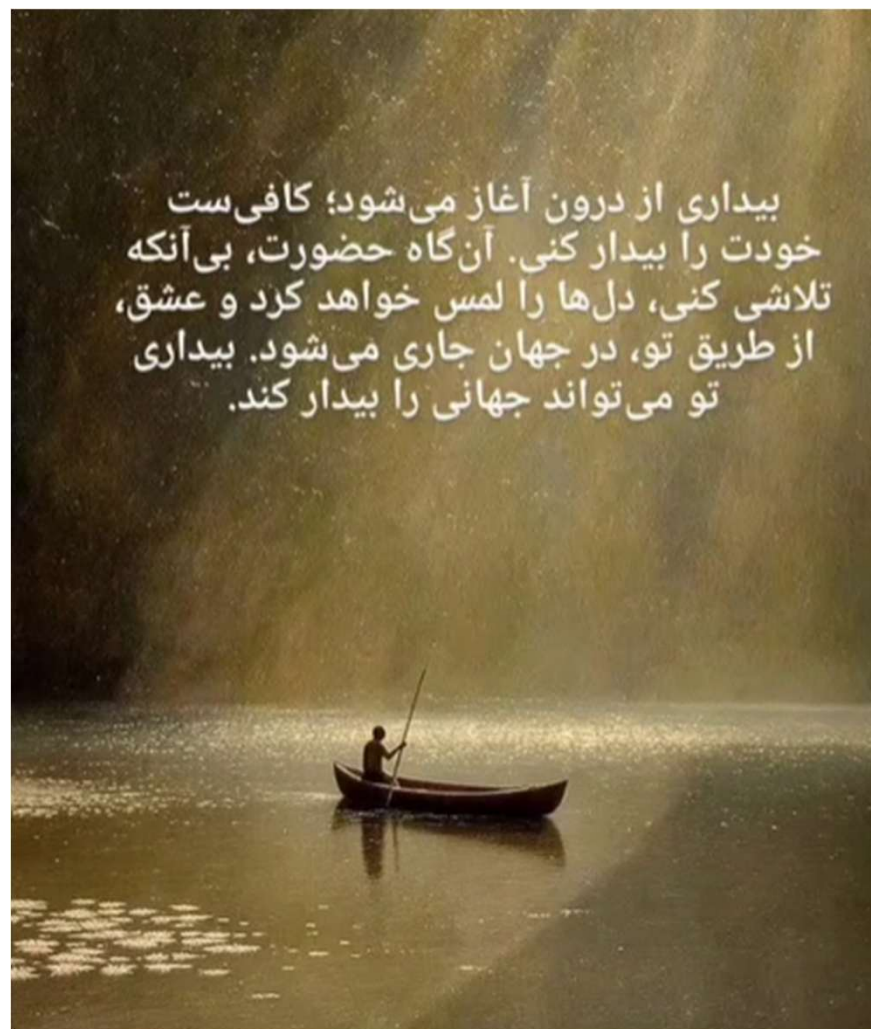


بسمه تعالی



مدارهای منطقی

فصل چهارم

منطق ترکیبی

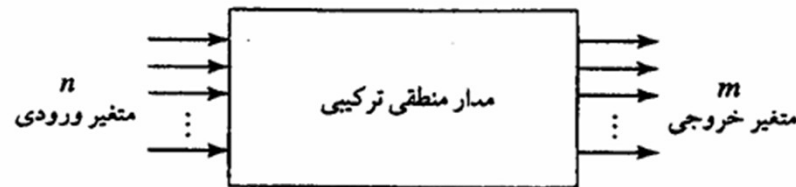
تحلیل و طراحی مدارات ترکیبی



۴-۱ مدارهای ترکیبی

مدارهای منطقی در سیستم‌های دیجیتال می‌توانند از نوع ترکیبی و یا ترتیبی باشند. یک مدار ترکیبی متشکل از تعدادی گیت منطقی است که خروجی آنها در هر لحظه از زمان مستقیماً به وسیله ورودی‌های همان لحظه معین می‌شود و به ورودی‌های قبلی بستگی ندارد. این نوع مدار، پردازشی را انجام می‌دهد که با مجموعه‌ای از توابع بولی مشخص می‌گردد. مدارهای ترتیبی علاوه بر گیت‌های منطقی از عناصر حافظه نیز استفاده می‌کنند. خروجی‌های آنها تابعی از ورودی‌ها و حالت عناصر حافظه است. در نتیجه خروجی یک مدار ترتیبی نه تنها به مقادیر فعلی ورودی‌ها بلکه به ورودی‌های قبلی وابسته بوده و عملکرد مدار باید به وسیله حالات داخلی و ترتیب زمانی ورودی‌ها مشخص گردد. مدارهای ترتیبی در فصل‌های ۵ و ۹ بحث شده‌اند.

یک مدار ترکیبی از متغیرهای ورودی، گیت‌های منطقی، و متغیرهای خروجی تشکیل شده است. گیت‌های منطقی سیگنال‌هایی را از ورودی‌ها دریافت کرده و سیگنال‌هایی را برای خروجی‌ها تولید می‌نمایند. این فرآیند اطلاعات دودویی مفروض در ورودی را به اطلاعات موردنیاز در خروجی تبدیل می‌کند. نمودار کلی یک مدار ترکیبی در شکل ۴-۱ دیده می‌شود. n متغیر دودویی ورودی از منبع



شکل ۴-۱. نمودار بلوکی یک مدار ترکیبی



۳-۴ روش طراحی

طراحی مدارهای ترکیبی با مشخصات مسئله آغاز و به فرم نمودار مدار منطقی یا مجموعه‌ای از توابع بول که به کمک آنها نمودار منطقی حاصل می‌شود، پایان می‌یابد. روال شامل موارد زیر است:

- ۱- با استفاده از مشخصات مدار تعداد ورودی‌ها و خروجی‌ها را معین کرده و به هر کدام سمبلی تخصیص دهید.

- ۲- جدول درستی مربوط به ورودی‌ها و خروجی‌های مدار را تشکیل دهید.

- ۳- توابع بولی ساده شده را برای هر خروجی به صورت تابعی از متغیرهای ورودی بدست آورید.

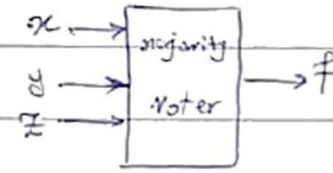
- ۴- نمودار منطقی را رسم کرده و صحت طراحی را تحقیق نمایید.

جدول درستی یک مدار ترکیبی، از ستون‌های ورودی و ستون‌های خروجی تشکیل می‌شود. ستون‌های ورودی از 2^n ترکیب مربوط به n متغیر ورودی بدست می‌آید. مقادیر دودویی خروجی‌ها از مشخصات بیان شده در مسئله حاصل می‌گردد. توابع خروجی مشخص شده در جدول درستی تعریف دقیقی از مدار ترکیبی را بدست می‌دهند. تفسیر لفظی صحیح جدول درستی از اهمیت خاصی برخوردار است. اغلب مشخصات لفظی کامل نیستند و تفسیر غلط ممکن است جدول درستی غلطی را تولید کند. توابع دودویی خروجی لیست شده در جدول با روش‌های موجود مانند دستکاری جبری، نقشه کارنو یا برنامه‌های ساده‌سازی مبتنی بر کامپیوتر ساده می‌شوند. غالباً عبارات ساده شده متعددی حاصل می‌شوند که باید مناسب‌ترین را انتخاب کرد.



مثال: مداری طراحی کنید که براساس اکثریت، اء سه متغیر را، را صادر کنند.

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

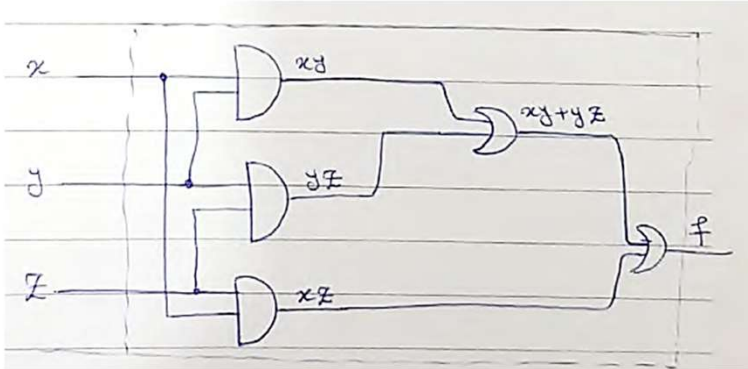


$$f = \sum (3, 5, 6, 7)$$

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

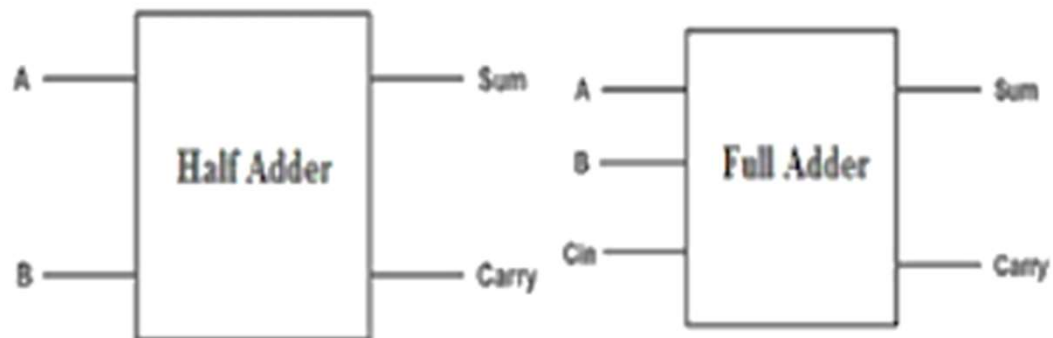
x \ yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$f = yz + xy + xz$$



جمع کننده ها

دو نوع جمع کننده داریم: (۱) نیم جمع کننده (۲) تمام جمع کننده



نیم جمع کننده

با توجه به توضیحات لفظی یک نیم جمع کننده، در می یابیم که مدار نیاز به دو ورودی دودویی و دو خروجی دودویی دارد. متغیرهای ورودی بیت های مضاف و مضاف الیه را مشخص می کنند. متغیرهای خروجی جمع و نقلی را تولید می نمایند. ما سبمل های x و y را به دو ورودی و S (برای جمع) و C (نقلی) را به خروجی ها تخصیص می دهیم. جدول درستی برای نیم جمع کننده در جدول ۳-۴ نشان داده شده است. خروجی C فقط هنگامی 1 است که هر دو ورودی 1 باشند. خروجی S ، بیت کم ارزش تر حاصل جمع را نشان می دهد.

توابع بولی ساده شده برای دو خروجی مستقیماً از جدول درستی بدست می آیند. عبارات جمع

$$S = x'y + xy'$$

$$C = xy$$

نمودار منطقی نیم جمع کننده پیاده شده با جمع حاصلضرب ها در شکل ۴-۵ (الف) دیده می شود. می توان آن را با گیت های XOR و AND طبق شکل ۴-۵ (ب) هم پیاده کرد. این نوع برای ساخت جمع کننده کامل از دو نیم جمع کننده به کار می رود.

جدول ۳-۴. نیم جمع کننده

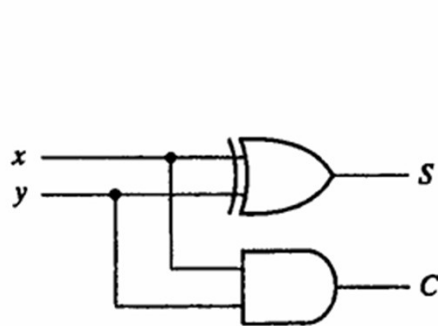
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



دانشگاه سمنان
Semnan University
پردیس فیروزانگان

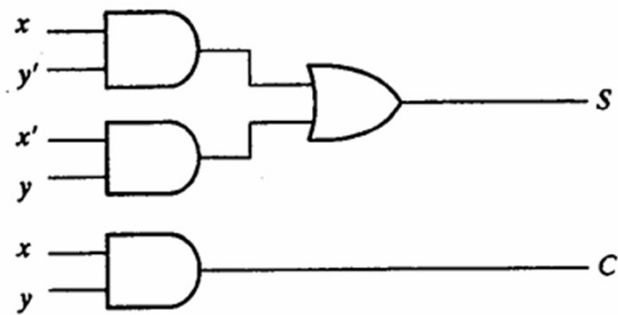
جدول ۳-۴. نیم جمع کننده

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



$$S = x \oplus y \quad (\text{ب})$$

$$C = xy$$



$$S = xy' + x'y \quad (\text{الف})$$

$$C = xy$$

شکل ۴-۵. پیاده سازی نیم جمع کننده



جمع‌کننده کامل

یک جمع‌کننده کامل مداری ترکیبی است که جمع حسابی سه بیت را تشکیل می‌دهد. این مدار دارای سه ورودی و دو خروجی است. دو متغیر ورودی که با x و y نشان داده شده‌اند دو بیت با ارزش جمع شونده را نشان می‌دهند. ورودی سوم، z ، نقلی حاصل از مکان کم‌ارزش‌تر قبلی است. دو خروجی لازم است زیرا جمع حسابی سه رقم دودویی بین 0 تا 3 می‌باشد و اعداد 2 و 3 به دو رقم دودویی نیاز دارند. دو خروجی با سمبل S برای جمع و C برای نقلی مشخص شده‌اند. متغیر دودویی S مقدار کم‌ارزش‌تر جمع را بدست می‌دهد. متغیر دودویی C نقلی خروجی را بیان‌گر است. جدول درستی جمع‌کننده کامل در جدول ۴-۴ دیده می‌شود. هشت سطر زیر سه متغیر همه ترکیبات ممکن سه متغیر را نشان می‌دهند. متغیرهای خروجی از جمع حسابی بیت‌های ورودی معین می‌شوند. وقتی همه بیت‌های ورودی 0 هستند، خروجی 0 است. خروجی S هنگامی 1 می‌شود که فقط یک ورودی برابر 1 باشد و یا اگر هر سه ورودی 1 باشند. خروجی C هم موقعی 1 است که دو یا سه ورودی برابر 1 باشند.



جدول ۴-۴. 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

		yz		y	
		00	01	11	10
x	0		1		1
x	1	1		1	
		z			

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

		yz		y	
		00	01	11	10
x	0			1	
x	1		1	1	1
		z			

$$S = xy + xz + yz$$

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



پیاده‌سازی با گیت‌های منطقی، همان مقادیر به عنوان متغیرهای بول تعبیر می‌شوند. نقشه خروجی‌های جمع‌کننده کامل در شکل ۴-۶ ملاحظه می‌شود. عبارات ساده شده عبارتند از:

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

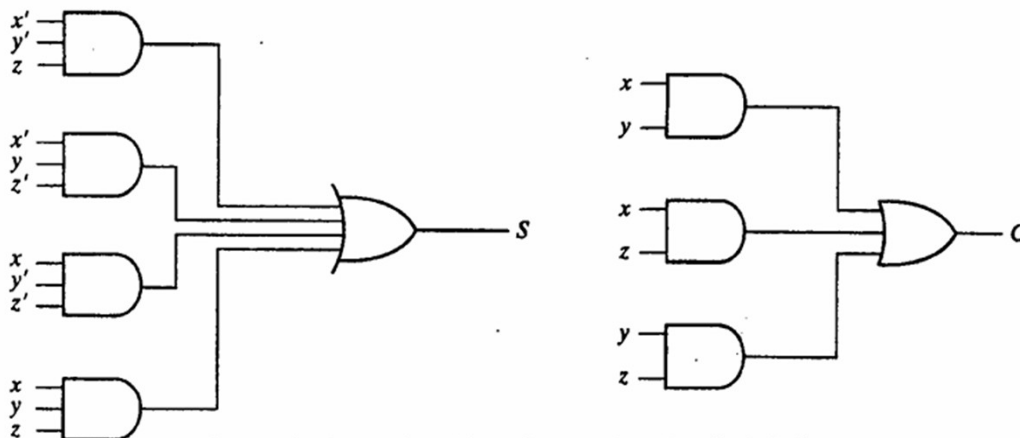
$$C = xy + xz + yz$$

مودار منطقی پیاده شده به صورت جمع حاصلضرب‌ها در شکل ۴-۷ مشاهده می‌شود. می‌توان با دو نیم جمع‌کننده و یک OR هم طبق شکل ۴-۸ آن را پیاده‌سازی کرد. خروجی S از نیم جمع‌کننده دوم XOR متغیر Z و خروجی نیم جمع‌کننده اول حاصل می‌شود، زیرا

$$\begin{aligned} S &= z \oplus (x \oplus y) \\ &= z'(xy' + x'y) + z(xy' + x'y)' \\ &= z'(xy' + x'y) + z(xy + x'y') \\ &= xy'z' + x'yz' + xyz + x'y'z \end{aligned}$$

و نقلی خروجی برابر است با

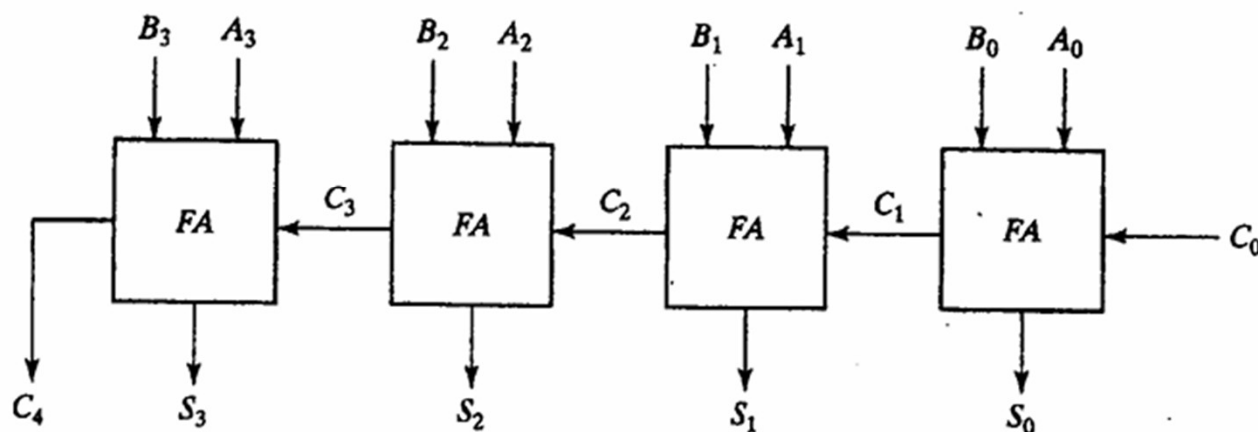
$$C = z(xy' + x'y) + xy = xy'z + x'yz + xy$$



شکل ۴-۷. پیاده‌سازی جمع‌کننده کامل با جمع حاصلضرب‌ها

جمع‌کننده دودویی

یک جمع‌کننده دودویی مداری دیجیتال است که جمع حسابی دو عدد دودویی را تولید می‌کند. می‌توان آن را از به هم پیوستن متوالی جمع‌کننده کامل ساخت، و در آن هر خروجی نقلی از هر جمع‌کننده کامل به ورودی نقلی جمع‌کننده کامل بعدی زنجیروار بسته می‌شود. شکل ۴-۹ اتصالات درونی مدار چهار جمع‌کننده کامل (FA)، برای تهیه جمع‌کننده دودویی ۴ بیت با نقلی موج‌گونه را نشان می‌دهد.



شکل ۴-۹. جمع‌کننده ۴ بیت



به منظور تشریح بیشتر مثالی را با اعداد دودویی $A = 1011$ و $B = 0011$ در نظر بگیرید. حاصل جمع آنها $S = 1110$ است که از جمع چهار بیت مطابق زیر بدست می آید.

اندیس i	3	2	1	0	
نقلی ورودی	0	1	1	0	C_i
مضاف	1	0	1	1	A_i
مضاف الیه	0	0	1	1	B_i
حاصل جمع	1	1	1	0	S_i
نقلی خروجی	0	0	1	1	C_{i+1}



تفریق کننده ها

دو نوع تفریق کننده داریم: (۱) نیم تفریق کننده (۲) تمام تفریق کننده

یک مدار نیم تفریق گر

یک نیم تفریق گر، یک مدار منطقی است؛ که عمل تفریق را بر روی دو رقم باینری، انجام می‌دهد. این نیم تفریق گر، یک اختلاف و یک بیت قرضی، برای مرحله‌ی بعد، تولید می‌کند.



نماد	جدول درستی			
	Y	X	اختلاف	رقم قرضی
	0	0	0	0
	0	1	1	0
	1	0	1	1
	1	1	0	0



یک مدار تفریق گر کامل

تفاوت اصلی بین مدار تفریق گر کامل و نیم تفریق گر پیشین، این است؛ که تفریق گر کامل، دارای سه ورودی می باشد. دو ورودی داده‌ی تک بیتی X (عددی که از آن تفریق، انجام می شود) و Y (مفروق)، همانند قبل، به علاوه‌ی یک ورودی قرضی (B_{in}) اضافی که بتواند رقم قرضی را از فرآیند تفریق مرحله‌ی قبلی، دریافت کند

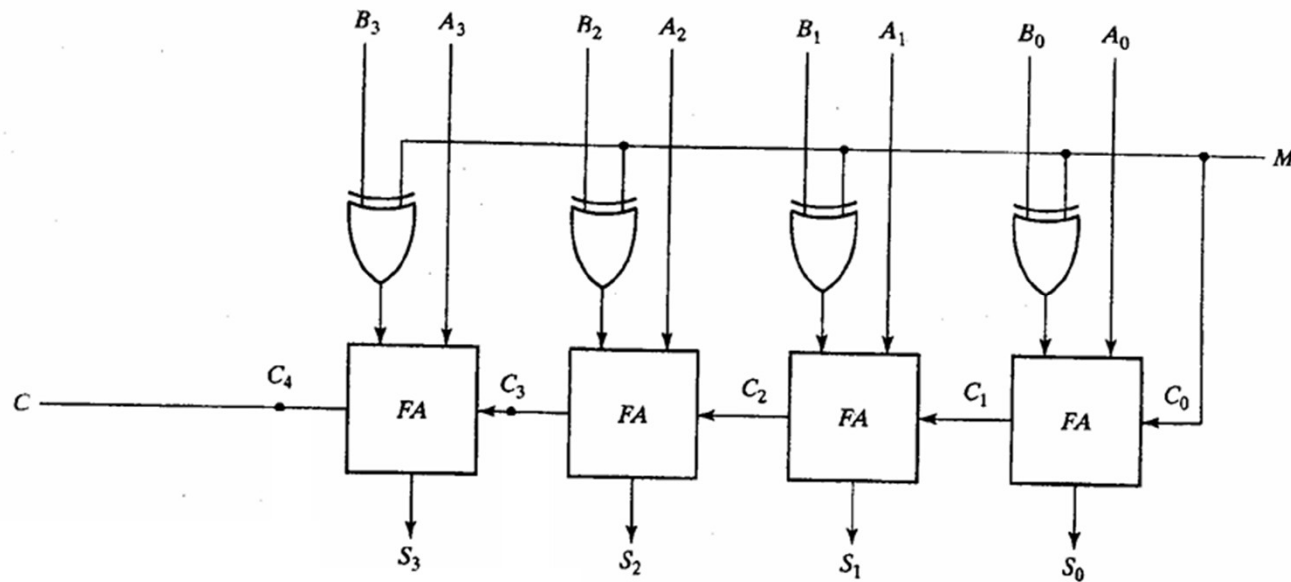


$$\begin{array}{r} B_{IN} \\ X \\ - Y \\ \hline B_{OUT} \leftarrow DIFF. \end{array}$$

نماد	جدول درستی				
	B-in	Y	X	Diff.	B-out
	0	0	0	0	0
	0	0	1	1	0
	0	1	0	1	1
	0	1	1	0	0
	1	0	0	1	1
	1	0	1	0	0
	1	1	0	0	1
	1	1	1	1	1



مدار جمع کننده-تفریق گر

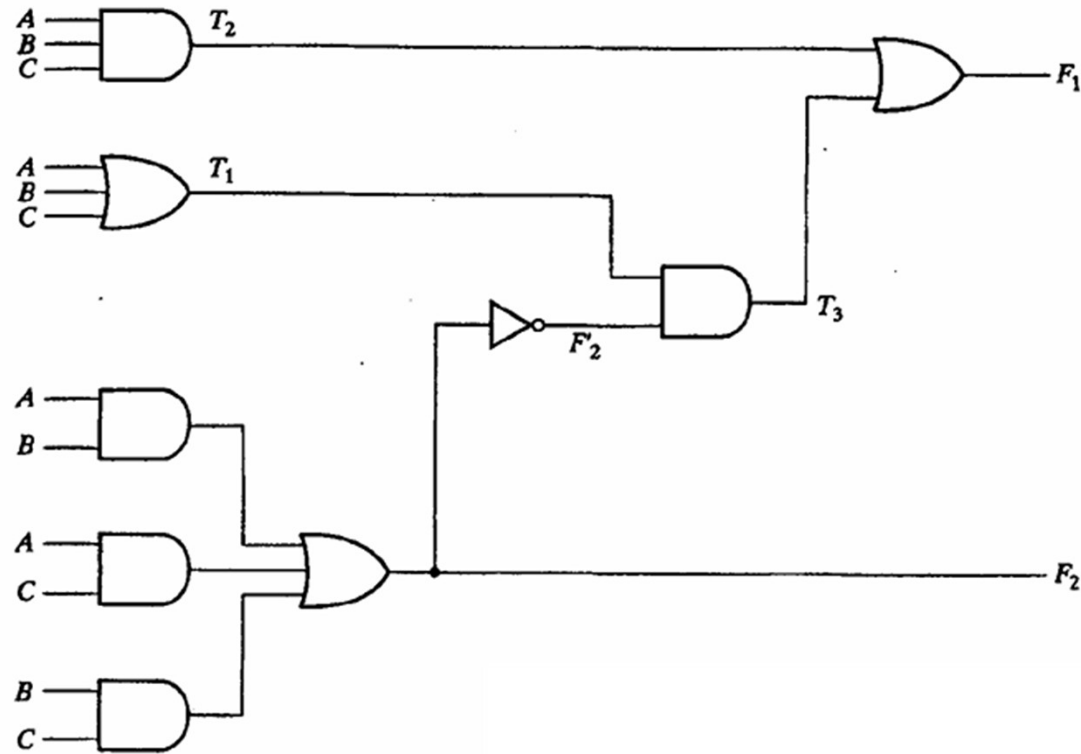


وقتی $M = 0$ است، مدار یک جمع کننده و وقتی $M = 1$ باشد، مدار یک تفریق گر خواهد بود. هر گیت XOR ورودی M و یکی از ورودی های B را دریافت می کند. وقتی $M = 0$ است داریم، $B \oplus 0 = B$. جمع کننده کامل B را دریافت می نماید، نقلی ورودی 0 است و بنابراین مدار عمل A بعلاوه B را اجرا می کند. اگر $M = 1$ باشد، $B \oplus 1 = B'$ بوده و $C_0 = 1$ است. ورودی های B همگی متمم شده و از طریق ورودی نقلی، یک 1 به آن اضافه می شود. در این حالت مدار یک عمل A بعلاوه متمم 2 عدد B را انجام می دهد.



تحلیل مدارات ترکیبی

روش تحلیل مدارات ترکیبی عکس روش طراحی آن است. به این معنی که با مدار منطقی آغاز و با توابع منطقی یا توزیع و طرز کار آنها پایان می یابد. برای این کار از ابتدای مدار منطقی شروع کرده و خروجی ها را بدست می آوریم و به سمت انتهای مدار می رویم.





برای این سه خروجی عبارتند از:

$$F_2 = AB + AC + BC$$

$$T_1 = A + B + C$$

$$T_2 = ABC$$

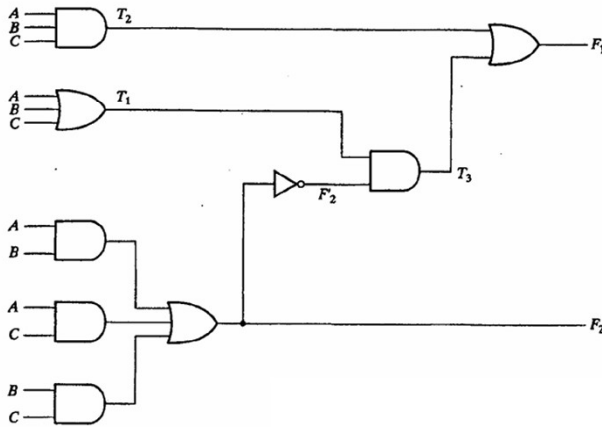
اکنون خروجی گیت‌هایی که تابعی از سمبل‌های قبلی می‌باشند را ملاحظه می‌نماییم:

$$T_3 = F_2' T_1$$

$$F_1 = T_3 + T_2$$

برای بدست آوردن F_1 برحسب A ، B و C ، یک سری جایگزینی‌ها را به فرم زیر انجام می‌دهیم:

$$\begin{aligned} F_1 &= T_3 + T_2 = F_2' T_1 + ABC = (AB + AC + BC)'(A + B + C) + ABC \\ &= (A' + B')(A' + C')(B' + C')(A + B + C) + ABC \\ &= (A' + B'C')(AB' + AC' + BC' + B'C) + ABC \\ &= A'BC' + A'B'C + AB'C' + ABC \end{aligned}$$





- برای تهیه مستقیم جدول درستی از نمودار منطقی و بدون نیاز به توابع بول به طریق زیر عمل کنید.
- ۱- تعداد متغیرهای ورودی در مدار را مشخص کنید. برای n ورودی 2^n ترکیب از ورودی‌ها را تشکیل دهید. آنگاه اعداد دودویی را در جدول از 0 تا $2^n - 1$ لیست نمایید.
 - ۲- خروجی‌های گیت‌های انتخابی را با سمبل‌های دلخواه برچسب بزنید.
 - ۳- برای آن دسته از خروجی‌های گیت‌ها که فقط تابعی از متغیرهای ورودی هستند جدول درستی را بدست آورید.
 - ۴- برای بدست آوردن خروجی‌های گیت‌هایی که تابعی از مقادیر تعریف شده قبلی هستند پیش بروید تا ستون همه خروجی‌ها معین شود.

A	B	C	F_2	F_2	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1

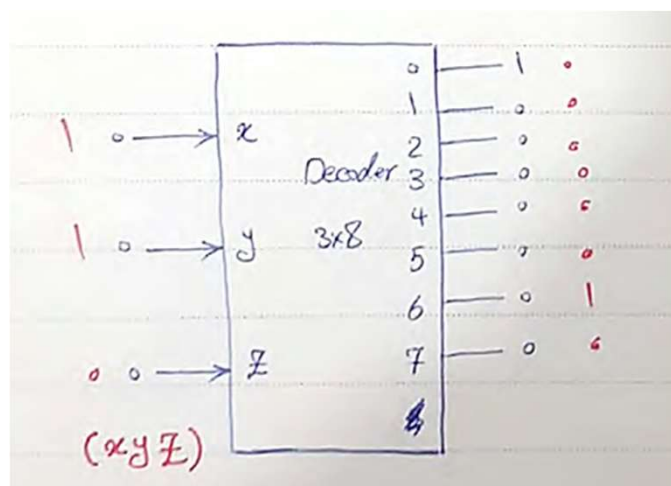


۴-۸ دیکدرها

کمیت‌های گسسته اطلاعاتی در سیستم‌های دیجیتال با کدهای دودویی نشان داده می‌شوند. یک کد دودویی n بیتی قادر است تا 2^n عنصر گسسته اطلاعات کد شده را نشان دهد. یک دیکدر مداری ترکیبی است که اطلاعات دودویی را از n خط ورودی به حداکثر 2^n خط خروجی منحصر به فرد تبدیل می‌کند. اگر کد n بیتی دارای ترکیبات بی‌استفاده باشد، دیکدر ممکن است خروجی‌های کمتر از 2^n داشته باشد. دیکدرهایی که در اینجا ارائه شده‌اند دیکدرهای n به m خوانده می‌شوند که $m \leq 2^n$ است. هدف از آنها تولید 2^n مینترم (یا کمتر) از n متغیر ورودی است.

جدول ۴-۶. جدول درستی دیکدر ۳ به ۸ خط

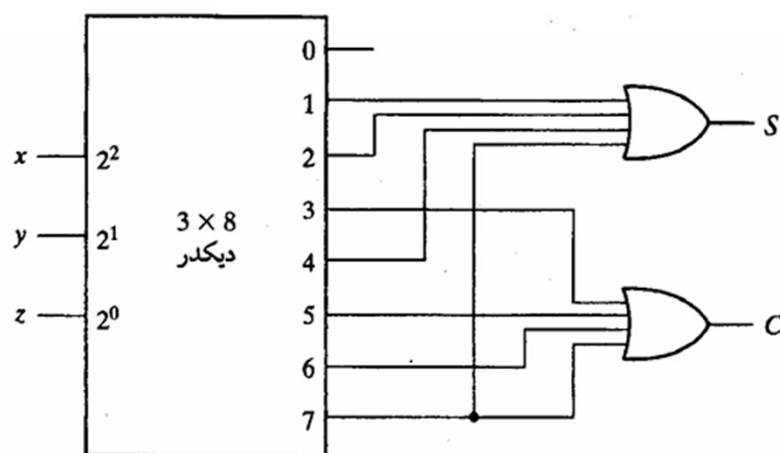
ورودی‌ها			خروجی‌ها							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1





پیاده‌سازی مدار منطقی ترکیبی

یک دیکدر، 2^n مینترم را برای n متغیر ورودی تهیه می‌کند. چون هر تابع بولی می‌تواند برحسب جمع مینترم‌ها بیان شود، می‌توان از دیکدر برای تولید مینترم استفاده کرده و با یک گیت OR بیرونی جمع منطقی آنها را تشکیل داد. به این ترتیب هر مدار ترکیبی n ورودی و m خروجی با یک دیکدر n به 2^n و m گیت OR قابل پیاده‌سازی است.



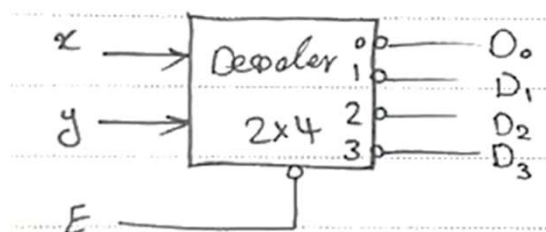
شکل ۴-۲۱. پیاده‌سازی یک جمع‌کننده کامل با دیکدر

با توجه به جدول درستی جمع‌کننده کامل (جدول ۴-۴)، توابع مدار ترکیبی را به صورت مجموع مینترم‌ها بدست می‌آوریم:

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

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

دیکدر معمولاً دارای یک یا دو ورودی تواناساز یا فعالساز برای کنترل کار مدار می باشند. مدار با خروجی های متمم شده و یک ورودی فعالساز متمم شده کار می کند. دیکدر هنگامی که E برابر 0 باشد فعال می گردد. هر بار تنها یک خروجی برابر 0 بوده و دیگر خروجی ها در وضعیت 1 قرار دارند. وقتی $E = 1$ باشد مدار غیرفعال است و به دو ورودی دیگر بستگی ندارد. هنگام غیرفعال شدن مدار، هیچ یک از خروجی ها در 0 نبوده و هیچ یک از مینترم ها انتخاب نمی شوند. به طور کلی، یک دیکدر ممکن است خروجی های متمم شده یا متمم نشده داشته باشد. ورودی فعالساز ممکن است با سیگنال 0 یا 1 فعال گردد. بعضی از دیکدرها دارای دو یا چند ورودی فعالساز می باشند که باید یک شرط منطقی مفروضی را برآورده سازند تا مدار فعال شود.

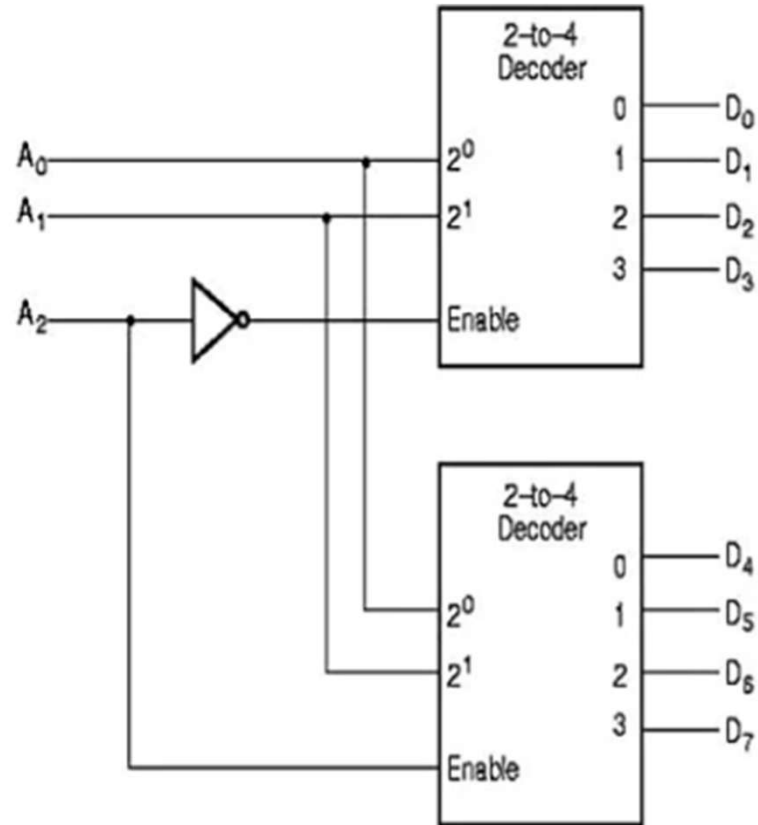


E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



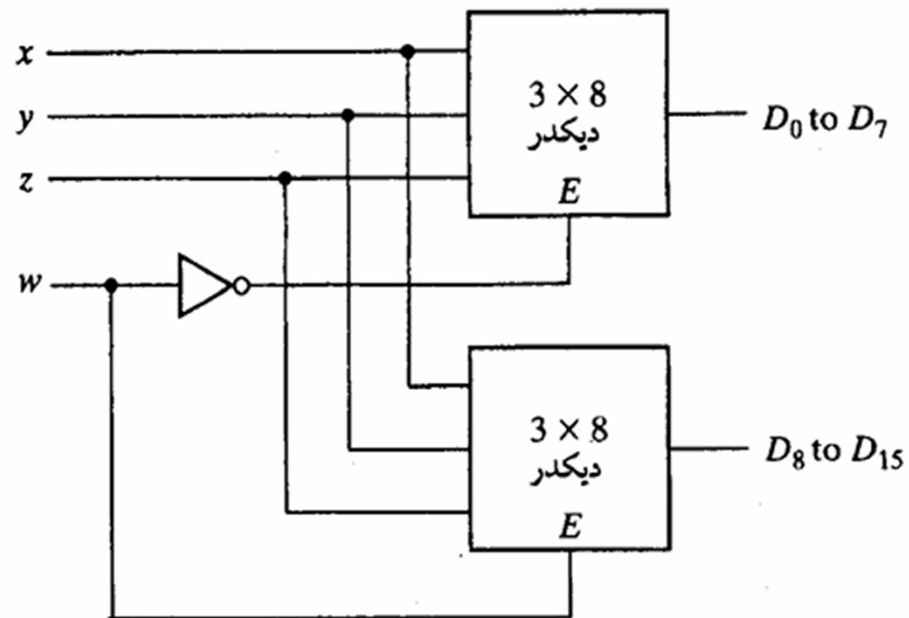
دانشگاه سمنان
Semnan University
پردیس فیروزانگان

مثال) با استفاده از دو دیکدر 2×4 یک دیکدر 3×8 بسازید.

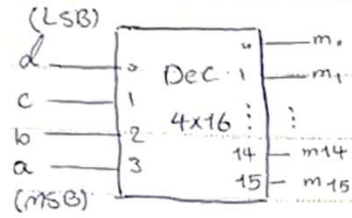




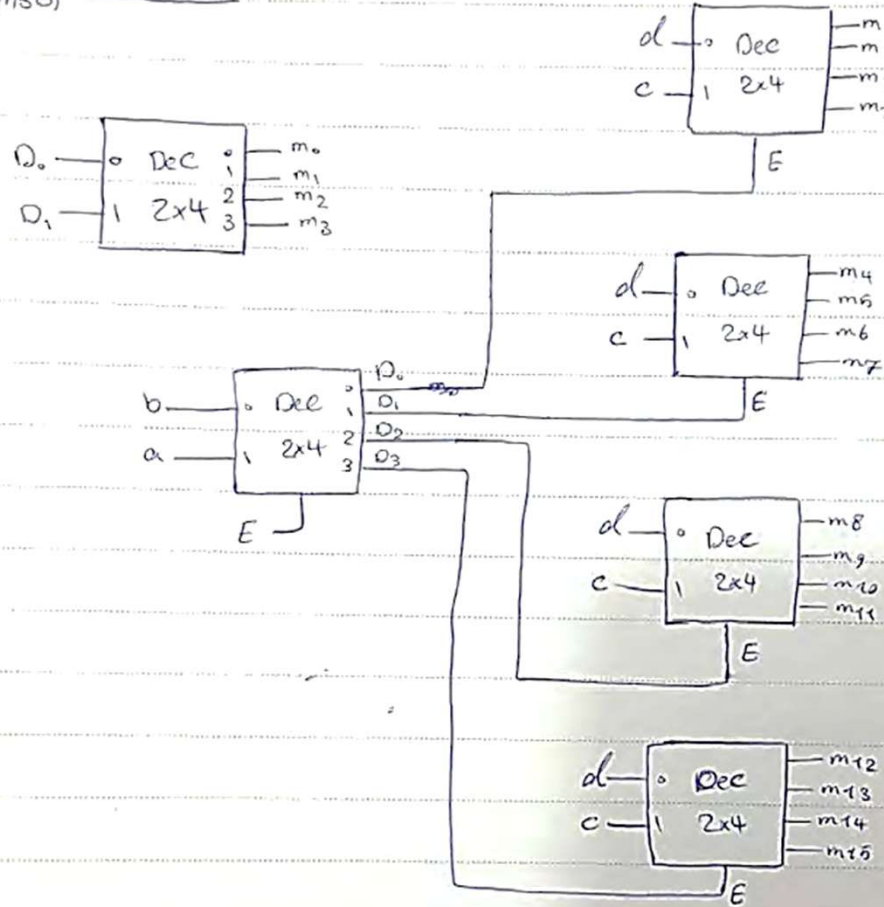
مثال) با استفاده از دو دیکدر 2×4 یک دیکدر 3×8 بسازید.



شکل ۲-۴. دیکدر 4×16 ساخته شده با دو دیکدر 3×8



با استفاده از دیکد ۲×۴ یک دیکد ۴×۱۶ بسازید.





۴-۹ انکدرها

یک انکدر مداری است که عمل عکس یک دیکدر را انجام می‌دهد. یک انکدر دارای 2^n (یا کمتر) خط ورودی و n خط خروجی است. خطوط خروجی کد دودویی مربوط به مقدار دودویی ورودی را تولید می‌نمایند.

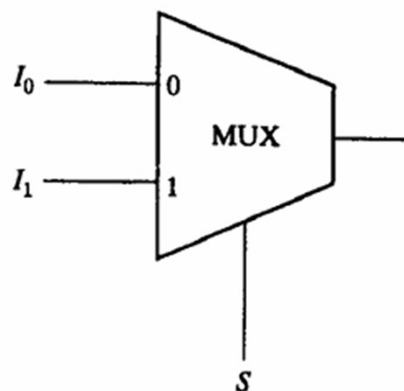
مثالی از یک انکدر، انکدر هشت هشتی به دودویی است که جدول درستی آن در جدول ۴-۷ داده شده است. این مدار دارای هشت ورودی (یک ورودی برای هر رقم هشت هشتی) و سه خروجی است که عدد دودویی مربوطه را تولید می‌نماید. فرض بر این است که در هر لحظه‌ای از زمان تنها یک ورودی مقدار 1 را داشته باشد.

ورودی‌ها								خروجی‌ها		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



۱۰-۴ مولتی پلکسرها

یک مولتی پلکسر مداری ترکیبی است که اطلاعات دودویی را از تعدادی خط ورودی دریافت کرده و آنها را به یک خط خروجی هدایت می‌نماید. انتخاب یک ورودی خاص به وسیله مجموعه‌ای از خطوط انتخاب انجام می‌شود. معمولاً 2^n خط ورودی و n خط انتخاب وجود دارد و ترکیب بیتی تعیین کننده ورودی انتخاب شده است.



یک مولتی پلکسر 2 به 1 یکی از دو منبع 1 بیت را طبق شکل ۲۴-۴ به یک مقصد مشترک متصل می‌کند. مدار دارای دو خط ورودی داده، یک خروجی و یک خط انتخاب S است. وقتی $S = 0$ باشد، گیت AND فوقانی فعال شده و I_0 به خروجی راه می‌یابد. وقتی $S = 1$ باشد، گیت AND تحتانی فعال شده و I_1 به خروجی متصل می‌شود. مولتی پلکسر مثل یک کلید الکترونیک عمل کرده و یکی از دو منبع را انتخاب می‌نماید. نمودار بلوکی یک مولتی پلکسر گاهی به شکل دوزنقه شکل ۲۴-۴ (ب) نشان داده می‌شود. این مدار چگونگی انتخاب و هدایت منابع متعدد داده را به یک مقصد نشان می‌دهد. مولتی پلکسر اغلب با نمودارهای بلوکی و کلمه MUX نشان داده می‌شود.



پیاده‌سازی تابع بول

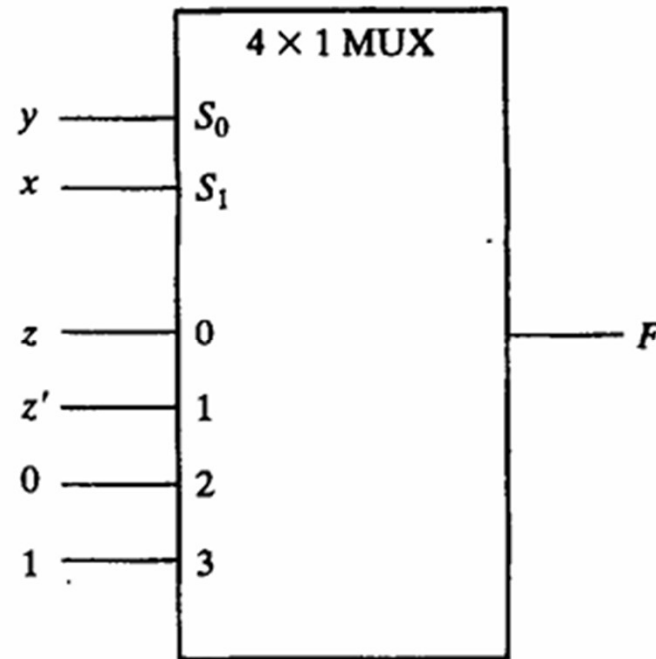
ابتدا $n-1$ متغیر به ورودی‌های انتخاب مولتی پلکسر وصل می‌شود. تنها متغیر باقیمانده تابع برای ورودی‌های داده مورد استفاده قرار می‌گیرد. اگر متغیر باقیمانده را z بنامیم هر ورودی داده مولتی پلکسر برابر z ، z' ، 1 و 0 خواهد بود. برای نمایش این رویه، تابع سه متغیره زیر را ملاحظه کنید:

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

تابع را می‌توان مطابق شکل ۲۷-۴ با یک مولتی پلکسر ۴ به ۱ پیاده‌سازی کرد. دو متغیر x و y به خطوط انتخاب وصل می‌شود، به این ترتیب که x به ورودی s_1 و y به ورودی s_0 متصل می‌گردد. مقادیر خطوط ورودی داده از جدول درستی تابع معین می‌شود. وقتی $xy = 00$ باشد خروجی F برابر z است زیرا وقتی $z = 0$ است F هم برابر 0 می‌باشد و وقتی $z = 1$ شود F نیز برابر 1 می‌گردد. این وضع لازم می‌دارد تا متغیر z به ورودی داده 0 متصل شود. عملکرد مولتی پلکسر به نحوی است که وقتی $xy = 00$ گردد، خط داده شماره 0 به خروجی وصل شده F را برابر z می‌نماید. به طریقی مشابه می‌توان نشان داد که خطوط ورودی 1 ، 2 و 3 وقتی که $xy = 01$ ، 10 و 11 است به ترتیب به F وصل می‌شوند. این مثال خاص هر چهار حالت ممکن را برای ورودی‌های داده تهیه می‌کند.



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



(ب) پیاده سازی مولتی پلکسر

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$F = z$
 $F = z'$
 $F = 0$
 $F = 1$

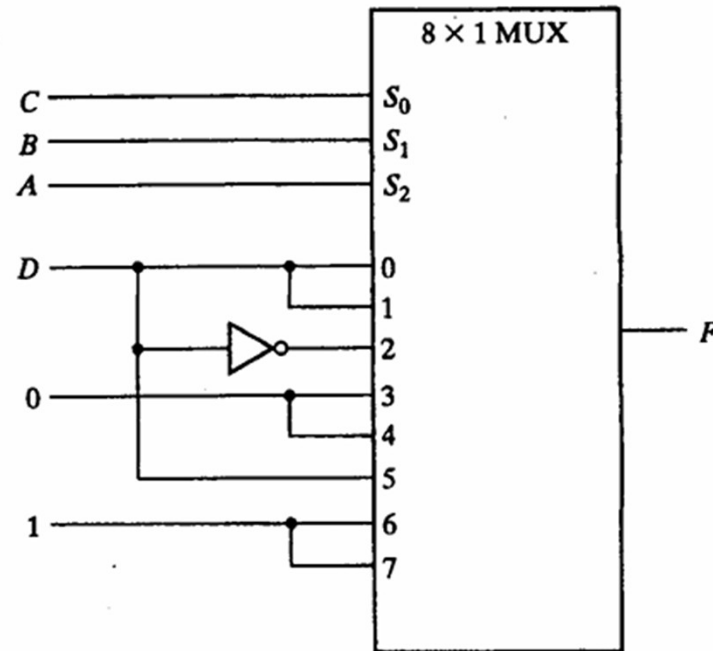
(الف) جدول درستی



$$F(A, B, C, D) = \sum(1, 3, 4, 11, 12, 13, 14, 15).$$

این تابع با مولتی پلکسری که سه ورودی انتخاب دارد، مطابق شکل ۴-۲۸ پیاده‌سازی می‌شود. توجه کنید که اولین متغیر A باید به ورودی انتخاب S_2 و دو خط B و C به ترتیب به S_1 و S_0 وصل شوند.

A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	



شکل ۴-۲۸. پیاده‌سازی یک تابع ۴ ورودی با یک مولتی پلکسر



با استفاده از 4×2 ماکس یک 8×1 بسازید.

