

بسم الله الرحمن الرحيم

امیر محمد رشیدی احمد آبادی

۴۰۱۱۰۵۹۶۷

طبق سوال متغیر ها را به نحوی که مشخص شده تعریف کرده ایم.

```
module parking_management (  
    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,  
    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 reg uni_is_vacated_space,  
    output reg is_vacated_space  
);
```

چند متغیر ثابت دیگر و یک رجیستر نیز تعریف می‌کنیم:

```
// Constants  
parameter TOTAL_PARKING_SPACE = 10'd700;  
parameter UNI_PARKING_CAPACITY = 10'd500;  
parameter INITIAL_FREE_SPACE = 10'd200;
```

```
// Internal registers  
reg [9:0] free_space;
```

حالا شرایط اولیه را تعریف میکنیم:

```
// Initial values  
initial begin  
    uni_parked_car = 10'd0;  
    parked_car = 10'd0;  
    uni_vacated_space = UNI_PARKING_CAPACITY;
```

```

vacated_space = INITIAL_FREE_SPACE;
uni_is_vacated_space = 1'b1;
is_vacated_space = 1'b1;
free_space = TOTAL_PARKING_SPACE - UNI_PARKING_CAPACITY;
end

```

در آخر یک بلوک always تعریف میکنیم که سه بخش دارد، اگر reset برابر با ۱ باشد پارکینگ به شرایط اولیه برمیگردد، در غیر این صورت ورود و خروج ها را ساماندهی می کند و در آخر مقادیر uni_is_vacated_space و is_vacated_space را به روزرسانی می کند.

Always block for managing parking:

```

always @(posedge clk or posedge reset) begin
    if (reset) begin
        uni_parked_car ≤ 10'd0;
        parked_car ≤ 10'd0;
        uni_vacated_space ≤ UNI_PARKING_CAPACITY;
        vacated_space ≤ INITIAL_FREE_SPACE;
        uni_is_vacated_space ≤ 1'b1;
        is_vacated_space ≤ 1'b1;
        free_space ≤ TOTAL_PARKING_SPACE - UNI_PARKING_CAPACITY;
    end else begin

```

Handle car entering:

```

    if (car_entered) begin
        if (is_car_entered) begin
            if (uni_vacated_space > 0) begin
                uni_parked_car ≤ uni_parked_car + 1;
                uni_vacated_space ≤ uni_vacated_space - 1;
            end
        end else begin
            if (vacated_space > 0) begin
                parked_car ≤ parked_car + 1;

```

```

        vacatedspace ≤ vacatedspace - 1;

    end

end

end

```

Handle car exiting:

```

if (carexited)begin

    if (isunicarexited)begin

        if (uniparkedcar > 0)begin

            uniparkedcar ≤ uniparkedcar - 1;

            univacatedspace ≤ univacatedspace + 1;

        end

    end else begin

        if (parkedcar > 0)begin

            parkedcar ≤ parkedcar - 1;

            vacatedspace ≤ vacatedspace + 1;

        end

    end

end

end

```

Update vacated space status:

```

uni_is_vacated_space ≤= (uni_vacated_space > 0);

is_vacated_space ≤= (vacated_space > 0);

end

end

```

Waveform حاصل از شبیه سازی برنامه نیز به شکل زیر است:

