



پروژه مدار منطقی

دکتر مهدیانی - زمستان ۱۴۰۰

امیرحسین منصوری

شماره دانشجویی: ۹۹۲۴۳۰۶۹

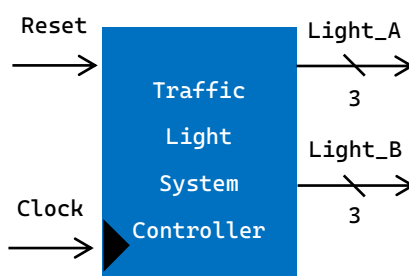
معرفی

این مدار، چراغ‌های راهنمایی یک چهارراه را با زمان‌بندی زیر کنترل می‌کند:

چراغ شرقی-غربی	چراغ شمالی-جنوبی	زمان
سبز	قرمز	۶ ثانیه
زرد	قرمز	۱ ثانیه
قرمز	قرمز	۱ ثانیه
قرمز	سبز	۶ ثانیه
قرمز	زرد	۱ ثانیه
قرمز	قرمز	۱ ثانیه

جدول ۱ - زمان‌بندی چراغ‌ها. در ابتدا با شروع از اولین ردیف جدول، وضعیت چراغ‌ها تغییر می‌کند و پس از رسیدن به آخرین ردیف، دوباره به ردیف اول باز می‌گردد.

طراحی مدار به صورت ترتیبی و سنکرون است. برای طراحی حالت‌ها، از ماشین حالت **Moore** استفاده شده است. همچنین سیگنال **clock** این مدار، با فرکانس **1kHz** کار می‌کند. ورودی‌ها و خروجی‌های مدار به شکل زیر است:



شکل ۱ - ورودی و خروجی مدار

که در آن، **clock** ورودی سیگنال **clock** و همچنین **reset**، برای ریست کردن مدار است. همچنین **Light_A** روشن یا خاموش بودن چراغ‌های شرقی-غربی، و **Light_B** روشن یا خاموش بودن چراغ‌های شمالی-جنوبی را نشان می‌دهد؛ به طوری که کم‌ارزش‌ترین بیت چراغ قرمز، بیت وسط چراغ زرد، و پرارزش‌ترین بیت چراغ سبز را مشخص می‌کند. همه ورودی‌ها و خروجی‌ها **active-high** هستند.

ماژول counter

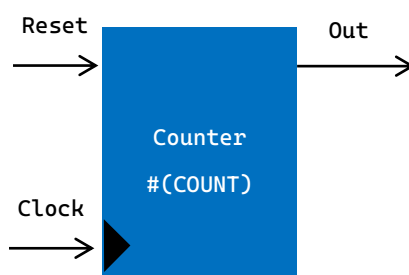
برای پیاده کردن زمان انتظار بین حالت‌های مختلف چراغ، از ماژول **counter** استفاده می‌کنیم. این ماژول، همواره خروجی **0** می‌دهد و تنها به صورت دوره‌ای پس از گذشتن تعداد مشخصی دوره **clock** (که به آن **COUNT** می‌گوییم)، برای یک دوره **clock** خروجی **1** می‌دهد. با تنظیم **COUNT** مناسب، می‌توان این ماژول را طوری تنظیم کرد که در بازه زمانی مشخصی به صورت چشمک‌زن خروجی **1** بدهد. مثلاً اگر فرکانس **clock** برابر **1kHz** باشد و بخواهیم این ماژول هر **۶** ثانیه خروجی **1** بدهد، کفایت **COUNT** آن را برابر **۶۰۰۰** تنظیم کنیم.

برای سازگاری با دیگر قسمت‌های مدار، این ماژول در عمل بعد از گذشت **COUNT-1** دوره **clock**، خروجی **1** می‌دهد. دلیل این امر این است که قسمت‌های دیگر مدار، دقیقاً بعد از **1** شدن خروجی این ماژول (و نه همزمان با آن) عمل می‌کنند. بنابراین این ماژول، یک دوره زودتر خروجی می‌دهد تا قسمت‌های دیگر مدار در زمان درست فعال شوند و کار کنند.

پیاده‌سازی این ماژول به این صورت است که با شروع از مقدار 1، در هر لبه بالارونده clock، این مقدار 1 را یکی یکی زیاد می‌کند تا به تعداد مشخص شده برسد. پس از رسیدن به این مقدار، خروجی را 1 می‌کند و شمارش را دوباره از 1 آغاز می‌کند و این روند را تکرار می‌کند.

مقدار COUNT در وریلاگ، به صورت یک parameter و در هنگام instantiation این ماژول داده می‌شود. به این صورت می‌توان تنها یک بار ماژول را در وریلاگ پیاده‌سازی کرد و در قسمت‌های مختلف مدار، آن را با COUNT های مختلف instantiate کرد تا بتوانیم counter با COUNT های مختلف داشته باشیم. (درواقع COUNT ورودی مدار نیست و در هنگام کامپایل مقدار آن مشخص است).

شکل ورودی و خروجی مدار به صورت زیر است:



شکل ۲- ماژول counter

همچنین با reset کردن این ماژول، شمارش از ابتدا آغاز می‌شود.

قسمت اول - ماشین حالت

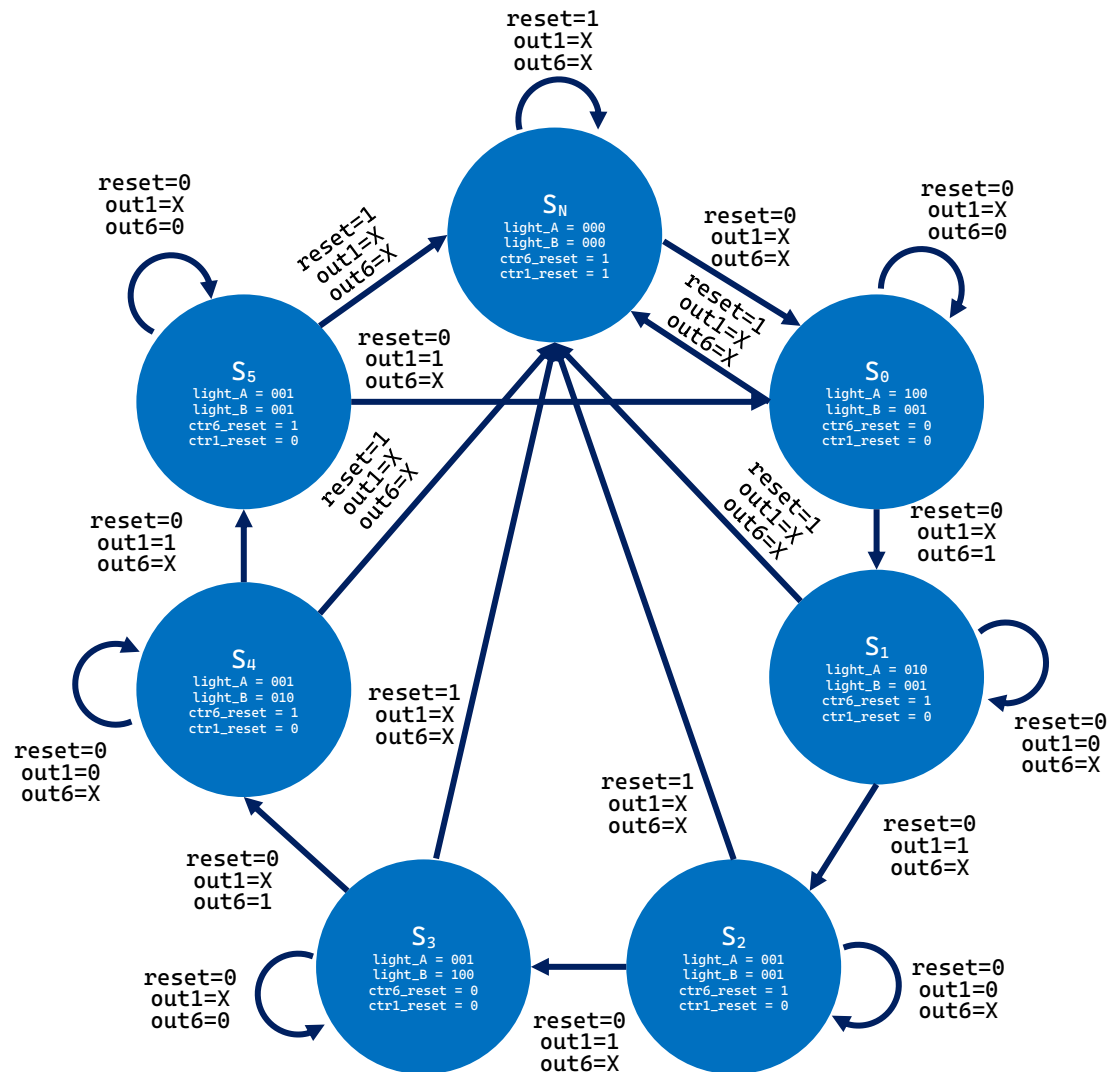
ماشین حالت مدار را به صورت ماشین Moore طراحی می‌کنیم. ۷ حالت زیر را در نظر می‌گیریم:

حالت	چراغ شرقی-غربی	چراغ شمالی-جنوبی	زمان
S_N	-	-	-
S_0	سبز	قرمز	۶ ثانیه
S_1	زرد	قرمز	۱ ثانیه
S_2	قرمز	قرمز	۱ ثانیه
S_3	قرمز	سبز	۶ ثانیه
S_4	قرمز	زرد	۱ ثانیه
S_5	قرمز	قرمز	۱ ثانیه

جدول ۳- توصیف حالت‌های مدار

حالت S_N برای وقتی است که مدار در حالت ریست شده قرار دارد. در واقع هر بار که ورودی reset مدار فعال است، به این وضعیت می‌رویم و در صورتی که ورودی reset دیگر فعال نباشد، به وضعیت S_0 می‌رویم (یعنی از ابتدا شروع می‌کنیم).

ماشین حالت را به صورت زیر رسم می‌کنیم:



شکل ۳- ماشین حالت مدار

در ماشین حالت بالا، `light_A`، `light_B` ورودی و خروجی های مدار هستند. همچنین `ctrl_reset` و `ctr6_reset`، ورودی ها و `out1` و `out6`، خروجی های دو ماژول `counter` هستند که به صورت زیر ساخته شده اند:

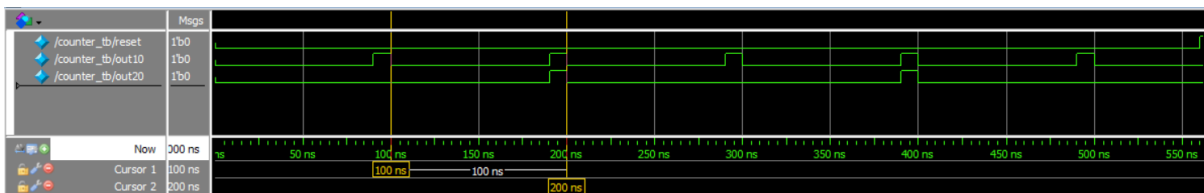
```
counter #(.COUNT(6000)) ctr6_module(out6, clk, ctr6_reset);
counter #(.COUNT(1000)) ctrl_module(out1, clk, ctrl_reset);
```

شکل ۴ - ماژول های `ctr6_module` و `ctrl_module` در کد Verilog

قسمت دوم - پیاده سازی مدار

تست ماژول `counter`

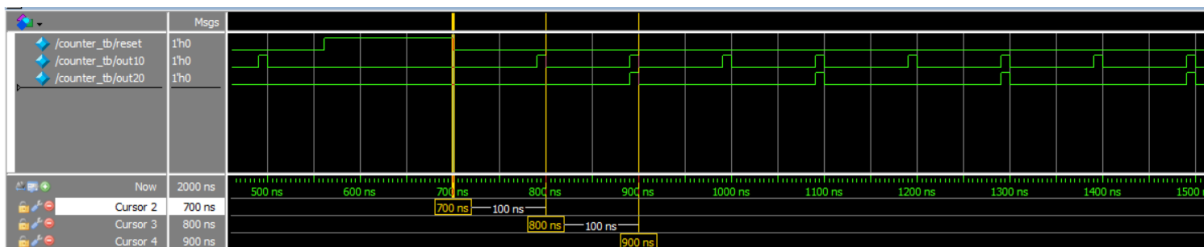
ماژول `counter_tb`، حاوی `TestBench` این ماژول است. در این تست، دو ماژول `counter` با مقادیرهای `COUNT` ۱۰ و ۲۰ ساخته شده اند. همچنین فرکانس `clock` روی ۱۰ns تنظیم شده است. بنابراین انتظار داریم ماژول اول هر ۱۰۰ns و ماژول دوم هر ۲۰۰ns و در دوره `clock` منتهی به این زمان ها، خروجی ۱ بدهد. تا زمان ۵۵۰ns، خروجی تست به صورت زیر است:



شکل ۵ - نمودار counter_tb تا حدود 550ns

در بالا مشخص است که ماژول اول (با خروجی out10)، دقیقاً قبل از 100ns خروجی ۱ داده است و در این زمان نیز خروجی خود را 0 کرده است، و این رفتار را هر 100ns تکرار کرده است، که نشان از کارکرد درست این ماژول تا این زمان دارد. همچنین ماژول دوم (با خروجی out20) به طور مشابه هر 200ns خروجی 1 داده است که عملکرد صحیح ماژول دوم را نیز تایید می‌کند.

در زمان 560ns، ورودی reset برابر 1 و در زمان 700ns برابر 0 می‌شود. خروجی تست به صورت زیر است:



شکل ۶ - نمودار counter_tb تا حدود 1500ns

در بالا مشخص است که 100ns بعد از 0 شدن reset، خروجی out10 و 200ns بعد از 0 شدن reset، خروجی out20 برابر 1 شده‌اند که عملکرد صحیح ماژول را همچنان تایید می‌کنند. همچنین این رفتار، در ادامه نمودار تکرار شده است.

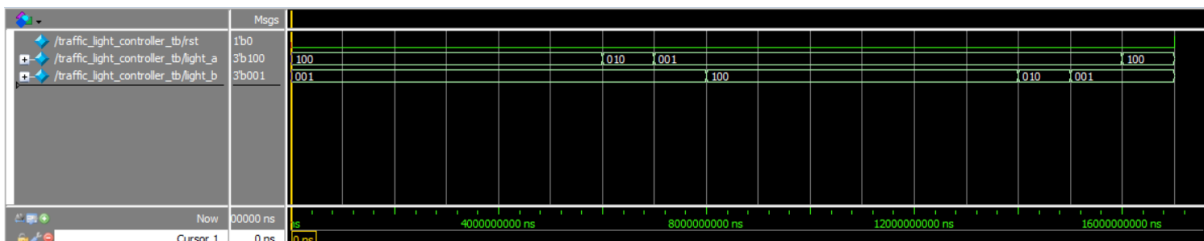
ماژول traffic_light_controller

ماژول traffic_light_controller، حاوی پیاده‌سازی مدار است. در این مدار، از ماژول counter برای پیاده‌سازی فاصله زمانی بین حالت‌ها استفاده شده است که شرح آن در ابتدای گزارش داده شد.

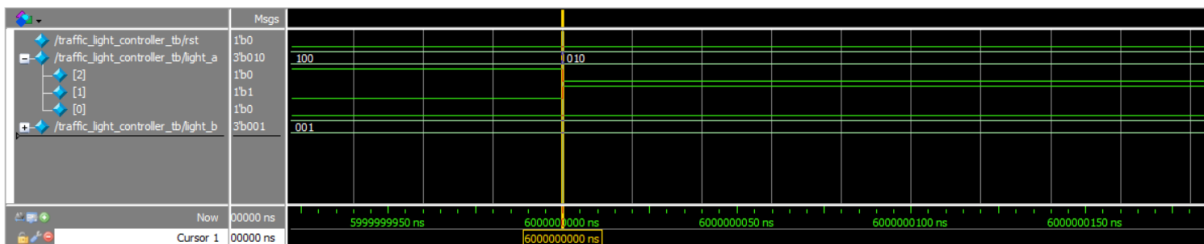
تست ماژول traffic_light_controller

در TestBench این ماژول، برای راحتی کار و بزرگ نشدن اعداد مربوط به delay، با استفاده از دستور timescale مقیاس زمانی را به میلی‌ثانیه تنظیم می‌کنیم. همچنین همانطور که در ابتدا اشاره شد، فرکانس سیگنال clock را روی 1kHz تنظیم می‌کنیم (برای این کار، دوره سیگنال را روی 1ms قرار می‌دهیم).

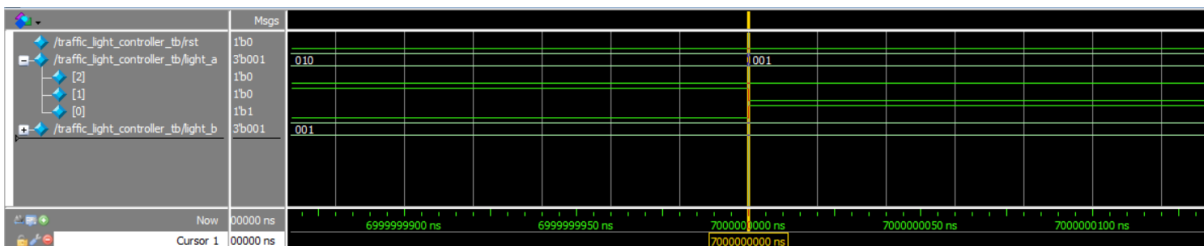
با اجرای تست، نمودار خروجی در ۱۷ ثانیه اول به صورت زیر است:



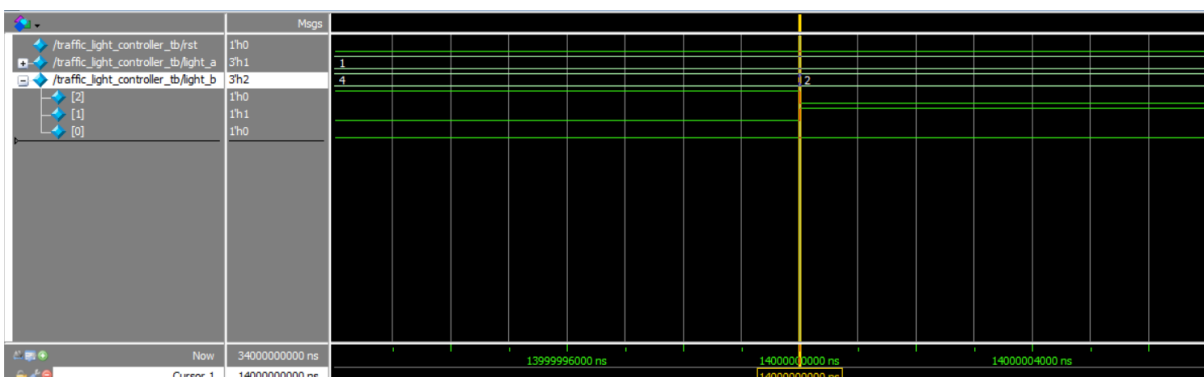
شکل ۷ - نمودار traffic_light_controller_tb تا حدود ۱۷ ثانیه



شکل ۸ - نمودار بزرگ‌نمایی شده **traffic_light_controller_tb** در اطراف ثانیه ۶. خروجی **light_a** دقیقاً در این زمان تغییر کرده است که با شرح حالت‌های مدار منطبق است.

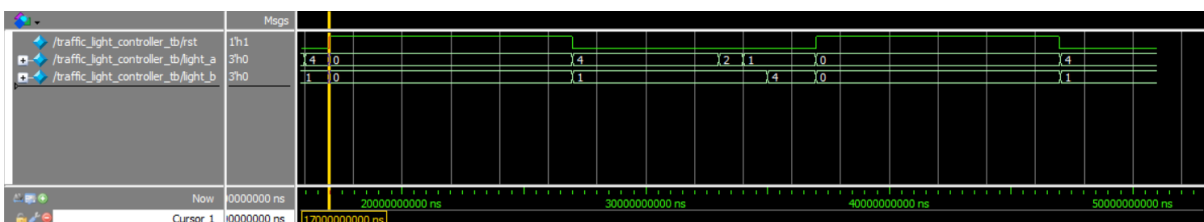


شکل ۹ - نمودار بزرگ‌نمایی شده **traffic_light_controller_tb** در اطراف ثانیه ۷. خروجی **light_a** دقیقاً در این زمان تغییر کرده است که با شرح حالت‌های مدار منطبق است.



شکل ۱۰ - نمودار بزرگ‌نمایی شده **traffic_light_controller_tb** در اطراف ثانیه ۱۴. خروجی **light_a** دقیقاً در این زمان تغییر کرده است که با شرح حالت‌های مدار منطبق است.

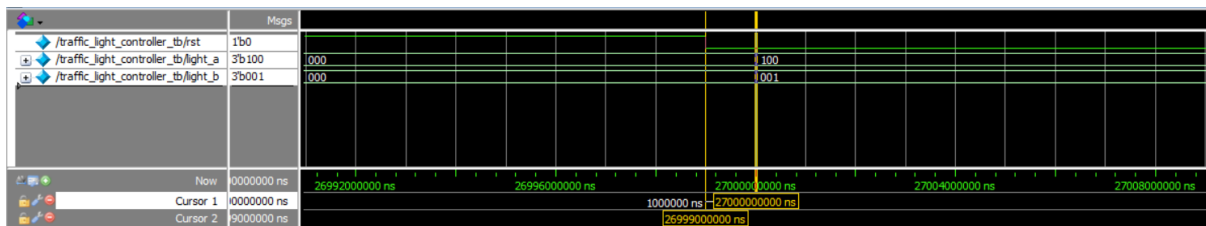
در ثانیه ۱۷، ورودی **reset** برابر ۱ می‌شود:



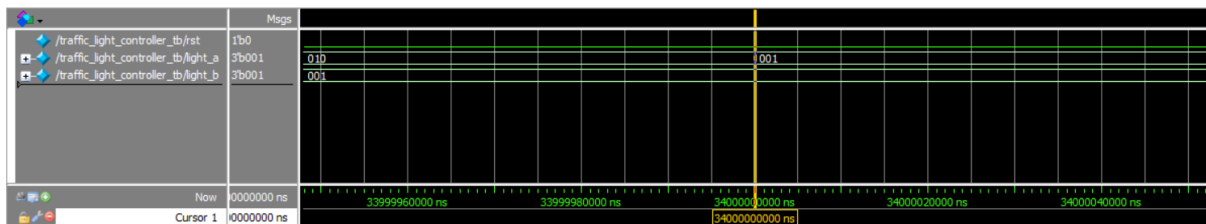
شکل ۱۱ - نمودار **traffic_light_controller** بعد از ثانیه ۱۷

و همانطور که مشخص است، هر دو خروجی مدار برابر ۰ می‌شود که عملکرد صحیح مدار را نشان می‌دهد.

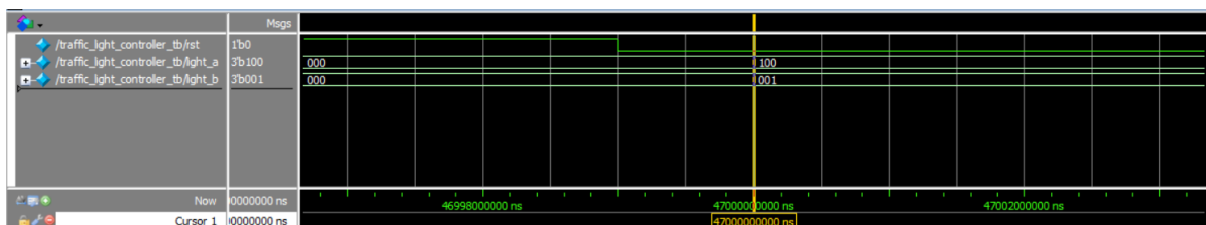
همچنین در زمان 269999ms، ورودی **reset** برابر ۰ می‌شود (به اندازه یک دوره **clock** - یعنی 1ms - طول می‌کشد تا مدار به صفر شدن **reset** واکنش نشان دهد، بنابراین به جای 27s، زمان 269999ms برای صفر شدن **reset** انتخاب شده است).



شکل ۱۲- نمودار `traffic_light_controller_tb` در اطراف ثانیه ۲۷. دقیقاً به اندازه یک دوره `clock` بعد از صفر شدن `reset` مدار به حالت `S0` بازگشته است.



شکل ۱۳ - در ثانیه ۳۴ (یعنی ۷ ثانیه پس از شکل ۱۲) باید به حالت `S2` برویم؛ و نمودار بالا نیز همین عملکرد را در مدار نشان می‌دهد.



شکل ۱۴- نمودار `traffic_light_controller_tb` در ثانیه ۴۷. مدار برای دومین بار `reset` شده است.