

Vivado软件使用

- 设计输入

采用Verilog HDL描述功能

- 功能仿真

验证设计文件的逻辑功能是否正确

- 用户约束

对I/O引脚位置、时序等附加约束

- 综合

将HDL转换为为门级（逻辑单元）表示的过程

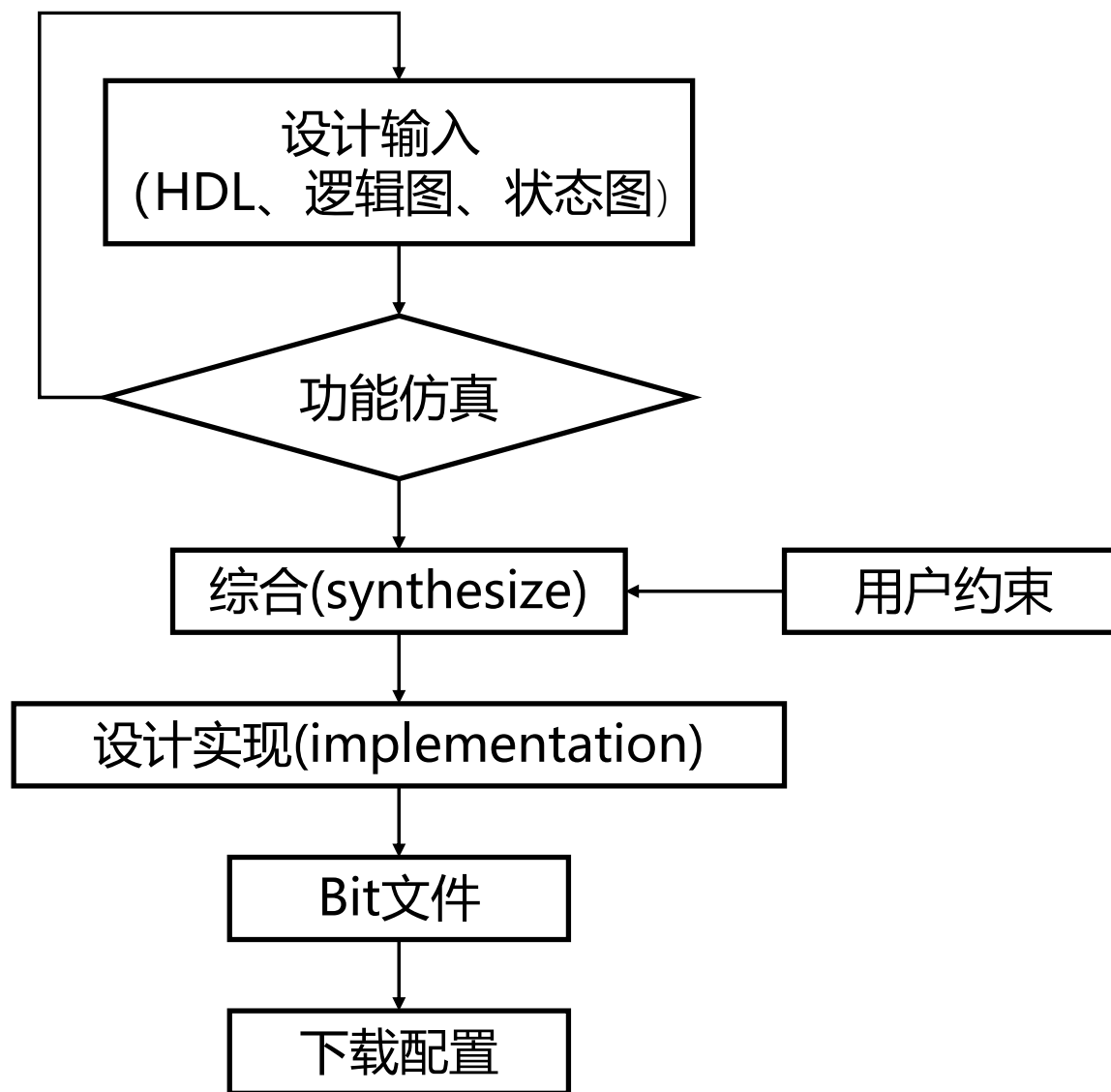
实验步骤

- 设计实现

包括逻辑优化、映射、布局和布线等过程

- 下载配置

将设计后的设计文件转化成bitstream文件，进行设备控制和通信，控制程序烧写到FPGA芯片中



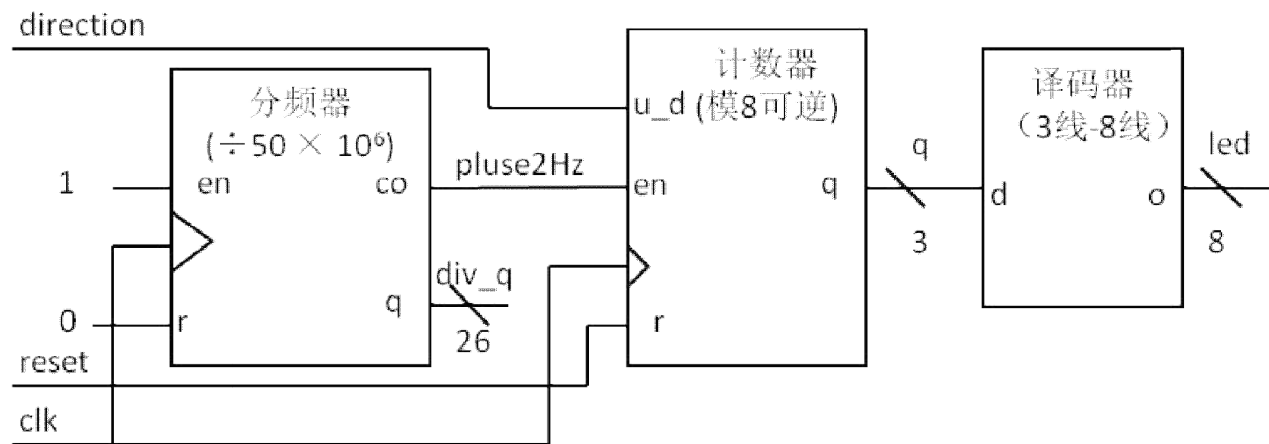
设计输入

- 流水灯电路

8个发光二极管间隔0.5s轮流发光

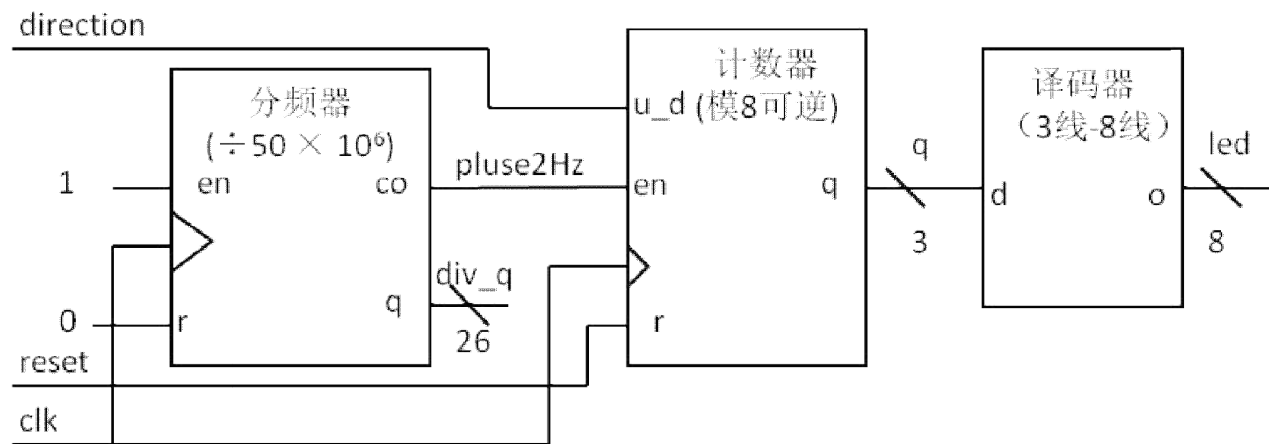
设置一个direction开关控制流水灯点亮方向。direction=0时，向右轮流点亮；direction=1时，向左轮流点亮。

设置一个reset按钮，复位时，最右边灯亮。



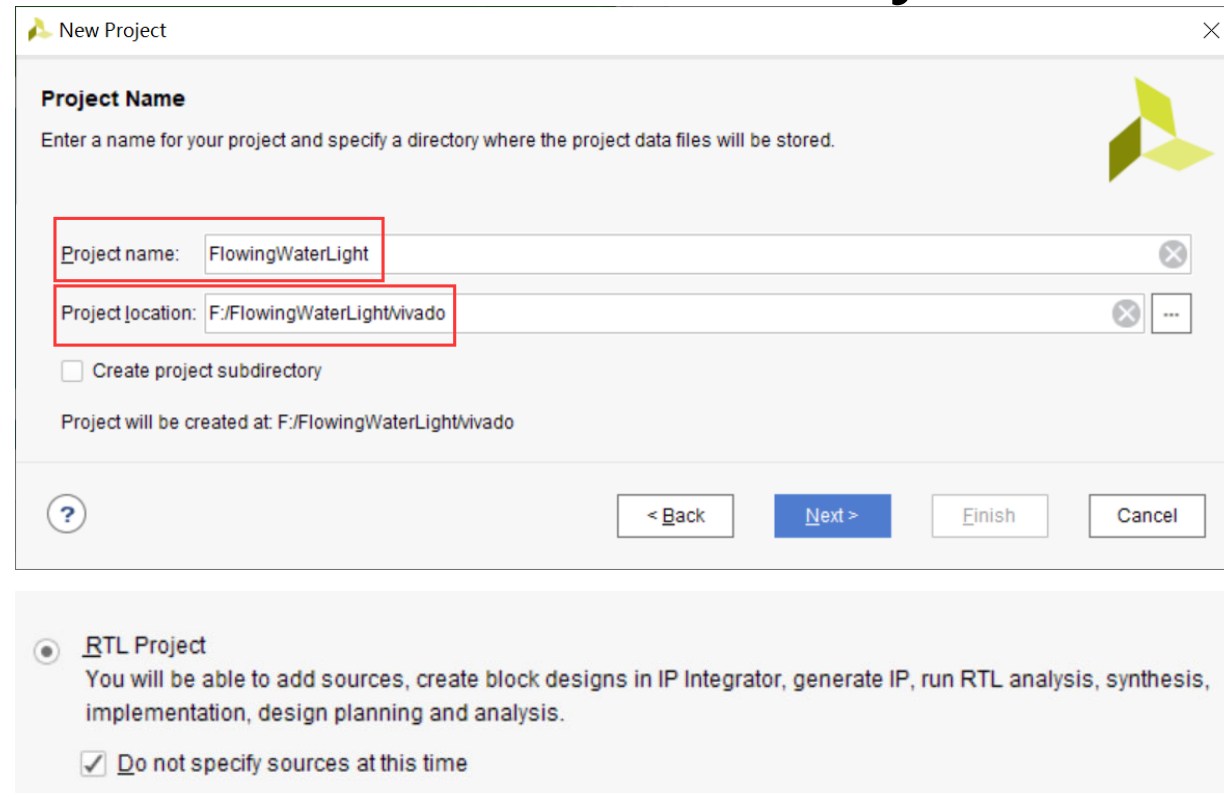
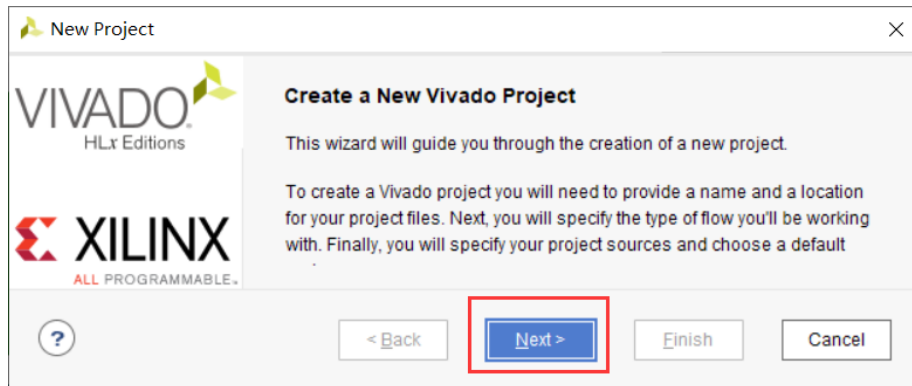
设计输入

- Verilog HDL代码，存放在F:\FlowingWaterLight\src下
- 分频器模块counter_n.v
计数器模块counter_up_down.v
译码器模块decode.v
顶层模块FlowingWaterLight.v
仿真测试代码FlowingWaterLight_tb.v



设计输入

- 建立Vivado工程文件，存放在F:\FlowingWaterLight\vivado下，工程名称和路径都不能出现中文和空格，下一步选RTL Project



设计输入

- 若目标器件为EGO1开发板，对应Artix-7 xc7a35tcsg324-1

New Project

Default Part
Choose a default Xilinx part or board for your project. This can be changed later.

Select: ☒ Parts ☐ Boards

Filter

Product category: All Speed grade: -1

Family: Artix-7 Temp grade: All Remaining

Package: csg324

Reset All Filters

Search: Q

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Transceivers	GTPE2 Transceiv
xc7a15tcsg324-1	324	210	10400	20800	25	0	45	0	0
xc7a35tcsg324-1	324	210	20800	41600	50	0	90	0	0
xc7a50tcsg324-1	324	210	32600	65200	75	0	120	0	0
xc7a75tcsg324-1	324	210	47200	94400	105	0	180	0	0
xc7a100tcsg324-1	324	210	63400	126800	135	0	240	0	0

< >

? < Back Next > Finish Cancel

New Project Summary

i A new RTL project named 'FlowingWaterLight' will be created.

i The default part and product family for the new project:

Default Part: xc7a35tcsg324-1

Product: Artix-7

Family: Artix-7

Package: csg324

Speed Grade: -1

设计输入

- 若目标器件为Basys3开发板，对应Artix-7 xc7a35tcpg236-1

New Project

Default Part
Choose a default Xilinx part or board for your project. This can be changed later.

Select: ☒ Parts ☐ Boards

Filter

Product category: All Speed grade: -1

Family: Artix-7 Temp grade: All Remaining

Package: cpg236

Reset All Filters

Search: Q

Part	I/O Pin	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Transceivers	GTPE2 Transceivers	GTXE2 Transceivers
xc7a15tcpg236-1	236	106	10400	20800	25	0	45	2	2	0
xc7a35tcpg236-1	236	106	20800	41600	50	0	90	2	2	0
xc7a50tcpg236-1	236	106	32600	65200	75	0	120	2	2	0

< Back Next > Finish Cancel

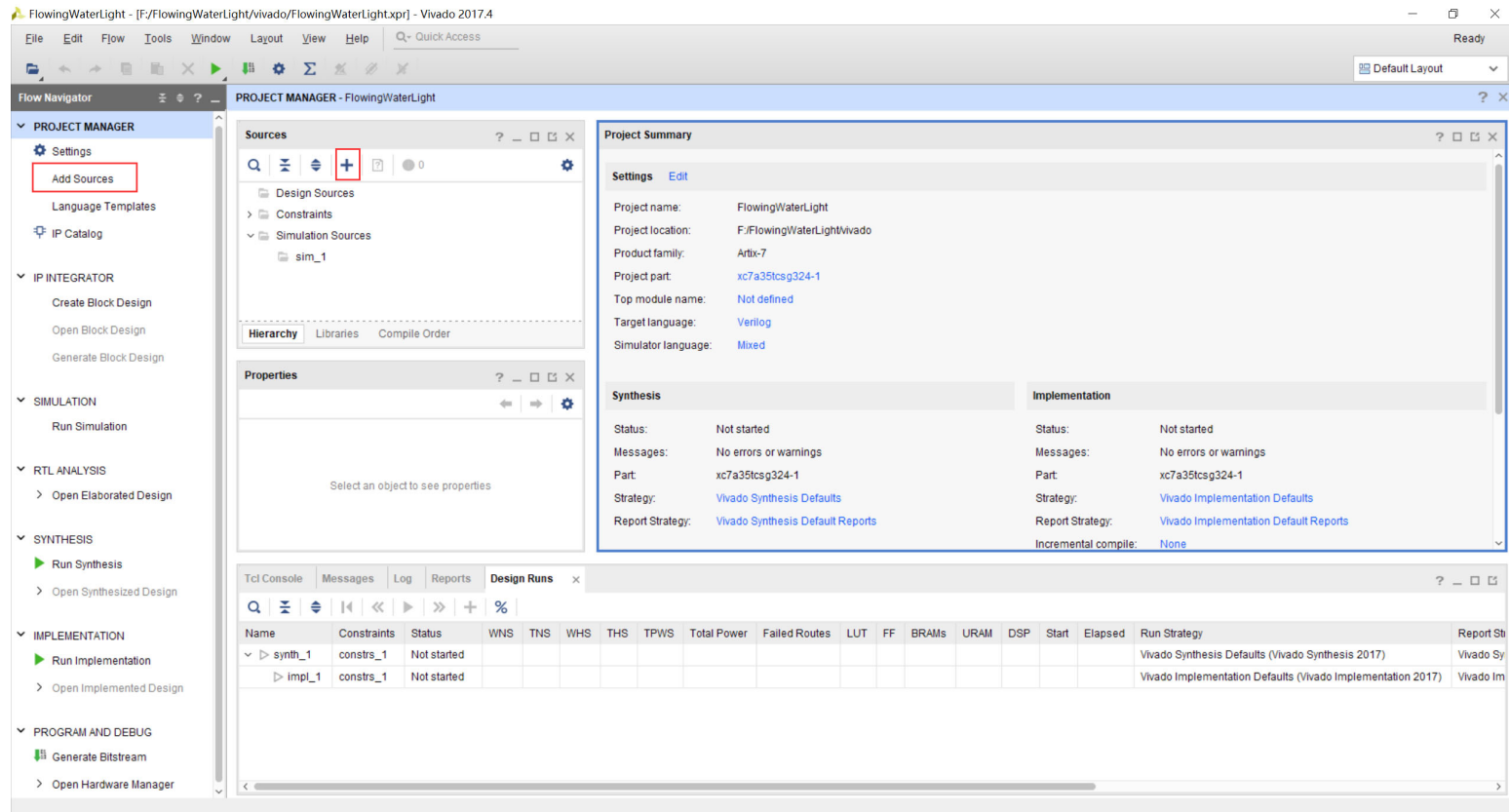
New Project Summary

i A new RTL project named 'FlowingWaterLight' will be created.

i The default part and product family for the new project:
Default Part: xc7a35tcpg236-1
Product: Artix-7
Family: Artix-7
Package: cpg236
Speed Grade: -1

设计输入

- 加入Verilog HDL源文件和仿真测试文件



设计输入

- 加入4个模块文件

Add Sources

This guides you through the process of adding and creating sources for your project

☐ Add or create constraints

☒ Add or create design sources

☐ Add or create simulation sources

Add Sources

Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

	Index	Name	Library	Location
	1	FlowingWaterLight.v	xil_defaultlib	F:/FlowingWaterLight/src
	2	counter_n.v	xil_defaultlib	F:/FlowingWaterLight/src
	3	counter_up_down.v	xil_defaultlib	F:/FlowingWaterLight/src
	4	decode.v	xil_defaultlib	F:/FlowingWaterLight/src

☐ Scan and add RTL include files into project

☐ Copy sources into project

☒ Add sources from subdirectories

设计输入

- 加入用于功能仿真的仿真文件

Add Sources

This guides you through the process of adding and creating sources for your project

☐ Add or create constraints

☐ Add or create design sources


☒ Add or create simulation sources

Add Sources

Add or Create Simulation Sources

Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.

Specify simulation set:

	Index	Name	Library	Location
	1	FlowingWaterLight_tb.v	xil_defaultlib	F:/FlowingWaterLight/src

☐ Scan and add RTL include files into project

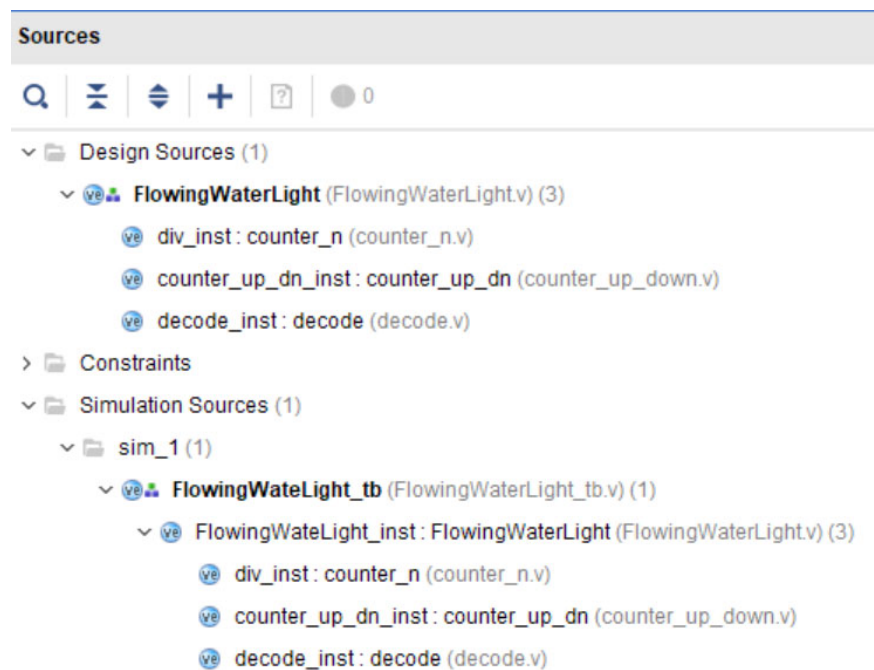
☐ Copy sources into project

☒ Add sources from subdirectories

☒ Include all design sources for simulation

设计输入

- 设计文件的层次结构



功能仿真

- 左侧Flow Navigator窗口下，点击SIMULATION->Run Simulation->Run Behavioral Simulation，Vivado将会打开仿真窗口并运行1000ns
- LED信号改成二进制观察，波形窗口右击led[7:0]信号，在弹出的快捷菜单执行Radix->Binary 命令。

SIMULATION - Behavioral Simulation - Functional - sim_1 - FlowingWaterLight_tb

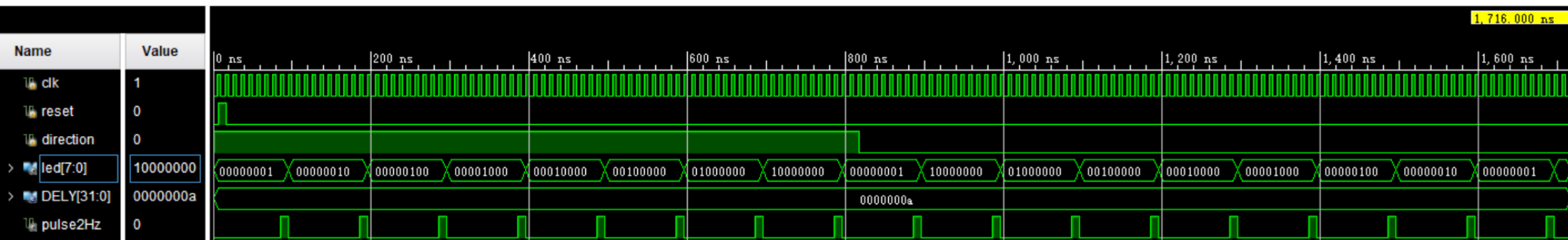
Scope		
Name	Design Unit	Block Type
FlowingWaterLight_tb	FlowingWat...	Verilog M
> FlowingWaterLight_inst	FlowingWat...	Verilog M
gbl	gbl	Verilog M

Objects	
Name	Value
clk	0
reset	0
direction	0
> led[7:0]	40
> DELY[31:0]	10

Untitled 1		1,000,000 ps				
Name	Value	999,996 ps	999,997 ps	999,998 ps	999,999 ps	1,000,000 ps
clk	0					
reset	0					
direction	0					
> led[7:0]	40					
> DELY[31:0]	0000000a		40			
			0000000a			

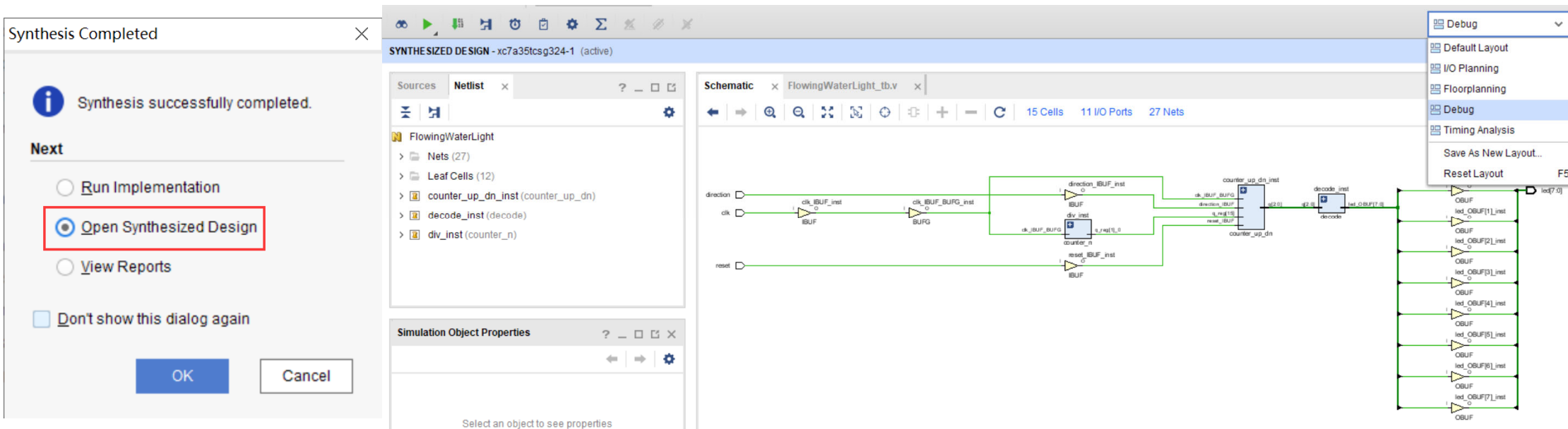
功能仿真

- 在文件层次结构区域中，选中顶层模块FlowingWaterLight_inst，在Objects中选定pulse2Hz信号，右键选Add to Wave Window加入到波形窗口。
- 执行菜单Run->Restart
- 本次仿真中，分频器被设置为10分频，以便于仿真观察
- 执行菜单Run->Run-ALL，再次仿真
- 若修改了代码，必须执行Run->Relaunch Simulation， Run->Run-ALL



电路代码综合

- 左侧Flow Navigator窗口下，点击SYNTHESIS->Run synthesis，弹出的窗口点OK，Vivado将开始综合。
- 综合完成后选择查看综合结果，在右上角Layout下拉列表选择Debug可查看原理图



约束

- 需要指定FPGA引脚位置和电器标准。因为本例流水灯是时序电路，还需要对主时钟的周期进行约束。
- 有两种方法可以添加约束文件：一是在图形界面进行设置；二是直接新建XDC约束文件，手动输入约束命令

• Basys3引脚约束

电器标准均为LVCMOS33

clk信号：系统100MHz主时钟信号

reset信号：中间按键（BTNC）

direction信号：逻辑开关SW0

led[7:0]信号：LED指示灯

LED	PIN	CLOCK	PIN	SWITCH	PIN	BUTTON	PIN	Seven-segment digital tube	PIN
LD0	U16	MRCC	W5	SW0	V17	BTNU	T18	AN0	U2
LD1	E19			SW1	V16	BTNR	T17	AN1	U4
LD2	U19			SW2	W16	BTND	U17	AN2	V4
LD3	V19			SW3	W17	BTNL	W19	AN3	W4
LD4	W18			SW4	W15	BTNC	U18	CA	W7
LD5	U15			SW5	V15			CB	W6
LD6	U14			SW6	W14			CC	U8
LD7	V14			SW7	W13			CD	V8
LD8	V13	USB(J2)	PIN	SW8	V2			CE	U5
LD9	V3	PS2_CLK	C17	SW9	T3			CF	V5
LD10	W3	PS2_DAT	B17	SW10	T2			CG	U7
LD11	U3			SW11	R3			DP	V7
LD12	P3			SW12	W2				
LD13	N3			SW13	U1				
LD14	P1			SW14	T1				
LD15	L1			SW15	R2				

约束

- EGO1引脚约束

电器标准均为LVCMOS33

clk信号：系统100MHz主时钟信号

reset信号：复位引脚

direction信号：逻辑开关SW0

led[7:0]信号：LED指示灯

名称	原理图标号	FPGA IO PIN
时钟引脚	SYS_CLK	P17

名称	原理图标号	FPGA IO PIN
复位引脚	FPGA_RESET	P15

名称	原理图标号	FPGA IO PIN
SW0	SW_0	P5

名称	原理图标号	FPGA IO PIN	颜色
D0	LED0	F6	Green
D1	LED1	G4	Green
D2	LED2	G3	Green
D3	LED3	J4	Green
D4	LED4	H4	Green
D5	LED5	J3	Green
D6	LED6	J2	Green
D7	LED7	K2	Green

约束

The screenshot displays the Xilinx Vivado IDE interface for a project named "SYNTHESIZED DESIGN * - xc7a35tcs324-1 (active)". The top toolbar includes icons for Run, Simulation, and other functions. The left sidebar shows the "Sources" pane with "Internal VREF" settings (0.6V, 0.675V, 0.75V, 0.9V, and NONE (5)). Below it, the "I/O Port Properties" pane shows the "direction" property set to "IN" and "Package pin" set to "P5".

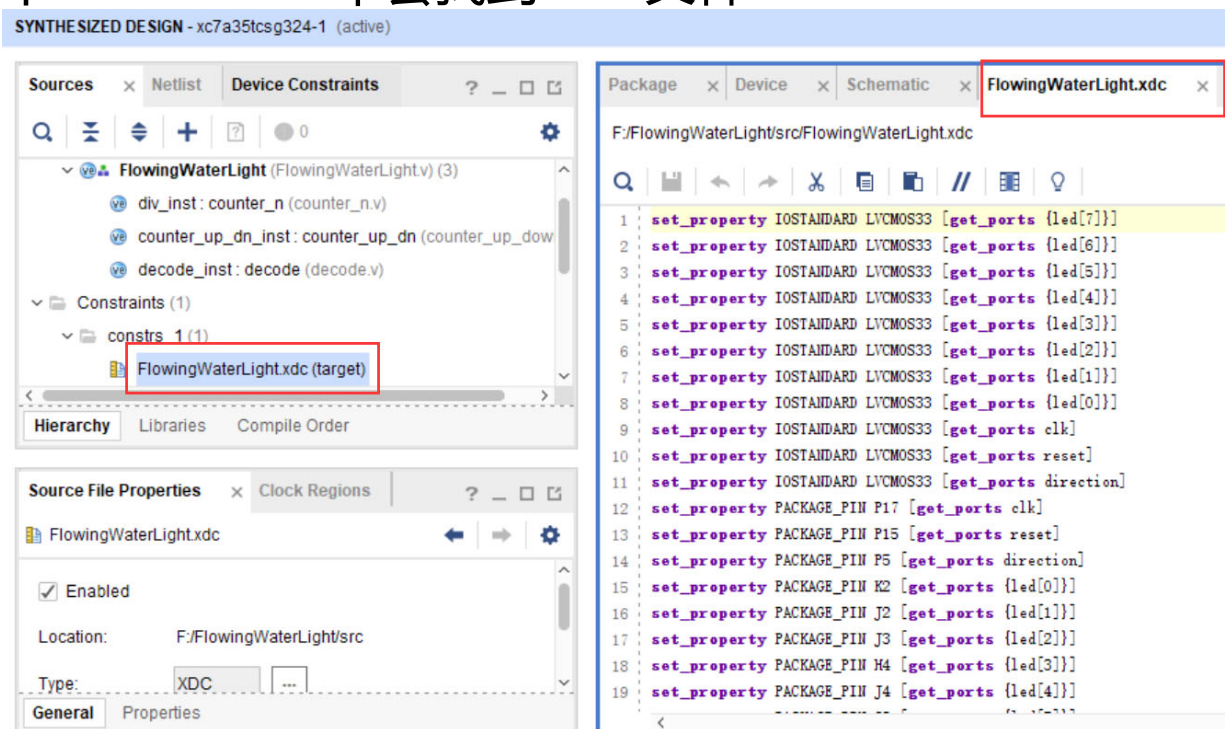
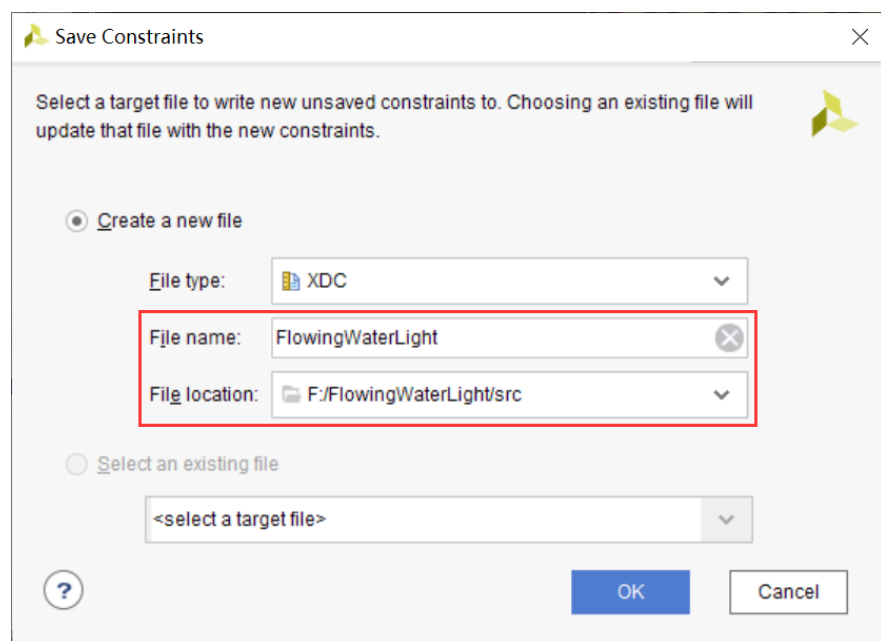
The main workspace shows the "I/O Planning" view, which is a grid of pins (A-U, 1-18) with various colored markers indicating pin assignments. A red box highlights the "I/O Planning" option in the top right menu.

The bottom pane shows the "I/O Ports" table, which lists the configured ports and their properties. The table has columns for Name, Direction, Neg Diff Pair, Package Pin, Fixed, Bank, I/O Std, Vcco, Vref, Drive Strength, Slew Type, Pull Type, and Off-Chip Termination. The "I/O Ports" tab is selected, and the table lists 11 ports, including "led" and "led[7]".

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination
led (8)	OUT			<input checked="" type="checkbox"/>	35	LVCN0533*	3.300	12	SLOW	NONE	FP_VTT_50	
led[7]	OUT		F6	<input checked="" type="checkbox"/>	35	LVCN0533*	3.300	12	SLOW	NONE	FP_VTT_50	
led[6]	OUT		G4	<input checked="" type="checkbox"/>	35	LVCN0533*	3.300	12	SLOW	NONE	FP_VTT_50	
led[5]	OUT		G3	<input checked="" type="checkbox"/>	35	LVCN0533*	3.300	12	SLOW	NONE	FP_VTT_50	
led[4]	OUT		J4	<input checked="" type="checkbox"/>	35	LVCN0533*	3.300	12	SLOW	NONE	FP_VTT_50	

约束

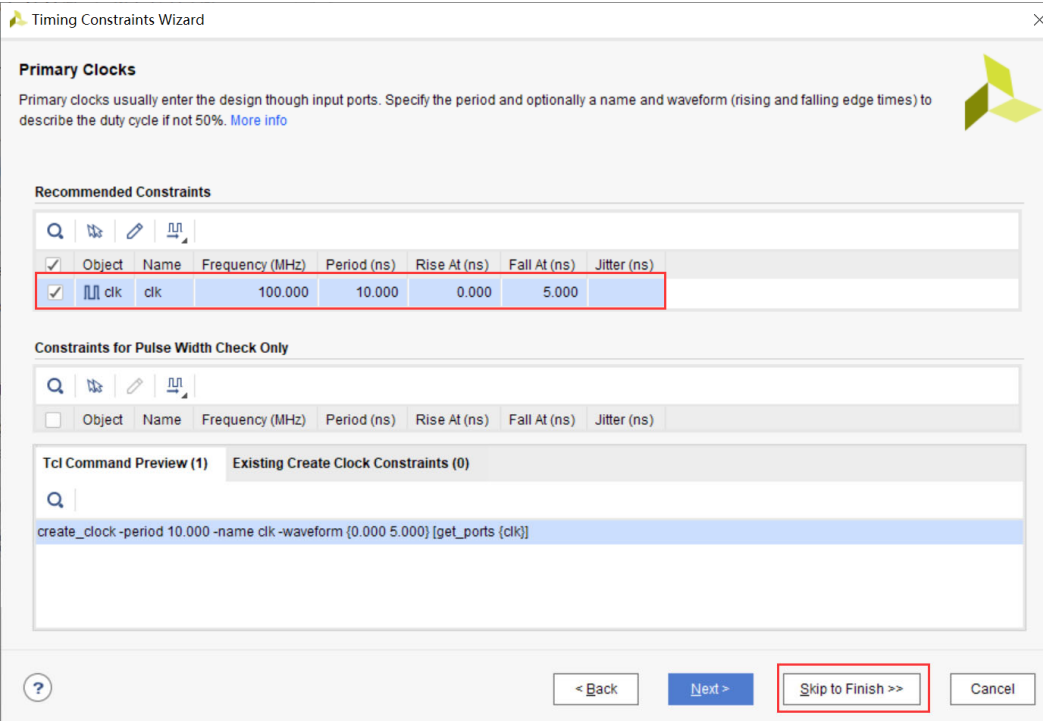
- 完成引脚约束后，点击工具栏的保存按钮，如图窗口输入XDC文件名和保存位置，本例保存在源文件夹src下。
- 完成引脚约束后，在Sources窗口下Constraints中会找到XDC文件



约束

- 时钟约束 (时序电路)

左侧Flow Navigator窗口下，点击SYNTHESIS->Open Synthesis Design -> Constraints->Constraints Wizard,在弹出窗口点Next，设置时钟频率100MHz即可



The image shows the 'Timing Constraints Wizard' dialog box. It has a title bar with a close button. The main content area is divided into sections. The 'Primary Clocks' section has a description and a 'Recommended Constraints' table. The table has columns for Object, Name, Frequency (MHz), Period (ns), Rise At (ns), Fall At (ns), and Jitter (ns). A row is highlighted with a red border, showing a clock object named 'clk' with a frequency of 100.000 MHz, a period of 10.000 ns, and a jitter of 5.000 ns. Below this is a section for 'Constraints for Pulse Width Check Only' with a similar table. At the bottom, there is a 'Tcl Command Preview' section showing the command 'create_clock -period 10.000 -name clk -waveform {0.000 5.000} [get_ports {clk}]'. The bottom of the dialog has navigation buttons: '< Back', 'Next >', 'Skip to Finish >>' (highlighted with a red border), and 'Cancel'.

Timing Constraints Wizard

Primary Clocks

Primary clocks usually enter the design through input ports. Specify the period and optionally a name and waveform (rising and falling edge times) to describe the duty cycle if not 50%. [More info](#)

Recommended Constraints

Object	Name	Frequency (MHz)	Period (ns)	Rise At (ns)	Fall At (ns)	Jitter (ns)
<input checked="" type="checkbox"/>	clk	100.000	10.000	0.000	5.000	

Constraints for Pulse Width Check Only

Object	Name	Frequency (MHz)	Period (ns)	Rise At (ns)	Fall At (ns)	Jitter (ns)
--------	------	-----------------	-------------	--------------	--------------	-------------

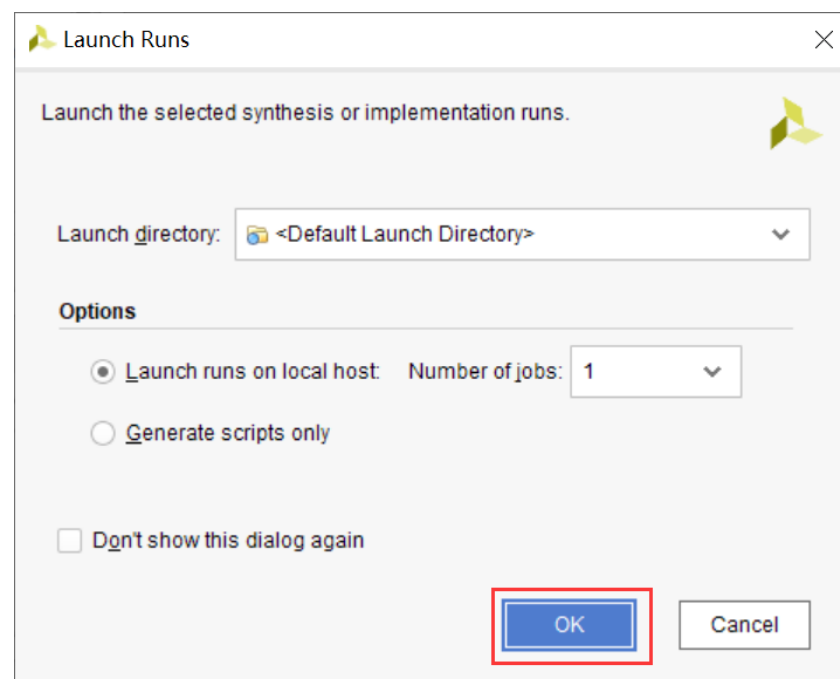
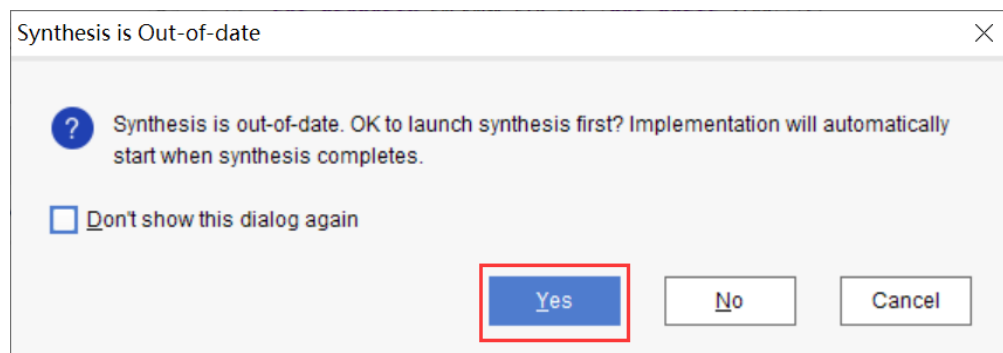
Tcl Command Preview (1) Existing Create Clock Constraints (0)

```
create_clock -period 10.000 -name clk -waveform {0.000 5.000} [get_ports {clk}]
```

< Back Next > Skip to Finish >> Cancel

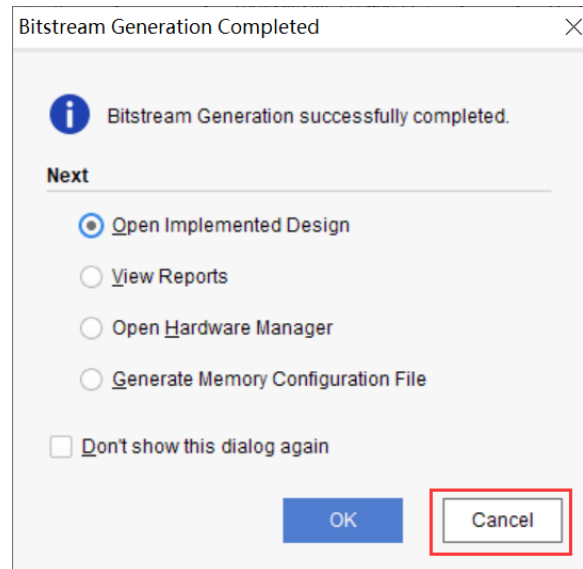
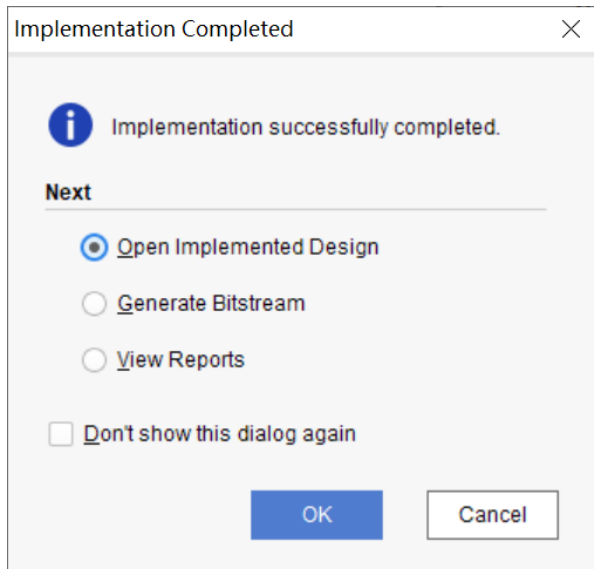
工程实现

- Implementation是指将综合输出的网表适配到FPGA上
- 左侧Flow Navigator窗口下，点击IMPLEMENTATION->Run Implementation
- 由于约束后没有重新进行综合，Vivado会依次完成综合、实现两个过程。



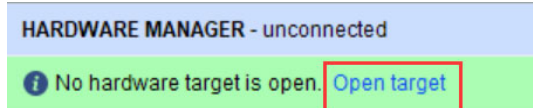
生成FPGA编程比特流

- Implementation完成后可以直接进行比特流生成工作
- 也可以通过左侧Flow Navigator窗口下，点击PROGRAM AND DEBUG->Generate Bitstream，在弹出窗口直接点击OK开始生成
- 完成后，在弹出的界面也可单击取消



编程下载

- 用USB线将开发板与计算机的USB接口连接
- 打开开发板电源
- 左侧Flow Navigator窗口下，点击PROGRAM AND DEBUG->Open Hardware Manager



- 在HARDWARE MANAGER窗口单击Open Target，在点击Auto Connect，系统会自动查找设备。
- 连接完成后，FPGA芯片会出现在Hardware窗口里
- 在HARDWARE MANAGER窗口单击Program device，在弹出窗口中单击Program即可完成下载编程。