

# 实验1-基本组合、时序逻辑电路实验 (第2次课)

2025.10



哈尔滨工程大学计算机实验教学中心

# 基本组合、时序逻辑电路实验

## 实验目的

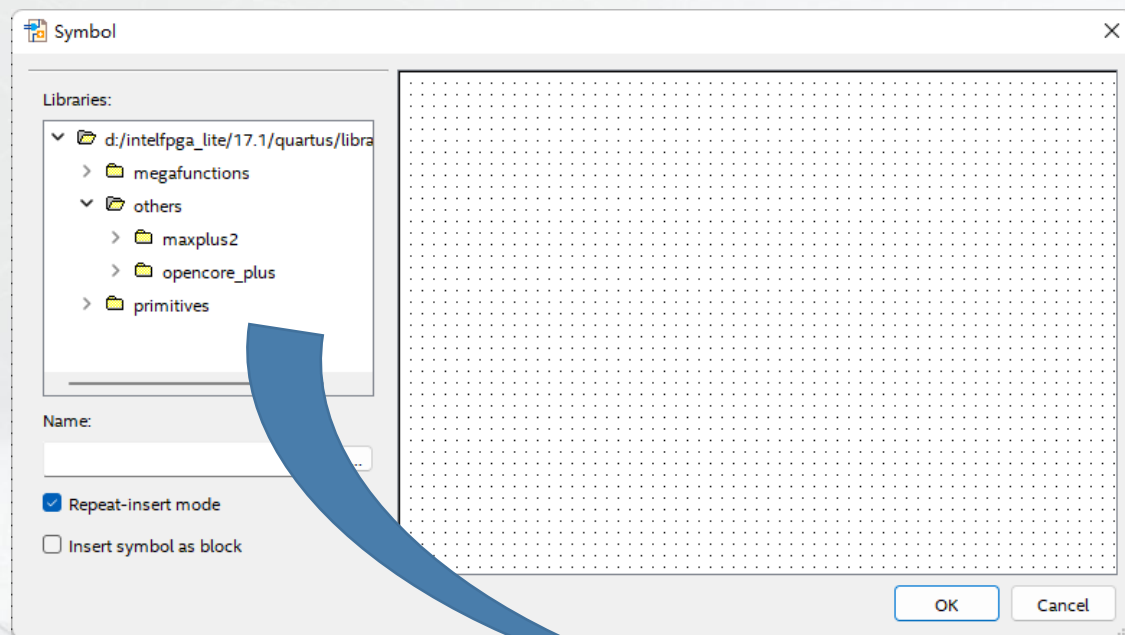
- 1.熟悉Quartus Prime开发环境和设计开发流程。
- 2.掌握利用原理图设计输入。
- 3.掌握Intel 宏功能块及IP核的使用方法。

## 实验内容

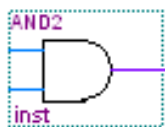
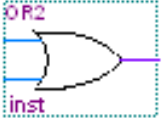
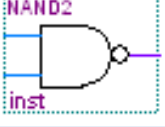
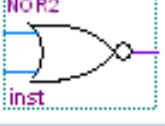
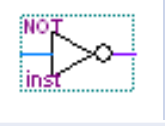
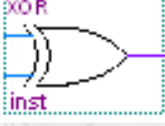
利用Quartus Prime元器件库中的基本单元，设计2-4译码器、计数器和8位数据寄存器。最后在实验台上进行硬件测试，验证电路的功能。

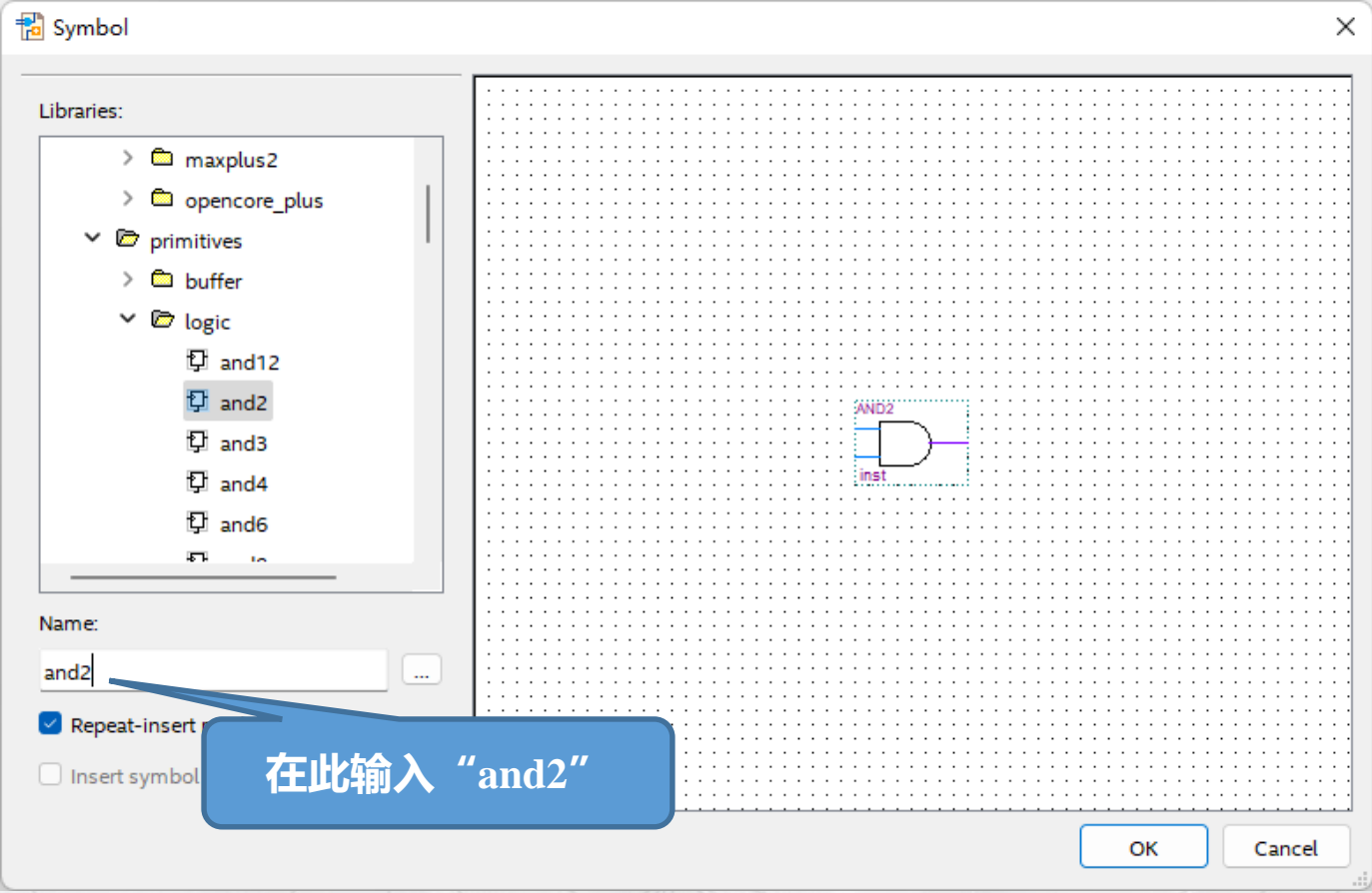
# 原理图输入法

Quartus Prime不仅提供了基本逻辑元件库(如与非门、反向器、D触发器等)、宏功能元件、Max+plusII库(包含了几乎所有74系列的器件)和基本逻辑元件库，还提供了参数可设置的宏功能块LPM库，利用框图编辑器，可以对宏功能模块进行例化。

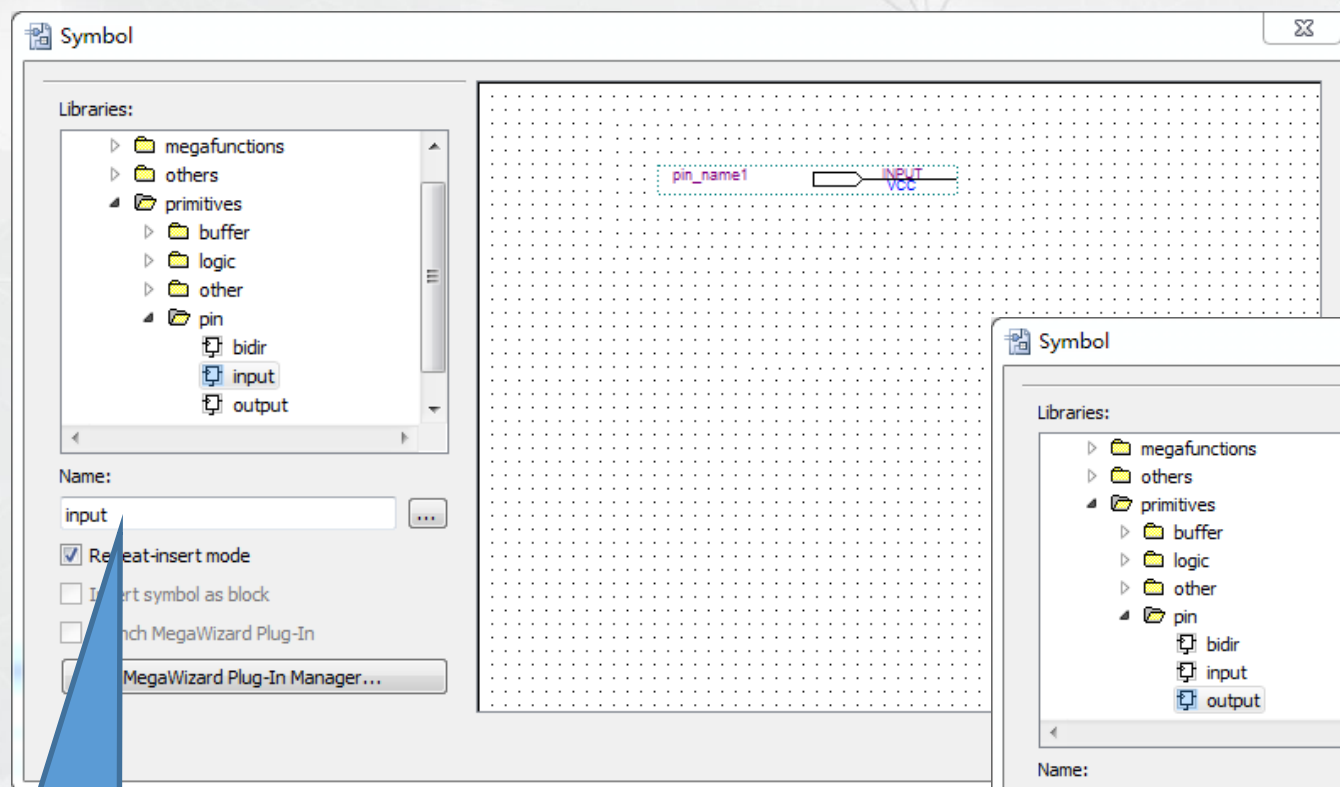


# 元器件库中的门电路

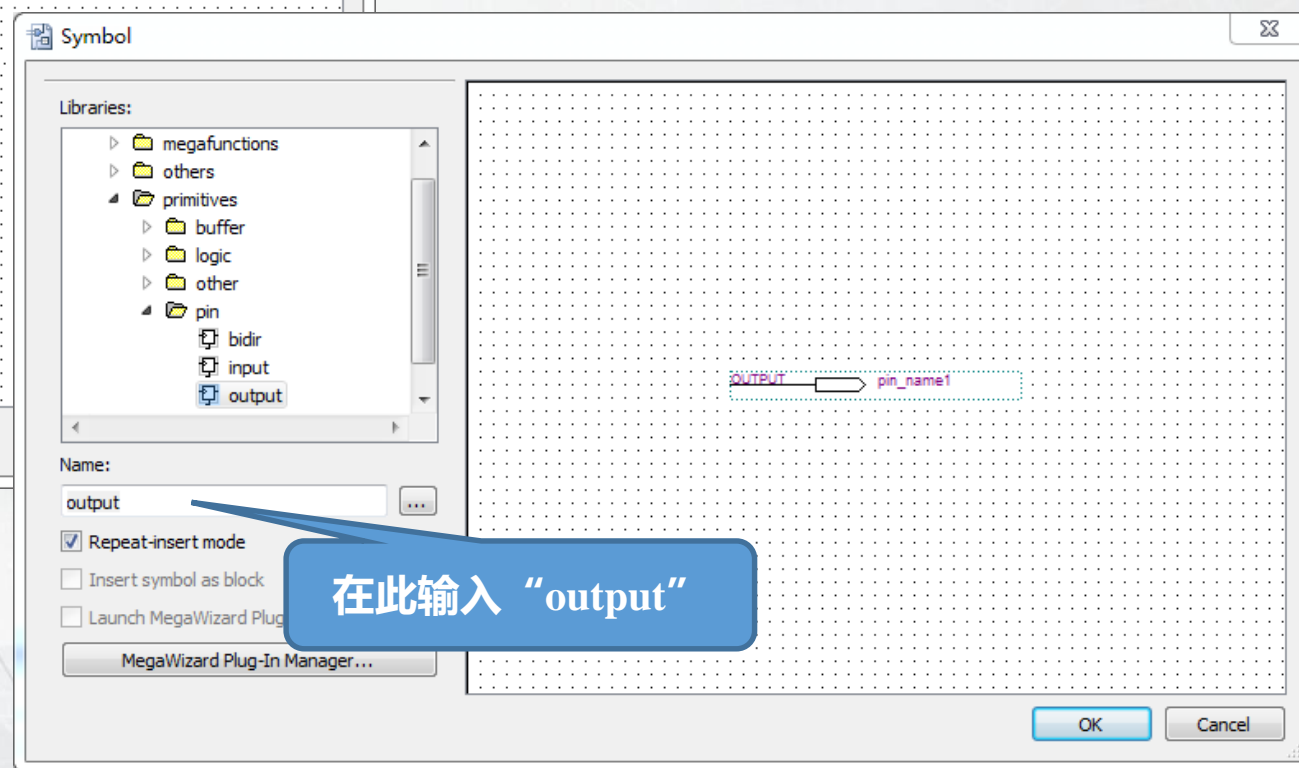
门电路	元器件名称	Intel元器件符号
二输入与门	and2	
二输入或门	or2	
二输入与非门	nand2	
二输入或非门	nor2	
非门（反相器）	not	
二输入异或门	xor	



# 元器件库中的输入输出端口



在此输入 “input”



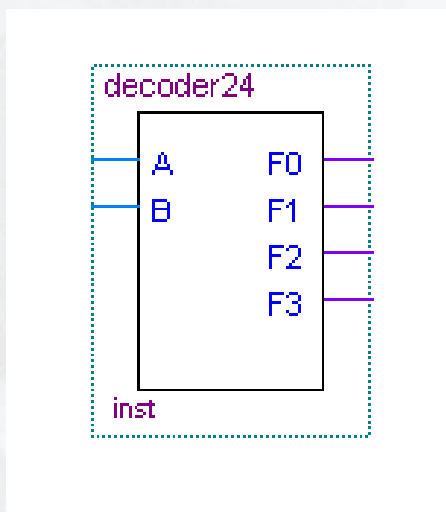
在此输入 “output”

## 2-4译码器基本原理

译码器(decoder)是一类多输入多输出组合逻辑电路器件。

2-4线两个输入端BA共有4种状态组合(00-11)，可译出4个输出信号F3-F0。

2-4译码器元件符号



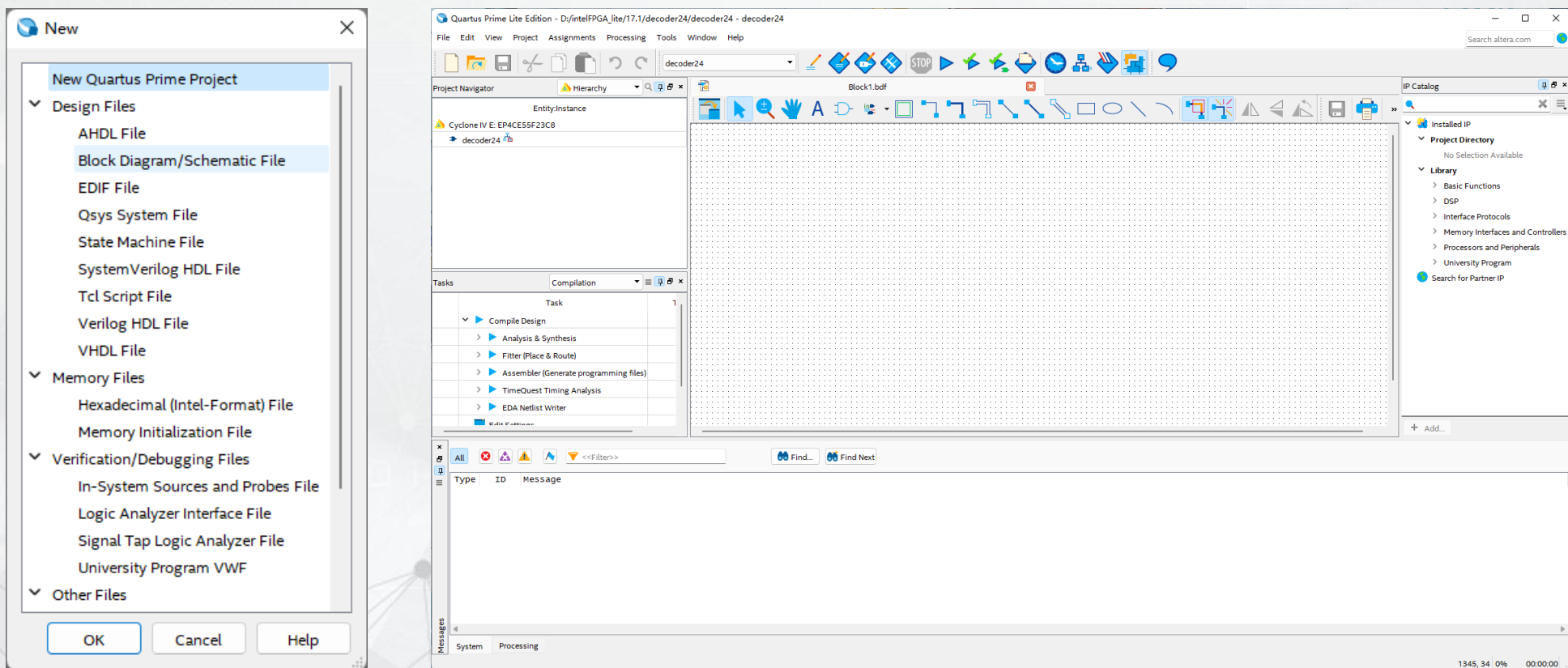
2-4译码器的真值表

输入		输出			
B	A	F3	F2	F1	F0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

# 框图编辑器(新建\*.bdf文件)

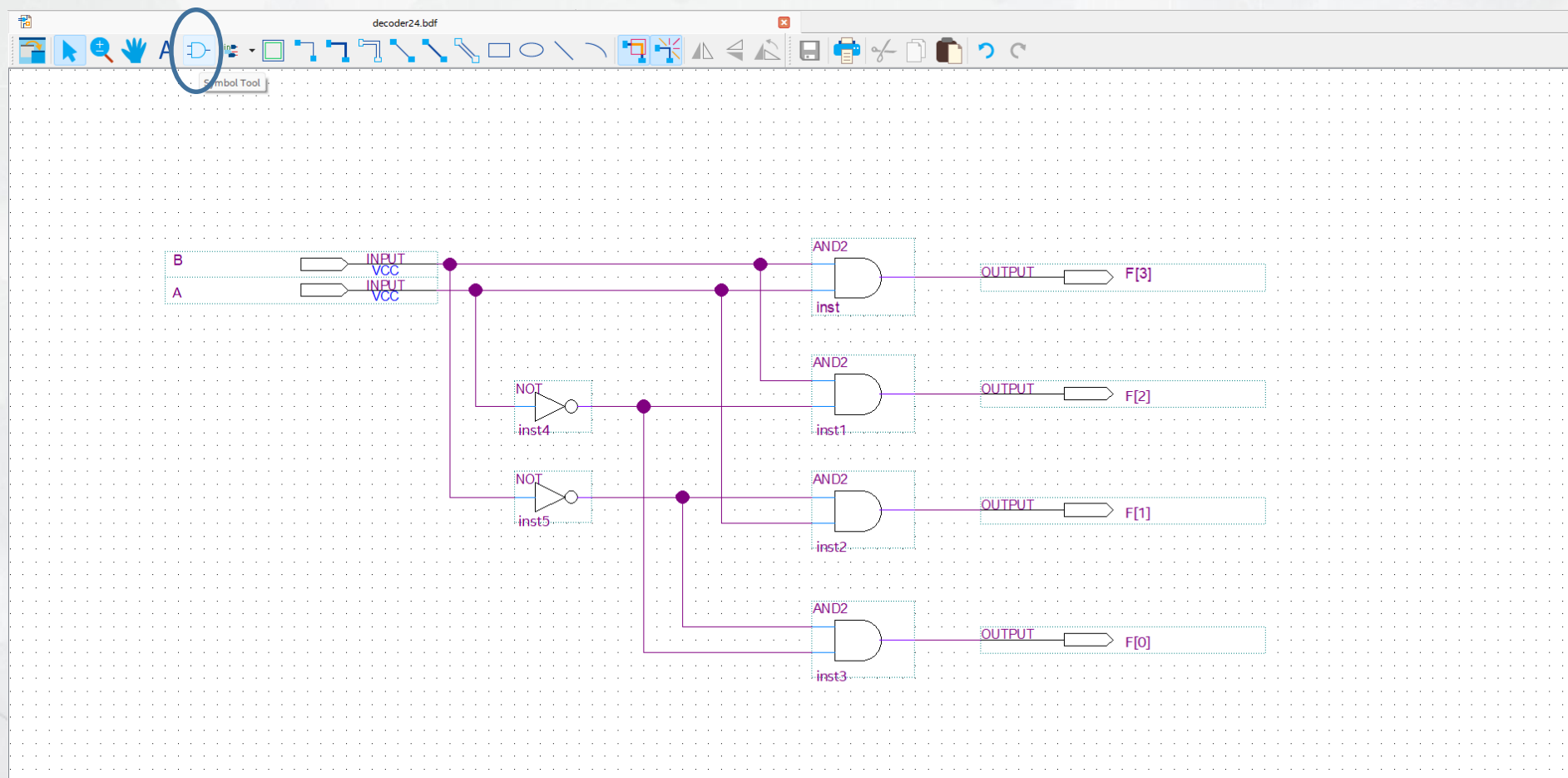
主菜单“File”→“New Project Wizard”，新建工程decoder24（实体名）

主菜单“File”→“New”项，选择Block Diagram/Schematic File，新建框图文件，保存为 decoder24.bdf。



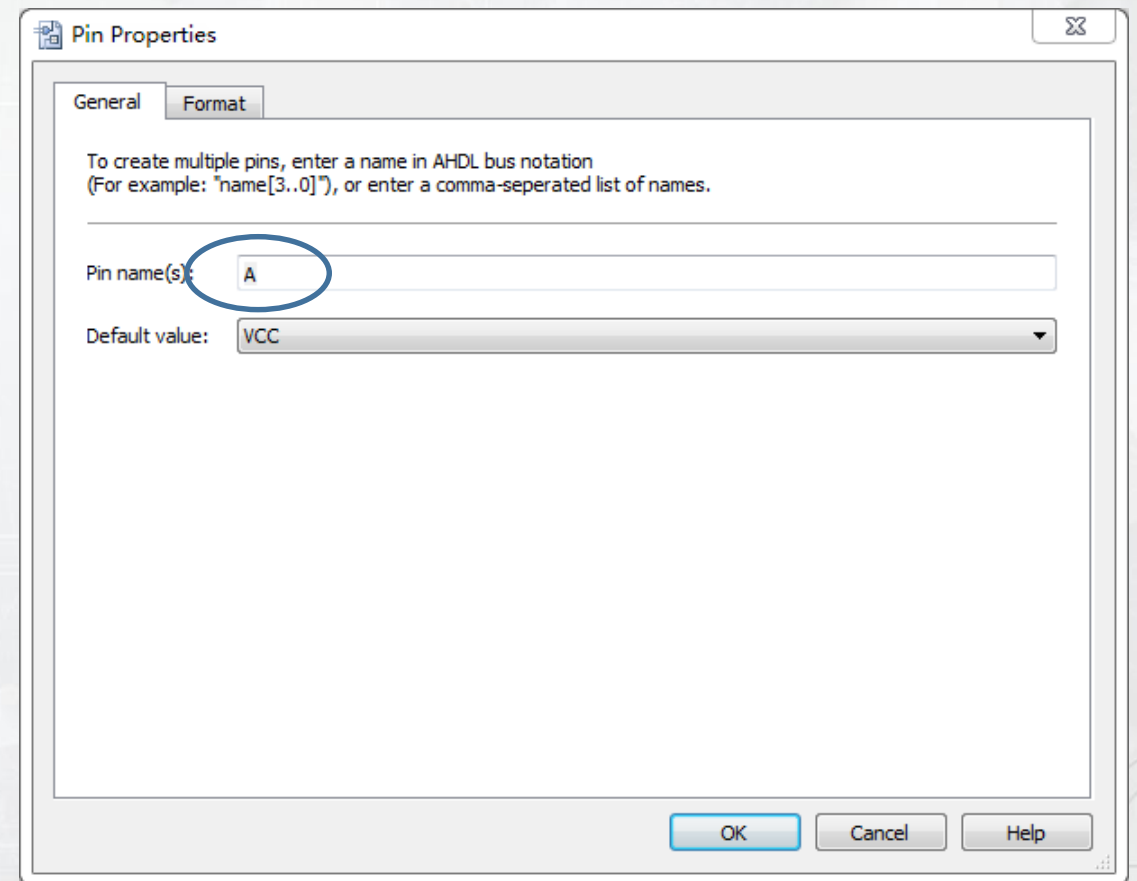
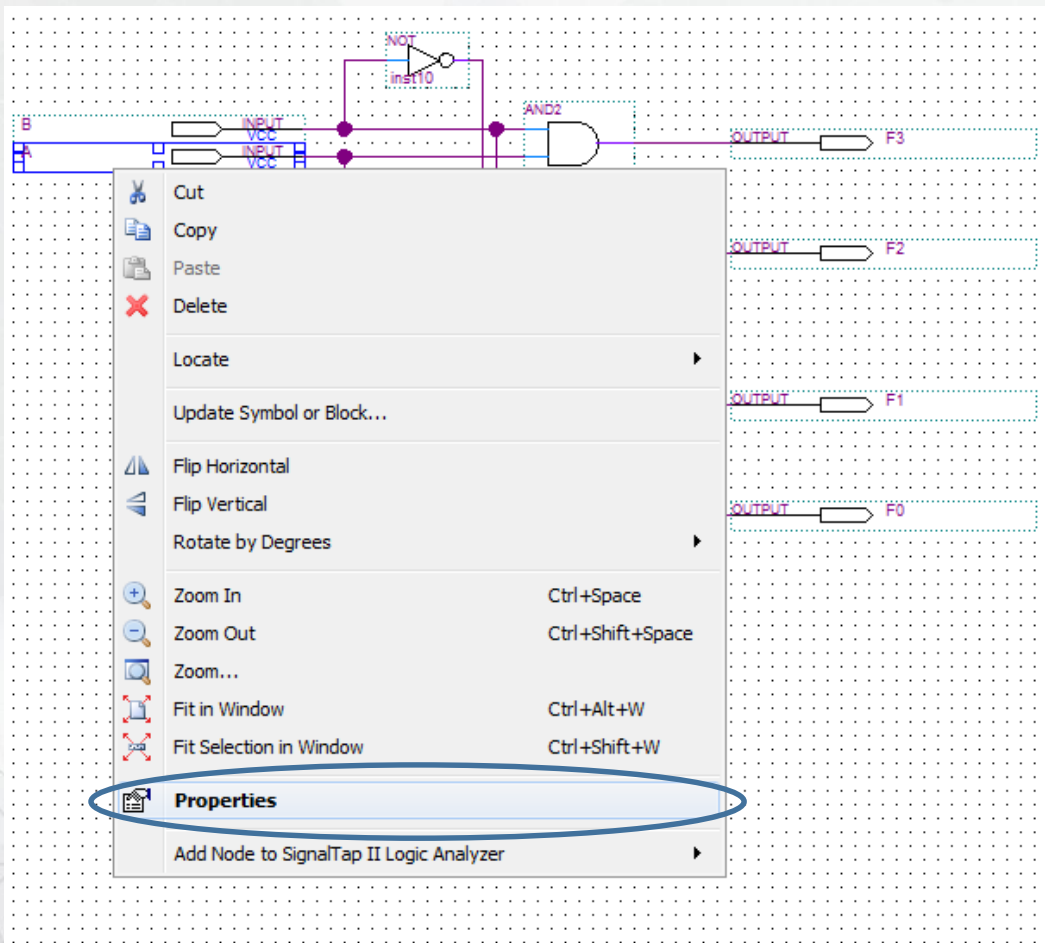
## 2-4译码器电路图

在框图编辑器中，利用元器件库中的门电路设计2-4译码器电路



# 修改节点与端口命名

右键点击节点或者输入输出端口，选择“Properties”，在Properties页面修改。



# 编译前设置与编译

## ✓ 目标芯片的选择

主菜单“Assignmemts”→“Device”项，选择Cyclone IV E系列  
EP4CE55F23C8芯片

## ✓ 编译电路

主菜单“Processing”→“Start Compliation”项，启动编译

## 2-4译码器波形仿真

新建波形图文件 (\*.vwf),设置仿真时间, 添加输入输出端口, 设置输入信号值, 保存文件。运行仿真

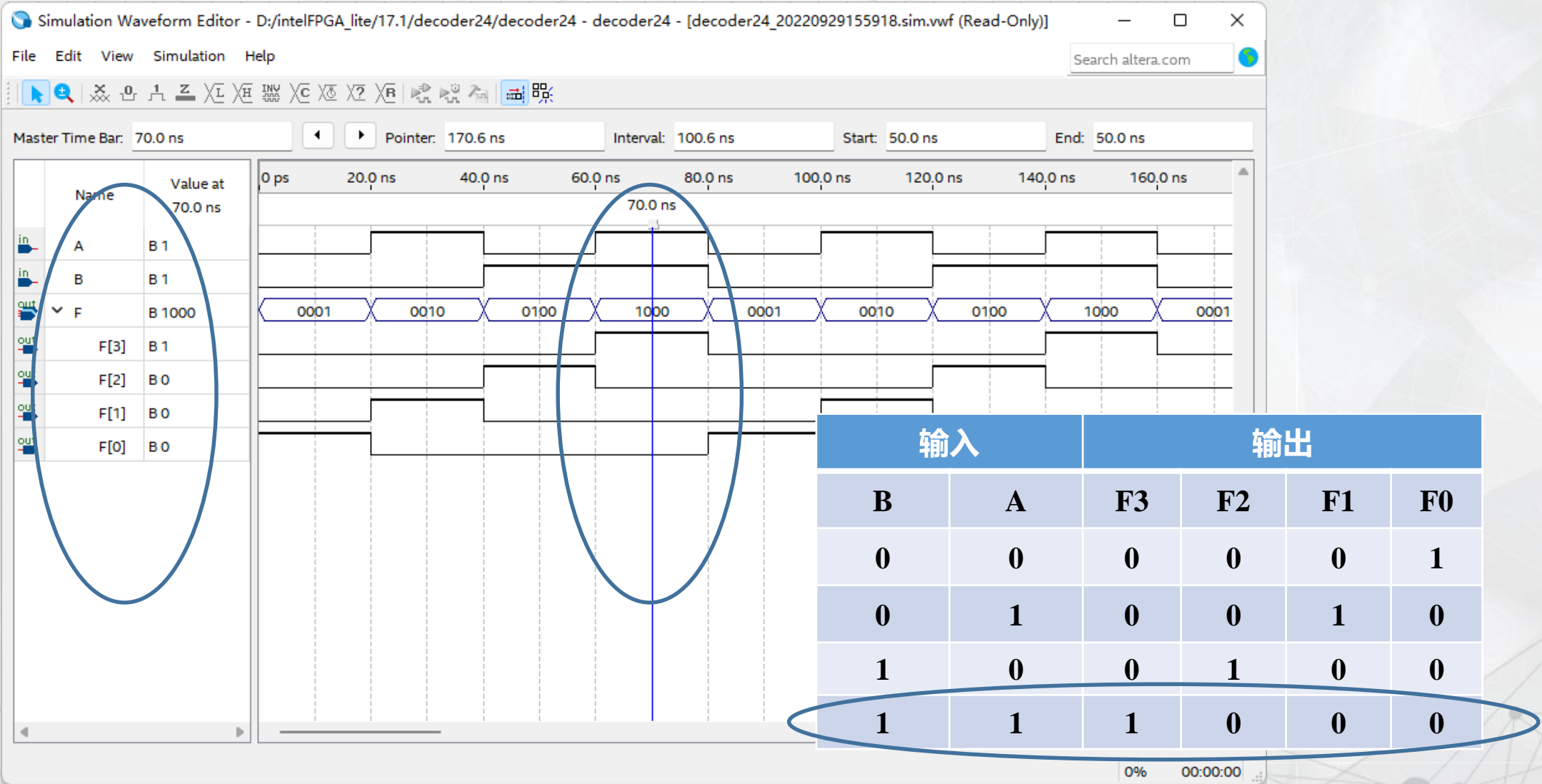
**建立仿真波形文件：**主菜单“File”→“New”项，选择University Program VWF，新建\*.vwf，打开波形编辑器。

**设置仿真时间：**主菜单“Edit”→“Set End Time”项。

**添加输入输出端口：**波形编辑器窗口主菜单 “Edit” → “Insert”→“Insert Node or Bus”

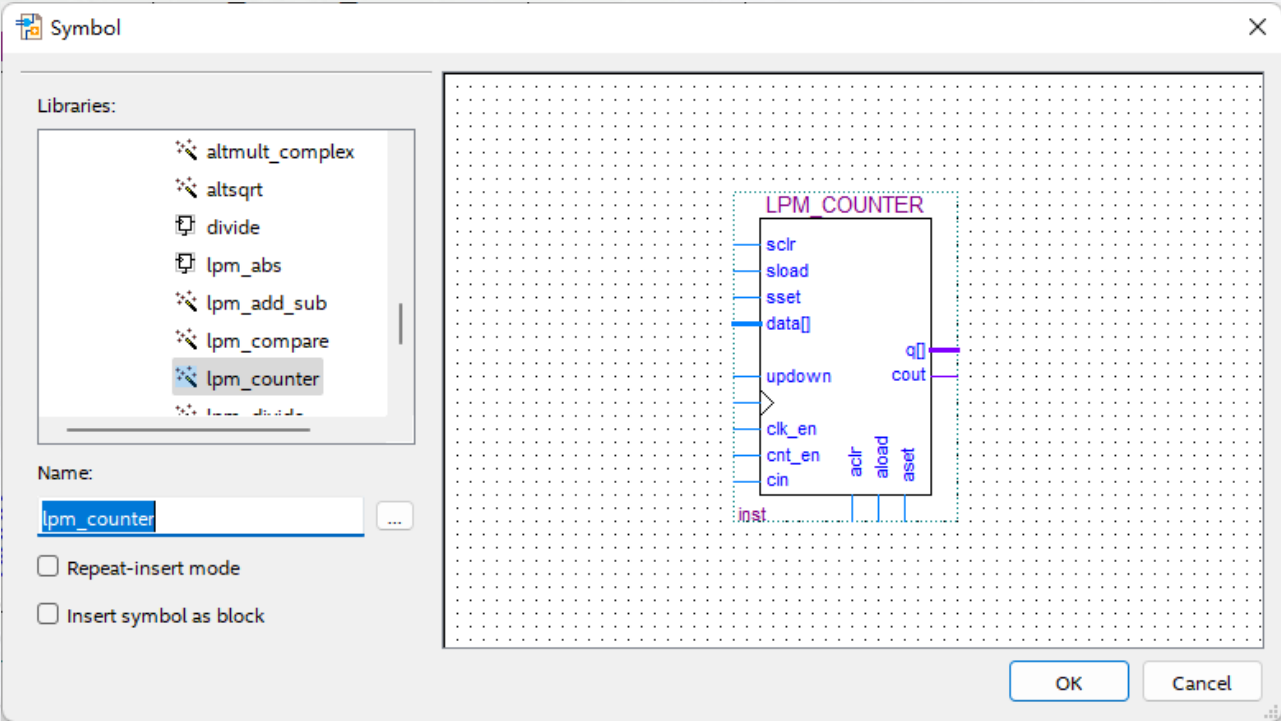
**运行仿真：**波形编辑器窗口主菜单“Simulation”→“Run Functional Simulation”项。

# 2-4译码器仿真波形图分析



# Intel宏功能模块

Quartus Prime软件针对常用功能，提供了参数化（Parameterized）的宏功能（Megafunctions）模块，通过在框图编辑器里面调用宏功能模块，可以减少工作量，加快设计的进程。

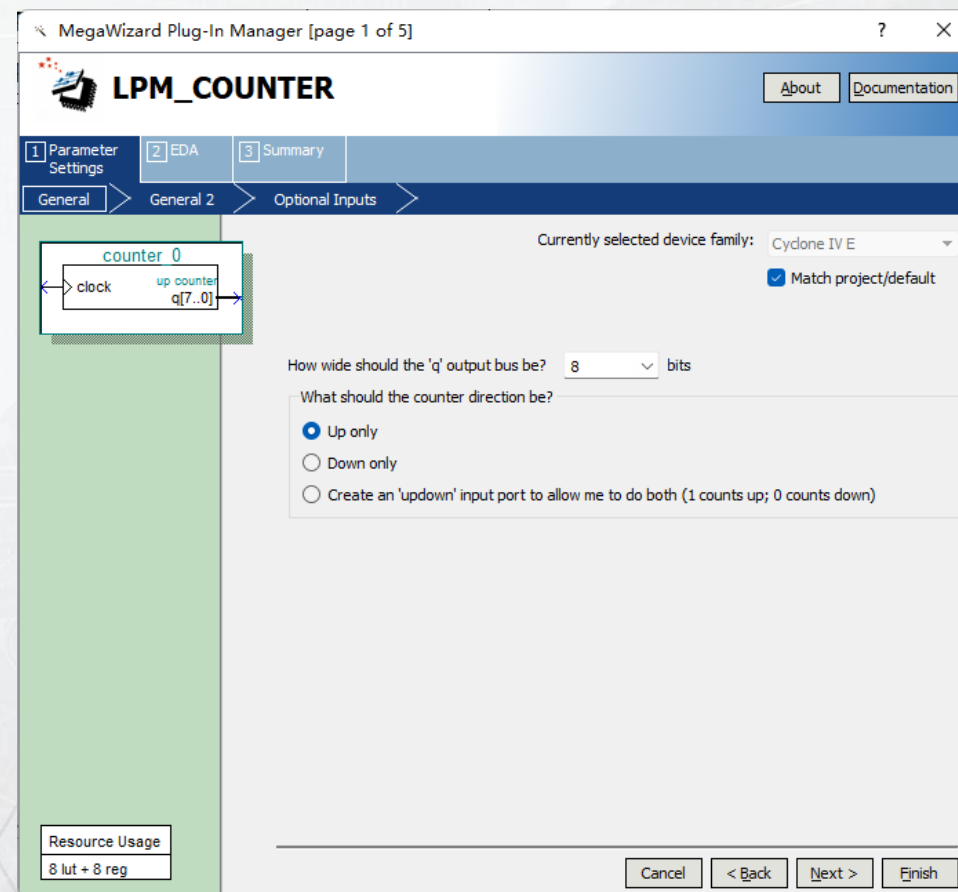
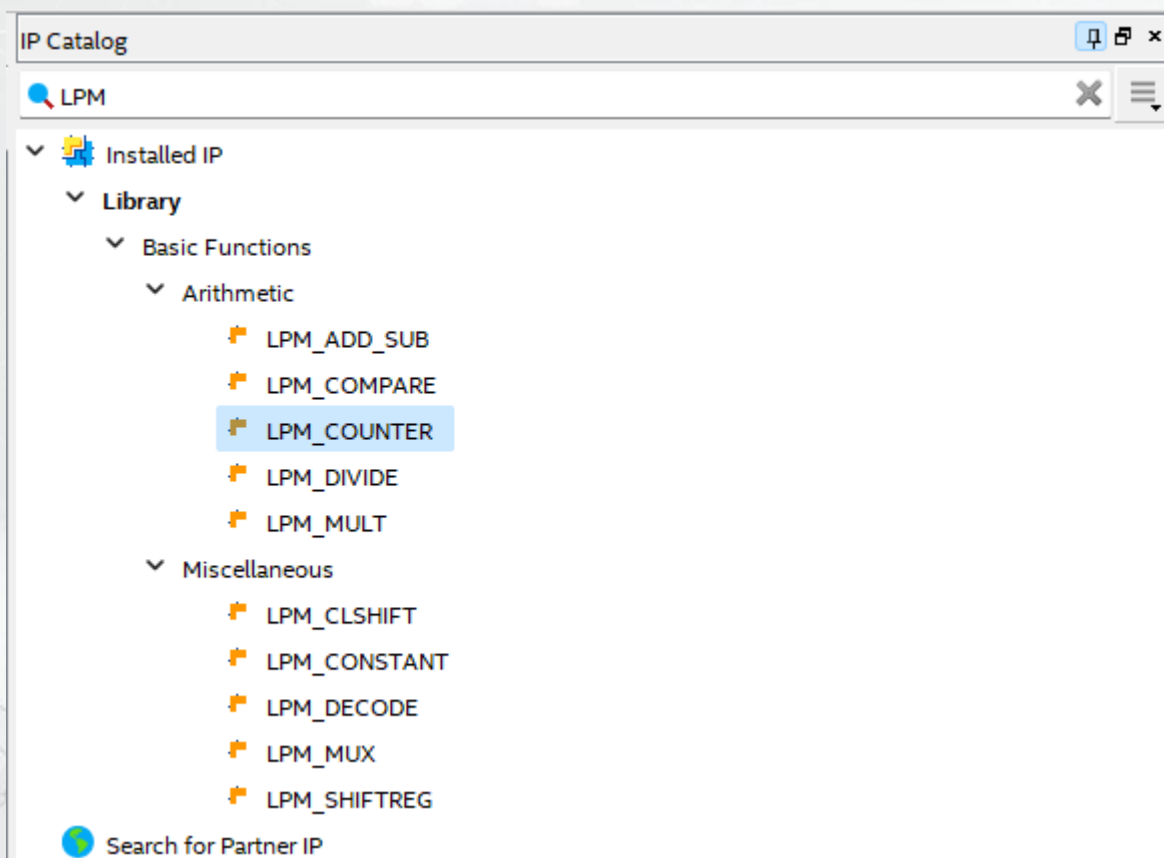


A schematic diagram of the 'LPM\_COUNTER' module. The module has inputs: 'sclr', 'sload', 'sset', 'data[]', 'updown', 'clk\_en', 'cnt\_en', and 'cin'. It has outputs: 'q[]' and 'cout'. There are also three pins labeled 'acir', 'aload', and 'aset'. The module is labeled 'LPM\_COUNTER' at the top.

Parameter	Value
LPM_SVALUE	
LPM_AVALUE	
LPM_MODULUS	
LPM_DIRECTION	
LPM_WIDTH	
LPM_PORT_UPDOWN	

# Intel IP Core

用户也可以通过向导工具**IP Catalog**调用IP核，这里面的IP核包含了带有配置向导的宏功能块。该向导工具帮助用户建立或修改包含自定义宏功能模块变量的设计文件，并在用户的设计中进行实例化。

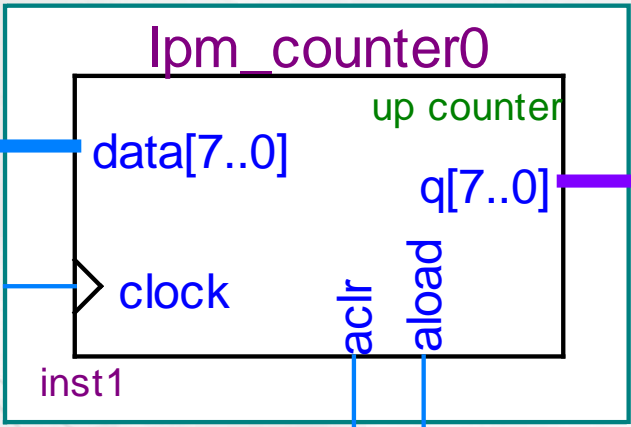


# 宏功能块及IP核分类

- ✓ Intel的参数化模块库 (LPM)
- ✓ 器件专有的IP Core
- ✓ 用IP生成工具调用的自定义IP Core
- ✓ 第三方IP Core。

# 宏功能模块——lpm\_counter计数器实验

计数器是实现计数运算的逻辑电路。计数器是由基本的计数单元和一些控制门所组成，计数单元则由一系列具有存储信息功能的各类触发器构成。

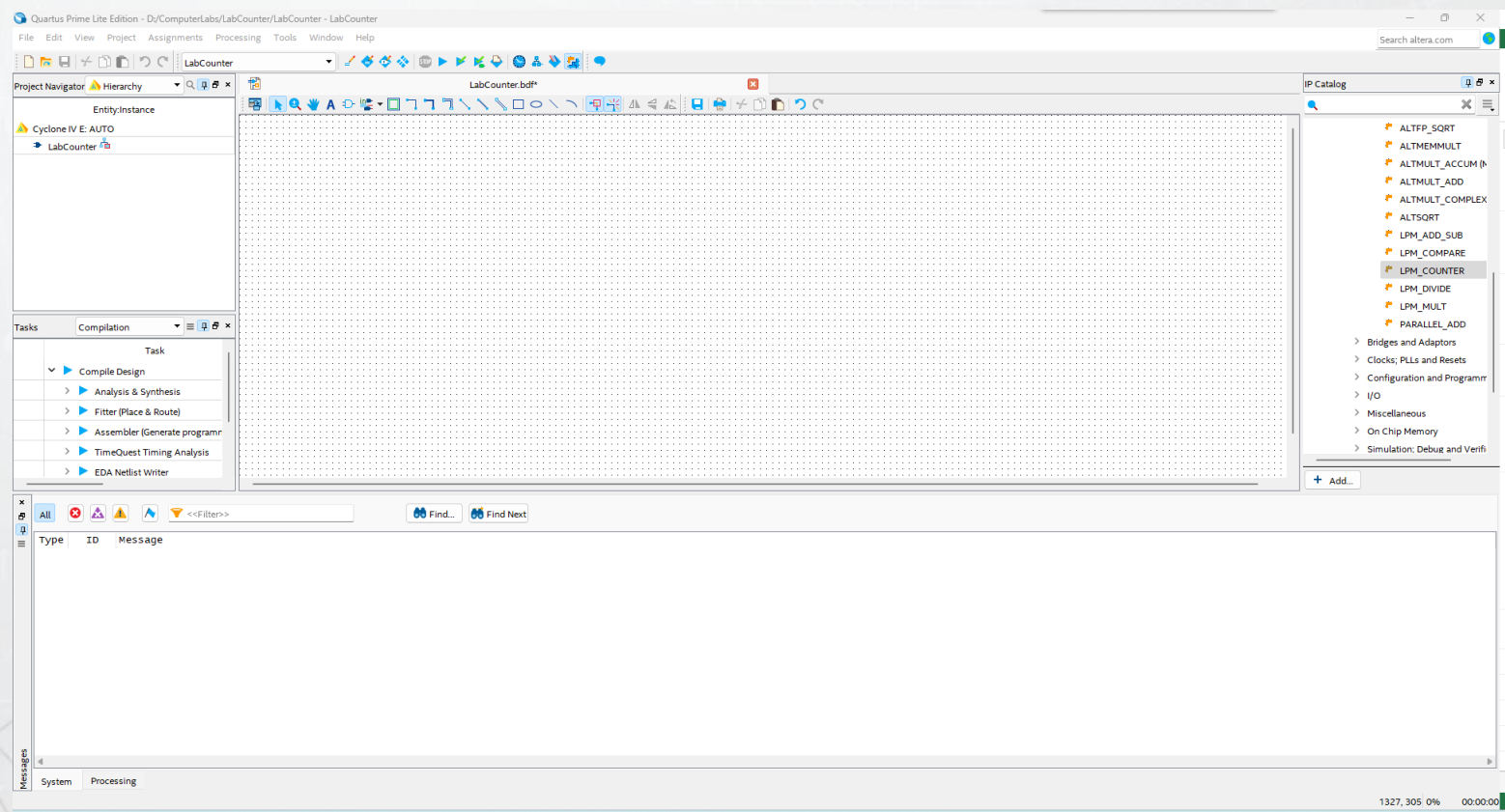
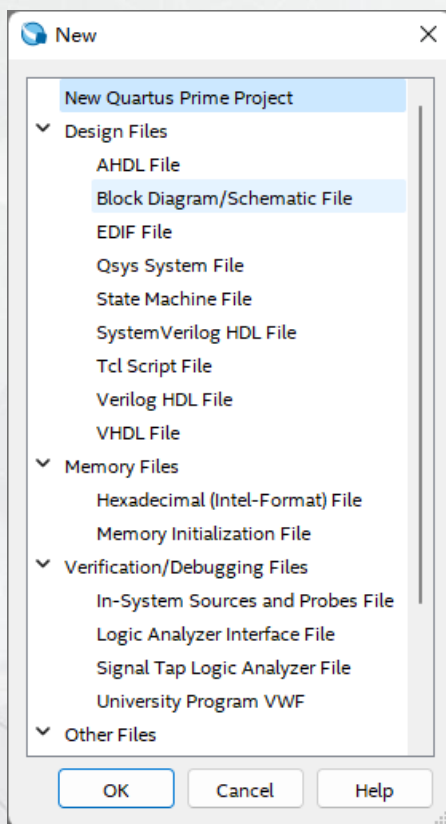


信号名称	功能说明
clock	输入时钟脉冲
data[7..0]	lpm_counter的8位数据端
q[7..0]	lpm_counter的8位数据输出端
aclr	异步清零信号，高电平有效
aload	异步加载数据信号，当aload为高电平时，计数器处于并行置数状态，输出q等于data，当aload为低电平时，计数器处于计数状态。

# 框图编辑器(新建\*.bdf文件)

主菜单“File”→“New Project Wizard”，新建工程LabCounter（实体名）

主菜单“File”→“New”项，选择Block Diagram/Schematic File，新建框图文件，保存为 LabCounter.bdf。



# 从IP Catalog中选择计数器(lpm\_counter)IP

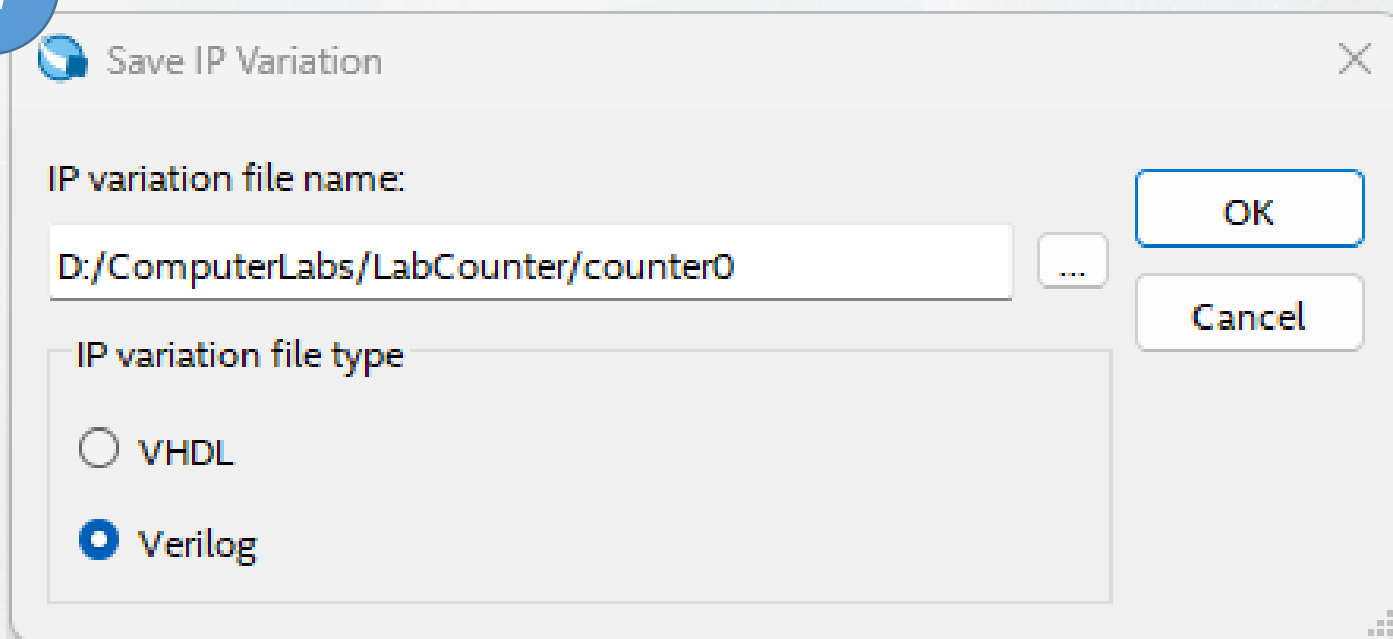
通过 “Tools” → “IP Catalog” 打开IP Catalog栏

1



# 设置lpm\_counter参数——设置输出文件类型

2



The image shows a 'Save IP Variation' dialog box. It has a title bar with a blue icon and a close button. The main area contains two sections. The first section is labeled 'IP variation file name:' and has a text input field containing 'D:/ComputerLabs/LabCounter/counter0'. To the right of the input field is a button with three dots. To the right of the input field are two buttons: 'OK' and 'Cancel'. The second section is labeled 'IP variation file type' and contains two radio buttons. The first radio button is labeled 'VHDL' and is unselected. The second radio button is labeled 'Verilog' and is selected, indicated by a blue dot.

Save IP Variation

IP variation file name:

D:/ComputerLabs/LabCounter/counter0

IP variation file type

☐ VHDL

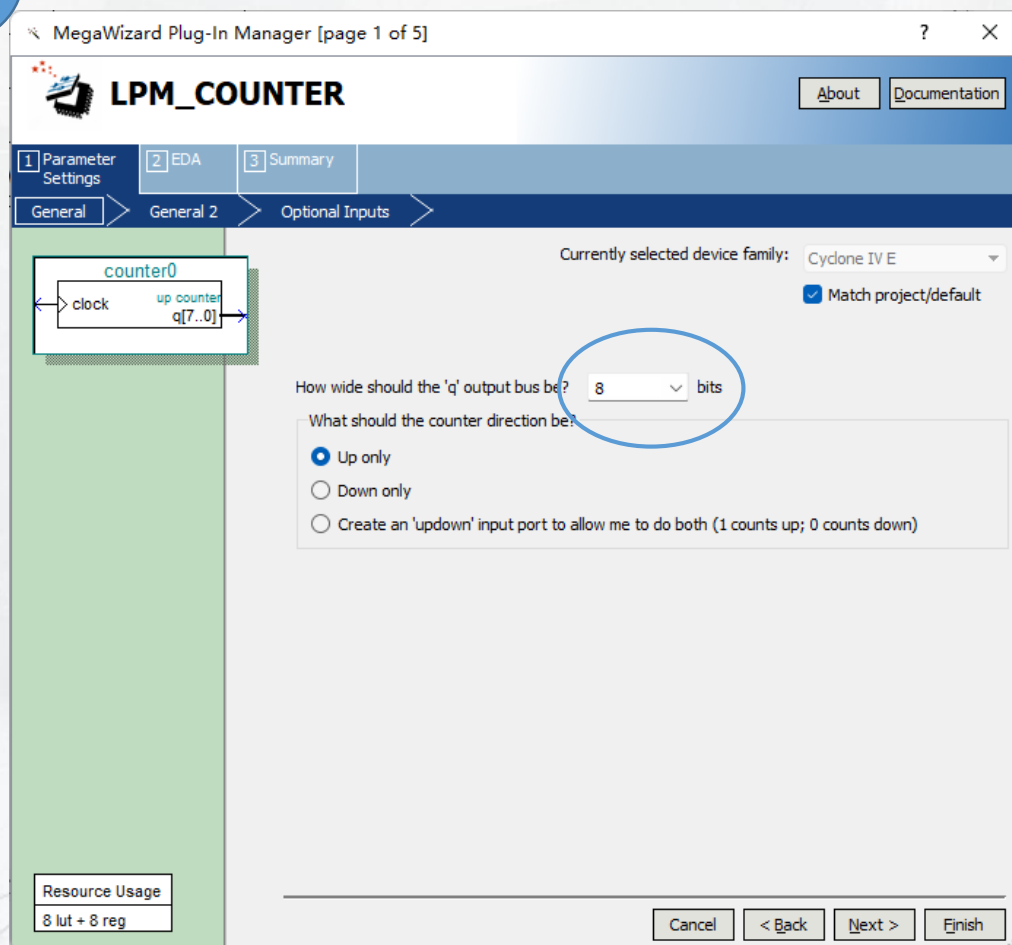
☒ Verilog

OK

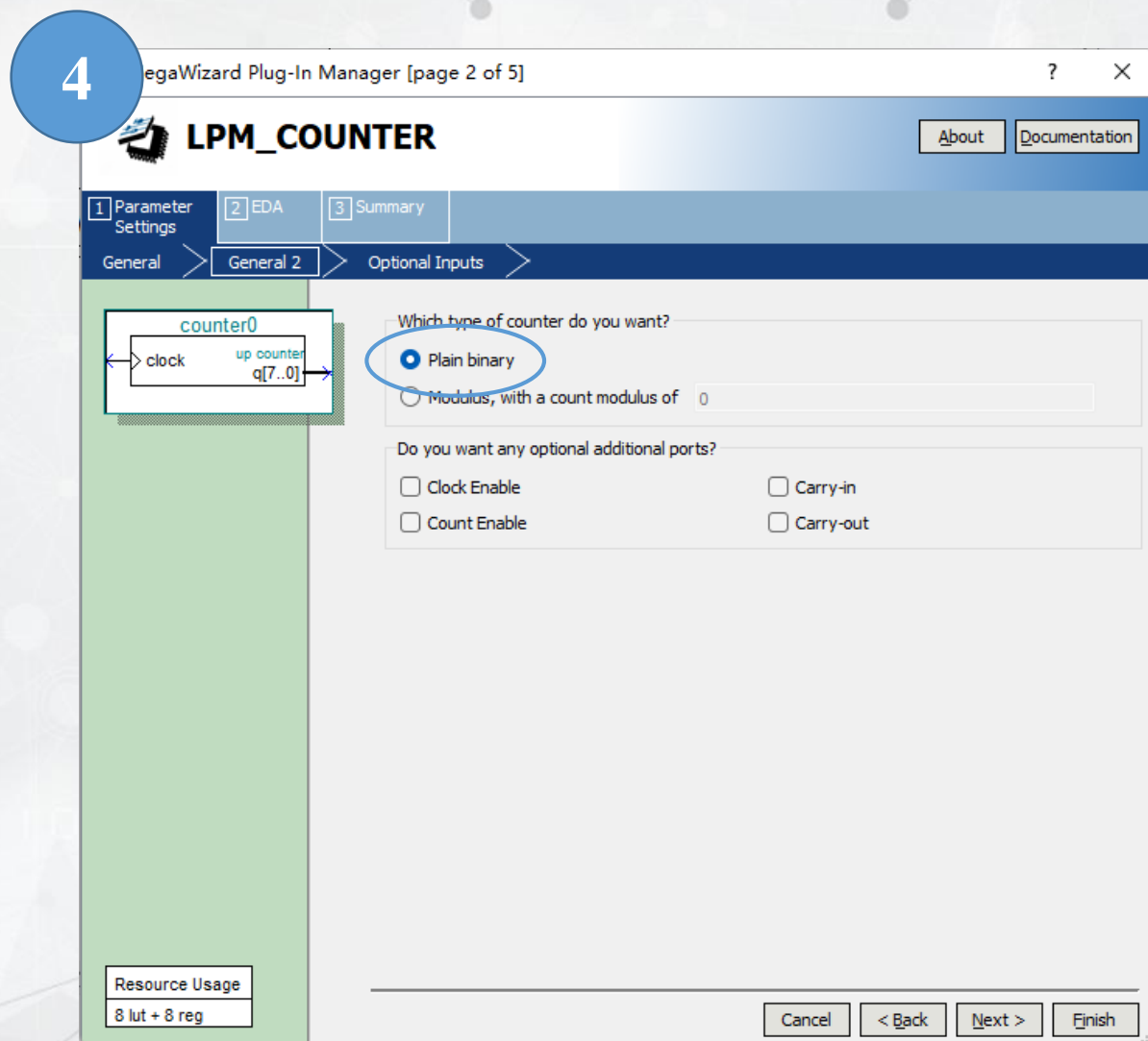
Cancel

# 设置lpm\_counter参数——设置输出端口宽度与递增方向

3



# 设置lpm\_counter参数——设置计数器模值



# 设置lpm\_counter参数——设置计数器同步或异步

5

MegaWizard Plug-In Manager [page 3 of 5]

**LPM\_COUNTER** [About](#) [Documentation](#)

1 Parameter Settings 2 EDA 3 Summary

General > General 2 > Optional Inputs >

**counter0**

data[7..0] up counter q[7..0]

clock acir aload

Do you want any optional inputs?

Synchronous inputs

☐ Clear

☐ Load

☐ Set

☒ Set to all 1's

☐ Set to 0

Asynchronous inputs

☒ Clear

☒ Load

☐ Set

☒ Set to all 1's

☐ Set to 0

Resource Usage

8 lut + 16 mux21 + 8 reg

Cancel < Back Next > Finish

# 设置lpm\_counter参数——设置是否创建网表netlist

6

MegaWizard Plug-In Manager [page 4 of 5]

**LPM\_COUNTER** [About](#) [Documentation](#)

1 Parameter Settings 2 EDA 3 Summary

**counter0**

data[7..0] up counter q[7..0]

clock acir aload

Simulation Libraries

To properly simulate the generated design files, the following simulation model file(s) are needed

File	Description
lpm	LPM megafunction simulation library

Timing and resource estimation

Generates a netlist for timing and resource estimation for this megafunction. If you are synthesizing your design with a third-party EDA synthesis tool, using a timing and resource estimation netlist can allow for better design optimization.

Not all third-party synthesis tools support this feature - check with the tool vendor for complete support information.

Note: Netlist generation can be a time-intensive process. The size of the design and the speed of your system affect the time it takes for netlist generation to complete.

☐ Generate netlist

Resource Usage

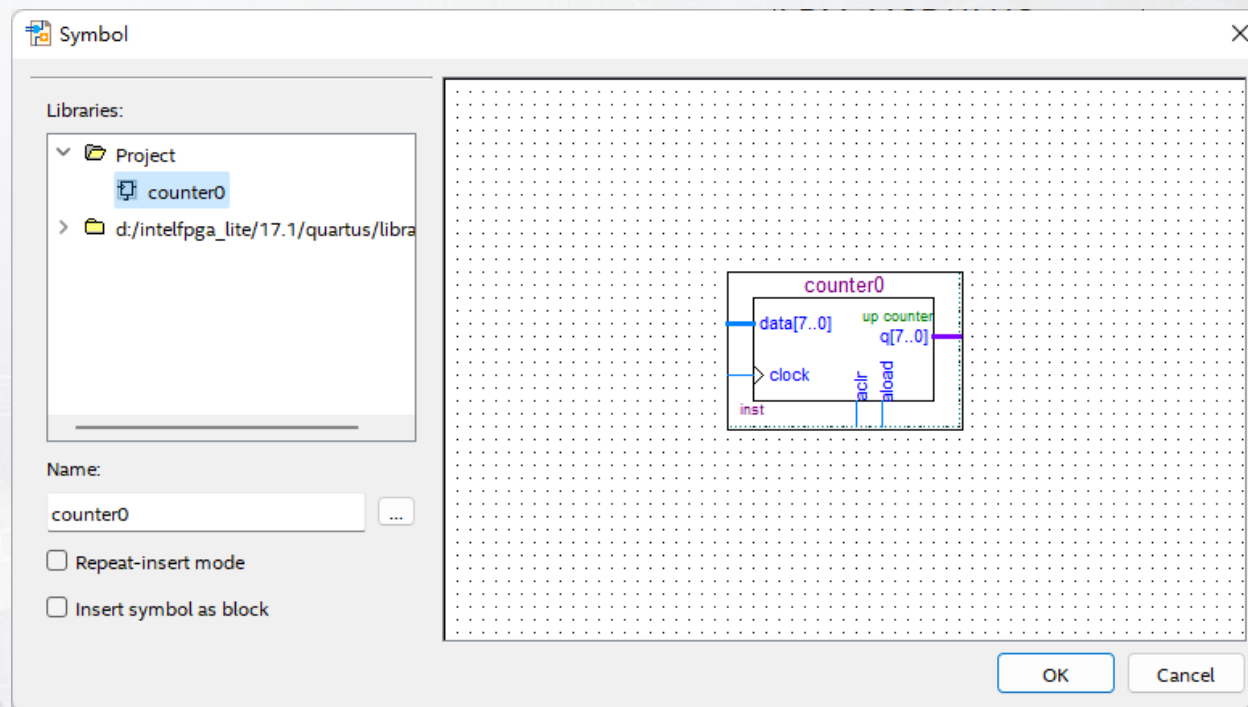
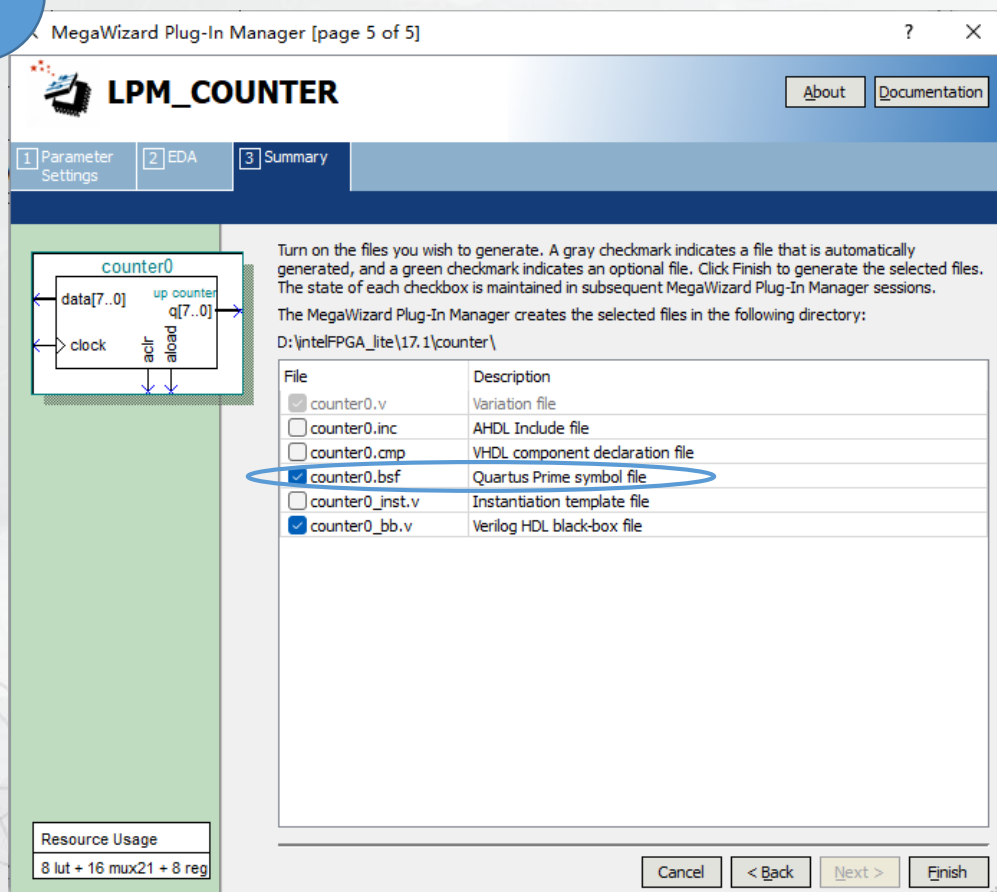
8 lut + 16 mux21 + 8 reg

Cancel < Back Next > Finish

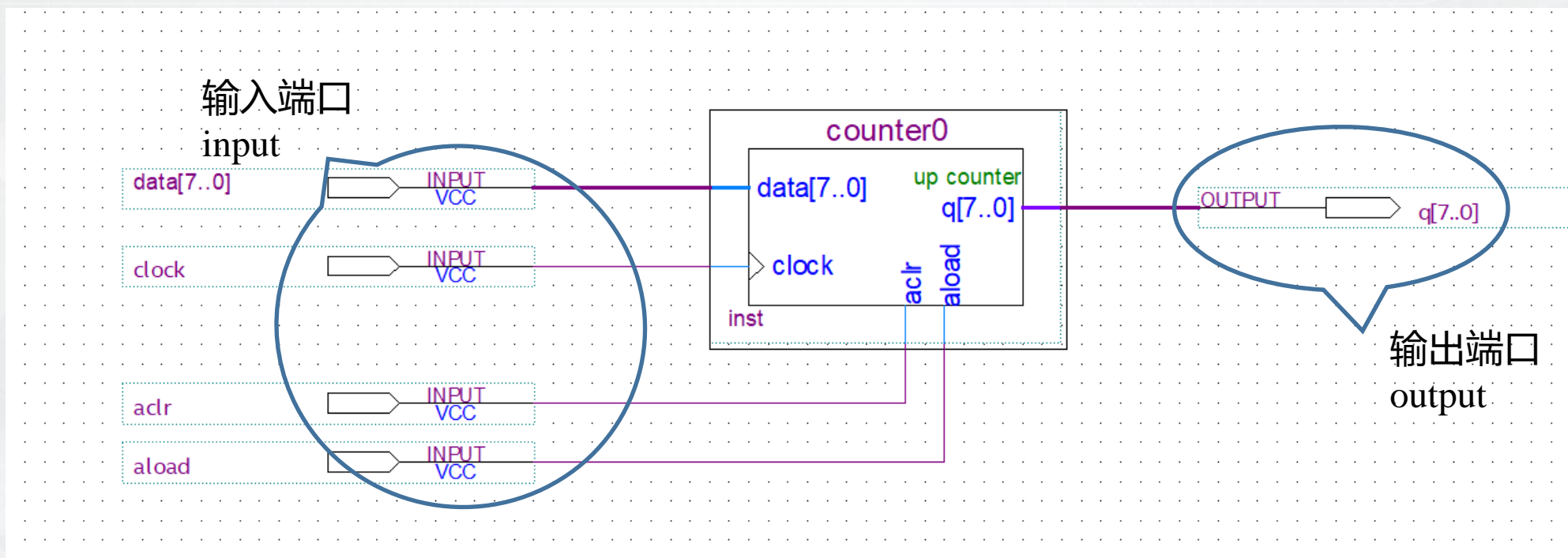
## 设置lpm\_counter参数

- 设置需要创建的文件，完成参数设置，在bdf中插入一个计数器元件
- 注意：必须选中bsf文件（Quartus Prime symbol file）
- 点击Finish后，一定要在弹出的窗口点击Yes，将IP添加到当前工程

7

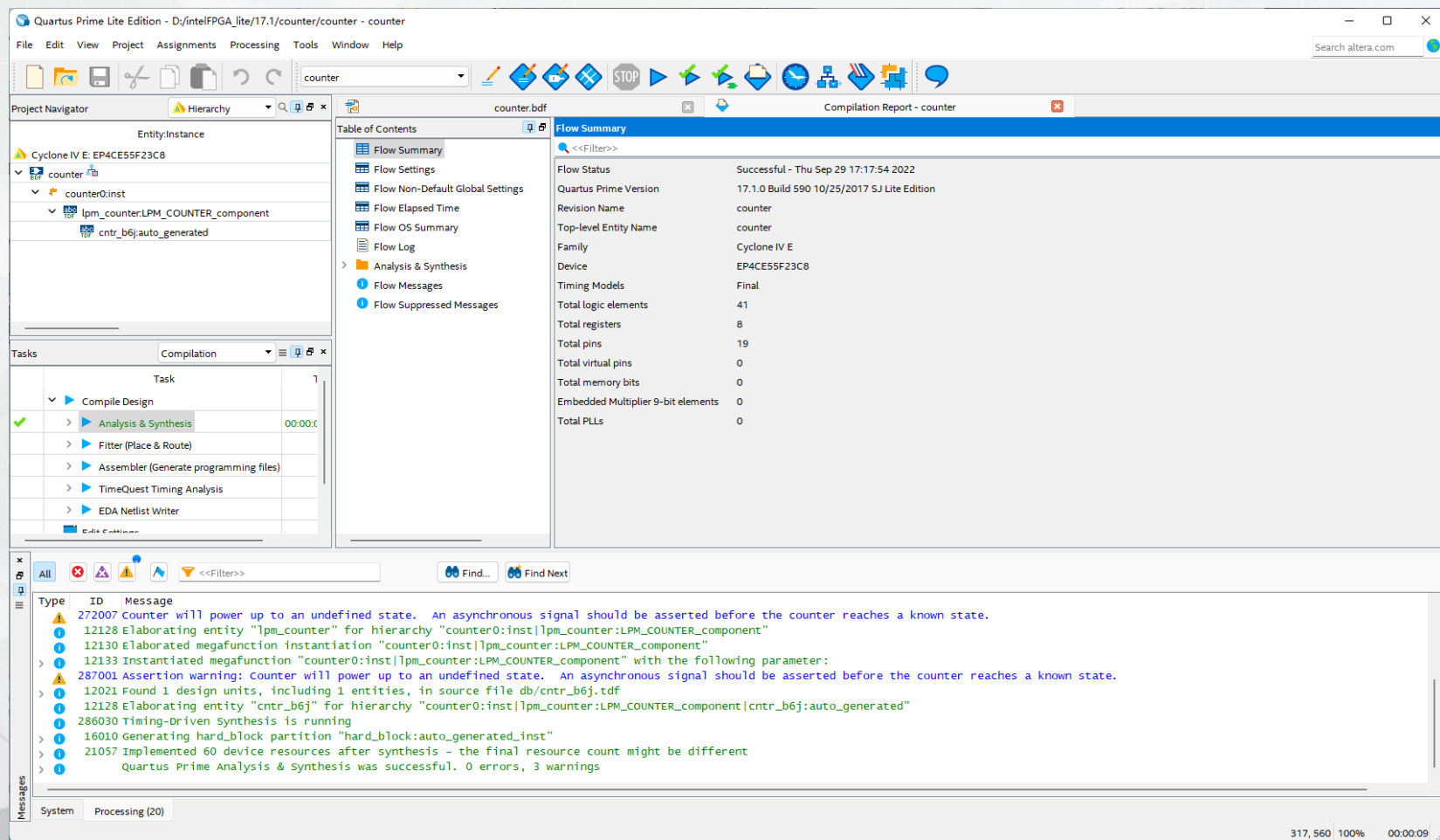


# lpm\_counter计数器实验——为计数器(lpm\_counter)添加输入输出端口



# lpm\_counter计数器实验——编译前设置与编译

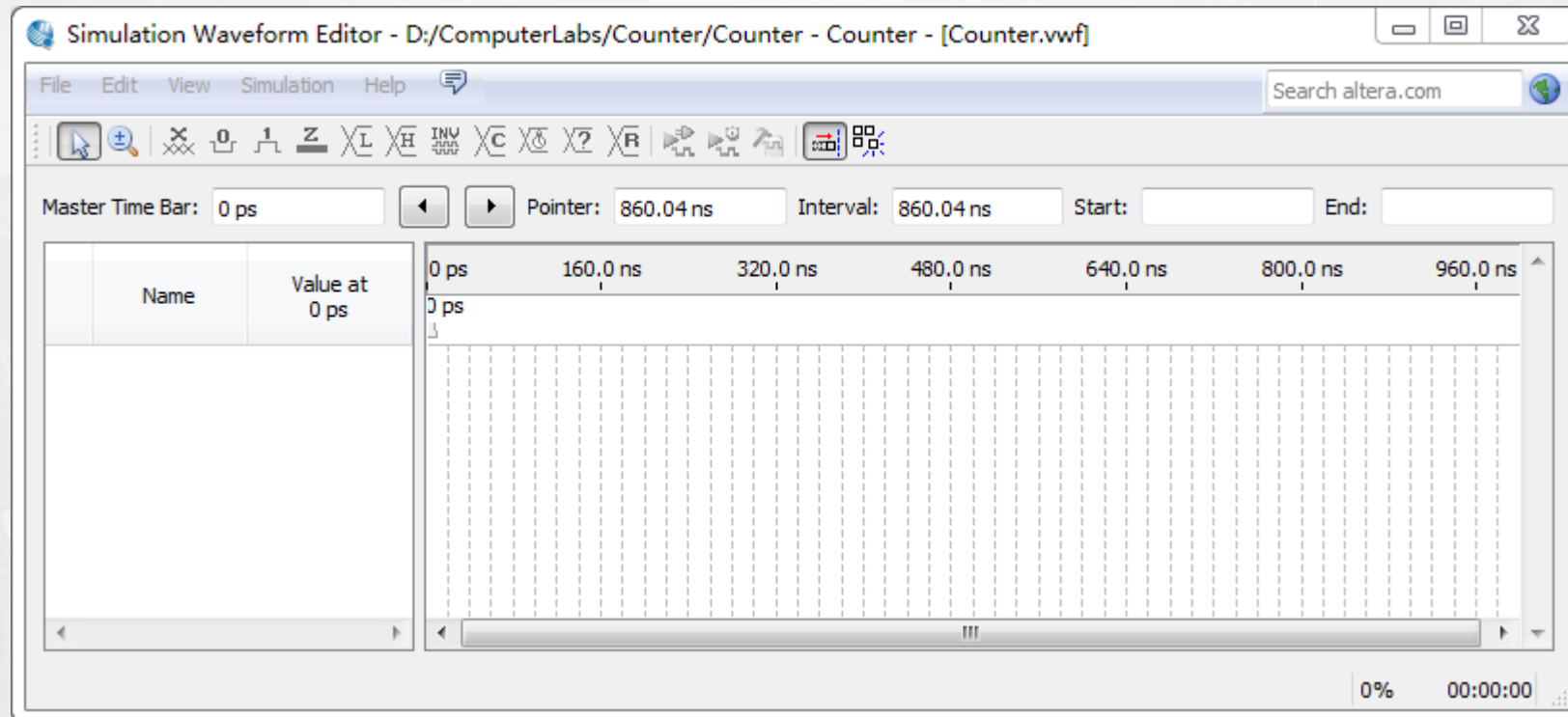
主菜单“Assignments”→“Device”项，选择Cyclone IV E系列 EP4CE55F23C8芯片  
主菜单“Processing”→“Start Compilation”项,或者工具栏 “▶”，启动编译。



# lpm\_counter计数器实验——新建仿真波形图\*.vwf

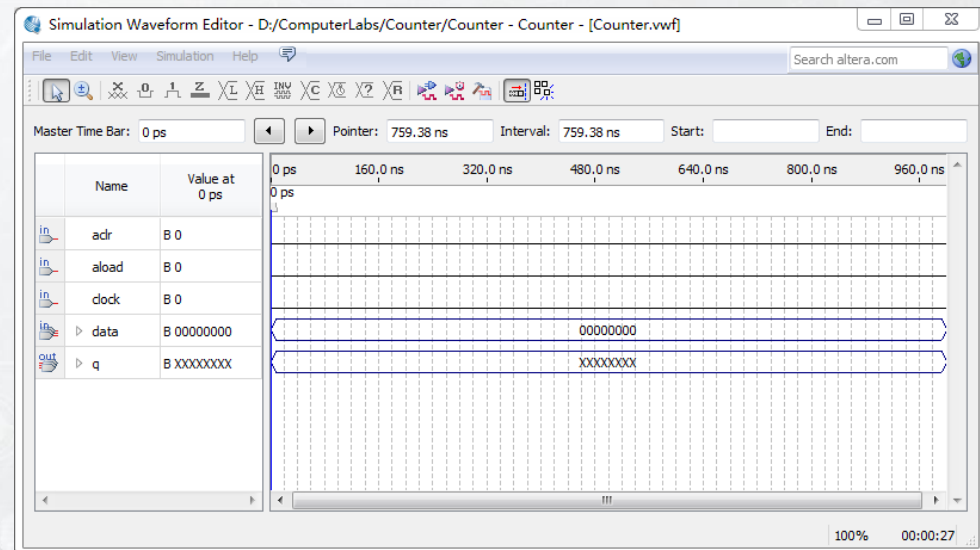
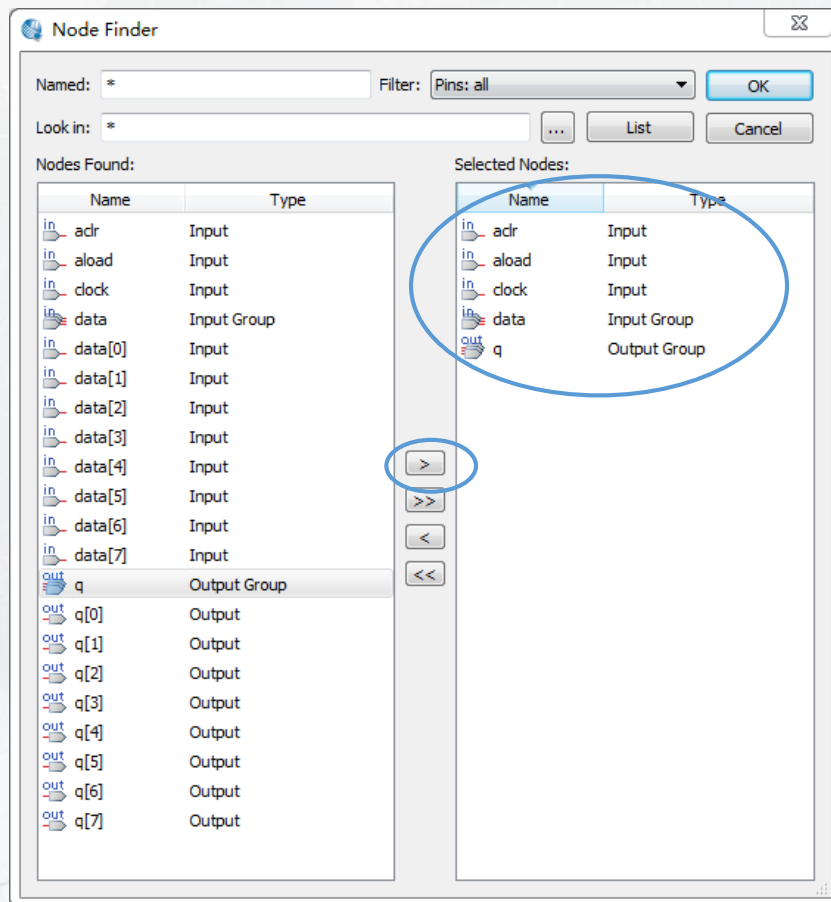
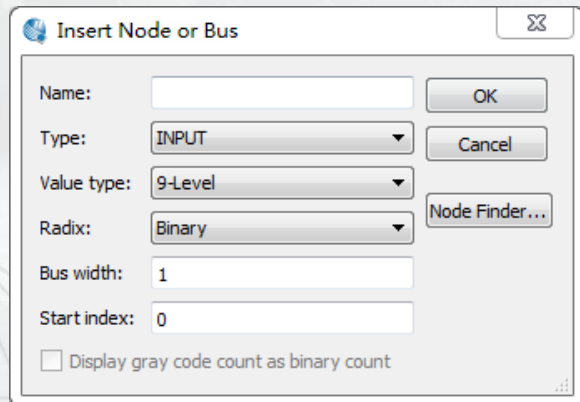
**建立仿真波形文件：**主菜单“File”→“New”项，选择University Program VWF，新建\*.vwf，打开波形编辑器。

**设置仿真时间：**主菜单“Edit”→“Set End Time”项。

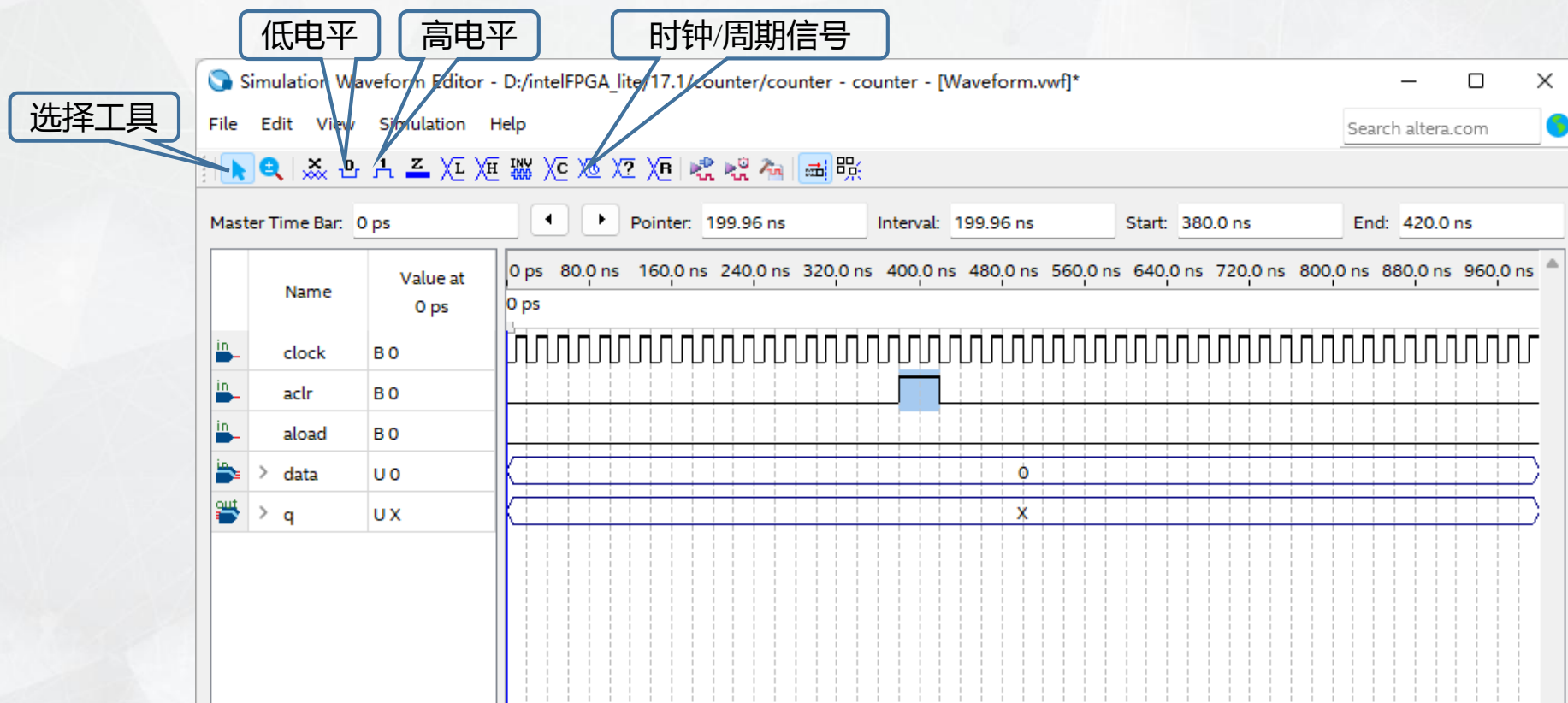


# lpm\_counter计数器实验——在波形图中添加输入输出端口

**添加输入输出端口：**波形编辑器菜单“Edit”→“Insert”→“Insert Node or Bus” 点击“Node Finder”按钮，添加输入输出端口



# lpm\_counter计数器实验——设置波形（设置输入信号值）



# 计数器(lpm\_counter)仿真要求

**运行仿真：**波形编辑器窗口主菜单 “Simulation” → “Run Functional Simulation” 项

**仿真要求：**

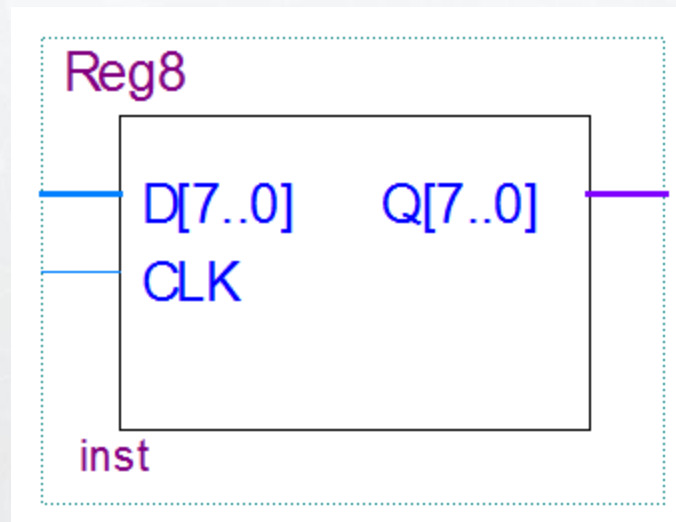
- 1、全面仿真计数器的功能，测试计数器的置数功能、计数功能和清零功能。
- 2、分析波形图，说明计数器的功能。

**注意：**先建工程，再建顶层设计文件，再加入宏单元lpm\_counter

# 8位寄存器基本原理

寄存器的功能是存储二进制代码，它是由具有存储功能的触发器组合起来构成的。一个触发器可以存储1位二进制代码，故存放n位二进制代码的寄存器，需用n个触发器来构成。

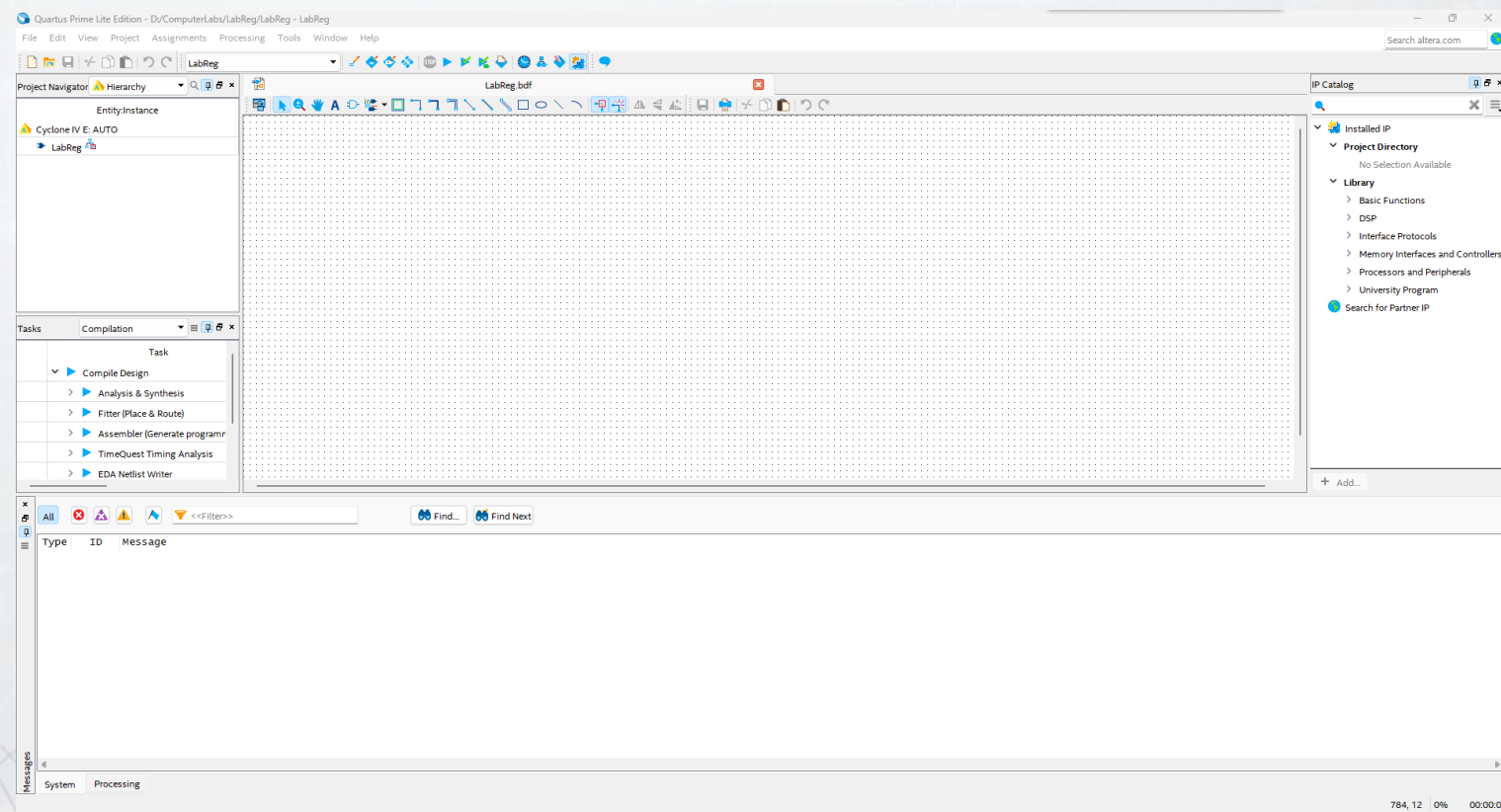
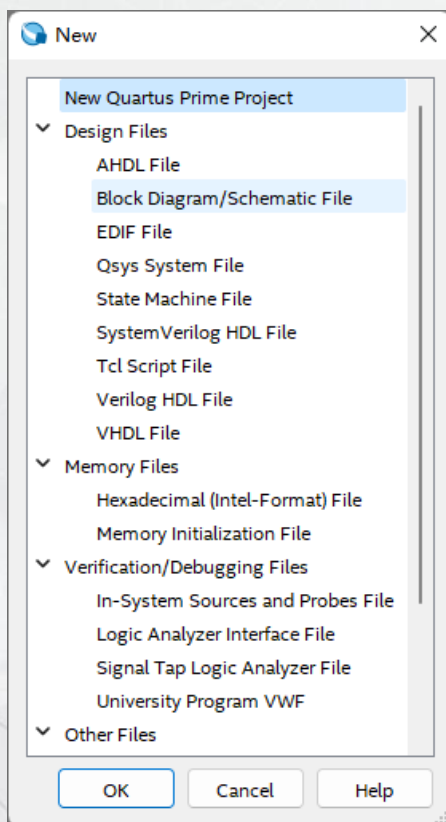
8位寄存器元件符号



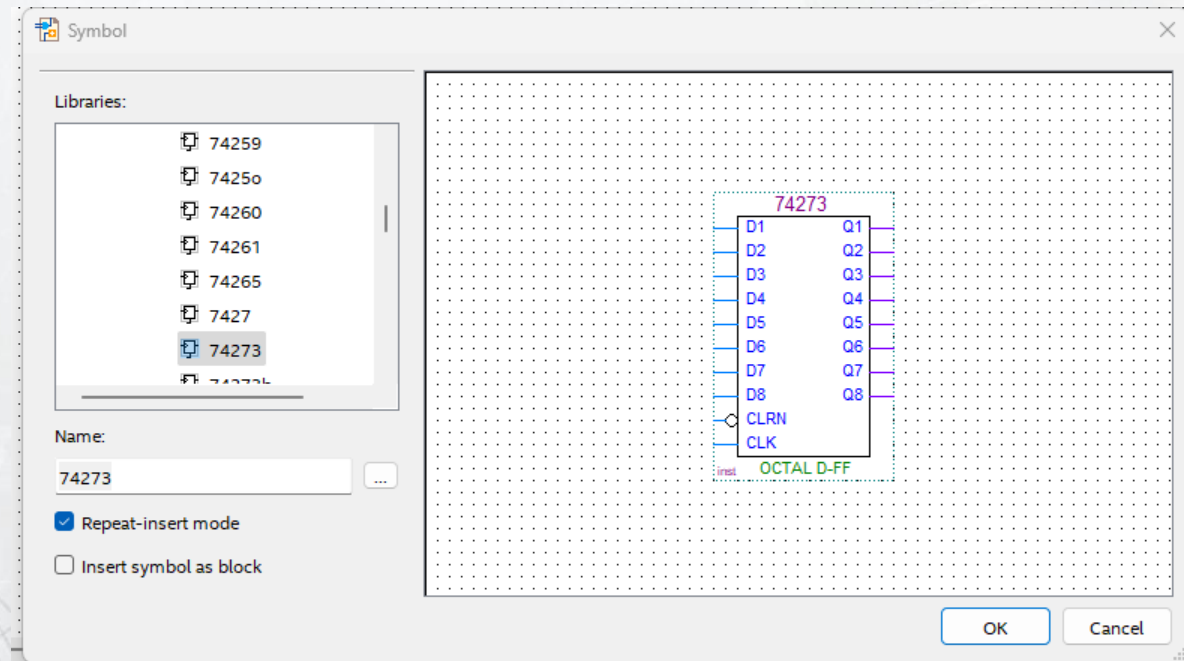
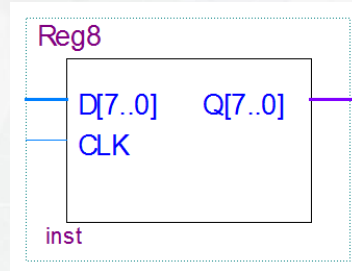
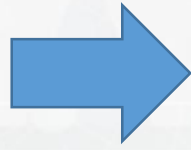
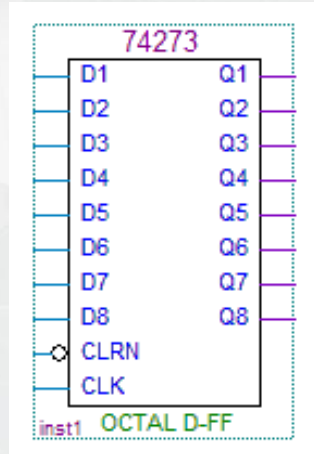
# 框图编辑器(新建\*.bdf文件)

主菜单“File”→“New Project Wizard”，新建工程LabReg（实体名）

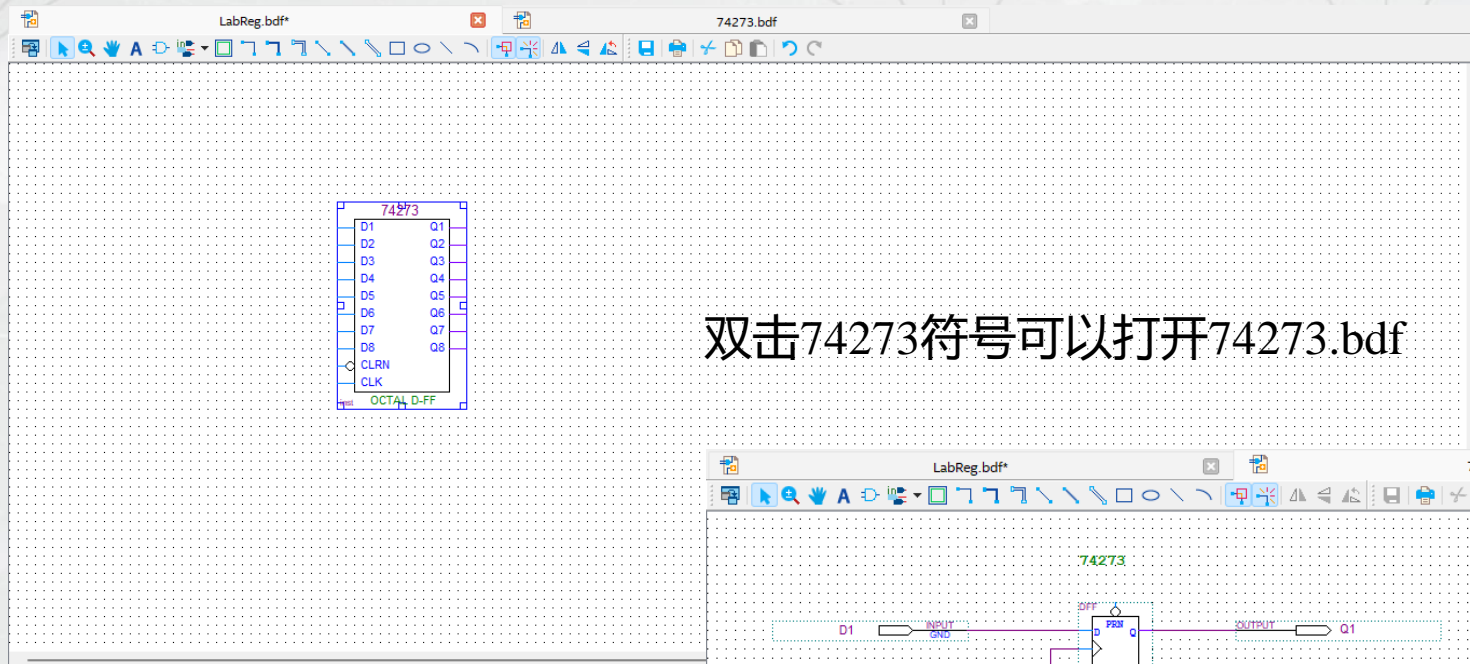
主菜单“File”→“New”项，选择Block Diagram/Schematic File，新建框图文件，保存为 LabReg.bdf。



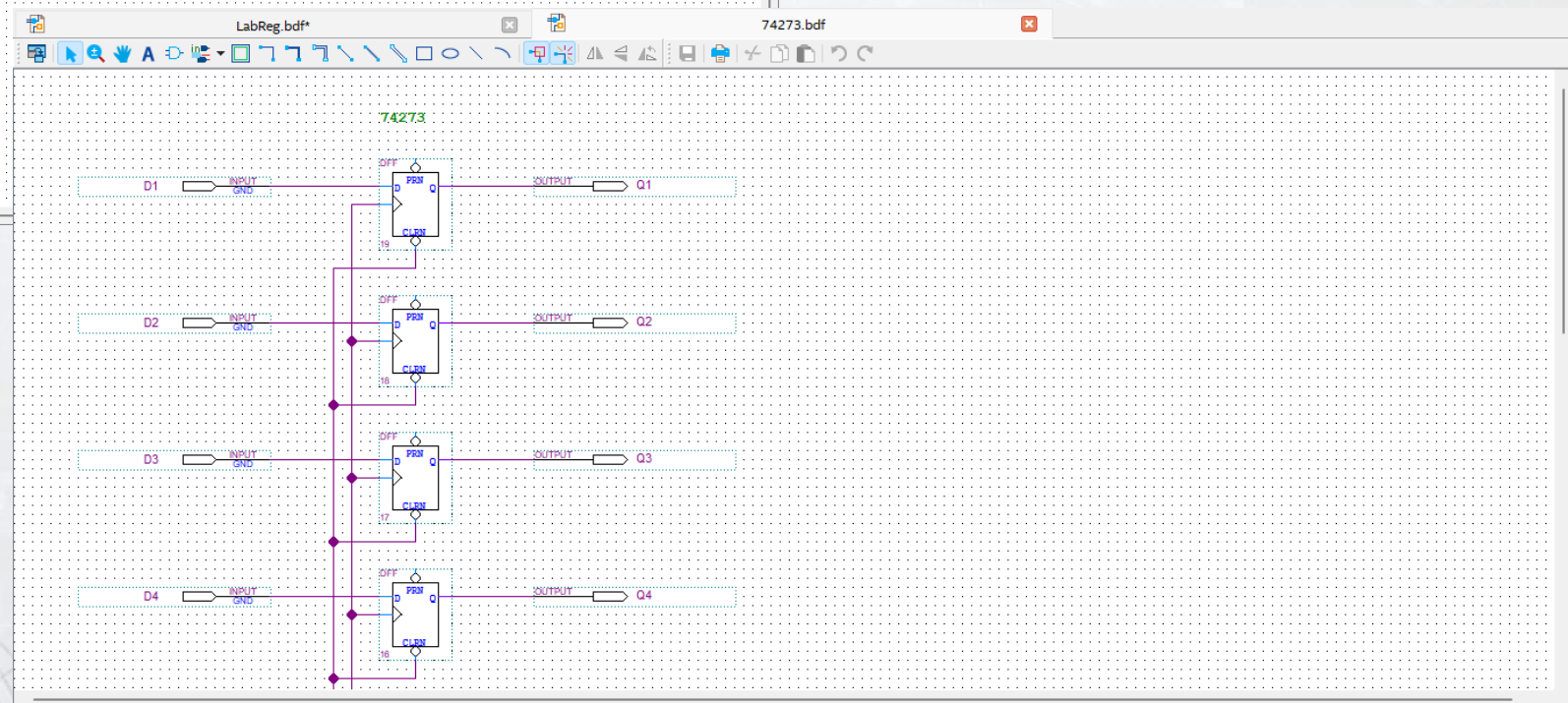
# 可以通过修改元器件库中的74273来实现电路



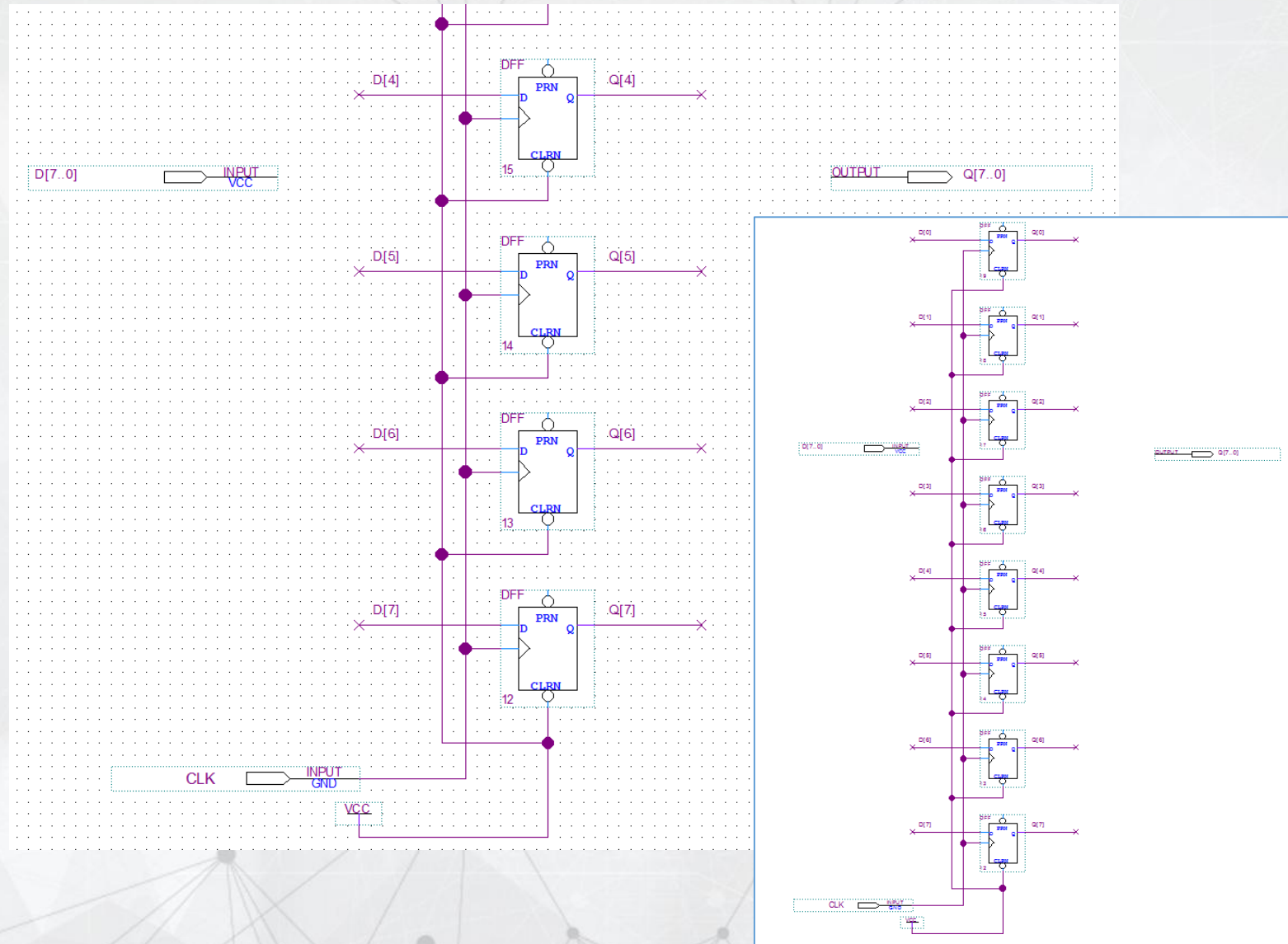
# 修改74273电路



双击74273符号可以打开74273.bdf

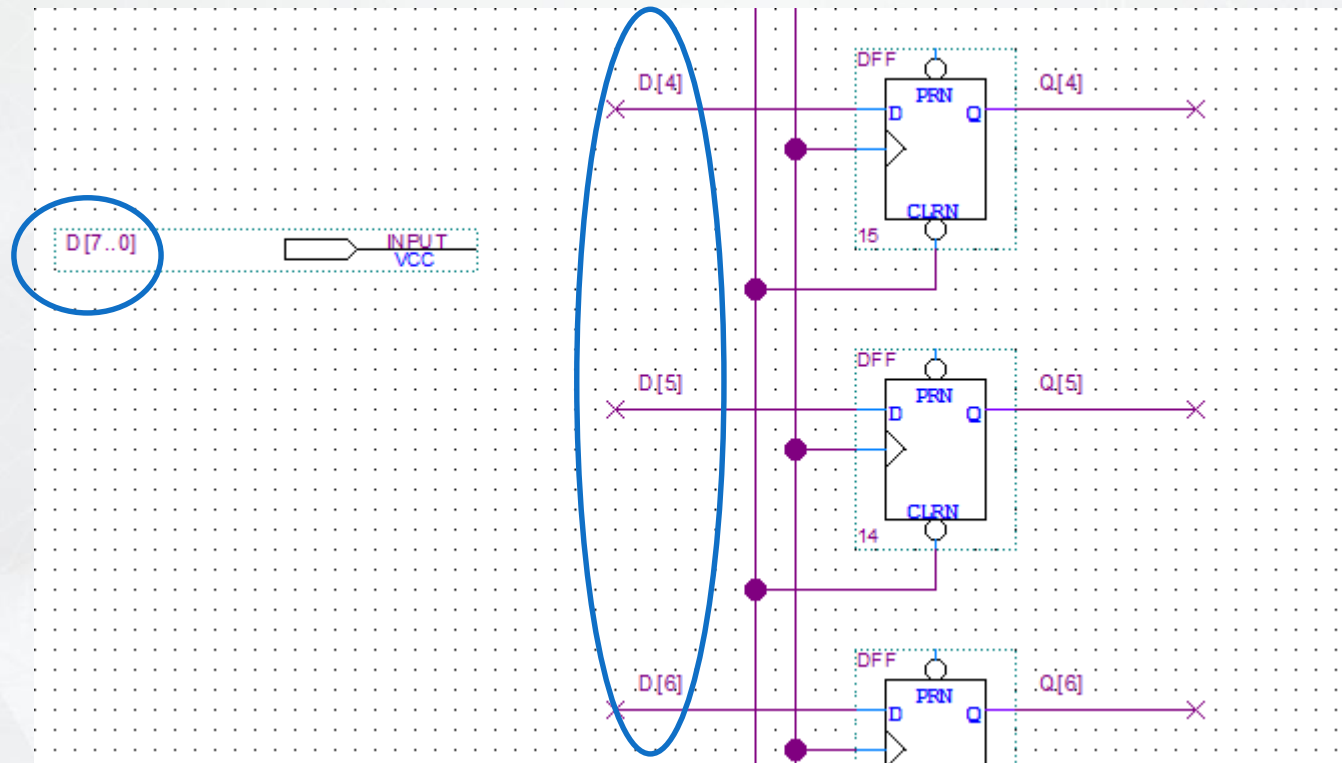


# 完成8位寄存器电路图



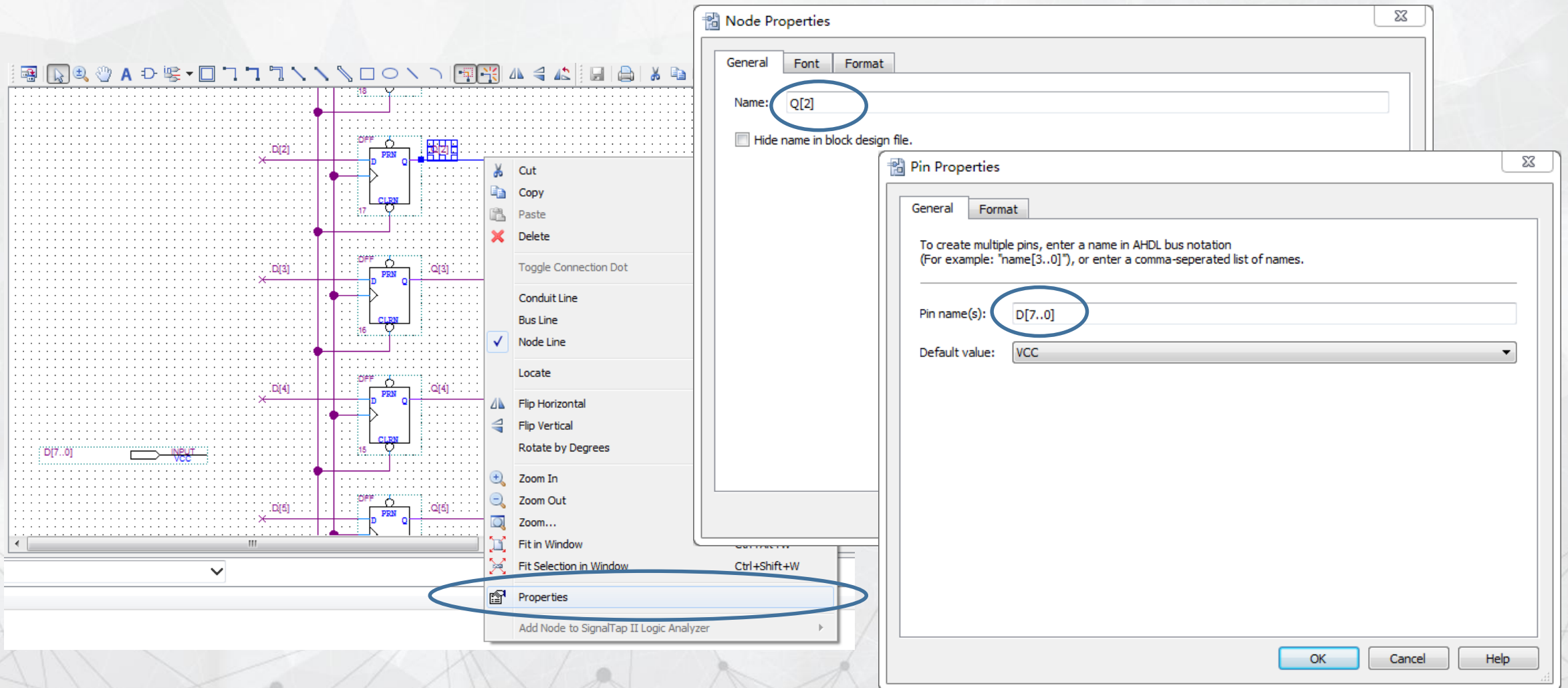
# 通过“网络标号”来连接总线和端口

要通过“网络标号”来连接总线和端口,可以在连线上设置网络标号。  
即利用网络标号,两个具有相同名称的线和端口就连在一起。

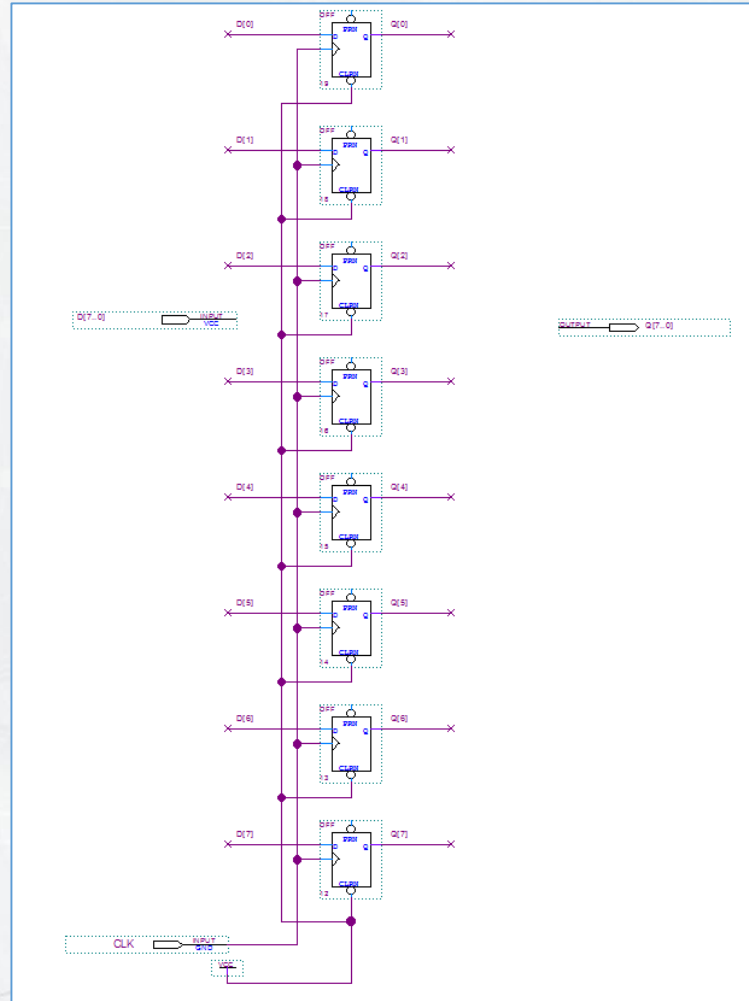


# 修改节点与端口命名

右键点击节点或者输入输出端口，选择“Properties”，在Properties页面修改。



# 将74273.bdf另存为Reg8.bdf（当前工程目录）

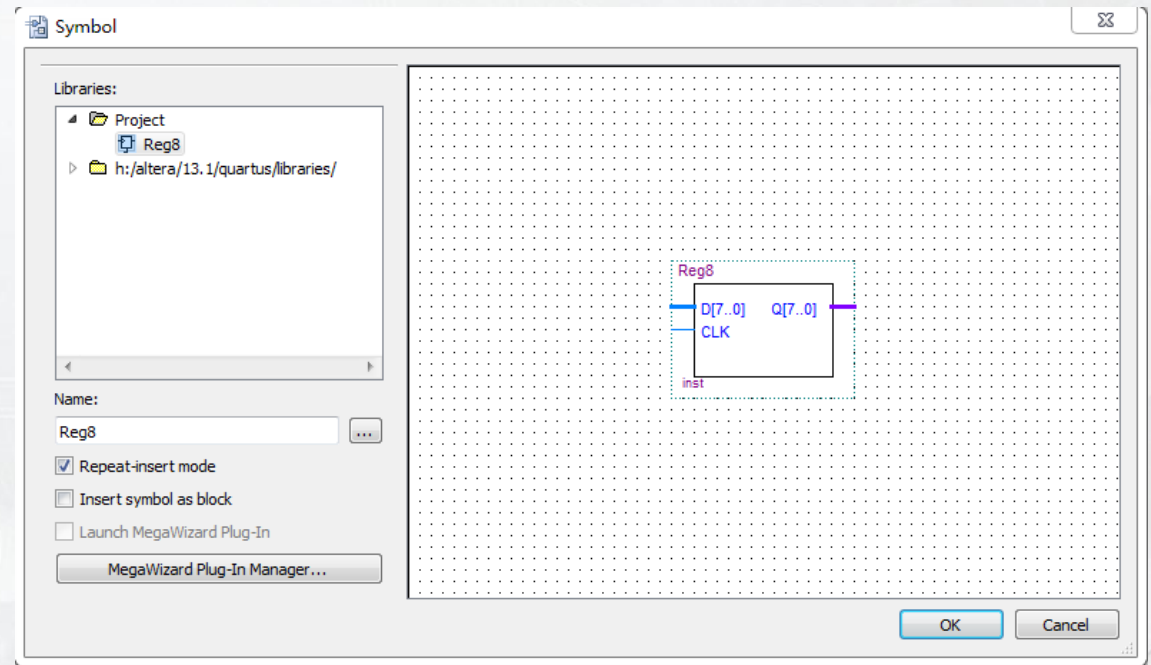


# 自定义8位寄存器元件

1、打开Reg8.bdf，选择主菜单“File”→“Create/Update”项，选择“Create Symbol Files for Current File”，由Reg8.bdf生成Reg8.bsf，即生成自定义8位寄存器元件符号。

2、在元器件库中，在Project目录下可以看到自定义元件Reg8。

3、在设计其他电路时，如果用到Reg8元件，只需要将Reg8.bdf和Reg8.bsf两个文件拷贝到工程目录中，就可以从元器件库中选择Reg8元件加入到电路图中。



# 8位寄存器实验——编译前设置与编译、仿真

- 1、主菜单“Assignmemts”→“Device”项，选择Cyclone IV E系列EP4CE55F23C8芯片
- 2、主菜单“Processing”→“Start Compilation”项,或者工具栏 “ ” ，启动编译。
- 3、建立仿真波形文件：主菜单“File”→“New”项，选择University Program VWF，新建\*.vwf，打开波形编辑器。
- 4、设置仿真时间：主菜单“Edit”→“Set End Time”项。
- 5、添加输入输出端口：波形编辑器窗口主菜单 “Edit”→ “Insert”→“Insert Node or Bus”
- 6、运行仿真：波形编辑器窗口主菜单“Simulation”→“Run Functional Simulation”项。

# 8位寄存器仿真要求

- 1、合理设置时钟脉冲，每次时钟上升沿，向寄存器打入一个数据，打入若干数据，观察输出端口的变化。
- 2、通过波形图分析寄存器的功能。

# 现在开始实验！（第2次课）

## 实验一、基本组合、时序逻辑电路实验

1、2-4译码器：参考教材135页-144页

计数器：参考教材144页-147页

8位数据寄存器：参考教材148页-151页

2、分别建立三个工程，完成电路设计、编译和仿真。按照要求：全面仿真各个电路的功能，自行设计仿真方案。

3、2个人一组。实体名后面加2个学号的后两位，例如mux21a0709

4、验收方式：按组验收。

5、选做内容：利用Verilog语言实现各个电路，完成编译与仿真。

6、下次课预习：

运算器实验：参考教材250页-254页，221页

存储器实验：ROM参考教材254页-260页，RAM参考教材260页-264页