

نام درس: طراحی سیستم‌های دیجیتال

شماره دانشجویی: ۴۰۱۱۰۶۳۶۳

نام و نام خانوادگی: میترا قلی پور

سؤال ۸:

مداری برای مدیریت پارکینگ دانشگاه طراحی کنید که امکانات زیر را داشته باشد:

- (۱) اولویت فضای پارکینگ با اساتید و کارمندان دانشگاه است و این ظرفیت بر اساس آمار حداکثر ۵۰۰ خودرو تعیین گردیده است.
- (۲) باتوجه به اینکه فضای کل پارکینگ ۷۰۰ خودرو است از ساعت ۸ تا ۱۳ فقط ۲۰۰ ظرفیت خالی برای ورود آزاد موجود است.
- (۳) از ساعت ۱۳ تا ۱۶ به ازای هر ساعت ظرفیت ورود آزاد ۵۰ خودرو افزایش می‌یابد و در ساعت ۱۶ ظرفیت ورود آزاد به ۵۰۰ خودرو می‌رسد.

(الف) اگر در هنگام ورود/خروج خودرو یک سیگنال ورودی به مدار نوع آن را مشخص کند (دانشگاه/آزاد)؛ با زبان وریلاگ مداری را توصیف کنید که دارای ورودی‌ها/خروجی‌های زیر باشد:

خروجی‌ها	
uni_parked_car	تعداد خودروهایی متعلق به دانشگاه که در پارکینگ پارک شده‌اند.
parked_care	تعداد خودروهای پارک شده در پارکینگ مربوط به ظرفیت آزاد
uni_vacated_space	تعداد فضای خالی متعلق به دانشگاه
vacated_space	تعداد فضاهای خالی مربوط به ظرفیت آزاد
uni_is_vacated_space	آیا فضای خالی برای دانشگاه موجود است؟
is_vacated_space	آیا فضای خالی برای ظرفیت آزاد موجود است؟
ورودی‌ها	
car_entered	ورود یک خودرو
is_uni_car_entered	آیا خودرو وارد شده متعلق به دانشگاه است؟
car_exited	خروج یک خودرو
is_uni_car_exited	آیا خودرو خارج شده متعلق به دانشگاه است؟

(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 ۸ 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 ۲۴ ساعت است هر ۲۴ 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
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
end

```

تنظیم متغیرها براساس ورودی ها

در صورتی که ماشین وارد شود، اگر همچنان در بخشش فضای آزاد باشد، می تواند پارک کند. همچنین هنگامی که خارج می شود باید تعداد ماشین های پارک شده مثبت باشد، در غیر این صورت خارج شدن بی معنی 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
    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
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 ۳ 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 ماشین‌های 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 در نتیجه 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 ۷۰۰ 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 ظرفیت‌ها بدون ورود و خروج ماشین‌ها:

```

VSIM 2> run -all
# At time 8, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 500, f_vacated_space = 200
# At time 9, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 500, f_vacated_space = 200
# At time 10, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 500, f_vacated_space = 200
# At time 11, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 500, f_vacated_space = 200
# At time 12, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 500, f_vacated_space = 200
# At time 13, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 450, f_vacated_space = 250
# At time 14, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 400, f_vacated_space = 300
# At time 15, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 350, f_vacated_space = 350
# At time 16, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 17, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 18, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 19, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 20, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 21, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 22, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 23, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 0, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 1, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 2, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 3, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 4, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 5, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 6, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500
# At time 7, uni_parked_car = 0, f_parked_car = 0, uni_vacated_space = 200, f_vacated_space = 500

```

ورود بیش از ظرفیت ماشین‌های آزاد:

```

# enter 202 free cars:
# At time 8, uni_parked_car = 0, f_parked_car = 200, uni_vacated_space = 500, f_vacated_space = 0

// 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:
# At time 10, uni_parked_car = 200, f_parked_car = 200, uni_vacated_space = 300, f_vacated_space = 0
# exit 100 university cars:
# At time 11, uni_parked_car = 100, f_parked_car = 200, uni_vacated_space = 400, f_vacated_space = 0

// 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:
# At time 12, uni_parked_car = 100, f_parked_car = 200, uni_vacated_space = 400, f_vacated_space = 0
```

```
// 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:  
# At time 12, uni_parked_car = 500, f_parked_car = 200, uni_vacated_space = 0, f_vacated_space = 0
```

```
// 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 ۷۰۰ هم در نظر گرفته شده است. همچنین ماشین دانشگاهی مازاد بر ظرفیت در پارکینگ هستند، پس دیگر نمی توانند وارد شوند.

```
# exit 25 university cars:  
# At time 12, uni_parked_car = 475, f_parked_car = 200, uni_vacated_space = 25, f_vacated_space = 0  
# enter 50 free cars:  
# At time 14, uni_parked_car = 475, f_parked_car = 225, uni_vacated_space = 0, f_vacated_space = 0
```

```
// 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:
# At time 14, uni_parked_car = 345, f_parked_car = 225, uni_vacated_space = 55, f_vacated_space = 75
```

```
// 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:
# At time 14, uni_parked_car = 345, f_parked_car = 225, uni_vacated_space = 55, f_vacated_space = 75
```

```
// 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 ۸۲ تا شمرده یعنی ۱۸ تا کلاک تا ساعت بعدی فاصله هست. طبق کد هر ماشین طی دوتا کلاک میرود پس ما انتظار داریم بعد ۹ کلاک دیگر ماشین‌ها وارد نشوند:

```
# At time 14, uni_parked_car = 345, f_parked_car = 225, uni_vacated_space = 55, f_vacated_space = 75
# clock counter:      82
# enter 200 university cars:
# At time 18, uni_parked_car = 354, f_parked_car = 225, uni_vacated_space = 0, f_vacated_space = 121
```

```

$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 ۷ به ۸:

```

# exit 40 free cars:
# At time 23, uni_parked_car = 354, f_parked_car = 306, uni_vacated_space = 0, f_vacated_space = 40
# exit 400 university cars :
# At time 7, uni_parked_car = 0, f_parked_car = 306, uni_vacated_space = 200, f_vacated_space = 194
# At time 8, uni_parked_car = 0, f_parked_car = 306, uni_vacated_space = 394, f_vacated_space = 0
# Break in Module tb_parking_controller at C:/Users/mi/OneDrive/Desktop/DSD-parking-project/tb_controller.v line 239

// 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 ۱۲ و از ۷ به ۸، ورود و خروج در لحظه تغییرات ظرفیت، ورود و خروج معمولی، خارج ظرفیت، همزمان و ۷۰۰ بودن مجموع فضاها و نامنفی بودن آن‌ها در هر کدام از حالت‌ها را بررسی کرده‌ایم و نتایج درستی دریافت کردیم.

(ب) سنتز کردن FPGA:

از نرم افزار کوآرتوس دیوایس MAX V استفاده کردم.

	Fmax	Restricted Fmax	Clock Name	Note
1	67.34 MHz	67.34 MHz	car_entered	
2	88.07 MHz	88.07 MHz	clk	

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_entered این فرکانس بالاترین نرخ 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.