第 3 章

CHAPTER 3

Quartus Prime 的应用

Quartus Ⅱ是 Altera 公司的 EDA 开发环境，支持从建立工程、设计输入，到编译、综合与适配、仿真，以及编程与配置的全部设计流程。2015 年 Intel 公司收购 Altera 后，从 15.1 版开始，将 Quartus Ⅱ更名为 Quartus Prime。

Quartus Prime 支持硬件描述语言、原理图和状态机等多种输入方式，并且能够生成和识别电子设计交换格式(Electronic Design Interchange Format，EDIF)网表文件和 HDL 网表文件，同时支持第三方工具软件，设计者可以在设计流程的各个阶段根据需要选用更专业的工具软件。

Quartus 开发环境支持 Altera 公司的可编程逻辑器件，不同版本支持的器件种类和范围有所不同。新版软件在增加对新器件支持的同时，也逐步放弃了对旧器件的支持。例如，Quartus Ⅱ 13.0 支持 MAX Ⅱ/Ⅴ系列 CPLD 和 Cyclone Ⅰ～Ⅴ系列 FPGA，而 Quartus Ⅱ 13.1 放弃了对 MAX Ⅱ系列 CPLD 和 Cyclone Ⅰ/Ⅱ系列 FPGA 的支持。发展到 18.1 版后，Quartus Prime 只支持 Cyclone Ⅳ/Ⅴ及以上系列 FPGA。因此，设计者需要根据自己所用可编程逻辑器件的类型和型号选用合适的 Quartus 版本。例如，需要使用 MAX Ⅱ系列 CPLD，或者需要使用基于 Cyclone Ⅱ FPGA 设计的 DE1/DE2 开发板时，那么最高只能使用 Quartus Ⅱ 13.0 版；而使用基于 Cyclone Ⅲ FPGA 设计的 DE0 开发板时，则可以使用 Quartus Ⅱ 13.1 版；使用基于 Cyclone Ⅳ系列 FPGA 设计的 DE2-115 开发板时，就可以使用 Quartus Prime。

另外，不同版本的 Quartus 软件在功能上也有差异。从 10.0 版开始，Quartus Ⅱ放弃了 9.1 版(及以前)自带的、直观易用的向量波形仿真工具，而是应用功能更为强大的专业软件 ModelSim 进行仿真。从 11.0 版开始，Quartus Ⅱ推出了片上系统开发组件 SOPC Builder 的升级版本 Qsys；而从 13.0 版开始，则完全淘汰了 SOPC Builder，只保留 Qsys，同时又恢复了 9.1 版直观易用的向量波形仿真方法，并更名为 University Program VWF，以方便高校 EDA 课程教学使用。从 Quartus Prime 16.1 版开始，Intel 公司将 Qsys 更名为 Platform Designer，同时将内嵌的逻辑分析仪 SignalTap Ⅱ更名为 Signal Tap Logic Analyzer。

还需要注意的是，Quartus Ⅱ 13.0/13.1 版既支持 Windows 32 位操作系统，又支持 Windows 64 位操作系统，而 Quartus Ⅱ 14.0 及以上版本只支持 Windows 64 位操作系统。因此，如果需要在 Windows 32 位操作系统下应用 Quartus 软件，那么最高只能安装 Quartus II 13.0/13.1。关于 Quartus 软件各版本特性的详细信息，可以访问 Altera 公司的官方网站查阅 Quartus 软件的版本发布(Release)获取相关信息。

Quartus Prime 提供了 pro(专业版)、standard(标准版)和 lite(精简版)3 种版本。专业版只支持先进的 SoC FPGA 器件，标准版提供最全面的器件支持，而精简版是大批量器件系列理想的设计起点。使用专业版和标准版需要许可证支持，而精简版可以免费使用。

实践出真知。本章以 Quartus Prime 18.1 标准版为基础，以 DE2-115 开发板为实现载体，以 4 选 1 数据选择器为例讲述 Altera 公司 EDA 开发环境 Quartus Prime 的应用，具体包括基本设计流程、原理图设计方法、仿真分析方法和嵌入式逻辑分析仪的应用。最后，讲述直接法频率计的原理图设计方法。

3.1 基本设计流程

设计流程是指基于可编程逻辑器件应用 EDA 开发环境进行数字系统设计的具体过程。在 Quartus Prime 开发环境下进行 EDA 设计的基本流程如图 3-1 所示，包括建立工程、设计输入、编译、综合与适配、引脚锁定、编程与配置等主要步骤，同时还可以根据需要进行仿真分析、在线测试或时序分析。


第 20 集
微课视频


第 21 集
微课视频


第 22 集
微课视频

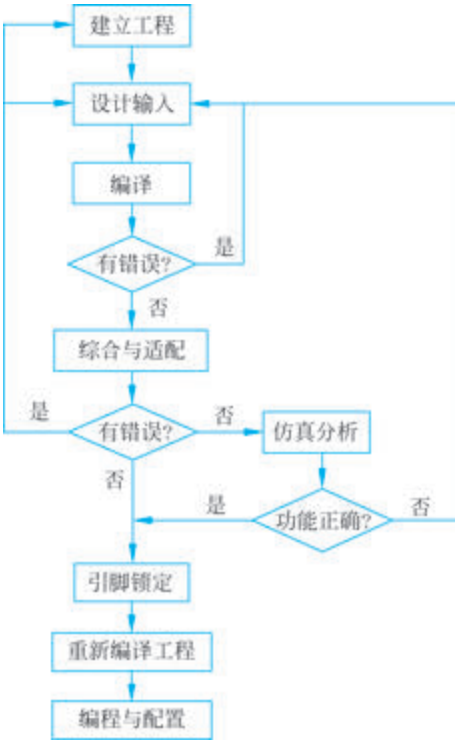


图 3-1 基本设计流程

本节讲述在 Quartus Prime 开发环境下进行 EDA 设计的基本步骤。

3.1.1 建立工程

Quartus Prime 以工程(Project)的方式对设计项目进行管理，将所有的文件存储在工程目录中，包括设计文件、波形文件、逻辑分析仪文件、存储器初始化文件和构成工程的编译器、仿真器和相关软件设置，以及编译、综合和适配过程中产生的相关文件和报告信息。因

此，在 Quartus Prime 开发环境下进行 EDA 设计，首先需要建立工程。

Quartus Prime 提供了新建工程向导(New Project Wizard)，通过新建工程向导引导设计者快速完成新工程的建立，包括指定工程目录、设置工程名和顶层模块文件名、添加或删除工程文件和库、指定目标器件类型和型号，以及设置 EDA 工具软件等工作。

建立工程的具体步骤如下。

(1) 启动 Quartus Prime。Quartus Prime 18.1 标准版启动后的主界面如图 3-2 所示，由左侧的项目管理器(Project Navigator)和任务(Tasks)指示窗口、底部的信息(Messages)窗口、中央的主窗口以及右侧的 IP 目录(IP Catalog)窗口 5 部分组成。除了主窗口外，其余窗口都可以通过 View→Utility Windows 菜单中的相关命令开启或关闭。

图 3-2 Quartus Prime 主界面

Quartus Prime 常用菜单命令如表 3-1 所示。除了 File、Edit 和 View 等 Windows 常用的菜单外，用于文件管理、工程设置、编译与综合以及内嵌工具的调用等命令分别在 Project、Assignments、Processing 和 Tools 菜单下。另外，Help 菜单中还提供了应用 Quartus Prime 和学习 HDL 的相关文档，帮助设计者迅速掌握软件和语言的应用要点。

表 3-1 Quartus Prime 常用菜单命令

菜　单	命　令	快 捷 键	功　能
File	New...	Ctrl＋N	新建文件
	Open...	Ctrl＋O	打开文件
	Close	Ctrl＋F4	关闭文件
	New Project Wizard...	—	新建工程向导
	Open Project...	Ctrl＋J	打开工程
	Save Project	—	保存工程
	Close Project	—	关闭工程
	Save	Ctrl＋S	保存文件
	Save as...	—	文件另存为
	Save All	Ctrl＋Shift＋S	保存所有文件

续表

菜　单	命　令	快　捷　键	功　能
File	File Properties...	—	文件属性
	Create/Update ▶	—	生成/更新
	Export...	—	输出
	Convert Programming Files...	—	转换编程文件
	Recent Files ▶	—	最近打开的文件
	Recent Projects ▶	—	最近打开的工程
	Exit	Alt+F4	退出
Edit	Undo	Ctrl+Z	撤销
	Redo	Ctrl+Y	重做
	Cut	Ctrl+X	剪切
	Copy	Ctrl+C	复制
	Paste	Ctrl+V	粘贴
	Delete	Del	删除
	Select All	Ctrl+A	全选
	Find...	Ctrl+F	查找
	Find Next	F3	查找下一个
	Find Previous	Shift+F3	查找上一个
	Replace...	Ctrl+H	替换
	Toggle Connection Dot	—	添加/删除连接点
	Flip Horizontal	—	水平翻转
	Flip Vertical	—	垂直翻转
	Update Symbol or Block...	—	更新符号或者模块图
View	Utility Windows ▶	—	添加/关闭应用窗口
	Address Radix	—	地址格式
	Memory Radix	—	存储格式
	Zoom In	Ctrl+Space	放大
	Zoom Out	Ctrl+Shift+Space	缩小
	Show Guidelines	—	显示/隐藏网格线
Project	Add Current File to Project	—	添加当前文件到工程中
	Add/Remove Files in Project	—	添加/删除工程文件
Assignments	Device...	—	器件
	Settings...	Ctrl+Shift+E	设置
	Pin planner	Ctrl+Shift+N	引脚锁定工具
Processing	Stop Processing	Ctrl+Shift+C	停止编译
	Start Compilation	Ctrl+L	开始编译
	Start ▶	—	开始
Tools	Run Simulation Tool ▶	—	运行仿真工具
	Netlist Viewers ▶	—	网表浏览器
	Signal Tap Logic Analyzer	—	逻辑分析仪
	Programmer	—	编程器
	IP Catalog	—	IP 目录
	Nios Ⅱ SBT for Eclipse	—	片上系统软件开发工具

续表

菜　单	命　令	快 捷 键	功　能
Tools	Platform Designer	—	片上系统硬件开发平台
	Tcl Scripts...	—	Tcl 脚本命令
	Options...	—	选择
	License Setup...	—	许可文件设置
	Install Devices...	—	安装器件库
Help	Help Topics	—	帮助主题
	PDF Tutorials ▶	—	HDL 学习文档
	What's New	—	新版本信息
	Release Notes	—	版本发布信息
	On the Web ▶	—	网络资源
	About Quartus Prime	—	关于 Quartus Prime

注：①▶表示命令有子菜单；...表示命令有弹出信息页；②不同文档类型的菜单项不同；③SBT 为 Software Build Tools 的缩写。

在 Quartus Prime 主界面中，执行 File→New Project Wizard 菜单命令，弹出如图 3-3 所示的 New Project Wizard 对话框，提示设计者建立新工程需要完成以下 5 项任务：①指定工程名和工程目录；②指定顶层模块文件名；③添加或删除工程文件和库；④指定目标器件类型和型号；⑤设置 EDA 工具。

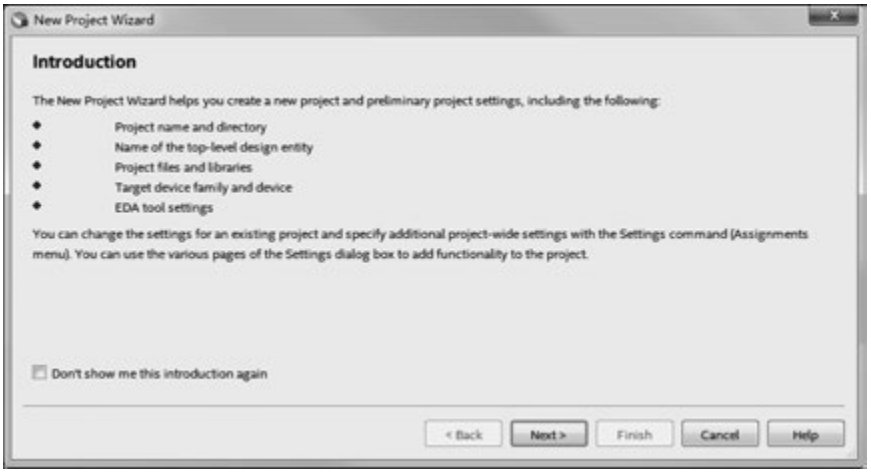


图 3-3　New Project Wizard 对话框

单击 Next 按钮进入设置工程目录、工程名和顶层模块名页面，如图 3-4 所示，其中第 1 栏用于指定存放工程文件的工作目录，第 2 栏用于指定工程名，第 3 栏用于指定顶层模块文件名。Quartus Prime 要求顶层模块文件名与工程名相同，因此在输入工程名时，顶层模块文件名也随之自动输入。

需要强调的是：①工程路径中不能含中文、空格和特殊字符，因为 Quartus 软件无法正确地识别汉字和特殊字符编码；②当前工程项目中包含低层工程项目时，当前工程项目文件和所有低层项目文件应保存在同一目录中，不要分子目录保存。

(2) 单击 Next 按钮进入工程类型(Project Type)页面，选择工程类型。对于初学者，建

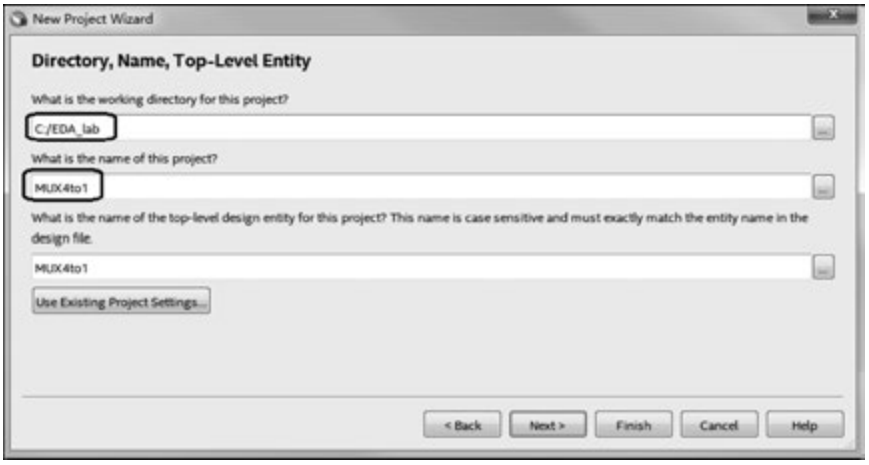


图 3-4　设置工程目录和工程名

议先选择空工程(Empty project)，如图 3-5 所示，以便熟悉在 Quartus Prime 开发环境下进行 EDA 设计的基本流程。对于已经熟悉基本设计流程的用户，则可以选择工程模板(Project template)以提高设计效率。

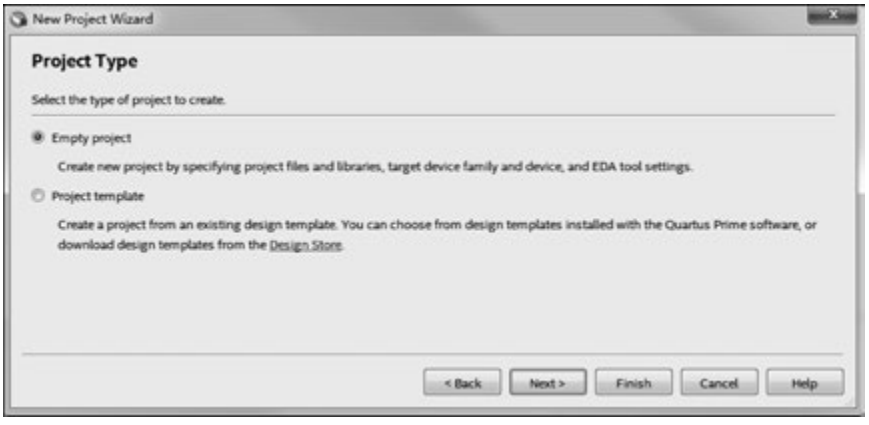


图 3-5　选择工程类型

(3) 单击 Next 按钮进入添加文件(Add Files)页面，如图 3-6 所示，向工程中添加所需要的文件和库，也可以删除工程中多余的文件和库。当新建的工程中不包含任何设计文件时，可以跳过这一步。

另外，工程建立完成后，还可以通过 Project→Add/Remove Files in Project 菜单命令再次进入 Add Files 页面，向工程中添加新文件或删除工程中的多余文件。

(4) 单击 Next 按钮进入目标器件设置页面，如图 3-7 所示。首先，在 Family 下拉列表中选择目标器件的类型，然后在 Available devices 列表中选择具体的器件型号。在选择目标器件型号时，还可以通过指定器件的封装(Package)、引脚数(Pin count)和核心速度等级(Core speed grade)缩小选择范围。

图 3-7 中已经选择了 DE2-115 开发板的 FPGA 器件型号 EP4CE115F29C7，器件类型为 Cyclone IV E，封装为 FBGA，引脚数量为 780 个，速度等级为 7。

(5) 单击 Next 按钮进入 EDA 工具设置页面，如图 3-8 所示，用于选择综合工具、仿真

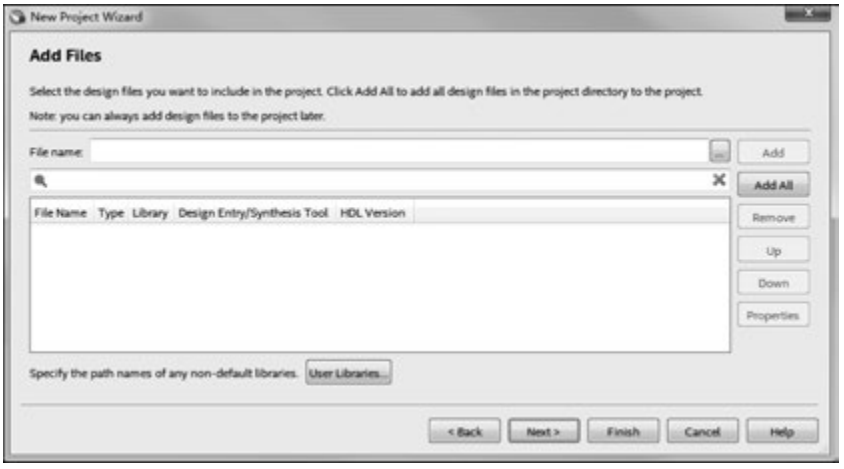


图 3-6 添加文件

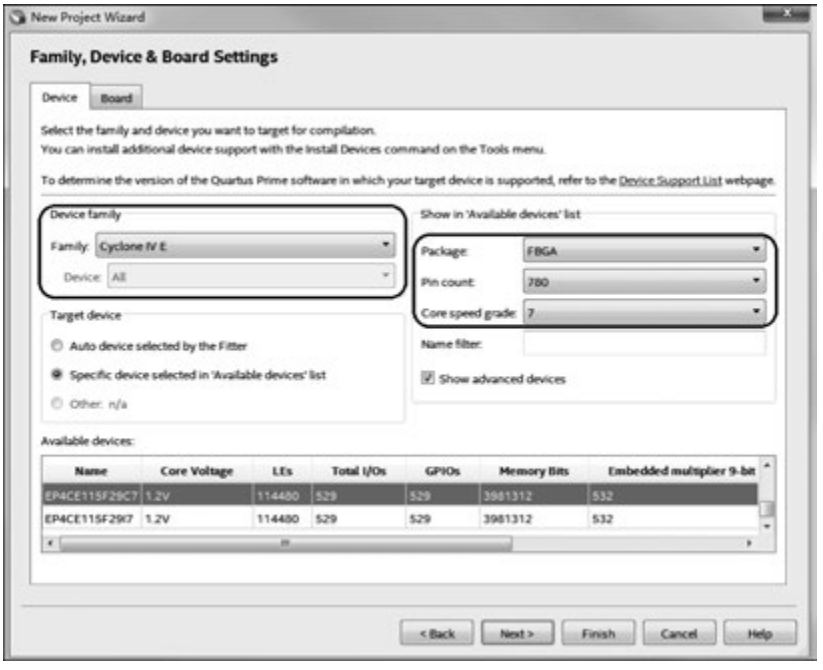


图 3-7 目标器件设置

工具和时序分析工具。不选择(None)时,Quartus Prime 应用默认的 EDA 工具。

对于复杂的数字系统设计,建议选用专业的 EDA 工具。对于一般的设计项目,Quartus Prime 提供的默认综合工具完全能够满足工程要求。

Quartus Prime 18.1 版应用 ModelSim 进行仿真。Intel 公司 OEM 版的 ModelSim 名为 ModelSim-Altera,仿真语言根据所用硬件描述语言的类型选定(图 3-8 中已选择 Verilog HDL)。

(6) 单击 Next 按钮,弹出工程汇总信息对话框,如图 3-9 所示,显示新建工程的相关信息。

若建立工程过程中的信息有误,还可以单击图 3-9 中的 Back 按钮返回相应的页面进行

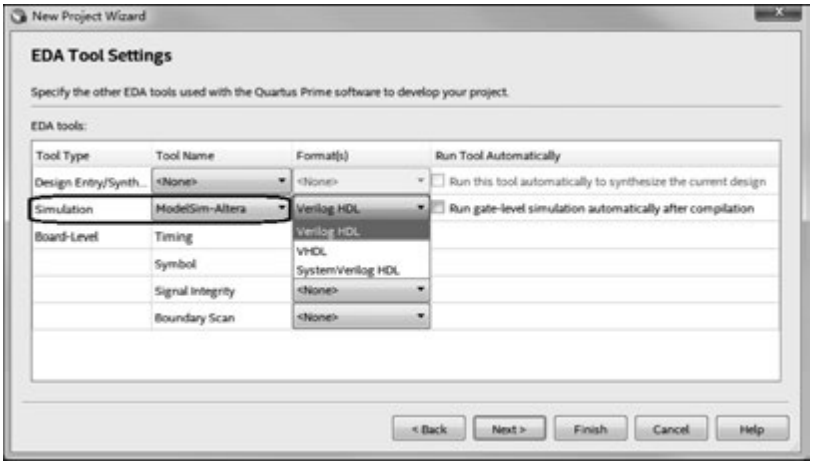


图 3-8　EDA 工具设置

图 3-9　工程汇总信息对话框

修改，无误则单击 Finish 按钮完成新工程的建立任务。这时 Quartus Prime 主界面最上方的标题栏已经包含了当前工程目录和工程名两项信息，如图 3-10 所示，同时在左侧的项目管理器中包含了目标器件和顶层模块名两项信息。

3.1.2　设计输入

设计输入(Design Entry)是将所设计的电路或系统以 EDA 软件支持的某种形式表达出来的过程。Quartus Prime 支持硬件描述语言(HDL)、原理图(Schematic)和状态机(State Machine)等多种文件类型，如表 3-2 所示，同时又提供了第三方设计输入接口，如 EDIF 网表文件等。

图 3-10 包含新建工程信息的 Quartus Prime 主界面

表 3-2 Quartus Prime 设计文件类型

设 计 文 件	扩展名	描 述
Verilog HDL	.v,.vlg,.verilog	应用 Verilog HDL 编写的设计文件
VHDL	.vhd,.vh,.vhdl	应用 VHDL 编写的设计文件
原理图	.bdf	应用 Quartus Prime Block Editor 建立的原理图设计文件
EDIF	.edf,.edif	应用任何标准 EDIF 编写程序生成的网表文件
图形	.gdf	应用 MAX+plus Ⅱ Graphic Editor 建立的原理图设计文件
AHDL	.tdf	应用原 Altera 公司硬件描述语言 AHDL 编写的设计文件
VQM	.vqm	通过 Synplify 或 Quartus Prime 生成的 Verilog 网表文件

硬件描述语言是 EDA 设计中最主要的输入方式。与传统的原理图方法相比,应用 HDL 有利于应用自顶向下的设计方式,有利于模块的分解与复用,并且 HDL 描述还具有与实现器件工艺无关的特点。因此,基于可移植方面的考虑,EDA 工程设计大多应用 HDL 进行描述,以方便模块的功能重构与复用。

应用硬件描述语言输入方式的基本步骤如下。

(1) 在 Quartus Prime 主界面中,执行 File→New 菜单命令,弹出如图 3-11 所示的 New 对话框,用于新建项目文件。

Quartus Prime 支持 4 类项目文件:设计文件(Design Files)、存储器文件(Memory Files)、验证与调试文件(Verification/Debugging Files)和其他文件(Other Files)。每种项目文件又包含多种具体的文件类型选项。

若应用 Verilog HDL 进行设计,则应选择 Design Files 下的 Verilog HDL File,然后单击 OK 按钮,弹出如图 3-12 所示的 HDL 代码编辑窗口。

(2) 在 HDL 代码编辑窗口中,输入 Verilog HDL 代

图 3-11 New 对话框

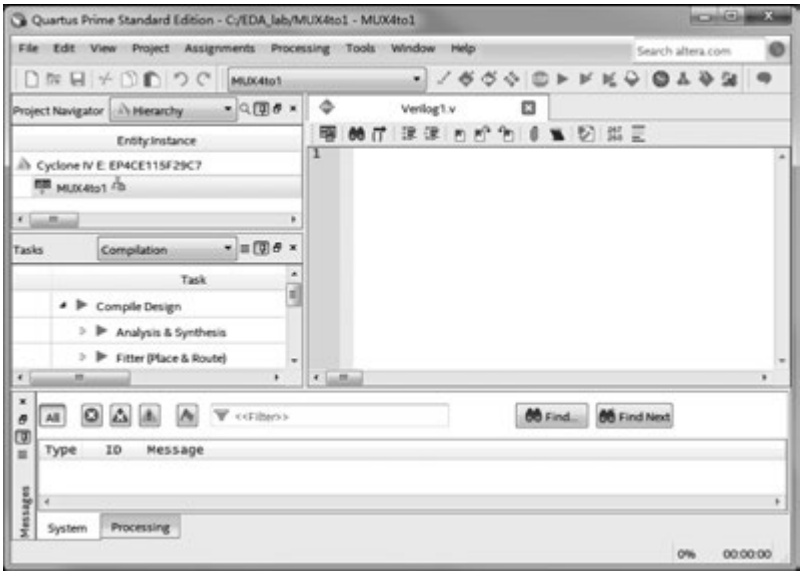


图 3-12　HDL 代码编辑窗口

码并以模块名(本例为 MUX4to1)为文件名进行保存(提示：Verilog HDL 文件的扩展名为. v,VHDL 文件的扩展名为. vhd),如图 3-13 所示。

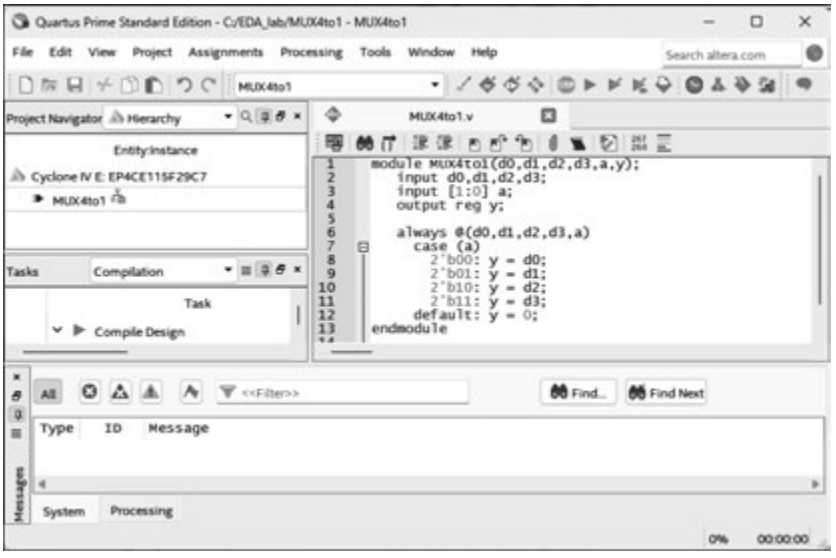


图 3-13　设计输入

需要注意的是：①Quartus Prime 要求顶层模块与工程同名,否则会导致编译失败。当工程只包含一个模块时,该模块即为顶层模块,因此一定要确保模块名、文件名与工程名严格一致；②对于顶层模块,新建工程和设计输入两个步骤可以互换。也就是说,可以先输入设计文件,保存设计文件时,在弹出的如图 3-14 所示的对话框中单击 Yes 按钮,启动新建工程的过程。

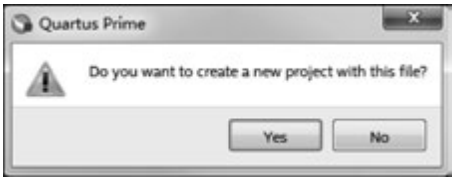


图 3-14　由文件建立工程

设计输入完成后，Verilog HDL 代码需要经过编译、综合与适配后才能生成可以下载到 CPLD 中的编程文件或 FPGA 中的配置文件。

3.1.3 编译、综合与适配

Quartus Prime 编译、综合与适配流程如图 3-15 所示，包括分析与建模、综合、适配、装配和时序分析等主要环节。

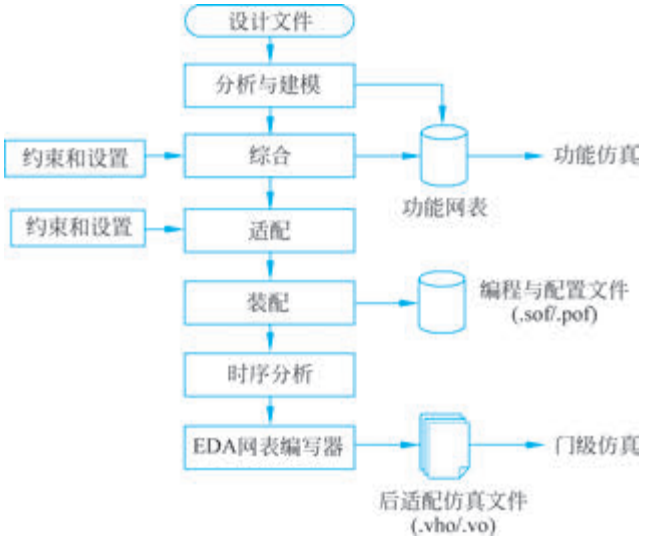


图 3-15 Quartus Prime 编译、综合与适配流程

分析与建模(Analysis and Elaboration)是调用 Quartus Prime 内嵌的编译器检查输入代码中是否存在语法错误以及潜在的逻辑错误的过程。

综合(Synthesis)是面向给定的约束和设置(Constraints and Settings)，将 HDL 代码转换为门级网表的过程，是将 HDL 描述转换为硬件电路的关键步骤。

综合的效果决定设计电路的性能和芯片的资源利用率。在综合前，需要对设计施加适当的约束，包括时序约束、面积约束和功耗约束。综合器根据设定的约束，在器件厂商提供的综合库支持下，针对具体的可编程逻辑器件类型进行优化，目的是使综合所生成的电路满足设计要求。

给设计添加适当的约束，编译后对综合结果进行分析，找出设计中的瓶颈，对设计后期优化过程的成败起着决定性的作用。但是，给设计添加约束时，设计者必须充分理解分析与综合设置以及适配设置等优化项目的含义。关于综合与优化设置相关内容，将在第 7 章中讲述。未添加约束文件时，Quartus Prime 按照默认的设置进行编译与综合。

编译与综合过程完成之后，Quartus Prime 会自动将综合后产生的网表文件针对选定的目标器件进行映射，将工程的逻辑和时序要求与目标器件的可用资源相匹配，包括逻辑分割、逻辑优化和布局布线，这一过程称为适配和装配(Fitter and Assembler)。完成后，Quartus Prime 会生成针对具体目标器件的编程与配置文件(Programming and Configuration Files)，并进行时序分析(Timing Analysis)，生成后适配仿真文件(Post-Fit Simulation Files)和工程统计报告，说明目标器件的资源占用等情况。

在 Quartus Prime 主界面中，执行 Processing→Start Complication 菜单命令或直接单击主界面中的 ▶ 按钮启动编译与综合过程。若描述代码没有错误，则编译、综合与适配过程会自动完成，生成能够下载到 CPLD/FPGA 或其外部配置芯片的编程与配置文件，并显示如图 3-16 所示的工程汇总信息，说明该工程的编译时间、版本信息、器件信息和资源占用等情况。

图 3-16　工程汇总信息

若在编译过程中发现错误，Quartus Prime 将在信息窗口中用红色字体显示错误信息(Error Message)。对于代码问题，双击错误信息提示，Quartus Prime 会自动定位到错误代码附近。仔细阅读错误信息说明，修改代码并重新启动编译与综合过程，直到完全正确为止。

需要注意的是，编译与综合过程中出现蓝色字体的警告信息(Warning Message)，虽然不影响编译、综合与适配过程的正常进行，但可能预示着描述代码中存在潜在的逻辑错误，因此，设计者应对编译与综合过程中输出的警告信息足够重视，建议仔细阅读相关信息，排除潜在的逻辑错误，确保综合出正确的功能电路。

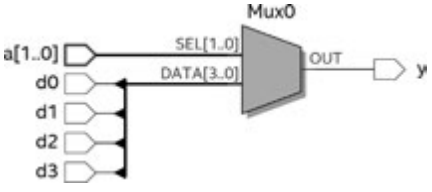


图 3-17　RTL 电路

编译、综合与适配过程完成之后，执行 Tools→Netlist Viewers 菜单命令，就可以查看综合出的 RTL 电路，如图 3-17 所示。可以看出，4 选 1 数据选择器是通过定制 Quartus Prime 中的通用数据选择器 IP 得到的。

3.1.4　引脚锁定

编译、综合与适配过程完成后，如果需要将综合出的电路下载到可编程逻辑器件中进行测试，那么还需要进行引脚锁定(或称为引脚约束)，以确定工程的顶层模块端口与目标器件引脚的对应关系。

Quartus Prime 提供了 3 种引脚锁定方法：①使用图形化工具 Pin Planner 锁定引脚；②编写 Tcl 文件锁定引脚；③应用属性定义锁定引脚。

1. 使用图形化工具 Pin Planner 锁定引脚

使用图形化工具 Pin Planner 是引脚锁定的基本方法，具体步骤如下。

(1) 在 Quartus Prime 主界面中执行 Assignments→Pin Planner 菜单命令，弹出如图 3-18 所示的 Pin Planner 窗口。需要在图 3-18 中的 Location 栏中输入模块 I/O 端口对应的引脚信息。

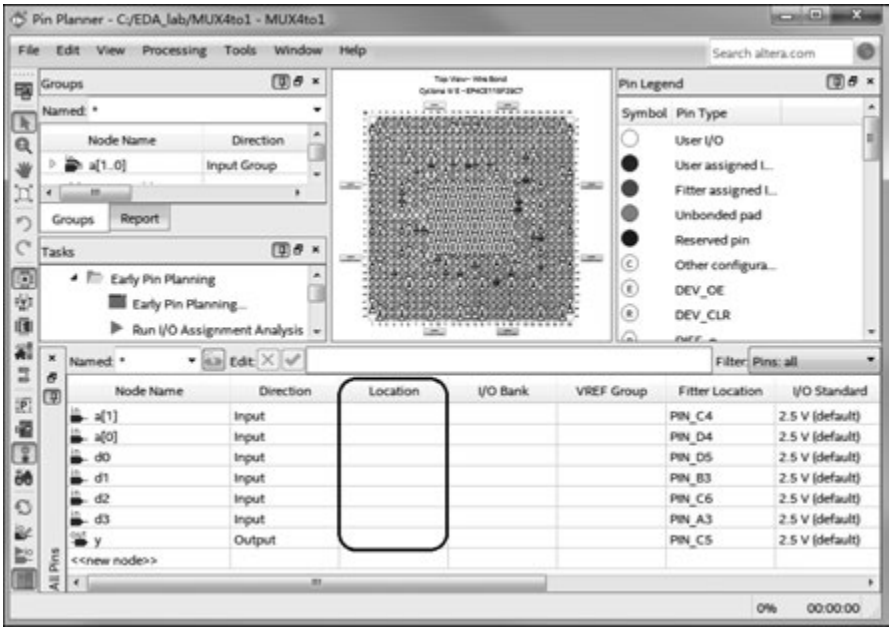


图 3-18 Pin Planner 窗口(1)

(2) 对于 DE2-115 开发板，需要查阅用户手册(User Manual)以确定 FPGA 外围电路组件与其 I/O 引脚的对应关系。DE2-115 开关量输入电路与 FPGA 器件 I/O 引脚的接口信息如图 3-19 所示，发光二极管(Light Emitting Diode，LED)驱动电路与 FPGA 器件 I/O 引脚的接口信息如图 3-20 所示。

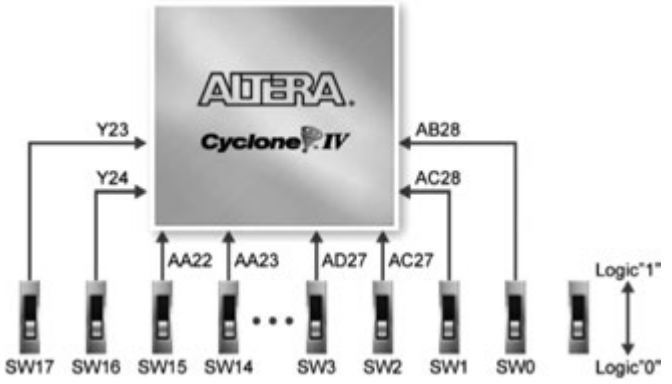


图 3-19 DE2-115 开发板开关量输入电路与 FPGA 器件 I/O 引脚的连接信息

若将 4 选 1 数据选择器的 4 路输入数据 d3、d2、d1 和 d0 分别锁定到开关 SW17～SW14 上，将 2 位地址 a[1]和 a[0]分别锁定到开关 SW1 和 SW0 上，将输出 y 锁定到发光二

图 3-20　DE2-115 开发板 LED 驱动电路与 FPGA 器件 I/O 引脚的接口信息

极管 LEDG8 上，则需要在 Pin Planner 窗口下部的 Location 栏中填入对应的 FPGA 引脚名，如图 3-21 所示。

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
a[1]	Input	PIN_AC28	5	B5_N2	PIN_C4	2.5 V (default)
a[0]	Input	PIN_AB28	5	B5_N1	PIN_D4	2.5 V (default)
d0	Input	PIN_AA23	5	B5_N2	PIN_D5	2.5 V (default)
d1	Input	PIN_AA22	5	B5_N2	PIN_B3	2.5 V (default)
d2	Input	PIN_Y24	5	B5_N2	PIN_C6	2.5 V (default)
d3	Input	PIN_Y23	5	B5_N2	PIN_A3	2.5 V (default)
y	Output	PIN_F17	7	B7_N2	PIN_C5	2.5 V (default)
<<new node>>						

图 3-21　Pin Planner 窗口(2)

(3) 引脚锁定完成后，需要重新编译工程，以生成带有引脚锁定信息的编程与配置文件。

2. 编写 Tcl 文件锁定引脚

对于顶层模块端口很多的应用工程，使用 Pin Planner 进行引脚锁定的方法费时费力，建议编写 Tcl 文件完成引脚锁定。

Tcl 是工具命令语言(Tool Command Language)，在 FPGA 开发中可以应用 Tcl 文件对引脚进行配置，既可以提高设计效率，又能够增强代码的可复用性。

Tcl 文件的格式定义如下。

```
# setup.tcl
# setup pin setting
set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-STATED"
set_global_assignment -name ENABLE_INIT_DONE_OUTPUT OFF
…
set_location_assignment PIN_引脚名 -to 端口名
…
```

其中，# setup.tcl 和 # setup pin setting 为 Tcl 文件的说明部分；setup 为 Tcl 文件名，建议修改为当前工程名。

编写 Tcl 文件进行引脚锁定的具体方法如下。

(1) 在 Quartus Prime 主界面中执行 Files→New 菜单命令,弹出 New 对话框(见图 3-11),选择 Design Files 栏下的 Tcl Script File,单击 OK 按钮进入 Tcl 文件编辑窗口。在窗口中输入 4 选 1 数据选择器 MUX4to1 的 Tcl 锁定信息,如图 3-22 所示。

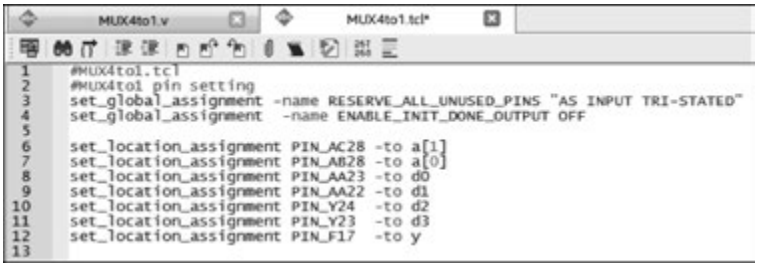


图 3-22 Tcl 文件编辑窗口

(2) Tcl 文件编辑好后,保存文件名为 MUX4to1. tcl,并执行 Projects→Add/Remove Files in Project 菜单命令将 Tcl 文件添加到工程中。

(3) 执行 Tools→Tcl Scripts 菜单命令,弹出如图 3-23 所示的 Tcl Scripts 对话框,打开相应的 Tcl 文件,单击 Run 按钮运行 Tcl 文件,即可一次完成引脚锁定。

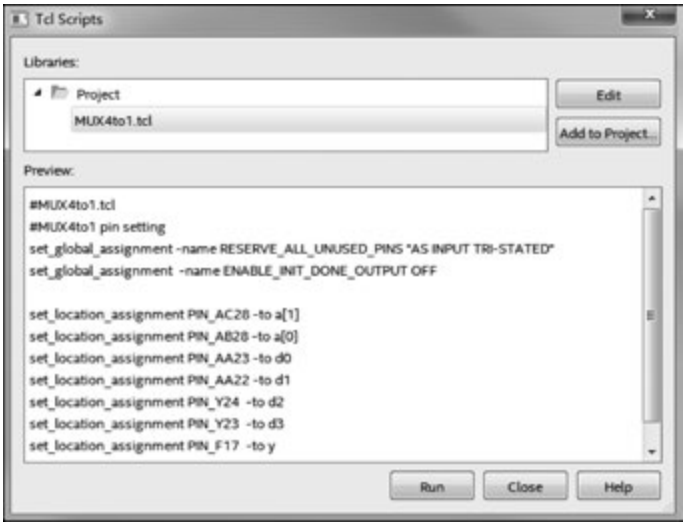


图 3-23 Tcl Scripts 对话框

通常,EDA 开发板配有示例工程模板,包含开发板所有引脚分配信息。例如,友晶公司为 DE2-115 开发板提供了一个名为 DE2_115_GOLDEN_TOP 的空工程模板,模板中已经包含了开发板所有的引脚锁定信息。因此,设计者可以基于 GOLDEN_TOP 工程进行设计,添加自己的功能代码,保持模块的端口名与模板已经锁定引脚的端口名一致,或者将模板中已经定义的端口名修改为自己定义的端口名,从而省去引脚锁定这个耗时费力的环节,有效地提高设计效率。

3. 应用属性定义锁定引脚

除了应用 Pin Planner 和编写 Tcl 文件锁定引脚的方法外,还可以应用 Quartus Prime 综合属性(Synthesis Attributes)语句中的芯片引脚属性(chip_pin)锁定引脚。

在 Verilog HDL 代码中添加芯片引脚属性不影响 Verilog HDL 的语法,但在布局布线

(Place & Route)过程中，Quartus Prime 会根据 chip_pin 属性锁定引脚。

应用属性定义实现 4 选 1 数据选择器引脚锁定的代码如下。

```
module MUX4to1(d0,d1,d2,d3,a,y);
  input d0                              /* synthesis chip_pin = "AA23" */;
  input d1                              /* synthesis chip_pin = "AA22" */;
  input d2                              /* synthesis chip_pin = "Y24" */;
  input d3                              /* synthesis chip_pin = "Y23" */;
  input [1:0] a                         /* synthesis chip_pin = "AC28,AB28" */;
  output wire/reg y                     /* synthesis chip_pin = "F17" */;
  ...
endmodule
```

需要说明的是：①chip_pin 属性应放在端口列表和表示语句结束的分号之间，以“/*”开始，以“*/”结束；②应用属性定义锁定引脚时，必须在新建工程中明确指出实现目标器件的具体型号；③应用属性定义锁定引脚的方法只能在顶层模块中应用。

3.1.5 编程与配置

编程与配置是指将经过编译与综合后产生的含有电路设计信息的目标文件(.sof 或 .pof)写入可编程逻辑器件(CPLD/FPGA)或 FPGA 外部配置器件(如 EPCS)中的过程。一般习惯将.sof 文件加载到 FPGA 内部查找表 SRAM 中的过程称为配置(Configuration)；将.pof 文件写入 CPLD 或固化到 FPGA 外部配置器件中的过程称为编程(Program)。

对 DE2-115 开发板进行编程与配置时，需要将装有 Quartus Prime 的计算机通过 USB 设备电缆(Type A to B)与 DE2-115 开发板的 USB-Blaster 接口连接起来并给开发板加电。

如果首次使用开发板，那么还需要安装 USB-Blaster 驱动程序。

USB-Blaster 的安装步骤如下。

图 3-24 未安装驱动的 USB-Blaster 设备图标

① 通过 Windows 控制面板中的“设备管理器”打开设备管理界面。在“其他设备”中找到带有感叹号的 USB-Blaster 设备图标，如图 3-24 所示。

② 右击带有感叹号的 USB-Blaster 图标，在弹出的快捷菜单中选择“更新驱动程序软件(P)”。

③ 在弹出的对话框中选择“浏览计算机以查找驱动程序软件(R)”。

④ 浏览并指定 USB-Blaster 驱动程序路径。

对于 Quartus Prime 18.1 标准版，USB-Blaster 驱动程序位于 Quartus Prime 安装目录(默认为 c:\intelFPGA\18.1)下的 quartus\drivers\usb-blaster 子目录中。浏览并选中驱动程序所在的子目录后，单击“下一步”按钮进行安装。安装完成后，在设备管理器的“通用串行总线控制器”下就可以看到正常的 USB-Blaster 设备图标了，如图 3-25 所示。

Quartus Prime 支持 4 种编程与配置方式：JTAG(Joint Test Action Group)配置方式、Socket 编程(In-Socket Programming)方式、PS(Passive Serial)方式和 AS 编程(Active Serial Programming)方式。常用的有 JTAG 配置和 AS 编程两种方式，其中应用 JTAG 配

图 3-25 安装好驱动的 USB-Blaster 设备图标

置方式可对多个 FPGA 进行配置，应用 AS 编程方式可对 CPLD/EPCS 配置器件进行编程。另外，Quartus Prime 还支持 JTAG 间接配置方式，用于对 EPCS 配置器件进行间接编程。

1. JTAG 配置方式

JTAG 配置方式用于将综合与适配后生成的目标文件. sof(SRAM Object File)配置到 FPGA 查找表的 SRAM 中，以实现可编程逻辑。由于 SRAM 为易失性存储器(Volatile Memory)，断电后存储数据会丢失，所以应用 JTAG 配置方式直接配置 FPGA 不能长期保存电路的设计信息，适用于实验或产品研发阶段硬件电路的测试。

DE2-115 开发板的 JTAG 配置链如图 3-26 所示。应用 JTAG 对 FPGA 进行配置时，需要确保运行与编程(RUN/PROG)开关 SW19 处于 RUN 位置。

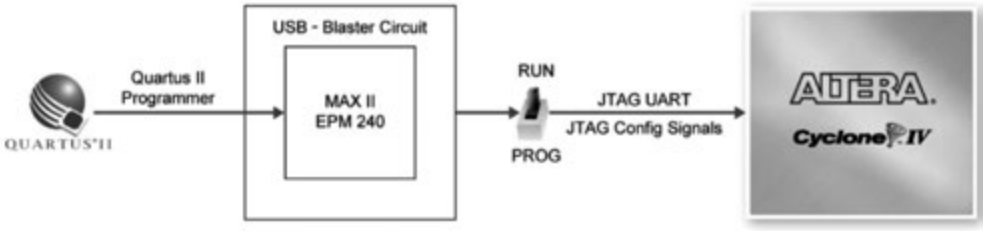


图 3-26 JTAG 配置链

应用 JTAG 方式配置 FPGA 的具体步骤如下。

(1) 打开编程器。

在 Quartus Prime 主界面中执行 Tools→Programmer 菜单命令，打开 Quartus Prime 编程器(Programmer)。

Quartus Prime 编程器窗口如图 3-27 所示。

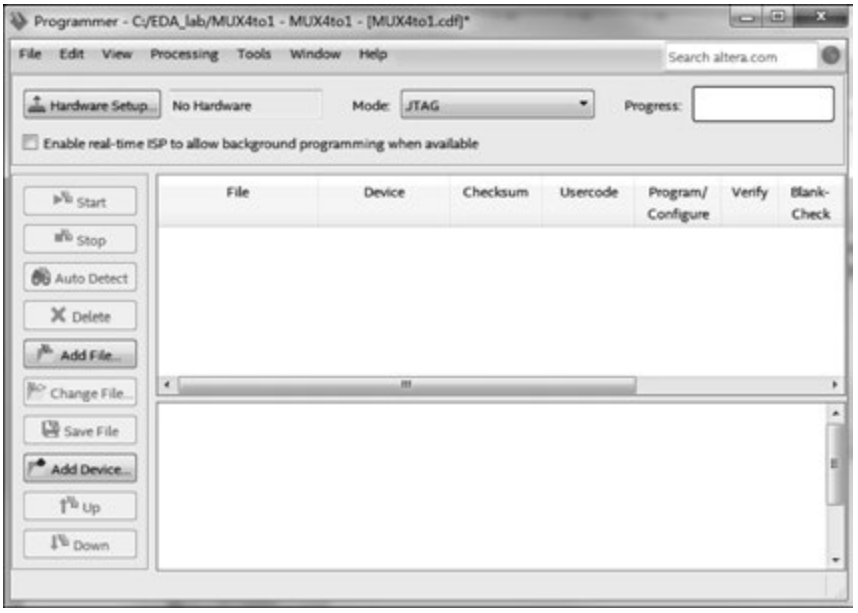


图 3-27 Quartus Prime 编程器窗口

(2) 设置硬件连接。

单击 Hardware Setup 按钮，弹出 Hardware Setup 对话框，在 Currently selected hardware 下拉列表中选择 USB-Blaster[USB-0]，如图 3-28 所示，设置硬件连接。完成后单击 Close 按钮关闭对话框。

图 3-28 设置硬件连接对话框

如果不能正确连接，则需要检查：①USB-Blaster 驱动程序是否已正确安装；②开发板是否已与计算机相连接；③开发板是否已经加电。

(3) 选择编程与配置方式。

将设计电路下载到 FPGA 目标器件中，需要应用 JTAG 配置方式。Quartus Prime 默认的编程与配置方式即为 JTAG。

(4) 添加配置文件。

在工程编译、综合与适配过程完成之后，Quartus Prime 会生成. sof 配置文件，并且将

文件自动添加到编程器窗口中。如果没有添加.sof文件或需要更换配置文件，单击编程窗口左侧的Add File按钮打开选择编程文件窗口，浏览并选择工程目录中output_files子目录下的.sof配置文件，如图3-29所示，单击Open按钮打开。

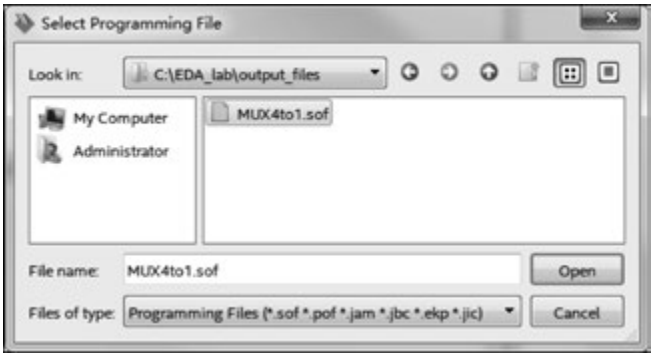


图3-29 选择.sof配置文件

(5) 启动配置过程。

在Quartus Prime编程器中勾选Program/Configure选项，如图3-30所示，并确认DE2-115开发板的运行与编程开关SW19处于RUN位置，然后单击Start按钮开始配置。当配置进度条(Progress)显示100%时，配置完成。

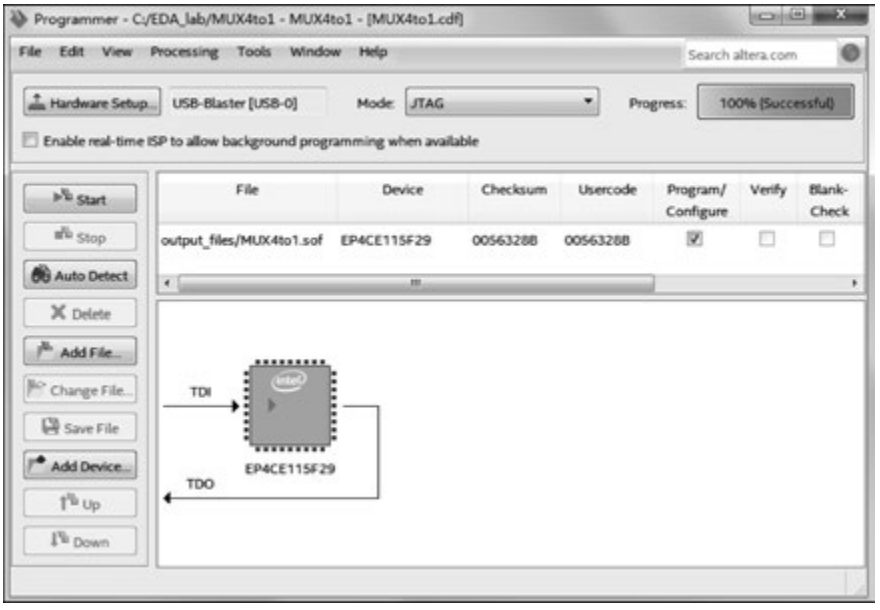


图3-30 配置过程窗口

配置完成后，就可以在开发板上测试4选1数据选择器的功能了。

2. AS编程方式

AS编程方式用于将综合与适配后生成的目标文件.pof(Program Object File)写入CPLD或固化到FPGA外部的EPCS配置器件中。由于CPLD和EPCS配置器件均为非易失性存储器(Non-volatile Memory)，所以应用AS编程方式能够永久保存电路的设计信息。

将编程文件固化到FPGA外部的EPCS配置器件后，每次开发板加电时，FPGA会主

动引导配置过程：读取 EPCS 配置器件中的数据，并加载到 FPGA 查找表的 SRAM 中。因此，应用 AS 编程方式将编程文件.pof 写入 EPCS 配置器件中能够使 FPGA 在上电后立即获得硬件电路的设计信息。

DE2-115 开发板的 AS 编程链如图 3-31 所示。应用 AS 方式编程 EPCS 器件时，需要将运行与编程(RUN/PROG)开关 SW19 拨到 PROG 位置。

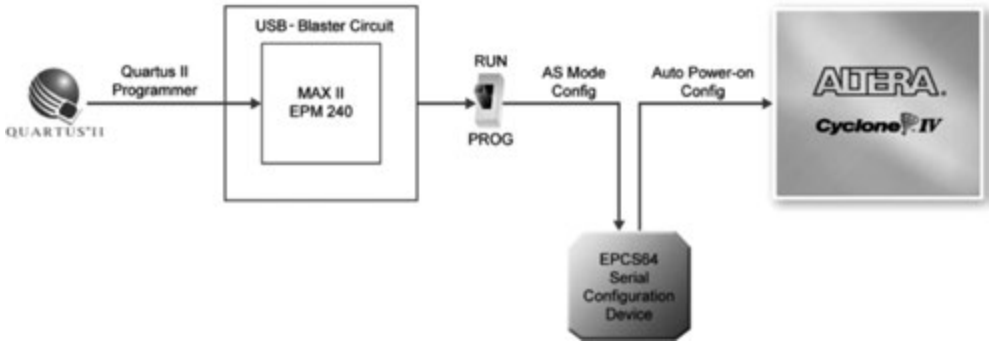


图 3-31 AS 编程链

需要注意的是，如果在建立工程的过程中没有指定 EPCS 配置器件，则综合与适配过程不会自动生成.pof 编程文件。因此，在应用 AS 编程方式之前，需要设置目标器件以指定 EPCS 配置器件的具体型号，然后重新启动编译、综合与适配过程才能生成.pof 编程文件。

应用 AS 方式编程 EPCS 器件的具体步骤如下。

(1) 在 Quartus Prime 主界面中执行 Assignments→Device 菜单命令，弹出 Device 对话框，如图 3-32 所示。

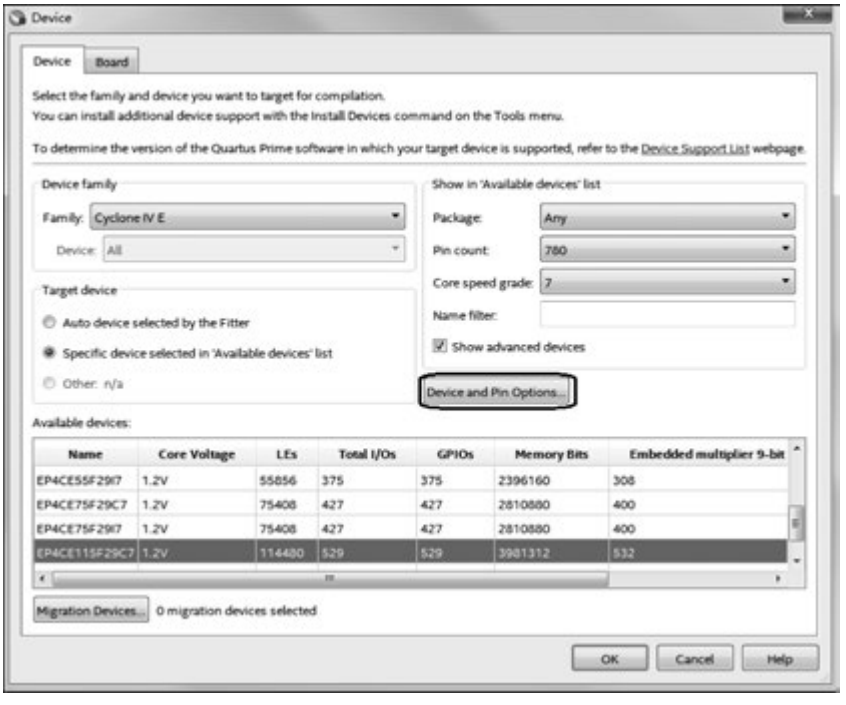


图 3-32 Device 对话框

(2) 单击图 3-32 中的 Device and Pin Options 按钮,在弹出的 Device and Pin Options (设备与引脚选择)对话框中选择 Category 栏下的 Configuration 选项,并勾选右侧的 Use configuration device 复选框,然后选择具体的 EPCS 配置器件型号(DE2-115 开发板的配置器件型号为 EPCS64),如图 3-33 所示,然后单击 OK 按钮完成设置过程。

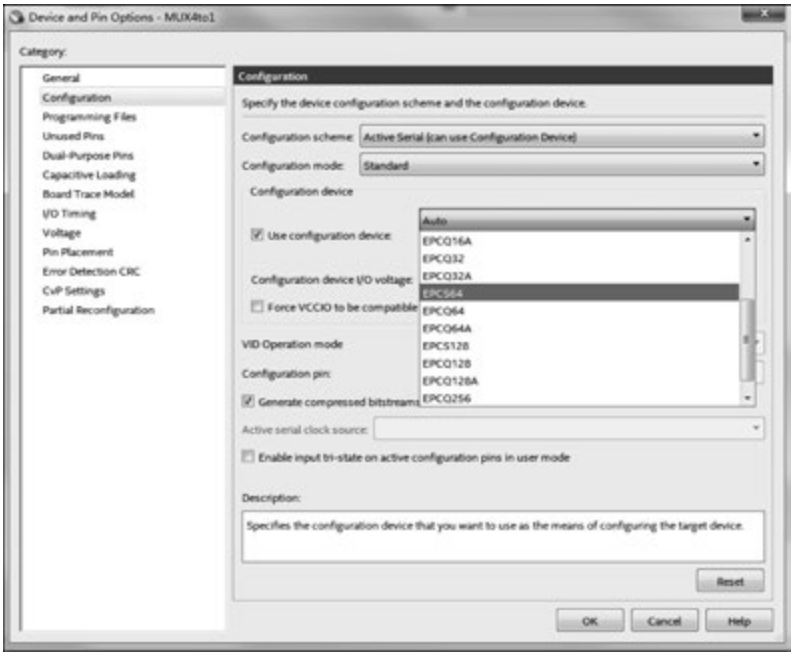


图 3-33 配置器件

(3) 重新启动编译、综合与适配过程,生成.pof 编程文件。

(4) 打开编程器,选择编程与配置方式。

在 Quartus Prime 主界面中执行 Tools→Programmer 菜单命令,打开 Quartus 编程器,在 Mode 下拉列表中选择 Active Serial Programming,如图 3-34 所示。

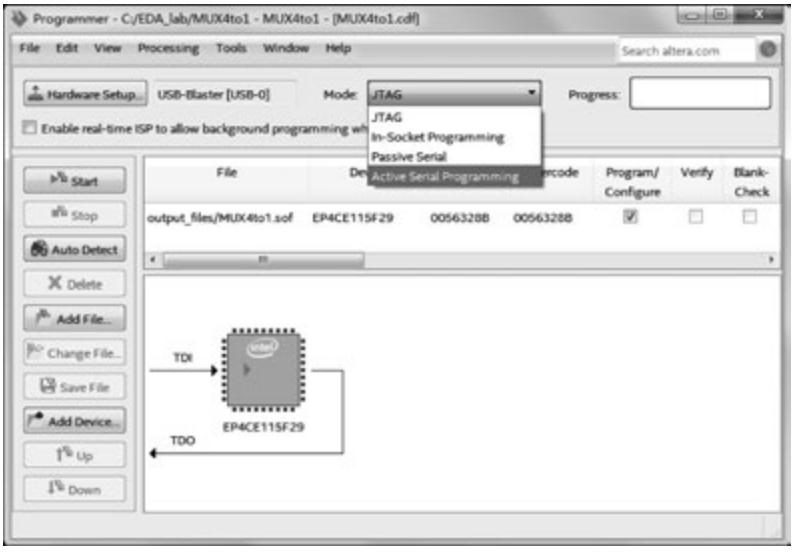


图 3-34 编程模式选择

（5）添加编程文件。

单击编程器窗口左侧的 Add File 按钮，弹出 Select Programming File 对话框，在 output_files 子目录中找到已经生成的.pof 编程文件，如图 3-35 所示。单击 Open 按钮打开。

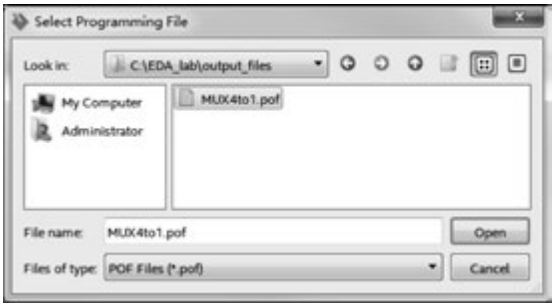


图 3-35　编程文件选择

（6）启动编程过程。

如图 3-36 所示，勾选 Program/Configure，并且确认开发板的 RUN/PROG 开关已经拨到 PROG 位置，单击 Start 按钮开始编程。当进度条（Progress）显示 100%时，编程过程完成。

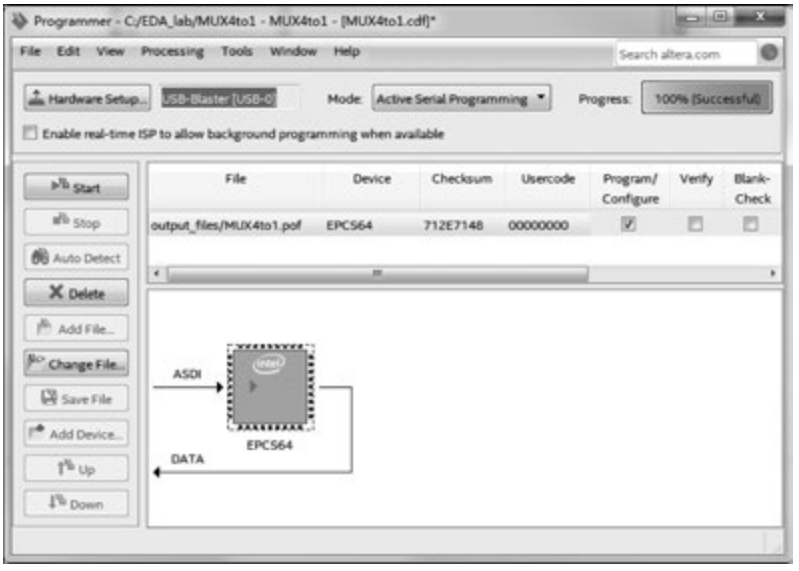


图 3-36　编程过程窗口

编程过程完成后，需要将 DE2-115 开发板的运行与编程开关 SW19 重新拨回 RUN 位置。开发板断电重启后，4 选 1 数据选择器应用电路信息就自动从 EPCS 配置器件加载到 FPGA 中，然后就可以在开发板上测试 4 选 1 数据选择器的功能了。

3. JTAG 间接配置方式

除了 JTAG 配置方式和 AS 编程方式外，Quartus Prime 还支持 JTAG 间接配置方式，用于应用 JTAG 对 EPCS 配置器件进行间接编程。

应用 JTAG 间接配置方式时，首先需要将配置文件.sof 转换为 JTAG 间接配置文件（.jic），然后再应用 JTAG 配置方式将间接配置文件写入 EPCS 器件，完成对配置器件的

编程。

应用 JTAG 间接配置方式的具体步骤如下。

(1) 启动编程文件转换器。

在 Quartus Prime 主界面中执行 File→Convert Programming Files 菜单命令，弹出如图 3-37 所示的 Convert Programming File 对话框。

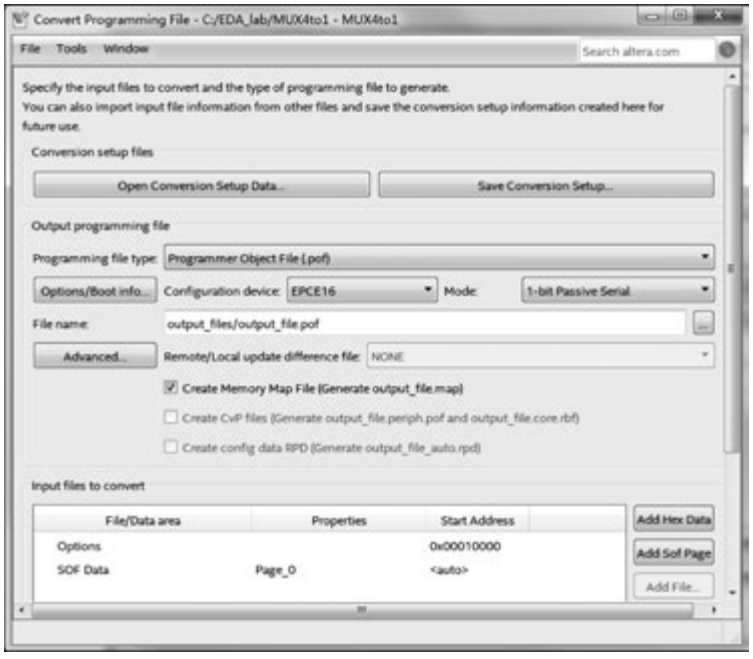


图 3-37 Convert Programming File 对话框

(2) 设置转换信息。

在 Programming file type 下拉列表中选择 JTAG Indirect Configuration File(.jic)，在 Mode 下拉列表中选择 Active Serial 模式，如图 3-38 所示，并在 Configuration device 下拉列表中选择 EPCS64(DE2-115 开发板的配置器件型号)，同时将 File name 文本框默认的文件 output_file.jic 改为 MUX4to1.jic。

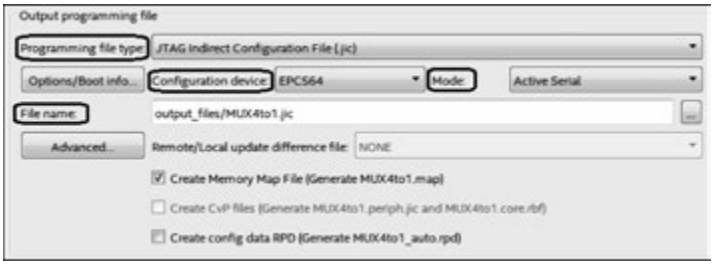


图 3-38 文件转换信息设置

(3) 指定 Flash Loader 器件。

在 Input files to convert 列表中单击 Flash Loader 项，再单击右侧的 Add Device 按钮添加 Flash Loader 器件，如图 3-39 所示。

在弹出的 Select Devices 对话框中，分别选择左侧的(DE2-115 开发板上的)Cyclone Ⅳ

图 3-39　添加 Flash Loader 器件

E 器件类型和右侧的 EP4CE115 器件型号，如图 3-40 所示，表示需要将 Flash Loader 配置到 EP4CE115 FPGA 中。单击 OK 按钮返回 Convert Programming File 对话框。

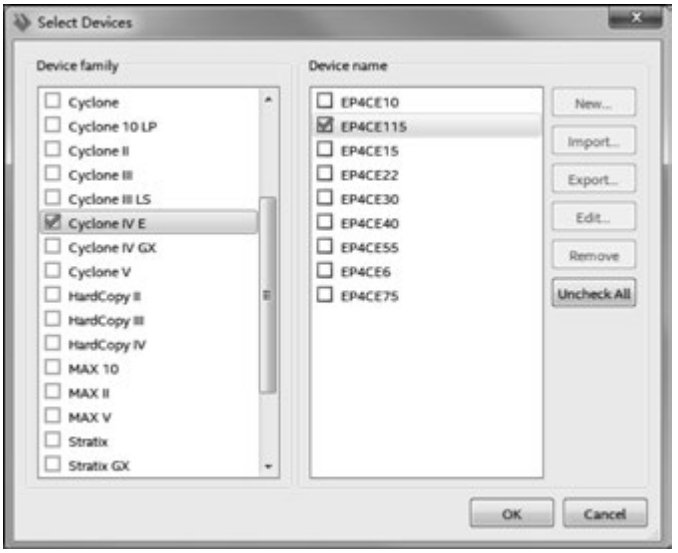


图 3-40　指定 Flash Loader 器件

(4) 添加需要转换的配置文件。

单击 SOF Data，如图 3-41 所示，再单击右侧的 Add File 按钮，添加需要转换的. sof 文件。

图 3-41　添加配置文件

将文件目录切换到当前工程目录的 output files 子目录下，找到配置文件 MUX4to1. sof，单击 Open 按钮打开，如图 3-42 所示，表示需要转换的 SOF Data 为配置文件 MUX4to1. sof。

(5) 生成间接配置文件。

单击 Convert Programming File 对话框中的 Generate 按钮，如图 3-43 所示，启动间接配置文件生成过程。完成后，弹出生成成功消息提示框后，单击 OK 按钮关闭消息提示框。

(6) 编程间接配置文件。

在 Quartus Prime 主界面中执行 Tools→Programmer 菜单命令，启动 Quartus 编程器，

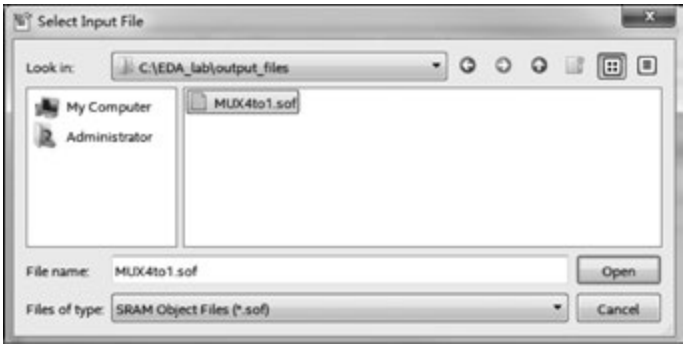


图 3-42 指定配置文件

图 3-43 间接配置文件生成

选择 JTAG 配置方式，单击 Add Files 按钮，添加工程目录的 output_files 子目录下的间接配置文件 MUX4to1.jic，勾选 Programming/Configuration 选项，如图 3-44 所示。

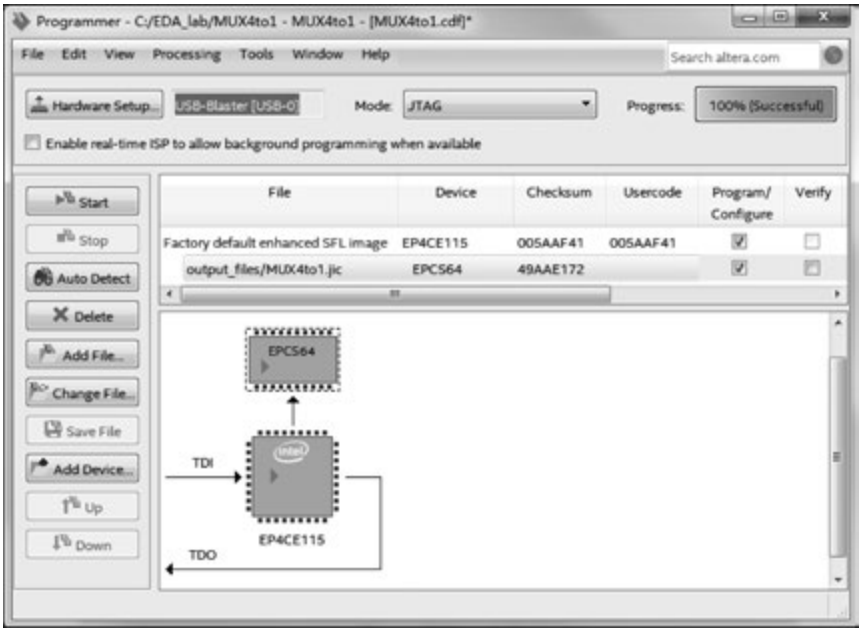


图 3-44 编程间接配置文件

从图 3-44 可以看出，添加间接配置文件后，Quartus 编程器窗口中出现了两项文件信息：①Factory default enhanced SFL image，作用为 Flash Loader，将配置到 FPGA 器件中，引导编程过程；②间接配置文件 MUX4to1.jic，将固化到 EPCS 配置器件中。

确认开发板的运行与编程开关 SW19 处于 RUN 位置后，单击 Start 按钮开始间接配置过程。过程分两步进行：①应用 JTAG 配置 Flash Loader 到目标 FPGA 中；②通过 Flash Loader 将间接配置文件 MUX4to1.jic 写入 EPCS 配置器件中。完成后，硬件电路的设计信息已经固化到 EPCS 配置器件中了。

将开发板断电重启后，FPGA 会主动从 EPCS 配置器件中读取硬件电路的设计信息，并配置到 FPGA 中，4 选 1 数据选择器同样可以正常工作。

3.2 原理图设计方法

Quartus Prime 既支持 HDL 输入方式，又支持原理图设计方法，还支持以 HDL 描述为主、原理图设计为辅的混合方式，以发挥两者各自的优点。底层模块通常应用 HDL 描述，以方便模块的功能定义和重构，使模块的功能既满足设计需求，又能节约 FPGA 资源。顶层电路既可以应用 Verilog HDL 模块例化方法描述各模块之间的连接关系，也可以采用传统的原理图方法进行设计。


第 23 集
微课视频

原理图具有直观形象的优点，应用于顶层电路设计时能够使系统的总体结构清晰明了。有了数字电路课程基础，就可以在电子技术课程设计等实践中应用 EDA 技术设计数字系统了。但是，原理图方法的缺点也很突出，一是设计效率低，二是可移植性差。

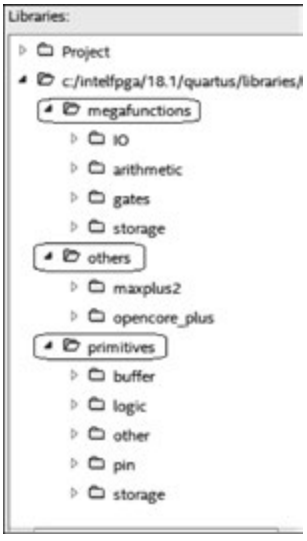


图 3-45 图形化元器件目录

简单的数字系统可以应用原理图方法设计顶层电路。对于复杂的数字系统，则建议应用结构描述方法，通过模块例化方法描述顶层电路的结构。虽然结构描述方法不如原理图方法直观，但有利于系统的功能重构，并且具有良好的可移植性，因而在复杂数字系统设计中应用广泛。

Quartus Prime 提供了 3 类用于原理图设计的符号库：megafunctions、others 和 primitives，如图 3-45 所示。另外，用户也可以将自己设计好的功能模块通过 Quartus Prime 主界面中 Files→Create／Update→Create Symbol Files for Current File 菜单命令封装成图形符号，以便在原理图设计中调用。

megafunctions 符号库包含 IO、arithmetic、gates 和 storage 4 类器件库，有参数化加/减法器、参数化乘/除法器、指数和对数运算、编码器与译码器、比较器、参数化触发器、参数化计数器、参数化 RAM 和参数化 FIFO 等多种类型的器件。

others 符号库包含 74 系列逻辑器件(maxplus2 子目录下)，如 7400/02/04、74138/139、74151/153、74160/161/162/163 等多种商品化数字器件。

primitives 符号库包含 5 类基本图形化符号，如缓冲器(buffer 目录下)、基本逻辑门(logic 目录下)、电源(vcc)和地(gnd)等符号(other 目录下)、输入/输出端口(pin 目录下)和

存储器件(storage 目录下)。

原理图设计方法的流程与 HDL 输入方法的流程相同,只是在选择设计文件类型时,在新建文件的 New 对话框中选择设计文件类型中的 Block diagram/Schematic Files 后,弹出如图 3-46 所示的原理图设计窗口,就可以编辑扩展名为.bdf 的原理图文件了。

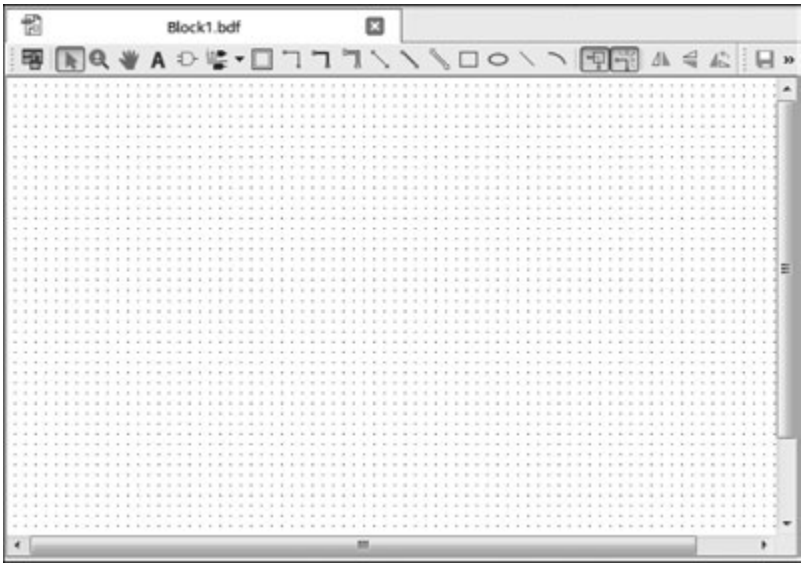


图 3-46 原理图设计窗口

原理图设计窗口包括原理图编辑区和工具栏两部分。原理图编辑区用于绘制原理图,工具栏则包含了绘制和编辑原理图所需要的工具,如表 3-3 所示。

表 3-3 原理图工具

按 钮	工具名称	描 述
	分离窗口	从 Quartus Prime 环境中分离或还原原理图窗口
	选择指针	选择器件、线条等图形元素
	缩放工具	单击放大图形,右击缩小图形
	放置器件	从元件库中选择要添加的器件或符号
	放置端口	从下拉列表中选择放置输入、输出或双向端口
	手工具	在原理图窗口中抓取器件或符号
A	文字工具	在原理图窗口中添加文字
	插入模块	插入已设计好的模块
	正交导线	用于绘制水平或垂直方向的导线
	正交总线	用于绘制水平或垂直方向的总线
	正交导管	用于绘制水平或垂直方向的导管(Conduit)
	对角导线	用于绘制对角方向的导线
	对角总线	用于绘制对角方向的总线

续表

按　钮	工具名称	描　述
	对角导管	用于绘制对角方向的导管(Conduit)
	矩形工具	用于画矩形窗口
	椭圆工具	用于画椭圆窗口
	画线工具	用于画不具有电气功能的连线
	弧线工具	用于画不具有电气功能的弧线
	打开/关闭橡皮筋连接功能	打开时移动元件连接在元件上的连线也跟着移动，不改变与其他元件的连接关系
	打开/关闭局部正交连线选择功能	打开局部正交连线选择功能，可以通过鼠标选择两条正交连线的局部
	垂直翻转	将选中的元件或模块进行垂直翻转
	水平翻转	将选中的元件或模块进行水平翻转
	旋转工具	将选中的元件或模块逆时针方向旋转 90°

应用原理图设计方法的基本步骤如下。

1. 添加图形符号

在原理图编辑区的任意空白处双击，或者单击工具栏中的按钮，弹出如图 3-47 所示的 Symbol 对话框。在左侧的 Libraries 列表中按分类查找所需要的元器件，选中后单击 OK 按钮，在原理图编辑区出现随鼠标移动的元器件符号，在绘图区适当的位置单击放置元器件。

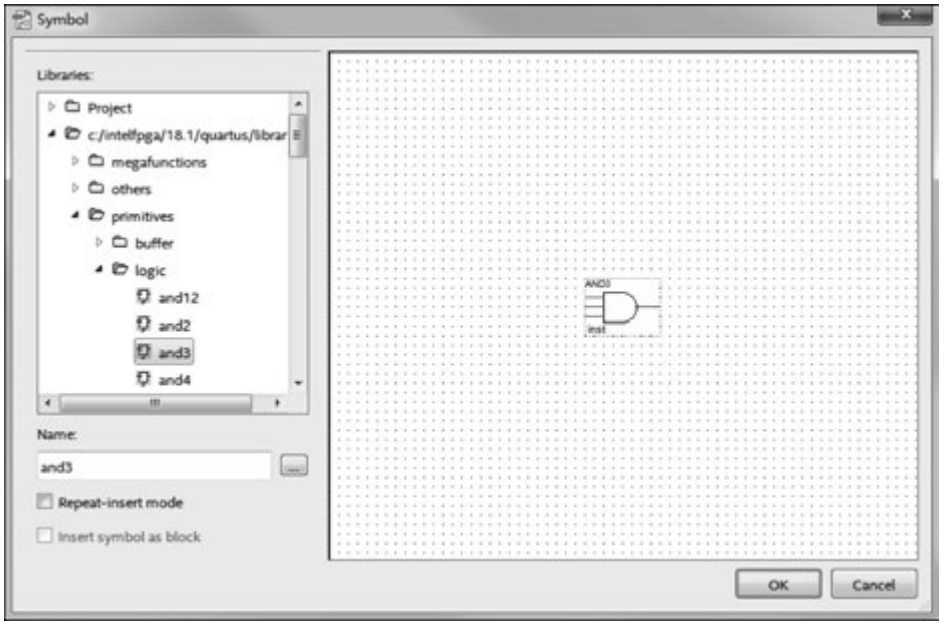


图 3-47　Symbol 对话框

2. 连接元器件

放置好元器件后，需要连线，将鼠标指向器件的端口，当鼠标指针变为直角形状时进入

连线状态。拖动鼠标到另一个连接点位置后松开，则会放置一段连线。反复操作完成所有的连线。

需要删除连线时，先单击连线使其处于选中状态，然后按 Delete 键删除，或者右击连线，在弹出的快捷菜单中选择 Delete。

需要注意的是，原理图中的总线(Bus)为粗线，导线(Wire)为细线。从端口引出的连线是导线还是总线，取决于端口的宽度。若端口的宽度为 1 位，则引出的是导线；若端口的宽度为多位，则引出的是总线。

3. 设置 I/O 端口

在原理图编辑区的任意空白处双击，或者单击工具栏中的 按钮，弹出 Symbol 对话框。展开 primitives→pin 列表选择 input、output 或 inout 端口符号，单击 OK 按钮，出现随鼠标移动的端口图标，在绘图区合适的位置单击放置端口符号，调整端口方向，连接到器件需要输入/输出的引脚上。

双击端口符号中的 pin_name，修改端口名。注意，端口名的格式为

```
端口名[msb..lsb]
```

其中，msb 和 lsb 用于定义端口的位宽，默认的位宽为 1。

需要注意的是：①在原理图中，用于定义端口位宽的 msb 和 lsb 用符号“..”连接，而不是冒号(用冒号会导致编译错误)；②I/O 端口的宽度必须与其相连接的器件端口位宽严格一致。

4. 保存文件

执行 File→Save/Save as 菜单命令保存原理图设计文件。如果原理图为顶层设计电路，则保存的文件名应为工程名加上原理图文件扩展名(.bdf)。

4 选 1 数据选择器原理图如图 3-48 所示，设置文件名为 MUX4to1.bdf。

原理图设计完成后，同样需要进行编译、综合与适配，以及引脚锁定、编程与配置。这些过程与 HDL 输入法完全相同，在此不再复述。

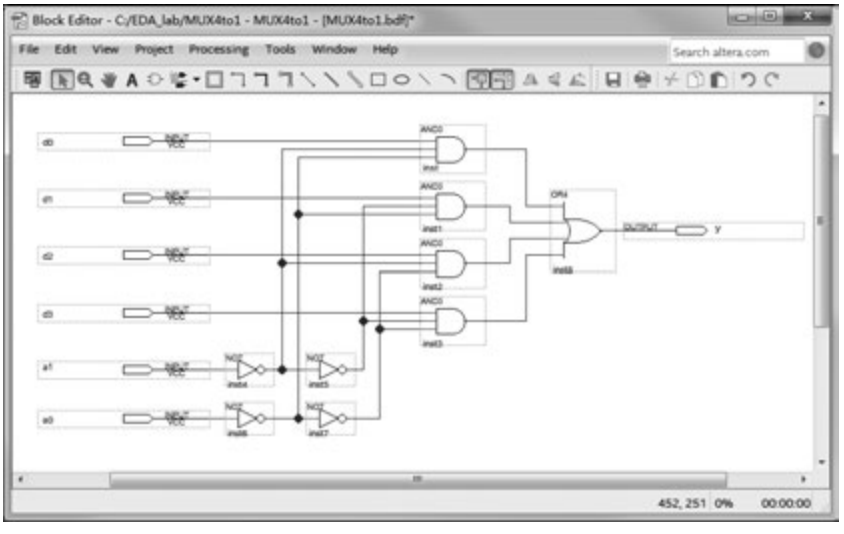


图 3-48 4 选 1 数据选择器原理图

3.3 仿真分析

EDA设计的基本流程包括建立工程，设计输入，编译、综合与适配，引脚锁定以及编程与配置5个主要环节。完成了设计输入并成功进行编译与综合后，并不能说明综合出的电路功能正确，性能能够满足设计要求。

模块功能的验证通常有两种方法：①应用ModelSim对设计电路进行仿真分析；②在设计电路中嵌入逻辑分析仪(Signal Tap Logic Analyzer)进行在线测试。

仿真(Simulation)是通过软件算法模拟电路的运行验证电路的逻辑功能，以及在适配过程中引入分布参数(包括器件传输延迟和布线延迟)后，验证其功能是否依然正确无误。

仿真分为功能仿真和时序仿真两种类型。

功能仿真(Functional Simulation)是指在不考虑器件的传输延迟和布线延迟情况下的仿真；而时序仿真(Timing Simulation)是指在完成布局布线之后进行包含了器件传输延迟和布线延迟信息的仿真。功能仿真为理想化仿真，而时序仿真结果能够较真实地反映电路的实际性能。


第24集
微课视频

Quartus Prime调用ModelSim进行仿真分析。Mentor公司为Altera公司提供了两种OEM版的ModelSim：ModelSim_ae和ModelSim_ase。使用ModelSim_ae需要有相应许可证的支持，而ModelSim_ase为入门版，虽然在应用上有一定的限制，只能仿真10000行(及以下)的可执行代码，但是完全能够满足大部分工程项目的仿真需求。


第25集
微课视频

Quartus Prime支持两种仿真方法：①基于Intel公司大学计划项目(University Program)的向量波形(Vector Waveform)仿真方法；②编写测试平台文件(testbench)进行仿真。无论哪种方法，都需要调用ModelSim进行仿真，只是方法不同而已。

3.3.1 基于向量波形的仿真方法

基于向量波形的仿真是图形化的仿真方法，具有直观易用的优点，适合简单工程的仿真需要。

下面介绍基于向量波形的仿真方法的具体步骤。

1. 建立向量波形文件

在设计工程中，在Quartus Prime主界面中执行File→New菜单命令，弹出New对话框(见图3-11)，选择Verification/Debugging Files栏下的University Program VWF，单击OK按钮，将弹出如图3-49所示的向量波形文件编辑窗口，其中窗口上方为波形编辑工具栏。

波形编辑工具栏的下方分为两部分：左侧为节点名称(Name)栏，用于添加需要设置的输入激励信号以及需要观察的内部节点和输出信号；右侧为波形区，用于编辑输入信号的波形和显示仿真后内部节点和输出信号的波形。

2. 插入节点或总线

在波形编辑窗口左侧的Name栏的空白处右击，在弹出的快捷菜单中选择Insert Node or Bus，弹出如图3-50所示的Insert Node or Bus对话框。单击Node Finder按钮，弹出Node Finder对话框，如图3-51所示。

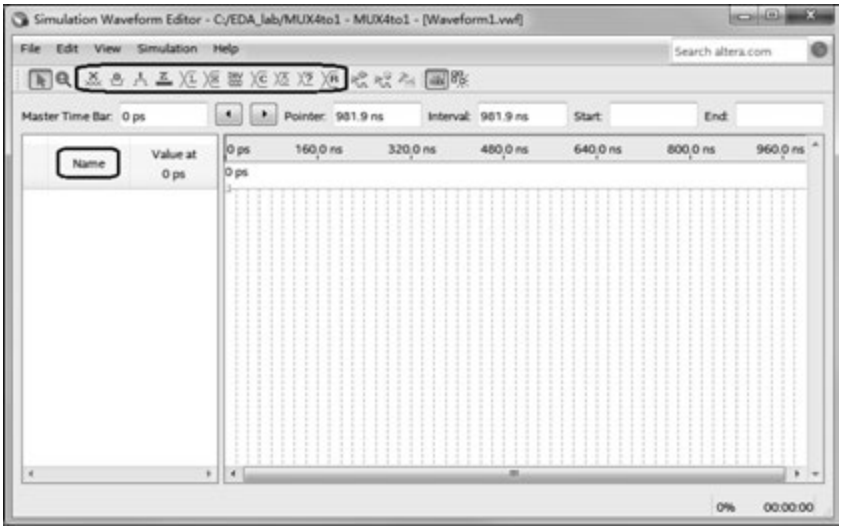


图 3-49 向量波形文件编辑窗口

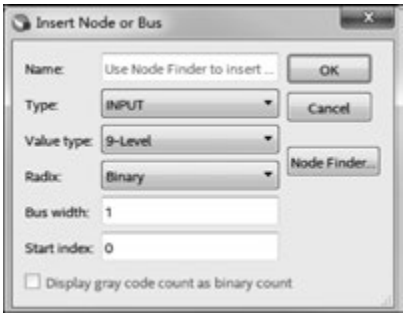


图 3-50 Insert Node or Bus 对话框

图 3-51 Node Finder 对话框

在插入节点或总线之前，可以先进行过滤(Filter)和定位(Look in)，以缩小查找范围。

选择 Filter 下拉列表中的 Pins:all，选择模块的端口，单击 List 按钮，将在 Nodes Found 列表中列出工程所有的 I/O 信号。选择需要设置和观测的端口，单击＞按钮将端口添加到右侧 Selected Nodes 列表(不需要的端口可以通过＜按钮移出，也可以通过>>和<<按钮将端口全部移入或移出)，如图 3-52 所示。完成后单击 OK 按钮返回，再单击对话框中的 OK 按钮返回波形文件主窗口。

3. 设置仿真参数

向量波形文件编辑窗口中 Edit 菜单下的 Grid Size 和 Set End Time 命令用于设置仿真步长和仿真结束时间。

Quartus Prime 默认的仿真步长为 10ns，仿真结束时间为 1μs。由此可以推出，Quartus Prime 默认的仿真次数为 1μs/10ns＝100 次。对于组合逻辑电路，对应输入信号 100 种组合；对于时序电路，对应 50 个时钟周期。若仿真次数不够，则可以通过加长仿真结束时间或减小仿真步长的方法增加仿真次数。

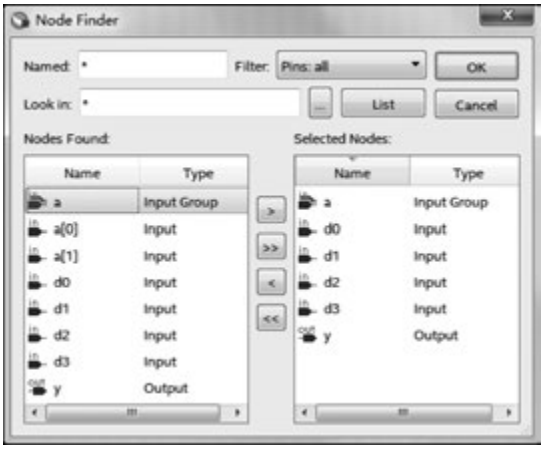


图 3-52　插入节点

一般推荐使用增加仿真结束时间的方法。例如，对 8 位二进制计数器进行分析时，至少需要仿真 $2^8=256$ 个时钟周期，才能观察到一个完整的计数循环过程，因此需要将 End Time 调整为 $256\times2\times10\text{ns}=5.12\mu\text{s}$；而对于基本的 3 线-8 线译码器进行仿真时，由于输入的二进制码只有 8 种取值组合，所以只需要仿真 8 次，因此设置 End Time 为 $8\times10\text{ns}=80\text{ns}$。

4. 设置输入信号波形

仿真参数设置完成后，接下来需要应用波形编辑工具设置输入激励信号的波形。Quartus Prime 提供的波形编辑工具的功能描述如表 3-4 所示。

表 3-4　波形编辑工具

按钮	工具名称	描　　述
	选择工具	选择信号或波形
	放缩工具	利用鼠标(左键)放大/(右键)缩小显示波形
	赋 x	将选中的波形段赋值为未知
	赋 0	将选中的波形段赋值为低电平
	赋 1	将选中的波形段赋值为高电平
	赋 z	将选中的波形段赋值为高阻状态
	取反	将选中的波形段反相
	计数赋值	以计数方式为周期性信号赋值
	时钟赋值	以时钟方式为周期性时钟信号赋值
	随机赋值	对选中的信号段进行随机赋值
	功能仿真	启动功能仿真(Functional Simulation)
	时序仿真	启动时序仿真(Timing Simulation)
	生成文件	生成 testbench 和 Script 文件
	栅格对齐	栅格对齐
	捕捉过渡	捕捉过渡

波形设置的具体方法：单击 Name 区的输入信号，当信号变为蓝色时，表示该信号已被选中。在波形区，拖动选择需要编辑的信号波形段，单击工具栏中的 0 或 1 按钮将该信号段设置为低电平或高电平，也可以单击工具栏中的 C（Count Value）按钮对信号进行周期赋值，以仿真步长的倍数为基准进行变化，或者单击工具栏中的 R（Random Values）按钮对信号进行随机赋值。

设置 4 选 1 数据选择器输入信号的波形，如图 3-53 所示。其中，4 路数据 d0、d1、d2 和 d3 分别以 10ns、20ns、30ns 和 40ns 为步长循环变化。单击信号名称左侧的 ◢ 按钮展开 2 位地址 a，设置 a[1]和 a[0]分别以 500ns 和 250ns 为步长变化，则 2 位地址 a 按 00、01、10 和 11 的顺序变化。

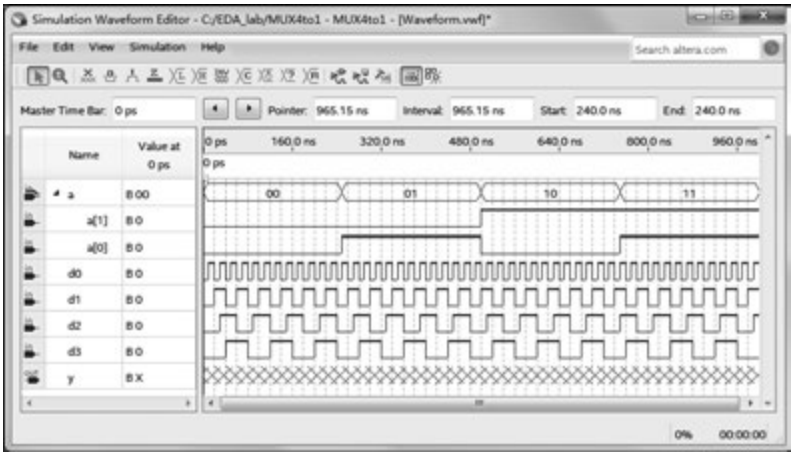


图 3-53 设置输入信号波形

5. 保存向量波形文件

执行向量波形文件编辑窗口 File→Save /Save as 菜单命令，将向量波形保存为.vwf 文件。

6. 启动仿真

需要进行功能仿真时，执行 Simulation→Run Functional Simulation 菜单命令，或者直接单击工具栏中的 按钮启动功能仿真。进行时序仿真时，执行 Simulation→Run Timing Simulation 菜单命令，或者直接单击工具栏中的 按钮启动时序仿真。

4 选 1 数据选择器的功能仿真结果如图 3-54 所示。分析仿真波形，验证逻辑功能是否正确。从图 3-54 中输出信号与输入信号的对应关系可以看出，当地址 a＝2'b00 时，输出 y 的波形与 d0 的波形相同；当 a＝2'b01 时，输出 y 的波形与 d1 的波形相同；当 a＝2'b10 时，输出 y 的波形与 d2 的波形相同；当 a＝2'b11 时，输出 y 的波形与 d3 的波形相同。因此，4 选 1 数据选择器功能正确。

如果仿真结果有误，就需要修改描述代码重新进行编译与综合过程，再进行仿真，直到模块功能正确为止。

3.3.2 基于 testbench 的仿真方法

基于向量波形的仿真方法适用于对简单工程进行仿真。对于复杂的数字系统，应用向

量波形进行仿真时难以设置复杂的输入信号波形组合，因而应用上有很大的局限性。

对于复杂的数字系统，建议应用基于测试平台文件（testbench）的仿真方法。基于testbench 进行仿真时，需要编写测试平台文件 testbench。

4 选 1 数据选择器的测试平台文件 testbench 与被测模块 MUX4to1 之间的关系如图 3-55 所示。仿真时，testbench 为 MUX4to1 模块为 4 路输入信号 d0、d1、d2、d3 以及 2 位地址 a 提供输入激励，例化 MUX4to1 并指定观测信号及显示数据格式，调用 ModelSim 仿真并输出观测信号以供设计者进行分析。

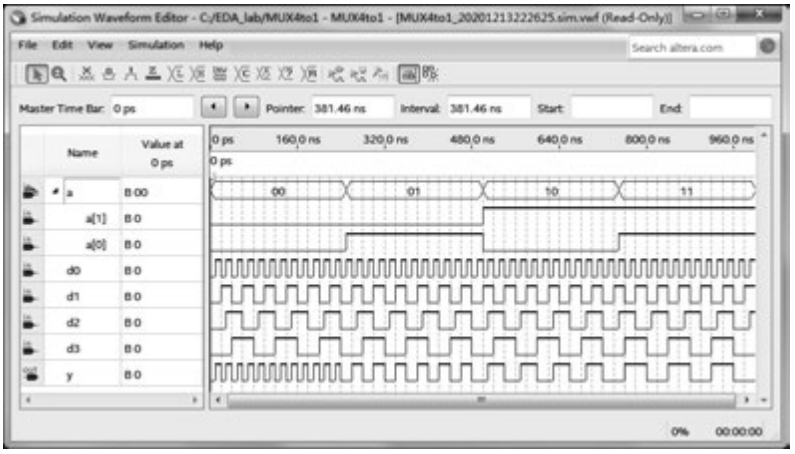


图 3-54　4 选 1 数据选择器功能仿真结果

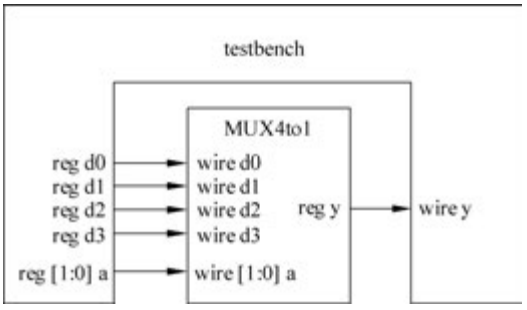


图 3-55　testbench 与被测模块的关系

需要说明的是，当 ModelSim 仿真软件随 Quartus Prime 一起安装时，Quartus Prime 会自动关联 ModelSim。在仿真过程时，如果出现 Quartus Prime 无法调用 ModelSim 的情况，则需要手动关联 ModelSim：在 Quartus Prime 主界面中，执行 Tools→Options 菜单命令，在弹出的 Options 对话框中选择 General 栏下的 EDA Tool Options，进入 EDA 工具选择对话框，在 ModelSim-Altera 栏中设置 ModelSim 的安装路径，如图 3-56 所示，具体与安装 Quartus Prime 时的路径设置有关。设置完成后单击 OK 按钮退出。

应用 testbench 进行 4 选 1 数据选择器仿真的具体步骤如下。

1. 生成 testbench 模板文件

在 Quartus Prime 主界面中，执行 Processing→Start→Start Test Bench Template Writer 菜单命令，将在“当前工程目录\simulation\modelsim”文件夹中自动生成一个类型

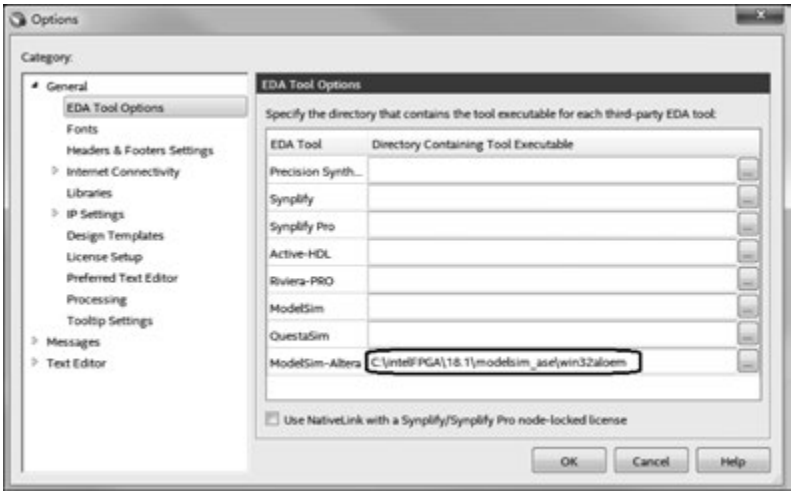


图 3-56 手动关联 ModelSim

名为.vt 的测试平台文件(vt 表示 Verilog testbench),文件名与工程名一致。

对于 4 选 1 数据选择器,测试平台文件文件名为 MUX4to1.vt。

打开测试平台文件 MUX4to1.vt,可以看到,自动生成的 testbench 模板文件已经完成了模块定义(默认模块名为“工程名_vlg_tst”)、端口声明和模块例化,但核心代码 initial 和 always 过程语句部分是空白的,如图 3-57 所示。用户需要添加代码设置输入信号的波形并指定需要观测的输出信号及其显示格式。

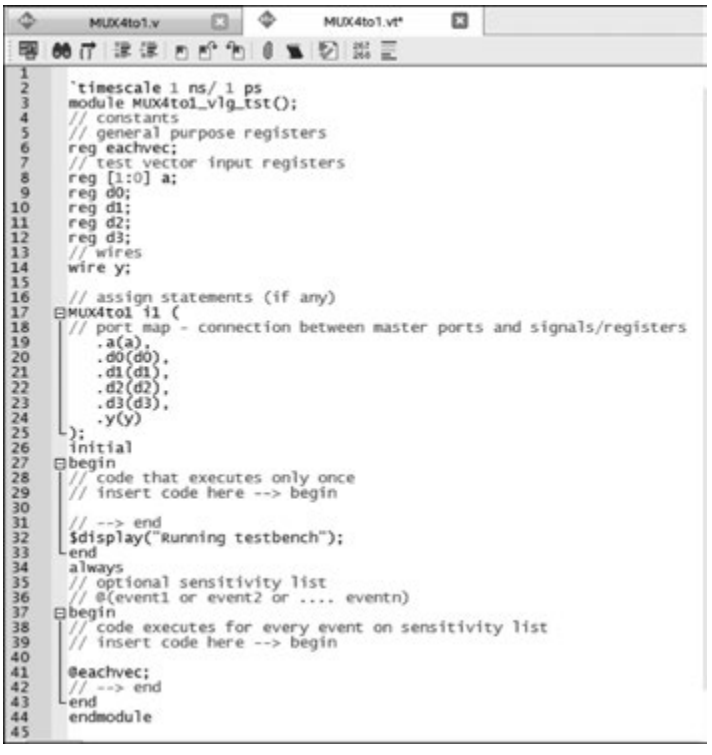


图 3-57 4 选 1 数据选择器 testbench 模板

2. 编辑 testbench

在测试平台文件中应用 initial 和 always 过程语句设置 4 路输入数据和 2 位地址的起始值和波形，如图 3-58 所示，同时调用系统任务 $monitor 持续监测输出信号 y 的变化。

图 3-58 编辑 testbench 代码

3. 关联 testbench 文件

执行 Assignments→Settings 菜单命令，在弹出的 Settings 对话框中的 Category 列表中选择 EDA Tool Settings→Simulation，进入如图 3-59 所示的仿真设置界面，将 Time scale 参数值设置为 1ns(与 MUX4to1.vt 中的参数设置一致)。

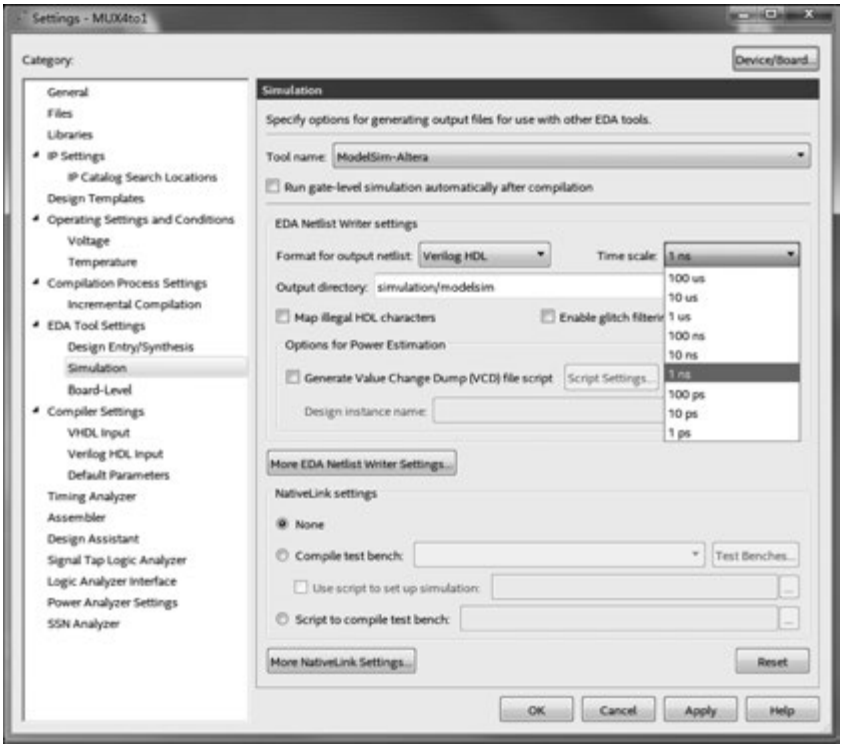


图 3-59 仿真设置

设置 NativeLink settings 栏中的相关信息以关联 testbench 文件。

(1) 勾选 Compile test bench，如图 3-60 所示，单击右侧的 Test Benches 按钮，弹出 Test Benches 对话框，以指定 testbench 文件。

图 3-60 关联 testbench 文件(1)

如果没有关联过 testbench 文件，则 Test Benches 文件列表是空白的，如图 3-61 所示。

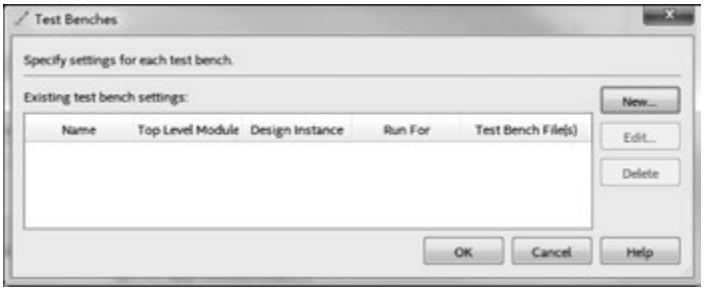


图 3-61 关联 testbench 文件(2)

(2) 关联 testbench 文件。单击 New 按钮，弹出 New Test Bench Settings 对话框，需要将相应的 testbench 文件名和相应的顶层模块名填入对应文本框中。对于 4 选 1 数据选择器的仿真，在 Test bench name 文本框中输入测试平台文件 MUX4to1.vt，在 Top level module in test bench 文本框中输入 testbench 模块名 MUX4to1_vlg_tst，如图 3-62 所示。

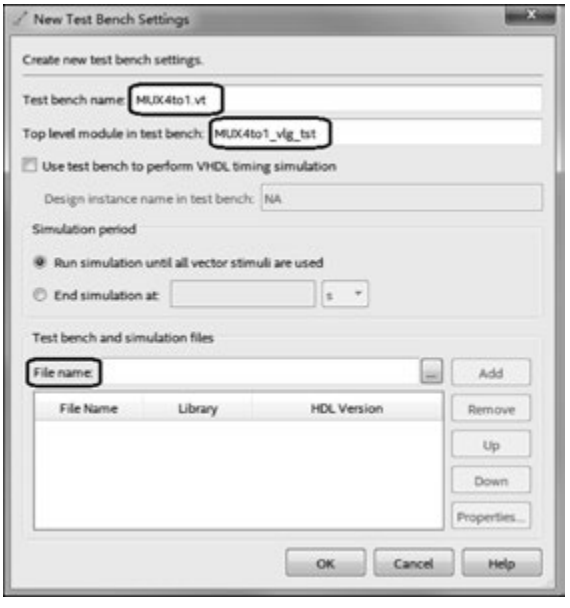


图 3-62 关联 testbench 文件(3)

(3) 添加 testbench 文件。单击 File name 文本框后的 ... 按钮，浏览“当前工程目录\simulation\modelsim”文件夹，查找 testbench 文件(MUX4to1. vt)，选中并加入(单击 Add 按钮)File Name 栏中，如图 3-63 所示。

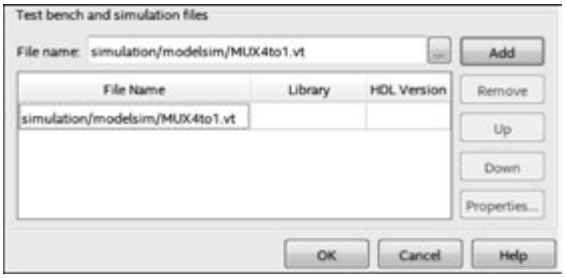
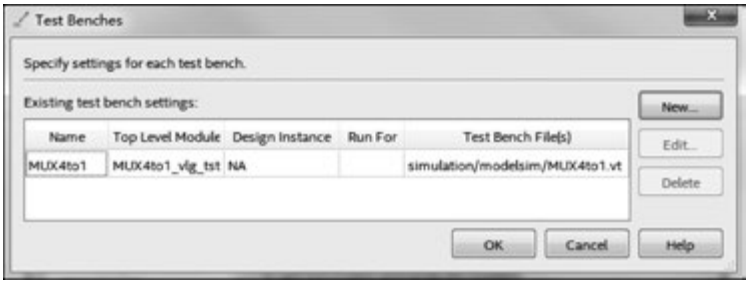


图 3-63 关联 testbench 文件(4)

单击 OK 按钮，弹出 Test Benches 对话框，如图 3-64 所示。连续单击 OK 按钮完成设置过程。



第 26 集
微课视频

图 3-64 Test Benches 对话框

4. 启动仿真

设定完成后，在 Quartus Prime 主界面中，执行 Tools→Run Simulation Tool→RTL Simulation 菜单命令，调用 ModelSim 进行功能仿真。如果需要进行时序仿真，则执行 Tools→Run Simulation Tool→Gate Level Simulation 菜单命令。仿真完成后，会自动弹出 ModelSim 波形窗口，如图 3-65 所示。选定波形栏，单击 按钮缩放波形以便于分析。

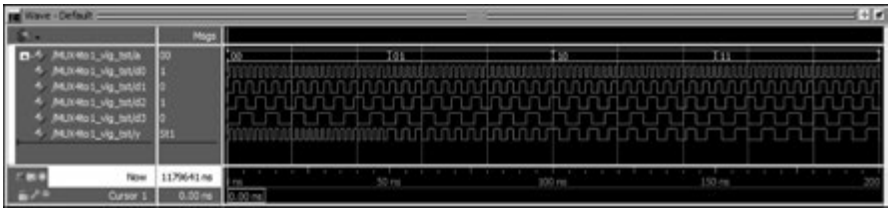


图 3-65 4 选 1 数据选择器 ModelSim 波形图

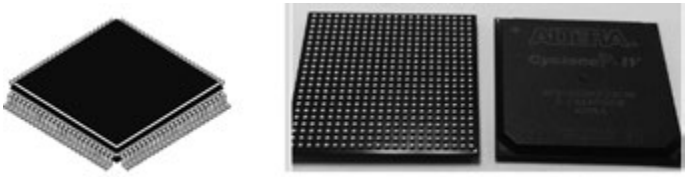
3.4 逻辑分析仪的应用

数字系统开发不但需要仿真分析，还需要进行实际测试。

传统的测试方法是将逻辑分析仪和示波器等仪器设备连接到芯片的引脚或电路连线上

进行测试。对于基于可编程逻辑器件设计的数字系统，这种测试方法需要预先将 PLD 内部需要观测的节点或总线锁定到芯片引脚上，再外接逻辑分析仪等设备进行测试。

随着半导体制造工艺水平的提高，FPGA 的密度越来越大，I/O 引脚数也越来越多，而且大多数 FPGA 采用了微间距的 TQFP(Thin Quad Flat Package)封装或 BGA(Ball Grid Array)封装，如图 3-66 所示，使采用传统的通过芯片引脚对内部信号进行测试的方法变得越来越难以实现。



(a) TQFP封装　　(b) BGA封装

图 3-66　FPGA 封装

Signal Tap Logic Analyzer 是内嵌于 Quartus Prime 开发环境中的逻辑分析仪。设计者可以将 Signal Tap Logic Analyzer 同设计电路一起配置到 FPGA 器件中。Signal Tap Logic Analyzer 能够在电路工作期间实时捕获电路内部节点的信号和总线上的信息流，然后通过 JTAG 接口将采集到的数据反馈给 Quartus Prime 开发环境以显示电路内部节点上的信号波形或者总线上的信息流。

使用 Quartus Prime 内嵌的 Signal Tap Logic Analyzer 无须外接逻辑分析设备，设计者只需要通过 USB-Blaster 连接到需要测试的目标器件上，就可以应用 Quartus Prime 开发环境对 FPGA 内部硬件电路工作时的数据进行采集和显示，而且不影响硬件电路的正常工作。

为了测试 4 选 1 数据选择器，需要为 4 选 1 数据选择器提供 4 路激励数据 d0、d1、d2、d3，然后在 2 位地址 a 的作用下对 4 选 1 数据选择器的 4 路数据和输出 y 进行采样并显示，以供设计者分析电路功能是否满足正确。

1. 建立测试工程

在 4 选 1 数据选择器工程目录(c:\EDA_lab)下，新建测试工程 MUX4to1_tst，然后定制锁相环 IP-ALTPLL(设模块名为 PLL_for_MUX4to1_tst)，设置锁相环的 5 路输出信号 c0、c1、c2、c3 和 c4 依次为 4MHz、3MHz、2MHz、1MHz 和 100MHz 方波，分别作为 4 选 1 数据选择器的 4 路输入数据 d0、d1、d2、d3 和逻辑分析仪 Signal Tap Logic Analyzer 的时钟。

锁相环的定制方法和具体步骤将在本书 5.2 节中讲述。

新建原理图顶层设计文件，将定制好的锁相环 PLL_for_MUX4to1_tst.qip 和 4 选 1 数据选择器原理图符号文件 MUX4to1.bsf(在 MUX4to1 工程中，通过 Files→Create ∠Update→Create Symbol Files for Current File 菜单命令生成)连接成如图 3-67 所示的测试工程顶层电路，同时将 4 选 1 数据选择器的 2 位地址 a[1:0]分别锁定到两个外接开关 SW_1 和 SW_0 上，通过改变开关的状态观察数据选择器输出信号 y 的变化。

2. 新建逻辑分析文件

在 Quartus Prime 主界面中，执行 File→New 菜单命令，弹出 New 对话框，选中 Verification/Debugging Files 栏下的 Signal Tap Logic Analyzer File 文件类型，单击 OK 按

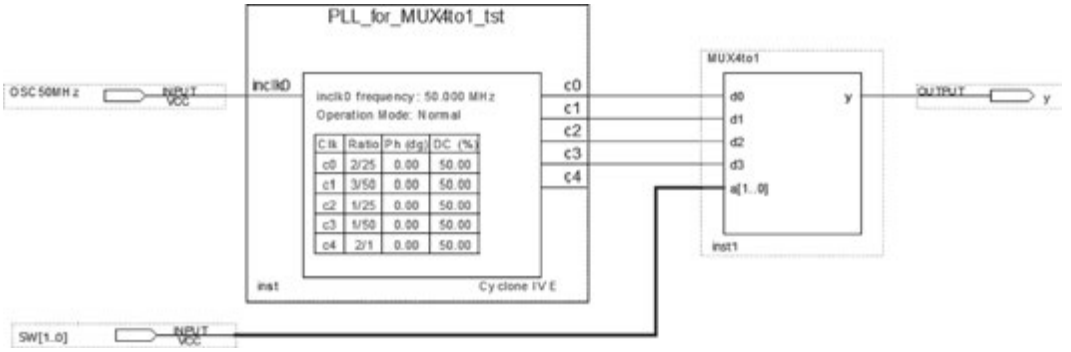


图 3-67　测试工程顶层电路

钮确认，将弹出 Signal Tap Logic Analyzer 主窗口，如图 3-68 所示。

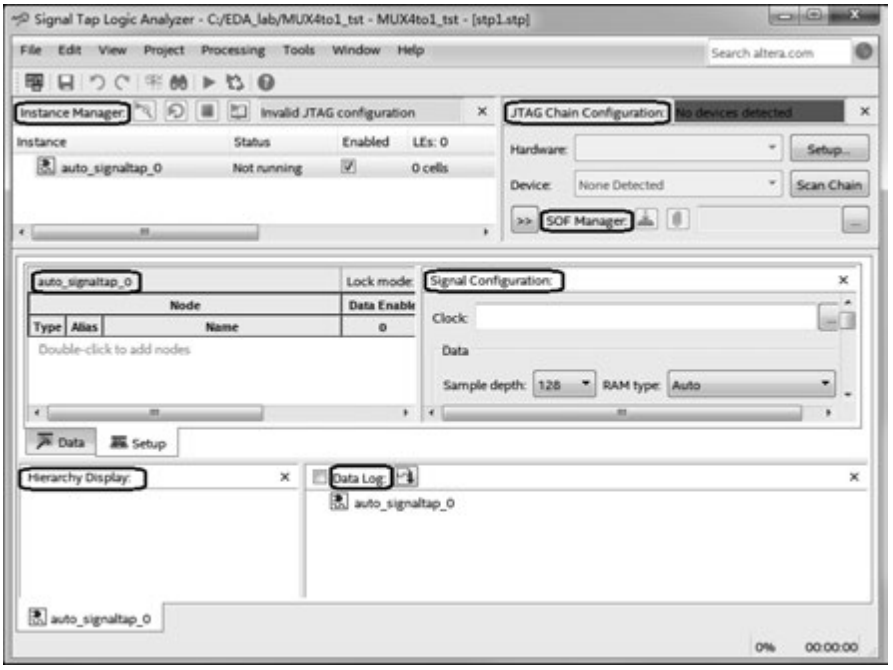


图 3-68　Signal Tap Logic Analyzer 主窗口(1)

Signal Tap Logic Analyzer 主界面主要包含例化管理器(Instance Manager)、JTAG 链配置区(JTAG Chain Configuration)、SOF 管理器(SOF Manager)、信号节点列表区(auto_signaltap_0)、信号配置区(Signal Configuration)、层次显示区(Hierarchy Display)和数据日志(Data Log)共 7 个区域。

例化管理器用于控制 Signal Tap Logic Analyzer 的工作过程，在没有设置待测信号和参数之前，例化管理器中的按钮是灰色的，为不可用状态。JTAG 链配置区用于指定具体的 JTAG 连接(对于 DE2-115 开发板，为 USB-Blaster)。SOF 管理器用于指定加载到 FPGA 的.sof 配置文件名和启动配置过程。信号节点列表区用于选择需要观测的信号并设置相关参数。信号配置区用于指定 Signal Tap Logic Analyzer 的时钟源、设置采样深度，以及触发控制和触发条件等相关参数。

单击例化管理区中的 auto_signaltap_0，将默认的逻辑分析仪名 auto_signaltap_0 更改为 signaltap_MUX4to1(方便记忆)。更改之后，信号列表区、层次显示区和数据日志中的名称也随之调整，如图 3-69 所示。

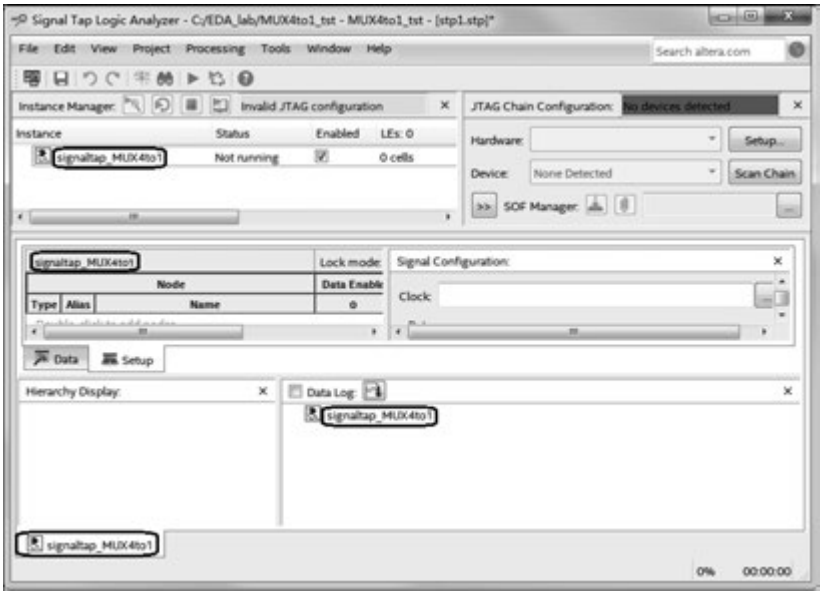


图 3-69 Signal Tap Logic Analyzer 主窗口(2)

3. 添加需要观测的信号

在信号节点列表区(signaltap_MUX4to1)的空白处双击，弹出 Node Finder 对话框。先单击 按钮展开 Options 选项，然后在 Filter 下拉列表中选择 Design Entry(all names)，单击 List 按钮列出电路中所有模块的对外端口，如图 3-70 所示。

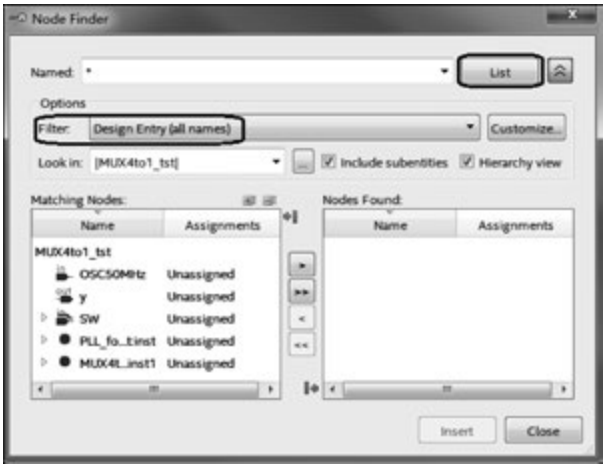


图 3-70 模块的对外端口

单击 Matching Nodes 列表中的 MUX4to1：inst1 左侧的 展开，分别将数据选择器的 4 路输入数据 d0、d1、d2、d3 和 2 位地址 a 以及输出 y 添加到 Nodes Found 列表，作为需要观测的信号，如图 3-71 所示。

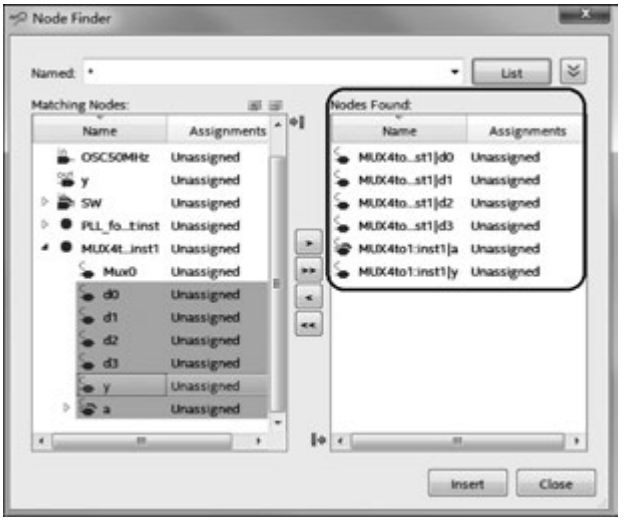


图 3-71　添加观测节点

单击图 3-71 中的 Insert 按钮将需要观测的信号添加到信号列表区，如图 3-72 所示。注意，不要添加多余的节点到信号列表区，因为添加过多的信号会导致 Signal Tap Logic Analyzer 占用更多的 FPGA 资源，可能会导致编译失败。

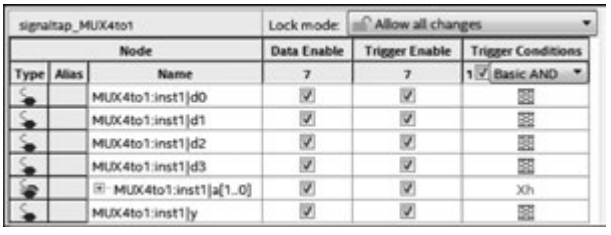


图 3-72　待测节点列表

节点列表中的 Data Enable 和 Trigger Enable 复选框栏用于启用或禁用已加入节点列表中相关信号的使用。如果禁用 Data Enable，则 Signal Tap Logic Analyzer 不会采集相应的信号。如果禁用 Trigger Enable，则相应的信号不用作触发条件定义。利用这些选项有助于减少逻辑分析仪所占用的资源。

触发条件(Trigger Conditions)有 Basic AND、Basic OR、Comparison 和 Advanced 4 个选项。Basic 用于为单个信号设置触发模式，其中 Basic AND 表示选定触发级数的所有信号相与的结果为真时，逻辑分析仪开始捕捉数据。如果需要将观察信号的不同组合作为触发条件，则应选择 Advanced，并且需要为逻辑分析仪定义触发条件表达式。

4. 配置采样参数

信号添加完成后，还需要配置 Signal Tap Logic Analyzer 的采样参数，指定采样时钟和设置采样深度。对于更深层次的应用，还需要设置触发流控制、触发位置设置和触发条件等相关信息。

(1) 设置采样时钟。在信号配置区，单击 Clock 文本框右侧的浏览按钮 ...，在弹出的 Node Finder 对话框中单击 List 按钮，列出工程中所有节点。展开锁相环 PLL_for_MUX4to1_tst：inst，将输出 c4 添加到 Nodes Found 列表中，如图 3-73 所示。单击 OK 按

钮，即指定 Signal Tap Logic Analyzer 的采样时钟来自锁相环 PLL_for_MUX4to1_tst 的 c4 输出的 100MHz 信号。

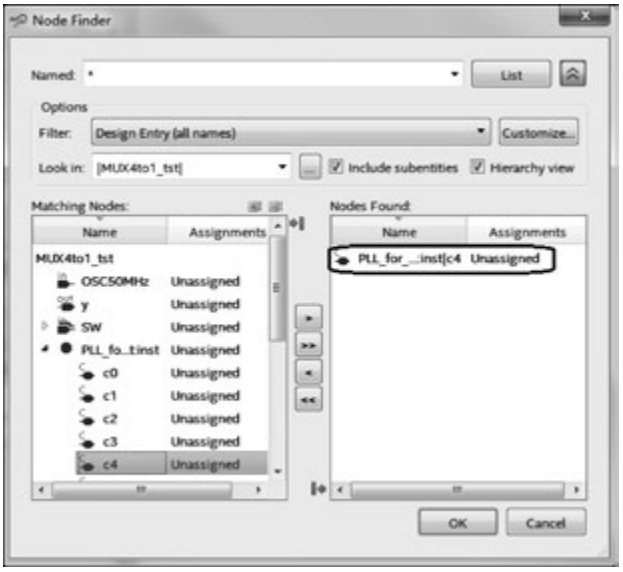


图 3-73 设置采样时钟

（2）设置采样深度。在信号配置区中 Data 区域 Sample depth 下拉列表中选择采样深度，如图 3-74 所示。采样深度决定了采样信号存储数据量的大小，应根据采样条件和被测信号的数量决定，同时受 FPGA 内部 RAM 资源量的限制。采样深度确定之后，所有待测信号都获得同样的采样深度。

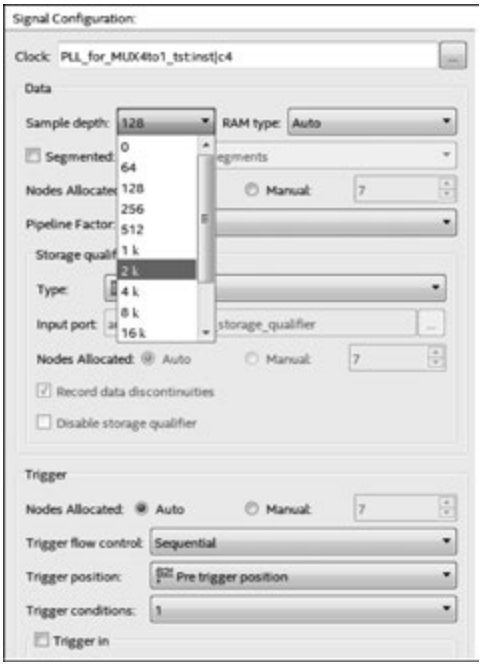


图 3-74 设置采样深度

当选择采样深度为 2k(2000 个采样点)时，在时钟频率为 100MHz(周期为 10ns)的情况下，每次采样时长为 2000×10ns=20μs。因此，4 路输入数据 d_0、d_1、d_2、d_3 为 4MHz、3MHz、2MHz 和 1MHz 的情况下，每次分别采样 80、60、40 和 20 个数据周期，能够满足分析要求。

对于复杂的工程测试，还可以设置 Trigger 区域中的触发流控制、触发位置、触发级数、触发信号和触发方式等相关参数，以控制数据采集过程。

(1) 触发流控制(Trigger flow control)有顺序(Sequential)和状态机(State-based)两个选项。其中，Sequential 表示按顺序计算所有的触发条件，即当一个触发条件满足后，判断第 2 个触发条件是否满足，以此类推，当所有的条件都满足时触发采集过程；State-based 表示应用状态机指定触发顺序，适用于较为复杂的触发控制。

Signal Tap Logic Analyzer 默认选择顺序控制，如图 3-74 所示。

(2) 触发位置(Trigger position)下拉列表用于设置触发位置，以确定触发位置前后应采集的数据量。其中，Pre trigger position 表示前点触发，保存触发前后 12%和 88%的数据信息；Center trigger position 表示中点触发，保存触发前后各 50%的数据信息；Post trigger position 表示后点触发，保存触发前后 88%和 12%的数据信息。

Signal Tap Logic Analyzer 默认选择前点触发，如图 3-74 所示。

(3) 触发级数(Trigger conditions)下拉列表用于设置触发级数，共有 10 级选项。在多级触发中，逻辑分析仪首先对第 1 级触发条件进行测试。当第 1 级触发表达式为真(TRUE)时，开始对第 2 级触发表达式进行测试。以此类推，直到完成所有触发级测试，并且最后一级触发条件测试结果为真时，逻辑分析仪开始捕获数据。

Signal Tap Logic Analyzer 默认的触发级数为 1，如图 3-74 所示。

(4) 选择触发信号和触发方式。若勾选 Trigger in 复选框，则应在 Source 栏中选择触发信号和触发电平，即当触发信号有效时，逻辑分析仪在采样时钟的作用下对待观测组中的信号进行单次或连续采样。

Signal Tap Logic Analyzer 默认不勾选 Trigger in 复选框，如图 3-74 所示。

5. 保存逻辑分析仪文件

执行 Signal Tap Logic Analyzer 主界面中 File→Save As 菜单命令，修改 Signal Tap Logic Analyzer 默认的文件名 stp1.stp 为 MUX4to1_tst.stp(.stp 为逻辑分析仪文件扩展名)，单击“保存”按钮，弹出如图 3-75 所示的添加逻辑分析仪文件提示框。单击 Yes 按钮，表示重新编译时将 Signal Tap Logic Analyzer 文件(MUX4to1_tst.stp)集成于工程中一起编辑、综合与适配，以便将逻辑分析仪 Signal Tap Logic Analyzer 同硬件电路一起配置到 FPGA 芯片中。

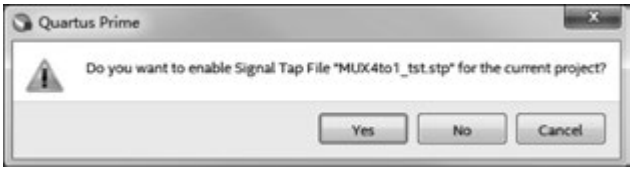


图 3-75 添加逻辑分析仪文件提示框

如果单击 No 按钮，则需要手动设置 Signal Tap Logic Analyzer。在 Quartus Prime 主界面中执行 Assignments→Settings 菜单命令，在弹出的 Settings 对话框的 Category 列表

中选择 Signal Tap Logic Analyzer，单击 Signal Tap File name 文本框右侧的浏览按钮 ...，在弹出的对话框中选择已经保存的逻辑分析仪文件（MUX4to1_tst. stp）添加到工程中，并勾选 Enable Signal Tap Logic Analyzer 复选框，如图 3-76 所示，单击 OK 按钮返回。

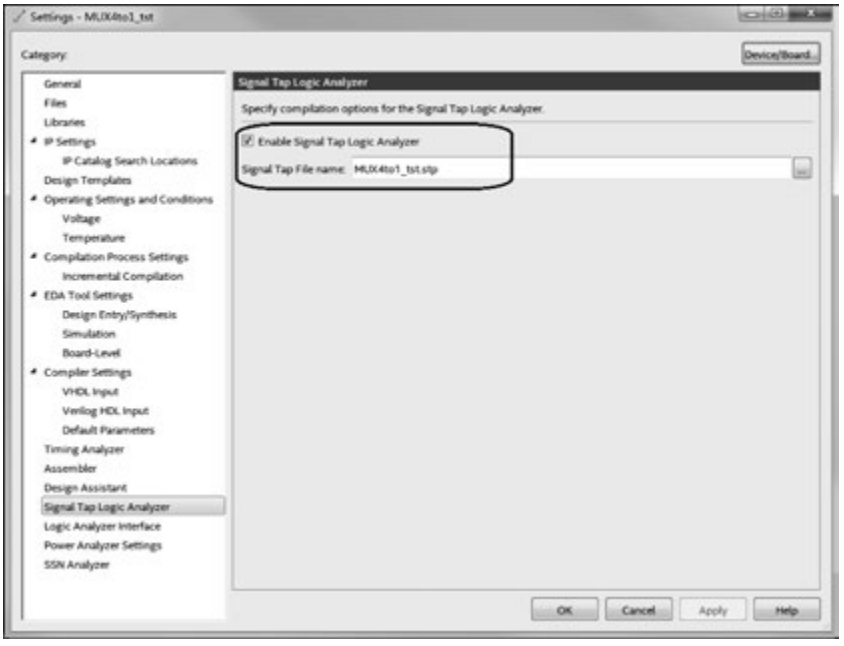


图 3-76 逻辑分析仪设置

6. 重新编译和配置

单击 ▶ 按钮对工程重新进行编译、综合与适配，将逻辑分析仪嵌入应用工程。

编译、综合与适配过程完成后，需要将 MUX4to1_tst. sof 文件配置到 FPGA 中。如果连接好硬件后没有自动检测到 USB-Blaster，则需要手动设置 USB-Blaster。单击图 3-77 中的 Setup 按钮，在 Hardware 下拉列表中选择 USB-Blaster [USB-0]。

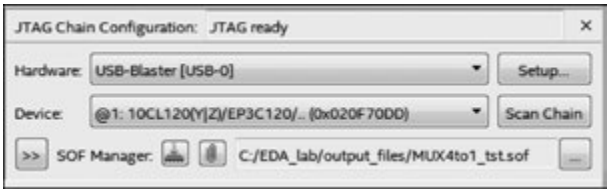


图 3-77 硬件连接设置

单击图 3-77 中的浏览按钮 ...，在目前工程目录 output_files 子目录下选中新生成的文件 MUX4to1_tst. sof，再单击 按钮进行配置。完成后，可以看到例化管理器 Instance Manager 右侧提示为 Ready to acquire，如图 3-78 所示，表示可以进行在线测试了。

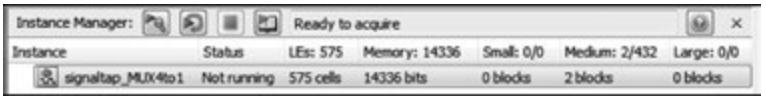


图 3-78 配置完成后的例化管理器窗口

7. 启动逻辑分析仪进行测试

设置开关 SW_1 和 SW_0 均为低电平(a＝00)，单击例化管理器中的按钮(或者执行 Processing→Run Analysis 菜单命令或按 F5 快捷键)启动 Signal Tap Logic Analyzer 进行单次数据采集，得到的信号波形如图 3-79 所示，可以看到输出 y 的波形与数据 d_0 的波形一致。

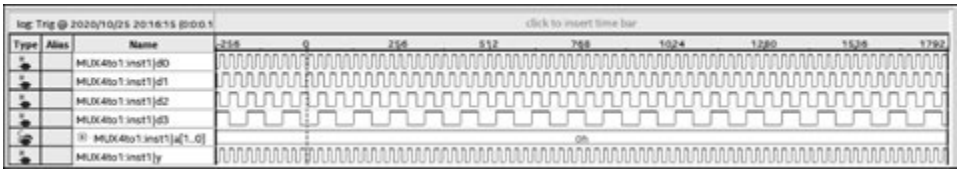


图 3-79　a＝00 时的测试波形图

分别设置开关 SW_1 和 SW_0 为 01、10 和 11，然后单击例化管理器中的按钮启动 Signal Tap Logic Analyzer 进行数据采集。得到的信号波形分别如图 3-80～图 3-82 所示，可以看到 4 选 1 数据选择器的实际工作情况与逻辑功能相同。

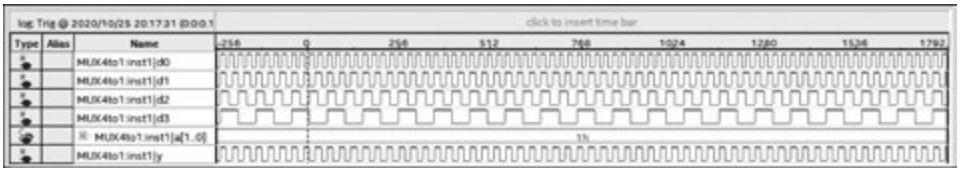


图 3-80　a＝01 时的测试波形图

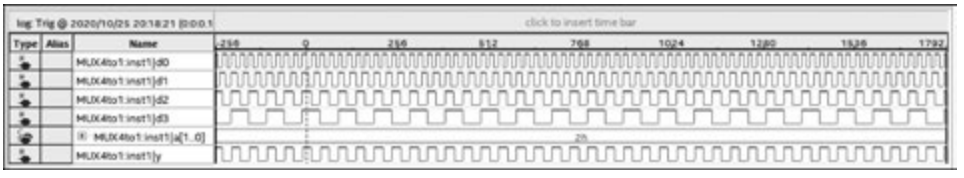


图 3-81　a＝10 时的测试波形图

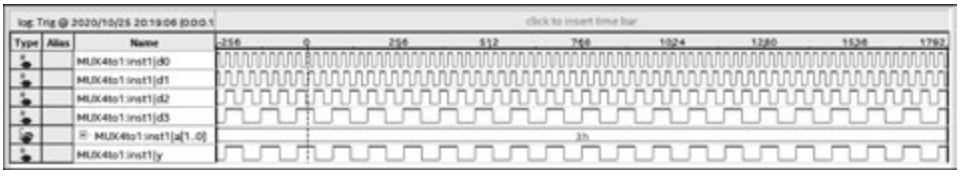


图 3-82　a＝11 时的测试波形图

另外，还可以单击例化管理器中的按钮(或者执行 Processing→AutoRun Analysis 菜单命令或按 F6 快捷键)进行连续测试，这时 Signal Tap Logic Analyzer 将实时采集和显示信号的波形，拨动 SW_1 和 SW_0 开关改变数据选择器的地址就可以观察到输出 y 的实时变化。单击按钮(或者执行 Processing→Stop Analysis 菜单命令或按 Esc 键)，Signal Tap Logic Analyzer 将停止数据采集。

测试完成后，需要从工程中移除 Signal Tap Logic Analyzer 时，在图 3-76 所示的设置界面中取消勾选 Enable Signal Tap Logic Analyzer 复选框，重新进行编译与综合后即可移除逻辑分析仪。

3.5 直接法频率计的设计——基于原理图方法

本节应用原理图方法设计直接法频率计，一方面是为了与数字电路课程相衔接，使只掌握了原理图设计方法的读者能够在电子技术课程设计和电子竞赛等实践环节中应用 EDA 技术设计中小规模数字系统；另一方面是为了与第 4 章应用 Verilog HDL 描述的方法进行比较，以体现应用 HDL 设计数字系统的优越性。

设计任务 设计能够测量 1Hz～100MHz 信号频率的数字频率计，应用 8 位数码管显示测频结果，要求测量误差不大于±1Hz。

分析 FPGA 内部逻辑单元的传输延迟很短，基于 FPGA 设计的频率计很容易测量 100MHz 及以上信号的频率(将在 7.3 节静态时序分析中进行验证)。要求测量误差不大于±1Hz 时，在闸门信号作用时间为 1s 的情况下，测量 1Hz～100MHz 信号的频率需要应用 10^8 进制计数器进行计数。

直接法频率计的原理电路如图 1-1 所示。当计数器带有计数允许控制端(如 74HC160 的 EP)时，原理电路中的与门可以省略。另外，为了能够连续测量信号的频率，还需要在原理电路的基础上设计主控电路，在时钟脉冲的作用下，循环对计数器清零、开启闸门和刷新显示 3 项任务。

能够实现连续测频的频率计总体设计方案如图 3-83 所示，其中 F_X 为被测信号。主控电路输出的 CLR′为清零信号，用于将计数器清零；CNTEN 为闸门信号，用于控制计数器在固定的时间范围内对 CLK 进行计数；DISPEN 为显示刷新信号，用于控制锁存译码电路刷新测量结果。如果被测信号为正弦波，那么还需要将被测信号 F_X 经过放大整形为脉冲信号后作为测频计数器的时钟。放大与整形电路属于模拟和模数混合电路，在此不再复述。

图 3-83 直接法频率计总体设计方案

设计过程 从设计方案可以看出，直接法频率计主要由主控电路、计数、锁存与显示译码电路和分频器构成。

1. 计数、锁存与显示译码电路设计

要求测量信号频率为 1Hz～100MHz 时，需要应用 10^8 进制计数器进行计数，驱动 8 位数码管显示频率值。10^8 进制计数器需要从 Quartus Prime 图形符号库中调用 8 个十进制计数器 74160 级联构成。

CD4511 是带有锁存功能的 BCD 显示译码器，因此锁存与译码电路基于 CD4511 设计最方便。但遗憾的是，Quartus Prime 提供的图形符号库中没有 CD4511，因此只能调用库


第 27 集
微课视频

中提供的 BCD 显示译码器 7448 设计。

由于 7448 没有锁存功能，为了能够锁存测量结果，还需要在每个计数器 74160 和显示译码器 7448 之间插入 4 位 D 锁存器（如 7475）以锁存需要显示的 BCD 码。另外，由于 DE2-115 开发板应用共阳数码管显示，因此基于 DE2-115 实现频率计时，还需要在 7448 的每个输出端加上反相器将高电平有效的驱动信号转换为低电平有效，以适应驱动共阳数码管的要求。

综上分析，BCD 码计数、锁存与显示译码电路如图 3-84 所示。

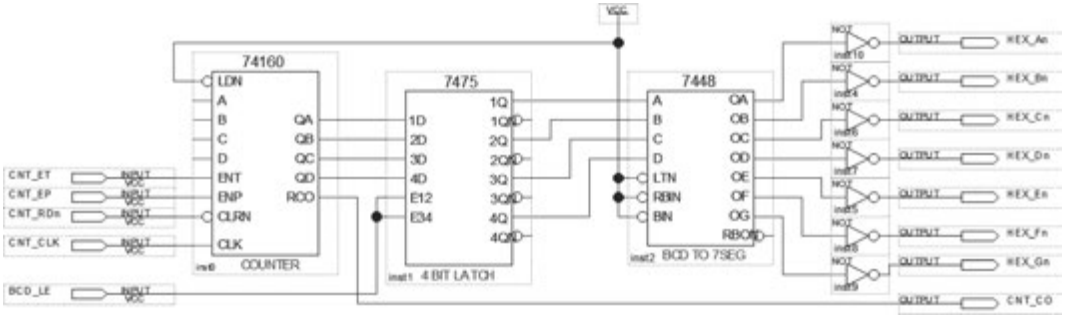


图 3-84　BCD 码计数、锁存与显示译码电路

为了使顶层设计电路简洁清晰，将图 3-84 所示的计数、锁存与显示译码电路通过 Files→Create／Update→Create Symbol Files for Current File 菜单命令封装成如图 3-85 所示的图形符号 BCD_CNT_LE_7SEG，以便在顶层设计电路中调用。

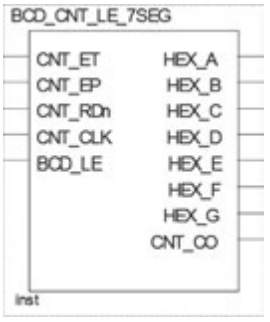


图 3-85　计数、锁存与显示译码模块图形符号

2. 主控电路设计

主控电路用于产生周期性的清零信号、闸门信号和显示刷新信号，以控制频率计的测频过程。

用十进制计数器 74160 作为主控器件，取时钟为 8Hz 时，测频计数器的清零信号 CLR′、闸门信号 CNTEN、显示刷新信号 DISPEN 与主控计数器的输出 $Q_3Q_2Q_1Q_0$ 之间的时序关系设计如表 3-5 所示。其中，清零信号的作用时间为 1/8s；闸门信号 CNTEN 的作用时间为 1s；显示刷新信号作用时间为 1/8s。

表 3-5　主控电路功能表

CLK	Q_3 Q_2 Q_1 Q_0	状态	CLR′	CNTEN	DISPEN
1	0 0 0 0	P_0	0	0	0
2	0 0 0 1	P_1	1	1	0
3	0 0 1 0	P_2	1	1	0
4	0 0 1 1	P_3	1	1	0
5	0 1 0 0	P_4	1	1	0
6	0 1 0 1	P_5	1	1	0
7	0 1 1 0	P_6	1	1	0
8	0 1 1 1	P_7	1	1	0
9	1 0 0 0	P_8	1	1	0
10	1 0 0 1	P_9	1	0	1

由表 3-5 可以写出 3 个控制信号的逻辑函数表达式，如下所示。

$$\begin{cases} \text{CLR}' = P'_0 = (Q'_3 Q'_2 Q'_1 Q'_0)' = Q_3 + Q_2 + Q_1 + Q_0 \\ \text{CNTEN} = (P_0 + P_9)' = \text{CLR}' \cdot \text{DISPEN}' \\ \text{DISPEN} = P_9 = C \end{cases}$$

其中，C 为计数器 74160 的进位信号。

由上述函数式设计出的主控电路如图 3-86 所示，其中，74175 用于将清零信号、闸门信号和显示信号锁存后输出，以避免组合逻辑输出所产生竞争-冒险现象，从而提高频率计工作的可靠性。

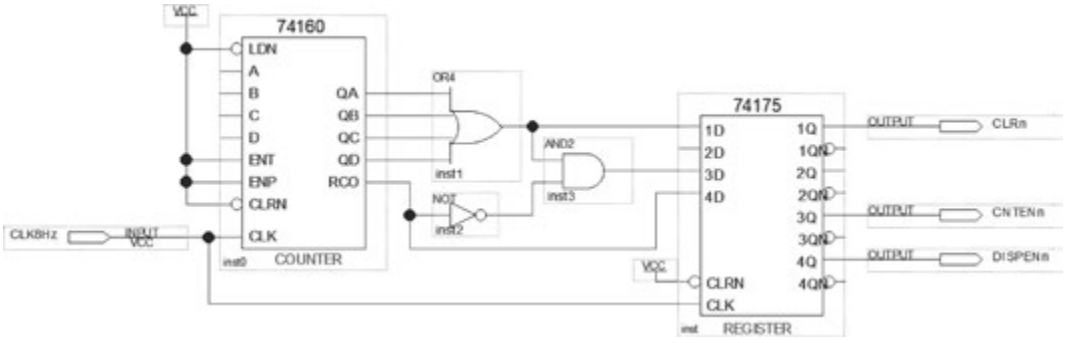


图 3-86 主控电路设计图

执行 Files→Create／Update→Create Symbol Files for Current File 菜单命令，将主控电路封装成如图 3-87 所示的图形符号 Freqer_CTRL，以便在顶层设计电路中调用。

3. 分频器设计

分频器用于为主控电路提供 8Hz 的时钟脉冲。

将 DE2-115 开发板 50MHz 晶振信号分频为 8Hz 时，需要应用分频系数为 $50\times10^6/8=6250000$ 的分频器。直接应用原理图设计如此大分频系数的分频器时，电路规模相当复杂。

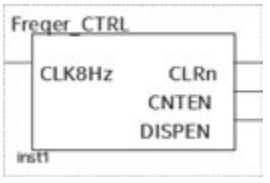


图 3-87 主控电路图形符号

为了简化分频电路设计，先定制锁相环（ALTPLL）将 50MHz 的晶振信号分频为 10kHz，然后再应用分频系数为 $10\times10^3/8=1250$ 的分频器将 10kHz 降为 8Hz。

锁相环的定制方法和具体步骤将在 5.2 节中讲述。

分频系数为 1250 的分频器应用 1250/2＝625 进制计数器级联一位二进制计数器实现，如图 3-88 所示。其中，625 进制计数器每经过 625 个脉冲输出一个进位信号，驱动一位二进制计器翻转输出 8Hz 方波。

执行 Files→Create／Update→Create Symbol Files for Current File 菜单命令，将图 3-88 所示的分频器封装成如图 3-89 所示的图形符号 FP10k_8Hz，以便在顶层设计电路中调用。

4. 顶层电路设计

频率计顶层电路是基于图 3-83 所示的设计方案，应用已经封装好的分频器，主控电路，计数、锁存与译码显示电路，以及定制的锁相环搭建而成，如图 3-90 所示。为了节约篇幅，顶层设计电路只连接了 4 组计数、锁存与显示译码模块，所以当闸门作用时间为 1s 时，能够测量信号的最高频率为 10kHz。将顶层电路中的计数、锁存与显示译码电路扩展为 6 组时，

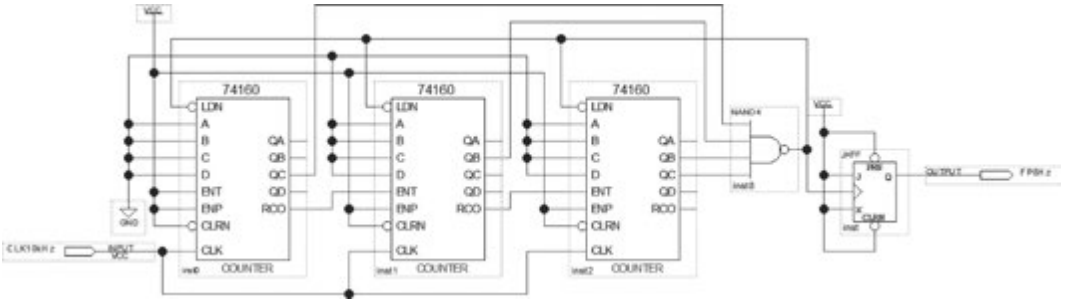


图 3-88　分频系数为 1250 的分频器

能够测量信号的最高频率为 1MHz，而设计测频范围为 1Hz～100MHz 的频率计需要将 4 组计数、锁存与显示译码电路扩展为 8 组才能实现。

图 3-89　分频器图形符号

需要说明的是，图 3-90 中的 D 触发器部分为超量程指示电路。当待测信号的频率超出测量范围时，D 触发器输出周期性脉冲，驱动外接的发光二极管 OVLED 闪烁，指示信号频率已经超出了测量范围。

另外，将测频信号的输入端(FX1_100MHz)连接到开发板 50MHz 晶振的输出端时，可以测试频率计功能的正确性。

对于应用数码管显示信息的多 I/O 端口数字系统，建议编写 Tcl 文件进行引脚锁定，以便可以在含有数码管的应用系统中重复使用。

驱动 8 位数码管显示信号频率的频率计 Tcl 文件内容参考如下。

```
# freqor_sch.tcl
# freqor pin setting
set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-STATED"
set_global_assignment -name ENABLE_INIT_DONE_OUTPUT OFF
set_location_assignment PIN_AG14 -to FX1_100MHz
set_location_assignment PIN_Y2 -to OSC50MHz
set_location_assignment PIN_AA14 -to D7_Gn
set_location_assignment PIN_AG18 -to D7_Fn
set_location_assignment PIN_AF17 -to D7_En
set_location_assignment PIN_AH17 -to D7_Dn
set_location_assignment PIN_AG17 -to D7_Cn
set_location_assignment PIN_AE17 -to D7_Bn
set_location_assignment PIN_AD17 -to D7_An
…
set_location_assignment PIN_H22 -to D0_Gn
set_location_assignment PIN_J22 -to D0_Fn
set_location_assignment PIN_L25 -to D0_En
set_location_assignment PIN_L26 -to D0_Dn
set_location_assignment PIN_E17 -to D0_Cn
set_location_assignment PIN_F22 -to D0_Bn
set_location_assignment PIN_G18 -to D0_An
```

Tcl 文件编写完成后，按 3.1.4 节所述方法完成频率计引脚的锁定。

将计数、锁存与显示译码电路扩展为 8 组(能够测量 100MHz 信号的频率)时，频率计顶层工程综合与适配的结果如图 3-91 所示。可以看出，频率计共使用了 733 个逻辑单元

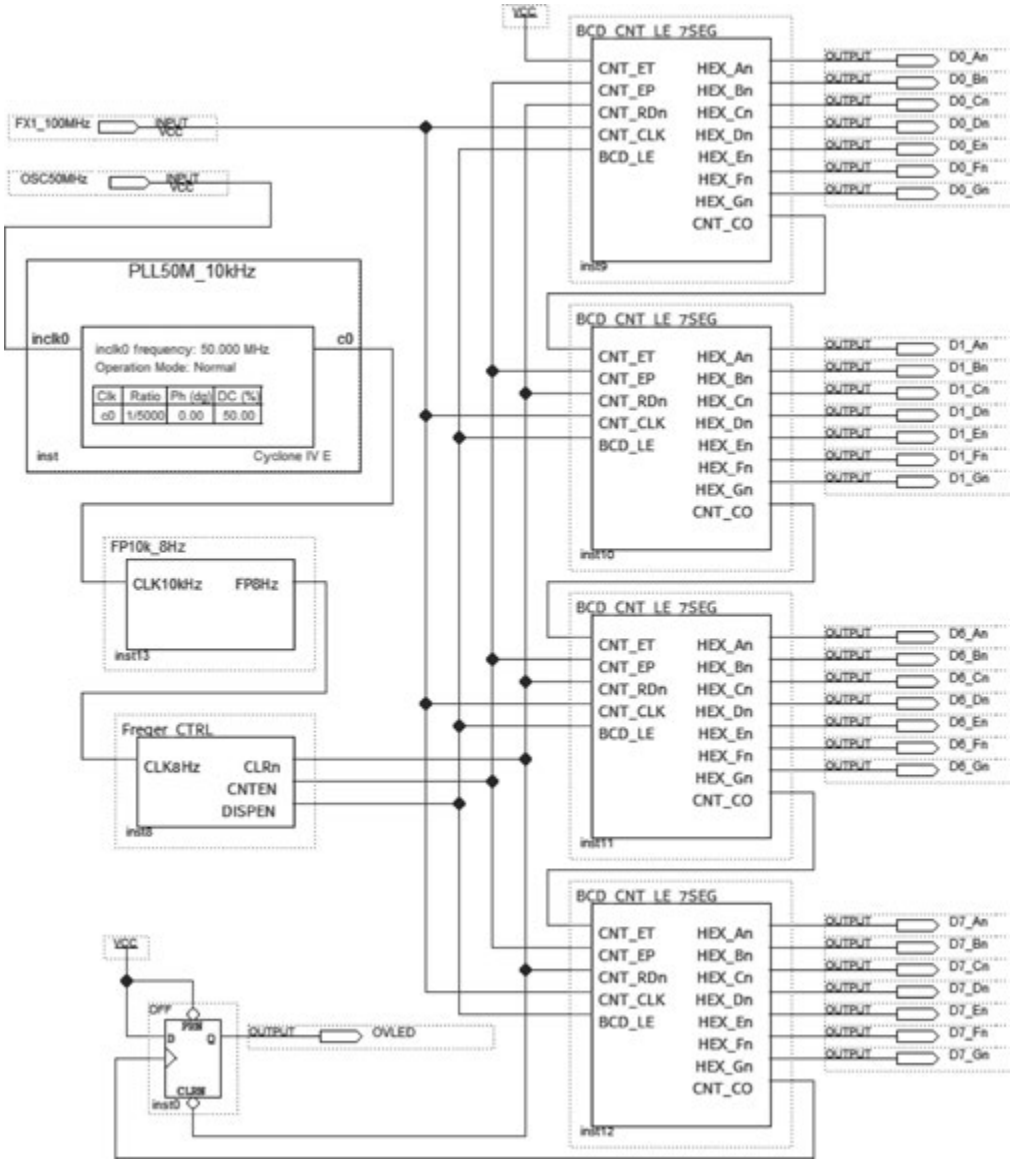


图 3-90 频率计顶层设计电路

Flow Status	Successful - Fri Apr 30 22:01:37 2021
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	Freqer8bcd_top
Top-level Entity Name	Freqer8bcd_top
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	733 / 114,480 (< 1 %)
Total registers	458
Total pins	59 / 529 (11 %)
Total virtual pins	0
Total memory bits	512 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	1 / 4 (25 %)

图 3-91 频率计综合与适配结果

(Logic Elements)、458 个寄存器(Registers)、512 位片内存储资源(Memory Bits)和一个锁相环(PLL)。

本章小结

本章讲述 Intel 公司的 EDA 综合开发环境 Quartus Prime 的基本应用，包括基本设计流程、原理图设计方法、仿真分析方法以及逻辑分析仪的应用。

Quartus Prime 能够完成建立工程，设计输入，编译、综合与适配，引脚锁定，以及编程与配置的全部设计流程，同时还可以根据需要进行仿真分析，以及在工程中嵌入逻辑分析仪进行在线测试。

Quartus Prime 支持硬件描述语言、原理图和状态机等多种输入方式，能够生成和识别 EDIF 网表文件和 HDL 网表文件，同时支持第三方工具软件，使设计者可以在设计流程的各个阶段能够根据需要选用更为专业的工具软件。

Verilog HDL 支持层次化设计方法。底层模块建议应用 HDL 描述，以方便模块功能的定义和重构。顶层设计电路既可以应用模块例化方式描述各模块之间的连接关系，也可以应用传统的原理图方法设计顶层电路。

受到 Quartus Prime 提供的原理图库中器件功能的限制，应用原理图设计数字系统既不利于优化设计，也不利于系统功能的重构。但是，原理图设计方法具有直观形象的优点。

仿真分析是通过计算机算法模拟代码的运行检查描述代码的逻辑是否正确，分为功能仿真和时序仿真两种类型。功能仿真是指在不考虑器件的传输延迟和布线延迟情况下的理想化仿真；而时序仿真是指在完成布局布线之后进行，包含了器件传输延迟和布线延迟信息的仿真。

Signal Tap Logic Analyzer 是集成于 Quartus Prime 开发环境中的逻辑分析仪。设计者可以将 Signal Tap Logic Analyzer 嵌入设计电路，然后一起配置到可编程逻辑器件中。Signal Tap Logic Analyzer 能够在电路工作期间实时捕获内部节点的信号或总线上的信息流，然后通过 JTAG 接口将采集到的数据传输给 Quartus Prime 开发环境显示电路内部节点的信号波形或总线上的数据信息，以供设计者进行分析。

思考与练习

3-1 简述在 Quartus Prime 开发环境下进行 EDA 设计的基本流程。

3-2 已知 4 线-16 线的译码器 74LS154 的两个功能端 S'_A 和 S'_B 均为低电平时，译码器正常工作，将输入的 4 位二进制码 $A_3A_2A_1A_0$ 翻译成低电平有效的输出信号 $Y'_0 \sim Y'_{15}$；否则译码器不工作，输出全部强制为高电平。

在 Quartus Prime 开发环境下应用 Verilog HDL 描述 74LS154 并进行仿真验证，然后下载到开发板进行功能测试。

3-3 已知 8 选 1 数据选择器 74HC151 的功能端 S' 为低电平时，数据选择器正常工作，根据地址码 $A_2A_1A_0$ 的不同从 8 路输入数据 $D_0 \sim D_7$ 中选择其中一路输出；否则数据选择器不工作，输出强制为低电平。

在 Quartus Prime 开发环境下应用 Verilog HDL 描述 74HC151 并进行仿真验证，然后下载到开发板进行功能测试。

3-4 应用原理图设计方法将两片十进制加/减计数器 74192 扩展为一百进制加/减计数器并进行仿真验证，然后下载到开发板进行功能测试。

3-5 在 Quartus Prime 开发环境下，应用 testbench 对例 2-26 中的 4 选 1 数据选择器进行仿真验证。

3-6 在 Quartus Prime 开发环境下，应用 testbench 对例 2-27 中的 4 位计数器进行仿真验证。

3-7 完成 3.5 节数字频率计的设计。下载到开发板进行功能测试。

3-8* 应用原理图方法设计数码序列控制电路。具体要求如下：

(1) 在单个数码管上依次显示自然数序列(0～9)、奇数序列(1、3、5、7 和 9)、音乐符号序列(0～7)和偶数序列(0、2、4、6 和 8)；

(2) 加电后先显示自然数序列，然后按上述规律循环显示。

在 Quartus Prime 开发环境下完成设计，并下载到 EDA 开发板进行功能测试。

3-9* 应用原理图方法设计洗衣机定时控制器。具体要求如下：

(1) 洗涤时间以分钟为单位，可以在 1～99min 任意设置；

(2) 用两位数码管显示洗涤剩余时间；

(3) 开机后按“停 10s→正转 20s→停 10s→反转 20s”的模式循环运转；

(4) 洗涤时间到后洗衣机停止工作，同时发出光电信号和音频信号提醒用户注意。

在 Quartus Prime 开发环境下进行设计，并下载到 EDA 开发板进行功能测试。

提示：用开发板上的数码管显示洗涤剩余时间，用 3 个 LED 表示洗衣机的工作状态。