

FPGA时序分析工具 (TimeQuest)

1、提出问题 (点灯程序)

观看以下程序：

```
1 module led(  
2     input      clk      ,//系统时钟, 50MHZ  
3     input      rst_n    ,//系统复位, 低电平有效  
4  
5     output reg  led  
6 );  
7  
8     reg [24:0] cnt      ;//定义一个计数器  
9  
10    always @(posedge clk or negedge rst_n)begin  
11        if(rst_n==1'b0)begin  
12            cnt <= 25'd0;  
13        end  
14        else begin  
15            cnt <= cnt + 1;  
16        end  
17    end  
18  
19    always @(posedge clk or negedge rst_n)begin  
20        if(rst_n==1'b0)begin  
21            led <= 1'b1;  
22        end  
23        else if(cnt == 25'd24_999_999)begin  
24            led <= ~led;  
25        end  
26    end  
27  
28    endmodule
```



问题：上述代码所描述的逻辑电路在Cyclone IV E的EP4CE10F17C8（65nm）这个器件上能最高运行在多少频率的时钟？

此问题需要 **时序分析** 软件来告知我们答案，Altera通过Quartus的TimeQuest软件对___进行分析，并给出报告。

💡 时序分析注意：

- 1、时序分析一定是基于特定器件（该器件需要具体到一个特定型号的特定速度等级）；
- 2、时序分析一定是基于某个逻辑设计在特定器件上经过布局布线之后的网表（该网表包含了设计中每一个逻辑具体在该器件的什么资源上实现，而且还包括该资源在器件内的具体位置，还包括信号从每个节点传输到下一个节点的具体延迟时间）。
- 3、时序分析不是在对代码进行分析，而是在对真实的门级传输路径进行分析。

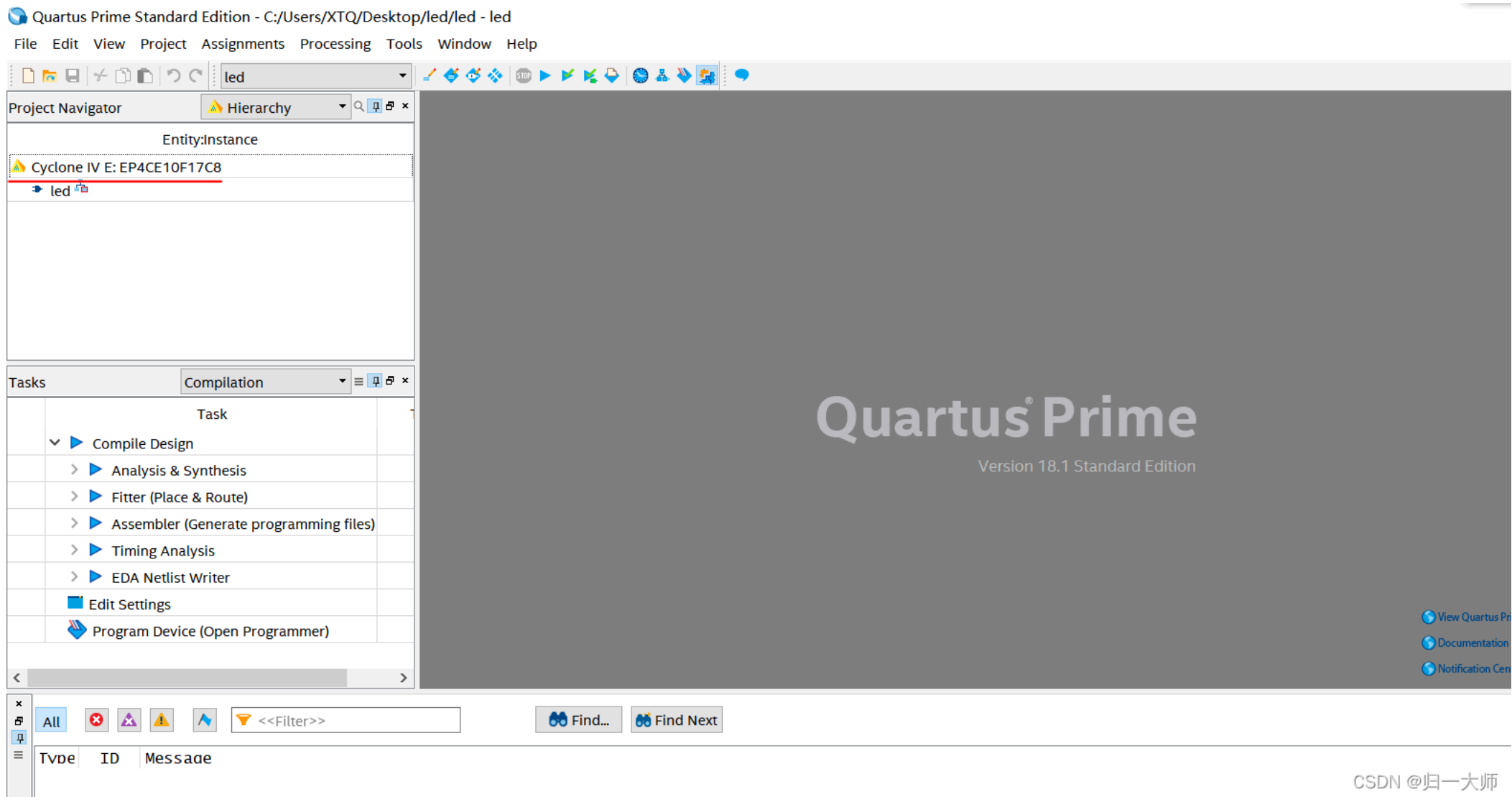


图1 quartus软件显示工程使用的器件

2、时序约束

2.1、全编译工程

准备好工程之后，先对工程全编译一次，注意，是全编译，不是分析和综合，新手如果不确定，直接使用组合键 CTRL + L 吧。

为什么要全编译呢？

1 |  这里有一个（知识）点：时序分析一定是基于软件编译好的具体到一个特定型号的芯片网表进行的。

时序分析的本质是什么？

是分析设计中每个信号在实际物理传输时的各种延迟是否满足一定的要求。既然是延迟，那么就必须对应到具体的一个物理网表，纯粹对代码进行时序分析是没有任何意义和依据的。时序分析分析的不是代码，而是代码描述的电路在某个具体型号芯片上实现时的物理特性。因此，进行时序分析前，全编译工程很重要。

正常编译之后，TimeQuest会变红报错，这是由于我们没有手动加入时序约束，系统默认给CLK添加的时序约束是1GHZ，现实综合之后的最高时钟频率只能达到304.79MHZ，不能满足默认的1GHZ，软件当然就报错了；

Quartus Prime Standard Edition - C:/Users/XTQ/Desktop/led/led - led

File Edit View Project Assignments Processing Tools Window Help

led

Project Navigator Hierarchy

Entity:Instance

Cyclone IV E: EP4CE10F17C8

led

Tasks Compilation

Task

Compile Design 00:00

Analysis & Synthesis 00:00

Fitter (Place & Route) 00:00

Assembler (Generate programming files) 00:00

Timing Analysis 00:00

EDA Netlist Writer 00:00

Edit Settings

Program Device (Open Programmer)

Compilation Report - led

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed Messages

Flow Summary

<<Filter>>

| | |
|------------------------------------|---|
| Flow Status | Successful - Thu Jan 20 15:23:31 2022 |
| Quartus Prime Version | 18.1.0 Build 625 09/12/2018 SJ Standard Edition |
| Revision Name | led |
| Top-level Entity Name | led |
| Family | Cyclone IV E |
| Device | EP4CE10F17C8 |
| Timing Models | Final |
| Total logic elements | 34 / 10,320 (< 1 %) |
| Total registers | 26 |
| Total pins | 3 / 180 (2 %) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 423,936 (0 %) |
| Embedded Multiplier 9-bit elements | 0 / 46 (0 %) |
| Total PLLs | 0 / 2 (0 %) |

IP Catalog

Installed IP

Project Direct

No Selecti

Library

- Basic Func
- DSP
- Interface F
- Memory Ir
- Processor
- University

Search for Pa

Find... Find Next

Type ID Message

Running Quartus Prime EDA Netlist Writer

Command: quartus_eda --read_settings_files=on --write_settings_files=off led -c led

18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value.

199027 Can't generate output files. Specify command-line options to generate output files, or update EDA tool settings using GUI or Tcl script.

Quartus Prime EDA Netlist Writer was successful. 0 errors, 2 warnings

CSDN @归一大师

图2 工程编译后TimeQuest报错

默认添加1GHZ的最大时钟频率约束：

Quartus Prime Standard Edition - C:/Users/XTC/Desktop/led/led - led

File Edit View Project Assignments Processing Tools Window Help

led

Project Navigator Hierarchy

Entity:Instance

Cyclone IV E: EP4CE10F17C8

led

Tasks Compilation

Task

Compile Design 00:00

Analysis & Synthesis 00:00

Fitter (Place & Route) 00:00

Assembler (Generate programming files) 00:00

Timing Analysis 00:00

EDA Netlist Writer 00:00

Edit Settings

Program Device (Open Programmer)

Table of Contents

- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
 - Summary
 - Parallel Compilation
 - Clocks
 - Slow 1200mV 85C Model
 - Slow 1200mV 0C Model
 - Fast 1200mV 0C Model
- Multicorner Timing Analysis Summary
- Advanced I/O Timing
- Clock Transfers
- Report TCCS
- Report RSKM
- Unconstrained Paths
- Messages
- EDA Netlist Writer
- Flow Messages

Clocks

<<Filter>>

| | Clock Name | Type | Period | Frequency | Rise | Fall | Duty Cycle | Divide by | Multiply by | P |
|---|------------|------|--------|------------|-------|-------|------------|-----------|-------------|---|
| 1 | clk | Base | 1.000 | 1000.0 MHz | 0.000 | 0.500 | | | | P |

CSDN @归一犬师

图3 quartus默认给工程添加1GHz时钟约束

系统运行最高时钟频率查看方式：

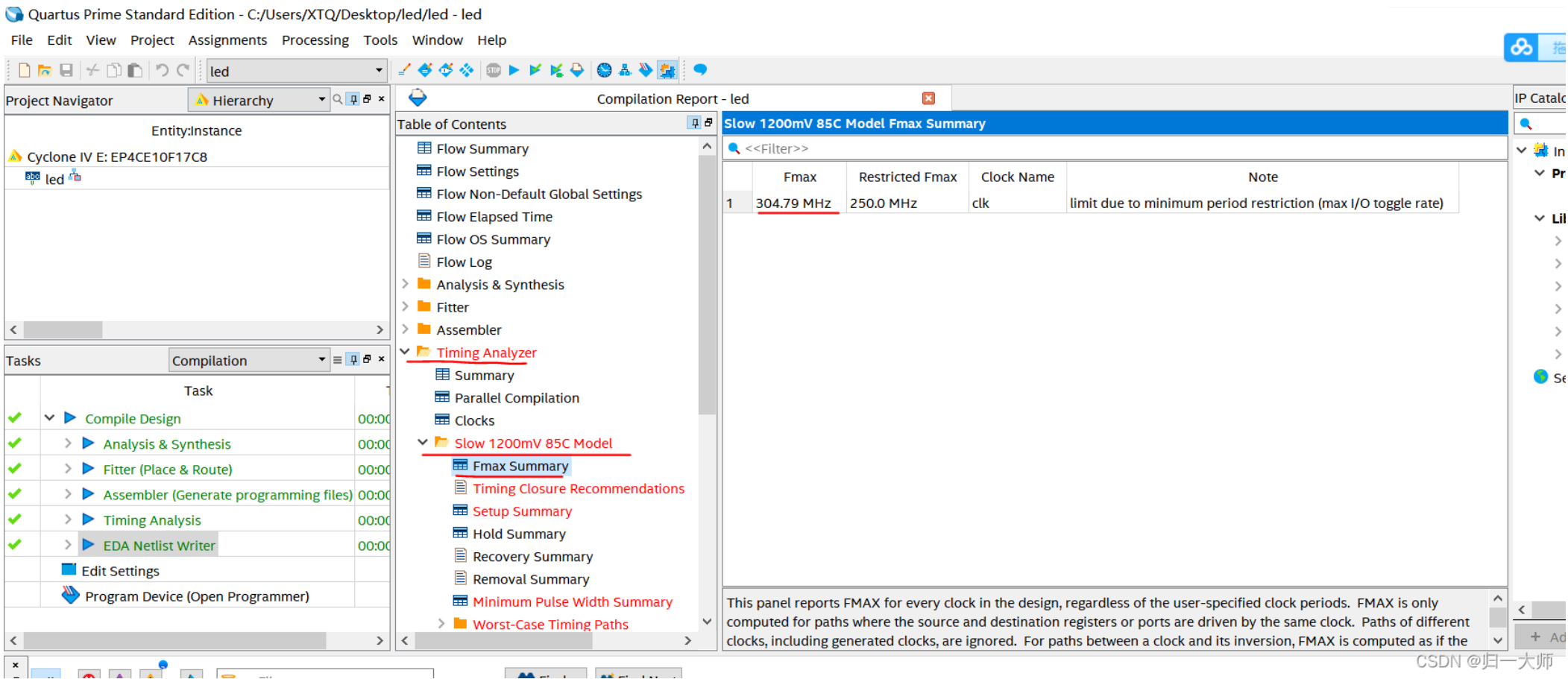
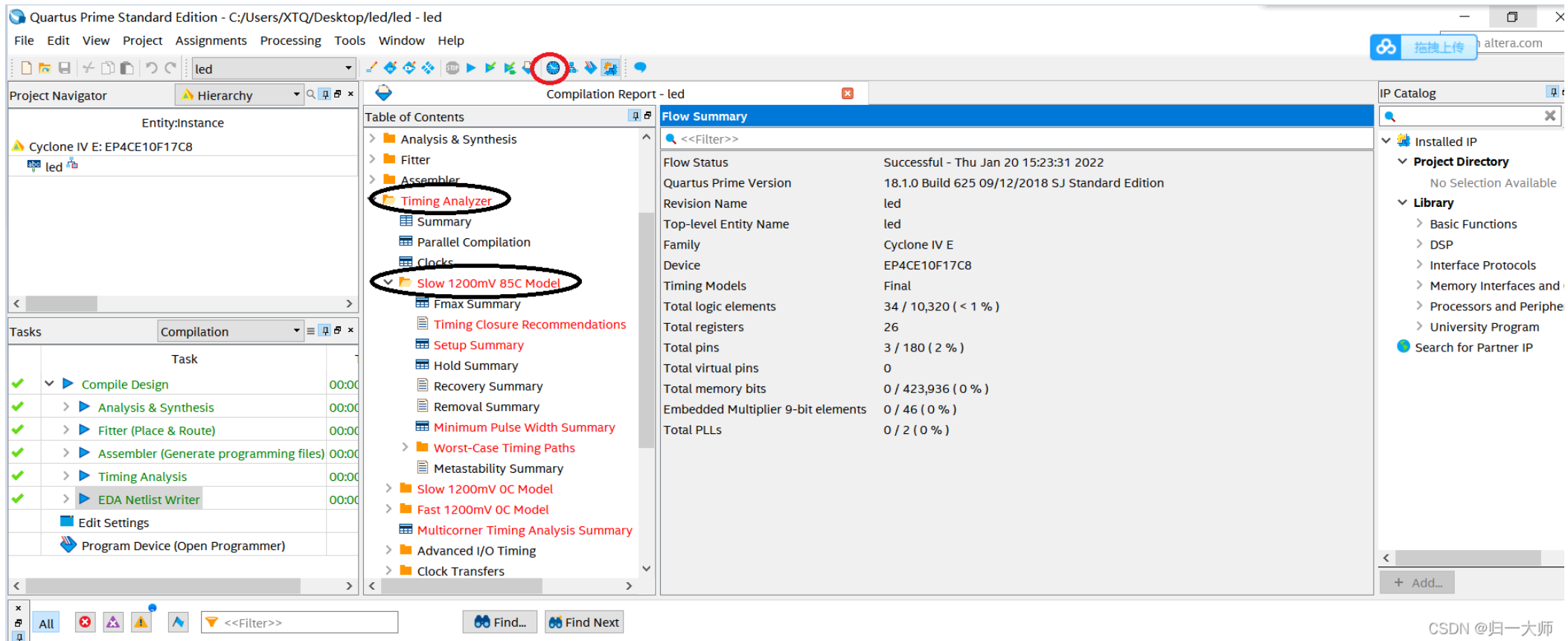


图4 quartus查看工程运行最大系统时钟

2.2、理解时序与温度的关联

执行全编译之后，就可以查看整个工程的时序报告了。怎么看呢？每次编译完成，Quartus软件都会自动显示编译报告界面，如果没有显示，也可以点击编译报告图标来切换到编译报告页面，编译报告图标就在下图所示的位置，打开之后，时序报告都在 TimeQuest Timing Analyzer 这个栏目下面，点击前面的三角符号即可展开，如下图所示：



CSDN @归一大师

图5 quartus全编译工程后查看时序报告

• 展开之后有好多个选项，例如 Clocks，Slow 1200mV 85C Model、Slow 1200mV 0C Model、Fast 1200mV 0C Model。这里先介绍下这几项的物理意义。

- 1、Clocks：这一项是显示当前的设计中有哪些时钟信号，比如本节所用的例子中，就只有一个 clk 的时钟。
- 2、Slow 1200mV 85C Model：芯片内核供电电压 1200mV，工作温度85度情况下的慢慢速传输模型。
- 3、Slow 1200mV 0C Model：芯片内核供电电压 1200mV，工作温度0度情况下的慢速传输模型。
- 4、Fast 1200mV 0C Model：芯片内核供电电压 1200mV，工作温度0度情况下的快速传输模型。

根据专家说法（我还没到能够引经据典讲解芯片中信号传输特性随温度变化的能力，即使有讲，也是BD来的，索性就不说原理了），温度越高，芯片内信号传输速度越慢，说是温度会改变硅片材料，泄露电流和电子移动能力等特性，导致较高温度下芯片内信号传输速度变慢（这几句我是借鉴来的）。信号在相同路径中的传输时间变长，也就是延时会增加，这

也就是做产品需要关心的问题。在实验室环境下运行良好的系统，能否在环境温度达到85度的情况下依旧运行良好？如果到了85度环境下运行不好了，可能就是高温导致的信号传输速度下降，从而导致时序不满足了。

好了，你既然说温度越高，芯片内信号传输速度越慢，那么请问你在“芯片内核供电电压 1200mV，工作温度 0”情况下也说个 Slow 又是什么意思呢？这个嘛，网上说法是晶体管阈值电压会随着温度降低而降低，出现逆温现象，有可能在低温是克服载流子迁移加速问题，从而导致延时时间变长，实际上时序最差情况往往出现在低温，而不是高温状态下。（不要问我更多，这几句内容我也是网上抄来的）。

不管是 0 度还是 85 度慢速条件下，信号传输速率变慢都有可能影响到目的寄存器的建立时间余量，信号传输速率变慢了，建立时间余量就小了。如果之前实验室环境下建立时间余量都已经处于临界状态了，结果一到高温环境，数据速率变慢，建立时间可能就无法再满足了，整个设计就工作异常了。

当然了，低温情况下并不是所有的特性都会变慢，还是有些地方信号传输会变快，一旦传输速度变快，这个时候就要考虑另一个问题了——保持时间余量。快速情况下数据传输速度变快了，如果之前实验室环境下保持时间余量都已经处于临界状态了，结果一到低温环境，数据速率变更快，保持时间可能就无法再满足了，整个设计就工作异常了。

所以，我们看报告的时候，一般 Slow 1200mV 85C Model 和 Slow 1200mV 0C Model 情况下重点关注寄存器建立时间余量，而 Fast 1200mV 0C Model 情况下重点关注寄存器保持时间余量。当然，话还是不说绝对了，在涉及到 IO 时序约束的时候，几种情况还是都得看（虽然我目前大部分情况下只关心了 Slow 1200mV 85C Model）

2.3、查看最大时钟运行频率

大部分情况下关心 Slow 1200mV 85C Model 时的情况，那么就先看这种模式下的时序报告。

展开“TimeQuest Timing Analyzer”下的“Slow 1200mV 85C Model”，可以看到，排在第一项的就是“Fmax Summary”。点击“Fmax Summary”，则右侧显示了当前设计中每个时钟域的逻辑能够运行的最大时钟频率。由于本例子只有一个时钟信号，所以只显示 clk这个时钟的最大运行频率，为 304.79MHz。

Quartus Prime Standard Edition - C:/Users/XTQ/Desktop/led/led - led

File Edit View Project Assignments Processing Tools Window Help

Project Navigator Hierarchy

Entity:Instance

Cyclone IV E: EP4CE10F17C8

led

Tasks Compilation

| Task | Time |
|--|-------|
| Compile Design | 00:00 |
| Analysis & Synthesis | 00:00 |
| Fitter (Place & Route) | 00:00 |
| Assembler (Generate programming files) | 00:00 |
| Timing Analysis | 00:00 |
| EDA Netlist Writer | 00:00 |
| Edit Settings | |
| Program Device (Open Programmer) | |

Compilation Report - led

Table of Contents

- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
 - Summary
 - Parallel Compilation
 - Clocks
 - Slow 1200mV 85C Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Metastability Summary
- Slow 1200mV 0C Model
- Fast 1200mV 0C Model
- Multicorner Timing Analysis Summary
- Advanced I/O Timing
- Clock Transfers

Slow 1200mV 85C Model Fmax Summary

| | Fmax | Restricted Fmax | Clock Name | lir |
|---|------------|-----------------|------------|-----|
| 1 | 304.79 MHz | 250.0 MHz | clk | |

This panel reports FMAX for every clock in the design, reg computed for paths where the source and destination regi clocks, including generated clocks, are ignored. For paths

图6 quartus查看最大系统时钟频率

等等，大家有没有发现一个神奇的事情，设计中有且仅有一个时钟信号，而我们并没有对设计加入任何的时序约束内容，甚至连 SDC 文件都没有，但是报告却自动的搜寻到了这个 clk 信号，并且报告了其时钟频率，这。。。是为啥呢？

这里就又有有一个（知识）点。因为，对于这样一个简单的设计，Quartus 软件能够在编译过程中自动的识别出设计中潜在的时钟属性的信号，并按照软件默认的对时钟的约束去分析设计。而对于一些较为复杂的设计或者多时钟信号的设计，软件可能就分析不出来或者无法确定到底谁是时钟信号了，所以就无法给出时序报告。

只需要点击“TimeQuest Timing Analyzer”下的 Clock 选项就可以查看软件默认的对该信号加的约束是多少了，如下图所示：

The screenshot shows the Quartus II Timing Analyzer window. The 'Table of Contents' pane on the left has 'Timing Analyzer' expanded, and 'Clocks' is selected. The 'Clocks' table on the right shows a single entry for 'clk' with a frequency of 1000.0 MHz, which is circled in red. The 'Duty Cycle' column is greyed out.

| | Clock Name | Type | Period | Frequency | Rise | Fall | Duty Cycle |
|---|------------|------|--------|------------|-------|-------|------------|
| 1 | clk | Base | 1.000 | 1000.0 MHz | 0.000 | 0.500 | |

CSDN @归一大师

图7 quartus默认添加的时钟约束

可以看到，软件自动识别了 clk信号为设计的时钟，而且认为其周期为 1ns，也就是时钟频率为 1000MHz，占空比为 50%。芯片也就最高 400 多兆的运行能力，软件上来就加个 1000MHz 的时钟信号，大概是想通过这种方式，让每个设计都报时序错误，好提醒用户记得加时序约束。当然了，另一个好处就是按照 1000MHz 这种超极限约束，全局会默认就按照最优的可能去对设计进行布局布线，使得设计即使忘了加约束，编译结果也能有个较好的时序性能。

当然，这只是对于简单的设计，软件能够分析出来，而对于复杂的设计，软件就没这个本事了。所以还是希望大家不要以为软件能帮你包办一切，该自己动手的，还是不要让软件去浮想联翩。

2.4、加入时钟约束

假如，这是一个很复杂的系统，那么软件将无法分析出系统中的时钟信号，也就不能自动加入约束，所以，我们是无法查看到最大时钟运行频率的，因此，接下来我们自己来为工程加入时钟约束。

加入时钟约束的方法有很多，比如：

- 1. 高手可以直接自己写 SDC 文件。
- 2. 可以用 TimeQuest Timing Analyzer Wizard 软件按照向导的指引一步一步操作。
- 3. 可以使用 TimeQuest Timing Analyzer 工具以图形化界面添加约束。

为了让大家更加熟悉 TimeQuest Timing Analyzer 的用法，所以还是使用 TimeQuest Timing Analyzer 工具来对工程添加约束。

2.4.1、打开 TimeQuest Timing Analyzer

打开TimeQuest Timing Analyzer的方法也有很多，通过菜单栏依次点击：Tools->TimeQuest Timing Analyzer 可以打开，也可以直接点击工具栏里的 TimeQuest Timing Analyzer 图标，如下图所示：



图8 quartus的TimeQuest打开方式

不管用什么方法，总之打开就是了。打开之后的界面就是下面的样子了：

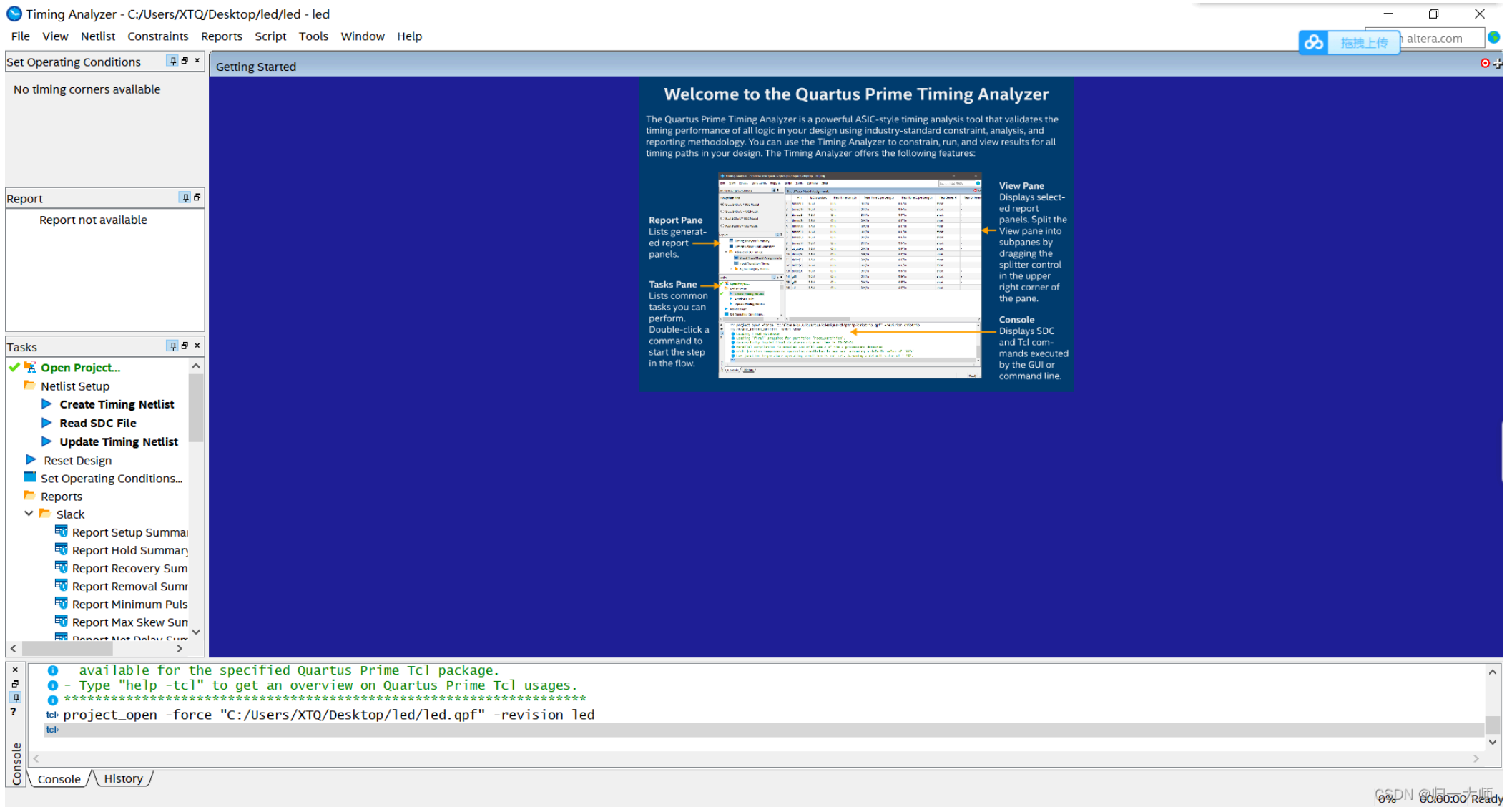


图9 quartus的TimeQuest界面

2.4.2、创建网表 (Create Timing Netlist)

要想进行时序约束或时序分析，必须要有一个能够进行分析或约束的基本对象，那么什么是这个对象呢？对象就是网表模型。

网表模型是啥？先来说说网表是啥吧。简单的理解，网表就是 Quartus 软件经过编译，布局布线后得到的实际适配某一个特定型号芯片的电路信息，该电路信息就是对应下载到芯片中后实际在芯片中呈现的电路。

一个具体的设计，在不同的温度环境下，其工作表现是不一样的，而分析的时候，取了三种模型，分别为 Slow 1200mV 85C Model、Slow 1200mV 0C Model 和 Fast 1200mV 0C Model，这三种模型，就是进行时序分析时候的对象。

打开 TimeQuest Timing Analyzer 工具之后，是没有选择网表对象的，所以要想能够进行时序分析，先要选择网表。选择网表的过程称为创建时序网表（Create Timing Netlist），至于如何创建网表，很多时序约束之类的文章会讲多种方法，这里只分享最简单直接明了的方法，就是在 TimeQuest Timing Analyzer 界面中直接双击 Create Timing Netlist 选项即可，如下图所示：

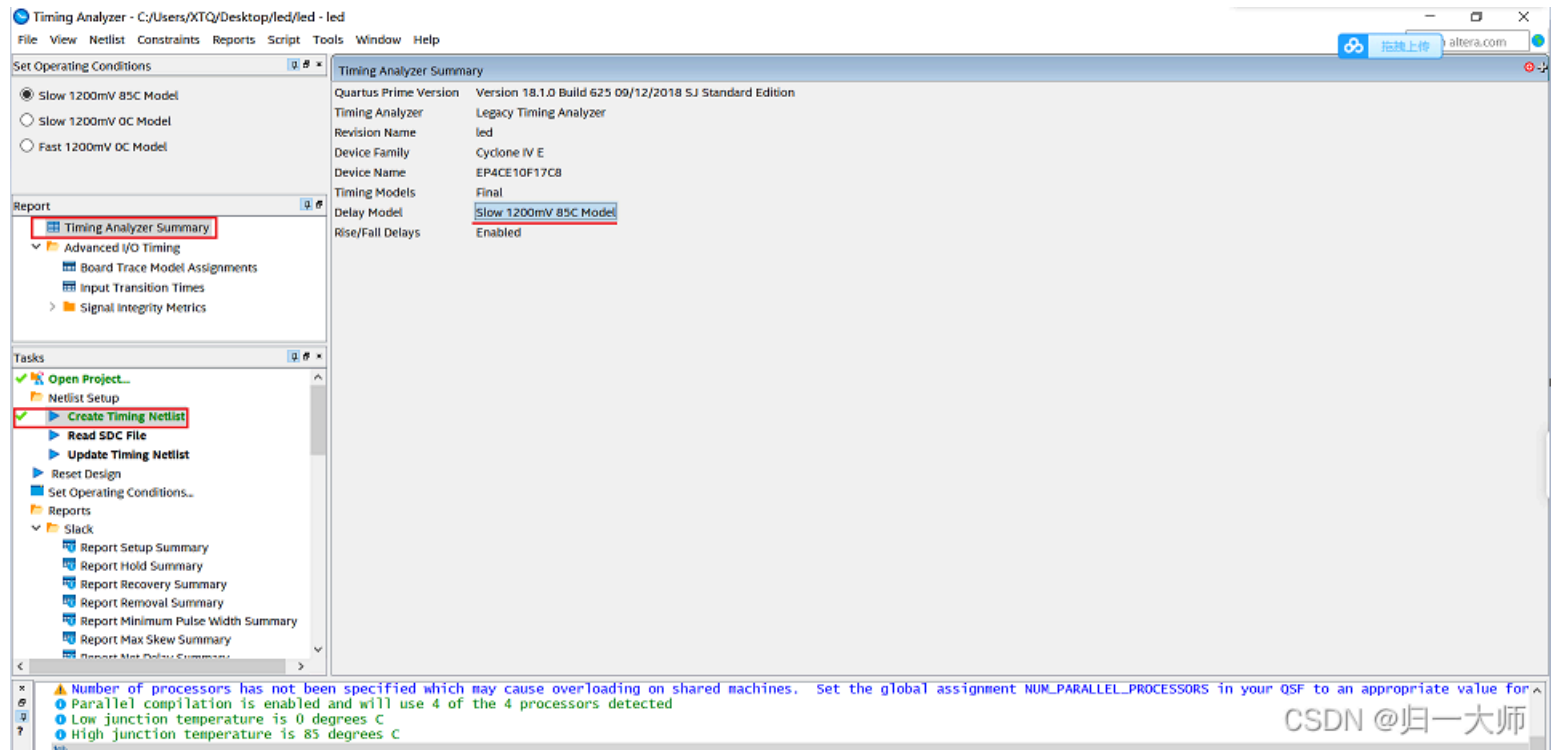


图10 TimeQuest创建网表

这种情况下，默认创建的就是 Slow 1200mV 85C Model，创建完成后该选项会变成绿色，表示创建完成，同时，在这一栏的上方，也就是 Report 栏里，会出现各种当前以及能提高的报告。如果选中 TimeQuest Timing Analyzer Summary 选项，就能看的当前使用的延时模型确实是 Slow 1200mV 85C Model。

2.4.3、读取时序约束文件（Read SDC File）

什么是读取时序约束文件？为什么要读取时序约束文件？

先说时序约束文件的作用吧，时序约束文件有两个作用，一是指导EDA软件（Quartus）进行布局布线，二是指导时序分析软件（TimeQuest Timing Analyzer）对EDA软件编译得到的网表进行分析报告。

指导EDA软件（Quartus）进行布局布线这个功能很好理解，正如上面提到的一样，如果没有为工程加入时序约束，软件就会默认自动分析出可能的时钟信号，并按照1000MHZ的高频率对该信号进行约束，那么软件在进行布局布线时候，就会认为用户希望所有的逻辑都必须能够稳定运行在1000M HZ的时钟频率下，软件会按照这个指示，尽可能的对工程中的布局布线进行优化，保证编译结果能够尽可能的稳定运行在1000MHZ的频率下（当然这在当前的技术下是不可能实现的）。这就是约束文件的第一个作用，即指导 EDA 软件进行布局布线。

指导时序分析软件（TimeQuest Timing Analyzer）对EDA软件编译得到的网表进行分析报告这个功能也很好理解，通过系统最大运行时钟频率的计算方法，要想知道当前设计实际能够运行在多少MHZ的频率下，需要有一个参考，既当前网表在指定的时钟频率下，寄存器的建立时间余量是多少，有了这个建立时间余量，就能够计算出该逻辑能运行的最高时钟频率了。

这里，我们就可以回答上面的两个问题了。

什么是读取时序约束文件？就是查询了解针对当前网表，添加了哪些约束内容。为什么要读取时序约束文件？就是为了能够基于该时序约束信息对当前的网表进行分析，查看当前的网表是否满足约束的内容，或者说计算出当前的网表实际的表现参数与约束信息之间的关系。

针对上述的操作只需一秒钟，读取时序约束文件的方法也很简单，直接双击“Read SDC File”选项即可。如下图所示：

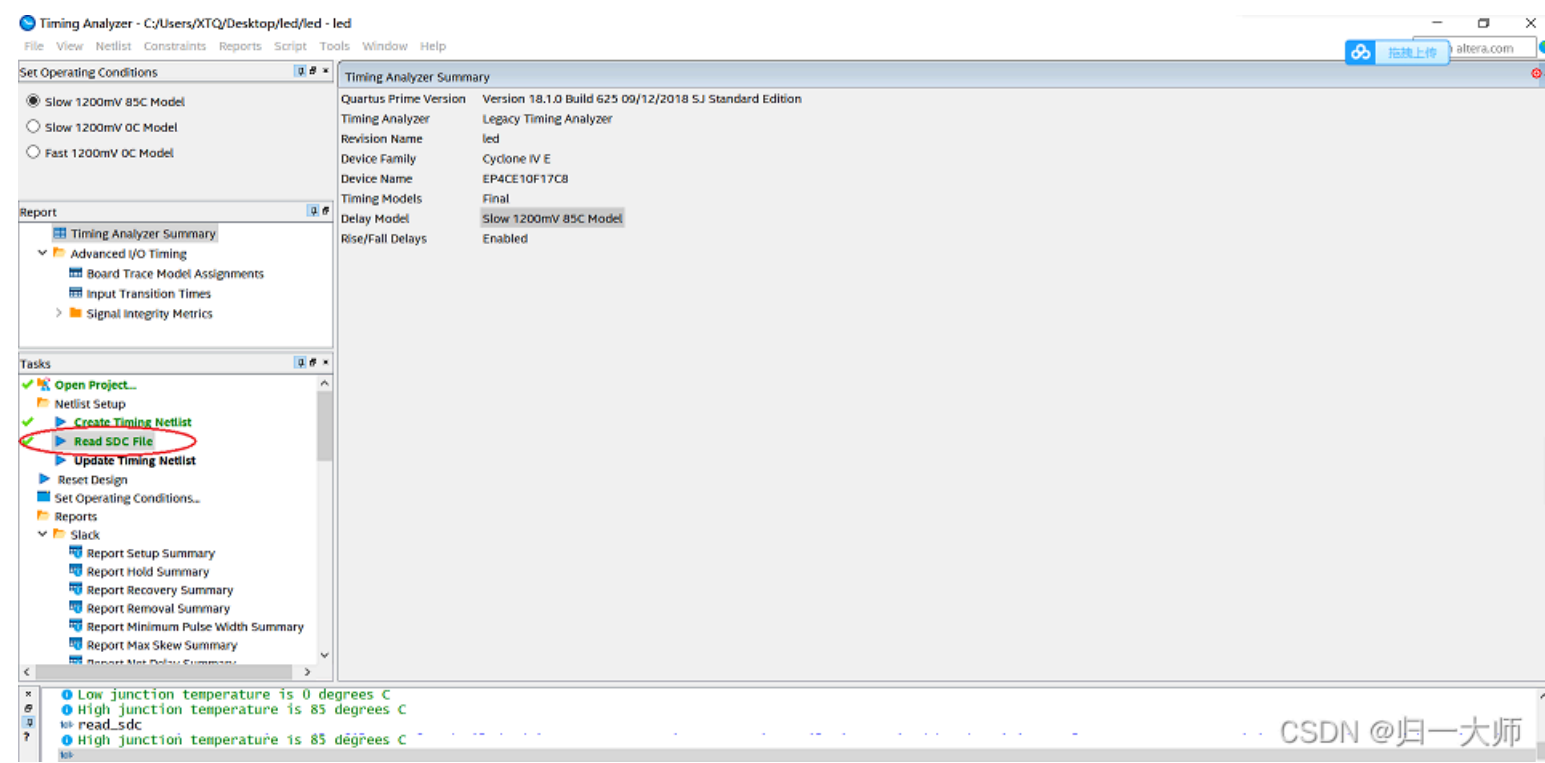


图11 TimeQuest读取工程的时序约束文件

做到这里，相信就会有人质疑了：这个工程并没有加入任何的 SDC 文件，那这个操作，读取的又是什么内容呢？实际上，和前面说的一样，因为没有加约束，所以软件就自己分析出了一个时钟，然后对那个时钟信号添加1000MHZ的约束。只是用户看不到具体的sdc文件而已。这里执行读取 Read SDC File 操作的时候，读取的就是这个看不见的默认约束文件。而如果用户主动为当前工程添加了时序约束文件，那么读取的就是用户添加的约束文件了。

2.4.4、创建时钟 (Create Clock)

创建时钟，严格来说应该叫创建时钟约束，就是为当前网表模型（注意，此时还只是针对网表模型）指定一个明确的时钟信号。这样时序分析软件就能够基于该时钟参数，对网表中的各信号进行分析了。创建时钟（约束）步骤如下所示：

- 1、菜单栏依次点击 Constraints -> Create Clock，打开时钟约束界面。

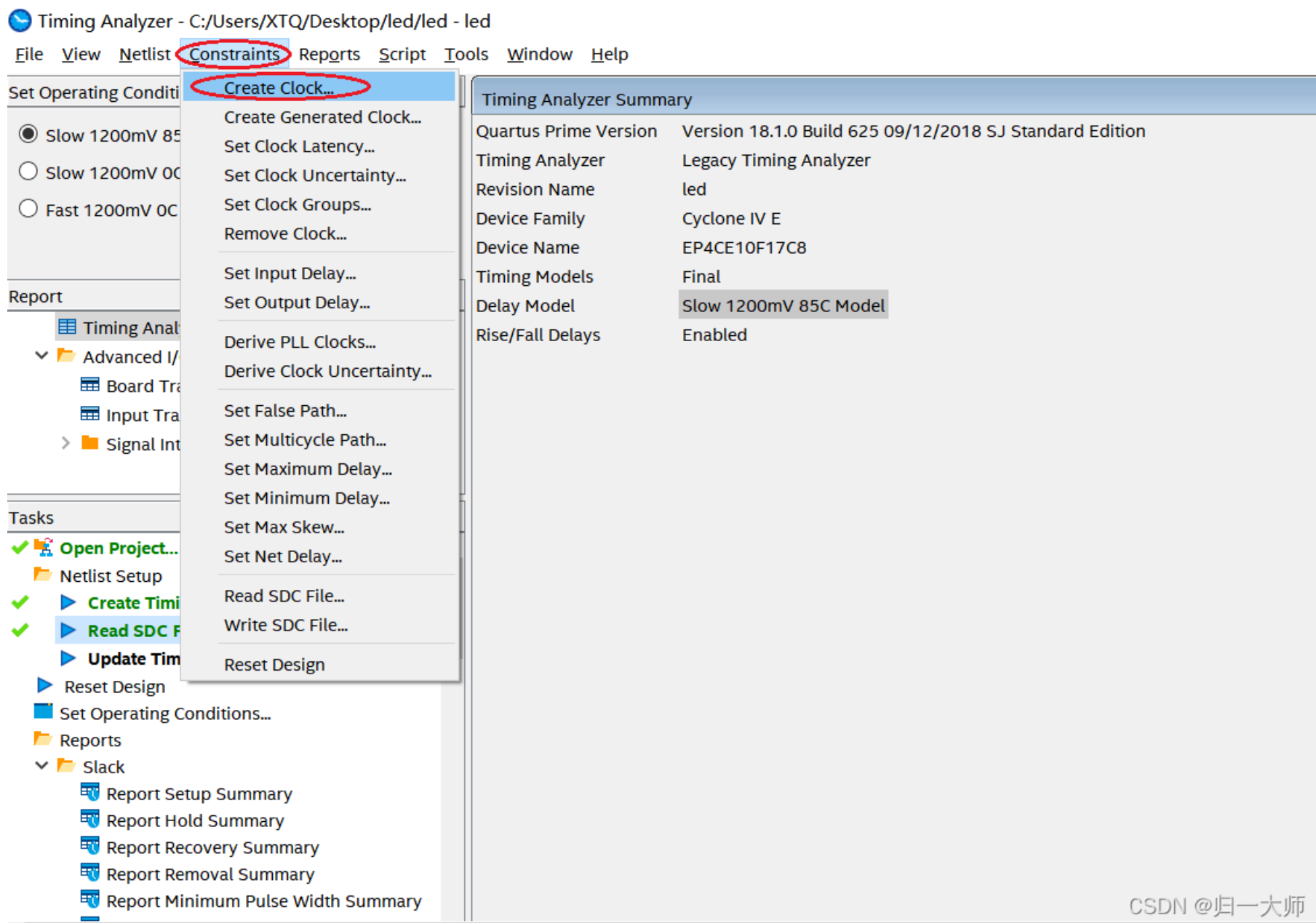


图12 TimeQuest创建时钟约束

2、在弹出的时钟约束图形界面中，输入该时钟的各项属性。如下图所示：

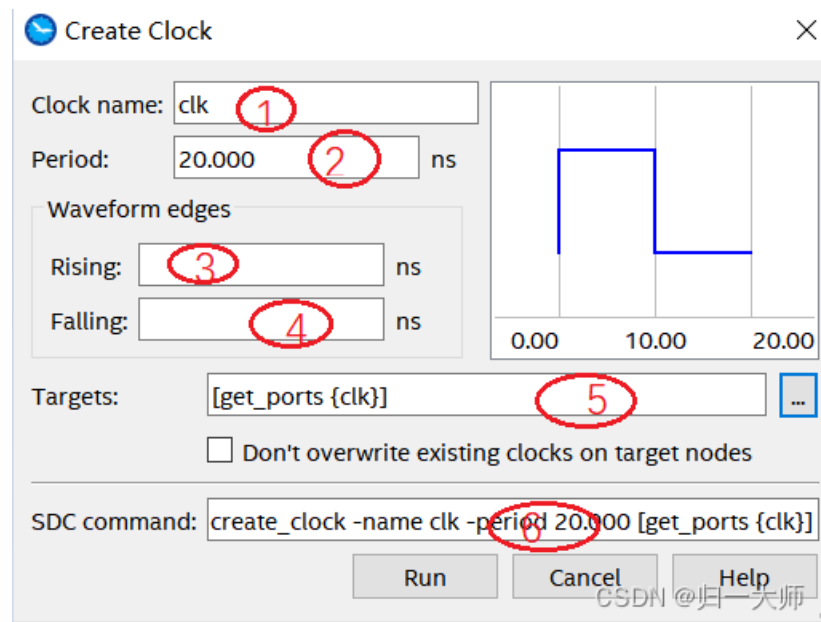


图13 TimeQuest创建时钟约束界面

1) 时钟名称，这里的时钟名称是用户为约束的该时钟信号取一个方便识别的别名，而非被约束的时钟的原本名称。比如周星星是一个特工，组织上叫他007，那么组织上在给他安排任务的时候，称呼他就是007，而不会叫他周星星，当然也可以使用原本名称。本工程中，需要被约束的时钟信号本身名称为 clk，这里约束也为 clk。

2) 周期，时钟周期，clk的时钟频率为 50MHz，所以换算过来其周期就是 $1000/50 = 20\text{ns}$ 。约束的时候，时钟信号的频率不能直接作为参数输入，而是需要换算为周期后再作为参数。

3) 上升时间，这里的上升时间是指什么时刻时钟信号出现上升沿，不写就默认是0时刻了，右侧的波形也能看的出来。

4) 下降时间，这里的下降时间是指什么时刻时钟信号出现下降沿，不写就默认是周期的一半时刻了，右侧的波形也能看的出来。

5) 目标，这才是真正的时钟信号的本名，这才是007的真名——周星星，只不过这个名字在执行任务的时候没有人理会而已。但是组织上为了知道007究竟是哪个人，必须给他建个档案，档案里会写，007，真名周星星。所以这里的目标就是指定约束的这个时钟信号具体是对应的哪个物理上的时钟。

6) 上述所有参数输入好之后，生成的一个真实起作用的脚本命令。换句话说，上面输入的各种参数只是为了方便用户更加方便的添加约束，这些参数输入之后，软件就会根据这些参数生成最终起作用的脚本命令。

3、上述参数输入完毕之后，点击 Run 即可完成创建时钟约束的操作。执行操作后可以看到，软件的信息窗口中执行了前面 UI 界面中展示的那句脚本。

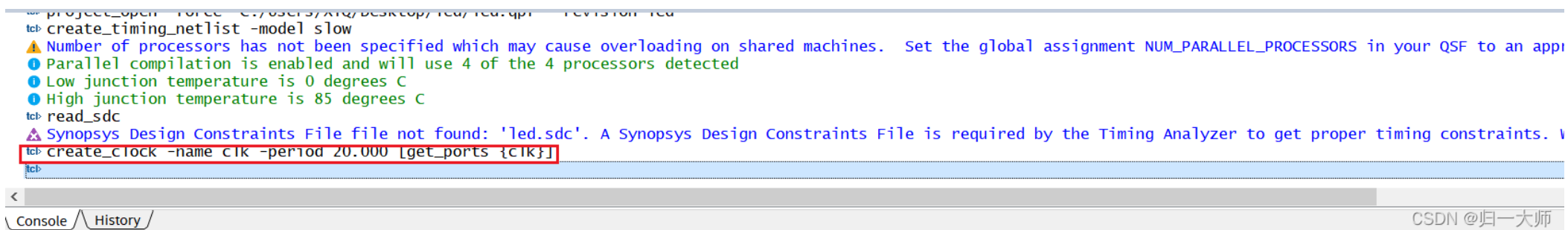


图14 TimeQuest创建时钟约束成功界面

```
create_clock -name clk -period 20.000 [get_ports {clk}]
```

部分Quartus版本可能会出现下面问题：

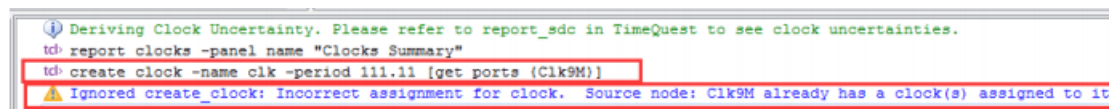


图15 TimeQuest创建时钟约束失败界面

这句脚本下面紧跟着的一条警告信息才是关键，这是都没想到的信息，警告原话是这么说的：

```
Ignored create_clock: Incorrect assignment for clock. Source node: Clk9M already has a clock(s) assigned to it. Use the -add option to assign multiple clocks to this node. Clock was not created or updated.
```

是的，忽略了，EDA软件居然把刚刚的命令给忽略了，为啥呢？就因为 Clk9M这个信号已经被绑定到一个时钟上了，哪里绑定的呢？就是软件默认自动的呀。哎，明明上面的约束才是真正想要的，结果软件居然不认，还是认他自己分析总结到的那个时钟信号。

到这里，也是希望借此例子告诉大家一个事情，那就是每次加入一个约束之后，一定要看工具对这条约束的执行报告，看看这条约束是否被正确执行了，有没有报告其他信息说约束失败之类的。否则很有可能出现上述情况，导致约束并没有生效。

警告说可以通过-add 选项来为此节点创建多个时钟，不能那样做，因为那就违背本意了，本意就是这个节点有且只有一个时钟约束，所以变换思路，既然约束加不了，改总行了吧。怎么改呢？

在 TimeQuest Timing Analyzer 中，找到 Reports 下面的 Diagnostic 下面的 Report Clocks 选项，如下图 16 所示，双击该选项，即可弹出时钟报告窗口，如下图 17 所示。

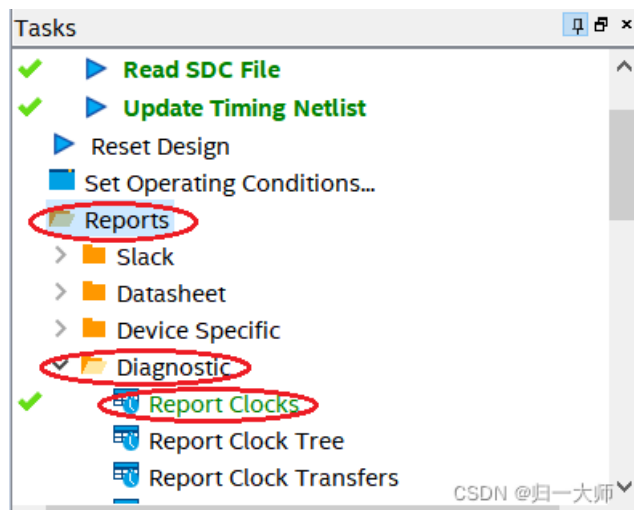


图16 TimeQuest打开时钟报告步骤

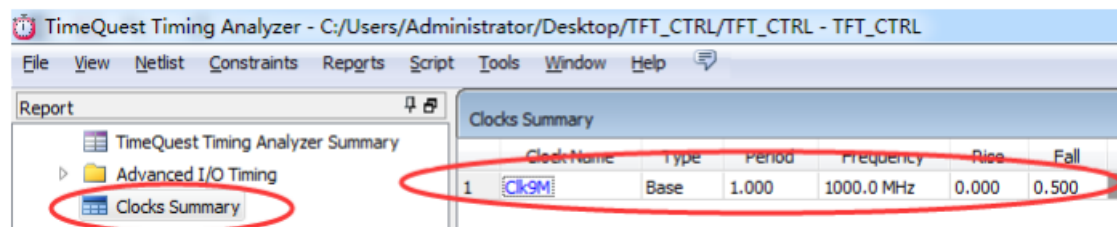


图17 TimeQuest时钟报告界面

上图 17 中可以看到，确实已经存在一个名为 Clk9M 的约束。本来想着，既然有这个约束存在，那就删了再创建，不是就能搞定了么。于是选中该信号，右键，选中 Remove Clock 选项，将其删除。本以为一切操作水到渠成，结果刚删完，然后重新 Report Clocks，这厮马上又回来了。

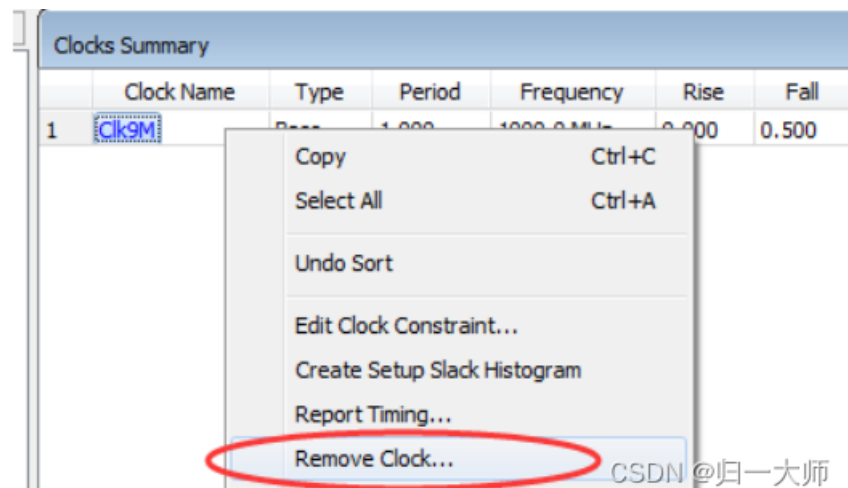


图18 TimeQuest删除时钟约束

删不掉，那试试编辑这个约束，改掉它呢？说干就干，还是选中该信号，右键，选中Edit Clock Constraint...选项，如下图所示：

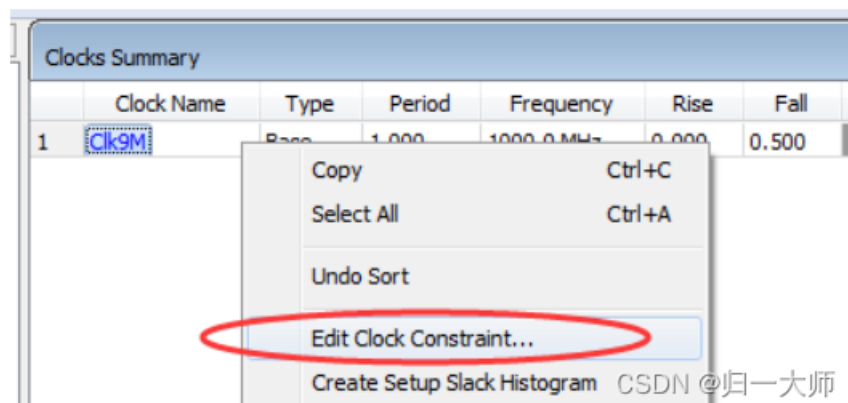


图19 TimeQuest修改时钟约束

点击之后，果然弹出了熟悉的约束编辑画面，如下图所示。

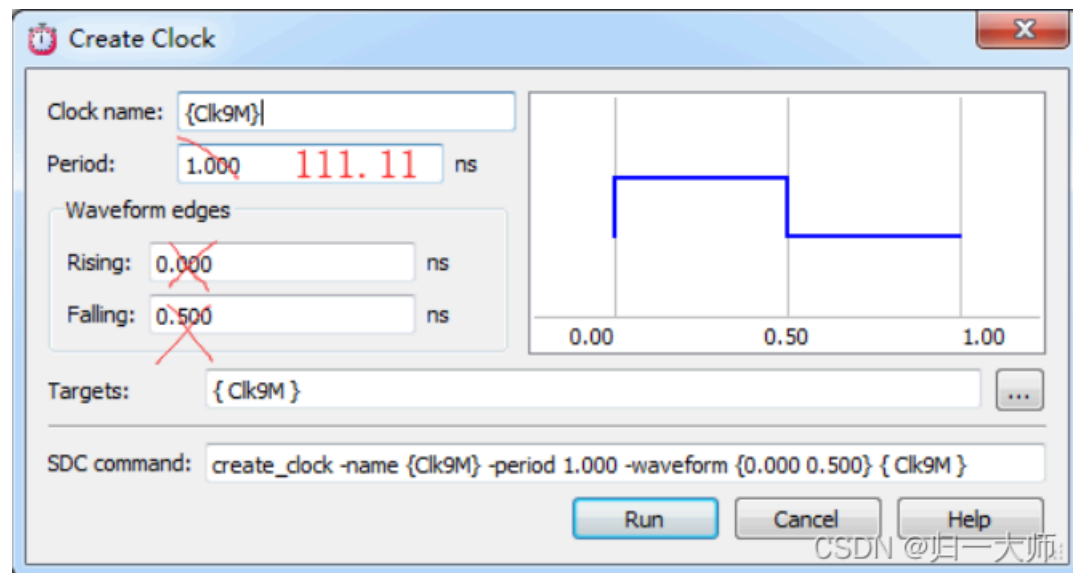


图19 TimeQuest修改时钟约束界面

弹出的编辑框中注意看，要改的地方很多，所以基本上除了 targets 一项不需要动以外，其他都得改。所以按照前面的约束方法，修改时钟约束参数，如下图所示。

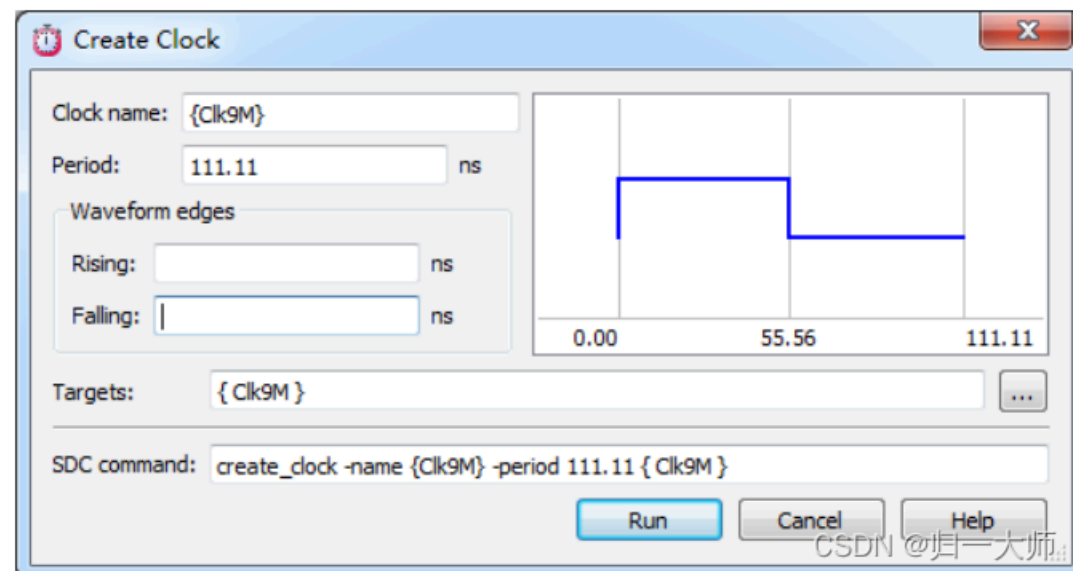


图20 TimeQuest修改时钟约束完成后的界面

• 这里面又有 2 个大坑：

- 1、Clock name 改不得，一改就又认为是新创建时钟，就是不给办，依旧忽略。所以不得已，名字继续保持为系统默认的 Clk9M。
- 2、Rising 和 Falling 两项的值，要改还得自己计算，太麻烦，干脆直接将这两个选项内容删除了，这样就可以让软件自动计算了。

改好之后，再 Run，没有报告忽略的信息了，再 Report Clocks，发现时钟的约束信息终于和设定的是一样的了。

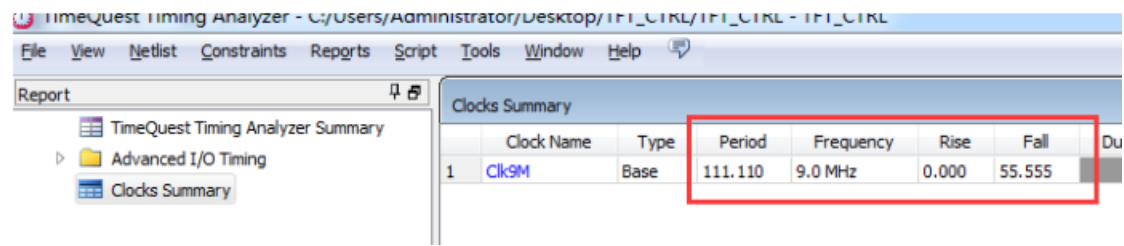


图21 TimeQuest时钟约束修改成功

到此为止，时钟约束就算是完成了。接下来，就应该根据该时钟约束生成对应的 sdc 文件了。TimeQuest Timing Analyzer 提供了将当前的所有 sdc 命令写入到 sdc 文件的方法，在菜单栏依次点击 Constraints -> Write SDC File 可以实现，也可以在 Task 栏中直接双击 Write SDC File 选项。

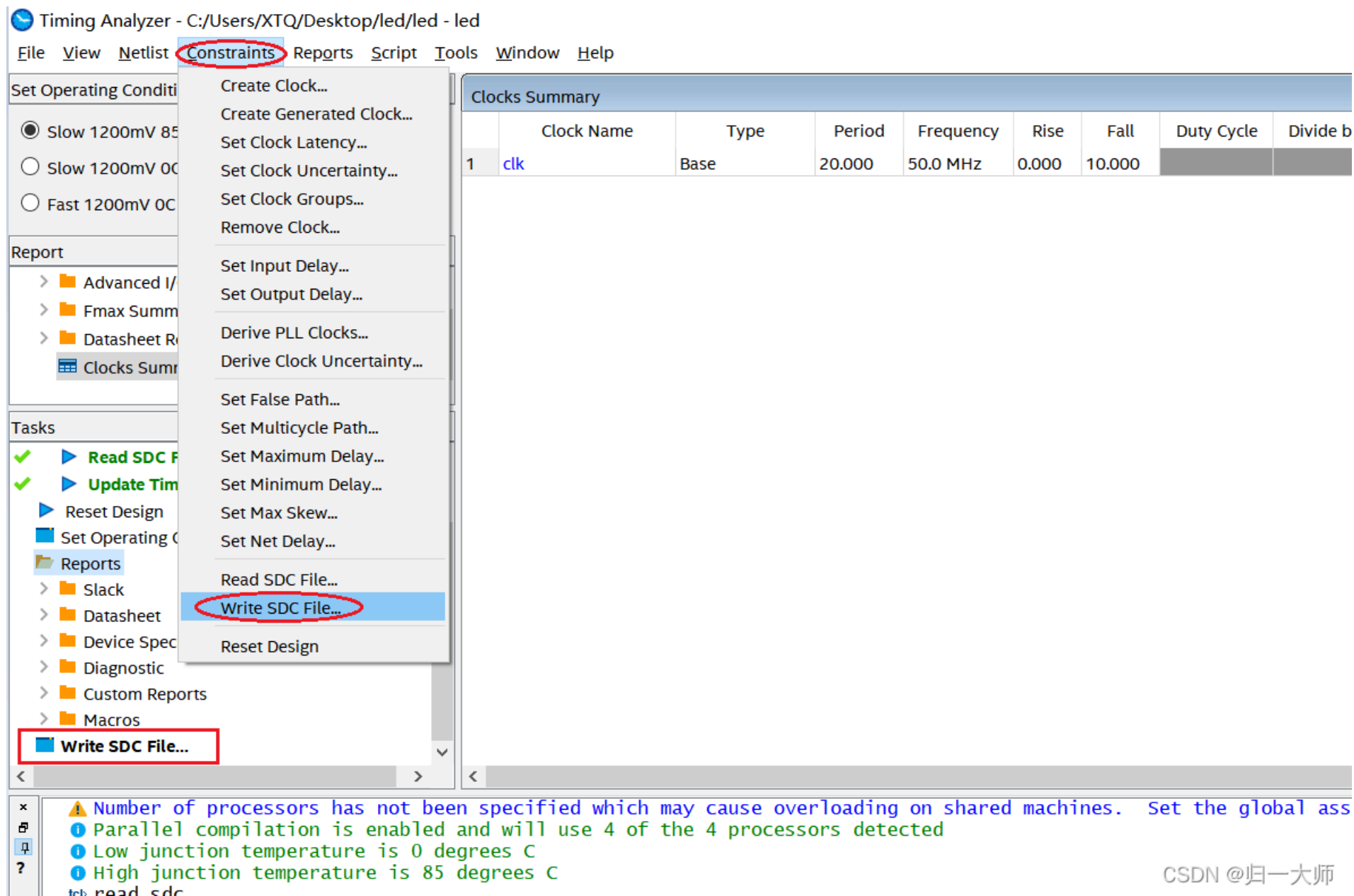


图22 TimeQuest保存时序约束文件

Write SDC File 打开之后，界面如下所示，默认会以工程名称+.out+.sdc 的格式命名该文件，可以把第一项中的.out 删掉，就变成了“led.sdc”，点击 OK，即可自动将约束内容写到“led.sdc”文件里。

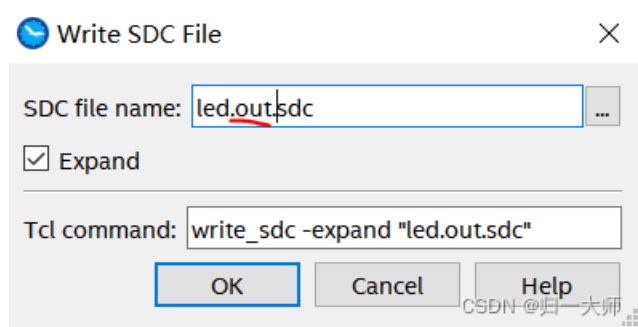


图22 TimeQuest时序约束文件名更改

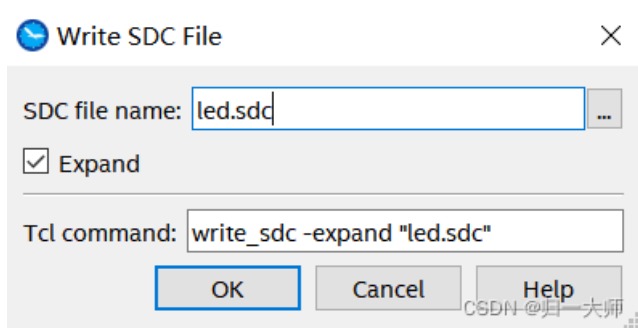


图23 TimeQuest时钟约束文件名修改完

补充说明，如果工程中已经有同名文件，那么该操作会直接使用新内容覆盖文件里原本的久内容。如果工程中没有该同名文件，软件就会新建此文件并将所有约束写入该内容。

上述操作完成后，TimeQuest Timing Analyzer 软件就可以关掉了。但每次关闭的时候，软件又会提示说让用户写 SDC 文件，如下图所示。刚刚已经写过了。直接选No就OK了，选 Yes 无非就是再c重新写一遍。

2.5、使用约束文件指导 Quartus 编译

通过前面的操作，时钟约束也约了，约束文件也创建了。但是这个文件是否就一定能够指导 Quartus 软件进行编译了呢？接下来执行下述几步，以确保该文件已经正确的被用于指导 Quartus 软件布局布线。

在 Quartus 菜单栏中，依次点击 Assignments -> Settings 打开工程设置界面，如下图所示。

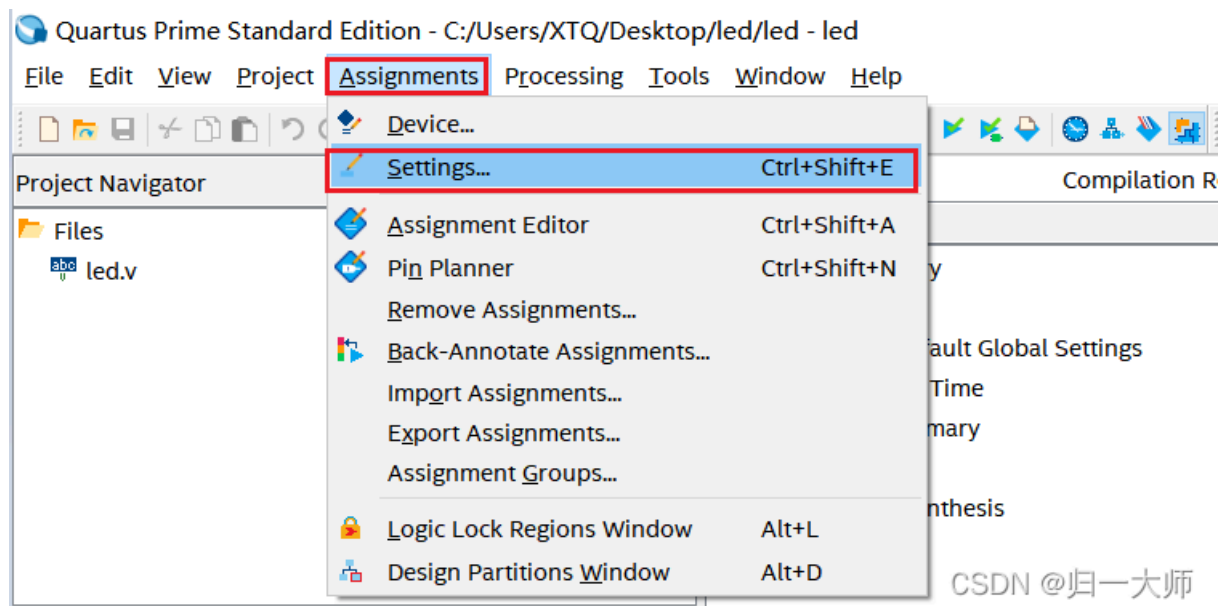


图24 quartus添加时序约束文件步骤1

设置界面如下图所示。点击设置里面的 TimeQuest Timing Analyzer，发现“SDC files to include in the project”里面，没有添加任何文件。要知道，一个工程可以有多个 SDC 文件，而 Quartus 最终使用哪个约束文件，一定是要在这里添加的。

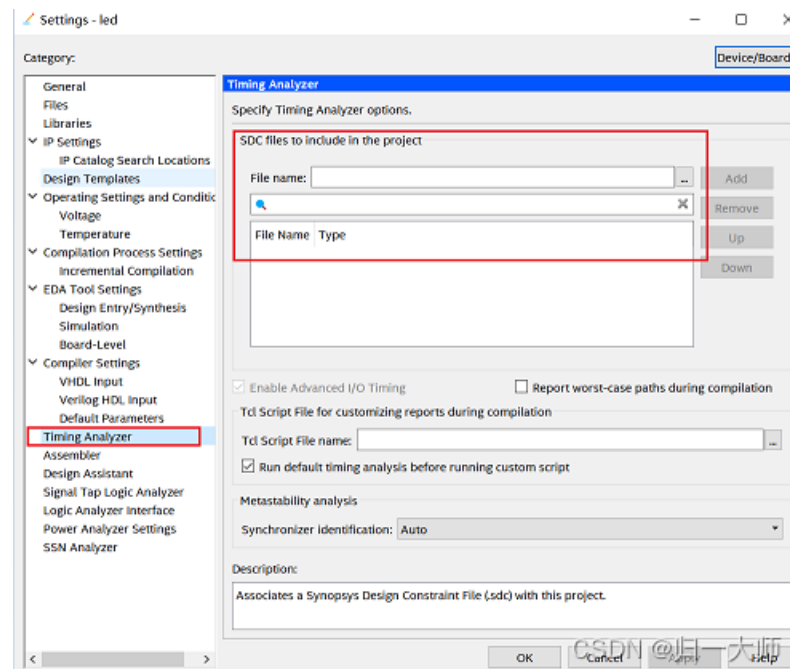


图25 quartus添加时序约束文件步骤2

所以在这里，将刚刚建好的 led.sdc 文件添加进来。如下图所示。

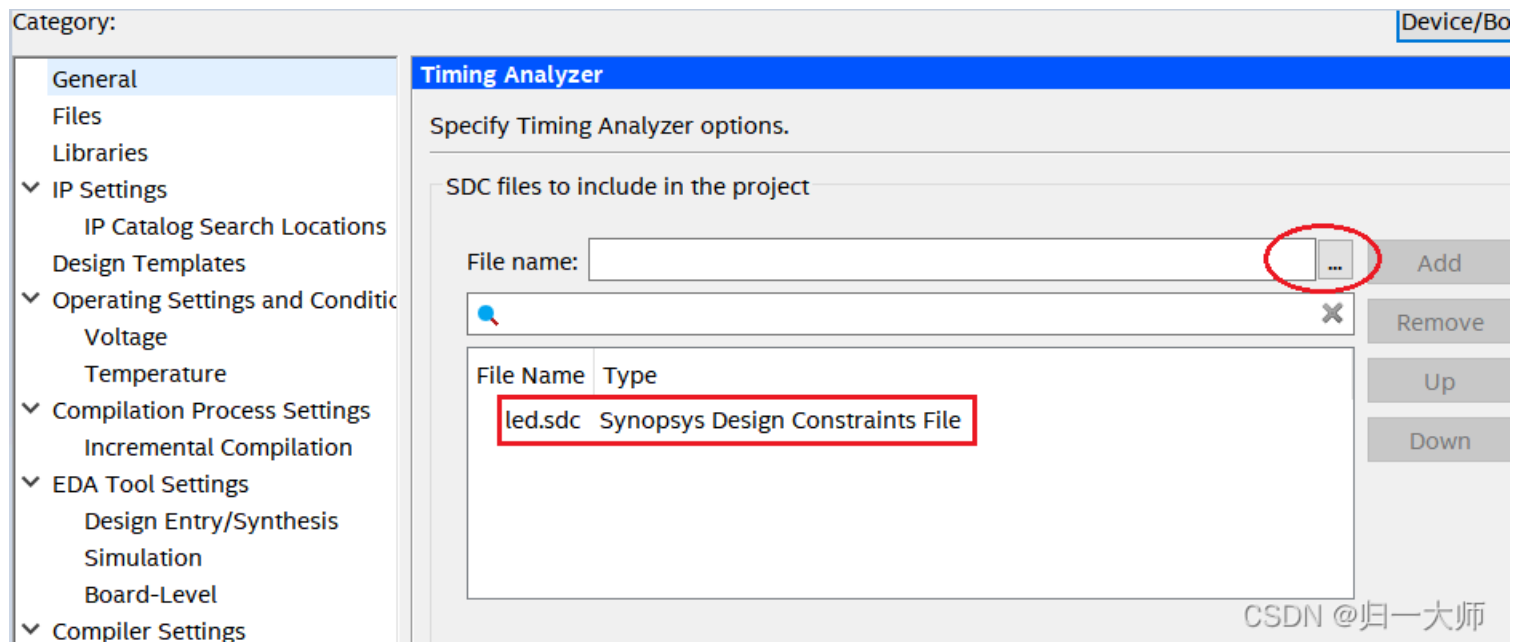


图26 quartus添加时序约束文件步骤3

添加好之后，应用设置并关闭，回到 Quartus 主界面中，会发现该文件已经被加入工程了，然后对工程进行全编译（CTRL + L），软件就会在该约束文件指导下进行编译了。编译完成之后，就可以根据本节内容开头的描述，查看最大运行时钟频率了，新的报告如下图所示。

Quartus Prime Standard Edition - C:/Users/XTQ/Desktop/led/led - led

File Edit View Project Assignments Processing Tools Window Help

led

Project Navigator

Files

- led.sdc
- led.v

Tasks

Compilation

| | Task | |
|---|--|-------|
| ✓ | Compile Design | 00:00 |
| ✓ | > Analysis & Synthesis | 00:00 |
| ✓ | > Fitter (Place & Route) | 00:00 |
| ✓ | > Assembler (Generate programming files) | 00:00 |
| ✓ | > Timing Analysis | 00:00 |
| ✓ | > EDA Netlist Writer | 00:00 |
| | Edit Settings | |

Compilation Report - led

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- > Analysis & Synthesis
- > Fitter
- > Assembler
- > Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - Slow 1200mV 85C Model
 - Fmax Summary
 - Timing Closure Recommendations
 - Setup Summary
 - Hold Summary

Slow 1200mV 85C Model Fmax Summary

<<Filter>>

| | Fmax | Restricted Fmax | Clock Name | Note |
|---|------------|-----------------|------------|------|
| 1 | 168.49 MHz | 168.49 MHz | clk | |

CSDN @归一大师

图27 quartus编译含有SDC文件工程后查看最大系统时钟频率

然后你会发现一个真理，Quartus 编译出来的结果，能够运行的最大频率与约束息息相关，约束要求不高，Quartus 也就随便编译优化下，能满足添加的约束要求即可。不会去做最优。

当然，也可以双击 led.sdc 文件，看看里面的内容，里面内容最关键的就是下面这句了。

```
create_clock -name {clk} -period 20.000 -waveform { 0.000 10.000 } [get_ports {clk}]
```

您的支持是我更新的最大动力！将持续更新工程，如果本文对您有帮助，还请多多点赞👍、评论💬和收藏★！