

# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

دانشگاه صنعتی شریف

سوال جبرانی درس طراحی سیستم های دیجیتال

استاد مهندس امین فصحتی

دانشجو سبحان ارشادی

## سؤال ۸:

مداری برای مدیریت پارکینگ دانشگاه طراحی کنید که امکانات زیر را داشته باشد:

- (۱) اولویت فضای پارکینگ با اساتید و کارمندان دانشگاه است و این ظرفیت بر اساس آمار حداکثر ۵۰۰ خودرو تعیین گردیده است.
- (۲) باتوجه به اینکه فضای کل پارکینگ ۷۰۰ خودرو است از ساعت ۸ تا ۱۳ فقط ۲۰۰ ظرفیت خالی برای ورود آزاد موجود است.
- (۳) از ساعت ۱۳ تا ۱۶ به ازای هر ساعت ظرفیت ورود آزاد ۵۰ خودرو افزایش می یابد و در ساعت ۱۶ ظرفیت ورود آزاد به ۵۰۰ خودرو می رسد.

**الف)** اگر در هنگام ورود/خروج خودرو یک سیگنال ورودی به مدار نوع آن را مشخص کند (دانشگاه/آزاد)؛ با زبان وریلاگ مداری را توصیف کنید که دارای ورودی‌ها/خروجی‌های زیر باشد:

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؟

در صورتی که نیاز به ورودی‌ها/خروجی‌های دیگری هم است آن را با ذکر دلیل به طراحی خود بیفزایید و جهت اطمینان از صحت عملکرد مدار، مدار خود را مورد آزمون قرار دهید (۵۰ نمره).

**ب)** مدار خود را برای یک FPGA به انتخاب خود سنتز کنید. از گزارش‌های سنتز، بیشترین فرکانس ممکن برای این مدار را با ذکر دلیل مشخص کنید (۱۰ نمره).

بخش الف)

شروع به پیاده سازی می کنیم. با توجه به جدول روی سوال ماژول خود را ساخته و ورودی و خروجی ها را مشخص می کنیم:

```
module parking (  
    input wire clk,  
    input wire reset,  
    input wire car_entered,  
    input wire is_uni_car_entered,  
    input wire car_exited,  
    input wire is_uni_car_exited,  
    input wire [4:0] hour,  
    output reg [8:0] uni_parked_car,  
    output reg [8:0] parked_car,  
    output reg [8:0] uni_vacated_space,  
    output reg [8:0] vacated_space,  
    output reg uni_is_vacated_space,  
    output reg is_vacated_space  
);
```

غیر از موارد صورت سوال ما دارای ورودی و خروجی های اضافی هستیم:

clk: برای نمایش کلاک و تایم مدار و نیز synchronys کردن مدار

reset: برای ریست مدار و برقراردادن به حالت اول

hour: یک خروجی 5 بیتی (برای نمایش اعداد از 0 تا 23 به 5 بیت نیاز داریم) است و ساعت پارکینگ را با آن تنظیم می کنیم.

```
// پارامترهای ثابت  
parameter MAX_UNI_CARS = 500;  
parameter TOTAL_CAPACITY = 700;
```

این دو متغیر برای ظرفیت پارکینگ طبق صورت سوال تعریف شده اند.

```
// ظرفیت آزاد  
reg [8:0] free_capacity;  
always @(*) begin  
    if (hour > 8 && hour < 13) begin  
        free_capacity = 200;  
    end else if (hour < 16) begin  
        free_capacity = 200 + (hour - 13) * 50;  
    end else begin  
        free_capacity = 500;  
    end  
end  
end
```

طبق صورت سوال باید برحسب ساعت، ظرفیت آزاد را تعریف کنیم. کد این بلاک خیلی سخت نیست و براساس صورت سوال نوشته شده است.

```
// شمارنده خودروها
always @(posedge clk or posedge reset) begin

    if (reset) begin
        uni_parked_car <= 0;
        parked_car <= 0;
    end else begin
        if (car_entered) begin
            if (is_uni_car_entered && uni_is_vacated_space) begin
                uni_parked_car <= uni_parked_car + 1;
            end else if ((!is_uni_car_entered) && is_vacated_space) begin
                parked_car <= parked_car + 1;
            end
        end
        if (car_exited) begin
            if (is_uni_car_exited && uni_parked_car > 0) begin
                uni_parked_car <= uni_parked_car - 1;
            end else if (!is_uni_car_exited && parked_car > 0) begin
                parked_car <= parked_car - 1;
            end
        end
    end
end
end
```

این بلاک در اصل کنترلر اصلی پارکینگ ما است. در آن ورود و خروج خودروها بر اساس دانشگاهی بودن یا نبودن آنان تعیین می شود. بخش reset نیز در اینجا طراحی شده است که تعداد خودروها را 0 می کند.

```
// محاسبه فضاهای خالی
always @(*) begin
    uni_vacated_space = MAX_UNI_CARS - uni_parked_car;
    vacated_space = free_capacity - parked_car;
    uni_is_vacated_space = (uni_vacated_space > 0) &&
        ((TOTAL_CAPACITY - parked_car - uni_parked_car) > 0);
    is_vacated_space = (vacated_space > 0) &&
        ((TOTAL_CAPACITY - parked_car - uni_parked_car) > 0);
end

endmodule
```

بخش پایانی ماژول. در این بلاک تعداد ظرفیت در لحظه برای ورود خودرو ها بررسی می شود. دو شرط برای داشتن ظرفیت بررسی می گردد: 1- اینکه ظرفیت مربوط به آن بخش پر نشده باشد. 2- اینکه تعداد کل ماشین ها از 700 تا تجاوز نکند.

حال به ماژول تست بنچ می رسیم.

```
module tb_parking();

    // Inputs
    reg clk;
    reg reset;
    reg car_entered;
    reg is_uni_car_entered;
    reg car_exited;
    reg is_uni_car_exited;
    reg [4:0] hour;

    // Outputs
    wire [8:0] uni_parked_car;
    wire [8:0] parked_car;
    wire [8:0] uni_vacated_space;
    wire [8:0] vacated_space;
    wire uni_is_vacated_space;
    wire is_vacated_space;

    // Instantiate the parking module
    parking dut (
        .clk(clk),
        .reset(reset),
        .car_entered(car_entered),
        .is_uni_car_entered(is_uni_car_entered),
        .car_exited(car_exited),
        .is_uni_car_exited(is_uni_car_exited),
        .hour(hour),
        .uni_parked_car(uni_parked_car),
        .parked_car(parked_car),
        .uni_vacated_space(uni_vacated_space),
        .vacated_space(vacated_space),
        .uni_is_vacated_space(uni_is_vacated_space),
        .is_vacated_space(is_vacated_space)
    );
```

در این تیکه از کد به تعریف tb و وصل کردن آن به ماژول پارکینگ می پردازیم. یعنی از ماژول parking اینستنت میگیریم.

```

task show; begin
    $display("Time: %t,
    vorodi (reset: %d, car_entered: %d, is_uni_car_entered: %d, car_exited: %d, is_uni_car_exited: %d, hour: %d)",
    khoroji (uni_parked_car: %d, parked_car: %d, uni_vacated_space: %d, vacated_space: %d, uni_is_vacated_space: %d, is_vacated_space: %d)",
    $time, reset, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, hour,
    uni_parked_car, parked_car, uni_vacated_space, vacated_space, uni_is_vacated_space, is_vacated_space)];
end
endtask

```

این تستک برای نمایش ورودی و خروجی مدار استفاده می شود که در کد هر کجا نیاز داشته باشیم آنرا فرا می خوانیم.

```

// Clock generation
always #5 clk = ~clk;
integer i;
// Test scenarios
initial begin
    $display("reset and hour is seted between 8 and 13:");
    clk = 0;
    reset = 1;
    car_entered = 0;
    is_uni_car_entered = 0;
    car_exited = 0;
    is_uni_car_exited = 0;
    hour = 10;

    #10
    show;

    reset = 0;

    $display("a uni car entered:");
    car_entered = 1;
    is_uni_car_entered = 1;
    #10;
    show;

    $display("a free car entered:");
    is_uni_car_entered = 0;
    #10;
    show;

```

در ابتدا کلاک را به طوری قرار می دهیم که هر 5 واحد زمانی تریگر کند.

سپس شروع به نوشتن تست بنچ ها می کنیم:

در اولی همه چیز را ریست می کنیم.

در دو tb بعدی دو نوع مختلف ماشین وارد پارکینگ می کنیم.

```

car_entered = 0;
$display("a uni car exited:");
car_exited = 1;
is_uni_car_exited = 1;
#10;
show;

$display("a free car exited:");
is_uni_car_exited = 0;
#10;
show;

$display("hour set to 13:");
hour=13;
#10;
show;

$display("hour set to 14:");
hour=14;
#10;
show;
#10

```

حال خروج را بررسی می کنیم و سپس به سراغ تغییر ساعت می رویم. با تغییر ساعت مشاهده خواهیم کرد که ظرفیت آزاد افزایش می یابد.

```

for (i = 0; i < 250 ;i=i+1 ) begin
    car_entered = 1;
    is_uni_car_entered = 0;
    #10;
end
$display("the space of free cars is 0:");
#10
show;
#10

for (i = 0; i < 500;i=i+1 ) begin
    car_entered = 1;
    is_uni_car_entered = 1;
    #10;
end
$display("the space of uni cars is 0:");
#10
show;

```



در دو تا for بعدی سعی داریم که ظرفیت پارکینگ را پرکنیم. و سیگنال خروجی را ببازماییم.

```
$display("reset:");
reset=1;
#10;
show;

$display("hour set to 16:");
hour=16;
#10;
show;

for (i = 0; i<500;i=i+1 ) begin
    car_entered = 1;
    is_uni_car_entered = 0;
    #10;
end
$display("the space of free cars is 0:");
#10
show;

for (i = 0; i<200;i=i+1 ) begin
    car_entered = 1;
    is_uni_car_entered = 1;
    #10;
end
$display("the space of uni cars is 0:");
#10
show;
```

حال پارکینگ را خالی می کنیم و به سراغ ساعت 16 می رویم که در آن ظرفیت آزاد به 500 می رسد. در دو تا tb پایین تصویر نیز مثل تصویر قبلی سعی می کنیم پارکینگ را این بار در این ساعت پرکنیم.

```

$display("reset:");
reset=1;
#10;
show;

for (i = 0; i<300;i=i+1 ) begin
    car_entered = 1;
    is_uni_car_entered = 0;
    #10;
end
$display("the space of free cars is 0:");
#10
show;

for (i = 0; i<400;i=i+1 ) begin
    car_entered = 1;
    is_uni_car_entered = 1;
    #10;
end
$display("the space of uni cars is 0:");
#10
show;

#10 $stop;
end
endmodule

```

این قسمت از تست بنچ ها مهم است. ابتدا reset می کنیم و سپس 300 ظرفیت از پارکینگ را به ظرفیت آزاد می دهیم. در آخر tb بقیه ظرفیت پارکینگ را با خودرو های دانشگاه پر می کنیم. با اینکه هنوز 100 نفر از ظرفیت دانشگاه مانده است اما ظرفیت کل پارکینگ پر شده است.

در آخر نتیجه تست بنچ ها را بعد از اجرا ملاحظه می کنید:

```
SIM 34> run -all
# reset and hour is seted between 8 and 13:
# Time: 10,
#
# vorodi (reset: 1, car_entered: 0, is_uni_car_entered: 0, car_exited: 0, is_uni_car_exited: 0, hour: 10)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 200, uni_is_vacated_space: 1, is_vacated_space: 1)
# a uni car entered:
# Time: 20,
#
# vorodi (reset: 0, car_entered: 1, is_uni_car_entered: 1, car_exited: 0, is_uni_car_exited: 0, hour: 10)
#
# khoroji (uni_parked_car: 1, parked_car: 0, uni_vacated_space: 499, vacated_space: 200, uni_is_vacated_space: 1, is_vacated_space: 1)
# a free car entered:
# Time: 30,
#
# vorodi (reset: 0, car_entered: 1, is_uni_car_entered: 0, car_exited: 0, is_uni_car_exited: 0, hour: 10)
#
# khoroji (uni_parked_car: 1, parked_car: 1, uni_vacated_space: 499, vacated_space: 199, uni_is_vacated_space: 1, is_vacated_space: 1)
# a uni car exited:
# Time: 40,
#
# vorodi (reset: 0, car_entered: 0, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 1, hour: 10)
#
# khoroji (uni_parked_car: 0, parked_car: 1, uni_vacated_space: 500, vacated_space: 199, uni_is_vacated_space: 1, is_vacated_space: 1)
# a free car exited:
# Time: 50,
#
# vorodi (reset: 0, car_entered: 0, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 0, hour: 10)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 200, uni_is_vacated_space: 1, is_vacated_space: 1)
# hour set to 13:
# Time: 60,
#
# vorodi (reset: 0, car_entered: 0, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 0, hour: 13)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 200, uni_is_vacated_space: 1, is_vacated_space: 1)
# hour set to 14:
# Time: 70,
#
# vorodi (reset: 0, car_entered: 0, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 0, hour: 14)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 250, uni_is_vacated_space: 1, is_vacated_space: 1)
# the space of free cars is 0:
# Time: 2590,
#
# vorodi (reset: 0, car_entered: 1, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 0, hour: 14)
#
# khoroji (uni_parked_car: 0, parked_car: 1, uni_vacated_space: 500, vacated_space: 249, uni_is_vacated_space: 1, is_vacated_space: 1)
# the space of uni cars is 0:
# Time: 7610,
#
# vorodi (reset: 0, car_entered: 1, is_uni_car_entered: 1, car_exited: 1, is_uni_car_exited: 0, hour: 14)
#
# khoroji (uni_parked_car: 500, parked_car: 0, uni_vacated_space: 0, vacated_space: 250, uni_is_vacated_space: 0, is_vacated_space: 1)
# reset:
# Time: 7620,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 1, car_exited: 1, is_uni_car_exited: 0, hour: 14)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 250, uni_is_vacated_space: 1, is_vacated_space: 1)
# hour set to 16:
# Time: 7630,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 1, car_exited: 1, is_uni_car_exited: 0, hour: 16)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 500, uni_is_vacated_space: 1, is_vacated_space: 1)
# the space of free cars is 0:
# Time: 12640,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 0, hour: 16)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 500, uni_is_vacated_space: 1, is_vacated_space: 1)
```

```

# the space of uni cars is 0:
# Time: 14650,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 1, car_exited: 1, is_uni_car_exited: 0, hour: 16)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 500, uni_is_vacated_space: 1, is_vacated_space: 1)
# reset:
# Time: 14660,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 1, car_exited: 1, is_uni_car_exited: 0, hour: 16)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 500, uni_is_vacated_space: 1, is_vacated_space: 1)
# the space of free cars is 0:
# Time: 17670,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 0, car_exited: 1, is_uni_car_exited: 0, hour: 16)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 500, uni_is_vacated_space: 1, is_vacated_space: 1)
# the space of uni cars is 0:
# Time: 21680,
#
# vorodi (reset: 1, car_entered: 1, is_uni_car_entered: 1, car_exited: 1, is_uni_car_exited: 0, hour: 16)
#
# khoroji (uni_parked_car: 0, parked_car: 0, uni_vacated_space: 500, vacated_space: 500, uni_is_vacated_space: 1, is_vacated_space: 1)
# ** Note: $stop : F:/DSD/Extra score/tb.v(169)
# Time: 21690 ps Iteration: 0 Instance: /tb_parking
# Break in Module tb_parking at F:/DSD/Extra score/tb.v line 169

```

: نتیجه waveform



