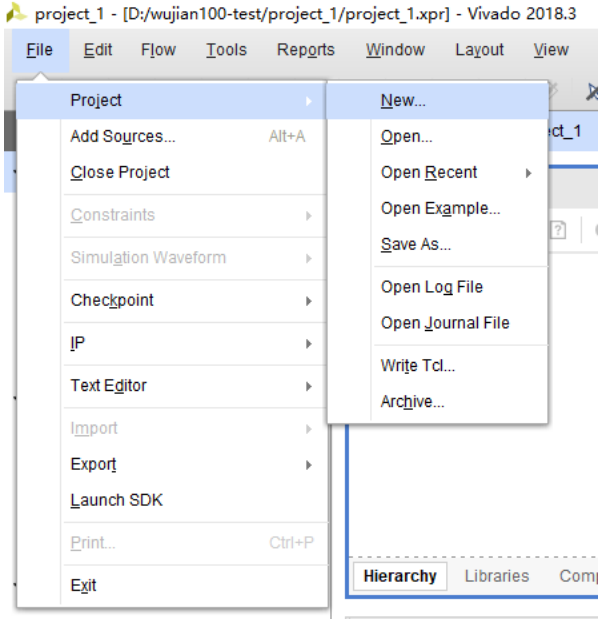
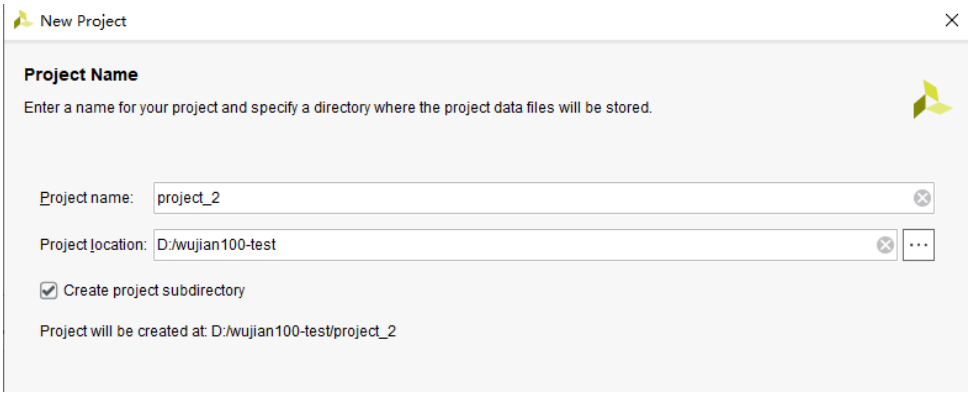
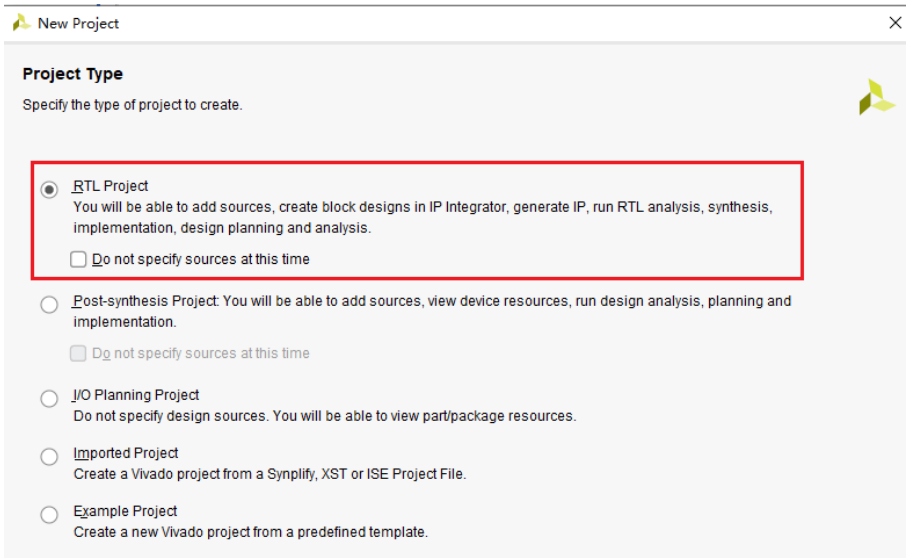
※Vivado 比特流生成部分

第一步——创建工程、加载文件

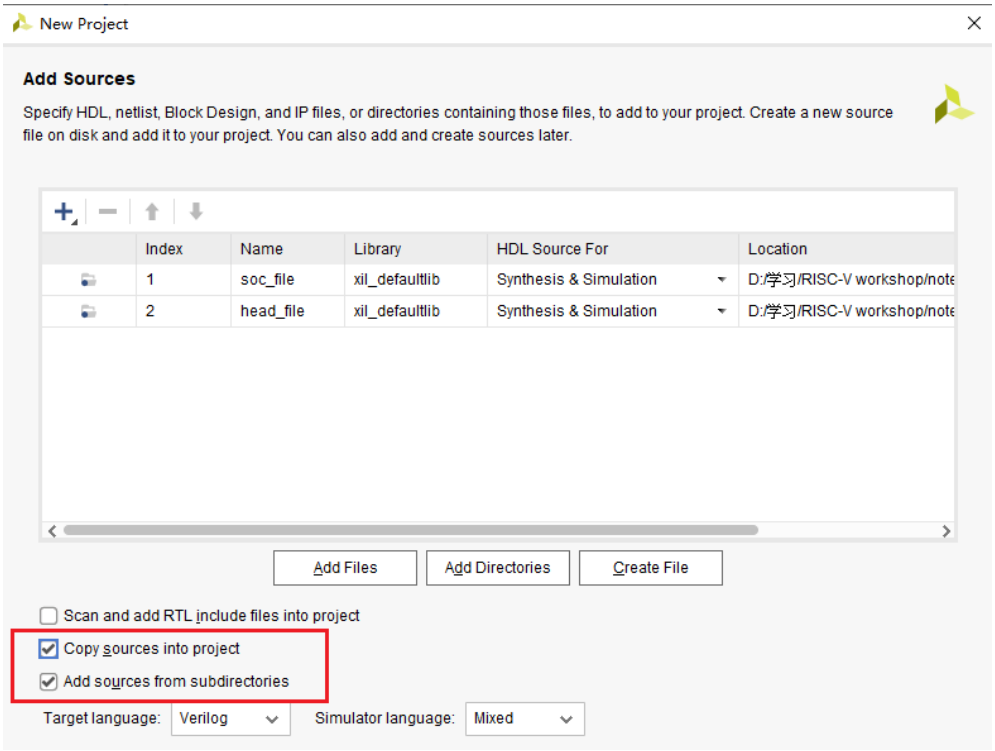
（1） 在 Vivado 中创建一个新的工程



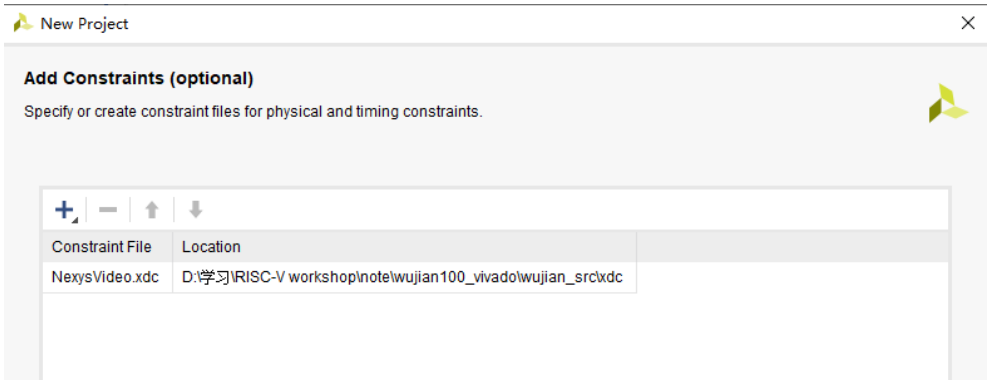




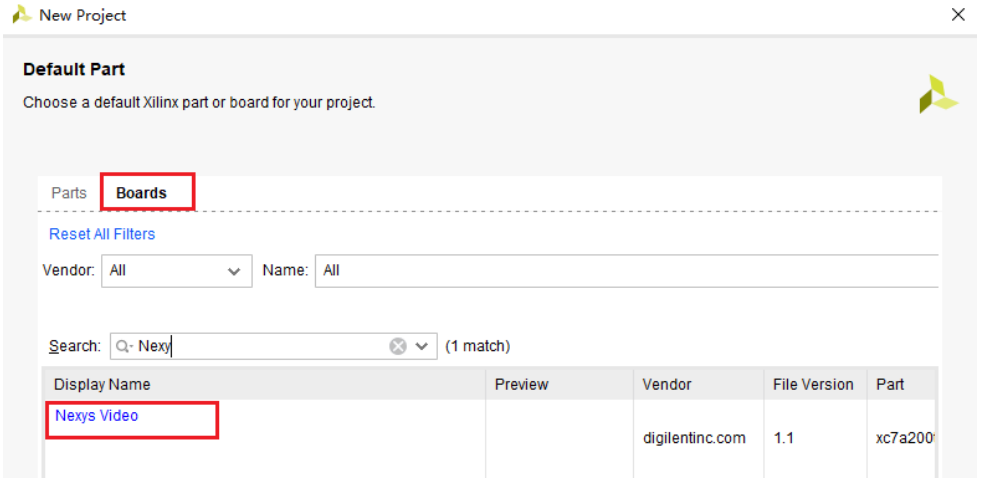
(2) 添加设计源文件和头文件



(3) 添加约束文件

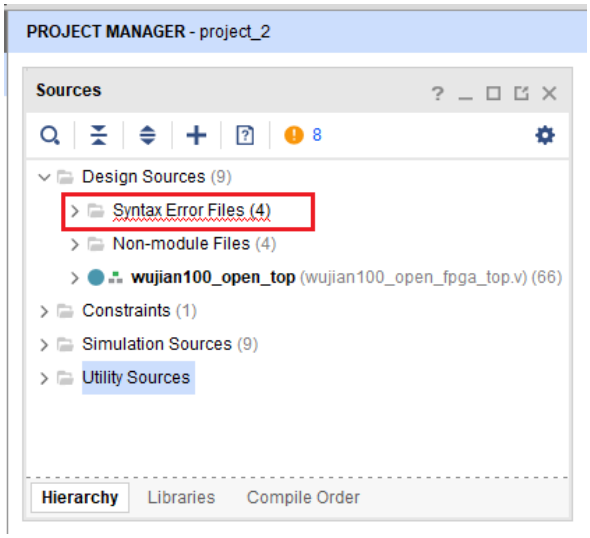


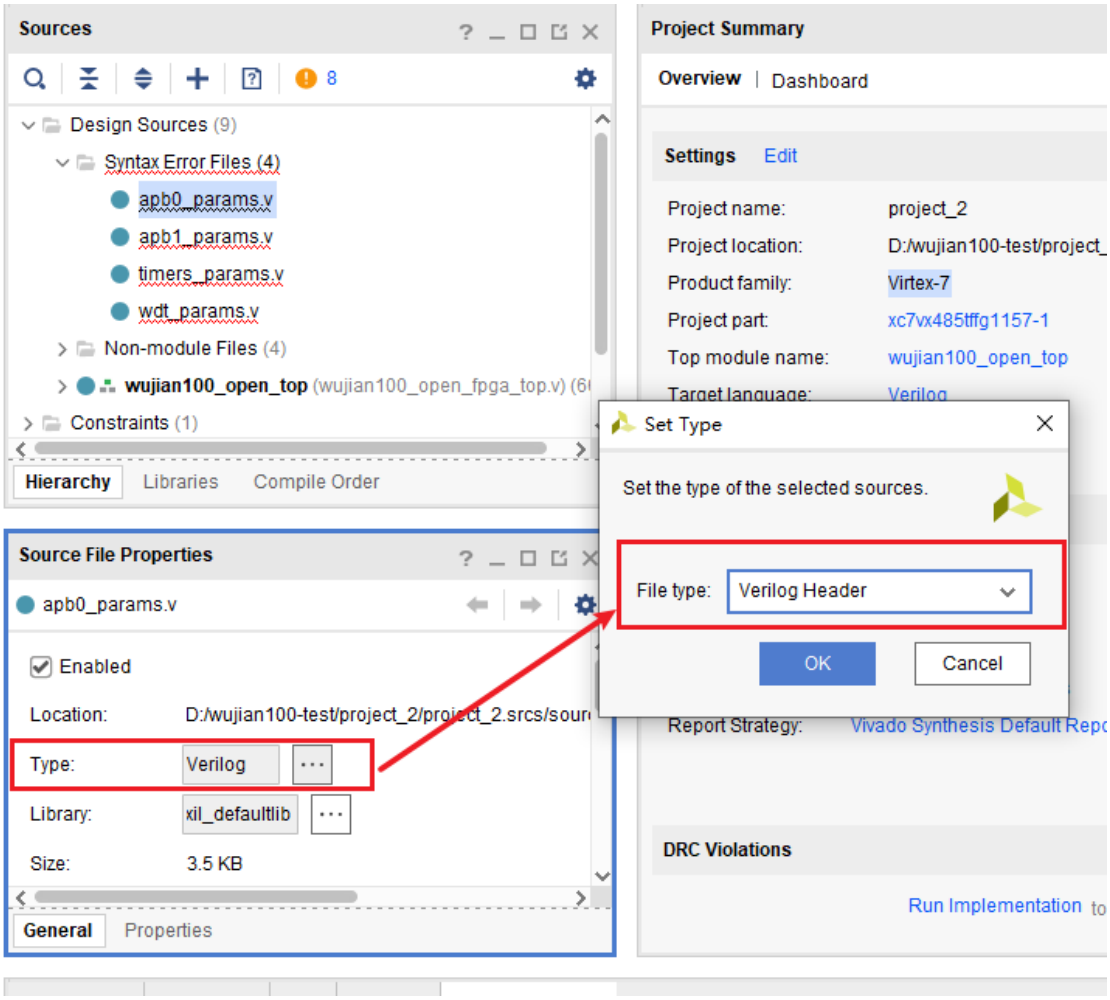
(4) 选择正确的板子

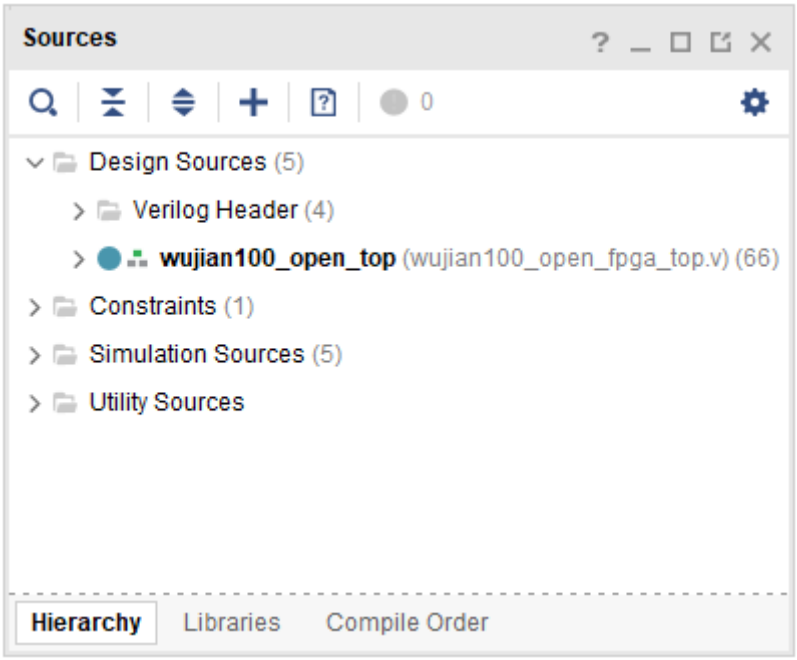


第二步——修正错误、加载 IP

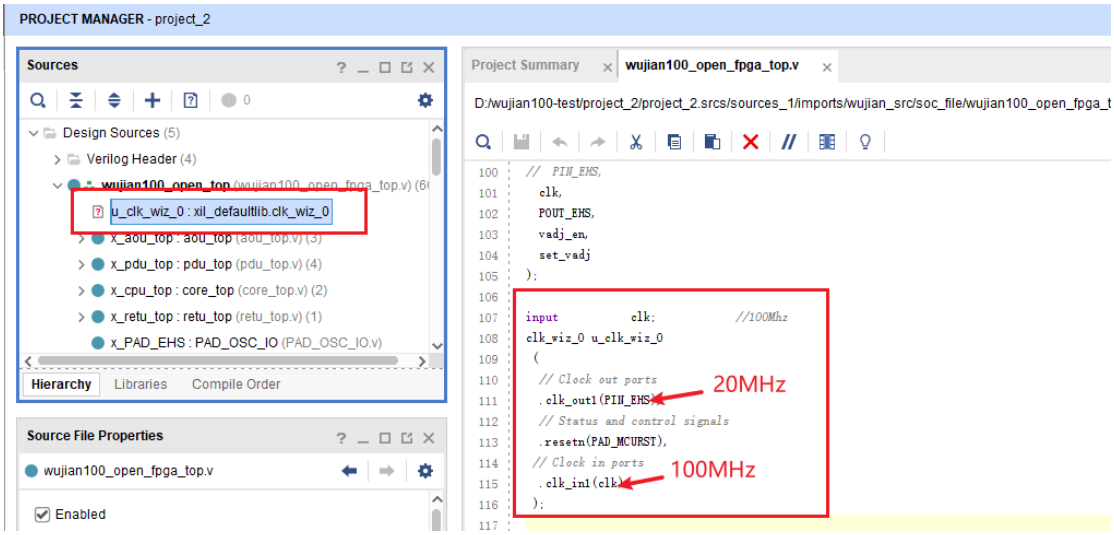
（1） 有错误，需修改类型

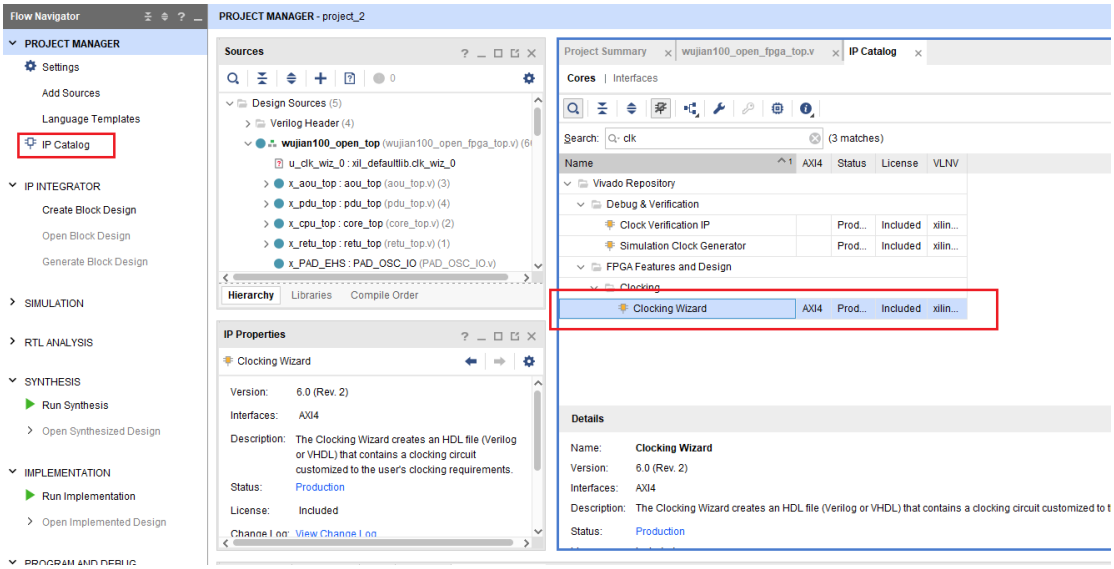


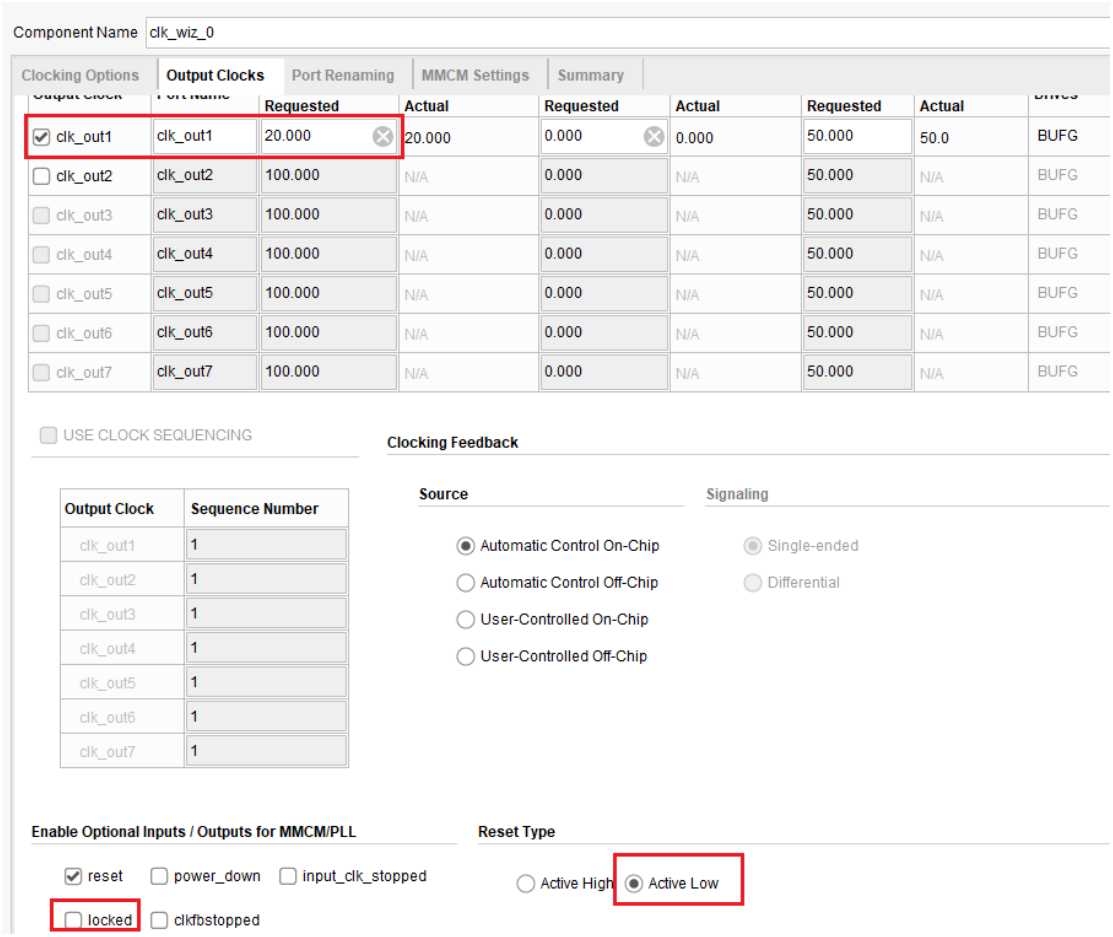



Sources
Design Sources (5)
Verilog Header (4)
wujian100_open_top (wujian100_open_fpga_top.v) (66)
Constraints (1)
Simulation Sources (5)
Utility Sources
Hierarchy
Libraries
Compile Order

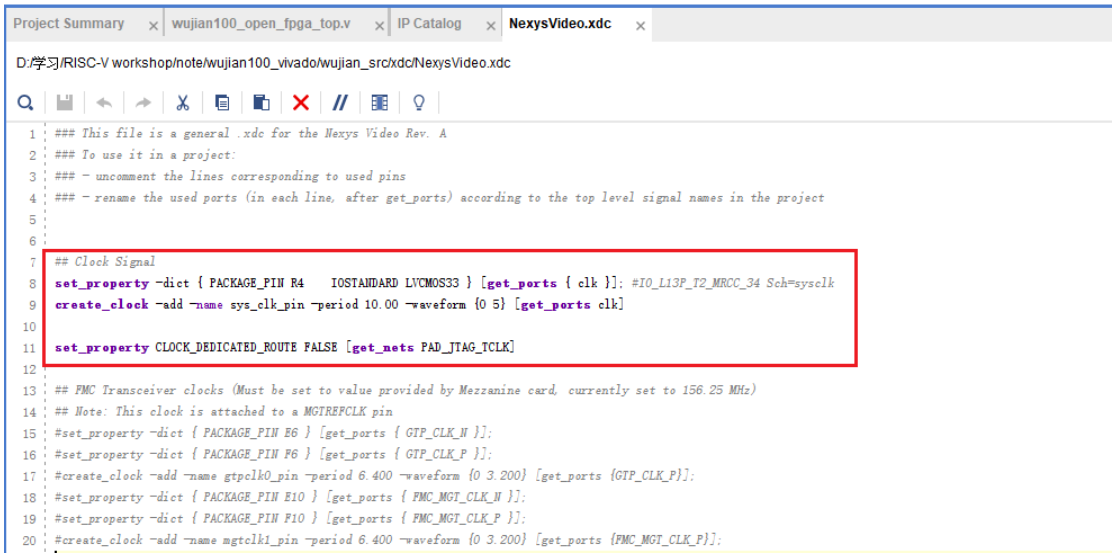
（2） 加时钟分频 IP

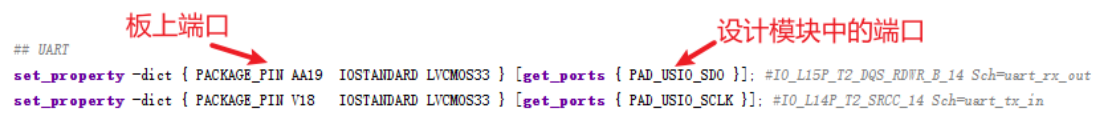

PROJECT MANAGER - project_2
Sources
Design Sources (5)
Verilog Header (4)
u_clk_wiz_0 : xil_defaultlib.clk_wiz_0
x_aou_top : aou_top (aou_top.v) (3)
x_pdu_top : pdu_top (pdu_top.v) (4)
x_cpu_top : core_top (core_top.v) (2)
x_retu_top : retu_top (retu_top.v) (1)
x_PAD_EHS : PAD_OSC_IO (PAD_OSC_IO.v)
Hierarchy
Libraries
Compile Order
Source File Properties
wujian100_open_fpga_top.v
Enabled
Project Summary
wujian100_open_fpga_top.v
D:/wujian100-test/project_2/project_2.srcs/sources_1/imports/wujian_src/soc_file/wujian100_open_fpga_t
// PIN_EHS,
clk,
POUT_EHS,
vadj_en,
set_vadj
);
input clk; //100Mhz
clk_wiz_0 u_clk_wiz_0
(
// Clock out ports
.clk_out1(PIN_EHS),
// Status and control signals
.resetn(PAD_MCURST),
// Clock in ports
.clk_in1(clk)
);
20MHz
100MHz


Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
RTL ANALYSIS
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROJECT MANAGER - project_2
Sources
IP Properties
Clocking Wizard
Version: 6.0 (Rev. 2)
Interfaces: AXI4
Description: The Clocking Wizard creates an HDL file (Verilog or VHDL) that contains a clocking circuit customized to the user's clocking requirements.
Status: Production
License: Included
Project Summary
wujian100_open_fpga_top.v
IP Catalog
Cores | Interfaces
Search: clk
(3 matches)
Name
AXI4
Status
License
VLNV
Vivado Repository
Debug & Verification
Clock Verification IP
Simulation Clock Generator
FPGA Features and Design
Clocking
Clocking Wizard
Details
Name: Clocking Wizard
Version: 6.0 (Rev. 2)
Interfaces: AXI4
Description: The Clocking Wizard creates an HDL file (Verilog or VHDL) that contains a clocking circuit customized to t
Status: Production


Component Name clk_wiz_0
Clocking Options
Output Clocks
Port Renaming
MMCM Settings
Summary
Requested
Actual
clk_out1 clk_out1 20.000 20.000 0.000 0.000 50.000 50.0 BUFG
clk_out2 clk_out2 100.000 N/A 0.000 N/A 50.000 N/A BUFG
clk_out3 clk_out3 100.000 N/A 0.000 N/A 50.000 N/A BUFG
clk_out4 clk_out4 100.000 N/A 0.000 N/A 50.000 N/A BUFG
clk_out5 clk_out5 100.000 N/A 0.000 N/A 50.000 N/A BUFG
clk_out6 clk_out6 100.000 N/A 0.000 N/A 50.000 N/A BUFG
clk_out7 clk_out7 100.000 N/A 0.000 N/A 50.000 N/A BUFG
USE CLOCK SEQUENCING
Output Clock
Sequence Number
clk_out1 1
clk_out2 1
clk_out3 1
clk_out4 1
clk_out5 1
clk_out6 1
clk_out7 1
Clocking Feedback
Source
Signaling
Automatic Control On-Chip
Automatic Control Off-Chip
User-Controlled On-Chip
User-Controlled Off-Chip
Single-ended
Differential
Enable Optional Inputs / Outputs for MMCM/PLL
reset
power_down
input_clk_stopped
locked
clkfbstopped
Reset Type
Active High
Active Low

(3) 确认约束内容

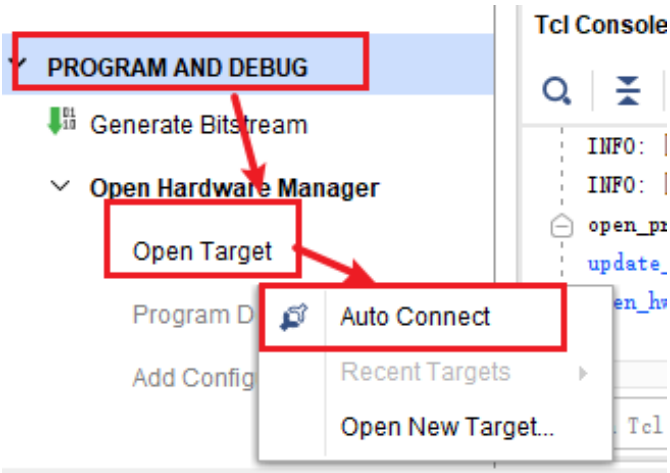

Project Summary
wujian100_open_fpga_top.v
IP Catalog
NexysVideo.xdc
D:/学习/RISC-V workshop/note/wujian100_vivado/wujian_src/xdc/NexysVideo.xdc
This file is a general .xdc for the Nexys Video Rev. A
To use it in a project:
- uncomment the lines corresponding to used pins
- rename the used ports (in each line, after get_ports) according to the top level signal names in the project
Clock Signal
set_property -dict { PACKAGE_PIN R4 IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L13P_T2_MRCC_34 Sch=sysclk
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets PAD_JTAG_TCLK]
FMC Transceiver clocks (Must be set to value provided by Mezzanine card, currently set to 156.25 MHz)
Note: This clock is attached to a MGTREFCLK pin
#set_property -dict { PACKAGE_PIN E6 } [get_ports { GTP_CLK_N }];
#set_property -dict { PACKAGE_PIN F6 } [get_ports { GTP_CLK_P }];
#create_clock -add -name gtpclk0_pin -period 6.400 -waveform {0 3.200} [get_ports {GTP_CLK_P}];
#set_property -dict { PACKAGE_PIN E10 } [get_ports { FMC_MGT_CLK_N }];
#set_property -dict { PACKAGE_PIN F10 } [get_ports { FMC_MGT_CLK_P }];
#create_clock -add -name mgtclk1_pin -period 6.400 -waveform {0 3.200} [get_ports {FMC_MGT_CLK_P}];


板上端口
设计模块中的端口
UART
set_property -dict { PACKAGE_PIN AA19 IOSTANDARD LVCMOS33 } [get_ports { PAD_USIO_SDO }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=uart_rx_out
set_property -dict { PACKAGE_PIN V18 IOSTANDARD LVCMOS33 } [get_ports { PAD_USIO_SCLK }]; #IO_L14P_T2_SRCC_14 Sch=uart_tx_in

```
## Buttons
set_property -dict { PACKAGE_PIN B22 IOSTANDARD LVCMOS33 } [get_ports { PAD_GPIO_8 }]; #IO_L20N_T3_16 Sch=btnc
set_property -dict { PACKAGE_PIN D22 IOSTANDARD LVCMOS33 } [get_ports { PAD_GPIO_9 }]; #IO_L22N_T3_16 Sch=btnd
set_property -dict { PACKAGE_PIN C22 IOSTANDARD LVCMOS33 } [get_ports { PAD_GPIO_10 }]; #IO_L20P_T3_16 Sch=btnl
set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { PAD_GPIO_11 }]; #IO_L6P_T0_16 Sch=btnr
set_property -dict { PACKAGE_PIN F15 IOSTANDARD LVCMOS33 } [get_ports { PAD_GPIO_12 }]; #IO_0_16 Sch=btnu
set_property -dict { PACKAGE_PIN G4  IOSTANDARD LVCMOS15 } [get_ports { PAD_MCURST }]; #IO_L12N_T1_MRCC_35 Sch=cpu_resetn
```

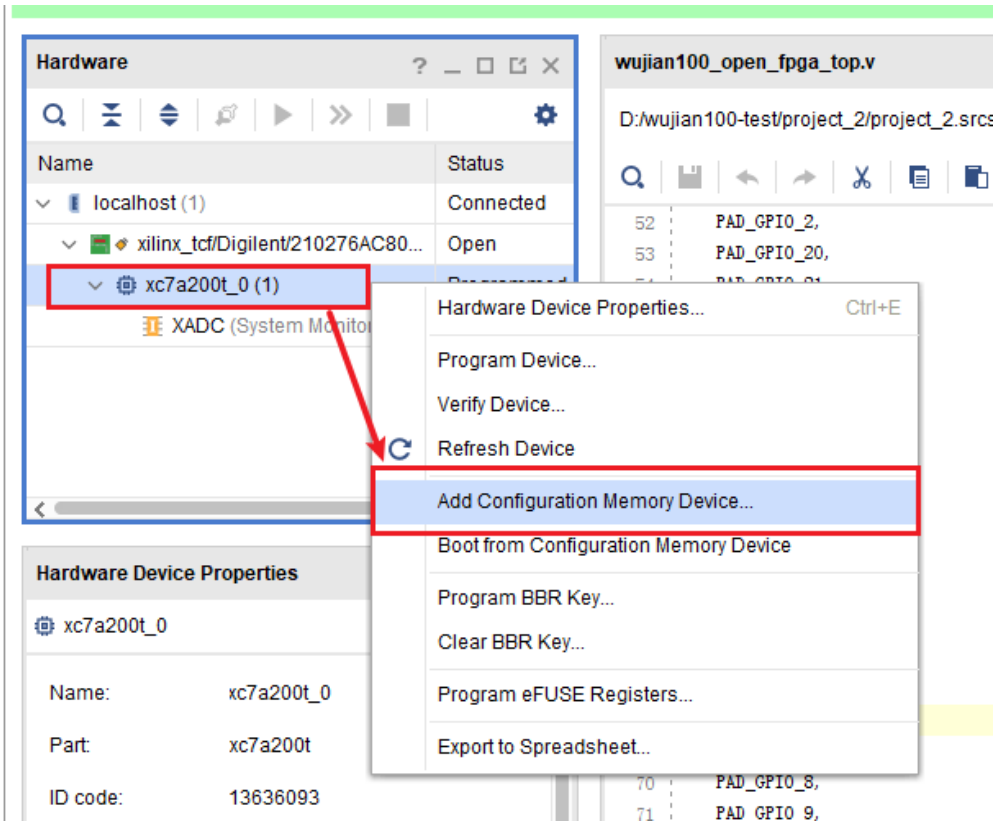
第三步——固化程序

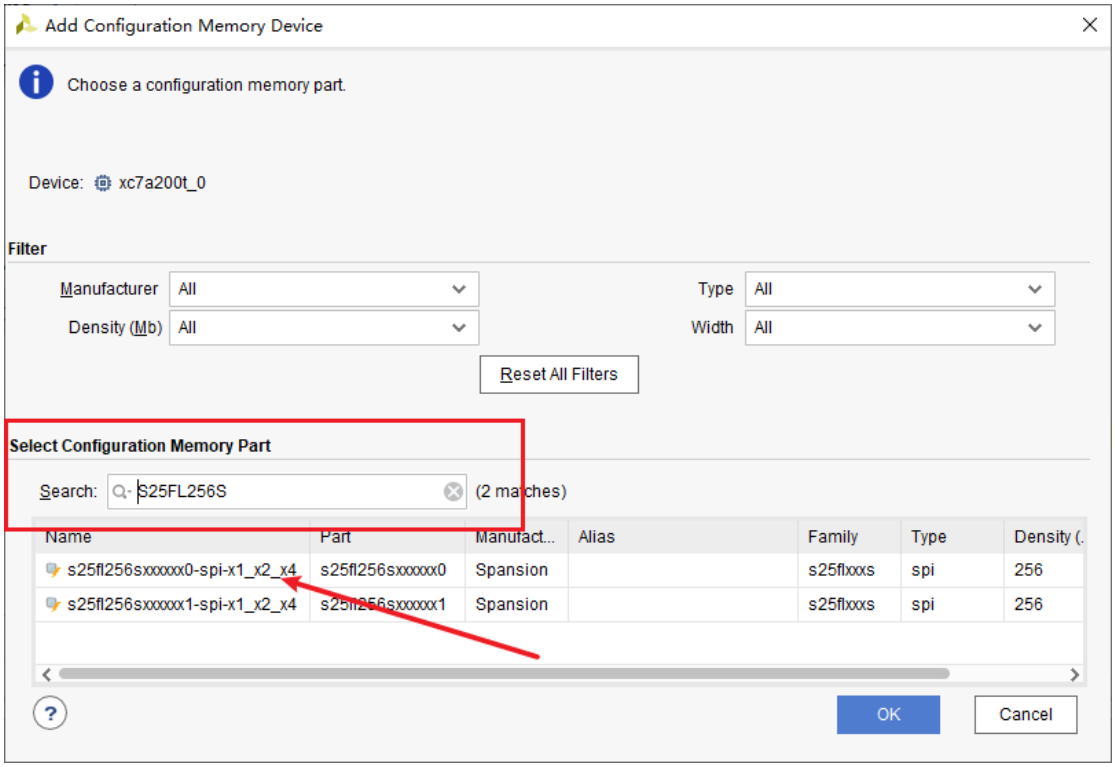
（1） 确认板子已连接



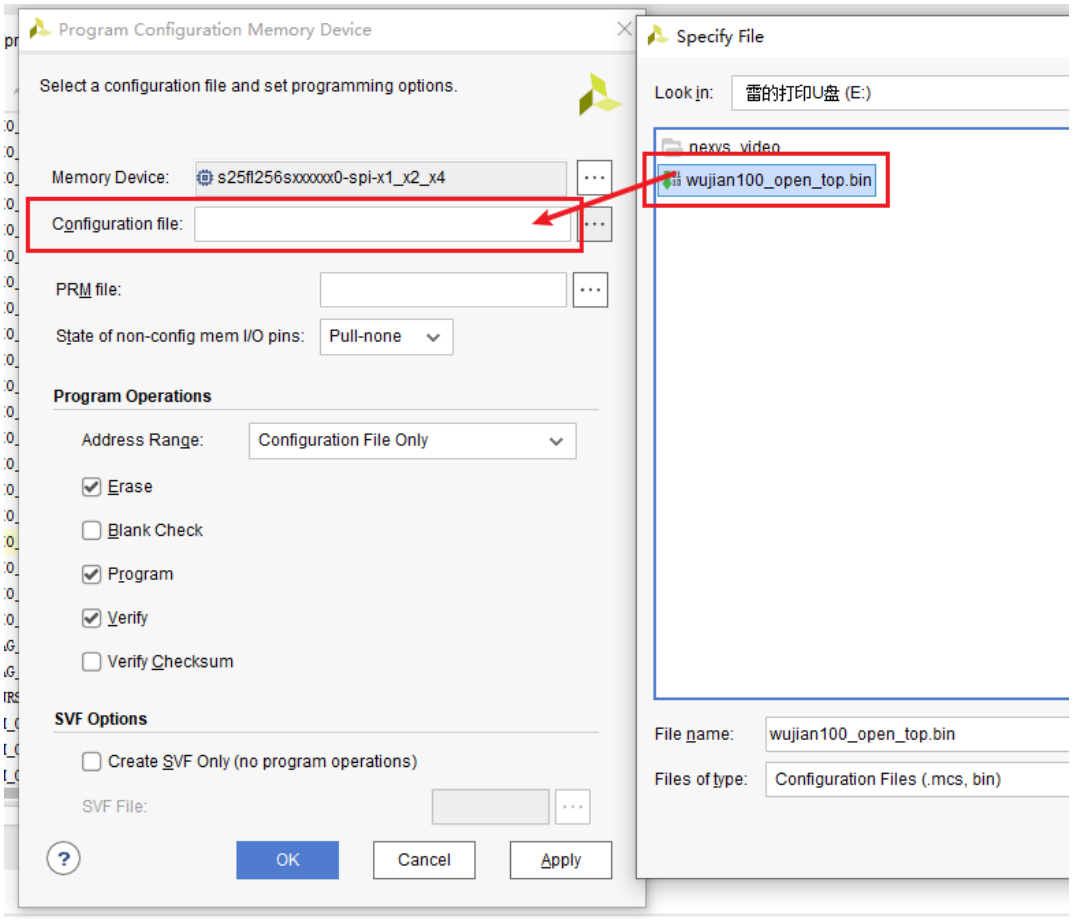
打开一个最近打开的目标

（2） 添加 FLASH 设备



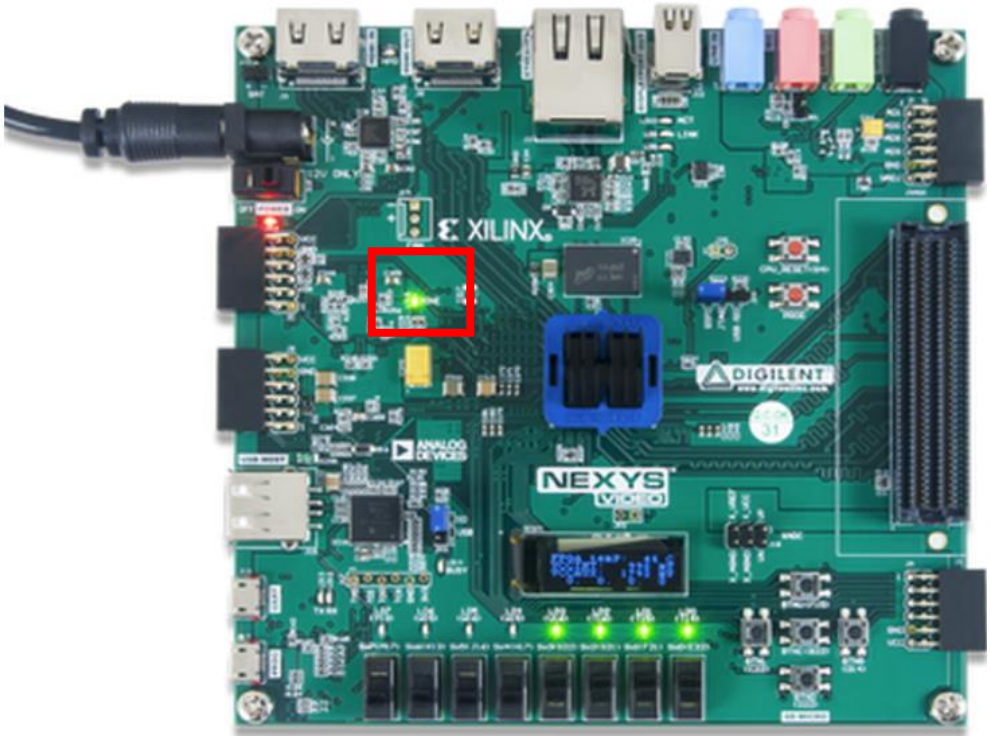


（3） 添加 bin 文件



点击 OK，需等待许久……

（4） 加载成功现象



先按下 PROG 按键，等待加载一会后 DONE 亮起。

※仿真下载部分

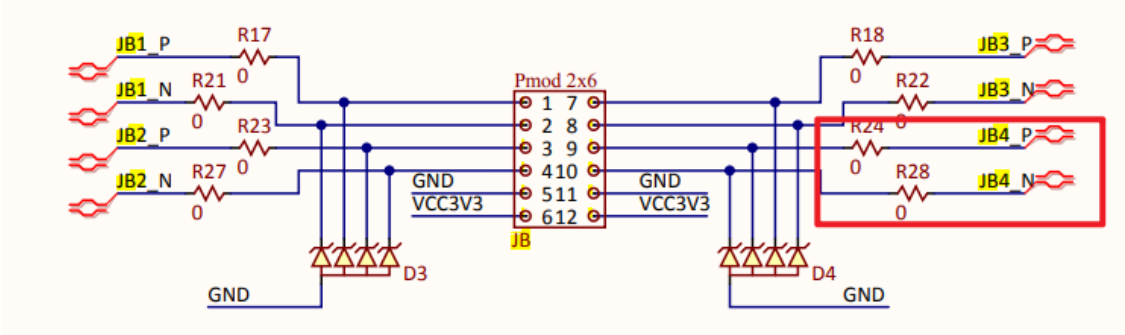
第一步——连接板子

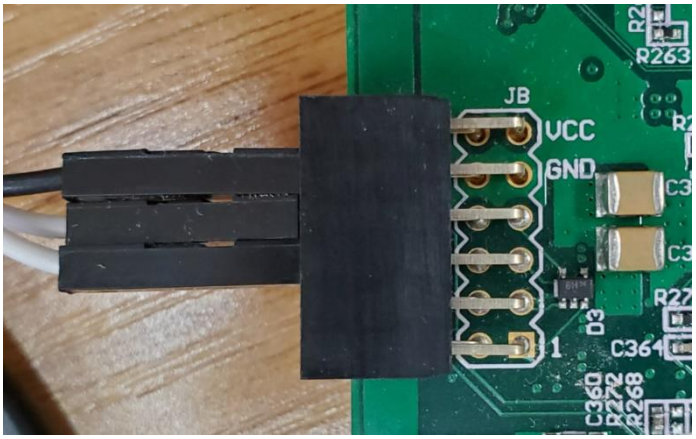
（1） 查看 CKLINK 与板子的连接方式

NexysVideo_Master.xdc - 记事本

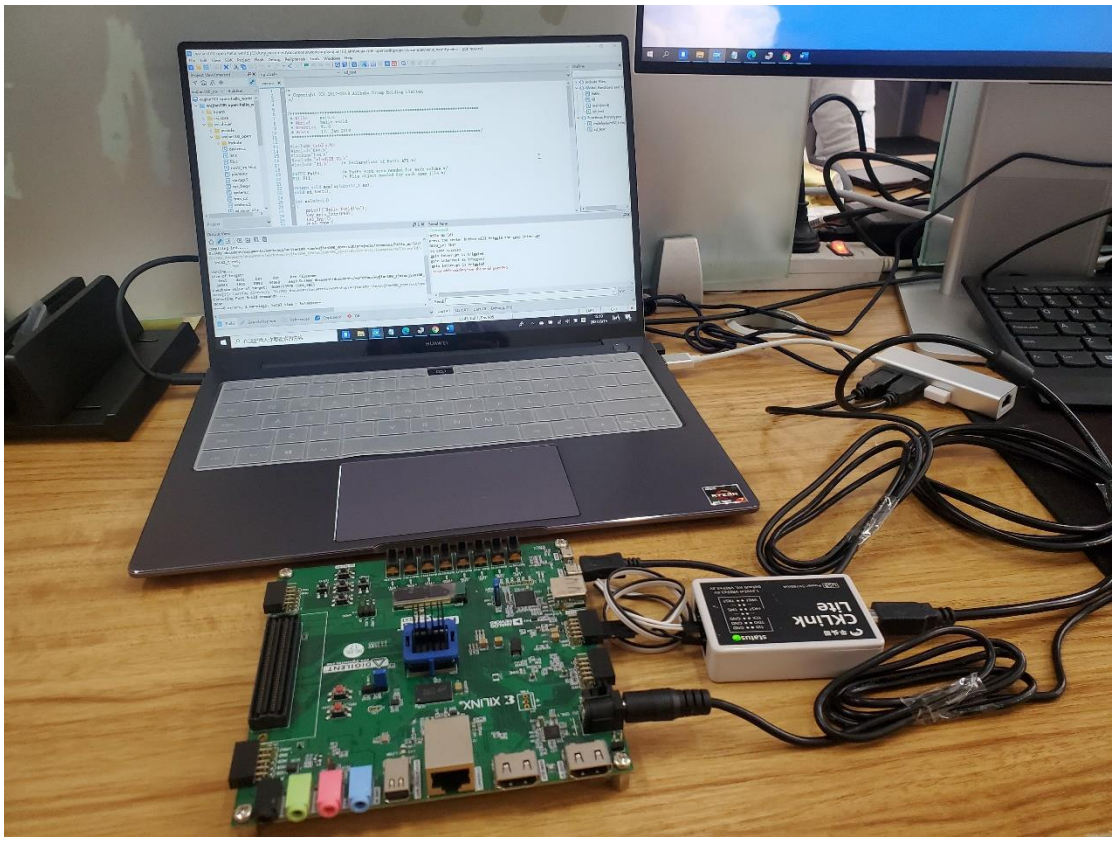
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
## Pmod header JB
set_property -dict { PACKAGE_PIN V9    IOSTANDARD LVCMOS33 } [get_ports { PAD_PWM_CH3 }]; #IO_L21P_T3_DQS_34 Sch=jb_p[1]
set_property -dict { PACKAGE_PIN V8    IOSTANDARD LVCMOS33 } [get_ports { PAD_PWM_CH4 }]; #IO_L21N_T3_DQS_34 Sch=jb_n[1]
set_property -dict { PACKAGE_PIN V7    IOSTANDARD LVCMOS33 } [get_ports { PAD_PWM_CH5 }]; #IO_L19P_T3_34 Sch=jb_p[2]
set_property -dict { PACKAGE_PIN W7    IOSTANDARD LVCMOS33 } [get_ports { PAD_PWM_CH6 }]; #IO_L19N_T3_VREF_34 Sch=jb_n[2]
set_property -dict { PACKAGE_PIN W9    IOSTANDARD LVCMOS33 } [get_ports { PAD_PWM_CH7 }]; #IO_L24P_T3_34 Sch=jb_p[3]
set_property -dict { PACKAGE_PIN Y9    IOSTANDARD LVCMOS33 } [get_ports { PAD_PWM_CH8 }]; #IO_L24N_T3_34 Sch=jb_n[3]
set_property -dict { PACKAGE_PIN Y8    IOSTANDARD LVCMOS33 } [get_ports { PAD_JTAG_TMS }]; #IO_L23P_T3_34 Sch=jb_p[4]
set_property -dict { PACKAGE_PIN Y7    IOSTANDARD LVCMOS33 } [get_ports { PAD_JTAG_TCLK }]; #IO_L23N_T3_34 Sch=jb_n[4]
```



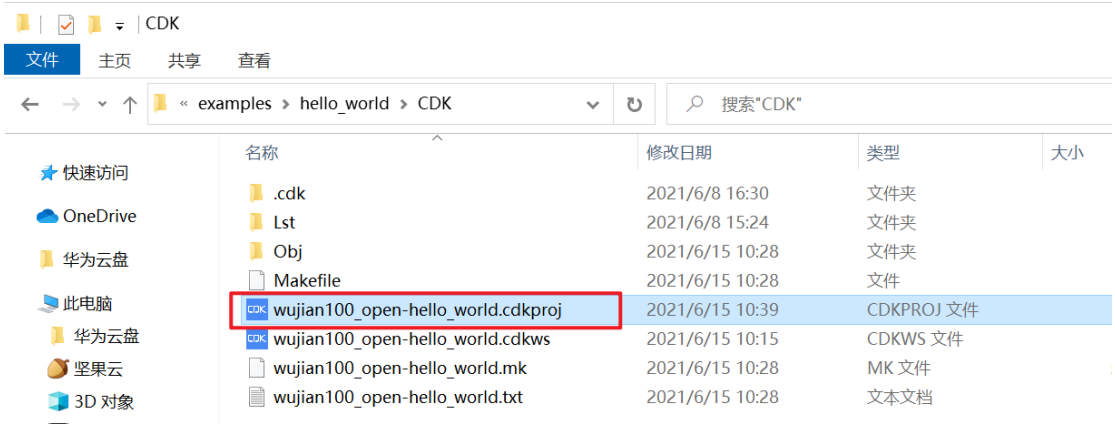


（2） 总体连接

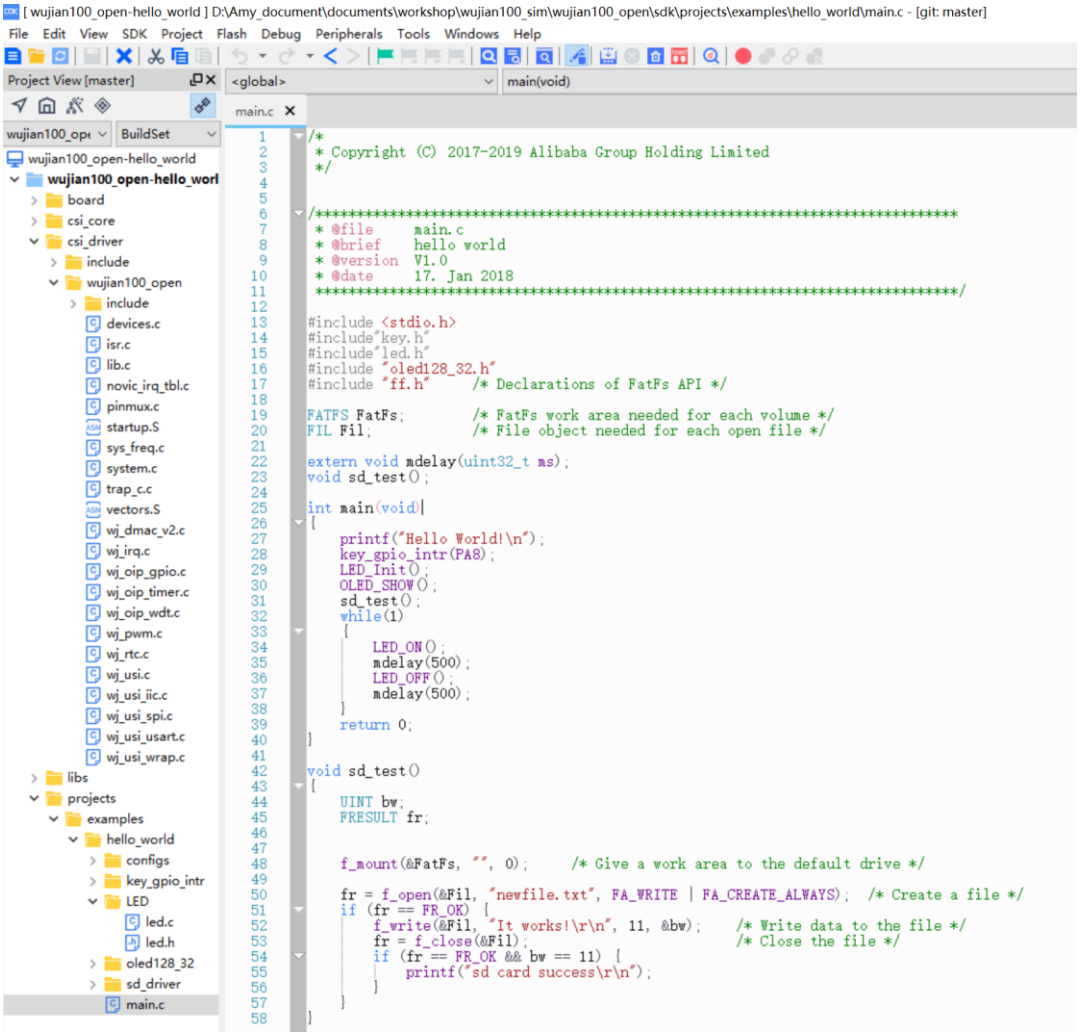


第二步——打开 project、下载程序

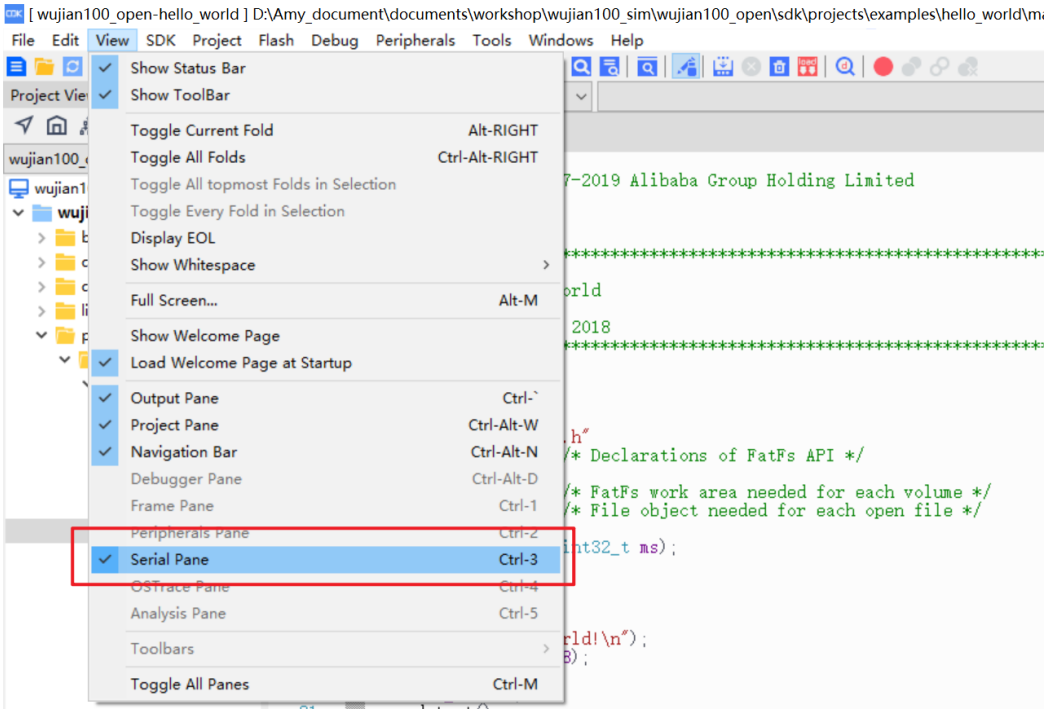
（1） 打开 project

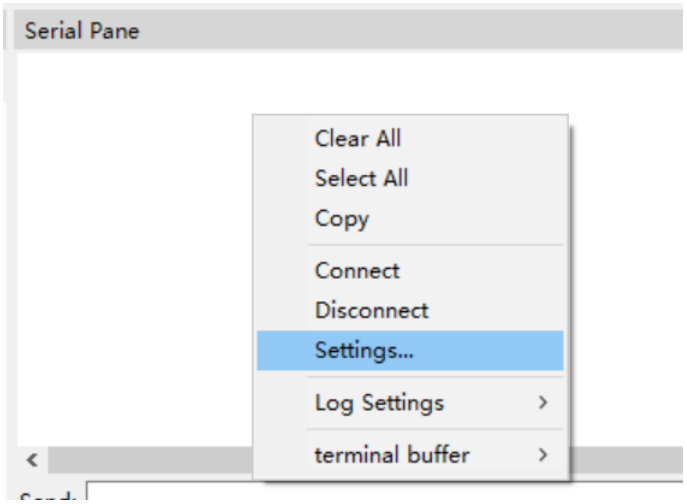


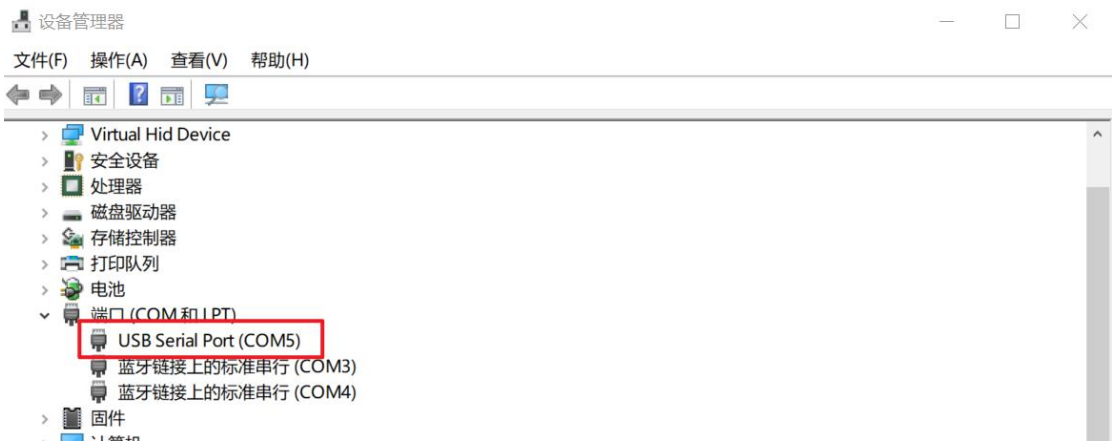
(2) 在 CDK 中打开 main.c

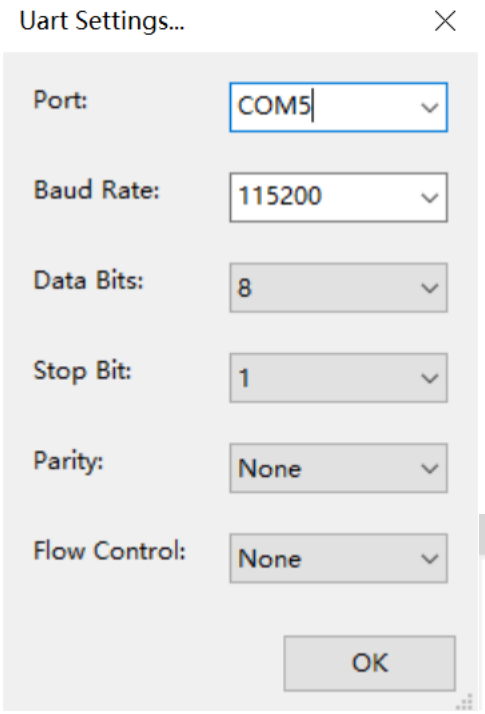


(3) 打开串口界面，设置串口



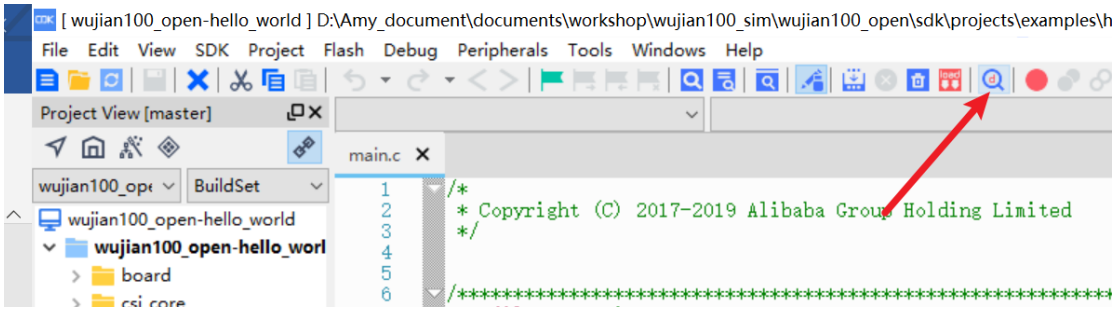

Serial Pane
Clear All
Select All
Copy
Connect
Disconnect
Settings...
Log Settings
terminal buffer


设备管理器
文件(F)
操作(A)
查看(V)
帮助(H)
Virtual Hid Device
安全设备
处理器
磁盘驱动器
存储控制器
打印队列
电池
端口 (COM 和 LPT)
USB Serial Port (COM5)
蓝牙链接上的标准串行 (COM3)
蓝牙链接上的标准串行 (COM4)
固件


Uart Settings...
Port:
COM5
Baud Rate:
115200
Data Bits:
8
Stop Bit:
1
Parity:
None
Flow Control:
None
OK


Serial Pane
Connected.

（4） 下载程序到板子上



点击一次开始调试，再点击一次终止调试，此时程序就下进板子里了

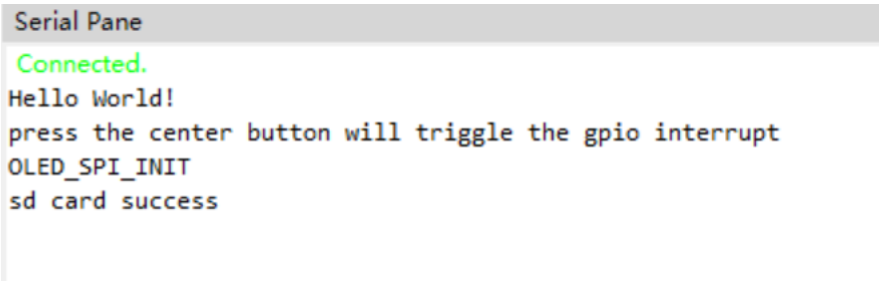
第三步——板上运行

（1） 按下板子上的 CPU_RESET 键，开始运行程序。

程序执行完成以下操作:

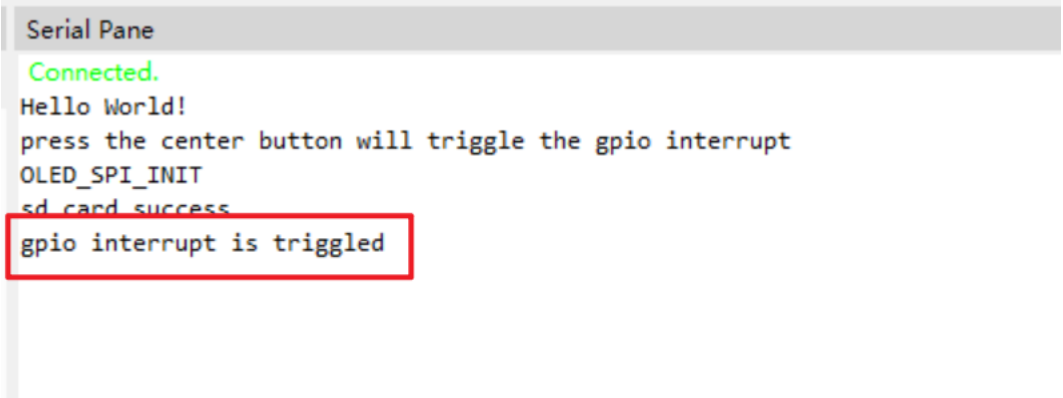
- 串口输出调试信息
- LED1-LED3 每秒闪烁一次
- OLED 显示字符串
- 按下 center button 之后，会触发 GPIO 中断输出信息。
- MCU 在 SD 卡中新建文件并写入内容

（2） 首先观察串口界面的调试信息。



（3） 观察 LED 闪烁情况和 OLED 显示内容

（4） 按 center button 测试中断功能



（5） 板子断电拔出 SD 卡，在电脑上查看是否写入内容。