

به نام خدا

# طراحی سیستم های دیجیتال ۱

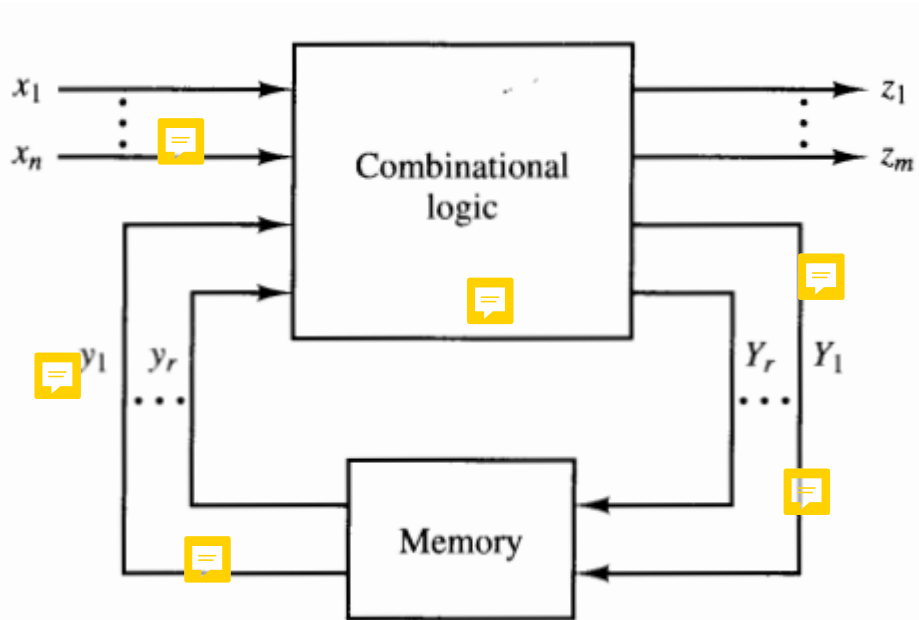
## فصل پنجم

### مدارهای منطقی ترتیبی

## Sequential Logic Circuits

## ✓ مدارهای منطقی ترتیبی

❖ خروجی در هر لحظه به ورودی در همان لحظه و به مقادیر قبلی خروجی وابسته است.



❖ از یک مدار ترکیبی و المان های حافظه تشکیل شده است.

❖ المان های حافظه قابلیت ذخیره سازی اطلاعات باینری را دارند.

❖ چند تعریف مهم:

✓ حالت (State)

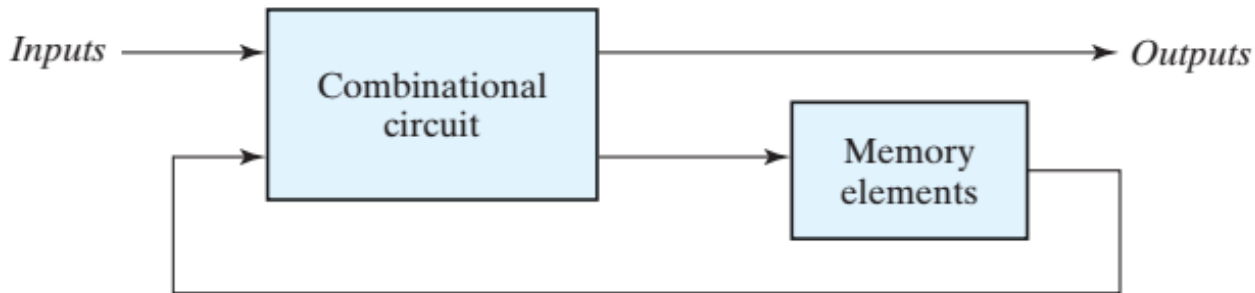
✓ حالت فعلی (Present State)

✓ حالت بعدی (Next State)

❖ مثال هایی از مدارهای منطقی ترتیبی: آسانسور، کنترل تلویزیون و ...

## ✓ مدارهای منطقی ترتیبی

❖ یکی از اجزای مدارهای ترتیبی، حافظه ها (Memory) هستند.



❖ توسط سیگنالی به نام کلاک (Clock-clk) کنترل می شوند. 🗨️

❖ قبل از زدن کلاک، مدار ترکیبی حالت بعدی مموری را آماده می کند.

❖ با زدن کلاک، اطلاعات در مموری ذخیره می شود.

❖ سرعت مدار ترکیبی خیلی مهم است.

❖ دو نوع حافظه داریم:

✓ Latch: حساس به سطح بوده و به محض تغییر ورودی، خروجی تغییر می کند.

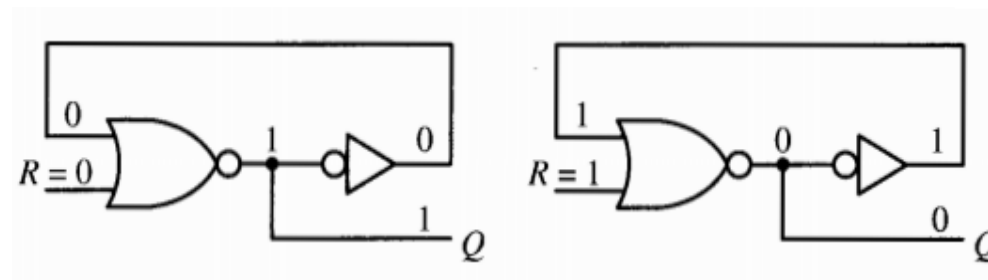
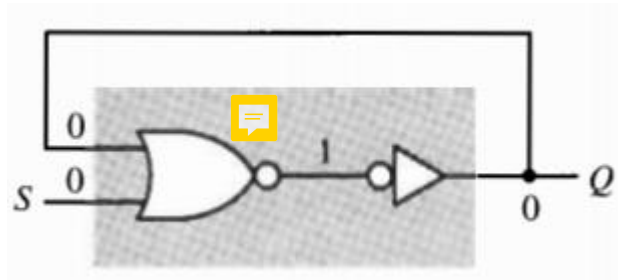
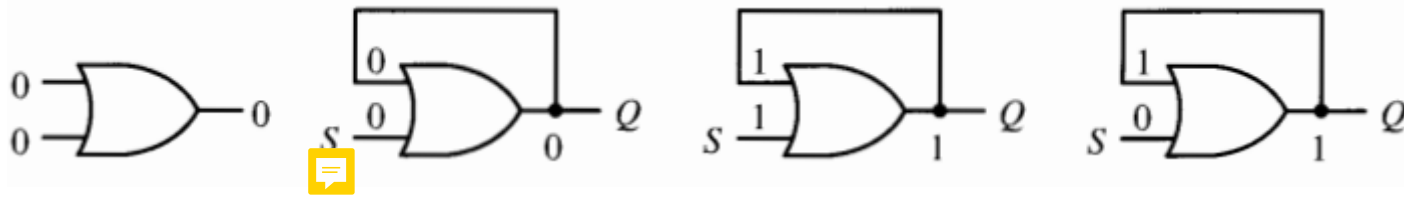
✓ Flip-Flop: حساس به لبه بوده و تا آمدن کلاک بعدی، خروجی تغییر نمی کند.



## ✓ مموری ها

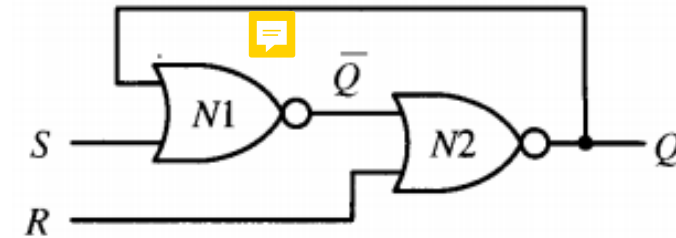
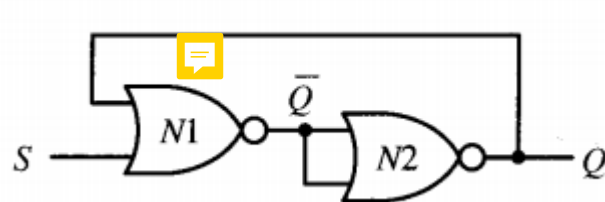
❖ خروجی Q در لاجیک "1"، set می شود.

❖ به آن **set latch** گویند.



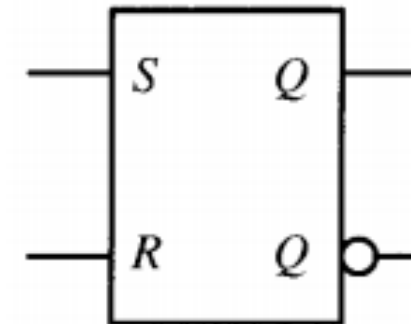
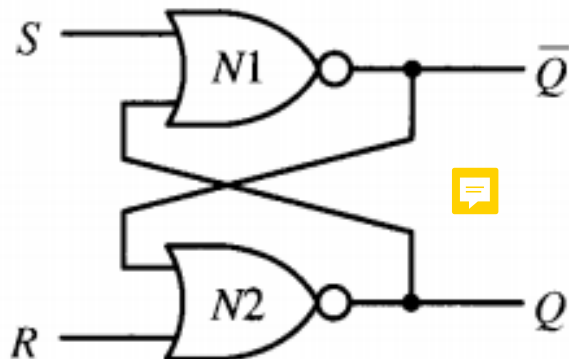
❖ خروجی همواره صفر می ماند.

❖ به آن **reset latch** گویند.

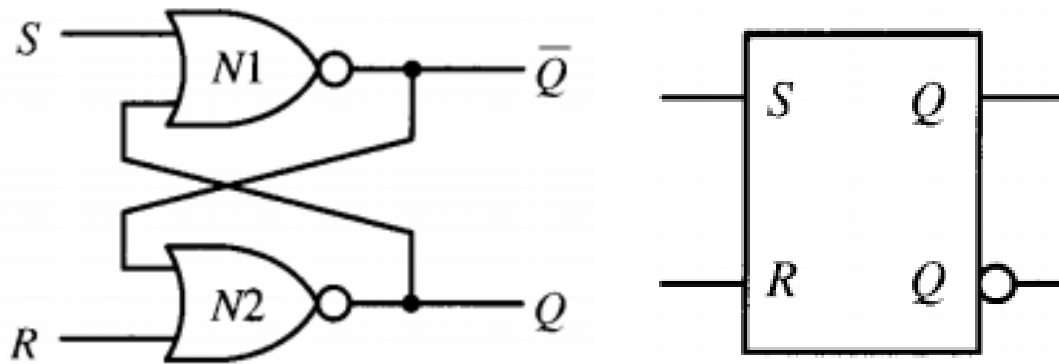


❖ ترکیب دو حالت را **set-reset latch** گویند.  
**(SR Latch)**

**Cross-Coupled  
Form**



## SR Latch ✓

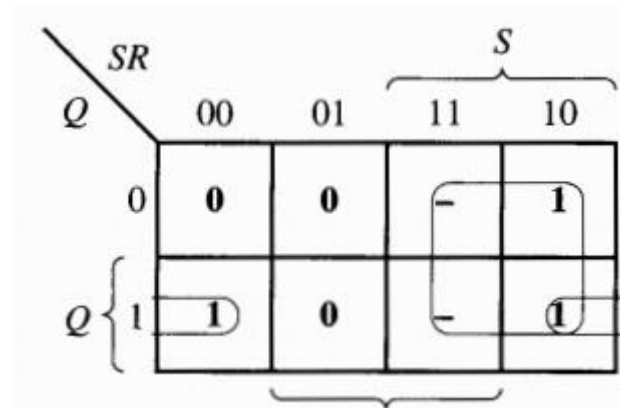


❖ دو ورودی “1” بی معنی است. مدار یا set است یا reset.

Excitation inputs		Present state	Next state	
S	R	Q	Q*	
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	×	Not allowed
1	1	1	×	

❖ جدول تحریک (Excitation Table):

❖ معادله مشخصه (Characteristic Equation):



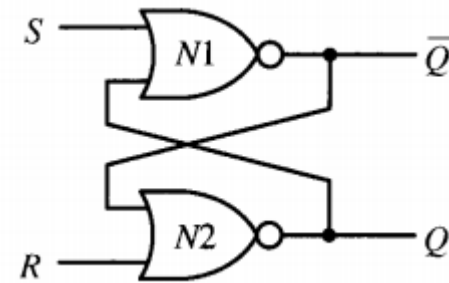
$$Q^* = S + \bar{R}Q$$

شرط:  $RS = 0$

## SR Latch ✓

❖ دیاگرام زمانی (Timing Diagram):

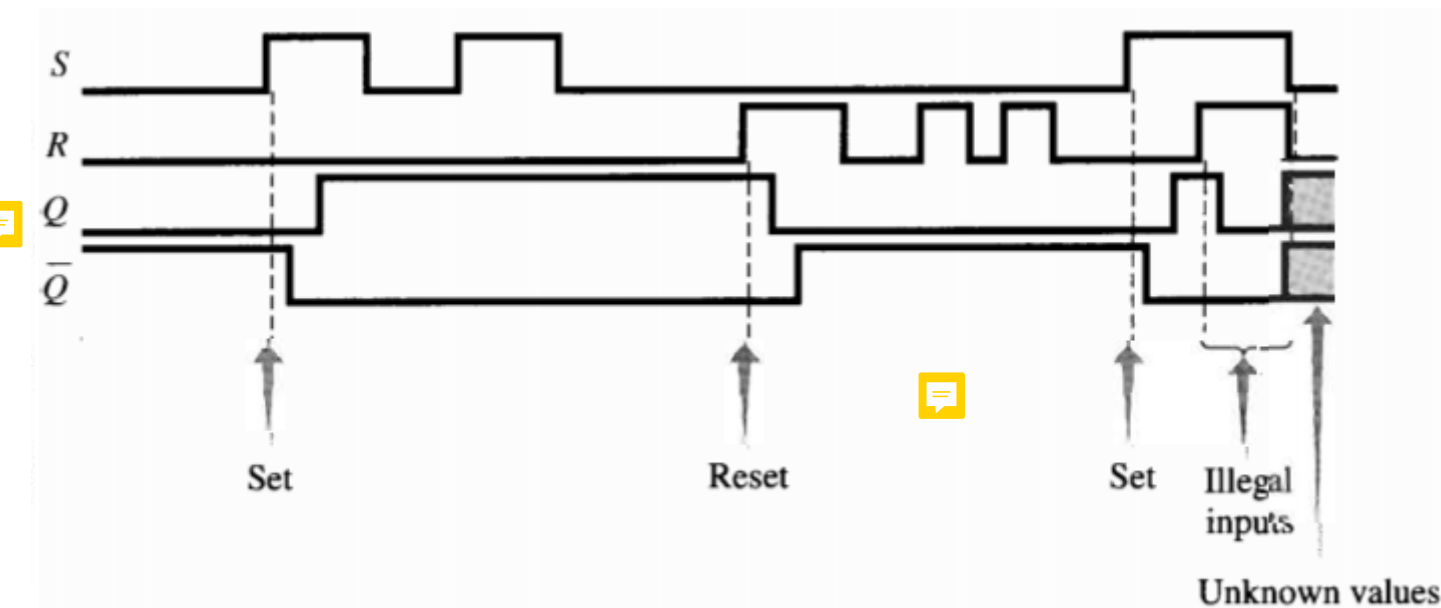
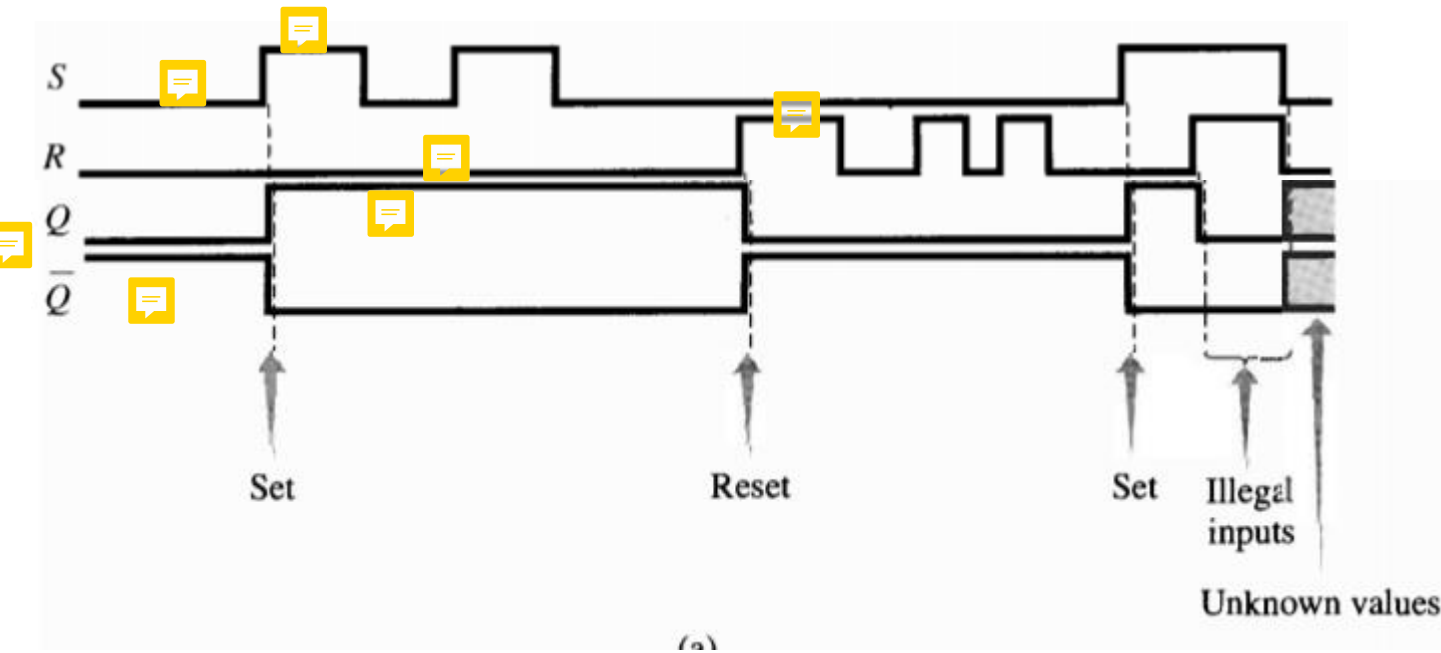
✓ حالت ایده آل:



✓ حالت غیرایده آل (همراه با تاخیر):

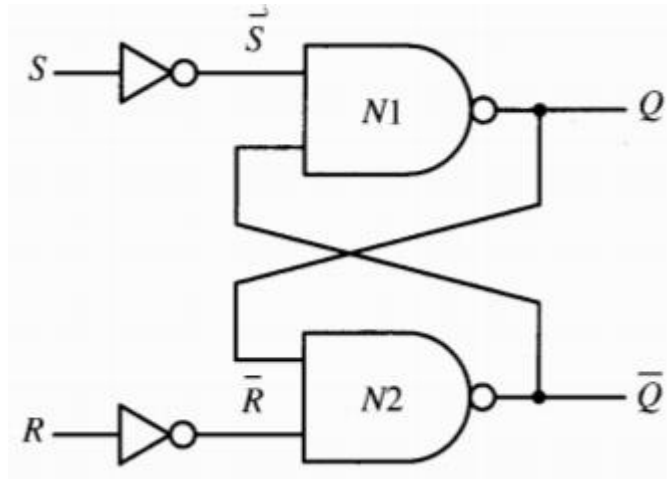
- در حالت set،  $\bar{Q}$  زودتر از  $Q$  تغییر می کند.
- $t_{pHL}$  از  $S$  به  $\bar{Q}$  دارای یک مرحله تاخیر است.
- $t_{pLH}$  از  $S$  به  $Q$  دارای دو مرحله تاخیر است.

سوال



# SR Latch ✓

## ❖ ساختار NAND (NAND Structure):



$$\text{Gate } N1 (S = 0) : \overline{\bar{S}} \cdot \overline{\bar{Q}} = \overline{1} \cdot \overline{\bar{Q}} = \overline{\bar{Q}} = Q$$

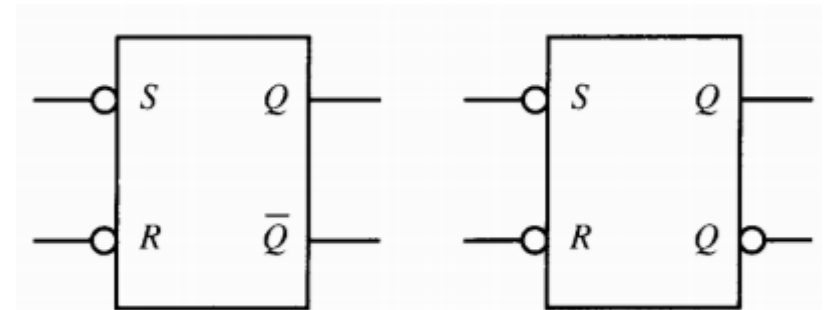
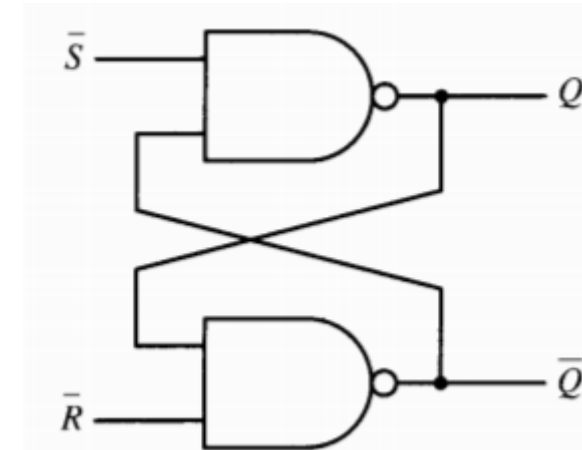
$$\text{Gate } N2 (R = 0) : \overline{\bar{R}} \cdot \overline{\bar{Q}} = \overline{1} \cdot \overline{\bar{Q}} = \overline{\bar{Q}} = Q$$

$$\text{Gate } N2 (R = 1) : \overline{\bar{R}} \cdot \overline{\bar{Q}} = \overline{0} \cdot \overline{\bar{Q}} = \overline{0} = 1$$

$$\text{Gate } N1 (S = 0, R = 1) : Q = \overline{\bar{S}} \cdot \overline{\bar{Q}} = \overline{0} \cdot \overline{1} = \overline{1 \cdot 1} = \overline{1} = 0$$

$$\text{Gate } N1 (S = 1) : \overline{\bar{S}} \cdot \overline{\bar{Q}} = \overline{0} \cdot \overline{\bar{Q}} = \overline{0} = 1$$

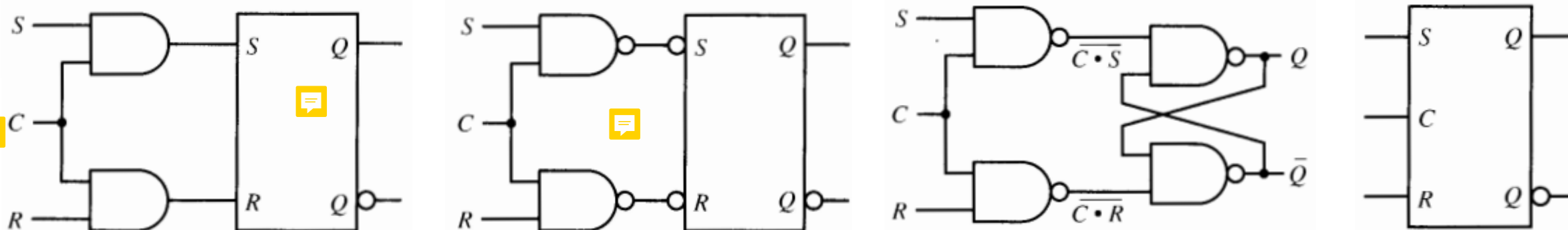
$$\text{Gate } N2 (S = 1, R = 0) : \overline{\bar{Q}} = \overline{\bar{R}} \cdot \overline{\bar{Q}} = \overline{0} \cdot \overline{1} = \overline{1 \cdot 1} = \overline{1} = 0$$



# SR Latch ✓

## Gated SR Latch ❖

✓ می خواهیم روی مدار SR Latch کنترل داشته باشیم تا با یک خط فرمان، به ورودی ها پاسخ دهد.



Enable inputs $C$	Excitation inputs		Present state $Q$	Next state $Q^*$	
	$S$	$R$			
0	x	x	0	0	Hold
0	x	x	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	x	Not allowed
1	1	1	1	x	

✓ جدول تحریک (Excitation Table):

✓ معادله مشخصه (Characteristic Eq.):

$$Q^* = SC + \bar{R}Q + \bar{C}Q$$

$$\text{if } C = 0 \rightarrow Q^* = Q \quad (\text{Hold})$$

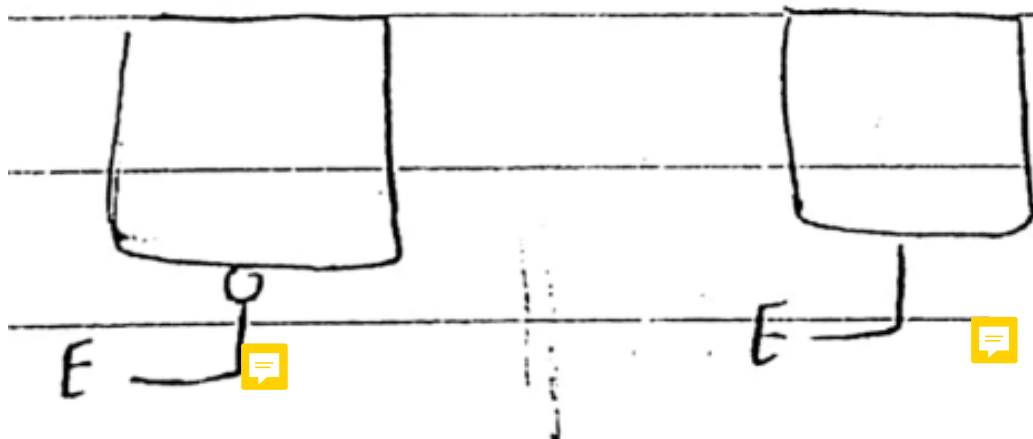
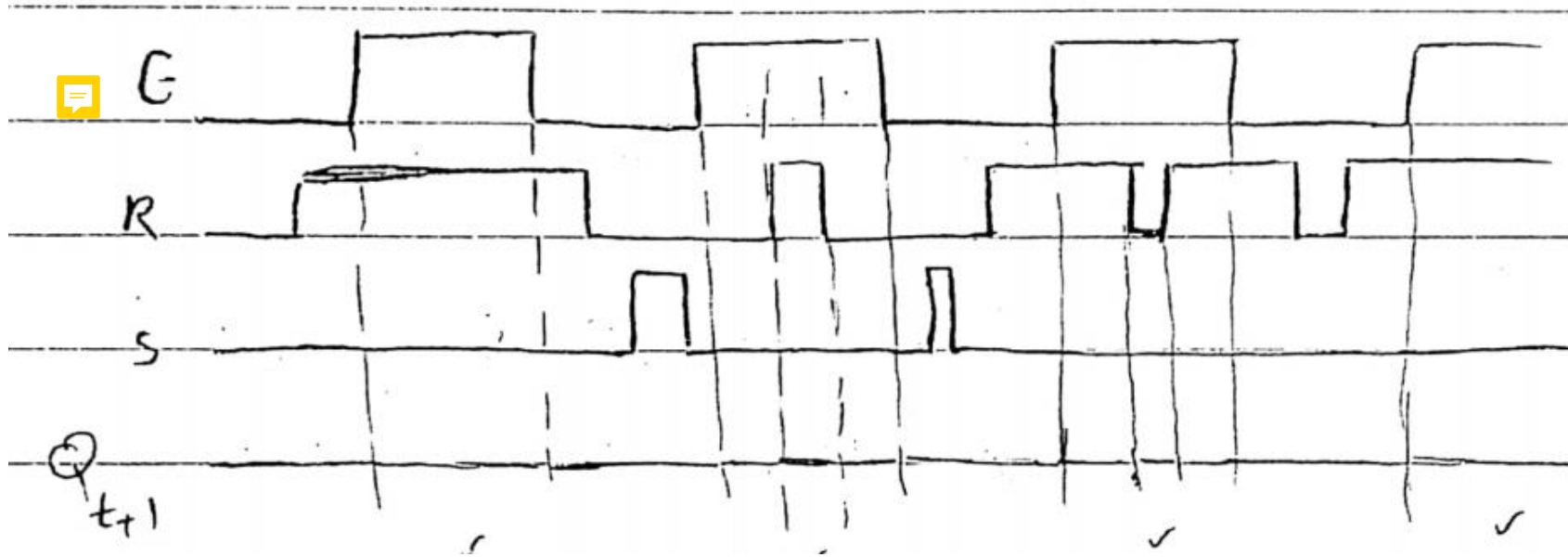
$$\text{if } C = 1 \rightarrow Q^* = S + \bar{R}Q \quad (\text{simple SR Latch})$$



## SR Latch ✓

### :Gated SR Latch ❖

✓ دیاگرام زمانی:

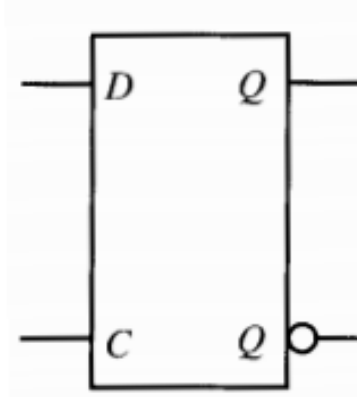
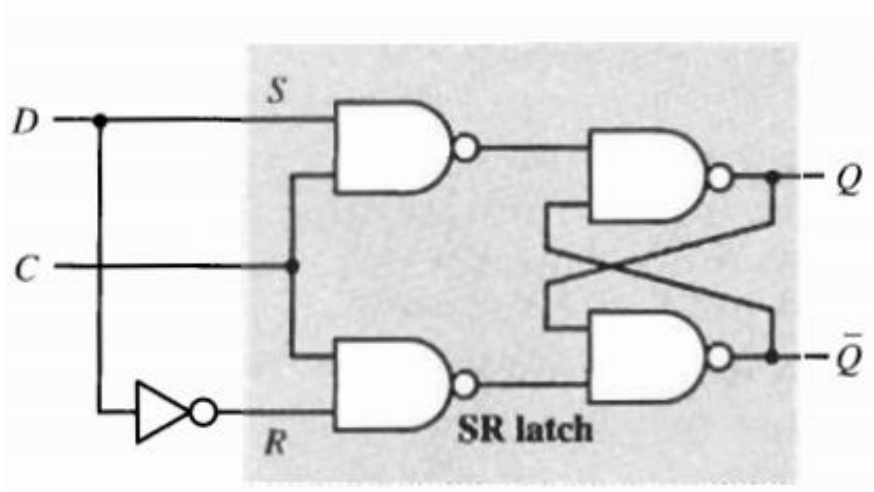


## Delay Latch (D-Latch) ✓

❖ با استفاده از SR Latch های کنترلی ساخته می شوند.

❖ دیگر حالت  $S=R=1$  رخ نخواهد داد.

$$S = D ; R = \bar{D}$$



✓ جدول تحریک (Excitation Table):

Enable input $C$	Excitation input $D$	Present state $Q$	Next state $Q^*$	
0	x	0	0	Hold
0	x	1	1	
1	0	0	0	Store 0
1	0	1	0	
1	1	0	1	Store 1
1	1	1	1	

$$Q^* = DC + \bar{C}Q$$

if  $C = 0 \rightarrow Q^* = Q$  (Hold)

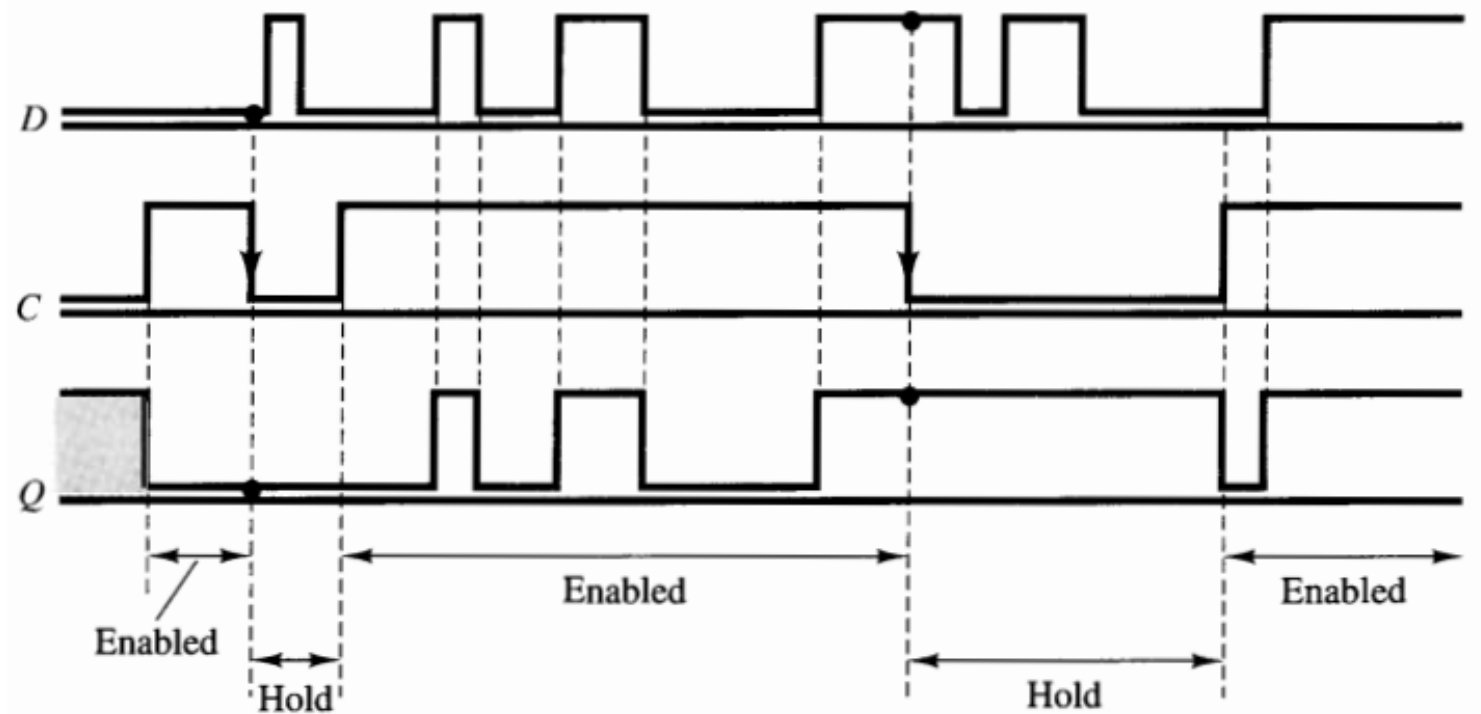
if  $C = 1 \rightarrow Q^* = D$  (Gated or Enabled Mode)

✓ معادله مشخصه (Characteristic Eq.):

در این مدار زمانی که  $C=1$  است، اطلاعات روی  $D$  عیناً به خروجی منتقل می شود. انگار مسیری از ورودی به خروجی داریم.

## Delay Latch (D-Latch) ✓

❖ دیاگرام زمانی:

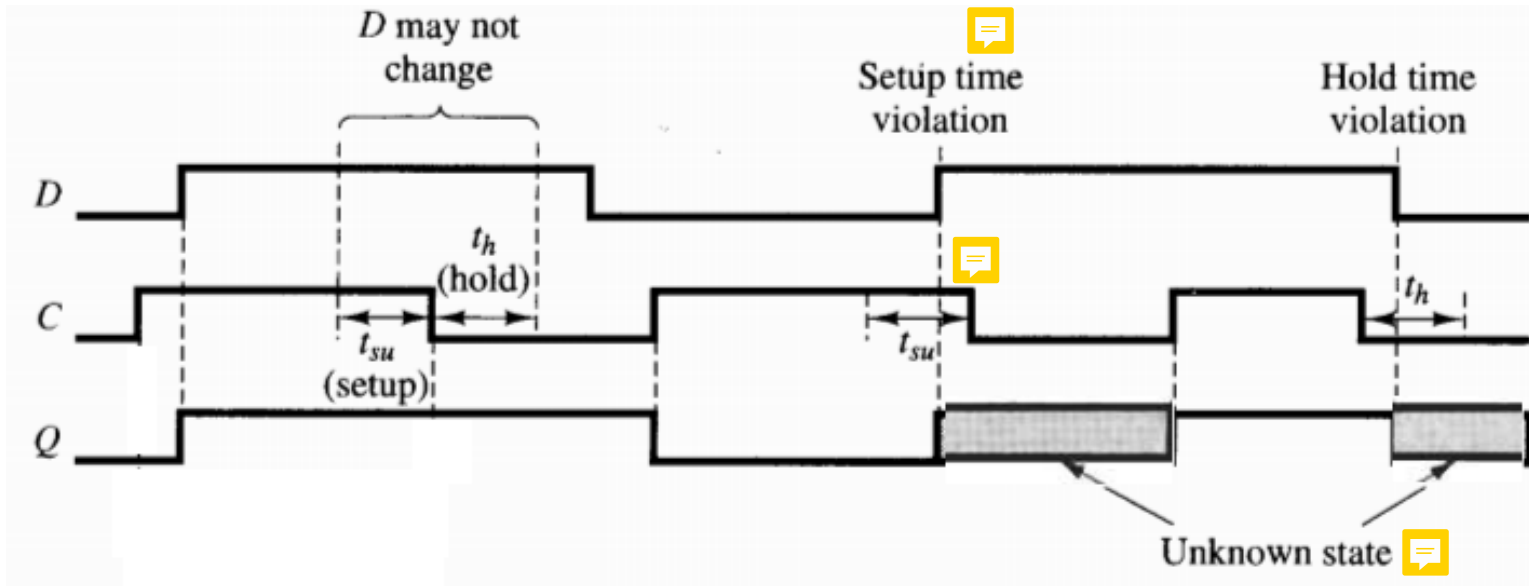


## Delay Latch (D-Latch) ✓

❖ D نباید در زمان های خیلی نزدیک به تغییرات C تغییر کند. ☞

❖ Setup time ( $t_{su}$ ): مدت زمان قبل از تغییرات C که در این زمان، D نباید تغییر کند. ☞

❖ Hold time ( $t_h$ ): مدت زمان بعد از تغییرات C که در این زمان، D نباید تغییر کند. ☞



## ✓ فلیپ-فلاپ (Flip-Flop)

❖ در latch ها تا زمانی که سیگنال کنترلی فعال است، مدار به تغییر ورودی ها پاسخ می دهد (حساس به سطح است).

❖ اگر چندین latch به یکدیگر متصل باشند، خروجی همه آن ها مدام در حال تغییر است.

❖ خروجی نهایی مدار غیرقابل پیش بینی و ممکن است نوسانی یا ناپایدار شود.

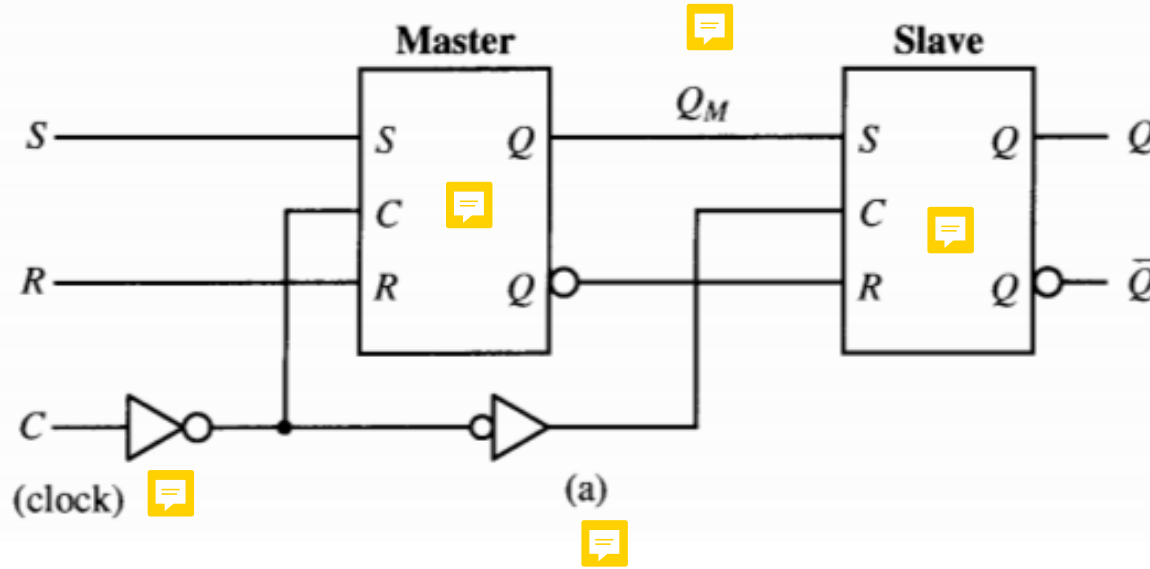
❖ برای کنترل بیشتر روی latch ها، مدار را به گونه ای طراحی می کنند که خروجی فقط در لبه های (لبه بالارونده یا پایین رونده) سیگنال کنترلی (کلاک) به تغییرات ورودی پاسخ دهد.

❖ به اینگونه مدارها که حساس به لبه کلاک هستند، فلیپ فلاپ گویند.



# Master-Slave SR Flip-Flop ✓

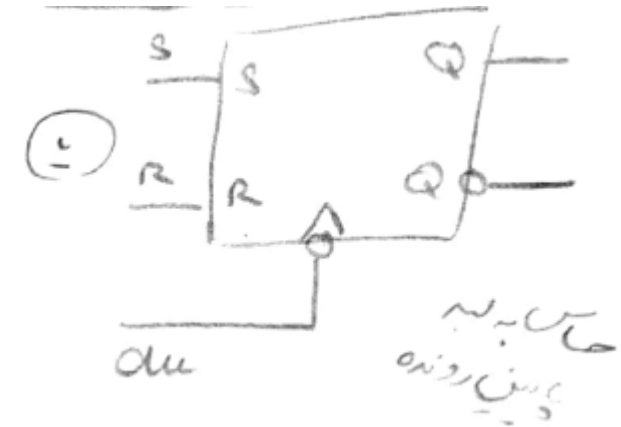
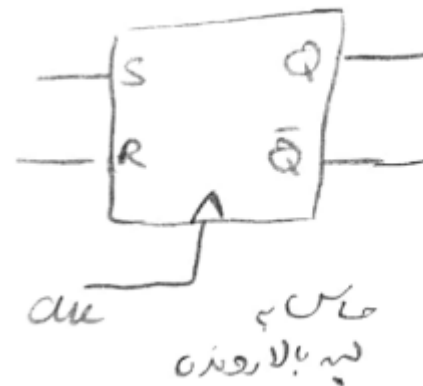
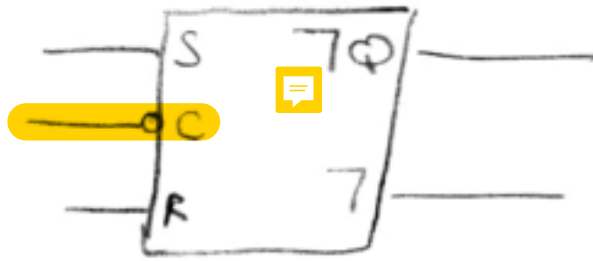
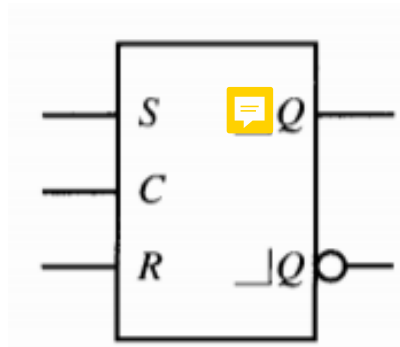
❖ از دو latch بصورت Master و Slave ساخته می شود.



$$C = 0 \Rightarrow \begin{cases} \text{Master: Gated} \\ \text{Slave: Hold} \end{cases}$$

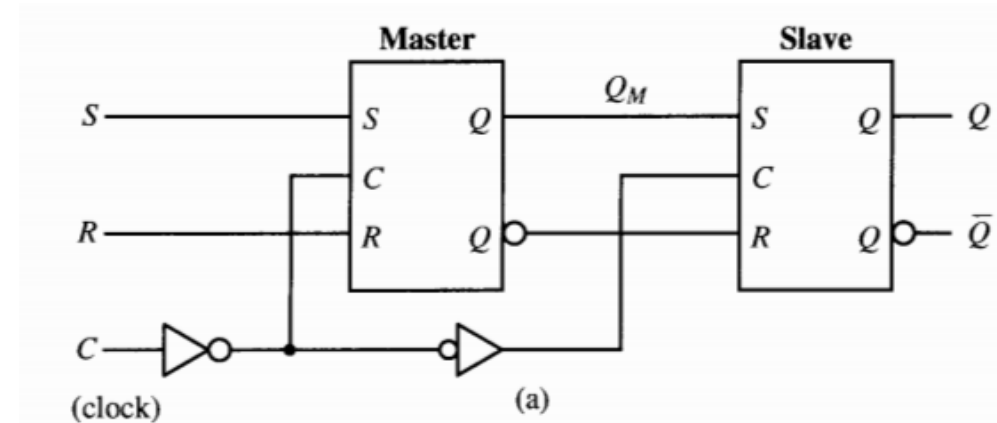
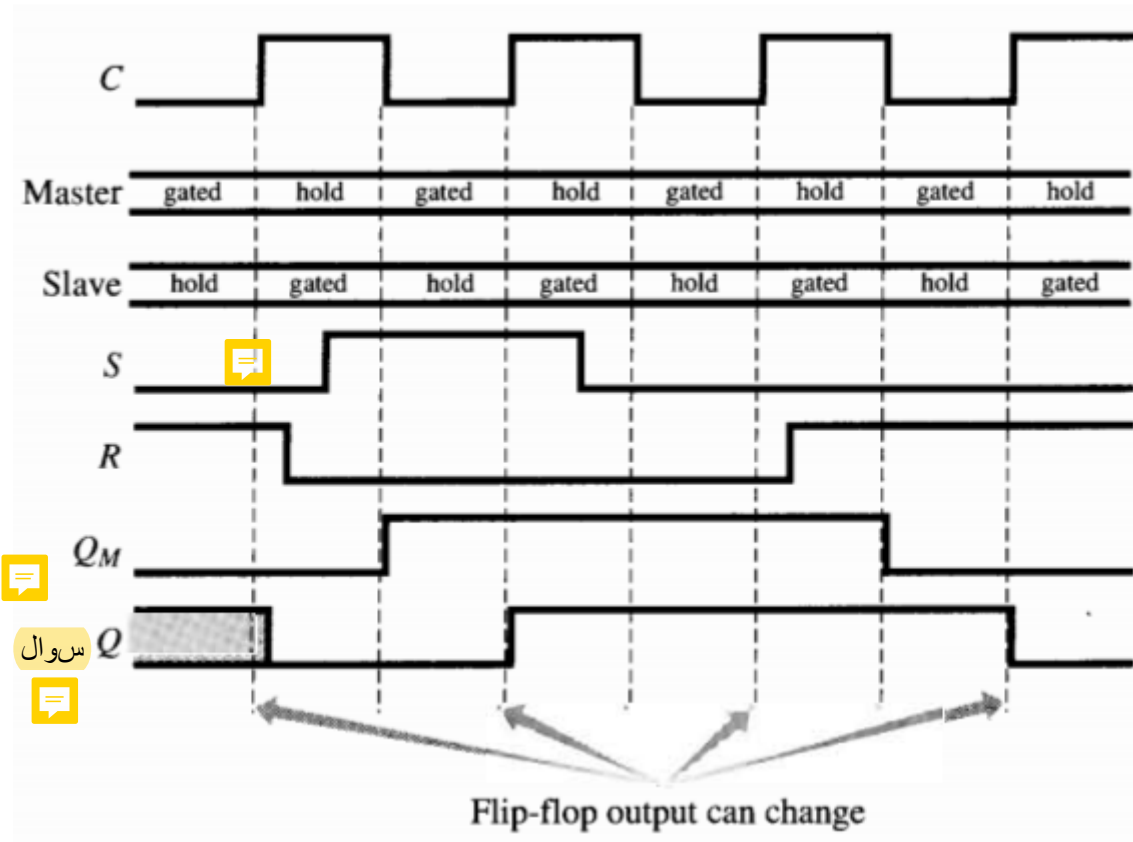
$$C = 1 \Rightarrow \begin{cases} \text{Master: Hold} \\ \text{Slave: Gated} \end{cases}$$

❖ به این مدارها Edge-triggered نیز گویند. چون به لبه کلاک حساس هستند.



# Master-Slave SR Flip-Flop ✓

❖ دیاگرام زمانی:

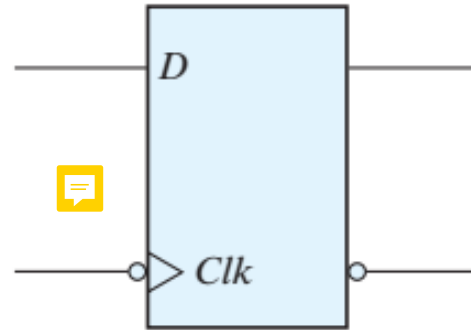
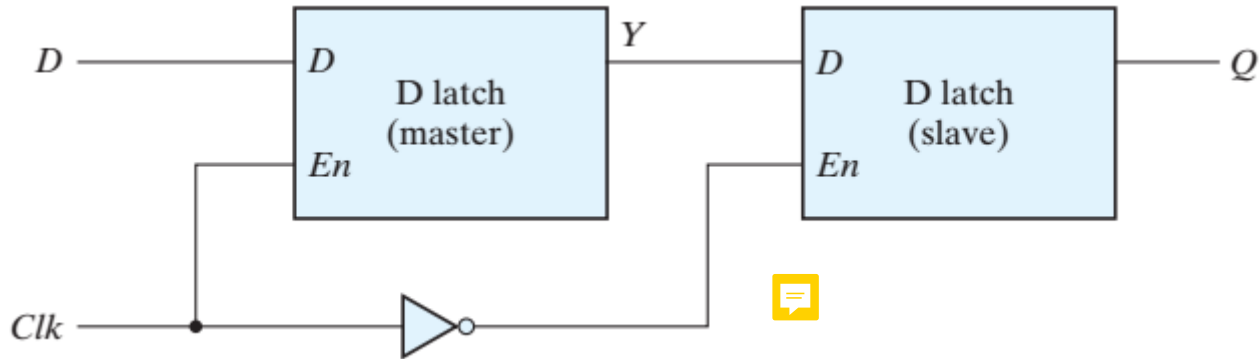
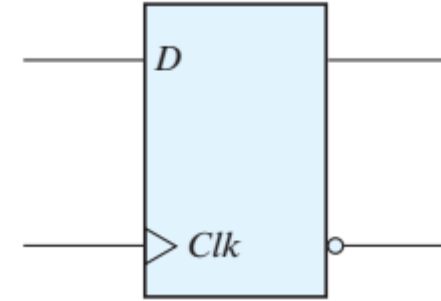
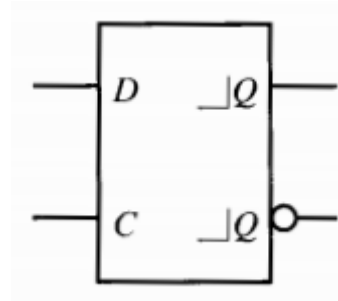
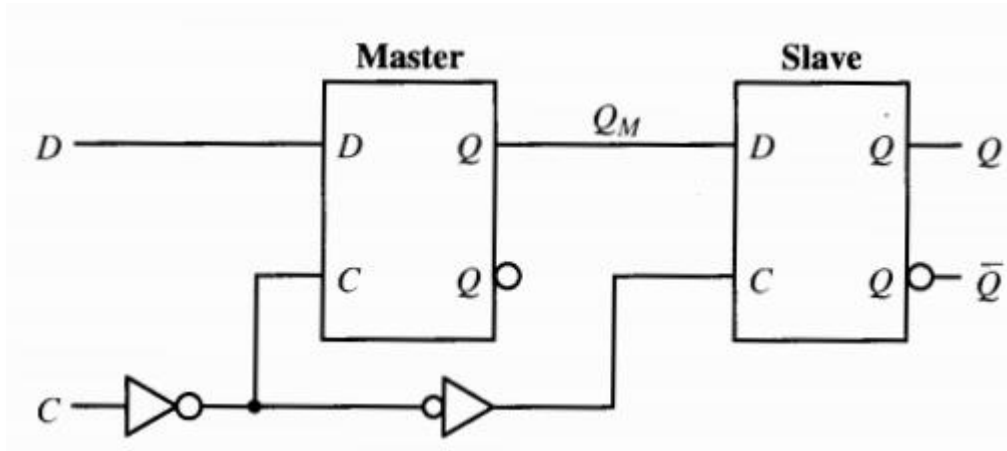


S	R	Q	C	Q*
0	0	0	⏏	0 No change
0	0	1	⏏	1
0	1	0	⏏	0 Reset
0	1	1	⏏	0
1	0	0	⏏	1 Set
1	0	1	⏏	1
1	1	0	⏏	× Not allowed
1	1	1	⏏	×

$$Q^* = S + \bar{R}Q$$

رفتار آن شبیه یک latch بوده با این تفاوت که خروجی تنها در لبه های کلاک تغییر می کند.

# Master-Slave D Flip-Flop ✓



❖ مقداری که در خروجی فلیپ فلاپ تولید می شود، همان مقداریست که تا قبل از کلاک، در طبقه Master تولید شده است.



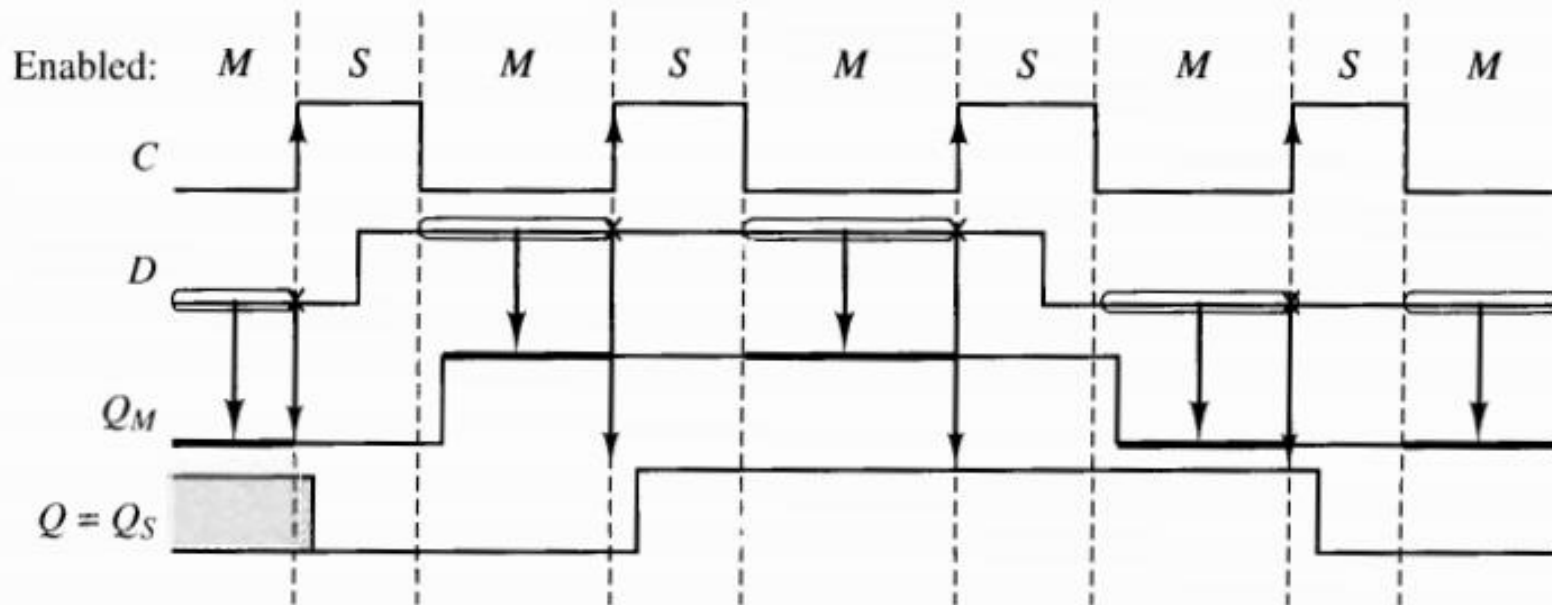
# Master-Slave D Flip-Flop ✓

❖ در لبه های کلاک، مقدار خروجی همان مقدار D خواهد بود.

$D$	$Q$	$C$	$Q^*$
0	0	⏏	0
0	1	⏏	0
1	0	⏏	1
1	1	⏏	1

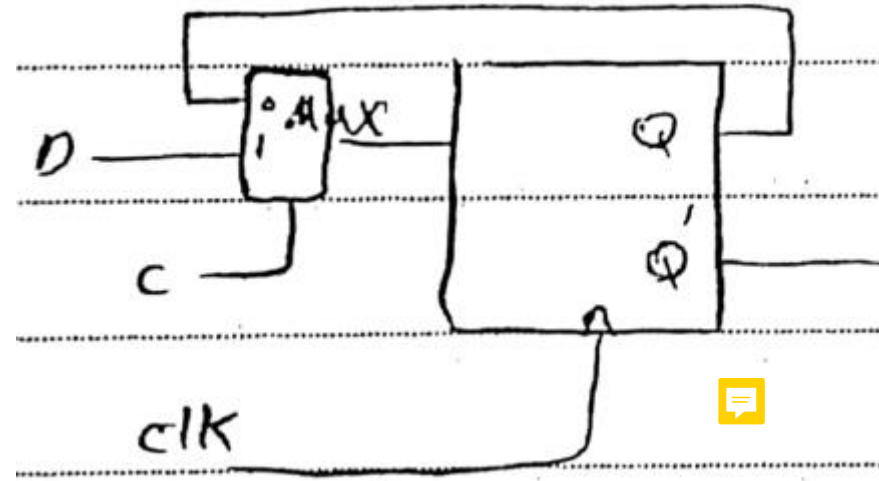
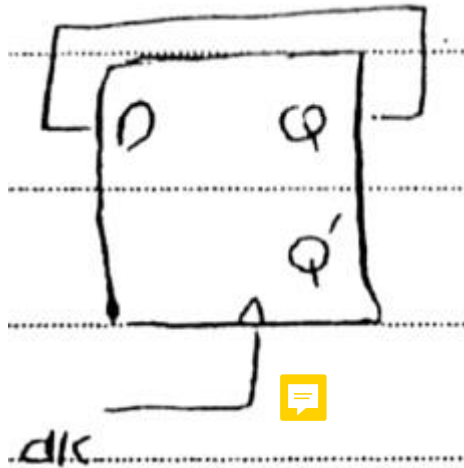


$$Q^* = D$$



## Master-Slave D Flip-Flop ✓

❖ در D-FF. ها حالت latch نداریم. مدار یا در حالت set است یا reset.



$C = 0 \Rightarrow \text{latch Mode}$

$C = 1 \Rightarrow D - FF.$

# JK Flip-Flop ✓

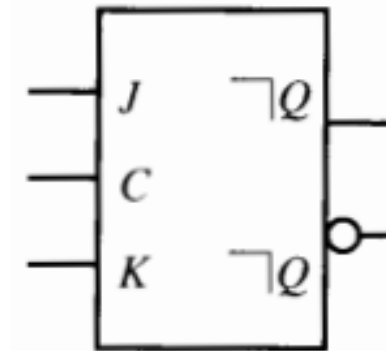
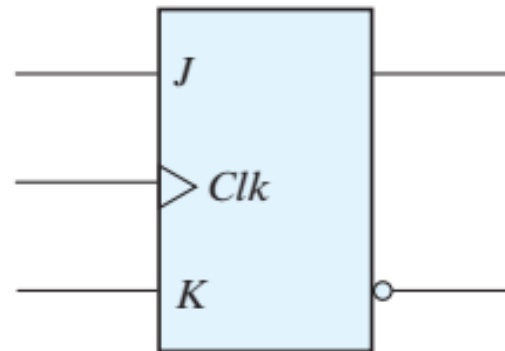
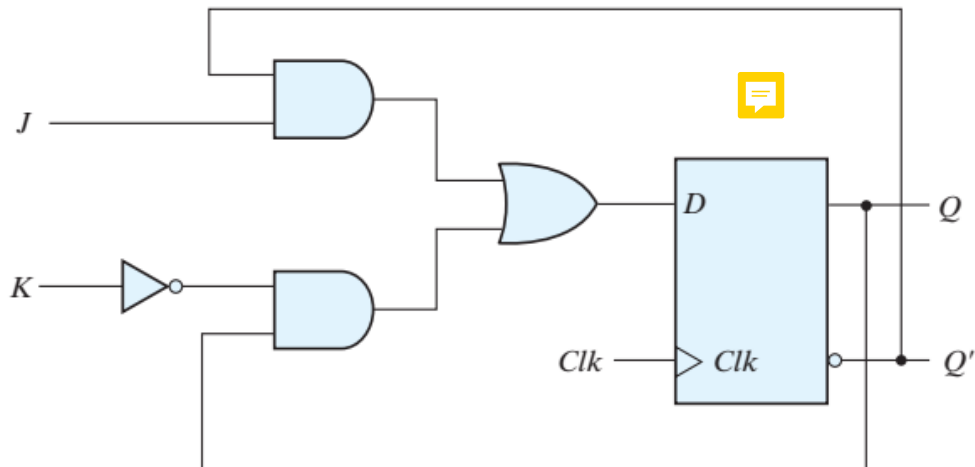
❖ هر نوع فلیپ فلاپ دیگری را می توان با استفاده از D-FF ها ساخت.

❖ JK FF دارای چهار حالت خواهد بود.

$J$	$K$	$Q$	$C$	$Q^*$
0	0	0		0 Hold
0	0	1		1
0	1	0		0 Reset
0	1	1		0
1	0	0		1 Set
1	0	1		1
1	1	0		1 Toggle
1	1	1		0

$JK$		$J$			
		00	01	11	10
$Q$	0	0	0	1	1
	1	1	0	0	1
		$K$			

$$Q^* = \bar{K}Q + J\bar{Q}$$

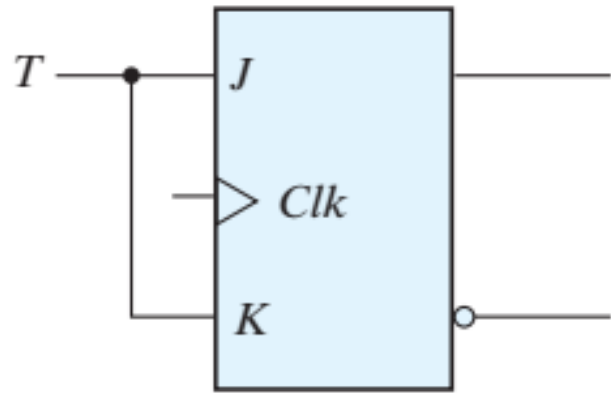


## T Flip-Flop ✓

❖ یک JK FF است که پایه های J و K آن بهم وصل شده باشند.

❖ این فلیپ فلاپ فقط دارای حالت latch و Toggle خواهد بود.

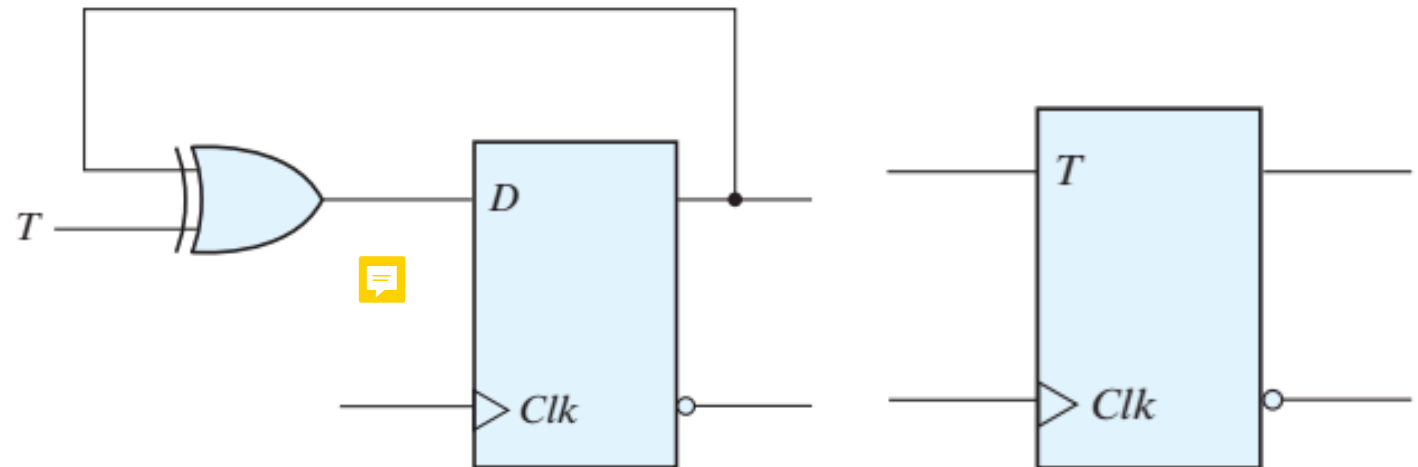
❖ برای طراحی شمارنده ها استفاده می شود



$T$	$Q(t + 1)$
0	$Q(t)$ No change
1	$Q'(t)$ Complement

$$Q^* = J\bar{Q} + \bar{K}Q$$

$$= T\bar{Q} + \bar{T}Q \quad Q^* = T \oplus Q$$



## ✓ خلاصہ

F.F	latch	set	Reset	Toggle	$Q_{t+1}$	بیس شرط
R.S	✓	✓	✓	X	$S + R'Q_t$	$RS = 0$
D	X	✓	✓	X	D	—
JK	✓	✓	✓	✓	$JQ' + K'Q$	—
T	✓	X	X	✓	$T \oplus Q$	—

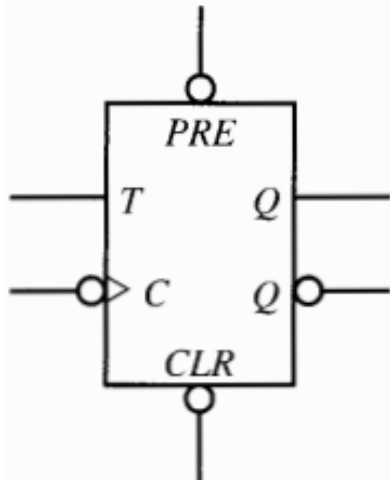
## Direct Inputs ✓

❖ روی همه فلیپ فلاپ ها دو پایه دیگر برای دادن حالت اولیه وجود دارد.

❖ این ورودی ها بصورت Active Low بوده و بصورت غیرهمزمان با کلاک کار می کنند. (Asynchronous Inputs)

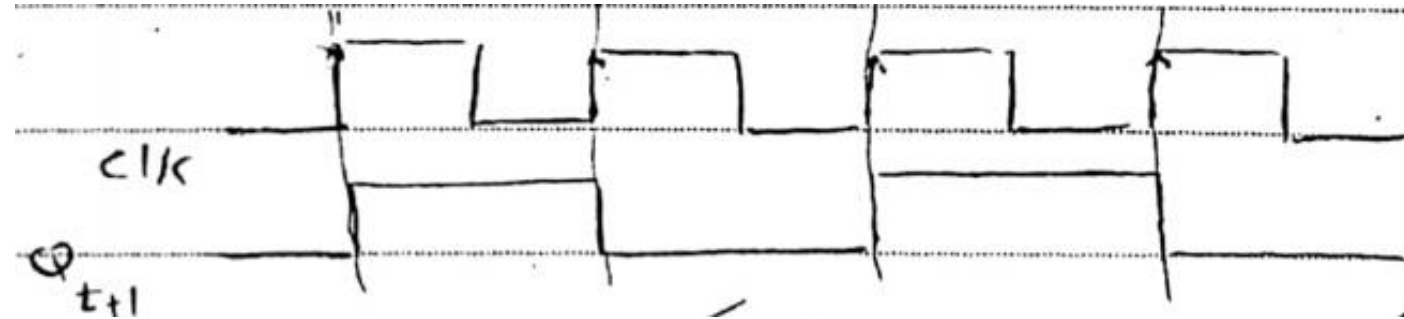
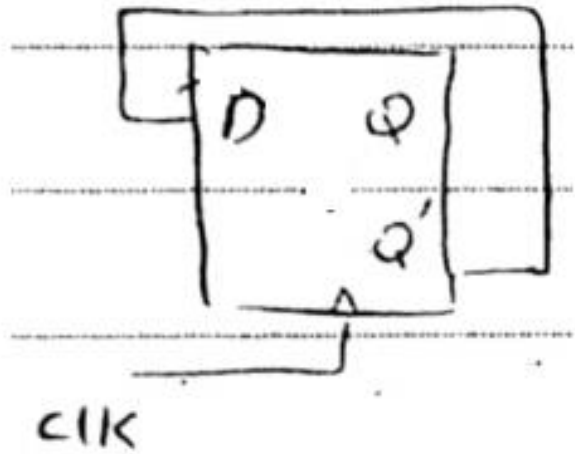
❖ اگر پایه CLR صفر باشد، خروجی فلیپ فلاپ صفر می شود.

❖ اگر پایه PRE صفر باشد، خروجی فلیپ فلاپ یک می شود.

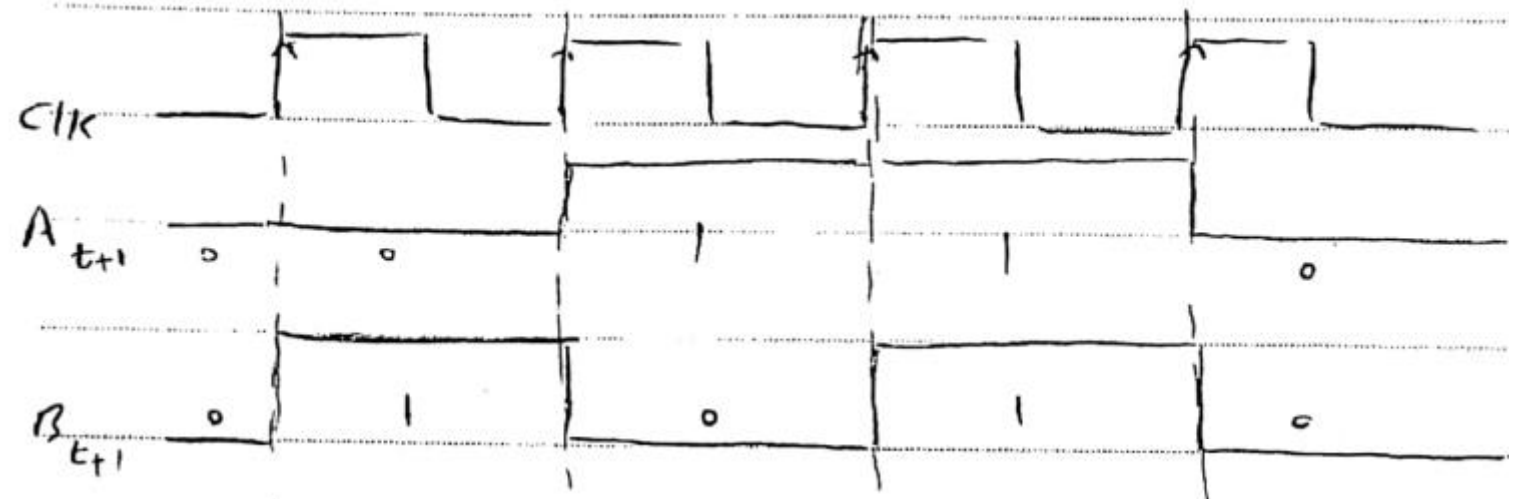
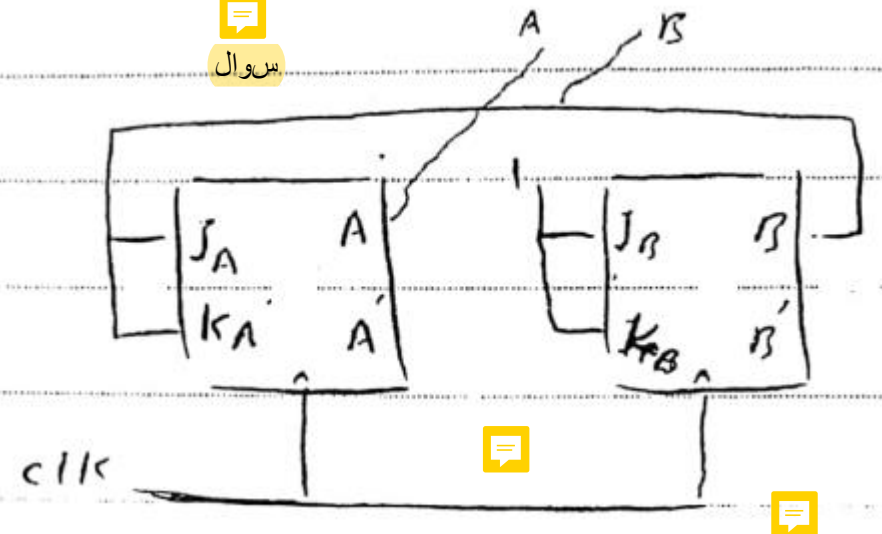


## ✓ چند مثال

❖ خروجی مدارهای زیر را به ازای حالت اولیه صفر رسم کنید.

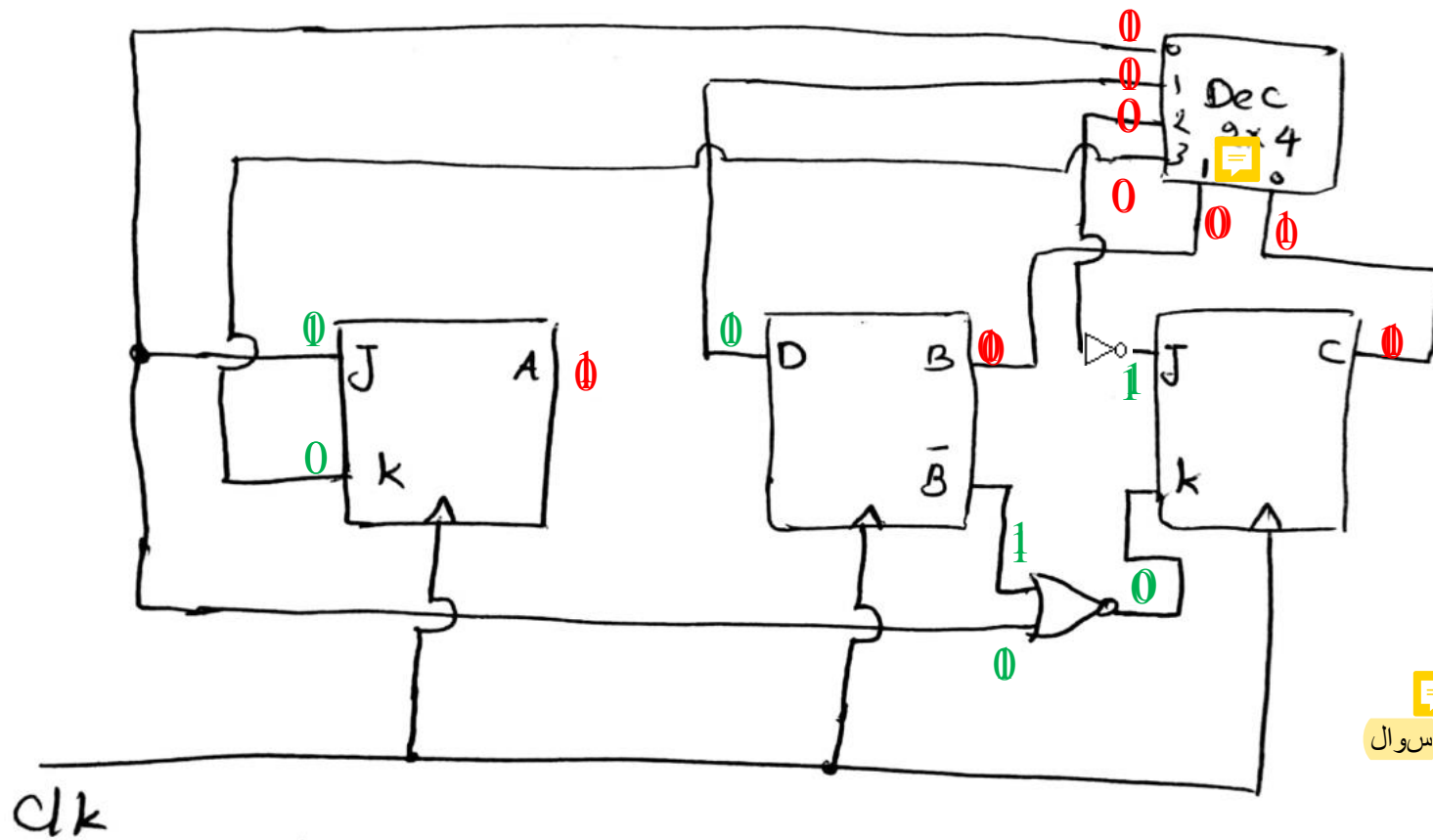


سوال



## ✓ چند مثال

❖ خروجی های مدار زیر را بعد از ۲ پالس ساعت بدست آورید. حالت اولیه همه فلیپ فلاپ ها صفر است.



	A	B	C
حالت اولیه	0	0	0
پالس اول	1	0	1
پالس دوم	1	1	1

سوال



## ✓ آنالیز مدارهای ترتیبی

- ❖ یک مدار ترتیبی داده شده را آنالیز کرده و رفتارش را تشخیص می دهیم.
- ❖ می خواهیم خروجی مدار را به ازای رشته ای از بیت های ورودی تشخیص دهیم.

### ❖ معادله حالت (State Equation):

✓ حالت بعدی را برحسب ورودی و حالت فعلی بیان می کند.

✓ در لحظه کلاک، ورودی D حالت فلیپ-فلاپ را تعیین می کند.

$$A(t + 1) = A(t)x(t) + B(t)x(t)$$

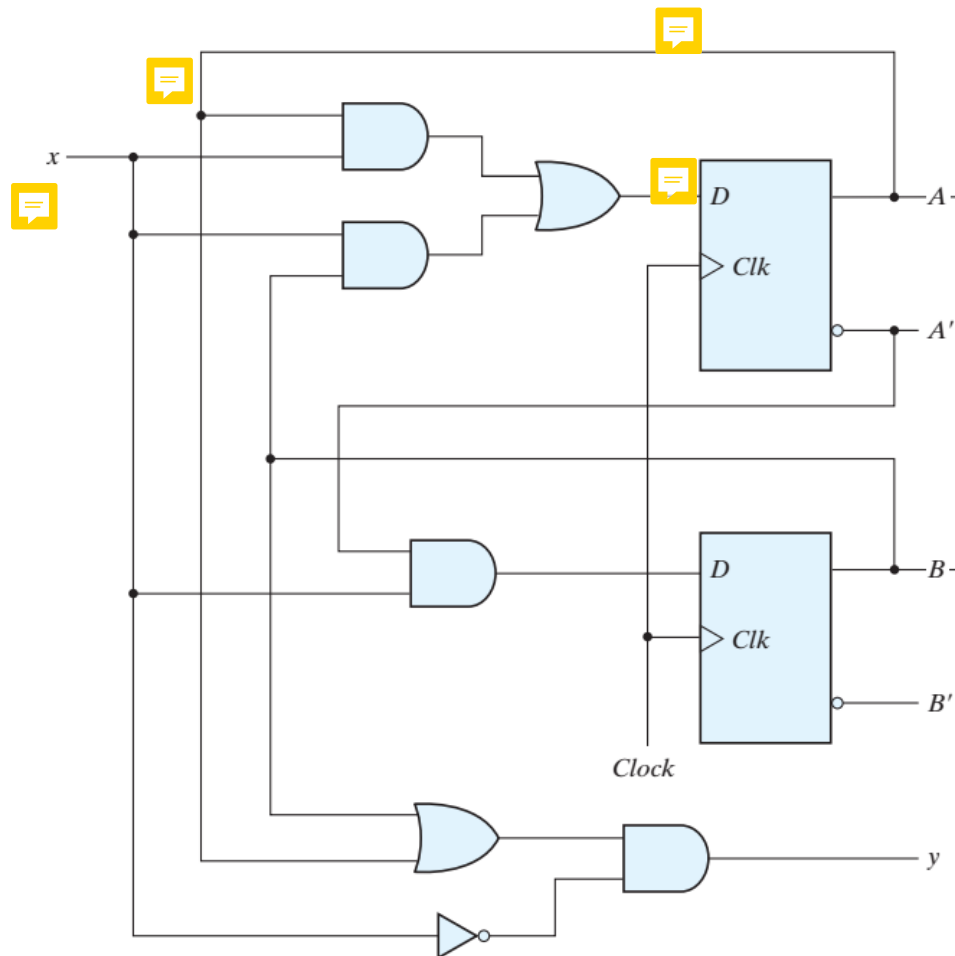
$$B(t + 1) = A'(t)x(t)$$

معادلات حالت

$$y(t) = [A(t) + B(t)]x'(t)$$

معادله خروجی برحسب ورودی و حالت فعلی فلیپ-فلاپ ها در زمان t

$$y = (A + B)x'$$



## ✓ آنالیز مدارهای ترتیبی

### ❖ جدول حالت (State Table):


✓ نشان دهنده رشته ای از ورودی ها، خروجی ها و حالات فلیپ-فلاپ ها است.

✓ براساس معادلات حالت تکمیل می شود.

✓ هر فلیپ-فلاپ دارای دو state است. پس در کل ۴ حالت داریم.

✓ اگر مدار دارای m فلیپ-فلاپ و n ورودی باشد،  $2^{m+n}$  ردیف داریم

Present State		Input	Next State		Output
A	B		A	B	
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



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

Input Present State	Input	
	0	1
0 0	00/0	01/0
0 1	00/1	11/0
1 0	00/1	10/0
1 1	00/1	10/0

nextstate/output

Input Present State	Input	
	0	1
A	A/0	B/0
B	A/1	D/0
C	A/1	C/0
D	A/1	C/0

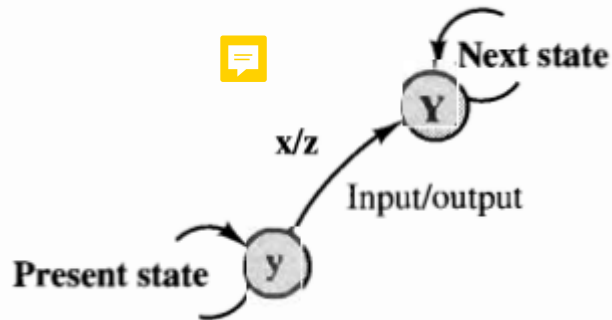
## ✓ آنالیز مدارهای ترتیبی

### ❖ دیاگرام حالت (State Diagram):

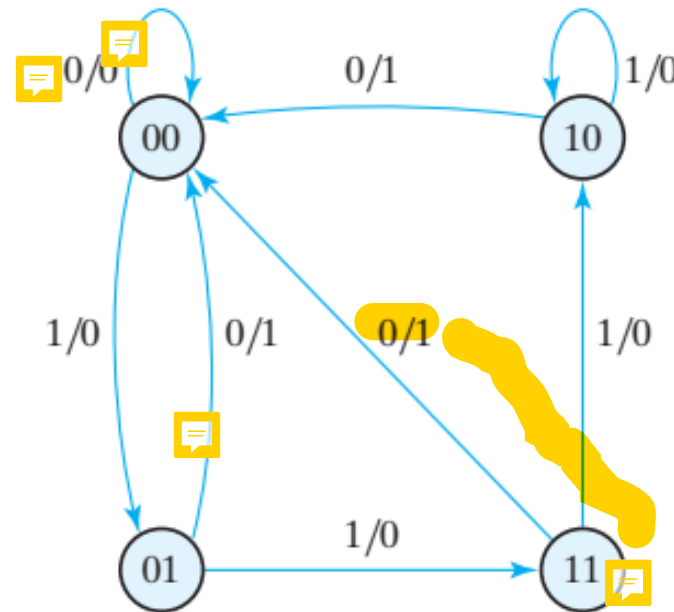
✓ اطلاعات جدول حالت را بصورت گرافیکی نشان می دهد.

✓ می توانیم رشته بیت خروجی را به ازای رشته بیت ورودی بیابیم.

✓ در این مدار تا زمانی که ورودی یک باشد، خروجی صفر است و به محض آمدن اولین صفر، خروجی یک شده و مدار به حالت اولیه برمی گردد.



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

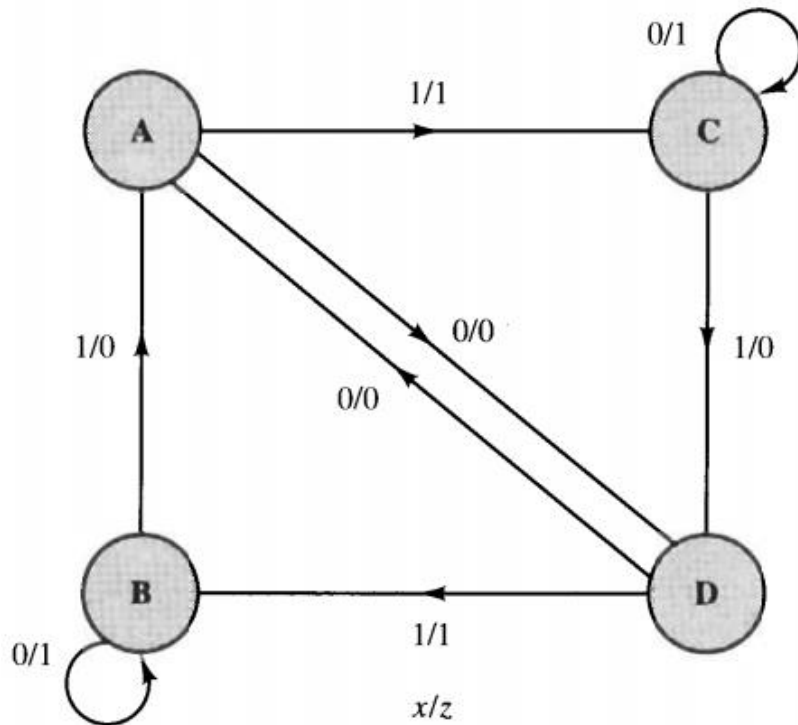


## ✓ آنالیز مدارهای ترتیبی

❖ مثال: یک مدار ترتیبی دارای جدول و دیاگرام حالت زیر است. رشته بیت خروجی را به ازای رشته بیت ورودی داده شده بیابید.

		Input $x$	
		0	1
Present state	A	D/0	C/1
	B	B/1	A/0
	C	C/1	D/0
	D	A/0	B/1

$x = 0110101100$



Time:	0	1	2	3	4	5	6	7	8	9	10
Present state:	A	D	B	A	D	B	B	A	C	C	C
Input:	0	1	1	0	1	0	1	1	0	0	
Next state:	D	B	A	D	B	B	A	C	C	C	
Output:	0	1	0	0	1	1	0	1	1	1	

## ✓ آنالیز مدارهای ترتیبی

### ❖ معادله ورودی فلیپ-فلاپ ها (FF Input Equations):

❏  $D_A = Ax + Bx$

❏  $D_B = A'x$

$y = (A + B)x' \quad Q(t + 1) = D_Q$

✓ به معادلاتی که ورودی فلیپ-فلاپ ها را مشخص می کنند، گویند.

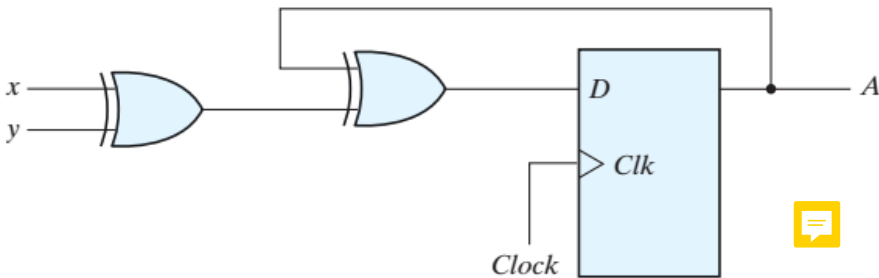
### ❖ آنالیز با D. Flip-Flop

✓ نوشتن معادله ورودی فلیپ-فلاپ ها و معادله خروجی

✓ نوشتن معادلات حالت با استفاده از معادلات ورودی

✓ تکمیل جدول حالت با استفاده از معادلات حالت

✓ رسم دیاگرام حالت



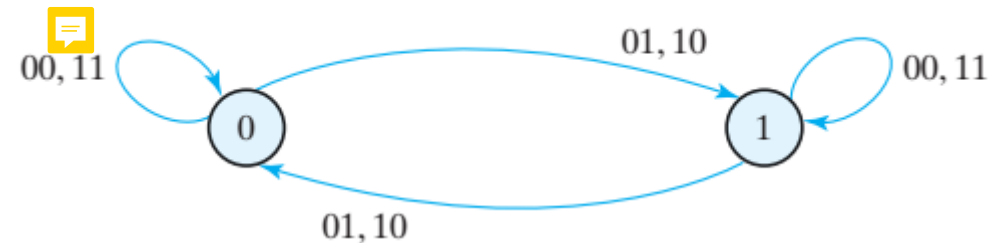
$D_A = A \oplus x \oplus y$

❏  $A(t + 1) = A \oplus x \oplus y$

❏

Input present State	00	01	10	11
0	0	1	1	0
1	1	0	0	1

Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



## ✓ آنالیز مدارهای ترتیبی

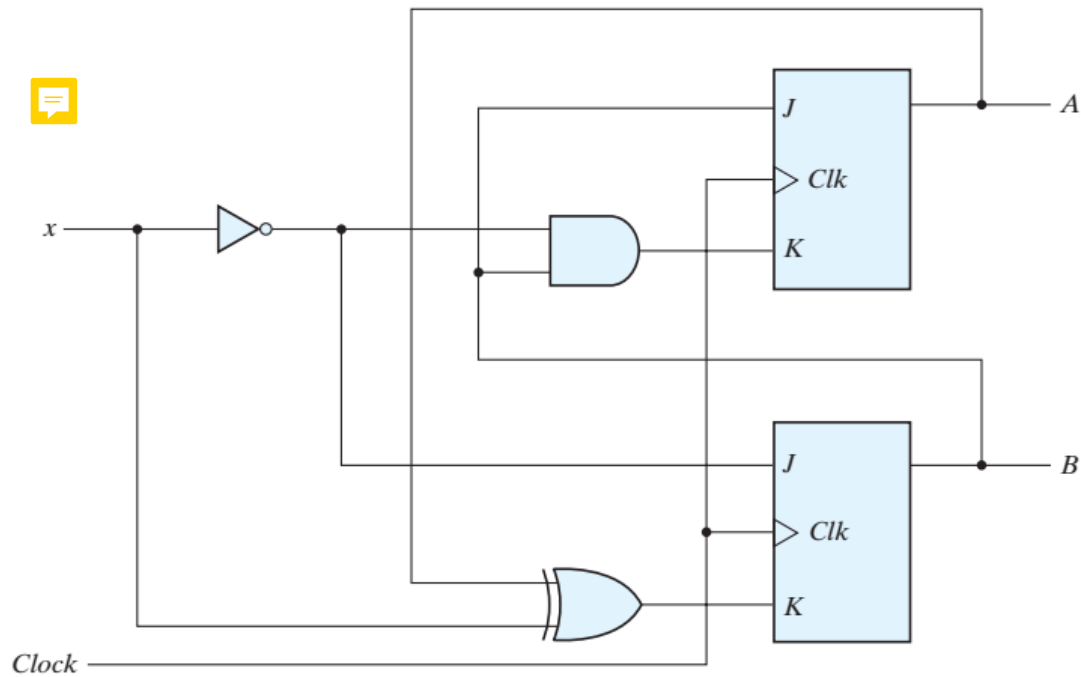
### ❖ آنالیز با JK. Flip-Flop

✓ نوشتن معادله ورودی فلیپ-فلاپ ها

✓ نوشتن مقادیر باینری هر معادله ورودی

✓ استفاده از جدول تحریک هر فلیپ-فلاپ برای تکمیل جدول حالت

✓ رسم دیاگرام حالت



$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	$J_A$	$K_A$	$J_B$	$K_B$
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0



$$J_A = B \quad K_A = Bx'$$

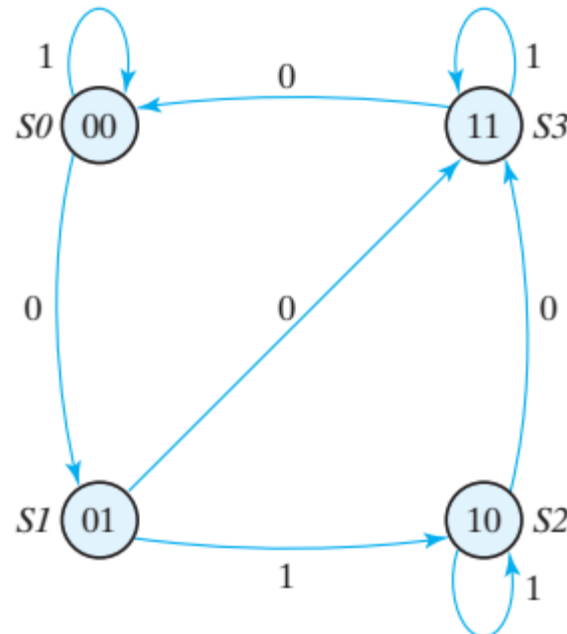
$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

$$A(t+1) = JA' + K'A$$

$$B(t+1) = JB' + K'B$$

$$A(t+1) = BA' + (Bx')'A = A'B + AB' + Ax \quad B(t+1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$$

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



✓ دیگر در جدول حالت نیازی به ستون Input FF. نیست.

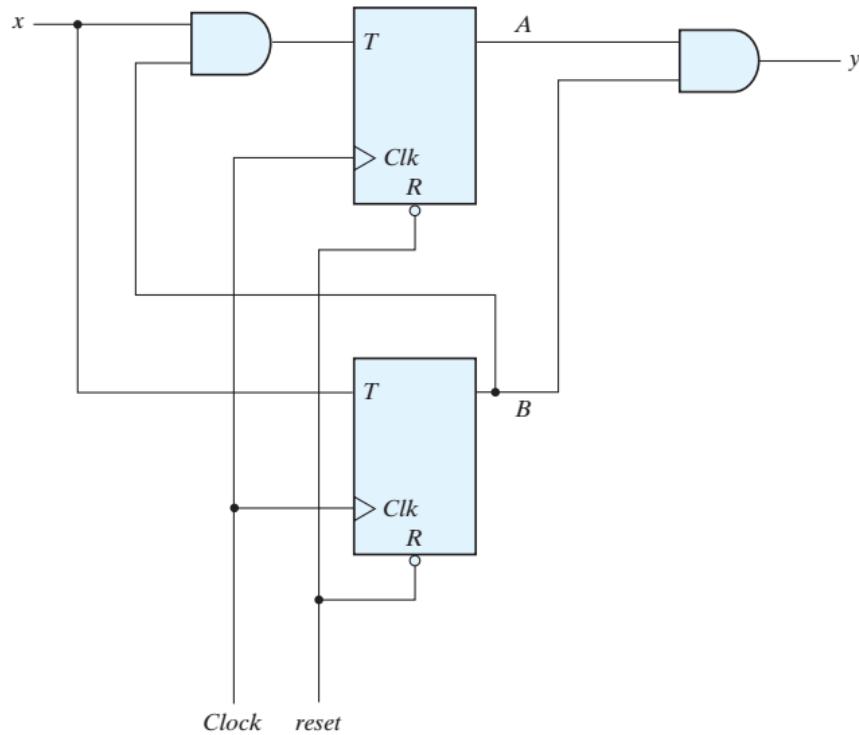
## ✓ آنالیز مدارهای ترتیبی

### ❖ آنالیز با JK. Flip-Flop

✓ می توان از معادله مشخصه فلیپ-فلاپ ها نیز استفاده کرده و معادلات حالت را نوشت.

# ✓ آنالیز مدارهای ترتیبی

## ❖ آنالیز با T Flip-Flop:



$$A(t + 1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$$

$$B(t + 1) = x \oplus B$$

Present State		Input $x$	Next State		Output $y$
$A$	$B$		$A$	$B$	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

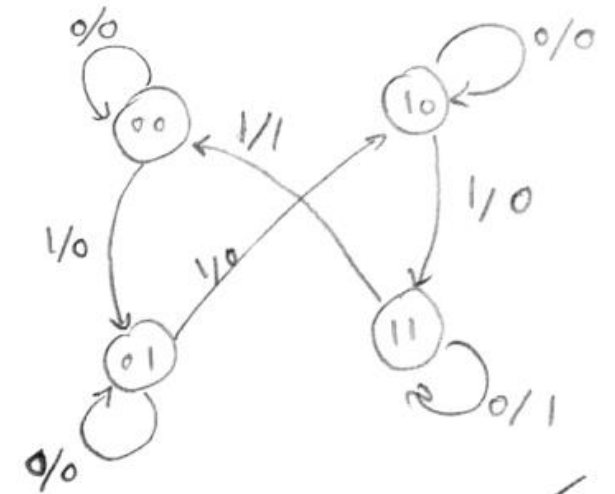
$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$

$$Q(t + 1) = T \oplus Q = T'Q + TQ'$$

	0	1
00	00/0	01/0
01	01/0	10/0
10	10/0	11/0
11	11/1	00/1



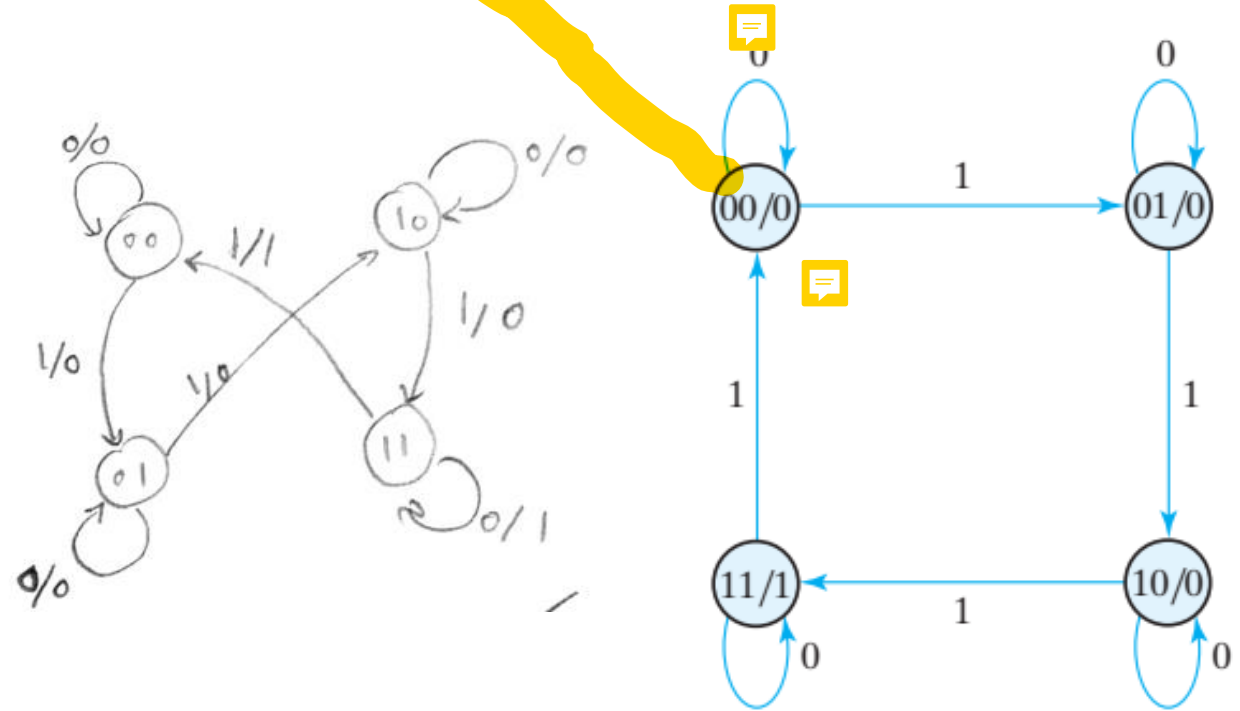


## ✓ آنالیز مدارهای ترتیبی

❖ آنالیز با T Flip-Flop:

$$y = AB$$

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



	0	1
00	00/0	01/0
01	01/0	10/0
10	10/0	11/0
11	11/1	00/1

✓ خروجی فقط به Present State ها وابسته است.

## ✓ آنالیز مدارهای ترتیبی


### ❖ مدل میلی (Mealy) و مور (Moore)


$$y = (A + B)x'$$

✓ در مدل میلی، خروجی تابعی از ورودی و حالت فعلی فلیپ-فلاپ ها است.

$$y = AB$$

✓ در مدل مور، خروجی فقط تابعی از حالت فعلی فلیپ-فلاپ ها است.

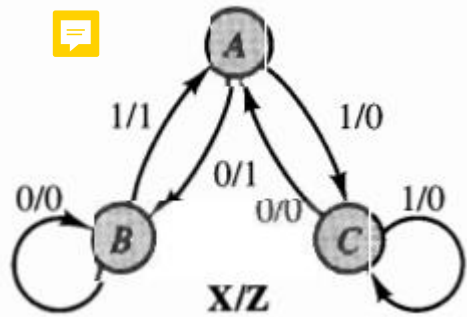
✓ در مدل مور، تغییرات خروجی مدار ترتیبی با کلاک همزمان می شود. 

✓ در مدل میلی، اگر در طول سیکل کلاک (قبل از آمدن کلاک بعدی)، ورودی تغییر کند، خروجی هم تغییر می کند. 

✓ در مدل میلی، خروجی ها می توانند مقادیر نادرست لحظه ای داشته باشند.

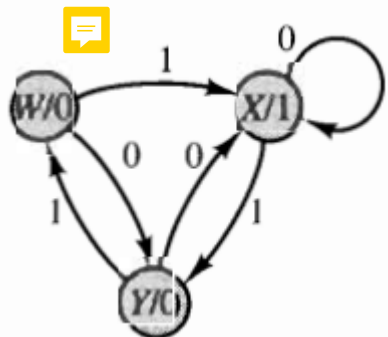
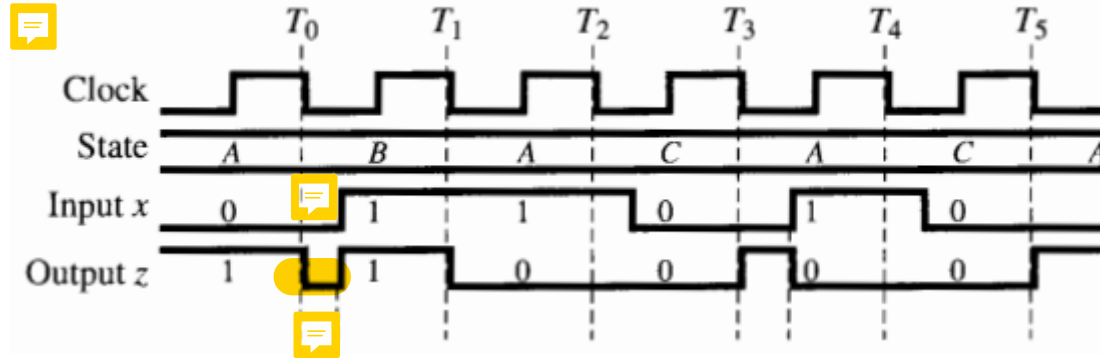
# ✓ آنالیز مدارهای ترتیبی

❖ مدل میلی (Mealy) و مور (Moore)

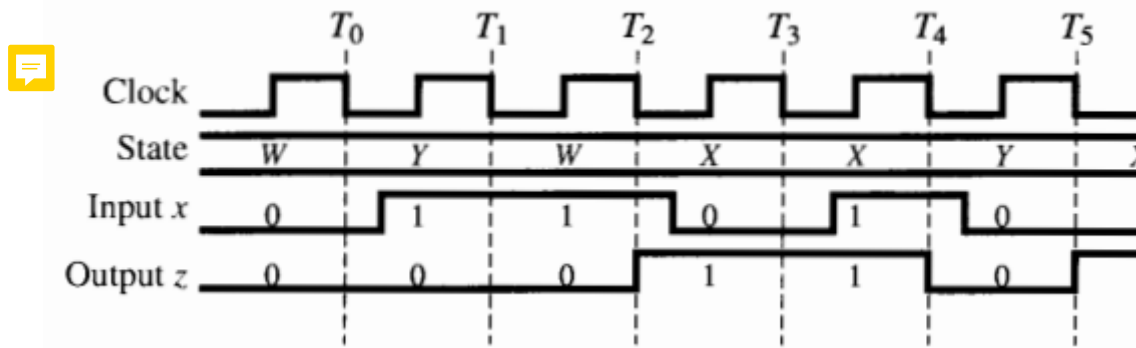


$x = 011010.$

$z = 110000$



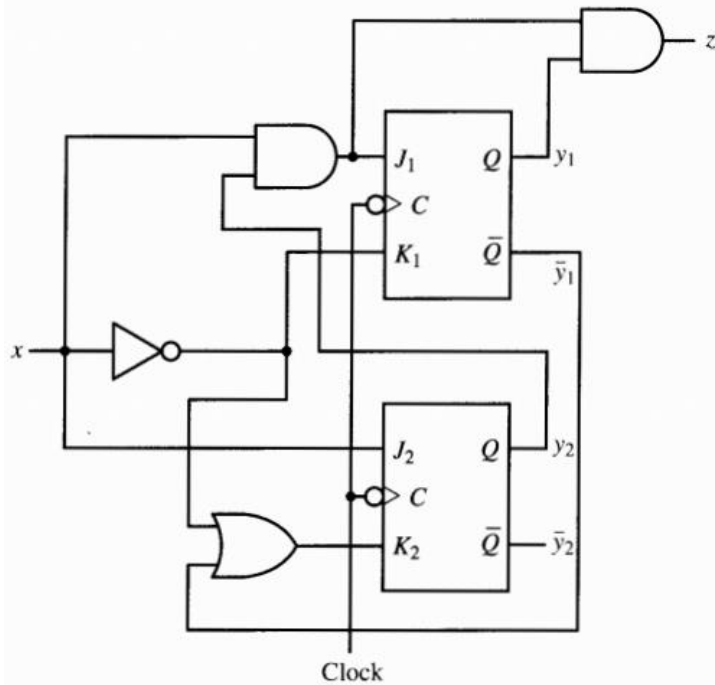
Input: 0 1 1 0 1 0  
Output: 0 0 0 1 1 0



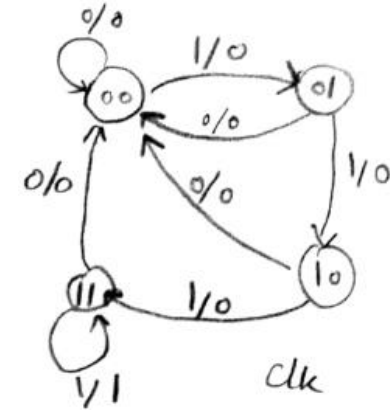
## ✓ آنالیز مدارهای ترتیبی

❖ مثال: در مدار روبرو، رشته بیت خروجی را به ازای رشته بیت ورودی داده شده بیابید.

$$x = 0011110$$



	0	1
00	00/0	01/0
01	00/0	10/0
10	00/0	11/0
11	00/0	11/1
A	B	

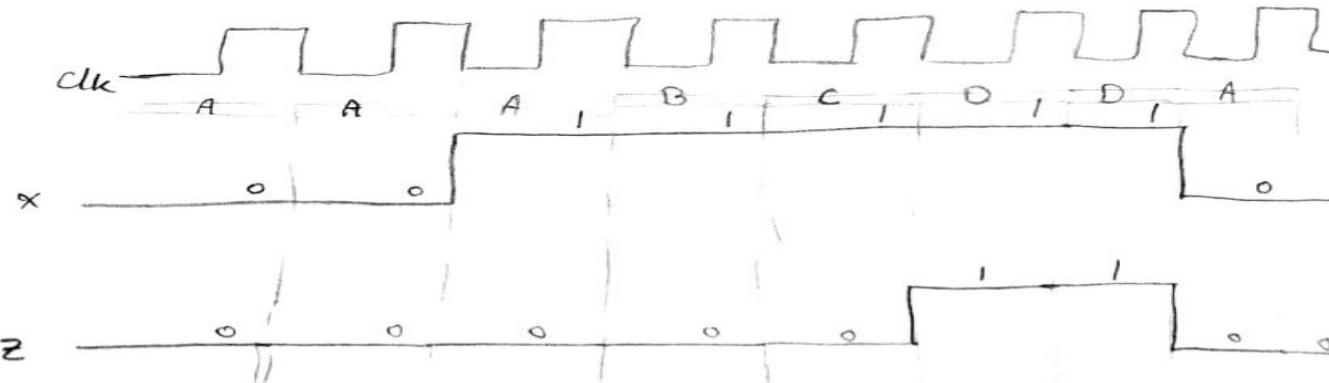


State	00	00	00	01	10	11	11	11	00
Input	0	0	1	1	1	1	1	0	
output	0	0	0	0	0	1	1	0	

$$\begin{cases} J_A = Bx & ; & K_A = \bar{x} \\ J_B = x & ; & K_B = \bar{x} + \bar{A} \end{cases} \quad Z = ABx$$

state equations:

$$\begin{cases} A(u+1) = J_A \bar{A} + K_A A = Bx \bar{A} + xA = x(A + \bar{A}B) = x(A + B) \\ B(u+1) = J_B \bar{B} + K_B B = x\bar{B} + xAB = x(\bar{B} + BA) = x(A + \bar{B}) \end{cases}$$



## ✓ طراحی (سنتز) مدارهای ترتیبی

❖ یک صورت مسئله داده می شود و باید براساس آن مدار ترتیبی مورد نظر را طراحی کنیم.

### ❖ کاهش حالت (State Reduction):

✓ در یک مدار ترتیبی،  $m$  فلیپ – فلاپ باعث ایجاد  $2^m$  حالت (state) می شود.

✓ با کاهش تعداد state ها در جدول حالت، تعداد فلیپ – فلاپ ها کاهش می باید.

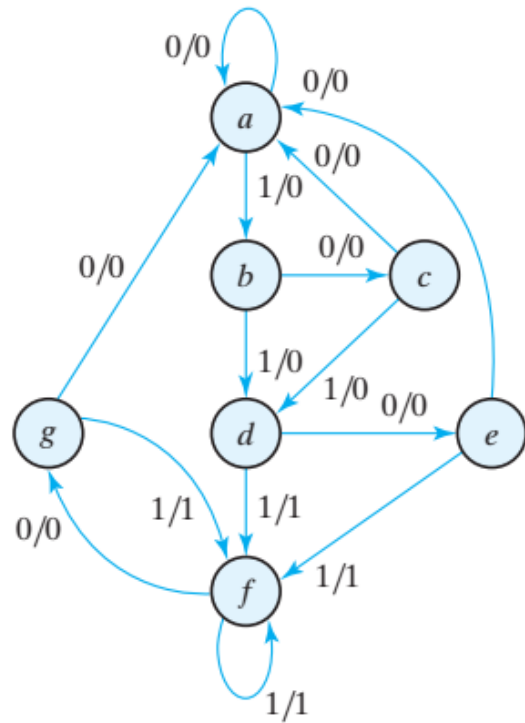
✓ کاهش state ها باید بگونه ای باشد که به ازای ورودی یکسان، خروجی تغییر نکند.

✓ در یک جدول حالت می توانیم از دو state معادل، یکی را حذف کنیم.

✓ دو state معادل هستند اگر به ازای ورودی مشابه، دقیقاً خروجی و next state یکسانی بدهند.

## ✓ طراحی (سنتز) مدارهای ترتیبی

### ❖ کاهش حالت (State Reduction):

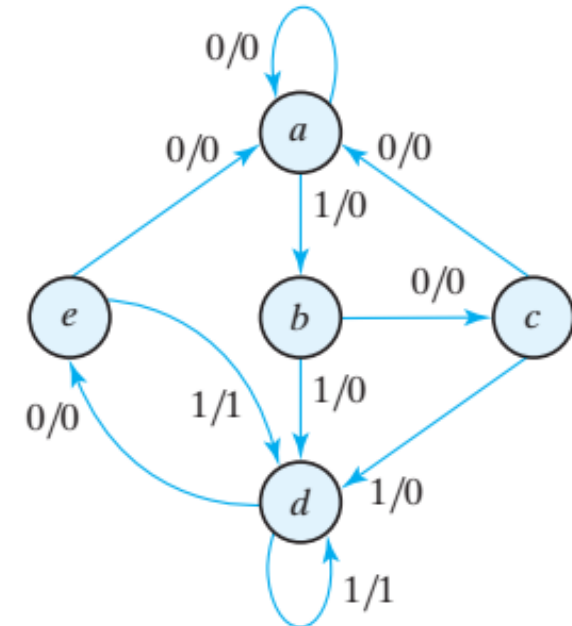


state	a	a	b	c	d	e	f	f	g	f	g	a
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1



## ✓ طراحی (سنتز) مدارهای ترتیبی

### ❖ State Assignment

- ✓ به منظور طراحی، لازم است که به state ها یک کد باینری اختصاص دهیم.
- ✓ برای یک مدار با  $m$  حالت، کد باینری باید  $n$  بیتی باشد بطوری که  $2^n \geq m$  باشد.
- ✓ حالت های غیراستفاده را بصورت don't care در نظر می گیریم.
- ✓ ساده ترین روش برای اختصاص کد به حالت ها، استفاده از اعداد باینری به ترتیب از کوچک به بزرگ است.
- ✓ به جدول حالتی که در آن کد باینری به state ها اختصاص داده شده باشد، Transition Table گویند.

## ✓ طراحی (سنتز) مدارهای ترتیبی

### ❖ روند طراحی (Design Procedure):

✓ رسم دیاگرام حالت از روی صورت مسئله و مشخصات خواسته شده

✓ کاهش تعداد state ها (State Reduction) در صورت نیاز

✓ اختصاص کد باینری به state ها (State Assignment)

✓ بدست آوردن جدول حالت با استفاده از کدها اختصاص یافته

✓ انتخاب نوع فلیپ – فلاپ

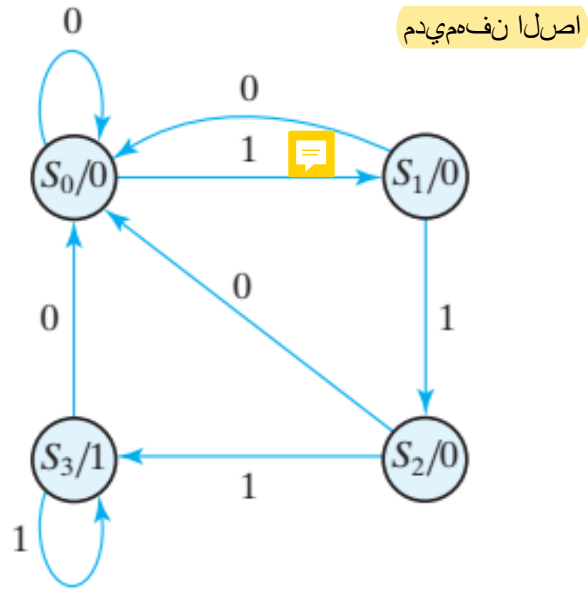
✓ بدست آوردن معادلات ورودی فلیپ – فلاپ ها و همچنین معادله خروجی (بصورت ساده شده)

✓ رسم مدار منطقی

❖ نکته: مهم ترین مرحله در طراحی، مرحله اول است که بتوانیم دیاگرام حالت را به درستی رسم کنیم.



## ✓ طراحی (سنتز) مدارهای ترتیبی



❖ مثال: با استفاده از D-FF ها یک مدار طراحی کنید که الگوی ۳ تا یک پشت سرهم یا بیشتر را در یک رشته بیت ورودی تشخیص دهد.

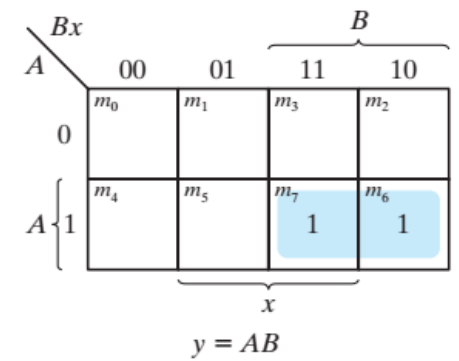
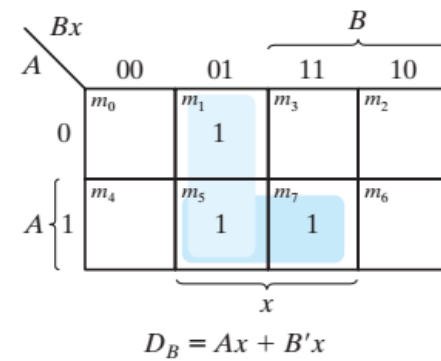
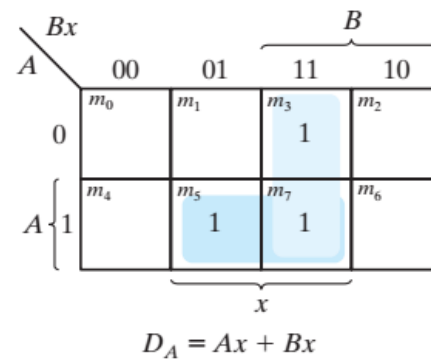
$$Q(t + 1) = D_Q$$

$$A(t + 1) = D_A(A, B, x) = \Sigma(3, 5, 7)$$

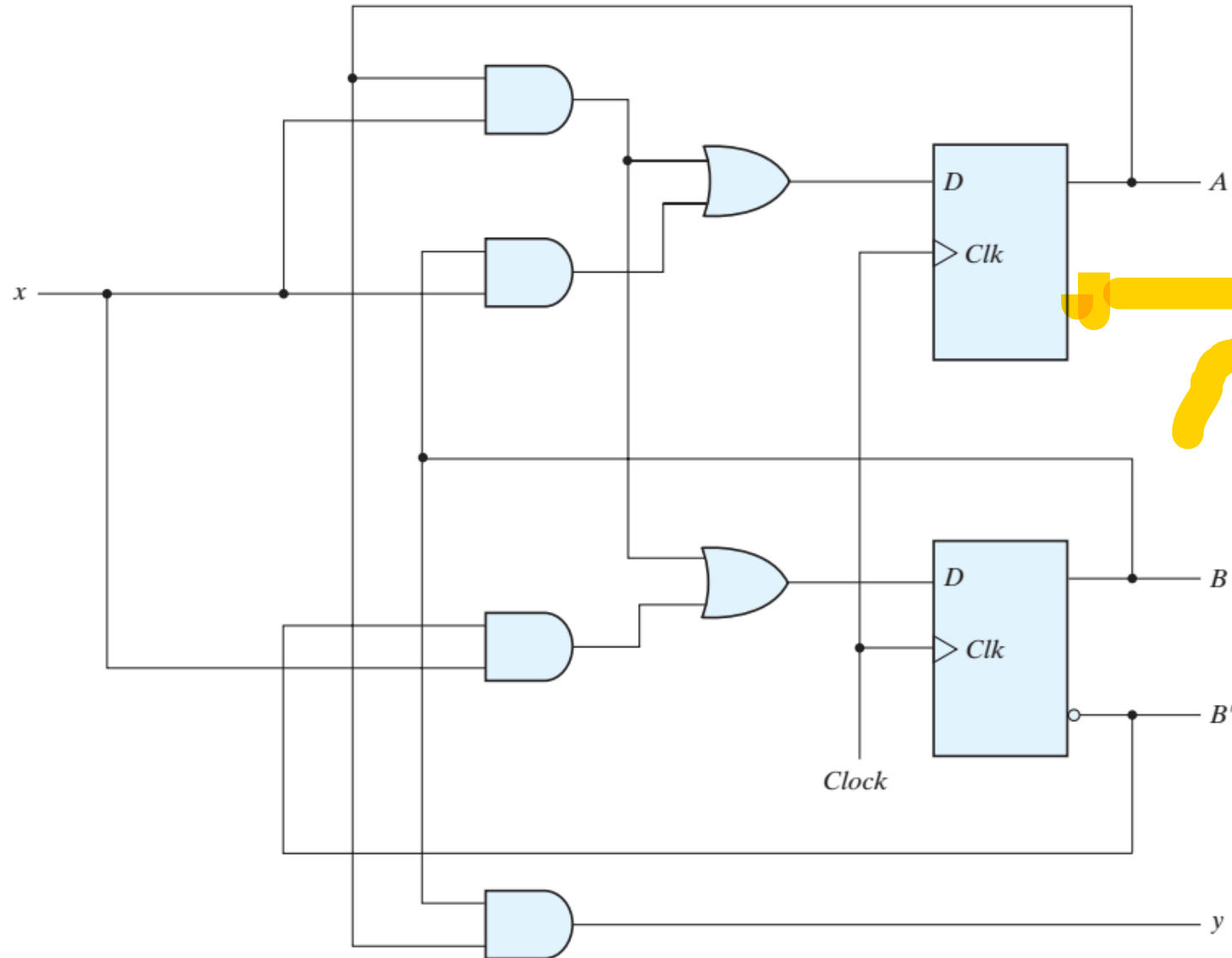
$$B(t + 1) = D_B(A, B, x) = \Sigma(1, 5, 7)$$

$$y(A, B, x) = \Sigma(6, 7)$$

Present State		Input $x$	Next State		Output $y$
$A$	$B$		$A$	$B$	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1



## ✓ طراحی (سنتز) مدارهای ترتیبی









## ✓ طراحی (سنتز) مدارهای ترتیبی


### ❖ جدول ورودی فلیپ-فلاپ (FF. Input Table):

✓ در طراحی با D-FF معادله ورودی فلیپ-فلاپ ها بصورت مستقیم از روی جدول حالت بدست می آید.

✓ برای انواع دیگر فلیپ-فلاپ ها باید رابطه ای بین حالت بعدی و معادلات ورودی بیابیم.

✓ جدول ورودی فلیپ فلاپ ها جدولی است که ورودی های مورد نیاز را برای هرتغییر حالتی لیست می کند و از روی جدول تحریک بدست می آید.

 $Q(t)$	$Q(t + 1)$	$J$	$K$	
0	0	0	X	
0	1	1	X	
1	0	X	1	
1	1	X	0	

 $Q(t)$	$Q(t + 1)$	$T$
0	0	0
0	1	1
1	0	1
1	1	0

✓ با استفاده از این جدول، معادله ورودی فلیپ-فلاپ ها را می یابیم.

## ✓ طراحی (سنتز) مدارهای ترتیبی

❖ مثال: جدول حالت روبرو را با JK FF پیاده سازی کنید.

Present State		Input	Next State		Flip-Flop Inputs			
A	B		A	B	$J_A$	$K_A$	$J_B$	$K_B$
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

	0	1
00	00	01
01	01	10
10	10	11
11	11	00

( $J_A K_A$ )

	0	1
00	00	01
01	01	10
10	10	11
11	11	00

( $J_B K_B$ )

$$J_A = \sum m(2) + d(4, 5, 6, 7)$$

$$\Rightarrow J_A = B \bar{x}$$

$$K_A = \sum m(7) + d(0, 1, 2, 3)$$

$$\Rightarrow K_A = Bx$$

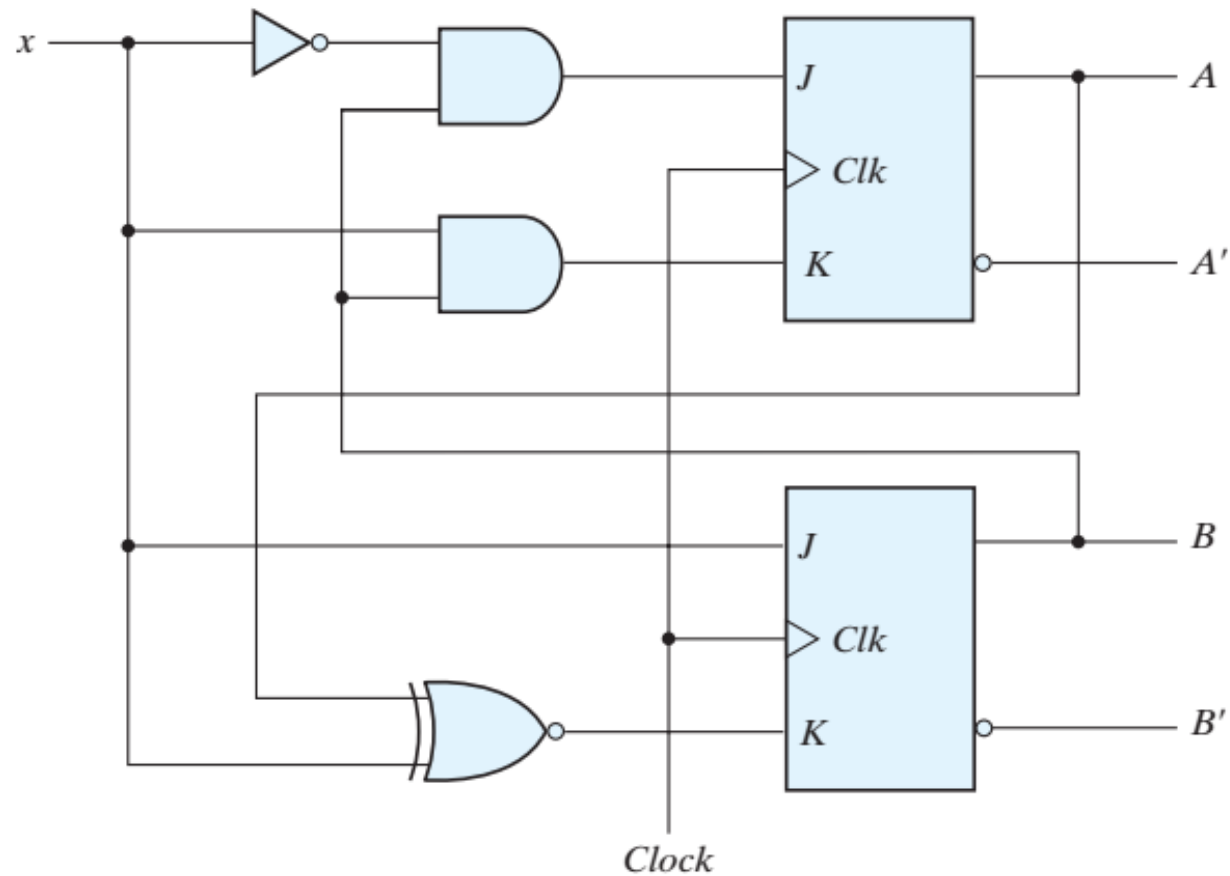
$$J_B = \sum m(1, 5) + d(2, 3, 6, 7)$$

$$\Rightarrow J_B = x$$

$$K_B = \sum m(2, 7) + d(0, 1, 4, 5)$$

$$\Rightarrow K_B = \bar{A} \bar{x} + Ax = A \odot x$$

## ✓ طراحی (سنتز) مدارهای ترتیبی



$$J_A = Bx'$$

..

$$K_A = Bx$$

..

$$J_B = x$$

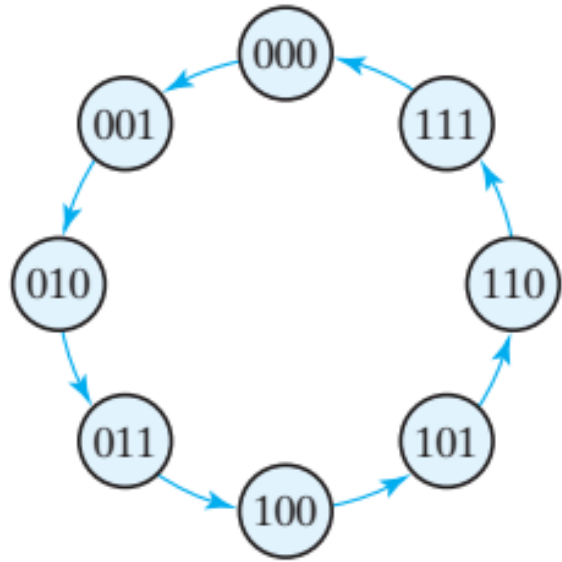
$$K_B = (A \oplus x)'$$

✓ طراحی با JK FF می تواند نسبت به D-FF دارای مدار ترکیبی ساده تری باشد. چون شرایط don't care دارد.

## ✓ طراحی (سنتز) مدارهای ترتیبی

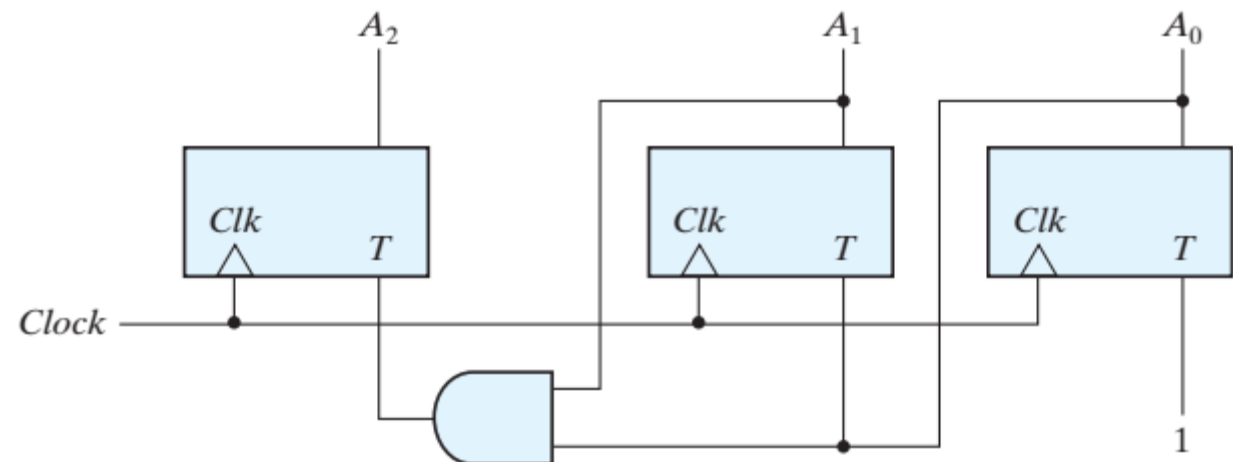
❖ مثال: با استفاده از T FF یک شمارنده ۳ بیتی چرخشی طراحی کنید.

✓ ورودی و خروجی نداریم و با هر کلاک به state بعدی می رویم.



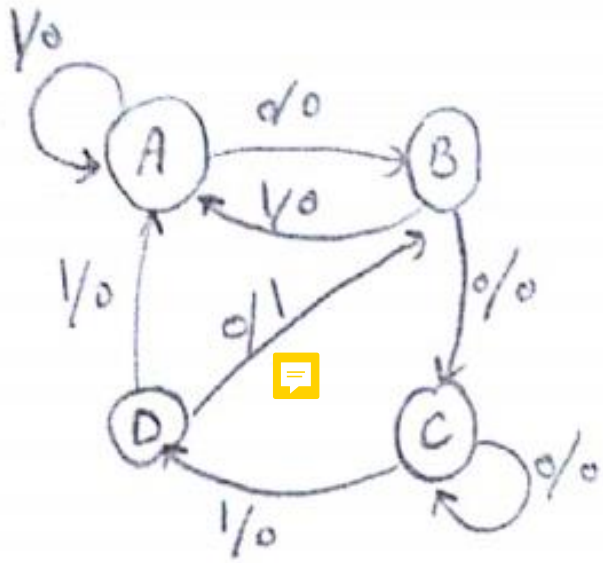
Present State			Next State			Flip-Flop Inputs		
$A_2$	$A_1$	$A_0$	$A_2$	$A_1$	$A_0$	$T_{A2}$	$T_{A1}$	$T_{A0}$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1

$A_2 \backslash A_1 A_0$		$A_1$			
		00	01	11	10
$A_2$	0	$m_0$	$m_1$	$m_3$	$m_2$
	1	$m_4$	$m_5$	$m_7$	$m_6$
$T_{A2} = A_1 A_0$				1	
$T_{A1} = A_0$			1	1	
$T_{A0} = 1$		1	1	1	1



## ✓ طراحی (سنتز) مدارهای ترتیبی

❖ مثال: با استفاده از JK FF مداری طراحی کنید که الگوی 0010 را تشخیص دهد.



	0	1
A	B/0	A/0
B	C/0	A/0
C	C/0	D/0
D	B/1	A/0

S.A →

	0	1
00	01/0	00/0
01	10/0	00/0
10	10/0	11/0
11	01/1	00/0

$A \uparrow D_B$

	0	1
00	0d	0d
01	1d	0d
10	d0	d0
11	d1	d0

$(J_A K_A)$

	0	1
00	1d	0d
01	d1	d1
10	0d	1d
11	d0	d1

$(J_B K_B)$

## ✓ طراحی (سنتز) مدارهای ترتیبی

	0	1
00	00	01
01	10	11
10	01	00
11	11	10

$$(J_A K_A)$$

00	1d	0d
01	d1	d1
10	od	1d
11	do	d1

 $(J_B k_B)$ 

$$J_A = \mathcal{L}^m(2) + d(4, 5, 6, 7)$$

0		1	d	d
1			d	d

$$\Rightarrow J_A = B \bar{K}$$

$$k_A = \bar{2}m(6) + \mathcal{L}(0, 1, 2, 3)$$

	00	01	11	10
0	d	d	d	
1	d	d		

$$\Rightarrow K_A = \beta \bar{x} \quad \text{--- } J_A$$

$$J_B = \bar{L}_m(0, 5) + d(2, 3, 6, 7)$$

0	1	1	1	1
1	1	1	1	1

$$J_3 = \bar{A}\bar{X} + AX = A \odot X$$

$$k_B = \bar{c}_m(2,3,7) + \mathcal{A}(0,1,4,5)$$

	10	11	12	13
0	1	1	1	1
1	1	1	1	1

$$K_3 = X + \bar{A}$$

