

第 6 章

组合电路时序分析 与自动化设计

• 6.1 手工数字技术存在的问题

- 6.2 数字系统自动设计流程
- 6.3 原理图输入法逻辑电路设计
- 6.4 硬件测试
- 6.5 用HDL来表述广义译码器
-

手工数字技术存在的问题

速度低

设计规模小

分析技术无法适应需求

设计效率低

成本高

可靠性低

体积大功耗大

功能有限

无法实现功能升级

知识产权不易保护



- 6.1 手工数字技术存在的问题

- **6.2 数字系统自动设计流程**

- 6.3 原理图输入法逻辑电路设计

- 6.4 硬件测试

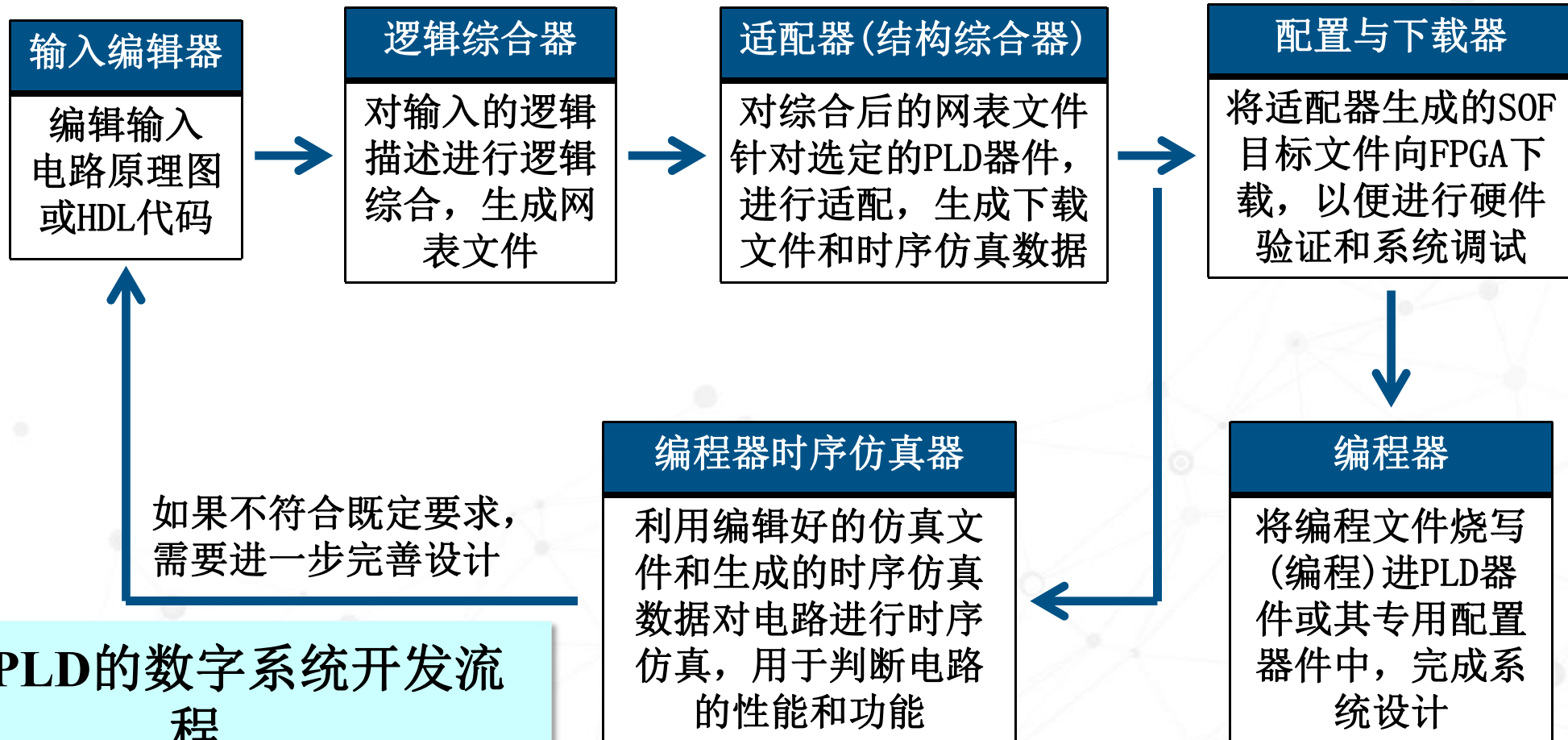
- 6.5 用HDL来表述广义译码器

- 6.6 数字方法去抖动和延时电路设计

-

数字系统自动设计流程

1. 设计输入



针对PLD的数字系统开发流
程

数字系统自动设计流程

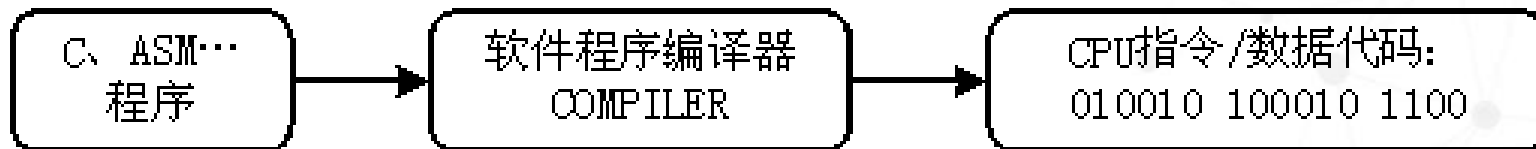
2. 硬件描述语言

3. 综合

4. 适配

5. 仿真

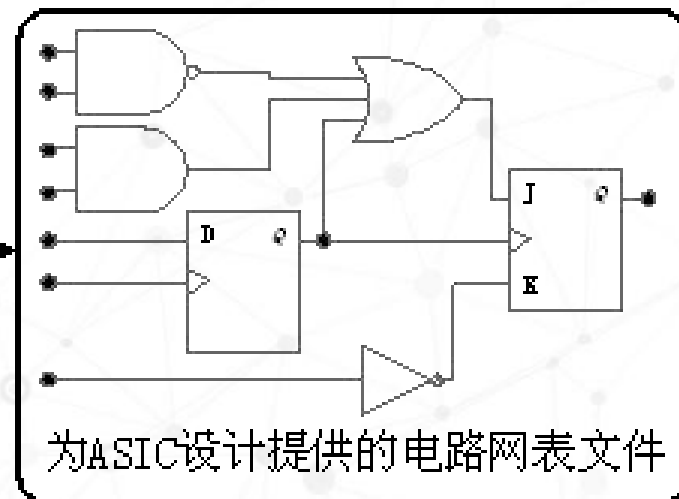
6. 硬件测试



(a) 软件语言设计目标流程



(b) 硬件语言设计目标流程



计算机软/硬件描述语言编译/综合工具的不同之处

- 6.1 手工数字技术存在的问题
- 6.2 数字系统自动设计流程
- **6.3 原理图输入法逻辑电路设计**
- 6.4 硬件测试
- 6.5 用HDL来表述广义译码器
- 6.6 数字方法去抖动和延时电路设计

原理图输入法逻辑电路设计

Quartus Prime 软件简介

Max Plus II

2004

Quartus II

2016

Quartus Prime

Altera公司

收购

2015.12.29

Intel公司

Intel FPGA网址: <https://www.intel.cn/content/www/cn/zh/products/programmable.html>

FPGA下载中心: <https://www.intel.cn/content/www/cn/zh/programmable/downloads/download-center.html>

原理图输入法逻辑电路设计

安装版本: Quartus II9.1

百度网盘链接:

<https://pan.baidu.com/s/1N83f0xnFMfxK5WZjNTWJ4w>

提取码: 5ct9

支持器件: Cyclone I/II/III/IV E

安装版本: Quartus Prime17.1

百度网盘链接:

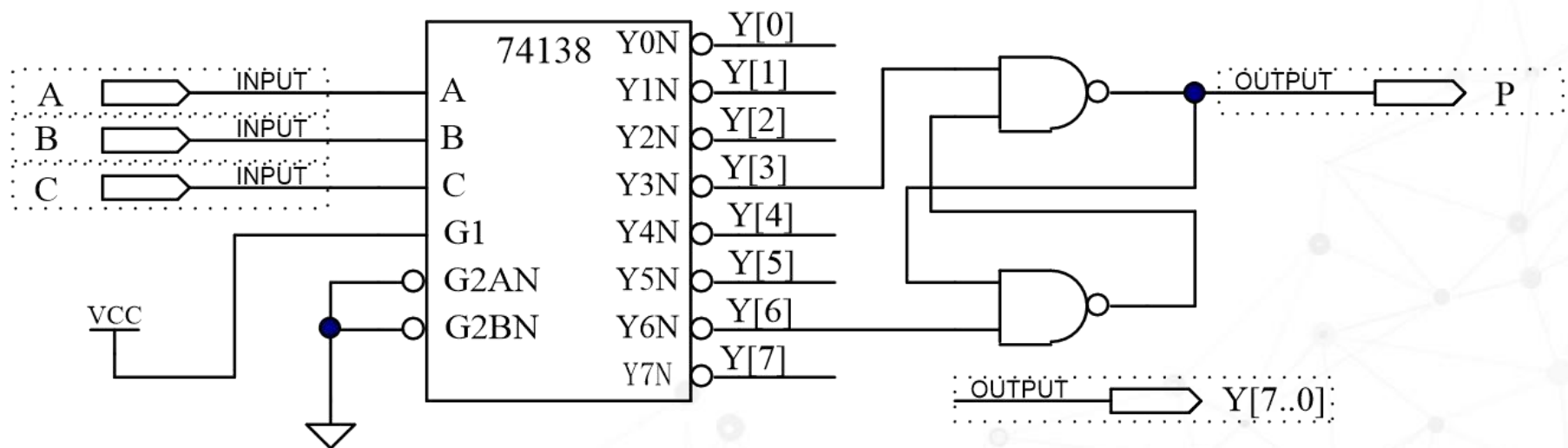
https://pan.baidu.com/s/1_2GnStTr8Z6yJ9aQWTd5sg

提取码: jhpz

支持器件: Cyclone 10 LP

原理图输入法逻辑电路设计

1. 原理图编辑输入方法

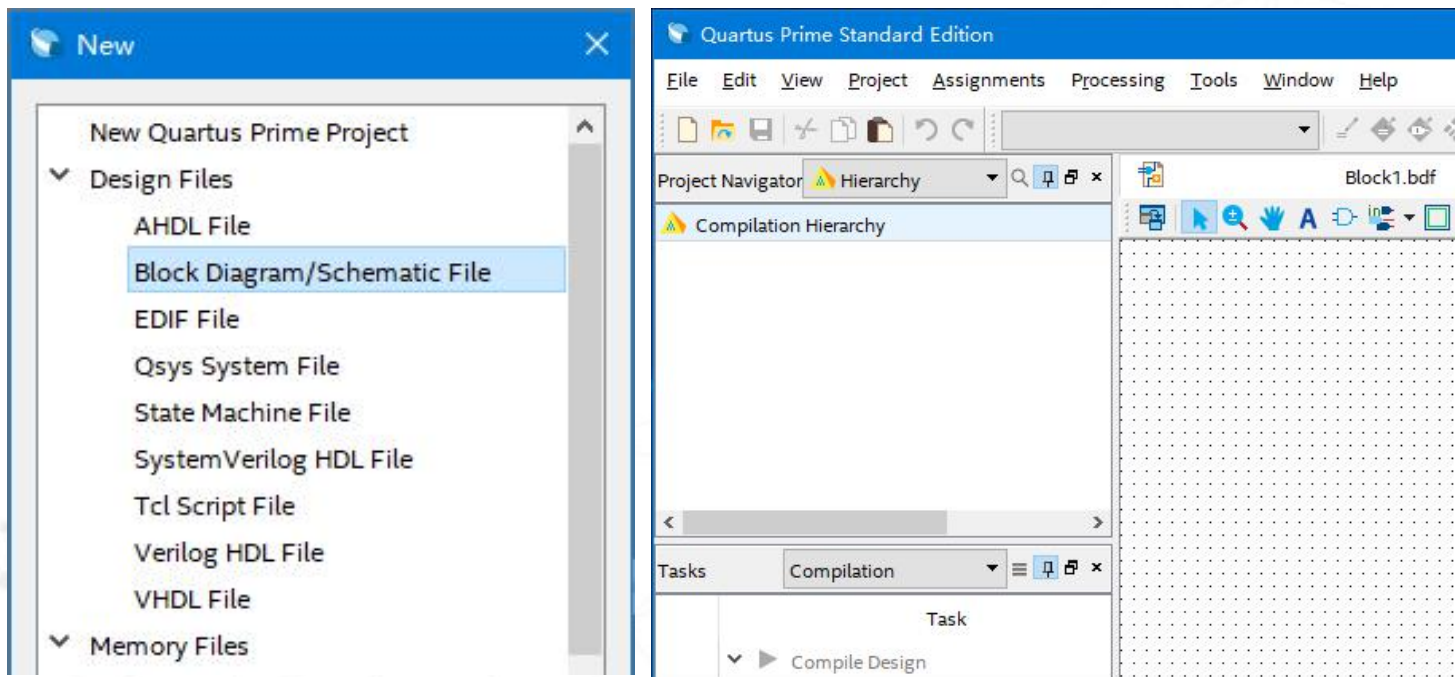


脉冲发生器示例电路图

原理图输入法逻辑电路设计

1. 原理图编辑输入方法

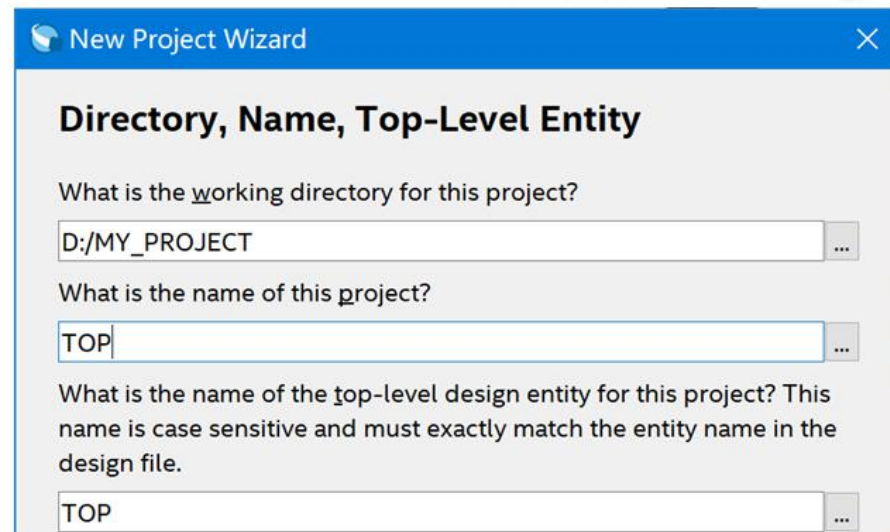
- (1) 新建一个文件夹。
- (2) 打开原理图编辑窗。
- (3) 文件存盘。



原理图输入法逻辑电路设计

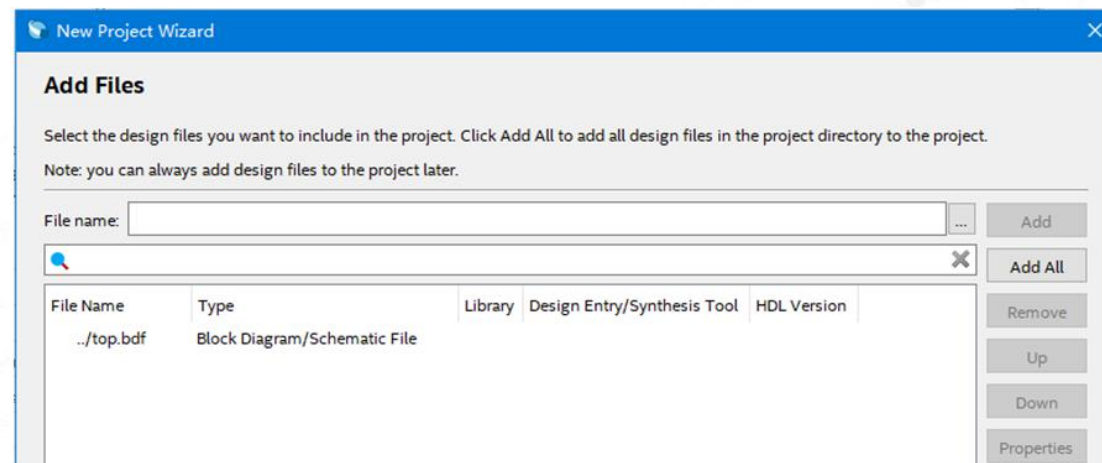
2. 创建工程

(1) 打开建立新工程管理窗。



利用“New Project Wizard”创建工程 TOP

(2) 将设计文件加入工程中。



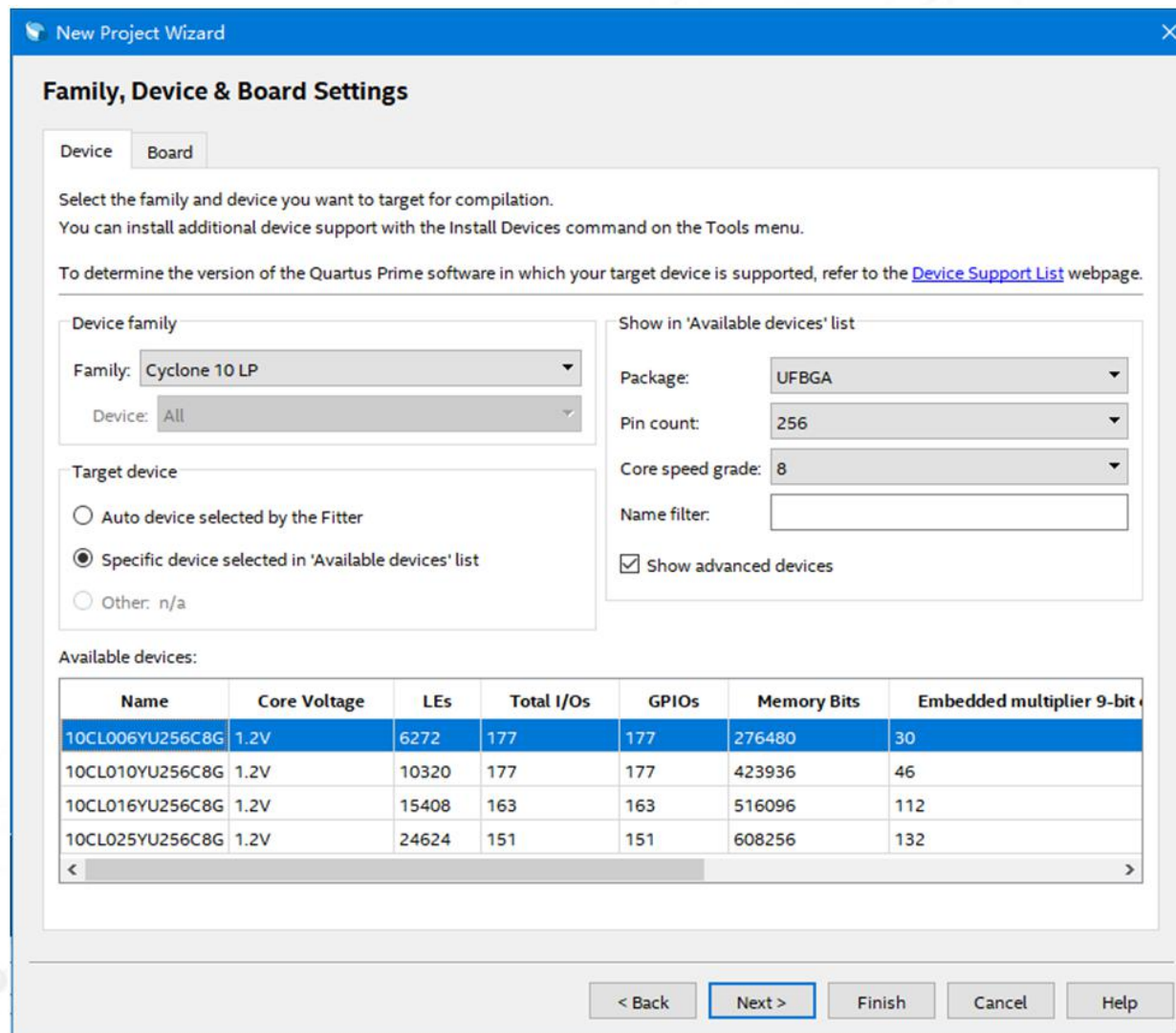
将所有相关的文件都加入此工程

原理图输入法逻辑电路设计

2. 创建工程

(3) 选择目标芯片。

(4) 工具设置。

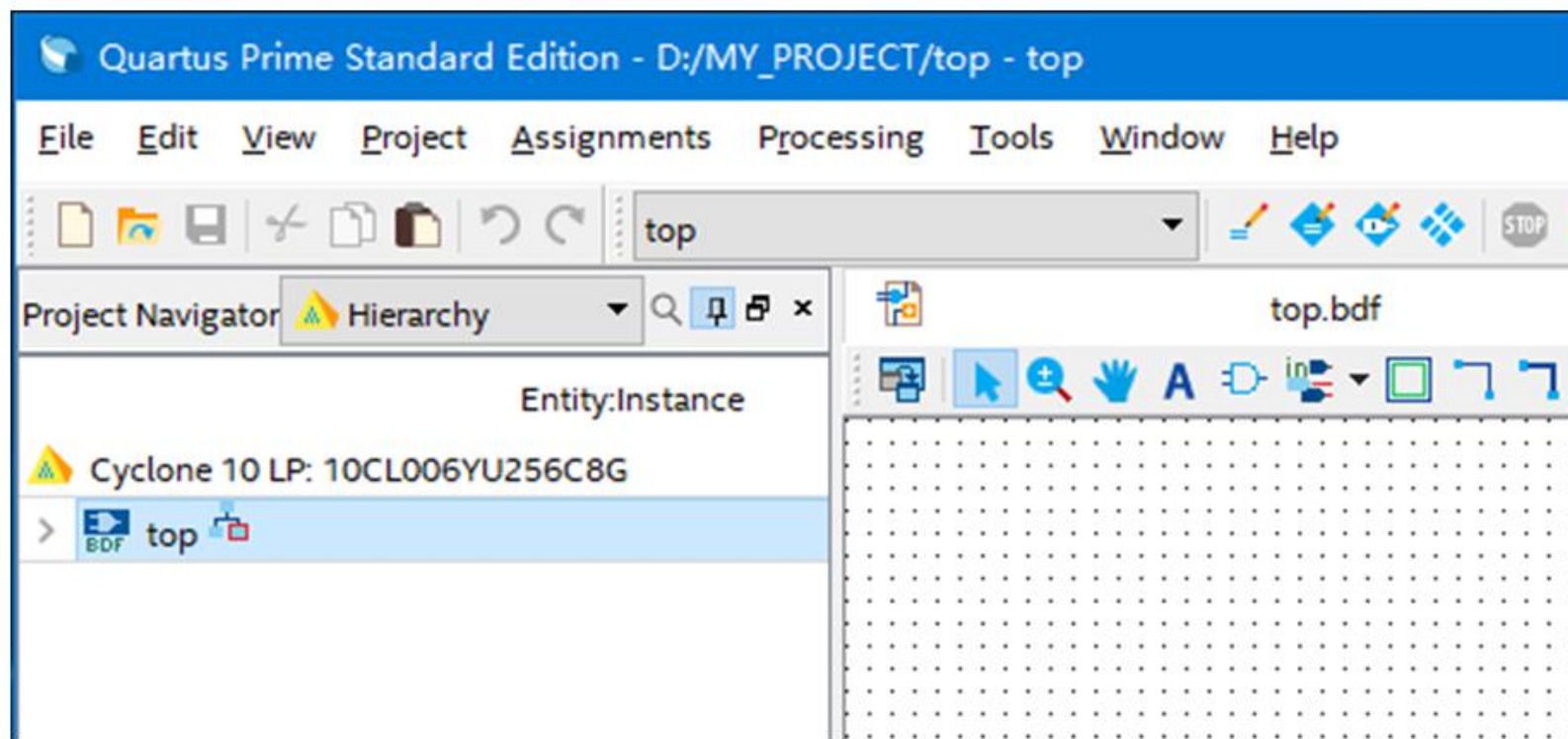


选择目标芯片 10CL006YU256C8G

原理图输入法逻辑电路设计

2. 创建工程

(5) 结束设置。

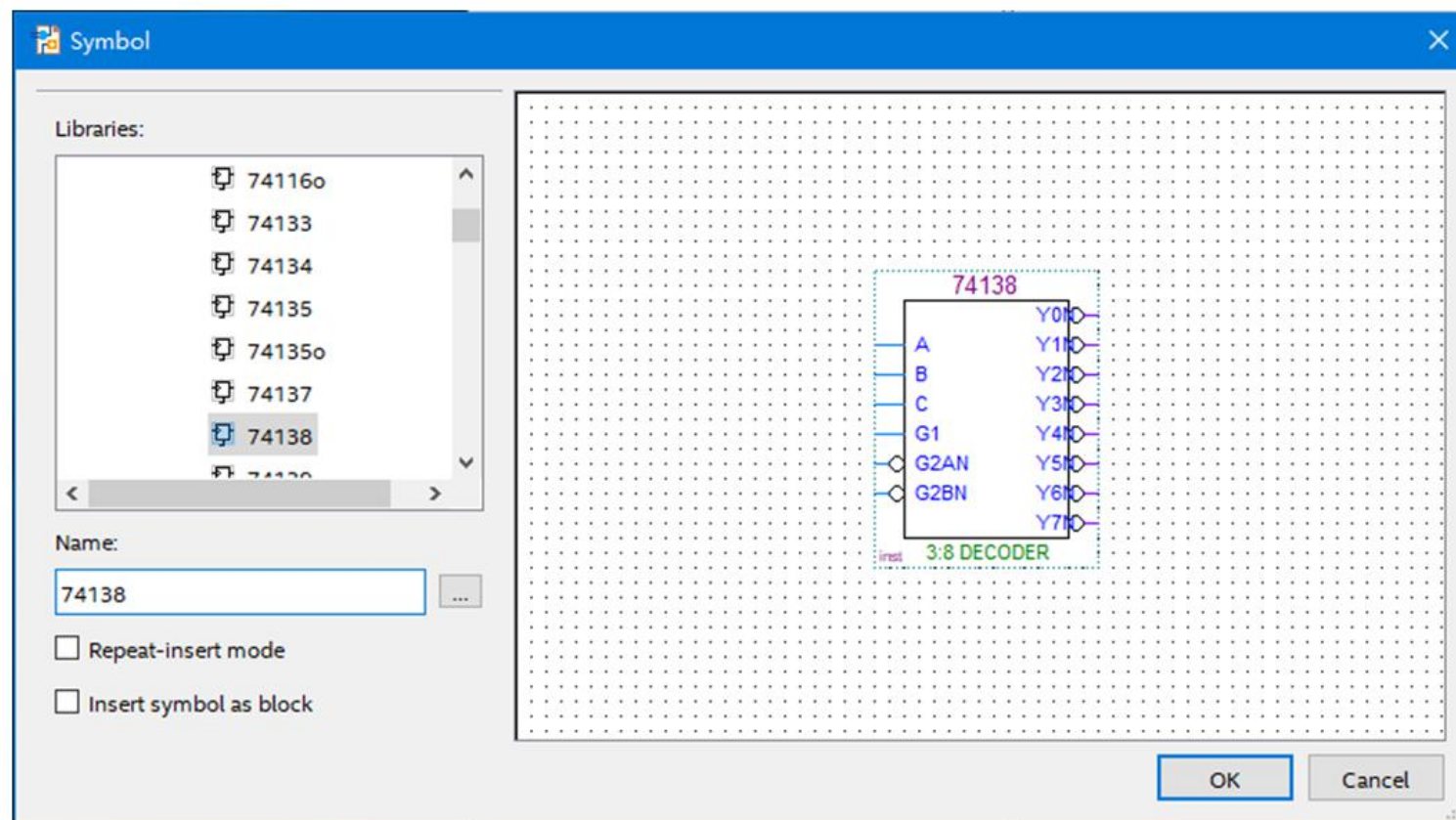


top 工程管理窗

原理图输入法逻辑电路设计

2. 创建工程

(6) 编辑构建电路图。



在元件调用对话框调出宏功能元件 74138

原理图输入法逻辑电路设计

3. 功能分析

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	L	H	H
H	L	H	H	H		H	H	H	H	H	H	L	H

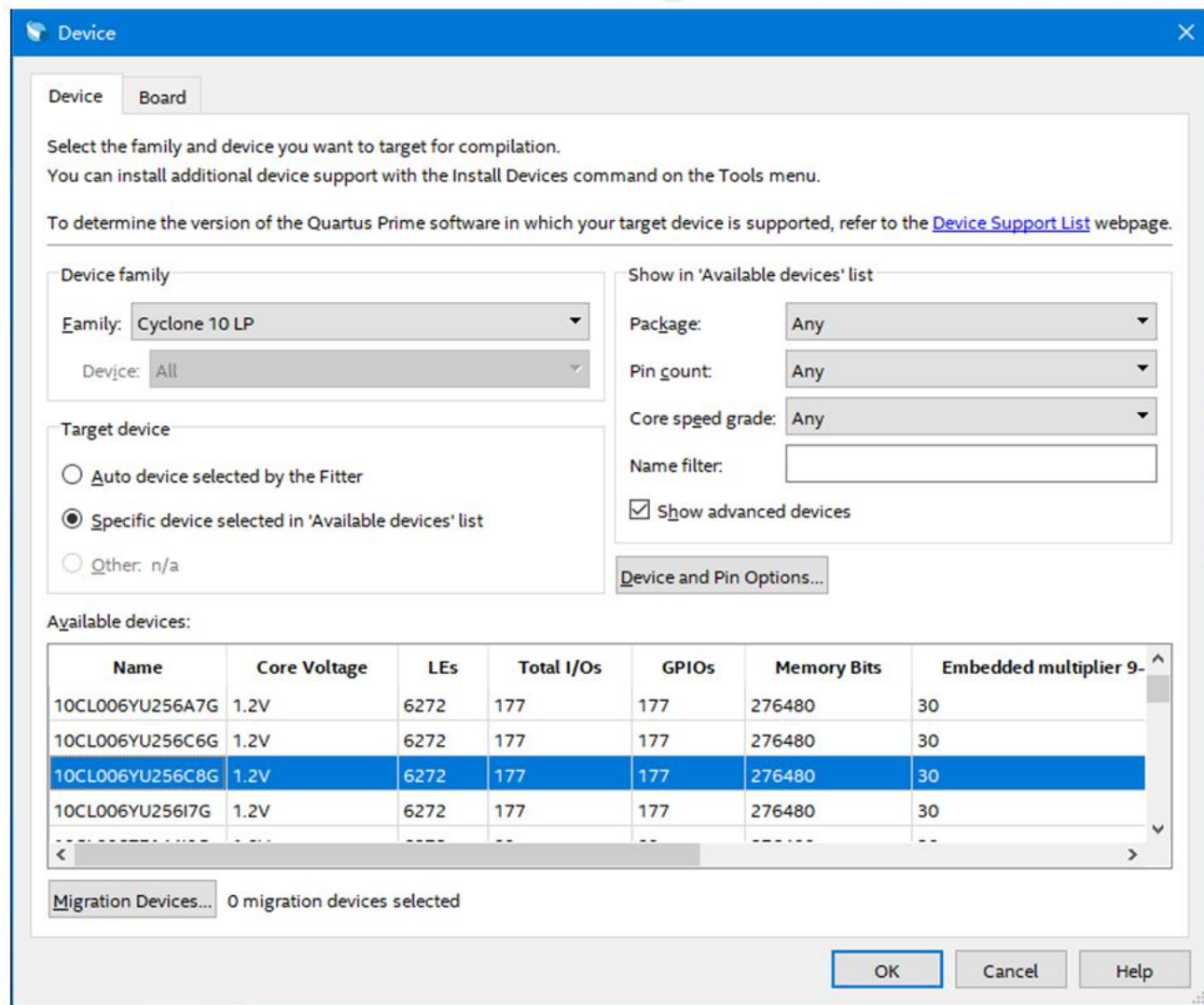
* $G2 = G2AN + G2BN$

来自帮助文件 Macrofunctions 中 74138 的真值表

原理图输入法逻辑电路设计

4. 编译前设置

(1) 选择FPGA目标芯片。

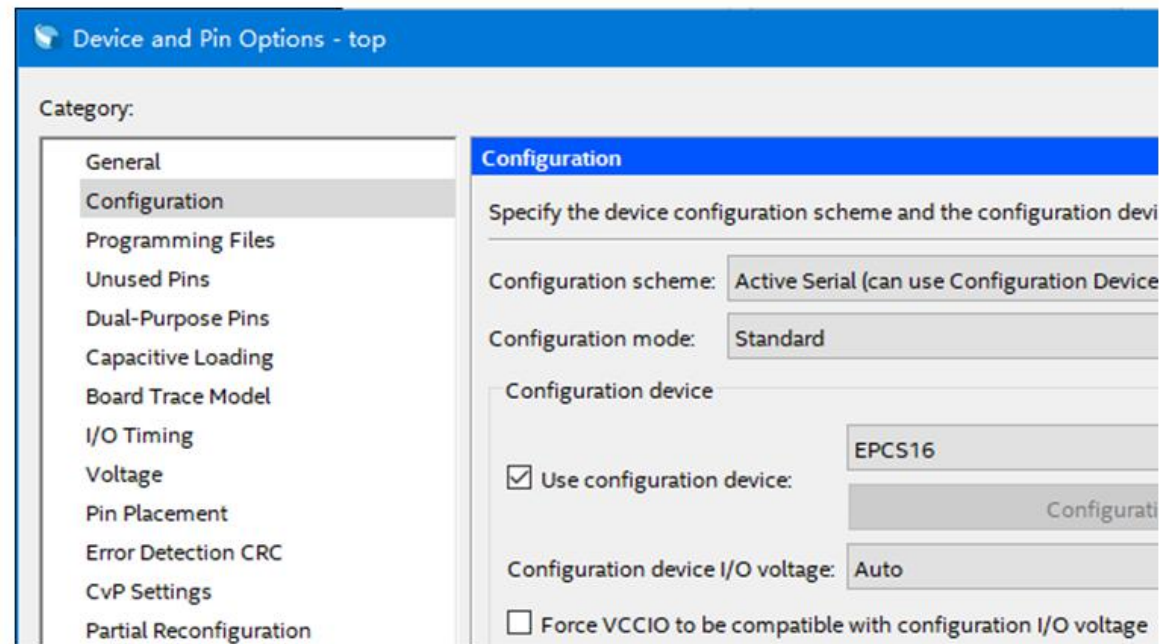


由 Device 对话框选择目标器件 10CL006YU256C8G

原理图输入法逻辑电路设计

4. 编译前设置

(2) 选择配置器件的工作方式。



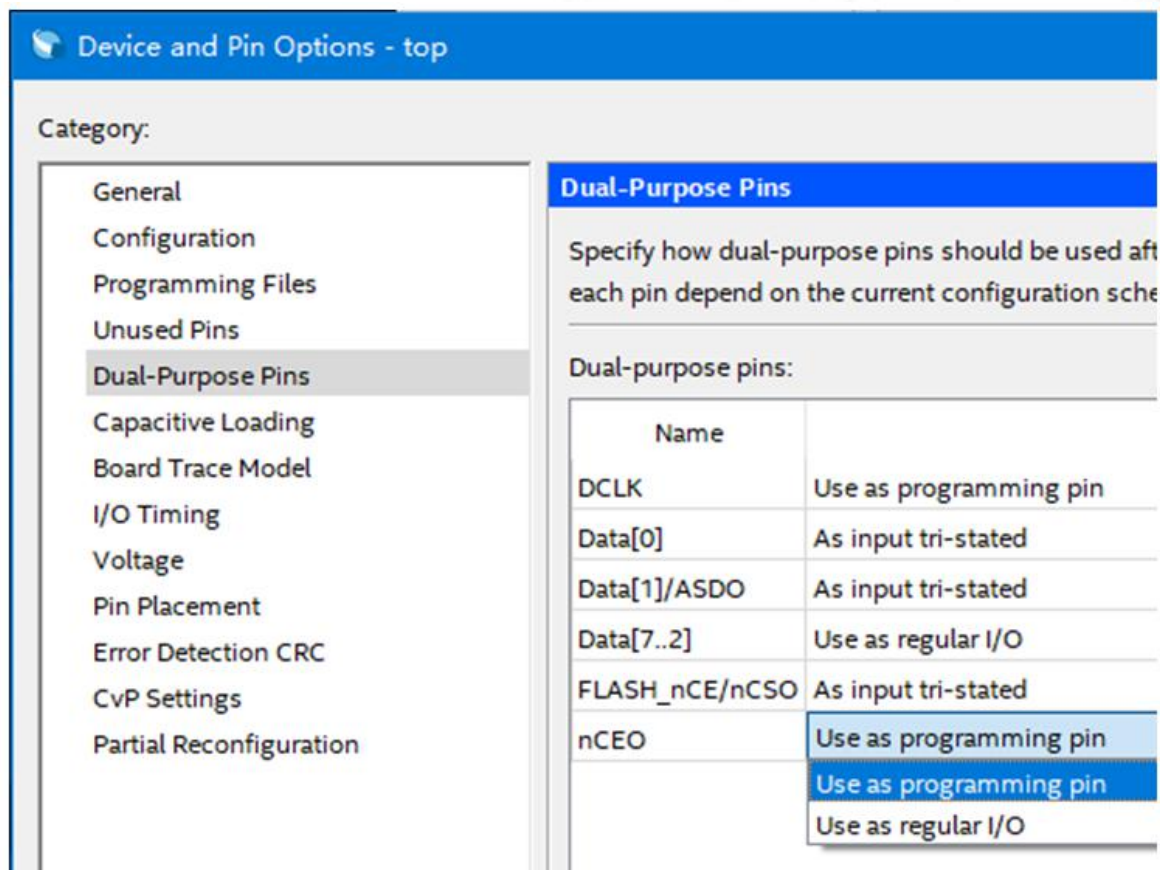
(3) 选择配置器件和编程方式。

选择配置器件型号和压缩方式

原理图输入法逻辑电路设计

4. 编译前设置

(4) 双功能输入输出端口设置。



(4) 双功能输入输出端口设置。

指定 nCEO 脚为普通 I/O 端口

原理图输入法逻辑电路设计

5. 全程编译

Quartus Prime Standard Edition - D:/MY_PROJECT/top - top

File Edit View Project Assignments Processing Tools Window Help

top

Project Navigator Hierarchy top.bdf Compilation Report - top

Entity:Instance
Cyclone 10 LP: 10CL006YU256C8G
top

Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global Settings
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Flow Messages
Flow Suppressed Messages
Assembler
TimeQuest Timing Analyzer

Flow Summary
<<Filter>>
Flow Status Successful - Fri Dec 14
Quartus Prime Version 17.1.0 Build 590 10/25
Revision Name top
Top-level Entity Name top
Family Cyclone 10 LP
Device 10CL006YU256C8G
Timing Models Final
Total logic elements 9 / 6,272 (< 1 %)
Total registers 0
Total pins 12 / 177 (7 %)
Total virtual pins 0
Total memory bits 0 / 276,480 (0 %)
Embedded Multiplier 9-bit elements 0 / 30 (0 %)
Total PLLs 0 / 2 (0 %)

Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmin
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)

All <<Filter>> Find... Find Next

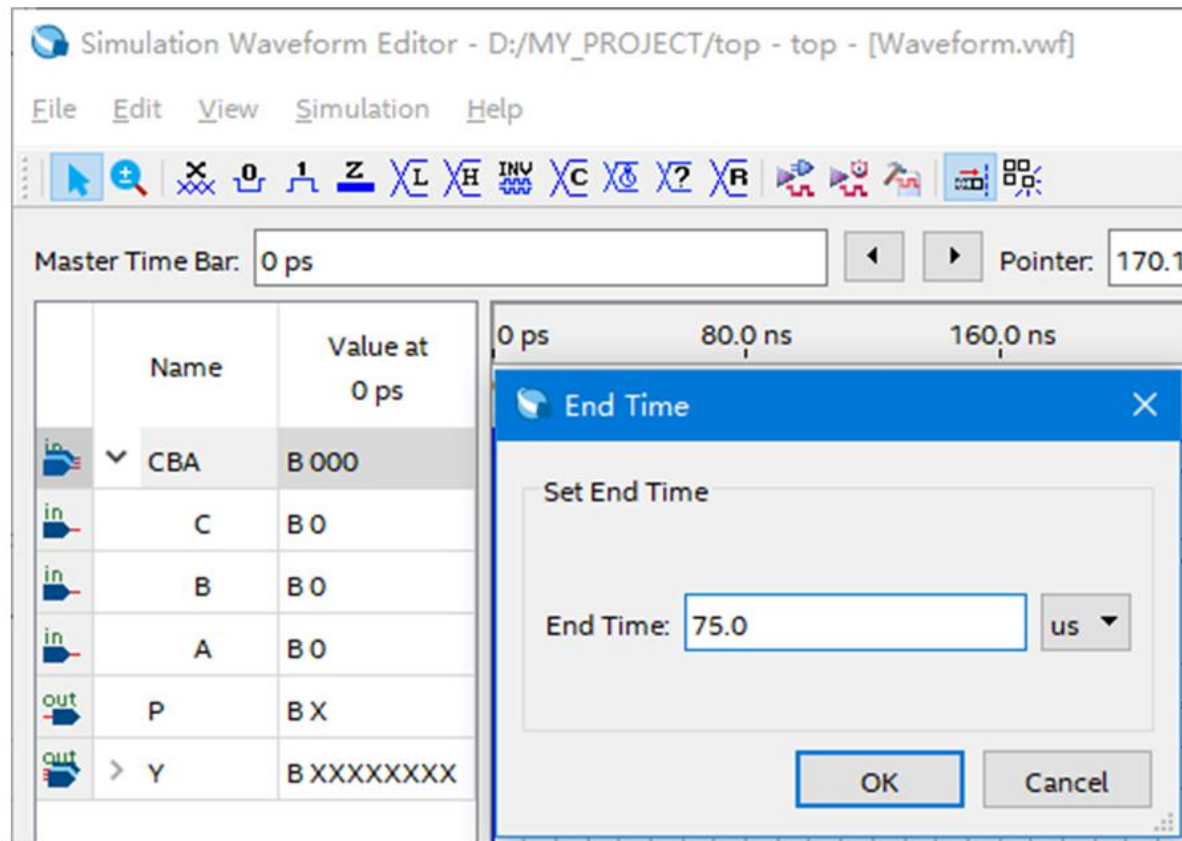
Type ID Message
> i Quartus Prime TimeQuest Timing Analyzer was successful. 0 errors, 8 warnings
i 293000 Quartus Prime Full Compilation was successful. 0 errors, 17 warnings

全程编译后工程管理窗的情况

原理图输入法逻辑电路设计

6. 逻辑功能测试

- (1) 打开波形编辑器。
- (2) 设置仿真时间区域。
- (3) 波形文件存盘。

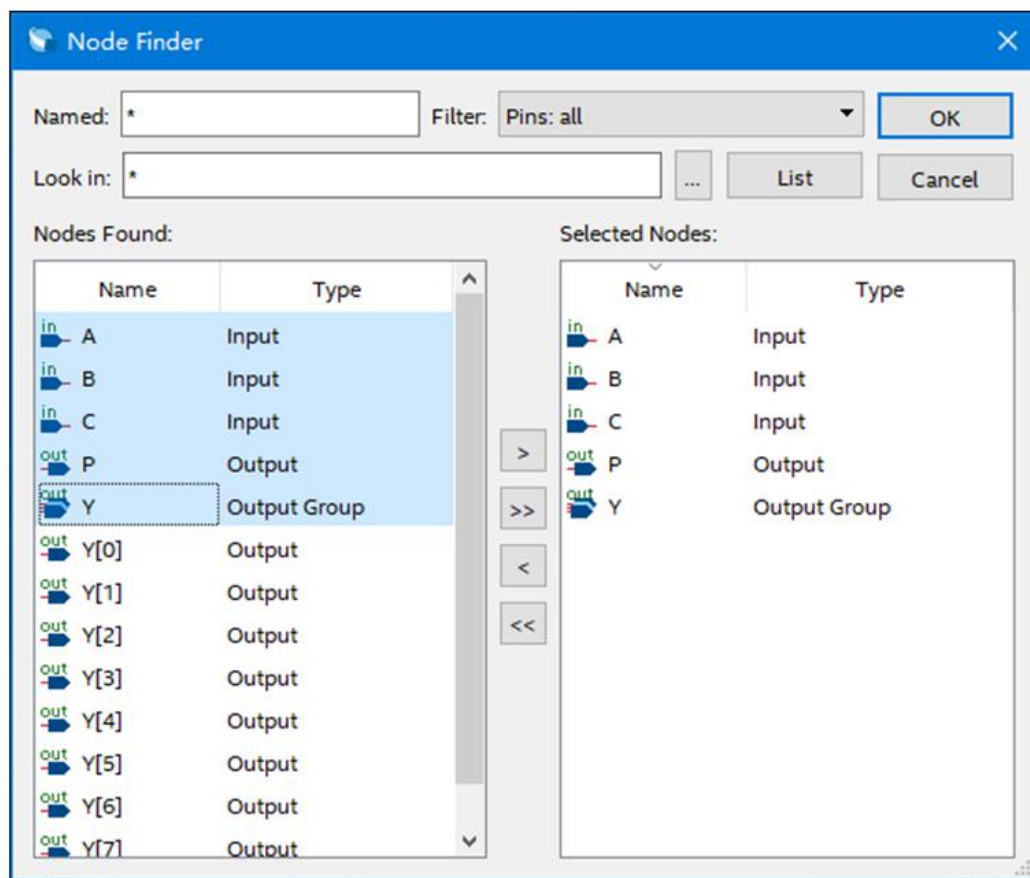


设置仿真时间长度

原理图输入法逻辑电路设计

6. 逻辑功能测试

(4) 将工程top的端口信号名选入波形编辑器中。

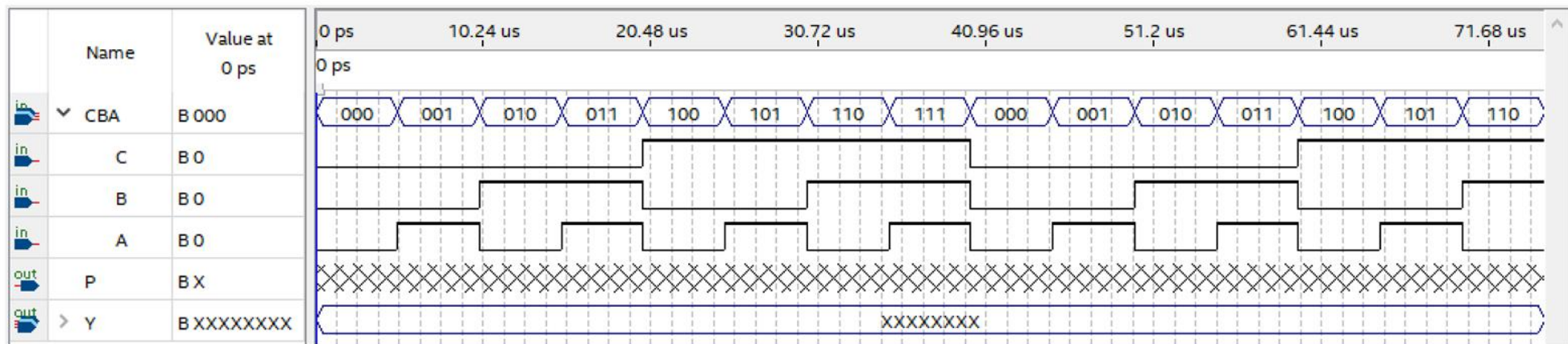


从 Node Finder 窗向波形编辑器拖入信号节点

原理图输入法逻辑电路设计

6. 逻辑功能测试

(5) 编辑输入波形（输入激励信号）。



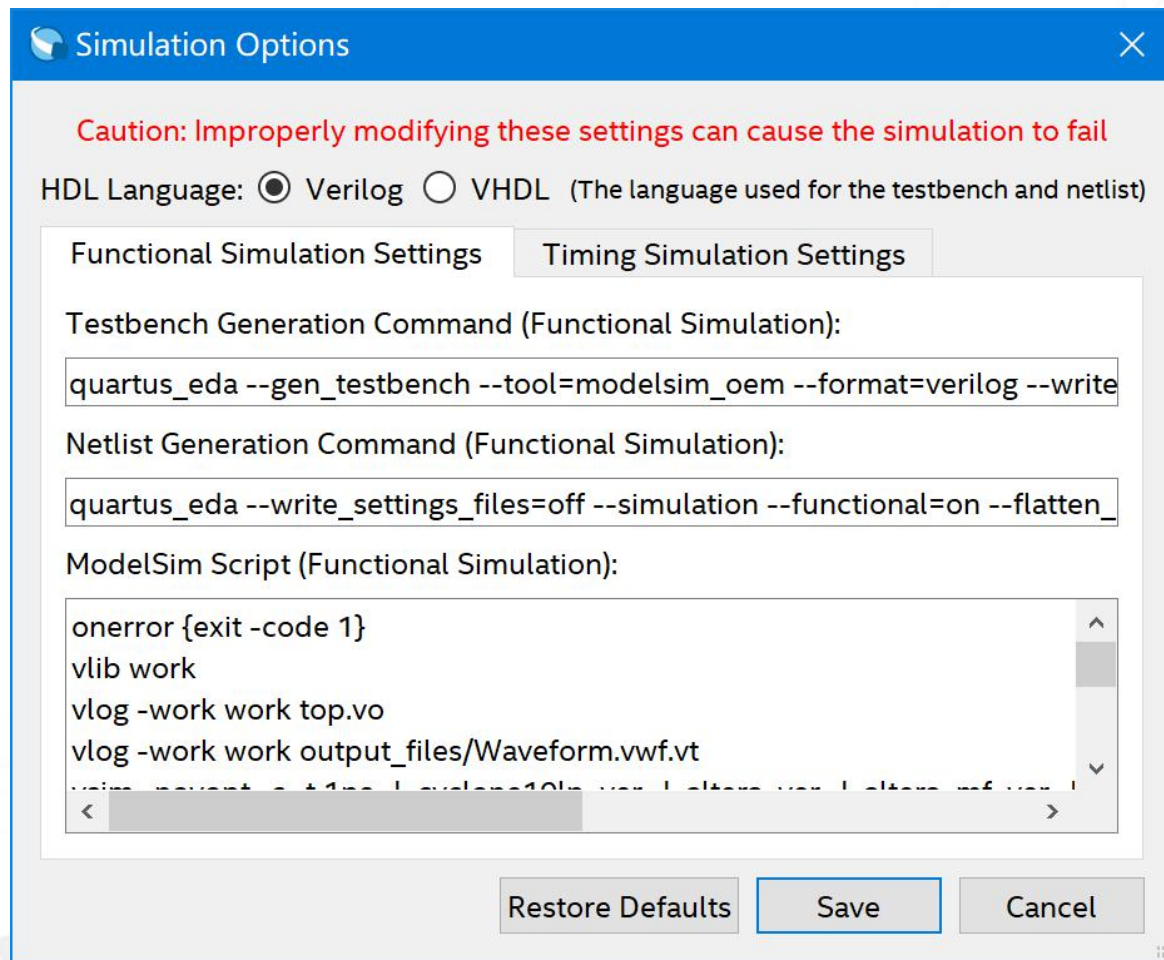
设置好的仿真文件激励波形图

原理图输入法逻辑电路设计

6. 逻辑功能测试

(6) 仿真器参数设置。

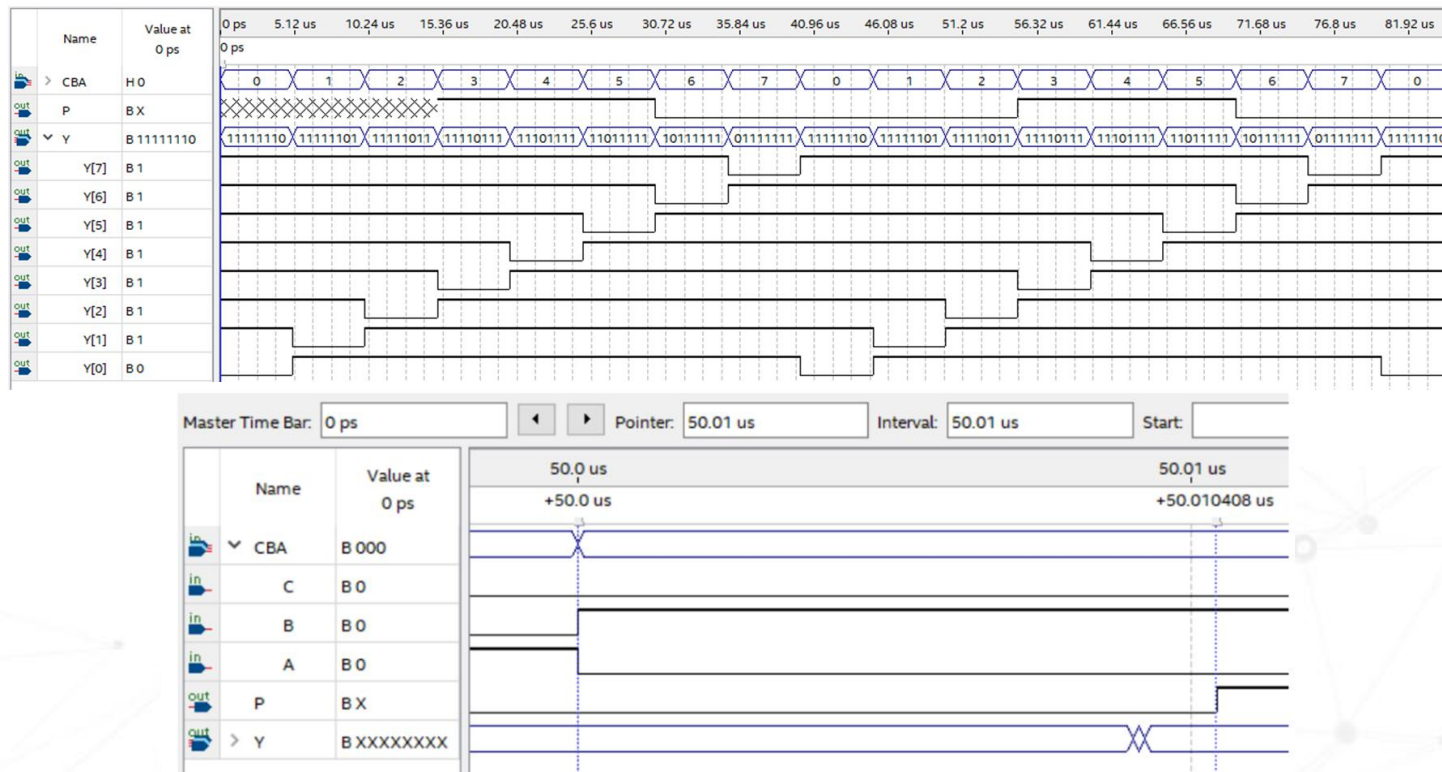
(7) 启动仿真器。



原理图输入法逻辑电路设计

6. 逻辑功能测试

(8) 观察分析仿真结果。



输入 A 与输出 P 的延时波形显示。

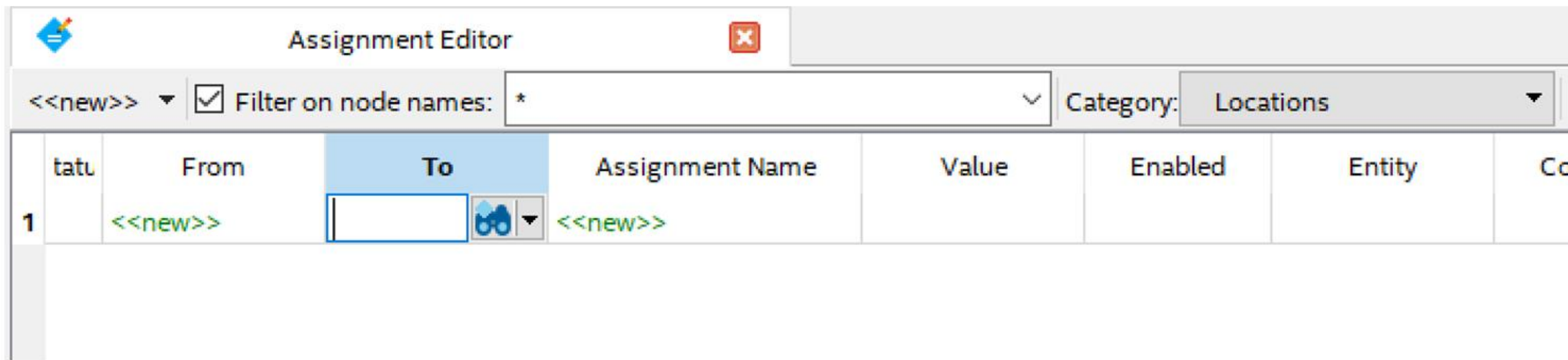
- 6.1 手工数字技术存在的问题
- 6.2 数字系统自动设计流程
- 6.3 原理图输入法逻辑电路设计

• **6.4 硬件设计**

- 6.5 用HDL来表述广义译码器
- 6.6 数字方法去抖动和延时电路设计

1. 引脚锁定

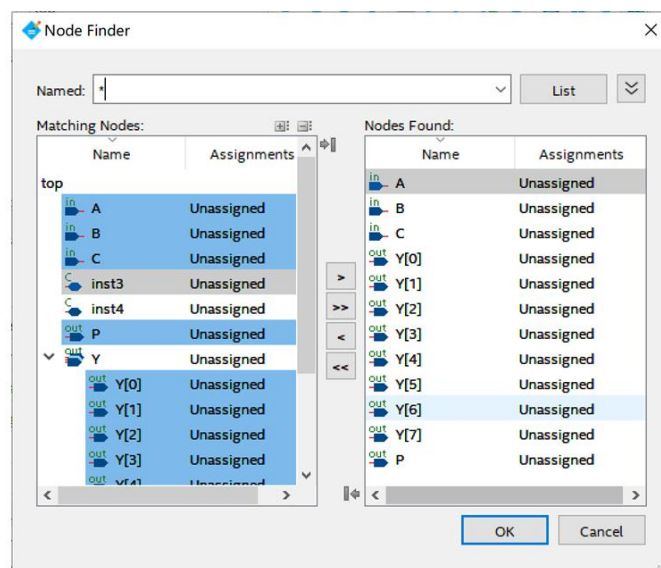
(1) 打开工程。



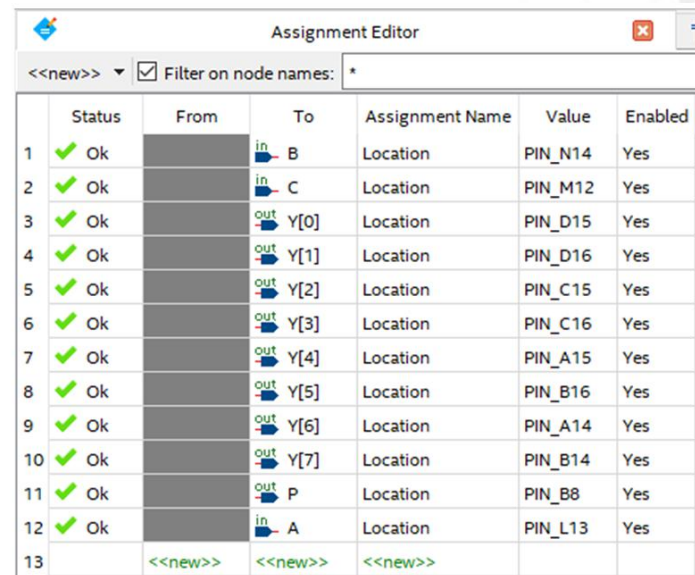
利用 Assignment Editor 编辑器锁定 FPGA 引脚

1. 引脚锁定

(2) 双击“**To**”栏的<<new>>, 即出现一个望远镜按钮, 点击此按钮, 弹出**Node Finder**对话框。



选择需要锁定的引脚信号

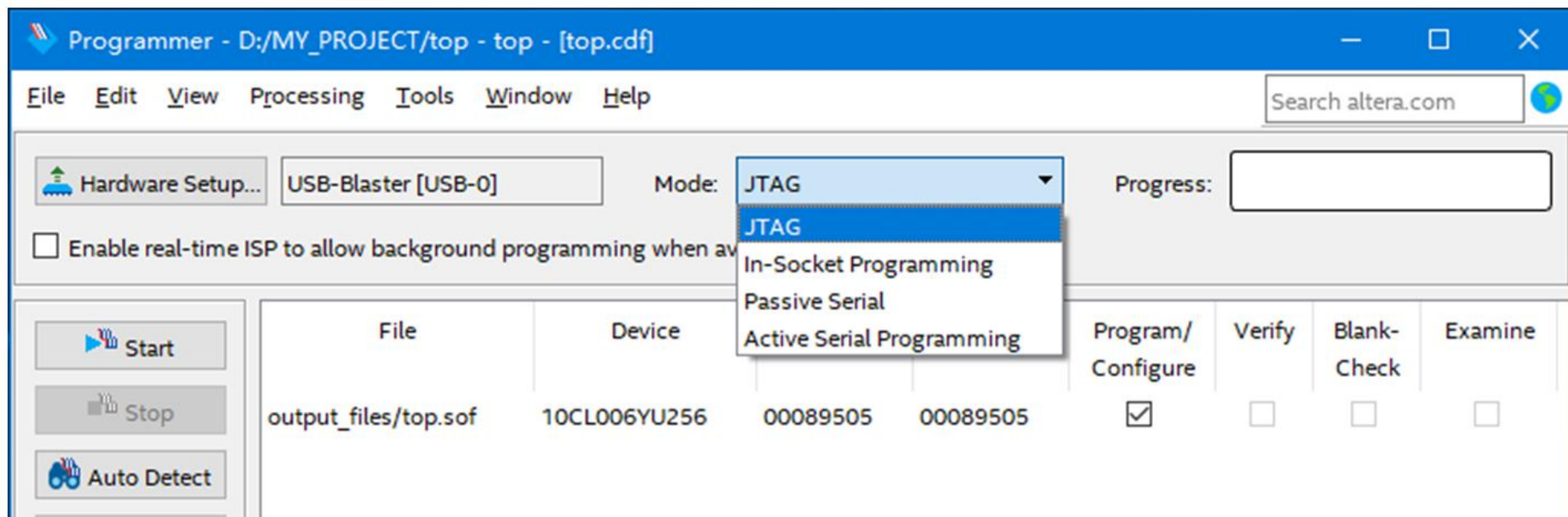


引脚锁定对话框

(3) 最后保存这些引脚锁定的信息后, 必须再编译 (启动**Start Compilation**) 一次, 才能将引脚锁定信息编译进编程下载文件中。

2. 对FPGA编程配置

(1) 打开编程窗。

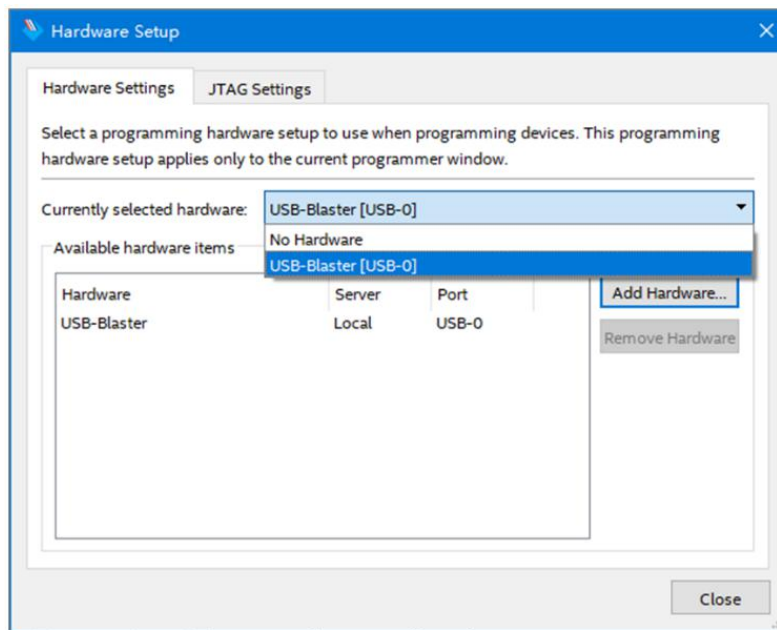


选择编程下载文件和下载模式

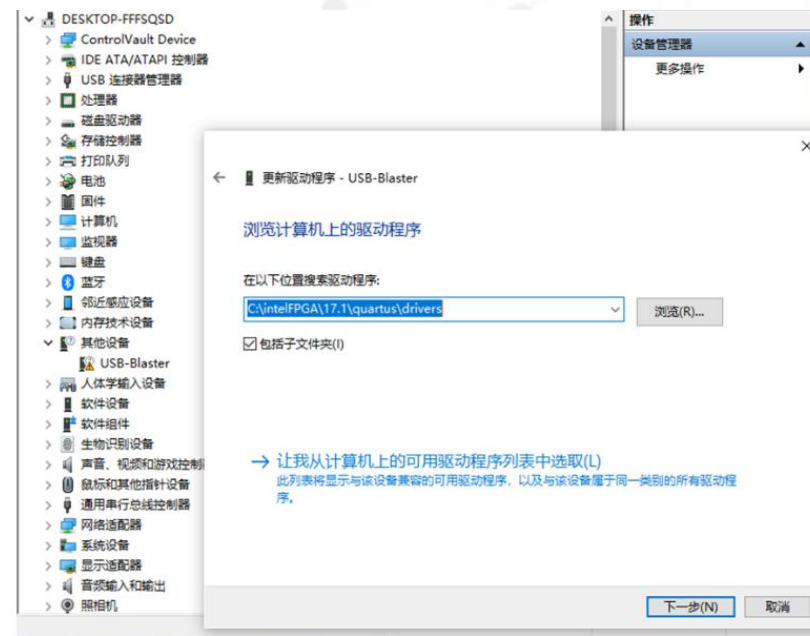
硬件测试

2. 对FPGA编程配置

- (2) 设置编程器。
- (3) 测试JTAG口。
- (4) 硬件测试。



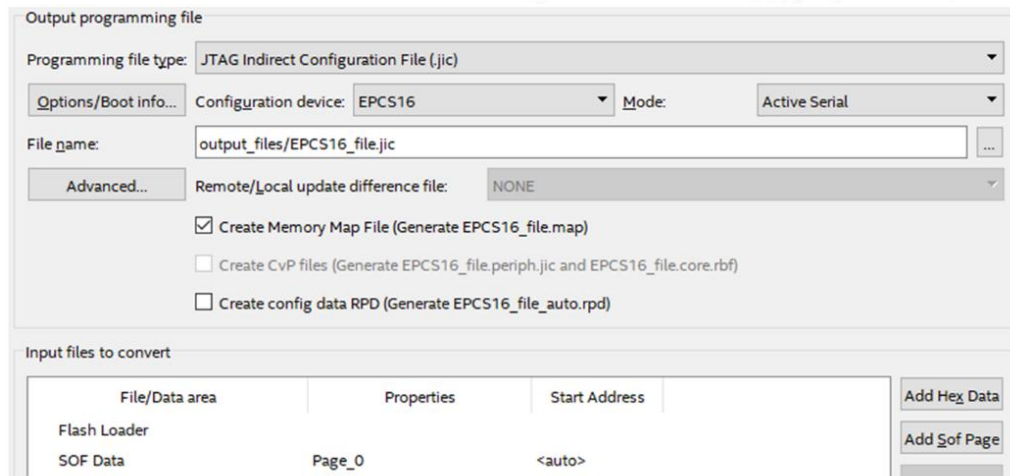
加入编程下载方式



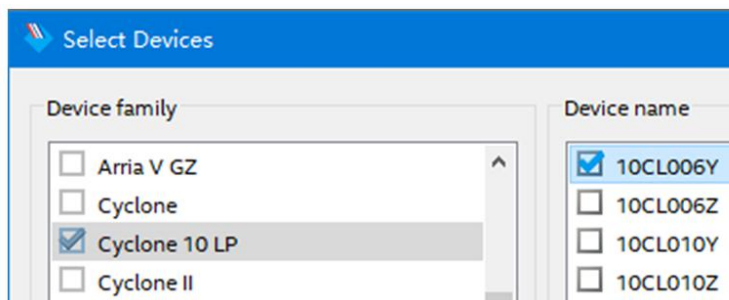
安装编程器驱动

3. 对FPGA配置器件编程

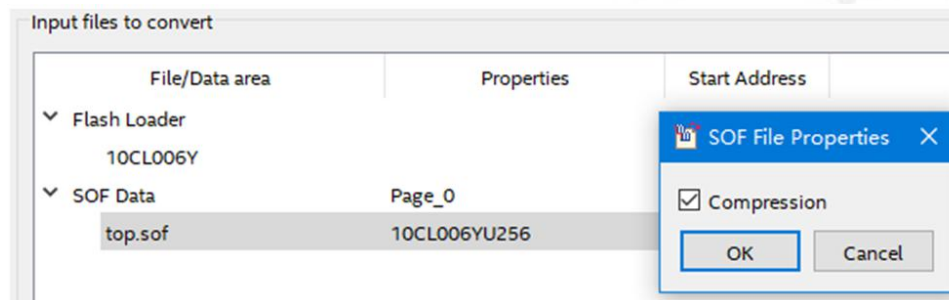
(1) 将SOF文件转化为JTAG间接配置文件。



设定 JTAG 间接编程文件



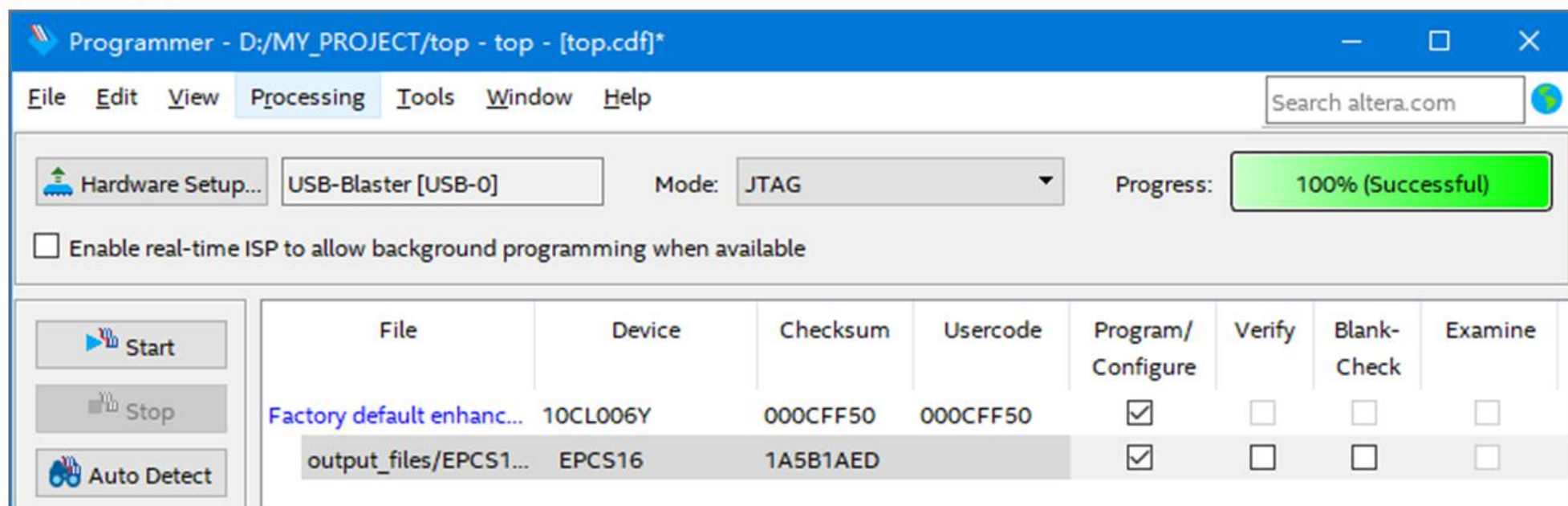
选择目标器件 10CL006Y



选定 SOF 文件后，选择文件压缩

3. 对FPGA配置器件编程

(2) 下载JTAG间接配置文件。



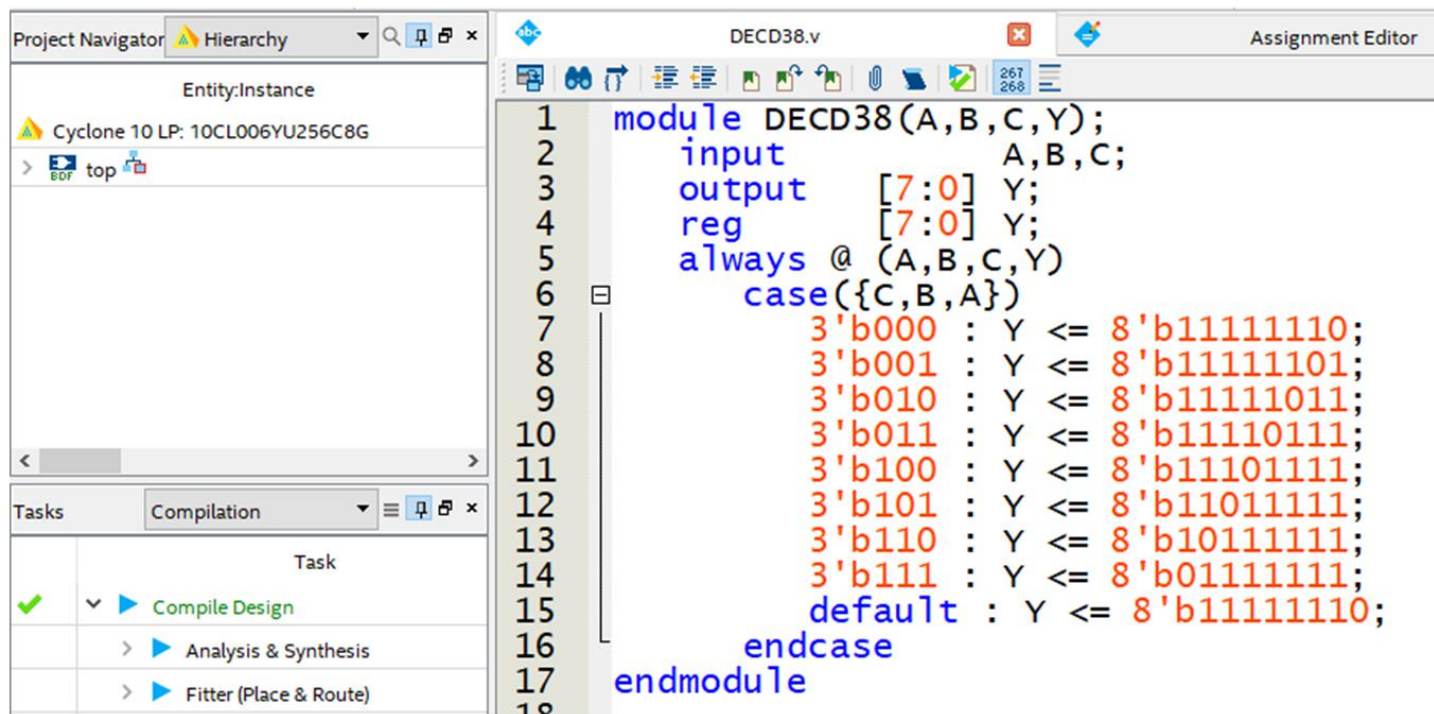
用 JTAG 模式经由 FPGA 对配置器件 EPCS16 进行间接编程

- 6.1 手工数字技术存在的问题
- 6.2 数字系统自动设计流程
- 6.3 原理图输入法逻辑电路设计
- 6.4 硬件设计
- **6.5 用HDL来表述广义译码器**
- 6.6 数字方法去抖动和延时电路设计

用HDL来表述广义译码器

1. 用HDL表述真值表及电路设计

(1) HDL表述



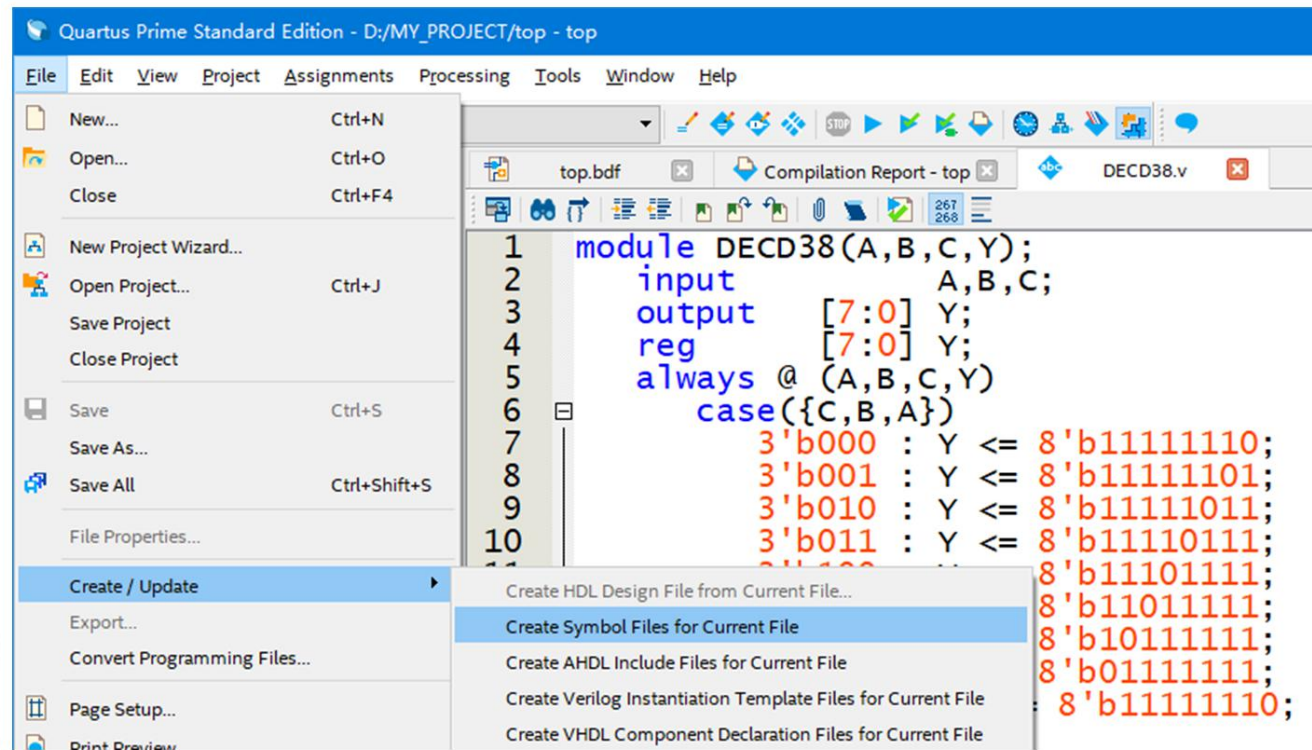
```
1 module DECD38(A,B,C,Y);
2   input      A,B,C;
3   output     [7:0] Y;
4   reg        [7:0] Y;
5   always @ (A,B,C,Y)
6     case({C,B,A})
7       3'b000 : Y <= 8'b11111110;
8       3'b001 : Y <= 8'b11111101;
9       3'b010 : Y <= 8'b11111011;
10      3'b011 : Y <= 8'b11110111;
11      3'b100 : Y <= 8'b11101111;
12      3'b101 : Y <= 8'b11011111;
13      3'b110 : Y <= 8'b10111111;
14      3'b111 : Y <= 8'b01111111;
15      default : Y <= 8'b11111110;
16    endcase
17 endmodule
```

3-8 译码器真值表对应的 HDL 的 case 语句表述

用HDL来表述广义译码器

1. 用HDL表述真值表及电路设计

(2) 将Verilog文本表述转化为电路元件



The screenshot shows the Quartus Prime Standard Edition interface. The main window displays a Verilog HDL code for a decoder module named DECD38. The code defines three 3-bit inputs (A, B, C) and an 8-bit output (Y). It uses a case statement to map the input combinations to the output bits. A context menu is open over the code, with the option 'Create Symbol Files for Current File' selected. The menu also includes options like 'Create HDL Design File from Current File...', 'Create AHDL Include Files for Current File', 'Create Verilog Instantiation Template Files for Current File', and 'Create VHDL Component Declaration Files for Current File'.

```
1 module DECD38(A,B,C,Y);
2     input      A,B,C;
3     output [7:0] Y;
4     reg [7:0] Y;
5     always @ (A,B,C,Y)
6         case({C,B,A})
7             3'b000 : Y <= 8'b11111110;
8             3'b001 : Y <= 8'b11111101;
9             3'b010 : Y <= 8'b11111011;
10            3'b011 : Y <= 8'b11110111;
11            3'b100 : Y <= 8'b11110111;
12            3'b101 : Y <= 8'b11101111;
13            3'b110 : Y <= 8'b11011111;
14            3'b111 : Y <= 8'b10111111;
15            3'b100 : Y <= 8'b01111111;
16            3'b100 : Y <= 8'b11111110;
```

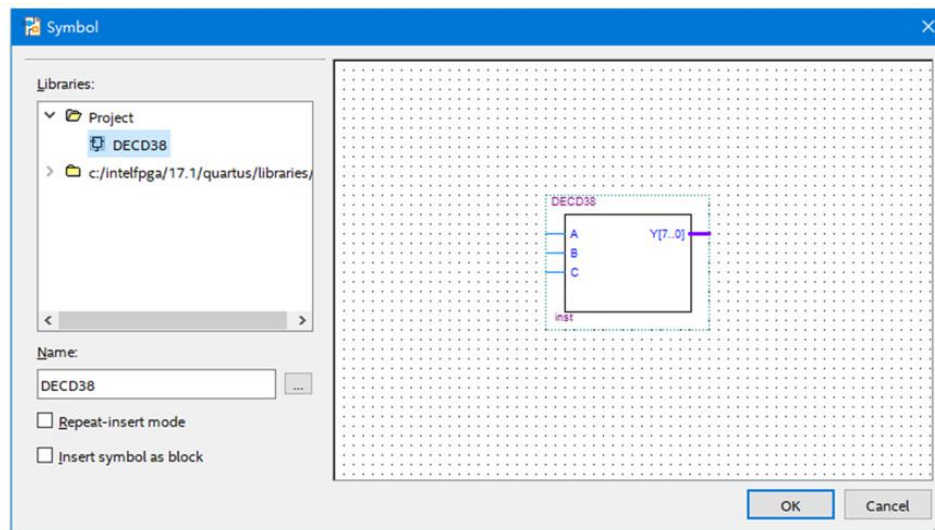
将 HDL 程序变成一个可以调用的原件模块

用HDL来表述广义译码器

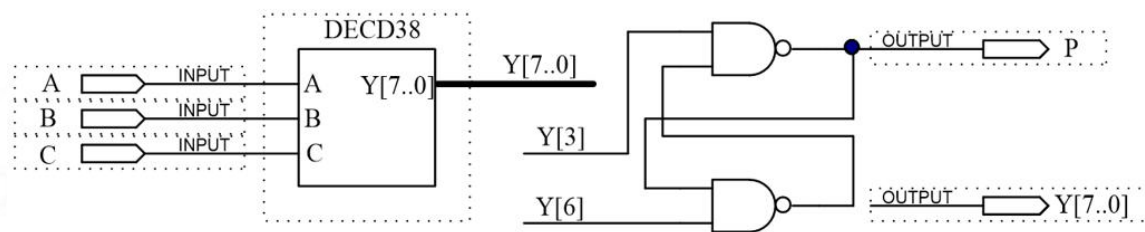
1. 用HDL表述真值表及电路设计

(3) 完成电路设计

(4) 逻辑功能测试



选择已生成好的元件 DECD38



用 HDL 文本表述的 3-8 译码器 DECD38 连接好电路

用HDL来表述广义译码器

2. 三人表决电路的HDL表述方式

```
1 module JG3(A,B,C,X,Y);
2   input  A,B,C;
3   output X,Y;
4   reg    X,Y;
5   always @ (A,B,C,X,Y)
6     case({A,B,C})
7       3'b000 : begin X <= 1'b0; Y <= 1'b1; end
8       3'b001 : begin X <= 1'b0; Y <= 1'b0; end
9       3'b010 : begin X <= 1'b0; Y <= 1'b0; end
10      3'b011 : begin X <= 1'b0; Y <= 1'b0; end
11      3'b100 : begin X <= 1'b0; Y <= 1'b0; end
12      3'b101 : begin X <= 1'b1; Y <= 1'b0; end
13      3'b110 : begin X <= 1'b1; Y <= 1'b0; end
14      3'b111 : begin X <= 1'b1; Y <= 1'b0; end
15      default : begin X <= 1'b1; Y <= 1'b0; end
16    endcase
17 endmodule
```

(a)

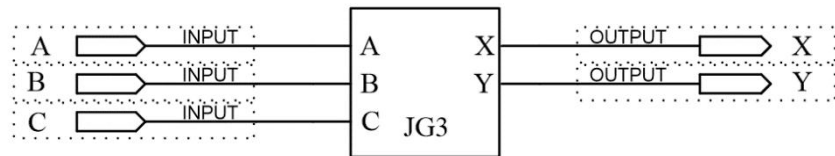
```
1 module JG3(A,B,C,X,Y);
2   input  A,B,C;
3   output X,Y;
4   reg    X,Y;
5   always @ (A,B,C,X,Y)
6     case({A,B,C})
7       3'b000 : {X,Y} <= 2'b01;
8       3'b001 : {X,Y} <= 2'b00;
9       3'b010 : {X,Y} <= 2'b00;
10      3'b011 : {X,Y} <= 2'b00;
11      3'b100 : {X,Y} <= 2'b00;
12      3'b101 : {X,Y} <= 2'b10;
13      3'b110 : {X,Y} <= 2'b10;
14      3'b111 : {X,Y} <= 2'b10;
15      default : {X,Y} <= 2'b10;
16    endcase
17 endmodule
```

(b)

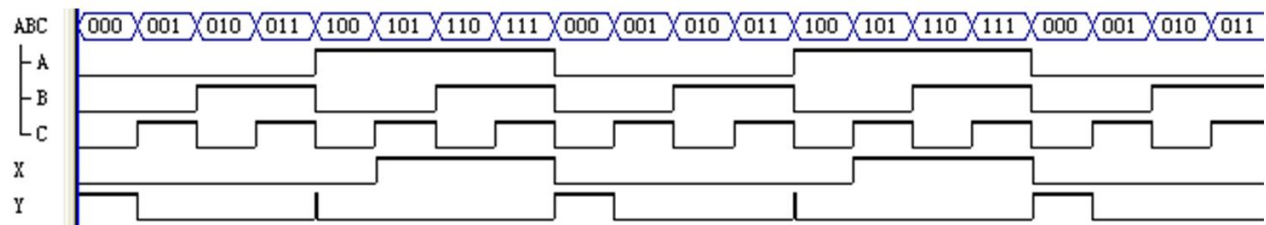
表 4-3 对应的两种 case 语句的表述方式

用HDL来表述广义译码器

2. 三人表决电路的HDL表述方式



表决电路测试电路原理图



Quartus II 9.1 下三人表决电路的仿真波形

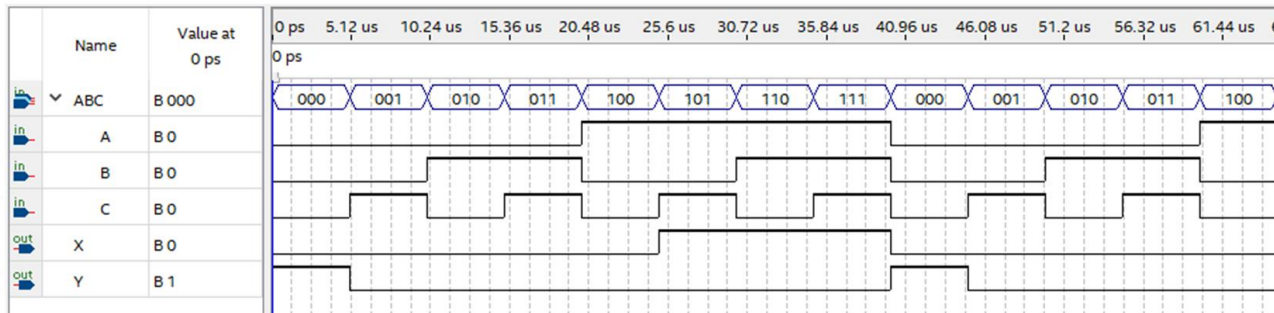


图 6-41 Quartus Prime 17.1 下三人表决电路的仿真波形

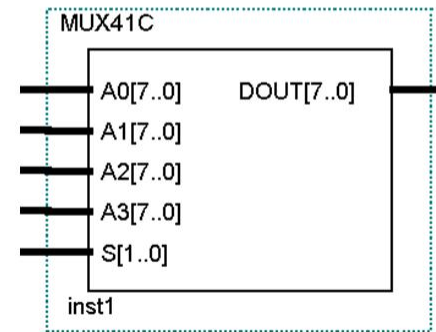
用HDL来表述广义译码器

3. 用HDL对真值表的其它表述方式

(1) 文字表达方式的多路选择器设计

```
MUX41C.v
1  module MUX41C (A0,A1,A2,A3,S,DOUT);
2      input  [7:0] A0,A1,A2,A3;
3      input  [1:0] S;
4      output [7:0]DOUT;
5      reg [7:0] DOUT;
6      always @ (A0,A1,A2,A3,S,DOUT)
7          case(S)
8              2'b00 : DOUT <= A0;
9              2'b01 : DOUT <= A1;
10             2'b10 : DOUT <= A2;
11             2'b11 : DOUT <= A3;
12             default : DOUT <= 8'b00000000;
13         endcase
14     endmodule
```

(a) 8位4选1型多路选择器的case语句描述



(b) 由(a)代码生成的原理图元件符号

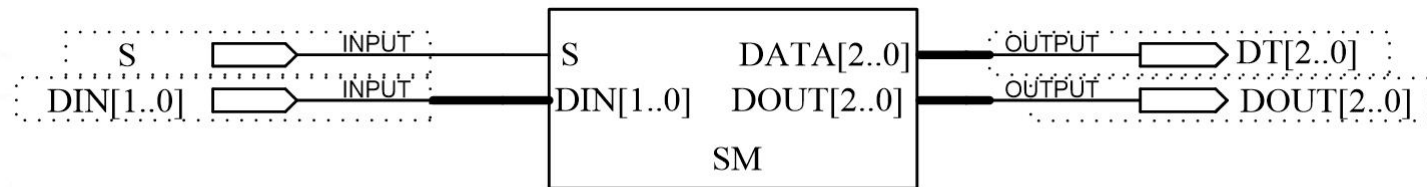
用HDL来表述广义译码器

3. 用HDL对真值表的其它表述方式

(2) 含有条件判定情况的真值表的case语句表述

```
SM.v
1 module SM (S,DIN,DATA,DOUT);
2   input
3   input [1:0] DIN;
4   output [2:0] DATA,DOUT;
5   reg [2:0] DATA,DOUT;
6   always @ (S,DIN,DATA)
7     case(DIN)
8       2'b00 : begin DOUT <= 3'b111; DATA <= 3'b001; end
9       2'b01 : begin DOUT <= 3'b010; if (S == 1'b1) DATA <= 3'b110; else DATA <= 3'b011; end
10      2'b10 : begin DOUT <= 3'b111; DATA <= 3'b000; end
11      2'b11 : begin DOUT <= 3'b111; DATA <= 3'b111; end
12      default : DOUT <= 3'b111;
13    endcase
14 endmodule
```

含条件判断情况的“真值表”表达样本

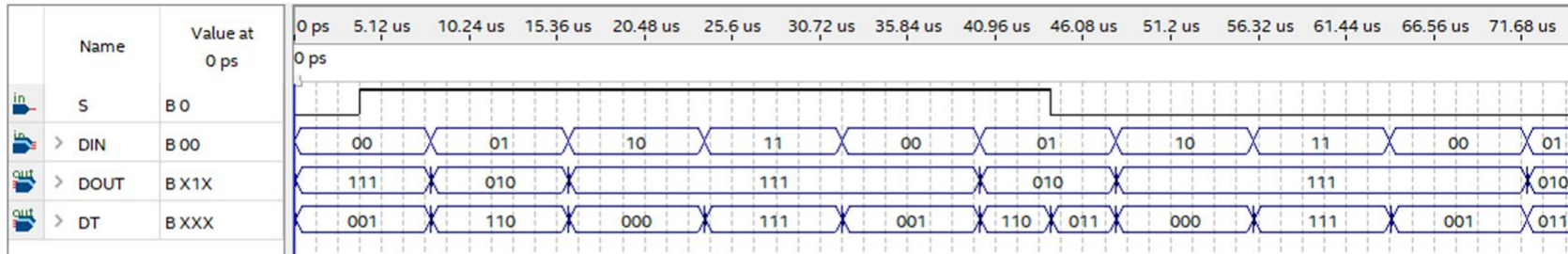


对应程序的测试电路原理图

用HDL来表述广义译码器

3. 用HDL对真值表的其它表述方式

(2) 含有条件判定情况的真值表的case语句表述



仿真波形

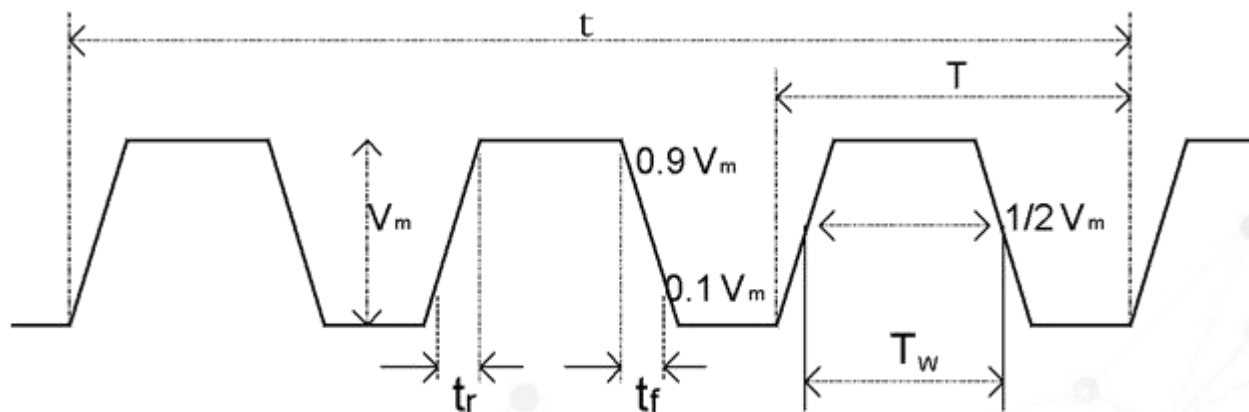
```
begin D<=2'b01; if (S==1) begin A<=2'b10; B<=2'b00; end  
else begin A<=2'b01; B<=2'b11; end ; end
```

- 6.1 手工数字技术存在的问题
- 6.2 数字系统自动设计流程
- 6.3 原理图输入法逻辑电路设计
- 6.4 硬件设计
- 6.5 用HDL来表述广义译码器
- **6.6 数字方法去抖动和延时电路设计**

数字方法去抖动和延时电路设计

1. 数字去抖动电路设计

(1) 脉冲参数和信号频率概念



脉冲波形概念和参数说明图

脉冲周期 T

脉冲频率 f

脉冲幅度 V_m

上升时间 t_r

下降时间 t_f

脉冲宽度 T_w

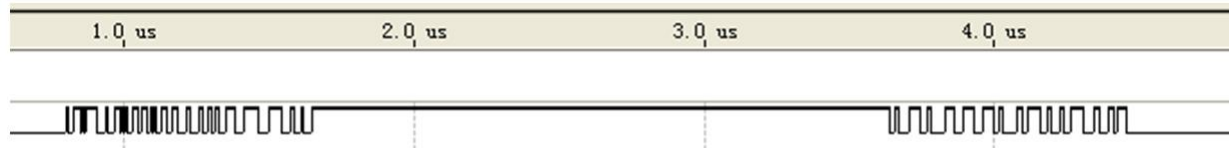
占空比 q

$q = T_w / (T - T_w)$ 或者 $q = \frac{T_w}{T} 100\%$

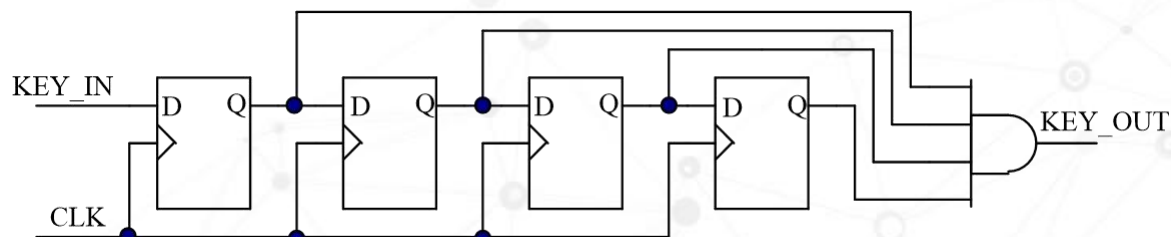
数字方法去抖动和延时电路设计

1. 数字去抖动电路设计

(2) 数字去抖动电路设计



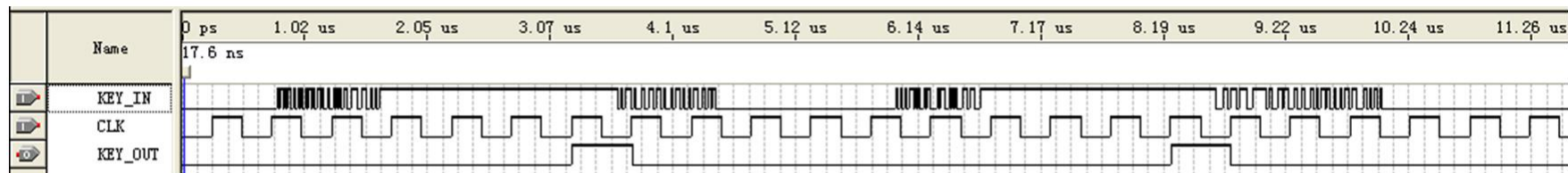
在信号上升与下降沿含随机干扰抖动信号的信号



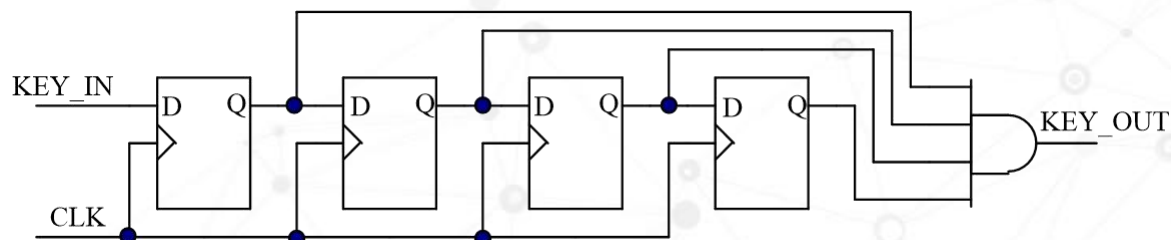
数字方法去抖动和延时电路设计

1. 数字去抖动电路设计

(3) 时序仿真



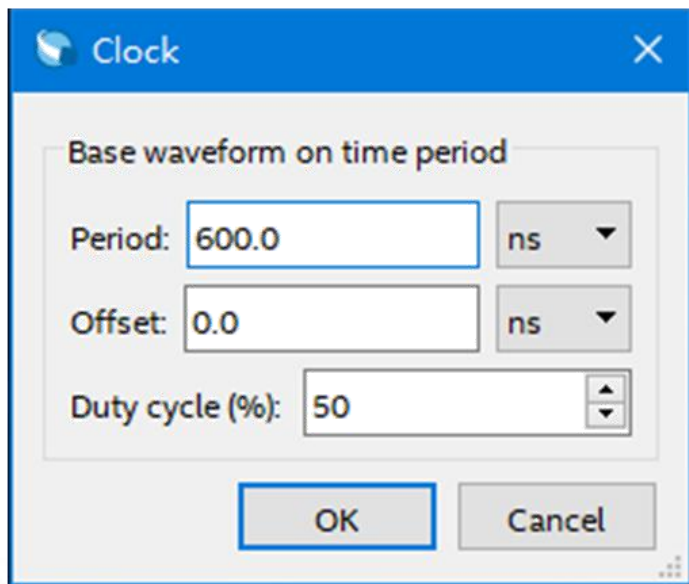
消抖动电路仿真波形



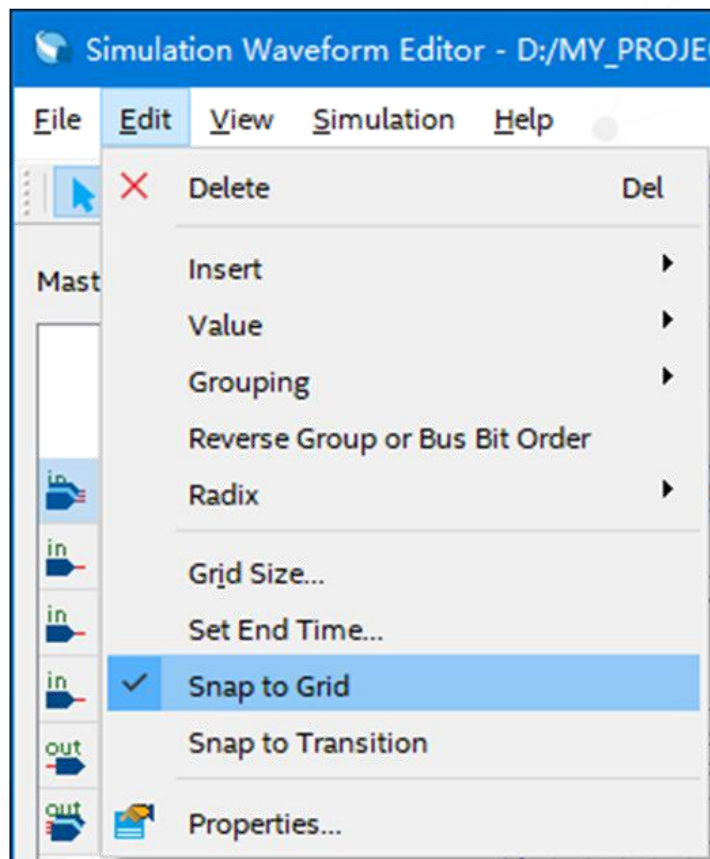
数字方法去抖动和延时电路设计

1. 数字去抖动电路设计

(3) 时序仿真



设置时钟周期

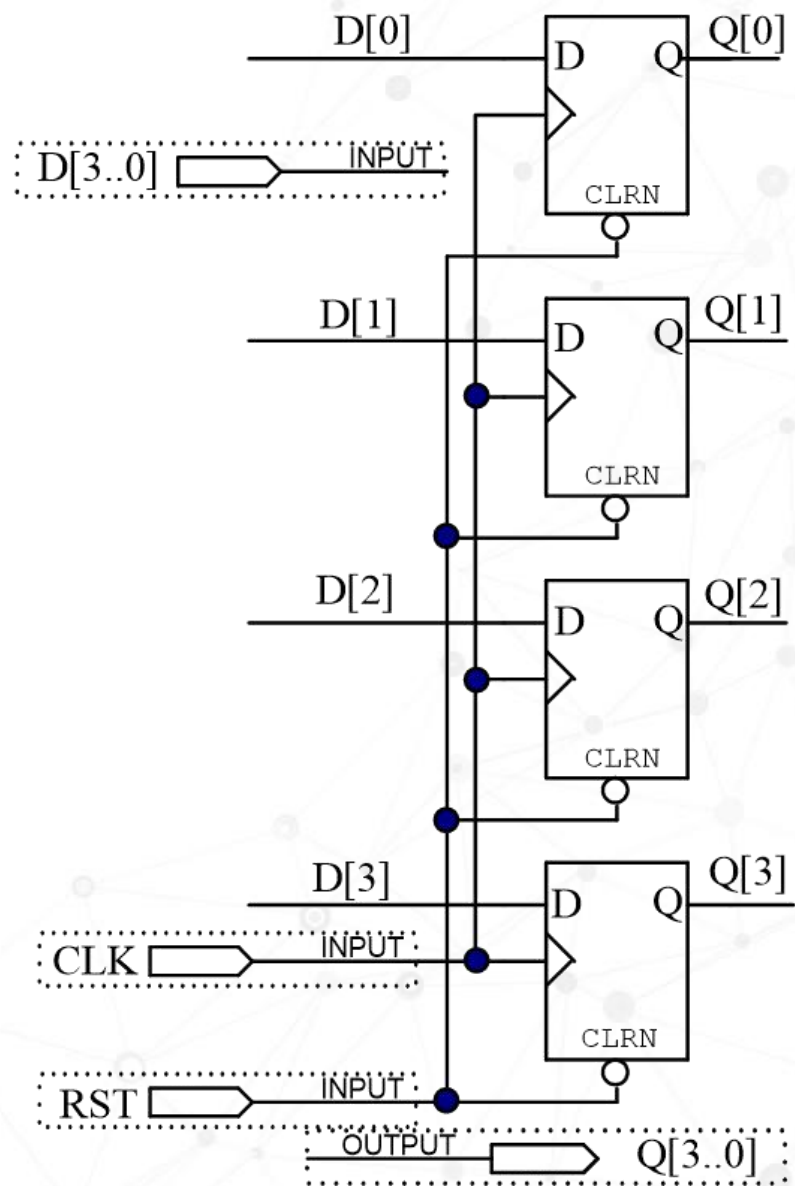


关闭分格限制

数字方法去抖动和延时电路设计

2. 数字延时电路的设计与测试

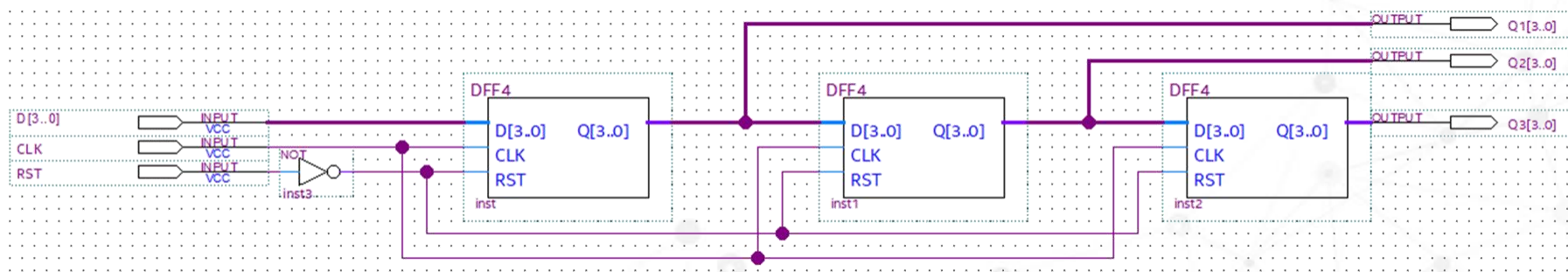
(1) 设计一个库元件（**DFF**四位寄存器）



数字方法去抖动和延时电路设计

2. 数字延时电路的设计与测试

(2) 设计顶层电路（延时测试电路）



数字方法去抖动和延时电路设计

2. 数字延时电路的设计与测试

(3) 时序仿真

	Name	Value at 0 ps
	CLK	B 0
	RST	B 0
	> D	H 5
	> Q1	H X
	> Q2	H X
	> Q3	H X

0 ps 5.12 us 10.24 us 15.3

Count Value

Radix: Hexadecimal

Start value: 5

Increment by: 1

Count type

Binary

Gray code

Transitions occur

Count every: 2.0 us

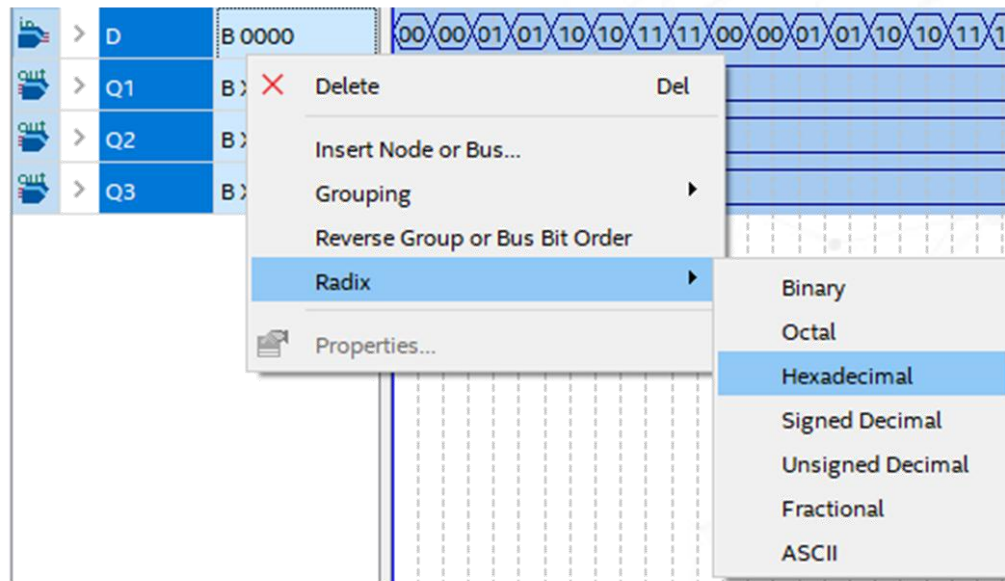
OK Cancel

设置仿真用输入数据和递增型输入数据时间间隔。

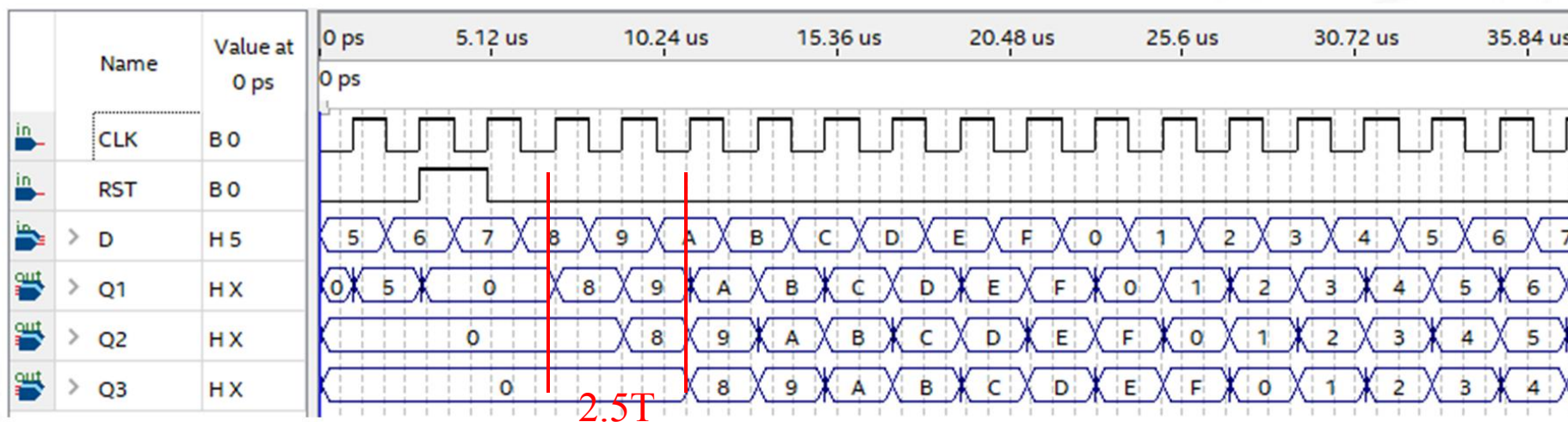
数字方法去抖动和延时电路设计

2. 数字延时电路的设计与测试

(3) 时序仿真



设置仿真信号数据表述格式



延时电路仿真波形

The background features a complex network of white lines connecting various nodes, overlaid on a pattern of semi-transparent, overlapping hexagons in shades of blue, purple, and gold. The overall aesthetic is modern and technical.

本章 完