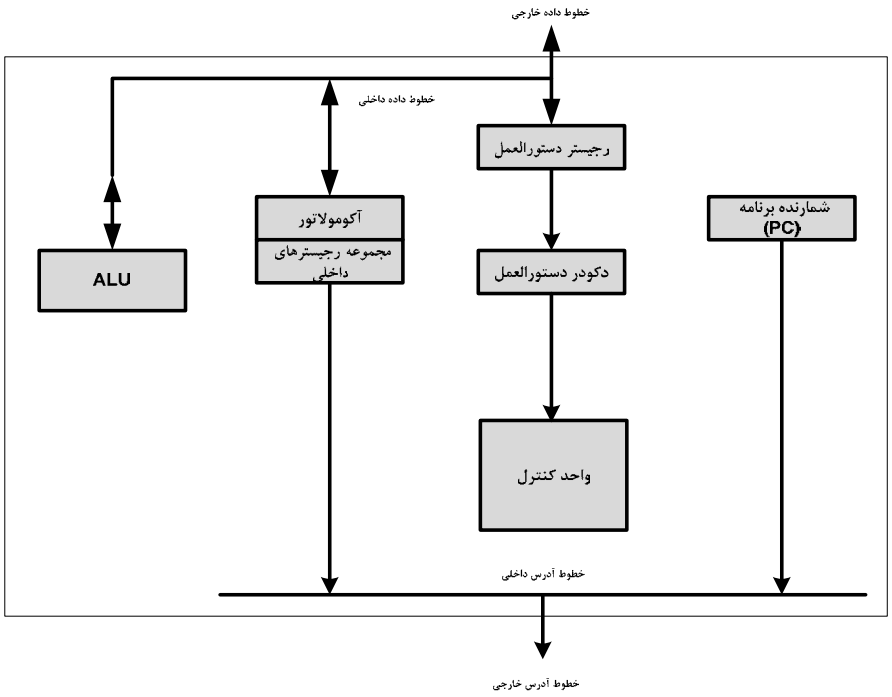
آزمایشگاه معماری کامپیوتر

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 ٤ بیتی می‌باشد. 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 می‌باشد. 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 معمولا دارای ساختمان داخلی بسیار پیچیده با اجزاء بسیار می‌باشند، ولی در مجموع می‌توان بلوکهای نشان داده شده در شکل (۱-۱) را به عنوان بلوکهای اصلی داخلی یک میکروپروسسور برشمرد و سایر اجزاء عمدتا 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 میکروپروسسوری 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 اعمل 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 مباشد، مجموعه‌ای از فلیپ فلاپها یک ثبات را بوجود می‌آورند. ثباتها یا رجیسترهای داخلی به طور معمول در اغلب میکروپروسسورها وجود دارند. این ثباتها عموما در داخل میکروپروسسور برای ثبت موقت اطلاعات بکار می‌روند. برخی از این ثباتها برای استفاده در اعمال داخلی میکروپروسسور اختصاص می‌یابند و برخی دیگراز ثباتها برای کاربرد عام می‌باشند و در دسترس استفاده کننده قرار دارند.

از میان این ثباتها، عموما یک یا چند ثبات با امکانات ویژه طراحی شده و اصطلاحا انباره یا آکومولاتور نامیده می‌شوند. در طراحی میکروپروسسورها، معمولا آکومولاتورها دارای ارتباط مستقیم با ALU بوده و بیشتر اعمال محاسباتی و منطقی مستقیما با آنها قابل انجام است.

۱-٤ 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 می یابد 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 معمولا یک مدار ترکیبی است، در واقع نگهداری اطلاعات در خروجی دیکدر توسط نگهداری اطلاعات در خروجی ثبات دستورالعمل امکان پذیر می‌گردد.

۱-٦ 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 می آورد. 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، وحد کنترل مجددا 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 معمولا مستقیما 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 دیکدر

مقدمه

حافظه ROM یک نوع مدار مجتمع[۱] است که در زمان ساخت داده‌هائی در آن ذخیره می‌گردد. این نوع از حافظه ها علاوه بر استفاده در کامپیوترهای شخصی در سایر دستگاههای الکترونیکی نیز به خدمت گرفته می شوند. حافظه های ROM از لحاظ تکنولوژی استفاده شده، دارای انواع زیر می‌باشند:

- ROM[۲]
- PROM[۳]
- EPROM[۴]
- EEPROM[۵]
- Flash Memory

هر یک از مدل های فوق دارای ویژگی های منحصر بفرد خود می‌باشند. حافظه های فوق در موارد زیردارای ویژگی مشابه می‌باشند:

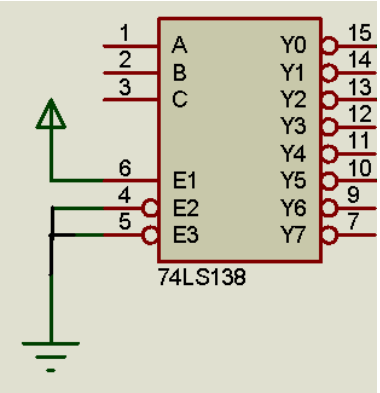
- داده‌های ذخیره شده در این نوع تراشه ها " غیر فرار " بوده و پس از خاموش شدن منبع تامین انرژی، اطلاعات خود را از دست نمی دهند.
- داده‌های ذخیره شده در این نوع از حافظه ها غیر قابل تغییر بوده و یا اعمال تغییرات در آنها مستلزم انجام عملیات خاصی است.

[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

مباحث 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=۱۰۱ باشد، خروجی D_5 یک شده و دیگر خطوط صفر خواهند بود.

آی سی ۷٤۱۳۸ که در شکل (۲-۱) 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 $\overline{E۲}, \overline{E۳}$ و E۱ می‌باشد. برای اینکه این آی سی بصورت دیکدر عمل کند، باید پایه های $\overline{E۲} = \overline{E۳}$ = Low و E۱ = High باشد. بنابراین بر اساس مقادیر ورودی A, B و 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(۲-۱): مدار دیکدر

روش انجام آزمایش

در اینجا می‌خواهیم با استفاده از این دیکدر یک حافظه ROM بصورت ۳×۸ طراحی کنیم که مقادیر ذخیره شده در خانه های حافظه آن بصورت جدول (۲-۱) باشد. همانطور که میدانید، یک دیکدر با استفاده از n متغییر ورودی، ۲^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 ۲^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؟).

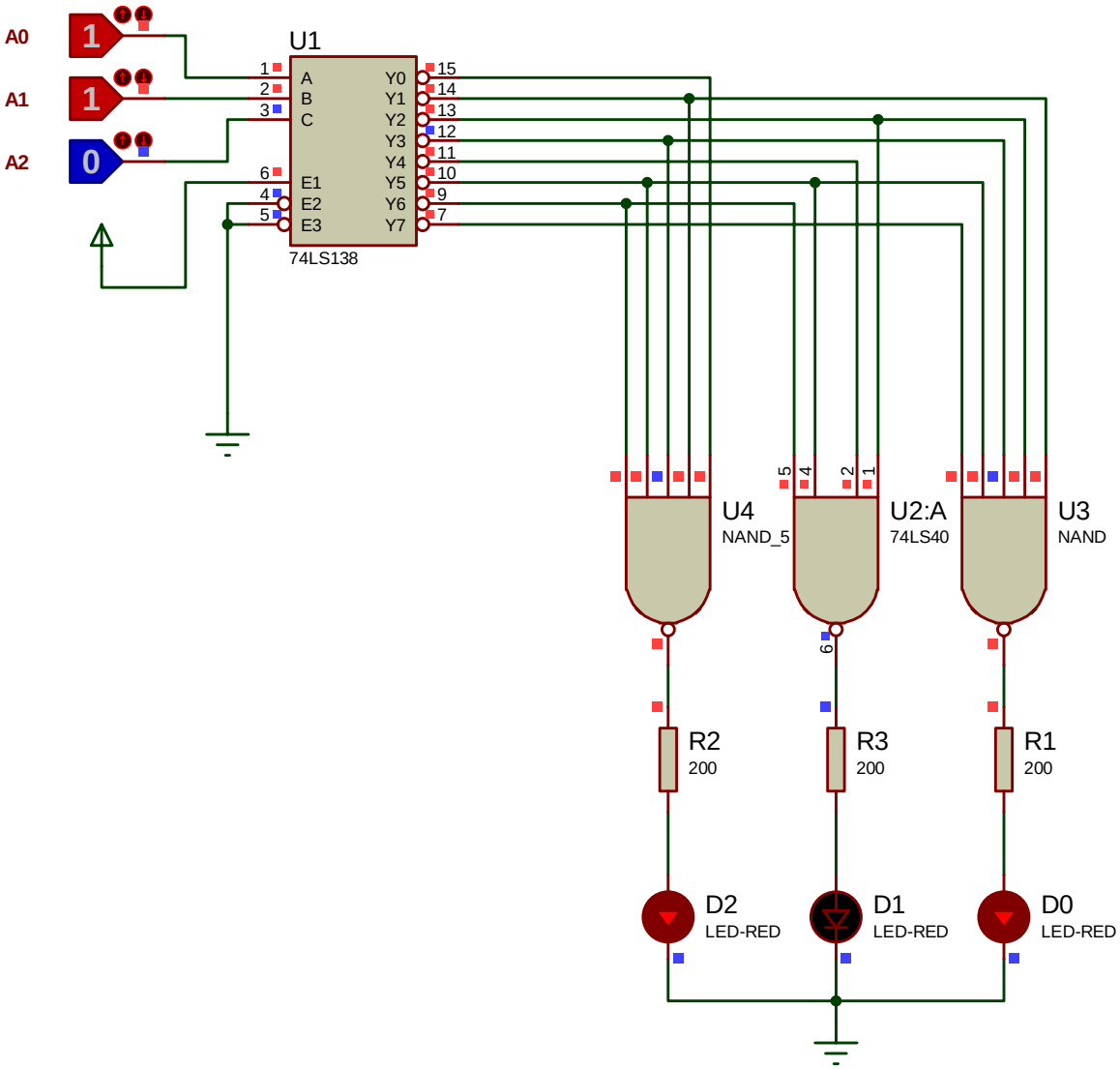
جدول (۲-۱). مقادیر مطلوب در آدرسهای مختلف حافظه

مقدار ذخیره شده در حافظه			آدرس حافظه ROM		
$D_۰$	$D_۱$	$D_۳$	$A_۰$	$A_۱$	$A_۲$
۰	۰	۱	۰	۰	۰
۱	۰	۱	۱	۰	۰
۱	۱	۰	۰	۱	۰
۱	۰	۱	۱	۱	۰
۰	۱	۰	۰	۰	۱
۱	۱	۱	۱	۰	۱
۰	۱	۱	۰	۱	۱
۱	۰	۰	۱	۱	۱

در شکل (۲-۲) مدار خواسته شده برای پیاده سازی مقادیر نشان داده شده در جدول (۲-۱) 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). اندازه‌گیری مقادیر موجود درحافظه بر اساس آدرسهای داده شده

آدرس حافظه ROM			مقدار ذخیره شده در حافظه		
A_2	A_1	A_0	D_3	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 (۲–۱)

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 (۲-۳). اطلاعات حافظه بر اساس آدرسها

مقدار ذخیره شده در حافظه			آدرس حافظه ROM		
D_0	D_1	D_3	A_0	A_1	A_2
۰	۱	۱	۰	۰	۰
۱	۰	۰	۱	۰	۰
۰	۱	۰	۰	۱	۰
۱	۰	۱	۱	۱	۰
۰	۱	۱	۰	۰	۱
۱	۱	۰	۱	۰	۱
۰	۱	۱	۰	۱	۱
۱	۰	۱	۱	۱	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 بورد و بصورت چاپی ساخته می‌شده 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 ٤ در ٤ طراحی کنیم، به صورتی که مقادیر نشان داده شده در جدول (۲-٤) را در خانه های حافظه ذخیره نماید.

جدول (۲-٤): 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

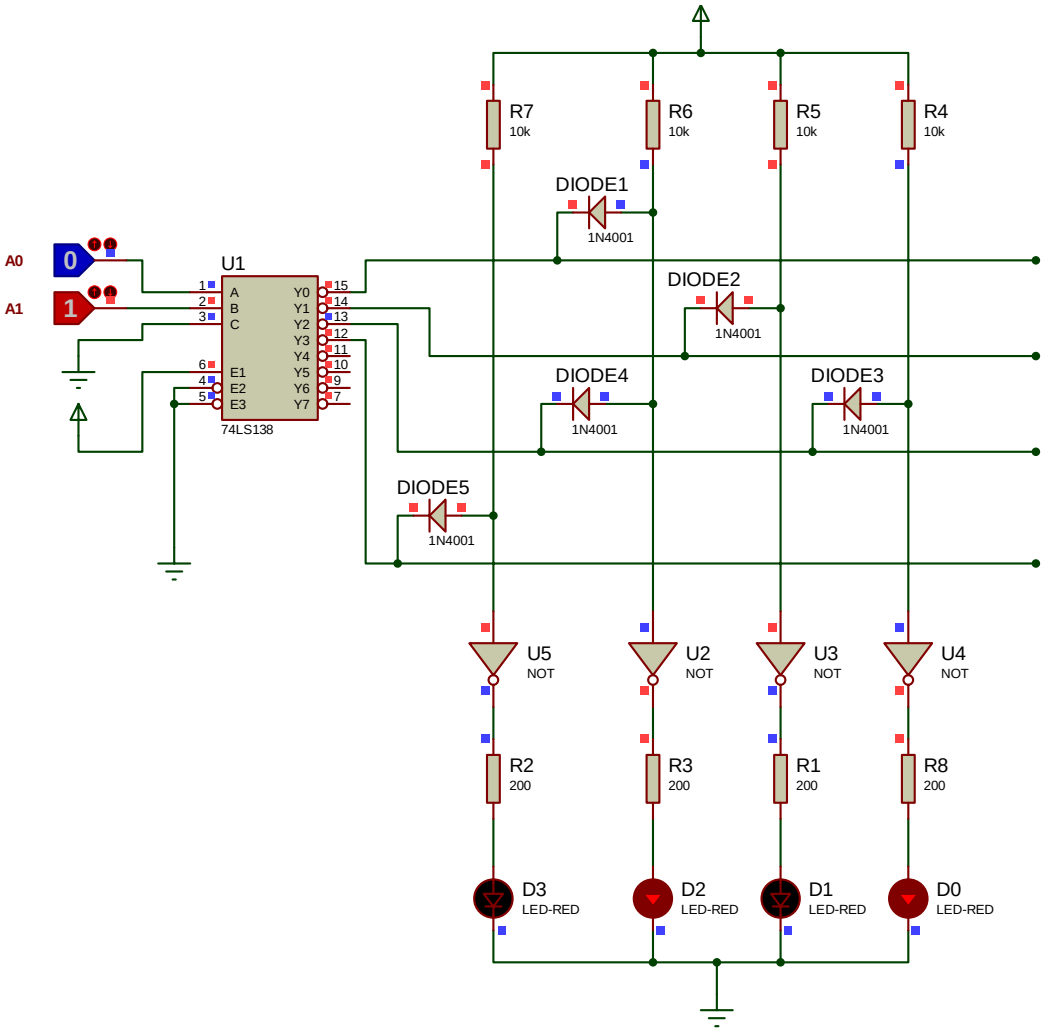
بر اساس ورودی دیکدر، یکی از خروجیها (سطرها) ۰ و بقیه ۱ منطقی می‌باشند. در اینجا ستونها با یک مقاومت به منبع ۵ ولت متصل شده‌اند. در اینجا منطق هر ستون ۱ می‌باشد، مگر اینکه یکی از دیودهایی که به آن ستون متصل شده است روشن شود. یک دیود در صورتی روشن می‌شود که آند آن صفر شود، یعنی آن سطر توسط دیکدر انتخاب شود. در اینصورت با توجه به اینکه اختلاف ولتاژ بین آند و کاتد ۰/٦ ولت می‌باشد، بنابراین با روشن شدن دیود، ولتاژ آند ۰/٦ خواهد شد که درواقع صفر منطقی در نظر گرفته شده است.

به عنوان مثال فرض کنید ورودی $A_1 A_0$=۱۰ 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 ٤ 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:

$A_1 A_0$=۱۰ → $D_3 D_2 D_1 D_0$=۰۱۰۱

روش انجام آزمایش

مدار شکل (۲-٤) را بر روی بردبورد ببندید. از LED های موجود بر روی برد نیز برای دیدن خروجی‌های مدار استفاده نمایید. توجه کنید که برای محدود کردن جریان بایستی یک مقاوم ۲۰۰ اهمی با LED سری شود که در مدار میز کار آزمایشگاه این کار بصورت داخلی انجام شده است و دیگری نیازی به اضافه کردن آن نمی‌باشد. همچنین برای دادن ورودی ۰ و ۱ منطقی (۰ و ۵ ولت) به گیت میتوانید از کلیدهای دیجیتالی استفاده کنید که قادر است ورودی صفر و یا یک را به مدار اعمال کند. با اعمال ورودی‌های متفاوت، خروجی مدار رادر جدول (۲-۵) ثبت کرده و نتایج عملی را با نتایج تئوری و شبیه سازی مقایسه کنید.



شکل (۲-٤). مدار یک حافظه ROM با ساختار دیودی

جدول (۲–۵). اندازه‌گیری مقادیر موجود درحافظه بر اساس آدرسهای داده شده

مقدار ذخیره شده در حافظه				آدرس حافظه ROM	
D.	D_1	D_2	D_3	A.	A_1
?	?	?	?	۰	۰
?	?	?	?	۱	۰
?	?	?	?	۰	۱
?	?	?	?	۱	۱

تمرین شبیه سازی

با استفاده از یک دیکدر ۳ به ۸ و دیود، یک حافظه ROM ٤ در ٤ طراحی کنید که مقادیر نشان داده شده در جدول (۲–٦) 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 (۲–٦). اطلاعات حافظه بر اساس آدرسها

مقدار ذخیره شده در حافظه				آدرس حافظه ROM	
D.	D_1	D_2	D_3	A.	A_1
۱	۰	۱	۰	۰	۰
۰	۱	۰	۱	۱	۰
۱	۰	۱	۰	۰	۱
۰	۰	۱	۱	۱	۱

آزمایش ۲

موضوع آزمایش: آشنایی با نحوه 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 خانـه‌ی قبلـی بیشتر است، هر آدرس حافظه، قابلیت نگهداری یک یا چند بایت را دارا است.

حافظه با قابلیت دسترسی تصادفی (در اصطلاح به آن RAM[۶] می گویند) یک نوع محـل ذخیـره اطلاعـات در کـامپیوتر می باشد از اطلاعات در هر زمانی بدون در نظر گرفتن موقعیت فیزیکی اطلاعات وترتیب آنها وجود دارد امروزه شـامل مـدارهای مجتمعی می‌باشد که امکان ذخیره سازی اطلاعات را به صورت تصادفی به ما می‌دهد. واژه تصادفی به این معنی است کـه امکـان دسترسی به هر مقدار اطلاعات در هر زمانی بدون در نظر گرفتن موقعیت فیزیکی اطلاعـات وترتیـب آنهـا وجـود دارد. داده‌هـای موجود در RAM قابل پاک شدن و جایگزینی با داده‌های دیگر هستند و هر نوع وقفه‌ای در جریان برق رایانه، موجب از بین رفتن داده‌های موجود در RAM می‌شود. استفاده از این نوع حافظه‌ها، برای نگهداری موقت اطلاعات تا زمان پردازش یا انتقال نتایج بـه بیرون از رایانه و یا ذخیره در حافظه‌های جانبی است.

از نظر تکنولوژی ساخت، دو نوع RAM وجود دارد :

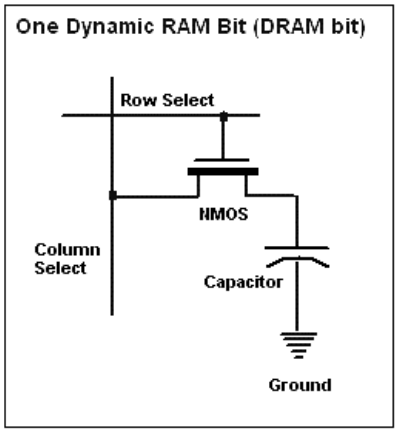
۱. RAM پویا (DRAM)[۷]

۲. RAM ایستا (SRAM)[۸]

در حافظه DRAM، با بکارگیری یک خازن و یک ترانزیستور می‌توان یک سلول را ایجـاد کـرد. سـلول فـوق قـادر بـه نگهداری یک بیت داده خواهد بود. خازن اطلاعات مربـوط بـه بیـت را کـه یـک یـا صـفر اسـت ، در خـود نگهـداری خواهـد

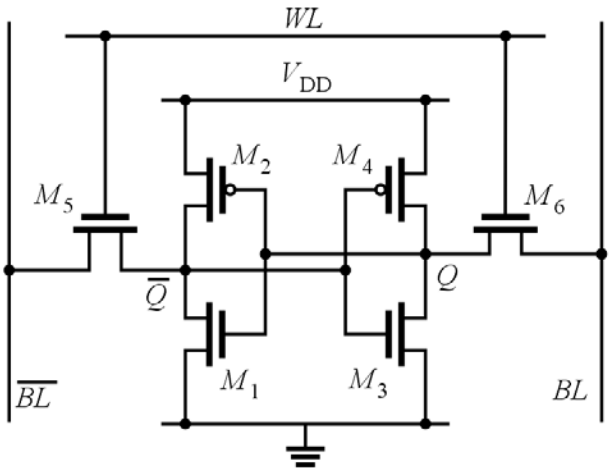
[۶] Random Access Memory
[۷] Dynamic RAM
[۸] Static Ram

کرد.عملکرد ترانزیستور مشابه یک سوییچ 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 الکترون ها است . به منظور ذخیره سازی مقدار" یک" درحافظه، ظرف فوق می بایست از الکترونها پرگردد. برای ذخیره سازی مقدار صفر، می بایست 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 ، می بایست 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 مجددا" اطلاعات را بازنویسی می‌نماید. عملیات بازنویسی اطلاعات، هزاران مرتبه در یک ثانیه تکرار خواهد شد. علت نامگذاری DRAM بدین دلیل است که این نوع حافظه ها مجبور به بازخوانی اطلاعات بصورت پویا خواهند بود. فرآیند تکراری " بازخوانی- بازنویسی اطلاعات" در این نوع حافظه ها باعث می‌شود که زمان تلف و سرعت حافظه کند گردد. سلول های حافظه بر روی یک تراشه سیلیکون و بصورت آرایه ای مشتمل از ستون ها (خطوط بیت) و سطرها (خطوط کلمات) تشکیل می گردند. نقطه تلاقی یک سطر و ستون بیانگر آدرس سلول حافظه است.



شکل (۲-۵): یک سلول حافظه دینامیکی

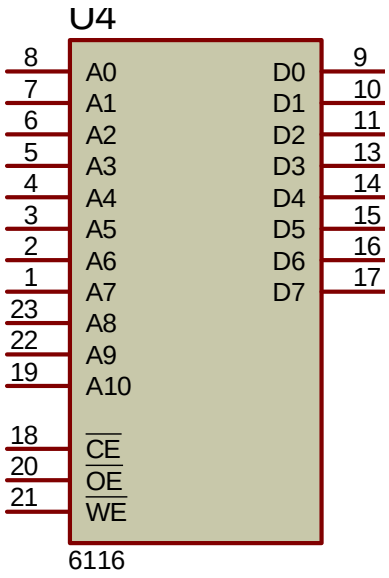
حافظه های SRAM دارای یک تکنولوژی کاملا" متفاوت می‌باشند. در این نوع از حافظه ها از فلیپ فلاپ برای ذخیره سازی هر بیت حافظه استفاده می‌گردد. یک فلیپ فلاپ برای یک سلول حافظه، از چهار تا شش ترانزیستور استفاده می کند . حافظه های SRAM نیازمند بازخوانی / بازنویسی اطلاعات نخواهند بود، بنابراین سرعت این نوع از حافظه ها به مراتب از حافظه های DRAM بیشتر است .با توجه به اینکه حافظه های SRAM از بخش های متعددی تشکیل می‌گردد، 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 می‌تواند باعث افزایش قیمت این نوع از حافظه ها گردد. بنابراین حافظه های SRAM سریع و گران و حافظه های DRAM ارزان و کند می‌باشند . 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 می‌گردد.



شکل (۲-٦): یک سلول حافظه استاتیکی

مباحث تئوری

آی سی ٦١١٦ نوعی **RAM** استاتیک می‌باشد که قابلیت گنجایش ۲ کیلو بایت اطلاعات را دارد. برای همین، ۱۱ خط آدرس دارد (**A۰** تا **A۱۰**). همان طور که در شکل (۲-۷)می بینید، این آی سی یک پایه فعال ساز $\overline{\mathbf{CE}}$ و دو پایه $\overline{\mathbf{WE}}$ و $\overline{\mathbf{OE}}$ دارد که به ترتیب برای عملیات نوشتن **(Write)** و خواندن **(Read)** مورد استفاده قرار می‌گیرد.



شکل (۲-۷): حافظه استاتیکی ٦١١٦

ما در آزمایش این بخش، تنها از پایه های آدرس **A.** و **A۱** و پایه های داده **D.** تا **D۳** استفاده خواهیم کرد. به عبارت دیگر، اطلاعات را در یک **RAM** ٤ در ٤ نوشته و آن را میخوانیم. به همین دلیل، همانطور که در شکل (۲-۸) نشان داده شده است، پایه های آدرس **A۲** تا **A۱.** همگی به زمین متصل شده‌اند. مودهای عملکرد این آی سی در جدول (۲-۷) نشان داده شده است. برای عملکرد آی سی، پایه $\overline{\mathbf{CE}}$ بایستی همواره زمین باشد. توجه کنید که در اینجا ما تنها از دو حالتی که در جدول (۲-۷)سایه دار شده‌اند استفاده خواهیم کرد.

جدول (۲–۷). مودهای عملکرد آی سی ٦١١٦

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

نحوه انجام آزمایش

نوشتن در حافظه RAM

همانطور که گفته شد، می‌خواهیم از آی سی ٦١١٦ به عنوان یک حافظه ٤ در ٤ استفاده کنیم. فرض کنید بخواهیم داده‌های نشان داده شده در جدول (۲–۸) را در حافظه ذخیره کرده و سپس آنها را از حافظه بخوانیم.

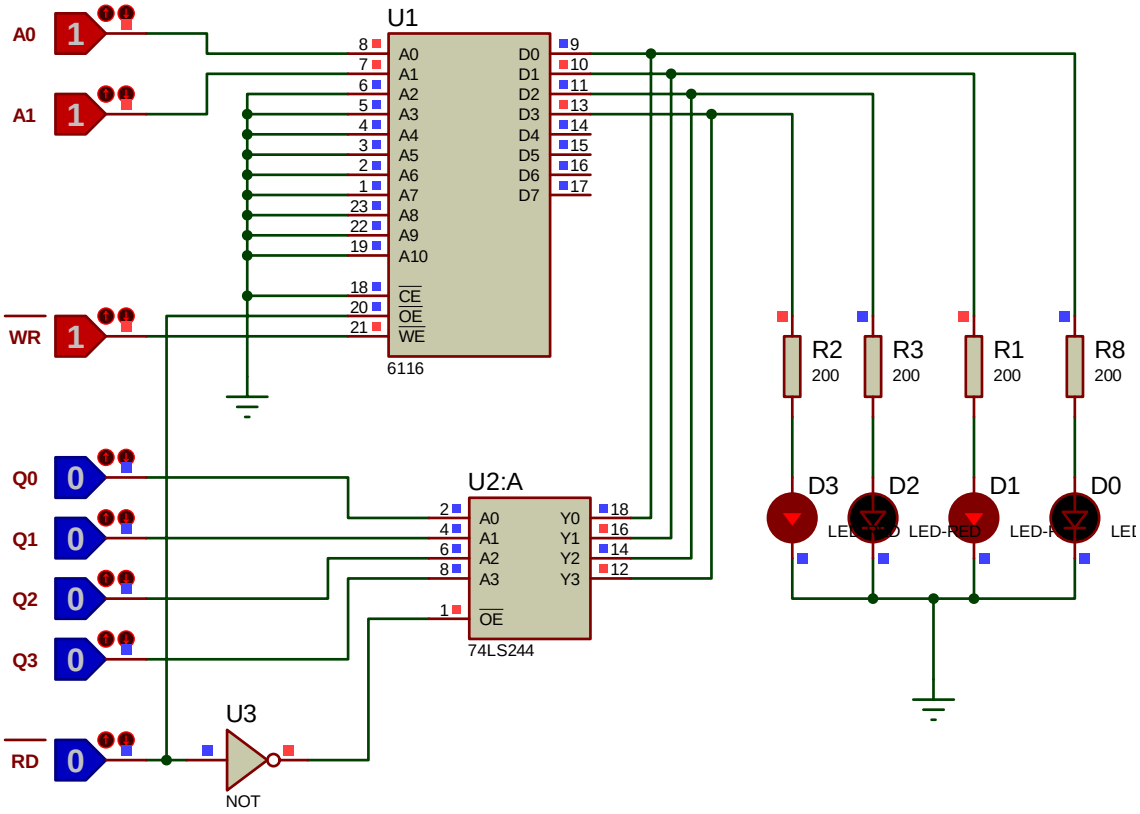
جدول (۲–۸). داده‌هایی که می‌خواهیم در حافظه ذخیره شوند.

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

بدین منظور ابتدا مدار شکل (۲–۸) را بر روی بردبورد بسته و داده‌های نشان داده در جدول (۲–۹) 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 (۲–۸) 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 $\overline{\mathbf{WR}}$ را یک بار '۰' کرده و سپس آنرا به حالت '۱' منطقی می‌بریم. در طول فرآیند ذخیره داده‌ها، پایه $\overline{\mathbf{RD}}$ همواره غیر فعال ('۱' منطقی) باقی می ماند.

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

شماره خانه حافظه	$A_۱$	$A_۱$	$Q_۳$	$Q_۲$	$Q_۱$	$Q_۰$			
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 (۲–۸). مدار یک حافظه **RAM** با استفاده از آی‌سی ٦١١٦

خواندن از حافظه RAM

بعد از ذخیره کردن داده‌ها بر روی **RAM**، نوبت به خواندن داده‌ها از آن می رسد. بدین منظور، پایه $\overline{\mathbf{WR}}$ را غیر فعال کرده (آن را '۱' کنید) و پایه $\overline{\mathbf{RD}}$ را فعال کنید (آن را '۰' کنید). اگه به شکل (۲–۸) دقت کنید مشاهده می‌کنید که با غیر فعال کردن پایه $\overline{\mathbf{WR}}$، آی سی ٧٤**LS**٢٤٤ که همان مدار بافر می‌باشد، غیر فعال شده و خروجی آن در حالت امپدانس بالا قرار می‌گیرد. به عبارت دیگر، ورودهای **Q** دیگر تاثیری نخواهند داشت. حال مقادیر داده شده در جدول (۲–۱۰) را به خطوط آدرس داده و داده‌های ذخیره شده در آنها را که بر روی **LED** ها نمایش داده می‌شود را در جدول ثبت نمائید. آیا داده‌های خوانده شده همان داده‌های ذخیره شده می‌باشند؟

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

A_1	A_0	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

مباحث تئوری

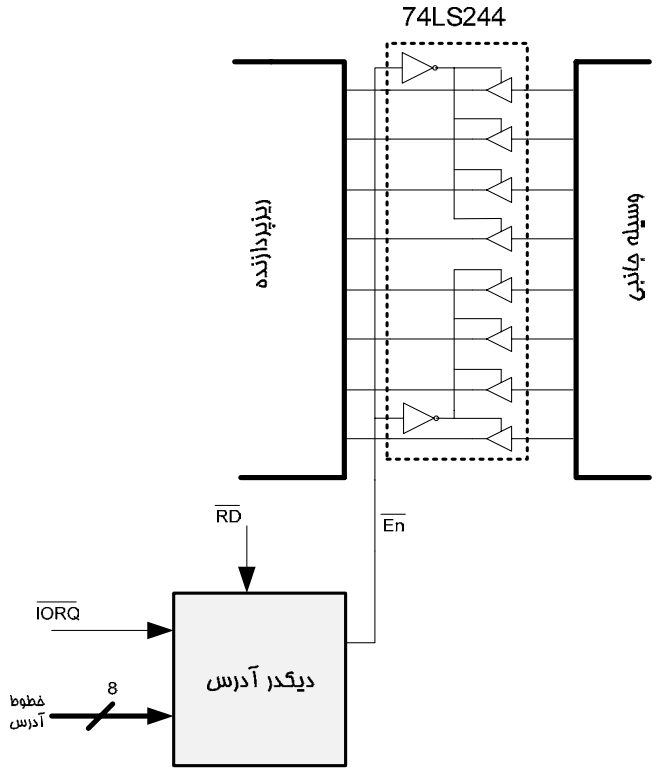
در سیستمهای میکروپروسسوری، از مدارهای مختلفی به عنوان درگاه موازی ورودی/خروجی **(I/O)** می‌توان استفاده کرد:

- ✓ بافرهای سه حالته
- ✓ رجیسترها
- ✓ مدارهای ورودی/خروجی قابل برنامه ریزی (همانند آی سی ۸۲۵۵)

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

بافرهای سه حالته از جمله مدارهایی هستند که به عنوان درگاه ورودی/خروجی در سیستمهای میکروپروسسوری مورد استفاده قرار می‌گیرند. این مدارها عموما به صورت گیتهای ۸ گانه در یک **IC** با پایه یا پایه های فعالساز[۹] مشترک قرار دارند (بعنوان یک نمونه از بافرهای سه حالته می‌توان **۷٤LS۲۷۷** 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.

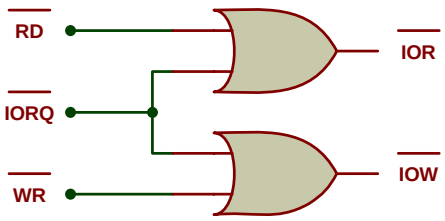
[۹] Enable



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

سیگنال‌های کنترلی و دستورالعمل‌های اصلی I/O

ریزپردازنده Z۸۰ 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 **A۰-A۷** از گذرگاه آدرس هستند به داخل انباره وارد می‌کند. در شکل (۲-۱۰)، نحوه تولید سیگنالهای کنترلی نشان داده شده است.



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

همانطور که گفته شد، در شکل (۲–۹) از آی سی **۷٤LS۲٤٤** که یک بافر سه وضعیتی می‌باشد، بـرای انتقـال اطلاعـات ورودی به گذرگاه اطلاعات استفاده می‌شود. در **Z۸۰** برای انتقال اطلاعات بین ریزپردازنده و بافر سه وضعیتی موجـود در **I/O** از دستورات **OUT A,( )** و **IN A,( )** استفاده می‌گردد. با این مدار هر داده **TTL** را می‌توان به گذرگاه اطلاعات داخلی وارد کـرد. از آنجایی که ممکن است چندین وسیله جانبی به خطوط داده میکروپروسسور متصل شده باشـند، بنـابراین خروجـی بایسـتی دارای حالت امپدانس بالا نیز باشد.

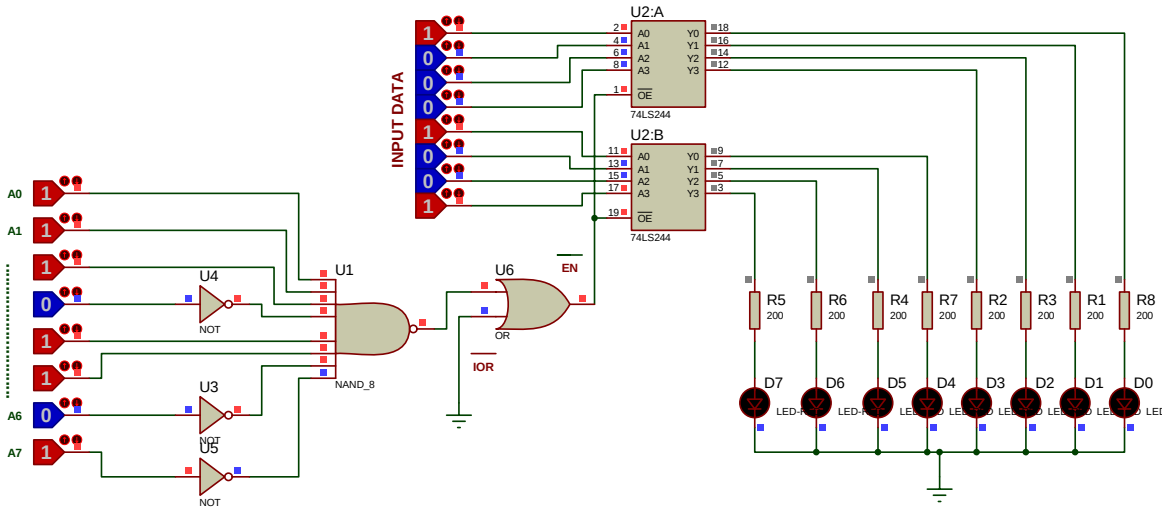
یکی از اهداف مهم این آزمایش، نحوه طراحی دیکدر آدرس می‌باشد. وظیفه دیکـدر آدرس آن اسـت کـه بـا توجـه بـه دستورالعمل ریزپردازنده، از چندین وسیله جانبی که ممکن است به آن متصل شده باشـند، تنهـا یکـی از آنهـا را فعـال و بقیـه را غیرفعال کند تا بتواند داده‌های خود را به ریزپردازنده انتقال دهد.

نحوه انجام آزمایش

فرض کنید ریزپردازنده دستورالعمل **[IN A, (۳۷H)]** را اجرا کند. با این دستورالعمل، بایستی وسیله جانبی که آدرس آن **۳۷H** می‌باشد، فعال شده و بقیه غیر فعال باقی بمانند. در مدار شکل (۲–۱۱)، دیکدر آدرس برای انجام این دسـتورالعمل طراحـی شده است. نحوه‌ی عملکرد دیکدر آدرس را تشریح کنید. سپس مدار را بر روی بردبورد بسـته و یـک ورودی دلخـواه بـه درگـاه 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 می‌توانید از **LED** استفاده کنید. توجه کنید که آی سی **۷٤LS۲٤٤** در نرم افزار شبیه سازی شامل ٤ پایـه ورودی می‌باشد، بنابراین بایستی از دو آی سی استفاده کرد تا بتوان ۸ بیت داده ریزپردازنده **Z۸۰** را انتقال داد.

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

$A_7A_6A_5\ A_4\ A_3\ A_2\ A_1A_0$	$D_7D_6D_5\ D_4\ D_3\ D_2\ D_1D_0$
۰۰۰۰ ۱۰۱۰	?
۰۰۱۱ ۱۱۱۱	?
۰۰۱۱ ۰۱۱۱	?



شکل (۲–۱۱) 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, (۵٤H)] 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 می شوند، یعنی در هر لحظه و یا سیکل زمانی، یک بیت داده از طریق سیم می‌تواند منتقل شود که ۰ و یا ۱می‌باشد. دو روش برای انتقال داده‌های سریال وجود دارد :

- ✓ **انتقال سنکرون**:

در انتقال سنکرون، داده‌ها در بلوک هایی فرستاده می‌شود و فرستنده و گیرنده به وسیله کاراکتر های 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،

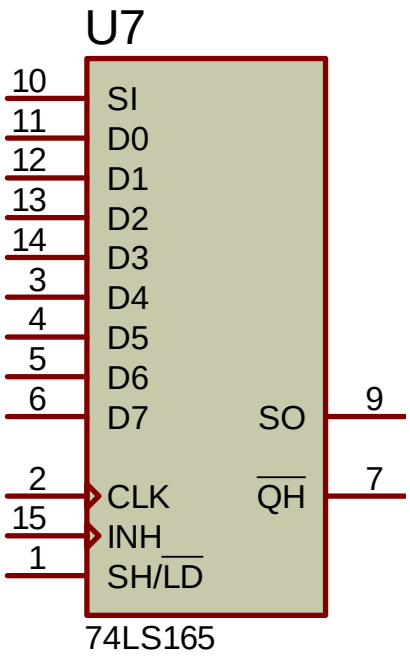
[۱۰] Sync Character

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، ۷٤LS١٦٥ می‌باشد که در شکل (۲–۱۲) نشان داده شده است. این آی‌سی، داده‌ها را بصورت هشت بیتی و از طریق پایه‌های $D_0,D_1,.....D_7$ 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 پایه های$D_0,D_1,.....D_7$ اعمال کرده و سپس پایه $\overline{Load}/Shift$ را صفر کنیم. در ادامه و برای انتقال سریال داده‌ها، پایه$\overline{Load}/Shift$ را '۱' کرده و با هر لبه بالا رونده کلاک، بیت‌ها به صورت سریال از آی سی خارج می‌شوند.

[11] Serial Output



شکل (۲-۱۲): شمای کلی آی سی ۷٤LS۱٦٥ با ورودی موازی و خروجی سریال

جدول (۲-۱۲). نحوه عملکرد آی سی ۷٤LS۱٦٥

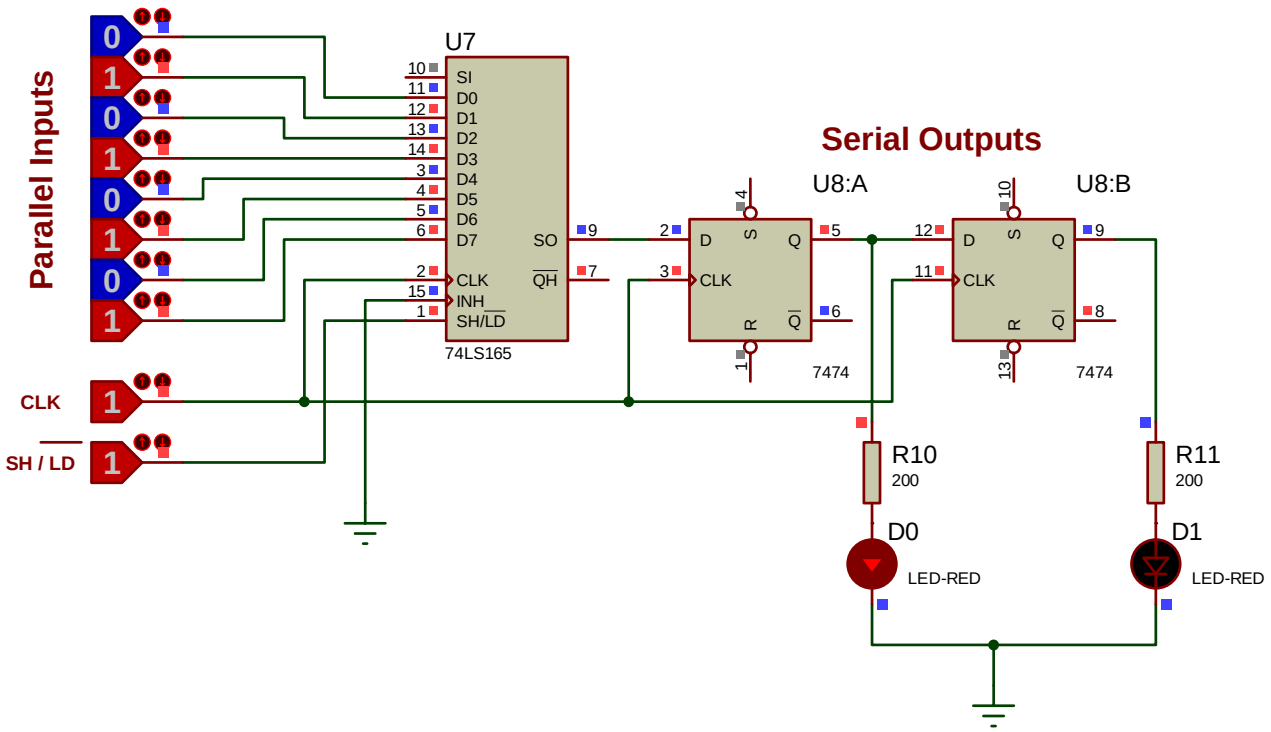
Inputs		Function
	Clock	
L	×	Parallel Load
H		Shift

نحوه انجام آزمایش

مدار شکل (۲-۱۳) را ببندید و یک ورودی دلخواه به مدار اعمال کنید. برای مشاهده بهتر تغییرات درخروجی، مقادیر نشان داده شده در شکل (۲-۱۳) را می‌توانید اعمال کنید. سپس پایه $\overline{\text{Load}}$/Shift 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. مشاهده نمایید. با هر کلاک، بیت قبلی به سمت راست شیفت داده می‌شود و منطق آن را می‌توانید در LED/D۱ 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	منطق مشاهده شده در LED/D۰							
$D_۷D_۶D_۵ D_٤ D_۳ D_۲ D_۱D_۰$	۱th CLK	۲th CLK	۳th CLK	٤th CLK	۵th CLK	٦th CLK	۷th CLK	۸th CLK
۱۰۱۰۱۰۱۰	؟	؟	؟	؟	؟	؟	؟	؟

آزمایش ۵

موضوع آزمایش: بررسی عملکرد باس موازی

مقدمه

به علت تعدد تجهیزات متصل شده بـه CPU در یـک سیسـتم میکروپروسسـوری بـرای داشـتن سـرعت بـالای انتقـال اطلاعات، باید از اتصالات و انتقال موازی داده‌ها اسـتفاده کـرد. عمـلا اتصـال مسـتقل تـک تـک اجـزاء و خطـوط داده زیـاد بـه میکروپروسسور غیر ممکن است، چرا که موجب افزایش فراوان اتصالات سیمی خواهد شد و در نهایت منجبر به کـاهش قابلیـت اطمینان در سیستم می‌گردد. به همین دلیل، در این سیستمها از مجموعه های ارتباطی تحت عنوان باس استفاده می‌شود.

منظور از باس مجموعه خطوطی است که به صورت یک مسیر مشترک به بخشـهای مختلـف سیسـتم متصـل شـده و امکـان برقراری ارتباط بین آنها را فراهم می‌کند. همانطور که در شکل (۲-۱٤) نیز نشان داده شده است، این باس خود می‌توانـد متشـکل از چند باس یا مجموعه خطوط دیگر باشد:

- ✓ **باس داده‌ها**

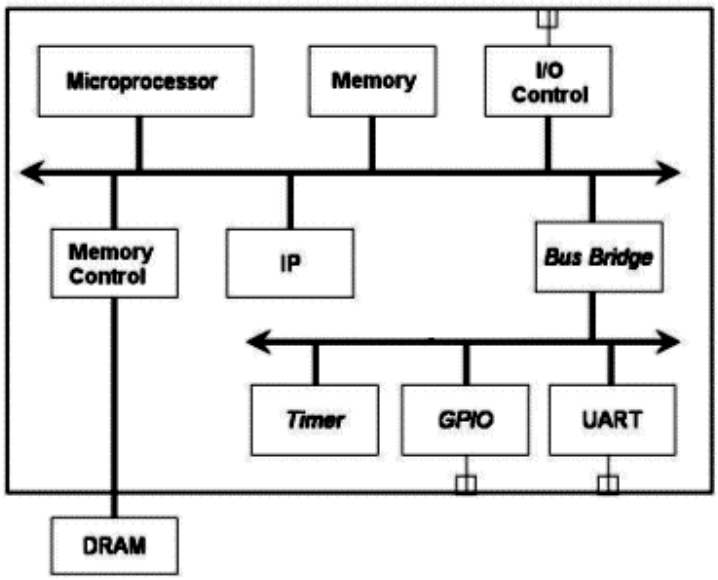
عبارتست از مجموعه‌ای از خطوط که برای مبادله داده‌ها، عمدتا بین CPU و بقیه اجزا سیستم مورد استفاده قرار می‌گیـرد. واضـح است که خطوط داده فوق باید خطوطی دو طرفه باشند و توانایی انتقال داده در دو جهت ورودی و خروجی را داشته باشند.

- ✓ **باس آدرس**

بر خلاف خطوط داده‌ها، خطوطی یک طرفه می‌باشند که معمولا 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 نحـوه ارتبـاط میکروپروسسور با سایر واحدها می‌باشند.



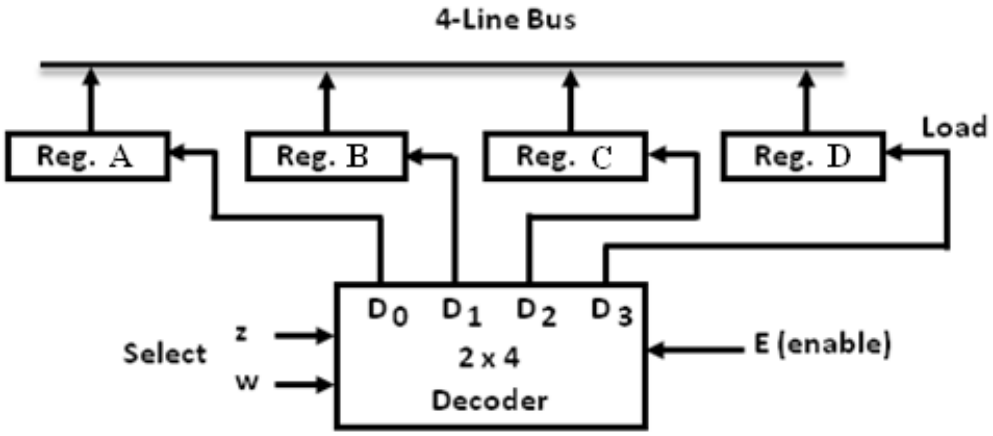
شکل (۲-۱٤). نحوه ارتباط اجزا سیستم توسط باس

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 حالته با قابلیت امپدانس بالا در خروجی استفاده کرد. در آزمایش ۳ در مورد آی سی بافر سه حالته 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 (۲-۱۵) یک گذرگاه مشترک ٤ بیتی را برای ٤ ثبات نشان می‌دهد که بر اساس دیکدر طراحی شده است. بر اساس مقادیر خطوط انتخا .S و S_1 یکی از ثباتهای A, B, C و یا D انتخاب می‌شوند (جدول ۲-١٤).



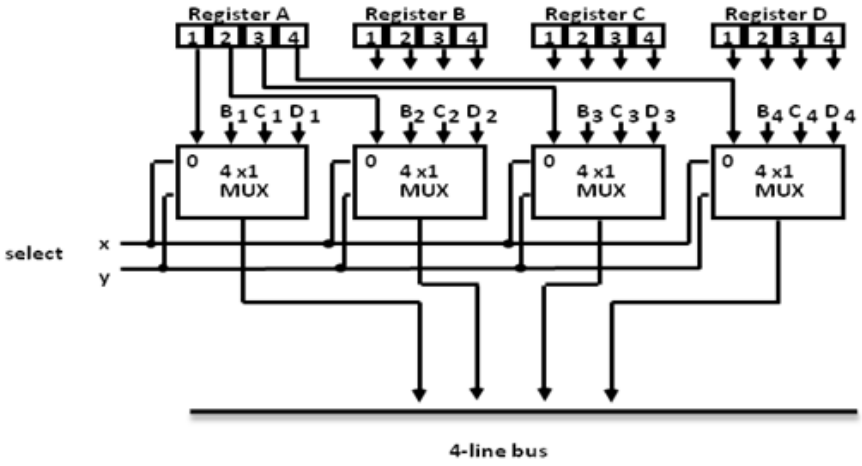
شکل (۲-۱۵). نحوه ارتباط چهار ثبات ٤بیتی با استفاده از باس

جدول (۲-١٤). نحوه انتخاب ثباتها در باس

ثباتی که انتخاب می‌شود	S.	S_1
A	۰	۰
B	۱	۰
C	۰	۱
D	۱	۱

طراحی باس موازی با استفاده از مالتی پلکسر

یک روش دیگر ساخت سیستم گذرگاه مشترک، استفاده از مولتی پلکسرها می‌باشد. مولتی پلکسرها از بین داده‌های ثباتهای مختلف، خروجی ثبات مبدا را انتخاب و سپس اطلاعات آن را روی گذرگاه قرار می‌دهند. توجه کنید که در اینجا لازم نیست از رجیسترهایی که دارای خروجی امپدانس بالا نیز هستند، استفاده کرد (چرا؟). شکل (۲–۱٦) یک گذرگاه مشترک ٤ بیتی را برای ٤ ثبات نشان می‌دهد که بر اساس مالتی پلکسر طراحی شده است. بر اساس مقادیر خطوط انتخا .S و S_1 یکی از ثباتهای A, B, C و یا D انتخاب می‌شوند. نحوه عمل گذرگاه چهاربیتی شکل (۲–۱٦) در جدول (۲–۱۵)نشان داده شده است.



شکل (۲–۱٦). گذرگاه چهار بیتی برای چهار ثبات

جدول (۲–۱۵). نحوه عملکرد گذرگاه

ثباتی که انتخاب می‌شود	S.	S_1
A	۰	۰
B	۱	۰
C	۰	۱
D	۱	۱

روش انجام آزمایش

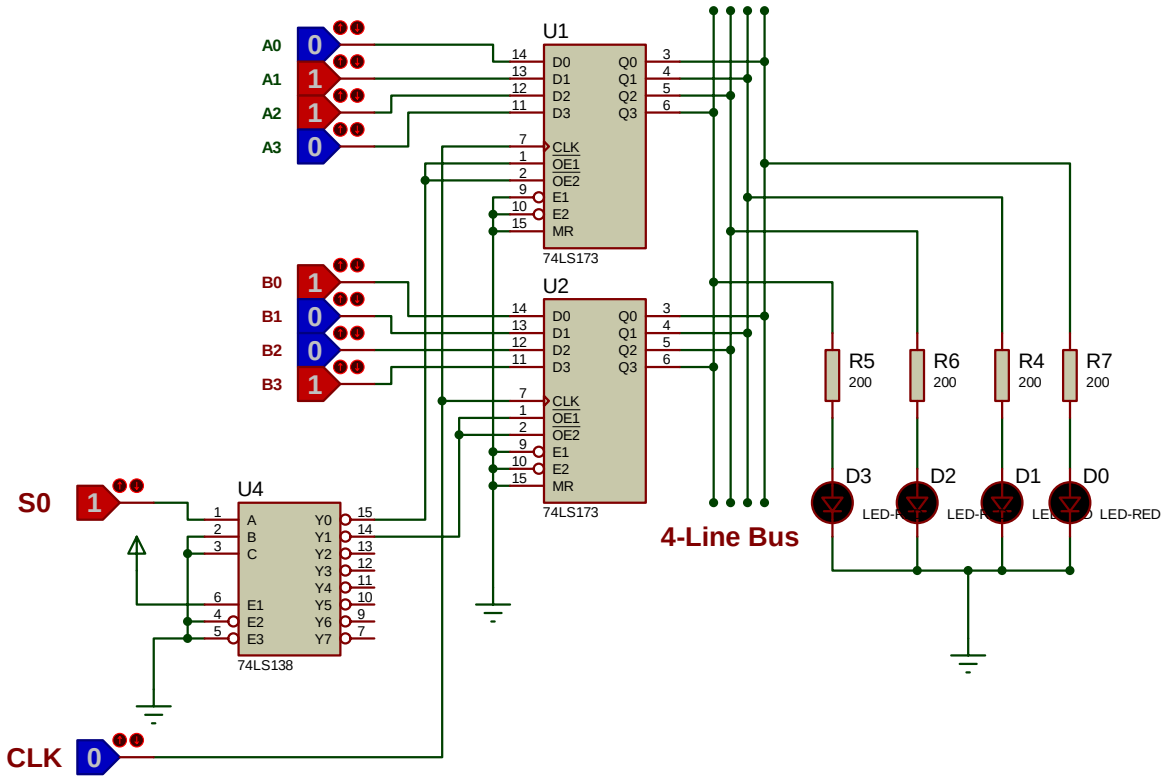
(طراحی باس موازی با استفاده از دیکدر)

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 می‌باشد، مجموعه‌ای از فلیپ فلاپها یک ثبات را بوجود می‌آورند. ثباتها یا رجیسترهای داخلی به طور معمول در اغلب میکروپروسسورها وجود دارند. این ثباتها عموما در داخل میکروپروسسور برای ثبت موقت اطلاعات بکار می‌روند. برخی از این ثباتها بطور ویژه در اعمال داخلی میکروپروسسور استفاده می‌شوند و برخی دیگر ثباتهای با کاربرد عام می‌باشند و در دسترس استفاده کننده قرار دارند. تعداد و اندازه این ثباتها بستگی به میکروپروسسور و تعداد بیت آن دارد. ثباتهای 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 سیکل‌های اضافی ارتباط با حافظه خارجی جهت ذخیره سازی و بازیابی اطلاعات می‌شوند.

آی سی ۷٤LS۱۷۳ یک ثبات با چهار فلیپ فلاپ D و خروجی سه حالته می‌باشد. در لبه بالا رونده کلاک، ورودی به خروجی آن منتقل می‌شود. برای عملکرد نرمال ثبات، پایه های فعال کننده ورودی IE۱ و IE۲ و همچنین پایه های فعال کننده خروجی OE۱ و OE۲ بایستی به زمین متصل شوند. اگر پایه های OE۱ و OE۲ به منطق '۱' متصل شوند، خروجی در حالت امپدانس بالا قرار می‌گیرد که اساس طراحی مدار باس می‌باشد. بنابراین با استفاده از یک دیکدر با خروجی مکمل شده، در هر لحظه تنها پایه های OE۱ و OE۲ یک ثبات '۰'، و بقیه ثباتها '۱' (امپدانس بالا) خواهند بود. برای سادگی، ما در اینجا مدار را برای تنها دو ثبات ٤ بیتی طراحی کردیم. قبل از بستن مدار، مشخصات آی سی‌ها را به دقت مطالعه کنید.

بعد از بستن مدار شکل (۲-۱۷)، 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. . 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. اگر بیت S۰، صفر باشد، ثبات A و اگر S۱ یک باشد، ثبات B به خروجی منتقل می‌شود. بـرای مشـاهده خروجی می‌توانید از LED استفاده کنید. این آزمایش را برای مقادیر جدول (۲–۱٤) نیز انجام داده و آنرا تکمیل کنید.



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

جدول (۲–۱٦). مقادیر مربوط به مرحله اول آزمایش پنجم

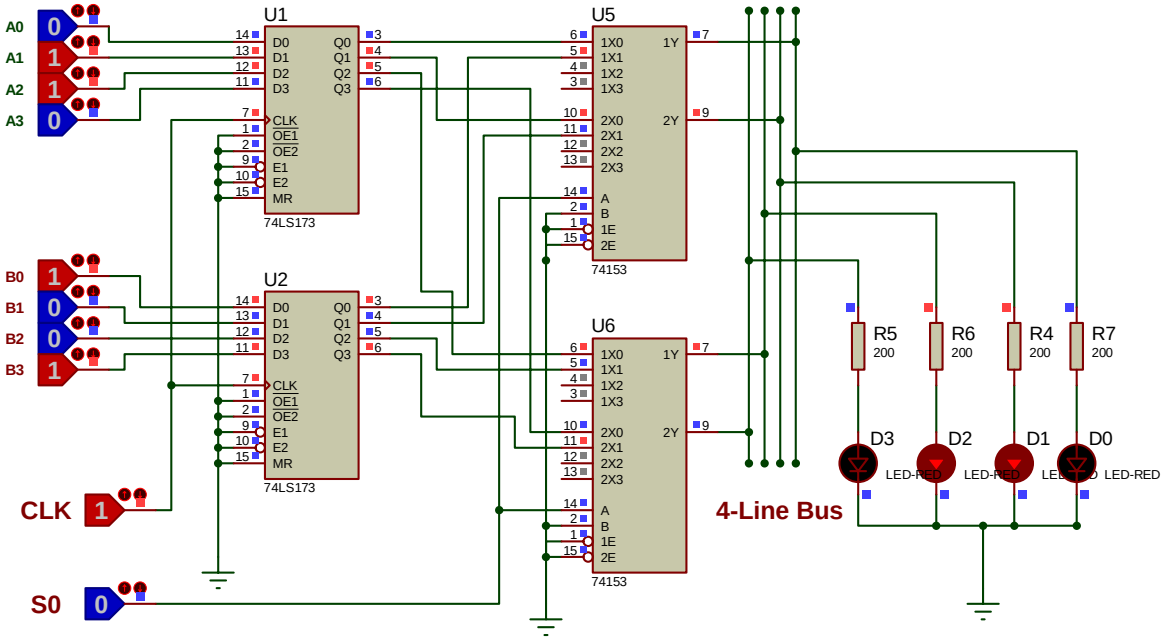
A_3 A_2 A_1A_0	B_3 B_2 B_1B_0	D_3 D_2 D_1D_0	
		S۰= "۱"	S۰= "۰"
۰۱۱۰	۱۰۰۱	?	?

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 که آی سی ۷٤LS۱۵۳ دارای دو مالتی پلکسر ٤ 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 S۰ 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

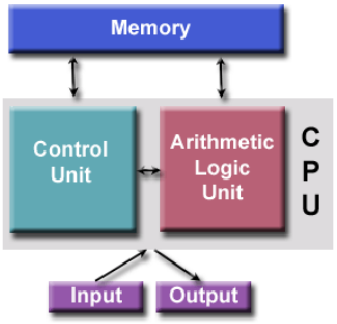
$A_3\ A_2\ A_1 A_0$	$B_3\ B_2\ B_1 B_0$	$D_3\ D_2\ D_1 D_0$	
		S_0= "۱"	S_0= "۰"
۰۱۱۰	۱۰۰۱	?	?

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 داده‌ها انجام دهد. ALU نتایج را بر روی یک رجیستر خروجی ذخیره می کند. اجزای دیگر این اطلاعات را بین رجیسترها و حافظه جابه‌جا می کنند. شمای کلی نحوه ارتباط ALU با بخشهای مختلف ریزپردازنده در شکل (۲-۱۹) نشان داده شده است.



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

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

[۱۲] Arithmetic Logic Unit
[۱۳] Control unit

✓ **عملیات های ساده:**

اکثر ALU ها می‌توانند این عملیات ها را انجام دهند:

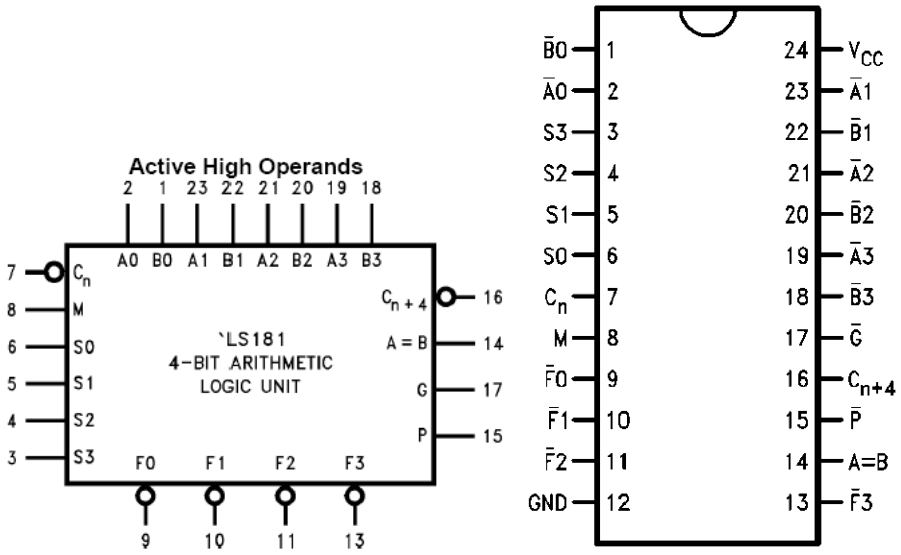
- عملیات های محاسباتی صحیح (جمع ، تفریق و بعضی ضرب و تقسیم ..)
- عملیات های بیتی منطقی (AND، NOT، OR، XOR)
- عملیات های انتقال بیتی (Bit-shifting) (انتقال یا چرخش یک کلمه با تعداد مشخصی بیت به راست یا چپ؛ با/بدون گسترش علامت)

✓ **عملیاتهای پیچیده:**

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

مباحث تئوری

هدف از این آزماش، آشنایی با نحوه کار یک ALU و انجام چندین عملیات ریاضی و منطقی ساده می‌باشد. در این آزمایش از آی سی ۷٤LS۱۸۱ استفاده خواهیم کرد که یک ALU چهار بیتی می‌باشد. در شکل (۲-۲۰)، شمای کلی این آی سی نشان داده شده است.



شکل (۲–۲۰): شمای کلی آی سی ۷٤LS۱۸۱ یک ALU چهاربیتی

در جدول(۲–۱۸) پایه های آی سی تشریح شده‌اند. این آی‌سی دارای ورودی بـرای دو عملونـد $\overline{A۰} - \overline{A۳}$ و $\overline{B۰} - \overline{B۳}$ و پایه‌های خروجی $\overline{F۰} - \overline{F۳}$ می‌باشد.

جدول (۲–۱۸). توصیف پایه‌های آی‌سی ۷٤LS۱۸۱ 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 می‌باشد. با پایه M می‌توان نوع عملیات را مشخص کرد. اگر M='۱' باشد، عملیات منطقی و اگر M='۰' باشد، عملیات ریاضی خواهد بود. با استفاده از پایه های S۰-S۳، می‌توان یکی از ١٦ عملیات منطقی و یا ریاضی مختلف را انتخاب کرد. در جدول (۲-۱۹) انواع مختلف عملیاتهای منطقی و ریاضی با توجه به ورودی‌های S۰-S۳ و پایه M نشان داده شده است. پایه C_n مربوط به بیت نقلی ورودی و $C_{n+٤}$ بیت نقلی خروجی را نشان می‌دهد. پایه (A=B) هنگامی که تمام بیت‌های خروجی '۱' باشند، به حالت امپدانس بالا می‌رود ودر غیر اینصورت '۰' می‌باشد. از این بیت می‌توان برای مشخص کردن مساوی بودن دو عملوند A و B هنگام انجام عمل تفریق استفاده کرد. برای مقایسه بیش از ٤ بیت، می‌توان از گیت AND سیمی[١٤] استفاده کرد. با استفاده از خروجی سیگنال A=B و $C_{n+٤}$ می‌توان حالتهای A>B و یا A<B را آشکار کرد.

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

جدول (۲-۱۹). نحوه عمل آی‌سی ٧٤LS١٨١ به ازای مقادیر مختلف ورودی‌های انتخاب

[١٤] -Wired-And

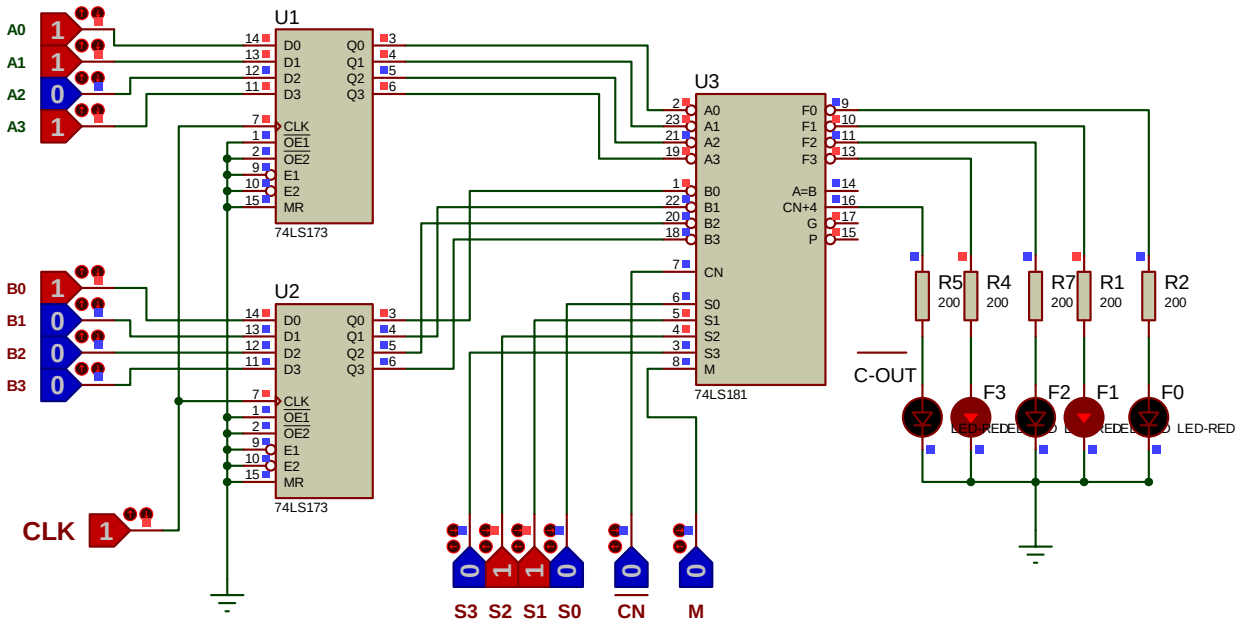
FUNCTION TABLE (ACTIVE HIGH)

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

نحوه 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 یاداشت نمایید.



شکل (۲–۲۱). یک واحد ALU چهاربیتی که طبق جدول (۲–۱۷) رفتار می‌کند.

جدول (۲–۲۰). یافتن خروجی‌های 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	$S_۳ S_۲ S_۱ S_۰$	$F_۳ F_۲ F_۱ F_۰$	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.

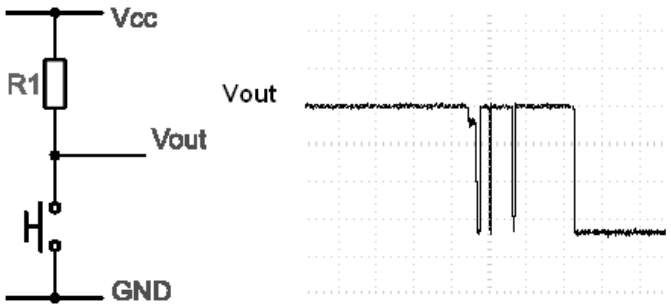
۲– همانطور که اشاره شد، برای مقایسه بیش از ٤ بیت، می‌توان از AND سیمی خروجی پایه های (A=B) استفاده کرد. فرض کنید می‌خواهیم دو عدد ۸ بیتی را با یکدیگر مقایسه کنیم. با استفاده از AND سیمی و یک LED مداری طراحی کنید که در صورت مساوی بودن این دو عملوند ۸ بیتی، LED روشن شود.

آزمایش ۷-الف

موضوع آزمایش: طراحی مدار نوسانگیر[۱۵]

مباحث تئوری

صفحه کلید از سوئیچ بمنظور اعمال تغییردر جریان مربوط به مدارات صفحه کلید استفاده می‌نماید. زمانی‌که 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.

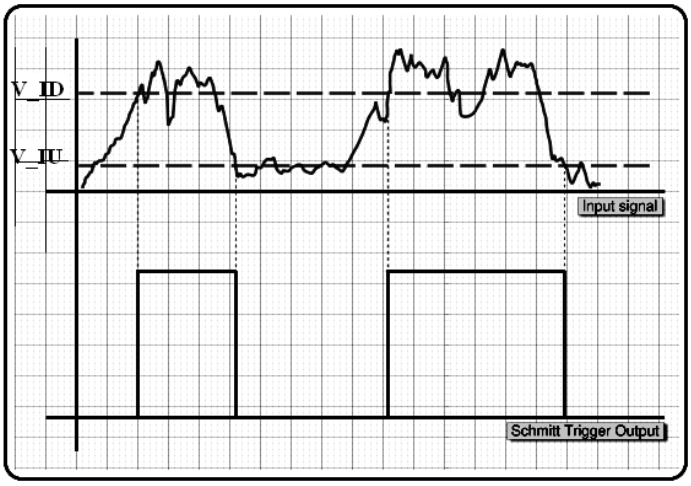


شکل (۲-۲۲): Bounce در کلیدهای مکانیکی

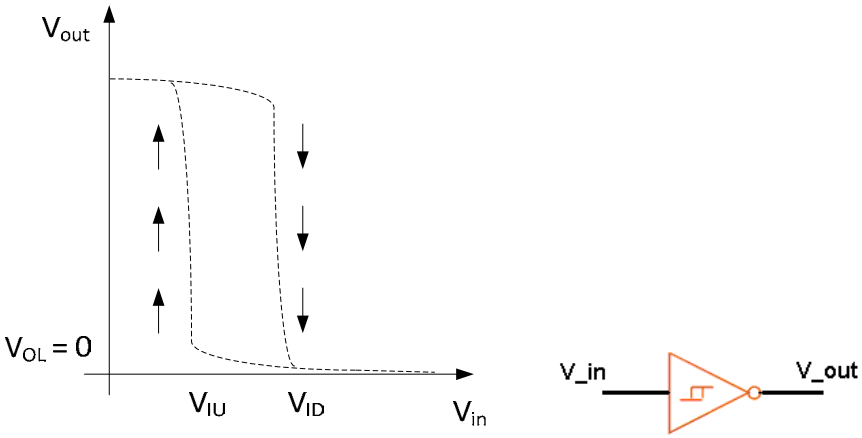
یکی از روشهای حذف لرزش در کلیدهای مکانیکی، استفاده از گیت اشمیت تریگر می‌باشد. در شکل (۲-۲۳) یک موج ورودی و شکل موج خروجی حاصل از یک معکوس کننده اشمیت تریگر نشان داده شده است. در بررسی این شکل موج ها مشاهده می‌شود که انتقال بین خروجی از بالا به پایین زمانی اتفاق می‌افتد که مقدار ورودی از ولتاژ V_{IU} کمتر شود. همچنین انتقال ولتاژ خروجی از حالت پایین به بالا به ازای V_{ID} صورت می‌گیرد. در شکل (۲-۲٤) مشخصه یک گیت اشمیت تریگر نشان داده

[۱۵] -Debouncer

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.



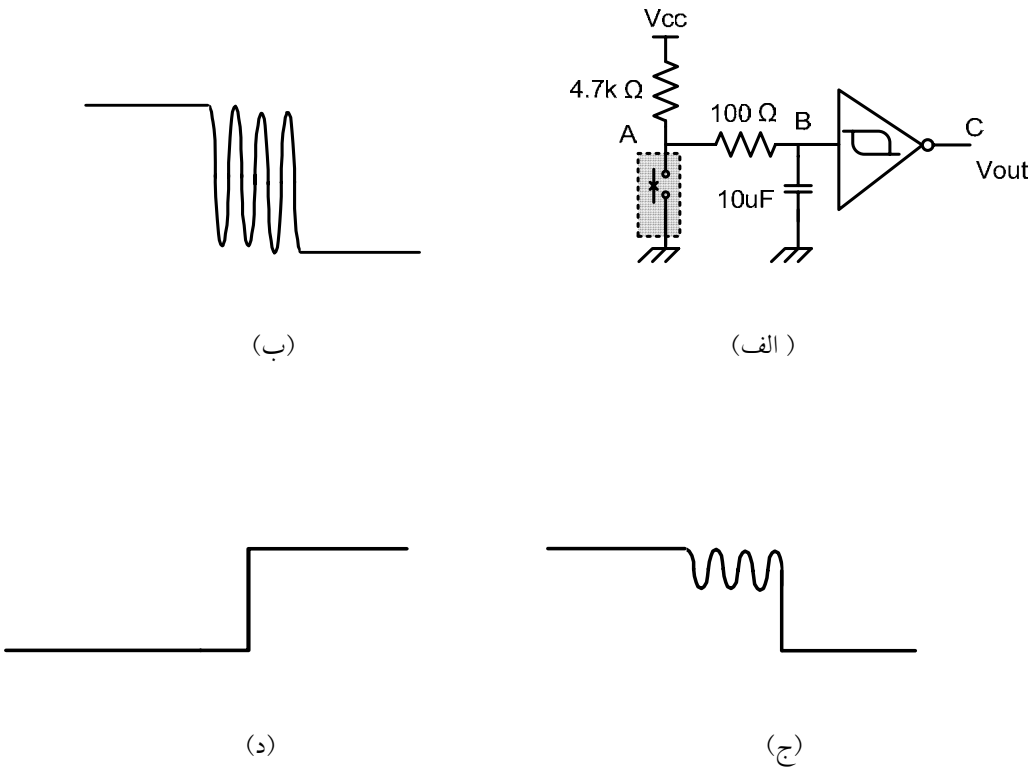
شکل (۲-۲۳)



شکل (۲-۲٤): نماد یک گیت معکوس کننده اشمیت تریگر و مشخصه آن

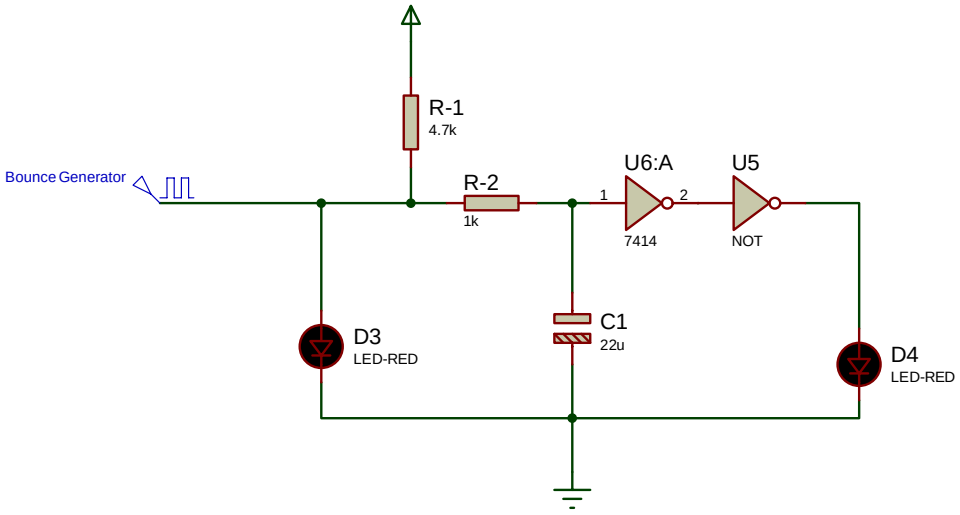
یکی از مدارهایی که می‌توان برای حذف لرزش استفاده کرد، در شکل (۲-۲۵) نشان داده شده است. هنگامی که دکمه را فشار می‌دهیم، طبق معمول در نقطه A لرزش داریم. در اثر این لرزش، خازن به طور پیاپی شارژ و دشارژ می‌شود. اما توجه کنید

که این خازن باعث می‌شود تا نسبت تغییرات در ولتاژ نقطه B کمتر از نقطه A شود که علت آن زمان نسبتا طولانی بـرای شـارژ و دشارژ آن می‌باشد. همانطور که در شکل (۲-۲۵) نشان داده شده است، لرزش بعد از عبور از گیت اشمیت تریگـر و در نقطـه C تقریبا به صورت کامل از بین می‌رود. از بین رفتن لرزشهای نقطه B به علت خواصیت هیسترزیس گیت اشمیت تریگر می‌باشد.



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 ایجـاد کنـیم. بـرای تولیـد نـویز از DCLOCK بـا فرکانس Hz ۲۰ استفاده شده است. اگر مدار شکل (۲-۲۶) را شبیه سازی شده کنید، مشـاهده مـی‌کنیـد LED قـرار داده شـده در ورودی به صورت پراکنده روشن و خاموش می‌شود اما LED قرار داده شده در خروجی روشن نمی‌شود، کـه بـه معنـای حـذف Bounce در خروجی می‌باشد. در این مدار کافی است NOT اول از نوع اشمیت تریگر باشد و NOT دوم می‌تواند معمولی باشد.



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

روش انجام آزمایش

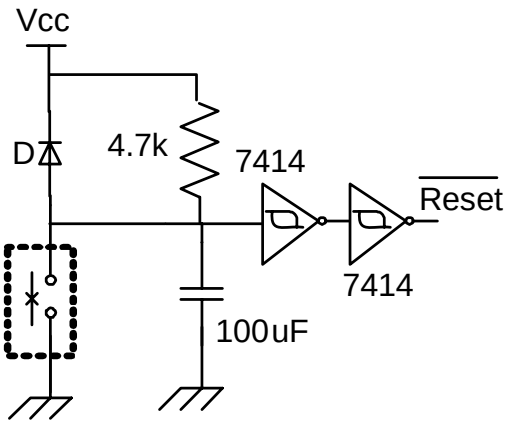
آی سی ۷٤۱٤ دارای ٦ گیت NOT اشمیت تریگر می‌باشد. پس از مطالعه مشخصات آن از دیتابوک، مدار طراحی شده در شکل (۲–۲۶) را برای حذف لرزش بر روی بردبورد بسته و با اعمال یک سیگنال مربعی بـا فرکـانس نسـبتا پـایین در ورودی، شکل موج خروجی را توسط LED یا اسیلوسکوپ مشاهده کنید.

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 Z۸۰، ۰۰۰۰H است. هر سیستم دیجیتال ترتیبی، پس از اعمال تغذیه معمولا شرایط غیر مشخصی در رجیسترها یا فلیپ فلاپهای خود دارد. به همین جهت و برای تعیین یک وضعیت مناسب برای شروع بکار میکروپروسسور، معمولا پس از اعمال تغذیه، میکروپروسسور باید به وضعیت Reset برده شود. این کار با استفاده از مدار شکل (۲-۲۷) امکان پذیر است.



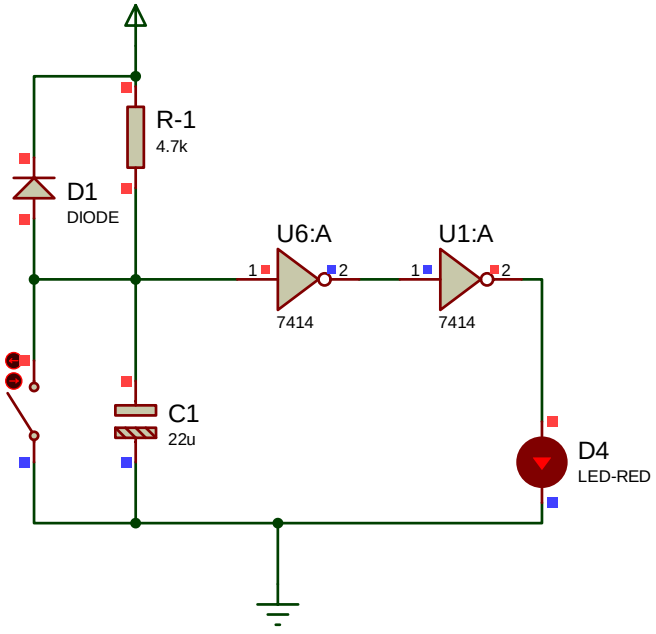
شکل (۲-۲۷) : یک نمونه مدار Reset

به این ترتیب، مدار تا زمان رسیدن مقدار شارژ خازن به یک حالت پایدار، در وضعیت Reset باقی می‌ماند و سپس به میکروپروسسور اجازه شروع به کار داده می‌شود. با فشار دادن سوئیچ، خازن C شروع به تخلیه می‌کند 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 (۲–۲۷) 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 ۷٤۱٤ از دیتابوک، مدار شکل (۲–۲۸) 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 (۲–۲۸) : یک مدار Reset ساده با استفاده از آی‌سی ۷٤۱٤

آزمایش ۸

موضوع آزمایش: تولید دنباله های زمانی

مقدمه

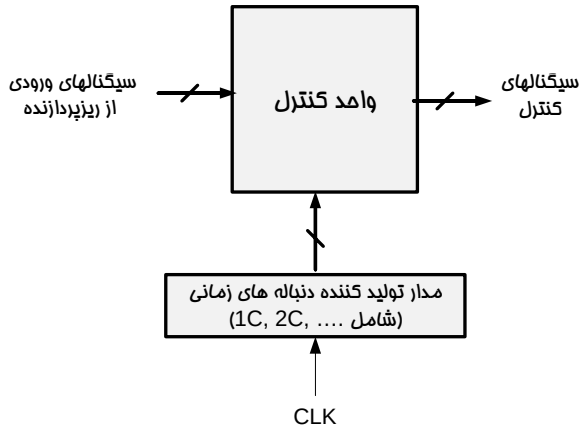
در هر میکروپروسسور، مجموعه‌ای از دستورالعملها که برنامه نامیده می‌شوند، در حافظه خارجی قرار دارد که باید یک به یک توسط میکروپروسسور فراخوانده[۱۶] و سپس اجرا[۱۷] شوند. به مجموع مراحل فراخوانی و اجرای هر دستورالعمل، اصطلاحا سیکل دستورالعمل گفته می‌شود. در هر میکروپروسسور، سیکل دستورالعمل می‌تواند به یک یا چند بخش کوچکتر تقسیم شود که اصطلاحا سیکل ماشین[۱۸] خوانده می‌شود. هر سیکل ماشین عموما 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 $\bar{C}$ 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 $۱C, ۲C, ۳C, ٤C$. 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 می آورد. 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، وحد کنترل مجددا شرایطی را فراهم می‌نماید تا میکروپروسسور آماده خواندن دستورالعمل بعدی از حافظه باشد.

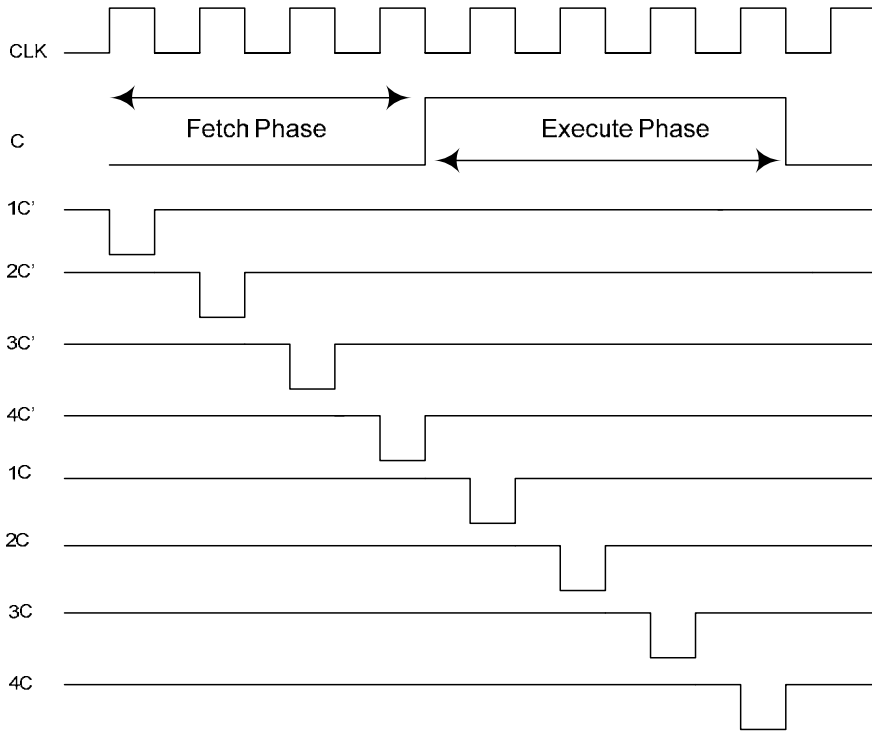
---

[۱۶] Fetch
[۱۷] Execute
[۱۸] Machine Cycle
[۱۹] Clock Cycle

همانطور که در شکل (۲-۲۹) مشاهده می‌کنید، برای تولید سیگنالهای کنترل، نیاز به تولید دنباله های زمانی می‌باشد کـه در واقع موضوع این آزماش می‌باشد.



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

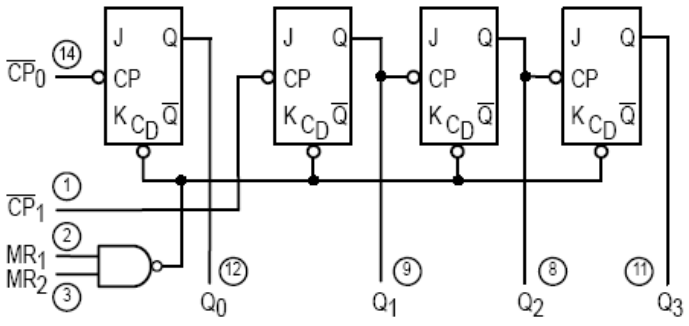


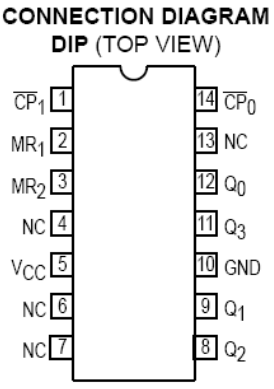
شکل (۲-۳۰). ارتباط زمانی کلاک و سیکل ماشین در فازهای واکشی و اجرا

مباحث تئوری

در شکل (۲-۳۱) یک آی‌سی شمارنده و مدار داخلی آن نشان داده شده است. این شمارنده چهار بیت است و قادر به شمارش اعداد ۰ تا ۱۵ می‌باشد. اساس ساخت مدار تولید کننده دنباله‌های زمانی با استفاده از یک شمارنده ۷٤۹۳ و یک دیکدر ۷٤۱۳۸ در شکل (۲-۳۲) نشان داده شده است. آی سی ۷٤۳۹ یک شمارنده چهاربیتی نشان داده شده است که می‌تواند از ۱۵-۰ را درحالتی که CLK-B به خروجی Q_A متصل می‌شود، شمارش نماید. با هر لبه پایین رونده سیگنال متصل شده به پایه CLK-A، خروجی شمارنده یکی اضافه می‌شود. همانطور که در جدول (۲-۲۲) نشان داده شده است، اگه پایه‌های MR۱ و MR۲ به '۱' متصل شوند، شمارنده بازنشانده 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' می‌کنیم.

آی سی ۷٤LS۱۳۸ یک دیکدر/ مالتی پلکسر می‌باشد. در اینجا برای استفاده از آی‌سی به صورت دیکدر، پایه های E_2 و E_3 باید زمین شوند. همانطور که در شکل (۲-۳۲) نیز نشان داده شده است، خروجی شمارنده به ورودی A, B ,C دیکدر متصل شده است. برای عملکرد آی‌سی بصورت دیکدر، پایه E_1 باید به '۱' متصل شود تا بسته به مقادیر ورودی A, B ,C، یکی از خروجی‌ها صفر و بقیه یک باشند. همانطور که مشاهده می‌کنید، پایه E_1 به خروجی CLK متصل شده است. هنگامی که کلاک به صورت دستی تغییر می‌کند، با '۱' شده خروجی آن، دیکدر نیز فعال شده و پایه مورد نظر را در خروجی صفر می‌کند؛ در صورت صفر بودن E_1، همه پایه های خروجی در حالت '۱' منطقی خواهند بود.



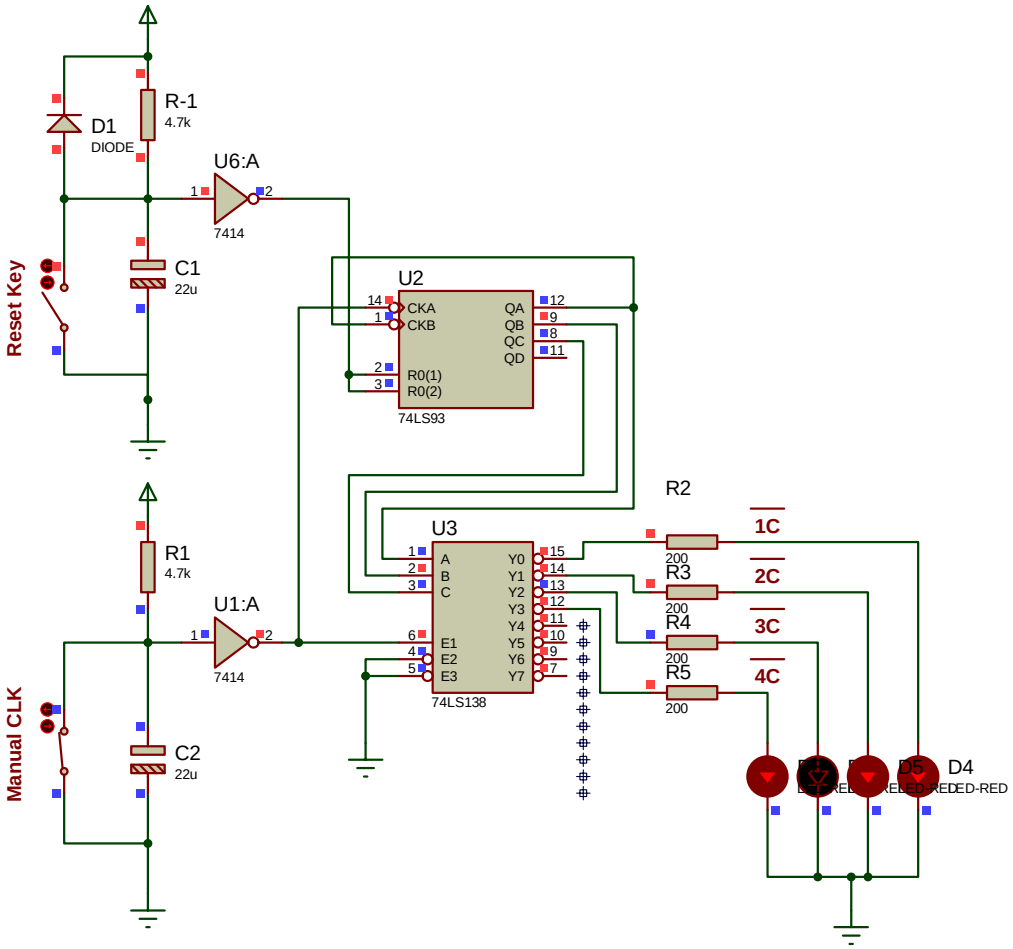


شکل (۲-۳۱): مدار داخلی و شکل آی‌سی شمارنده چهار بیت

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 آی‌سی‌های ۷٤LS۹۳ و ۷٤LS۱۳۸

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، مجددا 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

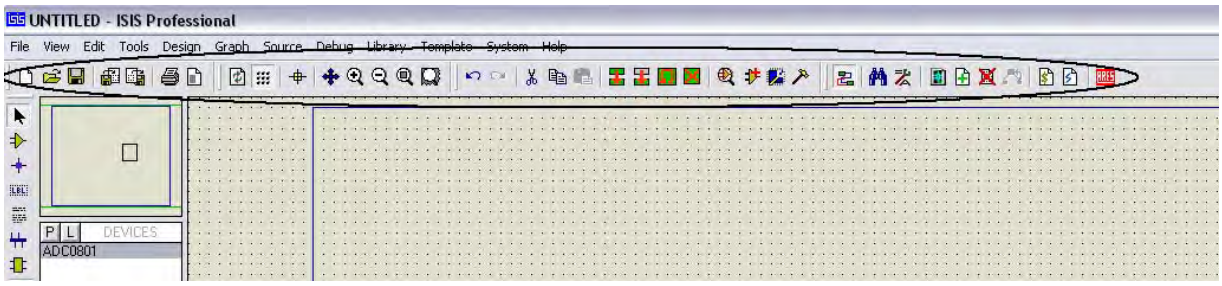
Proteus

مقدمه

نرم افزار Proteus یک شبیه ساز مخصوص ابزارهای الکترونیک است و قادر به شبیه‌سازی میکروکنترلرها، مدارهای مجتمع و مبدل‌ها، مدارات آنالوگ، دیجیتال و ... می‌باشد و به دلیل قدرت بالا و سادگی کار با آن، دارای محبوبیت خاصی در شبیه‌سازی و پیاده سازی مدارات می‌باشد. 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 شده‌اند و مسیری میانبر برای دستیابی به دستورات موجود در منوی File ، Edit می‌باشند را تحت عنوان نوار ابزار دستورات می شناسیم.



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

دراین نوار ابزار، آیکن‌هایی جهت دسترسی سریع و آسان به مواردی فراهم شده است که معمولا زیاد بکار برده می‌شود. تمامی این موارد در داخل منوهای نرم‌افزار نیز موجود و قابل استفاده هستند. بعلت سادگی استفاده، معمولا کاربران ترجیح می‌دهند از این نوار ابزار استفاده نمایند. در ادامه به توضیح بعضی از این آیکن‌ها 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 می‌توان 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 می‌دهیم 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 می‌گذارد 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 می‌دهیم. پنجره جدیدی باز می شود که درجه چرخش را از ما سوال می‌کند.


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


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


Make Device: مجموعه‌ای از قطعات انتخاب شده را به صورت یک بسته در آورده و به کتابخانه اضافه می‌کند.


Packaging Tool: می‌توان با استفاده از این کلید پکیج‌های قطعات موجود را تغییر داد.


Decompose: با فشار دادن این کلید می‌توانیم 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 می‌شود.

Design Explorer: با زدن این کلید پنجره Design Explorer باز می‌شود.

New “Root” Sheet: برای ایجاد یک شیت جدید از این کلید استفاده می‌شود.

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

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

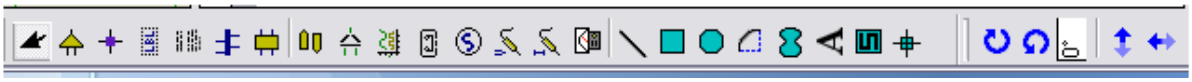
View Bom Report: لیستی از المان ها و توضیحات مدار را نشان می‌دهد.

View Electrical Report: گزارش الکتریکی مدار را نشان می‌دهد.

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 می‌نماید.



شکل (۳-۲). نوار ابزار انتخاب حالت در نرم‌افزار پروتئوس

(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 می‌توان از این کلید هم استفاده کرد.

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

 (Text Scrip Mode) جهت نوشتن متن داخل صفحه می‌توان 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 می‌شود.

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

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

 (Graph Mode) برای استفاده از گراف ها جهت رسم سیگنال های دلخاه مدار از این کلید استفاده می‌شود.

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

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

 (Voltage Probe Mode) پروب ولتاژ.

 (Current Probe Mode) پروب جریان.

 (Virtual Instruments Mode) برای استفاده از تجهیزات مجازی همچون اسکوپ ، سیگنال ژنراتر ، آمپرمتر ، ولـت متـر و غیره باید این کلید فعال باشد.

 (۲D Graphics Line Mode) برای کشیدن خط از این کلید استفاده می‌شود.

 (2D Graphics box Mode) برای کشیدن مستطیل از این کلید می‌توان 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 دلخاه از این کلید استفاده می‌شود.

 (2D Graphics Text Mode) برای نوشتن متن در جای دلخاه صفحه می‌توان 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) می‌توان 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 درجهت پاد ساعتگرد می چرخاند.

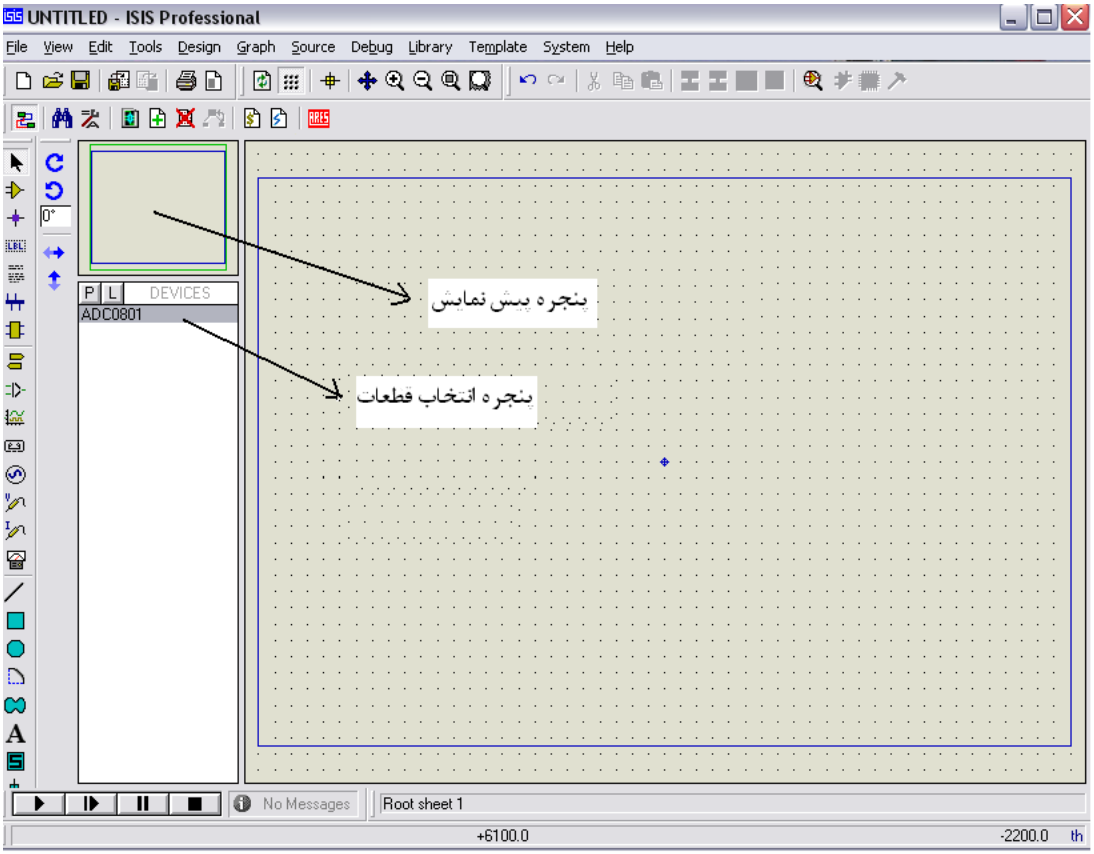
 (X-Mirror) قطعه را نسبت به محور X می چرخاند.

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

 (Editing BOX) این نوار ابزار به شما این امکان را می‌دهد 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 می‌شود که این پنجره از قسمت های مختلفی از جمله نوار ابـزار هـا ، پنجره ویرایش ، پنجره پیش نمایش و Object Selector تشکیل یافته است. پنجره Editing قسمت عمده ایـن پنجـره را بـه خـود اختصاص می‌دهد. این پنجره امکان طراحی، ویرایش و شبیه سازی انواع مـدارهای آنـالوگ و دیجیتـال را در اختیـار کـاربر قـرار می‌دهد. این پنجره با یک Outline (به صورت پیش فرض آبی رنگ) مرزبندی شده است و اگر قطعـه ای خـارج از ایـن Outline قرار گیرد غیر فعال خواهد بود و کابر دیگر، نمی‌تواند قطعه مزبور را انتخاب و ویرایش نماید.



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

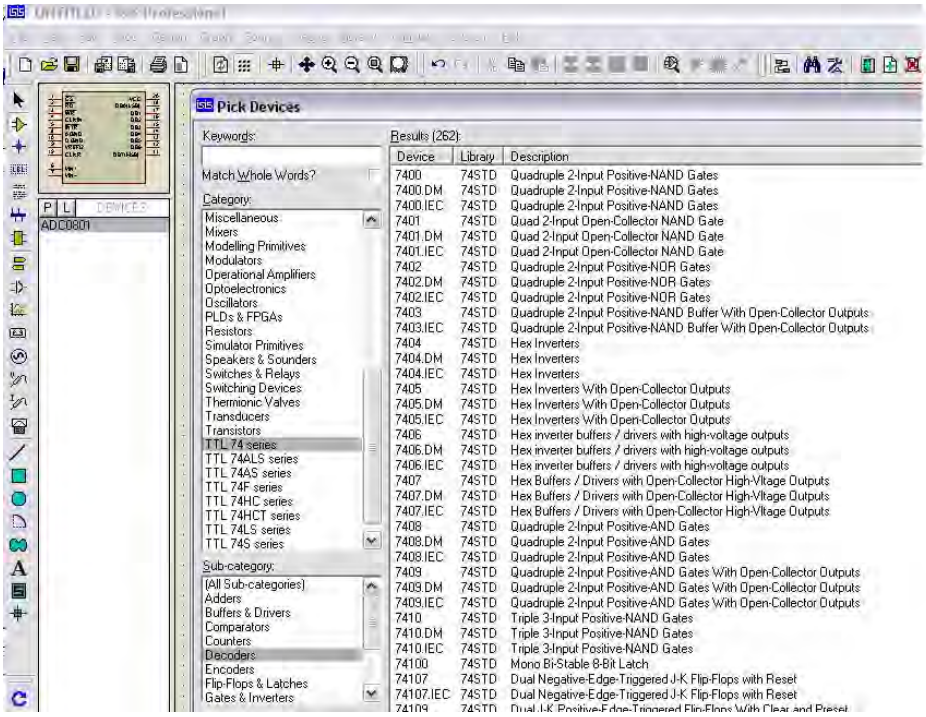
[۲۱] The Editing Window

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(۳-٤). پنجره انتخاب قطعات در نرم‌افزار پروتئوس

برای برقرار کردن اتصالات کافی است تا نوک موس خود را در ابتدا یا انتهای Pinهای 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 نرم‌افزار پروتئوس در هنگام برقراری اتصال

نکته دیگری که در رسم مدار به آن نیاز خواهید داشت، زمین ها و Vccها می‌باشند. برای انتخاب آنها می‌توانید از گزینه Inter-Sheet Terminal استفاده کنید.

شکل(۳-۶). نحوه تعیین زمین و VCC در نرم‌افزار 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(۳-۷). نحوه تغییر مقادیر المان در نرم‌افزار پروتئوس

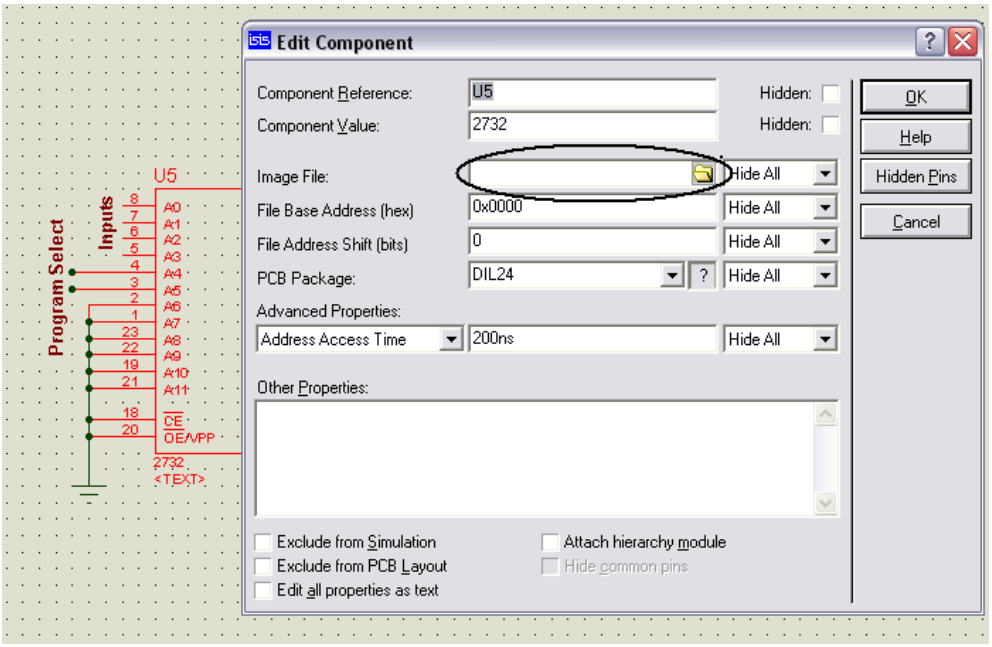
اجرا و شبیه سازی

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(۳-۸). دکمه‌های موجود در نرم‌افزار پروتئوس جهت اجرا و توقف

برنامه ریزی حافظه E^2PROM

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 E^2PROMمیکروکنترلر به Proteus معرفی نمایید. در شکل زیر می‌توانید این گزینه را برای آی‌سی ۷۴۳۲ مشاهده کنید.



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

با استفاده از برنامه EEPROM-TOOL می‌توانید فایل های hex خود را به فرمت بـاینری قابـل اسـتفاده در Proteus تبـدیل نمایید. فرض کنید می‌خواهیم جهار بایت از حافظه با مقادیر نشان داده شده در شکل ۰۰ برنامه ریزی شوند. توجه کنیـد از طریـق دکمه‌های Add و Remove می‌توانید بایت‌هایی را اضافه کرده و یا حذف کنید. در پایان انـدازه کـل حافظـه EEPROM-TOOL را در قسمت Length وارد کرده (به بایت) و در قسمت Other values نیز مقداری را که می‌خواهیـد بـه عنـوان مقـدار پـیش فـرض 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 موردنظر ذخیره نمایید. اکنون می‌توانید این فایل را در Proteus بصورتی که قبلاً توضیح داده شد، انتخـاب نموده و پروژه خود را شبیه‌سازی نمایید.