به نام خدا

آزمایشگاه معماری

آزمایش پنجم:تبدیل اعداد ۳ رقمی BCD به اعداد باینری

اعضای گروه:

امیراردلان دهقانپور 401105901

رادین شاهدایی 401106096

باربد شهرآبادی 401106125

مقدمه:

در این آزمایش قصد داریم که یك مدار درست کنیم تا اعداد سه رقمی BCD را به اعداد باینری با حداکثر ۱۰ بیت تبدیل کند،برای این کار طبق الگوریتمی که در بخش شرح آزمایش گفته شده است عمل می کنیم و برای تبدیل کردن اعداد BCD به اعداد باینری در هر مرحله عدد را یك بیت به راست شیفت می دهیم و سپس چك می کنیم که آیا بیت پرارزش هر رقم(به عبارتی بیت های ٤ و ٨ و١٢) یك هستند و یا خیر،اگر این بیت ها یك بودند عدد ۳ را از آن رقم یا به عبارتی آن ٤ بیت کم می کنیم و آنقدر این کار را انجام می دهیم تا ۱۲ بیت اولیه ای که داشتیم همگی صفر شوند،می دانیم که این کار حداکثر ۱۰ کلاك به طول می انجامد،می توانیم این الگوریتم را برای تبدیل عدد ۲۲ به صورت باینری به صورت زیر بنویسیم:

الگوریتم:

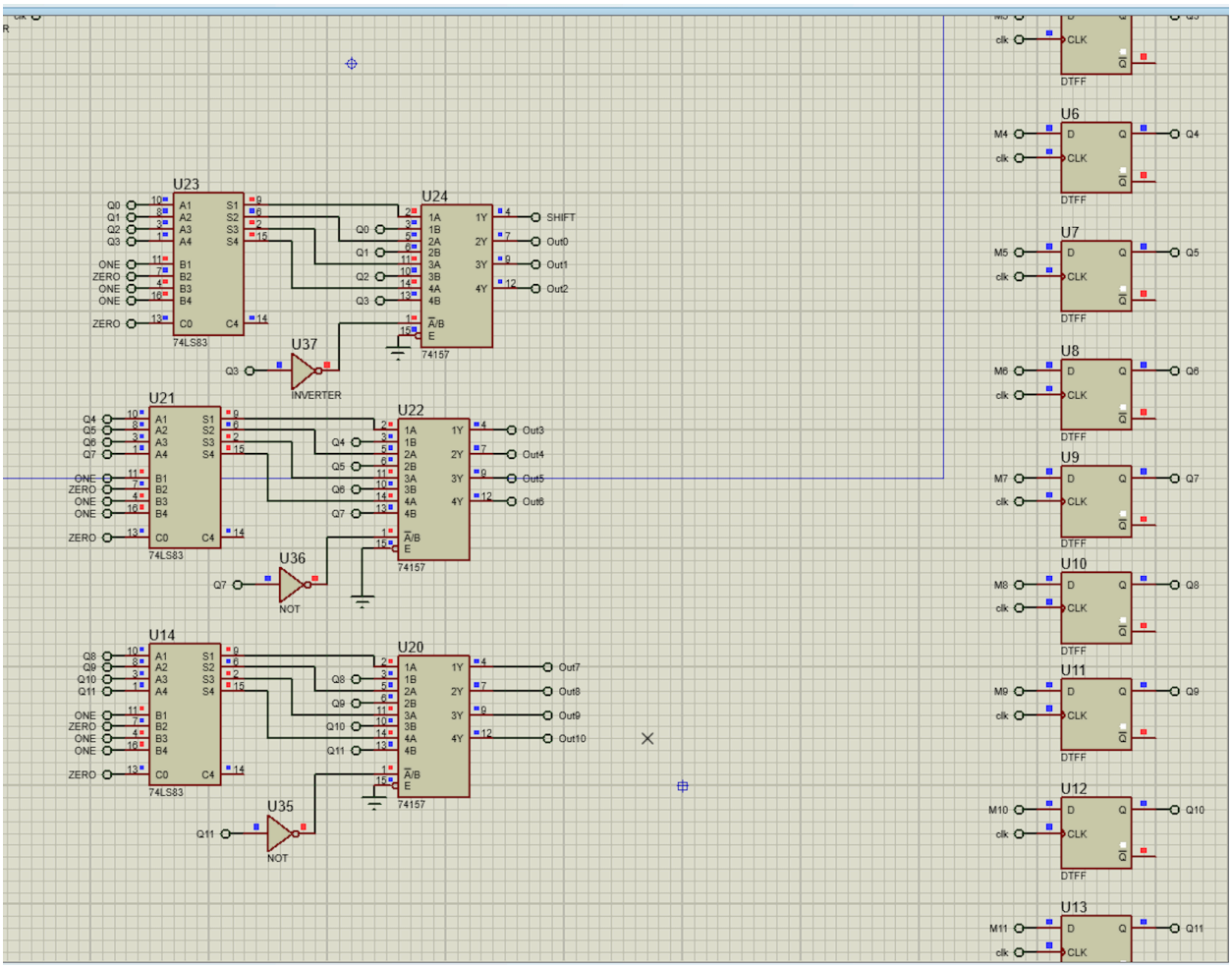
عمل	خروجی	رقم ۱	رقم ۲	رقم ۳
شیفت به راست	0	0010	0010	0000
شیفت به راست	1	0001	0001	0000
کم کردن ۳		1000	0000	0000
شیفت به راست	1	0101	0000	0000
شیفت به راست	0	0010	0000	0000
شیفت به راست	1	0001	0000	0000
پایان عملیات		0000	0000	0000

همانطور که مشاهده می کنیم به عدد ۱۰۱۱۰ رسیده ایم که معادل باینری ۲۲ است،حال به سراغ پیاده سازی این الگوریتم در نرم افزار پروتیوس می رویم.

پیاده سازی:

برای این کار باید یك رجیستر ۱۲ بیتی داشته باشیم که برای این کار از ۱۲ بیت d flip flop استفاده کرده ایم،سپس در پیاده سازیمان به این صورت عمل می کنیم که برای هر کدام از ارقام(٤ بیت) از یك

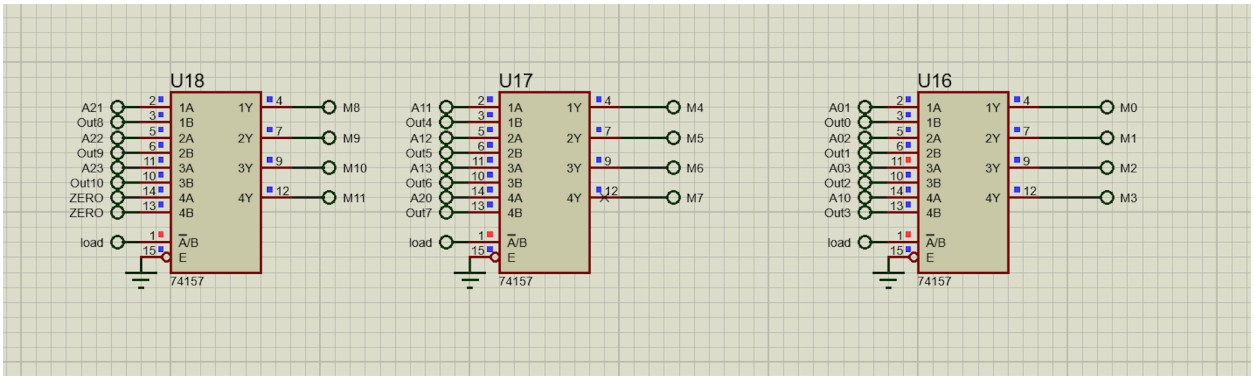
مالتی پلکسر استفاده می کنیم که ببینیم آیا صرفا کافی است آن رقم را شیفت بدهیم و یا عدد ۳ را از آن کم کنیم،که برای کم کردن ۳ کاری معادل با آن یعنی جمع کردن با ۱۳ را انجام داده ایم،پس از ۳ مالتی پلکسر ٤ بیتی در این بخش استفاده کرده ایم،که بر اساس بیت پر ارزش هر کدام از ارقام انتخاب می کند تا از کدام ٤ بیت استفاده کند که این بخش از مدار به صورت زیر طراحی شده است:



شکل ۱

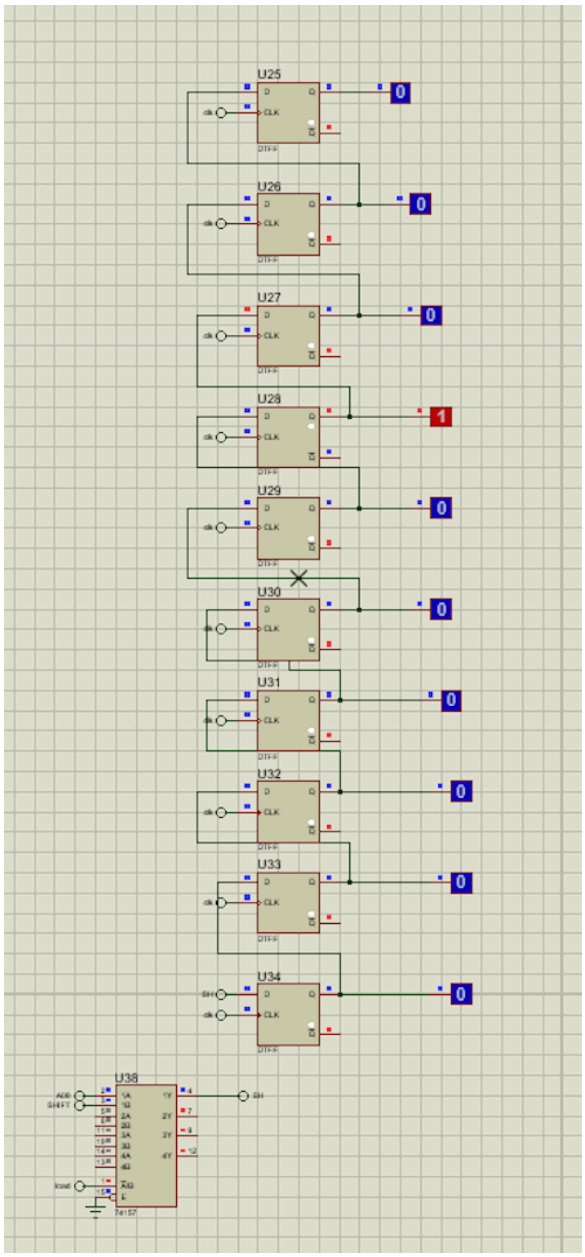
در مدار نیز این ۳ مالتی پلکسر و بیت های سلکت آنها و همچنین d flip flop ها مشخص هستند،در ادامه بخش دیگری از مدار را توضیح می دهیم که همان ورودی های flip flop ها هستند،این ورودی ها از یك بخش بالاتر در مدار می آیند که در آنجا نیز ۳ مالتی پلکسر داریم که بیت انتخابی آنها متصل به کلید start است که اگر صفر باشد در اولین کلاك عدد ورودیمان وارد flip flop ها می شوند و در غیر این صورت

حاصل شیفت به راست خورده ۳ مالتی پلکسر پایینی به آن وارد می شوند که این بخش نیز مطابق با شکل زیر طراحی شده است:



شکل ۲

در طراحی مدار نیز این بخش به صورت بخش بالا طراحی شده است که بیت های A01 و...،بیت های ورودی هستند و همچنین بیت های Out0 و... نیز بیت های شیفت خورده از مالتی پلکسرهای پایین هستند بخش دیگری که باید آن را پیاده سازی کنیم بخش ذخیره کردن عدد باینری است برای این بخش پیاده سازی آن به این صورت است که از ۱۰ عدد d flip flop استفاده کرده ایم که هرکدام به جز بیت سمت راست به ورودی فلیپ فلاپ کناری آن متصل هستند پس به این صورت داریم که در هر کلاك یك بیت عدد قبلی به راست شیفت می خورد و بیت جدید نیز برحسب عددی که در آن لحظه داریم وارد آن می شود برای این که طبق این روش جوابمان به صورت باینری کامل شود به دقیقا ۱۰ کلاك نیاز داریم،پیاده سازی این بخش از مدار به صورت زیر می باشد:

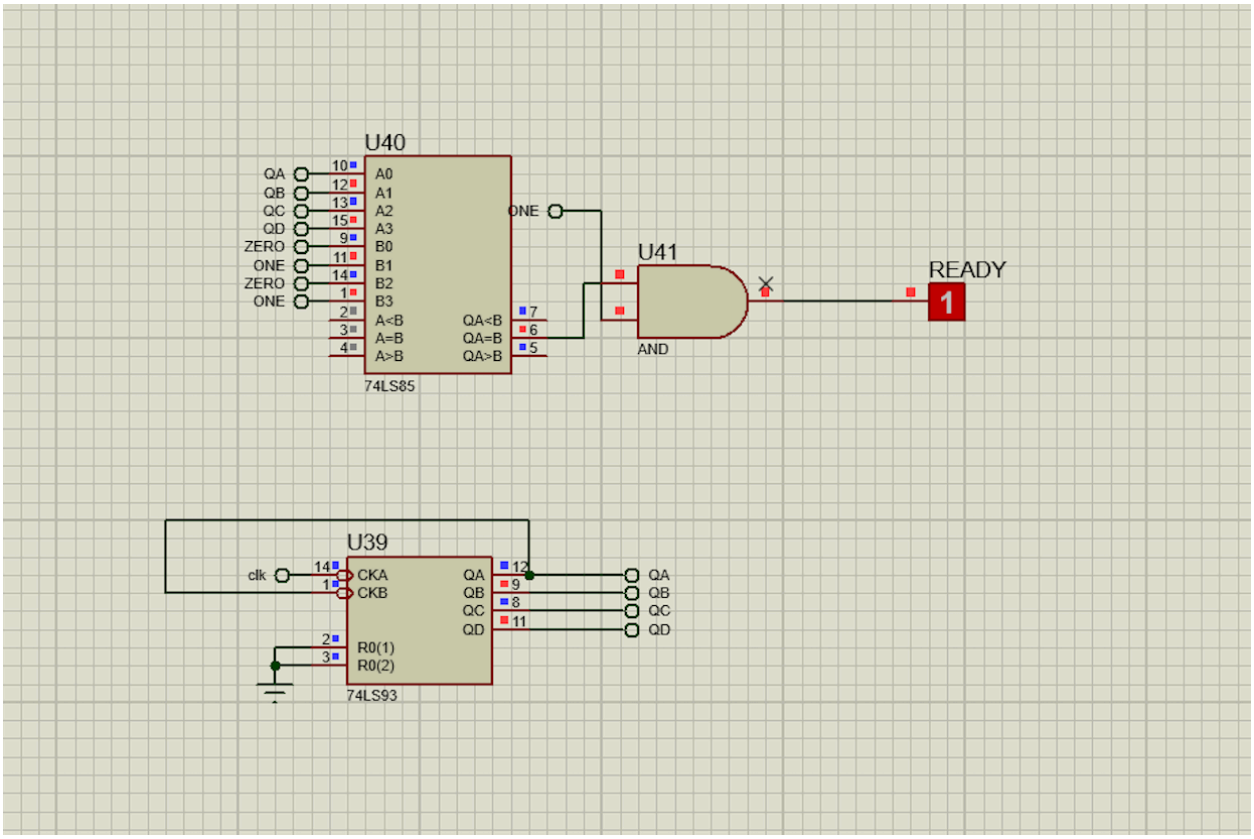


شکل ۳

مالتی پلکسر پایین نیز کارکردی که دارد این است که چك می کند که همواره در اولین کلاك بیت سمت راست عدد ورودی وارد آن شود و به خاطر کم کردن عدد ۳ از آن دچار اشتباه نشود.

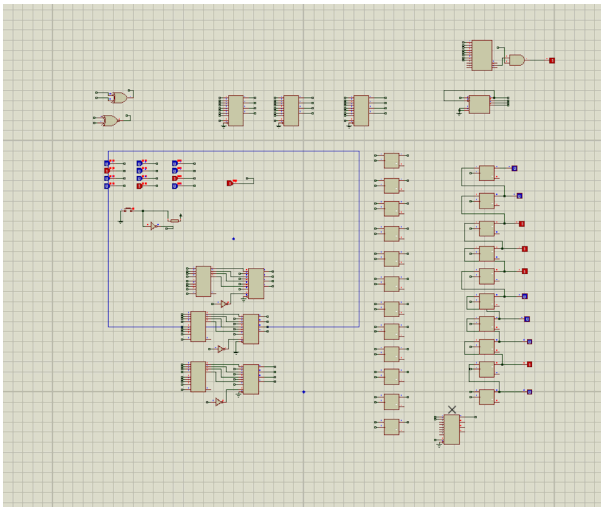
فعال شدن سیگنال end: برای این بخش نیز کاری که می کنیم به این صورت است که یك شمارنده گذاشته ایم تا بعد از هرکلاك یك واحد زیاد شود و هنگامی که به ۱۰ رسید،سیگنال برابر بودن در مقایسه

کننده فعال شود و در اثر آن این سیگنال برابر با یك شود و طراحی این بخش از مدار نیز به صورت زیر می باشد:



شکل ٤

یك شمای کلی از کل مدار نیز به صورت زیر می باشد:

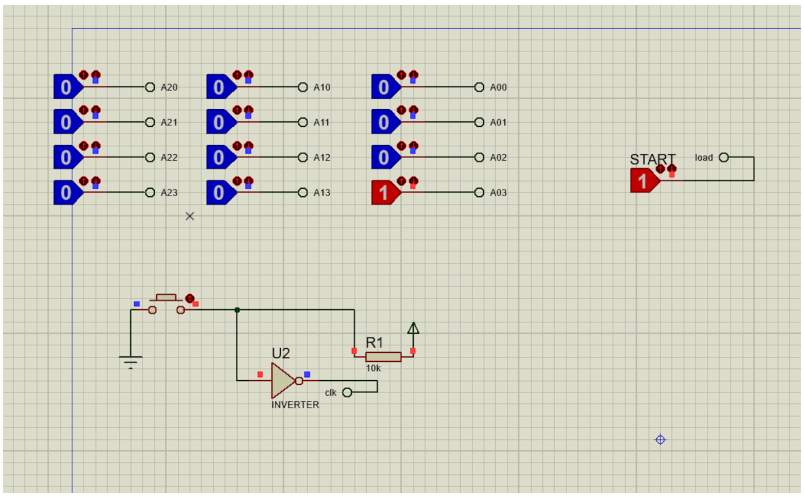


شکل ۵

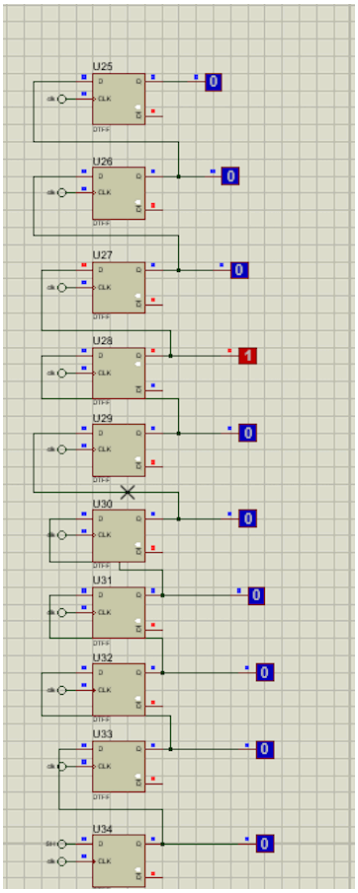
تست کردن مدار:

در اینجا دو ورودی مختلف را به مدار می دهیم و مدار را تست می کنیم.

1.تبدیل عدد ۸:

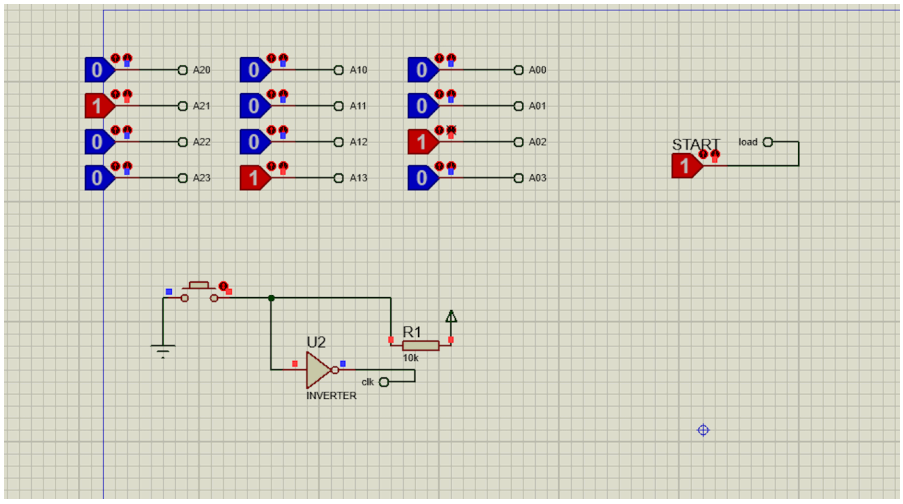


شکل٦

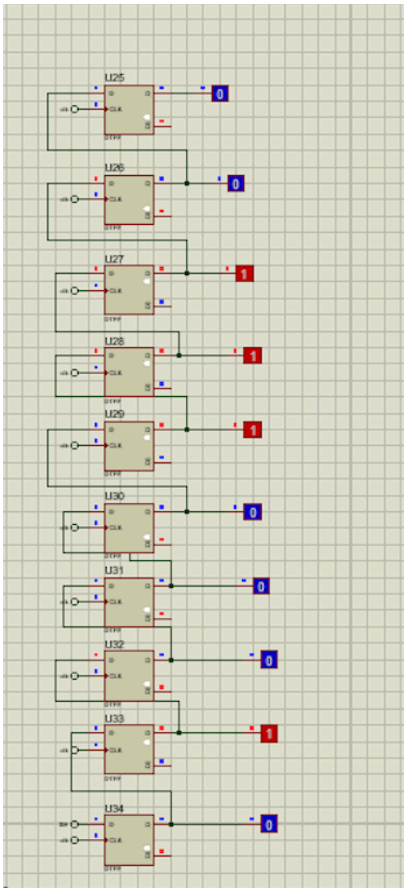


شکل ۷

2.تبدیل عدد ۲۸٤:



شکل ۸



شکل ۹

همانطور که مشخص است معادل باینری عدد ۲۸٤ به صورت ۱۰۰۰۱۱۱۰۰ در آمده است.