**کنترلر پارکینگ (سوال 8)**

یسنا نوشیروانی 401106674

**توضیح کد**

سیگنال‌های ورودی ماژول پارکینگ به شرح زیر است:

1. کلاک
2. reset: با فشردن این دکمه ساعت 8 می‌شود، ظرفیت‌ها به مقدار اولیه بر‌می‌گردند و تعداد ماشین‌های پارک شده صفر می‌شود.
3. car\_entered: هنگام رها شدن این دکمه یعنی یک ماشین درخواست ورود به پارکینگ داده است. اگر جای خالی وجود نداشته باشد سیگنال خروجی ja\_nist یک می‌شود و اجازه ورود به ماشین داده نمی‌شود.
4. is\_uni\_car\_entered: اگر این سیگنال هنگام رها شدن دکمه car\_entered یک باشد یعنی ماشین درخواست ورود دانشگاهی فرستاده و اگر صفر باشد یعنی درخواست از نوع ورود آزاد بوده است.
5. car\_exited: هنگام رها شدن این دکمه یعنی یک ماشین درخواست خروج از پارکینگ داده است. اگر ماشینی در پارکینگ (از نوع مورد نظر) نباشد سیگنال خروجی faulty\_exit یک می‌شود و نشان‌دهنده خروج نامعتبر است.
6. is\_uni\_car\_exited: اگر این سیگنال هنگام رها شدن دکمه car\_exited یک باشد یعنی ماشینی که درخواست خروج فرستاده در جایگاه دانشگاهی پارک کرده بوده و اگر صفر باشد یعنی از نوع آزاد بوده است.

سیگنال‌های خروجی به این صورت است:

1. hour: نشان‌دهنده ساعت کنونی‌ست. (هر ساعت 500 کلاک طول می‌کشد)
2. uni\_parked\_car: تعداد ماشین‌های دانشگاهی پارک‌شده
3. free\_parked\_car: تعداد ماشین‌های آزاد پارک‌شده
4. uni\_vacated\_space: تعداد جاهای دانشگاهی خالی
5. free\_vacated\_space: تعداد جاهای آزاد خالی
6. uni\_is\_vacated\_space: وقتی یک است که جای دانشگاهی خالی وجود داشته باشد، در غیر این صورت صفر است.
7. free\_is\_vacated\_space: وقتی یک است که جای آزاد خالی وجود داشته باشد، در غیر این صورت صفر است.
8. ja\_nist: این سیگنال‌ زمانی یک می‌شود که پارکینگ جای خالی برای قبول درخواست ورود صادر شده را نداشته باشد، در غیر این صورت صفر است. (در صورتی که درخواست از نوع دانشگاهی صادر شود تنها در صورتی پذیرفته است که پارکینگ جای خالی از نوع دانشگاهی داشته باشد و درخواست خود به خود از نوع آزاد در نظر گرفته نمی‌شود)
9. faulty\_exit: این سیگنال زمانی یک می‌شود که ماشینی از نوع موردنظر در پارکینگ نباشد و درخواست خروج نامعتبر صادر شود، در غیر این صورت صفر است.

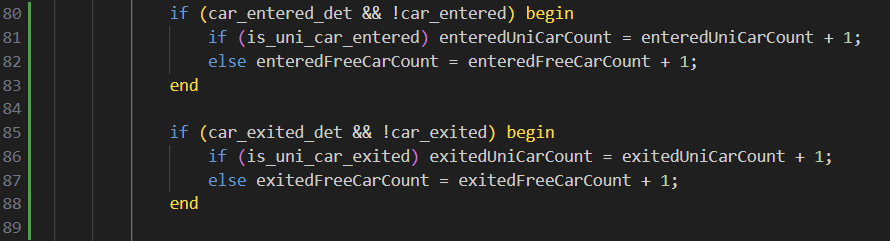
روش عملکرد ماژول:

هنگام رها شدن دکمه car\_entered با توجه به نوع درخواست ورود، اگر آزاد باشد enteredFreeCarCount و اگر دانشگاهی باشد enteredUniCarCount یکی زیاد می‌شود.

هنگام رها شدن دکمه car\_exited با توجه به نوع درخواست خروج، اگر آزاد باشد exitedFreeCarCount و اگر دانشگاهی باشد exitedUniCarCount یکی زیاد می‌شود.

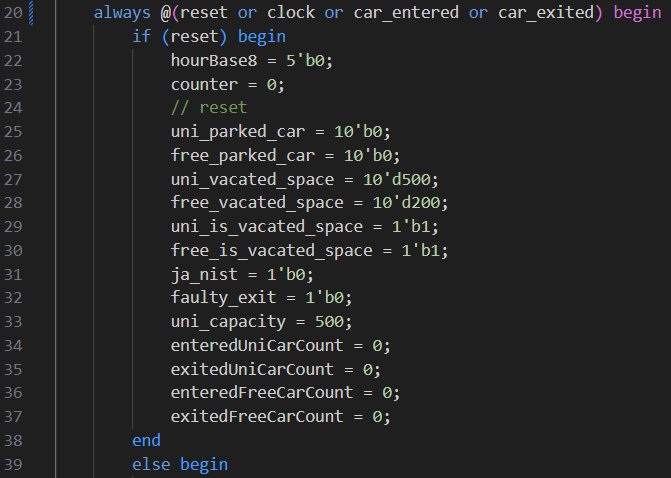
این چهار متغیر نشان‌دهنده تعداد کل ماشین‌های آزاد یا دانشگاهی ورودی و خروجی از آخرین reset قبلی است و اختلاف آن‌ها با تعداد ماشین‌های موجود در پارکینگ برابر است.

تکه کد مربوط به این عملیات:



در بخش اصلی کد ابتدا چک می‌کنیم سیگنال reset یک است یا نه. اگر یک بود مقدار متغیر‌ها و سیگنال‌ها را به حالت اولیه برمی‌گردانیم و اگر صفر بود سراغ ادامه کد می‌رویم.

قسمت reset کد:



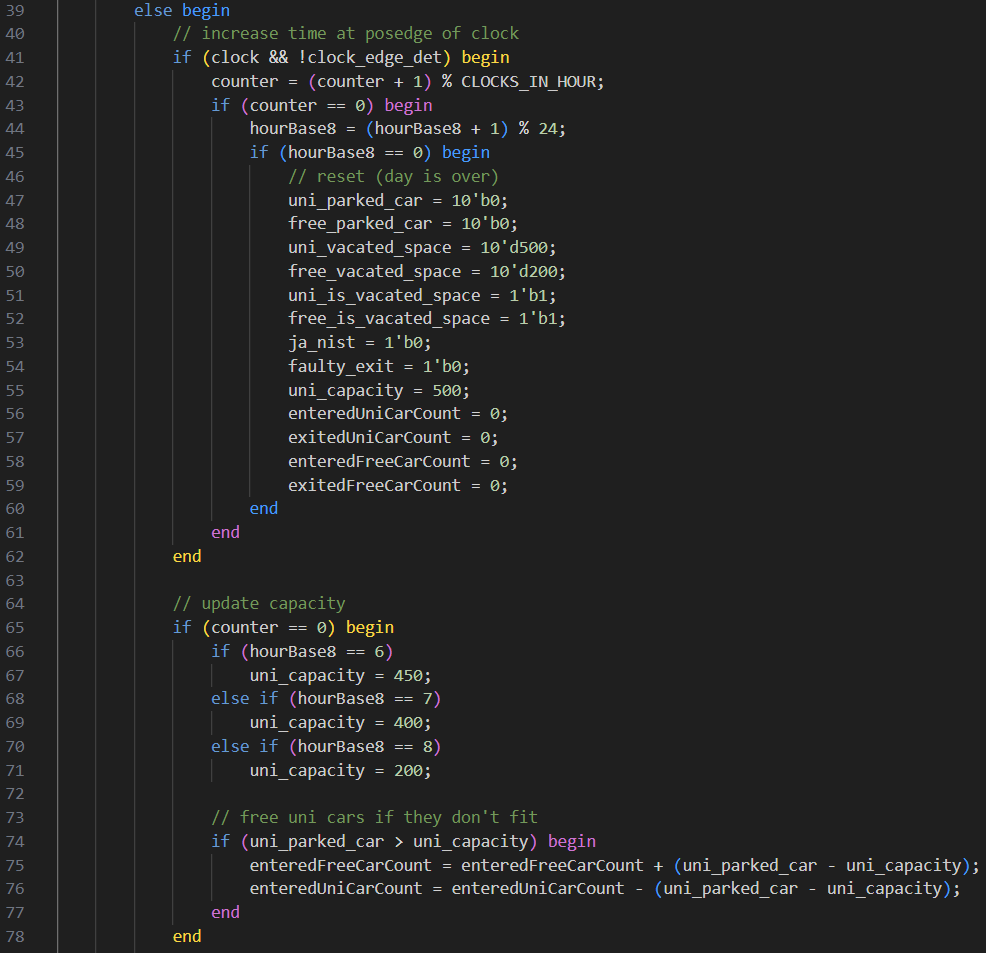
در ادامه اگر به خاطر posedge clock وارد always شده باشیم، ساعت به‌روز‌رسانی می‌شود و اگر به پایان روز رسیده بودیم یعنی هشت صبح روز بعد بود، انگار دکمه reset فشرده شده و دقیقا همان عملیات انجام می‌شوند.

همچنین در این قسمت پس از به‌روز‌رسانی ساعت، ظرفیت در آن ساعت را پیدا می‌کنیم و سیگنال‌ها را مقداردهی می‌کنیم تا در صورتی که اولین بار باشد که always در ساعت جدید اجرا می‌شود ظرفیت‌های پارکینگ به‌روز‌رسانی شوند.

همانطور که در سوال گفته شده بود، از ساعت 13 تا 16 ظرفیت دانشگاهی 50 تا 50 تا کم می‌شود و در ساعت 16 به 200 می‌رسد. (یعنی در انتهای ساعت 14 ظرفیت دانشگاهی 450، در انتهای ساعت 15 برابر با 400 و در انتهای ساعت 16 برابر با 200 است)

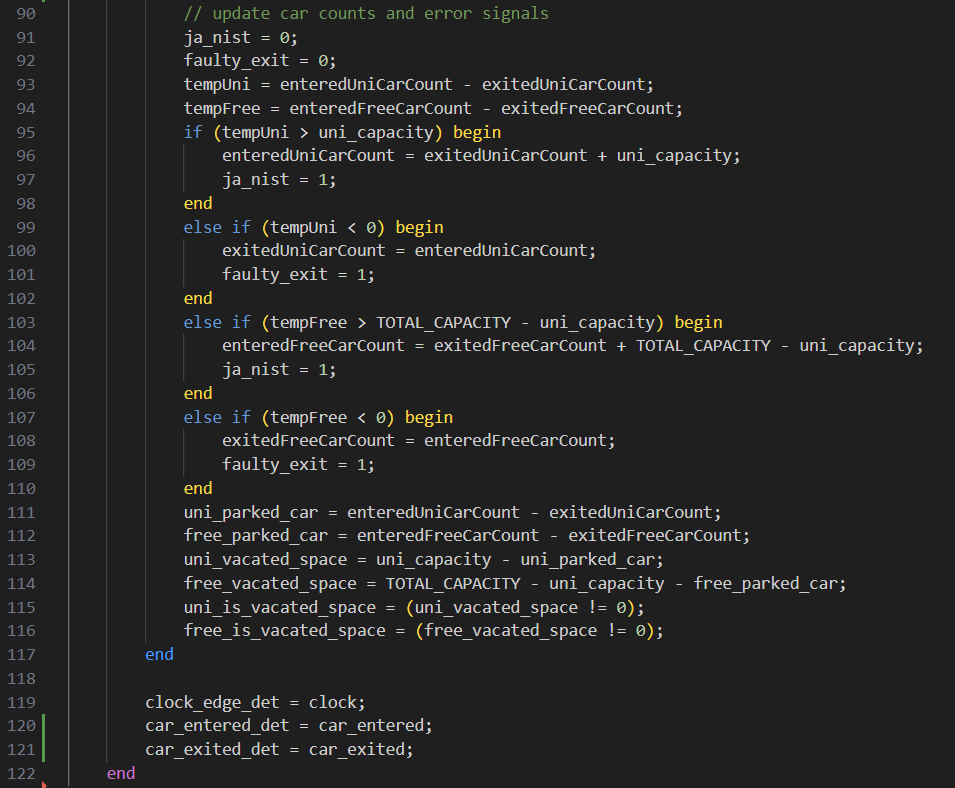
سپس تعداد ماشین‌های پارک‌شده بررسی می‌شود و اگر تعداد ماشین‌های دانشگاهی از ظرفیت جدید دانشگاهی بیشتر بود، آن ماشین‌های اضافه آزاد حساب می‌شوند.

کد مربوط به این قسمت:



در بخش آخر کد چهار متغیری که در ابتدا شرح داده شد (enteredUniCarCount enteredFreeCarCount, exitedUniCarCount, exitedFreeCarCount,) در سیگنال‌های خروجی تاثیر داده می‌شوند. به این‌صورت که ابتدا سیگنال‌های خروجی ja\_nist و faulty\_exit صفر می‌شوند. سپس تعداد ماشین‌های دانشگاهی و آزاد در پارکینگ اگر درخواست داده شده معتبر باشد محاسبه می‌شوند. اگر این مقدار کمتر از صفر یا بیشتر از ظرفیت پارکینگ بود یعنی درخواست نامعتبر بوده و تعداد ماشین‌های داخل پارکینگ (اختلاف دو متغیر برای هر یک از حالات آزاد و دانشگاهی) باید به حالت قبلی بر‌گردد و سیگنال‌های ja\_nist و faulty\_exit با توجه به خطای رخ داده مقداردهی می‌شوند و تا لبه بالارونده بعدی کلاک مقدارشان ثابت می‌ماند. پس از این زمان به حالت اولیه یعنی صفر بر‌می‌گردند.

کد این قسمت:

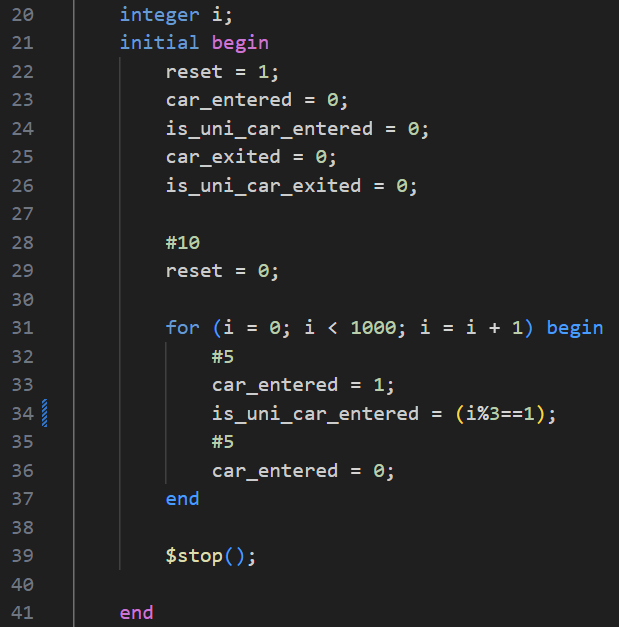


**تست‌های محک و تحلیل نتایج**

**تست محک TB1**

این تست به منظور چک تکمیل ظرفیت و تفکیک درست دانشگاهی و آزاد طراحی شده است.

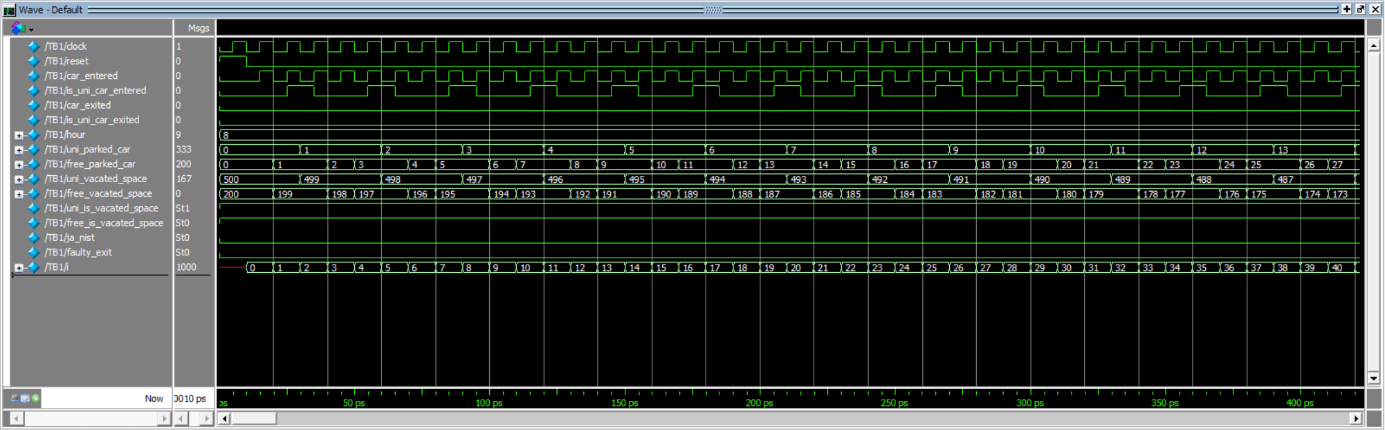
حلقه تولید سیگنال‌های ورودی به این شکل و دوره تناوب کلاک ده نانو‌ثانیه است:



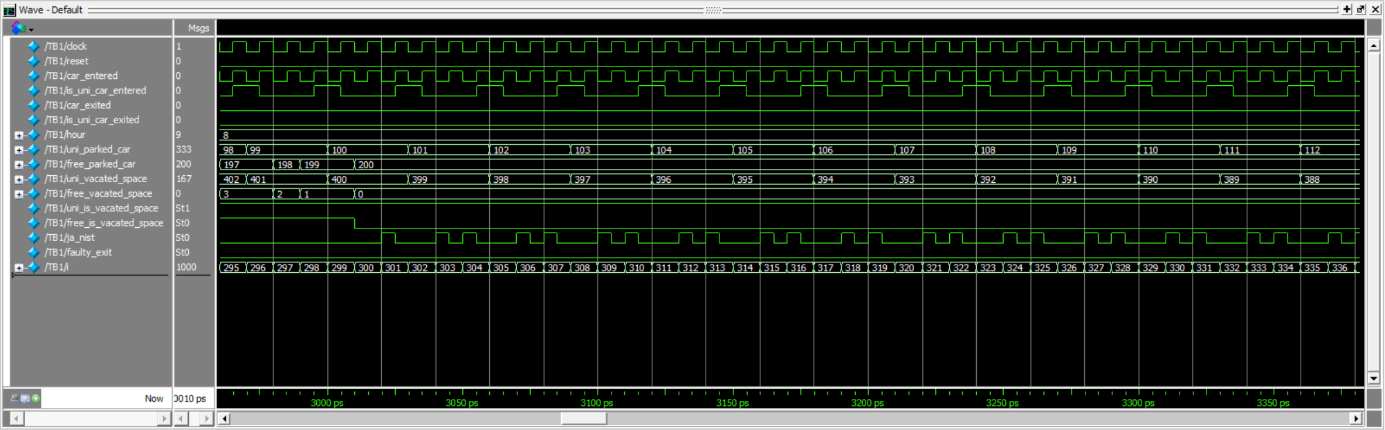
نتیجه محک مطابق انتظار است.

ابتدا مدار reset می‌شود. سپس در هر 10 نانو‌ثانیه یک ماشین وارد پارکینگ می‌شود به طوری که یکی از هر سه ماشین دانشگاهی‌ست. پس از 200 ماشین آزاد ظرفیت آن پر می‌شود و اجازه ورود ماشین‌های بیشتر را نمی‌دهد. به همین علت پس از مدتی سیگنال ja\_nist دو کلاک از هر سه کلاک یک می‌شود. در نهایت ظرفیت پر شده دانشگاهی برابر با 333 ماشین است. همچنین تا پایان شبیه‌سازی ساعت نه است و ظرفیت‌ها تغییری نکرده و برابر با 500 تا دانشگاهی و 200 تا آزاد است.

waveform ها در ابتدای شبیه‌سازی که هر دو قسمت دانشگاهی و آزاد جا دارند به این شکل است:



پس از پر شدن ظرفیت آزاد، waveform به این شکل ادامه می‌یابد:



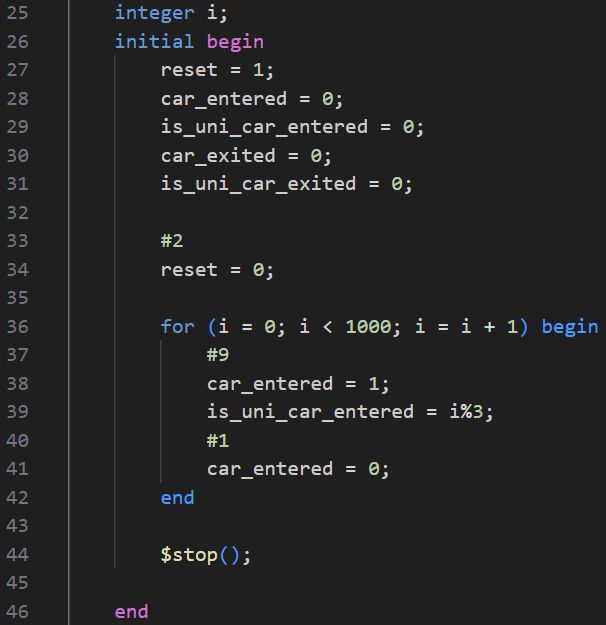
و پس از اتمام حلقه و ارسال سیگنال‌ها مقادیر خروجی به این ترتیبند:



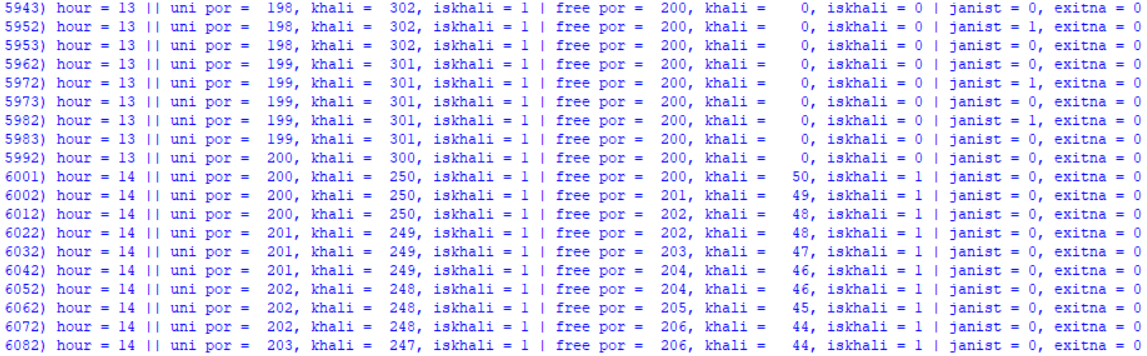
**تست محک TBFreeCapacityIncrease**

این تست برای چک افزایش ظرفیت آزاد پارکینگ پس از ساعت 13 است.

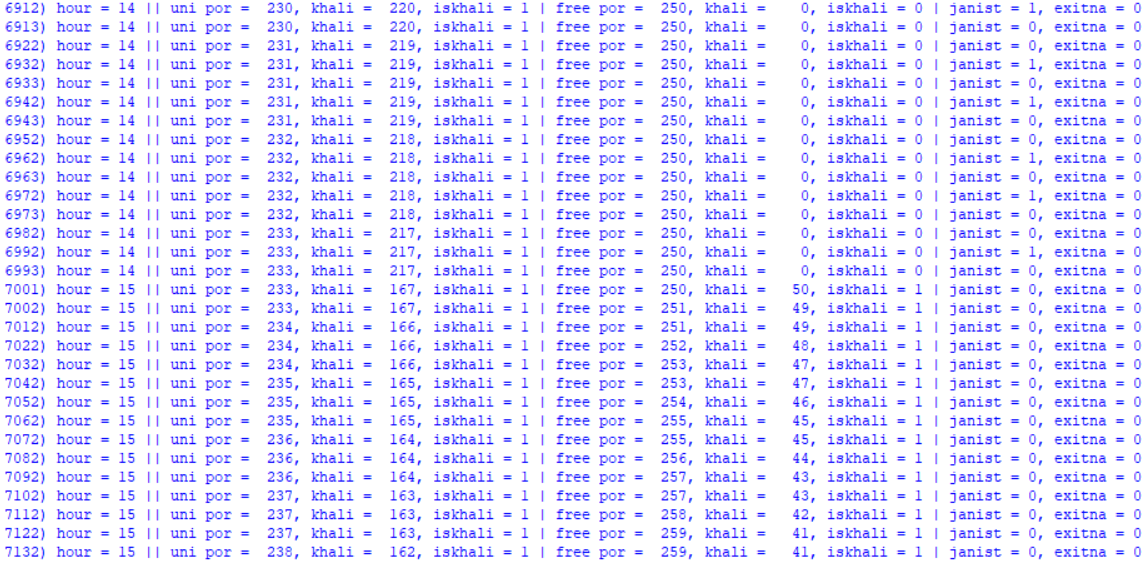
در این تست دوره تناوب کلاک دو نانوثانیه در نظر گرفته شده است و کد قسمت اصلی آن به این شکل است:



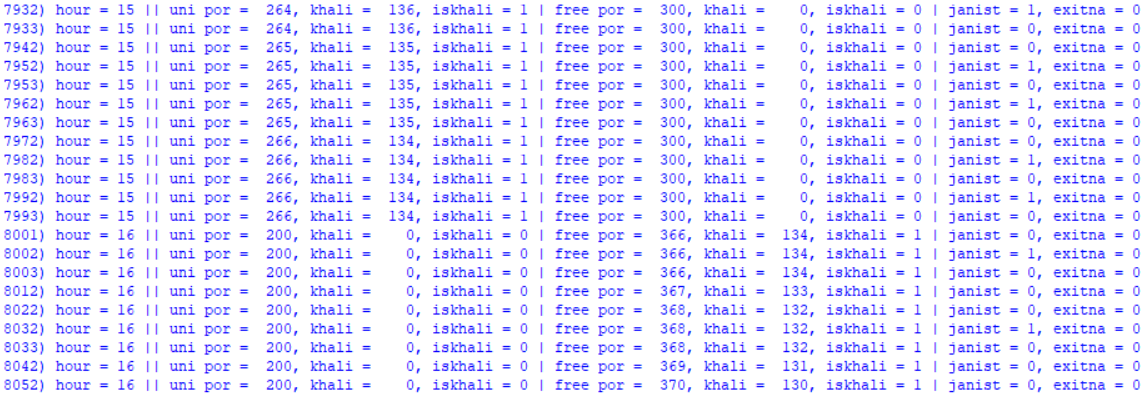
افزایش ظرفیت آزاد را پس از ساعت 13 مشاهده می‌کنیم:



پس از ساعت 14:



پس از ساعت 15:



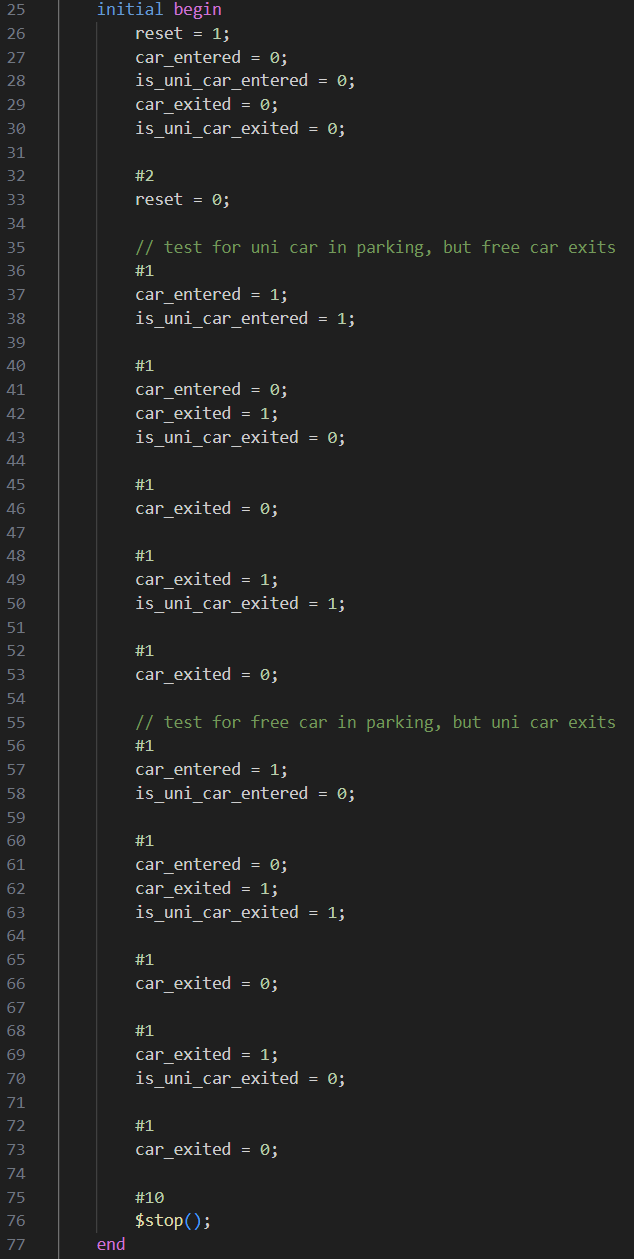
از ساعت سه به چهار ظرفیت آزاد 200 تا اضافه و ظرفیت دانشگاهی 200 تا کم می‌شود. ابتدا جاهای خالی دانشگاهی به ظرفیت آزاد اضافه می‌شوند اما اگر تعدادشان کافی نبود، تعدادی از فضاهای پر دانشگاهی هم به نوع آزاد پر تبدیل می‌شوند.

**تست محک خروج از پارکینگ**

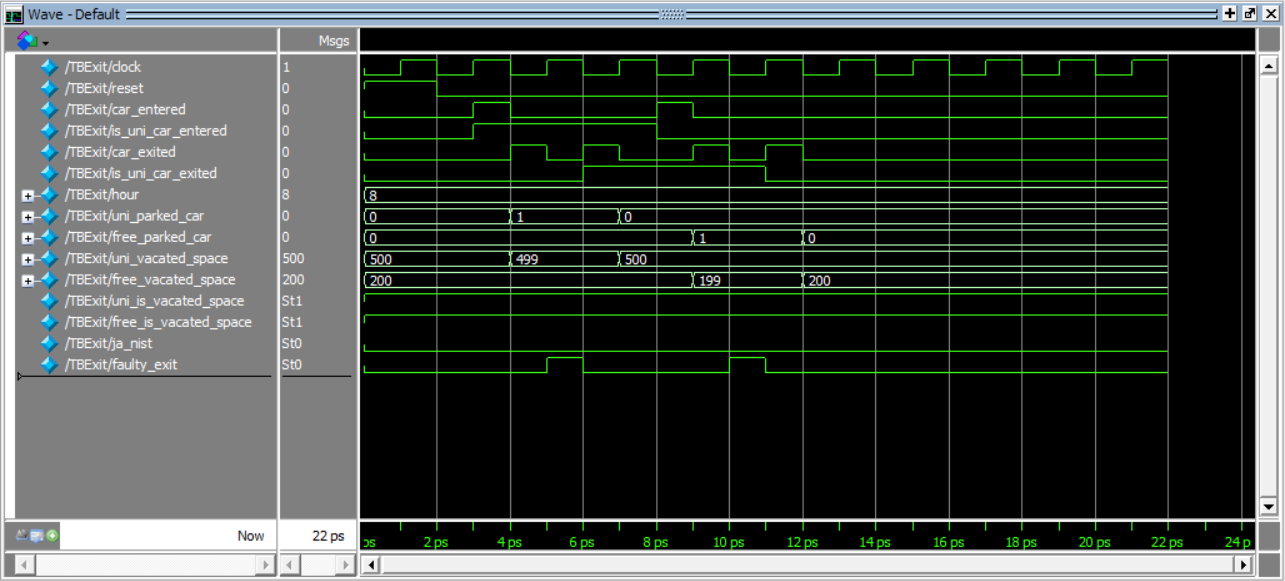
در این تست سیگنال faulty\_exit را بررسی می‌کنیم.

ابتدا یک ماشین دانشگاهی وارد پارکینگ می‌شود و درخواست خروج ماشین آزاد را می‌دهیم. همانطور که انتظار داریم سیگنال faulty\_exit یک می‌شود. سپس ماشین دانشگاهی را خارج می‌کنیم. حالا همین کار را با ماشین آزاد به جای دانشگاهی انجام می‌دهیم.

دوره تناوب کلاک دو نانو‌ثانیه است و قسمت اصلی برنامه محک به این صورت است:



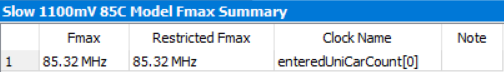
نتایج این تست مطابق زیر است:



سیگنال faulty\_exit با لبه پایین‌رونده car\_exited نا‌معتبر یک می‌شود و در لبه بالا‌رونده بعدی کلاک صفر و reset می‌شود.

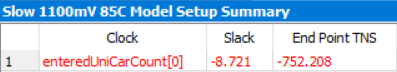
**گزارش سنتز**

با استفاده از نرم‌افزار کوارتوس کد را برای Cyclone IV GX سنتز می‌کنیم. گزارش سنتز به این شکل است:



برای تخمین حداکثر فرکانس مدار می‌توان setup time و hold time را بررسی کرد. چون هر کلاک باید حداقل به اندازه این دو طول بکشد تا سیگنال‌ها را تشخیص دهد.





همانطور که انتظار داشتیم فرکانس به دست آمده در روش دوم به فرکانس گزارش شده توسط ابزار سنتز بسیار نزدیک ولی کمی بیشتر است.