



# آزمایشگاه معماری کامپیوتر

گزارش آزمایش ششم

استاد: جناب آقای دکتر سربازی آزاد

ترم تابستان

موضوع: واحد محاسبه با امکان انتخاب ثبات مبدأ و مقصد

اعضای گروه:

فرزام کوهی رونقی ۴۰۱۱۰۶۴۰۳

مریم شیران ۴۰۰۱۰۹۴۴۶

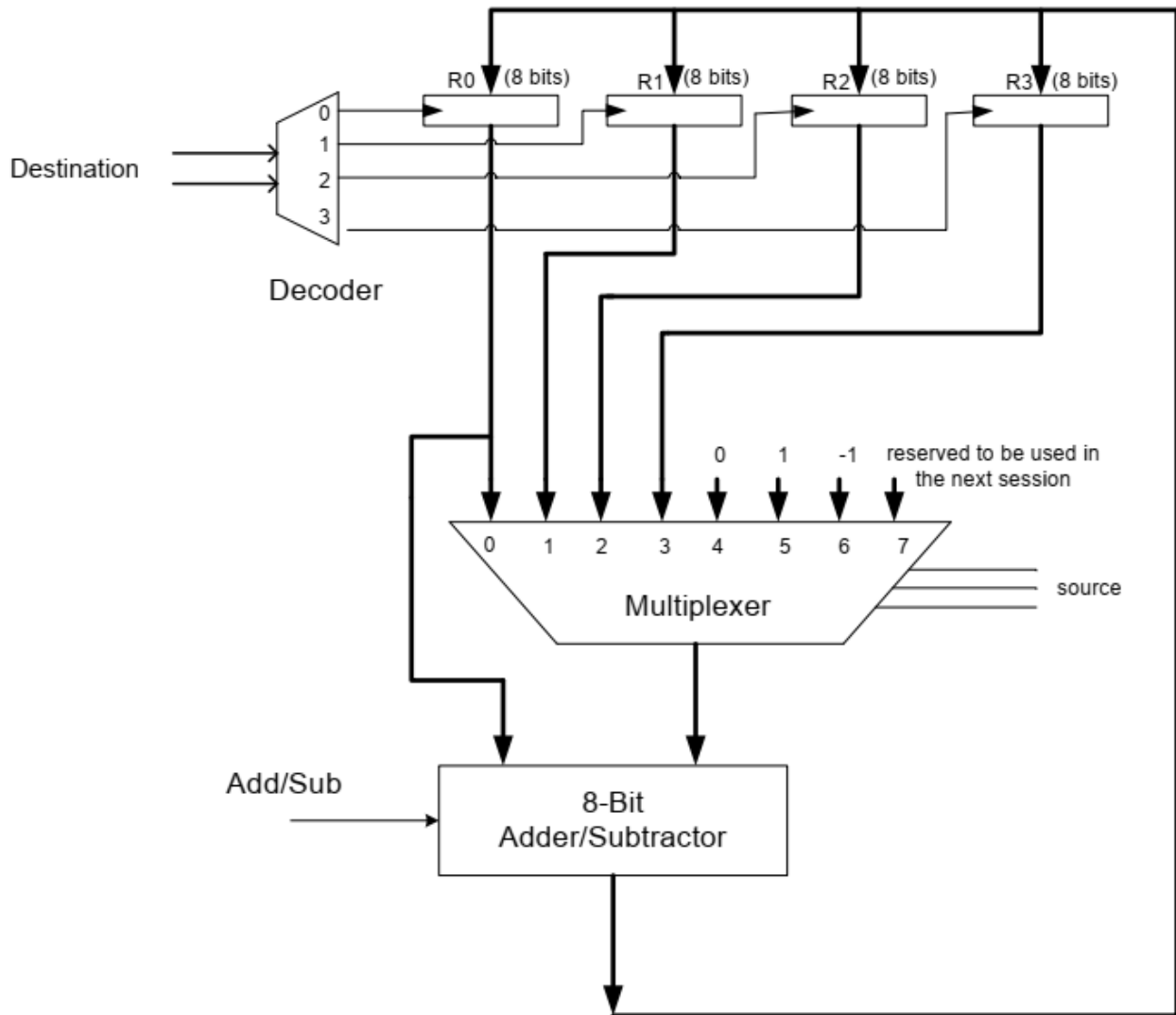
ثنا باباییان ونستان ۴۰۱۱۰۵۶۸۹

## اهداف آزمایش :

در این آزمایش قصد داریم که ی E واحد محاسبه گر طرح T کنیم که بسته به دستور ورودی که به آن م T آید، عملیات جم یا تفریق را روی ورودی هایی که به صورت ثابت یا عدد ثابت از پیش تعریف شده در آن قرار گرفته اند، اجرا کرده و نتیجه را در ثابت T که در دستور برای آن مشخص شده است. بریزد.

## توضیحات :

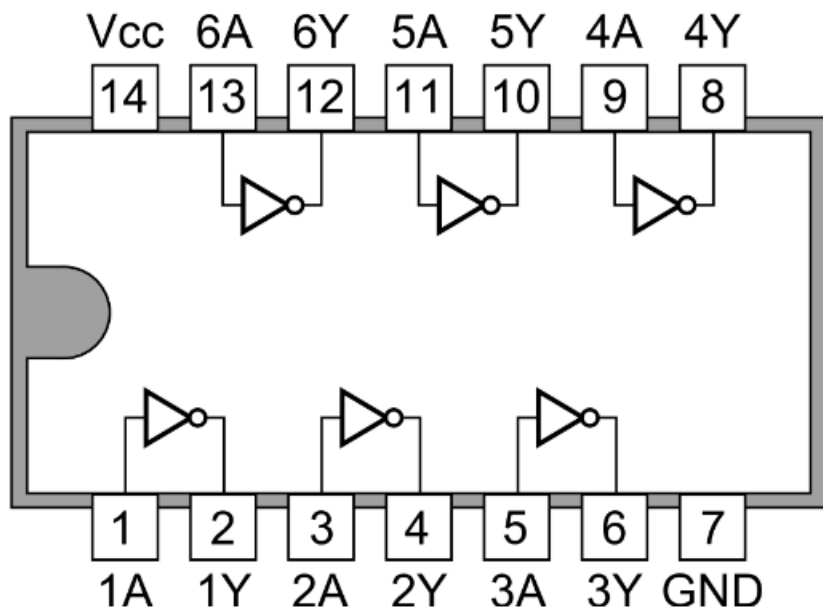
در این آزمایش از سیستم مکمل ۲ برای نمایش اعداد استفاده کردیم. برای انجام تفریق کی توانیم از جمع عدد اول با مکمل ۲ عدد دوم استفاده کنیم که یعنی تمام بیت های عدد دوم را نات کرده و با عدد اول جمع می کنیم البته با ۱ carry in .



شکل ۱. معماری واحد محاسبات

## قطعات مورد استفاده :

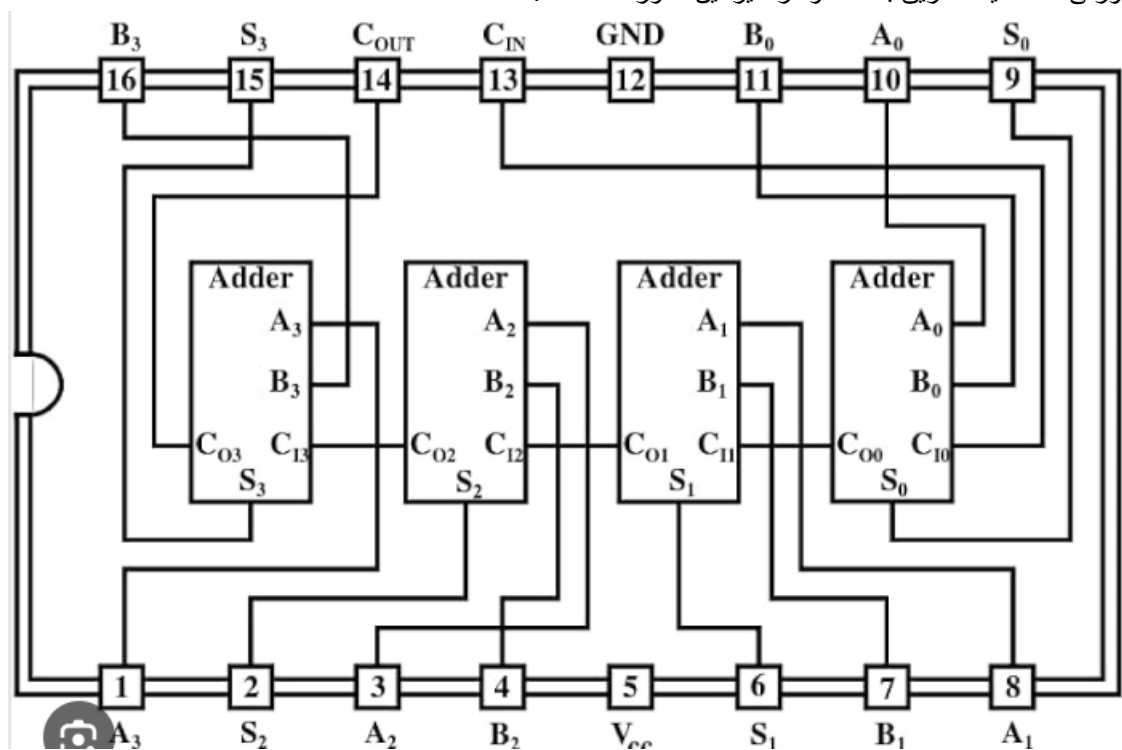
- 7404 hex inverter



شکل ۲. تراشه ۷۴۱۵۱

#### • 7483 4 bit full adder :

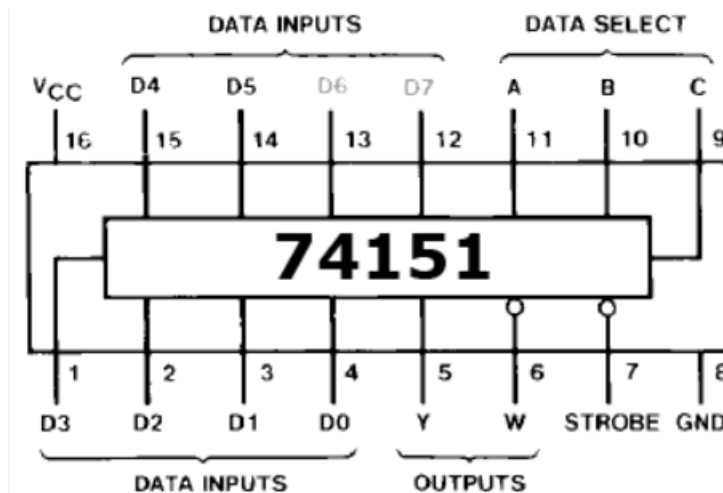
در درون آن از سیستم Lookahead Carry برای Carry استفاده شده است. از دو تا قطعه ۷۴۸۳ برای عملیات جمع هشت بیتی استفاده کرده ایم و آنان را به صورت Carry-Ripple به هم متصل نموده ایم. ورودی Carry جم کننده اول در صورتی که عملیات تفریق باشد ۱ و در غیر این صورت ۰ است.



شکل ۳. تراشه ۷۴۸۳

#### • 74151 8-input multiplexer :

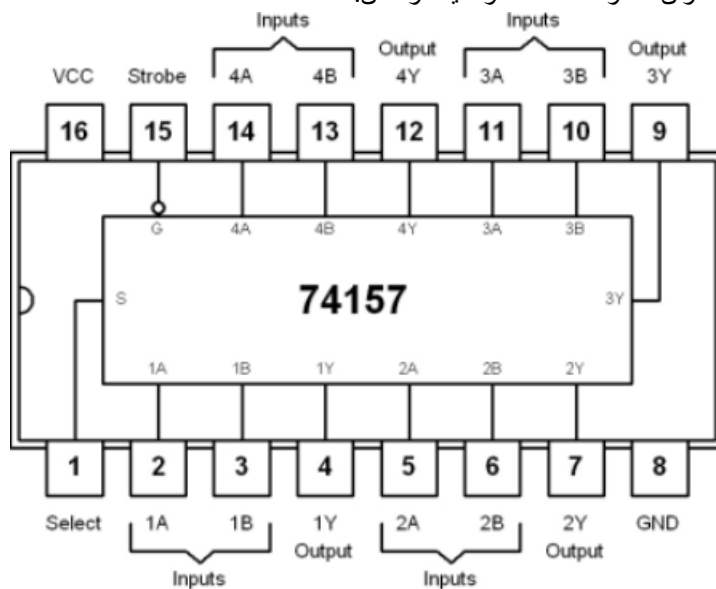
این قطعه یک مالتیپلکسر ۸ به ۱ است. از آن جایی که در سیستم مالتیپلکسر این مدار در اصل باید یک مالتیپلکسر ۸ به ۱ با ورودی های ۸ بیتی بسازیم، از هشت عدد قطعه ۷۴۱۵۱ در مدار استفاده شده است. ورودی های CBA به ترتیب ارقام سلکت آن را می سازند و D0-D7 ورودی های آن هستند.



شکل ۴. تراشه ۷۴۱۵۱

#### • 74157 Quad 2-line to 1-line multiplexer :

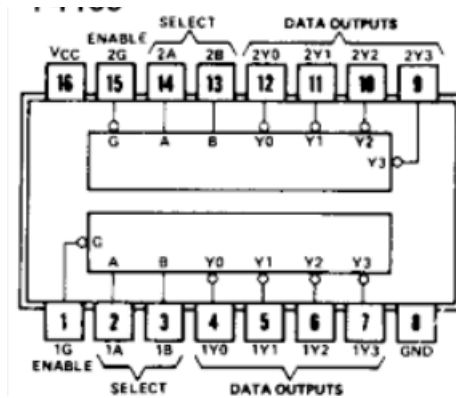
این قطعه یک مالتیپلکسر ۲ به ۱ است و در درون هر IC آن، چهار مالتیپلکسر از این نوع قرار گرفته است. از دو قطعه ۷۴۱۵۷ در این مدار در قسمت جمع کننده استفاده شده تا بسته به این که عملیات جمع یا تفریق است، مشخص کنیم که NOT ورودی ها باید به عنوان عملوند استفاده شوند یا خودشان.



شکل ۵. تراشه ۷۴۱۵۷

#### • 74139 dual 2-line to 4-line decoder :

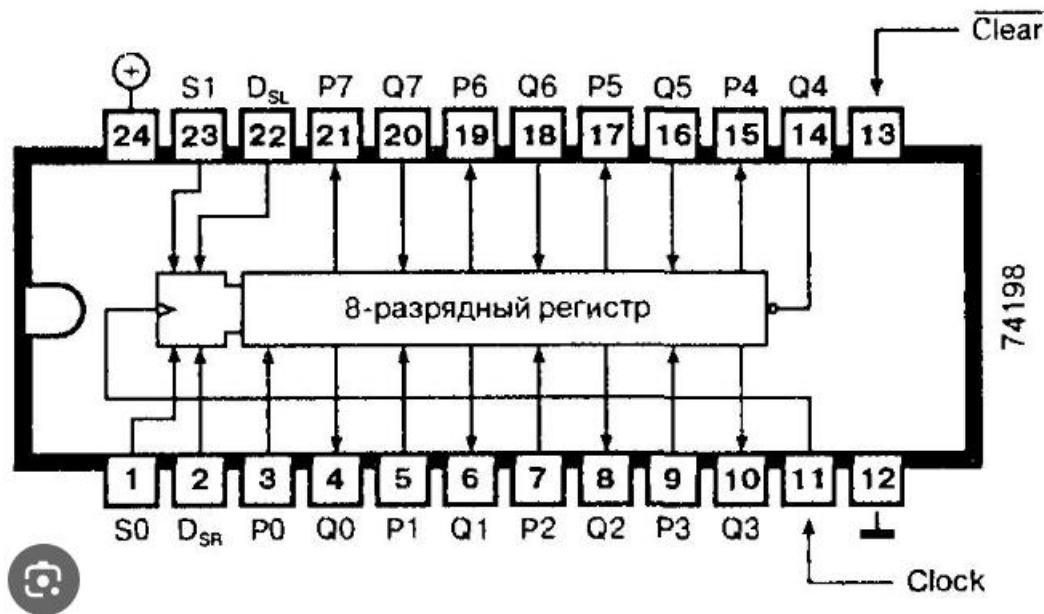
این قطعه در درون خود دو عدد دیکودر ۲ به ۴ دارد. ورودی های A سلکت آن را مشخص می کنند و خروجی های آن هستند. فقط باید توجه کرد که خروجی آن active low است و در نتیجه برای استفاده مناسب در مدار ما باید NOT آن را با تراشه رایج ۷۴۰۴ محاسبه کنیم.



شکل ۶. تراشه ۷۴۱۳۹

• 74198 8 bit shift register :

یک ثابت ۸ بیتی با امکان load و خروجی دادن موازی است. اصلی ترین بین های آن S0S1 هستند که اگر ۰۰ باشند، وضعیت قطعه ثابت می ماند. اگر ۰۱ باشند شیفت به چپ و اگر ۱۰ باشند شیفت به راست و در صورتی که ۱۱ باشند، load موازی صورت می گیرد. در این آزمایش ما به حالت ۰۰ و ۱۱ آن ها نیاز داریم.



شکل ۷. تراشه ۷۴۱۹۸

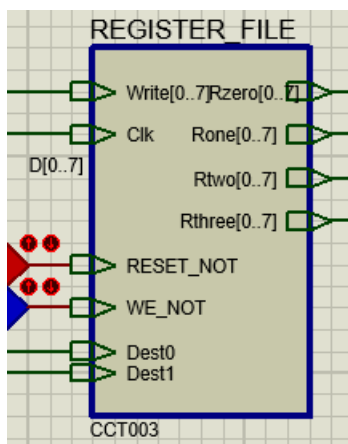
### شرح آزمایش و پیاده سازی :

برای ساده سازی پیاده سازی مدار در کلاس ماکس ۷ ورودی را به ماکس ۴ ورودی با ۰، ۱، r1، r0 تبدیل کردیم. حاصل نهایی به یکی از رجیسترهای مقصد r0 و یا r1 می رود که برابر با نتیجه جمع یا تفریق یکی از چهار ورودی گفته شده با مقدار درون رجیستر پایه r0 است.

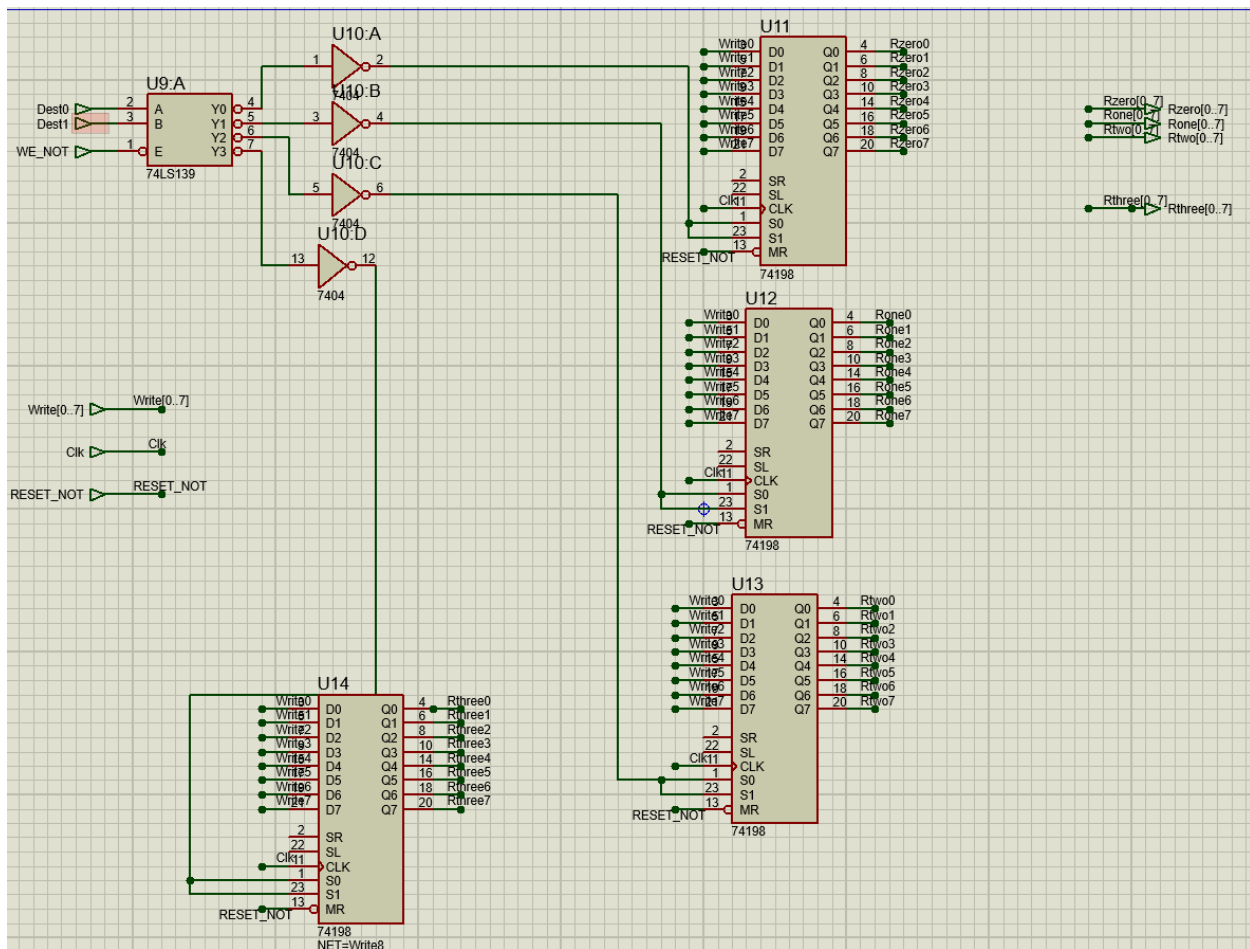
### : Register file

در این قسمت چهار ورودی ۸ بیتی در نظر گرفته شده و در کنار آن چهار خروجی ۸ بیتی از هر کدام از رجیستر فایل ها هم در نظر گرفته ایم. در کنار این موضوع، دو بیت برای این که مشخص شود در کدام رجیستر فایل قصد نوشتن داریم در نظر گرفته ایم که در داخل مدار، وارد دیکودر می شوند و براساس آن، مشخص می شود که S0S1 کدام رجیستر باید به حالت ۱۱ برای load

موازی ورودی داده شده در بیاید. همچنین یک ورودی write enable هم در نظر گرفته ایم که به صورت active low است و تنها در صورتی که ۰ باشد دیکودر فعالیت کرده و عملیات Write انجام خواهد شد؛ در صورت یک بودن آن رجیسترها همواره مقدار خود را حفظ می کنند. یک ورودی Reset هم که به صورت active low است در نظر گرفته شده که در صورت ۰ شدن مقدار ذخیره شده درون تمامی رجیسترها 0 می شود. همچنین یک کلاک هم برای این ماژول قرار گرفته که کلاک رجیسترها را تامین می کند. برای عملکرد معمول سیستم باید مقدار Reset برابر ۱ و مقدار write enable برابر ۰ باشد.



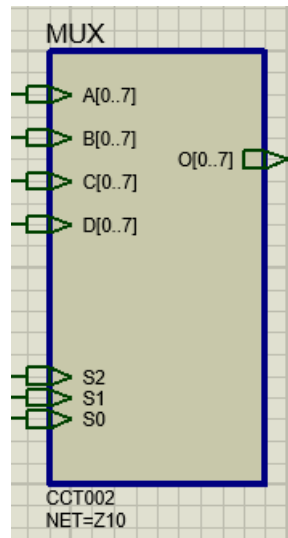
شکل 8. نمای خارجی register file



شکل ۹. نمای درونی register file

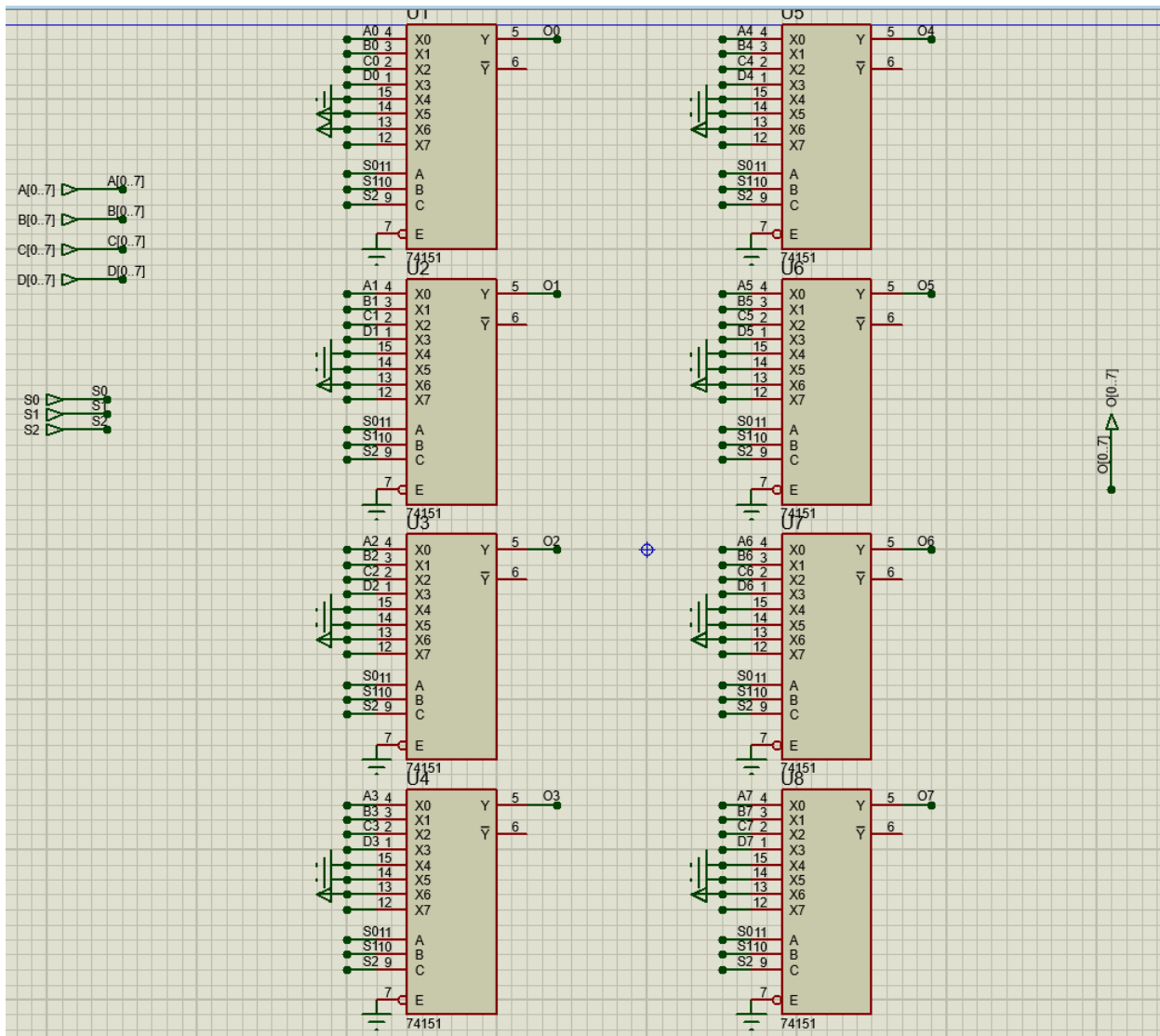
MUX:

در این بخش 8 حالت مختلفی که در آزمایش گفته شده را بررسی و پیاده سازی کرده ایم. 8 multiplexer 8 به 1 قرار گرفته و در اصل هر multiplexer ، یک بیت از خروجی را ایجاد می کند. چهار حالت اول که بر اساس ورودی هایی که از خروجی رجیستر ها می آیند تعیین می شوند. برای سه حالت باقی مانده هم اعداد گفته شده را از طریق اتصال GND و VCC ایجاد کرده ایم. مثلاً برای ۱- ورودی ۶ ام همه multiplexer ها به VCC وصل شده و برای ۱ ورودی ۵ اولین multiplexer به VCC و بقیه ورودی های ۵ به GND وصل شده اند. توجه کنید که حالت هشتم که با شماره ۷ مشخص می شود، به چیزی متصل نیست (چون قرار است در آزمایش های بعدی مورد استفاده قرار بگیرد پس تضمینی برای رفتار آن در این آزمایش وجود ندارد).



شکل ۱۰. نمای خارجی  $MUX$

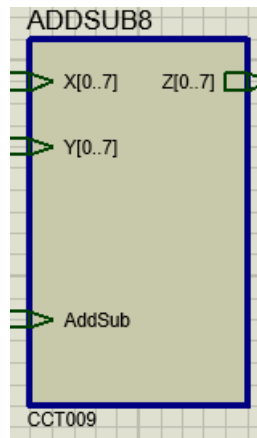




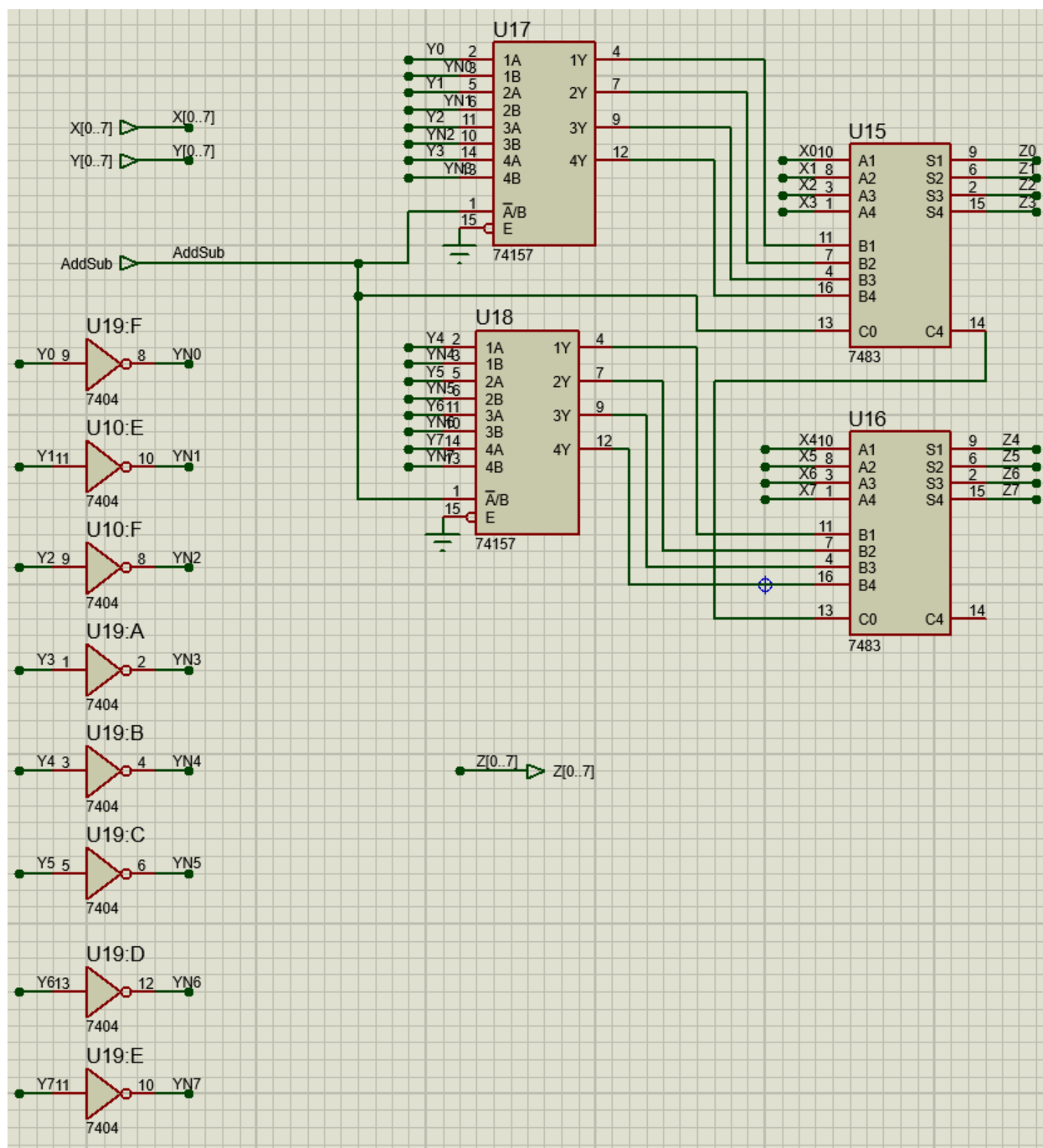
شکل ۱۱. نمای درونی MUX

: Logic

عملیات جمع و تفریق پیاده سازی شده است. ورودی اول که مستقیماً از رجیستر ۰ ام می آید. ورودی دوم از Multiplexer Custom آمده و در داخل مدار به طور کامل NOT می شود تا هم خود ورودی و هم NOT آن را در اختیار داشته باشیم. یک ورودی برای تعیین این که در حالت جمع یا تفریق هستیم هم وجود دارد که جمع 0 و تفریق با ۱ مشخص می شود. در صورتی که در حالت تفریق باشیم، NOT ورودی دوم به عنوان ورودی جمع کننده استفاده شده و Carry ورودی جمع کننده اول هم یک خواهد بود. در حالت جمع خود ورودی دوم استفاده شده و Carry ورودی جمع کننده هم صفر خواهد بود. دو جمع کننده هم به صورت Carry-Ripple به هم متصل شده اند. در نهایت خروجی این واحد به عنوان ورودی register file وارد آن می شود تا در صورت ۰ بودن مقدار write enable عملیات نوشتن انجام شود.



شکل ۱۲. نمای خارجی بخش *logic*

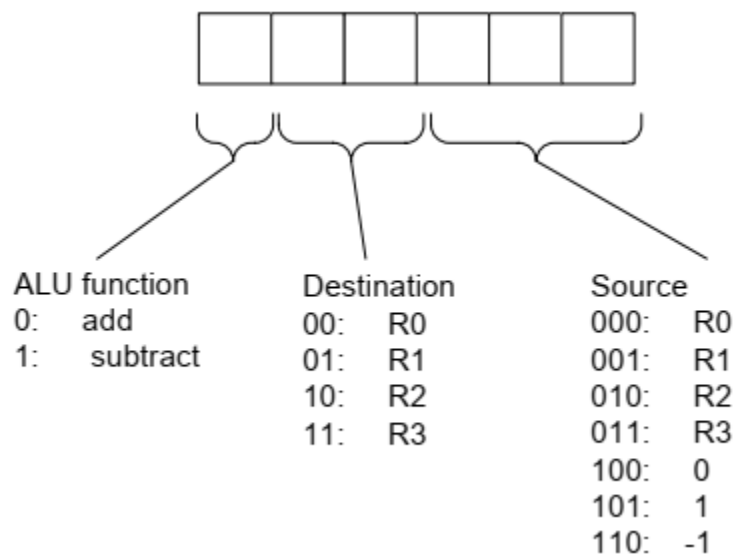


شکل ۱۳. نمای درونی بخش logic

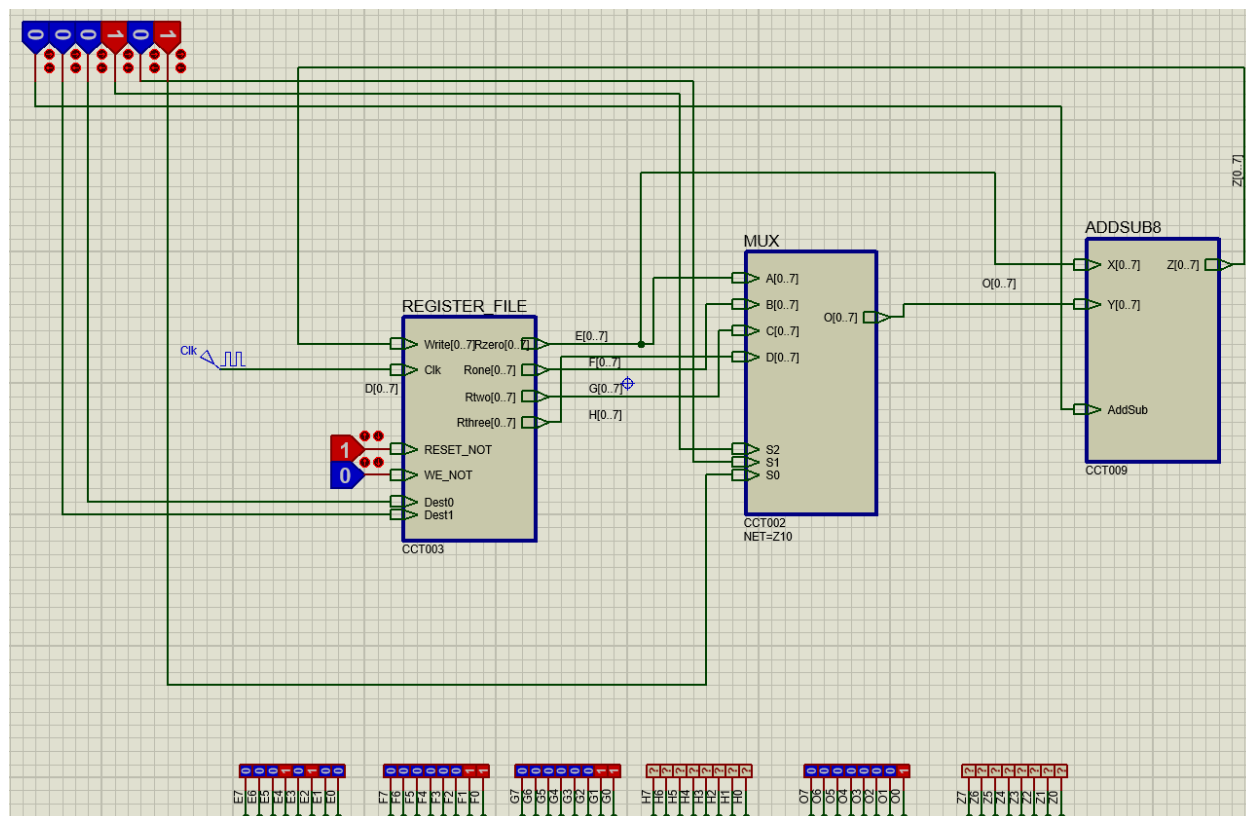
## روش کار با مدار :

ابتدا توجه کنید که باید Reset را در حالت ۱ قرار بدهید تا مدار در حالت Reset دائمی باشد. همچنین در صورتی که بخواهید عملیات نوشتن انجام بشود، باید Write Enable برابر ۰ باشد و ۱ بودن آن به نوعی عادل با Pause کردن کل روند مدار است. در گوشه مدار دستور ورودی را قرار داده ایم که دقیقاً مطابق با چیزی است که در صورت آزمایش از ما خواسته شده است. یعنی فرمت آن به صورت FDDSSS است که F در صورت ۱ بودن تفریق و در غیر این صورت جم است. دو بیت بعدی (D) مقصد و سه بیت بعدی (S) یکی از حالات Multiplexer را انتخاب می کنند. خروجی ها در بالای مدار قرار دارند که به ترتیب از چپ به راست،

خروجی رجیستر صفرم، خروجی رجیستر اول، خروجی رجیستر دوم، خروجی رجیستر سوم، خروجی Mux Custom و خروجی جمع کننده (بخش Logic) هستند.



شکل ۴. ۱. قالب دستورات شش بیتی



شکل ۵. ۱. نمایی کلی از مدار آزمایش