

به نام خدا



جبرانی آزمون پایانی

طراحی سیستم‌های دیجیتال

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نویسنده:

رادین چراغی

شماره دانشجویی:

۴۰۱۱۰۵۸۱۵

تاریخ ارائه تکلیف:

۱۴۰۳/۰۴/۶

توضیحات سوال ۸

در این سوال هدف ما طراحی مداری برای مدیریت پارکینگ یک دانشگاه می‌باشد. در این پارکینگ به طور کلی اولویت با کارمندان دانشگاه بوده و در ابتدا ظرفیت ۵۰۰ برای آن‌ها در نظر گرفته شده است. از آن جایی که ظرفیت کل پارکینگ ۷۰۰ می‌باشد، ظرفیت در نظر گرفته شده برای ورود آزاد در ابتدا ۲۰۰ می‌باشد. از ساعت ۱۳ تا ۱۶ ظرفیت ورود آزاد به ازای هر ساعت ۵۰ عدد افزایش پیدا کرده و در ساعت ۱۶ به ۵۰۰ می‌رسد.

حال به طراحی ماژول Parking در وریلاگ می‌پردازیم.

ورودی و خروجی‌های این ماژول با توضیحات در تصویر زیر مشخص شده است.

خروجی‌ها	
uni_parked_car	تعداد خودروهایی متعلق به دانشگاه که در پارکینگ پارک شده‌اند.
parked_care	تعداد خودروهای پارک شده در پارکینگ مربوط به ظرفیت آزاد
uni_vacated_space	تعداد فضای خالی متعلق به دانشگاه
vacated_space	تعداد فضاهای خالی مربوط به ظرفیت آزاد
uni_is_vacated_space	آیا فضای خالی برای دانشگاه موجود است؟
is_vacated_space	آیا فضای خالی برای ظرفیت آزاد موجود است؟
ورودی‌ها	
car_entered	ورود یک خودرو
is_uni_car_entered	آیا خودرو وارد شده متعلق به دانشگاه است؟
car_exited	خروج یک خودرو
is_uni_car_exited	آیا خودرو خارج شده متعلق به دانشگاه است؟

بایستی دقت شود که خروجی‌های uni_is_vacated_space و is_vacated_space تک بیتی و سایر خروجی‌ها ۱۰ بیتی می‌باشند. همچنین تمامی ورودی‌ها تک بیتی هستند.

علاوه بر ورودی‌های بالا یک ورودی ۵ بیتی دیگر به نام current_time برای این ماژول در نظر می‌گیریم که نشان‌دهنده ساعت زمان فعلی بوده و از ۰ تا ۲۳ متغیر می‌باشد.

کد زیر بیانگر تعریف ماژول و ورودی/ خروجی‌های آن می‌باشد.

```
module Parking
(
    input [4:0] current_time,
    input car_entered,
    input is_uni_car_entered,
    input car_exited,
    input is_uni_car_exited,
    output reg [9:0] uni_parked_car,
    output reg [9:0] parked_car,
    output reg [9:0] uni_vacated_space,
    output reg [9:0] vacated_space,
    output uni_is_vacated_space,
    output is_vacated_space
);
```

حال دو رجیستر ۱۰ بیتی `total_free_space` و `total_space_for_uni` را تعریف می‌کنیم که به ترتیب نشان‌دهنده ظرفیت بخش کارمندان و ظرفیت بخش آزاد می‌باشد.

با استفاده از دستور `assign` دو خروجی تک بیتی را مقدار دهی می‌کنیم. تخصیص مقدار به این صورت است که به عنوان مثال برای `is_vacated_space` در صورتی که تعداد ماشین‌های پارک شده به صورت آزاد از ظرفیت کلی در نظر گرفته شده برای ورود آزاد کمتر باشد مقدار ۱ و در غیر این صورت مقدار صفر را به آن می‌دهیم.

حال با استفاده از یک بلاک `initial` به خروجی‌ها یک مقدار اولیه می‌دهیم. تصویر نشان‌دهنده عملیات انجام شده می‌باشد.

```
reg[9:0] total_space_for_uni;
reg[9:0] total_free_space;

// setting booleans for spaces
assign uni_is_vacated_space = (uni_parked_car < total_space_for_uni);
assign is_vacated_space = (parked_car < total_free_space);

initial begin // TODO: check for synthesis
    uni_vacated_space = 500;
    vacated_space = 200;
    uni_parked_car = 0;
    parked_car = 0;
end
```

اکنون با استفاده از یک `always` حساس به `current_time` تاثیر تغییرات زمان که قبل تر توضیح داده شد را بر روی تغییر ظرفیت مربوط به کارمندان و ورود آزاد اعمال می‌کنیم. سپس فضای خالی مربوط هر یک را آپدیت می‌کنیم. در این قسمت یک نکته قابل توجه وجود دارد. در ساعت‌های مختلف که ظرفیت آزاد و ظرفیت مخصوص به استادها تغییر می‌کند، ممکن است

شرایطی پیش بیاید که تعداد فضاهای خالی برای یکی از گروه ها منفی شود. در صورتی که ظرفیت مربوط به کارمندان منفی شد و فضای خالی در قسمت ورود آزاد وجود داشت خودروهای کارمندان را به قسمت آزاد منتقل می کنیم. در غیر این صورت یا در شرایطی که ظرفیت بخش آزاد منفی شود خودروهای اضافی را از پارکینگ خارج می کنیم.

تصویر نشان دهنده عملیات انجام شده می باشد.

```
always @(current_time) begin
    // In three following if-else we set total_free_space
    // and total_space_for_uni with respect to current_time
    if(current_time >= 8 && current_time < 13) begin
        total_space_for_uni = 500;
        total_free_space = 200;
    end
    else if(current_time >= 13 && current_time < 16) begin
        total_free_space = 200 + 50 * (current_time - 12);
        total_space_for_uni = 700 - total_free_space;
    end
    else if(current_time >= 16) begin
        total_space_for_uni = 200;
        total_free_space = 500;
    end
    vacated_space = total_free_space - parked_car;
    uni_vacated_space = total_space_for_uni - uni_parked_car;

    // This if checks weather an uni cars are more that uni_space

    if(uni_vacated_space[9]) begin
        uni_vacated_space = 0;
        if(vacated_space > 0) begin
            parked_car = parked_car + uni_parked_car - total_space_for_uni;
            vacated_space = total_free_space - parked_car;
        end
        uni_parked_car = total_space_for_uni;
    end
    if(vacated_space[9]) begin
        vacated_space = 0;
        parked_car = total_free_space;
    end
end
```

حال با یک بلاک **always** حساس به لبه‌ی بالارونده **car_enter** و **car_exit** به بررسی ورود و خروج به پارکینگ می پردازیم.

در ابتدا بررسی می کنیم آیا زمان فعلی از ۸ گذشته و پارکینگ باز می باشد یا خیر. در صورتی از ۸ گذشته باشد، بررسی می کنیم که آیا خودرویی وارد شده یا خیر و با توجه به نوع خودرو بروزرسانی های لازم برای خروجی ها را انجام می دهیم. برای خروج خودرو نیز به همین صورت می باشد. حال یک نکته قابل توجه وجود دارد. در هنگام ورودی خودروی مربوط به کارمندان ممکن است شرایطی پیش بیاید ظرفیت مربوط به کارمندان صفر باشد و فضای خالی در قسمت ورود آزاد وجود داشته باشد. در این صورت

خودروی کارمند را به قسمت آزاد منتقل می‌کنیم. در شرایطی که ظرفیت بخش آزاد صفر باشد و خودرو ورودی آزاد باشد این خودرو اجازه ورود پیدا نخواهد کرد.

تصویر نشان‌دهنده عملیات انجام شده می‌باشد.

```
always @(posedge car_exited or posedge car_entered) begin
    // checking entering or exit of a car
    if(current_time >= 8) begin
        if(car_entered) begin
            if(is_uni_car_entered && uni_is_vacated_space) begin
                uni_parked_car = uni_parked_car + 1;
                uni_vacated_space = total_space_for_uni - uni_parked_car;
            end
            else if(is_vacated_space) begin
                parked_car = parked_car + 1;
                vacated_space = total_free_space - parked_car;
            end
        end
        end

        if (car_exited) begin
            if (!is_uni_car_exited && (parked_car > 0)) begin
                parked_car = parked_car - 1;
                vacated_space = total_free_space - parked_car;
            end
            else if (is_uni_car_exited && (uni_parked_car > 0)) begin
                uni_parked_car = uni_parked_car - 1;
                uni_vacated_space = total_space_for_uni - uni_parked_car;
            end
        end
    end
end
end
```

حال تست بنچ مربوط به مدار را طراحی می‌کنیم. ابتدا سیگنال‌های لازم برای ورودی و خروجی مدار را تعریف کرده و از ماژول Parking یک نمونه می‌گیریم. سپس با استفاده از یک بلاک initial و یک بلاک always زمان را شبیه‌سازی می‌کنیم. روش شبیه‌سازی به این صورت است که current_time را در ابتدا صفر کرده و سپس هر ۶۰ واحد زمانی یک بار به آن یک واحد اضافه کرده و به ۲۴ باقیمانده می‌گیریم.

تصویر نشان‌دهنده عملیات انجام شده می‌باشد.

```

module TB;
    reg [4:0] current_time;
    reg car_entered, is_uni_car_entered, car_exited, is_uni_car_exited;

    wire [9:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space;
    wire uni_is_vacated_space, is_vacated_space;

    Parking parking
    (
        current_time, car_entered, is_uni_car_entered, car_exited,
        is_uni_car_exited, uni_parked_car, parked_car,
        uni_vacated_space, vacated_space, uni_is_vacated_space, is_vacated_space
    );

    integer i;
    initial current_time = 0;
    always #60 current_time = (current_time + 1) % 24; // simulation of hour time

```

اکنون یک بلاک initial تعریف کرده و تست‌ها را در آن قرار می‌دهیم. در ابتدا برای آزمون بهتر ۴۰۰ خودرو را وارد پارکینگ می‌کنیم.

```

car_entered = 0; is_uni_car_entered = 0; car_exited = 0; is_uni_car_exited = 0;
for(i = 0; i < 400; i = i + 1) begin
    current_time = 8;
    car_entered = 1;
    is_uni_car_entered = 1;
    #10;
    car_entered = 0;
    is_uni_car_entered = 0; #10;
end

```

حال ۳۰۰ تست رندوم با تمپلیتی به صورت زیر قرار می‌دهیم. این تست‌ها را با یک کد پایتون جنریت کرده‌ایم که خود کد و خروجی آن نیز در پیوست آورده شده‌اند. برای جزئیات بیشتر به فایل تست بنچ مراجعه شود.

```

template = """
#10
$display("{display_message}");
is_uni_car_entered={uni_car_entered_flag};
is_uni_car_exited={uni_car_exited_flag};
car_entered={uni_car_flag}; car_exited={other_car_flag};
#10
$display("time:%d uni_parked_car: %d parked_car: %d uni_vacated_space: %d vacated_space: %d uni is vacated_space: %d is_vacated_space: %d\n",
current_time, uni_parked_car, parked_car, uni_vacated_space, vacated_space, uni_is_vacated_space, is_vacated_space);
is_uni_car_entered={other_car_entered_flag};
is_uni_car_exited={other_car_exited_flag};
car_entered={no_car_flag_1}; car_exited={no_car_flag_2};
"""

```

حال مدار را در نرم افزار ModelSim شبیه سازی کرده و خروجی را در transcript مشاهده می کنیم.

تصاویر زیر نشان دهنده خروجی مدار در بازه های زمانی مختلف می باشند.

```
# entrance of university
# time: 9 uni_parked_car: 400 parked_car: 1 uni_vacated_space: 100 vacated_space: 199 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of free car
# time: 9 uni_parked_car: 400 parked_car: 2 uni_vacated_space: 100 vacated_space: 198 uni is vacated_space: 1 is_vacated_space: 1
#
# exit of free car
# time: 9 uni_parked_car: 400 parked_car: 1 uni_vacated_space: 100 vacated_space: 199 uni is vacated_space: 1 is_vacated_space: 1
#
# exit of university car
# time:10 uni_parked_car: 399 parked_car: 1 uni_vacated_space: 101 vacated_space: 199 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of free car
# time:10 uni_parked_car: 399 parked_car: 2 uni_vacated_space: 101 vacated_space: 198 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of university
# time:10 uni_parked_car: 400 parked_car: 2 uni_vacated_space: 100 vacated_space: 198 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of free car
# time:11 uni_parked_car: 400 parked_car: 3 uni_vacated_space: 100 vacated_space: 197 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of university
# time:11 uni_parked_car: 401 parked_car: 3 uni_vacated_space: 99 vacated_space: 197 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of university
# time:11 uni_parked_car: 402 parked_car: 3 uni_vacated_space: 98 vacated_space: 197 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of free car
# time:12 uni_parked_car: 402 parked_car: 4 uni_vacated_space: 98 vacated_space: 196 uni is vacated_space: 1 is_vacated_space: 1
#
.
```

```
#
# entrance of university
# time:13 uni_parked_car: 402 parked_car: 5 uni_vacated_space: 98 vacated_space: 195 uni is vacated_space: 1 is_vacated_space: 1
#
# exit of free car
# time:13 uni_parked_car: 402 parked_car: 4 uni_vacated_space: 48 vacated_space: 246 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of university
# time:13 uni_parked_car: 403 parked_car: 4 uni_vacated_space: 47 vacated_space: 246 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of free car
# time:14 uni_parked_car: 403 parked_car: 5 uni_vacated_space: 47 vacated_space: 245 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of university
# time:14 uni_parked_car: 400 parked_car: 9 uni_vacated_space: 0 vacated_space: 291 uni is vacated_space: 0 is_vacated_space: 1
#
# exit of university car
# time:14 uni_parked_car: 399 parked_car: 9 uni_vacated_space: 1 vacated_space: 291 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of free car
# time:15 uni_parked_car: 399 parked_car: 10 uni_vacated_space: 1 vacated_space: 290 uni is vacated_space: 1 is_vacated_space: 1
#
# entrance of university
# time:15 uni_parked_car: 350 parked_car: 60 uni_vacated_space: 0 vacated_space: 290 uni is vacated_space: 0 is_vacated_space: 1
#
# exit of university car
# time:15 uni_parked_car: 349 parked_car: 60 uni_vacated_space: 1 vacated_space: 290 uni is vacated_space: 1 is_vacated_space: 1
#
# exit of university car
# time:16 uni_parked_car: 348 parked_car: 60 uni_vacated_space: 2 vacated_space: 290 uni is vacated_space: 1 is_vacated_space: 1
#
```

entrance of university	time:17	uni_parked_car: 200	parked_car: 207	uni_vacated_space: 0	vacated_space: 293	uni is vacated_space: 0	is_vacated_space: 1
entrance of university	time:17	uni_parked_car: 200	parked_car: 208	uni_vacated_space: 0	vacated_space: 292	uni is vacated_space: 0	is_vacated_space: 1
entrance of free car	time:17	uni_parked_car: 200	parked_car: 209	uni_vacated_space: 0	vacated_space: 291	uni is vacated_space: 0	is_vacated_space: 1
exit of free car	time:18	uni_parked_car: 200	parked_car: 208	uni_vacated_space: 0	vacated_space: 292	uni is vacated_space: 0	is_vacated_space: 1
exit of university car	time:18	uni_parked_car: 199	parked_car: 208	uni_vacated_space: 1	vacated_space: 292	uni is vacated_space: 1	is_vacated_space: 1
entrance of university	time:18	uni_parked_car: 200	parked_car: 208	uni_vacated_space: 0	vacated_space: 292	uni is vacated_space: 0	is_vacated_space: 1
exit of university car	time:19	uni_parked_car: 199	parked_car: 208	uni_vacated_space: 1	vacated_space: 292	uni is vacated_space: 1	is_vacated_space: 1
exit of free car	time:19	uni_parked_car: 199	parked_car: 207	uni_vacated_space: 1	vacated_space: 293	uni is vacated_space: 1	is_vacated_space: 1
exit of free car	time:19	uni_parked_car: 199	parked_car: 206	uni_vacated_space: 1	vacated_space: 294	uni is vacated_space: 1	is_vacated_space: 1
entrance of free car	time:20	uni_parked_car: 199	parked_car: 207	uni_vacated_space: 1	vacated_space: 293	uni is vacated_space: 1	is_vacated_space: 1
exit of university car	time:20	uni_parked_car: 198	parked_car: 207	uni_vacated_space: 2	vacated_space: 293	uni is vacated_space: 1	is_vacated_space: 1
# exit of free car	time:21	uni_parked_car: 198	parked_car: 205	uni_vacated_space: 2	vacated_space: 295	uni is vacated_space: 1	is_vacated_space: 1
# entrance of free car	time:21	uni_parked_car: 198	parked_car: 206	uni_vacated_space: 2	vacated_space: 294	uni is vacated_space: 1	is_vacated_space: 1
# exit of university car	time:21	uni_parked_car: 197	parked_car: 206	uni_vacated_space: 3	vacated_space: 294	uni is vacated_space: 1	is_vacated_space: 1
# exit of free car	time:22	uni_parked_car: 197	parked_car: 205	uni_vacated_space: 3	vacated_space: 295	uni is vacated_space: 1	is_vacated_space: 1
# entrance of university	time:22	uni_parked_car: 198	parked_car: 205	uni_vacated_space: 2	vacated_space: 295	uni is vacated_space: 1	is_vacated_space: 1
# entrance of free car	time:22	uni_parked_car: 198	parked_car: 206	uni_vacated_space: 2	vacated_space: 294	uni is vacated_space: 1	is_vacated_space: 1
# entrance of free car	time:23	uni_parked_car: 198	parked_car: 207	uni_vacated_space: 2	vacated_space: 293	uni is vacated_space: 1	is_vacated_space: 1
# entrance of free car	time:23	uni_parked_car: 198	parked_car: 208	uni_vacated_space: 2	vacated_space: 292	uni is vacated_space: 1	is_vacated_space: 1
# exit of university car	time:23	uni_parked_car: 197	parked_car: 208	uni_vacated_space: 3	vacated_space: 292	uni is vacated_space: 1	is_vacated_space: 1
# entrance of free car	time: 0	uni_parked_car: 197	parked_car: 209	uni_vacated_space: 3	vacated_space: 291	uni is vacated_space: 1	is_vacated_space: 1
# entrance of university	time: 0	uni_parked_car: 197	parked_car: 209	uni_vacated_space: 3	vacated_space: 291	uni is vacated_space: 1	is_vacated_space: 1