



**گزارش کار آزمایشگاه طراحی سیستم‌های دیجیتال**

**آزمایش دوم**

**محمدحسین حاجی حسینی**

99101727

**سید علی هزاوه**

99105804

## شرح آزمایش

در این آزمایش می‌خواهیم شبیه سازی برای یک اتاق انتظار طراحی کنیم. در این اتاق، حداکثر 15 نفر می‌توانند حضور داشته باشند. اگر تعداد افراد حاضر بیش از 15 نفر بود، امکان ورود فرد دیگری نیست، تا زمانی که فرد یا افرادی از اتاق خارج شوند، یا به حالت اولیه بازگردیم. (• شدن سیگنال CLR)

### ورودی‌ها:

- IN: اگر 1 شود، بیانگر این است که یک نفر وارد اتاق انتظار شده است و توسط حسگر در شناسایی شده است.
- OUT: اگر 1 شود، بیانگر این است که یک نفر از اتاق انتظار خارج شده است و توسط حسگر در فعال شده است.
- ENT: اگر 1 شود، بیانگر این است که یک نفر قصد وارد شدن به اتاق را دارد.
- T: اگر 1 باشد، یعنی ساعت برای ورود به اتاق مجاز است.
- CLR: اگر 0 شود، به حالت شروع باز می‌گردیم. یعنی هیچکس در اتاق نیست.
- CLK

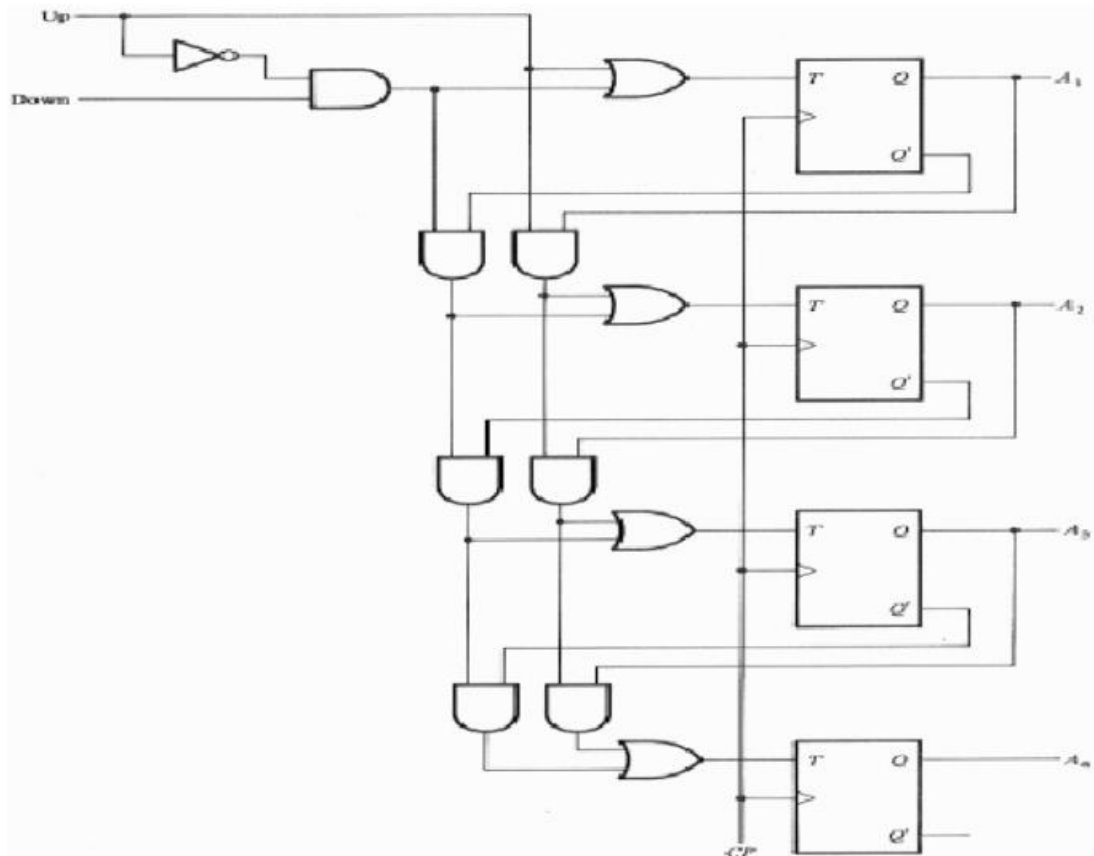
### خروجی‌ها:

- presents: سیگنال چهار بیتی که تعداد نفرات حاضر در اتاق انتظار را بیان میکند.
- CLOSE: اگر تعداد افراد حاضر در سالن انتظار 0 بود، 1 می‌شود و بیانگر بستن در است.
- OPEN: در صورت مساعد بودن شرایط برای ورود یک فرد، به مدت یک Clock برابر با 1 می‌شود. یعنی در برای ورود آن فرد باز است.

### طراحی مدار:

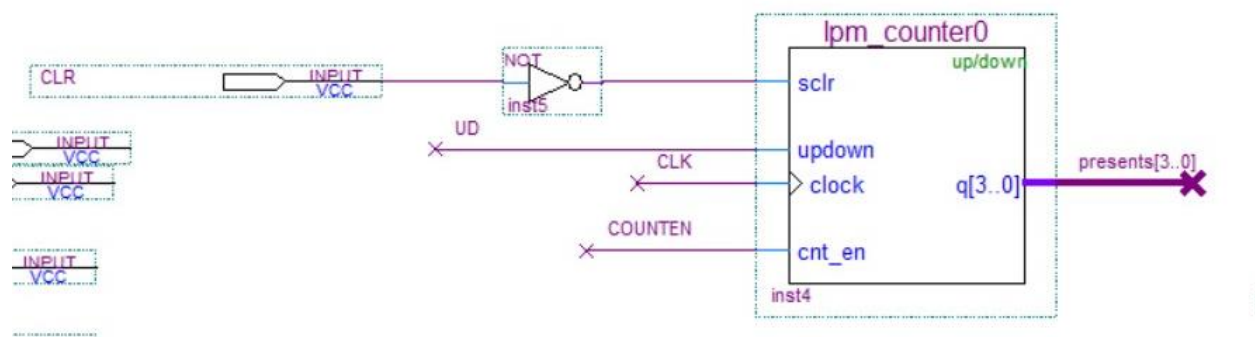
با توجه به صورت مسئله می‌توان برای تولید سیگنال‌های خروجی از روی سیگنال‌های ورودی مداری طراحی کرد؛ می‌دانیم که presents حتما خروجی یک شمارنده ۴ بیتی است با این شرایط که هم کاهشی و هم افزایشی می‌شمارد و با توجه به این که CLR آن را ۰ می‌کند پس یک ورودی reset هم لازم دارد. از طرف دیگر می‌دانیم در حالتی ممکن است در یک کلاک هم‌زمان یک نفر وارد و یک نفر خارج شود. در این صورت تنها راه برای ثابت

نگه داشتن خروجی شمارنده در یک کلاک تنها نشمردن آن ورود و خروج همزمان است. پس با این توصیف باید ورودی داشته باشیم که شبیه enable عمل کند و وقتی شمارنده کار کند که سیگنال مذکور high باشد. با تمام این توصیفات شمارنده‌ی مورد نظر را توسط ابزار کتابخانه خود Quartus ایجاد می‌کنیم و به پروژه اضافه می‌کنیم. منطق کارکرد این شمارنده را از درس مدارهای منطقی و DSD به خاطر داریم و شماتیک آن شبیه به چیزی مثل شکل زیر خواهد بود:



groom.dar

Compilation Report - Flow Summary



برای ورودی‌ها و خروجی‌های شمارنده مطابق شکل مدار زیر خواهیم داشت:

به این صورت که not سیگنال CLR مستقیماً به sclr شمارنده متصل است، سیگنال UP برای مشخص کردن کاهشی یا افزایشی بودن شمارش و سیگنال COUNTEN برای انجام یا عدم انجام شمارش در کلاک‌ها در ادامه تولید خواهند شد:

برای این که بدانیم شمارش افزایشی است یا کاهشی کافی است بدانیم آیا کسی داخل شده یا نه؛ اگر داخل شده بود شمارش افزایشی و در غیر این صورت کاهشی است. در حالتی که کسی به صورت هم‌زمان خارج شود یا این که هیچ کس خارج نشود باید سیگنال COUNTEN را low کنیم تا مشکلی پیش نیاید. اگر تمام حالات را چک کنیم خواهیم دید که COUNTEN در واقع xor دو سیگنال ورودی OUT , AND(IN,T) خواهد بود پس این سیگنال هم تولید شد.

برای خروجی‌های ما یکی سیگنال CLOSE مانده که نشان در صورت High بودن بیانگر خالی شدن اتاق و بسته شدن در می‌باشد. برای این کار باید شمارنده در یک کلاک صفر شود که یعنی ۴ بیت [3..0] presents • شوند پس سیگنال CLOSE در واقع NOR هر چهار بیت سیگنال مذکور است.

سیگنال OPEN در صورت مساعد بود همه شرایط High خواهد بود پس اولاً باید در اتاق جا باشد و در ثانی زمان مجاز باشد و یک نفر بخواهد وارد شود. به این صورت که تا ورود کامل آن نفر یک بماند. پس ما یک فلیپ فلاپ خواهیم داشت که این جا از نوع jkff استفاده می‌کنیم تا بتوان از خاصیت toggle کردن آن هم استفاده کرد. ورودی k فلیپ‌فلاپ ما AND سه سیگنال خواهد بود که شرایط ورود را چک می‌کنند: ENT, T, NAND(presents) و ورودی k ما هم سیگنال IN خواهد بود و خروجی ما هم به output اوپن متصل می‌شود. با این طراحی در صورت وجود شرایط و درخواست برای ورود یک نفر تا ورود کامل او سیگنال open یک است و بعد از ورود او تاگل می‌کند.

در آخر می‌توان با دادن انواع فرم موج مدار را شبیه‌سازی کرد و نتیجه را با انتظار ما مقایسه کرد که انجام شده.

از ما خواسته شده که فرکانس کاری مدار را حساب کنیم؛ برای این کار ابتدا مفهوم فرکانس کاری مدار را یادآوری می‌کنیم. می‌دانیم کلاک ما فرکانس معینی دارد و گیت‌ها هم تاخیر خاص خودشان را دارند و فرکانس کلاک باید طوری تنظیم شود که نتیجه هیچ گیتی جا نماند و به عبارت دیگر بیش‌ترین تاخیر (تقریباً) برابر تناوب کلاک ما خواهد بود. در نرم‌افزار کوارتوس ما پس از کامپایل مدار می‌توانیم از قسمت report فرکانس محاسبه شده را ببینیم:

