

# گزارشکار آزمایش نهم

پیاده‌سازی استک با مدارات منطقی  
دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر



استاد راهنما: استاد انصاری

روژین تقی‌زادگان ۴۰۱۱۰۵۷۷۵

رادین شاه‌دائی ۴۰۱۱۰۶۰۹۶

باربد شهرآبادی ۴۰۱۱۰۶۱۲۵

## هدف آزمایش:

هدف این آزمایش پیاده‌سازی استک یا همان پشته به کمک دانشی که از مدارمنطقی داریم بوده است. یعنی هم باید از شمارنده‌ها برای نگهداری استک پوینتر و هم از رجسیتورها (یا فلیپ فلاپ‌ها) برای نگهداری مقادیر ذخیره شده در استک استفاده می‌کردیم.

## وسایل مورد نیاز:

در این آزمایش تنها از شبیه‌ساز پروتئوس برای ساختن این مدار به علت پیچیدگی و زیاد بودن قطعات و سیم‌ها استفاده کردیم. در این شبیه‌ساز از گیت‌ها و آیسی‌های متعددی استفاده کردیم:

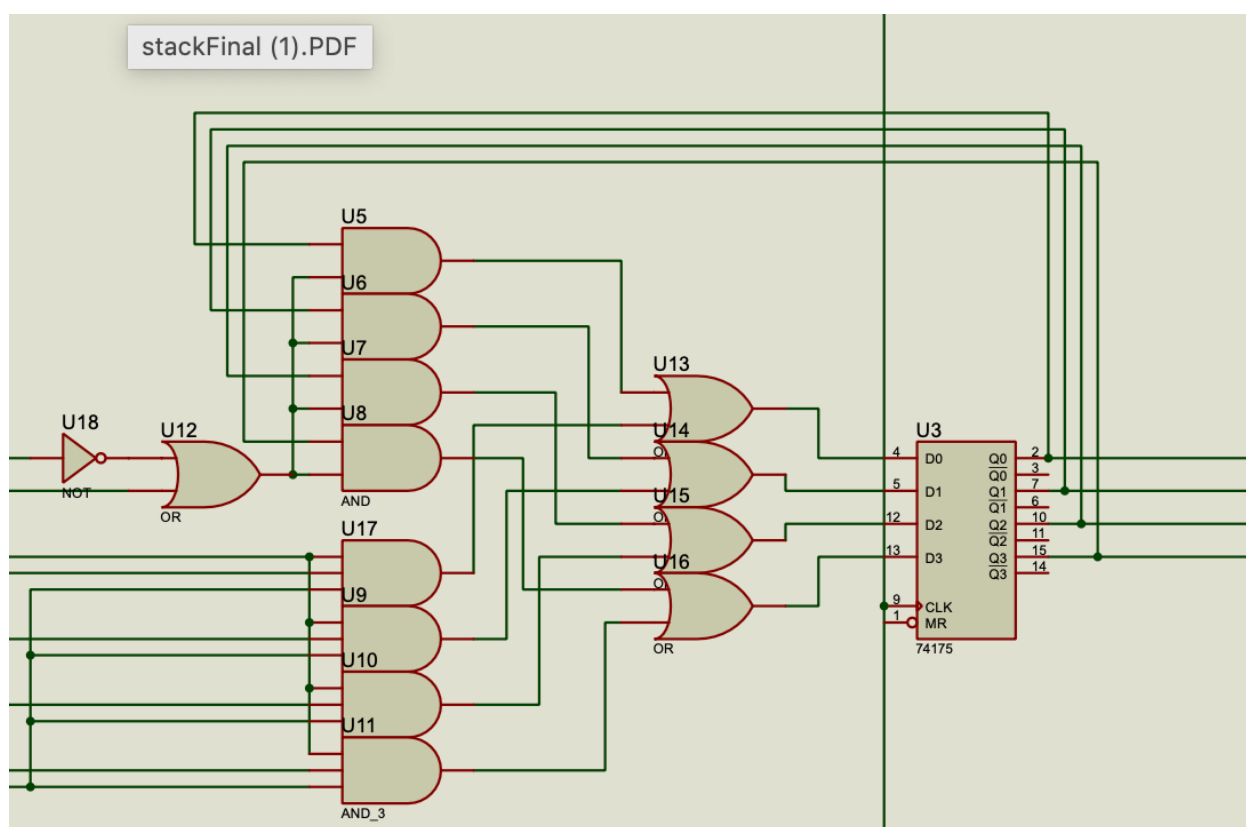
1. گیت AND
2. گیت OR
3. گیت NOT
4. شمارنده‌ی ۴ بیتی (آیسی ۴۰۱۹۲)
5. فلیپ فلاپ ۴ بیتی (آیسی ۷۴۱۷۵)
6. دیکودر ۲ بیتی (آیسی ۴۵۵۵)

## شرح آزمایش:

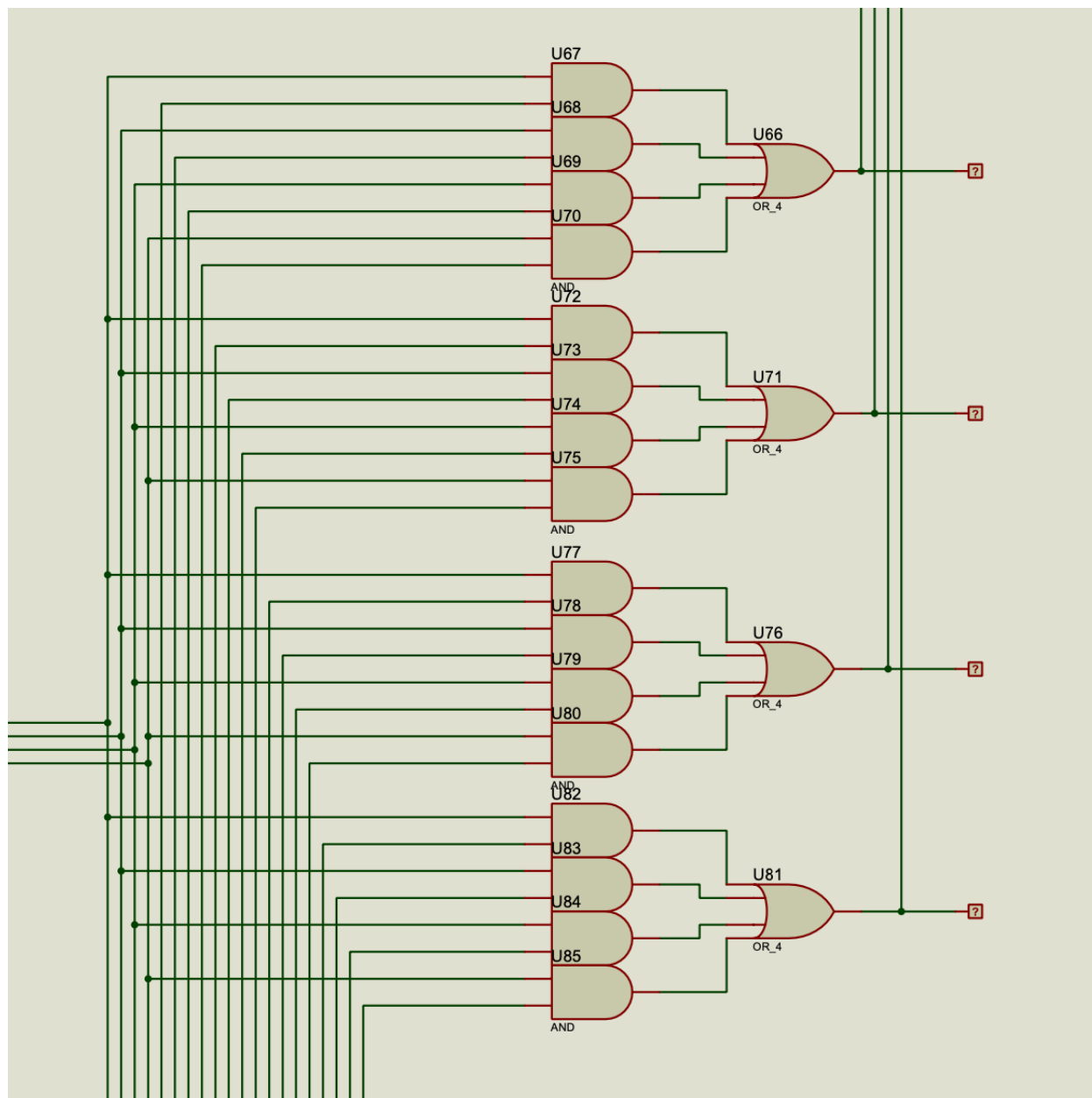
در این آزمایش ابتدا از یک شمارنده‌ی ۴ بیتی برای نگهداری استک پوینتر، که به سر خالی استک اشاره میکند استفاده شده است. در این استک از ۴ خانه‌ی حافظه استفاده کردیم در نتیجه از ۲ بیت اول خروجی شمارنده استفاده کردیم. این شمارنده قابلیت خوبی داشت که از آن استفاده کردیم، این شمارنده دارای دو ورودی برای کلاک، یعنی CLOCKUP و CLOCKDOWN داشت که آنها را به ترتیب به دکمه‌های push و pop وصل کردیم. یعنی اگر ما دکمه‌ی pop را بزنیم مقدار شمارنده ی‌دونه کم می‌شود و اگر دکمه‌ی push را فشار دهیم مقدار شمارنده‌ی دونه افزایش می‌یابد. نکته‌ی حائز اهمیت این است که مقدار شمارنده‌ی ما باید بین ۲ عدد 0 و 3 باقی بماند. این نکته را جلوتر توضیح داده‌ام. ۲ خروجی اول شمارنده را به ورودی‌های دیکودر ۲ بیتی داده‌ایم که در واقع ADDRESS BUS درست کنیم که هرکدام از سیم‌های خروجی این دیکودر به یکی از فلیپ‌فلاپ‌ها که خانه‌های حافظه‌ی ما هستند اشاره می‌کند. نکته‌ای که قبل تر بدان اشاره کردم اینجا نشان داده می‌شود. بدین صورت که خروجی دکمه‌ی push را با معکوس خروجی چهارم دیکودر AND می‌کنیم. یعنی اگر شمارنده به حافظه‌ی چهارم اشاره می‌کرد دیگر نمیتوانیم push کنیم و شمارنده بیشتر از 3 نمی‌شود. به همین ترتیب خروجی دکمه‌ی pop را با معکوس خروجی اول دیکودر AND می‌کنیم که مقدار شمارنده کمتر از 0 نشود. حال اینکه دکمه‌ی push چه میکند. ورودی‌هایی که برای داده دادن به فلیپ‌فلاپ‌ها استفاده می‌کنیم با سیگنال دکمه‌ی push، اگر استک پوینتر یا همان خروجی دیکودر به فلیپ‌فلاپ X اشاره کند، مقادیر ذخیره شده در آن تغییر می‌کند و

استک پوینتر یدونه زیاد می‌شود. در هر کلاک یا خروجی‌های خود فلیپ‌فلاپ X در فلیپ‌فلاپ ریخته می‌شود و یا ورودی‌هایی که به برنامه دادیم در فلیپ‌فلاپ ریخته می‌شود. نکته این است که تنها زمانی ورودی‌های ما درون یک فلیپ‌فلاپ ریخته می‌شوند که هم استک پوینتر بدان اشاره کند هم سیگنال دکمه‌ی push فعال باشد. Pop نیز اینگونه عمل میکند که هر بار فشرده می‌شود، یدونه از استک پوینتر کم می‌شود و خانه‌ای از حافظه که بدان اشاره شده در خروجی برنامه قرار می‌گیرد. هرکدام از اینکه کدام در کدام خانه ریخته می‌شود، از کدام خانه خوانده می‌شود و ... با استفاده از گیت‌های AND و OR صورت می‌گیرد. فایل پروتئوس آزمایش و pdf خروجی پروتئوس در پیوست قرار گرفته است. همچنین عکس‌هایی از مدار طراحی شده اینجا قرار داده‌ام.

- نحوه‌ی تعیین ورودی هر خانه‌ی حافظه:



- تعیین خروجی برنامه:



## - ورودی و خروجی گرفتن برنامه:

