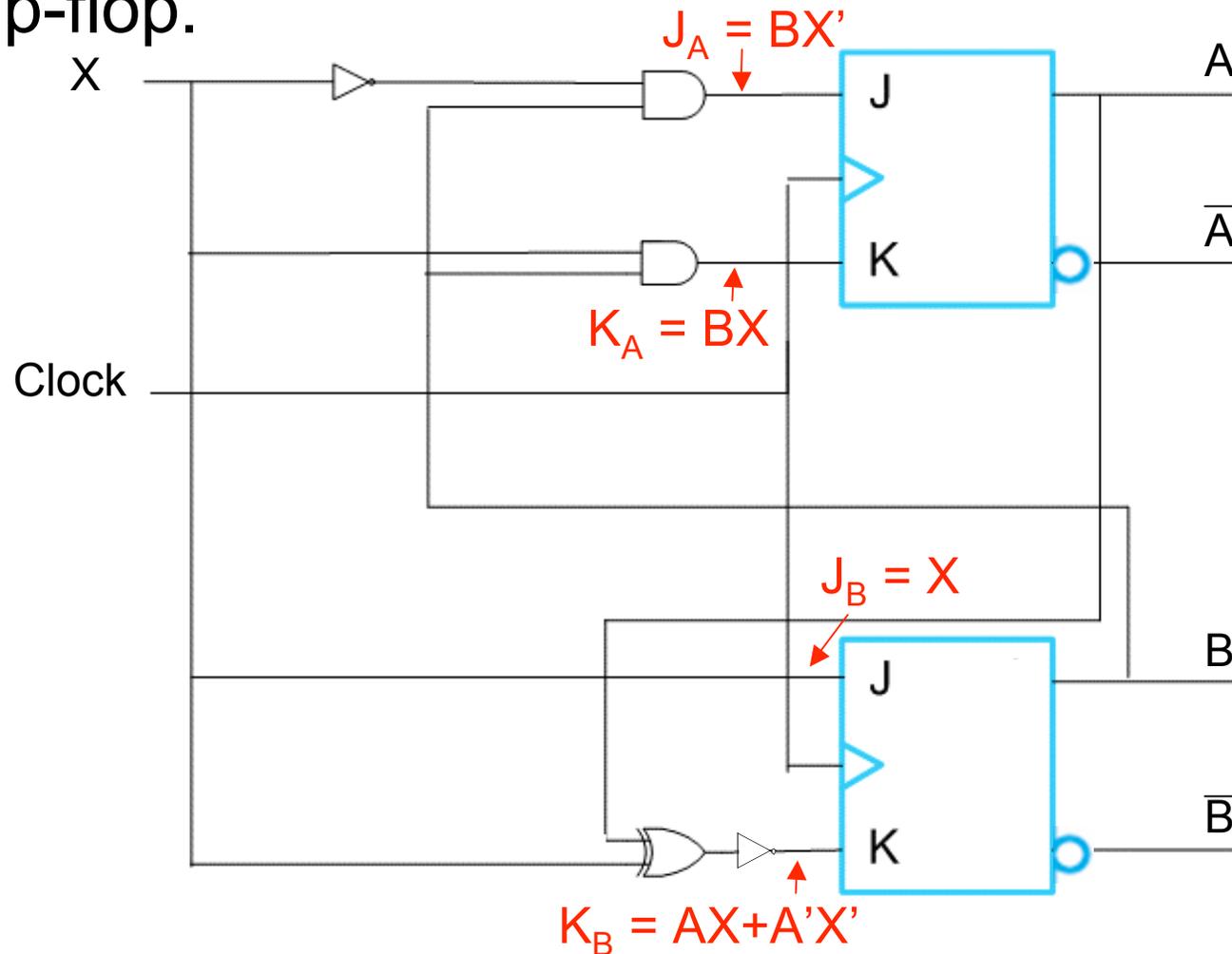
$$J_B = X$$

	<b>BX</b>			
	00	01	11	10
<b>A</b>				
0	X	X		1
1	X	X	1	

$$K_B = A X + A' X'$$

# Esercitazione 4

5. Infine, disegniamo il diagramma logico utilizzando flip-flop.



# Esercitazione 5

- Progettare un circuito sequenziale che funga da controllare per un sistema di allarme.
  - Un segnale binario proviene da un sensore, che rileva la presenza di estranei.
  - L'unica uscita e' collegata ad un dispositivo sonoro.
  - Il dispositivo deve essere attivato in un generico istante  $t$  se e solo se in almeno tre dei quattro fronti di salita del clock precedenti a  $t$  e' stata rilevata la presenza di un estraneo.

# Esercitazione 5

- Costruiamo solo il diagramma di stato.
- Una volta costruito il diagramma, possiamo facilmente ottenere la relativa tabella di stato.
- L'ingresso e' una sequenza di bit 0 e 1, in cui 1 rappresenta la rilevazione della presenza di estranei.
- Supponiamo che l'uscita venga valorizzata a 1 quando il dispositivo sonoro e' attivato.
- L'uscita vale 1 quando almeno 3 degli ultimi 4 bit della sequenza d'ingresso (escluso il bit corrente) valgono 1.

# Esercitazione 5

- Ricordate che in un circuito sequenziale utilizziamo gli stati per ricordare la storia degli ingressi precedenti.
- Dal momento che l'ordine degli 1 non conta, ma invece conta il numero di 1, dobbiamo tenere traccia di ogni possibile storia fino agli ultimi 4 bit della sequenza.
- Cominciamo con lo stato in cui il circuito si trova all'inizio.
- Poi, creiamo uno stato per ciascuna sequenza d'ingresso 0, 1, 00, 01, 10, 11, ...