باسمه تعالی

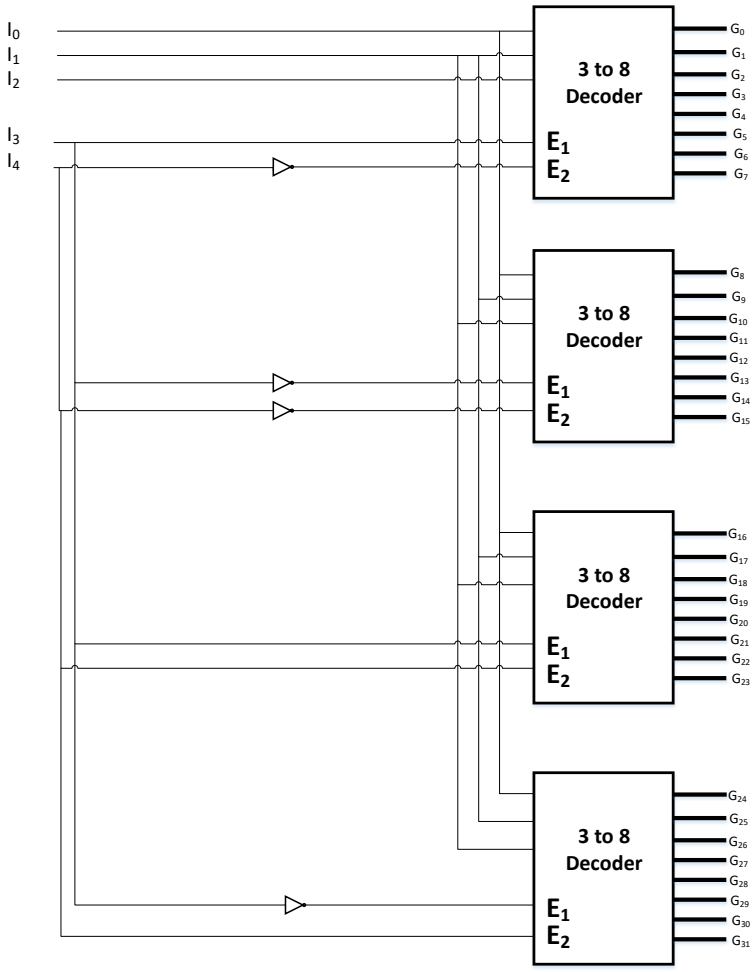
مدار منطقی

نیم‌سال دوم ۰۰-۹۹

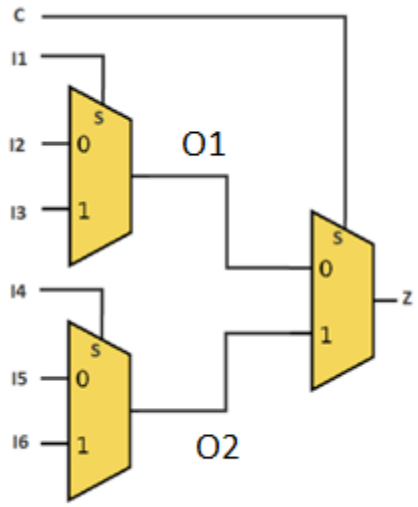
پاسخنامه تمرین
سری چهارم

سوال ۱)

وجود دو سیگنال enable در هر کدگشا به این معناست که برای فعال شدن هر کدام، هر دو سیگنال enable باید فعال شوند. برای ساختن یک کدگشای ۵ به ۳۲ با استفاده از کدگشاهای ۳ به ۸، می‌توان چهار کدگشای ۳ به ۸ را به هم متصل کرد. از آنجایی که در هر لحظه فقط یکی از کدگشاها باید فعال باشند، این کار را می‌توان با استفاده از گیت‌های منطقی و دو بیت پرارزش ورودی به انجام رساند. راه‌حل دیگر، استفاده از یک کدگشای ۳ به ۸ دیگر است که با استفاده از دو ورودی آن بتوان فعال شدن چهارکدگشای خروجی را کنترل کرد.



سوال ۲)



در ابتدا توابع *O1* و *O2* را بر حسب ورودی‌های داده شده بدست می‌آوریم.

$$O_1 = \bar{I_1}I_2 + I_1I_3$$

$$O_2 = \bar{I_4}I_5 + I_4I_6$$

تابع Z برابر است با $Z = \bar{c}(a+b) + (ab)$ و از طرفی $Z = \bar{c}(O_1) + c(O_2)$ در نتیجه داریم :

$$O_1 = \bar{I_1}I_2 + I_1I_3 = a + b$$

$$O_2 = \bar{I_4}I_5 + I_4I_6 = ab$$

به صورت خیلی ساده می‌توان با جایگذاری ورودی‌ها را بر حسب a و b بدست آورد.

$$I_4 = a, I_5 = 0, I_6 = b \rightarrow O_2 = ab$$

$$I_1 = a, I_2 = b, I_3 = 1 \rightarrow O_1 = \bar{a}b + b = a + b$$

در نتیجه داریم $I_1I_2I_3I_4I_5I_6 = ab1a0b$

سوال ۳)

الف) برای طراحی یک جمع کننده کامل، ابتدا لازم است تا جدول درستی آن را به ازای ورودی‌هایش رسم کنیم تا خروجی‌ها مشخص گردند. این جدول برای قسمت الف از این قرار است.

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

مطابق این جدول درستی، جدول کارنو را برای دو خروجی تشکیل می‌دهیم.

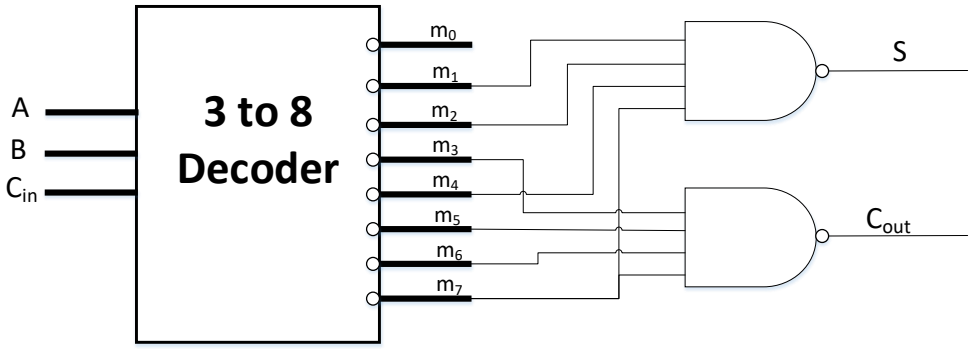
C_{in} \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

برای خروجی S

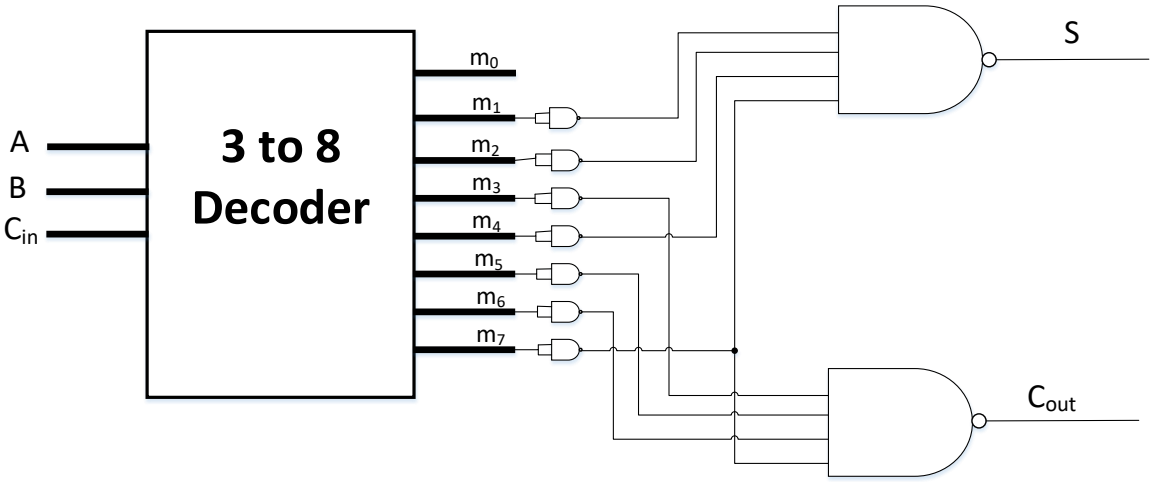
C_{in} \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

برای خروجی C_{out}

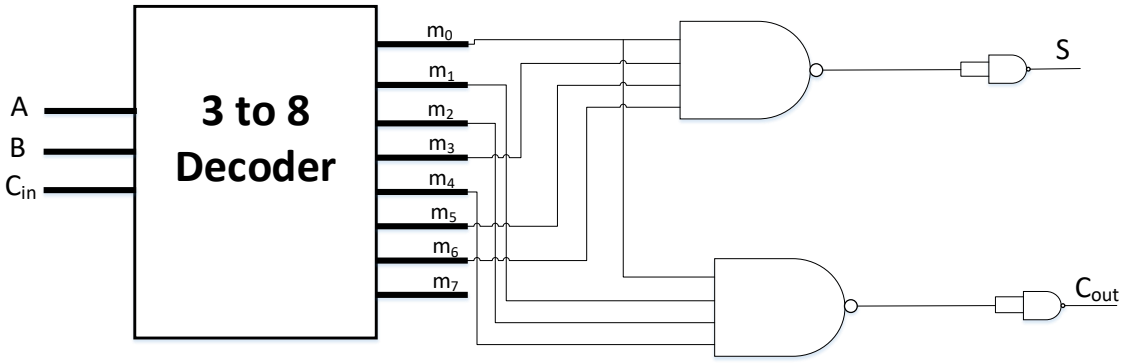
از روی جدول کارنو، بلوک دیاگرام مدار حاصل به شکل زیر خواهد بود. اگر کدگشا را به صورت فعال‌پایین فرض کنیم:



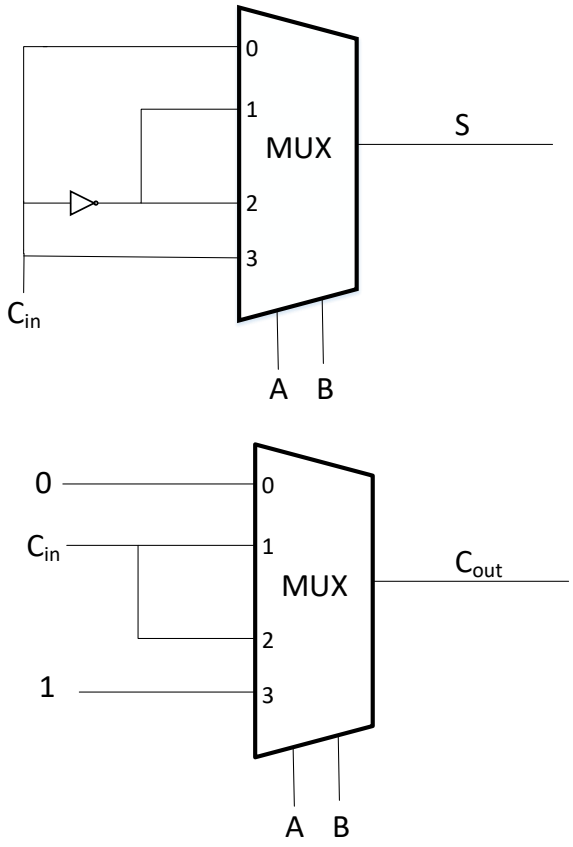
جهت پیاده‌سازی با استفاده از کدگشای فعال‌بالا، باید در خروجی‌های کدگشا از گیت‌های معکوس‌کننده (NOT) استفاده کنیم. اما چون تنها مجاز به استفاده از گیت NAND هستیم، عمل معکوس کردن را هم با همان گیت انجام می‌دهیم. بدین منظور، خروجی کدگشا را به هر دو ورودی گیت NAND می‌دهیم چرا که می‌دانیم: $\overline{a.a} = \bar{a}$ یا به عبارت دیگر: NAND(a, a) = NOT(a). حال بلوک دیاگرام آن را رسم می‌کنیم:



همچنین، در حالتیکه از روش ضرب ماکسترم‌ها عمل کنیم، بلوک دیاگرام آن شبیه شکل زیر خواهد شد:



ب) مطابق با جدول درستی قسمت الف، مدار را رسم می‌کنیم.



سوال ٤)

الف) در کدگشاهای فعال-پایین زمانی که یک خروجی فعال است مقدار ۰ را به خورد می‌گیرد و خروجی‌های غیر فعال مقدار ۱. یک راه برای دست یافتن به تابع F بدست آوردن حالت‌هایی است که یکی از ورودی‌های and برابر ۰ شود (توجه شود به علت اینکه خروجی کدگشاها در حالت فعال ۰ می‌شود فقط یکی از ورودی‌های and همزمان می‌تواند ۰ باشد) در نتیجه خروجی F برابر ۰ خواهد شد. این حالت زمانی رخ خواهد داد که abc = 001, 011, 110 باشد و در بقیه حالت‌ها خروجی F برابر ۱ خواهد بود. در نتیجه جدول کارنو مربوطه به شکل زیر در خواهد آمد.

A \ BC	00	01	11	10
0	**1** m0	**0** m1	**0** m3	**1** m2
1	**1** m4	**1** m5	**1** m7	**0** m6

که فرم SOP آن برابر است با $f(a,b,c) = \sum m(0,2,4,5,7)$.

ب) جواب بله است. در واقع این دو کدگشا به واسطه‌ی ورودی Enable در هم ادغام شده‌اند و یک کدگشا 3×8 را ساخته اند. جدول درستی این دو کدگشا همراه با ورودی Enable به شکل زیر می‌باشد. (خروجی Y_0, Y_1, Y_2, Y_3 مربوط به کدگشای اول و خروجی Y_4, Y_5, Y_6, Y_7 مربوط به کدگشای دوم هستند.)

Inputs			Outputs							
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

این جدول برابر است با جدول عملکرد یک کدگشا ۳ ورودی بدون Enable با خروجی فعال-پایین.

سول ۵)

در این سوال تابع F تا زمانی که Enable کدگشا فعال نباشد قطعا برابر ۰ خواهد بود. از طرفی ورودی‌های کدگشا به شکل 1w هستند که دو حالت دارد ۱۰ و ۱۱ این یعنی خروجی ۰ کدگشا هیچگاه فعال نمی‌شود و همیشه مقدار ۰ را دارد در نتیجه ورودی D_0 مالتی‌پلکسر همیشه ۰ می‌باشد. F زمانی می‌تواند ۱ باشد که $w = 1$ باشد به این دلیل که باید ابتدا F فعال شود. از طرفی خط D_0 مالتی‌پلکسر نباید انتخاب شود چون می‌دانیم مقدار ۰ را دارد و در صورت انتخاب خط Enable کدگشا فعال نمی‌شود. همچنین تا زمانی که خط D_3 مالتی‌پلکسر فعال نشده باشد یعنی $xy = 11$ مقدار z مهم نیست. در این صورت می‌توان جدول درستی تابع F را به شکل زیر رسم کرد:

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

پس تابع $F = wxyz$ می‌باشد. دقت کنید با محدود کردن حالت‌های موجود تا حد امکان تعداد حالات جدول درستی را کاهش داده‌ایم. گاهی اوقات مخصوصا در حالت‌هایی که فیدبک داریم تابع‌نویسی پیچیدگی حل مسئله را افزایش می‌دهد.

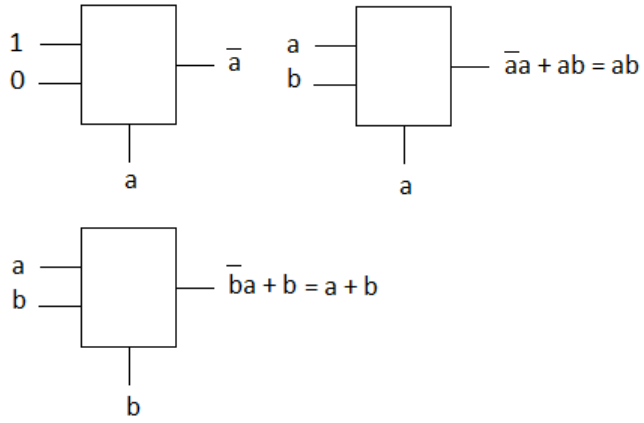
سوال ٦)

نیم‌جمع کننده:

یک راه آسان و راحت برای بررسی کامل بودن یک واحد منطقی می‌تواند ساختن یک واحد منطقی کامل که از قبل می‌شناسیم باشد. خروجی‌های نیم‌جمع کننده به صورت $s = a \oplus b, c_{out} = ab$ می‌باشد. and را که داریم، از طرفی xor را داریم که می‌توان با آن not را به صورت $a \oplus 1 = \bar{a}$ ساخت. and و not را داریم در نتیجه nand که یک واحد منطقی کامل است را داریم پس نیم‌جمع کننده کامل است.

مالتی‌پلکسر:

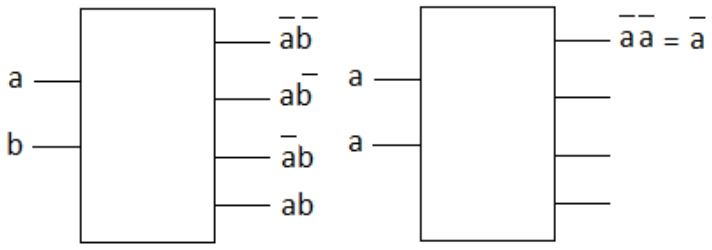
یک مالتی‌پلکسر 2×1 را در نظر میگیریم.



همانطور که مشاهده می‌شود توانستیم and,or,not را بسازیم و این واحد منطقی کامل است.

کدگشا:

کدگشا تولید کننده مینترم‌های ورودی می‌باشد. در نتیجه اگر یک کدگشا دو ورودی را در نظر بگیریم داریم :



همانطور که مشاهده می‌شود not و and را ساختیم و در نهایت می‌توان به nand رسید که یک واحد منطقی کامل است.