实验十二

数码管显示译码器的实现

实验目的

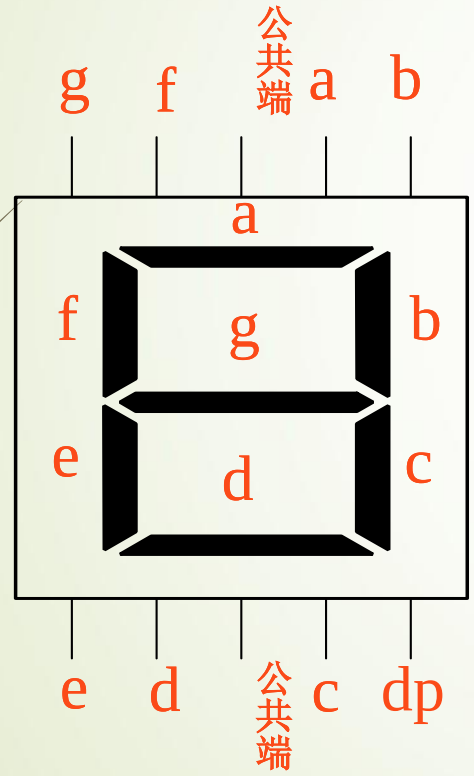
- 掌握一位数码管显示译码器的工作原理
- 学习Verilog HDL语言中的CASE语句
- 学习在Vivado开发环境下电路仿真的过程和方法
- 熟练掌握在Vivado开发环境下运用Verilog HDL语言实现显示译码器电路的描述、综合、实现、仿真和下载

（一）
数码管显示译码器的工作原理及硬件语言描述

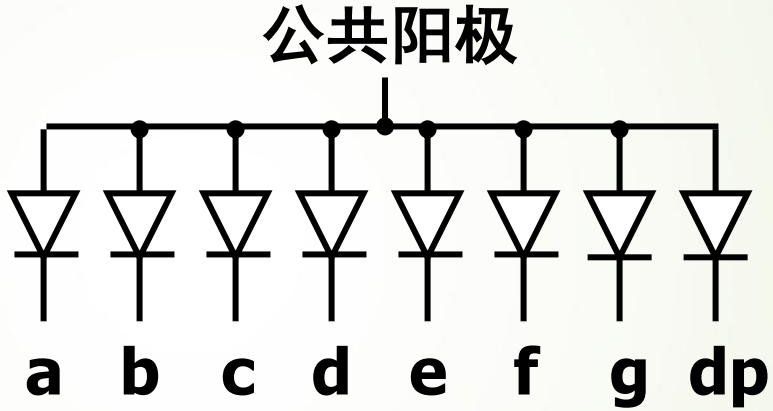
实验原理

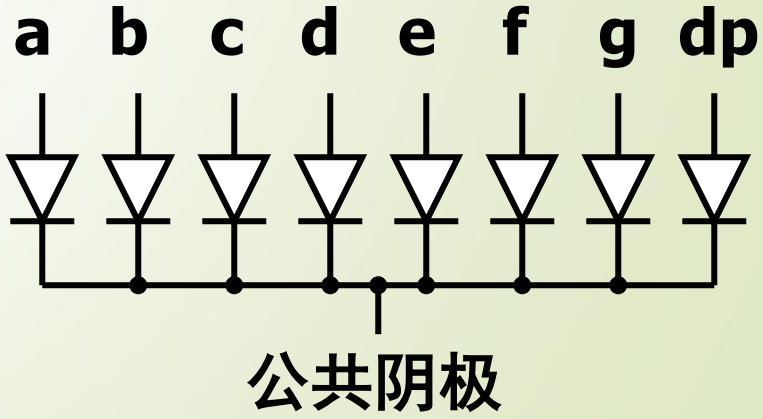
七段显示数码管的使用

LED七段显示器分为共阴极和共阳极两种。它们所需要的显示译码器各不相同。



七段显示器的外形结构

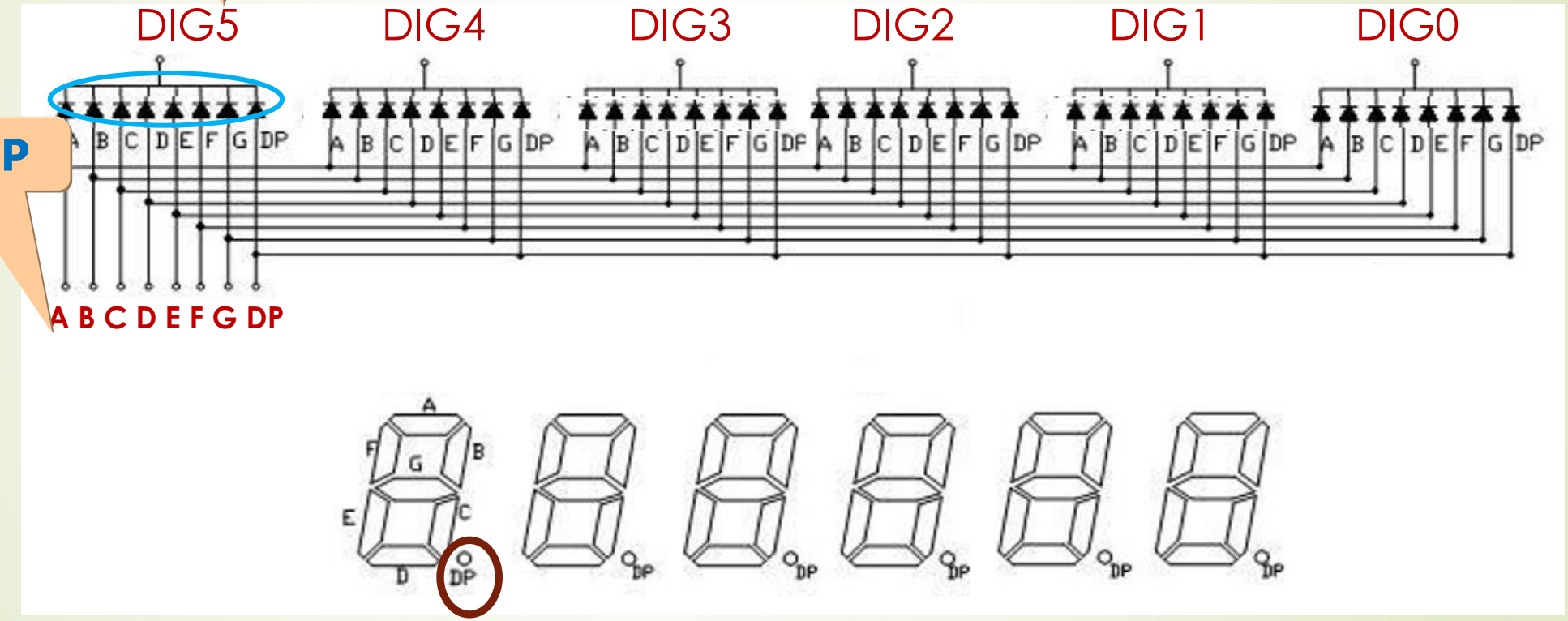




七段显示数码管的使用

位码
DIG[5:0]

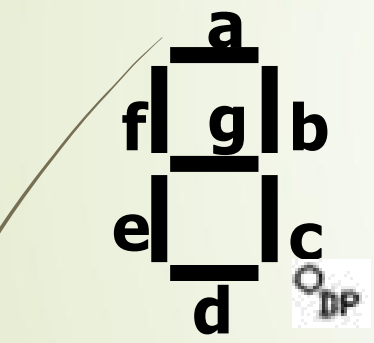
段码**A~DP**

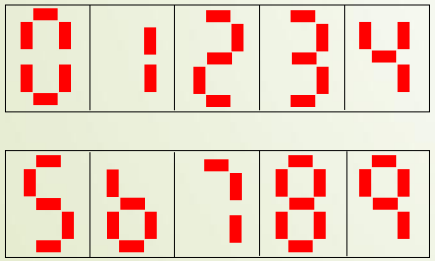


本实验开发板上有6位一体七段共阴LED数码管，位码低有效，段码高有效

8421BCD码——七段显示译码器真值表

驱动共阴极LED
七段显示器

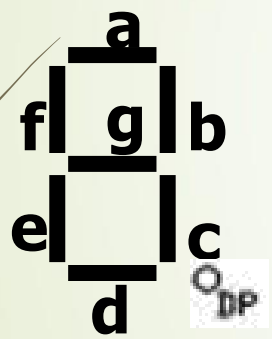


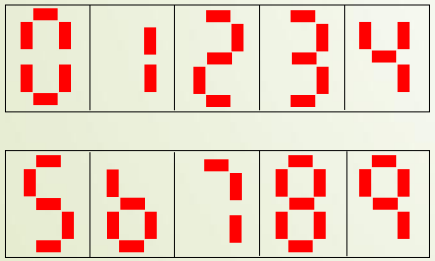


	D	C	B	A	a	b	c	d	e	f	g	DP
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1	0
3	0	0	1	1	1	1	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	0	1	1	0	1	1	0
6	0	1	1	0	0	0	1	1	1	1	1	0
7	0	1	1	1	1	1	1	0	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1	0
9	1	0	0	1	1	1	1	0	0	1	1	0
10	1	0	1	0	任意值							
11	1	0	1	1								
12	1	1	0	0								
13	1	1	0	1								
14	1	1	1	0								
15	1	1	1	1								

8421BCD码——七段显示译码器真值表

驱动共阴极LED
七段显示器



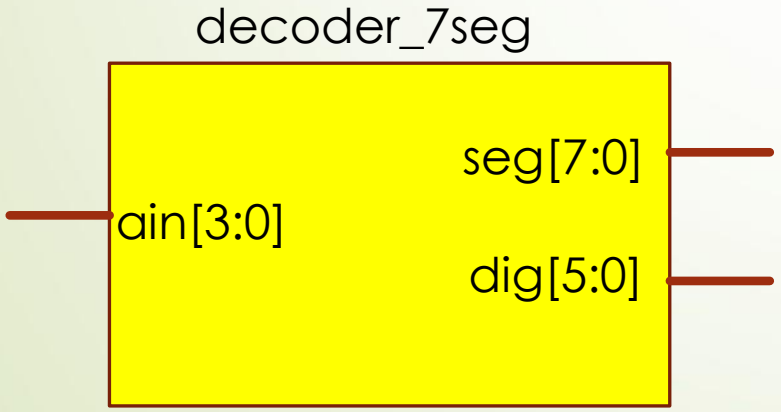


	D	C	B	A	a	b	c	d	e	f	g	DP
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1	0
3	0	0	1	1	1	1	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	0	1	1	0	1	1	0
6	0	1	1	0	0	0	1	1	1	1	1	0
7	0	1	1	1	1	1	1	0	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1	0
9	1	0	0	1	1	1	1	0	0	1	1	0
10	1	0	1	0	无显示							
11	1	0	1	1								
12	1	1	0	0								
13	1	1	0	1								
14	1	1	1	0								
15	1	1	1	1								

8421BCD-七段显示译码器（显示0~9）
主程序

- module decoder_7seg (
- input [3:0] ain, //输入四位二进制码
- output reg [7:0] seg, //段码
- output [5:0] dig //位码
-);
- assign dig = 6‘b011111;//显示在左边第一个数码管

```
always@(ain)
        case(ain)
        4'b0000: seg = 8'b11111100;//a~g
        4'b0001: seg = 8'b01100000;
        4'b0010: seg = 8'b11011010;
        4'b0011: seg = 8'b11110010;
        4'b0100: seg = 8'b01100110;
        4'b0101: seg = 8'b10110110;
        4'b0110: seg = 8'b00111110;
        4'b0111: seg = 8'b11100000;
        4'b1000: seg = 8'b11111110;
        4'b1001: seg = 8'b11100110;
          default: seg = 8'b00000000;
        endcase
endmodule
```



本实验中Verilog的关键语句

case语句 —— 多分支语句

适于对同一个控制信号取不同的值时，输出取不同的值！

- 当敏感表达式取不同的值时，执行不同的语句。
- **功能**：当某个（控制）信号取不同的值时，给另一个（输出）信号赋不同的值。常用于多条件译码电路（如译码器、数据选择器、状态机、微处理器的指令译码）！
- case语句有3种形式：case，casez，casex —— 是case语句的两种变体

case语句

```
case（敏感表达式）
      值1：语句1；
      值2：语句2；
        …
      值n：语句n；
      default：语句n+1；
endcase
```

case语句与if-else语句有什么区别呢？

If-else对于每个判定只有两个分支

本实验中Verilog的关键语句

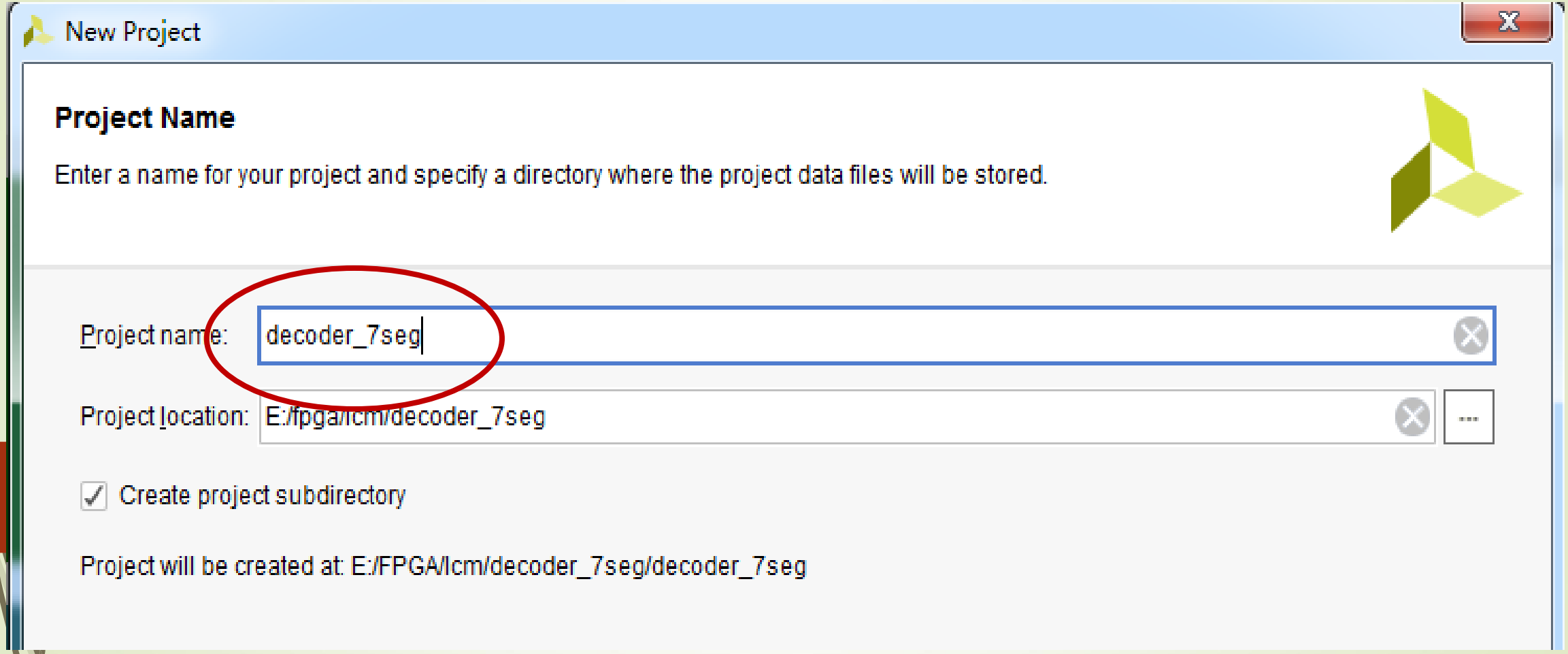
- 说明：
  - 其中“敏感表达式”又称为“控制表达式”，通常表示为控制信号的某些位。
  - 值1~值n称为分支表达式，用控制信号的具体状态值表示，因此又称为常量表达式。
  - default项可有可无，一个case语句里只能有一个default项!但在组合电路设计中，为避免生成隐含锁存器，在case语句最后必须写上default项。
  - 值1~值n必须互不相同，否则矛盾。
  - 值1~值n的位宽必须相等，且与控制表达式的位宽相同。

实验十二
数码管显示译码器的实现

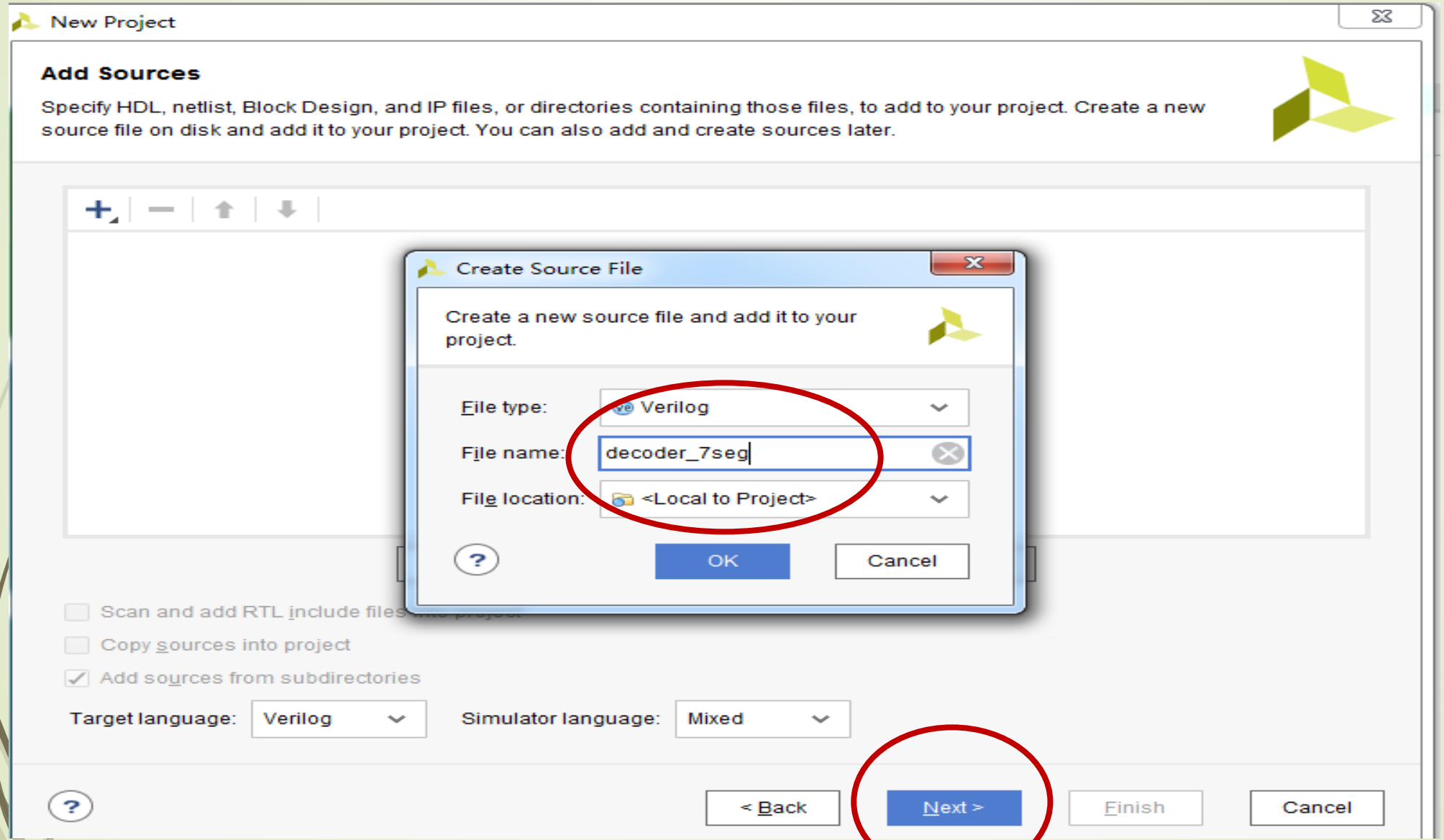
（二）

BCD码-七段显示译码器（0~9）
设计到下载主要流程

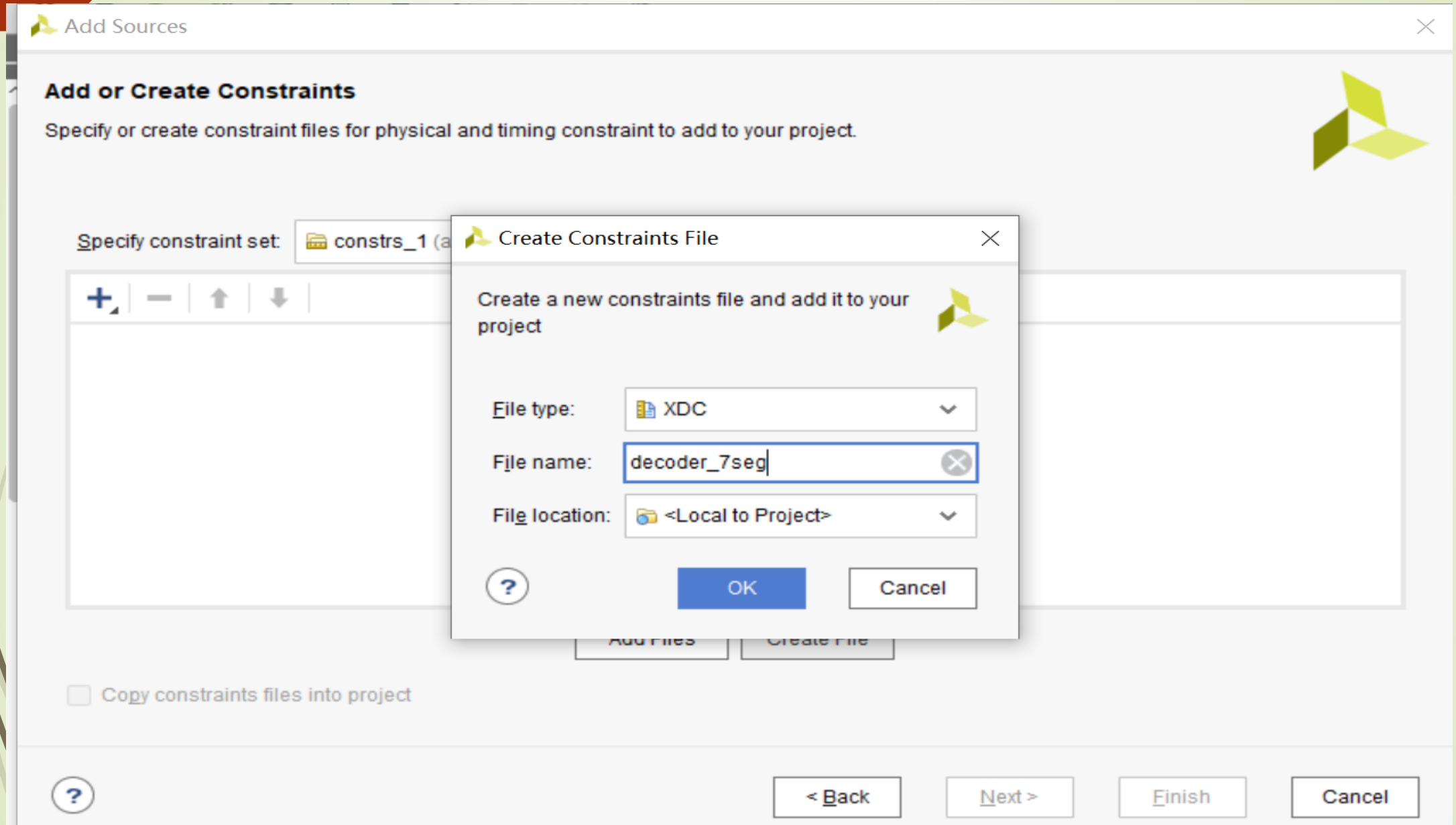
创建新的工程


New Project
Project Name
Enter a name for your project and specify a directory where the project data files will be stored.
Project name:
decoder_7seg
Project location:
E:/fpga/lcm/decoder_7seg
Create project subdirectory
Project will be created at: E:/FPGA/lcm/decoder_7seg/decoder_7seg

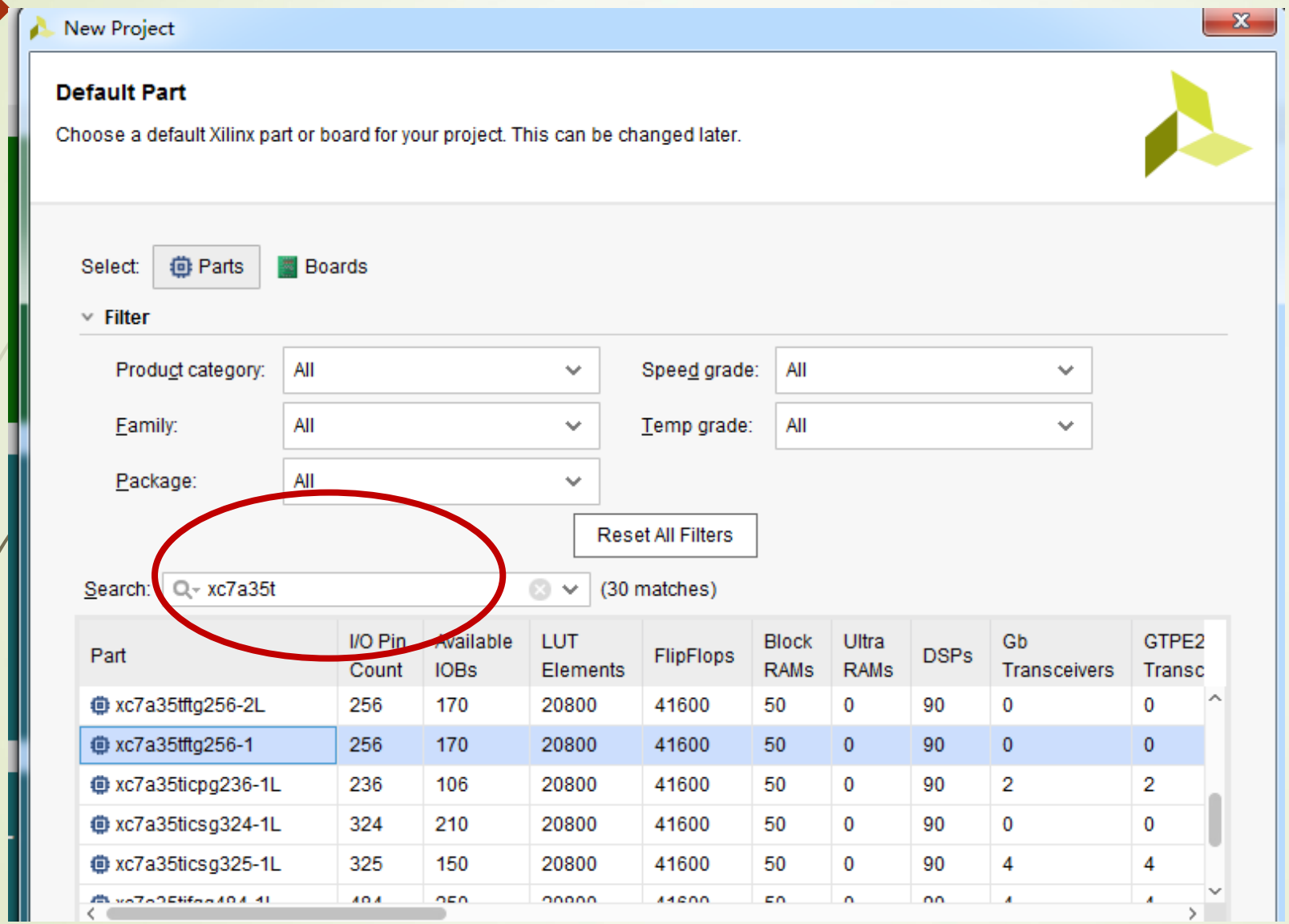
创建一个新的设计文件


New Project
Add Sources
Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.
Create Source File
Create a new source file and add it to your project.
File type:
Verilog
File name:
decoder_7seg
File location:
<Local to Project>
OK
Cancel
Scan and add RTL include files
Copy sources into project
Add sources from subdirectories
Target language:
Verilog
Simulator language:
Mixed
< Back
Next >
Finish
Cancel

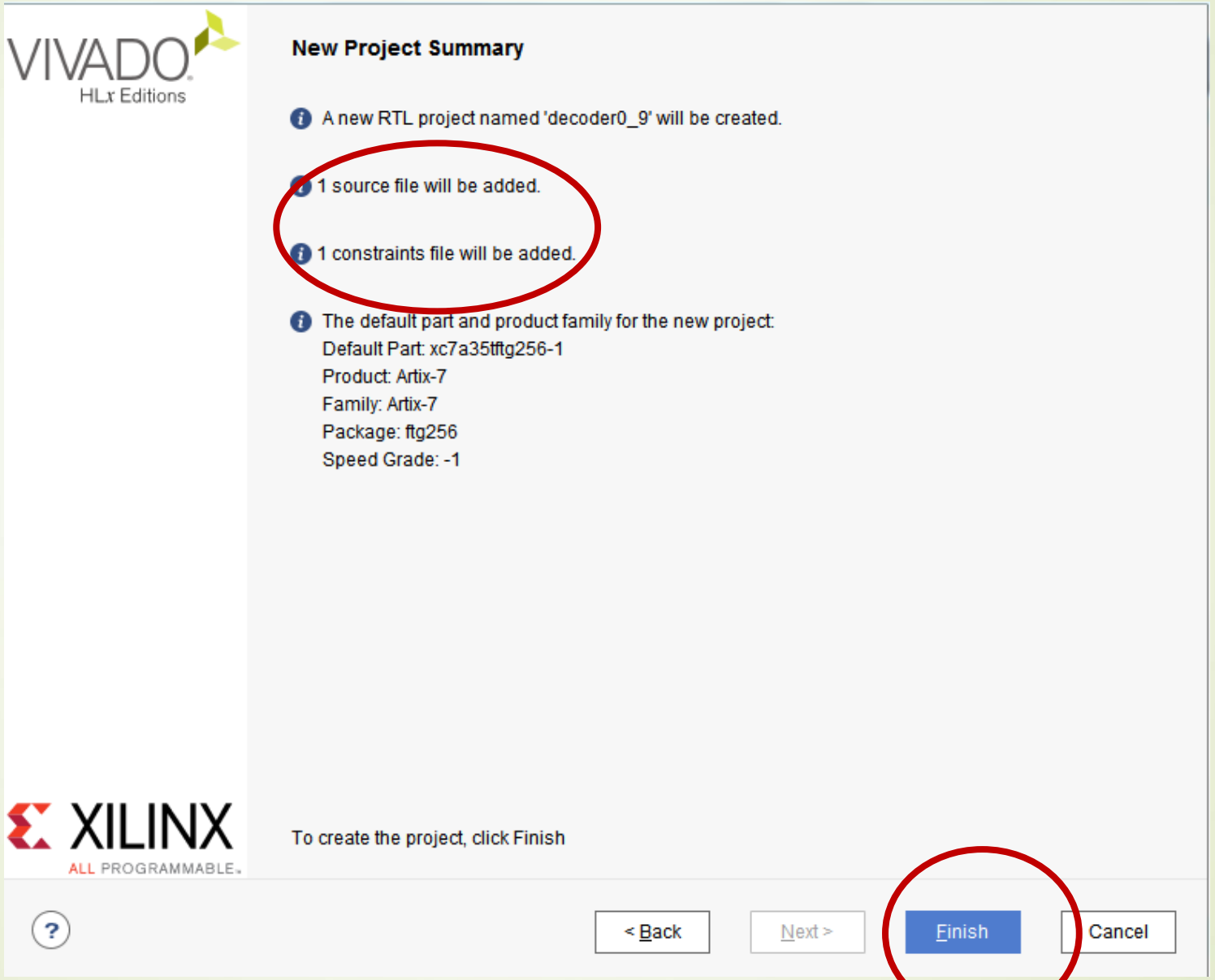
创建一个新的约束文件


Add Sources
Add or Create Constraints
Specify or create constraint files for physical and timing constraint to add to your project.
Specify constraint set:
constrs_1
Create Constraints File
Create a new constraints file and add it to your project
File type:
XDC
File name:
decoder_7seg
File location:
<Local to Project>
OK
Cancel
Copy constraints files into project
< Back
Next >
Finish
Cancel

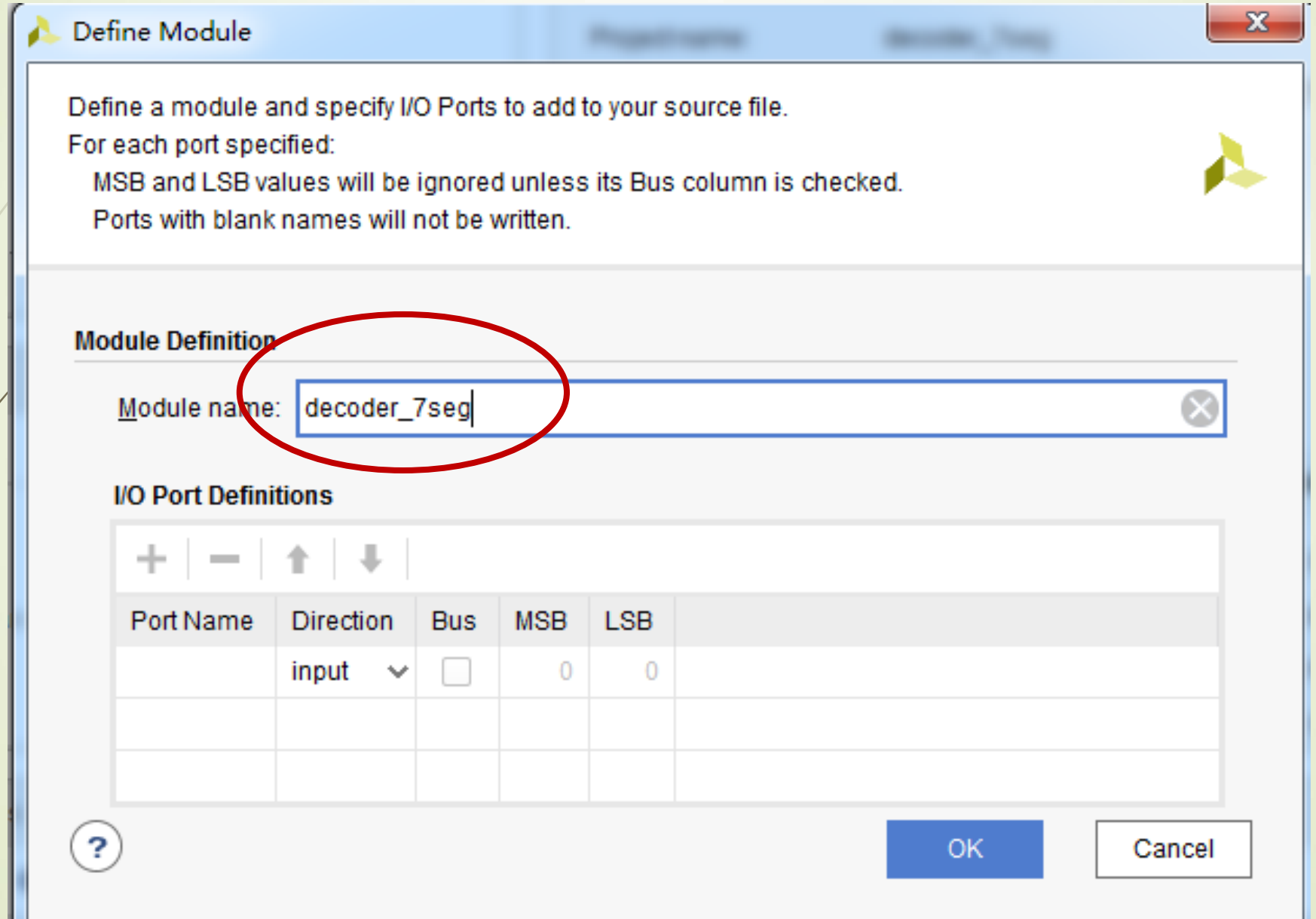
选择器件类型（xc7a35tftg256-1）



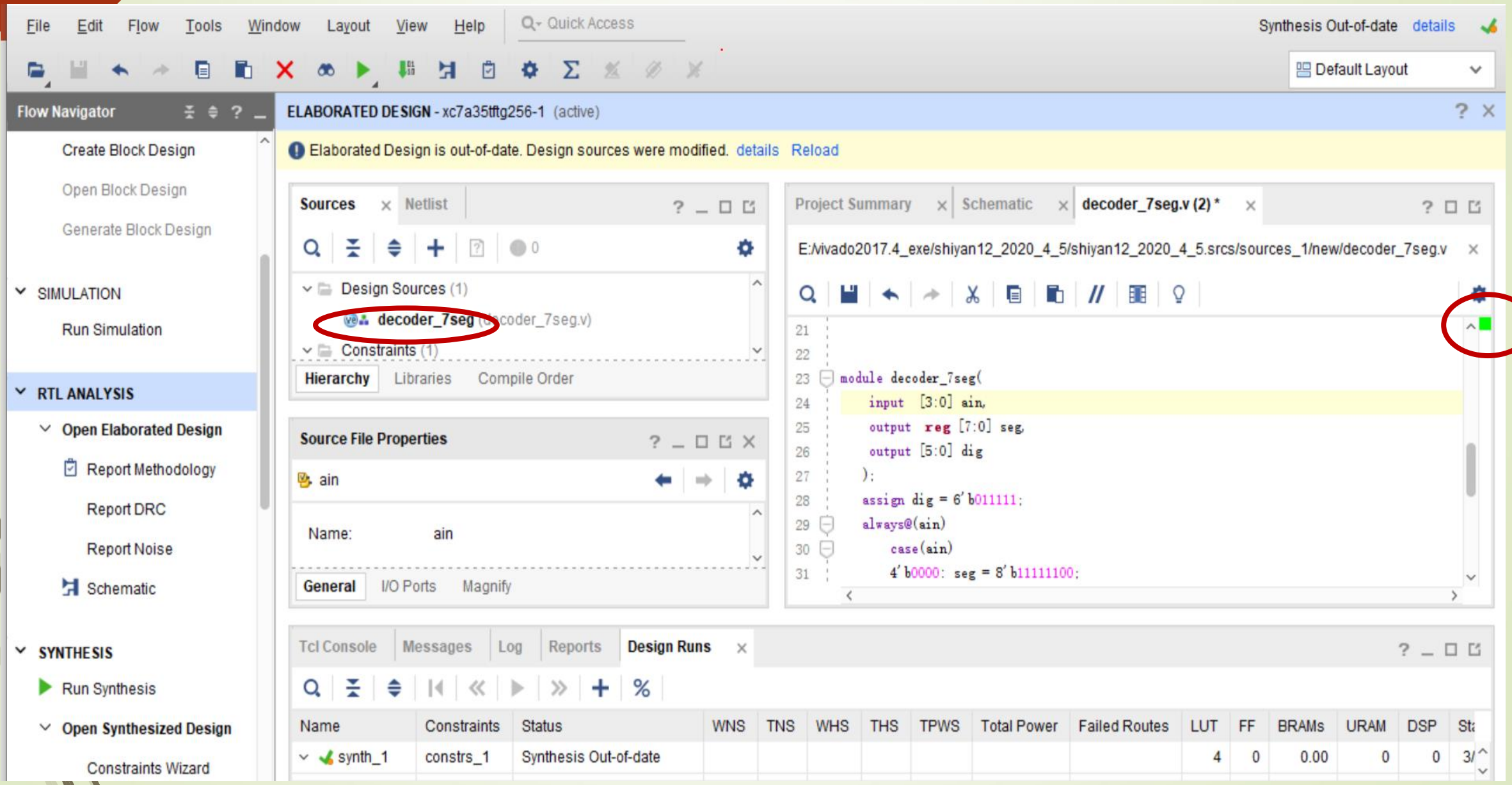
新工程概要


VIVADO
HLx Editions
New Project Summary
A new RTL project named 'decoder0_9' will be created.
1 source file will be added.
1 constraints file will be added.
The default part and product family for the new project:
Default Part: xc7a35tftg256-1
Product: Artix-7
Family: Artix-7
Package: ftg256
Speed Grade: -1
XILINX
ALL PROGRAMMABLE
To create the project, click Finish
?
< Back
Next >
Finish
Cancel

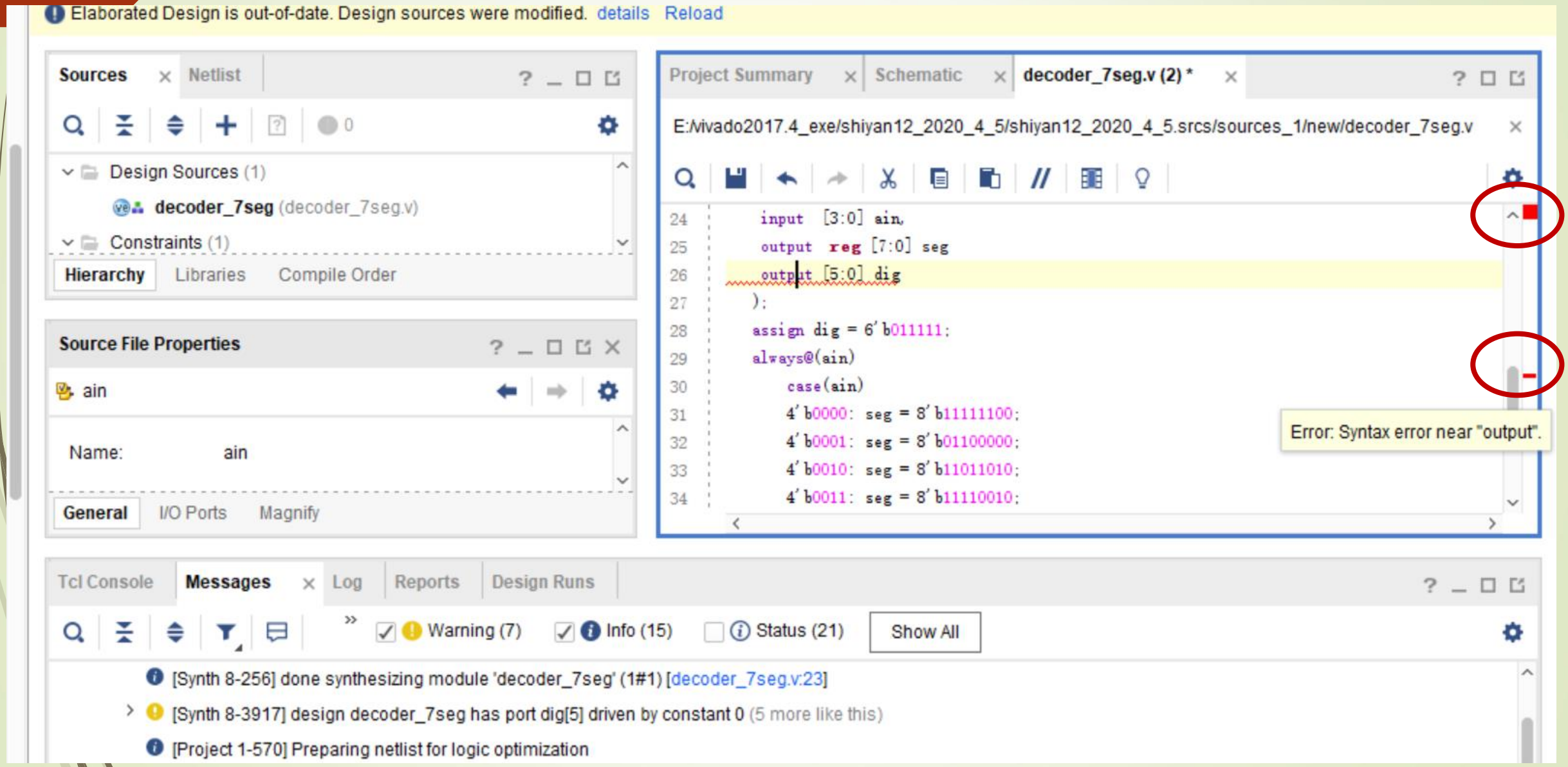
定义新的模块名


Define Module
Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.
Module Definition
Module name:
decoder_7seg
I/O Port Definitions
Port Name
Direction
Bus
MSB
LSB
input
0
0
OK
Cancel

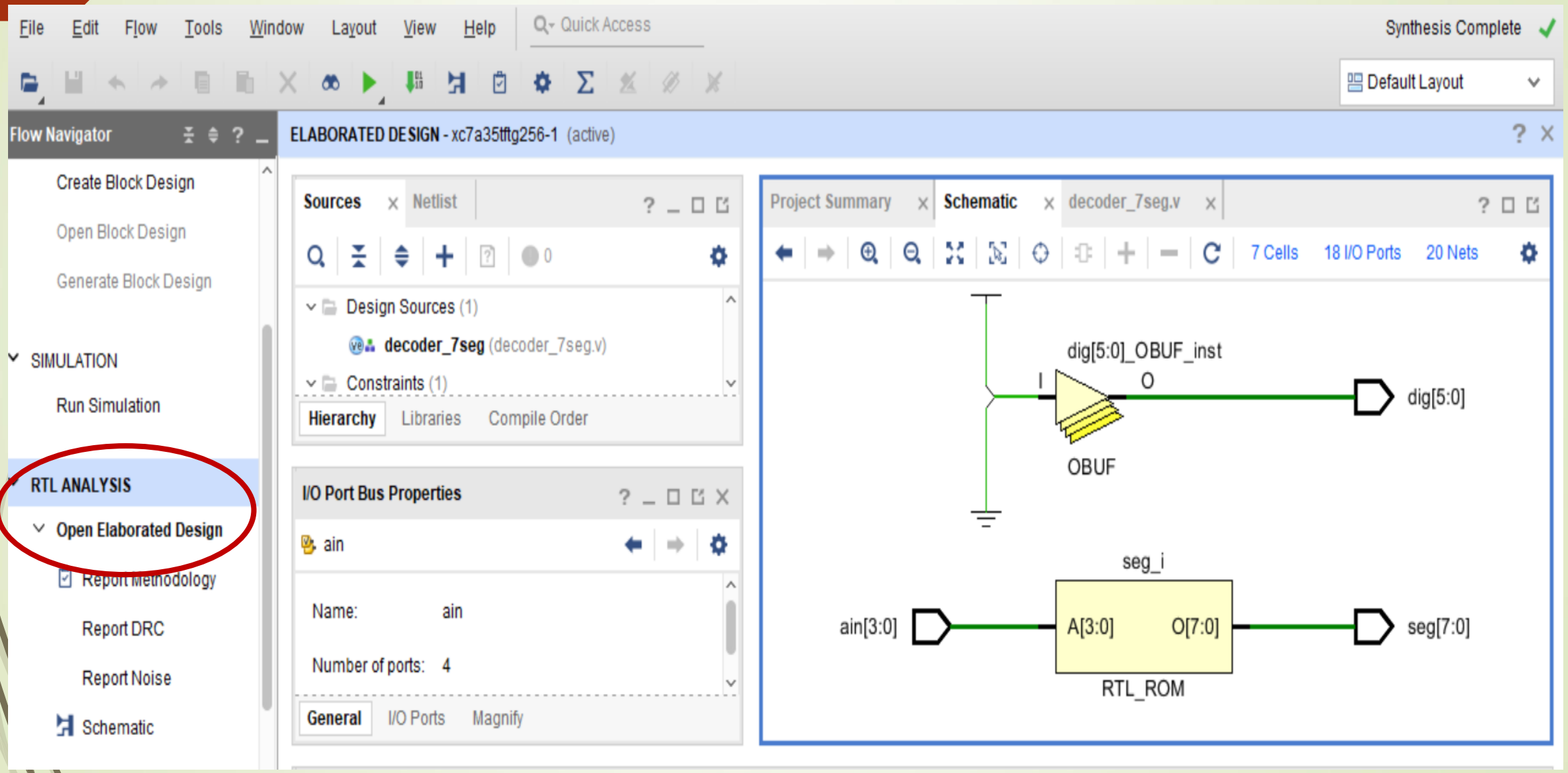
双击模块名，添加设计文件代码，按保存



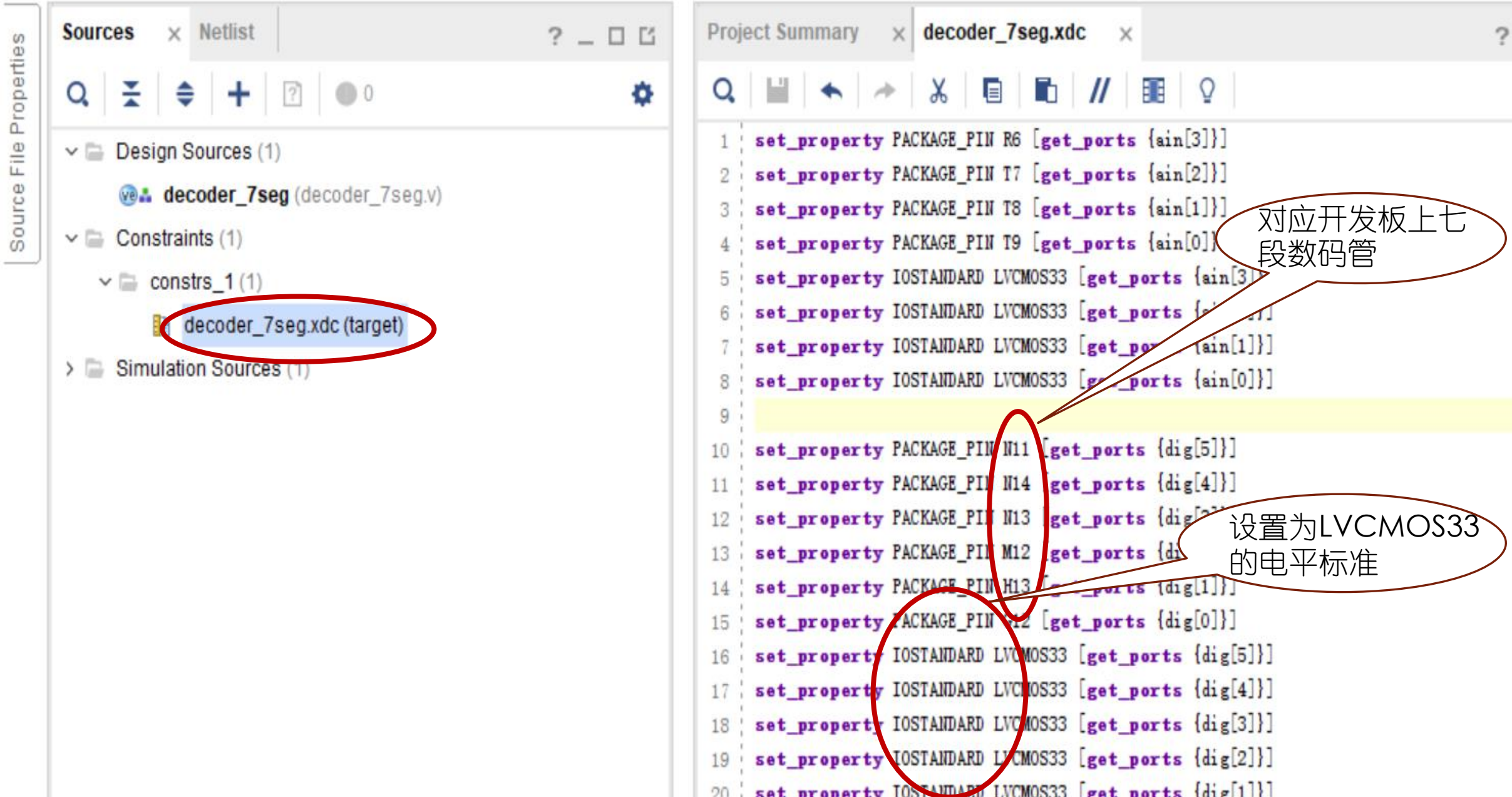
如果设计文件代码有语法错误，右侧色块显示为红色



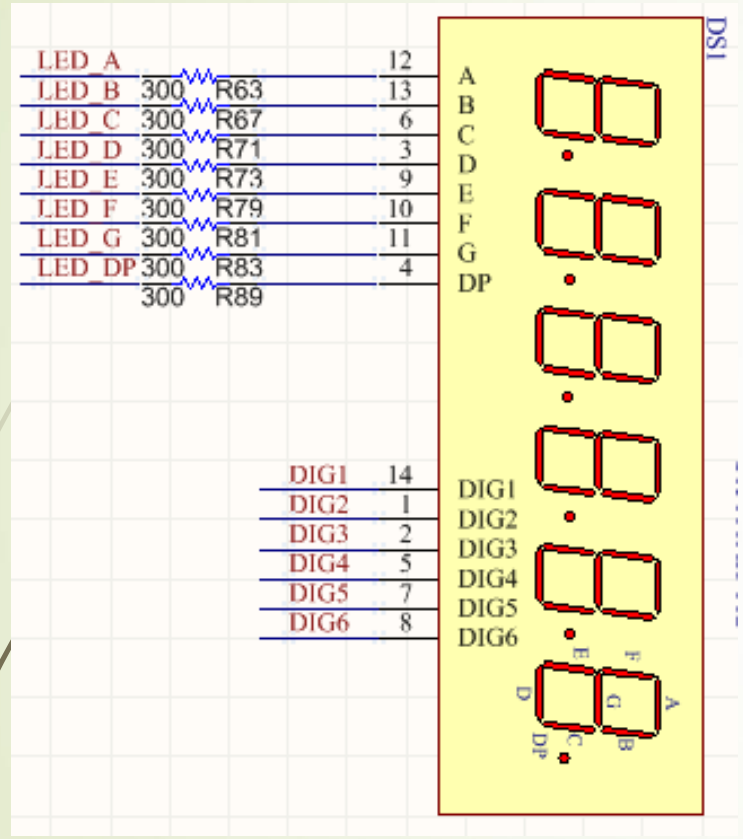
查看RTL分析原理图


File
Edit
Flow
Tools
Window
Layout
View
Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
ELABORATED DESIGN - xc7a35tftg256-1 (active)
Sources
Netlist
Design Sources (1)
decoder_7seg (decoder_7seg.v)
Constraints (1)
Hierarchy
Libraries
Compile Order
I/O Port Bus Properties
ain
Name: ain
Number of ports: 4
General
I/O Ports
Magnify
Project Summary
Schematic
decoder_7seg.v
7 Cells
18 I/O Ports
20 Nets
dig[5:0]_OBUF_inst
I
O
OBUF
dig[5:0]
seg_i
ain[3:0]
A[3:0]
O[7:0]
RTL_ROM
seg[7:0]

添加约束文件

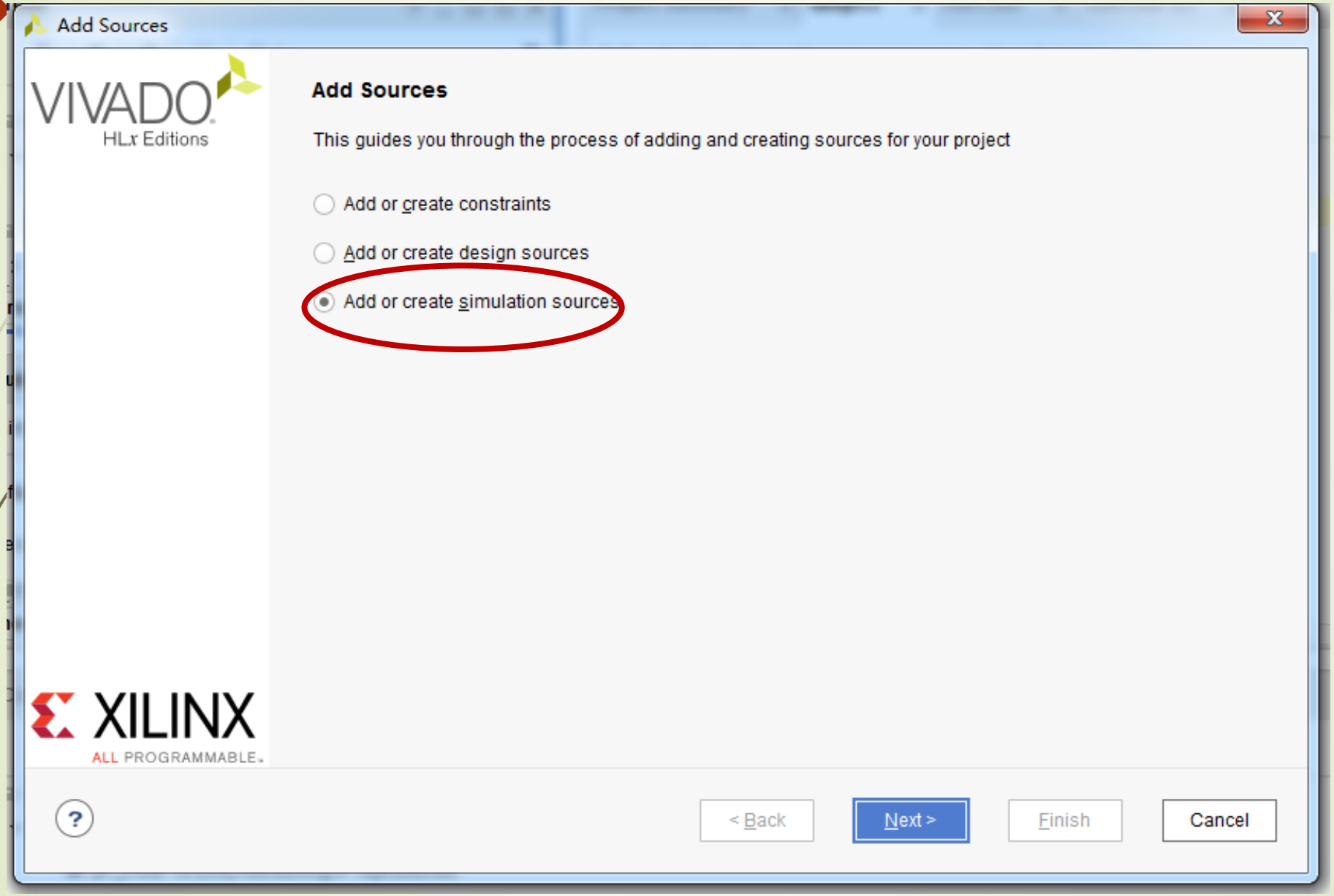

Source File Properties
Sources
Netlist
Design Sources (1)
decoder_7seg (decoder_7seg.v)
Constraints (1)
constrs_1 (1)
decoder_7seg.xdc (target)
Simulation Sources (1)
Project Summary
decoder_7seg.xdc
set_property PACKAGE_PIN R6 [get_ports {ain[3]}]
set_property PACKAGE_PIN T7 [get_ports {ain[2]}]
set_property PACKAGE_PIN T8 [get_ports {ain[1]}]
set_property PACKAGE_PIN T9 [get_ports {ain[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ain[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ain[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ain[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {ain[0]}]
set_property PACKAGE_PIN N11 [get_ports {dig[5]}]
set_property PACKAGE_PIN N14 [get_ports {dig[4]}]
set_property PACKAGE_PIN N13 [get_ports {dig[3]}]
set_property PACKAGE_PIN M12 [get_ports {dig[2]}]
set_property PACKAGE_PIN H13 [get_ports {dig[1]}]
set_property PACKAGE_PIN J12 [get_ports {dig[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {dig[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {dig[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {dig[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {dig[2]}]
对应开发板上七段数码管
设置为LVCMOS33的电平标准

七段数码管引脚资料



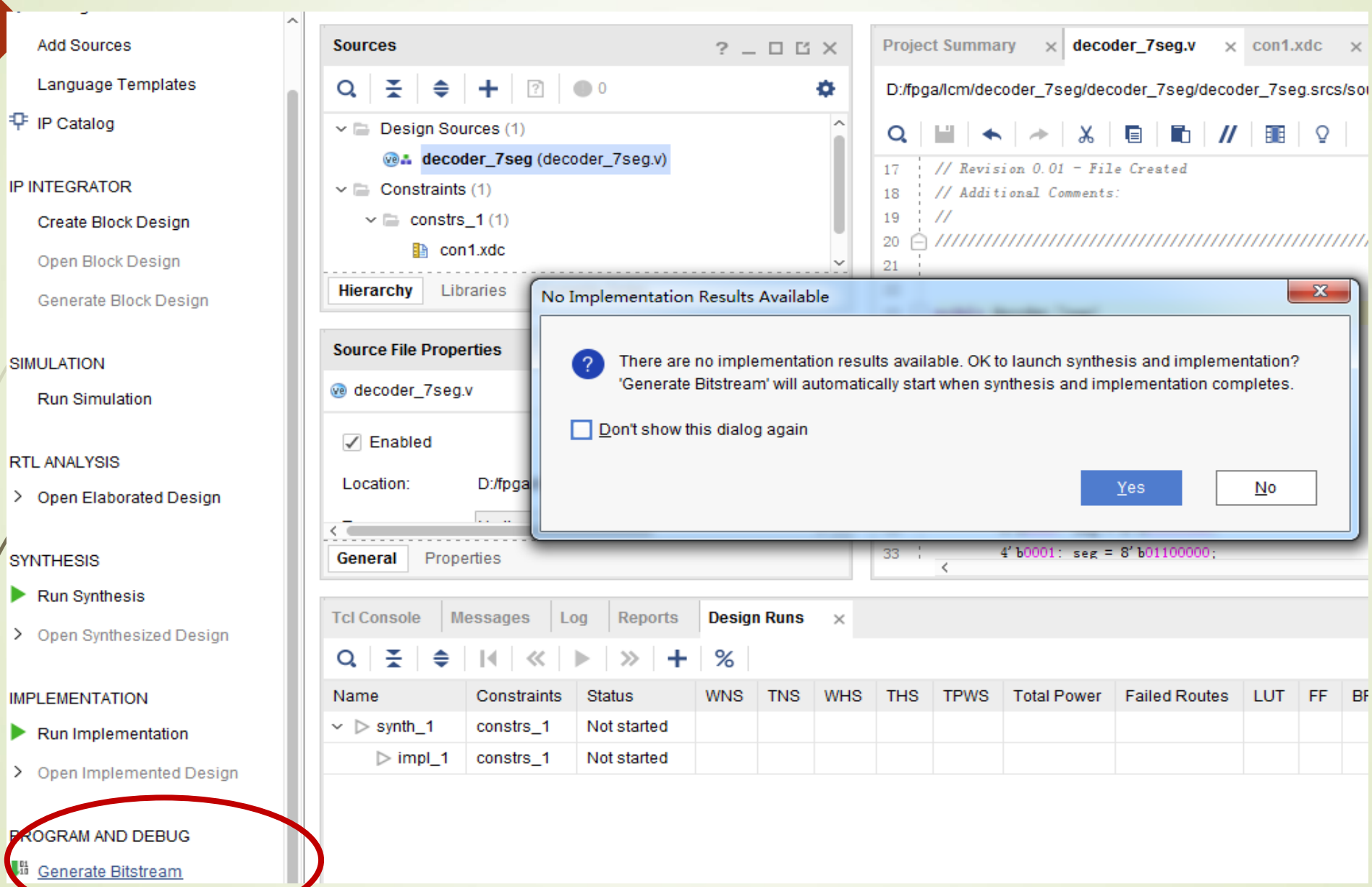
数码管	N11	DIG5	IO	输出	位码 5
	N14	DIG4	IO	输出	位码 4
	N13	DIG3	IO	输出	位码 3
	M12	DIG2	IO	输出	位码 2
	H13	DIG1	IO	输出	位码 1
	G12	DIG0	IO	输出	位码 0
	P11	A	IO	输出	段码 0
	N12	B	IO	输出	段码 1
	L14	C	IO	输出	段码 2
	K13	D	IO	输出	段码 3
	K12	E	IO	输出	段码 4
	P13	F	IO	输出	段码 5
	M14	G	IO	输出	段码 6
	L13	DP	IO	输出	段码 7

创建一个新的仿真文件（此步可跳过）

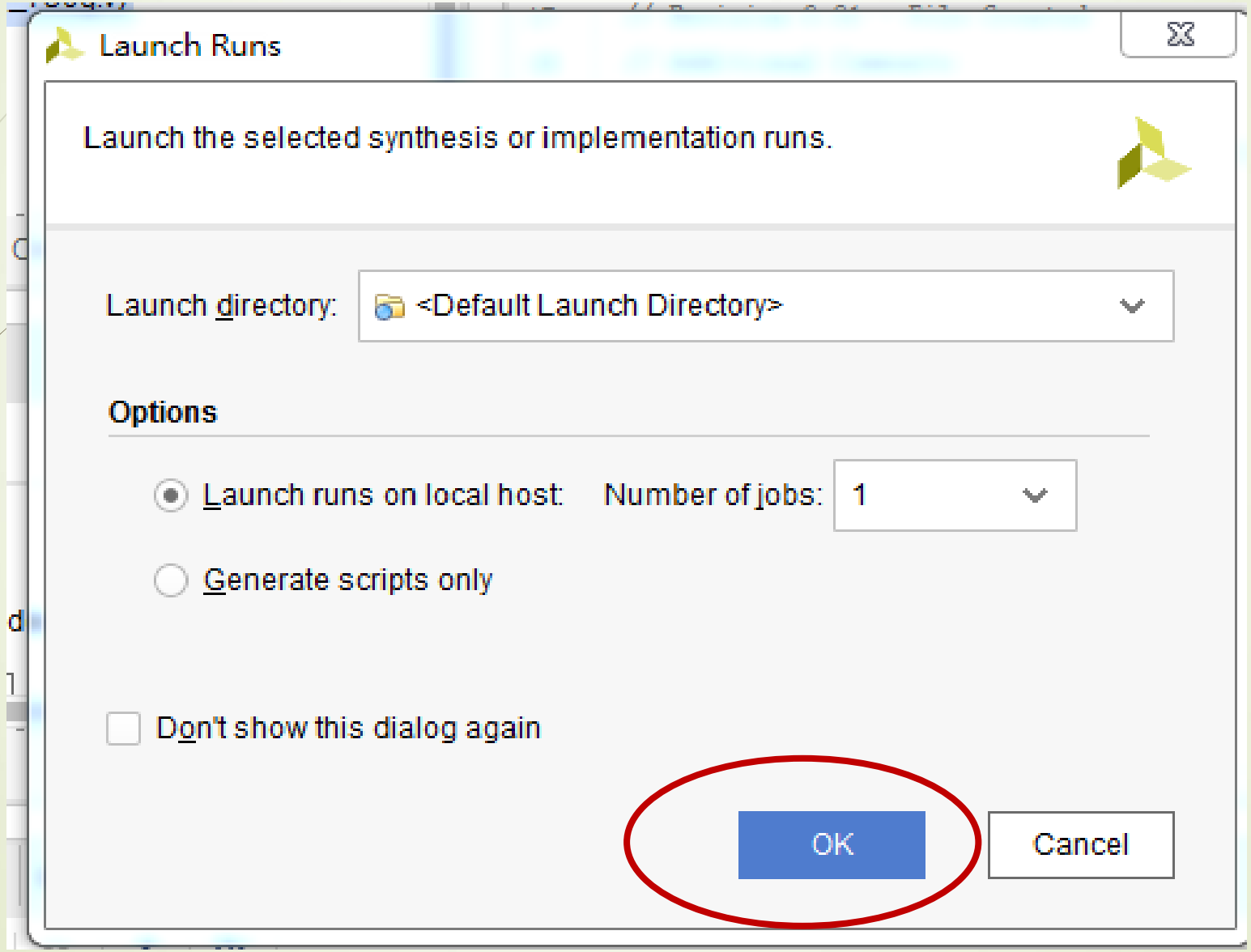


此时只能进行行为仿真

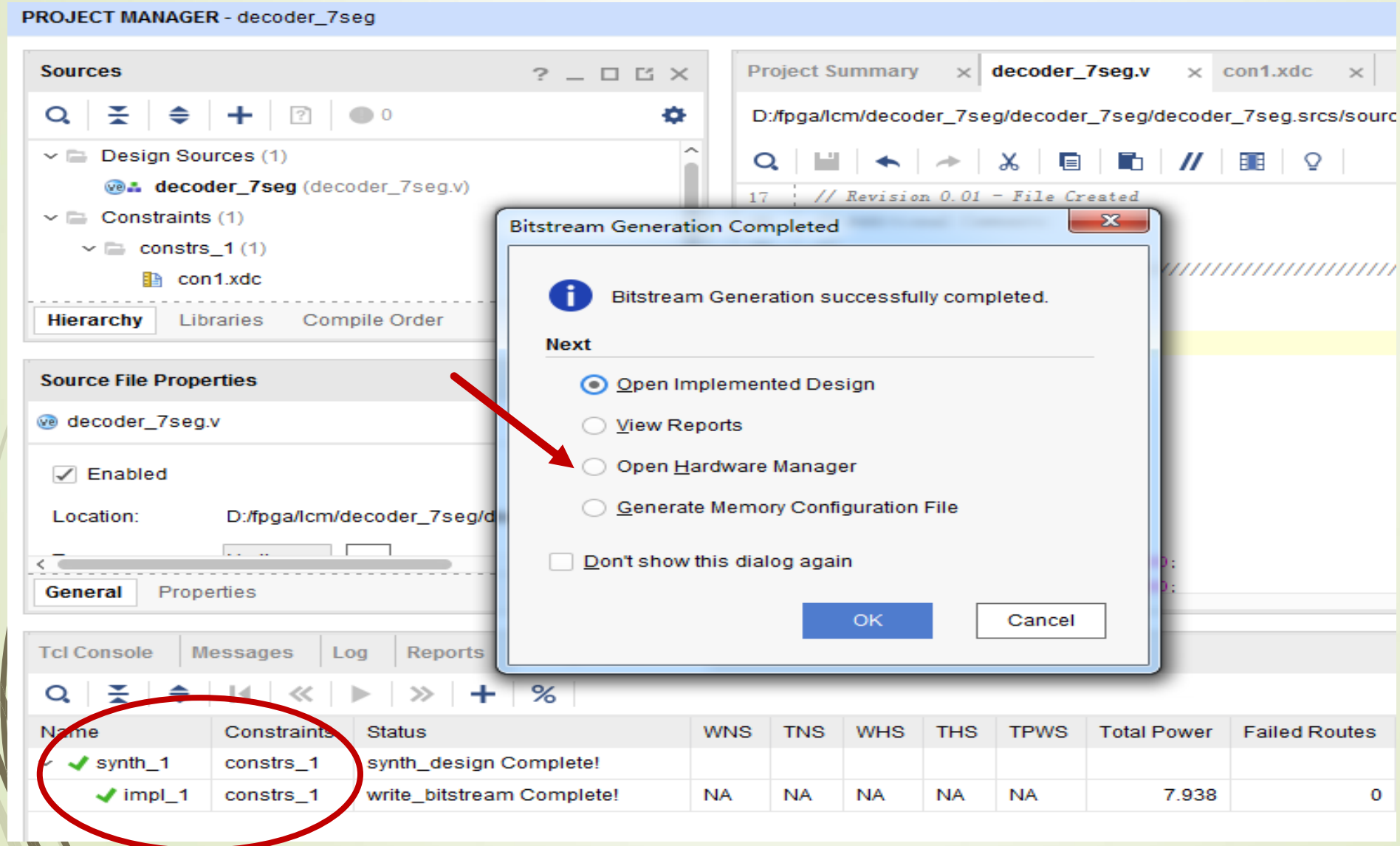
选择生成比特流文件（同时自动开始执行前面的综合和实现步骤）



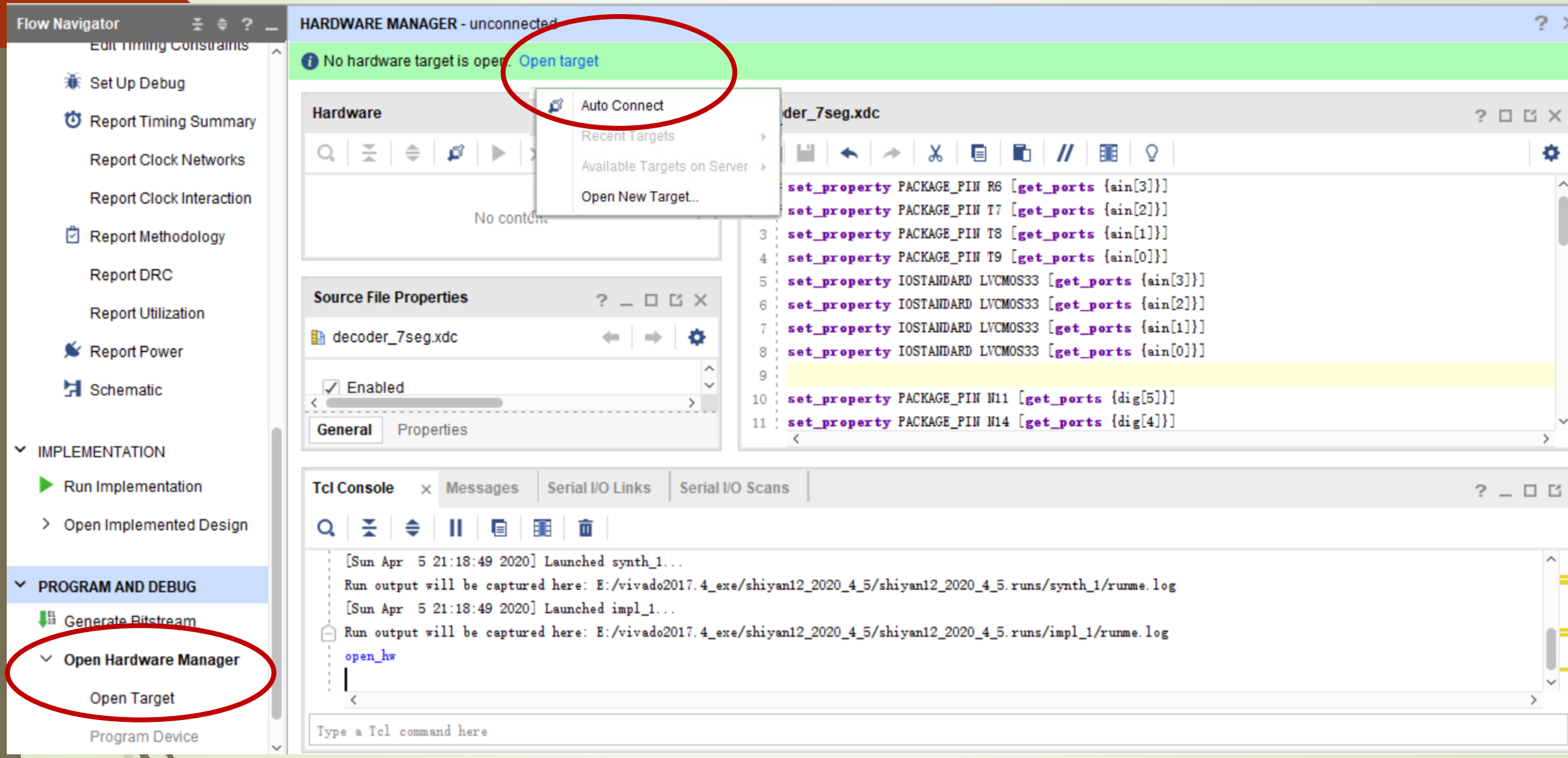
执行生成比特流文件


Launch Runs
Launch the selected synthesis or implementation runs.
Launch directory:
<Default Launch Directory>
Options
Launch runs on local host:
Number of jobs:
1
Generate scripts only
Don't show this dialog again
OK
Cancel

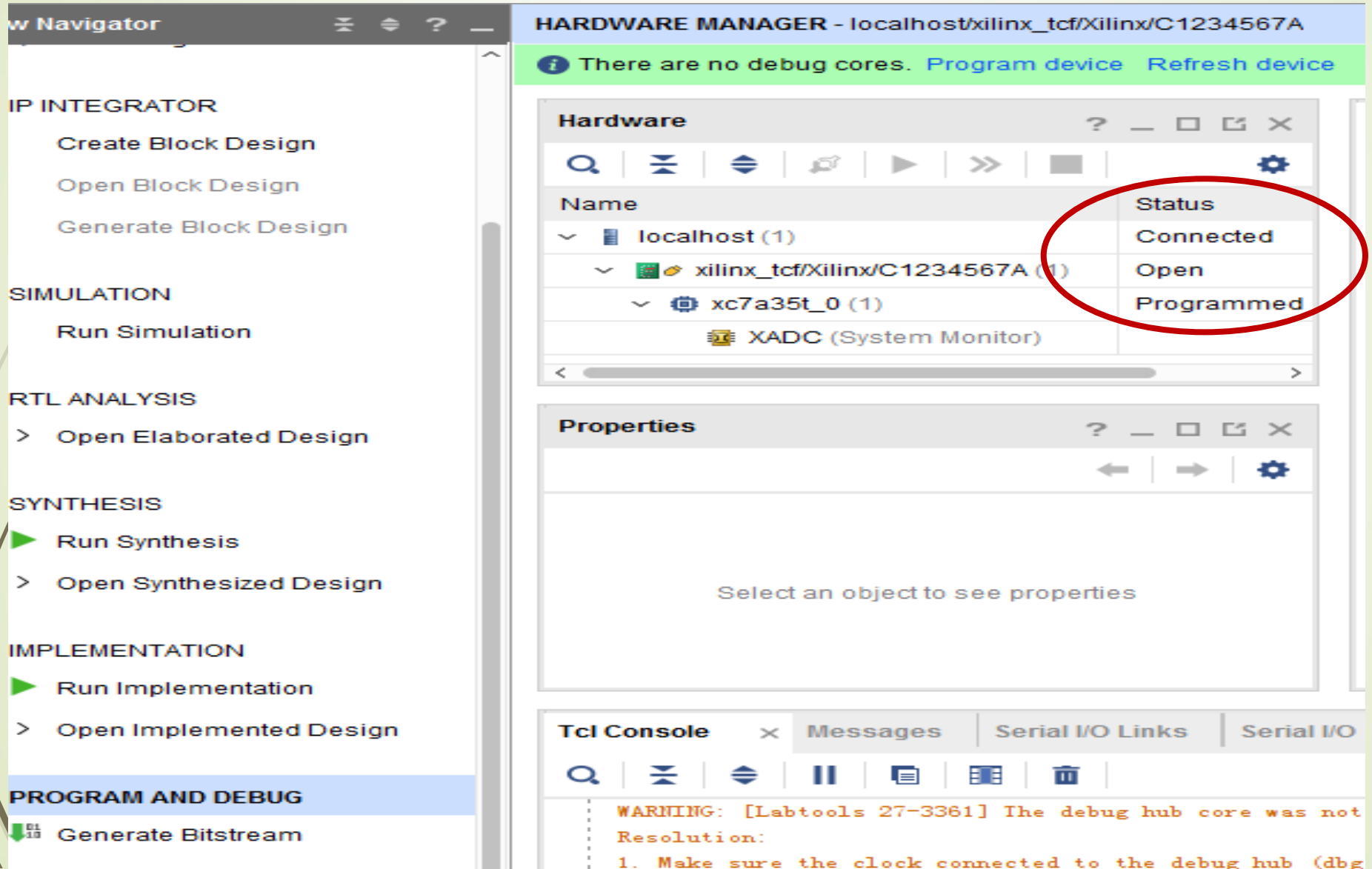
已生成比特流文件，选打开硬件管理器



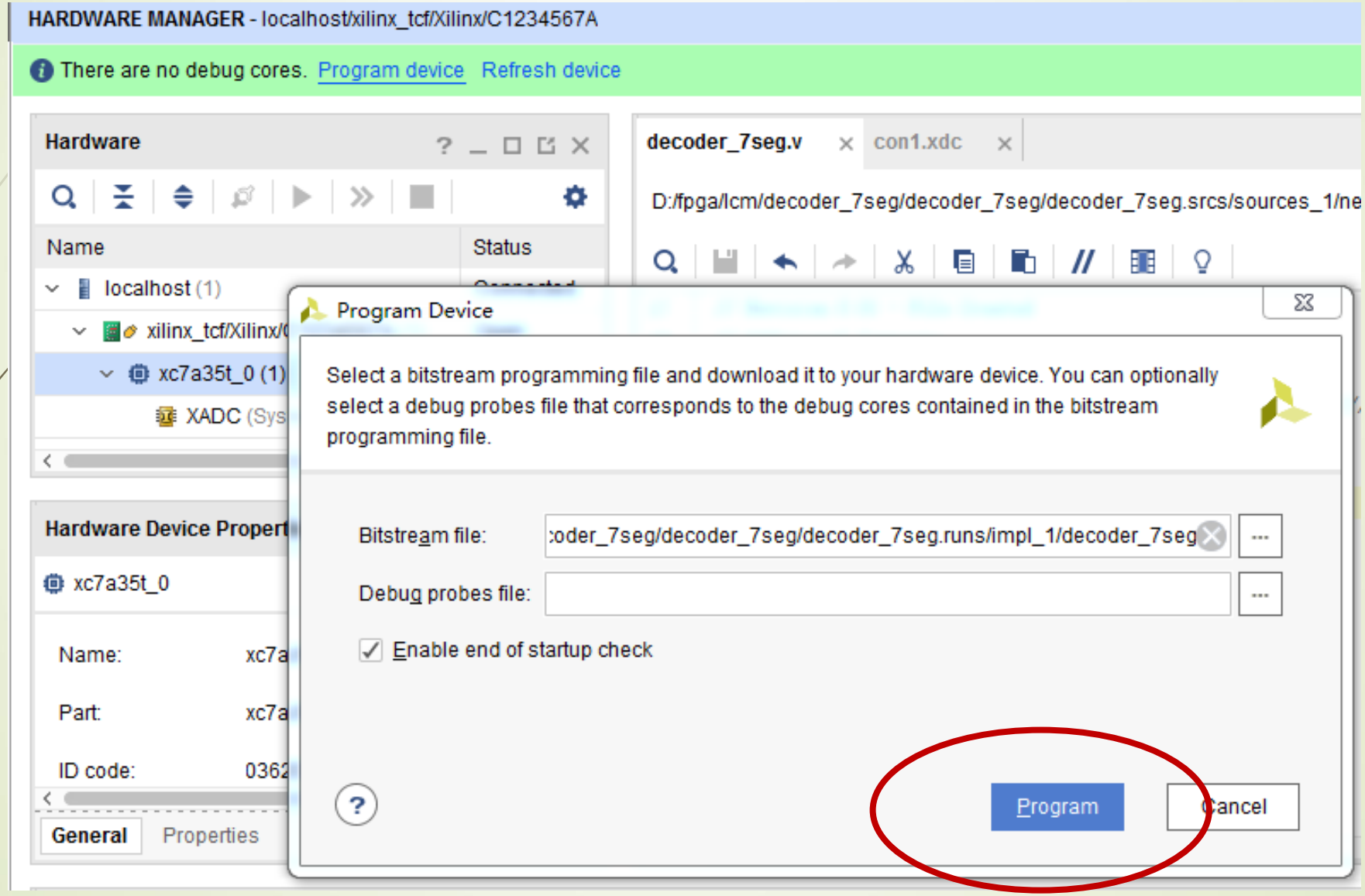
将硬件插入USB口，打开目标器件，选自动连接到硬件



已连接上FPGA硬件


w Navigator
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
HARDWARE MANAGER - localhost/xilinx_tcf/Xilinx/C1234567A
There are no debug cores. Program device Refresh device
Hardware
Name
Status
localhost (1)
Connected
xilinx_tcf/Xilinx/C1234567A (1)
Open
xc7a35t_0 (1)
Programmed
XADC (System Monitor)
Properties
Select an object to see properties
Tcl Console
Messages
Serial I/O Links
Serial I/O
WARNING: [Labtools 27-3361] The debug hub core was not
Resolution:
1. Make sure the clock connected to the debug hub (dbg

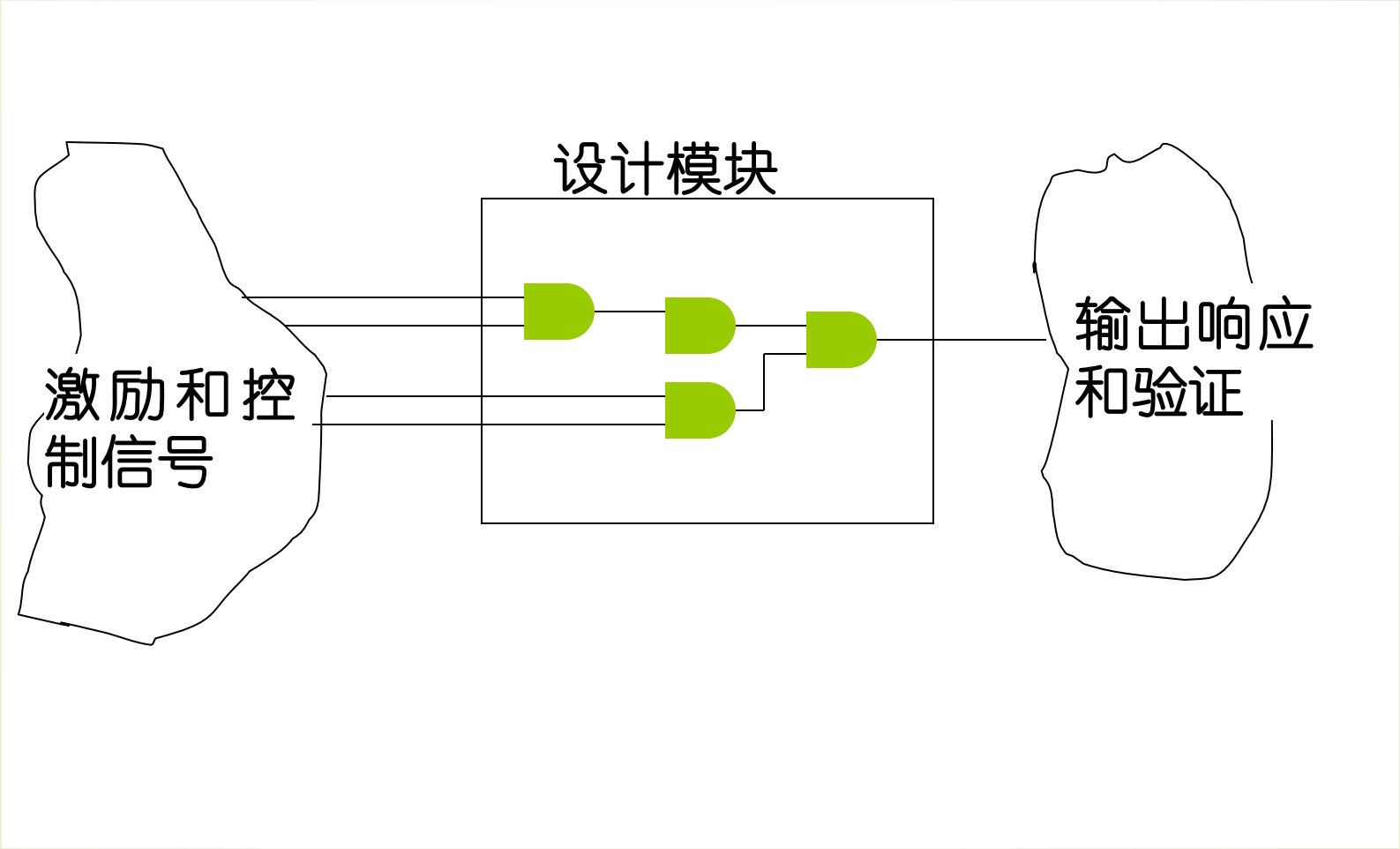
下载比特流文件到FPGA底板



实验十二
数码管显示译码器的实现

（三）七段显示译码器的行为仿真主要流程

仿真文件



仿真文件常见的形式

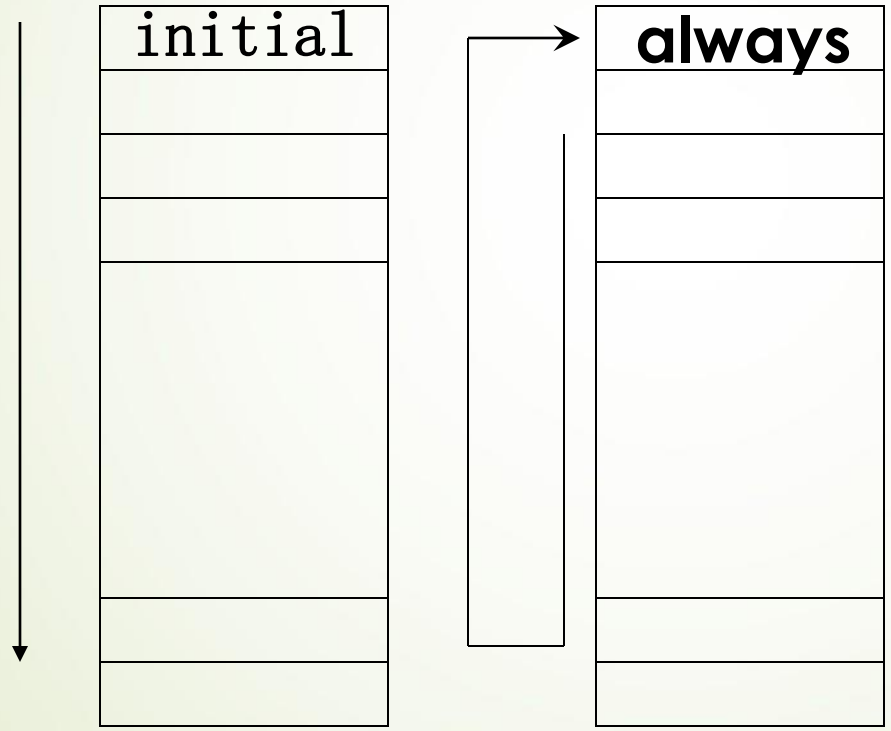
```
module t;
reg …;          //输入/输出变量类型定义
wire…;        //输入/输出变量类型定义
initial begin …;  …;  …;  end … …//产生激励信号
always #delay  begin …;  end  … …//产生激励信号

Testedmd      m(.in1(ina),      .in2(inb),      .out1(outa),
    .out2(outb) );
//设计模块的实例引用

initial begin ….; ….; ….    end      //记录输出和响应
endmodule
```

设计模块名

实例名

仿真文件

仿真文件中常用的过程块：



所有的过程块都在0时刻同时启动；它们是并行的，在模块中不分前后。

- initial块 只执行一次。
- always块 只要符合触发条件可以循环执行。

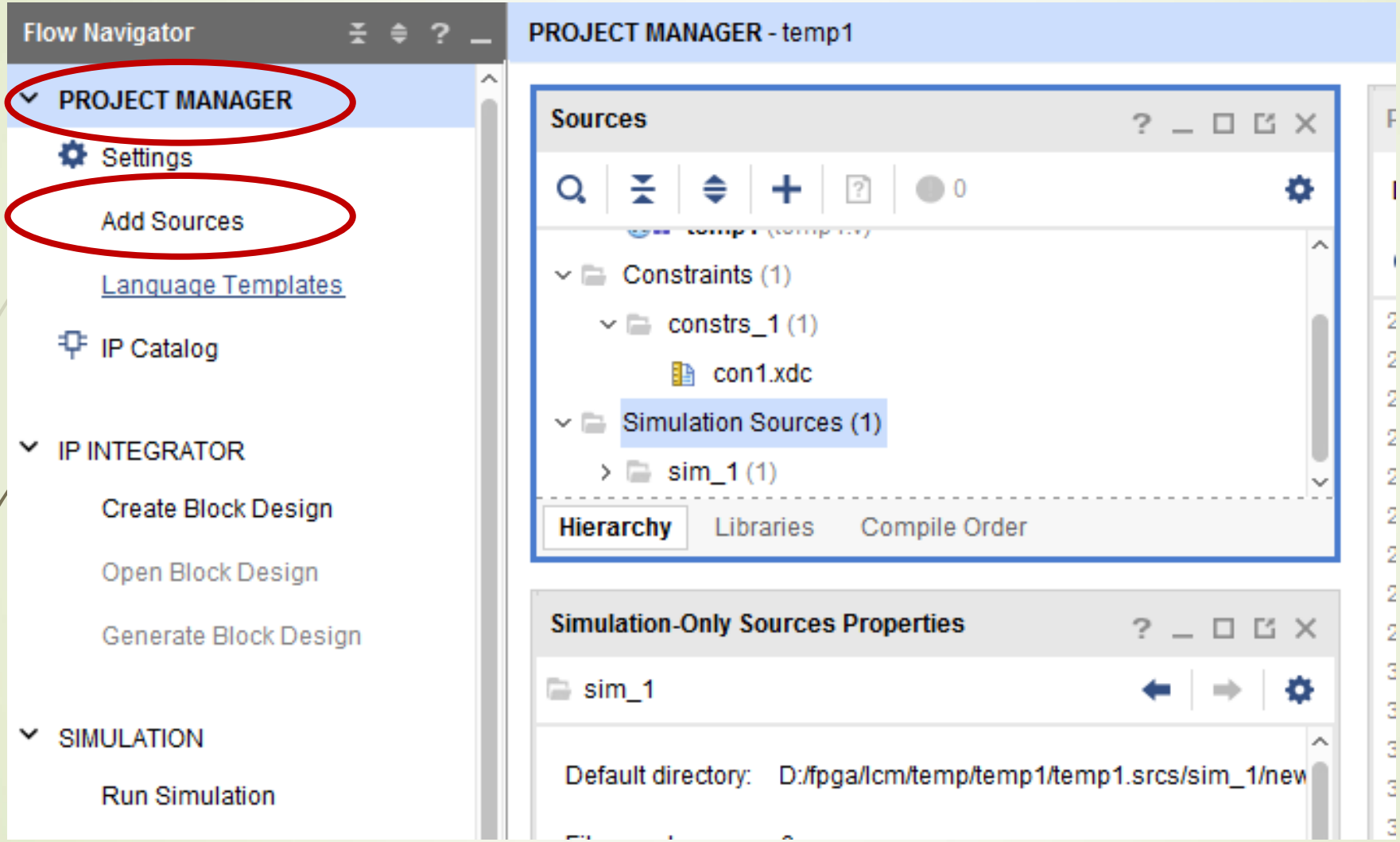
七段显示译码器仿真文件

```
module sim_1;
    reg  [3:0] ain;
    wire [7:0] seg;
    wire [5:0] dig;

    decoder0_9  uut(ain,seg,dig);
    initial
       begin
        ain = 0;
       end
    always #10 ain = ain+1;
endmodule
```

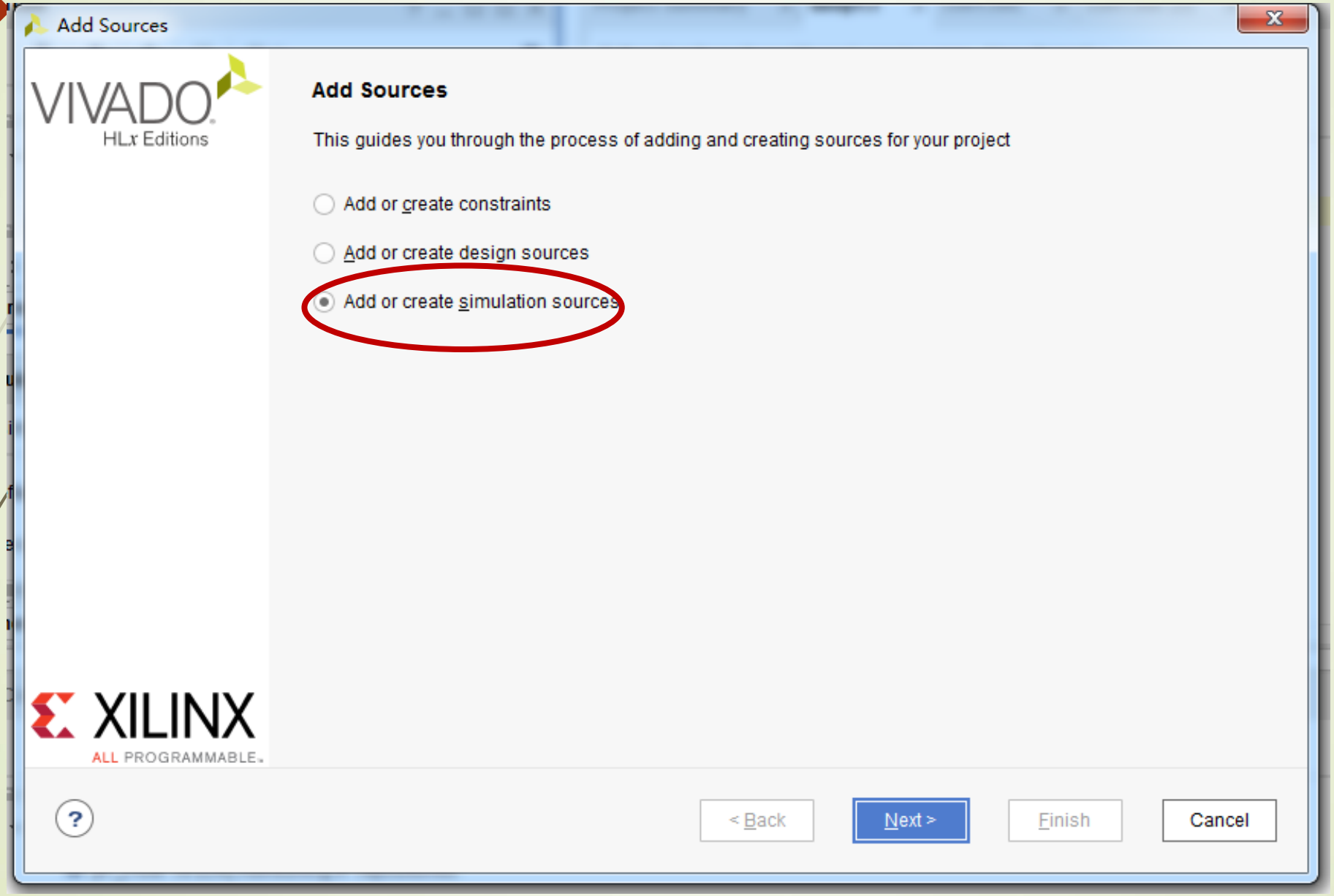
Unit under test的缩写

打开当前工程，点击创建文件

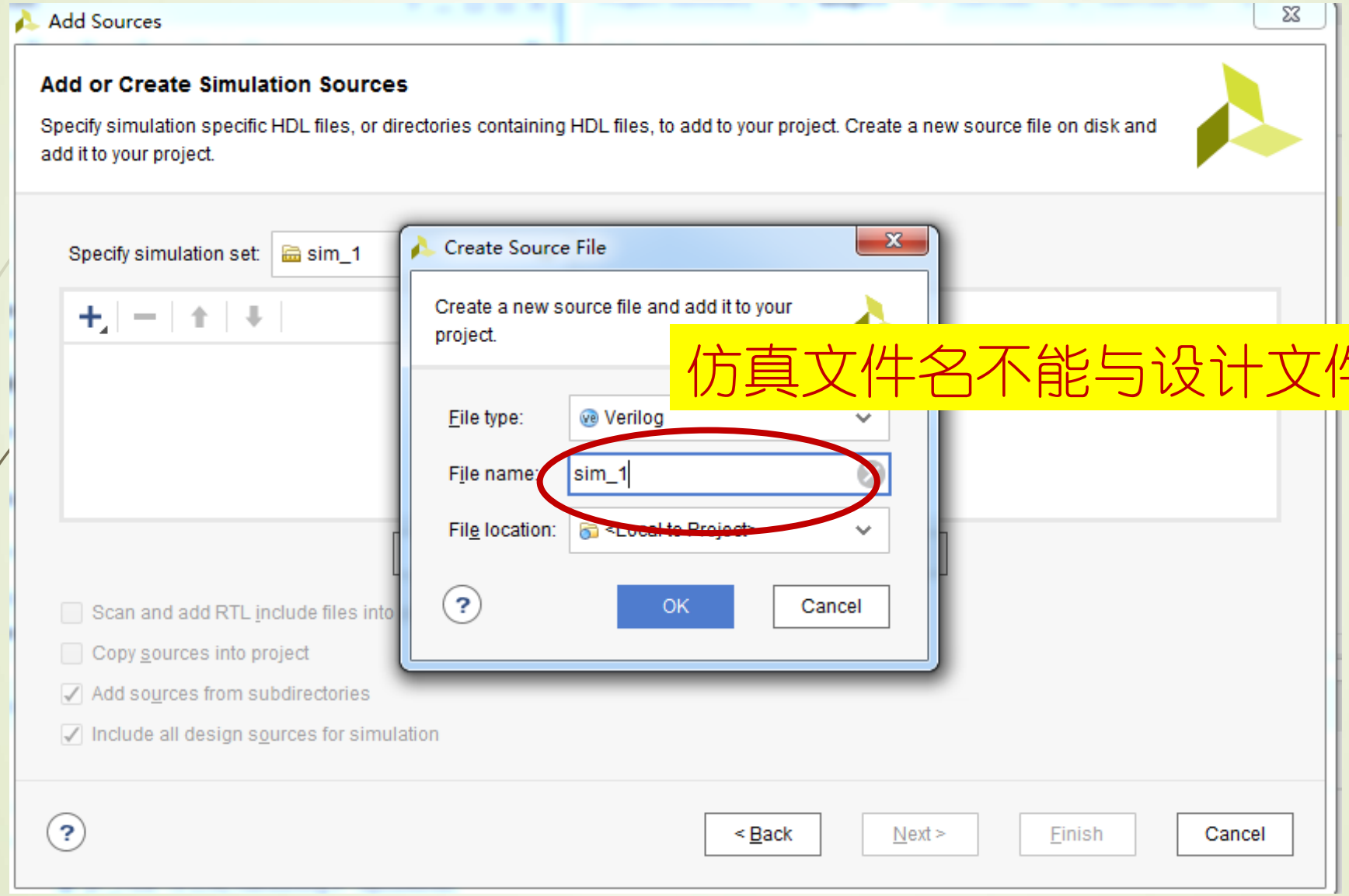
回到默认布局界面



创建一个新的仿真文件

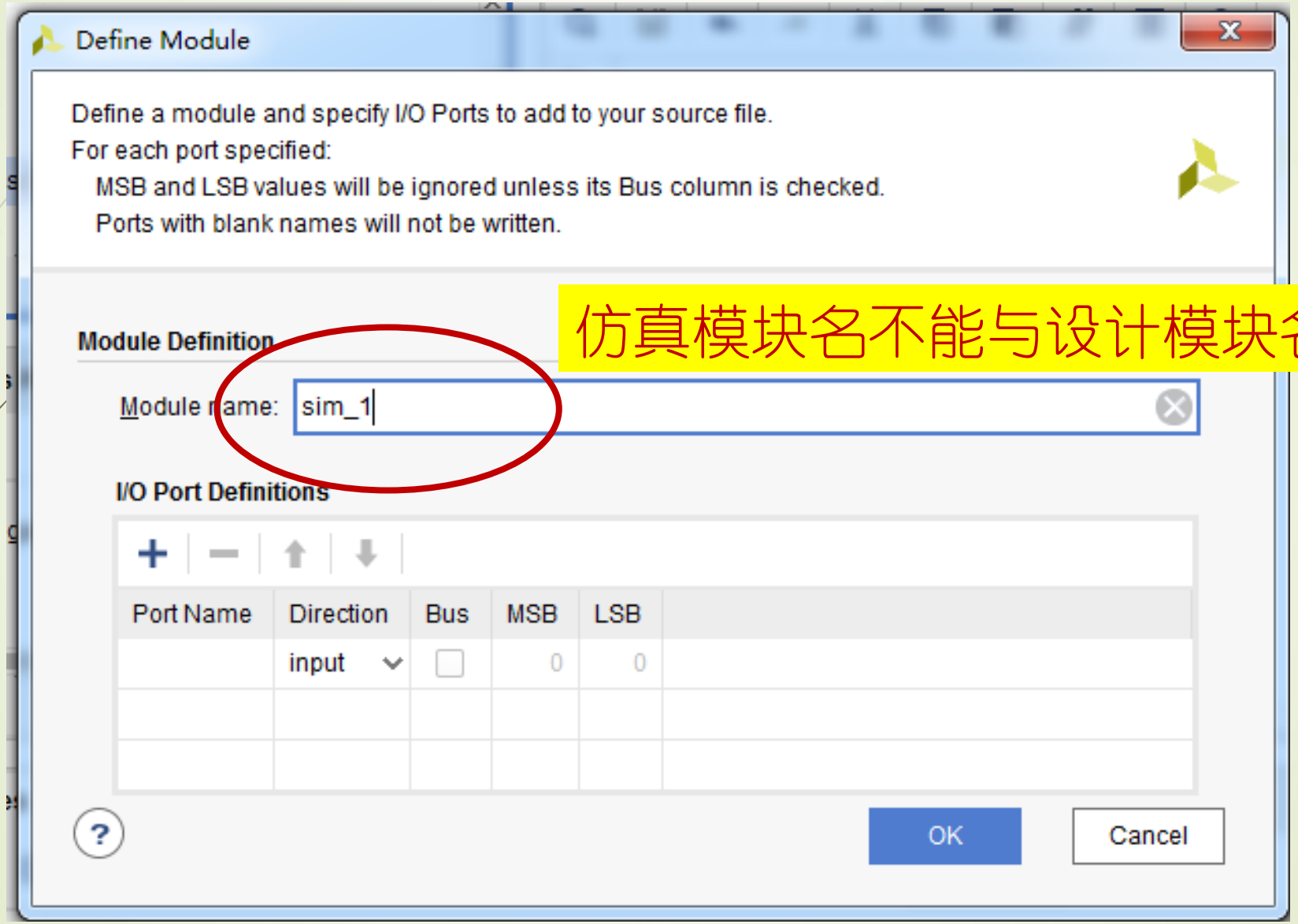

Add Sources
VIVADO
HLx Editions
Add Sources
This guides you through the process of adding and creating sources for your project
Add or create constraints
Add or create design sources
Add or create simulation sources
XILINX
ALL PROGRAMMABLE
?
< Back
Next >
Finish
Cancel

输入仿真文件名



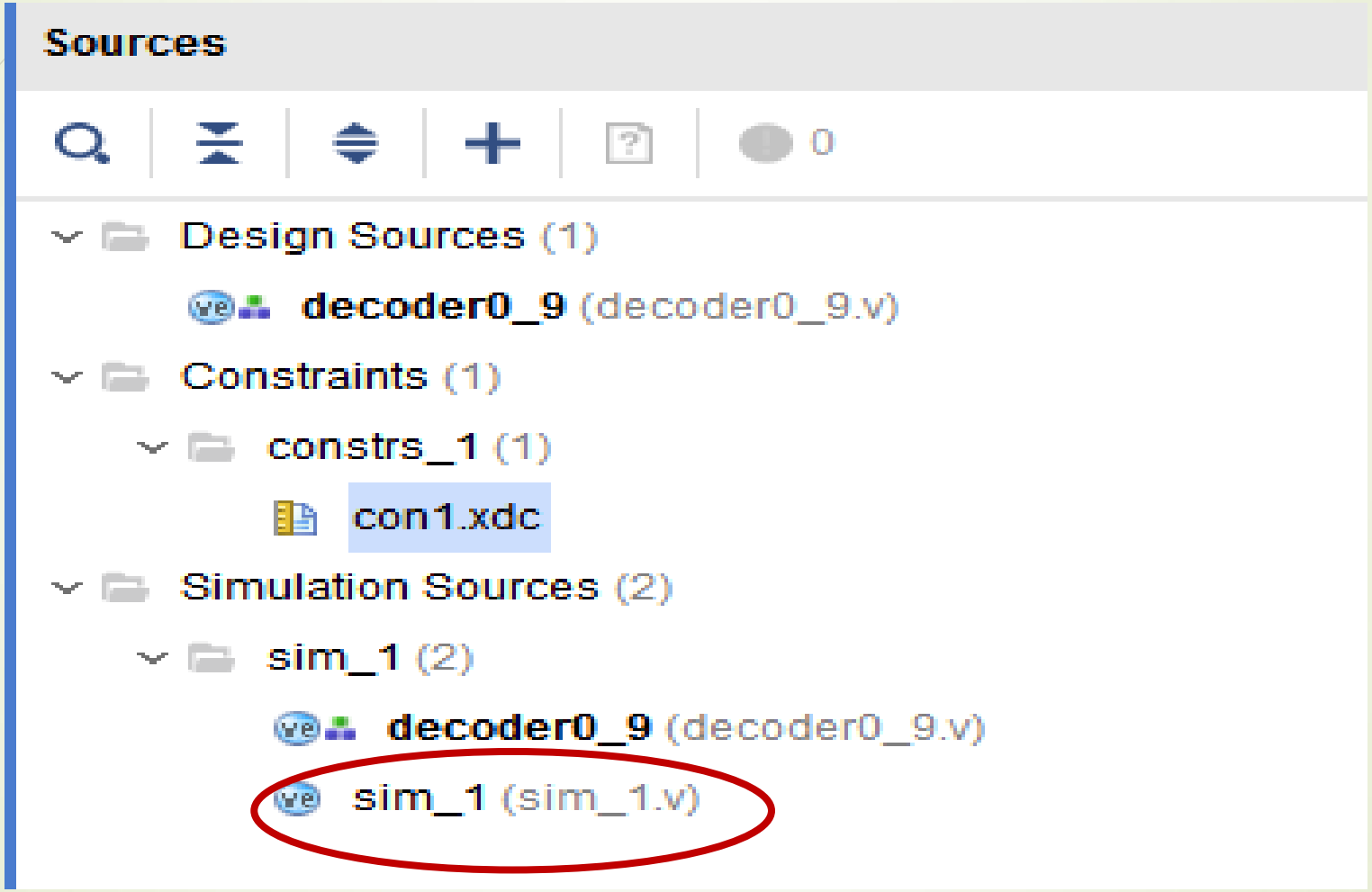
仿真文件名不能与设计文件名完全相同

输入仿真模块名

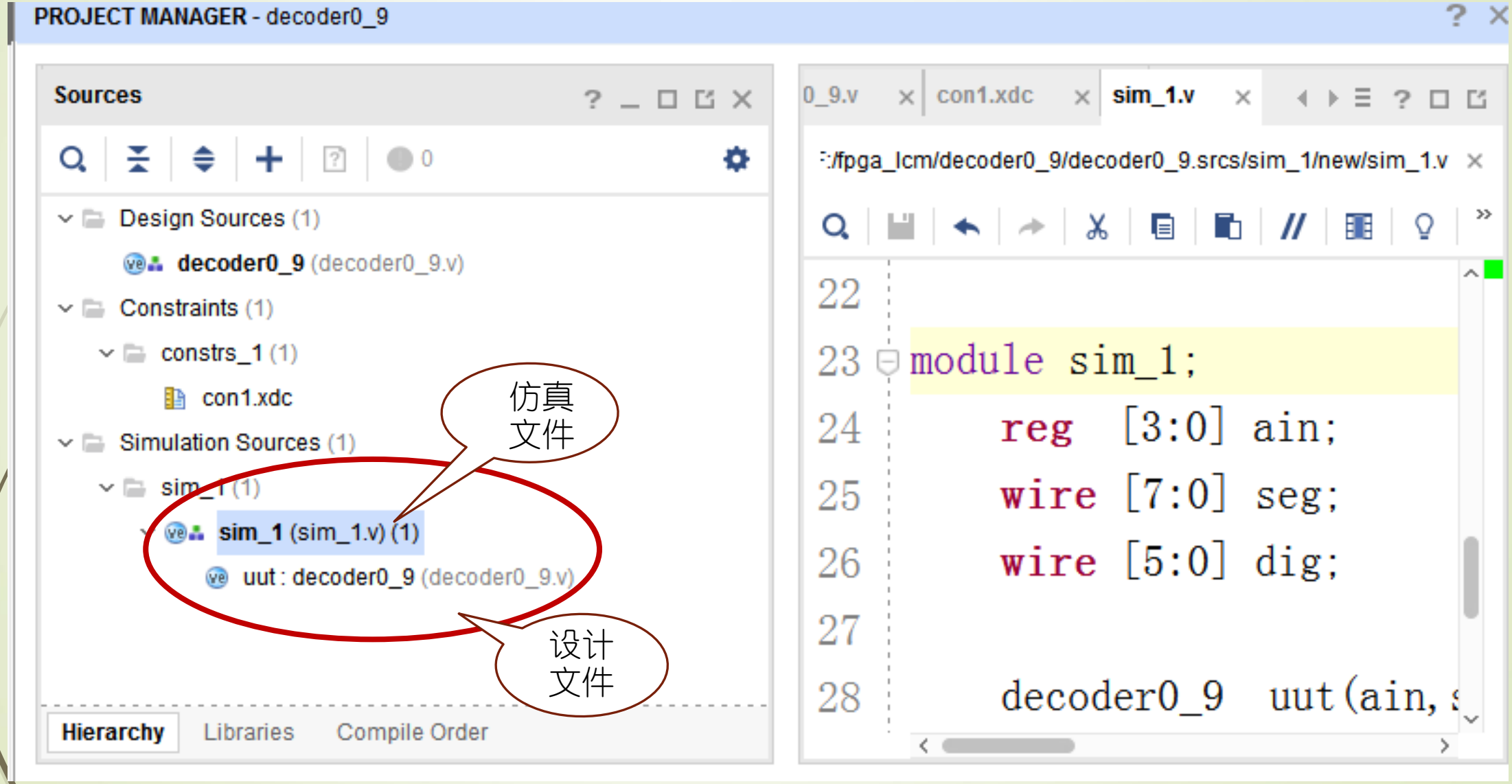


仿真模块名不能与设计模块名完全相同

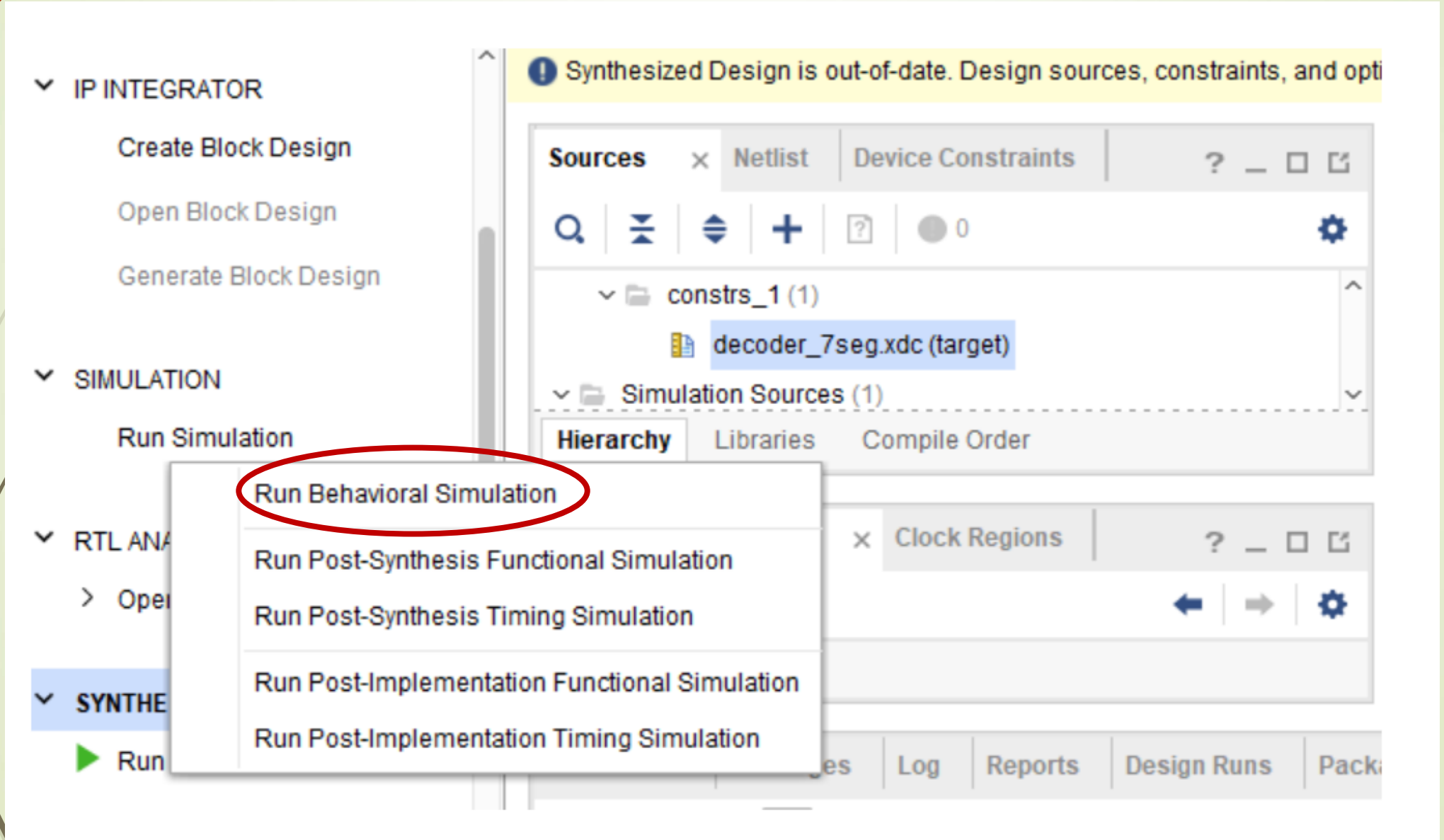
创建了仿真文件后的源窗口



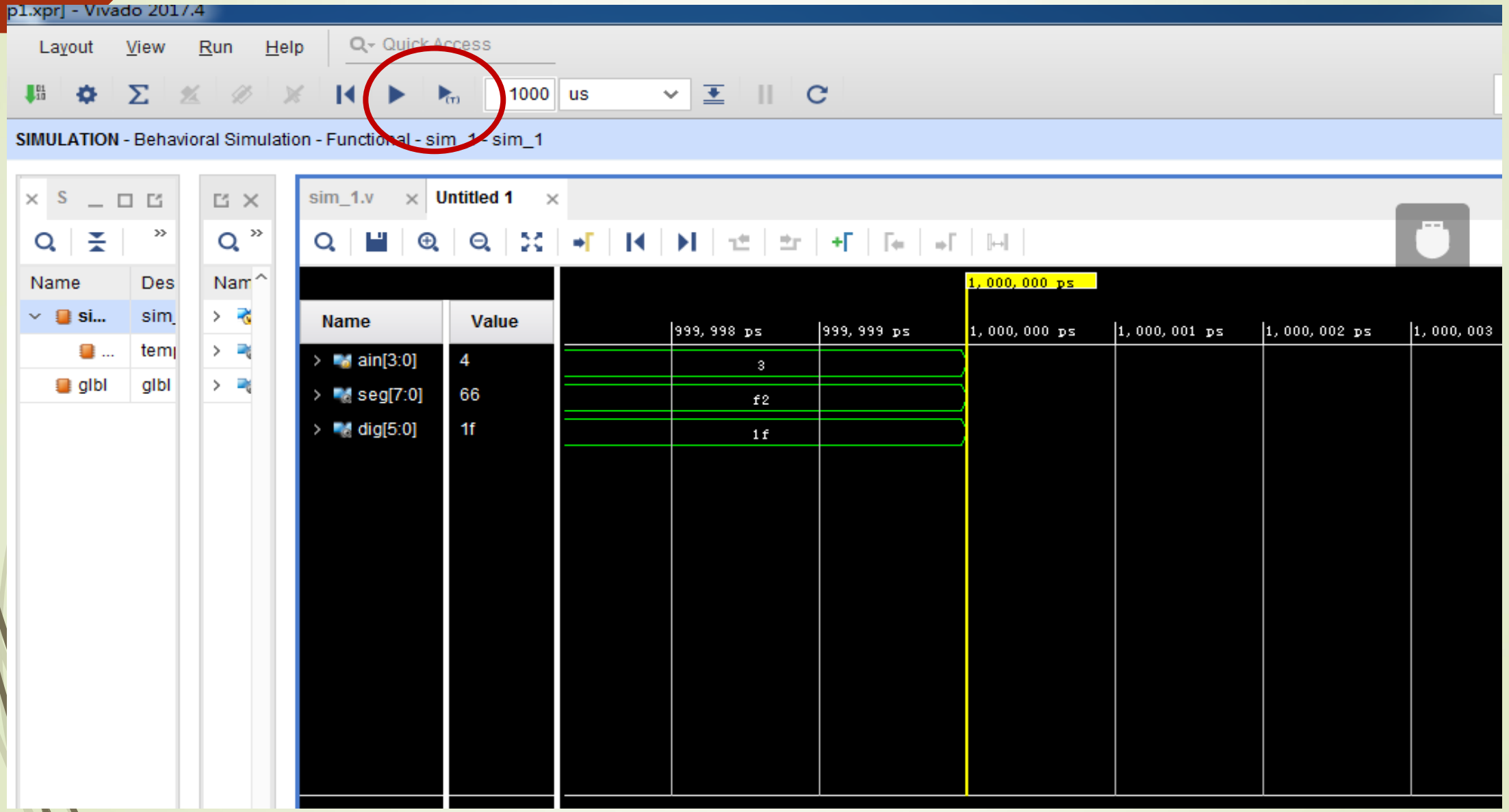
添加了仿真文件代码后的源窗口



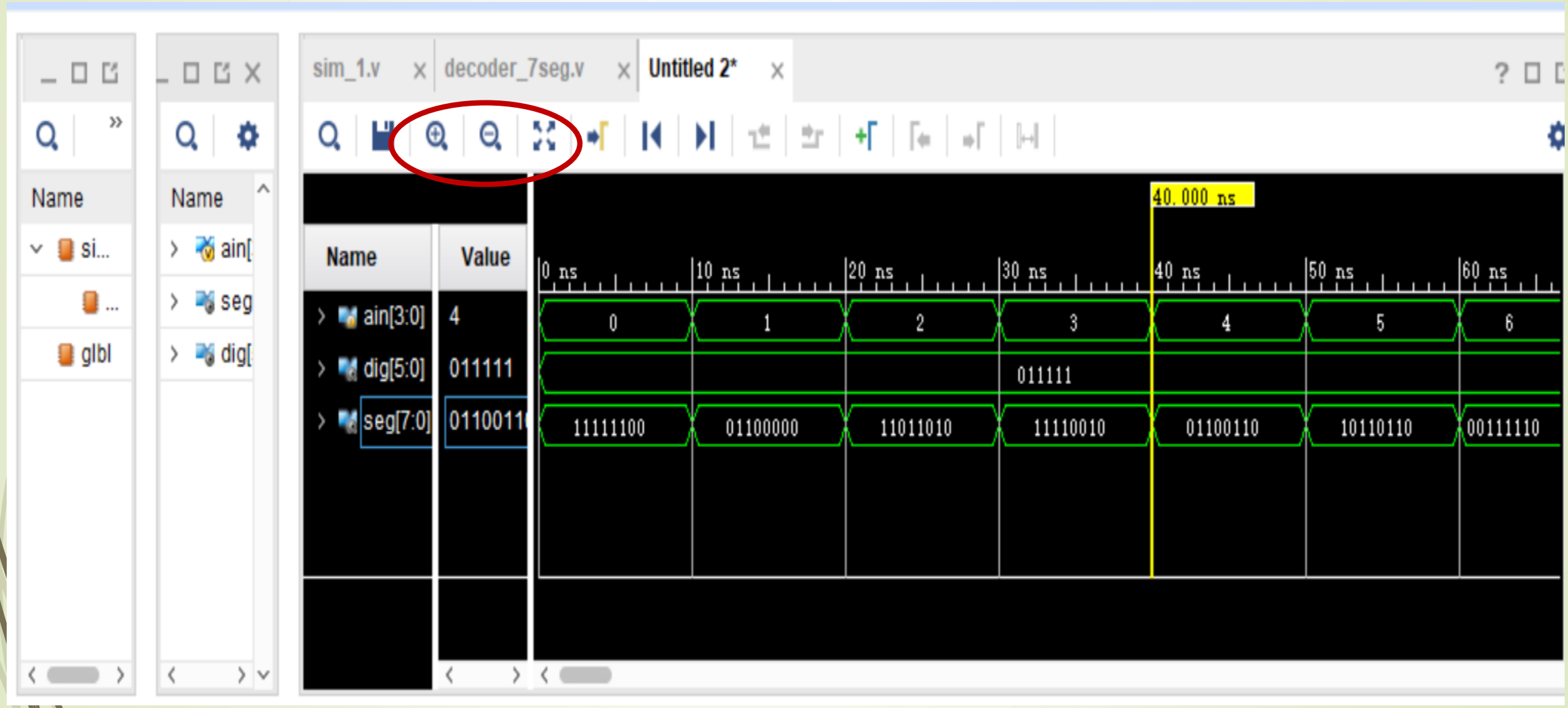
选择行为级仿真


IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANA
Oper
SYNTHE
Run
Run Behavioral Simulation
Run Post-Synthesis Functional Simulation
Run Post-Synthesis Timing Simulation
Run Post-Implementation Functional Simulation
Run Post-Implementation Timing Simulation
Synthesized Design is out-of-date. Design sources, constraints, and opti
Sources
Netlist
Device Constraints
constrs_1 (1)
decoder_7seg.xdc (target)
Simulation Sources (1)
Hierarchy
Libraries
Compile Order
Clock Regions
Log
Reports
Design Runs

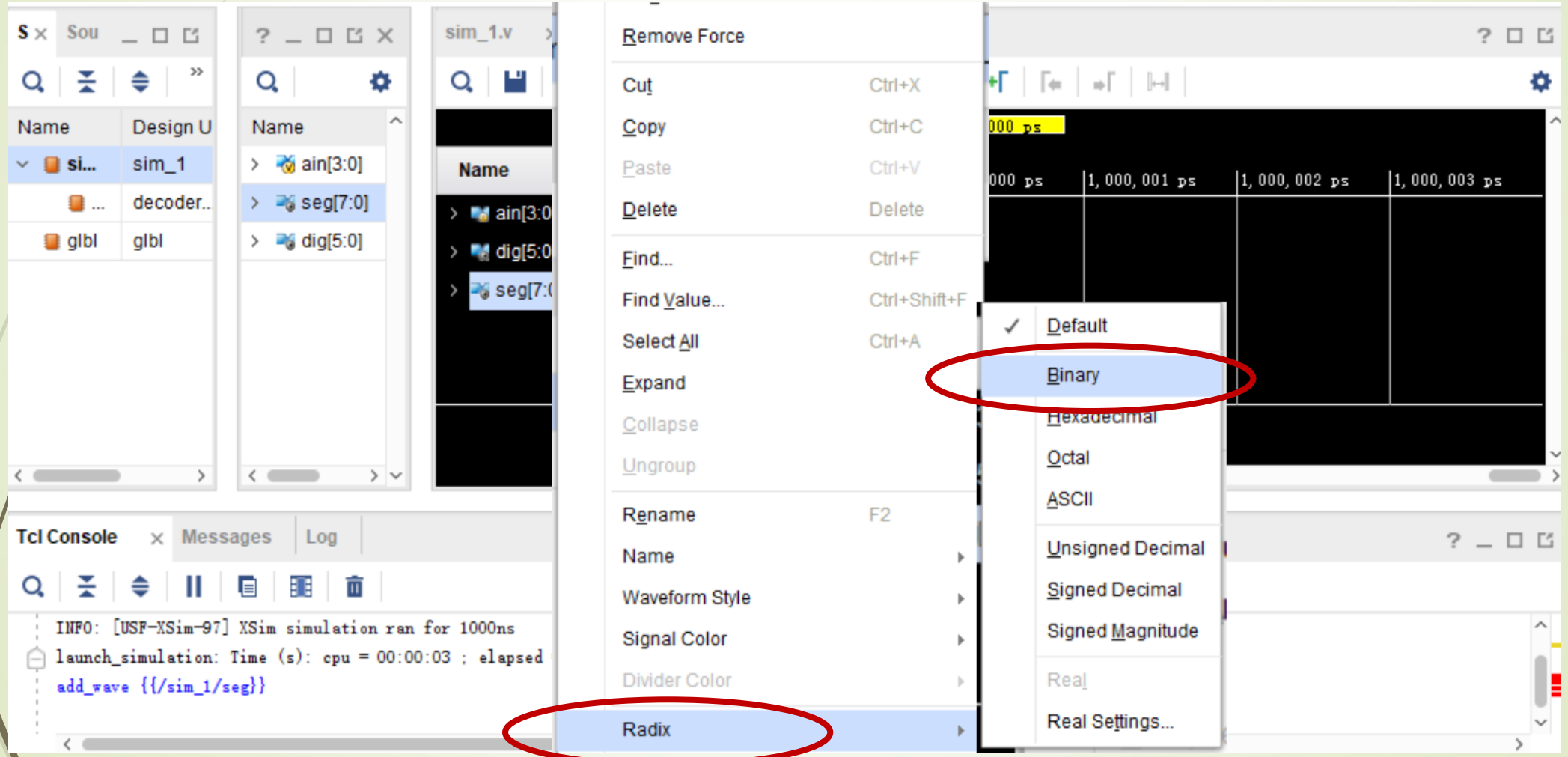
按三角符号开始仿真


Layout
View
Run
Help
Quick Access
1000
us
SIMULATION - Behavioral Simulation - Functional - sim_1 - sim_1
sim_1.v
Untitled 1
Name
Value
ain[3:0]
4
seg[7:0]
66
dig[5:0]
1f
1,000,000 ps
999,998 ps
999,999 ps
1,000,000 ps
1,000,001 ps
1,000,002 ps
1,000,003
3
f2
1f

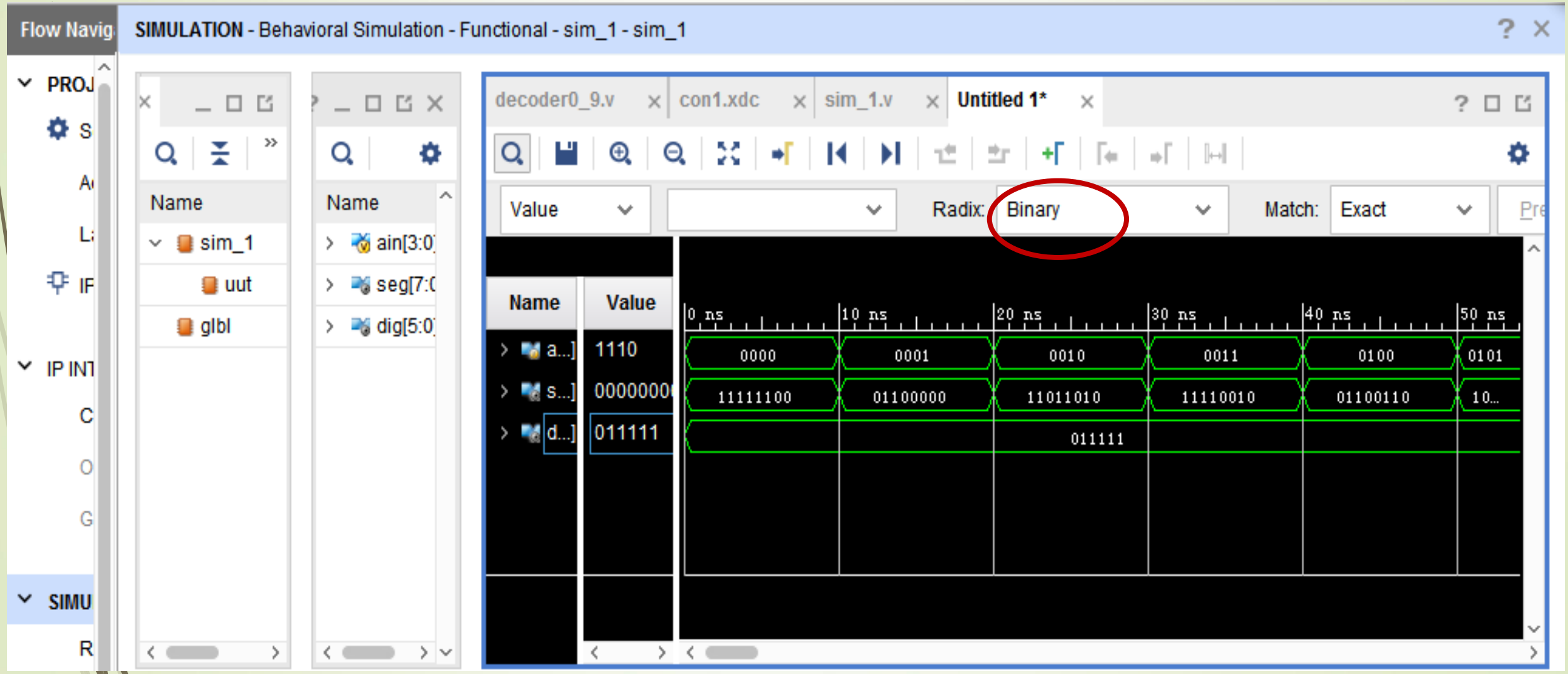
调整仿真波形的显示


sim_1.v
decoder_7seg.v
Untitled 2*
Name
si...
glbl
Name
ain[
seg
dig[
Name
Value
ain[3:0]
4
dig[5:0]
011111
seg[7:0]
0110011
40.000 ns
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
0
1
2
3
4
5
6
011111
11111100
01100000
11011010
11110010
01100110
10110110
00111110

选中变量，单击鼠标右键，改变变量的进制数（便于查看结果）



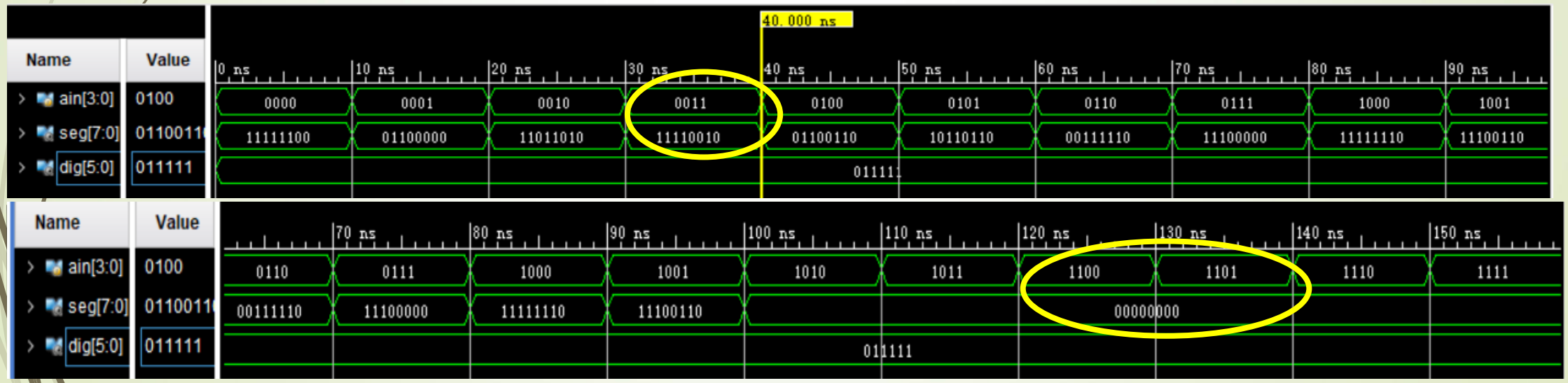
行为仿真波形图


Flow Navig
SIMULATION - Behavioral Simulation - Functional - sim_1 - sim_1
PROJ
IP INT
SIMU
Name
sim_1
uut
glbl
Name
ain[3:0]
seg[7:0]
dig[5:0]
decoder0_9.v
con1.xdc
sim_1.v
Untitled 1*
Value
Radix:
Binary
Match:
Exact
Name
Value
a...]
1110
s...]
0000000
d...]
011111
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
0000
0001
0010
0011
0100
0101
11111100
01100000
11011010
11110010
01100110
10...
011111

```
always@(ain)
        case(ain)
        4'b0000: seg = 8'b11111100;
        4'b0001: seg = 8'b01100000;
        4'b0010: seg = 8'b11011010;
        4'b0011: seg = 8'b11110010;
        4'b0100: seg = 8'b01100110;
        4'b0101: seg = 8'b10110110;
        4'b0110: seg = 8'b00111110;
        4'b0111: seg = 8'b11100000;
        4'b1000: seg = 8'b11111110;
        4'b1001: seg = 8'b11100110;
          default: seg = 8'b00000000;
        endcase
```

将仿真波形里的输入输出值与设计文件里的对应值相比较，验证设计结果是否正确。

将下面的仿真波形和上面的设计文件相对照



实验十二
数码管显示译码器的实现

（四）实验内容

实验内容

1、用Verilog HDL语言完成8421BCD码译码显示电路设计。在最左边的1位七段数码管显示数字0~9，以开发板上拨动开关为输入、数码管为输出，完成下载和仿真测试。查看RTL分析原理图。

2、用Verilog HDL语言完成4位二进制译码显示电路设计。在最右边的1位数码管上显示数字0~F（0、1、2.....9、A、b、c、d、E、F），完成下载和仿真测试。查看RTL分析原理图。