

به نام خدا



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

جبرانی پایان ترم

استاد: مهندس امین فصحتی

نام و نام خانوادگی: اشکان تارپوردی

شماره دانشجویی: ۴۰۱۱۰۵۷۵۳

سوال ۸

پاسخ: با توجه مفروضات ذکر شده در مسئله، طراحی وریلاگ آن به صورت زیر است:

```

1  module ParkingLot #(
2      parameter UNI_CAP = 500,
3      parameter PUB_CAP = 200,
4      parameter PHASE = 50
5  )
6      input wire car_entered,
7      input wire is_uni_car_entered,
8      input wire car_exited,
9      input wire is_uni_car_exited,
10     input wire [5:0] hour,
11     output reg signed [9:0] uni_parked_car = 0,
12     output reg signed [9:0] parked_car = 0,
13     output reg signed [9:0] uni_vacated_space = UNI_CAP,
14     output reg signed [9:0] vacated_space = PUB_CAP,
15     output wire uni_is_vacated_space,
16     output wire is_vacated_space
17 );
18
19 reg [9:0] uni_entire_space;
20 reg [9:0] entire_space;
21
22
23 task update_spaces;
24     input [5:0] hour;
25     begin
26         if (hour >= 13 && hour < 16) begin
27             uni_entire_space = UNI_CAP - (hour - 12) * PHASE;
28             entire_space = PUB_CAP + (hour - 12) * PHASE;
29         end else begin
30             uni_entire_space = PUB_CAP;
31             entire_space = UNI_CAP;
32         end
33     end
34 endtask
35
36 task handle_uni_overflow;
37     begin
38         if (uni_entire_space < uni_parked_car) begin
39             parked_car = parked_car + (uni_parked_car - uni_entire_space);
40             uni_parked_car = uni_entire_space;
41             vacated_space = entire_space - parked_car;
42         end
43     end
44 endtask
45
46 task handle_car_enter;
47     input is_uni;
48     begin
49         if (is_uni) begin
50             if (uni_is_vacated_space) begin
51                 uni_vacated_space = uni_vacated_space - 1;
52                 uni_parked_car = uni_parked_car + 1;
53             end
54             else if (is_vacated_space) begin
55                 vacated_space = vacated_space - 1;
56                 parked_car = parked_car + 1;
57             end
58         end else if (is_vacated_space) begin
59             vacated_space = vacated_space - 1;
60             parked_car = parked_car + 1;
61         end
62     end
63 endtask
64

```

```

65 task handle_car_exit;
66     input is_uni;
67     begin
68         if (is_uni && uni_parked_car > 0) begin
69             uni_parked_car = uni_parked_car - 1;
70             uni_vacated_space = uni_vacated_space + 1;
71         end else if (!is_uni && parked_car > 0) begin
72             parked_car = parked_car - 1;
73             vacated_space = vacated_space + 1;
74         end
75     end
76 endtask
77
78 always @(posedge car_entered or posedge car_exited) begin
79     update_spaces(hour);
80
81     if ((hour >= 13 && hour < 16) || hour >= 16) begin
82         handle_uni_overflow();
83     end
84
85     if (car_entered && hour >= 8) begin
86         handle_car_enter(is_uni_car_entered);
87     end
88
89     if (car_exited && hour >= 8) begin
90         handle_car_exit(is_uni_car_exited);
91     end
92 end
93
94 assign uni_is_vacated_space = (uni_vacated_space > 0);
95 assign is_vacated_space = (vacated_space > 0);
96
97 endmodule

```

همانطور که مشاهده می‌کنید، ابتدا ورودی و خروجی‌های ماژول را مشخص می‌کنیم. نخست پارامترهای ماژول که مرتبط با ظرفیت پارکینگ می‌باشد را مشخص می‌کنیم. سپس ورودی‌های مدار که شامل سیگنال‌های ورود و خروج خودرو و همچنین سیگنال مرتبط با تعلق یا عدم تعلق خودرو وارد یا خارج شده به دانشگاه است را مشخص می‌کنیم. در نهایت نیز خروجی‌های ماژول مانند تعداد خودروهای پارک شده متعلق به دانشگاه، تعداد خودروهای پارک شده متعلق به ظرفیت آزاد و ... را مشخص می‌کنیم.

سپس برای افزایش خوانایی کد و ریلگ، از task استفاده می‌کنیم که همانند تابع در سایر زبان‌های برنامه نویسی است. تسک update_spaces، ظرفیت پارکینگ متعلق به دانشگاه و ظرفیت پارکینگ متعلق به آزاد را با توجه به ساعت، مقداردهی می‌کند. هنگامی که ساعت از ۱۳ می‌گذرد و ظرفیت پارکینگ مربوط به آزاد افزایش می‌یابد، در صورتی که ظرفیت پارکینگ مربوط به دانشگاه پر باشد، باید این ظرفیت به ظرفیت پارکینگ آزاد منتقل شود؛ یعنی خودرو در پارکینگ دانشگاه با افزایش ظرفیت پارکینگ آزاد، تعلق به پارکینگ آزاد پیدا می‌کند در این صورت ظرفیت پارکینگ دانشگاه برابر مقدار جدید خود است و ظرفیت پارکینگ آزاد نیز آپدیت می‌شود. تسک handle_uni_overflow این امر را انجام می‌دهد. تسک handle_car_enter، ظرفیت پارکینگ دانشگاه و آزاد و تعداد خودروهای پارک شده دانشگاه و آزاد را با توجه به سیگنال ورودی تعلق یا عدم تعلق خودرو وارد شده به دانشگاه و وجود فضای خالی در پارکینگ، آپدیت می‌کند. تسک handle_car_exit نیز مشابه تسک handle_car_enter، ظرفیت و تعداد خودروهای پارک شده را به کمک سیگنال ورودی تعلق یا عدم تعلق خودرو خارج شده به دانشگاه و خالی نبودن پارکینگ دانشگاه و آزاد، آپدیت می‌کند.

در نهایت، در بلاک always با لبه بالارونده سیگنال ورودی car_entered و لبه بالارونده سیگنال ورودی car_exited با بررسی شروط لازم، تسک‌های مورد نظر را صدا می‌زنیم. همچنین برای مقداردهی دو خروجی uni_is_vacated_space و vacated_space از دستور assign استفاده می‌کنیم چرا که در هر زمان که مقادیر uni_vacted_space و vacated_space دچار تغییر شدند، این دو مقدار نیز آپدیت شده و منتظر تغییر سیگنال‌های دیگر نمانند.

اکنون برای صحت‌سنجی ماژول طراحی شده، یک تست‌بنچ برای آن طراحی می‌کنیم و خروجی آن را مورد بررسی قرار می‌دهیم. تست‌بنچ طراحی شده به صورت زیر است:

```

1  module TB();
2      parameter UNI_CAP = 5;
3      parameter PUB_CAP = 2;
4      parameter PHASE = 1;
5
6      reg car_entered = 0;
7      reg is_uni_car_entered = 0;
8      reg car_exited = 0;
9      reg is_uni_car_exited = 0;
10     reg [5:0] hour;
11
12     wire [9:0] uni_parked_car;
13     wire [9:0] parked_car;
14     wire [9:0] uni_vacated_space;
15     wire [9:0] vacated_space;
16     wire uni_is_vacated_space;
17     wire is_vacated_space, error;
18
19     ParkingLot #(
20         .UNI_CAP(UNI_CAP),
21         .PUB_CAP(PUB_CAP),
22         .PHASE(PHASE)
23     ) uut (
24         .car_entered(car_entered),
25         .is_uni_car_entered(is_uni_car_entered),
26         .car_exited(car_exited),
27         .is_uni_car_exited(is_uni_car_exited),
28         .hour(hour),
29         .uni_parked_car(uni_parked_car),
30         .parked_car(parked_car),
31         .uni_vacated_space(uni_vacated_space),
32         .vacated_space(vacated_space),
33         .uni_is_vacated_space(uni_is_vacated_space),
34         .is_vacated_space(is_vacated_space)
35     );
36
37     initial begin
38         car_entered = 0;
39         car_exited = 0;
40         #480
41         repeat(32) begin
42             hour = $time / 60;
43             if (hour <= 13) begin
44                 car_entered = 1;
45                 car_exited = 0;
46             end
47             else begin
48                 car_exited = $urandom_range(0, 1);
49                 car_entered = ~car_exited;
50             end
51
52             is_uni_car_entered = car_entered & $urandom_range(0, 1);
53             is_uni_car_exited = car_exited & $urandom_range(0, 1);
54
55             #0
56             $display("\nhour= %02d:%02d | car_entered= %0b | is_uni_car_entered= %0b | car_exited= %0b |",
57                 is_uni_car_exited= %0b | uni_vacated_space= %0d | vacated_space= %0d |",
58                 uni_is_vacated_space= %0b | is_vacated_space= %0b | uni_parked_car= %0d | parked_car= %0d",
59                 hour, $time % 60, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, uni_vacated_space,
60                 vacated_space, uni_is_vacated_space, is_vacated_space, uni_parked_car, parked_car);
61
62             car_entered = 0;
63             car_exited = 0;
64             #30;
65         end
66     end
67
68 endmodule

```


همانطور که مشاهده می‌کنید، ابتدا ورودی و خروجی‌های لازم را تعیین کرده و سپس از ماژول ParkingLot نمونه‌گیری یا instantiate می‌کنیم و ورودی‌ها و خروجی‌های لازم را به آن می‌دهیم. سپس در بلاک initial مقدار اولیه دو متغیر car_entered و car_exited را برابر صفر قرار می‌دهیم. سپس از دستور repeat استفاده می‌کنیم؛ در این بلاک، ابتدا تا قبل از ساعت ۱۳، مقدار سیگنال ورود خودرو را همواره یک قرار می‌دهیم تا پارکینگ به صورت کامل پر شود و شروط داده شده مشهودتر باشد. پس از ساعت ۱۳ سیگنال ورود خودرو به صورت تصادفی با استفاده از دستور urandom مقداردهی می‌کنیم. در نهایت نیز سیگنال‌های ورودی و خروجی را با استفاده از دستور display، در طول زمان مشاهده می‌کنیم.

اکنون به حالت شبیه‌سازی می‌رویم و ماژول و تست‌بنچ طراحی شده را شبیه‌سازی می‌کنیم. خروجی تست‌بنچ به صورت زیر است:

```
# hour= 08:00 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 4 | vacated_space= 2 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 0
# hour= 08:30 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 4 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 1
# hour= 09:00 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 3 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 2 | parked_car= 1
# hour= 09:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 2 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 3 | parked_car= 1
# hour= 10:00 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 4 | parked_car= 1
# hour= 10:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 1 | uni_is_vacated_space= 0 | is_vacated_space= 1 | uni_parked_car= 5 | parked_car= 1
# hour= 11:00 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 5 | parked_car= 2
# hour= 11:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 5 | parked_car= 2
# hour= 12:00 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 5 | parked_car= 2
# hour= 12:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 5 | parked_car= 2
# hour= 13:00 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 4 | parked_car= 3
# hour= 13:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 4 | parked_car= 3
# hour= 14:00 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 1 | uni_is_vacated_space= 0 | is_vacated_space= 1 | uni_parked_car= 3 | parked_car= 3
# hour= 14:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 3 | parked_car= 4
# hour= 15:00 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 1 | uni_vacated_space= 1 | vacated_space= 0 | uni_is_vacated_space= 1 | is_vacated_space= 0 | uni_parked_car= 1 | parked_car= 5
# hour= 15:30 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 1 | uni_vacated_space= 2 | vacated_space= 0 | uni_is_vacated_space= 1 | is_vacated_space= 0 | uni_parked_car= 0 | parked_car= 5
# hour= 16:00 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 2 | vacated_space= 0 | uni_is_vacated_space= 1 | is_vacated_space= 0 | uni_parked_car= 0 | parked_car= 5
# hour= 16:30 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 0 | uni_vacated_space= 2 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 0 | parked_car= 4
# hour= 17:00 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 0 | uni_vacated_space= 2 | vacated_space= 2 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 0 | parked_car= 3
# hour= 17:30 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 1 | uni_vacated_space= 2 | vacated_space= 2 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 0 | parked_car= 3
# hour= 18:00 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 2 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 3
# hour= 18:30 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 3 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 2
# hour= 19:00 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 2 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 3
# hour= 19:30 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 4
# hour= 20:00 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 0 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 5
# hour= 20:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 2 | parked_car= 5
# hour= 21:00 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 2 | parked_car= 5
# hour= 21:30 | car_entered= 1 | is_uni_car_entered= 0 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 2 | parked_car= 5
# hour= 22:00 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 1 | uni_is_vacated_space= 0 | is_vacated_space= 1 | uni_parked_car= 2 | parked_car= 4
# hour= 22:30 | car_entered= 1 | is_uni_car_entered= 1 | car_exited= 0 | is_uni_car_exited= 0 | uni_vacated_space= 0 | vacated_space= 0 | uni_is_vacated_space= 0 | is_vacated_space= 0 | uni_parked_car= 2 | parked_car= 5
# hour= 23:00 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 1 | uni_vacated_space= 1 | vacated_space= 0 | uni_is_vacated_space= 1 | is_vacated_space= 0 | uni_parked_car= 1 | parked_car= 5
# hour= 23:30 | car_entered= 0 | is_uni_car_entered= 0 | car_exited= 1 | is_uni_car_exited= 0 | uni_vacated_space= 1 | vacated_space= 1 | uni_is_vacated_space= 1 | is_vacated_space= 1 | uni_parked_car= 1 | parked_car= 4
```

با توجه به خروجی فوق، نکات آن را بصورت جداگانه بررسی می‌کنیم.

باکس قرمز

ابتدا پارکینگ خالی است و هیچ خودرویی در آن قرار ندارد پس ظرفیت هر بخش برابر بیشینه خود است. با گذر زمان، پارکینگ به مرور زمان پر می‌شود. در ساعت ۱۰:۳۰، هنگامی که خودرو مربوط به دانشگاه وارد می‌شود، ظرفیت پارکینگ دانشگاه پر می‌شود اما ظرفیت پارکینگ آزاد همچنان پر نشده است بنابراین اگر در زمان بعدی یک خودرو متعلق به دانشگاه وارد شود، بایستی آن خودرو به محل پارکینگ آزاد منتقل شود. همانطور که مشاهده می‌کنید در ساعت ۱۱:۰۰ این اتفاق افتاده است؛ با ورودی یک خودرو دانشگاهی، ظرفیت پارکینگ آزاد پر شده است و آن خودرو به پارکینگ آزاد منتقل شده است.

باکس سبز

می‌دانیم با گذر از ساعت ۱۳:۰۰، به ازای هر یک ساعت، ظرفیت پارکینگ مربوط به آزاد به میزان PHASE افزایش می‌یابد و به همین ترتیب ظرفیت پارکینگ مربوط به دانشگاه نیز به اندازه PHASE، کاهش می‌یابد. هنگامی که پارکینگ پر است، افزایش ظرفیت تنها می‌تواند در تغییر تعداد خودروهای دانشگاهی و خودروهای آزاد نمایش یابد؛ یعنی اگر برای مثال ۵ خودرو دانشگاهی و ۲ خودرو آزاد پارک شده است، با گذر از ساعت ۱۳:۰۰ و یک PHASE جلو رفتن، باید تعداد خودروهای دانشگاهی پارک شده به ۴ و تعداد خودروهای آزاد به ۳ تغییر یابد. همانطور که طبق خروجی مشاهده می‌کنید، در ساعت ۱۳:۰۰ این اتفاق افتاده است و با گذشتن هر یک ساعت، این اتفاق تکرار می‌شود.

باکس زرد

در این قسمت می‌خواهیم نشان دهیم که هنگامی که سیگنال خروج یک خودرو فعال شود اما خودرو پارک شده‌ای در پارکینگ نباشد، اتفاق نمی‌افتد و تاثیری در روند برنامه ما ندارد. در ساعت ۱۷:۳۰ این اتفاق افتاده است که سیگنال خروج خودرو متعلق به دانشگاه فعال شده است در حالیکه هیچ خودرو دانشگاهی در پارکینگ نیست. همانطور که مشاهده می‌کنید اتفاق نیافتاده است.

سایر موارد را نیز می‌توان از خروجی تست‌بنچ به سادگی استنباط کرد و از صحت عملکرد آن اطمینان کامل حاصل نمود. خروجی waveform آن نیز به صورت زیر است:

