دانشگاه آزاد اسلامی واحد زاهدان

# دستور کار آزمایشگاه معماری کامپیوتر

گردآوری کننده:

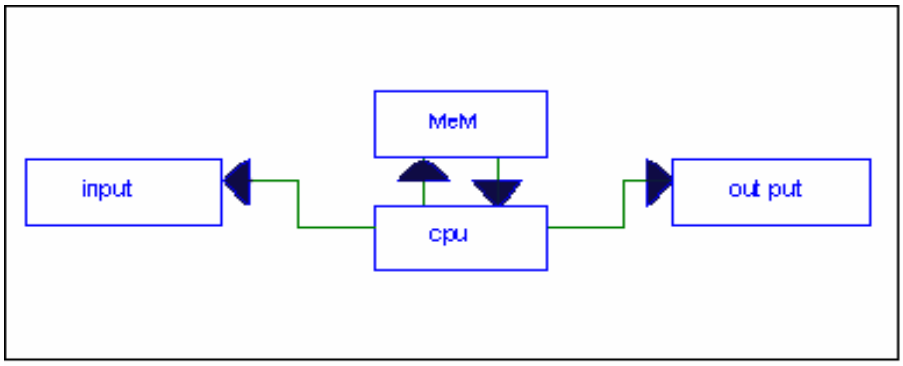
*سرکار خانم مهندس جعفریان*

دانشجویان کارشناسی کامپیوتر

# آزمایش شماره ۱

## عنوان آزمایش: حافظه‌ها

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




شکل(۱)

در رابطه با عمل نهایی کامپیوتر هر کدام از بخش‌های فوق دارای یک وظیفه خاص می باشـند در این رابطه **CPU** قلب سیستم است وظیفه این واحد انجام کلیه عملیـات ریاضـی ومنطقـی که توسط یک برنامه تعیین می شود، می باشد علاوه بر این **CPU** نقـش کنتـرل نهـایی یـک سیستم کامپیوتری را نیز به عهـده دارد و برخـی از مـولفین **CPU** را بـه دو بخـش **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 می‌توانـد یک صفحه کلید باشد بعد از پردازش داده‌های ورودی نتایج عملیات باید به خارج از کامپیوتر

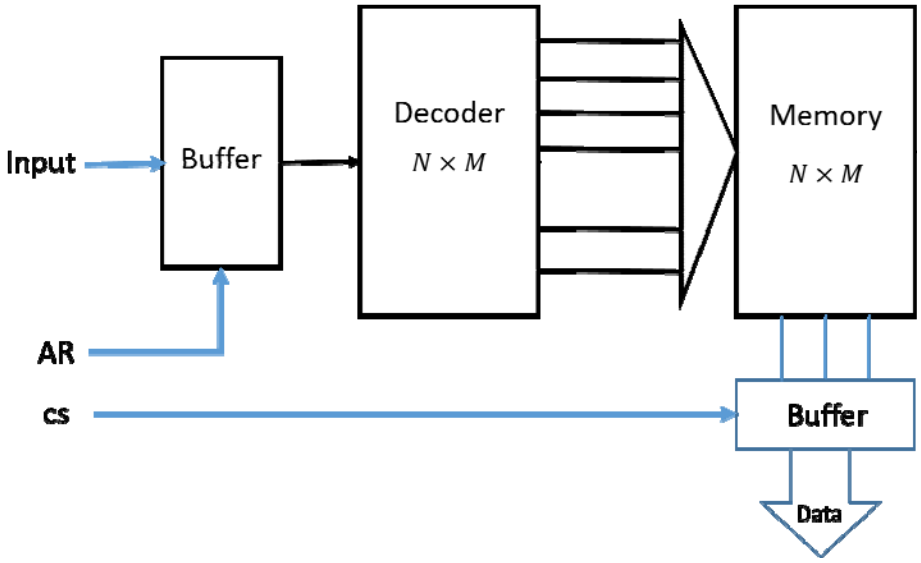
انتقال یابد و این وظیفه بعهده واحد خروجی است که باز به عنوان مثال می‌تواند یک نمایشگر رقمی و یا یک **CRT** باشد حافظه برا ی ذخیره کردن اطلاعاتی از قبیل اعداد، اسامی و یا آدرس‌ها و غیره به کار می‌رود. این بخش علاوه بر اینکه اطلاعاتی را جهت پردازش استفاده‌های بعدی حفظ می‌کند برنامه‌های کامپیوتر برطبق آن به پردازش دیتای ورودی می پردازد و نیز مسلماً در همین واحد ذخیره می‌گردد

در یک سیستم کامپیوتری حافظه‌ها را می‌توان به دو بخش حافظه‌های اولیه و ثانویه تقسیم نمود این دو نوع را می‌توان حافظه‌های داخلی و جانبی نیز نام نهاد حافظه‌های خارجی برای ذخیره اطلاعات و حجم داده مانند برنامه فایل‌های اطلاعات یا فایل‌هایی که دائماً مورد استفاده قرار نمی‌گیرند به کار می‌روند. از سوی دیگر حافظه‌های داخلی جهت ذخیره سازی موقتی اطلاعات برنامه و یا دیتا بکار می‌روند به عنوان یک مثال هنگامی که یک برنامه ی کامپیوتری در حال اجراست دستورالعمل‌ها ودیتا ابتدا از حافظه‌های خارجی و حافظه‌های داخلی منتقل شده وپس از پردازش مجدداً به حافظه خارجی برمی‌گردد. از نظر نوع ساختمان می‌توان حافظه‌ها را به دو نوع اساسی نیمه‌هادی و مغناطیسی تقسیم کرد امروزه حافظه‌های خارجی که نیاز به حجم زیادی دارند توسط حافظه‌های مغناطیسی و حافظه‌های داخلی که نیاز به سرعت زیادی دارند توسط حافظه‌های نیمه‌هادی ساخته می‌شوند. دراین آزمایش به بررسی دو حافظه نیمه‌هادی که به عنوان حافظه اصلی به کار می‌رود می‌پردازیم.

حافظه **ROM** (حافظه خواندنی): در این نوع حافظه تنها می‌توان اطلاعات را که 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 می‌گیرند.

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 دست یابی نمود. حافظه‌های **RAM** به دونوع استاتیک و دینامیک تقسیم نمود

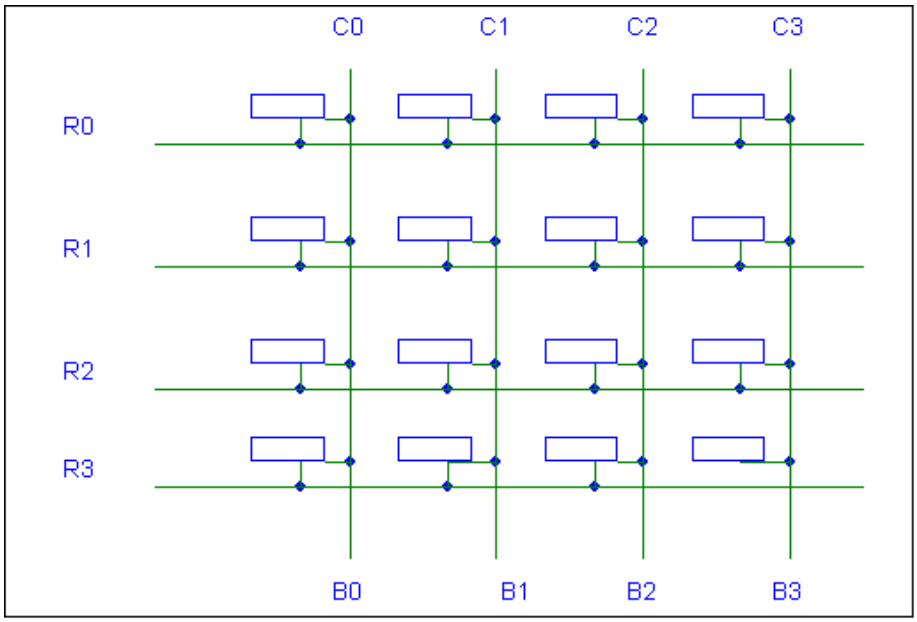
اینجا لازم است تا به بررسی ساختمان داخلی یا به عبارتی بلوک دیاگرام یک حافظه **ROM** بپردازیم. بلوک دیاگرام یک حافظه **ROM** در شکل ۲ نشان داده شده است همان گونه که دیده می‌شود این بلوک دیاگرام بخش‌های بافر آردس، دیکدر آدرس، آرایه حافظه، بافر خروجی می‌باشد که ذیلاً به تشریح وظیفه هر بلوک می‌پردازیم




شکل(۲)

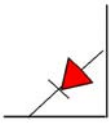
آرایه حافظه:

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




شکل(۳)

در این آرایه هر کدام از سلول‌ها بایستی قادر باشند یک بیت اطلاعات در خود نگهدارد 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 ۰ یا ۱ 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.

-بافرخروجی: وظیفه این واحد نیز بافر نمودن اطلاعات خروجی واعمال کنترل بر آن است.

«کار آزمایشگاهی»۱

مشخصات **IC** های شماره۷٤۱۵٤ و ۷٤۱۲۵ را مطالعه و با پایه‌ها و عملکرد آنها آشنا شوید.

با استفاده از این **IC** ها یک حافظه **ROM** باظرفیت ۸×۱٦ بسازید که دارای دو خط کنترل **AR**، **cs** 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 برا ی یک حافظه **RAM** در حالت کلی بیان کنید و به همراه گزارش کار تحویل دهید.

«کار آزمایشگاهی»۲

مشخصات و پایه‌های **IC** های ۲۱۱٤ و٦۱۱٦ را مطالعه کنید و با استفاده از این **IC** ها حافظه ای ایجاد نمائید که بتوان در آدرس‌های مختلف، مطابق جدول زیر مقادیر داده شده را نوشت.

پس از نوشتن این اطلاعات در محل‌های مربوطه محتوای آدرس‌های فـوق الـذکر را بخوانیـد.

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

| | | | | | | |
|---|---|---|---|---|---|---|
| Addres | A5 | Bc | F | 00 | 75 | 0F |
| Data | 23 | 00 | 45 | 56 | 75 | 67 |

کلیه مراحل به همراه گزارش کار تحویل گردد.

## آزمایش شماره ۲

**عنوان آزمایش: مالتی پلکسرها با استفاده از IC شماره ۷٤۱۵۳.**

**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 ۷٤۱۵۳ آشنا شوید و در صورت لزوم دیگر گیت‌های منطقی مورد نیاز یک مالتی پلکسر ۱×۸ طراحی نموده و تحقیق کنیدکه آیا طراحی شما دارای یجدول حقیقت یک مالتی پلکسر ۱×۸ می‌باشد؟**

**سئوال: چگونه می‌توان با استفاده از یک مالتی پلکسر یک تابع n متغیره منطقی طراحی نمود در گزارش کارشرح دهید**

**کار آزمایشگاهی ۱- با استفاده از مالتی پلکسر ۱×٤ طراحی شده در قسمت ۱ تابع سه متغیر زیر را طراحی کنید**

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

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

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

## آزمایش شماره۳

**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 دونمونه آن می‌پردازیم (یادآوری می‌کنیم که مدارهای لازم برای این گونه کدکردن اغلب در مدارهای مجتمعی موجود می‌باشد) قابل بذکر 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.

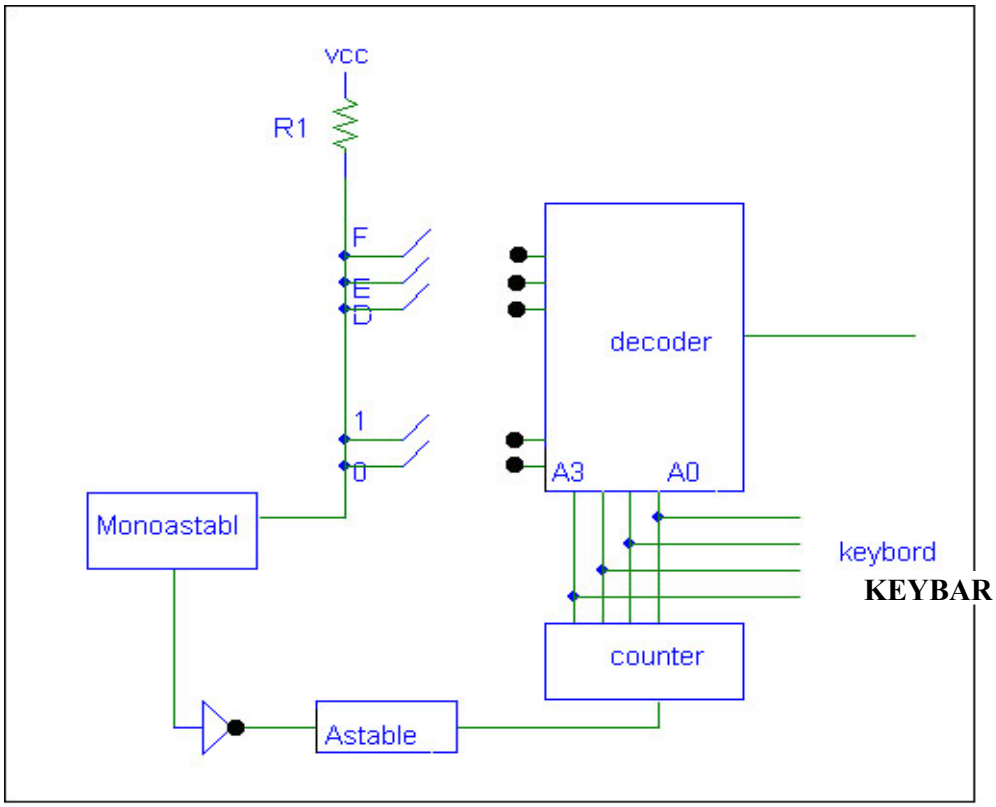
۱- با استفاده از یک ماتریس دیودی، ترکیب لازم برای کد نمودن اعداد ۰ تا F به صورت ASCI را طرح نمائید.

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 می‌توان یک صفحه کلید ساخت بررسی می‌شود. قطعات لازم برای انجام این عمل در شکل (۱) دیده می‌شود.




شکل«۱»

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

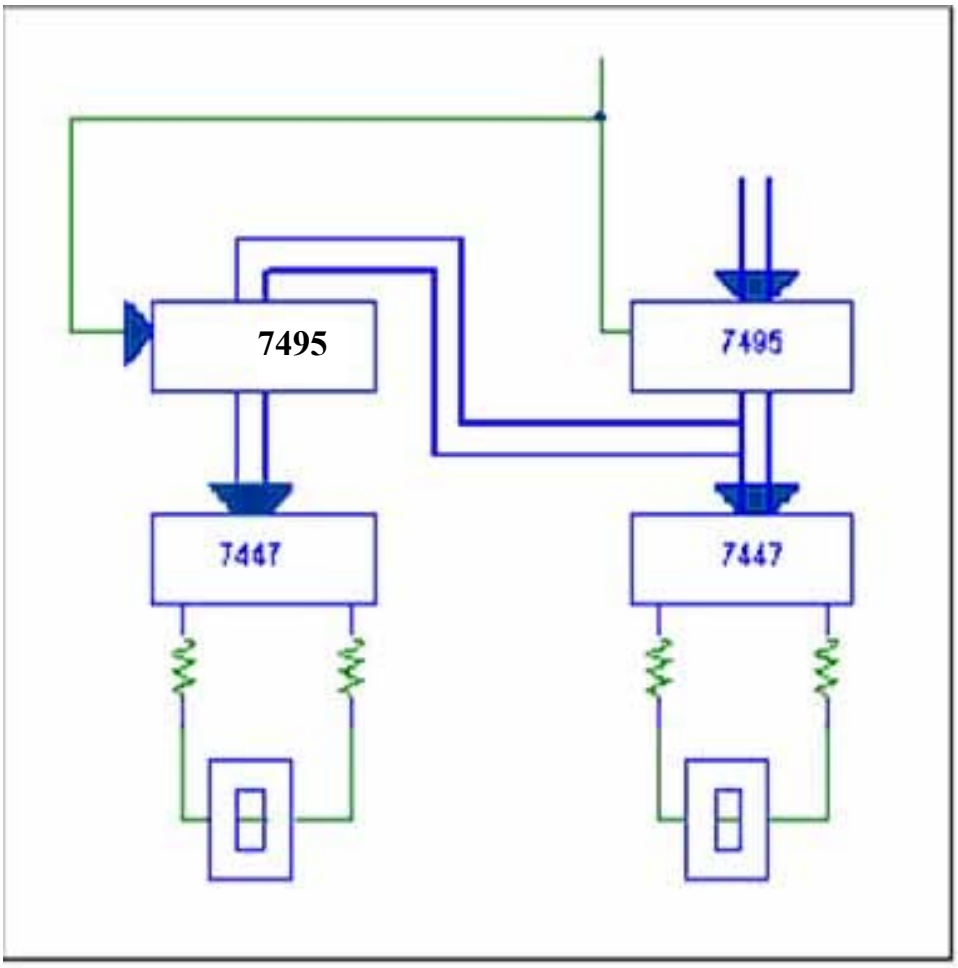
**کار آزمایشگاهی ۲**

فرکانس آستابل را 2kH3 و پهنای پالس منواستابل را 50ms درنظر گرفته و عناصر لازم برای ساخت کد فوق را انتخاب نمائید. سپس مدار طراحی شده را بر روی بردبرد بسته و خروجی آن را برروی LED مشاهده نمائید.[1] 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 کدیدها را حذف نموده تا اطلاعات به فرم صحیحی وارد کامپیوتر شدند. در عمل حذف نویز کلیدها به روش‌ها نرم افزاری و سخت افزاری صورت می‌گیرد که در دنبالهٔ این آزمایش با روش سخت افزاری آشنا می‌شویم.

**کار آزمایشگاهی ۳** (در صورت موجودبودن 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.

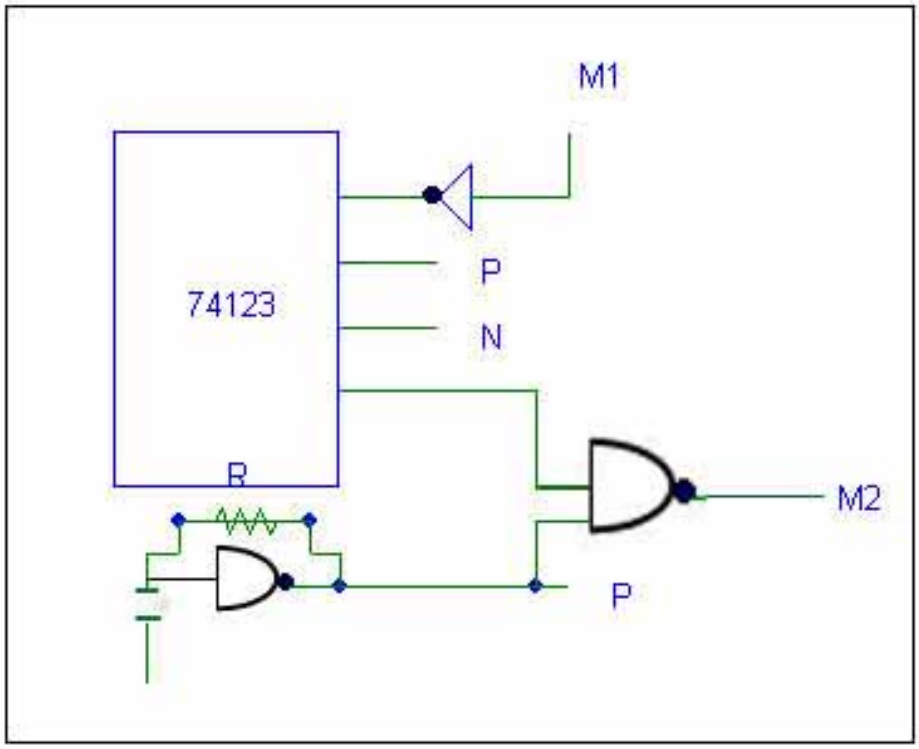
[1] میتوانید پس از مشاهدهٔ خروجی برروی LED از 7.seg هم استفاده نمائید و خروجی را برروی 7.seg مشاهده کنید.




با تغییری در رابطه با انتقال اطلاعات حاصل گردید؟(درهربار 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 می‌نماید.(یک مدار عملی بایستی قادر به دفع هردو نوع نویز باشد. آیا می‌توان مدار فوق را برای این منظور به کار برد؟چگونه؟)

**کار آزمایشگاهی ٤- 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 برروی بردبرد بسته و آزمایشات را تکرار کنید.

## آزمایش شماره ٤

عنوان آزمایش:جمع کننده و تفریق گر

در این آزمایش، شما انواع مدارهای جمع و تفریق گر را ساخته و تست خواهید کرد. سپس از تفریق گر برای مقایسه اندازه نسبی دو عدد استفاده می کنید.

### آزمایش ٤-١) نیم جمع کننده

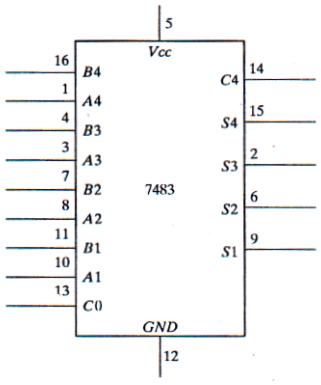
یک مدار نیم جمع کننده را با استفاده از یک گیت XOR و دو گیت NAND، طراحی، ساخته و تست نمایید.

### آزمایش ٤-٢) جمع کننده کامل

یک مدار جمع کننده کامل را با استفاده از دو آی سی ۷۴۸۶ و ۷۴۰۰، طراحی، ساخته و تست نمایید.

### آزمایش ٤-٣) جمع کننده موازی

آی سی ۷۴۸۳ یک جمع کننده موازی ۴ بیت است. تخصیص پایه در شکل (۵-۱) نشان داده شده است. دو عدد دودویی ۴ بیتی ورودی به $A_4$ تا $A_1$ و $B_4$ تا $B_1$ اعمال می‌شوند. حاصل جمع چهار بیتی از پایه‌های $S_4$ تا $S_1$ گرفته می‌شود. $C_0$ رقم نقلی ورودی و $C_4$ رقم نقلی خروجی است.

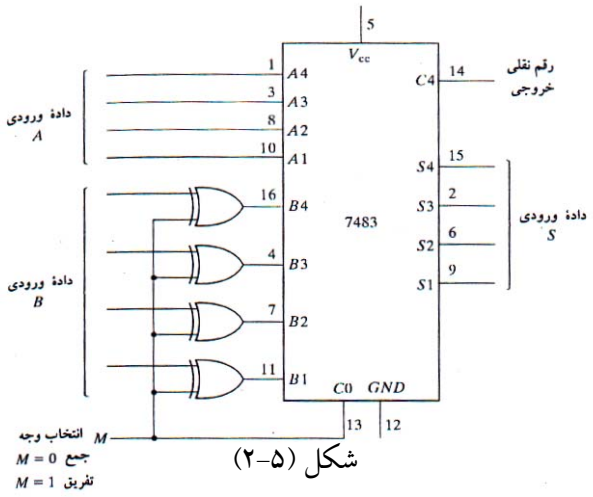



شکل (۵-۱)

با اتصال پایانه های منبع تغذیه و زمین، جمع کننده دودویی ۴ بیت ۷۴۸۳ را تست نمایید. سپس چهار ورودی A را به عدد ثابت دودویی ۱۰۰۱ و ورودی های B و ورودی نقلی را به پنج کلید دو وضعیتی وصل کنید. پنج خروجی به پنج لامپ متصل اند. جمع چند عدد دیگر را هم اجرا نموده و ببینید آیا جمع خروجی و نقلی خروجی مقادیر صحیحی را ارائه می‌دهند یا خیر. نشان دهید که وقتی رقم نقلی ورودی ۱ است به خروجی یک واحد اضافه می گردد.

## آزمایش ٤-٤) جمع- تفریق گر

تفریق دو عدد دودویی می‌تواند با استفاده از متمم ۲ مفروق و جمع آن با مفروق منه بدست آید. متمم ۲ را می‌توان با جمع متمم ۱ باعدد ۱ بدست آورد. برای اجرای A-B، ما چهار بیت از B را متمم نموده و آن را به چهار بیت A اضافه می کنیم، آنگاه یک واحد نیز از طریق رقم نقلی به مجموع آن دو می‌افزاییم. این کار در شکل (۵-۲) نشان داده شده است. چهار گیت XOR بیت های B را وقتی حالت M=۱ برقرار است متمم می‌نمایند (زیرا $x \oplus 1 = x'$) و اگر M=۰ باشد، نقلی ورودی هم ۰ و حاصل جمع تولید شده A+B خواهد بود.



شکل (۵-۲)

مدار جمع- تفریق گر را ببندید و آن را آزمایش کنید. چهار ورودی A را به عدد ثابت دودویی ۱۰۰۱ و B را به چهار سوئیچ وصل نمایید. اعمال زیر را اجرا و مقادیر خروجی جمع و نقلی خروجی C را ثبت کنید.

۹-۵ ۹+۵

۹-۹ ۹+۹

۹-۱۵ ۹+۱۵

نشان دهید که ضمن عمل جمع، اگر حاصل جمع از ۱۵ بیشتر شود، نقلی خروجی ۱ خواهد بود. و نیز نشان دهید که وقتی A>B باشد، عمل تفریق جواب صحیح A-B را تولید می کند و رقم نقلی $C_{-1}$ است. ولی وقتی A<B باشد، تفریق متمم ۲ را برای B-A را تولید کرده و نقلی خروجی هم ° است.

## آزمایش شماره ۵

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 می‌باشد. در این روش کوشش می‌شده تا تمام علائم کنترلی موردنظر به کمک تحولات منطقی در مدارات فیزیکی ساخته می‌شوند. چنین سیستم‌هایی از ترکیب واحدها یا مداراتی شبیه شمارنده‌ها، دیکدرها، انکدرها و... به وجود می‌آیند.

در این روش وقتی مدار طراحی شده اگر بخواهیم تغییری در سیگنال‌های کنترلی بدهیم، لازم است مدار موردنیاز دوباره طراحی و ساخته شود. دومین روش طراحی واحد کنترل، که اصطلاحاً میکروپروگرام نامیده می‌شود، به سیستم کنترلی گفته می‌شود که برای ساخت و تهیه تمام علائم کنترلی تنها از مدارات منطقی استفاده نکرده باشد. در این گونه سیستم‌ها معمولاً تمامی یا بخشی از علائم کنترلی به کمک سیکل اطلاعات باینری که در درون یک حافظه و به صورت یک برنامهٔ میکروپروگرام در ROM انبار شده است، تهیه و تدارک دیده می‌شود. برای طراحی واحد کنترل هر ماشینی در مرحله اول بایستی وضعیت ماشین از نظر اجرا در دستورالعمل فرامین‌ها روشن گردد. بعد از روشن شدن دستورالعمل‌های موردنیاز در ماشین می‌بایست تسلسل میکروکدهای لازم را برای اجرای آن دستورالعمل‌ها در تشکیلات طراحی شده پیدا کرد و بدین ترتیب در پایان این

مرحله هم طرح تشکیلاتی ئ هم میکرو کدهای دستورالعمل روشن شده اند. با استفاده از جدول میکروکدها به سادگی می‌توان 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

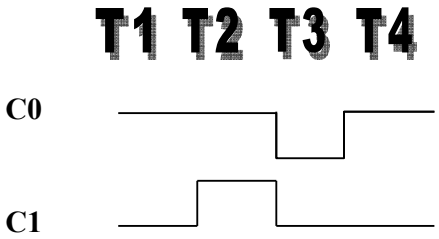
۳- روش استفاده از عناصر تأخیری

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 دیجیتالی 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 ازگیت‌های 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 (pos , sop ,...) 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 |

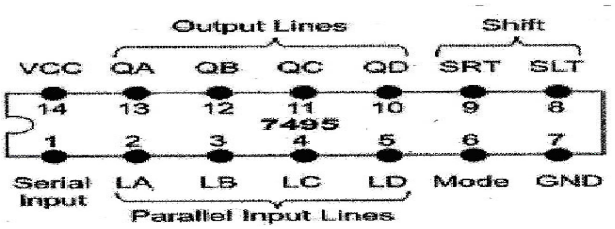
C0,C1: سیگنالهای خروجی

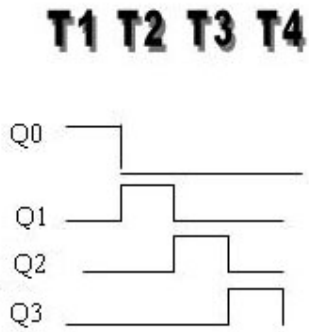
X,Y :خروجی شمارنده

«شکل B1»

**کار آزمایشگاهی ۱:**

IC 7495 را بدقت بررسی نمایید و جدول صحت آن را در تمامی حالات بدست آورید سپس این IC ر ا در حالتی که بتواند شیفت به راست یک (۱) را انجام دهد بسته و تحقیق کنید که خروجی‌های آن مانند شکل ۲ باشد.

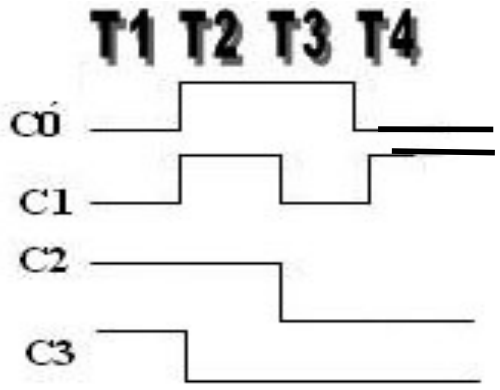



«شکل شماره 2»

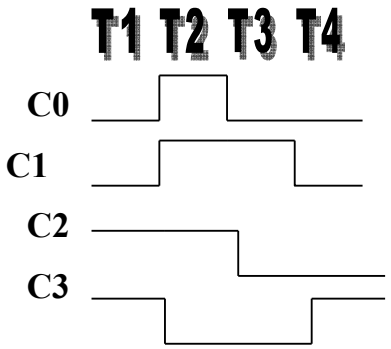
حال که توانسته اید شیفت به راست یک (۱) منطقی را روی خروجی‌های Q0 تا Q1 مشاهده نمایید ازخروجی‌های $Q$ 0 تا $Q_3$ استفاده نموده و با استفاده از گیت‌های NOR یا OR

سیگنال‌های شکل۳ 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 شمارهٔ ۷ باشد.




«شکل شماره 4»

**کار آزمایشگاهی ۲**

به کمک IC شمارندهٔ 7493 ویک 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 وببندید.

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

| $Q_B$ | $Q_A$ | $C_0$ | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|---|---|
| 0 | 0 | | | | |
| 0 | 1 | | | | |
| 1 | 0 | | | | |
| 1 | 1 | | | | |

## آزمایش شمارهٔ ٦

عنوان آزمایش: **واحد کنترلی به روش میکروپروگرام**

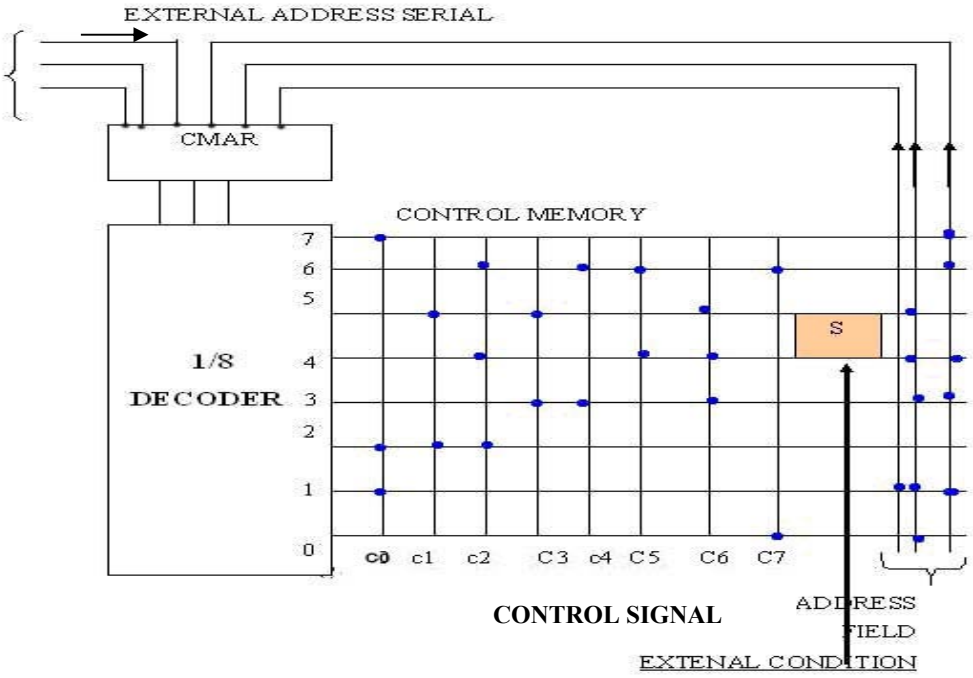
**مقدمه:**

یکی از روش‌های جالب برای طرح واحد کنترل را می‌توان روش میکروپروگرام دانست. این روش برای اولین توسط M-V- Wilks انگلیسی در سال ۱۹٦۱ 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 ویژه ای 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 ریر 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 ۱۹٦۱ پیشنهاد شده نشان می‌دهد. در این شکل حافظه کنترل از نوع ROM می باشد. ردیف‌های M مشخص کننده ریزدستورالعمل‌ها و ستونهای آن مشخص کنندهٔ 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 می‌شود و محتوای آن در خروجی قرار می‌گیرد (نقاط توپر علامت وجود دیود هستند) برای مثال وقتی که آدرس ۱۱۰ در CMAR قرار می‌گیرد سطر هفتم حافظه فعال شده و به دنبال آن مطابق شکل خطوط فعال می شود و بقیه غیرفعال می‌گردد. همانگونه که در شکل ۱ مشخص شده است، CMAR در خارج یا وسط بخش آدرس حافظه کنترل تغذیه می‌گردد. اساساً آدرسی که از خارج داده می‌شود، آدرس شروع ریزدستورالعمل یا ریزبرنامه را مشخص می‌کند و به دنبال آن آدرس‌های بعدی توسط حافظه کنترل تأمین می‌شود. برای مثال اگر یک CPU که با تکنیک میکروپروگرام طراحی شده است، هر

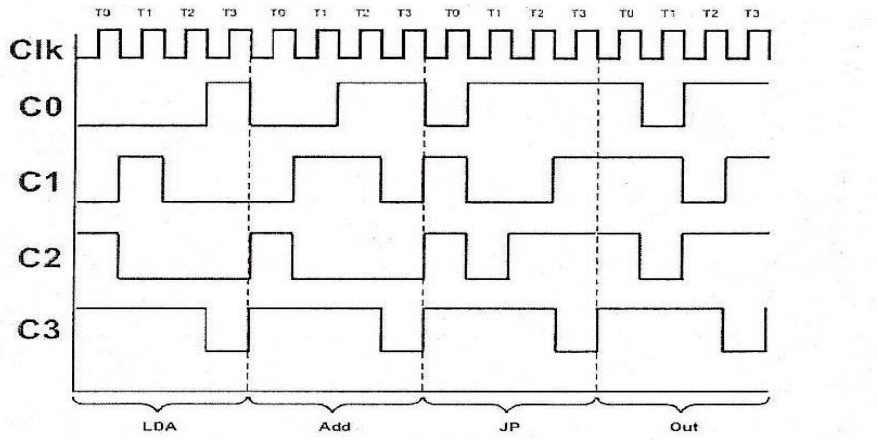
دستورالعمل با یک میکروپروگرام مشخص می‌شود. بدین ترتیب وقتی که دستورالعمل در رجیستر دستورالعمل (IR) قرار گرفت، قسمت Opcode آن آدرس شروع برنامه را مشخص می‌کند. همانطور که می‌دانید از توانایی‌های دیگری که هر واحد کنترل باید داشته باشد، این است که می تواند با توجه به شرایط اعمال شده سیگنال‌های کنترلی مناسب 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 شمارهٔ ٤ 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 ٤۰ پالس ساعت لازم باشد. بنابراین هر چهار سئوال 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

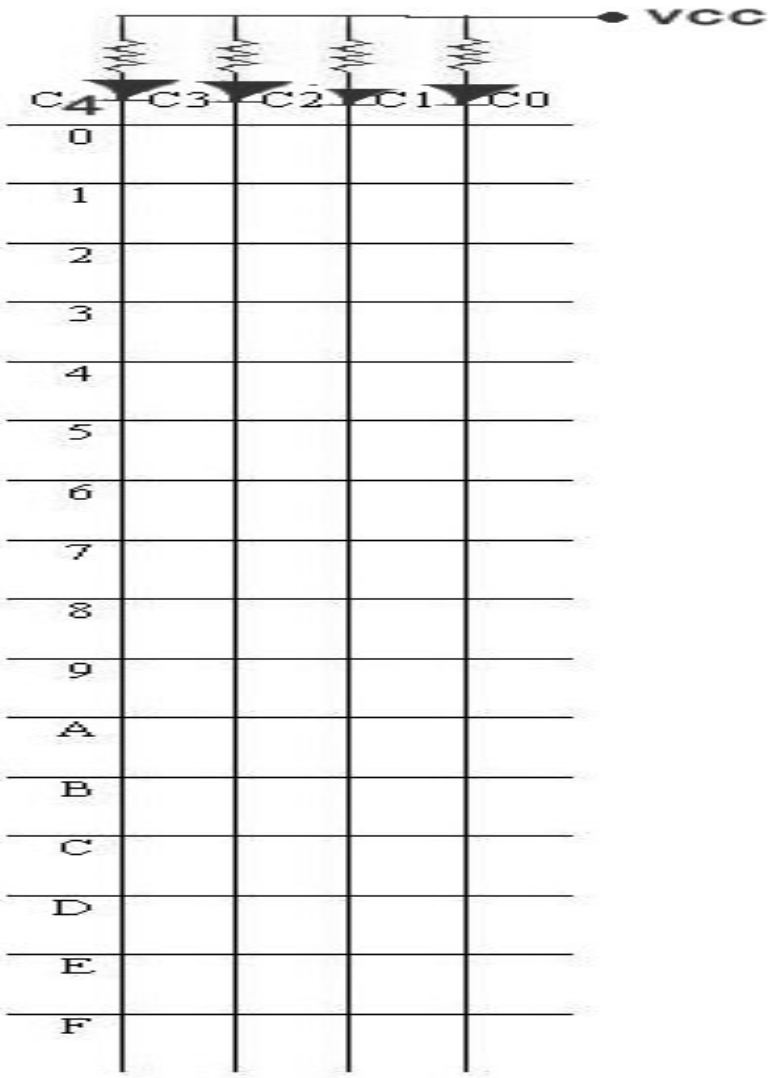
JP 1000 ⟶ 1011 , OUTA 1100 ⟶ 1111

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)

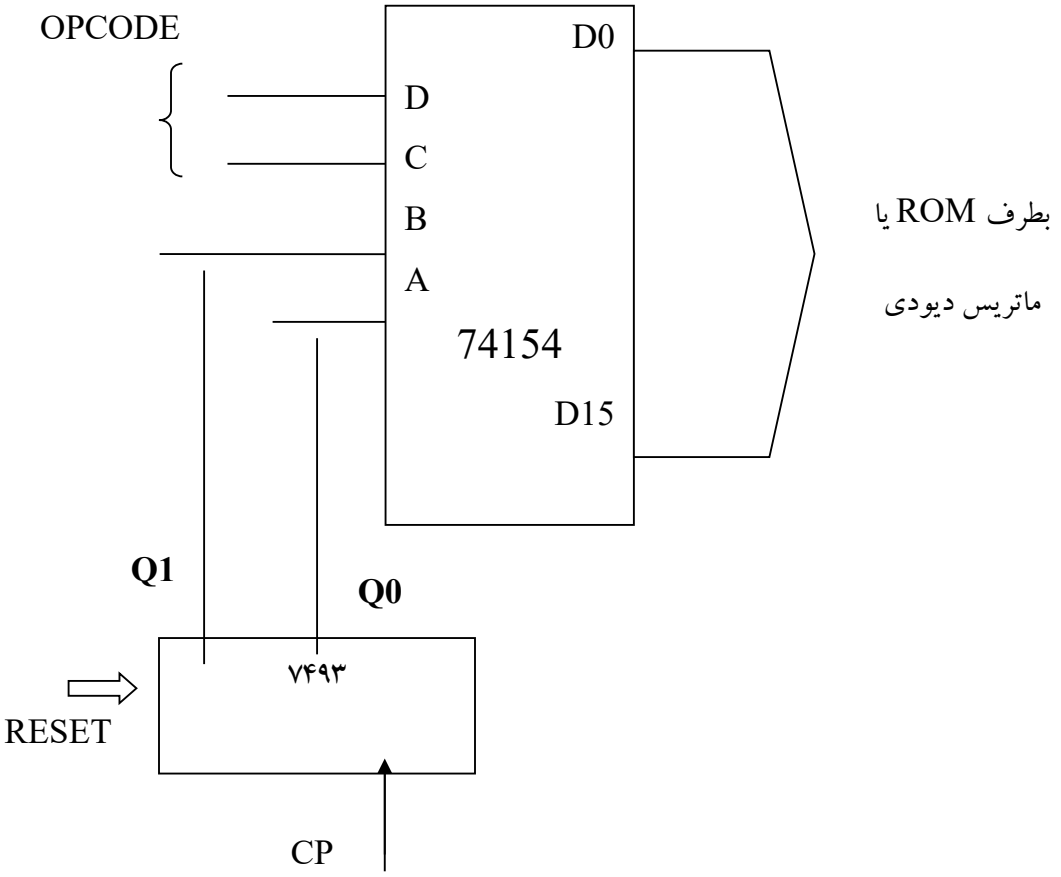



شکل (۳)

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

**کار آزمایشگاهی ۱**: برای آزمایش فوق مدار شکل ٤ 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 (٤)

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

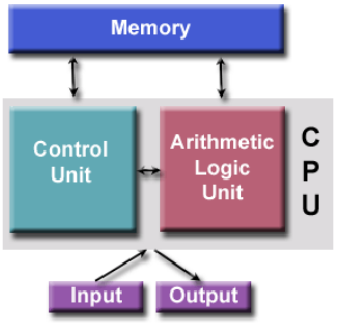
**تمرین:** در صورتیکه opcodeهای LDA ، ADD و outA-JP به ترتیب ۰۰ و ۰۱ و ۱۰ و۱۱ باشد، حافظه کنترل را مطابق حالت قبلی طوری تکمیل کنید که حداقل حافظهٔ کنترل استفاده گردد. لازم است جهت انجام آزمایشات دیتابوک همراه داشته باشید.

# آزمایش ۷

## 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 می‌کند. اجزای دیگر این اطلاعات را بین رجیسترها و حافظه جابه‌جا می‌کنند. 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:

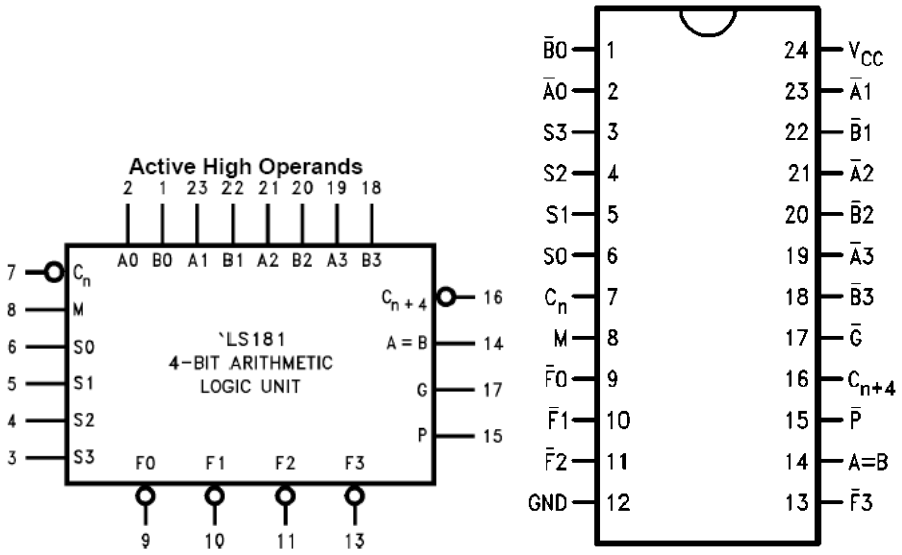
- عملیات های محاسباتی صحیح ( جمع ، تفریق و بعضی ضرب و تقسیم .. )
- عملیات های بیتی منطقی ( AND، NOT، OR، XOR )
- عملیات های انتقال بیتی (Bit-shifting) (انتقال یا چرخش یک کلمه با تعداد مشخصی بیت به راست یا چپ؛ با/بدون گسترش علامت )

✓ **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 $\overline{A۳} - \overline{A۰}$ و $\overline{B۳} - \overline{B۰}$ و پایه‌های خروجی $\overline{F۳} - \overline{F۰}$ 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 | $\overline{C}n$ = H (no carry) | $\overline{C}n$ = 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.