

جبرانی پایانترم طراحی سیستم‌های دیجیتال، سوال 8

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؟

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) ابتدا ماژولی به نام TimeHandler ساختم که در آن با یک پارامتر n ، معلوم می‌شود که هر ساعت چند کلاک پالس است. برای مثال مقدار دیفالت آن را برابر 700 تعیین کردم (برای اینکه ظرفیت پارکینگ 700 نفر است و برای اینکه بتوانم در هر ساعت، تمامی ظرفیت را پر یا خالی کنم)

ماژول TimeHandler در زیر موجود است:

```

1  module TimeHandler #(parameter n = 700) (input CLK, Start, output reg [4:0]hour);
2      reg [31:0] counter = 0;
3      always @(posedge Start or posedge CLK) begin
4          if (Start) begin
5              hour = 32'd0;
6              counter = 0;
7          end else begin
8              counter = counter + 1;
9              if (counter % n == 0) begin
10                 counter = 0;
11                 hour = hour + 1;
12                 if (hour == 32'd24) begin
13                     hour = 32'd0;
14                 end
15             end
16         end
17     end
18 endmodule
19

```

ماژول Parking، ماژول اصلی برنامه است، بدین صورت که علاوه بر سیگنال‌های ورودی و خروجی گفته شده در فایل سوالات میانترم، تعدادی ورودی و خروجی به آن افزوده شده که در زیر بیان می‌شود:

CLK: کلاک دستگاه

Start: سیگنال استارت که می‌توان آن را به عنوان ریست نیز در نظر گرفت، می‌توان با 1 کردن آن، شمارش ساعات را از اول آغاز کرد و مدار از ساعت 0 مجدد شروع به کار می‌کند.

overflow: حالتی را در نظر بگیرید که در ساعت 7 صبح هستیم و ظرفیت ماشین‌های اساتید و کارکنان برابر 200 و ظرفیت ماشین‌های آزاد برابر 500 است. همچنین فرض کنید تعداد ماشین‌های اساتید و کارکنان برابر 100 و ماشین‌های آزاد برابر 400 باشد. حال اگر ساعت 8 صبح شود، طبق گفته سوال، ظرفیت ماشین‌های آزاد برابر 200 و ظرفیت ماشین‌های اساتید و کارکنان برابر 500 خواهد شد و این یعنی آن 400 ماشین آزاد، دیگر در ظرفیت 200 تایی جا نمی‌شوند! از آنجا که در سوال گفته نشده در این شرایط چه باید کرد، بنده فرض کردم که ناگهان 200 تا ماشین اضافی از پارکینگ خارج می‌شوند (!) و پارکینگ با 200 ماشین آزاد از ظرفیت آزاد پر خواهد شد و سیگنال overflow برابر 1 می‌شود.

Overflow_uni: همین حالت قبل را اینبار برای زمانی فرض کنید که ظرفیت تعداد ماشین‌های اساتید و کارکنان، بدلیل تغییر ساعت، به طور ناگهانی کمتر از تعداد ماشین‌های اساتید و کارکنان موجود در پارکینگ شود، در اینصورت همان اتفاق مشابه بالا رخ خواهد داد و سیگنال overflow_uni برابر یک می‌شود.

Rejected: این سیگنال زمانی 1 می‌شود که تعداد ماشین‌های آزاد در پارکینگ برابر 0 باشد و car_exited مساوی 1 شود. در اینصورت چون خروج ماشینی که وجود ندارد (!) غیر ممکن است، درخواست reject می‌شود. همچنین اگر ظرفیت ماشین‌های آزاد پر شده باشد و قصد افزودن ماشین جدید داشته باشیم همین اتفاق رخ می‌دهد.

Rejected_uni: این سیگنال مشابه سیگنال rejected اما برای ماشین‌های اساتید و کارکنان است.

حال کد ماژول Parking را در زیر قرار می‌دهم:

```

1 module Parking(input CLK, Start, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited,
2   output reg [9:0]uni_parked_car, output reg [9:0]parked_car, output reg[9:0]uni_vacated_space, output reg[9:0]vacated_space,
3   output reg uni_is_vacated_space, is_vacated_space, overflow_uni, overflow, rejected_uni, rejected, output wire [4:0] hour);
4   //reg [4:0]hour;
5   reg [9:0]uni_space;
6   reg [9:0]space;
7   TimeHandler #(700) th(CLK, Start, hour);
8   always @(Start or CLK or hour) begin
9       overflow = 0;
10      overflow_uni = 0;
11      if (hour == 8) begin
12          space = 10'd200;
13          uni_space = 10'd500;
14          if (space < parked_car) begin
15              overflow = 1;
16              parked_car = space;
17              vacated_space = 0;
18              uni_vacated_space = uni_space - uni_parked_car;
19          end else begin
20              vacated_space = space - parked_car;
21              uni_vacated_space = uni_space - uni_parked_car;
22          end
23      end else if (hour >= 13 && hour < 16) begin
24          space = space + 50;
25          uni_space = uni_space - 50;
26          if (uni_space < uni_parked_car) begin
27              overflow_uni = 1;
28              uni_parked_car = uni_space;
29              vacated_space = space - parked_car;
30              uni_vacated_space = 0;
31          end else begin
32              vacated_space = space - parked_car;
33              uni_vacated_space = uni_space - uni_parked_car;
34          end
35      end else if (hour == 16) begin
36          space = 500;
37          uni_space = 200;

```

```

38      if (uni_space < uni_parked_car) begin
39          overflow_uni = 1;
40          uni_parked_car = uni_space;
41          vacated_space = space - parked_car;
42          uni_vacated_space = 0;
43      end else begin
44          vacated_space = space - parked_car;
45          uni_vacated_space = uni_space - uni_parked_car;
46      end
47  end
48  if (Start) begin
49      uni_parked_car = 0;
50      parked_car = 0;
51      uni_vacated_space = 10'd200;
52      vacated_space = 10'd500;
53      space = 10'd500;
54      uni_space = 10'd200;
55      uni_is_vacated_space = 1;
56      is_vacated_space = 1;
57      rejected = 0;
58      overflow = 0;
59      rejected_uni = 0;
60      overflow_uni = 0;
61  end else if (car_entered == 1) begin
62      if (is_uni_car_entered) begin
63          if (uni_vacated_space == 0) begin
64              rejected_uni = 1;
65          end else begin
66              rejected_uni = 0;
67              uni_vacated_space = uni_vacated_space - 1;
68              uni_parked_car = uni_parked_car + 1;
69          end
70      end else begin
71          if (vacated_space == 0) begin

```

```

72         rejected = 1;
73     end else begin
74         rejected = 0;
75         vacated_space = vacated_space - 1;
76         parked_car = parked_car + 1;
77     end
78 end
79 end else if (car_exited == 1) begin
80     if (is_uni_car_exited) begin
81         if (uni_parked_car == 0) begin
82             rejected_uni = 1;
83         end else begin
84             rejected_uni = 0;
85             uni_vacated_space = uni_vacated_space + 1;
86             uni_parked_car = uni_parked_car - 1;
87         end
88     end else begin
89         if (parked_car == 0) begin
90             rejected = 1;
91         end else begin
92             rejected = 0;
93             vacated_space = vacated_space + 1;
94             parked_car = parked_car - 1;
95         end
96     end
97 end
98 end
99 endmodule
100

```

حال برای اطمینان از صحت عملکرد این ماژول‌ها، یک تست بنچ نوشتم که حالات بحرانی (مثل حالاتی که سیگنال‌های اورفلو برابر 1 می‌شوند) را در آن تست کرده‌ام: کد ماژول تست بنچ:

```

1 module Parking_TB;
2     reg CLK, Start, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited;
3     wire [9:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space;
4     wire uni_is_vacated_space, is_vacated_space, overflow_uni, overflow, rejected_uni, rejected;
5     wire [4:0] hour;
6     integer i;
7
8     Parking p(CLK, Start, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited,
9         uni_parked_car, parked_car, uni_vacated_space, vacated_space,
10        uni_is_vacated_space, is_vacated_space, overflow_uni, overflow, rejected_uni, rejected, hour
11    );
12
13    initial begin
14        CLK = 0;
15        Start = 1;
16        #1
17        Start = 0;
18        for (i = 0; i < 100; i = i + 1) begin
19            car_entered = 1;
20            is_uni_car_entered = 1;
21            #2;
22        end
23        for (i = 0; i < 500; i = i + 1) begin
24            car_entered = 1;
25            is_uni_car_entered = 0;
26            #2;
27        end
28        car_entered = 0;
29        // uni_parked_car = 100, parked_car = 500
30        #(1400 * 8); // 8 hours later -- uni_parked_car = 100, parked_car = 200 (overflow = 1)
31
32
33
34        for (i = 0; i < 300; i = i + 1) begin
35            car_entered = 1;
36            is_uni_car_entered = 1;
37            #2;

```

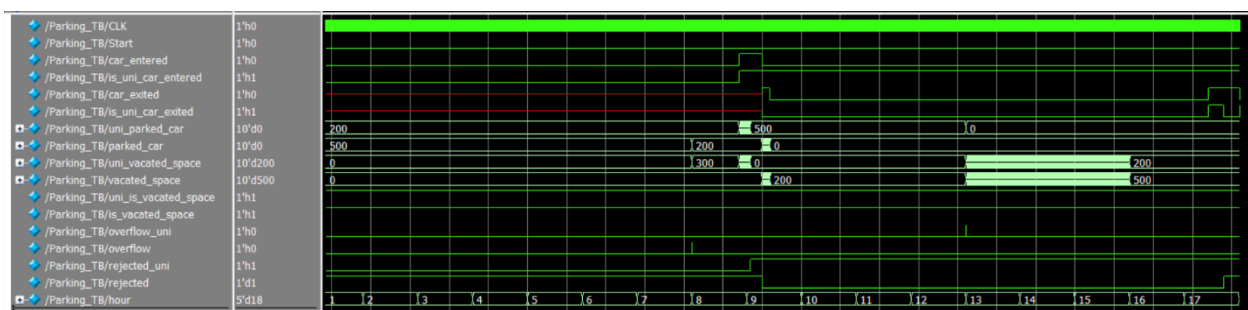
```

38     end
39     car_entered = 0;
40     for (i = 0; i < 100; i = i + 1) begin
41         car_exited = 1;
42         is_uni_car_exited = 0;
43         #2;
44     end
45     car_exited = 0;
46     // uni_parked_car = 400, parked_car = 100
47     #(1400 * 5); // 5 hours later
48     #(1400 * 3); // 3 hours later -- uni_parked_car = 200, parked_car = 100
49
50
51
52     for (i = 0; i < 200; i = i + 1) begin
53         car_exited = 1;
54         is_uni_car_exited = 1;
55         #2;
56     end
57     for (i = 0; i < 200; i = i + 1) begin
58         car_exited = 1;
59         is_uni_car_exited = 0;
60         #2;
61     end
62
63
64
65     car_exited = 1;
66     is_uni_car_exited = 0;
67     #2;
68     car_exited = 1;
69     is_uni_car_exited = 1;
70     #2;
71     car_exited = 0;
72
73     $stop();
74
75     end
76     always begin
77         #1 CLK = ~CLK;
78     end
79 endmodule

```

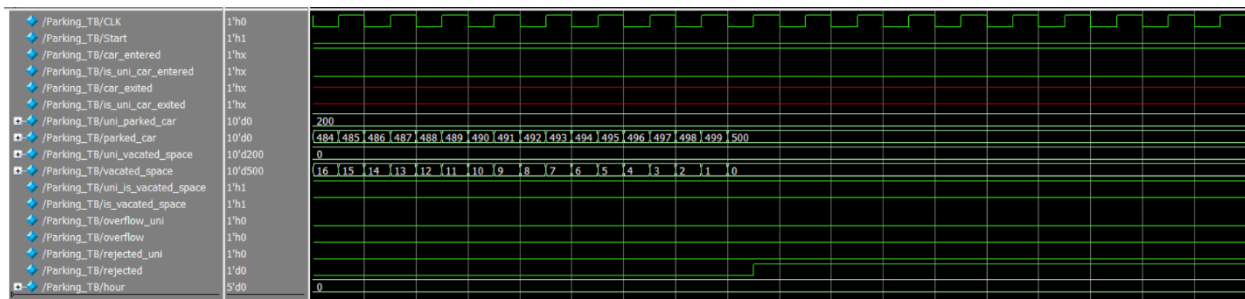
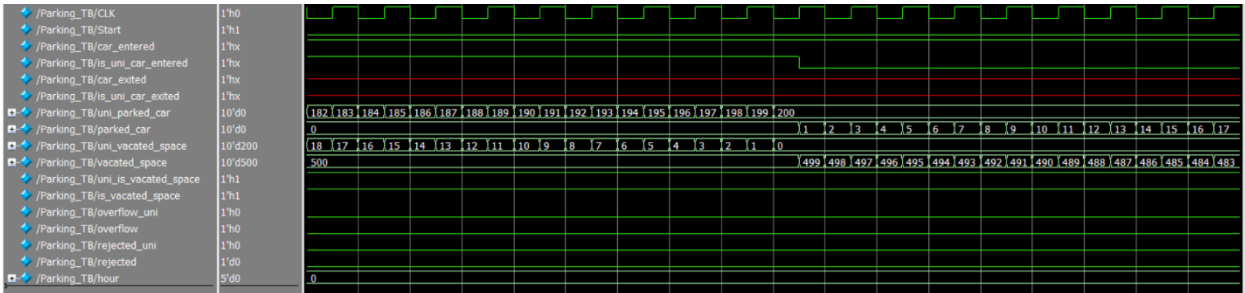
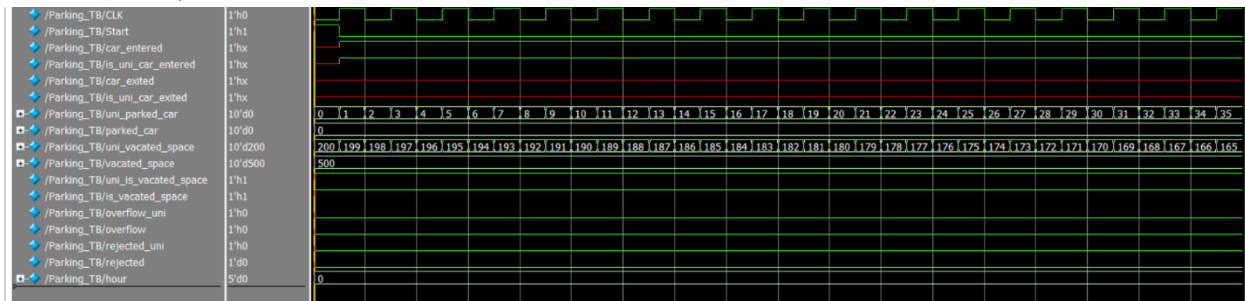
در خود کد تست بنچ به صورت کامنتریال مقادیر مورد انتظار در هر گام را مشخص کرده ام. حال تصاویر ویو فرم را قرار می‌دهم:

همانطور که از تصویر کلی ویو فرم معلوم است، شبیه سازی را برای 17 ساعت و تعداد بالا ماشین انجام داده ام که به بخش‌های مهم آن اشاره می‌کنم:

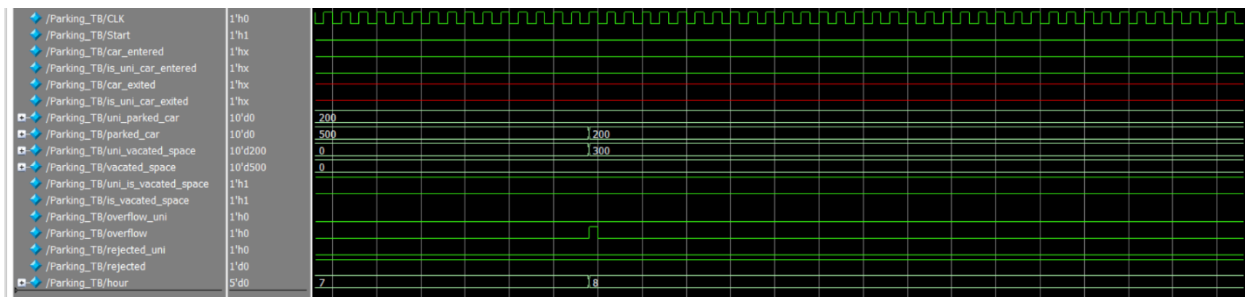


مرحله اول تست بنچ (افزودن 100 ماشین کارکنان و اساتید و 500 ماشین آزاد):

جبرانی پایانترم – سوال 8



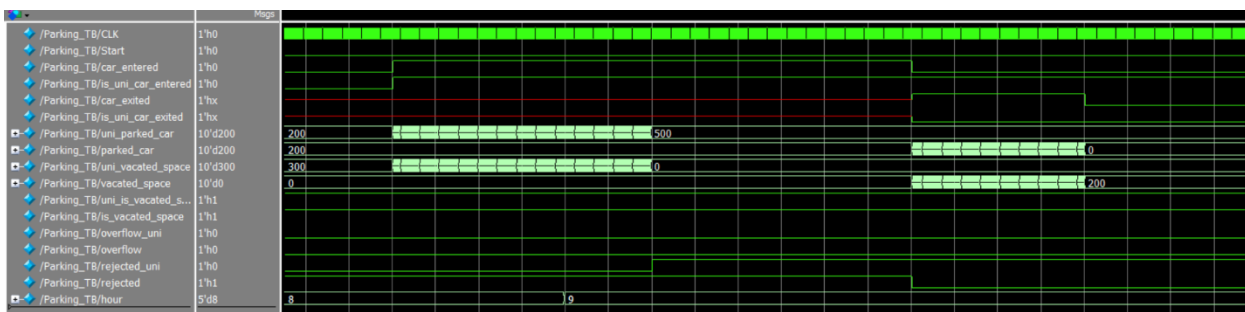
در تصویر آخر به ریجکت شدن سیگنال:



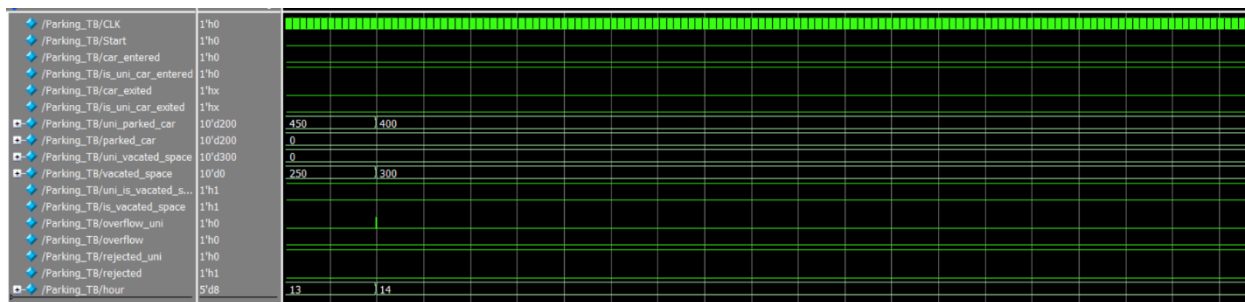
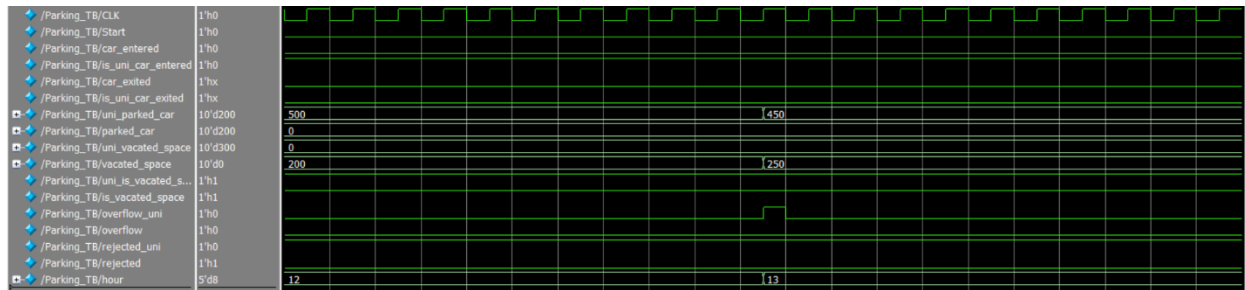
تصویر قبل نقطه بحرانی در ساعت 8 را نشان می‌دهد که اورفلو رخ می‌دهد.

در مرحله زیر، قصد داریم تعداد ماشین اساتید و کارکنان را به 500 برسانیم.

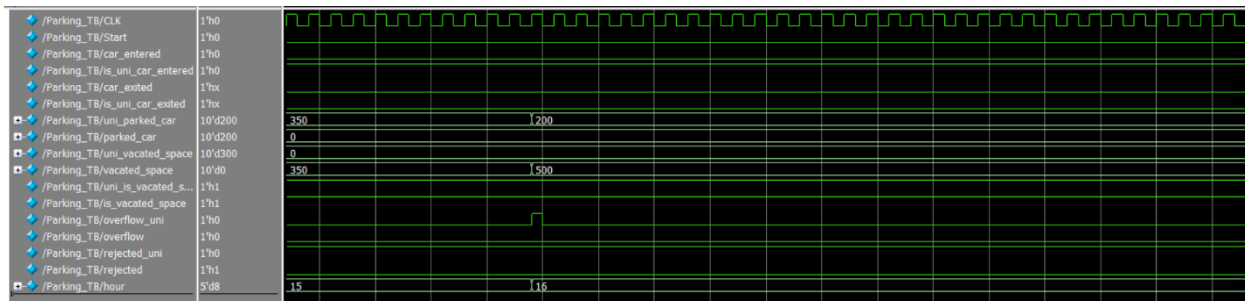
و همچنین تعداد ماشین‌های آزاد را به 0 می‌رسانیم:



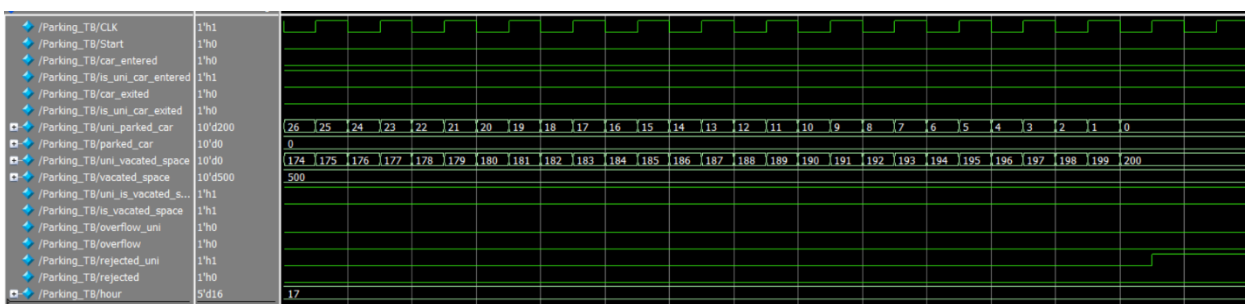
از ساعت 13 تا ساعت 16، ساعت ساعات بحرانی خواهیم بود: (ظرفیت ها 50 تا 50 تا انتقال می یابند) عکس در زیر:



و ناگهان در ساعت 16، انتقال ناگهانی ظرفیت رخ می دهد:



در نهایت هم پارکینگ را از هر ماشینی خالی می کنم:



ب) برای سنتز، از ابزار سنتز کوآرتوس استفاده کردم:

فایل حاصل سنتز این مدار در پیوست قرار گرفته است. همچنین بخش حداکثر فرکانس نیز در بخش Slow 1100mV 85C model برای Cyclone V FPGA به شکل زیر است:

	Fmax	Restricted Fmax	Clock Name	Note
1	14.81 MHz	14.81 MHz	CLK	
2	28.71 MHz	28.71 MHz	TimeHandler:th hour[0]	
3	34.01 MHz	34.01 MHz	car_entered	

که طبق آن باید کوچکترین فرکانس را به عنوان ماکسیمم فرکانس انتخاب کنیم تا تمام دستگاه بتواند همگام بماند، بنابراین، 14MHz را به عنوان فرکانس دستگاه انتخاب می‌کنم.