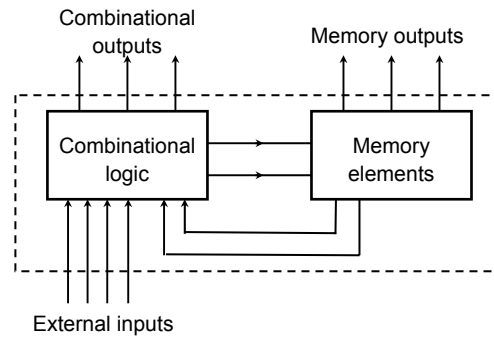


فصل ششم: طراحی و تحلیل مدارهای منطقی ترتیبی Sequential Logic Circuits: Analysis & Design

مقدمه:

- یک مدار ترتیبی شامل یک مسیر فیدبک بوده و از عناصر حافظه استفاده می کند.



Sequential circuit = Combinational logic + Memory Elements

مقدمه

- دو نوع مدار ترتیبی وجود دارد:

❖ **سنکرون (Synchronous)**: خروجی ها فقط در زمانهای مشخصی تغییر میکنند

❖ **آسنکرون (asynchronous)**: خروجی ها در هر زمانی می توانند تغییر کنند.

- مولتی وایبراتورها (Multivibrator) یک گروه از مدارهای ترتیبی را تشکیل میدهند:

❖ **Bistable** (2 stable states)

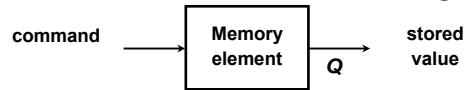
❖ **Monostable** or **one-shot** (1 stable state)

❖ **Astable** (no stable state)

- ❖ **flip-flop** و **latch** نمونه ای از مدارهای Bistable می باشند.

عناصر حافظه

- عنصر حافظه (Memory element): وسیله ای است که یک مقدار را بطور نامحدود نگه داشته و یا با اعمال ورودی، آنرا تغییر می دهد.



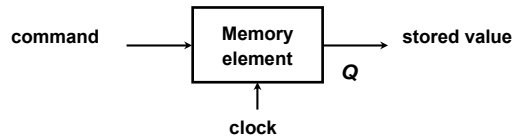
Command (at time t)	$Q(t)$	$Q(t+1)$
Set	X	1
Reset	X	0
Memorise / No Change	0	0
	1	1

- جدول مشخصه:
(Characteristic table)

$Q(t)$: current state

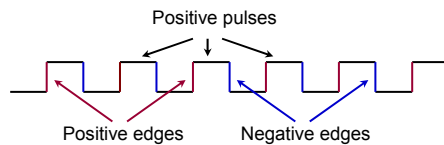
$Q(t+1)$ or Q^+ : next state

- عنصر حافظه با پالس ساعت: فلیپ-فلاپ ها عناصر حافظه می باشند که با اعمال سیگنال پالس ساعت حالت آنها تغییر می کند.



عناصر حافظه

- پالس ساعت به طور معمول یک پالس مربعی می باشد.



- دو روش برای تریگر یا تحریک (triggering/activation) فلیپ فلاپ وجود دارد:

❖ تحریک با سطح پالس (pulse-triggered):
ON = 1, OFF = 0

❖ تحریک با لبه پالس (edge-triggered):

positive edge-triggered (ON = from 0 to 1; OFF = other time)

negative edge-triggered (ON = from 1 to 0; OFF = other time)

S-R Latch

- خروجی های مکمل: Q و Q'
- وقتی $Q=High$ است لچ در حالت **SET** است.
- وقتی $Q=Low$ است لچ در حالت **RESET** است.
- برای **S-R latch** با ورودی های **Active-HIGH** که به عنوان لچ با گیت NOR نیز شناخته می شود داریم:

$R=HIGH$ (and $S=LOW$) \Rightarrow RESET state
 $S=HIGH$ (and $R=LOW$) \Rightarrow SET state
 both inputs LOW \Rightarrow no change
 both inputs HIGH \Rightarrow Q and Q' both LOW (invalid)!

- برای **S'-R' latch** با ورودی های **Active-LOW** که به عنوان لچ با گیت NAND نیز شناخته میشود داریم:

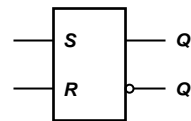
$R'=LOW$ (and $S'=HIGH$) \Rightarrow RESET state
 $S'=LOW$ (and $R'=HIGH$) \Rightarrow SET state
 both inputs HIGH \Rightarrow no change
 both inputs LOW \Rightarrow Q and Q' both HIGH (invalid)!

- عیب S-R latch : دارای شرایط نامعتبر بوره و باید از این حالت پرهیز کرد.

S-R Latch

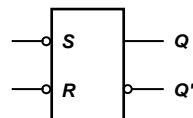
- جدول مشخصه (Characteristics table) برای **S-R latch** با ورودی های **Active-HIGH**

S	R	Q	Q'	
0	0	NC	NC	No change. Latch remained in present state.
1	0	1	0	Latch SET.
0	1	0	1	Latch RESET.
1	1	0	0	Invalid condition.



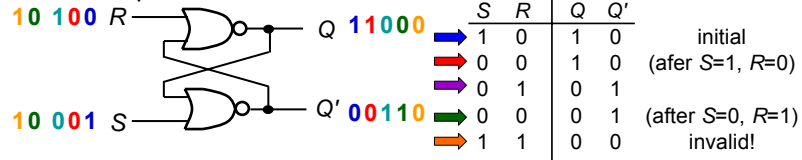
- جدول مشخصه (Characteristics table) برای **S'-R' latch** با ورودی های **Active-LOW**

S'	R'	Q	Q'	
1	1	NC	NC	No change. Latch remained in present state.
0	1	1	0	Latch SET.
1	0	0	1	Latch RESET.
0	0	1	1	Invalid condition.

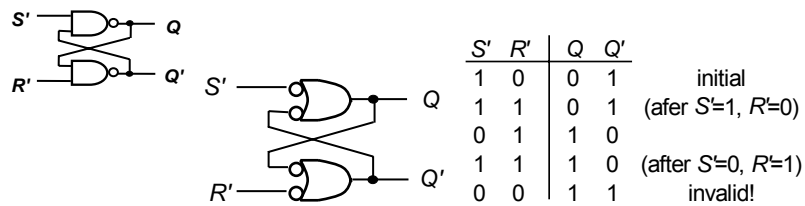


S-R Latch

- Active-HIGH input S-R latch

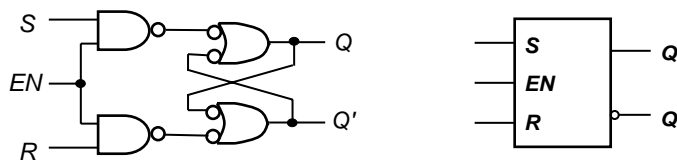


- Active-LOW input S'-R' latch



Gated S-R Latch

- S-R latch + enable input (EN) and 2 NAND gates → gated S-R latch.



- در صورت نیاز خروجی فقط وقتی تغییر می یابد که EN=HIGH باشد.

- جدول مشخصه:

EN=1			
Q(t)	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

S	R	Q(t+1)	
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	indeterminate	

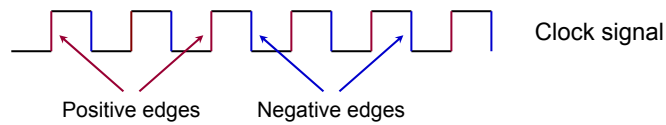
معادله مشخصه:

$$Q(t+1) = S + R' \cdot Q$$

$$S \cdot R = 0$$

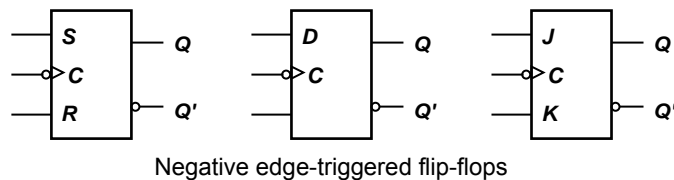
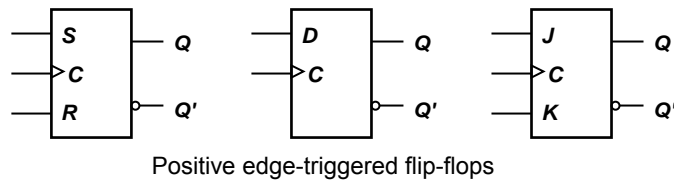
فلیپ فلاپ های تحریک با لبه پالس (Edge-Triggered Flip-flops)

- **فلیپ فلاپ:** مدار بای استابل سنکرون می باشد.
- حالت خروجی در نقاط مشخصی از ورودی محرک (پالس ساعت) تغییر می کند.
- تغییر یا در لبه مثبت یا بالا رونده (Rising edge) صورت می گیرد یا در لبه منفی یا پایین رونده (Falling edge).



فلیپ فلاپ های تحریک با لبه پالس (Edge-Triggered Flip-flops)

- فلیپ فلاپ های حساس به لبه S-R ، D و J-K :
- دقت کنید که علامت ">" نشان دهنده پالس ساعت می باشد.



S-R Flip-flop

جدول مشخصه فلیپ فلاپ S-R حساس به لبه مثبت:

S	R	CLK	Q(t+1)	Comments
0	0	X	Q(t)	No change
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	?	Invalid

X = don't care

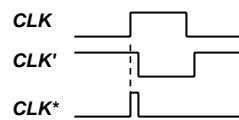
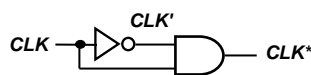
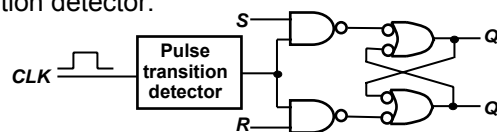
↑ = clock transition LOW to HIGH

این فلیپ فلاپ از سه قسمت تشکیل شده است:

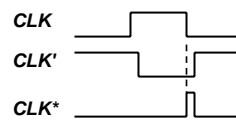
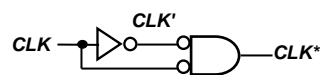
- لچ پایه با گیت NAND
- مدار هدایت پالس ساعت (*Pulse-Steering Circuit*)
- آشکار ساز انتقال یا لبه پالس (*Edge Detector Circuit*): این قسمت محل لبه بالا رونده یا پایین رونده را مشخص کرده و در این نقطه یک پالس با عرض بسیار کم (*Spike*) تولید می کند.

S-R Flip-flop

The pulse transition detector.



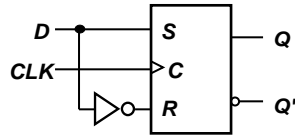
Positive-going transition
(rising edge)



Negative-going transition
(falling edge)

D Flip-flop

▪ فلیپ فلاپ D : نوع تک ورودی فلیپ فلاپ S-R به صورت زیر می باشد:



D	CLK	Q(t+1)	Comments
1	↑	1	Set
0	↑	0	Reset

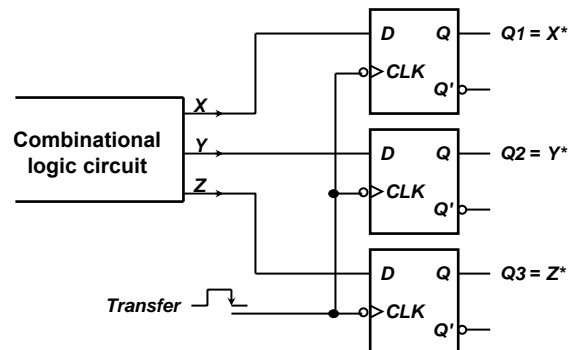
↑ = clock transition LOW to HIGH

در لبه های پالس ساعت ورودی D (Data) به خروجی Q منتقل می گردد.

D Flip-flop

▪ کاربرد: انتقال داده موازی (*Parallel data transfer*)

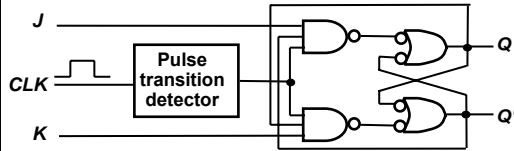
خروجی های مدار منطقی برای ذخیره به فلیپ فلاپ های D منتقل شده است.



* After occurrence of negative-going transition

J-K Flip-flop

▪ J-K flip-flop.



▪ Q و Q' بصورت فید بک به مدار هدایت پالس متصل هستند.

▪ این نوع فلیپ فلاپ حالت نامعتبر نداشته و دارای حالتی بنام Toggle است.

▪ جدول مشخصه:

J	K	CLK	Q(t+1)	Comments
0	0	↑	Q(t)	No change
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	Q(t)'	Toggle

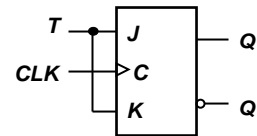
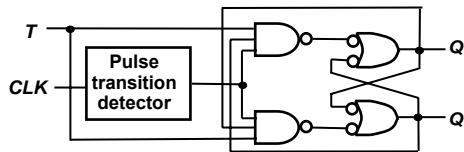
Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

▪ معادله مشخصه:

$$Q(t+1) = J \cdot Q' + K' \cdot Q$$

T Flip-flop

▪ T flip-flop : نوع تک ورودی فلیپ فلاپ J-K بوده و با اتصال دو ورودی J و K به هم بدست می آید.



▪ جدول مشخصه:

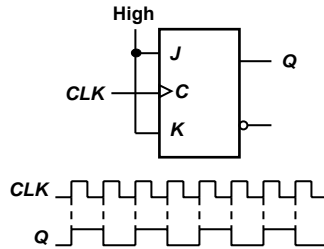
T	CLK	Q(t+1)	Comments
0	↑	Q(t)	No change
1	↑	Q(t)'	Toggle

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

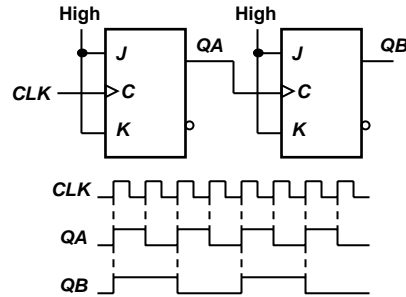
$$Q(t+1) = T \cdot Q' + T' \cdot Q$$

T Flip-flop

■ کاربرد: تقسیم فرکانس (Frequency division) و شمارنده (Counter)



Divide clock frequency by 2.



Divide clock frequency by 4.

ورودی های غیر همزمان (Asynchronous Inputs)

■ ورودی های S-R ، J-K ، D و T ورودی های همزمان (synchronous inputs) نامیده می شوند، چراکه داده اعمالی به این ورودی ها فقط در لبه های پالس ساعت (بالا یا پایین رونده) به خروجی فلیپ فلاپ منتقل می شود.

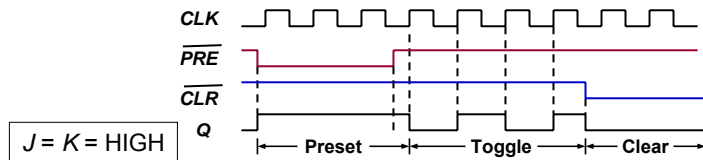
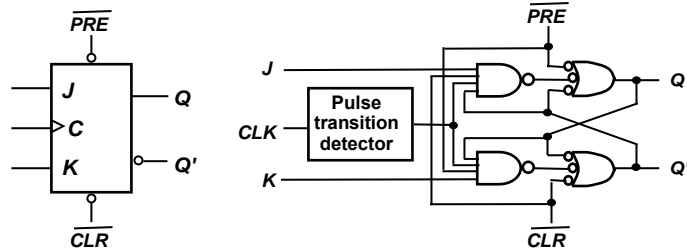
■ ورودی های غیر همزمان حالت فلیپ فلاپ را مستقل از پالس ساعت ورودی تغییر می دهند.

مانند: *preset (PRE)* یا *direct set (SD)* و *clear (CLR)* یا *direct reset (RD)*

- وقتی ورودی PRE فعال باشد خروجی Q فوراً SET خواهد شد.
- وقتی ورودی CLR فعال باشد خروجی Q فوراً RESET خواهد شد.
- وقتی هر دو ورودی غیر همزمان غیر فعال باشند، فلیپ فلاپ در حالت عادی و بر اساس جدول مشخصه و ورودی های آن کار خواهد کرد.

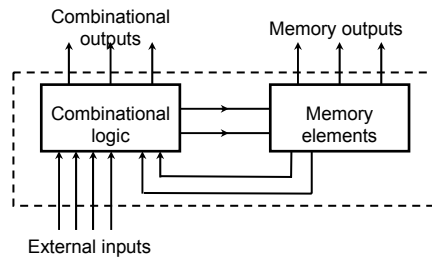
ورودی های غیر همزمان (Asynchronous Inputs)

- فلیپ فلاپ J-K با ورودی های Preset و Clear به صورت Active-LOW



طراحی مدار های منطقی ترتیبی با فلیپ فلاپ (Sequential Logic Design with Flip-flops)

- مدار های ترتیبی دارای یک بعد اضافی هستند: زمان
- خروجی های مدار های ترکیبی فقط به ورودی های زمان حال بستگی دارند.
- خروجی های مدار های ترتیبی به ورودی های قبلی نیز بستگی دارند.
- مدار های ترتیبی دارای قدرت طراحی بالاتری بوده و قادر به مدلسازی وضعیت هایی هستند که با مدار های ترکیبی امکان پذیر نیست.
- عناصر سازنده مدار های ترتیبی گیت ها و فلیپ فلاپ ها هستند.
- فلیپ فلاپ ها عناصر حافظه و گیت ها مدار ترکیبی را تشکیل می دهند.



جدول مشخصه فلیپ فلاپ ها

- هر فلیپ فلاپ دارای رفتار منحصر به فردی است. جدول مشخصه انواع فلیپ فلاپ ها در زیر آورده شده است.

S	R	Q(t+1)	Comments
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Unpredictable

SR Flip-flop

J	K	Q(t+1)	Comments
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q(t)'	Toggle

JK Flip-flop

D	Q(t+1)
0	0 Reset
1	1 Set

D Flip-flop

T	Q(t+1)
0	Q(t) No change
1	Q(t)' Toggle

T Flip-flop

تحلیل مدارهای ترتیبی (Sequential Circuit Analysis)

- با داشتن دیاگرام منطقی مدار ترتیبی، عملکرد آنرا با به دست آوردن جدول حالت (*state table*) و دیاگرام حالت (*state diagram*)، تحلیل می کنیم.
- برای بدست آوردن جدول حالت، معادلات حالت *state equations* برای ورودی های فلیپ فلاپ ها نوشته می شود. اگر مدار دارای خروجی غیر از وضعیت فلیپ فلاپ ها باشد، توابع خروجی *output functions* برای آنها نیز نوشته می شود.
- برای فلیپ فلاپ A، برای نمایش وضعیت فعلی و بعدی به ترتیب از $A(t)$ و $A(t+1)$ استفاده می گردد.
- همچنین برای سادگی می توان از A و A^+ به ترتیب برای حالت فعلی و بعدی استفاده کرد.

تحلیل مدارهای ترتیبی

■ مثال (فلیپ فلاپ D) :

در فلیپ فلاپ D بعد از اعمال پالس ساعت، حالت بعدی همان حالت فعلی ورودی D خواهد بود.

State equations:

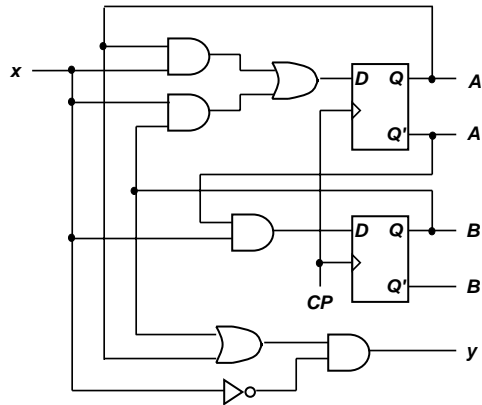
$$A^+ = A.x + B.x$$

$$B^+ = A'.x$$

Output function:

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

شکل (۱)



تحلیل مدارهای ترتیبی

■ با استفاده از معادلات حالت و تابع خروجی، جدول حالت به دست می آید، که شامل تمام حالت‌های ممکن فعلی و ورودی‌ها می باشد.

■ جدول حالت:

❖ شبیه جدول درستی است.

❖ در سمت چپ، ورودی‌ها و حالت فعلی فلیپ فلاپ‌ها نوشته می شود.

❖ در سمت راست، خروجی‌ها و حالت بعدی فلیپ فلاپ‌ها نوشته می شود.

■ با داشتن m فلیپ فلاپ و n ورودی، جدول حالت دارای 2^{m+n} سطر خواهد بود.

تحلیل مدارهای ترتیبی

■ جدول حالت برای شکل (۱):

State equations:

$$A^+ = A.x + B.x$$

$$B^+ = A'.x$$

Output function:

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

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

تحلیل مدارهای ترتیبی

■ فرم دیگری برای جدول حالت:

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

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
AB	A ⁺ B ⁺	A ⁺ B ⁺	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

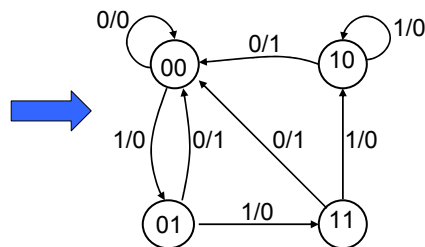
تحلیل مدارهای ترتیبی

- از روی جدول حالت می توان **دیاگرام حالت (state diagram)** را رسم کرد.
- **دیاگرام حالت:**
 - ❖ هر حالت با یک دایره مشخص می گردد.
 - ❖ هر پیکان (بین دو دایره) نشانگر یک تغییر حالت در مدار ترتیبی است (یک سطر در جدول حالت)
 - ❖ یک علامت به فرم alb روی هر پیکان نوشته می شود که a نشان دهنده ورودی و b نشان دهنده خروجی در تغییر حالت مورد نظر است.
 - ❖ هر ترکیب از فلیپ فلاپ ها نمایش دهنده یک حالت است. از اینرو با داشتن m فلیپ فلاپ، حداکثر 2^m حالت خواهیم داشت.

تحلیل مدارهای ترتیبی

- **دیاگرام حالت** مربوط به شکل (۱):

Present State AB	Next State		Output	
	$x=0$ A^+B^+	$x=1$ A^+B^+	$x=0$ y	$x=1$ y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0



توابع ورودی فلیپ فلاپ (Flip-flop Input Functions)

■ خروجی های مدار ترتیبی توابعی از حالت فعلی فلیپ فلاپ ها و ورودی ها هستند. این خروجی ها بطور جبری توسط توابع خروجی مدار (*circuit output functions*) توصیف می شوند.

❖ در شکل (۱) : $y = (A + B).x'$

■ قسمتی از مدار که ورودی فلیپ فلاپ ها را تولید می کند، بطور جبری با توابع ورودی فلیپ فلاپ (*flip-flop input functions*) توصیف می شود.

■ توابع ورودی برای تعیین حالت بعدی بکار می روند.

■ با استفاده از توابع ورودی و جدول مشخصه فلیپ فلاپ، حالت بعدی فلیپ فلاپ تعیین می گردد.

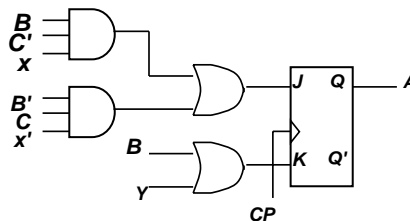
توابع ورودی فلیپ فلاپ (Flip-flop Input Functions)

■ مثال: مدار ترتیبی با فلیپ فلاپ J-K :

■ از دو حرف برای نمایش ورودی هر فلیپ فلاپ استفاده می گردد: حرف اول نشان دهنده نوع ورودی فلیپ فلاپ است (J و K برای فلیپ فلاپ JK، S و R برای فلیپ فلاپ SR، D برای فلیپ فلاپ D و T برای فلیپ فلاپ T). حرف دوم نشان دهنده اسم فلیپ فلاپ است.

$$JA = B.C'.x + B'.C.x'$$

$$KA = B + y$$



Analysis: Example #2

$$JA = B$$

$$JB = x'$$

$$KA = B.x'$$

$$KB = A'.x + A.x' = A \oplus x$$

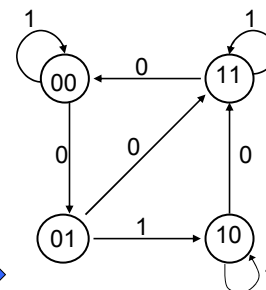
▪ تعیین جدول حالت با استفاده از توابع فوق و جدول مشخصه فلیپ فلاپ :

J	K	Q(t+1)	Comments	Present state		Input	Next state		Flip-flop inputs			
				A	B		A ⁺	B ⁺	JA	KA	JB	KB
0	0	Q(t)	No change	0	0	0	0	1	0	0	1	0
0	1	0	Reset	0	0	1	0	0	0	0	0	1
1	0	1	Set	0	1	0	1	1	1	1	1	0
1	1	Q(t)'	Toggle	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

Analysis: Example #2

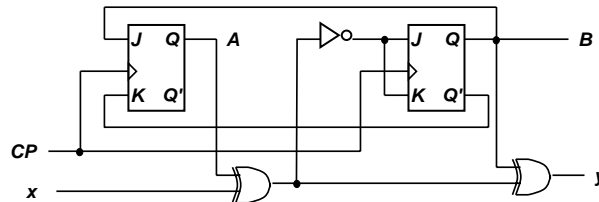
▪ رسم دیاگرام حالت از روی جدول حالت:

Present state		Input	Next state		Flip-flop inputs			
A	B		A ⁺	B ⁺	JA	KA	JB	KB
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



Analysis: Example #3

جدول حالت و دیاگرام حالت مدار شکل (۳) را به دست آورید:



شکل (۳)

توابع ورودی فلیپ فلاپ:

$$JA = B \quad JB = KB = (A \oplus x)' = A.x + A'.x'$$

$$KA = B'$$

Analysis: Example #3

J	K	Q(t+1)	Comments
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q(t)'	Toggle

$$JA = B \quad KA = B'$$

$$JB = KB = (A \oplus x)' = A.x + A'.x'$$

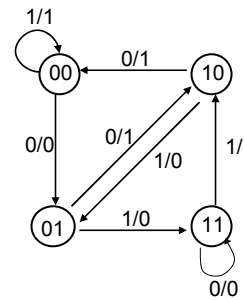
جدول حالت:

Present state		Input	Next state		Output	Flip-flop inputs			
A	B		A*	B*		JA	KA	JB	KB
0	0	0	0	1	0	1	1	1	
0	0	1	0	0	0	1	0	0	
0	1	0	1	0	1	0	1	1	
0	1	1	1	1	1	0	0	0	
1	0	0	0	0	0	1	0	0	
1	0	1	0	1	0	1	1	1	
1	1	0	1	1	0	0	0	0	
1	1	1	1	0	1	0	1	1	

Analysis: Example #3

▪ دیاگرام حالت:

Present state		Input x	Next state		Output y	Flip-flop inputs			
A	B		A ⁺	B ⁺		JA	KA	JB	KB
0	0	0	0	1	0	1	1	1	
0	0	1	0	0	0	1	0	0	
0	1	0	1	0	1	0	1	1	
0	1	1	1	1	1	0	0	0	
1	0	0	0	0	0	1	0	0	
1	0	1	0	1	0	1	1	1	
1	1	0	1	1	1	0	0	0	
1	1	1	1	0	1	0	1	1	



جدول تحریک فلیپ فلاپ (Flip-flop Excitation Tables)

- **تحلیل (Analysis):** شروع از دیاگرام منطقی ← بدست آوردن جدول حالت یا دیاگرام حالت.
- **طراحی (Design):** شروع از یک سری مشخصه (به فرم معادلات حالت، جدول حالت یا دیاگرام حالت) ← بدست آوردن دیاگرام منطقی (logic circuit).
- جدول مشخصه (Characteristic tables) در تحلیل استفاده می گردند.
- جدول تحریک (Excitation tables) در طراحی استفاده می گردند.

جداول تحریک فلیپ فلاپ

- **جداول تحریک:** با داشتن انتقال از حالت فعلی به حالت بعدی، ورودی فلیپ فلاپ را تعیین می کند. این جداول را می توان از روی جداول مشخصه به دست آورد.

J	K	Q(t+1)	Comments
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q(t)'	Toggle

جدول مشخصه JK

S	R	Q(t+1)	Comments
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Unpredictable

جدول مشخصه SR

Q	Q ⁺	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

جدول تحریک JK

Q	Q ⁺	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

جدول تحریک SR

جداول تحریک فلیپ فلاپ

- **جداول تحریک:** با داشتن انتقال از حالت فعلی به حالت بعدی، ورودی فلیپ فلاپ را تعیین می کند. این جداول را می توان از روی جداول مشخصه به دست آورد.

D	Q(t+1)	
0	0	Reset
1	1	Set

جدول مشخصه D

T	Q(t+1)	
0	Q(t)	No change
1	Q(t)'	Toggle

جدول مشخصه T

Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

جدول تحریک D

Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

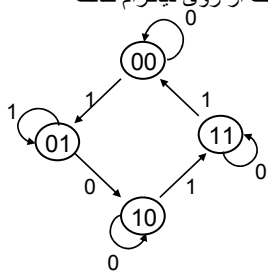
جدول تحریک T

طراحی مدارهای ترتیبی (Sequential Circuit Design)

- Design procedure:
 - ❖ Start with circuit specifications – description of circuit behaviour.
 - ❖ Derive the state table.
 - ❖ Perform state reduction if necessary.
 - ❖ Perform state assignment.
 - ❖ Determine number of flip-flops and label them.
 - ❖ Choose the type of flip-flop to be used.
 - ❖ Derive circuit excitation and output tables from the state table.
 - ❖ Derive circuit output functions and flip-flop input functions.
 - ❖ Draw the logic diagram.

Design: Example #1

- مدار ترتیبی متناظر با دیاگرام حالت زیر را با استفاده از فلیپ فلاپ JK طراحی کنید:
- تعیین تعداد فلیپ فلاپ ها و اسم گذاری آنها، بدست آوردن جدول حالت از روی دیاگرام حالت



Present State	Next State	
	x=0	x=1
AB	A^+B^+	A^+B^-
00	00	01
01	10	01
10	10	11
11	11	00

- تعیین ورودی فلیپ فلاپ ها از روی جدول تحریک

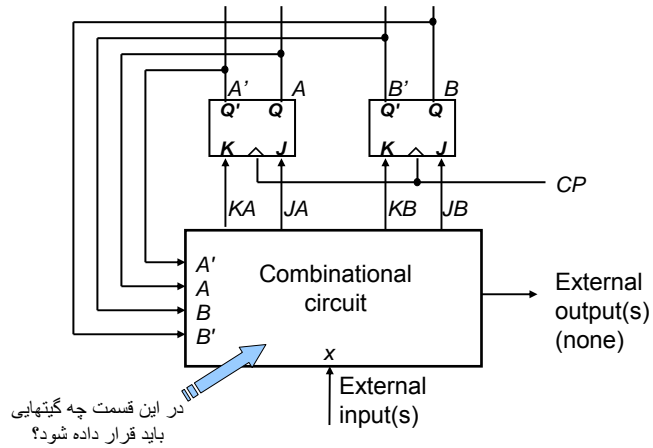
Q	Q ⁺	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

جدول تحریک JK

Present state		Input x	Next state		Flip-flop inputs			
A	B		A ⁺	B ⁺	JA	KA	JB	KB
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

Design: Example #1

■ بلوک دیاگرام:

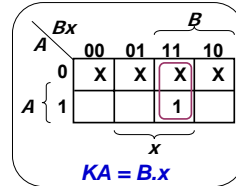
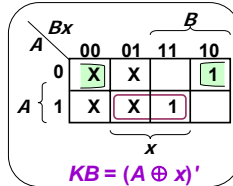
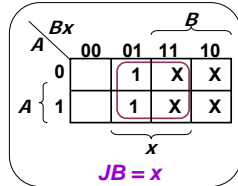
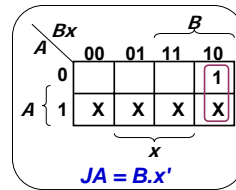


در این قسمت چه گیت‌هایی
باید قرار داده شود؟

Design: Example #1

■ ساده سازی توابع ورودی فلیپ فلاپ (معمولاً با جداول کارنو)

Present state		Input x	Next state		Flip-flop inputs			
A	B		A'	B'	JA	KA	JB	KB
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



Design: Example #1

توابع ورودی فلیپ فلاپ:

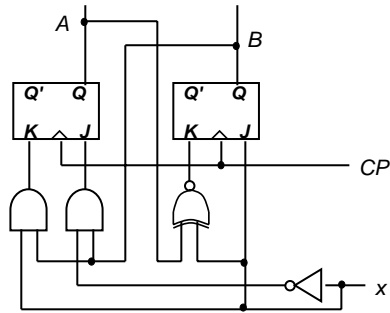
$$JA = B.x'$$

$$KA = B.x$$

$$JB = x$$

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

دیاگرام منطقی:



Design: Example #2

با استفاده از فلیپ فلاپ D ، مداری ترتیبی متناظر با جدول حالت زیر طراحی کنید:

در فلیپ فلاپ D حالت بعدی با ورودی فلیپ فلاپ برابر است و نیازی به جدول تحریک و ستون جداگانه برای ورودی فلیپ فلاپ نیست.

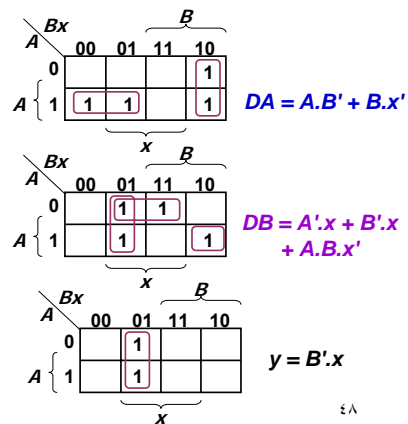
به دست آوردن توابع ورودی فلیپ فلاپ و خروجی مدار:

Present state		Input	Next state		Output
A	B		x	A ⁺	
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0

$$DA(A, B, x) = \Sigma m(2,4,5,6)$$

$$DB(A, B, x) = \Sigma m(1,3,5,6)$$

$$y(A, B, x) = \Sigma m(1,5)$$



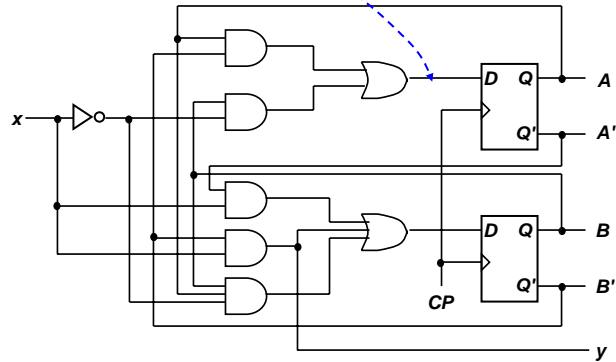
Design: Example #2

■ رسم دیاگرام منطقی از روی معادلات به دست آمده:

$$DA = A.B' + B.x'$$

$$DB = A'.x + B'.x + A.B.x'$$

$$y = B'.x$$



■ تمرین: مدار را با فلیپ فلاپ JK طراحی کنید.

Design: Example #3

■ طراحی با حالت های بلا استفاده : با استفاده از فلیپ فلاپ SR مدار متناظر با جدول حالت داده شده را طراحی کنید:

Present state		Input	Next state			Flip-flop inputs						Output	
A	B	C	x	A*	B*	C*	SA	RA	SB	RB	SC	RC	y
0	0	1	0	0	0	1	0	X	0	X	X	0	0
0	0	1	1	0	1	0	0	X	1	0	0	1	0
0	1	0	0	0	1	1	0	X	X	0	1	0	0
0	1	0	1	1	0	0	1	0	0	1	0	X	0
0	1	1	0	0	0	1	0	X	0	1	X	0	0
0	1	1	1	1	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	1	X	0	0	X	1	0	0
1	0	0	1	1	0	0	X	0	0	X	0	X	1
1	0	1	0	0	0	1	0	1	0	X	X	0	0
1	0	1	1	1	0	0	X	0	0	X	0	1	1

جدول تحریک SR

با داشتن این ستون ها و از روی جدول تحریک

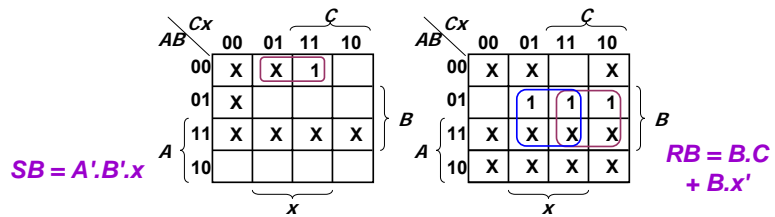
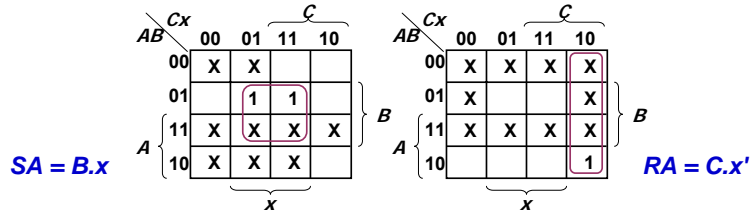
این ستون ها را به دست می آوریم.

Unused state 000:

0	0	0	0	X	X	X	X	X	X	X	X	X	X
0	0	0	1	X	X	X	X	X	X	X	X	X	X

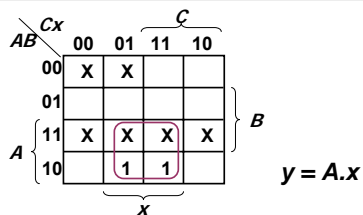
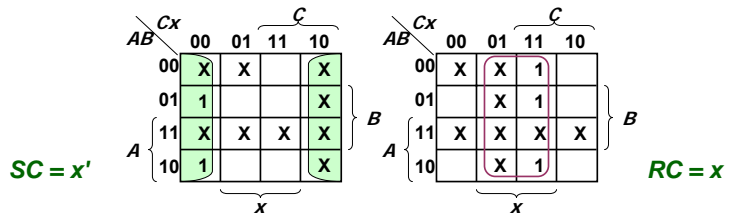
Design: Example #3

توابع ورودی فلیپ فلاپ ها و خروجی مدار را از جدول حالت به دست می آوریم:



Design: Example #3

توابع ورودی فلیپ فلاپ ها و خروجی مدار را از جدول حالت به دست می آوریم:



مدار های خود اصلاح

- در Design Example #3 به معادلات زیر رسیدیم:

$$\begin{array}{lll}
 SA = B.x & SB = A'.B'.x & SC = x' \\
 RA = C.x' & RB = B.C + B.x' & RC = x \\
 y = A.x & &
 \end{array}$$

- در جدول حالت تمام X ها را با مقادیرشان پر می کنیم که شامل تمام حالت های بلا استفاده 000، 110 و 111 نیز خواهد بود.

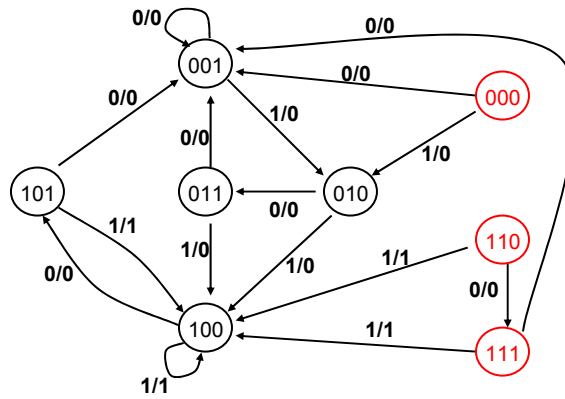
مدار های خود اصلاح

- جدول حالت بر اساس معادلات به دست آمده:

	Present state			Input	Next state			Flip-flop inputs						Output
	A	B	C		A*	B*	C*	SA	RA	SB	RB	SC	RC	
	0	0	0	0	0	0	1	0	0	0	0	1	0	0
	0	0	0	1	0	1	0	0	0	1	0	0	1	0
	0	0	1	0	0	0	1	0	1	0	0	1	0	0
	0	0	1	1	0	1	0	0	0	1	0	0	1	0
	0	1	0	0	0	1	1	0	0	0	0	0	1	0
	0	1	0	1	1	0	0	1	0	0	1	0	1	0
	0	1	1	0	0	0	1	0	1	0	1	1	0	0
	0	1	1	1	1	0	0	1	0	0	1	0	1	0
	1	0	0	0	1	0	1	0	0	0	0	1	0	0
	1	0	0	1	1	0	0	0	0	0	0	0	1	1
	1	0	1	0	0	0	1	0	1	0	0	1	0	0
	1	0	1	1	1	0	0	0	0	0	0	0	1	1
	1	1	0	0	1	1	1	0	0	0	0	1	0	0
	1	1	0	1	1	0	0	1	0	0	1	0	1	1
	1	1	1	0	0	0	1	0	1	0	1	1	0	0
	1	1	1	1	1	0	0	1	0	0	1	0	1	1

مدار های خود اصلاح

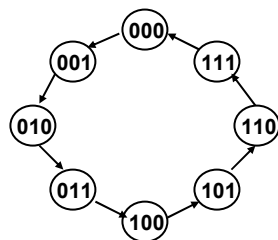
- رسم دیاگرام حالت بر اساس جدول حالت به دست آمده:



- حلقه ای بین حالت های نا معتبر وجود نداشته و مدار خود اصلاح است.

طراحی شمارنده های همزمان (Design of Synchronous Counters)

- شمارنده (Counter) : مداری ترتیبی است که بین دنباله ای از حالت ها جرخش می کند.
- شمارنده باینری (Binary counter) : از یک دنباله باینری پیروی می کند. یک شمارنده باینری n بیتی (با n عدد فلیپ فلاپ) از 0 تا $2^n - 1$ شمارش می کند.
- مثال ۱: شمارنده باینری ۳ بیتی (با فلیپ فلاپ T) :



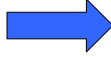
Present state			Next state		
A_2	A_1	A_0	A_2^+	A_1^+	A_0^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

طراحی شمارنده های همزمان

شمارنده باینری ۳ بیتی (ادامه):

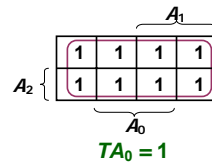
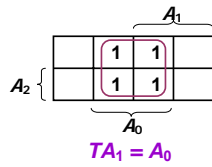
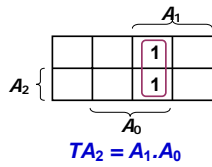
Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

جدول تحریک T



Present state			Next state			Flip-flop inputs		
A ₂	A ₁	A ₀	A ₂ ⁺	A ₁ ⁺	A ₀ ⁺	TA ₂	TA ₁	TA ₀
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	0	1
1	1	1	0	0	0	1	1	1

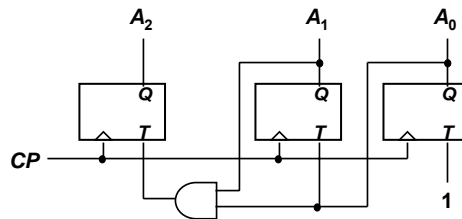
ورودی فلیپ فلاپ ها



طراحی شمارنده های همزمان

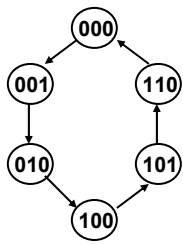
شمارنده باینری ۳ بیتی (ادامه):

$$TA_2 = A_1 \cdot A_0 \quad TA_1 = A_0 \quad TA_0 = 1$$



طراحی شمارنده های همزمان

■ مثال ۲: شمارنده با ترتیب غیر باینری (Counter with non-binary sequence):



Present state			Next state			Flip-flop inputs					
A	B	C	A ⁺	B ⁺	C ⁺	JA	KA	JB	KB	JC	KC
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

$$JA = B$$

$$KA = B$$

$$JB = C$$

$$KB = 1$$

$$JC = B'$$

$$KC = 1$$

طراحی شمارنده های همزمان

شمارنده با ترتیب غیر باینری (ادامه):

$$JA = B$$

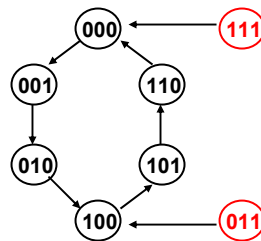
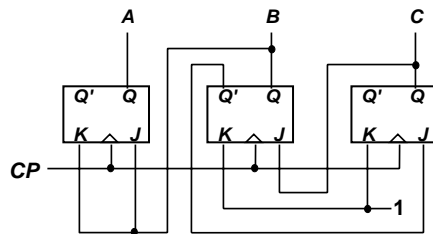
$$KA = B$$

$$JB = C$$

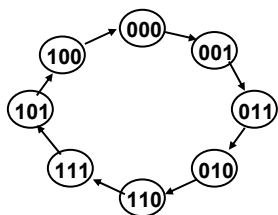
$$KB = 1$$

$$JC = B'$$

$$KC = 1$$



تمرین



تمرین: یک شمارنده گری سنکرون سه بیتی با فلیپ فلاپ JK طراحی کنید:

تمرین: یک شمارنده BCD سنکرون با فلیپ فلاپ JK طراحی کنید:

تمرین: یک شمارنده بالا-پایین شمار (Up-Down Counter) سه بیتی با فلیپ فلاپ T طراحی کنید:

Clock pulse	Up	Q_2	Q_1	Q_0	Down
0	↗	0	0	0	↘
1	↗	0	0	1	↘
2	↗	0	1	0	↘
3	↗	0	1	1	↘
4	↗	1	0	0	↘
5	↗	1	0	1	↘
6	↗	1	1	0	↘
7	↗	1	1	1	↘

شمارنده غیر همزمان

Asynchronous (Ripple) Counters

مثال: 3-bit ripple binary counter

