طراحی سیستم‌های دیجیتال

پروژه جبرانی پایانترم

امیدرضا معصومی ۴۰۱۱۰۶۵۲۲

سوال انتخابی سوال ۸ میانترم است.

برای پیاده سازی از دو ماژول parking و چندین tb که همان testbench ها هستند استفاده کردم که کدهای آن‌ها را پایین تر آوردم.
ورودی‌ها و خروجی‌های ماژول parking همان ورودی و خروجی‌های صورت سوال با آن عملکرد هستند به علاوه reset و clk که کلاک است. همچنین از ساعت ۱۳ تا ۱۶ هر ساعت ۵۰ تا ظرفیت افزایش می یابد که نهایتا به ۴۰۰ میرسد. همچنین در صورت پر بودن پارکینگ و فعال شدن سیگنال ورود تغییری در اعداد اتفاق نمی‌افتد.

ماژول parking به این شکل است:

```
parking.v
1   module parking(input car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk, reset,
2   output reg [15:0] uni_parked_car, reg [15:0] parked_cars,
3   reg [15:0] uni_vacated_space, reg [15:0] vacated_space, reg uni_is_vacated_space, reg is_vacated_space);
4
5   parameter space = 700;
6   reg [15:0] uni_space, not_uni_space, hour;
7
8   initial begin
9       uni_space = 500;
10      not_uni_space = 200;
11      hour = 8;
12      uni_parked_car = 0;
13      parked_cars = 0;
14      uni_vacated_space = 500;
15      vacated_space = 200;
16      uni_is_vacated_space = 1;
17      is_vacated_space = 1;
18  end
19
20  always @(posedge clk, posedge reset) begin
21      if (reset) begin
22          uni_space = 500;
23          not_uni_space = 200;
24          hour = 8;
25          uni_parked_car = 0;
26          parked_cars = 0;
27          uni_vacated_space = 500;
28      end
29      else begin
30          hour = hour + 1;
31          if (hour >= 13 && hour <= 16) begin
32              uni_space = uni_space - 50;
33              not_uni_space = not_uni_space + 50;
34              vacated_space = vacated_space + 50;
35              uni_vacated_space = uni_vacated_space - 50;
36              if (uni_vacated_space < 0) begin
37                  uni_parked_car = uni_space;
```

```
36              if (uni_vacated_space < 0) begin
37                  uni_parked_car = uni_space;
38                  uni_vacated_space = 0;
39                  uni_is_vacated_space = 0;
40              end
41          end
42      end
43  end
44
45  always @(posedge car_entered, posedge car_exited) begin
46      if (car_entered) begin
47          if (is_uni_car_entered) begin
48              if (uni_is_vacated_space) begin
49                  uni_parked_car = uni_parked_car + 1;
50                  uni_vacated_space = uni_vacated_space - 1;
51              end
52              if (uni_vacated_space == 0)
53                  uni_is_vacated_space = 0;
54          end
55          else begin
56              if (is_vacated_space) begin
57                  parked_cars = parked_cars + 1;
58                  vacated_space = vacated_space - 1;
59              end
60              if (vacated_space == 0)
61                  is_vacated_space = 0;
62          end
63      end
64      else begin
65          if (is_uni_car_exited) begin
66              uni_vacated_space = uni_vacated_space + 1;
67              uni_parked_car = uni_parked_car - 1;
68              if (uni_is_vacated_space == 0)
```

```
66              uni_vacated_space = uni_vacated_space + 1;
67              uni_parked_car = uni_parked_car - 1;
68              if (uni_is_vacated_space == 0)
69                  uni_is_vacated_space = 1;
70          end
71          else begin
72              vacated_space = vacated_space + 1;
73              parked_cars = parked_cars - 1;
74              if (is_vacated_space == 0)
75                  is_vacated_space = 1;
76          end
77      end
78  end
79
80  endmodule
```

ماژول tb:

```
module tb;

reg car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk, reset;
wire [15:0] uni_parked_car;
wire [15:0] parked_cars;
wire [15:0] uni_vacated_space;
wire [15:0] vacated_space;
wire uni_is_vacated_space, is_vacated_space;

parking parking (car_entered, is_uni_car_entered, car_exited, is_uni_car_exited, clk, reset,
uni_parked_car, parked_cars, uni_vacated_space, vacated_space, uni_is_vacated_space, is_vacated_space);

always begin
    #50;
    clk = ~clk;
    if (clk == 1)
        $display("time: %d, uni_cars: %d, uni_space: %d, free_cars: %d, empty_space: %d",
        parking.hour, uni_parked_car, parked_cars, uni_vacated_space, vacated_space);
end

integer i;
initial begin
    clk = 0;
    #50;
    reset = 0;
    for (i = 0; i < 50; i = i + 1) begin
        car_entered = 1;
        is_uni_car_entered = 1;
        #1
        car_entered = 0;
        is_uni_car_entered = 0;
        #1
        car_entered = 1;
        #1
        car_entered = 0;
        #1;
    end
        #1;
    end
    for (i = 0; i < 50; i = i + 1) begin
        car_exited = 1;
        is_uni_car_exited = 1;
        #1
        car_exited = 0;
        #1;
    end
    is_uni_car_exited = 0;
    #20;
    for (i = 0; i < 30; i = i + 1) begin
        car_entered = 1;
        is_uni_car_entered = 1;
        #1
        car_entered = 0;
        #1;
    end
end

initial #1200 $stop();

endmodule
```

خروجی این تست بنچ:

```
# time:      8, uni_cars:      0, uni_space:   500, free_parked_cars:      0, empty_space:   200
# time:      9, uni_cars:     25, uni_space:   475, free_parked_cars:     25, empty_space:   175
# time:     10, uni_cars:     50, uni_space:   450, free_parked_cars:     50, empty_space:   150
# time:     11, uni_cars:      0, uni_space:   500, free_parked_cars:     50, empty_space:   150
# time:     12, uni_cars:     30, uni_space:   470, free_parked_cars:     50, empty_space:   150
# time:     13, uni_cars:     30, uni_space:   420, free_parked_cars:     50, empty_space:   200
# time:     14, uni_cars:     30, uni_space:   370, free_parked_cars:     50, empty_space:   250
# time:     15, uni_cars:     30, uni_space:   320, free_parked_cars:     50, empty_space:   300
# time:     16, uni_cars:     30, uni_space:   270, free_parked_cars:     50, empty_space:   350
# time:     17, uni_cars:     30, uni_space:   270, free_parked_cars:     50, empty_space:   350
```

این تست بنچ شامل تعدادی ورود و خروج بدون مشکل برای تست کردن بود.

کد tb2 (تنها تفاوت تست بنچ ها در قسمت initial است و از اوردن بقیه قسمت‌ها خودداری کردم):

```
21    integer i;
22    initial begin
23        clk = 0;
24        #50;
25        reset = 0;
26        for (i = 0; i < 700; i = i + 1) begin
27            car_entered = 1;
28            is_uni_car_entered = 1;
29            #1;
30            car_entered = 0;
31            is_uni_car_entered = 0;
32            #1;
33        end
34    end
```

که خروجی آن:

```
VSIM 16> run
# time:      8, uni_cars:      0, uni_space:   500, free_parked_cars:      0, empty_space:   200
# time:      9, uni_cars:     50, uni_space:   450, free_parked_cars:      0, empty_space:   200
# time:     10, uni_cars:    100, uni_space:   400, free_parked_cars:      0, empty_space:   200
# time:     11, uni_cars:    150, uni_space:   350, free_parked_cars:      0, empty_space:   200
# time:     12, uni_cars:    200, uni_space:   300, free_parked_cars:      0, empty_space:   200
# time:     13, uni_cars:    250, uni_space:   200, free_parked_cars:      0, empty_space:   250
# time:     14, uni_cars:    300, uni_space:   100, free_parked_cars:      0, empty_space:   300
# time:     15, uni_cars:    350, uni_space:     0, free_parked_cars:      0, empty_space:   350
# time:     16, uni_cars:    300, uni_space:     0, free_parked_cars:      0, empty_space:   400
# time:     17, uni_cars:    300, uni_space:     0, free_parked_cars:      0, empty_space:   400
VSIM 17> run
# time:     18, uni_cars:    300, uni_space:     0, free_parked_cars:      0, empty_space:   400
# time:     19, uni_cars:    300, uni_space:     0, free_parked_cars:      0, empty_space:   400
```

این تست پر بودن پارکینگ دانشگاه را بررسی میکند.

:Tb3

```
22    initial begin
23        clk = 0;
24        #50;
25        reset = 0;
26        for (i = 0; i < 700; i = i + 1) begin
27            car_entered = 1;
28            is_uni_car_entered = 1;
29            #1
30            car_entered = 0;
31            is_uni_car_entered = 0;
32            #1
33            car_entered = 1;
34            #1
35            car_entered = 0;
36            #1;
37        end
38    end
```

این تست هر دو پارکینگ را پر میکند.

خروجی:

```
# time:      8, uni_cars:      0, uni_space:   500, free_parked_cars:      0, empty_space:   200
# time:      9, uni_cars:     25, uni_space:   475, free_parked_cars:     25, empty_space:   175
# time:     10, uni_cars:     50, uni_space:   450, free_parked_cars:     50, empty_space:   150
# time:     11, uni_cars:     75, uni_space:   425, free_parked_cars:     75, empty_space:   125
# time:     12, uni_cars:    100, uni_space:   400, free_parked_cars:    100, empty_space:   100
# time:     13, uni_cars:    125, uni_space:   325, free_parked_cars:    125, empty_space:   125
# time:     14, uni_cars:    150, uni_space:   250, free_parked_cars:    150, empty_space:   150
# time:     15, uni_cars:    175, uni_space:   175, free_parked_cars:    175, empty_space:   175
# time:     16, uni_cars:    200, uni_space:   100, free_parked_cars:    200, empty_space:   200
# time:     17, uni_cars:    225, uni_space:    75, free_parked_cars:    225, empty_space:   175
# time:     18, uni_cars:    250, uni_space:    50, free_parked_cars:    250, empty_space:   150
# time:     19, uni_cars:    275, uni_space:    25, free_parked_cars:    275, empty_space:   125
# time:     20, uni_cars:    300, uni_space:     0, free_parked_cars:    300, empty_space:   100
# time:     21, uni_cars:    300, uni_space:     0, free_parked_cars:    325, empty_space:    75
# time:     22, uni_cars:    300, uni_space:     0, free_parked_cars:    350, empty_space:    50
# time:     23, uni_cars:    300, uni_space:     0, free_parked_cars:    375, empty_space:    25
# time:     24, uni_cars:    300, uni_space:     0, free_parked_cars:    400, empty_space:     0
```

:Tb4

این تست پر شدن پارکینگ آزاد در زمان افزایش ظرفیت را نشان میدهد.

```
integer i;
initial begin
    clk = 0;
    #50;
    reset = 0;
    is_uni_car_entered = 0;
    for (i = 0; i < 800; i = i + 1) begin
        car_entered = 1;
        #0.5
        car_entered = 0;
        #0.5;
    end
end
```

خروجی:

```
# time:     8, uni_cars:     0, uni_space:   500, free_parked_cars:     0, empty_space:   200
# time:     9, uni_cars:     0, uni_space:   500, free_parked_cars:    50, empty_space:   150
# time:    10, uni_cars:     0, uni_space:   500, free_parked_cars:   100, empty_space:   100
# time:    11, uni_cars:     0, uni_space:   500, free_parked_cars:   150, empty_space:    50
# time:    12, uni_cars:     0, uni_space:   500, free_parked_cars:   200, empty_space:     0
# time:    13, uni_cars:     0, uni_space:   450, free_parked_cars:   249, empty_space:     1
# time:    14, uni_cars:     0, uni_space:   400, free_parked_cars:   299, empty_space:     1
# time:    15, uni_cars:     0, uni_space:   350, free_parked_cars:   349, empty_space:     1
# time:    16, uni_cars:     0, uni_space:   300, free_parked_cars:   399, empty_space:     1
# time:    17, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    18, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    19, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    20, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    21, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    22, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    23, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
# time:    24, uni_cars:     0, uni_space:   300, free_parked_cars:   400, empty_space:     0
```

:Tb4

```
22  initial begin
23      clk = 0;
24      #50;
25      reset = 0;
26      is_uni_car_entered = 0;
27      for (i = 0; i < 10; i = i + 1) begin
28          car_entered = 1;
29          #0.5
30          car_entered = 0;
31          #0.5;
32      end
33      is_uni_car_entered = 1;
34      for (i = 0; i < 20; i = i + 1) begin
35          car_entered = 1;
36          #0.5
37          car_entered = 0;
38          #0.5;
39      end
40      is_uni_car_entered = 0;
41      for (i = 0; i < 30; i = i + 1) begin
42          car_entered = 1;
43          #0.5
44          car_entered = 0;
45          #0.5;
46      end
47      is_uni_car_entered = 1;
48      for (i = 0; i < 40; i = i + 1) begin
```

```
47      is_uni_car_entered = 1;
48      for (i = 0; i < 40; i = i + 1) begin
49          car_entered = 1;
50          #0.5
51          car_entered = 0;
52          #0.5;
53      end
54      is_uni_car_entered = 0;
55      for (i = 0; i < 50; i = i + 1) begin
56          car_entered = 1;
57          #0.5
58          car_entered = 0;
59          #0.5;
60      end
61      is_uni_car_exited = 0;
62      for (i = 0; i < 50; i = i + 1) begin
63          car_exited = 1;
64          #0.5
65          car_exited = 0;
66          #0.5;
67      end
68      is_uni_car_exited = 1;
69      for (i = 0; i < 40; i = i + 1) begin
70          car_exited = 1;
71          #0.5
72          car_exited = 0;
73          #0.5;
74      end
75      is_uni_car_exited = 0;
76      for (i = 0; i < 30; i = i + 1) begin
77          car_exited = 1;
78          #0.5
79          car_exited = 0;
80          #0.5;
```

```
79          car_exited = 0;
80          #0.5;
81      end
82      is_uni_car_exited = 1;
83      for (i = 0; i < 20; i = i + 1) begin
84          car_exited = 1;
85          #0.5
86          car_exited = 0;
87          #0.5;
88      end
89      is_uni_car_exited = 0;
90      for (i = 0; i < 10; i = i + 1) begin
91          car_exited = 1;
92          #0.5
93          car_exited = 0;
94          #0.5;
95      end
96  end
```

که در این تست بنچ تعدادی ورود و خروج نامنظم به هر دو پارکینگ انجام میشود.

خروجی:

```
# time:     8, uni_cars:     0, uni_space:   500, free_parked_cars:     0, empty_space:   200
# time:     9, uni_cars:    20, uni_space:   480, free_parked_cars:    30, empty_space:   170
# time:    10, uni_cars:    60, uni_space:   440, free_parked_cars:    40, empty_space:   160
# time:    11, uni_cars:    60, uni_space:   440, free_parked_cars:    90, empty_space:   110
# time:    12, uni_cars:    60, uni_space:   440, free_parked_cars:    40, empty_space:   160
# time:    13, uni_cars:    20, uni_space:   430, free_parked_cars:    30, empty_space:   220
# time:    14, uni_cars:     0, uni_space:   400, free_parked_cars:     0, empty_space:   300
# time:    15, uni_cars:     0, uni_space:   350, free_parked_cars:     0, empty_space:   350
# time:    16, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
# time:    17, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
# time:    18, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
# time:    19, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
# time:    20, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
# time:    21, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
# time:    22, uni_cars:     0, uni_space:   300, free_parked_cars:     0, empty_space:   400
```

:Tb5

```
22  initial begin
23      clk = 0;
24      #50;
25      reset = 0;
26      for (i = 0; i < 700; i = i + 1) begin
27          car_entered = 1;
28          is_uni_car_entered = 1;
29          #0.5
30          car_entered = 0;
31          #0.5;
32      end
33      is_uni_car_exited = 1;
34      #50;
35      for (i = 0; i < 300; i = i + 1) begin
36          car_exited = 1;
37          #0.5
38          car_exited = 0;
39          #0.5;
40      end
41      for (i = 0; i < 200; i = i + 1) begin
42          car_entered = 1;
43          is_uni_car_entered = 0;
44          #0.5
45          car_entered = 0;
46          #0.5;
47      end
48      for (i = 0; i < 200; i = i + 1) begin
49          car_exited = 1;
50          is_uni_car_exited = 0;
51          #0.5
52          car_exited = 0;
53          #0.5;
54      end
```

که در این تست بنچ هم ابتدا پارکینگ دانشگاه پر و سپس خالی میشود و بعد از ان تعدادی ماشین وارد پارکینگ آزاد و سپس خارج میشوند.

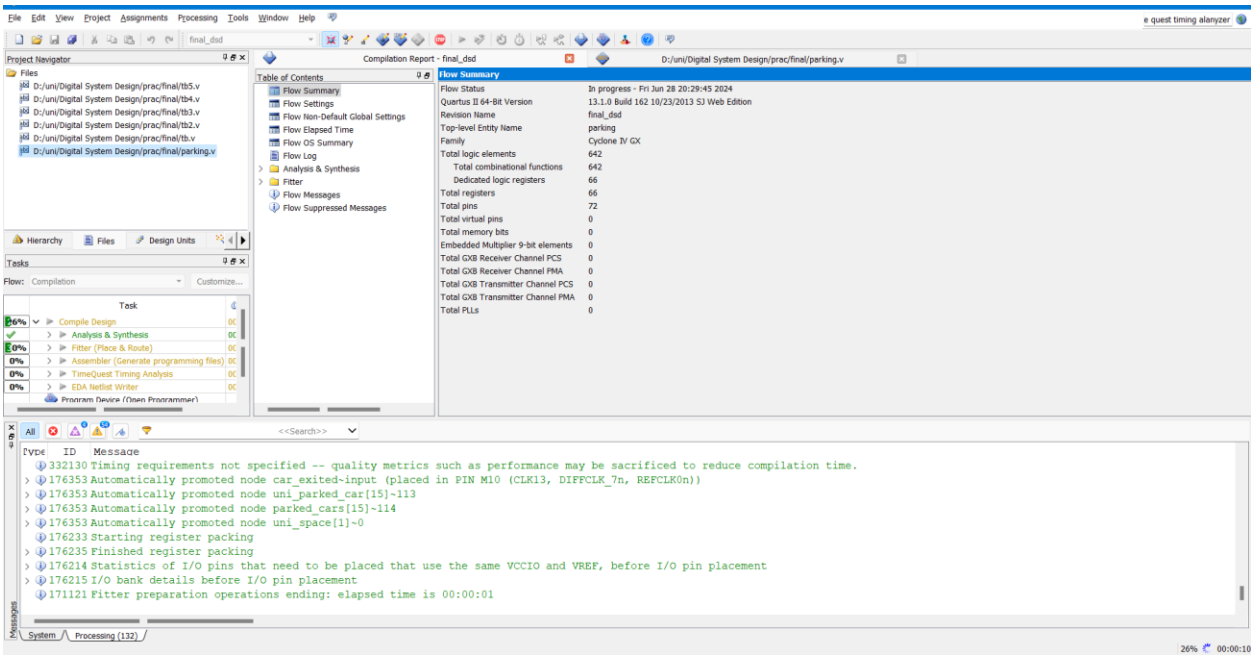
خروجی آن:

```
# time:      8, uni_cars:      0, uni_space:   500, free_parked_cars:      0, empty_space:   200
# time:      9, uni_cars:    100, uni_space:   400, free_parked_cars:      0, empty_space:   200
# time:     10, uni_cars:    200, uni_space:   300, free_parked_cars:      0, empty_space:   200
# time:     11, uni_cars:    300, uni_space:   200, free_parked_cars:      0, empty_space:   200
# time:     12, uni_cars:    400, uni_space:   100, free_parked_cars:      0, empty_space:   200
# time:     13, uni_cars:    450, uni_space:     0, free_parked_cars:      0, empty_space:   250
# time:     14, uni_cars:    400, uni_space:     0, free_parked_cars:      0, empty_space:   300
# time:     15, uni_cars:    350, uni_space:     0, free_parked_cars:      0, empty_space:   350
# time:     16, uni_cars:    250, uni_space:    50, free_parked_cars:      0, empty_space:   400
# time:     17, uni_cars:    150, uni_space:   150, free_parked_cars:      0, empty_space:   400
# time:     18, uni_cars:     50, uni_space:   250, free_parked_cars:      0, empty_space:   400
# time:     19, uni_cars:      0, uni_space:   300, free_parked_cars:     50, empty_space:   350
# time:     20, uni_cars:      0, uni_space:   300, free_parked_cars:    150, empty_space:   250
# time:     21, uni_cars:      0, uni_space:   300, free_parked_cars:    150, empty_space:   250
# time:     22, uni_cars:      0, uni_space:   300, free_parked_cars:     50, empty_space:   350
# time:     23, uni_cars:      0, uni_space:   300, free_parked_cars:      0, empty_space:   400
# time:     24, uni_cars:      0, uni_space:   300, free_parked_cars:      0, empty_space:   400
```

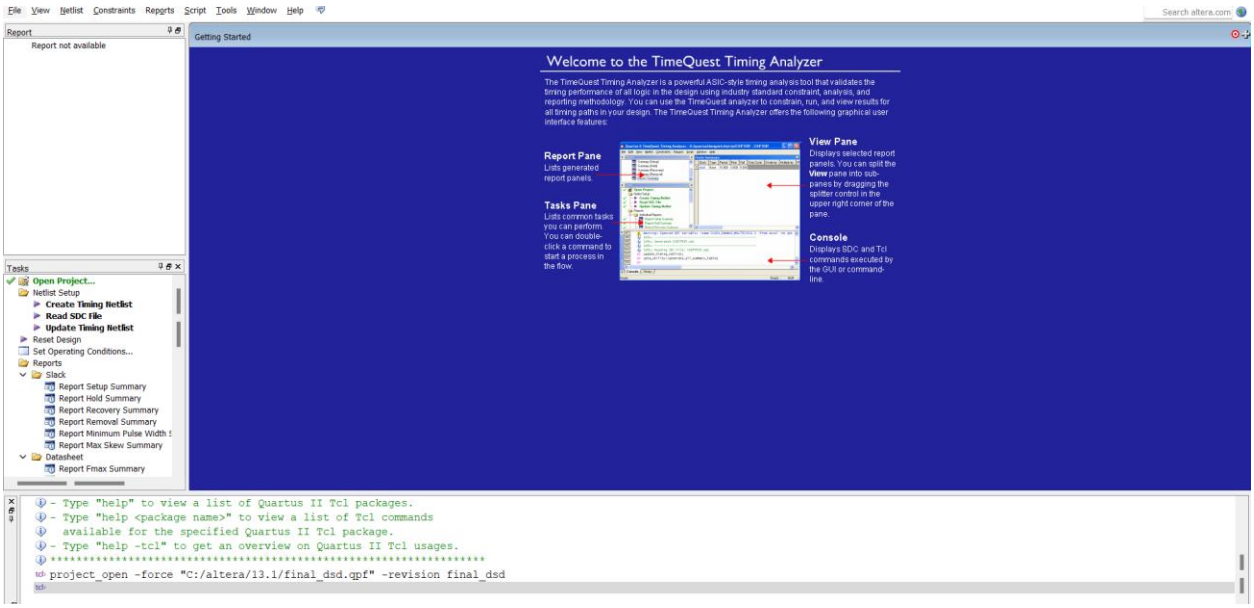
ب)

برای سنتز از کوارتوس استفاده میکنیم.

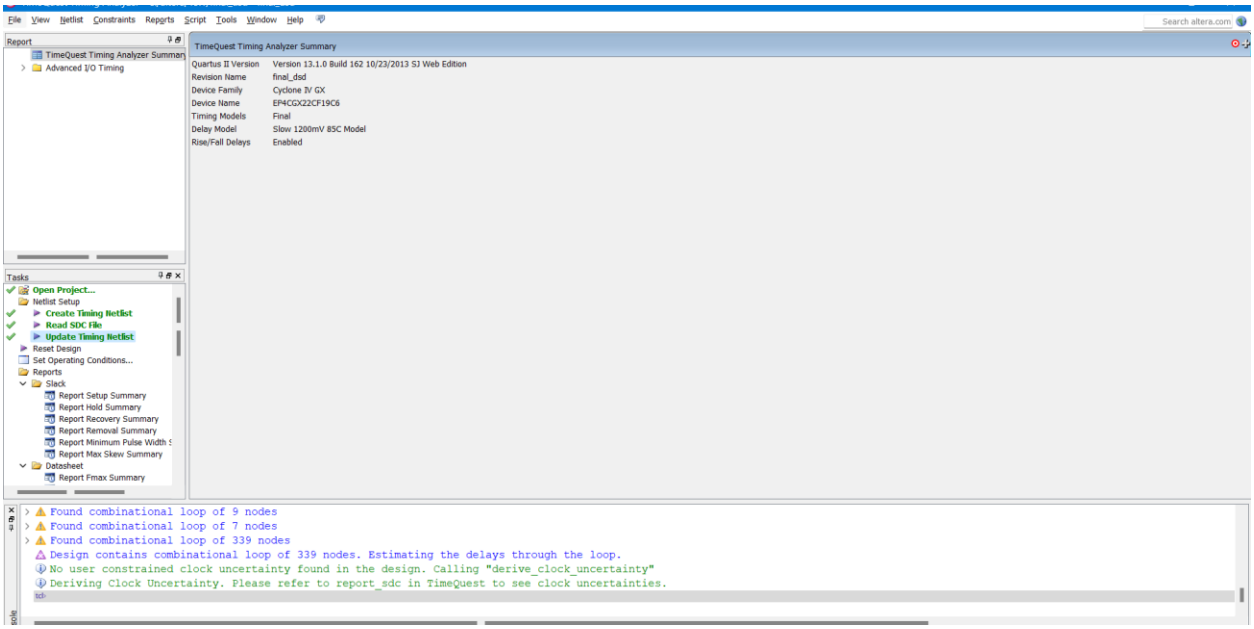
ابتدا برنامه را در کوارتوس کامپایل میکنیم.



سپس time quest analyzer را باز میکنیم.



و بعد از آن create timing netlist و read sdc file و update timing netlist را انتخاب میکنیم.



و در نهایت report fmax summary را باز میکنیم:

Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	5.36 MHz	5.36 MHz	clk	
2	10.38 MHz	10.38 MHz	car_exited	