گزارش سوال هشتم میانترم

محمد میرزادی

401110104

استاد فصحتی

پارکینگ به این گونه کار میکند که در ساعات بامداد تا 8 صبح تعطیل است و ظرفیت صفر برای هر دو گروه دارد و از ساعت 8 تعداد 500 ظرفیت برای ماشین های دانشگاه و 200 ظرفیت آزاد وجود دارد.

از ساعت 13 تا 16 به ازای هر ساعت ظرفیت ماشین های دانشگاه پنجاه تا زیاد و ظرفیت ماشین های آزاد 50 تا کم میشود تا اینکه یک دفعه از 16 به بعد ظرفیت ماشین های آزاد یکباره به 500 ظرفیت آزاد و 200 ظرفیت دانشگاه میرسد.

و تا ساعت شب 8 پارکینگ باز است و از آن به بعد بسته میشود و ظرفیت صفر میشود.

در کل زمانی که پارکینگ باز است اگر یک ماشین از طرف دانشگاه بخواهد وارد شود در صورتی که ظرفیت دانشگاه پر نشده باشد وارد ظرفیت دانشگاه میشود اگر پر شده باشد جزو ظرفیت آزاد محصوب میشود و وارد پارکینگ آزاد میشود.

در زمان هایی که ظرفیت کل ماشین های دانشگاه و آزاد دچار تغییر میشود اگرتعداد ماشین هایی که در ظرفیت دانشگاه هستند از ظرفیت دانشگاه بیشتر شد به اندازه ای که از ظرفیت بیشتر شده به ظرفیت آزاد منتقل میشود و بیرون ریخته نمیشود.

**خلاصه پیاده سازی**

به صورت کلی کد ما از دو فایل parking.v و testbench.v تشکیل شده است.

که در فایل parking.v با دریافت اطلاع از ورود/خروج ماشین جدید و نوع ظرفیتی که به آن باید اخصاص داده شده و اطلاع از زمان در سه بلاک always هم درخواست ها را هندل میکنیم هم بر اساس زمان تغییرات ظرفیت ها را تغییر میدهیم.

1. بلاک always اول بر اساس تغییرات timer کار میکند و اگر تایمر دقیقا ساعت های 13 و 14 و 15 باشد مقدار ظرفیت آزاد 50 تا زیاد میشود.
2. بلاک always دوم بر اساس لبه ی بالا رونده ی car\_entered کار میکند و درخواست ها را بر اساس ظرفیت هندل میکند.
3. بلاک آخر هم exit ها را هندل میکند.

در فایل testbench.v ابتدا کلاک تعیین میکنیم و سپس بر اساس آن تایمر را بالا میبریم. **هر کلاک را معادل یک ثانیه قرار میدهیم و زمان از 12 بامداد آغاز میشود.**

در یک بلاک initial با جلو بردن زمان و استفاده از for به تعداد مشخص ماشین را وارد میکنیم.

**بررسی کد**

در اینجا کد را بررسی میکنیم ابتدا کد parking.v بررسی میکنیم

module parking(input car\_entered,is\_uni\_car\_enterd,car\_exited,is\_uni\_car\_exited,clk,input integer timer,

    output integer uni\_parked\_car,parked\_car,output wire [31:0] uni\_vacated\_space,vacated\_space,output uni\_is\_vacated\_space,is\_vacated\_space);

    integer uni\_car\_space; //maximum cars that can be parked as university cars

    integer free\_car\_space; //maximum cars that can be parked as free cars (not university cars)

    initial begin

        uni\_parked\_car = 0;

        parked\_car = 0;

        uni\_car\_space = 0;

        free\_car\_space = 0;

    end

*ابتدا ورودی و خروجی ماژول را مشخص میکنیم*

*همه چیز مطابق ورودی خروجی مشخص شده در سوال است به جز اینکه دو ورودی clk و time اضافه شده اند.*

***Clk : کلاک را داریم که طول دوره ی آن 10 واحد زمان است.***

***Timer: تایمر در فایل تست ما تغییر میکند و به ماژول ما متصل میشود.***

*دو integer را تعریف میکنیم که uni\_car\_space و free\_car\_space هستند که به ترتیب نشان دهنده ی ماکسیمم تعداد ماشینی هستند که در ظرفیت دانشگاهی و آزاد قرار میگیرد.*

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

    assign vacated\_space = free\_car\_space - parked\_car;

    assign uni\_is\_vacated\_space = (uni\_car\_space - uni\_parked\_car) > 0; //check if there is a vacated space

    assign is\_vacated\_space = (free\_car\_space - parked\_car) > 0;

*در اینجا خروجی های uni\_vacated\_space,vacated\_space, uni\_is\_vacated\_space, is\_vacated\_space را براساس uni\_car\_space و uni\_parked\_car و free\_car\_space و car\_parked معین میکنیم.*

*اکنون بلاک اول always را آغاز میکنیم.*

    always @(posedge clk) begin

        if(timer<8\*60||timer>=20\*60) begin // if clock is not between 8 Am to 8 pm the parking is closed to everyone

            uni\_car\_space = 0;

            free\_car\_space = 0;

        end

اگر زمان در ساعت تعطیلی پارکینگ بود ظرفیت را صفر کن

else if( timer>=8\*60 && timer<13\*60 ) begin //if we are in official hours

            if(timer == 8\*60) begin //reset variables after opening parking

                uni\_parked\_car = 0;

                parked\_car = 0;

            end

            uni\_car\_space = 500;

            free\_car\_space = 200;

        end

اگر در ساعت کار دانشگاه بودیم ظرفیت را روی 500 و 200 فیکس میکند.

else if ( timer >= 13\*60 && timer < 16\*60 ) begin

            if( timer % 60 == 0) begin

                uni\_car\_space = uni\_car\_space - 50;

                free\_car\_space = free\_car\_space + 50;

        end

        end

در بازه ی زمانی 13 تا 16 در هر ساعت 50 تا ظرفیت دانشگاه کم میشود و به آزاد اضافه میشود.

else if ( timer>= 16\*60 && timer <20\*60 ) begin

            uni\_car\_space = 200;

            free\_car\_space = 500;

        end

        if (uni\_parked\_car>uni\_car\_space) begin //if university parking is full move the cars into free cars parking

            parked\_car = parked\_car+(uni\_parked\_car-uni\_car\_space);

            uni\_parked\_car = uni\_car\_space;

        end

        if (parked\_car>free\_car\_space) begin // if parking is full pick them out

            parked\_car = free\_car\_space;

        end

    end

در اینجا ابتدا چک میکنیم اگر از ساعت 16 گذشته باشد ظرفیت روی 200 و 500 فیکس میشود.

در انتهای بلاک هم چک میکنیم که اگر با تغییر ظرفیت ها مقدار ماشین های دانشگاهی سرریز کرد اضافه ها را به ظرفیت آزاد منتقل میکنیم واگر ظرفیت آزاد سریز کرد ظرفیت با ظرفیت ماکسیمم فیکس میشود.

    always @(posedge car\_entered) begin

        #0

        if(is\_uni\_car\_enterd && (uni\_car\_space-uni\_parked\_car)>0) begin //if the car is university car and it has capacity

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

        end

        else if (free\_car\_space-parked\_car>0) begin

            parked\_car = parked\_car + 1;

        end

    end

در بلاک دوم با لبه ی بالا رونده ی car\_entered میکند اگر از صفر یک شود یعنی ماشین جدید وارد شده است.

در صورتی که ماشین مورد نظر دانشگاهی بود و ظرفیت ماشین های دانشگاهی پر نبود به ظرفیت دانشگاهی و در غیر این صورت اگر ظرفیت آزاد خالی داشت به ظرفیت آزاد میرود در غیر این صورت از ورودش ممانعت به عمل می آید.

    always @(posedge car\_exited) begin

        #0

        if(is\_uni\_car\_exited && uni\_parked\_car>0) begin

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

        end

        else if(parked\_car>0) begin

            parked\_car = parked\_car - 1;

        end

    end

endmodule;

در این بلاک که با لبه ی بالا رونده ی کلاک کار میکند ماشین ها را خارج میکند.

اکنون به ماژول testbench.v میرویم

module testbench();

    reg clk;

    integer timer;

    initial begin

        timer = 0;

        clk = 0;

    end

    always #5 clk = ~clk;

    always @(posedge clk) timer = (timer+1)%(24\*60);

    //inputs

    reg car\_entered,is\_uni\_car\_enterd,car\_exited,is\_uni\_car\_exited;

    //outputs

    wire uni\_is\_vacated\_space,is\_vacated\_space;

    wire [31:0] uni\_parked\_car,parked\_car;

    wire [31:0] uni\_vacated\_space,vacated\_space;

    //get instance

    parking parking1(.car\_entered(car\_entered),.is\_uni\_car\_enterd(is\_uni\_car\_enterd),

    .car\_exited(car\_exited),.is\_uni\_car\_exited(is\_uni\_car\_exited),.clk(clk),.timer(timer),

        .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 اینجا ابتدا کلاک و تایمر را راه می اندازیم و ورودی های تابع را از نوع رجیستر و خروجی ها را سیم تعریف میکنیم و از parking اینستنس میسازیم.*

*در ادامه یک بلاک اینیشیال تعریف میکنیم*

*هر تست بنچ به فرم زیر نوشته میشود.*

        $display("50 ordinay cars want to enter");

        for(i = 0;i<50;i= i+1) begin

            is\_uni\_car\_enterd = 0;

            #0 car\_entered = 1;

            #0 car\_entered = 0;

        end

        #0

        $display("timer = %d:%d    uni\_parked\_car = %d  parked\_car = %d  uni\_vacated\_space = %d  vacated\_space = %d  uni\_is\_vacated\_space = %d  is\_vacated\_space = %d",

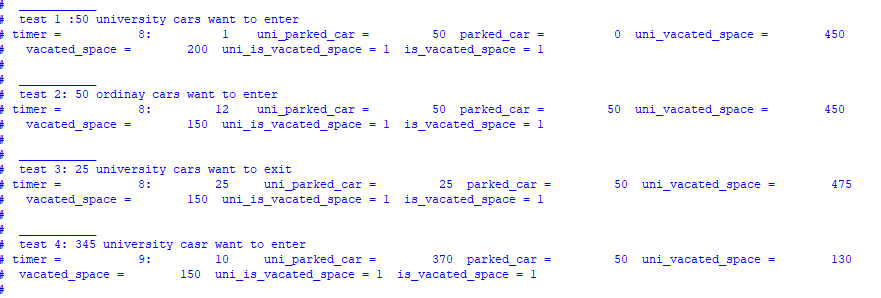
        timer/60,timer%60,uni\_parked\_car,parked\_car,uni\_vacated\_space,vacated\_space,uni\_is\_vacated\_space,is\_vacated\_space);

        #130

*در اینجا ابتدا تعداد ماشین های ورودی ها را اعلام میکنیم سپس با یک حلقه 50 بار car\_entered را تریگر میکنیم تا 50 ماشین وارد شود و بعد با اعلام ساعت و دقیقه ما و متغیر های خروجی ما را از روند کار برنامه مطلع میسازد.*

***اکنون سراغ تست کیس ها میرویم***

**تست کیس ها**

****

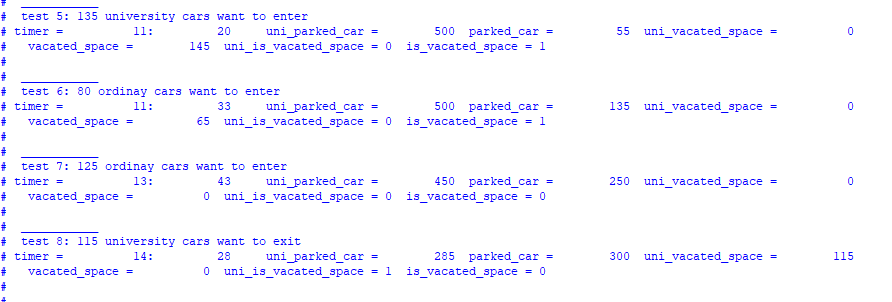
در ساعت 00:01 بامداد ظرفیت همه صفر است و هیچ فضای خالی ای وجود ندارد.

تست 1: در ساعت 8:01 50 تا ماشین دانشگاهی وارد پارکینگ میشوند و تعداد ماشین های دانشگاه برابر 50 فضای خالی آن 450 و فضای خالی آزاد 200 میماند و هر دو ظرفیت جا دارند.

تست 2: در ساعت 8:12 50 ماشین معمولی وارد ظرفیت آزاد میشوند و فضای خالی آزاد برابر 150 میشود.

تست 3: در ساعت 8:25 25 ماشین دانشگاهی خارج میشوند تعداد ماشین های موجود در ظرفیت دانشگاهی 25 میشود.

تست 4: در ساعت 9:10 دقیقه 345 ماشین دانشگاهی وارد میشوند و تعداد ماشین های موجود در ظرفیت دانشگاهی 370 میشود و ظرفیت خالی برابر 130 میشود.



تست 5: در ساعت 11:20 تعداد 135 ماشین دانشگاهی وارد میشوند اما چون قبل آن ظرفیت خالی ماشین های دانشگاهی 130 بود 5 ماشین دانشگاهی از ورود به ظرفیت دانشگاهی باز میمانند و به ظرفیت آزاد افزوده میشوند در این صورت ظرفیت آزاد به 55 ظرفیت دانشگاهی به 500 میرسد که یعنی پر است و دیگر نمیپذیرد.

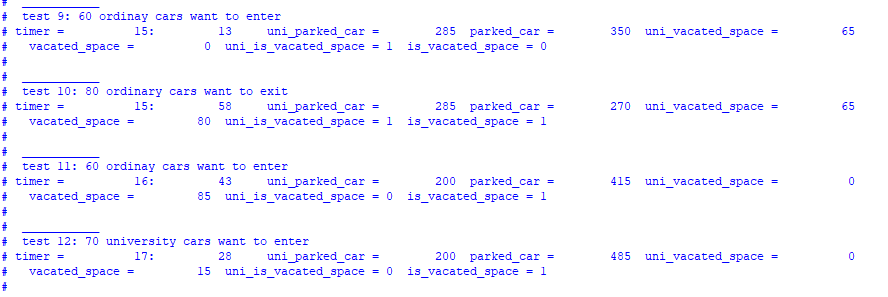
تست 6: در ساعت 11:33 تعداد 80 ماشین معمولی اضافه میشوند و ماشین های موجود در ظرفیت آزاد به 135 میرسد.

در ساعت 13:00 ظرفیت ماشین های دانشگاهی 50 تا کم و به ماشین های آزاد اضافه میشود و طبق آن تعداد ماشین های موجود در ظرفیت دانشگاهی به 450 کاهش و به ظرفیت آزاد افزوده میشود و برابر 185 میشود.

تست 7: در ساعت 13:43 125 ماشین معمولی تقاضای ورود به پارکینگ را میدهند اما چون ظرفیت محدود است تنها 65 تا از آن ها تایید میشوند. و ظرفیت آزاد پر میشود یعنی 250 تا.

در ساعت 14 ظرفیت ماشین های دانشگاهی 50 تا کم و به ماشین های آزاد افزوده میشود یعنی 400 ماشین در ظرفیت دانشگاهی خواهند بود و ظرفیت ماشین های موجود در آزاد برابر 300 میشود.

تست 8: در ساعت 14:28 115 ماشین دانشگاهی خارج میشوند و ماشین های موجود در ظرفیت دانشگاهی برابر 285 میشود و ظرفیت خالی آن برابر 115 میشود. ظرفیت آزاد هم پر باقی میماند.



در ساعت 15 ظرفیت آزاد 50 تا زیاد میشود و ظرفیت دانشگاهی 50 تا کم میشود اما چون جا دارد تعداد ماشینی به ظرفیت آزاد منتقل نمیشود و تعداد ماشین های موجود در ظرفیت دانشگاهی همان 285 باقی میماند وفضای خالی فضای آزاد 50 تا زیاد میشود.

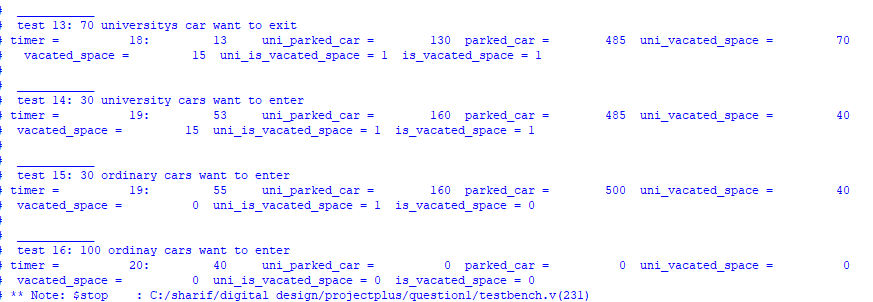
تست 9: در ساعت 15:13 60 تا ماشین آزاد وارد میشود اما چون 50 تا ظرفیت آزاد داریم 10 تای آن ها راه داده نمیشوند و ظرفیت ماشین های آزاد پر میشود.

تست 10: در ساعت 15:58 80 تا ماشین عادی خارج میشوند ظرفیت خالی ماشین های آزاد برابر 80 میشود.

در ساعت 16 یک دفعه ای ظرفیت ماشین آزاد برابر 500 و ماشین دانشگاهی برابر 200 میشود و چون تعداد ماشین های پارک شده در ظرفیت دانشگاهی برابر 285 بود تعداد 85 تا از آن ها وارد ظرفیت آزاد میشوند و تعداد ماشین های پارک شده در ظرفیت آزاد برابر 355 میشود. ظرفیت ماشین های دانشگاهی هم پر و برابر 200 میماند.

تست 11: در ساعت 16:43 60 ماشین آزاد وارد میشوند و ظرفیت ماشین آزاد برابر 415 میشود. و ظرفیت خالی آزاد برابر 58 میشود.

تست 12: در ساعت 17:28 تعداد 70 ماشین معمولی وارد میشود و ظرفیت خالی آزاد 15 وظرفیت خالی دانشگاهی صفر میماند.



تست 13:در ساعت 18:13 70 تا ماشین دانشگاهی خارج میشوند و تعداد ماشین های موجود در ظرفیت دانشگاهی برابر 130 میشود با 70 ظرفیت خالی برای دانشگاهی و 15 ظرفیت خالی برای آزاد.

تست 14: در ساعت 19:53 30 تا ماشین دانشگاهی وارد میشوند و ظرفیت ماشین های دانشگاهی برابر 160 میشود.

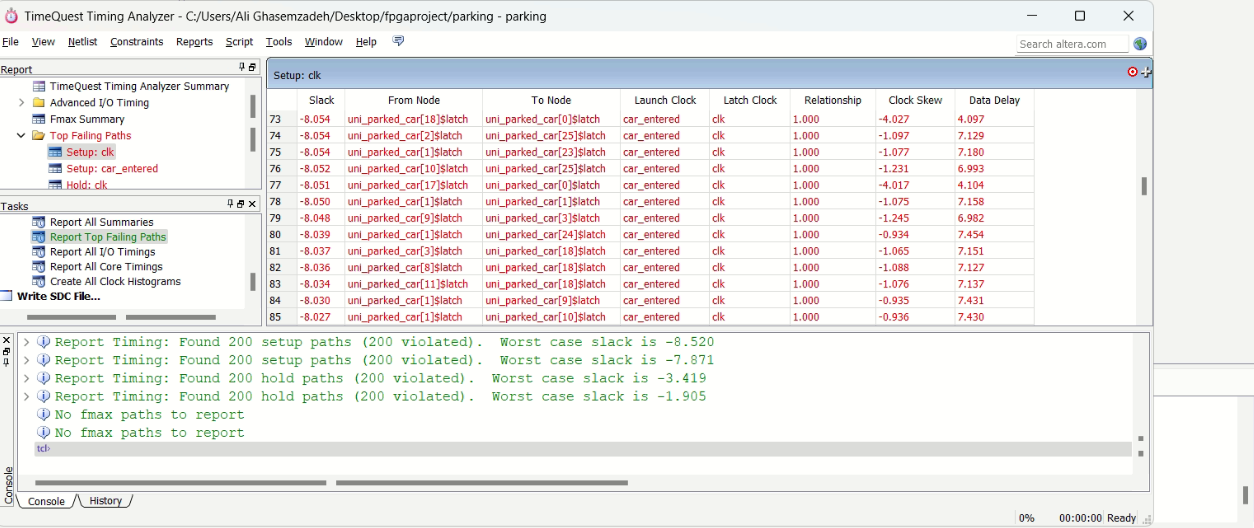
تست 15: در ساعت 19:55 30 ماشین عادی وارد میشوند و چون 15 تا ظرفیت آزاد باقی مانده بود ماشین های آزاد 500 تا میشوند.

تست 16: در ساعت 20:40 100 ماشین عادی درخواست ورود میدهند اما چون بعد 20 پارکینگ تعطیل است درخواستشان تایید نمیشود.

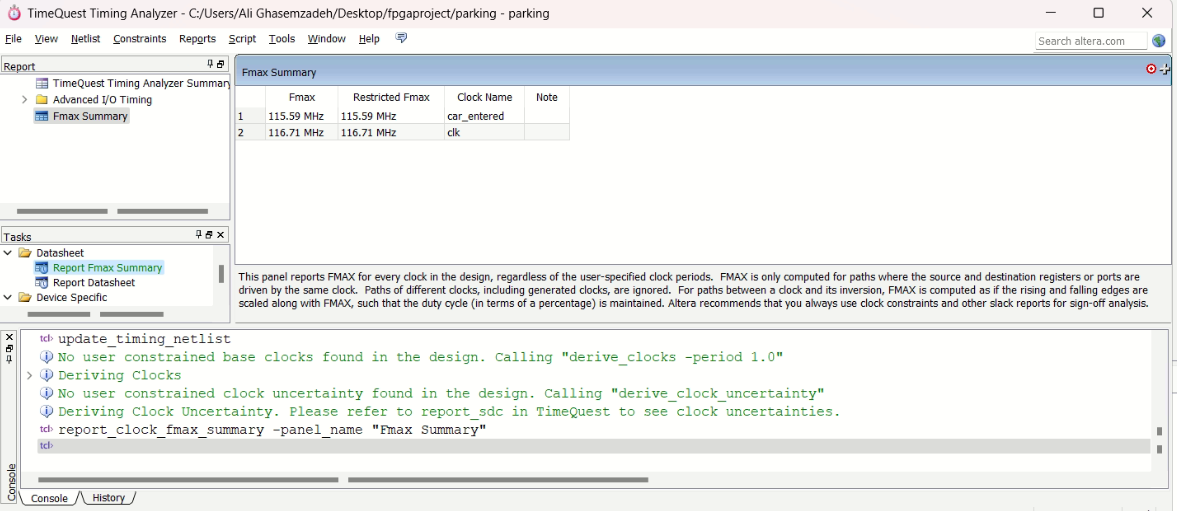
حال به حل بخش ب سوال میپردازیم

برای سنتز مدار به FPGA ابتدا یک پروژه جدید میسازم و کد parking.v را در آنجا کپی میکنم **(که در فولدرmiscellaneous گیت هاب قرار گرفته است)** با این تفاوت که تمام integer ها را به reg[31:0] تبدیل میکنیم همچنین سه بلاک always را به یک بلاک حساس به clk و enter\_car و exit\_car وارد میکنم

بعد از سنتز با استفاده از گزینه ی report\_top\_failing\_path مقدار تاخیر تمام مسیر ها را مشاهده میکنیم که با بررسی های بیشتر ماکسیمم تاخیر7.454 است.



و با گزینه ی report Fmax summery فراکانس ماکسیمم برای کلاک و دکمه ی car\_enter به دست می آوریم.



فرکانس ماکسیمم برابر کلاک برابر 116.71 مگاهرتز است.