

第4章

时序仿真与硬件实现

4.1 代码编辑输入和系统编译

4.1.1 编辑和输入设计文件

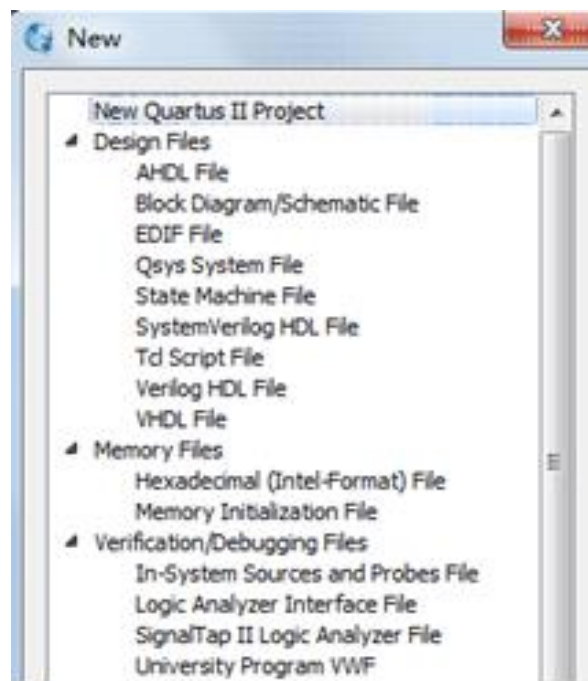


图 4-1 选择编辑文件类型

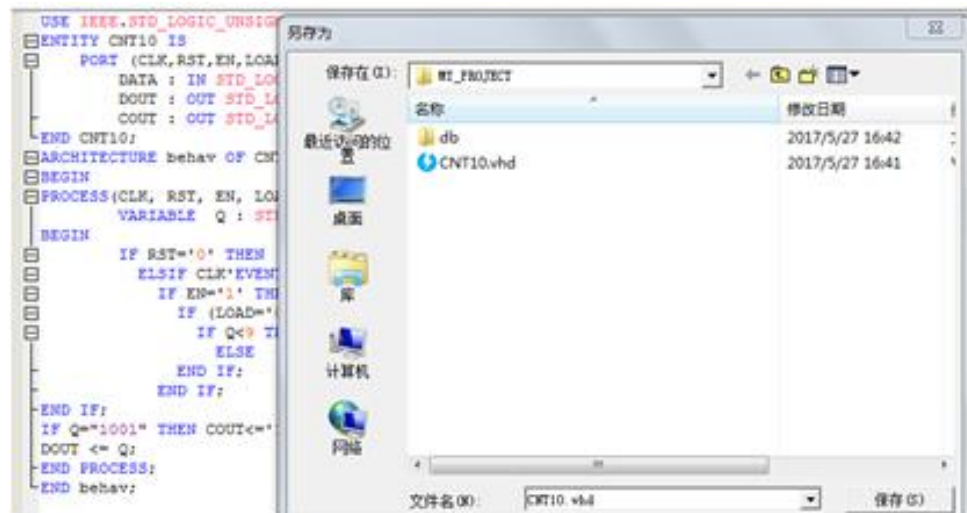


图 4-2 编辑输入源程序并存盘

4.1 代码编辑输入和系统编译

4.1.2 创建工程

(1) 打开并建立新工程管理窗口。

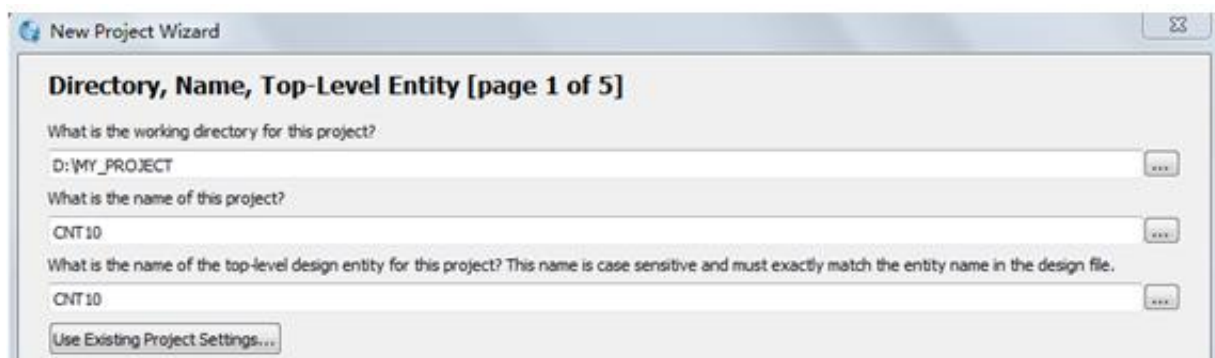


图 4-3 利用 New Project Wizard 创建工程 CNT10

(2) 将设计文件加入工程中。

4.1 代码编辑输入和系统编译

4.1.2 创建工程

(3) 选择目标芯片。

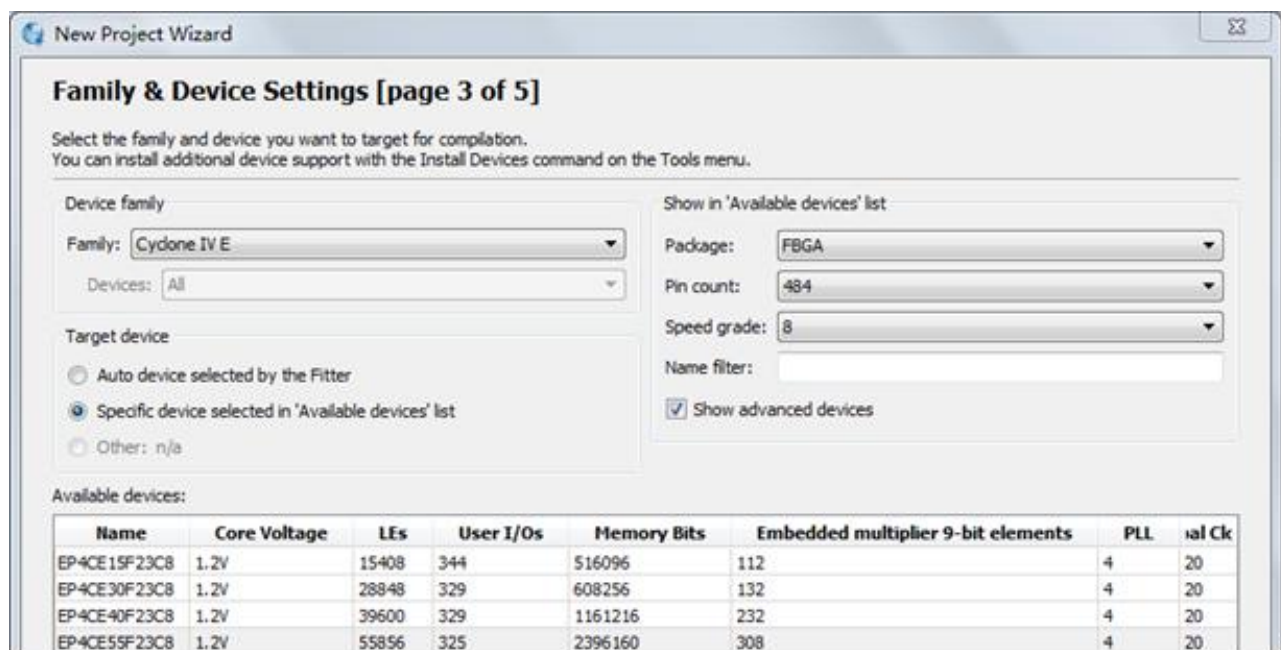


图 4-4 选择目标器件 EP4CE55F23C8

4.1 代码编辑输入和系统编译

4.1.2 创建工程

(4) 工具设置。

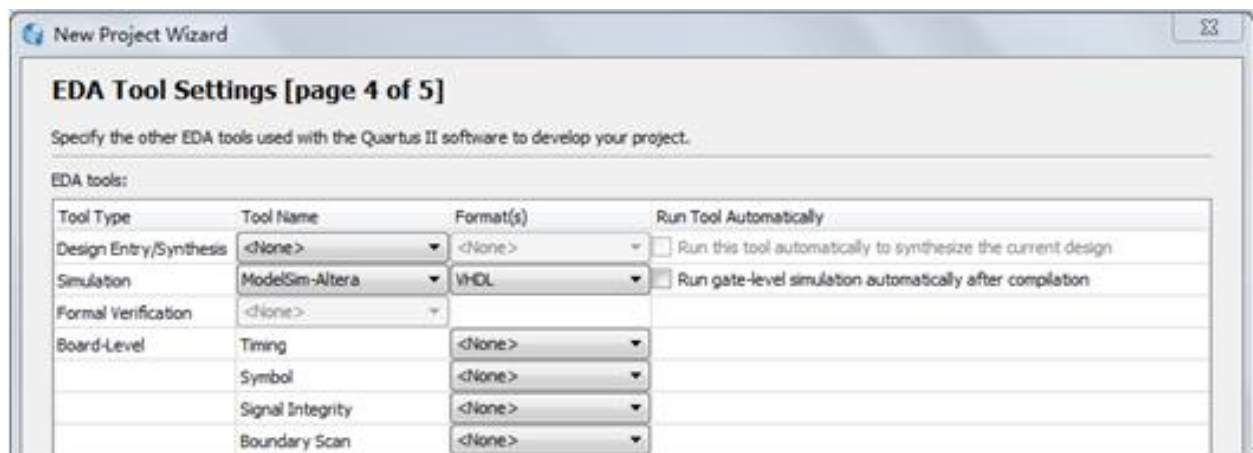


图 4-5 设计与验证工具软件选择

(5) 结束设置。

4.1 代码编辑输入和系统编译

4.1.3 约束项目设置

(1) 选择编译约束条件。

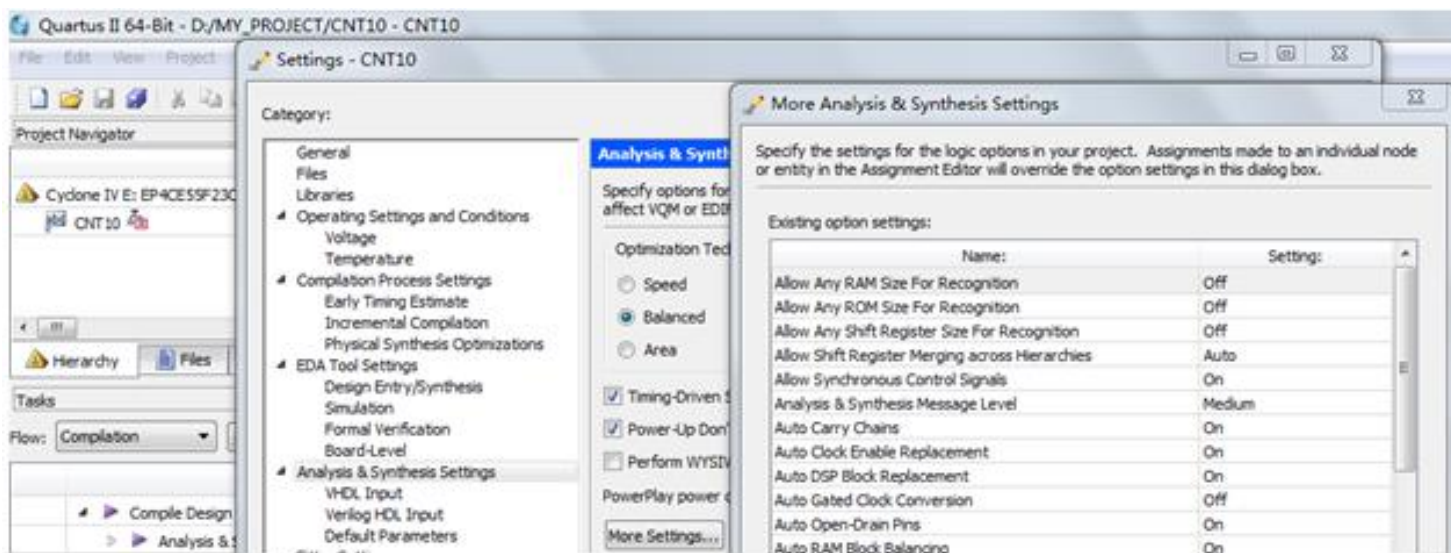


图 4-6 选择编译综合的工作方式

4.1 代码编辑输入和系统编译

4.1.3 约束项目设置

(2) 选择目标芯片的其它控制项。

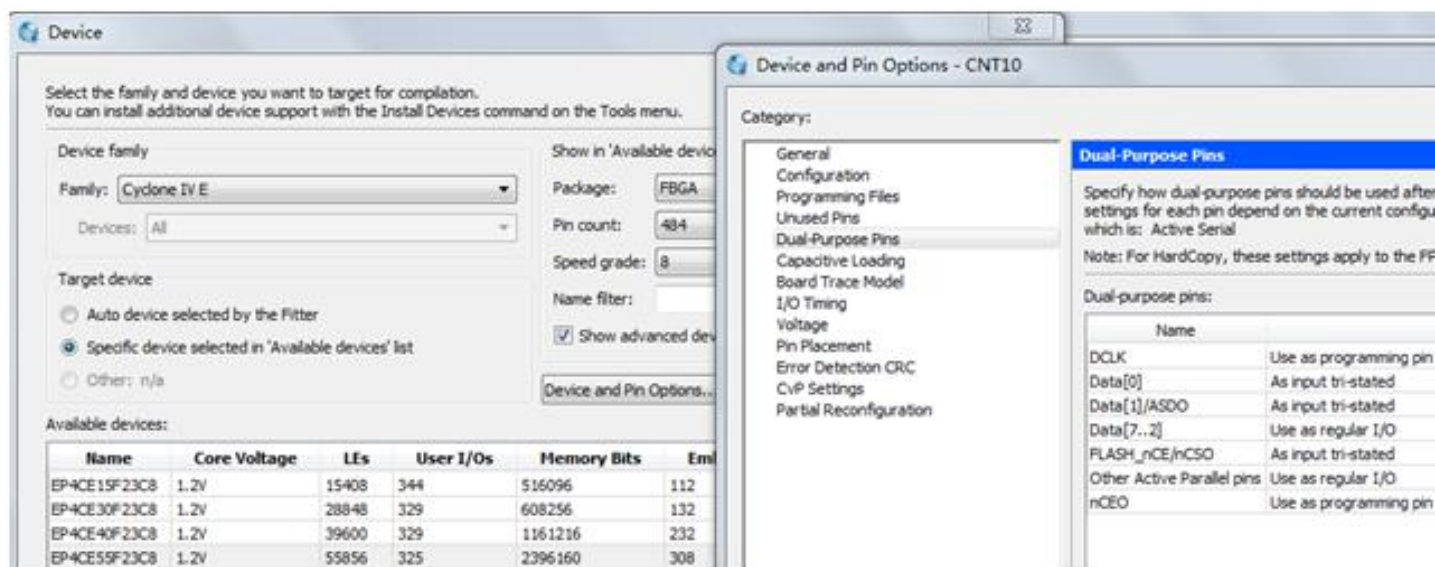


图 4-7 选择目标器件和工作方式

4.1 代码编辑输入和系统编译

4.1.3 约束项目设置

(3) 选择配置器件的工作方式。

(4) 选择目标器件引脚端口状态。

4.1 代码编辑输入和系统编译

4.1.4 全程综合与编译

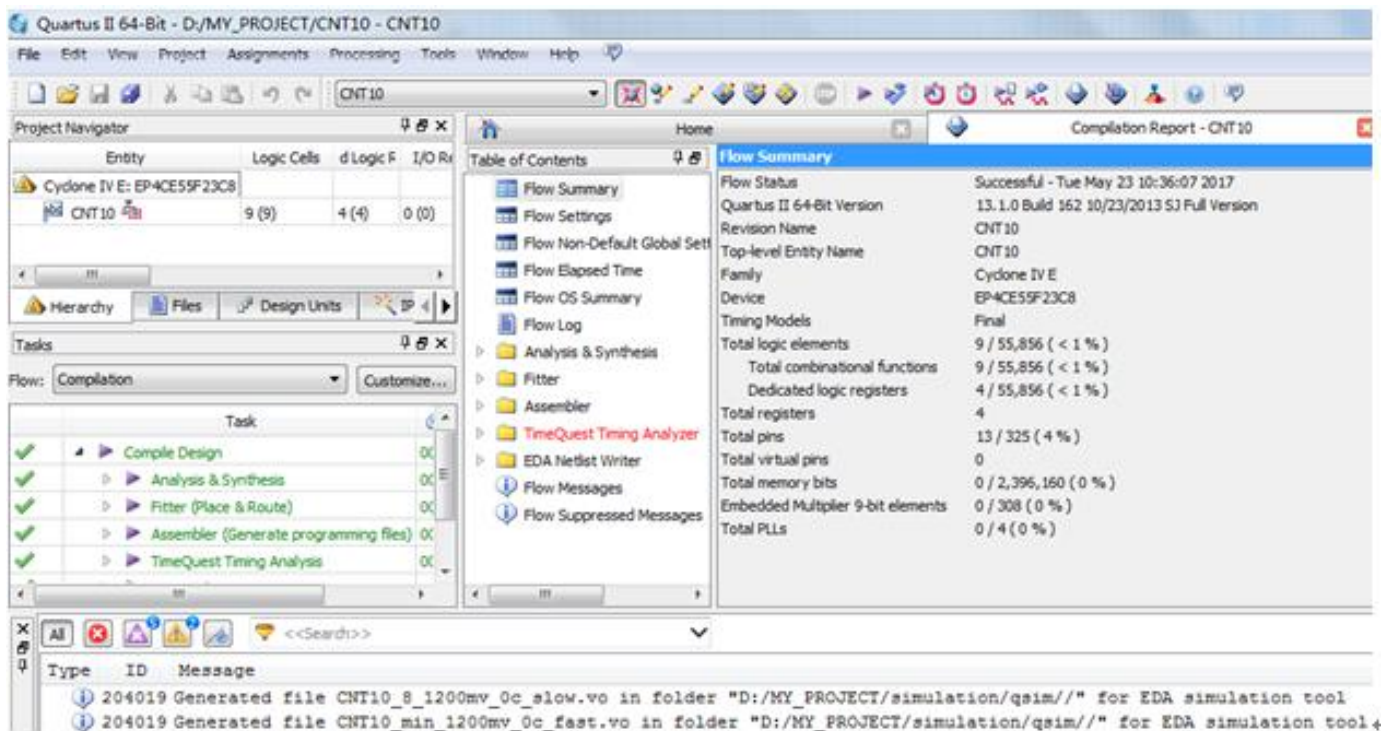


图 4-8 全程编译无错后的报告信息

4.1 代码编辑输入和系统编译

4.1.5 RTL图观察器应用

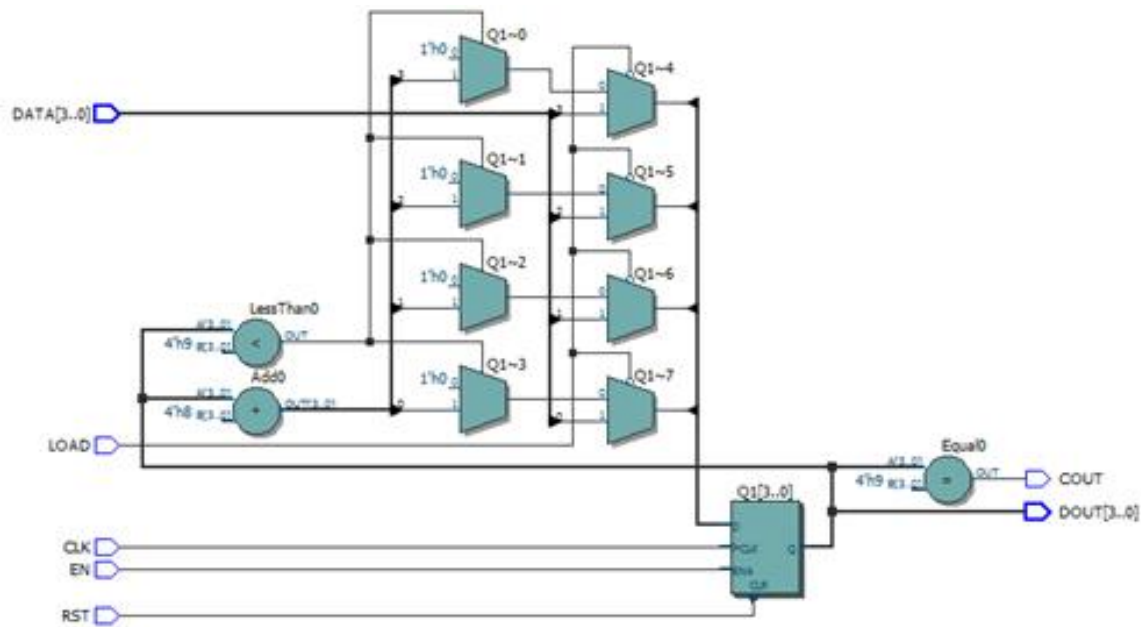


图 4-9 CNT10 工程的 RTL 图

4.2 时序仿真

(1) 确认Quartus II中的仿真工具是否指向Modelsim所在路径。

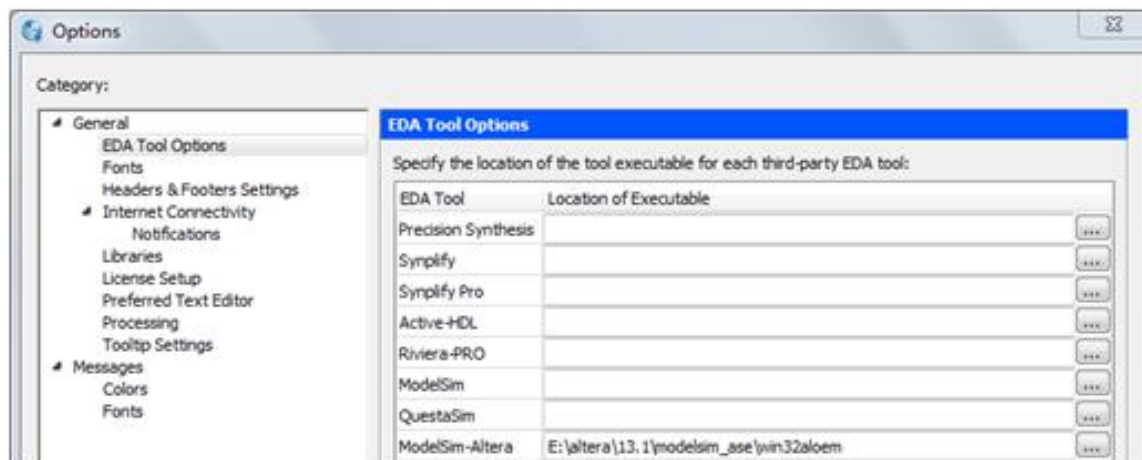


图 4-10 查看 Quartus 仿真工具指向 Modelsim 仿真软件的路径

4.2 时序仿真

(2) 打开波形编辑器。

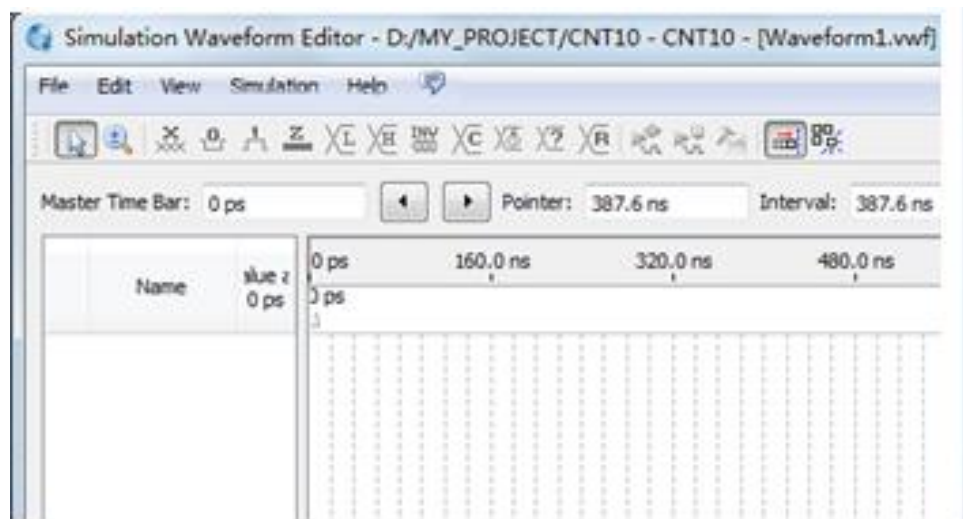


图 4-11 Vector Waveform File 文件编辑窗

4.2 时序仿真

(3) 设置仿真时间区域

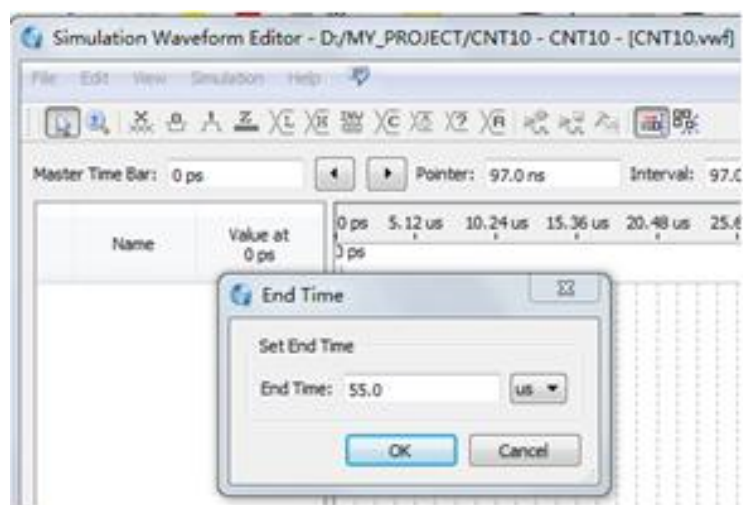


图 4-12 设置仿真时间长度

(4) 波形文件存盘。

4.2 时序仿真

(5) 将工程CNT10的端口信号节点选入波形编辑器中。

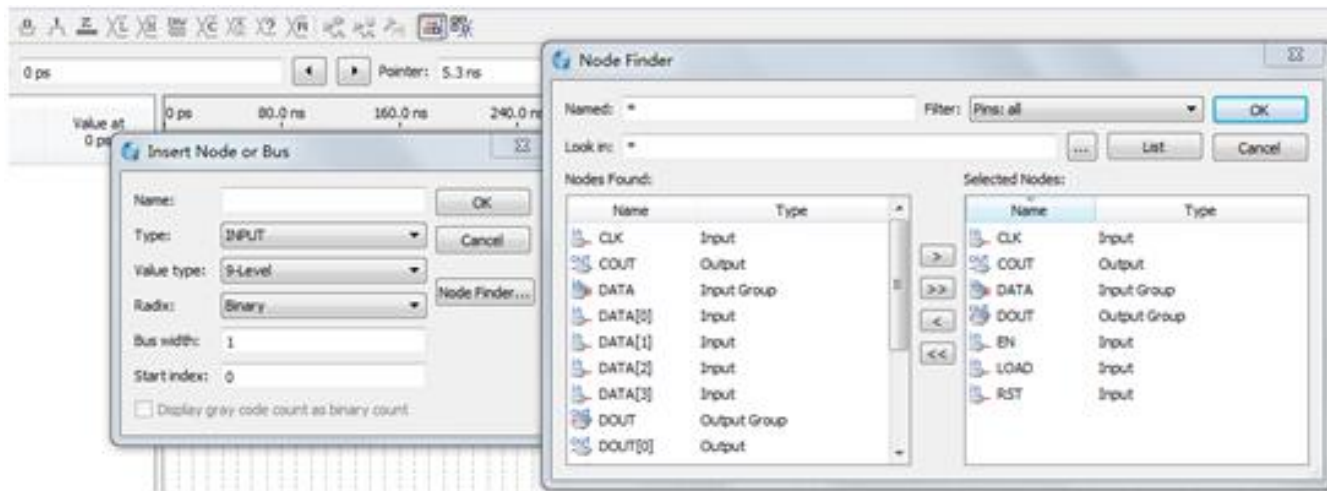


图 4-13 加入仿真需要的信号节点

4.2 时序仿真

(6) 设置激励信号波形。



图 4-14 设置总线数据格式

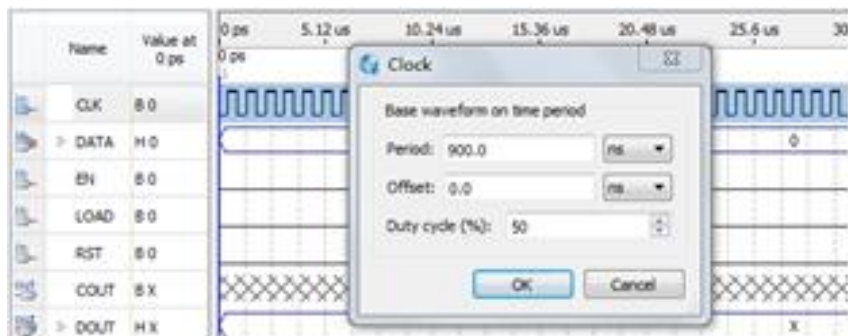


图 4-15 设置时钟参数

4.2 时序仿真

(7) 图4-16是最后设置好的.vwf仿真激励波形文件图。

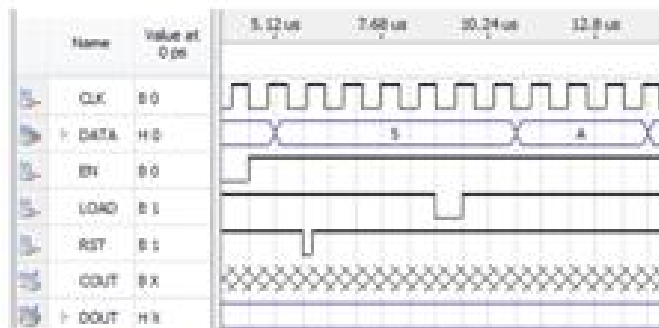


图 4-16 编辑好激励波形

(8) 启动仿真器。

4.2 时序仿真

(9) 观察仿真结果。

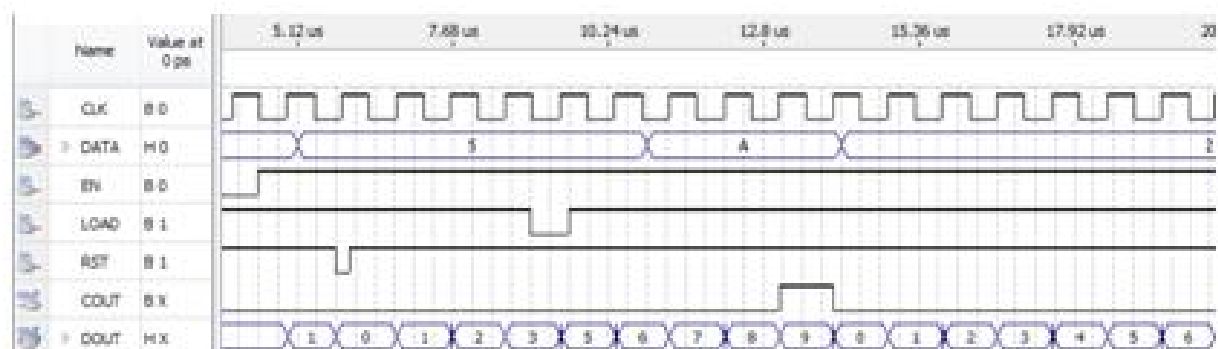


图 4-17 仿真输出的波形文件

4.3 硬件测试

4.3.1 引脚锁定

表 4-1 基于 EP4CE55F23C8 的引脚锁定情况（可通过附录 A.4 的列表获得）

计数器信号名	CLK	EN	LOAD	RST	DATA(3)	DATA(2)	DATA(1)
模式 0 电路控制	键 8	键 7	键 6	键 5	键 1:D4	键 1:D3	键 1:D2
模式 0 电路信号	PIO7	PIO6	PIO5	PIO4	PIO11	PIO10	PIO9
对应 FPGA 引脚	AB6	Y7	AA6	AB3	AB5	AA3	W2
计数器信号名	DATA(0)		COUT	DOUT(3)	DOUT(2)	DOUT(1)	DOUT(0)
模式 0 电路控制	键 1:D1		数码 2:a 段	数码 1	数码 1	数码 1	数码 1
模式 0 电路信号	PIO8		PIO20	PIO19	PIO18	PIO17	PIO16
对应 FPGA 引脚	U2		AA1	V2	W1	R2	U1

4.3 硬件测试

4.3.1 引脚锁定

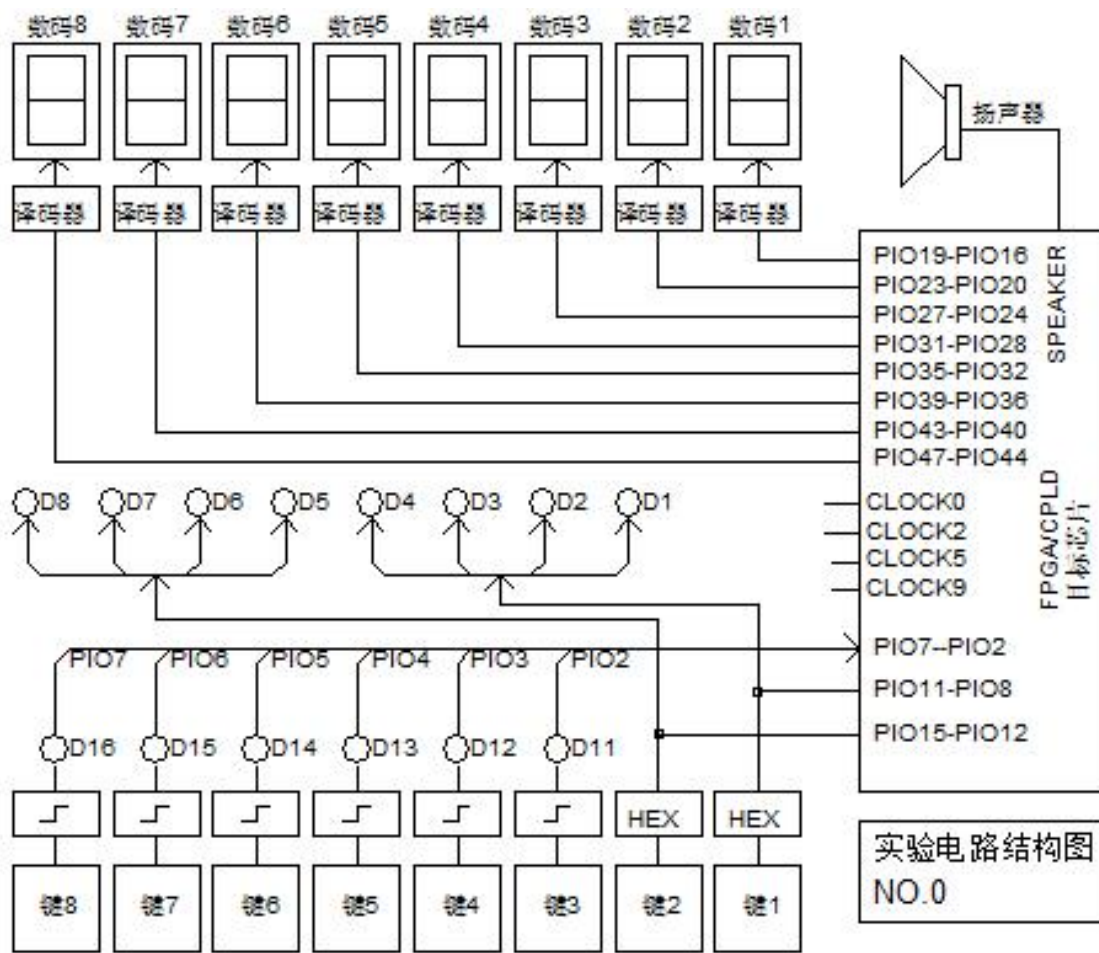


图 4-18 模式 0 的对 FPGA 的实验电路

4.3 硬件测试

4.3.1 引脚锁定

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location
CLK	Input				PIN_J4
COUT	Output				PIN_T8
DATA[3]	Input				PIN_R8
DATA[2]	Input				PIN_R10
DATA[1]	Input				PIN_T9
DATA[0]	Input				PIN_V6
DOUT[3]	Output				PIN_R9
DOUT[2]	Output				PIN_T5
DOUT[1]	Output				PIN_R6
DOUT[0]	Output				PIN_P8
EN	Input				PIN_T7
LOAD	Input				PIN_R7
RST	Input				PIN_P4

图 4-19 编译完成后刚打开的 Pin Planner 窗

Node Name	Direction	Location
in CLK	Input	PIN_AB6
out COUT	Output	PIN_AA1
in DATA[3]	Input	PIN_AB5
in DATA[2]	Input	PIN_AA3
in DATA[1]	Input	PIN_W2
in DATA[0]	Input	PIN_U2
out DOUT[3]	Output	PIN_V2
out DOUT[2]	Output	PIN_W1
out DOUT[1]	Output	PIN_R2
out DOUT[0]	Output	PIN_U1
in EN	Input	PIN_Y7
in LOAD	Input	PIN_AA6
in RST	Input	PIN_AB3

图 4-20 引脚锁定完成后的情况

4.3 硬件测试

4.3.2 编译文件下载

(1) 打开编程窗和配置文件。

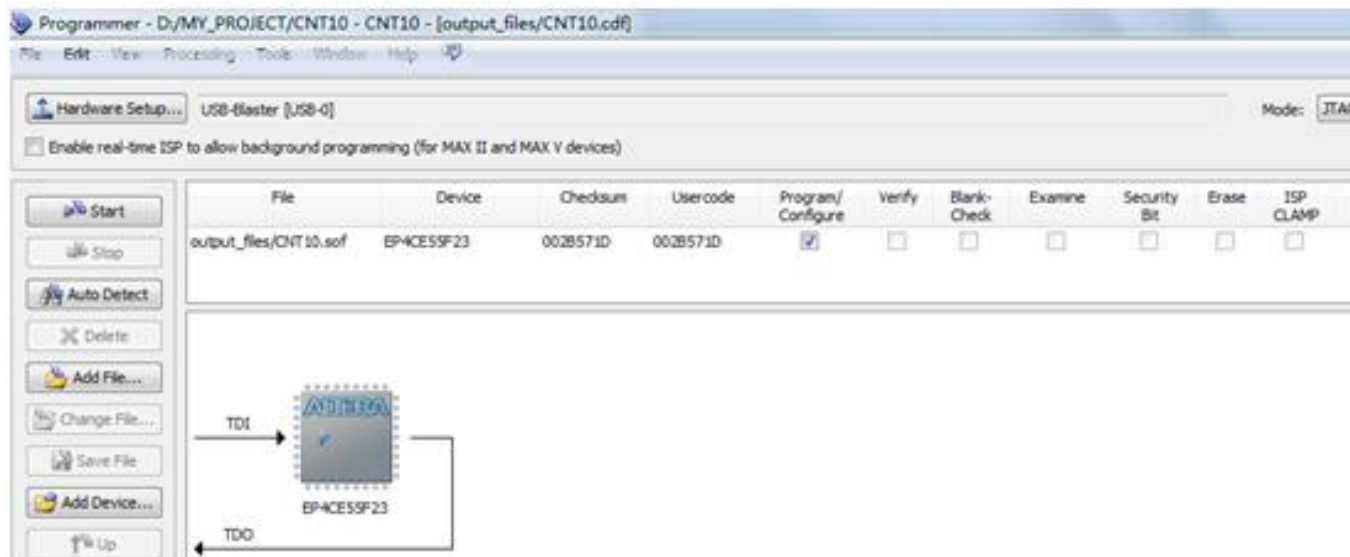


图 4-21 选择 JTAG 编程模式，将 SOF 文件载入 FPGA

4.3 硬件测试

4.3.2 编译文件下载

(2) 设置编程器。

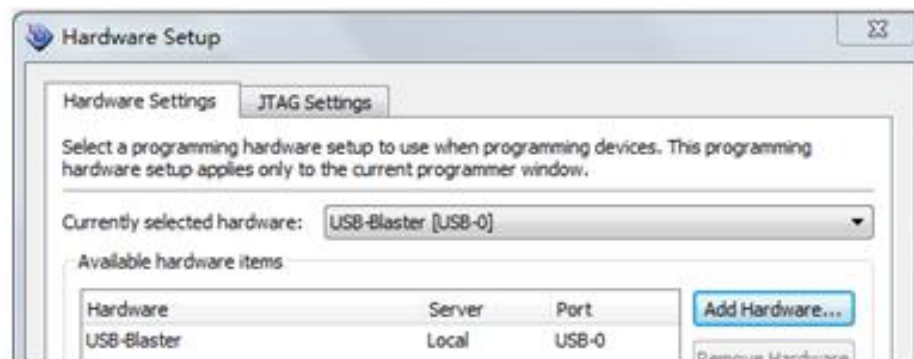


图 4-22 加入编程下载方式

(3) 硬件测试。

4.3 硬件测试

4.3.3 通过JTAG口对配置芯片进行间接编程

1. 将SOF文件转化为JTAG间接配置文件

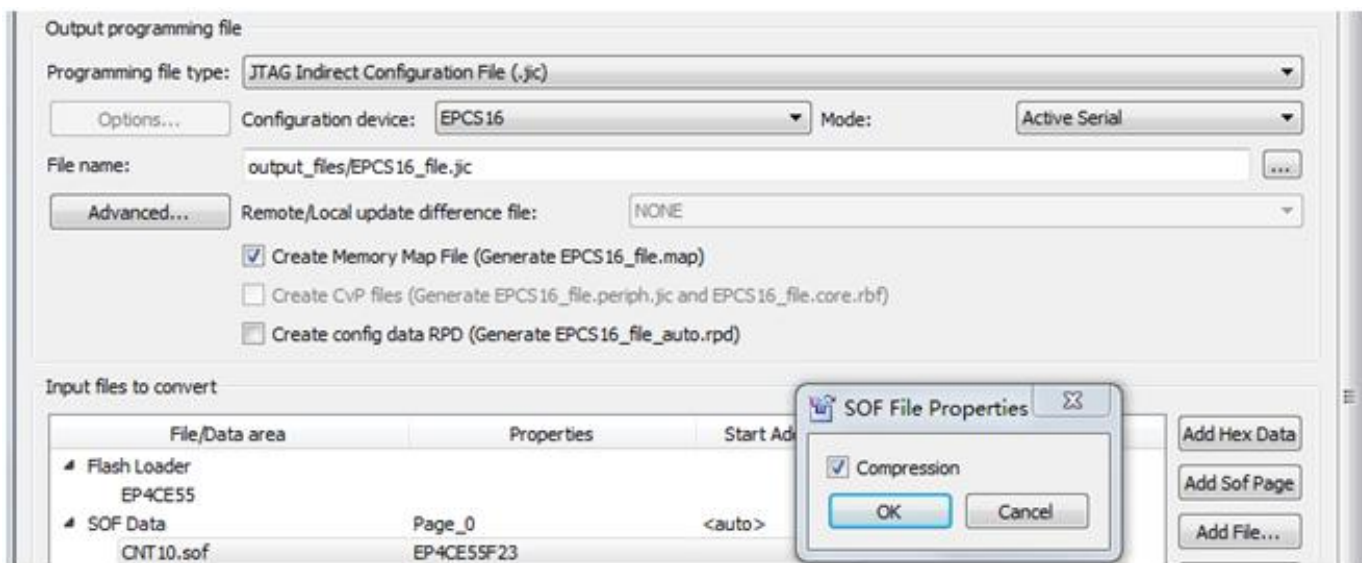


图 4-23 设定 JTAG 间接编程文件

4.3 硬件测试

4.3.3 通过JTAG口对配置芯片进行间接编程

2. 下载JTAG间接配置文件

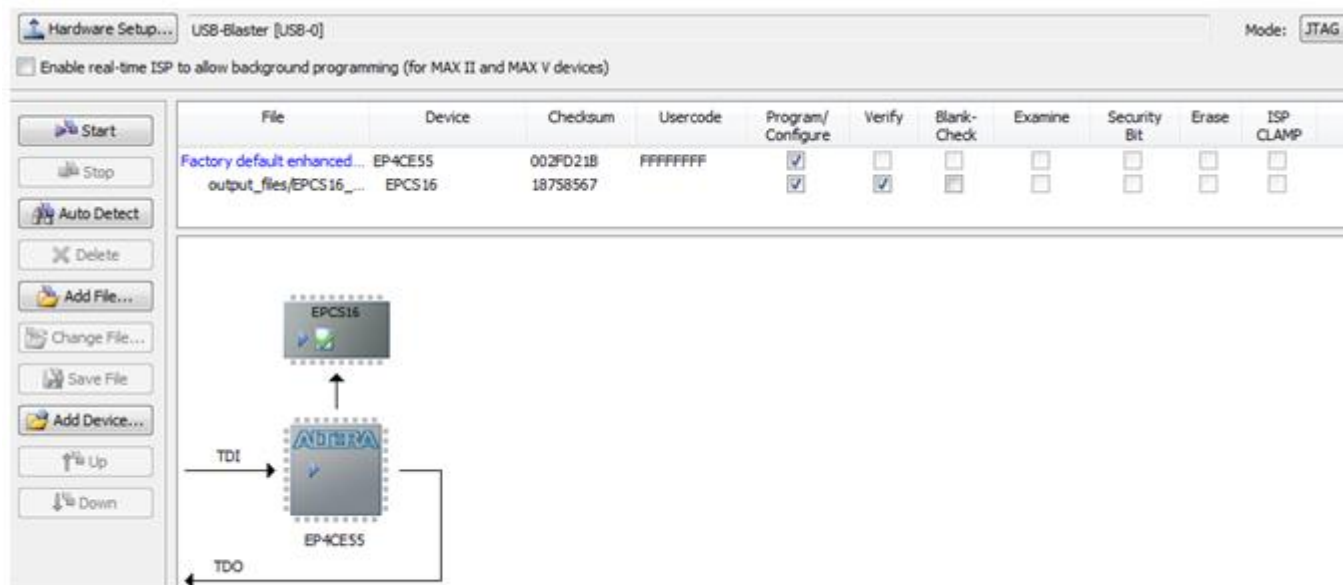


图 4-24 用 JTAG 模式将间接配置文件烧入配置器件 EPCS16 中

4.3 硬件测试

4.3.4 USB-Blaster驱动程序安装方法

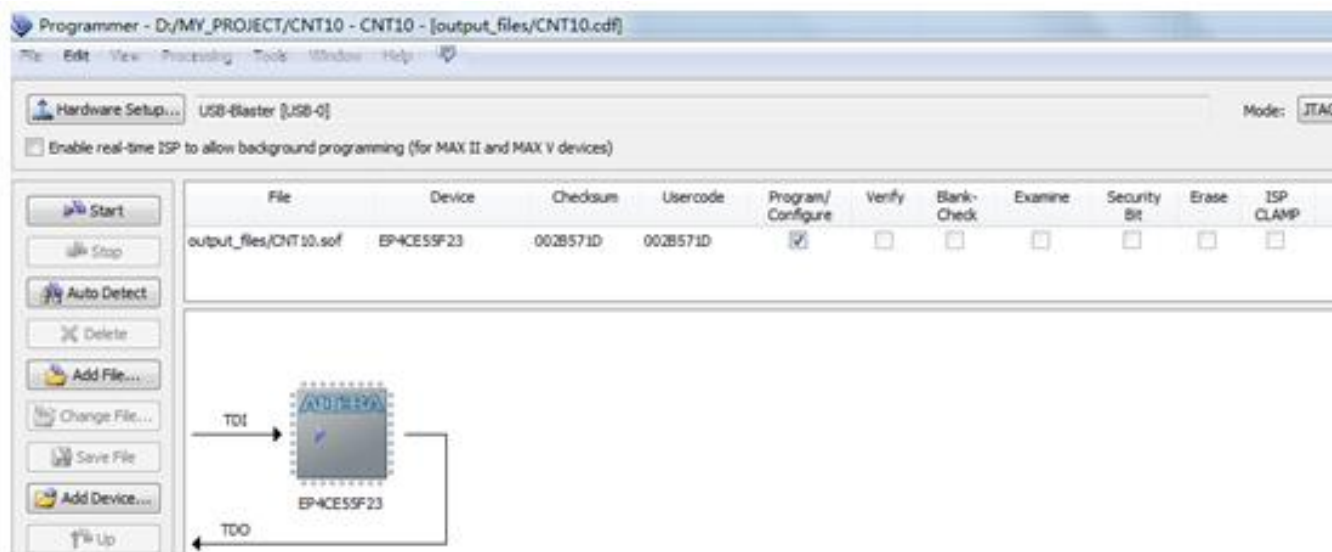


图 4-21 选择 JTAG 编程模式，将 SOF 文件载入 FPGA

4.4 电路原理图设计流程

4.4.1 设计一个半加器

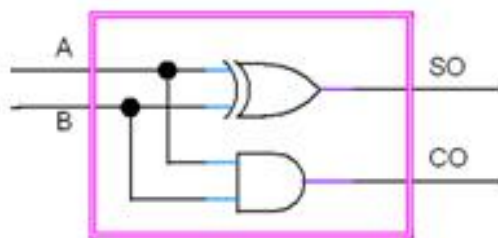


图 4-25 半加器的电路结构

A	B	SO	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

图 4-26 半加器的真值表

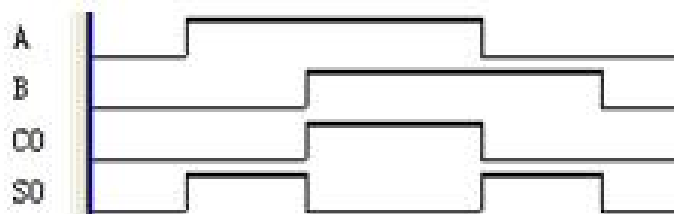


图 4-27 半加器的仿真功能波形图

4.4 电路原理图设计流程

4.4.1 设计一个半加器

- (1) 打开原理图编辑窗。
- (2) 建立一个初始原理图文件。



图 4-28 选择打开元件输入窗

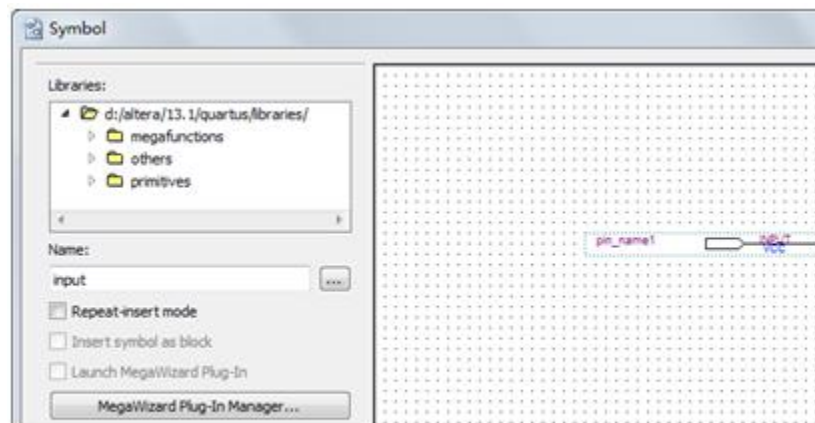


图 4-29 在元件输入对话框输入引脚

4.4 电路原理图设计流程

4.4.1 设计一个半加器

(3) 原理图文件存盘。

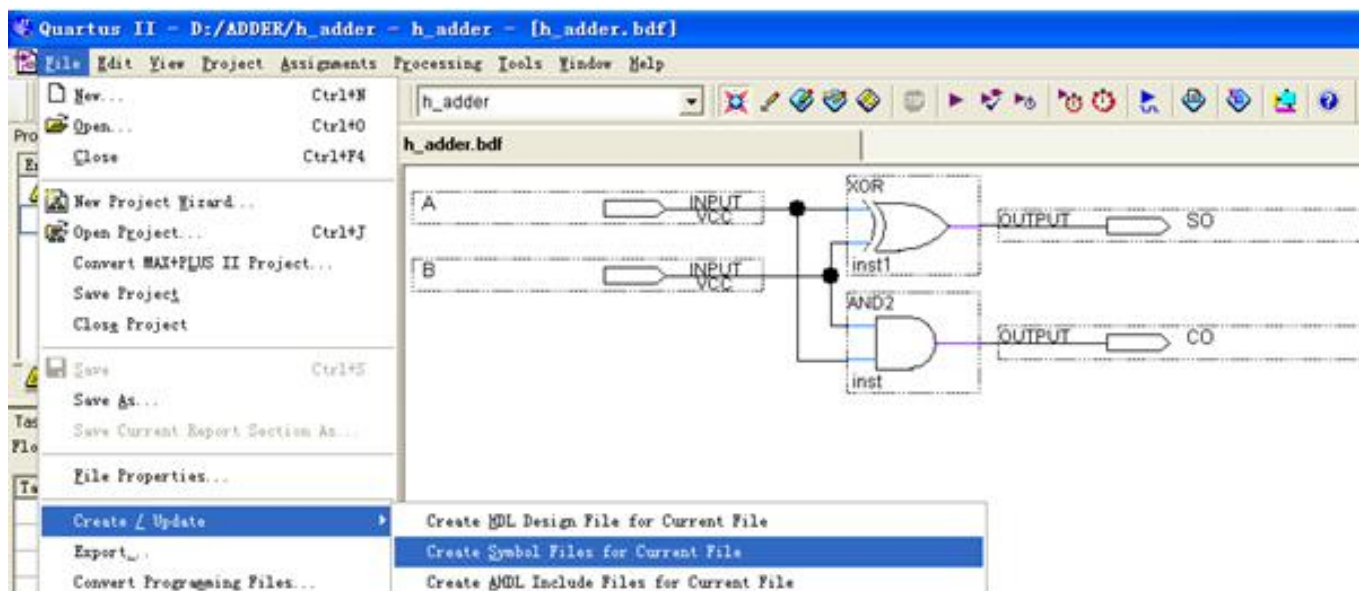


图 4-30 完成设计并将半加器封装成一个元件，以便在更高层设计中调用

4.4 电路原理图设计流程

4.4.1 设计一个半加器

(4) 创建原理图文件为顶层设计的工程。

(5) 绘制半加器原理图。

(6) 仿真测试半加器。

4.4 电路原理图设计流程

4.4.2 完成全加器顶层设计

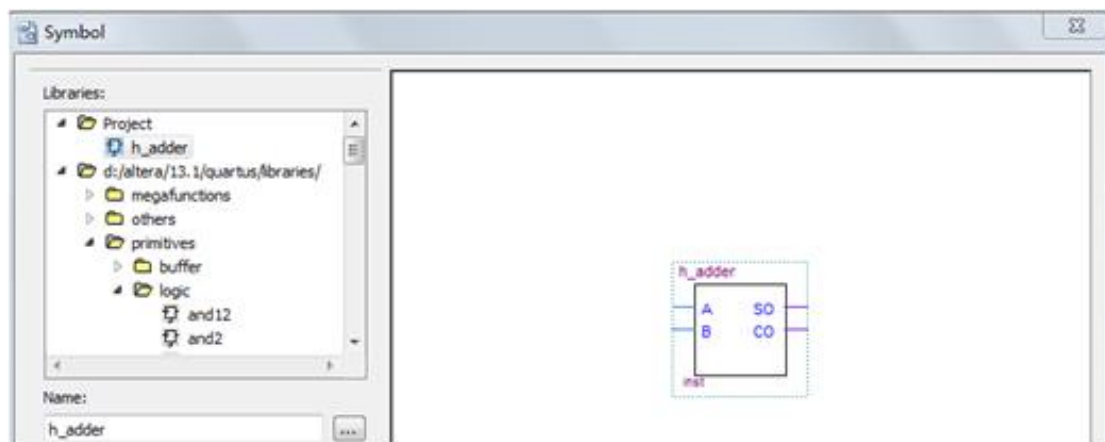


图 4-31 在 f_adder 工程下加入半加器原件

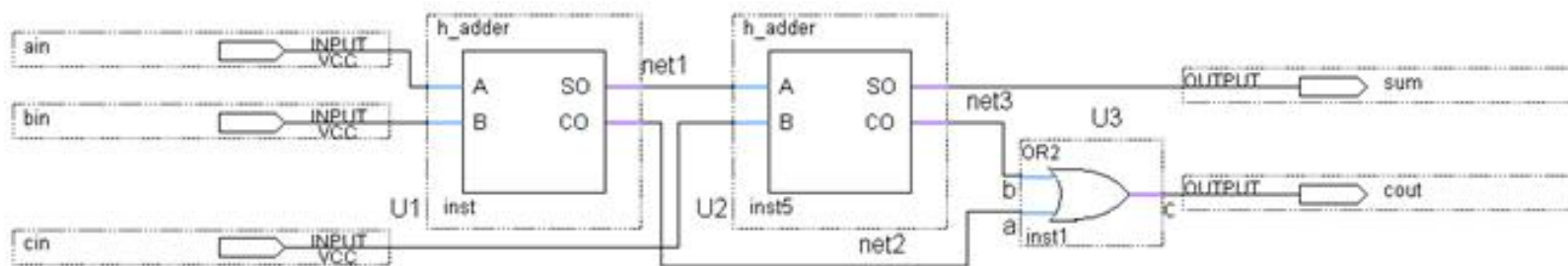


图 4-32 全加器 f_adder 电路图

4.4 电路原理图设计流程

4.4.3 对全加器进行时序仿真和硬件测试

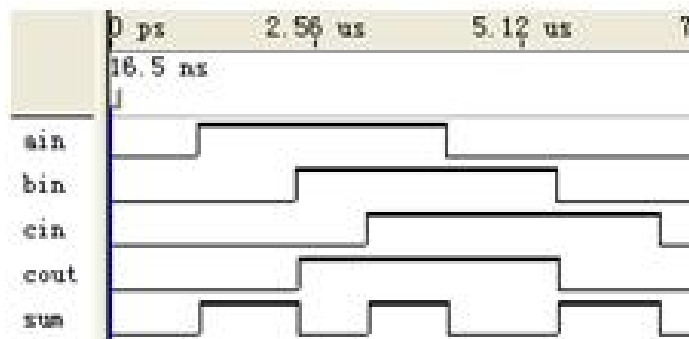


图 4-33 全加器的仿真波形

4.4 电路原理图设计流程

4.4.3 对全加器进行时序仿真和硬件测试

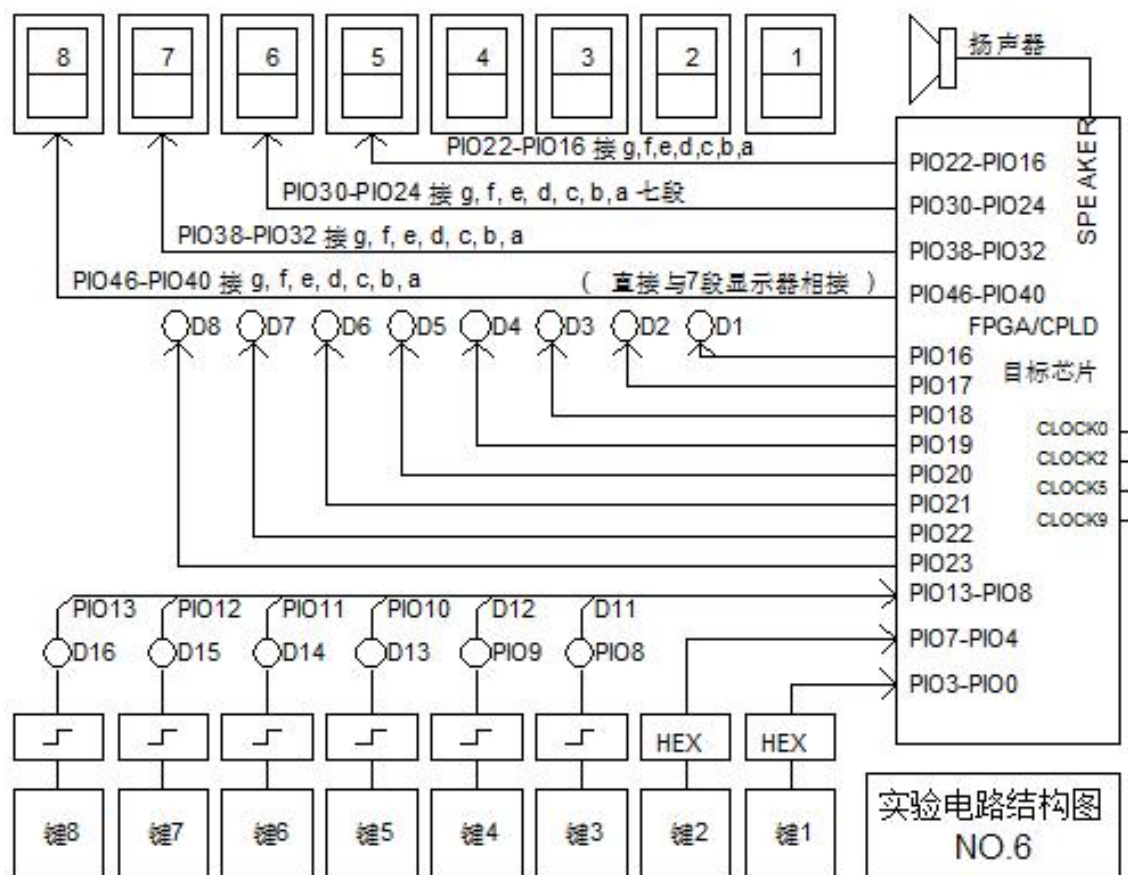


图 4-34 对应模式 6 的对 FPGA 的实验电路

4.5 利用属性表述实现引脚锁定

【例 4-1】

```
ARCHITECTURE ONE OF CNT10 IS
  attribute chip_pin : string; -- chip_pin被定义为字符串数据类型string
  attribute chip_pin of CLK : signal is "AB6";
  attribute chip_pin of EN  : signal is "Y7";
  attribute chip_pin of DATA : signal is "AB5,AA3,W2,U2";
  ...
BEGIN
```

Edit -> Insert Template -> VHDL -> Synthesis Attributes;

4.6 SignalTap II的用法

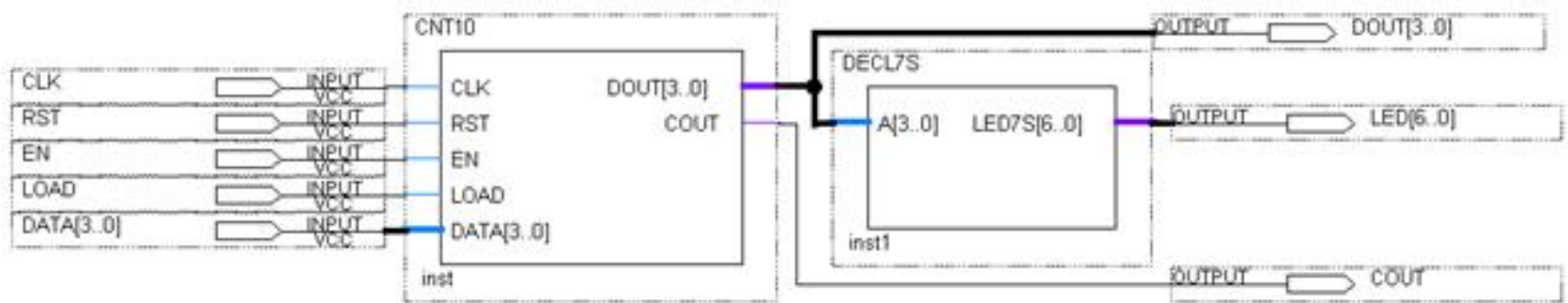


图 4-35 十进制计数器设计示例电路

4.6 SignalTap II 的用法

1. 打开SignalTap II编辑窗口

2. 调入待测信号

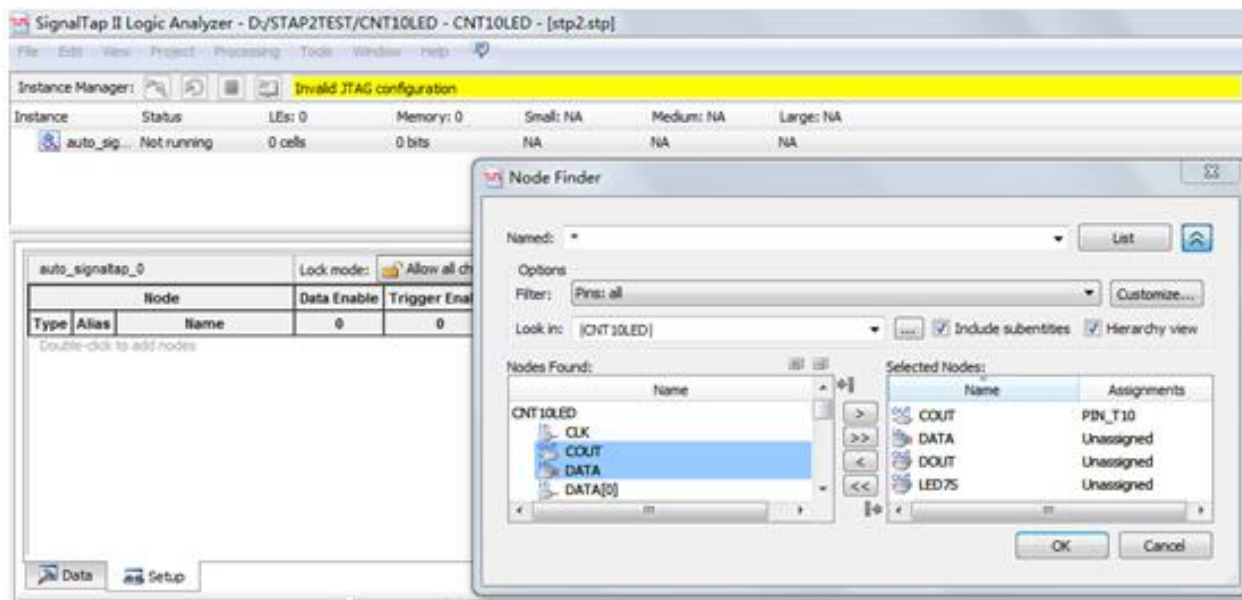


图 4-36 输入逻辑分析仪测试信号

4.6 SignalTap II 的用法

3. SignalTap II 参数设置

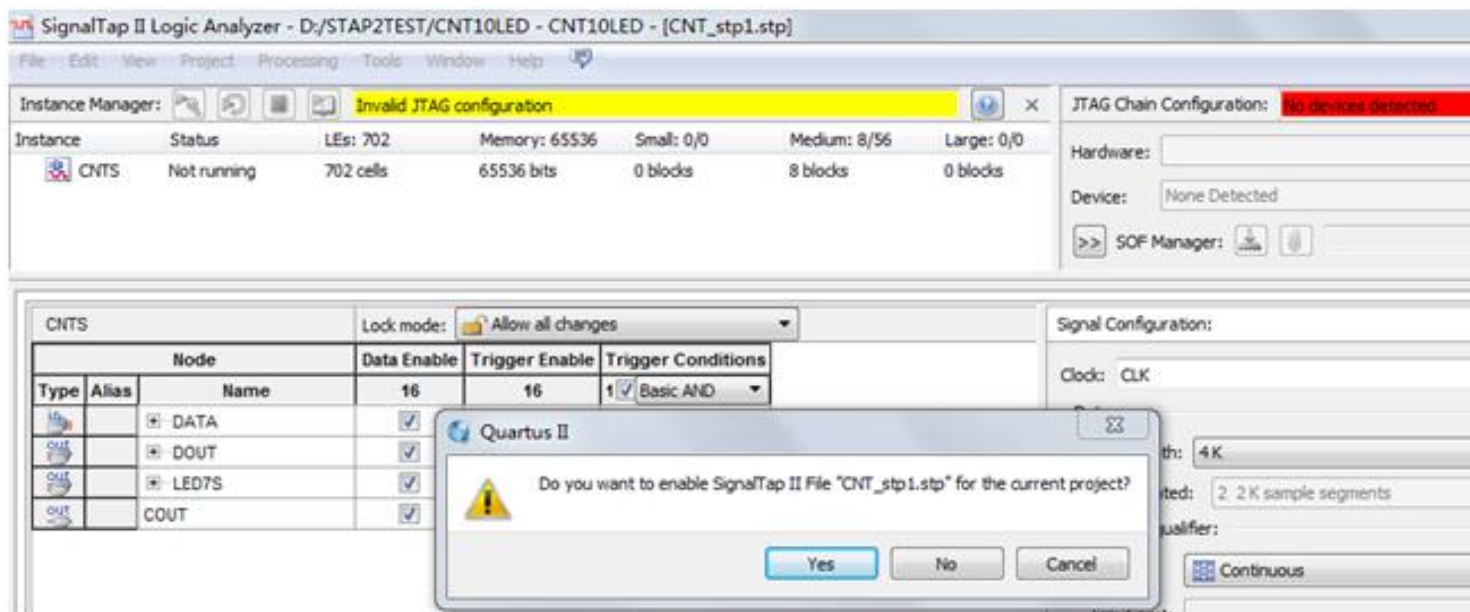


图 4-37 SignalTap II 编辑窗口

4.6 SignalTap II 的用法

3. SignalTap II 参数设置

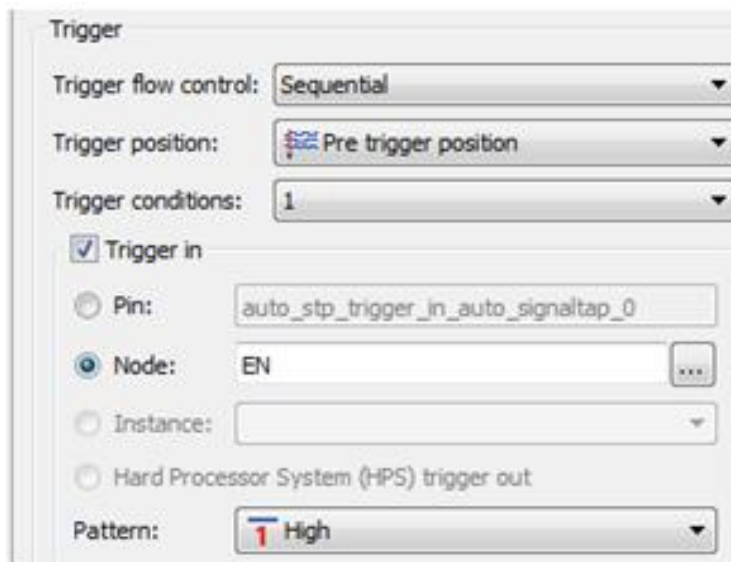


图 4-38 设置 EN 为触发信号

4.6 SignalTap II 的用法

4. 文件存盘

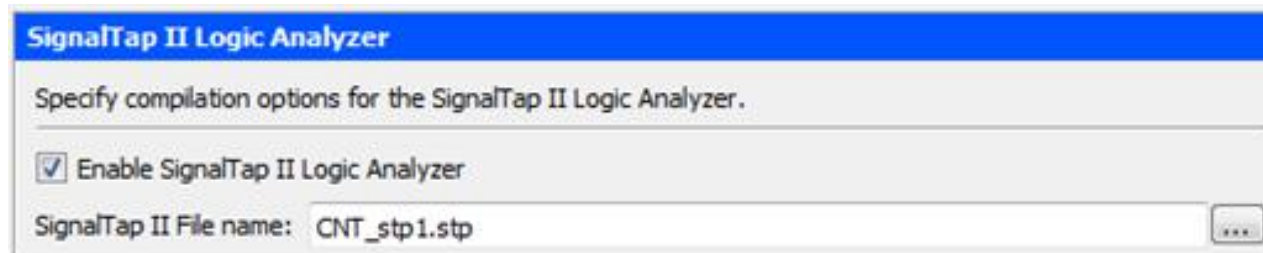


图 4-39 选择或删除 SignalTap II 文件加入综合编译

4.6 SignalTap II 的用法

5. 编译下载

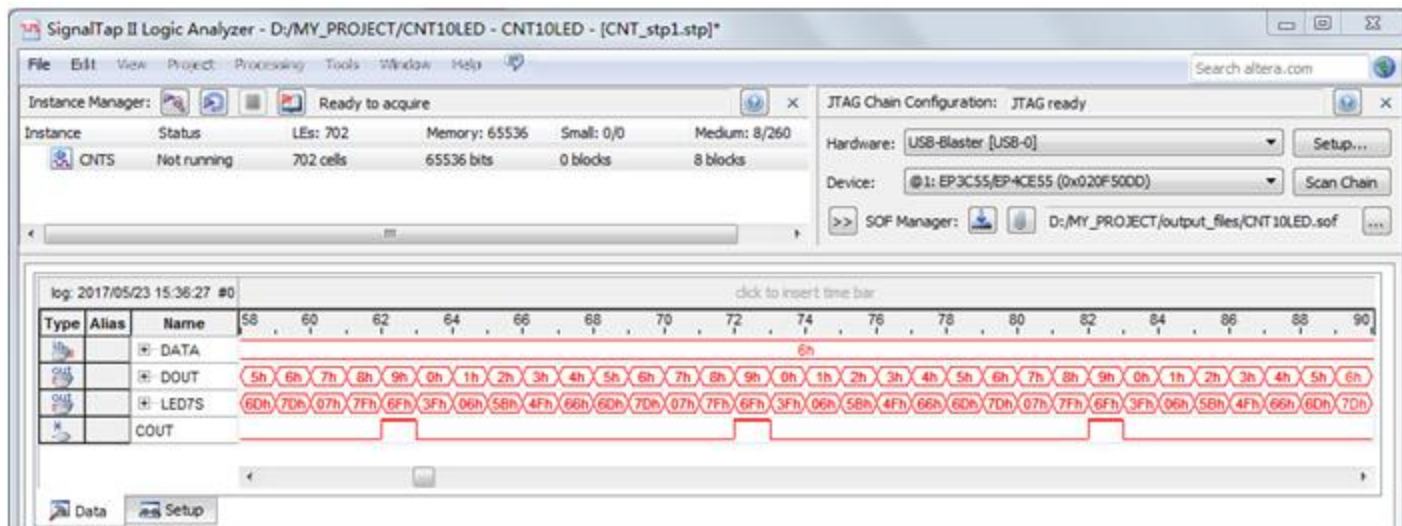


图 4-40 SignalTap II 实时数据采样显示界面

4.6 SignalTap II 的用法

6. 启动 SignalTap II 进行采样与分析

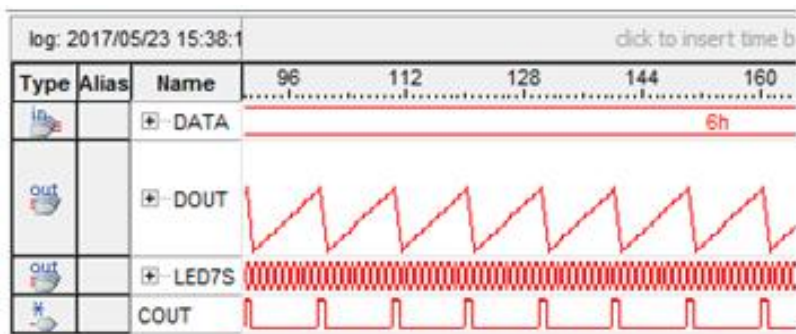


图 4-41 改变 DOUT 数据显示的方式

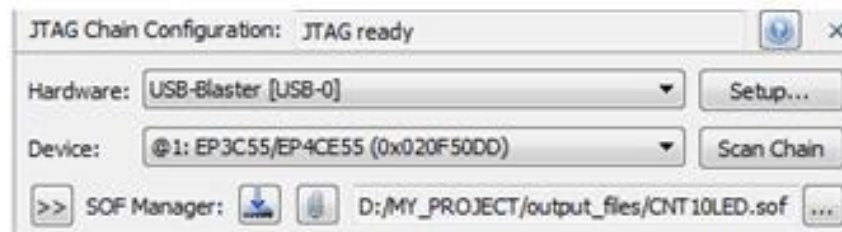


图 4-42 扫描 FPGA，并下载 SOF 文件

7. SignalTap II 的其他设置和控制方法

4.7 编辑SignalTap II的触发信号

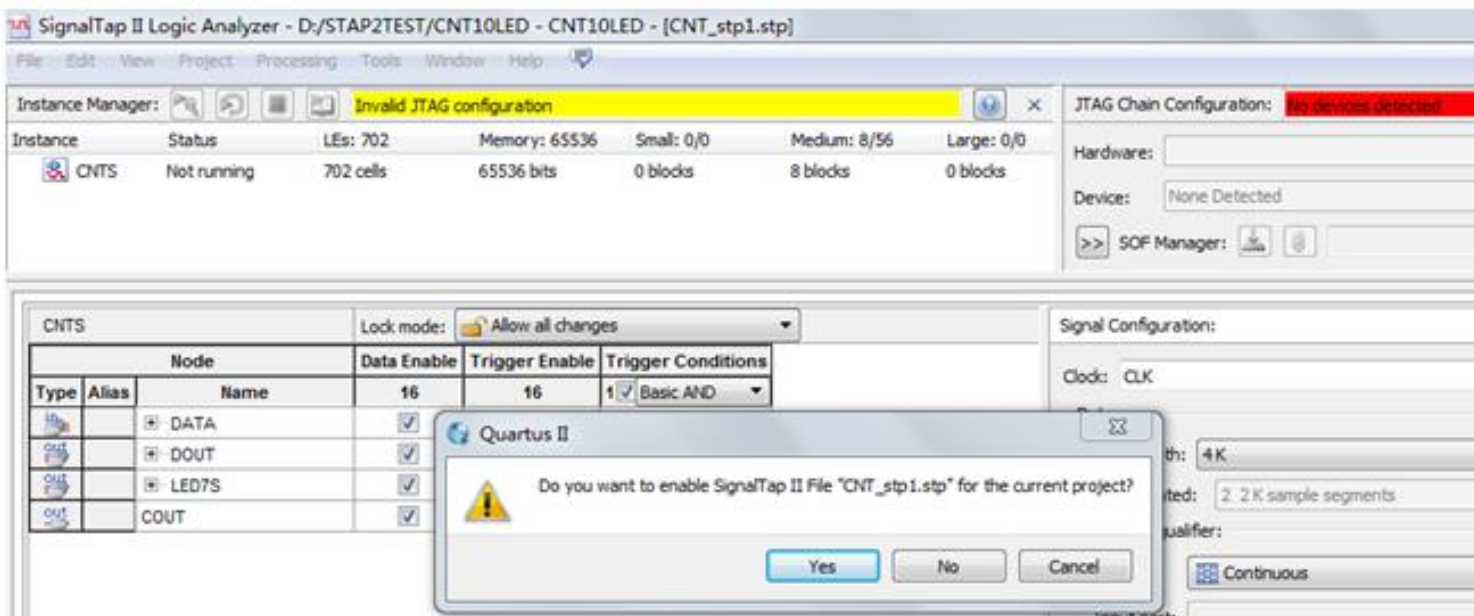


图 4-37 SignalTap II 编辑窗口

4.8 安装Quartus II 13.1说明

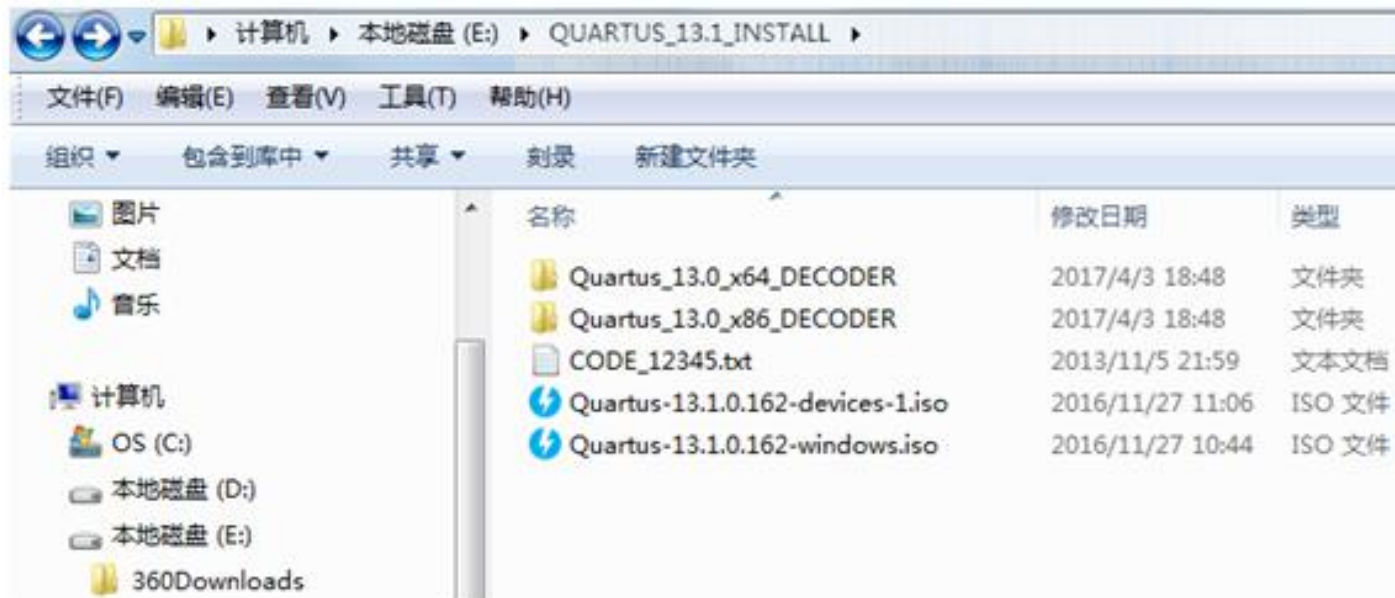


图 4-43 QuartusII 13.1 的 4 个安装文件

4.8 安装Quartus II 13.1说明



图 4-44 首先安装 QuartusII 13.1 设计文件：Quartus-13.1.0.162-devices-1

4.8 安装Quartus II 13.1说明

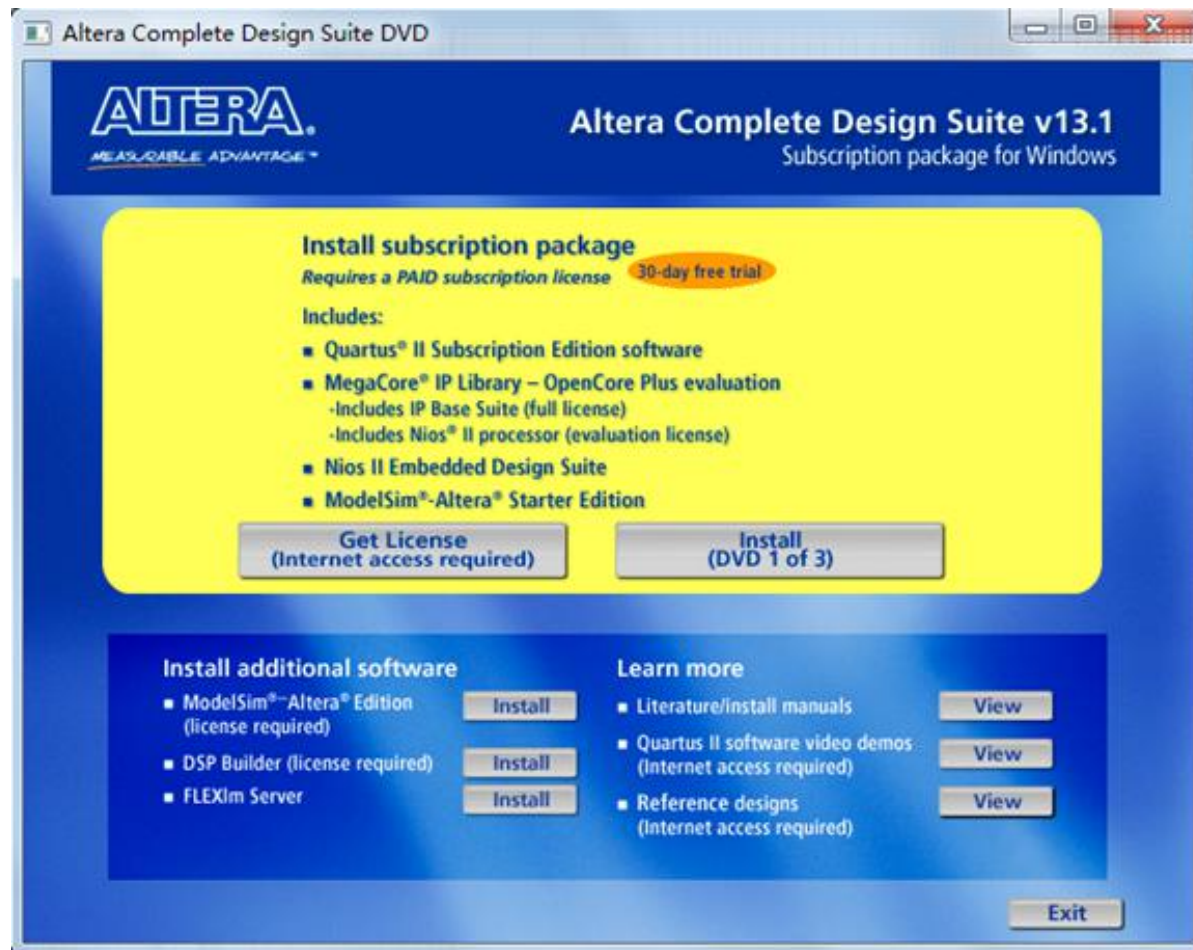


图 4-45 安装 QuartusII 13.1 设计文件界面，点击右侧安装按钮

4.8 安装Quartus II 13.1说明

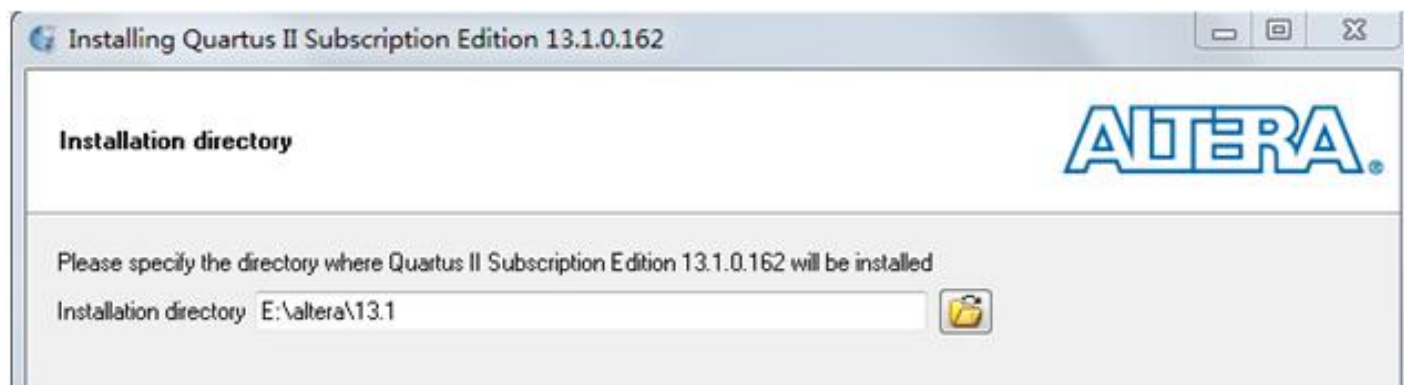


图 4-46 设定 QuartusII 13.1 设计文件安装路径

4.8 安装Quartus II 13.1说明

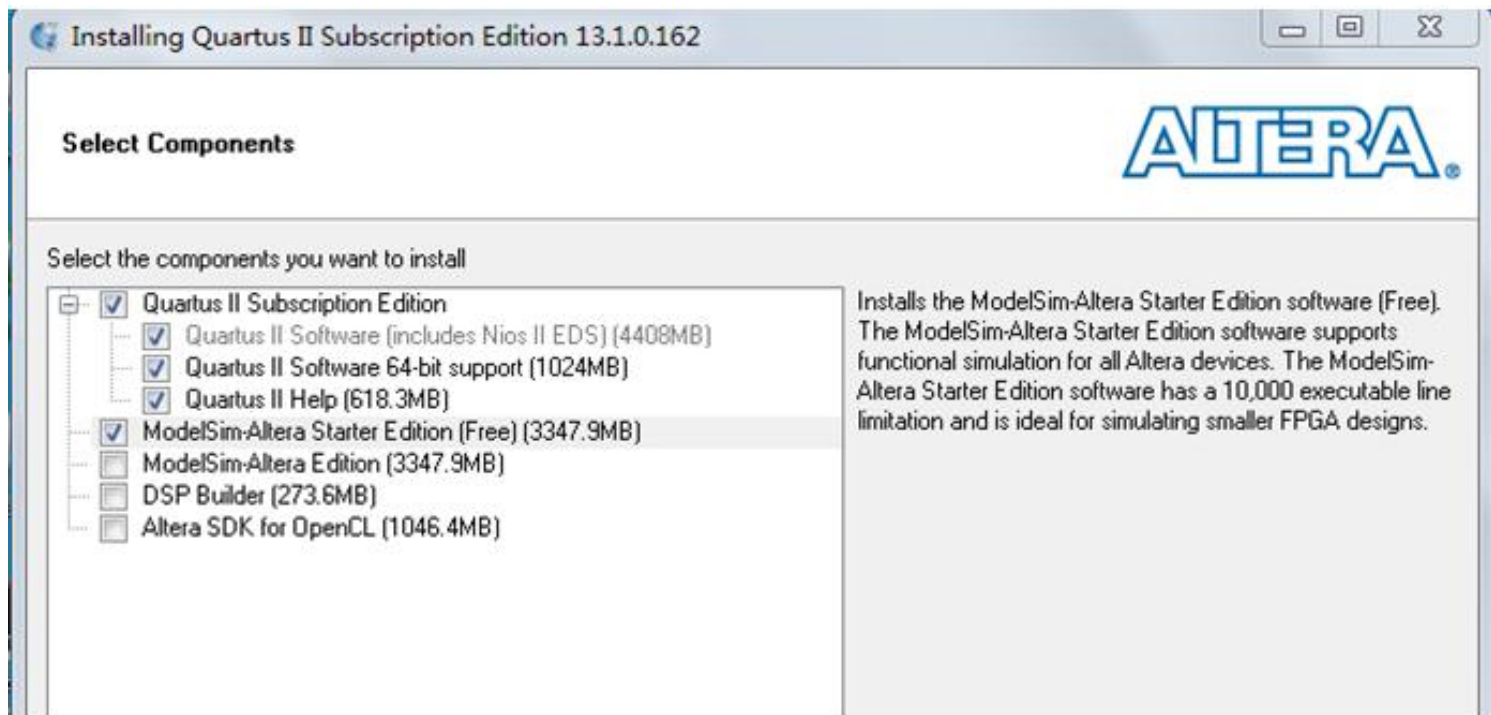


图 4-47 选择安装软件。注意不要漏了 ModelSim-Altera Starter Edition

4.8 安装Quartus II 13.1说明

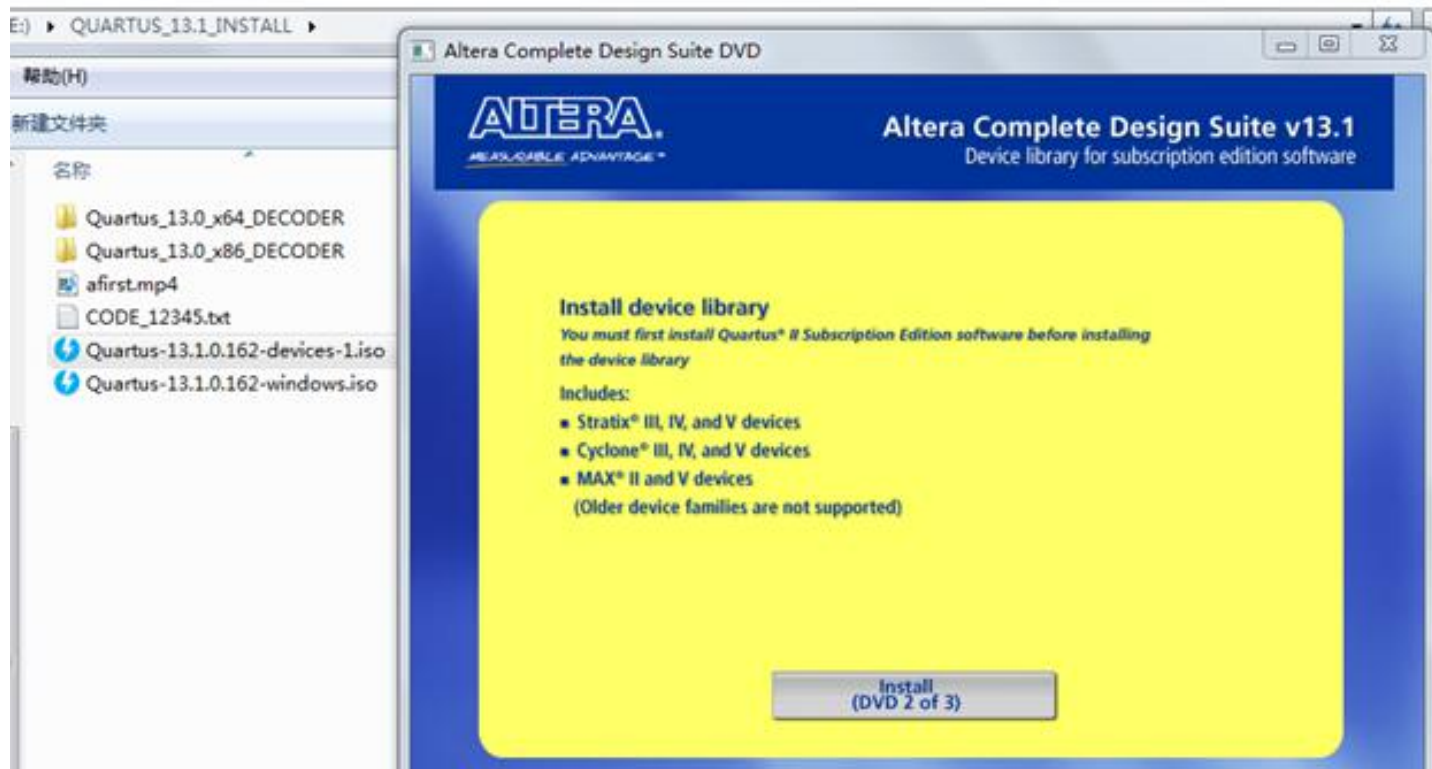


图 4-48 安装完 QuartusII 13.1 设计文件后，再安装器件文件

4.8 安装Quartus II 13.1说明

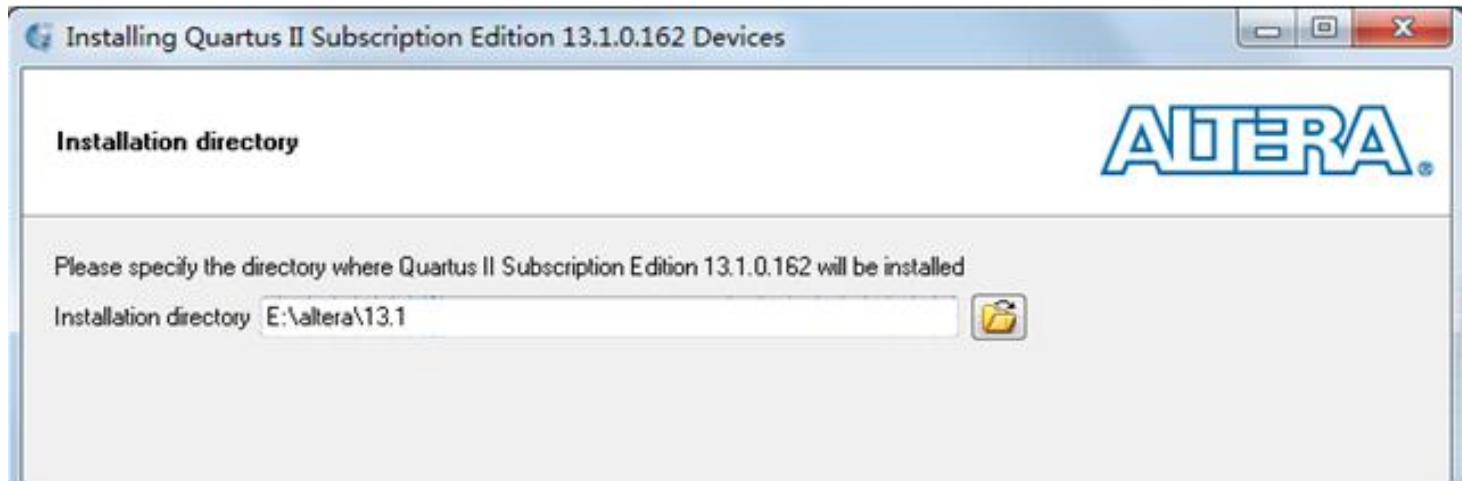


图 4-49 设定器件文件安装的路径

4.8 安装Quartus II 13.1说明

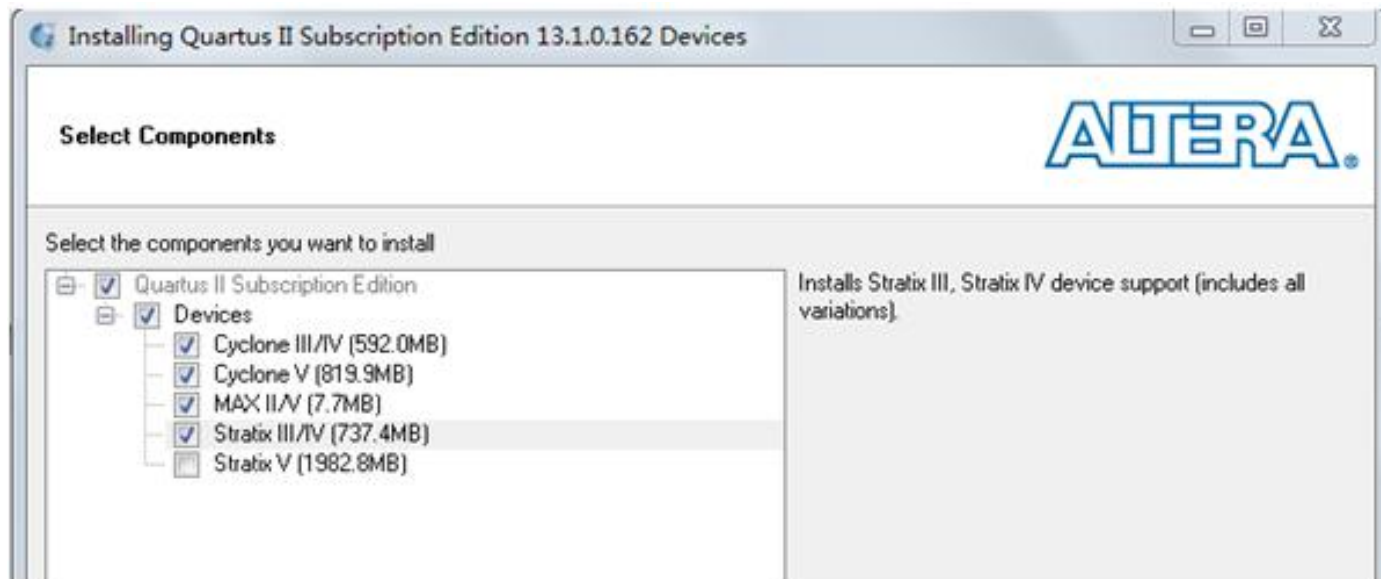


图 4-50 选择需要安装的器件系列

4.8 安装Quartus II 13.1说明

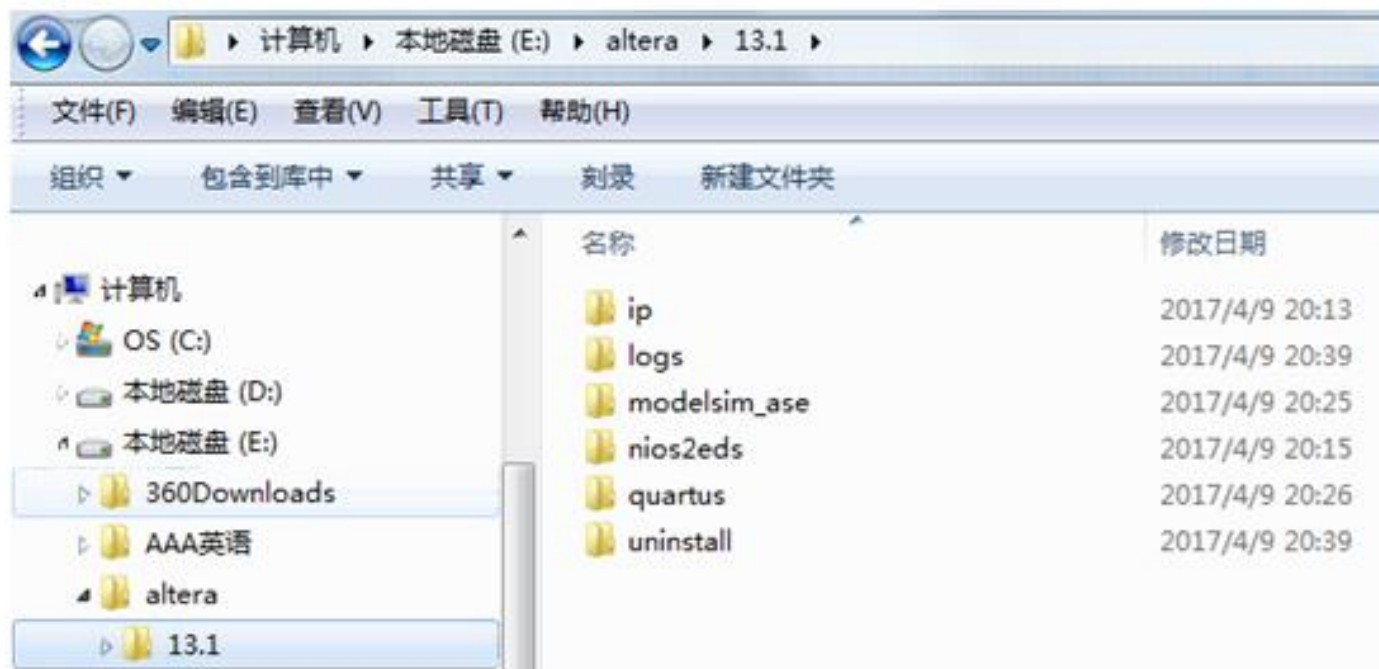


图 4-51 QuartusII 13.1 安装完成后查看一下是否已经安装上了 ModelSim-Altera Starter Edition

4.8 安装Quartus II 13.1说明

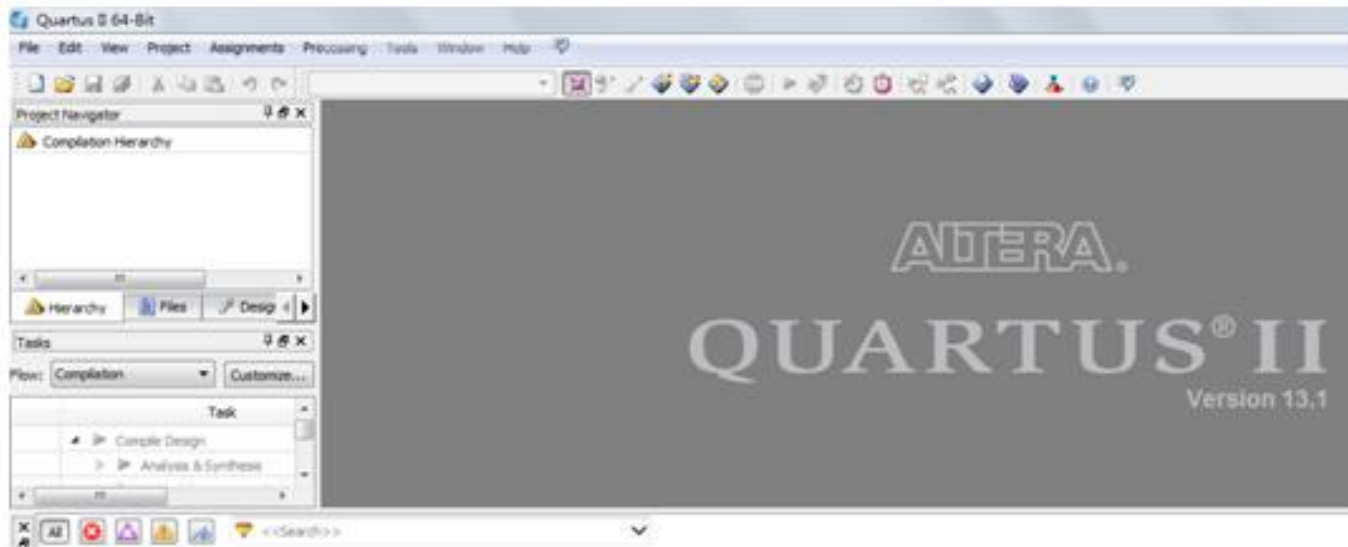


图 4-52 打开 QuartusII 13.1

4.8 安装Quartus II 13.1说明

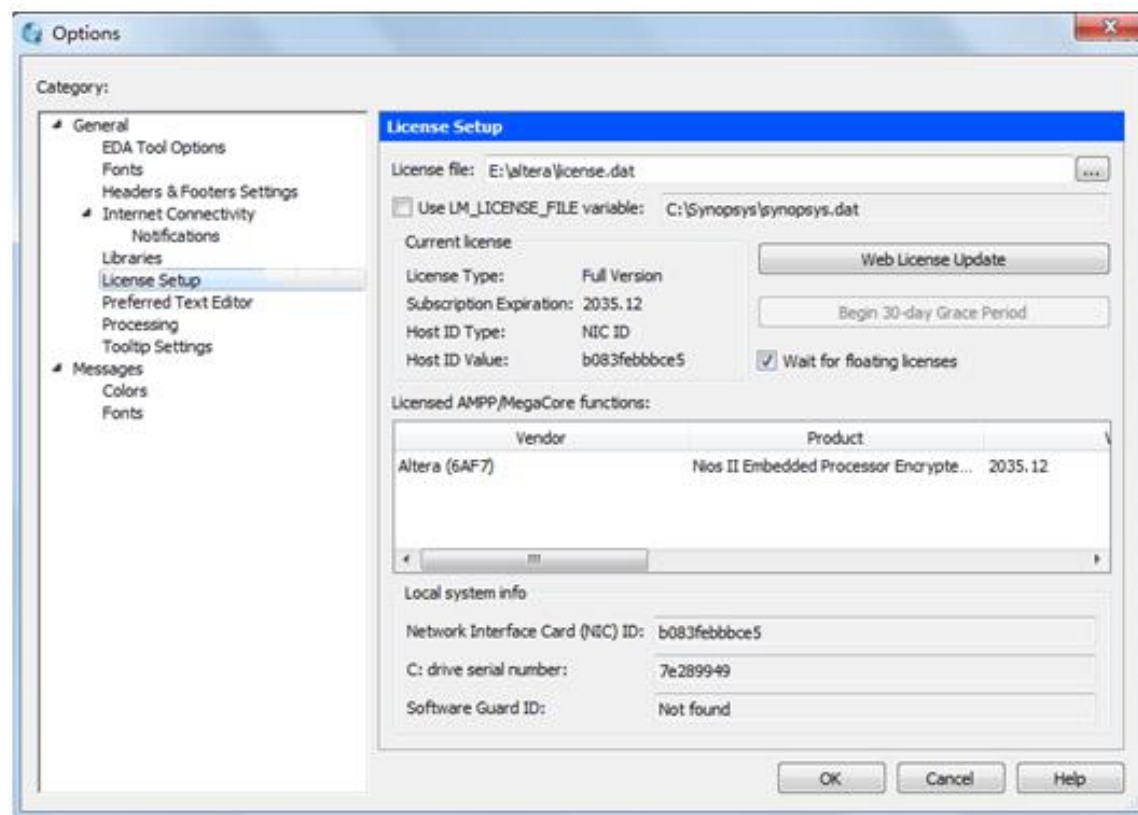


图 4-53 在此窗口获取网卡 ID 码，以及设定授权文件 License.dat

实验与设计

实验4-1 多路选择器设计实验

实验4-2 十六进制7段数码显示译码器设计

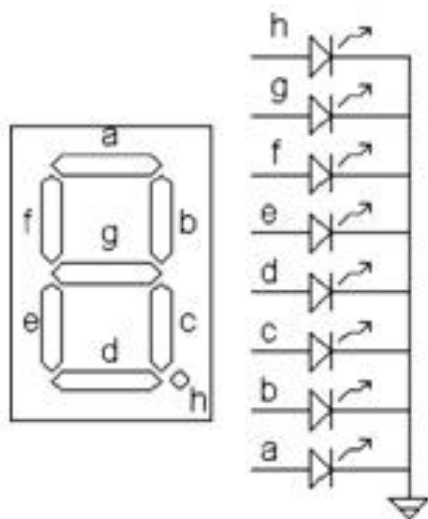


图 4-54 共阴数码管

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

【例 4-2】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY DECL7S IS
    PORT ( A : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
          LED7S : OUT STD_LOGIC_VECTOR(6 DOWNTO 0));
END;
ARCHITECTURE one OF DECL7S IS
    BEGIN
        PROCESS( A ) BEGIN
            CASE A IS
                WHEN "0000" => LED7S <= "0111111";
                WHEN "0001" => LED7S <= "0000110";
                WHEN "0010" => LED7S <= "1011011";
                WHEN "0011" => LED7S <= "1001111";
                WHEN "0100" => LED7S <= "1100110";
                WHEN "0101" => LED7S <= "1101101";
                WHEN "0110" => LED7S <= "1111101";
                WHEN "0111" => LED7S <= "0000111";
                WHEN "1000" => LED7S <= "1111111";
                WHEN "1001" => LED7S <= "1101111";
                WHEN "1010" => LED7S <= "1110111";
                WHEN "1011" => LED7S <= "1111100";
                WHEN "1100" => LED7S <= "0111001";
                WHEN "1101" => LED7S <= "1011110";
                WHEN "1110" => LED7S <= "1111001";
                WHEN "1111" => LED7S <= "1110001";
                WHEN OTHERS => NULL;
            END CASE;
        END PROCESS;
    END;
```

实验与设计

实验4-3 应用宏模块设计数字频率计

1. 2位十进制计数器设计

(1) 设计电路原理图。

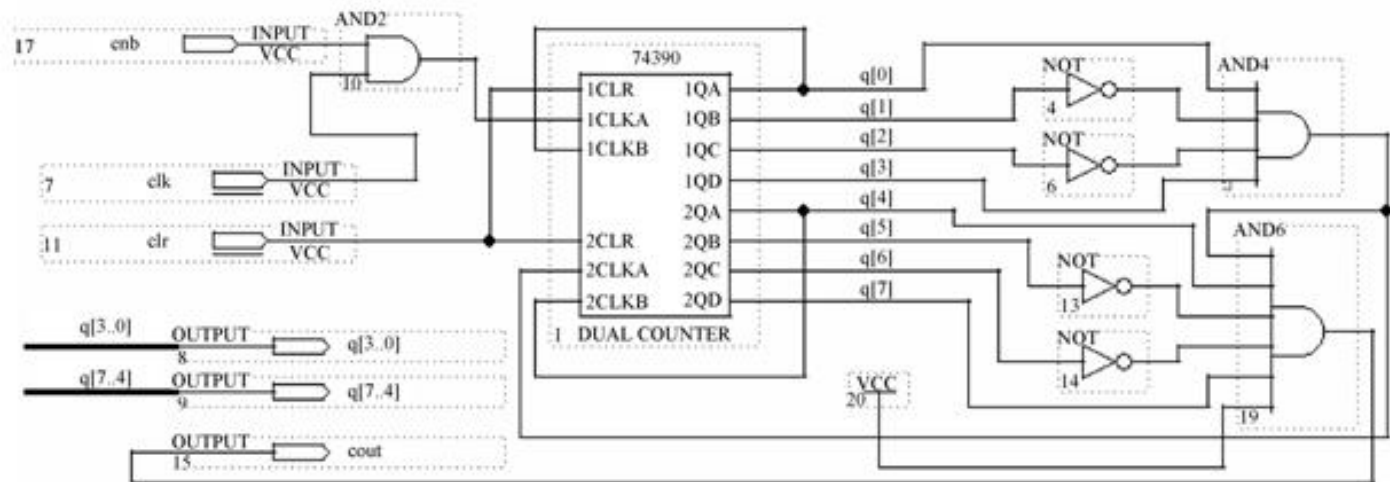


图 4-55 含有时钟使能的 2 位十进制计数器

实验与设计

(2) 建立工程。

(3) 系统仿真。

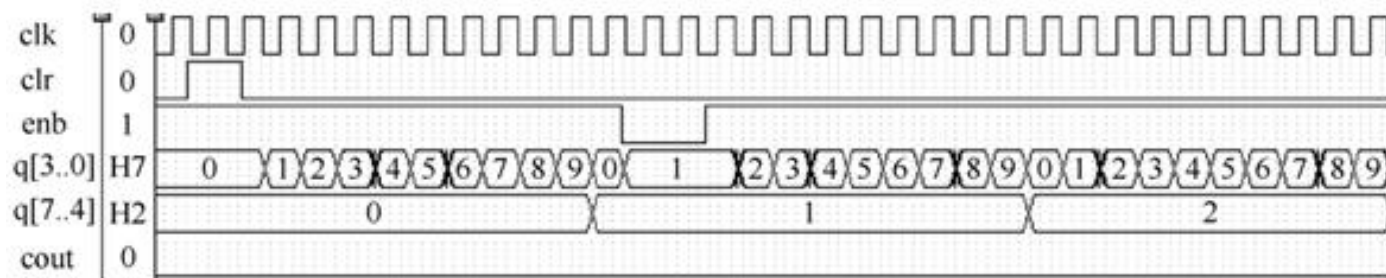


图 4-56 2 位十进制计数器工作波形

(4) 生成元件符号。

实验与设计

2. 频率计主结构电路设计

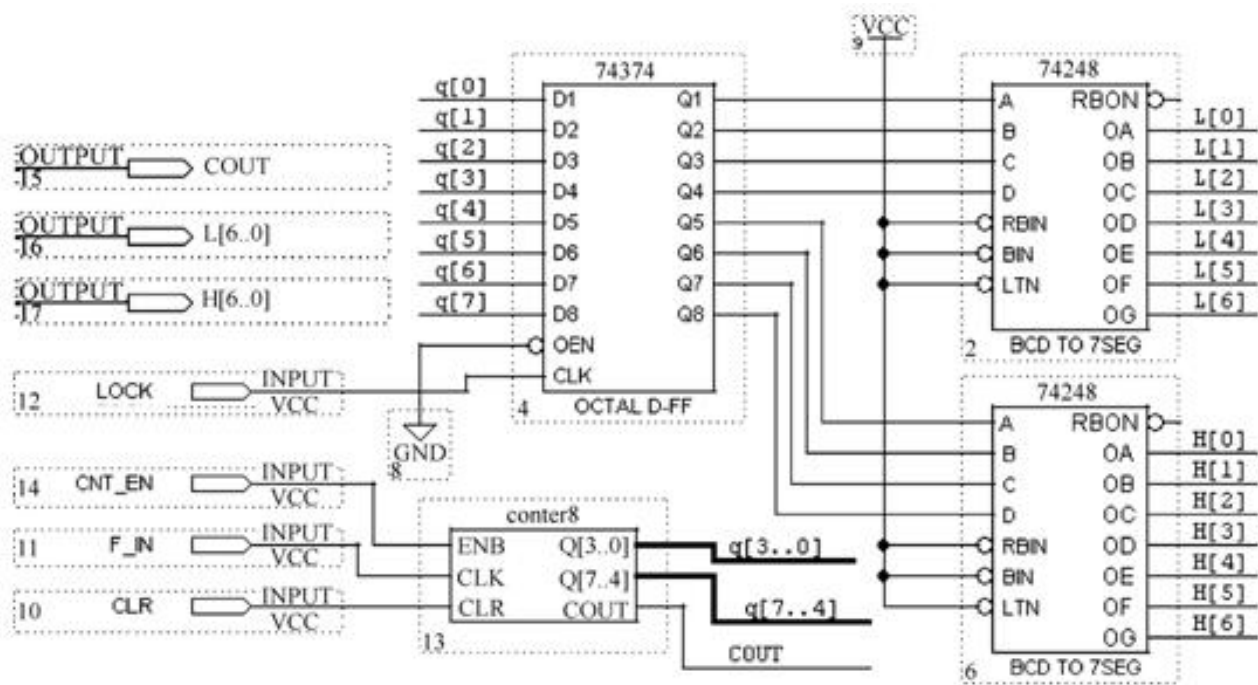


图 4-57 2 位十进制频率计顶层设计原理图文件

实验与设计

2. 频率计主结构电路设计

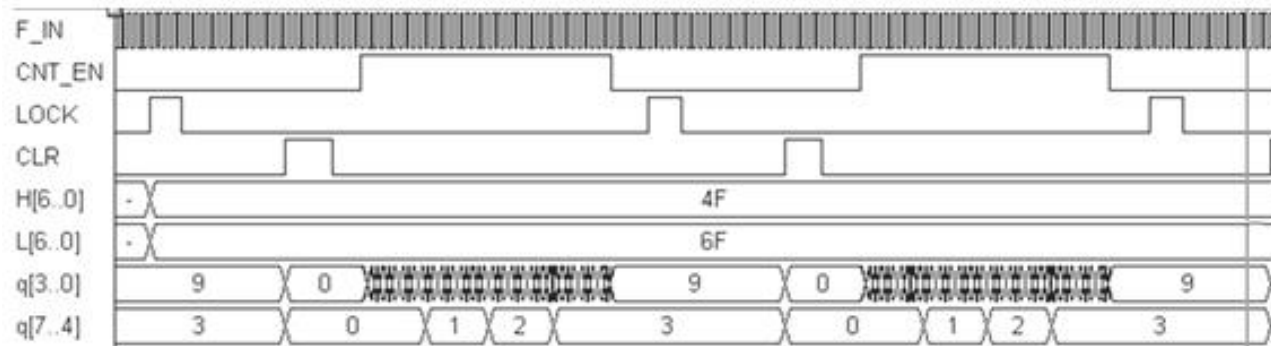


图 4-58 2 位十进制频率计测频仿真波形

实验与设计

3. 时序控制电路设计

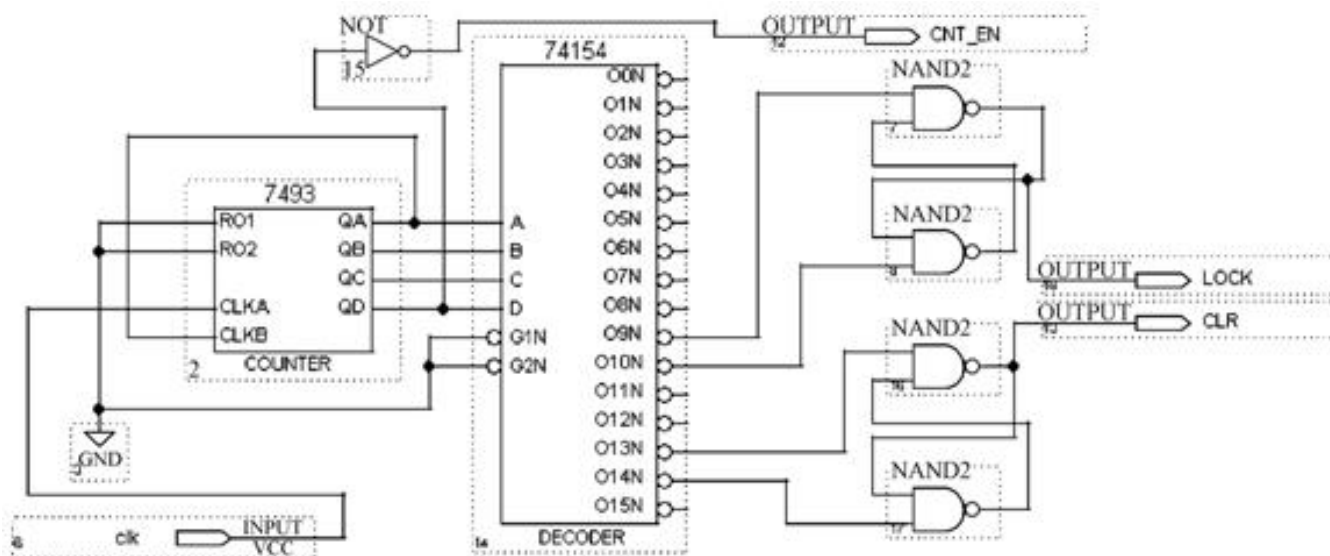


图 4-59 测频时序控制电路

实验与设计

3. 时序控制电路设计

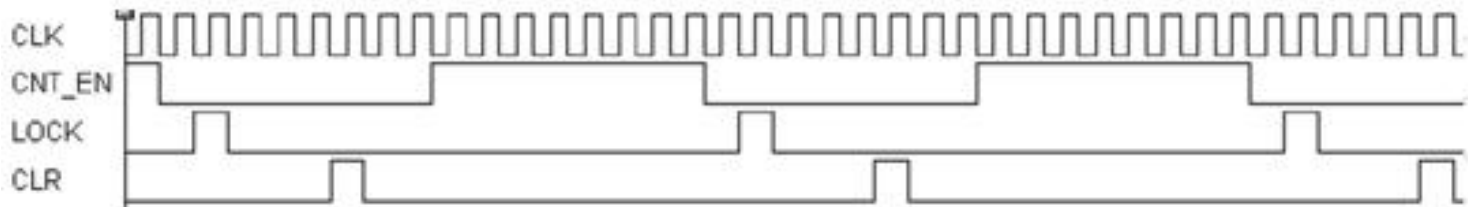


图 4-60 测频时序控制电路工作波形

实验与设计

4. 频率计顶层电路设计

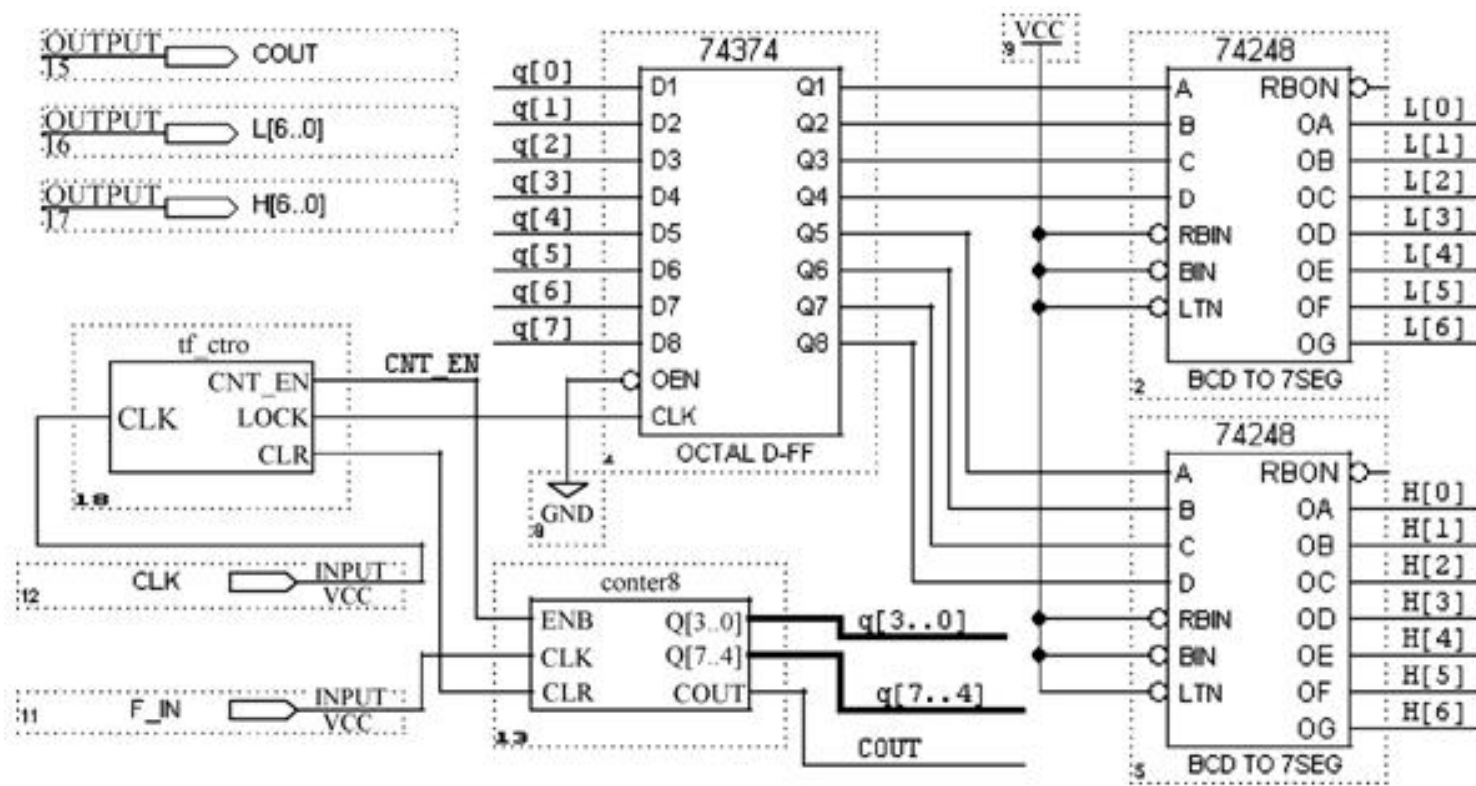


图 4-61 频率计顶层电路原理图

实验与设计

4. 频率计顶层电路设计

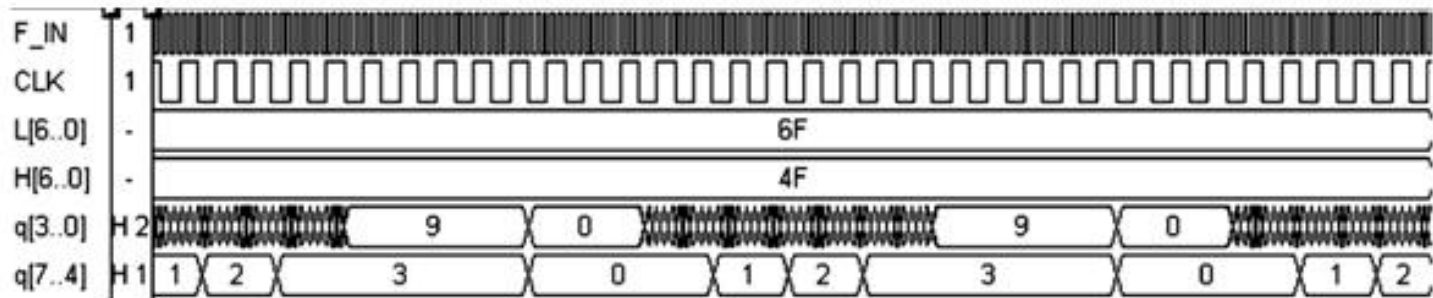


图 4-62 频率计工作时序波形

实验与设计

实验4-4 计数器设计实验

实验4-5 数码扫描显示电路设计

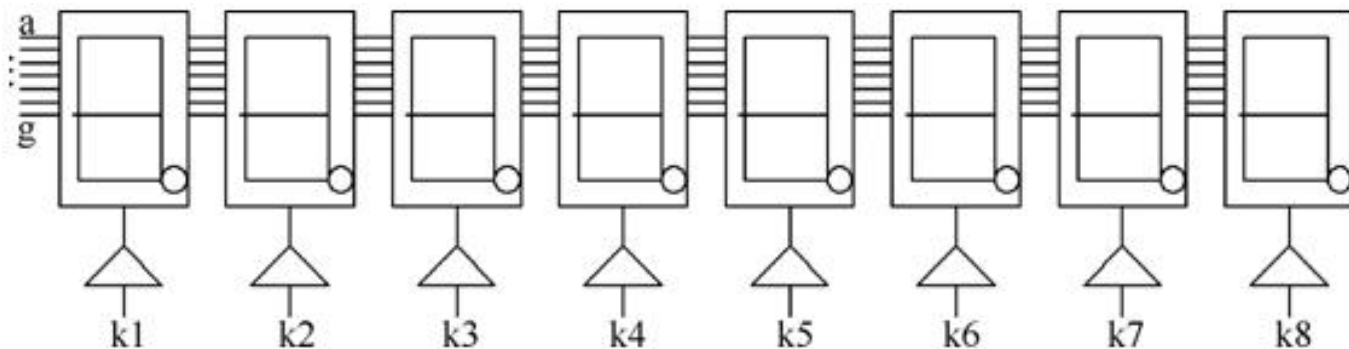


图 4-63 8 位数码扫描显示电路

实验与设计

实验4-6 硬件消抖动电路设计

【例 4-3】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY ERZP IS
    PORT ( CLK,KIN : IN STD_LOGIC;           --工作时钟和输入信号
          KOUT : OUT STD_LOGIC );          --消抖动后的输出信号
END;
ARCHITECTURE BHV OF ERZP IS
    SIGNAL KL,KH : STD_LOGIC_VECTOR (3 DOWNTO 0);
    BEGIN
    PROCESS (CLK,KIN,KL,KH) BEGIN
        IF CLK'EVENT AND CLK = '1' THEN
            IF (KIN='0') THEN KL<=KL+1;      --对键输入的低电平脉宽计数
        ELSE KL<="0000"; END IF;           --若出现高电平，则计数器清零
            IF (KIN='1') THEN KH<=KH+1;      --同时对键输入的高电平脉宽计数
        ELSE KH<="0000"; END IF;           --若出现高电平，则计数器清零
            IF (KH>"1100") THEN KOUT<='1';   --对高电平脉宽计数一旦大于12，则输出1
            ELSIF (KL>"0111") THEN KOUT<='0'; --对低电平脉宽计数若大于7，则输出0
        END IF; END IF;
    END PROCESS;
END;
```


实验与设计

实验4-6 硬件消抖动电路设计



图 4-64 例 4-3 消抖动电路仿真波形

实验与设计

实验4-7 串行静态显示控制电路设计

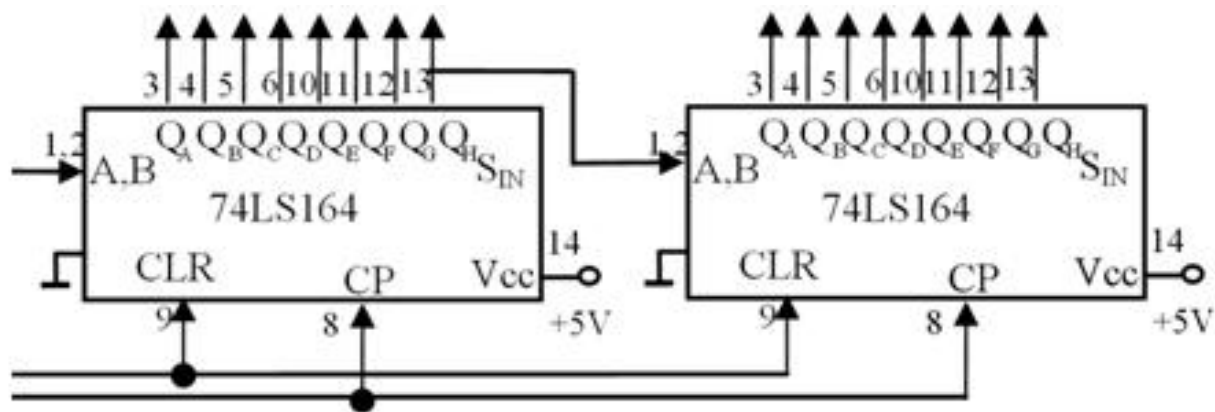


图 4-65 串/并转换数码管静态显示电路