

به نام خدا

طراحی سیستم‌های دیجیتال - پروژه‌ی جبرانی

مهدی شکوفی مقیمان

۴۰۱۱۱۰۱۱۵

با توجه به خواسته‌های سوال، ماژول parking_management را طراحی می‌کنیم:

ورودی‌ها و خروجی‌های مورد نیاز (همان ورودی و خروجی‌های متن سوال است، با توجه به نیازمان یک سیگنال ورودی current_hour نیز اضافه کرده‌ایم):

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

تعریف محدودیت‌های پارکینگ:

```
//parking constraints  
parameter MAX_TOTAL_CAPACITY = 700;  
reg [9:0] max_free_capacity;  
wire [9:0] uni_space;  
  
assign uni_space = MAX_TOTAL_CAPACITY - max_free_capacity;
```

محاسبه‌ی میزان فضای خالی:

```
//determine free spaces
assign uni_vacated_space = uni_space - uni_parked_car;
assign vacated_space = max_free_capacity - parked_car;
```

محاسبه‌ی خالی بودن پارکینگ:

```
//determine does free space exist
assign uni_is_vacated_space = (uni_vacated_space > 0);
assign is_vacated_space = (vacated_space > 0);
```

محاسبه‌ی ظرفیت‌ها با توجه به ساعت کنونی:

```
//determine capacity with current_hour
always @(posedge clk or posedge reset) begin
    if (reset) begin
        max_free_capacity <= 500;
    end else begin
        if (current_hour < 8)
            max_free_capacity <= 500;
        else if (current_hour >= 8 && current_hour < 13)
            max_free_capacity <= 200;
        else if (current_hour >= 13 && current_hour < 16)
            max_free_capacity <= 200 + (current_hour - 12) * 50;
        else
            max_free_capacity <= 500;
    end
end
```

قسمت اصلی کد (کنترل ورود و خروج ماشین‌ها):

```
//main
always @(posedge clk or posedge reset) begin
    if (reset) begin
        uni_parked_car <= 0;
        parked_car <= 0;
    end else begin
        //car entered
        if (car_entered) begin
            //uni car
            if (is_uni_car_entered) begin
                if (uni_is_vacated_space)
                    uni_parked_car <= uni_parked_car + 1;
            //normal car
            end else begin
                if (is_vacated_space)
                    parked_car <= parked_car + 1;
            end
        end
    end

    //car exited
    if (car_exited) begin
        //uni car
        if (is_uni_car_exited) begin
            if (uni_parked_car > 0)
                uni_parked_car <= uni_parked_car - 1;
        //normal car
        end else begin
            if (parked_car > 0)
                parked_car <= parked_car - 1;
        end
    end
end
end
end
```

سپس برای این ماژول یک تست بنچ می‌نویسیم که تمام حالات خاص ممکن را پوشش دهد:

```
module TB;

reg clk;
reg reset;
reg car_entered;
reg is_uni_car_entered;
reg car_exited;
reg is_uni_car_exited;
reg [4:0] current_hour;

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

// get instant from parking management module
parking_management PM (
    clk,
    reset,
    car_entered,
    is_uni_car_entered,
    car_exited,
    is_uni_car_exited,
    current_hour,
    uni_parked_car,
    parked_car,
    vacated_space,
    uni_vacated_space,
    uni_is_vacated_space,
    is_vacated_space
);

integer i;

//clock
always begin
    for (i = 0; i < 120; i = i + 1) begin
        #5 clk = ~clk;
    end
    current_hour = current_hour + 1;
    if (current_hour > 23)
```

```
        current_hour = 0;
end

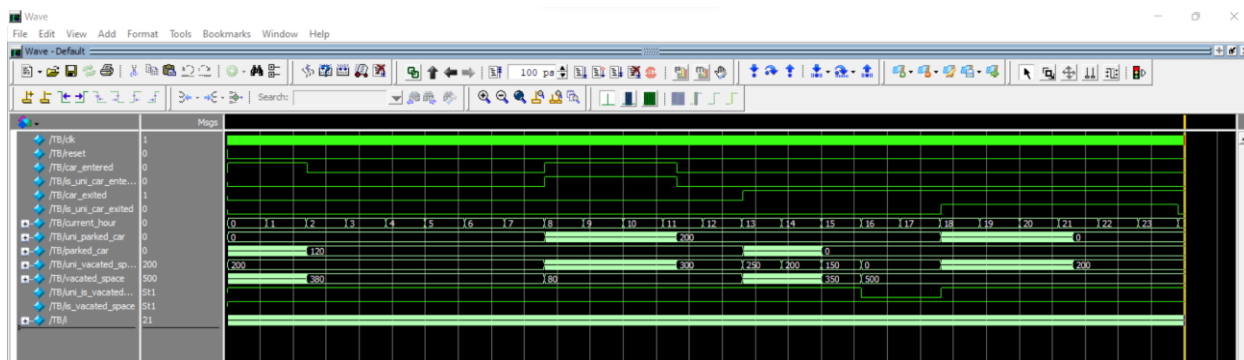
initial begin
    //initialize
    clk = 0;
    reset = 1;
    car_entered = 0;
    is_uni_car_entered = 0;
    car_exited = 0;
    is_uni_car_exited = 0;
    current_hour = 0;

    //reset
    #10 reset = 0;

    car_entered = 1;
    #1200
    car_entered = 0;
    #3600
    car_entered = 1;
    is_uni_car_entered = 1;
    #2000
    car_entered = 0;
    is_uni_car_entered = 0;
    #1000
    car_exited = 1;
    #3000
    is_uni_car_exited = 1;
    #3600
    is_uni_car_exited = 0;
    //stop simulation
    #100 $stop;
end

endmodule
```

خروجی waveform:



همانطور که مشاهده می‌کنیم، خروجی مطابق انتظار است و طراحی‌مان درست می‌باشد.