مقدمه

پارسه تاکنون سعی کرده است با ارائه جزوات و برگزاری کلاس‌های حضوری و آزمون‌های آمادگی کنکور راهکار مناسب را برای کسب بهترین نتیجه، در اختیار خانواده خود قرار دهد تا همانند سال‌های قبل، این اعضای خانواده بزرگ پارسه باشند که اکثریت جایگاه‌های کارشناسی ارشد را در دانشگاه‌های سراسری به خود اختصاص دهد.

این جزوه به منظور کمک افزایش آمار قبولی دانشجویان و رسیدن به هدف اصلی پارسه به معنی کسب ۹۰ درصد قبولی در سال ۹۰ طراحی شده است.

این جزوه در ۵ فصل تنظیم شده است: جبر بول و ساده‌سازی توابع – مدارهای منطقی ترکیبی – مدارهای ترتیبی – نمایش اعداد و حافظه‌ای مطالب اصلی درس منطقی در آزمون سراسری را تشکیل می‌دهند. در سال‌های اخیر سوالات مطرح شده از سه فصل اول اکثریت تست‌ها را تشکیل داده است. در پایان هر فصل تعدادی تست به همراه حل تشریحی آورده شده است.

با آروزی موفقیت
سیما سینایی

در جدول ذیل دروس به سرفصلهای مهم آن طبقه بندی شده و مشخص شده است که در هر سال از هر مبحث چند تست سوال شده است و دانشجوی محترم می تواند زمان باقیمانده تا کنکور را با توجه به اهمیت مباحث مدیریت نماید.

	رشته: مهندسی کامپیوتر درس: مدار منطقی							
ردیف	مبحث	۱۳۸۵	۱۳۸۶	۱۳۸۷	۱۳۸۸	۱۳۸۹	مجموع ۵ سال	نسبت از کل
		تعداد تست	تعداد تست	تعداد تست	تعداد تست	تعداد تست		
1	جبربول و ساده سازی توابع بولی	1	0	1	1	1	4	12%
2	مخاطرات (هازارد)	1	2	1	0	1	5	15%
3	مدارهای ترکیبی	1	1	1	1	2	6	18%
4	مدارهای ترتیبی و تحلیل آنها	2	2	5	2	2	13	38%
5	طراحی مدارهای ترتیبی	1	0	0	2	1	4	12%
6	شیوه نمایش اطلاعات	0	0	0	0	0	0	0%
7	حافظه ها	0	1	0	1	0	2	6%
جمع		**6**	**6**	**8**	**7**	**7**	**34**	**100%**

فصل اول

جبر بول و ساده‌سازی توابع بولی

جبر بول

یک شبکه مکمل‌پذیر و توزیع‌پذیر را جبر بول گویند.

اصول جبر بول:

عضو همانی	$a.1=a \quad \& \quad a+0=a$
جابجاپذیری	$a.b=b.a \quad \& \quad a+b=b+a$
شرکت‌پذیری	$a.(b.c)=(a.b).c \quad \& \quad a+(b+c)=(a+b)+c$
توزیع‌پذیری	$a+(b.c)=(a+b).(a+c) \quad \& \quad a.(b+c)=a.b+a.c$

اصل همزادی (Duality): اگر یک عبارت بولی درست باشد همزاد یا دوگان آن نیز درست است. باید به جای هر + و . به ترتیب . و + بگـذاریم و به جای هر یک و صفر، صفر و یک قرار دهیم. در به دست آوردن همزاد باید مراقب باشیم جای پرانتزها را تغییر ندهیم.

قضایای اساسی جبر بول:

خودتوانی	$a+a=a \quad \& \quad a.a=a$
عناصر بی‌اثر	$a+0=a \quad \& \quad a.1=a$
متمم متمم	$\overline{\overline{a}}=a$
جذب	$a.(a+b)=a \quad \& \quad a+ab=a$

یادداشت:

.....

.....

.....

.....

دمورگان $\overline{a+b}=\bar{a}.\bar{b}$ & $\overline{a.b}=\bar{a}+\bar{b}$

تراضی $(a+b)(\bar{a}+c)(b+c)=(a+b)(\bar{a}+c)$ & $ab+\bar{a}c+bc=ab+\bar{a}c$

بسط شانون:

اگر تابع f را حول n متغیر آن بسط دهیم، تابع به‌صورت مجموع جملات حاصل‌ضرب n متغیری یا به‌صورت ضرب جملات حاصل‌جمع، بیان خواهد شد. در نتیجه مدار آن به‌صورت دو سطحی $AND-OR$ قابل پیاده‌سازی است که سریع‌ترین مدار ترکیبی است.

$$f(x_1,x_2,...,x_n)=\left[x_1.f(1,x_2,...,x_n)\right]+\left[\bar{x}_1.f(0,x_2,....x_n)\right] \quad (I)$$

$$f(x_1,x_2,...,x_n)=\left[x_1+f(0,x_2,...,x_n)\right].\left[\bar{x}_1+f(1,x_2,....x_n)\right] \quad (II)$$

فرم POS و SOP:

تابع f به فرم SOP (sum of product) می‌باشد، هرگاه جملات آن همگی به‌صورت جمع تعدادی جمله‌ی ضرب باشند.

تابع f به فرم POS (product of sum) می‌باشد، هرگاه جملات آن همگی به‌صورت ضرب تعدادی جمله‌ی جمع باشند.

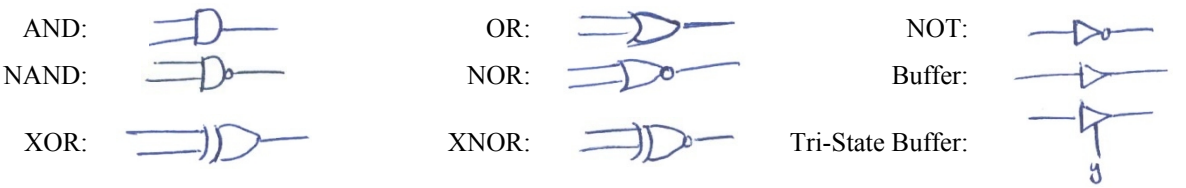
مینترم (Minterm) جمله‌ای است به‌صورت حاصل‌ضرب، که در آن همه‌ی متغیرها یا مکمل آنها دقیقاً یکبار ظاهر باشند. (m_i)

ماکسترم (Maxterm) جمله‌ای است به‌صورت حاصل‌جمع، که در آن همه‌ی متغیرها یا مکمل آنها دقیقاً یکبار ظاهر باشند. (M_i)

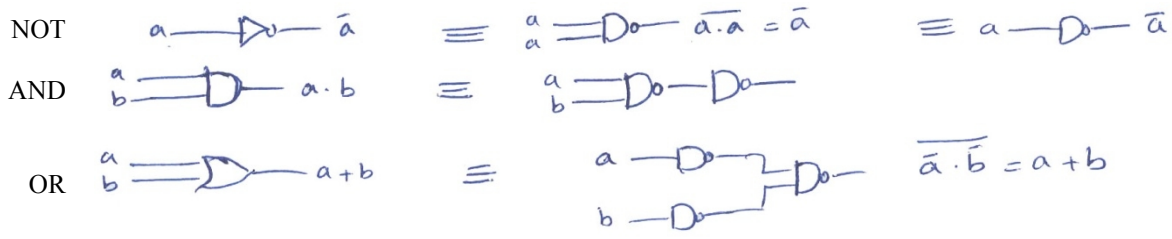
به‌طور کلی برای n متغیر 2^n مینترم و 2^n ماکسترم وجود دارد.

نکته: مینترم‌ها و ماکسترم‌ها مکمل یکدیگرند. $\sum(3,5,6,7)=\pi(0,1,2,4)$

گیت‌های منطقی



با استفاده از گیت NAND می‌توان هر سه عملگر AND و OR و NOT را ساخت. گیت NAND به تنهایی یک گیت کامل است.



یادداشت:

.....

.....

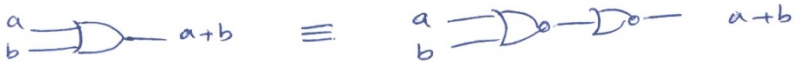
.....

.....

با استفاده از گیت NOR می‌توان هر سه عملگر AND و OR و NOT را ساخت. گیت NOR نیز یک گیت کامل است.

NOT

AND



OR

نکته: مجموعه‌های $\{NAND\},\{NOR\},\{AND, NOT\},\{OR, NOT\}$ کامل هستند.

پیاده‌سازی با گیت NAND

باید ابتدا تابع را به‌صورت SOP ساده کرده و سپس مدار را با استفاده از گیت‌های AND – OR بسازیم و مدار را با کمک تساوی زیر به NAND – NAND تبدیل کنیم.

پیاده‌سازی با گیت NOR

در این روش باید تابع را به‌صورت POS ساده کنیم و سپس آن را با گیت‌های OR – AND بسازیم. سپس خروجی گیت‌های OR و ورودی گیت‌های AND را مکمل کنیم. سپس با کمک رابطه‌ی زیر همه‌ی گیت‌ها را به NOR تبدیل کنیم.

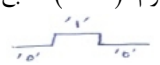
مخاطره (Hazard)

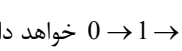
تأخیر جابجایی (Delary) در گیت‌ها باعث بروز تغییرات ناخواسته‌ای در مدارات منطقی می‌گردد که این تغییرات ناخواسته Hazard نام دارد.

انواع Hazard:
- استاتیک
  - سطح یک
  - سطح صفر
- دینامیک

مخاطره استاتیک سطح یک: اینگونه مخاطرات اغلب در مدارهای AND – OR یعنی فرم مینترم (SOP) تابع رخ می‌دهد. اینگونه مدارها دارای خروجی یک هستند اما در یک لحظه (مدت زما بسیار کوتاه) خروجی صفر شده و سپس به حالت 1 برمی‌گردد.

مخاطره استاتیک سطح صفر: اینگونه مخاطرات اغلب در مدارهای OR – AND یعنی فرم ماکسترم (POS) تابع رخ می‌دهد. اینگونه مدارها دارای خروجی صفر هستند اما در یک لحظه خروجی مدار یک شده و سپس به حالت صفر برمی‌گردد.



هازارد دینامیک: اغلب در مدارهای ترتیبی رخ می‌دهد و به این معنی است که خروجی از صفر به یک یا از یک به صفر می‌خواهد تغییر کند ولی تغییرات ناخواسته‌ای دارد مثلاً به جای $1 \to 0$ تغییرات $1 \to 0 \to 1 \to 0$ خواهد داشت یعنی خروجی به صورت  تغییر می‌کند یا به جای اینکه خروجی $0 \to 1$ تغییر کند به‌صورت $0 \to 1 \to 0 \to 1$ یعنی می‌شود.

یادداشت:

.....

.....

.....

.....

نکته: اگر در جدول کارنو دو مینترم یا ماکسترم مجاور هم در یک دسته نباشند آنگاه مدار قطعاً دارای مخاطره است.

نکته: زمان Hazard برابر با اختلاف زمان دو مسیر متفاوتی است که از یک متغیر روی خروجی گذاشته می‌شود.

نکته: در روش کوئین مک‌کلاسکی اگر PIهای تابع را پیدا کنید، برای آنکه تابع مخاطره نداشته باشد باید شامل همه‌ی PIها باشد یعنی نیازی به کشیدن جدول پوشش و انتخاب حداقل PIها نیست.

ساده‌سازی توابع

توابع بولی را می‌توان به یکی از سه روش جبری، جدول کارنو و رویه‌ی کوئین مک کلاسک ساده نمود.

روش جبری:

با استفاده از قوانینی جبر بول می‌توان توابع ترکیبی را ساده نمود. هدف ایجاد مدارهای دو سطحی است که از جنبه‌هایی بهنیه باشد. ضوابط متعددی برای بهینگی وجود دارد که ما دو ضابطه‌ی زیر را در نظر می‌گیریم.

۱. مدار دو سطحی که در آن تعداد کل ورودی به Gateها حداقل باشد.

۲. مدار دو سطحی که در آن تعداد کل گیت‌ها حداقل باشد و بین تمام مدارهایی که از نظر تعداد گیت‌ها مینیمم باشند آن یکی که دارای کم‌ترین ورودی باشد.

روش جدول کارنو:

جدولی کارنو همان جدول درستی است ولی خانه‌ها طوری کنار هم قرار می‌گیرند که فقط در یک بیت اختلاف داشته باشند. در واقع خانه‌ها طبق کدگری چیده می‌شوند. جدول کارنو برای $4,3,2$ متغیر در زیر آمده است. هر تابع برنولی را می‌توان با درج نقاط 1 و نامشخص تابع در جدول کارنو نمایش داد.

x_1 / x_2	0	1
0	m_0	m_2
1	m_1	m_3

x_1,x_2 / x_3	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

x_1,x_2 / x_3,x_4	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

برای ساده‌سازی تابع با جدول کارنو، مینترم‌های تابع را در جدول قرار می‌دهیم سپس یک‌های مجاور هم را دسته‌بندی می‌کنیم. تعداد یک‌های هر دسته باید توانی از 2 باشد.

باید تمام یک‌ها را دسته‌بندی کنیم طوری‌که تعداد دسته‌ها حداقل باشد و تعداد یک‌های داخل هر دسته حداکثر باشد. اگر بخواهیم تابع به‌صورت POS ساده‌کنیم، می‌توان صفرهای جدول را دسته‌بندی کرده، مکمل تابع یعنی $\overline{f}$ را به دست آورد. سپس از مکمل تابع، مکمل بگیریم. مطابق قانون دمورگان تابع به‌صورت POS به دست می‌آید.

یادداشت:

.....

.....

.....

.....

حالات بی‌اهمیت: حالاتی هستند که برای ما مهم نیستند و در جدول کارنو هم می‌توان این حالات را یک فرض کرد و هم صفر. این حالات در جدول کارنو با علامت x یا d یا $-$ مشخص می‌شوند.

روش کوئین مک کلاسی:

۱. در جدولی که ستون‌های آن با متغیرهای تابع یعنی $x_1, x_2, ..., x_n$ نام‌گذاری شده‌اند، نقاط -1 و نامشخص تابع را رده‌بندی می‌کنیم. به‌طوری که رده‌ی s_i شامل نقاط -1 و نامشخص تابع باشد که در ترکیب آن، i تا یک و بقیه صفر می‌باشند.

۲. برای هر $i=1,2,...,n-1$ هر عضو از رده‌ی s_i را با هر عضو از رده‌ی s_{i+1} مقایسه می‌کنیم. در صورتی‌که فقط در یک لفظ متفاوت باشند از ترکیب آنها جمله‌ی حاصلضربی می‌نویسیم که فاقد لفظ متفاوت و شامل لفظ‌های ثابت باشد. آن را در رده‌ی s'_i قرار می‌دهیم و نقاط ترکیب شده را با علامت ✓ مشخص می‌کنیم تا معلوم شوند که prime Implicant نیستند.

۳. عضوی از رده‌ی s'_i با عضوی از رده‌ی s'_{i+1} به دو شرط قابل ترکیب خواهد بود.

اولاً: موضوع don't care آنها یکسان باشد (یعنی هر دو فاقد آن متغیر باشند) .

ثانیاً: فقط در یک لفظ متفاوت باشد.

از ترکیب آنها جمله حاصلضربی می‌نویسیم که فاقد لفظ متفاوت و شامل لفظ‌های ثابت باشد. آن را در رده‌ی s''_i قرار داده و نقاط قابل ترکیب را با علامت ✓ مشخص می‌کنیم. این عمل را تا جایی ادامه می‌دهیم که دیگر نتوان جمله‌ی حاصلضرب جدیدی به دست آورد. از مرحله‌ی اول تا پایان جملات حاصلضربی که علامت ✓ ندارند مجموعه‌ی prime Implicant تابع را تشکیل می‌دهند.

یادداشت:

.....

.....

.....

.....

تست‌های فصل اول

۱ـ تابع g را تا حد امکان ساده کنید.

$g(a,b,c,d,e,f) = a'b'c'def' + a'b'c'def + ab'cde'f' + ab'cde'f + ab'cdef' + ab'cdef$

۱) $ab'cd + b'cdef'$ ۲) $ab'cd + a'b'c'de$

۳) $a'b'de + b'cdef' + a'b'c'de$ ۴) $a'b'de + a'b'c'de + abc'd$

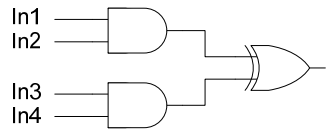
۲ـ در تابع $f(a,b,c,d) = \prod M(2,3,6,7,8,12) \cdot D(1,5,9,13)$ **به ترتیب چند** Prime Implicate **و** Essential Prime Implicate **وجود دارد؟**

۱) ۳ و ۲ ۲) ۳ و ۳ ۳) ۴ و ۲ ۴) ۴ و ۳

۳ـ کدام گزینه Prime Implicant **های تابع** $f(a,b,c,d) = \sum m(1,2,3,9,13,15) + d(5,8,10,11,12,14)$ **را نشان می‌دهد؟**

۱) $a,\bar{c}d,\bar{b}d$ ۲) $a,\bar{c}d,\bar{b}c$ ۳) $a,\bar{b}c,\bar{b}d$ ۴) $a,\bar{c}d,\bar{b}c,\bar{b}d$

۴ـ در شکل زیر ورودی‌های مدار به چه مقادیری متصل شوند که تابع $f(a,b,c,d) = \sum m(10,11,12,13)$ **را پیاده‌سازی کند.**



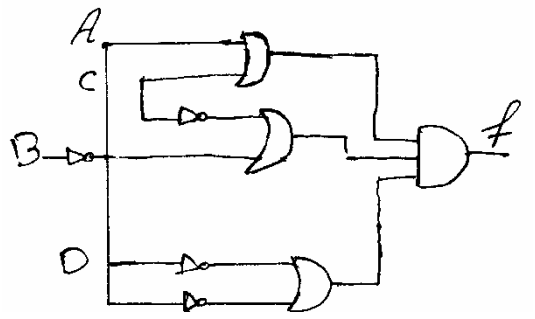
۱) In1 = a, In2 = b, In3 = c, In4 = d ۲) In1 = a, In2 = b, In3 = a, In4 = c

۳) In1 = a, In2 = c, In3 = b, In4 = c ۴) In1 = a, In2 = b, In3 = a, In4 = d

۵ـ در تابع $f(a,b,c,d) = bd + \bar{a}\bar{b}\bar{c} + ac\bar{d}$ **برای کدام تغییرات ورودی، پتانسیل بروز** Hazard **وجود دارد؟**

۱) $0001 \leftrightarrow 0101$ ۲) $0101 \leftrightarrow 1101$ ۳) $1111 \leftrightarrow 1101$ ۴) $1010 \leftrightarrow 1110$

۶ـ در مدار زیر در کدام حالت هازارد وجود دارد و نوع هازارد مورد نظر کدام است.



۱) $ABCD = 0111 \leftrightarrow 0101,0110 \leftrightarrow 0100$ ، هازارد ایستای سطح صفر

۲) $ABCD = 0111 \leftrightarrow 0101,0110 \leftrightarrow 0100$ ، هازارد ایستای سطح یک

۳) $ABCD = 0101 \leftrightarrow 0111,1111 \leftrightarrow 1011$ ، هازارد ایستای سطح صفر

۴) $ABCD = 0101 \leftrightarrow 0111,1111 \leftrightarrow 1011$ ، هازارد ایستای سطح یک

یادداشت:

.....

.....

.....

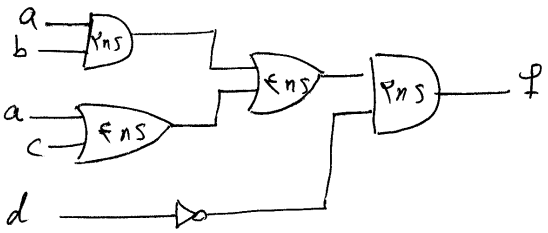
.....

۷ ـ با اضافه کردن کدام گیت به تابع زیر هازارد مدار از بین می‌رود؟

$$F(a,b,c,d,e) = b\bar{c}\bar{e} + \bar{a}c\bar{d}e + bc\bar{d}e + \bar{a}\bar{b}\bar{c}d + a\bar{b}de$$

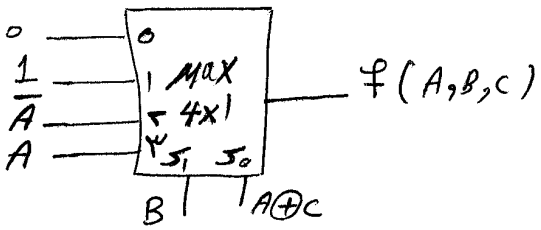
۱) $bcd\bar{e}$ ۲) $\bar{b}\bar{c}de$ ۳) $\bar{b}cde$ ۴) $ab\bar{c}d$

۸ ـ مشخص کنید خروجی مدار زیر پس از چه زمانی 1 می‌شود،؟ فرض‌کنید در زمان صفر ورودی‌های مدار از 0000 به $abcd = 10000$ تغییر می‌کند.



۱) 8nS
۲) 9nS
۳) 10nS
۴) 11nS

۹ ـ خروجی مدار زیر برابر کدام تابع است؟



۱) $\sum(0,2,4,6)$
۲) $\sum(1,2,4,6)$
۳) $\sum(0,4,6)$
۴) $\sum(1,4,6)$

۱۰ ـ ساده‌ترین شکل POS تابعی که اعداد اول بین 0 تا 9 را تشخیص می‌دهد کدام است؟

۱) $\bar{a}(b+c)(\bar{b}+d)$
۲) $\bar{a}(b+c)(\bar{b}+d)(c+d)$
۳) $(b+c)(\bar{b}+d)$
۴) $(b+c)(\bar{b}+d)(\bar{a}+d)(\bar{a}+\bar{b}+\bar{c})$

یادداشت:

.....

.....

.....

.....

پاسخ‌ها

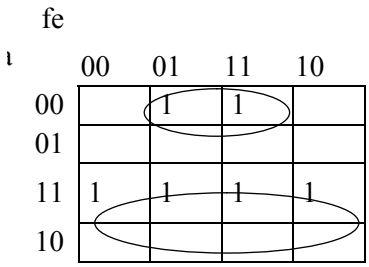
۱ـ گزینه ۲ درست است.

به نظر می رسد که باید از جدول کارنوی 6 متغیری استفاده نمود. ولی در کلیه جملات متغیر b به صورت مکمل و متغیر d به صورت واقعی ظاهر شده‌اند بنابراین:

$$g(a,b,c,d,e,f) = b'd(a'c'ef'+a'c'ef+ace'f'+ace'f+acef'+acef)$$

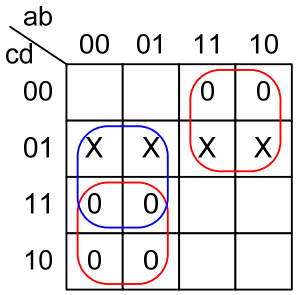
حال داخل پرانتز را بوسیله جدول کارنو 4 متغیری ساده می‌کنیم.

$$F = b'd(ac+a'ce') = ab'cd+a'b'c'de$$



۲ـ گزینه ۱ درست است.

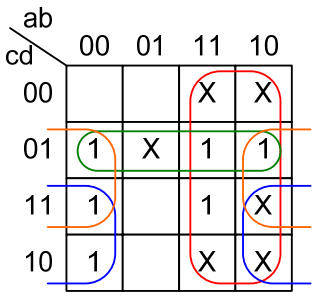
جدول کارنوی زیر را در نظر می‌گیریم:



همانطور که دیده می‌شود، تابع دارای ۳ عدد Implicate Prime است. ولی از این تعداد، طبق تعریف گروه a′d از نوع Essential Prime Implicate نیست.

۳ـ گزینه ۴ درست است.

جدول کارنو را رسم می‌کنیم:



طبق تعریف، چهار گروه داریم که نشان‌دهنده‌ی Prime Implicant ها است. پس گزینه‌ی ۴ درست است.

یادداشت:

.....

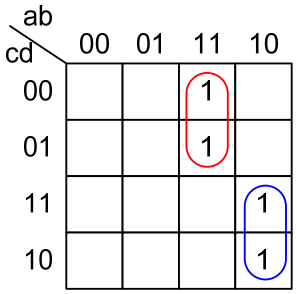
.....

.....

.....

۴ ـ گزینه ۲ درست است.

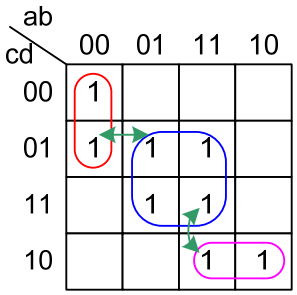
با استفاده از جدول کارنو داریم:



$$f(a,b,c,d)=a\bar{b}c+ab\bar{c}=a(b\oplus c)=ab\oplus ac$$

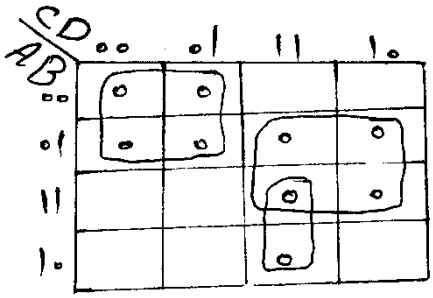
۵ ـ گزینه ۱ درست است.

برای این منظور نخست باید جدول کارنو را رسم کنیم. هر جا که بتوان با تغییر یک ورودی از یک گروه به گروه دیگری تغییر وضعیت داد، به ازاء آن ترکیب ورودی پتانسیل بروز Hazard در مدار وجود دارد.



۶ ـ گزینه ۱ درست است.

$$F(A,B,C,D)=(A+C)(\bar{C}+\bar{B})(\bar{A}+\bar{C}+\bar{D})$$

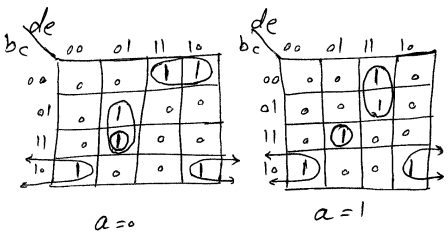


انتقال و انتقال $0101\leftrightarrow0111$

$0110\leftrightarrow0100$ باعث هازارد می‌شود.

در دو حالت مخاطره ایستای سطح صفر داریم.

۷ ـ گزینه ۲ درست است.



یادداشت:

.....

.....

.....

.....

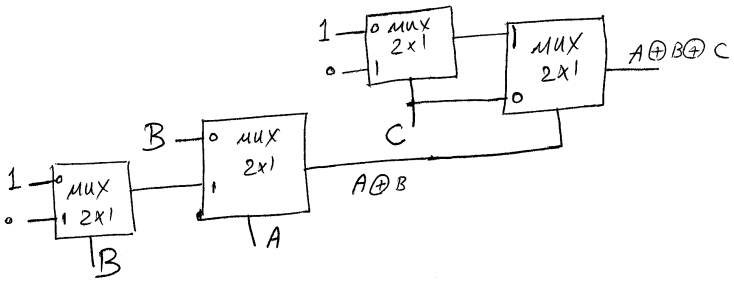
همانطور که در جدول مشخص است تنها 00011 و 10011 دو خانهٔ مجاور از جدول هستند که در یک دسته‌بندی قرار نگرفته‌اند و بنابراین عامل ایجاد مخاطره هستند در نتیجه با اضافه کردن گیت $\bar{b}\bar{c}de$ به تابع مورد نظر مخاطره از بین می‌رود.

۸ ـ گزینه ۳ درست است.

از آنجا که فقط a تغییر کرده است و تمام ورودی‌ها به نحوی تنظیم شده‌اند که با 1 شدن a خروجی یک شود کافی است تأخیر تمام گیت‌ها را که در تغییر خروجی از صفر به 1 پس از تغییر a مؤثرند جمع کنیم. بنابراین پس از $4+4+2=10nS$ خروجی 1 می‌شود.

۹ ـ گزینه ۴ درست است.

برای ایجاد $A\oplus B\oplus C$ از مدار زیر استفاده می‌کنیم که با Mux 2×1 طراحی شده است.



بنابراین به 4 عدد Mux 2×1 نیاز داریم.

۱۰ ـ گزینه ۳ درست است.

این تابع به صورت $f(a,b,c,d)=\sum m(2,3,5,7)+d(10,11,12,13,14,15)$ بیان می‌شود. با توجه به جدول کارنوی زیر پاسخ ۳ صحیح است.

ab / cd	00	01	11	10
00	0	0	X	0
01	0		X	0
11			X	X
10		0	X	X

یادداشت:

.....

.....

.....

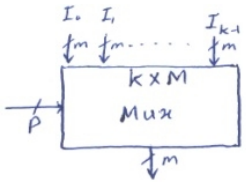
.....

فصل دوم

مدارهای منطقی ترکیبی

تسهیم کننده(*Multiplexer*)

از بین چند منبع ورودی متمایز، یکی را انتخاب و اطلاعات آن را به یک خروجی منفرد و مشترک هدایت می‌کند. باید تساوی $k = 2^p$ برقرار باشد.



نکته: هر تابع ترکیبی n متغیر را می‌توان به وسیله مالتی‌پلکسر ساخت.

روش اول: هر تابع ترکیبی n متغیری را می‌توان به وسیله‌ی یک $2^n \times 1$ Mux انجام داد، مشروط به اینکه n متغیر تابع به عنوان n خط انتخاب $\text{Mux}(\text{select})$ به کار روند. این روش پیاده سازی را Mux نوع -0 می‌نامند، زیرا از صفر متغیر در منابع ورودی استفاده می‌شود.

روش دوم: هر تابع ترکیبی n متغیری را می‌توان به وسیله یک $2^{n-1} \times 1$ Mux انجام داد. مشروط بر این $(n-1)$ متغیر تابع به عنوان $(n-1)$ خط انتخاب وصل شوند و از متغیر باقی‌مانده در ورودی Mux استفاده شود. این روش پیاده‌سازی را Mux نوع -1 می‌نامند. زیرا فقط از یک متغیر در منابع ورودی استفاده می‌شود.

یادداشت:

.....

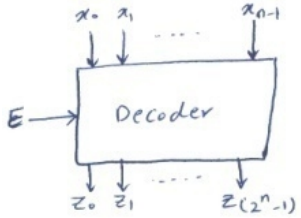
.....

.....

.....

رمزگشا *(Decoder)*

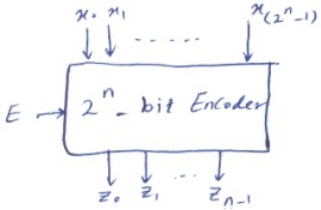
مدار ترکیبی است با n خط ورودی و 2^n خط خروجی که در آن به ازای هریک از ترکیبات ورودی فقط یکی از خروجی‌ها فعال می‌باشد. می‌توان دیکدر را تولیدکننده مینترم نامید.



ورودی (E)اگر یک باشد مدار فعال است و کار عادی خود را انجام می‌دهد ولی اگر $E=0$ هیچ‌یک از خروجی‌ها فعال نیستند. هر مدار ترکیبی با n ورودی و m خروجی را می‌توان به وسیله یک Decoder و m عدد گیت OR طرح نمود.

رمزگذار (Encoder)

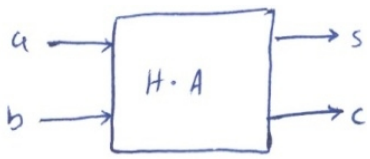
مدار ترکیبی است با 2^n خط ورودی و n خروجی که در آن ترکیبات $1,0$ ظاهر شده در خروجی، آدرس خط ورودی فعال را تولید می‌کند. در واقع رمزگذار، عکس رمزگشا عمل می‌کند. در صورتی‌که بیش از یک خط ورودی به‌طور همزمان فعال باشند، مدار ممکن است آدرس نامعتبر تولید کند. برای رفع این مشکل می‌توان از رمزگذار با حق تقدم (Priority Encoder) استفاده نمود. می‌توان بالاترین اولویت را به آن ورودی نسبت داد که دارای اندیس بزرگ‌تر است.



جمع کننده‌ها

نیم‌جمع‌کننده (Half Adder)

دو بیت ورودی را با هم جمع می‌کند و حاصل‌جمع (s) و رقم نقلی (c_{out}) تولید می‌کند.



a	b	c_{out}	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$c_{out} = a.b$$
$$s = \bar{a}b + a\bar{b}$$

تمام جمع‌کننده (Full Adder)

سه بیت ورودی را با هم جمع می‌کند و حاصل‌جمع (s) و رقم نقلی خروجی (c_{out}) را تولید می‌کند.

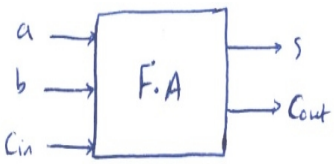
یادداشت:

.....

.....

.....

.....



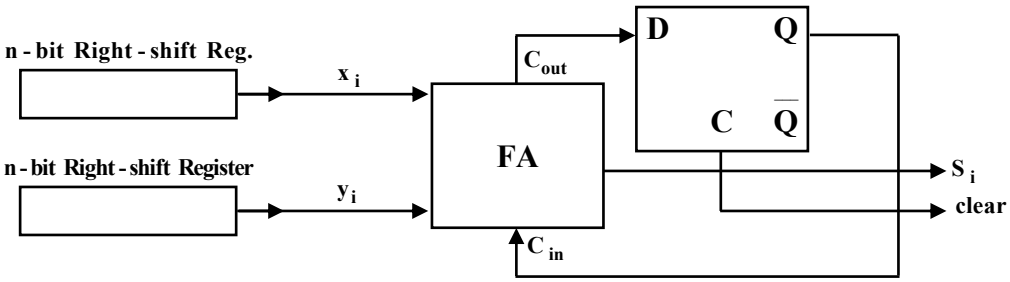
a	b	c_{in}	s	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$c_{out} = a.b + b.c_{in} + a.c_{in}$$
$$s = \bar{a}.\bar{b}.c_{in} + a.\bar{b}.\bar{c}_{in} + \bar{a}b\bar{c}_{in} + a.b.c_{in}$$
$$\rightarrow \quad s = a \oplus b \oplus c_{in}$$

جمع‌کننده سری (serial Adder)

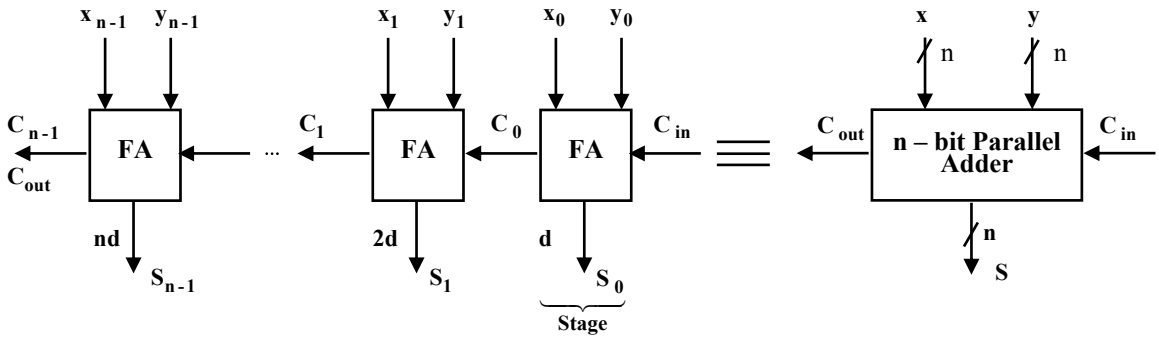
با اتصال یک Full Adder و یک D-flip flop به وجود می‌آید.

در صورتی‌که تأخیر مدار در سطح جمع‌کننده برابر d و تأخیر فلیپ‌فلاپ مساوی D باشد آنگاه حاصل‌جمع دو عدد تک‌بیتی بعد از تـأخیر $(d+D)$ تولید خواهد شد.



جمع‌کننده‌ی موازی (parallel Adder – Ripple Adder)

مدار ترکیبی است که قادر است حاصل جمع دو عدد n بیتی را تولید کند و از اتصال n عدد F.A به‌طور پشت سرهم به وجود می‌آید. به‌طوری کـه رقم نقلی خروجی از یک F.A رقم نقلی ورودی به F.A مجاور خواهد بود.



در بدترین حالت ممکن است رقم نقلی تولید شده از سمت راست‌ترین stage تا سمت چپ‌ترین stage به‌طور موجی پخش شود در این‌صورت مـدار را $n-$Bit Carry Adder Ripple گویند که حاصل‌جمع پس از تأخیر nd تولید خواهد شد.

یادداشت:

.....

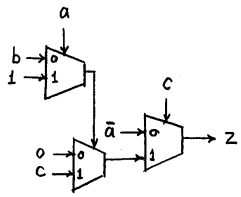
.....

.....

.....

تست‌های فصل دوم

۱ ـ خروجی مدار را به دست آورید؟



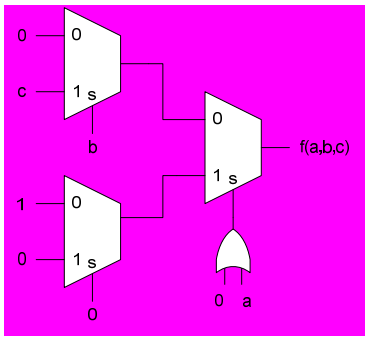
۱) $z = (\bar{a} \oplus c) + bc$

۲) $z = \bar{a}b + (a \oplus b)$

۳) $z = ab + \bar{a}c + a\bar{c}$

۴) $z = \bar{a}\,\bar{c} + c(a + b + c)$

۲ ـ مدار زیر کدام یک از توابع منطقی زیر را پیاده‌سازی می‌کند؟



۱) $f(a,b,c) = ab + bc$

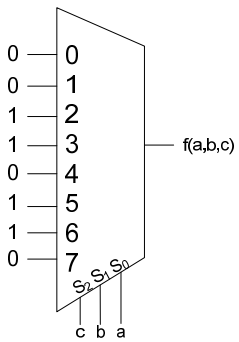
۲) $f(a,b,c) = a + bc$

۳) $f(a,b,c) = a(b + c)$

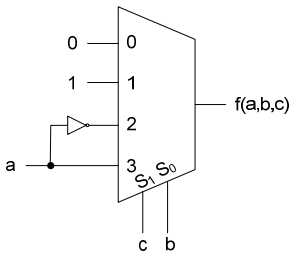
۴) $f(a,b,c) = b + ac$

۳ ـ کدام یک از مدارهای زیر تابع $f(a,b,c) = \sum m(2,3,5,6)$ را پیاده‌سازی می‌کند؟

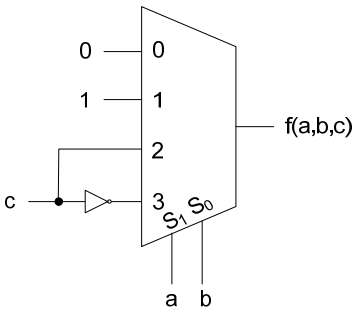
۱)



۲)



۳)



۴) هیچ‌کدام

یادداشت:

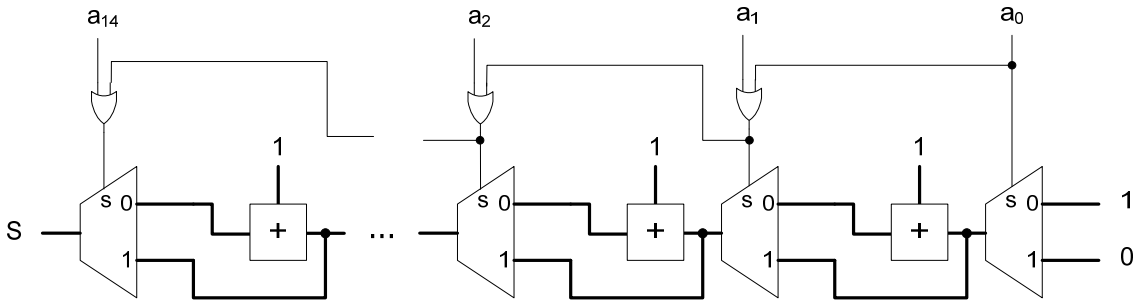
.....

.....

.....

.....

۴ ـ در مدار زیر فرض کنید که خطوط پررنگ نشان‌دهنده‌ی گذرگاه (Bus) چهار بیتی هستند و نیز فرض کنید جمع‌کننده‌ها جمع‌کننده‌ی چهاربیتی باشد. مدار دارای یک ورودی پانزده بیتی a و یک خروجی چهاربیتی S است. این مدار چه عملیاتی را انجام می‌دهد؟



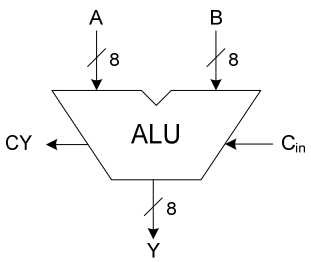
۱) خروجی S نشان‌دهنده‌ی تعداد یک های موجود بر روی ورودی پانزده‌بیتی a است.

۲) خروجی S نشان‌دهنده‌ی تعداد صفر های موجود بر روی ورودی پانزده‌بیتی a است.

۳) خروجی S نشان‌دهنده‌ی تعداد یک های ابتدایی (Leading Ones) ورودی پانزده‌بیتی a است.

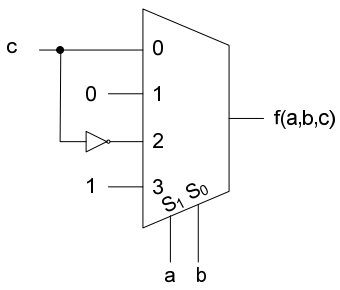
۴) خروجی S نشان‌دهنده‌ی تعداد صفر های ابتدایی (Leading Zeros) ورودی پانزده‌بیتی a است.

۵ ـ واحد محاسباتی زیر را در نظر بگیرید که برای محاسبه‌ی تفریق A - B به صورت داخلی از عملیات + استفاده می‌کند. اگر مقادیر A و B به ترتیب برابر 10110010 و 11001011 باشند، پس از انجام عملیات تفریق مقادیر Y و CY به ترتیب برابر کدام گزینه خواهند بود؟



۱) 0, 11100111 ۲) 1, 11100111 ۳) 0, 00111001 ۴) 1, 00111001

۶ ـ مدار زیر کدام تابع را پیاده‌سازی می‌کند؟



۱) $f(a,b,c)=\sum(1,4,6)$

۲) $f(a,b,c)=\sum(4,6,7)$

۳) $f(a,b,c)=\sum(1,4,6)+d(7)$

۴) $f(a,b,c)=\sum(1,4)+d(7)$

یادداشت:

.....

.....

.....

.....

پاسخ‌ها

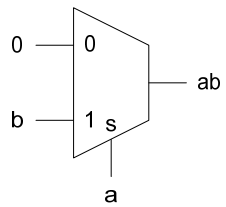
۱ ـ گزینه ۱ درست است.

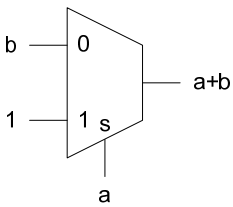
$$z = \bar{a}\,\bar{c} + c.\left[\overline{(a + \bar{a}\,b)}.0 + (a + \bar{a}\,b).c\right] = \bar{a}\,\bar{c} + c.(a + \bar{a}\,b).c = \bar{a}\,\bar{c} + c.(a + \bar{a}\,b)$$

$$= \bar{a}\,\bar{c} + c.\left[(a + \bar{a}).(a + b)\right] = \bar{a}\,\bar{c} + ac + bc = (\bar{a}\,\bar{c} + ac) + bc = (\bar{a} \oplus c) + bc$$

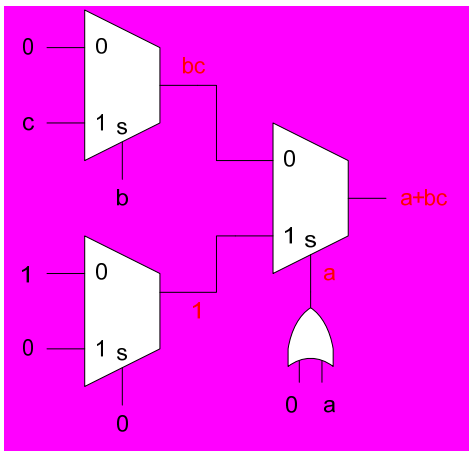
۲ ـ گزینه ۲ درست است.

برای پیاده‌سازی مدار با استفاده از مالتی‌پلکسر می‌توان از الگوهای زیر استفاده نمود.





بنابراین خواهیم داشت:



۳ ـ گزینه ۳ درست است.

برای پیاده‌سازی یک مدار با استفاده از مالتی‌پلکسر جدولی مشابه جدول کارنو می‌سازیم که در آن ساده‌سازی فقط در ستون‌ها انجام می‌شود:

ab / c	I_0 00	I_1 01	I_3 11	I_2 10
0		1	1	
1		1		1

$$I_0 = 0, I_1 = 1, I_2 = c, I_3 = \bar{c}$$

یادداشت:

.....

.....

.....

.....

۴ ـ گزینه ۴ درست است.

در این مدار به محض 1 شدن یکی از بیت‌های ورودی به دلیل استفاده از گیت‌های OR سیگنال‌های انتخاب تمام مالتی‌پلکسرها تا آخر 1 شده و در نتیجه خروجی مالتی‌پلکسر مرحله‌ی قبل را به مرحله‌ی بعدی ارسال می‌کند. در مرحله‌ی آغازین، در صورت 0 بودن بیت a_0، مقدار 1 و در صورت 1 بودن آن، مقدار 0 به مرحله‌ی بعدی ارسال می‌شود که نشان‌دهنده‌ی تعداد 0 ها تا این مرحله است. این امر در مراحل بعدی با افزودن 1 واحد به خروجی مراحل قبلی انجام می‌شود. با مشاهده‌ی اولین 1 در یک بیت ورودی، خروجی مرحله‌ی قبل (تعداد صفرهای تا این مرحله) بدون تغییر به خروجی نهایی مدار ارسال می‌شود. بنابراین این مدار تعداد صفر های ابتدایی (Leading Zeros) ورودی پانزده‌بیتی a را می‌شمارد.

۵ ـ گزینه ۲ درست است.

در حالت تفریق CY نشان‌دهنده‌ی رقم قرضی (Borrow) است. بنابراین باید A را با مکمل ۲ عدد B جمع کرده و سپس CY را NOT کنیم.

	1	0	1	1	0	0	1	0	+
	0	0	1	1	0	1	0	1	
CY=0	1	1	1	0	0	1	1	1	
CY=1									

۶ ـ گزینه ۳ درست است.

برای پیاده‌سازی یک مدار با استفاده از مالتی‌پلکسر جدولی مشابه جدول کارنو می‌سازیم که در آن ساده‌سازی فقط در ستون‌ها انجام می‌شود:

ab / c	I_0 00	I_1 01	I_3 11	I_2 10
0			1	1
1	1		1	

$$I_0 = c, I_1 = 0, I_2 = 1, I_3 = \bar{c}$$

با توجه به این جدول گزینه‌ی ۳ صحیح است که در مین‌ترم شماره‌ی ۷ مقدار تابع Don't Care بوده که در گروه‌بندی شرکت کرده است.

یادداشت:

.....

.....

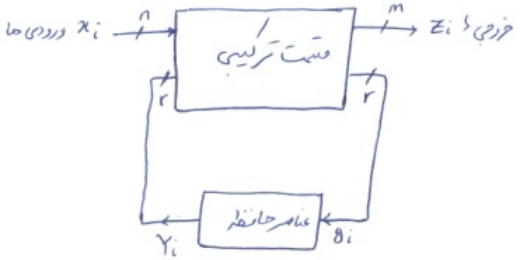
.....

.....

فصل سوم

مدارهای ترتیبی

مدارهای ترتیبی می‌توانند اطلاعت را در خود ذخیره سازند یعنی دارای حافظه می‌باشند بلاک دیاگرام این مدارها به‌صورت زیر است:

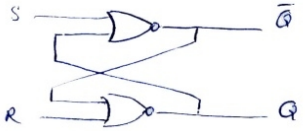


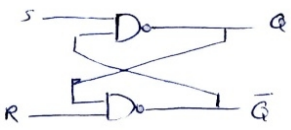
بردار x و z به ترتیب بردار ورودی و خروجی و بردار y و Y به ترتیب حالات فعلی و حالات بعدی هستند.
در مدارهای ترتیبی خروجی‌ها علاوه بر ورودی به متغیرهای حالت نیز وابسته هستند.

عناصر حافظه بر دونوعند: ۱. لچ ۲. فلیپ‌فلاپ

لچ (latch) یا Asynchronous flip flop:

ساده‌ترین مداری که بتواند اطلاعات تک‌بیتی را به‌طور نامحدود از نظر زمان در خود ذخیره‌سازد، می‌توان با اتصال دو NAND یا دو NOR به دست آورد که آن را لچ یا فلیپ‌فلاپ غیرهمگام گویند.





Y \ SR	00	01	11	10
0	0	0	–	1
1	1	0	–	1

y حالت درونی است.

یادداشت:

.....

.....

.....

.....

مدار لچ که به وسیله‌ی دو گیت NOR ایجاد شده دارای دو ورودی محرک R,S و دو خروجی $Q,\bar{Q}$ می‌باشد که مکمل یکدیگرند.

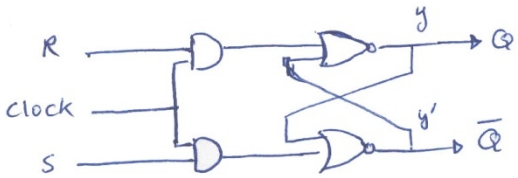
ـ اگر $S=1,R=0$ باشد، آنگاه خروجی Q مساوی با یک خواهد بود که در این‌صورت می‌گوییم مدار Set شده است و با قطع ورودی‌ها یعنی $R=0,S=0$ خروجی Q تغییر نخواهد کرد.
ـ اگر $S=0,R=1$ باشد، آنگاه خروجی Q مساوی با صفر خواهد بود که در این‌صورت می‌گوییم مدار Reset شده است و با قعط منابع ورودی، خروجی مدار تغییر نخواهد یافت.
ـ ورودی‌های $R=1,S=1$ ترکیبات ورودی ممنوعه می‌باشند، زیرا خروجی‌های مدار در نوسان خواهند بود.

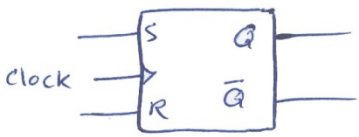
به علت ناهماهنگ بودن تأخیر گیت‌ها، ممکن است در خروجی مدارهای ترکیبی خطای گذرا یا هازارد ایجاد گردد و اگر خروجی این مدار ترکیبی به عنوان ورودی یک فلیپ‌فلاپ یا لچ باشد، گلیچ 1 نادرست ممکن است فلیپ‌فلاپ را اشتباهاً به‌طور دایمی set یا Reset کند و در نتیجه خطای ثابت و دائمی ایجاد می‌گردد.
راه حل استفاده از کلاک است. کلاک را می‌توان به نحوی به کار برد که تغییر در ورودی فلیپ‌فلاپ فقط در مدت $clock=1$ اتفاق بیفتد و در نتیجه خطای گذرا در خروجی مدارهای ترکیبی در رفتارها مدار ترتیبی اثر منفی نگذارد و خطای دائمی تولید کند.

مدارهای ترتیبی که در آن از clock استفاده می‌شود و مدارهای سنکرون یا مدارهای ترتیبی همگام نامیده می‌شود.
مدارهای ترتیبی که در آن از clock استفاده نمی‌شود و مدارهای آسنکرون یا مدارهای ترتیبی غیرهمگام نامیده می‌شود.

انواع فلیپ فلاپ‌ها:

$SR-FF$: این فلیپ‌فلاپ را می‌توان با اضافه کردن clock به مدار لچ قبلی به وجود آورد.



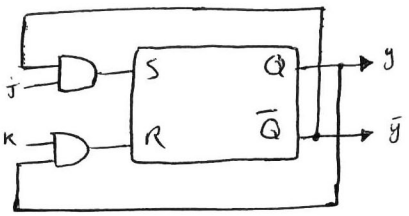


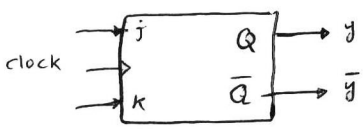
Y \ SR	00	01	11	10
0	1	0	–	1
1	1	0	–	1

$$Y=S+\bar{R}y$$

حالت فعلی با y و حالت بعدی با Y نشان داده شده است.

$JK-FF$: در این نوع فلیپ‌فلاپ برخلاف فلیپ‌فلاپ SR که در آن هر دو ورودی $S=1$ و $R=1$ مجاز نبودند، هر دو ورودی یک مجاز می‌باشند و باعث تغییر حالت فلیپ‌فلاپ از صفر به یک و بالعکس خواهد بود.





یادداشت:

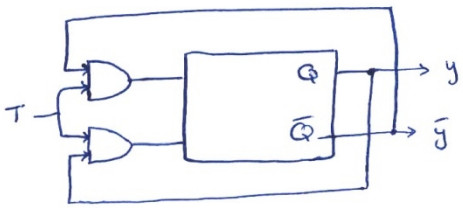
.....
.....
.....
.....

y \ jk	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$$Y = j\bar{y} + ky$$

$y(t)$	$y(t+1)$	$j(t)$	$k(t)$
0	0	0	–
0	1	1	–
1	0	–	1
1	1	–	0

$T-FF$: مهم‌ترین ویژگی فلیپ‌فلاپ T در این است که اگر $T=1$ باشد آنگاه فلیپ‌فلاپ تغییر حالت خواهد داد و اگر $T=0$ باشد فلیپ‌فلاپ در همان حالت قبلی می‌ماند و تغییر حالت نخواهد داد.

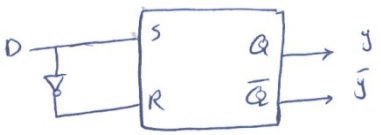


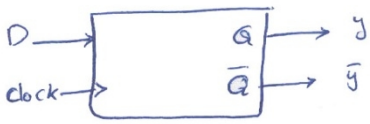
y \ T	0	1
0	0	1
1	1	0

$$y = \bar{T}.y + T.\bar{y} = T \oplus y$$

$y(t)$	$y(t+1)$	$T(t)$
0	0	0
0	1	1
1	0	1
1	1	0

$D-FF$: مهم‌ترین ویژگی فلیپ‌فلاپ D در این است که هر مقداری که به ورودی D نسبت دهیم بعد از یک پریود کلاک حالت بعدی آن را تشکیل خواهد داد.





y \ D	0	1
0	0	1
1	0	1

$$y = D$$

$y(t)$	$y(t+1)$	$D(t)$
0	0	0
0	1	1
1	0	0
1	1	1

نکته: لچ با تغییر ورودی‌های تحریکش تغییر حالت می‌دهد ولی فلیپ‌فلاپ قبل از تغییر حالت منتظر سیگنال ساعت می‌ماند.

خلاصه‌ای از معادله‌ی مشخصه‌ی فلیپ‌فلاپ‌ها:

معادله مشخصه $SR-FF$: $Y = S + \bar{R}y$

معادله مشخصه $jk-FF$: $Y = j.\bar{y} + \bar{k}.y$

معادله مشخصه $D-FF$: $Y = D$

معادله مشخصه $T-FF$: $Y = T \oplus y$

یادداشت:

.....

.....

.....

.....

تأخیر در فلیپ‌فلاپ:

$$T_{FF} = t_s + \max\left(t_n + t_p\right) \xrightarrow{t_p \gg t_n} T_{FF} = t_s + t_p$$

t_{FF} = تأخیر فلیپ‌فلاپ

t_h : مدت زمانی است که ورودی فلیپ فلاپ پس از اعمال پالس ساعت نباید تغییر کند.

t_s : مدت زمانی است که قبل از اعمال پالس ساعت ورودی باید تغییر کند.

t_p : از لحظه‌ی اعمال پالس ساعت تا لحظه‌ی تغییر خروجی

$t_h = (\text{hold})$ زمان نگهداری

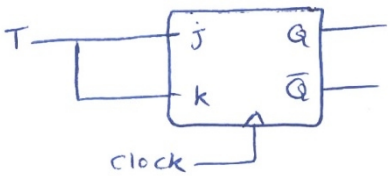
$t_s = (\text{setup})$ زمان راه‌اندازی

$t_p = (\text{progagate})$ تأخیر انتشار

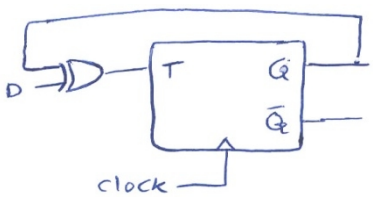
تبدیل فلیپ‌فلاپ‌ها:

ـ تبدیل فلیپ‌فلاپ SR و JK به D:

ـ تبدیل فلیپ‌فلاپ JK به T:

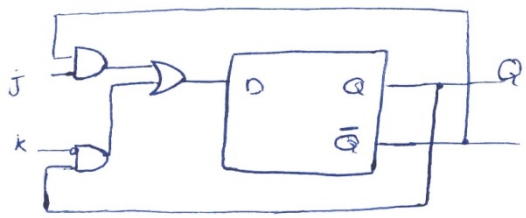


ـ تبدیل فلیپ‌فلاپ T **به** D:



$$T = D \oplus Q$$

ـ تبدیل فلیپ‌فلاپ D به jk:



$$\begin{cases} DFF: Q^* = D \\ JK-FF: Q^* = \bar{K}Q + J\bar{Q} \end{cases}$$

$$\Rightarrow \quad D = \bar{K}Q + J\bar{Q}$$

یادداشت:

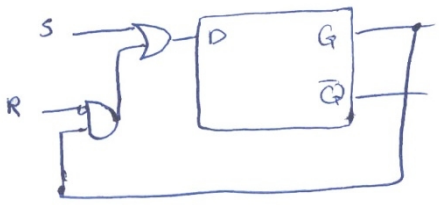
.....

.....

.....

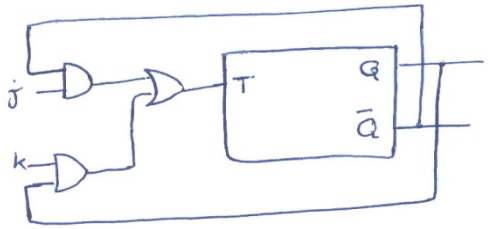
.....

ـ تبدیل فلیپ‌فلاپ D به SR:



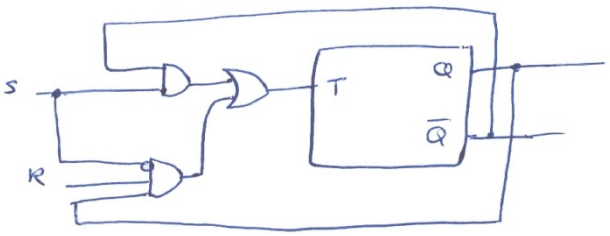
$$\begin{cases} D-FF: Q^* = D \\ SR-FF: Q^* = S+\bar{R}Q \end{cases} \Rightarrow \quad D = S+\bar{R}Q$$

ـ تبدیل فلیپ‌فلاپ T به jk:



$$\begin{cases} T-FF: Q^* = T \oplus Q \\ JK-FF: Q^* = \bar{K}Q + J\bar{Q} \end{cases} \Rightarrow \quad T+Q = \bar{K}Q + J\bar{Q} \quad \Rightarrow \quad \ldots \quad \Rightarrow \quad T = KQ + J\bar{Q}$$

ـ تبدیل فلیپ‌فلاپ T به SR:



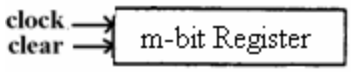
$$\begin{cases} T-FF: Q^* = T \oplus Q \\ SR-FF: Q^* = S+\bar{R}Q \end{cases} \Rightarrow \quad T \oplus Q = S+\bar{R}Q \quad \Rightarrow \quad \ldots \quad \Rightarrow \quad T = S\bar{Q} + \bar{S}RQ$$

ورودی‌های آسنکرون فلیپ‌فلاپ‌ها:

بعضی مواقع برای فلیپ‌فلاپ‌ها ورودی آسنکرون preset و clear را در نظر می‌گیرند. این ورودی‌ها برای این است که بتوان فلیپ‌فلاپ را به‌طور غیرهمگام set و Reset کرد.

ثبات‌ها:

ثبات m بیتی مجموعه‌ی مرتبی است از m عدد فلیپ‌فلاپ که برای ذخیره کردن یک کلمه‌ی m بیتی به کار می‌رود که در آن هر بیت از کلمه در یک FF ذخیره می‌شود. سیگنال‌های clear و clock در همه‌ی فلیپ‌فلاپ‌ها مشترک خواهد بود.



ثبات‌ها دو نوعند:

۱. storage Register که فقط قادر به ذخیره کرده اطلاعات هستند.

۲. Functional Register که علاوه بر ذخیره نمودن اطلاعات، قادرند روی محتوای خود، محاسباتی نیز انجام دهند.

با توجه به روش ورود و خروج اطلاعات به چهار دسته تقسیم می‌شوند:

یادداشت:

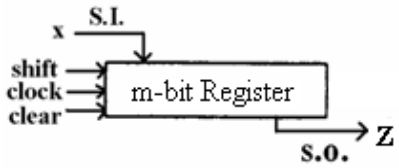
.....

.....

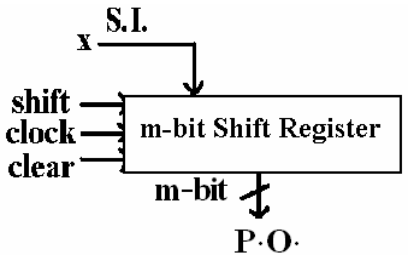
.....

.....

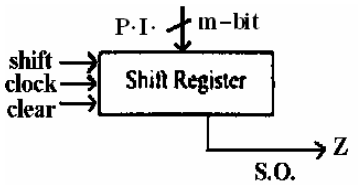
۱) serial input – serial output: در این ثبات اطلاعات به‌صورت $1-bit$ در هر پریود کلاک از یک سو وارد و از سوی دیگر خارج می‌شود بنـابراین باید از نوع عملیاتی باشند.



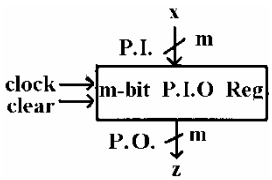
۲) serial input-parallel output: در این ثبات‌ها اطلاعات به‌صورت $1-bit$ در هر پریود کلاک وارد ولی کلیه بیت‌های کلمه‌ی m بیتی بـه‌صـورت موازی و هم‌زمان خارج می‌شوند. بنابراین از نوع عملیاتی هستند.



۳) parallel input – serial output : در این ثبات‌ها کلیه‌ی بیت‌های یک کلمه m بیتی به‌صورت همزمان در آن ذخیره می‌شوند ولـی محتـوی m بیتی آن باید بیت به بیت به خارج آن انتقال یابد. بنابراین از نوع عملیاتی هستند.



۴) parallel input – parallel output : کلیه‌ی بیت‌ها یک کلمه‌ی m بیتی به‌صورت همزمان وارد و به‌طور همزمان از ثبات خارج می‌شوند بنـابراین از نوع storage می‌باشند.



طراحی مدارهای ترتیبی سنکرون:

رویه‌ای برای طراحی مدارهای ترتیبی سنکرون:

۱. از توصیف لفظی مسأله جدول حالت (state table) را به دست می‌آوریم.

۲. به حالت جدول کد نسبت داده و در جدول درج می‌کنیم. در این مرحله جدول انتقال حالات (states transition) به دست می‌آید.

۳. نوع و تعداد فلیپ‌فلاپ‌ها را مشخص می‌کنیم. مثلاً اگر n حالت متمایز وجود داشته باشد آنگاه تعداد فلیپ‌فلاپ‌های لازم برابر $\log_2^n$ می‌باشد.

۴. جدول تحریک یا Exitation table را بدست می‌آوریم.

۵. عبارت‌های بولی برای ورودی‌های محرک و هم‌چنین برای خروجی‌های مدار را به دست می‌آوریم.

۶. پیاده‌سازی مدار (طراحی مدار) را انجام می‌دهیم.

یادداشت:

.....

.....

.....

.....

در تجزیه و تحلیل مدارهای ترتیبی اصطلاحاتی وجود دارد که باید با آنها آشنا باشید:

حالت درونی (Internal state): هر مدار دارای تعدادی حالت درونی محدود است که منظور از تجزیه و تحلیل مدارهای ترتیبی پیدا کردن تعداد همین حالات درونی است.

حالت فعلی (present state) : حالتی از مدار است که قبل از اعمال ورودی‌های خارجی (External input) مدار در آن حالت قرار داد.

حالت بعدی (Next state): حالتی از مدار است که تحت تأثیر ورودی‌های خارجی و حالت فعلی مدار به آن حالت انتقال خواهد یافت. بنابراین Next state تابعی از External Input و Present state است.

جدول حالات (state table): گرافی که در آن هر حالت (state) را با یک دایره نشان می‌دهند و هر انتقال از یک حالت به حالت دیگر را با یک یال نشان می‌دهند. و بیت‌های نوشته شده بر روی یال‌ها به ترتیب ورودی‌ها و خروجی‌ها را نشان می‌دهند.

تخصیص حالت (state Assignment): مرحله‌ای است که در آن حالت‌های یک وسیله‌ی فیزیکی مانند فلیپ‌فلاپ به حالت‌های درونی مدار اختصاص داده شده است.

جدول انتقال حالات (Transition or state transition) : با استفاده از تخصیص حالات و نوع فلیپ‌فلاپ جدول انتقال به دست می‌آید که در آن درایه هر خانه فقط Next state را نشان می‌دهد.

جدول تحریک (Exitation table) : با استفاده از جدول حالات و نوع فلیپ‌فلاپ می‌توان جدول تحریک را به دست آورد و در صورتی که نوع فلیپ‌فلاپ انتخاب شده $D-FF$ باشد، آنگاه taransition table همان Exitation table خواهد بود.

کاهش حالات:

کاهش تعداد فلیپ‌فلاپ‌ها در مدارات ترتیبی به مساله کاهش حالات برمی‌گردد. برای این کار ابتدا باید جدول حالت را تشکیل دهیم. در حالت کلی دو حالت را هم‌ارز می‌نامیم که با اعمال ورودی و مشاهده خروجی نتوانیم تعیین کنیم که مدار در کدامیک از این دو حالت است. اگر این شرط به ازای تمام رشته‌های ورودی برقرار باشد، یکی از این دو حالت زائد است و می‌توان آن را حذف کرد، بدون اینکه رفتار مدار تغییر کند.

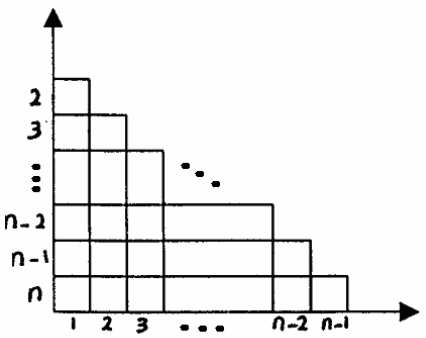
s_i و s_j هم‌ارزند اگر و تنها اگر به ازای هر ورودی ممکن:

۱) خروجی ناشی از حالت s_i با خروجی ناشی از حالت s_j برابر باشد.

۲) حالت‌های بعدی s_i^* و s_j^* هم‌ارز باشند.

برای ساده کردن جدول حالات می‌توان از pair chart یا زوج‌نما ستفاده نمود. برای جدول n حالتی فرم کلی pair chart به‌صورت زیر می‌باشد که در آن برای هر زوج از حالات مانند (q_i, q_j) اگر $i < j$ باشد، آنگاه در pair chart یک خانه واقع در ستون iام و سطر jام وجود خواهد داشت.

برای تشخیص حالات معادل در این جدول، همه‌ی حالات با هم مقایسه می‌شوند. ابتدا حالاتی که خروجی متمایز دارند علامت × می‌خورند به این معنی که با هم معادل نیستند. سپس خانه‌هایی که علامت × نخورده‌اند، مقایسه می‌شوند و به همین ترتیب ادامه پیدا می‌کند.



یادداشت:

.....

.....

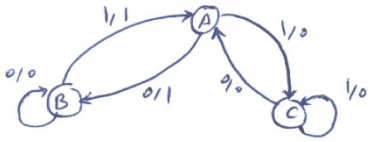
.....

.....

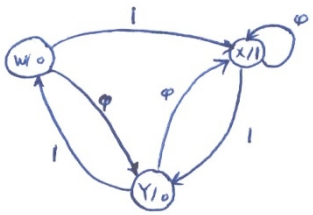
به‌طور کلی دو حالت هم‌ارز هستند، اگر ردیف‌های حالت بعدی آنها مشابه باشند و یا ردیف‌های بعدی‌شان تنها در برگشت به خود متفاوت باشند.

مدارهای ترتیبی همگام به دو گروه تقسیم می‌شوند:

۱) **مدل میلی (MEALY)**: خروجی‌ها توابعی از ورودی‌ها و حالت فعلی هستند.



۲) **مدل مور (MOORE)**: خروجی تنها به ازای حالت فعلی مدار بیان می‌شود.



در مدل مور خروجی‌ها به علت حالت فعلی مربوط می‌شوند، نه حالت بعدی.

نکته‌ای در خصوص تحلیل نمودار میلی: تغییر حالت با پالس ساعت (معمولاً z) رخ می‌دهد، ولی خروجی می‌تواند هر موقعی که ورودی یا حالت تغییر می‌کند، چون تابعی از هر دو است.

نکته در خصوص تحلیل نمودار مور: خروجی فقط به حالت مدار بستگی دارد. با تغییر حالت، خروجی هم تغییر می‌کند یعنی خروجی فقط هنگام تغییر حالت تغییر می‌کند.

در مدل مور علی‌رغم تغییرات ورودی، خروجی پایدار می‌ماند، برخلاف آن چیزی که در مدل میلی رخ می‌دهد. لذا تغییرات ورودی باعث ایجاد گلیچ‌های ناخواسته در خروجی مدل مور نمی‌شود.

یادداشت:

.....

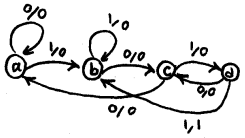
.....

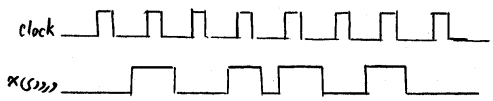
.....

.....

تست‌های فصل سوم

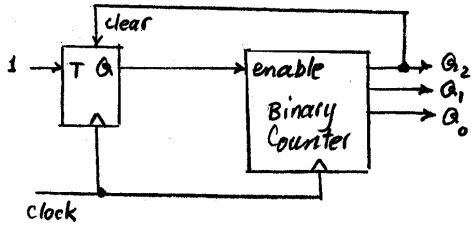
۱ ـ با فرض حالت اولیه a ، مدار در چند نقطه Glitch دارد؟





۱) یک ۲) دو ۳) سه ۴) فاقد Glitch است.

۲ ـ در شروع بکار T – FF و شمارنده مقدار صفر دارند. خروجی شمارنده مطابق کدام گزینه تغییر می‌کند؟



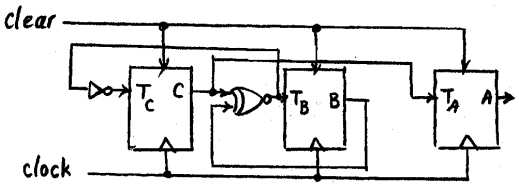
۱) $0 \to 2 \to 3 \to 4$

۲) $0 \to 0 \to 1 \to 1 \to 2 \to 2 \to 3 \to 3 \to 4$

۳) $0 \to 1 \to 2 \to 3 \to 4$

۴) $0 \to 0 \to 1 \to 1 \to 2 \to 2 \to 3 \to 3 \to 4$

۳ ـ شمارنده زیر قبل از شروع به شمارش Clear شده است. با اعمال پالس ساعت، سیکل شمارش را به دست آورید (فرض کنید A بیت پائین‌ترین رتبه باشد)



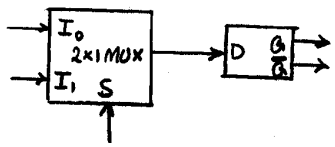
۱) $0 \to 2 \to 4 \to 6$

۲) $0 \to 4 \to 2 \to 6$

۳) $0 \to 6 \to 4 \to 2$

۴) $0 \to 2 \to 6 \to 4$

۴ ـ با استفاده از مدار زیر می‌خواهیم یک JK –FF طرح کنیم. مقادیر منابع ورودی و خط انتخاب MUX را به دست آورید؟



۱) $S = \bar{Q}\ ,\ I_0 = \bar{J}\ ,\ I_1 = K$

۲) $S = Q\ ,\ I_0 = J\ ,\ I_1 = K$

۳) $S = Q\ ,\ I_0 = J\ ,\ I_1 = \bar{K}$

۴) $S = \bar{Q}\ ,\ I_0 = \bar{J}\ ,\ I_1 = \bar{K}$

یادداشت:

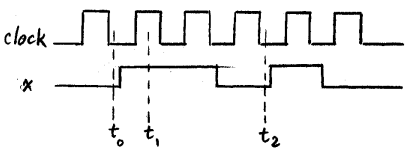
.....

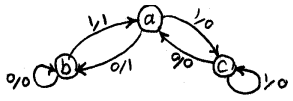
.....

.....

.....

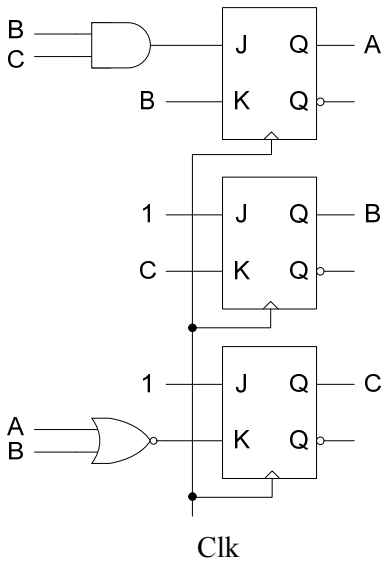
۵ ـ با فرض State Diagram **زیر، حالت شروع** a **و ورودی** $x = 0$ **با توجه به ورودی موجی، خروجی مدار را در لحظات** $t_0 t_1 t_2$ **مشـخص نمائید. (ترتیب از چپ به راست)**





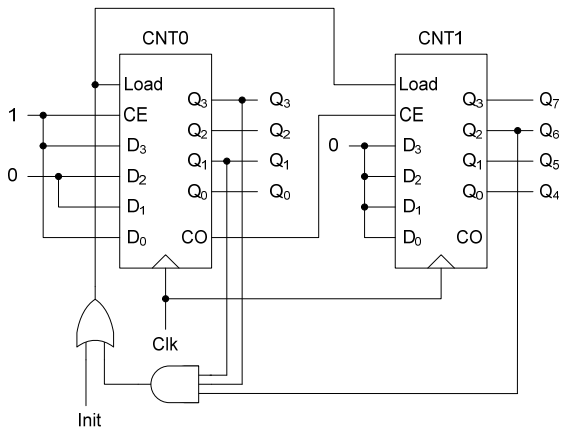
۱) 011 ۲) 111 ۳) 110 ۴) 010

۶ ـ مدار زیر را در نظر بگیرید. با فرض این‌که خروجی مدار مقادیر ABC **باشد، کدام‌یک از گزاره‌های زیر صحیح است؟**



۱) مدار اعداد فرد سه بیتی را می‌شمارد
۲) مدار اعداد زوج سه بیتی را می‌شمارد
۳) مدار اعداد اول سه بیتی را می‌شمارد
۴) هیچ‌کدام

۷ ـ شکل زیر یک شمارنده‌ی 8 **بیتی را نشان می‌دهد که از اتصال دو شمارنده‌ی** 4 **بیتی ساخته شده است. این شمارنده چه اعـدادی را می‌شمارد؟**



۱) 5 تا 69 ۲) 9 تا 69 ۳) 5 تا 74 ۴) 9 تا 74

یادداشت:

.....

.....

.....

.....

۸ ـ Flow Table **زیر را در نظر بگیرید. کدام یک از معادلات زیر ساده‌ترین شکل پیاده‌سازی این جدول را نشان می‌دهد؟**

y_1y_2		x_1x_2 00	01	11	10
00	a	ⓐ /0	b/-	ⓐ /0	d/-
01	b	a/-	ⓑ /1	ⓑ /1	c/-
11	c	d/-	ⓒ /0	ⓒ /0	ⓒ /1
10	d	ⓓ /0	c/-	c/-	ⓓ /0

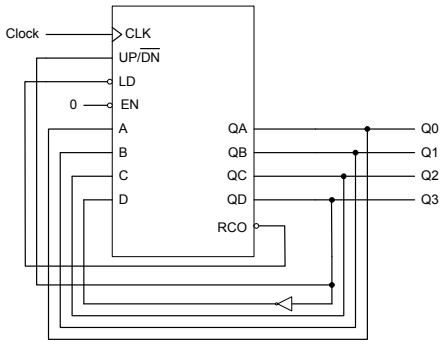
۱) $Y_1 = x_1\overline{x}_2 + y_1$
$Y_2 = \overline{x}_1x_2 + x_1y_2 + x_2y_1$
$Z = \overline{y}_1y_2 + \overline{x}_2y_2$

۲) $Y_1 = x_1\overline{x}_2 + y_1$
$Y_2 = \overline{x}_1x_2\overline{y}_1 + \overline{x}_1y_2 + x_2y_1$
$Z = \overline{y}_1\overline{y}_2 + \overline{x}_2y_2$

۳) $Y_1 = x_1\overline{x}_2\overline{y}_1 + y_1$
$Y_2 = \overline{x}_1x_2 + x_1y_2 + x_2y_1$
$Z = \overline{y}_1y_2 + \overline{x}_2y_2$

۴) هیچ‌کدام

۹ ـ مدار زیر را که با استفاده از یک شمارنده‌ی چهار بیتی ساخته شده، در نظر بگیرید. فرض کنید این شمارنده از حالت 0000 شروع به کار کند، در این‌صورت این شمارنده چه اعدادی را می‌شمارد؟



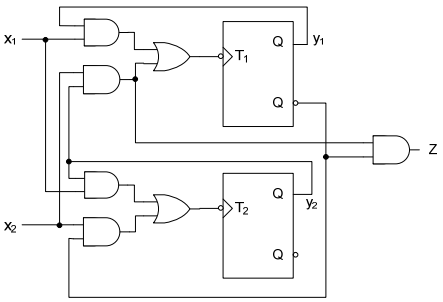
۱) 0, 15, 0, 15, 0, 15, …

۲) 0, 15, 7, 6, 5, 4, 3, 2, 1, 0, 15, 7, 6, …

۳) 0, 15, 8, 9, 10, 11, 12, 13, 14, 15, 8, …

۴) هیچ‌کدام

۱۰ ـ مدار Pulse Mode **زیر چه کاری را انجام می‌دهد؟**



یادداشت:

.....

.....

.....

.....

۱) خروجی مدار با دیدن یک پالس روی x1 و یک پالس روی x2 برابر 1 خواهد شد.

۲) خروجی مدار با دیدن یک پالس روی x1 و دو پالس روی x2 برابر 1 خواهد شد.

۳) خروجی مدار با دیدن دو پالس روی x1 و یک پالس روی x2 برابر 1 خواهد شد.

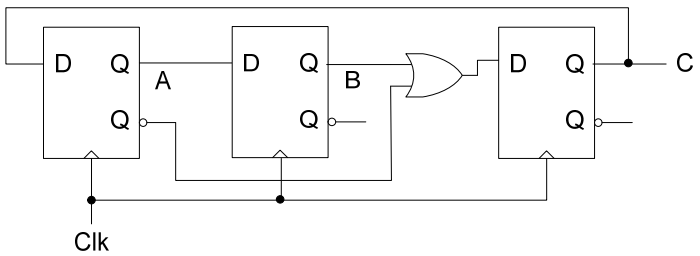
۴) هیچ‌کدام

۱۱ ـ جدول حالت زیر را در نظر بگیرید. کدام یک از گزاره‌های زیر صحیح است؟

PS	NS/Z	
	X=0	X=1
1	1/0	1/0
2	1/1	6/1
3	4/0	7/0
4	1/1	7/0
5	2/0	3/0
6	4/0	5/0
7	2/0	6/0

۱) 3≡5 ۲) 6≡7 ۳) 5≡7 ۴) 1≡5

۱۲ ـ مدار زیر در نظر بگیرید. کدام یک از گزینه‌های زیر صحیح نیست؟



۱) این مدار سیکل 3, 4, 2, 1, 5, 6 را می‌شمارد.

۲) در صورتی که در شروع کار همه‌ی فلیپ‌فلاپ‌ها Clear شوند، مدار می‌تواند به سیکل شمارشی خود ادامه دهد.

۳) در صورتی که در شروع کار همه‌ی فلیپ‌فلاپ‌ها Preset شوند، مدار برای همیشه در همین حالت باقی می‌ماند.

۴) مدار Self Starting است.

۱۳ ـ در جدول حالت زیر Maximal Compatible ها کدامند؟

	X_2X_1			
	00	01	10	11
a	-/-	f/-	a/0	b/-
b	g/-	-/-	c/-	b/1
c	-/-	h/-	c/1	d/-
d	e/-	-/-	a/-	d/0
e	e/0	f/-	-/-	d/-
f	e/-	f/0	a/-	-/-
g	g/1	h/-	-/-	b/-
h	g/-	h/1	c/-	-/-

یادداشت:

.....

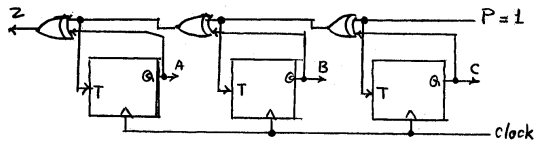
.....

.....

.....

۱) (a, f), c, (d, e), (b, g, h)
۲) a, c, (d, e, f), (b, g, h)
۳) a, c, h, (b, g), (d, e, f)
۴) a, c, e, (d, f), (b, g, h)

۱۴ ـ اگر حالت اولیه مدار $ABC = 000$ باشد، چهار حالت بعدی مدار کدام خواهد بود؟



۱) $000 \rightarrow 110 \rightarrow 000 \rightarrow 001$
۲) $000 \rightarrow 111 \rightarrow 010 \rightarrow 001$
۳) $000 \rightarrow 111 \rightarrow 000 \rightarrow 001$
۳) $000 \rightarrow 111 \rightarrow 101 \rightarrow 001$

یادداشت:

.....

.....

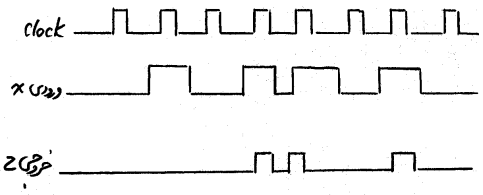
.....

.....

پاسخ‌ها

۱ـ گزینه ۳ درست است.

با توجه به نمودار زمانی زیر و خروجی z، مدار در سه نقطه Glitch دارد.



۲ـ گزینه ۲ درست است.

با توجه به این‌که ورودی محرک T –FF مقدار ثابت یک دارد. بنابراین شمارنده هر حالت شمار را دوبار می‌شمارد و با رسیدن به حالت شمار 4 فلیپ‌فلاپ T به حالت صفر (Clear) می‌رود و در نتیجه عمل شمارش قطع شده و شمارنده در همان حالت 4 باقی می‌ماند.

۳ـ گزینه ۴ درست است.

با توجه به مدار داریم:

P.S.						N.S.		
C	B	A	T_C	T_B	T_A	C	B	A
0	0	0	0	1	0	0	1	0
0	1	0	1	0	0	1	1	0
1	1	0	0	1	0	1	0	0
1	0	0	1	0	0	0	0	0

$T_A = A$

$T_B = \overline{B \oplus C}$

$T_C = B \oplus C$

با توجه به جدول داریم: $0 \to 2 \to 6 \to 4$

۴ـ گزینه ۳ درست است.

J	k	y	Y	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

y \ Jk	00	01	11	10
0			1	1
1	1			1

$Y = D = J\overline{y} + \overline{k}y = J\overline{Q} + \overline{k}Q$

$S = Q\ ,\ I_0 = J\ ,\ I_1 = \overline{K}$

یادداشت:

.....

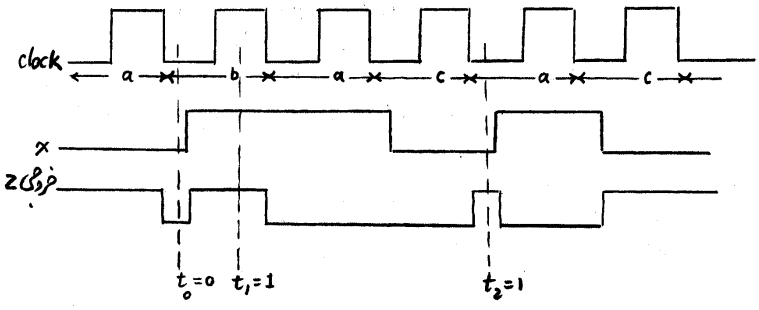
.....

.....

.....

۵ ـ گزینه ۱ درست است.

مدار از نوع Mealy است بنابراین با تغییر در ورودی یا حالت درونی، خروجی مدار ممکن است تغییر یابد.

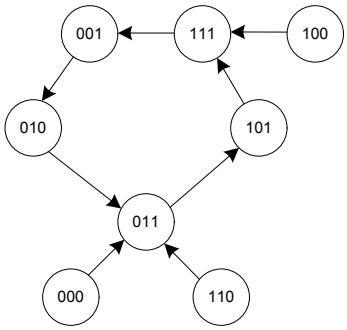


بنابراین در لحظات t_0 و t_2 در خروجی مدار Glitch وجود دارد.

۶ ـ گزینه ۴ درست است.

جدول و نمودار حالت مدار را رسم می‌کنیم. خواهیم داشت:

A	B	C	J_A	K_A	J_B	K_B	J_C	K_C	A^+	B^+	C^+
0	0	0	0	0	1	0	1	1	0	1	1
0	0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	1	1	0	1	0	0	1	1
0	1	1	1	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1
1	1	0	0	1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1	0	0	0	1



با توجه به نمودار حالت مشخص می‌شود که شمارنده اعداد 1, 2, 3, 5, 7 را به صورت متوالی می‌شمارد.

۷ ـ گزینه ۴ درست است.

با توجه به اتصال CE شمارنده‌ی 0 به مقدار 1، این شمارنده همیشه در حال شمارش است. با توجه به اتصال CO شمارنده‌ی 0 بـه CE شـمارنده‌ی 1، پس از هر 16 بار شمارش شمارنده‌ی 0 شمارنده‌ی 1، یک بار می‌شمارد. از طرف دیگر وقتی مقدار هر دو شمارنده به مقـدار 74 = 01001010 رسـید، شمارنده‌ی 0 با مقدار 9=1001 و شمارنده‌ی 1 با مقدار 0=0000 لود می‌شـوند. بنـابراین ایـن شـمارنده سـیکل اعـداد 9 تـا 74 را بـه‌صـورت متـوالی می‌شمارد. سیگنال Init برای مقداردهی اولیه به شمارنده به کار می‌رود.

یادداشت:

.....

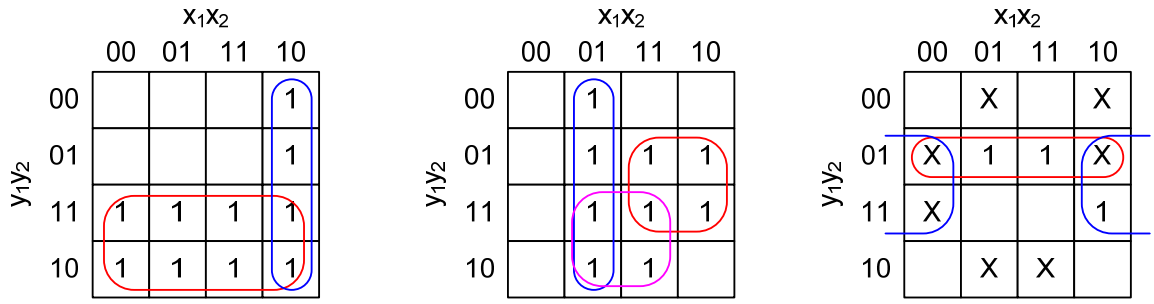
.....

.....

.....

۸ ـ گزینه ۱ درست است.

جدول را برای Y1، Y2 و Z تشکیل می‌دهیم. داریم:



$Y_1 = x_1\bar{x}_2 + y_1$ $\quad Y_2 = \bar{x}_1x_2 + x_1y_2 + x_2y_1$ $\quad Z = \bar{y}_1y_2 + \bar{x}_2y_2$

۹ ـ گزینه ۲ درست است.

با توجه به شکل مدار، مشخص می‌شود هرگاه شمارنده 0xxx باشد، شمارنده رو به پایین و هرگاه 1xxx باشد، شمارنده رو به بالا می‌شمارد. هم‌چنین هرگاه شمارنده مقدار 1111 را داشته باشد، یک پالس 0 (به مدت یک Clock) روی خروجی RCO ایجاد می‌شود که منجر به لود شدن مقدار 0111 در شمارنده می‌شود. داریم:

0000 : شمارش رو به پایین ← 1111 : لود مقدار 0111 ← 0111 : شمارش رو به پایین ← 0110 : شمارش رو به پایین ← 0110 : شمارش رو به پایین ← 0101 : شمارش رو به پایین ← 0100 : شمارش رو به پایین ← 0011 : شمارش رو به پایین ← 0010 : شمارش رو به پایین ← 0010 : شمارش رو به پایین ← 0001 : شمارش رو به پایین ← 0000 : شمارش رو به پایین ← 1111 : لود مقدار 0111 ← 0111 : شمارش رو به پایین ← ...

بنابراین گزینه‌ی ب صحیح است.

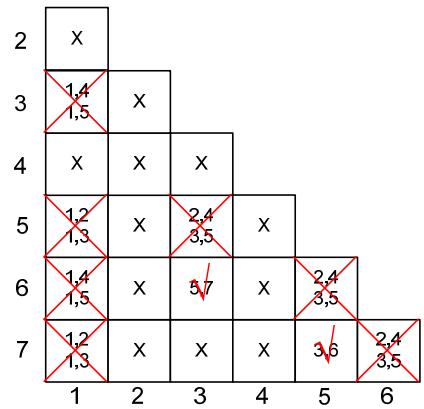
۱۰ ـ گزینه ۲ درست است.

معادلات ورودی فلیپ‌فلاپ‌ها و معادله‌ی خروجی را می‌نویسیم. داریم:

$$T_1 = x_1y_1 + x_2y_2\ ,\quad T_2 = x_1y_2 + x_2\bar{y}_1\ ,\quad Z = x_2\bar{y}_1y_2$$

۱۱ ـ گزینه ۳ درست است.

جدول کاهش تعداد حالت را رسم می‌کنیم. داریم:



یادداشت:

.....

.....

.....

.....

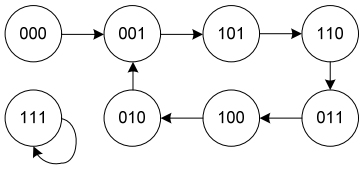
بنابراین حالت‌های معادل عبارتند از:

3≡6, 5≡7

۱۲ ـ گزینه ۴ درست است.

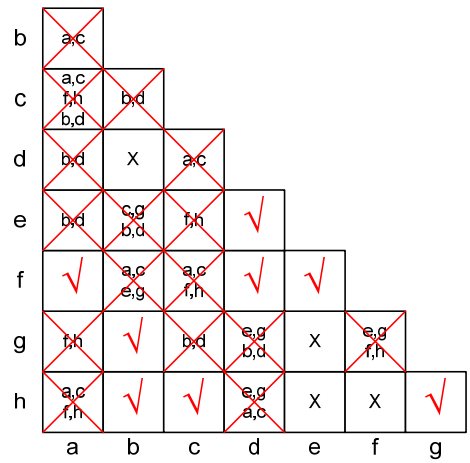
جدول و نمودار حالت این مدار را رسم می‌کنیم:

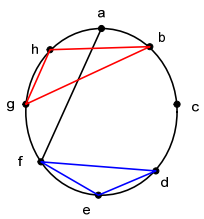
A	B	C	A^+	B^+	C^+
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	1	1	1



۱۳ ـ گزینه ۲ درست است.

جدول کاهش تعداد حالت و نمودار ادغام را رسم می‌کنیم. داریم:





بنابراین Maximal Compatible ها عبارتند از:

a, c, (b, g, h), (d, e, f)

۱۴ ـ گزینه ۲ درست است.

FF اول همیشه toggel می‌شود و بقیه FF ها با توجه به این‌که $p = 1$ می‌باشد وقتی toggel می‌شوند که تعداد زوجی از FF های قبلی در حالت یک باشند.

$000 \rightarrow 111 \rightarrow 010 \rightarrow 001$
ttt tNt Ntt

یادداشت:

.....

.....

.....

.....

فصل چهارم

شیوه‌ی نمایش اعداد

- اعداد
  - باینری
    - علامت مقدار
    - مکمل ۱
    - مکمل ۲
  - دسیمال
    - NBCD
    - Excess-3
    - Gray code

اعداد باینری:

از همان روش ارزش مکانی همانند اعداد دهدهی استفاده می‌شود.

$$\left(b_m b_{m-1}\ldots b_1 b_0 . b_{-1} b_{-2}\ldots b_{-n}\right)_2 = \left(\sum_{i=-n}^{m}\left(b_i \times 2^i\right)\right)_{10} \qquad b_i \in \{0,1\}$$

اعداد علامت‌دار:

۱) روش علامت‌مقدار (sign-magnitude) : بیت سمت چپ عدد نشان‌دهنده‌ی علامت است و هیچ ارزش دیگری ندارد. اگر این بیت 1 باشد عدد منفی و اگر صفر باشد عدد مثبت است.

یادداشت:

.....

.....

.....

.....

۲) روش مکمل ۱ (1's complement) : به‌طور کلی مکمل $r-1$ عدد صحیح N در مبنای r با فرمول زیر محاسبه می‌شود:

$$\text{مکمل } r-1 \text{ عدد } N = r^n - 1 - N$$

نمایش اعداد منفی در سیستم‌های عددنویسی با پایه‌ی r متفاوت:

$$\begin{cases} \text{Binary} & r=2 \Rightarrow 1\text{'s complement} \\ \text{Octal} & r=8 \Rightarrow 7\text{'s complement} \\ \text{Decimal} & r=10 \Rightarrow 9\text{'s complement} \end{cases}$$

الگوریتم: برای به دست آوردن $(r-1)$'s complement کافی است که هر رقم را از $(r-1)$ تفریق می‌کنیم. بنابراین برای پیدا کردن 1's complement اعداد باینری کافی است بیت‌های عدد را یک به یک مکمل کنیم (یعنی صفرها را به یک و یک‌ها را به صفر تبدیل کنیم) .

۳) روش مکمل 2 (2's complement): به‌طور کلی مکمل r عدد صحیح N در مبنای r با فرمول زیر محاسبه می‌شود:

$$\text{مکمل } r \text{ عدد } B = r^n - N$$

نمایش اعداد منفی در سیستم‌های عددنویسی با پایه‌ی (r) متفاوت:

$$\begin{cases} \text{Binary} & r=2 \Rightarrow 1\text{'s complement} \\ \text{Octal} & r=8 \Rightarrow 8\text{'s complement} \\ \text{Decimal} & r=10 \Rightarrow 10\text{'s complement} \end{cases}$$

الگوریتم: برای به دست آوردن r's complement کافی است صفرهای مقدم بدون تغییر، اولین رقتم غیرصفر از r و بقیه از $(r-1)$ تفریق شوند. بنابراین برای به دست آوردن 2's complement، صفرهای مقدم و اولین یک بدون تغییر و بقیه را بیت به بیت مکمل می‌کنیم. در این روش کافی است ابتدا 1's complement را به دست آورده و سپس به بیت سمت راست آن 1 اضافه کنیم.

مقایسه روش‌های فوق:

۱. به دست آوردن مکمل 1 نسبت به مکمل 2 ساده‌تر است.
۲. جمع و تفریق در مکمل 2 نسبت به مکمل 1 و علامت مقدار ساده‌تر است.
۳. در مکمل 1، هم صفر منفی و هم صفر مثبت وجود دارد ولی در مکمل 2 فقط یک صفر وجود دارد.
۴. مکمل 1، بیشتر برای عملیات منطقی و مکمل 2 بیشتر برای عملیات حسابی به کار می‌رود.
۵. دامنه‌ی اعداد باینری قابل نمایش در رجیستر n بیتی:

علامت مقدار: $-\left(2^{n-1}-1\right),\ldots,+\left(2^{n-1}-1\right)$

مکمل 1: $-\left(2^{n-1}-1\right),\ldots,+\left(2^{n-1}-1\right)$

مکمل 2: $-\left(2^{n-1}\right),\ldots,+\left(2^{n-1}-1\right)$

محاسبات دودویی:

$$A-B=A+(-B)=A+2\text{'s}(B)=A+1\text{'s}(B)+1$$

جمع در روش علامت مقدار:

بیت علامت را در عمل جمع شرکت نمی‌دهیم. در صورتی‌که دو عدد هم‌علامت باشند. قدرمطلق آنها را با هم جمع و علامت مشترک را به عنوان علامت نتیجه می‌گذاریم. در صورتی‌که علامت دو عدد مخالف باشد، قدرمطلق عدد کوچک‌تر را از عدد بزرگ‌تر تفریق و سپس نتیجه را هم علامت عدد بزرگ‌تر می‌گذاریم.

یادداشت:

.....

.....

.....

.....

جمع در مکمل 2:

دو عدد را به انضمام بیت علامت با هم جمع می‌کنیم و از رقم نقلی حاصل در موضع بیت علامت صرف‌نظر می‌کنیم.

جمع در مکمل 1:

دو عدد را به انضمام بیت علامت با هم جمع می‌کنیم و در صورت وجود رقم نقلی در موضع بیت علامت آن را با حاصل جمع مرحله‌ی اول دوباره جمع می‌کنیم و در صورت وجود رقم نقلی در مرحله‌ی دوم، از آن صرف‌نظر می‌کنیم.

اعداد دهدهی:

ورودی در سیستم به‌صورت دسیمال و خروجی از آن نیز دسیمال می‌باشد ولی سخت‌افزار در سیستم اعداد باینری کار می‌کند. بنابراین اطلاعات عددی به هنگام ورود به کامپیوتر باید از دسیمال به باینری تبدیل و سپس محاسبات توسط سخت‌افزار انجام و به هنگام خروج، نتایج دوباره از باینری به دسیمال تبدیل گردد.
برای کدگذاری ارقام دهدهی $9,...,2,1,0$ حداقل به چهار بیت نیاز داریم که انتخاب 10 ترکیب از بین 16 ترکیب به طرق مختلف انجام می‌گیرد و در نتیجه کدهای متفاوتی ایجاد می‌شود.

	NBCD 8421	excess – 3	Gray code
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0011
3	0011	0110	0010
4	0100	0111	0110
5	0101	1000	0111
6	0110	1001	0101
7	0111	1010	0100
8	1000	1011	1100
9	1001	1100	1101

کدهای 4 بیتی:

کد NBCD (Natural Binary Coded Decimal)

یک کد وزن‌دار است. متداول‌ترین روش برای نشان دادن ارقام دهدهی است که دو عیب عمده دارد:

۱. در جمع دو رقم در NBCD ممکن است رقم نقلی لازم تولید نشود که در این‌صورت به منظور تصحیح، ۶ واحد باید به آن اضافه شود.

۲. مکمل 9 هر رقم NBCD را نمی‌توان از تبدیل صفرها به یک و یک‌ها به صفر به دست آورد. بنابراین NBCD یک کد خود متمم نیست.

یادداشت:

.....

.....

.....

.....

کد افزونی ۳ یا مازاد ۳ excess − 3

اولین کد با افزودن 3 به NBCD به دست می‌آید که بدون وزن است. دارای خاصیت خودمتممی می‌باشد یعنی مکمل 9 آن را می‌توان از تبدیل صفرها به یک و یک‌ها به صفر به دست آورد. همچنین در جمع دو رقم، رقم نقلی لازم تولید می‌شود، هر چند که نیاز به تصحیح دارد.

کدگری (Gray code)

یکی دیگر از کدهای بدون وزن است که در آن هر کد مجاور تنها در یک بیت تفاوت دارند.
برای تبدیل کد باینری به کدگری، کافی است اولین بیت از سمت چپ را تغییر ندهید و سایر بیت‌ها را با بیت سمت چپ خود XoR کنید.

$(0110)_2 =$ باینری $\Rightarrow$ $(0101) =$ گری

برای تبدیل کدگری به کد باینری، بیت سمت چپ را تغییر نمی‌دهیم و برای یافتن سایر بیت‌ها، بیت متناظر درگری را بایت قبلی در باینری xoR می‌کنیم.

$$\left(g_3 g_2 g_1 g_0\right)_{gray} = \left(b_3, b_2, b_1, b_0\right)_2$$

$$\begin{cases} b_3 = g_3 \\ b_2 = b_3 \oplus g_2 \\ b_1 = b_2 \oplus g_1 \\ b_0 = b_1 \oplus g_0 \end{cases}$$

کدهای با بیش از 4 بیت (کدهای تشخیص و تصحیح و خطا):

کدهای 4 بیتی مانند NBCD و ... کد خطایابی نیستند. یعنی در صورت مخابره پیام از یک مبدأ به یک مقصد درصورتی‌که به دلیل وجود پارازیت بین راه یک بیت تغییر یابد، در مقصد قادر به تشخیص خطا نخواهیم بود. به‌طور کلی برای تشخیص خطا باید کلمات معتبر و نامعبر وجود داشته باشند، برای این منظور نیاز به افزودن تعدادی بیت به کد می‌باشد.
معمول‌ترین روش خطایابی استفاده از بیت توازن است دو نوع توازن داریم: زوج و فرد. در توان زوج، P طوری انتخاب می‌شود که تعداد یک‌های کد زوج می‌شود و در توازن فرد، P طوری انتخاب می‌شود که تعداد یک‌ها فرد شود. کد همینگ روشی برای تصحیح خطا است. در کد همینگ به جای یک بیت، چند بیت اضافه می‌شود. کدهای همینگ انواع مختلف دارند با یک مثال یک حالت را توضیح می‌دهیم.
فرض کنید کلمات NBCD را بخواهیم ارسال کنیم. به منظور تشخیص و تصحیح خطا آن را به‌صورت کدهای جدیدی درمی‌آوریم. به هر کلمه سه بیت اضافه می‌کنیم. این سه بیت را باید در مواضعی متناظر با وزن‌های 2^i قرار داد. (یعنی در مکان 4,2,1) بنابراین کلیه بیت‌هایی که موضع آنها متناظر با توان صحیحی از 2 باشند، مربوط به parity − bit و بقیه مربوط به بیت‌های پیام خواهند بود. اگر بیت‌های توازن را با P_i و پیام واقعی را با bi نشان دهیم داریم:

Bit position	1	2	3	4	5	6	7
Bit Name	P_1	P_2	b_3	P_4	b_5	b_6	b_7

جدول زیر مکان وقوع خطا را مشخص می‌سازد:

یادداشت:

.....

.....

.....

.....

	c_1	c_2	c_3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

با بررسی این جدول متوجه می‌شویم:

اگر خطا در مکان‌های $7,5,3,1$ رخ داده باشد، آنگاه $c_3 = 1$ و در غیر این‌صورت $c_3 = 0$ خواهد بود.

اگر خطا در مکان‌های $7,6,3,2$ رخ داده باشد، آنگاه $c_2 = 1$ و در غیر این‌صورت $c_2 = 0$ خواهد بود.

اگر خطا در مکان‌های $7,6,5,4$ رخ داده باشد، آنگاه $c_1 = 1$ و در غیر این‌صورت $c_1 = 0$ خواهد بود.

بنابراین باید کد چنان بنا گردد که بیت‌های توازن روی بیت‌های پیام کنترل زیر را انجام دهد.

- P_1 باید طوری انتخاب شود که روی بیت‌های مواضع $7,5,3,1$ دارای توان زوج باشد.
- P_2 باید طوری انتخاب شود که روی بیت‌های مواضع $7,6,3,2$ دارای توان زوج باشد.
- P_3 باید طوری انتخاب شود که روی بیت‌های مواضع $7,6,5,4$ دارای توان زوج باشد.

یادداشت:

.....

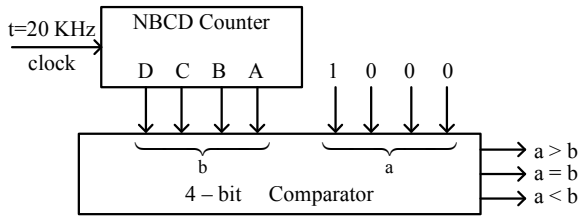
.....

.....

.....

تست‌های فصل چهارم

۱ ـ خروجی مدار مقایسه‌کننده زیر Active High است زمان Low بودن خروجی L را به دست آورید؟



۱) $400\mu sec$

۲) $200\ \mu sec$

۳) $100\mu sec$

۴) $50\mu sec$

۱ ـ گزینه ۱ درست است.

دوره تناوب پالس ساعت برابر است با: $T=\frac{1}{20}KHz=50\mu sec$

صفر بودن خروجی L در مدت برقرار نبودن a>b اتفاق می‌افتد. با توجه به اینکه NBCD Counter از صفر تا 9 را می شمارد. بنابراین مدت صفر بودن L برابر است با زمان هشت پالس ساعت یا $50\times 8=400\mu sec$

یادداشت:

.....

.....

.....

.....

فصل پنجم

حافظه‌ها

آرایه‌های منطقی (قطعات منطقی برنامه‌پذیر): مدارهای ترکیبی دو سطحی AND – OR متناظر با مجموع حاصل‌ضرب‌ها هستند که به‌صورت مدارهای مجتمع با مقیاس بزرگ و خیلی بزرگ (LSI , VLSI) موجودند و معمولاً به سه صورت تجاری وجود دارند:

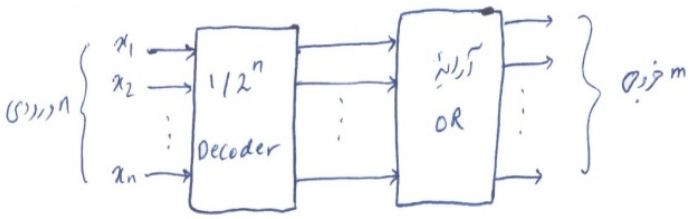
۱. Read Only Memory (ROM) : که در آن ورودی گیت AND ثابت ولی ورودی‌های گیت OR قابل برنامه‌ریزی است.

۲. Programmable Logic Array (PLA) : که در آن هم ورودی‌های آرایه گیت AND و هم ورودی‌های گیت OR قابل برنامه‌ریزی هستند.

۳. Programmable Array Logic (PAL) : که در آن ورودی‌های آرایه گیت AND قابل برنامه‌ریزی ولی ورودی‌های آرایه گیت OR ثابت هستند. (برخلاف ROM)

ROM:

بلوک دیاگرام یک ROM که دارای n خط ورودی و m خط خروجی می‌باشد در زیر نشان داده شده است:



هریک از ترکیبات ورودی یک آدرس نامیده می‌شود.

هریک از ترکیبات خروجی یک word نامیده می‌شوند.

تعداد خطوط خروجی را word – length گویند.

2^n خط خروجی رمزگشا توسط اتصال‌های فیوزدار به هریک از گیت‌های OR متصل شده‌اند. به هنگام درج یک جدول ارزش در ROM اتصال‌هایی که موردنیاز نیستند را می‌توان با سوزاندن فیوزها از بین برد و به این عمل برنامه‌ریزی ROM می‌گویند.

یادداشت:

……

……

……

……

انواع ROM

ROM معمولی: که در آن محتوای ROM به دلخواه مشتری به وسیله‌ی سازنده ROM و به هنگام ساخت به‌طور دائمی در درون ROM درج می‌شود.

PROM: در آن به هنگام ساخت اتصال‌های فیوزدار در درون PROM تدارک دیده می‌شود و پس از تولید به کمک PROM Programmer می‌توان محتوای آن را به دلخواه استفاده کننده و در محل استفاده تعیین نمود.

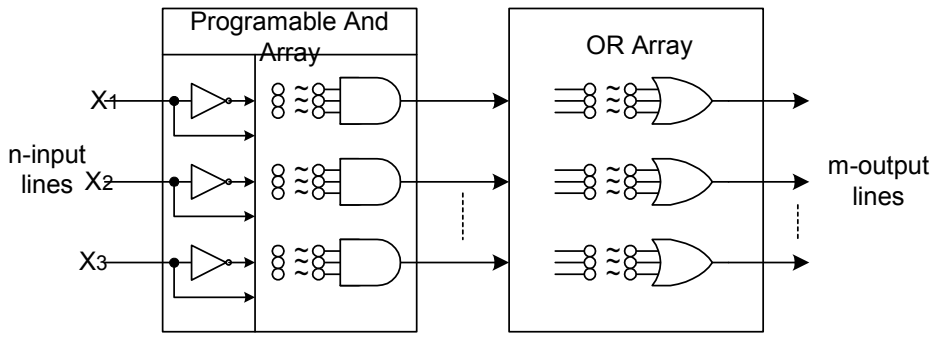
EPROM: محتوایش توسط استفاده‌کننده قابل تغییر است و پاک کردن EPROM با اشعه‌ی ماوراء بنفش صورت می‌گیرد.

EEPROM: همانند EPROM است ولی پاک کردن توسط سیگنال‌های الکتریکی صورت می‌گیرد.

یک ROM را هم می‌توان مدار ترکیبی در نظر گرفت که در هر مدار ترکیبی را پیاده‌سازی می‌کند و هم می‌توان یک حافظه در نظر گرفت که با گرفتن آدرس مشخص کلمه‌ای را تولید می‌کند.

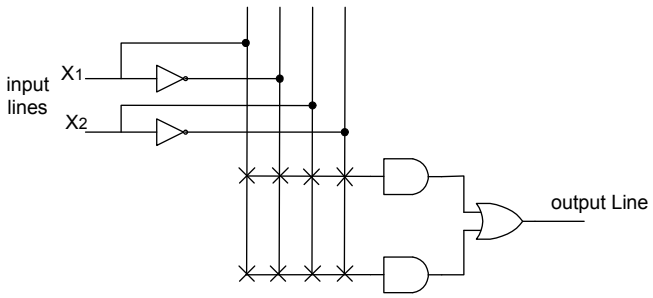
PLA

در طراحی ROM، مینترم متناظر با خروجی رمزگشا، آدرس کلمه موردنظر را تولید می‌کند، بنابراین اگر تعداد زیادی از آدرس‌ها در ورودی مدار ظاهر نشوند به دلیل عدم‌استفاده از کل اتصالات درونی، استفاده از ROM مقرون به صرفه نیست و بهتر است از PLA استفاده شود. PLA متغیرها را به‌طور کامل دیکد نمی‌کند و همه‌ی مینترم‌ها را تولید نمی‌کند. در PLA دیکدر با تعدادی گیت AND جایگزین شده است که هریک از آنها را می‌توان به‌منظور تولید جمله‌های حاصلضرب متغیرهای ورودی، برنامه‌ریزی کرد.



PAL

نوعی خاصی از PLA می‌باشد که در آن آرایه‌ی AND قابل برنامه‌ریزی ولی آرایه‌ی OR ثابت است. (برعکس ROM) بنابراین PAL برای برنامه‌ریزی راحت‌تر است ولی مثل PLA انعطاف‌پذیر نیست.



در PAL طرح مجاز به تعیین نوع جمله‌ی حاصل‌ضرب می‌باشد ولی تعداد ورودی‌های OR ثابت و غیرقابل تغییر است.

یادداشت:

.....

.....

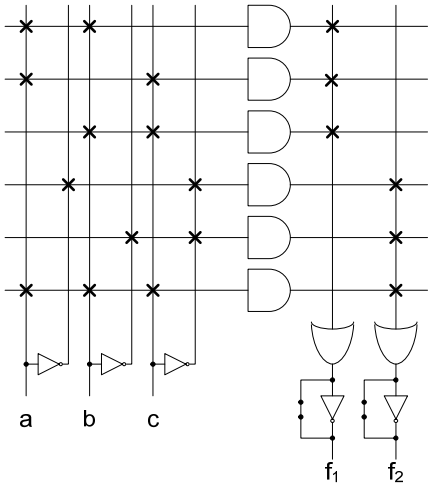
.....

.....

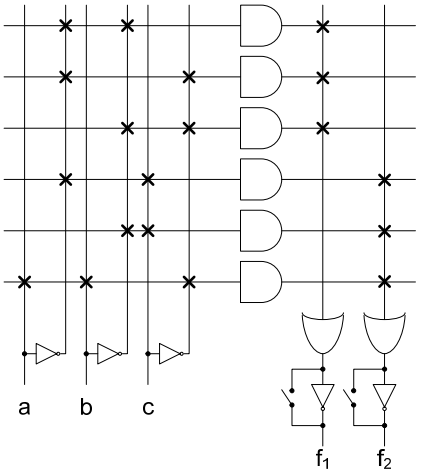
تست‌های فصل پنجم

۱ ـ با توجه به ساختار PLA، ساده‌ترین شکل پیاده‌سازی دو تـابع $f_1(a,b,c)=\sum m(3,5,6,7)$ و $f_2(a,b,c)=\sum m(0,2,4,7)$ کـدام است؟

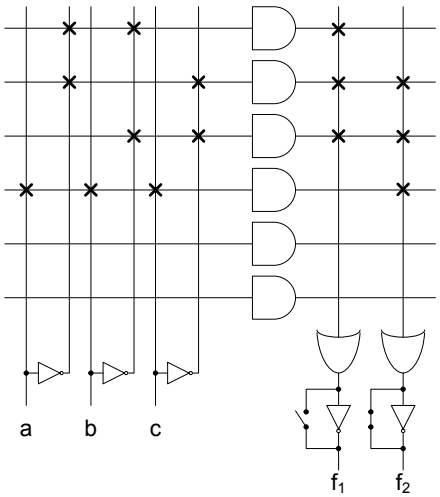
۱)



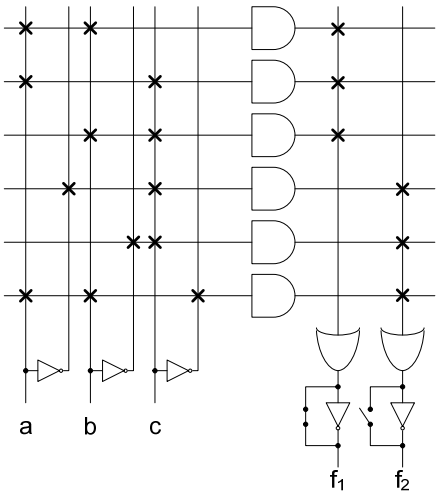
۲)



۳)



۴)



یادداشت:

.....

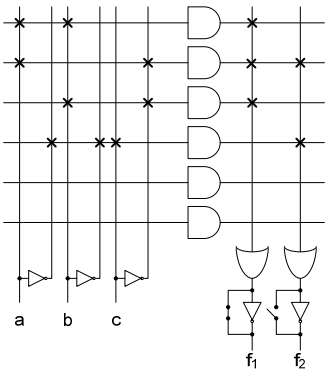
.....

.....

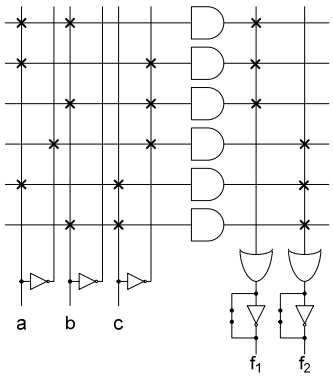
.....

۲ ـ با توجه به ساختار PLA، **ساده‌ترین شکل پیاده‌سازی دو تابع** $f_1(a,b,c)=\sum m(2,4,6,7)$ **و** $f_2(a,b,c)=\sum m(0,2,3,5,7)$ **کدام است؟**

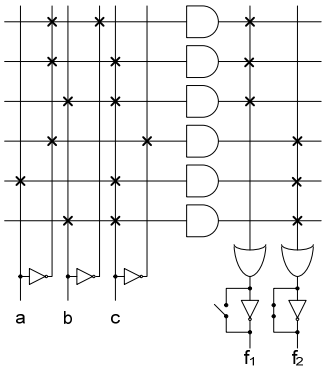
۱)



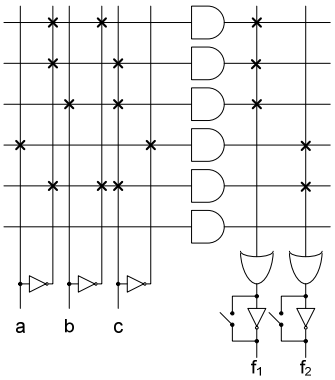
۲)



۳)



۴)



یادداشت:

.....

.....

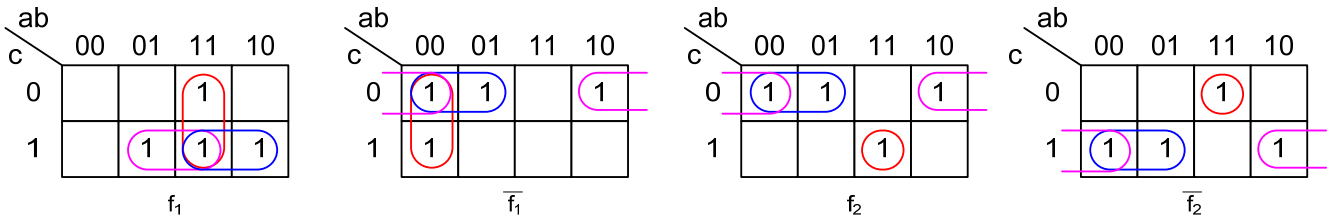
.....

.....

پاسخ‌ها

۱ـ گزینه ۳ درست است.

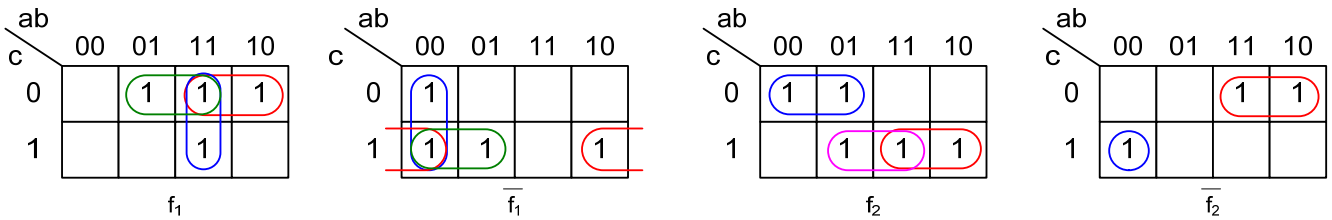
برای این منظور باید جدول کارنوی f_1 و f_2 و مکمل‌های آن‌ها را رسم کنیم و سعی کنیم زوجی را انتخاب کنیم که بیشترین جملات حاصل‌ضربی مشترک را دارا باشند. با توجه به شکل زیر $\bar{f}_1$ و f_2 مناسب‌ترین انتخاب هستند.



$$f_1(a,b,c) = ab + ac + bc \quad , \quad \bar{f}_1(a,b,c) = \bar{a}\bar{b} + \bar{a}\bar{c} + \bar{b}\bar{c}$$
$$f_2(a,b,c) = \bar{a}\bar{c} + \bar{b}\bar{c} + abc \quad , \quad \bar{f}_2(a,b,c) = \bar{a}c + \bar{b}c + ab\bar{c}$$

۲ـ گزینه ۱ درست است.

برای این منظور باید جدول کارنوی f_1 و f_2 و مکمل‌های آن‌ها را رسم کنیم و سعی کنیم زوجی را انتخاب کنیم که بیشترین جملات حاصل‌ضربی مشترک را دارا باشند. با توجه به شکل زیر $\bar{f}_2$ و f_1 مناسب‌ترین انتخاب هستند.



$$f_1(a,b,c) = ab + a\bar{c} + b\bar{c} \quad , \quad \bar{f}_1(a,b,c) = \bar{a}\bar{b} + \bar{a}c + \bar{b}c$$
$$f_2(a,b,c) = \bar{a}\bar{c} + ac + bc \quad , \quad \bar{f}_2(a,b,c) = a\bar{c} + \bar{a}\bar{b}c$$

یادداشت:

.....

.....

.....

.....