

تدریس پودمان سوم

تحلیل مدارهای دیجیتال

(واحد یادگیری ۵ و ۶)

واحد یادگیری ۵: به کارگیری قوانین جبر بول و نقشه (به صورت محاسباتی و نرم افزاری)

واحد یادگیری ۶: طراحی مدارهای دیجیتالی و اجرای توابع آن

وزارت آموزش و پرورش سازمان پژوهش و برنامه ریزی آموزشی دفتر تألیف کتاب های درسی فنی و حرفه ای و کاردانش		استاندارد تربیت و یادگیری شایستگی دانش فنی تخصصی - پایه ۱۲ - شاخه فنی و حرفه ای	
هدف کلی درس: ایجاد چارچوبی مفهومی، ادراکی و عملکردی در ساختار فکری هنرجویان به منظور سازگاری فناورانه و تعمیم دانش، مهارت و شایستگی های حرفه ای در موقعیت های جدید کاری و حرفه ای در دوران زندگی		تعداد جلسه: ۷ جلسه	تجهیزات آموزشی: وایت برد - رایانه - نرم افزار مرتبط
پودمان سوم: تحلیل مدارهای دیجیتال	استاندارد عملکرد پودمان: طراحی مدارهای ترکیبی و اجرای توابع و طراحی مدارهای ترتیبی	مکان اجرای آموزش: کلاس درسی مجهز به رایانه و ویدئو پروژکتور	ساعت آموزش پودمان: ۳۰ ساعت
وسعت محتوا: ساده سازی توابع منطقی و طراحی مدارهای ترکیبی و اجرای توابع با دیکندر و مالتی پلکسر، بررسی حافظه ها و شمارنده ها		زمان پیشنهادی آموزش: (غیردرسی) ۲۸+۲	ارزشیابی: سنجش مبتنی بر شایستگی

پیامدهای یادگیری:

اهداف توانمندساز	جلسه آموزشی	شایستگی‌ها
یادآوری گیت های منطقی	جلسه اول	به کارگیری قوانین جبر بول و نقشه (به صورت محاسباتی و نرم افزاری)
شرح قوانین جبر بول		
ساده سازی توابع منطقی با دو و سه متغیر با استفاده از قوانین جبر بول		
حل تمرین و رفع اشکال		
شرح نقشه کارنو و ساده سازی توابع منطقی با دو متغیر	جلسه دوم	
شرح نقشه کارنو و ساده سازی توابع منطقی با سه متغیر		
بررسی صحت توابع ساده شده با دو و سه متغیر با نرم افزار		
رفع اشکال و ارزشیابی		
شرح نقشه کارنو و ساده سازی توابع منطقی با چهار متغیر	جلسه سوم	
بررسی صحت توابع ساده شده با چهار متغیر با نرم افزار		
حل چند نمونه مثال از ساده سازی توابع با چهار متغیر		
رفع اشکال و ارزشیابی با کمک نرم افزار		
شرح چگونگی طراحی مدار ترکیبی	جلسه چهارم	طراحی مدارهای دیجیتال و اجرای توابع آن
شرح طراحی مدار ترکیبی با دیکدر		
شرح طراحی مدار ترکیبی با مالتی پلکسر		
اجرای تابع با دیکدر و مالتی پلکسر و بررسی صحت آن با نرم افزار		
شرح عملکرد انواع حافظه	جلسه پنجم	
شرح روش های انتقال اطلاعات به حافظه ها		
شرح عملکرد شمارنده ها		
تشریح شمارنده های صعودی و نزولی و اجرای آن با نرم افزار		
تشریح مدار مبدل دیجیتال به آنالوگ	جلسه ششم	
تشریح مدار مبدل آنالوگ به دیجیتال		
اجرای مدار مبدل دیجیتال به آنالوگ و آنالوگ به دیجیتال با نرم افزار		
رفع اشکال و ارزشیابی با کمک نرم افزار		
ارزشیابی پودمان سوم	جلسه هفتم	

توصیه‌های کاربردی در ارتباط با تدریس پودمان‌های سوم، چهارم و پنجم

پودمان سوم: تحلیل مدارهای دیجیتال
پودمان چهارم: محاسبه مقادیر DC در مدارهای الکترونیکی
پودمان پنجم: محاسبه مقادیر AC در مدارهای الکترونیکی

هدف از تدریس این پودمان‌ها آشنایی بیشتر هنرجویان با مباحث دیجیتال و مدارهای الکتریکی DC و AC در راستای آموخته‌های قبلی آنان به منظور رشد دانش نظری و نرم‌افزاری و آمادگی بیشتر برای اشتغال و حرکت در مسیر توسعه حرفه جهت صعود به مراتب بالاتر شایستگی و سایر فعالیت‌های اجتماعی است. فرایند کار و اجرای آموزش تقریباً مشابه پودمان‌های اول و دوم است. یادآور می‌شود که راهکارها و توصیه‌های ارائه شده صرفاً پیشنهادی است، لذا لازم است هنرآموزان با توجه به شرایط محیط آموزشی که تدریس می‌کنند از خلاقیت‌هایی که دارند بهره بگیرند و راهکارهای جدید را تجربه کنند و در اختیار سایر هنرآموزان قرار دهند.

■ پس از اتمام پودمان سوم، هنرجو باید موارد زیر را فرا گرفته باشد:

- ✓ یادآوری گیت‌های منطقی و شرح قوانین جبر بول
- ✓ توابع منطقی با دو، سه و چهار متغیر را با استفاده از قوانین جبر بول ساده کند.
- ✓ نقشه کارنو و ساده‌سازی توابع منطقی با دو، سه و چهار متغیر را تشریح کند و صحت توابع ساده شده با دو و سه متغیر را با استفاده از نرم‌افزار راستی آزمایی نماید.

✓ طراحی مدار ترکیبی با دیکدر و مالتی پلکسر را شرح داده و صحت آن را با نرم‌افزار راستی‌آزمایی کند.

✓ چگونگی عملکرد انواع حافظه و شمارنده‌ها را شرح دهد و آنها را با نرم‌افزار اجرا نماید.

✓ مدار مبدل دیجیتال به آنالوگ و آنالوگ به دیجیتال را تشریح و آنها را با نرم‌افزار اجرا نماید.

■ پس از اتمام پودمان چهارم، هنرجو باید موارد زیر را فرا گرفته باشد:

- ✓ روش‌های حل مدارهای مقاومتی یک حلقه‌ای و دو حلقه‌ای و حل مدار یک حلقه‌ای با یک و چند دیود را تشریح و آنها را با نرم‌افزار اجرا کند.
- ✓ مدار ترانزیستوری یک طبقه و دو طبقه را حل کند، سپس پاسخ‌ها را با استفاده از نرم‌افزار راستی‌آزمایی نماید.
- ✓ مدار معادل تونن و نورتن را تشریح و حل کند، سپس پاسخ‌ها را با استفاده از نرم‌افزار راستی‌آزمایی نماید.

- پس از اتمام پودمان پنجم، هنرجو باید موارد زیر را فرا گرفته باشد:
- ✓ مدارهای RC، RL و RLC سری به عنوان فیلتر عبور باند و RLC موازی به عنوان فیلتر حذف باند را تشریح کند و محاسبات مربوطه را انجام دهد. در نهایت پاسخ‌ها را با استفاده از نرم‌افزار راستی‌آزمایی نماید.
- ✓ مدارهای تک‌فاز و سه‌فاز را تشریح کند و جریان‌ها و ولتاژهای مدار سه‌فاز ستاره و مثلث با بار متعادل مقاومتی را محاسبه کند. در نهایت پاسخ‌ها را با استفاده از نرم‌افزار راستی‌آزمایی نماید.
- ✓ مدار معادل DC و AC تقویت‌کننده CE را رسم کند و آن را با نرم‌افزار اجرا کند.
- ✓ کمیت‌های تقویت‌کننده CE یک طبقه و دو طبقه در حالت AC را محاسبه کند و آنها را با نرم‌افزار اجرا کند.
- ✓ اصول نوسان‌سازی و تحلیل مدار نوسان‌سازهای RC انتقال فاز و RC پل وین را شرح دهد و فرکانس نوسان آنها را محاسبه کند و سپس با نرم‌افزار اجرا نماید.
- ✓ مدار نوسان‌سازهای LC (هارتلی)، LC (کولپیتس) و کریستالی را تحلیل کند و فرکانس نوسان‌سازهای LC را محاسبه و آنها را با نرم‌افزار اجرا کند.

تدریس پودمان سوم تحلیل مدارهای دیجیتال

واحد یادگیری ۵

کسب شایستگی در به‌کارگیری قوانین جبر بول و نقشه کارنو

دانش افزایی:

■ صورت استاندارد عبارات منطقی (توابع بولی)

یک تابع ممکن است با متغیرها یا نفی متغیرها (مثلاً \bar{A} و A) در تابع بولی ظاهر شود. n متغیر می‌توانند با نفی یا بدون نفی باشند. مثلاً در تابع بولی $Y = \bar{A}BC + AB\bar{C} + \bar{A}\bar{B}C + ABC$ سه متغیر A ، B و C یا نفی آنها وجود دارد. توابع بولی به شکل‌های مختلف نوشته می‌شوند.

✓ عبارت منطقی به‌صورت حاصل‌ضرب (Product):

اگر عمل منطقی بین متغیرها یا نفی آنها (مکمل آنها) AND باشد، عبارت را حاصل‌ضرب می‌گویند. تابع‌های بولی Y_1 ، Y_2 و Y_3 به‌صورت حاصل‌ضرب هستند.

$$Y_3 = A\bar{B}C\bar{D} \quad Y_7 = \bar{A}BC \quad Y_1 = ABC$$

✓ عبارت منطقی به صورت مجموع (Sum):

اگر عمل منطقی بین متغیرها یا نفی آنها (مکمل آنها) OR باشد، عبارت منطقی را مجموع می‌گویند. تابع های بولی Y_5 و Y_4 به صورت مجموع هستند.

$$Y_5 = A + \bar{B} + C + \bar{D} \quad Y_4 = A + B + C$$

✓ عبارت منطقی به صورت مجموع حاصل ضرب ها (SOP - Sum Of Product):

اگر چند جمله بولی حاصل ضرب با هم OR شده باشند، تابع بولی حاصل به صورت مجموع حاصل ضرب ها (SOP) است. تابع $Y = ABC + \bar{A}B + ABC\bar{D}$ به شکل SOP است، زیرا سه عبارت حاصل ضرب با هم OR شده‌اند.

✓ عبارت منطقی به صورت استاندارد مین ترم (Minterm):

تابع بولی به صورت استاندارد مین ترم باید شرایط زیر را داشته باشد: □ به صورت مجموع حاصل ضرب ها باشد.

□ در هر عبارت حاصل ضرب همه متغیرها یا نفی متغیرها وجود داشته باشد.

در کتاب درسی چگونگی نوشتن یک تابع به صورت استاندارد مین ترم با استفاده از جدول درستی توضیح داده شده است. در این مرحله، روش های دیگر ایجاد تابع منطقی به شکل استاندارد مین ترم شرح داده می شود.

□ استفاده از قوانین جبر بول

با استفاده از قوانین جبر بول می توان یک تابع منطقی را به صورت استاندارد مین ترم درآورد.

مثال ۱- تابع $F = \bar{A}C + AB$ را به صورت استاندارد مین ترم در آورید. این تابع به صورت SOP است ولی در هر عبارت آن همه متغیرها یا نفی آنها وجود ندارد.

چون یک منطقی (۱) عضو بی اثر در عبارت حاصل ضرب است، تابع را به صورت زیر بنویسیم:

$$F = \bar{A}C(1) + AB(1)$$

به جای یک منطقی (۱) می توانیم $C + \bar{C}$ یا $B + \bar{B}$ را قرار دهیم:

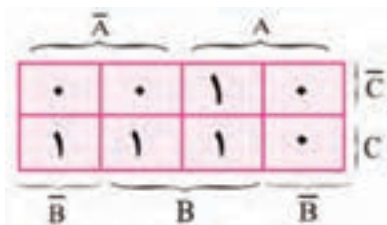
$$F = \bar{A}C(B + \bar{B}) + AB(C + \bar{C}) = \bar{A}CB + \bar{A}C\bar{B} + ABC + AB\bar{C}$$

$$F = ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C$$

پس از ساده سازی تابع به صورت استاندارد مین ترم در می آید.

استفاده از نقشه کارنو

تابع $F = \bar{A}C + AB$ را در نقشه کارنو نمایش می‌دهیم، شکل ۲۹. تابع را در هر خانه که ۱ است با سه متغیر می‌نویسیم. با توجه به شکل ۳۰ به صورت زیر در می‌آید.



شکل ۲۹

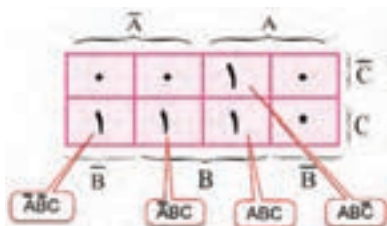
$$F = ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C$$

$$F = \sum_m (1, 3, 6, 7)$$

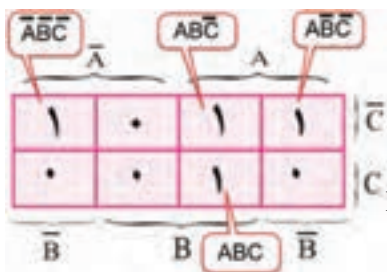
مثال ۲:

تابع $F = \bar{A}\bar{B}\bar{C} + AB + \bar{B}\bar{C}$ را توسط نقشه کارنو به صورت استاندارد مین‌ترم بنویسید.

تابع را طبق شکل ۳۱ در نقشه کارنو نمایش می‌دهیم.



شکل ۳۰



شکل ۳۱

مین‌ترم تابع را از طریق خانه‌هایی که در آن یک منطقی (۱) قرار دارد به صورت زیر می‌نویسیم.

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$$

$$F = \sum_m (0, 4, 6, 7)$$

✓ عبارت منطقی به صورت حاصل ضرب مجموع ها (POS_Product Of Sum)

اگر چند جمله بولی به صورت مجموع، با هم AND شده باشند، تابع بولی حاصل به شکل حاصل ضرب مجموع ها (POS) در می آید.

تابع ($(A+B+C)(\bar{A}+B)(\bar{A}+B+C+\bar{D})$) به صورت POS است، زیرا سه عبارت مجموع با هم AND شده اند.

✓ عبارت منطقی به صورت استاندارد ماکس ترم (Maxterm)

تابع بولی به شکل استاندارد ماکس ترم باید دارای شرایط زیر باشد:

□ به صورت حاصل ضرب مجموع ها باشد.

□ در هر عبارت مجموع همه متغیرها یا نفی متغیرها وجود داشته باشد.

تابع ($(\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+\bar{C})$) به شکل استاندارد ماکس ترم است زیرا اولاً عبارت به شکل POS است. ثانیاً در هر عبارت مجموع، هر سه متغیر A، B، و C یا نفی آنها وجود دارد. برای تبدیل توابع به صورت استاندارد ماکس ترم روش های مختلفی وجود دارد.

□ استفاده از قوانین جبر بول

✓ تابع $F=A+BC$ را در نظر بگیرید. برای تبدیل تابع به صورت POS از قانون توزیع استفاده می کنیم:

$$F=A+BC = (A+B)(A+C)$$

✓ چون صفر منطقی (0) یک عضو بی اثر در عبارت مجموع است، تابع را به صورت زیر می نویسیم:

$$F=(A+B+0) + (A+C+0)$$

به جای صفر منطقی (0) عبارت C و $B\bar{B}$ را قرار می دهیم:

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

✓ از قانون توزیع OR در AND استفاده می کنیم:

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

✓ جمله اول و سوم مشابه هستند، یکی را حذف می کنیم. تابع به شکل استاندارد ماکس ترم در می آید:

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

جدول ۲۸

A	B	C	BC	F
۰	۰	۰	۰	۰
۰	۰	۱	۰	۰
۰	۱	۰	۰	۰
۰	۱	۱	۱	۱
۱	۰	۰	۰	۱
۱	۰	۱	۰	۱
۱	۱	۰	۰	۱
۱	۱	۱	۱	۱

□ استفاده از جدول درستی

✓ در جدول ۲۸ جدول درستی تابع نوشته شده است.

✓ تابع \bar{F} را به صورت استاندارد مین ترم می نویسیم:

$$\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

✓ با نفی \bar{F} ، تابع F به صورت ماکس ترم نوشته می شود:

$$\bar{F} = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}}$$

✓ با استفاده از قانون دمورگان تابع F به صورت ماکس ترم در می آید.

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

☑ استفاده از نقشه کارنو

✓ تابع $F=A+BC$ را در نقشه کارنو نمایش می دهیم. از روی نقشه کارنو تابع \bar{F} (جایی که $F=0$ است) را به صورت استاندارد مین ترم می نویسیم:

$$\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

✓ با نفی \bar{F} و به کارگیری قانون دمورگان تابع F به صورت ماکس ترم نوشته می شود:

$$\bar{F} = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}}$$

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

■ نقشه کارنو با ۴ متغیر

در کتاب درسی ساده‌سازی توابع منطقی با سه متغیر توسط نقشه کارنو توضیح داده شده است. برای ساده‌سازی توابع منطقی با ۴ متغیر توسط نقشه کارنو به جدول با ۱۶ خانه ($2^4=16$) نیاز داریم. در شکل ۳۲ جمله مین‌ترم مربوط به هر ردیف جدول درستی نشان داده شده است. در شکل ۳۳ ارتباط بین متغیرها برای هر خانه جدول نوشته شده است.

AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}\bar{B}CD$
01	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BC\bar{D}$	$\bar{A}BCD$
11	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$AB\bar{C}\bar{D}$	$AB\bar{C}D$
10	$A\bar{B}C\bar{D}$	$A\bar{B}CD$	$ABC\bar{D}$	$ABCD$

شکل ۳۳

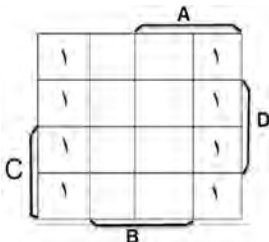
AB \ CD	00	01	11	10
00	m ₀	m ₁	m ₁₂	m ₈
01	m ₄	m ₅	m ₁₃	m ₉
11	m ₃	m ₇	m ₁₅	m ₁₁
10	m ₂	m ₆	m ₁₄	m ₁₀

شکل ۳۲

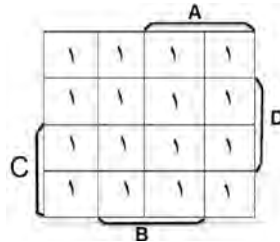
ساده کردن توابع چهار متغیر با استفاده از نقشه کارنو:

بعد از نمایش تابع در نقشه کارنو، برای ساده‌سازی باید نکات زیر را مورد توجه قرار دهید:

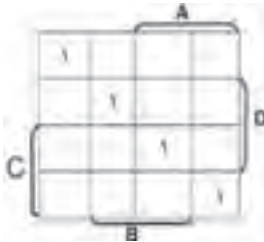
- ✓ شانزده مربع هم‌جوار (مجاور) نشان‌دهنده تابعی معادل یک منطقی (۱) است.
 - ✓ هشت مربع مجاور یک عبارت با یک متغیر را نشان می‌دهد.
 - ✓ چهار مربع مجاور نشان‌دهنده تابعی با دو متغیر است.
 - ✓ دو مربع مجاور، یک عبارت با سه متغیر را نشان می‌دهد.
 - ✓ یک مربع بدون خانه مجاور، با یک جمله مین‌ترم با چهار متغیر نوشته می‌شود.
- در شکل‌های ۳۴ تا ۳۸ برای هر مورد مثالی آورده شده است.



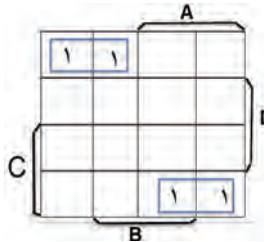
شکل ۳۵



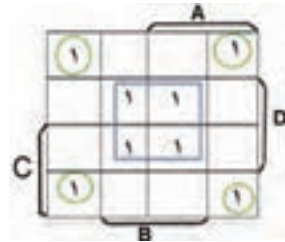
شکل ۳۴



شکل ۳۸



شکل ۳۷



شکل ۳۶

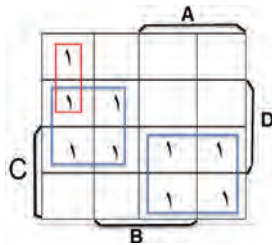
$$F = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} B \bar{C} D + A \bar{B} C \bar{D} + A B C D \quad F = A C \bar{D} + \bar{A} \bar{C} \bar{D} \quad F = B D + \bar{B} \bar{D}$$

مثال ۳: تابع F را در نقشه کارنو نمایش دهید و ساده‌ترین تابع منطقی را از روی نقشه کارنو بنویسید.

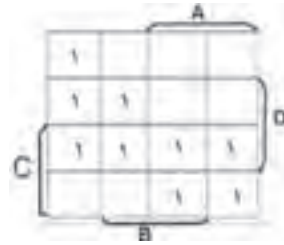
$$F = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} D + A C + C D + A B C D + \bar{A} \bar{B} C \bar{D}$$

پاسخ: تابع در نقشه کارنو مطابق شکل ۳۹ نشان داده شده است. در شکل ۴۰ خانه‌های مجاور مشخص شده‌اند.

ساده‌ترین شکل تابع به صورت $\bar{F} = \bar{A} \bar{B} \bar{C} + \bar{A} D + A C$ است.



شکل ۴۰



شکل ۳۹

نکته مهم



همان‌طور که در کتاب درسی گفته شد، با استفاده از محاسبه‌گرهای مجازی مبدل جدول کارنو به تابع و بالعکس می‌توانید ساده‌ترین حالت را برای جداول به‌دست آورید.

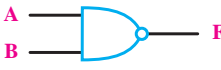
کاربرگ ارزشیابی واحد یادگیری ۵

نام و نام خانوادگی هنرجو:	کد کار: ۰۵۰۴	تاریخ:
---------------------------	--------------	--------

آزمون نظری: سؤال براساس الگوی پرسش

بارم آزمون: ۲۰ نمره

۱ نام گیت مقابل است و F زمانی صفر است که ورودی های A و B باشند. تابع منطقی F را به صورت F= می نویسند.



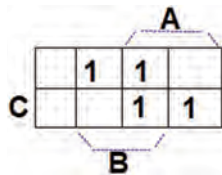
۲ تابع منطقی F را با استفاده از قوانین جبر بول ساده کنید.

$$F = \bar{A} \bar{B} \bar{C} + A \bar{C} + \bar{A} B \bar{C} + A B C + A \bar{B}$$

۳ رابطه منطقی جدول درستی را به شکل استاندارد مین ترم بنویسید.

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

۴ ساده ترین تابع منطقی نقشه کارنو را بنویسید و تابع را به وسیله فقط گیت NAND با دو ورودی طرح کنید.



۵

آزمون نرم افزاری: -

آزمون سخت افزاری: -

شایستگی های غیر فنی: بارم آزمون: ۲۰ نمره

کلیه آزمون ها براساس استاندارد عملکرد نمون برگ ۱-۸ انجام می شود.

واحد یادگیری ۶

نکته مهم



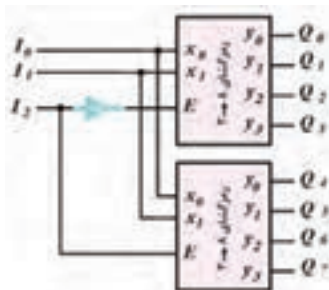
طراحی مدارهای دیجیتالی و اجرای توابع آن

قابل توجه هنرآموزان گرامی، محتوای ارائه شده در این قسمت جنبه دانش‌افزایی داشته و به هنرجویان آموزش داده نمی‌شود.

■ رمزگشا (Decoder)

توسعه رمزگشا

در واحد یادگیری ۶ مدار رمزگشا با ورودی تواناساز شرح داده شد. یکی از کاربردهای متداول ورودی تواناساز، گسترش رمزگشا از طریق سری کردن آنها است.



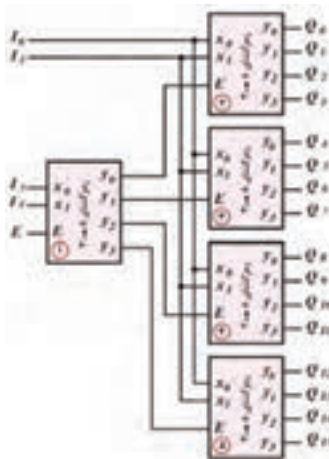
شکل ۴۱

✓ در شکل ۴۱ چگونگی تشکیل مدار رمزگشای ۳ به ۸ با استفاده از دو مدار رمزگشای ۲ به ۴ با ورودی تواناساز و یک دروازه NOT نشان داده شده است.

✓ اگر ورودی I_2 برابر صفر باشد، مدار رمزگشای بالا فعال شده و با انتخاب یکی از حالت‌های ۰۰۰۰، ۰۰۰۱، ۰۰۱۰، ۰۰۱۱ برای ورودی‌های رمزگشا، یکی از خروجی‌های رمزگشای بالایی یعنی Q_0 ، Q_1 ، Q_2 یا Q_3 ، یک می‌شود.

✓ رمزگشای پایین به ازای $I_2=1$ فعال می‌شود و امکان پاسخ‌دهی به حالت‌های ورودی ۱۰۰ تا ۱۱۱ را امکان‌پذیر می‌کند.

✓ در شکل ۴۲ روش ساخت رمزگشای ۴ به ۱۶ با استفاده از ۵ رمزگشای ۲ به ۴ با ورودی تواناساز را مشاهده می‌کنید.



شکل ۴۲

✓ در این مدار، رمزگشای ۱ برای انتخاب یکی از ۴ رمزگشای دیگر به کار رفته است.

✓ با ورودی‌های I_2 و I_1 ، یکی از رمزگشاهای ۲ تا ۵ فعال می‌شود و ورودی‌های I_0 و I_1 ، یکی از خروجی‌های رمزگشای فعال شده را انتخاب می‌کند.

✓ ورودی تواناساز رمزگشای ۱ به عنوان تواناساز کل مدار عمل می‌کند، یعنی اگر این ورودی در سطح منطقی صفر قرار گیرد، همه رمزگشاهای غیر فعال می‌شوند.

✓ به عنوان تمرین می‌توانید با توجه به مطالب گفته شده، با استفاده از دو رمزگشای

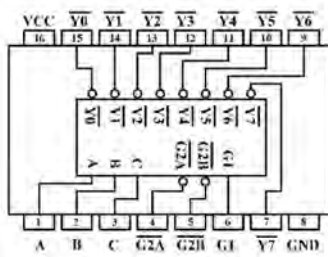
۳ به ۸، رمزگشای ۴ به ۱۶ بسازید.

✓ معرفی آی سی رمزگشای ۷۴۱۳۸

شکل ۴۳ آرایش پایه‌ها و جدول درستی آی سی ۷۴۱۳۸ را که یک رمزگشای ۳ به ۸ پرکاربرد است، نشان می‌دهد. در این آی سی، خروجی‌ها (Y_7 تا Y_0) از نوع صفر فعال هستند و توسط ترکیبی از سه ورودی توانا ساز G_1 ، G_{2A} و G_{2B} فعال می‌شوند. رمزگشا وقتی توانا است که $G_1 = 1$ ، $\overline{G_{2A}} = 0$ و $\overline{G_{2B}} = 0$ باشد، زیرا G_1 یک فعال و $\overline{G_{2A}}$ و $\overline{G_{2B}}$ صفر فعال هستند. در جدول درستی رسم شده در شکل ۴۳، $\overline{G_2} = \overline{G_{2A}} + \overline{G_{2B}}$ به دست می‌آید.

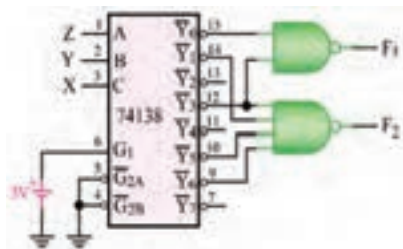
ورودی‌ها		خروجی‌ها							
Enable	Select								
G_1 G_2	A B C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
H L	L L L	L	H	H	H	H	H	H	H
H L	L L H	H	L	H	H	H	H	H	H
H L	L H L	H	H	L	H	H	H	H	H
H L	L H H	H	H	H	L	H	H	H	H
H L	H L L	H	H	H	H	L	H	H	H
H L	H L H	H	H	H	H	H	L	H	H
H L	H H L	H	H	H	H	H	H	L	H
H L	H H H	H	H	H	H	H	H	H	L
L ×	× × ×	H	H	H	H	H	H	H	H
L ×	× × ×	H	H	H	H	H	H	H	H

جدول درستی
 $\overline{G_2} = \overline{G_{2A}} + \overline{G_{2B}}$



آرایش پایه‌ها

شکل ۴۳



شکل ۴۴

چون خروجی‌های آی سی رمزگشای ۷۴۱۳۸ صفر فعال هستند، بنابراین برای ساخت توابع مختلف با این دیکودر، باید به جای استفاده از دروازه OR از دروازه NAND استفاده کرد. مثال: با استفاده از رمزگشای ۷۴۱۳۸، توابع $F_1 = \sum_m(0, 3)$ و $F_2 = \sum_m(1, 3, 5, 6)$ را اجرا کنید.

در شکل ۴۴ مدار مورد نظر با استفاده از دیکودر ۷۴۱۳۸ نشان داده شده است.

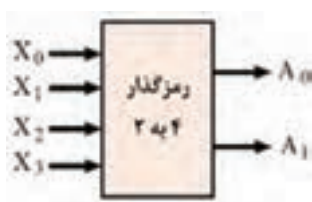
■ رمزگذار (Encoder)

رمزگذار مداری است که به ازای هر ورودی انتخاب شده، یک کد دودویی انحصاری در خروجی ایجاد می‌کند. به عبارت دیگر، رمزگذار برعکس مدار رمزگشا عمل می‌کند. یک مدار رمزگذار با 2^n خط ورودی (یا کمتر)، دارای n خط خروجی است.

ساختار مدار رمزگذار

مدار رمزگذار را در دو حالت می‌توان بررسی کرد. در حالت اول، در هر زمان فقط یکی از ورودی‌ها فعال می‌شود و در هیچ شرایطی دو ورودی یا بیشتر از دو ورودی به‌طور هم‌زمان فعال نمی‌شوند.

✓ طراحی رمزگذار ۴ به ۲ با تنها یک ورودی فعال در هر زمان در شکل ۴۵ بلوک دیاگرام مدار رمزگذار ۴ به ۲ رسم شده است. عملکرد مدار رمزگذار بدون



شکل ۴۵

X_3	X_2	X_1	X_0	A_1	A_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

شکل ۴۶

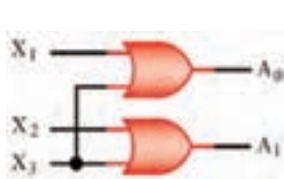
X_3	X_2	X_1	X_0	A_1	A_0
0	0	0	0	d	d
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	d	d
0	1	0	0	1	0
0	1	0	1	d	d
0	1	1	0	d	d
0	1	1	1	d	d
1	0	0	0	1	1
1	0	0	1	d	d
1	0	1	0	d	d
1	0	1	1	d	d
1	1	0	0	d	d
1	1	0	1	d	d
1	1	1	0	d	d
1	1	1	1	d	d

شکل ۴۷

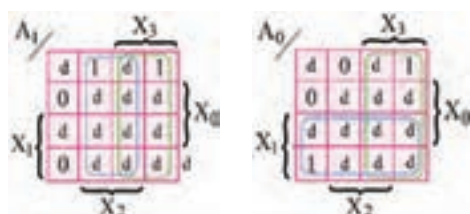
در نظر گرفتن حالت‌هایی که در ورودی اتفاق نمی‌افتند را در جدول درستی شکل ۴۶ مشاهده می‌کنید. در این جدول، تابع خروجی برابر با مقدار دودویی زیرنویس متغیر ورودی در نظر گرفته شده است. مثلاً وقتی ورودی X_2 فعال شود، در خروجی مدار، مقدار دودویی عدد ۲، یعنی ۱۰ نشان داده می‌شود.

با توجه به اینکه جدول درستی برای ۴ متغیر ورودی دارای ۱۶ حالت است، ترکیب‌هایی از ورودی‌ها که رخ نمی‌دهند، حالت رها، بی‌اهمیت یا آزاد (don't care) محسوب می‌شوند. جدول درستی با در نظر گرفتن حالت‌های رها در شکل ۴۷ رسم شده است. در این جدول، حالت‌های رها با حرف d نشان داده شده است.

در شکل ۴۸ نقشه کارنو به همراه تابع ساده شده خروجی‌های A_0 و A_1 و در شکل ۴۹ مدار منطقی ساده شده رمزگذار ۴ به ۲ را مشاهده می‌کنید. برای اینکه مدار منطقی رسم شده در شکل ۴۹ به‌طور صحیح کار کند، باید در هر زمان تنها یک ورودی فعال شود.



شکل ۴۹



$$A_1 = X_2 + X_3$$

$$A_0 = X_0 + X_1$$

شکل ۴۸

	A_2	A_1	A_0
$X_1 \rightarrow$	0	0	1
$X_2 \rightarrow$	0	1	0
$X_3 \rightarrow$	0	1	1
$X_4 \rightarrow$	1	0	0
بقیه حالت‌ها	0	0	0

شکل ۵۰

✓ به عنوان تمرین می‌توانید مدار رمزگذار طراحی شده را طوری اصلاح کنید که اگر هیچ‌یک از ورودی انتخاب نشوند یا بیش از یک ورودی انتخاب شود، همه خروجی‌ها صفر شوند، (شکل ۵۰).



شکل ۵۱

✓ طراحی رمزگذار ۴ به ۲ اولویت‌دار

✓ در رمزگذار اولویت‌دار، اگر دو یا چند ورودی به‌طور هم‌زمان فعال شوند، با توجه به نوع طراحی انجام شده، یکی از ورودی‌ها در اولویت قرار می‌گیرند.

✓ برای سادگی طرح، فرض می‌کنیم ورودی با اندیس بزرگ‌تر، در اولویت قرار داشته باشد.

✓ در شکل ۵۱ بلوک دیاگرام رمزگذار اولویت‌دار رسم شده است. همان‌طور که در این شکل مشاهده می‌کنید، در این طرح علاوه بر دو خروجی اصلی A_1, A_0 ، یک خروجی دیگر، یعنی E_0 به مدار اضافه شده است. خروجی E_0 وقتی فعال می‌شود که هیچ‌یک از ورودی‌ها فعال نباشد.

X_3	X_2	X_1	X_0	A_1	A_0
0	0	0	0	d	d
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	d	d
0	1	0	0	1	0
0	1	0	1	d	d
0	1	1	0	d	d
0	1	1	1	d	d
1	0	0	0	1	1
1	0	0	1	d	d
1	0	1	0	d	d
1	0	1	1	d	d
1	1	0	0	d	d
1	1	0	1	d	d
1	1	1	0	d	d
1	1	1	1	d	d

شکل ۵۲

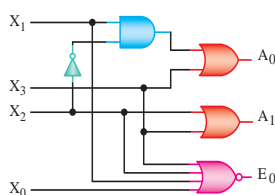
✓ اگر هیچ کدام از ورودی‌ها فعال نباشد، خروجی‌های A_1 و A_0 برابر صفر خواهند بود، $(A_1 A_0 = 00)$.

✓ اگر تنها یکی از ورودی‌ها فعال باشند، خروجی $A_1 A_0$ با مقدار دودویی زیرنویس خط ورودی برابر است. در شکل ۵۲ جدول درستی این رمزگذار را مشاهده می‌کنید.

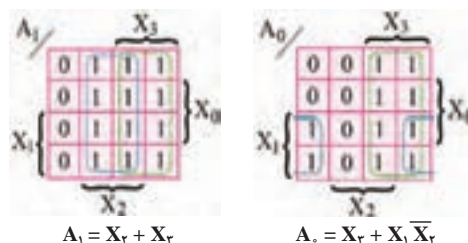
✓ همان‌طور که اشاره شد، اگر دو یا چند ورودی به‌طور هم‌زمان فعال شوند، ورودی با اندیس بزرگ‌تر، در اولویت قرار دارد و خروجی $A_1 A_0$ کد دودویی بزرگ‌ترین زیرنویس خط ورودی را در خروجی نشان می‌دهد.

✓ در شکل ۵۳ نقشه کارنو و تابع منطقی ساده شده خروجی‌های A_1 و A_0 را مشاهده می‌کنید. تابع منطقی خروجی E برابر با عبارت زیر می‌شود.

$$E = X_r + X_r + X_r + X_r$$



شکل ۵۴



شکل ۵۳

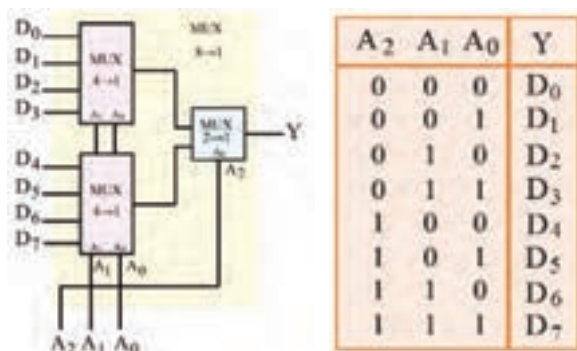
✓ در شکل ۵۴ مدار منطقی رمزگذار اولویت دار رسم شده است.

توسعه مالتی پلکسر

با اتصال چند مالتی پلکسر، می‌توان تعداد ورودی‌های مالتی پلکسر را افزایش داد. مثلاً می‌توان با استفاده از دو مالتی پلکسر ۴ به ۱ و یک مالتی پلکسر ۲ به ۱، مالتی پلکسر ۸ به ۱ را مانند شکل ۵۵ طراحی کرد.

همان طور که در شکل ۵۵ مشاهده می کنید، دو مالتی پلکسر ۴ به ۱، مالتی پلکسر ۲ به ۱ را تحریک می کنند و با توجه به آدرس A_2 ، یکی از ورودی های D_0 تا D_7 در خروجی ظاهر می شود.

✓ به عنوان تمرین می توانید با استفاده از مالتی پلکسرهای ۴ به ۱، مالتی پلکسر ۱۶ به ۱ طراحی کنید.



شکل ۵۵



شکل ۵۶

مقایسه کننده چهار بیتی

مقایسه کننده مداری است که اندازه دودویی دو عدد A و B را مقایسه می کند و بزرگ تر بودن، کوچک تر بودن یا مساوی بودن آنها را مشخص می کند.

مقایسه کننده دو بیتی

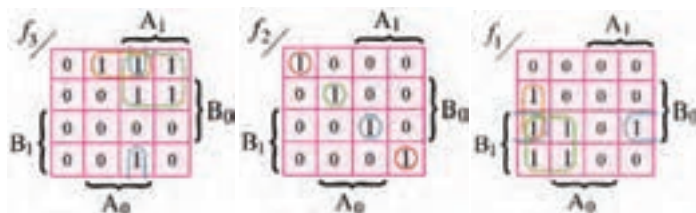
$A_1 A_0$	$B_1 B_0$	f_1 $A < B$	f_2 $A = B$	f_3 $A > B$
00	00	0	1	0
00	01	1	0	0
00	10	1	0	0
00	11	1	0	0
01	00	0	0	1
01	01	0	1	0
01	10	1	0	0
01	11	1	0	0
10	00	0	0	1
10	01	0	0	1
10	10	0	1	0
10	11	1	0	0
11	00	0	0	1
11	01	1	0	1
11	10	0	0	1
11	11	0	1	0

شکل ۵۷

✓ بلوک دیگرام مدار مقایسه کننده دو عدد دو بیتی

$A = (A_1 A_0)_2$ و $B = (B_1 B_0)_2$ در شکل ۵۶ رسم شده است. جدول درستی این مدار را در شکل ۵۷ مشاهده می کنید.

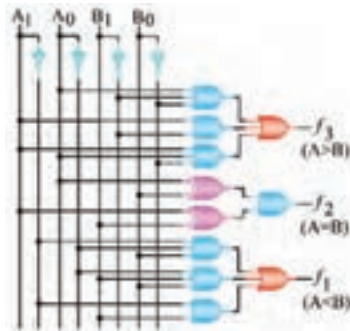
✓ در شکل ۵۸ نقشه کارنو برای خروجی های f_1 ، f_2 و f_3 را مشاهده می کنید.



شکل ۵۸

✓ تابع ساده شده خروجی‌های f_1 ، f_2 و f_3 با استفاده از نقشه‌های کارنو شکل ۵۸ به صورت زیر به دست می‌آید:

$$\begin{aligned} f_1 &= \overline{A_1}B_1 + \overline{A_1} \overline{A_0} B_0 + \overline{A_0}B_1B_0 \\ f_2 &= A_1\overline{B_1} + A_0 \overline{B_1} \overline{B_0} + A_1A_0 \overline{B_0} \\ f_3 &= \overline{A_1}\overline{A_0}\overline{B_1} \overline{B_0} + \overline{A_1}A_0\overline{B_1} B_0 + A_1\overline{A_0}B_1\overline{B_0} + A_1A_0B_1B_0 \\ &= \overline{A_1}\overline{B_1}(\overline{A_0}\overline{B_0} + A_0B_0) + A_1B_1(\overline{A_0}\overline{B_0} + A_0B_0) \\ &= (\overline{A_1}\overline{B_1} + A_1B_1)(\overline{A_0}\overline{B_0} + A_0B_0) \\ &= (\overline{A_1} \oplus \overline{B_1})(\overline{A_0} \oplus \overline{B_0}) \end{aligned}$$



شکل ۵۹

✓ همان طور که ملاحظه می‌شود، تابع f_2 را نمی‌توان به کمک نقشه کارنو ساده کرد. برای ساده‌سازی باید قوانین جبر بول مورد استفاده قرار گیرد.

✓ مدار منطقی مقایسه‌کننده دو بیتی در شکل ۵۹ رسم شده است.

معرفی آی‌سی مقایسه‌کننده ۷۴۸۵

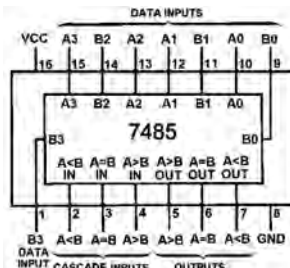
✓ آی‌سی ۷۴۸۵ یک مقایسه‌کننده ۴ بیتی است که دو عدد چهار بیتی $A = (A_3A_2A_1A_0)_2$ و $B = (B_3B_2B_1B_0)_2$ را با هم مقایسه می‌کند و نتیجه مقایسه را با توجه به شرایط، با فعال کردن یکی از سه خروجی $A > B$ ، $A = B$ و $A < B$ نشان می‌دهد.

شکل ۶۰ بلوک دیاگرام داخلی آی‌سی ۷۴۸۵ و در شکل ۶۱ جدول درستی آن را مشاهده می‌کنید.

✓ آی‌سی ۷۴۸۵ دارای سه ورودی برای سری کردن با آی‌سی‌های مشابه است و می‌توان اعداد بزرگ‌تر از چهار بیت را نیز با سری کردن چند آی‌سی ۷۴۸۵، مقایسه کرد.

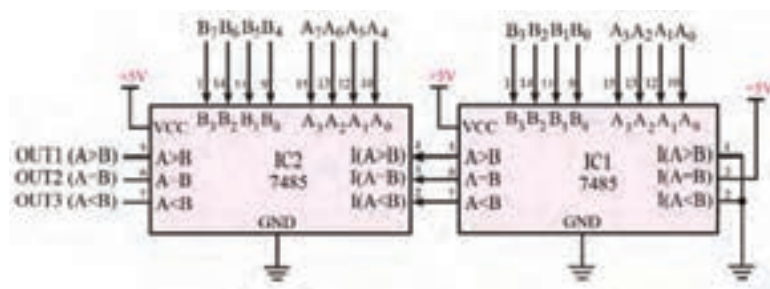
Comparing				Cascading				Outputs			
A ₃ B ₃	A ₂ B ₂	A ₁ B ₁	A ₀ B ₀	A ₃ A ₂	A ₂ A ₁	A ₁ A ₀	A ₃ B ₂	A ₂ B ₁	A ₁ B ₀	A ₀ B ₀	A ₃ B ₀
A ₃ <B ₃	X	X	X	X	X	X	X	X	X	X	X
A ₃ =B ₃	X	X	X	X	X	X	X	X	X	X	X
A ₃ >B ₃	X	X	X	X	X	X	X	X	X	X	X
A ₂ <B ₂	X	X	X	X	X	X	X	X	X	X	X
A ₂ =B ₂	X	X	X	X	X	X	X	X	X	X	X
A ₂ >B ₂	X	X	X	X	X	X	X	X	X	X	X
A ₁ <B ₁	X	X	X	X	X	X	X	X	X	X	X
A ₁ =B ₁	X	X	X	X	X	X	X	X	X	X	X
A ₁ >B ₁	X	X	X	X	X	X	X	X	X	X	X
A ₀ <B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₀ =B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₀ >B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₃ <B ₂	X	X	X	X	X	X	X	X	X	X	X
A ₃ =B ₂	X	X	X	X	X	X	X	X	X	X	X
A ₃ >B ₂	X	X	X	X	X	X	X	X	X	X	X
A ₂ <B ₁	X	X	X	X	X	X	X	X	X	X	X
A ₂ =B ₁	X	X	X	X	X	X	X	X	X	X	X
A ₂ >B ₁	X	X	X	X	X	X	X	X	X	X	X
A ₁ <B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₁ =B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₁ >B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₀ <B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₀ =B ₀	X	X	X	X	X	X	X	X	X	X	X
A ₀ >B ₀	X	X	X	X	X	X	X	X	X	X	X

شکل ۶۱



شکل ۶۰

✓ در شکل ۶۲ مدار مقایسه کننده دو عدد ۸ بیتی A و B که با سری کردن دو آی سی ۷۴۸۵ طراحی شده است را مشاهده می کنید.



شکل ۶۲

✓ به عنوان تمرین می توانید با استفاده از چهار عدد آی سی ۷۴۸۵، مدار مقایسه کننده ای طراحی کنید که دو عدد ۱۶ بیتی را با هم مقایسه کند.

■ مدارهای جمع کننده

جمع کننده کامل (Full Adder)

✓ در درس دانش فنی تخصصی، جمع کننده کامل معرفی گردید و مدار آن به کمک رمزگشای ۳ به ۸ طراحی شد. مدار جمع کننده کامل را می توان با استفاده از دروازه های منطقی نیز طراحی کرد. در شکل ۶۳ جدول درستی مدار جمع کننده کامل رسم شده است.

شماره سطر	A	B	C _{in}	C _{out}	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

شکل ۶۳

✓ با توجه به جدول درستی، می توان تابع منطقی خروجی های S و C_{out} را به دست آورده و ساده کرد.

$$\begin{aligned}
 S &= \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in} = C_{in}(\overline{A}\overline{B} + AB) + \overline{C}_{in}(\overline{A}\overline{B} + AB) \\
 &= C_{in}(\overline{A \oplus B}) + \overline{C}_{in}(A \oplus B) = C_{in} \oplus (A \oplus B) \\
 C_{out} &= \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in} = C_{in}(\overline{A}\overline{B} + AB) + AB(C_{in} + \overline{C}_{in}) \\
 &= C_{in}(A \oplus B) + AB
 \end{aligned}$$

✓ مدار منطقی خروجی‌های ساده شده مدار جمع‌کننده کامل در شکل ۶۴ رسم شده است.

نکته



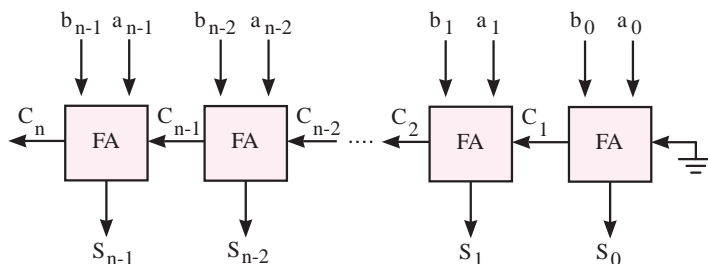
همان‌طور که در شکل مشاهده می‌کنید، مدار جمع‌کننده کامل را می‌توان با استفاده از دو نیم جمع‌کننده و یک دروازه منطقی OR طراحی کرد.



شکل ۶۴

✓ جمع دو عدد باینری n بیتی

برای جمع کردن دو عدد n بیتی $(a_{n-1} \dots a_2 a_1 a_0)_2$ و $(b_{n-1} \dots b_2 b_1 b_0)_2$ ، باید n طبقه مدار تمام جمع‌کننده را طوری به هم وصل کنیم که رقم نقلی خروجی هر طبقه به رقم نقلی ورودی طبقه بعد وصل شود، (شکل ۶۵). رقم نقلی ورودی طبقه اول نیز به زمین وصل می‌شود.

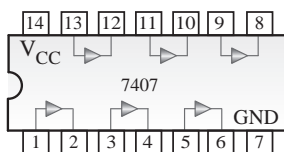


شکل ۶۵

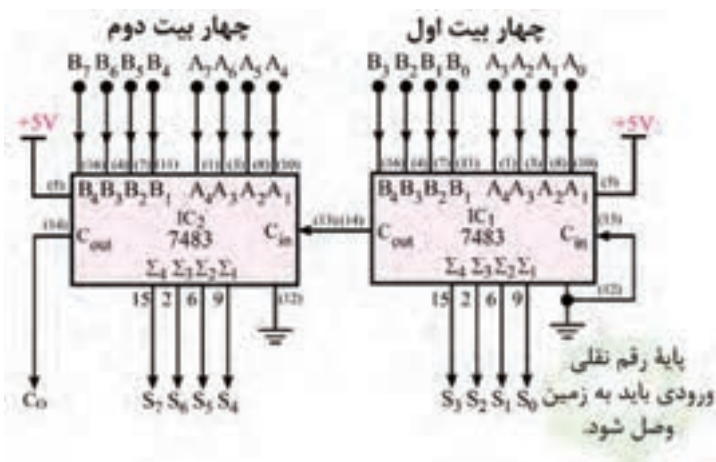
✓ مدار جمع‌کننده با آی‌سی ۷۴۸۳

آی‌سی ۷۴۸۳ یک جمع‌کننده کامل چهار بیتی است که قابلیت توسعه داشته و می‌توان به کمک آن جمع‌کننده‌های بزرگ‌تر ساخت. در شکل ۶۶ آرایش پایه‌های این آی‌سی نشان داده شده است.

در این آی سی، A_1 تا A_4 و B_1 تا B_4 بیت های اعداد دودویی ورودی و Σ_1 تا Σ_4 بیت های حاصل جمع هستند. همچنین C رقم نقلی ورودی و C_4 رقم نقلی خروجی است. در شکل ۶۷ مدار جمع کننده ۸ بیتی را که با استفاده از دو آی سی ۷۴۸۳ ساخته شده است، مشاهده می کنید.



شکل ۶۶



شکل ۶۷

■ مدارهای تفریق کننده

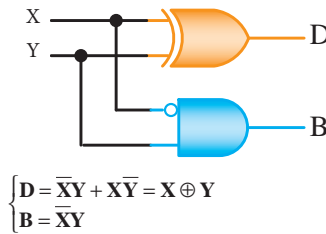
تفریق کننده ناقص (Half Subtractor)

✓ مدارهای تفریق کننده مانند مدارهای جمع کننده طراحی می شوند. در تفریق کننده ناقص یا نیم تفریق کننده، تفاضل دو عدد یک بیتی X و Y محاسبه شده و دو خروجی مدار، D و B هستند. D حاصل تفریق (Difference) و B رقم قرضی (Borrow) است. در شکل ۶۸ جدول درستی مدار تفریق کننده ناقص رسم شده است.

ورودی ها	خروجی ها	
X	Y	B D
0	0	0 0
0	1	1 1
1	0	0 1
1	1	0 0

شکل ۶۸

✓ در تفریق $X-Y$ ، اگر $X > Y$ باشد، عمل تفریق بدون نیاز به رقم قرضی انجام می‌شود و بیت قرضی، یعنی B برابر صفر خواهد بود. ولی اگر $X < Y$ باشد، برای انجام عمل تفریق نیاز به یک بیت قرضی خواهد بود، یعنی $B=1$ است. عمل قرض مشابه تفریق دهمی است و با قرض گرفتن یک بیت، دو واحد به عدد X اضافه می‌شود.



شکل ۶۹

✓ با توجه به جدول درستی مدار تفریق کننده ناقص، توابع خروجی‌های این مدار را می‌نویسیم:

✓ در شکل ۶۹ مدار تفریق کننده ناقص که براساس توابع خروجی آن رسم شده است را مشاهده می‌کنید.

تفریق کننده کامل (Full Subtractor)

✓ مدار تفریق کننده کامل را می‌توان مشابه مدار جمع کننده کامل طراحی کرد. عمل تفریق و قرض گرفتن نیز مشابه تفریق ناقص است.

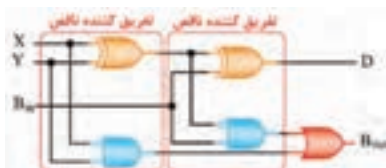
شماره سطر	X	Y	B_{in}	B_{out}	D
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

شکل ۷۰

✓ در شکل ۷۰ جدول درستی مدار تفریق کننده کامل رسم شده است.

✓ با توجه به جدول درستی، می‌توان تابع منطقی خروجی‌های D و B_{out} را به دست آورده و ساده کرد.

$$\begin{aligned} D &= \overline{X}\overline{Y}B_{in} + \overline{X}Y\overline{B_{in}} + X\overline{Y}\overline{B_{in}} + XYB_{in} = B_{in}(\overline{X}\overline{Y} + XY) + \overline{B_{in}}(\overline{X}Y + X\overline{Y}) \\ &= B_{in}(\overline{X} \oplus \overline{Y}) + \overline{B_{in}}(X \oplus Y) = B_{in} \oplus (X \oplus Y) \\ B_{out} &= \overline{X}\overline{Y}B_{in} + \overline{X}Y\overline{B_{in}} + \overline{X}YB_{in} + XYB_{in} = B_{in}(\overline{X}\overline{Y} + XY) + \overline{X}Y(\overline{B_{in}} + B_{in}) \\ &= B_{in}(\overline{X} \oplus \overline{Y}) + \overline{X}Y \end{aligned}$$



شکل ۷۱

✓ مدار منطقی خروجی‌های ساده شده مدار تفریق کننده کامل در شکل ۷۱ رسم شده است.

تفریق به روش متمم ۲

✓ یکی از روش‌های عملی تفریق دو عدد دودویی، استفاده از روش متمم ۲ است. نکته: برای به‌دست آوردن متمم ۲ یک عدد دودویی، ابتدا تمام بیت‌های آن عدد را متمم کرده، سپس یک واحد با آن عدد اضافه می‌کنیم. به‌طور مثال، متمم ۲ عدد ۱۰۱۱ برابر ۰۱۰۱ است:

متمم ۲

$$1011 \xrightarrow[\text{متمم شوند}]{\text{تمام بیت‌ها}} 0100 \xrightarrow{+1} 0101$$

$$\begin{array}{r} 11011 \\ + 10010 \\ \hline \text{رقم نقلی نهایی} \quad \text{حاصل تفریق} \quad 01101 \end{array}$$

شکل ۷۲

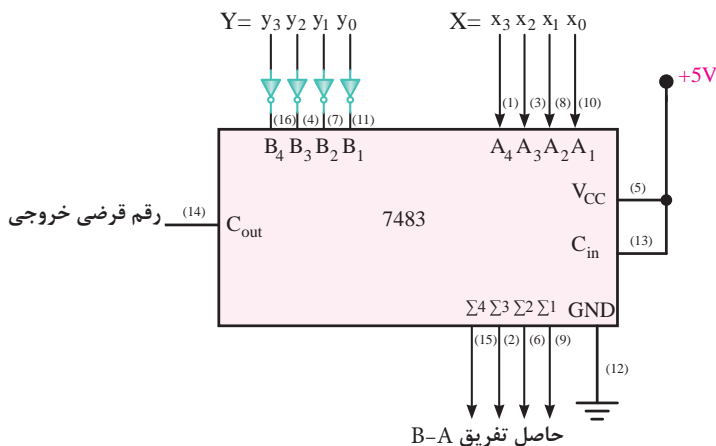
✓ انجام تفریق به روش متمم ۲:
برای محاسبه حاصل تفریق دو عدد X و Y به روش متمم ۲، ابتدا متمم ۲ عدد Y را به‌دست می‌آوریم و سپس آن را با X جمع می‌کنیم. در حاصل جمع، یک بیت نقلی اضافه‌ایجاد می‌شود که باید حذف شود، شکل ۷۲.

مثال: حاصل تفریق ۰۱۱۱۰ - ۱۱۰۱۱ را به‌دست آورید.

$$01110 \xrightarrow{\text{متمم ۲}} 10010$$

ابتدا متمم ۲ عدد ۰۱۱۱۰ را به‌دست می‌آوریم:
حال ۱۱۰۱۱ را با متمم ۲ عدد ۰۱۱۱۰، یعنی ۱۰۰۱۰ جمع می‌کنیم و رقم نقلی نهایی را حذف می‌کنیم.
حاصل تفریق برابر ۰۱۱۰۱ است.

✓ اگر بخواهیم حاصل تفریق دو عدد چند بیتی X و Y، یعنی X-Y را با استفاده از روش متمم ۲ به‌دست آوریم، ابتدا بیت‌های عدد Y را با استفاده از دروازه‌های NOT متمم می‌کنیم؛ سپس رقم ورودی نقلی را در سطح منطقی یک قرار می‌دهیم. به‌این ترتیب، یک واحد به عدد متمم شده اضافه می‌شود و متمم ۲ عدد Y به‌دست می‌آید. بعد از این مرحله، عمل جمع عدد X با متمم ۲ عدد Y را با استفاده از مدار جمع‌کننده کامل چند بیتی انجام می‌دهیم. شکل ۷۳ تفریق چهار بیتی را با استفاده از روش متمم ۲ و به کمک آی‌سی ۷۴۸۳ نشان می‌دهد.

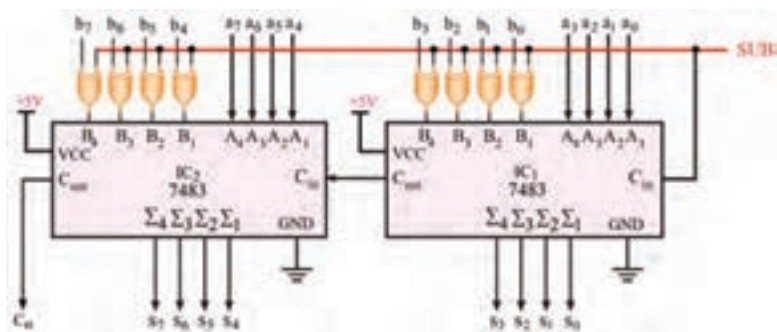


شکل ۷۳

مدار جمع کننده/تفریق کننده

✓ با استفاده از دروازه‌های منطقی XOR و مدار جمع کننده کامل می‌توان مداری طراحی کرد که بتوان به کمک خط کنترل ورودی، حالت جمع یا تفریق را انتخاب کرد.

✓ شکل ۷۴ مدار جمع کننده/تفریق کننده ۸ بیتی را نشان می‌دهد. در این مدار، ورودی SUB، خط کنترل برای انتخاب حالت جمع ($A+B$) یا تفریق ($A-B$) است.

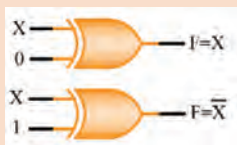


شکل ۷۴

نکته



اگر یکی از ورودی‌های دروازه منطقی XOR برابر با یک باشد، این دروازه مانند دروازه NOT عمل می‌کند و خروجی برابر با متمم ورودی می‌شود. همچنین اگر یکی از ورودی‌های دروازه منطقی XOR برابر با صفر باشد، خروجی برابر با ورودی دوم خواهد شد، (شکل ۷۵).



شکل ۷۵

✓ اگر $SUB=0$ باشد، بیت‌های عدد B بدون تغییر به ورودی جمع‌کننده می‌رسند. همچنین ورودی بیت نقلی برابر صفر می‌شود. در نتیجه عمل جمع صورت می‌گیرد.

✓ اگر $SUB=1$ باشد، بیت‌های عدد B متمم شده و به ورودی جمع‌کننده می‌رسند. همچنین با توجه به این که ورودی بیت نقلی برابر یک است، عدد B متمم دو می‌شود. با جمع عدد A و متمم ۲ عدد B، عمل تفریق (A-B) انجام می‌شود.

■ مدار انواع شیفت رجیستر

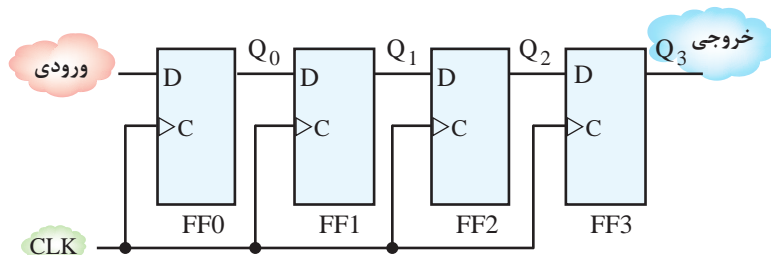
ثبات انتقالی یا شیفت رجیستر (Shift Register) یک واحد مدار منطقی ترتیبی است که از تعدادی فلیپ فلاپ ساخته شده است و با جابه‌جا کردن بیت‌های داده به سمت راست یا چپ، امکان کار با آنها را فراهم می‌کند. انتقال اطلاعات در شیفت رجیستر از طریق ورودی‌های اصلی فلیپ فلاپ یا ورودی‌های Preset و Clear صورت می‌گیرد. هر فلیپ فلاپ به کار رفته در شیفت رجیستر، یک سلول حافظه است و امکان نگهداری یک بیت داده را دارد.

شیفت رجیسترها امکان دریافت یا ارسال داده‌ها به صورت سری یا موازی را دارند. بنابراین می‌توان شیفت رجیسترها را به چهار دسته کلی تقسیم کرد:

شیفت رجیستر ورودی سری - خروجی سری

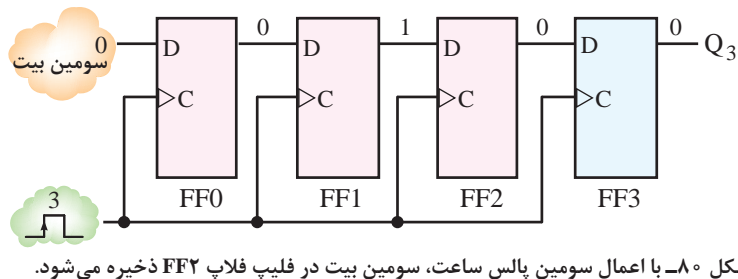
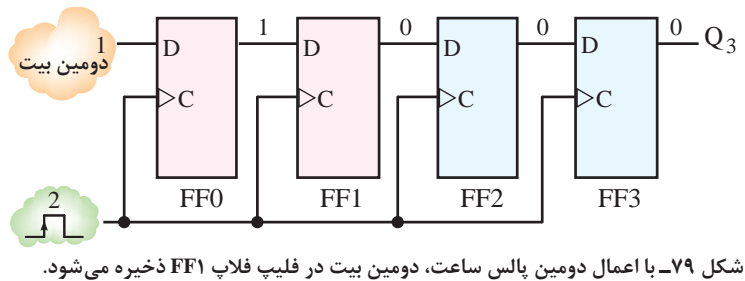
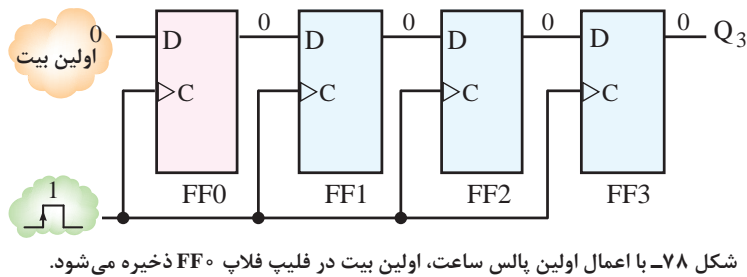
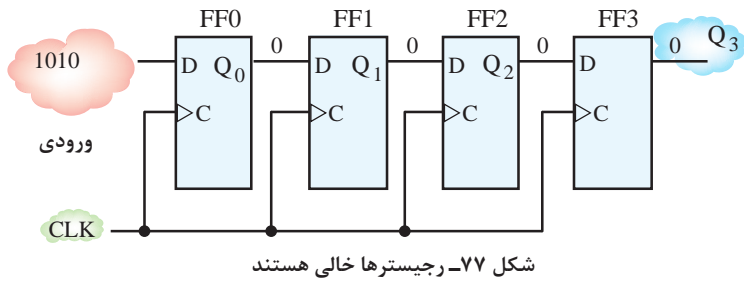
(Serial Input Serial Output-SISO)

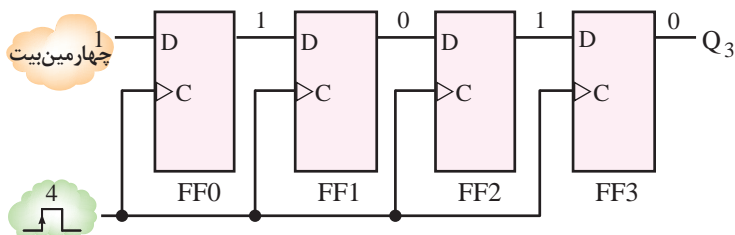
□ در این نوع شیفت رجیستر، داده‌ها به صورت سری وارد شده و به صورت سری نیز خارج می‌شوند. با اعمال هر پالس ساعت، داده‌ها یک خانه به سمت راست انتقال (شیفت) پیدا می‌کنند. مثلاً اگر بخواهیم یک داده چهار بیتی را در یک شیفت رجیستر SISO چهار بیتی ذخیره و ارسال کنیم، برای ذخیره‌سازی همه بیت‌ها به چهار پالس ساعت نیاز داریم. برای ارسال داده‌ها نیز نیاز به چهار پالس ساعت است تا داده‌ها به طور کامل خارج شوند. در شکل ۷۶ بلوک دیاگرام شیفت رجیستر SISO چهار بیتی رسم شده است.



شکل ۷۶

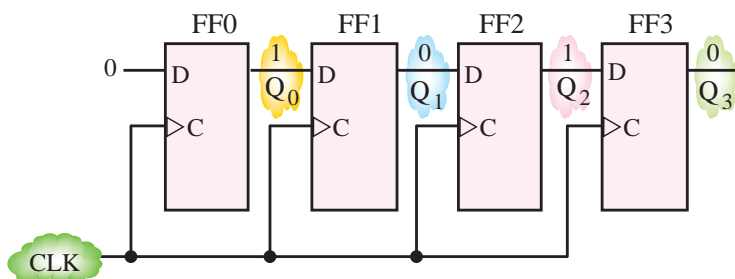
- ✓ برای بررسی عملکرد این شیفت رجیستر فرض کنید می‌خواهیم عدد دودویی ۴ بیتی ۱۰۱۰ را در شیفت رجیستر ذخیره و سپس از آن خارج کنیم.
- ✓ مراحل ذخیره کردن بیت‌ها در داخل رجیسترها در شکل‌های ۷۷ تا ۸۱ نشان داده شده است.



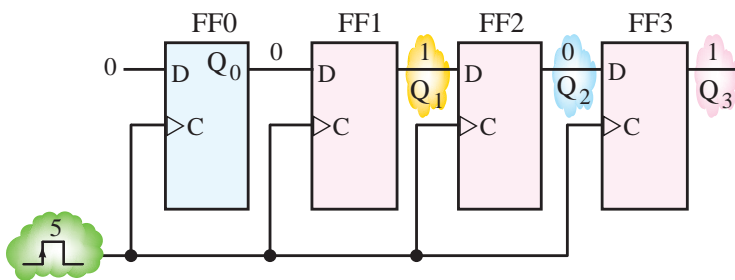


شکل ۸۱- با اعمال چهارمین پالس ساعت، چهارمین بیت در فلیپ فلاپ FF۳ ذخیره می شود.

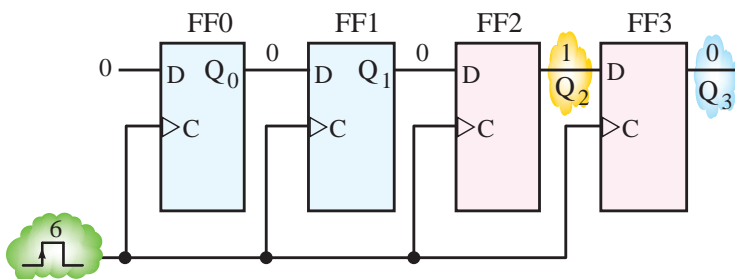
✓ در شکل های ۸۲ تا ۸۶ مراحل خالی شدن شیفت رجیستر را مشاهده می کنید.



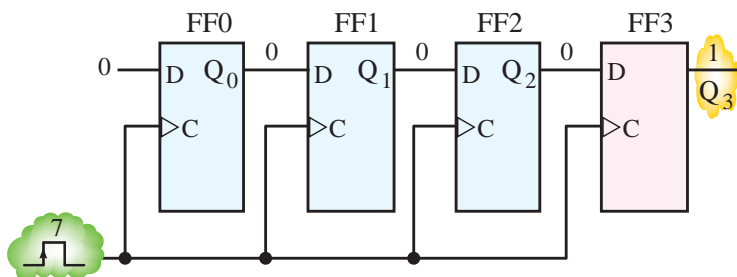
شکل ۸۲- حافظ شیفت رجیستر پر است



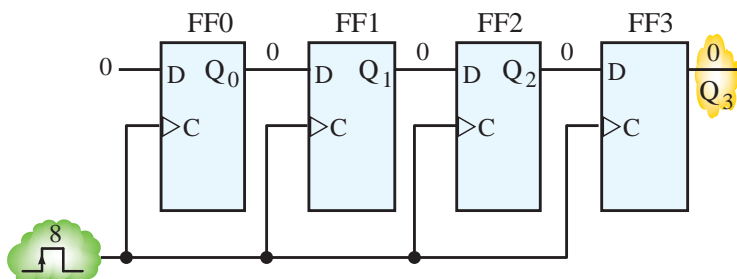
شکل ۸۳- با اعمال پنجمین پالس ساعت، اولین بیت (۰) از شیفت رجیستر خارج می شود.



شکل ۸۴- با اعمال ششمین پالس ساعت، اولین بیت (۱) از شیفت رجیستر خارج می شود.



شکل ۸۵- با اعمال هفتمین پالس ساعت، اولین بیت (۱) از شیفت رجیستر خارج می شود.



شکل ۸۶- با اعمال هشتمین پالس ساعت، اولین بیت (۰) از شیفت رجیستر خارج می شود.

✓ شکل ۸۷ مراحل ورود و خروج عدد چهار بیتی ۰۱۰۱ را در شیفت رجیستر SISO نشان می دهد.

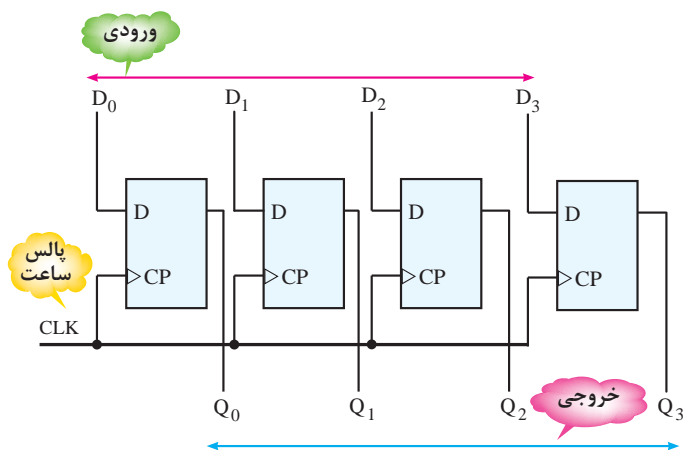
شیفت رجیستر ورودی موازی - خروجی موازی (Parallel Input Parallel Output-PIPO)

	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	0	1	0
5	0	1	0	1
6	0	0	1	0
7	0	0	0	1
8	0	0	0	0

شکل ۸۷

✓ در شیفت رجیستر PIPO، داده ها به صورت موازی وارد شده و ذخیره می شوند. همچنین ارسال داده ها به صورت موازی صورت می گیرد. مثلاً اگر بخواهیم یک داده چهار بیتی را در یک شیفت رجیستر PIPO چهار بیتی ذخیره و ارسال کنیم، برای ذخیره سازی همه بیت ها و ارسال آنها به خروجی، تنها به یک پالس ساعت نیاز داریم.

✓ در شکل ۸۸ بلوک دیاگرام شیفت رجیستر PIPO چهار بیتی رسم شده است.



شکل ۸۸

شیفت رجیستر ورودی سری - خروجی موازی

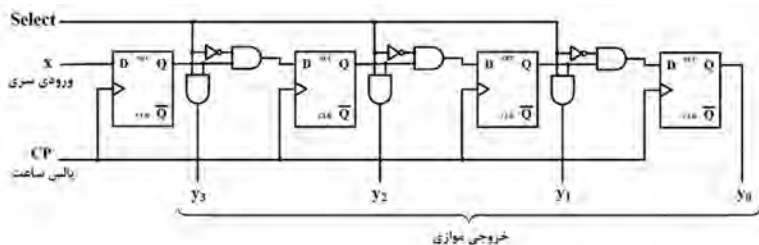
(Serial Input Parallel Output-SIPO)

✓ در شیفت رجیستر SIPO، ورودی Select برای انتخاب یکی از دو حالت انتقال (شیفت) یا ارسال داده مورد استفاده قرار می‌گیرد. در شکل ۸۹ بلوک دیاگرام شیفت رجیستر SIPO چهار بیتی رسم شده است.

✓ در این شیفت رجیستر، ابتدا ورودی Select در سطح منطقی صفر قرار می‌گیرد تا داده ورودی به صورت سری دریافت شده و با اعمال پالس ساعت منتقل شوند.

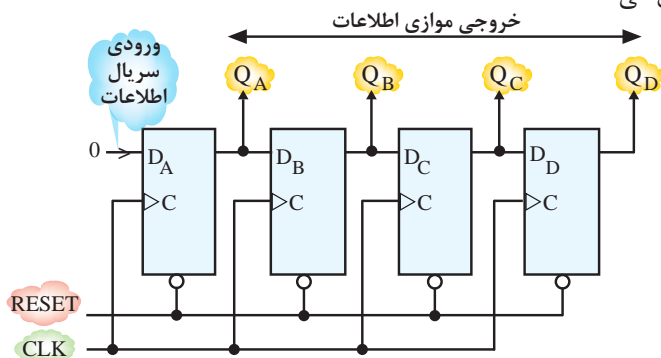
✓ پس از اعمال چهار پالس ساعت، هر چهار بیت داده ورودی ذخیره می‌شوند. پس از ذخیره‌سازی تمام بیت‌های داده ورودی در فلیپ فلاپ‌ها، ورودی Select در سطح منطقی یک قرار می‌گیرد.

✓ به این ترتیب، داده‌ها از خروجی فلیپ فلاپ‌ها به خروجی‌های شیفت رجیستر انتقال یافته و ارسال می‌شوند.



شکل ۸۹

✓ شکل ۹۰ نمونه دیگری از مدار شیفت رجیستر ورودی سری - خروجی موازی را نشان می‌دهد.



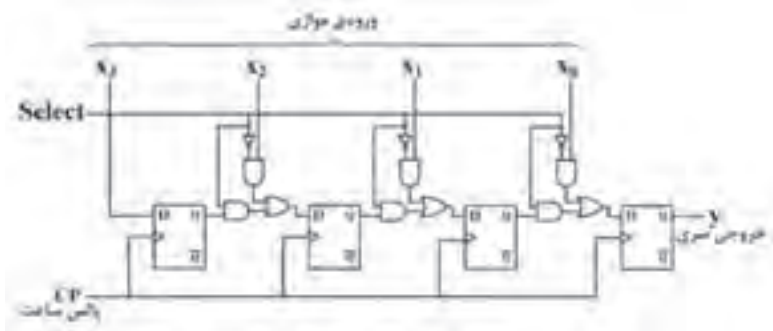
شکل ۹۰

شیفت رجیستر ورودی موازی - خروجی سری (Parallel Input Serial Output_PISO)

✓ در شیفت رجیستر PISO، ورودی Select برای انتخاب یکی از دو حالت دریافت یا انتقال داده مورد استفاده قرار می‌گیرد.

✓ در شکل ۹۱ بلوک دیاگرام شیفت رجیستر PISO چهار بیتی رسم شده است. در این شیفت رجیستر، ابتدا ورودی Select در سطح منطقی صفر قرار می‌گیرد تا هر چهار بیت داده ورودی، به صورت موازی دریافت و در ورودی فلیپ فلاپ‌ها قرار گیرند. سپس ورودی Select در سطح منطقی یک قرار می‌گیرد.

✓ به این ترتیب، ارتباط بین خروجی هر فلیپ فلاپ به ورودی فلیپ فلاپ بعدی برقرار می‌شود و با اعمال هر پالس ساعت، داده‌ها یک بیت منتقل می‌شوند. برای انتقال و ارسال هر چهار بیت به خروجی سری، چهار پالس ساعت نیاز است.

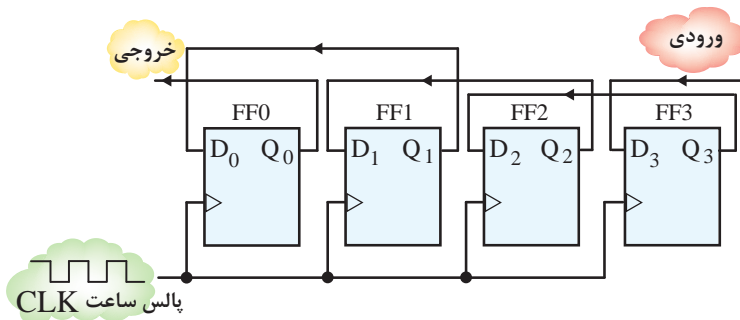


شکل ۹۱

شیفت رجیستر چپ رو / راست رو

✓ شیفت رجیسترهایی که تا کنون مورد بررسی قرار گرفت، شیفت رجیسترهای راست رو هستند. در این شیفت رجیسترها، داده از فلیپ فلاپ سمت چپ وارد شیفت رجیستر شده و از فلیپ فلاپ سمت راست خارج می شود.

✓ شیفت رجیسترها را می توان طوری طراحی کرد که داده از فلیپ فلاپ سمت راست وارد شیفت رجیستر شده و از فلیپ فلاپ سمت چپ خارج شود. به این نوع شیفت رجیسترها، شیفت رجیستر چپ رو می گویند. شکل ۹۲ یک نمونه شیفت رجیستر ورودی سری - خروجی سری چپ رو را نشان می دهد.



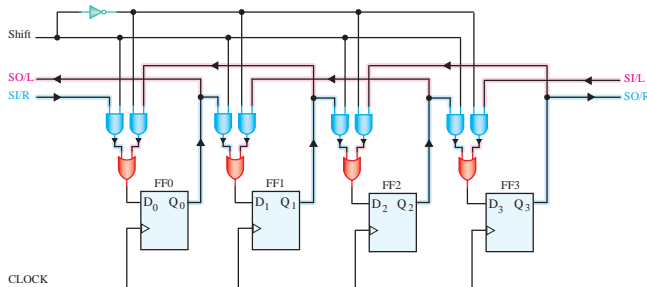
شکل ۹۲

✓ شکل ۹۳ چگونگی انتقال اطلاعات در شیفت رجیستر ورودی سری - خروجی سری چپ رو را نشان می دهد.



شکل ۹۳

✓ از ترکیب شیفت رجیسترهای راسترو و چپرو می‌توان شیفت رجیستر چپرو / راسترو ساخت. شکل ۹۴ یک نمونه شیفت رجیستر چپرو / راسترو را نشان می‌دهد.



✓ به کمک یک مالتی پلکسر ۲ به ۱ و خط کنترل Shift می‌توانیم جهت انتقال داده‌ها را تعیین کنیم. اگر خط کنترل Shift در سطح منطقی یک قرار داشته باشد، انتقال داده‌ها از چپ به راست صورت می‌گیرد و اگر در سطح منطقی صفر قرار داشته باشد، داده‌ها از راست به چپ منتقل می‌شوند.

طراحی مدار شمارنده هم‌زمان (سنکرون – Synchronous)

طراحی مدارات ترتیبی هم‌زمان

طراحی مدارهای ترتیبی هم‌زمان، با مشخص کردن جدول یا نمودار حالت مطلوب شروع می‌شود. قبل از اینکه مراحل طراحی یک مدار ترتیبی را شرح دهیم، ابتدا جدول حالت و جدول تحریک فلیپ فلاپ‌ها را معرفی می‌کنیم.

✓ جدول حالت: ترتیب زمانی ورودی‌ها، خروجی‌ها و وضعیت فلیپ فلاپ‌ها در جدولی به نام جدول حالت بیان می‌شود. این جدول، شامل قسمت‌های حالت فعلی، ورودی، حالت بعدی و خروجی است. جدول ۲۹ یک نمونه جدول حالت را نشان می‌دهد.

جدول ۲۹

خروجی		حالت بعدی		ورودی	حالت فعلی
Y		A	B	X	A B
۰	۰	۰	۰	۰	۰ ۰
۰	۰	۰	۱	۱	۰ ۰
۰	۰	۰	۰	۰	۰ ۱
۰	۰	۱	۰	۱	۰ ۱
۰	۰	۰	۰	۰	۱ ۰
۰	۰	۱	۱	۱	۱ ۰
۱	۱	۰	۰	۰	۱ ۱
۱	۱	۱	۱	۱	۱ ۱

ممکن است در بعضی از طراحی‌ها، ورودی خاصی وجود نداشته باشد و مدار تنها با اعمال پالس ساعت عمل کند. همچنین ممکن است خروجی مدار ترتیبی همان خروجی فلیپ فلاپ‌ها باشد.

✓ جدول تحریک فلیپ فلاپ‌ها: جدول تحریک فلیپ فلاپ‌ها را می‌توان براساس جدول درستی فلیپ فلاپ‌ها و با توجه به تغییرات خروجی آنها در هنگام اعمال پالس ساعت، رسم کرد. در جدول ۳۰، جدول تحریک انواع فلیپ فلاپ‌ها رسم شده است.

جدول ۳۰- جدول تحریک فلیپ فلاپ‌ها

ورودی‌های مورد نیاز		تغییر حالت	
J	K	Q(t)	Q(t+1)
0	d	0	0
1	d	0	1
d	1	1	0
d	0	1	1

ورودی‌های مورد نیاز		تغییر حالت	
S	R	Q(t)	Q(t+1)
0	D	0	0
1	0	0	1
0	1	1	0
d	0	1	1

ورودی‌های مورد نیاز		تغییر حالت	
T	Q(t)	Q(t+1)	
0	0	0	0
1	0	1	0
1	1	0	1
0	1	1	1

ورودی‌های مورد نیاز		تغییر حالت	
D	Q(t)	Q(t+1)	
0	0	0	0
1	0	1	0
0	1	0	1
1	1	1	1

✓ مراحل طراحی مدار ترتیبی عبارت‌اند از:

- ✓ گام ۱: با توجه به توصیف مسئله، جدول حالت را رسم می‌کنیم.
- ✓ گام ۲: با توجه به جدول حالت‌ها، تعداد فلیپ فلاپ‌های مورد نیاز را مشخص می‌کنیم. تعداد فلیپ فلاپ‌های مورد نیاز به تعداد تغییر حالت‌ها بستگی دارد. اگر

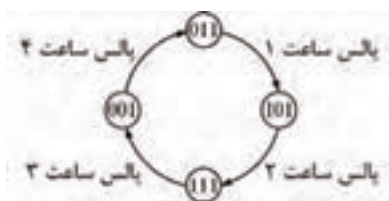
تعداد تغییر حالت‌ها برابر با 2^n باشد، به n فلیپ فلاپ نیاز داریم. نوع فلیپ فلاپ‌ها معمولاً در صورت مسئله مشخص می‌شود.

✓ **گام ۳:** با توجه به جدول حالت، جدول تحریک و خروجی را به دست می‌آوریم.
 ✓ **گام ۴:** تابع منطقی خروجی‌های جدول حالت را با توجه به وضعیت حالت فعلی و نیز ورودی جدول حالت به دست آورده و با روش‌های جبری یا به کمک نقشه کارنو ساده می‌کنیم.

✓ **گام ۵:** مدار منطقی توابع ساده شده را رسم می‌کنیم.

□ **مثال:** با استفاده از فلیپ فلاپ‌های نوع T شمارنده‌ای طراحی کنید که اعداد فرد از ۱ تا ۷ را شمارش کرده و تکرار نماید.

حل:



شکل ۹۵

✓ با اعمال هر پالس ساعت، یک شمارش انجام می‌شود و خروجی فلیپ فلاپ‌ها باید مطابق شکل ۹۵ تغییر کند. با توجه به سه بیتی بودن اعداد و تعداد حالت‌های شمارش، به سه فلیپ فلاپ نوع T نیاز داریم.

✓ **جدول حالت مدار را با توجه به عملکرد مدار و تعداد فلیپ فلاپ‌های مورد نیاز رسم می‌کنیم، جدول ۳۱.**

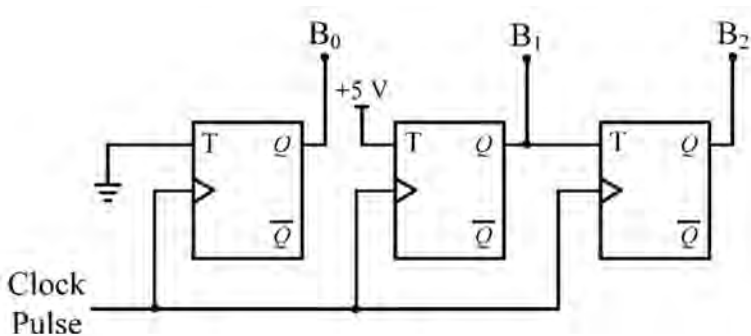
جدول ۳۱

شماره پالس ساعت	حالت فعلی			حالت بعدی			جدول تحریک فلیپ فلاپ‌ها		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
۱	0	0	1	0	1	1	0	1	0
۲	0	1	1	1	0	1	1	1	0
۳	1	0	1	1	1	1	0	1	0
۴	1	1	1	0	0	1	1	1	0

✓ تابع منطقی خروجی‌های جدول حالت، که همان ورودی فلیپ فلاپ‌ها است را به‌دست می‌آوریم:

$$\begin{cases} T_0 = 0 \\ T_1 = 1 \\ T_2 = Q_2 \end{cases}$$

✓ مشاهده می‌شود که توابع منطقی به‌دست آمده ساده بوده و نیازی به ساده‌کردن نیست.
در مرحله آخر، با توجه به توابع منطقی به‌دست آمده، مدار شمارنده را رسم می‌کنیم، (شکل ۹۶).



شکل ۹۶

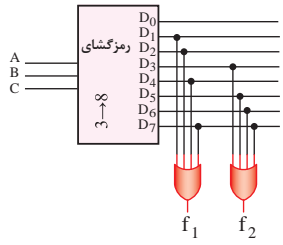
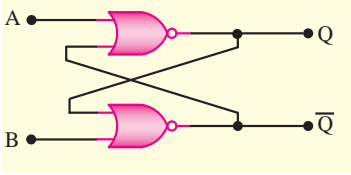
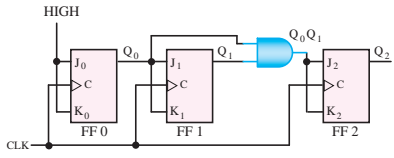
در مدار رسم شده، عدد دودویی به‌صورت $B_2B_1B_0$ است.

✓ به عنوان تمرین می‌توانید با توجه به مطالب گفته شده، شمارنده‌های زیر را طراحی کنید.

✓ با استفاده از فلیپ فلاپ‌های نوع T شمارنده‌ای طراحی کنید که اعداد صفر، ۲، ۴، ۶ و ۸ را به‌ترتیب شمارش کرده و تکرار نماید.

✓ با استفاده از فلیپ فلاپ‌های نوع T شمارنده‌ای طراحی کنید که اعداد صفر تا ۱۵ را به‌صورت صعودی شمارش کرده و تکرار کند.

کاربرگ ارزشیابی واحد یادگیری ۶

نام و نام خانوادگی هنرجو:	کد کار:	تاریخ:
<p>کار: طراحی مدارهای دیجیتالی و اجرای توابع آن</p>		
<p>آزمون نظری: سؤال براساس الگوی پرسش</p> <p>بارم آزمون: در ارزشیابی ۲۰ نمره‌ای، شایستگی ۱۵ نمره و مستمر ۵ نمره دارد که معادل ۱+۳ در ارزشیابی بر مبنای شایستگی است.</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;">  </div> <div style="width: 50%;"> <p>۱ در یک مدار ترکیبی با سه متغیر ورودی و یک خروجی، زمانی خروجی مدار «۱» است که تعداد یک‌های موجود در متغیرهای ورودی در هر ردیف، فرد باشد. مدار را طراحی کنید.</p> <p>۲ تابع خروجی مدار مقابل را به دست آورید و به کمک نقشه کارنو ساده کنید.</p> </div> </div>		
<p>۳ تابع $F_1 = \sum_m (0, 2, 4, 6)$ را به کمک یک مالتی پلکسر ۸ به ۱ اجرا کنید.</p>		
<p>۴ در شکل زیر اگر $A=1$ و $B=2$ باشد، خروجی‌های Q و \bar{Q} در چه حالت منطقی قرار دارند؟</p> <div style="text-align: center;">  </div>		
<p>۵ شمارنده شکل زیر چه اعدادی را می‌شمرد؟ توضیح دهید.</p> <div style="text-align: center;">  </div>		
<p>۶ مقایسه‌کننده ولتاژ از اجزای مدار مبدل آنالوگ به دیجیتال نیست.</p> <p style="text-align: right;"> <input type="checkbox"/> صحیح <input type="checkbox"/> غلط </p>		
<p>۷ ...</p>		