

实验四：使用按键和LED，实现自定义功能的数字系统（必须使用状态机）

1. 案头准备——确定实验项目并审核
2. 上机过程——在Proscise V2020.7平台上实现项目的电路图（Verilog文件）
3. 下载实验板——将电路图下载到FPGA上进行模拟、仿真
4. 实验报告提交——按照实验指导书完成最终实验报告并在规定时间内提交，报告形式为电子版（包括原代码）。

最终的实验报告：

封页

标题：（设计项目的名称）

姓名及同组成员：学号、姓名、班级

日期：

联系电话：

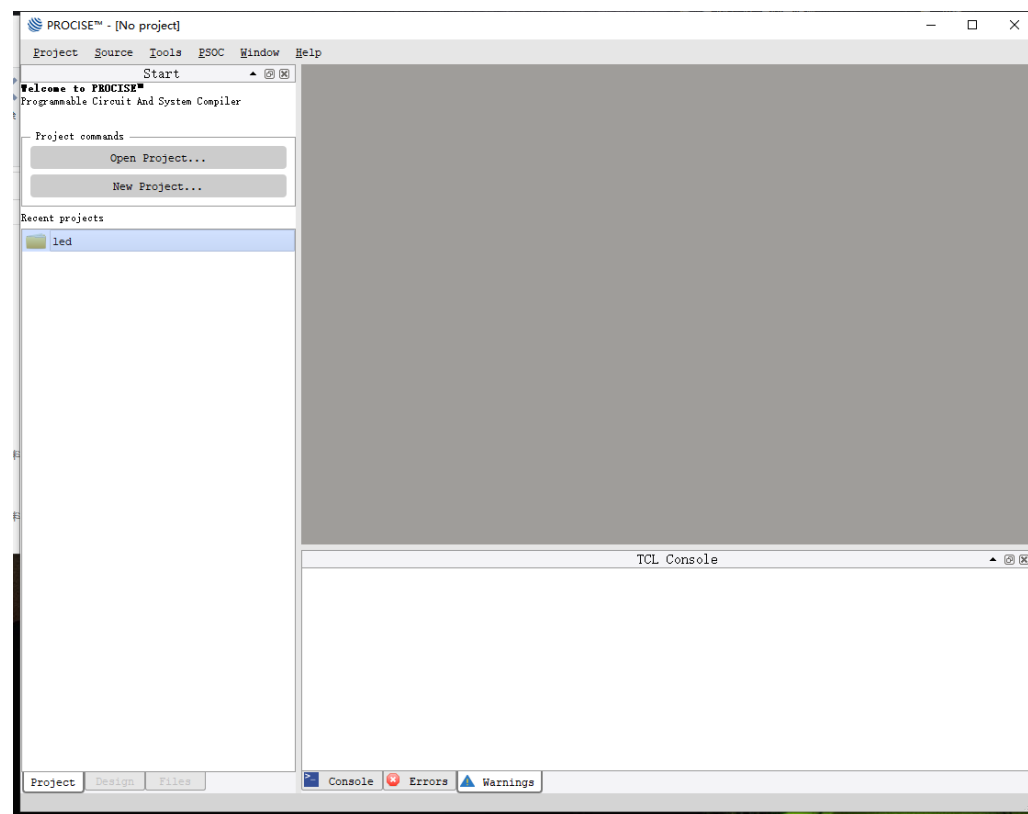
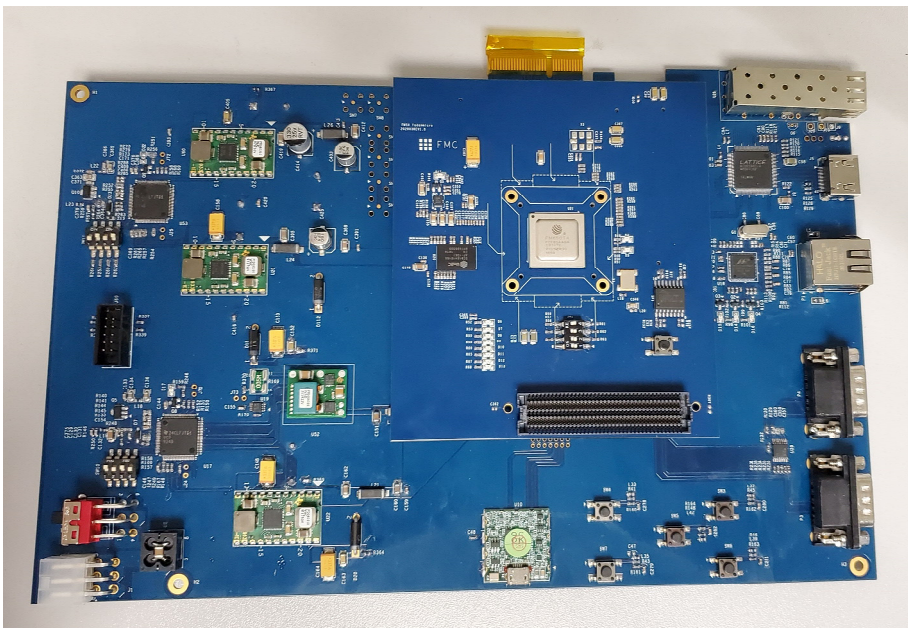
报告的正文：

- 1、项目的目的与目标。
- 2、项目的概要介绍。
- 3、系统详细设计方案：模块图、状态图、状态表等，及各模块说明；项目设计的特点（或技术难点）；各种逻辑图、表达式、Verilog代码的分析及其相关说明等；输入、输出信号的时间图（时序图）。
- 4、测试结果的分析与结论：模拟仿真时各种输入、输出信号的时间图（时序图）；实验的测试结果的讨论：包括实验中间结果、仿真的最终结果的分析，是否达到预期的目的与效果。
- 5、项目的总结：对设计实现的项目进行评价，总结经验，尤其是对项目的进一步完善提出意见。

本次实验平台

硬件平台：FuDan Micro新型数字电路开发板

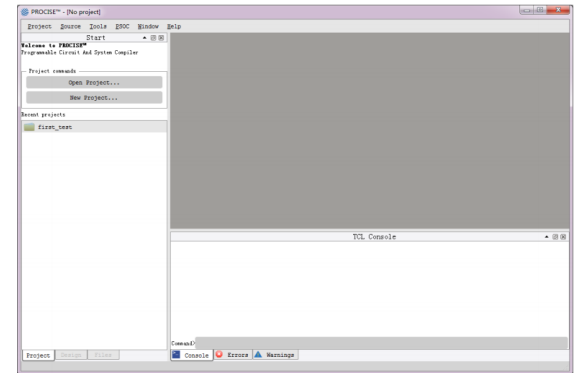
软件平台：Procise V2020.7



软件使用——GUI模式

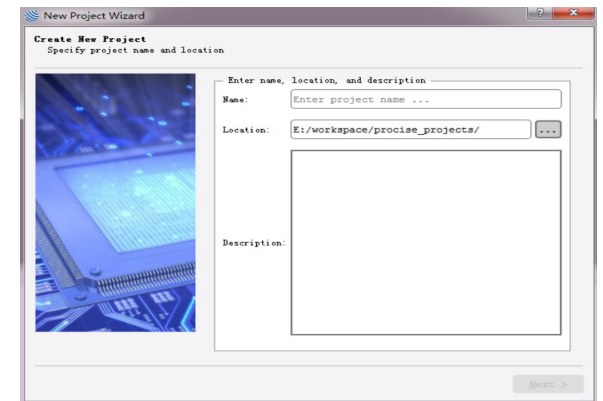
(一) 启动软件

- 双击桌面上 Procise 快捷方式或选择开始 -> 所有程序 -> Procise -> Start Procise 来启动软件。
- 启动后的界面如下图所示：

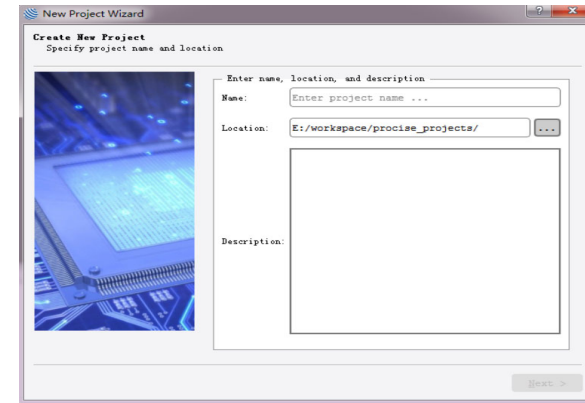


(二) 新建工程

- 通过点击 Project tab 中的 New Project...按钮；
- 或点击菜单栏中Project -> New...选项，弹出 New Project Wizard 界面，进入进入 “Create New Project” 窗口：



1) 给工程命名，输入存放路径，
进入 New Project Wizard 界面下的
“Project Settings” 窗口：



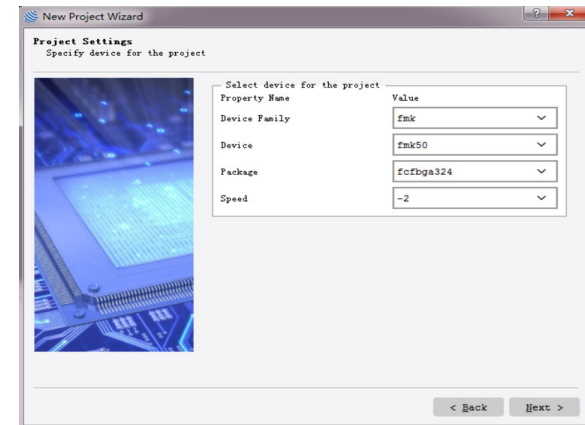
2) 选择参数，点击 “Next”，进入
“Assign source file” 窗口：

【Device Family】：Device 所属类别。

【Device】：工程所用 Device 名字。

【Package】：Device 封装类型。

【Speed】：Device 速度。



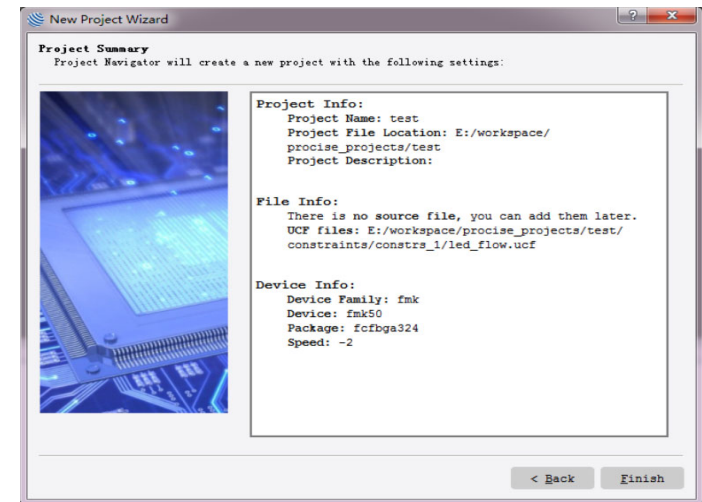
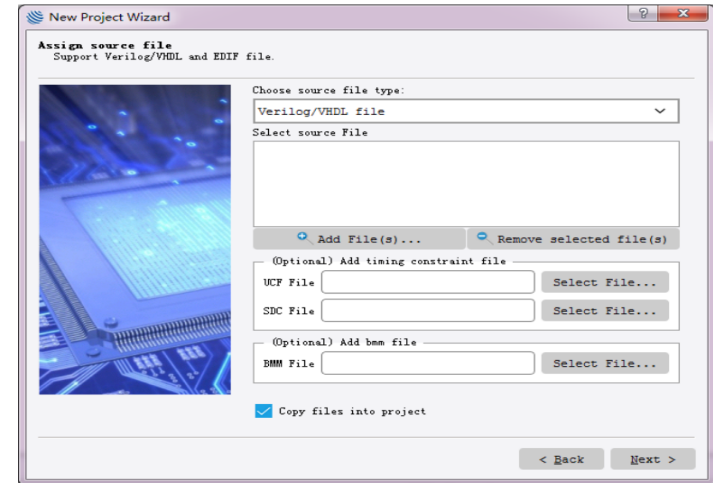
3) Assign source file

在Choose source file type项选择源文件类型：Verilog/VHDL和 EDIF 。

Select source file: 添加源文件或删除已选中的源文件

选择工程所需的 UCF 文件、 SDC 文件、 bmm 文件。

文件选好添加之后，点击“Next”：



该界面总结了新建工程的相关信息。点击Finish，工程新建完成。

（三）文件操作

Procise 支持新建、添加、打开、移除、编辑文件等操作。

1) 新建文件：三种途径

- 菜单栏中 Source -> New Source...; Hierarchy 窗口，点鼠标右键，选择 New Source...; Files tab 中，选择任意文件，或鼠标在空白处，点击右键，选择 New Source...;

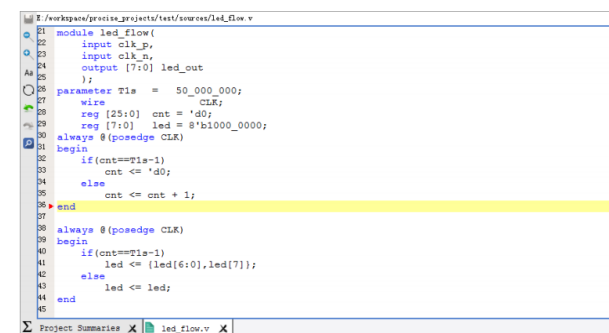
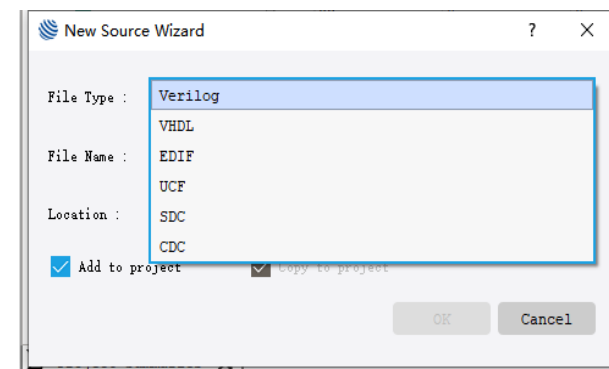
Procise支持六种文件：Verilog、VHDL、EDIF、UCF、SDC、CDC

2) 添加文件：同新建文件中三种途径，选择“Add Source”

3) 打开文件：在 Hierarchy 窗口或 Files tab 中选择想要打开的文件，点鼠标右键，选择 Open; 或在菜单栏中选择 Source -> Open。

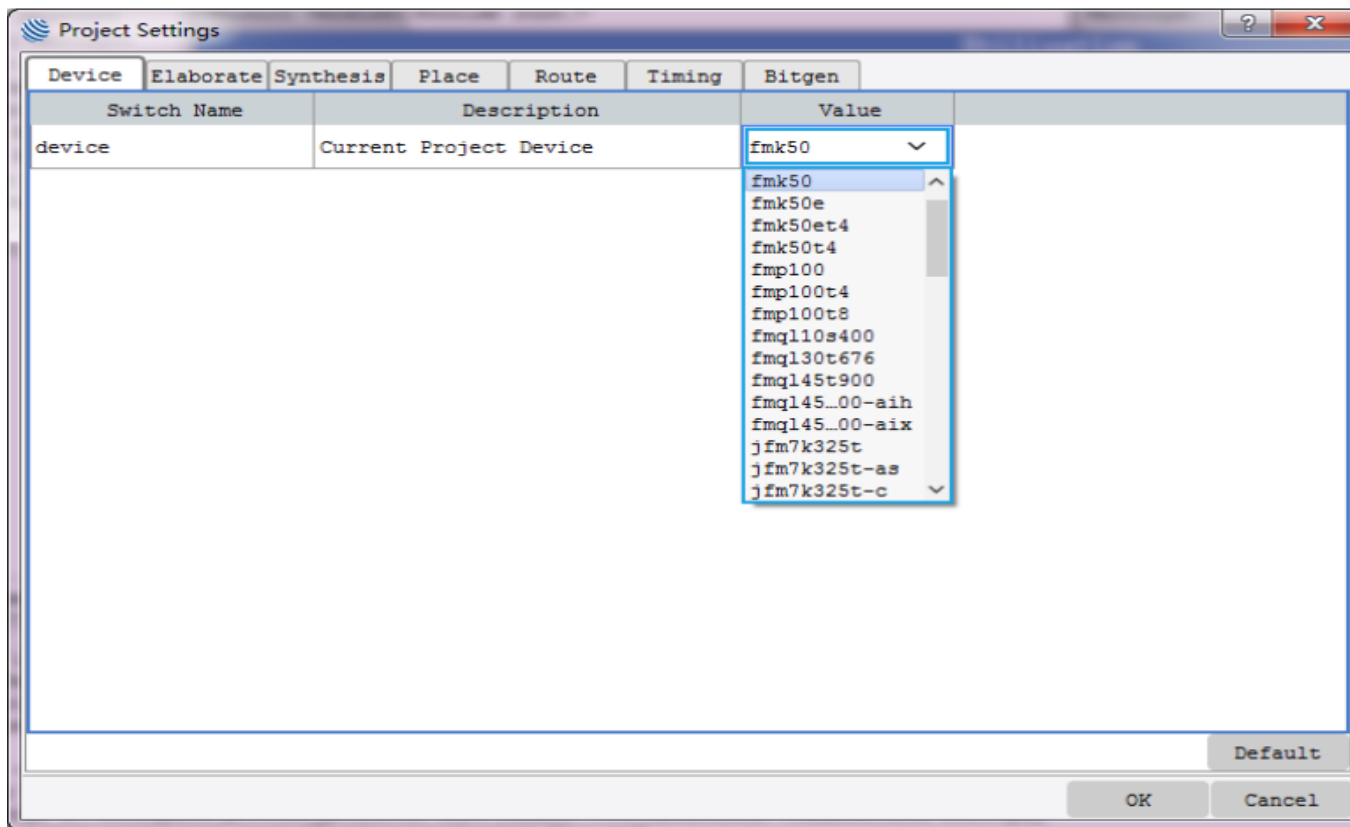
4) 删除文件：在 Hierarchy 窗口或 Files tab 中选择想要从工程中移除的文件，点鼠标右键，选择 Remove; 或在菜单栏中选择 Source -> Remove。

5) 编辑文件：打开需要编辑的文件可直接做增删改查操作。



（四）工程操作

Procise 支持在创建工程之后更改 device, package 和 speed 信息。通过选择菜单栏中 Project -> Project Settings, 打开对话框进行更改。



点击“value”下拉按钮，即可查看并更改当前工程为目标 device。

（五）RTL综合

RTL 综合是将 Verilog/VHDL 描述转换成门级网表的过程。在 GUI 界面上双击 Elaborate，或选中 Elaborate，点鼠标右键，选择 Run，可执行 Elaborate 操作。

Elaborate 阶段支持的配置选项有：

- -infer_ram：支持 On 和 Off 两个选项，On 为默认选项。
 - 选择 On，用户设计中可综合为 RAM 的逻辑用 Device 提供的 RAM 实现。
 - 选择 Off，用户设计中可综合为 RAM 的逻辑用 Device 提供的 SLICE 实现。
- -infer_rom：支持 On 和 Off 两个选项，On 为默认选项。
 - 选择 On，用户设计中可综合为 ROM 的逻辑用 Device 提供的 RAM 实现。
 - 选择 Off，用户设计中可综合为 ROM 的逻辑用 Device 提供的 SLICE 实现。
- -infer_dsp：支持 On 和 Off 两个选项，On 为默认选项。
 - 选择 On，用户设计中可综合为 DSP 的逻辑用 Device 提供的 DSP 实现。
 - 选择 Off，用户设计中可综合为 DSP 的逻辑用 Device 提供的 SLICE 实现。
- -infer_srl：支持 On 和 Off 两个选项，On 为默认选项。
 - 选择 On，用户设计中可综合为 SRL 的逻辑用 Device 提供的 SRL 实现。
 - 选择 Off，用户设计中可综合为 SRL 的逻辑用 Device 提供的 FF 实现。
- -mux_opt：支持 On 和 Off 两个选项，On 为默认选项。
 - 选择 On，打开数据选择器优化。
 - 选择 Off，关闭数据选择器优化。
- -share_opt：支持 On 和 Off 两个选项，On 为默认选项。
 - 选择 On，打开资源共享优化。
 - 选择 Off，关闭资源共享优化。
- 配置 Elaborate 参数有两种方式：
 - 一是菜单栏 Project -> Project Settings -> Elaborate tab;
 - 二是选中 Elaborate，点鼠标右键，选择 Settings。

（六）逻辑优化

1) 逻辑优化为 RTL 综合的下一个关键步骤，把 RTL 综合后未优化的门级网表，通过面积优化、时序优化以及工艺映射 3 个子步骤，转化为由 FPGA 器件支持的库单元组成的逻辑网表。在 GUI 界面上双击 Synthesize，或选中 Synthesize，点鼠标右键，选择 Run，可执行 Synthesize 操作。

2) Synthesize 阶段支持的配置选项有：-bufg_limit_cnt、-lut_mapping 和-iobuf:

- -bufg_limit_cnt: 用户可指定设计中最多可用 bufg 数量，默认数量为 16
- -lut_mapping: 选中该选项，Procise 将在逻辑优化阶段执行逻辑化简及工艺映射。该选项默认选中。若使用第三方综合工具生成工艺映射后的 edif 网表文件创建 Procise 工程，可不选中该选项。
- -iobuf: procise 在逻辑优化的最后阶段插入输入输出缓冲器，该选项默认选中。若不选中，可以作为子模块放到其他工程里。
- 配置 Synthesize 优化参数有两种方式：
 - 一是菜单栏 Project -> Project Settings -> Synthesis tab;
 - 二是选中 Synthesize，点鼠标右键，选择 Settings。

（七）布局

1) Place 阶段将 Synthesize 阶段生成的逻辑单元摆放到器件的具体位置上。在 GUI 界面上双击Place，或选中 Place，点鼠标右键，选择 Run，可执行 Place 操作。

2) Place 阶段支持的配置选项有：-thread thread_num

3) Place 阶段支持多线程运行，用户指定的线程数目与运行 Procise 的电脑 CPU 内核数目相同时效率最高。

4) 进入 Place 配置界面的方式有两种：

- 一是菜单栏 Project -> Project Settings -> Place tab;
- 二是选中 Place，点鼠标右键，选择 Settings。

（八）布线

1) Route 阶段将 Place 阶段生成网表中的连线资源配置到 FPGA 上,以实现用户功能。在 GUI 界面上双击 Route, 或选中 Route, 点鼠标右键, 选择 Run, 可执行 Route 操作。

2) Route 阶段支持的配置选项有: -routability、-high_effort、-dummy 和 swap_pins:

-routability: Route 阶段将不考虑时序约束, 完成布线。该选项在用户设计很大很复杂且时延约束不重要时有用。

-high_effort: Route 阶段会花费更多时间和 cpu 资源去完成布线。该选项在用户设计很大很复杂, 默认条件下布线失败时有用。但该选项会延长布线时间。

-dummy: Route 阶段会花费更多时间去除拥挤从而完成布线, 该选项在用户设计较大较复杂, 默认条件下布线无法去除拥挤导致失败时有用。但该选项会延长布线时间。

-swap_pins: route 阶段会交换 lut 的输入 pin 的连接。该选项默认打开。

3) 进入 Route 配置界面的方式有两种:

- 一是菜单栏 Project -> Project Settings -> Route;
- 二是选中 Route, 点鼠标右键, 选择 Settings。

（九）时序分析

1) 时序分析模块在综合、布局布线等各个阶段，都可以调用。通过在不同层次的网表结构上进行时序计算和验证，给各功能模块提供时序信息，指导各功能模块完成关键路径的时延优化，获得满足用户时序约束的电路设计方案。

2) 在 Procise GUI 界面上，时序分析会在布局、布线后自动执行（前提是设计有时序约束）。

在 GUI 界面上，Timing Constraints 一栏会显示用户设计的时序约束是否满足，Post-Place/Route Static Timing Report 可以分别查看布局/布线后时序分析的详细报告，如下图：

3) 时序分析支持的配置选项有：-max_path 和 -nworst。

-max_paths n: 显示各个 group 的最 worst 的 N 条 path。

-nworst m: 显示每个 end point 最 worst 的 M 条 path。

进入时序分析配置界面的方式有两种：

一是菜单栏 Project -> Project Settings -> Timing；

二是选中 View Post-Map/Place/Route STA，点鼠标右键，选择 Settings。

The screenshot shows the PROCISE GUI with the 'View reports' panel open. The panel displays a table of timing constraints. The table has columns: Net, Constraint, Check, Worst Case Slack, and Timing Errors. The first row shows '1 Yes' for the 'Setup' constraint, with a worst case slack of 4.635ns. The second row shows '2 Yes' for the 'Hold' constraint, with a worst case slack of 0.1199ns. Both rows have a 'Timing Errors' value of 0. A red arrow points to the 'Yes' status in the first row, with the text '时序满足设计需求' (Timing meets design requirements). Another red arrow points to the 'Post-Place Static Timing Report' and 'Post-Route Static Timing Report' links in the 'Detailed Reports' section, with the text '布局布线后详细时序报告' (Detailed timing report after layout and routing).

Net	Constraint	Check	Worst Case Slack	Timing Errors
1 Yes	View Timing Constraints Report	SETUP	4.635ns	0
2 Yes	View Timing Constraints Report	HOLD	0.1199ns	0

时序满足设计需求

布局布线后详细时序报告

（十）位流生成

Generate Bitstream 阶段将 Route 阶段生成的网表转化为可以下载到 FPGA 上的位流文件。在GUI 界面上双击 Generate Bitstream，或选中 Generate Bitstream，点鼠标右键，选择 Run，可执行 Generate Bitstream 操作。

（十一）编程下载

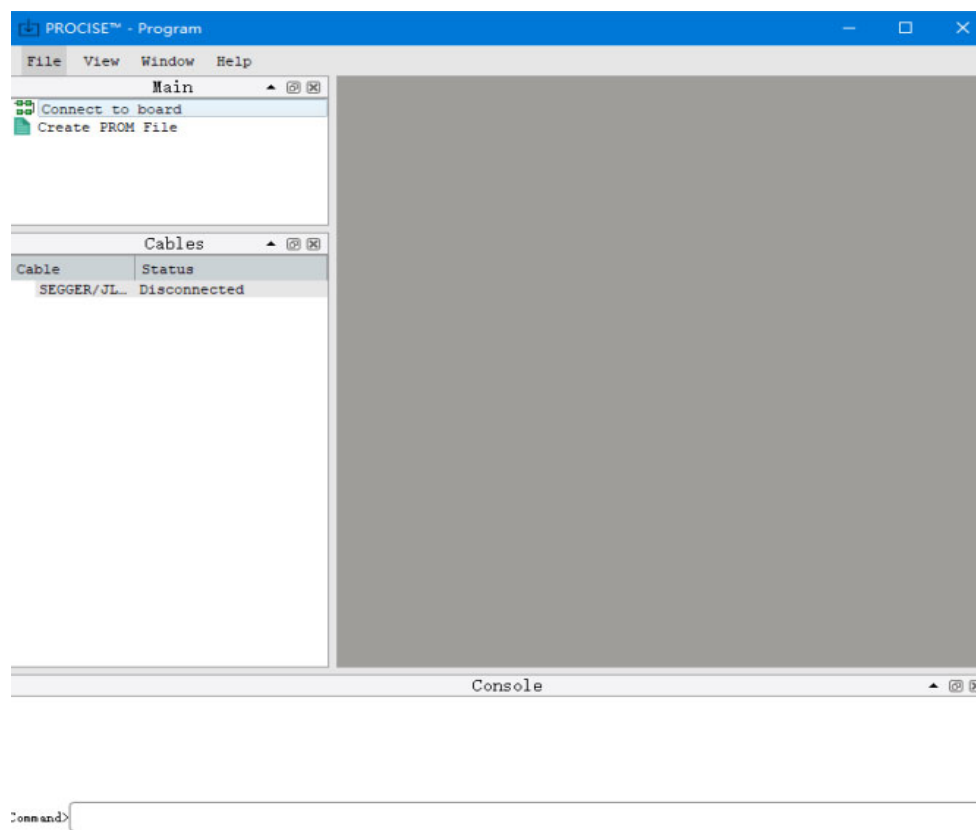
1) 编程下载功能用于将用户产生的 bit 文件下载到相应的 FPGA 芯片上或者将位流文件转换为mcs 文件后再下载到相应的Flash/Rom 芯片上。

2) 进入编程下载界面的方式有两种：

一是在 GUI 界面上双击 Program Device；

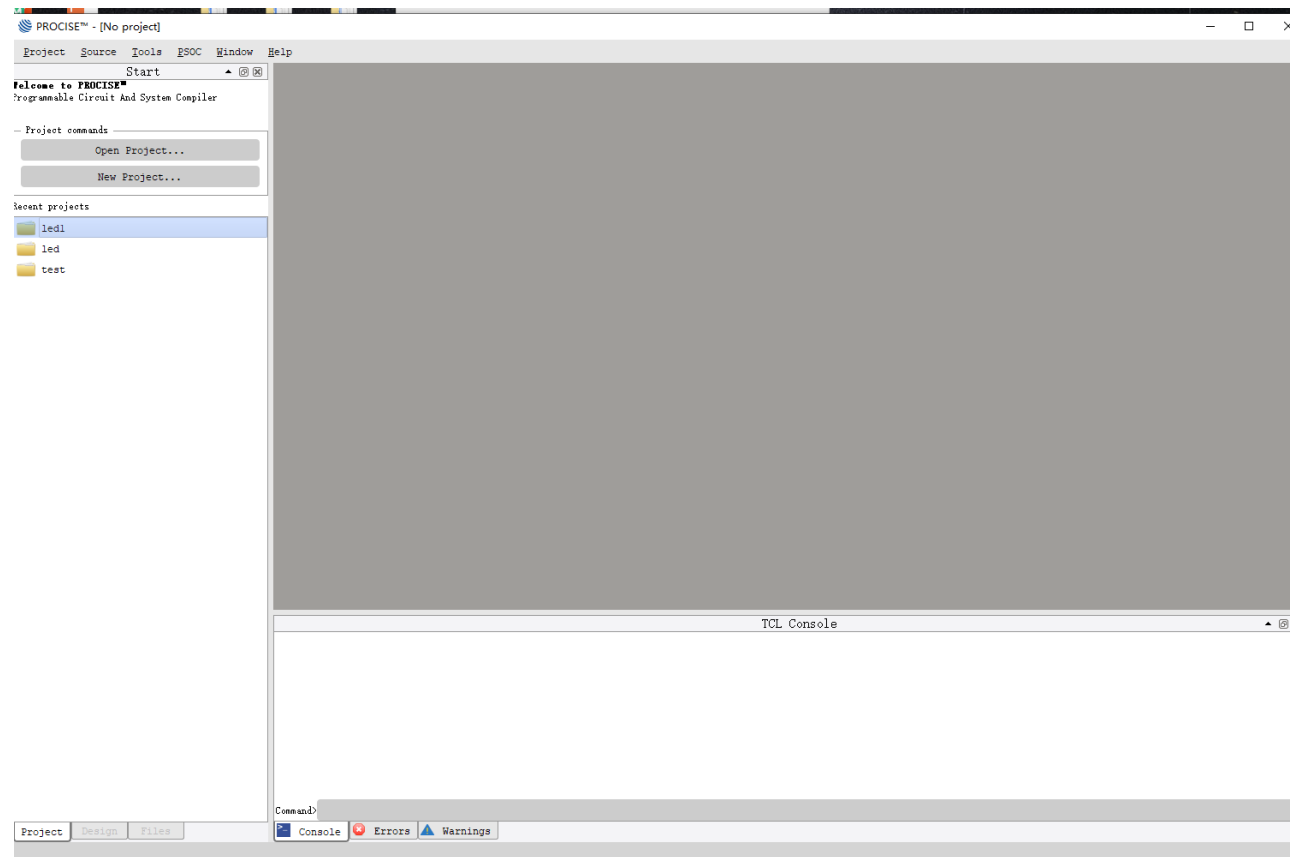
二是在菜单栏中选择 Tools -> Configure Device...

Procise 的下载界面如下图：



例、LED流水灯实验

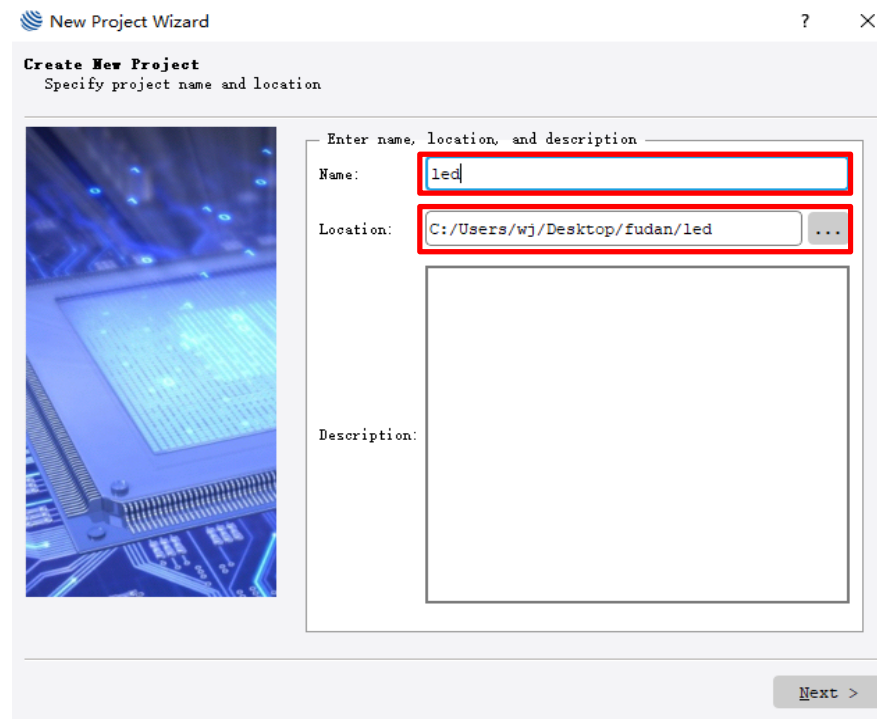
第一步：打开软件



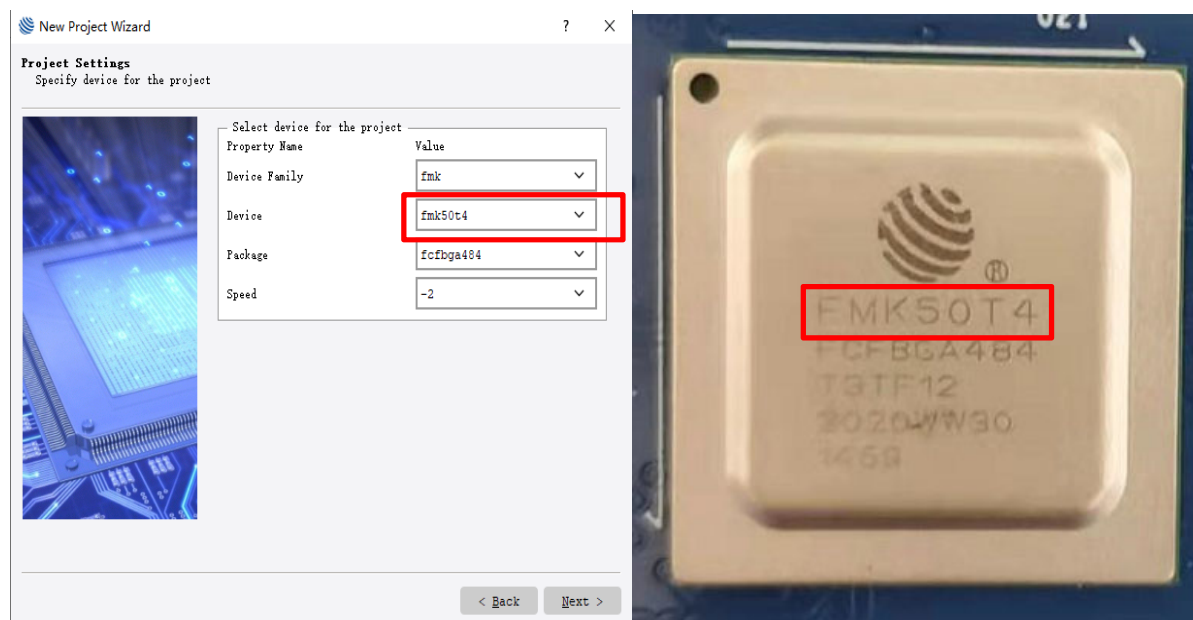
第二步：新建工程（Project tab> New Project...）

1、输入工程名称“led”。

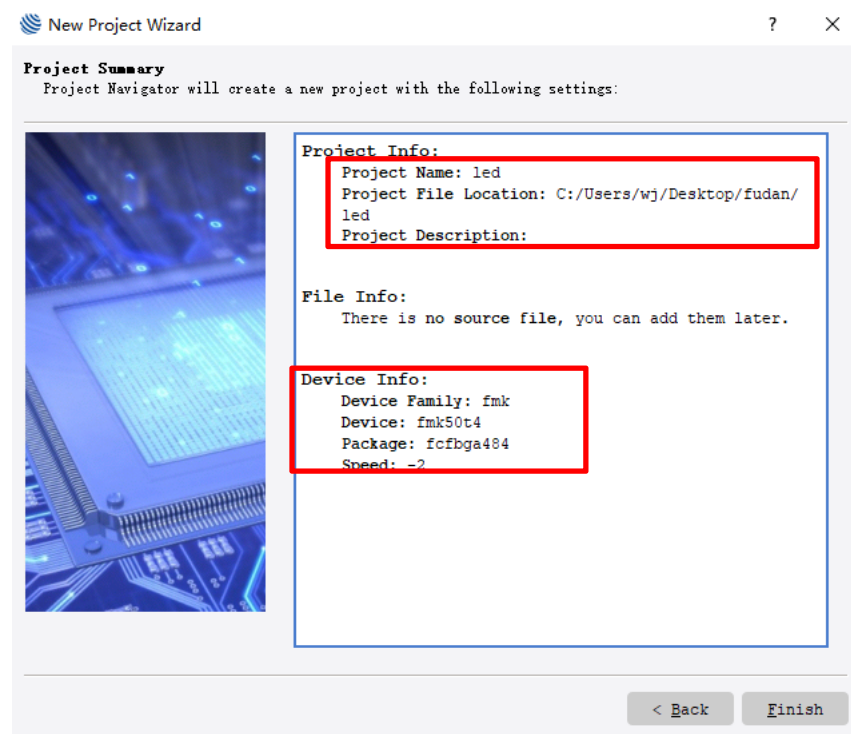
2、输入工程存放路径。



3、器件选择，选择与开发板对应的芯片型号“fmk50t4”，点击“Next”。



- 4、添加文件，如没有准备好的文件，点击“Next”跳过此步骤。
- 5、工程信息如下所示：点击“Finish”。

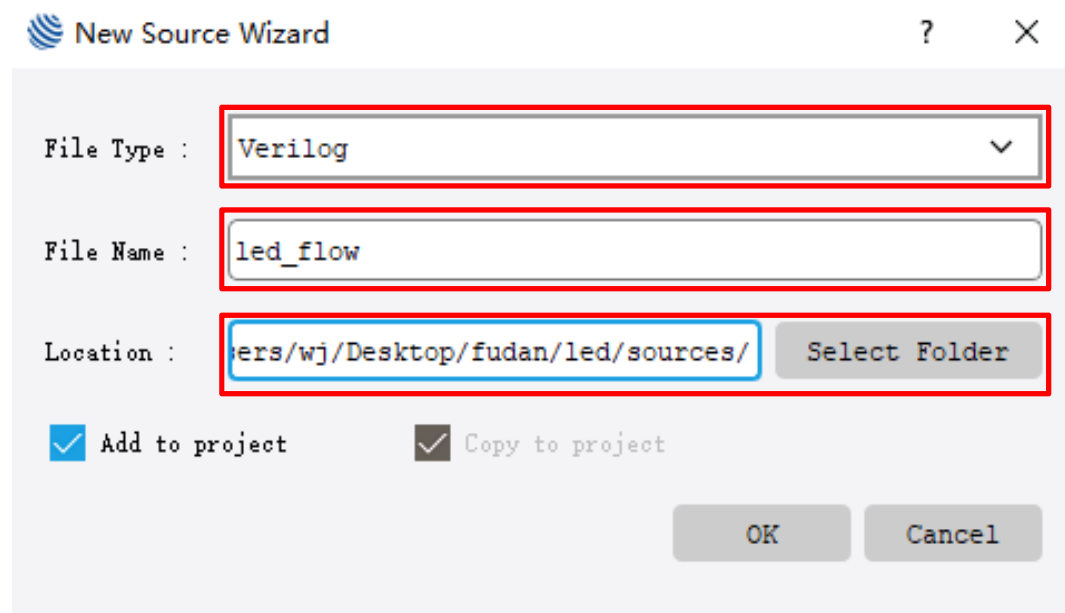


第三步：添加工程文件（Source>New Source...）

1、添加led流水灯源文件，选择文件类型“Verilog”；

文件名“led_flow”；

软件默认文件存放路径为“X:/xxx/xxx/.../led/sources”，点击“OK”。



2、编辑源码，在led_flow.v文件中输入如下源码，点击保存。

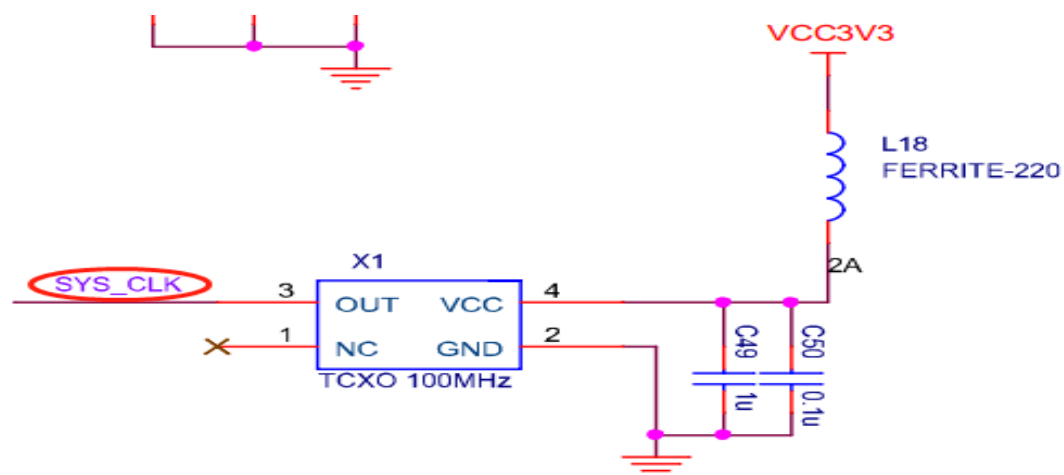
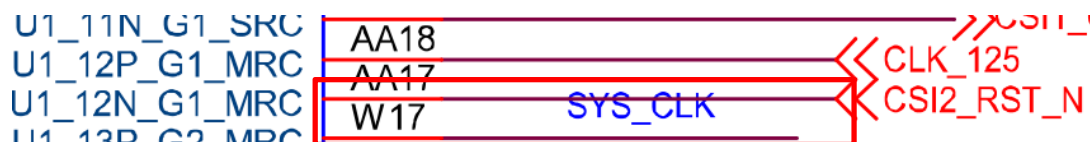
```
...rs/wj/Desktop/fudan/led/sources/led_flow.v  [保存] [上一步] [下一步] [加] [减] [搜索]
1  `timescale 1ns / 1ps
2
3  module led_flow(
4      input clk,
5      output [7:0] led_out
6  );
7      parameter T1s = 50_000_000;
8      reg [25:0] cnt = 'd0;
9      reg [7:0] led = 8'b01111111;
10     always @(posedge clk)
11     begin
12         if(cnt==T1s-1)
13             cnt <= 'd0;
14         else
15             cnt <= cnt + 1;
16     end
17
18     always @(posedge clk)
19     begin
20         if(cnt==T1s-1)
21             led <= {led[6:0],led[7]};
22         else
23             led <= led;
24     end
25
26     assign led_out = led;
27
28 endmodule
```

- 3、添加工程约束文件，选择文件类型“UCF”；
文件名“led_flow”；
软件默认文件存放路径为“X:/xxx/xxx/.../led/constraints/constrs_1”，
点击“OK”。

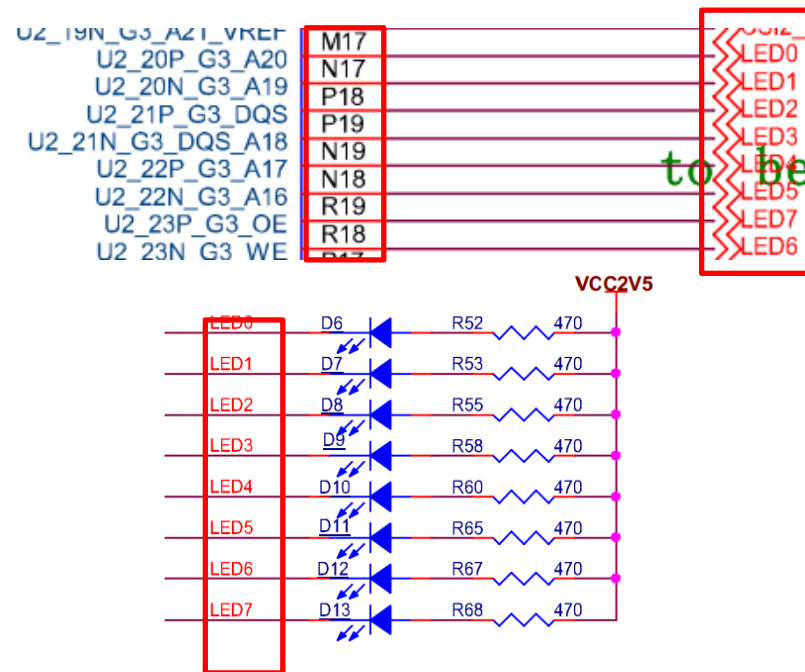


4、编辑工程约束文件:

1) 查看开发板原理图，确定对应的引脚编号。如图，输入“clk”选择100MHz的系统时钟“SYS_CLK”，对应引脚编号为“W17”。



2) 输出“led[7:0]”，查看原理图，确定LED0~LED7分别对应的引脚编号。



3) 在led_flow.ucf文件中添加输入、输出管脚约束及时钟约束，点击保存。
如下所示：

```
..dan/led/constraints/constrs_1/led_flow
1  NET "clk" LOC = W17;
2
3  NET "led_out[7]" LOC = M17;
4  NET "led_out[6]" LOC = N17;
5  NET "led_out[5]" LOC = P18;
6  NET "led_out[4]" LOC = P19;
7  NET "led_out[3]" LOC = N19;
8  NET "led_out[2]" LOC = N18;
9  NET "led_out[1]" LOC = R18;
10 NET "led_out[0]" LOC = R19;
11
12 NET "clk" TNM_NET = clk;
13 TIMESPEC TS_clk = PERIOD "clk" 10 ns HIGH 50%;
```

为主时钟定义PERIOD约束

第四步：编译综合（Run FPGA flow）：点击“Run FPGA flow”，编译综合完成后，将生成led.bit 文件。

```
>>bitgen led.bit -g bpi_sync_mode:Disable spi_32bit_addr:No spi_buswidth:1 Encrypt:No  
preprocess elapsed_time 0.02 seconds, cpu_time 16.00 seconds  
assemble_bitstream elapsed_time 0.03 seconds, cpu_time -16.00 seconds  
Writing out bitstream file led.bit  
writing out bgn file led.bgn  
bitgen elapsed_time 0.93 seconds, cpu_time 1.02 seconds  
bitgen used memory 320MB, procise used peak memory 1353MB, current used memory 510MB
```

The screenshot shows the PROCISE IDE interface. On the left sidebar, under 'Design Utilities', the 'Run FPGA flow' button is highlighted with a red box. The main window displays the 'led Project Status' window, which includes a 'Summary' section with utilization data and a 'TCL Console' at the bottom showing the bitgen command output.

led Project Status

Project File: led.fpe
Module Name: led_flow
Target Device: fmc50t4
Product Version: Procise 2021.1

Parser Errors: 0
Implementation State: Programming File Gener...
Errors: 0
Warnings: 0

Utilization

Instances	LUTs	FFs	CUss	RAMB
32	37	0	0	0

Summary

Instances	LUTs	FFs	CUss	RAMB
32	37	0	0	0

Plot

Slice LUTs	37/32600	0.11%	Logic	37/32600	0.11%
Bonded IOUs	9/250	3.60%	GCPU/GCPU_CTRLs	1/32	3.13%

Usage

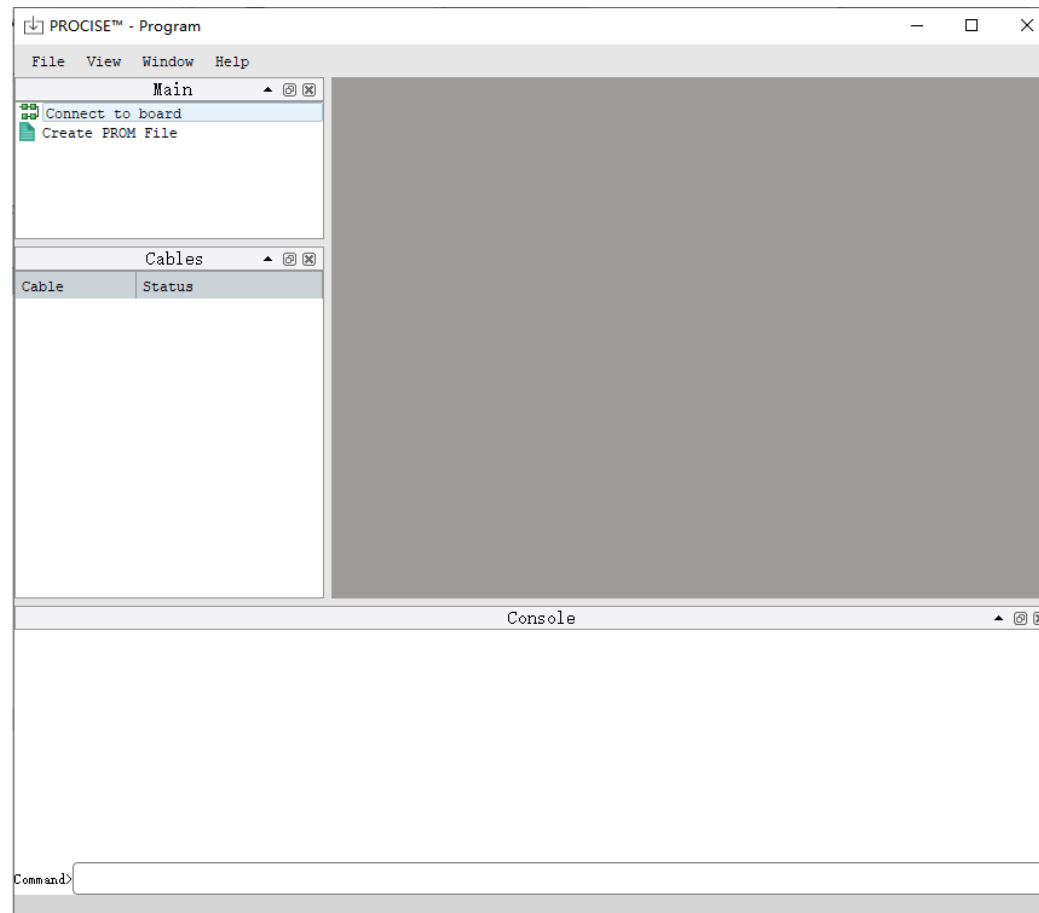
Utilization	Used	Available	Percent
Logic	37	32600	0.11%
Bonded IOUs	9	250	3.60%
GCPU/GCPU_CTRLs	1	32	3.13%

TCL Console

```
>>bitgen led.bit -g bpi_sync_mode:Disable spi_32bit_addr:No spi_buswidth:1 Encrypt:No  
preprocess elapsed_time 0.02 seconds, cpu_time 16.00 seconds  
assemble_bitstream elapsed_time 0.03 seconds, cpu_time -16.00 seconds  
Writing out bitstream file led.bit  
writing out bgn file led.bgn  
bitgen elapsed_time 0.93 seconds, cpu_time 1.02 seconds  
bitgen used memory 320MB, procise used peak memory 1353MB, current used memory 510MB  
>>write_phy_design C:/Users/wj/Desktop/fudan/led/rundir/.0b9ef7bc56c6af5e799347f69c1cc770.phy  
write_phy_design used memory 0MB, procise used peak memory 1353MB, current used memory 460MB
```

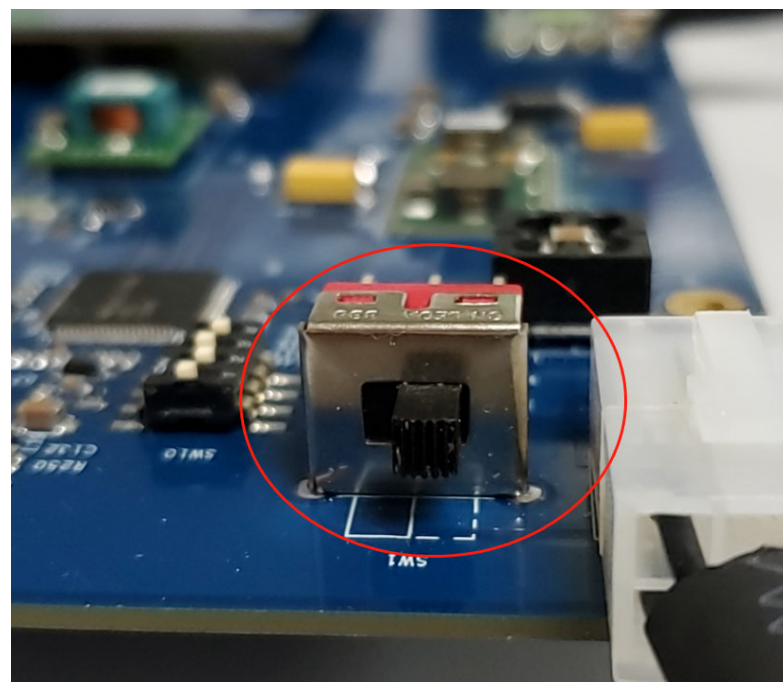
第五步：编程下载（Program Device/Tools>Configure Devices...）

1、打开编程下载窗口。

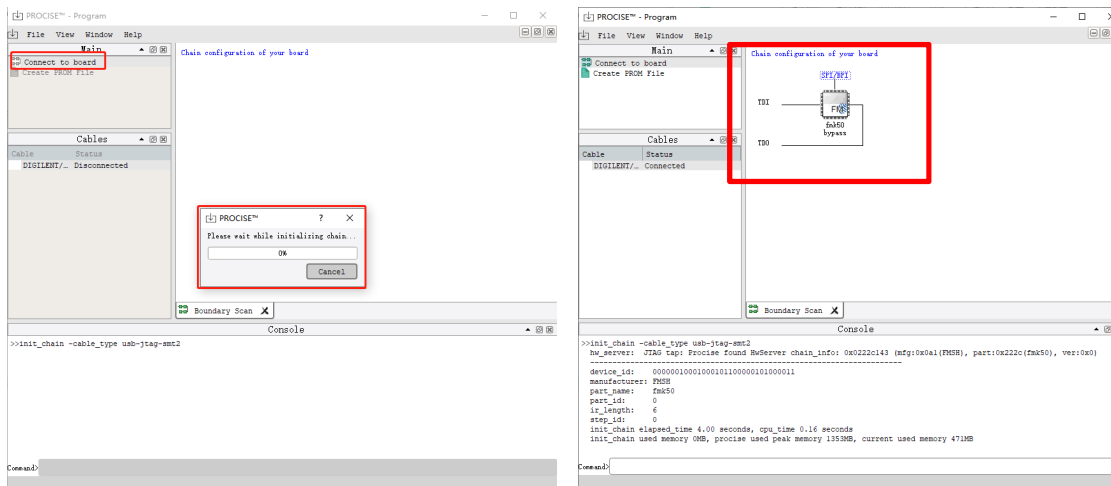


2、连接开发板与PC；

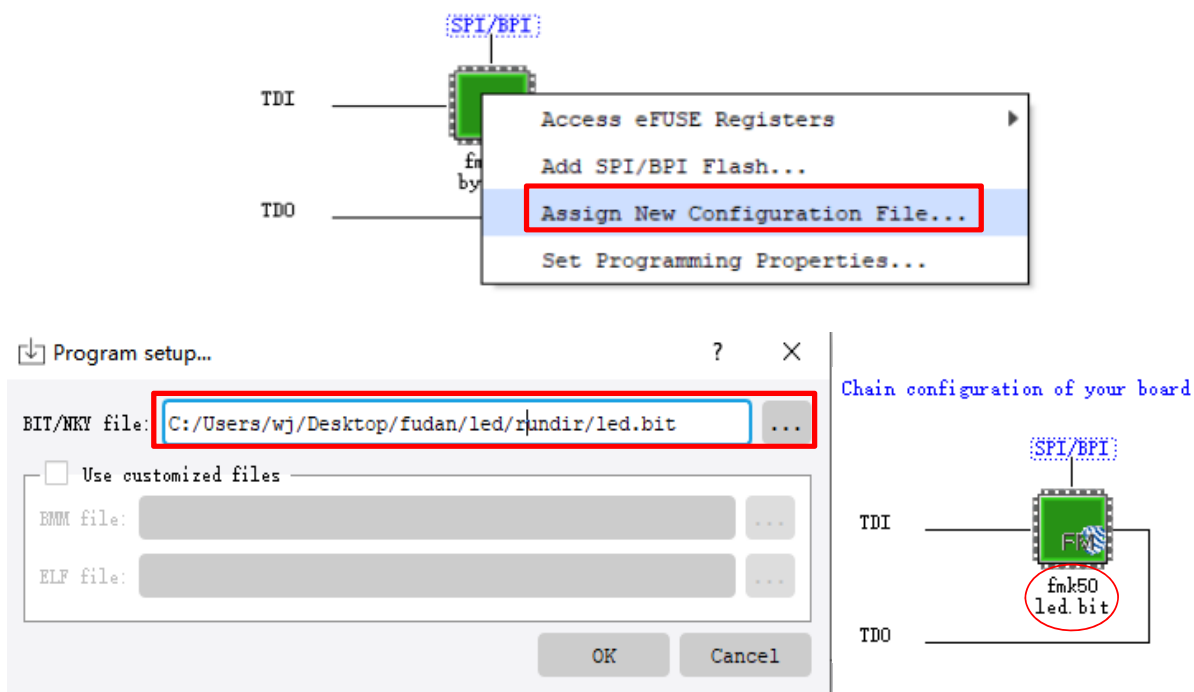
3、拨动开关“SW1”至如下状态，确保开发板上电。



4、双击“Connect to board”，连接成功后如右图所示。



5、选择将要配置的FPGA芯片，点击鼠标右键。选择“Assign New Configuration File...”，选择生成的led.bit文件，点击“OK”。



6、下载成功后，开发板led灯实现流水点亮的效果。

