

کنترلر پارکینگ (سوال ۸)

یسنا نوشیروانی ۴۰۱۱۰۶۶۷۴

توضیح کد

سیگنال‌های ورودی مازول پارکینگ به شرح زیر است:

- (۱) کلاک
- (۲) reset: با فشردن این دکمه ساعت ۸ می‌شود، ظرفیت‌ها به مقدار اولیه برمی‌گردند و تعداد ماشین‌های پارک شده صفر می‌شود.
- (۳) car_entered: هنگام رها شدن این دکمه یعنی یک ماشین درخواست ورود به پارکینگ داده است. اگر جای خالی وجود نداشته باشد سیگنال خروجی ja_nist یک می‌شود و اجازه ورود به ماشین داده نمی‌شود.
- (۴) is_uni_car_entered: اگر این سیگنال هنگام رها شدن دکمه car_entered یک باشد یعنی ماشین درخواست ورود دانشگاهی فرستاده و اگر صفر باشد یعنی درخواست از نوع ورود آزاد بوده است.
- (۵) car_exited: هنگام رها شدن این دکمه یعنی یک ماشین درخواست خروج از پارکینگ داده است. اگر ماشینی در پارکینگ (از نوع مورد نظر) نباشد سیگنال خروجی faulty_exit یک می‌شود و نشان‌دهنده خروج نامعتبر است.
- (۶) is_uni_car_exited: اگر این سیگنال هنگام رها شدن دکمه car_exited یک باشد یعنی ماشینی که درخواست خروج فرستاده در جایگاه دانشگاهی پارک کرده بوده و اگر صفر باشد یعنی از نوع آزاد بوده است.

سیگنال‌های خروجی به این صورت است:

- (۱) hour: نشان‌دهنده ساعت کنونی است. (هر ساعت ۵۰۰ کلاک طول می‌کشد)
- (۲) uni_parked_car: تعداد ماشین‌های دانشگاهی پارک شده
- (۳) free_parked_car: تعداد ماشین‌های آزاد پارک شده
- (۴) uni_vacated_space: تعداد جاهای دانشگاهی خالی
- (۵) free_vacated_space: تعداد جاهای آزاد خالی
- (۶) uni_is_vacated_space: وقتی یک است که جای دانشگاهی خالی وجود داشته باشد، در غیر این صورت صفر است.
- (۷) free_is_vacated_space: وقتی یک است که جای آزاد خالی وجود داشته باشد، در غیر این صورت صفر است.
- (۸) ja_nist: این سیگنال زمانی یک می‌شود که پارکینگ جای خالی برای قبول درخواست ورود صادر شده را نداشته باشد، در غیر این صورت صفر است. (در صورتی که درخواست از نوع دانشگاهی صادر شود تنها در صورتی پذیرفته است که پارکینگ جای خالی از نوع دانشگاهی داشته باشد و درخواست خود به خود از نوع آزاد در نظر گرفته نمی‌شود)
- (۹) faulty_exit: این سیگنال زمانی یک می‌شود که ماشینی از نوع موردنظر در پارکینگ نباشد و درخواست خروج نامعتبر صادر شود، در غیر این صورت صفر است.

روش عملکرد ماژول:

هنگام رها شدن دکمه `car_entered` با توجه به نوع درخواست ورود، اگر آزاد باشد `enteredFreeCarCount` و اگر دانشگاهی باشد `enteredUniCarCount` یکی زیاد می‌شود.

هنگام رها شدن دکمه `car_exited` با توجه به نوع درخواست خروج، اگر آزاد باشد `exitedFreeCarCount` و اگر دانشگاهی باشد `exitedUniCarCount` یکی زیاد می‌شود.

این چهار متغیر نشان‌دهنده تعداد کل ماشین‌های آزاد یا دانشگاهی ورودی و خروجی از آخرین `reset` قبلی است و اختلاف آن‌ها با تعداد ماشین‌های موجود در پارکینگ برابر است.

تکه کد مربوط به این عملیات:

```
80 if (car_entered_det && !car_entered) begin
81     if (is_uni_car_entered) enteredUniCarCount = enteredUniCarCount + 1;
82     else enteredFreeCarCount = enteredFreeCarCount + 1;
83 end
84
85 if (car_exited_det && !car_exited) begin
86     if (is_uni_car_exited) exitedUniCarCount = exitedUniCarCount + 1;
87     else exitedFreeCarCount = exitedFreeCarCount + 1;
88 end
89
```

در بخش اصلی کد ابتدا چک می‌کنیم سیگنال `reset` یک است یا نه. اگر یک بود مقدار متغیرها و سیگنال‌ها را به حالت اولیه برمی‌گردانیم و اگر صفر بود سراغ ادامه کد می‌رویم.

قسمت `reset` کد:

```
20 always @(reset or clock or car_entered or car_exited) begin
21     if (reset) begin
22         hourBase8 = 5'b0;
23         counter = 0;
24         // reset
25         uni_parked_car = 10'b0;
26         free_parked_car = 10'b0;
27         uni_vacated_space = 10'd500;
28         free_vacated_space = 10'd200;
29         uni_is_vacated_space = 1'b1;
30         free_is_vacated_space = 1'b1;
31         ja_nist = 1'b0;
32         faulty_exit = 1'b0;
33         uni_capacity = 500;
34         enteredUniCarCount = 0;
35         exitedUniCarCount = 0;
36         enteredFreeCarCount = 0;
37         exitedFreeCarCount = 0;
38     end
39     else begin
```

در ادامه اگر به خاطر **posedge clock** وارد **always** شده باشیم، ساعت به روزرسانی می شود و اگر به پایان روز رسیده بودیم یعنی هشت صبح روز بعد بود، انگار دکمه **reset** فشرده شده و دقیقاً همان عملیات انجام می شوند.

همچنین در این قسمت پس از به روزرسانی ساعت، ظرفیت در آن ساعت را پیدا می کنیم و سیگنال ها را مقداردهی می کنیم تا در صورتی که اولین بار باشد که **always** در ساعت جدید اجرا می شود ظرفیت های پارکینگ به روزرسانی شوند.

همانطور که در سوال گفته شده بود، از ساعت ۱۳ تا ۱۶ ظرفیت دانشگاهی ۵۰ تا ۵۰ تا کم می شود و در ساعت ۱۶ به ۲۰۰ می رسد. (یعنی در انتهای ساعت ۱۴ ظرفیت دانشگاهی ۴۵۰، در انتهای ساعت ۱۵ برابر با ۴۰۰ و در انتهای ساعت ۱۶ برابر با ۲۰۰ است)

سپس تعداد ماشین های پارک شده بررسی می شود و اگر تعداد ماشین های دانشگاهی از ظرفیت جدید دانشگاهی بیشتر بود، آن ماشین های اضافه آزاد حساب می شوند.

کد مربوط به این قسمت:

```
39     else begin
40         // increase time at posedge of clock
41         if (clock && !clock_edge_det) begin
42             counter = (counter + 1) % CLOCKS_IN_HOUR;
43             if (counter == 0) begin
44                 hourBase8 = (hourBase8 + 1) % 24;
45                 if (hourBase8 == 0) begin
46                     // reset (day is over)
47                     uni_parked_car = 10'b0;
48                     free_parked_car = 10'b0;
49                     uni_vacated_space = 10'd500;
50                     free_vacated_space = 10'd200;
51                     uni_is_vacated_space = 1'b1;
52                     free_is_vacated_space = 1'b1;
53                     ja_nist = 1'b0;
54                     faulty_exit = 1'b0;
55                     uni_capacity = 500;
56                     enteredUniCarCount = 0;
57                     exitedUniCarCount = 0;
58                     enteredFreeCarCount = 0;
59                     exitedFreeCarCount = 0;
60                 end
61             end
62         end
63
64         // update capacity
65         if (counter == 0) begin
66             if (hourBase8 == 6)
67                 uni_capacity = 450;
68             else if (hourBase8 == 7)
69                 uni_capacity = 400;
70             else if (hourBase8 == 8)
71                 uni_capacity = 200;
72
73             // free uni cars if they don't fit
74             if (uni_parked_car > uni_capacity) begin
75                 enteredFreeCarCount = enteredFreeCarCount + (uni_parked_car - uni_capacity);
76                 enteredUniCarCount = enteredUniCarCount - (uni_parked_car - uni_capacity);
77             end
78         end
```

در بخش آخر کد چهار متغیری که در ابتدا شرح داده شد (enteredFreeCarCount, enteredUniCarCount, exitedFreeCarCount, exitedUniCarCount) در سیگنال‌های خروجی تأثیر داده می‌شوند. به این‌صورت که ابتدا سیگنال‌های خروجی ja_nist و faulty_exit صفر می‌شوند. سپس تعداد ماشین‌های دانشگاهی و آزاد در پارکینگ اگر درخواست داده شده معتبر باشد محاسبه می‌شوند. اگر این مقدار کمتر از صفر یا بیشتر از ظرفیت پارکینگ بود یعنی درخواست نامعتبر بوده و تعداد ماشین‌های داخل پارکینگ (اختلاف دو متغیر برای هر یک از حالات آزاد و دانشگاهی) باید به حالت قبلی برگردد و سیگنال‌های ja_nist و faulty_exit با توجه به خطای رخ داده مقداره‌ی می‌شوند و تا لبه بالا رونده بعدی کلاک مقدارشان ثابت می‌ماند. پس از این زمان به حالت اولیه یعنی صفر برمی‌گردند.

کد این قسمت:

```

90 // update car counts and error signals
91 ja_nist = 0;
92 faulty_exit = 0;
93 tempUni = enteredUniCarCount - exitedUniCarCount;
94 tempFree = enteredFreeCarCount - exitedFreeCarCount;
95 if (tempUni > uni_capacity) begin
96     enteredUniCarCount = exitedUniCarCount + uni_capacity;
97     ja_nist = 1;
98 end
99 else if (tempUni < 0) begin
100     exitedUniCarCount = enteredUniCarCount;
101     faulty_exit = 1;
102 end
103 else if (tempFree > TOTAL_CAPACITY - uni_capacity) begin
104     enteredFreeCarCount = exitedFreeCarCount + TOTAL_CAPACITY - uni_capacity;
105     ja_nist = 1;
106 end
107 else if (tempFree < 0) begin
108     exitedFreeCarCount = enteredFreeCarCount;
109     faulty_exit = 1;
110 end
111 uni_parked_car = enteredUniCarCount - exitedUniCarCount;
112 free_parked_car = enteredFreeCarCount - exitedFreeCarCount;
113 uni_vacated_space = uni_capacity - uni_parked_car;
114 free_vacated_space = TOTAL_CAPACITY - uni_capacity - free_parked_car;
115 uni_is_vacated_space = (uni_vacated_space != 0);
116 free_is_vacated_space = (free_vacated_space != 0);
117 end
118
119 clock_edge_det = clock;
120 car_entered_det = car_entered;
121 car_exited_det = car_exited;
122 end

```

تست‌های محک و تحلیل نتایج

تست محک TB1

این تست به منظور چک تکمیل ظرفیت و تفکیک درست دانشگاهی و آزاد طراحی شده است.

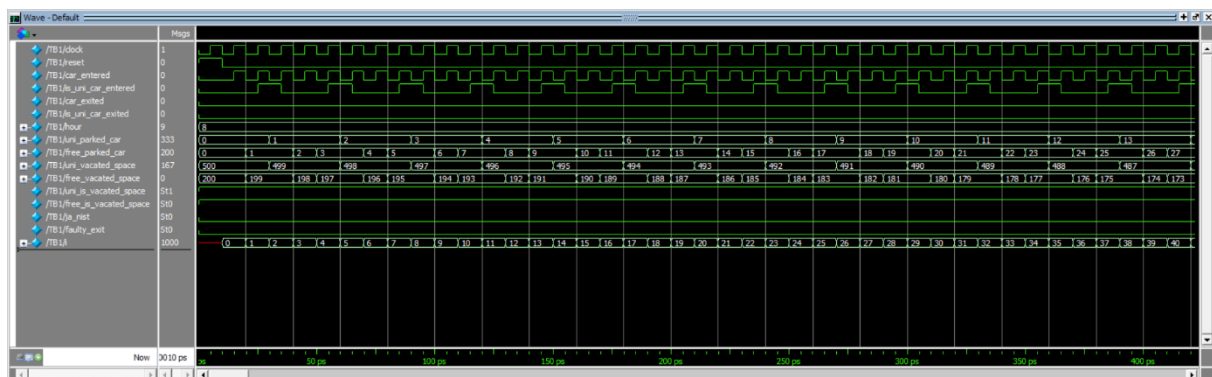
حلقه تولید سیگنال‌های ورودی به این شکل و دوره تناوب کلاک ده نانوثانیه است:

```
20 integer i;
21 initial begin
22     reset = 1;
23     car_entered = 0;
24     is_uni_car_entered = 0;
25     car_exited = 0;
26     is_uni_car_exited = 0;
27
28     #10
29     reset = 0;
30
31     for (i = 0; i < 1000; i = i + 1) begin
32         #5
33         car_entered = 1;
34         is_uni_car_entered = (i%3==1);
35         #5
36         car_entered = 0;
37     end
38
39     $stop();
40
41 end
```

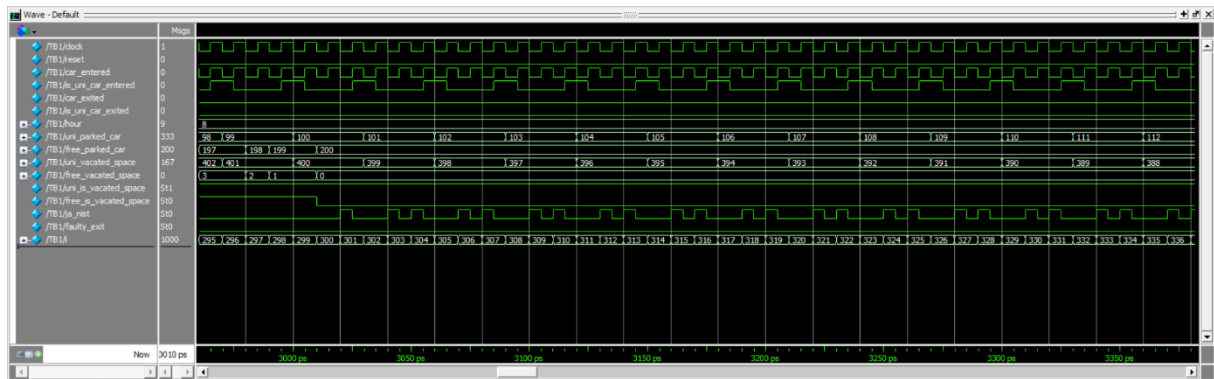
نتیجه محک مطابق انتظار است.

ابتدا مدار reset می‌شود. سپس در هر ۱۰ نانوثانیه یک ماشین وارد پارکینگ می‌شود به طوری که یکی از هر سه ماشین دانشگاهی است. پس از ۲۰۰ ماشین آزاد ظرفیت آن پر می‌شود و اجازه ورود ماشین‌های بیشتر را نمی‌دهد. به همین علت پس از مدتی سیگنال ja_nist دو کلاک از هر سه کلاک یک می‌شود. در نهایت ظرفیت پر شده دانشگاهی برابر با ۳۳۳ ماشین است. همچنین تا پایان شبیه‌سازی ساعت نه است و ظرفیت‌ها تغییری نکرده و برابر با ۵۰۰ تا دانشگاهی و ۲۰۰ تا آزاد است.

waveform ها در ابتدای شبیه‌سازی که هر دو قسمت دانشگاهی و آزاد جا دارند به این شکل است:



پس از پر شدن ظرفیت آزاد، waveform به این شکل ادامه می‌یابد:



و پس از اتمام حلقه و ارسال سیگنال‌ها مقادیر خروجی به این ترتیبند:

```
10005) hour = 9 || uni por = 333, khali = 167, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
```

تست محک TBFreeCapacityIncrease

این تست برای چک افزایش ظرفیت آزاد پارکینگ پس از ساعت ۱۳ است.

در این تست دوره تناوب کلاک دو نانو ثانیه در نظر گرفته شده است و کد قسمت اصلی آن به این شکل است:

```
25 integer i;
26 initial begin
27     reset = 1;
28     car_entered = 0;
29     is_uni_car_entered = 0;
30     car_exited = 0;
31     is_uni_car_exited = 0;
32
33     #2
34     reset = 0;
35
36     for (i = 0; i < 1000; i = i + 1) begin
37         #9
38         car_entered = 1;
39         is_uni_car_entered = i%3;
40         #1
41         car_entered = 0;
42     end
43
44     $stop();
45
46 end
```

افزایش ظرفیت آزاد را پس از ساعت ۱۳ مشاهده می‌کنیم:

```
5943) hour = 13 || uni por = 198, khali = 302, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
5952) hour = 13 || uni por = 198, khali = 302, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 1, exitna = 0
5953) hour = 13 || uni por = 198, khali = 302, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
5962) hour = 13 || uni por = 199, khali = 301, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
5972) hour = 13 || uni por = 199, khali = 301, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 1, exitna = 0
5973) hour = 13 || uni por = 199, khali = 301, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
5982) hour = 13 || uni por = 199, khali = 301, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 1, exitna = 0
5983) hour = 13 || uni por = 199, khali = 301, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
5992) hour = 13 || uni por = 200, khali = 300, iskhali = 1 | free por = 200, khali = 0, iskhali = 0 | janist = 0, exitna = 0
6001) hour = 14 || uni por = 200, khali = 250, iskhali = 1 | free por = 200, khali = 50, iskhali = 1 | janist = 0, exitna = 0
6002) hour = 14 || uni por = 200, khali = 250, iskhali = 1 | free por = 201, khali = 49, iskhali = 1 | janist = 0, exitna = 0
6012) hour = 14 || uni por = 200, khali = 250, iskhali = 1 | free por = 202, khali = 48, iskhali = 1 | janist = 0, exitna = 0
6022) hour = 14 || uni por = 201, khali = 249, iskhali = 1 | free por = 202, khali = 48, iskhali = 1 | janist = 0, exitna = 0
6032) hour = 14 || uni por = 201, khali = 249, iskhali = 1 | free por = 203, khali = 47, iskhali = 1 | janist = 0, exitna = 0
6042) hour = 14 || uni por = 201, khali = 249, iskhali = 1 | free por = 204, khali = 46, iskhali = 1 | janist = 0, exitna = 0
6052) hour = 14 || uni por = 202, khali = 248, iskhali = 1 | free por = 204, khali = 46, iskhali = 1 | janist = 0, exitna = 0
6062) hour = 14 || uni por = 202, khali = 248, iskhali = 1 | free por = 205, khali = 45, iskhali = 1 | janist = 0, exitna = 0
6072) hour = 14 || uni por = 202, khali = 248, iskhali = 1 | free por = 206, khali = 44, iskhali = 1 | janist = 0, exitna = 0
6082) hour = 14 || uni por = 203, khali = 247, iskhali = 1 | free por = 206, khali = 44, iskhali = 1 | janist = 0, exitna = 0
```

پس از ساعت ۱۴:

```
6942) hour = 14 || uni por = 231, khali = 219, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 1, exitna = 0
6943) hour = 14 || uni por = 231, khali = 219, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 0, exitna = 0
6952) hour = 14 || uni por = 232, khali = 218, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 0, exitna = 0
6962) hour = 14 || uni por = 232, khali = 218, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 1, exitna = 0
6963) hour = 14 || uni por = 232, khali = 218, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 0, exitna = 0
6972) hour = 14 || uni por = 232, khali = 218, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 1, exitna = 0
6973) hour = 14 || uni por = 232, khali = 218, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 0, exitna = 0
6982) hour = 14 || uni por = 233, khali = 217, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 0, exitna = 0
6992) hour = 14 || uni por = 233, khali = 217, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 1, exitna = 0
6993) hour = 14 || uni por = 233, khali = 217, iskhali = 1 | free por = 250, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7001) hour = 15 || uni por = 233, khali = 167, iskhali = 1 | free por = 250, khali = 50, iskhali = 1 | janist = 0, exitna = 0
7002) hour = 15 || uni por = 233, khali = 167, iskhali = 1 | free por = 251, khali = 49, iskhali = 1 | janist = 0, exitna = 0
7012) hour = 15 || uni por = 234, khali = 166, iskhali = 1 | free por = 251, khali = 49, iskhali = 1 | janist = 0, exitna = 0
7022) hour = 15 || uni por = 234, khali = 166, iskhali = 1 | free por = 252, khali = 48, iskhali = 1 | janist = 0, exitna = 0
7032) hour = 15 || uni por = 234, khali = 166, iskhali = 1 | free por = 253, khali = 47, iskhali = 1 | janist = 0, exitna = 0
7042) hour = 15 || uni por = 235, khali = 165, iskhali = 1 | free por = 253, khali = 47, iskhali = 1 | janist = 0, exitna = 0
7052) hour = 15 || uni por = 235, khali = 165, iskhali = 1 | free por = 254, khali = 46, iskhali = 1 | janist = 0, exitna = 0
7062) hour = 15 || uni por = 235, khali = 165, iskhali = 1 | free por = 255, khali = 45, iskhali = 1 | janist = 0, exitna = 0
7072) hour = 15 || uni por = 236, khali = 164, iskhali = 1 | free por = 255, khali = 45, iskhali = 1 | janist = 0, exitna = 0
7082) hour = 15 || uni por = 236, khali = 164, iskhali = 1 | free por = 256, khali = 44, iskhali = 1 | janist = 0, exitna = 0
7092) hour = 15 || uni por = 236, khali = 164, iskhali = 1 | free por = 257, khali = 43, iskhali = 1 | janist = 0, exitna = 0
```

پس از ساعت ۱۵:

```
7932) hour = 15 || uni por = 264, khali = 136, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 1, exitna = 0
7933) hour = 15 || uni por = 264, khali = 136, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7942) hour = 15 || uni por = 265, khali = 135, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7952) hour = 15 || uni por = 265, khali = 135, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 1, exitna = 0
7953) hour = 15 || uni por = 265, khali = 135, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7962) hour = 15 || uni por = 265, khali = 135, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 1, exitna = 0
7963) hour = 15 || uni por = 265, khali = 135, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7972) hour = 15 || uni por = 266, khali = 134, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7982) hour = 15 || uni por = 266, khali = 134, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 1, exitna = 0
7983) hour = 15 || uni por = 266, khali = 134, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
7992) hour = 15 || uni por = 266, khali = 134, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 1, exitna = 0
7993) hour = 15 || uni por = 266, khali = 134, iskhali = 1 | free por = 300, khali = 0, iskhali = 0 | janist = 0, exitna = 0
8001) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 366, khali = 134, iskhali = 1 | janist = 0, exitna = 0
8002) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 366, khali = 134, iskhali = 1 | janist = 1, exitna = 0
8003) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 366, khali = 134, iskhali = 1 | janist = 0, exitna = 0
8012) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 367, khali = 133, iskhali = 1 | janist = 0, exitna = 0
8022) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 368, khali = 132, iskhali = 1 | janist = 0, exitna = 0
8032) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 368, khali = 132, iskhali = 1 | janist = 1, exitna = 0
8033) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 368, khali = 132, iskhali = 1 | janist = 0, exitna = 0
8042) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 369, khali = 131, iskhali = 1 | janist = 0, exitna = 0
8052) hour = 16 || uni por = 200, khali = 0, iskhali = 0 | free por = 370, khali = 130, iskhali = 1 | janist = 0, exitna = 0
```

از ساعت سه به چهار ظرفیت آزاد ۲۰۰ تا اضافه و ظرفیت دانشگاهی ۲۰۰ تا کم می‌شود. ابتدا جاهای خالی دانشگاهی به ظرفیت آزاد اضافه می‌شوند اما اگر تعدادشان کافی نبود، تعدادی از فضاهای پر دانشگاهی هم به نوع آزاد پر تبدیل می‌شوند.

تست محک خروج از پارکینگ

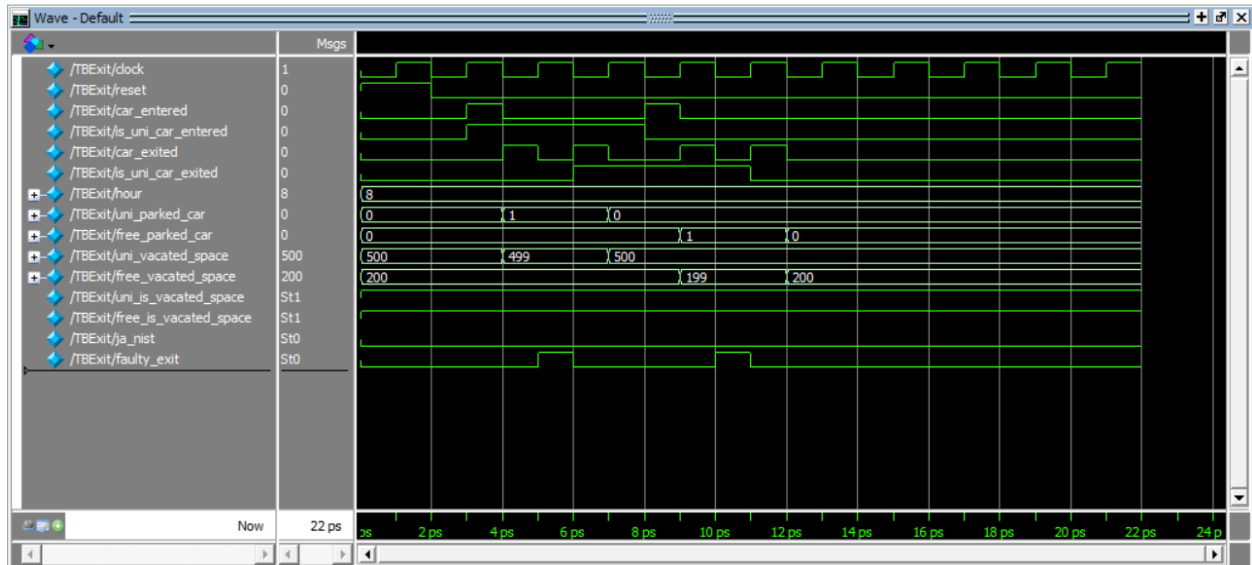
در این تست سیگنال `faulty_exit` را بررسی می‌کنیم.

ابتدا یک ماشین دانشگاهی وارد پارکینگ می‌شود و درخواست خروج ماشین آزاد را می‌دهیم. همانطور که انتظار داریم سیگنال `faulty_exit` یک می‌شود. سپس ماشین دانشگاهی را خارج می‌کنیم. حالا همین کار را با ماشین آزاد به جای دانشگاهی انجام می‌دهیم.

دوره تناوب کلاک دو نانو ثانیه است و قسمت اصلی برنامه محک به این صورت است:

```
25 initial begin
26     reset = 1;
27     car_entered = 0;
28     is_uni_car_entered = 0;
29     car_exited = 0;
30     is_uni_car_exited = 0;
31
32     #2
33     reset = 0;
34
35     // test for uni car in parking, but free car exits
36     #1
37     car_entered = 1;
38     is_uni_car_entered = 1;
39
40     #1
41     car_entered = 0;
42     car_exited = 1;
43     is_uni_car_exited = 0;
44
45     #1
46     car_exited = 0;
47
48     #1
49     car_exited = 1;
50     is_uni_car_exited = 1;
51
52     #1
53     car_exited = 0;
54
55     // test for free car in parking, but uni car exits
56     #1
57     car_entered = 1;
58     is_uni_car_entered = 0;
59
60     #1
61     car_entered = 0;
62     car_exited = 1;
63     is_uni_car_exited = 1;
64
65     #1
66     car_exited = 0;
67
68     #1
69     car_exited = 1;
70     is_uni_car_exited = 0;
71
72     #1
73     car_exited = 0;
74
75     #10
76     $stop();
77 end
```

نتایج این تست مطابق زیر است:



سیگنال faulty_exit با لبه پایین رونده car_exited نامعتبر یک می شود و در لبه بالا رونده بعدی کلاک صفر و reset می شود.

گزارش سنتز

با استفاده از نرم افزار کوارتوس کد را برای Cyclone IV GX سنتز می کنیم. گزارش سنتز به این شکل است:

Slow 1100mV 85C Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	85.32 MHz	85.32 MHz	enteredUniCarCount[0]	

برای تخمین حداکثر فرکانس مدار می توان setup time و hold time را بررسی کرد. چون هر کلاک باید حداقل به اندازه این دو طول بکشد تا سیگنال ها را تشخیص دهد.

Slow 1100mV 85C Model Hold Summary			
	Clock	Slack	End Point TNS
1	enteredUniCarCount[0]	-2.742	-163.216

Slow 1100mV 85C Model Setup Summary			
	Clock	Slack	End Point TNS
1	enteredUniCarCount[0]	-8.721	-752.208

$$freq = \frac{1}{(2.742 + 8.721) \times 10^{-9}} = 87.24 \text{ MHz}$$

همانطور که انتظار داشتیم فرکانس به دست آمده در روش دوم به فرکانس گزارش شده توسط ابزار سنتز بسیار نزدیک ولی کمی بیشتر است.