



پروژه درس طراحی سیستم های دیجیتال

استاد: دکتر فصحتی

صادق محمدیان

شماره دانشجویی: ۴۰۱۱۰۹۴۷۷

تیرماه ۱۴۰۳

## سیستم مدیریت پارکینگ:

می خواهیم یک مدار برای مدیریت یک پارکینگ طراحی کنیم. این مدار باید ویژگی های 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 می رسد.

ماژول اصلی ما باید ورودی های زیر را به همراه کلاک و سیگنال شروع داشته باشد:

ورودی ها	
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؟

همچنین باید خروجی های زیر را نیز داشته باشد:

خروجی ها	
uni_parked_car	تعداد خودروهایی متعلق به دانشگاه که در پارکینگ پارک شده اند.
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؟

ابتدا بهتر است برای زمان و ساعت و ارتباط آن با ظرفیت پارکینگ یک ماژول جداگانه طراحی کنیم:

```
module timer (input start,clk,
               output reg [11:0] Time);
    reg [12:0] count;
    always @(posedge clk)
    begin
        if (~start)
        begin
            Time <= 0;
            count <= 0;
        end
        else
        begin
            if (count >= 60)
            begin
                Time <= Time + 1;
                count <= 0;
            end

            if (Time >= 23)
            begin
                Time <= 0;
            end

            else
            begin
                count <= count + 30;
            end
        end
    end
endmodule
```

همانطور که مشاهده میکنید از یک متغیر `count` استفاده می شود که با رسیدن به عدد ۶۰ صفر می شود و متغیر `Time` زیاد می شود . این ماژول به طور مداوم در حال کار است پس باید بعد از به پایان رسیدن روز صفر شود.

حال با شرایط گفته شده اقدام به طراحی ماژول مورد نظر می پردازیم:

در صورتی سیگنال شروع صفر شود تغییرات زیر در خروجی های ماژول ایجاد می گردد:

```
uni_car_parked <= 0;
parked_care <= 0;
uni_vacated_space <= universityCapacity;
vacated_space <= outCapacity;
uni_is_vacated_space <= 1'b1;
is_vacated_space <= 1'b1;
```

در صورت ورود خودرو جدید منطق آن را در این قسمت پیاده سازی کرده ایم:

```
if (car_entered)
begin
    if (parked_care + uni_car_parked < MAX_CAP_PARKING)
    begin
        if (is_uni_car_entered)
        begin
            if ( 1 <= uni_is_vacated_space &&
$signed(uni_vacated_space))
            begin
                uni_car_parked = uni_car_parked + 1;
                uni_vacated_space = universityCapacity -
uni_car_parked;

                if (uni_car_parked >= universityCapacity)
                begin
                    uni_is_vacated_space = 1'b0;
                end
            end
        end
    end
else
    begin
        if (is_vacated_space && $signed(vacated_space) >= 1)
        begin
            parked_care = parked_care + 1;
            vacated_space = outCapacity - parked_care;
            if (parked_care >= outCapacity)
            begin
                is_vacated_space = 1'b0;
            end
        end
    end
end
end
end
```

در صورت خروج خودرو منطق زیر را پیاده سازی کرده ایم:

```
else if (car_exited)
begin
    if (is_uni_car_exited)
    begin
        if (uni_car_parked > 0)
        begin
            uni_car_parked = uni_car_parked - 1;
            uni_vacated_space <= universityCapacity - uni_car_parked;
            if (!uni_is_vacated_space)
            begin
                uni_is_vacated_space = 1'b1;
            end
        end
    end
end
else
begin
    if (parked_care > 0)
    begin
        parked_care = parked_care - 1;
        vacated_space <= outCapacity - parked_care;
        if (!is_vacated_space)
        begin
            is_vacated_space = 1'b1;
        end
    end
end
end
end
```

در طراحی خود را این را در نظر گرفته ایم که ممکن است در ساعات مختلف که ظرفیت پارکینگ تغییر می کند تغییر ظرفیت خودروهای گروه خودرو های عمومی یا خودروهای مربوط به دانشگاه عددی منفی (!) باشد که در این صورت خودرو ها نمی توانند ورود داشته باشند و این تازمانی ادامه پیدا می کند که همه ی خودرو های اضافه بر ظرفیت خارج شوند.

و این قسمت نیز مربوط به تغییر ظرفیت های پارکینگ با توجه به ساعت است که باید به صورت زیر باشد:

(۱) باتوجه هبه اینکه فضای کل پارکینگ ۷۰۰ خودرو است از ساعت ۸ تا ۱۳ فقط ۲۰۰ ظرفیت خالی برای ورود آزاد موجود است.

(۲) 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 می رسد.

```
//TIME AND CAPACITY
if (Time < 8)
begin
    universityCapacity = 0;
    outCapacity = 0;
end
else if (Time < 13)
begin
    universityCapacity = 500;
    outCapacity = 200;
end
else if (Time < 14)
begin
    universityCapacity = 450;
    outCapacity = 250;
end
else if (Time < 15)
begin
    universityCapacity = 400;
    outCapacity = 300;
end
else if (Time < 16)
begin
    universityCapacity = 350;
    outCapacity = 350;
end
else
begin
    universityCapacity = 200;
    outCapacity = 500;
end
```

صورت کلی ماژول به صورت زیر می باشد:

```
module Parking(input start, clk, car_entered, is_uni_car_entered, car_exited,
is_uni_car_exited,
                output reg [15:0] uni_car_parked, reg [15:0] parked_care, reg
[15:0] uni_vacated_space, reg [15:0] vacated_space, reg uni_is_vacated_space, reg
is_vacated_space, wire [11:0] Time);

    integer MAX_CAP_PARKING = 700;

    TIMER timer (clk,start,Time);

    integer universityCapacity;
    integer outCapacity;

    always @(car_entered or car_exited or start or Time)
    begin

        if (~start)
        begin
            uni_car_parked <= 0;
            parked_care <= 0;
            uni_vacated_space <= universityCapacity;
            vacated_space <= outCapacity;
            uni_is_vacated_space <= 1'b1;
            is_vacated_space <= 1'b1;
        end
        else if (start)
        begin
            vacated_space = outCapacity - parked_care;
            uni_vacated_space = universityCapacity - uni_car_parked;

            if (car_entered)
            begin
                if (parked_care + uni_car_parked < MAX_CAP_PARKING)
                begin
                    if (is_uni_car_entered)
                    begin
                        if ( 1 <= uni_is_vacated_space &&
$signed(uni_vacated_space))
                        begin
                            uni_car_parked = uni_car_parked + 1;
                            uni_vacated_space = universityCapacity -
uni_car_parked;
```

```

        if (uni_car_parked >= universityCapacity)
        begin
            uni_is_vacated_space = 1'b0;
        end
    end
end
else
begin
    if (is_vacated_space && $signed(vacated_space) >= 1)
    begin
        parked_care = parked_care + 1;
        vacated_space = outCapacity - parked_care;
        if (parked_care >= outCapacity)
        begin
            is_vacated_space = 1'b0;
        end
    end
end
end
end
else if (car_exited)
begin
    if (is_uni_car_exited)
    begin
        if (uni_car_parked > 0)
        begin
            uni_car_parked = uni_car_parked - 1;
            uni_vacated_space <= universityCapacity - uni_car_parked;
            if (!uni_is_vacated_space)
            begin
                uni_is_vacated_space = 1'b1;
            end
        end
    end
end
else
begin
    if (parked_care > 0)
    begin
        parked_care = parked_care - 1;
        vacated_space <= outCapacity - parked_care;
        if (!is_vacated_space)
        begin
            is_vacated_space = 1'b1;
        end
    end
end
end
end

```



```

        end
    end
end

//TIME AND CAPACITY
if (Time < 8)
begin
    universityCapacity = 0;
    outCapacity = 0;
end
else if (Time < 13)
begin
    universityCapacity = 500;
    outCapacity = 200;
end
else if (Time < 14)
begin
    universityCapacity = 450;
    outCapacity = 250;
end
else if (Time < 15)
begin
    universityCapacity = 400;
    outCapacity = 300;
end
else if (Time < 16)
begin
    universityCapacity = 350;
    outCapacity = 350;
end
else
begin
    universityCapacity = 200;
    outCapacity = 500;
end

end

endmodule

```

حال ماژول اصلی است و باید آن را در حالت های مختلف تست کنیم:

ابتدا ورودی ها و خروجی های مدار را میسازیم و از ماژول خود اینستنس میگیریم:

```
reg clk;
reg start;
reg car_entered;
reg is_uni_car_entered;
reg car_exited;
reg is_uni_car_exited;
wire [15:0] uni_car_parked;
wire [15:0] parked_care;
wire [15:0] uni_vacated_space;
wire [15:0] vacated_space;
wire uni_is_vacated_space;
wire is_vacated_space;
wire [11:0] Time;

Parking
parking(clk,start,car_entered,is_uni_car_entered,car_exited,is_uni_car_exited,uni
_car_parked,parked_care,uni_vacated_space,vacated_space,uni_is_vacated_space,is_v
acated_space,clock_time);

initial clk <= 0;
always #5 clk <= ~clk;
```

حال در یک بلاک initial تست های خود را می نویسیم که در فایل مربوطه نوشته شده است به علت طولانی بودن آن صرف نظر میکنیم و نتایج آن را نشان می دهیم:

```
time:  x car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 200 vacated space: 500
time: 0 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 0 car entered: 0 car exited: 1 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 0 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 1 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 2 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 3 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 4 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 5 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 6 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 7 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 0 vacated space: 0
time: 8 car entered: 0 car exited: 0 uni car parked: 0 parked car : 0 uni vacated space: 500 vacated space: 200
time: 8 car entered: 1 car exited: 0 uni car parked: 0 parked car : 100 uni vacated space: 500 vacated space: 100
time: 8 car entered: 0 car exited: 0 uni car parked: 0 parked car : 100 uni vacated space: 500 vacated space: 100
time: 9 car entered: 0 car exited: 0 uni car parked: 0 parked car : 100 uni vacated space: 500 vacated space: 100
time: 10 car entered: 0 car exited: 0 uni car parked: 0 parked car : 100 uni vacated space: 500 vacated space: 100
time: 10 car entered: 1 car exited: 0 uni car parked: 100 parked car : 100 uni vacated space: 400 vacated space: 100
time: 11 car entered: 0 car exited: 0 uni car parked: 100 parked car : 100 uni vacated space: 400 vacated space: 100
time: 12 car entered: 0 car exited: 0 uni car parked: 100 parked car : 100 uni vacated space: 400 vacated space: 100
time: 13 car entered: 0 car exited: 1 uni car parked: 100 parked car : 0 uni vacated space: 350 vacated space: 250
time: 13 car entered: 0 car exited: 0 uni car parked: 100 parked car : 0 uni vacated space: 350 vacated space: 250
time: 14 car entered: 0 car exited: 0 uni car parked: 100 parked car : 0 uni vacated space: 300 vacated space: 300
time: 15 car entered: 0 car exited: 0 uni car parked: 100 parked car : 0 uni vacated space: 250 vacated space: 350
time: 15 car entered: 1 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 150 vacated space: 350
time: 15 car entered: 0 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 150 vacated space: 350
time: 16 car entered: 0 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 0 vacated space: 500
time: 17 car entered: 0 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 0 vacated space: 500
time: 17 car entered: 1 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 0 vacated space: 500
time: 17 car entered: 0 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 0 vacated space: 500
time: 18 car entered: 0 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 0 vacated space: 500
time: 19 car entered: 0 car exited: 0 uni car parked: 200 parked car : 0 uni vacated space: 0 vacated space: 500
time: 19 car entered: 1 car exited: 0 uni car parked: 200 parked car : 100 uni vacated space: 0 vacated space: 400
time: 19 car entered: 0 car exited: 0 uni car parked: 200 parked car : 100 uni vacated space: 0 vacated space: 400
time: 20 car entered: 0 car exited: 0 uni car parked: 200 parked car : 100 uni vacated space: 0 vacated space: 400
```



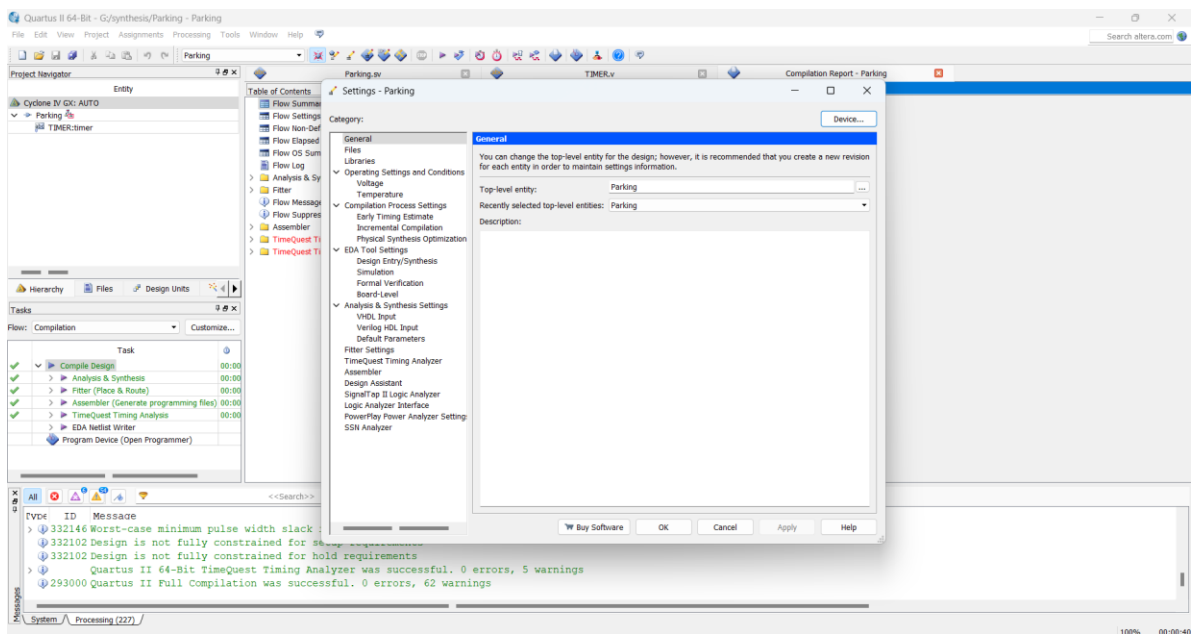
حال می خواهیم مدار خود را سنتز کنیم

برای اینکار از نرم افزار کوارتوس کمک می گیریم و مراحل گفته شده را انجام می دهیم.

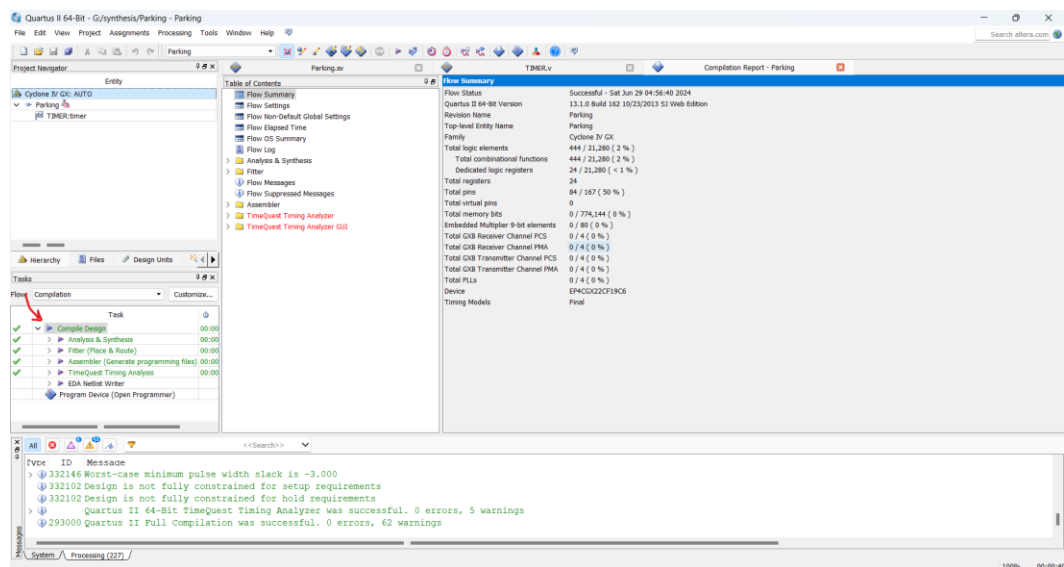
یک پروژه می سازیم و فایل های خود را آپلود می کنیم دقت کنید که باید system Verilog باشند.

نیازی به آپلود فایل تست نمی باشد.

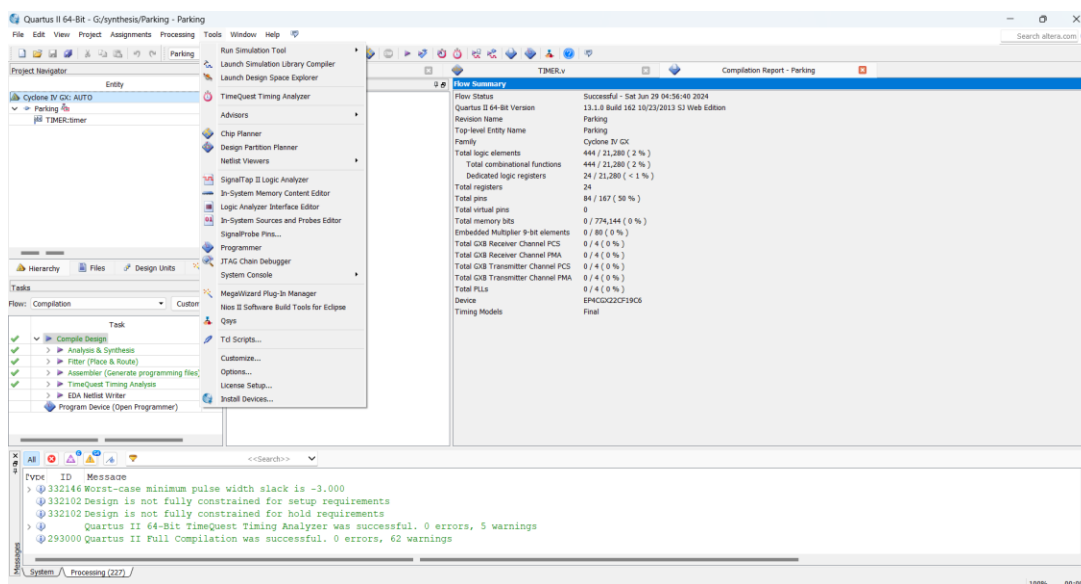
سپس ماژول top level خود را انتخاب می کنیم:



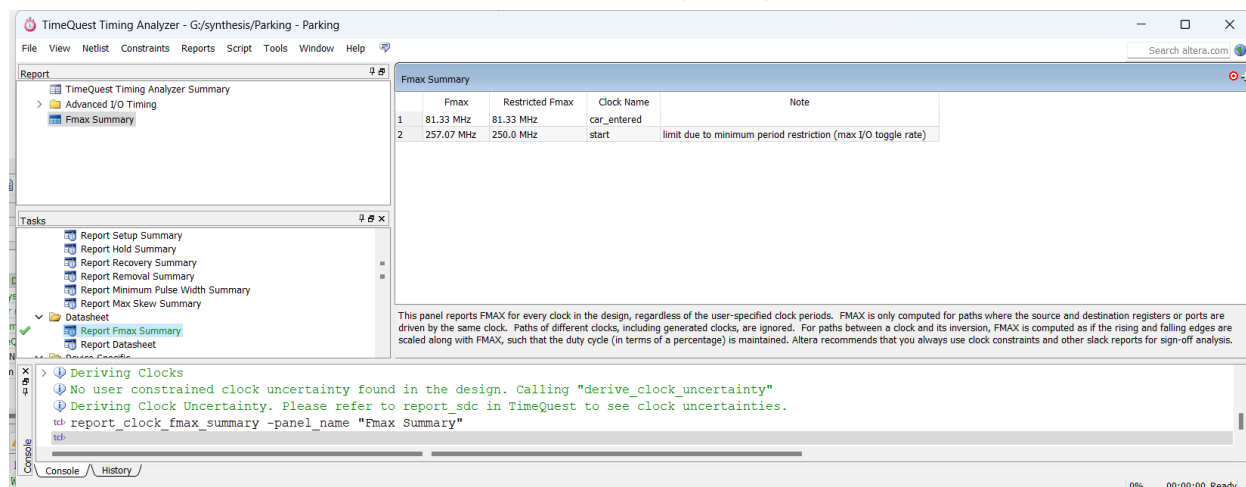
حال برنامه خود را کامپایل میکنیم. این مرحله ممکن است کمی زمان بر باشد:



سپس در منوی tools گزینه ی timeQuest Timing Analyzer را انتخاب می کنیم:



و در منوی سمت چپ به ترتیب create timing netlist، read sdc file و سپس report fmax summary را برای دیدن فرکانس ماکزیمم میزنیم.



این مقدار ۲۵۷.۰۷ مگاهرتز در حالت بدون محدودیت I/O و ۲۵۰ مگاهرتز در حالت با محدودیت I/O می باشد.

TimeQuest Timing Analyzer - G:/synthesis/Parking - Parking

File View Netlist Constraints Reports Script Tools Window Help

Search altera.com

Report

- TimeQuest Timing Analyzer Summary
- Advanced I/O Timing
- Fmax Summary
- Top Falling Paths
  - Setup: car\_entered
  - Setup: start
  - Hold: car\_entered
  - Recovery: car\_entered

Tasks

- Report Net Timing...
- Report Skew...
- Report Max Skew...
- Create Slack Histogram...
- Report Timing Closure Recommendations...
- Macros
  - Report All Summaries
  - Report Top Falling Paths
  - Report All I/O Timings

Setup: car\_entered

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-5.648	uni_car_parked[4]\$latch	uni_car_parked[11]\$latch	car_entered	car_entered	0.500	-2.718	2.626
2	-5.638	uni_car_parked[4]\$latch	uni_car_parked[9]\$latch	car_entered	car_entered	0.500	-2.719	2.498
3	-5.636	uni_car_parked[6]\$latch	uni_car_parked[11]\$latch	car_entered	car_entered	0.500	-2.841	2.491
4	-5.635	uni_car_parked[10]\$latch	uni_car_parked[11]\$latch	car_entered	car_entered	0.500	-2.838	2.493
5	-5.626	uni_car_parked[6]\$latch	uni_car_parked[9]\$latch	car_entered	car_entered	0.500	-2.842	2.363
6	-5.581	uni_car_parked[2]\$latch	uni_car_parked[11]\$latch	car_entered	car_entered	0.500	-2.800	2.477
7	-5.571	uni_car_parked[2]\$latch	uni_car_parked[9]\$latch	car_entered	car_entered	0.500	-2.801	2.349
8	-5.557	uni_car_parked[14]\$latch	uni_car_parked[15]\$latch	car_entered	car_entered	0.500	-2.718	2.455
9	-5.550	uni_car_parked[8]\$latch	uni_car_parked[11]\$latch	car_entered	car_entered	0.500	-2.714	2.532
10	-5.543	uni_car_parked[5]\$latch	uni_car_parked[9]\$latch	car_entered	car_entered	0.500	-2.714	2.408
11	-5.540	uni_car_parked[8]\$latch	uni_car_parked[9]\$latch	car_entered	car_entered	0.500	-2.715	2.404
12	-5.530	uni_car_parked[5]\$latch	uni_car_parked[11]\$latch	car_entered	car_entered	0.500	-2.713	2.513
13	-5.518	uni_car_parked[4]\$latch	uni_car_parked[5]\$latch	car_entered	car_entered	0.500	-2.720	2.506
14	-5.515	uni_car_parked[4]\$latch	uni_car_parked[13]\$latch	car_entered	car_entered	0.500	-2.715	2.529
15	-5.508	uni_car_parked[5]\$latch	uni_car_parked[12]\$latch	car_entered	car_entered	0.500	-2.592	2.529
16	-5.503	uni_car_parked[6]\$latch	uni_car_parked[13]\$latch	car_entered	car_entered	0.500	-2.838	2.394

Console

```

> Report Timing: Found 200 setup paths (200 violated). Worst case slack is -2.890
> Report Timing: Found 200 hold paths (200 violated). Worst case slack is -3.888
> Report Timing: Found 27 recovery paths (27 violated). Worst case slack is -0.371
> Report Timing: Found 127 removal paths (127 violated). Worst case slack is -3.334
  
```

0% 00:00:00 Ready

و سپس report top falling paths را میزنیم تا مشاهده کنیم مسیر هایمان چقدر تاخیر دارند. اگر این تاخیر دیتا را معکوس کنیم با در نظر گرفتن تقریب دهگان هم اندازه ی فرکانس ما می شود.

چون عامل موثر بر آن فرکانس تاخیر مسیر بحرانی می باشد.

$$2.626 + 2.609 = 5.235 - > f = 0.191 \sim 0.25707 \pm 0.05$$

برای دیدن فیلم نحوه سنتز با کوارتوس می توانید فیلم از لینک زیر را مشاهده فرمایید: