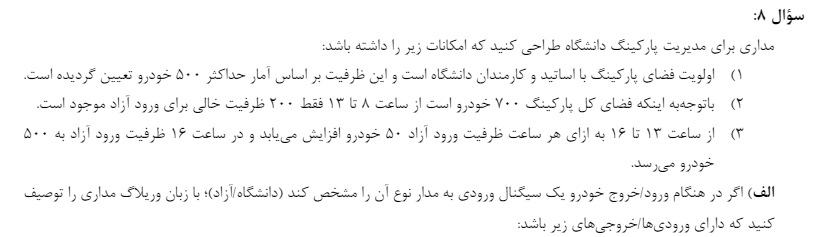
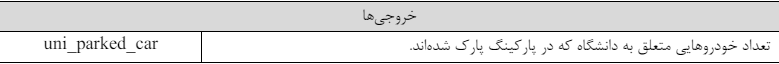
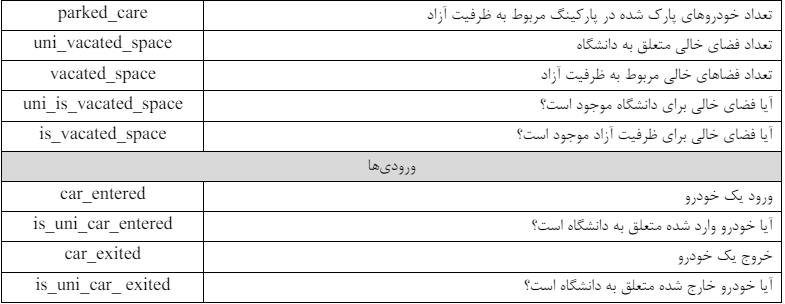
جبرانی پایانترم «به نام خدا» 05/04/1403

نام درس: طراحی سیستم‌های دیجیتال

نام و نام خانوادگی: میترا قلی پور شماره دانشجویی: 401106363



الف)

تعیین ورودی و خروجی‌های ماژول:

module parking\_controller(

    input wire clk,  // clock signal

    input wire rst,  // reset signal

    input wire car\_entered,  // car entering the parking lot

    input wire is\_uni\_car\_entered,  // is the entering car a university car?

    input wire car\_exited,  // car exiting the parking lot

    input wire is\_uni\_car\_exited,  // is the exiting car a university car?

    output reg [8:0] uni\_parked\_car,  // number of university cars parked

    output reg [8:0] f\_parked\_car,  // number of free cars parked

    output reg [8:0] uni\_vacated\_space,  // vacant university spaces

    output reg [8:0] f\_vacated\_space,  // vacant free spaces

    output reg is\_uni\_vacated\_space,  // is there a vacant university space?

    output reg is\_vacated\_space  // is there a vacant free space?

);

تنظیم سایر متغیر‌های مورد نیاز:

    // parameters

    parameter TOTAL\_UNI\_SPACES = 500;

    parameter TOTAL\_FREE\_SPACES\_MORNING = 200;

    parameter TOTAL\_SPACES = 700;

    parameter CLOCKS\_PER\_HOUR = 100;

    reg [31:0] clock\_counter;  // counter to count clock cycles

    reg [4:0] hour;  // current hour (0-23)

    reg [8:0] free\_capacity;

    wire [8:0] uni\_capacity;

assign uni\_capacity = TOTAL\_SPACES - free\_capacity;

تنظیم ساعت بر حسب کلاک برنامه:

این قسمت به این صورت کار می‌کند که ابتدا از ساعت 8 شروع می‌شود برنامه و تعداد کلاک‌های ورودی را شمارش می‌کند هرگاه با CLOCKS\_PER\_HOUR – 1 یک ساعت گذشته است، در نتیجه hour را یک عدد افزایش می‌دهد و از آنجا که یک روز 24 ساعت است هر 24 ساعت hour را صفر می‌کند.

    // increment hour based on clock cycles

    always @(posedge clk or posedge rst) begin

        if (rst) begin

            clock\_counter <= 0;

            hour <= 8;  // starting hour is 8 AM

        end else begin

            if (clock\_counter == CLOCKS\_PER\_HOUR - 1) begin

                clock\_counter <= 0;

                if (hour < 23) begin

                    hour <= hour + 1;

                end else begin

                    hour <= 0;  // reset to midnight after 23:00

                end

            end else begin

                clock\_counter <= clock\_counter + 1;

            end

        end

    end

تعیین ظرفیت بخش‎‌های مختلف:

با تنظیم شدن free-capacity، uni-capacity نیز براساس این خط که در قبل آمده است نیز تعیین می‌شود.

assign uni\_capacity = TOTAL\_SPACES - free\_capacity;

// calculate free capacity based on the current hour

    always @(hour or rst) begin

        if (rst) begin

            free\_capacity <= TOTAL\_FREE\_SPACES\_MORNING;

        end else begin

            case (hour)

                5'd8: free\_capacity <= 200;

                5'd9: free\_capacity <= 200;

                5'd10: free\_capacity <= 200;

                5'd11: free\_capacity <= 200;

                5'd12: free\_capacity <= 200;

                5'd13: free\_capacity <= 250;

                5'd14: free\_capacity <= 300;

                5'd15: free\_capacity <= 350;

                5'd16: free\_capacity <= 500;

                default: free\_capacity <= 500;

            endcase

        end

    end

تنظیم متغیر‌ها براساس ورودی ‌ها

در صورتی که ماشین وارد شود، اگر همچنان در بخشش فضای آزاد باشد، می‌تواند پارک کند. همچنین هنگامی که خارج می‌شود باید تعداد ماشین‌های پارک شده مثبت باشد، در غیر اینصورت خارج شدن بی‌معنی است.

    // update parking counts based on car entry and exit

    always @(posedge rst or posedge car\_entered or posedge car\_exited) begin

        if (rst) begin

            uni\_parked\_car <= 0;

            f\_parked\_car <= 0;

        end else begin

            // car entry

            if (car\_entered) begin

                if (is\_uni\_car\_entered) begin

                    if (is\_uni\_vacated\_space != 0) begin

                        uni\_parked\_car <= uni\_parked\_car + 1;

                    end

                end else begin

                    if (is\_vacated\_space != 0) begin

                        f\_parked\_car <= f\_parked\_car + 1;

                    end

                end

            end

            // car exit

            if (car\_exited) begin

                if (is\_uni\_car\_exited) begin

                    if (uni\_parked\_car > 0) begin

                        uni\_parked\_car <= uni\_parked\_car - 1;

                    end

                end else begin

                    if (f\_parked\_car > 0) begin

                        f\_parked\_car <= f\_parked\_car - 1;

                    end

                end

            end

        end

    end

تنظیم خروجی‌ها (آپدیت نگه داشتن آنها پس از هر تغییر):

    // calculate vacant spaces and availability

    always @\* begin

        if(uni\_parked\_car > uni\_capacity) begin

            uni\_vacated\_space = 0;

        end

        else if(TOTAL\_SPACES - f\_parked\_car >=  uni\_capacity) begin

            uni\_vacated\_space = uni\_capacity - uni\_parked\_car;

        end

        else begin

            uni\_vacated\_space = TOTAL\_SPACES - uni\_parked\_car - f\_parked\_car;

        end

        if(f\_parked\_car > free\_capacity) begin

            f\_vacated\_space = 0;

        end

        else if(TOTAL\_SPACES - uni\_parked\_car >=  free\_capacity) begin

            f\_vacated\_space = free\_capacity - f\_parked\_car;

        end

        else begin

            f\_vacated\_space = TOTAL\_SPACES - uni\_parked\_car - f\_parked\_car;

        end

        is\_uni\_vacated\_space = (uni\_vacated\_space > 0);

        is\_vacated\_space = (f\_vacated\_space > 0);

    end

endmodule

اینجا 3 شرط چک شده است که برای ظرفیت آزاد آن را توضیح می‌دهم، برای ظرفیت مربوط به دانشگاه مشابه همان است:

if(f\_parked\_car > free\_capacity) begin

            f\_vacated\_space = 0;

        end

این شرط برای زمانی است که ماشین ‌های پارک شده در بخش آزاد هنگامی که وارد شدند با ظرفیت تخصیص داده شده مطابقت داشتند ولی با گذشت زمان ظرفیت تخصیص داده شده کمتر شده درنتیجه تعداد ماشین‌های پارک شده بیشتر از ظرفیت کنونی(نه قبلی) است.

else begin

            f\_vacated\_space = TOTAL\_SPACES - uni\_parked\_car - f\_parked\_car;

end

حالت قبلی اگر برای ماشین‌های دانشگاه اتفاق بیافتد پس مقداری از ظرفیت بخش آزاد کم می‌شود، با توجه به اینکه حداکثر 700 تا ماشین می‌توانند وارد شوند، باید این موضوع در نظر گرفته شود که به اندازه مقدار باقی مانده ماشین می‌تواند وارد شود.

else if(TOTAL\_SPACES - uni\_parked\_car >=  free\_capacity) begin

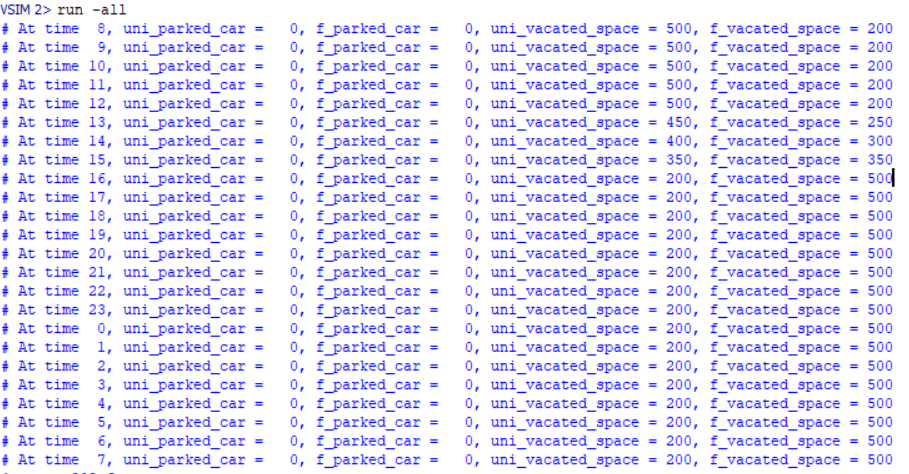
            f\_vacated\_space = free\_capacity - f\_parked\_car;

end

در حالت عادی از این شرط استفاده می‌کنیم.

تست:

برای تست حالت‌های زیر در نظر گرفته شده است:

بررسی ظرفیت ها بدون ورود و خروج ماشین‌ها:

ورود بیش از ظرفیت ماشین‌های آزاد:



        // enter 202 free cars

        for (i = 0; i < 202; i = i + 1) begin

            car\_entered = 1;

            is\_uni\_car\_entered = 0;

            #2;

            car\_entered = 0;

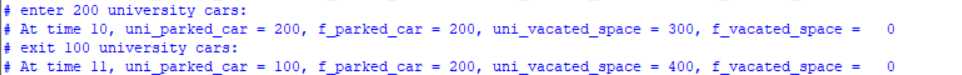
            is\_uni\_car\_entered = 0;

            #2;

        end

        $display("enter 202 free cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

ورود و خروجی معمولی ماشین‌های دانشگاهی:

        // enter 200 university cars

        for (i = 0; i < 200; i = i + 1) begin

            car\_entered = 1;

            is\_uni\_car\_entered = 1;

            #5;

            car\_entered = 0;

            is\_uni\_car\_entered = 0;

            #5;

        end

        $display("enter 200 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

        // exit 100 university cars

        for (i = 0; i < 100; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited= 1;

            #5;

            car\_exited = 0;

            is\_uni\_car\_exited = 0;

            #5;

        end

        $display("exit 100 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

خروج و ورود پشت سر هم ماشین دانشگاهی:

        // enter and exit 50 university cars

        for (i = 0; i < 50; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited= 1;

            #1 car\_exited = 0;

            is\_uni\_car\_exited = 0;

            #1 car\_entered = 1;

            is\_uni\_car\_entered = 1;

            #1 car\_entered = 0;

            is\_uni\_car\_entered = 0;

            #1;

        end

        $display("enter and exit 50 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

ورود بیش از حد مجاز ماشین ‌های دانشگاهی:



// enter 402 university cars

        for (i = 0; i < 402; i = i + 1) begin

            car\_entered = 1;

            is\_uni\_car\_entered = 1;

            #1;

            car\_entered = 0;

            is\_uni\_car\_entered = 0;

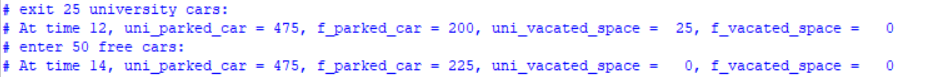
            #1;

        end

        $display("enter 402 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

نشان دادن افزایش ظرفیت با تغییر ساعت:

نکته قابل توجه در این تست کیس این است که قسمت آزاد تا جایی که توانست ماشین گرفت ولی ظرفیت کل یعنی 700 هم در نظر گرفته شده است. همچنین ماشین دانشگاهی مازاد بر ظرفیت در پارکینگ هستند، پس دیگر نمی‌توانند وارد شوند.

        // exit 25 university cars

        for (i = 0; i < 25; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited = 1;

            #1;

            car\_exited = 0;

            is\_uni\_car\_exited= 0;

            #1;

        end

        $display("exit 25 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

        #1000

        // enter 50 free cars

        for (i = 0; i < 50; i = i + 1) begin

            car\_entered = 1;

            is\_uni\_car\_entered = 0;

            #5;

            car\_entered = 0;

            is\_uni\_car\_entered = 0;

            #5;

        end

        $display("enter 50 free cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

نشان دادن ظرفیت واقعی آزاد پس خالی شدن بخش دانشگاهی:



        // exit 130 university cars

        for (i = 0; i < 130; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited = 1;

            #1;

            car\_exited = 0;

            is\_uni\_car\_exited= 0;

            #1;

        end

        $display("exit 130 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

خروج و ورود همزمان از بخش آزاد:



        // enter and exit 50 free cars

        for (i = 0; i < 50; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited= 0;

            #1 car\_exited = 0;

            is\_uni\_car\_exited = 0;

            #1 car\_entered = 1;

            is\_uni\_car\_entered = 0;

            #1 car\_entered = 0;

            is\_uni\_car\_entered = 0;

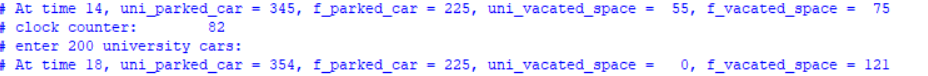
            #1;

        end

        $display("enter and exit 50 free cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

تست لحظه تغییر ساعت و ورود خارج ظرفیت در ساعت جدید:

دقت کنید در اینجا clock\_counter 82 تا شمرده یعنی 18 تا کلاک تا ساعت بعدی فاصله هست. طبق کد هر ماشین طی دوتا کلاک میرود پس ما انتظار داریم بعد 9 کلاک دیگر ماشین‌ها وارد نشوند:

$display("enter and exit 50 free cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

        $display("clock counter:%d",dut.clock\_counter);

        // enter 200 university cars

        for (i = 0; i < 200; i = i + 1) begin

            car\_entered = 1;

            is\_uni\_car\_entered = 1;

            #10;

            car\_entered = 0;

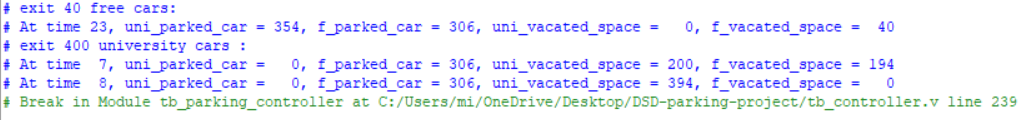
            is\_uni\_car\_entered = 0;

            #10;

        end

        $display("enter 200 university cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

خروج عادی ماشین های آزاد و خروج بیش از حد معمول تغییر ظرفیت درست از 7 به 8:

// exit 40 free cars

        for (i = 0; i < 40; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited = 0;

            #10;

            car\_exited = 0;

            is\_uni\_car\_exited= 0;

            #10;

        end

        $display("exit 40 free cars: \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

        // exit 400 university cars

        for (i = 0; i < 400; i = i + 1) begin

            car\_exited = 1;

            is\_uni\_car\_exited = 1;

            #10;

            car\_exited = 0;

            is\_uni\_car\_exited= 0;

            #10;

        end

        $display("exit 400 university cars : \nAt time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

        #1000

        $display("At time %d, uni\_parked\_car = %d, f\_parked\_car = %d, uni\_vacated\_space = %d, f\_vacated\_space = %d",

                    dut.hour, uni\_parked\_car, f\_parked\_car, uni\_vacated\_space, f\_vacated\_space);

        #1000 $stop;

بنابراین حالت‌های بدون ورود و خروج، تغییر ظرفیت پس از 12 و از 7 به 8، ورود و خروج در لحظه تغییرات ظرفیت، ورود و خروج معمولی ، خارج ظرفیت ، همزمان و 700 بودن مجموع فضاها و نامنفی بودن آن ها در هر کدام از حالت ها را بررسی کرده‌ایم و نتایج درستی دریافت کردیم.

ب) سنتز کردن FPGA:

از نرم افزار کوارتوس دیوایس MAX V استفاده کردم.



مدار خروجی در فایل export-circuit.pdf در دسترس است.

مقادیر فرکانس در گزارش سنتز نشان‌دهنده حداکثر فرکانس‌های کلاک هستند که بخش‌های مختلف طراحی FPGA می‌تواند به طور قابل اعتماد کار کنند.

برای car\_enteredاین فرکانس بالاترین نرخی است که منطق مرتبط با سیگنال car\_entered می‌تواند به طور قابل اعتماد کار کند. این بدان معناست که مسیر داده و منطق ترکیبی شامل سیگنال car\_entered دارای یک مسیر بحرانی با تأخیر است که حداکثر فرکانس کلاک را به این فرکانس محدود می‌کند. مقدار محدود شده Fmax نیز این محدودیت را تأیید می‌کند.

برای clk این مقدار بالاترین فرکانس کلاک است که کل طراحی می‌تواند در آن کار کند. این مقدار بازتابی از تأخیر مسیر بحرانی کل طراحی است که شامل تمام عناصر درایو شده توسط سیگنال کلاک اصلی (clk) است. مقدار محدود شده Fmax نیز با این مقدار مطابقت دارد، به این معنی که بعد از جای‌گذاری و مسیریابی، طراحی هنوز می‌تواند در این فرکانس بدون نقض زمان‌بندی کار کند. حداکثر فرکانس توسط طولانی‌ترین تأخیر در مسیر بحرانی مدار تعیین می‌شود. بخش‌های مختلف طراحی دارای مسیرهای بحرانی مختلفی هستند که منجر به فرکانس‌های حداکثر متفاوت می‌شود. فرکانس‌های مختلف، تنوع در محدودیت‌های زمان‌بندی را در بخش‌های مختلف طراحی FPGA نشان می‌دهد. منطق سیگنال car\_entered به دلیل مسیرهای بحرانی طولانی‌تر، عملیات خود را به فرکانس کمتری محدود می‌کند، در حالی که طراحی کلی درایو شده توسط کلاک اصلی (clk) می‌تواند در فرکانس بالاتری کار کند. این بینش‌ها به شناسایی و بهینه‌سازی مسیرهای بحرانی برای بهبود عملکرد طراحی کمک می‌کند.