معماری کامپیوتر

(فصل اول)

۱) گیت AND :

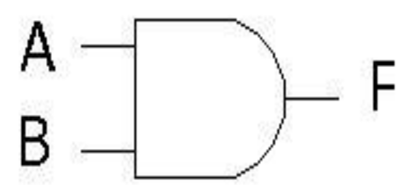
تابع جبری :

$$F = AB$$

جدول درستی :

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

سمبل گرافیکی :



۲) گیت OR :

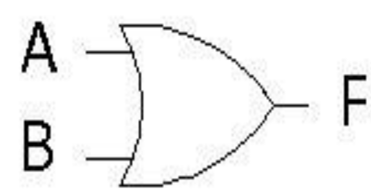
تابع جبری :

$$F = A+B$$

جدول درستی :

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

سمبل گرافیکی :

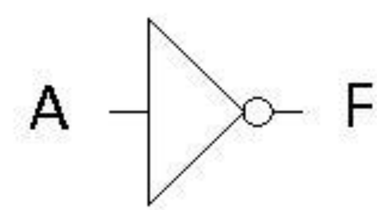


۳) گیت NOT:

تابع جبری :

$$F = A'$$

سمبل گرافیکی :



جدول درستی :

A	F
0	1
1	0

۴) گیت Buffer:

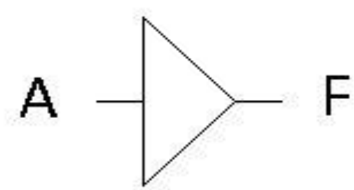
تابع جبری :

$$F = A$$

جدول درستی :

A	F
0	0
1	1

سمبل گرافیکی :



۵) گیت NAND :

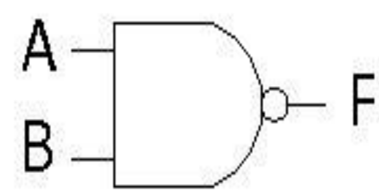
تابع جبری :

$$F = (AB)'$$

جدول درستی :

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

سمبل گرافیکی :



۶) گیت NOR :

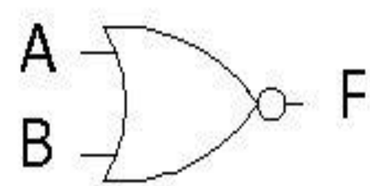
تابع جبری :

$$F = (A+B)'$$

جدول درستی :

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

سمبل گرافیکی :



۷) گیت XOR :

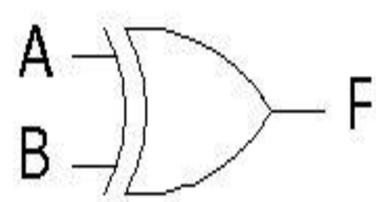
تابع جبری :

$$F = A'B + AB'$$

جدول درستی :

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

سمبل گرافیکی :



۸) گیت XNOR :

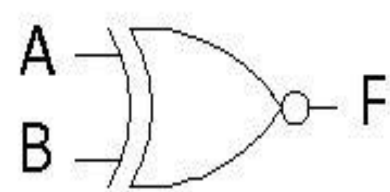
تابع جبری :

$$F = A'B'+AB$$

جدول درستی :

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

سمبل گرافیکی :



انواع مدارهای منطقی:

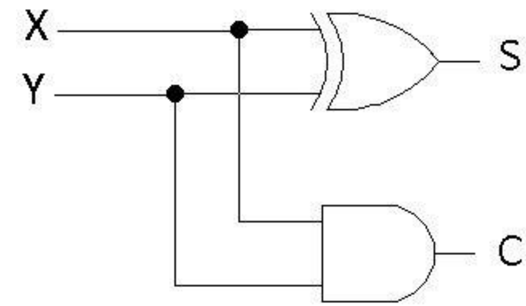
۱) مدار ترکیبی : حافظه ندارد.   ۲) مدار ترتیبی : حافظه دارد.

مدار ترکیبی به مدارهایی گفته می‌شود که خروجی در هر لحظه به ورودی در همان لحظه وابسته است.اما در مدارهای ترتیبی خروجی هر لحظه وابسته به لحظه قبل است.

Half Adder :

به مدار ترکیبی که جمع حسابی دو بیت را انجام دهد نیم جمع کننده گفته می‌شود. برای ترسیم مدار داخلی نیم جمع کننده جدول تغییرات آن را رسم کرده و سپس ارقام **S** و **C** مربوط به جمع دو بیت را محاسبه می‌کنیم

ترسیم مدار:

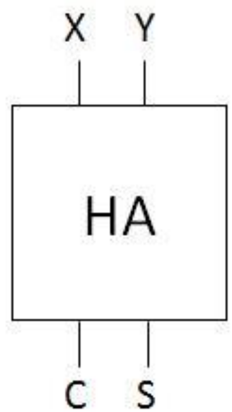


محاسبه ارقام **C** و **S**:

$$C = XY$$
$$S = XY' + X'Y$$

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

مدار HA را بطور خلاصه می توان به شکل مقابل نشان داد



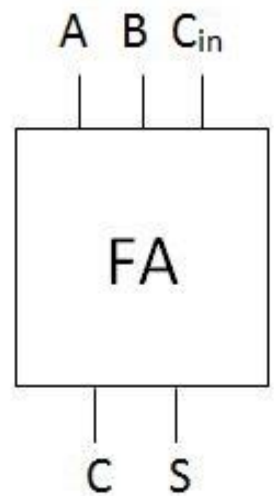
Full Adder :

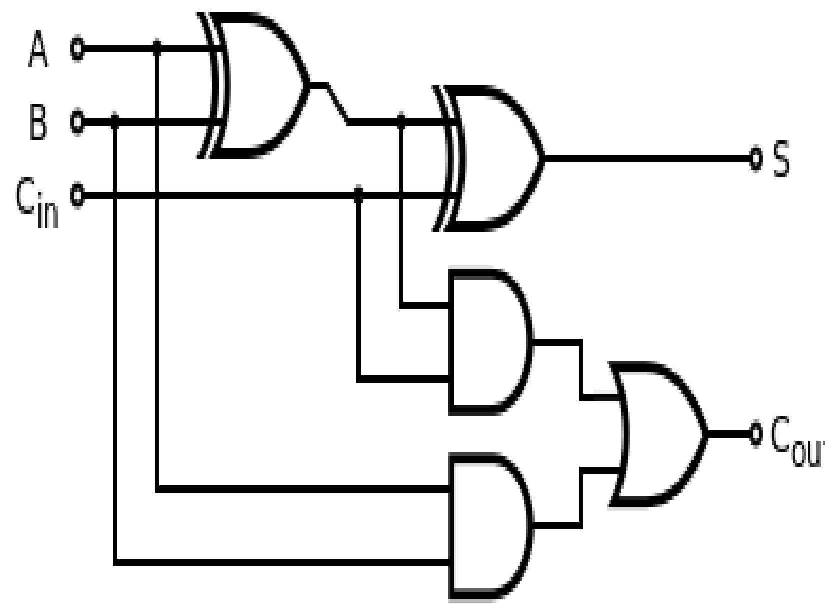
به مدار ترکیبی که جمع حسابی سه بیت را انجام دهد تمام جمع کننده گفته می‌شود.

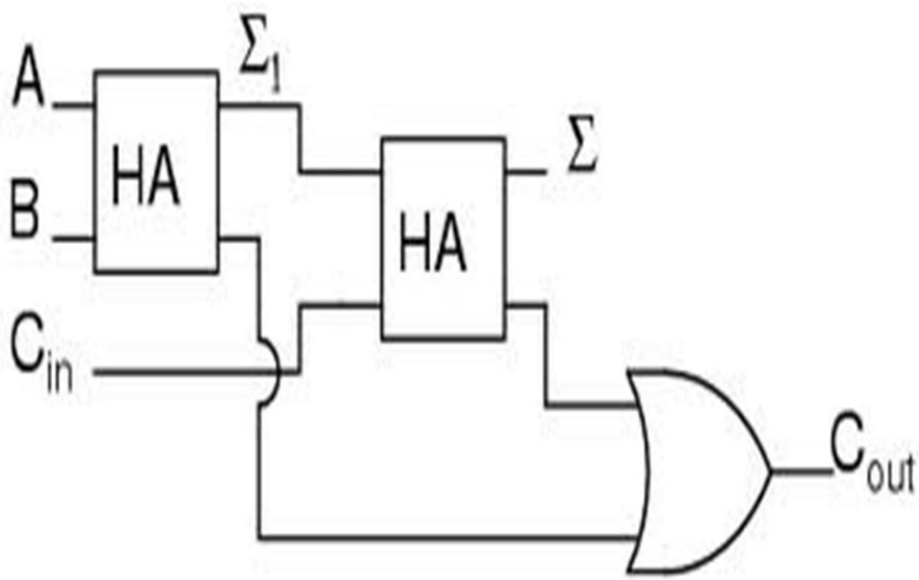
A	B	C_{in}	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

مدار داخلی Full Adder :

پس از محاسبه C و S مدار داخلی تمام جمع کننده به شکل زیر در می‌آید





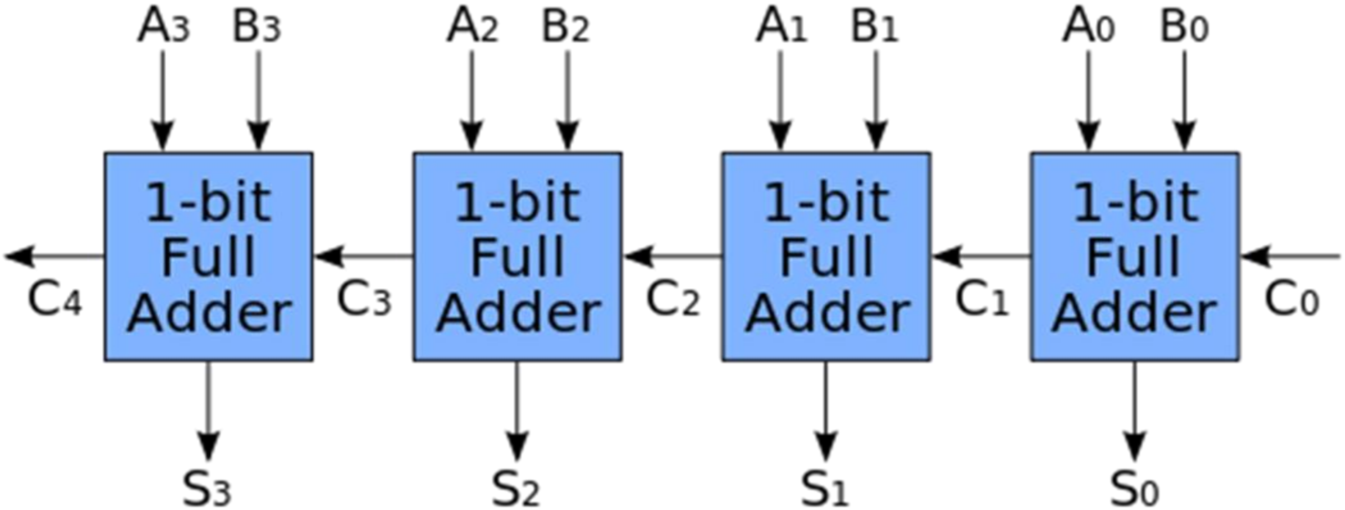


مدار FA را می‌توان با استفاده از دو HA به شکل مقابل ساخت

مدار جمع کننده چهاربیتی:

برای ساختن یک مدار جمع کننده چهار بیتی می‌توان از چهار FA استفاده کرد.

طریقه اتصال بین آنها در صفحه بعد آمده است.



به منظور تفریق دو عدد باید از عدد دوم متمم ۲ گرفته شود.

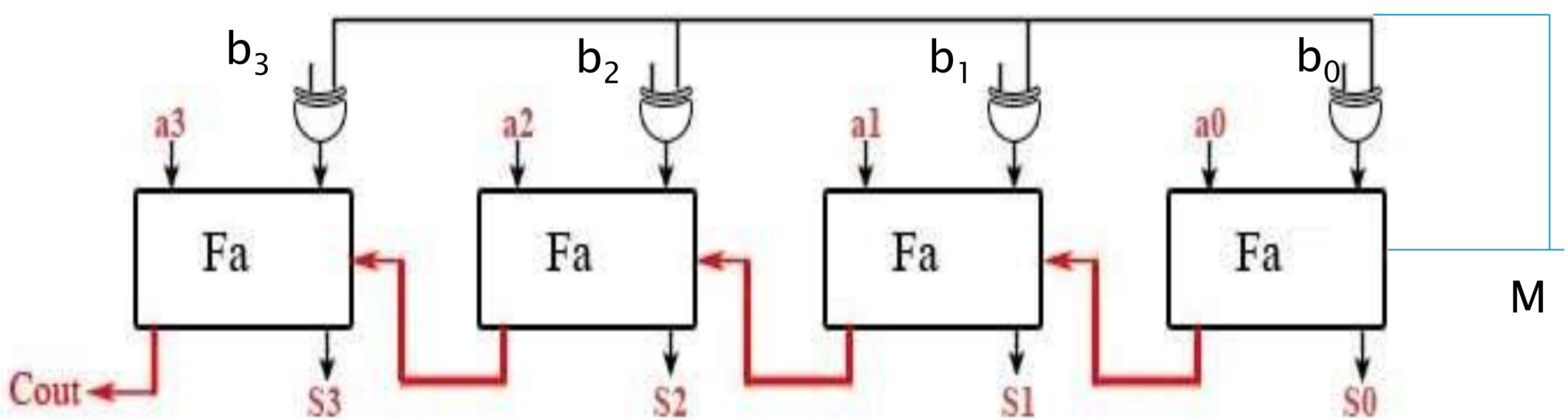
روش متمم ۲:

$$A - B = A + B' + 1$$

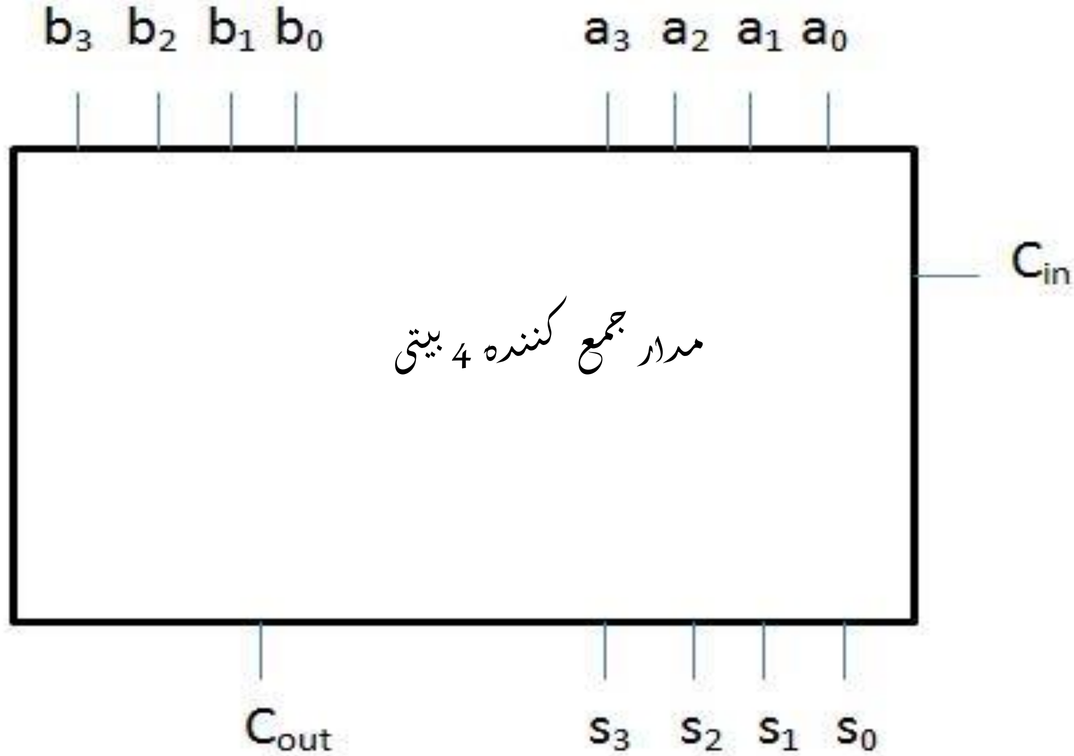
به طور مثال :

$$1101 - 1011 = 1101 + (0100+1)$$

در مدار زیر که یک جمع و تفریق کننده دودویی است زمانی که M (C_{in}) صفر باشد A+B میشود. اما زمانی که M ، یک است A−B صورت میگیرد.

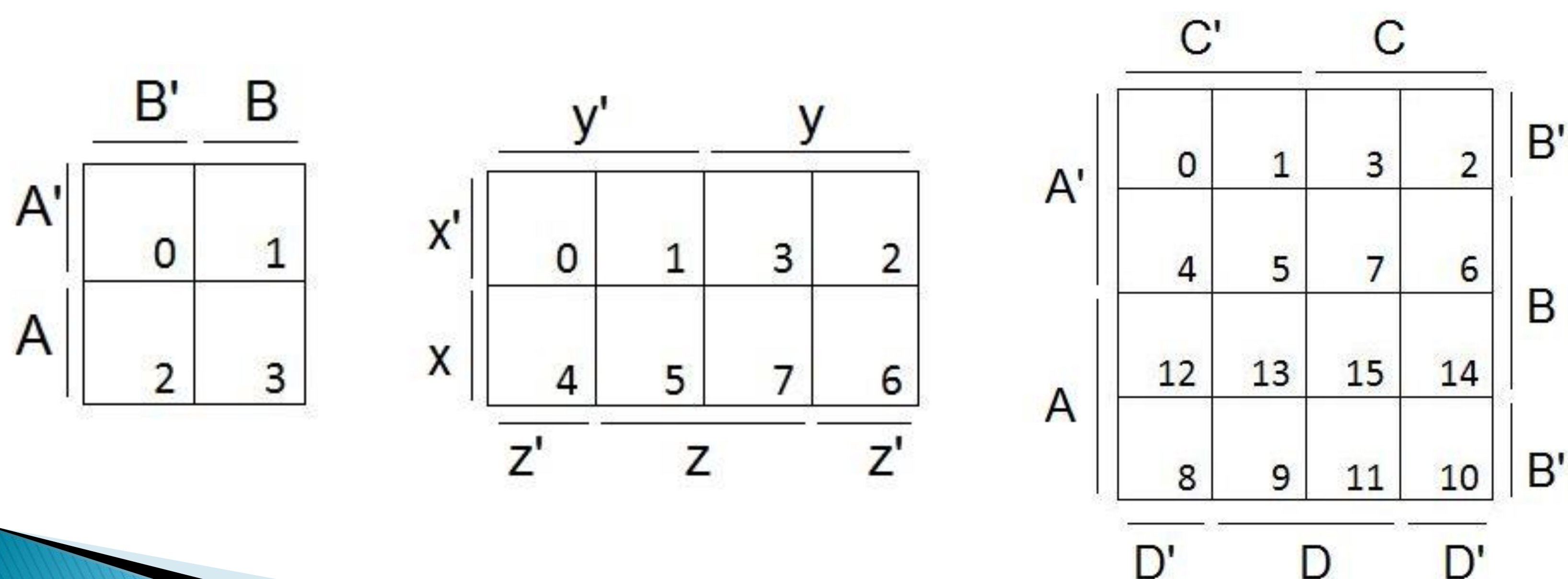


مدار صفحه قبل را می توان بصورت شکل زیر نمایش داد که در آن اگر C_{in} برابر صفر شود مدار جمع کننده و اگر برابر یک شود، مدار تفریقگر خواهد بود.



ساده سازی با جدول کارنو :

سریع‌ترین و راحت‌ترین راه برای ساده سازی توابع بولی دو، سه و چهار متغیره استفاده از جدول کارنو است. بدین صورت که مقدار خروجی (که بصورت ۰ و ۱ است) در جدول قرار می‌گیرد. نقشه جدول کارنو برای توابع دو ، سه و چهار متغیره به همراه محل قرار گرفتن خروجیها به شکل زیر می‌باشد:



* دو روش به منظور ساده سازی در جدول کارنو وجود دارد:

۱- **استفاده از جمع حاصلضرب‌ها (مینترم‌ها) :** در این روش آن خانه‌هایی از جدول که با ۱ پر شده اند به دستههای ۱ ، ۲ ، ۴ و بیشتر تقسیم می‌شوند (دسته‌های با ظرفیت بالاتر در اولویت هستند). سپس هر دسته به صورت ضربی از متغیرهایی که در ستونش قرار گرفته نوشته شده و در پایان عبارات مربوط به هر دسته با هم جمع می‌شوند.

۲-**استفاده از ضرب حاصلجمع‌ها (ماکسترم‌ها) :** این روش نیز همانند روش ۱ است، با این تفاوت که دسته‌بندی برای خانه‌هایی انجام می‌شود که با ۰ پر شده‌اند. در پایان چون عبارات بدست آمده متمم خروجی است ، یکبار آن را Not کرده تا خروجی مورد نظر بصورت ضرب حاصلجمع‌ها بدست آید.

در دو مثال زیر با استفاده از روشهای گفته شده تابع F را بدست آورید.

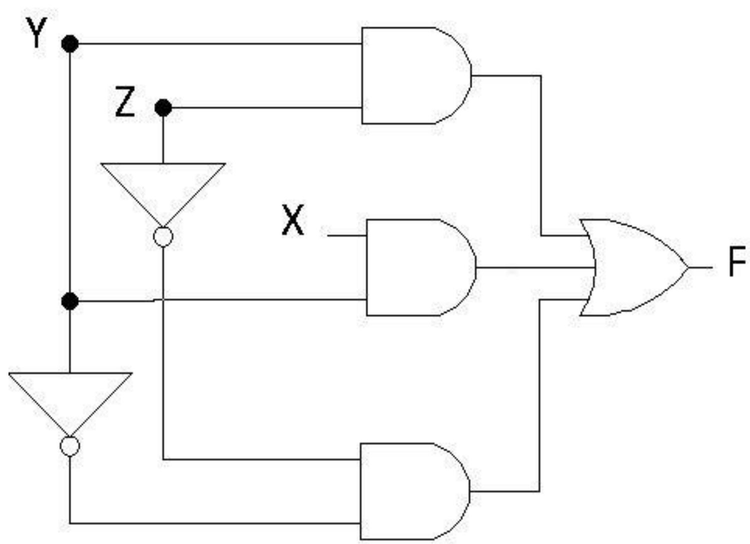
۱-مثال از مینترم)

$F(x,y,z) = \Sigma\ (0,3,4,6,7)$

الف- تشکیل جدول کارنو برای خروجی‌ها :

	Y'		Y	
X'	1	0	1	0
X	1	0	1	1
	Z'	Z		Z'

ب- ترسیم مدار



F X,Y,Z) =

$Y'Z' + YZ + XY$

۲-مثال از ماکسترم)

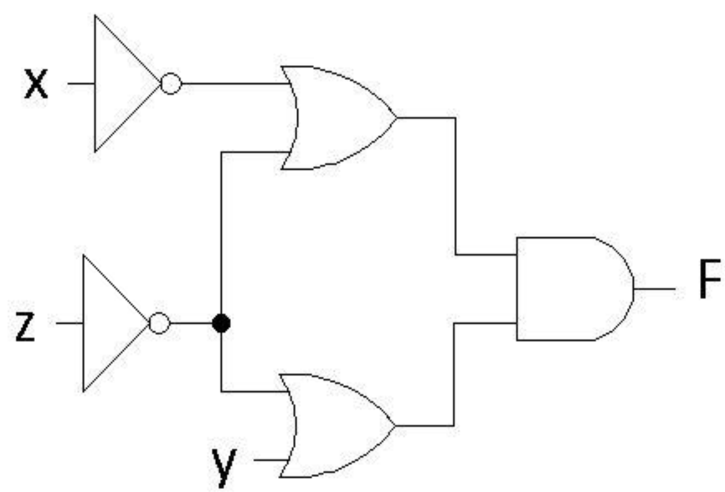
$F(x,y,z) = \pi(1,5,7)$

الف-تشکیل جدول کارنو برای خروجی‌ها :

	y'		y	
x'	1	0	1	1
x	1	0	0	1
	z'	z		z'

$F = (y+z')\,(x'+z')$

ب-ترسیم مدار :



مثال: تابع سه متغیره‌ای طراحی کنید به طوریکه اگر ورودی ۰،۱،۲،۳ باشد، خروجی یک واحد افزایش و اگر ورودی ۴،۵،۶،۷ باشد، خروجی یک واحد کاهش یابد.

حل :

۱- تشکیل جدول تغییرات :

x	y	z	M	N	W
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

۲- تشکیل جدول کارنو برای خروجی‌ها :

	y'		y	
x'	0	0	1	0
x	0	1	1	1
	z'	z		z'

	y'		y	
x'	0	1	0	1
x	1	0	1	0
	z'	z		z'

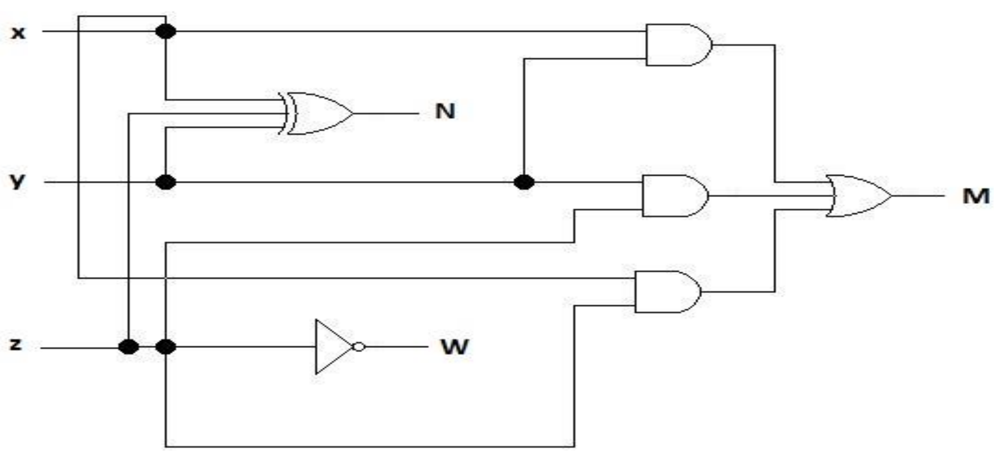
	y'		y	
x'	1	0	0	1
x	1	0	0	1
	z'	z		z'

$M = xy + yz + xz$

$N = x'y'z + x'yz' + xy'z' + xyz$

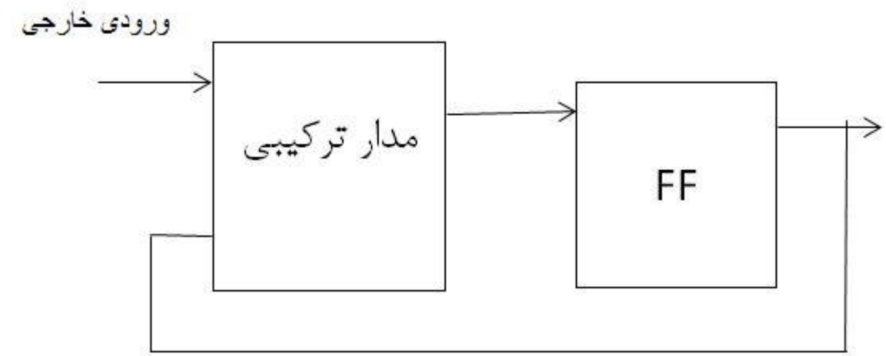
$W = z'$

۳- ترسیم دیاگرام مدار



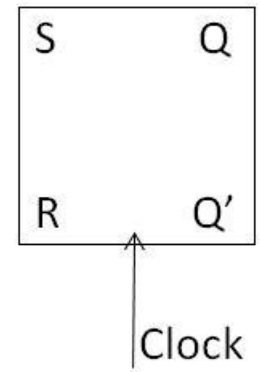
۲) مدارهای ترتیبی :

مدارهای ترتیبی مدارهایی هستند که دارای حافظه می‌باشند . کوچکترین جزء هر حافظه که قادر است ۱ بیت اطلاعات را در خود ذخیره کند ، فلیپ فلاپ می‌گویند . در مدارهای ترتیبی نیاز است از یک پالس ساعت استفاده گردد تا تمام قطعات با نظم کار کنند . تمامی مدارهای ترتیبی از بلاک دیاگرام زیر تبعیت می‌کنند:



انواع مختلف فلیپ فلاپ‌ها عبارتند از : فلیپ فلاپ SR ، D ، JK و T

۱- فلیپ فلاپ SR



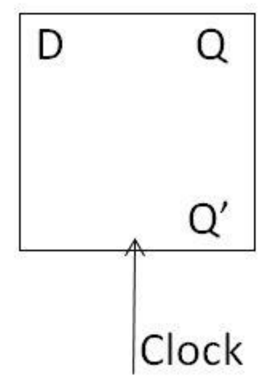
Q_t	Q_{t+1}	S	R
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

جدول تحریک فلیپ فلاپ SR

S	R	Q_t
0	0	Q_t
0	1	0
1	0	1
1	1	تعریف نشده

جدول حالت فلیپ فلاپ SR

۲- فلیپ فلاپ D: (از به هم وصل شدن پایه‌های ورودی فلیپ فلاپ SR بدست می‌آید.)



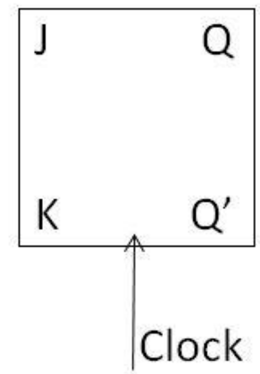
Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

جدول تحریک فلیپ فلاپ D

D	Q
0	0
1	1

جدول حالت فلیپ فلاپ D

۳- فلیپ فلاپ JK



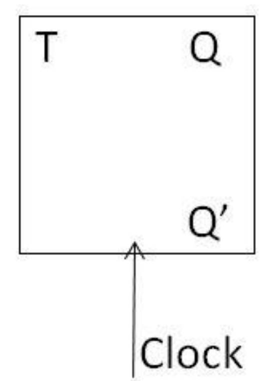
Q_t	Q_{t+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

جدول تحریک فلیپ فلاپ JK

J	K	Q_t
0	0	Q_t
0	1	0
1	0	1
1	1	Q_t'

جدول حالت فلیپ فلاپ JK

۴- فلیپ فلاپ T: (از به هم وصل شدن پایه‌های ورودی فلیپ فلاپ JK بدست می‌آید.)



Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

جدول تحریک فلیپ فلاپ T

T	Q
0	Q
1	Q’

جدول حالت فلیپ فلاپ T

❋ به طور کلی برای طراحی مدارهای ترتیبی باید مراحل زیر را طی نمود:

۱- رسم دیاگرام حالت به صورت اشکال خط و دایره که هر حالت را با یک دایره و مسیر حرکت را از یک حالت به حالت دیگر نمایش می‌دهد.

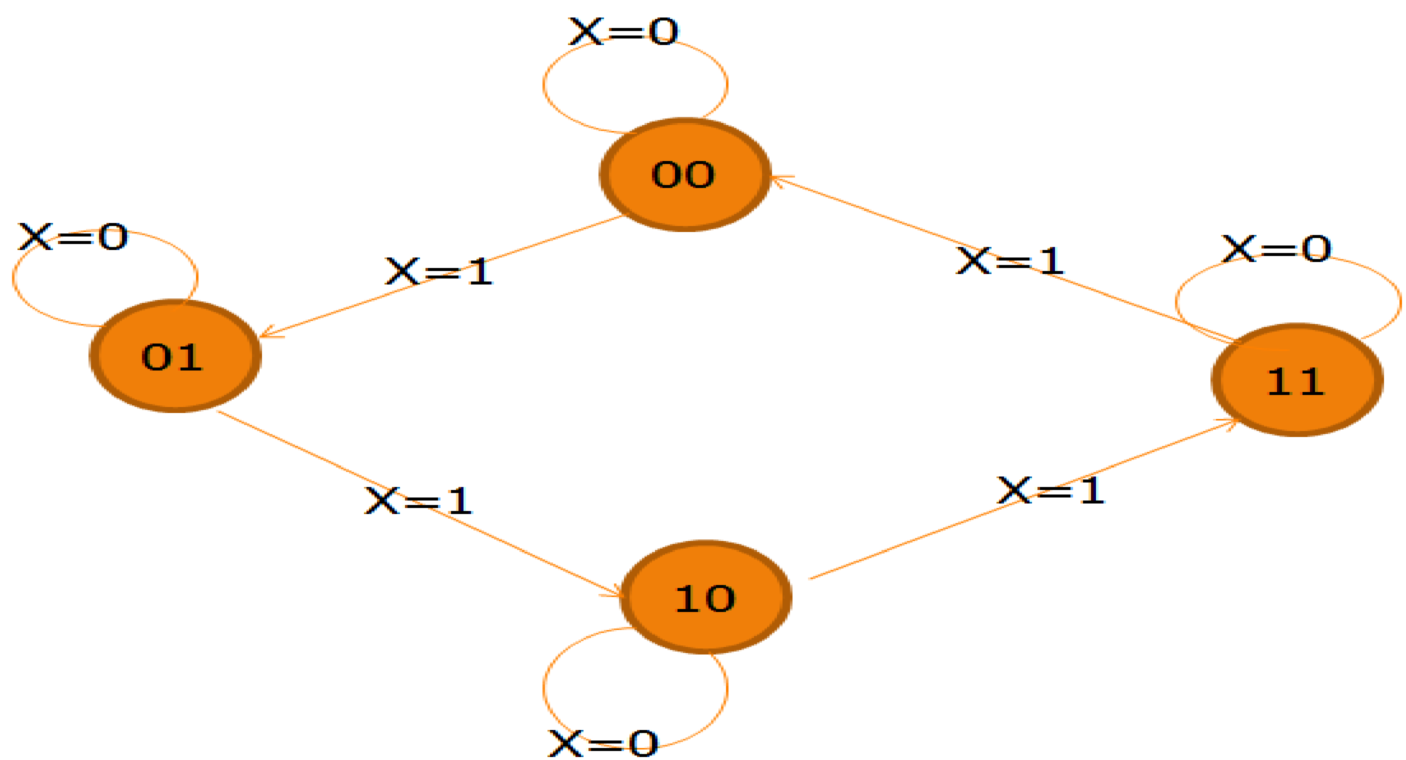
۲- رسم جدول تحریک .

۳- رسم جداول کارنو و ساده سازی .

۴- رسم مدار .

مثال طراحی :

مداری طراحی نمایید که در صورتی که ورودی خارجی X مساوی صفر است شمارشی صورت نگیرد و در زمانی که X مساوی یک است رشته ی ۰۰ ۰۱ ۱۰ ۱۱ ۰۰ و تکرار آن را ادامه دهد.


X=0
00
X=0
X=1
X=1
X=0
01
11
X=1
X=1
10
X=0

حالت اولیه			حالت ثانویه	
A	B	x	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

jA	KA	jB	KB
0	x	0	x
0	x	1	x
0	x	x	0
1	x	x	1
x	0	0	x
x	0	1	x
x	0	x	0
x	1	x	1

B

	0	0	1	0
A	X	X	X	x

x

$j_A = Bx$

B

	X	X	X	X
A	0	0	1	0

x

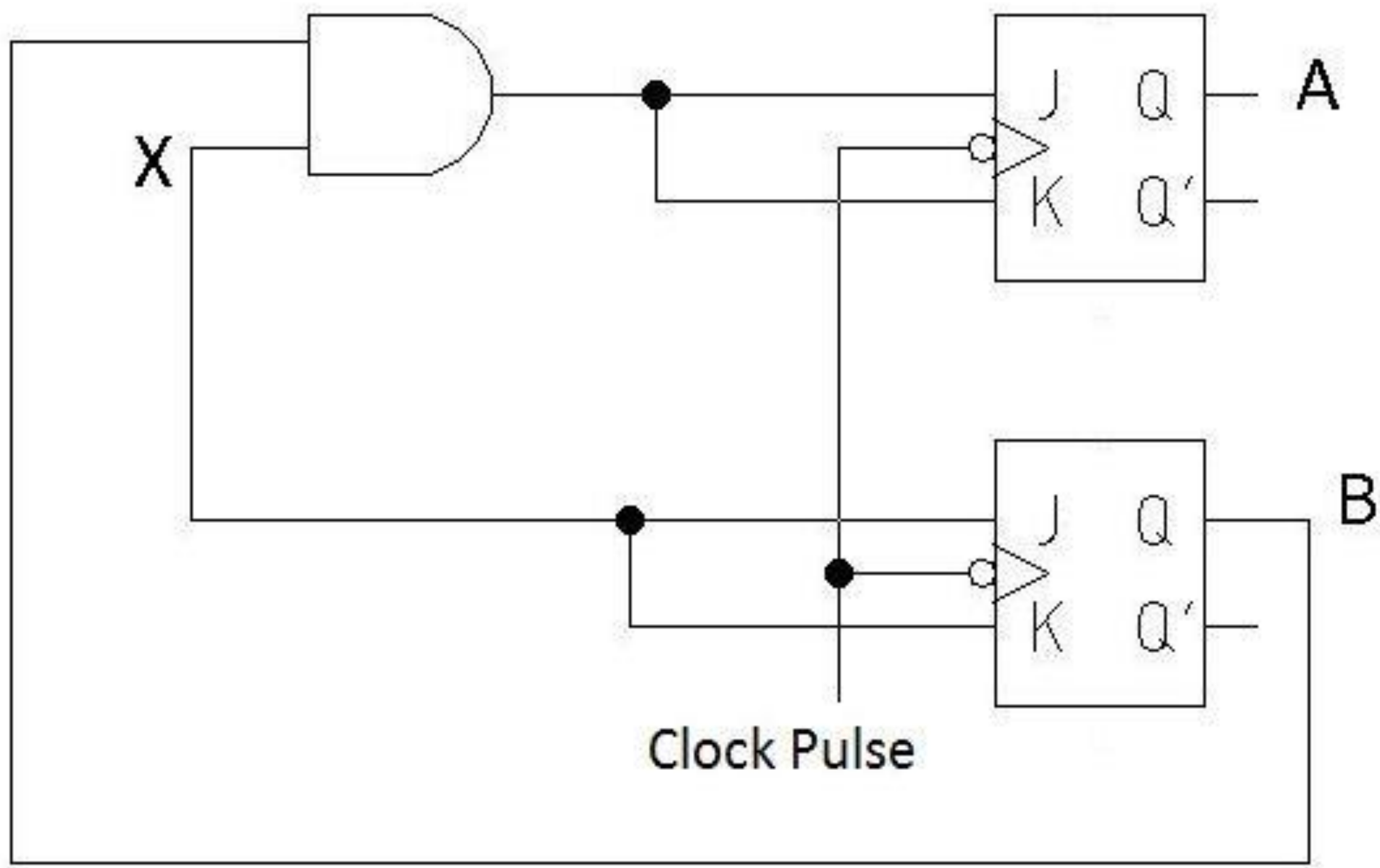
$K_A = Bx$

			B	B
	0	1	X	X
A	0	1	X	x
		x	x	

$j_{B=x}$

			B	B
	X	X	1	0
A	X	X	1	0
		x	x	

$K_{B=x}$


X
J
Q
A
K
Q'
J
Q
B
K
Q'
Clock Pulse

تمرین :

یک شمارنده سه بیتی طراحی کنید. زمانی که ورودی X یک است رشته نزولی ۰۰۰ ۰۰۱ ۰۱۰ ۰۱۱ ۱۰۰ ۱۰۱ ۱۱۰ ۱۱۱ ۰۰۰ وتکراری از آن را بشمارد و زمانیکه X صفر است در جای خود باقی بماند .

تمرین :

شمارنده ۲ بیتی طراحی کنید که دارای ۲ ورودی E , X باشد زمانیکه X یک است و E صفر رشته ۰۰ ۰۱ ۱۰ ۱۱ ۰۰ و تکراری از آن و زمانی که X یک و Eیک می باشد رشته ی ۱۱ ۱۰ ۰۱ ۰۰ ۱۱ و تکرار آن و زمانی که Xصفر است در جای خود باقی بماند .

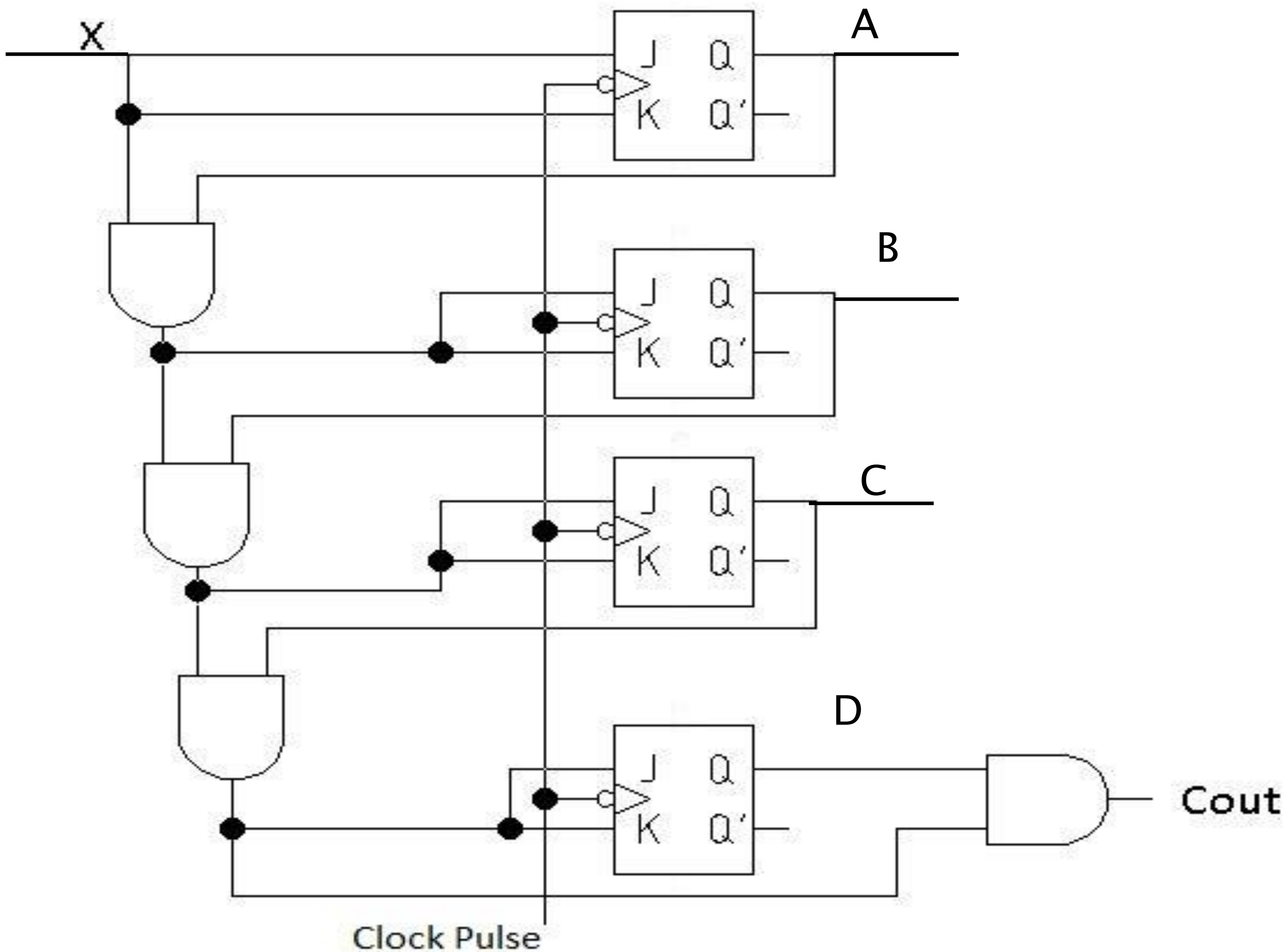
مثال :

یک شمارنده ۴ بیتی طراحی نمایید.

بطور کلی میتوان برای طراحی شمارنده از دو قانون زیر تبیعیت نمود:

بیت پایین رتبه ترین در هر مرحله متمم می شود و بیت بالا رتبه به شرطی متمم میشود که تمام بیت های پایین رتبه آن یک باشد.

0000
0001
0010
0100


X
A
B
C
D
J
Q
K
Q'
Cout
Clock Pulse