

(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 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؟

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)

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

سوال از ما خواسته ورودی‌ها و خروجی‌های مدنظر را در پورت‌های 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.

برای نگه داری حداکثر ظرفیت آزاد و دانشگاه دو متغیر `university_capacity` و

`public_capacity` را تعریف کردیم و برای محاسبه ساعت از متغیر `timer` 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 (input car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk,
2   output reg uni_is_vacated_space, is_vacated_space,
3   output reg [9:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space);
4
5   reg [9:0] university_capacity;
6   wire hour;
7   reg [9:0] public_capacity;
8   reg [9:0] changed_capacity;
9   integer timer;
10
11   initial begin
12     parked_car = 0;
13     uni_vacated_space = 500;
14     timer = 8;
15     uni_parked_car = 0;
16     university_capacity = 500;
17     vacated_space = 200;
18     public_capacity = 200;
19   end
20
```

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` نیز 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.

```

= Counter.v
1  module Counter #(parameter n = 3) (input clk, output reg hour);
2      integer count;
3
4      initial begin
5          hour = 0;
6          count = 0;
7      end
8
9      always @(posedge clk) begin
10         count = count + 1;
11         if (count % n == 0)
12             hour = 1;
13         else
14             hour = 0;
15     end
16 endmodule

```

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 به ازای گذشت هر یک ساعت ما

ظرفیت ها 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.

```

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

```

بخش مربوط به ورود ماشین و تغییر ظرفیت آزاد و دانشگاهی نیز که با سیگنال clk کار

میکند به شکل زیر میباشد:

```

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

```

در این قسمت نیز مجددا بحث اولویت ها و اینکه من باید ببینم اصلا ظرفیت دارم که

ماشین اضافه کنم را مشاهده میکنیم.

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:

```
71 endmodule
72 module Parking_TB();
73   reg car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk;
74   wire uni_is_vacated_space, is_vacated_space;
75   wire [9:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space;
76
77   always
78   | #5 clk = ~clk;
79
80   Parking parking(car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk,
81   uni_is_vacated_space, is_vacated_space, uni_parked_car, parked_car, uni_vacated_space, vacated_space);
82
83   initial begin
84     car_entered=0;
85     is_uni_car_entered=0;
86     car_exited=0;
87     is_uni_car_exited=0;
88     clk=0;
89     #300 car_entered = 1;
90     #60 is_uni_car_entered = 1;
91     #70 car_entered = 0; is_uni_car_entered = 0; car_exited = 1; is_uni_car_exited = 1;
92     #70 car_entered = 1; is_uni_car_entered = 1; car_exited = 0; is_uni_car_exited = 0;
93     #200 $finish();
94   end
95
96   initial begin
97     $monitor("time= %t\n: car_entered = %b || is_uni_car_entered = %b || car_exited = %b || is_uni_car_exited = %b >>>> uni_is_vacated_space = %d || ",
98     $time, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, uni_is_vacated_space,
99     "is_vacated_space = %d || uni_parked_car = %d || parked_car = %d || uni_vacated_space = %d || vacated_space = %d",
100     is_vacated_space, uni_parked_car, parked_car, uni_vacated_space, vacated_space);
101   end
102   initial begin
103     $dumpfile("Parking.vcd");
104     $dumpvars(0, Parking_TB);
105   end
106 endmodule
```

بخشی از خروجی این testBench به شکل زیر است:

```

2  time=          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=          5
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=        300
9  : 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
10 || uni_parked_car = 0 || parked_car = 0 || uni_vacated_space = 500 || vacated_space = 200
11 time=        305
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 = 1 || uni_vacated_space = 500 || vacated_space = 199
14 time=        315
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 = 2 || uni_vacated_space = 500 || vacated_space = 198
17 time=        325
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 = 3 || uni_vacated_space = 500 || vacated_space = 197
20 time=        335
21 : 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
22 || uni_parked_car = 0 || parked_car = 4 || uni_vacated_space = 500 || vacated_space = 196
23 time=        345
24 : 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
25 || uni_parked_car = 0 || parked_car = 5 || uni_vacated_space = 500 || vacated_space = 195
26 time=        355
27 : 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
28 || uni_parked_car = 0 || parked_car = 6 || uni_vacated_space = 500 || vacated_space = 194
29 time=        360
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 = 0 || parked_car = 6 || uni_vacated_space = 500 || vacated_space = 194
32 time=        365
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 = 1 || parked_car = 6 || uni_vacated_space = 499 || vacated_space = 194
35 time=        375
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 = 2 || parked_car = 6 || uni_vacated_space = 498 || vacated_space = 194
38 time=        385
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 = 3 || parked_car = 6 || uni_vacated_space = 497 || vacated_space = 194
41 time=        395

```

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

در ابتدا ( $t=0$ ) که مدول شروع به کار میکند باید ظرفیت های دانشگاه و آزاد به ترتیب 500 و 200 باشد که

همانطور که میبینیم اینگونه است.

در ( $t=300$ ) یک ماشین آزاد وارد میشود که نتیجتاً در کلاک بعدی باید ظرفیت آزاد یکی کم شود که همینطور است.

این روند ادامه دارد و مدام بر تعداد ماشین های آزاد افزوده میشود تا اینکه در زمان ( $t=360$ ) نوع این

ماشین عوض میشود و ماشین پرسنلی وارد میشود از این کلاک تا 70 واحد زمانی دیگر مدام ماشین پرسنلی وارد میشود و همانطور که میبینیم از اینجا به بعد دیگر ماشینی وارد نمیشود و ما با خروج ماشین های آزاد روبرو هستیم.

```

53 time= 430
54 : 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
55 || uni_parked_car = 7 || parked_car = 6 || uni_vacated_space = 493 || vacated_space = 194
56 time= 435
57 : 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
58 || uni_parked_car = 7 || parked_car = 5 || uni_vacated_space = 493 || vacated_space = 195
59 time= 445
60 : 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
61 || uni_parked_car = 7 || parked_car = 4 || uni_vacated_space = 493 || vacated_space = 196
62 time= 455
63 : 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
64 || uni_parked_car = 7 || parked_car = 3 || uni_vacated_space = 493 || vacated_space = 197
65 time= 465
66 : 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
67 || uni_parked_car = 7 || parked_car = 2 || uni_vacated_space = 493 || vacated_space = 198
68 time= 475
69 : 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
70 || uni_parked_car = 7 || parked_car = 1 || uni_vacated_space = 493 || vacated_space = 199
71 time= 485
72 : 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
73 || uni_parked_car = 7 || parked_car = 0 || uni_vacated_space = 493 || vacated_space = 200

```

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

به طور کلی تاخیر مسیر داده ها برابر با  $154\text{ns}$  است و همانطور که میبینیم فرکانس بیشینه تقریبا با

معکوس آن برابر است:

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