فصل چهارم

طراحی مدارهای ترتیبی

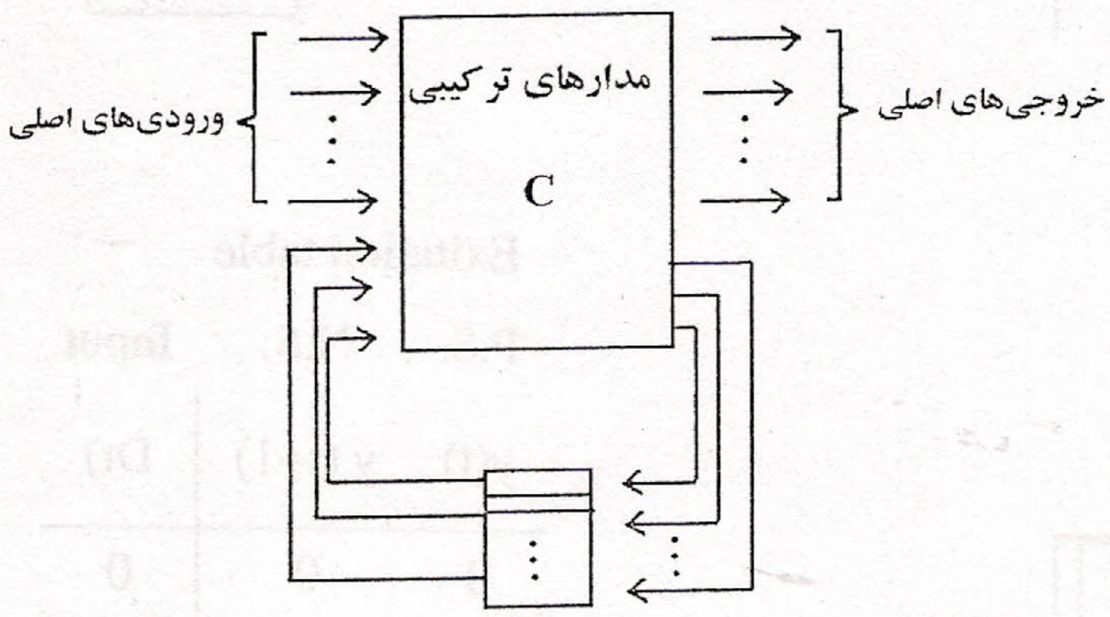
طراحی مدارهای ترتیبی

بهترین مدل برای طراحی مدارهای ترتیبی به‌وسیله هافمن (Huffman) ارائه شد که متشکل از 2 قسمت زیر می‌باشد:

۱. مدار ترکیبی C.

۲. مجموع‌های از عناصر حافظه M (همان FF ها).

که در آن خروجی نه تنها به ورودی‌های فعلی بلکه به ورودی‌های قبلی نیز بستگی دارد.



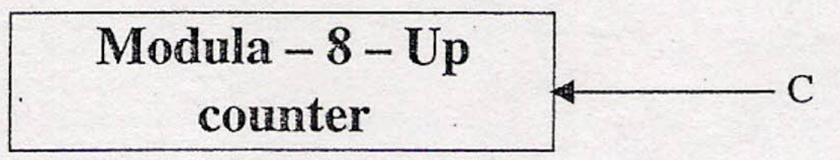
مزیت مدل هافمن در این است که طراحی مدارهای ترتیبی را به دو مرحله مستقل از هم تقسیم می‌کند.

۱) تعیین تعداد حالت درونی M و گذار حالات (رفتار). اگر تعداد حالات درونی n باشد، آن‌گاه تعداد FF های لازم مساوی $\lceil \log_2 n \rceil$ خواهد بود.

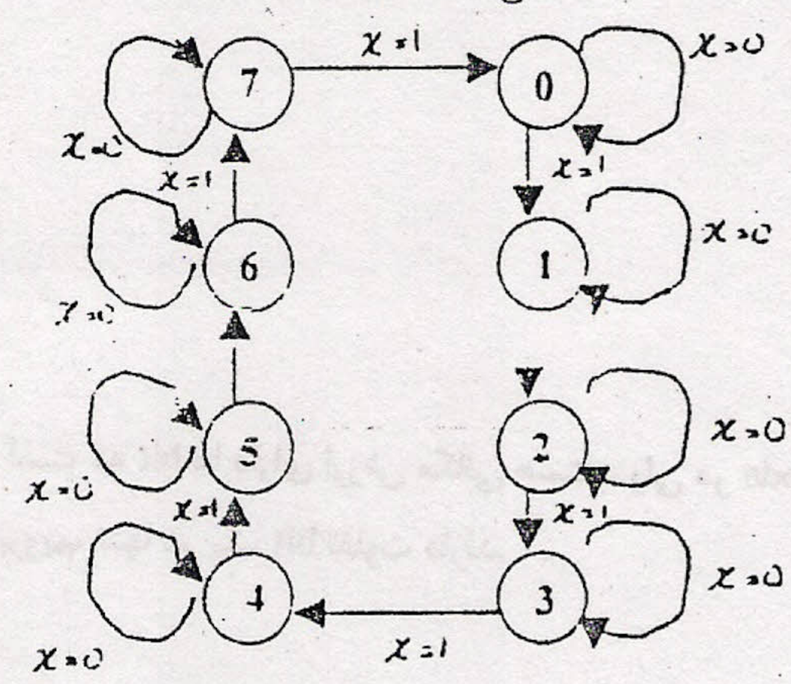
۲) طراحی مدار ترکیبی C برای تولید ورودی‌های محرک لازم به منظور گذار حالت FF ها و تولید خروجی اصلی مدار.

چند مثال ساده از مدارهای ترتیبی سنکرون:

مثال: مطلوب است طراحی یک شمارنده صعودی با پیمانه 8 به‌طوری که با دریافت هر سیگنال شمارش یک واحد به حالت شماره آن اضافه شود:



State diagram :

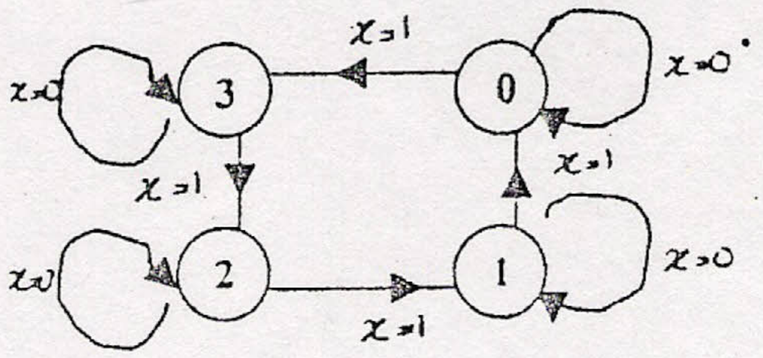


x	0	1
0	0	1
1	1	2
2	2	3
3	3	4
4	4	5
5	5	6
6	6	7
7	7	0

P.S — حالات جدید N.S

مثال: یک شمارنده نزولی با پیمانه "4" طراحی کنید به‌طوری که به ازا هر سیگنال شمارش یک واحد از حالت شماره آن کاسته شود.

x	0	1
1	0	3
1	1	0
2	2	1
3	3	2



رویه‌ای برای طراحی مدارهای ترتیبی سنکرون:

۱) از توصیف لفظی صورت مساله جدول حالات (State Table) را به‌دست می‌آوریم.

۲) به حالت جدول کد نسبت داده و در جدول درج می‌کنیم. در این مرحله جدول انتقال حالات (State Transition) به‌دست می‌آید.

۳) نوع و تعداد FF ها را مشخص می‌کنیم. مثلاً اگر n حالت متمایز وجود داشته باشد: آن‌گاه تعداد FFهای لازم = $\lceil \log_2 n \rceil$

۴) جدول تحریک یا Exitation Table را به‌دست می‌آوریم.

۵) عبارت‌های بولی برای ورودی‌های محرک و هم‌چنین برای خروجی‌های مدار را به‌دست می‌آوریم.

۶) پیاده‌سازی مدار (طراحی مدار) را انجام می‌دهیم.

توجه: دو نوع روش کد گذاری به شرح زیر مورد استفاده قرار می‌گیرند:

Digital	NBCD	Gray Code
0	0000	0000
1.	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

مزیت NBCD در این است که bit ها دارای ارزش مکانی هستند ولی در Gray code ها bit ها دارای ارزش مکـانی نیسـت ولـی از هـر code به code مجاور برویم، تنها در یک bit تفاوت دارند.

مثال: یک شمارنده صعودی با پیمانه 4 طراحی کنید به طوری که با دریافت هر سیگنال شمارش یک واحد به حالت شمار آن اضافه شود. برای طراحی مدار از T – FF استفاده نمایید.

1) S.T

y_1y_2 \ x	0	1
00	0	1
01	1	2
10	2	3
11	3	0

2) T.T

y_1y_2 \ x	0	1
00	y_1 y_2 0 0	y_1 y_2 0 1
01	0 1	1 0
10	1 0	1 1
11	1 1	0 0

3) n=4

$\lceil \log_2 4 \rceil = 2 - FF$

نوع T-FF

4) E.T.

y_1 y_2 \ x	0	1
00	T_1 T_2 0 0	T_1 T_2 0 1
01	0 1	1 1
10	1 0	0 1
11	1 1	1 1

5) عبارتهای بولی

y_1y_2 \ x	0	1
00	0	1
01	0	1
11	0	1
10	0	1

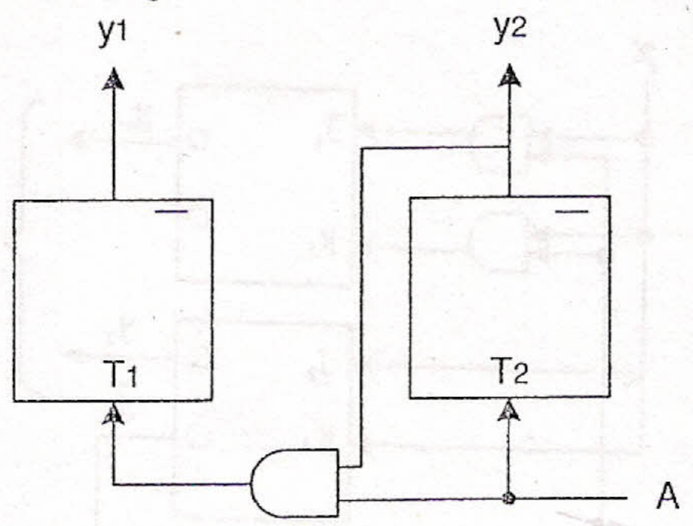
$T_2 = x$

y_1y_2 \ x	0	1
00	0	0
01	0	1
11	0	1
10	0	0

$T_1 = xy_2$

6)

نوع T - FF



مثال: یک شمارنده نزولی با پیمانه "4" طراحی کنید، به طوری که با دریافت هر سیگنال شمارش یک واحد از حالت شمار آن کاسته شود و برای طراحی مدار از JK-FF استفاده کنید.

1) S.T

NBCD	x=0	x=1
00 0	0	3
01 1	1	0
10 2	2	1
11 3	3	2

2) T.T

y_1y_2	x=0	x=1
00	$y_1\ y_2$ = 0 0	1 1
01	0 1	0 1
10	1 0	0 1
11	1 1	1 0

3) تعداد و نوع FF

نوع JK-FF:

$\lceil \log_2^4 \rceil = 2JK - FF$

4) $E.T_1$

y_1y_2	x=0	x=1
00	$J_1\ K_1\ J_2\ K_2$ = 0 – 0 –	1 – 1 –
01	0 – – –	0 – – –
10	– 0 – 1	– 0 – 1
11	– 0 0 –	– 1 1 –

5) عبارتهای بولی

y_1y_2	x=0	x=1
00	-	-
01	0	-
11	0	1
10	-	-

$K_2 = x$

y_1y_2	x=0	x=1
00	0	1
01	-	-
11	-	-
10	0	1

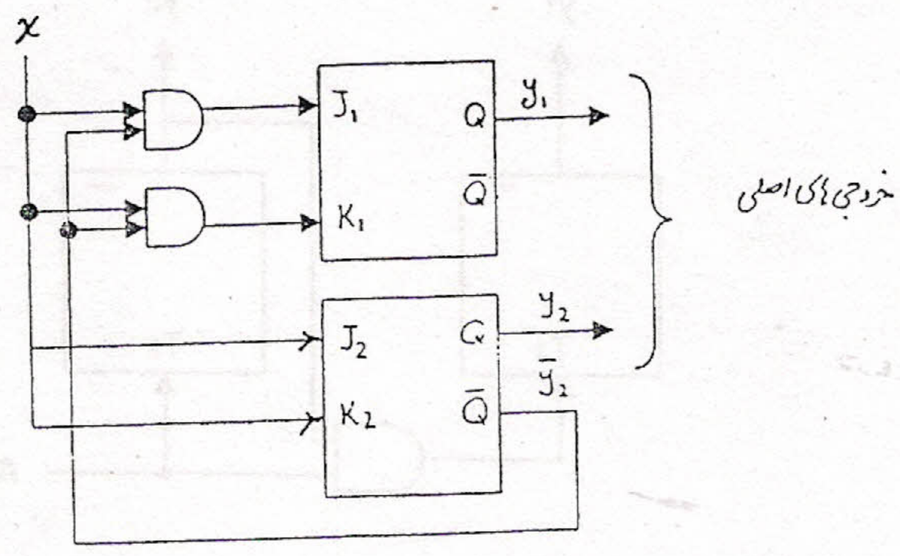
$J_2 = x$

y_1y_2	x=0	x=1
00	-	-
01	-	-
11	0	0
10	0	1

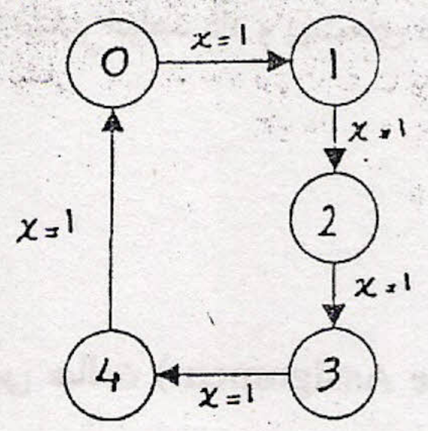
$K_1 = x\bar{y}_2$

y_1y_2	x=0	x=1
00	0	1
01	0	0
11	-	-
10	-	-

$J_1 = x\bar{y}_2$

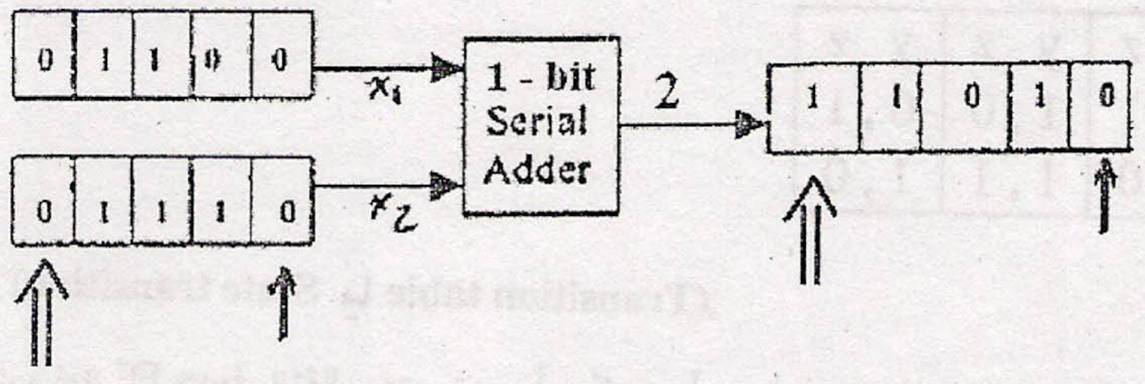


تمرین: با استفاده از SR-FF یک شمارنده با پیمانه "5" طرح کنید که عمل شمارش را به صورت زیر انجام دهد.



در تجزیه و تحلیل مدارهای ترتیبی اصطلاحاتی وجود دارد که با تعقیب یک مثال ساده با این اصطلاحات آشنا می‌شوید.

مثال: مطلوب است طراحی مدار جمع‌کننده سری تک بیتی (1-Bit Serial Adder)



با توجه به مثال فوق متوجه می‌شویم که در مدارهای ترتیبی بر خلاف مدارهای ترکیبی ورودی‌های یکسان ممکن است خروجی‌های متمایز ایجاد کنند:

۱) **حالات درونی (Internal State):** هر مدار دارای تعدادی حالات درونی محدود است که منظور از تجزیه و تحلیل مدارهای ترتیبی پیدا کردن تعداد همین حالات درونی است.

در این مثال فقط دو حالت داخلی وجود دارد:

State A: حالتی از مدار که در آن carry تولید نشود .

State B : حالتی از مدار که در آن carry تولید بشود.

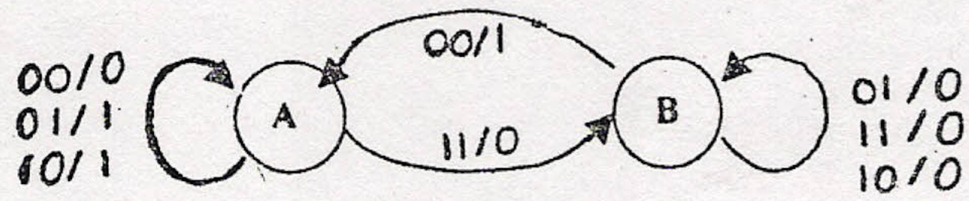
۲) **حالت فعلی (Present State)** : حالتی از مدار است که قبل از اعمال ورودی‌های خارجی (External Inputs) مدار در آن حالت قرار دارد.

۳) **حالت بعدی (Next State)** : حالتی از مدار است که تحت تاثیر ورودی‌های خارجی و حالت فعلی مدار به آن حالت انتقال خواهد یافت، بنابراین Next State تابعی از External Input و Present State است.

۴) **جدول حالات (State table)** : روش مناسبی است برای نشان دادن رفتار مدارهای ترتیبی که در آن هر سطر به یکی از حالت‌های درونی مدار و هر ستون به یکی از ترکیبات ورودی‌های خارجی مربوط می‌شود و درایه هر خانه حالت بعدی و خروجی حاصل را مشخص می‌کند که در مورد مثال فوق داریم:

y \ x_1x_2	00	01	11	10
A	A , 0	A , 1	B , 0	A , 1
B	A , 1	B , 0	B , 1	B , 0

۵) نمودار حالات (State Diagram) : گرافی که در آن هر State را با یک دایره نشان می‌دهند و هر انتقال از یک حالت به حالت دیگر را یال‌ها نشان خواهد داد و بیت‌های نوشته شده بر روی فلش به ترتیب ورودی‌ها و خروجی‌ها را نشان ی‌دهد.



۶) تخصیص حالت (State Assignment): تخصیص حالت مرحله‌ای است که در آن حالت‌های یک وسیله فیزیکی مانند FF به حالت‌های درونی مدار اختصاص داده می‌شود که در این مثال از D – FF استفاده شده و جدول حالت به جدول انتقال به شرح زیر تبدیل می‌شود.

y \ x_1x_2	00	01	11	10
0	y z 0 , 0	y z 0 , 1	y z 1 , 0	y z 0 , 1
1	0 , 1	1 , 0	1 , 1	1 , 0

۷) جدول انتقال حالات (State transition یا Transition table):

با استفاده از تخصیص حالات و نوع FF جدول انتقال به‌دست می‌آید که در آن درایه هر خانه فقط Next State را نشان می‌دهد.

y \ x_1x_2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

۸) جدول تحریک (Exitation table): با استفاده از جدول حالات و نوع FF می‌توان جدول تحریک را به‌دست آورد و در صورتی که نوع FF انتخاب شده D-FF باشد، آن‌گاه Transition table = Exitation table خواهد بود.

y \ x_1x_2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

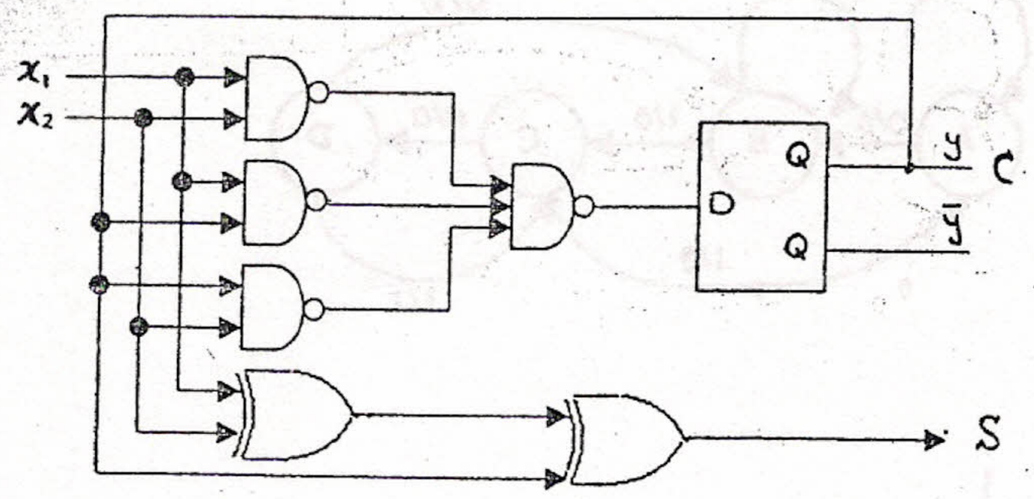
$D = x_1x_2 + x_1y + x_2y = x_1x_2.x_1y.x_2y$

y \ x_1x_2	00	01	11	10
0	0	(1)	0	(1)
1	(1)	0	(1)	0

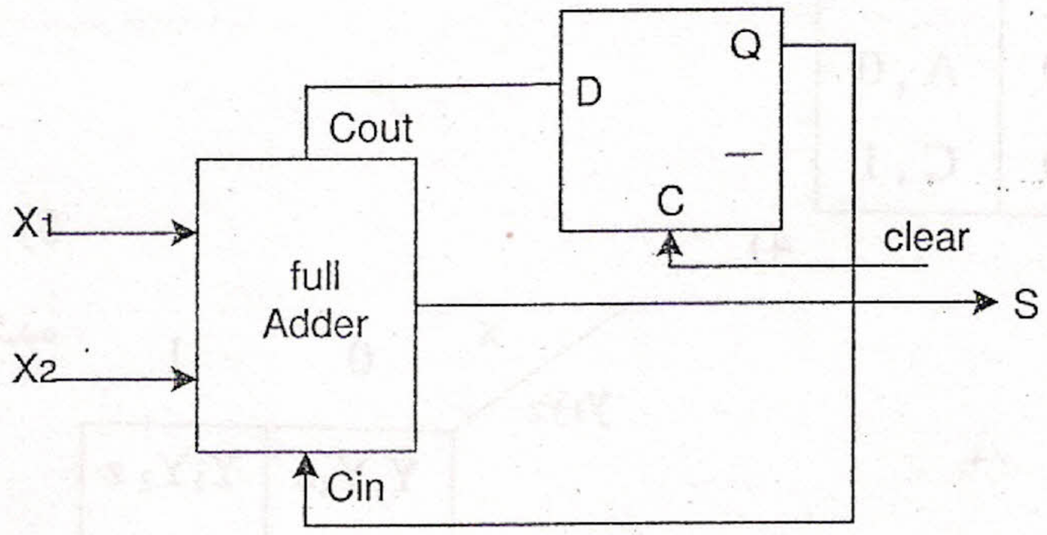
$S = x_1x_2y + x_1\bar{x}_2\bar{y} + \bar{x}_1x_2\bar{y} + \bar{x}_1\bar{x}_2y = x_1 \oplus x_2 \oplus y$

عبارت بولی متناظر با ورودی‌های محرک FFها و خروجی اصلی مدار را به‌دست آوریم.

۹) با استفاده از عبارات بولی به‌دست آمده از مرحله 8 و gate ها و FF ها می‌توان طراحی مدار را کامل کرد.

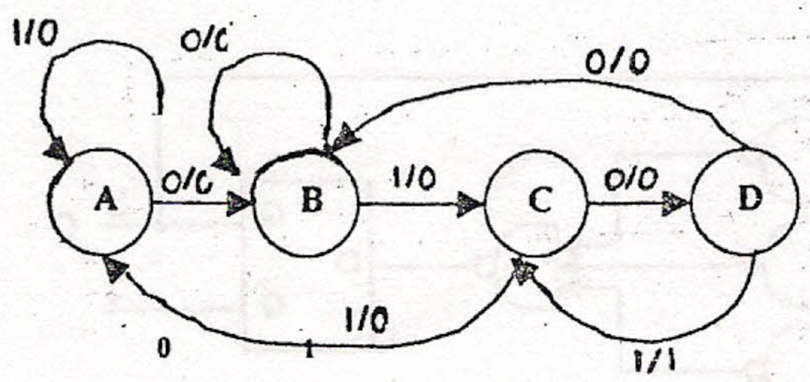


طراحی این مدار با استفاده از مدار مجتمع D-FF Full Adder به صورت زیر می باشد:



مثال: مطلوب است طراحی مدار یک Sequence Detector که بتواند دنباله 0101 را تشخیص دهد.

1) State Diagram:



2)

		x 0	1
00	A	B , 0	A , 0
01	B	B , 0	C , 0
11	C	D , 0	A , 0
10	D	B , 0	C , 1

3)

$\lceil \log 4 \rceil = 2$

2D-FF

4)

y_1y_2 \ x	0	1
	Y_1Y_2z	Y_1Y_2z
00	0 1, 0	0 0, 0
01	0 1, 0	1 1, 0
11	1 0, 0	0 0, 0
10	0 1, 0	1 1, 1

5)

چون از D-FF استفاده شده است

E.T=T.T

6)

y_1y_2 \ x	0	1
00	0	0
01	0	①
11	①	0
10	0	①

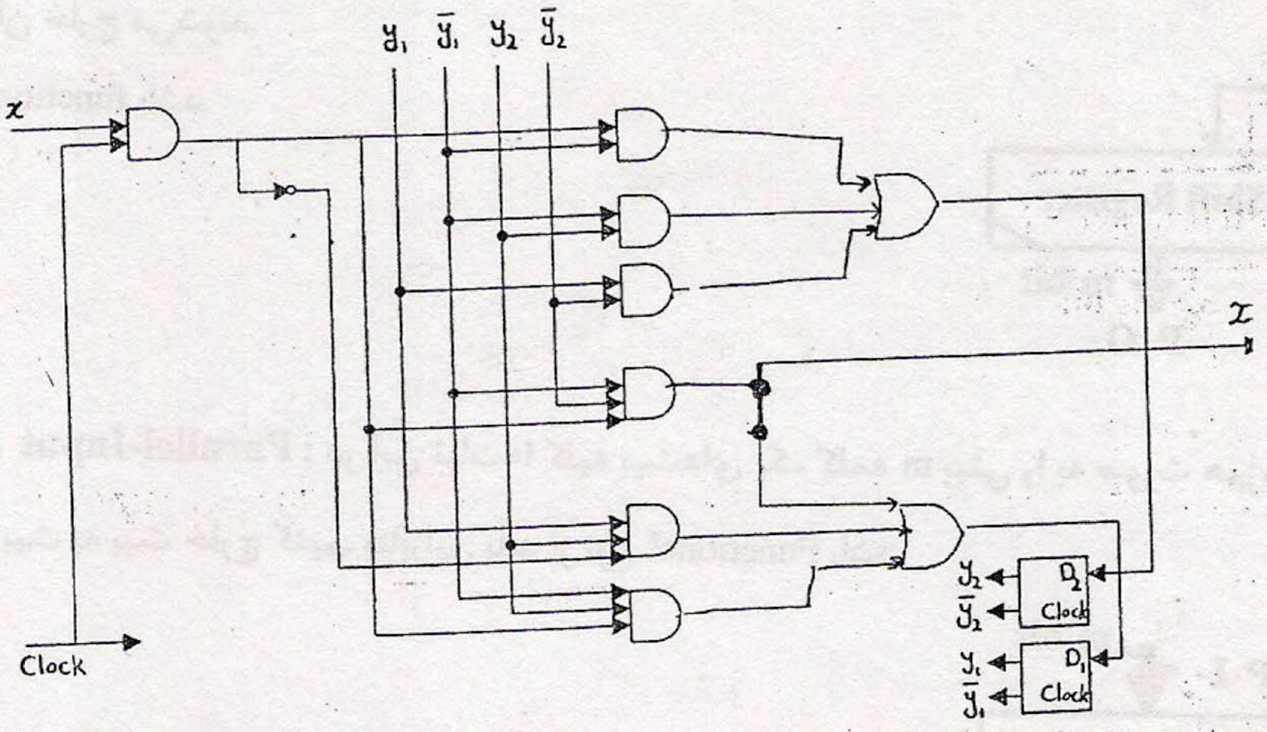
$D_1 = xy_1\overline{y}_2 + \overline{x}y_1y_2 + x\overline{y}_1y_2$

y_1y_2 \ x	0	1
00	1	0
01	1	1
11	0	0
10	1	1

$D_2 = x\overline{y}_1 + \overline{y}_1y_2 + y_1\overline{y}_2$

y_1y_2 \ x	0	1
00	0	0
01	0	0
11	0	0
10	0	①

$Z = X y_1 \overline{y}_2$



تمرین: مطلوب است طراحی مدار یک Sequence Detector که بتواند دنباله 100 را تشخیص بدهد.

ثبات‌ها : *(Registers)*

ثبات m بیتی مجموعه مرتبی است از m عدد FF که برای ذخیره کردن یک کلمه m بیتی به کار می‌رود که در آن هر بیت از کلمه در یک FF ذخیره می‌شود. چون اطلاعات ذخیره شده در داخل ثبات به عنوان یک کلمه واحد تلقی می‌شود، بنابراین Signalهای Clear و Clock در همه FF ها مشترک خواهد بود.

انواع ثبات‌ها:

ثباتها کلاً بر دو نوع هستند:

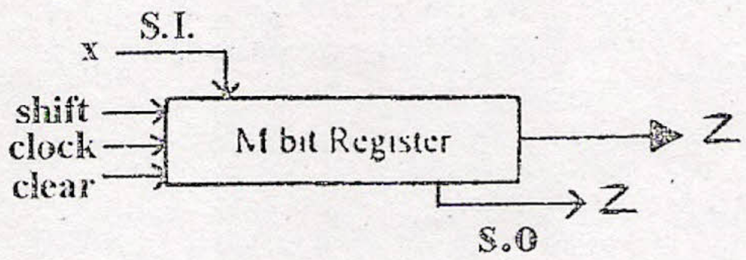
۱) Storage Register (انباره‌ای) که فقط قادر به ذخیره کردن اطلاعات می‌باشند.

۲) Functional Register (عملیاتی)که علاوه بر ذخیره نمودن اطلاعات، قادرند روی محتوای خود محاسباتی نیز انجام دهند. مانند Shift Register

با توجه به روش ورود و خروج اطلاعات ثبات‌ها به 4 گروه زیر تقسیم می‌شوند:

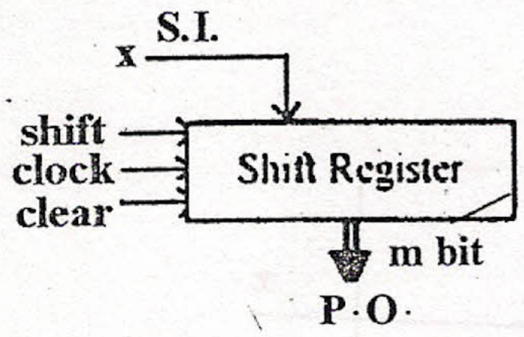
۱) **Serial Input – Serial Output :**

در این ثبات اطلاعات در هر لحظه 1-bit از یک سو وارد و از سوی دیگر در هر لحظه 1-bit خارج می‌شود، بنابراین باید از نوع functional باشد.

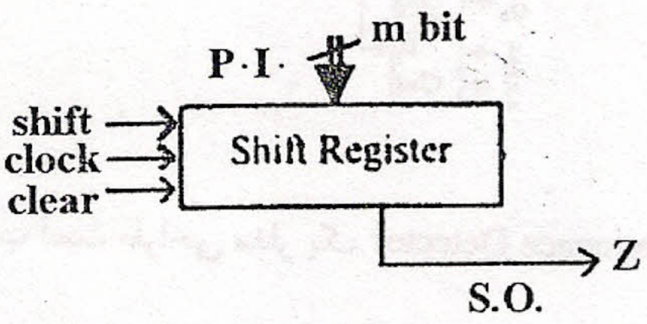


۲) **Serial Input – Parallel Output** : در این ثبات‌ها اطلاعات در هر لحظه یک bit وارد ولی کلیه بیت‌های کلمه m بیتی به صورت موازی و هم‌زمان خارج می‌شوند.

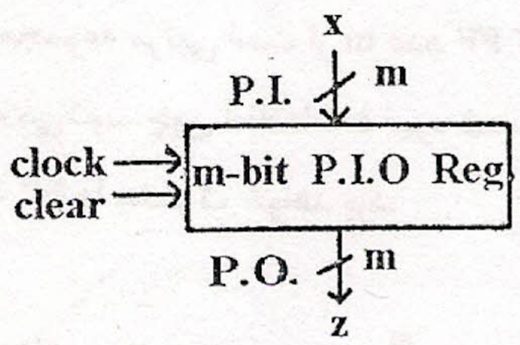
بنابراین باید از نوع functional باشد.



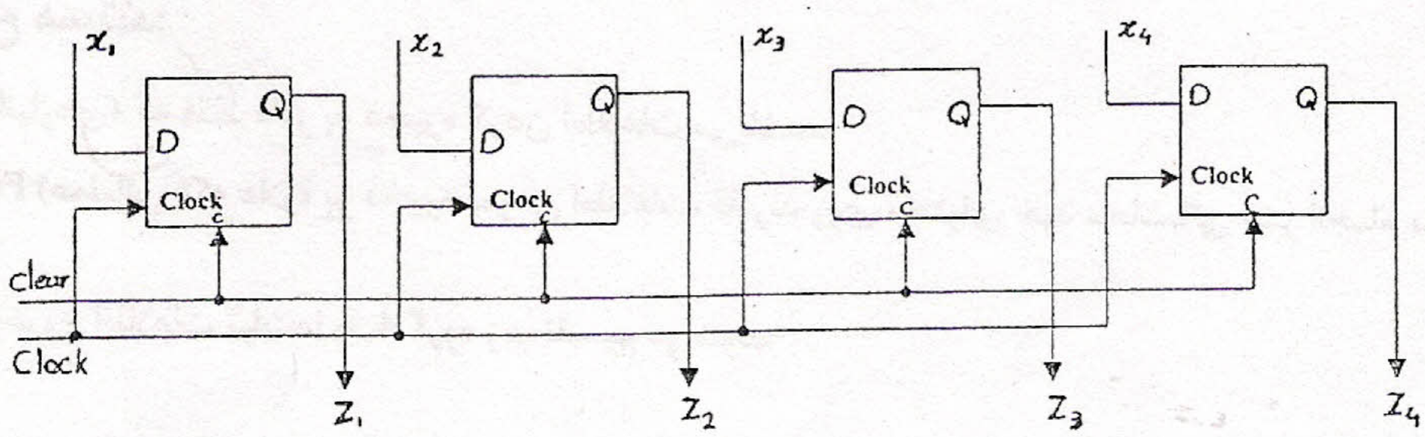
۳) **Parallel-Input Serial-Output** : در این ثبات‌ها کلیه بیت‌های یک کلمه m بیتی را به صورت هم‌زمان وارد می‌کنیم ولی برای خارج کردن باید بیت به بیت خارج کنیم. بنابراین باید از نوع Functional باشد.



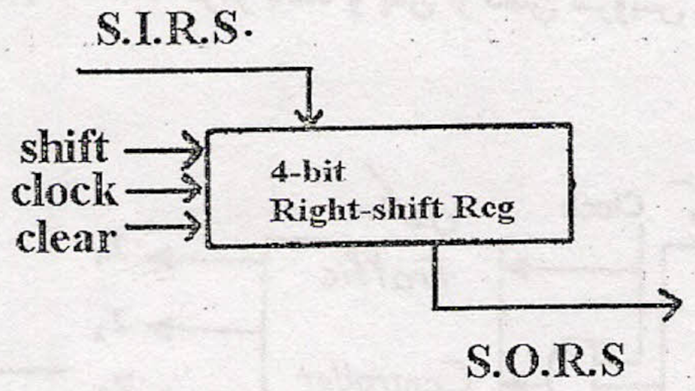
۴) **Parallel Input – Parallel Output** : کلیه بیت‌های یک کلمه m بیتی را به صورت هم‌زمان وارد و به‌طور هم‌زمان می‌توانیم از آن خارج کنیم. بنابراین از نوع Storage می‌باشد.

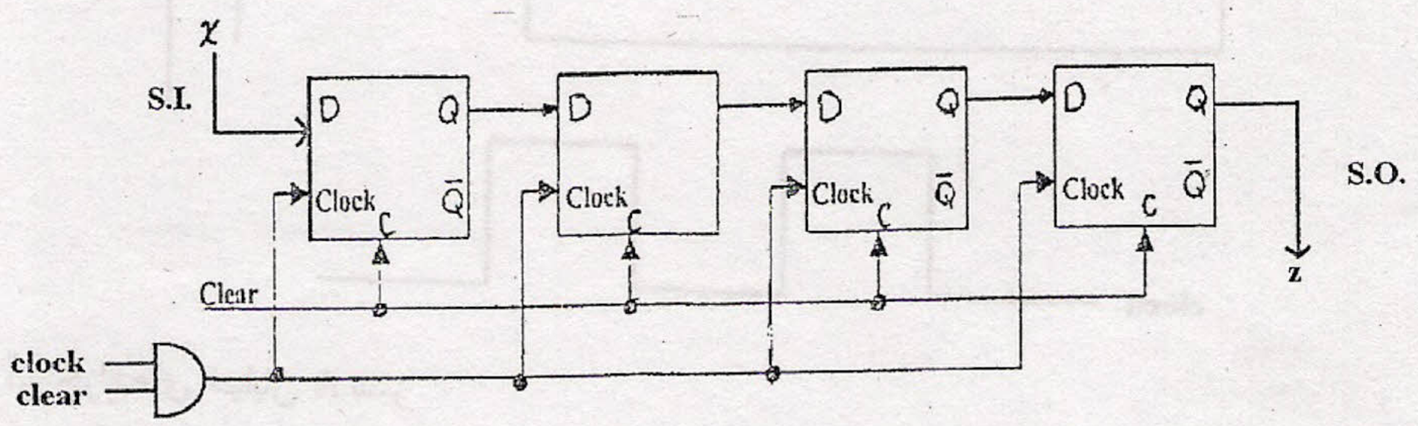


مثال: با استفاده از D – FF ، یک 4- Bit Parallel input Parallel output Register طرح کنید.



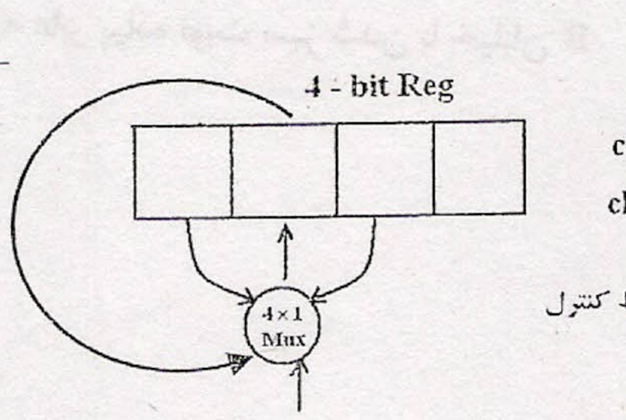
مثال: با استفاده از D-FF یک 4- Bit Right Shift Register طرح کنید.

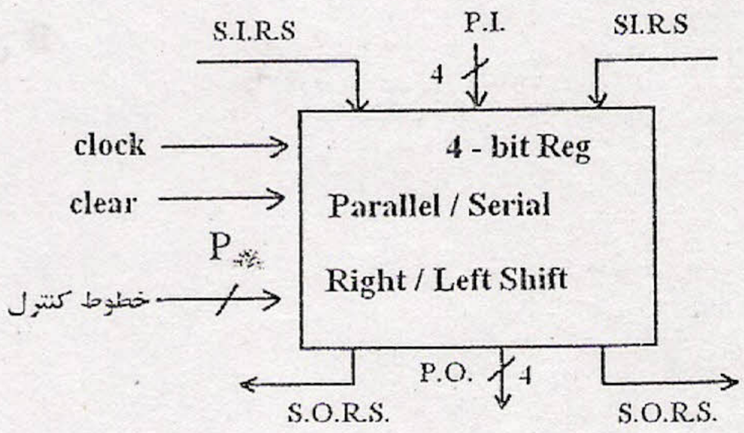


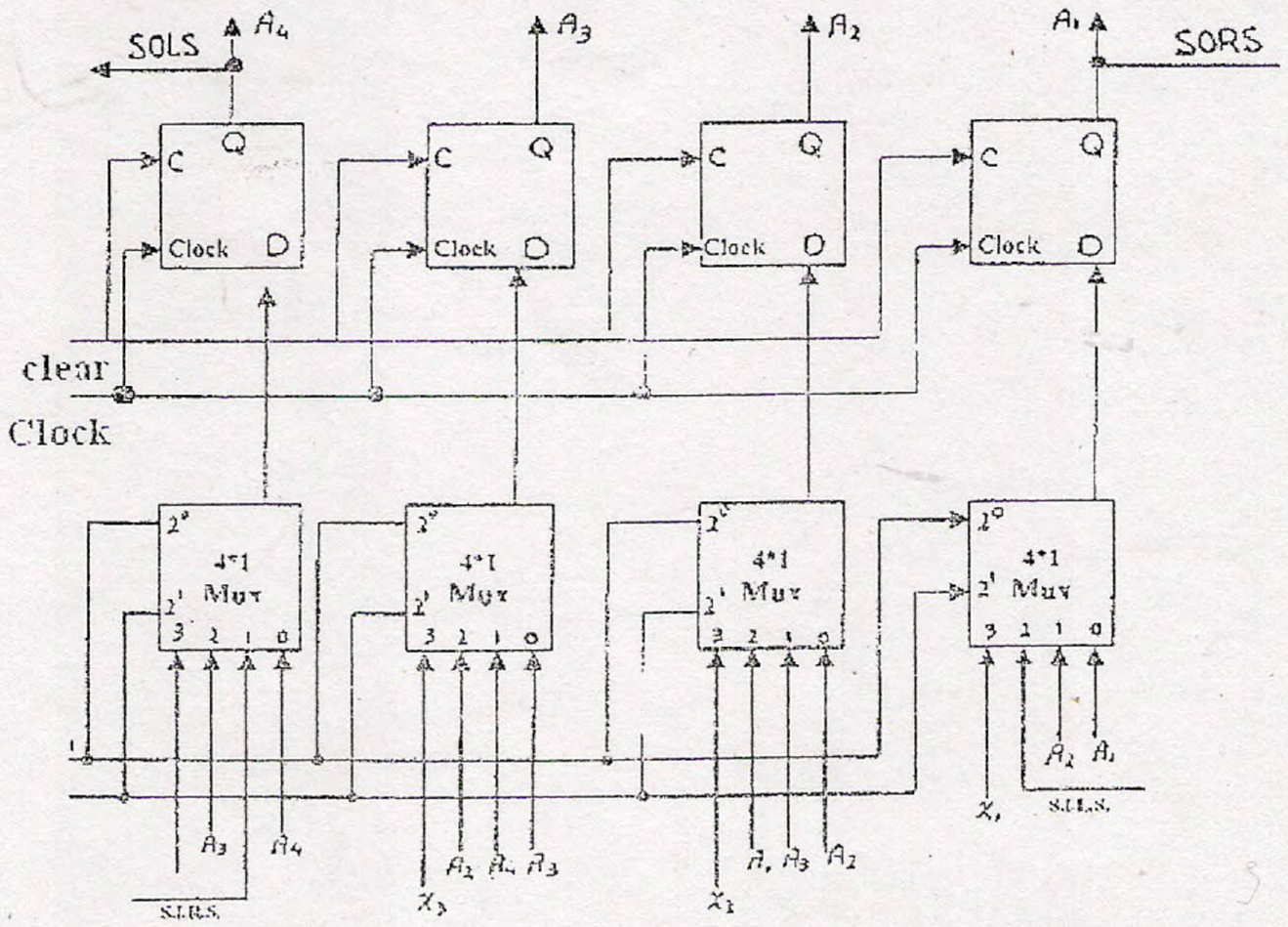


مثال: با استفاده از D-FF و MUX یک ثبات 4 بیتی با قابلیت تغییر مکان به راست، به چپ و بار موازی طرح کنید.

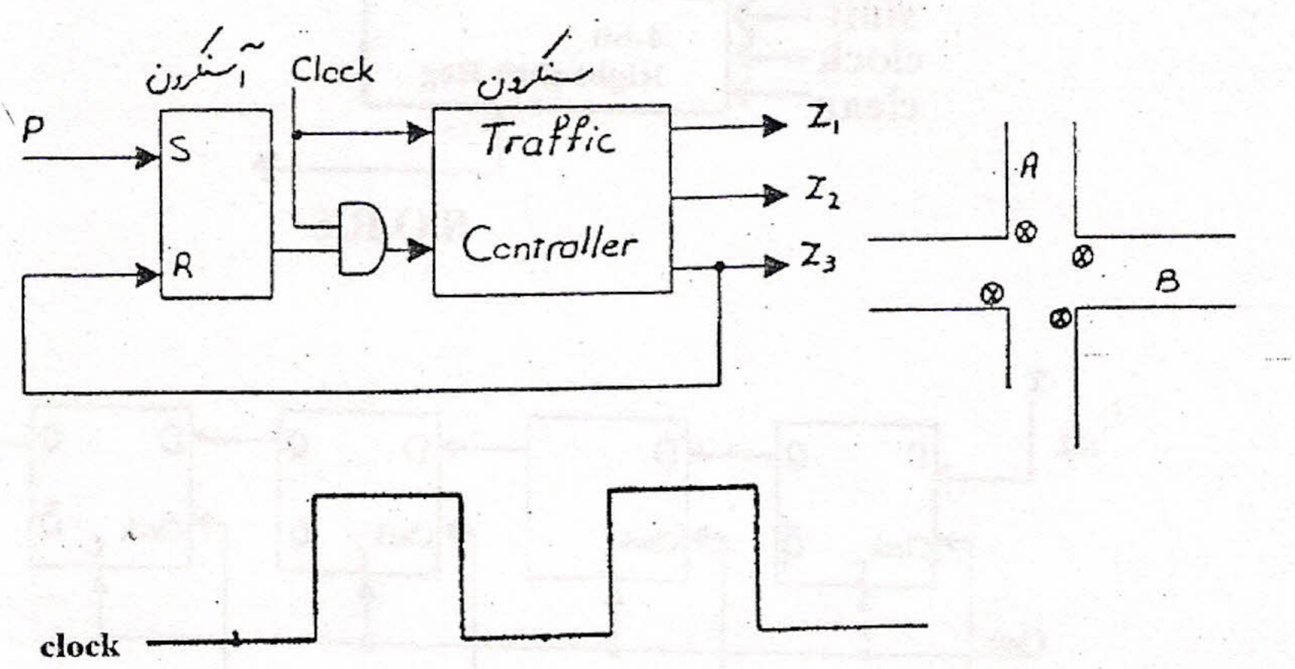
S_1S_2	Action
00	Unchanged
01	Right shift
10	Left shift
11	Parallel load







مثال: می‌خواهیم یک کنترل کننده چراغ راهنمایی برای تقاطع دو خیابان A , B طرح کنیم به طوری که چراغ خیابان A یک clock – period سبز یک clock – period قرمز باشد و همین سیکل برای خیابان B تکرار شود. اگر کلید عابر پیاده P زده شود، هر چهار چراغ به مدت دو clock – period قرمز باشد و پس از دادن سرویس به عابر پیاده، چراغ‌ها سیکل عادی خود را تکرار کنند.



A: حالتی از مدار که در آن چراغ‌های خیابان A سبز

B: حالتی از مدار که در آن چراغ‌های خیابان B سبز

C: حالتی از مدار که پس از دادن سرویس به عابر پیاده نوبت سبز شدن با خیابان A

D: حالتی از مدار که پس از دادن سرویس به عابر پیاده نوبت سبز شدن با خیابان B

1) State Table

P	0	1
	$N.S.z_1.z_2.z_3$	$N.S.z_1.z_2.z_3$
00 A	B , 1 0 0	C ,0 0 0
01 B	A 0 1 0	D ,0 0 0
10 C	- ,- - -	A ,0 0 1
11 D	- ,- - -	B ,0 0 1

2) Number of FF

$n=4$

$\lceil \log_2 4 \rceil = 2$

2T – FF

3) Transition Table

y_1y_2 \ P	0	1
	$Y_1Y_2 , z_1z_2z_3$	$Y_1Y_2 , z_1z_2z_3$
00	01 , 100	10 , 000
01	00 , 010	11 , 000
10	-- , ---	00 , 001
11	-- , ---	01 , 001

4) Exitation Table

y_1y_2 \ P	0	1
	$T_1T_2 z_1z_2z_3$	$T_1T_2 z_1z_2z_3$
00	0 1 , 1 0 0	1 0 , 0 0 0
01	0 1 , 0 1 0	1 1 , 0 0 0
10	- - , - - -	1 0 , 0 0 1
11	- - , - - -	1 0 , 0 0 1

۵. عبارت بولی متناظر با ورودی‌های محرک و خروجی اصلی مدار:

y_1y_2 \ P	0	1
00	0	1
01	0	1
11	-	1
10	-	1

$T_1 = P$

P	0	1
	1	1
	1	1
	-	1
	-	1

$T_2 = 1$

P	0	1
	1	0
	0	0
	-	0
	-	0

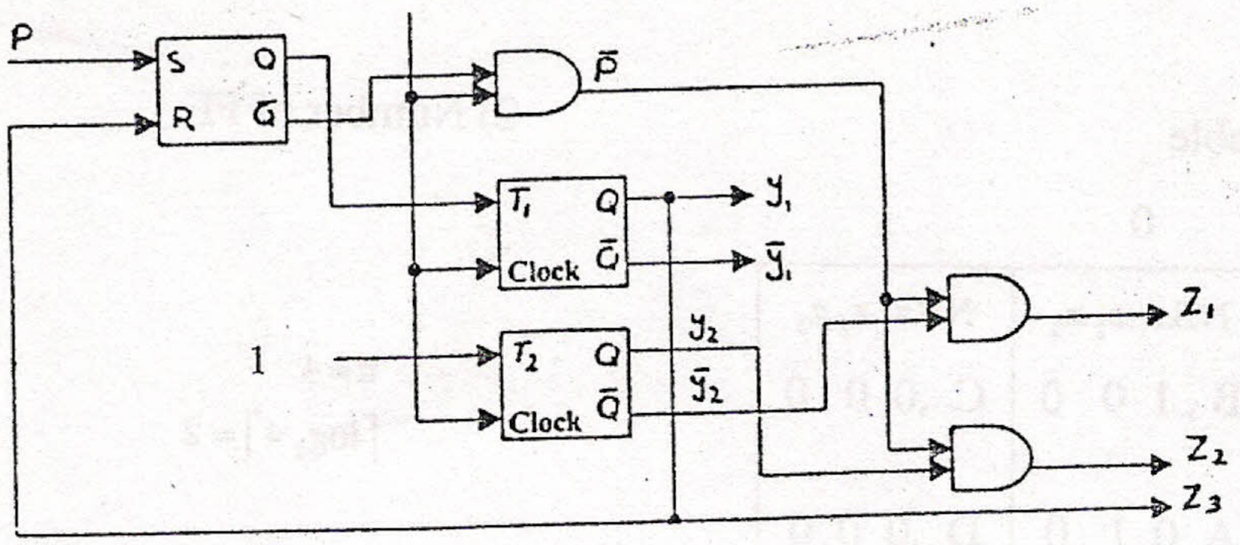
$Z_1 = \overline{P}\overline{y}_2$

P	0	1
	0	0
	1	0
	-	0
	0	0

$Z_2 = Py_2$

P	0	1
	0	0
	0	0
	-	1
	-	1

$Z_3 = y_1$

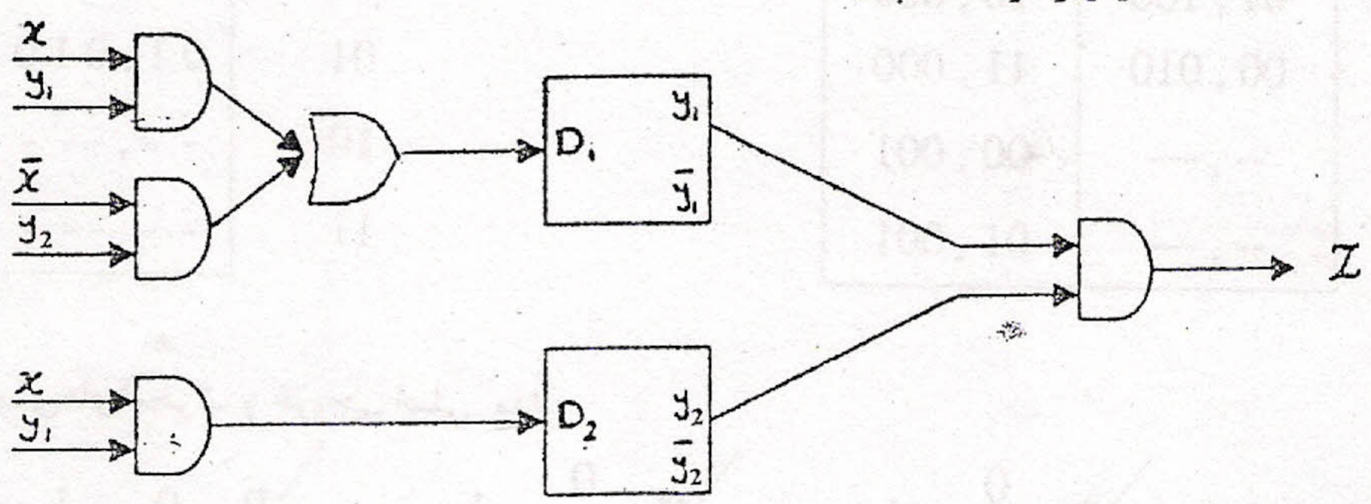


تمرین: یک کنترل کننده آسانسور برای ساختمان ۳ طبقه طرح کنید به طوری که اگر تقاضایی نباشد، آسانسور در طبقه دوم بایستد و اگر تقاضا به‌طور هم‌زمان از طبقه اول و سوم برسد، حق تقدم با طبقه سوم باشد.

پیدا کردن صورت مساله از مدار طراحی شده

مراحل: مدار طراحی شده ① ⇐ عبارت‌های بولی ② ⇐ جدول تحریک ③ ⇐ جدول گذار حالات ④ ⇐ جدول حالات ⑤ ⇐ صورت

مثال: فرض کنید مداری به شکل زیر در دست باشد:



۱) با استفاده از نمودار بلوکی داریم:

$$\text{عبارتهای بولی} \begin{cases} D_1 = xy_1 + \bar{x}y_2 \\ D_2 = xy_1 \\ z = y_1y_2 \end{cases}$$

۲) از روی عبارت بولی Exitation table را به‌دست می‌آوریم:

y_1y_2 \ x	0	1
	D_1D_2, Z	D_1D_2, Z
00	0 0 , 0	0 0 , 0
01	1 0 , 0	0 0 , 0
11	1 0 , 1	1 1 , 1
10	0 0 , 0	1 1 , 0

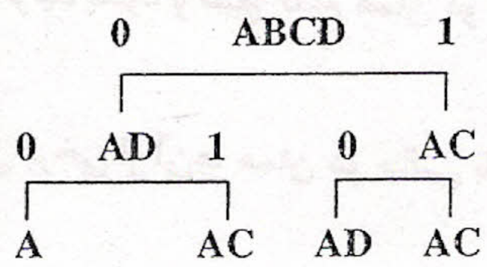
۳) از روی Exitation table می‌توان Transition table را به‌دست آورد چون از D – FF استفاده شده است. TT=ET

y_1y_2 \ x	0	1
	Y_1Y_2,Z	Y_1Y_2,Z
A 00	0 0, 0	0 0, 0
B 01	1 0, 0	0 0, 0
C 11	1 0, 1	1 1, 1
D 10	0 0, 0	1 1, 0

۴) از روی Transition table می‌توان State table را به‌دست آورد.

y_1y_2 \ x	0	1
A	A,0	A,0
B	D,0	A,0
C	D,1	C,1
D	A,0	C,0

تعریف: دنباله سنکرون Synchronizing Sequence یک دنباله ورودی است که اگر وارد مدار شود صرف‌نظر از این‌که مدار از قبل در چه حالتی باشد، آن‌را به یک حالت کاملاً مشخص انتقال می‌دهد.



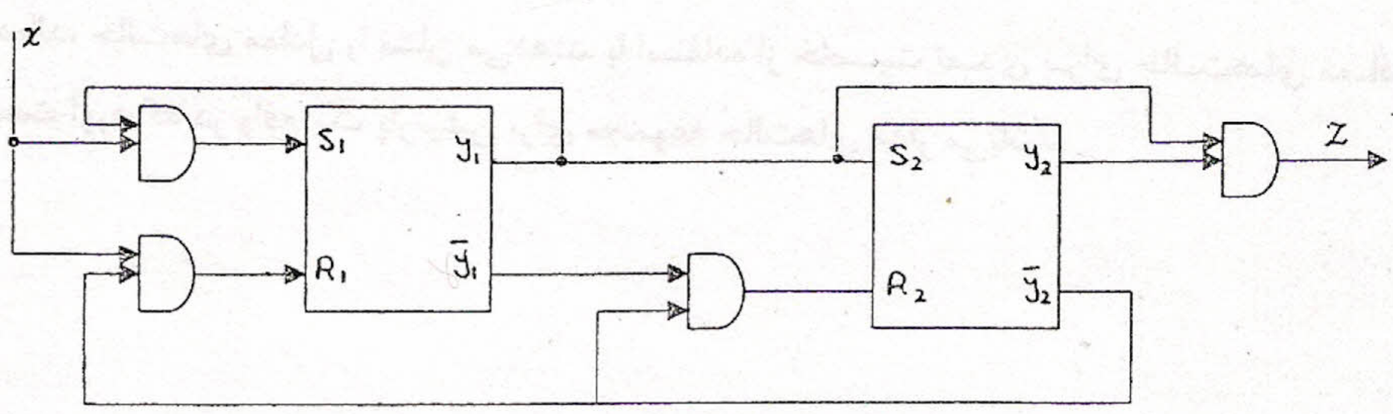
$A \overset{00}{\Rightarrow} A$

$B \overset{00}{\Rightarrow} A$

$C \overset{00}{\Rightarrow} A$

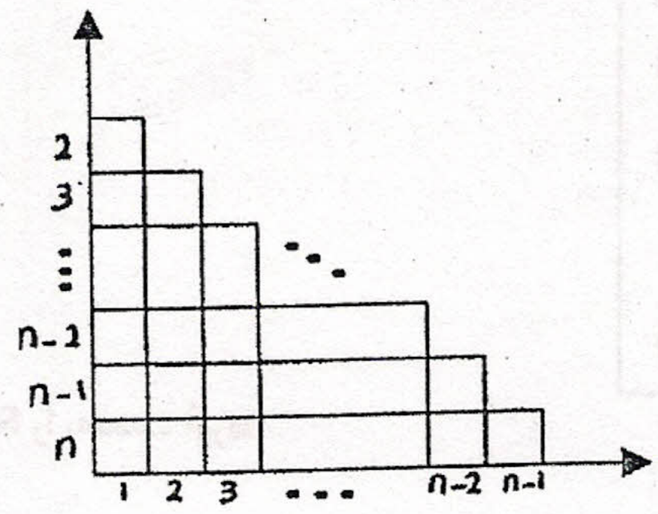
$D \overset{00}{\Rightarrow} A$

تمرین : صورت مساله مدار زیر را به‌دست آورید:



روش ساده کردن جدول حالات:

برای ساده کردن جدول حالات می‌توان از "Pair Chart" یا زوج‌نما استفاده نمود. برای جدول n حالتی فرم کلی "Pair Chart" به صورت زیر می‌باشد که در آن برای هر زوج از حالات مانند (q_i, q_j)، اگر $i < j$ باشد، آن‌گاه در "Pair Chart" یک خانه واقع در ستون iام و سطر jام وجود خواهد داشت.



هدف "Pair Chart" پیدا کردن حالتهای معادل می‌باشد. دو حالت A , B را معادل گویند، هر گاه به ازای کلیه ترکیبات ورودی دو حالت A , B دارای خروجی‌های یکسان باشند.

مراحل درج اطلاعات و ساده کردن *Pair – Chart* :

a- هر دو حالت را با هم مقایسه می‌کنیم. در صورتی‌که خروجی‌های یکسان نباشند، در خانه مربوطه علامت × درج می‌کنیم.

b- هر دو حالت را با هم مقایسه می‌کنیم. در صورتی‌که خروجی‌ها یکسان باشد، چهار حالت رخ می‌دهد.

۱ـ در صورتی که بیش از یک N. S (Next – State)تولید بشود و هیچ کدام همان دو حالتی نباشند که با هم مقایسه می‌شوند، همگی را در خانه مربوطه می‌نویسیم.

۲ـ در صورتی که بیش از یک . N .S تولید بشود ولی برخی از آن‌ها همان دو حالتی باشند که با هم مقایسه می‌شوند، به جز آن‌ها بقیه را در خانه مربوطه می‌نویسیم.

۳ـ در صورتی که بیش از یک . N . S تولید بشود ولی همگی آن ها همان دو حالتی باشند که با هم مقایسه می‌شوند، در خانه مربوطه علامت ✓ درج می‌کنیم.

۴ـ در صورتی که بیش از یک . N . S تولید نشود، در خانه مربوطه علامت ✓ درج می‌کنیم.

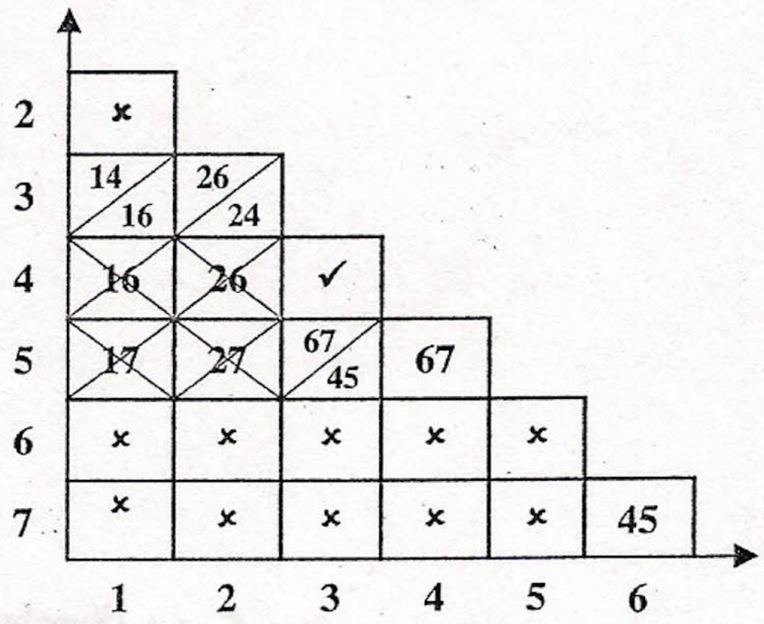
c- جدول را برای درج علامت × جدید دوباره بررسی می‌کنیم.

(در این مرحله اگر محتوی خانه‌ای چنان باشد که خانه مربوطه به آن در Pair – Chart قبلاً علامت × خورده است، در مرحلهٔ C آن خانه را نیز علامت × می‌زنیم)

d-خانه‌هایی که علامت ضرب‌در نخورده‌اند، حالت‌های معادل را نشان می‌دهند. با استفاده از خاصیت تعدی برای حالت‌های معادل می‌توان Minimal Equivalent را به‌دست آورد که در واقع یک پارتیشن برای مجموعه حالت‌های مدار می‌باشد.

مثال: جدول حالات زیر را ساده کنید.

x	0	1
1	0	1,1
2	,0	2,1
4	,0	4,1
5	,0	4,1
6	,0	5,1
7	,0	6,0
8	,0	7,0



$1 \equiv 2 \Rightarrow \{1,2\} : A$

$3 \equiv 4$

$3 \equiv 5 \Rightarrow \{3,4,5\} : B$

$4 \equiv 5 \Rightarrow \{6,7\} : C$

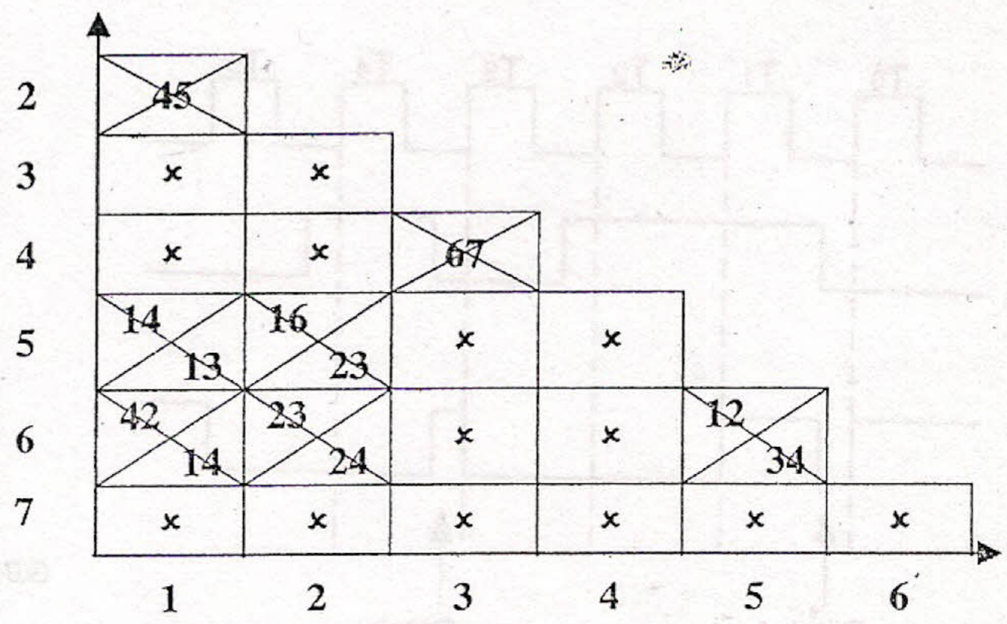
$\{\{1,2\},\{3,4,5\},\{6,7\}\} = \{1,2,3,4,5,6,7\}$

و جدول حالات خلاصه شده جدید به صورت زیر خواهد بود:

	x	0	1
{1,2}	A	A,0	A,1
{3,4,5}	B	C,0	B,1
{6,7}	C	B,0	C,0

مثال: جدول حالات زیر را ساده کنید.

x	0	1
1	4,0	1,0
2	5,0	2,0
3	6,1	3,1
4	7,1	4,1
5	1,0	3,0
6	2,0	4,0
7	7,1	5,0



چون همه خانه‌ها علامت ضرب‌در خورده است، بنابراین نمی‌توان این جدول را ساده کرد.

تمرین: جدول حالات زیر را ساده کنید.

x	0	1
1	1,0	4,0
2	2,0	5,0
3	3,1	6,0
4	1,1	4,1
5	2,0	5,1
6	3,0	6,1

تمرین: برای جدول کاملا مشخص شده زیر، جدول می‌نیمم را به‌دست آورید.

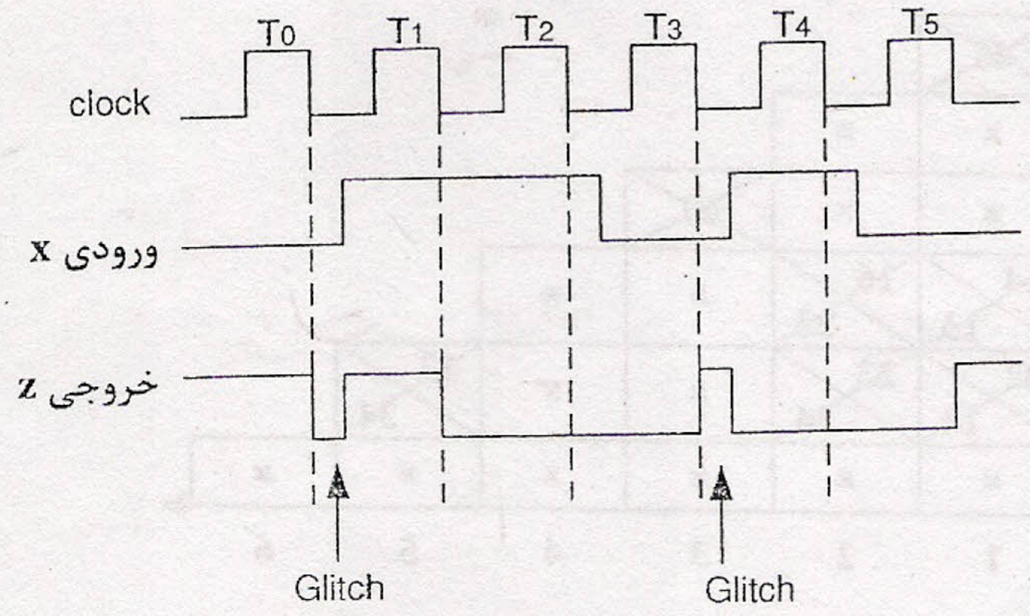
x	0	1
1	8,1	7,1
2	3,0	5,0
3	2,0	1,0
4	5,1	8,0
5	6,0	4,1
6	5,1	3,0
7	1,1	8,0
8	4,0	6,1

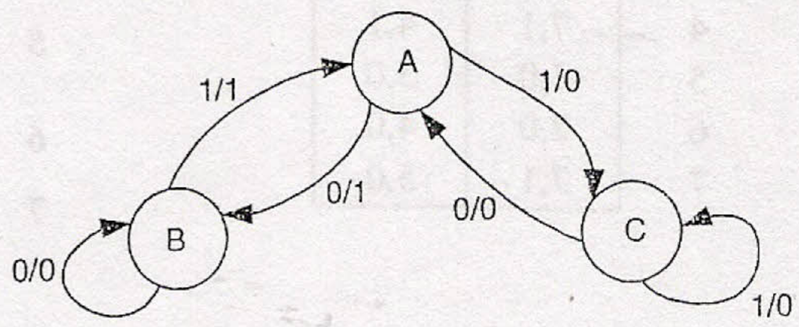
مدارهای ترتیبی همگام (Synchronous) به دو گروه تقسیم می‌شوند:

۱ـ مدارهای Moore : که در آن خروجی تنها به حالت فعلی مدار بستگی دارد یعنی با تغییر در ورودی خروجی مدار پایدار می‌ماند و در نتیجه در خروجی مدارهای Moore هرگز مقدار ناخواسته (Glitch) به طور موقتی ظاهر نمی‌شود، ولی نسبت به مدارهای Mealy به تعداد حالات بیشتری نیاز دارد.

۲ـ مدارهای Mealy : که در آن خروجی، تابعی از حالت فعلی و ورودی‌های مدار می‌باشد. توجه کنید در مدار Mealy هر موقع که ورودی مدار یا حالت مدار تغییر کند، خروجی مدار می‌تواند تغییر یابد. بدین دلیل ممکن است مقدار ناخواسته (Glitch) در خروجی مدار میلی دیده شود. پس در مدار میلی باید مراقب باشیم که خروجی‌ها را تنها در حالت پایدار و پس از تغییر در ورودی نگاه کنیم.

مثال: با فرض State Diagram زیر، حالت شروع مدار A و ورودی $x = 0$، خروجی موجی مدار را به ازای ورودی موجی به‌دست آورید.





به‌طوری که در خروجی مدار دیده می‌شود، دو مقدار ناخواسته (Glitch) یکی صفر و دیگری یک ایجاد شده است.

تمرین ـ چگونه می‌توان Moore یا Mealy بودن مدار را از روی State Diagran تشخیص داد.