

به نام خدا

جبرانی پایان ترم طراحی سیستم های دیجیتال

"سوال هشت"

توضیحات:

برای این مدار، باید یک بخش ترتیبی داشته باشیم که ساعت های شبانه روز رو هندل و شبیه سازی بکند. همچنین یک بخش دیگر هم میخواهیم که خروج ماشین ها، و یک بخش دیگر ورود ماشین ها رو مشخص بکند.

با توجه به اینکه ورود ماشین ها را به صورت Push Button در نظر گرفته ایم (در حالت عادی 0 است و سپس یک پالس یک می سازد)، با لبه منفی هر سیگنال کار می کنیم تا اثر آن Button را اعمال کنیم.

با توجه به توضیحات قبلی، می توان فهمید که مدار ما ویژگی های زیر را دارد :

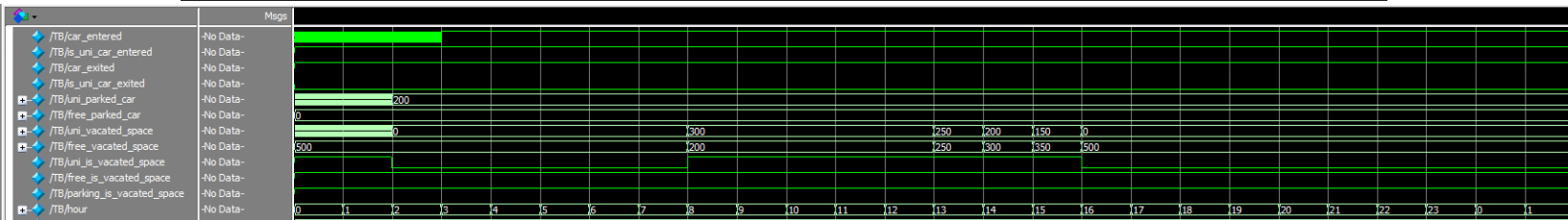
- بخش شبیه سازی زمان که هر 200 واحد زمانی ساعت رو یکی به جلو می برد و ظرفیت های مختلف پارکینگ (آزاد و دانشگاه) رو براساس ساعت جدید ست میکند. در این بخش دقیقاً شرایط گفته شده در مسئله اعمال شده است.
- بخش هندل کننده ورود خودروها که با لبه پایین رونده سیگنال ورود کار میکند. این بخش با استفاده از مقدار سیگنال مشخص کننده نوع ورودی (آزاد یا دانشگاه) در لبه پایین رونده سیگنال ورود، در صورت خالی بودن ظرفیت مربوطه (آزاد و دانشگاه) و همچنین خالی بودن ظرفیت کل پارکینگ (پارک شده آزاد + دانشگاه > ظرفیت کل) پارک شده ها را آپدیت کرده و ورود آن ماشین را اعمال می کند.
- نکته : با توجه به اینکه اگر مثلاً 215 ماشین آزاد در پارکینگ باشد و ظرفیت آزاد از 200 به 250 برود، اگر 486 امین ماشین بخواهد وارد شود، با توجه به اینکه ظرفیت دانشگاه 500 است و هنوز باید جای خالی داشته باشد، ولی چون ظرفیت پارکینگ پر شده است، نمیتواند وارد شود.
- به همین دلیل خروجی های free_is_vacated_space و uni_is_vacated_space که به ترتیب نشان دهنده خالی بودن ظرفیت آزاد و دانشگاه است علاوه بر مثبت بودن فضای خالی مربوطه اش، با خالی بودن ظرفیت کل پارکینگ هم چک و ست می - شوند.
- در نهایت بخش خروج ماشین ها که با لبه پایین رونده سیگنال خروج ماشین و استفاده از نوع ماشین خروجی (آزاد و دانشگاه) تاثیر خروج ماشین را اعمال می کند.

تنها نکته باقی مانده هم خروجی ای است که به مدار اضافه شده است که مربوط به بخش "و همچنین خالی بودن ظرفیت کل پارکینگ (پارک شده آزاد + دانشگاه > ظرفیت کل)" است. این خروجی چک می کند که در مجموع، بیشتر از ظرفیت پارکینگ ورودی نداشته باشیم (در بخش های قبل به کاربرد آن اشاره شد).

تست بنچ:

1. در این تست بنچ در 3 ساعت اول، 300 ماشین دانشگاه درخواست ورود می‌دهند. و چون ظرفیت پارکینگ در آن ساعات 200 ماشین است، ظرفیت دانشگاه پارکینگ پر می‌شود.

```
// Testbench 1:
for (i = 0; i < 600; i = i + 1) begin
    #1 car_entered <= !car_entered;
end
```



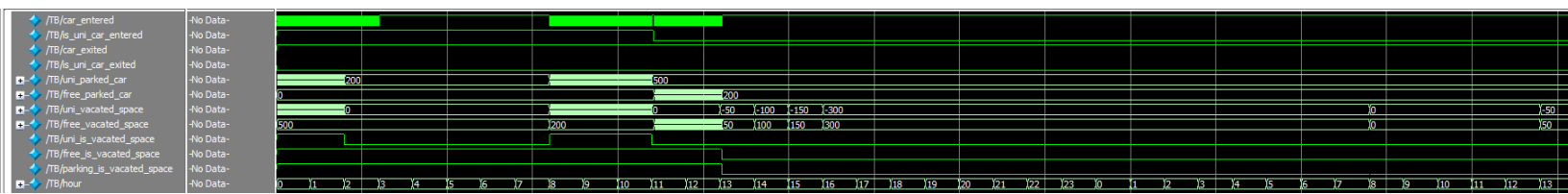
2. در این تست کیس ابتدا تا ساعت 8 صبح 200 ماشین دانشگاه وارد میکنیم، سپس از 8 تا 13، 300 ماشین دانشگاه و 200 ماشین آزاد هم داخل می‌کنیم تا ظرفیت پارکینگ پر شود. سپس وضعیت سیگنال‌های خروجی را مشاهده می‌کنیم. می‌بینیم که ابتدا ظرفیت دانشگاه پر می‌شود، سپس ظرفیت آزاد و در نتیجه ظرفیت کل پارکینگ. پس از ساعت 13 با کاهش ظرفیت دانشگاه و افزایش ظرفیت آزاد، مشاهده می‌شود که فضای خالی دانشگاه منفی و آزاد مثبت می‌شود، ولی چون پارکینگ جای خالی ندارد، سیگنال free_is_vacated_space هم یک نمی‌شود.

```
// Testbench 2:
for (i = 0; i < 600; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#1000
for (i = 0; i < 600; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#10
is_uni_car_entered = 0;

for (i = 0; i < 400; i = i + 1) begin
    #1 car_entered <= !car_entered;
end
```



3. در این تست کیس ابتدا هنگامی که ظرفیت کل پارکینگ پر است، درخواست ورود می‌دهیم و مشاهده می‌کنیم که تغییری در ماشین‌های پارک شده ایجاد نمی‌شود. سپس در ساعت 16 به بعد، تعدادی از ماشین‌های دانشگاه را خارج می‌کنیم، سپس اقدام به ورود تعدادی ماشین دانشگاه می‌کنیم که مشاهده می‌کنیم چون ظرفیت خالی برای دانشگاه هنوز وجود ندارد، موفق نمی‌شویم. سپس تعدادی ماشین آزاد وارد می‌کنیم تا دوباره ظرفیت پارکینگ پر شود.

```
// Testbench 3:
for (i = 0; i < 600; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#1000
for (i = 0; i < 600; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#10
is_uni_car_entered = 0;

for (i = 0; i < 400; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#100
for (i = 0; i < 50; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#500
is_uni_car_exited = 1;
for (i = 0; i < 50; i = i + 1) begin
    #1 car_exited <= !car_exited;
end
```

```

#100
is_uni_car_entered = 1;
for (i = 0; i < 50; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

#100
is_uni_car_entered = 0;
for (i = 0; i < 100; i = i + 1) begin
    #1 car_entered <= !car_entered;
end

```

