

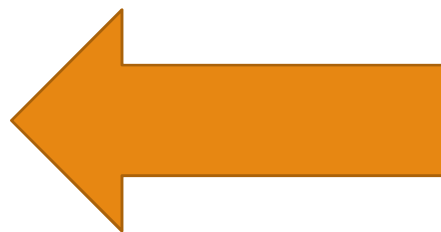


## انواع مدارهای ترتیبی

مدارهای ترتیبی شامل گروهی از فلیپ فلاپ ها و گیت های ترکیبی می باشند. فلیپ فلاپ ها عناصر اصلی سازنده ی مدارات ترتیبی هستند و بدون آن ها مدار ترتیبی به یک مدار ترکیبی ساده تبدیل می شود. مدار دارای فلیپ فلاپ حتی در صورتی که هیچ گیت منطقی در آن نباشد، یک مدار ترتیبی در نظر گرفته می شود. مدارهای ترتیبی بسته به کاری که انجام می دهند به گروه های مختلفی تقسیم می شوند. دو گروه مهم از مدار های ترتیبی عبارتند از:

**1) شمارنده ها (Counters)**

**2) ثبات (رجیستر) ها (Register)**





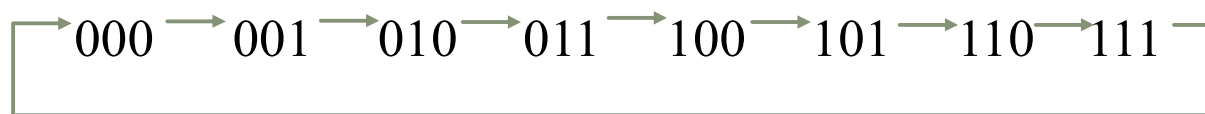
## انواع مدارهای ترتیبی

### ثبات (رجیسترها) (Registers)

یک ثبات  $n$  بیتی شامل  $n$  فلیپ فلاپ می باشد و قادر به ذخیره سازی  $n$  بیت اطلاعات می باشد. (هر فلیپ فلاپ یک بیت) ثبات ها علاوه بر فلیپ فلاپ ها می توانند شامل گیت های منطقی نیز باشند که این گیت ها چگونگی انتقال داده به ثبات را کنترل می کنند.

### شمارنده ها (Counters)

یک شمارنده نوعی ثبات است که توالی خاصی از اطلاعات باینری را طی می کند. مانند ثبات ها، یک شمارنده  $n$  بیتی دارای  $n$  فلیپ فلاپ می باشد. مثلاً یک شمارنده  $3$  بیتی افزایشی دارای  $3$  فلیپ فلاپ می باشد و توالی شمارش آن به صورت زیر خواهد بود. گذر از هر حالت به حالت دیگر می تواند با هر پالس ساعت انجام گیرد.

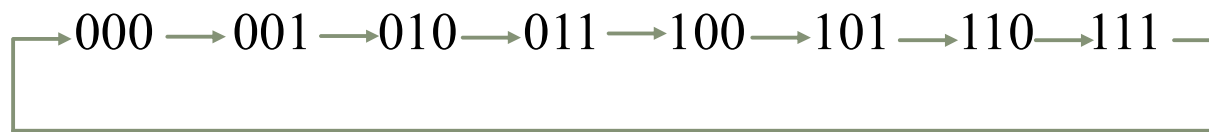




## شمارنده ها (Counters)

شمارنده، ثباتی است که با دریافت پالس های ورودی، توالی خاصی از حالت ها را طی می کند.

- ☐ یک شمارنده ی  $n$  بیتی دارای  $n$  فلیپ فلاپ می باشد.
- ☐ پالس های ورودی می تواند همان پالس های ساعت باشد یا از منبع خارجی دیگری وارد شود.
- ☐ توالی حالت ها ممکن است، یک توالی از اعداد باینری (صعودی یا نزولی) یا هر توالی خاص دیگر باشد.
- ☐ شمارنده ای که توالی اعداد باینری را طی می کند، شمارنده ی باینری (Binary Counter) نام دارد.





## شمارنده ها (Counters)

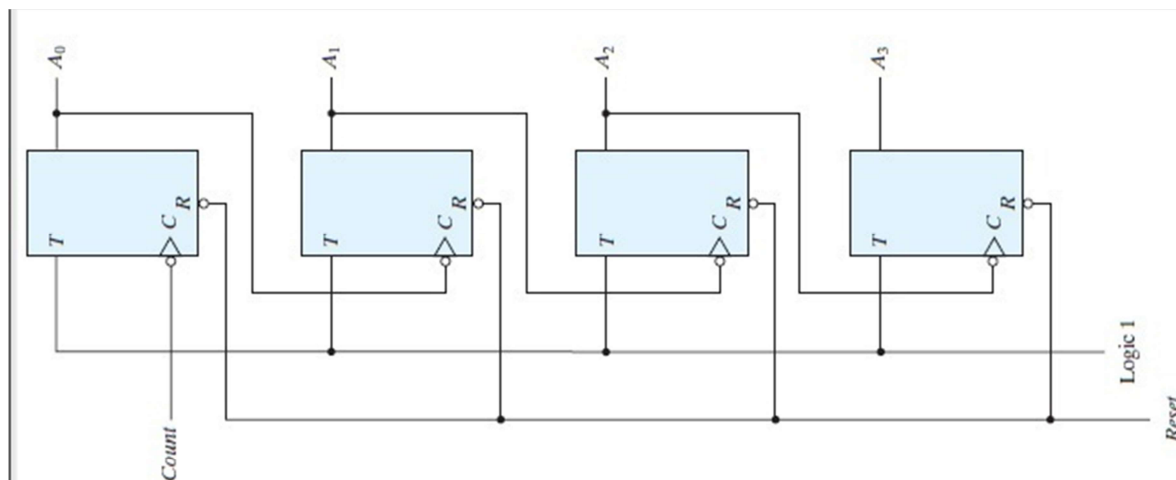
شمارنده ها به دو دسته ی کلی تقسیم می شوند:

### 1) شمارنده های سنکرون (Synchronous Counters)

در این شمارنده ها همه ی فلیپ فلاپ ها با یک پالس ساعتتریگر می شوند.

### ۲) شمارنده های ریپل (Ripple Counters)

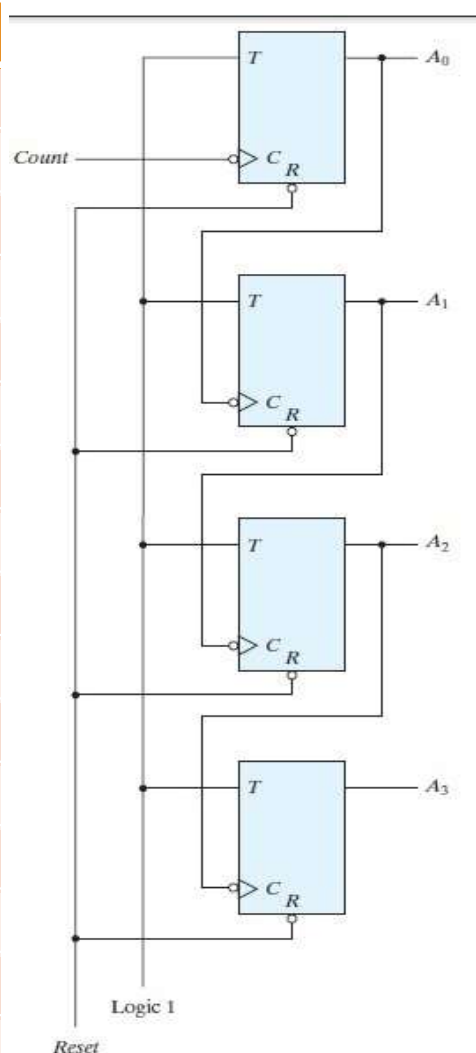
در این شمارنده ها همه ی فلیپ فلاپ ها با یک پالس ساعت یکسانتریگر نمی شوند، بلکه خروجی یک فلیپ فلاپ می تواند به عنوان سیگنالتریگرفلیپ فلاپ دیگر به کار برده شود.





دانشگاه علم و صنعت ایران

A3	A2	A1	A0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



## شمارنده ی رپل باینری

شمارنده ی رپل باینری ۴ بیتی را در نظر بگیرید. این شمارنده از وضعیت 0000 شروع کرده و با هر پالس ساعت به عدد باینری بعدی می رسد، این کار تا رسیدن به 1111 ادامه خواهد داشت، سپس به 0000 برگشته و این توالی تکرار می شود. با نگاه به جدول روبرو در می یابیم که

با هر لبه ی پالس A0 متمم می شود.

هر وقت A0 از 1 به 0 می رسد A1 متمم می شود!

هر وقت A1 از 1 به 0 می رسد، A2 متمم می شود!

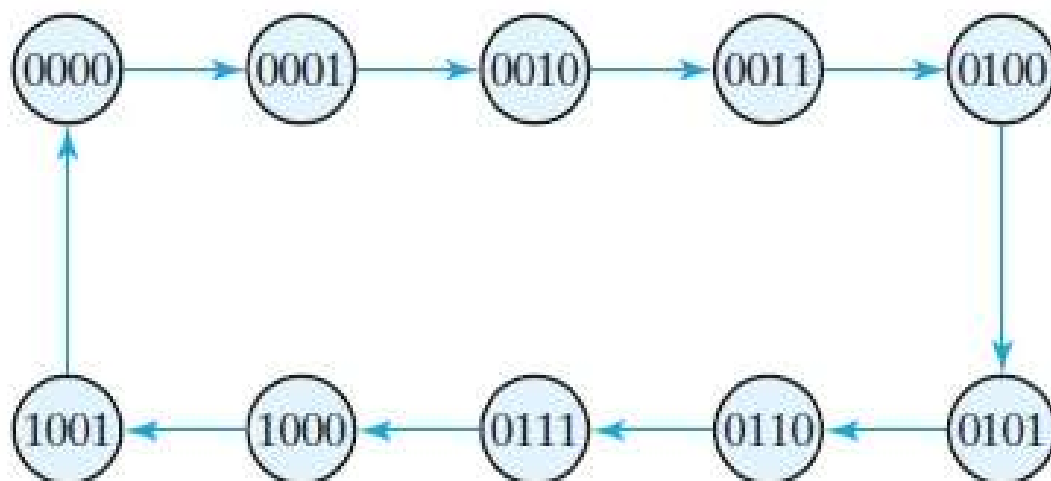
هر وقت A2 از 1 به 0 می رسد، A3 متمم می شود!

برای طراحی این مدار به ۴ فلیپ فلاپ احتیاج می باشد که خروجی فلیپ فلاپ ها A0، A1، A2 و A3 نام گذاری می شوند. با هر لبه ی پایین رونده A0 باید متمم شود، این کار را می توان با استفاده از فلیپ فلاپ T حساس به لبه ی پایین رونده که  $T=1$  باشد انجام داد. A0 می تواند به عنوان کلاک فلیپ فلاپ بعدی استفاده شود (حساس به لبه ی پایین رونده) چرا که هنگامی که A0 از 1 به 0 برسد فلیپ فلاپتریگر شده و چون  $T=1$  انتخاب شده است، خروجی آن که A1 است متمم می شود.



## شمارنده ی ریپل BCD

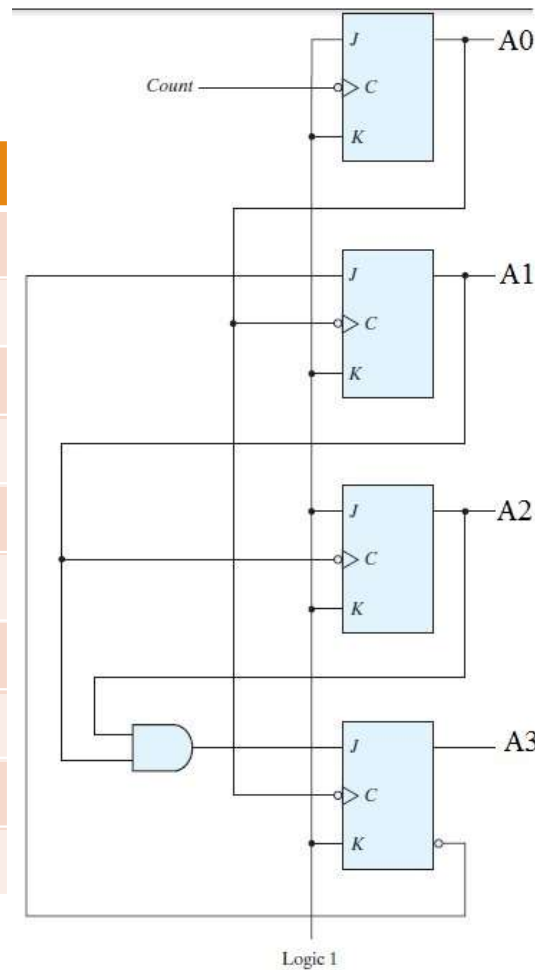
شمارنده ی ریپل BCD ، ۴ بیتی را در نظر بگیرید. این شمارنده از وضعیت 0000 شروع کرده و با هر پالس ساعت به عدد BCD بعدی می رسد، این کار تا رسیدن به 1001 ادامه خواهد داشت، سپس به 0000 برگشته و این توالی تکرار می شود.





دانشگاه علم و صنعت ایران

A3	A2	A1	A0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1



## شمارنده ی ریپل BCD

می خواهیم این شمارنده را با فلیپ فلاپ های JK پیاده سازی کنیم. با توجه به اینکه هر وضعیت BCD به یک کد ۴ بیتی احتیاج دارد، از ۴ فلیپ فلاپ باید استفاده شود.

با نگاه به جدول روبرو در می یابیم که :

با هر لبه ی پالس A0 متمم می شود. (استفاده از فلیپ فلاپ JK، حساس به لبه ی پایین رونده و  $J=K=1$ )

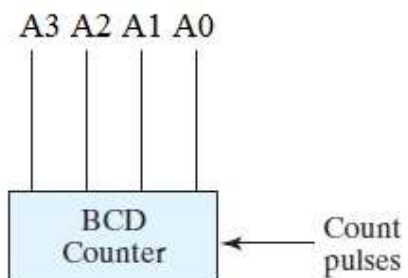
هر وقت A0 از 1 به 0 می رسد A1 متمم می شود به شرطی که  $A3=0$  (یا  $A3'=1$ ) باشد. (یعنی استفاده از فلیپ فلاپ Jk حساس به لبه ی پایین رونده، کلاک = A0 و  $K=1$  و  $J=A3'$ )

هر وقت A1 از 1 به 0 می رسد، A2 متمم می شود! (یعنی استفاده از فلیپ فلاپ Jk حساس به لبه ی پایین رونده، کلاک = A1 و  $J=K=1$ )

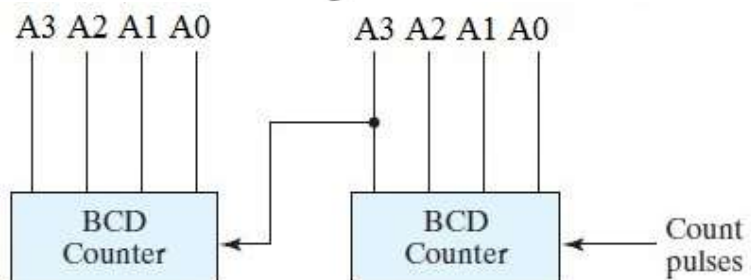
هر وقت  $A1 = A2 = 1$  باشد و A0 از 1 به 0 برسد، A3 متمم می شود. (یعنی استفاده از فلیپ فلاپ Jk حساس به لبه ی پایین رونده، کلاک = A0 و  $K=1$  و  $J=A2.A1$ )



## شمارنده های ریپل ۲ رقمی و ۳ رقمی

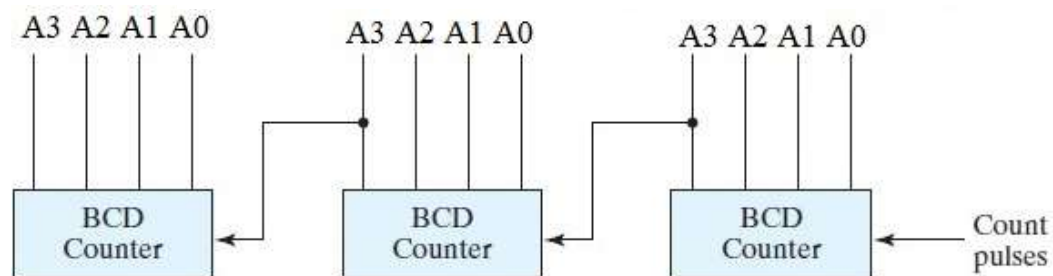


$10^0$  digit



$10^1$  digit

$10^0$  digit



$10^2$  digit

$10^1$  digit

$10^0$  digit

0000	0	از
1001	9	تا

0000 0000	00	از
1001 1001	99	تا

0000 0000 0000	000	از
1001 1001 1001	999	تا





## شمارنده های سنکرون (Synchronous Counters)

در این شمارنده ها پالس ساعت **یکسان و همزمانی** به تمامی فلیپ فلاپ ها می رسد.  
مهمترین شمارنده های سنکرون عبارتند از:

1-شمارنده های باینری

2-شمارنده های BCD

3-شمارنده های باینری با قابلیت بارگذاری

4-شمارنده های باینری با حالات بی استفاده

5-شمارنده های حلقه





## شمارنده ی سنکرون باینری افزایشی

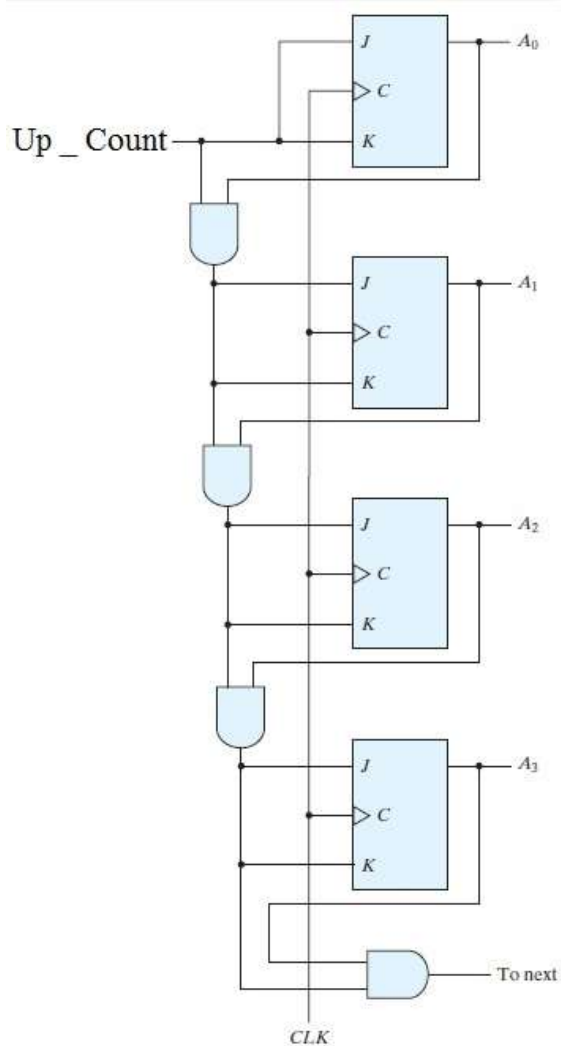
با نگاه به جدول روبرو در می یابیم که  
با هر لبه ی پالس  $A0$  متمم می شود. (یعنی استفاده از فلیپ فلاپ  
JK با شرط  $J=K=1$ )

هر وقت  $A0 = 1$  باشد، با لبه ی پالس  $A1$  متمم می شود. (یعنی  
استفاده از فلیپ فلاپ JK با شرط  $J=K=A0$ )

هر وقت  $A0 = A1 = 1$  باشد با لبه ی پالس  $A2$  متمم می  
شود. (یعنی استفاده از فلیپ فلاپ JK با شرط  $J=K=A1.A0$ )

هر وقت  $A2=A1=A0$  باشد با لبه ی پالس  $A3$  متمم می شود.  
(یعنی استفاده از فلیپ فلاپ JK با شرط  $J=K=A2A1A0$ )

A3	A2	A1	A0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



## شمارنده ی سنکرون باینری افزایشی

اینکه فلیپ فلاپ ها حساس به لبه ی بالا رونده باشند یا پایین رونده هیچ تاثیری در عملکرد مدار ندارد، متنها همگی باید به یک لبه حساس باشند.

## در مدار روبرو:

1) اگر  $Up\_Count = 1$  باشد ، شرایط شمارش نرمال خواهد بود.

(2) اگر  $Up\_Count = 0$  باشد، شمارش متوقف خواهد شد چرا که برای همه ی فلیپ فلاپ ها  $J=K=0$  می باشد و این به معنی حفظ حالت فلیپ فلاپ ها می باشد.



A3	A2	A1	A0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

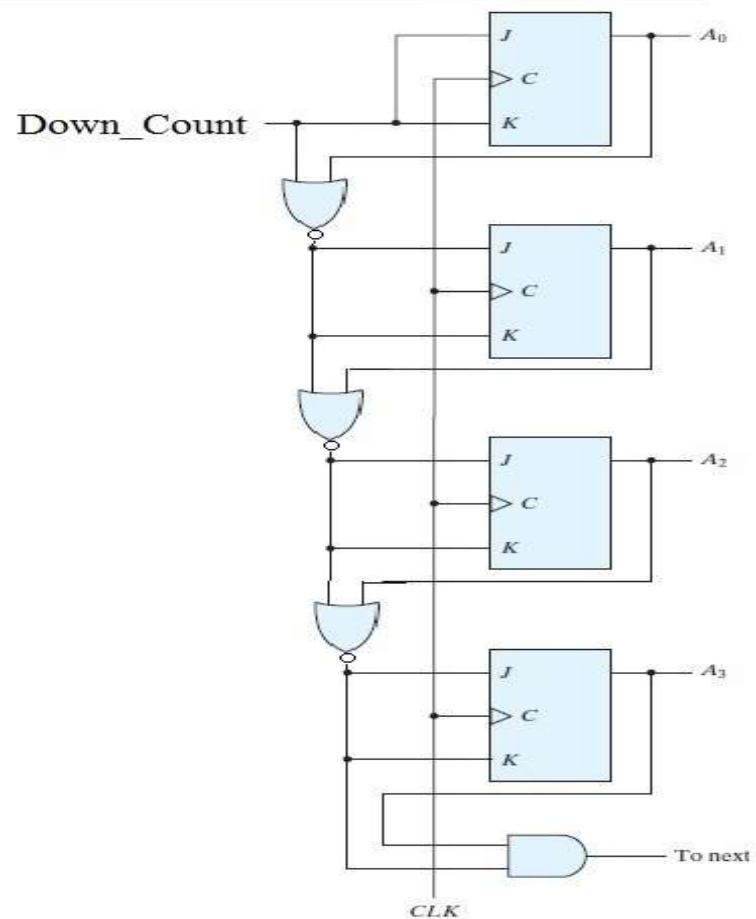
## شمارنده ی سنکرون باینری کاهشی

با نگاه به جدول روبرو در می یابیم که  
با هر لبه ی پالس A0 متمم می شود. (یعنی استفاده از فلیپ  
فلاپ JK با شرط  $J=K=1$ )

هر وقت  $A0 = 0$  باشد، با لبه ی پالس A1 متمم می شود.  
(یعنی استفاده از فلیپ فلاپ JK با شرط  $J=K=A0'$ )

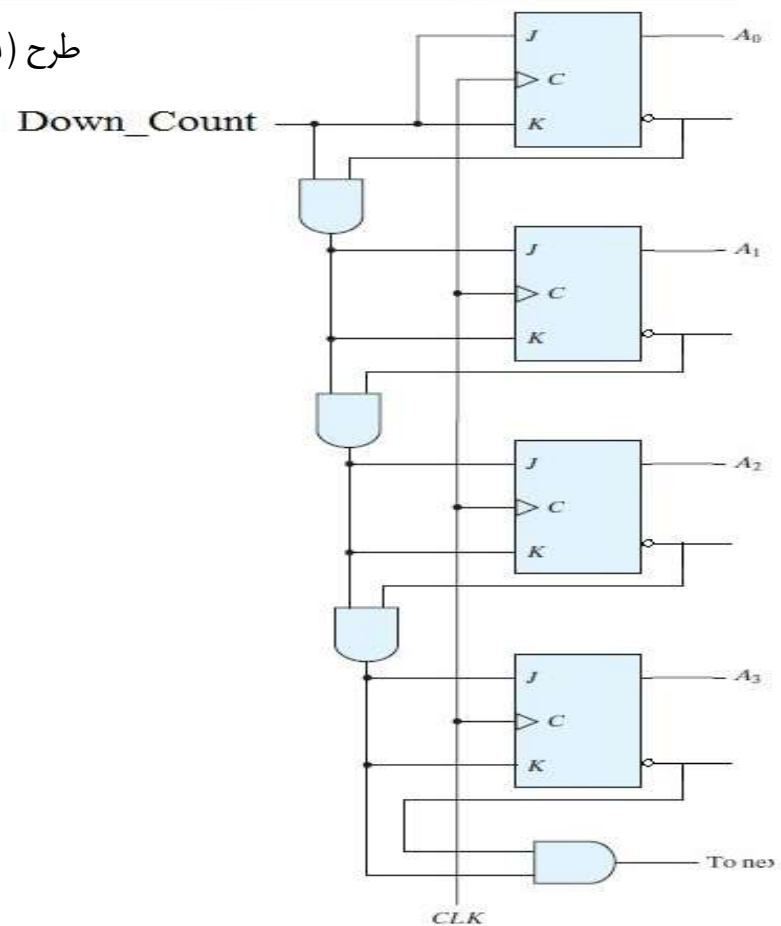
هر وقت  $A0 = A1 = 0$  باشد با لبه ی پالس A2 متمم می  
شود. (یعنی استفاده از فلیپ فلاپ JK با شرط  
 $J=K=A1'.A0' = (A1+A0)'$ )

هر وقت  $A2=A1=A0 = 0$  باشد با لبه ی پالس A3 متمم  
می شود. (یعنی استفاده از فلیپ فلاپ JK با شرط  
 $J=K=A2'A1'A0' = (A2+A1+A0)'$ )



## شمارنده ی سنکرون باینری کاهشی

طرح (۱)



طرح (۲)



## شمارنده ی سنکرون BCD

Present State				Next State				Output	Flip-Flop Inputs			
$Q_8$	$Q_4$	$Q_2$	$Q_1$	$Q_8$	$Q_4$	$Q_2$	$Q_1$	$y$	$TQ_8$	$TQ_4$	$TQ_2$	$TQ_1$
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

$$T_{Q1} = 1$$

$$T_{Q2} = Q_8'Q_1$$

$$T_{Q4} = Q_2Q_1$$

$$T_{Q8} = Q_8Q_1 + Q_4Q_2Q_1$$

$$y = Q_8Q_1$$

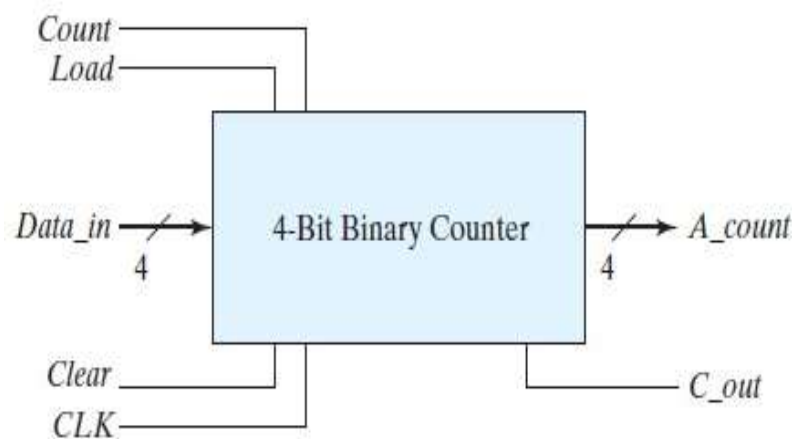
خروجی  $y$  هنگامی که حالت به 1001  
برسد برابر ۱ می شود. این خروجی می تواند  
شمارنده ی بعدی که در شمارش اعداد دو  
رقمی به کار می رود را فعال کند.



## شمارنده ی باینری با قابلیت بارگذاری

این مدار باید دو قابلیت شمارش و بارگذاری را فراهم کند. از دو ورودی کنترلی Count و Load برای شمارش و بارگذاری استفاده می شود. هنگامی که  $Clear = 0$  باشد مدار به صورت کامل ریست شده و نه بارگذاری می تواند انجام شود و نه شمارش. اگر ورودی Clear را غیر فعال کنیم ( $Clear = 1$ ) در لبه ی بالا رونده وضعیت Load و Count نحوه ی عملکرد مدار را طبق جدول زیر مشخص می کند.

اولویت با ورودی Load می باشد.

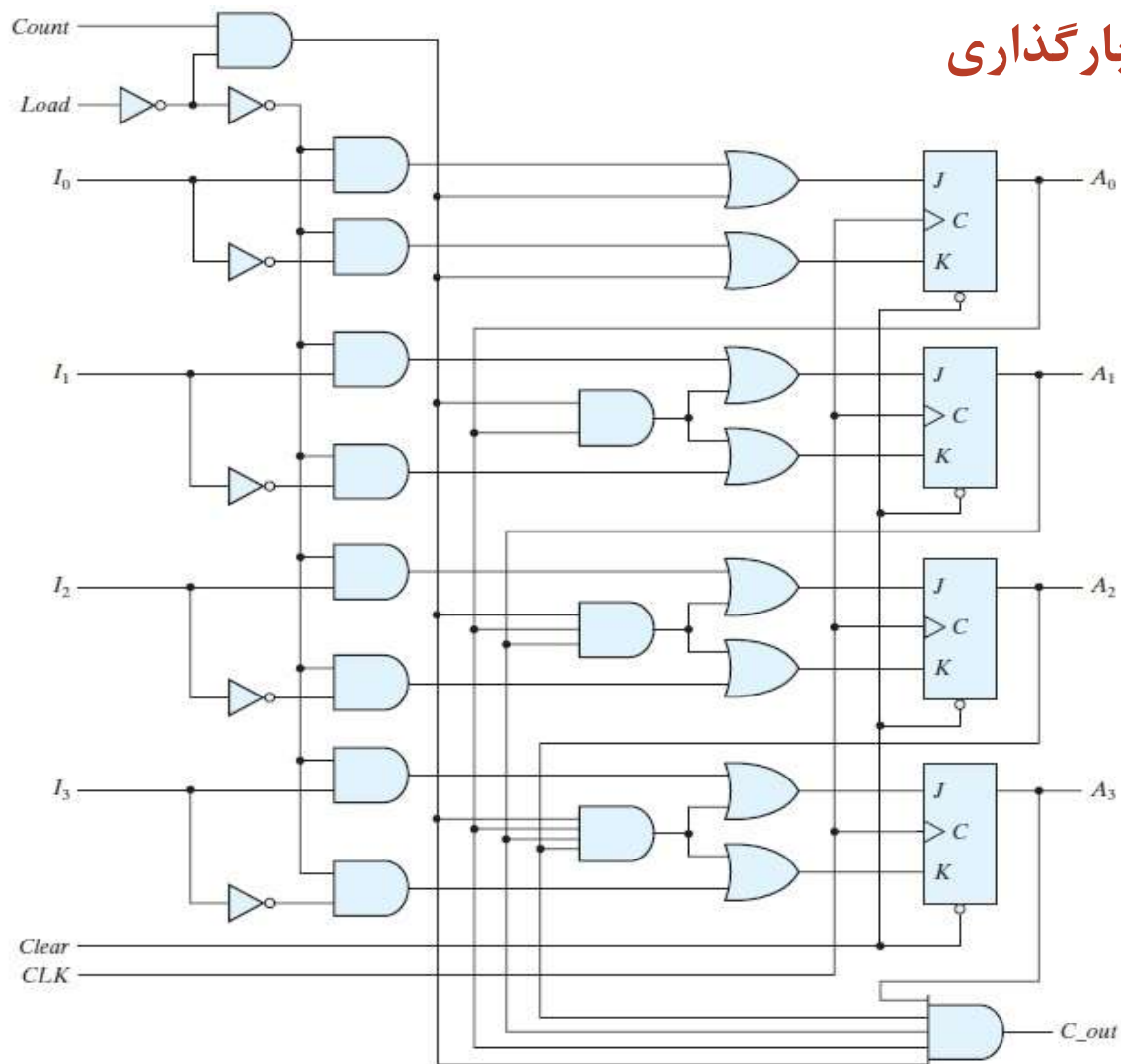


Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change



دانشگاه علم و صنعت ایران

## شمارنده ی باینری با قابلیت بارگذاری

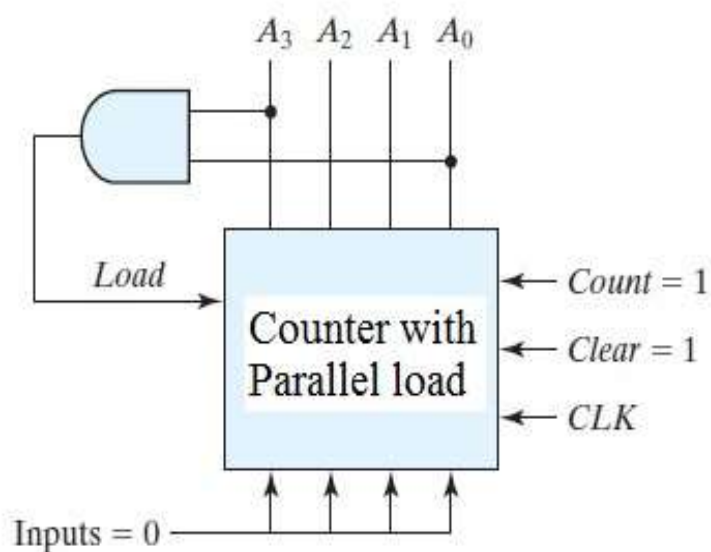






## شمارنده ی باینری با قابلیت بارگذاری

**مثال:** با استفاده از شمارنده ی باینری با قابلیت بارگذاری یک شمارنده ی BCD طراحی کنید.



ابتدا باید ورودی Clear غیر فعال شود تا مدار ریست نشود. بنابراین  $\text{Clear} = 1$  (چون Active – low می باشد)  
ورودی Count باید فعال شود تا در مد شمارش قرار گیرد. بنابراین  $\text{Count} = 1$

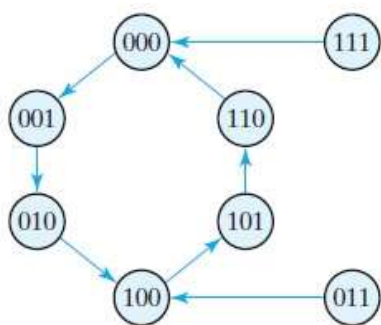
چون شمارش از 0000 شروع می شود ابتدا باید مقدار 0000 بارگذاری شود. این کار از طریق ورودی ها انجام می گیرد.  $\text{Inputs} = 0000$   
با اولین لبه ی بالا رونده  $A_3A_2A_1A_0 = 0000$  می شود. لبه ی بالا رونده ی دوم  $A_3A_2A_1A_0 = 0001$  و ... لبه ی بالا رونده ی نهم  $A_3A_2A_1A_0 = 1001$ . در این هنگام  $\text{Load} = A_3.A_0 = 1$  و با توجه به اینکه Load نسبت به Count در اولویت است مدار به مد بارگذاری می رود و چون  $\text{Inputs} = 0000$  است مدار از 1001 به 0000 می رسد و این روند ادامه می یابد.



## شمارنده ی باینری با حالات بی استفاده

یک شمارنده ی باینری  $n$  بیتی قادر به شمارش اعداد

باینری از 0 تا  $2^n - 1$  می باشد. (یا بالعکس) اما در بعضی شمارنده ها همه ی  $2^n$  حالت ها استفاده نمی شود. به عنوان مثال یک شمارنده ی باینری برای شمارش اعداد از  $0 = 000$  تا  $5 = 101$  را در نظر بگیرید. در این شمارنده حالات  $6 = 011$  و  $7 = 111$  حالات بی استفاده هستند و اگر مدار در این حالات قرار بگیرد با لبه ی بعدی کلاک به  $000$  بر می گردد. این حالات تحت عنوان حالات بی اهمیت نیز شناخته می شوند و در جدول حالت از آن ها استفاده نمی شود.



Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
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

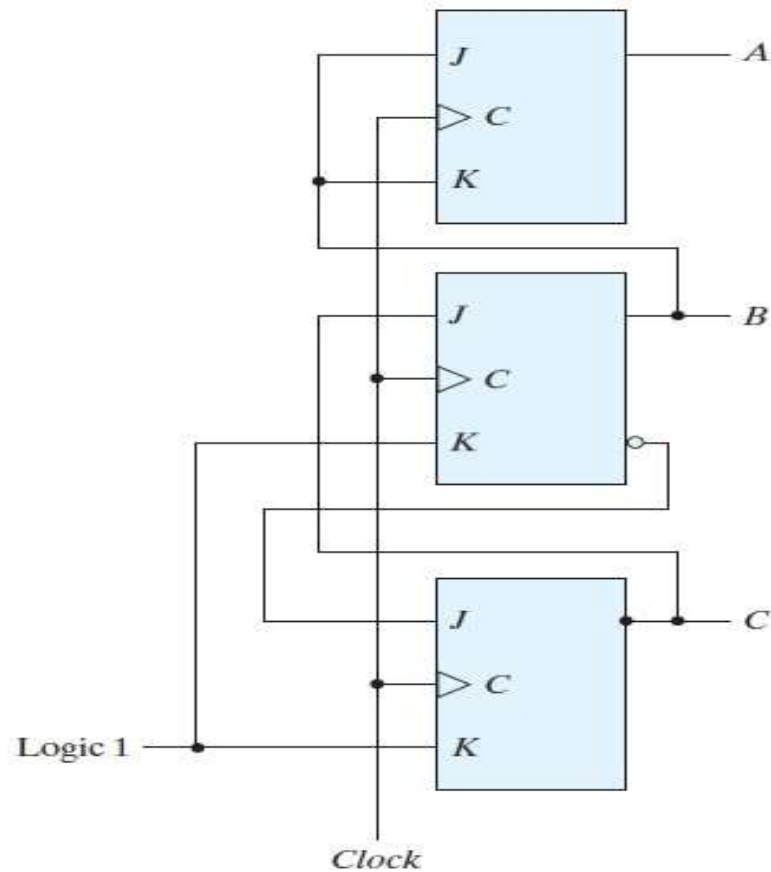
$$J_A = B \quad K_A = B$$

$$J_B = C \quad K_B = 1$$

$$J_C = B' \quad K_C = 1$$



## شمارنده ی باینری با حالات بی استفاده



$$J_A = B \quad K_A = B$$

$$J_B = C \quad K_B = 1$$

$$J_C = B' \quad K_C = 1$$



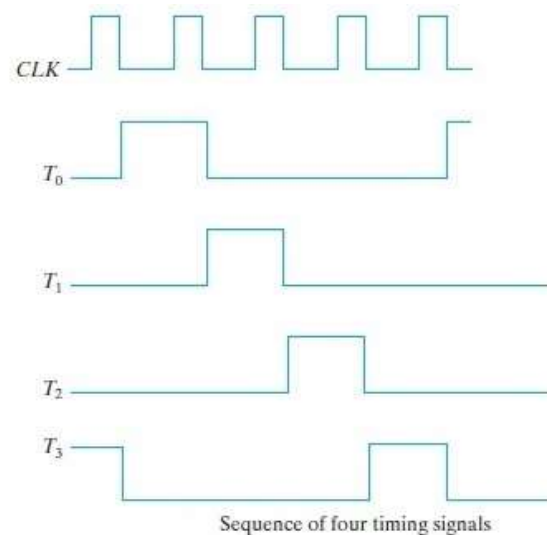
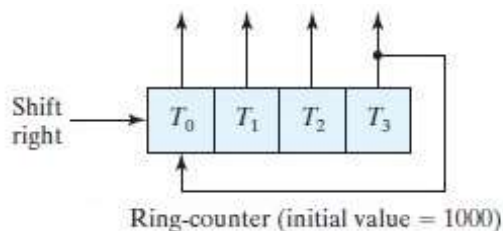
## شمارنده ی حلقه (Ring Counter)

سیگنال های زمان بندی که کنترل توالی عملکرد را در سیستم های دیجیتال به عهده دارند، می توانند توسط شیفت رجیسترها یا شمارنده های دیکدر ساخته شوند.

یک شمارنده ی حلقه یک شیفت رجیستر حلقوی (Circular) است که در هر زمان مشخص تنها یکی از فلیپ فلاپ های آن set می باشد و بقیه ی فلیپ فلاپ ها reset می باشند. یک بیت تکی از یک فلیپ فلاپ به فلیپ فلاپ دیگر منتقل می شود تا بدین ترتیب سیگنال های زمان بندی ساخته شوند. مقدار اولیه ی ثبات باید 1000 باشد و فلیپ فلاپ ها دارای پایه های Preset / Clear باشند.

در مدار زیر با هر لبه ی پایین رونده تنها یک فلیپ فلاپ set می شود و تا لبه ی بعدی 1 می ماند، بعد از آن 0 شده و به مدت ۳ پالس ساعت در 0 باقی می ماند.

لبه ی منفی اول	1	0	0	0
لبه ی منفی دوم	0	1	0	0
لبه ی منفی سوم	0	0	1	0
لبه ی منفی چهارم	0	0	0	1
لبه ی منفی پنجم	1	0	0	0



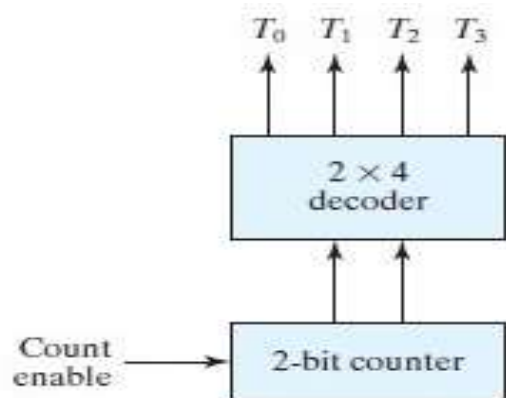


## شمارنده ی حلقه (Ring Counter)

برای ساخت  $2^n$  سیگنال زمان بندی دو راه وجود دارد:

(1) استفاده از شیفت رجیستر حلقوی  $2^n$  بیتی.

(2) استفاده از شمارنده ی  $n$  بیتی به همراه دیکدر  $n:2^n$



راه دیگر برای ساخت مدار قبلی استفاده از شمارنده و دیکدر می باشد. با توجه به اینکه به ۴ خروجی احتیاج داریم از دیکدر 2:4 استفاده می شود. بعد از فعال سازی شمارنده ی ۲ بیتی و با رسیدن اولین لبه ی حساس مقدار اولیه شمارنده 00 می شود، که این مقدار به دیکدر رسیده و داریم:

$$T_3T_2T_1T_0 = 0001$$

با لبه ی حساس بعدی مقدار شمارنده به 01 رسیده، این مقدار به دیکدر رسیده و داریم:

$$T_3T_2T_1T_0 = 00100$$

و این روند ادامه می یابد.

Count_enable	Clk	2-bit Counter	T0	T1	T2	T3
0	X	00	1	0	0	0
1		01	0	1	0	0
1		10	0	0	1	0
1		11	0	0	0	1
1		00	1	0	0	0

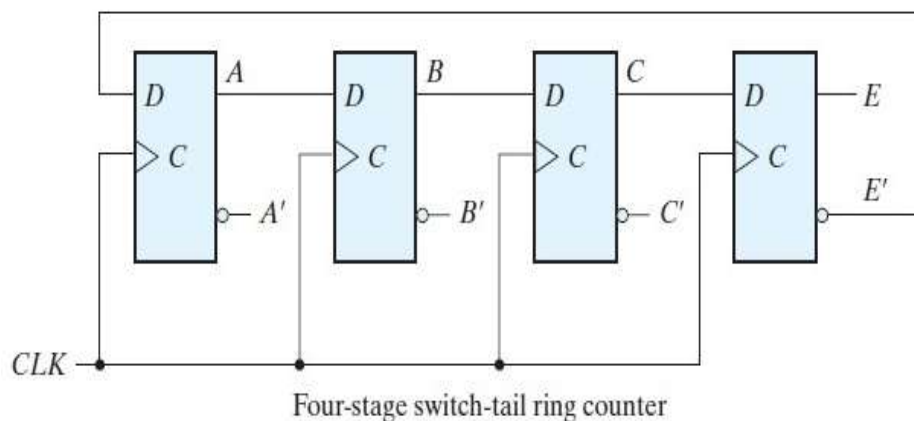


## شمارنده ی حلقه (Ring Counter)

در شمارنده ی حلقه ی ۴ بیتی مشاهده کردیم که ۴ حالت وجود دارد:  
0001، 0010، 0100، 1000.

در شمارنده ی حلقه ی switch tail امکان دو برابر کردن حالات فراهم شده است. در این مدار خروجی متمم آخرین فلیپ فلاپ به ورودی اولین فلیپ فلاپ وصل می شود و توالی زیر را با لبه های متوالی طی خواهد کرد.

وضعیت اولیه 0000 می باشد در این حالت  $DA = E' = 1$  می باشد. با لبه ی بالارونده ی بعدی  $A=1$  می شود و  $BCE = 000$ ، در این حالت  $DA = E' = 1$  و  $DB = A = 1$  می باشد. با لبه ی بعدی داریم  $AB = 11$  و  $CE = 00$ . این روند ادامه می یابد همان طور که مشاهده می شود این شمارنده با استفاده از ۴ فلیپ فلاپ ۸ حالت مختلف ایجاد کرد.

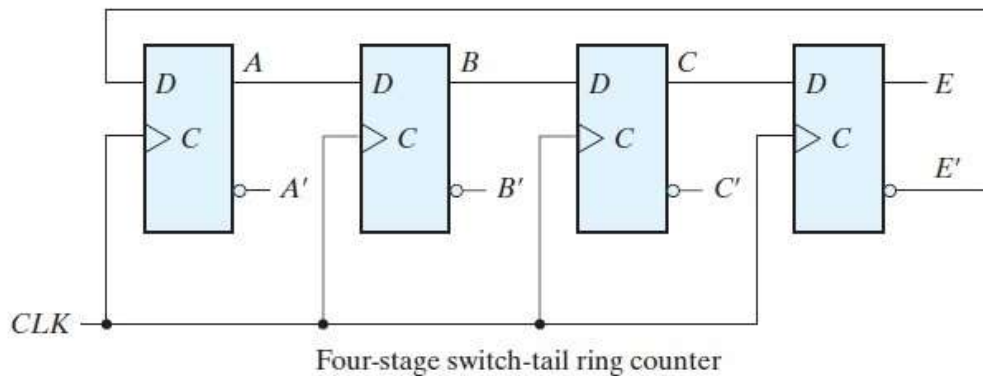


Sequence number	Flip-flop outputs			
	A	B	C	E
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1



## شمارنده ی جانسون (Johnson Counter)

مدار شمارنده ی حلقوی Switch tail ۴ بیتی قادر به ایجاد ۸ حالت مختلف بود. می توان از این ۸ حالت برای ایجاد ۸ شکل موج زمان بندی استفاده کرد. خروجی فلیپ فلاپ ها را باید مطابق جدول زیر با هم AND کرد. به این نوع شمارنده (با احتساب گیت ۸ گیت AND در خروجی فلیپ فلاپ ها) شمارنده ی جانسون گویند.

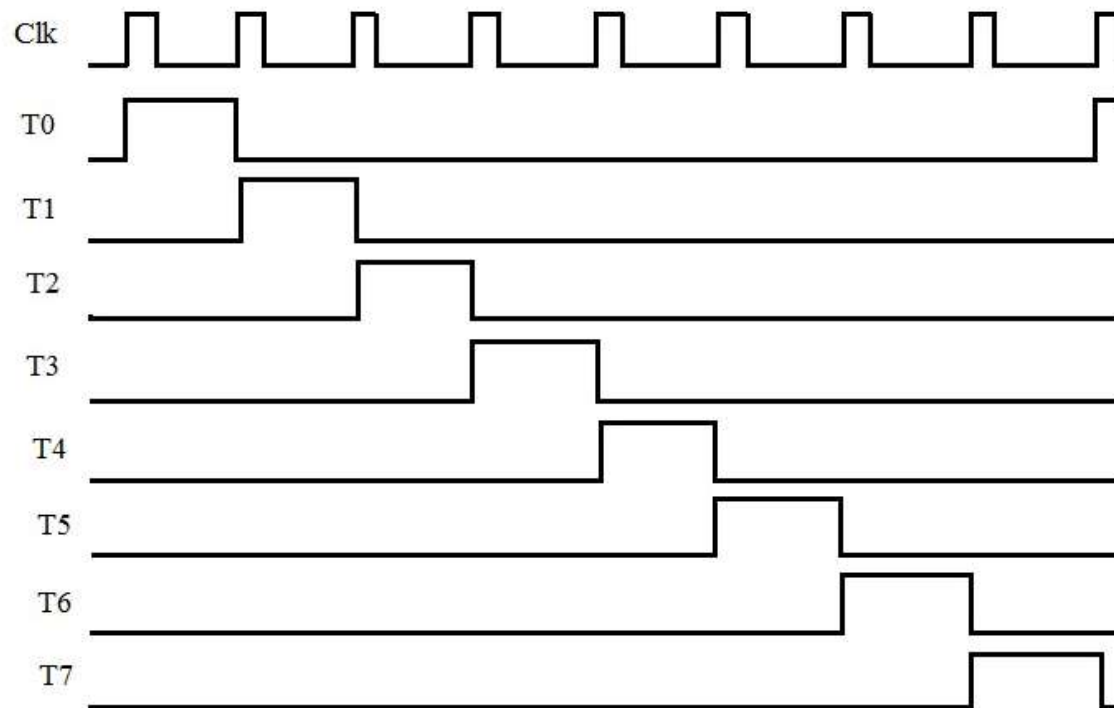


Sequence number	Flip-flop outputs				AND gate required for output
	A	B	C	E	
1	0	0	0	0	T0 = $A'E'$
2	1	0	0	0	T1 = $AB'$
3	1	1	0	0	T2 = $BC'$
4	1	1	1	0	T3 = $CE'$
5	1	1	1	1	T4 = $AE$
6	0	1	1	1	T5 = $A'B$
7	0	0	1	1	T6 = $B'C$
8	0	0	0	1	T7 = $C'E$

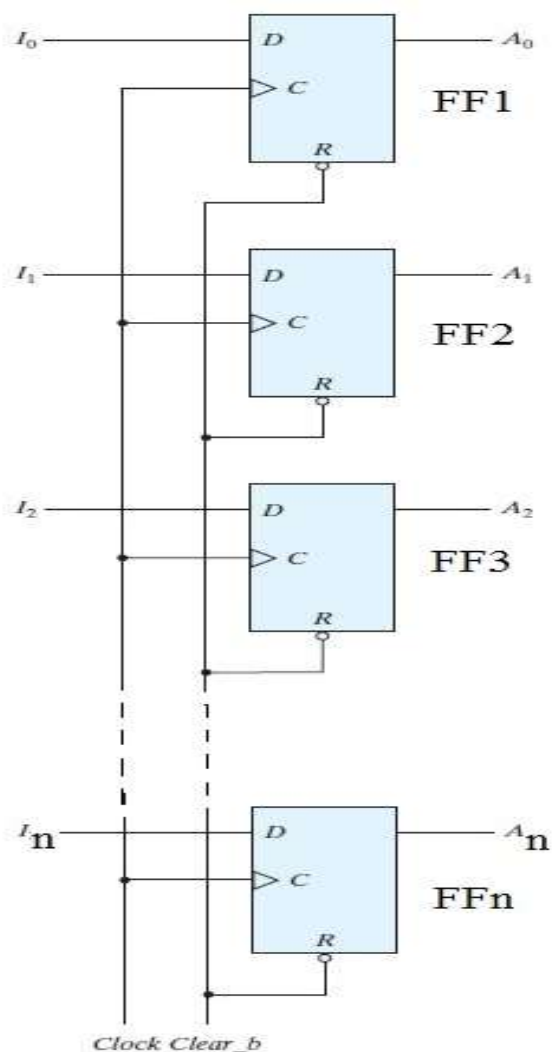


## شمارنده ی جانسون (Johnson Counter)

## شکل موج ها برای شمارنده ی جانسون

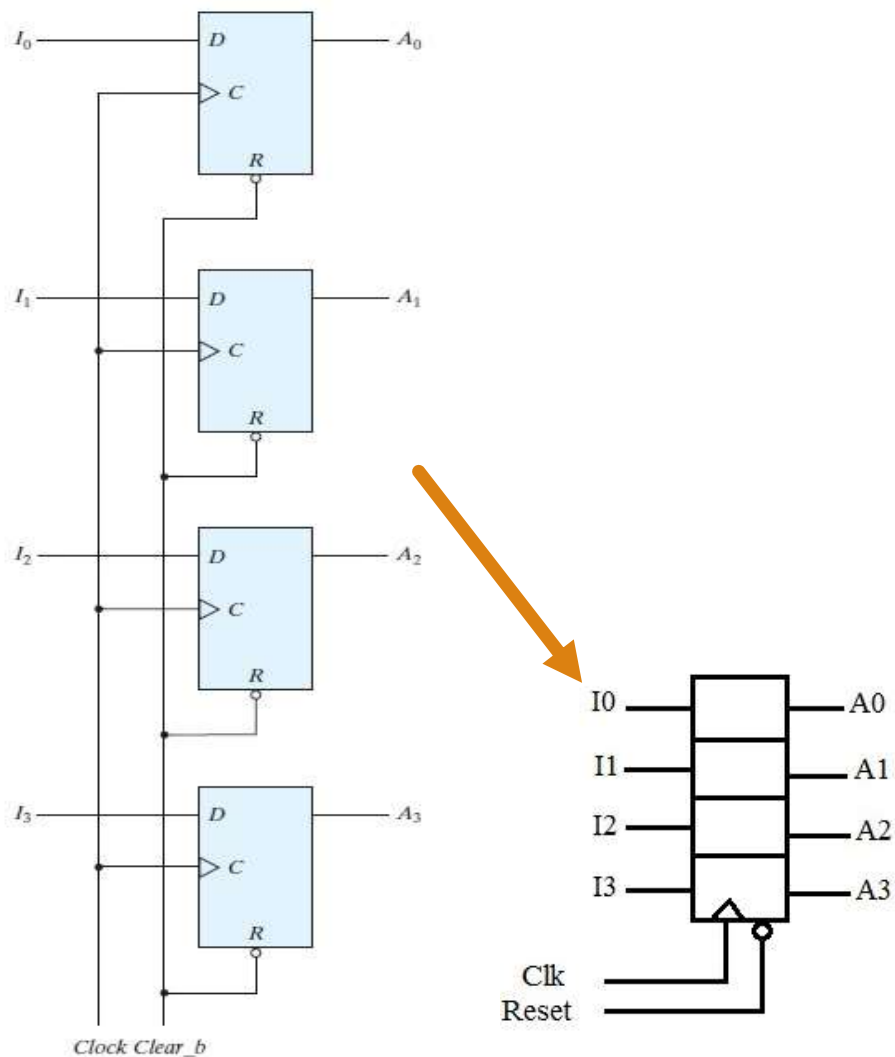






## ثبات n بیتی

ساده ترین نوع ثبات n بیتی تنها از n فلیپ فلاپ ساخته شده است که همگی دارای ورودی های کلاک و ریست سنکرون می باشند. مقدار ثبات در هر لحظه یک عدد n بیتی به صورت  $A_n \dots A_2 A_1 A_0$  می باشد. با لبه ی بالا رونده کلاک مقدار  $I_n \dots I_2 I_1 I_0$  در ثبات بارگذاری می شود.

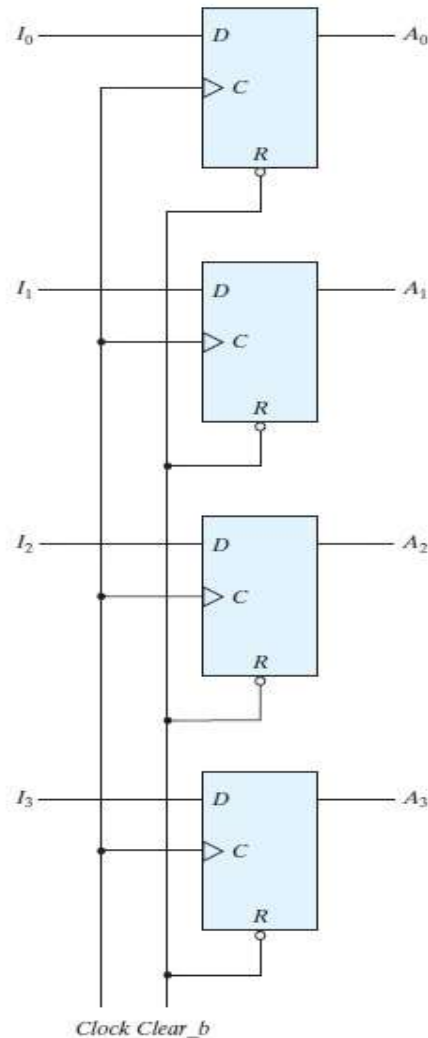


## ثبات با بارگذاری موازی

انتقال اطلاعات جدید به ثبات را بارگذاری (Loading) یا به روز رسانی (Updating) گویند.

برای بارگذاری ثبات لازم است، فلیپ فلاپ های آن تک تک بارگذاری شوند، چراکه مقدار ذخیره شده در ثبات، از کنار هم قرار گرفتن مقادیر فعلی فلیپ فلاپ ها در هر لحظه حاصل می شود.

اگر تمامی فلیپ فلاپ ها هم زمان و با یک پالس ساعت بارگذاری شوند، آنگاه ثبات به صورت موازی بارگذاری شده است.

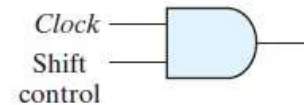


## ثبات با بارگذاری موازی

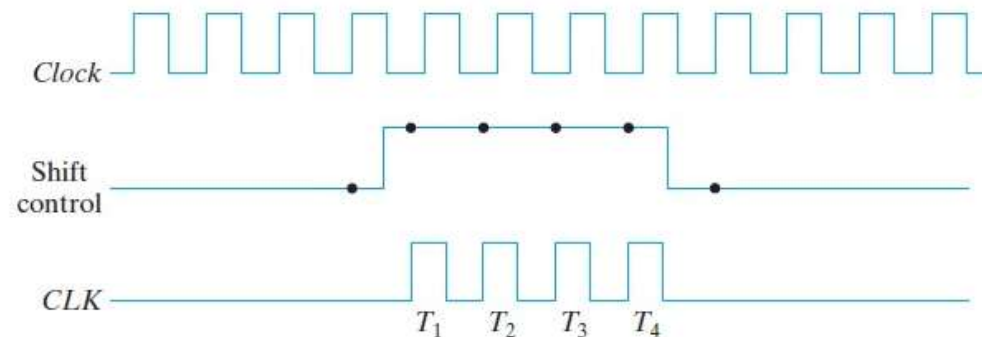
در ثبات روبرو برای حفظ مقدار ثبات می توان:

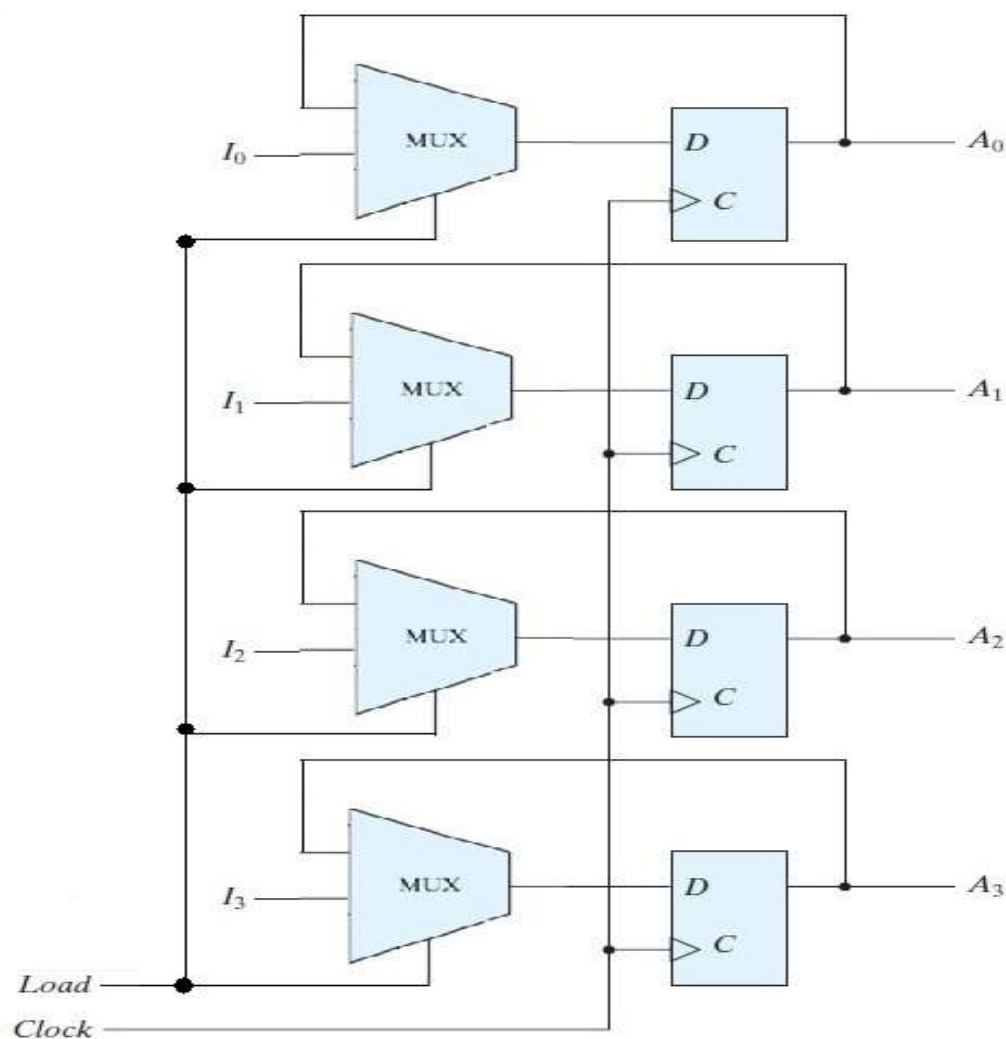
۱- مقادیر  $I_0, I_1, I_2$  و  $I_3$  را ثابت نگه داشت.

۲- کلاک را کنترل کرد که تنها در بازه ی زمانی خاصی به فلیپ فلاپ ها اعمال شود. این کار را می توان با استفاده از گیت AND و به صورت زیر انجام داد.



استفاده از این روش یک عیب عمده دارد و آن این که گیت AND به کار برده شده مقداری تاخیر دارد (ایده آل نیست) و این موجب می شود سیستم سنکرون کامل نباشد. راه حل این مشکل کنترل ورودی فلیپ فلاپ ها می باشد.





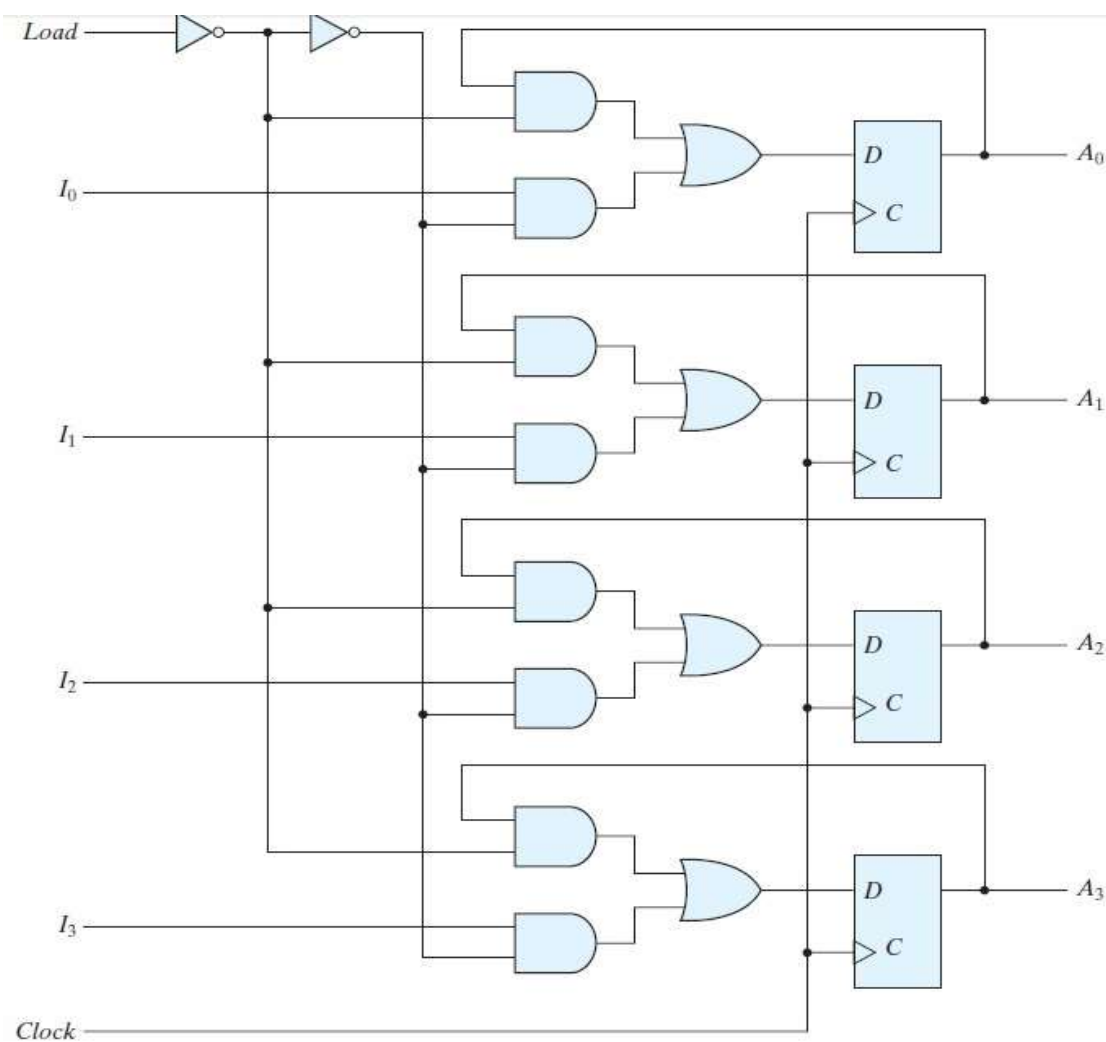
## ثبات با بارگذاری موازی

به جای کنترل روی کلاک می توان کنترل را روی داده انجام داد. به این ترتیب که مطابق مدار روبرو و با استفاده از مالتی پلکسر بارگذاری یا حفظ مقدار ثبات را کنترل کرد.

در مدار روبرو اگر  $Load = 0$  باشد با لبه ی بالا رونده تغییری در مقدار ثبات حاصل نمی شود. برای اینکه امکان بارگذاری اطلاعات جدید فراهم شود باید  $Load = 1$  شود. به این ترتیب در لبه ی بالا رونده مقدار  $I_n \dots I_2 I_1 I_0$  در ثبات بارگذاری می شود.



دانشگاه علم و صنعت ایران



## ثبات با بارگذاری موازی

مدار قبل را می توان با گیت های منطقی (که معادل مالتی پلکسر هستند) به صورت روبرو نیز پیاده سازی کرد.