

به نام خدا



آزمایش شماره ۲

آزمایش معماری - دکتر سربازی آزاد

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شیراز

نیمسال اول ۰۱-۰۰

گروه:

امیرحسین هادیان - ۹۷۱۰۲۶۰۹

محمدرضا مفیضی - ۹۸۱۰۶۰۵۹

علی حاتمی تاجیک - ۹۸۱۰۱۳۸۵



۱ هدف

در این آزمایش قصد طراحی و پیاده‌سازی یک ضرب‌کننده چهاربیتی ممیز ثابت را داریم که دو عدد چهاربیتی (دو بیت قبل از ممیز و دو بیت بعد از ممیز) و یک سیگنال شروع را در ورودی دارد و پس از اینکه سیگنال شروع ارسال شد ضرب‌کننده شروع به کار می‌کند و حداکثر پس از شش پالس ساعت جواب را در خروجی باز می‌گرداند.

۲ طراحی

می‌دانیم که قصد تولید یک مدار ضرب‌کننده با الگوریتم Shift and Add داریم. ابتدا نیازهای مدار را بررسی می‌کنیم. دو عدد ورودی چهار بیتی خواهیم داشت و یک خروجی هشت بیتی (چهار بیت قبل ممیز و چهار بیت بعد از ممیز)، اما باید توجه داشته باشیم که برای عملیات شیفت به چپ که روی عامل اول ضرب انجام می‌شود نیاز به ۸ بیت فضا خواهیم داشت. همینطور رجیسترها نیاز به لود موازی دارند. این رجیسترها دارای قابلیت بارگذاری موازی و شیفت هستند (البته طراحی مدار بدون قابلیت شیفت هم ممکن بود. تنها کاری که نیاز بود انجام بشود این بود که سیگنال‌هایی خروجی ثبات را با یکی اختلاف، حالا کمتر یا بیشتر بستگی به نوع شیفت دارد، به ورودی لود رجیستر می‌دهیم و آن سیگنالی که شیفت را کنترل می‌کرد به جای اینکه به خود ثبات بدهیم آنرا به مالتیپلکسری وصل می‌کنیم که سیگنال‌های لود معمولی و لود شیفت را کنترل می‌کند. پس در پیاده‌سازی تأثیری آنچنانی نخواهد داشت اما برای جلوگیری از شلوغی شکل مدار سنتز شده نهایی از شیفت‌رجیسترهای دوطرفه استفاده شده است).

همینطور درباره ضرب اعداد با این الگوریتم می‌دانیم که روی اعداد علامت دار کاربردی ندارد پس اعداد ورودی ما هر دو بدون علامت هستند. یک سیگنال ورودی برای اعلام شروع ضرب به مدار نیاز داریم. همینطور دو سری سیگنال ۴ بیتی نیز برای ورودی گرفتن اعداد نیاز دارد. یک سیگنال نیز برای اعلام اتمام عملیات نیاز است. فرض شده است که سیگنال شروع تنها یک لحظه یک می‌شود و پس از آن تا دریافت جواب نهایی صفر خواهد بود و تا رسیدن مدار به حالت اولیه سیگنال ۱ برابر با ۱ باقی خواهد ماند و نیازی به بلاک فاینال نخواهیم داشت (این فرض تأثیری چنان بر روند کار نخواهد داشت چون تنها یک بلاک اضافه می‌شود و یک سیگنال ریست که آوردن آنها در مدار نهایی کار ساده‌ای خواهد بود. البته این کار به هر نوع جواب نهایی را به ما خواهد داد چون بالاخره دست از روی پوش‌باتن برداشته خواهد شد و اعداد ورودی در این میان ثابت می‌مانند اما ممکن است در این حین چندین بار عملیات ضرب تا انتها انجام شود و دوباره از سر گرفته شود).

پس تا اینجا درباره مدار می‌دانیم:

- دو ورودی ممیز ثابت چهاربیتی داریم (IN1, IN2)
- یک خروجی هشت بیتی ممیز ثابت داریم (C)
- یک سیگنال شروع برای عملیات داریم (S)
- یک سیگنال نمایش پایان عملیات داریم (F)

۱.۲ طراحی ASM

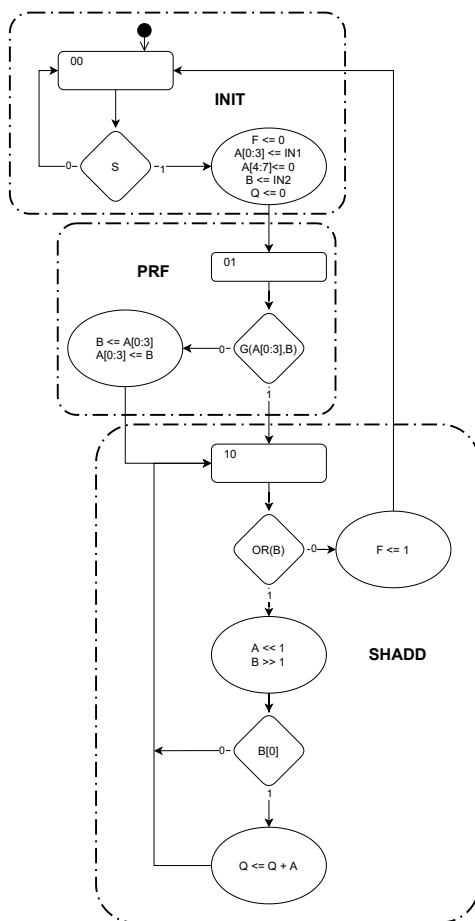
در ابتدای کار حالت‌های شکل ۱ در نظر گرفته شده بود. حالت‌های مختلف آن به شرح زیر است:

INIT زمانی که سیگنال شروع یک بشود، ابتدا در همان بلاک مقادیر اولیه ورودی‌ها در رجیسترها لود می‌شود. ۴ بیت مربوط به عامل دوم کامل به درون رجیستر B لود خواهد شد اما برای لود کردن A نیاز داریم که تعداد بیت آنرا اکستند کنیم چون مقدار این ثبات باید تا ۳۲ بیت قابل شیفت باشد و اطلاعات آن از بین نرود بنابراین در هنگام لود کردن ورودی اول ۱۶ بیت ورودی را در سمت کم‌ارزش بارگذاری کرده و ۱۶ بیت پرارزش را صفر می‌گذاریم. همینطور سیگنال اتمام کار را نیز صفر می‌کنیم تا در پایان کار دوباره یک بشود. رجیستر C نیز باید مقدار صفر را بگیرد تا هنگام جمع کردن آن با عامل اول مقدار غیر منتظره‌ای درون آن نباشد چون از این لحظه به بعد چیزی که تنها چیزی که در این ثبات باگذاری خواهد شد مقدار این ثبات به علاوه مقدار عامل اول شیفت‌داده شده است.

PRF سپس که ورودی‌ها را دریافت کردیم، برای افزایش کارایی مدار و حداقل زمان لازم برای انجام عملیات، در یک کلاک چک می‌کنیم که حتماً ورودی کوچکتر در عامل دوم (B) باشد.

SHADD این بلاک وظیفه اصلی عملیات ضرب را بر عهده دارد. در هر بلاک چند اتفاق به صورت همزمان می‌افتد:

- اگر عملیات به پایان رسیده باشد مقدار سیگنال پایان را یک کرده و به بلاک ابتدایی می‌رود.
- اگر عملیات به اتمام نرسیده باشد، عامل اول یکی به چپ و عامل دوم یکی به راست شیفت می‌خورد.



شکل ۱: نسخه ابتدایی طراحی شده

• اگر عملیات به اتمام نرسیده باشد و کم‌ارزشترین بیت عامل دوم نیز یک باشد عدد فعلی پاسخ با عدد فعلی عامل اول جمع می‌شود.

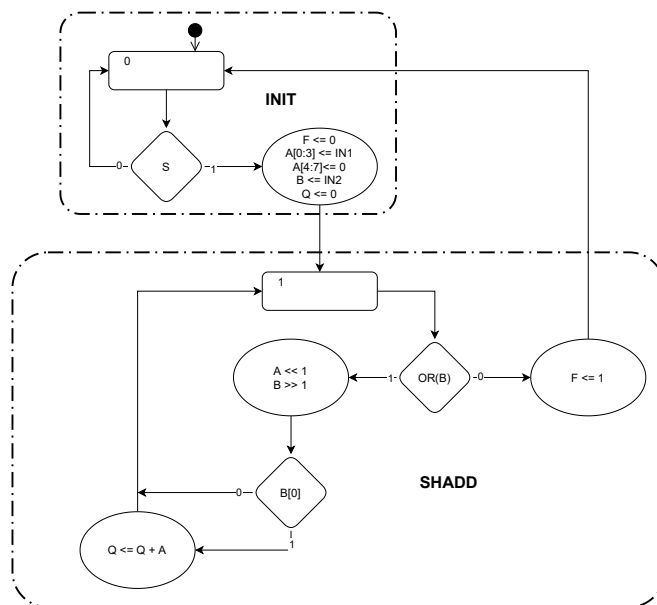
لازم به ذکر است که انجام عمل شیفت و جمع کردن در یک بلاک به صورت همزمان صورت می‌گیرد و این کار به خاطر تاخیر جزئی رجیسترها مشکلی برای ما بوجود نخواهد آورد. مثل این می‌ماند که اگر شرایط برقرار بود ابتدا عمل جمع صورت بگیرد و سپس شیفت‌ها به میزان لازم انجام بشود.

پایان یافتن عملیات هنگامی رخ می‌دهد که تمام بیت‌های عامل دوم صرف شده باشند (صفر شدن بیت‌های عامل اول غیر ممکن است مگر اینکه هر دو آنها از ابتدا صفر بوده باشند که باز تغییری در شرط ایجاد نخواهد کرد). همین‌طور طبق الگوریتم تنها زمانی باید عمل جمع انجام بشود که کم‌ارزشترین بیت عامل دوم، یک باشد.

اما در زمان سنتز اولیه این چارت به مشکلی برخوردیم و آن نبود تراشه‌های مناسب مالتیپلکسر TTL بود که زمانی که چند مورد ورودی برای یک شیفت رجیستر داشتیم باید از آنها استفاده می‌کردیم. به همین دلیل بلوک پرفورمنس را از چارت خارج کردیم تا چارت نهایی به شکل ۲ دربیاید (توضیحات همان است فقط بلوک کارایی حذف شده است و استیپ‌ها به دو کاهش یافته است).

۲.۲ سنتز و ساخت مدار

در این مرحله به سنتز و ساخت مدار می‌پردازیم. در ابتدای کار مدار اصلی را طبق چارت طراحی شده پیاده‌سازی می‌کنیم. مدار از دو بخش اصلی (Data Path (DP) و Control Unit (CU) تشکیل شده است که خطوط اصلی مدار در قسمت DP تشکیل شده و اجزای نگه‌دارنده



شکل ۲: نسخه اصلاح شده

داده و انتقال دهنده آنها در این بخش وجود دارد. خطوط کنترل این خطوط و کنترل وضعیت مدار در بخش CU وجود دارد. سیگنال‌های کنترلی از بخش کنترل به بخش مسیر داده می‌روند. در ادامه ابتدا تراشه‌هایی که مورد استفاده قرار گرفته است بررسی می‌شوند و سپس بخش‌های مختلف مدار بررسی می‌شوند.

۱.۲.۲ تراشه‌هایی که مورد استفاده قرار گرفته است

74283 این تراشه یک جمع کننده چهار بیتی است که برای ساخت یک جمع کننده هشت بیتی از دوتای آن استفاده کرده‌ایم و به صورت آبشاری به هم متصل کرده ایم. نامگذاری پایه‌ها به صورت رایج انجام شده بود و نیازی به خواندن دیتاشیت نبود.

74194 این یک شیفت رجیستر چهار بیتی است که چهار عمل برای آن در نظر گرفته شده است: بدون عمل، شیفت به راست، شیفت به چپ و بارگذاری داده. دارای یک ریست آسنکرون است. و بیتی که ورودی شیفت در آن قابل تعیین است (که ما بیت صفر را اضافه می‌کنیم). اعمال آن با توجه به خط سلکت آن به این شرح است:

$S_1 S_0$

00 → NOP

01 → Left

10 → Right

11 → Load

74198 این یک شیفت رجیستر هشت بیتی با رفتار کاملاً شبیه با تراشه ۷۴۱۹۴ است.

2732 این تراشه یک EPROM است و از آن برای دیکود کردن سون سگمنت‌هایی که برای نمایش نتیجه می‌خواستیم استفاده شده است. فایل‌های مربوط به داده شده به آن را به صورت دستی تهیه کرده ایم و به این صورت که هر کدام برای دو سون سگمنت استفاده می‌شود. برای ۱۶ آدرس ابتدایی آن داده در نظر گرفته ایم.

7404 این تراشه یک شامل شش گیت NOT است که در ساخت بخش‌هایی از مدار استفاده شده است. چون به صورت گیت منطقی در نرم‌افزار آمده بود نیازی به مطالعه دیتاشیت نبود.



7408 این تراشه یک شامل چهار گیت AND است که در ساخت بخش‌هایی از مدار استفاده شده است. چون به صورت گیت منطقی در نرم‌افزار آمده بود نیازی به مطالعه دیتاشیت نبود.

7432 این تراشه یک شامل چهار گیت OR است که در ساخت بخش‌هایی از مدار استفاده شده است. چون به صورت گیت منطقی در نرم‌افزار آمده بود نیازی به مطالعه دیتاشیت نبود.

7474 این تراشه یک شامل دو فلیپ‌فلاپ نوع D است که در ساخت بخش‌هایی از مدار استفاده شده است. چون به صورت فلیپ‌فلاپ تک و با نام‌گذاری رایج در نرم‌افزار آمده بود نیازی به مطالعه دیتاشیت نبود.

۲.۲.۲ طراحی مسی‌ر داده

در این بخش باید داده‌ها و روابط بین آنها را که در دسیژن‌باکس‌ها آمده است را بیاوریم. با توجه به شکل ۲ این بخش را سنتز می‌کنیم:

A یک شیفت رجیستر ۸ بیتی که چهاربیت داده ورودی آن به عدد اول و چهاربیت پرارزشر آن به عدد ثابت صفر (زمین) متصل است.

B یک شیفت رجیستر ۴ بیتی که داده ورودی آن به ورودی دوم متصل است. از خروجی آن برای تولید دو سیگنال کنترلی استفاده می‌کنیم.

C یک شیفت رجیستر ۸ بیتی که جواب نهایی را در خود نگه‌میدارد و ورودی آن هم خروجی جمع‌کننده هشت بیتی است.

• یک جمع‌کننده که همواره مقدار درون شیفت رجیستر A و رجیستر C را در خود نگه‌میدارد.

همچنین به سیگنال‌های وضعیت زیر نیز نیاز خواهیم داشت:

B0 این سیگنال بیت اول شیفت رجیستر دوم است.

ORB این سیگنال «یا» منطقی چهار بیت شیفت رجیستر دوم است.

S این سیگنال شروع کار است.

۳.۲.۲ طراحی واحد کنترل

در این بخش باید سیگنال‌های کنترلی را برای مسی‌ر داده تولید کنیم. همچنین از یک فلیپ‌فلاپ نوع دی برای نگهداری وضعیت فعلی مدار استفاده می‌کنیم. هر شیفت رجیستر (تراشه‌های ۷۴۱۷۹۴/۷۴۱۹۸) نیاز به دو سیگنال انتخاب (تغییر بین کارایی‌ها طبق بخش ۱.۲.۲) و یک سیگنال ریست دارند. یک سیگنال برای ورودی و یک سیگنال برای ریست فلیپ‌فلاپ F و همین‌ها را برای فلیپ‌فلاپی که حالت را نگه می‌دارد نیز نیاز خواهیم داشت.

از آنجایی که شیفت و لود رجیسترهای A, B با یکدیگر انجام می‌شود از یک سیگنال تولید کرده و از آن برای هردو (با تغییرات جزئی در نحوه فرستادن آنها) استفاده می‌کنیم (Q نمایش دهنده خروجی فلیپ‌فلاپ استیت‌های ماست).

$$\text{loadAB} = \overline{Q} \times S$$

$$\text{shiftAB} = Q \times \text{OR}(B)$$

$$\text{clearAB} = 1 \text{ (Never will be cleared)}$$

$$\text{clearC} = \overline{Q} \times S = \text{loadAB}$$

$$\text{loadC} = Q \times \text{OR}(B) \times B[0]$$

$$\text{shiftC} = 0 \text{ (Never will be shifted)}$$

با استفاده از این روابط سیگنال‌های مربوط به رجیسترها را به دست می‌آوریم:

$$A_{S0} = \text{loadAB} + \text{shiftAB}, A_{S1} = \text{loadAB}$$

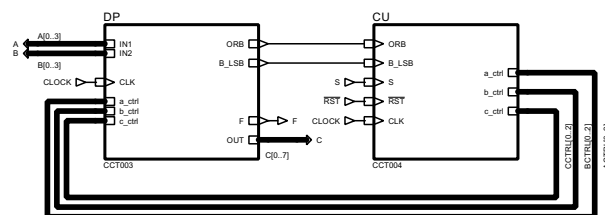
$$B_{S0} = \text{loadAB} = A_{S1}, B_{S1} = \text{loadAB} + \text{shiftAB} = A_{S0}$$

$$C_{S0} = \text{loadC}, C_{S1} = \text{loadC}$$

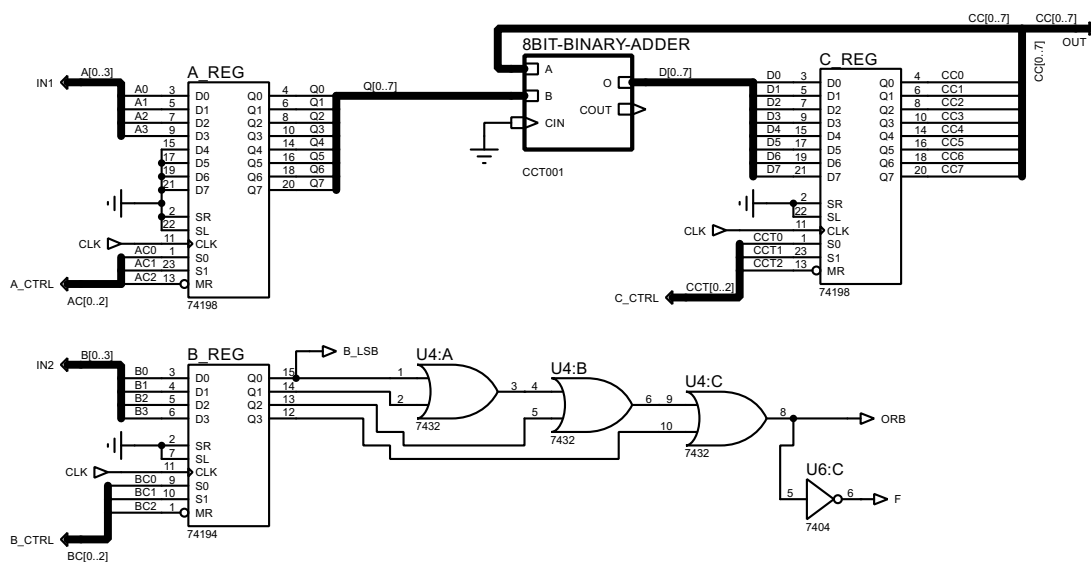
$$C_{\text{clear}} = \overline{\text{clearC}}$$

برای سیگنال F هم صفر شدن B را در نظر می‌گیریم.

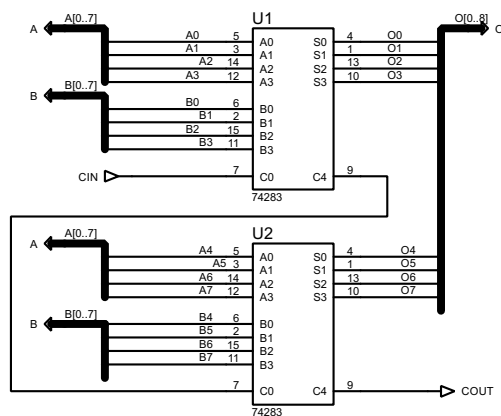
با یک نگاه به جدول حالت می‌توانیم بگوییم که $Q_D = (Q + S)(\overline{Q} + \text{OR}(B))$ (دقت کنید که سیگنال ست در زمانی که می‌خواهیم ست رخ بدهد صفر میشود نه یک).



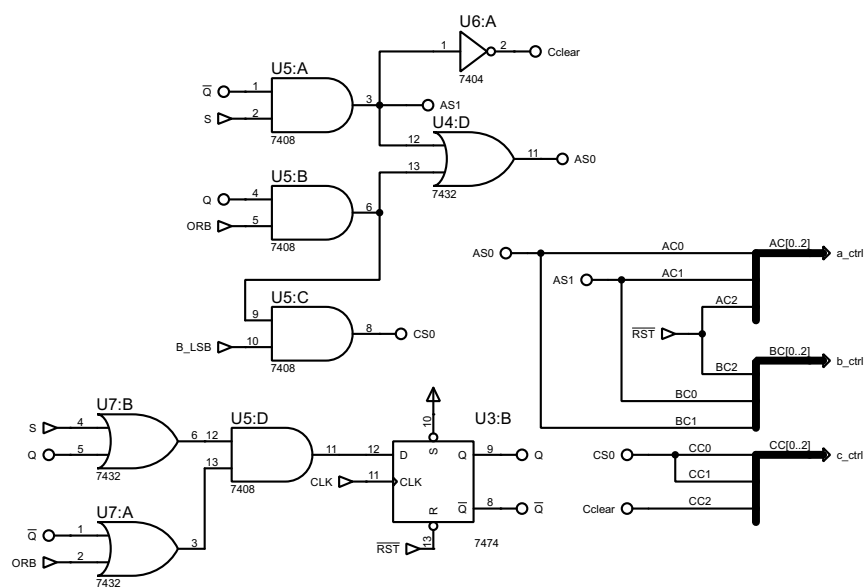
شکل ۳: داخل ضرب کننده چهار بیتی



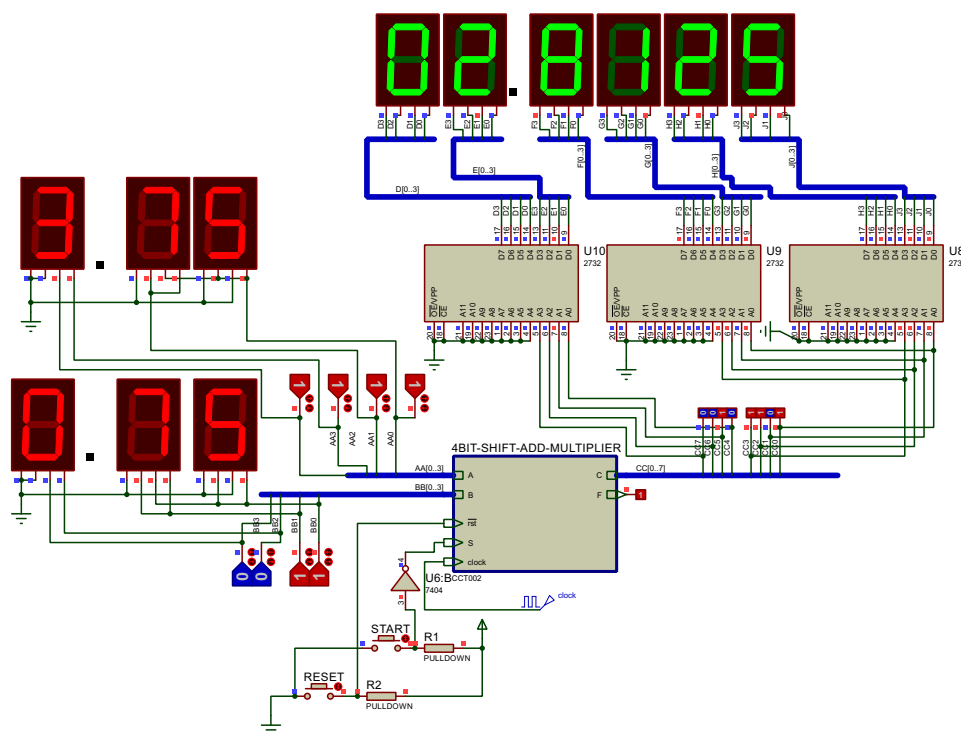
شکل ۴: مسیره داده



شکل ۵: جمع کننده استفاده شده در مسیرداده



شکل ۶: واحد کنترل



شکل ۷: تست ۱

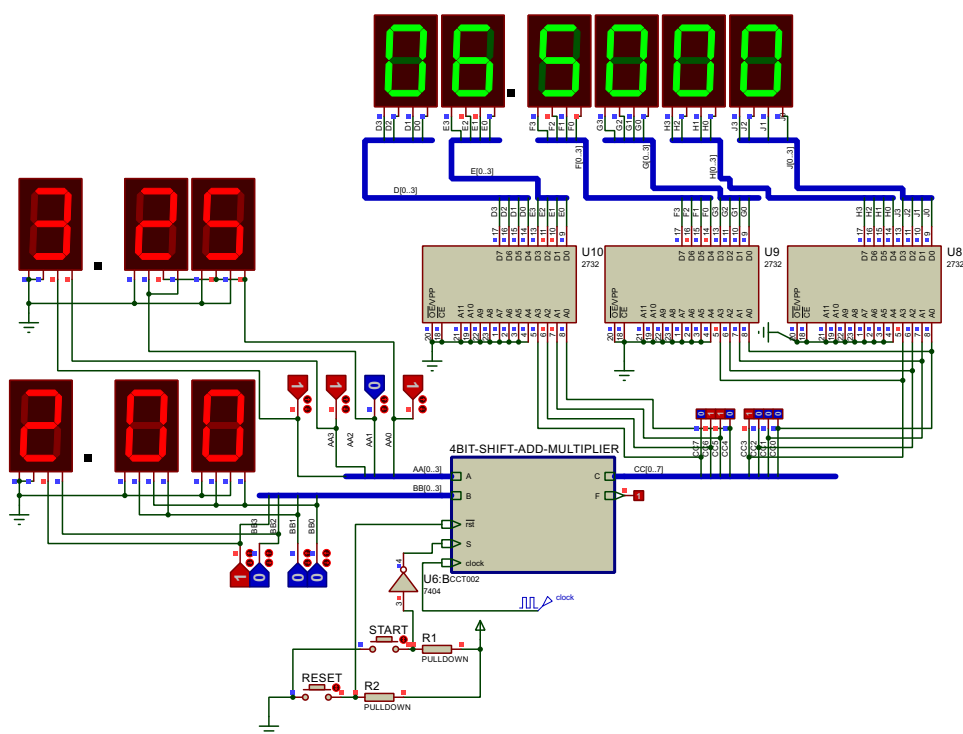
۳.۲ پیاده‌سازی

با توجه به اینکه مشخصات طراحی توضیح داده شده است در این بخش فقط تصاویر آنرا می‌آوریم (تصاویر در دو صفحه قبل آمده‌اند).

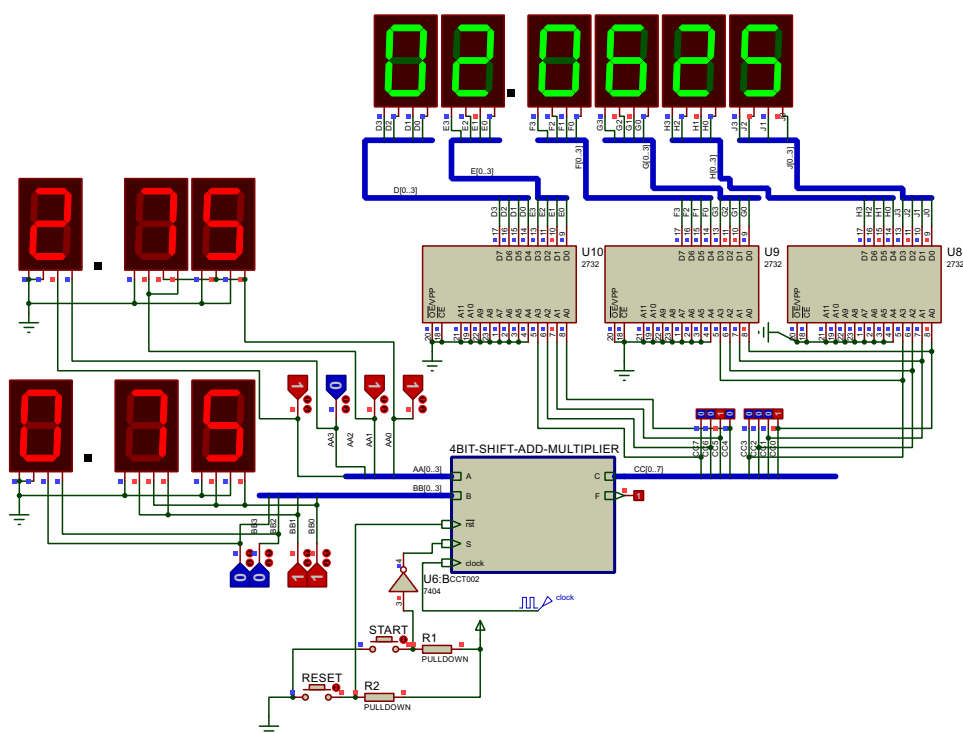
۳ تست

برای نمایش نتایج تست از 7 Segments استفاده شده است. برای اعداد ورودی قسمت غیراعشاری به صورت مستقیم متصل شده‌اند. قسمت اعشاری نیز به این صورت وصل شده‌اند که اگر کم ارزش ترین بیت روشن باشد 0.25 را خواهیم داشت و اگر بیت دوم روشن باشن 0.5 را خواهیم داشت که سیگنال مورد نیاز برای اینها تداخلی ندارد و می‌توان آنها را به صورت مستقیم به سون سگمنت‌ها متصل کرد. برای نمایش خروجی‌ها از سه EPROM استفاده کرده‌ایم که یک دیکودر هستند. دوتای آنها برای دیکود کردن ۴ بیت اعشار و دیگری برای دیکود کردن قسمت صحیح جواب است. داده نوشته شده در آنها در کنار فایل گزارش موجود است. هر کدام از آنها برای نمایش دو رقم دهدهی کافی هستند.

در ادامه چند تست انجام شده را مشاهده خواهید کرد که صحت آنها از نمایشگرها مشخص است:



شکل ۸: تست ۲



شکل ۹: تست ۳