



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر  
گزارش کار پنجم درس آزمایشگاه معماری

عنوان:

مبدل دهدهی به دودویی

**BCD To Binary Convertor**

نگارش

کیان قاسمی ۴۰۱۱۰۲۲۶۴  
آیین پوست فروشان ۴۰۱۱۰۵۷۴۲  
دیبا هادی اسفنگره ۴۰۱۱۱۰۲۴۵

استاد

دکتر حمید سربازی آزاد

دستیار آموزشی

مهندس عطیه غیبی فطرت

مرداد ۱۴۰۳

## فهرست مطالب

۳	۱	مقدمه
۳	۱.۱	قطعات لازم
۴	۲.۱	بلوک دیاگرام نهایی
۴	۳.۱	شرح الگوریتم
۵	۲	شبیه سازی مدار
۵	۱.۲	واحد کنترل
۷	۲.۲	واحد $SHIFT - AND - ADD$
۸	۳.۲	واحد شیفت رجیستر
۹	۴.۲	واحد <i>validator</i>
۱۰	۵.۲	آزمایش مدار و بررسی درستی عملکرد آن
۱۳	۳	چالش ها
۱۳	۴	نتیجه و بحث

## فهرست تصاویر

۴	۱	بلاک دیاگرام کلی
۵	۲	واحد کنترل
۶	۳	سیگنال های <i>shift</i> و <i>Add</i>
۶	۴	شمارنده در واحد کنترل
۷	۵	واحد <i>Shift And Add</i>
۸	۶	واحد شیفت رجیستر
۹	۷	واحد <i>validator</i>
۹	۸	شکل نهایی مدار
۱۰	۹	تست اول
۱۱	۱۰	تست دوم
۱۱	۱۱	تست سوم
۱۲	۱۲	تست چهارم

## ۱ مقدمه

در این آزمایش قصد داریم یک مبدل دهمی به دودویی را با استفاده از نرم افزار پروتئوس شبیه سازی کنیم. سه رقم دهمی و سیگنال *start* به عنوان ورودی داده می شوند و ۱۰ رقم دودویی و سیگنال *end* خروجی مدار هستند. سلسله مرتب کلی طراحی این مدار به صورت زیر است:

۱. ساخت شیفت رجیستر ۱۰ بیتی برای ذخیره نتیجه نهایی
۲. ساخت واحد شیفت دهنده و جمع کننده ارقام دهمی
۳. ساخت واحد کنترل برای تولید سیگنال های کنترلی دو واحد اول
۴. ساخت واحد *validator* برای بررسی معتبر بودن ارقام ورودی

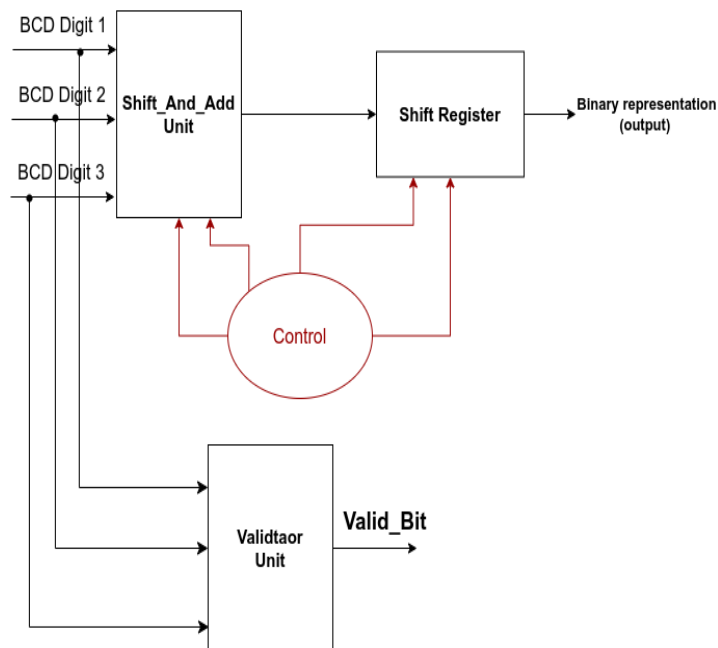
در نهایت بلوک دیاگرام نهایی برای شبیه سازی مدار و قطعات استفاده شده به صورت زیر می باشد:

### ۱.۱ قطعات لازم

- دیکودر ۴ به ۱۶ (قطعه 74154)
- گیت های منطقی *OR*، *AND*، *XOR* ۲ بیتی و گیت منطقی *NOT*
- مالتی پلکسر ۲ به ۱ چهاربیتی (قطعه 74157)
- شیفت رجیستر چهاربیتی (قطعه 74194)
- جمع کننده چهاربیتی (قطعه 7483)

## ۲.۱ بلوک دیاگرام نهایی

بلوک دیاگرام نهایی به صورت زیر است:



شکل ۱: بلاک دیاگرام کلی

## ۳.۱ شرح الگوریتم

الگوریتم تبدیل یک عدد دهدهی  $r$  رقمی به دودویی معادل به صورت زیر است:

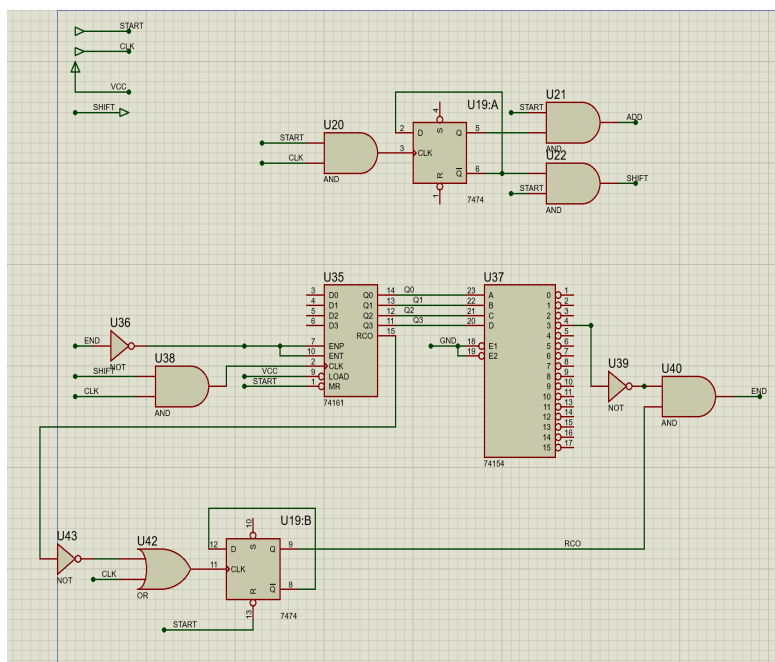
- الف) عدد دهدهی ورودی را یک بیت به راست شیفت دهید.
- ب) اگر با ارزشترین بیت رقم  $i$  ام یک باشد، از آن رقم ۳ تا کم کنید.  $i \leq i < r$
- ج) مراحل الف و ب را تا زمانی که تمام ارقام دهدهی صفر شوند تکرار کنید (حداکثر ۱۰ بار تکرار لازم است).

## ۲ شبیه سازی مدار

به طور کلی برای طراحی این مدار نیاز به چند قطعه برای پیاده سازی هر قسمت داریم که به شرح زیر است:

### ۱.۲ واحد کنترل

ساختار کلی واحد کنترل به صورت زیر است: واحد کنترل از دو بخش تشکیل شده:



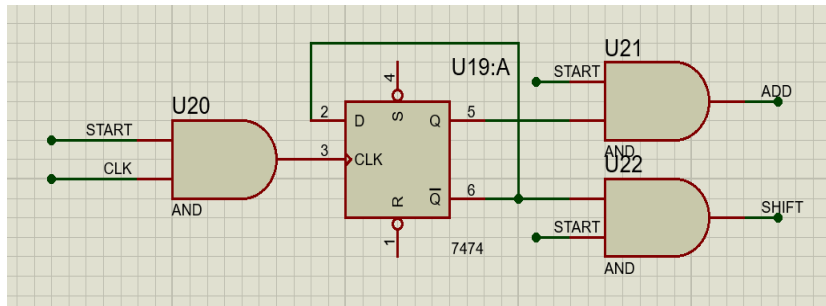
شکل ۲: واحد کنترل

(۱) بخش اول سیگنال های کنترلی *ADD* و *SHIFT* را تولید می کند.

- *ADD*: این سیگنال باعث می شود واحد شیفت دهنده و جمع کننده در صورتی که پرارزش ترین رقم هر رقم دهدهی یک باشد، آن را با عدد ۳ جمع کند. (در واقع عدد ۱۳ را از آن کم می کند)

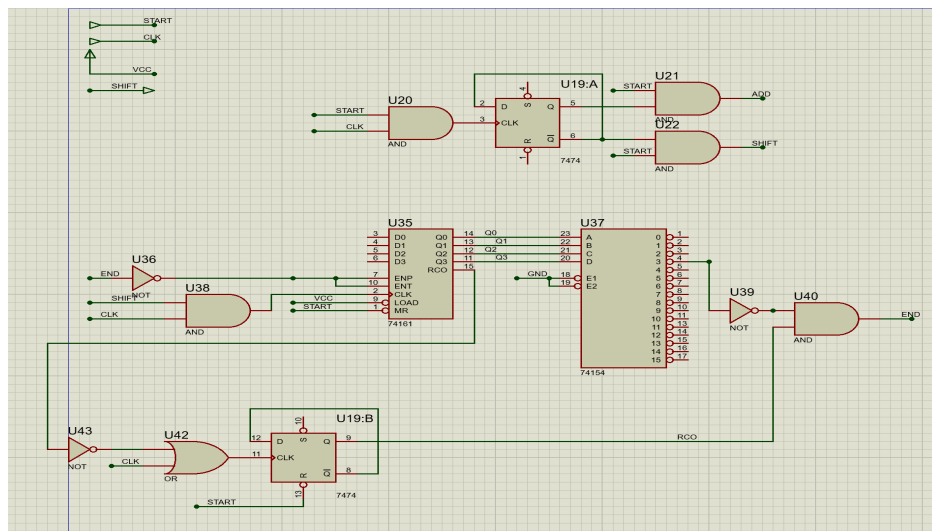
- *SHIFT*: این سیگنال باعث می شود واحد شیفت دهنده و جمع کننده ارقام ورودی را یک رقم به راست شیفت دهد.

این سیگنال ها تنها در صورتی که *START* روشن باشد تولید می شوند.



شکل ۳: سیگنال های *add* و *shift*

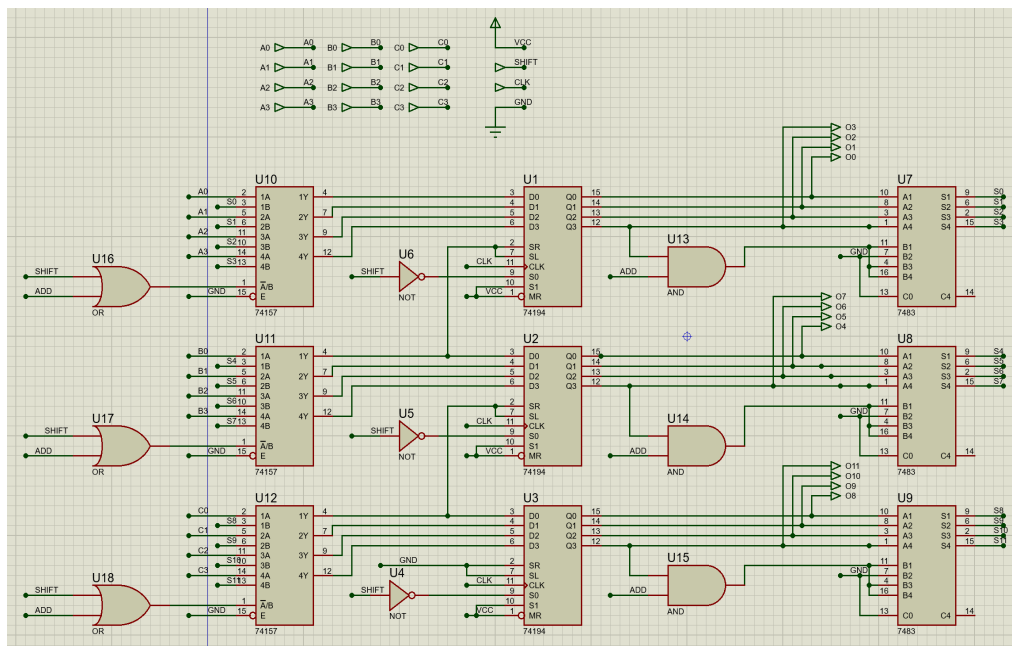
بخش دوم یک شمارنده است که تا ۲۰ می‌شمارد و سپس سیگنال *END* را روشن می‌کند که باعث متوقف شدن مدار می‌شود. با توجه به اینکه شمارنده استفاده شده تنها ۴ رقم دارد، برای شمارش تا عدد ۲۰، یک *DFF* قرار داده شده تا زمانی که *RCO* یک شود، مقدار یک را در خود ذخیره کند. در نهایت زمانی که شمارنده برای بار دوم عدد چهار را می‌شمارد سیگنال *END* روشن می‌شود.



شکل ۴: شمارنده در واحد کنترل

## ۲.۲ واحد $SHIFT - AND - ADD$

این واحد در صورتی که سیگنال  $ADD$  روشن باشد، در صورتی که پرازش ترین رقم هر رقم دهدهی یک باشد، آن را با عدد ۳ جمع کند. (در واقع عدد ۱۳ را از آن کم می کند). در صورتی که سیگنال  $SHIFT$  روشن باشد، ارقام ورودی را یک رقم به راست شیفت می دهد و رقم خارج شده را به عنوان خروجی به مدار اصلی برای ذخیره شدن می دهد.

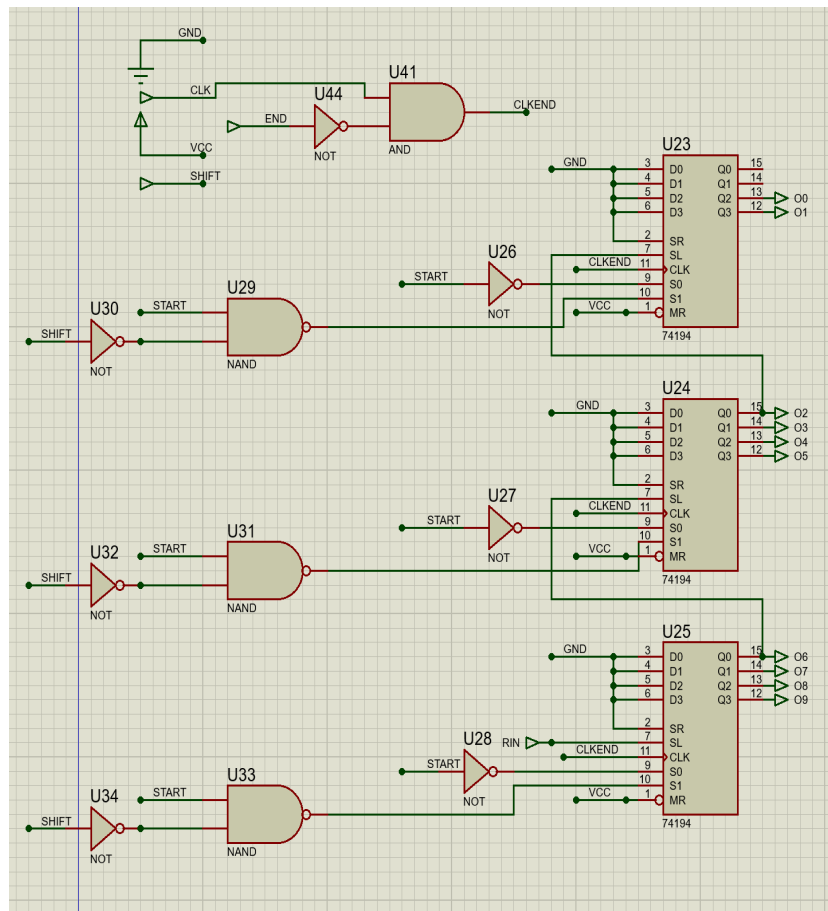


شکل ۵: واحد  $Shift And Add$

### ۳.۲ واحد شیفت رجیستر

این واحد در واقع یک شیفتر رجیسترده بیت‌ی است که با استفاده از چهار شیفتر رجیستر چهار بیت‌ی ساخته شده و تا زمانی که سیگنال  $END$  روشن شود، هر دو پالس ساعت ارقامش را شیفتر می‌دهد.

شکل کلی این واحد به صورت زیر است:

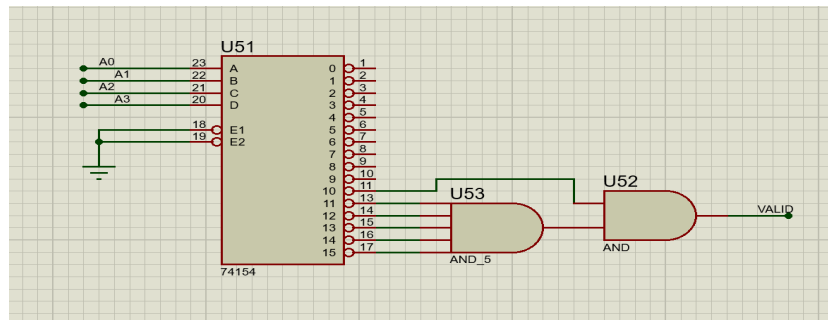


شکل ۶: واحد شیفت رجیستر



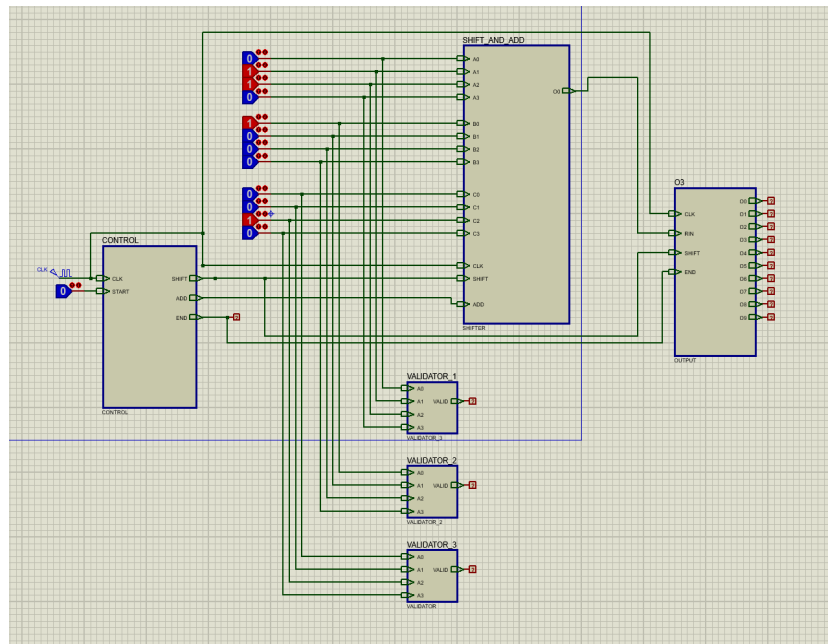
## ۴.۲ واحد validator

این واحد بررسی می کند که ارقام ورودی از ۹ بیشتر نباشند. به صورت جزئی تر در این قسمت از یک دیکورد تشکیل شده است که تنها زمانی بیت *valid* یک است که هیچ کدام از حالات ۱۰ تا ۱۵ رخ ندهد (دقت کنید که مجموعه چهار بیتی که برای *BCD* در نظر گرفته شده حداکثر ۱۵ خواهد بود)



شکل ۷: واحد validator

شکل نهایی مدار در نهایت به صورت زیر خواهد بود:



شکل ۸: شکل نهایی مدار

## ۵.۲ آزمایش مدار و بررسی درستی عملکرد آن

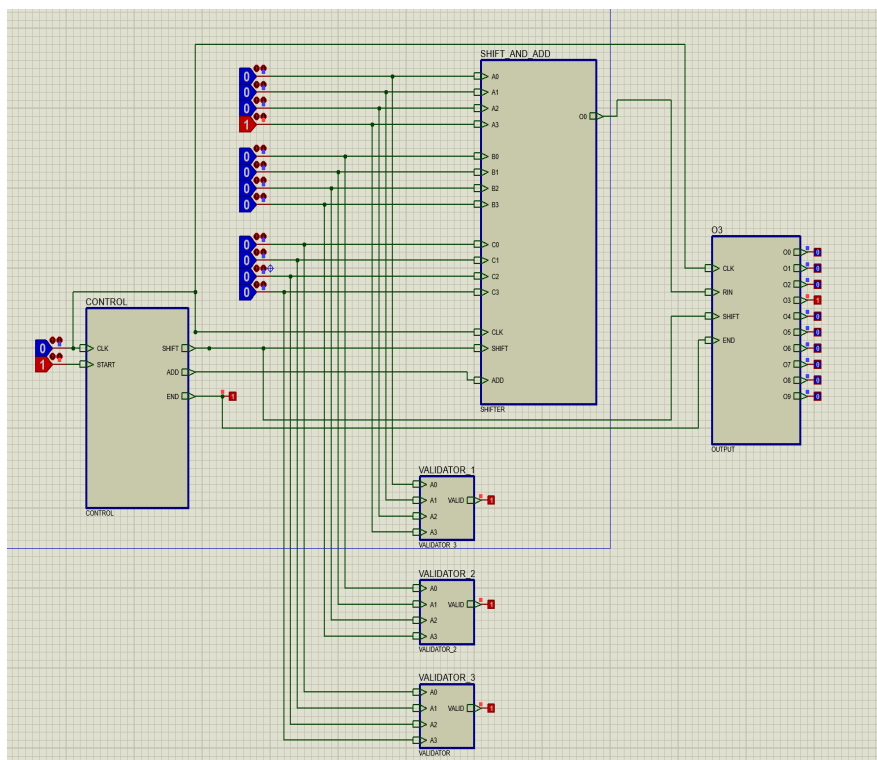
در این قسمت حالت های مختلف ورودی را به مدار دادیم و خروجی آن را بررسی کردیم، نتیجه ۴ تست را نیز در این قسمت آورده ایم:

تست اول) در این قسمت عدد ۸ را تبدیل کرده ایم که همانطور که می بینید حاصل ۱۰۰ شده است.

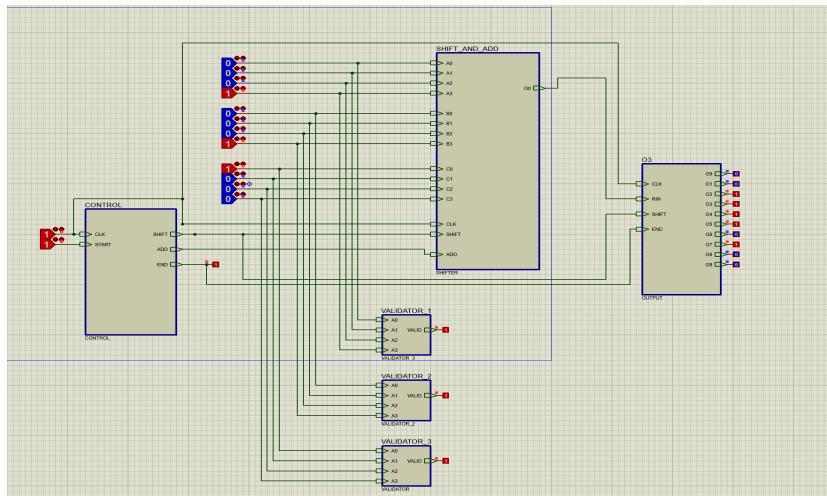
تست دوم) در این قسمت عدد ۱۸۸ را تبدیل کرده ایم که همانطور که می بینید حاصل ۱۰۱۱۱۱۰۰ شده است.

تست سوم) در این قسمت عدد ۴۲۰ را تبدیل کرده ایم که همانطور که می بینید حاصل ۱۱۰۱۰۰۱۰۰ شده است.

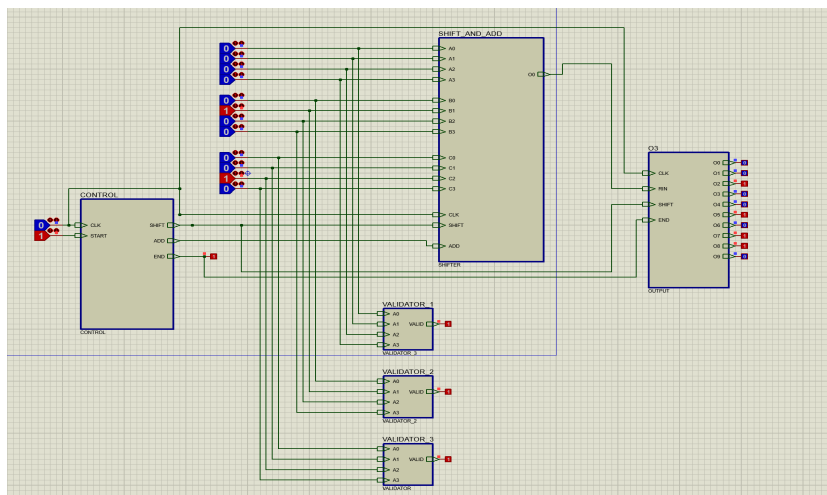
تست چهارم) در این قسمت یکی از ارقام ورودی نامعتبر است که به وسیله validator ها مشخص شده است.



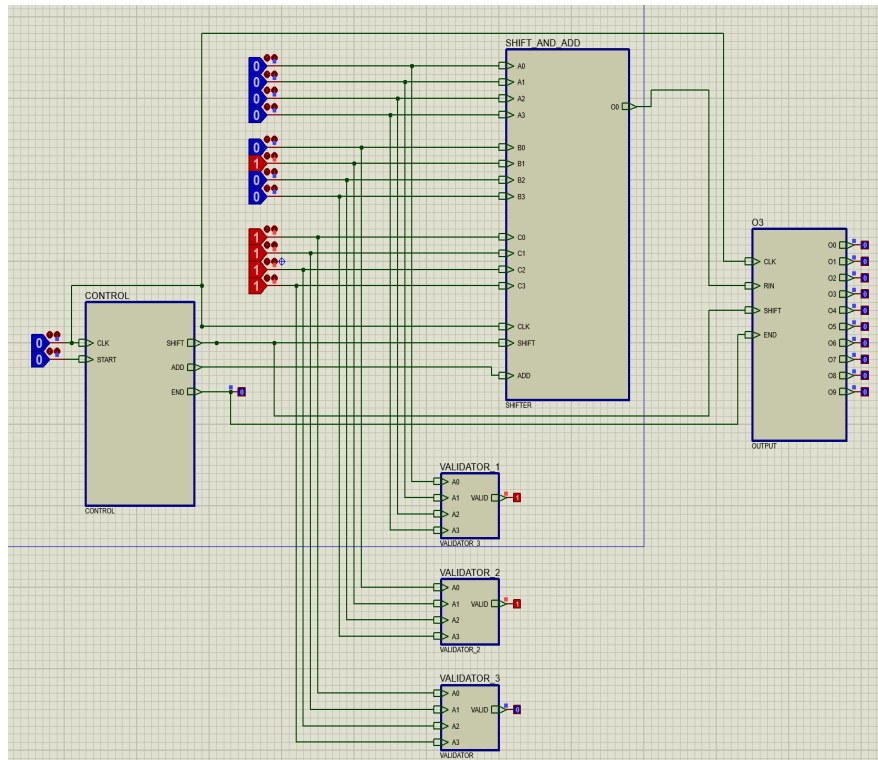
شکل ۹: تست اول



شکل ۱۰: تست دوم



شکل ۱۱: تست سوم



شکل ۱۲: تست چهارم

### ۳ چالش ها

در طراحی این مدار در پروتئوس چالش های کوچکی در هنگام استفاده از شیفت رجیستر ها داشتیم که با بازنگری به دیتاشیت مشکل حل شد.

### ۴ نتیجه و بحث

در این آزمایش قسمت کنترل و مسیر داده از هم جدا شده و با استفاده از سیگنال های قسمت کنترل توانستیم ارقام دهدهی ورودی را به ارقام باینری تبدیل کنیم. در کل استفاده از ساختاری مشخص و واحد کنترلی باعث بهبود سرعت و وقت پیاده سازی شد.