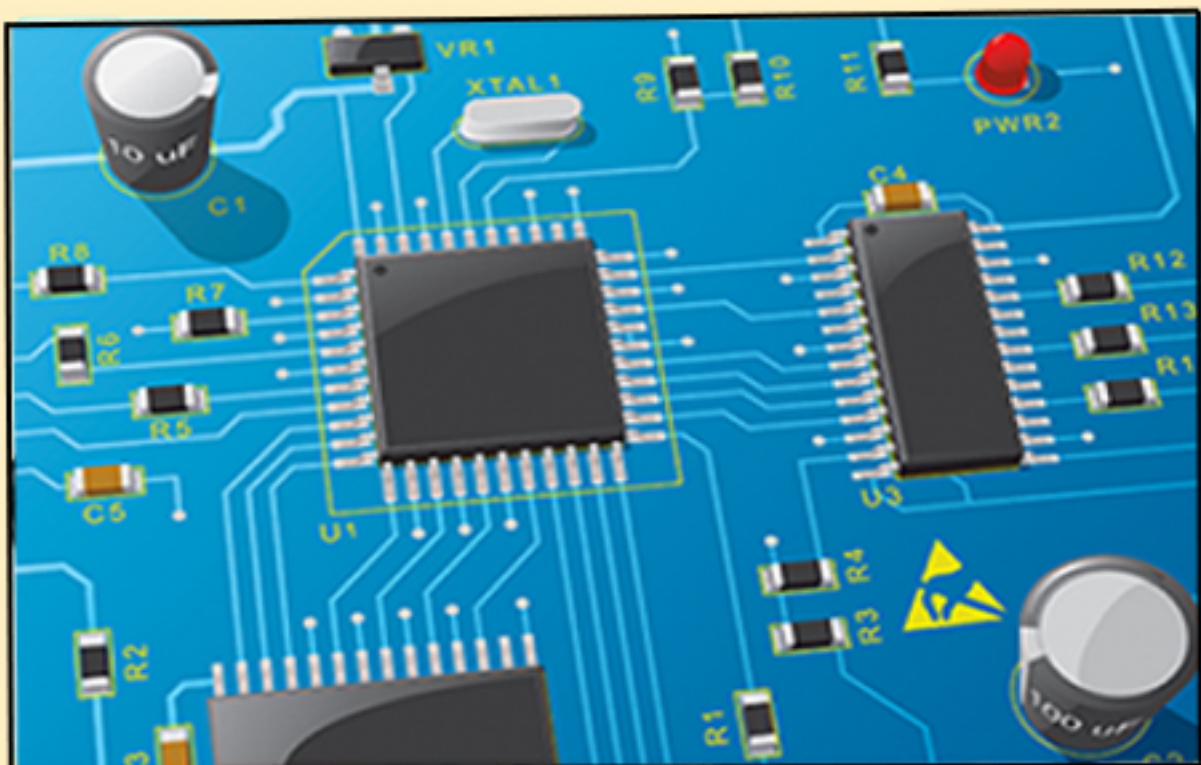




آزمایشگاه معماری کامپیوتر





آزمایشگاه معماری کامپیوتر

مولفین: رضا عسکری مقدم، مهدی عظیمی پور

پیشگفتار

کتاب درس آزمایشگاه معماری کامپیوتر به عنوان یک واحد درس عملی برای رشته‌های مهندسی برق و کامپیوتر با توجه به سرفصلهای این درس نوشته شده است و تمام آزمایشها به نحوی طراحی شده‌اند که بر روی بردهای آموزشی استاندارد موجود در مراکز آموزشی دانشگاه پیام‌نور قابل انجام می‌باشند. مطالب این کتاب در قالب چهارفصل، دسته‌بندی گردیده و شامل هشت آزمایش می‌باشند. فصل اول شامل تاریخچه و مباحث تئوری مربوط به مدارهای دیجیتال می‌باشد و برای یادآوری و آمادگی دانشجویان تهیه شده است. فصل دوم کتاب، نحوه انجام هشت آزمایش مختلف می‌باشد. انجام تمامی این آزمایش‌ها به صورت اجباری می‌باشد و استاد درس در طی هشت جلسه قادر به اتمام آنهاست. در صورتی که زمان کافی در اختیار استاد باشد، دانشجویان علاقمند می‌توانند آزمایشهای بیشتری را نیز انجام دهند. بعنوان مثال می‌توان با ترکیب اجزای ساخته شده در هشت آزمایش فصل دوم، یک پردازنده ساخت و عملکرد آنرا برای دستورات مختلف آزمایش کرد. پیشنهاد می‌شود این درس بصورت هشت جلسه سه ساعته در طول یک ترم ارائه شده و آزمایشهای فصل دوم حتماً انجام گردند. برای انجام آزمایشهای موجود در فصل دوم، می‌توان از تجهیزات آموزشی موجود در دانشگاه استفاده کرد. مطالب فصل سوم جنبه تکمیلی و اختیاری دارند و برای آموزش دانشجویان علاقمند به کار با نرم‌افزار پروتئوس اضافه شده‌اند. فصل چهارم نیز اطلاعاتی درباره بردهای آزمایشگاهی و آموزشی موجود در آزمایشگاه است. علاوه بر آن مشخصات آی‌سی‌های مورد استفاده در این کتاب در فصل چهارم آورده شده‌اند. در این مجموعه سعی شده است تا ضمن ارائه مسائل عملی هر آزمایش، جنبه‌های تئوری نیز تا حدودی مطرح شوند تا این کتاب بتواند به صورت مستقل مورد استفاده قرار گیرد. بنابراین توصیه می‌شود که دانشجویان قبل از انجام آزمایش‌ها مباحث تئوری فصل اول را مطالعه نمایند. در پایان لازم به ذکر است که علی‌رغم تلاش نویسندگان برای کامل و جامع بودن این کتاب، ممکن است نواقصی در محتوا یا نگارش آن موجود باشد. مولفان کتاب پذیرای نظرات ارزشمند اساتید و متخصصان در جهت بهبود کتاب در چاپهای آینده هستند.

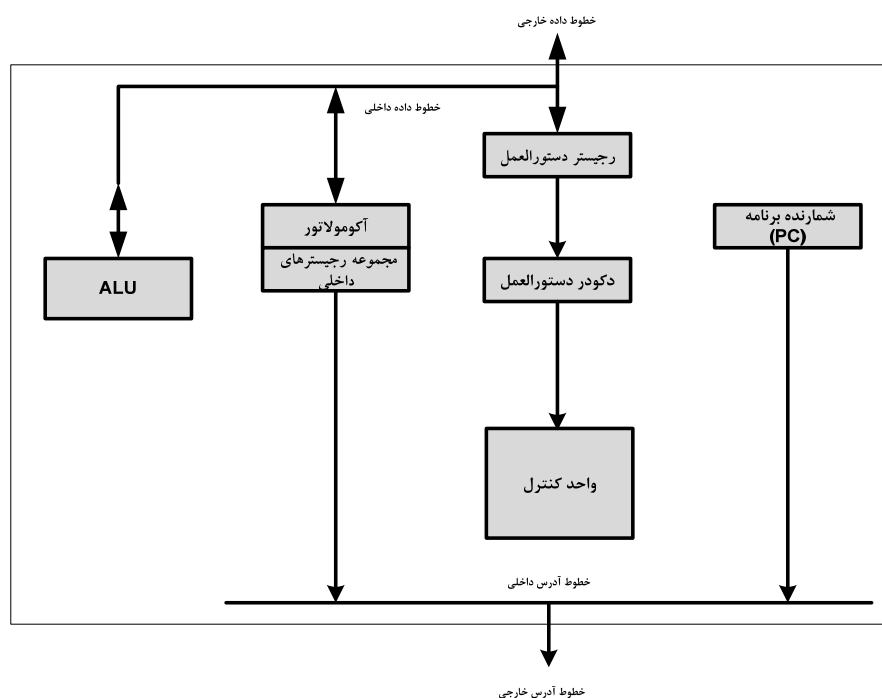
عسکری مقدم - عظیمی پور

فصل اول

معرفی و ساختار ریزپردازنده

۱-۱ اجزاء تشکیل دهنده یک میکروپروسسور

هدف از انجام آزمایشهای موجود در این کتاب تشریح عملکرد و پیاده سازی یک میکروپروسسور ۴ بیتی می باشد. در طرح ارائه شده سعی شده است حداقل سخت افزار لازم جهت آشنایی با عملکرد اجزای مختلف یک CPU وجود داشته باشد. در مدار طراحی شده امکان پیاده سازی انواع دستورالعمل وجود داشته و از طرفی از لحاظ سیم بندی دارای حجم متناسب می باشد. در طراحی مدار از تراشه های با سطح فشردگی متوسط MSI استفاده شده است. در شکل (۱-۱) بلوک دیاگرامی از اجزاء تشکیل دهنده یک میکروپروسسور نشان داده شده است. اگرچه میکروپروسسورها معمولاً دارای ساختمان داخلی بسیار پیچیده با اجزاء بسیار می باشند، ولی در مجموع می توان بلوکهای نشان داده شده در شکل (۱-۱) را به عنوان بلوکهای اصلی داخلی یک میکروپروسسور برشمرد و سایر اجزاء عمدتاً برای کمک در انجام وظایف این اجزاء اصلی در سیستم تعبیه میگردند.



شکل (۱-۱). ساختار داخلی میکروپروسسور

۱-۲ واحد محاسباتی-منطقی (ALU)

هر میکروپروسسور برای آنکه به خوبی از عهده انجام امور محوله برآید، لازم است که بتواند پاره ای اعمال محاسباتی و منطقی را به انجام برساند. بهمین دلیل، در اغلب میکروپروسسورها یک واحد ALU با قابلیت‌های لازم گنجانده می‌شود. در میکروپروسسورهای ساده اعمال محاسباتی قابل انجام توسط این واحد عبارت است از جمع و تفریق و برخی اعمال منطقی شامل انتقاب و چرخش داده‌ها، معکوس کردن، AND، OR و XOR.

۱-۳ ثبات‌های داخلی

یک ثبات، مجموعه‌ای از سلولهای حافظه است که می‌تواند اطلاعات دودویی را در خود نگهدارد. چون هر فلیپ فلاپ سلولی است که قادر به ذخیره یک بیت از اطلاعات باشد، مجموعه‌ای از فلیپ فلاپها یک ثبات را بوجود می‌آورند. ثباتها یا رجیسترهای داخلی به طور معمول در اغلب میکروپروسسورها وجود دارند. این ثباتها عموماً در داخل میکروپروسسور برای ثبت موقت اطلاعات بکار می‌روند. برخی از این ثباتها برای استفاده در اعمال داخلی میکروپروسسور اختصاص می‌یابند و برخی دیگر از ثباتها برای کاربرد عام می‌باشند و در دسترس استفاده کننده قرار دارند.

از میان این ثباتها، عموماً یک یا چند ثبات با امکانات ویژه طراحی شده و اصطلاحاً انبار یا آکومولاتور نامیده می‌شوند. در طراحی میکروپروسسورها، معمولاً آکومولاتورها دارای ارتباط مستقیم با ALU بوده و بیشتر اعمال محاسباتی و منطقی مستقیماً با آنها قابل انجام است.

۱-۴ شمارنده برنامه (PC)

از آنجا که میکروپروسسور یک سیستم ترتیبی است و لازم است که دستورالعملها یک به یک و به ترتیب از حافظه خوانده شده و سپس اجرا شوند، شمارنده ای به این منظور در میکروپروسسور تعبیه می‌شود که وظیفه اش اشاره به آدرسی از حافظه که دستورالعمل بعدی در آن قرار گرفته، می‌باشد. بنابراین در هنگام خواندن هر دستورالعمل از حافظه، محتوای شمارنده

برنامه است که بر روی خطوط آدرس قرار می‌گیرد و بلافاصله پس از خوانده شدن داده از طریق خطوط داده‌ها، مقدار فعلی شمارنده برنامه یکی افزایش می‌یابد تا به داده بعدی در حافظه اشاره نماید.

۱-۵ ثبات و دیکدر دستورالعمل

ثبات دستورالعمل، همچنانکه از نامش بر می‌آید، محل ذخیره سازی دستورالعمل در داخل میکروپروسسور می‌باشد. دستورالعمل پس از خوانده شدن از حافظه در داخل میکروپروسسور، در این ثبات قرار می‌گیرد تا امکان رمزگشایی آن فراهم شود. در دیکدر دستورالعمل که به خروجی ثبات دستورالعمل متصل می‌باشد، از دستورالعمل رمز برداری می‌شود و عملی را که باید به انجام برسد مشخص نموده و به واحد کنترل اعلام می‌نماید.

از آنجا که در طول مراحل انجام دستورالعمل، لازم است خروجی دیکدر مورد استفاده واقع شود، و با توجه به اینکه این دیکدر معمولاً یک مدار ترکیبی است، در واقع نگهداری اطلاعات در خروجی دیکدر توسط نگهداری اطلاعات در خروجی ثبات دستورالعمل امکان پذیر می‌گردد.

۱-۶ واحد کنترل

واحد کنترل به عنوان مغز و بخش تصمیم گیرنده در یک میکروپروسسور مطرح می‌گردد. وظیفه این واحد، کنترل تمامی اعمال داخلی و خارجی انجام شده توسط میکروپروسسور می‌باشد. این واحد ابتدا مقدار شمارنده برنامه را بر روی خطوط آدرس خارجی منتقل و سیگنالهای لازم برای امکان پذیر شدن خواندن حافظه را تامین می‌نماید. سپس در فاصله زمانی مناسبی، امکان انتقال داده‌های خوانده شده به رجیستر دستورالعمل را فراهم نموده و از خروجی دیکدر دستورالعمل، اطلاعات لازم در مورد دستورالعمل مربوطه را بدست می‌آورد. واحد کنترل بر اساس این اطلاعات، ترتیب انجام اعمال لازم جهت اجرای دستورالعمل مربوطه را تعیین و سیگنالهای لازم را به ترتیب فعال می‌نماید. در پایان اجرای هر دستورالعمل، واحد کنترل مجدداً شرایطی را فراهم می‌نماید تا میکروپروسسور آماده خواندن دستورالعمل بعدی از حافظه باشد.

۱-۷ خطوط داده‌های داخلی

خطوط داده داخلی در هر میکروپروسسور، محل انتقال داده‌های مورد مبادله بین اجزاء داخلی میکروپروسسور، یا بین دستگاه‌های خارجی با اجزاء داخلی آن می‌باشد. این خطوط محل عبور دستورالعملها و سایر داده‌ها بوده و معمولاً مستقیماً با اجزاء مهم و پر استفاده داخلی نظیر ALU، آکومولاتور و ثبات دستورالعمل ارتباط دارد. پهنای خطوط داده داخلی، در کنار ALU، یکی از عوامل تعیین کننده تعداد بیت‌ها در یک میکروپروسسور می‌باشد.

فصل دوم

آزمایشهای دیجیتال

آزمایش ۱- قسمت اول

موضوع آزمایش: طراحی یک حافظه ROM با استفاده از دیگدر

مقدمه

حافظه ROM یک نوع مدار مجتمع^۱ است که در زمان ساخت داده‌هایی در آن ذخیره می‌گردد. این نوع از حافظه‌ها علاوه بر استفاده در کامپیوترهای شخصی در سایر دستگاه‌های الکترونیکی نیز به خدمت گرفته می‌شوند. حافظه‌های ROM از لحاظ تکنولوژی استفاده شده، دارای انواع زیر می‌باشند:

- ROM^۲
- PROM^۳
- EPROM^۴
- EEPROM^۵
- Flash Memory

هر یک از مدل‌های فوق دارای ویژگی‌های منحصر بفرد خود می‌باشند. حافظه‌های فوق در موارد زیر دارای ویژگی مشابه می‌باشند:

- داده‌های ذخیره شده در این نوع تراشه‌ها "غیر فرار" بوده و پس از خاموش شدن منبع تامین انرژی، اطلاعات خود را از دست نمی‌دهند.
- داده‌های ذخیره شده در این نوع از حافظه‌ها غیر قابل تغییر بوده و یا اعمال تغییرات در آنها مستلزم انجام عملیات خاصی است.

^۱ Integrated Circuit

^۲ Read only Memory

^۳ Programmable Read only Memory

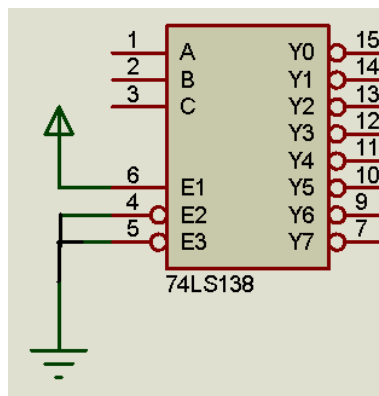
^۴ Erasable Programmable Read only Memory

^۵ Electrically Erasable Programmable Read only Memory

مبامث تئوری

یک دیکدر عبارتست از یک مدار ترکیبی که اطلاعات دودویی را از طریق n خط ورودی دریافت نموده و آنها را حداکثر به 2^n خط خروجی منحصر بفرد تبدیل می‌نماید. به عنوان مثال اگر در یک دیکدر ۳ به ۸ خطوط انتخاب بصورت $ABC=101$ باشد، خروجی D یک شده و دیگر خطوط صفر خواهند بود.

آی سی ۷۴۱۳۸ که در شکل (۱-۲) نشان داده شده است، می‌تواند بصورت یک دیکدر و یا دی مالتی پلکسر مورد استفاده قرار بگیرد. همانطور که در شکل نشان داده شده است، این آی سی دارای سه پایه فعال سازی $E1, E2, E3$ و $E1$ می‌باشد. برای اینکه این آی سی بصورت دیکدر عمل کند، باید پایه های $E1 = High$ و $E2 = E3 = Low$ باشد. بنابراین بر اساس مقادیر ورودی A, B, C ، یکی از خروجیها Low و بقیه در حالت $High$ باقی خواهند ماند. در مدار میز کار آزمایشگاه این کار بصورت داخلی انجام شده است و دیگری نیازی به انجام آن نمی‌باشد.



شکل (۱-۲): مدار دیکدر

روش انجام آزمایش

در اینجا می‌خواهیم با استفاده از این دیکدر یک حافظه ROM بصورت 8×3 طراحی کنیم که مقادیر ذخیره شده در خانه های حافظه آن بصورت جدول (۱-۲) باشد. همانطور که میدانید، یک دیکدر با استفاده از n متغییر ورودی، 2^n مینترم را تولید می‌کند. چون هر تابع بول می‌تواند بصورت مجموع مینترم ها بیان شود، لذا می‌توان با استفاده از یک دیکدر مینترم ها را تولید کرد و بوسیله یک گیت OR مجموع آنها را تشکیل داد. با استفاده از این روش هر مدار ترکیبی با n ورودی و m خروجی را می‌توان بوسیله یک دیکدر n به 2^n و m گیت OR پیاده سازی کرد. فقط باید توجه کنید که در صورت "NOT" بودن خروجیهای دیکدر، باید از گیت NAND بجای OR استفاده کنید (چرا؟).

جدول (۱-۲). مقادیر مطلوب در آدرسهای مختلف حافظه

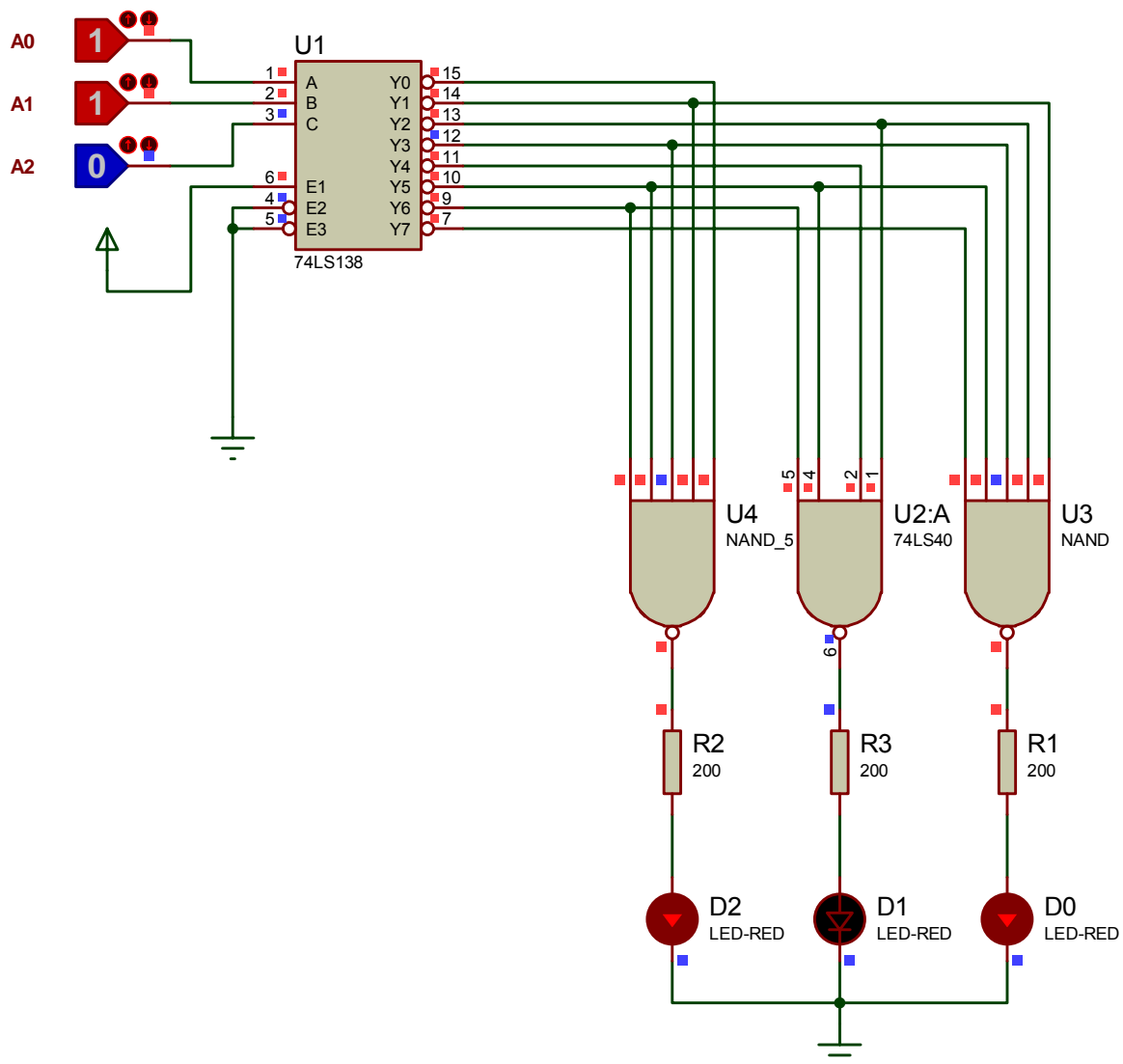
| آدرس حافظه ROM | | | مقدار ذخیره شده در حافظه | | |
|----------------|-------|-------|--------------------------|-------|-------|
| A_2 | A_1 | A_0 | D_2 | D_1 | D_0 |
| ۰ | ۰ | ۰ | ۱ | ۰ | ۰ |
| ۰ | ۰ | ۱ | ۱ | ۰ | ۱ |
| ۰ | ۱ | ۰ | ۰ | ۱ | ۱ |
| ۰ | ۱ | ۱ | ۱ | ۰ | ۱ |
| ۱ | ۰ | ۰ | ۰ | ۱ | ۰ |
| ۱ | ۰ | ۱ | ۱ | ۱ | ۱ |
| ۱ | ۱ | ۰ | ۱ | ۱ | ۰ |
| ۱ | ۱ | ۱ | ۰ | ۰ | ۱ |

در شکل (۲-۲) مدار خواسته شده برای پیاده سازی مقادیر نشان داده شده در جدول (۱-۲) نشان داده شده است. پس

از بستن مدار، مقادیر خواسته شده در جدول (۲-۲) را به مدار اعمال کرده و خروجی ROM را یاد داشت نمایید.

جدول (۲-۲). اندازه گیری مقادیر موجود در حافظه بر اساس آدرسهای داده شده

| آدرس حافظه ROM | | | مقدار ذخیره شده در حافظه | | |
|----------------|-------|-------|--------------------------|-------|-------|
| A_2 | A_1 | A_0 | D_2 | D_1 | D_0 |
| ۰ | ۰ | ۱ | ? | ? | ? |
| ۰ | ۱ | ۰ | ? | ? | ? |
| ۱ | ۰ | ۰ | ? | ? | ? |
| ۱ | ۰ | ۱ | ? | ? | ? |
| ۱ | ۱ | ۱ | ? | ? | ? |



شکل (۲-۲): مدار ROM بر طبق مقادیر جدول (۱-۲)

تمرین تئوری و شبیه سازی

با استفاده از یک دیکدر ۳ به ۸، یک حافظه ROM ۸ در ۳ طراحی کنید که مقادیر جدول (۲-۳) را در خانه های حافظه ذخیره کند. سپس مدار طراحی شده را در نرم افزار پروتئوس شبیه سازی کنید و صحت عملکرد مدار طراحی شده را بررسی کنید.

جدول (۳-۲). اطلاعات حافظه بر اساس آدرسها

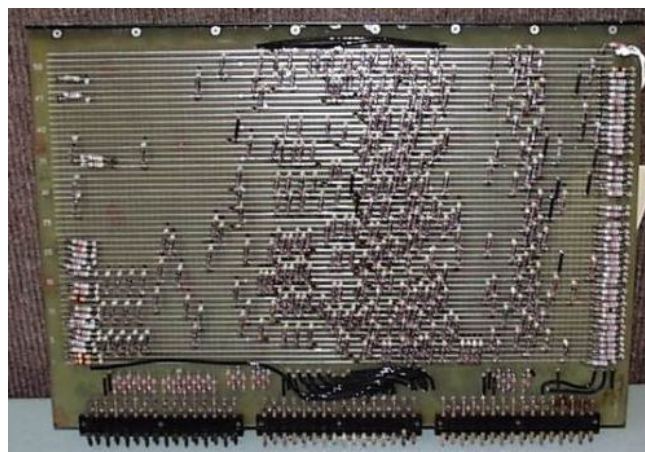
| آدرس حافظه ROM | | | مقدار ذخیره شده در حافظه | | |
|----------------|-------|-------|--------------------------|-------|-------|
| A_2 | A_1 | A_0 | D_2 | D_1 | D_0 |
| ۰ | ۰ | ۰ | ۱ | ۱ | ۰ |
| ۰ | ۰ | ۱ | ۰ | ۰ | ۱ |
| ۰ | ۱ | ۰ | ۰ | ۱ | ۰ |
| ۰ | ۱ | ۱ | ۱ | ۰ | ۱ |
| ۱ | ۰ | ۰ | ۱ | ۱ | ۰ |
| ۱ | ۰ | ۱ | ۰ | ۱ | ۱ |
| ۱ | ۱ | ۰ | ۱ | ۱ | ۰ |
| ۱ | ۱ | ۱ | ۱ | ۰ | ۱ |

آزمایش ۱- قسمت دوم

موضوع آزمایش: طراحی یک حافظه ROM با ساختار ماتریس دیودی

مقدمه

در سالهای ۱۹۶۰، در برخی از کامپیوترها و همچنین ماشین حسابهای الکترونیکی، این تکنولوژی برای ساخت حافظه های ROM مورد استفاده قرار میگرفته است. در این نوع ROM که بر روی برد و بصورت چاپی ساخته می شده است، برنامه ریزی حافظه به صورت دستی و تغییر دادن محل قرار گیری دیودها صورت میگرفت. در شکل (۲-۳) یک نمونه اولیه از این گونه حافظه ها نشان داده شده است.



شکل (۲-۳): حافظه ROM با ساختار ماتریس دیودی

مباحث تئوری

فرض کنید بخواهیم یک ROM ۴ در ۴ طراحی کنیم، به صورتی که مقادیر نشان داده شده در جدول (۲-۴) را در خانه های حافظه ذخیره نماید.

جدول (۲-۴): مقادیری که باید در حافظه ROM دیودی ذخیره شوند.

| آدرس حافظه ROM | | مقدار ذخیره شده در حافظه | | | |
|----------------|-------|--------------------------|-------|-------|-------|
| A_1 | A_0 | D_3 | D_2 | D_1 | D_0 |
| ۰ | ۰ | ۰ | ۱ | ۰ | ۰ |
| ۰ | ۱ | ۰ | ۰ | ۱ | ۰ |
| ۱ | ۰ | ۰ | ۱ | ۰ | ۱ |
| ۱ | ۱ | ۱ | ۰ | ۰ | ۰ |

بر اساس ورودی دیکدر، یکی از خروجیها (سطرها) ۰ و بقیه ۱ منطقی می‌باشند. در اینجا ستونها با یک مقاومت به منبع ۵ ولت متصل شده‌اند. در اینجا منطق هر ستون ۱ می‌باشد، مگر اینکه یکی از دیودهایی که به آن ستون متصل شده است روشن شود. یک دیود در صورتی روشن می‌شود که آند آن صفر شود، یعنی آن سطر توسط دیکدر انتخاب شود. در اینصورت با توجه به اینکه اختلاف ولتاژ بین آند و کاتد ۰/۶ ولت می‌باشد، بنابراین با روشن شدن دیود، ولتاژ آند ۰/۶ خواهد شد که درواقع صفر منطقی در نظر گرفته شده است.

به عنوان مثال فرض کنید ورودی $A_1 A_0 = 10$ به دیکدر داده شود. در اینصورت سطر سوم ۰ شده و بقیه سطرها ۱ خواهند بود. در این حالت، شما بایستی تنها به سطر سوم و دیودهای متصل شده به آن توجه کنید، زیرا سایر دیودها با توجه به اینکه آند و کاتد آنها ۱ می‌باشد، خاموش می‌باشند. در سطر سوم، دیودهای شماره ۳ و ۴ قرار دارند. بنابراین این دیودها روشن شده و ستونهای معادل آنها، یعنی ستونهای ۱ و ۳ منطق صفر بخود میگیرند. از آنجایی که خروجی دیکدر به صورت NOT می‌باشد، بنابراین در سر راه خروجی NOT قرار داده شده است. با قرار گرفتن NOT در خروجی، ستونهای ۱ و ۳ منطق یک بخود میگیرند و سایر ستونها صفر خواهند بود. بنابراین با توجه به توضیحات داده شده، خروجی بصورت زیر خواهد بود:

$$A_1 A_0 = 10 \rightarrow D_3 D_2 D_1 D_0 = 0101$$

جدول (۵-۲). اندازه‌گیری مقادیر موجود در حافظه بر اساس آدرسهای داده شده

| آدرس حافظه ROM | | مقدار ذخیره شده در حافظه | | | |
|----------------|-------|--------------------------|-------|-------|-------|
| A_1 | A_0 | D_3 | D_2 | D_1 | D_0 |
| ۰ | ۰ | ? | ? | ? | ? |
| ۰ | ۱ | ? | ? | ? | ? |
| ۱ | ۰ | ? | ? | ? | ? |
| ۱ | ۱ | ? | ? | ? | ? |

تمرین شبیه سازی

با استفاده از یک دیکدر ۳ به ۸ و دیود، یک حافظه ROM ۴ در ۴ طراحی کنید که مقادیر نشان داده شده در جدول (۶-۲) را در خانه های حافظه ذخیره کند. سپس مدار طراحی شده را در نرم افزار پروتئوس شبیه سازی کنید و نتایج تئوری را با نتایج شبیه سازی مقایسه کنید. نحوه شبیه‌سازی و کار با نرم افزار پروتئوس در پیوست کتاب موجود می‌باشد.

جدول (۶-۲). اطلاعات حافظه بر اساس آدرسها

| آدرس حافظه ROM | | مقدار ذخیره شده در حافظه | | | |
|----------------|-------|--------------------------|-------|-------|-------|
| A_1 | A_0 | D_3 | D_2 | D_1 | D_0 |
| ۰ | ۰ | ۰ | ۱ | ۰ | ۱ |
| ۰ | ۱ | ۱ | ۰ | ۱ | ۰ |
| ۱ | ۰ | ۰ | ۱ | ۰ | ۱ |
| ۱ | ۱ | ۱ | ۱ | ۰ | ۰ |

آزمایش ۲

موضوع آزمایش: آشنایی با نمونه نوشتن و خواندن از یک حافظه RAM

مقدمه

RAM یا حافظه دسترسی تصادفی از تعدادی خانه یا سلول تشکیل شده است و هر خانه، قابلیت نگهداری یک داده را دارد و با آدرسی منحصر به فرد مشخص می شود. آدرس اولین خانه حافظه، صفر است و آدرس هر خانه، یک واحد از خانه ی قبلی بیشتر است، هر آدرس حافظه، قابلیت نگهداری یک یا چند بایت را دارا است.

حافظه با قابلیت دسترسی تصادفی (در اصطلاح به آن RAM^۱ می گویند) یک نوع محل ذخیره اطلاعات در کامپیوتر می باشد از اطلاعات در هر زمانی بدون در نظر گرفتن موقعیت فیزیکی اطلاعات و ترتیب آنها وجود دارد امروزه شامل مدارهای مجتمعی می باشد که امکان ذخیره سازی اطلاعات را به صورت تصادفی به ما می دهد. واژه تصادفی به این معنی است که امکان دسترسی به هر مقدار اطلاعات در هر زمانی بدون در نظر گرفتن موقعیت فیزیکی اطلاعات و ترتیب آنها وجود دارد. داده های موجود در RAM قابل پاک شدن و جایگزینی با داده های دیگر هستند و هر نوع وقفه ای در جریان برق رایانه، موجب از بین رفتن داده های موجود در RAM می شود. استفاده از این نوع حافظه ها، برای نگهداری موقت اطلاعات تا زمان پردازش یا انتقال نتایج به بیرون از رایانه و یا ذخیره در حافظه های جانبی است.

از نظر تکنولوژی ساخت، دو نوع RAM وجود دارد :

۱. RAM پویا (DRAM)^۲

۲. RAM ایستا (SRAM)^۳

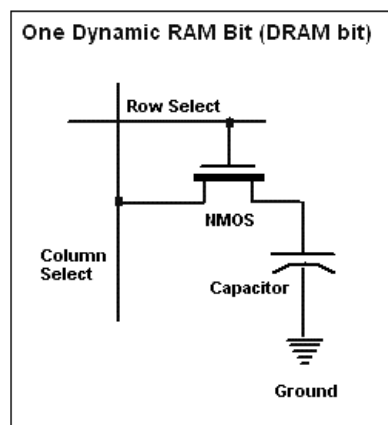
در حافظه DRAM، با بکارگیری یک خازن و یک ترانزیستور می توان یک سلول را ایجاد کرد. سلول فوق قادر به نگهداری یک بیت داده خواهد بود. خازن اطلاعات مربوط به بیت را که یک یا صفر است، در خود نگهداری خواهد

^۱ Random Access Memory

^۲ Dynamic RAM

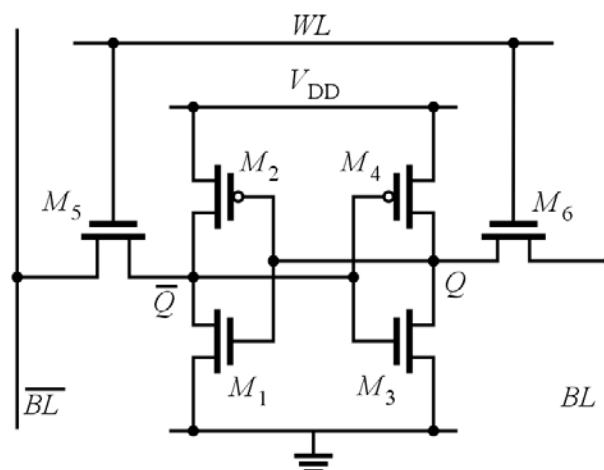
^۳ Static Ram

کرد. عملکرد ترانزیستور مشابه یک سویچ بوده که امکان کنترل مدارات موجود بر روی تراشه حافظه را به منظور خواندن مقدار ذخیره شده در خازن یا تغییر وضعیت مربوط به آن، فراهم می‌نماید. خازن مشابه یک ظرف بوده که قادر به نگهداری الکترون‌ها است. به منظور ذخیره سازی مقدار "یک" درحافظه، ظرف فوق می‌بایست از الکترون‌ها پرگردد. برای ذخیره سازی مقدار صفر، می‌بایست ظرف فوق خالی گردد. مسئله مهم در رابطه با خازن، نشت اطلاعات است، بدین ترتیب پس از گذشت چندین میلی ثانیه یک ظرف مملو از الکترون، تخلیه می‌گردد. بنابراین به منظور اینکه حافظه بصورت پویا اطلاعات خود را نگهداری نماید، می‌بایست پردازنده یا "کنترل کننده حافظه" قبل از تخلیه شدن خازن، مکلف به شارژ مجدد آن به منظور نگهداری مقدار "یک" باشند. بدین منظور کنترل کننده حافظه اطلاعات حافظه را خوانده و مجدداً اطلاعات را بازنویسی می‌نماید. عملیات بازنویسی اطلاعات، هزاران مرتبه در یک ثانیه تکرار خواهد شد. علت نامگذاری DRAM بدین دلیل است که این نوع حافظه‌ها مجبور به بازخوانی اطلاعات بصورت پویا خواهند بود. فرآیند تکراری "بازخوانی-بازنویسی اطلاعات" در این نوع حافظه‌ها باعث می‌شود که زمان تلف و سرعت حافظه کند گردد. سلول‌های حافظه بر روی یک تراشه سیلیکون و بصورت آرایه‌ای مشتمل از ستون‌ها (خطوط بیت) و سطرها (خطوط کلمات) تشکیل می‌گردند. نقطه تلاقی یک سطر و ستون بیانگر آدرس سلول حافظه است.



شکل (۲-۵): یک سلول حافظه دینامیکی

حافظه های SRAM دارای یک تکنولوژی کاملاً متفاوت می باشند. در این نوع از حافظه ها از فلیپ فلاپ برای ذخیره سازی هر بیت حافظه استفاده می گردد. یک فلیپ فلاپ برای یک سلول حافظه، از چهار تا شش ترانزیستور استفاده می کند . حافظه های SRAM نیازمند بازخوانی / بازنویسی اطلاعات نخواهند بود، بنابراین سرعت این نوع از حافظه ها به مراتب از حافظه های DRAM بیشتر است. با توجه به اینکه حافظه های SRAM از بخش های متعددی تشکیل می گردد، فضای استفاده شده آنها بر روی یک تراشه نیز بیشتر از یک سلول حافظه از نوع DRAM خواهد بود. در چنین مواردی میزان حافظه بر روی یک تراشه کاهش پیدا کرده و همین امر می تواند باعث افزایش قیمت این نوع از حافظه ها گردد. بنابراین حافظه های SRAM سریع و گران و حافظه های DRAM ارزان و کند می باشند . با توجه به موضوع فوق ، از حافظه های SRAM به منظور افزایش سرعت پردازنده (استفاده در حافظه Cache) و از حافظه های DRAM برای فضای حافظه RAM در کامپیوتر استفاده می گردد.



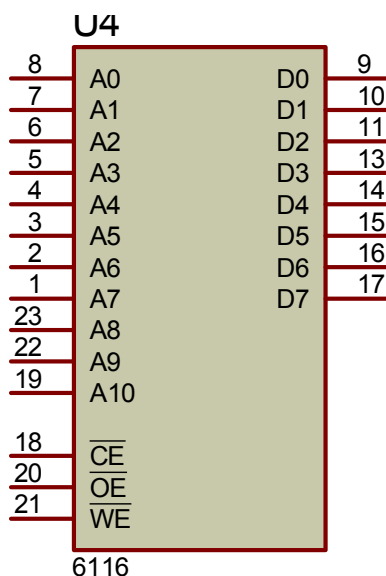
شکل (۲-۶): یک سلول حافظه استاتیکی

مباحث تئوری

آی سی ۶۱۱۶ نوعی RAM استاتیک می باشد که قابلیت گنجایش ۲ کیلو بایت اطلاعات را دارد. برای همین، ۱۱ خط

آدرس دارد (A۰ تا A۱۰). همان طور که در شکل (۷-۲) می بینید، این آی سی یک پایه فعال ساز \overline{CE} و دو پایه \overline{OE} و \overline{WE}

دارد که به ترتیب برای عملیات نوشتن (Write) و خواندن (Read) مورد استفاده قرار می گیرد.



شکل (۷-۲): حافظه استاتیکی ۶۱۱۶

ما در آزمایش این بخش، تنها از پایه های آدرس A۰ و A۱ و پایه های داده D۰ تا D۳ استفاده خواهیم کرد. به عبارت

دیگر، اطلاعات را در یک RAM ۴ در ۴ نوشته و آن را میخوانیم. به همین دلیل، همانطور که در شکل (۸-۲) نشان داده شده

است، پایه های آدرس A۲ تا A۱۰ همگی به زمین متصل شده اند. مودهای عملکرد این آی سی در جدول (۷-۲) نشان داده شده

است. برای عملکرد آی سی، پایه \overline{CE} بایستی همواره زمین باشد. توجه کنید که در اینجا ما تنها از دو حالتی که در جدول (۷-۲)

(۷) سایه دار شده اند استفاده خواهیم کرد.

جدول (۷-۲). مودهای عملکرد آی سی ۶۱۱۶

| | | | MODE |
|---|---|---|--------------|
| H | × | × | Not Selected |
| L | L | H | Read |
| L | H | L | Write |
| L | L | L | Write |

نمونه انجام آزمایش

نوشتن در حافظه RAM

همانطور که گفته شد، می‌خواهیم از آی سی ۶۱۱۶ به عنوان یک حافظه ۴ در ۴ استفاده کنیم. فرض کنید بخواهیم داده‌های

نشان داده شده در جدول (۸-۲) را در حافظه ذخیره کرده و سپس آنها را از حافظه بخوانیم.

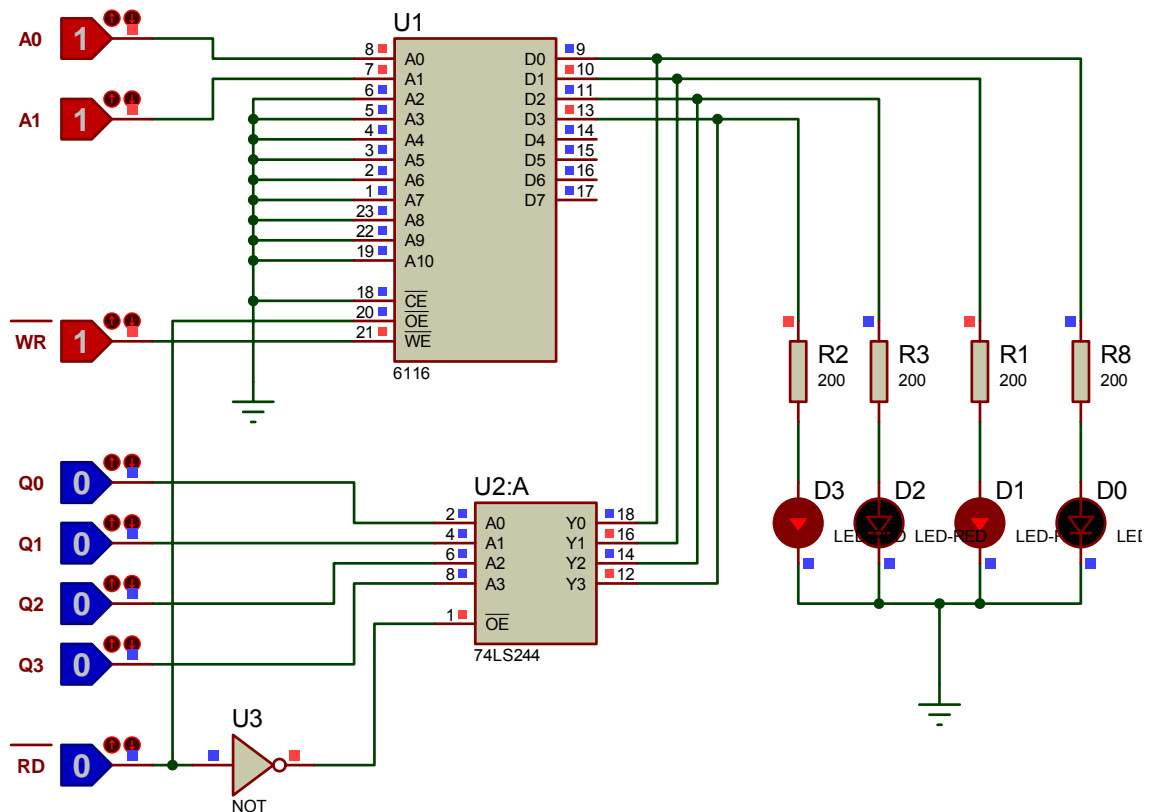
جدول (۸-۲). داده‌هایی که می‌خواهیم در حافظه ذخیره شوند.

| خانه حافظه | داده ذخیره شده |
|------------|----------------|
| ۰ | ۱۰۰۱ |
| ۱ | ۰۱۱۱ |
| ۲ | ۰۱۰۱ |
| ۳ | ۱۱۰۱ |

بدین منظور ابتدا مدار شکل (۸-۲) را بر روی بردبرد بسته و داده‌های نشان داده در جدول (۹-۲) را به ترتیب از سمت چپ به راست به مدار اعمال کنید. این سری عملیات به منظور ذخیره کردن داده‌های نشان داده شده در جدول (۸-۲) در حافظه **RAM** می‌باشد. همانطور که در جدول نیز نشان داده شده است، برای ذخیره کردن داده‌ها، ابتدا خانه حافظه مورد نظر را آدرس دهی می‌کنیم. سپس داده را بر روی خطوط داده **Q** قرار داده و پایه **\overline{WR}** را یک بار '۰' کرده و سپس آنرا به حالت '۱' منطقی می‌بریم. در طول فرآیند ذخیره داده‌ها، پایه **\overline{RD}** همواره غیر فعال ('۱' منطقی) باقی می‌ماند.

جدول (۹-۲). نحوه اعمال داده‌های ورودی به حافظه

| شماره خانه حافظه | A_1 | A_0 | Q_3 | Q_2 | Q_1 | Q_0 | | | |
|------------------|-------|-------|-------|-------|-------|-------|---|---|---|
| ۰ | ۰ | ۰ | ۱ | ۰ | ۰ | ۱ | ۱ | ۰ | ۱ |
| ۱ | ۰ | ۱ | ۰ | ۱ | ۱ | ۱ | ۱ | ۰ | ۱ |
| ۲ | ۱ | ۰ | ۰ | ۱ | ۰ | ۱ | ۱ | ۰ | ۱ |
| ۳ | ۱ | ۱ | ۱ | ۱ | ۰ | ۱ | ۱ | ۰ | ۱ |



شکل (۸-۲). مدار یک حافظه RAM با استفاده از آی سی ۶۱۱۶

خواندن از حافظه RAM

بعد از ذخیره کردن داده‌ها بر روی RAM، نوبت به خواندن داده‌ها از آن می‌رسد. بدین منظور، پایه \overline{WR} را غیر فعال کرده (آن را '۱' کنید) و پایه \overline{RD} را فعال کنید (آن را '۰' کنید). آگه به شکل (۸-۲) دقت کنید مشاهده می‌کنید که با غیر فعال کردن پایه \overline{WR} ، آی سی ۷۴LS۲۴۴ که همان مدار بافر می‌باشد، غیر فعال شده و خروجی آن در حالت امپدانس بالا قرار می‌گیرد. به عبارت دیگر، ورودی‌های Q دیگر تاثیری نخواهند داشت. حال مقادیر داده شده در جدول (۱۰-۲) را به خطوط آدرس داده و داده‌های ذخیره شده در آنها را که بر روی LED ها نمایش داده می‌شود را در جدول ثبت نمایید. آیا داده‌های خوانده شده همان داده‌های ذخیره شده می‌باشند؟

جدول (۱۰-۲). اندازه‌گیری مقادیر موجود در حافظه بر اساس آدرسهای داده شده

| A_1 | A_2 | D_3 | D_2 | D_1 | D_0 |
|-------|-------|-------|-------|-------|-------|
| ۰ | ۱ | ? | ? | ? | ? |
| ۱ | ۱ | ? | ? | ? | ? |
| ۱ | ۰ | ? | ? | ? | ? |
| ۰ | ۰ | ? | ? | ? | ? |

آزمایش ۳

موضوع آزمایش: بررسی یک درگاه ورودی موازی ساده

مبامث تئوری

در سیستمهای میکروپروسسوری، از مدارهای مختلفی به عنوان درگاه موازی ورودی/خروجی (I/O) می‌توان استفاده کرد:

✓ بافرهای سه حالت

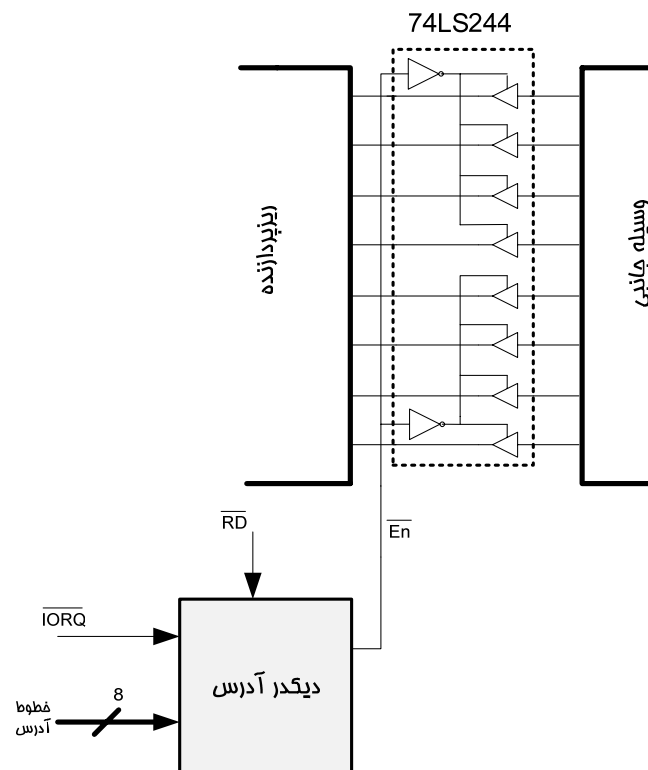
✓ رجیسترها

✓ مدارهای ورودی/خروجی قابل برنامه ریزی (همانند آی سی ۸۲۵۵)

در این آزمایش می‌خواهیم از بافرهای سه حالت برای طراحی درگاه موازی ورودی/خروجی استفاده کنیم. بنابراین ابتدا مقدمه‌ای در این مورد ارائه می‌شود.

بافرهای سه حالت از جمله مدارهایی هستند که به عنوان درگاه ورودی/خروجی در سیستمهای میکروپروسسوری مورد استفاده قرار می‌گیرند. این مدارها عموماً به صورت گیت‌های ۸ گانه در یک IC با پایه یا پایه های فعالساز^۹ مشترک قرار دارند (بعنوان یک نمونه از بافرهای سه حالت می‌توان ۷۴LS۲۷۷ را نام برد). در این صورت، اتصال این گیتها به خطوط داده‌های میکروپروسسور، مطابق شکل ۲-۹، موجب می‌گردد که در موقع فعال شدن پایه های فعالساز آنها، داده‌های موجود در پشت این بافرها بر روی خطوط داده‌ها قرار بگیرد. به این ترتیب با قرار دادن این مجموعه در طرح ورودی/خروجی سیستم و با استفاده از یکی از خروجی‌های مدار دیکدر I/O برای فعال کردن پایه فعالساز این مجموعه، میکروپروسسور می‌تواند در موقع لزوم با اجرای یک سیکل خواندن از درگاه، اطلاعات موجود در پشت این بافرها را بخواند.

^۹ Enable



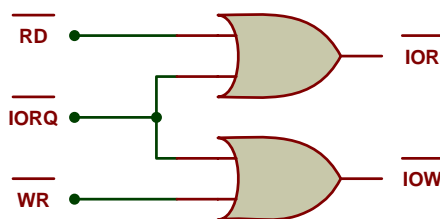
شکل (۹-۲) - نحوه اتصال بافر سه حالت به ریزپردازنده و دیکدر آدرس

سیگنال‌های کنترلی و دستورالعمل‌های اصلی I/O

ریزپردازنده Z80 از دو دستورالعمل **IN** و **OUT** برای انتقال داده از طریق درگاه **I/O** استفاده می‌کند. دستورالعمل **IN**

اطلاعات را از یک وسیله **I/O** خارجی که آدرس ۸ بیتی آن بر روی بیت‌های **A7-A0** از گذرگاه آدرس هستند به داخل انبار وارد

می‌کند. در شکل (۱۰-۲)، نحوه تولید سیگنال‌های کنترلی نشان داده شده است.



شکل (۱۰-۲) نحوه تولید سیگنال‌های کنترلی خواندن و نوشتن I/O

همانطور که گفته شد، در شکل (۲-۹) از آی سی ۷۴LS۲۴۴ که یک بافر سه وضعیتی می باشد، برای انتقال اطلاعات ورودی به گذرگاه اطلاعات استفاده می شود. در Z_{80} برای انتقال اطلاعات بین ریزپردازنده و بافر سه وضعیتی موجود در I/O از دستورات $OUT A,()$ و $IN A,()$ استفاده می گردد. با این مدار هر داده TTL را می توان به گذرگاه اطلاعات داخلی وارد کرد. از آنجایی که ممکن است چندین وسیله جانبی به خطوط داده میکروپروسور متصل شده باشند، بنابراین خروجی بایستی دارای حالت امپدانس بالا نیز باشد.

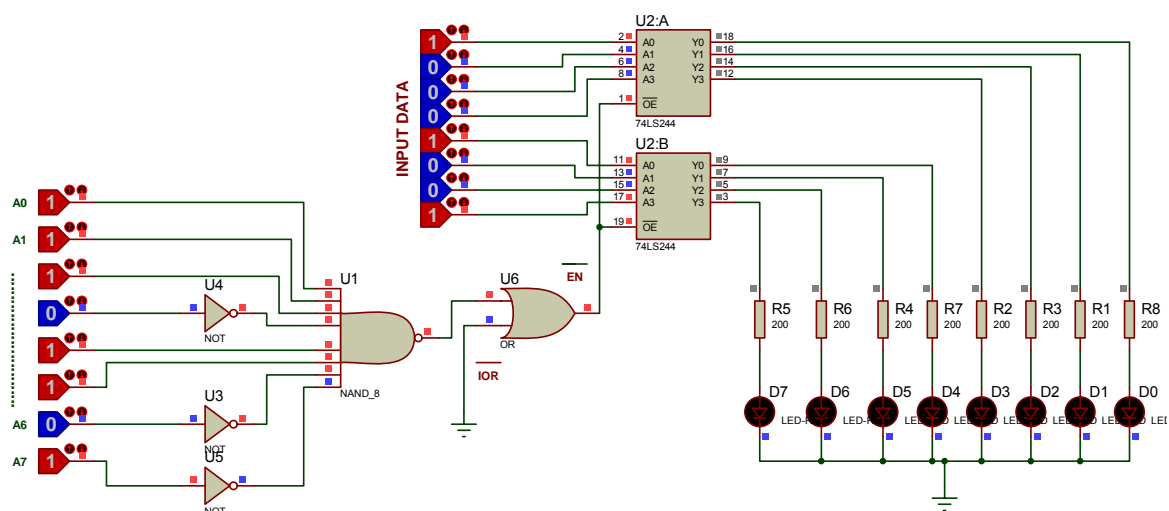
یکی از اهداف مهم این آزمایش، نحوه طراحی دیکدر آدرس می باشد. وظیفه دیکدر آدرس آن است که با توجه به دستورالعمل ریزپردازنده، از چندین وسیله جانبی که ممکن است به آن متصل شده باشند، تنها یکی از آنها را فعال و بقیه را غیرفعال کند تا بتواند داده های خود را به ریزپردازنده انتقال دهد.

نمونه انجام آزمایش

فرض کنید ریزپردازنده دستورالعمل $[IN A, (37H)]$ را اجرا کند. با این دستورالعمل، بایستی وسیله جانبی که آدرس آن $37H$ می باشد، فعال شده و بقیه غیر فعال باقی بمانند. در مدار شکل (۲-۱۱)، دیکدر آدرس برای انجام این دستورالعمل طراحی شده است. نحوه ی عملکرد دیکدر آدرس را تشریح کنید. سپس مدار را بر روی بردبورد بسته و یک ورودی دلخواه به درگاه ورودی مدار اعمال کنید. سپس مقادیر داده شده در جدول (۲-۱۱) را به گذرگاه آدرس اعمال کرده و جدول را تکمیل کنید. برای مشاهده خروجی می توانید از LED استفاده کنید. توجه کنید که آی سی ۷۴LS۲۴۴ در نرم افزار شبیه سازی شامل ۴ پایه ورودی می باشد، بنابراین بایستی از دو آی سی استفاده کرد تا بتوان ۸ بیت داده ریزپردازنده Z_{80} را انتقال داد.

جدول (۱۱-۲) مقادیر آدرس جهت انجام آزمایش سوم

| $A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$ | $D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$ |
|-----------------------------------|-----------------------------------|
| ۱۰۱۰ | ? |
| ... ۱۱ ۱۱۱۱ | ? |
| ... ۱۱ ۰۱۱۱ | ? |



شکل (۱۱-۲) مدار مربوط به آزمایش سوم

تمرین تئوری و شبیه سازی

در این قسمت، مدار دیکدر آدرس را طوری طراحی کنید که اطلاعات را به ازای دستورالعمل $[IN A, (5EH)]$ بافر کند.

سپس مدار طراحی شده را در نرم افزار پروتئوس شبیه سازی کنید و درستی عملکرد آن را تحقیق کنید.

آزمایش ۴

موضوع آزمایش: بررسی یک درگاه سریال

مقدمه

انتقال بیت به بیت اطلاعات به طور پشت سر هم از طریق یک کانال یا یک سیم را ارتباط سریال می‌گویند. در این روش اطلاعات که به صورت ۰ و ۱ درآمده‌اند، از طریق یک مسیر عبور، منتقل می‌شوند، یعنی در هر لحظه و یا سیکل زمانی، یک بیت داده از طریق سیم می‌تواند منتقل شود که ۰ و ۱ می‌باشد. دو روش برای انتقال داده‌های سریال وجود دارد:

✓ انتقال سنکرون:

در انتقال سنکرون، داده‌ها در بلوک‌هایی فرستاده می‌شود و فرستنده و گیرنده به وسیله کاراکترهای مخصوصی سنکرون می‌شوند که به آنها کاراکترهای سنکرون^{۱۰} می‌گویند.

✓ انتقال آسنکرون:

برای انتقال آسنکرون، یک بیت شروع انتقال را مشخص می‌کند و یک بیت پایان انتقال را اعلام می‌کند. اطلاعات در فاصله زمانی بین بیت شروع و بیت پایان ارسال می‌شوند. این در حالی است که در انتقال موازی اطلاعات، از چند مسیر همزمان استفاده می‌شود. برای مثال می‌توان از هشت رشته سیم برای انتقال هر هشت بیت از یک بایت اطلاعات، استفاده نمود و در هر سیکل زمانی هشت بیت داده را از هشت رشته سیم به صورت موازی انتقال داد. بنابراین، سرعت انتقال اطلاعات با این روش به علت موازی بودن مسیرها، افزایش پیدا خواهد کرد، اما از طرفی تعداد مسیرهای انتقال را بیشتر می‌نماید.

واقعیت استفاده از یک خط داده در تبادل سریال به جای خط داده ۸ بیت در تبادل موازی، نه تنها موجب ارزانی فرایند می‌شود بلکه موجب تبادل اطلاعات بین دو کامپیوتر واقع در دو شهر از طریق خط تلفن می‌گردد. برای راه اندازی ارتباط سریال،

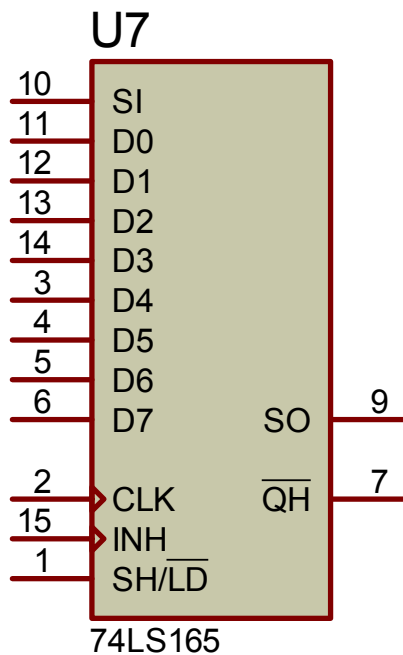
^{۱۰} Sync Character

داده‌ها باید از گذرگاه ۸ بیتی ریزپردازنده گرفته شده و با استفاده از شیفت رجیستر ورودی-موازی، خروجی-سریال، به بیت‌های سریال تبدیل گردد که آنگاه قابل ارسال به یک خط داده خواهد بود. واضح است که در سمت گیرنده، باید یک شیفت رجیستر ورودی-سریال، خروجی-موازی برای دریافت داده ارسالی وجود داشته باشد و پس از بسته بندی کردن بیت‌ها بصورت بایت، آنها را به سخت افزار موجود در گیرنده تحویل دهد. در این بخش می‌خواهیم با طرز کار یک فرستنده ورودی-موازی، خروجی سریال آشنا شویم.

مباحث تئوری

همانطور که گفته شد، یک درگاه سریال وظیفه تبدیل اطلاعات موازی به سری و جا دادن اطلاعات سری در یک قالب مشخص را برعهده دارد. یک نمونه از آی سی هایی که داده‌ها را به صورت موازی دریافت کرده و می‌تواند آنها را بصورت سریال ارسال کند، ۷۴LS۱۶۵ می‌باشد که در شکل (۲-۱۲) نشان داده شده است. این آی سی، داده‌ها را بصورت هشت بیتی و از طریق پایه‌های D_0, D_1, \dots, D_7 دریافت کرده و از طریق پایه SO^{11} ارسال می‌کند. توجه کنید که با توجه به جهت شیفت دادن بیت‌ها، ابتدا بیت D_7 از پایه SO خارج می‌شود. برای عملکرد نرمال مدار، پایه ۱۵ بایستی زمین شود. همانطور که در جدول (۲-۱۲) نشان داده شده است، برای بارگذاری بصورت موازی، باید داده‌ها را به پایه‌های D_0, D_1, \dots, D_7 اعمال کرده و سپس پایه $\overline{Shift/Load}$ را صفر کنیم. در ادامه و برای انتقال سریال داده‌ها، پایه $\overline{Shift/Load}$ را '۱' کرده و با هر لبه بالا رونده کلاک، بیت‌ها به صورت سریال از آی سی خارج می‌شوند.

^{۱۱} Serial Output



شکل (۱۲-۲): شمای کلی آی سی ۷۴LS۱۶۵ با ورودی موازی و خروجی سریال

جدول (۱۲-۲). نحوه عملکرد آی سی ۷۴LS۱۶۵

| Inputs | | Function |
|--------|-------|---------------|
| | Clock | |
| L | × | Parallel Load |
| H | | Shift |

نمونه انجام آزمایش

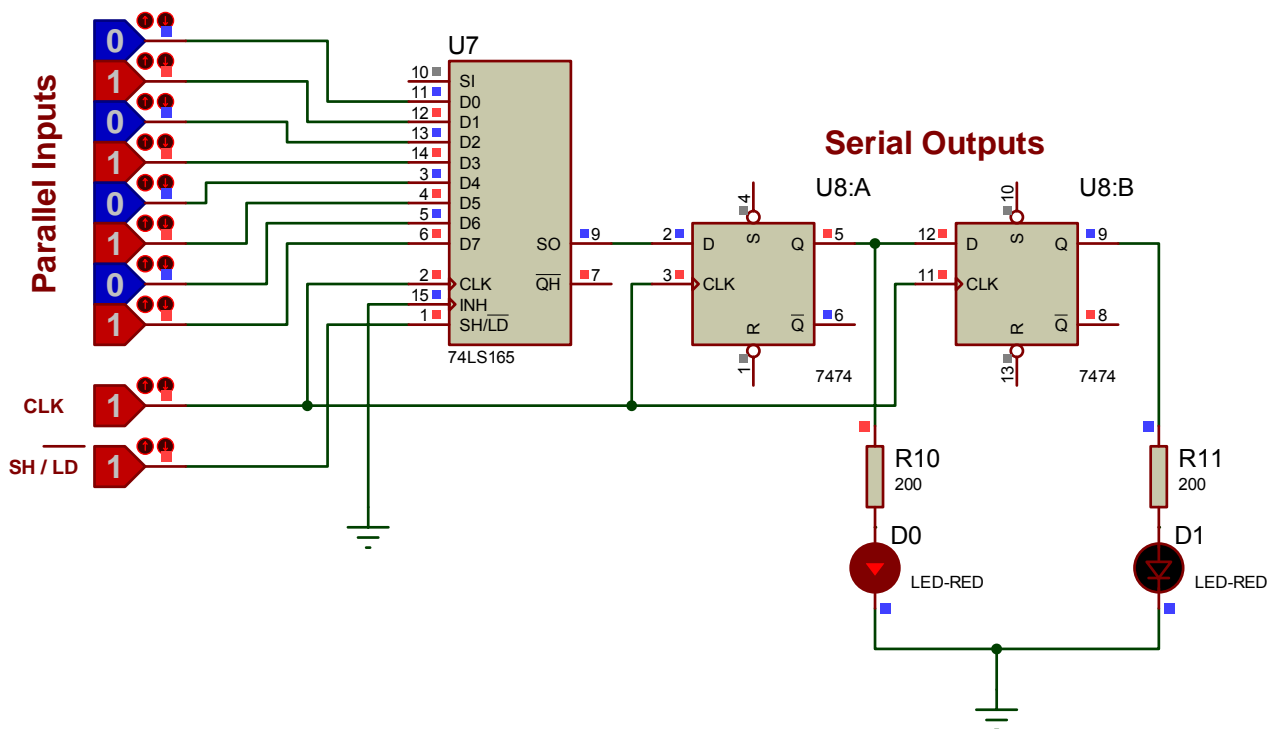
مدار شکل (۱۳-۲) را ببندید و یک ورودی دلخواه به مدار اعمال کنید. برای مشاهده بهتر تغییرات در خروجی، مقادیر

نشان داده شده در شکل (۱۳-۲) را می‌توانید اعمال کنید. سپس پایه Shift/Load را '۰' کنید تا ورودی‌ها به صورت موازی در

آی‌سی بارگذاری شوند. سپس این پایه را '۱' کرده و کلاک را به صورت دستی تغییر دهید. در لبه بالا رونده کلاک (تغییر از صفر

به یک)، بیت‌های بارگذاری شده به صورت سریال از پایه SO خارج خواهند شد و می‌توانید منطق بیت خارج شده را از طریق LED/D_۱ مشاهده نمایید. با هر کلاک، بیت قبلی به سمت راست شیفت داده می‌شود و منطق آن را می‌توانید در LED/D_۰ مشاهده کنید. با هشت کلاک، تمام بیت‌ها بصورت سریال از آی سی خارج خواهند شد. در حین انجام آزمایش، جدول (۲-۱۳) را نیز تکمیل کنید.

توجه کنید که برای انجام آزمایش می‌توانید تنها از یک LED در خروجی استفاده کنید. هرچه تعداد LED ها بیشتر باشد، بهتر می‌توانید نحوه انتقال بیت‌ها بصورت سریال را مشاهده کنید.



شکل (۲-۱۳). مدار مربوط به آزمایش چهارم

جدول (۲-۱۳). جدول مربوط به آزمایش چهارم

| مقدار بارگذاری شده $D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$ | منطق مشاهده شده در LED/D_0 | | | | | | | |
|---|------------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | ۱ th CLK | ۲ th CLK | ۳ th CLK | ۴ th CLK | ۵ th CLK | ۶ th CLK | ۷ th CLK | ۸ th CLK |
| ۱ ۰ ۱ ۰ ۱ ۰ ۱ ۰ | ؟ | ؟ | ؟ | ؟ | ؟ | ؟ | ؟ | ؟ |

آزمایش ۵

موضوع آزمایش: بررسی عملکرد باس موازی

مقدمه

به علت تعدد تجهیزات متصل شده به CPU در یک سیستم میکروپروسسوری برای داشتن سرعت بالای انتقال اطلاعات، باید از اتصالات و انتقال موازی داده‌ها استفاده کرد. عملاً اتصال مستقل تک تک اجزاء و خطوط داده زیاد به میکروپروسسور غیر ممکن است، چرا که موجب افزایش فراوان اتصالات سیمی خواهد شد و در نهایت منجر به کاهش قابلیت اطمینان در سیستم می‌گردد. به همین دلیل، در این سیستمها از مجموعه‌های ارتباطی تحت عنوان باس استفاده می‌شود. منظور از باس مجموعه خطوطی است که به صورت یک مسیر مشترک به بخشهای مختلف سیستم متصل شده و امکان برقراری ارتباط بین آنها را فراهم می‌کند. همانطور که در شکل (۲-۱۴) نیز نشان داده شده است، این باس خود می‌تواند متشکل از چند باس یا مجموعه خطوط دیگر باشد:

✓ باس داده‌ها

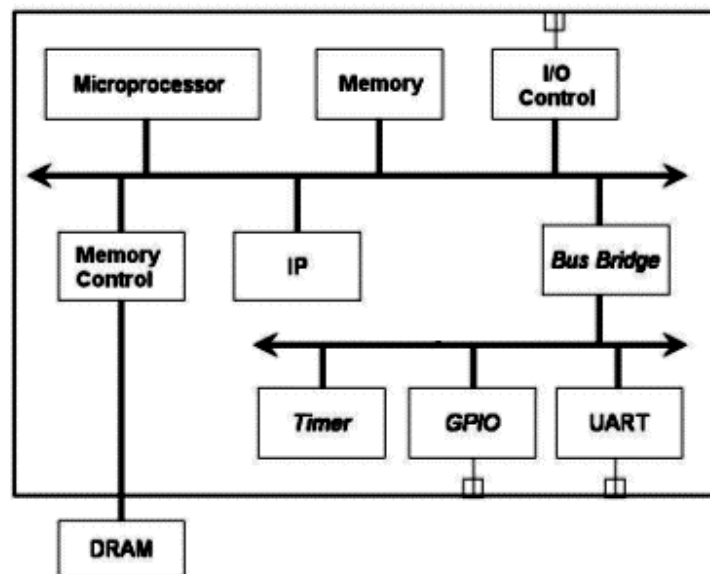
عبارتست از مجموعه‌ای از خطوط که برای مبادله داده‌ها، عمدتاً بین CPU و بقیه اجزا سیستم مورد استفاده قرار می‌گیرد. واضح است که خطوط داده فوق باید خطوطی دو طرفه باشند و توانایی انتقال داده در دو جهت ورودی و خروجی را داشته باشند.

✓ باس آدرس

بر خلاف خطوط داده‌ها، خطوطی یک طرفه می‌باشند که معمولاً از طرف میکروپروسسور به سمت سایر واحدهای سیستم سیگنال ارسال می‌نمایند و حامل آدرسی جهت خواندن و یا نوشتن داده‌ها می‌باشند.

✓ باس کنترل

این خطوط همچنانکه از نامشان برمی‌آید، برای کنترل اعمال مختلف در سیستم به کار می‌روند و عموماً عامل کنترل نحوه ارتباط میکروپروسسور با سایر واحدها می‌باشند.



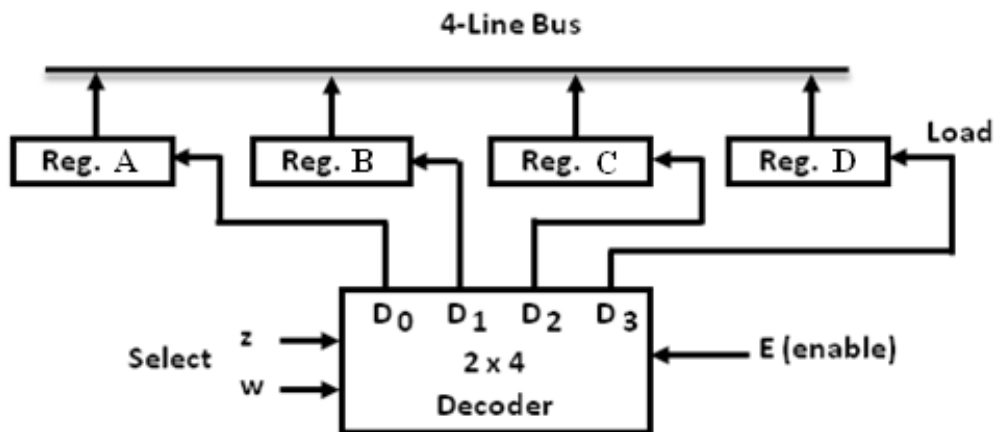
شکل (۲-۱۴). نحوه ارتباط اجزا سیستم توسط باس

آنچه در اینجا باید مورد توجه قرار بگیرد این است که به علت مشترک بودن مسیر ارتباطی، این مسیر در هر لحظه تنها می‌تواند ارتباط بین دو وسیله را برقرار کند که یکی از آنها اطلاعاتی را به صورت خروجی بر روی باس قرار داده و دیگری بصورت ورودی آن را دریافت نماید. به این ترتیب، در این شرایط سایر وسایل متصل به باس یا باید به طور کامل از باس جدا شوند و یا اینکه بخش اتصالی‌شان به باس به عنوان ورودی عمل نمایند. به عبارت دیگر، در هر لحظه تنها یک وسیله حق قرار دادن اطلاعات روی باس را دارد. بدین منظور، می‌توان از بافرهای سه حالت با قابلیت امپدانس بالا در خروجی استفاده کرد. در آزمایش ۳ در مورد آی سی بافر سه حالت و مدار داخلی آن توضیحاتی داده شده است که در صورت نیاز می‌توانید به آن مراجعه کنید.

طراحی باس موازی با استفاده از دیکدر

یکی از روشهای ساخت سیستم گذرگاه مشترک، استفاده از دیکدر می‌باشد. در اینجا انتخاب ثبات مبدا بر عهده دیکدر و خطوط انتخاب S_0 و S_1 می‌باشد. ثبات انتخاب شده داده‌ها را بر روی گذرگاه قرار داده و بقیه ثباتها در حالت امپدانس بالا باید

باشند. بنابراین باید از رجیسترهایی که دارای خروجی امپدانس بالا هستند، استفاده کرد. شکل (۱۵-۲) یک گذرگاه مشترک ۴ بیتی را برای ۴ ثابت نشان می‌دهد که بر اساس دیکدر طراحی شده است. بر اساس مقادیر خطوط انتخاب S_1 و S_0 یکی از ثباتهای A, B, C و یا D انتخاب می‌شوند (جدول ۱۴-۲).



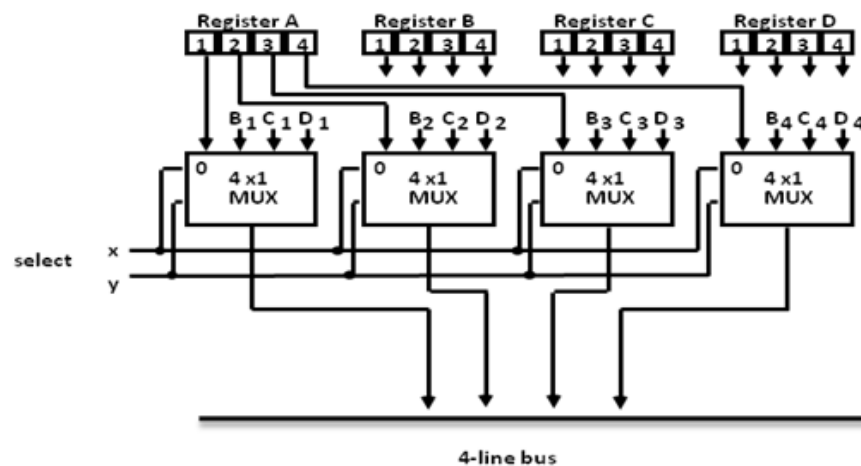
شکل (۱۵-۲). نحوه ارتباط چهار ثابت ۴ بیتی با استفاده از باس

جدول (۱۴-۲). نحوه انتخاب ثباتها در باس

| S_1 | S_0 | ثباتی که انتخاب می‌شود |
|-------|-------|------------------------|
| ۰ | ۰ | A |
| ۰ | ۱ | B |
| ۱ | ۰ | C |
| ۱ | ۱ | D |

طراحی باس موازی با استفاده از مالتی پلکسر

یک روش دیگر ساخت سیستم گذرگاه مشترک، استفاده از مولتی پلکسرها می باشد. مولتی پلکسرها از بین داده های ثباتهای مختلف، خروجی ثبات مبدا را انتخاب و سپس اطلاعات آن را روی گذرگاه قرار می دهند. توجه کنید که در اینجا لازم نیست از رجیسترهایی که دارای خروجی امپدانس بالا نیز هستند، استفاده کرد (چرا؟). شکل (۱۶-۲) یک گذرگاه مشترک ۴ بیتی را برای ۴ ثبات نشان می دهد که بر اساس مالتی پلکسر طراحی شده است. بر اساس مقادیر خطوط انتخاب S_1 و S_0 یکی از ثباتهای A, B, C و یا D انتخاب می شوند. نحوه عمل گذرگاه چهاربیتی شکل (۱۶-۲) در جدول (۱۵-۲) نشان داده شده است.



شکل (۱۶-۲). گذرگاه چهار بیتی برای چهار ثبات

جدول (۱۵-۲). نحوه عملکرد گذرگاه

| S_1 | S_0 | ثباتی که انتخاب می شود |
|-------|-------|------------------------|
| ۰ | ۰ | A |
| ۰ | ۱ | B |
| ۱ | ۰ | C |
| ۱ | ۱ | D |

روش انجام آزمایش

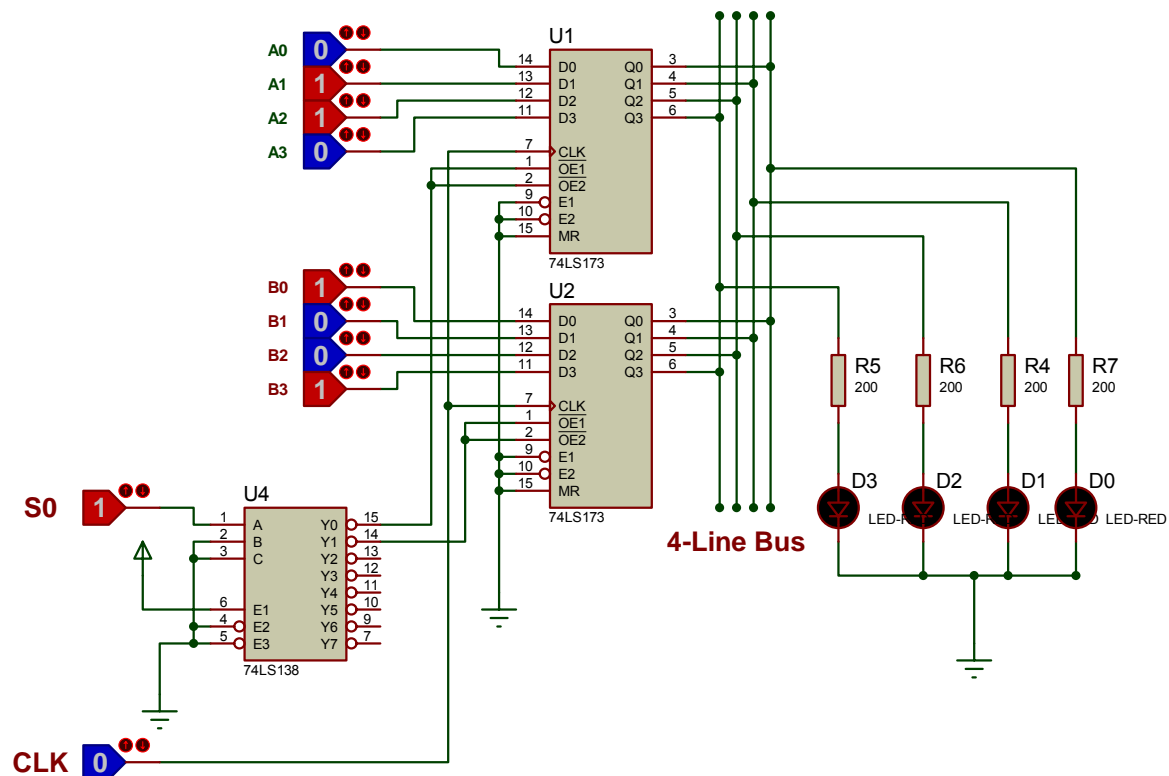
(طراحی باس موازی با استفاده از دیگدر)

یک ثابت، مجموعه‌ای از سلولهای حافظه است که می‌تواند اطلاعات دودویی را در خود نگهدارد. چون هر فلیپ فلاپ سلولی است که قادر به ذخیره یک بیت از اطلاعات می‌باشد، مجموعه‌ای از فلیپ فلاپها یک ثابت را بوجود می‌آورند. ثابتها یا رجیسترهای داخلی به طور معمول در اغلب میکروپروسسورها وجود دارند. این ثابتها عموماً در داخل میکروپروسسور برای ثبت موقت اطلاعات بکار می‌روند. برخی از این ثابتها بطور ویژه در اعمال داخلی میکروپروسسور استفاده می‌شوند و برخی دیگر ثابتهای با کاربرد عام می‌باشند و در دسترس استفاده کننده قرار دارند. تعداد و اندازه این ثابتها بستگی به میکروپروسسور و تعداد بیت آن دارد. ثابتهای با کاربرد عام برای ذخیره سازی موقت داده مورد پردازش توسط برنامه نویس سیستم بکار می‌رود و باعث سرعت بخشیدن به اجرای دستورالعمل ها یا جلوگیری از اجرای سیکل های اضافی ارتباط با حافظه خارجی جهت ذخیره سازی و بازیابی اطلاعات می‌شوند.

آی سی ۷۴LS۱۷۳ یک ثابت با چهار فلیپ فلاپ D و خروجی سه حالت می‌باشد. در لبه بالا رونده کلاک، ورودی به خروجی آن منتقل می‌شود. برای عملکرد نرمال ثابت، پایه های فعال کننده ورودی IE_1 و IE_2 و همچنین پایه های فعال کننده خروجی OE_1 و OE_2 بایستی به زمین متصل شوند. اگر پایه های OE_1 و OE_2 به منطق '۱' متصل شوند، خروجی در حالت امپدانس بالا قرار می‌گیرد که اساس طراحی مدار باس می‌باشد. بنابراین با استفاده از یک دیگدر با خروجی مکمل شده، در هر لحظه تنها پایه های OE_1 و OE_2 یک ثابت '۰'، و بقیه ثابتها '۱' (امپدانس بالا) خواهند بود. برای سادگی، ما در اینجا مدار را برای تنها دو ثابت ۴ بیتی طراحی کردیم. قبل از بستن مدار، مشخصات آی سی ها را به دقت مطالعه کنید.

بعد از بستن مدار شکل (۲-۱۷)، ورودی های دلخواهی را به ثابتهای A و B اعمال کنید. از آنجایی که ثابتها به لبه بالا رونده حساس می‌باشند، کلاک را یک بار به صورت دستی صفر و سپس یک کنید تا ورودی ها به داخل ثابت منتقل شوند. با هر بار تغییر در داده های ورودی، کلاک را باید به صورت دستی تغییر دهید. چون تنها دو ثابت داریم، بنابراین با یک بیت می‌توان

عمل انتخاب را انجام داد. اگر بیت S_0 ، صفر باشد، ثابت A و اگر S_1 یک باشد، ثابت B به خروجی منتقل می‌شود. برای مشاهده خروجی می‌توانید از LED استفاده کنید. این آزمایش را برای مقادیر جدول (۲-۱۴) نیز انجام داده و آنرا تکمیل کنید.



شکل (۲-۱۷). مدار باس موازی با استفاده از دیکدر

جدول (۲-۱۶). مقادیر مربوط به مرحله اول آزمایش پنجم

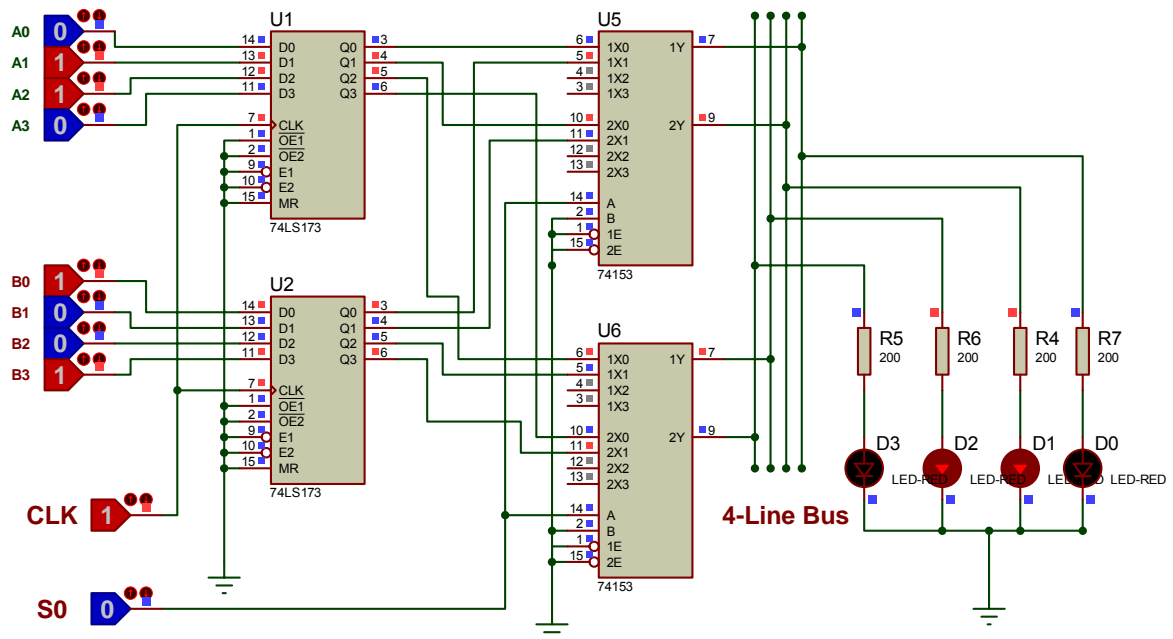
| $A_3 A_2 A_1 A_0$ | $B_3 B_2 B_1 B_0$ | $D_3 D_2 D_1 D_0$ | |
|-------------------|-------------------|-------------------|-------------|
| | | $S_0 = "1"$ | $S_0 = "0"$ |
| ۰۱۱۰ | ۱۰۰۱ | ? | ? |

روش انجام آزمایش

(طراحی باس موازی با استفاده از مالتی پلکسر)

با کمی تغییرات در مدار قبلی، می‌توانید آزمایش مرحله دوم را انجام دهید. در این آزمایش به جای دیکدر از مالتی پلکسر استفاده شده است. به تعداد بیت‌های ورودی، نیاز به بلوکهای مالتی پلکسر داریم. توجه کنید کنید که آی سی ۷۴LS۱۵۳ دارای دو مالتی پلکسر ۴ به ۱ می‌باشد. اگر پایه های $\overline{E1}$ و $\overline{E2}$ هر دو به زمین متصل شوند، هر دو مدار مالتی پلکسر در آی سی فعال می‌شوند. همانطور که در مقدمه نیز اشاره شد، یکی از تفاوت‌های این مدار با مدار قبلی آن است که دیگر لازم نیست ثبات دارای خروجی امپدانس بالا باشد. بنابراین در مدار شکل (۲-۱۸)، هر دو پایه $\overline{OE1}$ و $\overline{OE2}$ به زمین متصل شده‌اند.

بعد از بستن مدار، ورودی‌های دلخواهی را به ثبات‌های A و B اعمال کنید. از آنجایی که ثبات‌ها به لبه بالا رونده حساس می‌باشند، کلاک را یک بار به صورت دستی صفر و سپس یک کنید تا ورودی‌ها به داخل ثبات منتقل شوند. بنابراین با هر بار تغییر در داده‌های ورودی، کلاک را باید به صورت دستی تغییر دهید. چون تنها دو ثبات داریم، بنابراین با یک بیت می‌توان عمل انتخاب را انجام داد و به همین دلیل است که پایه انتخاب ورودی B صفر شده است. اگر بیت S_0 صفر باشد، ثبات A و اگر مقدارش یک باشد، ثبات B به خروجی منتقل می‌شود. برای مشاهده خروجی می‌توانید از LED استفاده کنید. این آزمایش را برای مقادیر جدول (۲-۱۷) نیز انجام داده و آنرا تکمیل کنید.



شکل (۲-۱۸). جایگزینی مالتی پلکسر بجای دیکدر در مدار باس موازی

جدول (۲-۱۷). مقادیر مربوط به مرحله دوم آزمایش پنجم

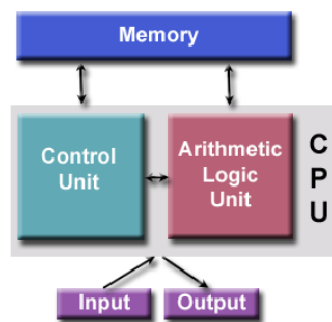
| $A_3 A_2 A_1 A_0$ | $B_3 B_2 B_1 B_0$ | $D_3 D_2 D_1 D_0$ | |
|-------------------|-------------------|-------------------|-------------|
| | | $S_0 = "۱"$ | $S_0 = "۰"$ |
| ۰۱۱۰ | ۱۰۰۱ | ? | ? |

آزمایش ۶

موضوع آزمایش: آشنایی با واحد محاسباتی-منطقی (ALU)

مقدمه

در علم کامپیوتر، واحد حساب و منطق (به اختصار ALU^{۱۲}) یک مدار دیجیتالی است که عملیات های محاسباتی و منطقی را انجام می دهد. ALU بخش بنیادی "واحد پردازش مرکزی (CPU)" می باشد. پردازنده های مرکزی یا پردازنده های گرافیکی مدرن، شامل ALU های بسیار قدرتمند و پیچیده هستند و ممکن است بیش از یک ALU داشته باشند. اکثر عملیات محاسباتی و منطقی پردازنده به وسیله یک یا چند ALU انجام می شود. ALU داده را از ثباتهای ورودی بارگذاری می کند، سپس واحد کنترل^{۱۳} به ALU می گوید که چه عملیاتی را بر روی داده ها انجام دهد. ALU نتایج را بر روی یک رجیستر خروجی ذخیره می کند. اجزای دیگر این اطلاعات را بین رجیسترها و حافظه جابه جا می کنند. شمای کلی نحوه ارتباط ALU با بخشهای مختلف ریزپردازنده در شکل (۱۹-۲) نشان داده شده است.



شکل (۱۹-۲): نحوه ارتباط ALU با بخشهای مختلف ریزپردازنده

ALU ها قادر به انجام عملیاتهای ساده و در مواردی بسیار پیچیده تر باشند:

^{۱۲} Arithmetic Logic Unit

^{۱۳} Control unit

✓ عملیات های ساده:

اکثر ALU ها می توانند این عملیات ها را انجام دهند:

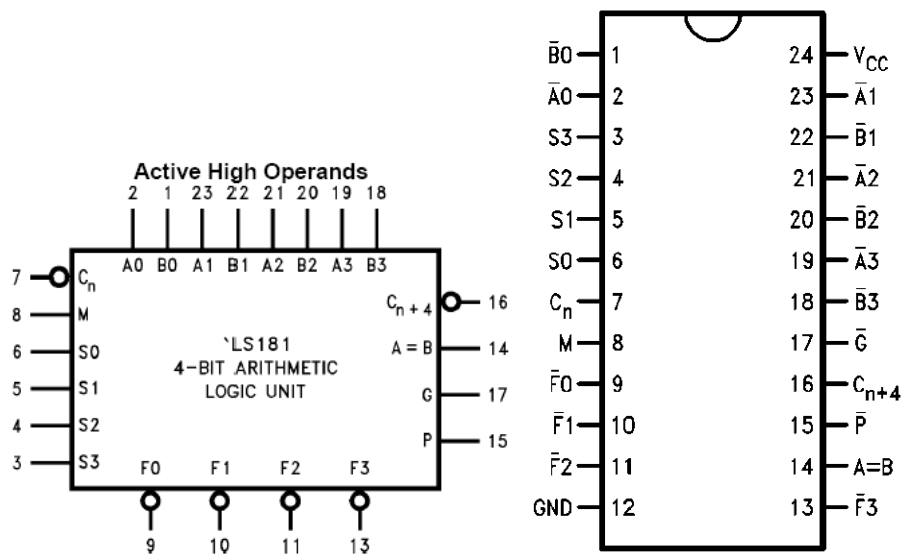
- عملیات های محاسباتی صحیح (جمع ، تفریق و بعضی ضرب و تقسیم ..)
- عملیات های بیتی منطقی (AND ، NOT ، OR ، XOR)
- عملیات های انتقال بیتی (Bit-shifting) (انتقال یا چرخش یک کلمه با تعداد مشخصی بیت به راست یا چپ؛ با/بدون گسترش علامت)

✓ عملیاتهای پیچیده:

مهندسان می توانند واحد حساب و منطق را برای هر عملیاتی طراحی کنند. هر چه قدر که آن پیچیده تر باشد ، به هزینه ، مصرف انرژی و فضای اشغال شده توسط ALU اضافه می شود. پس برای هر پردازنده ، با توجه به توان پردازشی مطلوب ، واحد حساب و منطق مناسب طراحی می شود.

مباحث تکویری

هدف از این آزمایش، آشنایی با نحوه کار یک ALU و انجام چندین عملیات ریاضی و منطقی ساده می باشد. در این آزمایش از آی سی ۷۴LS۱۸۱ استفاده خواهیم کرد که یک ALU چهار بیتی می باشد. در شکل (۲-۲۰)، شمای کلی این آی سی نشان داده شده است.



شکل (۲-۲۰): شمای کلی آی سی ۷۴LS۱۸۱ یک ALU چهاربیتی

در جدول (۲-۱۸) پایه های آی سی تشریح شده اند. این آی سی دارای ورودی برای دو عملوند $\overline{A^0} - \overline{A^3}$ و $\overline{B^0} - \overline{B^3}$ و

پایه های خروجی $\overline{F^0} - \overline{F^3}$ می باشد.

جدول (۲-۱۸). توصیف پایه های آی سی ۷۴LS۱۸۱ برای دو ورودی چهار بیتی

| Pin Names | Description |
|-----------------------------------|-------------------------------------|
| $\overline{A^0} - \overline{A^3}$ | Operand Inputs (Active LOW) |
| $\overline{B^0} - \overline{B^3}$ | Operand Inputs (Active LOW) |
| S0-S3 | Function Select Inputs |
| M | Mode Control Input |
| C_n | Carry Input |
| $\overline{F^0} - \overline{F^3}$ | Function Outputs (Active LOW) |
| A = B | Comparator Output |
| \overline{G} | Carry Generate Output (Active LOW) |
| \overline{P} | Carry Propagate Output (Active LOW) |
| C_{n+4} | Carry Output |

همانطور که گفته شد، ALU قادر به انجام عملیاتهای ریاضی ویا منطقی می باشد. با پایه M می توان نوع عملیات را مشخص کرد. اگر $M=1$ باشد، عملیات منطقی و اگر $M=0$ باشد، عملیات ریاضی خواهد بود. با استفاده از پایه های S_0-S_3 ، می توان یکی از ۱۶ عملیات منطقی و یا ریاضی مختلف را انتخاب کرد. در جدول (۲-۱۹) انواع مختلف عملیاتهای منطقی و ریاضی با توجه به ورودی های S_0-S_3 و پایه M نشان داده شده است. پایه C_n مربوط به بیت نقلی ورودی و C_{n+4} بیت نقلی خروجی را نشان می دهد. پایه $(A=B)$ هنگامی که تمام بیت های خروجی '۱' باشند، به حالت امیدانس بالا می رود و در غیر اینصورت '۰' می باشد. از این بیت می توان برای مشخص کردن مساوی بودن دو عملوند A و B هنگام انجام عمل تفریق استفاده کرد. برای مقایسه بیش از ۴ بیت، می توان از گیت AND سیمی^{۱۴} استفاده کرد. با استفاده از خروجی سیگنال $A=B$ و C_{n+4} می توان حالت های $A>B$ و یا $A<B$ را آشکار کرد.

همانطور که در جدول (۲-۱۹) نیز مشاهده می کنید، دستورالعملهای ریاضی $\overline{Cn} = L$ همان دستورالعملهای $\overline{Cn} = H$ می باشند، با این تفاوت که حاصل "بعلاوه یک" می شود.

جدول (۲-۱۹). نحوه عمل آی سی ۷۴LS۱۸۱ به ازای مقادیر مختلف ورودی های انتخاب

^{۱۴} -Wired-And

FUNCTION TABLE (ACTIVE HIGH)

| SELECTION | | | | | ACTIVE-HIGH DATA | | |
|-----------|----|----|----|-----------------------------|---|---|--|
| | | | | | M = L; ARITHMETIC OPERATIONS | | |
| S3 | S2 | S1 | S0 | M = H LOGIC FUNCTION | $\overline{Cn} = H$ (no carry) | $\overline{Cn} = L$ (with carry) | |
| L | L | L | L | $F = A$ | $F = A$ | $F = A \text{ PLUS } 1$ | |
| L | L | L | H | $F = \overline{A + B}$ | $F = A + B$ | $F = (A + B) \text{ PLUS } 1$ | |
| L | L | H | L | $F = \overline{AB}$ | $F = A + \overline{B}$ | $F = (A + \overline{B}) \text{ PLUS } 1$ | |
| L | L | H | H | $F = 0$ | $F = \text{MINUS } 1(2's \text{ COMPL})$ | $F = 0$ | |
| L | H | L | L | $F = \overline{AB}$ | $F = A \text{ PLUS } \overline{AB}$ | $F = A \text{ PLUS } \overline{AB} \text{ PLUS } 1$ | |
| L | H | L | H | $F = \overline{B}$ | $F = (A + B) \text{ PLUS } \overline{AB}$ | $F = (A + B) \text{ PLUS } \overline{AB} \text{ PLUS } 1$ | |
| L | H | H | L | $F = A \oplus B$ | $F = A \text{ MINUS } B \text{ MINUS } 1$ | $F = A \text{ MINUS } B$ | |
| L | H | H | H | $F = \overline{AB}$ | $F = \overline{AB} \text{ MINUS } 1$ | $F = \overline{AB}$ | |
| H | L | L | L | $F = \overline{A + B}$ | $F = A \text{ PLUS } AB$ | $F = A \text{ PLUS } AB \text{ PLUS } 1$ | |
| H | L | L | H | $F = \overline{A \oplus B}$ | $F = A \text{ PLUS } B$ | $F = A \text{ PLUS } B \text{ PLUS } 1$ | |
| H | L | H | L | $F = B$ | $F = (A + \overline{B}) \text{ PLUS } AB$ | $F = (A + \overline{B}) \text{ PLUS } AB \text{ PLUS } 1$ | |
| H | L | H | H | $F = AB$ | $F = AB \text{ MINUS } 1$ | $F = AB$ | |
| H | H | L | L | $F = 1$ | $F = A \text{ PLUS } A^*$ | $F = A \text{ PLUS } A \text{ PLUS } 1$ | |
| H | H | L | H | $F = A + \overline{B}$ | $F = (A + B) \text{ PLUS } A$ | $F = (A + B) \text{ PLUS } A \text{ PLUS } 1$ | |
| H | H | H | L | $F = A + B$ | $F = (A + \overline{B}) \text{ PLUS } A$ | $F = (A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$ | |
| H | H | H | H | $F = A$ | $F = A \text{ MINUS } 1$ | $F = A$ | |

نمونه انجام آزمایش

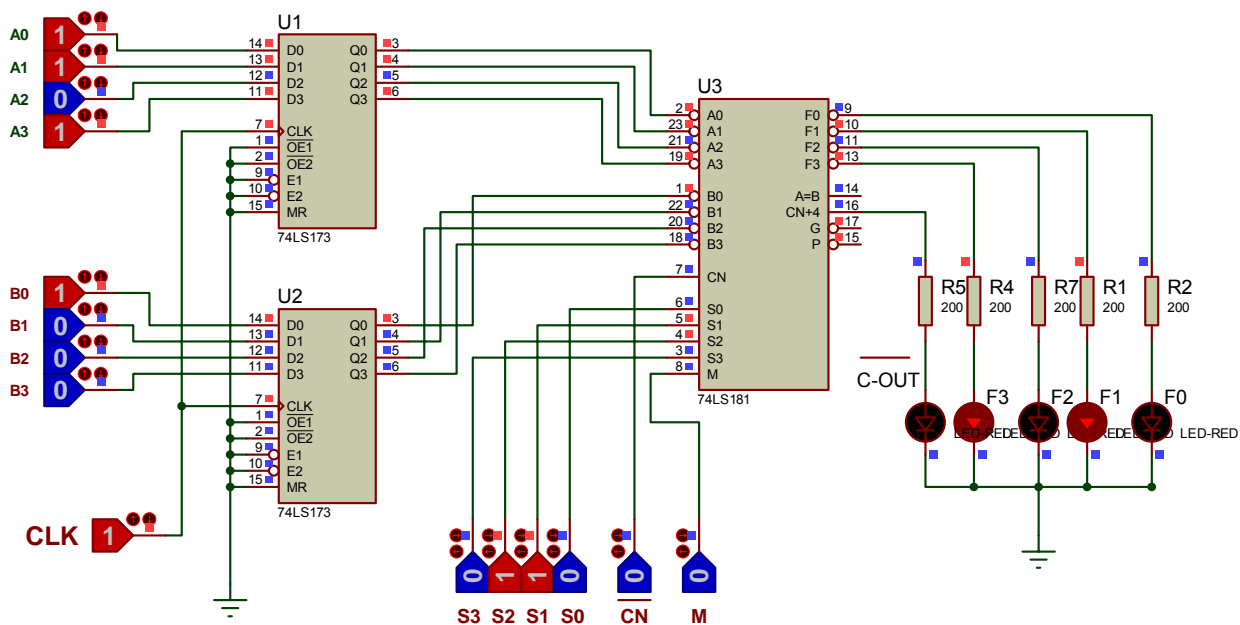
بعد از بستن مدار شکل (۲-۲۱)، ورودی‌های نشان داده شده در جدول ثباتهای A و B را اعمال کنید. از آنجایی که

ثباتها به لبه بالا رونده حساس می‌باشند، کلاک را یک بار به صورت دستی صفر و سپس یک کنید تا ورودی‌ها به داخل ثبات

منتقل شوند. بنابراین با هر بار تغییر در داده‌های ورودی، کلاک را باید به صورت دستی تغییر دهید.

عملیاتهای منطقی و ریاضی خواسته در جداول (۲-۲۰) و (۲-۲۱) را انجام داده و نتایج بدست آمده را در جدول

یادداشت نمایید.



شکل (۲-۲۱). یک واحد ALU چهاربیتی که طبق جدول (۲-۱۷) رفتار می کند.

جدول (۲-۲۰). یافتن خروجی های ALU برای دو ورودی چهاربیتی و مقادیر مختلف انتخاب

| A | B | M | $S_3 S_2 S_1 S_0$ | $F_3 F_2 F_1 F_0$ |
|------|------|---|-------------------|-------------------|
| ۱۰۰۰ | ۱۰۱۱ | H | LLHL | ? |
| ۱۰۰۰ | ۱۰۱۱ | H | LHHL | ? |
| ۱۰۰۰ | ۱۰۱۱ | H | HLLL | ? |
| ۱۰۰۰ | ۱۰۱۱ | H | HHHL | ? |
| ۱۰۰۰ | ۱۰۱۱ | H | HLHH | ? |

جدول (۲-۲۱). یافتن خروجی‌های ALU برای دو ورودی چهاربیتی و مقادیر مختلف انتخاب و C_n

| A | B | M | C_n | $S_3 S_2 S_1 S_0$ | $F_3 F_2 F_1 F_0$ | Cout |
|------|------|---|-------|-------------------|-------------------|------|
| ۱۰۰۰ | ۱۰۱۱ | H | ۰ | LLLH | ? | ? |
| ۱۱۰۰ | ۱۰۱۱ | L | ۱ | LHHL | ? | ? |
| ۱۰۰۰ | ۱۰۱۱ | H | ۰ | LHHH | ? | ? |
| ۱۰۰۰ | ۱۰۰۰ | L | ۱ | LHHL | ? | ? |
| ۱۰۰۰ | ۱۰۱۱ | L | ۱ | LHHL | ? | ? |

تمرین

۱- همانطور که اشاره شد، خروجی پایه $(A=B)$ برای مشخص کردن مساوی بودن دو عملوند A و B، هنگام انجام عمل تفریق با استفاده از مکمل ۲ می‌تواند بکار برده شود. با استفاده از یک LED و منطق دیود-مقاومت، مداری طراحی کنید که هنگام مساوی بودن A و B، LED روشن شود.

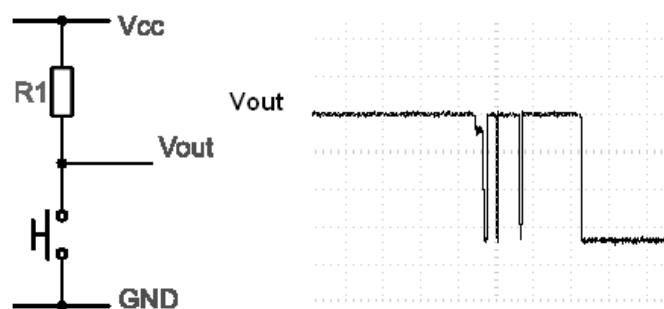
۲- همانطور که اشاره شد، برای مقایسه بیش از ۴ بیت، می‌توان از AND سیمی خروجی پایه‌های $(A=B)$ استفاده کرد. فرض کنید می‌خواهیم دو عدد ۸ بیتی را با یکدیگر مقایسه کنیم. با استفاده از AND سیمی و یک LED مداری طراحی کنید که در صورت مساوی بودن این دو عملوند ۸ بیتی، LED روشن شود.

آزمایش ۷-الف

موضوع آزمایش: طراحی مدار نویزگیر^{۱۵}

مباحث تئوری

صفحه کلید از سوئیچ بمنظور اعمال تغییر در جریان مربوط به مدارات صفحه کلید استفاده می‌نماید. زمانی که کلیدی فشرده می‌گردد، میزان اندکی لرزش بین سطوح تماس بوجود می‌آید که bounce نامیده می‌شود. به عنوان مثال، یک سوئیچ ساده مکانیکی را که در شکل (۲-۲۲) نشان داده شده است در نظر بگیرید. هنگامی که کلید بسته می‌شود، خروجی بلافاصله صفر نمی‌شود، بلکه به خاطر لرزشهای مکانیکی که ممکن است ۳۰۰ ms طول بکشد، خروجی در حال تغییر خواهد بود.

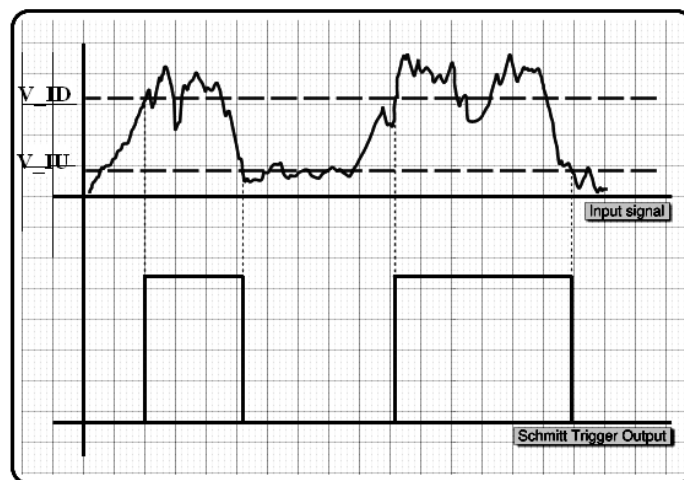


شکل (۲-۲۲): Bounce در کلیدهای مکانیکی

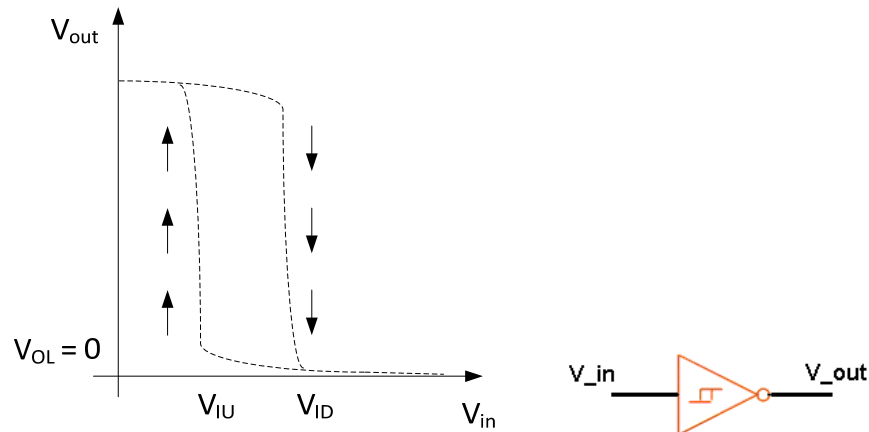
یکی از روشهای حذف لرزش در کلیدهای مکانیکی، استفاده از گیت اشmitt تریگر می‌باشد. در شکل (۲-۲۳) یک موج ورودی و شکل موج خروجی حاصل از یک معکوس کننده اشmitt تریگر نشان داده شده است. در بررسی این شکل موج ها مشاهده می‌شود که انتقال بین خروجی از بالا به پایین زمانی اتفاق می‌افتد که مقدار ورودی از ولتاژ V_{IU} کمتر شود. همچنین انتقال ولتاژ خروجی از حالت پایین به بالا به ازای V_{ID} صورت می‌گیرد. در شکل (۲-۲۴) مشخصه یک گیت اشmitt تریگر نشان داده

^{۱۵} -Debounce

شده است. اگر ولتاژ ورودی افزایش یابد خروجی در V_{ID} تغییر می‌کند و اگر ولتاژ ورودی کاهش یابد تغییر سطح در V_{IU} رخ می‌دهد.



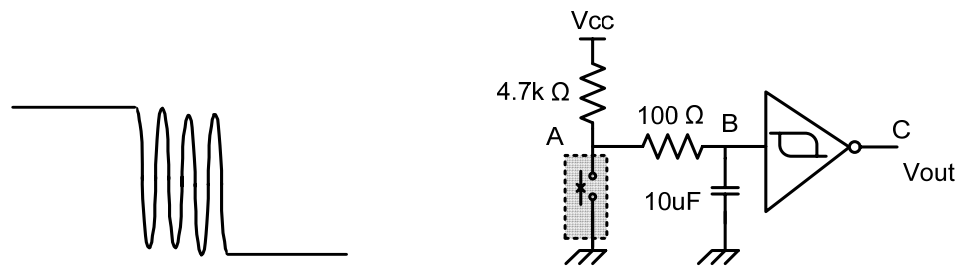
شکل (۲-۲۳)



شکل (۲-۲۴): نماد یک گیت معکوس کننده اشمیت تریگر و مشخصه آن

یکی از مدارهایی که می‌توان برای حذف لرزش استفاده کرد، در شکل (۲-۲۵) نشان داده شده است. هنگامی که دکمه را فشار می‌دهیم، طبق معمول در نقطه A لرزش داریم. در اثر این لرزش، خازن به طور پیاپی شارژ و دشارژ می‌شود. اما توجه کنید

که این خازن باعث می‌شود تا نسبت تغییرات در ولتاژ نقطه B کمتر از نقطه A شود که علت آن زمان نسبتاً طولانی برای شارژ و دشارژ آن می‌باشد. همانطور که در شکل (۲-۲۵) نشان داده شده است، لرزش بعد از عبور از گیت اشمیتتریگر و در نقطه C تقریباً به صورت کامل از بین می‌رود. از بین رفتن لرزشهای نقطه B به علت خواص هیستریزس گیت اشمیتتریگر می‌باشد.



(ب)

(الف)

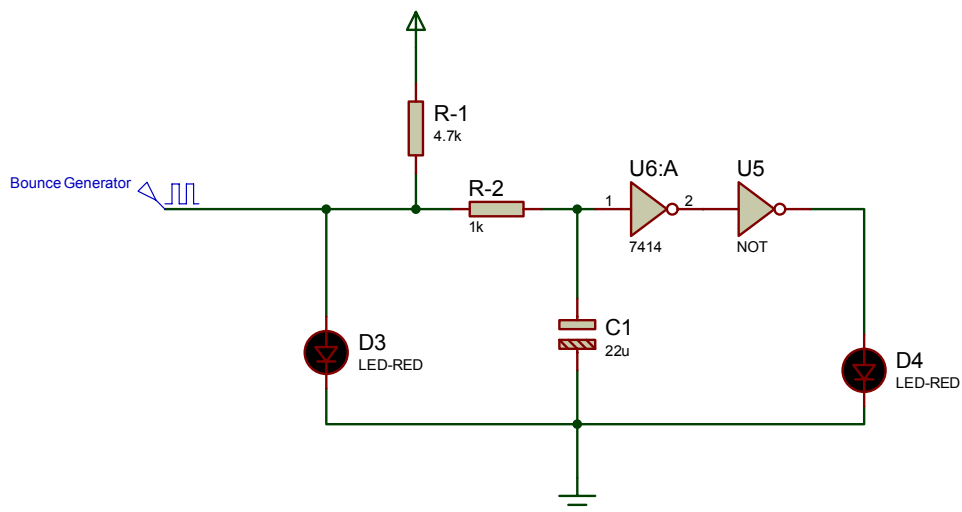


(د)

(ج)

شکل (۲-۲۵): (الف) مدار طراحی شده برای حذف Bounce (ب) نوسانات در نقطه A (ج) نوسانات در نقطه B (د) ولتاژ در نقطه C

نرم افزار پروتئوس، یک محیط ایده آل و عاری از نویز می‌باشد. بنابراین در سوئیچهای استفاده شده Bounce یا لرزشی هم روی نمی‌دهد. برای بررسی مدار طراحی شده باید نویز را بصورت دستی ایجاد کنیم. برای تولید نویز از DCLOCK با فرکانس ۲۰ Hz استفاده شده است. اگر مدار شکل (۲-۲۶) را شبیه سازی شده کنید، مشاهده می‌کنید LED قرار داده شده در ورودی به صورت پراکنده روشن و خاموش می‌شود اما LED قرار داده شده در خروجی روشن نمی‌شود، که به معنای حذف Bounce در خروجی می‌باشد. در این مدار کافی است NOT اول از نوع اشمیتتریگر باشد و NOT دوم می‌تواند معمولی باشد.



شکل (۲-۲۶): مدار شبیه سازی شده برای حذف Bounce

روش انجام آزمایش

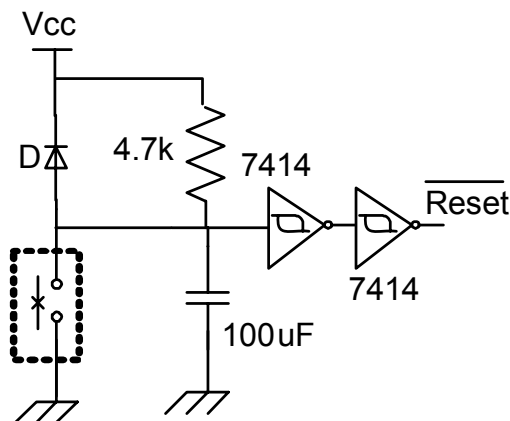
آی سی ۷۴۱۴ دارای ۶ گیت NOT اشمیت ترینر می باشد. پس از مطالعه مشخصات آن از دیتابوک، مدار طراحی شده در شکل (۲-۲۶) را برای حذف لرزش بر روی بردبرد بسته و با اعمال یک سیگنال مربعی با فرکانس نسبتاً پایین در ورودی، شکل موج خروجی را توسط LED یا اسیلوسکوپ مشاهده کنید.

آزمایش ۷-ب

موضوع آزمایش: طراحی مدار Reset

مباحث تئوری

ورودی Reset که در تمامی میکروپروسسورها وجود دارد، در بعضی به صورت فعال سطح بالا و در بعضی دیگر فعال روی سطح پائین است. وظیفه این پایه بردن میکروپروسسور به یک وضعیت مشخص و از پیش تعیین شده است. به عنوان مثال، این حالت در $Z80$ ، $H000$ است. هر سیستم دیجیتال ترتیبی، پس از اعمال تغذیه معمولاً شرایط غیر مشخصی در رجیسترها یا فلیپ فلاپهای خود دارد. به همین جهت و برای تعیین یک وضعیت مناسب برای شروع بکار میکروپروسسور، معمولاً پس از اعمال تغذیه، میکروپروسسور باید به وضعیت Reset برده شود. این کار با استفاده از مدار شکل (۲-۲۷) امکان پذیر است.



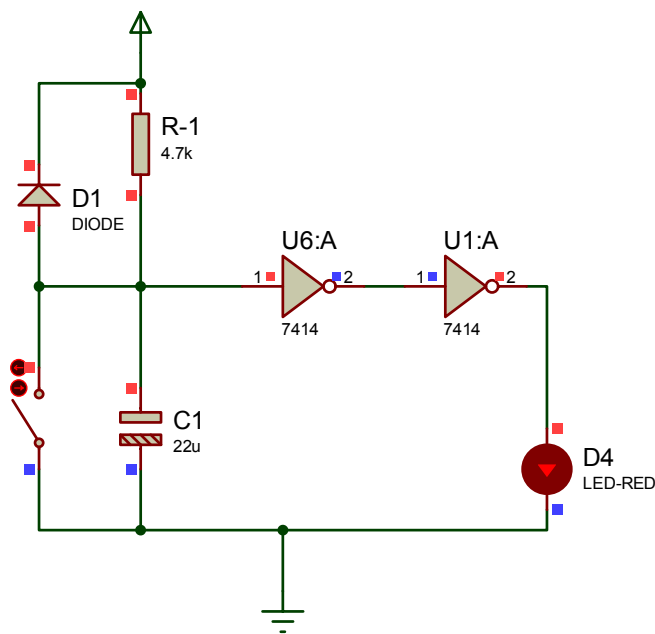
شکل (۲-۲۷): یک نمونه مدار Reset

به این ترتیب، مدار تا زمان رسیدن مقدار شارژ خازن به یک حالت پایدار، در وضعیت Reset باقی می ماند و سپس به میکروپروسسور اجازه شروع به کار داده می شود. با فشار دادن سوئیچ، خازن C شروع به تخلیه می کند و باعث Reset مدار در

زمان دلخواه می‌شود. دیود D در این مدار به این منظور قرار داده شده است که چنانچه به هر علت اتصال کوتاهی در تغذیه مدار ایجاد شود و موجب صفر شدن لحظه ای ولتاژ تغذیه گردد، دیود D باعث تخلیه سریع خازن C گشته و بنابراین مدار را به نحو مناسبی Reset می‌نماید. در مدار شکل (۲۷-۲) از دو گیت اشمیت تریگر استفاده شده است. نحوه عملکرد مدار را به صورت دقیقتر تشریح کنید.

روش انجام آزمایش

پس از مطالعه مشخصات آی سی ۷۴۱۴ از دیتابوک، مدار شکل (۲۸-۲) را بر روی بردبورد ببندید. در حالت عادی، LED روشن می‌باشد و پس از فشار دادن دکمه Reset، LED سریع خاموش می‌شود. با رها کردن دکمه Reset، LED دوباره پس از چند ثانیه روشن می‌شود. با قطع منبع تغذیه نیز مشاهده خواهید کرد LED بلافاصله از طریق دیود دشارژ شده و خاموش می‌گردد.



شکل (۲۸-۲): یک مدار Reset ساده با استفاده از آی سی ۷۴۱۴

آزمایش ۸

موضوع آزمایش: تولید دنباله های زمانی

مقدمه

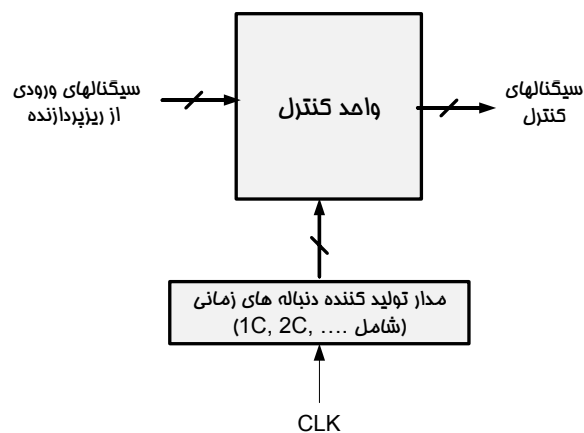
در هر میکروپروسسور، مجموعه‌ای از دستورالعملها که برنامه نامیده می‌شوند، در حافظه خارجی قرار دارد که باید یک به یک توسط میکروپروسسور فراخوانده^{۱۶} و سپس اجرا^{۱۷} شوند. به مجموع مراحل فراخوانی و اجرای هر دستورالعمل، اصطلاحاً سیکل دستورالعمل گفته می‌شود. در هر میکروپروسسور، سیکل دستورالعمل می‌تواند به یک یا چند بخش کوچکتر تقسیم شود که اصطلاحاً سیکل ماشین^{۱۸} خوانده می‌شود. هر سیکل ماشین عموماً متشکل از چند سیکل ساعت^{۱۹} است. در یک میکروپروسسور ساده، هر سیکل ماشین از چهار سیکل ساعت تشکیل شده است. در اینجا، سیکل ماشین فراخوانی را با C و سیکل ماشین دستورالعمل را با C نشان خواهیم داد. برای نشان دادن چهار سیکل ساعت موجود در هر سیکل ماشین از شماره استفاده خواهیم کرد؛ به عنوان مثال $C, 2C, 3C, 4C$. در شکل (۲-۲۹) سیکل‌های ساعت نشان داده شده‌اند.

واحد کنترل به عنوان مغز و بخش تصمیم گیرنده در یک میکروپروسسور مطرح می‌گردد. وظیفه این واحد، کنترل تمامی اعمال داخلی و خارجی انجام شده توسط میکروپروسسور می‌باشد. این واحد ابتدا مقدار شمارنده برنامه را بر روی خطوط آدرس خارجی منتقل و سیگنالهای لازم برای امکان پذیر شدن خواندن حافظه را تامین می‌نماید. سپس در فاصله زمانی مناسبی، امکان انتقال داده‌های خوانده شده به رجیستر دستورالعمل را فراهم نموده و از خروجی دیکدر دستورالعمل، اطلاعات لازم در مورد دستورالعمل مربوطه را بدست می‌آورد. واحد کنترل بر اساس این اطلاعات، ترتیب انجام اعمال لازم جهت اجرای دستورالعمل مربوطه را تعیین و سیگنالهای لازم را به ترتیب فعال می‌نماید. در پایان اجرای هر دستورالعمل، واحد کنترل مجدداً شرایطی را فراهم می‌نماید تا میکروپروسسور آماده خواندن دستورالعمل بعدی از حافظه باشد.

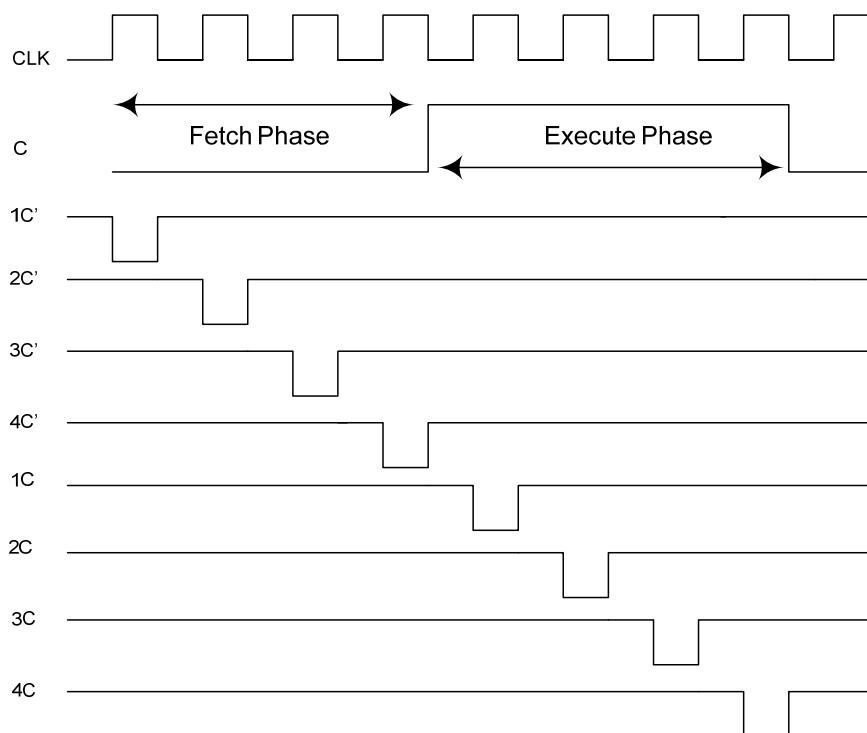
^{۱۶} Fetch^{۱۷} Execute^{۱۸} Machine Cycle^{۱۹} Clock Cycle

همانطور که در شکل (۲-۲۹) مشاهده می کنید، برای تولید سیگنالهای کنترل، نیاز به تولید دنباله های زمانی می باشد که

در واقع موضوع این آزمایش می باشد.



شکل (۲-۲۹). ارتباط دنباله های زمانی با سیگنالهای کنترلی

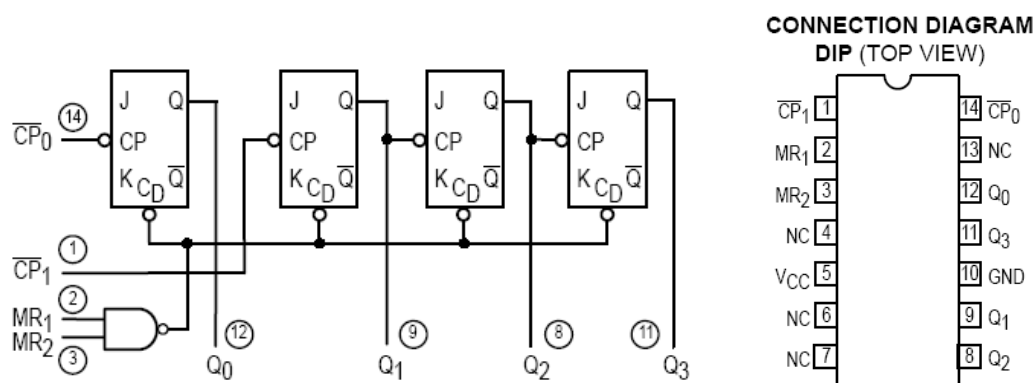


شکل (۲-۳۰). ارتباط زمانی کلاک و سیکل ماشین در فازهای واکنشی و اجرا

مباحث تئوری

در شکل (۳۱-۲) یک آی سی شمارنده و مدار داخلی آن نشان داده شده است. این شمارنده چهار بیت است و قادر به شمارش اعداد ۰ تا ۱۵ می باشد. اساس ساخت مدار تولید کننده دنباله های زمانی با استفاده از یک شمارنده ۷۴۹۳ و یک دیکدر ۷۴۱۳۸ در شکل (۳۲-۲) نشان داده شده است. آی سی ۷۴۳۹ یک شمارنده چهاربیتی نشان داده شده است که می تواند از ۰-۱۵ را درحالتی که CLK-B به خروجی Q_A متصل می شود، شمارش نماید. با هر لبه پایین رونده سیگنال متصل شده به پایه CLK-A، خروجی شمارنده یکی اضافه می شود. همانطور که در جدول (۲۲-۲) نشان داده شده است، آگه پایه های MR_1 و MR_2 به '۱' متصل شوند، شمارنده بازنشاندن شده و خروجی '۰' خواهد شد. بنابراین در ابتدای آزمایش با استفاده از مدار Reset، این دو پایه را یک بار '۱' و سپس '۰' می کنیم.

آی سی ۷۴LS۱۳۸ یک دیکدر/ مالتی پلکسر می باشد. در اینجا برای استفاده از آی سی به صورت دیکدر، پایه های E_2 و E_3 باید زمین شوند. همانطور که در شکل (۳۲-۲) نیز نشان داده شده است، خروجی شمارنده به ورودی A, B, C دیکدر متصل شده است. برای عملکرد آی سی بصورت دیکدر، پایه E_1 باید به '۱' متصل شود تا بسته به مقادیر ورودی A, B, C، یکی از خروجی ها صفر و بقیه یک باشند. همانطور که مشاهده می کنید، پایه E_1 به خروجی CLK متصل شده است. هنگامی که کلاک به صورت دستی تغییر می کند، با '۱' شده خروجی آن، دیکدر نیز فعال شده و پایه مورد نظر را در خروجی صفر می کند؛ در صورت صفر بودن E_1 ، همه پایه های خروجی در حالت '۱' منطقی خواهند بود.



شکل (۲-۳۱): مدار داخلی و شکل آی سی شمارنده چهار بیت

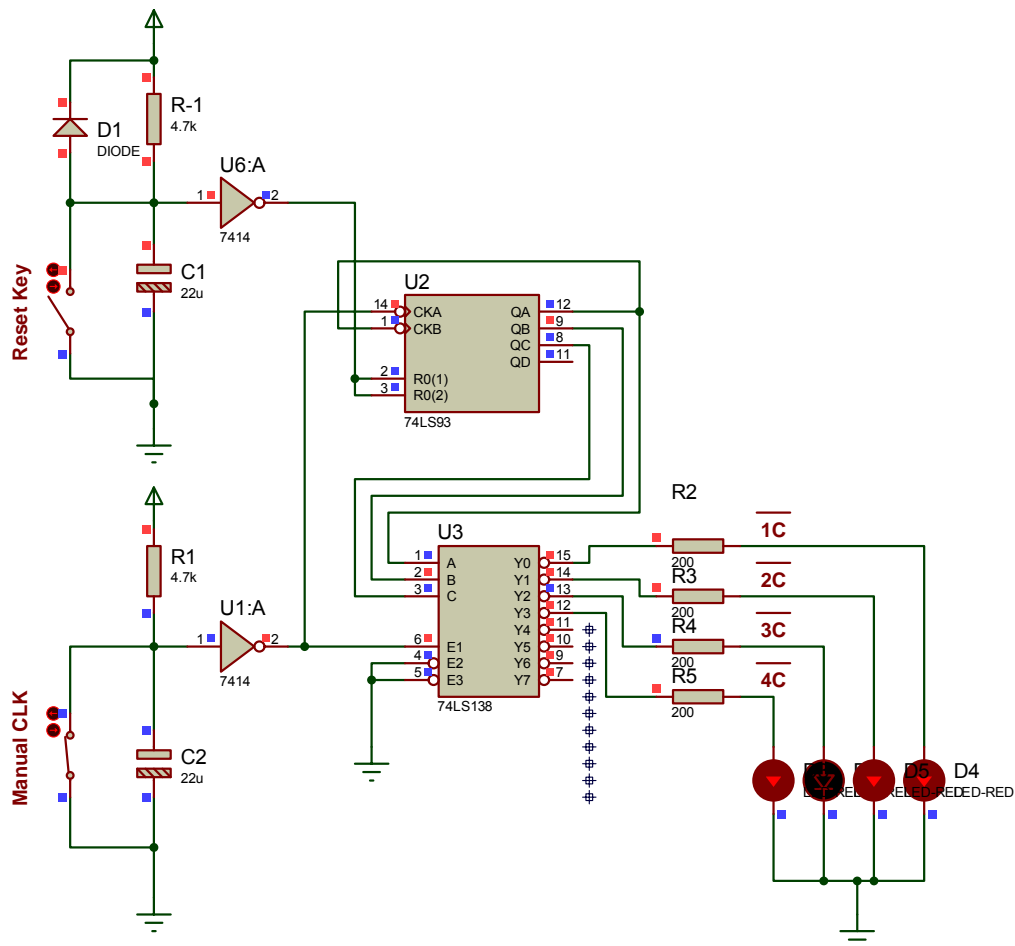
جدول (۲-۲۲). نحوه عملکرد شمارنده چهار بیت به ازای ورودی‌های مختلف

| RESET INPUTS | | OUTPUTS | | | |
|-----------------|-----------------|----------------|----------------|----------------|----------------|
| MR ₁ | MR ₂ | Q ₀ | Q ₁ | Q ₂ | Q ₃ |
| H | H | L | L | L | L |
| L | H | Count | | | |
| H | L | Count | | | |
| L | L | Count | | | |

H = HIGH Voltage Level

L = LOW Voltage Level

X = Don't Care



شکل (۲-۳۲). مدار تولید کننده دنباله‌های زمانی با استفاده از آی سی‌های ۷۴LS۱۳۸ و ۷۴LS۹۳

روش انجام آزمایش

مدار شکل (۲-۳۲) را ببندید. در ابتدا هر دو کلید CLK و Reset را در حالت باز قرار دهید. اکنون دکمه Reset را ببندید. در این لحظه می‌توانید مطمئن شوید که تمام خروجی‌های شمارنده همگی در حالت Low هستند. برای فعال شدن شمارنده، دوباره کلید Reset را باز کنید. در این لحظه چون پایه E_1 دیکدر صفر می‌باشد، بنابراین تمام خروجی‌ها در حالت High هستند و در نتیجه تمام LEDها روشن می‌باشند.

با بستن دکمه پالس دستی، LED متصل شده به خروجی Y_0 خاموش می‌گردد. با باز کردن دکمه، مجدداً روشن خواهد شد. این امر نشانه تولید اولین پالس پائین رونده در خروجی Y_0 می‌باشد ($\overline{1C}$). با فشار دادن مجدد این دکمه، خروجیهای Y_1 تا Y_7 به ترتیب با بستن کلید خاموش و با باز کردن دکمه، روشن می‌شوند. به این ترتیب تمام هشت سیکل ساعت مورد نظر تولید می‌شوند.

فصل سوم

آشنایی با نرم افزار شبیه سازی و تست

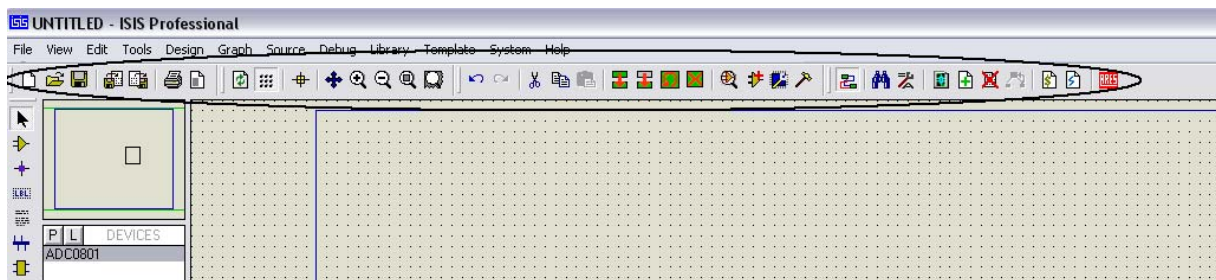
Proteus

مقدمه

نرم افزار Proteus یک شبیه ساز مخصوص ابزارهای الکترونیک است و قادر به شبیه سازی میکروکنترلرها، مدارهای مجتمع و مبدلها، مدارات آنالوگ، دیجیتال و ... می باشد و به دلیل قدرت بالا و سادگی کار با آن، دارای محبوبیت خاصی در شبیه سازی و پیاده سازی مدارات می باشد. در این بخش سعی شده است تا اصول اولیه برای پیاده سازی مدارات دیجیتال شرح داده شود. ابتدا به توضیح بخشهای مختلف که در صفحه نمایش وجود دارند خواهیم پرداخت.


نوار ابزار دستورات


ابزارهایی که به صورت پیش فرض در بالای صفحه نمایش واقع شده اند و مسیری میانبر برای دستیابی به دستورات موجود در منوی File، Edit می باشند را تحت عنوان نوار ابزار دستورات می شناسیم.



شکل (۳-۱). نوار ابزار دستورات در نرم افزار پروتئوس

در این نوار ابزار، آیکن هایی جهت دسترسی سریع و آسان به مواردی فراهم شده است که معمولاً زیاد بکار برده می شود. تمامی این موارد در داخل منوهای نرم افزار نیز موجود و قابل استفاده هستند. بعلا سادگی استفاده، معمولاً کاربران ترجیح می دهند از این نوار ابزار استفاده نمایند. در ادامه به توضیح بعضی از این آیکن ها خواهیم پرداخت:

new File : ایجاد یک طراحی جدید. 

Open Design : باز کردن فایل طراحی از حافظه. 



Save Design: ذخیره کردن طراحی در حافظه.



Import Section: وارد کردن قسمتی از طرح که قبلاً ذخیره شده.



Export Section: ذخیره کردن قسمتی از مدار که انتخاب شده است.



Print Design: چاپ کردن مدار طراحی شده.



Mark Output Area: انتخاب قسمتی که قصد چاپ آن را داریم.



Refresh: تازه کردن کردن شماتیک.



Toggle Grid: چنانچه به صفحه نگاه کنید ، نقطه چین است. با استفاده از این دکمه می توان نمایش نقطه چین ها را فعال

یا غیر فعال کرد.



Toggle False Origin: برای مشخص کردن مبدا جدید از این کلید استفاده می شود. به این صورت که ابتدا کلید را فشار

می دهیم و سپس با کلیک چپ در نقطه دلخواهی از صفحه مبدا جدید را تعیین می کنیم.



Center At Cursor: با کلیک روی این دکمه و کلیک روی قسمت دلخواهی از صفحه می توانیم روی قسمت مورد نظر

متمرکز شویم.



Zoom In: بزرگ کردن شماتیک.



Zoom Out: کوچک کردن شماتیک.



Zoom To View Entire Sheet: چنانچه قبلاً از Zoom استفاده کرده باشیم ، با کلیک روی این دکمه تمام صفحه نشان داده

می شود.



Zoom To Area: برای Zoom کردن روی قسمت خاصی از این کلید استفاده می شود.



Undo Changes: برای برگرداندن طرح به نقشه یک مرحله قبل.



Redo Changes: برای تغییر طرح به نقشه یک مرحله جلوتر.



Cut To Clipboard: می توان با این کلید قسمتی از مدار را برید.



Copy To Clipboard: می توان با این کلید قسمتی از مدار را کپی کرد.



Paste From Clipboard: میتوان با این کلید قسمتی از مدار را که قبلاً بریده یا کپی شده را جاگذاری کرد.



Block Copy: این کلید قسمتی از مدار را که انتخاب شده کپی کرده و در اختیار ما می گذارد تا آن را به هر تعداد که مورد

نیاز است، جاگذاری کنیم.



Block Move: این کلید قسمتی از مدار را که انتخاب شده بریده و در اختیار ما قرار می دهد تا در قسمت دلخواه

جاگذاری کنیم.



Block Rotate: اگر بخواهیم قسمتی از مدار را بچرخانیم، آن قسمت را انتخاب کرده و این کلید را فشار می دهیم. پنجره

جدیدی باز می شود که درجه چرخش را از ما سوال می کند.



Block Delete: با این کلید می توان قسمتی از مدار را که انتخاب کردیم پاک کرد.



Pick Parts From Libraries: برای انتخاب قطعه از کتابخانه استفاده می شود.



Make Device: مجموعه ای از قطعات انتخاب شده را به صورت یک بسته در آورده و به کتابخانه اضافه می کند.



Packaging Tool: می توان با استفاده از این کلید پکیج های قطعات موجود را تغییر داد.



Decompose: با فشار دادن این کلید می توانیم در شکل قطعاتی که انتخاب شده تغییراتی ایجاد کنیم.

Toggle Wire Auto router: چنانچه فعال باشد می توان به صورت خودکار سیم کشی کرد.



Search Tag Components: برای جستجو در شماتیک کشیده شده از این کلید استفاده می شود.



Design Explorer: با زدن این کلید پنجره Design Explorer باز می شود.



New "Root" Sheet: برای ایجاد یک شیت جدید از این کلید استفاده می شود.



Remove/Delete Sheet: برای پاک کردن یک شیت کاربرد دارد.



Exit To Parent Sheet: با استفاده از این کلید به شیت اصلی می رویم.



View Bom Report: لیستی از المان ها و توضیحات مدار را نشان می دهد.



View Electrical Report: گزارش الکتریکی مدار را نشان می دهد.



Netlist Transfer To Ares: این کلید شماتیک طراحی شده را به قسمت طراحی PCB می برد.



نوار ابزار انتخاب حالت^{۲۰}

این نوار ابزار دستورات در گوشه چپ صفحه نمایش قرار دارد و عملیات انجام شده بر روی پنجره ویرایش را کنترل

می نماید.



شکل (۲-۳). نوار ابزار انتخاب حالت در نرم افزار پروتوس

(Selection Mode) برای انتخاب المان ها این کلید باید فعال باشد.



^{۲۰} Mode Selector Toolbars



(Component Mode) برای اضافه کردن المان به مدار این کلید باید فعال باشد.



(Junction Dot Mode) برای ایجاد انشعاب در سیم کشی می توان از این کلید هم استفاده کرد.



(Wire Label Mode) برای برچسب گذاشتن روی سیم ها از کلید استفاده می شود.



(Text Scrip Mode) جهت نوشتن متن داخل صفحه می توان از این کلید استفاده کرد.



(Buses Mode) برای کشیدن باس باید این کلید فعال باشد.



(Sub-circuit Mode) جهت استفاده از المان هایی مانند تغذیه و زمین از این کلید استفاده می شود.



(Terminals Mode) برای استفاده از ترمینال های مختلف از این کلید استفاده می شود.



(Device Pins Mode) برای استفاده از پین های مختلف از این کلید استفاده می شود.



(Graph Mode) برای استفاده از گراف ها جهت رسم سیگنال های دلخواه مدار از این کلید استفاده می شود.



(Tape Recorder Mode) برای ضبط کردن یا پخش کردن سیگنال درون مدار از این کلید استفاده می شود.



(Generator Mode) برای استفاده از سیگنال ژنراتور های مختلف از این کلید استفاده می شود.



(Voltage Probe Mode) پروب ولتاژ.



(Current Probe Mode) پروب جریان.



(Virtual Instruments Mode) برای استفاده از تجهیزات مجازی همچون اسکوپ ، سیگنال ژنراتر ، آمپر متر ، ولت متر و

غیره باید این کلید فعال باشد.



(2D Graphics Line Mode) برای کشیدن خط از این کلید استفاده می شود.



(2D Graphics box Mode) برای کشیدن مستطیل از این کلید می توان استفاده کرد.



(2D Graphics Circle Mode) برای رسم یک دایره از این کلید استفاده می کنیم.



(2D Graphics Arc Mode) جهت کشیدن کمان دلخواه از این کلید استفاده می کنیم.



(2D Graphics Closed Path Mode) برای ایجاد یک شکل بسته دلخواه از این کلید استفاده می شود.



(2D Graphics Text Mode) برای نوشتن متن در جای دلخواه صفحه می توان از این کلید استفاده کرد.



(2D Graphics Symbols Mode) برای گذاشتن نماد ها از کتابخانه مربوط باید این کلید فعال باشد.



(2D Graphics Markers Mode) می توان به این وسیله یک علامت در جای دلخواه صفحه گذاشت.



(Rotate Clock-Wise) قطعه را ۹۰ درجه در جهت ساعتگرد می چرخاند.



(Rotate Anti-Clock-Wise) قطعه را ۹۰ درجه در جهت پاد ساعتگرد می چرخاند.



(X-Mirror) قطعه را نسبت به محور X می چرخاند.



(Y-Mirror) قطعه را نسبت به محور Y می چرخاند.

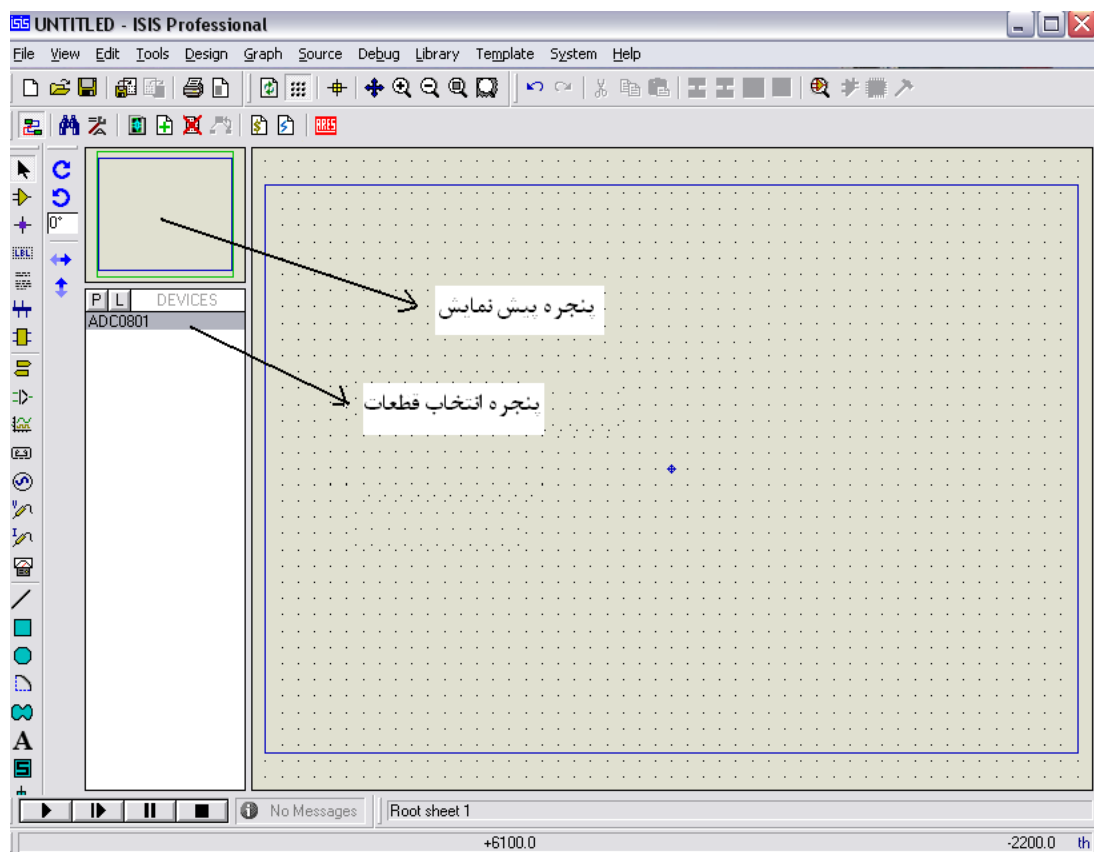


(Editing BOX) این نوار ابزار به شما این امکان را می دهد که زاویه دلخواه را جهت چرخش اعمال نمایید ، اما باید به

خاطر داشته باشید که ISIS فقط زوایای متعامد را می پذیرد.

پنجره اصلی یا پنجره ویرایش^{۲۱}

زمانی که ISIS را باز می کنید ، پنجره اصلی برنامه باز می شود که این پنجره از قسمت های مختلفی از جمله نوار ابزار ها ، پنجره ویرایش ، پنجره پیش نمایش و Object Selector تشکیل یافته است. پنجره Editing قسمت عمده این پنجره را به خود اختصاص می دهد. این پنجره امکان طراحی، ویرایش و شبیه سازی انواع مدارهای آنالوگ و دیجیتال را در اختیار کاربر قرار می دهد. این پنجره با یک Outline (به صورت پیش فرض آبی رنگ) مرزبندی شده است و اگر قطعه ای خارج از این Outline قرار گیرد غیر فعال خواهد بود و کابر دیگر، نمی تواند قطعه مزبور را انتخاب و ویرایش نماید.



شکل (۳-۳). صفحه اصلی به همراه نوار ابزارها در نرم افزار پروتئوس

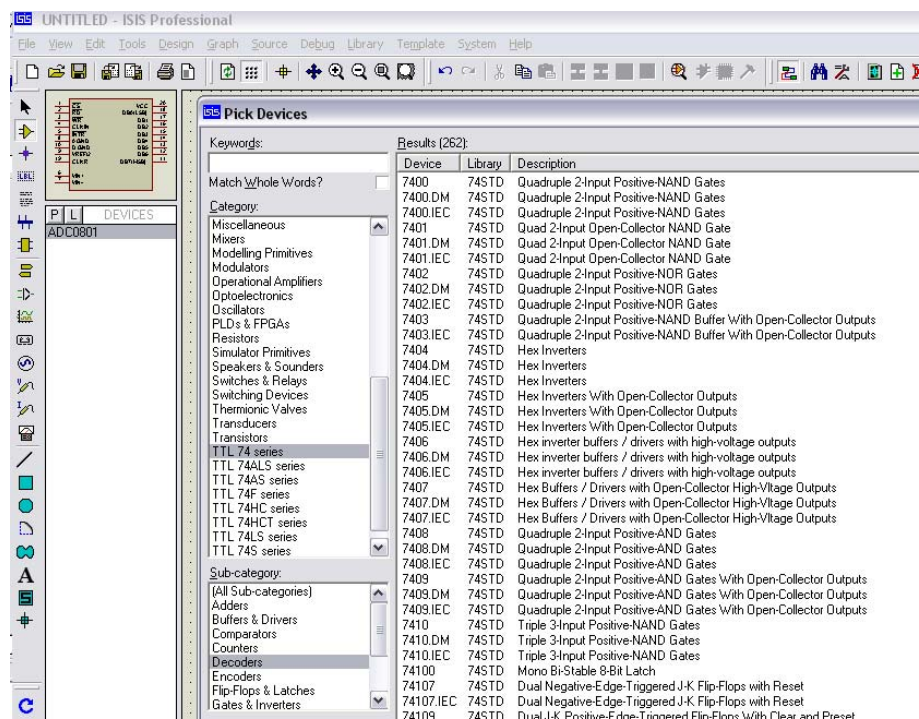
^{۲۱} The Editing Window

پنجره پیش نمایش

این پنجره یک نمایش خلاصه از تمام طرح را نشان می‌دهد. در این پنجره، رنگ آبی مایل به سبز حاشیه صفحه را مشخص می‌کند و حاشیه قرمز رنگ ناحیه ای از طرح جاری که در پنجره ویرایش قابل مشاهده است را نشان می‌دهد. بیشتر اوقات پنجره پیش نمایش برای پیش نمایش المان انتخابی برای جایگذاری استفاده می‌گردد. می‌توانید قبل از جایگذاری المان، جهت دلخواه آن را با جهت مشاهده شده روی پنجره پیش نمایش مقایسه کرده و با استفاده از دستورات mirror و orientation جهت را به جهت مطلوب اصلاح نمایید. پنجره پیش نمایش پس از جایگذاری المان، بطور اتوماتیک پاک خواهد شد.

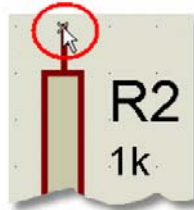
پنجره انتخاب قطعات

برای انتخاب قطعه ای باید آن قطعه را از کتابخانه فرا خوانی نمایید که این کار را از طریق بخش Devices از بخش انتخابگر قطعات می‌توان انجام داد. قطعات با توجه به طبقه بندی قطعات الکترونیک بصورت تفکیک شده موجود است.



شکل (۳-۴). پنجره انتخاب قطعات در نرم‌افزار پروتئوس

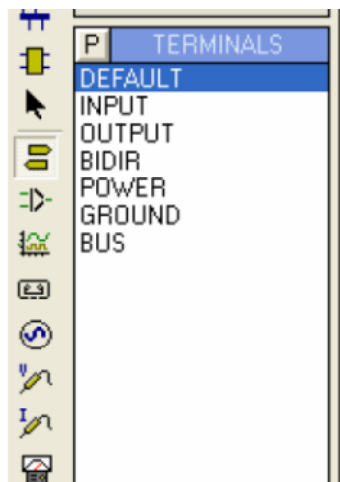
برای برقرار کردن اتصالات کافی است تا نوک موس خود را در ابتدا یا انتهای Pin های هر المان قرار دهید. علامت ضربدر ایجاد شده در نوک موس، نشان از قرار گرفتن موس بر روی Pin ورودی یا خروجی المان دارد. در صورتی که قصد انصراف از برقراری اتصال یا Wiring را دارید، کلیک راست کرده و یا دکمه Esc را فشار دهید.



شکل (۳-۵). نحوه نمایش المان مقاومت در نرم افزار پروتئوس در هنگام برقراری اتصال

نکته دیگری که در رسم مدار به آن نیاز خواهید داشت، زمین ها و Vcc ها می باشند. برای انتخاب آنها می توانید از گزینه

Inter-Sheet Terminal استفاده کنید.



شکل (۳-۶). نحوه تعیین زمین و VCC در نرم افزار پروتئوس

برای تغییر دادن مقادیر مقاومتها، خازنها و یا منابع ولتاژ کافی است بر روی عنصر مورد نظر دوبار کلیک کرده و در صفحه مربوط به مشخصات آن عنصر، مقدار مورد نظر را وارد کنید.



شکل (۷-۳). نحوه تغییر مقادیر المان در نرم افزار پروتئوس

اجرا و شبیه سازی

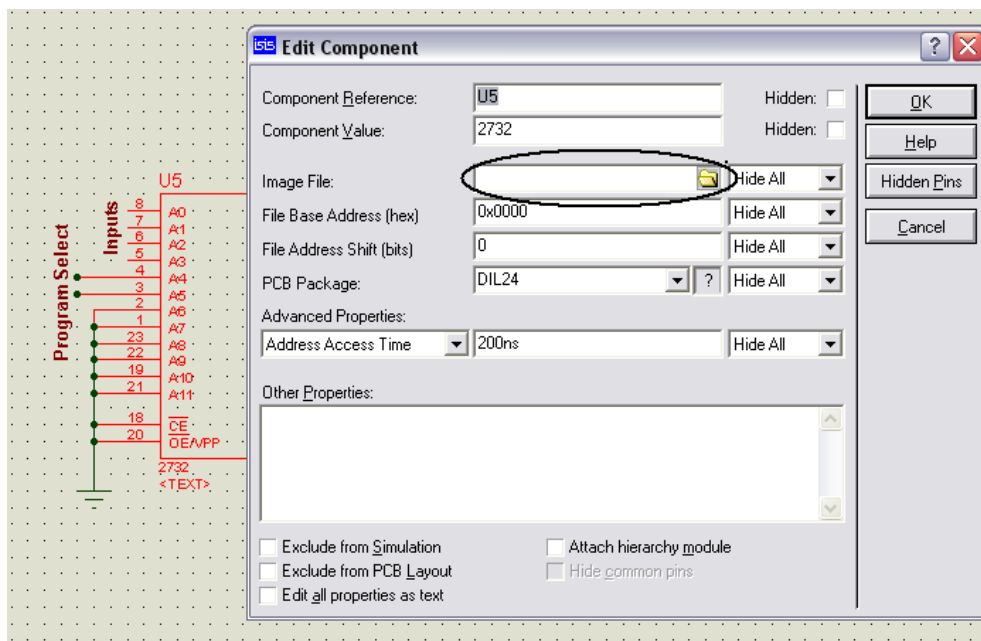
بعد از طراحی مدار مورد نظر و دیدن نتیجه از بخش پایین صفحه اصلی که در شکل (۳-۳) نمایش داده شده است، برنامه را اجرا و کنترل نمایید.



شکل (۸-۳). دکمه‌های موجود در نرم افزار پروتئوس جهت اجرا و توقف

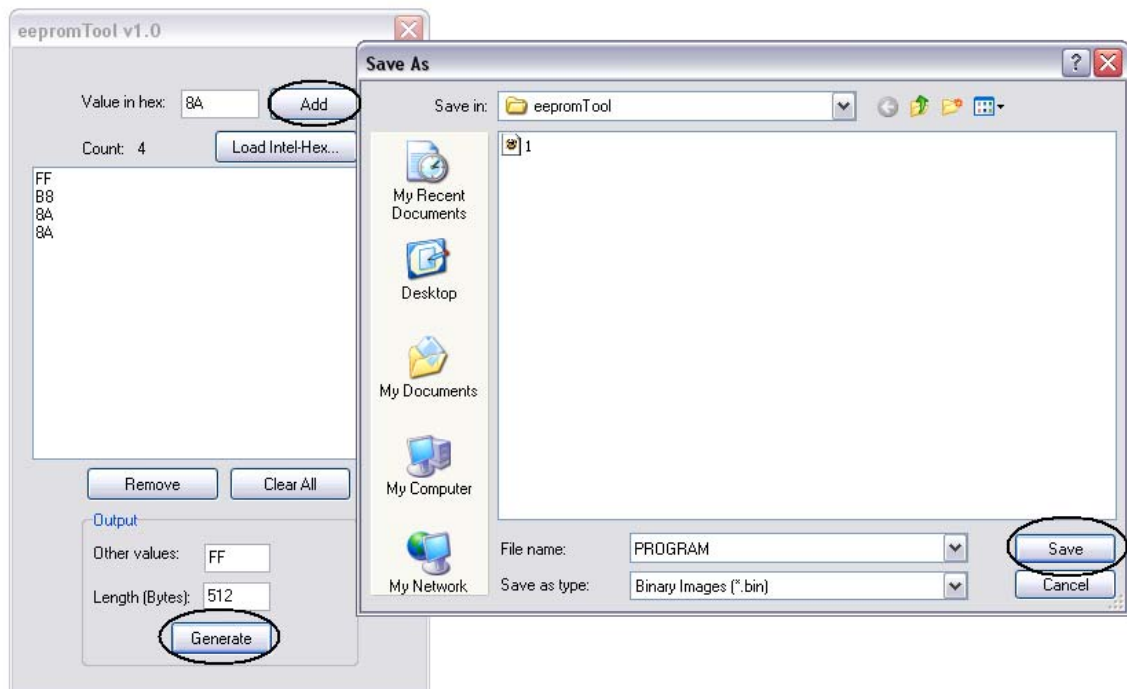
برنامه ریزی حافظه EPROM

در طراحی میکروپروسسور از آی سی ۷۴۳۲ به عنوان حافظه و برای ذخیره کردن برنامه ها استفاده شده است. بنابراین برای شبیه سازی این آی سی نیاز به برنامه ریزی آن دارید. برای انجام این کار در Proteus، در قسمت مشخصات مربوط به هر میکروکنترلر و یا عنصر حافظه ای، گزینه‌ای وجود دارد که با انتخاب آن می‌توانید یک فایل را به عنوان محتویات اولیه حافظه EPROM میکروکنترلر به Proteus معرفی نمایید. در شکل زیر می‌توانید این گزینه را برای آی سی ۷۴۳۲ مشاهده کنید.



شکل (۳-۹). نحوه تعریف یک المان موجود در حافظه در نرم افزار پروتئوس

با استفاده از برنامه EEPROM-TOOL می‌توانید فایل های hex خود را به فرمت باینری قابل استفاده در Proteus تبدیل نمایید. فرض کنید می‌خواهیم چهار بایت از حافظه با مقادیر نشان داده شده در شکل ۰۰ برنامه ریزی شوند. توجه کنید از طریق دکمه‌های Add و Remove می‌توانید بایت‌هایی را اضافه کرده و یا حذف کنید. در پایان اندازه کل حافظه EEPROM-TOOL را در قسمت Length وارد کرده (به بایت) و در قسمت Other values نیز مقداری را که می‌خواهید به عنوان مقدار پیش فرض قسمت‌های اشغال نشده EEPROM-TOOL در آن ذخیره شود را به فرمت عدد مبنای شانزده وارد نماید. دکمه Generate را فشار داده و داده‌ها را در فایل مورد نظر ذخیره نمایید. اکنون می‌توانید این فایل را در Proteus بصورتی که قبلاً توضیح داده شد، انتخاب نموده و پروژه خود را شبیه‌سازی نمایید.

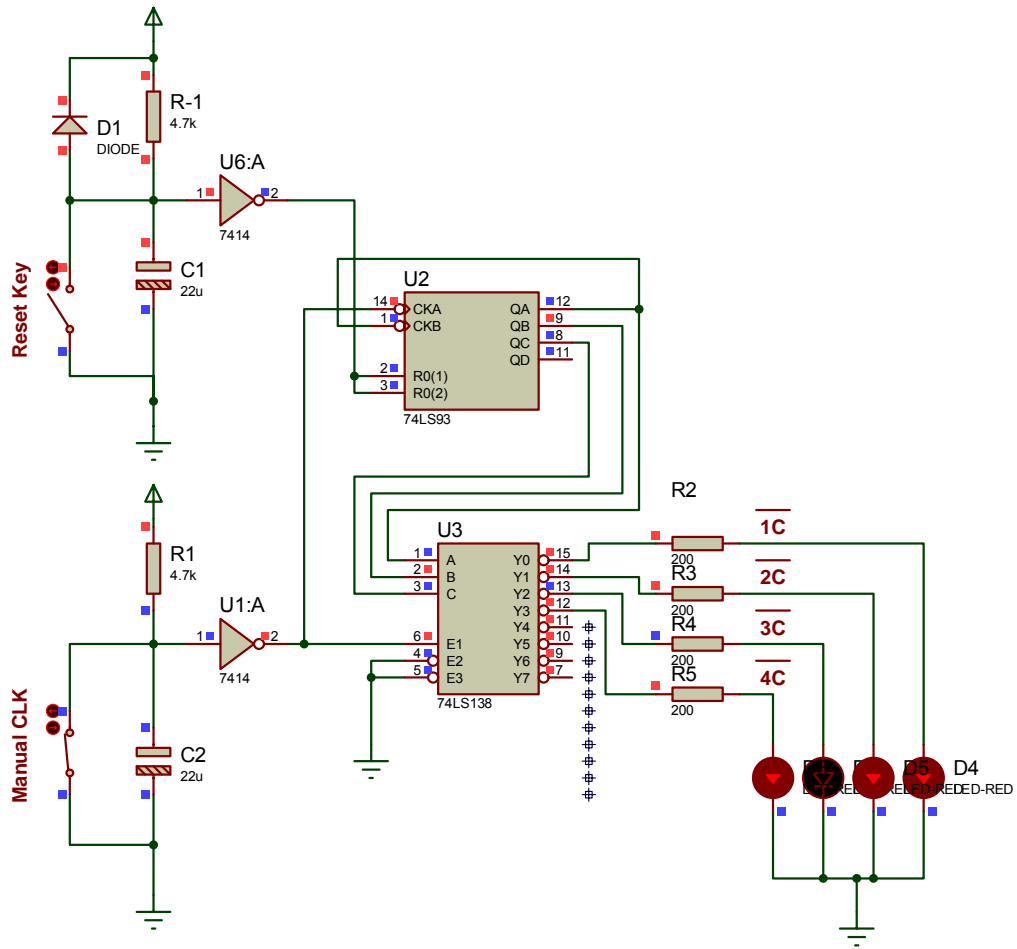


شکل (۳-۱۰). نحوه تعریف مقادیر حافظه و ذخیره آن در نرم افزار پروتئوس

پیاده سازی یک مدار نمونه

به عنوان مثال، مدار مولد سیگنال سیکل ماشین که در آزمایش ۸ از فصل اول مورد استفاده قرار گرفته است، در شکل زیر

نشان داده شده است.



شکل (۳-۱۱). نحوه تعریف و ترسیم مدار مربوط به آزمایش ۸ در نرم افزار پروتئوس

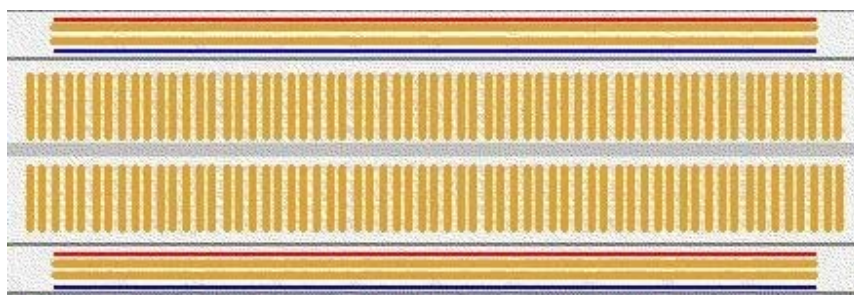
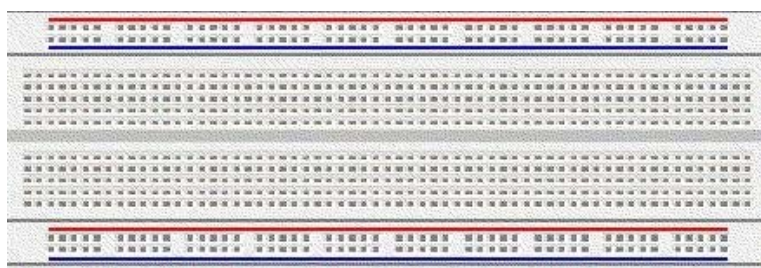
فصل چهارم

آشنایی با برخی از تجهیزات و آی‌سی‌های مورد

استفاده در آزمایشگاه

برد برد

برد برد^{۲۲} از تعداد زیادی سیم ردیف شده‌ی فلزی (معمولاً مسی) ساخته شده است، که در زیر برد ردیف شده‌اند. ردیف‌های فلزی همانند شکل زیر با هم موازی هستند. این ردیف‌ها سوراخ‌های روی برد را به هم متصل می‌نمایند که باعث آسان‌تر شدن اتصال قطعات برای ساخت مدارها می‌شود. برای استفاده از بردبرد پایه‌های قطعات را در حفره‌های آن فرو کنید. هر حفره به یکی از ردیف‌های زیر برد متصل شده است.

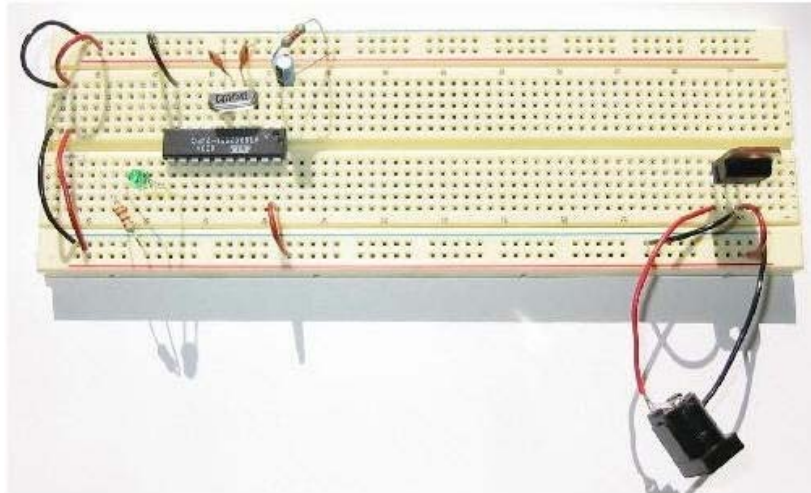


شکل (۴-۱). تصویر بردبرد و نحوه اتصالات موجود در آن

هر سیم تشکیل یک گره می‌دهد، گره نقطه‌ای از مدار است که دو قطعه به هم اتصال می‌یابند. اتصال بین قطعات مختلف با قرار دادن پایه‌ی آن‌ها در گره مشترک تشکیل می‌شود. روی برد برد، گره سطری از حفره‌هایی است که توسط ردیف فلزی اتصال یافته است. سطرهای بالا و پایین معمولاً برای اتصال منابع تغذیه استفاده می‌شود. باقی مدار توسط قرار دادن قطعات و اتصال آنها با یکدیگر توسط سیم‌ها تشکیل می‌شود. چپ‌هایی که پایه‌های زیادی دارند (آی‌سی‌ها) را در وسط برد قرار دهید و نیمی از پایه‌ها را روی یک طرف و نیمی دیگر را روی طرف دیگر قرار دهید. برای درک بهتر مطلب به شکل زیر توجه

^{۲۲} Bread Board

نمایید. سوراخهای قرار گرفته در نیمه بالایی برد بطور عمودی بهم متصل هستند و می توان پایه های عناصر مختلفی که باید بهم متصل باشند را در سوراخهای عمودی قرار داد و مدار را بست.

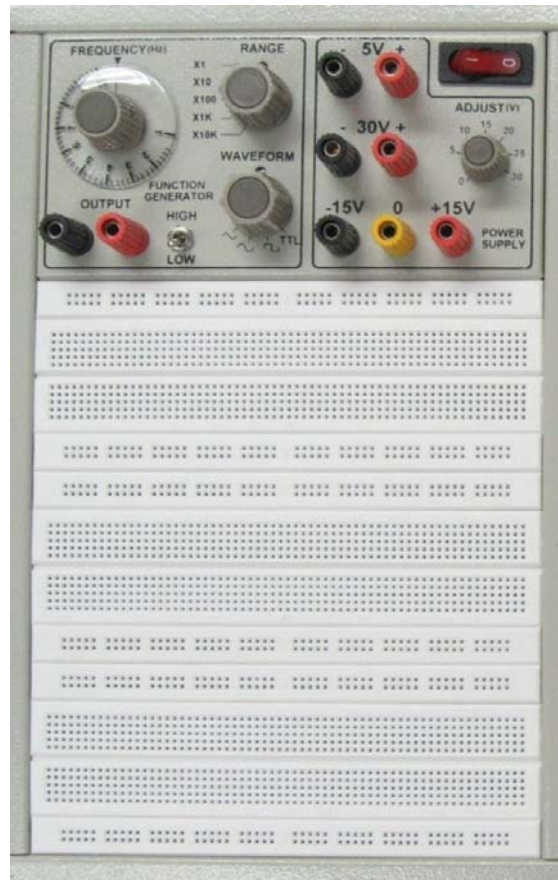


شکل (۴-۲). یک روش معمول در نصب آی سی ها و ارتباطات آنها

در برد دیجیتال موجود در آزمایشگاه معمولاً سه برد مورد موجود است که دانشجویان می توانند در صورت نیاز برای بستن مدارها یا تست قطعات از آنها استفاده نمایند. در شکل زیر این بردها قابل مشاهده هستند.

برد آموزشی دیجیتال

بردهای آموزشی موجود در آزمایشگاههای دیجیتال دانشگاه پیام نور در سراسر کشور مشابه و یکسان است. این بردها دارای منابع تغذیه، مولد پالس مربعی، سینوسی و مثلثی است. بنابراین نیازی به استفاده از سیگنال ژنراتور برای تولید سیگنالهای با شکل موجها و فرکانسهای متفاوت نمی باشد. برای انجام آزمایشات این درس، می توان از برد بردها برای نصب آی سی ها، مقاومتها، خازنها و سایر المانهای الکتریکی استفاده کرد. در شکل (۴-۳) نحوه قراگیری بردبردها، منابع تغذیه و مولد موجهای مربعی، سینوسی و دندانه اره ای مشاهده می شود.



شکل (۴-۳). نحوه قرار گرفتن بردبوردها، منابع تغذیه و مولد موج در بردهای آزمایشگاه

در بردهای موجود در آزمایشگاه، تمامی اتصالات از زیر برقرار شده و دانشجو نیازی به بستن کل مدار و همه سیم‌کشی‌ها ندارد. تنها آی‌سی‌های مورد نیاز و سرهای اتصال ورودی و خروجی آنها بر روی برد و در اختیار دانشجوین قرار گرفته‌اند. این برد دارای امکانات دیگری همچون برد بورد، نمایشگر هفت قطعه‌ای، سوئیچ‌ها و تعدادی LED می‌باشد که انجام آزمایشها را سریعتر و مفهوم‌تر خواهد کرد. جهت اطلاع بیشتر، در شکل (۴-۴) تمامی قسمتهای این برد قابل مشاهده هستند.



شکل (۴-۴). تصویری از کل برد موجود در آزمایشگاه

در شکل (۴-۴) تصویری از کل برد قابل مشاهده است که سمت چپ آن بردبرد و سمت راست گیت‌های مختلف منطقی هستند. تمامی آزمایشهای فصل دوم توسط این برد و گیت‌های موجود بر آن قابل پیاده‌سازی هستند. برای کار با آی‌سی‌هایی که در مجموعه گیت‌های سمت راست موجود نیستند، می‌توان آنها را بر روی برد برد سمت چپ نصب و استفاده کرد.

در ادامه این فصل مشخصات آی‌سی‌های بکار رفته در آزمایشها بطور مختصر آورده شده‌اند.

آی سی ۷۴۱۴ :

این آی سی دارای شش عدد گیت معکوس کننده اشmitt تریگر است که ترتیب قرار گرفتن آنها در آی سی در شکل (۴-۵) نشان داده شده است.

Description:
This package contains six Schmitt-trigger inverters.

Mode of operation:
All six inverters can be used independently of one another.
For each inverter, if an input is low, the output will be high and vice versa.
The internal hysteresis makes the device ideal for noisy or slowly changing input signals.
In addition, it is also suitable for debouncing or conditioning inputs from switch contacts and for astable or monostable circuits.
If the input voltage is rising in a positive direction, the output will change at 1.7 V. If the input voltage is moving in the negative direction, the change takes place at 0.9 V. For this reason, the hysteresis, or the "dead band" is 0.8 V. This is temperature-compensated internally.
The pinout is compatible with the 7404.

Application:
Level detection and pulse shaping, use in systems with noisy signals, monostable and astable multivibrators, implementation of inverter functions.

| | | | | | | | | |
|-------------------|----|-----|-----|--|-----|--|--|-----|
| Data: | | | | | | | | |
| Propagation delay | ns | 15 | 8 | | 5 | | | 15 |
| Hysteresis | V | 0.8 | 0.8 | | 0.8 | | | 0.8 |
| Supply current | mA | 30 | 12 | | 18 | | | 10 |

| | | | | | | | | |
|------------------|-----|-----|----|---|---|---|----|---|
| Families: | Std | ALS | AS | F | H | L | LS | S |
| | ● | ● | | ● | | | ● | |

Hex Schmitt-trigger INVERTER

7414

آی سی ۷۴۷۴:

این آی سی دارای دو فلیپ فلاپ نوع D است که دارای پایه های clear و preset نیز می باشد. در جدولی که در پایین دیده می شود، رفتار این فلیپ فلاپ به ازای ورودی های مختلف قابل مشاهده است.

Description:
This package contains two D-type positive-edge-triggered flip-flops with separate preset and clear inputs.

Mode of operation:
Each flip-flop can be used independently.
Data at the D-input is transferred to output Q (and inverted to output \bar{Q}) whenever the clock input changes from low to high. Until this rising clock edge occurs, no change of data at the D-input will result in a change in the output.
If D is high, Q goes high and \bar{Q} goes low when triggered by the clock pulse.
If D is low, Q goes low and \bar{Q} goes high when triggered by the clock pulse.
Data on the D-input can be changed at any time. What counts is the level at the moment that the clock goes from low to high. This state is then latched.
For normal operation, the Preset and Reset inputs should be held high. If the Reset input is taken low, \bar{Q} goes low and Q goes high. If the Preset input is taken low, Q immediately goes high and \bar{Q} goes low. These two inputs should never be low simultaneously, since an unstable state is created which is not retained once Preset and Reset become inactive (high).

| Inputs | | | | Outputs | |
|--------|-------|-------|---|-----------|-----------|
| Preset | Reset | Clock | D | Q | \bar{Q} |
| L | H | X | X | H | L |
| H | L | X | X | L | H |
| L | L | X | X | H* | H* |
| H | H | ✓ | H | H | L |
| H | H | ✓ | L | L | H |
| H | H | L | X | no change | no change |
| H | H | H | X | no change | no change |
| H | H | ✓ | X | no change | no change |

* unstable state

Application:
Registers, counters, control circuits

| Data: | | 15 | 34 | 105 | 100 | 35 | 2.5 | 25 | 75 |
|------------------------------------|-----|----|-----|-----|-----|----|-----|----|----|
| Minimum guaranteed clock frequency | MHz | 15 | 34 | 105 | 100 | 35 | 2.5 | 25 | 75 |
| Propagation delay | ns | 17 | 10 | 6 | 4 | 13 | 65 | 19 | 6 |
| Supply current | mA | 17 | 1.2 | 5 | 10 | 30 | 1.6 | 4 | 30 |

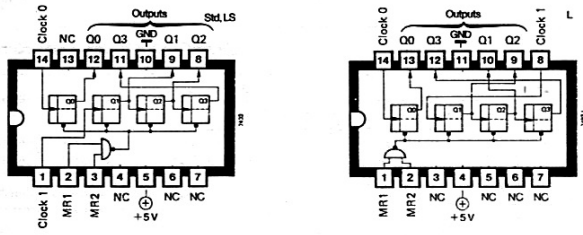
| Families: | Std | ALS | AS | F | H | L | LS | S |
|-----------|-----|-----|----|---|---|---|----|---|
| | ● | ● | ● | ● | ● | ● | ● | ● |

Dual D-TYPE FLIP-FLOP with preset and clear

7474

آی سی ۷۴۹۳ :

این آی سی یک شمارنده چهاربیتی باینری است به ازای هر پالس ورودی، محتویات آن یک شماره تغییر می کند. دارای دو ورودی ریست است و تنها در حالتی که هر دوی آنها در وضعیت High قرار داشته باشند، شمارنده ریست شده و مقدارش صفر خواهد شد. مشخصات بیشتر این شمارنده در شکل زیر قابل مشاهده است.



Description:
This package contains a divide-by-two and a divide-by-eight counter.

Mode of operation:
The device consists of 4 flip-flops which are connected internally to create one divide-by-two and one divide-by-eight counter.
All flip-flops have a common reset line controlled by two inputs MR1 and MR2. When both MR1 and MR2 are high the counters are cleared.
Flip-flop 1 is not internally connected to the other stages, thus providing a variety of counter sequences:

- Counting to 16: Output Q0 is connected to the Clock 1 input. The input pulses are applied to the Clock 0 input and the divided symmetrical output signal is extracted at Q3. The device counts in binary code up to 16 (0-15), with the outputs returning to zero on the 16th clock pulse.
- Divide by 2 and divide by 8: Flip-flop 1 is used as a 2:1 divider and flip-flops 2,3 and 4 are used as an 8:1 divider. The flip-flops are triggered on the negative-edge of the clock pulse. For normal counting, at least one of the two reset pins MR1 or MR2 must be taken low.

| Reset inputs | | Outputs | | | |
|--------------|-----|----------|----|----|----|
| MR1 | MR2 | Q0 | Q1 | Q2 | Q3 |
| H | H | L | L | L | L |
| L | H | Counting | | | |
| H | L | Counting | | | |
| L | L | Counting | | | |

Application:
Counters and dividers 2:1, 8:1 and 16:1

| Data: | | | | | | | | | |
|---|-----|-----|-----|----|---|-----|----|----|---|
| Min. guaranteed clock frequency, flip-flop 1 | MHz | 32 | | | | 3 | 32 | | |
| Min. guaranteed clock frequency, flip-flops 2-4 | MHz | 16 | | | | 3 | 16 | | |
| Supply Current | mA | 26 | | | | 3.2 | 9 | | |
| Families: | | Std | ALS | AS | F | H | L | LS | S |
| | | ● | | | | | ● | ● | |

4-bit BINARY COUNTER

7493

آی سی ۷۴۱۳۸ :

این آی سی یک دیکودر ۳ به ۸ است. دارای سه ورودی و هشت خروجی است. هر خروجی تنها به ازای یک ترکیب مشخص ورودی فعال و بقیه خروجی ها غیر فعال هستند. خروجی فعال دارای منطق Low بوده و خروجی های غیرفعال منطق High دارند. آی سی سه ورودی فعال ساز دارد که تنها با مقداردهی مناسب آنها آی سی فعال شده و بطور صحیح کار می کند در غیر اینصورت همه خروجی ها High و غیرفعال هستند.

Description:
This package contains a high-speed 3 line to 8 line decoder/demultiplexer with 3 enable inputs.

Mode of operation:
When a 3-bit code is applied to the three binary weighted address inputs (A0, A1 and A2), the \bar{Q} output corresponding to this code goes low, while the other outputs remain high. However, this applies only if the enable inputs E1 and E2 are low and E3 is high. The multiple enable inputs to this circuit permit the parallel expansion of the device to a 1 of 32 decoder requiring only four 74138s plus an inverter. This package can also be used as a demultiplexer with 8 outputs, in which one of the $\bar{E1}$ or $\bar{E2}$ inputs (with active low) is used as a data input and the other enable inputs as strobe inputs. The unused enable inputs in this case must be taken to their corresponding active level with a high or low.

| Enable inputs | | | Address inputs | | | Outputs | | | | | | | |
|---------------|----|----|----------------|----|----|---------|----|----|----|----|----|----|----|
| E3 | E2 | E1 | A2 | A1 | A0 | Q0 | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
| X | H | X | X | X | X | H | H | H | H | H | H | H | H |
| X | X | H | X | X | X | H | H | H | H | H | H | H | H |
| L | X | X | X | X | X | H | H | H | H | H | H | H | H |
| H | L | L | L | L | L | L | H | H | H | H | H | H | H |
| H | L | L | L | L | H | L | H | H | H | H | H | H | H |
| H | L | L | L | H | L | H | H | L | H | H | H | H | H |
| H | L | L | L | H | H | H | H | H | L | H | H | H | H |
| H | L | L | H | L | L | H | H | H | H | L | H | H | H |
| H | L | L | H | L | H | H | H | H | H | H | L | H | H |
| H | L | L | H | H | L | H | H | H | H | H | H | L | H |
| H | L | L | H | H | H | H | H | H | H | H | H | H | L |

Application:
Digital demultiplexing, address decoding, control decoding

| | | | | | | | | |
|-------------------|----|-----|-----|-----|-----|---|---|----|
| Data: | | | | | | | | |
| Propagation delay | ns | | 13 | 5.8 | 5.8 | | | 22 |
| Supply current | mA | | 6 | 5 | 13 | | | 13 |
| Families: | | Std | ALS | AS | F | H | L | LS |
| | | | ● | ● | ● | | | ● |

3-to-8 LINE DECODER/DEMULPLEXER with inverted outputs

74138

آی سی ۷۴۱۵۳ :

این آی سی دارای دو عدد مالتی پلکسر ۴ به ۱ است که برای هر کدام دارای چهار ورودی و یک خروجی است. با اعمال سیگنالهای مناسب به دو ورودی آدرس A_1 و A_0 خط داده ورودی انتخاب شده و به خروجی منتقل می شود. آی سی دارای دو ورودی فعالساز مستقل از هم می باشد که برای فعالسازی هر یک از مالتی پلکسرها از آنها استفاده می شود.

Description:
This package contains two 1-of-4 data selectors with separate strobe inputs and common address inputs

Mode of operation:
The required input (D0 - D4) for both selectors is obtained by applying a binary address to the common Address inputs (A0 - A1). The signal at the selected input appears at the corresponding Q output in non-inverted form.
For normal operation, the Enable inputs ($\overline{1G}$ $2\overline{G}$), which operate independently, are taken low. If an Enable input is taken high, the corresponding output goes low, regardless of the state of the other inputs.

| Address inputs | | Data inputs | | | | Enable | Output |
|----------------|----|-------------|----|----|----|----------------|--------|
| A1 | A0 | D0 | D1 | D2 | D3 | \overline{G} | Q |
| X | X | X | X | X | X | H | L |
| L | L | L | X | X | X | L | L |
| L | L | H | X | X | X | L | H |
| L | H | X | L | X | X | L | L |
| L | H | X | H | X | X | L | H |
| H | L | X | X | L | X | L | L |
| H | L | X | X | H | X | L | H |
| H | H | X | X | X | L | L | L |
| H | H | X | X | X | H | L | H |

Application:
Multiplexing, address decoding, serial data transmission.

| Data: | ns | 17 | 10 | 6.6 | 7 | | 17 | 19 |
|-------------------|----|----|-----|------|----|--|----|----|
| Propagation delay | ns | 36 | 7.5 | 18.5 | 12 | | 6 | 45 |
| Supply current | mA | | | | | | | |

Families:

| | Std | ALS | AS | F | H | L | LS | S |
|--|-----|-----|----|---|---|---|----|---|
| | ● | ● | ● | ● | | ● | ● | ● |

Dual 4-LINE-TO-1-LINE DATA SELECTOR/MULTIPLEXER

74153

آی سی ۷۴۱۶۵ :

این آی سی یک شیفتر رجیستر ۸ بیتی است. وظیفه اصلی آن شیفتر دادن اطلاعات موجود در آن به ازای هر کلاک پالس ورودی است. در لبه بالا رونده هر پالس ساعت، داده موجود در رجیستر یک واحد به راست جابجا می شود. این آی سی همچنین توانایی بار کردن یک داده چهاربیتی بصورت همزمان و در یک پالس ساعت را نیز داراست.

Description:
This package contains an 8-stage shift register with serial or parallel input and serial output.

Mode of operation:
For normal operation, the $\overline{\text{Enable}}$ input is held low. Each low-to-high transition (positive edge) of the clock pulse applied to the Clock input shifts the data one step to the right. The shift register can be loaded with parallel data from inputs P1 through P8, by taking the Load input momentarily low. The load operation is independent of the clock pulse. Data present at the Serial Input (pin 10) is entered into the register on each positive edge of the clock pulse. The serial output is obtained at Q8 and inverted at $\overline{\text{Q8}}$. The Clock input can be disabled by taking the Enable input high. The Clock and Enable inputs may be swapped, because they form a 2-input OR gate. The 74166 is a similar device with a Clear input.

| Shift/Load | Inputs | | Function |
|------------|--------------|----------------------------|---------------|
| | Clock | $\overline{\text{Enable}}$ | |
| L | X | X | Parallel load |
| H | H | X | no change |
| H | X | H | no change |
| H | L | \downarrow | Shift |
| H | \downarrow | L | Shift |

Application:
Parallel-serial conversion, temporary data storage.

| Data: | | 20 | 60 | | | | 25 | |
|---------------------------------|-----|----|----|--|--|--|----|--|
| Min. guaranteed clock frequency | MHz | 42 | 16 | | | | 21 | |
| Supply current | mA | | | | | | | |

| Families: | Std | ALS | AS | F | H | L | LS | S |
|-----------|-----|-----|----|---|---|---|----|---|
| | ● | ● | | | | ● | ● | |

8-BIT SHIFT REGISTER (parallel/serial-in, serial-out)

74165

آی سی ۷۴۱۷۳ :

این آی سی دارای یک فلیپ فلاپ نوع D چهار بیتی است. دارای دو ورودی فعالساز است که برای عملکرد معمول فلیپ فلاپ باید هر دو مقدار Low داشته باشند. چهار ورودی داده و چهار خروجی داده در شکل زیر قابل مشاهده هستند. رفتار فلیپ فلاپ نیز در جدول زیر قابل مشاهده است.

Description:
This package contains four D-type flip-flops with clock and clear inputs and three-state outputs.

Mode of operation:
The parallel data is applied via inputs D1 through D4. Data present at the D inputs is entered into the appropriate flip-flop on the low-to-high transition (positive edge) of the clock pulse. Both data enable inputs IE1 and IE2 must be low. If one of these enable inputs is taken high, the data already stored remains unchanged with subsequent clock pulses. The stored data is present at outputs Q1 through Q4, provided that the output enable inputs OE1 and OE2 are low. If one of these pins is taken high, the outputs are driven into a high-impedance state (three-state).

| Inputs | | | | | | Outputs Q1-Q4 |
|--------------------------|-------|-------|-------------------------|---------------------|-----------|------------------|
| Output enable OE1 OE2 | Clear | Clock | Input enable IE1 IE2 | Data input D1-D4 | | |
| L L | H | X | X X | X | L | |
| L L | L | L | X X | X | no change | |
| L L | L | H | X X | X | no change | |
| L L | L | ↗ | H X | X | no change | |
| L L | L | ↗ | X H | X | no change | |
| L L | L | ↗ | L L | L | L | |
| L L | L | ↗ | L L | H | H | |
| L L | L | ↗ | X X | X | no change | |
| L H | X | X | X X | X | Z | |
| H L | X | X | X X | X | Z | |
| H H | X | X | X X | X | Z | |

Application:
Buffer-registers.

| | | | | | | | | |
|---------------------------------|-----|------|-----|----|---|---|----|------|
| Data: | | | | | | | | |
| Min. guaranteed clock frequency | MHz | 35 | | | | | 50 | |
| Propagation delay | ns | 23.5 | | | | | 17 | |
| Supply current | mA | 50 | | | | | 19 | |
| Families: | | Std | ALS | AS | F | H | L | LS S |
| | | ● | | | ● | | ● | |

4-BIT D-TYPE REGISTER with enable and clear (three state)

74173

آی سی ۷۴۱۸۱ :

این آی سی یک ALU چهار بیتی است که توانایی دریافت دو داده چهاربیتی و انجام ۱۶ عمل محاسباتی و ۱۶ عمل منطقی را داراست. برای ساختن ALUهای بزرگتر می توان چند عدد از این آی سی را به هم متصل کرد.

Description:
This package contains an arithmetic logic unit (ALU) which can perform 16 logical and 16 arithmetic operations on two 4-bit words.

Mode of operation:
The two operands A and B are supplied to the corresponding inputs (active low). With M taken high, Logic mode is selected, with M low Arithmetic. The required function is then selected according to the truth table via inputs S0 through S3 with the result being obtained at F0 - F3 (active low).
Expansion to n x 8 bits is possible using the 74182 (carry unit) package and additional 74181s. This package can also be used as a comparator. If the operands are the same the A = B output goes high (open-collector). Negative logic operation is also possible if the pins are renamed accordingly.

| Function selection | | | Inputs and outputs active low | | Inputs and outputs active high | |
|--------------------|----------------|----------------|--|---------------|--|---------------|
| S ₃ | S ₂ | S ₁ | Arithmetic (M = L, C _n = L) | Logic (M = H) | Arithmetic (M = L, C _n = H) | Logic (M = H) |
| L | L | L | A minus 1 | A | A | A |
| H | L | L | AB minus 1 | AB | A + B | A ⊕ B |
| L | H | L | AB minus 1 | A + B | A + B | AB |
| H | H | L | minus 1 (2's comp.) | Logic '1' | minus 1 (2's comp.) | Logic '0' |
| L | L | H | A plus [A + B] | A + B | A plus AB | AB |
| H | L | H | AB plus [A + B] | B | AB plus [A + B] | B |
| L | H | H | A minus B minus 1 | A ⊕ B | A minus B minus 1 | A ⊕ B |
| H | H | H | A + B | A + B | AB minus 1 | AB |
| L | L | L | A plus [A + B] | AB | A plus AB | A + B |
| H | L | L | A plus B | A ⊕ B | A plus B | A ⊕ B |
| L | H | L | AB plus [A + B] | B | AB plus [A + B] | B |
| H | H | L | A + B | A + B | AB minus 1 | AB |
| L | L | H | A plus A (2 × A) | Logic '0' | A plus A (2 × A) | Logic '1' |
| H | L | H | A plus AB | AB | A plus [A + B] | A + B |
| L | H | H | A plus AB | AB | A plus [A + B] | A + B |
| H | H | H | A | A | A minus 1 | A |

Application:
Arithmetic and logic operations.

| Data: | Typ. | ns | 24 | 5 | 7 | 24 | 11 |
|-------------------------------|------|----|-----|----|----|-----|----|
| Typ. addition time for 4 bits | ns | 24 | 5 | 7 | 24 | 11 | |
| Supply current | mA | 91 | 135 | 43 | 20 | 120 | |

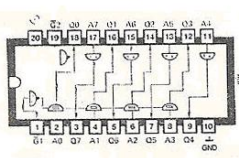
| Families: | Std | ALS | AS | F | H | L | LS | S |
|-----------|-----|-----|----|---|---|---|----|---|
| | ● | | ● | ● | | | ● | ● |

4-BIT ARITHMETIC LOGIC UNIT/FUNCTION GENERATOR

74181

آی سی ۷۴۲۴۴ :

این آی سی دارای دو بافر مستقل چهاربیتی است که می توان برای انتقال ۸ بیت داده نیز از آن استفاده کرد. هر مجموعه ۴ بیتی دارای یک فعالساز مستقل است که در حالت Low فعال می شوند. ۷۴۲۴۴ بافر سه حالتی است یعنی در حالتی که فعالساز غیر فعال است، خروجی بافر امپدانس بالاست. در غیر اینصورت مقدار خروجی برابر با مقدار ورودی است. بافرها معمولاً برای محافظت مدارها و جلوگیری از تداخل سیگنال دو طرف بافر بکار می روند.



Description:
This package comprises eight non-inverting line drivers/buffers, with 3-state outputs, organised as two blocks of four each with its own enable input.

Operation:
The line drivers are arranged in two groups. Group 1 (A0-A3) is controlled by the enable input $\overline{G1}$. Data applied to inputs A0-A3 appears at outputs Q0-Q3 in non-inverted form. Group 2 (A4-A7) is controlled by the enable input $\overline{G2}$. Data applied to inputs A4-A7 appears at outputs Q4-Q7 in non-inverted form.
If $\overline{G1}$ or $\overline{G2}$ are taken high, the associated outputs are driven to their high-impedance state (Z). The line drivers may drive bus lines terminated with at least 133 Ω .
All inputs have PNP inputs stages to reduce the DC loading on the bus. All inputs on the LS and S versions have Schmitt-trigger action which makes this device ideal for receiving signals on noisy lines.
The outputs can source 15 mA when high and LS & ALS types can sink 24 mA when low (64 mA for S & AS types).

| Inputs | | Output |
|----------------|---|--------|
| \overline{G} | A | Q |
| L | H | H |
| L | L | L |
| H | X | Z |

Application:
Buffers and line drivers for data and address bus systems

| | | | | | | | | |
|---|-----|-----|-----|----|---|---|----|-----|
| Data: Propagation delay Supply current | ns | 6.5 | 4.1 | 4 | | | 12 | 6 |
| | mA | 16 | 39 | 53 | | | 27 | 112 |
| Families: | Std | ALS | AS | F | H | L | LS | S |
| | | ● | ● | ● | | | ● | ● |

Octal non-inverting BUFFER/LINE DRIVER (3-state)

74244

مراجع:

- [۱] موریس مانو، "معماری کامپیوتر"، انتشارات خراسان، ۱۳۸۵.
- [۲] موریس مانو، "مدارهای دیجیتال"، انتشارات دانشگاه صنعتی امیرکبیر، ۱۳۸۷.