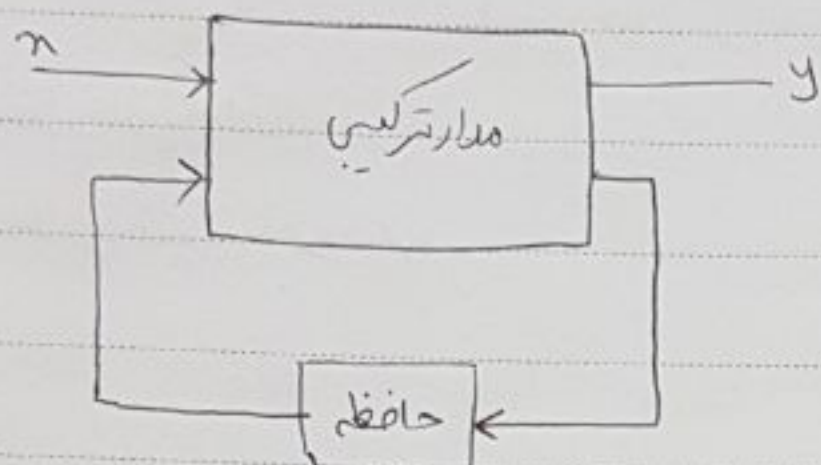


## مدارهای ترتیبی sequential circuits

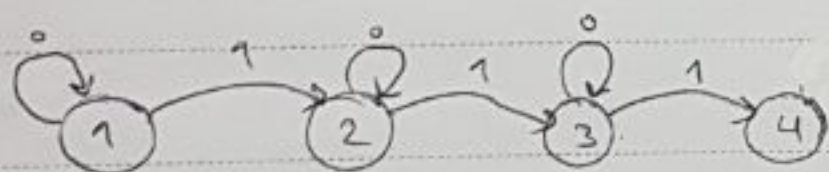
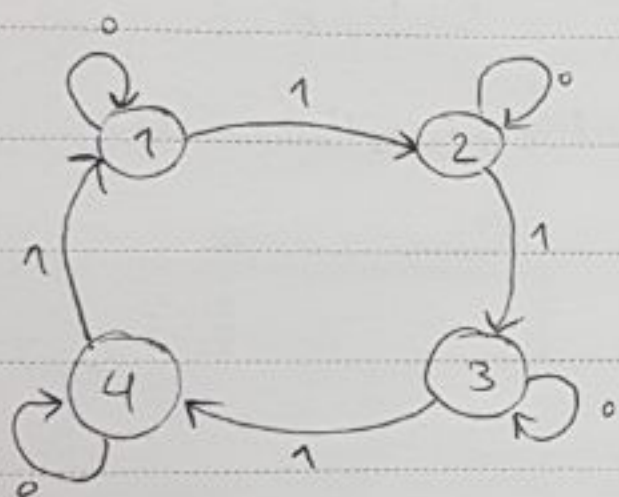
تا بحال مدارهای ترکیبی (combinational circuits) را بررسی نمودیم که در آنجا خروجی مدار به ورودی لحظه ای وابسته می باشد  
مدارهای ترتیبی دارای حافظه هستند که با گذر آن می توانند اطلاعات ورودی و اطلاعات قبلی را به صورت یکپارچه بررسی نمایند  
به عبارتی ترتیب اعمال ورودی (ها) در نتیجه خروجی موثر خواهد بود



اطلاعات پردازش شده قبلی توسط مدار ترکیبی در حافظه ذخیره شده  
و سپس در لحظه های بعدی مجدداً به مدار ترکیبی اعمال شده است

### مثال: مدارهای شمارنده

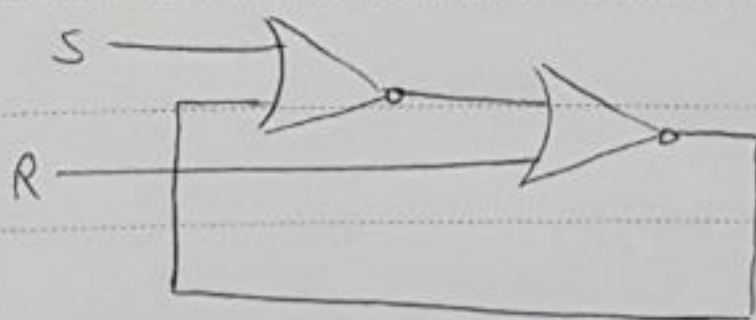
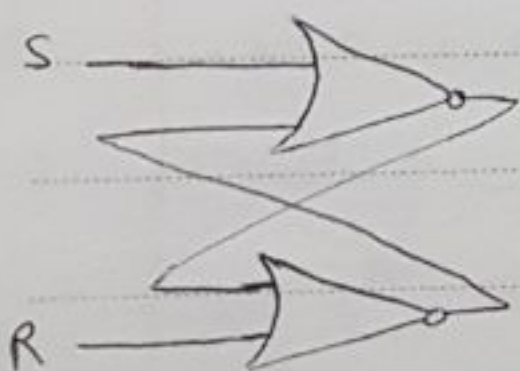
#### ماشین حالت



مدارهایی که بست شده اند بعد از گذشت زمانی کاری انجام دهد

مثال بچ ها

**فلیپ فلوپ** مشاهده می خروجی و اعمال تا شیر یا راهبرهای آن مجدداً در ورودی  
در مدارهای دیجیتال طراحی مناسب فلیپ فلوپ می تواند باعث ایجاد حافظه شود





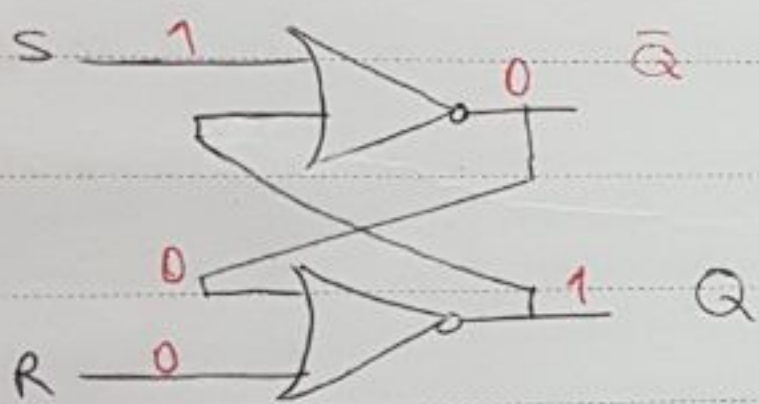
در مدار صفحه قبل حلقه سیگنال بین ورودی و خروجی وجود دارد  
به شکلی که اطلاعات خروجی به سمت ورودی بازگشته و در گیت‌ها تأثیر خواهد گذاشت

به آن مدار SR Latch ایچ SR گفته می‌شود که ساده‌ترین و پایه‌ای‌ترین عنصر حافظه است

Set	Reset	$Q^*$	
1	0	1	Set
0	1	0	Reset
0	0	Q	Memory (Hold)
1	1	غیر مجاز	

$Q^*$ : مقدار جدید حافظه

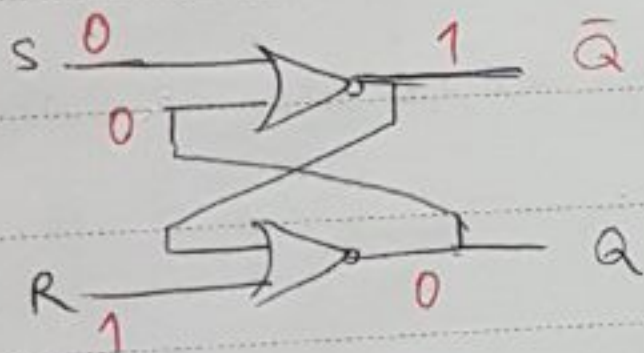
Q: مقدار ذخیره شده قبلی



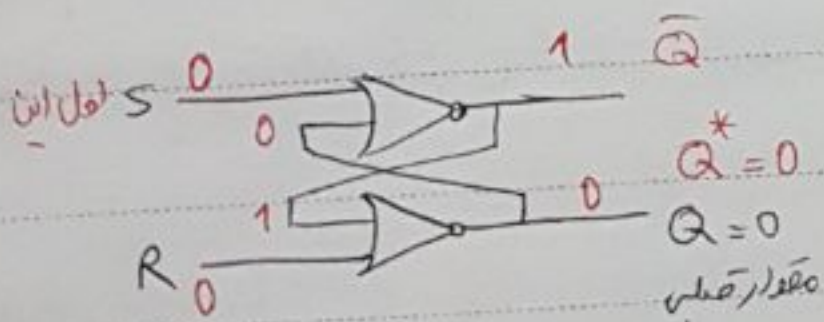
حالت Set  $S=1 / R=0$

از جایی شروع می‌کنیم که یکی از ورودی‌ها 1 است چون می‌توان نتیجه را حدس زد

حالت Reset

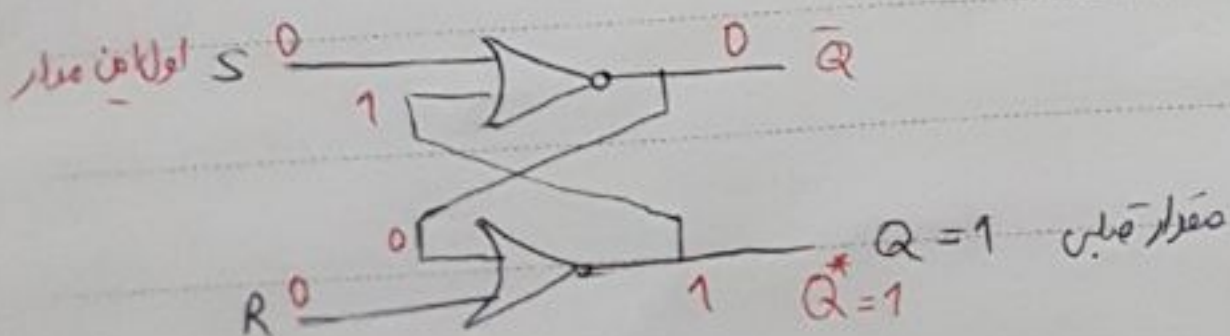


حالت Hold



$Q=0$ : مقدار قبلی حافظه

$Q=1$

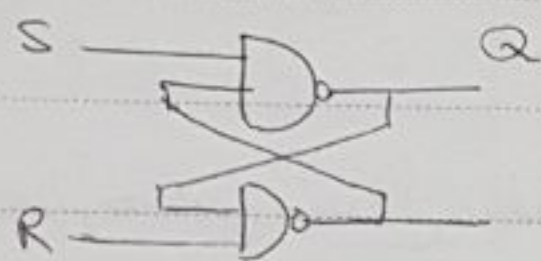




هنگامی که  $S = R = 1$  و در عمل به معنای زمان Set و Reset هر زمان سیستم می باشد:

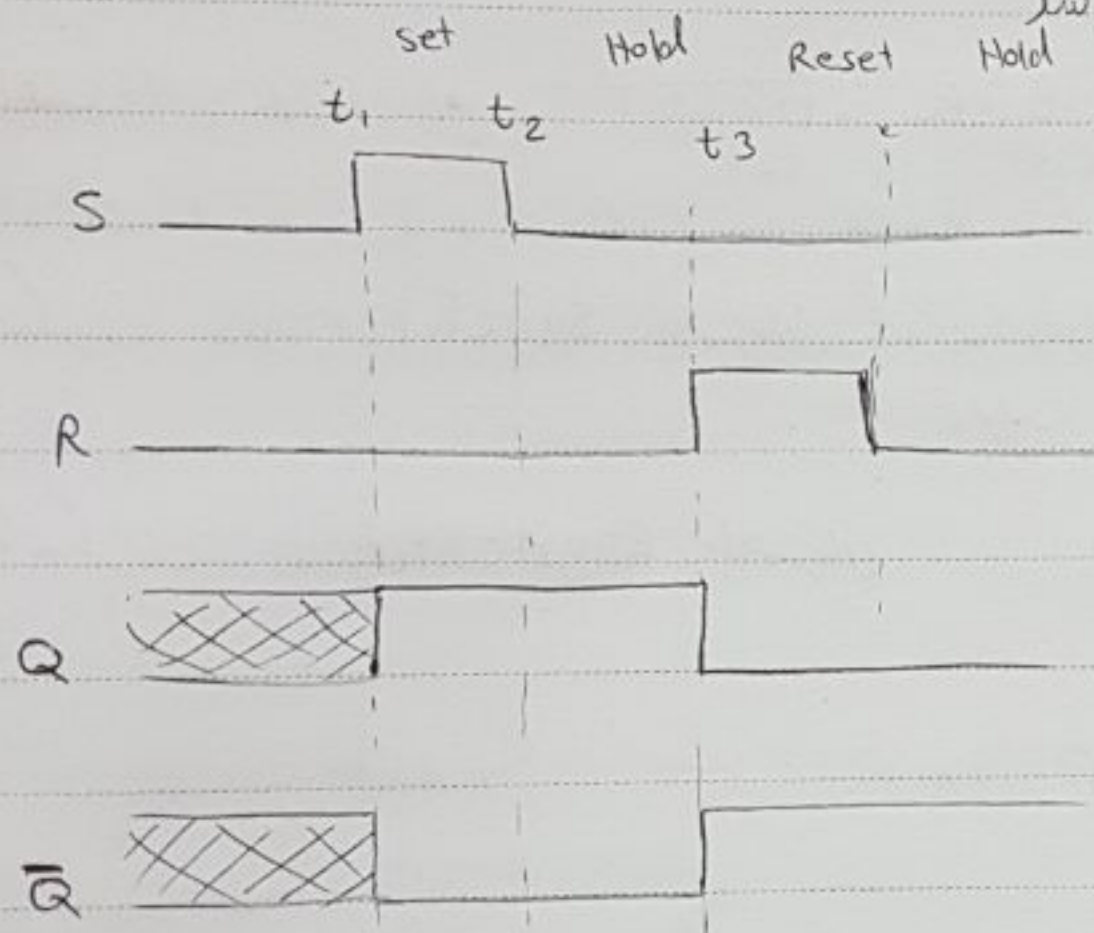
تاخیر لیتها بسیار مهم خواهد بود و وابسته به اختلاف تاخیر لیتها احتمال ناپایداری مدار و خروجی بسیار زیاد است

در استفاده از لچ SR طراحی با بیسی از این اتفاق جلوگیری نماید



لچ SR را می توان با نند هم پیاده سازی کرد با این تفاوت که خروجی نند سیستم که ورودی دارد

با توجه به نتیجه ها در حالت های مختلف خروجی دوم  $\bar{Q}$  می باشد



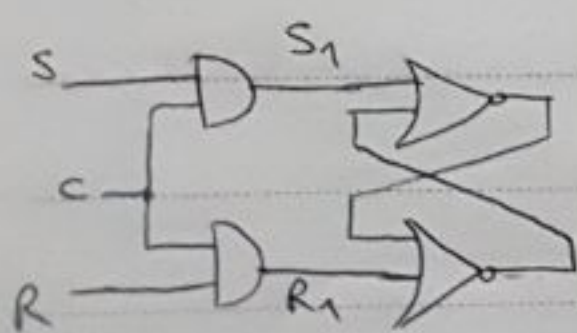
مثال:

SR Flip-Flop بیت کنترلی C

$$f = 20 \text{ MHz} \rightarrow T = \frac{1}{20 \text{ M}} = 50 \text{ ns}$$

$f = \frac{1}{T}$  فراکانس یعنی تعداد دورها در ثانیه

فراکانس clock در CPU هم قرار دارد که باعث ایجاد نظم داخلی آن می شود



با کمک بیت کنترلی C در لچ SR می توان اعمال ورودی به مدار را کنترل نمود

وقتی  $C = 0$  در حالت Hold قرار می گیریم که مدار را متوقف کرده



توقتی  $c=0$  :  $S_1=R_1=0$  و اچ در حالت Hold است به عبارتی مدار در حالت حافظه کار می کند و ورودی ها تا  $S$  و  $R$  بی تاثیرند

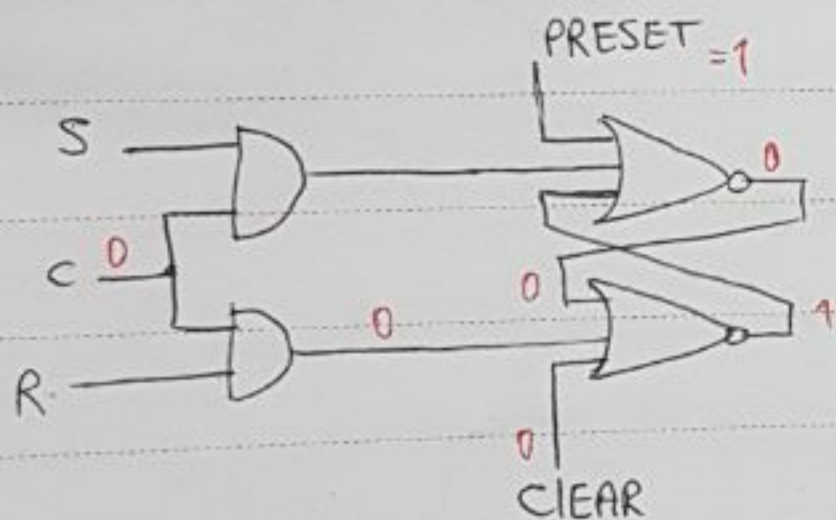
اگر  $c=1$  در این صورت  $S_1=S$  و  $R_1=R$  و عملیات مورد نظر ورودی انجام خواهد شد

این مدار حساس به سطح منطق ورودی کنترل است  
در ادامه خواهیم دید که برای نظم بهتر مدارهای حساس به لبه طراحی خواهند شد

فلیپ فلاپ SR با قابلیت PRESET و CLEAR  
set reset

سکرون Synchronous همزمان ← که در مدارهایی است که منقظر اجازه ی  $clk$  ساعت است برای انجام عملیات

آسکرون Asynchronous نامهمزمان



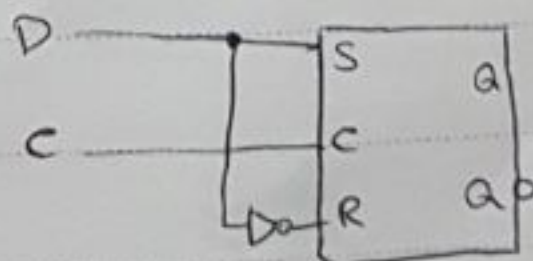
ورودی  $S$  و  $R$  عملیات Set و Reset سکرون انجام می دهند

ورودی های PRESET و CLEAR عملیات Set و Reset آسکرون انجام می دهند

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

C	$Q^*$
0	Q
1	D

وقتی ارتباط وصل است فقط D را به خروجی که Q است انتقال می دهیم



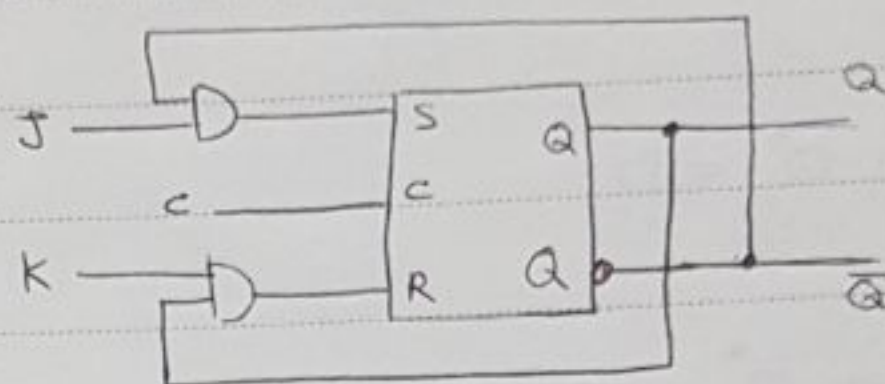
Delay Flip Flup DFF



تنها از حالت های Set و Reset در SRFF استفاده شده است

JK FF

C	J	K	$Q^*$	
0	x	x	Q	Hold
1	0	0	Q	Hold
1	1	0	1	Set
1	0	1	0	Reset
1	1	1	$\bar{Q}$	toggle

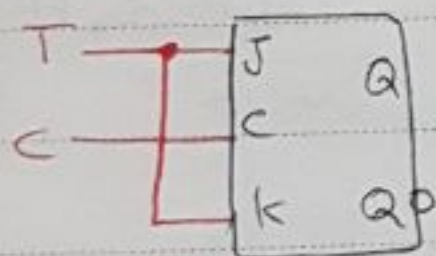


1: C=1 : J = K = 0  $\Rightarrow$  S = R = 0 Hold

J = 1, K = 0  $\Rightarrow$  S =  $\bar{Q}$  (مستقیم معکوس قبلی) R = 0  $\left\{ \begin{array}{l} Q=0, \bar{Q}=1 \quad Q^*=1 \quad SR=Set \\ Q=1, \bar{Q}=0 \quad Q^*=Q=1 \quad SR=Set \end{array} \right.$

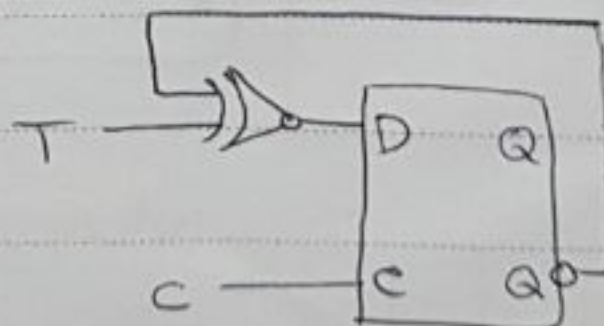
J = 0, K = 1

J = K = 1  $\Rightarrow$  S =  $\bar{Q}$ , R = Q  $\left\{ \begin{array}{l} Q=0 \quad S=1, R=0 \quad SR=Set \quad Q^*=1=\bar{Q} \\ Q=1 \quad S=0, R=1 \quad SR=Reset \quad Q^*=0=\bar{Q} \end{array} \right.$



C	T	$Q^*$
0	x	Q
1	0	Q
1	1	$\bar{Q}$

(Toggle Flip Flop) TFF



مبارد دیگر TFF  $\leftarrow$  بررسی نشود



روابط جبری بین ورودی‌های SRFF ها

$$SRFF: Q^* = \bar{S}\bar{R}Q + \underbrace{\bar{S}\bar{R}}_{\bar{R}Q} + \bar{S}R = \bar{R}Q + S$$

$$DFF: Q^* = D$$

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

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

فلپ فلاپ‌های حساس به لبه Edge Triggered FF

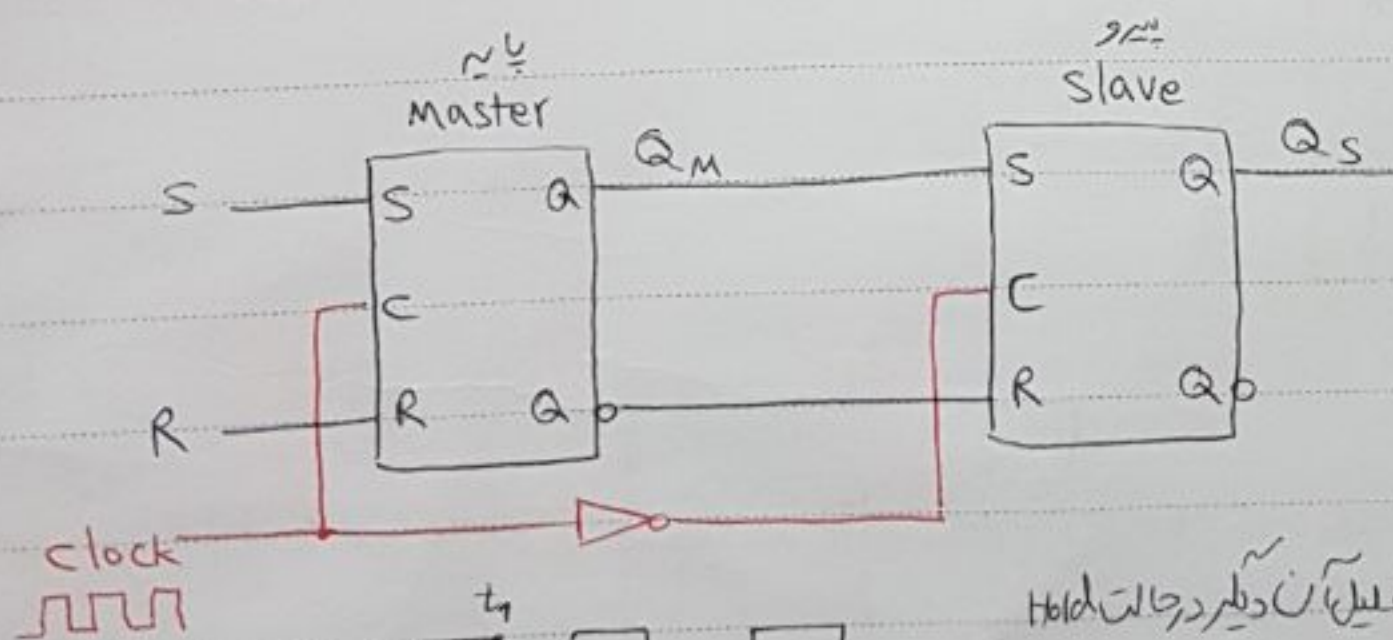
Master/slave

طراحی مستقل

SRFF حساس به لبه بارش یا لبه پیرو

FF های حساس به سطح عمل کار بردن ندارند چون مثلاً در حالت  $T=1$  و  $C=1$  در TFF های مقولر Q تغییر می کنند

حالت قلبی



$t_1$ : آخرین ورودی تحویل به Master و پس از تحلیل آن دیگر در حالت Hold

باقی می ماند ( $t_1$  لبه پایین رونده است)

تراشه Slave در بازه  $[t_1, \dots]$  شروع به دریافت ورودی می نماید که

حاصل از تراشه Master است و متناظر با ورودی اصلی در زمان  $t_1$

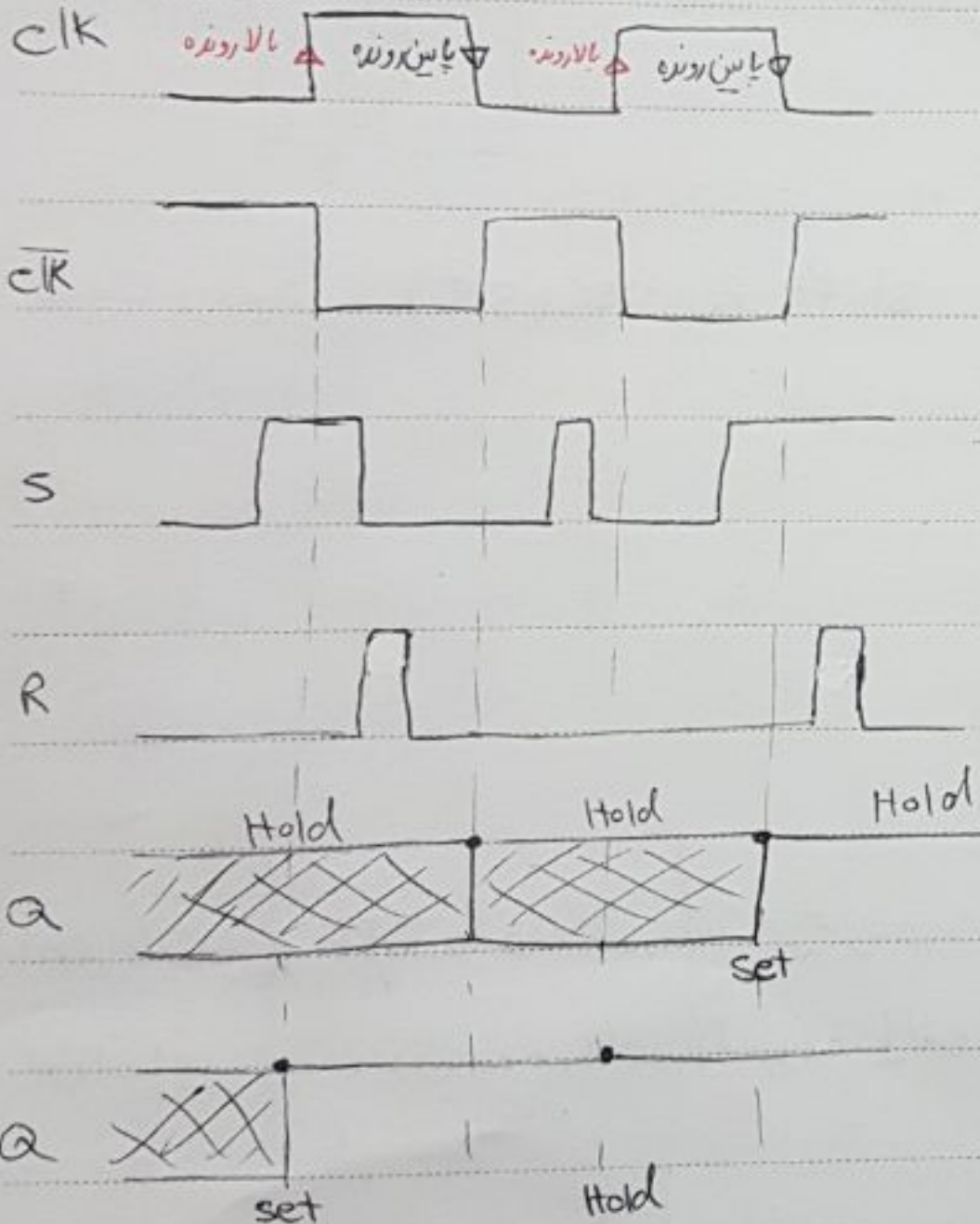
Master does not let input affecting

Slave is on Hold



در نهایت خروجی این سیستم تنها به مقادیر ورودی در لبه های پایین رونده مرتبط خواهد بود  
به راحتی می توان با جابجایی Not در مسیر پالس ساعت یا لبه پایین رونده را به بالا رونده تفسیر داد

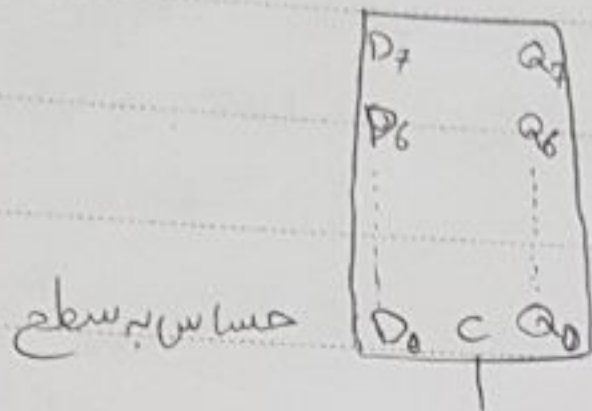
clk	S	R	Q*
0	x	x	Q
1	x	x	Q
↓ = ]	0	0	Q
]	0	1	0
	1	0	1



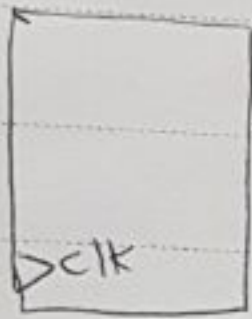


مدارهای ترکیبی مایکروار

رجیستر Register حافظه کوتاه مدت ساده



مثلاً: فلیپ فلاپ‌ها D که در مثال فوق با رجیستر با کمک 8 عدد DFF در حافظه آنها ذخیره شده است (از ذخیره می‌کنند تا همی 8 بیت پر شود)



حساس به لبه مثبت

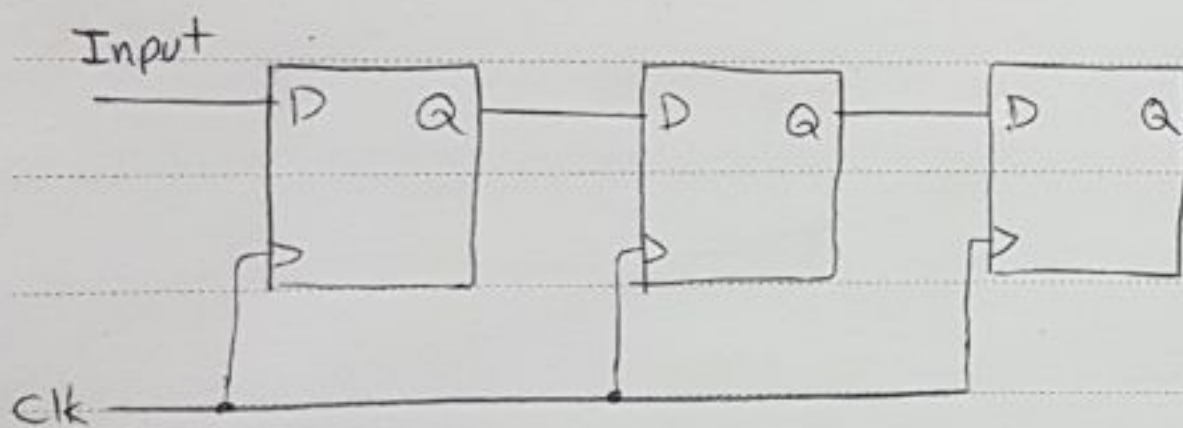


حساس به لبه منفی



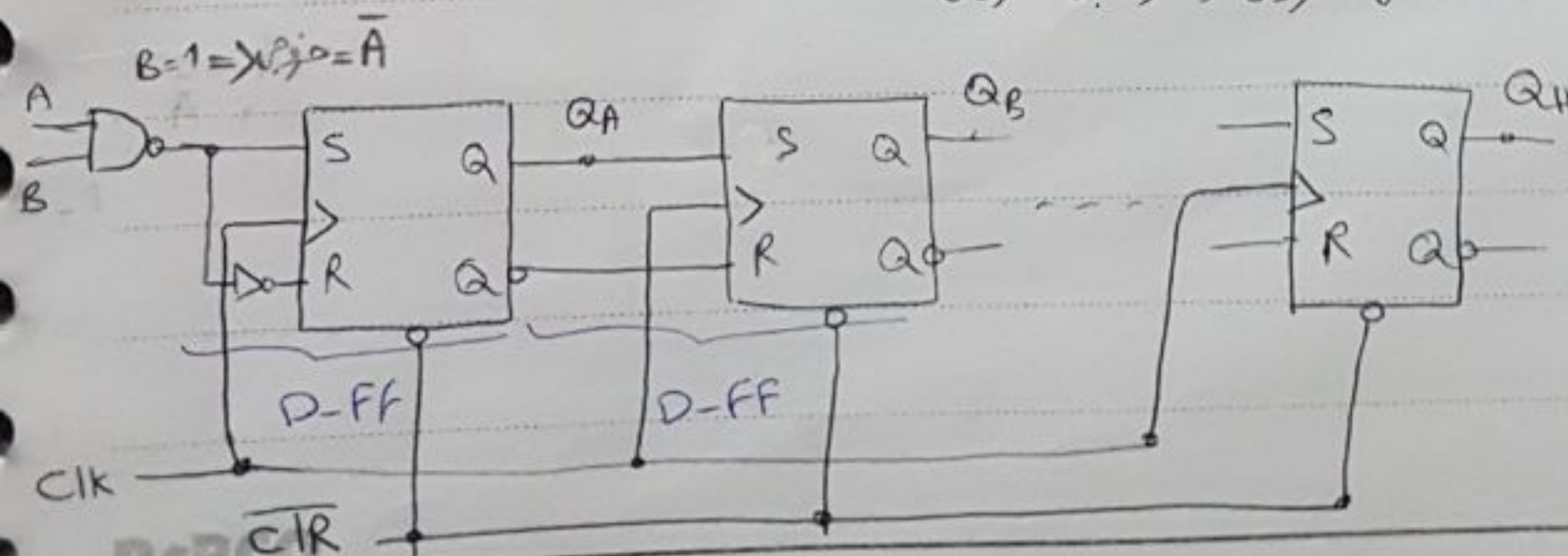
حساس به لبه مثبت

شیفت رجیستر Shift Register (SR)



اطلاعات بین DFF ها دست به دست می‌شود و به صورت ایده آل از آخرین FF خارج می‌شود  
این اتفاق در مثال فوق در لبه‌های بالا رونده یا لبه ساعت رخ می‌دهد

تراشه 7491 SR هفت بیتی با ورودی سریال و خروجی سریال



CLR آسکرون پایین / باعث صفر شدن بدون نیاز به لبه و clk ها

کاره تاخیر 8 تا لبه ساعت

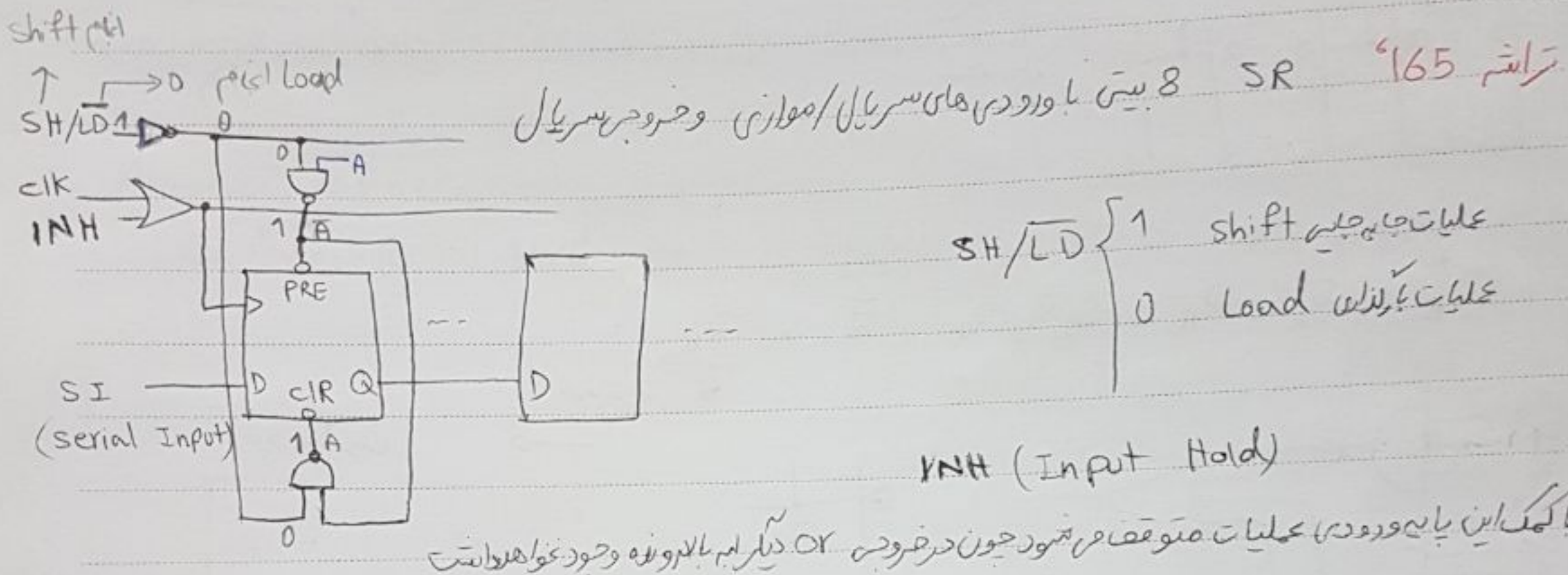
خروجی فقط QH است

Set کردن به صورت آسکرون



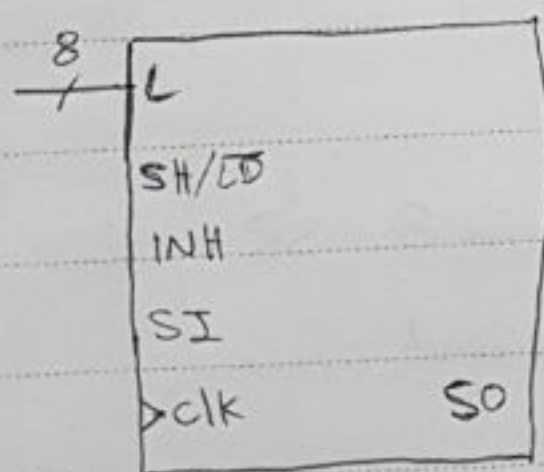
CIR در صورت آنکه خروجی تمام خروجی ها را صفر می نماید  
B عملیات set (یک خروجی ها) را به صورت سنکرون پس از 8 بایس ساعت انجام می دهد. منطق متغیر

خروجی A  $B=1 \Rightarrow \bar{A}$   
 $B=0 \Rightarrow A \cdot B = 1$



هر دو ورودی CIR و PRE غیر فعال خواهند بود  $\Rightarrow$  عملیات shift سنکرون رخ می دهد  $SH/LD = 1$  Shift 8

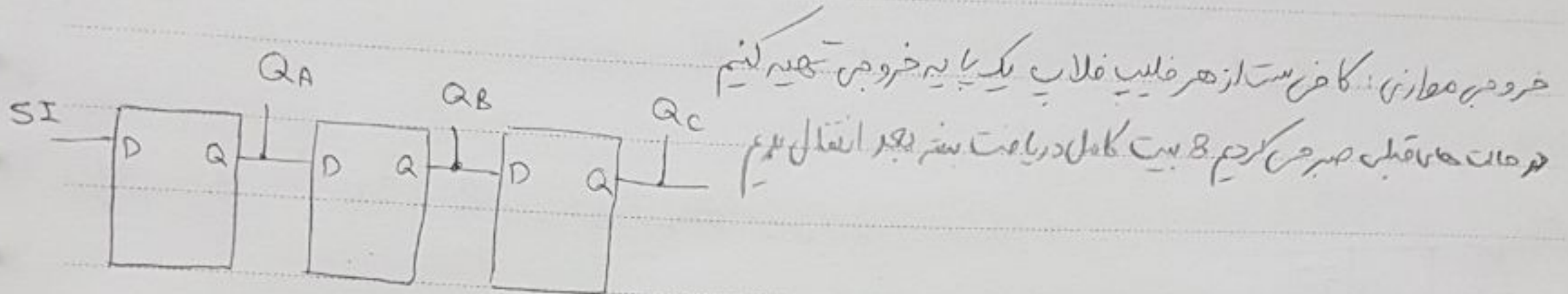
Load:  $SH/LD = 0$   $A=0$   $CIR=0$  فعال  $\Rightarrow Q=0=A$   
 $A=1$   $PRE=0$  فعال  $\Rightarrow Q=1=A$  }  $Q=A$



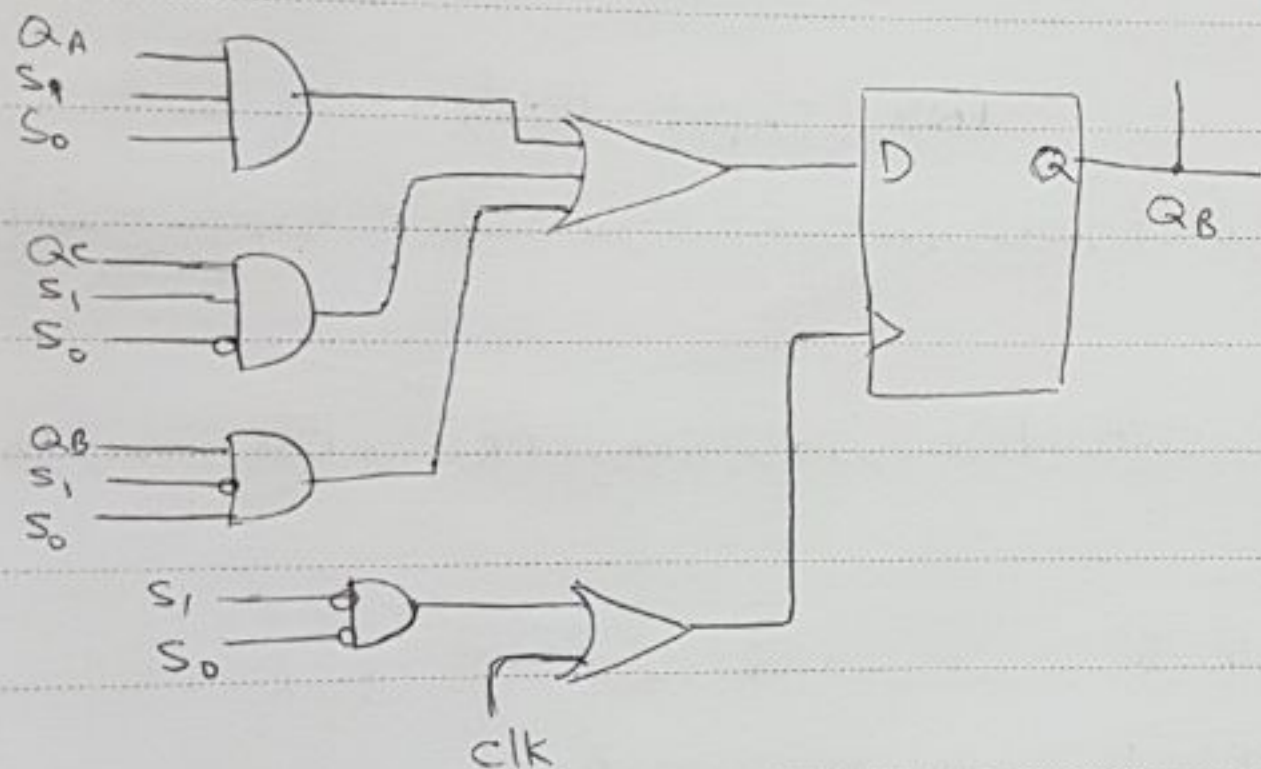
نمونه مدار برای طراحی که 8 بیت بگیرد، انتقال بدهد و...



ترانه 194 یک SR 4 بیتی دو طرفه (Bidirectional) با ورودی و خروجی سریال / موازی



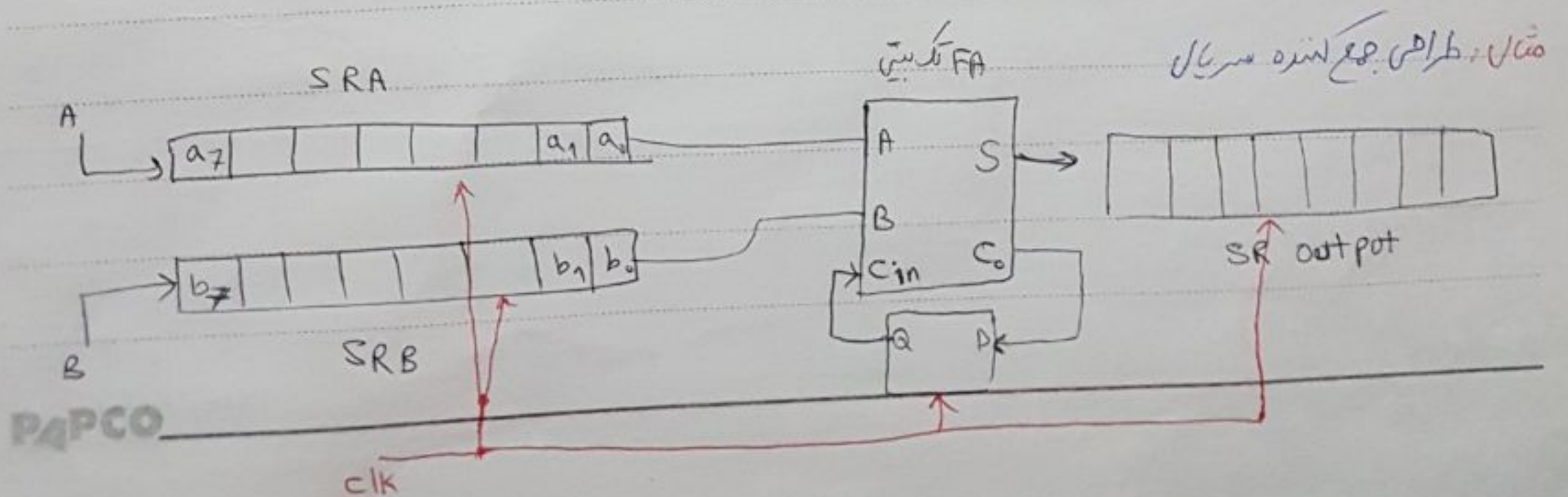
$$Q_B^* = S_1 S_0 Q_A + S_1 \bar{S}_0 Q_C + \bar{S}_1 S_0 Q_B$$



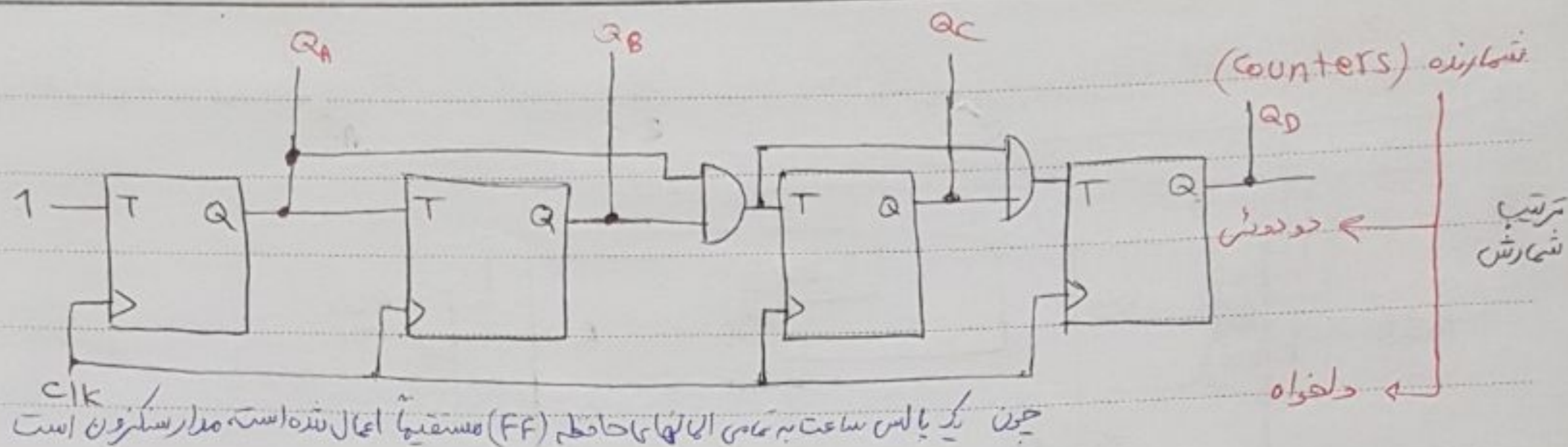
$S_1$	$S_0$	Operation
1	1	shift to right
1	0	shift to Left
0	1	sync load
0	0	clock hold

$S_1, S_0, clk$ , 4 تا ورودی 4 خروجی و ورودی سریال، زمین و  $V_{cc}$  به تعداد پایه ها

ترانه 299 SR 8 بیتی دو طرفه ورودی / خروجی سریال / موازی  
ورودی load با خروجیها حالتی یکسره اند به علاوه پایه های یکسان دارند



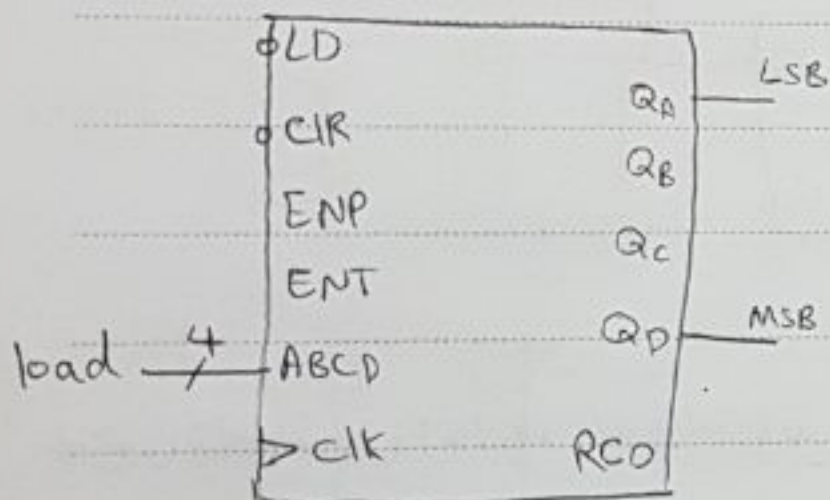




چون یک پالس ساعت به همگی آنها می‌خورد (FF) مستقیماً اعمال شده است. مدار سنگین است

به علت تأخیر انتشار در تقسیم خروجی هر FF تأخیر این تغییر در گام بعدی پالس ساعت بر روی FF بعدی اعمال خواهد شد. با توجه به عملکرد مدار پس از رسیدن به گام آخر ( $Q_D Q_C Q_B Q_A = 1111$ ) مجدداً به گام اول (0000) باز خواهد گشت

مراشه 163 یک شمارنده 4 بیتی سنگین است با قابلیت load و Reset سنگین



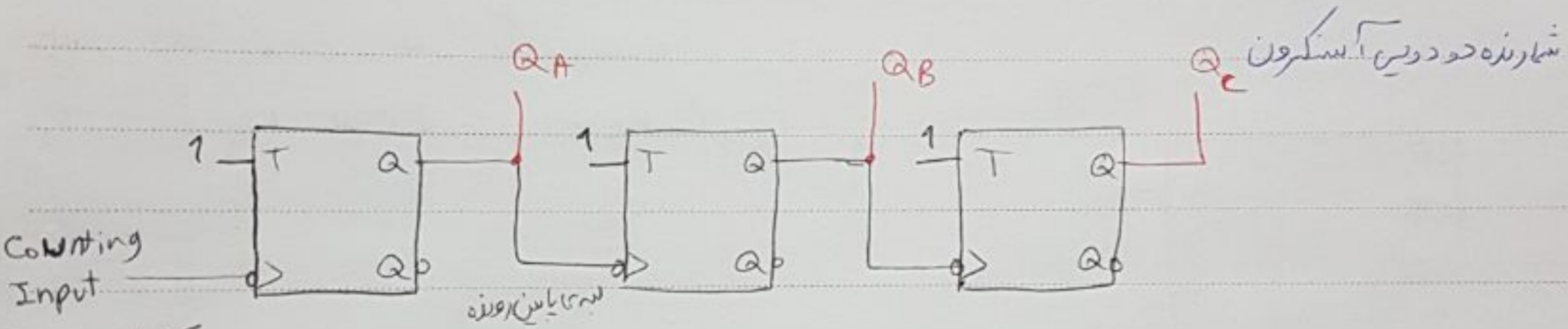
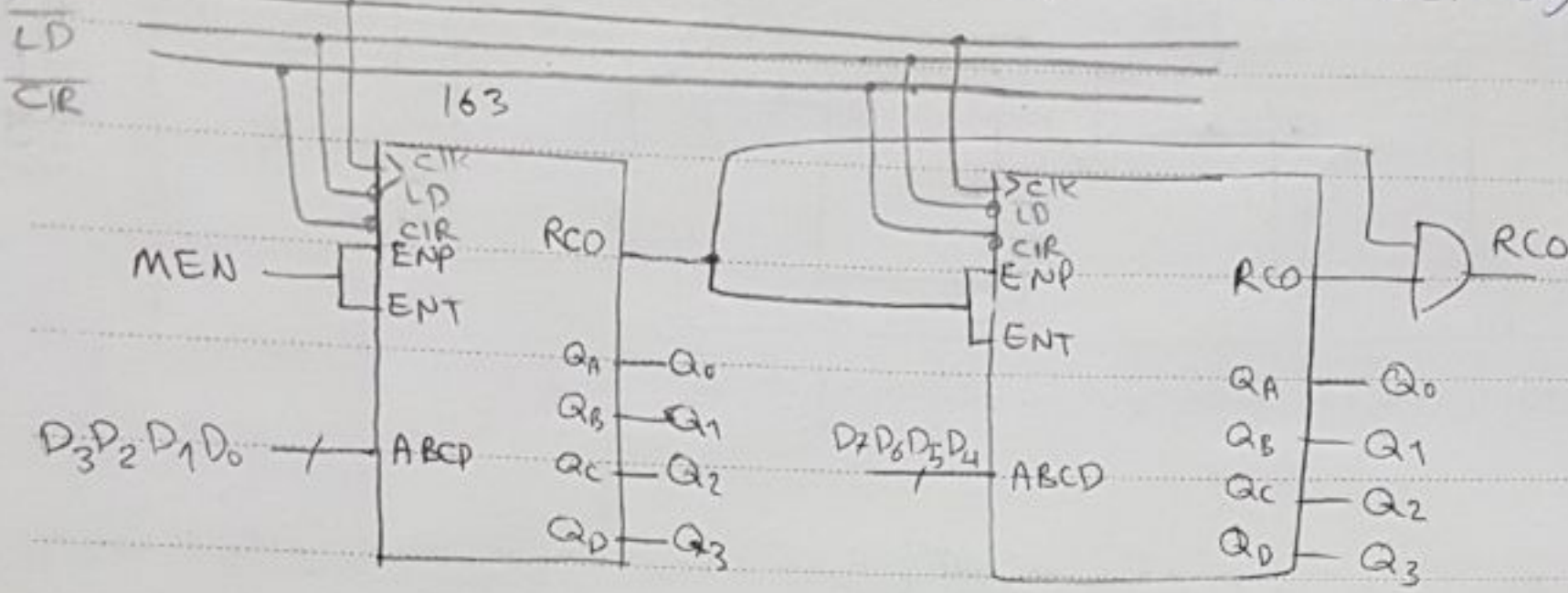
CLK	$\overline{CLR}$	$\overline{LD}$	ENP	ENT	Mode
↑	0	1	X	X	Reset
↑	1	0	X	X	load
↑	1	1	1	1	Count
X	1	1	X	0	Hold
X	1	1	0	n	Hold

هنگامی که شمارش به انتها برسد و ENT فعال باشد RCO فعال خواهد بود

Round Count out ← استانی رسیدن به شمارش و ریست شدن را خبر می‌دهد



مثال: طراحی شمارنده 8 بیتی با کمک 163 می توان بدون کمک از تراشه و با 8 TFF هم طراحی کرد



تعداد این ها در می شمارند  
clk نوشته شده که با سیکل  
اشتباه نداریم

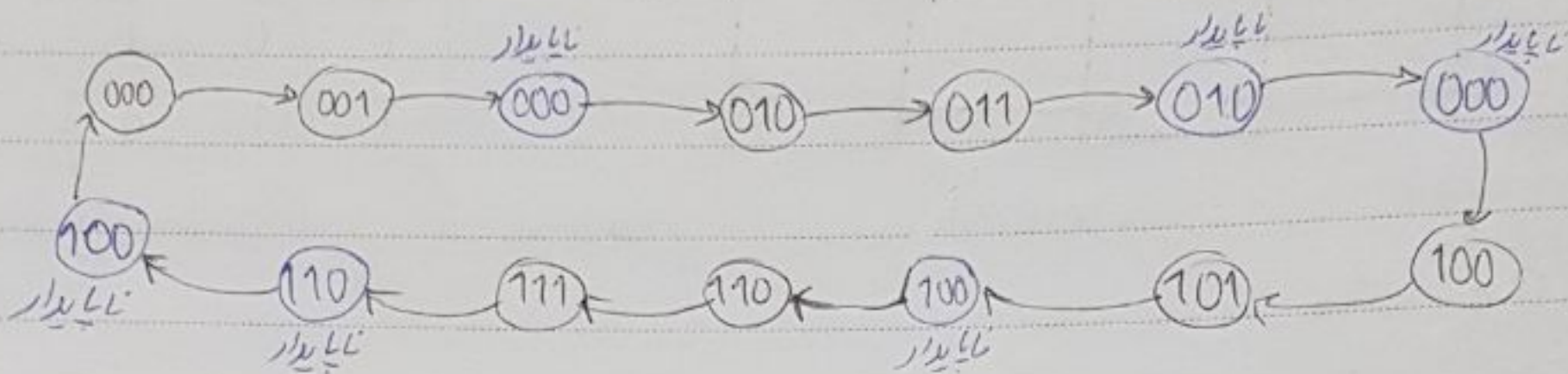
\* چون با لیس شمارش (جایگاه یا لیس ساعت) بین همگی FF ها مشترک اعمال نشده است مدار آسکرون است  
\* ورودی تها می TFF ها یک صرا باشد (فرمان Toggle) اما این امر هنگام ریخ ص ده که بیت خروجی FF قبل از  
1 به 0 تغییر کند (لبه پایین رونده)

Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

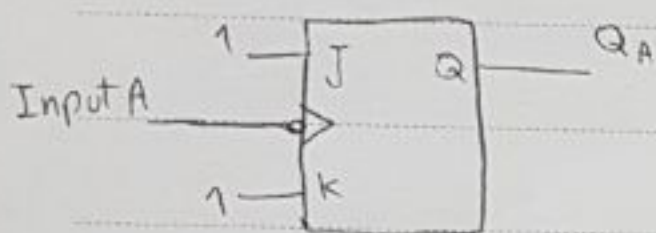
لبه پایین رونده که ورودی شمارش



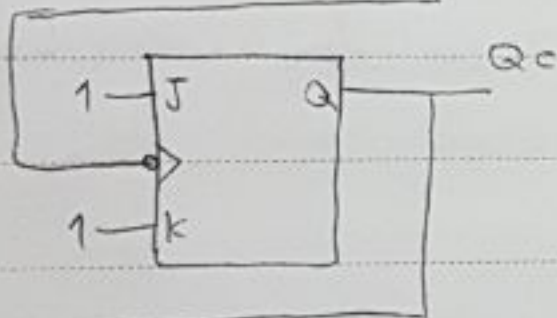
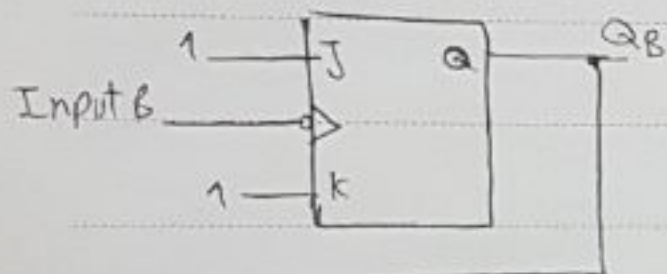
در مدارهای آسکرون حالت پایدار بین دو حالت داریم چون به ترتیب تغییر می کنند



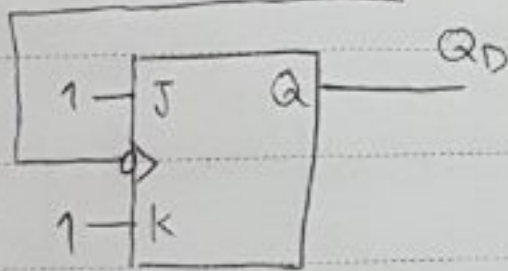
ترانه 293 یک شمارنده خود برآسکرون (ریپل) (4 بیتی)  
حارای CTR آسکرون



شمارنده تک بیتی

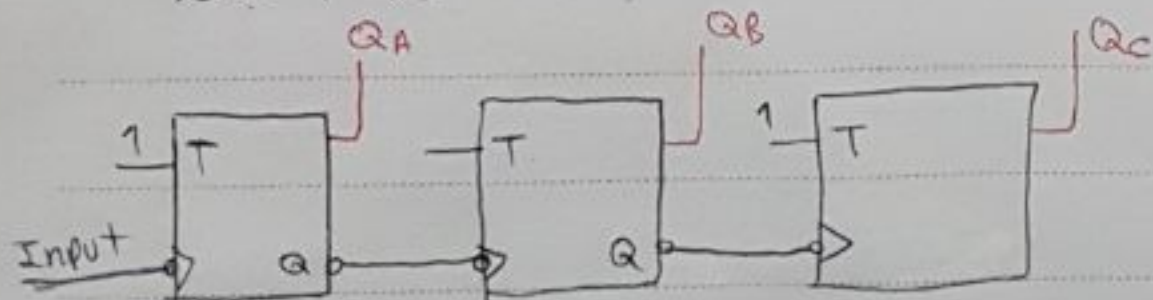


شمارنده 3 بیتی



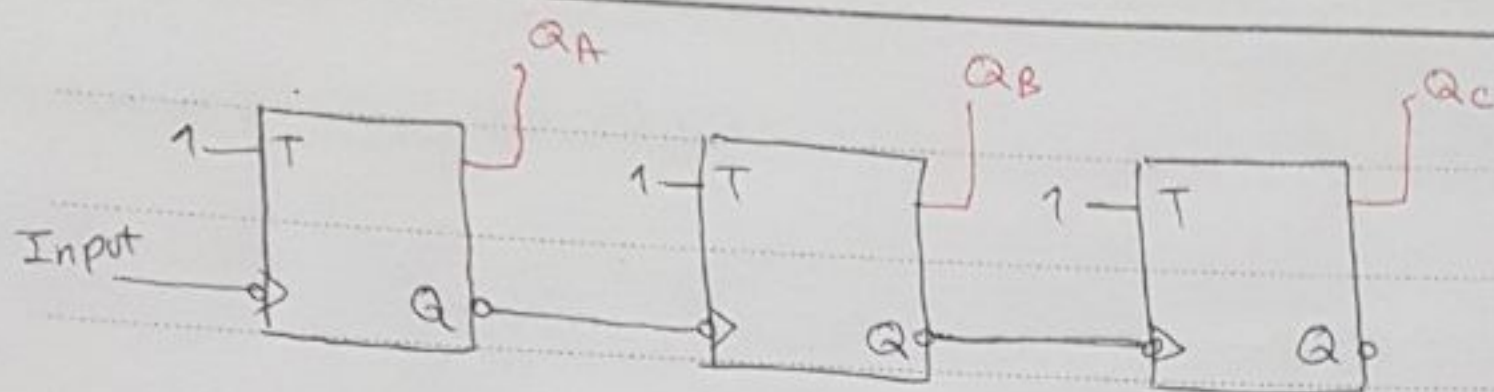
ترانه 177 شمارنده ریپل 4 بیتی با قابلیت بارگذاری، clear و preset می باشد  
برای وقتیکه که می خواهیم یکم از این مدار شروع کنیم به شمارش

شمارنده های پایین شمار در این شمارنده ها جدا از انجام شمارش به صورت افزایشی شمارش به صورت کاهش نیز انجام می پذیرد



آسکرون پایین شمار





من تو نیم از خود Q بگیریم و clk هم 0 نداریم ← انکار دوتا Not شده

QA QB QC

1 1 1

1 1 0

1 0 1

1 0 0

0 1 1

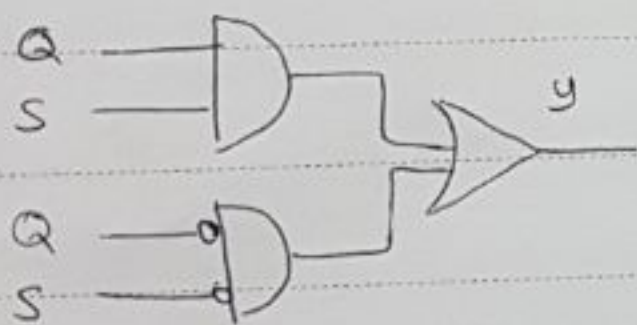
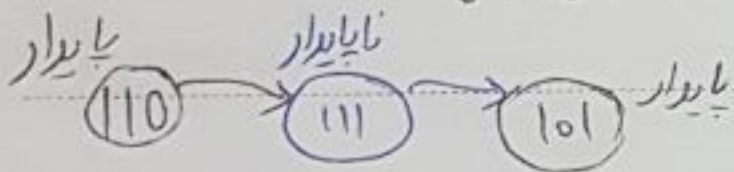
⋮

0 0 0

1 1 1

در این تغییر QA لبه پایین رونده ایجاد کرده است  
اما QA لبه بالا رونده ← QB بدون تغییر مانده است

همچنان حالات تا ۸ بار ما بین حالاتی که بیش از یک بیت تغییر می کنند به وجود خواهند آمد به عنوان مثال

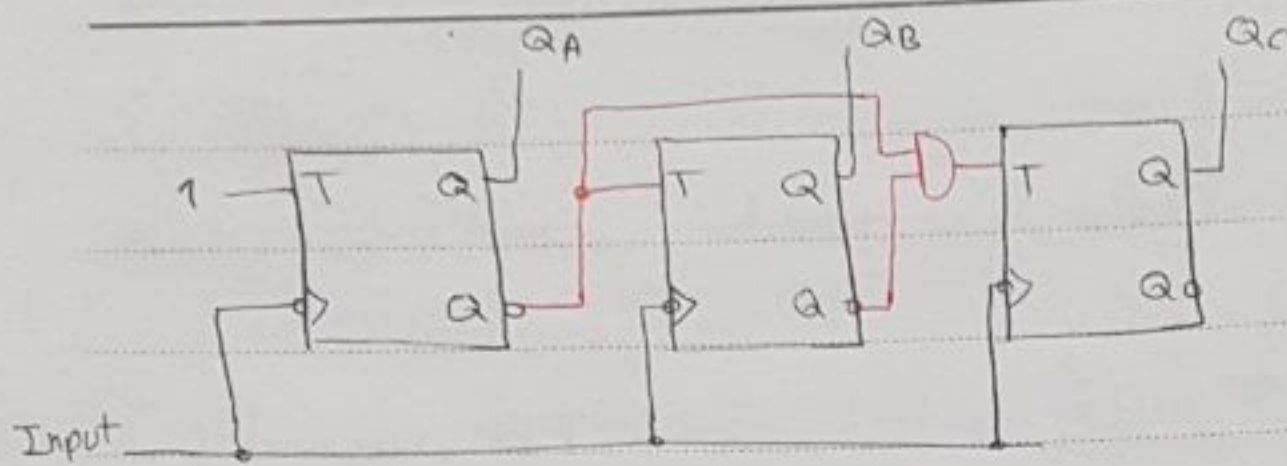


بخواهیم هم با بین شمار باشیم هم بلا شمار بین هر دو TFF از این میزنیم  $Y = S \oplus Q$

$$C_{next} = S \odot Q_{before}$$



شماره سنکرون پایین شمار



$Q_C \quad Q_B \quad Q_A$

1 1 1

1 1 0

1 0 1

1 0 0

0 1 1

چون تاخیر زمانی داریم برای  $Q_B$  اینجا به  $Q_A$  قبل تویم کنیم

برای  $Q_C$  باید به هر دو  $Q_B$  و  $Q_A$  در قبل تویم کنیم که هر دو مشترک بشوند  $Q_C = 1$

$X \cdot Q_A = 1$

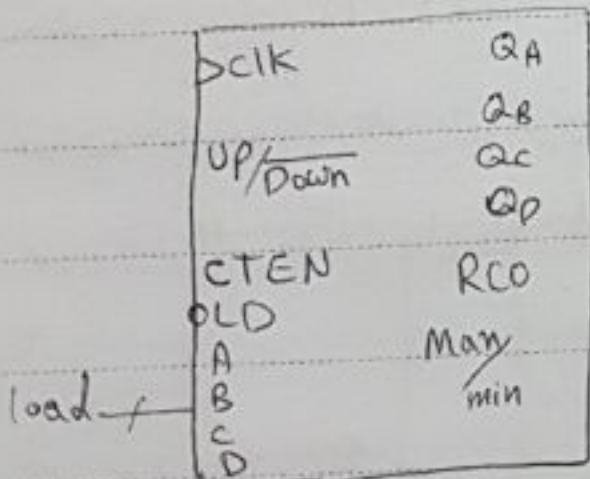
$Q_A = Q_B = 0$  باعث تغییر  $Q_C$  میشه

آیا اگر در شمارنده ای بالا شمار و خاص خروجی هارا مقیم کنیم، شماره های پایین شمار بدست می آید

در شماره ها ورودی Up/Down تعیین می شود که وظیفه انتخاب بین شمارش به بالا و یا به پایین را بر عهده دارد

مراقبه ۱۹۱ شماره سنکرون ۴ بیتی همراه با گذارهای سنکرون و قابلیت بالا / پایین شمار

۱۹۱



CTEN: Counting Enable

RCO: رسیدن به انتهای شمارش را نشان می دهد

Max/Min: اطلاع رسیدن (نتیجه به Max مقدار و یا Min



عملیات بارگذاری:

همچنان بارگذاری آسکرون راه حل سریع و بدون تشبیه است

مقداری که می‌خواهیم در TFF بارگذاری شود

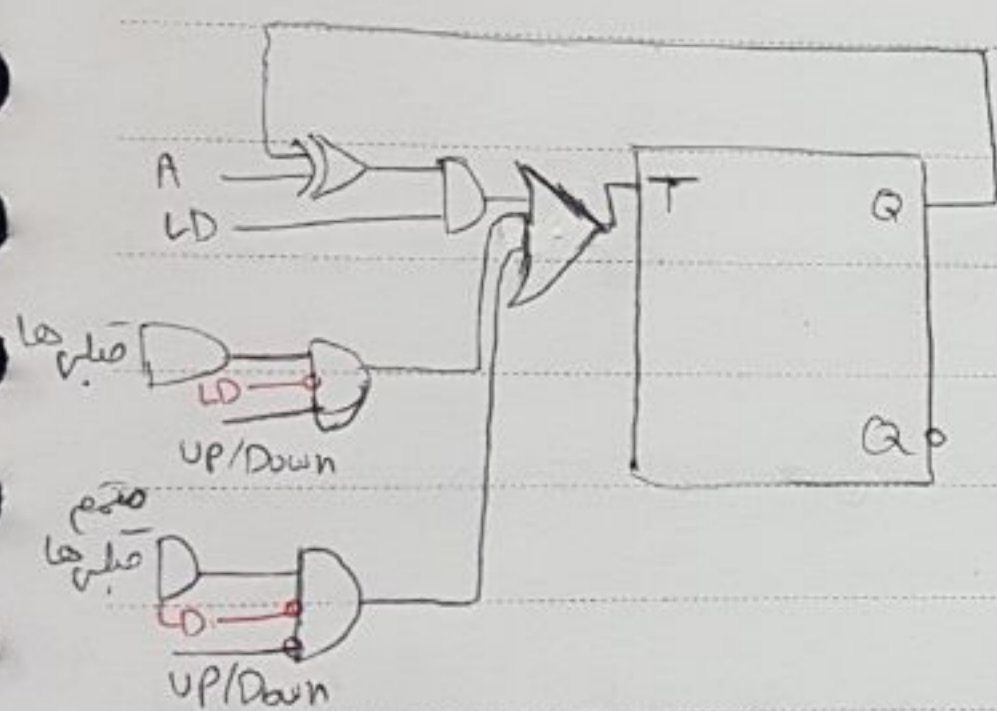
T	Q	Q*	A	Q	T	Q*
0	0	0	0	0	0	0
1	0	1	0	1	1	0
0	1	1	1	0	1	1
1	1	0	1	1	0	1

TFF در سیکرون

با کد جدول صحت و یافتن مقدار مناسب ورودی FF (ورودی T در مثال فوق) با در نظر گرفتن مقدار قبلی حافظه و مقدار

معلوم بارگذاری می‌توان مقدار لازم را تهیه نمود

در مثال فوق برای عملیات بارگذاری می‌توان از لیت NOR استفاده کرد



منطق LD مثبت است ←

شمارنده مبنای N به مبنای ریاضی اشتباه نیست

در این شمارنده‌ها که همچنان ترتیب دودش را رعایت می‌کنند (عمداً) تعداد کم‌های شمارش برای n بیت حافظه برابر  $2^n$  کم نمی‌باشد بلکه n کم دلخواه خواهد بود

شمارنده مبنای 10

نام دیگر این شمارنده ها شمارنده BCD می‌باشد

تراشه 160 شمارنده BCD سیکرون همراه با RCO برای دسته بندی شمارش و قابلیت بارگذاری Clear سیکرون



به بیت اول مشابه شماره ده با پیشوند ساده است

$Q_D \quad Q_C \quad Q_B \quad Q_A$

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

بیت چهارم ( $Q_D$ ) به غیر از کام نهم در برگشت به کام اول نیز تغییر می کند

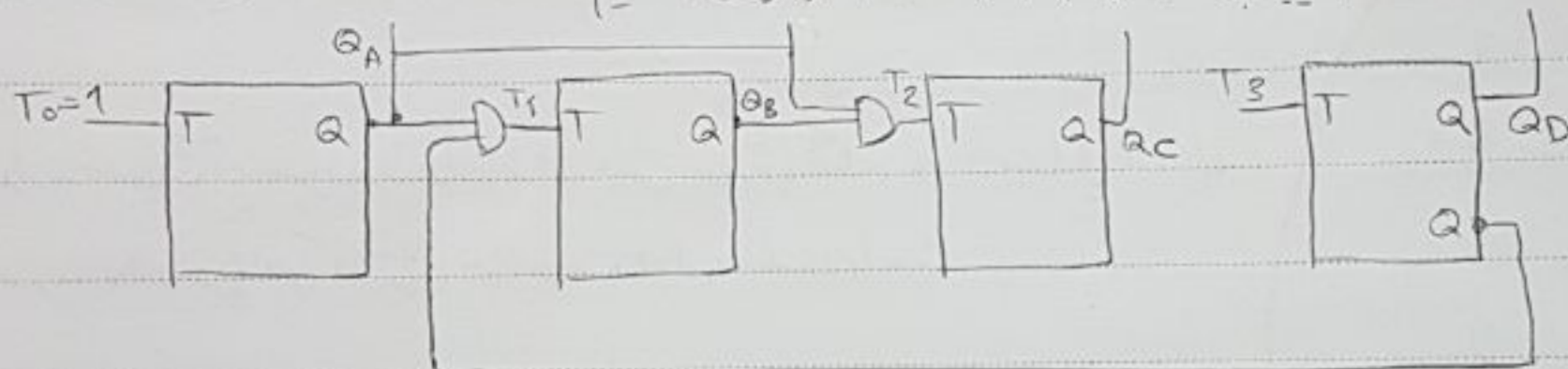
کام نهم: هر سه بیت پایین به باشند  $Q_A Q_B Q_C$

برگشت به کام اول: وقت بیت ها به صورت 1001 باشند باستی Toggle رخ دهد  $Q_D \bar{Q}_C \bar{Q}_B Q_A$

$\rightarrow Q_D Q_A$

$$Q_A Q_B Q_C + Q_A Q_D = Q_A (Q_D + Q_B Q_C) = T_3$$

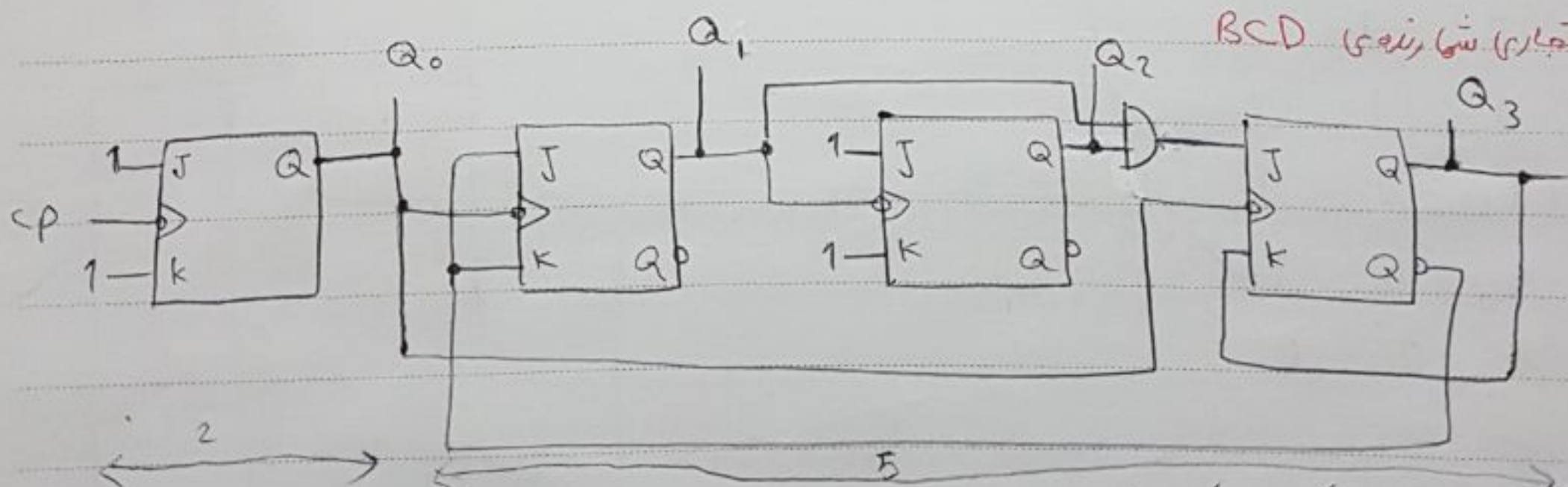
فلیپ فلاپ دوم Toggle می کند اگر بیت پایین به باشد ( $Q_A$ ) + به حالت آخر نرسیده باشیم



$$Q_D \bar{Q}_C \bar{Q}_B Q_A \cdot Q_A = \bar{Q}_D \cdot Q_A$$

بیت پایین یک باشند ولی به حالت آخر نرسیده باشیم

مربوط به شمارنده BCD



تحلیل کنید در آن سنکرون امکان دارد بین حالات یا در حالت ناپایدار بوجود بیاید (شبیه سازی VHDL ویا Verilog)

و یا صورتی رسم

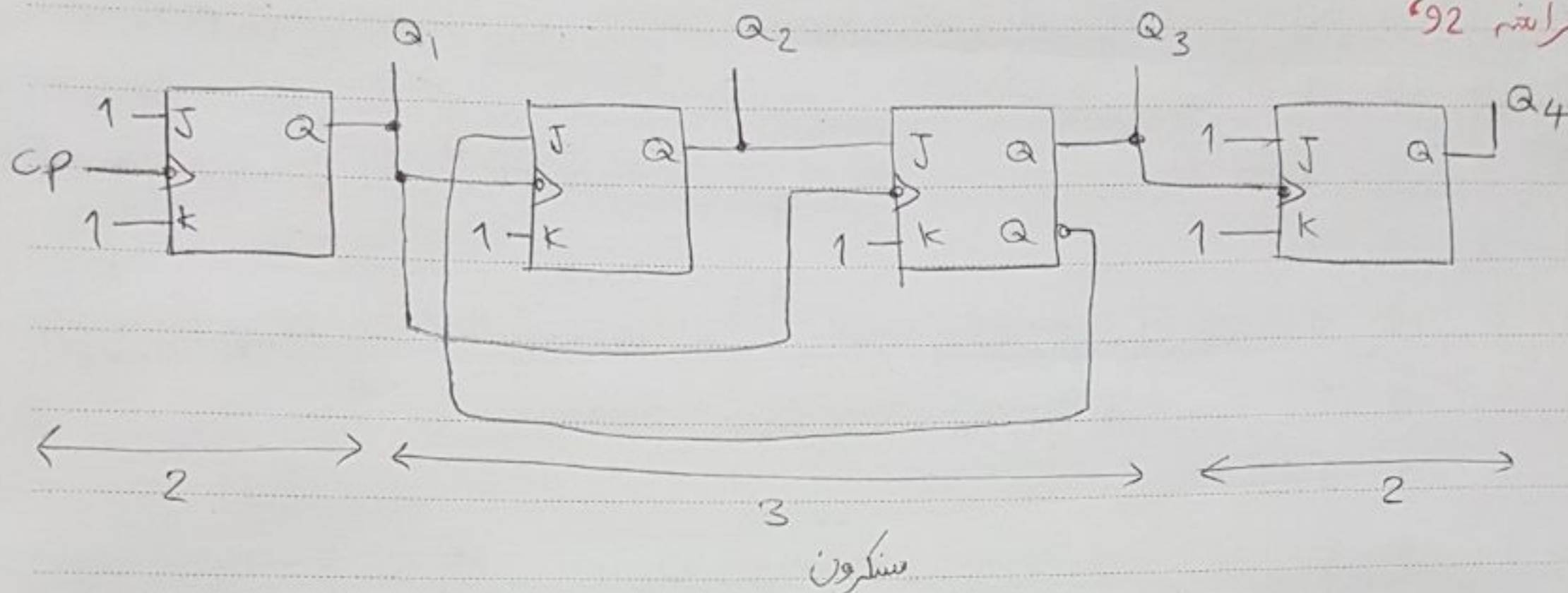
حالت پایدار به به خاطر تغییر یا پس ساعت ایجاد می شه، تا وقتی ورودی تغییر نکرده تغییر نمی کند

حالت ناپایدار بدون تغییر ورودی، تغییر می کند



شماره منهای 6 و 12

تاریخ 92



سنگین

Q<sub>4</sub> Q<sub>3</sub> Q<sub>2</sub> Q<sub>1</sub>

0 0 0 0

0 0 0 1

0 0 1 0

0 0 1 1

0 1 0 0

0 1 0 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

0 0 0 0

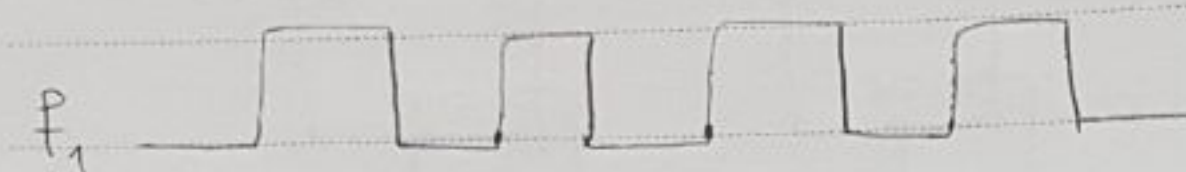
در مدار سنگین به معنای دقیق در مدار آسانترین به معنای جدید نگاه میکنیم

مدار به صورت آسانترین رفتار میکند اما بخش شمارش سنگین و وسط به دلیل اتصال همزمان هر دو پالس کنترل به صورت داخلی سنگین است



یکی از کاربرد های مهم شمارنده ها تقسیم فرکانس است

اگر تعداد عدد TFF داشته باشیم که ورودی پالس ساعت یا فرکانس  $P$  به آن اعمال می شود

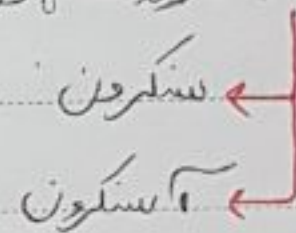


$$P_2 = \frac{P_1}{2}$$

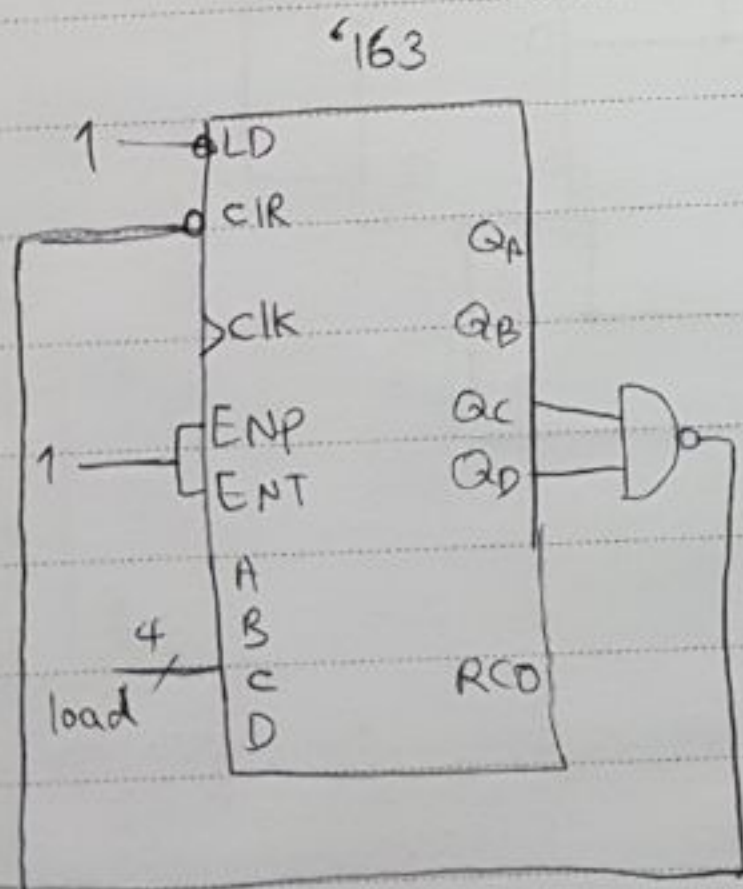
تقریباً ۱۰ هزار تقسیم فرکانس به صورت  $\frac{P}{6}$  تهیه کنید

شمارنده میبایستی دلفوا

در مبنای دلفوا (گام های دلفوا) می خواهیم شمارش حلقوی ایجاد شود  
استفاده از شمارنده های دودویی



مثال: با کمک شمارنده های سنکرون شمارنده میبایستی ۱۳ تهیه می کنیم



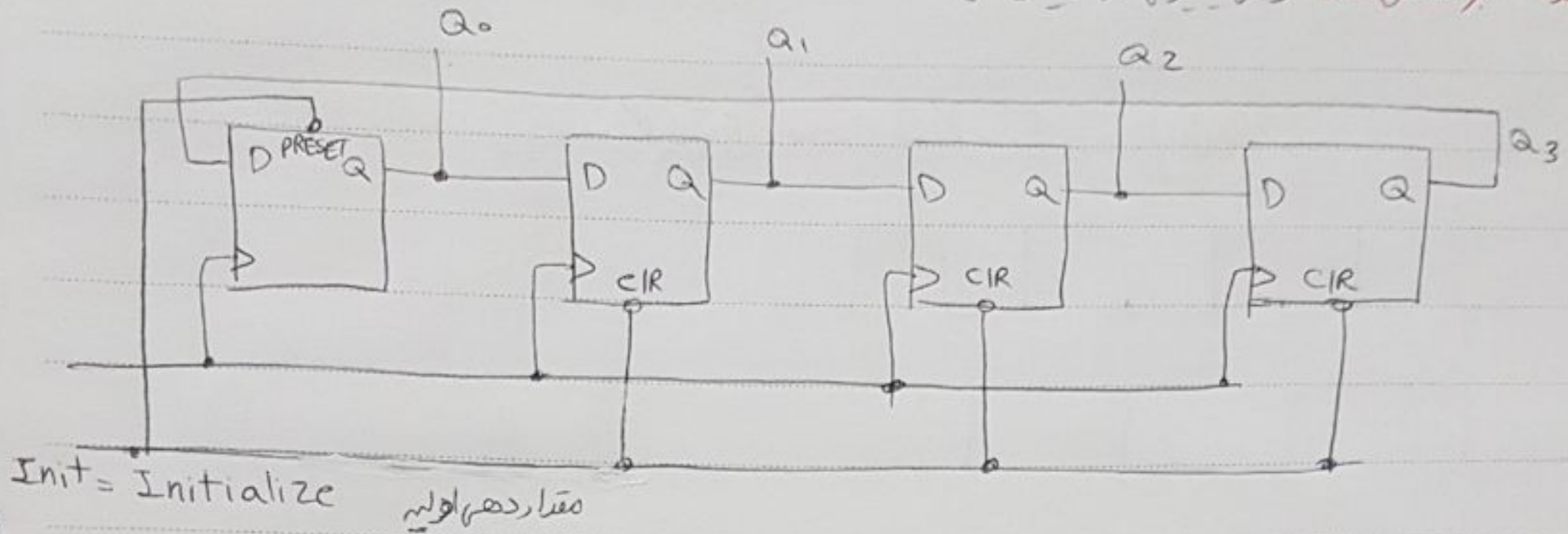
163 CIR سنکرون

\* به دلیل عملیات CIR سنکرون، این اتفاق در گام بعد از (۱۲) رخ می دهد  
که نتیجه آن پایداری بودن حالت ۱۲ (گام ۱۳) می باشد

\* اگر عملیات CIR آسنکرون باشد، با بستن گام شمارش متناظر با عدد ۱۳  
آشکار شود که با عملیات CIR در همان لحظه به عدد صفر تبدیل خواهد شد  
(حالت ناپایدار ۱۳)



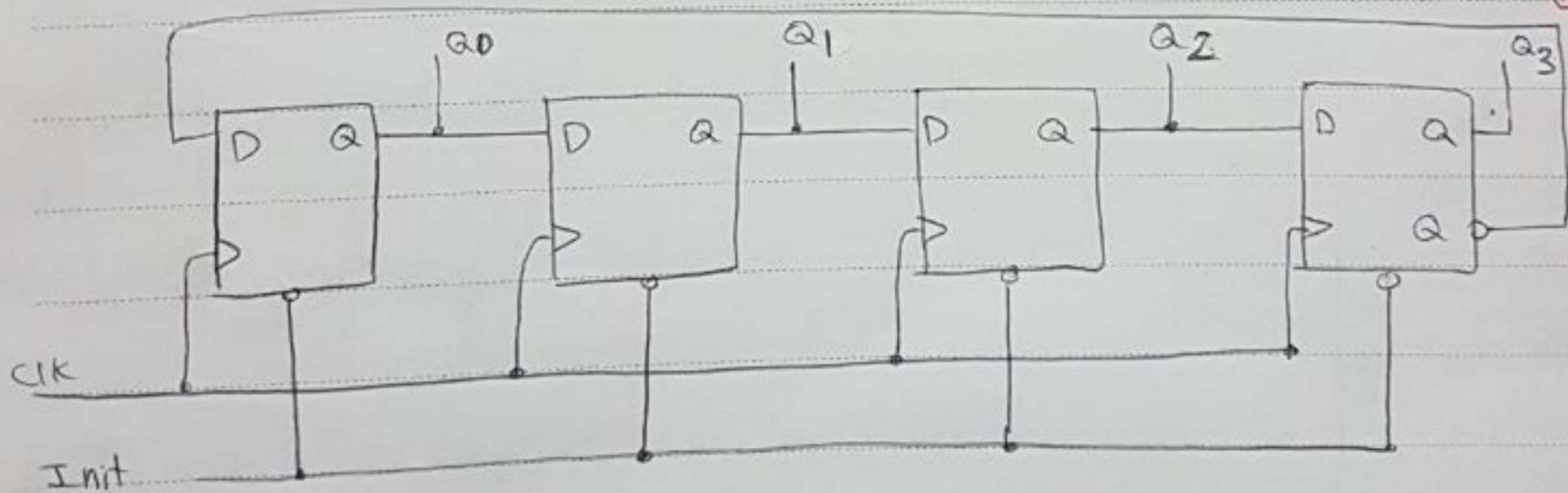
شمارنده براساس SR و قسیمی به نام Shift register



یا بدترین شمارنده

	$Q_3$	$Q_2$	$Q_1$	$Q_0$
→	0	0	0	1
	0	0	1	0
	0	1	0	0
	1	0	0	0

شمارنده چابک



	$Q_3$	$Q_2$	$Q_1$	$Q_0$
	0	0	0	0
	0	0	0	1
	0	0	1	1
	0	1	1	1
	1	1	1	1

1110  
1100  
1000



تمرین محوین: مدار جبراع را هفای طراحی کنید

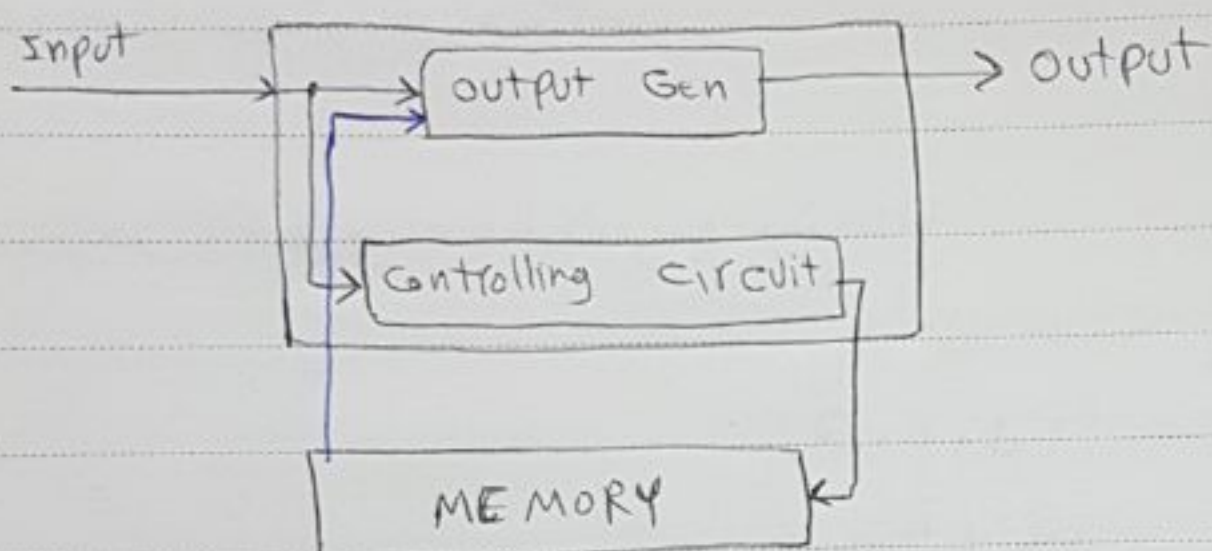
\* قابلیت شروع دوباره (Reset) سیستم  
سبز ← قرمز

\* استفاده از پالس ساعت 10 KHz

\* 20<sup>s</sup> سبز، 30<sup>s</sup> قرمز، 5<sup>s</sup> زرد

\* نمایش مقدار زمان با قرمز مایه مبر روی 7S

تحلیل و طراحی مدارهای ترتیبی منگرون



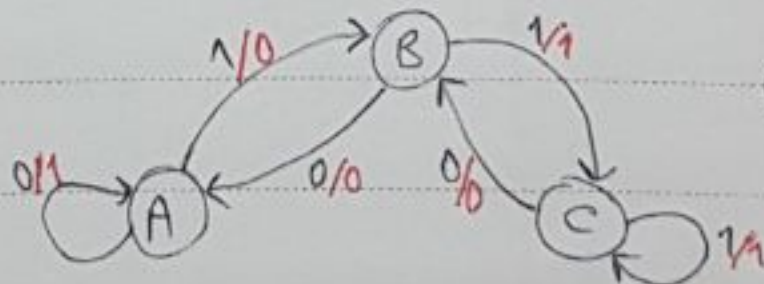
ماشین حالت میل (Mealy)

در این سیستم ترتیبی خروجیها تابع از حالت فعلی مدار و ورودیهای لحظه ای می باشند

ماشین حالت مور (Moore)

در این سیستم های ترتیبی خروجیها تنها تابع از حالت فعلی مدار می باشند

S. state	Input (n)	
	0	1
A	A/1	B/0
B	A/0	C/1
C	B/0	C/1



مثال: ماشین حالت میل

\* درون جدول ماشین میل  $\frac{S^*}{Z}$  حالت فعلی خروجی

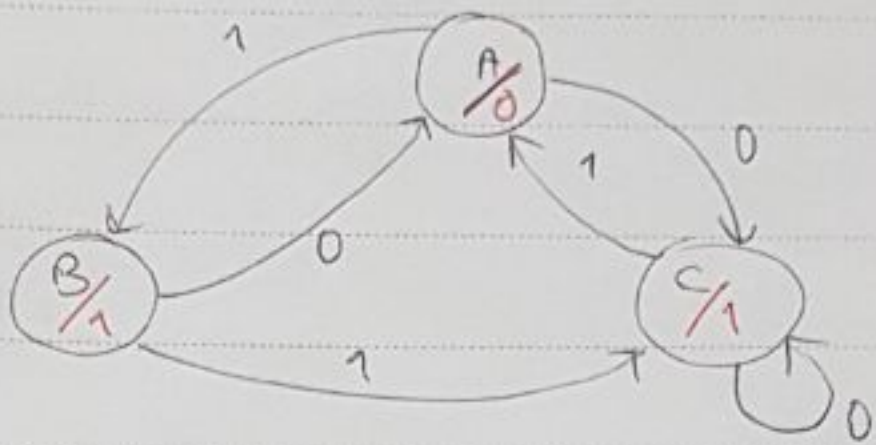
\* روی فلش ها: Input/output نوشته شده است

\* داخل دایره ها حالت فعلی مدار اشاره شده است (حافظ)

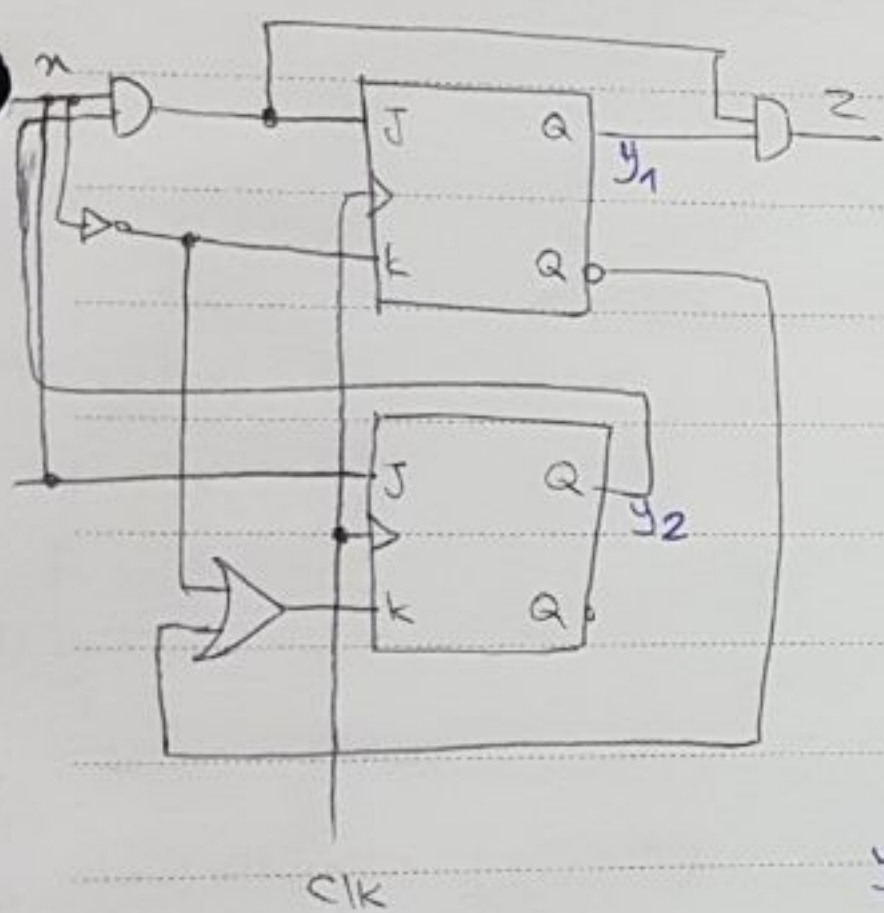
همین حالت ها در نظر گرفته شده از A و B و C هم 0 و هم 1 یکجا رفتن و خروجی مشخص



مثالی از ماشین مور



\* درون دایره ها:  $S/2$  اشاره می شود حالت فعل  $\leftarrow$  پس از افعال ورودی (ویدیو فلش ها) حالت جری حافظه تعیین خواهد شد خروجی



مثال: جدول عبور سوار مقابل را بدست آورید

به ازای هر  $FF$  یک سیگنال داریم و 2 سیگنال

$$Q^* = f(J, K, Q) \\ = J\bar{Q} + \bar{K}Q$$

برای هر بیت حافظه (هر کدام از ۴۴ ها) رابطه فوق را بدست می آوریم

$$\left. \begin{aligned} y_1^* &= J_1 \bar{y}_1 + \bar{k}_1 y_1 \\ J_1 &= n y_2 \\ \bar{k}_1 &= \bar{n} \end{aligned} \right\} \Rightarrow y_1^* = n y_2 \bar{y}_1 + n y_1$$

$$\left. \begin{aligned} y_2^* &= j_2 \bar{y}_2 + \bar{k}_2 y_2 \\ j_2 &= n \\ k_2 &= \bar{n} + \bar{y}_1 \end{aligned} \right\} \Rightarrow y_2^* = n \bar{y}_2 + n y_1 y_2$$

بالک روابط جبری مقادیر جدید حافظہ و خروجی را بر حسب ورودی و مقادیر فعلی حافظہ می یابیم

$$z = y_1 \wedge y_2$$



ماشین حالت مورد است

اسم بیت می دهیم (A و B) و دوباره جدول را می کشیم

x			x		
S	0	1	S	0	1
A	00	01/0	A	A/0	B/0
B	01	00/0	B	A/0	C/0
C	10	00/0	C	A/0	D/0
D	11	00/0	D	A/0	D/1

این مدار تشخیص می دهد که آیا ورودی 4 عدد 1 پشت سر هم اعمال شده است یا خیر؟

### طراحی ماشین حالت سکرون

(1) طراحی جدول عبور بر حسب صورت مسئله

(2) طراحی مدار ترتیبی بر حسب جدول عبور ← ساده تر و با این شروع می کنیم

فرض: طراحی جدول عبور انجام شده است

S	0	1	S	0	1
A	A/0	B/0	00	00/0	01/0
B	A/0	C/1	01	00/0	11/1
C	B/0	D/0	11	01/0	10/0
D	C/1	D/0	10	11/1	10/0

$\xrightarrow{\text{کاهش مقادیر مختلف به دلخواه}}$

برای راحتی محاسبات از گذری استفاده نموده ایم

با کمک جدول عبور حاوی مقادیر حافظه، روابط عبور را بدست می آوریم

$y_1 y_2$

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

$$Z = \bar{x} y_1 \bar{y}_2 + x \bar{y}_1 y_2$$

\* برای ادامه راه حل بایستی نوع فلیپ فلاپ مورد استفاده تعیین شود

ساده ترین راه حل متناظر با DFF می باشد چون  $Q^* = D$

و کافیست در مدارهای روابط محاسبه شده را عیناً به ورودی D متصل کنیم



0	0
0	1
0	1
1	1

$y_1^*$

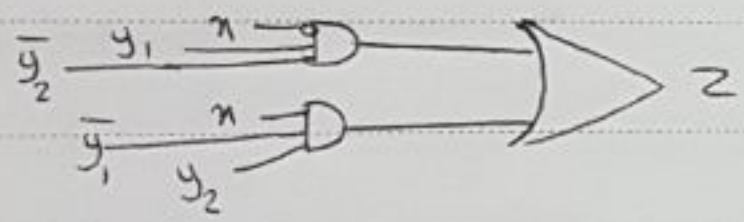
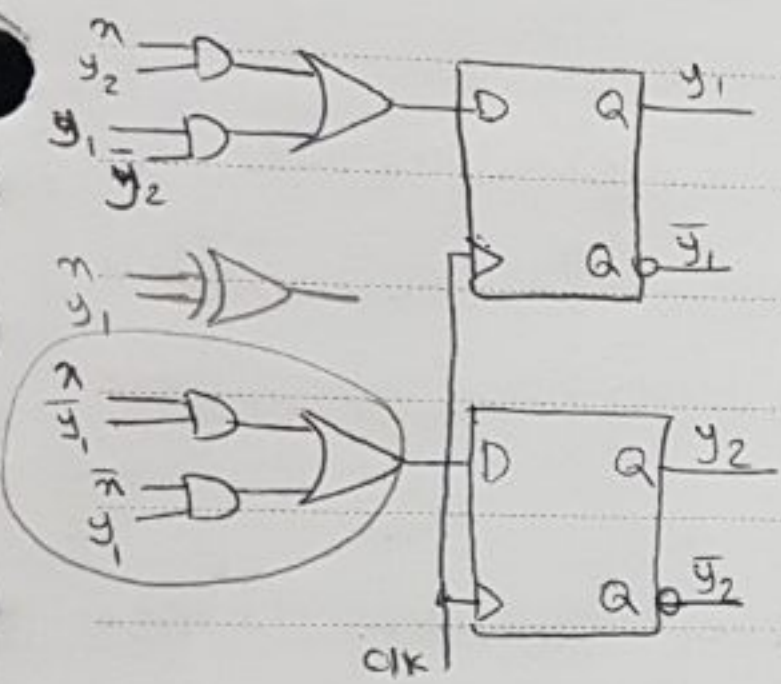
0	1
0	1
1	0
1	0

$y_2^*$

$$y_1^* = \overline{n}y_2 + y_1\overline{y}_2$$

$$y_2^* = \overline{n}\overline{y}_1 + \overline{n}y_1 = \overline{n} \oplus y_1$$

مدار نهایی:



طراحی بیت های حافظه JK FF

رابطه جبری بدست آمده را به صورتی تفسیر دهیم که به شکل  $y_i^* = (\overline{K_i})y_i + (J_i)\overline{y}_i$  درآید  
 مستقل از  $y_i$       مستقل از  $y_i$

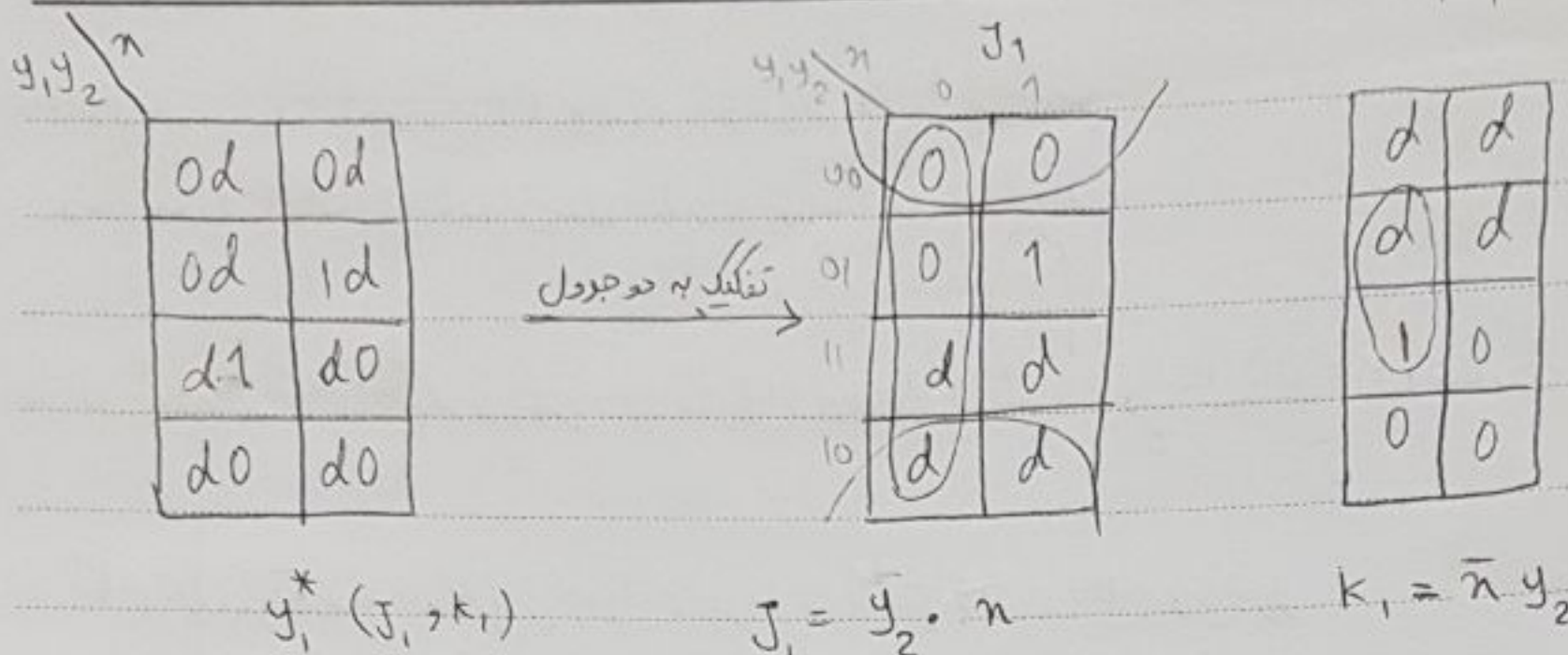
$$Q^* = J\overline{Q} + \overline{K}Q \quad \text{JK در حالتی:}$$

$$\text{مثال: } y_1^* = \overline{n}y_2 + y_1\overline{y}_2 = \overline{n}y_2(y_1 + \overline{y}_1) + y_1\overline{y}_2 = (\overline{n}y_2 + \overline{y}_2)y_1 + (\overline{n}y_2)\overline{y}_1$$

مستقیماً با کمک جدول عبور در JK FF می توانیم برای برقراری رابطه مشخص در خروجی از هندسه مکوس استفاده کنیم

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



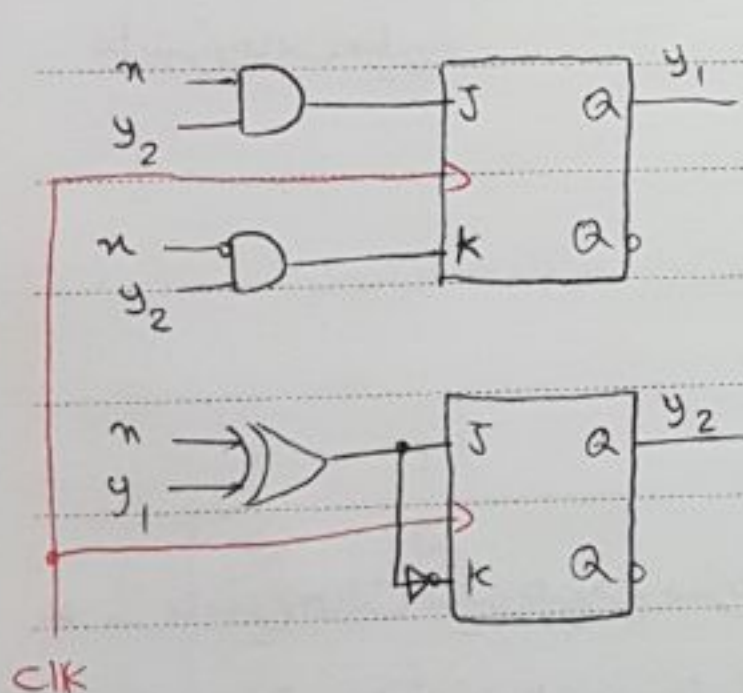


FF<sub>2</sub> مساویات برای

0	1
1	0
0	1
1	0

$$J_2 = \bar{n} y_1 + n \bar{y}_1 = n \oplus y_1$$

$$K_2 = n y_1 + \bar{n} \bar{y}_1 = n \odot y_1$$



مدارهای:

Q	S	R	$Q^*$
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	1

8 SRFF



می‌توان هر مدار را با هر نوع FF و حتی انواع از FF ها طراحی کرد  
و تنها مورد با اهمیت ساده تر شدن مدار نهایی بسکرون می‌باشد

مثال: مدار می‌خواهیم که در ورودی سریال ارسال ترتیب  $10 \rightarrow 00$  را آشکار نماید

گام اول: تقسیم به طراحی جدول عبور با ماشین حالت میلی و یا مور

Input(n)

	S	D	1	Z
شروع	A	B	A	0
0	B	C	A	0
00	C	C	D	0
001	D	E	A	0
0010	E	C	A	1

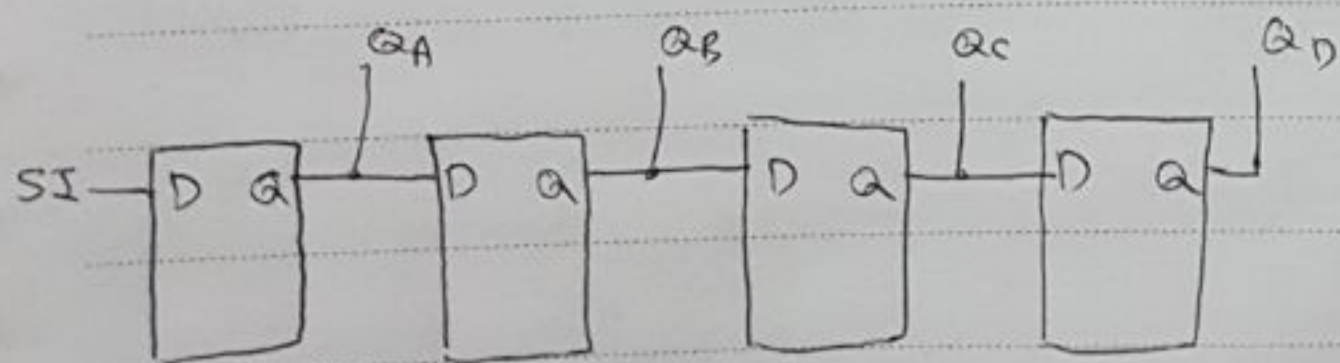
از ماشین حالت مور آغاز می‌کنیم

Input

	S	D	1
	A	B/0	A/0
0	B	C/0	A/0
00	C	C/0	D/0
001	D	B/1	A/0

ماشین حالت میلی

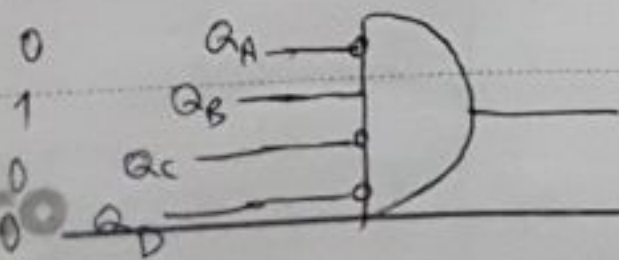
\* در ماشین حالت میلی خروجی سریعاً با ورودی لحظه‌ای می‌تواند تغییر کند و نیازی به تعریف حالت جدید نمی‌باشد  
در مثال فوق ماشین حالت مور با 3bit حافظه و 4 ماشین حالت میلی با 2 بیت حافظه قابل طراحی می‌باشد



راه حل هایلدر

استفاده از شیفتر رجیستر SR

(درامتان نحوه ندارد)





مداری طراحی کنید که اگر  $n=0$  باشد دو دوش بالا شمار و اگر  $n=1$  باشد دو دوش پایین شمار باشد (شمارنده 2 بیتی)

ماشین حالت مورد  $\rightarrow$  چون 2 حافظه را تقسیم می کند

S	Input (n)		
	0	1	Z
00 A	B	D	A
01 B	C	A	B
10 C	D	B	C
11 D	A	C	D

ترتیب 2.2 می  
در شماره

- \* اگر در صورت مساله به هر نحوی به مقادیر حافظه و یا ترتیب آنها اشاره شود با بیتی در مقداردهی حالت ها این مساله را لحاظ نمود
- \* حقت می کنیم که برای منتقل نمودن مقادیر به جدول کارنو با بیتی ترتیب مقادیر به صورت گذری باشد در نتیجه در مثال فوق کل سطر C را با کل سطر D جابه جا می کنیم (بدون تغییر حالات و یا تغییر مفهوم عملکرد مساله)

تبدیل غیر کوپلی (مهم) مدار کنترلی آسانسور را طراحی کنید

جدول ایجاب:

هدف کاهش تعداد حالات جدول عبور طراحی شده (حذف حالات غیر ضروری) و در نهایت رسیدن به مدار بهینه

S	n	
	0	1
A	C <sub>1</sub>	B <sub>10</sub>
B	C <sub>1</sub>	E <sub>10</sub>
C	B <sub>1</sub>	E <sub>10</sub>
D	D <sub>10</sub>	B <sub>1</sub>
E	E <sub>10</sub>	A <sub>1</sub>

مثال:

مشکلات طراحی در تقسیم جدول عبور  $\rightarrow$  اشتباه در طراحی X  
استقاب حالات اضافی به صورت ناشیانه ✓

حالات مفاد: (1) خروجی مدار به ازای آیف همواره یکسان باشد A و B / B و C

(2) به ازای هر دنباله دلخواه ورودی که در نتیجه آیف تغییر حالات رخ می دهد

(حتی با گذر از حالات مورد نظر) خروجی مدار یکسان باشد



\* در دنباله حالات اثرهای ۲ حالت مشکوک را به دلخواه تعریف کنیم دنباله خروجی هیچ تفسیری نباید داشته باشد  
اثر ۲ حالت مذکور معادل باشند

	B			
B	X			
C	X	✓		
D	X	X	X	
E	X	X	X	X
	A	B	C	D

هر حالت با حالت های برابر مقایسه می شود (خودش و قبلی ها نه)

\* **گام اول:** در هر خانه جدول برابری خروجیها به ازای دو حالت در حال مقایسه را یک می کنیم و اگر برابر نباشند علامت X نشان می دهیم که معادل بودن غیر ممکن است

\* **گام دوم:** در خانه های خالی شرط برابری حالات را می نویسیم (بشرط لازم برای برابری حالات بعدی)  
برای برابری B و A باید E و B با هم برابر باشند

\* **گام سوم:** تعیین تکلیف حالات نوشته شده با یک نمون آفاد در خانه های دیگر جدول

\* اثر برابری دو حالتی به برابری خود آنها وابسته باشد در عمل می توان آنها را برابر در نظر گرفته  
اثر برابری زوج حالاتی به یکدیگر وابسته باشند باز هم آن زوج حالات 2 به 2 با یکدیگر معادل اند

مثال:

S	D	1
A	E/0	D/0
B	A/1	F/0
C	C/0	A/1
D	B/0	A/0
E	D/1	C/0
F	C/0	D/1
G	H/1	G/1
H	C/1	B/1

B	X					
C	X	X				
D	✓	X	X			
E	X	✓	X	X		
F	X	X	✓	X	X	
G	X	X	X	X	X	X
H	X	X	X	X	X	X
	A	B	C	D	E	F



امراز حالات عبور

$\{A, D\}, \{B, E\}, \{C, F\}, \{G\}, \{H\}$

جدول عبور ساده شده

S	0	1
A	B/0	A/0
B	A/1	C/0
C	C/0	A/1
G	H/1	G/1
H	C/1	B/1

تحلیل و طراحی مدارهای ترتیبی آسنکرون

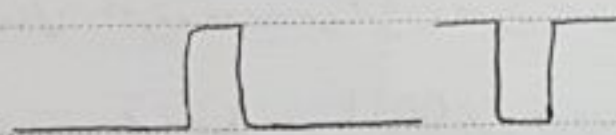
این مدارها به خود سستی ازیر تقسیم می شوند

(1) مدارهای مود پالس Pulse - Mode در امتحان می آید

(2) مدارهای مود اصلی Fundamental - Mode

در مدارهای مود پالس، ورودی و خروجی مدار (سطح منطقی متناظر) با یک حضور و یا عدم حضور پالس تعیین می شود

مثل کلیدهای فشرده که وقتی میزنیم بهر ترمز به حالت قبل



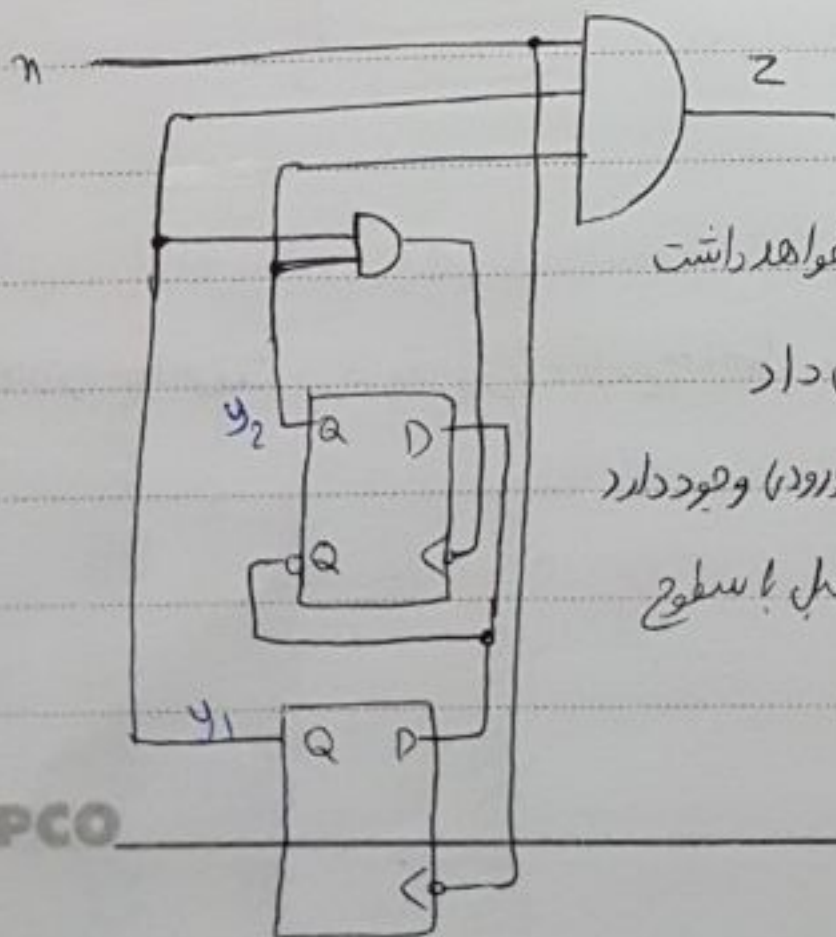
پالس:

عدها با یک فلپ فلاپ طراحی می شوند

ارتباط بین دو ایمان

در مدارهای مود اصلی، با یک ثبت ها یا به ای منطق ترهای حافظه و خروجی با یک تاخیر ثبت ها بیان می شوند

مثال، مدار مود پالس



\* در این مدارها اگر  $n$  ورودی وجود داشته باشد  $n+1$  حالت مجاز وجود خواهد داشت

\* البته می توان با یک مدار واسطه حالت های مختلف ورودی را به  $2^n$  افزایش داد

\* دلیل در این مدارها خطرناک یا بیاری در صورت اعمال همزمان پالس های ورودی وجود دارد

\* ورودی ها و خروجی ها با منطق پالس کار می کنند اما عناصر ما به نقل با سطح

ولتاژ کار خواهند کرد



pulse input				
$y_1, y_2$	S	PI <sub>0</sub>	PI <sub>1</sub>	
00		00/p0 <sub>0</sub>	10/p0 <sub>0</sub>	<div> </div>
				در ورودی پالس اعمال نشده است
				در ورودی پالس آمده است

اگر در ورودی پالس دیده نشود عملاً هیچ لبه‌ای به FF<sub>1</sub> اعمال نشده است  
 $\Rightarrow y_1^* = y_1 = 0$  (در ابتدای کار) و در ادامه خروجی And  $y_1, y_2$  صفر بوده است و صفر هم باقی می‌ماند  
در نتیجه FF<sub>2</sub> نیز در حالت Hold باقی خواهد ماند

S	PI <sub>0</sub>	PI <sub>1</sub>
00	00/p0 <sub>0</sub>	10/p0 <sub>0</sub>
01	01/p0 <sub>0</sub>	11/p0 <sub>0</sub>
10	10/p0 <sub>0</sub>	10/p0 <sub>0</sub>
11	11/p0 <sub>0</sub>	00/p0 <sub>1</sub>

حالتی که در این حالت خارج نمی‌شود

مثال: اگر در سیستم فرض شود که ورودی سریال همواره به صورت رنج رنج ارسال شده است (سنگرون)  
ترتیب 0011 در ورودی را آشکار کنید

n		
S	0	1
A	B/0	A/0
0 B	C/0	-/-
00 C	B/0	D/0
001 D	-/-	A/1

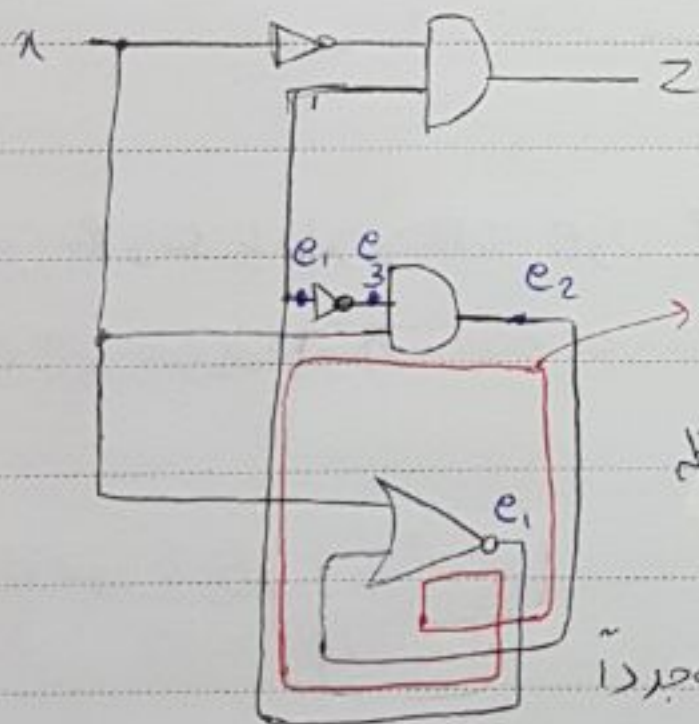
\* حالات با اهمیت در خروجی و یا مقادیر حافظه ممکن است مطرح شوند که در نهایت در ساده سازی مدار نهایی کمک می‌نمایند



مثال (ترم پیش) ترتیب 1001 را در هر چهار تایی ورودی سریال بیابید

S	n	
	0	1
A	w'/0	B/0
B	c/0	w'/0
C	D/0	w'/0
D	A/0	A/1
w'	w'/0	w'/0
w''	w'/0	w'/0
w'''	A/0	A/0

مثال مدار مورد اصلی



فیدبک را شناسایی می‌کنیم  
در حلقه‌ی فیدبک گره‌ها را شناسایی می‌کنیم  
آیا بین گره‌ها (سطح منطقی) ارتباط دائمی وجود دارد یا خیر؟  
اگر آنگاه وجود دارد (در مثال فوق  $e_1 = \bar{e}_1$ ) آن گره را از فهرست گره‌های حافظه خارج می‌کنیم  
در بعضی مدارها می‌توان از ابتدا پیش‌بینی نمود پس از تحلیل مسئله مجدداً بررسی خواهیم کرد

مقدار فعلی و مقدار جدید مقادیر مشکوک به حافظه وابسته به تأخیر است با توجه به تأخیر در مسئله با در نظر گرفتن تأخیر جدول عبور را کامل می‌کنیم

e <sub>1</sub> e <sub>2</sub> S	Input	
	0	1
00	10/0	01/0
01	00/0	01/0
10	10/1	00/0
11	00/1	00/0

$$\frac{s}{z} \begin{matrix} * & * \\ e_1 & e_2 \end{matrix}$$

$$e_1^* = \bar{x} + e_2$$

$$e_2^* = x \cdot \bar{e}_1$$

$$z = \bar{x} \cdot e_1$$



آیا  $e_1$  و  $e_2$  بیت‌ها حافظه هستند؟ بله

به ازای ورودی مشخص (در مثال ورودی یک بیتی،  $n=0$  یا  $n=1$ ) یک می‌کنیم که آیا مقدار جدید عناصر حافظه فرسوده تغییر دارند یا خیر به عبارتی آیا به مقدار فعلی (قبل) نیز وابسته هستند یا خیر

$$e_2 = e_1 \quad \times$$

$$e_2 = \bar{e}_1 \quad \times$$

آیا هر دو بیت  $e_1$  و  $e_2$  عناصر حافظه هستند؟ به چون رابطه بدیهی (رو به رو)

وجود ندارد

برای تشخیص سه بیت حافظه با بستی تمام حالات ممکن برای تابعیت بین 3 متغیر را یک نمود

$$e_3 = f(e_1, e_2)$$

		Input(n)	
S		0	1
A		C/0	B/0
B		A/0	(B/0)
C		(C/1)	A/0
D		A/1	A/0

حالات پایدار و ناپایدار

اگر بدون اینکه ورودی تغییر کند: حالت مولد ثابت بماند ← حالت پایدار  
حالت تغییر کند ← حالت ناپایدار

حالات B و C پایدار و حالات A و D ناپایدارند

حالت B در شرایط  $n=1$  و حالت C در شرایط  $n=0$  پایدار هستند

S		0	1
B		C/1	(B/0)
C		(C/1)	B/0

جدول عبور پایدار و

\* حالات ناپایدار میانی دیگر ذکر نمی‌شوند



$y_1 y_2$	$x_1 x_2$			
	00	01	11	10
00	00	01	00	01
01	00	01	11	01
10	00	10	10	11
11	00	10	00	11

مثال  
اگر در حالت نا یا بدار 11 قرار داشته باشیم و ورودی به 11 تغییر کند چه اتفاقی می افتد؟

انتظار از جدول این است که به حالت نا یا بدار 00 منتقل شود

مشکل اصلی در مدارهای آسکرون عدم تغییر همزمان الحاقی حافظه است که در طی تغییر حالت از 11 به 00، بیت های حافظه  $y_1 y_2$  هیچ گاه همزمان تغییر نمی کنند

فرض کنیم ابتدا  $y_1$  تغییر کند Race یا مسابقه رخ می دهد  $11 \leftarrow 01 \leftarrow 11 \leftarrow 01$

ریشه این مسابقه تغییر بیش از یک بیت در بیت های حافظه می باشد

فرض کنیم ابتدا  $y_2$  تغییر کند:  $11 \leftarrow 10$  یا بدار

طراحی مناسب باید به شکلی باشد که بین دو حالت نا یا بدار تنها یک بیت حافظه تغییر کند

مثال: طراحی شمارنده 4 بیتی آسکرون **با ترتیب گذری** (در هر دو سه مقدار دهی عناصر حافظه بین از جدول عبور ابتدایی)

$y_1 y_2$	$x_1 x_2$			
	00	01	11	10
00				
01				
10				
11			101	
101				

تعریف حالات نا یا بدار میانی با هدف رفع مشکل مسابقه

برای طراحی مدار ابتدا کارنو هم تقسیم برای هر کدام از  $y_1^*$  و  $y_2^*$  بعد برای رابطه بدست آمده کیت می گذاریم (فقط دقت شود که ترتیب جدول کارنو بر حسب قری است)