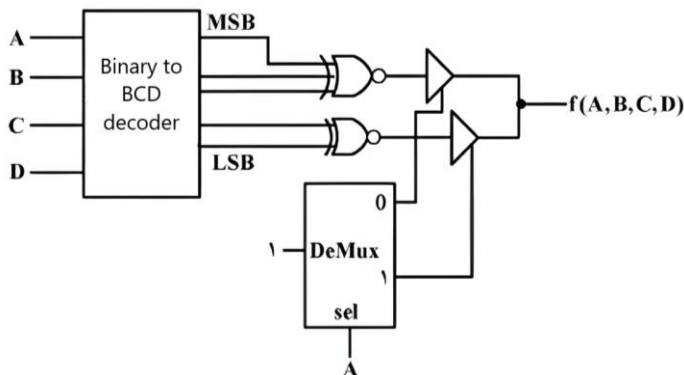




به موارد زیر توجه کنید:

- ۱- حتما نام و شماره دانشجویی خود را روی پاسخنامه بنویسید.
- ۲- در حل سوالات به نوشتن جواب آخر اکتفا نکنید. همه مراحل میانی را هم بنویسید.
- ۳- کل پاسخ تمرینات را در قالب یک فایل pdf با شماره دانشجویی خود نام گذاری کرده در سامانه CW بارگذاری کنید.
- ۴- در صورت مشاهده هر گونه مشابهت نامتعارف هر دو (یا چند) نفر کل نمره این تمرین را از دست خواهند داد.

۱- (۲ نمره) مشخص کنید مدار زیر بر حسب مینترمها چه تابعی را پیاده سازی می کند. توجه کنید در تبدیل کننده باینری به BCD، ورودی A با ارزش ترین و ورودی D کم ارزش ترین هستند. همچنین خروجی MSB با ارزش ترین و خروجی LSB کم ارزش ترین هستند.



پاسخ:

از با ارزش ترین تا کم ارزش ترین بیت خروجی دیکودر را به ترتیب f_4 تا f_0 می نامیم.

موقعی که $A = 0$ آنگاه $f = \overline{(f_4 \oplus f_3 \oplus f_2)}$ و موقعی که $A = 1$ آنگاه $f = \overline{(f_1 \oplus f_0)}$ است.

پس به ازای حالت های زیر، خروجی f یک می شود.

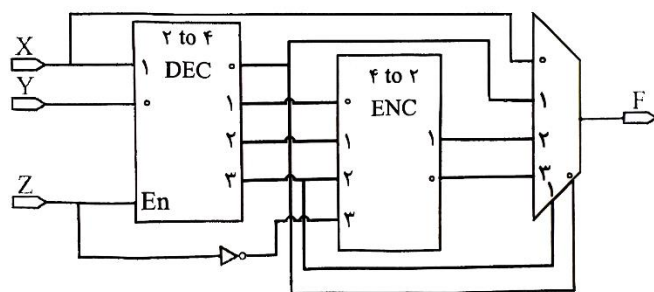
$ABCD = 0000, 0001, 0010, 0011, 1000, 1010, 1101, 1110$

که مینترم های معادل آن عبارتند از:

$m_0, m_1, m_2, m_3, m_8, m_{10}, m_{13}, m_{14}$

ABCD	f_4	f_3	f_2	f_1	f_0
0000	0	0	0	0	0
0001	0	0	0	0	1
0010	0	0	0	1	0
0011	0	0	0	1	1
0100	0	0	1	0	0
0101	0	0	1	0	1
0110	0	0	1	1	0
0111	0	0	1	1	1
1000	0	1	0	0	0
1001	0	1	0	0	1
1010	1	0	0	0	0
1011	1	0	0	0	1
1100	1	0	0	1	0
1101	1	0	0	1	1
1110	1	0	1	0	0
1111	1	0	1	0	1

۲- (۲ نمره) خروجی تابع F را به دست آورید.



XYZ	$D_0D_1D_2D_3$	$E_0E_1E_2E_3$	$I_0I_1I_2I_3$	S_1S_0	F
000	0000	0001	0011	00	0
001	0000	0001	0011	00	0
010	0000	0001	1011	00	1
011	0000	0001	1011	00	1
100	1000	0000	01dd	01	1
101	0100	1001	0000	00	0
110	0010	0101	1001	00	1
111	0011	0010	1010	10	1

پاسخ:

در جدول بالا D_i ها خروجی های دیکودر، E_i ها ورودی های انکودر، I_i ها ورودی های مالتی پلکسر و S_i ها خطوط انتخاب مالتی پلکسر هستند. بنابراین تابع F به دست می آید.

$$F = x + \bar{y}z$$

۳- (۲ نمره) تابع $F(a,b,c) = a\bar{b} + \bar{a}c$ را با هر کدام از موارد زیر پیاده سازی کنید. در صورت نیاز، می توانید از گیت های پایه استفاده کنید.

الف- 8×1 Mux

ب- 4×1 Mux با خطوط انتخاب ab

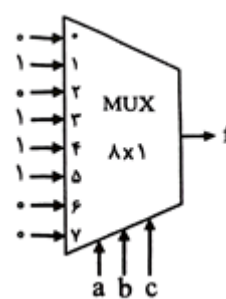
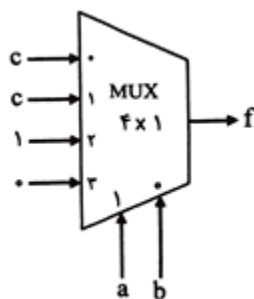
ج- 4×1 Mux با خطوط انتخاب bc

د- 2×1 Mux با خط انتخاب c

پاسخ:

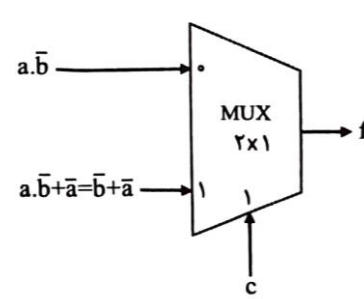
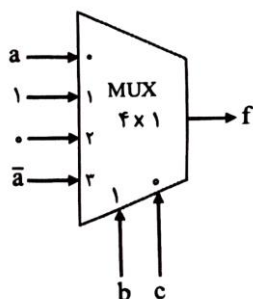
بخش های الف و ب را می توانیم با رسم جدول درستی پاسخ دهیم.

abc	f	
000	0	$I_0=c$
001	1	
010	0	$I_1=c$
011	1	
100	1	$I_2=1$
101	1	
110	0	$I_3=0$
111	0	



برای پاسخ به بخش های ج و د می توانیم جدول درستی را طور دیگری رسم کنیم.

cba	f		
000	0	I ₀ =a	J ₀ =ab'
001	1		
010	0	I ₂ =0	
011	0		
100	1	I ₁ =1	J ₁ =a'+b'
101	1		
110	1	I ₃ =a'	
111	0		



۴- (۲ نمره) بررسی کنید که هر یک از واحدهای زیر یک منطق کامل هست یا خیر.

الف- نیم تفریق کننده

ب- مقایسه کننده دوبیتی

پاسخ:

الف- در نیم تفریق کننده دو خروجی داریم که با توابع زیر بیان می شوند:

$$b(x, y) = x'y, \quad d(x, y) = x \oplus y$$

با هیچکدام از این دو تابع نمی توانیم NOT را بسازیم، بنابراین نیم تفریق کننده منطق کامل نیست.

ب- در مقایسه کننده سه خروجی داریم که با توابع زیر بیان می شوند:

$$E(a, b) = a \odot b \quad L(a, b) = a'b \quad G(a, b) = ab'$$

داریم:

$$E(a, a) = aa + a'a' = 1$$

$$L(a, E(a, a)) = L(a, 1) = a' \cdot 1 = a'$$

$$L(L(a, E(a, a)), b) = L(a', b) = ab$$

بنابراین مقایسه کننده منطق کامل است.

۵- (۲ نمره) با حداقل تعداد مالتی پلکسر 2×1 ، یک مالتی پلکسر 8×1 بسازید. سپس بررسی کنید برای ساخت یک

مالتی پلکسر 64×1 به چه تعداد مالتی پلکسر 2×1 نیاز است.

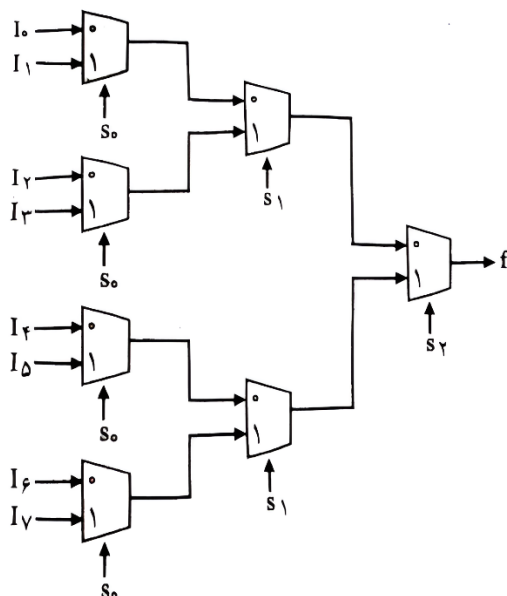
پاسخ:

یک مالتی پلکسر 8×1 دارای ۸ ورودی است. پس باید ۴ تا یک مالتی پلکسر 2×1 زیر هم قرار دهیم که ۸ تا

ورودی را دریافت کنند. این چهار مالتی پلکسر چهار خروجی دارند که به کمک دو مالتی پلکسر 2×1 دیگر تبدیل

به دو خروجی و سپس با کمک یک مالتی پلکسر 2×1 دیگر تبدیل به یک خروجی می شود.

با همین استدلال، برای ساخت یک مالتی پلکسر 64×1 به ۶۳ مالتی پلکسر 2×1 زیر نیاز است.

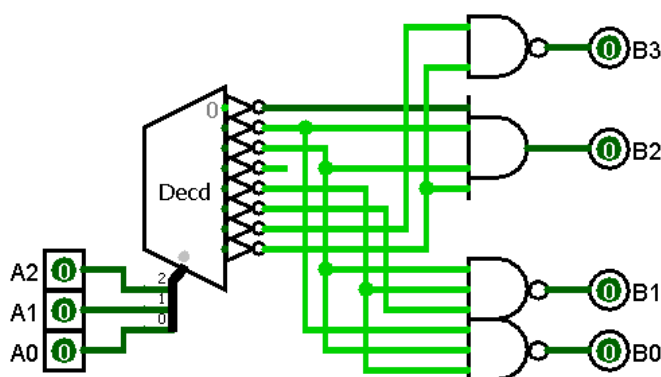


۶- (۲ نمره) به کمک یک دیکودر ۳ به ۸ با خروجی‌های active-low و کمترین گیت اضافه با کمترین تعداد ورودی مداری بسازید که یک عدد ۳ بیتی دریافت کند و حاصل جمع مقسوم‌علیه‌هایش را تولید کند.

پاسخ:

A ₂	A ₁	A ₀		B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	3	0	0	1	1
0	1	1	4	0	1	0	0
1	0	0	7	0	1	1	1
1	0	1	6	0	1	1	0
1	1	0	12	1	1	0	0
1	1	1	8	1	0	0	0

چون خروجی دیکودر active-low است، برای ساخت خروجی‌ها، اگر تعداد صفرها کمتر بود، خروجی‌های متناظر را صفر را AND می‌کنیم و اگر تعداد یک‌ها کمتر بود، خروجی‌های متناظر با یک را NAND می‌کنیم. برای ساخت B2 هم می‌توانیم از گیت AND استفاده کنیم و هم از گیت NAND.



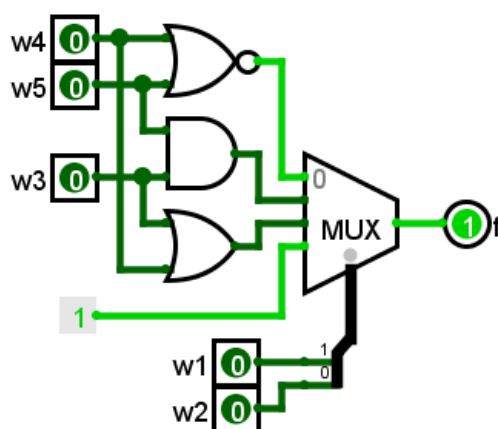
۷- (۲ نمره) تابع زیر را به کمک یک مالتی‌پلکسر ۴ به ۱ و حداقل گیت ممکن پیاده سازی کنید. (هر یک از گیت‌های AND, OR, NAND, NOR, XOR و NOT را بدون توجه به تعداد ورودی یک گیت محسوب کنید). راهنمایی: متغیرهای ارزش بالا را به ورودی‌های انتخاب مالتی‌پلکسر بدهید.

$$f(w_1, w_2, w_3, w_4, w_5) = \overline{w_1} \overline{w_2} \overline{w_4} \overline{w_5} + w_1 w_2 + w_1 w_3 + w_1 w_4 + w_2 w_3 w_5$$

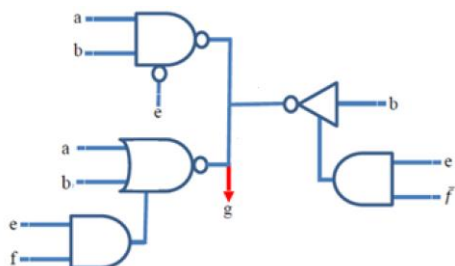
پاسخ:

تابع بالا را می‌توانیم به صورت زیر ساده کنیم تا ورودی‌های مالتی‌پلکسر به دست آید.

$$\begin{aligned} f(w_1, w_2, w_3, w_4, w_5) &= \overline{w_1} \overline{w_2} \overline{w_4} \overline{w_5} + w_1 w_2 + w_1 \overline{w_2} w_3 + w_1 w_2 w_3 + w_1 \overline{w_2} w_4 + w_1 w_2 w_4 + \overline{w_1} w_2 w_3 w_5 + w_1 w_2 w_3 w_4 \\ &= \overline{w_1} \overline{w_2} (\overline{w_4} \overline{w_5}) + w_1 w_2 (1 + w_3 + w_4 + w_3 w_4) + w_1 \overline{w_2} (w_3 + w_4) + \overline{w_1} w_2 (w_3 w_5) \\ &= \overline{w_1} \overline{w_2} (\overline{w_4} \overline{w_5}) + w_1 w_2 (1) + w_1 \overline{w_2} (w_3 + w_4) + \overline{w_1} w_2 (w_3 w_5) \end{aligned}$$



۸- (۳ نمره) تابع g مدار زیر را به شکل ساده شده SOP بنویسید.



پاسخ:

می بینیم که ورودی های فعال ساز بافرهای سه حالته طوری است که هیچوقت دو بافر همزمان فعال نمی شوند. بنابراین می توانیم تابع خروجی g را به صورت زیر بنویسیم و سپس ساده کنیم.

$$g = e'(ab)' + ef(a+b)' + ef'b'$$

این تابع را می توانیم با جدول کارنو یا با قوانین جبر بول ساده کنیم.

ab \ ef	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	0	0	0
10	1	0	0	1

$$g = a'e' + b'f' + b'e' + a'b'$$

۹- (۳ نمره) با استفاده از تعداد کافی مالتی پلکسر 2×1 یک جمع کننده کامل (full adder) بسازید. فرض کنید ورودی های صفر و یک در دسترس هستند، اما هیچ گیت منطقی ندارید.

پاسخ:

حداقل به دو مالتی پلکسر نیاز داریم که خروجی های s و c را طبق روابط زیر بسازیم:

$$c(x, y, z) = xy + yz + xz = z'(xy) + z(x + y)$$

$$s(x, y) = x \oplus y \oplus z = xy'z' + x'yz' + x'y'z + xyz = z'(xy' + x'y) + z(xy + x'y')$$

اما برای ساخت ورودی های هر کدام از این مالتی پلکسرها هم نیاز به مالتی پلکسرهایی داریم که توابع NOT، XOR، AND و OR را بسازند.

بنابراین مدار نهایی به صورت زیر خواهد شد.

