مبحث چهارم

منطق ترکیبی

Presented by A. Maleki Fall Semester, 2021

فهرست مطالب

منطق ترکیبی (Combinatorial Logic)

- مدارهای منطقی
  - مدارهای ترتیبی
    - ثبات
    - فلیپ فلاپ
    - شمارنده
  - مدارهای ترکیبی
    - دیکدر
    - مالتی پلکسر
    - جمع کننده

مدار ترکیبی: مداری شامل گیت‌های منطقی که خروجی‌های آن تنها به ورودی‌ها در همان زمان وابسته است.

مدار ترتیبی: مداری شامل گیت‌های منطقی و عناصر حافظه که خروجی آن علاوه بر ورودی‌های فعلی به حالت مدار (ورودی‌های قبلی) نیز وابسته است.

واژه نامه:

مدار منطقی ترکیبی: Combinational Logic, Combinatorial Logic

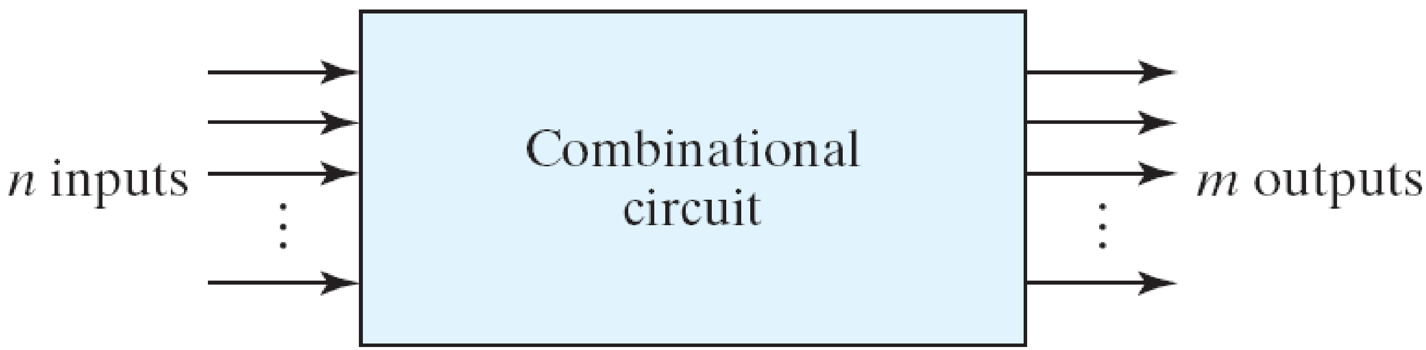
مدار منطقی ترتیبی: Sequential Logic

شیوه‌های مختلف توصیف مدار منطقی ترکیبی

- شماتیک مداری
- جدول درستی
- توابع بولی

شیوه‌های مختلف توصیف مدار منطقی ترکیبی

- شماتیک مداری
- جدول درستی
- توابع بولی



شیوه‌های مختلف توصیف مدار منطقی ترکیبی

- شماتیک مداری
- **جدول درستی**
- توابع بولی

جدولی شامل 2^n ترکیب مختلف ورودی

و m ستون برای متغیرهای خروجی

شیوه‌های مختلف توصیف مدار منطقی ترکیبی

- شماتیک مداری
- جدول درستی
- **توابع بولی**

m تابع بولی که هر یک تابعی از n متغیر ورودی است.

فهرست مطالب

- مقدمه‌ای بر مدارهای ترکیبی
- روش تحلیل و روش طراحی
- جمع‌کننده – تفریق‌گر دودویی
- جمع‌کننده‌ی دهدهی
- ضرب‌کننده‌ی دودویی
- مقایسه‌گر اندازه
- دیکدرها
- انکدرها
- مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- گیت سه حالته

تحلیل مدارهای ترکیبی

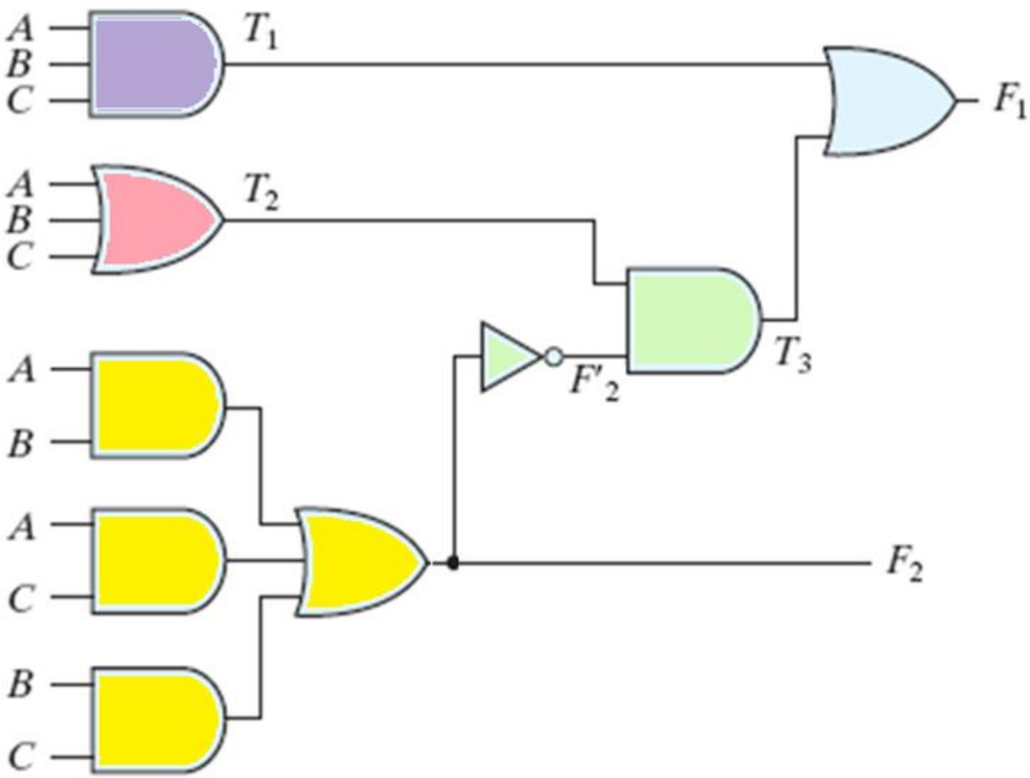
- مشخص کردن تابعی که مدار منطقی ترکیبی پیاده‌سازی می‌کند.
- نتیجه می‌تواند جدول درستی یا تابع‌های بولی باشد.

مثال

مدار منطقی شماتیک شکل زیر را تحلیل نموده و

الف: تابع‌های بولی F_1 و F_2 را به دست آورید.

ب : جدول درستی مدار را به دست آورید.

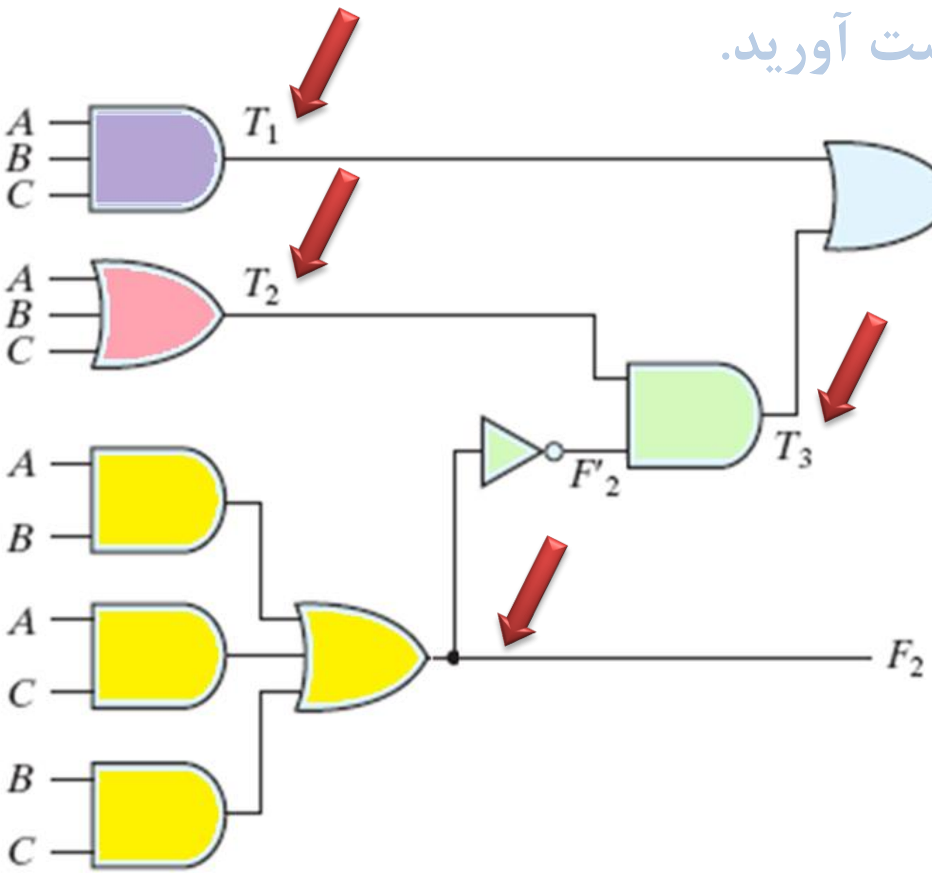


مثال

مدار منطقی شماتیک شکل زیر را تحلیل نموده و

الف: تابع‌های بولی F_1 و F_2 را به دست آورید.

ب : جدول درستی مدار را به دست آورید.



$T1 = ABC$

$T2 = A + B + C$

$F2 = AB + AC + BC$

$T3 = T2F2'$

$= (A + B + C)(AB + AC + BC)'$

$F1 = T1 + T3$

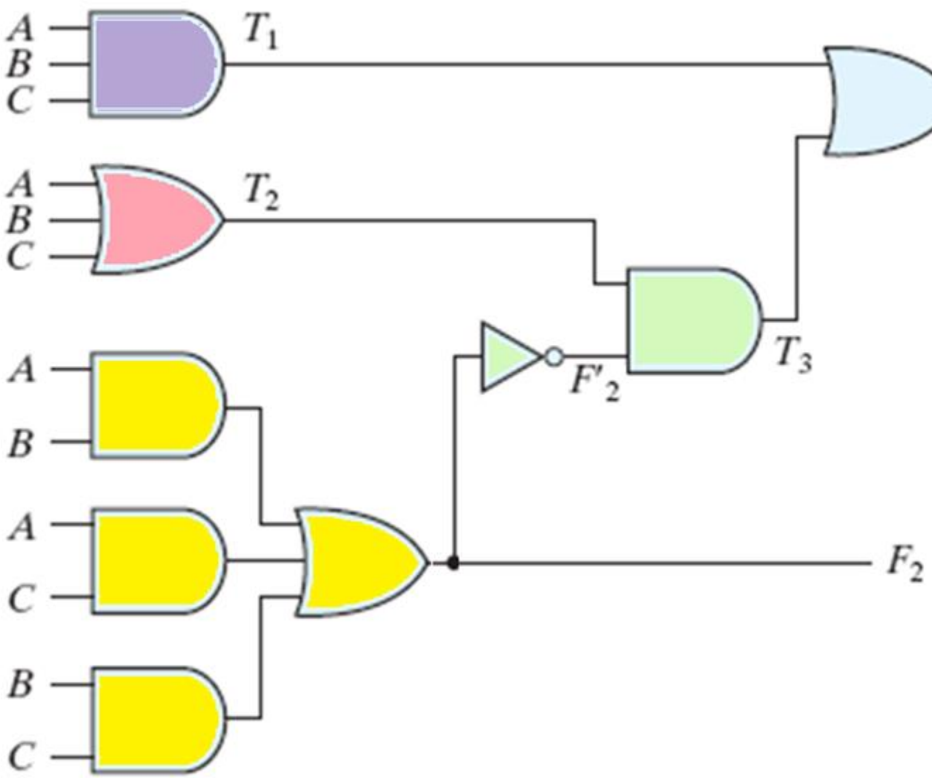
$= ABC + (A + B + C)(AB + AC + BC)'$

مثال

مدار منطقی شماتیک شکل زیر را تحلیل نموده و

الف: تابع‌های بولی F_1 و F_2 را به دست آورید.

ب : جدول درستی مدار را به دست آورید.



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

طراحی مدارهای ترکیبی

روش طراحی:

۱- تعیین ورودی‌ها و خروجی‌ها بر اساس مشخصات مدار

(n ورودی و m خروجی)

۲- تشکیل جدول درستی بر اساس مشخصات مدار

(2^n ترکیب ورودی و m ستون خروجی)

۳- ساده‌سازی و به دست آوردن عبارت‌های منطقی

۴- ترسیم نمودار شماتیک

مثال

مداری جهت تبدیل کد BCD به کد افزونی۳ (Excess-3) طراحی نمایید.

مرحله اول: تعیین ورودی‌ها و خروجی‌ها بر اساس مشخصات مدار

متغیرهای ورودی: چهار متغیر A ، B ، C و D

متغیرهای خروجی: چهار متغیر w ، x ، y و z

مثال

مداری جهت تبدیل کد BCD به کد افزونی۳ (Excess-3) طراحی نمایید.

مرحله دوم: تشکیل جدول درستی بر اساس مشخصات مدار

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

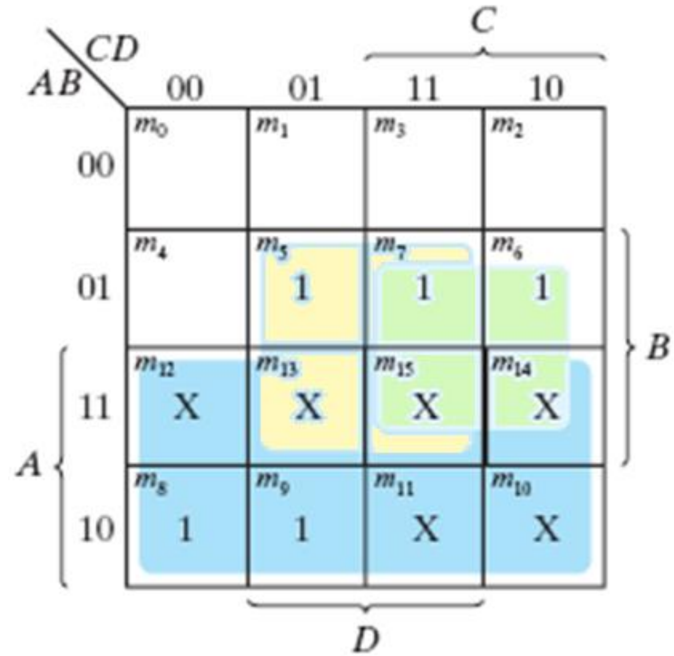
مثال

مداری جهت تبدیل کد BCD به کد افزونی۳ (Excess-3) طراحی نمایید.

مرحله سوم: ساده‌سازی و به دست آوردن تابع‌های بولی

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



w = A + BC + BD

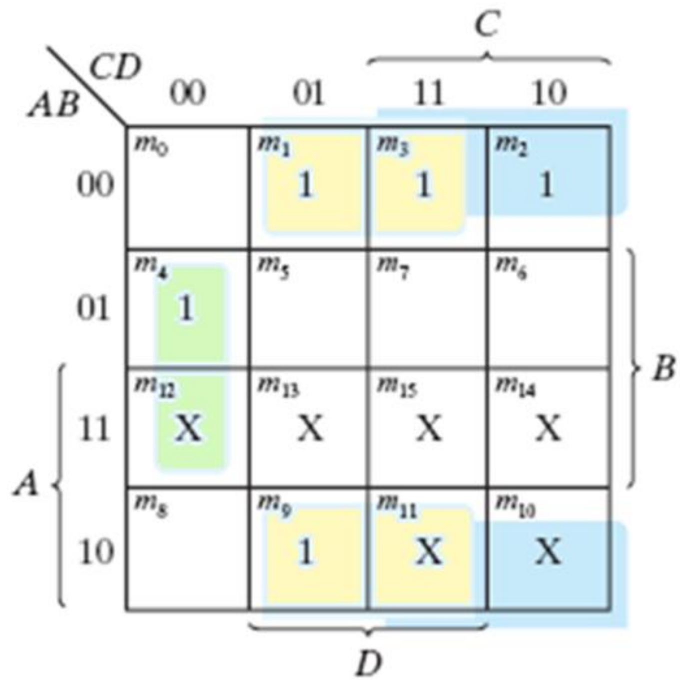
مثال

مداری جهت تبدیل کد BCD به کد افزونی۳ (Excess-3) طراحی نمایید.

مرحله سوم: ساده‌سازی و به دست آوردن تابع‌های بولی

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	**B**	**C**	**D**	**w**	**x**	**y**	**z**
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



$w = A + BC + BD$

$x = B'C + B'D + BC'D'$

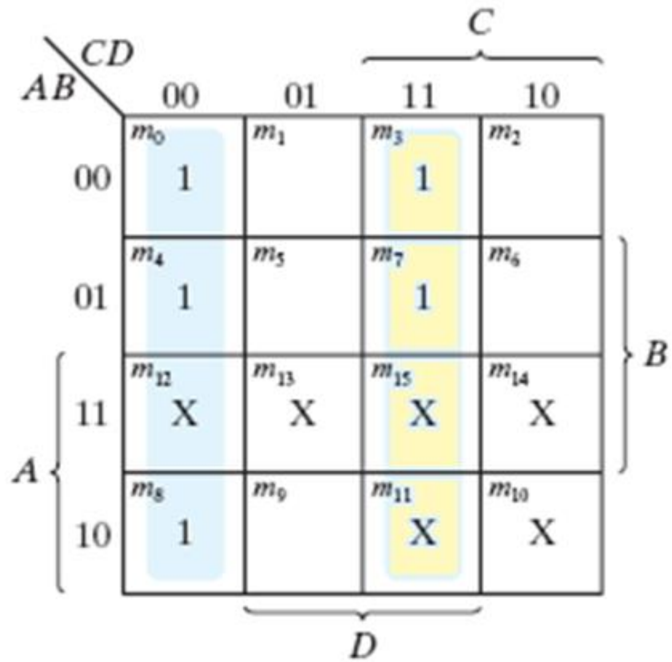
مثال

مداری جهت تبدیل کد BCD به کد افزونی۳ (Excess-3) طراحی نمایید.

مرحله سوم: ساده‌سازی و به دست آوردن تابع‌های بولی

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



w = A + BC + BD

x = B'C + B'D + BC'D'

y = CD + C'D'

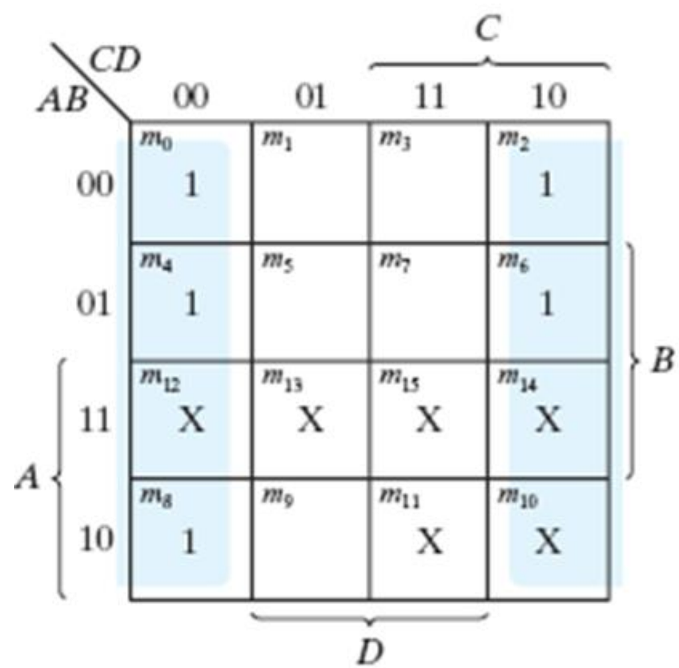
مثال

مداری جهت تبدیل کد BCD به کد افزونی۳ (Excess-3) طراحی نمایید.

مرحله سوم: ساده‌سازی و به دست آوردن تابع‌های بولی

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



w = A + BC + BD

x = B’C + B’D + BC’D’

y = CD + C’D’

z = D’

مثال

مداری جهت تبدیل کد BCD به کد افزونی ۳ (Excess-3) طراحی نمایید.

مرحله چهارم: ترسیم نمودار شماتیک

در سیستم‌های چندخروجی، ممکن است با دستکاری‌های جبری بتوان به پیاده‌سازی های مناسب‌تری دست یافت.

w = A + BC + BD

= A + B(C + D)

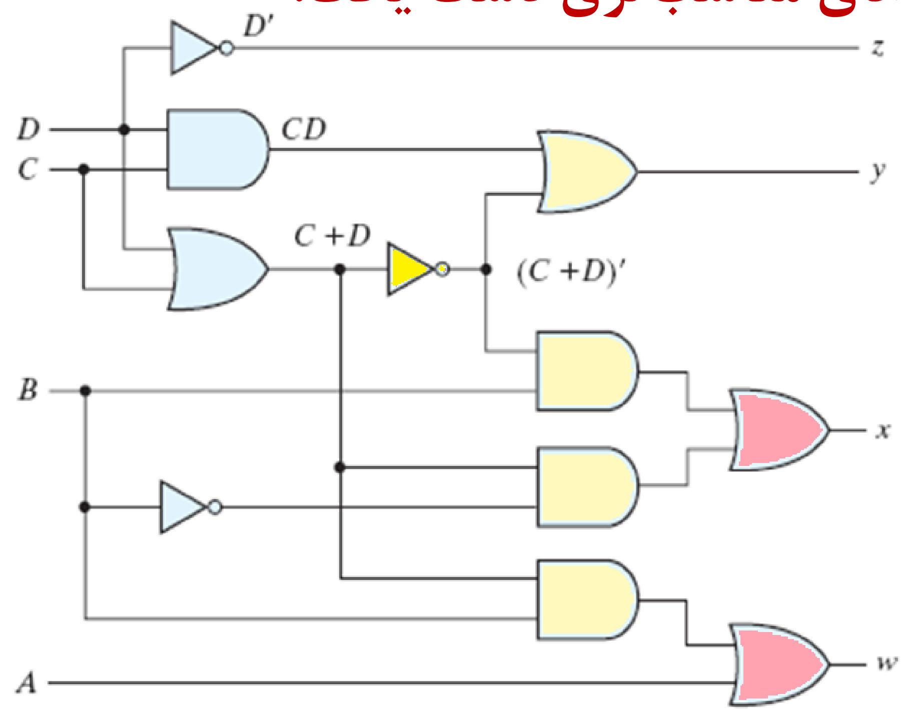
x = B’C + B’D + BC’D’

= B’(C + D) + B(C + D)’

y = CD + C’D’

= CD + (C + D)’

z = D’



این طرح نیاز به ۸ گیت دو ورودی و یک گیت Not دارد. ولی طرح دیگر دوسطحی نیست.

فهرست مطالب

- مقدمه‌ای بر مدارهای ترکیبی
- روش تحلیل و روش طراحی
- جمع‌کننده – تفریق‌گر دودویی
- جمع‌کننده‌ی دهدهی
- ضرب‌کننده‌ی دودویی
- مقایسه‌گر اندازه
- دیکدرها
- انکدرها
- مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- گیت سه حالته

جمع‌کننده- تفریق‌گر دودویی

- جمع دو رقم باینری
- جمع دو عدد باینری
- نیم جمع‌کننده
- تمام جمع‌کننده

0 + 0 = 0 ,   0 + 1 = 1 ,

1 + 0 = 1 ,   1 + 1 = 10   Output Carry

1 01 + 101 = ?

a 3-bit adder uses 3 one-bit adder

for second digit, there is input carry

مداری که دو بیت را به عنوان ورودی پذیرفته و
حاصل‌جمع و نقلی خروجی را تولید می‌نماید.

مداری که دو بیت را به همراه نقلی ورودی با هم جمع می‌کند.
تمام‌جمع‌کننده را می‌توان با دو نیم‌جمع‌کننده ایجاد نمود.

نیم جمع کننده (Half Adder)

طراحی نیم‌جمع‌کننده:

۱- تعیین ورودی‌ها و خروجی‌ها

۲- تشکیل جدول درستی

۳- ساده‌سازی و به دست آوردن تابع‌های منطقی

۴- ترسیم نمودار شماتیک

Two inputs: x,y

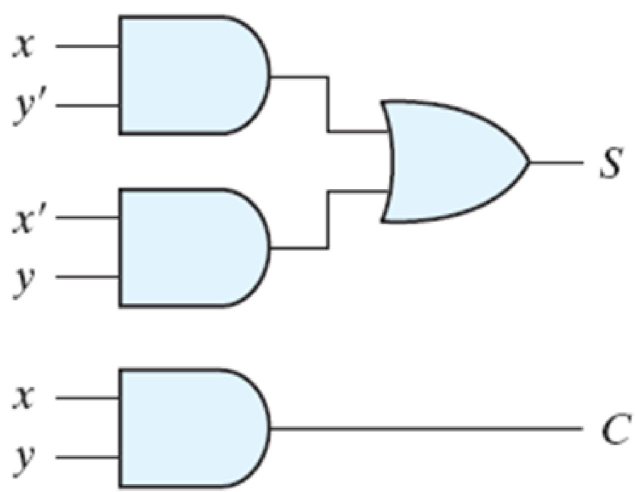
Two Outputs: S (sum), C (carry)

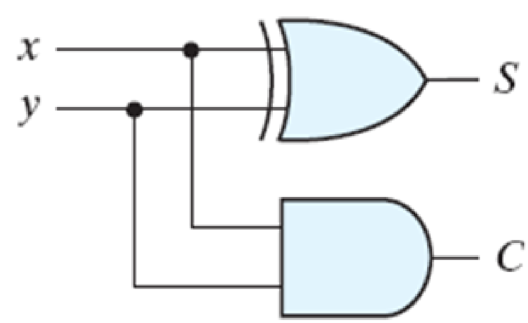
Half Adder

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

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

$C = xy$





تمام جمع‌کننده (Full Adder)

Three inputs: x,y,z

Two Outputs: S (sum), C (carry)

طراحی تمام جمع‌کننده:

۱– تعیین ورودی‌ها و خروجی‌ها

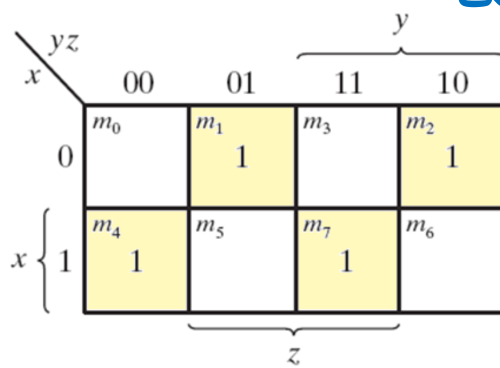
۲– تشکیل جدول درستی

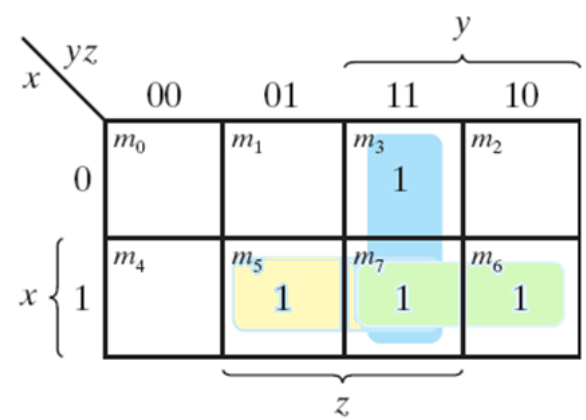
۳– ساده‌سازی و به دست آوردن تابع‌های منطقی

۴– ترسیم نمودار شماتیک

Full Adder

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





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

$= x \oplus y \oplus z$

$C = xy + xz + yz = xy + z(x \oplus y)$

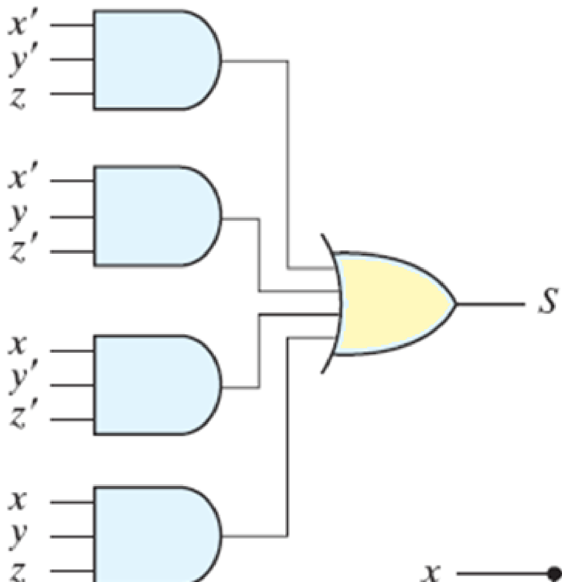
تمام جمع کننده (Full Adder)

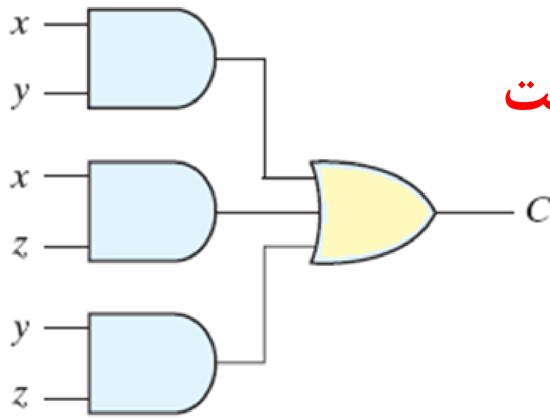
$S = x'y'z + x'yz' + xy'z' + xyz = x \oplus y \oplus z$

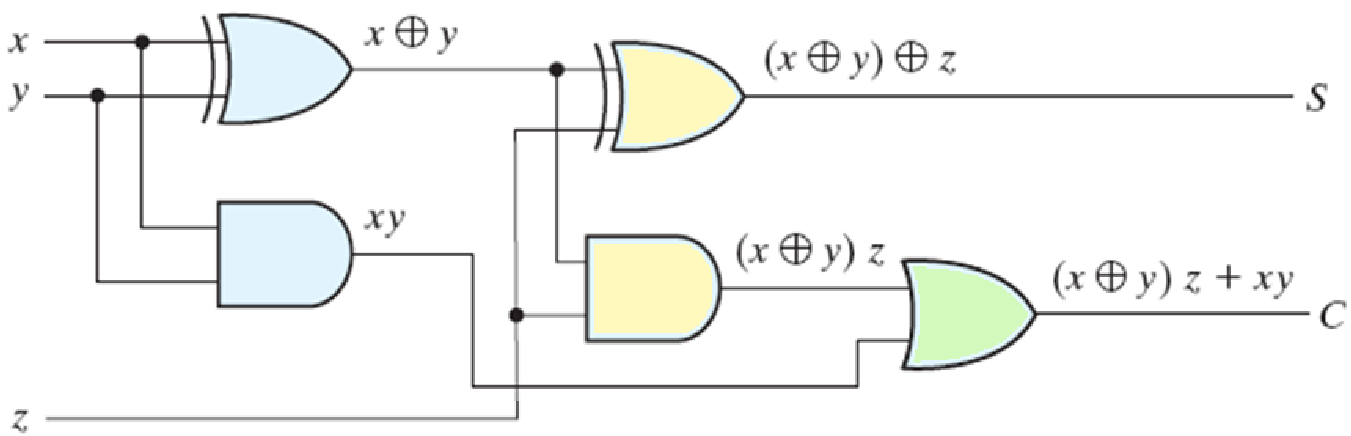
$C = xy + xz + yz = xy + z(x \oplus y)$

پیاده‌سازی دو سطحی، ۹ گیت

پیاده‌سازی سه سطحی، ۵ گیت







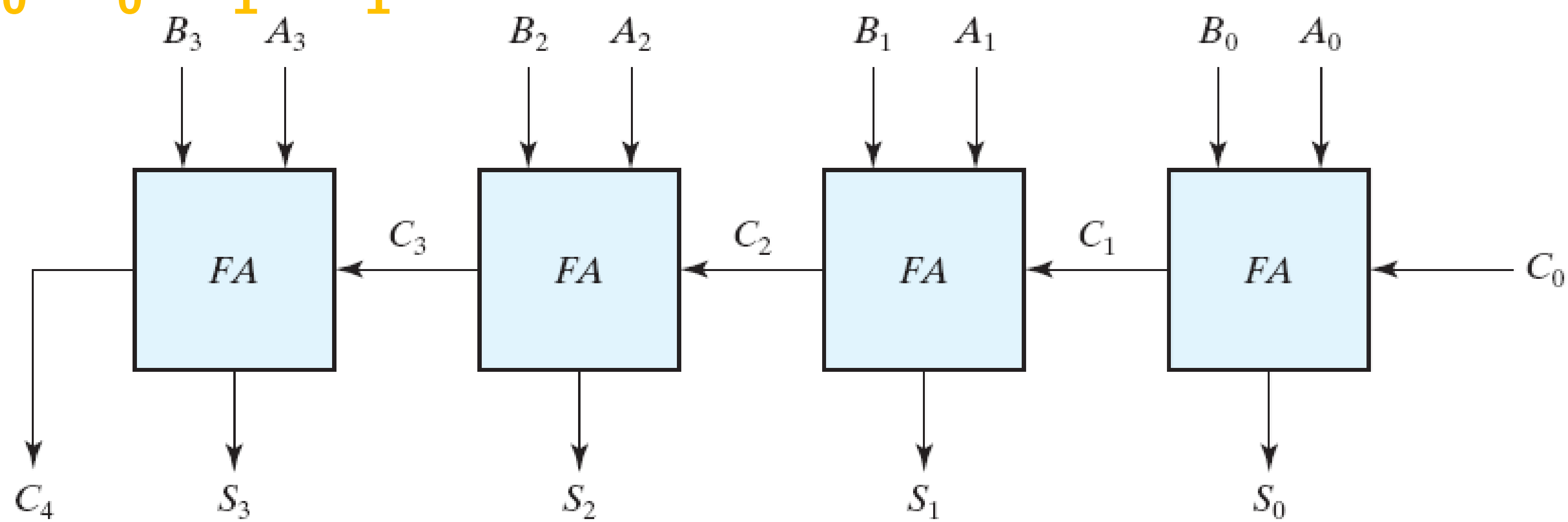
جمع‌کننده‌ی دودویی

- جمع‌کننده‌ی دودویی یک مدار منطقی است که دو عدد را با هم جمع می‌کند.
- جمع‌کننده‌ی دودویی n بیتی را می‌توان با n تمام جمع‌کننده پیاده‌سازی نمود.

for example: 1011 + 0011

C_i	0	1	1	0
A_i	1	0	1	1
B_i	0	0	1	1
S_i	1	1	1	0
C_{i+1}	0	0	1	1

جمع‌کننده‌ی دودویی ۴بیتی با نقلی موج‌گونه (rippe)



توجه: برای طراحی به شیوه‌ی کلاسیک نیاز به ارزیابی ۵۱۲= $۲^۹$ ترکیب ورودی است.

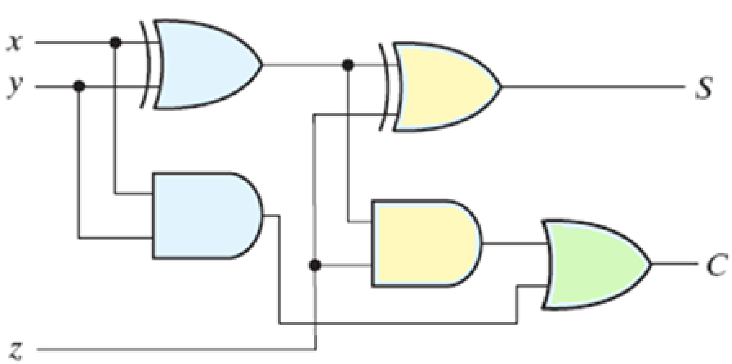
مصالحه در طراحی جمع‌کننده‌ی دودویی با نقلی موج‌گونه:

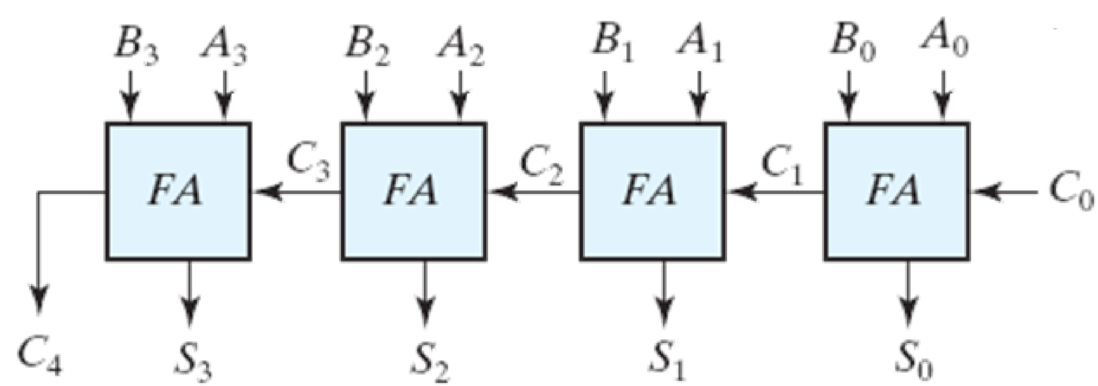
مزیت: سادگی طراحی (رهایی از جدول درستی با ۵۱۲ ترکیب ورودی)

ایراد: به واسطه‌ی اتصال موج‌گونه‌ی رقم نقلی، زمان انتشار سیگنال در این مدار، ۲+۲+۲+۳ برابر زمان انتشار سیگنال در یک گیت است.

راهکار: ۱- استفاده از گیت‌های سریع‌تر

۲- افزایش پیچیدگی مدار با هدف کاهش زمان تاخیر انتشار رقم نقلی





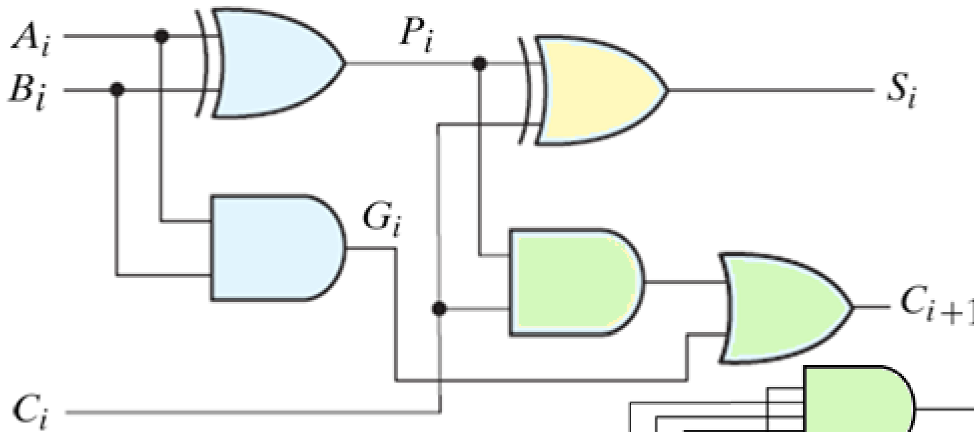
تکنیک پیش‌بینی نقلی (Carry Lookahead)

$P_i = A_i \oplus B_i$

$G_i = A_i B_i$

$S_i = P_i \oplus C_i$

$C_{i+1} = G_i + P_i C_i$



$C_1 = G_0 + P_0 C_0$

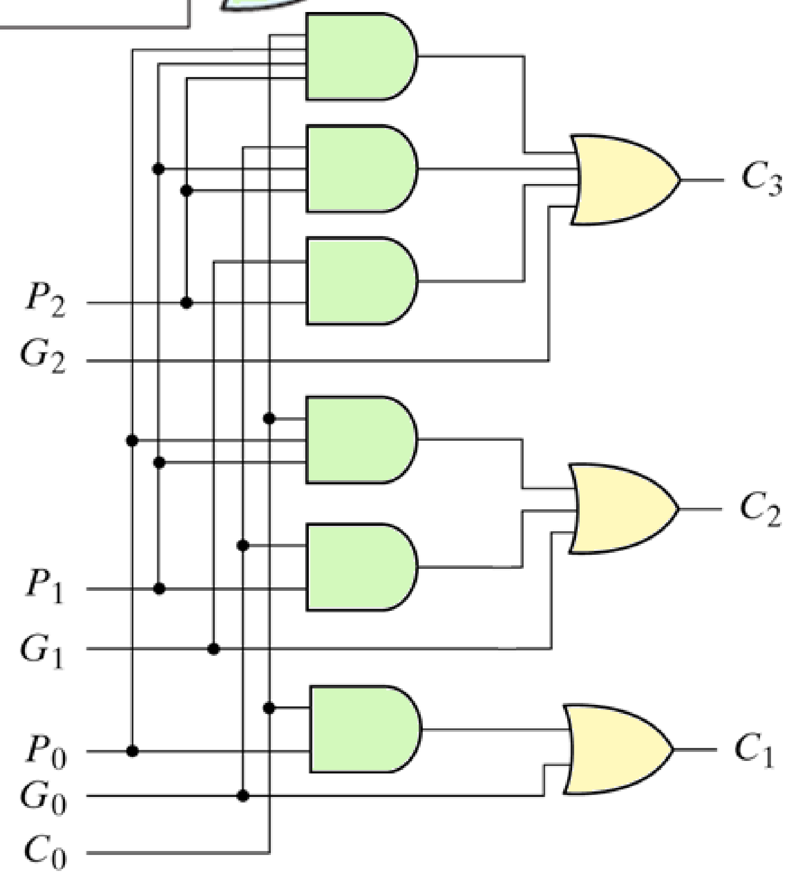
$C_2 = G_1 + P_1 C_1$

$\quad = G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = G_2 + P_2 C_2$

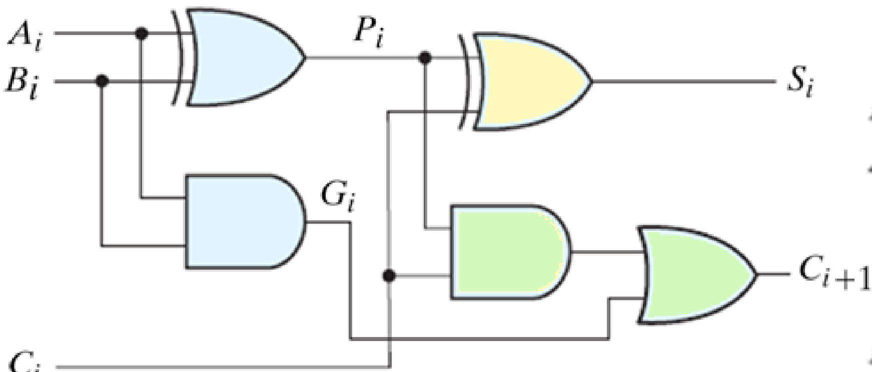
$\quad = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

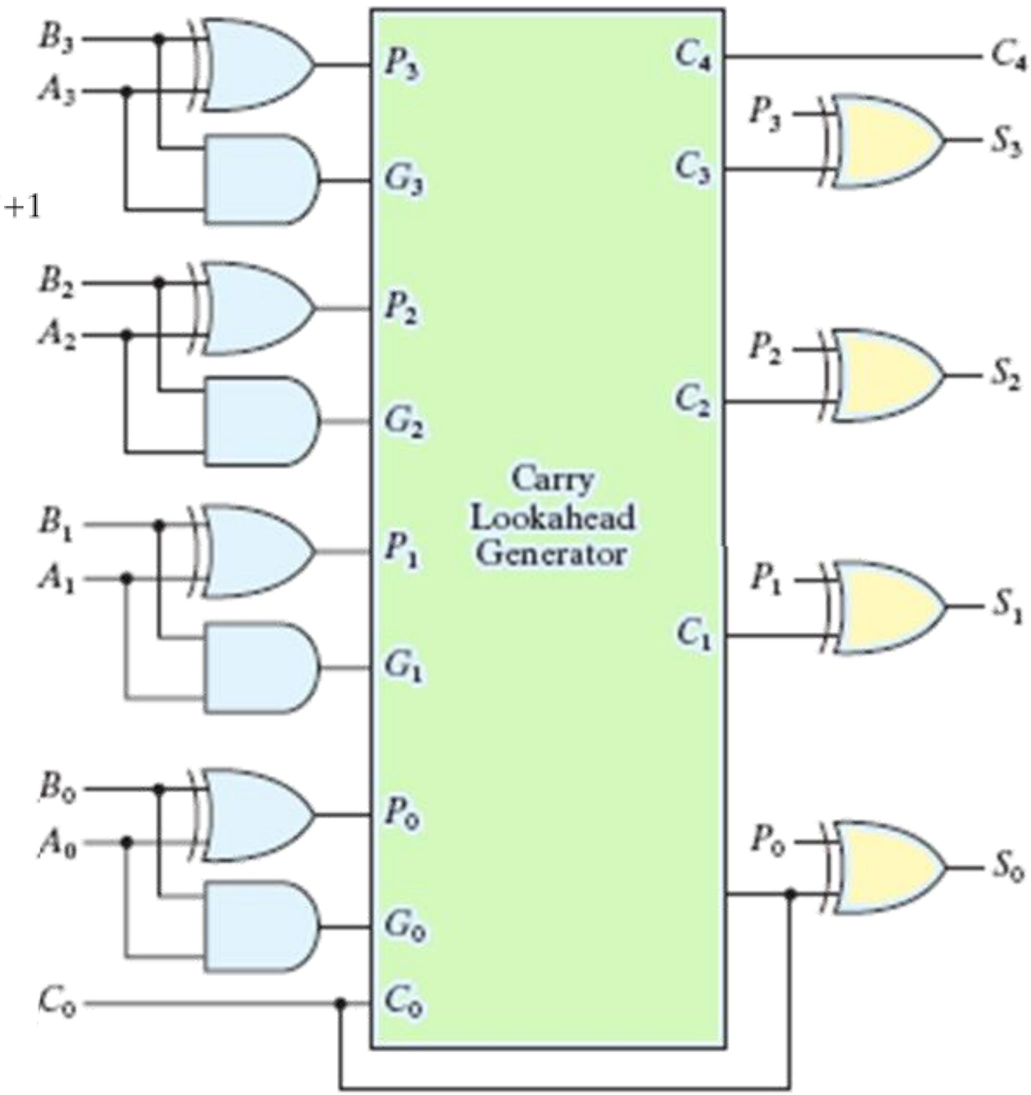
$C_4 = \ldots$



افزایش سرعت در قبال پیچیدگی سخت‌افزار

جمع‌کننده‌ی ۴-بیتی با تکنیک پیش‌بینی نقلی





$P_i = A_i \oplus B_i$

$G_i = A_i B_i$

$S_i = P_i \oplus C_i$

$C_{i+1} = G_i + P_i C_i$

توجه کنید:

این مدار ۴ لایه است.

تفریق دودویی

برای تفریق کردن دو عدد،

می‌توان عدد اول را با مکمل ِ ۲ی عدد دوم جمع کرد.

برای دستیابی به مکملِ ۲ی عدد،

می‌توان تمام بیت‌های عدد را مکمل نموده و نتیجه را با ۱ جمع نمود.

برای دستیابی به مدار تفریق گر،

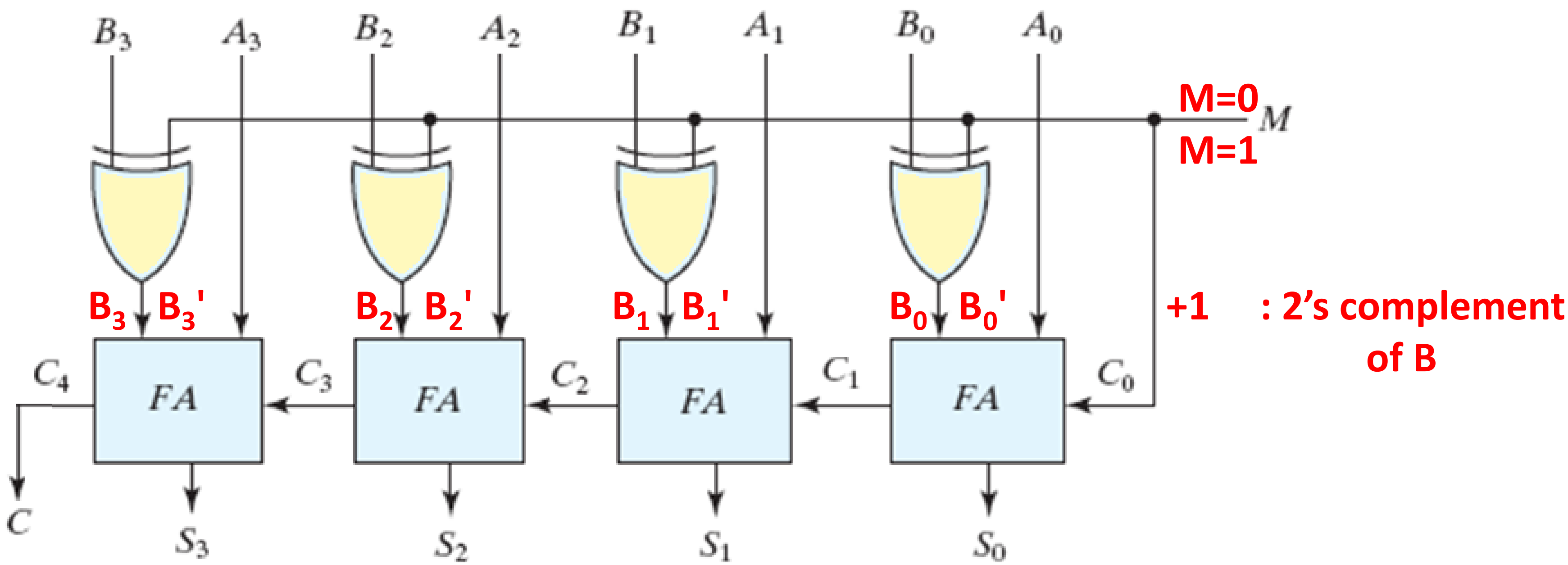
می‌توان از مدار جمع‌کننده استفاده نموده و عدد اول را با مکمل بیتی (مکمل ۱) عدد دوم و نقلی ورودی ۱ جمع نمود.

یادآوری:

XOR را می‌توان یک NOT کنترل شده با یک ورودی کنترل در نظر گرفت.

جمع کننده-تفریق گر دودویی (adder-subtractor)

M: Mode (Control Signal)



S = sum of A & B
S = subtract of A & B

M=0, Adder
M=1, Subtractor

سرریز (overflow)

مفهوم سرریز :

شرایطی که دو عدد n رقمی با هم جمع می‌شوند ولی برای نمایش خروجی به n+1 رقم نیاز است.

سرریز در جمع‌کردن دو عدد بدون‌علامت:

وجود نقلی انتهایی (end carry)

سرریز در جمع‌کردن دو عدد علامت‌دار (مکمل ۲-علامت):

۱- اگر یکی از اعداد مثبت و دیگری منفی باشد سرریز رخ نمی‌دهد.
۲-اگر هر دو عدد مثبت یا هر دو منفی باشند امکان سرریز وجود دارد.

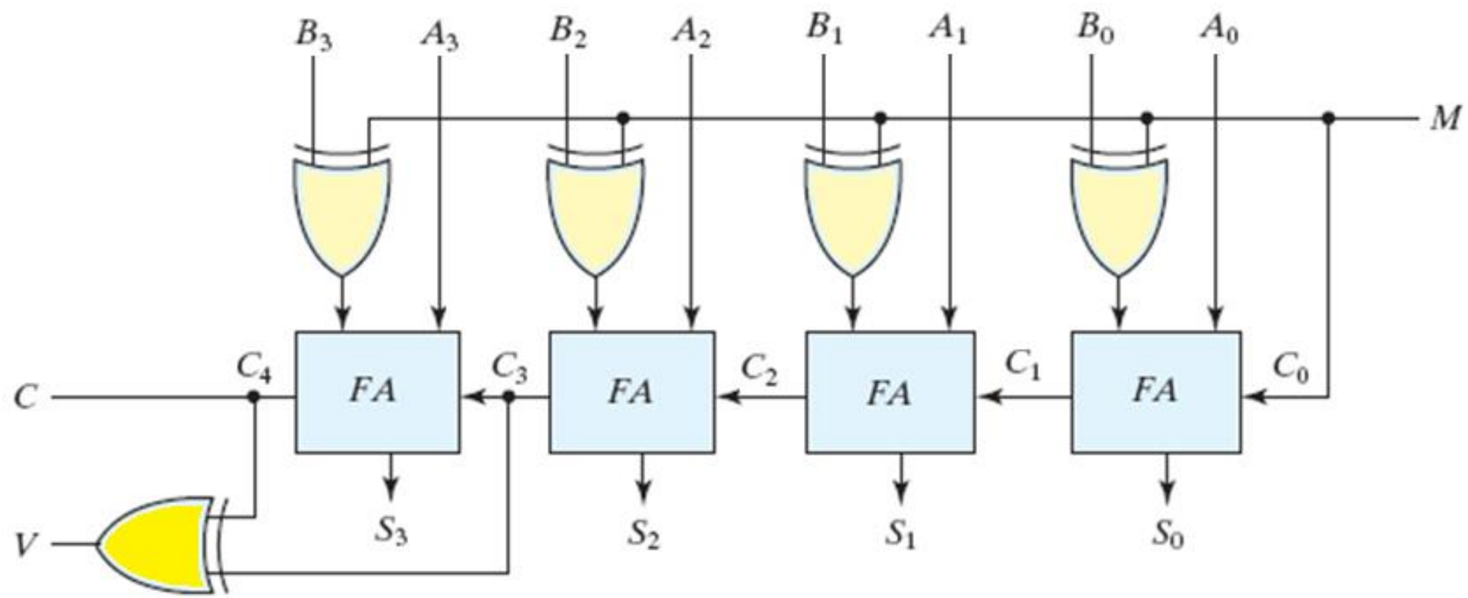
مثال:

(+65):	01000001	(-65):	10111111	(+65):	01000001
+(-65):	+10111111	+(-65):	+10111111	+(+65):	+ 01000001

بنابراین هنگام جمع کردن دو عدد علامت دار مکمل۲-علامت،
یکسان نبودن نقلی بیت علامت و نقلی انتهایی نشانگر سرریز است.

سرریز

- تشخیص سرریز در جمع‌کننده- تفریق‌گر دودویی



فهرست مطالب

- مقدمه‌ای بر مدارهای ترکیبی
- روش تحلیل و روش طراحی
- جمع‌کننده – تفریق‌گر دودویی
- جمع‌کننده‌ی دهدهی
- ضرب‌کننده‌ی دودویی
- مقایسه‌گر اندازه
- دیکدرها
- انکدرها
- مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- گیت سه حالته

جمع‌کننده‌ی BCD

جمع‌کننده‌ی BCD مداری است که دو رقم دهدهی BCD را به همراه رقم نقلی ورودی با هم جمع نموده و حاصل‌جمع را به صورت BCD فراهم می‌سازد.

ورودی‌ها: دو رقم دهدهی BCD که هرکدام با ۴ بیت مشخص می‌گردند و رقم نقلی ورودی که یک بیتی است.

ورودی‌های BCD دارای مقداری در محدوده‌ی 0000 تا 1001 می‌باشند.

رقم نقلی ورودی می‌تواند 0 یا 1 باشد.

خروجی‌ها: حاصل‌جمع دو رقم دهدهی و نقلی ورودی در محدوده‌ی 0=0+0+0 تا 19=1+9+9 می‌باشد که با یک رقم دهدهی BCD و یک رقم نقلی خروجی قابل نمایش است.

یادآوری روش جمع کردن دو عدد BCD از مبحث نخست درس:

برای جمع‌کردن دو عدد BCD :

۱- رقم‌های BCD متناظر دو عدد را از کم ارزش‌ترین رقم با هم **جمع باینری** می‌کنیم.

۲- در صورتی که حاصل جمع بزرگتر از ۹ باشد نتیجه را اصلاح می‌کنیم.

(وجود نقلی خروجی یا به دست آمدن یکی از مقادیر نامعتبر BCD)

(تصحیح با **جمع باینری** نتیجه با **0110** انجام می‌گردد.)

۳- همین روند را برای رقم‌های BCD بعدی نیز پی می‌گیریم.

مثال: جمع BCD زیر را انجام دهید.

```
  197
+ 284
-----
  481
```

```
     1        1
   0001     1001     0111
+  0010     1000     0100
---------------------------
   0100   1 0010     1011
           + 0110   + 0110
           ------   ------
             1000     0001
```

مقدار نامعتبر BCD

وجود نقلی انتهایی

جمع‌کننده‌ی BCD

برای طراحی مدار جمع‌کننده‌ی BCD چه راهکاری را پیشنهاد می‌کنید؟

راهکار نخست:

طراحی مدار به شیوه‌ی طراحی کلاسیک

با توجه به تعداد ۱+۴+۴ ورودی مدار لازم است جدول درستی با $۲^۹$ حالت مختلف ورودی تشکیل گردد و ساده سازی ...

راهکار دوم:

استفاده از مدار جمع‌کننده‌ی باینری و تکمیل آن با مدار تصحیح خروجی

جمع‌کننده‌ی BCD

Five inputs: K, Z_8, Z_4, Z_2, Z_1

One Output: C

طراحی بخش تصحیح خروجی:

۱- تعیین ورودی‌ها و خروجی‌ها
۲- تشکیل جدول درستی
۳- ساده‌سازی
۴- ترسیم نمودار شماتیک

با ترسیم نقشه، ساده‌سازی را انجام دهید.

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

رهیافتی دیگر:

$C = K + Z_4Z_8 + Z_2Z_8$

جمع کننده‌ی BCD

طراحی بخش تصحیح خروجی:

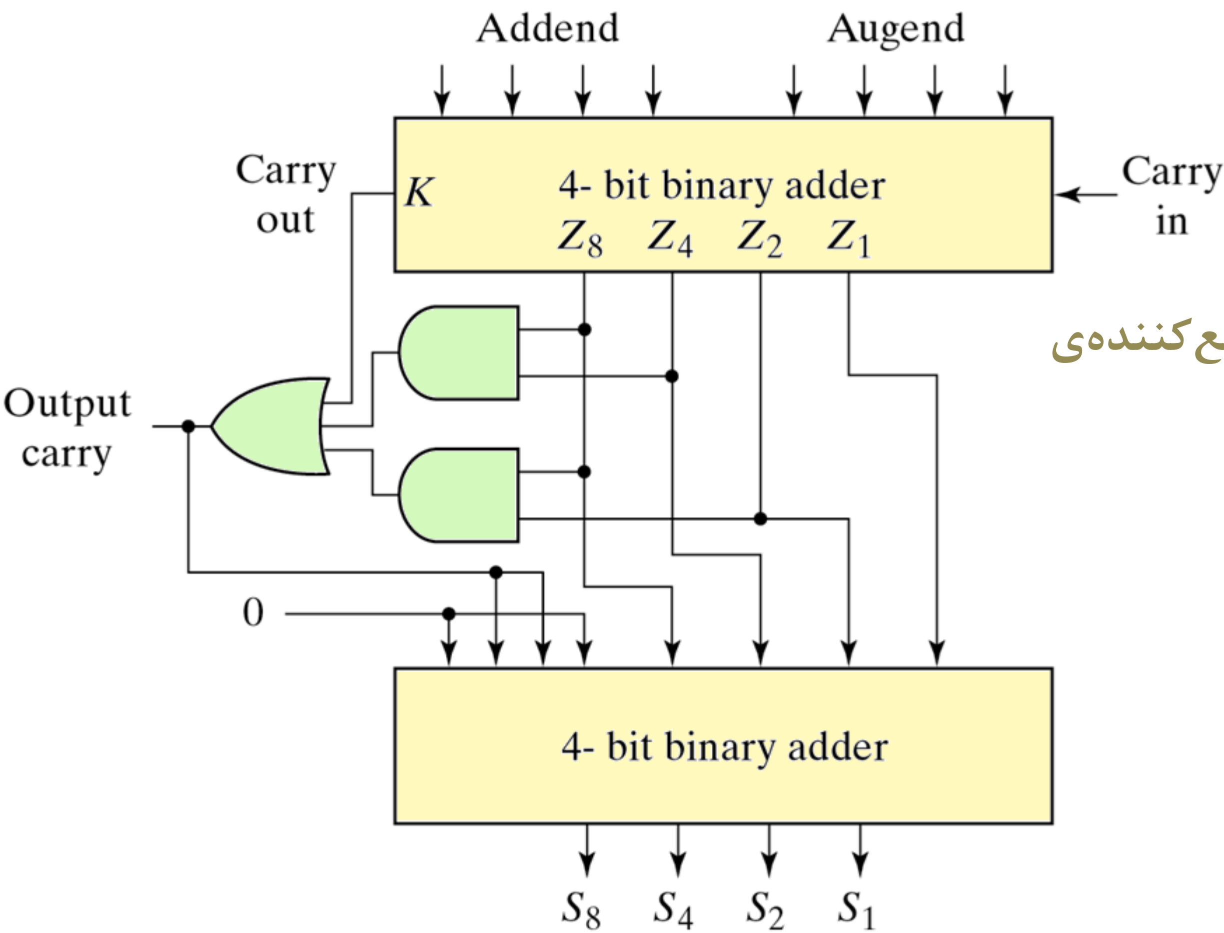
۱- تعیین ورودی‌ها و خروجی‌ها

۲- تشکیل جدول درستی

۳- ساده‌سازی

۴- ترسیم نمودار شماتیک

$C = K + Z_2Z_8 + Z_4Z_8$



شماتیک اجزای بخش جمع‌کننده‌ی چهار بیتی را رسم کنید.

فهرست مطالب

- مقدمه‌ای بر مدارهای ترکیبی
- روش تحلیل و روش طراحی
- جمع‌کننده – تفریق‌گر دودویی
- جمع‌کننده‌ی دهدهی
- ضرب‌کننده‌ی دودویی
- مقایسه‌گر اندازه
- دیکدرها
- انکدرها
- مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- گیت سه حالته

ضرب‌کننده‌ی دودویی(Binary Multiplier)

مثال: یادآوری از مبحث اول درس- ضرب دو عدد دودویی چهاربیتی:

ضرب دودویی زیر را انجام دهید.

1011 * 1001

ضرب‌کننده‌ی دودویی (Binary Multiplier)

مثال: ضرب دو عدد دودویی چهاربیتی:

ضرب دودویی زیر را انجام دهید. $X_3X_2X_1X_0 \;*\; Y_3Y_2Y_1Y_0$

				X_3	X_2	X_1	X_0
			x	Y_3	Y_2	Y_1	Y_0
				$X_3.Y_0$	$X_2.Y_0$	$X_1.Y_0$	$X_0.Y_0$
			$X_3.Y_1$	$X_2.Y_1$	$X_1.Y_1$	$X_0.Y_1$	
		$X_3.Y_2$	$X_2.Y_2$	$X_1.Y_2$	$X_0.Y_2$		
	$X_3.Y_3$	$X_2.Y_3$	$X_1.Y_3$	$X_0.Y_3$			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

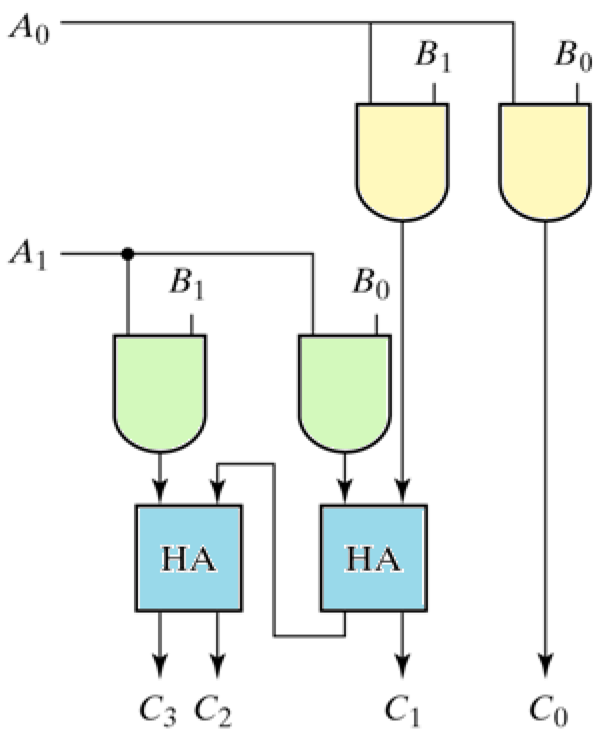
برای پیاده‌سازی این عمل به چه اجزای سخت‌افزاری نیاز است؟

برای ضرب‌کردن یک عدد J بیتی در یک عدد K بیتی، به J*K گیت AND و (K-1) جمع‌کننده‌ی J بیتی نیاز است.

ضرب‌کننده‌ی دودویی (Binary Multiplier)

مثال: ضرب دو عدد دودویی دو بیتی:

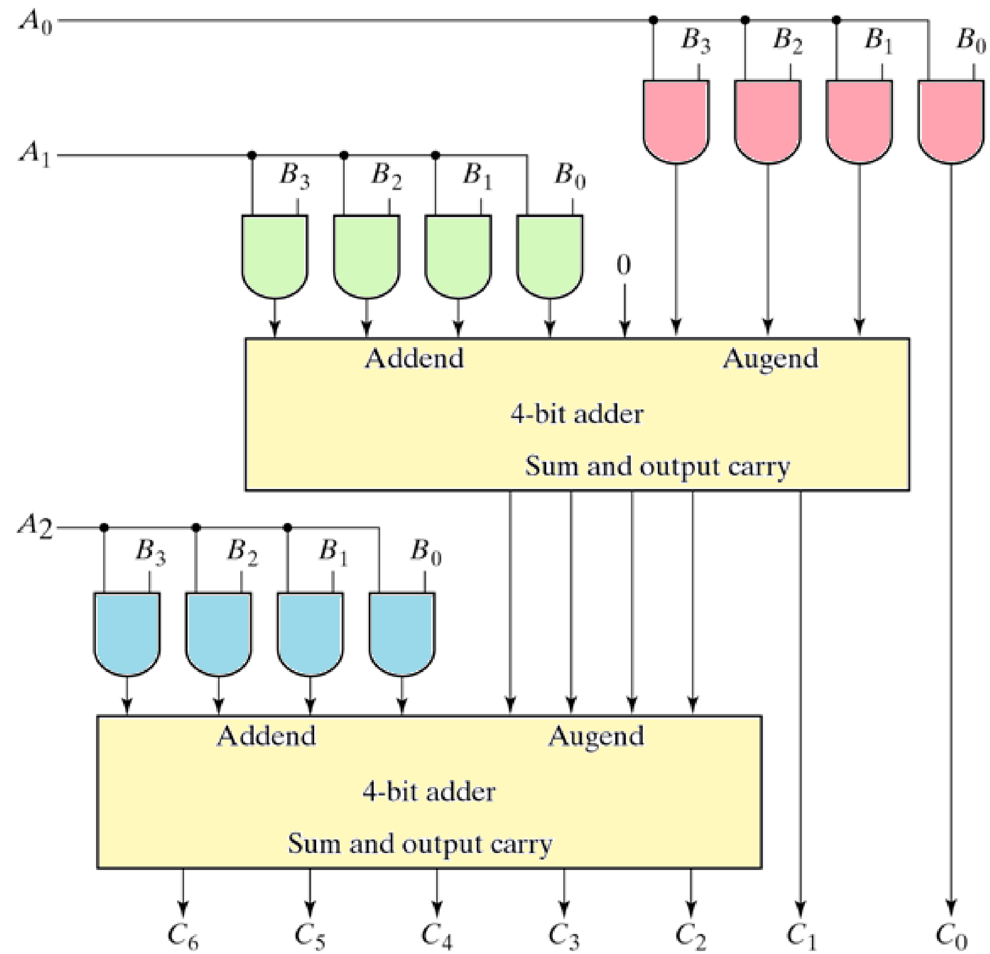
		B_1	B_0
		A_1	A_0
		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	
C_3	C_2	C_1	C_0



ضرب‌کننده‌ی دودویی (Binary Multiplier)

مثال: ضرب دودویی چهار بیت در سه بیت:

			B_3	B_2	B_1	B_0
		×		A_2	A_1	A_0
			B_3A_0	B_2A_0	B_1A_0	B_0A_0
		B_3A_1	B_2A_1	B_1A_1	B_0A_1	
	B_3A_2	B_2A_2	B_1A_2	B_0A_2		
C_6	C_5	C_4	C_3	C_2	C_1	C_0

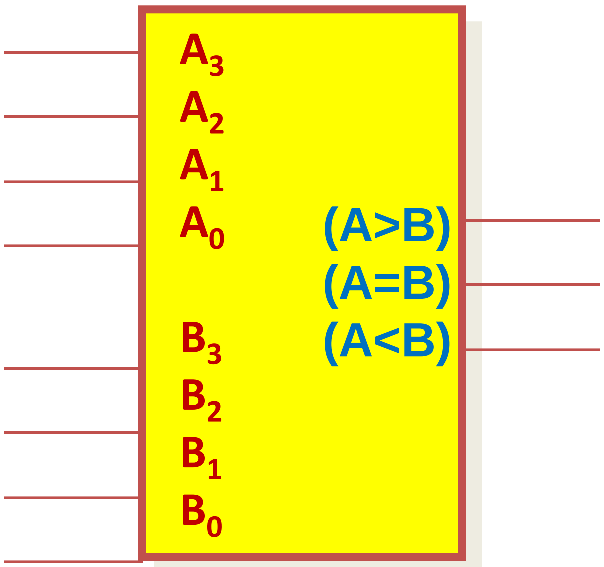


فهرست مطالب

- مقدمه‌ای بر مدارهای ترکیبی
- روش تحلیل و روش طراحی
- جمع‌کننده – تفریق‌گر دودویی
- جمع‌کننده‌ی دهدهی
- ضرب‌کننده ی‌دودویی
- مقایسه‌گر اندازه
- دیکدرها
- انکدرها
- مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- گیت سه حالته

مقایسه‌گر اندازه (Magnitude Comparator)

مقایسه‌گر اندازه مداری است که دو عدد n بیتی ورودی را مقایسه نموده و بزرگتر، کوچکتر یا مساوی بودن آنها را تعیین می‌کند.



Input relation A, B	Output (A>B)	(A=B)	(A<B)
A > B	1	0	0
A = B	0	1	0
A < B	0	0	1

مقایسه‌گر اندازه (Magnitude Comparator)

مثال: اعداد دودویی زیر را مقایسه نموده و بین آنها نمادهای > ، < یا = قرار دهید.

1000 > 0111

1011 < 1111

1000 < 1010

1010 = 1010

مثال: اعداد دودویی ۱۶ بیتی زیر را مقایسه نموده و بین آنها نماد مناسب قرار دهید.

1010101010101010 > 0011111111111111

1010101010101010 < 1011111111111111

1010101010101010 < 1010101010101011

1010101010101010 = 1010101010101010

مقایسه‌گر اندازه (Magnitude Comparator)

بررسی مساوی بودن دو عدد:

دو عدد با هم مساوی است اگر تمامی بیت‌های متناظر دو عدد با هم مساوی باشند.

مساوی بودن بیت‌های i ام دو عدد:

یا هر دو صفر باشند یا هر دو یک باشند.

$$X_i = A_i B_i + A_i' B_i' = (A_i \oplus B_i)'$$

مساوی بودن دو عدد:

مساوی بودن تمام بیت‌های دو عدد

$$F_{(A = B)} = X_3 X_2 X_1 X_0$$

مقایسه‌گر اندازه (Magnitude Comparator)

بررسی بزرگتر بودن A از B :

عدد A از عدد B بزرگتر است اگر:

پرارزش‌ترین رقم A بزرگتر از پرارزش‌ترین رقم B باشد ،

یا

پرارزش‌ترین رقم دو عدد مساوی باشد و رقم پرارزش بعدی A بزرگتر از رقم متناظر در B باشد ،

یا

دو رقم پرارزش دو عدد مساوی باشد و رقم پرارزش بعدی A بزرگتر از رقم متناظر در B باشد ،

یا

سه رقم پرارزش دو عدد مساوی باشد و رقم پرارزش بعدی A بزرگتر از رقم متناظر در B باشد ،

....

$$F_{(A > B)} = A_3B_3' + X_3A_2B_2' + X_3X_2A_1B_1' + X_3X_2X_1A_0B_0'$$

مقایسه‌گر اندازه (Magnitude Comparator)

بررسی کوچک‌تر بودن A از B :

عدد A از عدد B کوچک‌تر است اگر:

پرارزش‌ترین رقم A کوچک‌تر از پرارزش‌ترین رقم B باشد ،

یا

پرارزش‌ترین رقم دو عدد مساوی باشد و رقم پرارزش بعدی A کوچک‌تر از رقم متناظر در B باشد ،

یا

دو رقم پرارزش دو عدد مساوی باشد و رقم پرارزش بعدی A کوچک‌تر از رقم متناظر در B باشد ،

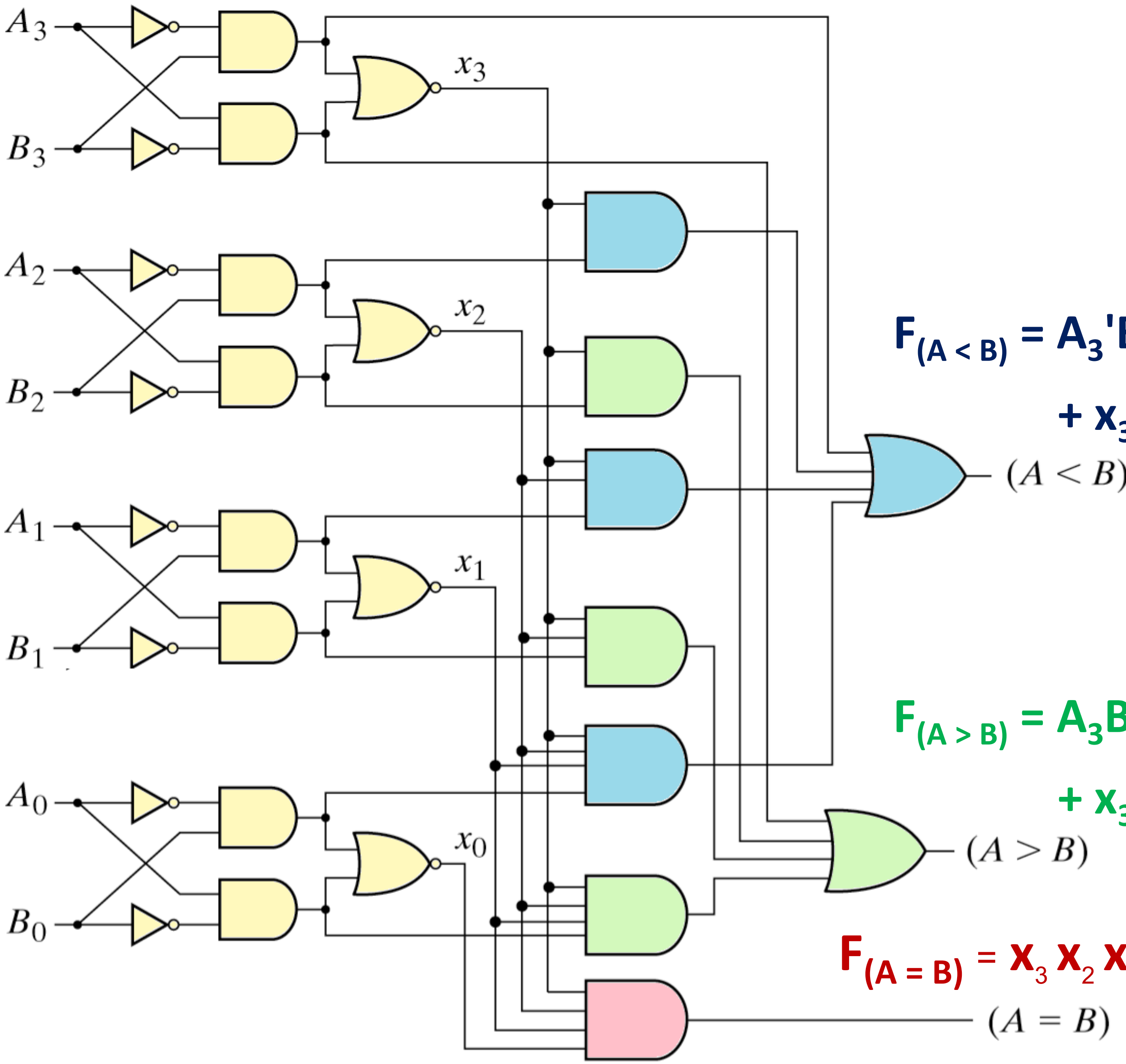
یا

سه رقم پرارزش دو عدد مساوی باشد و رقم پرارزش بعدی A کوچک‌تر از رقم متناظر در B باشد ،

....

$$F_{(A < B)} = A_3'B_3 + X_3A_2'B_2 + X_3X_2A_1'B_1 + X_3X_2X_1A_0'B_0$$

مقایسه‌گر اندازه (Magnitude Comparator)



$$F_{(A<B)} = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$$

$$F_{(A>B)} = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$$

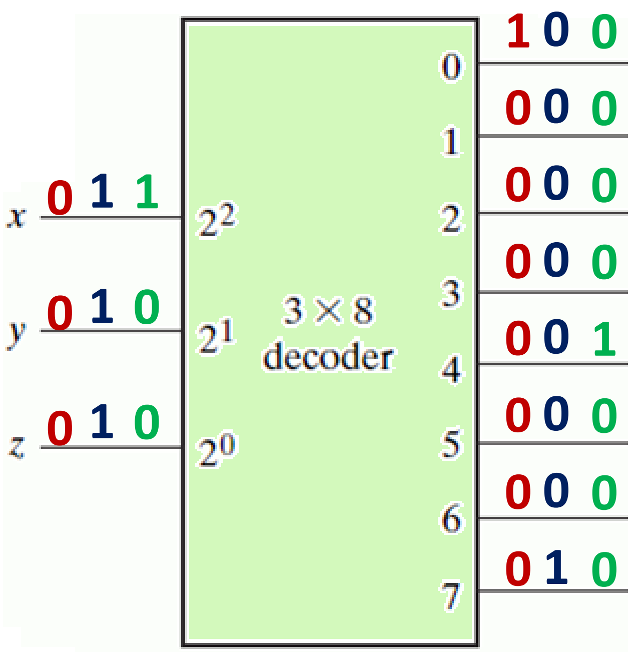
$$F_{(A=B)} = x_3\,x_2\,x_1\,x_0$$

فهرست مطالب

دیکُدر (Decoder)

مداری شامل n ورودی و 2^n خروجی که در هر زمان، تنها یکی از خروجی‌ها که متناظر با کد باینری ورودی است فعال می‌گردد.

مثال: دیکدر ۳ به ۸



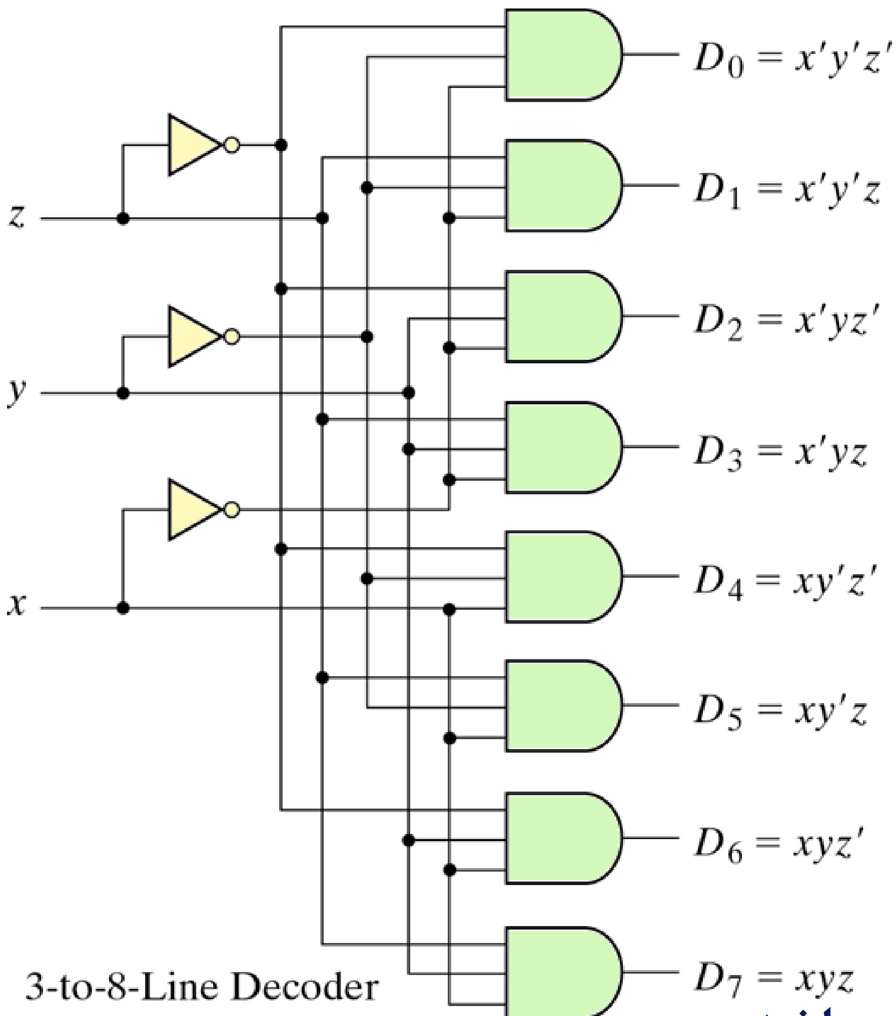
Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
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

دیکُدر (Decoder)

خط D_0 هنگامی فعال می‌گردد که هر سه ورودی صفر باشند.

. . . .



3-to-8-Line Decoder

Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
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

دقت کنید که:

خروجی‌ها minterm‌های مختلف ورودی‌ها می‌باشند.

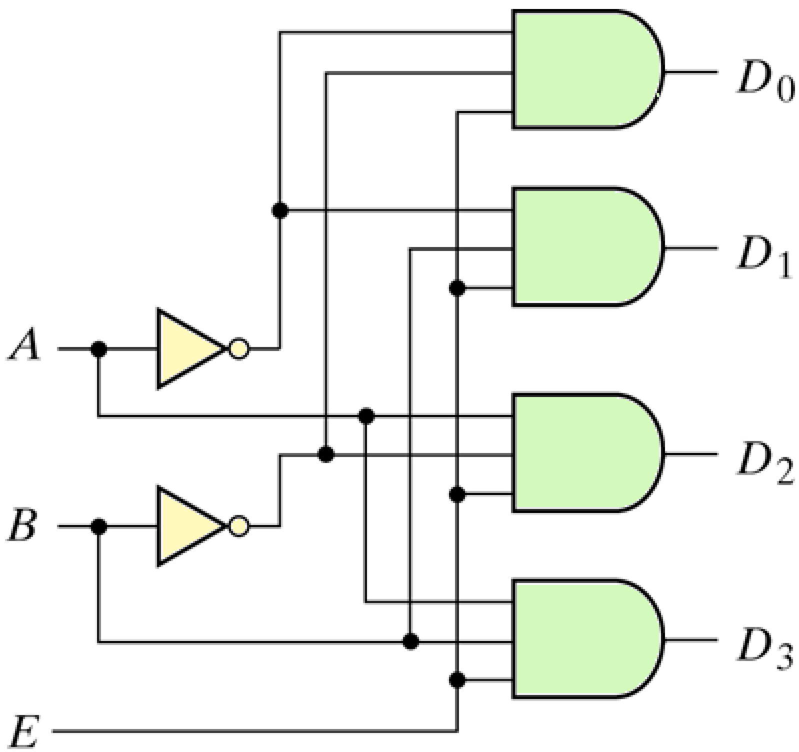
دیکُدر (Decoder)

دیکُدر با ورودی کنترلی فعال‌ساز (Enable):

اگر ورودی فعال‌ساز غیرفعال باشد هیچ کدام از خروجی‌ها فعال نخواهد بود.

اگر ورودی فعال‌ساز فعال گردد خط متناظر با کد دودویی ورودی فعال خواهد شد.

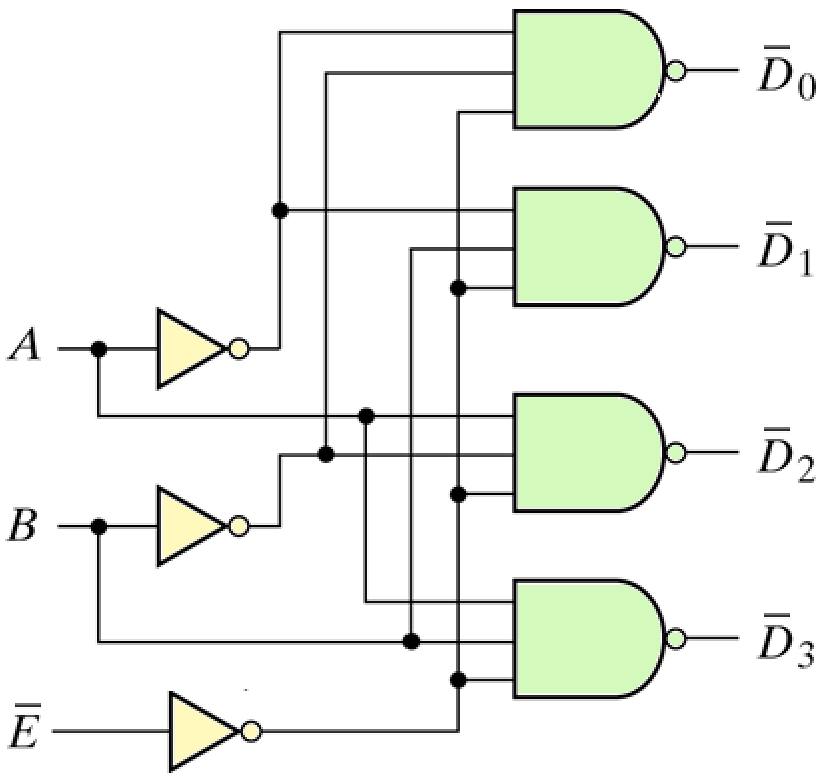
مثال: دیکُدر ۲ به ۴ با ورودی فعال‌ساز:



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

دیکُدر (Decoder)

دیکُدر با ورودی کنترلی فعال‌ساز و خروجی Active Low :



$\bar{E}$	A	B	$\bar{D}_0$	$\bar{D}_1$	$\bar{D}_2$	$\bar{D}_3$
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

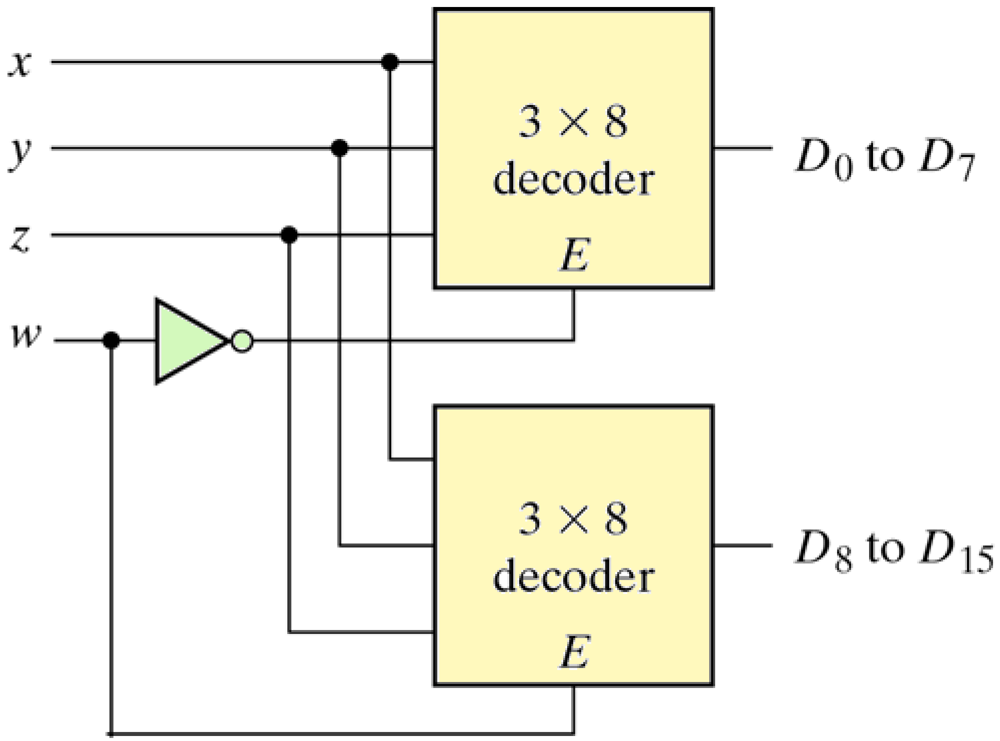
معمولا برای مشخص کردن active low بودن ورودی‌ها و خروجی‌ها، از نمادهای خاصی همراه نام متغیر آنها استفاده می‌گردد.

دیکُدر (Decoder)

ایجاد دیکُدر بزرگتر با دیکُدرهای دارای فعال‌ساز:

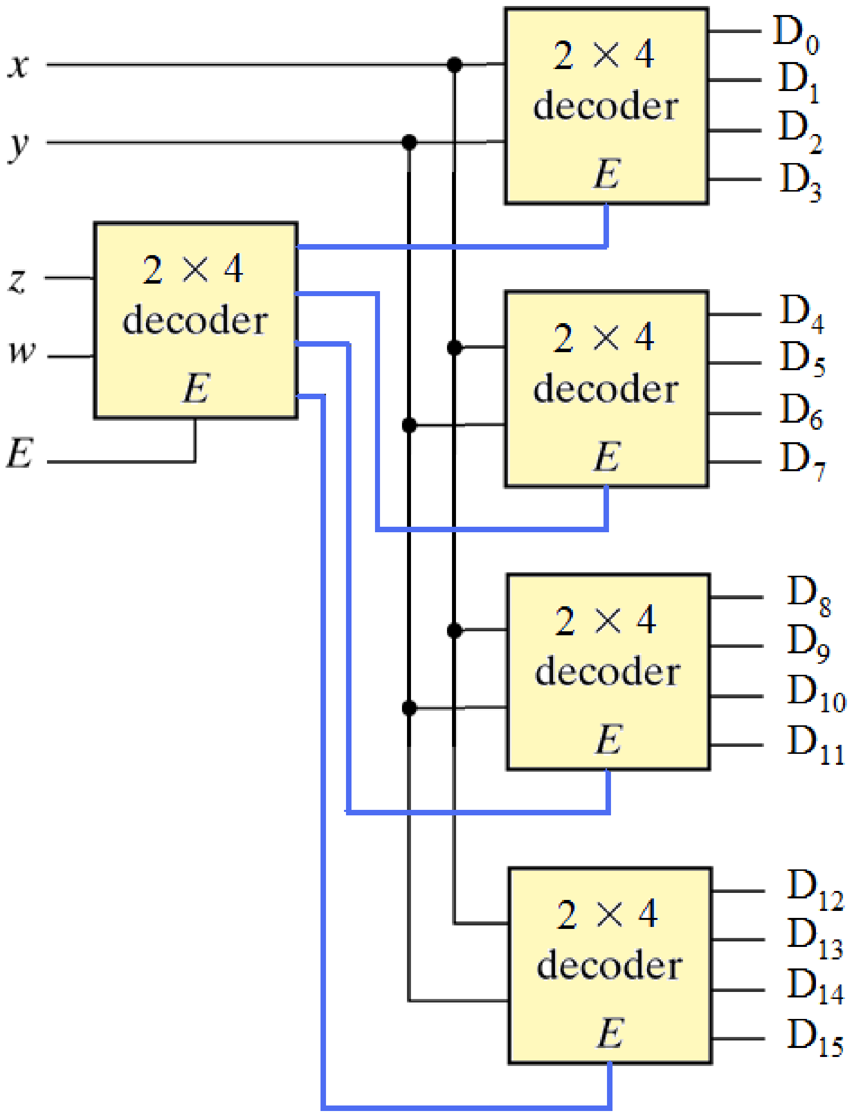
ترکیب‌های مختلف ورودی به زیرمجموعه‌هایی تقسیم می‌گردد که به ازای هر زیرمجموعه، یکی از دیکُدرهای کوچک فعال می‌گردد.

مثال: ایجاد دیکُدر ۴ به ۱۶ با استفاده از دیکُدرهای ۳ به ۸:



دیکُدر (Decoder)

مثال: تنها با استفاده از دیکُدرهای ۲ به ۴ دارای فعال‌ساز، یک دیکُدر ۴ به ۱۶ بسازید.



دیکُدر (Decoder)

پیاده‌سازی توابع با استفاده از دیکُدر:

از آنجا که خروجی‌های دیکُدر در واقع mintermهای متناظر با ورودی‌ها می‌باشند می‌توان از آن برای پیاده‌سازی توابعی که به فرم جمع mintermها توصیف شده‌اند استفاده نمود.

دیکُدر (Decoder)

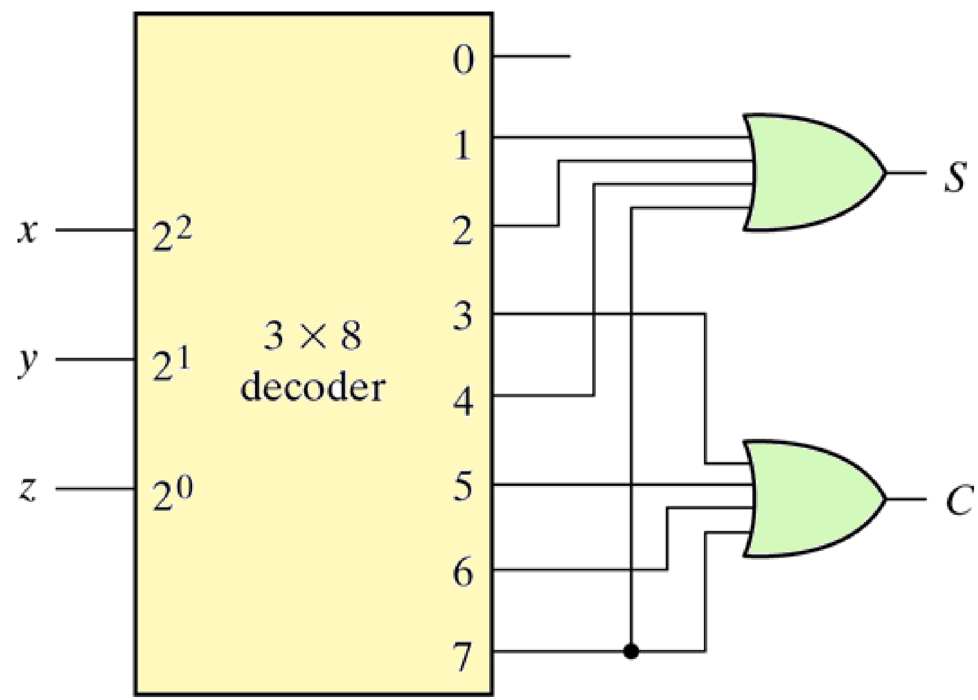
مثال: با استفاده از یک دیکُدر ۳ به ۸ و دیگر گیت‌های مورد نیاز، یک تمام جمع‌کننده پیاده‌سازی نمایید.

Full Adder

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

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

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

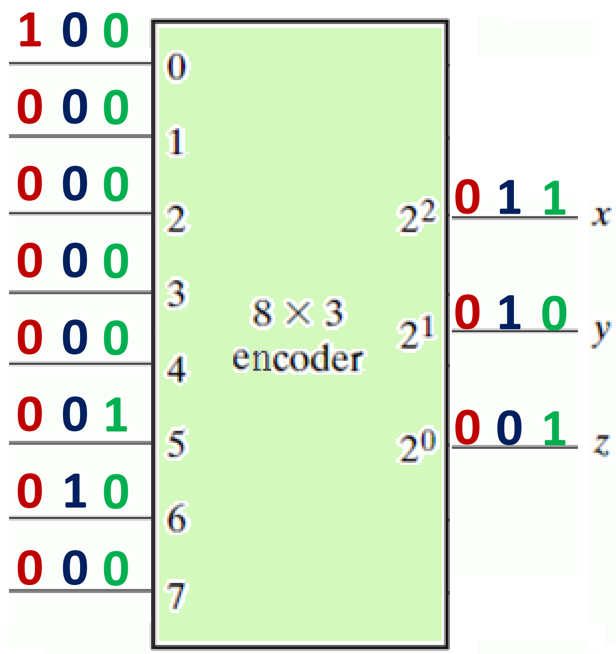


فهرست مطالب

اِنکُدر (Encoder)

مداری دارای 2^n ورودی و n خروجی که در هر زمان تنها یکی از ورودی‌ها فعال بوده و خروجی نشانگر کد دودویی متناظر با ورودی فعال است.

مثال: انکُدر ۸ به ۳



$$Z = D_1 + D_3 + D_5 + D_7$$
$$Y = D_2 + D_3 + D_6 + D_7$$
$$X = D_4 + D_5 + D_6 + D_7$$

Truth Table of an Octal-to-Binary Encoder

Inputs								Outputs		
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

اشکالات اِنکُدر معمولی

برای اِنکُدر معمولی، فرض بر آن است که در هر زمان، یکی و فقط یکی از ورودی‌ها یک باشد.

Truth Table of an Octal-to-Binary Encoder

Inputs								Outputs		
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

- ✓ اگر هیچ کدام از ورودی ها یک نباشد ...
- ✓ اگر دو ورودی (مثلا D_5 و D_6) با هم یک باشد ...

$$Z = D_1 + D_3 + D_5 + D_7$$
$$Y = D_2 + D_3 + D_6 + D_7$$
$$X = D_4 + D_5 + D_6 + D_7$$

برای برطرف کردن این اشکال‌ها چه پیشنهادی دارید؟

انکدر اولویت

اضافه کردن خروجی‌ای که نشانگر معتبر بودن یا نبودن کد ورودی است.

(در صورت غیرفعال بودن تمام ورودی‌ها، نامعتبر بودن ورودی مشخص گردد.)

اولویت‌بندی ورودی‌ها

(اگر چند ورودی با هم فعال شوند ورودی با اولویت بالاتر در نظر گرفته شود.)

Truth Table of a Priority Encoder

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

انکدر اولویت

با استفاده از نقشه‌ی کارنو، توابع خروجی انکُدر اولویت را ساده نموده و شماتیک مداری طرح را رسم کنید.

Truth Table of a Priority Encoder

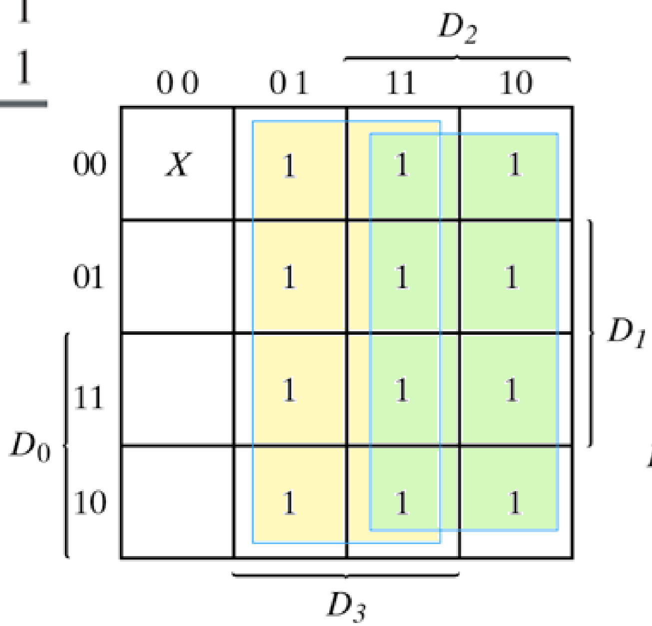
Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

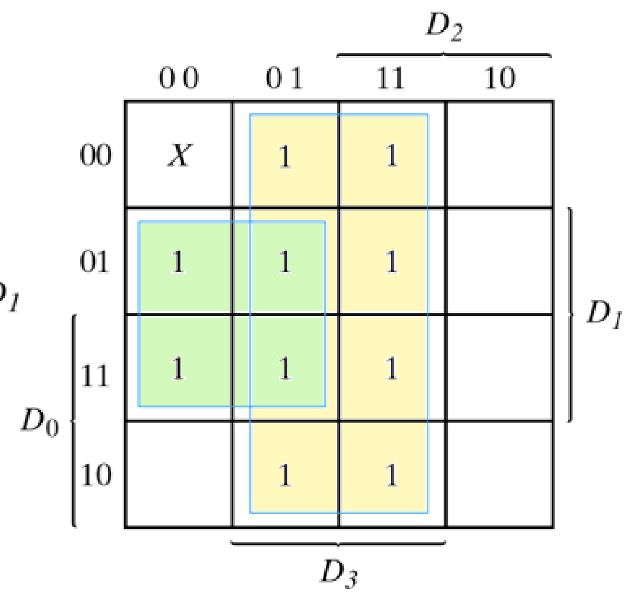
خروجی x را وارد نقشه نمایید.

$x = D_2 + D_3$

$y = D_3 + D_1D_2'$

$V = D_0 + D_1 + D_2 + D_3$





انکدر اولویت

با استفاده از نقشه‌ی کارنو، توابع خروجی انکُدر اولویت را ساده نموده و شماتیک مداری طرح را رسم کنید.

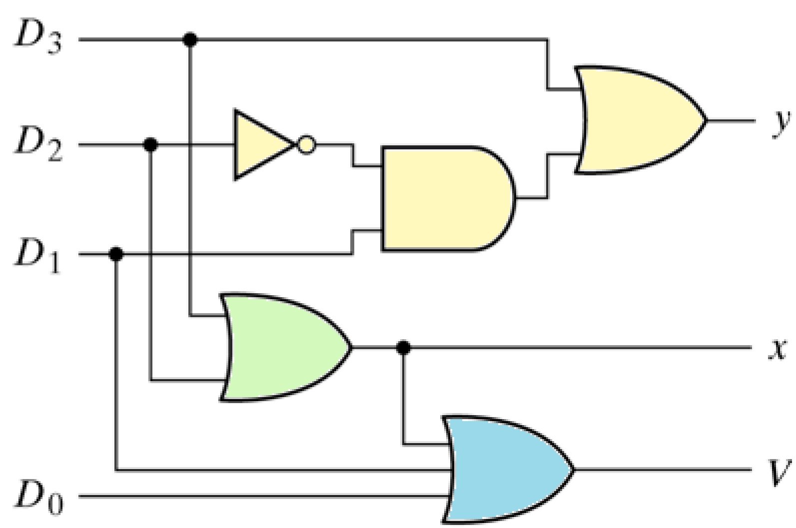
Truth Table of a Priority Encoder

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

$x = D_2 + D_3$

$y = D_3 + D_1 D_2'$

$V = D_0 + D_1 + D_2 + D_3$



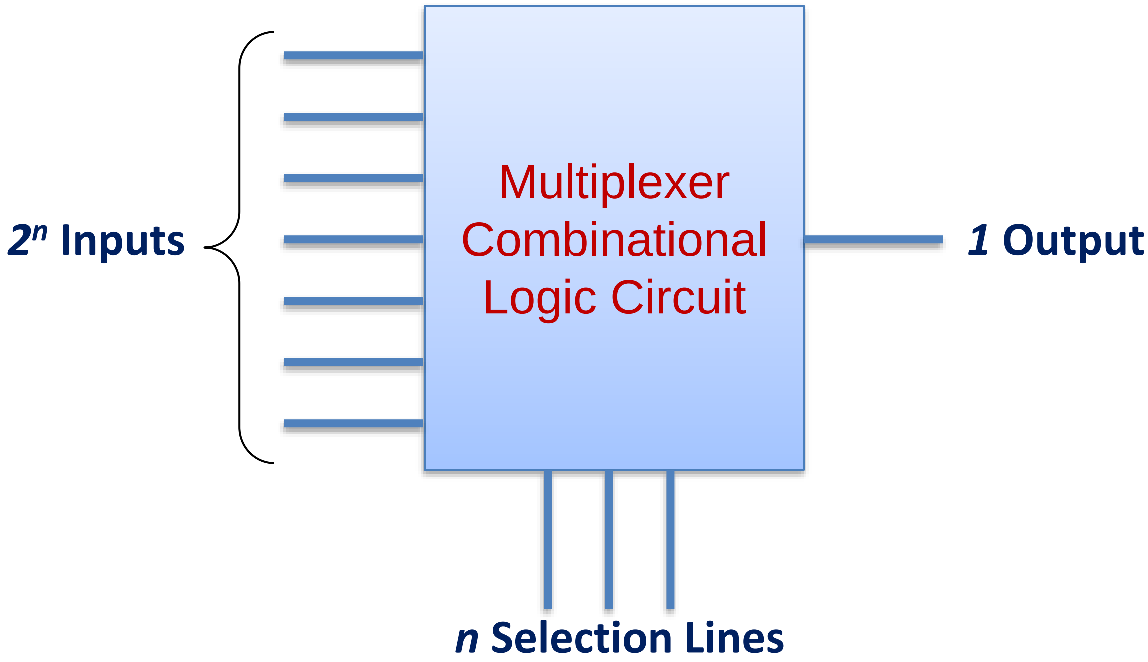
فهرست مطالب

- ☑ مقدمه‌ای بر مدارهای ترکیبی
- ☑ روش تحلیل و روش طراحی
- ☑ جمع‌کننده – تفریق‌گر دودویی
- ☑ جمع‌کننده‌ی دهدهی
- ☑ ضرب‌کننده‌ی دودویی
- ☑ مقایسه‌گر اندازه
- ☑ دیکدرها
- ☑ انکدرها
- ☐ مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- ☐ گیت سه حالته

مالتی‌پلکسرها (Multiplexer-MUX)

مداری که اطلاعات دودویی یکی از خطوط ورودی را انتخاب کرده و به خط خروجی انتقال می‌دهد.

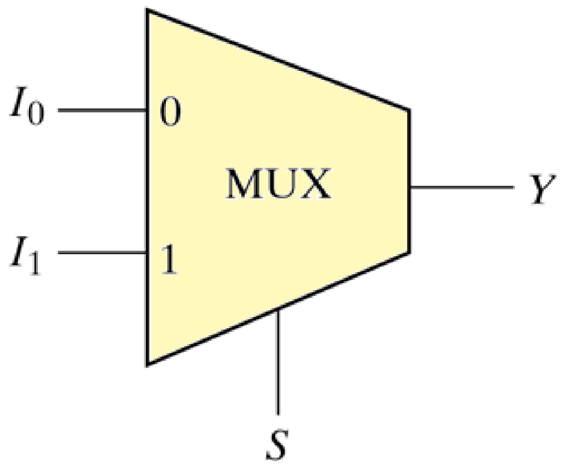
مالتی‌پلکسر دارای ۲^n ورودی، ۱ خروجی و **n** خط انتخاب است. کد باینری خطوط انتخاب تعیین می‌کند که کدام خط انتخاب گردد.



مالتی‌پلکسر ۲ به ۱

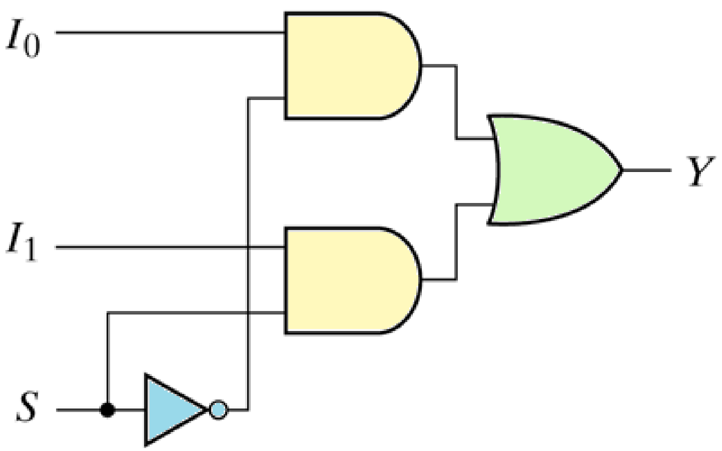
مثال: طراحی مالتی‌پلکسر ۲ به ۱:

مدار مالتی‌پلکسر ۲ به ۱ را طراحی نمایید.



Function table

s	Y
0	I_0
1	I_1



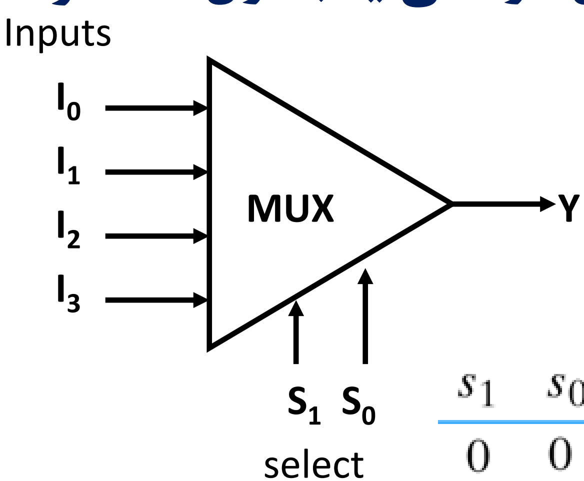
مالتی‌پلکسر ۴ به ۱

مثال: طراحی مالتی‌پلکسر ۴ به ۱:

الف: جدول درستی و جدول عملکرد مالتی‌پلکسر ۴ به ۱ شکل زیر را رسم کنید.

ب: از دیدگاه کاربرپسند بودن، کدام یک برتر است، جدول درستی یا جدول عملکرد؟

ج : مدار مالتی پلکسر ۴ به ۱ را طراحی نمایید.

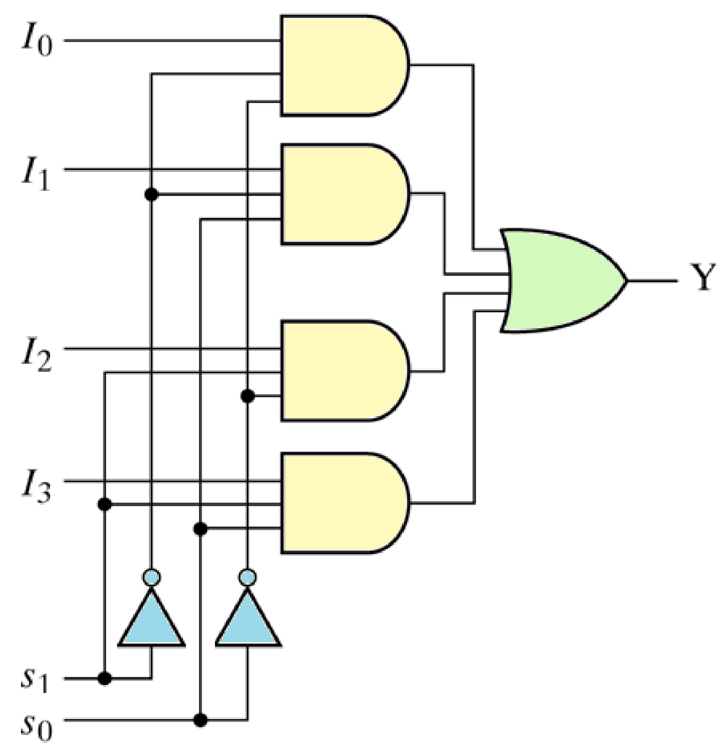


Function table

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

Truth table

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



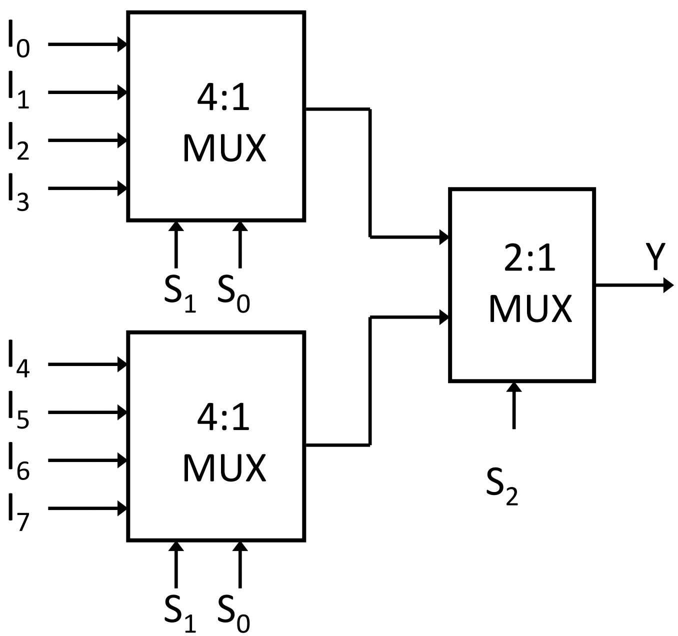
مالتی‌پلکسر ۸ به ۱

مثال: طراحی مالتی‌پلکسر ۸ به ۱ با استفاده از دیکدر ۳ به ۸:

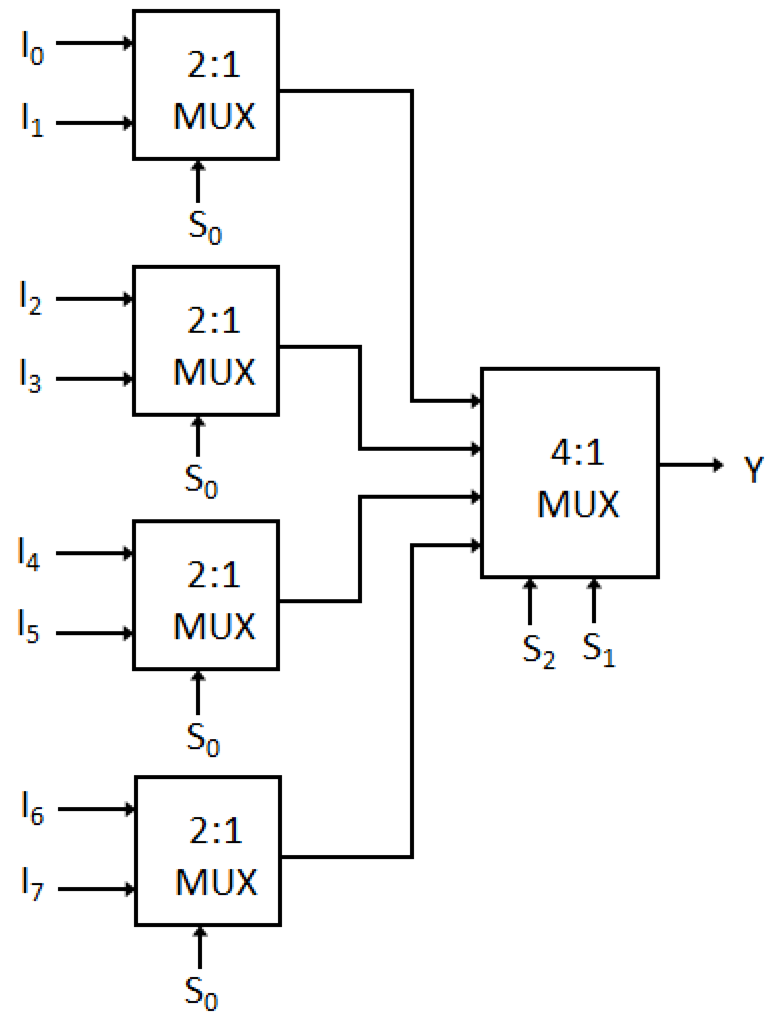
با استفاده از یک دیکُدر ۳ به ۸ و دیگر گیت‌های مورد نیاز، یک مالتی‌پلکسر ۸ به ۱ طراحی نمایید.

مثال: ساخت مالتی‌پلکسر ۸ به ۱ با استفاده از نمونه‌های کوچکتر:

با استفاده از مالتی‌پلکسرهای ۴ به ۱ و ۲ به ۱، یک مالتی‌پلکسر ۸ به ۱ بسازید.



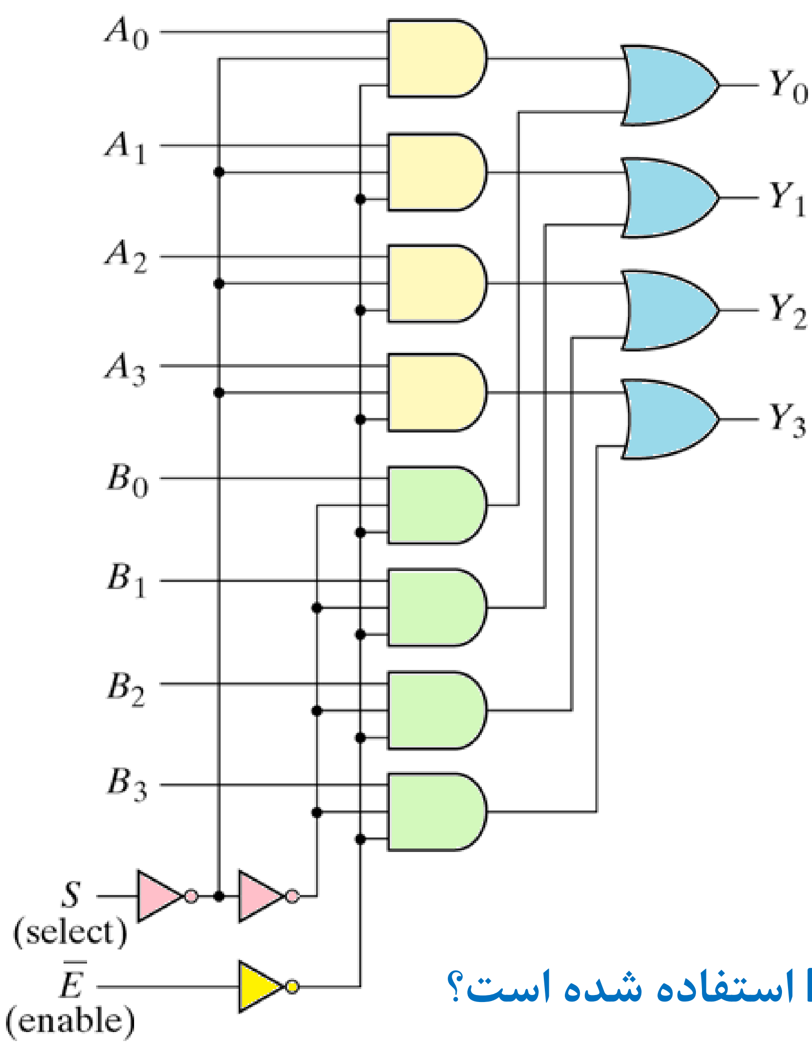
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7



مالتی‌پلکسر ۲ به ۱ (چهارخطی)

اگر بخواهیم از بین چند گذرگاه یکی را انتخاب کنیم نیاز به مالتی‌پلکسر گذرگاه داریم.

مثال: مالتی‌پلکسر ۲ به ۱ چهار خطی:



Function table

$\overline{E}$	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

در این طرح،

چرا در مسیر خط انتخاب (S) از دو گیت **NOT** استفاده شده است؟

پیاده‌سازی تابع با استفاده از مالتی‌پلکسر

- برای پیاده‌سازی یک تابع n متغیره، به یک مالتی پلکسر ۲^{n-1} به یک نیاز است.

- تابع را به فرم جدول درستی توصیف می‌کنیم.

- اولین n-1 متغیر جدول به خطوط انتخاب مالتی‌پلکسر متصل می‌گردند.

- برای هر ترکیبی از متغیرهای انتخاب، خروجی را به صورت تابعی از آخرین متغیر ارزیابی می‌کنیم. این تابع می‌تواند 0 ، 1 ، متغیر یا مکمل متغیر باشد. تابع تعیین شده به ورودی مالتی‌پلکسر متصل می‌گردد.

پیاده‌سازی تابع با استفاده از مالتی‌پلکسر

مثال: تابع زیر را با استفاده از مالتی‌پلکسر پیاده‌سازی نمایید.

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

۱- توصیف طرح به فرم جدول درستی

۲- انتخاب مالتی‌پلکسر و تعیین خطوط انتخاب آن

۳- تعیین ورودی‌های مالتی‌پلکسر به کمک جدول درستی

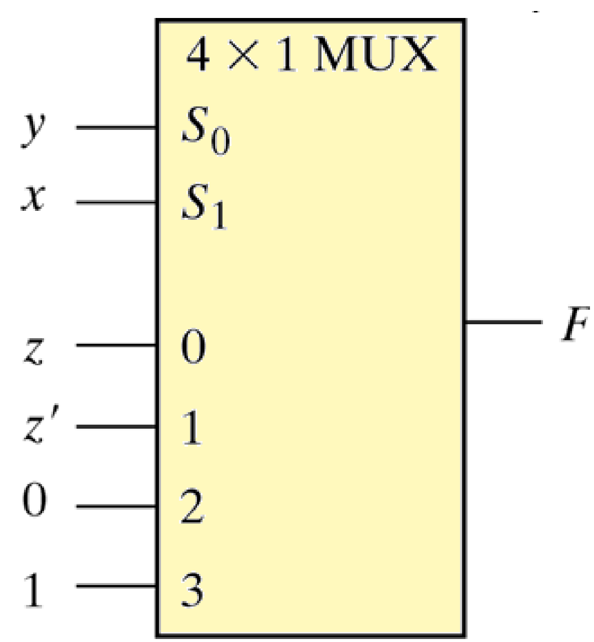
۴- ترسیم مدار

Truth table

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	

مالتی پلکسر ۴ به ۱

خطوط انتخاب: x و y



پیاده‌سازی تابع با استفاده از مالتی‌پلکسر

مثال: تابع زیر را با استفاده از مالتی‌پلکسر پیاده‌سازی نمایید.

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

۱- توصیف طرح به فرم جدول درستی

۲- انتخاب مالتی‌پلکسر و تعیین خطوط انتخاب آن

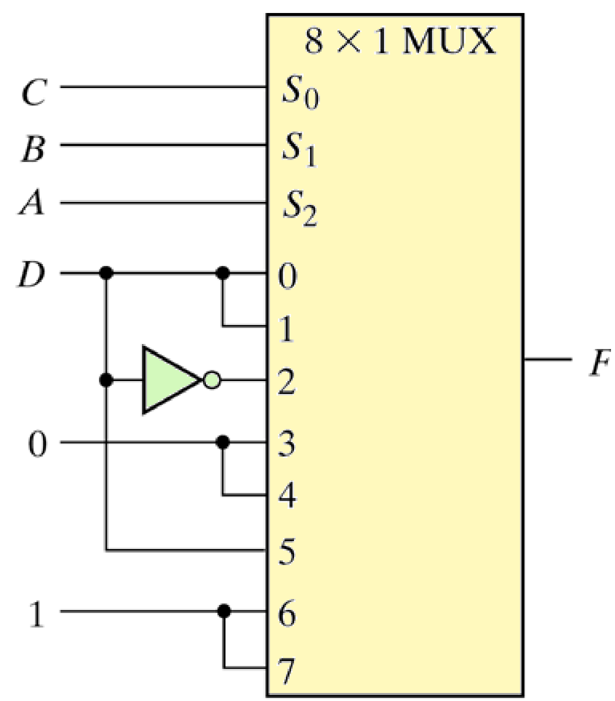
۳- تعیین ورودی‌های مالتی‌پلکسر به کمک جدول درستی

۴- ترسیم مدار

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	

مالتی پلکسر ۸ به ۱

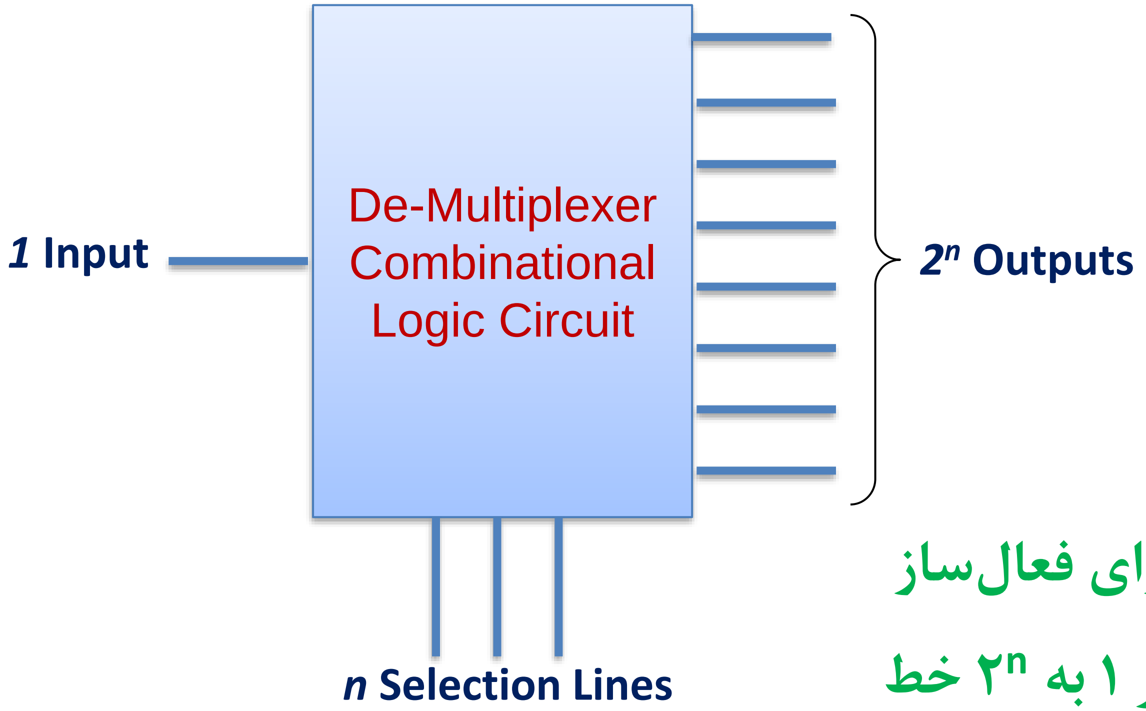
خطوط انتخاب: A، B و C



دی‌مالتی‌پلکسرها (Demultiplexer)

مداری که اطلاعات دودویی خط ورودی را به یکی از خطوط خروجی انتقال می‌دهد.

دی‌مالتی‌پلکسر دارای یک ورودی، ۲^n خروجی و n خط انتخاب است. کد باینری خطوط انتخاب تعیین می‌کند که ورودی به کدام خط خروجی انتقال یابد.



معمولا از دیکُدر n به ۲^n خط دارای فعال‌ساز برای پیاده‌سازی دی‌مالتی‌پلکسر ۱ به ۲^n خط استفاده می‌گردد.

دی‌مالتی‌پلکسرها (Demultiplexer)

مثال: پیاده‌سازی دی‌مالتی‌پلکسر با استفاده از دیکُدر:

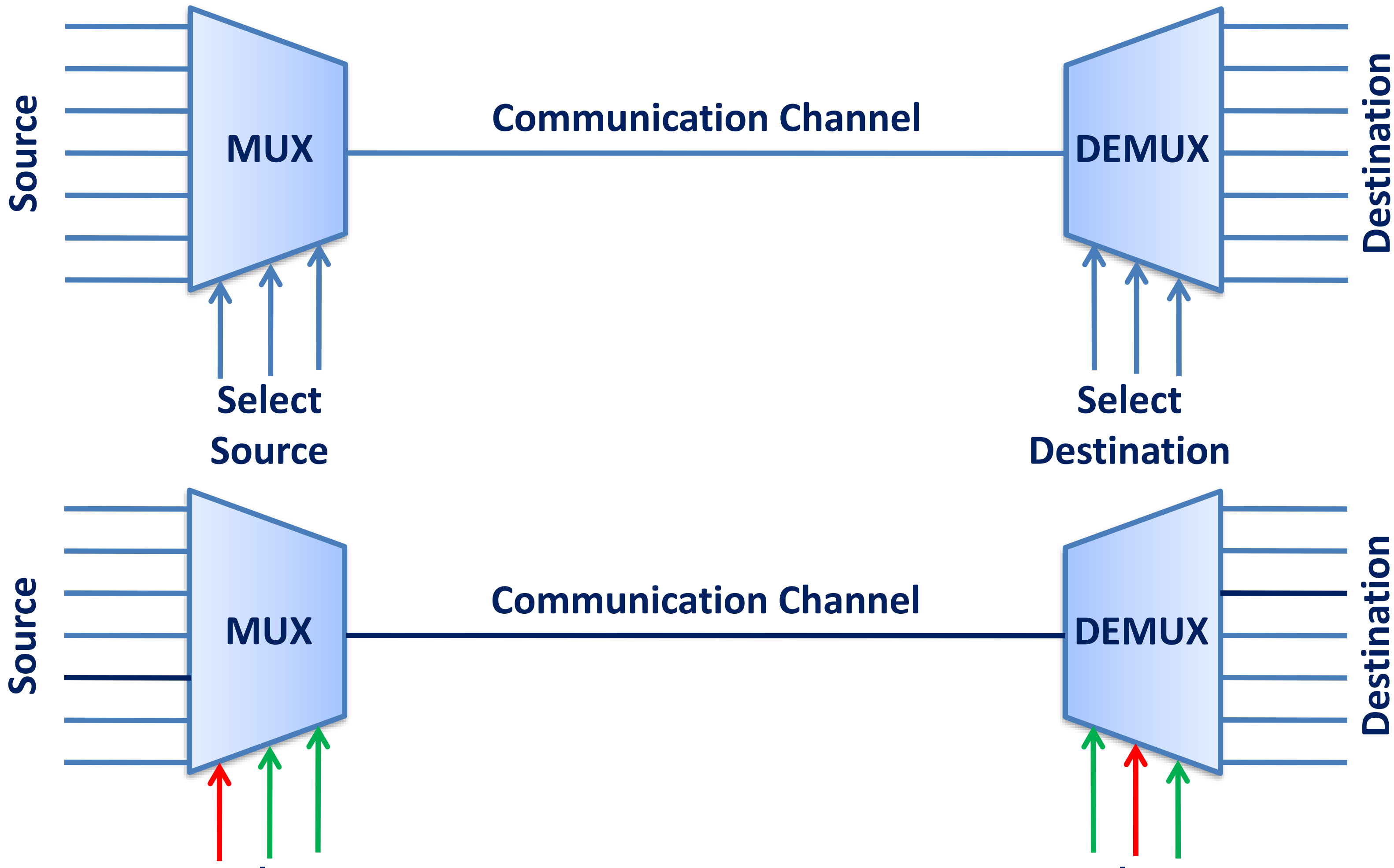
با استفاده از یک دیکُدر ۳ به ۸ دارای پایه‌ی فعال‌ساز، یک دی‌مالتی‌پلکسر ۱ به ۸ پیاده‌سازی نمایید.

بدین منظور لازم است از ورودی‌های دیکُدر به عنوان خطوط انتخاب دی‌مالتی‌پلکسر، و از خط فعال‌ساز آن به عنوان ورودی دی‌مالتی‌پلکسر استفاده گردد.

مثال کاربردی مالتی‌پلکسر – دی‌مالتی‌پلکسر

مثال: به اشتراک گذاشتن یک کانال ارتباطی بین چندین دستگاه:

در هر زمان تنها یکی از مبداها و یکی از مقصدها به کانال ارتباطی دسترسی دارند.



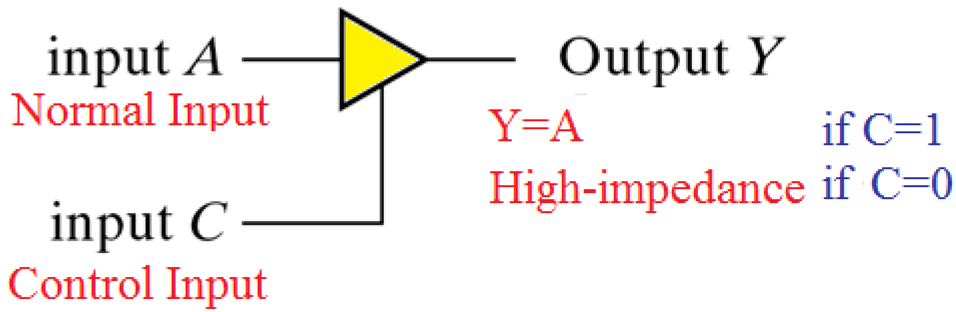
فهرست مطالب

- ☑ مقدمه‌ای بر مدارهای ترکیبی
- ☑ روش تحلیل و روش طراحی
- ☑ جمع‌کننده – تفریق‌گر دودویی
- ☑ جمع‌کننده‌ی دهدهی
- ☑ ضرب‌کننده‌ی دودویی
- ☑ مقایسه‌گر اندازه
- ☑ دیکدرها
- ☑ انکدرها
- ☑ مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- ☐ گیت سه‌حالته

گیت سه‌حالته (tri-state gate)

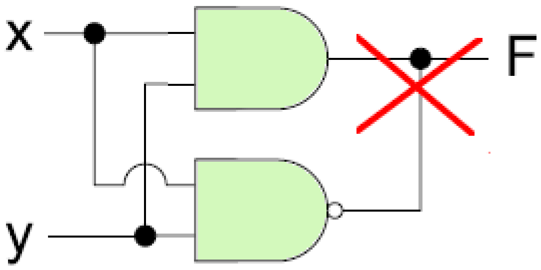
گیت سه‌حالته به گیتی گفته می‌شود که خروجی آن علاوه بر وضعیت‌های **low** و **high** ، می‌تواند در وضعیت امپدانس بالا (**high-impedance**) نیز قرار گیرد.

مثال: بافر سه‌حالته



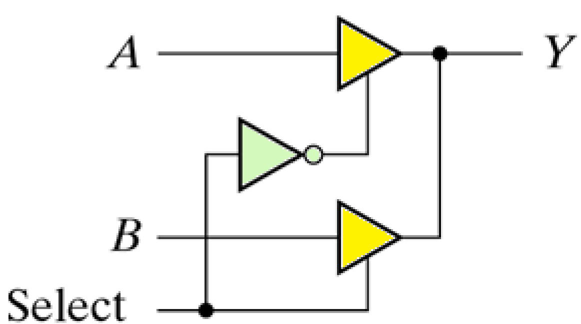
گیت سه‌حالته (tri-state gate)

در گیت‌های معمولی نمی‌توان خروجی گیت‌ها را به طور مستقیم به هم متصل نمود.



ولی در گیت‌های سه‌حالته، اگر جز خروجی یکی از گیت‌ها، خروجی بقیه گیت‌ها در وضعیت امپدانس- بالا باشد می‌توان خروجی گیت‌ها را به هم متصل نمود.

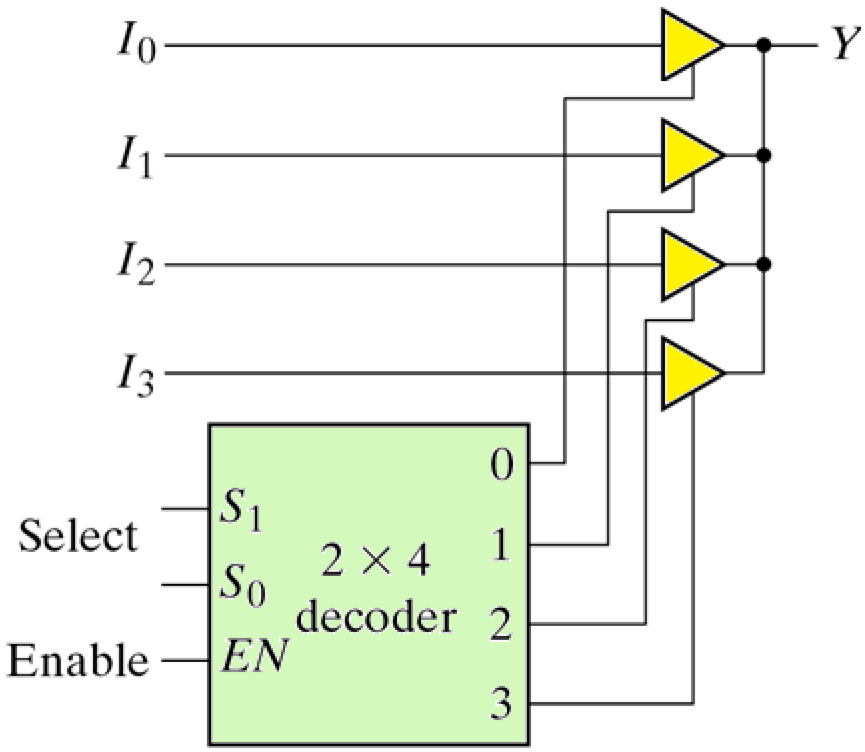
مثال: پیاده‌سازی مالتی‌پلکسر ۲ به ۱ با استفاده از گیت‌های بافر سه‌حالته



گیت سه‌حالته (tri-state gate)

مثال: پیاده‌سازی مالتی‌پلکسر توسط گیت‌های سه‌حالته:

با استفاده از دیکدر ۲ به ۴ و گیت‌های بافر سه‌حالته، یک مالتی‌پلکسر ۴ به ۱ بسازید.



فهرست مطالب

- مقدمه‌ای بر مدارهای ترکیبی
- روش تحلیل و روش طراحی
- جمع‌کننده – تفریق‌گر دودویی
- جمع‌کننده‌ی دهدهی
- ضرب‌کننده‌ی دودویی
- مقایسه‌گر اندازه
- دیکدرها
- انکدرها
- مالتی‌پلکسرها و دی‌مالتی‌پلکسرها
- گیت سه حالته