به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

آزمایشگاه معماری کامپیوتر

آزمایش ششم:

کنترل توسط برنامه ذخیره شده در حافظه

اطلاعات تیم	
نام اعضا	شماره دانشجویی
متین داغیانی	۹۸۱۰۶۴۵۶
بردیا محمدی	۹۸۱۷۱۱۰۴
محمدجواد هزاره	۹۸۱۰۱۰۷۴

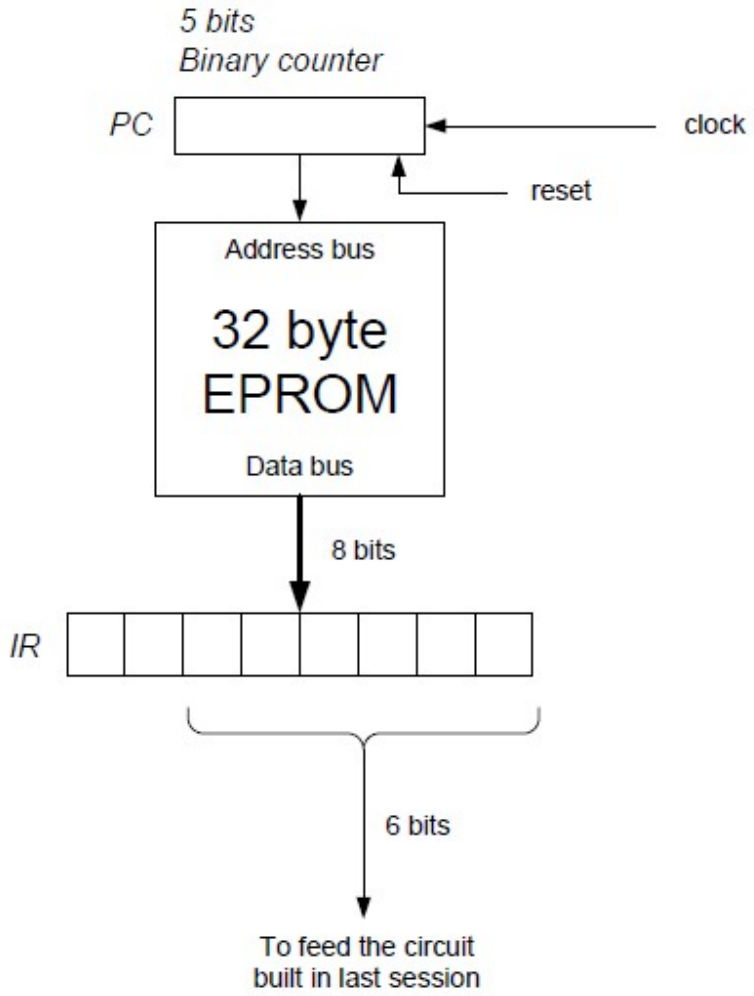
پاییز ۱۴۰۰

فهرست مطالب

فهرست تصاویر

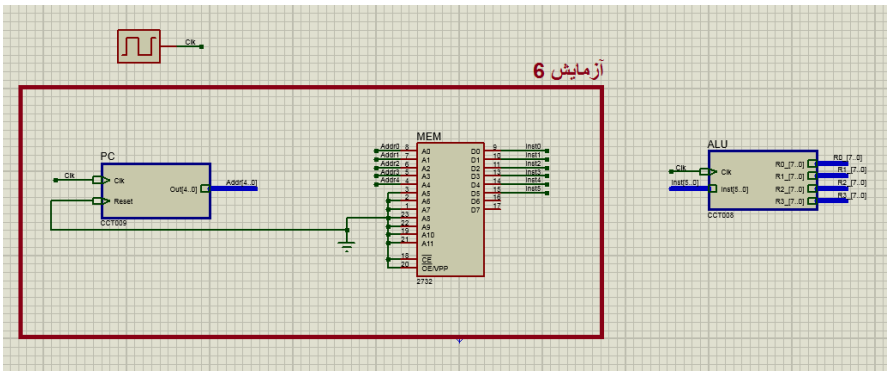
۱ هدف آزمایش

هدف از این آزمایش آشنایی با نحوه واکشی و اجرای دستورات از حافظه دستور در پردازنده است. برای پیاده سازی لام است تا فرمانهای لازم جهت کنترل مدار را از برنامه ذخیره شده در EPROM واکشی کرده و سپس اجرا کنیم. بدین جهت مدار زیر را به واحد چریان داده طراحی شده در آزمایش پنجم اضافه می کنیم.



شکل ۱: بلوک دیاگرام سیستم

همان طور که در تصویر بالا مشخص است، این سیستم شامل یک PC پنج بیتی است که آدرس دستور بعدی در حافظه را مشخص می کند. هم چنین دستورات برنامه در حافظه ذخیره شده اند. پس از واکشی دستور بعدی وارد رجیستر IR که ورودی واحد جریان داده است می شود. مطابق آزمایش پنجم، برای تعیین هر دستور به ۶ بیت نیاز داریم که در شکل نیز معلوم است. نمای کلی مدار پیاده سازی شده در ادامه آمده است.



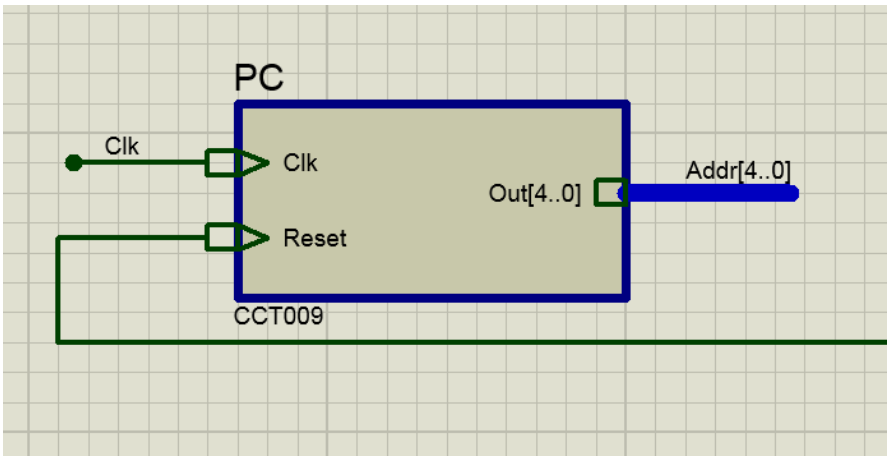
شکل ۲: نمای کلی سیستم پیاده سازی شده

۲ مراحل طراحی و پیاده‌سازی مدار

ماژول‌های مورد نیاز و شروع به کار مدار

همانطور که در شکل ۱ مشخص است، برای پیاده سازی سیستم فوق به یک شمارنده باینری ۵ بیتی نیاز داریم، به طوری که در هر واحد کلاک، یک واحد به آن اضافه شود تا دستور بعدی در حافظه دستورات را مشخص کند. به علاوه به یک EPROM با ۵ خط آدرس نیاز داریم. طول هر واحد حافظه ۱ بایت است که در نتیجه ظرفیت نهایی برابر با ۳۲ بایت می شود.

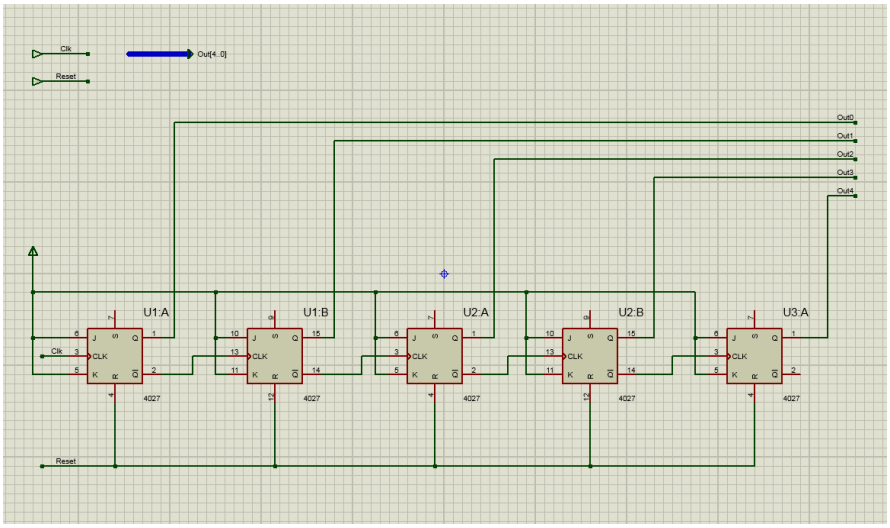
۱.۲ ماژول PC



شکل ۳: ماژول شمارنده ۵ بیتی

برای پیاده سازی این ماژول از معماری ripple استفاده شده است. به طور دقیق تر، شمارنده فوق از ۵ فلیپ فلاپ JK تشکیل شده است، به طوری که خروجی Q هر یک مشخص کننده یک بیت از

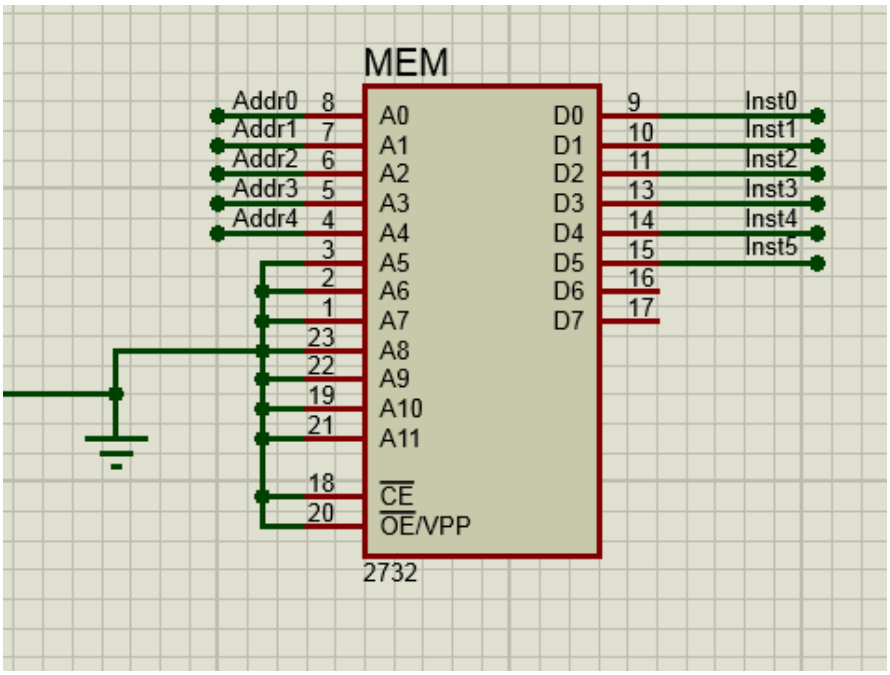
خروجی مدار است(خروجی های Out). هم چنین ورودی ها J و K تمامی آن ها به ورودی ۱ متصل هستند تا هر کدام در وضعیت Toggle قرار گیرند.



شکل ۴: نمای داخلی ماژول شمارنده ۵ بیتی

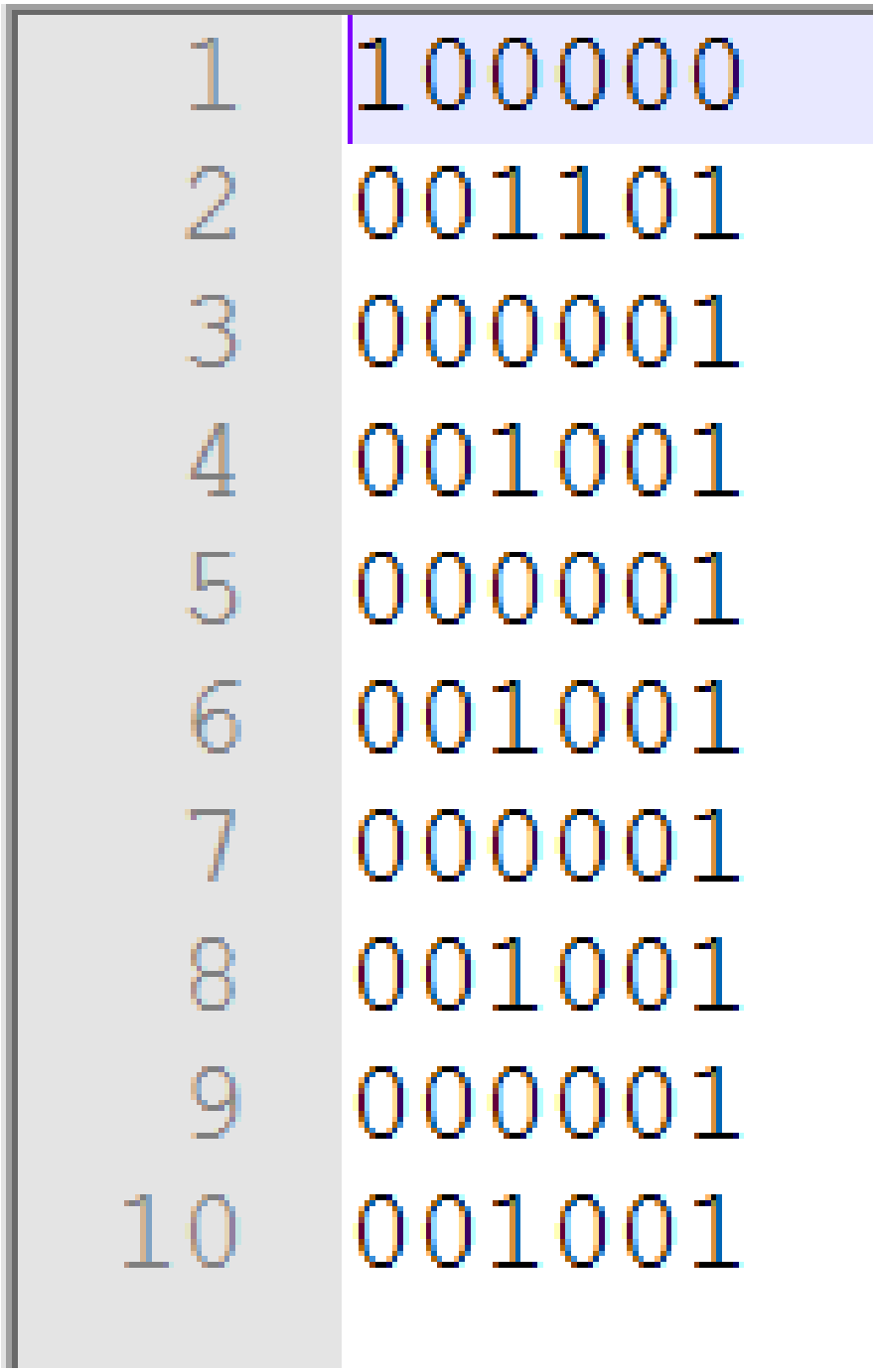
۲.۲ ماژول Memory

برای پیاده سازی این ماژول از تراشه آماده ۲۷۳۲ استفاده شده است. این تراشه یک Erasable Programmable ROM است که از ۱۲ خط آدرس و ۸ خط خروجی تشکیل شده است. با توجه به نیاز این آزمایش، تنها از خطوط آدرس A0 - A4 و خطوط خروجی D0 - D5 مورد استفاده قرار گرفته اند. لذا تمامی خطوط آدرس دیگر به ورودی صفر متصل هستند. در شکل زیر این ماژول را ملاحظه می کنید:



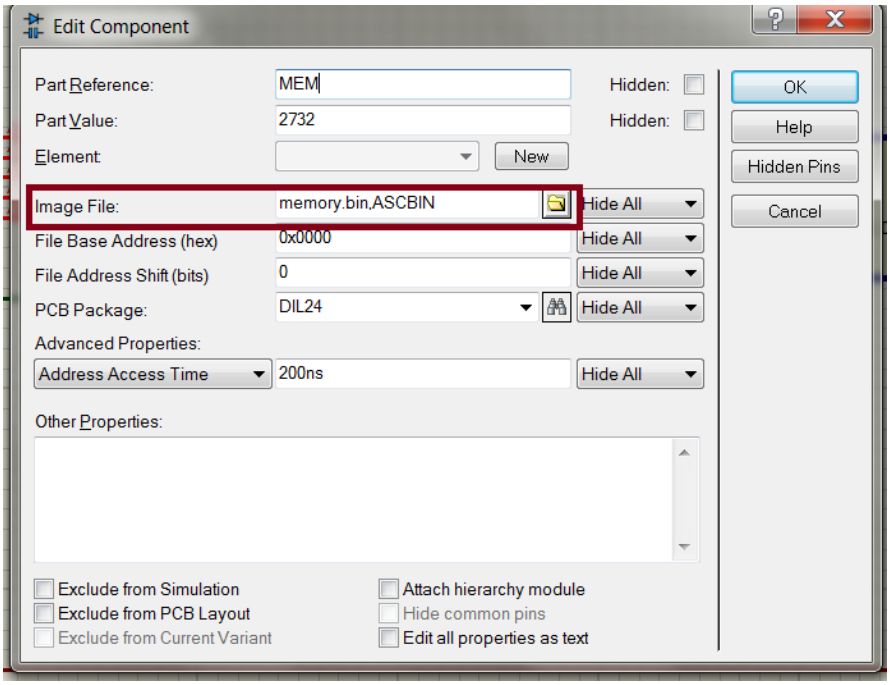
شکل ۵: ماژول حافظه دستورات

برای برنامه ریزی محتوای این ماژول، از فایل memroy.bin که هم سطح فایل پروژه واقع است استفاده کردی ایم. در این فایل محتوای هر یک از خطوط حافظه در یک خط مضخص شده اند. محتوای این فایل را در شکل زیر ملاحظه می کنید:



شکل ۶: محتوای فایل memory.bin

سپس برای شناساندن این فایل به ماژول، properties این تراشه را به صورت زیر به روز می کنیم:



شکل ۷: به روز رسانی تنظیمات تراشه ۲۷۳۲

در نهایت توجه داشته باشید که خروجی این حافظه که مشخص کننده دستور بعدی است، به ورودی های ماژول محاسبات ALU متصل است. (۲)ماژول فوق در آزمایش شماره ۵ پیاده سازی شده است.

۳ تست مدار

برای بررسی عملکرد و صحت سیستم طراحی شده می خواهیم برنامه محاسبه ۱۰ جمله ابتدایی سری فیبوناچی را پس از بارگذاری در حافظه دستورات اجرا کنیم. این دنباله به صورت زیر تعریف شده است:

$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

شکل ۸: سری فیبوناچی

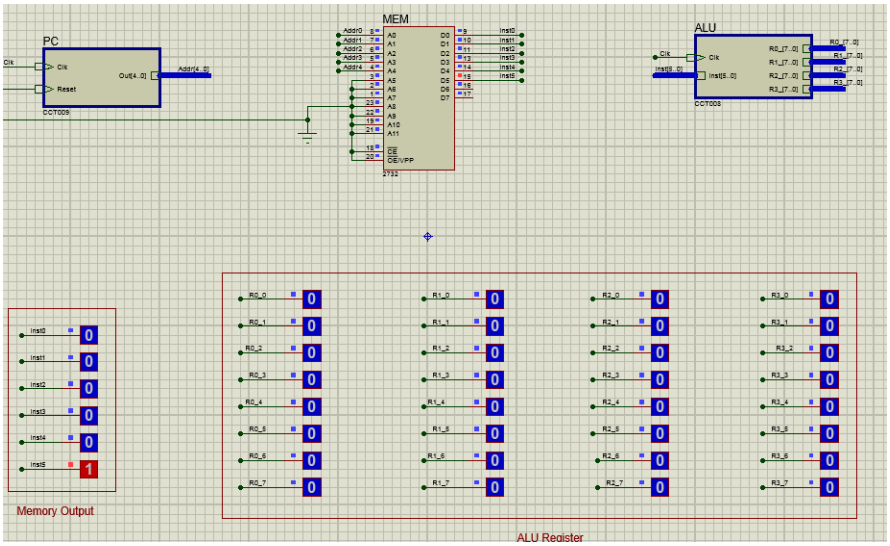
هم چنین قطعه برنامه زیر ده جمله ابتدایی این سری را در رجیسترهای R0 و R1 تولید و می کند، بدین صورت که در هر کلاک، دوجمله انتهایی دنباله تا آن لحظه در ثبات های مذکور ذخیره می شوند.

Address	Code	Instruction	Comment	
00000		Sub R0,R0	Clear R0	جمله اول در R0
		Add R1, 1	R1← 1	جمله دوم در R1
		Add R0,R1	R0←1	جمله سوم در R0
		Add R1,R1	R1←2	جمله چهارم در R1
		Add R0,R1	R0←3	جمله پنجم در R0
		Add R1,R1	R1←5	جمله ششم در R1
		Add R0,R1	R0←8	جمله هفتم در R0
		Add R1,R1	R1←13	جمله هشتم در R1
		Add R0,R1	R0←21	جمله نهم در R0
		Add R1, R1	R1←34	جمله دهم در R1

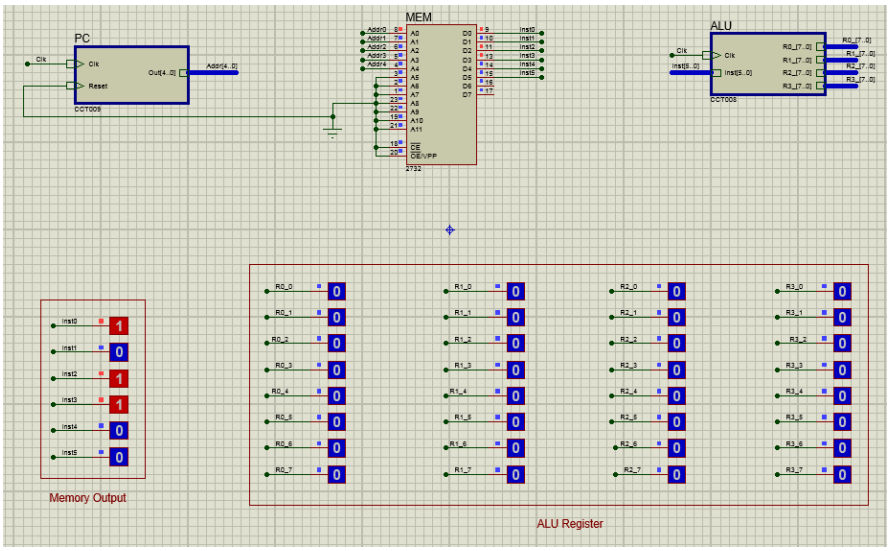
شکل ۹: برنامه محاسبه سری فیبوناچی

توجه داشته باشید که کد هر دستور در شکل ۶ قابل مشاهده است.

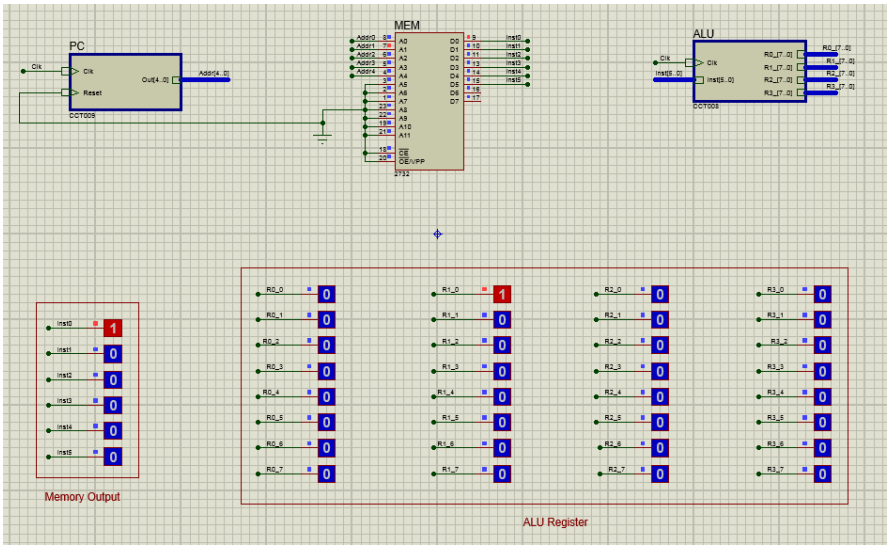
در ادامه وضعیت اجرای مدار در هر کلاک به همراه توضیحات مربوطه آمده اند. در کلیه اشکال، دستور بعدی در قسمتMemory Output قابل مشاهده می باشد.



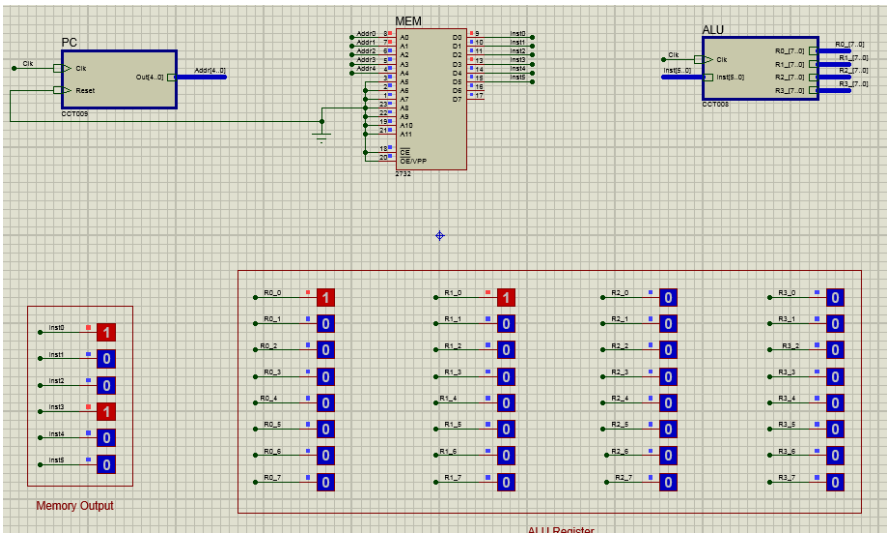
شکل ۱۰: وضعیت مدار در هنگام شروع



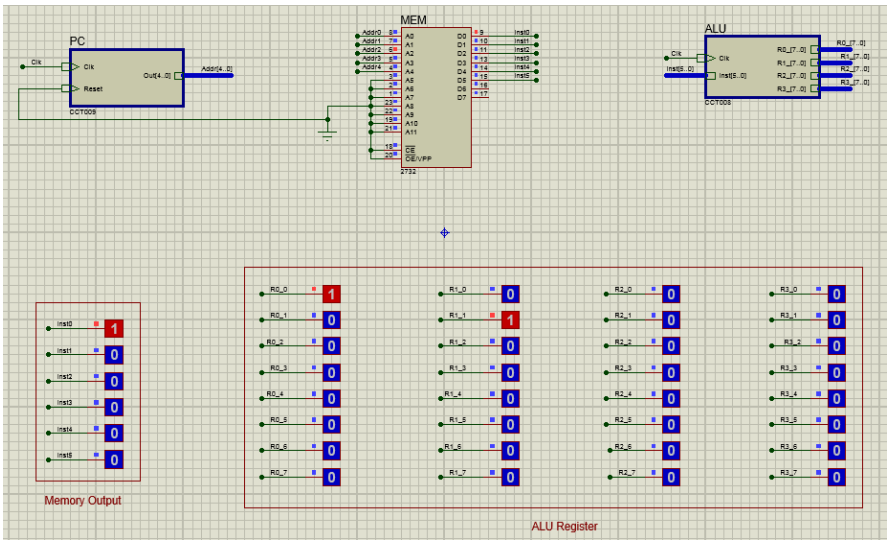
شکل ۱۱: Clear R0



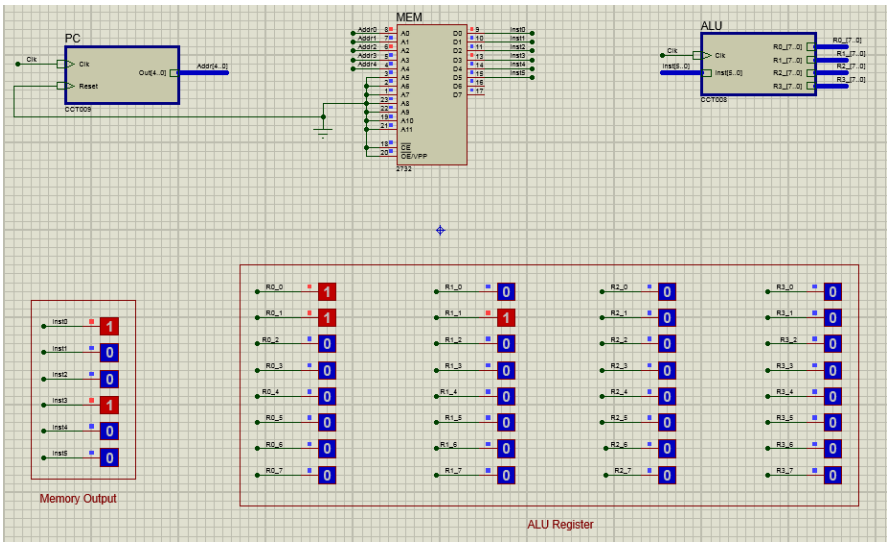
شکل ۱۲: $R1 \leftarrow 1$



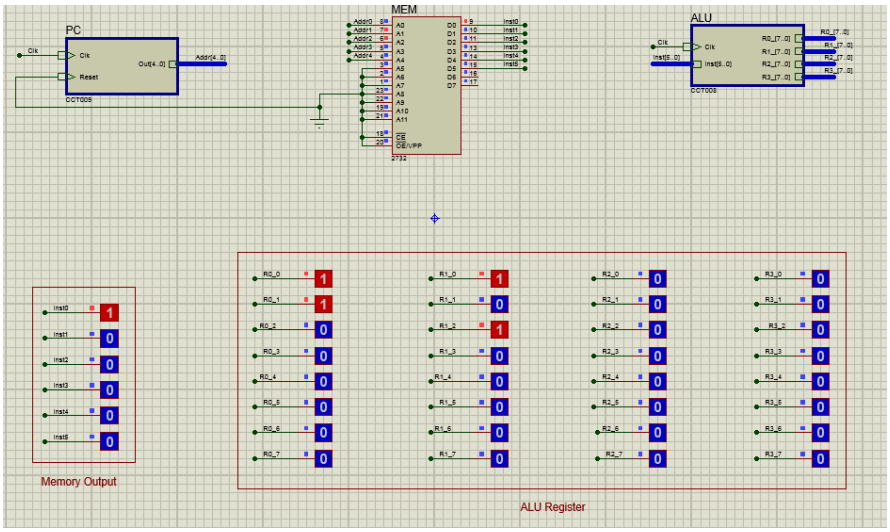
شکل ۱۳: $R0 \leftarrow 1$



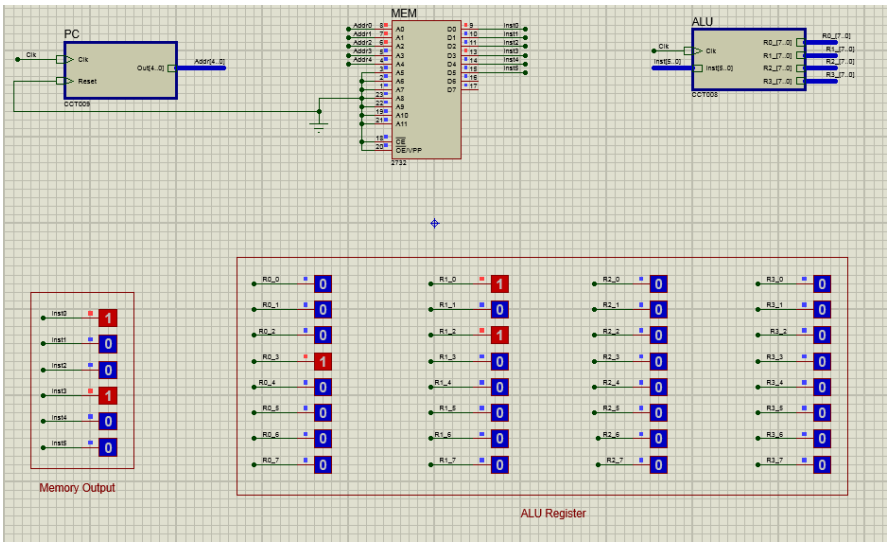
شکل ۱۴: $R1 \leftarrow 2$



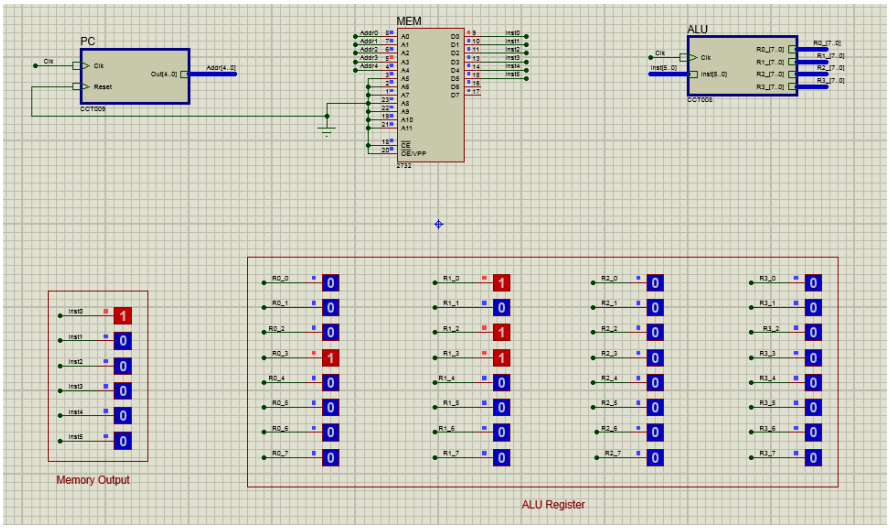
شکل ۱۵: $R0 \leftarrow 3$



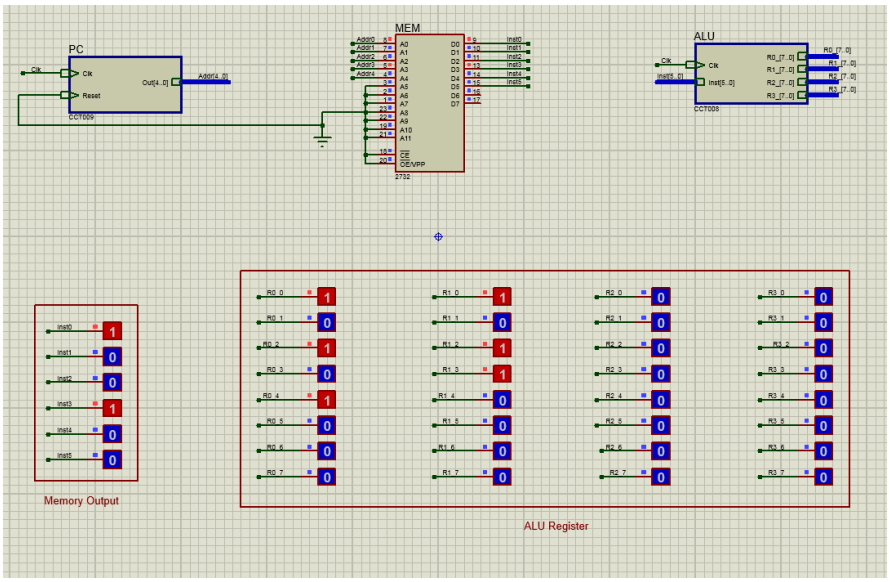
شکل ۱۶: $R1 \leftarrow 5$



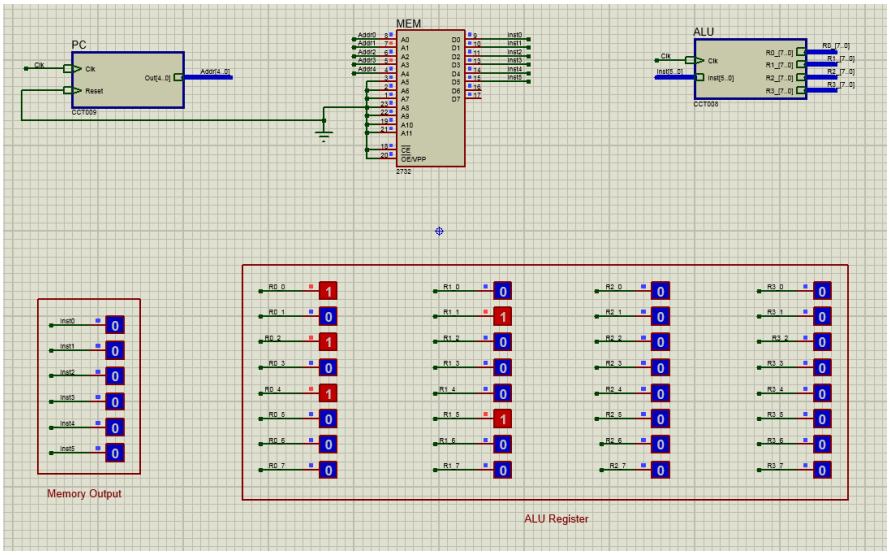
شکل ۱۷: $R0 \leftarrow 8$



شکل ۱۸: $R1 \leftarrow 13$



شکل ۱۹: $R0 \leftarrow 21$



شکل ۲۰: $R1 \leftarrow 34$