

全球提供 FPGA 开发工具的厂商有近百家之多,大体分为两类:一类是专业软件公司研发的 FPGA 开发工具,独立于半导体器件厂商;另一类是半导体器件厂商为了推广本公司产品研发的 FPGA 开发工具,只能用来开发本公司的产品。本章介绍的 Quartus 开发工具属于后者,早期的 Quartus 由原 Altera 公司研发,Quartus 版本 15.1 之前的所有版本称作 Quartus II,从 Quartus 15.1 开始软件称作 Quartus Prime,Quartus Prime 由 Intel 公司维护。Quartus Prime 是在 Altera 公司成熟可靠而且用户友好的 Quartus II 软件基础上的优化,采用了新的高效能 Spectra-Q 引擎。Spectra-Q 引擎的 Quartus Prime 采用一组更快、更易于扩展的新算法,减少了设计迭代;同时具有分层数据库,保留了 IP 模块的布局布线,保证了设计的稳定性,避免了不必要的时序收敛投入,使其所需编译时间在业界最短,增强了 FPGA 和 SoC FPGA 设计性能。

Quartus II 和 Quartus Prime 的主要功能基本相同,只是有些界面有所不同。本章以 Quartus Prime 16.0 的基本使用方法为例进行设计开发环境的介绍。Quartus Prime 16.0 提供的功能很多,读者可参考其他书籍或 Quartus Prime 用户手册,学习更多的内容。

3.1 Quartus Prime 概述

Quartus Prime 支持 Intel 公司的各系列可编程逻辑器件的开发,包括 Cyclone 系列、Arria 系列、MAX 系列、Stratix 系列等。

Quartus Prime 提供了与第三方开发工具的无缝连接,支持 Cadence、Mentor、Synopsys 等专业软件公司的综合工具和校验工具,能读入和生成标准的 EDIF、VHDL 及 Verilog HDL 网表文件。无论使用个人计算机、UNIX 或 Linux 工作站,Quartus Prime 都提供了方便的实体设计、快速的编译处理以及编程功能。

运行 Quartus Prime,可以看到 Quartus Prime 的管理器窗口如图 3-1 所示。管理器窗口主要包含项目导航窗口、任务窗口、消息窗口,可以通过 View→Utility Windows 菜单下的选项添加或隐藏这些窗口。

为了保证 Quartus Prime 的正常运行,第一次运行软件时,需要设置 license.dat 文件,否则工具的许多功能将被禁用。在 Quartus Prime 管理器窗口选择 Tools→License Setup...,单击 License file 的“...”按钮,在出现的对话框中选择 license.dat 文件或直接输入具有完整路径的文件名,如图 3-2 所示。

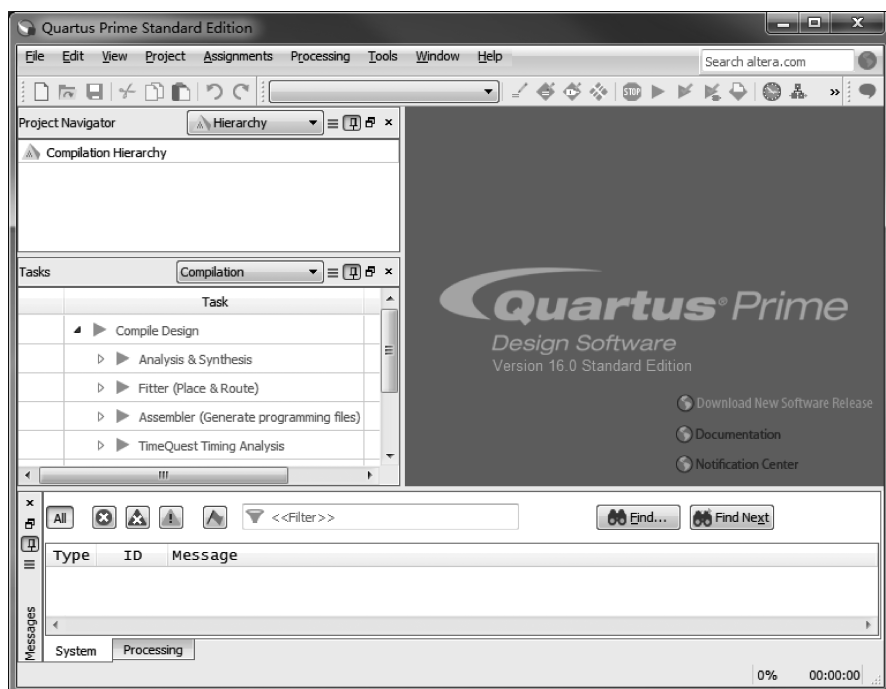


图 3-1 Quartus Prime 16.0 管理器窗口

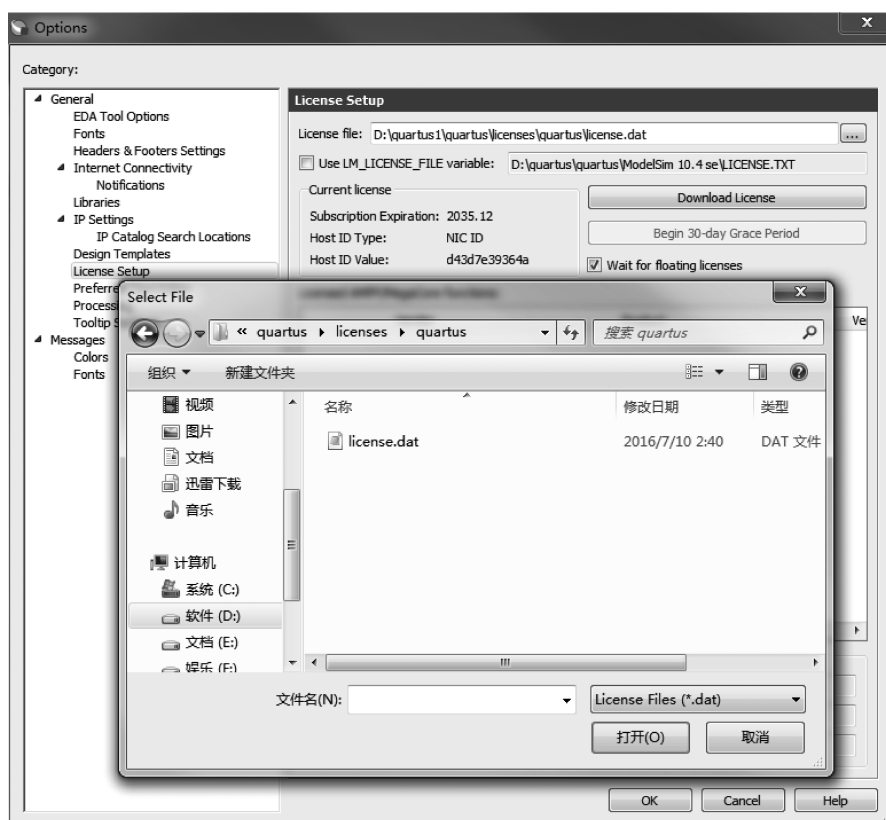


图 3-2 设置 license.dat 文件

3.2 Quartus Prime 设计流程

使用 Quartus Prime 开发工具进行 FPGA 器件的开发和应用,其过程主要有设计输入、设计处理、波形仿真和器件编程等阶段。在设计任何阶段出现错误,都需要进行修改,纠正错误,重复上述过程,直至每个阶段都正确为止。

下面将以一个 6 位二进制计数器设计项目 myexam 设计为例,介绍 Quartus Prime 的使用流程,介绍如何经过设计各个阶段,最终将 myexam.vhd 设计下载到 FPGA 芯片,完成 6 位二进制计数器设计的完整过程。

3.2.1 设计输入

Quartus Prime 编辑器的工作对象是项目,项目用来管理所有设计文件以及编辑设计文件过程中产生的中间文档,建议读者在开始设计之前先建立一个文件夹,方便项目的管理。在一个项目下,可以有多个设计文件,这些设计文件的格式可以是原理图文件、文本文件(如 AHDL、VHDL、Verilog HDL 等文件)、符号文件、底层输入文件以及第三方 EDA 工具提供的多种文件格式,如 EDIF、Tcl 等。下面以文本文件为例,学习设计输入过程中的主要操作。

1. 建立设计项目

在 Quartus Prime 管理器窗口中选择菜单 File→New Project Wizard...,出现新建项目向导 New Project Wizard 对话框的第一页,如图 3-3 所示。在对话框下输入项目目录、项目名称和顶层实体文件名,如 myexam。顶层实体文件名可以与项目名称不一致,系统默认一致的名称。

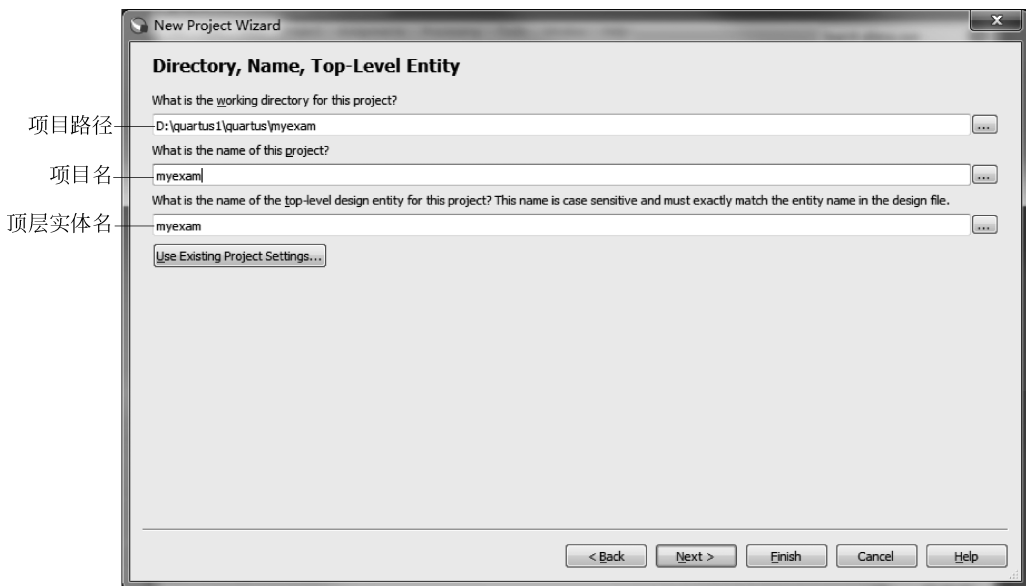


图 3-3 New Project Wizard 对话框第一页

新建项目向导第三页,单击按钮“...”可浏览文件选项,添加或删除与该项目有关的文件。初学者还没有建立文件,可以先跳过该页。

新建项目向导第四页,根据器件的封装形式、引脚数目和速度级别,选择目标器件。用户可以根据具备的实验条件进行选择,这里选择的芯片是 Cyclone V 系列中 5CSEMA5F31C6 芯片,如图 3-4 所示。

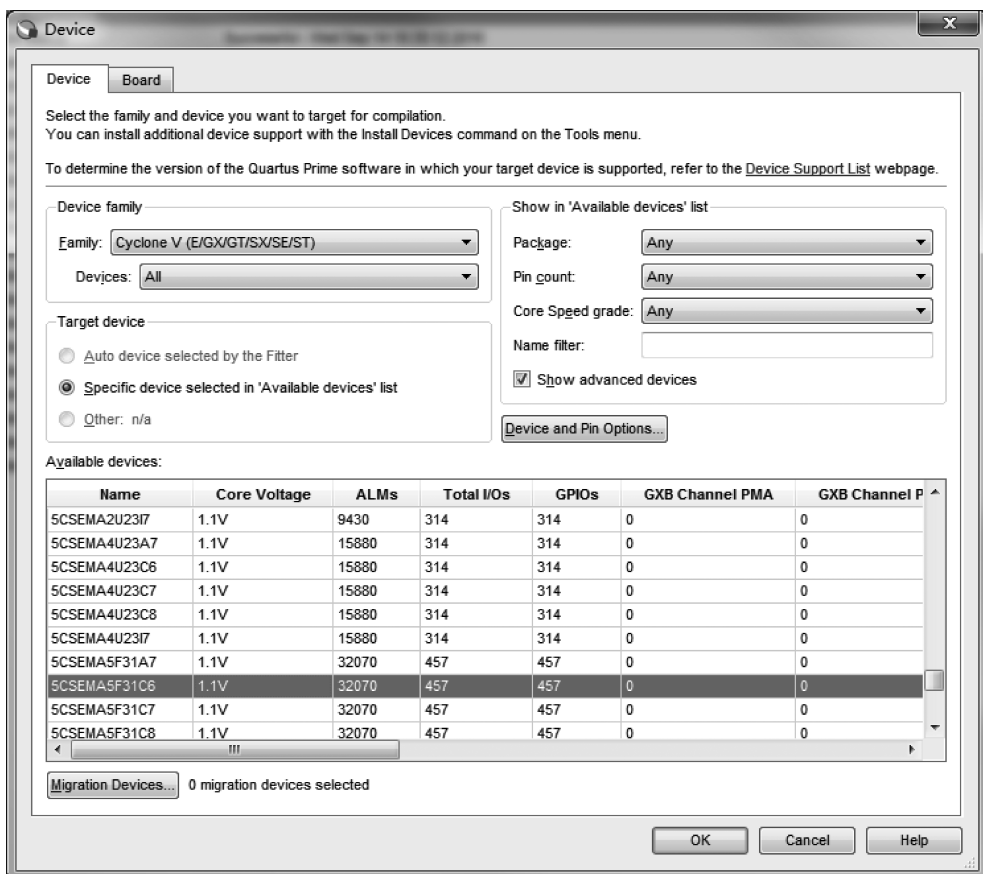


图 3-4 New Project Wizard 对话框第四页

新建项目向导第五页,添加第三方 EDA 综合、仿真、定时等分析工具,系统默认选择 Quartus Prime 的分析工具,对开发工具不熟悉的读者,建议采用系统默认选项。

在新建项目向导对话框的最后一页,给出前面输入内容的总览。单击 Finish 按钮,myexam 项目出现在项目导航窗口,myexam 表示顶层实体文件,如图 3-5 所示。在任务窗口出现设计项目过程中的全部操作,执行操作命令的方法可以在菜单栏下选择命令、单击工具栏中对应的工具按钮或者在任务窗口双击命令。

新建项目向导中的各个选项,在新建项目结束后,仍然可以修改或重新进行设置,通过选择菜单命令 Assignments→Settings...→General 实现。

2. 输入文本文件

Quartus Prime 支持 AHDL、VHDL 及 Verilog HDL 等硬件描述语言描述的文本文件,关于如何用 VHDL 描述硬件电路请参考本书中的第 2 章。这里将结合实例说明如何使

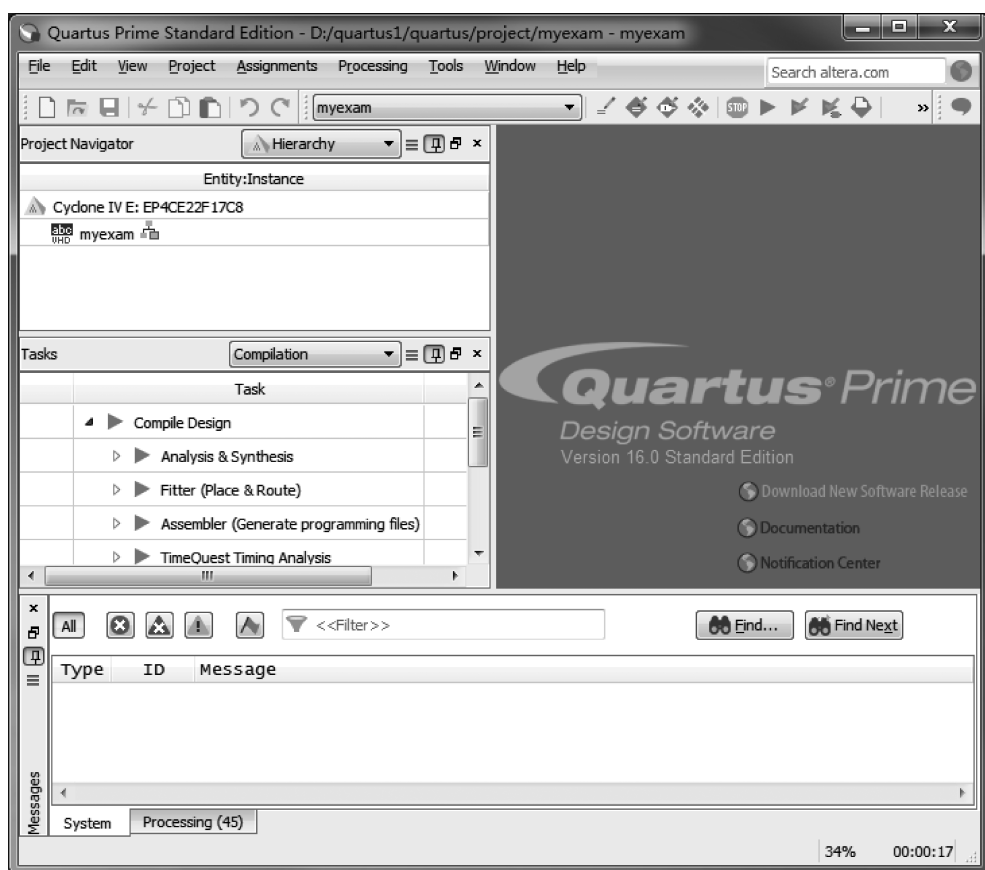


图 3-5 建立项目 myexam

用文本编辑器模板输入 VHDL 文本文件。

新建 VHDL 文本文件,在 Quartus Prime 管理器界面中选择菜单 File→New...,或单击新建文件按钮,出现 New 对话框,如图 3-6 所示。在对话框 Design Files 中选择 VHDL File,单击 OK 按钮,打开文本编辑器。在文本编辑器窗口下,按照 VHDL 语言规则输入设计文件,并将其保存,VHDL 文件的扩展名为.vhd。

Quartus Prime 支持多种硬件描述语言,不同的硬件描述语言编写的文件,其文件扩展名不同,如 AHDL 文件扩展名为.tdf,Verilog HDL 文件扩展名为.v。

Quartus Prime 提供了文本文件的编辑模板,使用这些模板可以快速准确地创建 VHDL 文本文件,避免语法错误,提高编辑效率。例如,用 VHDL 模板设计一个 6 位二进制计数器的 VHDL 文本文件。

(1) 选择菜单 Edit→Insert Template...,打开 Insert Template 对话框,单击左侧 Language templates 栏目打开 VHDL,VHDL 栏目下显示出所有 VHDL 的程序模板,如图 3-7 所示。

(2) 在 VHDL 模板中选择 Full Design→Arithmetic→Counters→Binary Counter,Insert Template 对话框的右侧会出现计数器模板程序的预览。这是一个带清零和使能端的计数器模板。单击 Insert,模板程序出现在文本编辑器中,其中蓝色的字母是关键字,绿

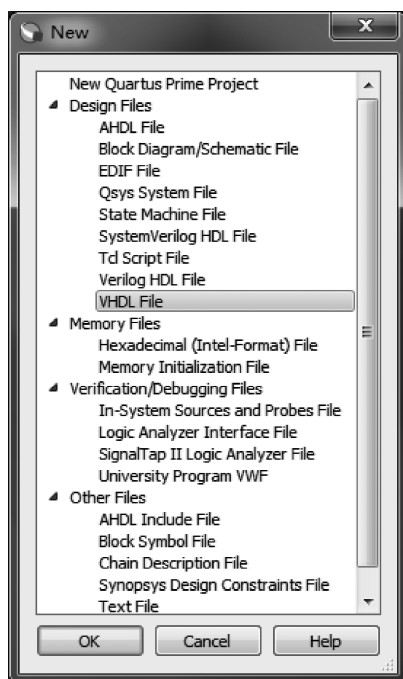


图 3-6 新建文本文件对话框

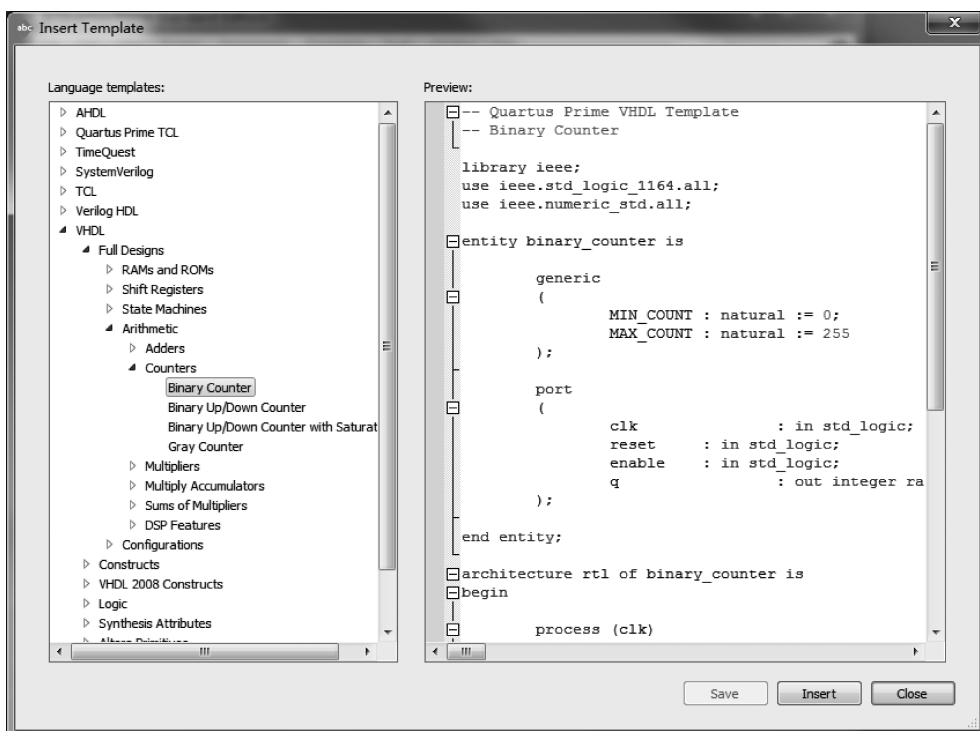


图 3-7 程序模板对话框

色部分为注释语句。

(3) 根据设计要求,对模板中的文件名、信号名、变量名等黑色部分的内容进行修改。将实体名 `binary_counter` 修改为 `myexam`; 将程序中的变量表示改为常数形式; 删掉 `enable` 输入信号等。修改后的 VHDL 代码如下:

```
-- Quartus Prime VHDL Template
-- Binary Counter
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity myexam is                                -- 实体名为 myexam
    port
    (   clk      : in std_logic;                -- 时钟信号 clk 定义
        reset    : in std_logic;                -- 复位信号 reset 定义
        q        : out integer range 0 to 63);   -- 输出信号 q 定义
end entity;

architecture rtl of myexam is
begin
    process (clk)
        variable cnt: integer range 0 to 63;
    begin
        if (rising_edge(clk)) then              -- 时钟 clk 上升沿
            if reset = '1' then                  -- 复位 reset 为高电平
                cnt := 0;                        -- 计数器复位
            else
                cnt := cnt + 1;                   -- 计数器工作
            end if;
        end if;
        q <= cnt;                                -- 输出当前的计数值
    end process;
end rtl;
```

3. 添加或删除与当前项目有关的文件

如果希望将存放在别处的文件加入到当前的设计项目中,需要选择菜单 `Assignments→Settings...`,打开如图 3-8 所示的 `Settings` 对话框。在 `Settings` 对话框左侧的 `Category` 栏目下选择 `Files` 项,通过右边 `File Name` 栏的“...”按钮查找文件选项,单击 `Add` 按钮添加文件。`Add All` 按钮的作用是将当前目录下的所有文件添加到项目中。

如果希望将当前项目中的文件从项目中删除,首先选中待删除文件,`Remove` 按钮则被激活,单击 `Remove` 按钮即可。

如图 3-8 所示,在 `Settings` 对话框,除了可以进行设计项目的文件设置外,还可以进行与设计有关的各种其他功能设置,如:库 `Libraries`、IP、EDA Tool、Compilation、定时分析 `Timing Analysis`、SSN Analyzer 等设置。

4. 指定目标器件

如果在建立项目时,没有指定目标器件,可以通过选择菜单 `Assignments→Device...`,

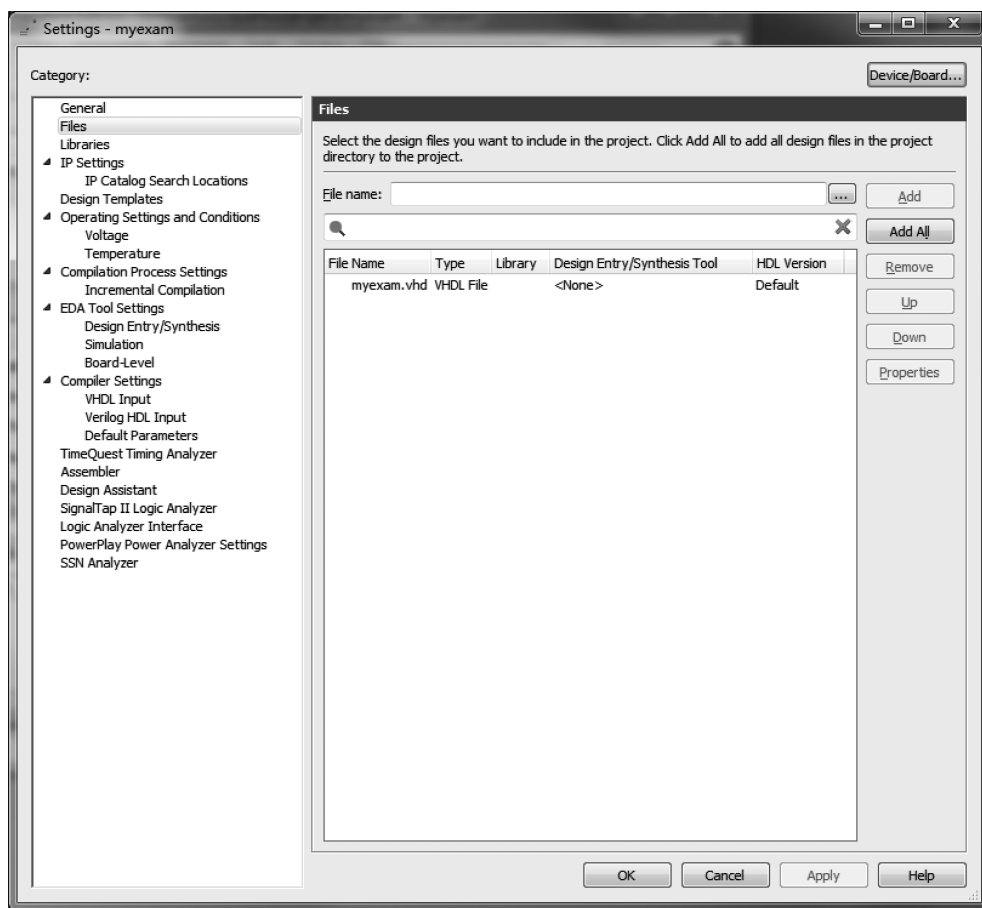


图 3-8 Settings 对话框

打开如图 3-9 所示的 Device 对话框,指定设计项目使用的目标器件。在 Family 下拉列表中选择器件系列;在 Show in 'Available devices' list 中选择封装形式、引脚数和速度级别;在 Available devices 中选择目标器件;单击 Device and Pin Options 按钮,出现器件和引脚选项对话框,根据设计需要进行配置、编程文件、不用引脚、双用途引脚以及引脚电压等选项的详细设置。

3.2.2 设计处理

Quartus Prime 设计处理的功能包括设计错误检查、逻辑综合、器件配置以及产生下载编程文件等,称作编译 Compilation。编译后生成的编程文件可以用 Quartus Prime 编程器或其他工业标准的编程器对器件进行编程或配置。

编辑设计文件后可以直接执行编译 Compilation 操作,对设计进行全面的设计处理。也可以分步骤执行,首先进行分析和综合处理 Analysis & Synthesis,检查设计文件有无错误,基本分析正确后,再进行项目的完整编译 Compilation。

1. 设置编译器

初学者如果选择系统默认的设置,可以跳过编译器设置。

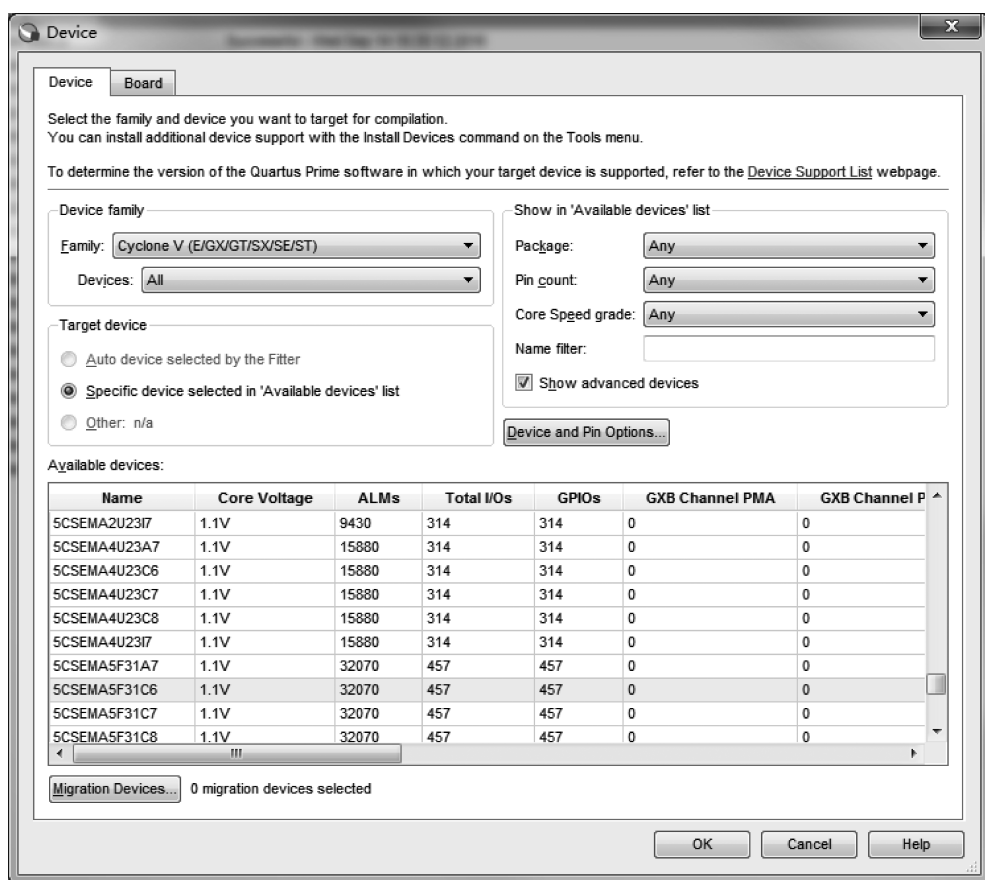


图 3-9 Device 对话框

如果确实需要对编译器进行专门的设置,选择菜单 Assignments→Settings...,在 Settings 对话框 Category 栏目下选择 Compilation Process Settings 项,可以设置与编译相关的内容,如图 3-10 所示。

2. 执行编译

Quartus Prime 软件实行的是项目管理,一个项目中可能会有多个文件,如果要对其中的某一个文件进行编译处理时,需要将该文件设置成顶层文件。

设置顶层文件:首先打开准备进行编译的文件,如打开前面编辑的文件 myexam.vhd,执行菜单命令 Project→Set as Top-Level Entity。下面进行设计处理的各项操作就是针对这一顶层文件 myexam.vhd 进行的。

执行编译:选择菜单 Processing→Start Compilation 或直接单击工具栏中编译按钮,开始执行编译操作,对设计文件进行全面的检查,编译操作结束后,出现如图 3-11 所示的界面,界面中给出编译后的信息。

任务窗口:显示编译过程中编译进程以及具体操作的项目。

信息窗口:显示所有信息、警告和错误。如果编译有错误,需要修改设计,重新进行编译。双击某个错误信息项,可以定位到原设计文件并高亮显示。

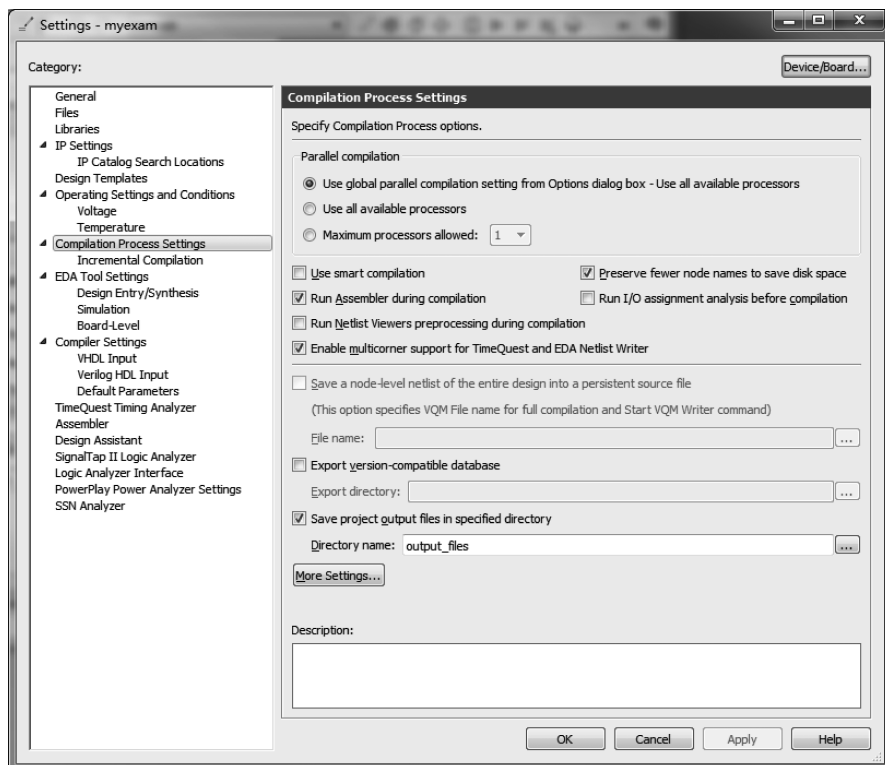


图 3-10 在 Settings 对话框的编译设置选项

4.编译报告栏 1.编译快捷按钮 5.编译总结报告

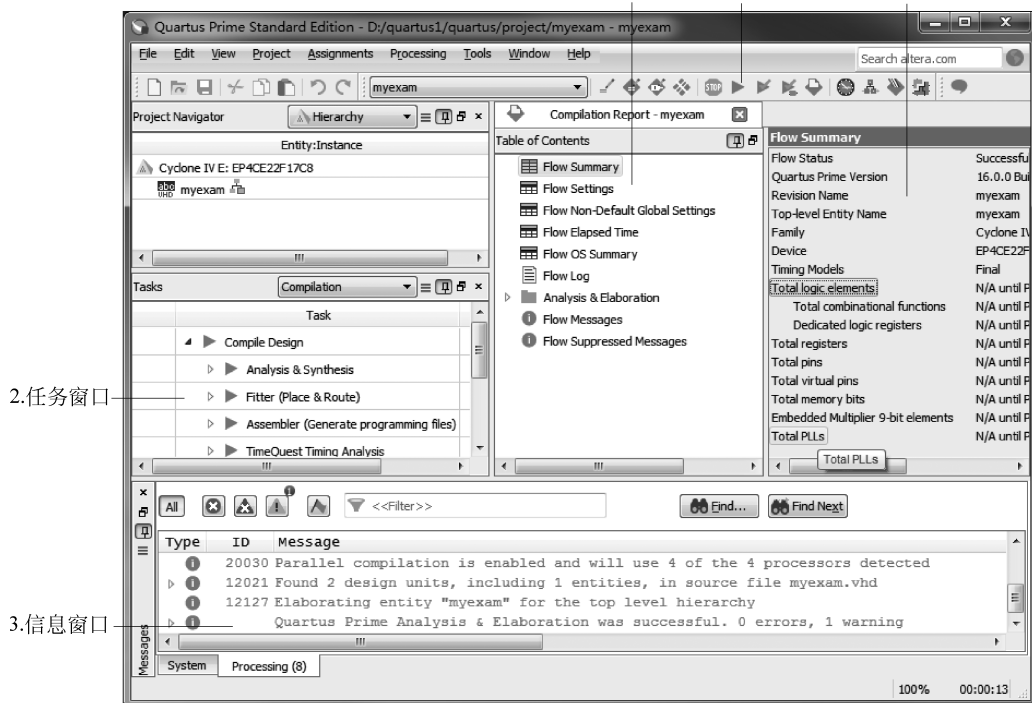


图 3-11 设计项目的编译

编译报告栏：编译完成后显示编译报告，编译报告栏包含了将一个设计编译正确后，将设计放到器件中的所有信息，如器件资源统计、编译设置、底层显示、器件资源利用率、适配结果、延时分析结果等。编译报告栏是一个只读窗口，选中某项可获得详细信息。

编译总结报告：编译完成后直接给出该报告，报告中给出编译的主要信息：项目名、文件名、选用器件名、占用器件资源、使用器件引脚数等。

3. 锁定引脚

锁定引脚是指将设计文件的输入输出信号分配到器件指定引脚，这是设计文件下载到FPGA芯片必须完成的过程。在Quartus Prime中，锁定引脚分为前锁定和后锁定两种。前锁定指的是编译之前的引脚锁定，后锁定是指对设计项目编译后的引脚锁定，这里介绍后锁定引脚的操作过程。

值得注意的是，在后锁定引脚完成之后，必须再次进行编译。

选择菜单 Assignments→Pins Planner，出现 Pins Planner 对话框如图 3-12 所示。由于设计项目已经进行过编译，因此在节点列表区会自动列出所有信号的名称，在需要锁定的节点名处，双击引脚锁定区 Location，在列出的引脚号中进行选择。例如，选择 clk 节点信号，锁定在 PIN_AF14 号引脚上，如图 3-12 所示。重复此过程，逐个进行引脚锁定，所有引脚锁定完成后，再次进行编译。

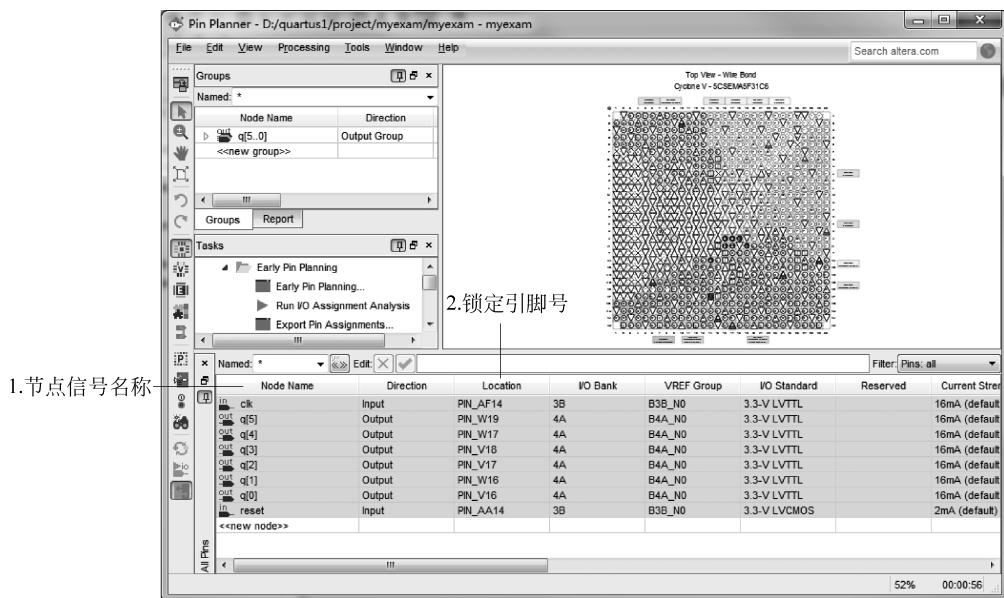


图 3-12 引脚锁定

3.2.3 波形仿真

当一个设计项目的编译通过之后，能否实现预期的逻辑功能，仍需要进一步的检验，波形仿真分析是必不可少的一个环节。波形仿真就是在波形编辑器中将设计的逻辑功能用波形图的形式显示，通过查看波形图，检查设计的逻辑功能是否符合设计要求。Quartus II 13.0 及之后的版本包含了 Simulation Waveform Editor 仿真工具，除此之外，Quartus Prime 16.0 也支持 ModelSim、questasim 等第三方仿真工具软件，Simulation Waveform

Editor 仿真也借助了仿真工具 ModelSim。如果安装了 ModelSim 和 ModelSim-Altera, Simulation Waveform Editor 默认选择 ModelSim-Altera。本节主要以 Simulation Waveform Editor 和 ModelSim 为例介绍仿真流程。

1. ModelSim 仿真

ModelSim 是 Mentor Graphics 公司开发的一款功能强大的仿真软件,具有速度快、精度高和便于操作的特点,此外还具有代码分析能力,可以看出不同代码段消耗资源的情况。ModelSim 的功能侧重于编译和仿真,但不能指定编译的器件和下载配置,需要和 Quartus Prime 等软件关联。

在 Quartus Prime 16.0 界面菜单栏中选择 Tools→options 选项卡中的 EDA tool options,在 ModelSim 一项指定 ModelSim 安装的路径。本文安装并指定的 ModelSim 路径为 D:\quartus\quartus\ModelSim 10.4 se\win64。

在 Quartus Prime 16.0 界面菜单栏中选择 Assignments→Settings。选中该界面中 EDA Tool settings 中的 Simulation。在 Tool name 中选择 ModelSim,Format for output netlist 中选择开发语言的类型 VHDL(如果项目是基于 Verilog 语言写的,则此处选择 Verilog),如图 3-13 所示。然后单击 Apply 和 OK 按钮。

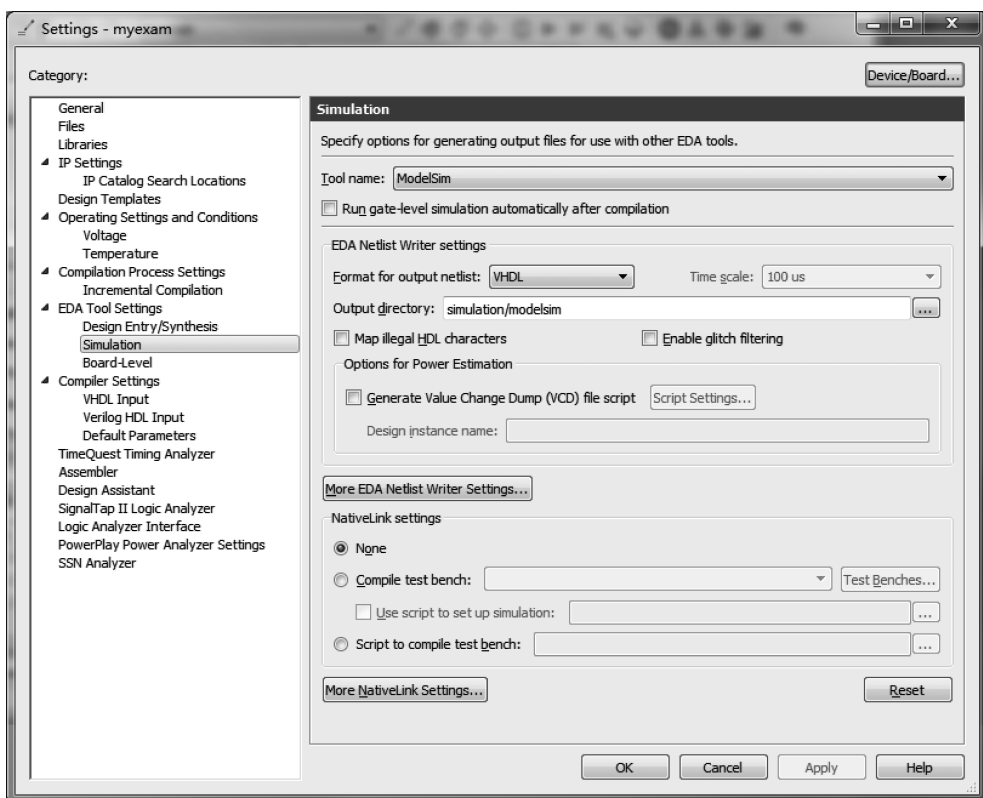


图 3-13 设置仿真工具

设置完成后,编译工程。在 Quartus Prime 16.0 菜单栏中选择 Processing→Start Compilation,等待编译,编译无错后会在 myexam 目录下生成 simulation 目录。单击菜单

栏 Processing→Start→Start Test Bench Template Writer,如图 3-14 所示,在 myexam/simulation/modelsim 下会生成一个与项目顶层文件同名的 testbench 测试文件模板: myexam.vht(Verilog 语言环境下生成的测试文件为 myexam.vt)。

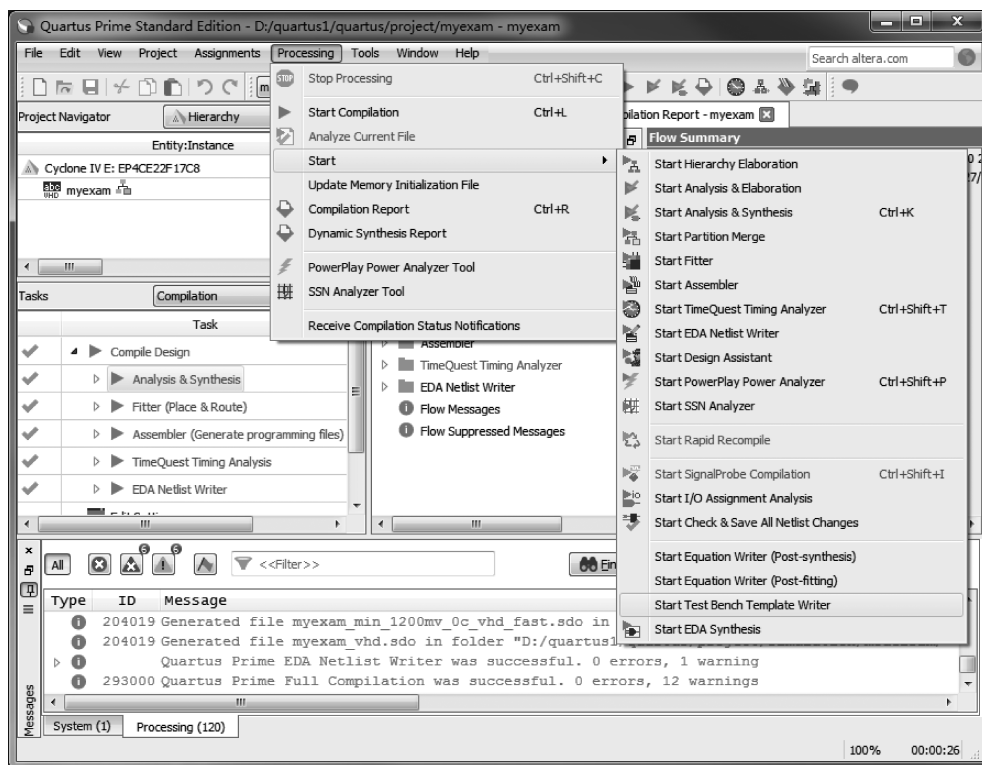


图 3-14 生成 test bench 文件模板

打开 myexam.vht 文件,可以看到此时生成的 testbench 文件是基于 VHDL 语言的,程序结构包含了 library 库、无端口实体 myexam_vhd_tst、结构体 myexam_arch,程序如下所示。结构体分为三部分:信号定义、实例化、施加激励。施加激励通过程序中的两个进程实现,设计者需要根据测试需求,设计需要的激励信号,其中 init 进程主要产生执行一次的激励信号,如复位信号、非周期性输入信号等;always 进程中主要产生由敏感事件列表触发的信号,如时钟信号、周期性输入信号等。

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY myexam_vhd_tst IS
END myexam_vhd_tst;
ARCHITECTURE myexam_arch OF myexam_vhd_tst IS
    -- constants
    -- signals
    SIGNAL clk : STD_LOGIC;
    SIGNAL q : STD_LOGIC_VECTOR(5 DOWNT0 0);
    SIGNAL reset : STD_LOGIC;
    COMPONENT myexam

```

```

    PORT (
        clk : IN STD_LOGIC;
        q : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
        reset : IN STD_LOGIC
    );
END COMPONENT;
BEGIN
    il : myexam
    PORT MAP (
        -- list connections between master ports and signals
        clk => clk,
        q => q,
        reset => reset
    );
    init : PROCESS
        -- variable declarations
    BEGIN
        -- code that executes only once
    WAIT;
    END PROCESS init;
    always : PROCESS
        -- optional sensitivity list
        -- (      )
        -- variable declarations
    BEGIN
        -- code executes for every event on sensitivity list
    WAIT;
    END PROCESS always;
END myexam_arch;

```

根据测试需求在模板中修改测试文件,在 init 进程中添加 reset 激励信号,always 进程中添加周期为 20ns 的时钟信号,编写完成仿真测试文件后保存。

```

LIBRARY ieee;                                -- 使用 IEEE 库
USE ieee.std_logic_1164.all;                 -- 使用 std_logic_1164 程序包的所有设计单元

ENTITY myexam_vhd_tst IS                      -- 实体描述
END myexam_vhd_tst;                          -- 结束实体描述
ARCHITECTURE myexam_arch OF myexam_vhd_tst IS -- 结构体描述
    SIGNAL clk : STD_LOGIC := '1';            -- 内部时钟信号 clk 的定义
    SIGNAL q : integer range 0 to 63;         -- 内部信号 q 的定义
    SIGNAL reset : STD_LOGIC := '1';          -- 内部复位信号 reset 的定义
    COMPONENT myexam                          -- 元件说明语句,形成底层元件
    PORT (
        clk : IN STD_LOGIC;
        q : OUT integer range 0 to 63;
        reset : IN STD_LOGIC
    );
END COMPONENT;
BEGIN

```

```

    il : myexam                                -- 元件例化语句,调用底层元件
    PORT MAP (
-- list connections between master ports and signals
        clk => clk,
        q  => q,
        reset => reset
    );
init : PROCESS                                -- init 进程
    BEGIN
        wait for 4 ns; reset <= '0';          -- 4ns 后复位信号无效
        WAIT;
    END PROCESS init;                          -- 结束进程 init
always : PROCESS                              -- always 进程
    BEGIN
        wait for 10 ns; clk <= not clk;        -- 产生周期为 20ns 的时钟信号
    END PROCESS always;                       -- 结束进程 always
END myexam_arch;                             -- 结构体结束

```

在 Quartus Prime 界面菜单栏中选择 Assignments→Settings→EDA Tool Settings→Simulation 界面,在界面 NativeLink settings 项中单击 Compile test bench 右边的 Test Benches 按钮,如图 3-15 所示。弹出界面如图 3-16 所示,在界面中单击 New...按钮。在新

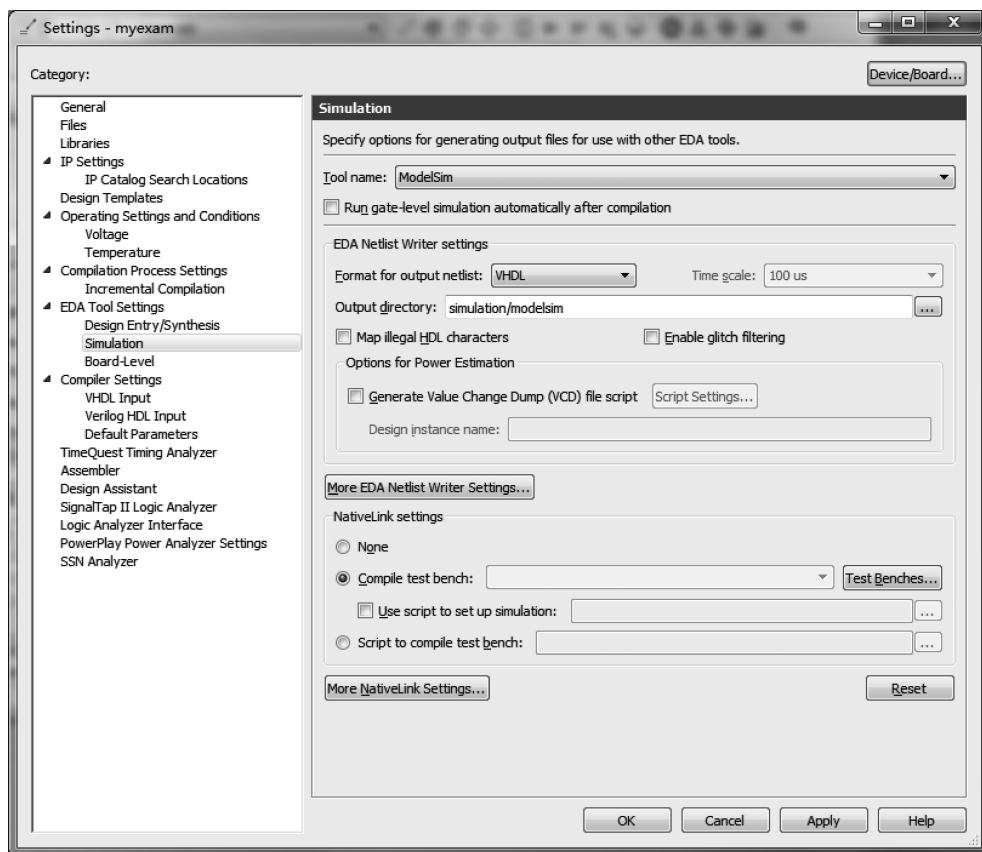


图 3-15 选择仿真文件步骤 1

出现的界面图 3-17 中 Test bench name 输入测试文件名字,在 Top level module in test bench 栏中输入测试文件中的顶层模块名。选中 Use test bench to perform VHDL timing simulation 并在 Design instance name in test bench 中输入设计测试文件中设计例化名默认为 i1。然后在 Test bench and Simulation files 栏下的 File name 选择测试文件 myexam.vht,然后单击 add 按钮,单击 OK 按钮设置完成。

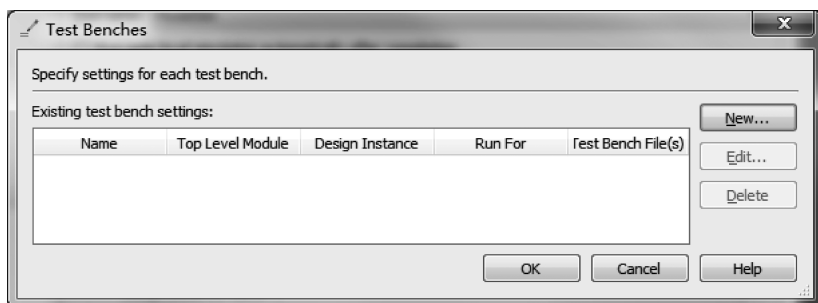


图 3-16 选择仿真文件步骤 2

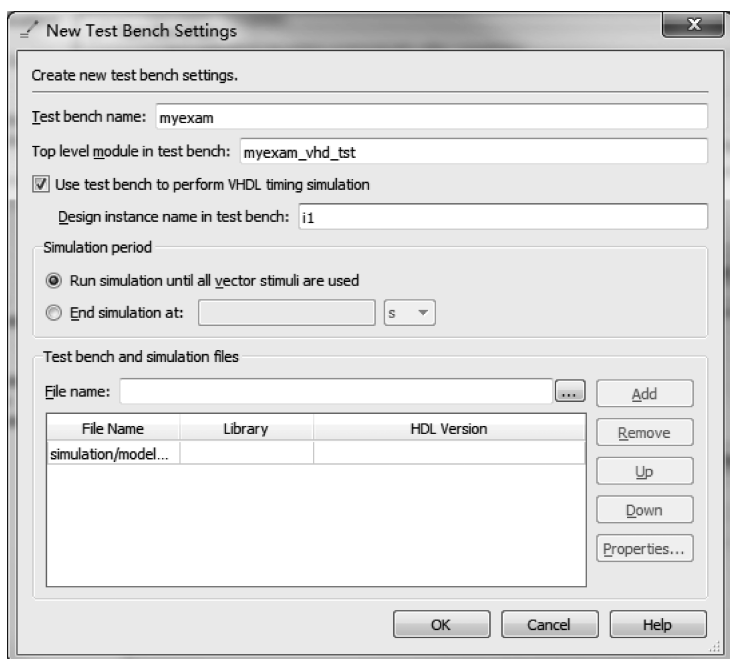


图 3-17 选择仿真文件步骤 3

仿真文件配置完成后回到 Quartus Prime 16.0 开发界面,在菜单栏中选择菜单栏 Tools 中的 Run Simulation Tool→RTL Simulation 进行行为级仿真,即功能仿真,接下来就可以看到 ModelSim 的运行界面,观察仿真波形如图 3-18 所示。通过功能仿真波形,可以验证设计文件逻辑功能的正确性。如果选择 Run Simulation Tool→Gate Level Simulation 可以进行门级仿真,即时序仿真。时序仿真中可以看到信号的传输延迟,以及可能产生的竞争冒险现象。

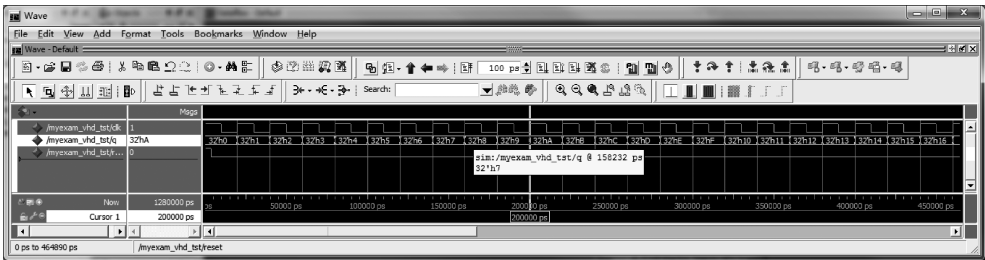


图 3-18 仿真结果

下面把基于 Verilog 语言的测试文件 myexam.vt 也提供给读者,以供参考。

```
'timescale 1 ps/ 1 ps
module myexam_vlg_tst();                                //端口信号定义
    // constants
    // general purpose registers
    // test vector input registers
    reg clk;
    reg reset;
    // wires
    wire [63:0] q;
    // assign statements (if any)
    myexam i1 (                                          //元件例化
        .clk(clk),
        .q(q),
        .reset(reset)
    );
    initial                                              // initial 语句
    begin                                              //内部放置只需执行一次的激励信号
        $display("Running testbench");
        clk <= 0;
        reset <= 1;
        #4reset <= 0;                                  //4ns 后复位信号无效
    end
    always #10 clk <= ~clk;                             //always 语句,产生周期为 20ns 的时钟信号
endmodule
```

2. Simulation Waveform Editor 仿真

当 myexam 工程编译成功后,在 Quartus Prime 管理器界面中选择菜单 File→New,或单击新建文件按钮,出现 New 对话框。在对话框 Verification→Debugging Files 中选择 University Program VWF,单击 OK 按钮,然后弹出 Simulation Waveform Editor 界面,如图 3-19 所示。

添加信号之前先设置仿真截止时间,在管理器界面选择菜单 Edit→Set End Time,弹出界面 End Time,如图 3-20 所示。End Time 的时间范围是 10ns~100μs,如果设置的时间不在这个时间范围内,单击 OK 按钮会有时间范围设置的提示,关闭 End Time 界面。

仿真运行时间设置后,需在图 3-19 中的 Name 栏添加仿真信号。在管理器界面选择菜单 Edit→Insert→Insert Node or Bus...,或者双击图 3-19 中 Name 栏的空白处,会弹出

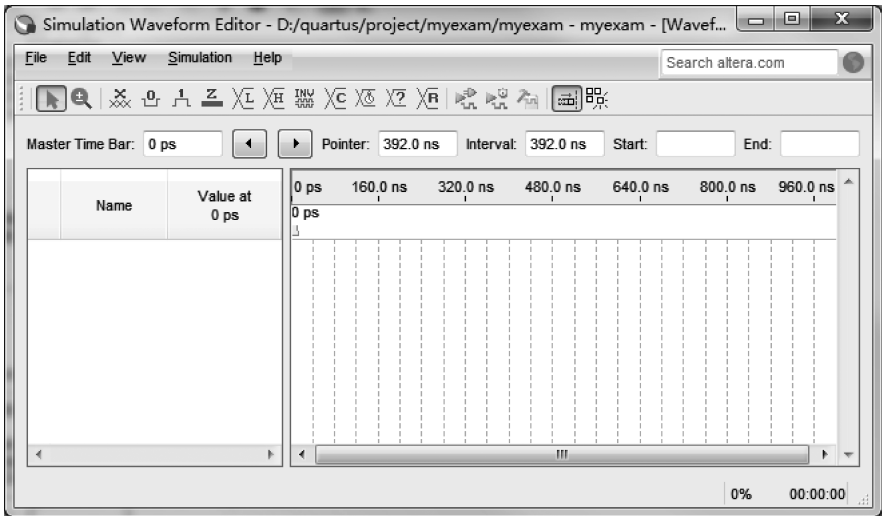


图 3-19 University Program VWF 界面

Insert Node or Bus 界面,如图 3-21 所示。图中 Name 中没有任何信号,我们需要单击 Node Finder...,弹出如图 3-22 所示 Node Finder 界面。图中 Look in 右边需要放置工程文件名,我们单击“...”,然后在弹出的界面中选择 myexam 工程文件并单击 OK,如果是对当前工程的仿真,此步可省略;接下来单击 List,myexam 工程中的信号就会出现在 Nodes Found 下方的空白处。

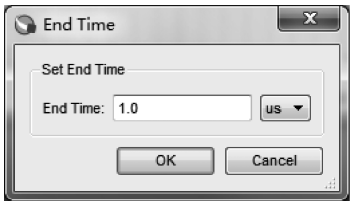


图 3-20 End Time 界面

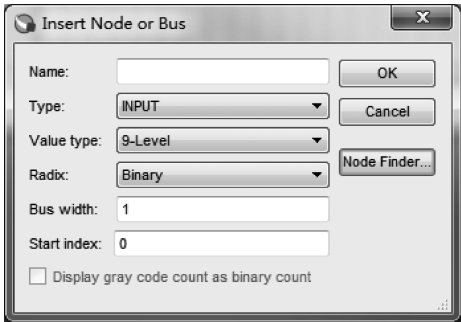


图 3-21 Insert Node or Bus 界面

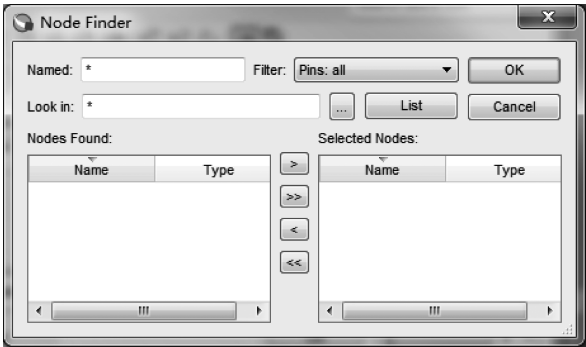


图 3-22 Node Finder 界面

在 Nodes Found 中单击需要仿真的输出信号和全部的输入信号,然后单击“>”,将选择的信号放入 Selected Nodes 栏中。不需要仿真的信号,可以单击“<”进行删除。如果需要仿真所有的信号,直接单击“>>”,Nodes Found 栏中的所有信号会出现在 Selected Nodes 栏中。当信号选定后,单击 OK 按钮,则返回到图 3-21,再单击 OK 后,信号和信号默认的波形图会出现在 Simulation Waveform Editor 界面中,如图 3-23 所示。

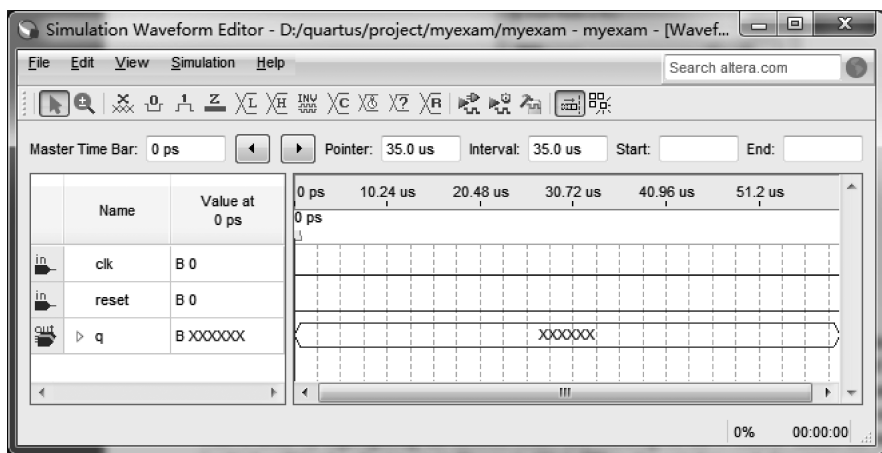


图 3-23 Simulation Waveform Editor 界面

现在需要为输入信号赋值。在 Simulation Waveform Editor 界面的图标中,共有 11 种赋值方式,设计者可以根据需要选取。我们选择  对 clk 赋值,单击  弹出 Clock 界面,将时钟周期 Period 设置为 20ns。reset 赋值时,如图 3-24 单击鼠标选中其中的一段后单击 ,选中的一段将会变成高电平 1。信号 clk 和 reset 赋值完成后,如图 3-25 所示;在管理器界面选择菜单 File→Sava As...,将文件名改为 myexam,最好与要仿真的项目同名,然后单击保存。

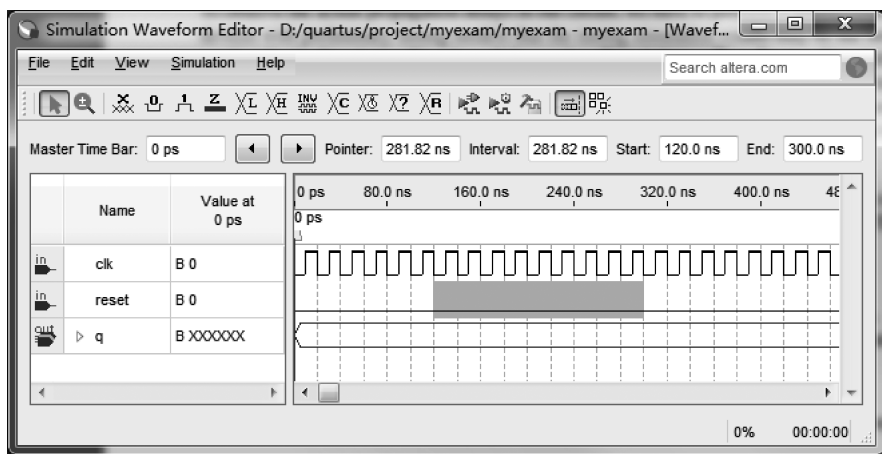


图 3-24 reset 信号赋值

Simulation Waveform Editor 包含功能仿真和时序仿真。这里进行功能仿真,在管理器界面选择菜单 Simulation→Run Functional Simulation 或者单击 ,弹出仿真进程窗

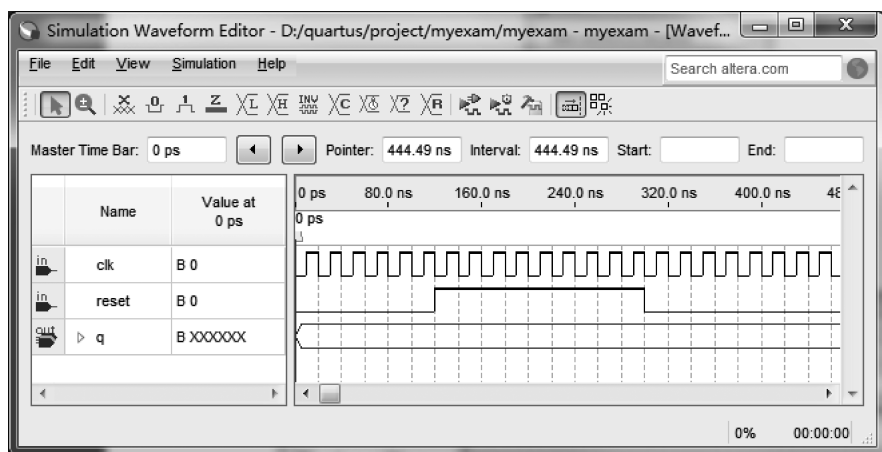


图 3-25 完成信号赋值

口,仿真完成自动关闭,并弹出包含输出波形的仿真完成界面,如图 3-26 所示。注意对输入波形的任何改动,都需要重新进行仿真。

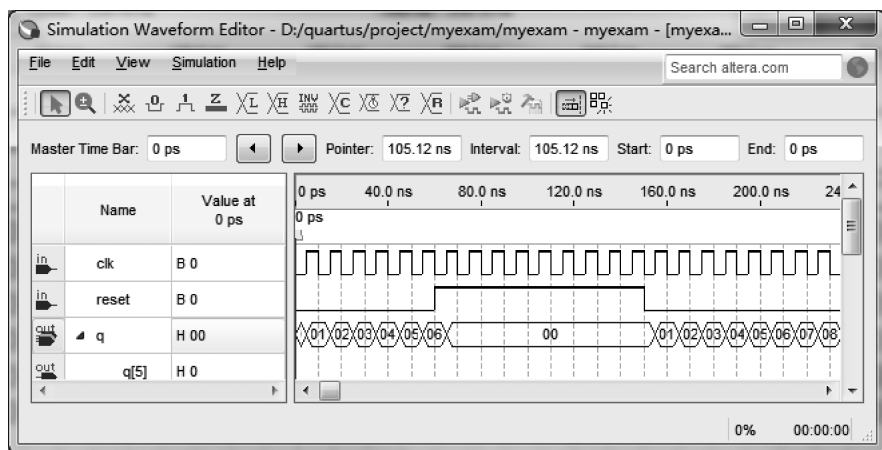


图 3-26 功能仿真图

时序仿真能观察到电路信号的实际延迟情况。只有 Cyclone IV 和 Stratix IV 支持时序仿真,如果 Quartus 工程所选择的芯片不是这两种芯片,那么时序仿真会定义为功能仿真。

3.2.4 器件编程

编译成功后,Quartus Prime 将生成编程数据文件,如 .pof 和 .sof 等编程数据文件,通过下载电缆将编程文件下载到预先选择的 FPGA 芯片中,该芯片就会执行设计文件描述的功能。

1. 编程连接

在进行编程操作之前,首先将下载电缆的一端与 PC 机对应的端口相连。使用 MasterBlaster 下载电缆编程,将 MasterBlaster 电缆连接到 PC 机的 RS-232C 串行端口。使用 ByteBlasterMV 下载电缆,将 ByteBlasterMV 电缆连接到 PC 机的并行端口。使用

USB Blaster 下载电缆,则连接到 PC 机的 USB 端口。下载电缆的另一端与编程器件相连,连接好后进行编程操作。

2. 编程操作

选择菜单 Tools→Programmer 或单击工具栏中编程快捷按钮,打开编程窗口如图 3-27 所示。读者需要根据自己的实验设备情况,进行器件编程的设置。



图 3-27 Programmer 编程窗口

作者根据自己的实验设备,进行设置的情况如下:

(1) 下载电缆 Hardware Setup...设置: USB Blaster。注意,编程设置时要保证下载电缆连接,且设备上电。

(2) 配置模式 Mode 设置: JTAG 模式。

(3) 配置文件: 自动给出当前项目的配置文件 myexam. sof。如果需要自己添加配置文件,则单击 Add File...添加配置文件。

(4) 执行编程操作: 单击编程按钮 Start,开始对器件进行编程。编程过程中进度表显示下载进程,信息窗口显示下载过程中的警告和错误信息。

(5) 实际检验: 器件编程结束后,在实验设备上实际查看 FPGA 芯片作为计数器的工作情况,应当给计数器加入频率为 1Hz 的时钟信号,方便观察计数器的变化。如果计数器工作正常,说明读者已经基本学会了 FPGA 的开发流程以及 Quartus Prime 16.0 的使用。

3. 其他编程文件的产生

Quartus Prime 在编译过程中会自动产生编程文件,如. sof 文件。但对于其他格式的文件,如二进制格式的. rbf 配置数据文件,需要专门进行设置才能产生。

编译后产生. rbf 文件过程如下: 选择菜单 File→Convert Programming File...,出现图 3-28 所示的对话框。首先,在 Output programming file 列表中选择 Raw Binary File(. rbf)。下一步,将 File name 一栏改成 myexam. rbf。然后,单击 Input files to convert 栏中的 SOF Data,此时 Add File 按钮被激活,单击 Add File 按钮,添加输入数据文件 myexam. sof,单击 Generate 即可产生. rbf 文件。查找设计项目目录,可以找到 myexam. rbf 文件。

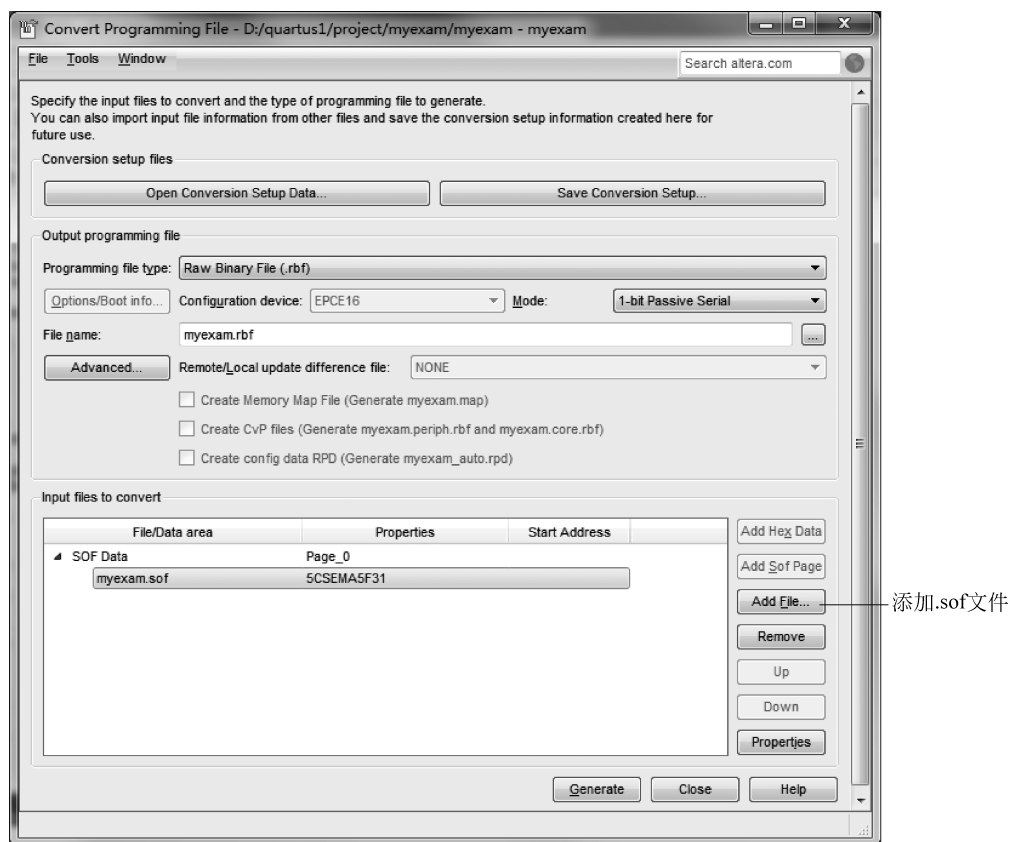


图 3-28 编译后生成.rbf 文件

3.3 嵌入式逻辑分析仪使用

Quartus Prime 软件提供了波形仿真工具,读者可以运行波形仿真工具,分析了解设计系统各信号波形。书中 3.2.3 波形仿真一节中专门介绍如何使用波形仿真工具对设计系统的信号进行波形仿真的测试,通过信号波形分析了解设计系统的工作是否正常。

这里介绍嵌入式逻辑分析仪的使用,就是将逻辑分析仪嵌入到 FPGA 芯片内部,测试 FPGA 芯片内部或外部引脚实际信号波形,分析系统工作是否正常的方法。

嵌入式逻辑分析仪的使用分为以下几个步骤:打开 Tools→Signal Tap II Logic Analyzer 编辑窗口,输入待测信号,Signal Tap II 参数设置,编译下载,运行 Signal Tap II 分析被测信号。

下面以前面已经输入的文件 myexam.vhd 为例,学习嵌入式逻辑分析仪的使用。

1. Signal Tap II 编辑窗口

选择菜单 Tools→Signal Tap II Logic Analyzer,出现 Signal Tap II 编辑窗口,如图 3-29 所示,显示一个空的 Signal Tap II 文件。

Signal Tap II 编辑窗口主要分为以下 5 个栏目:

(1) 实例管理 Instance Manager: 管理分析程序。

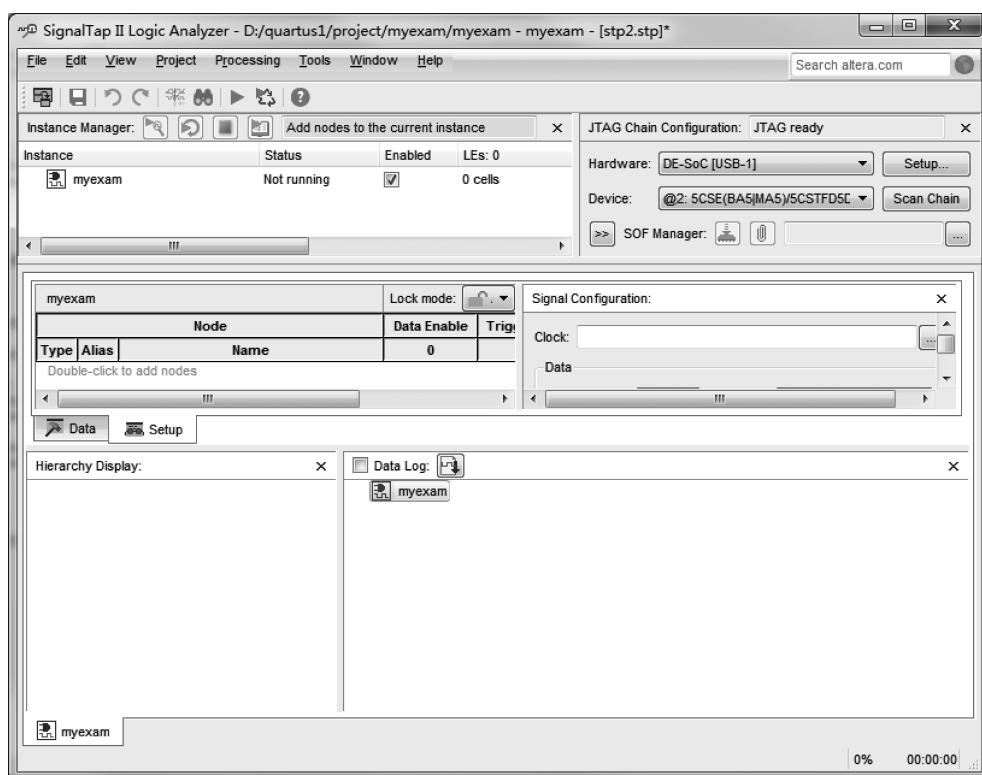


图 3-29 Signal Tap II 编辑窗口

(2) JTAG 链配置 JTAG Chain Configuration: 管理配置硬件和文件。

(3) 设置/数据 Setup/Data: 设置测试信号或者观察测试数据。

(4) 信号设置 Signal Configuration: 设置逻辑信号分析仪。

(5) 层次显示 Hierarchy Display: 显示分析文件的结构层次。

2. 输入文件和待测信号

在实例管理 Instance Manager 栏目下,单击 Instance 下面的 auto_signaltap_0,将其更名为准备分析的文件名 myexam。

双击设置测试信号 Setup 空白处,弹出 Node Finder 对话框,在对话框中选择测试信号。这里选择观察 myexam 模块的 cnt。插入节点的过程与波形仿真选择信号完全相同。

3. Signal Tap II 参数设置

在信号设置 Signal Configuration 栏目下,完成对逻辑信号分析仪参数的设置,设置窗口如图 3-30 所示。

(1) 设置 Signal Tap II 工作时钟:单击图 3-30 所示 Clock 右侧的“...”按钮,在 Node Finder 对话框中,选择 clk 信号作为逻辑分析仪的采样时钟。

(2) 设置采样数据:采样数据深度设置为 1KB,根据待测信号的数量和 FPGA 芯片内部的存储器的大小决定采样数据深度。

(3) 触发设置:触发器流程控制、触发位置、触发条件均采用默认值。

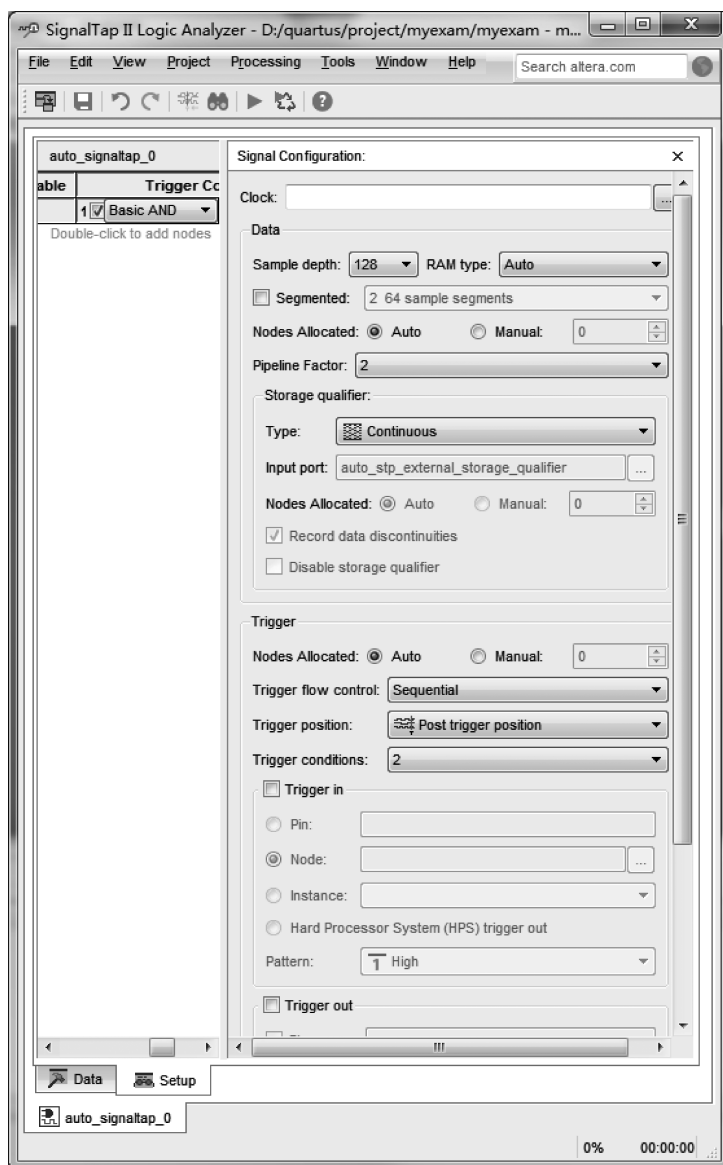


图 3-30 设置 Signal Tap II 参数

(4) 触发输入：首先选中触发输入 Trigger in,接着在触发源 Node 处选择 myexam 设计中的复位信号 reset,触发方式采用下降沿 Falling Edge。

(5) 保存文件：设置完成后,保存该文件 myexam.stp,保存时,系统出现提示信息：Do you want to enable Signal Tap II,单击 yes,表示同意使用 Signal Tap II,并准备将其与 myexam 文件捆绑在一起进行综合和适配,一同下载到 FPGA 芯片中。

也可以通过选择菜单 Assignments→Settings...,打开如图 3-31 所示的 Settings 对话框。在 Settings 对话框左侧的 Category 栏目下选择 Signal Tap II Logic Analyzer 项,选中 Enable Signal Tap II Logic Analyzer,添加 myexam.stp 文件,完成 Signal Tap II 与 myexam 源文件的捆绑。

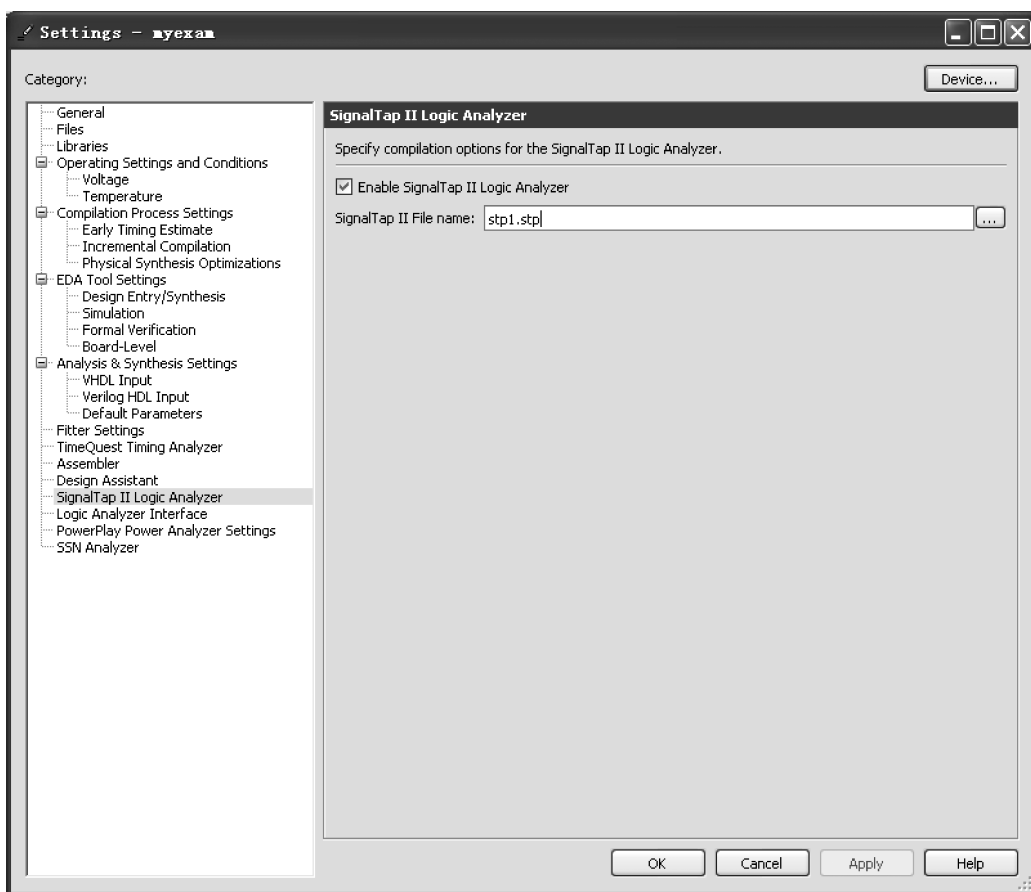


图 3-31 Signal Tap II 与 myexam 源文件的捆绑设置

4. 编译下载

(1) 编译：完成上述设置并保存文件后，必须要再次进行完整编译 Compilation。选择菜单 Processing→Start Compilation 或直接单击工具栏中编译按钮，执行编译操作，对设计文件进行检查。

(2) 连接硬件：在进行下载操作之前，首先将下载电缆的一端与 PC 机对应的端口相连，作者使用 USB Blaster 下载电缆，连接到 PC 机的 USB 端口，下载电缆的另一端与编程器件相连。

(3) 下载设置：如图 3-32 所示。Hardware 设置为 USB Blaster；连接硬件正常，系统会自动找到下载器件 Device 为 5CES；通过“...”按钮设置下载文件为 myexam.sof。

(4) 执行下载操作：单击编程按钮 ，开始对器件 5CSE 进行编程。

5. Signal Tap II 信号分析

如图 3-33 所示，在实例管理 Instance Manager 栏目下，选中 Instance 下面的文件 myexam，再单击 Autorun Analysis 启动分析按钮 ，启动 Signal Tap II 信号分析。只有当器件编程成功后，该分析按钮才会激活。

在 Setup/Data 栏目下，选择观察测试数据 Data 窗口。

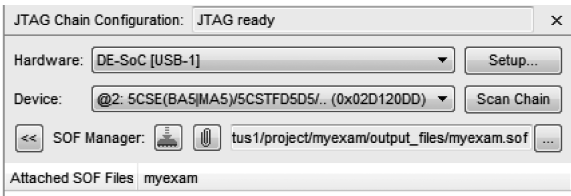


图 3-32 下载设置界面

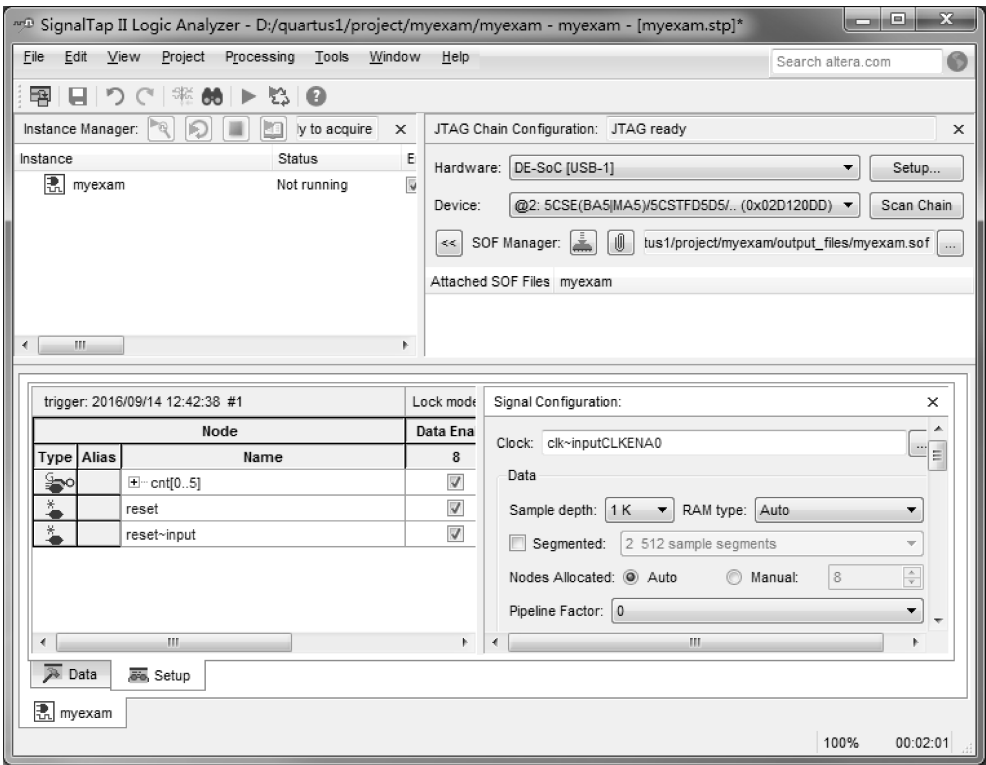


图 3-33 启动 Signal Tap II 信号分析

单击复位 reset 键,使 reset 信号发生一次从高电平到低电平的变化,为 Signal Tap II 逻辑分析仪提供采样触发信号。这时,在 Signal Tap II 的数据窗口就会观察到来自 FPGA 目标器件 5CES 芯片的实时信号,信号如图 3-34 所示。

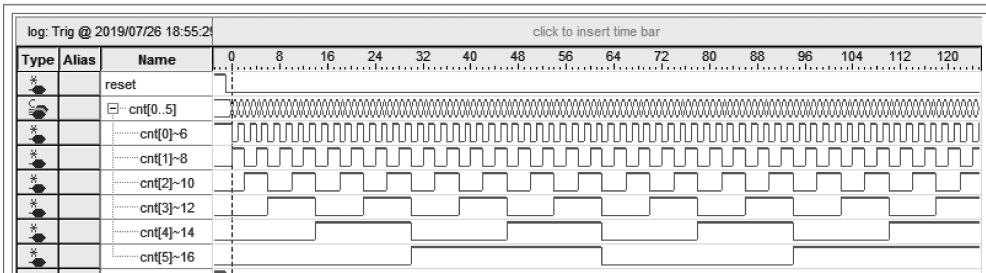


图 3-34 Signal Tap II 采样的信号波形

按下 Stop Analysis 停止分析按钮,结束分析过程。鼠标移动到分析波形处,单击鼠标右键和左键,将缩放波形的显示,使之适合观察。这里的观察可以看清输出信号 cnt 的变化规律,与设计的六进制计数器功能一致。

6. 撤销 Signal Tap II 信号分析

结束 Signal Tap II 逻辑分析后,应撤销 Signal Tap II 逻辑分析与 myexam 源文件的捆绑,释放出嵌入式逻辑分析仪对 FPGA 芯片资源的占用。

撤销 Signal Tap II 逻辑分析与 myexam 源文件捆绑的方法:选择菜单 Assignments→Settings...,在 Settings 对话框左侧的 Category 栏目下选择 Signal Tap II Logic Analyzer 项,撤销对 Enable Signal Tap II Logic Analyzer 的选中,单击 OK 按钮确认后,重新对 myexam 源文件进行完整编译,就可以撤销嵌入式逻辑分析仪对 FPGA 芯片资源的占用。

Quartus Prime 开发软件除了提供设计输入、设计处理、波形仿真等设计流程中必备的工具外,还集成了一些辅助设计工具,包括 I/O 分配验证工具、功率估计和分析工具、RTL 阅读器、SignalProbe(信号探针)及 Chip Editor(底层编辑器)、Timing Closure Floorplan(时序收敛平面布局规划器)。

在设计的任何阶段都可以使用 I/O 分配验证工具来验证引脚分配的合理性,保证在设计早期尽快确定引脚分配。功率估计工具可以对设计的功耗进行估算,以便于电源设计和热设计。RTL 视图则是用户在设计中查看设计代码的 RTL 结构的一种工具。SignalProbe 和 Signal Tap II 逻辑分析器都是调试工具,SignalProbe 可以在不影响设计中现有布局布线的情况下将内部电路中特定的信号迅速布线到输出引脚,从而无须对整个设计另做一次全编译。Chip Editor 能够查看编辑后布局布线的详细信息,且可以使用 Resource Property Editor(资源特性编辑器)对逻辑单元、I/O 单元或 PLL 的原始属性和参数执行编译后的重新编辑。Timing Closure Floorplan 可以通过控制设计的平面布局来达到时序目标。在综合以及布局布线期间可以对设计使用网表优化,同时使用 Timing Closure Floorplan 分析设计并执行面积约束,或者使用 LogicLock™ 区域分配进一步优化设计。

这些辅助设计工具本章不再一一介绍,如有需求的读者可参考相关书籍或 Quartus Prime 16.0 用户手册,学习更多的内容。