



دانشگاه آزاد اسلامی واحد زاهدان

دستور کار آزمایشگاه سیستم‌های دیجیتال ۲

گردآوری کننده:

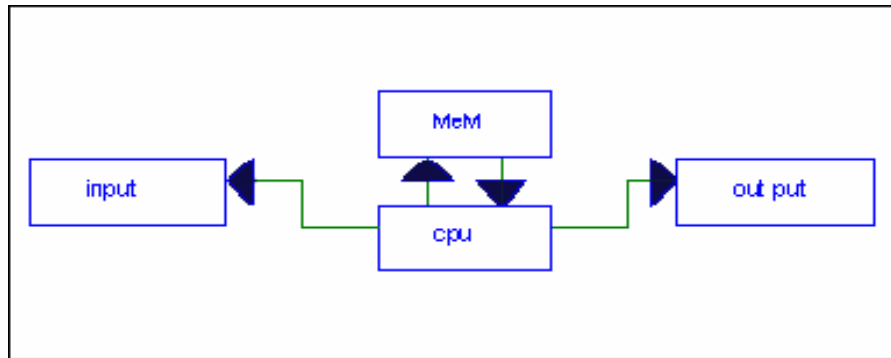
سرکار خانم مهندس جعفریان

دانشجویان کارشناسی برق

آزمایش شماره ۱

عنوان آزمایش: حافظه‌ها

کامپیوتر دیجیتال را میتوان به چهار بخش اساسی تقسیم نمود این شکل چگونگی این تقسیم بندی را نشان می‌دهد.



شکل (۱)

در رابطه با عمل نهایی کامپیوتر هر کدام از بخش‌های فوق دارای یک وظیفه خاص می‌باشند در این رابطه **CPU** قلب سیستم است وظیفه این واحد انجام کلیه عملیات ریاضی و منطقی که توسط یک برنامه تعیین می‌شود، می‌باشد علاوه بر این **CPU** نقش کنترل نهایی یک سیستم کامپیوتری را نیز به عهده دارد و برخی از مولفین **CPU** را به دو بخش **ALU** و کنترل تفکیک می‌نمایند از طرفی واحدهای ورودی و خروجی امکان ارتباط **CPU** را با جهان خارج برقرار می‌سازد واحد ورودی ورود اطلاعات را بر عهده دارد و به عنوان مثال می‌تواند یک صفحه کلید باشد بعد از پردازش داده‌های ورودی نتایج عملیات باید به خارج از کامپیوتر

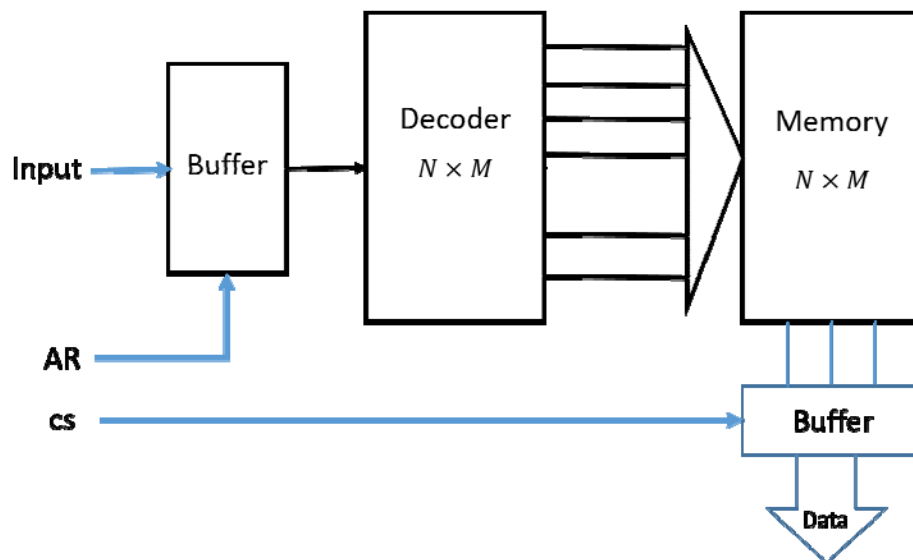
انتقال یابد و این وظیفه بعهدہ واحد خروجی است کہ باز بہ عنوان مثال می تواند یک نمایشگر رقمی و یا یک **CRT** باشد حافظه برای ذخیره کردن اطلاعاتی از قبیل اعداد، اسامی و یا آدرس‌ها و غیره بہ کار می رود. این بخش علاوه بر اینکه اطلاعاتی را جهت پردازش استفاده‌های بعدی حفظ می کند برنامه‌های کامپیوتر برطبق آن بہ پردازش دیتای ورودی می پردازد و نیز مسلماً در همین واحد ذخیره می گردد

در یک سیستم کامپیوتری حافظه‌ها را می توان بہ دو بخش حافظه‌های اولیه و ثانویه تقسیم نمود این دو نوع را می توان حافظه‌های داخلی و جانبی نیز نام نهاد حافظه‌های خارجی برای ذخیره اطلاعات و حجم داده مانند برنامه فایل‌های اطلاعات یا فایل‌هایی کہ دائماً مورد استفاده قرار نمی گیرند بہ کار می روند. از سوی دیگر حافظه‌های داخلی جهت ذخیره سازی موقتی اطلاعات برنامه و یا دیتا بکار می روند بہ عنوان یک مثال هنگامی کہ یک برنامه ی کامپیوتری در حال اجراست دستورالعمل‌ها و دیتا ابتدا از حافظه‌های خارجی و حافظه‌های داخلی منتقل شدہ و پس از پردازش مجدداً بہ حافظه خارجی برمی گردد. از نظر نوع ساختمان می توان حافظه‌ها را بہ دو نوع اساسی نیمه‌هادی و مغناطیسی تقسیم کرد امروزه حافظه‌های خارجی کہ نیاز بہ حجم زیادی دارند توسط حافظه‌های مغناطیسی و حافظه‌های داخلی کہ نیاز بہ سرعت زیادی دارند توسط حافظه‌های نیمه‌هادی ساخته می شوند. در این آزمایش بہ بررسی دو حافظه نیمه‌هادی کہ بہ عنوان حافظه اصلی بہ کار می رود می پردازیم.

حافظه **ROM** (حافظه خواندنی): در این نوع حافظه تنها می توان اطلاعات را کہ در درون حافظه وجود دارد خواند. این اطلاعات توسط سازنده در درون این حافظه قرار گرفته است و یا توسط خود استفاده کننده در درون حافظه قرار می گیرند.

حافظه **RAM** (حافظه خواندنی و نوشتنی): در این نوع حافظه هم می‌توان اطلاعاتی به آن وارد نمود و هم اطلاعات را خواند، به عبارتی این حافظه‌ها را می‌توان به طریقی دلخواه دست‌یابی نمود. حافظه‌های **RAM** به دو نوع استاتیک و دینامیک تقسیم نمود

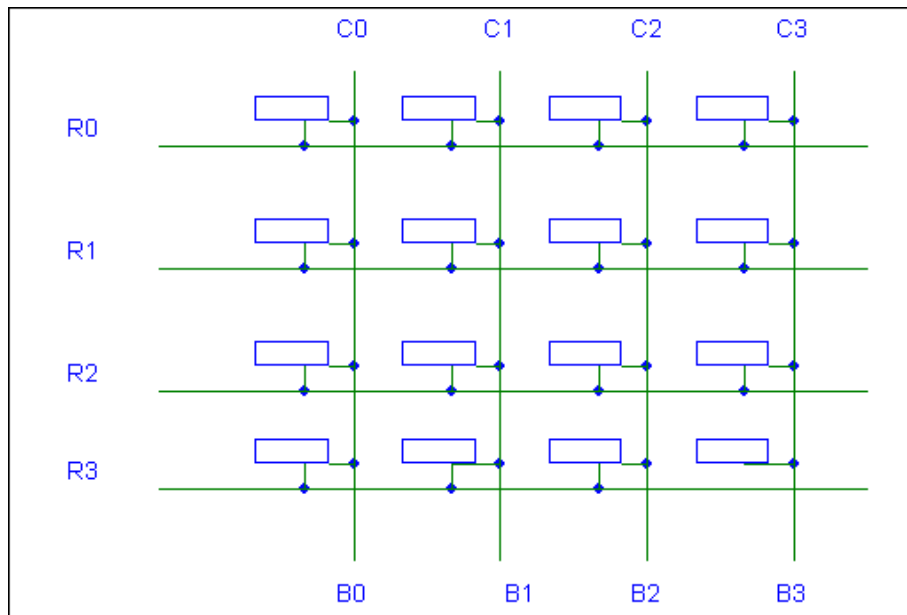
اینجا لازم است تا به بررسی ساختمان داخلی یا به عبارتی بلوک دیاگرام یک حافظه **ROM** بپردازیم. بلوک دیاگرام یک حافظه **ROM** در شکل ۲ نشان داده شده است همان گونه که دیده می‌شود این بلوک دیاگرام بخش‌های بافر آدرس، دیکدر آدرس، آرایه حافظه، بافر خروجی می‌باشد که ذیلاً به تشریح وظیفه هر بلوک می‌پردازیم



شکل (۲)

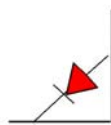
آرایه حافظه:

آرایه حافظه بخشی از حافظه **ROM** است که دیتای تشکیل دهنده برنامه‌ی **ROM** در آن ذخیره می‌شود. این آرایه شامل تعدادی سلول حافظه است که هر کدام قادر به نگهداری یک **BUFFER** می‌باشد. شکل ۳ بخشی از یک آرایه حافظه را نشان می‌دهد.



شکل (۳)

در این آرایه هر کدام از سلول‌ها بایستی قادر باشند یک بیت اطلاعات در خود نگهدارند حال بسته به اینکه کدام سطر توسط دیکدر آدرس انتخاب شده محتوای سلول‌هایی که در این سطر قرار دارد به خروجی منتقل می‌شوند. به عنوان مثال در صورت انتخاب سطر R_0 محتوای سلول‌های 0 تا 3 به خروجی منتقل می‌شوند. در آرایه‌های حافظه‌های ROM به عنوان یک سلول می‌توان از دیود یا ترانزیستور FET استفاده نمود به عنوان مثال یک سلول دیودی



دارای شکل زیر خواهد بود

حال بسته به اینکه به هنگام انتخاب دارای چه طولی باشد میتوان با وصل یا قطع این دیود در محل سلول آن را برای ذخیره 0 یا 1 برنامه‌ریزی کرد.

-بافر آدرس: این بافر برای بافر نمودن سیگنال‌های آدرس و همچنین اعمال کنترل به روی آنها

بکار می‌رود.

-دیکدر آدرس: این واحد، آدرس را دیکد کرده و برای انتخاب محل‌های مورد نظر در حافظه بکار می‌برد.

-بافر خروجی: وظیفه این واحد نیز بافر نمودن اطلاعات خروجی و اعمال کنترل بر آن است.

«کار آزمایشگاهی» ۱

مشخصات **IC** های شماره ۷۴۱۵۴ و ۷۴۱۲۵ را مطالعه و با پایه‌ها و عملکرد آنها آشنا شوید. با استفاده از این **IC** ها یک حافظه **ROM** با ظرفیت ۱۶×۸ بسازید که دارای دو خط کنترل **cs**، **AR** یا **RD** باشد. به عنوان آرایه حافظه از یک ماتریس دیودی استفاده کرده و آن را به گونه‌ای برنامه ریزی نماید که با انتخاب هر سطر آن توسط آدرس ورودی، شماره این سطر بصورت **HEX** در خروجی و بر روی **7. seg** خوانده شود. با دادن آدرس‌های گوناگون به این حافظه از صحت کار خود مطمئن شوید

۲- حافظه‌های **RAM**

همانگونه که گفته شد در حافظه‌های **RAM**، علاوه بر خواندن، قادر به نوشتن اطلاعات در حافظه نیز می‌باشیم.

تمرین: مراحل مختلف برای انجام اعمال خواندن و نوشتن را برای یک حافظه **RAM** در حالت کلی بیان کنید و به همراه گزارش کار تحویل دهید.

«کار آزمایشگاهی» ۲

مشخصات و پایه‌های **IC** های ۲۱۱۴ و ۶۱۱۶ را مطالعه کنید و با استفاده از این **IC** ها حافظه ای ایجاد نمایید که بتوان در آدرس‌های مختلف، مطابق جدول زیر مقادیر داده شده را نوشت.

پس از نوشتن این اطلاعات در محل های مربوطه محتوای آدرس های فوق الذکر را بخوانید.

(در خروجی برای خواندن اعداد می توانید از یک **7. seg** استفاده نمائید).

Address	A5	Bc	F	00	75	0F
Data	23	00	45	56	75	67

کلیه مراحل به همراه گزارش کار تحویل گردد.

عنوان آزمایش: مالتی پلکسرها با استفاده از IC شماره ۷۴۱۵۳.

مقدمات: هدف از انجام این آزمایش آشنایی با برخی کاربردهای مالتی پلکسرها میباشد بعنوان مقدمه ای بر آزمایش چگونگی حل مالتی پلکسرها را مرور نمائید سپس با مشخصات و پایه‌های IC شماره ۷۴۱۵۳ آشنا شوید و در صورت لزوم دیگر گیت‌های منطقی مورد نیاز یک مالتی پلکسر ۸×۱ طراحی نموده و تحقیق کنید که آیا طراحی شما دارای یجدول حقیقت یک مالتی پلکسر ۸×۱ می‌باشد؟

سؤال: چگونه می‌توان با استفاده از یک مالتی پلکسر یک تابع n متغیره منطقی طراحی نمود در گزارش کار شرح دهید

کار آزمایشگاهی ۱- با استفاده از مالتی پلکسر ۴×۱ طراحی شده در قسمت ۱ تابع سه متغیره زیر را طراحی کنید

$$F(A, B, C) = A'B + BC' + AC' + AB'C$$

کار آزمایشگاهی ۲- با استفاده از مالتی پلکسر ۸×۱ طراحی شده در قسمت ۱ تابع چهارمتغیره زیر را طراحی کنید

$$F(A, B, C, D) = \sum (1, 4, 5, 6, 10, 11, 13, 14)$$

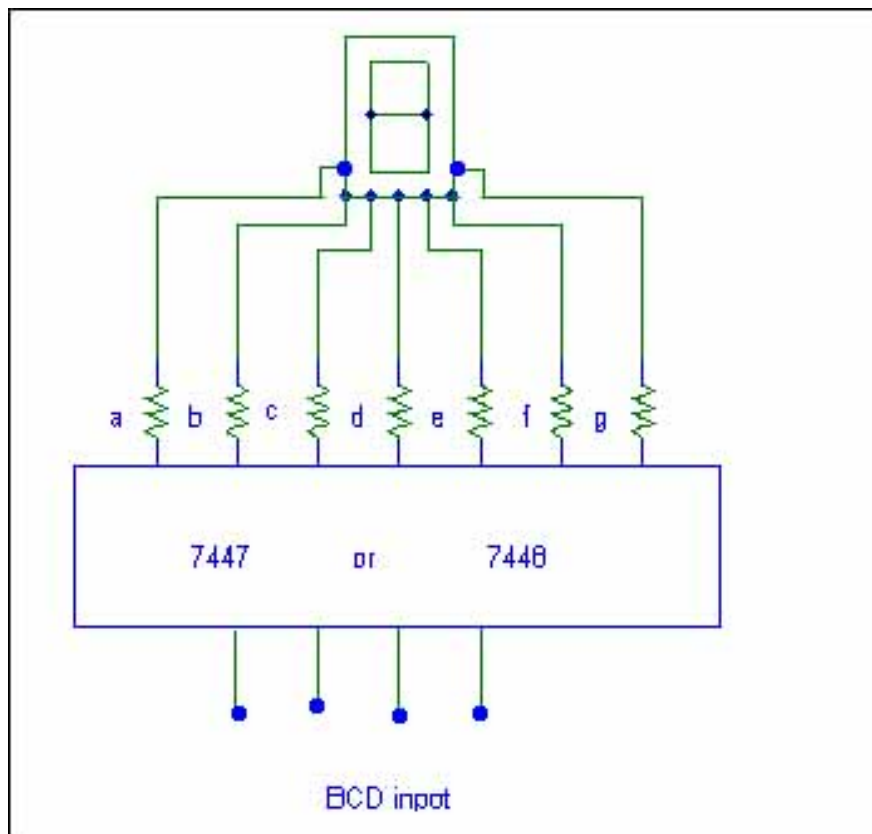
عنوان آزمایش: واحد خروجی

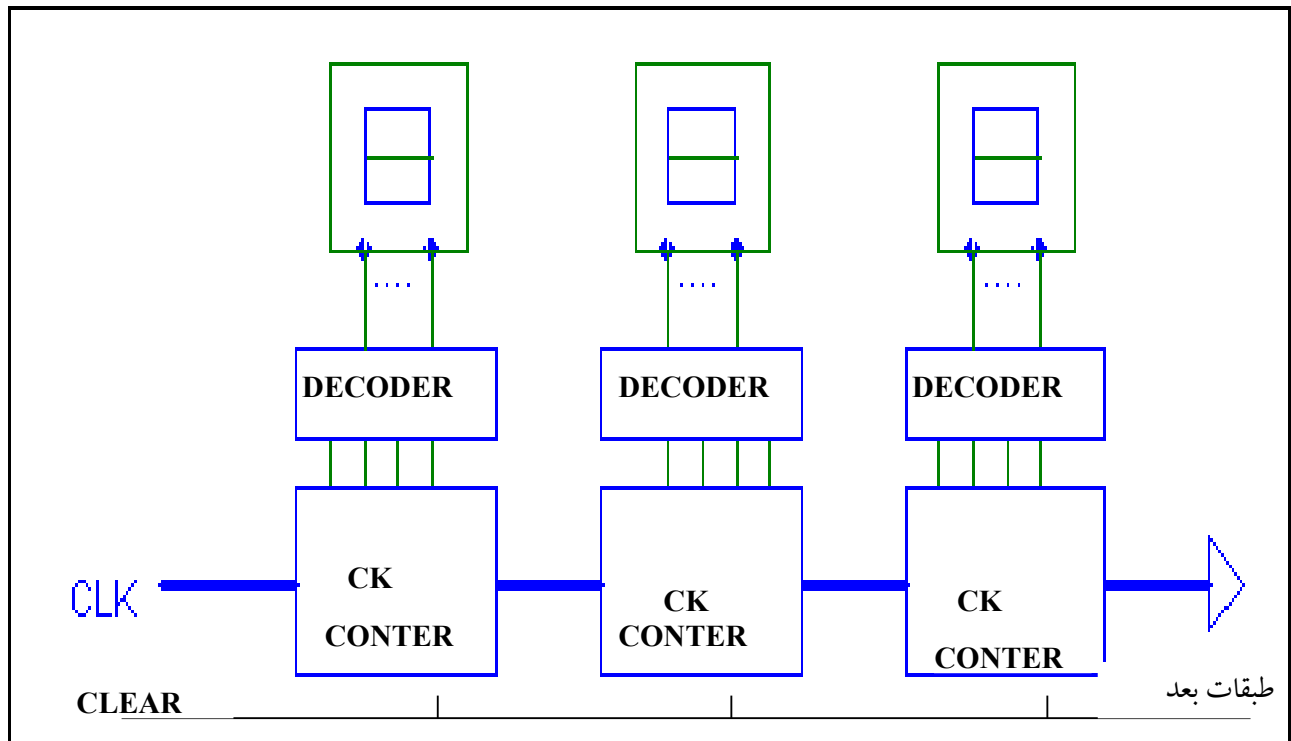
مقدمه

برای انجام یک وظیفه توسط کامپیوتر، بایستی برنامه لازم به همراه دیتای موردنیاز، به کامپیوتر منتقل شود. بعد از اینکه عملیات بر روی آن انجام شده کامپیوتر بایستی این نتیجه را به دیتای خارج منتقل سازد. طرف محاوره کامپیوتر می تواند انسان، دستگاههای اندازه گیری و... باشند. امروزه کامپیوترها قادرند با وسایل و دستگاههای متعددی به عنوان واحد ورودی و خروجی ارتباط برقرار کنند و توسط آنها اطلاعات را با دیتای خارج مبادله نمایند. تعدادی از این دستگاهها عبارتند از: کارت خوان یا نوارخوان به عنوان دستگاه ورودی و وسائلی مانند تله تایپ، CRT، دستگاههای رسام به عنوان خروجی. علاوه بر این در حالت های ساده تری نظیر واحدهای پردازش کوچک از نمایشگرهای رقمی نیز به عنوان واحد خروجی استفاده می شود. البته واضح است که بسته به کاربرد کامپیوتر هدف می تواند ارتباط با سنسورهای مختلف، وسایل صوتی (آلارمها)، موتورهای پله ای، ولتمترها نیز باشد. از آنجایی که نمایشگرهای رقمی می توانند به عنوان واحد خروجی به نحو ساده در عین حال گویایی نحوه انتقال اطلاعات را از کامپیوتر به دنیای خارج و نمایش آن را، نشان دهند. ما نیز در اینجا این را بدین منظور برگزیده و مورد بررسی قرار می دهیم. به طور کلی برای نمایش اطلاعات خروجی از کامپیوتر توسط نمایشگرهای رقمی (مثلاً 7.seg) می توان مدل زیر را ارائه داد:



برای ساختن مبدل BCD راههای متفاوتی وجود دارد که یک روش آن با استفاده از یک دیکدر و یک ماتریس دیودی که در ((آزمایش ۱)) بیان شد. علاوه بر این مدارهای مجتمع IC نیز وجود دارند که عمل هر دو قسمت گفته شده در بالا را به طور کلی انجام می دهند. در شکل زیر یک IC نمونه که این عمل را انجام می دهد و همچنین نحوه اتصال آن را با ورودی ها، نمایشگر رقمی دیده می شود.





در عمل معمولاً جهت نمایش اطلاعات خروجی روی نمایشگرها از روش مالتی پلکسر استفاده می شود (چرا؟) با توجه به شکل شرح لازم را در گزارش کار بیان کنید.

کار آزمایشگاهی ۱

نخست به عنوان مقدمه با مشخصات و پایه های IC های شماره 74153, 7442, 7493, 555 آشنا شوید.

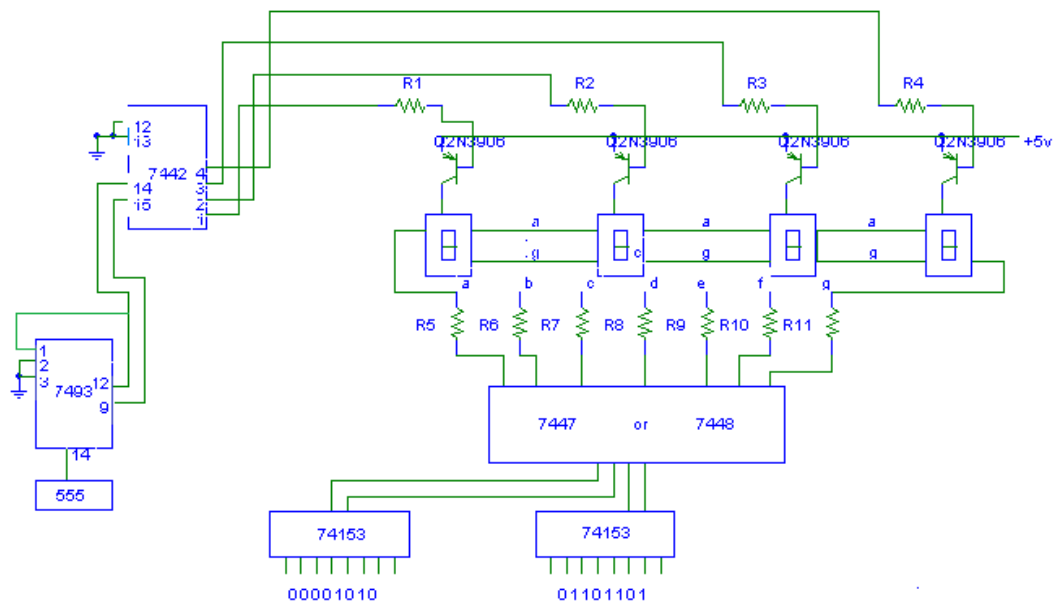
مدار شکل صفحه را که به منظور انتقال چهار رقم BCD بر روی نمایشگرهای 7.seg به روش مالتی پلکسر طراحی شده است تکمیل نمایید. ورودی های نشان داده شده در شکل را (ورودی های 74153) به مدار اعمال نموده و خروجی را بر روی 7.seg مشاهده نمایید. بر روی 7.seg چه عددی را مشاهده می کنید؟ آیا با اعداد داده شده در ورودی مطابقت دارد؟

فرکانس اسیلاتور را تغییر دهید تا این فرکانس به حدود 1KHZ برسد.(بهتر است به عنوان مقاومت لازم برای ساخت اسیلاتور از یک پتانسیومتر استفاده کنید) چه مشاهده می‌کنید؟ حال فرکانس را به آهستگی افزایش دهید تا ارقام به همانگونه ای که می‌خواهیم بر روی خروجی (7.seg) ظاهر شوند.(از این طریق می‌توان به خوبی با نحوه عمل سیستم آشنا شوید). توضیح دهید ارقام ورودی چه ترتیبی و چگونه بر روی 7.seg ظاهر می‌گردند.

ارقام ورودی را به دلخواه تغییر دهید و هربار خروجی را بر روی نمایشگرها مشاهده کنید. آیا به ازاء تمام ارقام ممکن ورودی و خروجی قابل قبولی داریم؟ حداکثر چه عددی می‌توان به ورودی اختصاص داد؟

در صورتیکه ترتیب اعداد ظاهر شده بر روی نمایشگرها برخلاف ترتیب ورودی‌های داده شده باشد، اشکال از کدام یک قسمتهای مدار خواهد بود؟ با آزمایش این مطلب را تحقیق نمائید.
تذکر: در صورتیکه مدار به طور صحیح عمل نکرد، قسمتهای زیر را به ترتیب انجام دهید:
خروجی اسیلاتور را آزمایش کنید.

خروجی‌های ۸ تا ۱۱، شمارنده را بررسی کنید و اطمینان حاصل کنید که از صفر تا ۳ را به خوبی می‌شمارد... خروجی‌های دیکدر را تست کرده، دقت نمائید که در هر لحظه بایستی یکی از خروجی‌های موردنظر High باشد و این عمل به طور متناوب ادامه پیدا کند. (7.seg)ها ترانزیستورها را به ترتیب از نظر انجام عمل سوئیچینگ مورد بررسی قرار دهید. توجه کنید که در خروجی IC شماره 7442 Low اکتیو است و ترانزیستورها از نوع مثبت PNP هستند. عملکرد خواندن اطلاعات و مشاهده آن را در گزارش کار تحویل نمایند.



مربوط به طرح آزمایش واحد خروجی

این طرح باید توسط دانشجو تکمیل گردد

عنوان آزمایش: دستگاه ورودی

همانگونه که قبلاً دیدیم، واحد ورودی یک کامپیوتر می‌تواند شکل‌های مختلفی داشته باشد و صرفنظر از نوع دستگاه ورودی در تمام موارد بایستی علائم ورودی به کامپیوتر به ۰ و ۱ کد شوند. اعداد، علائم و سیگنال‌هایی که به کامپیوتر وارد می‌شوند، غالباً براساس کد Asell و EBCDIC کد می‌شوند که برای کدکردن آنها روش‌های متفاوتی وجود دارد که ما در اینجا به بررسی دو نمونه آن می‌پردازیم (یادآوری می‌کنیم که مدارهای لازم برای این گونه کدکردن اغلب در مدارهای مجتمعی موجود می‌باشد) قابل‌بذکر است که برای استفاده از این کدها عملاً به یک صفحه کلید (Keyboard) نیاز است.

صفحه کلید:

۱- روش ماتریس دیودی

یک روش بسیار ساده برای کد نمودن، استفاده از یک ماتریس دیودی می‌باشد. در این روش هنگامی که یکی از کلیدهای مربوطه زمین می‌شود و از ستون‌هایی که به این کلیدها متصل هستند ولتاژ آن افت نموده و سایر کلیدها در همان سطح ولتاژ اولیه باقی می‌مانند.

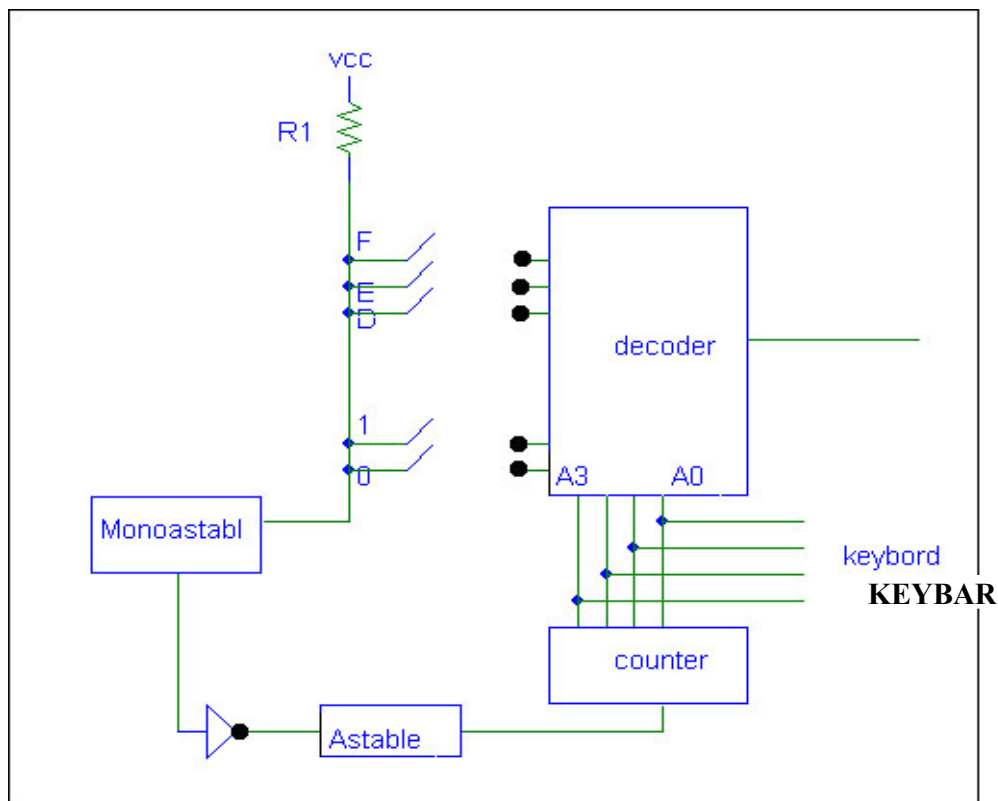
۱- با استفاده از یک ماتریس دیودی، ترکیب لازم برای کد نمودن اعداد ۰ تا F به صورت ASCII را طرح نمائید.

تنها رسم مدار لازم است و انجام آزمایش مربوطه اختیاری می‌باشد.

تمرین ۲- اعداد ۰ تا F را با روش فوق به صورت باینری BINARY کد نمائید.

کار آزمایشگاهی ۱- مدار لازم برای تمرین فوق را بسته و توسط LED خروجی مدار را بررسی نموده و از صحت کار خود مطمئن شوید.

در این قسمت یک روش دیگر که در آن به کمک یک شمارنده یک یکدر می توان یک صفحه کلید ساخت بررسی می شود. قطعات لازم برای انجام این عمل در شکل (۱) دیده می شود.



شکل «۱»

۱- طرز کار مدار فوق را که یک صفحه کلید است تشریح نمایید.

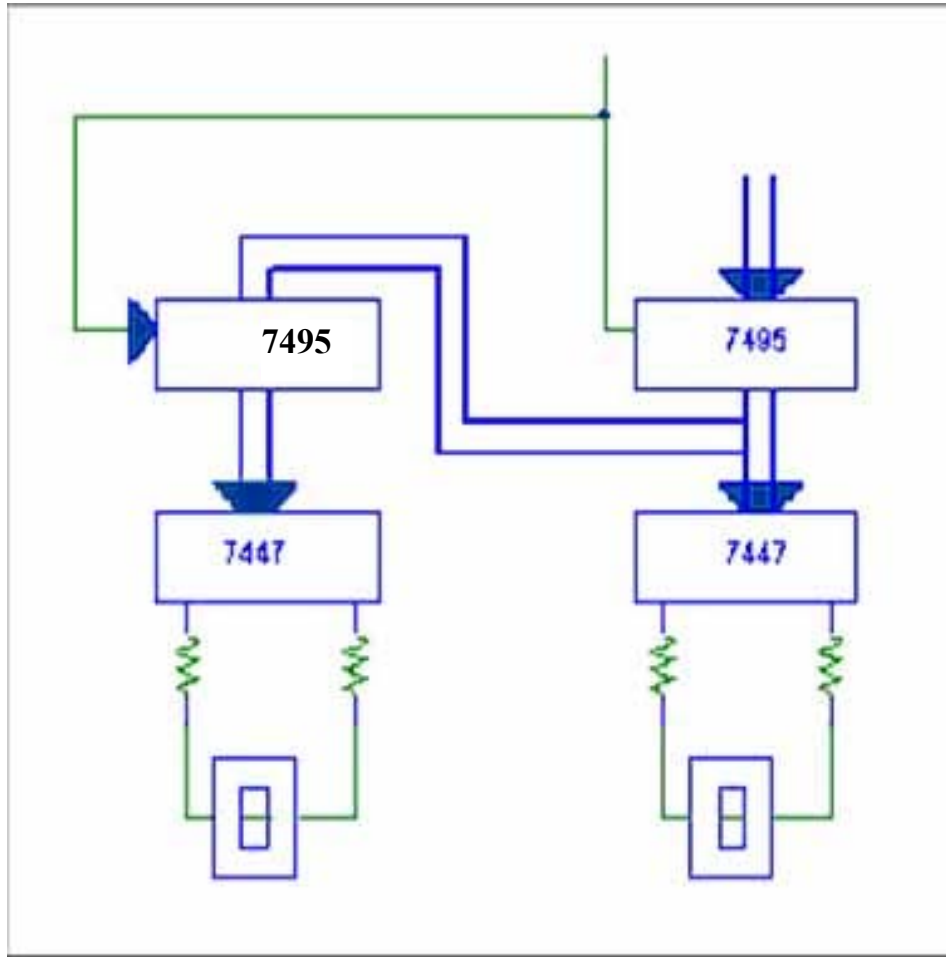
کار آزمایشگاهی ۲

فرکانس آستابل را 2kHz و پهنای پالس منواستابل را 50ms در نظر گرفته و عناصر لازم برای ساخت کد فوق را انتخاب نمائید. سپس مدار طراحی شده را بر روی بردبرد بسته و خروجی آن را بر روی LED مشاهده نمائید.^۱ با وصل کلیدهای مختلف و به صورت تصادفی خروجی را مشاهده نمائید. آیا مدار به طور صحیح کار می‌کند؟ (می‌توانید قبل از انجام این آزمایش به عنوان راهنمایی یک کانتر معمولی را بسته و پالس ورودی آن را با استفاده از یک فانکش ژنراتور ۱ هرتز انتخاب نموده و عمل شمردن را مشاهده نمائید. سپس، پالس را به صورت دستی اعمال نمائید «سعی کنید که فرکانس کار مجدداً همان 1Hz باشد» و این بار نیز صحت کار کانتر را بررسی کنید.) لازم به تذکر است که یک مدار عملی بایستی به ویژه کدیها را حذف نموده تا اطلاعات به فرم صحیحی وارد کامپیوتر شدند. در عمل حذف نویز کدیها به روش‌ها نرم افزاری و سخت افزاری صورت می‌گیرد که در دنباله این آزمایش با روش سخت افزاری آشنا می‌شویم.

کار آزمایشگاهی ۳ (در صورت موجود بودن وسایل انجام خواهد گرفت)

مدار شکل زیر را می‌توان به جای LED در خروجی مدار قبلی جهت وارد نمودن اعداد ۸ بیتی مورد استفاده قرار داد. اساس کار مدار تشریح کرده و پس از تکمیل نمودن مدار آن را به خروجی مدار قبلی متصل نمائید.

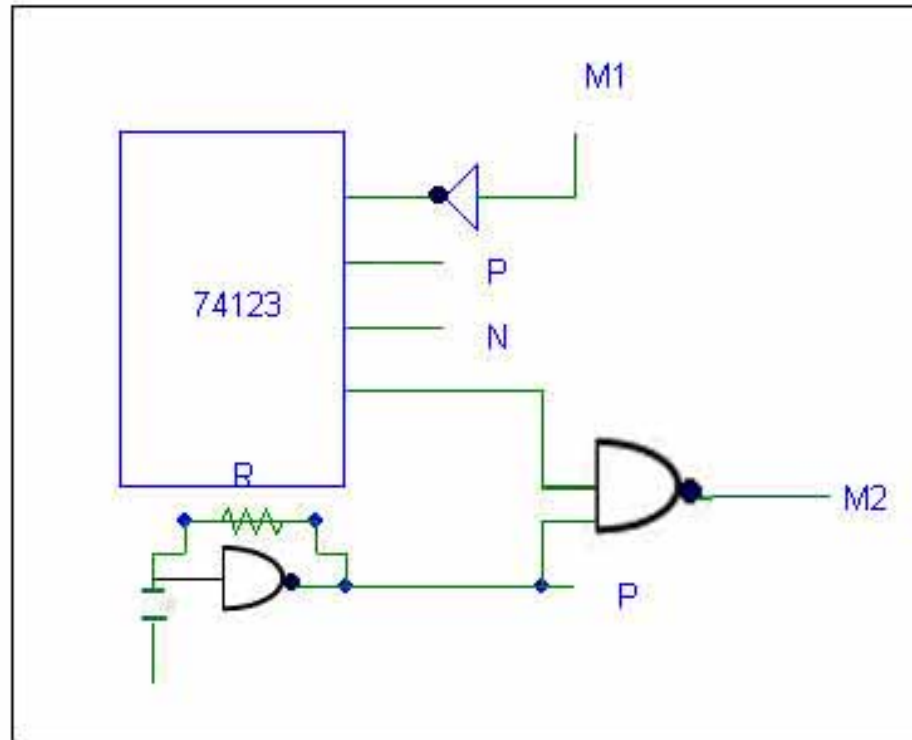
^۱ می‌توانید پس از مشاهده خروجی بر روی LED از 7seg هم استفاده نمائید و خروجی را بر روی 7seg مشاهده کنید.



با تغییری در رابطه با انتقال اطلاعات حاصل گردید؟ (در هر بار فشردن کلید، کلید را مدتی (۱ تا ۲ ثانیه) نگه دارید). اشکال مدار فوق رابطه با قطع و وصل کلید جهت دادن اطلاعات توضیح دهید. عملاً کلیدها هم در موقع قطع و هم در وصل نویز ایجاد می‌نمایند. (یک مدار عملی بایستی قادر به دفع هر دو نوع نویز باشد. آیا می‌توان مدار فوق را برای این منظور به کار برد؟ چگونه؟)

کار آزمایشگاهی ۴- اختیاری

مدار شکل ۱ را از نقاط M_1 و M_2 قطع نموده و به جای آن مدار شکل زیر را قرار دهید: سپس مدار را رسم نموده و طرز کار مدار را در رابطه با حذف نویزها به دقت توضیح دهید:



مدار تکمیل شده را بر روی بردبرد بسته و آزمایشات را تکرار کنید.

عنوان آزمایش: واحد کنترل به روش سخت افزار

با استفاده از IC های (74157,7495,7493)

مقدمه

به طور کلی طرح هر واحد کنترل کامپیوتری به دو صورت امکان پذیر است. یکی از این دو روش طرح واحد کنترل به کمک سخت افزار می باشد. در این روش کوشش می شده تا تمام علائم کنترلی موردنظر به کمک تحولات منطقی در مدارات فیزیکی ساخته می شوند. چنین سیستم هایی از ترکیب واحدها یا مداراتی شبیه شمارنده ها، دیکدرها، انکدرها و... به وجود می آیند.

در این روش وقتی مدار طراحی شده اگر بخواهیم تغییری در سیگنال های کنترلی بدهیم، لازم است مدار موردنیاز دوباره طراحی و ساخته شود. دومین روش طراحی واحد کنترل، که اصطلاحاً میکروپروگرام نامیده می شود، به سیستم کنترلی گفته می شود که برای ساخت و تهیه تمام علائم کنترلی تنها از مدارات منطقی استفاده نکرده باشد. در این گونه سیستم ها معمولاً تمامی یا بخشی از علائم کنترلی به کمک سیکل اطلاعات باینری که در درون یک حافظه و به صورت یک برنامه میکروپروگرام در ROM انبار شده است، تهیه و تدارک دیده می شود. برای طراحی واحد کنترل هر ماشینی در مرحله اول بایستی وضعیت ماشین از نظر اجرا در دستورالعمل فرامین ها روشن گردد. بعد از روشن شدن دستورالعمل های موردنیاز در ماشین می بایست تسلسل میکروکدهای لازم را برای اجرای آن دستورالعمل ها در تشکیلات طراحی شده پیدا کرد و بدین ترتیب در پایان این مرحله هم طرح تشکیلاتی ی هم میکرو کدهای دستورالعمل روشن شده اند. با استفاده از جدول میکروکدها به سادگی می توان معادلات منطقی سیستم را استخراج نمود.

طراحی واحد کنترلی به روش سخت افزار:

در این روش سه متد وجود دارد:

۱- روش استفاده از مدارات ترتیبی

۲- روش استفاده از مدارهای ترتیبی و ترکیبی

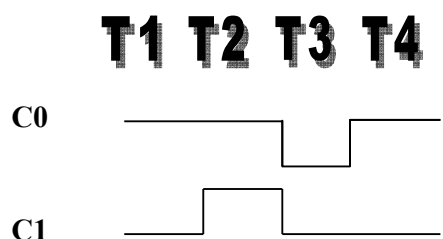
۳- روش استفاده از عناصر تأخیری

روش اول برای واحد کنترل سیستم‌های کوچک به ویژه اگر بتوان سیستم را به واحدهای کوچک قوی تجزیه نمود. روشی ساده و عملی است زیرا همانگونه که می‌دانید در طرح یک سیستم ترتیبی با تعداد زیاد اتصالات ورودی و خروجی، خطوط برگشتی و وضعیت‌ها به سرعت روبه فزونی نهاده و موجب می‌گردد که کنترل و ادغام وضعیت‌ها برای رسیدن به طرح اقتصادی با مشکل مواجه گردد. گرچه این روش سینماتیک ترین روش برای طرح سیستم‌های دیجیتال است ولی به علت اینکه در سیستم‌های بزرگ و متوسط تعداد وضعیت‌های ممکنه، فوق العاده زیاد است، عملاً استفاده از این تکنیک در چنین سیستم‌هایی غیرممکن خواهد بود.

روش دوم که روش استفاده از شمارنده‌ها نیز است، در این روش از پیچیدگی طراحی تا حد زیادی کاسته می‌شود. طراحی می‌تواند هرکجا که مناسب تشخیص داده شود از مدارهای متعدد ترکیبی و ترتیبی استفاده کند. روش جالب در این متد استفاده از شمارنده‌هایی می‌باشد که بعداً در مورد آن توضیح داده خواهد شد.

در روش سوم، برای طراحی واحد کنترل تنها کافی است که روند سیگنال‌های کنترلی را بدانیم و سپس با استفاده از گیت‌های موردنظر توالی این روند را دنبال کنیم. این روش علیرغم سادگی دارای این اشکال است که در آن تعداد حالت عنصر تأخیردهنده لازم است.

برای ایجاد سیگنال‌های کنترلی با استفاده از شمارنده‌ها، با استفاده از خروجی یک شمارنده و مدارهای منطقی لازم می‌توان سیگنال‌های کنترلی را ایجاد نمود. این کار با رسم جدول حقیقت و استفاده از روش‌های (sop , pos , ...) صورت می‌گیرد. به عنوان مثال اگر بخواهیم سیگنال‌های کنترلی مطابق شکل (3a) ایجاد کنیم، باید مدارهایی براساس جدول 3-b طراحی کنیم.



«شکل شماره A»

	X	Y	C0	C1
T1	0	0	1	0
T2	0	1	1	1
T3	1	0	0	0
T4	1	1	1	1

C0, C1: سیگنال‌های خروجی

X, Y: خروجی شمارنده

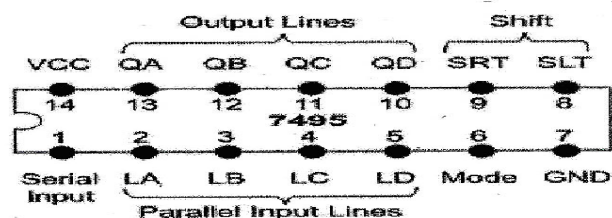
«شکل B1»

کار آزمایشگاهی ۱:

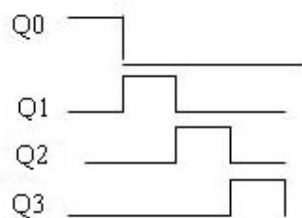
IC 7495 را بدقت بررسی نمایید و جدول صحت آن را در تمامی حالات بدست آورید سپس

این IC را در حالتی که بتواند شیفت به راست یک (1) را انجام دهد بسته و تحقیق کنید که

خروجی‌های آن مانند شکل ۲ باشد.



T1 T2 T3 T4

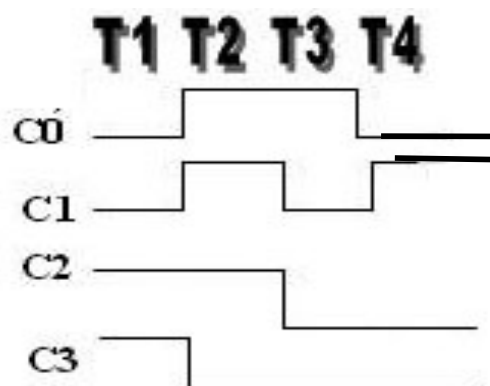


«شکل شماره 2»

حال که توانسته اید شیفت به راست یک (1) منطقی را روی خروجی‌های Q_0 تا Q_1 مشاهده

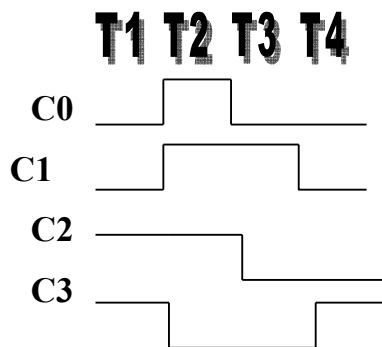
نمایید از خروجی‌های Q_0 تا Q_3 استفاده نموده و با استفاده از گیت‌های NOR یا OR

سیگنال‌های شکل ۳ را ایجاد کنید. سپس مدار موردنظر وابسته و از صحت پاسخ خود مطمئن شوید.



«شکل شماره 3»

در مدار فوق تغییرات لازم را داده و با استفاده (IC7495) مداری ایجاد نمائید به طوریکه اگر سیگنال خارجی مانند $x=0$ باشد، سیگنال‌های C_1 تا C_3 مانند قبل باشند و در غیر این صورت یعنی $x=1$ سیگنال‌های C_0 تا C_3 مطابق شکل شماره ۷ باشد.



«شکل شماره 4»

کار آزمایشگاهی ۲

به کمک IC شمارنده 7493 و یک مدار منطقی ترکیبی سیگنال‌های شکل ۳ را ایجاد کنید. کلیه مراحل را به همراه گزارش کار تحویل نمایید..

جهت ایجاد سیگنالهای قبلی جدول را کامل نموده و با استفاده از گیت‌های منطقی دیگر مدار را

طراحی و ببندید.

QB	QA	C0	C1	C3	C4
0	0				
0	1				
1	0				
1	1				

QB	QA	C ₀	C ₁	C ₂	C ₃
0	0				
0	1				
1	0				
1	1				

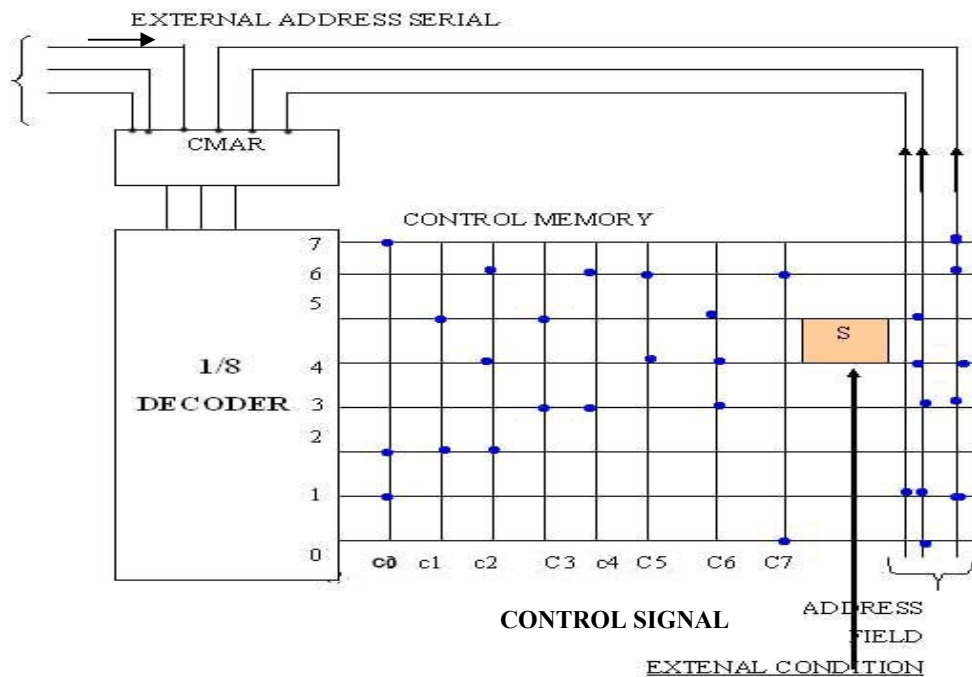
عنوان آزمایش: واحد کنترلی به روش میکروپروگرام

مقدمه:

یکی از روش‌های جالب برای طرح واحد کنترل را می‌توان روش میکروپروگرام دانست. این روش برای اولین توسط M-V- Wilks انگلیسی در سال ۱۹۶۱ پیشنهاد گردید. در این روش معمولاً از یک ROM استفاده می‌شود. در اطلاعات مورد لزوم جهت کنترل سیستم به صورت دسته‌هایی از صفر و یک در آن ذخیره می‌گردد. تعدادی از این اطلاعات ذخیره شده مناسب با نوع دستورالعمل از ROM خارج شده و مستقیماً و یا بعد از تغییر شکل مناسب به قسمت‌های مختلف ماشین اعمال می‌گردد. بدین ترتیب در چنین روشی برای ایجاد علائم کنترل کننده احتیاج به واحدهایی شبیه نوع اول بوده و اصولاً اکثر سیکل‌ها و تسلسل‌ها به صورت کد باینری در یکی از انواع حافظه ROM انبار خواهد شد و سپس به ترتیب ویژه‌ای و به ازاء دستورالعمل‌های مختلف از حافظه خارج خواهند شد.

طرح‌های اولیه Wilkes

- بخش کنترل که خطوط حافظه را فعال می‌کند.
- بخش آدرس که آدرس ریز دستورالعمل بعدی در حافظه کنترل که باید اجراء شود را مشخص می‌کند. در طرح Wilkes هر بیت از حافظه کنترل مربوط به یک خط کنترل بود. شکل ۱ طرح وی را که در سال ۱۹۶۱ پیشنهاد شده نشان می‌دهد. در این شکل حافظه کنترل از نوع ROM می‌باشد. ردیف‌های M مشخص کننده ریز دستورالعمل‌ها و ستون‌های آن مشخص کننده خطوط کنترل و یا آدرس می‌باشند.



«شکل شماره (۱)»

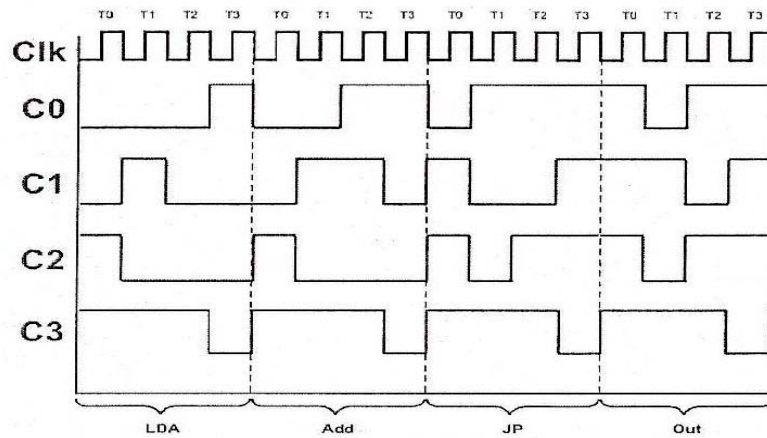
رجیستر آدرس حافظه کنترل CMAR آدرس ریزدستورالعملی را که در حال اجرا است ذخیره می کند. این آدرس دیکد شده و به دنبال آن یکی از سطرهای حافظه فعال می شود و محتوای آن در خروجی قرار می گیرد (نقاط توپر علامت وجود دیود هستند) برای مثال وقتی که آدرس ۱۱۰ در CMAR قرار می گیرد سطر هفتم حافظه فعال شده و به دنبال آن مطابق شکل خطوط فعال می شود و بقیه غیرفعال می گردد. همانگونه که در شکل ۱ مشخص شده است، CMAR در خارج یا وسط بخش آدرس حافظه کنترل تغذیه می گردد. اساساً آدرسی که از خارج داده می شود، آدرس شروع ریزدستورالعمل یا ریزبرنامه را مشخص می کند و به دنبال آن آدرس های بعدی توسط حافظه کنترل تأمین می شود. برای مثال اگر یک CPU که با تکنیک میکروپروگرام طراحی شده است، هر

دستورالعمل با یک میکروپروگرام مشخص می‌شود. بدین ترتیب وقتی که دستورالعمل در رجیستر دستورالعمل (IR) قرار گرفت، قسمت Opcode آن آدرس شروع برنامه را مشخص می‌کند. همانطور که می‌دانید از توانایی‌های دیگری که هر واحد کنترل باید داشته باشد، این است که می‌تواند با توجه به شرایط اعمال شده سیگنال‌های کنترلی مناسب را ارائه دهد که این مسئله را Wilkes در طرح موجود در نظر داشته است. در شکل ۱ حالت S توسط شرایط خارجی کنترل می‌شود و براساس آن آدرس بعدی تعیین می‌گردد. در شکل ۱ پس از ریزدستورالعمل شماره ۴ به آدرس (۰۱۱) یا (۱۰۰) می‌توان رفت و بدین ترتیب یک میکروپروگرام خودپوش نیز می‌توانیم داشته باشیم. برای کار عملی ابتدا یک حالت ساده که در آن حافظه فقط شامل بخش کنترل است را بررسی می‌کنیم. در اینجا لازم است یادآوری شود که در کامپیوترها برای اجرای هر دستورالعمل ابتدا دستورالعمل از حافظه به IR آورده می‌شود و به دنبال آن اجرا می‌گردد. هر دستورالعمل لازم است که برابر چند پالس ساعت باشد. در اینجا سعی بر این است که برای اجرای هر دستورالعمل در واحد کنترل موردنظر ما ۴۰ پالس ساعت لازم باشد. بنابراین هر چهار سؤال حافظه کنترل را به یک دستورالعمل اختصاص می‌دهیم. فرض کنید که خانه‌های حافظه را به صورت زیر به زیردستورالعمل‌ها اختصاص می‌دهیم:

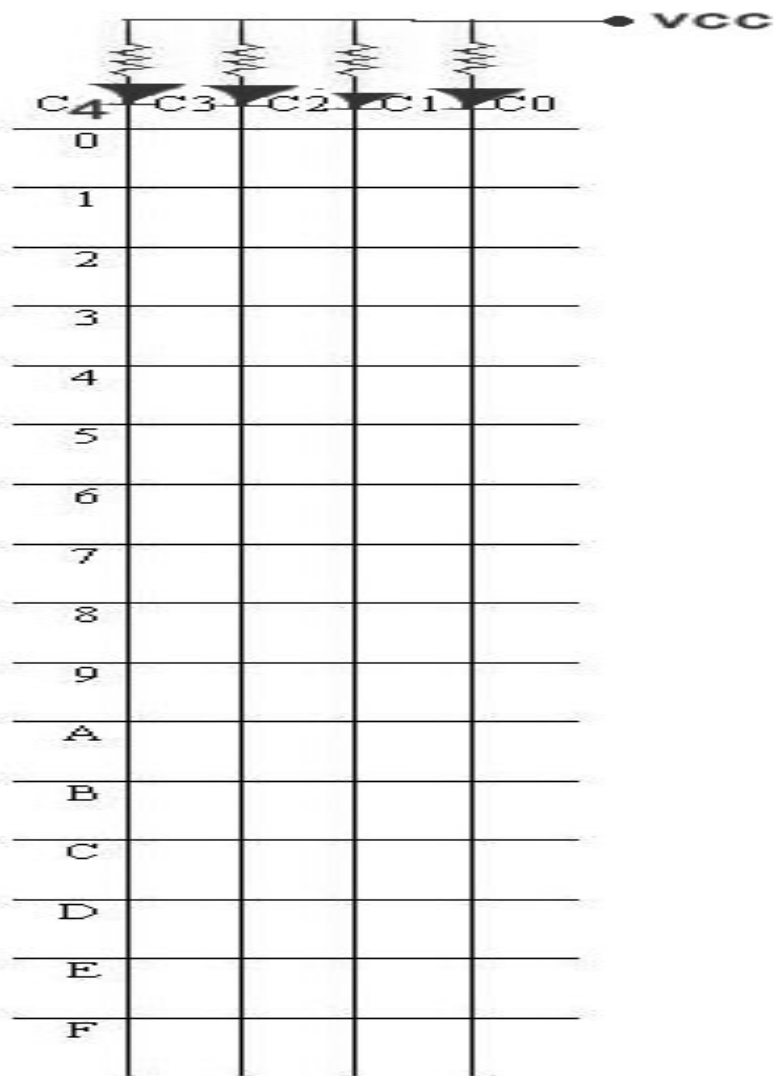
LDA 0000 → 0011 , ADD 0100 → 0111
 JP 1000 → 1011 , OUTA 1100 → 1111

بدین ترتیب وقتی که دستورالعمل LDA صادر می‌گردد. با توجه به Opcode آن رقم 0000 در CMAR قرار می‌گیرد و به دنبال آن سه محل بعدی جارو شده و سیگنال‌های لازم در خروجی ظاهر می‌گردد.

تمرین: اگر سیگنال‌های C_0 تا C_3 برای انجام دستورالعمل‌های فوق به صورت شکل ۲ در نظر گرفته شود. حافظه کنترل شکل ۳ را تکمیل کنید. توجه داشته باشید که دو بیت پردازش دستورالعمل را مشخص می‌کند و با توجه به دو بیت پردازش محتوای چهار محل متوالی حافظه به خروجی منتقل می‌گردد.



شکل (۲)

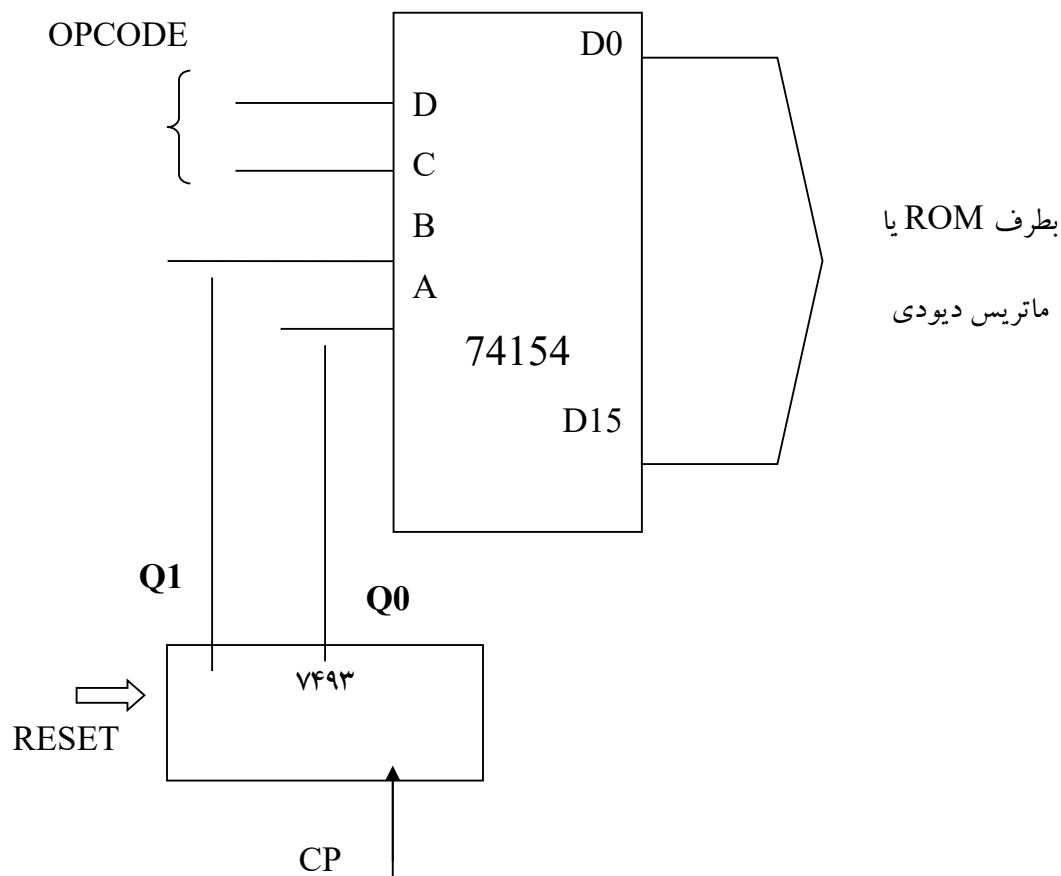


شکل (۳)

تذکر: برای اینکه در خروجی یک داشته باشیم در بیت مربوطه دیود قرار می‌دهیم. (ماتریس دیودی)

کار آزمایشگاهی ۱: برای آزمایش فوق مدار شکل ۴ پیشنهاد شده است. طرز کار مدار را مختصراً توضیح دهید. سپس مدار بسته و opcode مربوط به دستورالعمل LDA را به آن اعمال کنید. خروجی‌های C0 تا C3 را مشاهده نموده تا از صحت کار ماتریس دیودی اطمینان حاصل نمایید.

در مراحل بعدی به ترتیب سه دستورالعمل باقیمانده دیگر را آزمایش کرده و نتایج را با تئوری مقایسه کنید.



شکل (۴)

پس از انجام آزمایش فوق بخش آدرس را به حافظه کنترل می‌افزاییم. در این قسمت آدرس ریزدستورالعمل بعدی اقرار می‌گیرد.

تمرین: در صورتیکه opcodeهای ADD، LDA و outA-JP به ترتیب ۰۰ و ۰۱ و ۱۰ و ۱۱ باشد، حافظه کنترل را مطابق حالت قبلی طوری تکمیل کنید که حداقل حافظه کنترل استفاده گردد. لازم است جهت انجام آزمایشات دیتابوک همراه داشته باشید.

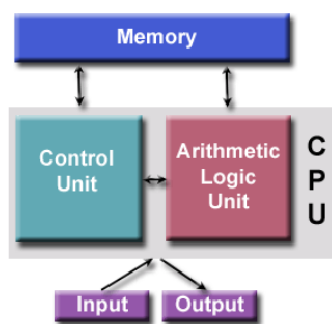
- در هر ترم بسته به نوع امکانات تعدادی آزمایش کم یا زیاد خواهد شد.
- با توجه به علاقه دانشجویان به کار میکروکنترلرها یک آزمایش به عنوان کمک به درس به آزمایشات افزوده خواهد گردید.

آزمایش ۸

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 (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 پیچیده تر باشند:

^{۱۲} Arithmetic Logic Unit

^{۱۳} Control unit

✓ عملیات های ساده:

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

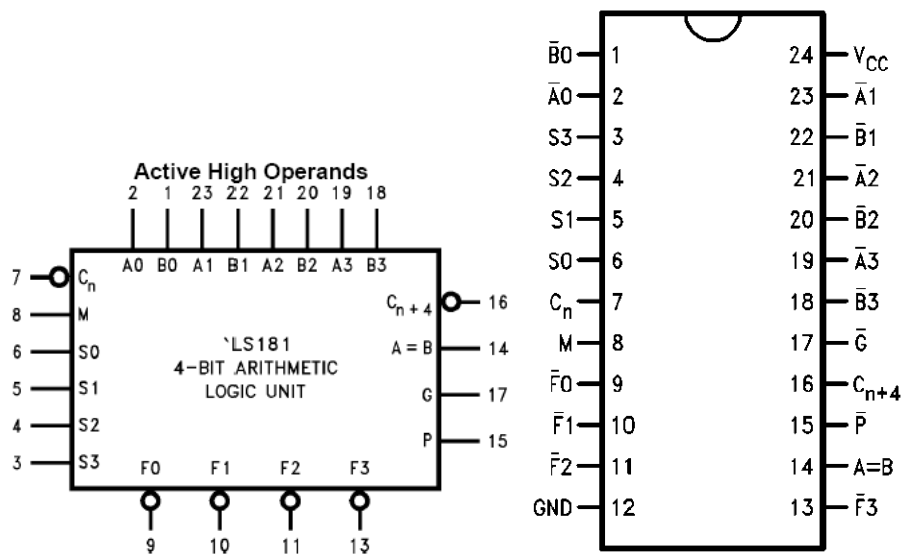
- عملیات های محاسباتی صحیح (جمع ، تفریق و بعضی ضرب و تقسیم ..)
- عملیات های بیتی منطقی (AND ، NOT ، OR ، XOR)
- عملیات های انتقال بیتی (Bit-shifting) (انتقال یا چرخش یک کلمه با تعداد مشخصی بیت به راست یا چپ؛ با/بدون گسترش علامت)

✓ 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 تئوری

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 $\bar{A}^3 - \bar{A}^0$ و $\bar{B}^3 - \bar{B}^0$ و

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

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

Pin Names	Description
$\bar{A}^0 - \bar{A}^3$	Operand Inputs (Active LOW)
$\bar{B}^0 - \bar{B}^3$	Operand Inputs (Active LOW)
S0-S3	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\bar{F}^0 - \bar{F}^3$	Function Outputs (Active LOW)
A = B	Comparator Output
\bar{G}	Carry Generate Output (Active LOW)
\bar{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 $M=1$ باشد، عملیات منطقی و اگر $M=0$ باشد، عملیات ریاضی خواهد بود. با استفاده از پایه های S_0-S_3 ، می‌توان 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 ورودی‌های S_0-S_3 و پایه M نشان داده شده است. پایه C_n مربوط به بیت نقلی ورودی و C_{n+1} بیت نقلی خروجی را نشان می‌دهد. پایه $(A=B)$ هنگامی که تمام بیت‌های خروجی '۱' باشند، به حالت امیدانس بالا می‌رود و در 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 ۴ بیت، می‌توان از گیت AND سیمی^{۱۴} استفاده کرد. با استفاده از خروجی سیگنال $A=B$ و C_{n+1} می‌توان حالت‌های $A > B$ و یا $A < B$ را آشکار کرد.

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

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

^{۱۴} -Wired-And

FUNCTION TABLE (ACTIVE HIGH)

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

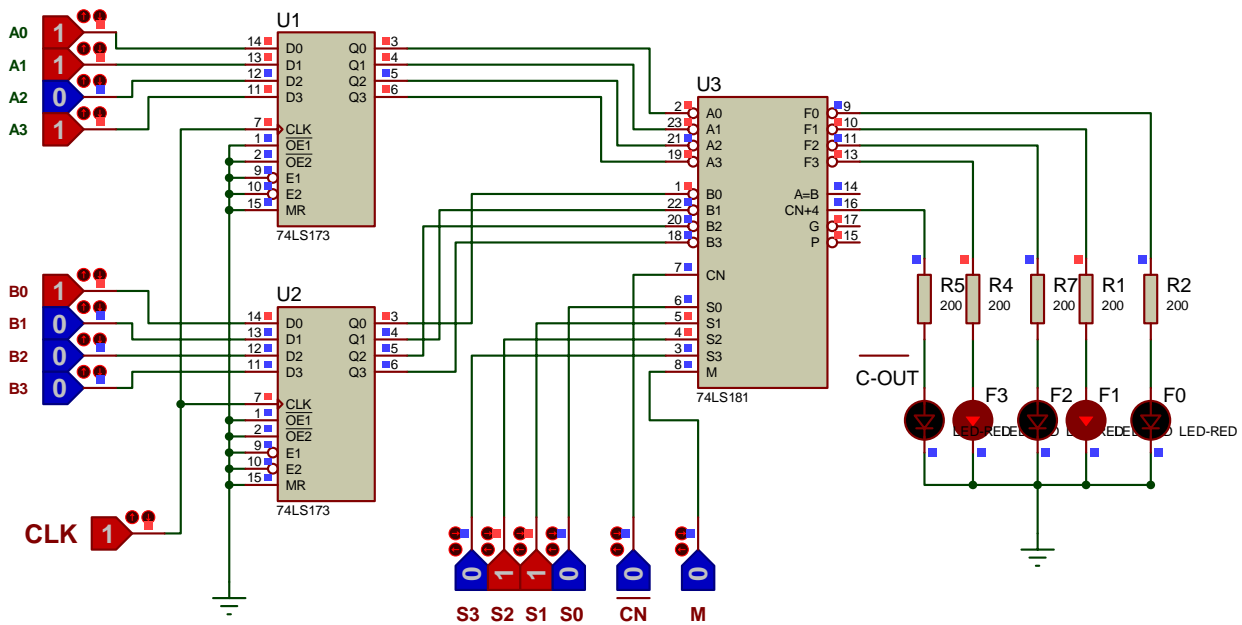
نمونه انجام آزمایش

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 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	$S_3 S_2 S_1 S_0$	$F_3 F_2 F_1 F_0$	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.