



# 第4章

---

# EDA工具应用初步

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.1 编辑和输入设计文件

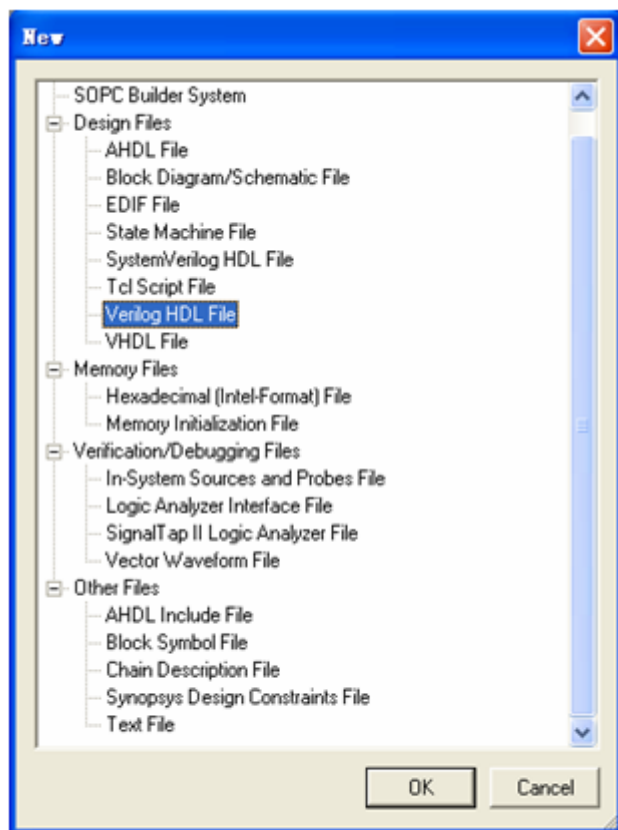


图 4-1 选择编辑文件类型

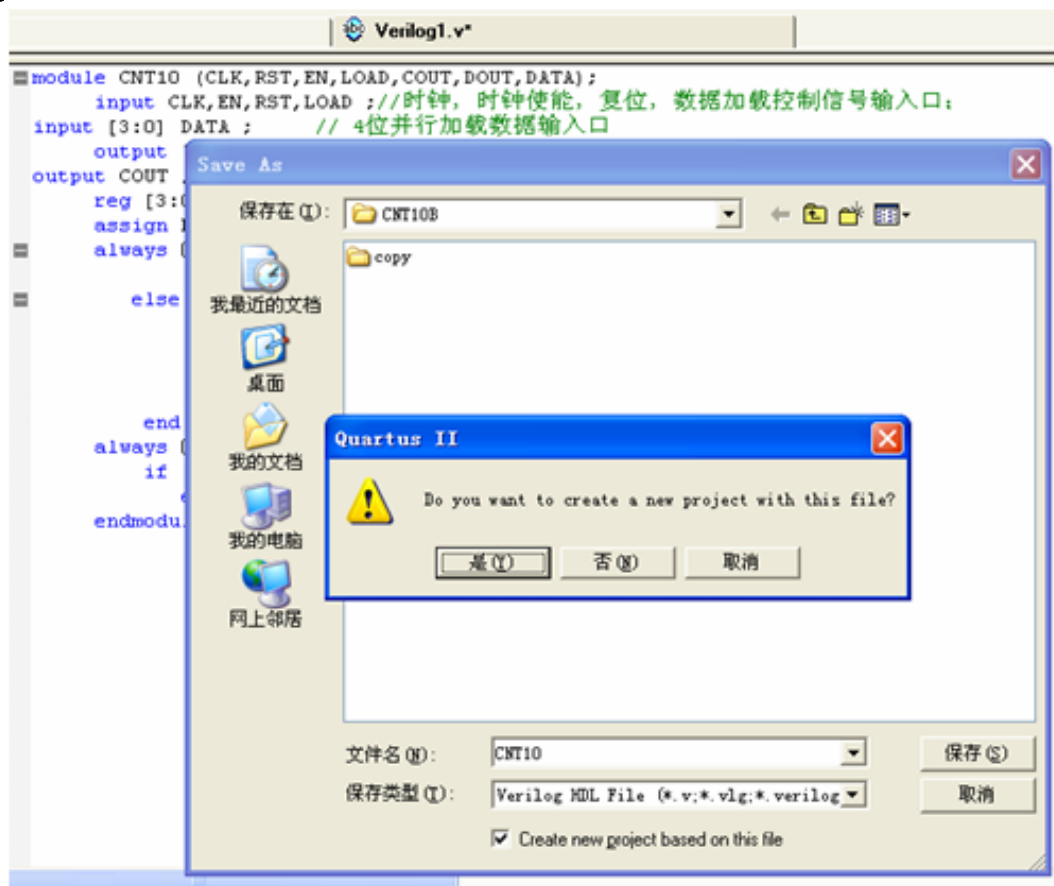


图 4-2 编辑输入源程序并存盘

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.2 创建工程

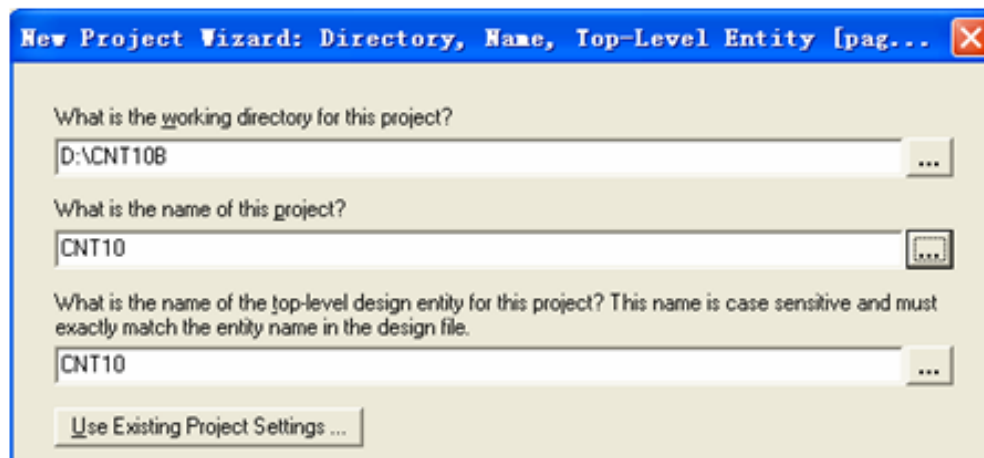


图 4-3 利用 New Project Wizard 创建工程 cnt10

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.2 创建工程

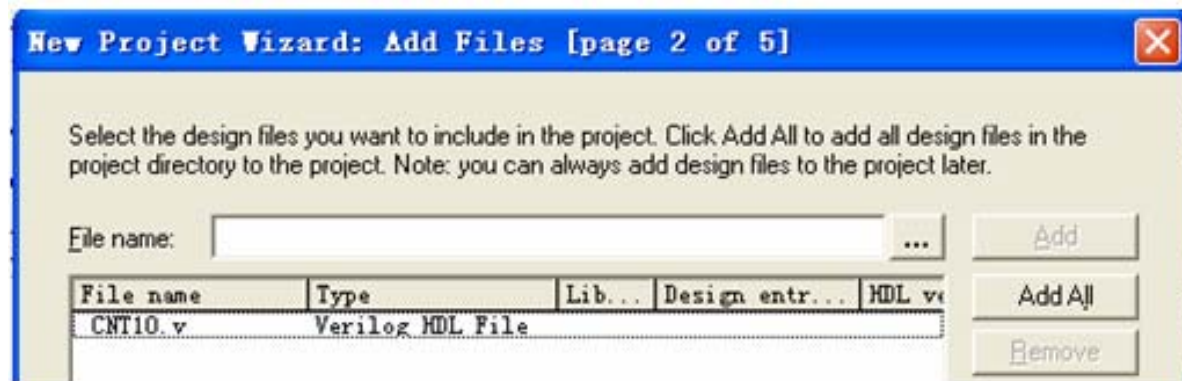


图 4-4 将所有相关的文件都加入进此工程

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.2 创建工程

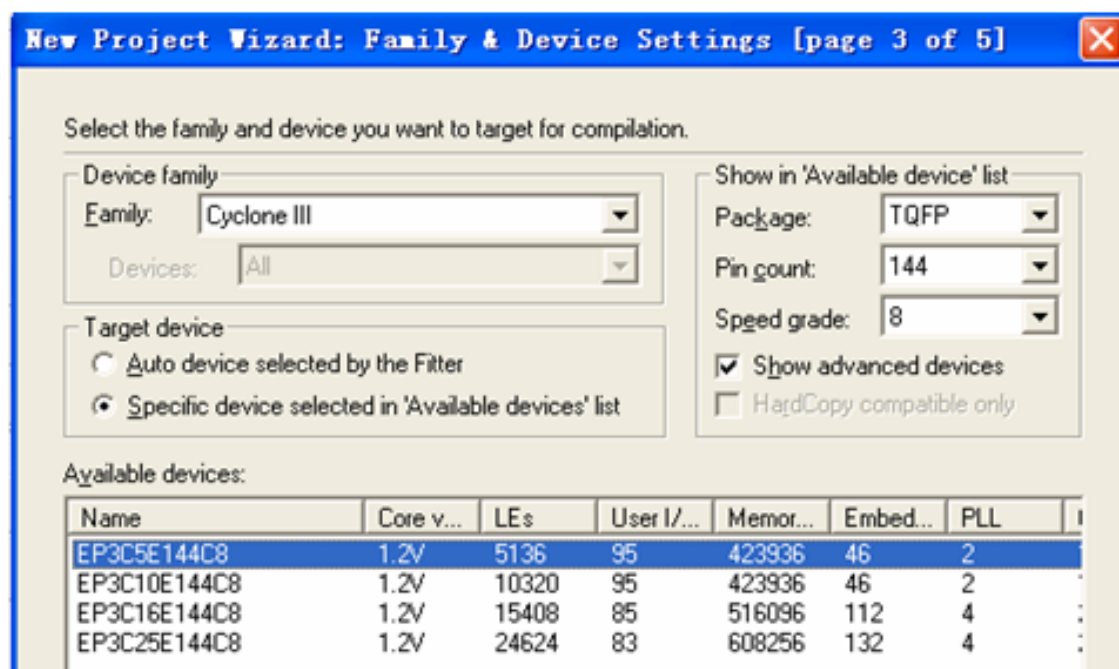


图 4-5 选择目标器件 EP3C5E144C8

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.3 全程编译前约束项目设置

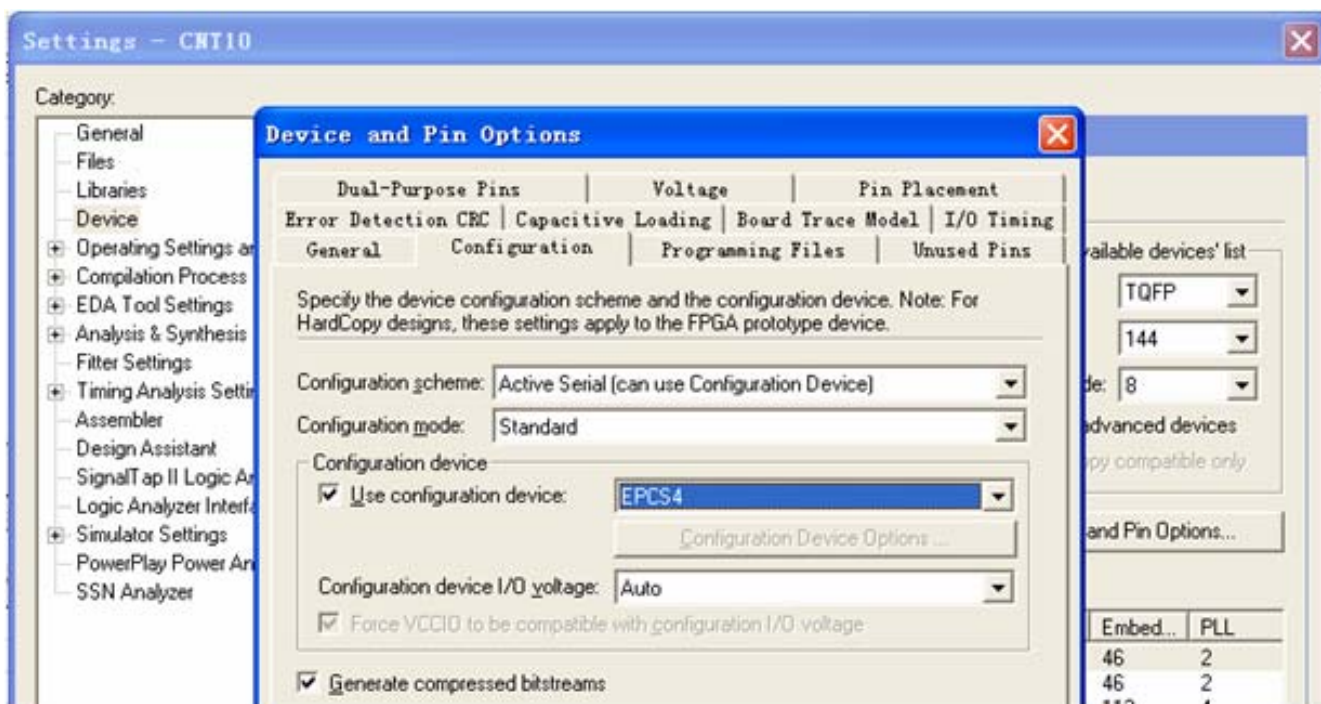


图 4-6 选择配置器件的工作方式

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.3 全程编译前约束项目设置

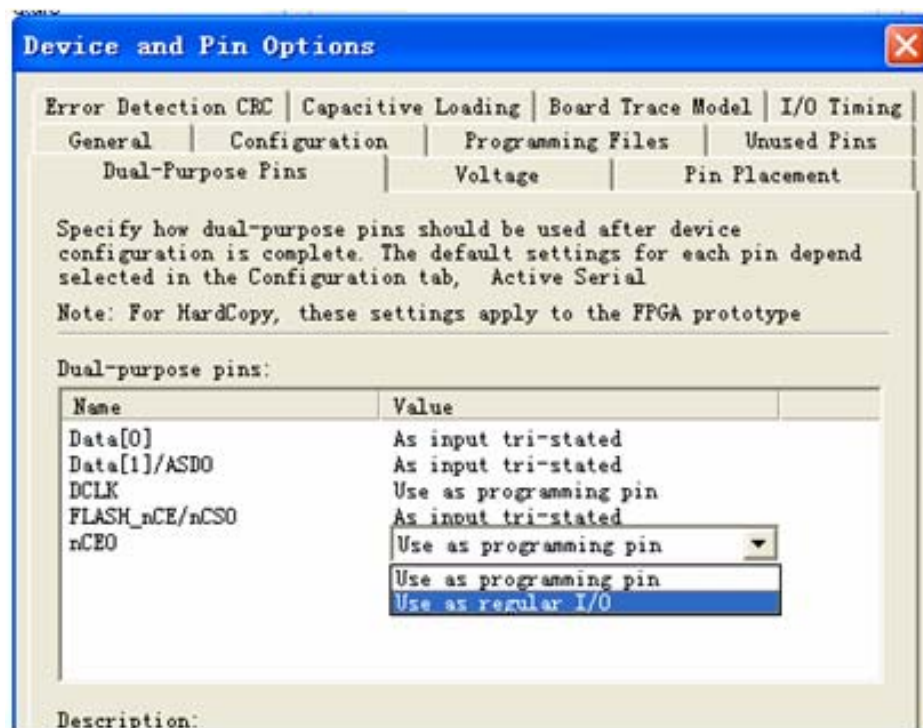


图 4-7 指定 nCEO 脚为普通 I/O pin

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.4 全程综合与编译

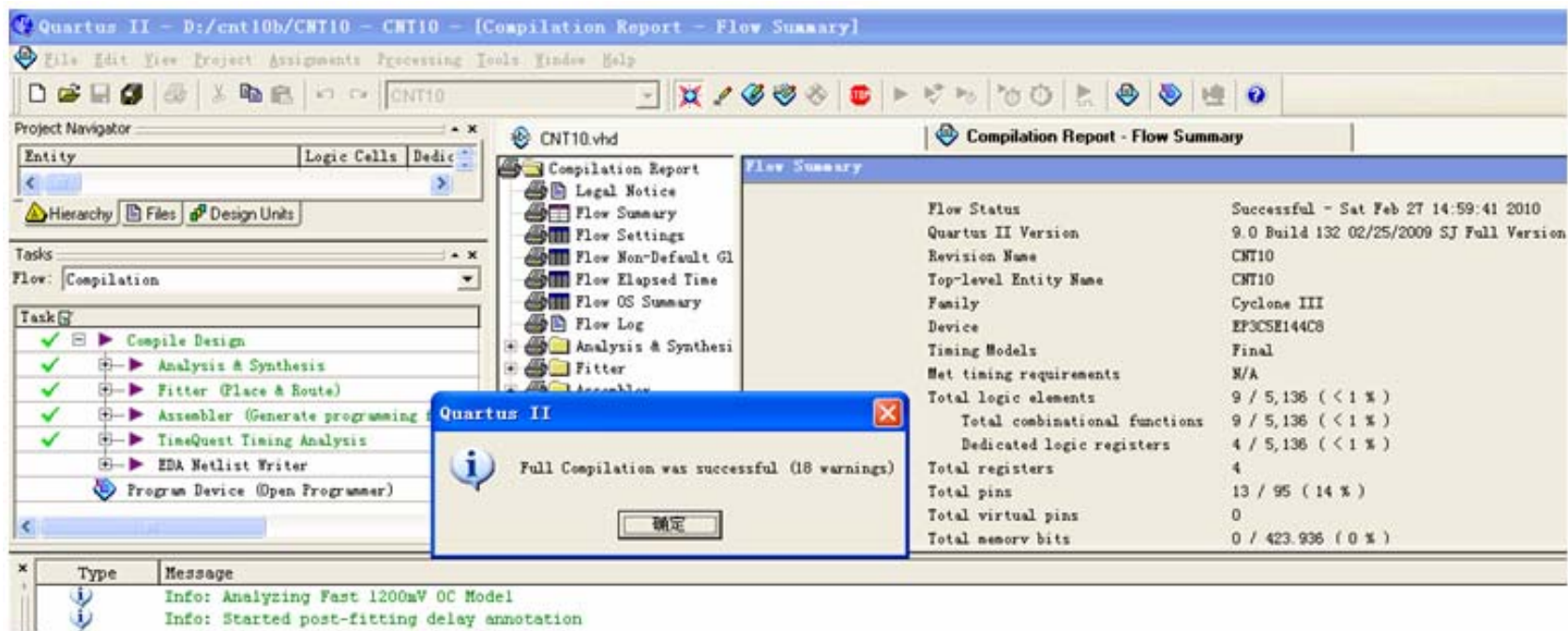


图 4-8 全程编译无错后的报告信息



# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

### (1) 打开波形编辑器

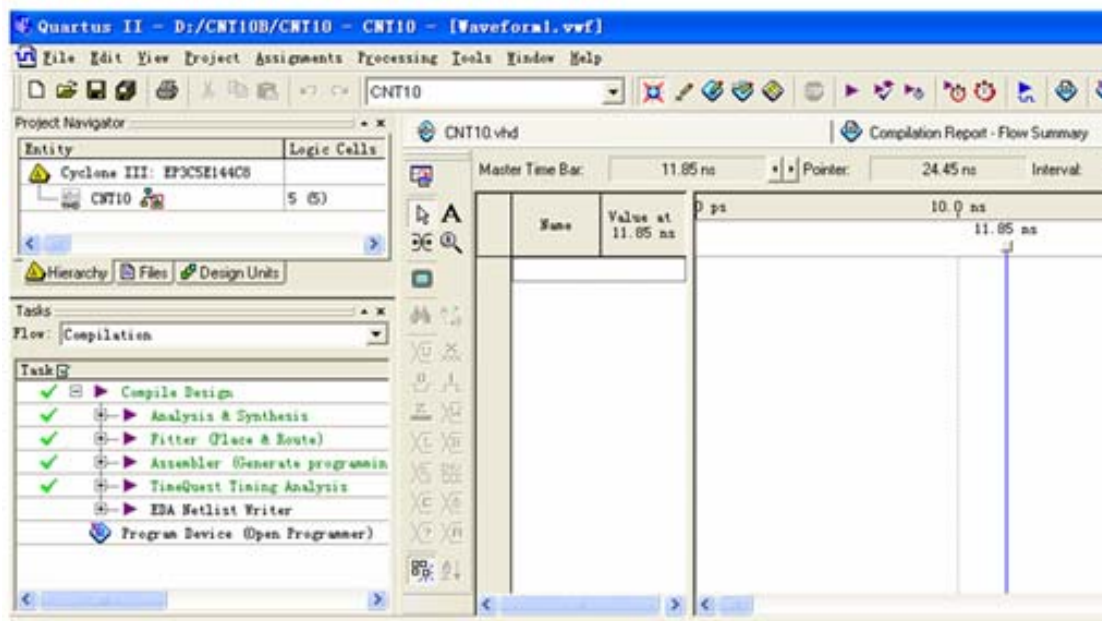


图 4-9 激励信号波形编辑器窗口

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

### (2) 设置仿真时间区域



图 4-10 设置仿真时间长度

### (3) 波形文件存盘

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

(4) 将工程**CNT10**的端口信号节点选入波形编辑器中

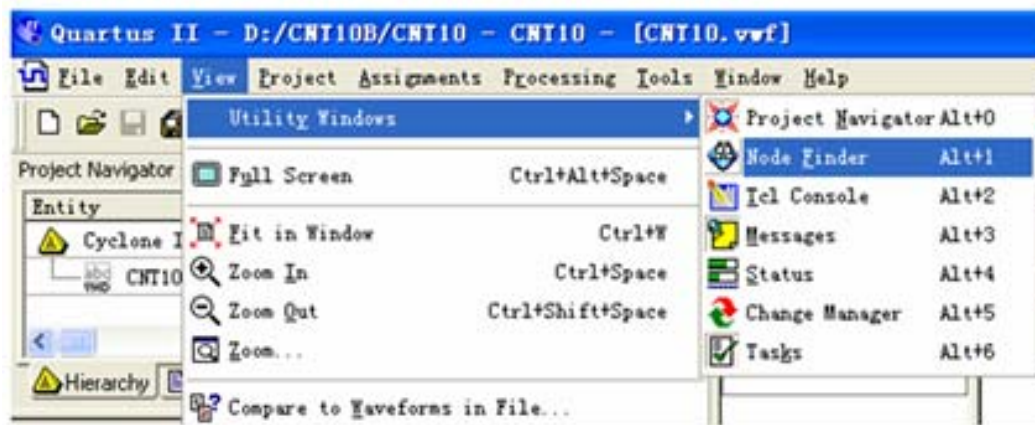


图 4-11 打开信号节点查询窗口

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

(4) 将工程**CNT10**的端口信号节点选入波形编辑器中

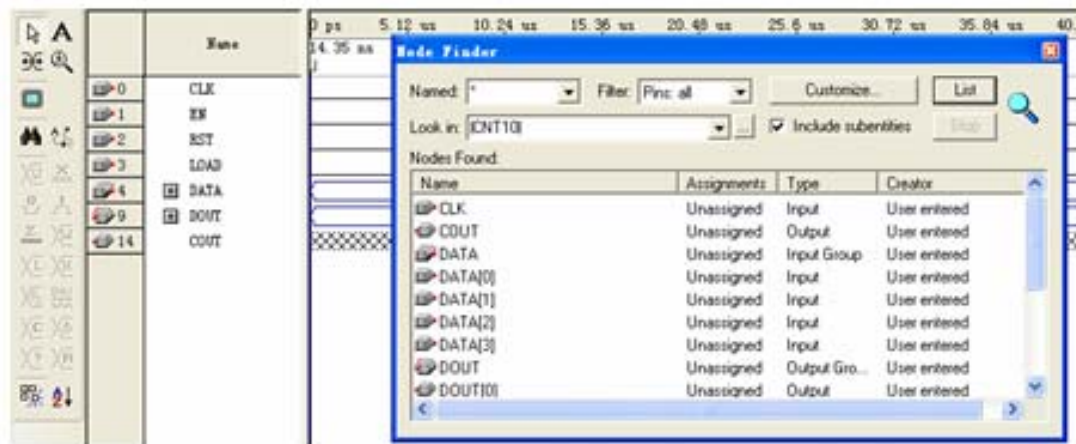


图 4-12 向波形编辑器拖入信号节点

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

### (5) 编辑输入波形(输入激励信号)

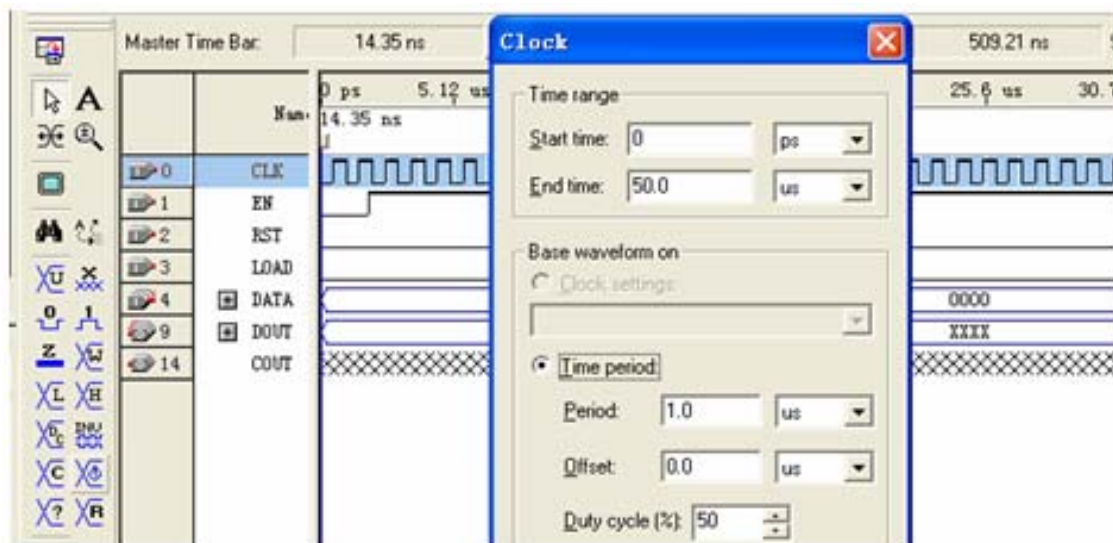


图 4-13 设置时钟 CLK 的周期

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

### (5) 编辑输入波形(输入激励信号)

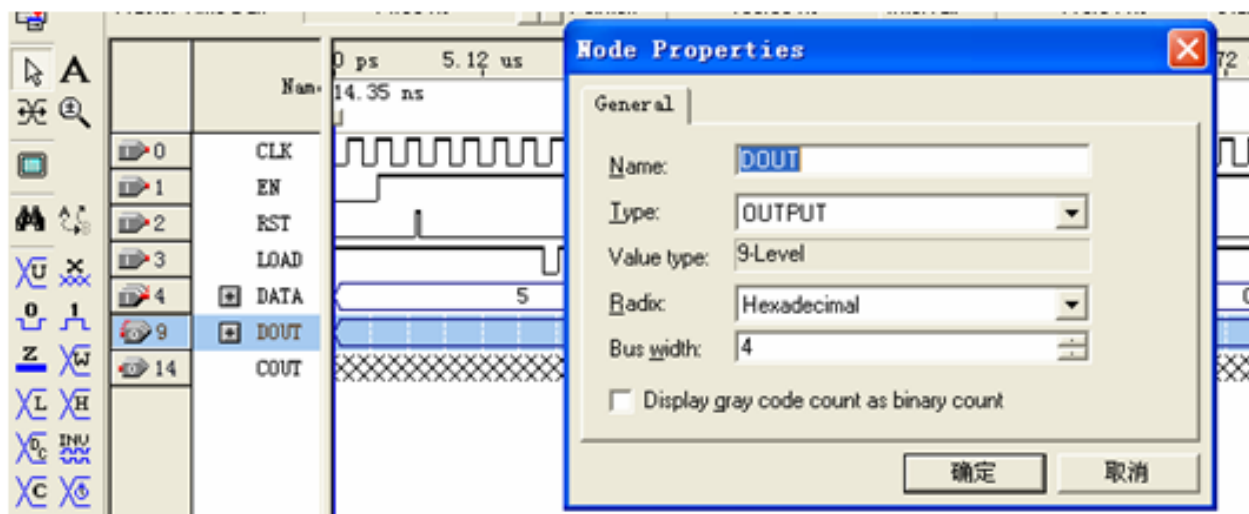


图 4-14 设置好的激励波形图，选择总线数据格式

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

### (6) 总线数据格式设置和参数设置

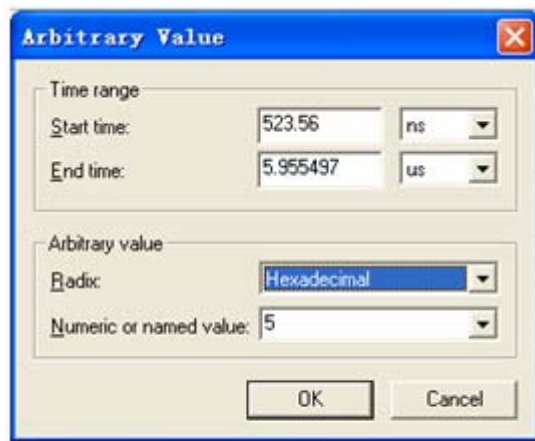


图 4-15 设置输入数据

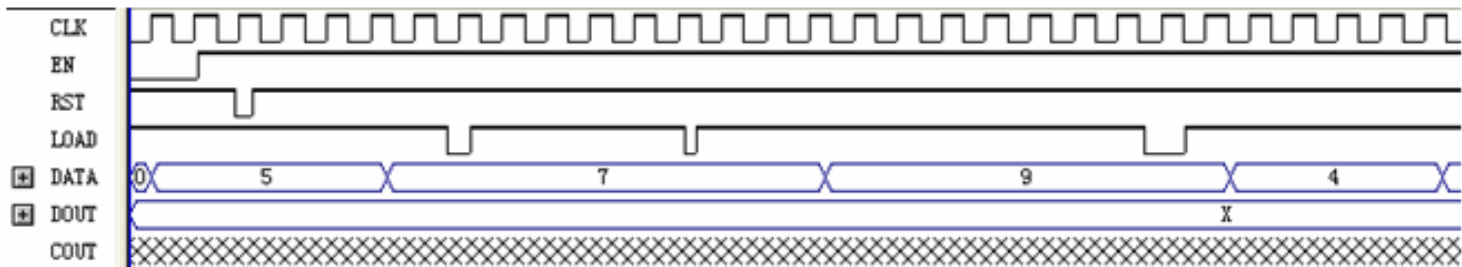


图 4-16 最后设置好的 vwf 仿真激励波形文件图

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

### (7) 仿真器参数设置

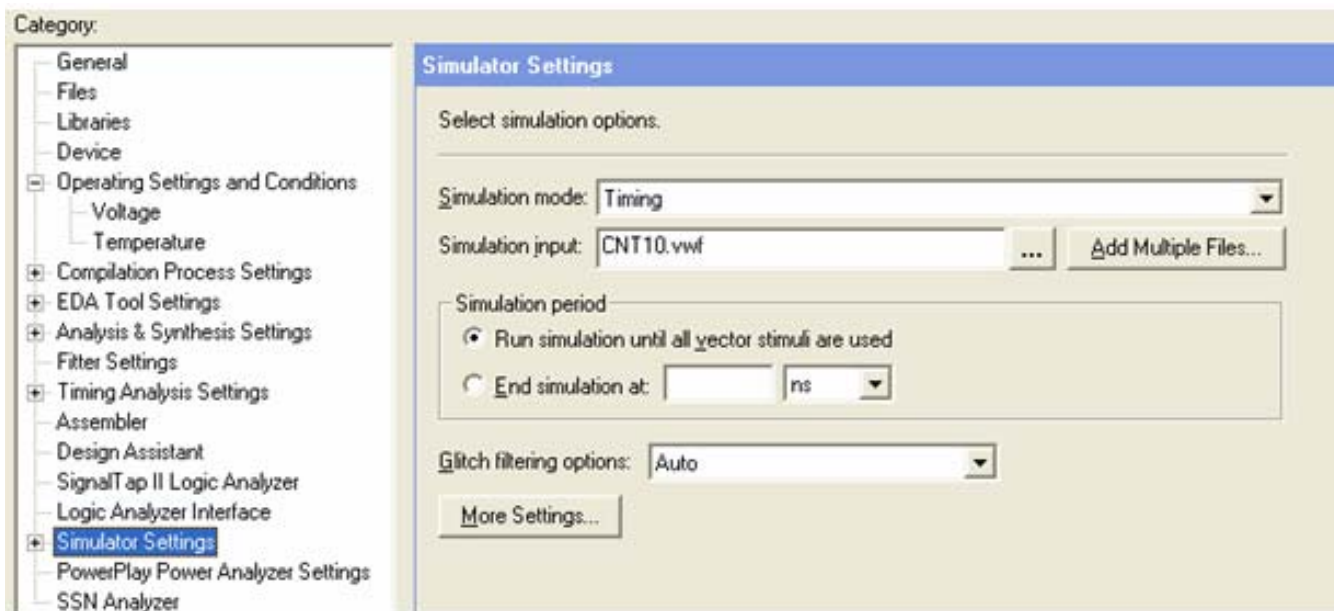


图 4-17 选择仿真模式



# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.5 仿真测试

(8) 启动仿真器

(9) 观察仿真结果

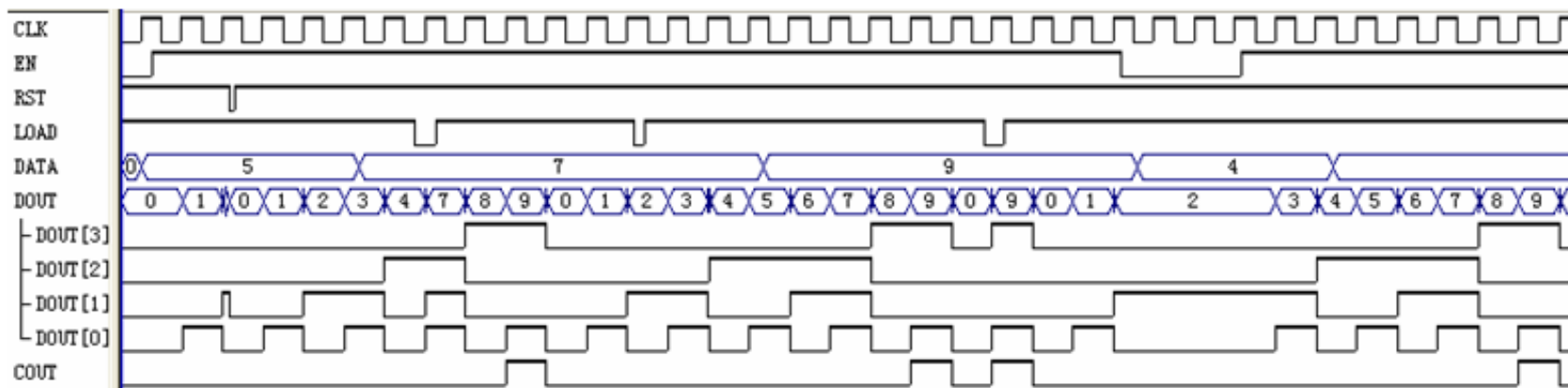


图 4-18 仿真波形输出报告 Simulation Report

# 4.1 硬件逻辑电路的一般设计和测试流程

## 4.1.6 RTL图观察器应用

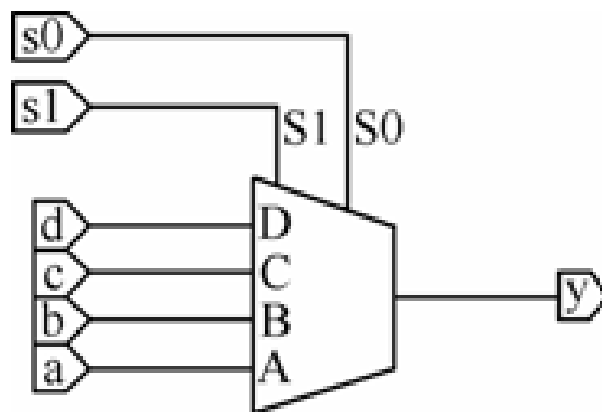


图 3-4 4 选 1 多路选择器

## 4.2 引脚锁定与硬件测试

### 4.2.1 引脚锁定

表 4-1 基于 KX-7C5E+开发板的 EP3C5 开发板的引脚锁定表

信号控制端	拨码 4	拨码 3	拨码 2	拨码 1	键 K5	键 K6	键 K7	键 K8
功能信号设定	D[3]	D[2]	D[1]	D[0]	RST	LOAD	EN	CLK
引脚编号	Pin: 91	Pin: 90	Pin: 89	Pin: 88	Pin: 66	Pin: 67	Pin: 68	Pin: 69
功能信号设定	DOUT3	DOUT2	DOUT1	DOUT0	COUT			
引脚编号	Pin:43	Pin:44	Pin:46	Pin:32	Pin:144			

# 4.2 引脚锁定与硬件测试

## 4.2.1 引脚锁定

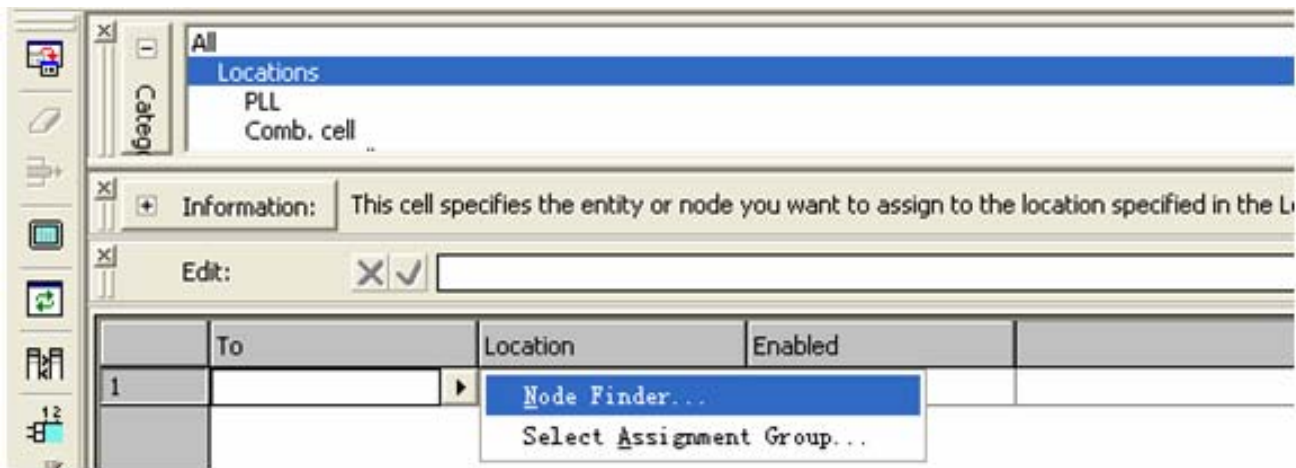


图 4-19 利用 Assignment Editor 编辑器锁定 FPGA 引脚

# 4.2 引脚锁定与硬件测试

## 4.2.1 引脚锁定

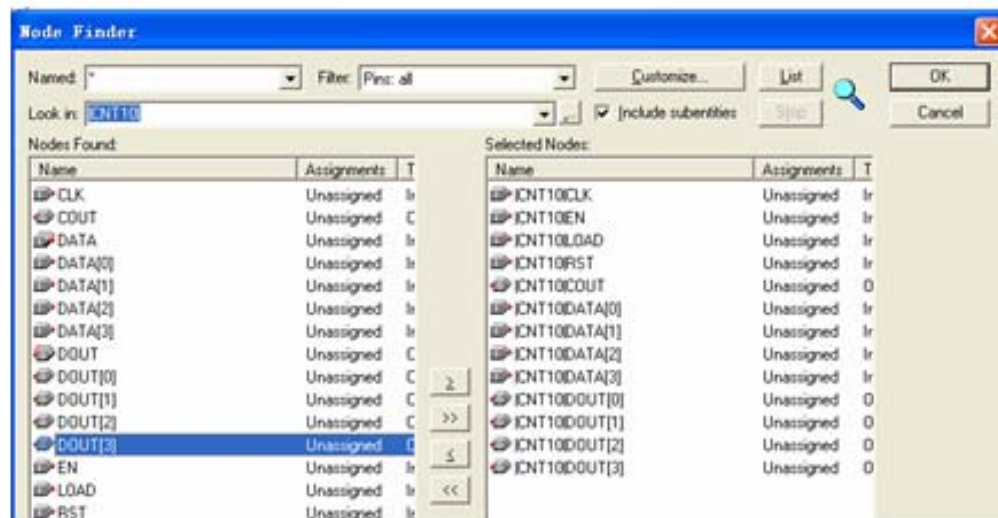


图 4-20 利用 Node Finder 工具选择需要锁定引脚的信号

# 4.2 引脚锁定与硬件测试

## 4.2.1 引脚锁定

	To	Location	Enabled
1	CLK	PIN_69	Yes
2	EN	PIN_68	Yes
3	RST	PIN_66	Yes
4	LOAD	PIN_67	Yes
5	DATA[0]	PIN_88	Yes
6	DATA[1]	PIN_89	Yes
7	DATA[2]	PIN_90	Yes
8	DATA[3]	PIN_91	Yes
9	DOUT[0]	PIN_32	Yes
10	DOUT[1]	PIN_46	Yes
11	DOUT[2]	PIN_44	Yes
12	DOUT[3]	PIN_43	Yes
13	<<new>>		

图 4-21 表格方式引脚锁定窗口

# 4.2 引脚锁定与硬件测试

## 4.2.2 编译文件下载

### (1) 打开编程窗和配置文件

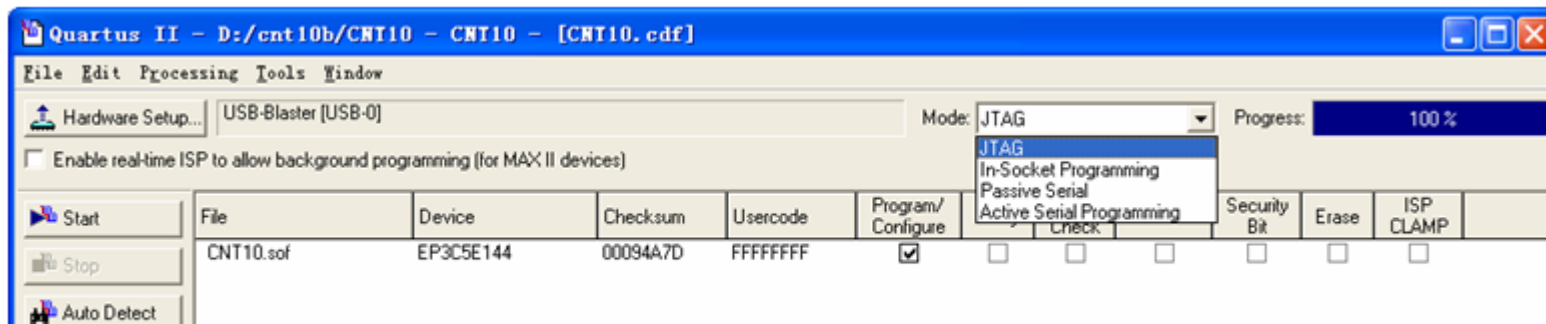


图 4-22 选择编程下载文件

# 4.2 引脚锁定与硬件测试

## 4.2.2 编译文件下载

### (2) 设置编程器



图 4-23 加入编程下载方式

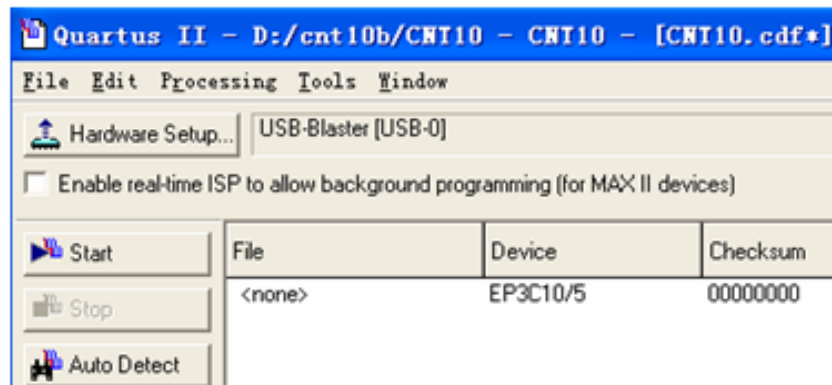


图 4-24 检测实验板上的 FPGA 器件

### (3) 硬件测试



# 4.2 引脚锁定与硬件测试

## 4.2.3 AS直接编程模式

## 4.2.4 JTAG间接编程模式

### 1. 将SOF文件转化为JTAG间接配置文件

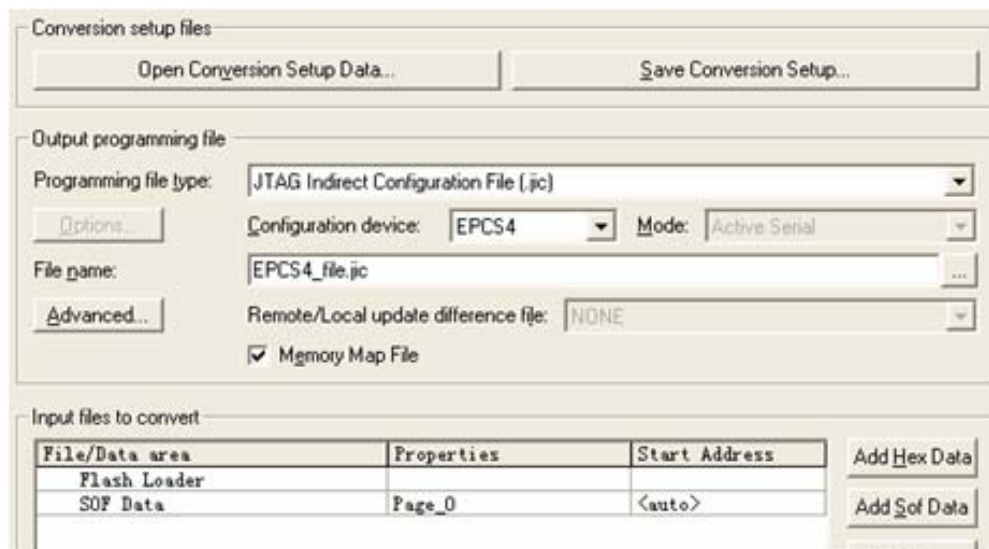


图 4-25 设定 JTAG 间接编程文件

## 4.2 引脚锁定与硬件测试

### 1. 将SOF文件转化为JTAG间接配置文件

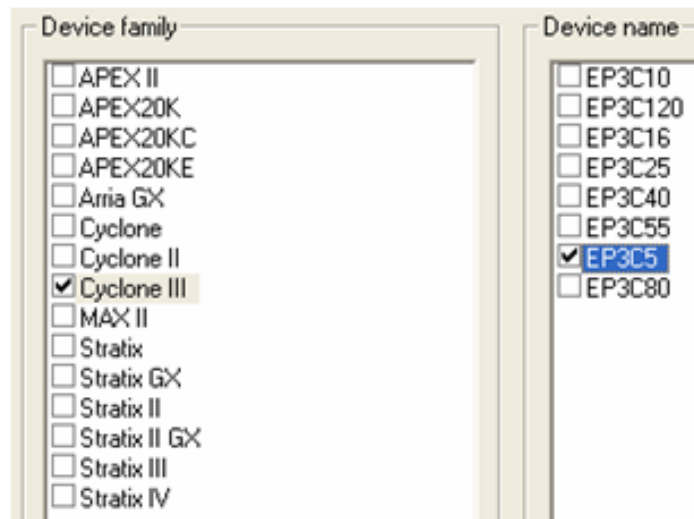


图 4-26 选择目标器件 EP3C5

## 4.2 引脚锁定与硬件测试

### 1. 将SOF文件转化为JTAG间接配置文件

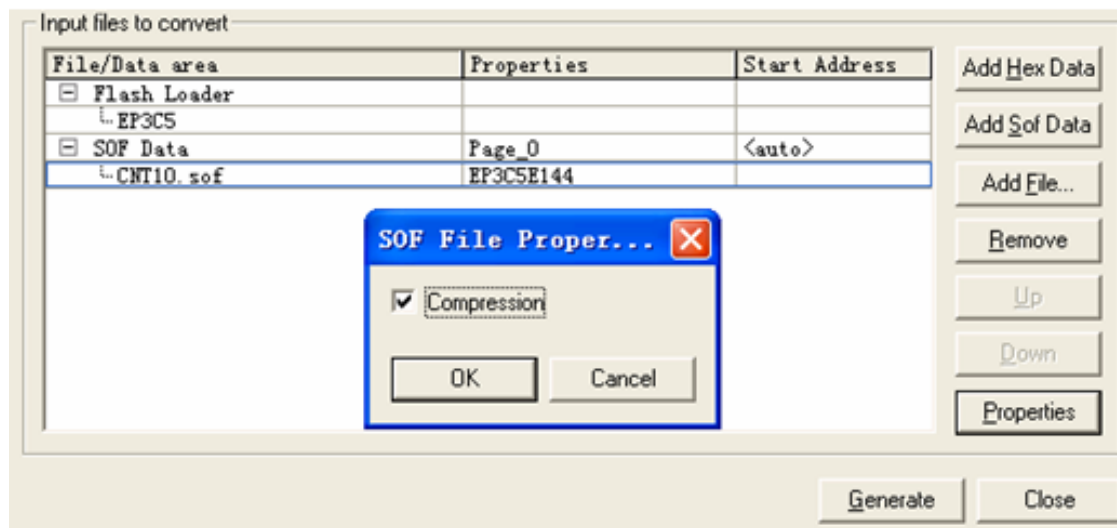


图 4-27 选定 SOF 文件后，选择文件压缩

# 4.2 引脚锁定与硬件测试

## 2. 下载JTAG间接配置文件

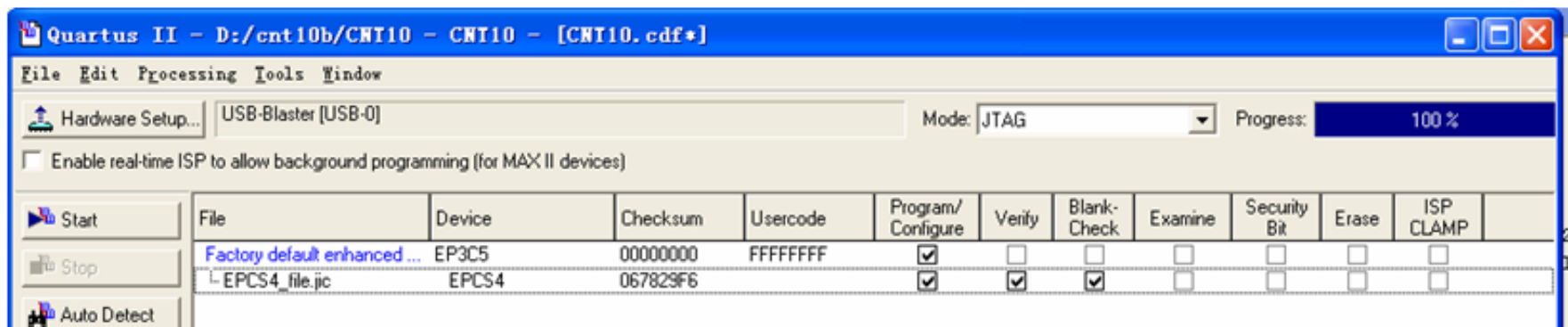


图 4-28 用 JTAG 模式对配置器件 EPCS4 进行间接编程

# 4.2 引脚锁定与硬件测试

## 4.2.5 USB-Blaster编程配置器件使用方法

## 4.2.6 图形方式设置引脚锁定

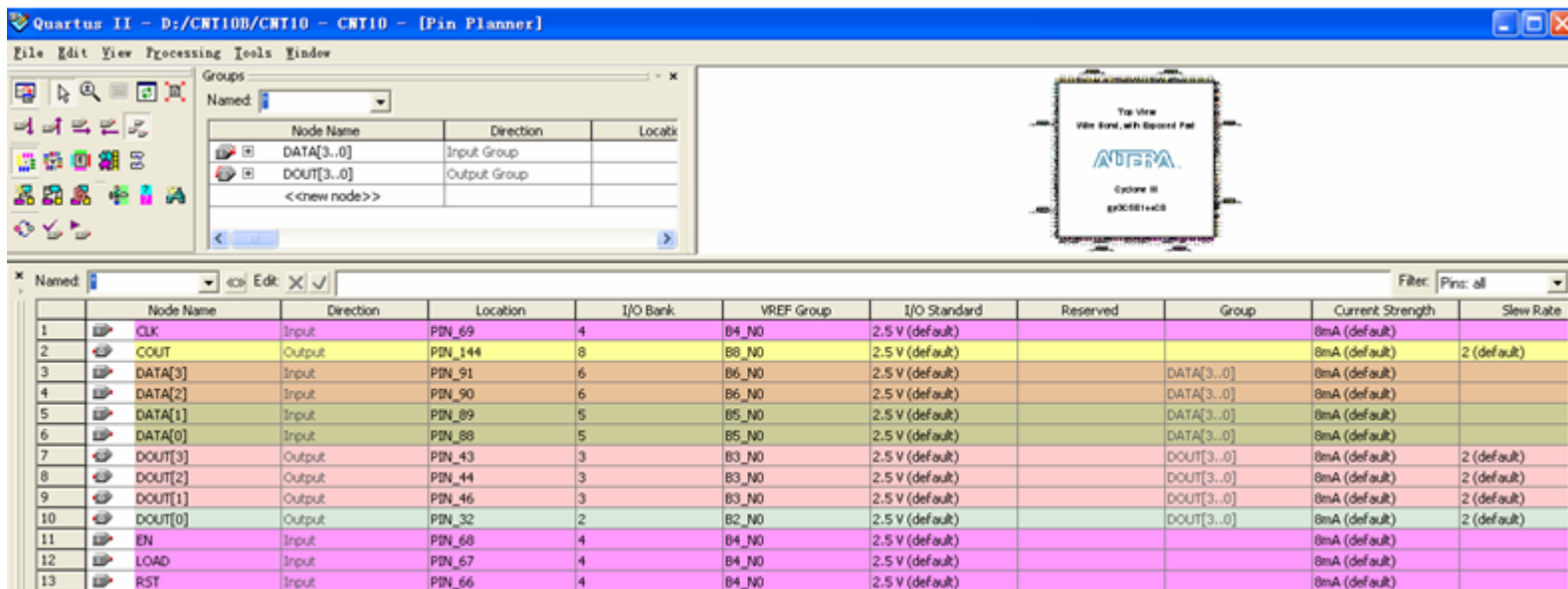


图 4-29 图形方式引脚锁定窗口



## 4.2 引脚锁定与硬件测试

### 4.2.6 利用引脚属性定义方式锁定引脚

#### 【例 4-1】对应基于 EP3C5 的 5E+系统

```
input CLK /* synthesis chip_pin = "69" */ ; //无抖动信号可锁定在 P25
input EN /* synthesis chip_pin = "68" */ ;
input RST /* synthesis chip_pin = "66" */ ;
input LOAD /* synthesis chip_pin = "67" */ ;
input [3:0] DATA /* synthesis chip_pin = "91,90,89,88" */ ;
output [3:0] DOUT /* synthesis chip_pin = "43,44,46,32" */ ;
output COUT /* synthesis chip_pin = "144" */ ;
```

# 4.3 嵌入式逻辑分析仪使用方法

## 1. 打开SignalTap II编辑窗口



图 4-30 SignalTap II 编辑窗口

# 4.3 嵌入式逻辑分析仪使用方法

## 2. 调入待测信号

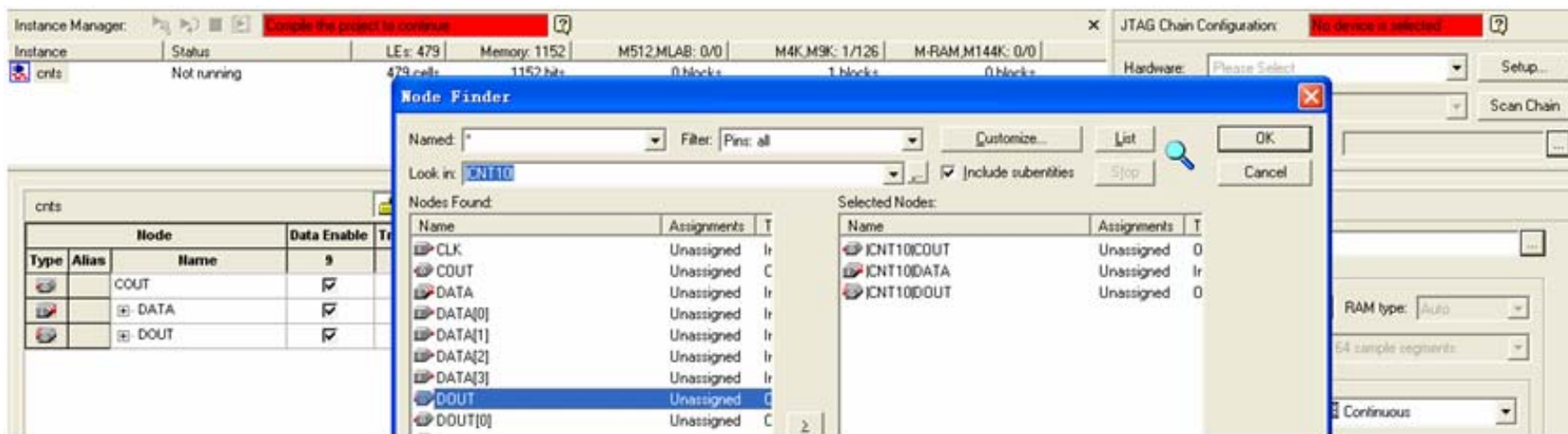


图 4-31 输入逻辑分析仪测试信号



# 4.3 嵌入式逻辑分析仪使用方法

## 3. SignalTap II 参数设置

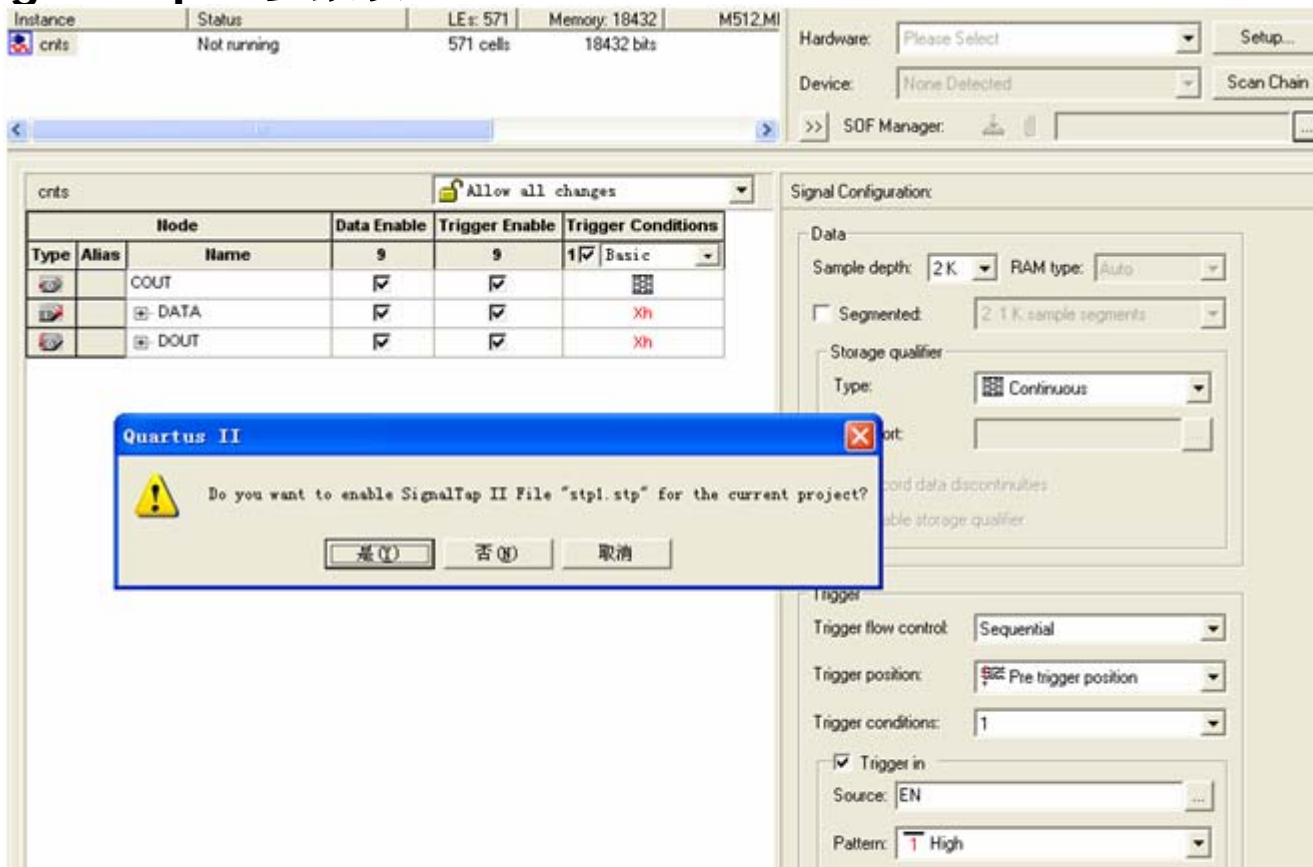


图 4-32 SignalTap II 编辑窗口

# 4.3 嵌入式逻辑分析仪使用方法

## 4. 文件存盘

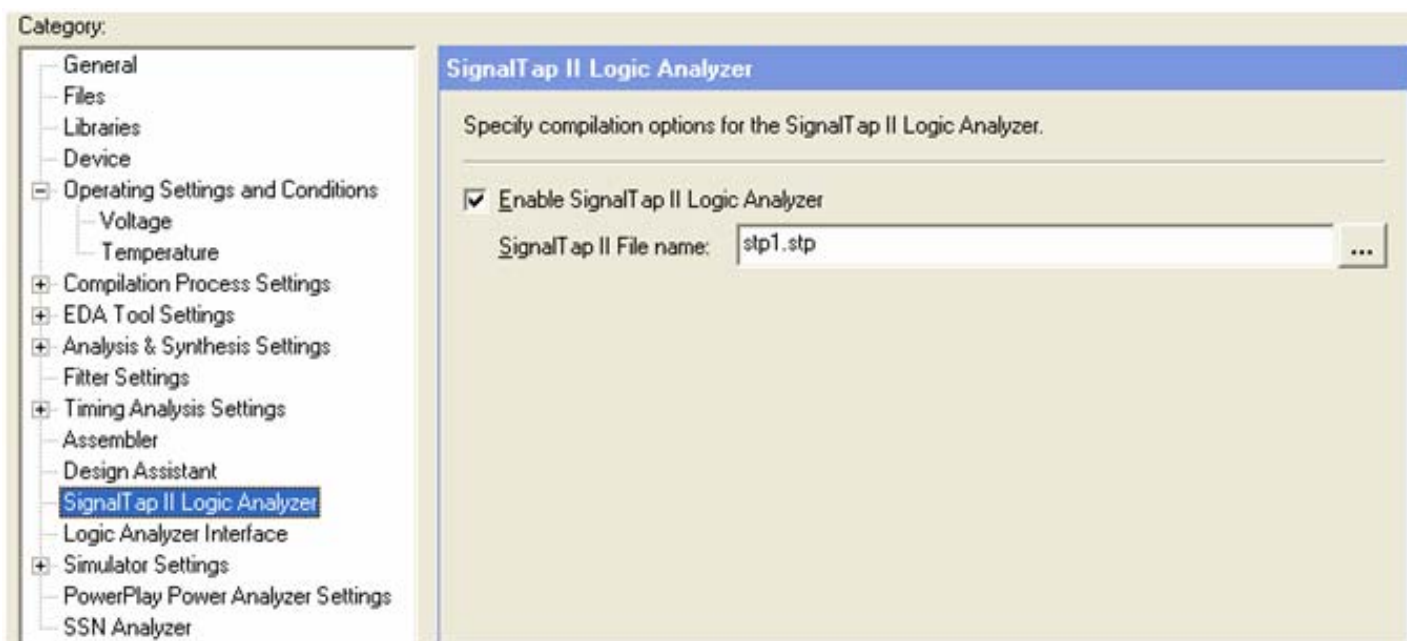


图 4-33 选择或删除 SignalTap II 文件加入综合编译

# 4.3 嵌入式逻辑分析仪使用方法

## 5. 编译下载

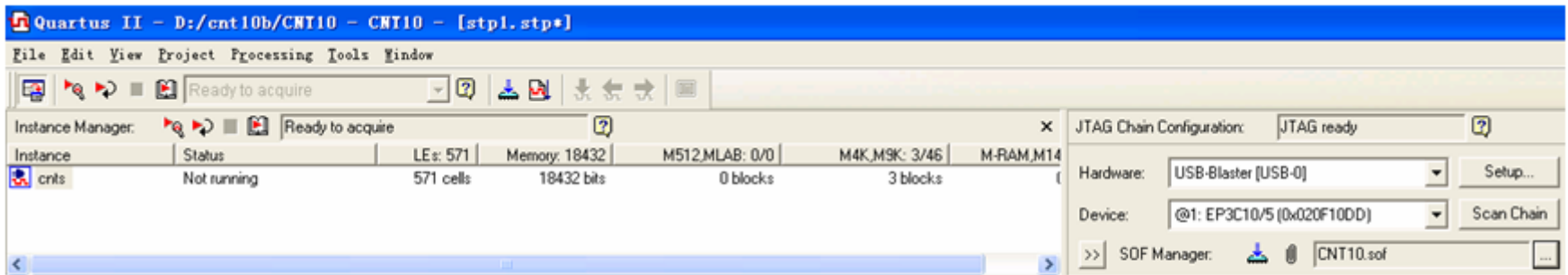


图 4-34 设定 SignalTap II 对 FPGA 的通信接口

# 4.3 嵌入式逻辑分析仪使用方法

## 6. 启动SignalTap II进行采样与分析

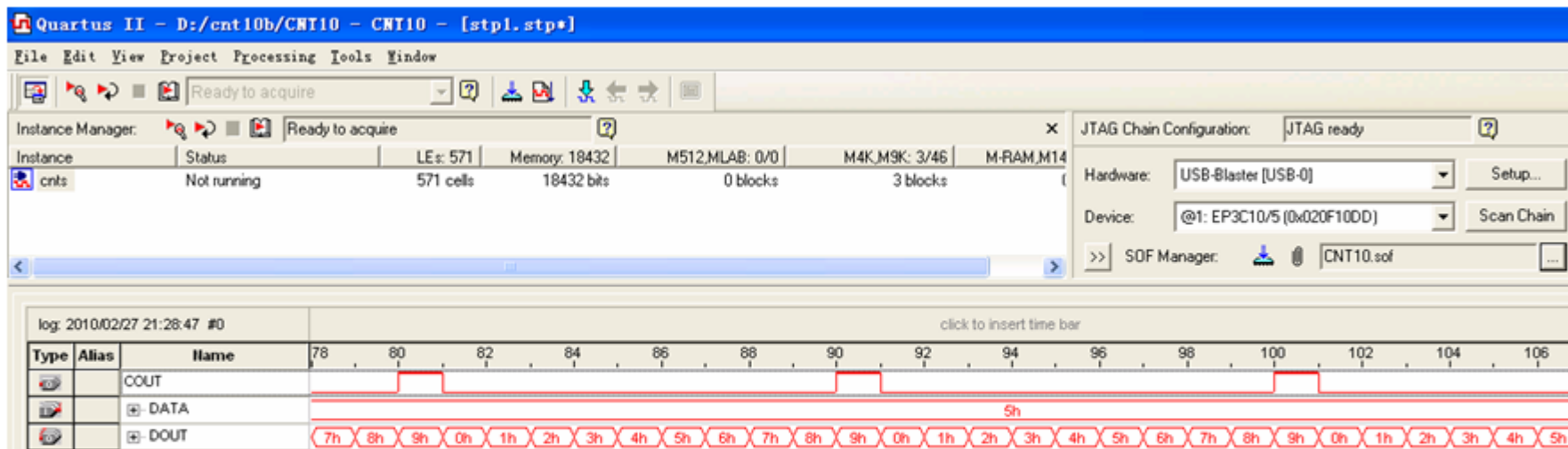


图 4-35 下载 CNT10.sof 并启动 SignalTap II

## 4.3 嵌入式逻辑分析仪使用方法

### 6. 启动SignalTap II进行采样与分析

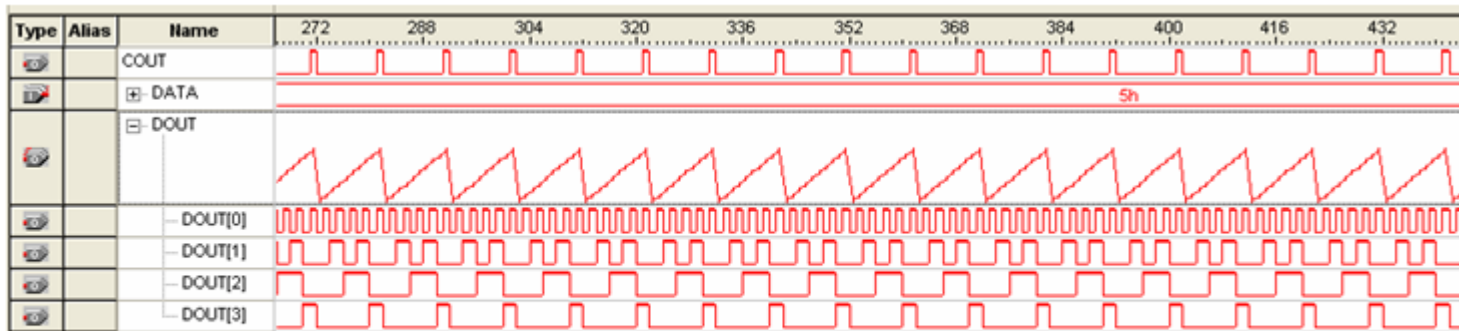


图 4-36 SignalTap II 数据窗口设置后的信号波形

#### 【例 4-2】

```
module CNT10 (CLK, RST, EN, LOAD, COUT, DOUT, DATA, CLK0);  
    input CLK /* synthesis chip_pin = "32" */ ; // 计数器工作时钟  
    input CLK0 /* synthesis chip_pin = "152" */ ; // 逻辑分析仪采样时钟
```

### 7. SignalTap II 的其他设置和控制方法

## 4.4 编辑SignalTap II的触发信号

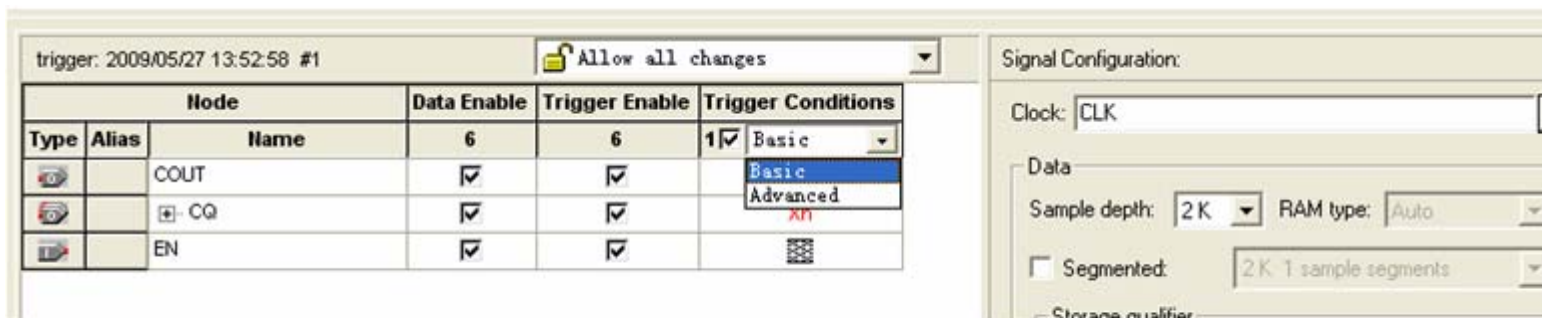


图 4-37 选择高级触发条件

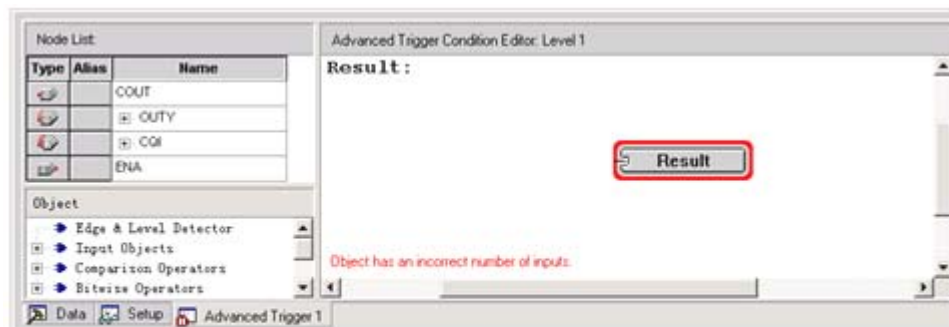


图 4-38 进入触发条件函数编辑窗口

## 4.4 编辑SignalTap II的触发信号

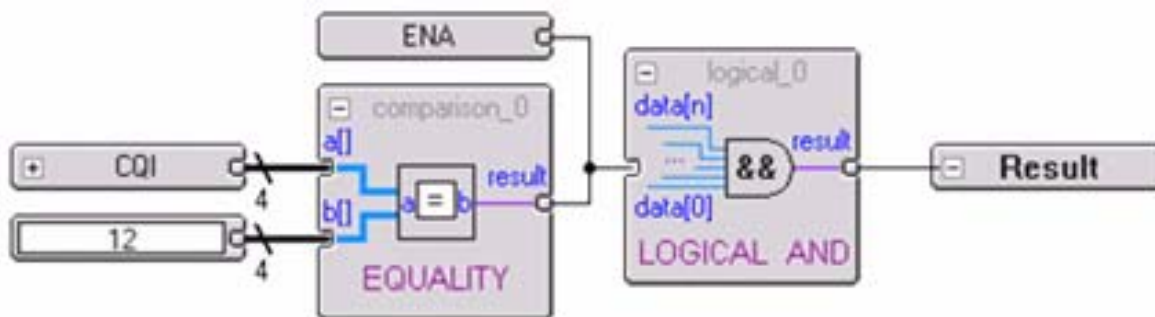


图 4-39 编辑触发函数

# 4.5 原理图编辑输入设计流程

## 4.5.1 基于原理图的层次化设计流程

1. 为本项工程设计建立文件夹
2. 建立原理图文件工程和仿真
  - (1) 打开原理图编辑窗
  - (2) 建立一个初始原理图



图 4-40 选择打开元件输入窗



# 4.5 原理图编辑输入设计流程

## 2. 建立原理图文件工程和仿真

### (2) 建立一个初始原理图

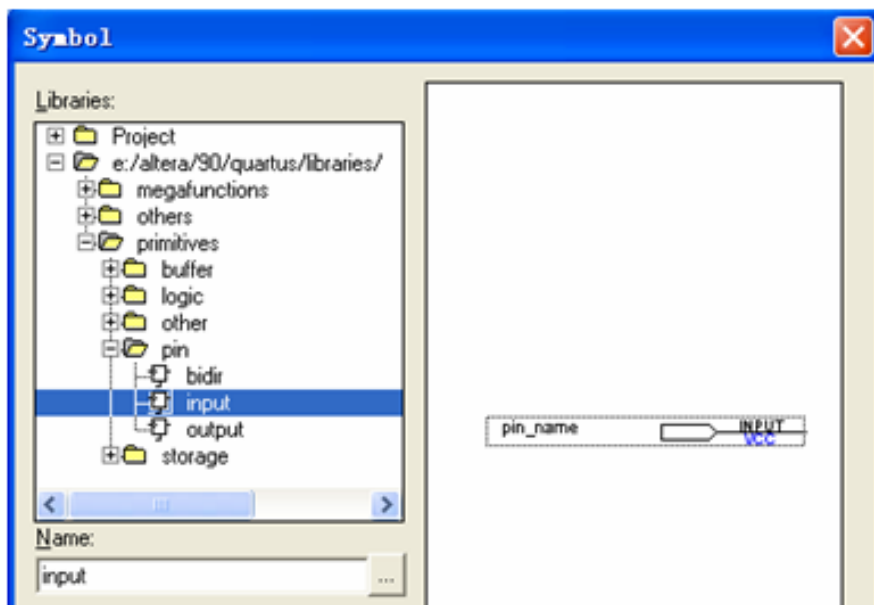


图 4-41 在元件输入对话框输入引脚

# 4.5 原理图编辑输入设计流程

## 2. 建立原理图文件工程和仿真

(3) 原理图文件存盘

(4) 建立原理图文件为顶层设计的工程

(5) 绘制半加器原理图

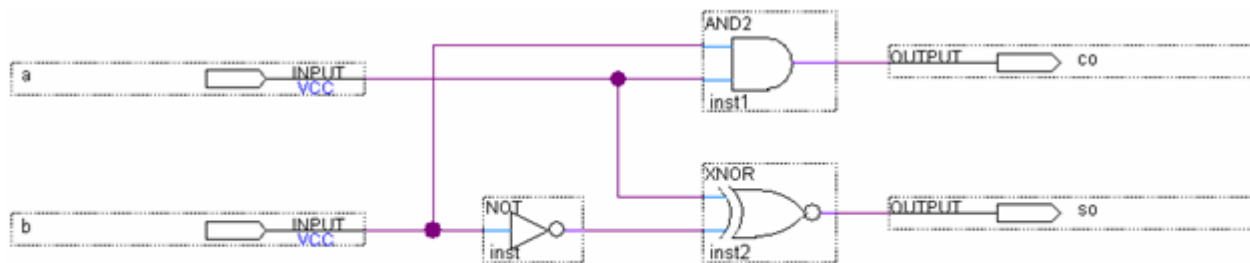


图 4-42 半加器原理图

# 4.5 原理图编辑输入设计流程

## 2. 建立原理图文件工程和仿真

### (6) 仿真测试半加器

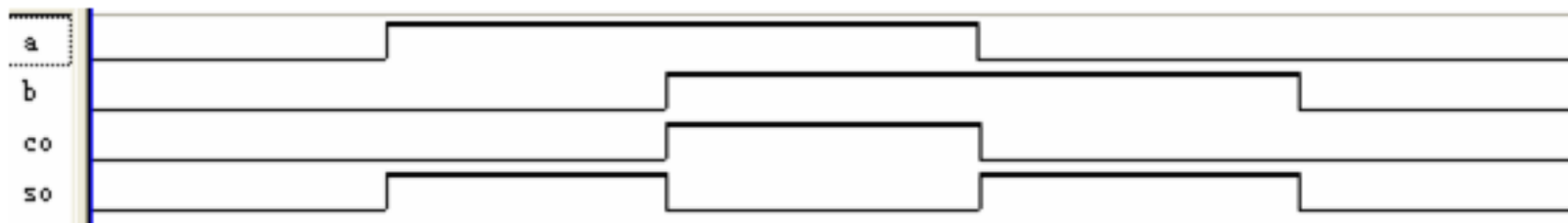


图 4-43 半加器仿真波形

# 4.5 原理图编辑输入设计流程

## 3. 将设计项目设置成可调用的元件

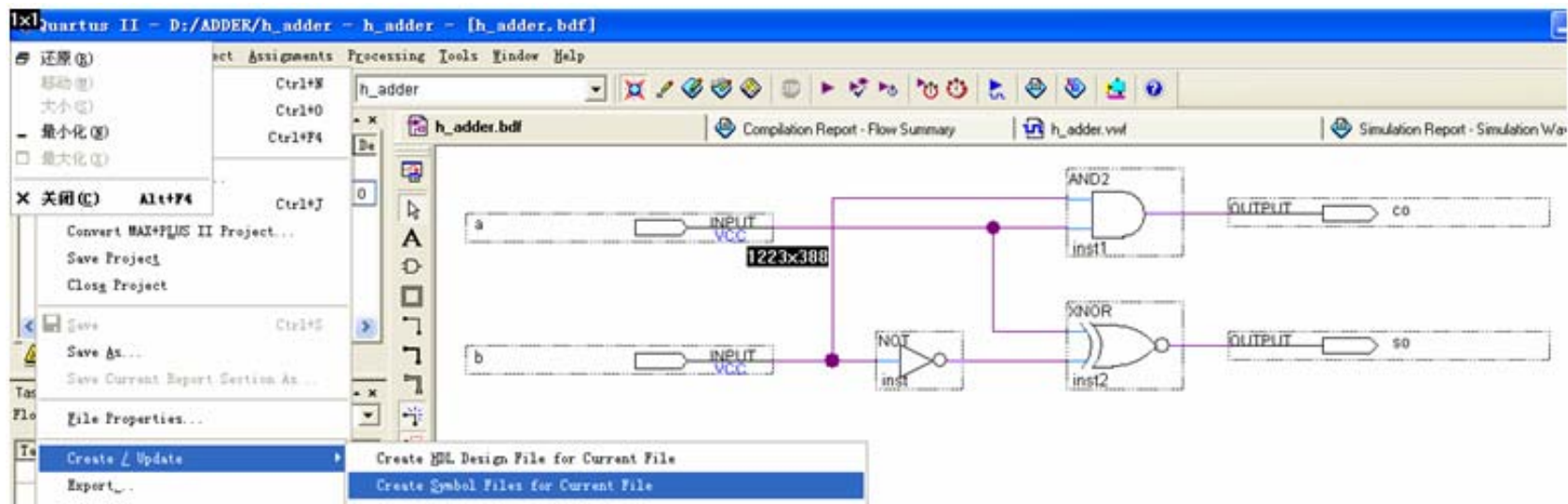


图 4-44 将半加器封装成一个元件，以便在更高层设计中调用

# 4.5 原理图编辑输入设计流程

## 4. 设计全加器顶层文件



图 4-45 全加器 f\_adder.bdf 工程设置

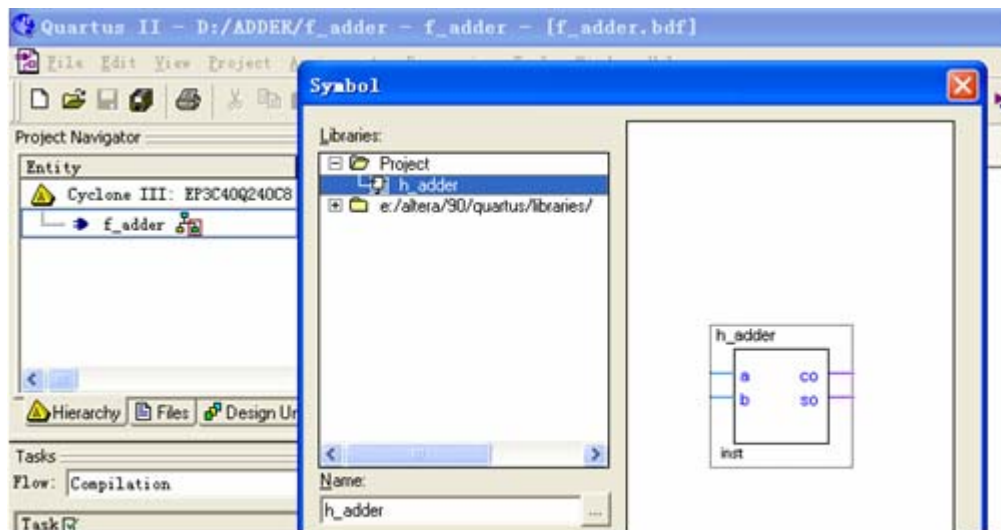


图 4-46 在 f\_adder 工程下的原理图编辑窗中加入半加器

# 4.5 原理图编辑输入设计流程

## 4. 设计全加器顶层文件

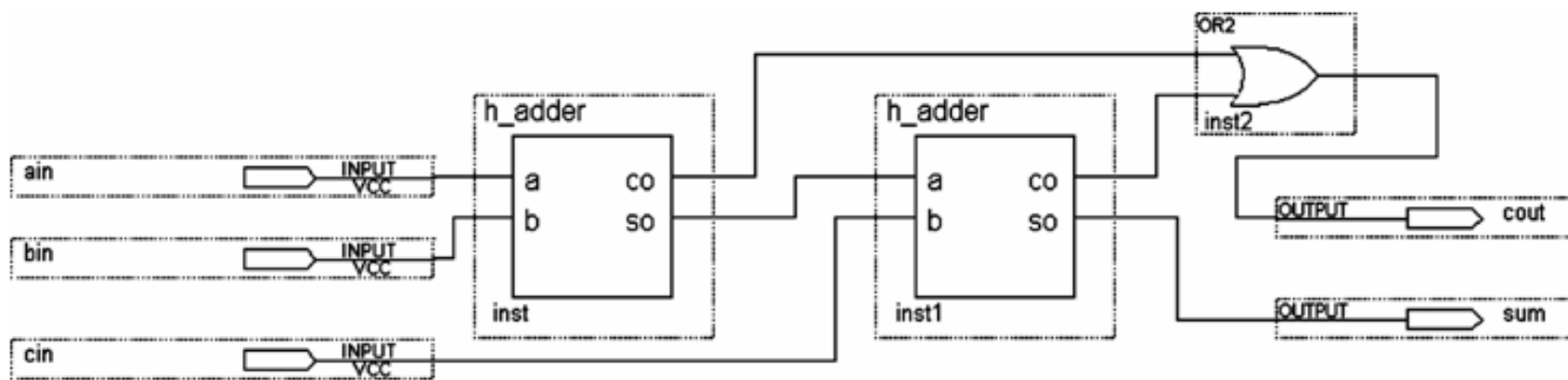


图 4-47 连接好的全加器原理图 f\_adder.bdf

# 4.5 原理图编辑输入设计流程

## 5. 将设计项目进行时序仿真

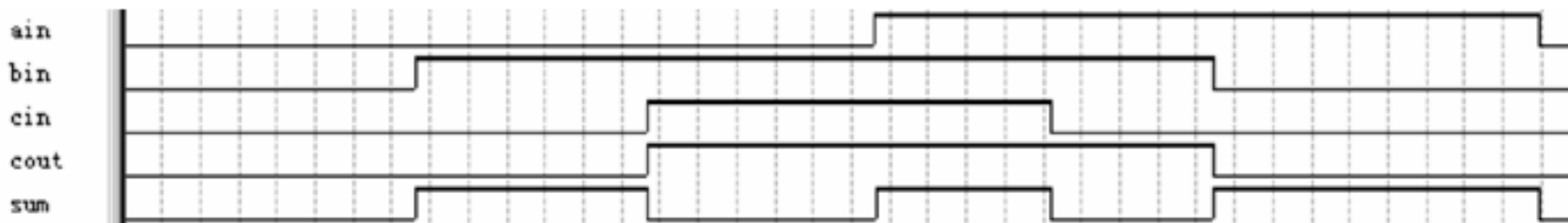


图 4-48 全加器工程 f\_adder 的仿真波形

# 4.5 原理图编辑输入设计流程

## 4.5.2 应用宏模块设计频率计

### 1. 计数器设计

#### (1) 设计电路原理图

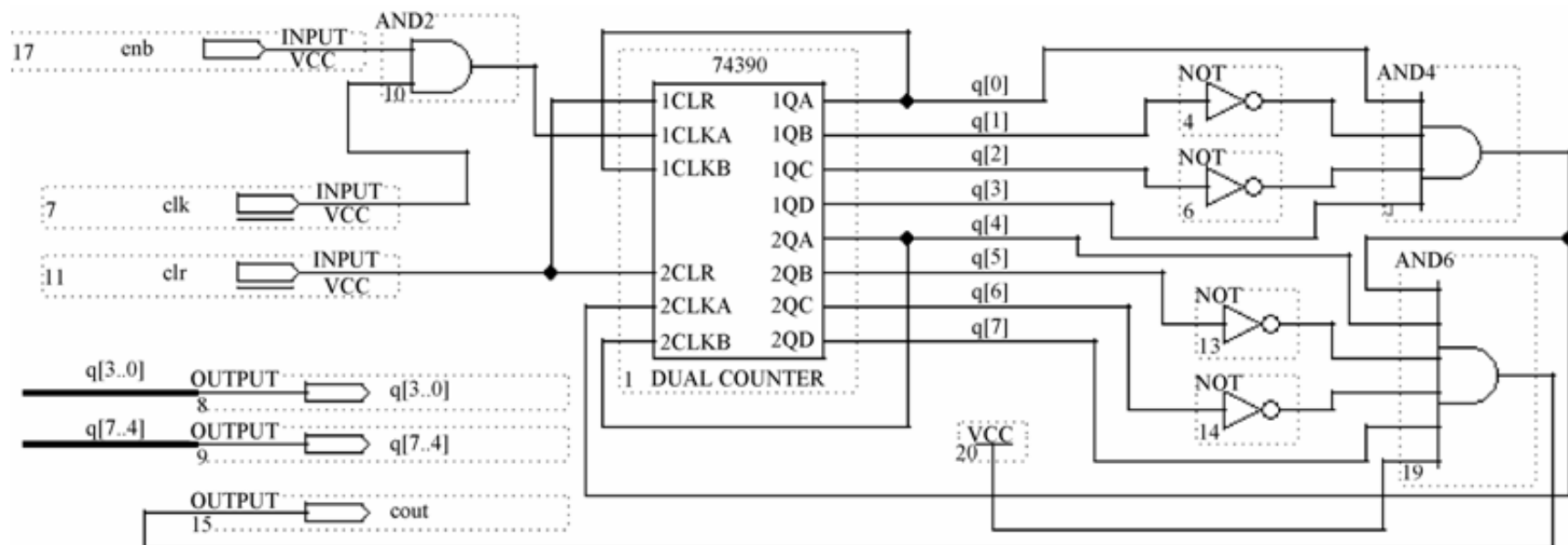


图 4-49 含有时钟使能的 2 位十进制计数器



# 4.5 原理图编辑输入设计流程

## 1. 计数器设计

(2) 建立工程

(3) 系统仿真

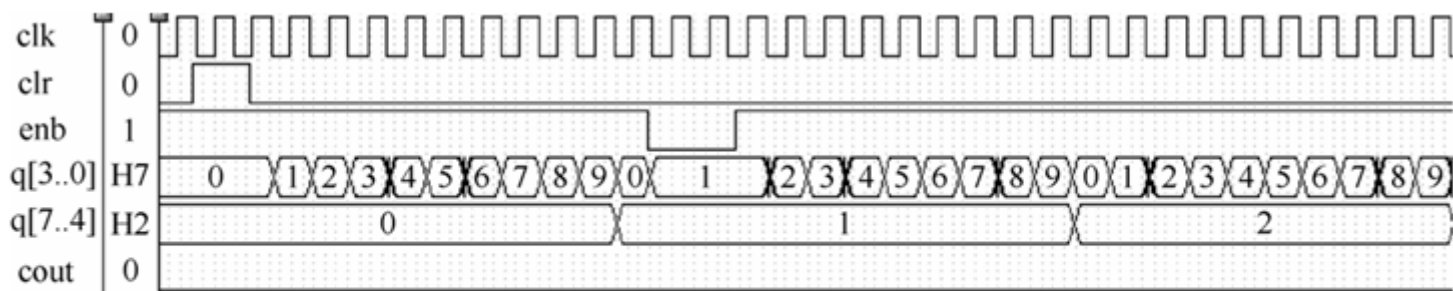


图 4-50 两位十进制计数器工作波形

(4) 生成元件符号

# 4.5 原理图编辑输入设计流程

## 2. 频率计主结构电路设计

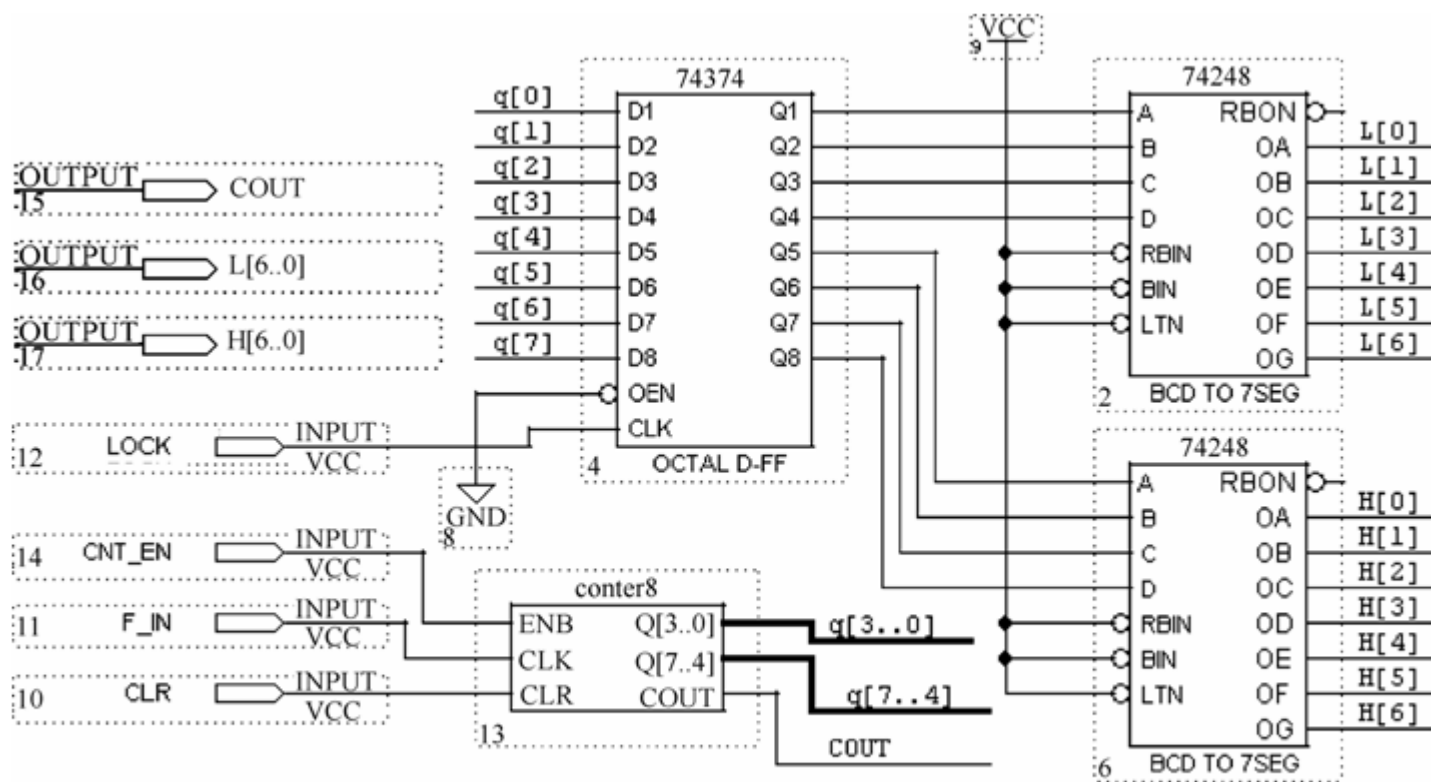


图 4-51 2 位十进制频率计顶层设计原理图文件

# 4.5 原理图编辑输入设计流程

## 2. 频率计主结构电路设计

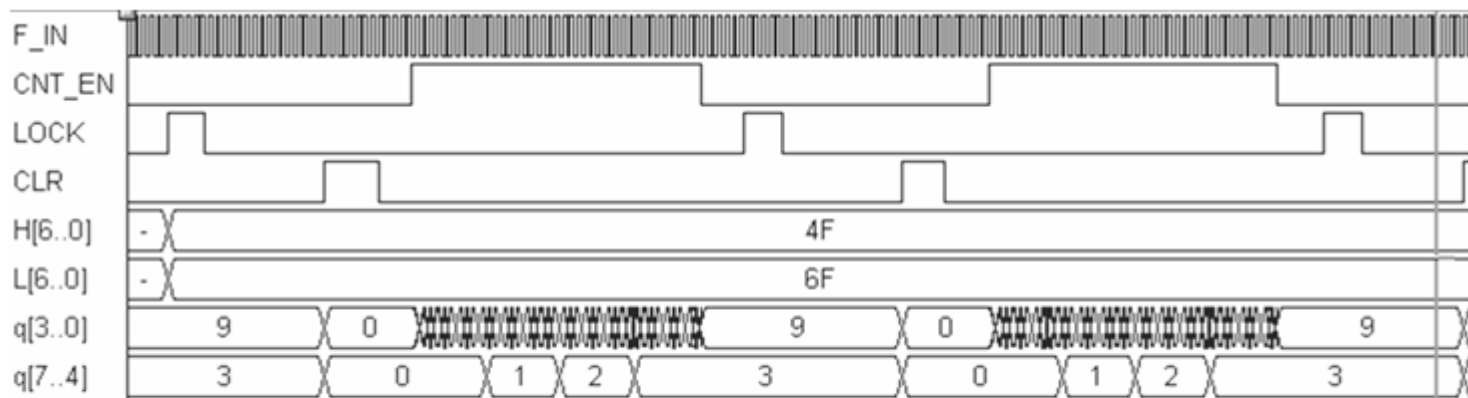


图 4-52 两位十进制频率计测频仿真波形

# 4.5 原理图编辑输入设计流程

## 3. 时序控制电路设计

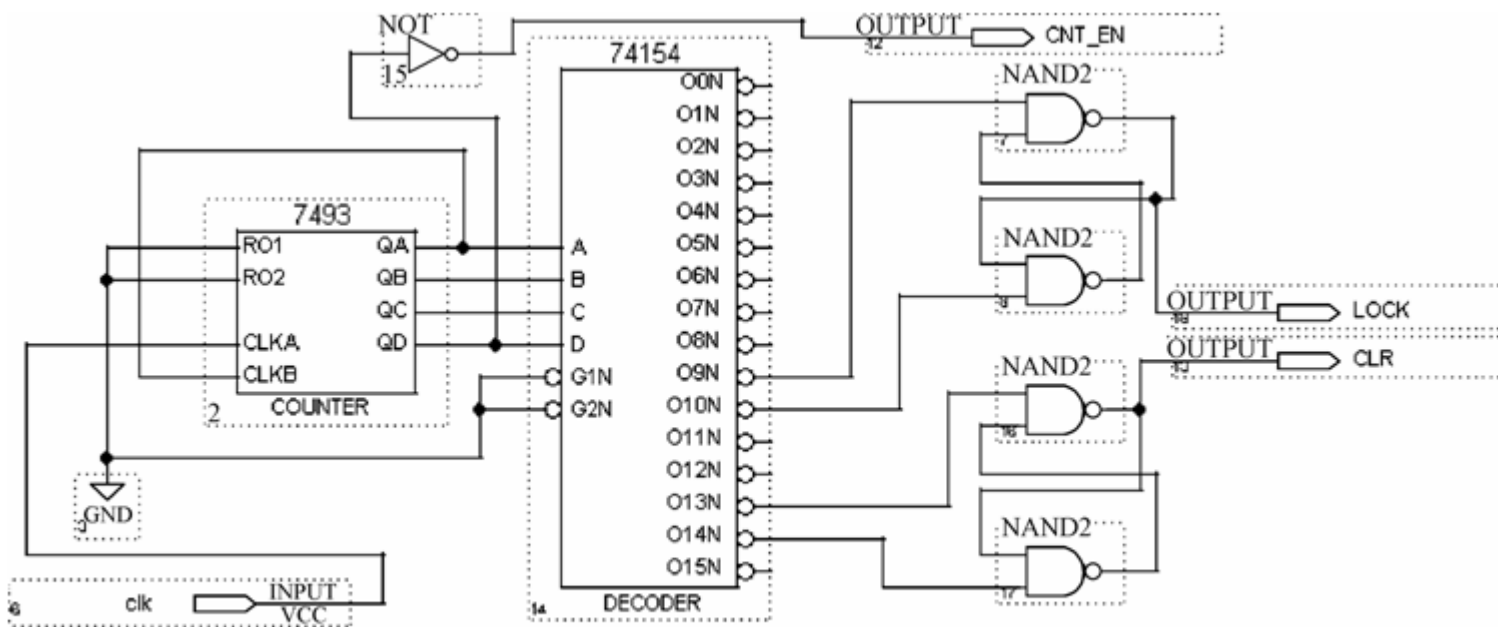


图 4-53 测频时序控制电路

# 4.5 原理图编辑输入设计流程

## 3. 时序控制电路设计

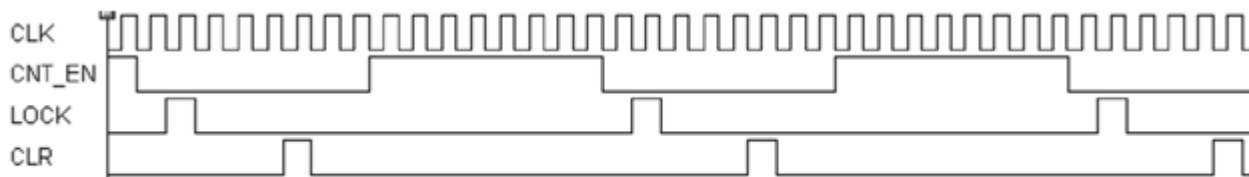


图 4-54 测频时序控制电路工作波形

# 4.5 原理图编辑输入设计流程

## 4. 顶层电路设计

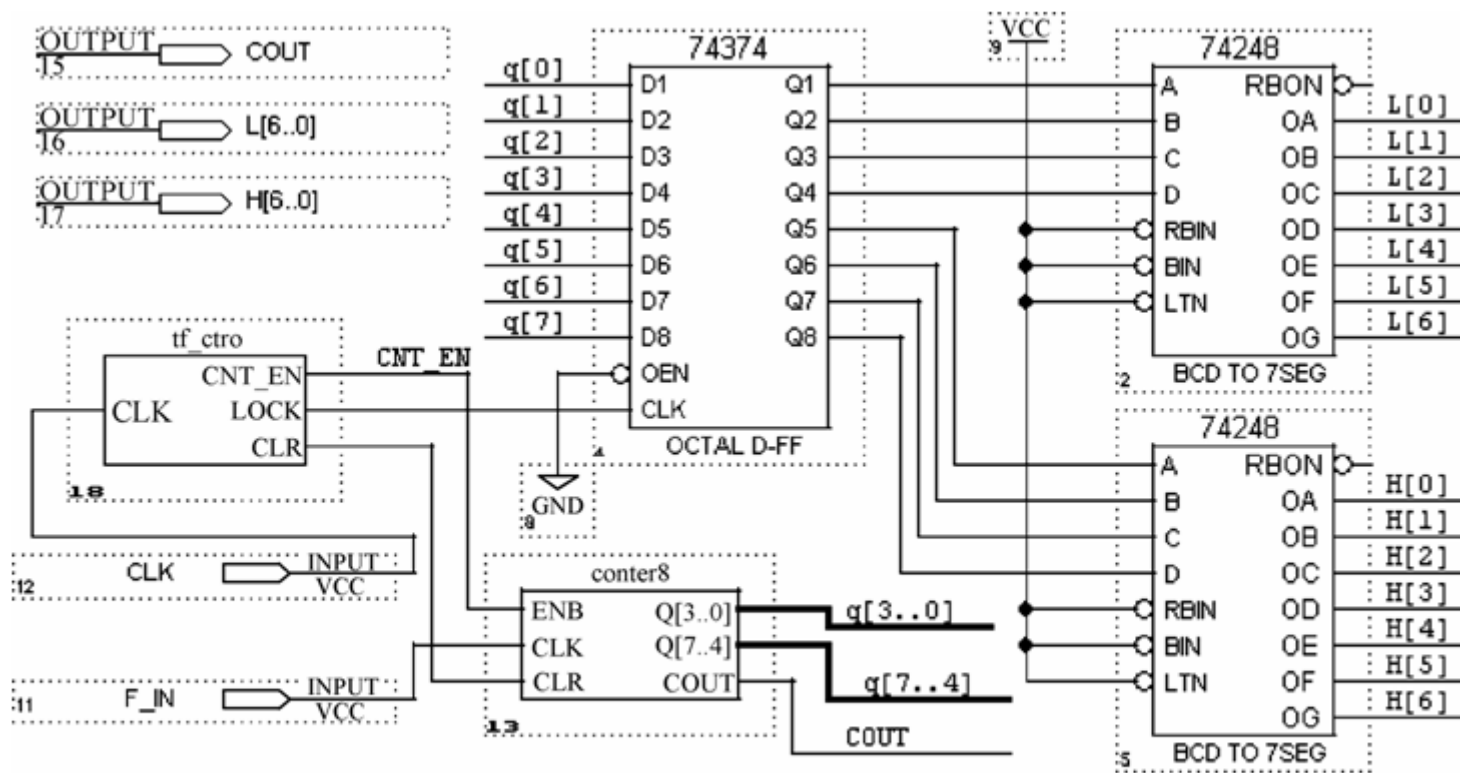


图 4-55 频率计顶层电路原理图

# 4.5 原理图编辑输入设计流程

## 4. 顶层电路设计

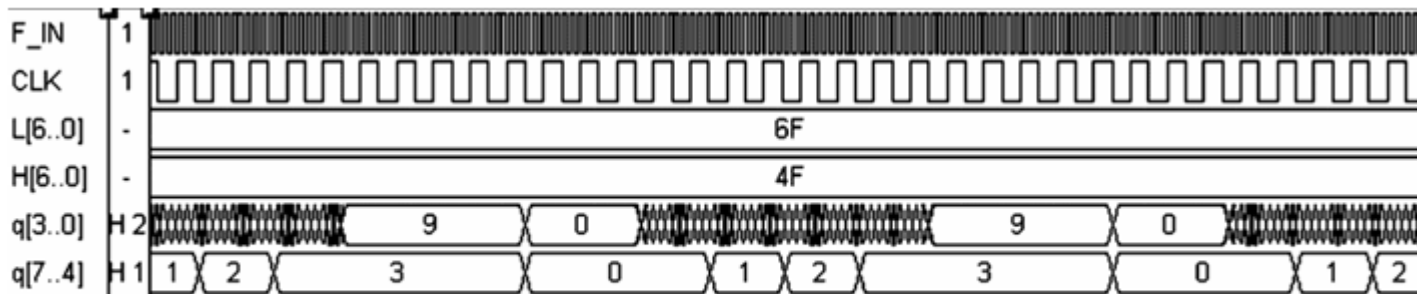


图 4-56 频率计工作时序波形

# 4.5 原理图编辑输入设计流程

## 4.5.3 宏模块逻辑功能查询

```
FUNCTION 74138 (g1, g2an, g2bn, c, b, a)
  RETURNS (y0n, y1n, y2n, y3n, y4n, y5n, y6n, y7n);
```

Inputs						Outputs							
Enable		Select				Y0N	Y1N	Y2N	Y3N	Y4N	Y5N	Y6N	Y7N
G1	G2*	C	B	A									
X	H	X	X	X		H	H	H	H	H	H	H	H
L	X	X	X	X		H	H	H	H	H	H	H	H
H	L	L	L	L		L	H	H	H	H	H	H	H
H	L	L	L	H		H	L	H	H	H	H	H	H
H	L	L	H	L		H	H	L	H	H	H	H	H
H	L	L	H	H		H	H	H	L	H	H	H	H
H	L	H	L	L		H	H	H	H	L	H	H	H
H	L	H	L	H		H	H	H	H	H	L	H	H
H	L	H	H	L		H	H	H	H	H	H	L	H
H	L	H	H	H		H	H	H	H	H	H	H	L

\* G2 = G2AN + G2BN

图 4-57 74138 真值表





## 4.6 keep属性应用

---

### 【例 4-3】

```
module f_adder(ain,bin,cin,cout,sum);
    output cout,sum ;    input ain,bin,cin ;
    wire d,f ;
    (* synthesis, keep *) wire e ;
    h_adder u1( ain, bin, e, d );
    h_adder u2(.a(e), .so(sum), .b(cin),.co(f) );
    or2a u3(.a(d), .b(f), .c(cout) );
endmodule
```

(\* synthesis, keep \*) 或 (\* synthesis, probe\_port, keep \*)

## 4.6 keep属性应用

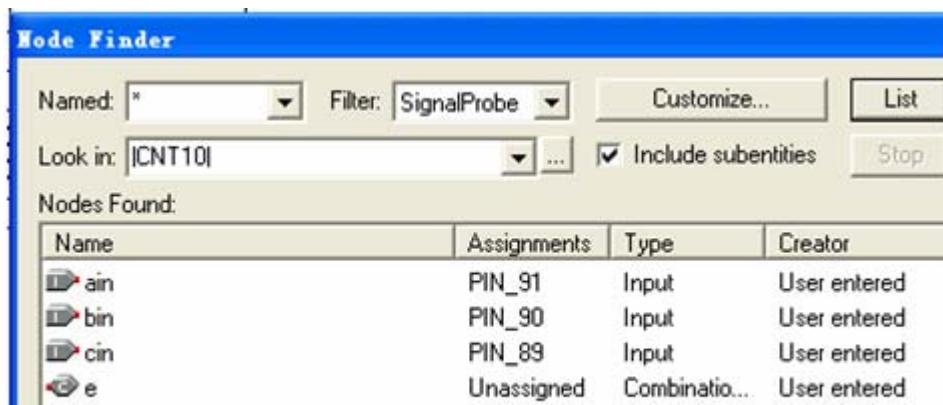


图 4-58 加入仿真测试信号 e

## 4.6 keep属性应用

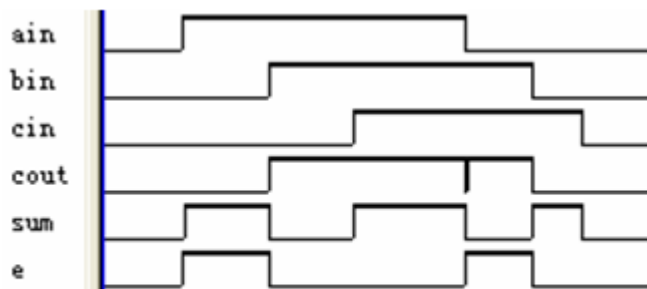


图 4-59 例 4-3 的仿真波形

```
(* synthesis, probe_port, keep *) wire e;
```

```
(* synthesis, probe_port, keep *)wire e reg [7:0] A ;
```

# 4.7 SignalProbe使用方法

1. 按常规流程完成设计仿真和硬件测试
2. 设置SignalProbe Pins

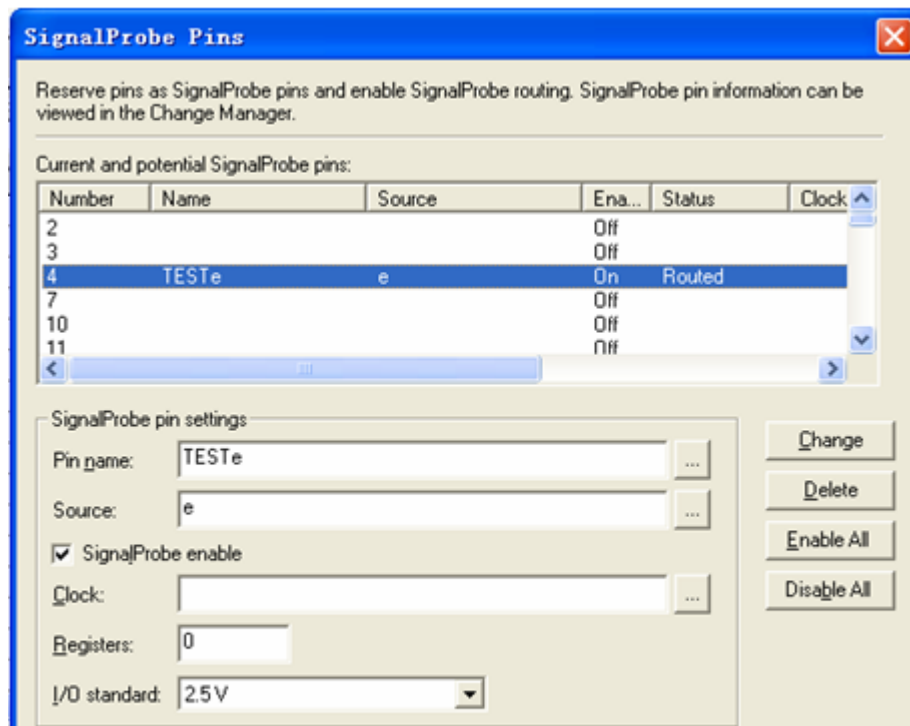


图 4-60 在 SignalProbe 对话框设置探测信号 e

# 4.7 SignalProbe使用方法

## 3. 编译SignalProbe Pins测试信息并下载测试



图 4-61 ECO 文件编译成功



## 4.8 Settings设置

---

- (1) 修改工程设置。
- (2) HDL设置。
- (3) 时序设置。
- (4) 编译器设置。
- (5) 仿真器设置。
- (6) 软件构建设置。。
- (7) HardCopy时序设置。

# 4.9 Fitter Settings项设置

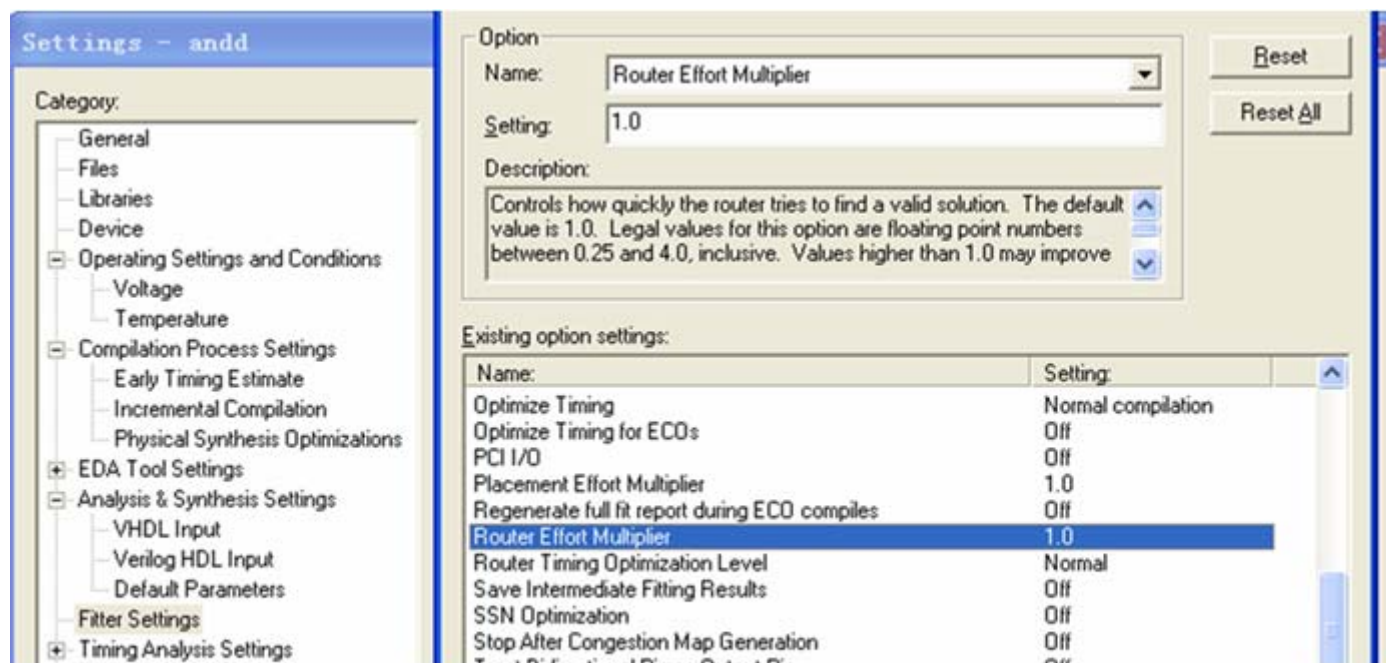


图 4-62 布线倍增器优化程度指数选择



## 4.10 HDL版本设置及**Analysis & Synthesis**功能

---



# 4.11 功能块Chip Planner应用

## 4.11.1 Chip Planner应用流程说明

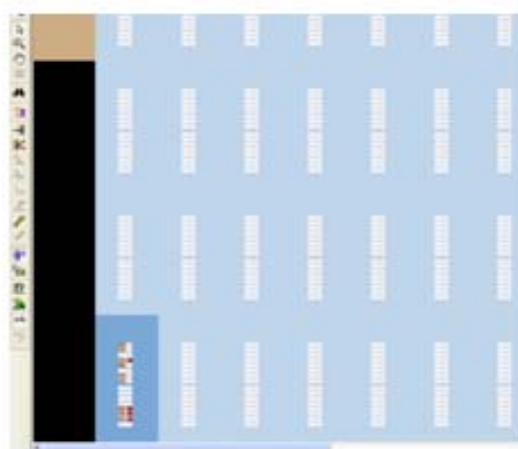


图 4-63 左下侧是已占用的 LAB

# 4.11 功能块Chip Planner应用

## 4.11.1 Chip Planner应用流程说明

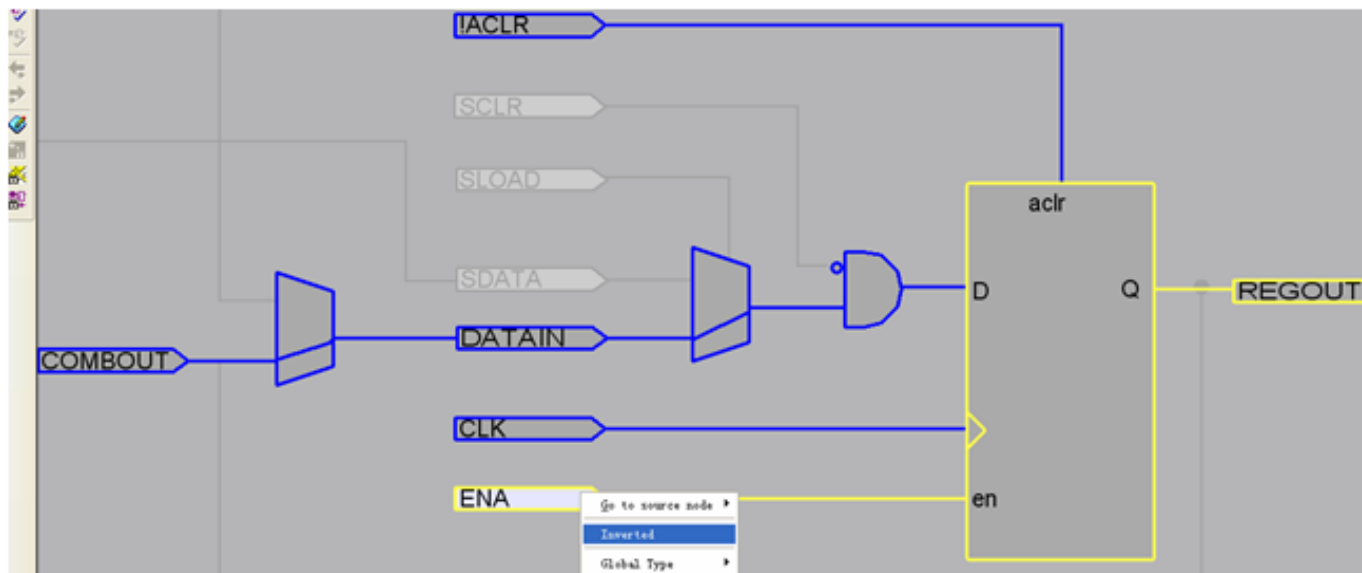


图 4-64 Resource Property Editor 的门级原理图编辑窗口

# 4.11 功能块Chip Planner应用

## 4.11.2 Chip Planner说明

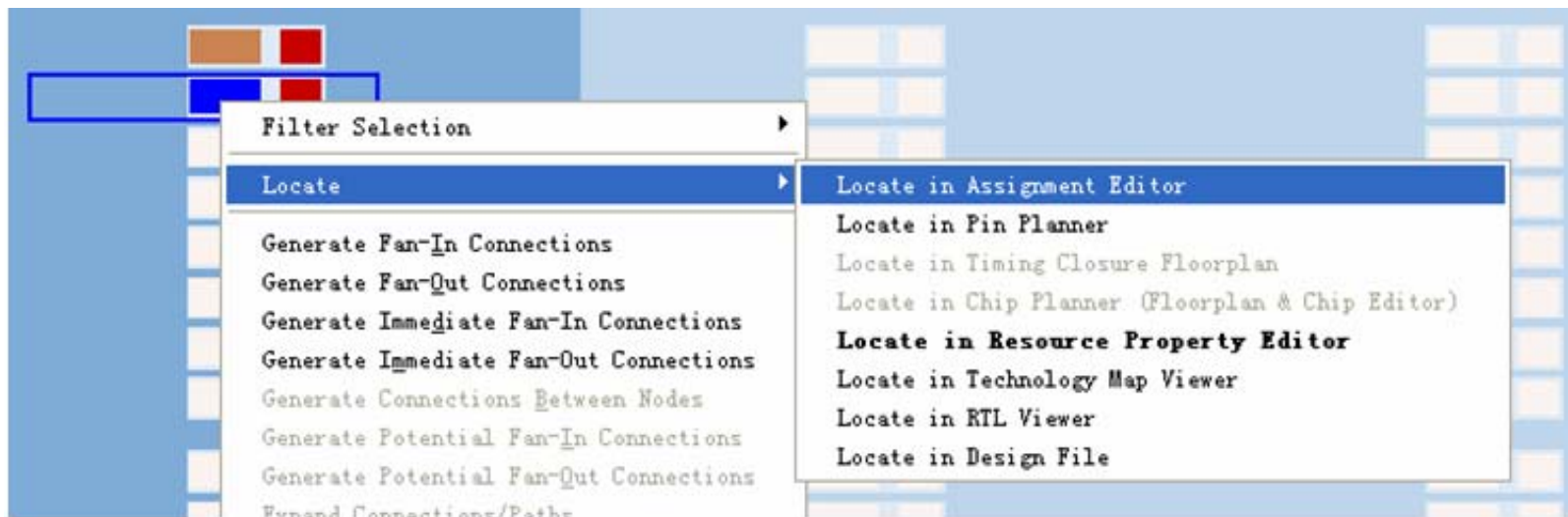


图 4-65 Locate 多项选择

# 4.11 功能块Chip Planner应用

## 4.11.2 Chip Planner说明

The screenshot displays the Chip Planner interface. The top portion shows a logic diagram with several input ports (DATAA, DATAIN, DATA, SCLR, SLOAD, S0010, S001A, CLK, ETC) and a central logic block 'A'. The output of block 'A' is labeled 'COMBOUT', which is connected to the 'D' input of a register block 'acr'. The 'en' input of the register is connected to 'DATAIN'. The 'Q' output of the register is labeled 'REGOUT'. Below the diagram are three windows:

- Connectivity:** A table showing input and output port connections.
- Properties:** A table showing properties for the selected node.
- Node:** A table showing timing information for the selected node.

Input Port name	Signal name	Latch info	Inverted
Register Node			
SLOAD	<Disconnected>	N/A	False
DATAIN	[CNT10 Q1[0]]^3	N/A	False
DATA	<Disconnected>	N/A	False
SCLR	<Disconnected>	N/A	False
IACL	[CNT10 RST^inputclkctrl]	N/A	True

Output Port name	Signal name	Latch info
Register Node		
REGOUT	[CNT10 Q1[0]]	N/A
Combinational Node		
COMBOUT	[CNT10 Q1[0]]^3	N/A
COU	<Disconnected>	N/A

Properties/Modes	Values
Sum LUT Mask	OFOF
Carry LUT Mask	N/A
Operation Mode	normal
Latch Type	none

Properties	Values
Sum Equation	1C
Carry Equation	N/A

Node:	[CNT10 Q1[0]]	Go To
Q		
ACL	706/678 ps	
CLK	667/680 ps	
DATAIN	99/115 ps	

图 4-66 打开属性和端口连接窗口



# 4.11 功能块Chip Planner应用

## 4.11.3 利用Change Manager检测底层逻辑

- (1) 更改编号。
- (2) 节点名称 (Node Name) 。
- (3) 更改类型 (Change Type) 。
- (4) 旧值 (Old Value) 。
- (5) 目标值 (Target Value) 。
- (6) 当前值 (Current Value) 。
- (7) 用户添加的有关ECO更改的备注。
- (8) 状态 (Status)

- |   |
|---|
| <ol style="list-style-type: none"><li>1、待定：</li><li>2、已应用：</li><li>3、无效：</li><li>4、未应用：</li></ol> |
|---|

# 4.12 Synplify的应用及接口方法

## 4.12.1 Synplify使用流程

### 1. 启动Synplify

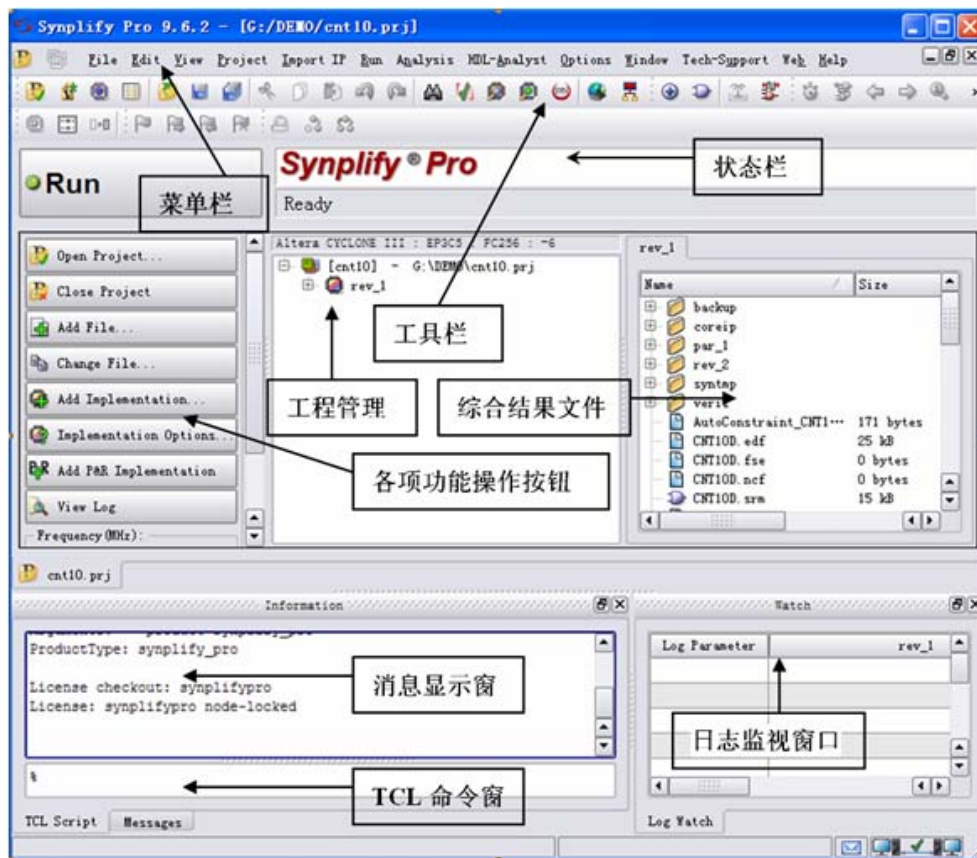


图4-67 Synplify Pro启动后界面

# 4.12 Synplify的应用及接口方法

## 4.12.1 Synplify使用流程

### 2. 创建工程

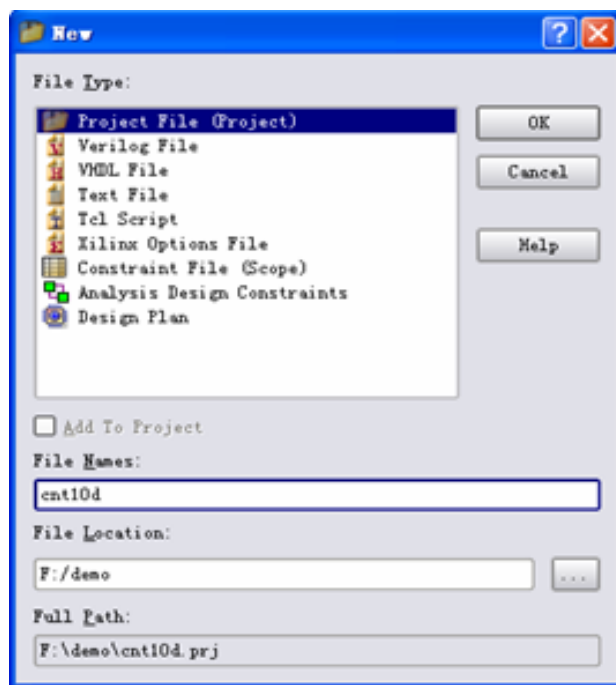


图4-68 创建工程

# 4.12 Synplify的应用及接口方法

## 4.12.1 Synplify使用流程

3. 加入源文件
4. 选择顶层文件
5. 设置工程属性

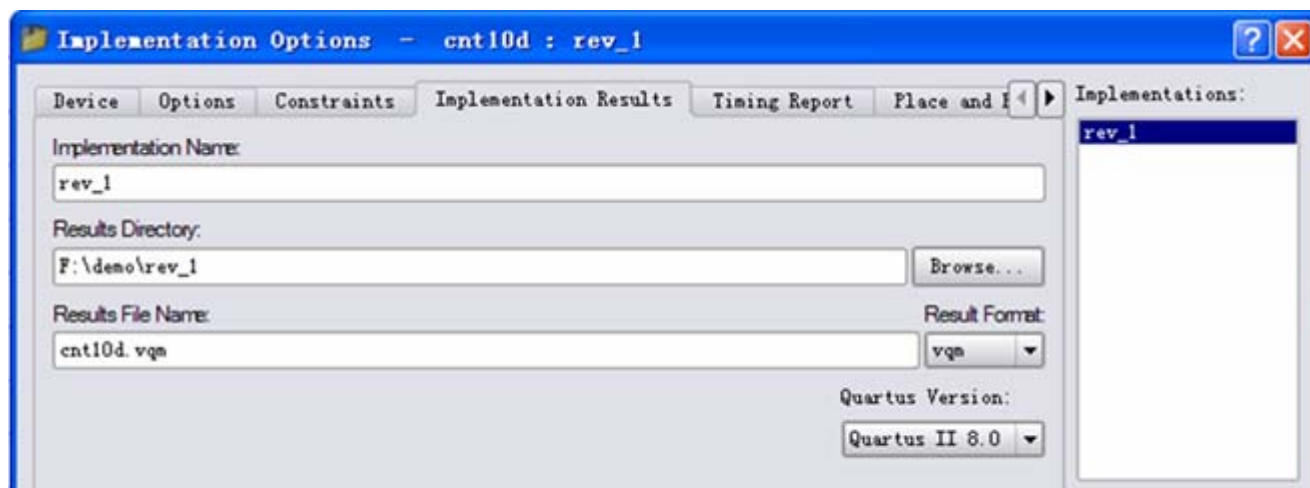


图4-69 综合目标详细信息





## 4.12 Synplify的应用及接口方法

---

### 4.12.1 Synplify使用流程

6. 综合前设置约束
7. 综合
8. 检测结果

# 4.12 Synplify的应用及接口方法

## 4.12.2 Synplify与Quartus II接口

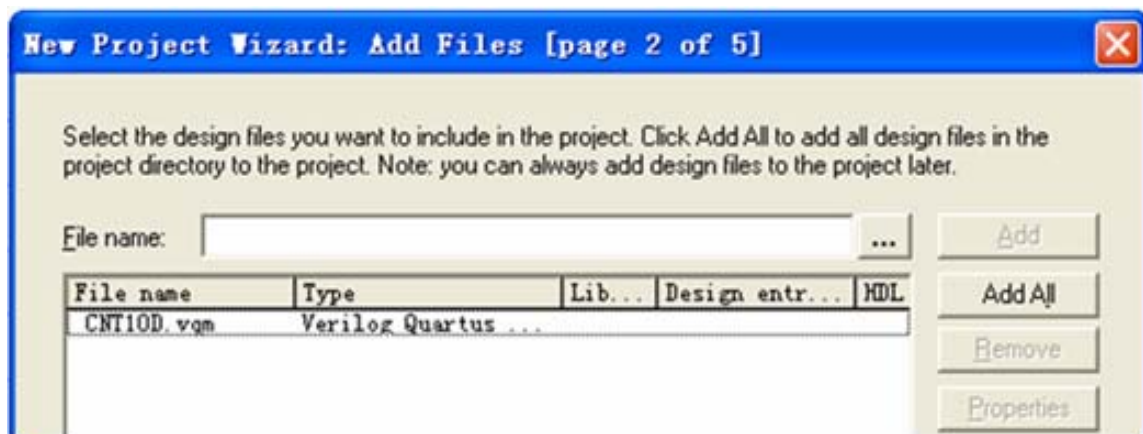


图 4-70 加入 Cnt10d.vqm 文件

# 4.12 Synplify的应用及接口方法

## 4.12.2 Synplify与Quartus II接口

### 1. Synplify软件路径设置

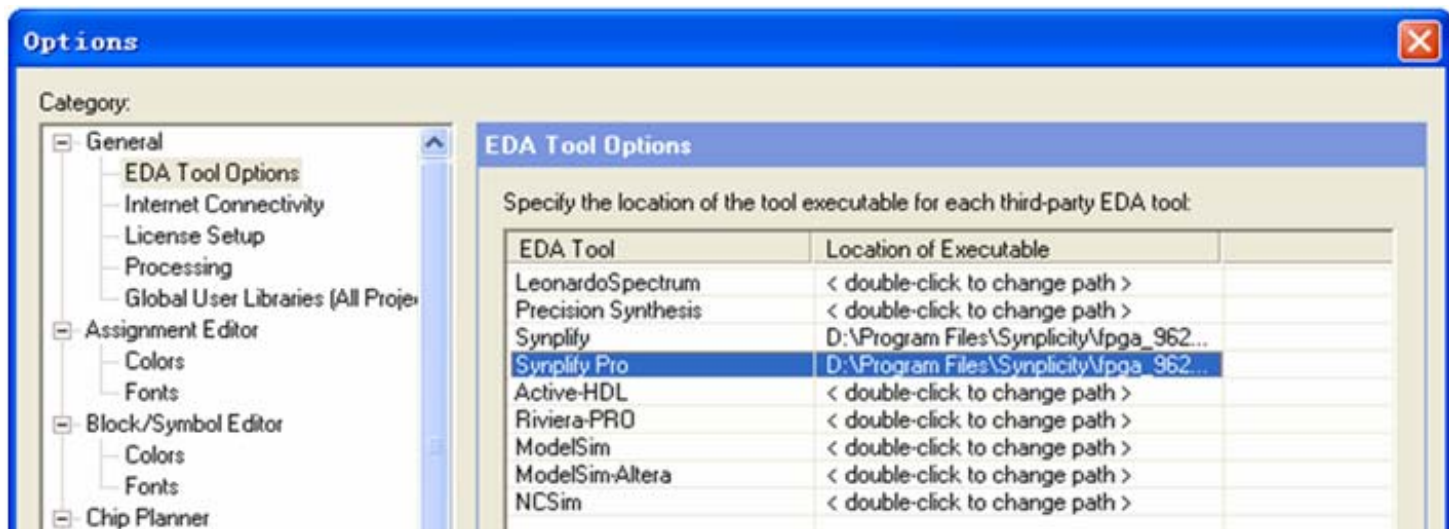


图4-71 Synplify软件路径设置

# 4.12 Synplify的应用及接口方法

## 4.12.2 Synplify与Quartus II接口

### 2. 设置Synplify Pro综合器

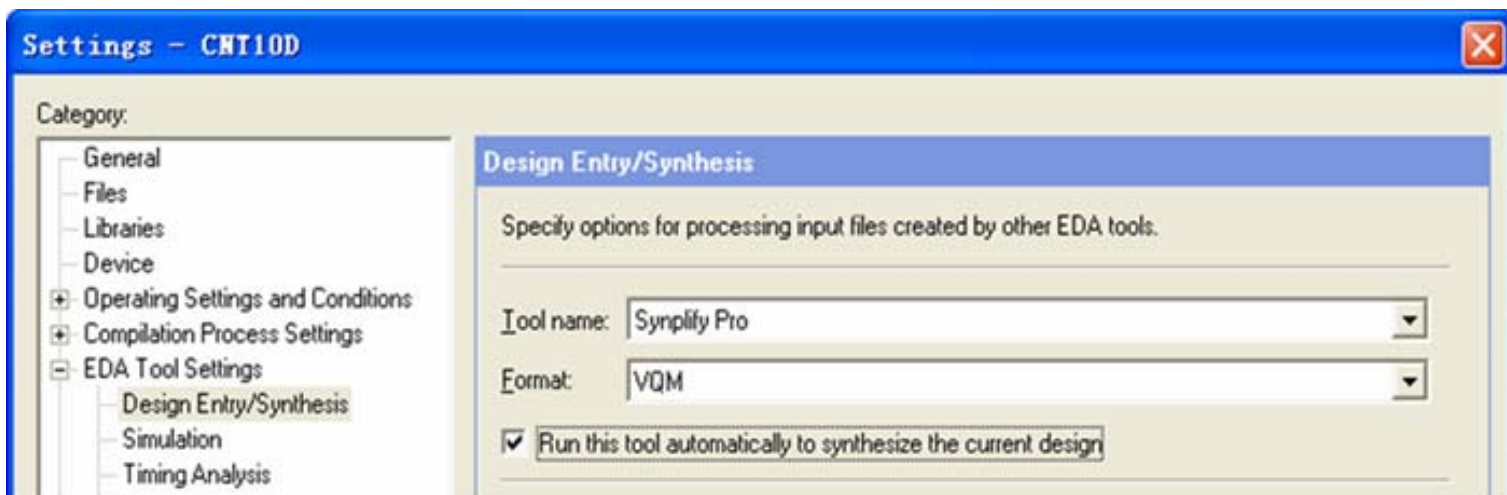


图4-72 设置综合器

# 实验与设计

## 4-1. 计数器设计实验

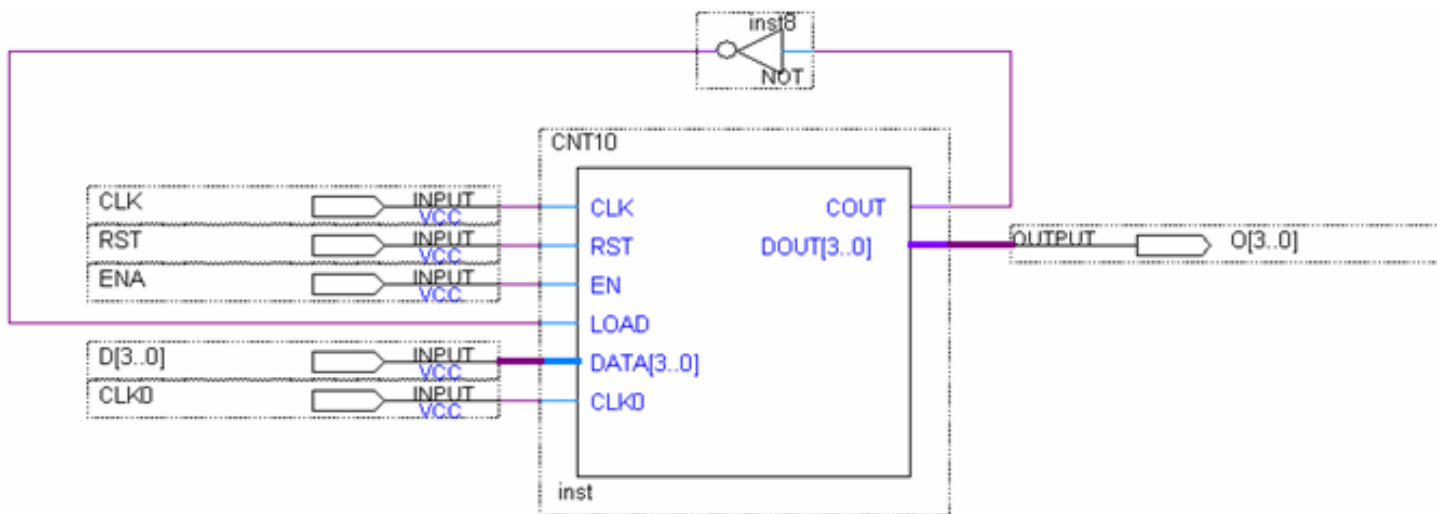


图 4-73 原理图示意图



# 实验与设计

---

4-2. 多路选择器设计实验

4-3. 8位全加器设计实验

4-4. 原理图输入法设计频率计



# 实验与设计

## 4-5. 十六进制7段数码显示译码器设计

表 4-2 7 段译码器真值表

输入码	输出码	代表数据
0000	0111111	0
0001	0000110	1
0010	1011011	2
0011	1001111	3
0100	1100110	4
0101	1101101	5
0110	1111101	6
0111	0000111	7
1000	1111111	8
1001	1101111	9
1010	1110111	A
1011	1111100	B
1100	0111001	C
1101	1011110	D
1110	1111001	E
1111	1110001	F

# 实验与设计

## 4-5. 十六进制7段数码显示译码器设计

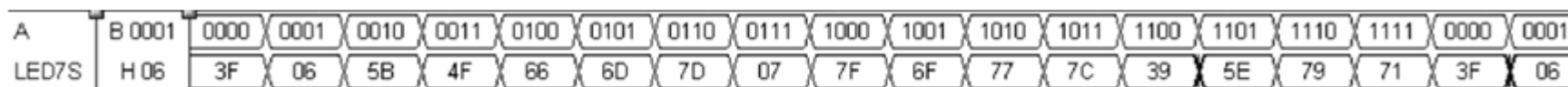


图 4-74 7 段译码器仿真波形

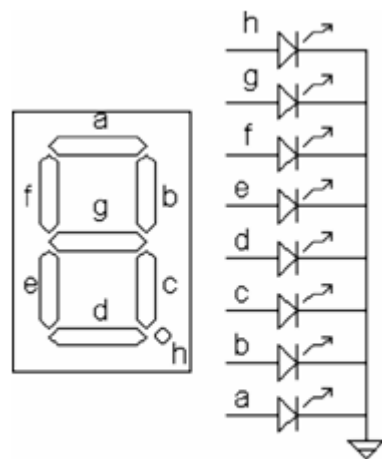


图 4-75 共阴数码管

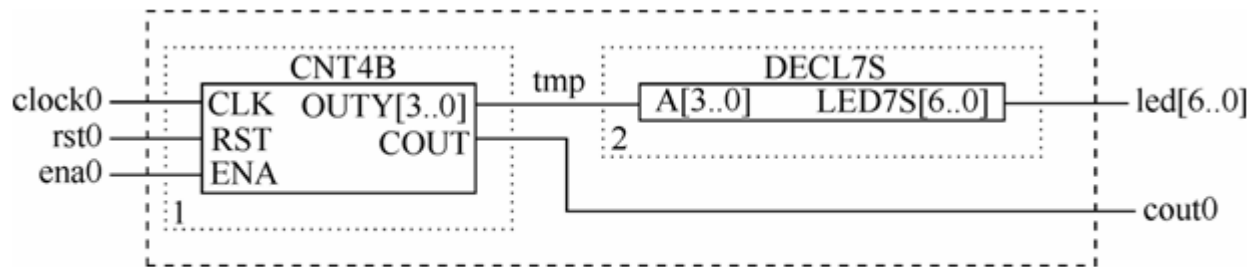


图 4-76 计数器和译码器连接电路的顶层文件原理图



# 实验与设计

## 4-6 数码扫描显示电路设计

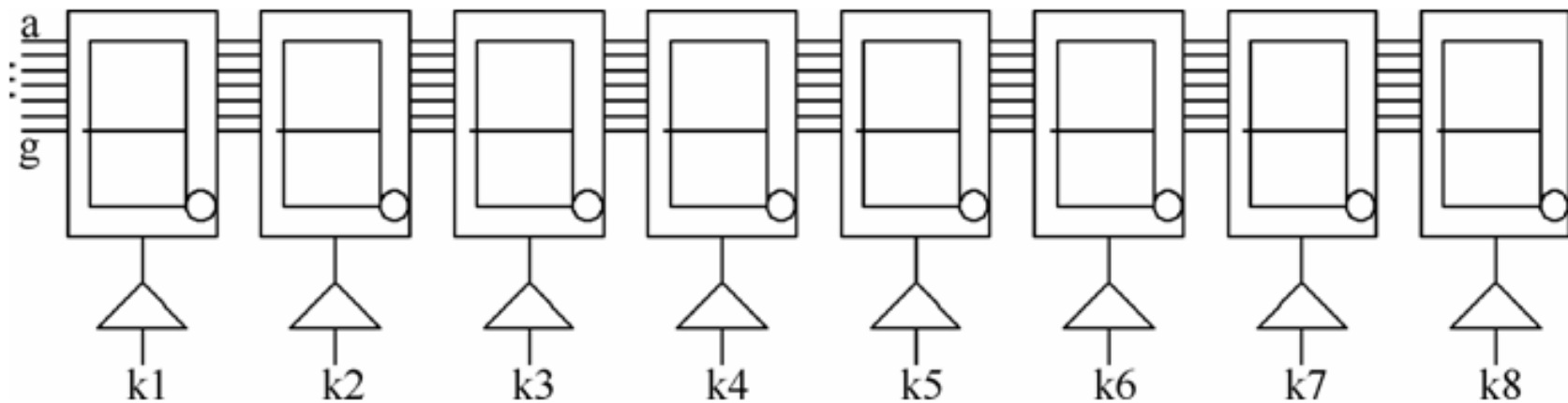


图 4-77 8 位数码扫描显示电路