

(سوال 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 می یابد و در ساعت ۱۶ ظرفیت ورود آزاد به ۵۰۰ خودرو می رسد.

(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؟

در صورتی که نیاز به ورودی‌ها/خروجی‌های دیگری هم است آن را با ذکر دلیل به طراحی خود بیفزایید و جهت اطمینان از صحت عملکرد مدار، مدار خود را مورد آزمون قرار دهید (۵۰ نمره).

(ب) مدار خود را برای یک FPGA به انتخاب خود سنتز کنید. از گزارش‌های سنتز، بیشترین فرکانس ممکن برای این مدار را با ذکر دلیل مشخص کنید (۱۰ نمره).

(الف)

برای محاسبه زمان من مدول Counter را طراحی کردم که این مدول پارامتر n را

میگیرد ( که مشخص کننده این است که هر چند کلاک را معادل یک ساعت میداند ).

و همچنین سیگنال ورودی clk و سیگنال خروجی hour و سیگنال count که در واقع به

نوعی نشان دهنده دقیقه هست برای Counter است.

با متغیر count تعداد کلاک های گذشته را می‌شماریم و اگر برابر با پارامتر n شد سیگنال

hour را فعال می‌کنیم و در بقیه حالات hour را با 0 مقداردهی می‌کنیم.

من هر 60 کلاک را معادل گذشت یک ساعت میدانم.

```

58 module Counter #(parameter n = 60) (input clk, output reg hour,output reg[6:0] count);
59
60     initial begin
61         hour = 0;
62         count = 0;
63     end
64
65     always @(posedge clk) begin
66         count <= count + 1;
67         if (count % (n-1) == 0 && count!=0)begin
68             hour <= 1;
69             count <= 0;
70         end
71         else
72             hour <= 0;
73     end
74 endmodule

```

سپس مدول Parking را طبق خواسته سوال ایجاد کردم و سپس همانطوری که

سوال از ما خواسته ورودی ها و خروجی های مدنظر را در پورت های مناسب قرار دادم و

از طرفی برای اعداد ۲۰۰ و ۵۰۰ همان ۹ بیت کافی است. و همینطور ورودی کلاک را نیز

برای مدول Parking در نظر گرفتم و از طرفی ورودی و خروجی در صورت سوال را نیز

در پورت های مناسب قرار دادم. موضوع دیگر در مورد مقدار ساعت و دقیقه است که

برای نمایش ساعت و دقیقه به ترتیب سیگنال های timer\_hour و timer\_counter را

تعریف کردم.

برای نگه داری حداکثر ظرفیت آزاد و دانشگاه دو متغیر university\_capacity و

public\_capacity را تعریف کردیم و برای مقداردهی اولیه دلخواه به آنها از تعریف پارامتر

استفاده کردیم.

و همچنین برای اینکه بفهمیم ساعت بعدی فرا رسیده از سیگنال hour استفاده میکنیم

( که به مدول Counter وصل است ).

مقداردهی های اولیه را مطابق خواسته سوال در یک بلاک initial انجام میدهیم.

قسمت پورت و مقدار دهی اولیه سیگنال های خروجی و سیگنال های داخلی به شرح زیر

است:

```
1 module Parking #(parameter uni_vacated_space_initial=500, parameter time_initial=8,parameter university_capacity_initial=500,
2 parameter vacated_space_initial=200,parameter public_capacity_initial=200)
3 (input car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk,
4 output reg uni_is_vacated_space, is_vacated_space,
5 output reg [9:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space,output wire [6:0]time_counter,output reg[4:0] timer_hour);
6
7 reg [9:0] university_capacity;
8 wire hour;
9 reg [9:0] public_capacity;
10 reg [9:0] changed_capacity;
11
12 initial begin
13     parked_car = 0;
14     uni_vacated_space = uni_vacated_space_initial;
15     timer_hour= time_initial;
16     uni_parked_car = 0;
17     university_capacity = university_capacity_initial;
18     vacated_space = vacated_space_initial;
19     public_capacity = public_capacity_initial;
20 end
```

خب حالا به سراغ توضیح مدول parking و نحوه عملکرد آن میرویم:

در این قسمت یک نکته کلیدی وجود دارد و آن هم این است که همواره اولویت با پرسنل

دانشگاهی است. به این معنا که اگر قرار بر اضافه کردن ظرفیت به قسمت آزاد باشد اگر

به اندازه سقف تعیین شده جای خالی باشد همان مقدار و گرنه به اندازه ظرفیت خالی

اضافه میکنیم.

نکته دیگر در مورد اضافه کردن ظرفیت است که چون از ساعت ۱۳ به بعد به ازای گذشت هر یک ساعت ما ظرفیت های آزاد را زیاد میکنیم پس فقط به ازای ساعات ۱۴ و ۱۵ ظرفیت تا حداکثر ۵۰ تا زیاد میشود و همچنین در ساعت ۱۶ نیز باید ظرفیت را با توجه به اینکه دانشگاه اولویت دارد زیاد کنیم. مثلاً ببینیم آیا فضای خالی که از دانشگاه باقی مانده به اندازه ای است که بتوان ظرفیت آزاد را تا ۵۰۰ افزایش داد.

```

23 always @(posedge hour or posedge clk) begin
24     if (hour) begin
25         timer_hour = timer_hour + 1;
26         if (timer_hour % 24 == 14 || timer_hour % 24 == 15) begin
27             changed_capacity = (uni_vacated_space < 50) ? uni_vacated_space : 50;
28             university_capacity = university_capacity - changed_capacity;
29             uni_vacated_space = uni_vacated_space - changed_capacity;
30             public_capacity = public_capacity + changed_capacity;
31             vacated_space = vacated_space + changed_capacity;
32         end else if (timer_hour % 24 == 16) begin
33             university_capacity = (uni_parked_car > 200) ? uni_parked_car : 200;
34             public_capacity = 700 - university_capacity;
35             vacated_space = public_capacity - parked_car;
36             uni_vacated_space = university_capacity - uni_parked_car;
37         end
38     end else if (clk) begin

```

بخش مربوط به ورود ماشین و تغییر ظرفیت آزاد و دانشگاهی نیز که با سیگنال clk کار میکند به شکل زیر میباشد:

```

38     end else if (clk) begin
39         uni_is_vacated_space = (uni_vacated_space != 0) ? 1 : 0;
40         is_vacated_space = (vacated_space != 0) ? 1 : 0;
41         if (car_entered && is_uni_car_entered && uni_is_vacated_space ) begin
42             uni_parked_car = uni_parked_car + 1;
43             uni_vacated_space = uni_vacated_space - 1;
44         end else if (car_entered && !is_uni_car_entered && is_vacated_space) begin
45             parked_car = parked_car + 1;
46             vacated_space = vacated_space - 1;
47         end else if (car_exited && is_uni_car_exited && uni_parked_car > 0 ) begin
48             uni_parked_car = uni_parked_car - 1;
49             uni_vacated_space = uni_vacated_space + 1;
50         end else if (car_exited && !is_uni_car_exited && parked_car > 0) begin
51             parked_car = parked_car - 1;
52             vacated_space = vacated_space + 1;
53         end
54     end if(timer_hour%24==0)
55         timer_hour=0;
56 end

```

در این قسمت نیز مجدداً بحث اولویت ها و اینکه من باید ببینم اصلاً ظرفیت دارم که ماشین اضافه کنم را مشاهده میکنیم و به موضوع دیگر این که برای من در موقع تست گرفتن پیش آمد این بود که ممکن است سیگنال خروج ماشین زیاد فعال بماند و تعداد ماشین های باقی مانده به حدی کم شود که عدد حاصل منفی شود و من از این رو، شرط مثبت بودن تعداد ماشین های پارک شده را چک کردم. نکته دیگر چون من میخوام ساعت را هم نمایش دهم چک میکنم که اگر ساعت ۲۴ شد، به صفر تغییر کند.

حال به سراغ testBench نوشتن میرویم. تست بنچ من به شکل زیر است و خروجی های آن نیز در فایل output.txt آورده شده است:

```
75 module Parking_TB();
76     reg car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk;
77     wire uni_is_vacated_space, is_vacated_space;
78     wire [9:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space;
79     wire [6:0] time_counter;
80     wire [4:0] time_hour;
81
82     always
83     #5 clk = ~clk;
84
85     Parking #(500,8,500,200,200) parking(car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk,
86     uni_is_vacated_space, is_vacated_space, uni_parked_car, parked_car, uni_vacated_space, vacated_space, time_counter, time_hour);
87
88     initial begin
89         car_entered=0;
90         is_uni_car_entered=0;
91         car_exited=0;
92         is_uni_car_exited=0;
93         clk=0;
94         #20 car_entered = 1;
95         #30 is_uni_car_entered = 1;
96         #50 car_entered = 0; is_uni_car_entered = 0; car_exited = 1; is_uni_car_exited = 0;
97         #50 car_exited = 1; is_uni_car_exited = 1;
98         #10000 $finish();
99     end
100
101     initial begin
102         $monitor("time= %d:%d\n: car_entered = %b || is_uni_car_entered = %b || car_exited = %b || is_uni_car_exited = %b >>>> uni_is_vacated_space = %d ||",
103         time_hour, time_counter, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, uni_is_vacated_space,
104         "is_vacated_space = %d \n|| uni_parked_car = %d || parked_car = %d || uni_vacated_space = %d || vacated_space = %d",
105         is_vacated_space, uni_parked_car, parked_car, uni_vacated_space, vacated_space);
106     end
107     initial begin
108         $dumpfile("Parking.vcd");
109         $dumpvars(0, Parking_TB);
110     end
111 endmodule
```

بخشی از خروجی این testBench به شکل زیر است:

```

2  time= 8: 0
3  : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = x || is_vacated_space = x
4  || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200
5  time= 8: 1
6  : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
7  || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200
8  time= 8: 2
9  : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
10 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200
11 time= 8: 2
12 : car_entered = 1 || is_uni_car_entered = 0 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
13 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200
14 time= 8: 3
15 : car_entered = 1 || is_uni_car_entered = 0 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
16 || uni_parked_car = 0 || parked_car = 1 || uni_vacated_space = 500 || vacated_space = 199
17 time= 8: 4
18 : car_entered = 1 || is_uni_car_entered = 0 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
19 || uni_parked_car = 0 || parked_car = 2 || uni_vacated_space = 500 || vacated_space = 198
20 time= 8: 4
21 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
22 || uni_parked_car = 0 || parked_car = 2 || uni_vacated_space = 500 || vacated_space = 198
23 time= 8: 5
24 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
25 || uni_parked_car = 1 || parked_car = 2 || uni_vacated_space = 499 || vacated_space = 198
26 time= 8: 6
27 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
28 || uni_parked_car = 2 || parked_car = 2 || uni_vacated_space = 498 || vacated_space = 198

```

اگر بخواهیم کمی خروجی را تحلیل کنیم تا از عملکرد درست آن مطمئن شویم:

در ساعت ۸ که سیستم شروع به کار میکند باید ظرفیت های دانشگاه و آزاد به ترتیب 500 و 200 باشد که

همانطور که میبینیم اینگونه است.

در ساعت 8:02 یک ماشین آزاد وارد میشود که نتجیتا در کلاک بعدی باید ظرفیت آزاد یکی کم شده باشد، که همینطور نیز است.

این روند ادامه دارد و مدام بر تعداد ماشین های آزاد افزوده میشود تا اینکه در زمان 8:05 نوع این

ماشین عوض میشود و ماشین پرسنلی وارد میشود از این زمان تا ساعت 8:10 مدام ماشین

پرسنلی وارد میشود

```

24 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
25 || uni_parked_car = 0 || parked_car = 3 || uni_vacated_space = 500 || vacated_space = 197
26 time= 8: 6
27 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
28 || uni_parked_car = 1 || parked_car = 3 || uni_vacated_space = 499 || vacated_space = 197
29 time= 8: 7
30 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
31 || uni_parked_car = 2 || parked_car = 3 || uni_vacated_space = 498 || vacated_space = 197
32 time= 8: 8
33 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
34 || uni_parked_car = 3 || parked_car = 3 || uni_vacated_space = 497 || vacated_space = 197
35 time= 8: 9
36 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
37 || uni_parked_car = 4 || parked_car = 3 || uni_vacated_space = 496 || vacated_space = 197
38 time= 8: 10
39 : car_entered = 1 || is_uni_car_entered = 1 || car_exited = 0 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
40 || uni_parked_car = 5 || parked_car = 3 || uni_vacated_space = 495 || vacated_space = 197

```

و همانطور که میبینیم از اینجا به بعد دیگر ماشینی وارد نمیشود و تا ساعت 8:13 ما با خروج ماشین

های آزاد روبرو هستیم.

```

41 time= 8: 10
42 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
43 || uni_parked_car = 5 || parked_car = 3 || uni_vacated_space = 495 || vacated_space = 197
44 time= 8: 11
45 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
46 || uni_parked_car = 5 || parked_car = 2 || uni_vacated_space = 495 || vacated_space = 198
47 time= 8: 12
48 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
49 || uni_parked_car = 5 || parked_car = 1 || uni_vacated_space = 495 || vacated_space = 199
50 time= 8: 13
51 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 0 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
52 || uni_parked_car = 5 || parked_car = 0 || uni_vacated_space = 495 || vacated_space = 200

```

سپس در ساعت 8:15 ماشین های دانشگاهی شروع به خروج میکنند و این روند تا ساعت 8:20 ادامه

خواهد داشت:

```

59 time= 8: 15
60 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
61 || uni_parked_car = 5 || parked_car = 0 || uni_vacated_space = 495 || vacated_space = 200
62 time= 8: 16
63 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
64 || uni_parked_car = 4 || parked_car = 0 || uni_vacated_space = 496 || vacated_space = 200
65 time= 8: 17
66 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
67 || uni_parked_car = 3 || parked_car = 0 || uni_vacated_space = 497 || vacated_space = 200
68 time= 8: 18
69 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
70 || uni_parked_car = 2 || parked_car = 0 || uni_vacated_space = 498 || vacated_space = 200
71 time= 8: 19
72 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
73 || uni_parked_car = 1 || parked_car = 0 || uni_vacated_space = 499 || vacated_space = 200
74 time= 8: 20
75 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
76 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200

```

از نکات دیگر طراحی میتوان به اینکه در ساعت 14 و 15 و 16 باید ظرفیت آزاد زیاد شود اشاره کرد که در ادامه میبینیم:

```

554 time= 14: 0
555 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
556 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200
557 time= 14: 1
558 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
559 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 450 || vacated_space = 250
560 time= 14: 2
561 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
562 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 450 || vacated_space = 250
563 time= 14: 3
564 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
565 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 450 || vacated_space = 250
566 time= 14: 4
567 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
568 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 450 || vacated_space = 250
569 time= 14: 5
570 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
571 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 450 || vacated_space = 250
572 time= 14: 6
573 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
574 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 450 || vacated_space = 250

```

```

734 time= 15: 0
735 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
736 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 400 || vacated_space = 300
737 time= 16: 1
738 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
739 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 200 || vacated_space = 500
740 time= 16: 2
741 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
742 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 200 || vacated_space = 500
743 time= 16: 3
744 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
745 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 200 || vacated_space = 500
746 time= 16: 4
747 : car_entered = 0 || is_uni_car_entered = 0 || car_exited = 1 || is_uni_car_exited = 1 >>>> uni_is_vacated_space = 1 || is_vacated_space = 1
748 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 200 || vacated_space = 500
749 time= 16: 5

```



ب) فایل مربوط به Quartus این پروژه در فولدر پروژه آمده است.

همانطور که در تصویر زیر میبینیم بیشینه فرکانس برابر با 6.46MHz است.

	Fmax	Restricted Fmax	Clock Name	Note
1	6.46 MHz	6.46 MHz	clk	
2	134.93 MHz	134.93 MHz	Counter:counter hour	

سیگنال clk همان سیگنال کلاک مدول Parking است. ما میتوانیم درستی این فرکانس را با مراجعه

کردن به گزارش تاخیر مسیر داده ها چک کنیم:

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-153.895	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.624
2	-153.885	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.614
3	-153.885	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.614
4	-153.885	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.614
5	-153.885	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.614
6	-153.884	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.613
7	-153.884	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.613
8	-153.884	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.613
9	-153.884	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.613
10	-153.883	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.612
11	-153.882	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.611
12	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
13	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
14	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
15	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
16	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
17	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
18	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
19	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
20	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
21	-153.875	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.604
22	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
23	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
24	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
25	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
26	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
27	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
28	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
29	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
30	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
31	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
32	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
33	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
34	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603
35	-153.874	Counter:counter count[2]	Counter:counter hour	clk	clk	1.000	0.000	154.603

به طور کلی تاخیر مسیر داده ها برابر با 154ns است و همانطور که میبینیم فرکانس بیشینه تقریباً با

معکوس آن برابر است:

$$1/154\text{ns} \sim 6.49\text{MHz}$$