

گزارش سوال هشتم میانترم

محمد میرزادی

401110104

استاد فصحتی

فهرست

| | |
|------------------|--------------|
| خلاصه پیاده سازی | صفحه 3 |
| بررسی کامل کد | صفحه 4 تا 8 |
| بررسی تست کیس ها | صفحه 9 تا 12 |
| بررسی بخش ب سوال | صفحه 13 |

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 محسوب میشود و وارد پارکینگ آزاد میشود.

در زمان هایی که ظرفیت کل ماشین های دانشگاه و آزاد دچار تغییر میشود اگر تعداد 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 اختصاص 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 میدهم.

1- بلاک always اول بر اساس تغییرات timer کار میکند و اگر تایمر دقیقاً 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 مائول 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 مائول 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,` 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
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 و اگر ظرفیت آزاد سرریز 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
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 میروید 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
end

```

```

        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(ti
mer),
    .uni_parked_car(uni_parked_car),.parked_car(parked_car),.uni_vacated_space(uni_va
cated_space),
    .vacated_space(vacated_space),.uni_is_vacated_space(uni_is_vacated_space),.is_vac
ated_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 ordinary 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_i
s_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

```
# -----
# test 1 :50 university cars want to enter
# timer =      8:      1    uni_parked_car =      50 parked_car =      0 uni_vacated_space =      450
# vacated_space =      200 uni_is_vacated_space = 1 is_vacated_space = 1
#
# -----
# test 2: 50 ordinary cars want to enter
# timer =      8:      12    uni_parked_car =      50 parked_car =      50 uni_vacated_space =      450
# vacated_space =      150 uni_is_vacated_space = 1 is_vacated_space = 1
#
# -----
# test 3: 25 university cars want to exit
# timer =      8:      25    uni_parked_car =      25 parked_car =      50 uni_vacated_space =      475
# vacated_space =      150 uni_is_vacated_space = 1 is_vacated_space = 1
#
# -----
# test 4: 345 university cars want to enter
# timer =      9:      10    uni_parked_car =      370 parked_car =      50 uni_vacated_space =      130
# vacated_space =      150 uni_is_vacated_space = 1 is_vacated_space = 1
#
```

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.

```
#
# test 5: 135 university cars want to enter
# timer = 11: 20 uni_parked_car = 500 parked_car = 55 uni_vacated_space = 0
# vacated_space = 145 uni_is_vacated_space = 0 is_vacated_space = 1
#
#
# test 6: 80 ordinary cars want to enter
# timer = 11: 33 uni_parked_car = 500 parked_car = 135 uni_vacated_space = 0
# vacated_space = 65 uni_is_vacated_space = 0 is_vacated_space = 1
#
#
# test 7: 125 ordinary cars want to enter
# timer = 13: 43 uni_parked_car = 450 parked_car = 250 uni_vacated_space = 0
# vacated_space = 0 uni_is_vacated_space = 0 is_vacated_space = 0
#
#
# test 8: 115 university cars want to exit
# timer = 14: 28 uni_parked_car = 285 parked_car = 300 uni_vacated_space = 115
# vacated_space = 0 uni_is_vacated_space = 1 is_vacated_space = 0
#
#
#
```

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.

```
#
# -----
# test 9: 60 ordinary cars want to enter
# timer = 15: 13 uni_parked_car = 285 parked_car = 350 uni_vacated_space = 65
# vacated_space = 0 uni_is_vacated_space = 1 is_vacated_space = 0
#
# -----
# test 10: 80 ordinary cars want to exit
# timer = 15: 58 uni_parked_car = 285 parked_car = 270 uni_vacated_space = 65
# vacated_space = 80 uni_is_vacated_space = 1 is_vacated_space = 1
#
# -----
# test 11: 60 ordinary cars want to enter
# timer = 16: 43 uni_parked_car = 200 parked_car = 415 uni_vacated_space = 0
# vacated_space = 85 uni_is_vacated_space = 0 is_vacated_space = 1
#
# -----
# test 12: 70 university cars want to enter
# timer = 17: 28 uni_parked_car = 200 parked_car = 485 uni_vacated_space = 0
# vacated_space = 15 uni_is_vacated_space = 0 is_vacated_space = 1
#
```

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 و فضای 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 و ظرفیت خالی دانشگاهی صفر میماند.

```

test 13: 70 universitys car want to exit
timer =      18:      13      uni_parked_car =      130 parked_car =      485 uni_vacated_space =      70
vacated_space =      15 uni_is_vacated_space = 1 is_vacated_space = 1

test 14: 30 university cars want to enter
timer =      19:      53      uni_parked_car =      160 parked_car =      485 uni_vacated_space =      40
vacated_space =      15 uni_is_vacated_space = 1 is_vacated_space = 1

test 15: 30 ordinary cars want to enter
timer =      19:      55      uni_parked_car =      160 parked_car =      500 uni_vacated_space =      40
vacated_space =      0 uni_is_vacated_space = 1 is_vacated_space = 0

test 16: 100 ordinay cars want to enter
timer =      20:      40      uni_parked_car =      0 parked_car =      0 uni_vacated_space =      0
vacated_space =      0 uni_is_vacated_space = 0 is_vacated_space = 0
** Note: $stop      : C:/sharif/digital design/projectplus/question1/testbench.v(231)

```

تست 13: در ساعت 18:13 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 تا ماشین دانشگاهی وارد میشوند و ظرفیت ماشین های دانشگاهی برابر 160 میشود.

تست 15: در ساعت 19:55 تا 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 تا 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 فولدر miscellaneous 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 ماکسیم تاخیر 7.454 است.

TimeQuest Timing Analyzer - C:/Users/Ali Ghasemzadeh/Desktop/fpgaproject/parking - parking

Report: Setup: clk

| Slack | From Node | To Node | Launch Clock | Latch Clock | Relationship | Clock Skew | Data Delay |
|--------|----------------------------|----------------------------|--------------|-------------|--------------|------------|------------|
| -8.054 | uni_parked_car[18]\$slatch | uni_parked_car[0]\$slatch | car_entered | clk | 1.000 | -4.027 | 4.097 |
| -8.054 | uni_parked_car[2]\$slatch | uni_parked_car[25]\$slatch | car_entered | clk | 1.000 | -1.097 | 7.129 |
| -8.054 | uni_parked_car[1]\$slatch | uni_parked_car[23]\$slatch | car_entered | clk | 1.000 | -1.077 | 7.180 |
| -8.052 | uni_parked_car[10]\$slatch | uni_parked_car[25]\$slatch | car_entered | clk | 1.000 | -1.231 | 6.993 |
| -8.051 | uni_parked_car[17]\$slatch | uni_parked_car[0]\$slatch | car_entered | clk | 1.000 | -4.017 | 4.104 |
| -8.050 | uni_parked_car[1]\$slatch | uni_parked_car[1]\$slatch | car_entered | clk | 1.000 | -1.075 | 7.158 |
| -8.048 | uni_parked_car[9]\$slatch | uni_parked_car[3]\$slatch | car_entered | clk | 1.000 | -1.245 | 6.982 |
| -8.039 | uni_parked_car[1]\$slatch | uni_parked_car[24]\$slatch | car_entered | clk | 1.000 | -0.934 | 7.454 |
| -8.037 | uni_parked_car[3]\$slatch | uni_parked_car[18]\$slatch | car_entered | clk | 1.000 | -1.065 | 7.151 |
| -8.036 | uni_parked_car[8]\$slatch | uni_parked_car[18]\$slatch | car_entered | clk | 1.000 | -1.088 | 7.127 |
| -8.034 | uni_parked_car[11]\$slatch | uni_parked_car[18]\$slatch | car_entered | clk | 1.000 | -1.076 | 7.137 |
| -8.030 | uni_parked_car[1]\$slatch | uni_parked_car[9]\$slatch | car_entered | clk | 1.000 | -0.935 | 7.431 |
| -8.027 | uni_parked_car[1]\$slatch | uni_parked_car[10]\$slatch | car_entered | clk | 1.000 | -0.936 | 7.430 |

Report Timing: Found 200 setup paths (200 violated). Worst case slack is -8.520
 Report Timing: Found 200 setup paths (200 violated). Worst case slack is -7.871
 Report Timing: Found 200 hold paths (200 violated). Worst case slack is -3.419
 Report Timing: Found 200 hold paths (200 violated). Worst case slack is -1.905
 No fmax paths to report
 No fmax paths to report

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

TimeQuest Timing Analyzer - C:/Users/Ali Ghasemzadeh/Desktop/fpgaproject/parking - parking

Report: Fmax Summary

| Fmax | Restricted Fmax | Clock Name | Note |
|------|-----------------|------------|-------------|
| 1 | 115.59 MHz | 115.59 MHz | car_entered |
| 2 | 116.71 MHz | 116.71 MHz | clk |

This panel reports FMAX for every clock in the design, regardless of the user-specified clock periods. FMAX is only computed for paths where the source and destination registers or ports are driven by the same clock. Paths of different clocks, including generated clocks, are ignored. For paths between a clock and its inversion, FMAX is computed as if the rising and falling edges are scaled along with FMAX, such that the duty cycle (in terms of a percentage) is maintained. Altera recommends that you always use clock constraints and other slack reports for sign-off analysis.

update_timing_netlist
 No user constrained base clocks found in the design. Calling "derive_clocks -period 1.0"
 Deriving Clocks
 No user constrained clock uncertainty found in the design. Calling "derive_clock_uncertainty"
 Deriving Clock Uncertainty. Please refer to report_sdc in TimeQuest to see clock uncertainties.
 report_clock_fmax_summary -panel_name "Fmax Summary"

فراکانس ماکسیمم برابر کلاک برابر 116.71 مگاهرتز است.