طراحی کامپیوتری سیستم های دیجیتال

فرهادی

پاییز ۹۶

- منبع اصلی درس:

طراحی خودکار مدارهای دیجیتال با FPGA

و

زبان توصیف سخت افزار VHDL

- مولف: دکتر حسن سیدرضی
- انتشارات: ناقوس

– زمان بندی

- طول ترم: ۱۶هفته
- تعطیلات: -- جلسه
- تعداد جلسات: ۲۴ جلسه

– نحوه ارتباط

- سایت دانشگاه www.Shahroodut.ac.ir
- email: mfarhadi@shahroodut.ac.ir

– نحوه ارزیابی

تمرین:	٪۲۰ (دوره‌ای)
کوئیز:	+٪۵
میان ترم ۱:	٪۲۰ (۱۵ اسفند)
میان ترم۲:	٪۲۰ (۱۳ اردیبهشت)
پایان ترم:	٪۴۰ (تقویم آموزشی)
پروژه:	+٪۱۵

شرط محاسبه نمرات امتیازی کسب حداقل ۴۰٪ از مجموع میان ترم و پایان ترم است.

فهرست مطالب

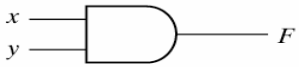
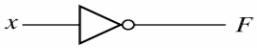
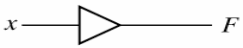
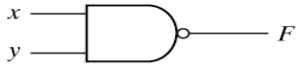
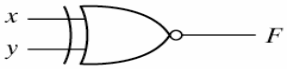
- یادآوری مدار منطقی
- مقدمه
  - انواع مدارهای منطقی برنامه پذیر PLA و PAL
  - سوئیچ های قابل برنامه ریزی CPLD و FPGA
  - ساختار FPGA
  - روش طراحی سیستم های دیجیتال با FPGA
- زبان توصیف سخت افزار VHDL
  - بلوک پایه یک طرح دیجیتال
  - سیگنال
  - نکات عمومی
  - مقدار اولیه دادن به سیگنال
  - نوع های سیگنال
  - عملگرها
  - روش های مدلسازی مدارهای دیجیتال
  - مدل کردن تأخیر
  - حلقه
  - پروسس
  - طراحی سلسله مراتبی
  - برنامه تست مدارهای دیجیتال
- تعریف نوع آرایه و رکورد برای سیگنال
- برنامه های فرعی، توابع و پکیج ها

یادآوری مدارمنطقی

- گیتهای منطقی
- ساده سازی سطح گیت
- مدارهای ترکیبی
- مدارهای ترتیبی

سمبولهای استاندارد گیتی

- جدول درستی
- سمبل استاندارد گیت
- عملیات منطقی
- Single input یا Multiple

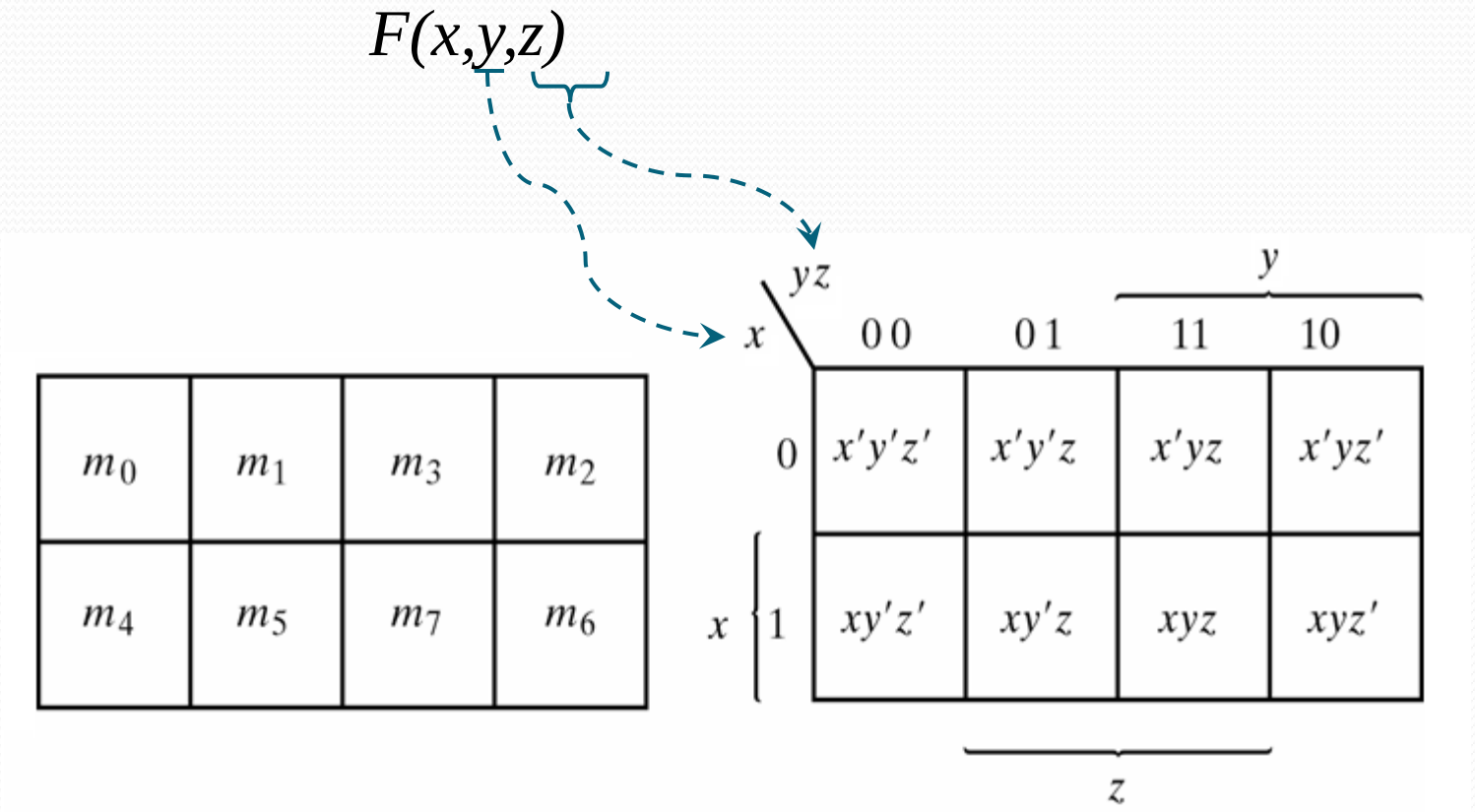
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

مشخصات جدول کارنو

- هر تابع بولی را می‌توان بصورت مجموعی از مینترمها نشان داد. جدول کارنو از مربع هایی تشکیل شده است که **هر مربع، نشان دهنده یک مینترم است.**
- سطرها و ستونهای این جدول به روش **کد گری**، کدگذاری می گردند.
- هر دومربع همسایه فقط در یک متغیر با هم اختلاف دارند. لذا می توان با توجه به خواص جبر بول، مجموع آنها را ساده کرد.(متغیر مشترک را حذف نمود)
- خانه‌هایی از جدول که **مقدارتابع در آنها برابر با یک** می‌باشد را مشخص می‌کنیم.
- بطور کلی برای توابع با n **متغیر**، جدول کارنو دارای 2^n خانه است.

جدول کارنو سه متغیره

- برای سه متغیر، **هشت مینترم** وجود دارد و بنابراین جدول کارنو باید هشت خانه داشته باشد.

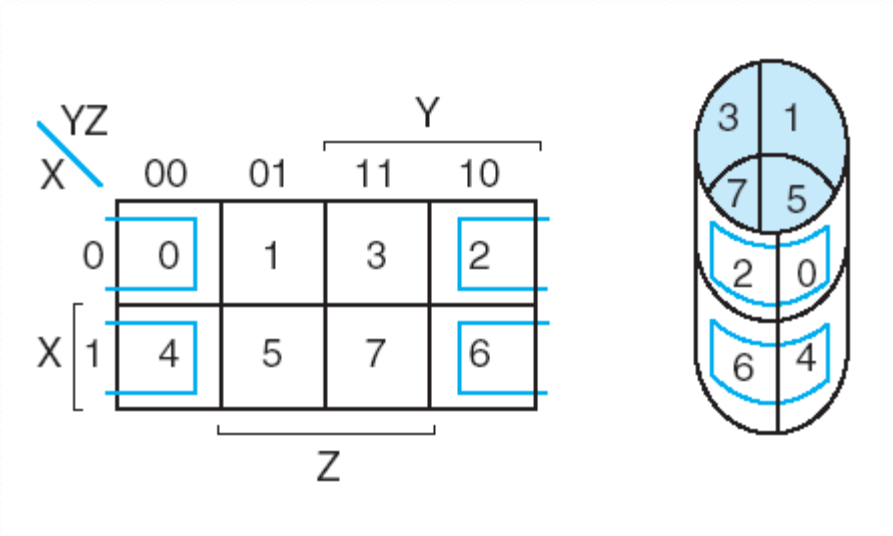


Three-variable map

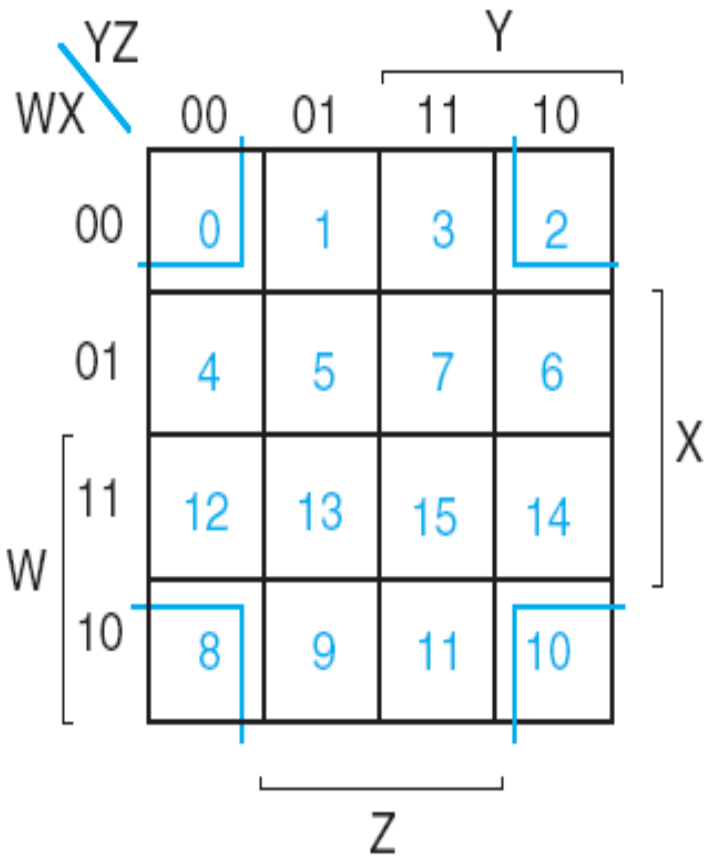
همسایگی(Adjacency) در جدول کارنا –

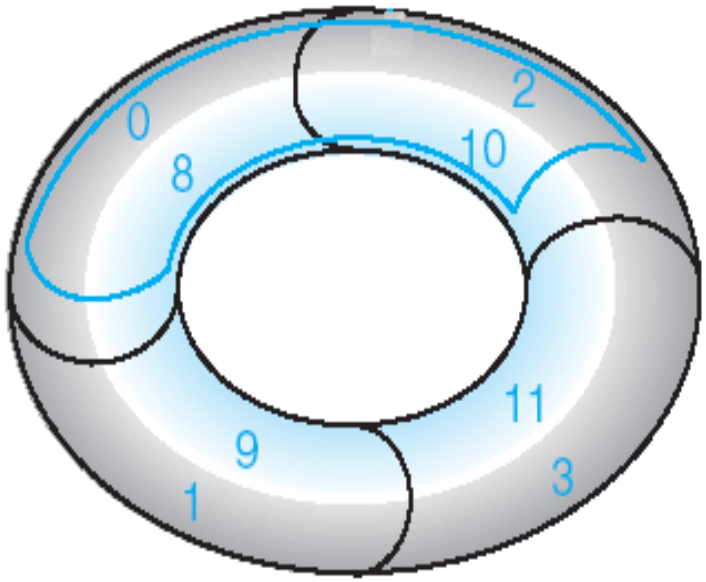
- به مربع(خانه هایی) که در کنار هم قرار گرفته اند، همسایه می گویند. هر خانهٔ جدول با خانهٔ مجاور خود تنها در یک لیترال تفاوت دارد.

- علاوه بر مربع های همجوار ظاهری، خانه‌های لبه بالا و پایین و نیز لبه چپ و راست هم مجاور یکدیگر هستند، اگرچه در کنار یکدیگر قرار ندارند.



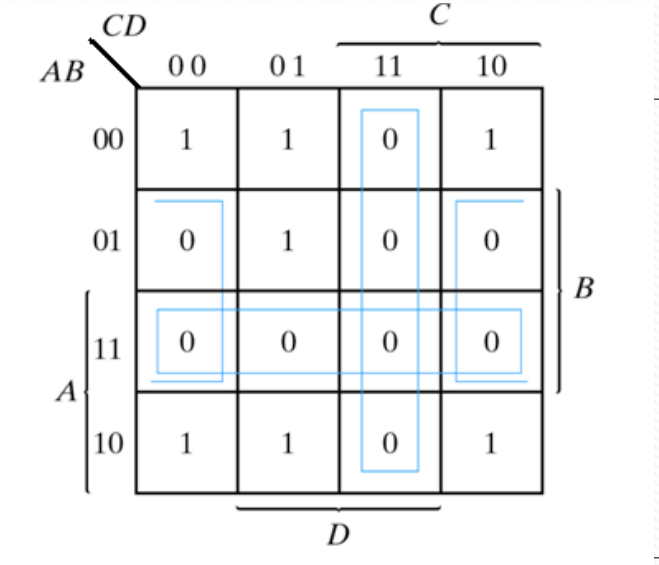
همسایگی(Adjacency) در جدول کارنا –





SOP و POS –

$$F(A,B,C,D) = \Sigma(0,1,2,5,8,9,10)$$

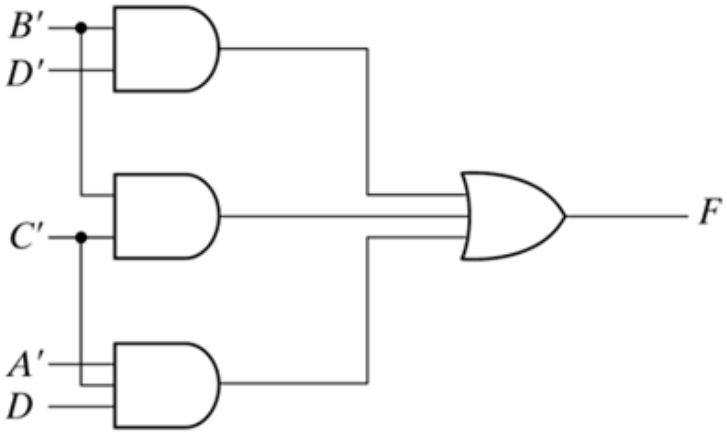


- یک ها را ترکیب می کنیم:

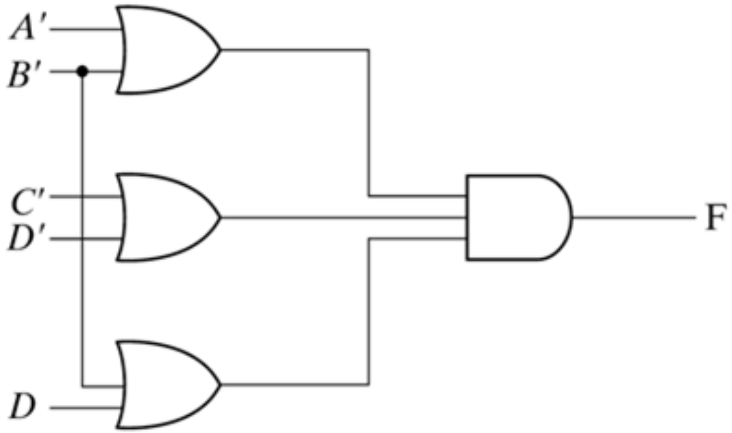
$$F = B'D' + B'C' + A'C'D$$

- صفرها را ترکیب می کنیم:

$$F' = AB + CD + BD'$$

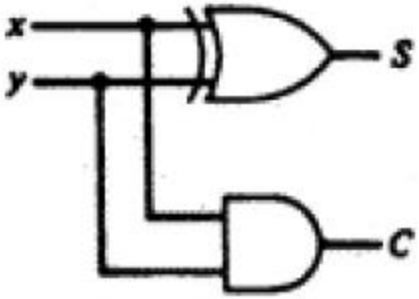


$F = B'D' + B'C' + A'C'D$



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

مدار نیم جمع کننده



(ب) دیاگرام منطقی

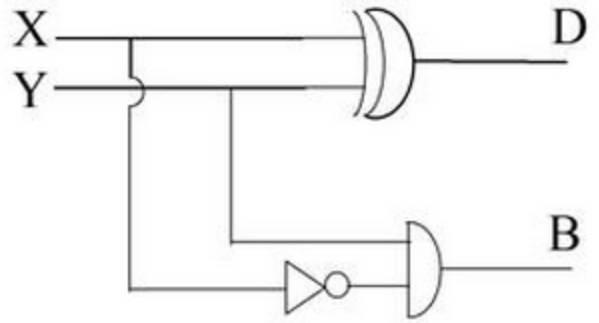
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(الف) جدول درستی

مدار تفریق‌کننده ناقص

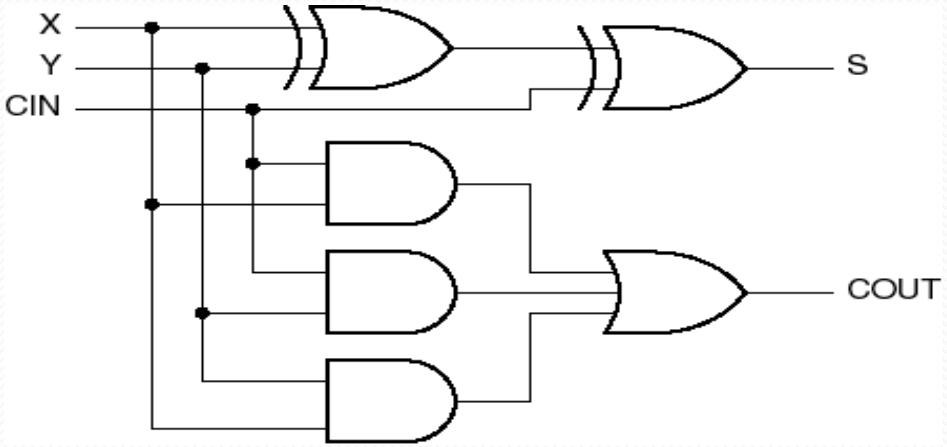
- B : مقداری که از مرتبه بالا قرض گرفته می‌شود.
- D : حاصل تفریق

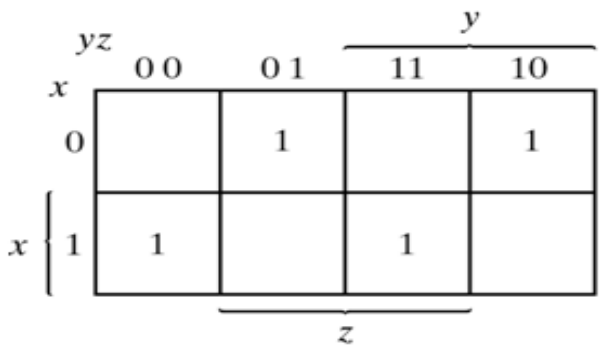
X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



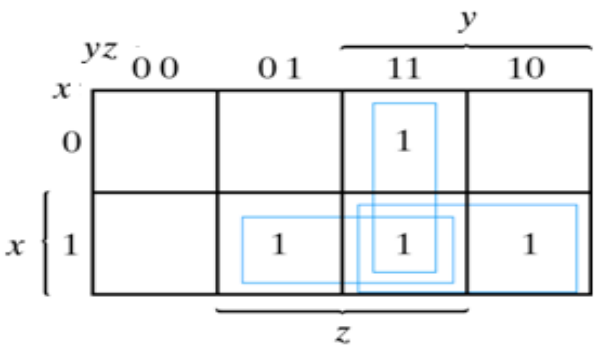
جمع کننده کامل

X	Y	Z	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





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

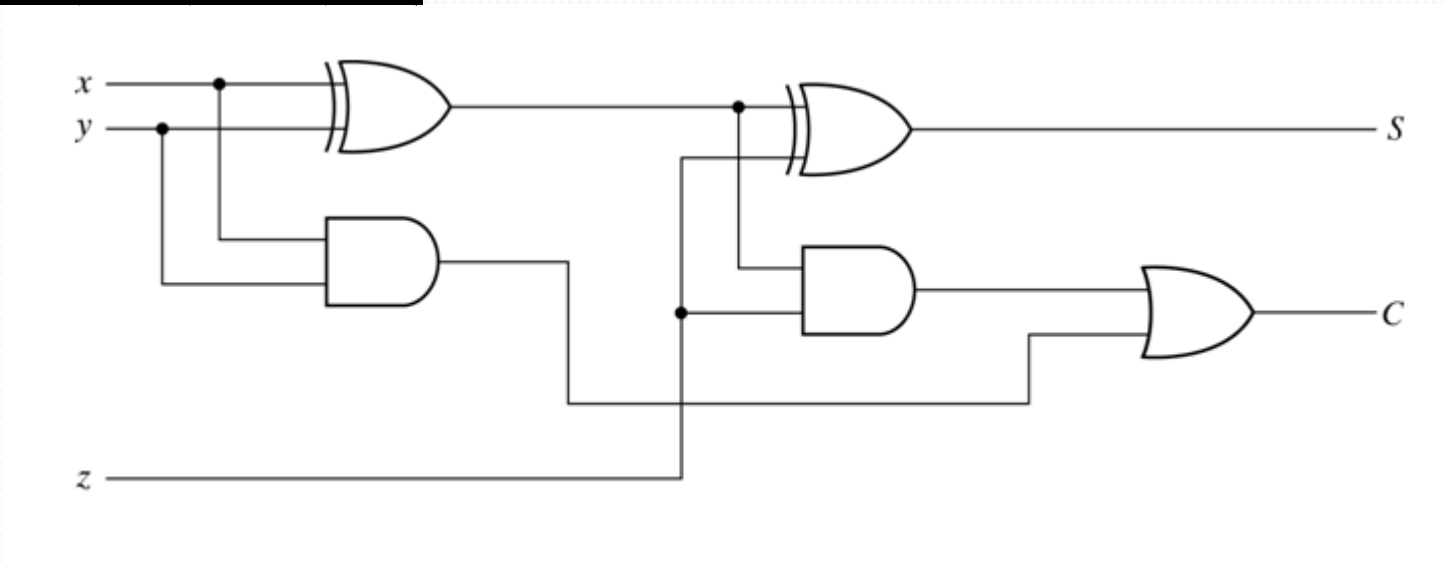


$$S = xy + xz + yz$$
$$= xy + xy'z + x'yz$$

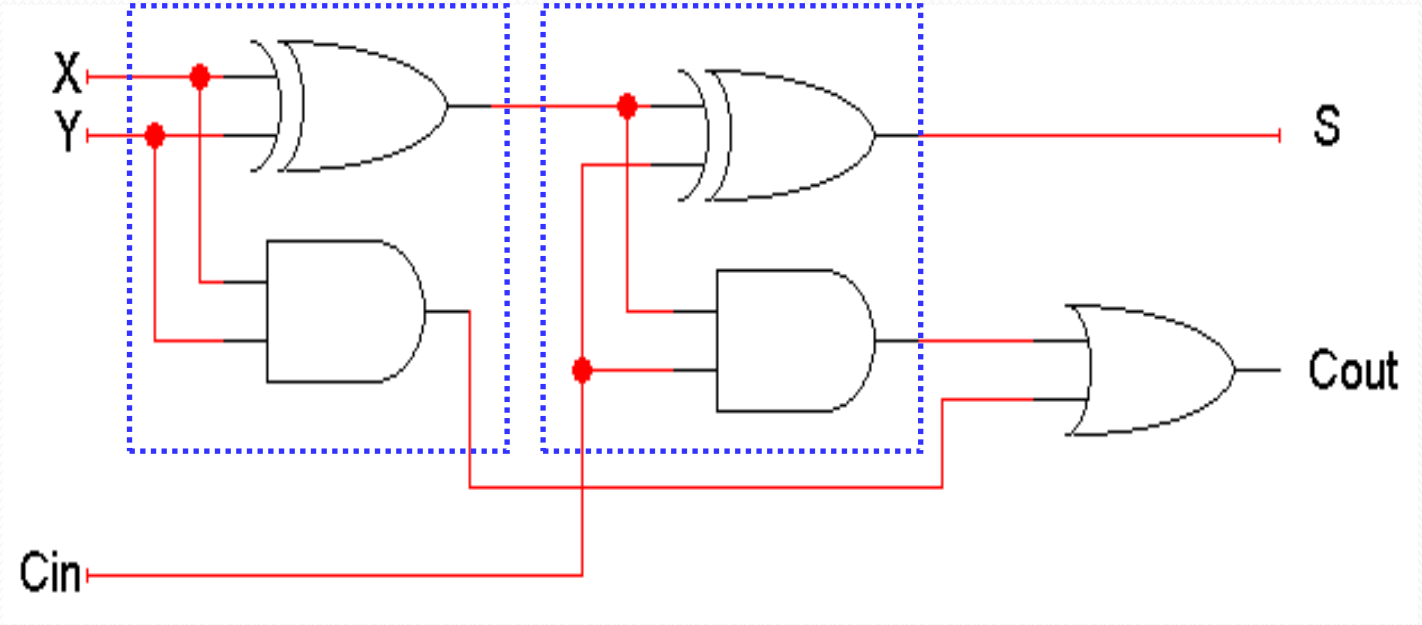
جمع کننده کامل

X	Y	Z	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

$$C_{out} = \Sigma m(3,5,6,7)$$
$$= X' Y C_{in} + X Y' C_{in} + X Y C_{in}' + X Y C_{in}$$
$$= (X' Y + X Y') C_{in} + XY(C_{in}' + C_{in})$$
$$= (X \oplus Y) C_{in} + XY$$

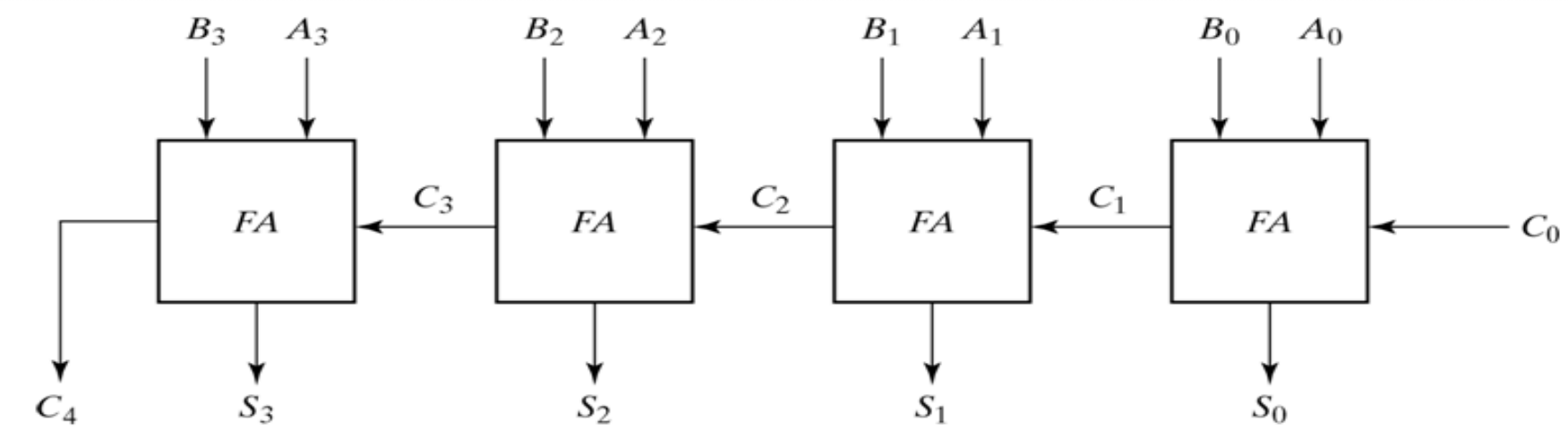


طراحی مدار تمام‌جمع کننده با نیم جمع کننده



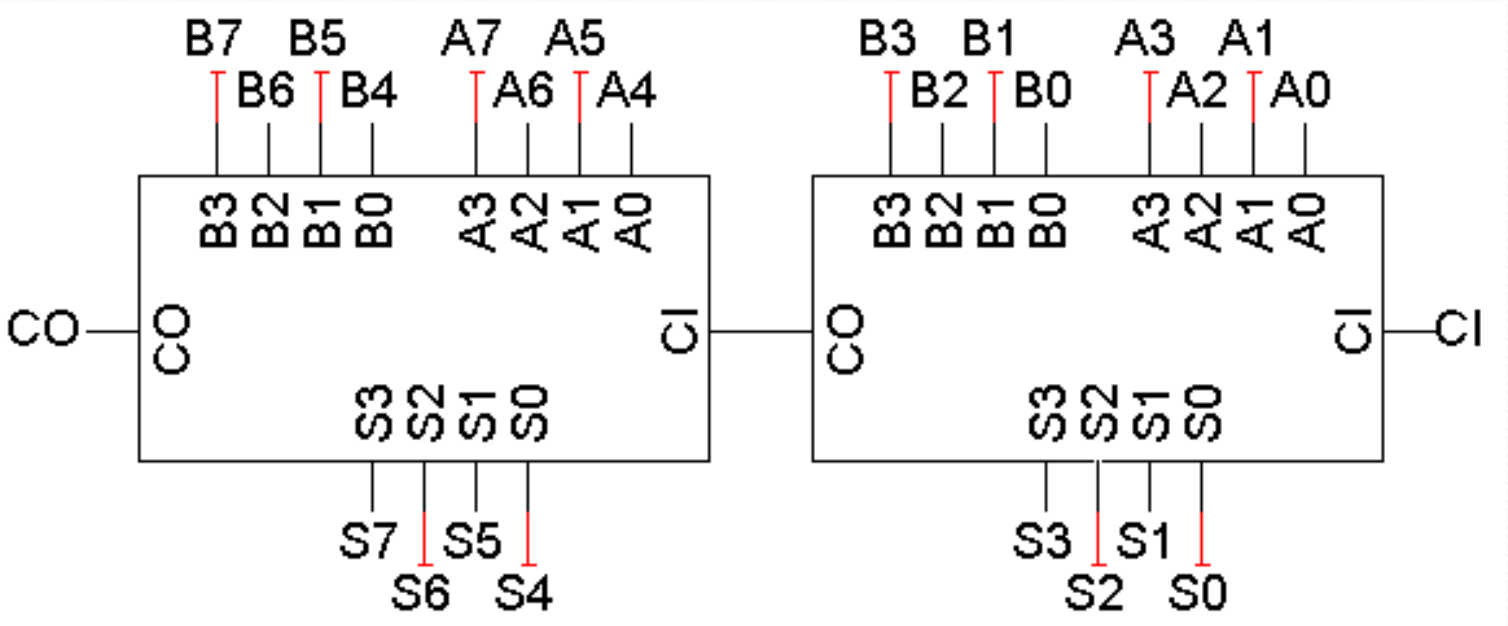
جمع کننده ٤ بیتی

- با قرار دادن ۴، full adder به دنبال هم همانند شکل زیر می‌توان یک full adder چهار بیتی طراحی کرد.

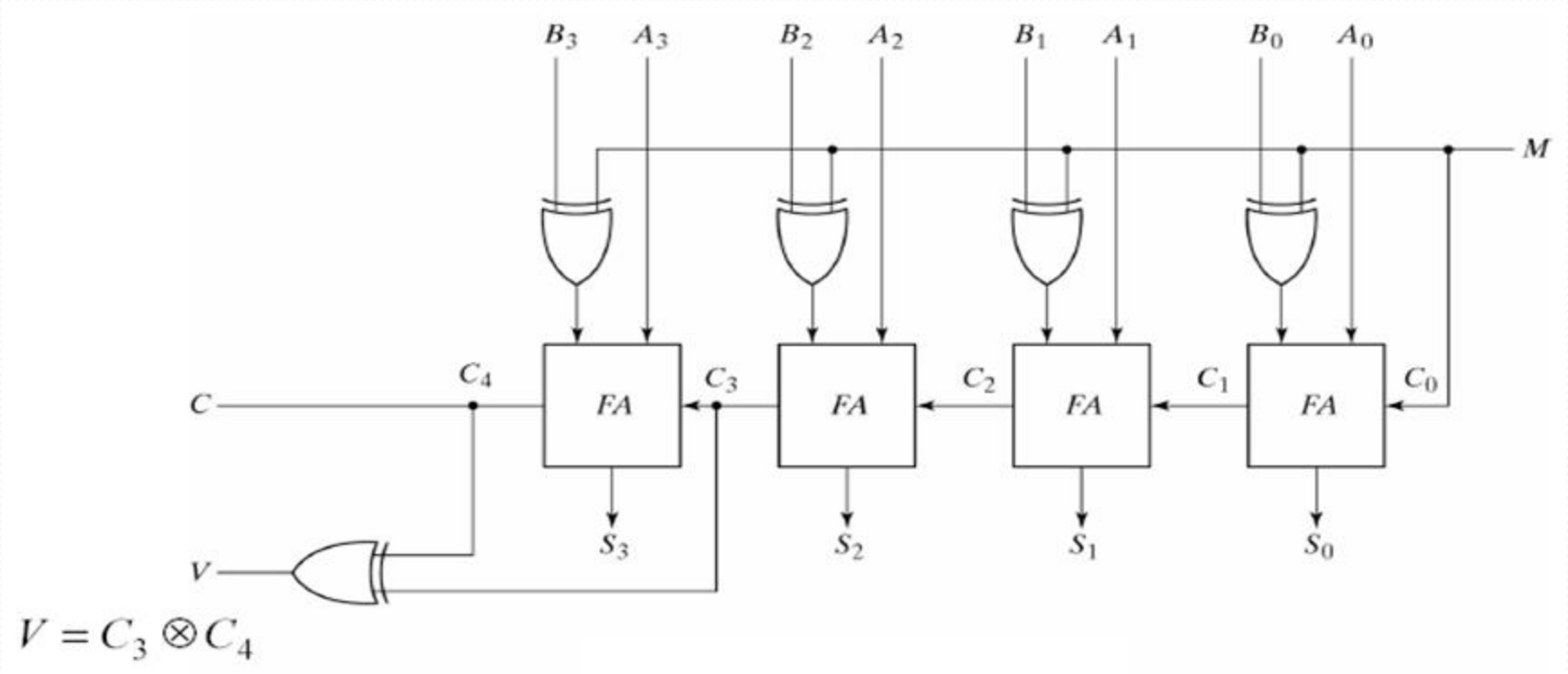


– ساخت یک جمع کننده ۸بیتی توسط جمع کننده های ۴بیتی

- روش ساخت جمع کننده های ۸ بیتی با استفاده از جمع کننده های ۴ بیتی به صورت زیر است:
  - رقم نقلی ورودی $C_{in}=o$ قرار داده می شود.
  - C_{out} مربوط به جمع کننده ۴ بیت کم ارزشتر وارد C_{in} جمع کنندهٔ ۴-بیت با ارزش بالاتر می شود.

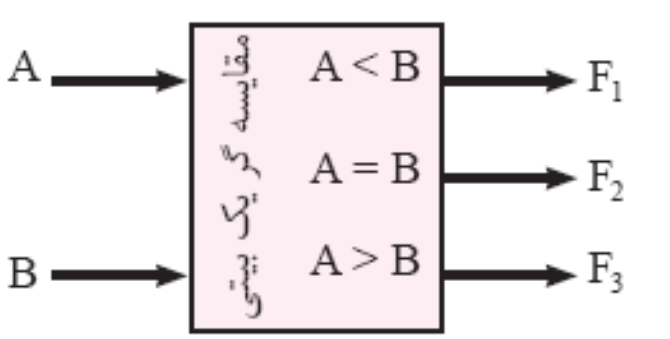


جمع کننده/تفریق کننده ٤ بیتی



مقایسه کننده تک‌بیتی

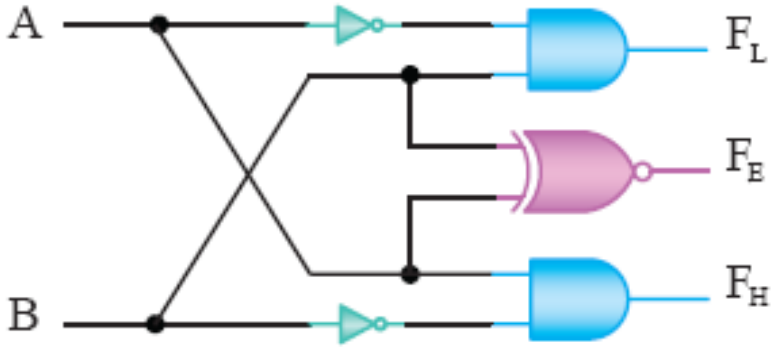
A	B	F_1 $A < B$	F_2 $A = B$	F_3 $A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



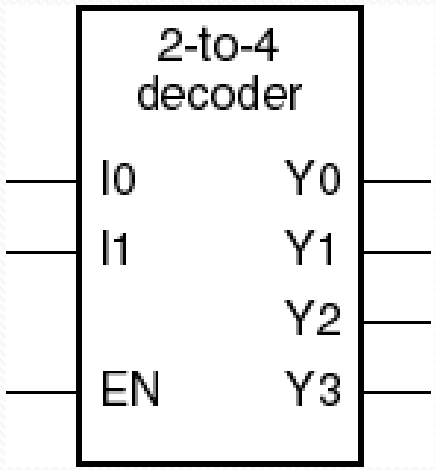
$F_L = \bar{A}B$

$F_E = \bar{A}\bar{B} + AB$

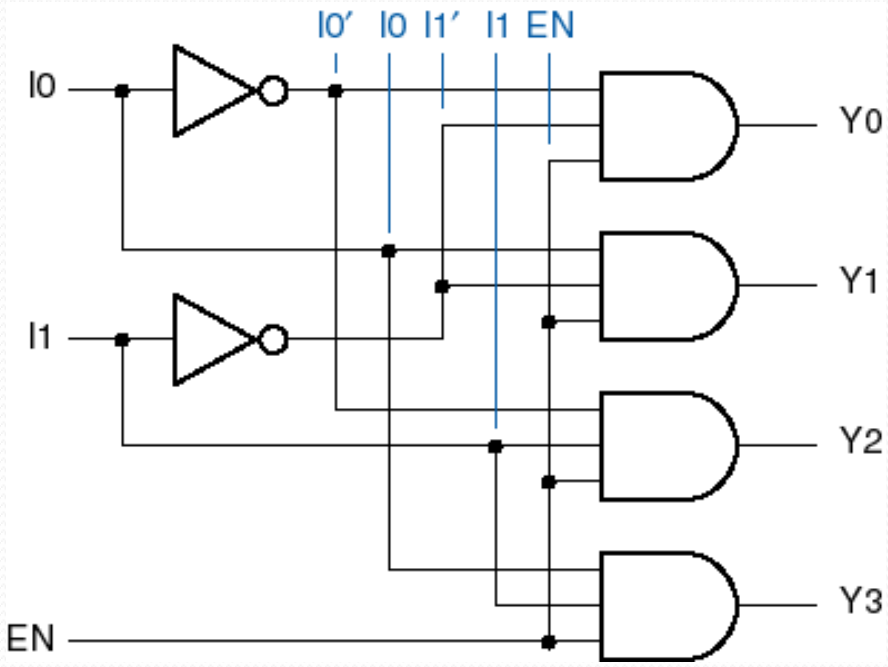
$F_H = A\bar{B}$



Binary 2-to-4 decoder

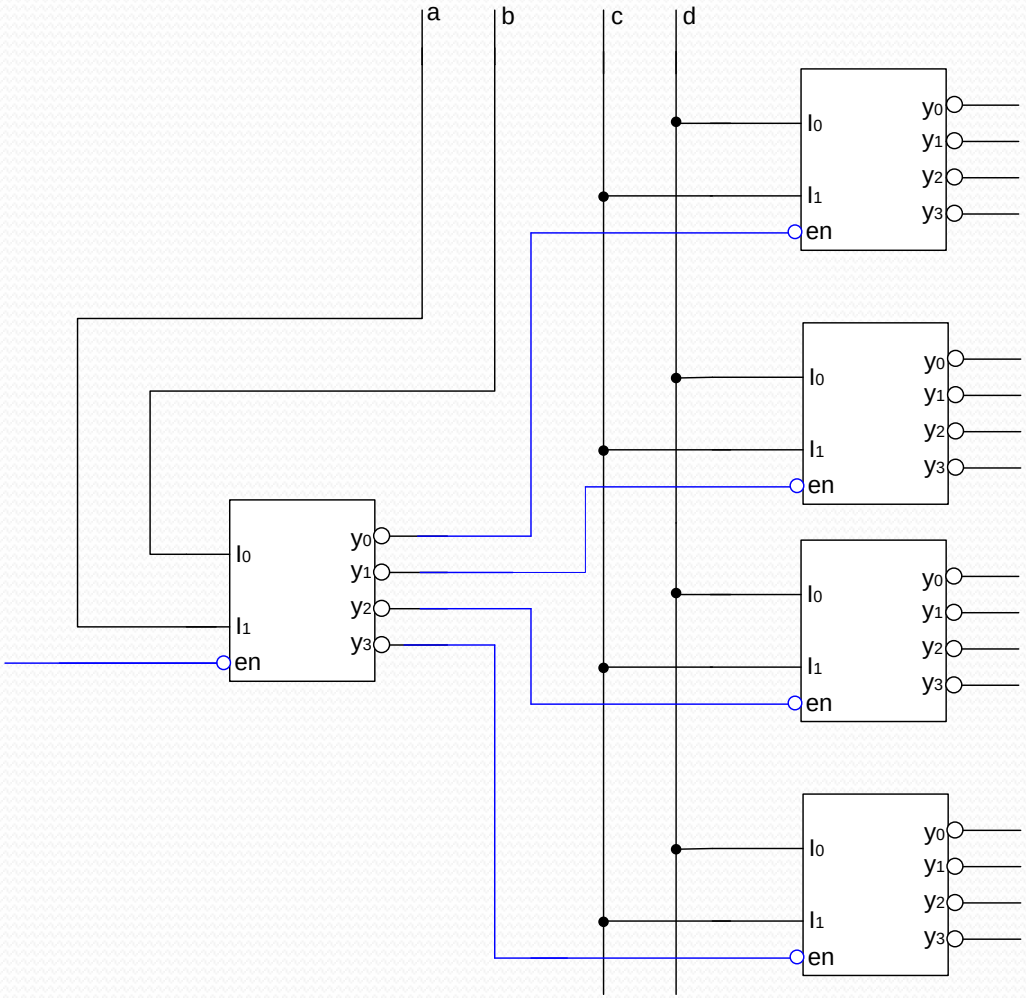


Inputs			Outputs			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



- Decoder Cascading

- با استفاده از اتصال درختی تعدادی رمزگشای **2^n به n** می توان بیشتر از n خط را رمزگشایی نمود.

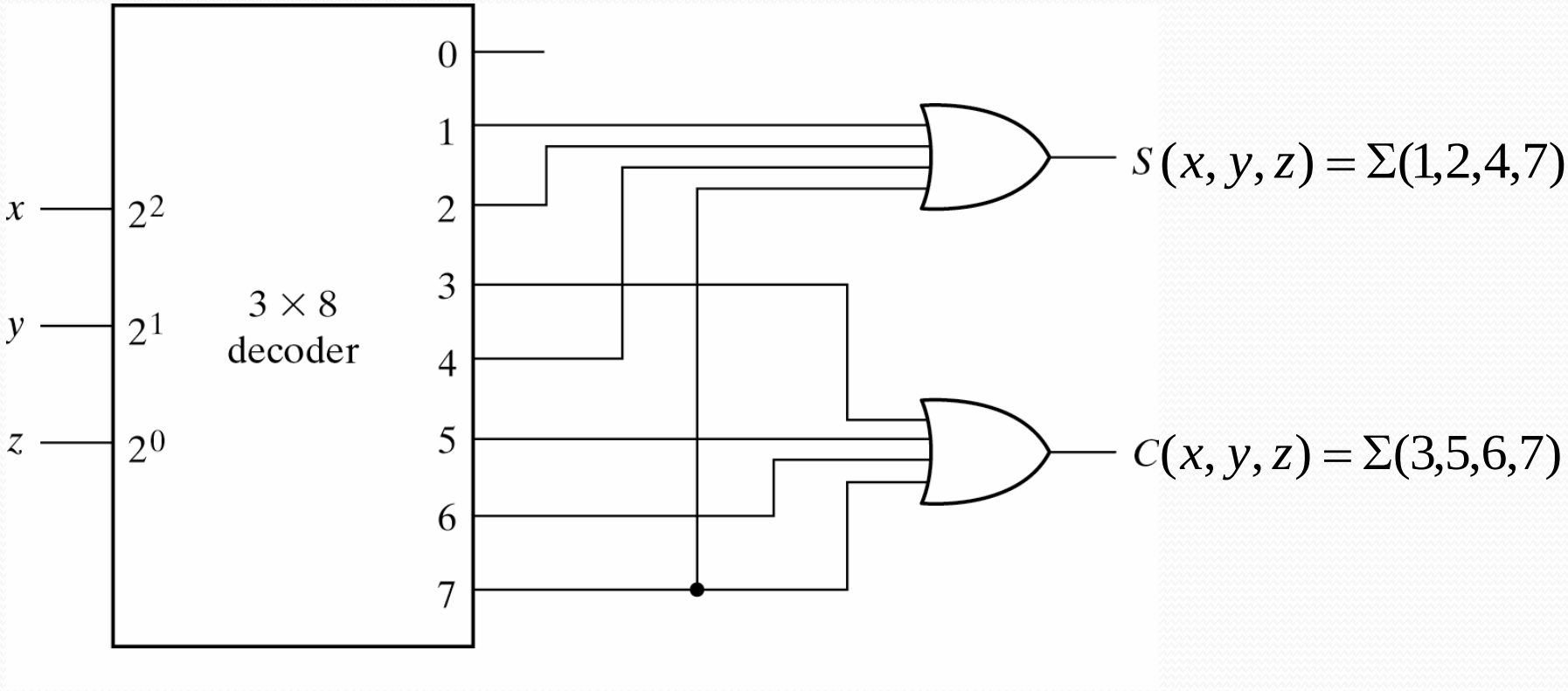


– ساخت تمام جمع کننده با دیکدر

- مداری است با سه ورودی، روابط مربوط به تمام‌جمع‌کننده:

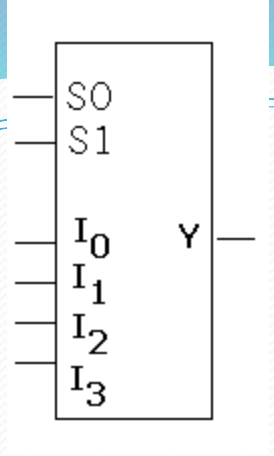
$S(x,y,z)=\sum(1,2,4,7)$

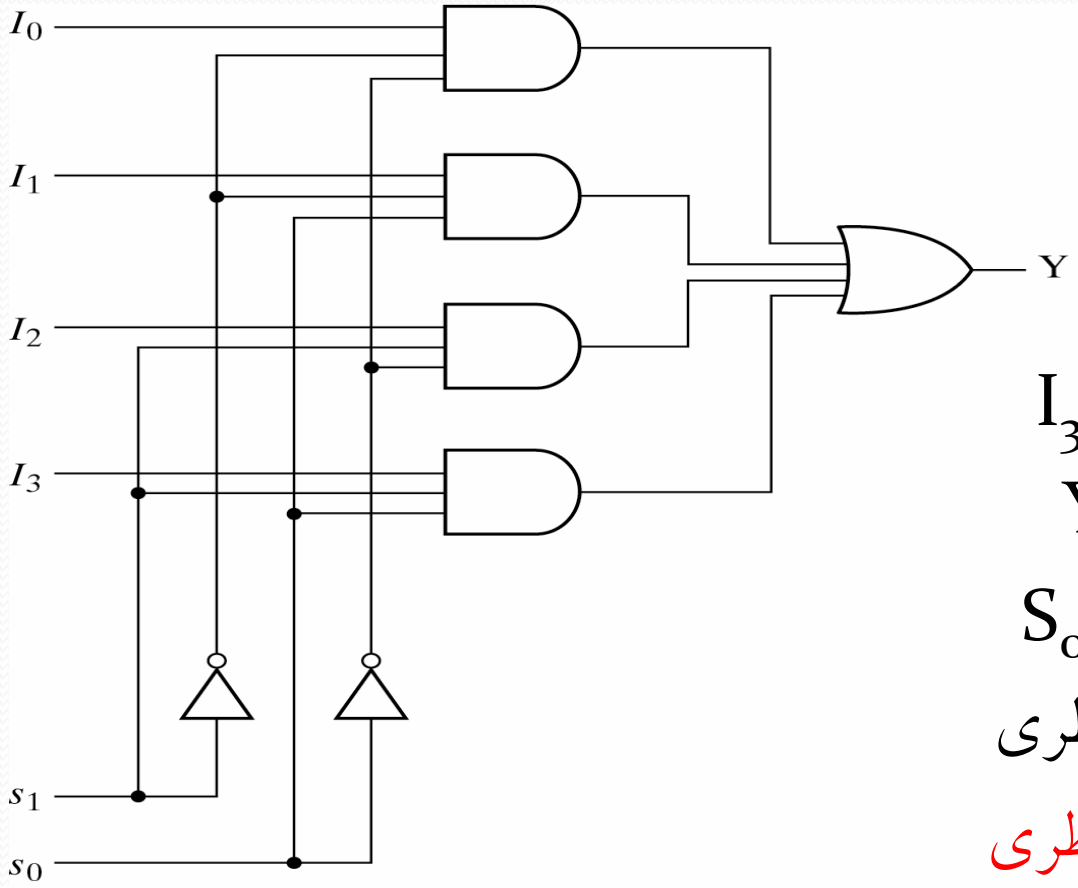
$C(x,y,z)=\sum(3,5,6,7)$



- ذکر چند مثال دیگر

MUX 4x1 – (مدار داخلی)





جدول عملکرد

s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

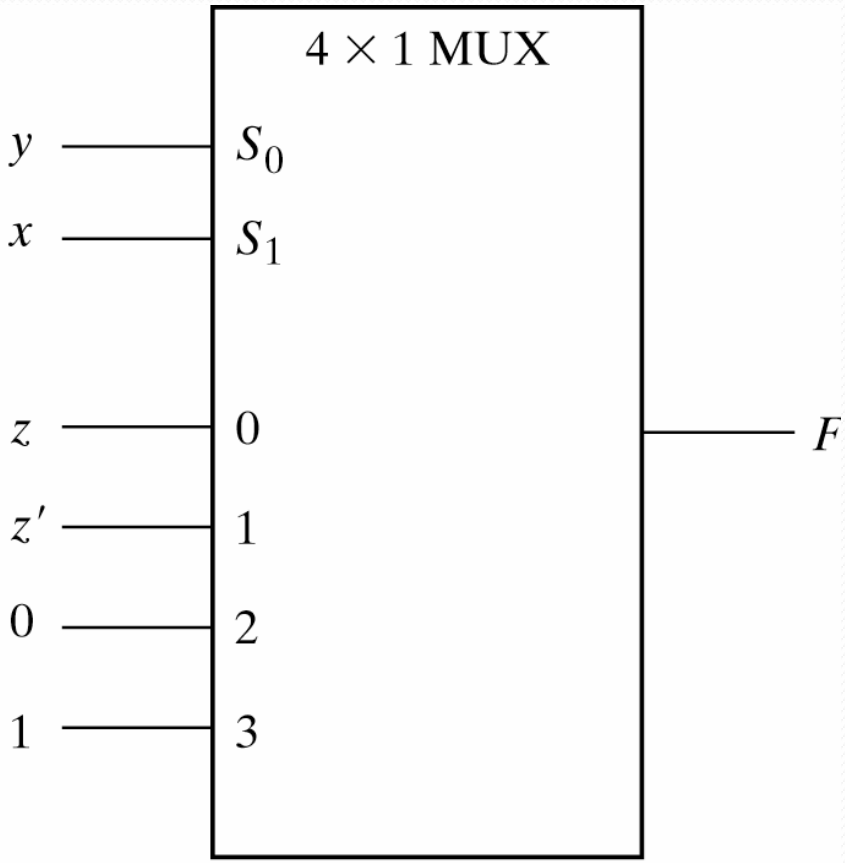
- ٤ خط ورودی I_o تا I_3
- یک خروجی با نام Y
- ٢ خط انتخاب S_1 و S_o
- جدول ارزش ٦٤ سطری
- جدول عملکرد ٤ سطری

– پیاده سازی توابع بولی توسط مولتی پلکسرها

- جدول درستی تابع بولی زیر همراه با نحوهٔ پیاده‌سازی آن توسط MUX نشان داده شده است:

$$F(x, y, z) = \Sigma(1,2,6,7)$$

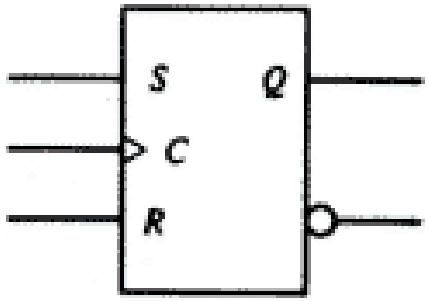
x	y	z	F	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	



فلیپ فلاپ SR

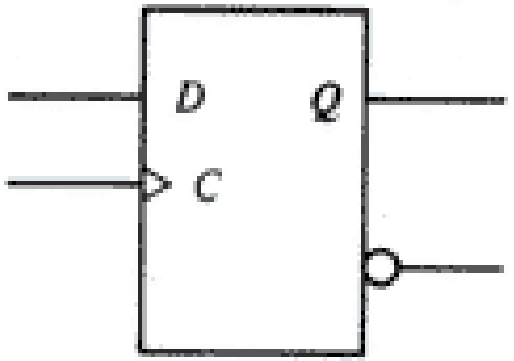
- **حساس به لبه‌ی بالا رونده**

S	R	Q (t + 1)	
0	0	Q (t)	بلاتغییر
0	1	0	پاک شدن، 0
1	0	1	نشانده شدن، 1
1	1	?	نامعین



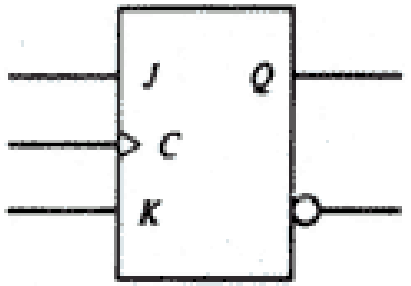
فلیپ فلاپ D

D	Q (t + 1)	
0	0	پاک شدن، 0
1	1	نشانده شدن، 1



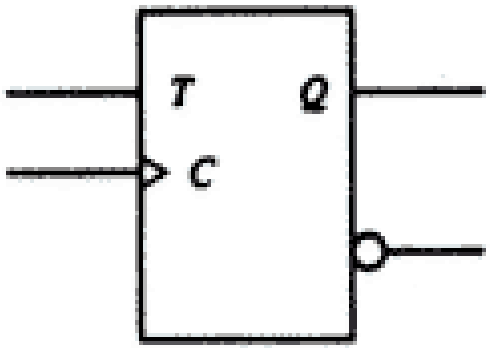
فلیپ فلاپ JK

J	K	Q (t + 1)	
0	0	Q (t)	بلاتغییر
0	1	0	پاک شدن، 0
1	0	1	نشانده شدن، 1
1	1	Q' (t)	متمم



فلیپ فلاپ T

T	Q (t + 1)	
0	Q (t)	بلاتغییر
1	Q' (t)	متمم



جدول مشخصات فلیپ فلاپها و معادلات آنها

J	K	$Q(t+1)$	
0	0	$Q(t)$	بدون تغییر
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	معکوس

$$Q(t+1) = JQ' + K'Q$$

$Q(t)$ = حالت فعلی
$Q(t+1)$ = حالت بعدی

T	$Q(t+1)$	
0	$Q(t)$	بدون تغییر
1	$Q'(t)$	معکوس

$$Q(t+1) = T \oplus Q = TQ' + T'Q$$

D	$Q(t+1)$	
0	0	Reset
1	1	Set

$$Q(t+1) = D$$

انواع مدلهای مدارهای ترتیبی:

مدل مور(MOORE): در این مدل، خروجی ها فقط تابع حالت فعلی است

مدل (MELAY): در این مدل خروجی ها بر اساس حالت فعلی و ورودی ها مشخص می‌گردد

کلیه مدارات ترتیبی در یکی از این دو مدل قرار می گیرند.

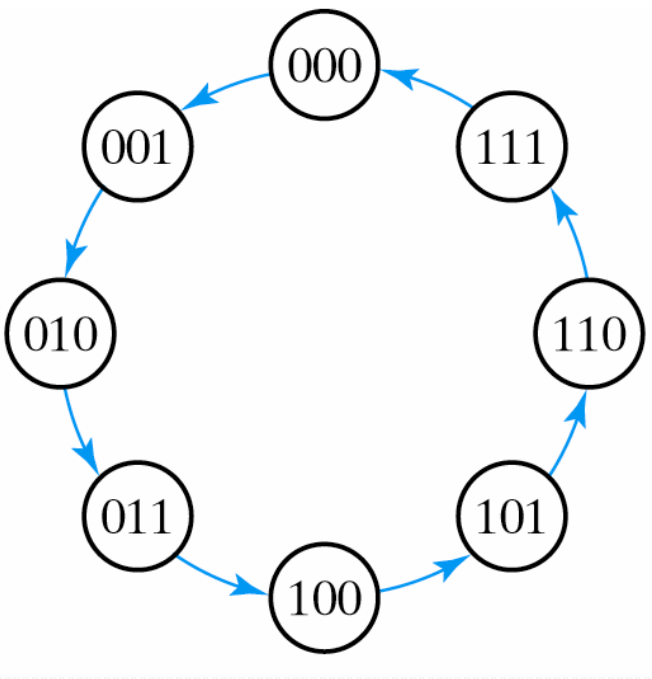
مدل مختلط: ماشین حالتی که دارای خروجی های نوع مور و میلی است

مراحل طراحی مدارهای ترتیبی همزمان

۱- توصیف و فهم مسأله
۲- بدست آوردن جدول حالت با استفاده از اطلاعات داده شده از مسأله
۳- تعیین تعداد فلیپ فلاپهای مورد نیاز
۴- انتخاب نوع فلیپ فلاپ
۵- یافتن جداول تحریک و خروجی مدار با استفاده از جدول حالت
۶- ساده سازی جهت یافتن توابع ورودی فلیپ فلاپها
۷- رسم نمودار منطقی

مثال

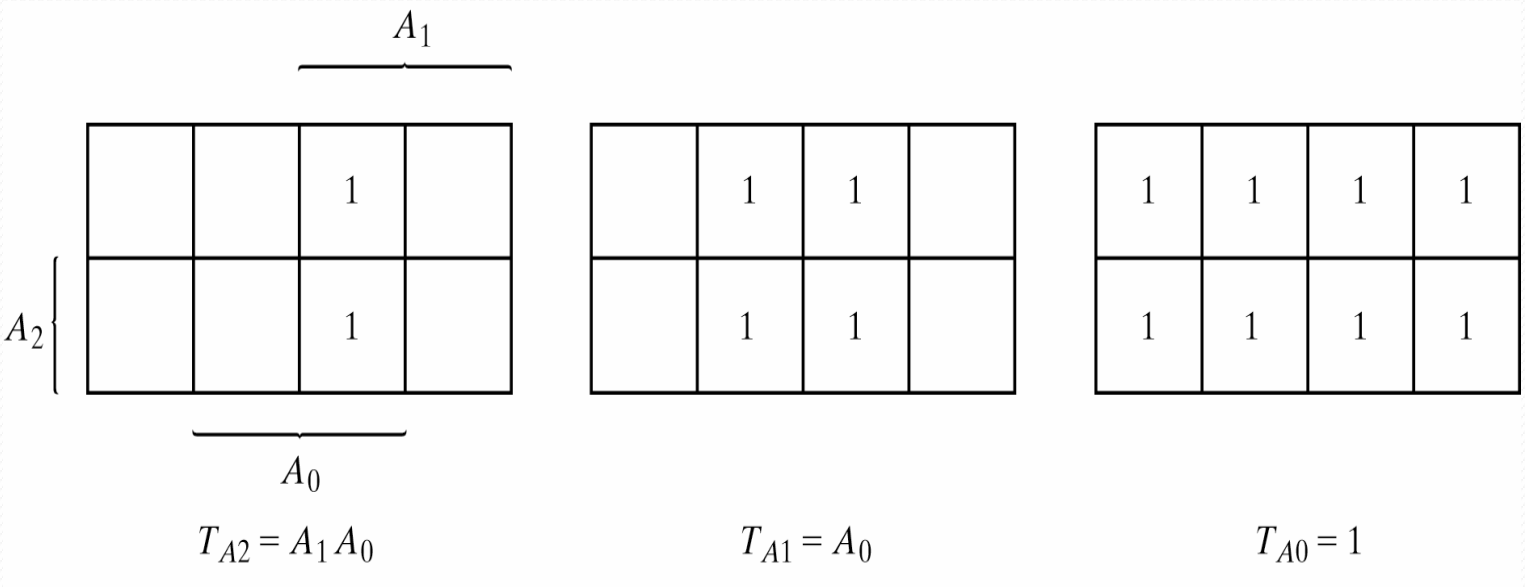
- یک شمارندهٔ سه بیتی با استفاده از فلیپ فلاپ T طراحی کنید که به ترتیب از 000 تا 111 را بشمارد.



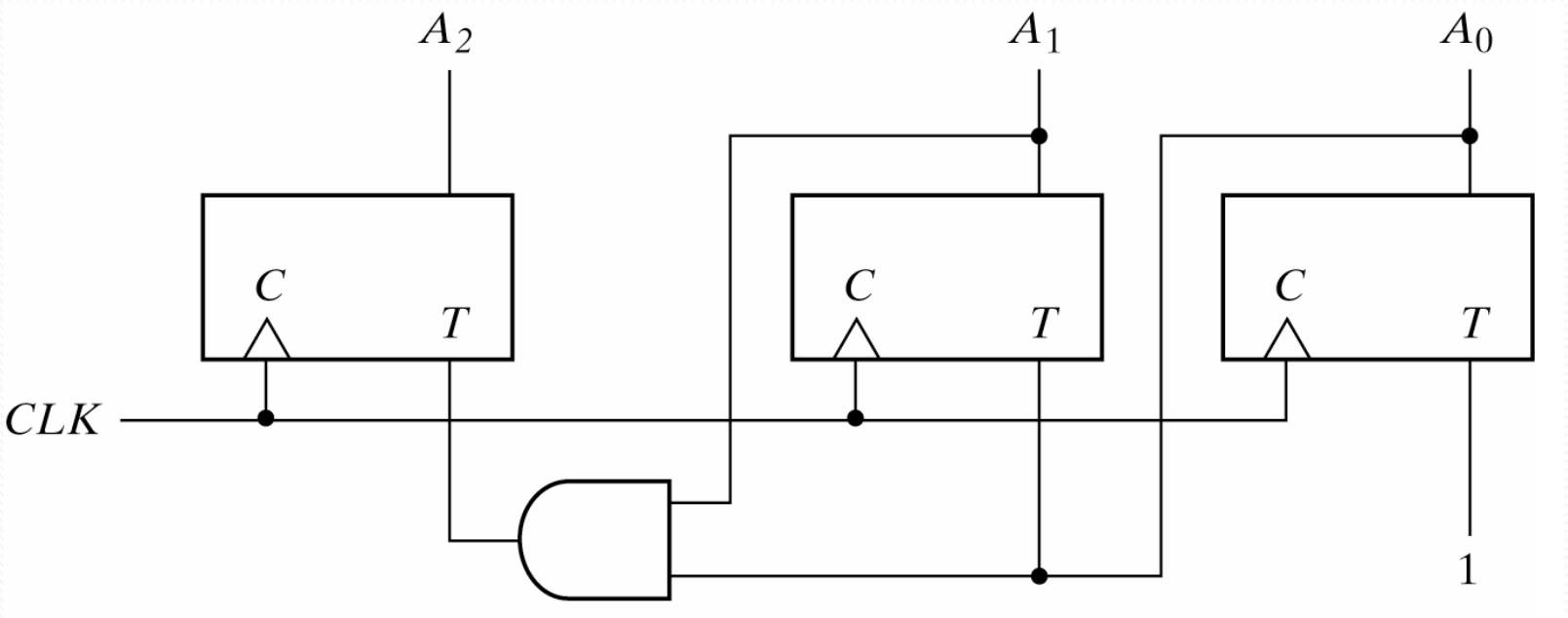
- جدول تحریک فلیپ فلاپها بصورت زیر می‌باشد:

حالت فعلی			حالت بعدی			ورودیهای فلیپ فلاپ		
A_2	A_1	A_0	A_2	A_1	A_0	TA_2	TA_1	TA_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

جداول کارنو خروجیهای مدار ترکیبی

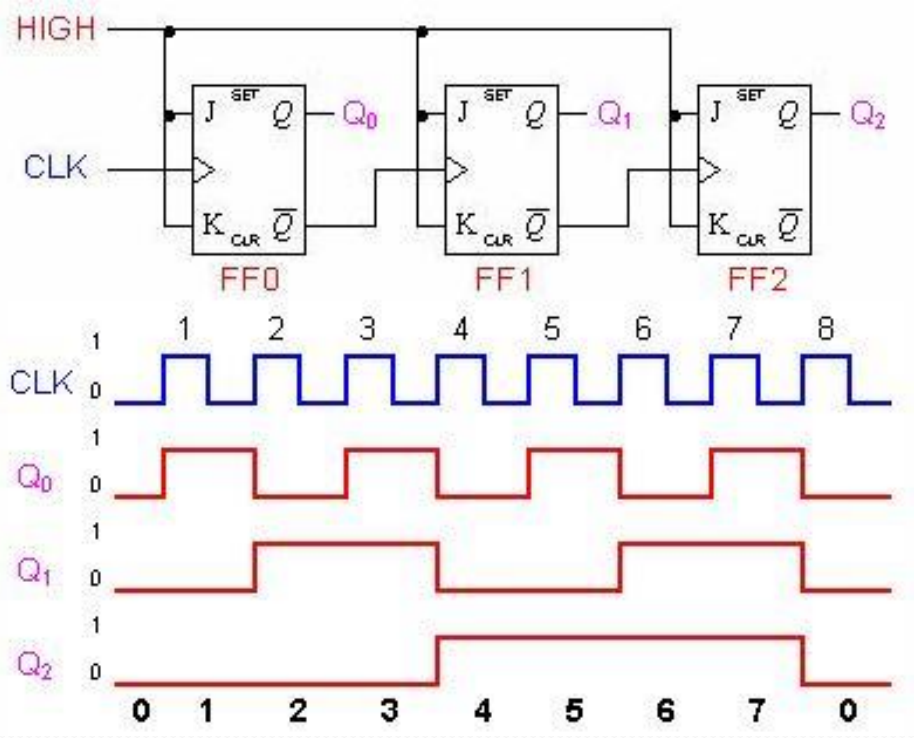


نمودار منطقی شمارندهٔ سه بیتی



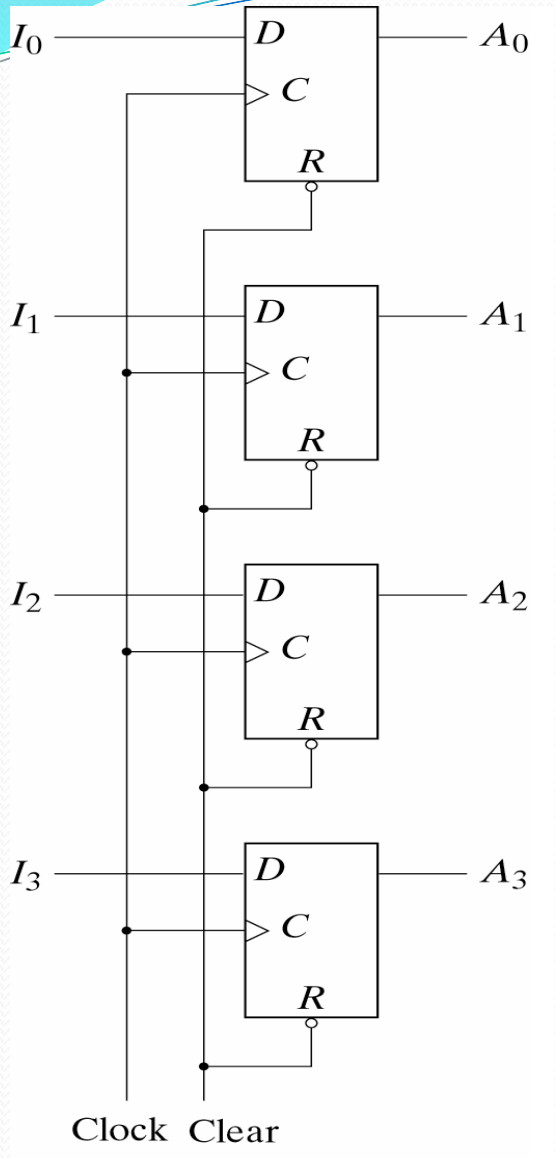
– شمارنده موج گونه دودویی

Q_3	Q_2	Q_1	Q_0
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



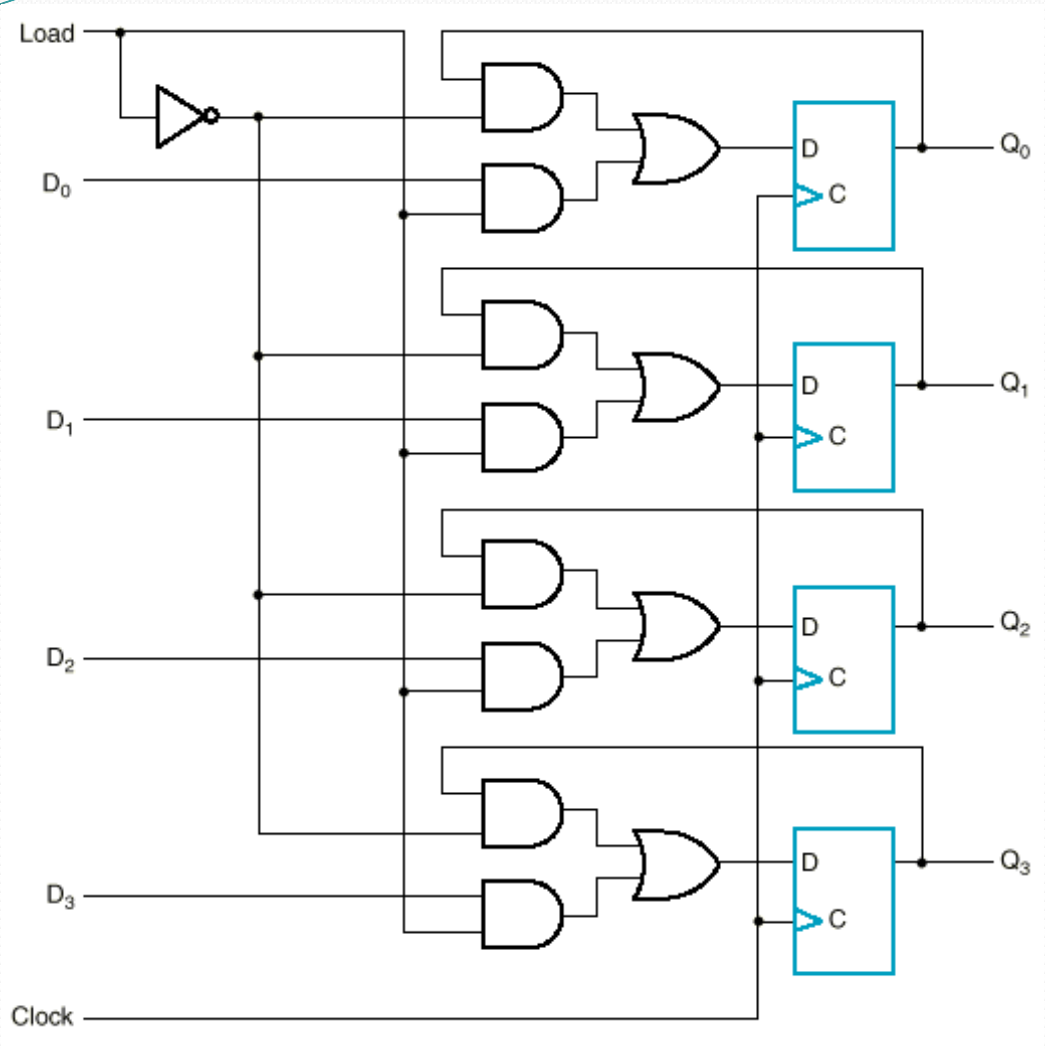
- Qo در اثر کلاک ورودی تغییر می کند. وقتی که Qo از ۱ به ۰ برود، Q1 تریگر می شود. لذا، خروجی آن معکوس می گردد. وقتی که Q1 از ۱ به ۰ برود، Q2 تریگر می شود. لذا، خروجی آن معکوس می گردد. و ....

رجیستر چهار بیتی



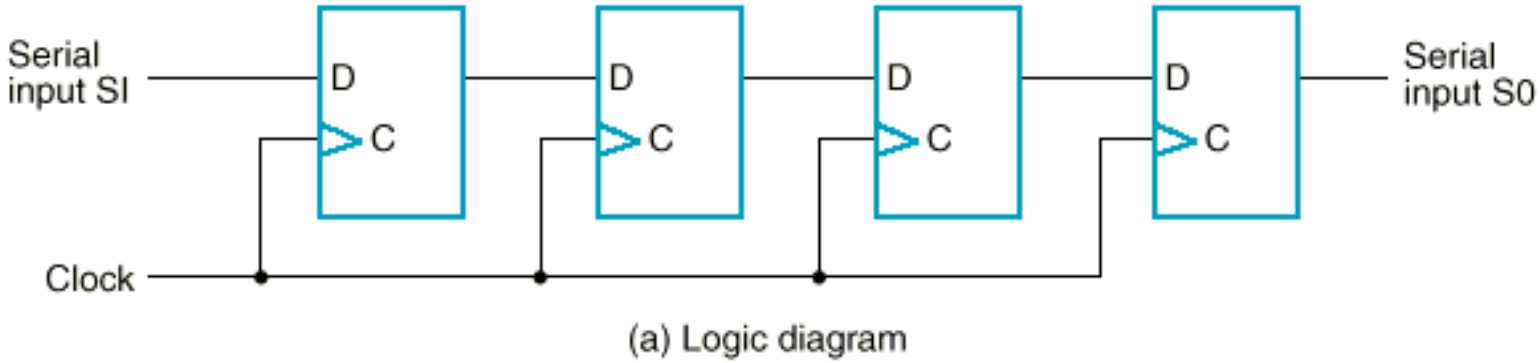
- در لبه مثبت کلاک، اطلاعات روی ورودیها به خروجی منتقل می گردند.
- برای خواندن محتویات رجیستر باید به خروجی آن نگاه کرد.
- اگر ورودی clear صفر شود باعث می گردد که تمام فلیپ فلاپها reset شوند. و خروجی رجیستر برابر صفر می شود.

– ثبات ۴ بیتی با قابلیت بار شدن موازی



- اگر load=1 باشد، در لبه بعدی کلاک، ورودیها به خروجی منتقل خواهند شد.
- اگر load=0 باشد، ورودی فلیپ فلاپها به خروجی آنها وصل است.

– نمودار بلوکی ثبات انتقالی چهار بیتی



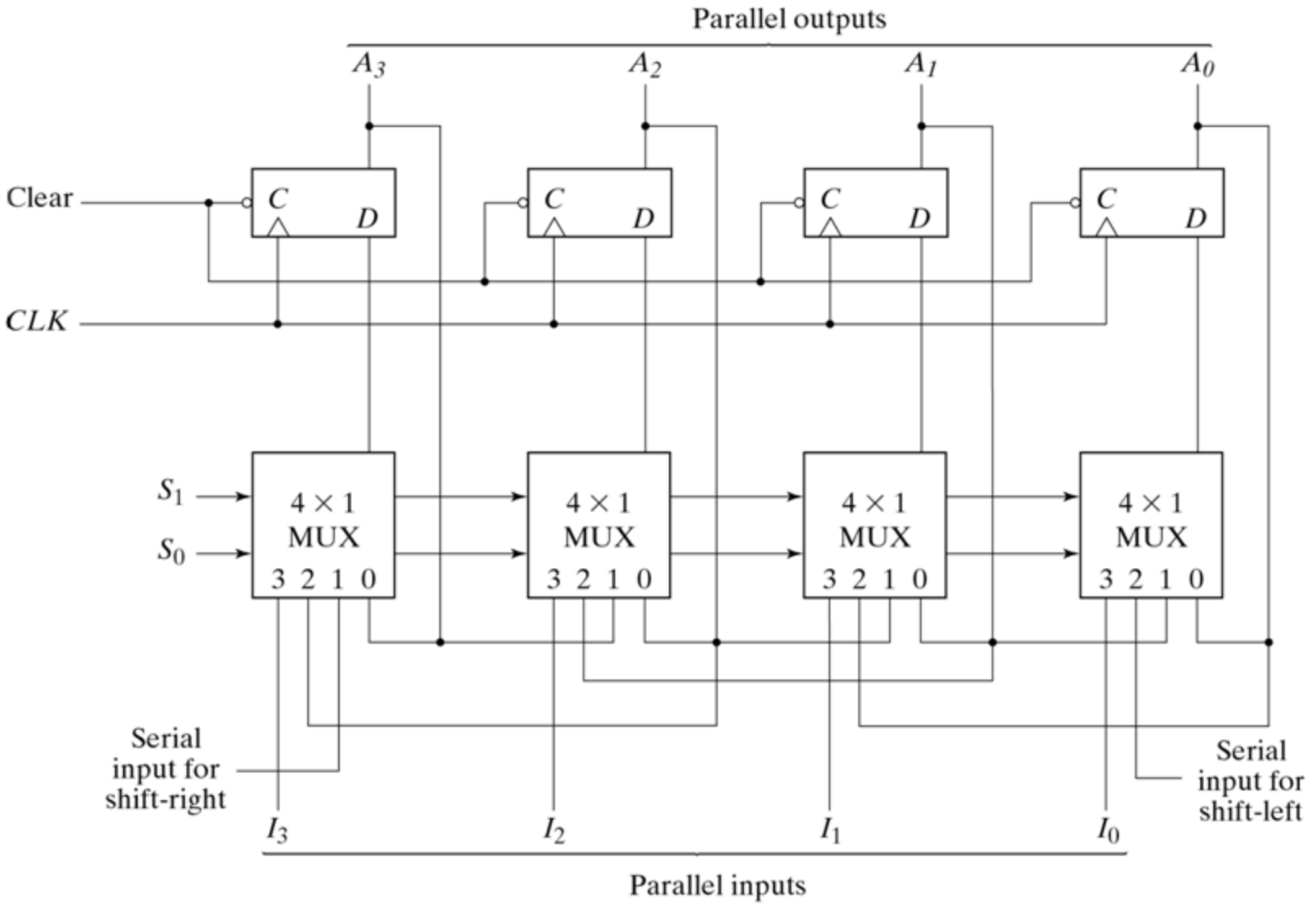
(a) Logic diagram

SISO(Serial In-Serial Out)

(b) Symbol

- در لبه مثبت کلاک، فلیپ فلاپ اول از ورودی نمونه می گیرد. در لبه مثبت بعدی، خروجی فلیپ فلاپ اول در فلیپ فلاپ دوم ذخیره می شود و فلیپ فلاپ اول یک نمونه تازه می گیرد. و ....

شیفت رجیستر یونیورسال



– شیفت رجیستر یونیورسال

Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

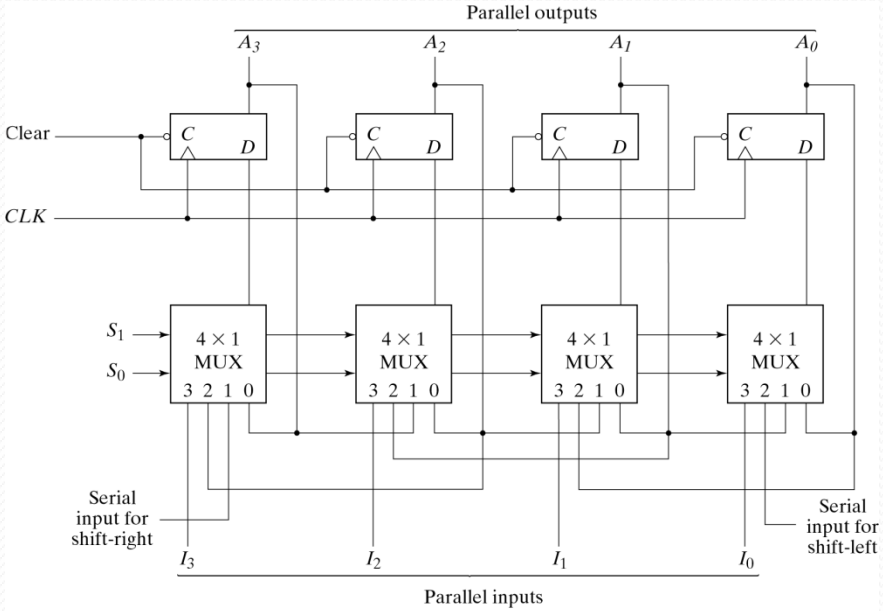


Fig. 6-7 4-Bit Universal Shift Register

- ورودیهای کنترل **S1** و **SO** مولتی پلکسرهای ٤ به ١ را کنترل می کنند.
- وقتی **S1SO=OO** باشد، هر مولتی پلکسرخروجی فلیپ فلاپ مربوطه را به ورودی آن وصل می کند.
- وقتی **S1SO=O1** باشد، هر مولتی پلکسرخروجی فلیپ فلاپ سمت چپ را به ورودی فلیپ فلاپ خودش هدایت می کند.
- وقتی **S1SO=1O** باشد، هر مولتی پلکسرخروجی فلیپ فلاپ سمت راست را به ورودی فلیپ فلاپ خودش هدایت می کند.
- وقتی **S1SO=11** باشد، هر مولتی پلکسرخروجی ورودی فلیپ فلاپ را به ورودی موازی مربوطه وصل می کند.

جلسه آینده

- PROM, PAL , PLA