

# آز سیستم های دیجیتال

دکتر اجلائی

مبینا حیدری، عاطفه قندهاری، نیکا قادری  
بهار ۱۴۰۳



آزمایش دوم

## طراحی مدارهای ترتیبی با امکانات شماتیک

تاریخ گزارش: ۲۰ اسفند ۱۴۰۲

### ۱ شرح آزمایش

در این آزمایش قصد داریم یک مدار ترتیبی طراحی کنیم. خواسته مساله به صورت زیر است:  
یک اتاق انتظار با ظرفیت ۱۵ نفر داریم. این اتاق دو در دارد که تحت شرایطی و با توجه به ورودی هایی که از حسگر های محیطی به دست می آیند، دو حالت باز و بسته دارند؛ که باید منطق باز و بسته بودن آن ها طراحی شود.

- ورودی T بیانگر ساعت مجاز ورود است. در صورتی که یک باشد ورود در آن زمان مجاز است و در صورتی که صفر باشد، یعنی ورود غیر مجاز است.
- سیگنال in بیان می کند که فردی وارد اتاق شده است.
- سیگنال out مشخص می کند که فردی از اتاق خارج شده است.
- فردی که می خواهد وارد اتاق شود باید ابتدا دکمه ent را فشار دهد. سپس در صورت مهیا بودن شرایط، به او اجازه ورود داده می شود.
- برای باز کردن در، باید سیگنال open را به مدت لازم روشن نگه داشت.
- در خروجی (در دوم) همواره باز است، مگر این که تعداد افراد حاضر در اتاق به صفر برسد. در این صورت سیگنال close باید یک شود، که نشانگر بسته بودن در می باشد.

### ۲ مازول ها

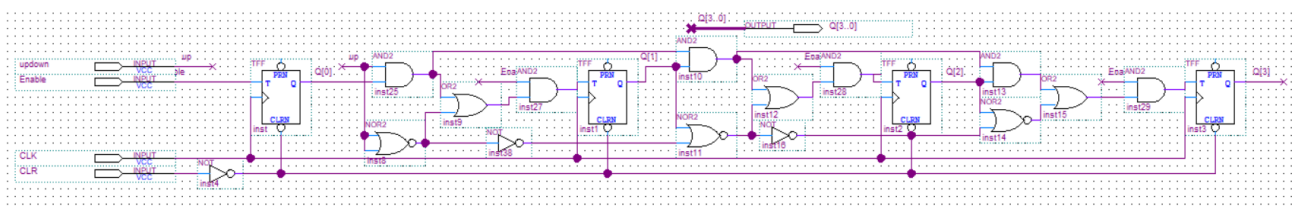
#### ۱.۲ counter

U	Clk	Clr	Enable	Function
X	X	0	X	Reset counter to 0
X	X	1	0	Hold previous number
1	↑	1	1	Up count
0	↑	1	1	Down count

شکل ۱: شمارنده

counter یک شمارنده است که نشان دهنده تعداد افراد حاضر در اتاق می باشد. ورودی U مشخص می کند که شمارنده رو بالا بشمارد و یا رو به پایین. اگر این ورودی یک باشد، بدین معنی است که باید به تعداد افراد اضافه کنیم و اگر صفر باشد یعنی از تعداد افراد کم شده است و شمارنده باید رو به پایین بشمارد.

شمارده به لبه حساس ساعت وابسته است و با یک شدن enable نیز شروع به کار می کند. برای ساخت این شمارنده، از چهار تی فلیپ فلاپ سری استفاده می کنیم که خروجی هر کدام، به ورودی بعدی متصل شده است. در سمت راست



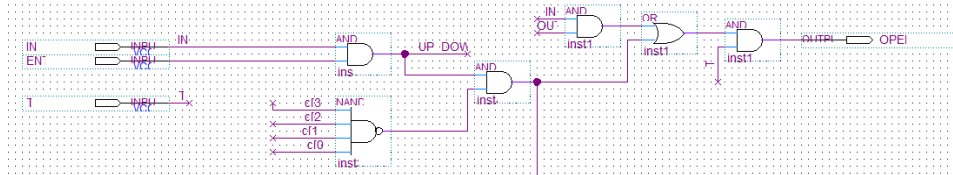
شکل ۲: شمارنده

پرازش ترین بیت، و در سمت چپ کم ارزش ترین بیت قرار دارد. طرز کار این شمارنده به این صورت است که یک بیت هنگامی سوییچ می شود، که یا در حالت افزایشی بیت های کم ارزش تر از آن همه یک باشند، و یا در حالت کاهشی تمام بیت های کم ارزش تر از آن صفر باشند. با توجه به این که اگر ورودی تی فلیپ فلاپ یک باشد مقدار موجود تاگل می شود، کافی است با استفاده از گیت های پایه حالتی که هر بیت فعال می شود را به ورودی فلیپ فلاپ متناظرش متصل کنیم.

توضیحاتی در مورد سیگنال های ورودی این شمارنده:

#### ● enable

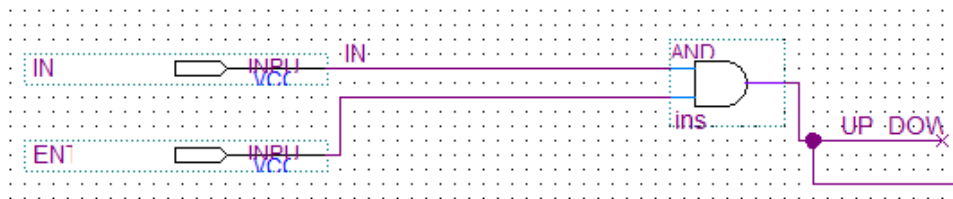
اگر کسی بخواهد خارج شود (یعنی out فعال شود) و تعداد افراد بیشتر از صفر باشد، هم چنین اگر کسی بخواهد وارد شود و در برایش باز شود و تعداد افراد کمتر از ۱۵ باشد و در زمان مجازی نیز باشیم، این سیگنال باید فعال شود. در نتیجه سیگنال کنترلی enable به صورت زیر ساخته می شود:



شکل ۳: ورودی enable

#### ● Up/Down

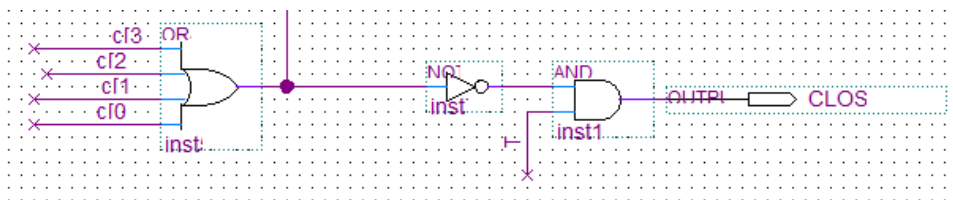
واضح است که این سیگنال تنها وقتی یک می باشد که فردی درخواست ورود بدهد و درخواستش هم قبول شود (یعنی in و ent هر دو یک باشند). در غیر این صورت هم باید صفر باشد. البته توجه داشته باشید که این باعث نمی شود شمارنده به صورت اضافی رو به پایین بشمارد چراکه در زمان هایی که نیاز نیست ورودی enable صفر می شود و از کاهش اضافی تعداد جلوگیری می کند.



شکل ۴: ورودی Up/Down

## ۲.۲ close

در خروجی فقط وقتی باید بسته باشد که تعداد افراد صفر باشد و در زمان مجازی هم باشیم. چون در این حالت عدد نمایش دهنده در شمارنده صفر است، پس با nor کردن آن و گرفتن با T می توان این سیگنال را ساخت.



شکل ۵: سیگنال close

## ۳.۲ open

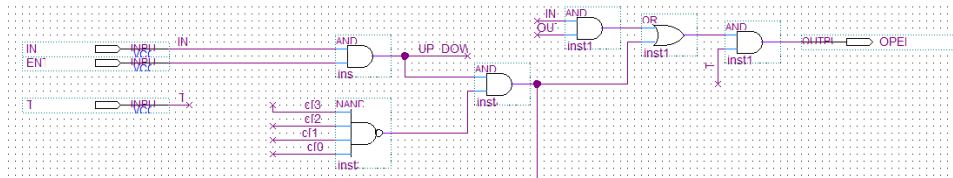
در در این حالات باز می باشد:

۱. در ساعت مجازی، فرد درخواست ورود بدهد، تعداد افراد حاضر در اتاق از ۱۵ کمتر باشد، و in فعال باشد.

۲. دو فرد همزمان در ساعت مجاز وارد و خارج شوند. (یعنی in و out یک باشد)

برای ساده سازی این حالت می توان به این اشاره نمود که حالات مجاز ورود قبلا در سیگنال Up/Down تعیین شده است. بنابراین تنها کافی است آن را با nand بیت های شمارنده and کنیم. این کار به این دلیل انجام می شود که در صورتی که تعداد افراد ۱۵ باشد، nand صفر می شود که باعث صفر شدن کل سیگنال می شود و اجازه نمی دهد در باز شود.

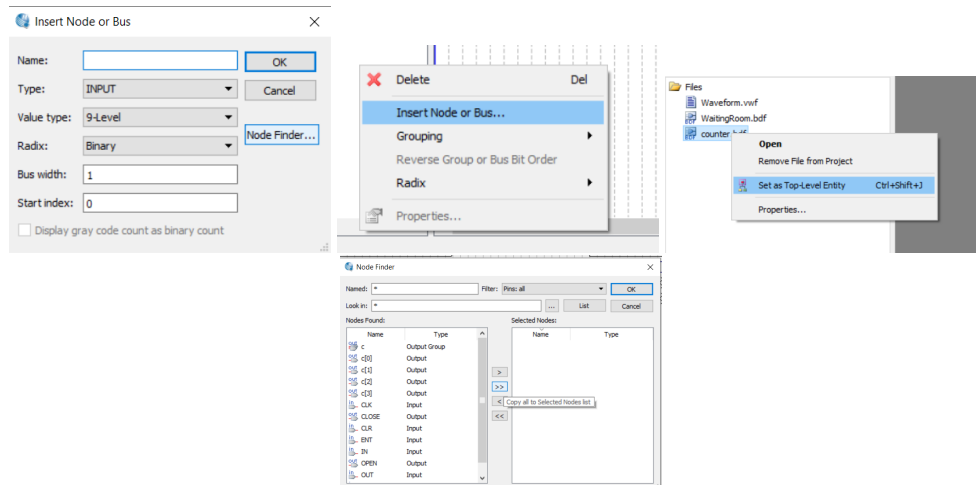
هم چنین چون در هر دو حالت باید زمان مجاز لحاظ شود، می توان سیگنال نهایی را یک بار با T and گرفت تا در زمان و تعداد گیت ها صرفه جویی شود.



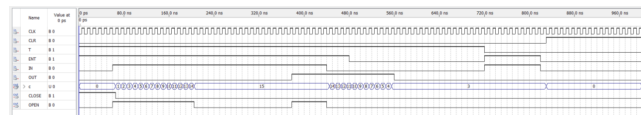
شکل ۶: سیگنال open

## ۳ تست با waveform

جهت تست برنامه، ابتدا آن را کامپایل می‌کنیم. سپس باید برنامه کامپایل شده را به صورت top level entity تعریف می‌کنیم. سپس، یک فایل .vwf می‌سازیم و در مراحل زیر، ورودی‌های لازم را وارد می‌کنیم:



حال می‌توان با ورودی‌های رندوم و یا دستی، عملکرد مدار را شبیه‌سازی، و راستی‌آزمایی کرد.



با کمی دقت می‌توان دریافت که خروجی‌های داده شده همان مقادیر مورد انتظار ما هستند. برای مثال هنگامی که تعداد افراد حاضر در اتاق صفر می‌باشد و در زمان مجازی از روز هم هستیم، در خروجی بسته می‌شود، در حالتی که ظرفیت اتاق کمتر از ۱۵ است و در زمان مجازی هم هستیم، امکان اضافه شدن فرد جدید وجود دارد اما پس از آن این اتفاق نمی‌افتد و تعداد فقط می‌تواند کم شود.

## ۱.۳ فرکانس کاری

ابتدا تعریفی از فرکانس کاری ارائه می‌دهیم. فرکانس کاری به تعداد دفعاتی گفته می‌شود که یک فرآیند یا سیکل در یک واحد زمان تکرار می‌شود. واحد اندازه‌گیری فرکانس کاری، هرتز (Hz) است. برای مثال، اگر یک فرآیند در هر ثانیه یک بار افتاده باشد، فرکانس کاری آن یک هرتز خواهد بود. فرکانس مدار ساخته شده به صورت زیر می‌باشد:

