



دانشکده برق، کامپیوتر و فناوری های پیشرفته

گروه مهندسی برق - الکترونیک

دستور کار آزمایشگاه سیستم‌های دیجیتال ۱

تهیه کننده:

خانم دکتر بابایی صداقت

تاریخ تنظیم:

مهرماه ۱۴۰۳

۱. آشنایی با اصول مدارهای منطقی	۳
۲. آشنایی با بردبورد و اصول بستن گیت‌های پایه بر روی برد	۵
۳. بدست آوردن جدول صحت برخی از گیت‌های منطقی بر روی برد	۱۱
۴. طراحی توابع مختلف تنها با گیت پایه NAND	۱۲
۵. مقایسه‌گرها	۱۸
۶. جمع‌کننده و تفریق‌کننده	۱۹
۷. مالتی‌پلکسر	۲۳
۸. انکدر	۲۶
۹. سون‌سگمنت + دیکودر + شمارنده	۳۰
۱۰. دیکدر BCD و نمایشگر ۷ قسمتی	۳۲
۱۱. لچ و فلیپ‌فلاپ	۳۶
۱۲. فلیپ‌فلاپ T و شیفت رجیستر	۴۰

با تشکر از دانشجویان عباس نژاد، پورسلیمانی و تاج مراغه

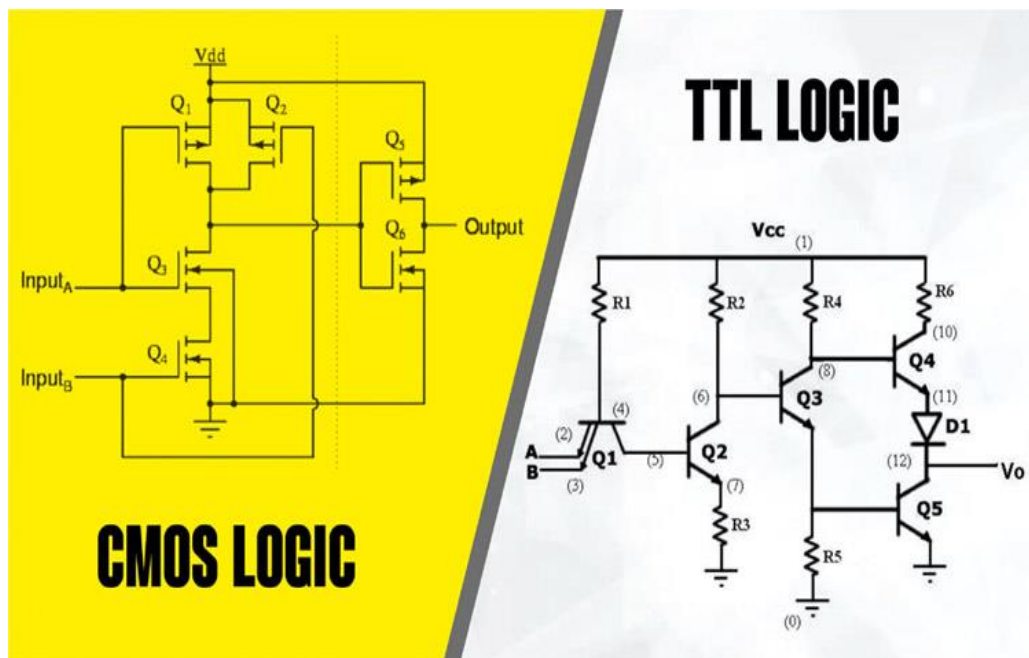
۱. آشنایی با اصول مدارهای منطقی

آزمایشگاه مدارهای منطقی یکی از بخش های اساسی در رشته های مهندسی برق و کامپیوتر است که دانشجویان را با اصول و کاربردهای مدارهای دیجیتال آشنا می کند. مدارهای دیجیتال با مدارهای مجتمع ساخته می شوند. مدار مجتمع یا IC، یک کریستال نیمه هادی است که قطعات الکترونیکی لازم برای گیت های دیجیتال را در خود دارد. تراشه در داخل یک محفظه پلاستیکی یا سرامیکی جای می گیرد. اتصالات آن با سیم های نازک به پایه های خارجی جوش داده می شود تا مدار مجتمع به وجود آید. تعداد پایه های یک IC می تواند از 14 پایه در بسته های کوچک تا 64 پایه باشد.

در آزمایشگاه، به منظور آشنایی عمیق تر با مدارهای دیجیتال و نحوه کار با آنها، از فناوری های مختلفی همچون TTL و CMOS استفاده می شود.

TTL مخفف کلمه Transistor to Transistor Logic است که از ترانزیستورهای BJT ساخته شده اند. این خانواده منطقی بر روی پالس های جریان مستقیم عمل می کند. به طور کلی، اکثر IC های مربوط به گیت های منطقی جزو این دسته هستند. مثالی از IC های منطقی TTL عبارتند از 74XX

CMOS مخفف کلمه Complementary Metal Oxide Semiconductor است و متداول ترین تکنولوژی ساخت MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) می باشد.



برخی از تفاوت های TTL و CMOS

- اجزای CMOS معمولاً گران تر از TTL هستند، اما در سطح تراشه ارزان تر هستند، زیرا اندازه آنها از اندازه TTL کوچکتر است.
- سطح تغذیه در خانواده TTL، 5 ولت و در خانواده CMOS بین 3 تا 15 ولت است.

- مدارهای CMOS نسبت به TTL ایمنی بهتری در برابر نویز دارند.
- گیت‌های اساسی در ساخت TTL، NAND و در ساخت CMOS گیت‌های NOR و NAND هستند.
- به دلیل اینکه در CMOS از FET و مدارهای TTL از BJT استفاده شده‌است، تراشه‌های CMOS بسیار سریع‌تر و کارآمدتر هستند.
- توان مصرفی مدارهای TTL بیشتر از CMOS است.

CMOS	TTL	خروجی
بیش از 2/3 تغذیه	بیش از 2V	HIGH – سطح منطقی ۱
کمتر از 1/3 تغذیه	کمتر از 0.8V	LOW – سطح منطقی ۰
بین 1/3 و 2/3 تغذیه	بین 0.8V و 2V	نامعتبر

- در زمان استفاده از خانواده TTL توجه داشته باشید که برخی از انواع آن به قرار زیر است:
 TTL استاندارد (سری 74)
 TTL سرعت بالا (سری 74H)
 TTL کم مصرف (سری 74L)
 TTL شاتکی (سری 74S)
 TTL شاتکی کم مصرف (سری 74LS)
 TTL شاتکی کم مصرف پیشرفته (سری 74ALS)

مشخصه خانواده‌های منطقی دیجیتال معمولاً با تحلیل گیت مبنا در هر خانواده با یکدیگر مقایسه می‌گردند. برای این کار باید پارامترهای توان مصرفی، تأخیر در انتشار و مصنویت در مقابل نویز را در نظر داشت.

CMOS	MOS	ECL	TTL						DTL	
			74 ALS	74LS	74S	74L	74H	74		
۲۵	۱۰۰/۵۵۰	۲	۴	۹	۳	۳۳	۶	۱۰	۳۰	تأخیر در انتشار (ns)
^{nw} ۱۰	۰/۲۵	۲۵	۱	۲	۲۰	۱	۲۲/۵	۱۰	۱۵	توان مصرفی هر دروازه (mw)
>۵۰	>۵۰	۲۵	۱۰	۱۰	۱۰	۱۰	۱۰	۱۰	۸	برون دهی (f.o)
عالی	عالی	متوسط	خوب	خوب	خوب	خوب	خوب	خوب	خوب	مصنویت در مقابل نویز

هر آی‌سی در یک بسته با پایه‌های خاص بسته‌بندی میشود. نحوه صحیح خواندن پایه‌های یک تراشه و نیز روش تشخیص اولین پایه آن و مشخصات دیگری که برای شناسایی و کار با تراشه لازم است، ابزارهای اولیه کار در آزمایشگاه مدارهای منطقی هستند. چنانکه در شکل زیر نیز مشخص است در محل پایه 1 آی‌سی علامت خاصی وجود دارد. از جمله مشخصاتی که بر روی تراشه حک می‌گردد، علامت کارخانه سازنده است که میتواند یک یا چند حرف و یا شکل باشد. نام تراشه نیز توسط شماره مشخص می‌شود. البته باید توجه داشت که روش شماره‌گذاری در شرکت‌های مختلف یکسان نبوده و در رابطه با شرکت‌های مختلف باید به برگه اطلاعات (data sheet) منتشر شده

همان شرکت مراجعه نمود. اطلاعات دیگری که ممکن است بر روی تراشه مشاهده گردد، شامل تاریخ ساخت تراشه (دو رقم برای سال و دو رقم برای هفته)، نام کشور سازنده و شماره انبار باشد.



M: سازنده آی سی / ۴۰۲۱: شماره آی سی / B: Buffered CMOS

• : نشانه شروع شمارش پایه / CP: Data Code / ۷۸: ۱۹۷۸

24th week: ۲۴

به هر آی سی شماره ای اختصاص می یابد که بر روی آن به منظور شناسایی چاپ شده است. تولیدکنندگان کتابچه هایی را چاپ نموده اند که حاوی توضیحات و کلیه اطلاعات مربوط به آی سی های تولید شده آنها است. به عنوان مثال، 6 گیت NOT را داخل یک آی سی قرار داده اند و به آن شماره 7404 را اختصاص داده اند.

نکات ایمنی

همیشه دقت کنید که ولتاژ و جریان مورد استفاده در محدوده ایمن باشد تا از آسیب به قطعات و تجهیزات جلوگیری شود. قبل از اتصال منبع تغذیه، مطمئن شوید که اتصالات صحیح و محکم هستند. مراقب باشید که هیچ گاه دو پایه مجاور را به طور ناخواسته به هم وصل نکنید، زیرا این امر ممکن است به قطعات آسیب برساند.

نحوه نتیجه گیری از آزمایش ها

برای نتیجه گیری از هر آزمایش، مراحل زیر را دنبال کنید:

مشاهده و ثبت نتایج: نتایج حاصل از پیاده سازی مدارها را با دقت مشاهده و ثبت کنید.

تحلیل نتایج: نتایج به دست آمده را تحلیل کنید و با تئوری های مطرح شده در درس مطابقت دهید.

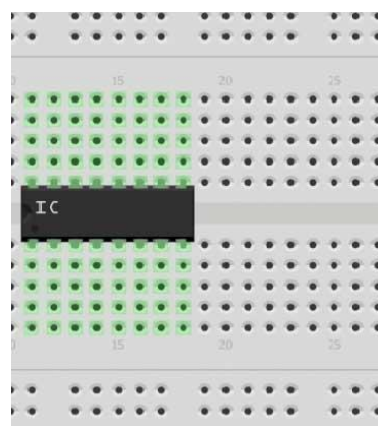
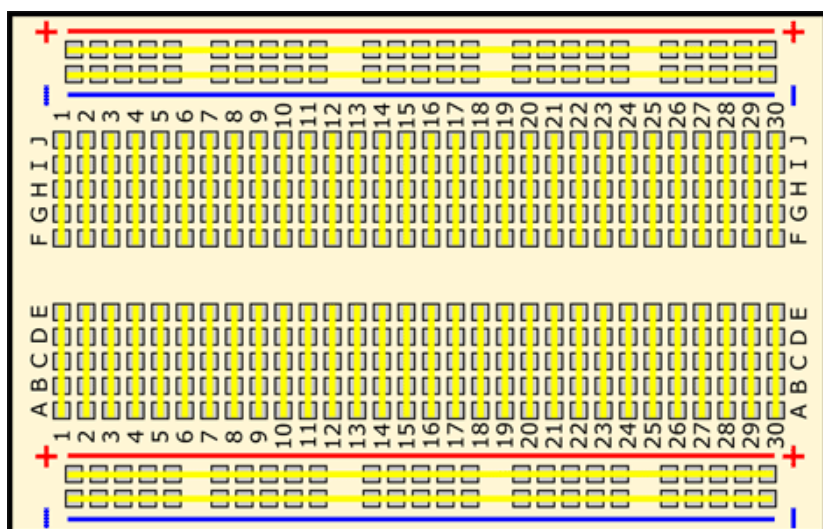
مقایسه با پیش بینی ها: نتایج عملی را با پیش بینی های تئوریک مقایسه کنید و دلیل اختلافها را بررسی کنید.

ارائه گزارش: به صورت گروهی و با هماهنگی بین اعضا، گزارشی کامل از مراحل انجام آزمایش، نتایج و تحلیل های خود تهیه کنید و در انتهای ترم به استاد مربوطه ارائه دهید.

۲. آشنایی با بردبورد و اصول بستن گیت‌های پایه بر روی برد

خلاصه آزمایش: در این آزمایش قصد داریم با ساختار بردبورد و نحوه ی بستن مدار بر روی برد آشنا شویم. سپس گیت‌های AND ، NOT و OR بر روی بردبورد می‌بندیم و جدول صحت آن‌ها را بدست می‌آوریم.

ابتدا شرح مختصری از بردبورد داریم. بردبورد (Bread Board) یک تخته معمولاً پلاستیکی است که سوراخ‌های کوچکی دارد و به راحتی می‌توان قطعات الکترونیکی را در آن قرار داد و نمونه‌های اولیه مدارها را پیاده‌سازی کرد. خطوط طولی رنگی روی بردبورها با دو رنگ قرمز و آبی به ترتیب با علامت‌های مثبت و منفی مشخص می‌شود. این خطوط باس (Bus) یا شین نامیده می‌شوند و برای اتصال زمین یا منبع تغذیه به مدار، مورد استفاده قرار می‌گیرند. معمولاً ترمینال قرمز با علامت مثبت و ترمینال آبی با علامت منفی نشان داده می‌شود. لازم به ذکر است که بین باس‌های مثبت و منفی تفاوت فیزیکی وجود ندارد و تعیین علامت آن‌ها تنها برای مدیریت ساده‌تر مدار است. هر یک از ردیف‌های درون بردبورد به صورت مجموعه‌های پنج‌تایی به هم متصل هستند (در واقع هم پتانسیل می‌باشند). این بدین معنی است که هر پنج سوراخ، یک نیم ردیف را تشکیل می‌دهند.



تعداد ردیف‌ها ۳۰ تا می‌باشد و هر نیم ردیف دارای پنج حرف انگلیسی است.

در ادامه به کاربرد بردبورد بپردازیم. برد بورد وسیله‌ای برای بستن سریع مدارهای مختلف الکترونیکی جریان و فرکانس پایین است و از آنجا که میتوان به راحتی و به سرعت بسیاری از IC ها و قطعات مختلف الکترونیکی را روی آن سوار و ارتباط‌های لازم را با سیم و بدون لحیم کاری برقرار کرد، کاربرد وسیعی در آزمایشگاه‌ها و مراکز تحقیقاتی دارد.

* هنگام نصب آی سی روی برد باید توجه کنیم که فرو رفتگی آن سمت چپ قرار داشته باشد.

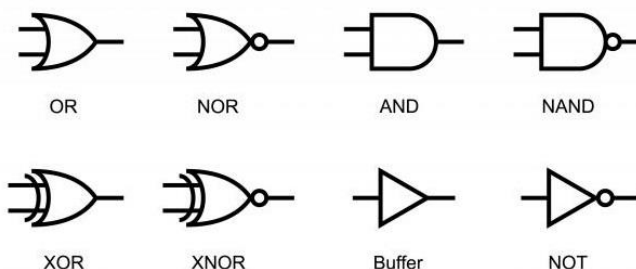
گیت‌های منطقی، به عنوان عناصر اصلی در الکترونیک دیجیتال، برای انجام عملیات منطقی بر روی سیگنال‌های دیجیتال استفاده می‌شوند. این گیت‌ها معمولاً با استفاده از ترانزیستورها و مقاومت‌ها ساخته می‌شوند و وظیفه انجام عملیات منطقی مانند جمع، ضرب، نمایش، تغییر و ترکیب سیگنال‌های دیجیتال را بر عهده دارند. یک گیت منطقی ورودی‌های خود را دریافت کرده و طبق استانداردهای منطقی، خروجی مشخصی را تولید می‌کند.

اساس کار گیت منطقی، اعداد باینری (0 و 1) است. این گیت‌ها روی یک یا چند ورودی عملیات منطقی انجام می‌دهند تا در نهایت یک خروجی حاصل شود. سطح ولتاژ ورودی و خروجی گیت‌ها، با دو مؤلفه High Voltage و Low Voltage سنجیده می‌شود. زمانی که سطح ولتاژ High باشد، به آن 1 منطقی و زمانی که سطح ولتاژ Low باشد، 0 منطقی به آن اطلاق می‌شود.

هر گیت منطقی از یک جدول صحت (Truth Table) پیروی می‌کند که ترکیب‌های ممکن ورودی و خروجی به دست آمده مربوطه را نشان می‌دهد. هفت گیت منطقی اصلی وجود دارد: AND، OR، XOR، NOT، NAND، NOR و XNOR.

که همه ی این گیت ها را می توان با سه گیت پایه AND، OR، و NOT تشکیل داد.

نماد بولی	تابع منطقی
$A \cdot B$	AND
$A + B$	OR
\bar{A}	NOT
$\overline{A \cdot B}$	NAND
$\overline{A + B}$	NOR
$(A \cdot \bar{B}) + (\bar{A} \cdot B)$ or $A \oplus B$	EX-OR
$(A \cdot B) + (\bar{A} \cdot \bar{B})$ or $\bar{A} \oplus \bar{B}$	EX-NOR

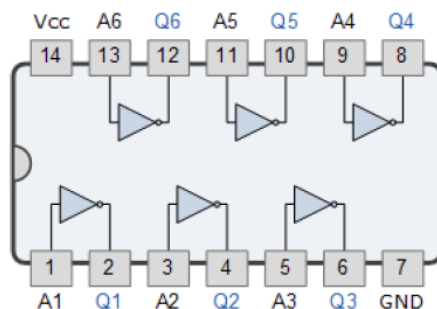


IC: تراشه، مدار یکپارچه، مدار مجتمع یا آی سی (Integrated Circuit یا Chip) به مجموعه‌ای از مدارات الکترونیکی اطلاق می‌گردد که با استفاده از مواد نیمه‌رسانا (عموماً سیلیکون همراه با میزان کنترل شده‌ای ناخالصی) در ابعادی کوچک (معمولاً کمتر از یک سانتی متر مربع) ساخته می‌شود. هر تراشه معمولاً حاوی تعداد بسیار زیادی ترانزیستور می‌باشد که با استفاده از فناوری پیچیده‌ای در داخل یک لایه از سیلیکن همگون و با ضخامتی یکنواخت و بدون ترک تزریق شده‌اند. هر IC دارای تعدادی پایه می‌باشد که هر کدام دارای شماره مشخص می‌باشد.



مرحله (۱) بدست آوردن جدول صحت گیت NOT

آی سی ۷۴۰۴ از نوع آی سی های دیجیتال است که شامل ۶ عدد گیت NOT می‌باشد. پین دیاگرام این آی سی به صورت زیر است. همانطور که مشهود است پایه ۷ به زمین و پایه ۱۴ به تغذیه متصل می‌شود. پایه های A ورودی و پایه های Q خروجی می‌باشند.



نکته: قبل از بستن مدار از سالم بودن تک تک سیم های مورد استفاده و LED با آومتر اطمینان حاصل کنید.

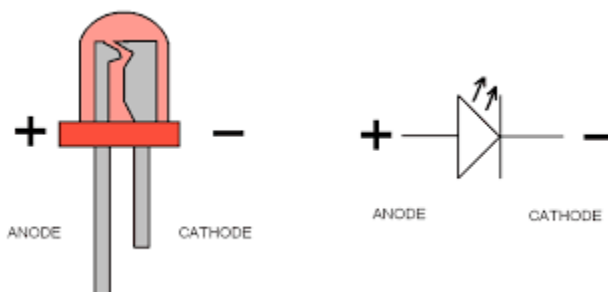
بعنوان یک قرارداد، نحوه بدست آوردن جدول صحت اولین IC مورد بحث بطور کامل توضیح داده می شود. IC7404 در مرکز بردبورد قرار می گیرد (بطوریکه هیچکدام از پایه ها هم پتانسیل نباشند). سر مثبت منبع تغذیه به ردیف بالایی و سر منفی تغذیه به ردیف پایینی برد متصل می شود. سپس پایه ۷ آی سی به ردیف پایینی و پایه ۱۴ به ردیف بالایی برد متصل می شوند (ردیف بالایی برد معرف منطق ۱ و ردیف پایین برد معرف منطق ۰ است که می توان از آنها برای قسمت های مختلف مدار استفاده کرد). حال ورودی را بعنوان مثال به پایه ۱ وصل می کنیم و با توجه به پین دیاگرام باید خروجی را از پایه ۲ بگیریم. برای ارزیابی منطقی خروجی از یک LED سری با مقاومت ۲۲۰ اهمی استفاده می شود که سر آند آن به پایه مورد نظر و سر کاتد آن به مقاومت وصل می شود (عدم استفاده از مقاومت سری و اعمال ولتاژ مستقیم به LED، بسته به مقدار ولتاژ اعمالی باعث آسیب دیدن و حتی سوختن LED خواهد شد).

تست سلامت LED:

- ۱- ابتدا اهم متر را روی حالت بوق قرار دهید.
- ۲- رابط های اهم متر را به پایه های ال ای دی وصل کنید.
- ۳- اگر اهم متر صدا داد به معنای این است که ال ای دی از داخل اتصال کوتاه می باشد و خراب است.

روش های تشخیص پایه های آند و کاتد LED:

عموما پایه ی بلندتر معرف آند می باشد، اما در صورتیکه پایه های آن کوتاه شده باشند، می توان با استفاده از شکل زیر پایه های LED را تشخیص داد.

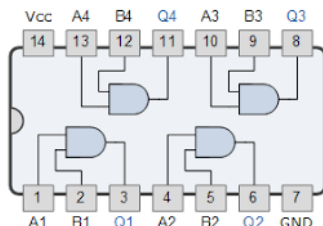


۱- جدول زیر را بر اساس ورودی اعمالی پر کنید. اثر مدار باز بودن ورودی را چگونه تفسیر می کنید؟

Input	Output
Low	
High	
Open Circuit(OC)	

مرحله ۲) بدست آوردن جدول صحت گیت AND

آی سی ۷۴۰۸ شامل ۴ عدد گیت AND دو ورودی می باشد. پین دیاگرام این آی سی به صورت زیر است. همانطور که مشهود است پایه ۷ به زمین و پایه ۱۴ به تغذیه متصل می شود. پایه های A, B, ورودی و پایه های Q خروجی می باشند.

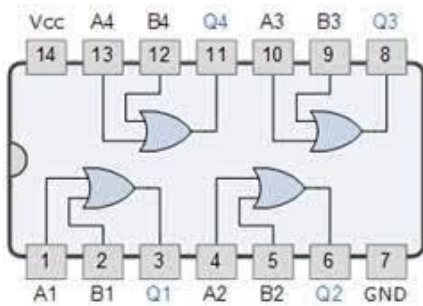


۲- جدول زیر را براساس ورودی های اعمال شده پر کنید.

A	B	Output
L	L	
L	H	
H	L	
H	H	
OC	OC	
L	OC	
OC	L	
H	OC	
OC	H	

مرحله 3) بدست آوردن جدول صحت گیت OR

آی سی ۷۴۳۲ دارای ۱۴ پایه است که شامل ۴ عدد گیت OR دو ورودی می باشد. یونیت شامل ۸ ورودی و ۴ خروجی می باشد. پین دیاگرام این آی سی به صورت زیر است. همانطور که مشهود است پایه ۷ به زمین و پایه ۱۴ به تغذیه متصل می شود. پایه های A, B ورودی و پایه های Q خروجی می باشند.



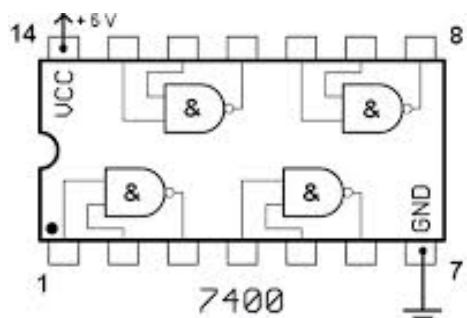
۳- جدول زیر را براساس ورودی های اعمال شده پر کنید.

A	B	Output
L	L	
L	H	
H	L	
H	H	
OC	OC	

L	OC	
OC	L	
H	OC	
OC	H	

مرحله ۴) بدست آوردن جدول صحت گیت NAND

آی سی 7400، دارای ۱۴ پایه و شامل چهار گیت NAND دو ورودی می باشد که بین دیاگرام آن در زیر نشان داده شده است. گیت NAND منطقی به عنوان گیت کلی (UNIVERSAL GATE) طبقه بندی می شود زیرا یکی از متداول ترین انواع گیت منطقی است.



۴- جدول زیر را براساس ورودی های اعمال شده پر کنید.

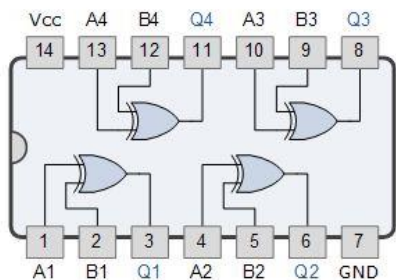
A	B	Output
L	L	
L	H	
H	L	
H	H	

۳. بدست آوردن جدول صحت برخی از گیت‌های منطقی بر روی برد

خلاصه آزمایش: در این آزمایش آی سی مربوط به گیت‌های NAND، NOR و XOR بر روی بردبرد بسته می‌شوند و جدول صحت و پین دیاگرام آن‌ها را بدست می‌آورید.

مرحله ۱) بدست آوردن جدول صحت گیت XOR

آی سی 7486 از سری آی سی های TTL شامل ۴ عدد گیت XOR دو ورودی می باشد که دارای ۸ ورودی و ۴ خروجی است. وضعیت خروجی بر اساس حالت ورودی ها تعیین می شود. در شکل زیر نمای شماتیک پایه های IC7486 نشان داده شده است.



۱- جدول زیر را براساس ورودی های اعمال شده پر کنید.

A	B	Output
L	L	
L	H	
H	L	
H	H	

مرحله ۲) بدست آوردن جدول صحت IC7402

۲- آی سی 7402 را بر روی برد ببندید. ورودی ها را به پایه‌های ۲ و ۳ اعمال کنید و خروجی را از پایه ۱ بگیرید. با استفاده از جدول صحت بدست آمده، تابع را بدست آورید. سپس پین دیاگرام آن را بدست آورید.

پایه ۲	پایه ۳	خروجی- پایه ۱
۰	۰	
۰	۱	
۱	۰	
۱	۱	

مرحله ۳) بدست آوردن پین دیاگرام و جدول صحت IC7410

۳- آی سی 7410 را بر روی برد ببندید. ورودی ها را به پایه های 1, 2, 13 اعمال کنید و خروجی را از پایه ی ۱۲ بگیرید. با استفاده از جدول صحت بدست آمده، تابع عملکرد آن را بدست آورید. سپس پین دیاگرام این آی سی را بدست آورید.

A	B	C	Output
۰	۰	۰	
۰	۰	۱	
۰	۱	۰	
۰	۱	۱	
۱	۰	۰	
۱	۰	۱	
۱	۱	۰	
۱	۱	۱	

مرحله ۴) بدست آوردن پین دیاگرام و جدول صحت IC7420

۴- آی سی 7420 را بر روی برد ببندید. ورودی ها را به پایه های 1, 2, 4, 5 اعمال کنید و خروجی را از پایه ی 6 بگیرید. با استفاده از جدول صحت بدست آمده، تابع عملکرد آن را بدست آورید. سپس پین دیاگرام این آی سی را بدست آورید.

A	B	C	D	Output
۰	۰	۰	۰	
۰	۰	۰	۱	
۰	۰	۱	۰	
۰	۰	۱	۱	
۰	۱	۰	۰	
۰	۱	۰	۱	
۰	۱	۱	۰	
۰	۱	۱	۱	
۱	۰	۰	۰	
۱	۰	۰	۱	
۱	۰	۱	۰	
۱	۰	۱	۱	
۱	۱	۰	۰	
۱	۱	۰	۱	
۱	۱	۱	۰	
۱	۱	۱	۱	

۴. طراحی توابع مختلف تنها با گیت پایه‌ی NAND

خلاصه آزمایش: مدارهای دیجیتال اغلب به جای اینکه با گیت‌های AND, OR ساخته شوند، با گیت‌های NAND, NOR ساخته می‌شوند. ساختن گیت‌های NAND, NOR ساده‌تر بوده و به‌عنوان گیت‌های پایه در تمام خانواده‌های آی‌سی‌های منطقی به کار می‌روند.

بدلیل مزایای موجود، قاعده‌ی کلی برای بدست آوردن دیاگرام منطقی NAND از یک تابع بولی به شرح زیر است:

- ۱- تابع را ساده کرده و آن را به فرم جمع حاصلضرب‌ها بنویسید.
- ۲- برای هر جمله ضرب موجود در تابع که حداقل دارای دو متغیر است یک گیت NAND بکشید. ورودی هر گیت NAND متغیرهای آن جمله هستند. این مجموع، گیت‌های طبقه اول را تشکیل می‌دهند.
- ۳- در طبقه دوم یک گیت NAND با ورودی‌هایی که از خروجی‌های طبقه اول می‌آیند، بکشید.
- ۴- یک جمله تک‌متغیری در طبقه اول، نیازمند یک معکوس کننده است.

تابع NOR دوگان تابع NAND است. به همین دلیل همه روال‌ها و قواعد منطقی NOR، دوگان روال و قواعد منطقی NAND می‌باشد. از گیت‌های NAND برای تولید هر نوع عملکرد گیت منطقی نیز استفاده کرد و در عمل گیت NAND اساس بیشتر مدارهای منطقی عملی است.

مرحله ۱) پیاده‌سازی گیت‌های NOT, AND, OR, XOR با استفاده از IC7400

1- با استفاده از چمدان آماده بر روی میز کارتان و IC7400، عملکرد گیت‌های NOT, AND, OR, XOR را پیاده‌سازی کنید: ابتدا به‌صورت شماتیک طراحی کنید و سپس بعد از بستن مدار، جدول صحت آن را تکمیل کرده و درستی مدار بسته شده را بررسی کنید.

الف) NOT

Input	Output
۰	
۱	

ب) AND

A	B	Output
۰	۰	
۰	۱	
۱	۰	
۱	۱	

OR (ج)

A	B	Output
۰	۰	
۰	۱	
۱	۰	
۱	۱	

XOR (د)

A	B	Output
۰	۰	
۰	۱	
۱	۰	
۱	۱	

مرحله ۲) پیاده‌سازی عملکرد گیت‌های AND, NAND سه ورودی با استفاده از IC7400

۲-الف) ابتدا شماتیک مداری AND سه ورودی را با استفاده از AND دو ورودی ترسیم کنید و سپس جدول صحت آن را تکمیل کنید.

A	B	C	Output
۰	۰	۰	
۰	۰	۱	
۰	۱	۰	
۰	۱	۱	
۱	۰	۰	
۱	۰	۱	
۱	۱	۰	
۱	۱	۱	

۲-ب) ابتدا شماتیک مداری NAND سه ورودی را با استفاده از NAND دو ورودی ترسیم کنید و سپس جدول صحت آن را تکمیل کنید.

A	B	C	Output
۰	۰	۰	
۰	۰	۱	
۰	۱	۰	
۰	۱	۱	
۱	۰	۰	
۱	۰	۱	
۱	۱	۰	
۱	۱	۱	

مرحله ۳) پیاده‌سازی توابع بولی داده شده با استفاده از IC7400

۳-الف) تابع بولی $F_1(A, B, C, D) = \sum(0,1,4,5,8,9,10,12,13,14)$ را فقط با استفاده از گیت NAND دو ورودی پیاده‌سازی کنید و بعد از بستن مدار، جدول درستی را تکمیل کرده و درستی مدار پیاده‌سازی شده را بررسی کنید.

A	B	C	D	Output
۰	۰	۰	۰	
۰	۰	۰	۱	
۰	۰	۱	۰	
۰	۰	۱	۱	
۰	۱	۰	۰	
۰	۱	۰	۱	
۰	۱	۱	۰	
۰	۱	۱	۱	
۱	۰	۰	۰	
۱	۰	۰	۱	
۱	۰	۱	۰	
۱	۰	۱	۱	
۱	۱	۰	۰	
۱	۱	۰	۱	
۱	۱	۱	۰	
۱	۱	۱	۱	

۳-ب) تابع بولی $F_2(A, B, C, D) = \sum(0,1,2,3,4,5,10,11)$ را فقط با استفاده از گیت NAND دو ورودی پیاده‌سازی کنید و بعد از بستن مدار، جدول درستی را تکمیل کرده و درستی مدار پیاده‌سازی شده را بررسی کنید.

A	B	C	D	Output
۰	۰	۰	۰	
۰	۰	۰	۱	
۰	۰	۱	۰	
۰	۰	۱	۱	
۰	۱	۰	۰	
۰	۱	۰	۱	
۰	۱	۱	۰	
۰	۱	۱	۱	
۱	۰	۰	۰	
۱	۰	۰	۱	
۱	۰	۱	۰	
۱	۰	۱	۱	
۱	۱	۰	۰	
۱	۱	۰	۱	
۱	۱	۱	۰	
۱	۱	۱	۱	

۳-ج) تابع $F_3 = CB + AD$ را با استفاده از گیت NAND دو ورودی طراحی کنید و بعد از بستن مدار، جدول درستی را تکمیل کرده و درستی مدار پیاده‌سازی شده را بررسی کنید.

A	B	C	D	Output
۰	۰	۰	۰	
۰	۰	۰	۱	
۰	۰	۱	۰	
۰	۰	۱	۱	
۰	۱	۰	۰	
۰	۱	۰	۱	
۰	۱	۱	۰	
۰	۱	۱	۱	
۱	۰	۰	۰	
۱	۰	۰	۱	
۱	۰	۱	۰	
۱	۰	۱	۱	

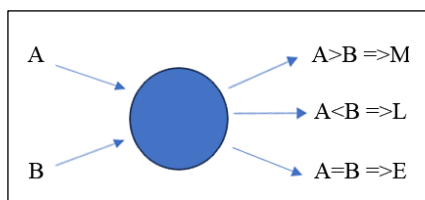
۱	۰	۱	۱	
۱	۱	۰	۰	
۱	۱	۰	۱	
۱	۱	۱	۰	
۱	۱	۱	۱	

خلاصه آزمایش: مدار یک مقایسه گر تک بیتی طراحی و پیاده سازی می شود و در مرحله ی بعد عملکرد IC7485 بررسی می شود.

مرحله (۱) طراحی و پیاده سازی مقایسه گر تک بیتی

۱- با توجه عملکردی که از یک مقایسه کننده تک بیتی انتظار داریم، بعد از ترسیم جدول صحت، مدار آن را طراحی کنید و سپس با استفاده از IC7486, IC7400 مدار آن را پیاده سازی کرده و صحت مدار طراحی شده را چک کنید.

عملکرد مورد انتظار:



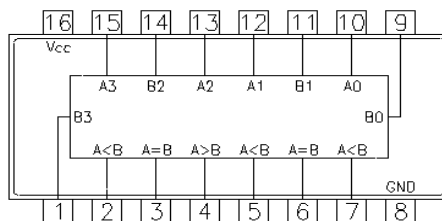
A	B	M	L	E
۰	۰			
۰	۱			
۱	۰			
۱	۱			

مرحله (۲) بررسی عملکرد مقایسه گر چهار بیتی

در مقایسه اعداد باینری یا BCD بزرگ. همواره برای کاهش زمان ابتدا با ارزش ترین بیت ها (MSB) با یکدیگر مقایسه می شوند. اگر این دو بیت مساوی بودند، به ترتیب تا رسیدن به کم ارزش ترین بیت (LSB) بیت های کم ارزش تر با یکدیگر مقایسه می شوند. در صورت برابری همه ی بیت ها، دو عدد مساوی بوده و خروجی $A=B$ فعال می شود. اما اگر در هر مرحله ای از این فرآیند دو بیت با یکدیگر برابر نبوده و مثلاً $A > B$ یا $A < B$ باشد، رابطه ی بین دو عدد نیز همان گونه خواهد بود.

۲-الف) با استفاده از بلوک های مقایسه گر تک بیتی، یک مقایسه گر چهار بیتی طراحی کنید.

۲-ب) IC7485 یک آی سی ۱۶ پایه است که در واقع مقایسه گر چهار بیتی می باشد که دو عدد چهار بیتی $A_3A_2A_1A_0$ و $B_3B_2B_1B_0$ را با هم مقایسه می کند. ۴ بیت A و ۴ بیت B را ورودی بدهیم ۳ بیت خروجی می گیریم که یا $A > B$ یا $A=B$ یا $A < B$. پین دیاگرام این



7485
4-Bit Magnitude Comparator

آی سی در شکل نشان داده شده است. تفاوت پایه های 2,3,4 با پایه های 5,6,7 چیست و چه کاربردی دارند؟ با استفاده از IC7485 چگونه می توان مقایسه گر هشت بیتی طراحی کرد؟

۲-ج) با اعمال ورودی ها متناسب با جدول داده شده، جدول را تکمیل کنید.

A ₃	A ₂	A ₁	A ₀	B ₃	B ₂	B ₁	B ₀	E	M	L
0	0	1	0	1	0	0	0			
1	0	0	0	0	0	1	0			
1	1	0	0	0	0	1	1			
1	1	0	0	1	1	1	0			
0	0	1	1	1	1	1	0			
0	1	1	1	1	1	1	0			
0	1	1	0	0	0	0	1			
1	1	1	1	1	1	1	1			
1	0	0	0	0	0	0	0			
1	0	.	.	1	1	۱	1			
۱	۱	۱			
۱	۱	۱	۱			
۱	1	0	1	1	۱	.	1			
0	1	0	1	1	1	1	1			
0	1	0	0	1	1	0	1			
0	1	0	0	0	1	0	0			

خلاصه آزمایش: ابتدا با آی سی های ۷۴۰۰ و ۷۴۸۶ یک جمع کننده تک بیتی طراحی می کنیم و سپس با استفاده از تراشه 7483 که یک جمع کننده 4 بیتی است، جدول صحت درستی را برای آن تعیین می کنیم.

کامپیوترهای دیجیتال اعمال پردازش اطلاعات گوناگونی را انجام می دهند که از آن میان، اعمال مختلف حسابی جز اعمال اصلی به شمار می روند و بدون شک اساسی ترین عمل حساب، جمع دو رقم دودویی است. این جمع ساده از چهار عمل مقدماتی $0+0=0$ و $0+1=1$ و $1+0=1$ و $1+1=10$ تشکیل می شود، که حاصل سه جمع اول عددی تک رقمی و آخرین عمل حاصل دو رقمی دارد که رقم مرتبه ی بالای آن را نقلی یا carry می نامند. مداري که جمع دو رقم دودویی ورودی را انجام داده و حاصل جمع و رقم نقلی را تولید کند، نیم جمع کننده یا Half-adder نامیده می شود. و مداري که جمع سه رقم دودویی (دو رقم دودویی و رقم نقلی حاصل از ستون قبل) را انجام داده و حاصل جمع و نقلی تولید می کند، تمام جمع کننده یا Full-adder نامیده می شود.

یک تمام جمع کننده، جمع مقادیر وارد شده را حساب و همچنین مقادیر نقلی شده خروجی را نمایش می دهد. یک تمام جمع کننده یک بیتی، سه عدد یک بیتی را جمع می کند که اغلب به صورت A و B و C_{in} نوشته می شود. A و B عملوندها هستند و C_{in} که بیت نقلی که از جمع قبلی (در صورت وجود) است.



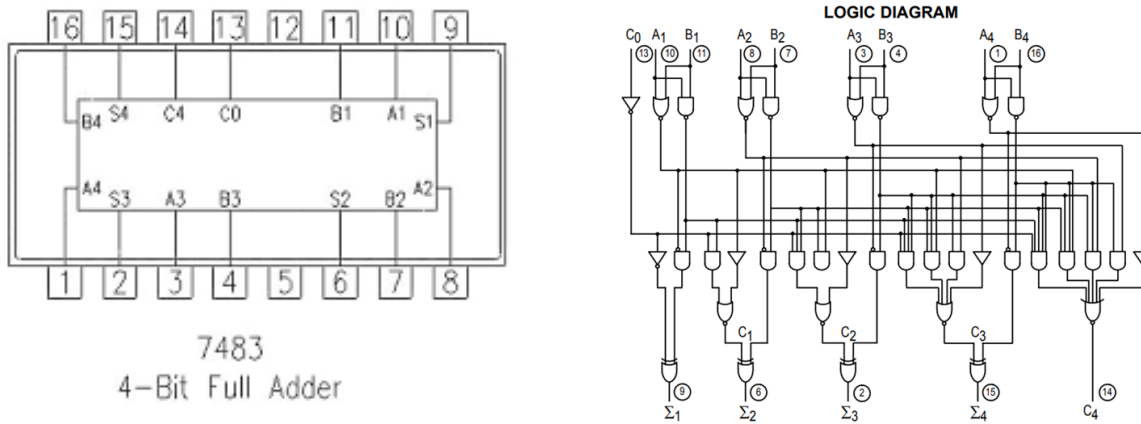
مرحله (۱) پیاده سازی تمام جمع کننده تک بیتی

۱- با استفاده از IC7410, IC7486 مدار یک تمام جمع کننده ی تک بیتی را طراحی کرده و سپس پیاده سازی کنید.

A	B	C_{in}	C_4	S
۰	۰	۰		
۰	۰	۱		
۰	۱	۰		
۰	۱	۱		
۱	۰	۰		
۱	۰	۱		
۱	۱	۰		
۱	۱	۱		

مرحله ۲) پیاده‌سازی تمام‌جمع‌کننده چهاربیتی

آی‌سی 7483 به‌عنوان تمام‌جمع‌کننده استفاده می‌شود. پین‌دیگرام و مدار داخلی آن در زیر نشان داده شده است.



۲- با استفاده از چمدان‌های روی میزکارتان و IC7483 عملکرد یک جمع‌کننده‌ی چهاربیتی را بررسی کنید. برای این کار، مطابق مقادیر داده شده در جدول زیر، جدول را تکمیل کنید.

A ₄	A ₃	A ₂	A ₁	B ₄	B ₃	B ₂	B ₁	C _{in}	C ₄	S ₄	S ₃	S ₂	S ₁
0	0	0	0	0	0	0	0	0					
0	0	0	1	0	0	0	1	0					
0	0	1	1	0	0	0	1	0					
1	1	1	1	0	0	0	1	0					
1	1	1	1	0	1	0	0	0					
0	1	1	1	1	0	0	0	0					
0	1	1	0	0	0	0	1	0					
0	1	1	1	0	1	1	1	۰					
0	1	1	0	0	0	1	0	۱					
۰	۱	۰	۰	۰	۰	۱	۰	۱					
۱	۰	۰	۰	۰	۰	۱	۱	۱					
۱	۱	۰	۰	۰	۰	۱	۱	۱					
۱	۰	۱	۰	۰	۱	۰	۰	۱					

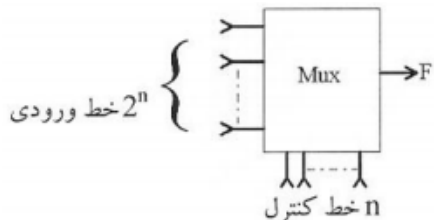
مرحله 3) پیاده‌سازی تمام جمع‌کننده - تفریق‌کننده چهاربیتی با یک بیت کنترلی

۳- با استفاده از IC7483 یک تمام جمع‌کننده و تفریق‌کننده هم‌زمان طراحی کنید، به‌صورتی‌که با استفاده از یک بیت کنترلی M بطوریکه اگر M=0 باشد، عمل جمع انجام شود و اگر M=1 باشد، عمل تفریق انجام شود. ابتدا ساختار را طراحی کرده و سپس جدول زیر را تکمیل کنید و صحت مدار خود را بررسی کنید.

A ₄	A ₃	A ₂	A ₁	B ₄	B ₃	B ₂	B ₁	M	C ₄	S ₄	S ₃	S ₂	S ₁
0	0	0	0	0	0	0	0	0					
0	0	0	1	0	0	0	1	0					
0	0	1	1	0	0	0	1	0					
1	1	1	1	0	0	0	1	0					
1	1	1	1	0	1	0	0	0					
0	1	1	1	1	0	0	0	0					
0	1	1	0	0	0	0	1	۱					
0	1	1	1	0	1	1	1	۱					
0	1	1	0	0	0	1	0	۱					
۰	۱	۰	۰	۰	۰	۱	۰	۱					
۱	۰	۰	۰	۰	۰	۱	۱	۱					
۱	۱	۰	۰	۰	۰	۱	۱	۱					
۱	۰	۱	۰	۰	۱	۰	۰	۱					

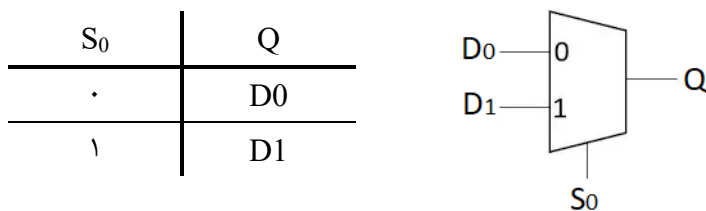
خلاصه آزمایش: ابتدا مالتی پلکسر ۲ در ۱ با استفاده از گیت‌های منطقی طراحی و پیاده‌سازی می‌شود. سپس توابع ۳ و ۴ متغیره را با استفاده از IC74151 که یک مالتی پلکسر ۸ در ۱ است، پیاده‌سازی خواهند شد و در نهایت طراحی مالتی پلکسر ۴ در ۱ و طراحی یک مالتی پلکسر ۱۶ در ۱ با استفاده از MUX4×1 خواسته شد.

مالتی پلکسر یا به اختصار MUX قطعه‌ای است که با استفاده از انتخاب‌گر یا selector، یکی از چند سیگنال ورودی را انتخاب کرده و سپس آن ورودی انتخاب شده را به یک خط خروجی هدایت می‌کند. تسهیم‌کننده‌ای با 2^n ورودی، n خط انتخاب‌گر دارد که این خطوط انتخاب مشخص می‌کنند که کدام خط ورودی باید به خط خروجی هدایت شود. از تسهیم‌کننده‌ها بیشتر برای افزایش مقدار اطلاعاتی که می‌توان در یک مدت زمان مشخص، با یک پهنای باند مشخص بر روی شبکه ارسال کرد، استفاده می‌شوند. یک تسهیم‌کننده الکترونیکی به چند سیگنال امکان می‌دهد تا از یک دستگاه یا منبع به صورت اشتراکی استفاده کنند. برای مثال می‌توان یک تبدیل‌کننده آنالوگ به دیجیتال یا یک خط ارتباطی را در بین خطوط ورودی به اشتراک گذاشت، به جای اینکه برای هر ورودی یک دستگاه مجزا در نظر گرفته شود.



مرحله (۱) پیاده‌سازی مالتی پلکسر ۲×۱

۱- برای شروع ابتدا باید مسئله را به شکل بلوک دیاگرامی در بیاوریم.



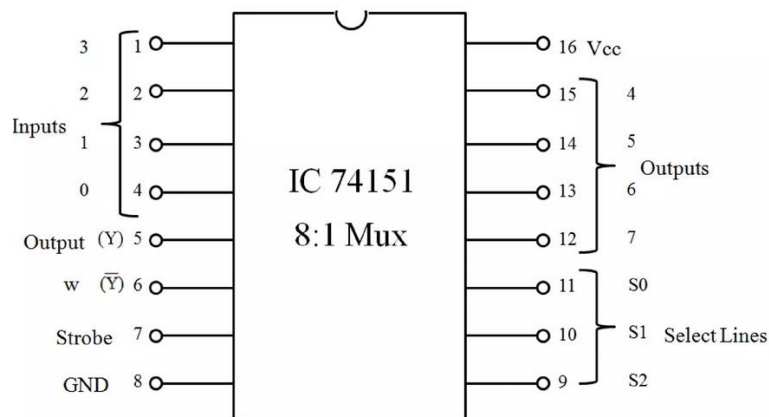
با توجه به عملکردی که از یک مالتی پلکسر ۲ در ۱ انتظار دارید، جدول زیر را تکمیل کرده و سپس تابع بولی خروجی را بدست آورید. سپس برای پیاده‌سازی تابع به کمک گیت‌های NAND، فقط با استفاده از IC7400 یک مالتی پلکسر ۲×۱ طراحی و پیاده‌سازی کنید و در نهایت صحت طراحی خود را بررسی کنید.

D1	D0	S0	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	

1	0	0	
1	0	1	
1	1	0	
1	1	1	

مرحله 2) پیاده‌سازی تابع با استفاده از MUX 8×1

توابع مختلف را با کمک مالتی پلکسرها می‌توان پیاده‌سازی کرد. آی‌سی ۷۴۱۵۱ یک مالتی پلکسر است که با استفاده از ۳ خط آدرس، یکی از ۸ سیگنال ورودی را انتخاب و به خروجی انتقال می‌دهد. پین دیاگرام آن در زیر نشان داده شده است.



۲-الف) با استفاده از IC74151 که در چمدان‌های روی میز کارتان است، توابع بولی $F_1(C,B,A)=\sum(1,3,5,7)$ و $F_2(C,B,A)=\sum(0,2,5,6,7)$ را پیاده‌سازی کنید و سپس با پر کردن جدول زیر، صحت عملکرد مدار طراحی شده را بررسی کنید. توجه شود که در این آی‌سی A کم‌ارزش‌ترین بیت و C با ارزش‌ترین بیت می‌باشد.

S ₂ =C	S ₁ =B	S ₀ =A	F ₁	F ₂
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

۲-ب) به نظر شما کدام یک از خروجی‌های 5,6 سریعتر هستند. چرا؟

۲-ج) نقش پایه‌ی هفت IC را توضیح دهید و اگر یک باشد، چه تاثیری بر خروجی مدار دارد؟

۲-د) تابع $F_3(C,B,A,D)=(0,1,3,4,8,9,15)$ را با استفاده از مالتی‌پلکسر 8 در 1 طراحی و پیاده‌سازی کنید. راهنمایی: ورودی‌های C,B,A بعنوان سلکتور و D را به‌عنوان ورودی در نظر بگیرید.

C	B	A	D	F_3
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

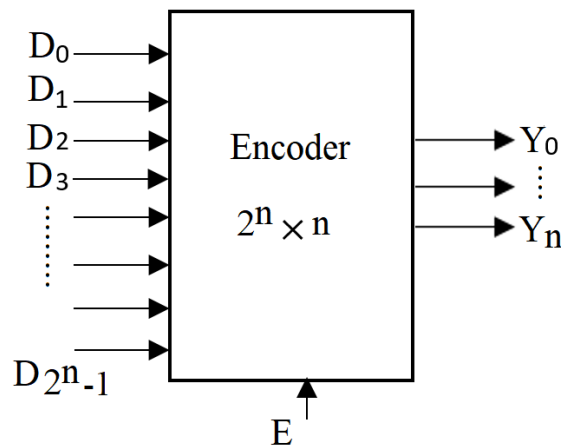
مرحله ۳) طراحی با استفاده از $MUX 4 \times 1$

۳-الف) با استفاده از مالتی‌پلکسر 4 در 1 یک مالتی‌پلکسر 16 در 1 طراحی کنید.

۳-ب) مدار داخلی یک مالتی‌پلکسر 4 در 1 را طراحی کنید.

خلاصه آزمایش: ابتدا با استفاده از گیت‌های منطقی، انکدرهای ۴ در ۲ و ۸ در ۳ طراحی و پیاده سازی می‌شود. سپس عملکرد IC 74148 بررسی می‌شود و سپس انکدر الویت ۴ در ۲ با استفاده از گیت‌های منطقی طراحی و پیاده سازی خواهد شد.

«انکدرهای دیجیتالی» (Digital Encoders) مدارهایی هستند که یک داده‌ی ورودی را گرفته و کد باینری معادل آن را در خروجی تولید می‌کنند. برخلاف مالتی‌پلکسرها که تنها داده‌ی یک خط ورودی را انتخاب کرده و آن را به یک خط خروجی ارسال یا سوئیچ می‌کنند، انکدرهای دیجیتالی یا باینری در هر لحظه از زمان بر اساس مقدار تمامی ورودی‌ها تصمیم‌گیری کرده و یک خروجی باینری معادل تولید می‌کند. لذا می‌توان گفت انکدر باینری یک مدار منطقی ترکیبی با چند ورودی است که بسته به شماره‌ی پایه‌ای از ورودی که مقدار آن ۱ منطقی است، در خروجی مقدار باینری معادل آن شماره را قرار می‌دهد. انکدرهای دیجیتالی بسته به تعداد خطوط داده‌ی ورودی، ممکن است خروجی‌های ۲ بیتی، ۳ بیتی و یا ۴ بیتی تولید کنند. به طور کلی یک انکدر n بیتی دارای 2^n خط ورودی و n خط خروجی است. مرسوم‌ترین انکدرها عبارتند از انکدرهای ۴ به ۲، ۸ به ۳ و ۱۶ به ۴.



مرحله (۱) پیاده‌سازی انکدر به کمک گیت‌های منطقی

۱-الف) با توجه به عملکردی که از یک انکدر ۴ در ۲ انتظار دارید، ابتدا جدول صحت آن را تکمیل کنید و سپس توابع متناظر با خروجی را بدست آورید. در نهایت فقط با استفاده از IC7400 یک انکدر ۴ در ۲ پیاده‌سازی کنید. راهنمایی: در هر حالت تنها یکی از ورودی‌ها می‌تواند یک باشد.

D_3	D_2	D_1	D_0	Y_1	Y_0

۱-ب) با توجه به عملکردی که از یک انکدر 8 در 3 انتظار دارید، ابتدا جدول صحت آن را تکمیل کنید و سپس توابع متناظر با خروجی را بدست آورید. در نهایت فقط با استفاده از IC7432 یک انکدر 8 در 3 پیاده‌سازی کنید.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Y ₂	Y ₁	Y ₀

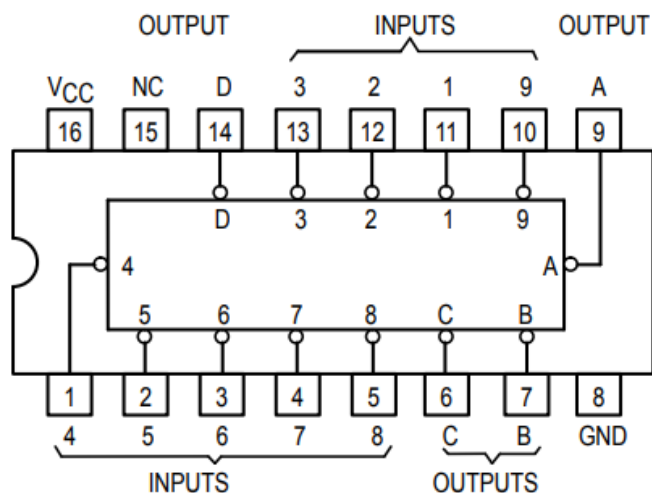
مرحله ۲) بررسی عملکرد انکدر الویت

74148

یکی از معایب اصلی انکدرهای استاندارد دیجیتال، این است؛
که در صورت وجود بیش از یک ورودی در سطح منطقی "۱"،

می‌توانند کد خروجی اشتباه، تولید نمایند. برای مثال، اگر ما دو ورودی و را، به صورت همزمان، در حالت HIGH و سطح منطقی "۱" قرار دهیم؛ خروجی نهایی ما نه "۰۱" و نه "۱۰"، بلکه برابر با "۱۱" خواهد بود؛ که یک خروجی با یک عدد باینری متفاوت از ورودی واقعی ما است. هم‌چنین، کد خروجی برای تمام منطق‌های "۰"، زمانی می‌تواند تولید شود؛ که تمام ورودی‌های آن در حالت "۰" یا ورودی برابر با "۱" باشد.

یک راه ساده برای غلبه بر این مشکل، "اولویت‌بندی" هر پین ورودی است. بنابراین، اگر بیش از یک ورودی، به‌طور همزمان، در سطح منطقی "۱"، موجود باشد؛ کد خروجی واقعی، فقط با ورودی دارای بالاترین اولویت تعیین‌شده، تطابق خواهد داشت. در نتیجه، این نوع از انکدر دیجیتال، معمولاً انکدر اولویت یا برای اختصار انکدر P- نامیده می‌شود. انکدر اولویت، با اختصاص سطح اولویت به هر ورودی، مشکلات ذکر شده در بالا را حل می‌کند. خروجی انکدرهای اولویت، مطابق با ورودی فعالی است؛ که بالاترین اولویت را دارد. پس زمانی که، یک ورودی با اولویت بالاتر موجود باشد؛ تمام ورودی‌های دیگری که دارای اولویت‌های پایین‌تری می‌باشند؛ نادیده گرفته می‌شوند. انکدرهای اولویت به شکل



آی‌سی‌های استاندارد تولید می‌شوند؛ مثلاً 74LS148 TTL یک انکدر اولویت ۸ به ۳ بیتی است که ورودی‌های آن active LOW هستند. این آی‌سی با توجه به ورودی با بالاترین اولویت یک کد ۳ بیتی در خروجی تولید می‌کند. شکل پین دیاگرام این آی‌سی در شکل نشان داده شده است.

۲-الف) ورودی‌ها را متناسب با جدول به آی‌سی اعمال کنید و خروجی‌ها را بدست آورید. D7 با ارزش‌ترین ورودی می‌باشد.

(توجه شود همانطور که در شرح آی‌سی گفته شد، برای گرفتن خروجی درست، باید خروجی را به LED های active low وصل کنید)

D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	Y ₂	Y ₁	Y ₀	\bar{E}_I	\bar{E}_O
X	X	X	X	X	X	X	X				1	
1	1	1	1	1	1	1	1				.	
X	X	X	X	X	X	X	.				.	
0	0	0	0	0	0	0	0				1	
1	0	1	0	1	0	1	0				0	
X	X	X	X	X	X	0	1				0	
X	X	X	X	X	0	1	1				0	
X	X	X	X	0	1	1	1				0	

X	X	X	0	1	1	1	0				0
X	X	X	X	1	0	0	1				0
X	X	X	0	1	1	1	1				0
X	X	X	0	1	1	0	1				0
X	X	0	1	1	1	1	1				0
X	X	0	0	1	1	1	1				0
X	0	1	1	1	1	1	1				0
X	0	0	0	1	1	1	1				0
0	1	1	1	1	1	1	1				0
0	1	0	0	1	1	0	1				0
0	0	0	0	0	0	0	0				0
0	1	0	1	0	1	0	0				0
0	1	0	1	0	1	0	1				0
0	1	0	1	0	1	0	1				۱

۲-ب) نقش پایه‌های E_i, E_o را شرح دهید.

۲-ج) یک انکدر الویت ۴ در ۲ با منطق high active طراحی کنید، بطوری که اگر ورودی‌ها اعمال شوند، در خروجی کد متناسب با بارزش‌ترین ورودی ایجاد شود.

راهنمایی ۱: در هر حالت تعدادی یا همه‌ی ورودی‌ها می‌توانند مقدار یک باشد.

راهنمایی ۲: برای خروجی‌ها بطور جداگانه، جدول کارنو تشکیل دهید و بر حسب ورودی‌ها توابع را بدست آورید.

راهنمایی ۳: در حالتی که همه ورودی‌ها صفر است، خروجی نا معتبر خواهد بود، در آن صورت نیاز به تعریف خروجی V داریم که نشان‌دهنده‌ی این حالت باشد.

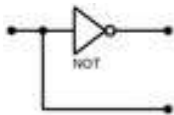
D_3	D_2	D_1	D_0	Y_1	Y_0	V
۰	۰	۰	۰			
۰	۰	۰	۱			

۰	۰	۱	۰			
۰	۰	۱	۱			
۰	۱	۰	۰			
۰	۱	۰	۱			
۰	۱	۱	۰			
۰	۱	۱	۱			
۱	۰	۰	۰			
۱	۰	۰	۱			
۱	۰	۱	۰			
۱	۰	۱	۱			
۱	۱	۰	۰			
۱	۱	۰	۱			
۱	۱	۱	۰			
۱	۱	۱	۱			
۱	۱	۱	0			
۱	۱	۱	1			

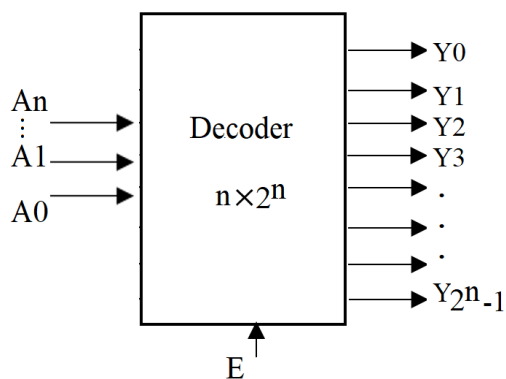
۹. دیکدر (Decoder)

خلاصه آزمایش: ابتدا با استفاده از گیت‌های منطقی، دیکدر ۲ به ۴ طراحی و پیاده سازی می‌شود. سپس عملکرد IC 74138 بررسی می‌شود و سپس با استفاده از دیکدر ۳ به ۸ (IC74138)، چند تابع بولی طراحی و پیاده‌سازی خواهد شد.

دیکدرهای باینری (Binary Decoders) دسته‌ای از مدارهای منطقی ترکیبی هستند که در ساخت آنها از گیت‌های منطقی پایه استفاده شده و عملکرد آنها دقیقاً عکس انکدرهاست. کلمه‌ی Decoder به معنای ترجمه یا رمزگشایی اطلاعات رمزنگاری شده از یک قالب به قالبی دیگر است. یک دیکدر ۱ به ۲ را می‌توان با استفاده از یک گیت NOT مطابق شکل زیر طراحی کرد.



دیکدرها دارای انواع دو به چهار، سه به هشت، چهار به شانزده و ... هستند. کاربرد اصلی دیکدر در مدارهای دیجیتال، دسترسی به خانه‌های حافظه است. مدار دیکدر می‌تواند شامل یک سیگنال فعال ساز (En) باشد. اگر سیگنال فعال ساز وجود نداشته باشد، مدار دیکدر غیرفعال خواهد شد و عمل نخواهد کرد. در دیکدر مذکور اگر n ورودی داشته باشیم 2^n تا خروجی دریافت خواهد شد.



مرحله 1) پیاده‌سازی دیکدر ۲ به ۴ با استفاده از گیت‌های منطقی

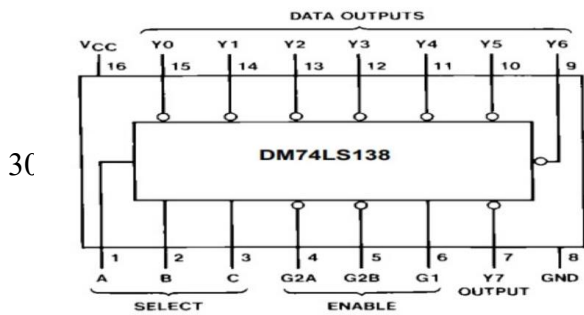
۱- با توجه به عملکردی که از یک دیکدر ۲ به ۴ انتظار دارید، ابتدا جدول صحت آن را تکمیل کنید و سپس توابع متناظر با خروجی را بدست آورید. در نهایت با استفاده از IC7400، دیکدر ۲ به ۴ را پیاده‌سازی کنید.

A	B	Y ₃	Y ₂	Y ₁	Y ₀
0	0				
0	1				
1	0				
1	1				

مرحله ۲) بررسی عملکرد دیکدر ۳ به ۸ (IC74138)

IC74138 دارای ۱۶ پایه است که سه ورودی برای کد و سه ورودی برای فعال‌سازی آن‌ها و هشت خروجی دارد. پین دیاگرام آن در شکل نشان داده شده است. توجه شود که در این آی‌سی، کم ارزش‌ترین و

C با ارزش‌ترین بیت است و همانطور که در نمای شماتیک پین دیاگرام



مشاهده می شود، تمامی خروجی ها not دارند، پس برای ارزیابی خروجی ها باید به LED های active low وصل کنیم.

۲- با استفاده از IC74238 جدول زیر را تکمیل کنید.

C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
۰	۰	۰								
۰	۰	۱								
۰	۱	۰								
۰	۱	۱								
۱	۰	۰								
۱	۰	۱								
۱	۱	۰								
۱	۱	۱								

مرحله ۳) پیاده سازی توابع مختلف با دیکدر IC74138

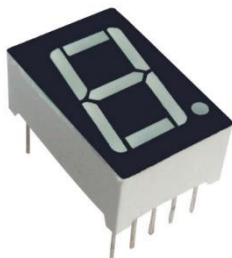
۳- برای اجرای توابع $F_1 = \Sigma(1,3)$, $F_2 = \Sigma(3,4,7)$ با استفاده از دیکدر ۳ به ۸ بگونه ای طراحی را انجام دهید که فقط با کمک گیت ۷۴۰۰ قابل اجرا باشد و بعد از پیاده سازی، صحت مدار بسته شده را چک کرده و جدول را تکمیل کنید.

C	B	A	F ₁	F ₂
۰	۰	۰		
۰	۰	۱		
۰	۱	۰		
۰	۱	۱		
۱	۰	۰		
۱	۰	۱		
۱	۱	۰		
۱	۱	۱		

۱۰. دیکدر BCD و نمایشگر ۷ قسمتی (Seven Segment)

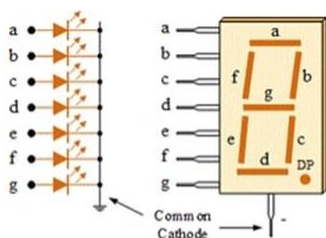
خلاصه آزمایش: در این جلسه با سون سگمنت و دیکدر BCD آشنا شده و به کمک آن ها اعداد را بر روی نمایشگر، نمایش خواهیم داد.

آشنایی با سون سگمنت (Seven Segment) : از هفت LED تشکیل شده است که این هفت LED در کنار هم قرار گرفته اند و یک آبی را می سازند. برای نمایش هریک از اعداد ۰ تا ۹ و کارکترهای هگزا دسیمال A تا F بر روی آن، لازم است تا ترکیب خاصی از این LED ها روشن شوند. از کاربردهای اصلی نمایش گر سون سگمنت می توان در دستگاه های الکترونیکی، ابزارهای اندازه گیری، تجهیزات ارتباطی، ساعت های دیجیتال، تابلوهای اعلانات و سایر سیستم های نمایش اطلاعات مشابه، اشاره کرد. یک Seven Segment استاندارد، ۸ پایه ورودی دارد که ۷ تای آن متعلق به هفت LED بوده و هشتمین پایه به عنوان پایه مشترک به کار می رود که مخصوص تمامی سگمنت های داخلی نمایشگر است. برخی از نمایشگرها یک پایه ی ورودی دیگر نیز دارند که به منظور نمایش نقطه ی اعشار در گوشه ی پایین سمت راست به کار می رود.



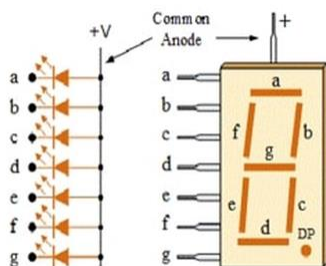
همان طور که می دانیم، هر LED دو پایه دارد که یکی از آن ها آند و دیگری کاتد نامیده می شود. بر همین اساس، دو دسته سون سگمنت خواهیم داشت: کاتد مشترک (Common Cathode) و آند مشترک (Common Anode).

۱- **سون سگمنت کاتد مشترک:** همه کاتدها به یکدیگر متصل شده و با هم به "0" یا زمین وصل می شوند. بنابراین، هر سگمنت را می توان با اعمال سیگنال High یا "1" منطقی به ترمینال آند، روشن کرد.



کاتد مشترک

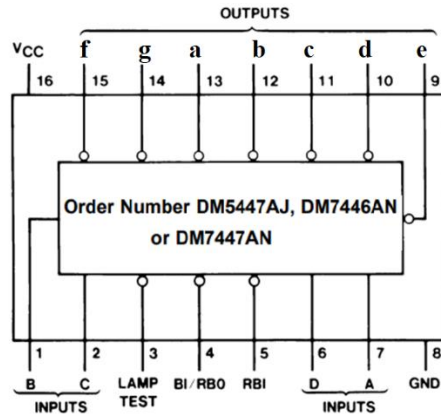
۲- **سون سگمنت آند مشترک:** همه آندها به یکدیگر متصل شده و با هم به ولتاژ High یا "1" منطقی وصل می شوند. بنابراین برای روشن شدن هر کدام از سگمنت ها، تنها لازم است که به پایه ی ورودی آن (کاتد)، ولتاژ Low یا "0" منطقی اعمال کنیم.



آند مشترک

برای مثال برای نمایش عدد ۳، باید سگمنت های a,b,g,c,d باید روشن شوند. یا برای نمایش حرف F، باید سگمنت های a,f,g,e باید روشن شوند.

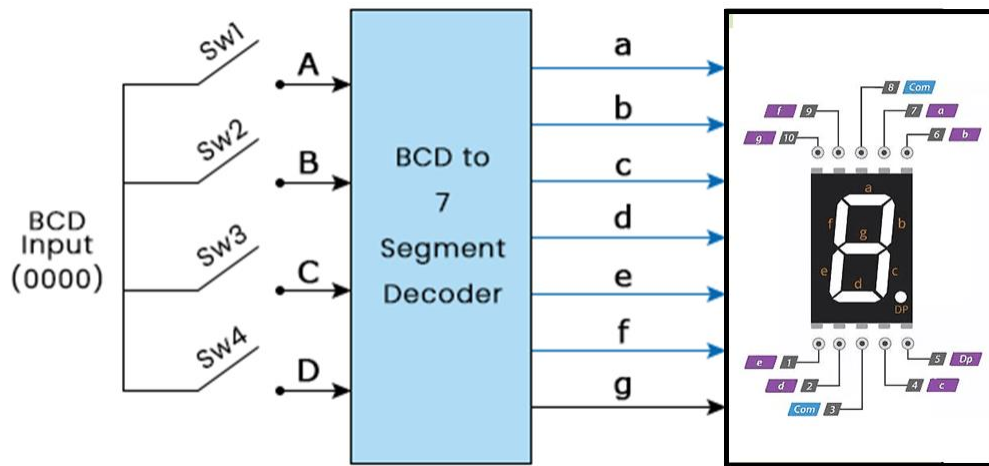
آشنایی با IC7447: یکی از انواع مختلف دیکدرها، دیکدر BCD به دهدهی است. ورودی آن کدها و خروجی آن اعداد در محدوده ۰ تا ۹ هستند. آی سی ۷۴۴۷ از سری آی سی های TTL شامل یک دیکدر BCD به سون سگمنت بوده که برای راه اندازی سون سگمنت آند مشترک به کار می رود. آی سی ۷۴۴۷ دارای ۴ ورودی BCD و ۸ خروجی مناسب سون سگمنت دارد. از ویژگی های این آی سی میتوان به پایه lamp test و پایه RBI, RBO اشاره کرد. پین دیاگرام آن در شکل نشان داده شده است. توجه شود که D با ارزش ترین بیت کد ورودی و A کم ارزش ترین بیت کد ورودی است.



مرحله 1) مبدل BCD به Seven Segment

با استفاده از IC7447 که یک دیکدر چهار به هشت است، به سون سگمنت ورودی می شود و خروجی آن را ثبت می گردد.

شرح آزمایش: برای اینکار به وسیله چهار کلید، کد متناسب را به ورودی دیکدر اعمال می شود و خروجی دهدهی متناسب با کد، از خروجی دیکدر، به ورودی سون سگمنت متصل می شود. نحوه ی اتصال به صورت نمای شماتیک نشان داده شده در شکل زیر است.



الف) ورودی ها را مطابق جدول زیر به IC7447 اعمال کنید، شکل خروجی نشان داده شده در خروجی را ثبت کنید.

D	C	B	A	Output
.	.	.	.	

۰	۰	۰	۱
۰	۰	۱	۰
۰	۰	۱	۱
۰	۱	۰	۰
۰	۱	۰	۱
۰	۱	۱	۰
۰	۱	۱	۱
۱	۰	۰	۰
۱	۰	۰	۱
۱	۰	۱	۰
۱	۰	۱	۱
۱	۱	۰	۰
۱	۱	۰	۱
۱	۱	۱	۰
۱	۱	۱	۱

۱-ب) نمایشگر تا چه عددی را درست نمایش داد؟ چرا؟

۱-ج) فرق IC 7447 و IC 7448 چیست؟ برای طراحی این آزمایش با IC 7448 به چه صورت باید عمل می‌شد و چه تغییری لازم بود؟

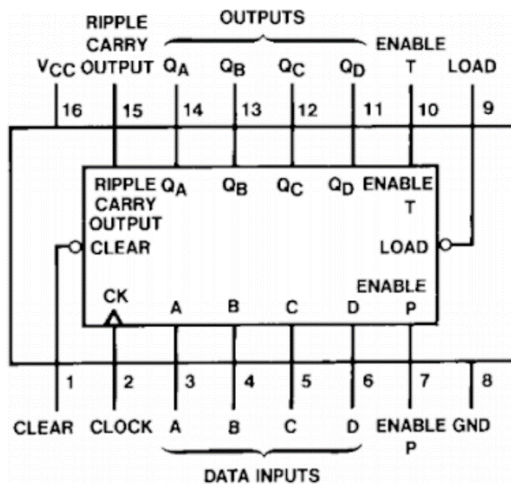
۱-د) وظیفه ی پایه‌های شماره ۳ و ۴ و ۵ آی سی 7447 چیست؟

مرحله 2) اتصال شمارنده به Seven Segment

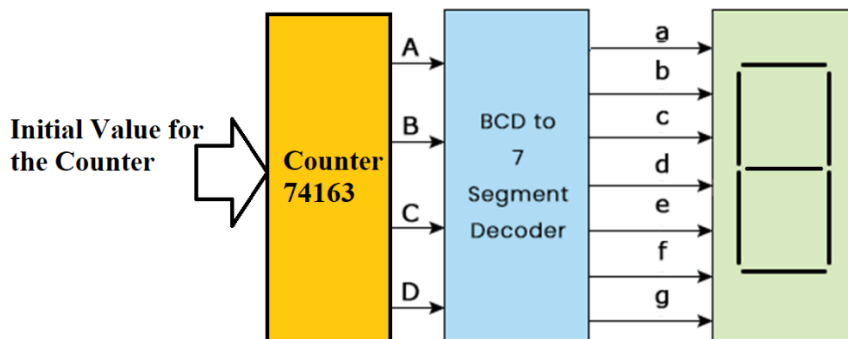
شمارنده‌ها گروهی از مدارات منطقی هستند که یک سری از پالس‌های ورودی را می‌شمارند.

در این مرحله تصمیم است تا با اتصال دیکدر ، سون سگمنت و شمارنده (74163) بتوان شمارش اعداد به صورت نزولی و صعودی و یا با شروع از اعداد مختلف را نمایش داد.

IC74163 در واقع یک شمارنده است که دارای ۱۶ پایه امی باشد، که چهار ورودی و چهار خروجی دارد. چهار خروجی را می‌توان به ورودی دیکدر BCD متصل کرد و سپس خروجی دیکدر را به Seven Segment متصل کرد تا بتوان شاهد شمارش اعداد دهدهی بود. **توجه شود** که D ورودی باارزش‌ترین بیت و A ورودی کم ارزش‌ترین بیت است.



نحوه‌ی اتصال آن به صورت بلوک دیاگرامی در شکل زیر نشان داده شده است.



- ۲-الف) بعد از بستن مدار، نحوه‌ی شمارشی را که بر روی seven segment می‌بینید، شرح دهید.
- ۲-ب) شرح دهید که چگونه می‌توان شمارنده را طراحی کرد تا از ۰ تا ۹ را به صورت بالاشمار بشمارد.
- ۲-ج) شرح دهید که چگونه می‌توان شمارنده را طراحی کرد تا از ۰ تا ۹ را به صورت پایین‌شمار بشمارد.
- ۲-د) شرح دهید که چگونه می‌توان شمارنده را طراحی کرد تا از ۳ تا ۹ بشمارد.
- ۲-ه) نقش پایه‌ی carry out را تحقیق کنید.

۱۱. لچ و فلیپ فلاپ

خلاصه آزمایش: در این جلسه با مفاهیم لچ و فلیپ فلاپ و مسیر رفع اشکال ساختارها آشنا می‌شویم. با استفاده از گیت‌های پایه لچ‌های SR, JK, D پیاده‌سازی می‌شود و سپس با کمک لچ‌ها، ساختار فلیپ‌فلاپ‌های SR, JK پیاده‌سازی خواهد شد.

مدارهای ترتیبی، اطلاعات دودویی را از ورودی‌ها دریافت می‌کنند و این ورودی‌ها به همراه حالت عناصر حافظه، یک مقدار دودویی را در پایانه‌های خروجی مشخص می‌نمایند. مدارهای ترتیبی از نظر مسائل زمانی سیگنال‌هایشان به دو نوع اساسی سنکرون و آسنکرون طبقه‌بندی می‌شوند. در سیستم سنکرون کلاک همگی عناصر حافظه همگام با هم هستند. در حالی که در آسنکرون کلاک‌ها غیرهمگام هستند.

عناصر پایه‌ی حافظه لچ‌ها هستند. که اگر ورودی enable باشد، داده‌ی ورودی روی خروجی می‌تواند تاثیر بگذارد، پس در کل بازه‌ی فعال بودن، هر تغییری در ورودی بر روی خروجی تغییر ایجاد می‌کند پس لچ‌ها حساس به سطح (سطح + یا سطح -) هستند. درحالی‌که در مدارات دیجیتال نیاز داریم تنها در بازه‌ی خاصی اطلاعات را ذخیره کنیم نه در یک بازه‌ی طولانی. علاوه بر این، در لچ‌ها مشکل همگام سازی هم داریم، یعنی به راحتی نمی‌توانیم همه حافظه‌ها را با هم سنکرون کنیم.

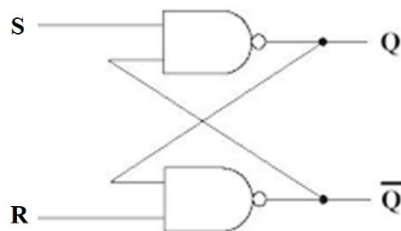
می‌توان از فلیپ‌فلاپ‌ها به جای لچ استفاده کرد که فقط در زمانهای خاصی، خروجی‌ها به‌روز می‌شوند. در واقع FF ها از لچ‌ها ساخته می‌شوند و حافظه‌های حساس به لبه (لبه پایین‌رونده یا لبه بالا‌رونده) هستند. یک FF قادر است تا وقتی که تحت تاثیر یک سیگنال ورودی برای تغییر حالت قرار نگرفته، یک حالت دودویی را به‌طور نامحدود در خود نگهداری کند. مدار یک FF دارای دو خروجی است (یکی برای مقدار طبیعی بیت ذخیره شده و دیگری برای مکمل آن).

- مرسوم‌ترین فلیپ‌فلاپ‌ها عبارتند از: D, T, JK, SR

FF- ها ورودی‌هایی برای clear یا reset دارند. این ورودی‌ها توسط یک سیگنال مثبت یا منفی روی FF اثر می‌کنند و برای این کار نیاز به پالس ساعت ندارند.

مرحله 1) طراحی و پیاده‌سازی SR-Latch

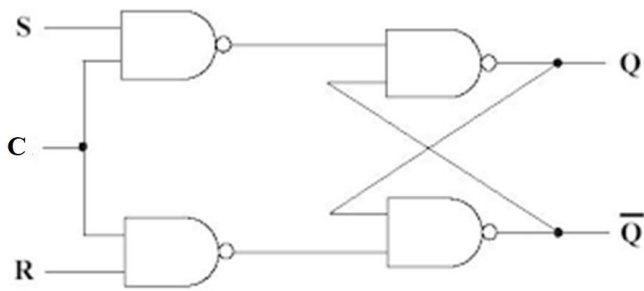
۱-الف) مدار SR-latch زیر را مطابق شکل زیر ببینید. با اعمال ورودی‌ها مطابق جدول، خروجی‌ها را ثبت کنید. مشکل مدار حاضر چیست؟ طرز کار مدار را شرح دهید.



اگر به طرح قبلی، یک ورودی کنترلی به صورت زیر اضافه چه اتفاقی می‌افتد؟ نقش ورودی C چیست؟

S	R	Q	\bar{Q}
0	1		
0	0		
1	0		
0	0		
1	1		
.	.		
۱	.		
۱	۱		
1	0		

۱-ب) کنیم



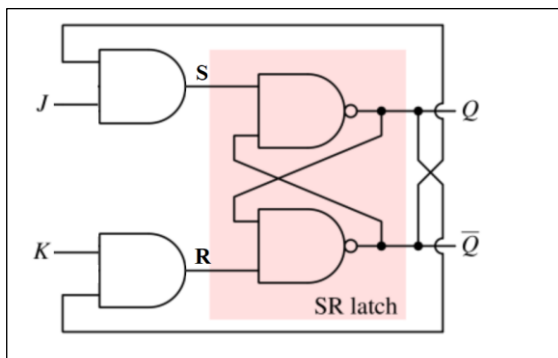
S	R	C	Q	\bar{Q}
0	1	۱		
0	0	۱		
1	0	۱		
0	0	۱		
1	1	۱		
.	.	۱		
0	1	۰		
0	0	۰		
1	0	۰		
۱	۰	۱		

مرحله 2) طراحی و پیاده‌سازی JK- D- Latch

۲-الف) در مدار قبلی (SR-latch کنترلی) ، ورودی S را با یک Not به ورودی R وصل کنید. بعد از بستن مدار، جدول را تکمیل کنید و عملکرد آن را شرح دهید. مزیت این ساختار نسبت به SR-latch کنترلی چیست؟

D	En	Q
1	0	
0	0	
0	۱	
1	۱	

۲-ب) مطابق مدار شکل زیر، یک JK-Latch را با SR-Latch ساده، پیاده‌سازی کنید.

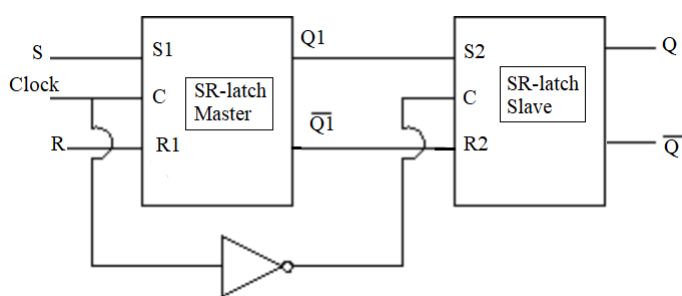


J	K	Q	\bar{Q}
0	1		
0	0		
1	0		
0	0		
1	1		
.	.		
0	1		
1	1		
1	0		
۱	1		

۲-ج) مشکل مدار فوق چیست؟

مرحله ۳) طراحی و پیاده‌سازی فلیپ فلاپ‌ها

۳-الف) با استفاده از SR-Latch با آرایش master-slave، فلیپ فلاپ SR را پیاده‌سازی کرده و جدول زیر را تکمیل کنید.

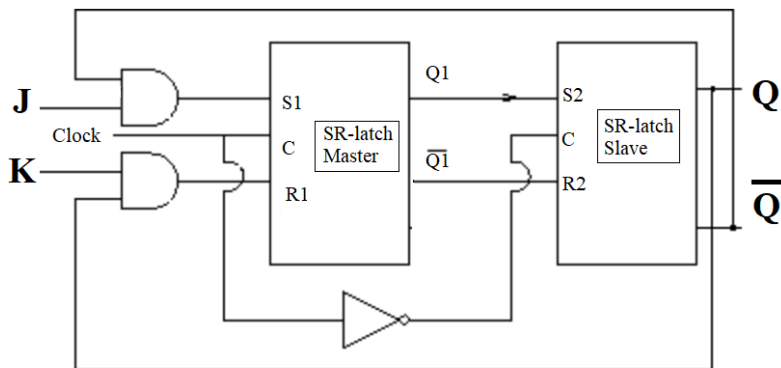


S	R	Q	\bar{Q}
0	1		
0	0		
1	0		
0	0		
1	1		
.	.		
0	1		

۳-ب) در شکل‌بندی فوق فلیپ فلاپ به کدام لبه حساس است؟

۳-ج) با استفاده از SR-Latch با آرایش master-slave، فلیپ فلاپ JK را پیاده‌سازی کرده و جدول زیر را تکمیل کنید. آیا مشکل موجود

در JK-latch همچنان وجود دارد؟

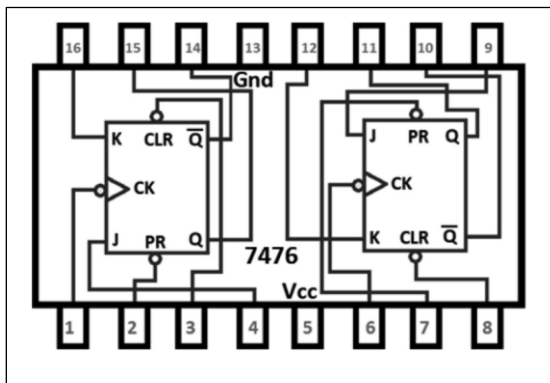


J	K	Q	\bar{Q}
0	1		
0	0		
1	0		
0	0		
1	1		
·	·		
0	1		
1	1		
1	0		
∨	1		

خلاصه آزمایش: با IC7476 و IC74194 آشنا شده و عملکرد این دو آی سی بررسی می شود.

مرحله (۱) ساخت T-FF با استفاده از JK-FF

فلیپ فلاپ JK یکی از پرکاربردترین و مهم ترین فلیپ فلاپ ها در طراحی مدارات منطقی است. این نوع فلیپ فلاپ دارای دو ورودی است که به طور سنتی با نمادهای J و K برچسب گذاری می شوند. IC7476 شامل دو JK-FF با پایه ی پاک کننده است و شکل پین دیاگرام آن در شکل زیر نشان داده شده است.



۱-الف) عملکرد فلیپ فلاپ JK را بررسی کنید و جدول را تکمیل کنید.

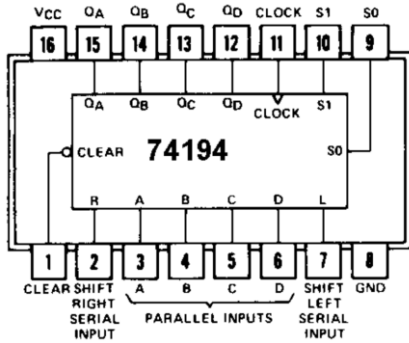
J	K	Q	\bar{Q}
0	1		
0	0		
1	0		
0	0		
1	1		
.	.		
0	1		
1	1		
1	0		
۱	1		

۱-ب) با استفاده از T-FF، IC7476، طراحی کنید. ابتدا فلیپ فلاپ را reset کنید و سپس جدول را پر کنید.

۱-ج) T-FF دارای چند حالت است؟

مرحله 2) بررسی عملکرد شیفت رجیستر

IC74194 یک شیفت رجیستر چهاربیتی است که دارای ۱۶ پایه است و در دو جهت قابلیت شیفت ایجاد می کند. پین دیاگرام آن در شکل زیر نشان داده شده است.



۲-الف) در مورد نقش پایه های 2 و ۷ تحقیق کنید. در آزمایشات بعدی این پایه ها را به یک وصل کنید.

۲-ب) اگر $S1S0 = 11$ باشد. چه عملکردی صورت می پذیرد؟ (برای بررسی عملکرد در ورودی ۰۱۰۰ اعمال شود و خروجی را ثبت کنید)

Input	Output
0100	

۲-ج) اگر $S1S0 = 01$ باشد. به ترتیب مقدار خروجی را که می بینید، ثبت کنید. چه عملکردی صورت می پذیرد؟

Input	Output
0100	

۲-د) اگر $S1S0 = 10$ باشد. به ترتیب مقدار خروجی را که می بینید، ثبت کنید. چه عملکردی صورت می پذیرد؟

Input	Output
0100	