

مدارهای ترتیبی :**مدارهای منطقی ترتیبی :**

خروجی علاوه بر اینکه به ورودیهای مدار بستگی دارد به خروجیهای قبلی مدار (و در نتیجه به ورودیهای قبلی) بستگی دارد . پس می توان گفت مدار ترتیبی حافظه دار است .

انواع مدارهای ترتیبی :

۱- سنکرون یا همزمان : همزمان با سیگنالی بنام کلاک تغییرات در خروجی مدار و در اثر ورودی صورت می گیرد .

۲- آسنکرون یا غیر همزمان : تغییرات خروجی بدون سیگنال با تغییر ورودی امکان پذیر است .

موضوعات :

۱- آشنایی با انواع فلیپ فلاپها

۲- آشنایی با برخی مدارهای ترتیبی

۳- طراحی مدارهای منطقی ترتیبی

فلیپ فلاپها (Flip-Flop یا FF) :

مدار منطقی است که می تواند يك بیت اطلاعات را برای ما نگه دارد .

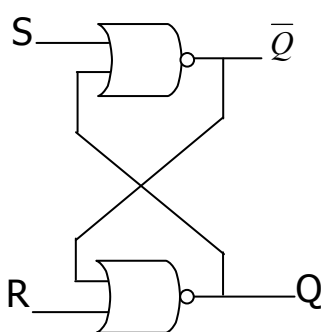
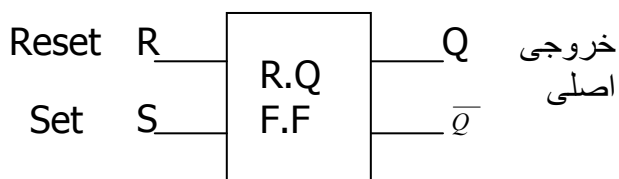
انواع FF :

۱- RS-FF

۲- D-FF

۳- JK-FF

T-FF -۴

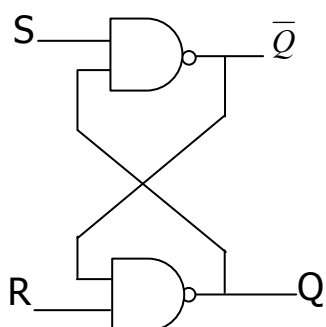
: **Rs-FF**

R	S	Q	\bar{Q}
۱	۰	۰	۱
۰	۰	۰	۱
۰	۱	۱	۰
۰	۰	۱	۰
۱	۱	۰	۰

یا FF Set - Reset

$$\bar{Q} = \text{NOT } Q$$

از خروجی Q یک Feed back یا باز خورد زدیم به ورودی NOR



R	S	Q_{n-1}	\bar{Q}_{n+1}	وضعیت (عمل)
۱	۱	Q_n	\bar{Q}_n	وضعیت قبلی (Hold)
۰	۱	۰	۱	Reset
۱	۱	۰	۱	Hold
۱	۰	۱	۰	Set
۱	۱	۱	۰	Hold
۰	۰	۱	۱	غیر مجاز

نکته : هرگاه ورودی را صفر کنیم تاثیر منفی می گذارد.

نکته : خروجی وضعیت قبلی به صورت یک ورودی به طور منفی منظور می گردد.

جدول صحت (برای ساختار NOR) :

R	S	Q_n	Q_{n-1}	\bar{Q}_{n+1}
•	•	•	•	۱
•	•	۱	۱	•
•	۱	•	۱	•
•	۱	۱	۱	•
۱	•	•	•	۱
۱	•	۱	•	۱
۱	۱	•	*	Don't care
۱	۱	۱	*	Don't care

RQn	S	Q_{n-1}
•	۱	•
•	۱	•
•	*	*
•	*	*

RQn	S	\bar{Q}_{n-1}
۱	•	۱
۱	•	۱
*	*	*
*	*	*

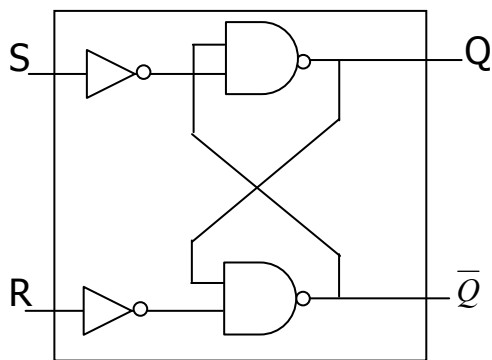
$$Q_{n+1} = S + R'Q_n$$

$$Rs = 0$$

$$\bar{Q}_{n+1} = R + SQ_n$$

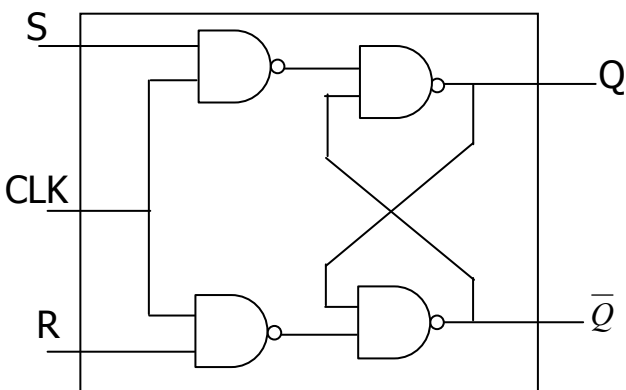
S,R با هم ترکیب نمی شود .

High Active : يك موثر است .



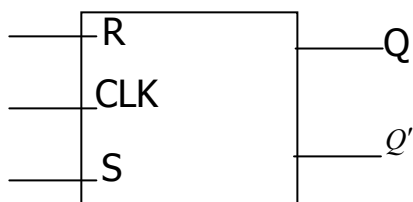
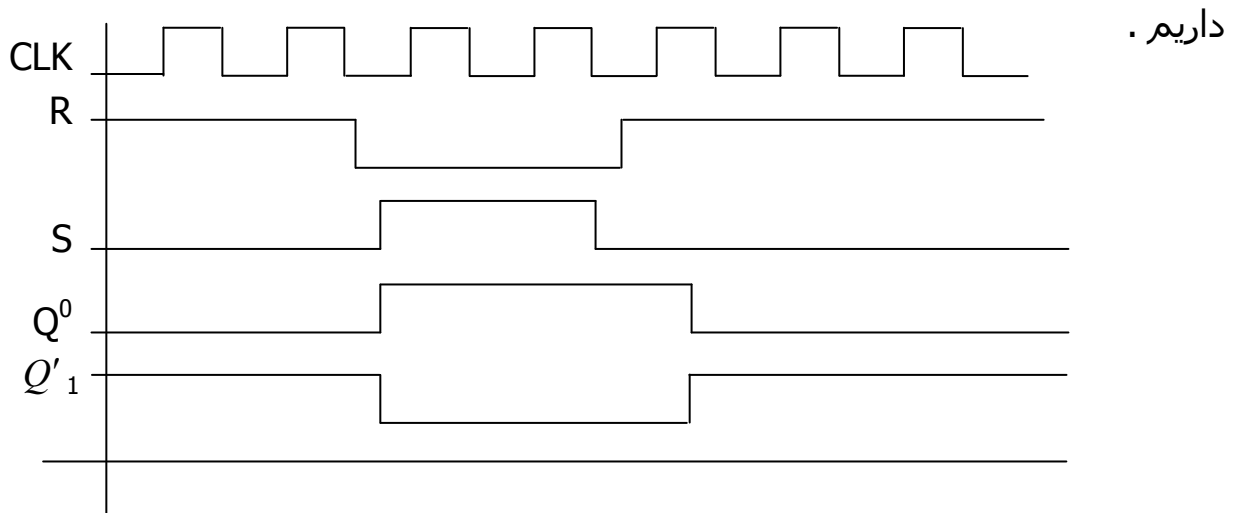
S	R	Q	Q'
•	•	Hold	
۱	•	۱	•
•	۱	•	۱
۱	۱	غير مجاز	

فلیپ فلاپ با کلاک :



وضعیت	Q'_{n+1}	Q_n	S	R	CLK
Hold	Q'_n	Q_n	*	*	.
Set	0	1	1	0	1
Reset	1	0	0	1	1

نکته: Hold: زمانی خروجی تغییر می‌کند که Clock نباید در غیر این صورت Hold یعنی باید غیر مجاز شود تا بعد R و S را بررسی کنیم در غیر این صورت R و S هر چه باشند Hold

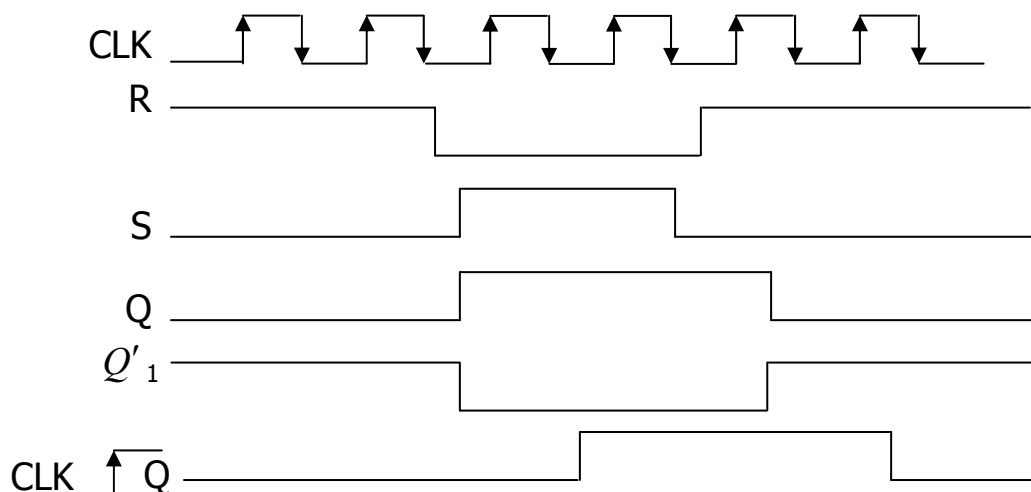


ورودی کلاک حساس به سطح یا FF یا حساس به سطح کلاک

F.F حساس به لبه کلاک : مدار داخلی از کتاب مانول ، زمانی خروجی تغییر می‌کند که به لبه کلاک نباید .

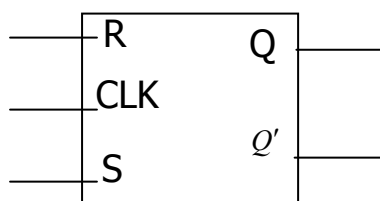


دو حالت می تواند پیش آید : حساس به لبه مثبت کلاک (بالا رونده ، یا حساس به لبه منفی کلاک (پایین رونده)



نکته : فقط در لحظه لبه ورودی های R,S را بررسی می کند و در غیر از آن هیچ کار نمی کند (Hold) حتی برای پایین رونده نیز هیچ کاری نمی کند .
 نکته : F.F حساس به لبه در مدارهای ترتیبی سنکرون استفاده می شود و F.F های حساس به سطح در مدارهای ترتیبی آسنکرون استفاده می شود .

نمایش سمبلیک :



ترسیم جدول :

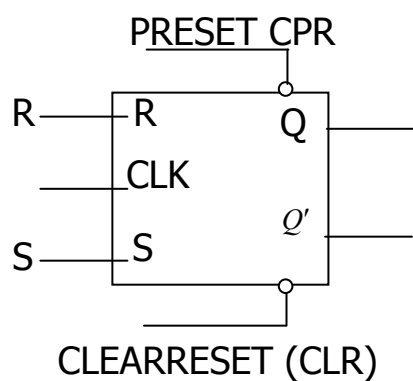
CLK	R	S	Q_{n-1}	Q'_{n+1}
.	*	*	Q_n	Q'_n
۱	.	۱	۱	.
۱	۱	.	.	۱
۱	۱	.	.	۱
۱	۱	۱	۱	۱

نکته : کلاک يك سيگنال

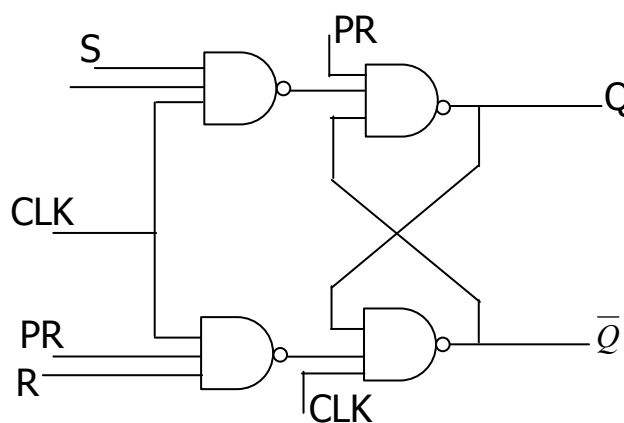
پريوديک مي باشد.

نکته : خروجي به خروجي هاي قبلي و به ورودي ها و هم به سيگنال زمانبندي

بستگي دارد .



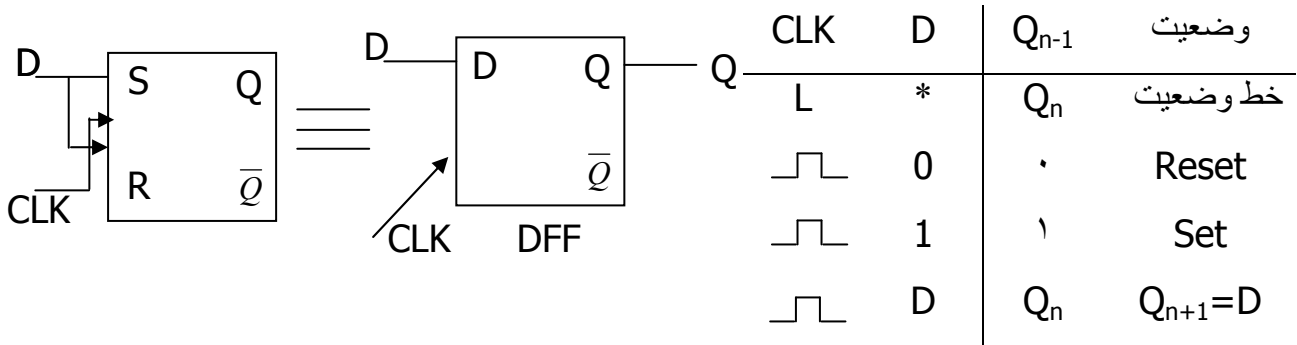
توسعه فلیپ فلاپ با کلاک :



CLK	PR	CLK	R	S	Q_{n-1}	Q'_{n+1}
*	.	۱	*	*	۱	.
*	۱	.	*	*	.	۱
*	.	.	*	*	۱	۱
.	۱	۱	*	*	Q_n	Q'_n
	۱	۱	.	.	Q_n	Q'_n
	۱	۱	.	۱		
	۱	۱	۱	.		
	۱	۱	۱	۱		

نکته : ورودی آسنکرون PR (CLR) بدون در نظر کلاک (زمانبندی) یا به صورت غیر همزمان روی خروجی تاثیر می گذارد .

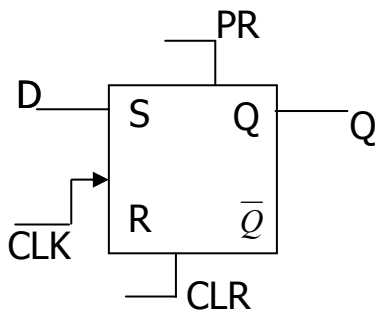
D-FF : کاربرد در طراحی مدارهای ترتیبی و خصوصاً در رجیسترها :



مزایا :

۱- برای Set و Reset و ... از یک ورودی استفاده می کنیم .

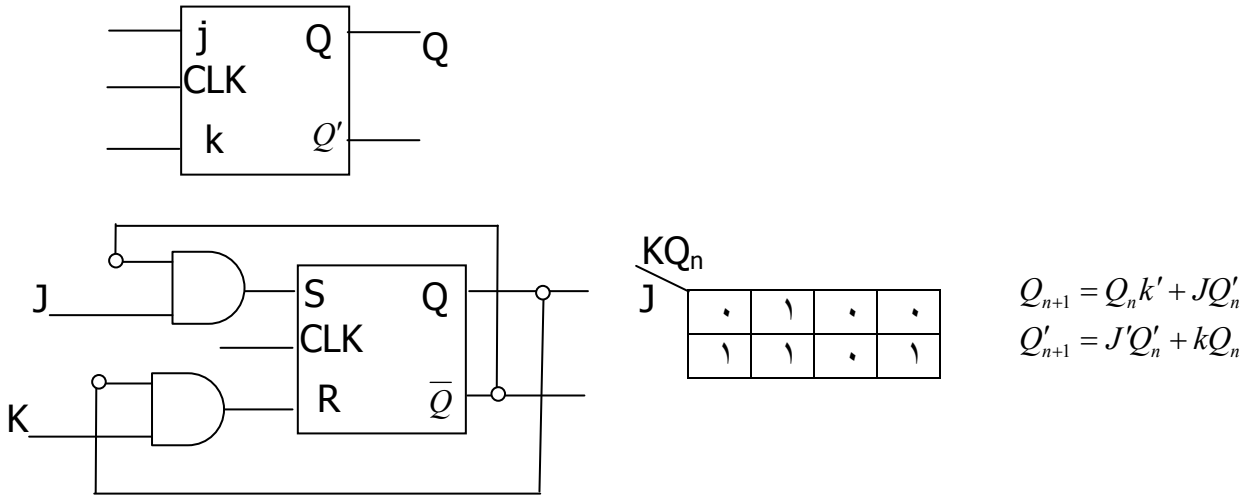
۲- حالت غیر مجاز را از بین برده است .



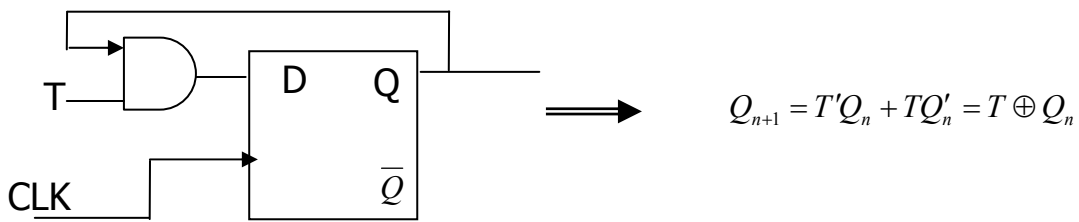
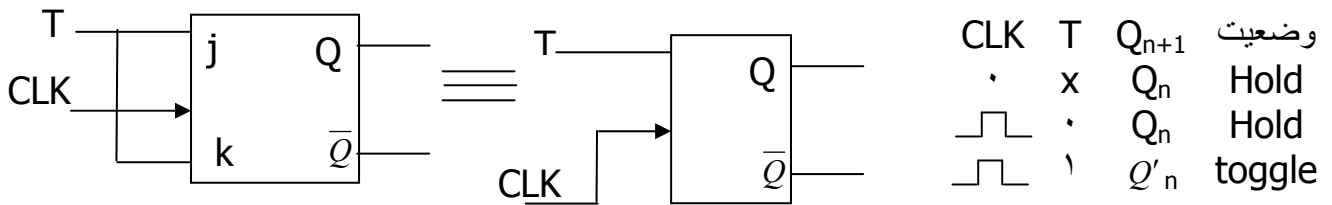
PR	CLK	CLD	D	
۰	۱	*	*	۱
۱	۰	*	*	۰
۰	۰	*	*	غیر مجاز *
۱	۱	⏏	D	D
۱	۱	۱	*	Q_n

Jk-ff : کاربرد در طراحی مدارهای ترتیبی و مخصوصاً در شمارنده ها .

PR	CLK	CLD	Q_{n+1}	Q'_{n-1}	وضعیت
L	*	*	Q_n	Q'_n	Hold
⏏	1	0	1	0	Set
⏏	0	1	0	1	Reset
⏏	0	0	Q_n	Q'_n	Hold
⏏	1	1	Q'_n	Q_n	Toggle (NOT)



T-FF : JK-FF ای است که Z و K آن به هم متصل است. کاربرد و شمارنده ها .

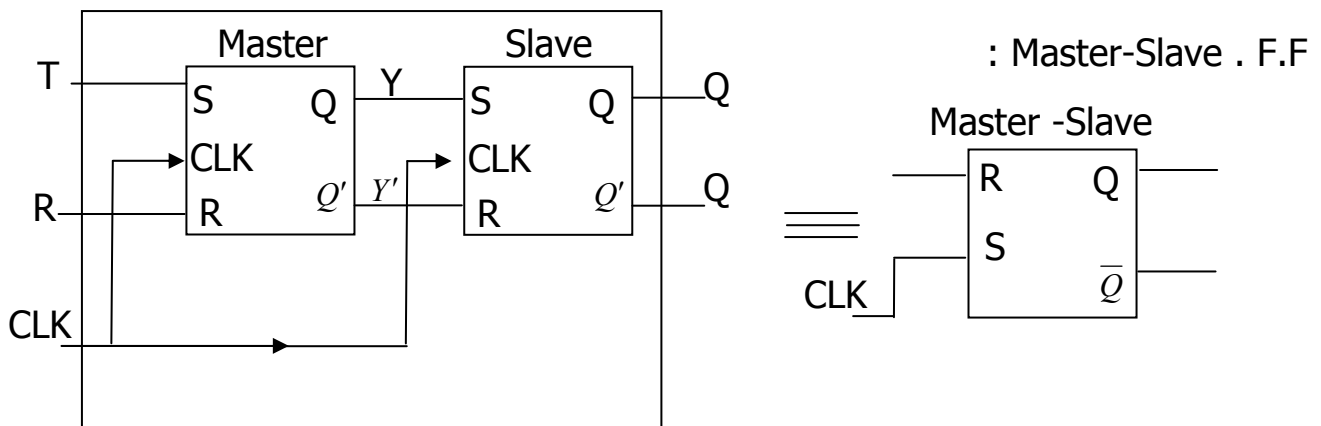


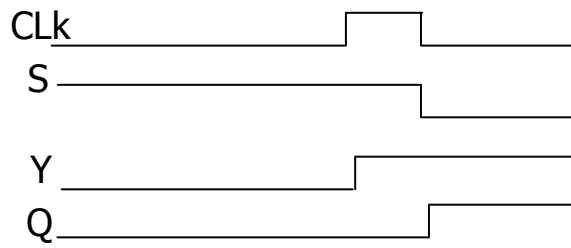
نکته : در سطح مادامی که CLK وجود دارد و k و z هر دو یک باشند و JKFF حساس

به سطح باشد خروجی نوسان می کند برای رفع این مشکل :

۱- استفاده از JK-FF حساس به لبه

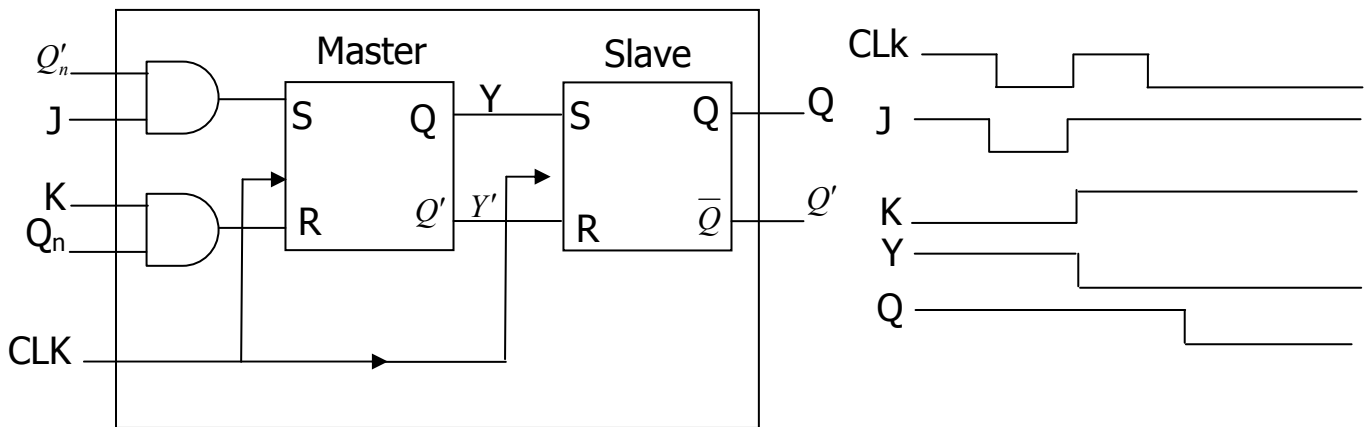
۲- با استفاده از F.F Master-Slave آنرا JK تبدیل به حساس به لبه می کند .



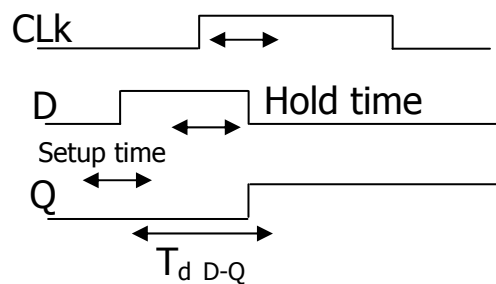
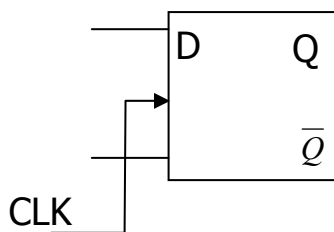


CLK	Master	Slave
Low	disActive	Active
High	Active	disActive

R = "C"



و بدین ترتیب مشکل حل شد .



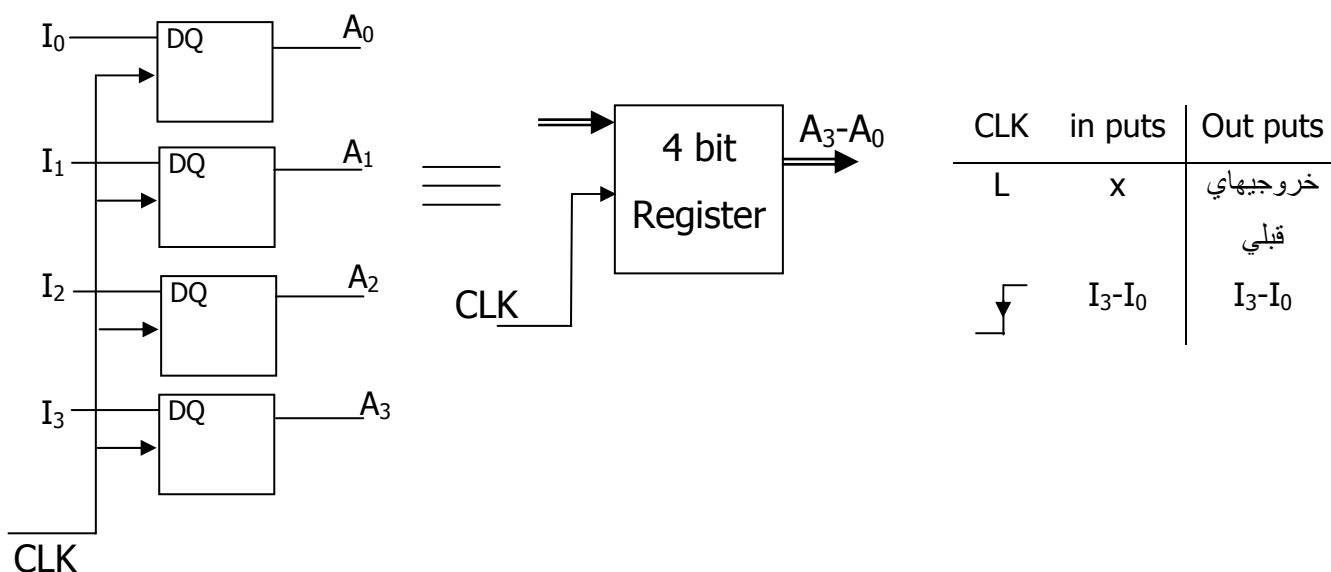
تاخیر

نکته : باید کمی قبل از آمدن CLK «1» شود تا بتوانند تغییرات لازم را در مدار بدهد (آماده باشد) که به این زمان Setup time گویند (حداقل زمان کمی می بایست قبل از آمدن به کلاک ورودی (Stable بماند) همچنین برای انجام تغییرات لازم در مدارات داخلی D نباید همزمان با آمدن CLK صفر شود بلکه باید کمی بگذرد که به این مدت

Hold time (گویند) = مدت زمانی که لازم است تا ورودی D پس از آمدن به کلاک ثابت بماند تا تغییر مناسب در خروجی صورت گیرد.

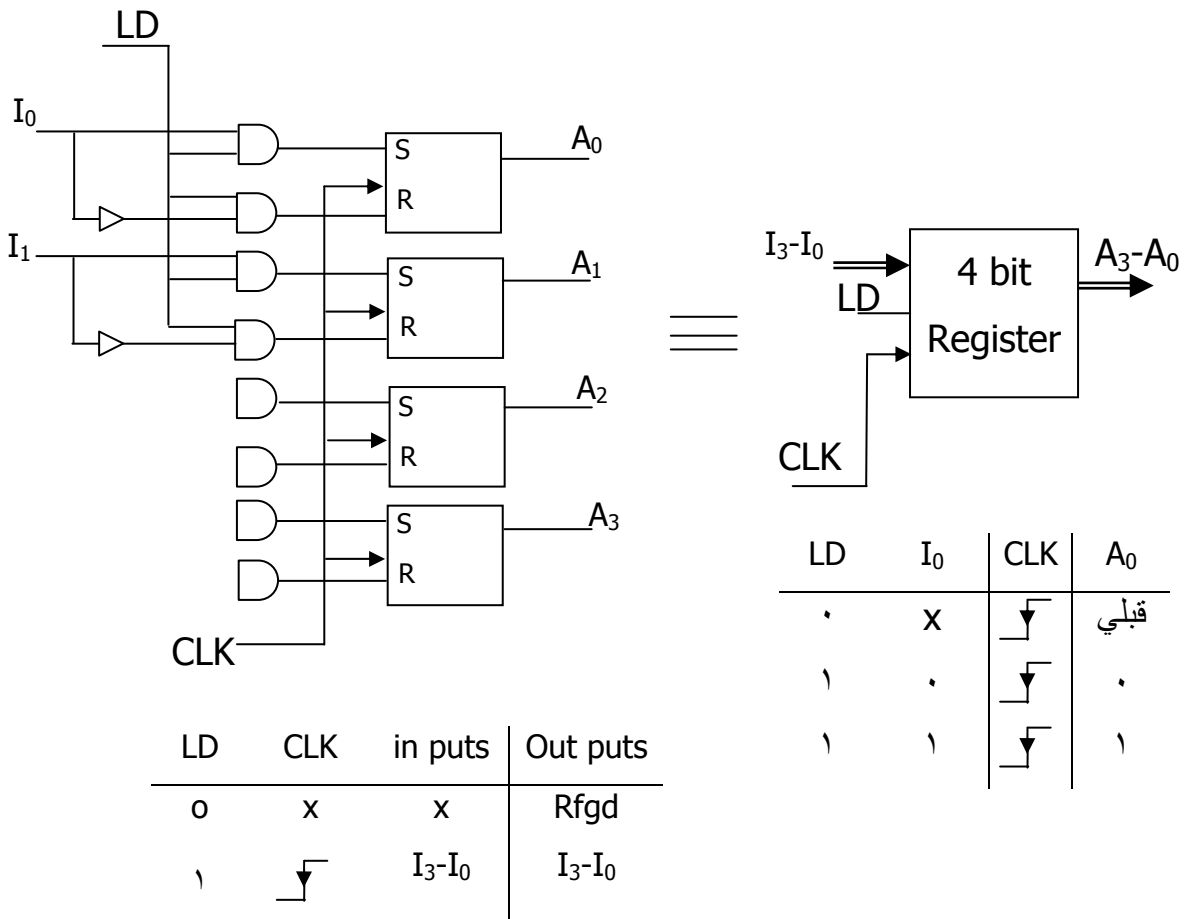
رجیسترها یا ثابتها :

به مجموعه ای از FF ها که با کلاک مشترک کار می کنند و برای ثبت اطلاعات دودویی بکار می روند ثابت یا رجیستر گویند.

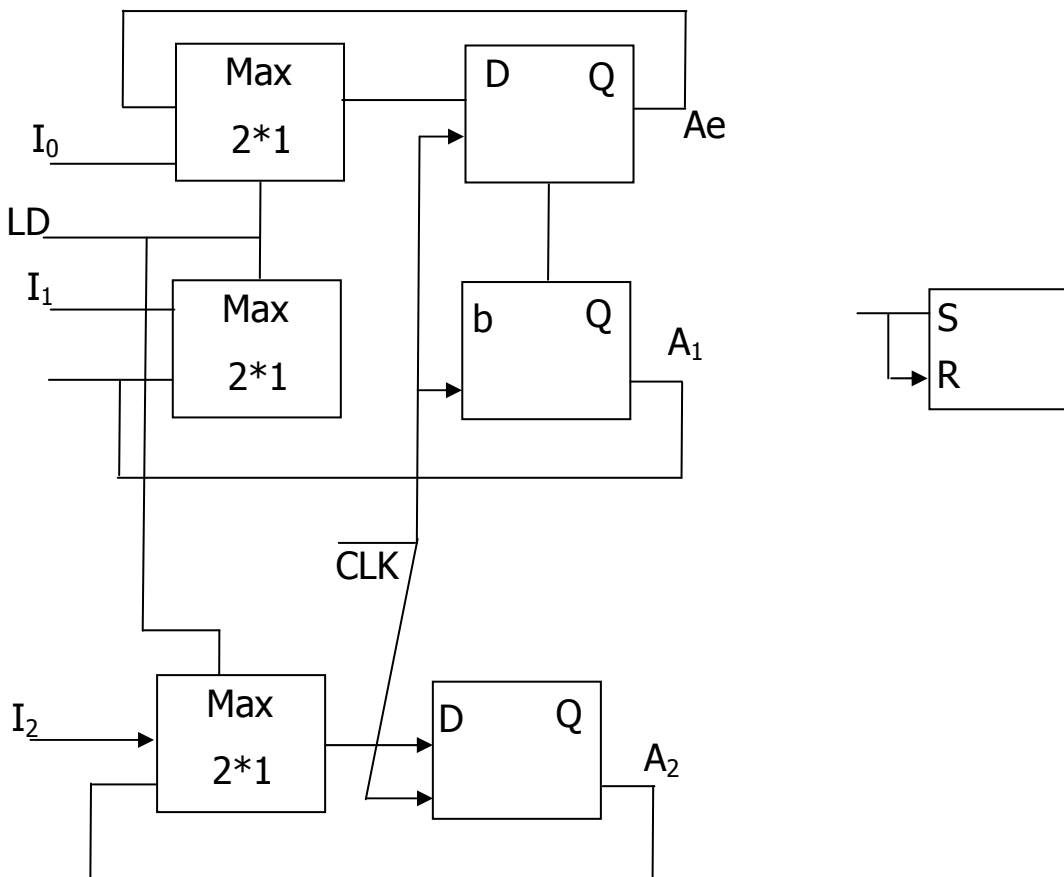


نکته : اگر D-FF حساس به لبه باشد کلاک ورودی ، CLK نام دارد و اگر حساس به سطح باشد ورودی کلاک G (گیت) نام دارد.

نکته : چون بایستی CLK به کلاک سیستم ، هیچ کنترلی روی ثابت ندایم (زیرا کلاک سیستم توأمآ می آید) لذا از یک ورودی دیگر به نام LD (Load) استفاده می کنیم که به چند تو مایه های Enable است .



اگر ثبات ۴ بیتی با کمک D-FF و با ورودی LD :



شیفت رجیسترها :

برای جابه جایی اطلاعات باینتری (شیفت) به سمت چپ یا راست.

کاربرد : ضرب ، تقسیم ، جابه جایی اطلاعات .

۱۰

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

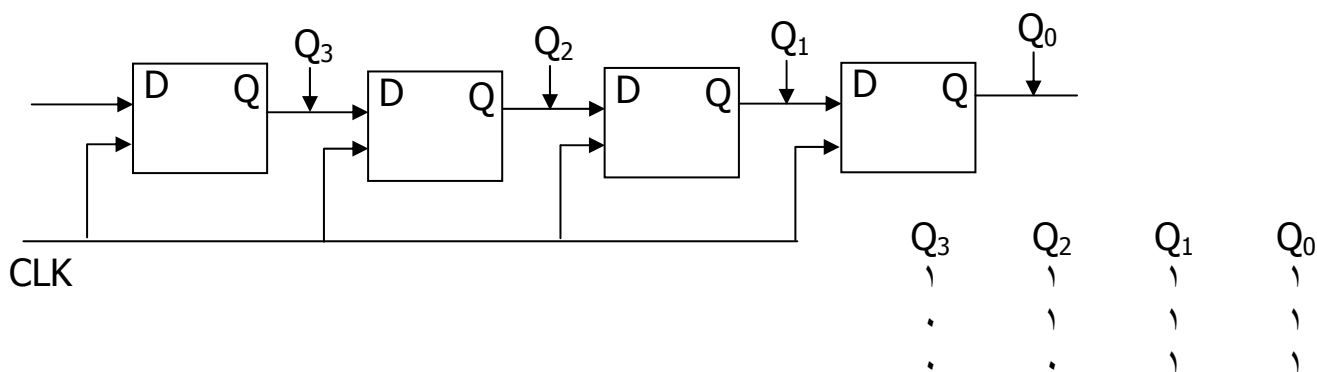
۵

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

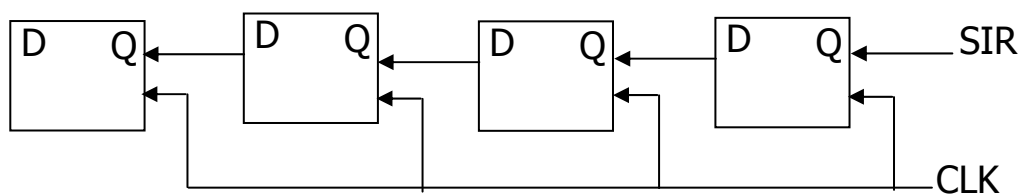
۱۰

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

مثال :



مدار شیفت رجیستر شیفت به چپ :



تقسیم بندی با توجه به ورودی و خروجی :

SI →

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

۱- ورودی سری - خروجی سری SI/SO

SI →

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

۲- ورودی سری - خروجی موازی SI/PO

→

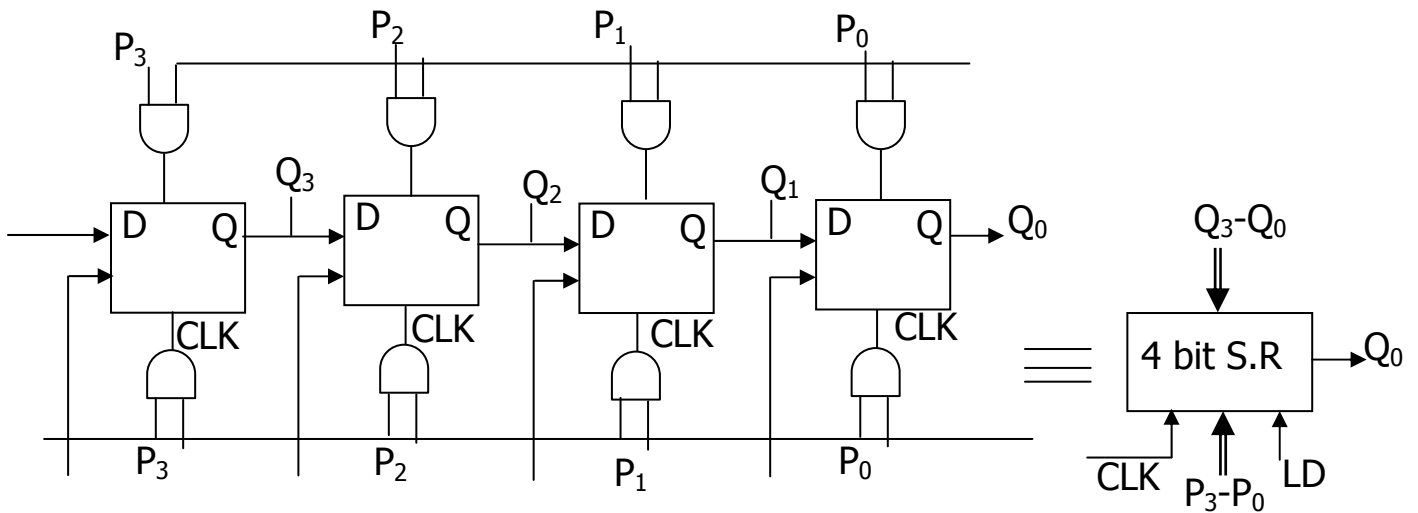
1	0	1	0
---	---	---	---

 →

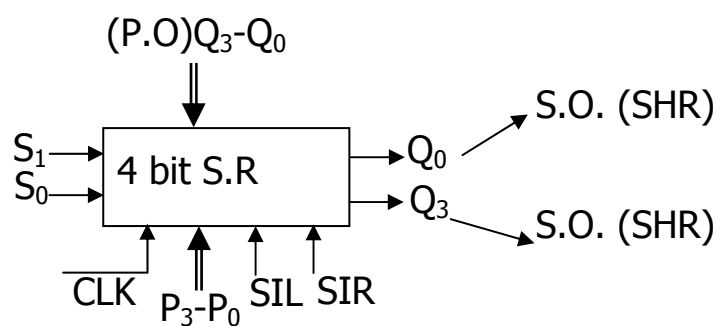
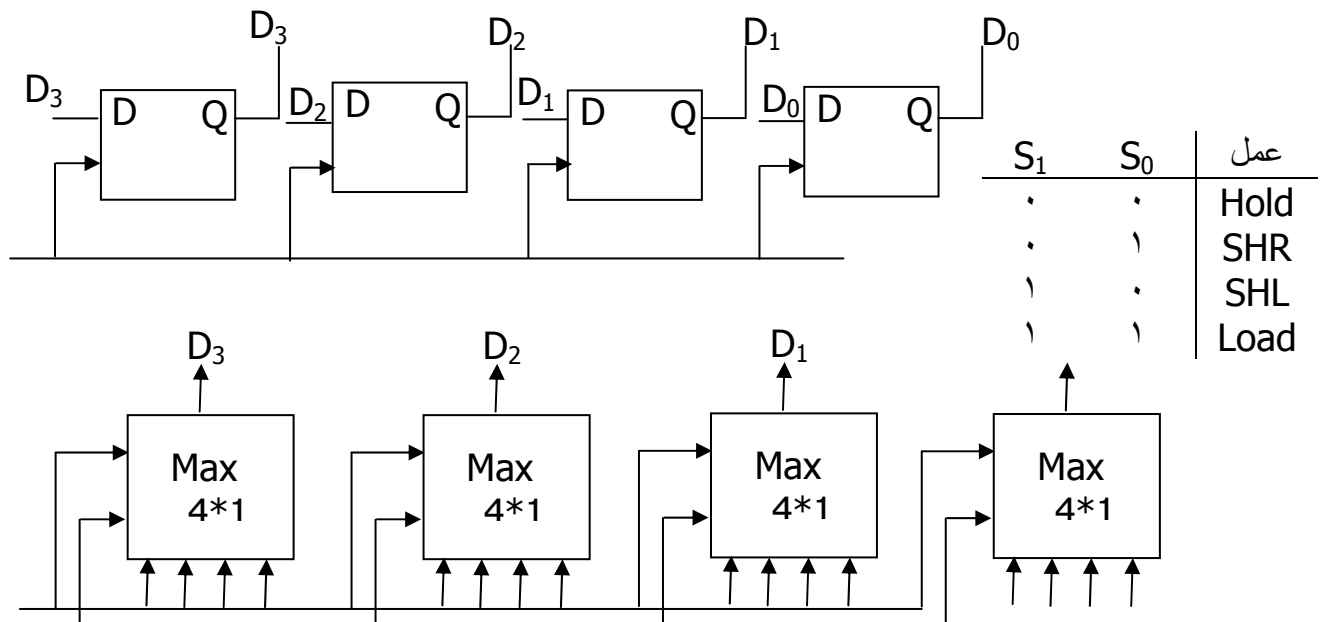
۲- ورودی موازی - خروجی سری PI/SO

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

۴- ورودی موازی - خروجی موازی PI/PO



هدف: طراحی شیفت رجیستری که با دو خط انتخاب S_1 , S_0 اعمال باردهی، جابه جایی به چپ یا راست به همراه نگهداری اطلاعات را انجام می دهد. که به آن شیفت رجیستری عمومی یا همه منظوره گفته می شود.

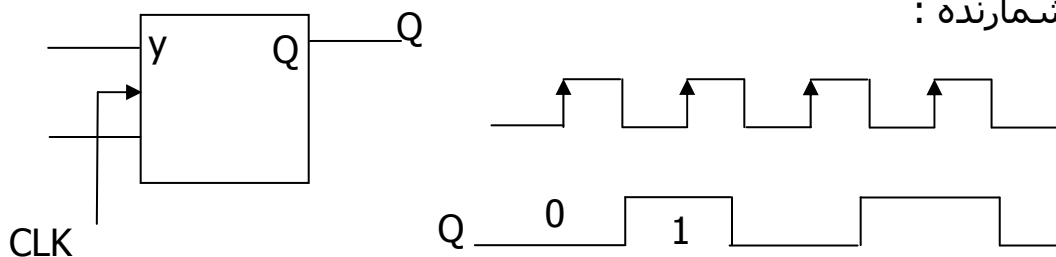


شمارنده ها :

در حالت کلی : ۱- سنکرون یا همزمان : تغییر خروجی تمام F.F ها با تاخیر ثابتی نسبت به CLK صورت می گیرد .

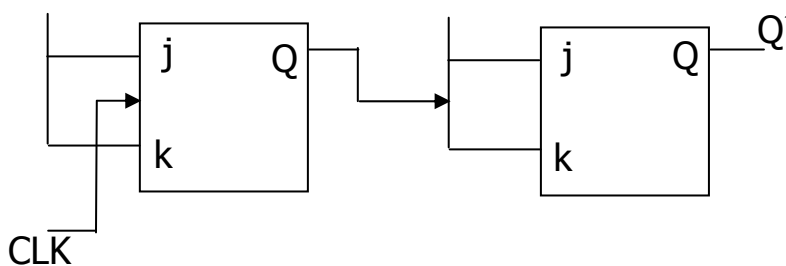
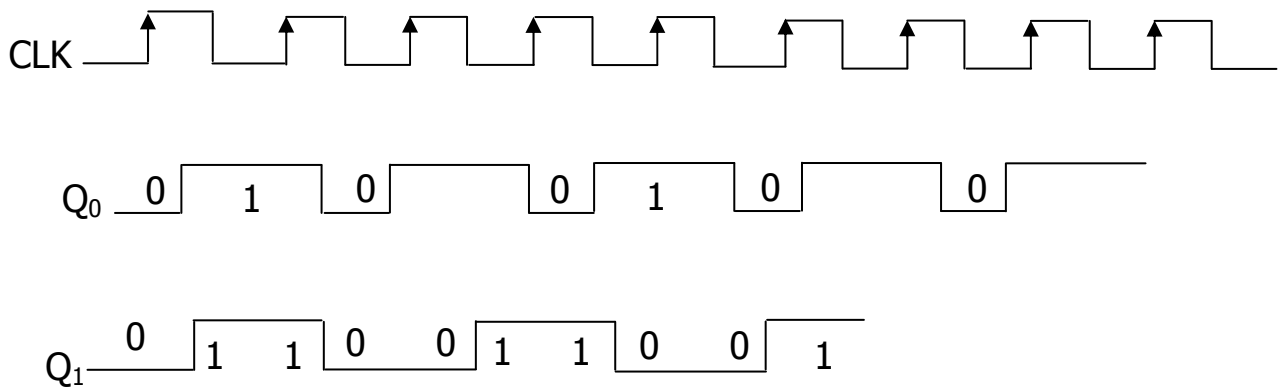
۲- آنسکرون یا غیر همزمان یا ضربان : تغییر خروجی تمام F.F ها با تاخیر متغیر (ثابت به CLK) صورت می گیرد .

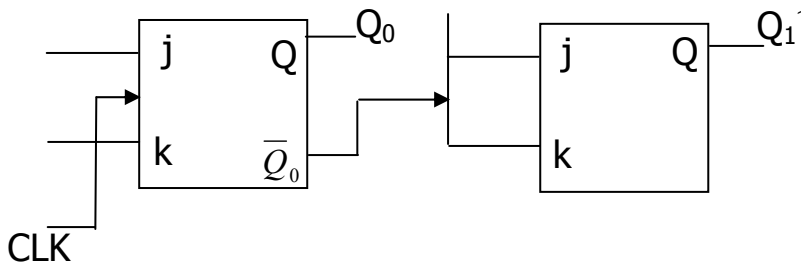
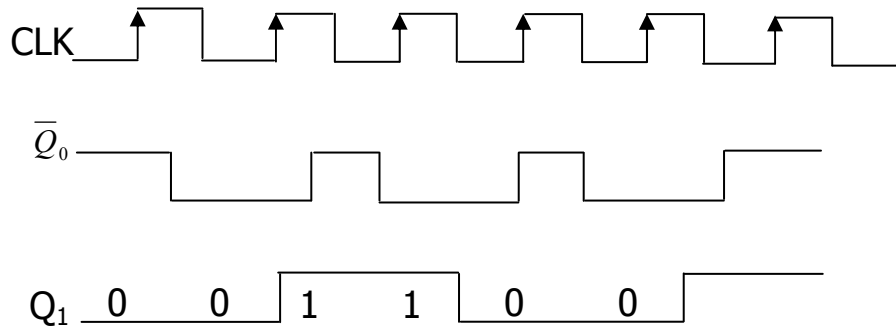
ساده ترین شمارنده :



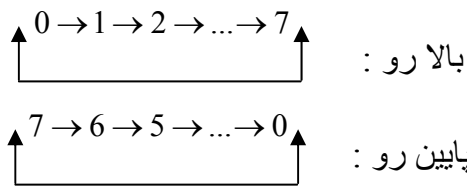
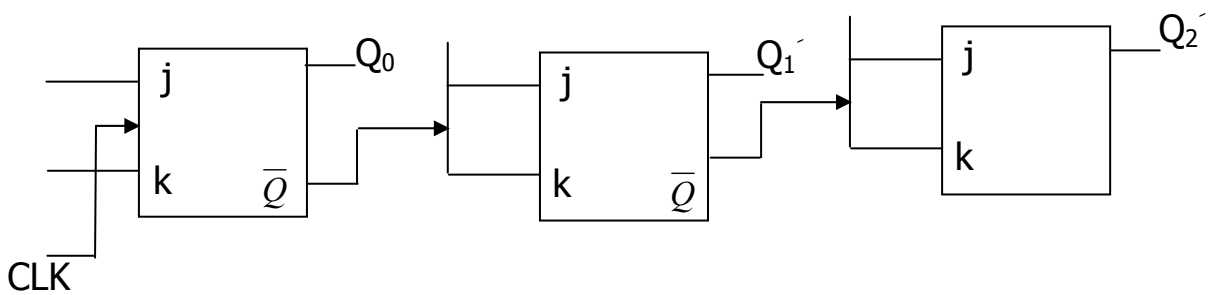
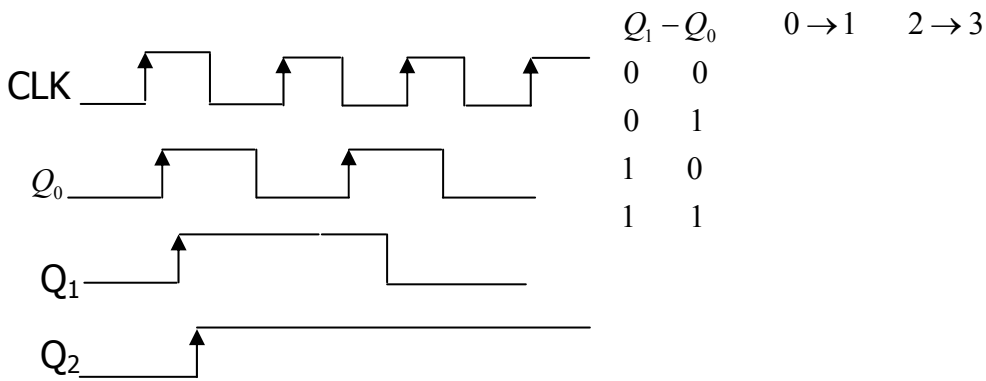
$$F_Q = \frac{f_{CLK}}{2} \text{ خروجی تقسیم بر 2}$$

توسعه شمارنده :

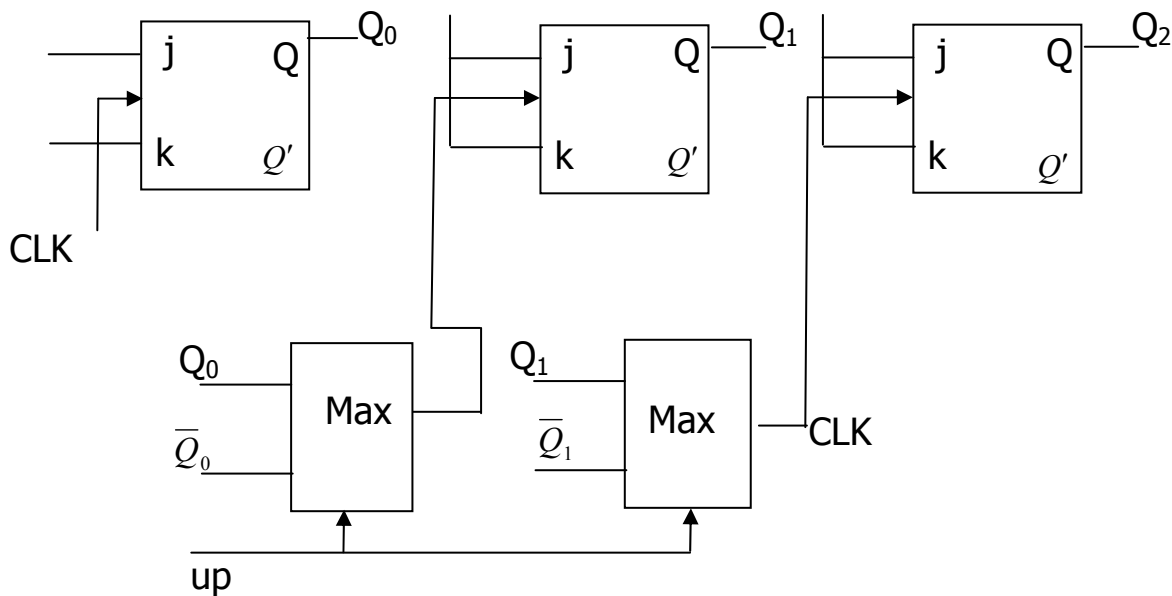




شمارنده آسنکرون



مثال : شمارنده آسنکرونی طراحی کنید که با یک خط کنترل به سمت بالا یا پایین شمارش کند .



نکته :

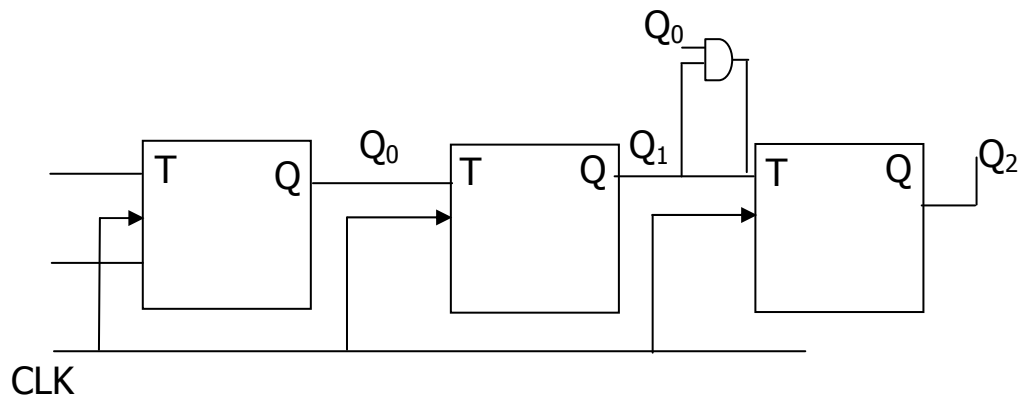
- در 2^n لزومی ندارد که ورودیهای J, k حتماً به ۱ وصل می شود .

- برای شمارنده های آسنکرون غیر 2^n روش خاصی وجود ندارد .

ورودیهای	خروجیها	نحوه شمارش
Q ها	Q ها	پایین رو
Q ها	\bar{Q} ها	بالا رو
\bar{Q} ها	Q ها	بالا رو
\bar{Q} ها	\bar{Q} ها	پایین رو

شمارنده سنکرون :

Q_2	Q_1	Q_0	
۰	۰	۰	۰
۰	۰	۱	۱
۰	۱	۰	۲
۰	۱	۱	۳
۰	۰	۰	۴
۱	۰	۱	۵
۱	۱	۰	۶
۱	۱	۱	۷
۰	۰	۰	۸



نکته :

فقط تاخیر کلاک تا خروجی را داریم . چون این AND قبل از آمدن کلاک خروجی اش را آماده می کند .

$$\text{Max } F_{\text{CLK}} = \frac{1}{\text{تایخیر AND} + \text{تایخیر FF}}$$

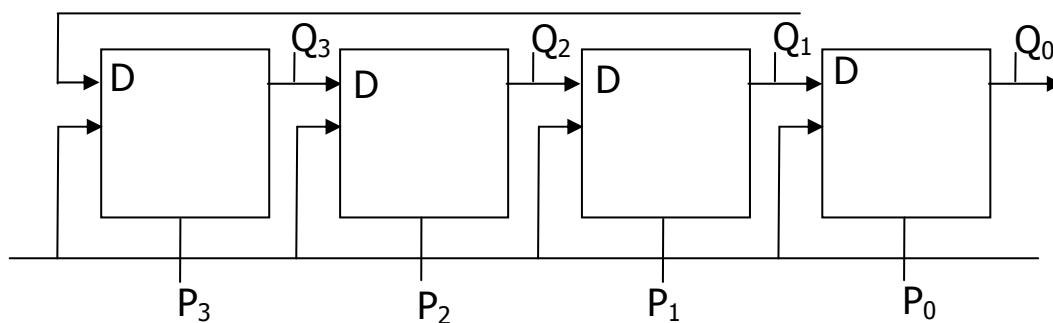
$$\text{Max } F_{\text{CLK}} = \frac{1}{ntd} \quad \text{ولی برای آنسکرون :}$$

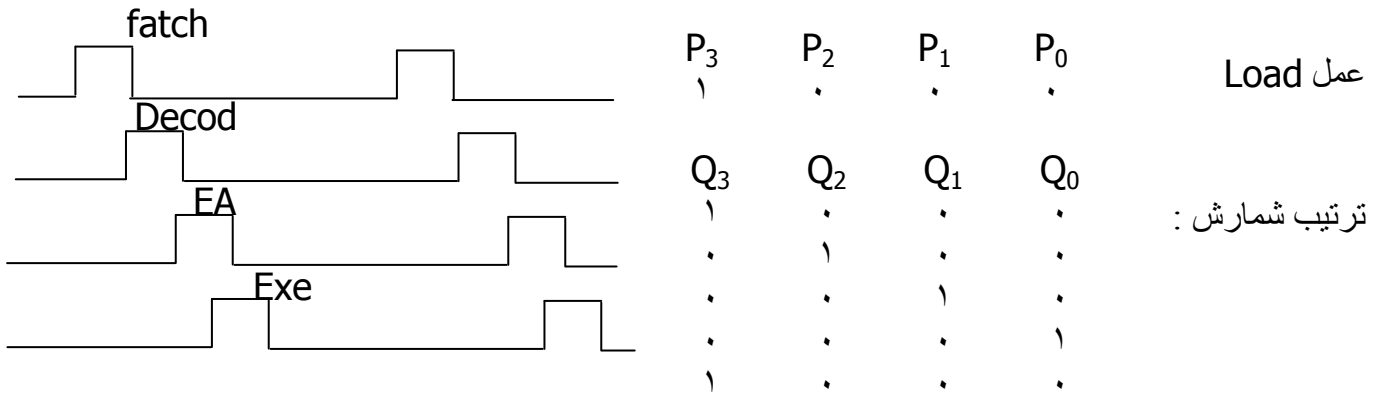
مثال : اگر $FF_{td} = 20_{ns}$ باشد آنگاه بیشینه ی F_{CLK} چقدر است ؟ (برای شمارنده ی آنسکرون ۴ بیتی)

$$td_{tot} = 4 * td = 80_{ns} \quad \rightarrow F_{max} = 12.5 \text{ MH}$$

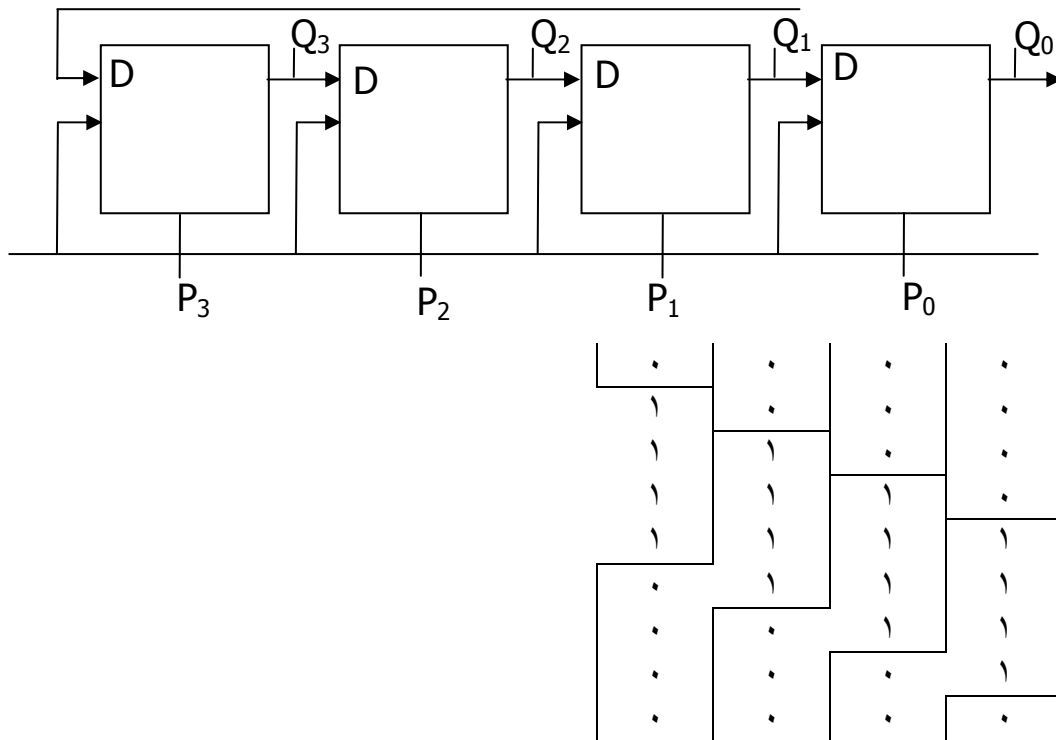
شمارنده های خاص :

۱- شمارنده حلقوی



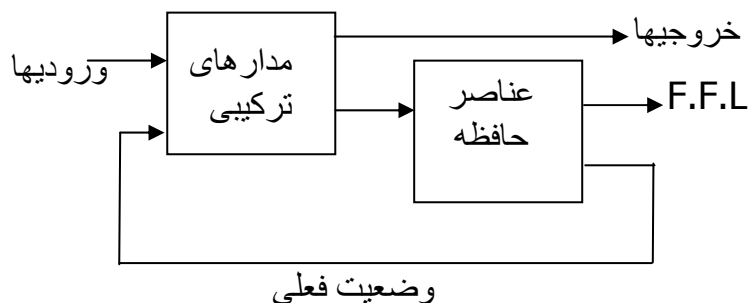


۲- شمارنده ی جانسون :



طراحی مدارهای منطقی ترتیبی :

ساختار کلی مدارهای ترتیبی :



مثال از مدار منطقی ترتیبی :

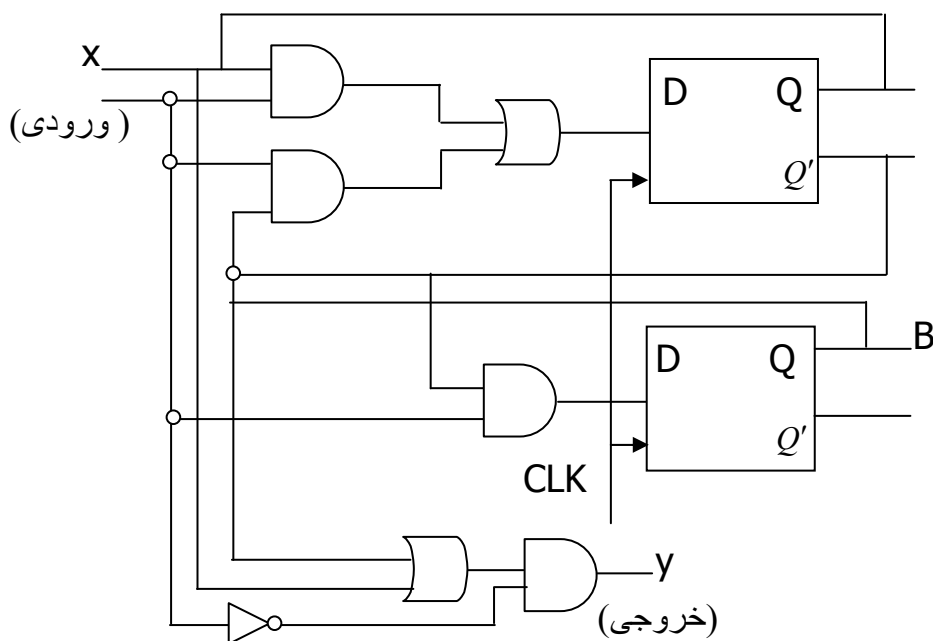
$$A(t+1) = A(t)x(t) + B(t)x(t)$$

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

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

$$\downarrow x'(t)$$

$$A(t)$$



جدول حالت :

ترتیب زمانی ورودیها ، خروجیها و وضعیت FF ها را در جدولی بنام جدول حالات می توان بیان نمود . این جدول شامل قسمتهای حالت فعلی ، ورودی ، حالت بعدی و خروجی است .

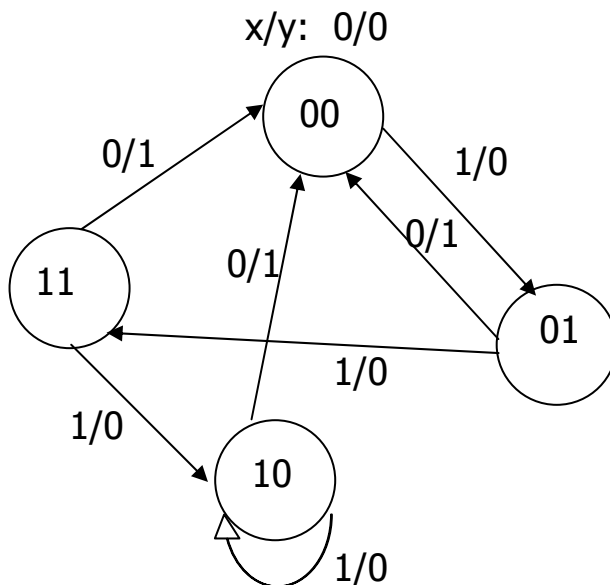
جدول حالت مثال بالا :

حالت فعلی		ورودی	حالت بعدی		خروجی
A	B		A	B	
۰	۰	x	۰	۰	y
۰	۰	۱	۰	۱	۰
۰	۱	۰	۰	۰	۱
۰	۱	۱	۱	۱	۰
۱	۰	۰	۰	۰	۱
۱	۰	۱	۱	۰	۰
۱	۱	۰	۰	۰	۱
۱	۱	۱	۱	۰	۰

A	B	x		y	
		x=0	x=1	x=0	x=1
۰	۰	۰	۰	۰	۰
۰	۱	۰	۱	۱	۰
۱	۰	۰	۰	۰	۱
۱	۱	۰	۱	۰	۰

طراحی جدول به روش دیگر :

دیاگرام حالت (State Diagram) :



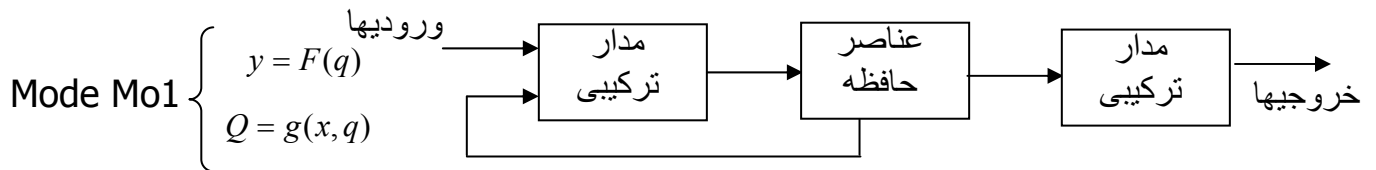
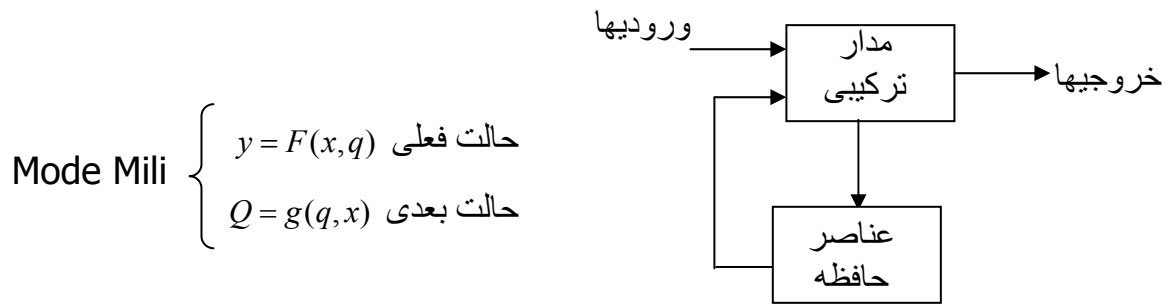
انواع مدارهای ترتیبی :

مدهای میلی و مور :

۱- مد میلی : خروجی بر اساس حالت فعلی ورودیها مشخص می شود .

۲- مد مور : خروجی فقط از روی حالات فعلی مشخص می شود ، به صورت مستقیم

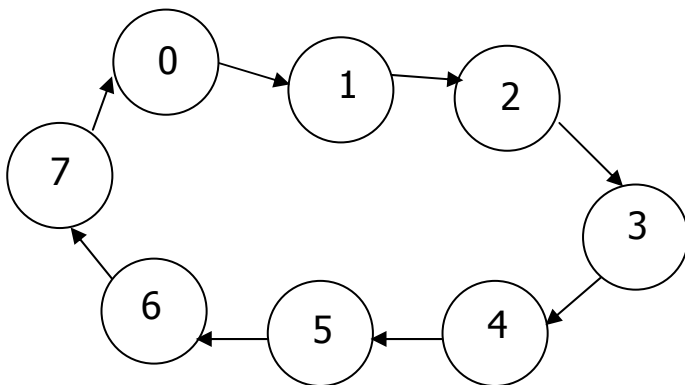
به ورودی ربط ندارد یا در رابطه ی خروجی ورودیها دیده نمی شود .



نکته : در مد میلی اگر ورودی تغییر کند حتی بدون آمدن CLK ممکن است که خروجی تغییر کند ولی در مد مور چنین نیست لذا برای رفع این مشکل ورودیهای مد میلی را با CLK همزمان می کنند .

نکته : مدار صفحه ی قبلی مثالی از یک مد و شمارنده های مثالی از مد مور هستند .

نکته : در مد مور نمایش State Diagram تنها برحسب ورودی است یعنی :



جدول تحریک F.F ها :

S.R را چه در نظر بگیریم تا به حالت Q ها برسیم .

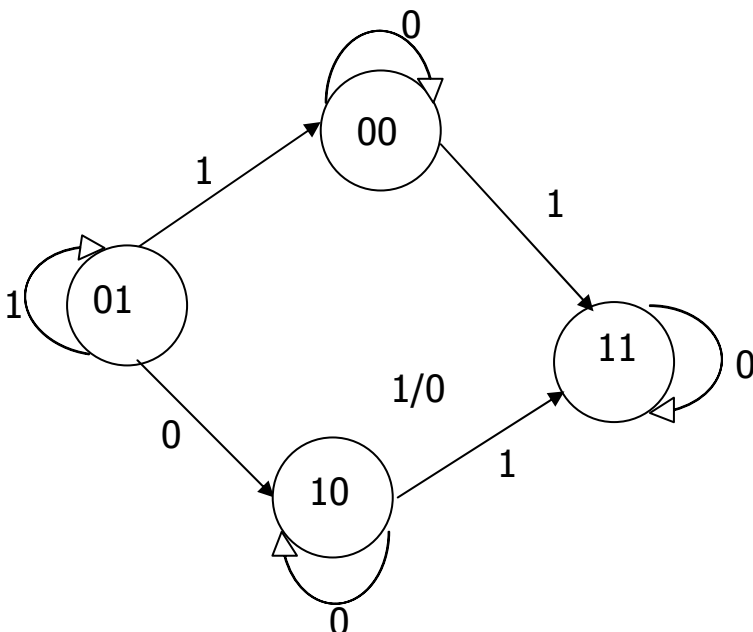
Q(t+1)	Q(t)	S	R	D	J	K	T
۰	۰	۰	X	۰	۰	X	۰
۰	۱	۱	۰	۱	۱	X	۱
۱	۰	۰	۱	۰	X	۱	۱
۱	۱	X	۰	۱	X	۰	۰

مراحل طراحی مدارهای ترتیبی :

- ۱- توصیف لفظی عملکرد مدار : ۱- دیاگرام حالت ۲- دیاگرام زمانی
- ۲- با توجه به اطلاعات مفروض در مورد مدار جدول حالات تنظیم شود .
- ۳- تعداد حالات را در صورت امکان کاهش دهید .
- ۴- اگر در جدول حالات ، سمبل های حرفی وجود دارد ، آنها را با مقادیر دودویی جایگزین کنید (تشخیص حالت)
- ۵- تعداد F.F ها را مشخص کنید (با توجه به جدول حالات) و به هر کدام سمبل حرفی تخصیص دهید .
- ۶- نوع F.F ها را مشخص کنید (معمولاً در صورت مساله)
- ۷- با توجه به جدول حالات ، جداول تحریک و خروجی را بدست آورید .
- ۸- ساده سازی روابط خروجی ، ورودیهای تحریک از روی جدول خروجی و تحریک با کمک روشهای متعارف مثل روش کارنو .

مثال : دیاگرام حالت :

دیاگرام حالت زیر مفروض است . مدار ترتیبی لازم را به کمک kffz رسم کنید .



فعلی		ورودی	بعدی		
A	B	x	A	B	
۰	۰	۰	۰	۰	3 امکان ندارد
۰	۰	۱	۰	۱	4 نداریم
۰	۱	۰	۱	۰	5 دوتا FF
۱	۰	۰	۱	۰	JKFF 6
۱	۰	۱	۱	۱	
۱	۱	۰	۱	۱	
۱	۱	۱	۰	۰	

J_A	K_A	J_B	K_B
۰	x	۰	x
۰	x	۱	x
۱	x	x	۱
۰	x	x	۰
x	۰	۰	x
x	۰	1	x
x	۰	x	۰
x	۱	x	۱

A \ AB	00	01	10	11
0	۰	۰	۰	۱
1	x	x	x	x

$: j_A = Bx'$

A \ Bx	00	01	10	11
0	x	x	x	x
1	۰	۰	۱	۰

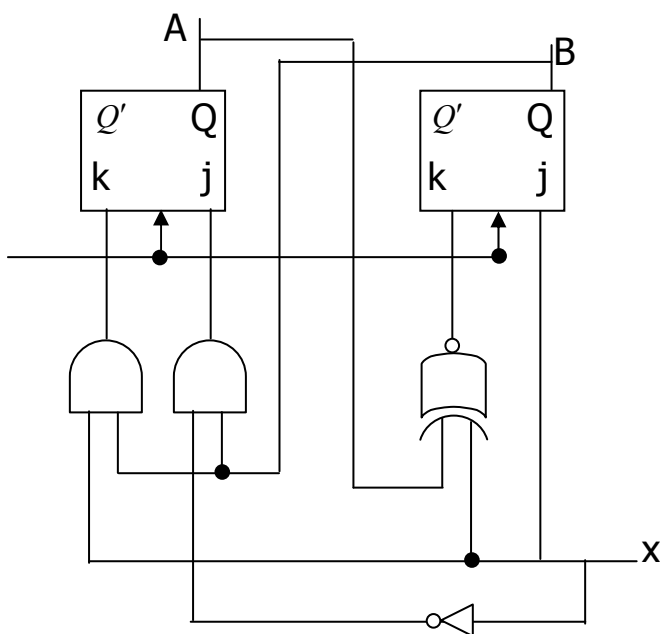
$: k_A = Bx$

A \ 0x	00	01	10	11
0	۰	۱	x	x
1	۰	۱	x	x

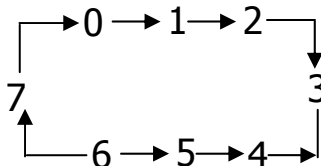
$: j_B = x$

A \ Bx	00	01	10	11
0	x	x	۰	۱
1	x	x	۱	۰

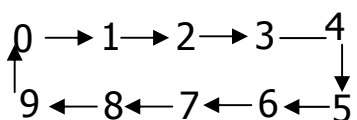
$: k_B = (A \oplus x)'$



تمرین : شما رنده ای طراحی کنید که ترتیب زیر بشمارد با کمک T.F.F

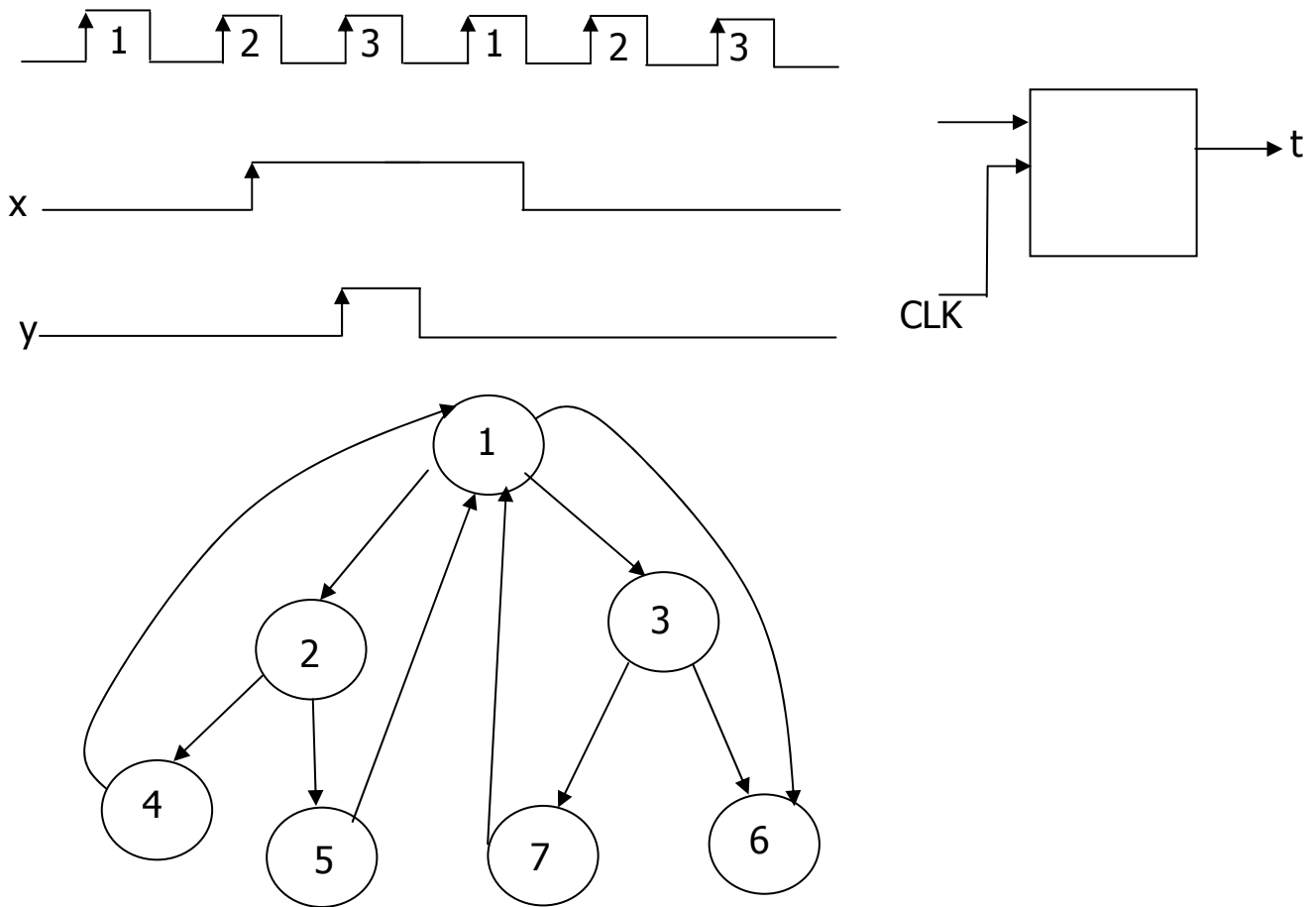


مثال : شما رنده BCD با کمک T.F.F :



نکته : می توانستیم از همان ابتدا مدار را بر این مبنا طراحی کنیم که اعداد Dont Care به صفر بروند .

مثالی از آخر : مداری طراحی کنید که به صورت پریودیک پالس ساعت ورودی x را بررسی و در صورت فرود بودن تعداد یکهای دریافتی ، خروجی یک شود .



جدول حالات :

حالت فعلی	حالت بعدی		خروجی	
	x=0	x=1	x=0	x=1
1	2	3	0	0
2	5	4	0	0
3	6	7	0	0
4	1	1	1	0
5	1	1	0	1
6	1	1	1	0
7	9	1	0	1

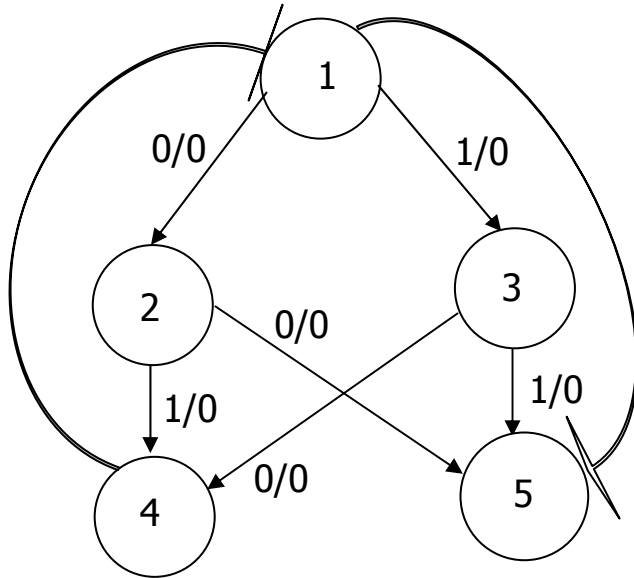
حالت‌های مشابه :

به ازای ورودیهای یکسان

حالت بعدی و خروجی

یکسان نتیجه شود.

لذا 4 با 6 و 5 با 7 مشابه هستند در نتیجه می توان گفت 6 و 7 زیادی هستند .



: Up dated State Diagram

تخصیص حالات : سعی می شود از يك حالت به حالت دیگر مجاور منطقی در جدول کار نو باشند .

	حالت فعلی			بعدي		خروجی	
	A	B	C	x=0	x=1	x=0	x=1
1	۰	۰	۰	۰۰۱	۰۱۱	۰	۰
2	۰	۰	۱	۱۱۱	۱۰۱	۰	۰
3	۰	۱	۱	۱۰۱	۱۱۱	۰	۰
4	۱	۰	۱	۰۰۰	۰۰۰	۱	۰
5	۱	۱	۱	۰۰۰	۰۰۰	۰	۱

با کمک T.F.F

خلاصه ...

$$T_A = C$$

$$T_B = AB + \bar{C}x + \bar{A}C\bar{x}$$

$$T_C = A + \bar{C}$$

$$y = \bar{A}\bar{B}x + ABx$$