

Quartus® II 简介

ALTERA®

Altera Corporation
101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com



Altera、Altera 标识、FastTrack、HardCopy、MAX、MAX+PLUS、MAX+PLUS II、MegaCore、MegaWizard、NativeLink、Nios、OpenCore、Quartus、Quartus II、Quartus II 标识和 SignalTap 是 Altera 公司在美国和其它国家的注册商标。Avalon、ByteBlaster、ByteBlasterMV、Cyclone、Excalibur、IP MegaStore、Jam、LogicLock、MasterBlaster、MegaLAB、PowerFit、SignalProbe、Stratix 和 USB-Blaster 是 Altera 公司在美国和其它国家的商标以及服务标志。Altera 公司使用的产品设计单元和助记符受版权法以及商标法的保护。

Altera 公司承认本文档提及的其它组织的产品或商标以及服务标志，特别是：ARM 是注册商标，AMBA 是 ARM 公司的商标。Mentor Graphics 和 ModelSim 是 Mentor Graphics 公司的注册商标。

Altera 保留更改本文档中所述器件或器件规范的权利，恕不另行通知。Altera 建议客户在下订单之前取得器件规范的最新版本，以确认您所获得的信息是最新的。Altera 保证自己的半导体产品性能符合当前规范，与 Altera 的标准担保一致。Altera 使用了必要的测试和其它质量控制技术，足以支持此担保。除非政府管制要求，否则没有必要对每个设备的所有参数都进行具体测试。如果没有书面协议另做规定，Altera 对于使用本文档所述半导体器件而引起有关的 Altera 应用协助、客户产品设计或对第三方专利或版权的侵权均不承担任何责任。Altera 不保证或代表 Altera 涵盖或相关的可能使用或正在使用此类半导体器件在内的任何组合、机械或过程的任何专利权、版权或其它知识产权。

没有 Altera 公司总裁以书面形式明确同意，Altera 产品不得用作生命支持器件或系统中的关键元件。其中：

1. 生命支持器件或系统是指这类器件或系统：(a) 用于外科手术植入人体内；(b) 支持或维持生命，而且当这种器件或系统在按照标签上提供的使用说明正确使用时，却无法发挥正常功效时，可能会对用户造成严重伤害。
2. 关键元件是指生命支持设备或系统的任何元件，如果不能正常发挥功效，可能会导致生命支持器件或系统出现故障，或影响它的安全性或有效性。

Altera 产品受多种美国和国外专利以及未决专利、外观权利和版权的保护。

版权所有 © 2005 Altera 公司。保留所有权利。



I.S. EN ISO 9001

目 录

前言	ix
文档编制约定	xi
第 1 章：设计流程	1
简介	2
图形用户界面设计流程	3
EDA 工具设计流程	10
命令行设计流程	15
命令行可执行文件	16
使用标准命令行命令和脚本	20
使用 Tcl 命令	22
建立 Makefile 脚本	25
设计方法和设计规划	27
自上而下与自下而上的设计方法比较	27
自上而下渐进式编译设计流程	28
自下而上基于 LogicLock 的设计流程	29
第 2 章：设计输入	31
简介	32
建立工程	33
使用修订	34
使用版本兼容的数据库	37
转换 MAX+PLUS II 工程	38
建立设计	39
使用 Quartus II Block Editor	40
使用 Quartus II Text Editor	41
使用 Quartus II Symbol Editor	42
使用 Verilog HDL、VHDL 与 AHDL	42
使用 Altera 宏功能模块	43
使用知识产权 (IP) 宏功能模块	44
使用 MegaWizard Plug-In Manager	45
在 Quartus II 软件中例化宏功能模块	46
在 Verilog HDL 和 VHDL 中例化	46
使用端口和参数定义	47
推断宏功能模块	47
在 EDA 工具中例化宏功能模块	47
使用 Black-Box 方法	47
按推断进行例化	48
使用 Clear-Box 方法	48
第 3 章：约束输入	51
简介	52
使用 Assignment Editor	52
使用 Pin Planner	54
使用 Settings 对话框	56
分配设计分区	57

在 Project Navigator 中分配设计分区	57
使用 Design 使用 Partitions 窗口分配设计分区	58
导入分配	59
验证引脚分配	60
第 4 章：综合	61
简介	62
使用 Quartus II Verilog HDL & VHDL Integrated Synthesis	63
使用其它 EDA 综合工具	66
控制 Analysis & Synthesis	68
使用 Complier 指令和属性	68
使用 Quartus II 逻辑选项	69
使用 Quartus II 综合网表优化选项	71
使用 Design Assistant 检查设计可靠性	71
使用 RTL Viewer 分析综合结果	73
采用 Technology Map Viewer 分析综合结果	77
进行渐进式综合	78
第 5 章：布局布线	81
简介	82
进行完整的渐进式编译	83
分析适配结果	84
使用 Messages 窗口查看适配结果	84
使用 Report 窗口或 Report 文件查看适配结果	86
使用 Timing Closure Floorplan 分析结果	87
使用 Design Assistant 检查设计的可靠性	89
优化适配	89
使用位置分配	90
设置用于控制布局布线的选项	90
设置 Fitter 选项	90
设置物理综合优化选项	91
设置影响布局布线的个别逻辑选项	91
使用 Resource Optimization Advisor	92
使用 Design Space Explorer	94
通过反标保留分配	98
第 6 章：基于模块的设计	101
简介	102
Quartus II 基于模块的设计流程	102
使用 LogicLock 区域	103
在自上而下渐进式编译流程中使用 LogicLock 区域	107
保存自下而上 LogicLock 流程的中间综合结果	108
反标 LogicLock 区域分配	109
导出与导入 LogicLock 分配	109
LogicLock 与 EDA 工具结合使用	111

第 7 章：仿真.....	113
简介.....	114
使用 EDA 工具进行设计仿真	115
进行 EDA 仿真工具设置.....	116
生成仿真输出文件	117
EDA 仿真流程.....	118
功能仿真流程	118
NativeLink 仿真流程	118
手动时序仿真流程	119
仿真库	119
使用 Quartus II Simulator 进行仿真设计.....	121
建立波形文件	123
使用 Simulator Tool.....	124
第 8 章：时序分析.....	125
简介.....	126
在 Quartus II 软件中进行时序分析	127
指定时序要求	127
进行工程全局范围的时序设置	129
进行个别时序分配	130
进行时序分析	131
进行早期时序估算	133
查看时序分析结果	135
使用报告窗口	135
进行分配与查看延时路径	136
使用 Technology Map Viewer.....	138
使用 EDA 工具进行时序分析	139
使用 PrimeTime 软件	141
使用 Tau 软件.....	141
第 9 章：时序逼近.....	143
简介.....	144
使用 Timing Closure Floorplan	144
查看分配与布线.....	145
进行分配	146
使用 Timing Optimization Advisor	147
使用网表优化实现时序逼近	148
使用 LogicLock 区域达到时序逼近	150
软 LogicLock 区域	151
基于路径的分配.....	151
使用 Design Space Explorer 达到时序逼近	153
使用渐进式编译达到时序逼近	153
第 10 章：功耗分析	155
简介.....	156
使用 PowerPlay Power Analyzer 分析功耗.....	156

指定 Power Analyzer 选项	158
使用 PowerPlay Early Power Estimator.....	160
第 11 章：编程和配置	163
简介	164
使用 Programmer 对一个或多个器件编程	167
建立辅助编程文件	168
建立其它编程文件格式	169
转换编程文件	171
使用 Quartus II 软件通过远程 JTAG 服务器进行编程	174
第 12 章：调试.....	175
简介	176
使用 SignalTap II Logic Analyzer.....	177
设置和运行 SignalTap II Logic Analyzer	177
渐进式编译使用 SignalTap II Logic Analyzer	181
分析 SignalTap II 数据	182
使用 SignalProbe	184
使用 In-System Memory Content Editor	187
使用 RTL Viewer 和 Technology Map Viewer.....	189
使用 Chip Editor.....	190
第 13 章：工程更改管理	191
简介	192
使用 Chip Editor 识别延时与关键路径	193
在 Chip Editor 中编辑基元	194
使用 Resource Property Editor 修改资源属性	194
使用 Change Manager 查看和管理更改	196
验证 ECO 更改的效果	198
第 14 章：形式验证	199
简介	200
使用 EDA 形式验证工具	201
指定其他设置	203
第 15 章：系统级设计	205
简介	206
使用 SOPC Builder 建立 SOPC 设计	208
建立系统	208
生成系统	209
使用 DSP Builder 建立 DSP 设计	210
例化功能	210
生成仿真文件	210
生成综合文件	211
第 16 章：软件开发	213
简介	214

在 Quartus II 软件中使用 Software Builder	214
进行软件构建设置	215
生成软件输出文件	215
生成闪存编程文件	216
生成被动编程文件	217
生成存储器初始化数据文件	219
 第 17 章：安装、许可和技术支持	221
安装 Quartus II 软件	222
许可 Quartus II 软件	222
获取技术支持	224
 第 18 章：文档和其他资源	227
获取在线帮助	228
使用 Quartus II 在线教程	229
其它 Quartus II 软件文档	230
其他 Altera 文献	230
索引	233

前言

Altera® Quartus® II 设计软件为可编程芯片系统 (SOPC) 提供最全面的设计环境。如果您以前使用 MAX+PLUS® II 软件、其它设计软件或 ASIC 设计软件，现在准备改用 Quartus II 软件，您对 Quartus II 软件有一些了解但想进一步了解它的功能，那么本手册非常适合您使用。

本手册针对的读者是 Quartus II 软件初学者，它概述了可编程逻辑设计中 Quartus II 软件的功能。不过，本手册并不是 Quartus II 软件的详尽参考手册。相反，本手册只是一本指导书，它解释软件的功能以及这些功能如何帮助您进行 FPGA 和 CPLD 设计。本手册按一系列特定的可编程逻辑设计任务来组织内容。无论是使用 Quartus II 图形用户界面、其它 EDA 工具还是 Quartus II 命令行界面，本手册都将为您介绍最适合设计流程的功能。

第一章概述主要的图形用户界面、EDA 工具和命令行界面设计流程。接下来每一章开头都介绍该章具体目的，并对每个任务流加以概述，描述如何将 Quartus II 软件与现有 EDA 工具和命令行设计流程集成在一起。另外，手册还推荐了能够帮助您使用 Quartus II 软件的其它资源，如 Quartus II 在线帮助和 Quartus II 在线教程、应用笔记、白皮书以及 Altera 网站提供的其它文档和资源。

跟随本手册学习 Quartus II 软件，您可以了解此软件如何帮助您提高效率并缩短设计周期，如何与现有可编程逻辑设计流程集成以及如何快速有效地达到设计、性能和时序要求。

文档编制约定

Quartus® II 简介手册采用以下约定，方便您查找和理解信息。

印刷约定

Quartus II 文档使用下表所示的印刷约定：

视觉提示	含义
粗体首字母大写	命令名称；对话框、页面和选项标签标题；按钮名称以粗体显示，且首字母大写。例如：Find Text 命令, Save As 对话框以及 Start 按钮。
粗体	目录名称、工程名称、磁盘驱动器名称、文件名称、文件名扩展、软件应用程序名称、软件可执行文件名称以及对话框中的选项以粗体显示。例如 :quartus 目录, d: 驱动器, license.dat 文件。
首字母大写	键盘键、用户可编辑的应用程序窗口域和菜单名称以首字母大写。例如：Delete 键, Options 菜单。
“副标题”	手册章节中的副标题加有引号。在手册中，帮助主题的标题也加有引号。
斜体首字母大写	帮助类别、手册标题、手册中的章节标题、应用笔记和简短名称以斜体显示，且首字母大写。例如：FLEXIm End Users Guide。
斜体	变量放在尖括号 (< >) 内且以斜体显示。例如：< 文件名称 >, < CD-ROM 驱动器 >。
Courier 字体	必须和显示内容完全一样的键入内容以 Courier 字体显示。例如 :\quartus\bin\lmulti lmhostid。
←	Enter 或回车键。
◇	当条目的顺序不重要时，在条目列表中使用该符号。
脚印	脚印符号指示去哪里获取某特定主题的详细信息。
✓	对号标记指示仅由一个步骤构成的过程。
手形	手形表示需要特别注意的信息。

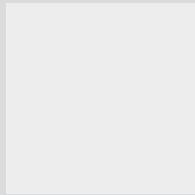
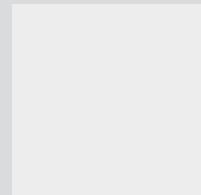
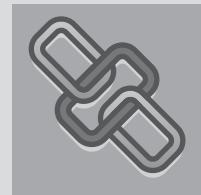
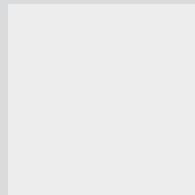
术语

在 *Quartus II* 简介手册中使用下表所示术语：

术语	含义
“单击”	表示快速按下并释放鼠标左键。
“双击”	表示连续两次快速的单击。
“选择”	表示需要使用鼠标或键组合启动一个操作。
“选定”	表示需要使用键组合或鼠标高亮显示文本以及对象或对话框中的选项。选定不会启动操作。例如：选定 Chain Description File，然后单击 OK。
“打开” / “关闭”	表示必须单击复选框以打开或关闭一个功能。

第一章

设计流程



第1章 内容:

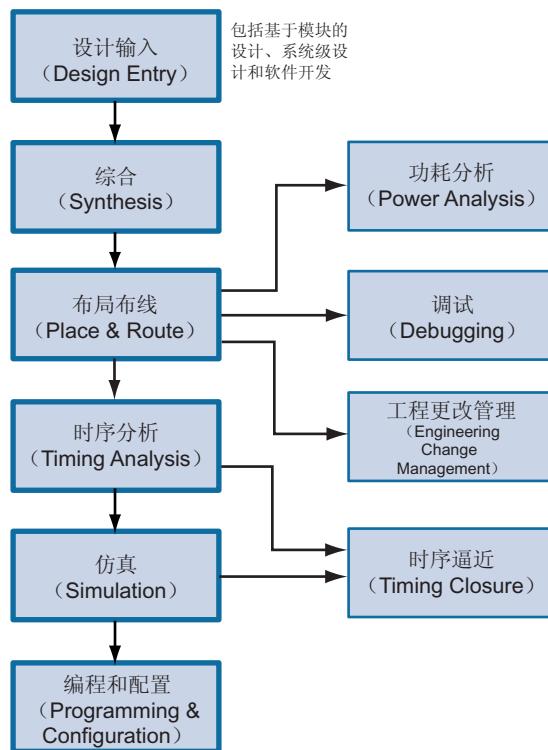
简介	2
图形用户界面设计流程	3
EDA 工具设计流程	10
命令行设计流程	15
设计方法和设计规划	27

1

简介

Altera® Quartus®II 设计软件提供完整的多平台设计环境，能够直接满足特定设计需要，为可编程芯片系统 (SOPC) 提供全面的设计环境。QuartusII 软件含有 FPGA 和 CPLD 设计所有阶段的解决方案。有关 Quartus II 设计流程的图示说明，请参见 图 1。

图 1. Quartus II 设计流程



此外，Quartus II 软件为设计流程的每个阶段提供 Quartus II 图形用户界面、EDA 工具界面以及命令行界面。可以在整个流程中只使用这些界面中的一个，也可以在设计流程的不同阶段使用不同界面。本章介绍适用于每个设计流程的界面和设计方法。本手册其余章节详细说明设计流程的各个阶段。

图形用户界面设计流程

您可以使用 Quartus II 软件完成设计流程的所有阶段；它是一个全面的、易于使用的独立解决方案。图 2 显示 Quartus II 图形用户界面为设计流程每个阶段所提供的功能。

图 2. Quartus II 图形用户界面的功能

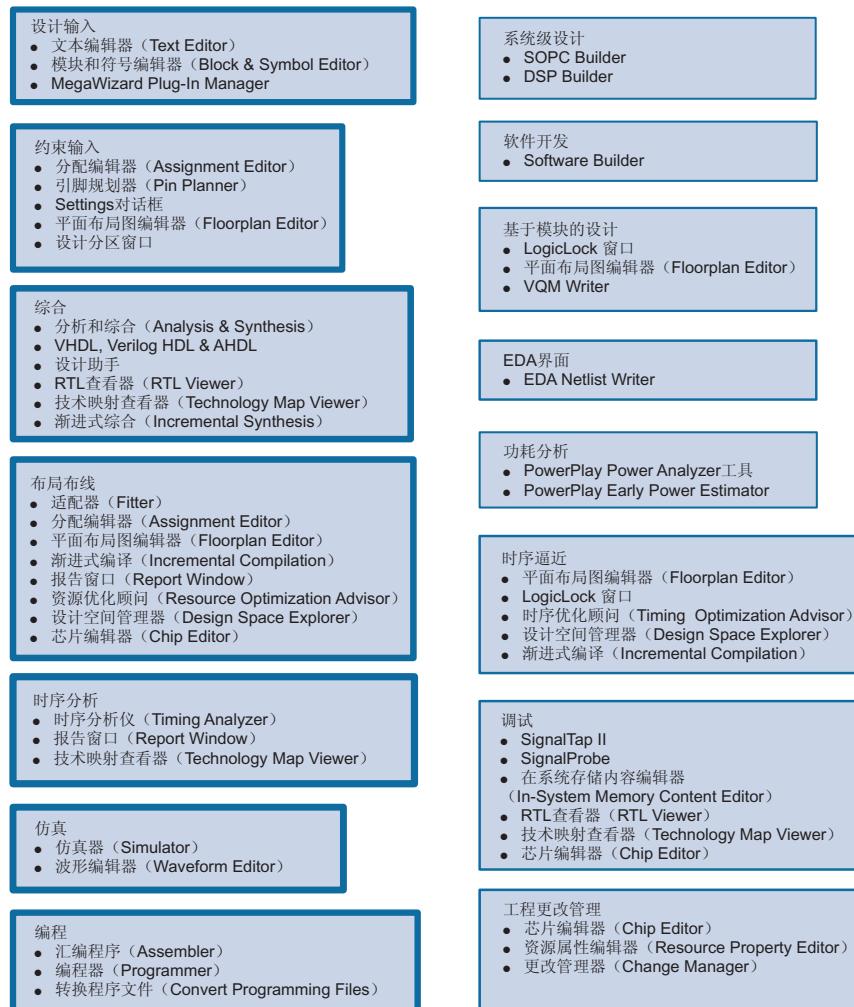
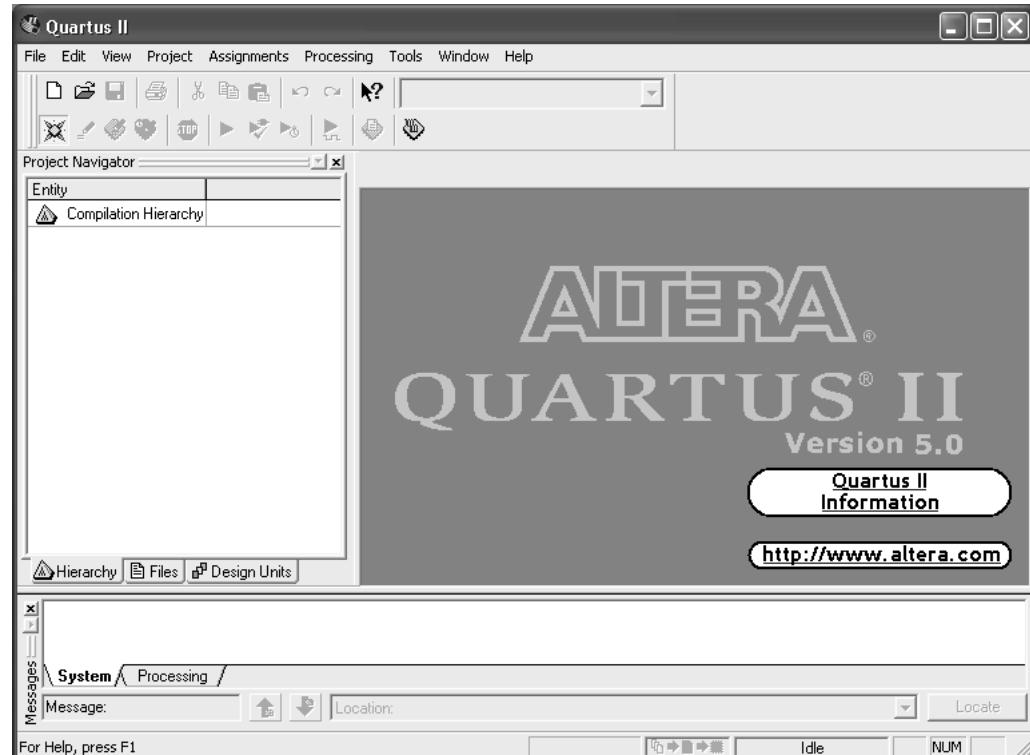


图 3 显示首次启动 Quartus II 软件时出现的 Quartus II 图形用户界面。

图 3. Quartus II 图形用户界面



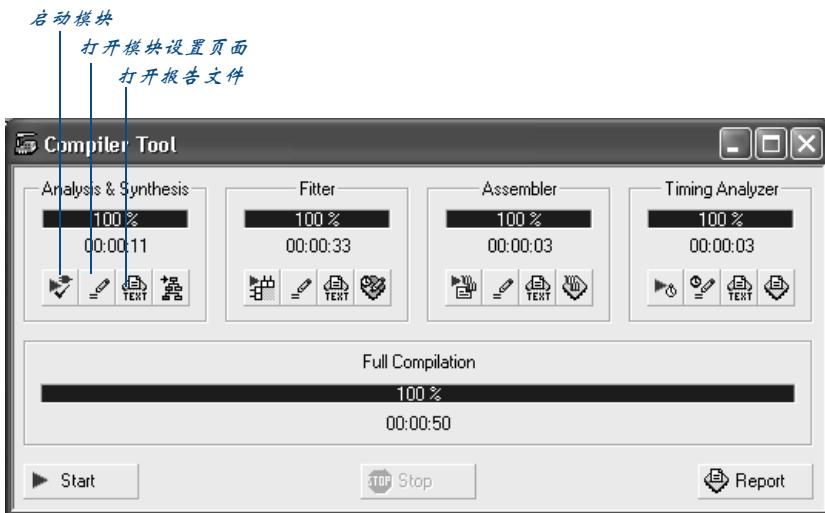
Quartus II 软件包括一个模块化 Compiler。Compiler 包括以下模块（标有星号的模块表示根据设置，在完整编译时可选择使用）：

- ◇ Analysis & Synthesis(分析和综合)
- ◇ Partition Merge*(分区合并)
- ◇ Fitter(适配器)
- ◇ Assembler*(汇编器)
- ◇ Timing Analyzer*(时序分析仪)
- ◇ Design Assistant*(设计助手)
- ◇ EDA Netlist Writer*(EDA 网表编写程序)
- ◇ HardCopy®Netlist Writer*(HardCopy 网表编写程序)

所有的 Compiler 模块可以做为完整编译的一部分，通过选择 Start Compilation (Processing 菜单) 来运行。若要单独运行各个模块，可以选择 Start (Processing 菜单)，然后从 Start 子菜单中为模块选择相应的指令。还可以逐步运行一些 Compiler 模块。参见第 28 页 “自上而下渐进式编译设计流程”，了解更详细的信息。

此外，还可以通过选择 **Compiler Tool** (**Tools** 菜单)，并在 **Compiler Tool** 窗口中运行该模块来启动 **Compiler** 模块。在 **Compiler Tool** 窗口中，可以打开该模块的设置文件或报告文件，还可以打开其它相关窗口。请参见图 4。

图 4. Compiler Tool 窗口



Quartus II 软件也提供一些预定义的编译流程，您可以利用 Processing 菜单中的命令来使用这些流程。表 1 列出一些最常用编译流程的命令。

表 1. 常用编译流程命令（第 1 部分，共 2 部分）

流程	说明	Processing 菜单的 Quartus II 命令
完整编译流程	进行当前设计的完整编译。	Start Compilation 命令
编译和仿真流程	如果仿真模式为时序仿真，流程执行完整编译，然后仿真当前设计。如果仿真模式为功能仿真，则流程只执行 Generate Functional Simulation Netlist 命令，然后对当前设计进行仿真。	Start Compilation and Simulation 命令
SignalProbe™ 流程	在不影响设计中现有适配的情况下，将用户指定的信号引至输出引脚，无需进行完整编译，就可以调试信号。	Start > Start SignalProbe Compilation 命令

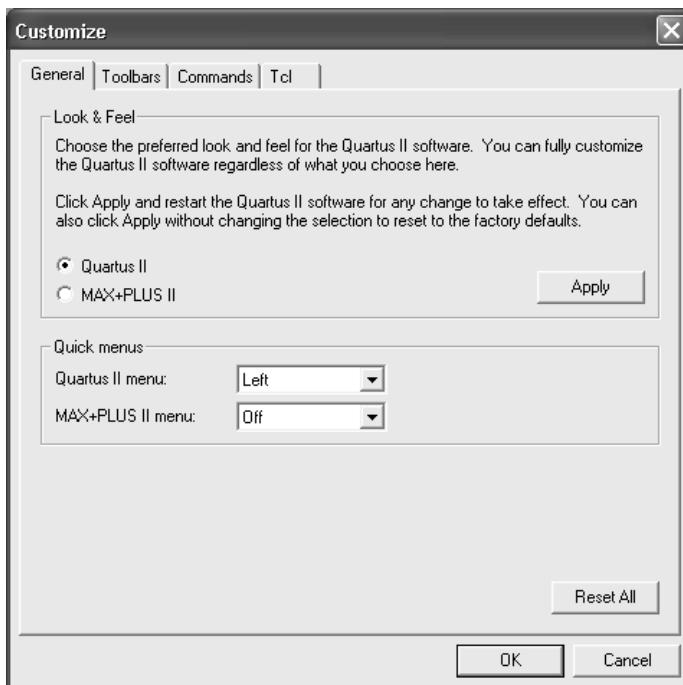
表1. 常用编译流程命令（第2部分，共2部分）

流程	说明	Processing 菜单的 Quartus II 命令
早期时序估算	进行部分编译，但是在 Fitter 完成之前停止并生成早期时序估算。	Start > Start Early Timing Estimate 命令
分区合并	在渐进式综合之后合并设计分区，建立单一网表，在后续编译阶段使用。如果采用渐进式综合，或者在修改设计之后，没有重新完整编译设计，则必须运行此命令。	Start > Start Partition Merge 命令



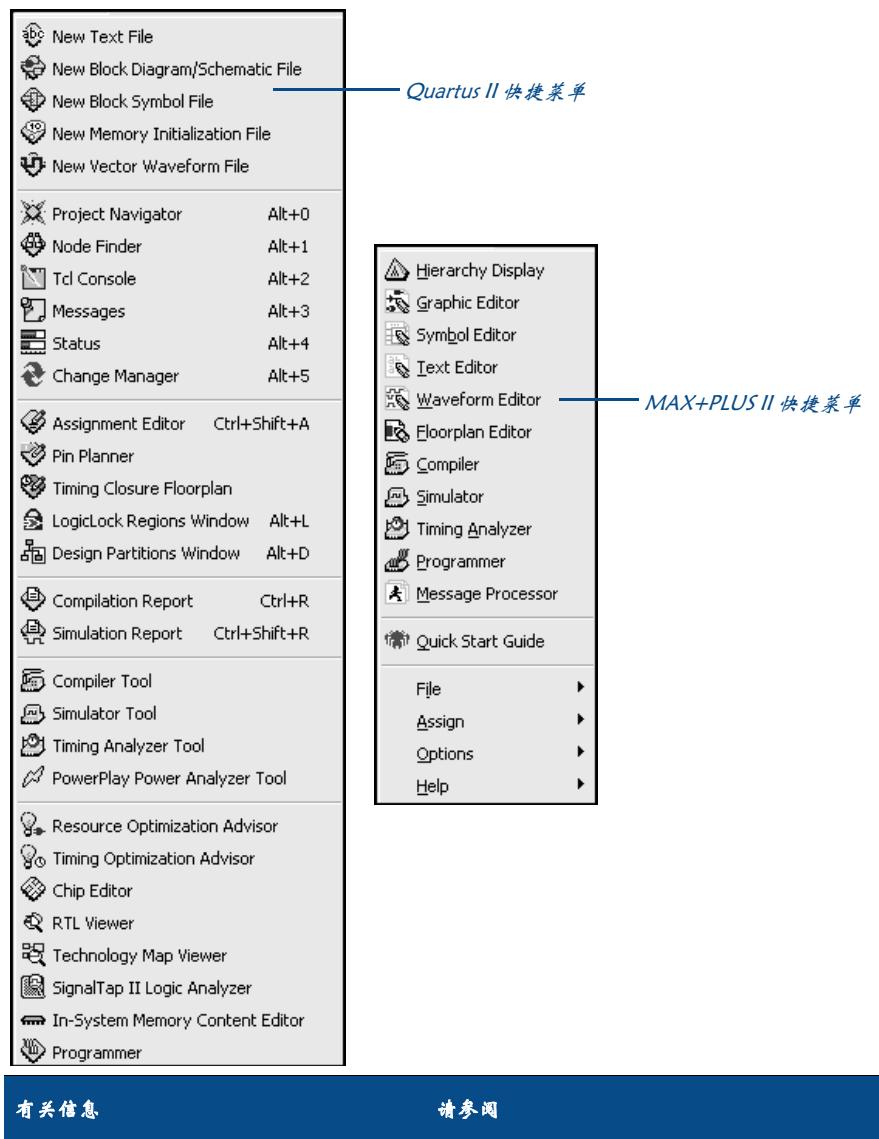
您可以按照个人偏好，自定义 Quartus II 软件的布局、菜单、命令和图标。您第一次打开 Quartus II 软件时，可以在标准 Quartus II 用户界面和 MAX+PLUS®II 界面之间选择外观和操作习惯，或者稍候使用 **Customize** 对话框 (Tools 菜单) 选择外观和操作习惯。如果您先前使用 MAX+PLUS II 软件，MAX+PLUS II 的外观和操作习惯设置允许您使用熟悉的 MAX+PLUS II 布局、命令和图标以便控制 Quartus II 软件的功能。图 5 显示了 **Customize** 对话框。

图 5. Customize 对话框



Customize 对话框也允许您设置是否显示 Quartus II 或者 MAX+PLUS II 快捷菜单，是设置在菜单栏的左边还是右边。Quartus II 快捷菜单包含的菜单命令用于每个 Quartus II 应用程序，是常用处理命令。MAX+PLUS II 快捷菜单与 MAX+PLUS II 软件的 MAX+PLUS II 菜单类似，提供用于应用程序的命令，也是常用的 MAX+PLUS II 菜单命令。MAX+PLUS II 菜单上的命令与相应的 Quartus II 命令功能相同。图 6 显示了 Quartus II 和 MAX+PLUS II 快捷菜单。

图 6. Quartus II 和 MAX+PLUS II 快捷菜单



有关信息

MAX+PLUS II 用户使用 Quartus II 设计流程

请参阅

Altera 网站上 *Quartus II Handbook 第 1 卷*
“*Quartus II Design Flow for MAX+PLUS II Users*”。

定制用户界面

Quartus II Tutorial 的 MAX+PLUS II Conversion 部分

Quartus II Help 中的 “Overview: Working With the User Interface”



有关信息

请参阅

使用 MAX+PLUS II 外观和操作习惯

Quartus II Help 中的“MAX+PLUS II Quick Start Guide for the Quartus II Software”以及“List of MAX+PLUS II Commands”

以下步骤描述了使用 Quartus II 图形用户界面的基本设计流程：

1. 使用 New Project Wizard (File 菜单) 建立新工程并指定目标器件或器件系列。
2. 使用 Text Editor 建立 Verilog HDL、VHDL 或 Altera 硬件描述语言 (AHDL) 设计。根据需要，使用 Block Editor 建立表示其他设计文件的符号框图，也可以建立原理图。还可以使用 MegaWizard® Plug-In Manager (Tools 菜单) 生成宏功能模块和 IP 功能的自定义变量，在设计中将它们例化。
3. (可选) 使用 Assignment Editor、Pin Planner、Settings 对话框 (Assignments 菜单)、Floorplan Editor、Design Partitions 窗口、LogicLock™ 功能指定初始设计约束。
4. (可选) 进行 Early Timing Estimate，在完成 Fitter 之前生成时序结果的早期估算。
5. (可选) 使用 SOPC Builder 或 DSP Builder 建立系统级设计。
6. (可选) 使用 Software Builder 为 Excalibur™ 器件处理器或 Nios® 嵌入式处理器建立软件和编程文件。
7. 使用 Analysis & Synthesis 对设计进行综合。
8. (可选) 如果设计含有分区，而没有进行完整编译，则需要采用 Partition Merge 合并分区。
9. (可选) 通过使用 Simulator 和 Generate Functional Simulation Netlist 命令在设计中执行功能仿真。
10. 使用 Fitter 对设计进行布局布线。
11. 使用 PowerPlay Power Analyzer 进行功耗估算和分析。
12. 使用 Timing Analyzer 对设计进行时序分析。

13. 使用 Simulator 对设计进行时序仿真。
14. (可选) 使用物理综合、Timing Closure 平面布局图、LogicLock 功能、**Settings** 对话框和 Assignment Editor 改进时序，达到时序逼近。
15. 使用 Assembler 为设计建立编程文件。
16. 使用编程文件、Programmer 和 Altera 硬件对器件进行编程；或将编程文件转换为其它文件格式以供嵌入式处理器等其它系统使用。
17. (可选) 使用 SignalTap® II Logic Analyzer、SignalProbe™ 功能或 Chip Editor 对设计进行调试。
18. (可选) 使用 Chip Editor、Resource Property Editor 和 Change Manager 管理工程更改。

EDA 工具设计流程

Quartus II 软件允许在设计流程的不同阶段使用您熟悉的 EDA 工具。可以与 Quartus II 图形用户界面或者 Quartus II 命令行可执行文件一起使用这些工具。图 7 显示了 EDA 工具设计流程。



图 7. EDA 工具设计流程

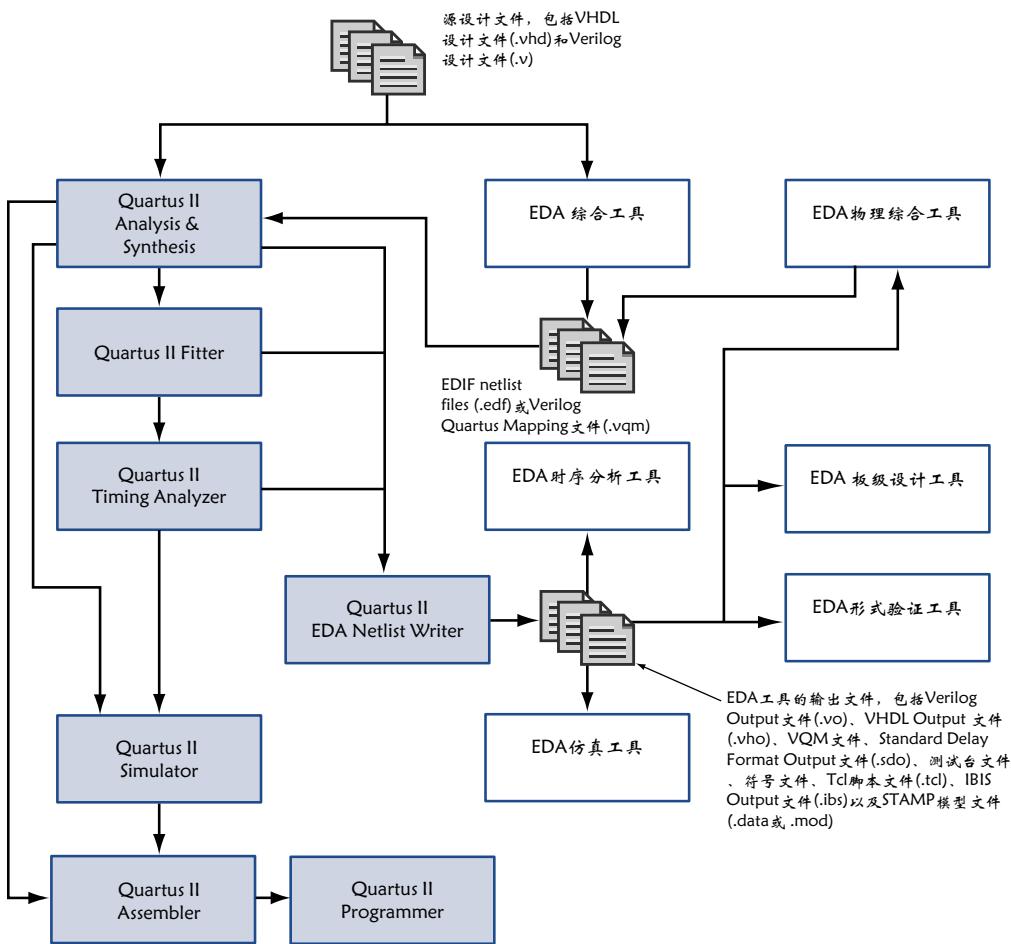


表 2 是 Quartus II 软件支持的 EDA 工具，并指出哪个 EDA 工具有 NativeLink® 支持。NativeLink 技术在 Quartus II 软件和其它 EDA 工具之间实现信息的无缝传递，并允许在 Quartus II 软件中自动运行 EDA 工具。

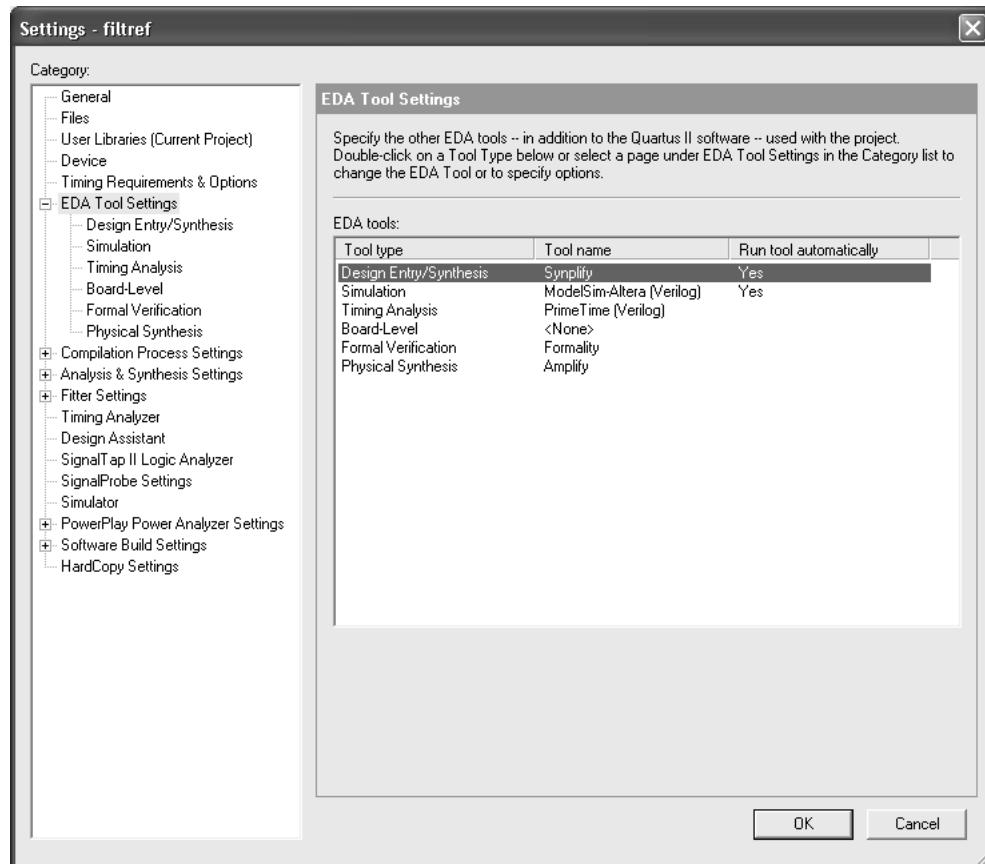
表 2. Quartus II 软件支持的 EDA 工具（第 1 部分，共 2 部分）

功能	支持的 EDA 工具	NativeLink 支持
设计输入和综合	Mentor Graphics Design Architect	
	Mentor Graphics LeonardoSpectrum	✓
	Mentor Graphics Precision RTL Synthesis	✓
	Mentor Graphics ViewDraw	
	Synopsys Design Compiler	
	Synopsys Design Compiler FPGA	
	Synopsys FPGA Compiler II	✓
	Synplicity Synplify	✓
	Synplicity Synplify Pro	✓
仿真	Cadence NC-Verilog	✓
	Cadence NC-VHDL	✓
	Cadence Verilog-XL	
	Mentor Graphics ModelSim	✓
	Mentor Graphics ModelSim-Altera	✓
	Synopsys VCS MX	✓
	Synopsys VCS	✓
	Synopsys VSS	
时序分析	Mentor Graphics Tau (through Stamp)	
	Synopsys PrimeTime	✓
板级设计	Hyperlynx (through Signal Integrity IBIS)	
	XTK (through Signal Integrity IBIS)	
	ICX (through Signal Integrity IBIS)	
	SpectraQuest (through Signal Integrity IBIS)	
	Mentor Graphics Symbol Generation (Viewdraw)	
形式验证	Cadence Encounter Conformal	
	Synopsys Formality	

表 2. Quartus II 软件支持的 EDA 工具（第 2 部分，共 2 部分）

功能	支持的 EDA 工具	NativeLink 支持
物理综合	Magma Design Automation PALACE	✓
	Synplicity Amplify	

Settings 对话框 (Assignments 菜单) 的 **EDA Tool Settings** 页面允许您指定与 Quartus II 软件一起使用的 EDA 工具。请参见图 8。

图 8. *Settings* 对话框的 *EDA Tool Settings* 页面

EDA Tool Settings 下的几个页面为每种类型的 EDA 工具提供其他选项。

以下步骤说明其它 EDA 工具与 Quartus II 软件配合使用时的基本设计流程。有关所支持的 EDA 工具列表，请参阅 第 12 页上表 2。

1. 创建新工程并指定目标器件或器件系列。
2. 使用标准文本编辑器建立 VHDL 或 Verilog HDL 设计文件。根据需要，例化库功能或使用 MegaWizard Plug-In Manager (Tools 菜单) 建立宏功能模块自定义变量。
3. 使用 Quartus II 支持的 EDA 综合工具之一综合您的设计，并生成 EDIF 网表文件 (.edf) 或 Verilog Quartus Mapping 文件 (.vqm)。
4. (可选) 使用 Quartus II 支持的仿真工具之一对您的设计进行功能仿真。
5. 在 Quartus II Settings 对话框 (Assignments 菜单) 中，指定与 Quartus II 软件一起使用的 EDA 设计输入、综合、仿真、时序分析、板级验证、形式验证和物理综合工具，并为这些工具指定附加选项。
6. 使用 Quartus II 软件编译设计并进行布局布线。可以执行完整编译，或者单独运行 Compiler 模块：
 - a. 运行 Analysis & Synthesis，对设计进行分析，将设计中的功能映射到正确的库模块中。
 - b. 运行 Fitter，对设计进行布局布线。
 - c. 运行 Timing Analyzer，对设计进行时序分析。
 - d. 运行 EDA Netlist Writer，生成其它 EDA 工具使用的输出文件。
 - e. 运行 Assembler，建立设计的编程文件。
7. (可选) 使用 Quartus II 支持的 EDA 时序分析工具之一对设计进行时序分析。
8. (可选) 使用 Quartus II 支持的 EDA 仿真工具之一对设计进行时序仿真。
9. (可选) 使用 Quartus II 支持的 EDA 板级验证工具之一进行板级验证。
10. (可选) 使用 Quartus II 支持的 EDA 形式验证工具之一进行形式验证，确保 Quartus 布线后网表与综合的网表一致。
11. (可选) 使用 Quartus II 支持的 EDA 物理综合工具之一进行物理综合。

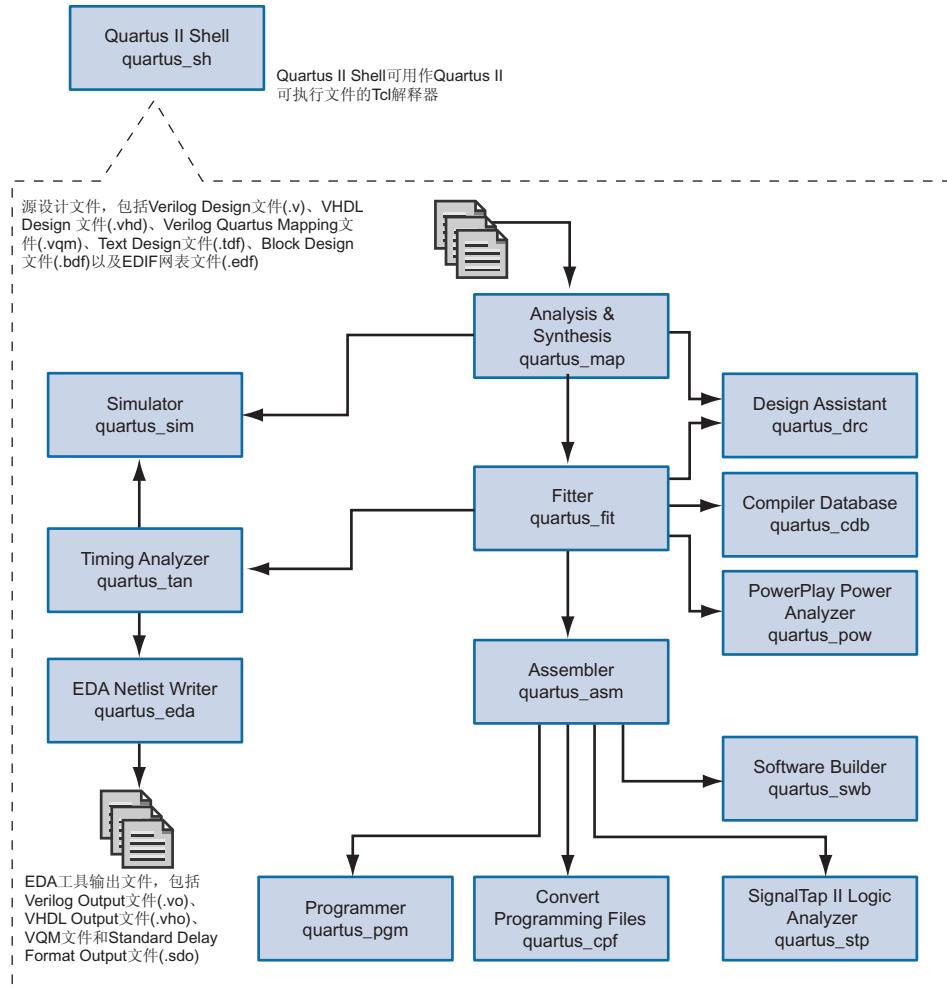
使用编程文件、Programmer 和 Altera 硬件对器件进行编程；或将编程文件转换为其它文件格式以供嵌入式处理器等其它系统使用。

有关信息	请参阅
与 Synplicity Synplify 和 Synplify Pro 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Synplicity Synplify and Synplify Pro Support”
与 Mentor Graphics LeonardoSpectrum 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Mentor Graphics LeonardoSpectrum Support”
与 Mentor Graphics Precision RTL Synthesis 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Mentor Graphics Precision RTL Synthesis Support”
与 Synopsis DC FPGA 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Synopsis Design Compiler FPGA Support”
与 Synplicity Amplify 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 2 卷 “Synplicity Amplify Physical Synthesis Support”
与 Mentor Graphics ModelSim 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷 “Mentor Graphics ModelSim Support”
与 Synopsis VCS 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷 “Synopsis VCS Support”
与 Cadence NC-Sim 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷 “Cadence NC-Sim Support”
与 Synopsis PrimeTime 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷 “Synopsis PrimeTime Support”
与 Cadence Encounter Conformal 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷 “Cadence Encounter Conformal Equivalency Checker Support”
与 Synopsis Formality 软件一起使用 Quartus II 软件	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷 “Synopsis Formality Support”

命令行设计流程

Quartus II 软件提供全面的命令行界面解决方案。它允许您使用命令行可执行文件和选项完成设计流程的每个阶段。使用命令行流程可以降低内存要求，并可使用脚本或标准的命令行选项和命令（包括 Tcl 命令）控制 Quartus II 软件，建立 Makefile。有关命令行设计流程的图示说明，请参见图 9。

图 9. 命令行设计流程



命令行可执行文件

Quartus II 软件包括用于设计流程每个阶段不同的可执行文件。每个可执行文件仅在开始运行时才占用内存。这些可执行文件能够与标准命令行命令和脚本、Tcl 脚本以及 Makefile 脚本一起使用。有关所有命令行可执行文件的列表，请参阅表 3。



独立的图形用户界面可执行文件

Quartus II 软件还提供一些独立的图形用户界面 (GUI) 可执行文件。**qmegawiz** 可执行文件提供独立的 **MegaWizard Plug-In Manager** GUI 版本。**quartus_pgmw** 可执行文件为 Programmer 提供独立的 GUI。**quartus_stpw** 可执行文件为 SignalTap II Logic Analyzer 提供独立的 GUI。

表3. 命令行可执行文件（第1部分，共2部分）

可执行文件	标题	功能
quartus_map	Analysis & Synthesis	建立工程（如果尚未建立），然后建立工程数据库、综合设计并对工程设计文件进行技术映射。
quartus_fit	Fitter	对设计进行布局布线。在运行 Fitter 之前必须成功运行 Analysis & Synthesis。
quartus_drc	Design Assistant	根据一组设计规则检查设计的可靠性。在为 HardCopy 器件转换设计之前，Design Assistant 非常有助于检查可靠性。在运行 Design Assistant 之前必须成功运行 Analysis & Synthesis 或 Fitter。
quartus_tan	Timing Analyzer	分析已实现电路的速度性能。在运行 Timing Analyzer 之前必须成功运行 Fitter。
quartus_asm	Assembler	为目标器件编程或配置建立一个或多个编程文件。在运行 Assembler 之前必须成功运行 Fitter。
quartus_eda	EDA Netlist Writer	生成与其它 EDA 工具配合使用的网表文件和其它输出文件。根据使用的选项，在运行 EDA Netlist Writer 之前，必须成功运行 Analysis & Synthesis、Fitter 或者 Timing Analyzer。
quartus_cdb	Compiler Database Interface (包括 VQM Writer)	生成内部网表文件，包括用于 Quartus II Compiler 数据库的 VQM 文件，使它们可以用于反标和 LogicLock 功能，反标器件和资源分配以保持适配不变，方便今后的编译还可以导入、导出版本兼容的数据库，并合并分区。在运行 Compiler Database Interface 之前必须成功运行 Fitter 或 Analysis & Synthesis。
quartus_sim	Simulator	对设计进行功能或时序仿真。在进行功能仿真之前必须运行 Analysis & Synthesis。在进行时序仿真之前，必须运行 Timing Analyzer。
quartus_pow	Power Analyzer	分析并估算设计的全部动态和静态功耗。计算输出信号的触发速率和静态概率。在运行 PowerPlay Power Analyzer 之前，必须成功运行 Fitter。
quartus_pgm	Programmer	对 Altera 器件编程。

表3. 命令行可执行文件（第2部分，共2部分）

可执行文件	标题	功能
quartus_cpf	Convert Programming Files	将编程文件转换为辅助编程文件格式。
quartus_stp	SignalTap II Logic Analyzer	建立 SignalTap II 文件 (.stp)。在 Assembler 之后运行时，器件全速运行的情况下，SignalTap II Logic Analyzer 捕获内部器件节点信号。
quartus_swb	Software Builder	为 Excalibur 嵌入式处理器进行设计。
quartus_sh	Tcl Shell	为 Quartus II 软件提供 Tcl 脚本 shell。



获取有关 Quartus II 可执行文件的帮助信息

若要获取 Quartus II 每个可执行文件命令行选项的帮助信息，请在命令提示符下键入以下命令之一：

```
<executable name> -h ↵
<executable name> --help ↵
<executable name> --help=<topic or option name> ↵
```

还可以使用 Quartus II Command-Line Executable 和 Tcl API Help Browser 获取有关命令行可执行文件的帮助信息，Help Browser 是基于 Tcl 和 Tk 的 GUI，可以用它浏览命令行和 Tcl API 帮助信息。若要使用此帮助，请在命令提示符下键入以下命令：

```
quartus_sh --qhelp ↵
```

可以单独运行每个可执行文件，也可以使用以下命令一次运行所有 Compiler 可执行文件：

```
quartus_sh --flow compile <project name> [-c <revision name>] ↵
```

此命令将在完整编译过程中运行 quartus_map、quartus_fit、quartus_asm 和 quartus_tan 可执行文件。根据不同的设置，它还可能运行可选的 quartus_drc、quartus_eda 和 quartus_cdb 可执行文件。



quartus_cmd 可执行文件

若要在以前版本的 Quartus II 软件中使用 **quartus_cmd** 可执行文件进行工程编译，由于其后向兼容性，因此仍支持该可执行文件；但是，Altera 建议在所有新设计中，不要使用 **quartus_cmd** 可执行文件，而使用第 17 页上表 3 中列出的可执行文件。如果您习惯使用 **quartus_cmd** 可执行文件来编译设计，您可以通过使用带有以下选项的 **quartus_sh** 可执行文件来实现同样的功能：

```
quartus_sh --flow compile <project name> [-c <Revision Name>] ↵
```

有些可执行文件建立单独的文本型报告文件，您可以使用任何文本编辑器查看这个文件。每个报告文件的名称使用以下格式：

<revision name>.<abbreviated executable name>.rpt

例如，如果要为 **chiptrip** 工程运行 **quartus_map** 可执行文件，您可以在命令提示符下键入以下命令：

```
quartus_map chiptrip ↵
```

quartus_map 可执行文件将进行分析和综合，并生成名为 **chiptrip.map.rpt** 的报告文件。



利用 Quartus II 可执行文件使用 Quartus II 设置文件

您在使用 Quartus II 可执行文件时，Quartus II 软件默认使用和工程名称相同的修订。如果您想要使用与工程名称不同的修订，则可以使用 -c 选项来指定修订的名称，以及与此相关联的 Quartus II Settings File (.qsf)。例如，如果您想要为修订名称为 **speed_ch**、关联文件为 **speed_ch.qsf** 的 **chiptrip** 工程运行 **quartus_map** 可执行文件，可在命令提示符状态下键入如下命令：

```
quartus_map chiptrip -c speed_ch ↵
```

quartus_map 可执行文件使用此修订和设置执行分析与综合，并生成名称为 **speed_ch.map.rpt** 的报告文件。

Quartus II 软件还提供几种使用 Quartus II 可执行文件的预定义编译流程。您可以使用带有 **quartus_sh --flow**，或者 **Tcl execute_flow** 命令的命令。**表 4** 显示了一些最常用的编译流程。

表 4. 命令行编译流程

流程	说明	用于 quartus_sh --flow 或者 execute_flow 的命令行选项
完整编译流程	进行当前设计的完整编译。	compile
编译和仿真流程	如果是时序仿真模式，则执行完整编译，然后仿真当前设计。如果是功能仿真模式，则生成功能仿真网表，然后对当前设计进行仿真。	compile_and_simulate
尝试相似布局流程	对先前编译的设计执行完整编译，在此过程中 Fitter 会比较先前和当前编译中的网表和布局，以便在当前编译中尽可能多地使用先前编译中的节点布局。	attempt_similar_placement
SignalProbe 流程	在不影响设计中现有适配的情况下，将用户指定信号引至输出引脚，以便在完成完整编译之前，进行信号调试。	signalprobe
早期时序估算	进行部分编译，但是在 Fitter 完成之前，停止并生成早期时序估算。	early_timing_estimate
分区合并	在渐进式综合之后合并设计分区以建立单一网表，以便后续编译阶段使用。如果采用渐进式综合，或者在修改设计之后，没有重新完整编译设计，则必须运行此命令。	merge



有关信息

请参阅

使用编译流程

Quartus II Help 中的“Overview: Using Compilation Flows”

使用标准命令行命令和脚本

可以将 Quartus II 可执行文件与任何命令行脚本方法（例如，Perl 脚本、批处理文件和 Tcl 脚本）配合使用。可以设计这些脚本，用以建立新工程或编译现有工程。还可以由命令提示符或控制台运行可执行文件。

图 10 是一个标准命令行脚本的示例。该示例说明了建立工程、进行分析和综合、进行布局布线、时序分析以及为 Quartus II 软件 filtref 教程设计生成编程文件的方法。如果已安装教程设计，它位于 `/altera/qdesigns <version number>/tutorial` 目录中。Altera 建议您建立新目录，并将所有设计文件 (*.v, *.bsf, *.bdf) 从 `/altera/qdesigns <version number>/tutorial` 目录复制到新目录下，以便编译设计流程示例。可以从新工程目录中的命令提示符下运行图 10 中的四个命令，也可以将这些命令存储在批处理文件或 shell 脚本中。这些示例假定在您的 PATH 环境变量中包括 `/<Quartus II system directory>/bin` 目录（或 UNIX 以及 Linux 工作站上的 `/<Quartus II system directory>/<platform>` 目录，其中，`<platform>` 可以是 `solaris`、`linux` 或者 `hp_ll`）。

图 10. 命令行脚本示例

```
quartus_map filtref --family=Stratix ----- 建立针对 Stratix
                                                    器件系列的新
                                                    Quartus II 工程

quartus_fit filtref --part=EP1S10F780C5 --fmax=80MHz --tsu=8ns ----- 为 EP1S10F780C5
                                                    器件进行布局布
                                                    线并指定全局时
                                                    序要求

quartus_tan filtref ----- 进行时序分析

quartus_asm filtref ----- 生成编程文件
```

图 11 是一个用于 UNIX 工作站命令行脚本示例的摘录。此脚本假定当前目录中存在名为 fir_filter 的 Quartus II 教程工程。此脚本分析 fir_filter 工程中的每个设计文件，并报告任何语法错误的文件。

图 11. UNIX 命令行 Shell 脚本示例（第 1 部分，共 2 部分）

```
#!/bin/sh

FILES_WITH_ERRORS=""

for filename in `ls *.bdf *.v`
```

图 11. UNIX 命令行 Shell 脚本示例（第 2 部分，共 2 部分）

```
do
    quartus_map fir_filter --analyze_file=$filename

    if [ $? -ne 0 ]
    then
        FILES_WITH_ERRORS="$FILES_WITH_ERRORS $filename"
    fi
done

if [ -z "$FILES_WITH_ERRORS" ]
then
    echo "All files passed the syntax check"
    exit 0
else
    echo "There were syntax errors in the following file(s)"
    echo $FILES_WITH_ERRORS
    exit 1
fi
```



有关信息

命令行脚本

请参阅

Altera 网站上 *Quartus II Handbook* 第 2 卷
“Command-Line Scripting”

Altera 网站上 *Quartus II Scripting Reference Manual*



使用 Tcl 命令

在 Quartus II 软件中，可以使用 Quartus II 可执行文件运行 Tcl 命令或者建立并运行 Tcl 脚本，执行 Quartus II 工程中的以下任务。Tcl API 功能包括以下类别：

- ◇ 工程与分配功能
- ◇ 器件功能
- ◇ 高级器件功能
- ◇ 流程功能
- ◇ 时序功能
- ◇ 高级时序功能
- ◇ Simulator 功能
- ◇ 报告功能
- ◇ 时序报告功能
- ◇ 反标功能

- ◇ LogicLock 功能
- ◇ Chip Editor 功能
- ◇ 其它功能

可以采用多种方法在 Quartus II 软件中使用 Tcl 脚本。可以使用针对 Tcl 的 Quartus II API 命令建立 Tcl 脚本。应将 Tcl 脚本另存为 Tcl Script File(.tcl)。

可以使用 Quartus II Text Editor 中的 **Templates** 命令 (Edit 菜单) 在文本文件中插入 Tcl 模板和 Quartus II Tcl 模板 (对于 Quartus II 命令)，建立 Tcl 脚本。Quartus II Tcl 模板中使用的命令与 Tcl API 命令语法相同。若要将现有工程引入到另一个工程中，可以使用 **Generate Tcl File for Project** 命令 (Project 菜单) 为该工程生成 Tcl 脚本文件。

可以在 Quartus II Tcl Console 窗口或在 **Tcl Scripts** 对话框 (Tools 菜单) 中使用 **quartus_sh** 可执行文件在命令行模式下运行 Tcl 脚本。



获取有关 Tcl 命令的信息

Quartus II 软件包括 Quartus II 命令行和 Tcl API Help 浏览器，此浏览器是基于 Tcl 和 Tk 的 GUI，用于浏览命令行和 Tcl API 帮助。若要使用此帮助，请在命令提示符下键入以下命令：

```
quartus_sh --qhelp ←
```

也可以在 Quartus II Help 中查看 TCL API Help，该帮助为图形用户界面。更详细的信息，请参阅 Quartus II Help 中的“Overview: Using Tcl Scripting”和“API Functions for Tcl”。

图 12 是一个 Tcl 脚本示例。

图 12. Tcl 脚本示例 (第 1 部分，共 3 部分)

```
# Since ::quartus::report is not pre-loaded
# by quartus_sh, load this package now
# before using the report Tcl API
load_package report

# Since ::quartus::flow is not pre-loaded
# by quartus_sh, load this package now
# before using the flow Tcl API
# Type "help -pkg flow" to view information
# about the package
load_package flow

----- Get Actual Fmax data from the Report File -----
proc get_fmax_from_report {} {
#-----#
```

图 12. Tcl 脚本示例（第 2 部分，共 3 部分）

```
global project_name

# Load the project report database
load_report $project_name

# Get the actual Fmax
set actual_fmax [get_timing_analysis_summary_results -clock_setup
    clock -actual]

# Now unload the project report database
unload_report

return $actual_fmax
}

#----- Set the project name to chiptrip -----#
set project_name chiptrip

#----- Create or open project -----#
if {project_exists $project_name} {

#----- Project already exists -- open project -----#
    project_open $project_name
} else {

#----- Project does not exist -- create new project -----#
    project_new $project_name
}

#----- Fmax requirement: 155.55MHz -----#
set required_fmax 155.55MHz

#----- Make a clock assignment with the Fmax requirement -----#
create_base_clock clock -fmax $required_fmax

#----- Make global assignments -----#
set_global_assignment -name family STRATIX
set_global_assignment -name device EP1S10F484C5
set_global_assignment -name tsu_requirement 7.55ns

#----- Make instance assignments -----#
# The following is the same as doing:
#   "set_instance_assignment -name location -to clock Pin_M20"
set_location_assignment -to clock Pin_M20

#----- Compile using ::quartus::flow -----#
execute_flow -compile
```

图 12. *Tcl 脚本示例（第 3 部分，共 3 部分）*

```
----- Report Fmax from report -----#
set actual_fmax [get_fmax_from_report]
puts ""
puts "-----"
puts "Required Fmax: $required_fmax Actual Fmax: $actual_fmax"
puts "-----"
```

The screenshot shows a search result for 'Tcl 脚本' (Tcl Script). The results are categorized under '有关信息' (Related Information) and '请参阅' (See Also).

- 有关信息** (Related Information):
 - Tcl 脚本**: Description: 'Quartus II 支持使用 Tcl 脚本。通过将您的脚本与各种脚本语言集成在一起，可以提高设计效率。' Reference: 'Altera 网站上 *Quartus II Handbook 第 2 卷 “Tcl Scripting”*'.
- 请参阅** (See Also):
 - Quartus II Help 中的“Overview: Using Tcl Scripting”**: Description: 'Quartus II Help 中的“Overview: Using Tcl Scripting”'.
 - Altera 网站上 *Quartus II Scripting Reference Manual***: Description: 'Altera 网站上 *Quartus II Scripting Reference Manual*'.

建立 Makefile 脚本

Quartus II 软件支持使用 Quartus II 可执行文件的 Makefile 脚本，将您的脚本与各种脚本语言集成在一起。图 13 是标准 Makefile 脚本的摘录。

图 13. *Makefile 脚本摘录（第 1 部分，共 3 部分）*

```
#####
# Project Configuration:
#
# Specify the name of the design (project) and Quartus II Settings
# File (.qsf) and the list of source files used.
#####

PROJECT = chiptrip
SOURCE_FILES = auto_max.v chiptrip.v speed_ch.v tick_cnt.v time_cnt.v
ASSIGNMENT_FILES = chiptrip.qpf chiptrip.qsf

#####
# Main Targets
#
# all: build everything
# clean: remove output files and database
#####
```

图 13. Makefile 纹本摘录（第 2 部分，共 3 部分）

```
all: smart.log $(PROJECT).asm.rpt $(PROJECT).tan.rpt

clean:
    rm -rf *.rpt *.chg smart.log *.htm *.eqn *.pin *.sof *.pof db

map: smart.log $(PROJECT).map.rpt
fit: smart.log $(PROJECT).fit.rpt
asm: smart.log $(PROJECT).asm.rpt
tan: smart.log $(PROJECT).tan.rpt
smart: smart.log

#####
# Executable Configuration
#####

MAP_ARGS = --family=Stratix
FIT_ARGS = --part=EP1S20F484C6
ASM_ARGS =
TAN_ARGS =

#####
# Target implementations
#####

STAMP = echo done >

$(PROJECT).map.rpt: map.chg $(SOURCE_FILES)
    quartus_map $(MAP_ARGS) $(PROJECT)
    $(STAMP) fit.chg

$(PROJECT).fit.rpt: fit.chg $(PROJECT).map.rpt
    quartus_fit $(FIT_ARGS) $(PROJECT)
    $(STAMP) asm.chg
    $(STAMP) tan.chg

$(PROJECT).asm.rpt: asm.chg $(PROJECT).fit.rpt
    quartus_asm $(ASM_ARGS) $(PROJECT)

$(PROJECT).tan.rpt: tan.chg $(PROJECT).fit.rpt
    quartus_tan $(TAN_ARGS) $(PROJECT)

smart.log: $(ASSIGNMENT_FILES)
    quartus_sh --determine_smart_action $(PROJECT) > smart.log

#####
# Project initialization
#####

$(ASSIGNMENT_FILES):
    quartus_sh --prepare $(PROJECT)
```

图 13. Makefile 脚本摘录（第 3 部分，共 3 部分）

```
map.chg:  
    $(STAMP) map.chg  
fit.chg:  
    $(STAMP) fit.chg  
tan.chg:  
    $(STAMP) tan.chg  
asm.chg:  
    $(STAMP) asm.chg
```

有关信息	请参阅
使用命令行可执行文件	Quartus II Help 中的“Overview: Using Command-Line Executables”
Tcl 命令和 Tcl 脚本	Altera 网站上 <i>Quartus II Handbook 第 2 卷 “Command-Line Scripting”</i>
	Quartus II Help 中的“Overview: Using Tcl from the User Interface”、“Overview: Using Tcl Scripting”以及“API Functions for Tcl”
	Altera 网站上 <i>Quartus II Handbook 第 2 卷 “Tcl Scripting”</i>
	Altera 网站上 <i>Quartus II Scripting Reference Manual</i>

设计方法和设计规划

在建立新设计时，应重视考虑 Quartus II 软件提供的设计方法。例如，LogicLock 功能提供自上而下或自下而上的设计方法，以及自上而下渐进式编译设计流程和基于块的设计流程。不管是否使用 EDA 设计输入和综合工具，都可以使用这些设计流程。

自上而下与自下而上的设计方法比较

在自上而下的设计流程中，整个设计只有一个输出网表，您可以对整个设计进行跨设计边界和层次结构的优化处理，而且管理起来也比较容易。

在自下而上的设计方法中，每个设计模块具有单独的网表。此功能允许您单独编译每个模块，并对每个模块应用不同的优化技巧。修改单个模块不会影响其它模块的优化。自下而上的设计还有助于在其它设计中重新使用设计模块。

自上而下渐进式编译设计流程

自上而下渐进式编译设计流程重新使用以前的编译结果，确保只对修改过的设计重新编译，因此能够保持设计性能不变，节省编译时间。处理其他设计分区时，可以只修改设计中关键单元的布局，也可以只对设计的指定部分限定布局，使 Compiler 能够自动优化设计的其余部分，从而提高了设计时序。

在渐进式编译流程中，可以为设计分区分配一个设计输入实例，然后使用 Timing Closure Floorplan 和 LogicLock 功能为分区分配一个器件物理位置，进行完整的设计编译。在编译过程中，Compiler 将综合和适配结果保存在工程数据库中。第一次编译之后，如果对设计做进一步的修改，只有改动过的分区需要重新编译。设计修改完成后，所有分区合并，进行完整的编译。可指定是否只需要进行渐进式综合，以节省编译时间，还是进行完整的渐进式编译，以保持性能不变，并节省大量编译时间。

由于渐进式编译流程能够防止 Compiler 跨分区边界进行优化，因此 Compiler 不会象常规编译那样对面积和时序进行大量优化。为获得最佳的面积和时序结果，Altera 建议记录设计分区的输入和输出，尽量将设计分区数量控制在合理范围内，避免跨分区边界建立过多的关键路径，不要建立太小的分区，如数量少于 1000 的逻辑单元和 Adaptive Logic Modules (ALM) 分区。

关于分区分配和渐进式编译流程其他步骤的详细信息，请参阅以下章节：

- ◇ 第3章，“约束输入”第57页“分配设计分区”。
- ◇ 第4章“综合”第78页“进行渐进式综合”。
- ◇ 第5章“布局布线”第83页“进行完整的渐进式编译”。
- ◇ 第6章“基于模块的设计”第107页“在自上而下渐进式编译流程中使用 LogicLock 区域”。
- ◇ 第9章“时序逼近”第153页“使用渐进式编译达到时序逼近”。
- ◇ 第12章“调试”第181页“渐进式编译使用 SignalTap II Logic Analyzer”。



有关信息

请参阅

使用 Quartus II 演进式编译、演进式综合以及演进式编译流程

Altera 网站上 *Quartus II Handbook 第 1 卷*
“Quartus II Incremental Compilation”

Quartus II Help 中的 “Overview: Using Incremental Compilation” 和 “Overview: Using Incremental Synthesis”

自下而上基于 LogicLock 的设计流程

在基于模块的自下而上 LogicLock 设计流程中，可以独立设计和优化每个模块，在顶层设计中集成所有已优化的模块，然后验证总体设计。每个模块具有单独的网表，在综合和优化之后可以将它们整合在顶层设计中。在顶层设计中，每个模块都不影响其它模块的性能。一般基于模块的设计流程可以在模块化、分层、演进式和团队设计流程中使用。

可以对每个节点和各个层次模块进行 LogicLock 约束，例如，对关键路径进行基于路径的通配符 LogicLock 分配，有利于实现跨越多个设计模块的关键路径。

可以在基于模块的设计流程中使用 EDA 设计输入和综合工具，设计和综合各个模块，然后将各模块整合到 Quartus II 软件的顶层设计中，也可以在 EDA 设计输入和综合工具中完整地进行设计，综合基于模块的设计。关于基于模块设计流程的详细信息，请参考第 101 页，第 6 章：基于模块的设计。



有关信息

请参阅

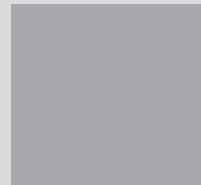
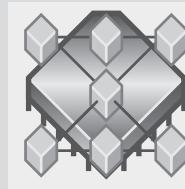
使用 Quartus II 演进式编译和演进式综合

Altera 网站上 *Quartus II Handbook 第 1 卷*
“Quartus II Incremental Compilation”

Quartus II Help 中的 “Overview: Using Incremental Compilation” 和 “Overview: Using Incremental Synthesis”

第二章

设计输入



第2章 内容：	
简介	32
建立工程	33
建立设计	39
使用 Altera 宏功能模块	43

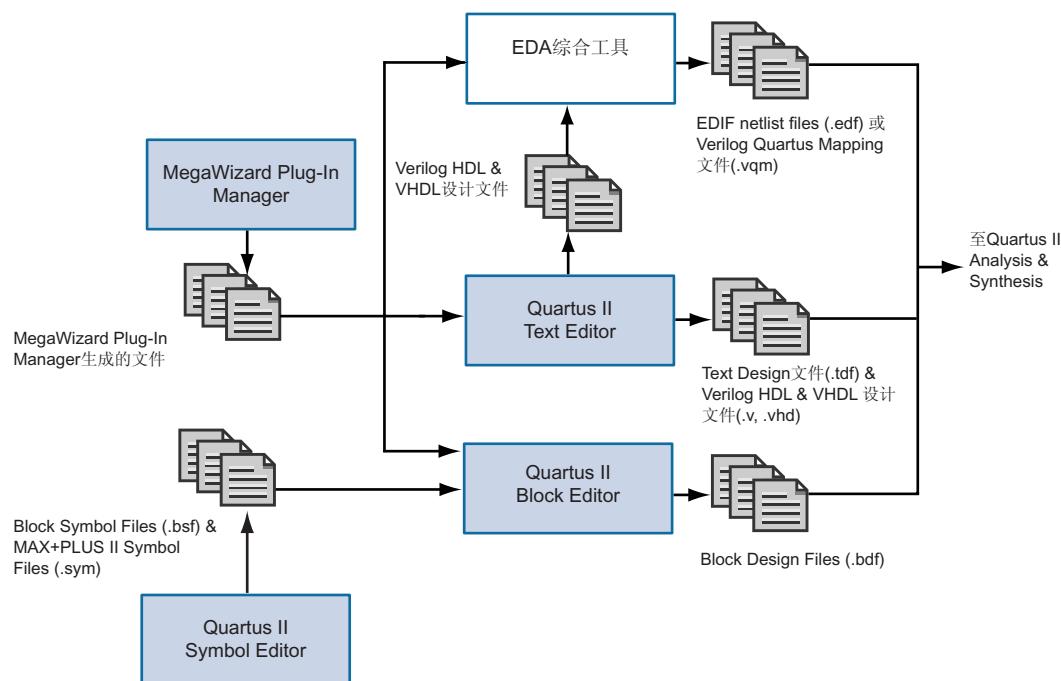
2

简介



Quartus® II 工程包括实现成功设计所需的所有设计文件，软件源文件和其他相关文件。使用修订，可以比较工程多个版本的设置和分配，更快、更有效地满足设计要求。使用 Quartus II Block Editor, Text Editor, MegaWizard® Plug-In Manager (Tools 菜单) 和 EDA 设计输入工具可以建立包括 Altera® 宏功能模块、参数化模块库 (LPM) 功能和知识产权 (IP) 功能在内的设计。图 1 所示为设计输入流程。

图 1. 设计输入流程



Quartus II 的 Altera SOPC Builder 和 DSP Builder 软件支持系统级设计输入流程。关于这些系统级设计输入流程的详细信息，请参阅第 205 页，第 15 章“系统级设计”。

建立工程

使用 **New Project Wizard** (**File** 菜单) 或 **quartus_map** 可执行文件建立新工程。建立新工程时，指定工程工作目录，分配工程名称，指定顶层设计实体的名称。还可以指定在工程中使用的设计文件、其它源文件、用户库和 EDA 工具，以及目标器件（或者让 Quartus II 软件自动选择器件）。表 1 列出了一個 Quartus II 工程的工程和设置文件。

表 1. Quartus II 工程文件

文件类型	说明
Quartus II Project File (.qpf)	指定用来建立工程和与工程相关修订的 Quartus II 软件版本。
Quartus II Settings File (.qsf)	包括 Assignment Editor、Floorplan Editor、 Settings 对话框 (Assignments 菜单)、Tcl 脚本或者 Quartus II 可执行文件产生的所有修订范围内或者独立的分配。工程中每个修订有一个 QSF。
Quartus II Workspace File (.qws)	包含用户偏好和其他信息，例如窗口位置，窗口中打开文件及其位置。
Quartus II Default Settings File (.qdf)	位于 \<Quartus II system directory>\bin 目录下，包括所有全局默认工程设置。QSF 中的设置将替代这些设置。

一旦建立了工程，可以使用 **Settings** 对话框 (**Assignments** 菜单) 从工程中添加和删除设计文件以及其他文件。在执行 Quartus II Analysis & Synthesis 期间，Quartus II 软件将按文件在 **Files** 页面中显示的顺序来处理文件。

可以使用 **Copy Project** 命令 (**Project** 菜单)，将整个工程复制到新的目录下，包括工程设计数据库文件、设计文件、设置文件和报告文件，然后在新目录下，打开该工程。如果不存在新目录，Quartus II 将建立该目录。

Project Navigator 显示与当前修订相关的信息并且以图形表示工程层次、文件和设计单元，以及各种菜单命令的快捷键。您也可以利用 **Customize Columns** 命令 (右键弹出菜单) 自定义 Project Navigator 所显示的信息。

图 2. Project Navigator 窗口

Entity	Logic Cells	LC Registers	Memory Bits	Pins	Virtual Pins	LUT-Only LCs	Register-0
Cyclone: EP1C6F256C6							
filter	102 (9)	58	0	22	0	44 (0)	25 (9)
taps:inst	32 (32)	32	0	0	0	0 (0)	16 (16)
state_minst1	5 (5)	5	0	0	0	0 (0)	0 (0)
hvalues:inst2	0	0	0	0	0	0	0
acc:inst3	23 (23)	12	0	0	0	11 (11)	0 (0)
accum:inst_1	0	0	0	0	0	0	0
lpm_add_sub:lpm_add_sub_component	0	0	0	0	0	0	0
mult:inst6	33 (0)	0	0	0	0	33 (0)	0 (0)
lpm_mult:lpm_mult_component	33 (0)	0	0	0	0	33 (0)	0 (0)
altshift:external_latency_ifs	0	0	0	0	0	0	0
multcore:mult_core	33 (15)	0	0	0	0	33 (15)	0 (0)
mpar_add:padder	18 (0)	0	0	0	0	18 (0)	0 (0)
lpm_add_sub:adder	0	0	0	0	0	0	0

Project Navigator Window 也可以分配设计分区。更详细的信息，请参考第 57 页“分配设计分区”。

有关信息

建立并操作 Quartus II 工程

请参阅

Quartus II Help 中的“Overview: Using the New Project Wizard”、“Overview: Using the Project Navigator”以及“Overview: Working with Quartus II Projects”

管理 Quartus II 工程

Quartus II 教程中的 Design Entry 部分

Altera 网站上 *Quartus II Handbook* 第 2 卷
“Quartus II Project Management”

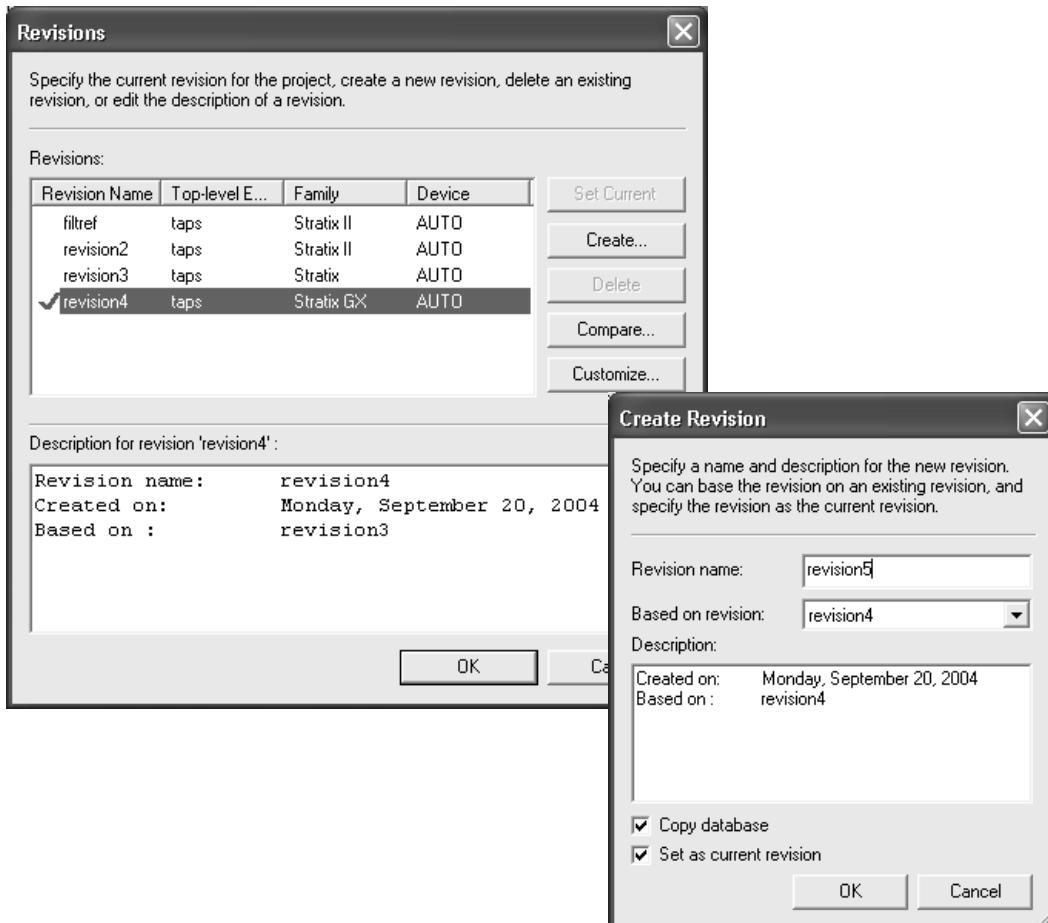
使用修订

您可以使用修订为设计中的设计文件指定、保存、使用不同的设置和分配组。对于设计中的相同设计文件，修订可以采用不同的设置和分配来比较结果。

建立修订时，Quartus II 软件建立一个单独的 QSF，包括所有与修订相关的设置和分配，并将其放置在设计的顶层目录下。可对设计的任何实体建立修订。在 **Revisions** 对话框中可以查看所有修订的顶层实体，或者在 **Settings** 对话框 (Assignments 菜单)General 页面中，查看当前顶层设计实体。

Rewvisions 对话框 (Project 菜单) 让您查看当前工程的所有修订、为特定设计实体建立一个修订、删除修订，或者设置一个特殊修订作为当前修订，以便编译、仿真和进行时序分析。Revision 对话框中的信息显示了特殊修订的顶层设计实体以及修订的器件系列。选中标记图标显示当前修订。使用 **Create Revision** 对话框，可建立新的修订（基于当前的修订）、输入修订的说明，复制用于建立修订的数据库，并设置某个修订为当前修订。还可以选定哪个栏目出现在 Revision 对话框中。请参阅图 3。

图 3. *Rewvisions* 对话框



建立修订并不影响工程的源设计文件。可以建立修订，将其设置为设计的当前修订，然后为实体建立分配和设置。此功能可以让您为同一个设计实体建立不同的设置和分配，并将这些设置保存为不同的修订，以便比较。每一个

修订都有相应的报告文件，可以打开它，查看并且对照其他修订，比较设置效果和分配更改的结果。还可以使用 **Revisions** 对话框下的 **Compare Revisions** 对话框来比较不同修订的编译结果。**Compare Revisions** 对话框具有 **Results** 标签和 **Assignments** 标签选项。默认情况下，将显示所有修订的比较，也可以选择希望显示的修订并调整显示顺序来自定义比较。还可以从比较中输出一个 Comma-Separated Values 文件 (.csv)。图 4 显示了 **Compare Revisions** 对话框的 **Results** 标签选项，以比较每个修订结果。

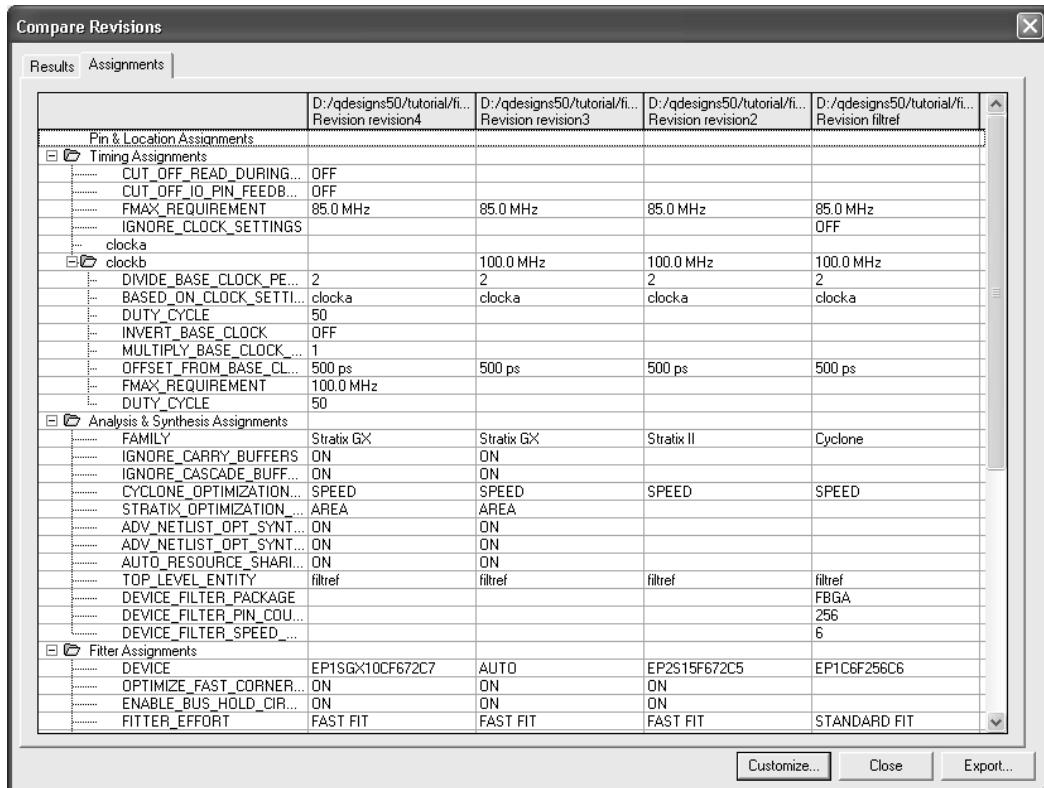
图 4. *Compare Revisions* 对话框的 *Results* 标签选项

The screenshot shows the 'Compare Revisions' dialog box with the 'Results' tab selected. The dialog has two tabs at the top: 'Results' (selected) and 'Assignments'. Below the tabs is a table comparing four revisions: revision4, revision3, revision2, and filtre. The table is divided into sections: Analysis & Synthesis and Fitter. The Analysis & Synthesis section includes fields like Flow Status, Quartus II Version, Revision Name, Top-level Entity Name, Family, Met timing requirements, Total logic elements, Total pins, Total virtual pins, Total memory bits, DSP block 9-bit elements, Total PLLs, Total DLLs, Device, and Timing Models. The Fitter section includes fields like Flow Status, Quartus II Version, Revision Name, Top-level Entity Name, Family, Met timing requirements, Total logic elements, Total pins, Total virtual pins, Total memory bits, DSP block 9-bit elements, Total PLLs, Total DLLs, Device, Timing Models, Total ALUTs, and Total registers. The table uses a light gray background with alternating row colors (white and light gray). The right side of the dialog features scroll bars and buttons for 'Customize...', 'Close', and 'Export...'.

	D:/qdesigns50/tutorial/lu... Revision revision4	D:/qdesigns50/tutorial/lu... Revision revision3	D:/qdesigns50/tutorial/lu... Revision revision2	D:/qdesigns50/tutorial/lu... Revision filtre
Analysis & Synthesis				
Flow Status	Successful - Wed Apr 2...	Successful - Wed Apr 2...	Successful - Wed Apr 2...	Successful - Wed Apr 2...
Quartus II Version	5.0	5.0	5.0	5.0
Revision Name	revision4	revision3	revision2	filref
Top-level Entity Name	filref	filref	filref	filref
Family	Stratix GX	Stratix GX	Stratix II	Cyclone
Met timing requirements	N/A	N/A	N/A	N/A
Total logic elements	129	121	121	121
Total pins	22	22	22	22
Total virtual pins	0	0	0	0
Total memory bits	0	0	0	0
DSP block 9-bit elements	0	0	0	0
Total PLLs	0	0	0	0
Total DLLs	0	0	0	0
Device				EP1C6F256C6
Timing Models				Final
Fitter				
Flow Status	Successful - Wed Apr 2...	Successful - Wed Apr 2...	Successful - Wed Apr 2...	Successful - Wed Apr 2...
Quartus II Version	5.0	5.0	5.0	5.0
Revision Name	revision4	revision3	revision2	filref
Top-level Entity Name	filref	filref	filref	filref
Family	Stratix GX	Stratix GX	Stratix II	Cyclone
Met timing requirements	N/A	N/A	N/A	N/A
Total logic elements	111 / 10,570 (1 %)	111 / 10,570 (1 %)	102 / 5,980 (1 %)	102 / 5,980 (1 %)
Total pins	22 / 349 (6 %)	22 / 349 (6 %)	22 / 367 (5 %)	22 / 185 (11 %)
Total virtual pins	0	0	0	0
Total memory bits	0 / 920,448 (0 %)	0 / 920,448 (0 %)	0 / 419,328 (0 %)	0 / 92,160 (0 %)
DSP block 9-bit elements	0 / 48 (0 %)	0 / 48 (0 %)	0 / 96 (0 %)	0 / 96 (0 %)
Total PLLs	0 / 5 (0 %)	0 / 5 (0 %)	0 / 6 (0 %)	0 / 2 (0 %)
Total DLLs	0 / 2 (0 %)	0 / 2 (0 %)	0 / 2 (0 %)	0 / 2 (0 %)
Device	EP1SGX10CF672C7	EP1SGX10CF672C5	EP2S15F672C5	EP1C6F256C6
Timing Models	Final	Final	Preliminary	Final
Total ALUTs			92 / 12,480 (< 1 %)	
Total registers			58	

图 5 所示为 **Compare Revisions** 对话框的 **Assignments** 标签选项，对每个修订的分配设置进行比较。

图 5. Compare Revisions 对话框的 Assignments 标签选项



有关信息	请参阅
使用修订	Altera 网站上 <i>Quartus II Handbook</i> 第 2 卷 “Quartus II Project Management”
建立和使用修订	Quartus II Help 中的 “Overview: Using Revisions”

使用版本兼容的数据库

Quartus II 软件能够导出版本兼容的数据库文件，不必再对设计进行完整编译，便可在 Quartus II 软件后续版本中使用。在运行 Analysis & Synthesis 或者 quartus_map 可执行文件之后，可以在设计流程的任一阶段导出数据库。

该功能可以建立和优化设计，为 Quartus II 软件后续版本保留 Timing Analysis 数据库，确保时序模型更新后，设计仍能符合时序要求。

为导出数据库，以便在 Quartus II 软件后续版本中使用，由 **Export Database** 命令 (Project 菜单) 选择目录，导出数据库。Quartus II 软件导出设计数据库。然后，在 Quartus II 软件后续版本中使用 **Import Database** 命令 (Project 菜单) 选择工程文件夹，导入设计数据库，分析时序，验证设计的时序要求。

也可以使用 **quartus_cdb** 命令行可执行文件来导出或导入设计数据库。版本兼容数据库由 Quartus II 软件 4.1 及其后续版本提供支持。



使用 quartus_cdb 可执行文件

您可以使用 **quartus_cdb** 可执行文件来导出或导入版本兼容的数据库。

使用 **quartus_cdb** 可执行文件导入或导出数据库，请在命令提示符下，键入以下命令之一：

```
quartus_cdb <project> -c <revision> --import_database=<project directory> ↵
quartus_cdb <project> -c <revision> --export_database=<project directory> ↵
```

如果您希望得到 **quartus_cdb** 可执行文件的帮助，请在命令提示符下，键入以下命令之一：

```
quartus_cdb -h ↵
quartus_cdb --help ↵
quartus_cdb --help=<topic name> ↵
```

有关信息

请参阅

使用版本兼容的数据库

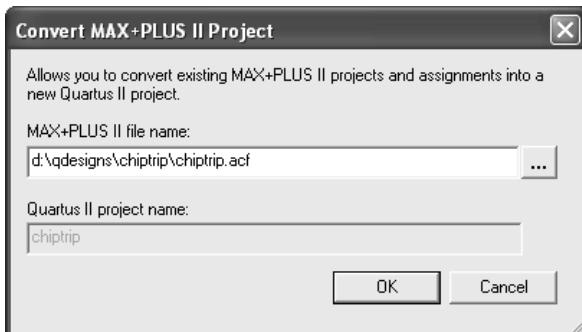
Altera 网站上 *Quartus II Handbook 第 2 卷*
“*Quartus II Project Management*”

转换 MAX+PLUS II 工程



Convert MAX+PLUS II Project 命令 (File 菜单) 使您能够从原有 MAX+PLUS II 工程中选定一个现有 MAX+PLUS II 工程的 Assignment & Configuration File(.acf)，或者设计文件，将其转换为一个新的 Quartus II 工程，包含所有支持的分配和约束条件。**Convert MAX+PLUS II Project** 命令会自动导入 MAX+PLUS II 分配和约束条件、建立新的工程文件，并打开新的 Quartus II 工程。图 6 所示为 **Convert MAX+PLUS II Project** 对话框。

图 6. Convert MAX+PLUS II Project 对话框



有关信息

转换 MAX+PLUS II 工程

请参阅

Quartus II Tutorial 中的 MAX+PLUS II Conversion 部分

Altera 网站上 *Quartus II Handbook 第 1 卷 “Quartus II Design Flow for MAX+PLUS II Users”*

建立设计

可以使用 Quartus II 软件在 Quartus II Block Editor 中建立设计，或使用 Quartus II Text Editor 通过 AHDL、Verilog HDL 或 VHDL 设计语言来建立 HDL 设计。

Quartus II 软件还支持采用 EDA 设计输入和综合工具生成的 EDIF Input Files (.edf) 或 Verilog Quartus Mapping Files (.vqm) 建立的设计。还可以在 EDA 设计输入工具中建立 Verilog HDL 或 VHDL 设计，生成 EDIF 输入文件和 VQM 文件，或在 Quartus II 工程中直接使用 Verilog HDL 或 VHDL 设计文件。有关使用 EDA 综合工具生成 EDIF 输入文件或 VQM 文件的详细信息，请参阅第 4 章“综合”，第 66 页“使用其它 EDA 综合工具”。

可以使用表 2 列出的设计文件类型在 Quartus II 软件或 EDA 设计输入工具中建立设计。

表2. 支持的设计文件类型

类型	说明	扩展名
Block Design File	使用 Quartus II Block Editor 建立的原理图设计文件。	.bdf
EDIF Input File	使用任何标准 EDIF 网表编写程序生成的 EDIF 200 版网表文件。	.edf .edif
Graphic Design File	使用 MAX+PLUS II Graphic Editor 建立的原理图设计文件。	.gdf
Text Design File	以 Altera 硬件描述语言 (AHDL) 编写的设计文件。	.tdf
Verilog Design File	包含使用 Verilog HDL 定义设计逻辑的设计文件。	.v .vlg .verilog
VHDL Design File	包含使用 VHDL 定义设计逻辑的设计文件。	.vh .vhd .vhdl
Verilog Quartus Mapping File	Synplicity Synplify 软件或 Quartus II 软件生成的 Verilog HDL 格式网表文件。	.vqm

使用 Quartus II Block Editor



Block Editor 用于以原理图和框图的形式输入和编辑图形设计信息。Quartus II Block Editor 读取并编辑 Block Design Files(框图设计文件) 和 MAX+PLUS II Graphic Design Files(图形设计文件)。可以在 Quartus II 软件中打开 Graphic Design Files 并将其另存为 Block Design Files。Block Editor 替换来自 MAX+PLUS II 软件的 Graphic Editor。

每一个 Block Design Files 包含设计中代表逻辑的框图和符号。Block Editor 将每一个框图、原理图或者符号代表的设计逻辑合并到工程中。

可以用 Block Design Files 中的框图建立新设计文件，在修改框图和符号时更新设计文件，也可以在 Block Design Files 的基础上生成 Block Symbol Files(.bsf)、AHDL Include 文件 (.inc) 和 HDL 文件。还可以在编译之前分析 Block Design Files 是否出错。Block Editor 提供有助于您在框图设计文件中连接框图和基本单元（包括总线和节点连接以及信号名称映射）的一组工具。

可以更改 Block Editor 的显示选项，例如根据您的偏好更改导向线和网格间距、橡皮带式生成线、颜色和像素、缩放以及不同的框图和基本单元属性。

Block Editor 的以下功能可以帮助您在 Quartus II 软件中建立 Block Design Files:

- ◇ 对 Altera 提供的宏功能模块进行例化：**MegaWizard Plug-In Manager**(Tools 菜单) 用于建立或修改包含宏功能模块自定义变量的设计文件。这些自定义宏功能模块变量是基于 Altera 提供的包括 LPM 功能在内的宏功能模块。宏功能模块以 Block Design Files 中的框图表示。请参阅 第 45 页 “使用 MegaWizard Plug-In Manager”。
 - ◇ 插入框图和基本单元符号：结构图使用称为框图的矩形符号代表设计实体以及相应的分配信号，在自上而下的设计中很有用。框图由代表相应信号流程的管道连接起来。可以将结构图专用于代表您的设计，也可以将其与原理单元结合使用。
- Quartus II 软件提供可在 Block Editor 中使用的各种逻辑功能符号，包括基本单元、参数化模块库 (LPM) 功能和其它宏功能模块。
- ◇ 从框图或 Block Design Files 中建立文件：为方便设计层次化工程，可以在 Block Editor 中使用 **Create/Update** 命令 (File 菜单) 从 Block Design Files 中的框图开始建立其它 Block Design Files、AHDL Include 文件、Verilog HDL 和 VHDL 设计文件以及 Quartus II Block Symbol Files。还可以从 Block Design File 本身建立 Verilog 设计文件、VHDL 设计文件和 Block Symbol Files。

使用 Quartus II Text Editor



Quartus II Text Editor 是一个灵活的工具，用于以 AHDL、VHDL 和 Verilog HDL 语言以及 Tcl 脚本语言输入文本型设计。还可以使用 Text Editor 输入、编辑和查看其它 ASCII 文本文件，包括为 Quartus II 软件或由 Quartus II 软件建立的文本文件。

还可以用 Text Editor 将任何 AHDL 声明或节段模板、Tcl 命令或所支持的 VHDL 以及 Verilog HDL 构造模板插入到当前文件中。AHDL、VHDL 和 Verilog HDL 模板为您输入 HDL 语法提供了简便方法，提高了设计输入的速度和准确度。还可获取有关所有 AHDL 单元、关键字和声明以及宏功能模块和基本单元的上下文相关帮助。

使用 Quartus II Symbol Editor



Symbol Editor 用于查看和编辑代表宏功能、宏功能模块、基本单元或设计文件的预定义符号。每个 Symbol Editor 文件代表一个符号。对于每个符号文件，均可以从包含 Altera 宏功能模块和 LPM 功能的库中选择。可以自定义这些 Block Symbol Files，然后将这些符号添加到使用 Block Editor 建立的原理图中。Symbol Editor 读取并编辑 Block Symbol Files 和 MAX+PLUS II Symbol Files(.sym)，并将这两种类型的文件存储为 Block Symbol Files。

使用 Verilog HDL、VHDL 与 AHDL

可以使用 Quartus II Text Editor 或其它文本编辑器建立 Text Design Files, Verilog 和 VHDL 设计文件，并在层次化设计中将这些文件与其它类型设计文件相组合。



Verilog 和 VHDL 设计文件可以包含 Quartus II 所支持构造的任意组合。还可以包含 Altera 提供的逻辑功能，如基本单元和宏功能模块以及用户自定义的逻辑功能。



在 Text Editor 中，使用 **Create/Update** 命令 (File 菜单) 从当前的 Verilog HDL 或 VHDL 设计文件中建立 Block Symbol File，然后将其合并到 Block Design File 中。同样，可以建立代表 Verilog HDL 或 VHDL 设计文件的 AHDL Include 文件，并将其合并到 Text Design File 中或另一个 Verilog HDL 或 VHDL 设计文件中。

对于 VHDL 设计，可以在 **Settings** 对话框 (Assignments 菜单) **Files** 页面中，或者 Project Navigator 的 **Files** 标签选项中指定 VHDL 库的名称。

有关在 Quartus II 软件中使用 Verilog HDL 和 VHDL 语言的详细信息，请参阅第 4 章 “综合”，第 63 页 “使用 Quartus II Verilog HDL & VHDL Integrated Synthesis”。



AHDL 是一种完全集成到 Quartus II 系统中的高级模块化语言。AHDL 支持布尔等式、状态机、条件逻辑和解码逻辑。AHDL 还可用于建立和使用参数化功能，并全面支持 LPM 功能。AHDL 特别适合设计复杂的组合逻辑、批处理、状态机、真值表和参数化逻辑。



有关信息

请参阅

使用 Quartus II Block Editor 和 Symbol Editor

Quartus II Help 中的“Block Editor & Symbol Editor Introduction”

使用 Quartus II Text Editor

Quartus II Help 中的“Text Editor Introduction”

在 Quartus II 软件中建立设计

Quartus II Tutorial 中的 Design Entry 模块

使用 Altera 宏功能模块



Altera 宏功能模块是复杂的高级构建模块，可以在 Quartus II 设计文件中与逻辑门和触发器基本单元一起使用。Altera 提供的参数化宏功能模块和 LPM 功能均为 Altera 器件结构做了优化。必须使用宏功能模块才可以使用一些 Altera 专用器件的功能，例如，存储器、DSP 块、LVDS 驱动器、PLL 以及 SERDES 和 DDIO 电路。

可以使用 **MegaWizard Plug-In Manager** (Tools 菜单) 建立 Altera 宏功能模块、LPM 功能和 IP 功能，用于 Quartus II 软件和 EDA 设计输入与综合工具中的设计。表 3 列出了能够由 **MegaWizard Plug-In Manager** 建立的 Altera 提供的宏功能模块和 LPM 功能类型。

表 3. Altera 提供的宏功能模块与 LPM 功能

类型	说明
算术组件	包括累加器、加法器、乘法器和 LPM 算术功能。
逻辑门	包括多路复用器和 LPM 门功能。
I/O 组件	包括时钟数据恢复 (CDR)、锁相环 (PLL)、双数据速率 (DDR)、千兆收发器块 (GXB)、LVDS 接收器和发送器、PLL 重新配置和远程更新宏功能模块。
存储器编译器	包括 FIFO Partitioner、RAM 和 ROM 宏功能模块。
存储组件	存储器、移位寄存器宏功能模块和 LPM 存储器功能。

为节省宝贵的设计时间，Altera 建议使用宏功能模块，而不是对您自己的逻辑进行源代码编写。此外，这些功能可以提供更有效的逻辑综合和器件实现。只需通过设置参数便可方便地将宏功能模块定制为不同的大小。Altera 还为宏功能模块和 LPM 功能提供 AHDIL Include 文件和 VHDL Component Declarations。

有关信息	请参阅
使用 MegaWizard Plug-In Manager	Quartus II Help 中的“Overview: Using the MegaWizard Plug-In Manager”
	Quartus II Tutorial 中的 Design Entry 部分

使用知识产权 (IP) 宏功能模块

Altera 提供多种方法来获取 Altera Megafunction Partners Program(AMPP™) 和 MegaCore® 宏功能模块，这些功能经严格的测试和优化，可以在 Altera 专用器件结构中发挥出最佳性能。使用这些知识产权参数化模块能够减少设计和测试时间。MegaCore 和 AMPP 宏功能模块包括应用于嵌入式处理器、接口和外设、数字信号处理 (DSP) 以及通信应用的宏功能模块。

Altera 提供以下程序、特性和功能，协助您在 Quartus II 软件和 EDA 设计输入工具中使用 IP 功能。

- ◇ **AMPP 程序：**AMPP 程序支持第三方供应商，以便建立和发布与 Quartus II 软件一起使用的宏功能模块。AMPP 合作伙伴提供一系列为 Altera 器件优化过的现成宏功能模块。

AMPP 功能的评估期由各供应商决定。可以从 Altera 网站 www.altera.com/ipmegastore 上的 IP MegaStore™ 下载和评估 AMPP 功能。

- ◇ **MegaCore 功能：**功能是用于复杂系统级功能的预验证 HDL 设计文件，它经过了优化，并由 MegaWizard Plug-In Manager 和 IP Toolbench 进行了全面参数化。工具条 IP Toolbench 可用于快速方便的查看文档、指定参数、建立其他 EDA 工具，并生成将参数化 MegaCore 功能集成到设计中所需的所有文件。

可以在安装 Quartus II 软件过程中，或者安装完毕后，由 MegaCore IP Library

CD-ROM 来安装 MegaCore 功能。也可以从 Altera 网站，通过 IP

MegaStore 下载单独的 IP MegaCore 功能，并分别进行安装。或通过将 MegaWizard Portal Extension 用于 **MegaWizard Plug-In Manager** 来使用 MegaCore 功能。

- ◇ **OpenCore 评估功能：**OpenCore® 评估功能使您能够在采购之前评估 AMPP 功能。可以使用 OpenCore 功能编译、仿真并验证设计的性能，但不支持编程文件的生成。
- ◇ **OpenCore Plus 硬件评估功能：**免费的 OpenCore Plus 硬件评估功能使您能够对系统中的 MegaCore 功能进行行为仿真、验证设计的功能，迅速方便的评估系统大小和速度。此外，Quartus II 软件为含有 MegaCore 功能的设计生成时限编程文件，使您能够在购买 IP 宏功能许可之前，对器件进行编程、在硬件中验证设计。

Settings 对话框 (Assignments 菜单) **Compilation Process** 页面下的 OpenCore Plus 硬件特性打开后，Quartus II 软件插入一小部分控制逻辑，这些逻辑会对适配有不利的影响（特别是对小型器件）。您可以关掉 OpenCore Plus 硬件评估特性，使 Quartus II 软件忽略这些附加逻辑。

使用 **MegaWizard Plug-In Manager**



MegaWizard Plug-In Manager 可以帮助您建立或修改含有自定义宏功能模块变量的设计文件，然后可以在设计文件中对其进行例化。这些自定义宏功能模块变量基于 Altera 提供的宏功能模块，包括 LPM、MegaCore 和 AMPP 功能。**MegaWizard Plug-In Manager** 运行一个向导，帮助您轻松地为自定义宏功能模块变量设定选项。该向导用于设置参数和部分端口数值。可以从 Tools 菜单或从 Block Design File 中打开 **MegaWizard Plug-In Manager**，也可以将它作为独立的应用程序来运行。表 4 列出了 **MegaWizard Plug-In Manager** 为用户生成的每个自定义宏功能模块变量而生成的文件。

表 4. *MegaWizard Plug-In Manager* 生成的文件 (第 1 部分, 共 2 部分)

文件名称	说明
<output file>.bsf	Block Editor 中使用的宏功能模块符号。
<output file>.cmp	Component Declaration File(组件声明文件)。
<output file>.inc	宏功能模块封装文件中模块的 AHDL Include 文件。
<output file>.tdf	在 AHDL 设计中例化的宏功能模块封装文件。
<output file>.vhd	在 VHDL 设计中例化的宏功能模块封装文件。

表4. *MegaWizard Plug-In Manager*生成的文件（第2部分，共2部分）

文件名称	说明
<code><output file>.v</code>	在Verilog HDL设计中例化的宏功能模块封装文件。
<code><output file>_bb.v</code>	Verilog HDL设计所用宏功能模块封装文件中模块的空体或black-box声明，用于在使用EDA综合工具时指定端口方向。
<code><output file>_inst.tdf</code>	宏功能模块封装文件中子设计的AHDL例化示例。
<code><output file>_inst.vhd</code>	宏功能模块封装文件中实体的VHDL例化示例。
<code><output file>_inst.v</code>	宏功能模块封装文件中模块的Verilog HDL例化示例。



使用独立的 *MegaWizard Plug-In Manager*

可以在命令提示符下键入以下命令，实现在Quartus II软件之外使用*MegaWizard Plug-In Manager*：

```
qmegawiz ↵
```

在Quartus II软件中例化宏功能模块

可以在Block Editor中直接例化、在HDL代码中例化（通过端口和参数定义例化或使用*MegaWizard Plug-In Manager*参数化宏功能模块，并建立封装文件），也可以通过界面，在Quartus II软件中对Altera宏功能模块和LPM功能进行例化。

Altera建议您使用*MegaWizard Plug-In Manager*例化宏功能模块，建立自定义宏功能模块变量。向导将提供一个供自定义和参数化宏功能模块使用的图形界面，并确保您正确设置所有宏功能模块的参数。

在Verilog HDL和VHDL中例化

可以使用*MegaWizard Plug-In Manager*建立宏功能模块或自定义宏功能模块变量。然后通过*MegaWizard Plug-In Manager*建立包含宏功能模块实例的Verilog HDL或VHDL封装文件，您可以在设计中使用此文件。对于VHDL宏功能模块，*MegaWizard Plug-In Manager*还建立Component Declaration File。

使用端口和参数定义

可以采用与调用任何其它模块或组件相类似的调用功能，直接在 Verilog HDL 或 VHDL 设计中对宏功能模块进行例化。在 VHDL 中，还需要使用 Component Declaration。

推断宏功能模块

Quartus II Analysis & Synthesis 可以自动识别某些类型的 HDL 代码，推断相应的宏功能模块。由于 Altera 宏功能模块已对 Altera 器件进行了优化，并且性能要好于标准的 HDL 代码，因此 Quartus II 软件可以使用推断。对于一些专用体系结构功能，例如 RAM 和 DSP 块，必须使用 Altera 宏功能模块。

Quartus II 软件在综合过程中将以下逻辑映射到宏功能模块：

- ◇ 计数器
- ◇ 加法 / 减法器
- ◇ 乘法器
- ◇ 乘 - 累加器和乘 - 加法器
- ◇ RAM
- ◇ 移位寄存器

在 EDA 工具中例化宏功能模块



可以在 EDA 设计输入和综合工具中使用 Altera 提供的宏功能模块、LPM 功能和 IP 功能。可以通过为功能建立 black-box、通过推断或通过使用 clear-box 方法在 EDA 工具中例化宏功能模块。

使用 Black-Box 方法



可以使用 **MegaWizard Plug-In Manager** 为宏功能模块生成 Verilog HDL 或 VHDL 封装文件。对于 Verilog HDL 设计，**MegaWizard Plug-In Manager** 还生成包含模块空体声明的 Verilog 设计文件，用于指定端口方向。

Verilog HDL 或 VHDL 封装文件包含宏功能模块的端口和参数，可以将其用在顶层设计文件中，例化宏功能模块，并指示 EDA 工具在综合期间将宏功能模块作为 black-box 处理。

以下步骤描述使用 **MegaWizard Plug-In Manager**，在 EDA 设计输入和综合工具中为 Altera 宏功能模块或 LPM 功能建立 black-box 的基本流程。

1. 使用 **MegaWizard Plug-In Manager** 建立和参数化宏功能模块或 LPM 功能。
2. 使用 **MegaWizard Plug-In Manager** 生成的 black-box 文件或者组件声明（与实例化文件一起）在 EDA 综合工具中例化。
3. 在 EDA 综合工具中进行设计综合和优化。EDA 综合工具在综合期间将宏功能模块作为 black-box 处理。

按推断进行例化

EDA 综合工具可自动识别某些类型的 HDL 代码，推断相应的宏功能模块。可以在 Verilog HDL 或 VHDL 代码中直接对存储器块 (RAM 和 ROM)、DSP 块、移位寄存器和一些算术组件进行例化。然后，EDA 工具在综合期间将逻辑映射到相应的 Altera 宏功能模块。

使用 Clear-Box 方法

在 black-box 流程中，EDA 综合工具将 Altera 宏功能模块和 LPM 功能作为 black-box 处理。因此，EDA 综合工具不能使用 Altera 宏功能模块实现完全综合和设计优化，因为此工具没有该功能的完整模型或时序信息。使用 clear-box 流程，用户可以由 **MegaWizard Plug-In Manager** 建立 EDA 综合工具能够使用的完全可综合 Altera 宏功能模块或 LPM 功能。

以下步骤描述在 EDA 综合工具中使用 clear-box 宏功能模块的基本流程：

1. 使用 **MegaWizard Plug-In Manager** 建立和参数化宏功能模块或 LPM 功能。检查是否已经在 **MegaWizard Plug-In Manager** 中打开 **Generate a Clearbox body**。
2. 使用 **MegaWizard Plug-In Manager** 生成的 Verilog 或 VHDL 设计文件在 EDA 综合工具中对功能进行例化。
3. 在 EDA 综合工具中对设计进行综合和优化。

由于 clear-box 宏功能模块或 LPM 功能包括较详细的信息（所使用的时序信息和器件资源），使用 clear-box 方法时，在 EDA 仿真工具中的仿真速度通常较慢（但对 Quartus II Simulator 没有影响）。此外，clear-box 宏功能模块或 LPM 功能中包括专用器件的详细信息，因此设计中使用不同器件时，需要为新器件重新生成 clear-box 功能。



有关信息

请参阅

宏功能模块的端口和参数列表

如果正在使用 IP 功能，请参阅 IP 文档。有关 Altera 宏功能模块的信息，请参阅 Quartus II Help。

在 EDA 工具中使用 Altera 提供的宏功能模块和 LPM 功能

Quartus II Help 中的“Overview:Creating & Instantiating Altera-Provided Functions in Other EDA Tools”。

Altera 网站上 *Quartus II Handbook* 第 1 卷
“Synplicity Synplify and Synplify Pro Support”

Altera 网站上 *Quartus II Handbook* 第 1 卷
“Mentor Graphics LeonardoSpectrum Support”

Altera 网站上 *Quartus II Handbook* 第 1 卷
“Mentor Graphics Precision RTL Synthesis Support”

Altera 网站上 *Quartus II Handbook* 第 1 卷
“Synopsys Design Compiler FPGA Support”

在 Quartus II 软件中使用 Altera 提供的宏功能模块和 LPM 功能

Quartus II Tutorial 中的 Design Entry 部分

使用 **MegaWizard Plug-In Manager** 和 Altera 提供的宏功能模块及 LPM 功能

Quartus II Help 中的“Overview:Using the MegaWizard Plug-In Manager”

MegaCore 功能和 OpenCore Plus 硬件评估功能

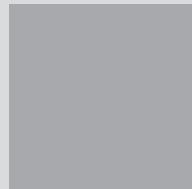
Altera 网站上的 *Application Note 343 (OpenCore Evaluation of AMPP Megafunctions)*

Altera 网站上的 *Application Note 343 (OpenCore Plus Evaluation of Megafunctions)*

Altera 网站上的白皮书 *Using IP Functional Simulation Models to Verify Your System Design*

第三章

约束输入



第3章 内容：

简介	52
使用 Assignment Editor	52
使用 Pin Planner	54
使用 Settings 对话框	56
分配设计分区	57
导入分配	59
验证引脚分配	60

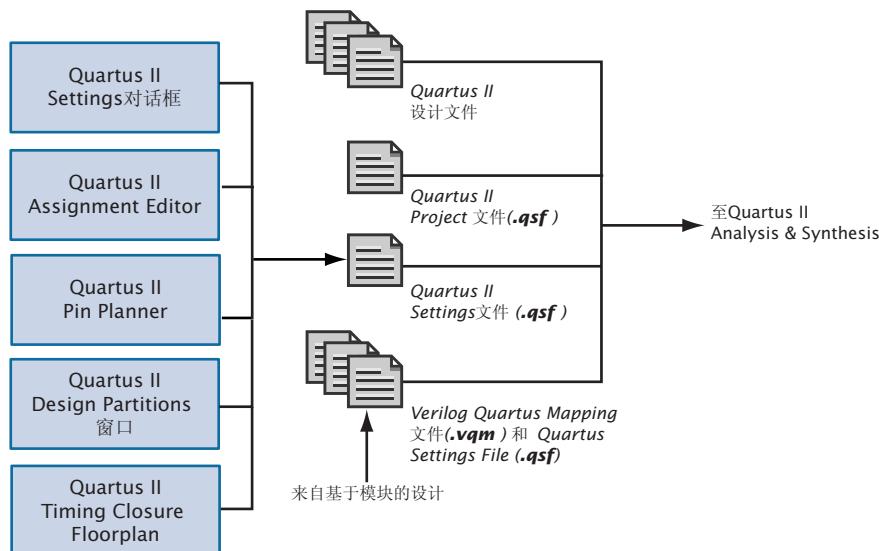
3

简介



建立工程和设计之后，可以使用 Quartus® II 软件中的 **Settings** 对话框 (Assignment 菜单)、Assignment Editor、Pin Planner、Design Partitions 窗口和 Timing Closure 布局图指定初始设计约束条件，例如，引脚分配、器件选项、逻辑选项和时序约束条件。您可以使用 **Import Assignments** 命令 (Assignments 菜单) 导入分配，或者 **Export** 命令 (File 菜单) 导出分配。还可以使用 Tcl 命令或脚本从其它 EDA 综合工具中导入分配。Quartus II 软件还提供 **Timing** 向导 (Assignments 菜单)，协助用户指定初始设计的约束条件。许多设置通过 MAX+PLUS® II 快捷菜单中的 **Assign** 命令使用，并在 Assignment Editor 和 Settings 对话框中进行设置。图 1 是约束和分配输入流程。

图 1. 约束和分配输入流程



使用 Assignment Editor



Assignment Editor 界面用于在 Quartus II 软件中建立、编辑节点和实体级别分配。分配用于在设计中为逻辑指定各种选项和设置，包括位置、I/O 标准、时序、逻辑选项、参数、仿真和引脚分配。

使用 Assignment Editor 可以选择分配类别；使用 Quartus II Node Finder 选择要约束的特定节点和实体；显示有关特定约束的信息；添加、编辑或删除选定节点的分配。可以使用或者禁止单独分配，还可以向分配添加备注。

以下步骤描述使用 Assignment Editor 进行分配的基本流程：

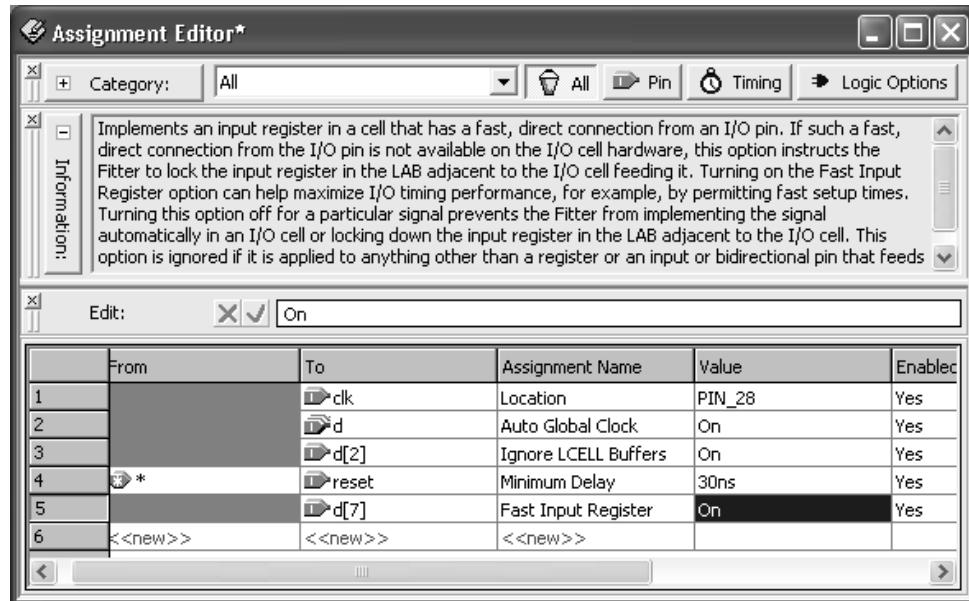
1. 打开 Assignment Editor。
2. 在 **Category** 栏中选择相应的分配类别。
3. 在 **Node Filter** 栏中指定相应的节点或实体，或使用 **Node Finder** 对话框查找特定的节点或实体。
4. 在显示当前设计分配的电子表格中，添加相应的分配信息。

Assignment Editor 中的电子表格提供适用的下拉列表，也可以键入分配信息。当您添加、编辑和删除分配时，Messages 窗口中将出现相应的 Tcl 命令。

可以使用 **Export** 命令 (File 菜单) 将数据从 Assignment Editor 导出到 Tcl Script File (.tcl) 或者 Comma-Separated Value File (.csv) 中。还可以使用 **Import Assignments** 命令 (Assignments 菜单) 从 CSV 或者文本文件中导入分配数据。关于导入分配的详细信息，请参阅第 59 页“**导入分配**”。

建立和编辑分配时，Quartus II 软件对适用的分配信息进行动态验证。如果分配或分配值非法，Quartus II 软件不会添加或更新数值，而是转换为当前值或不接受该值。当您查看所有分配时，Assignment Editor 将显示适用于当前器件为当前工程而建立的所有分配，但当您分别查看各个分配类别时，Assignment Editor 将仅显示与所选特定类别相关的分配。

图 2. Quartus II Assignment Editor



有关信息

使用 Assignment Editor

请参阅

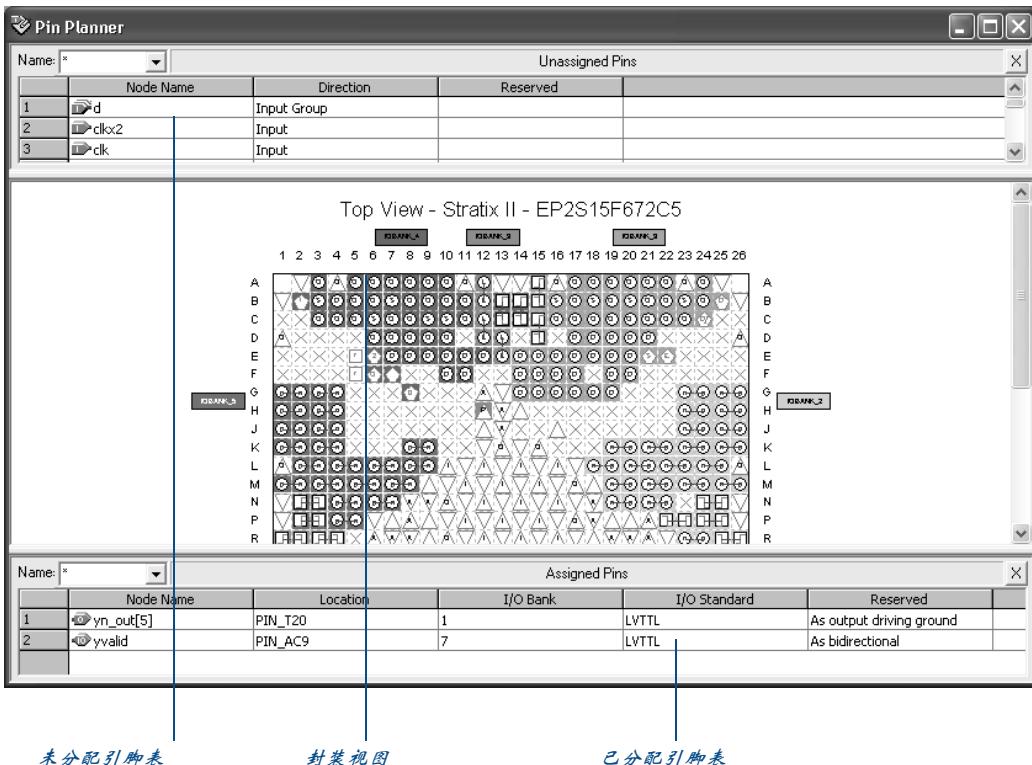
Altera 网站上 *Quartus II Handbook* 第 2 卷
“Assignment Editor”

Quartus II Help 中的 “Assignment Editor Introduction” 和 “Overview: Making Assignments”

使用 Pin Planner

Assignments 菜单下的 Pin Planner 视觉工具为引脚和引脚组分配提供了另一种途径。它包括器件的封装视图，以不同的颜色和符号表示不同类型的引脚，并以其他符号表示 I/O 块。Pin Planner 使用的符号与器件系列数据手册中的符号非常相似。它还包括已分配和未分配引脚的表格。图 3 是 Pin Planner 窗口。

图 3. Pin Planner



默认状态下，Pin Planner 显示未分配引脚的列表，包括节点名称列、方向和类型；器件封装视图；已分配引脚列表，包括节点名称列、引脚位置和 I/O 块。还可以通过将 Unassigned Pins 表中的一个或多个引脚拖至封装图中的可用引脚或 I/O 块来进行引脚分配。在 Assigned Pins 表中，可以滤除节点名称、改变 I/O 标准，指定保留引脚的选项。在 Unassigned Pins 表中，可以改变节点名称或用户加入节点的方向。还可以为保留引脚指定选项。根据需要，可关掉已分配和未分配引脚列表。

在 Pin Planner 窗口中，可调整视图，放大或缩小视图。选择是否显示 I/O 块、VREF 组、可分配 I/O 引脚或者差分引脚对连接等。还可以显示所选引脚的属性和可用资源，以及 Pin Planner 中说明不同颜色和符号的图例。

有关信息

请参阅

使用 Pin Planner 分配引脚

Altera 网站上 *Quartus II Handbook* 第 2 卷 “I/O Planning using the Pin Planner”

使用 Settings 对话框



您可以使用 **Settings** 对话框 (Assignments 菜单) 为您的工程指定分配和选项。您可以设置一般工程范围的选项以及综合、适配、仿真、和时序分析选项。

使用 **Settings** 对话框可以执行以下类型的任务：

- ◇ **修改工程设置**: 为工程和修订信息指定和查看当前顶层实体; 从工程中添加和删除文件; 指定自定义的用户库; 为封装、引脚数量和速度等级指定器件选项; 指定移植器件。
- ◇ **指定 EDA 工具设置**: 为设计输入 / 综合、仿真、时序分析、板级验证、正规验证、物理综合以及相关工具选项指定 EDA 工具。
- ◇ **指定 Analysis & Synthesis 设置**: 用于 Analysis & Synthesis、Verilog HDL 和 VHDL 输入设置、默认设计参数和综合网表优化选项工程范围内的设置。
- ◇ **指定编译过程选项**: 智能编译选项，在编译过程中保留节点名称，运行 Assembler，以及渐进式编译或综合，并且保存节点级的网表，导出版本兼容数据库，显示实体名称，使能或者禁止 OpenCore® Plus 评估功能。还为生成早期时序估算提供选项。
- ◇ **指定适配设置**: 时序驱动编译选项、Fitter 等级、工程范围的 Fitter 逻辑选项分配，以及物理综合网表优化。
- ◇ **指定时序分析设置**: 为工程设置默认频率，定义各时钟的设置、延时要求和路径排除选项以及时序分析报告选项。
- ◇ **指定 Simulator 设置**: 模式 (功能或时序)、源向量文件、仿真周期以及仿真检测选项。
- ◇ **指定 PowerPlay Power Analyzer 设置**: 输入文件类型、输出文件类型和默认触发速率，以及结温、散热方案要求、器件特性等工作条件。

- ◇ 指定软件构建设置：工具集目录、处理器体系结构和软件工具集、编译器、汇编器和连接器设置。
- ◇ 指定 Design Assistant、SignalTap II、SignalProbe 和 HardCopy 设置：打开 Design Assistant 并选择规则；启动 SignalTap®II Logic Analyzer，指定 SignalTap II File(.stp) 名称；自动布线 SignalProb™ 信号选项，为 SignalProbe 功能修改适配结果的选项；并指定 HardCopy™ 时序选项，生成 HardCopy 文件。



分配设计分区

如果您希望使用渐进式编译或渐进式综合功能，可以将设计的不同等级部分指定为设计分区，进行 Analysis & Synthesis 或者渐进式完整编译，而不会影响工程的其他部分。关于渐进式编译和渐进式综合的详细信息，请参阅以下章节：

- ◇ 第1章 “设计流程”，第28页 “自上而下渐进式编译设计流程”。
- ◇ 第4章 “综合”，第78页 “进行渐进式综合”。
- ◇ 第5章 “布局布线”，第83页 “进行完整的渐进式编译”。

您可以在设计中分配分区，用于渐进式综合或者完整的渐进式编译。Project Navigator 和 Design Partitions 窗口均支持分配设计分区。

在 Project Navigator 中分配设计分区

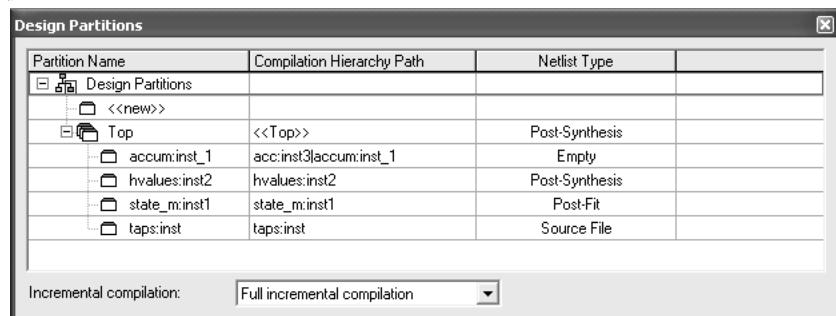
在 Project Navigator 的 Hierarchy 标签选项下，您可以使用 Set as Design Partition 命令（右键弹出菜单）指定所选实体实例为设计分区。当您指定设计的第一个分区时，软件将提示是仅采用渐进式综合还是完整的渐进式编译，还是希望现在禁止使用该功能。

如果您希望为分区进行 LogicLock™ 分配，可以将分区由 Project Navigator 窗口直接拖至 LogicLock Regions 窗口，或者 Timing Closure 布局图中的 LogicLock 区域。

使用 Design 使用 Partitions 窗口分配设计分区

可以 Design Partitions Window 命令 (Assignments 菜单) 将实体指定为设计分区。图 4 所示为 Design Partitions 窗口。

图 4. Design Partitions 窗口



在 Design Partitions 窗口中，可以指定采用以下哪种渐进式编译模式：

- ◇ 仅采用 Incremental Synthesis(渐进式综合)
- ◇ 完整的 Incremental Compilation(渐进式编译)
- ◇ 关闭

如果不希望使用完整层次路径名称，可以在右键弹出菜单中重新命名分区。
还可以选择 Generate Incremental Compilation Tcl Script (右键弹出菜单) 生成一个渐进式编译 Tcl 脚本。

如果所选 Incremental Compilation Mode 为 Full Incremental Design, Design Partitions 窗口也会显示 Netlist Type 列，使您能够为 Netlist Type 指定以下选项之一：

- ◇ Source File directs the 指示 Compiler 由源设计文件编译。
- ◇ Post-Synthesis 为分区保留综合结果 (这是新分区的默认选项)
- ◇ Post-Fit 为分区保留布局结果
- ◇ Empty 为分区跳过编译

通过选择 Netlist Type 列表中的类型或者选择分区并选取 Properties (右键弹出菜单) 来指定网表类型。

如果希望为分区进行 LogicLock 分配，可以将分区由 Design Partitions 窗口直接拖至 LogicLock Regions 窗口，或者 Timing Closure 布局图中的 LogicLock 区域。

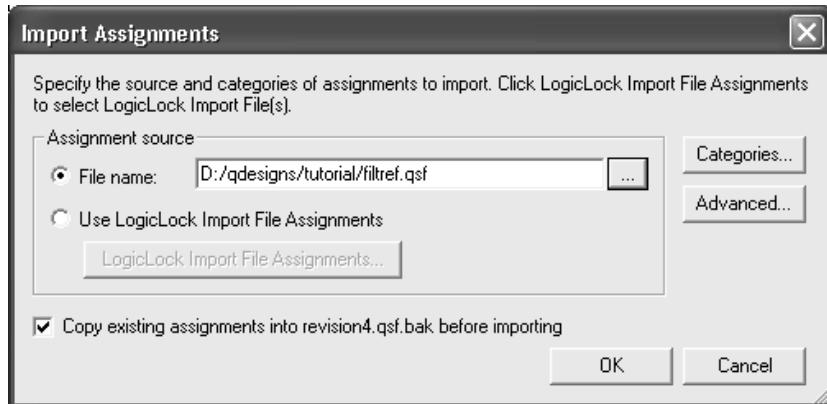


导入分配

Import Assignments 命令 (Assignments 菜单) 允许您从 Quartus II 软件或者 MAX+PLUS II 软件建立的工程中将分配导入到 Quartus II 软件工程中。

Import Assignments 对话框允许您指定含有将要导入分配的文件以及导入到当前工程 QSF 中分配 (Assignment Categories) 的特定类型。单击 **Import Assignments** 对话框中的 **Advanced**，指定导入分配的属性、全局和实例级别，并指定分配如何影响当前设计。使用对话框导入设置文件、**Export** 命令 (File 菜单) 建立的 CSV Files 以及 FPGA Xchange Files (.fx)，还可以在导入分配之前，为设计的当前 QSF 建立备份。请参阅图 5。

图 5. 导入分配对话框



可以使用此命令将 MAX+PLUS II Assignment & Configuration 文件 (包含 MAX+PLUS II 工程分配和设置) 导入到 Quartus II 工程中。也可以使用此命令从 Quartus II 软件建立的其他工程中将设置和分配导入到当前工程中。例如，可以使用此命令将以前 Quartus II 工程中的引脚分配导入到当前 Quartus II 工程中。

关于使用 **Import Assignments** 命令导入 LogicLock™ 区域分配的信息信息，请参阅第6章“基于模块的设计”，第108页“保存自下而上 LogicLock 流程的中间综合结果”。

有关信息	请参阅
导入分配	Quartus II Help 中的“Importing Assignments”

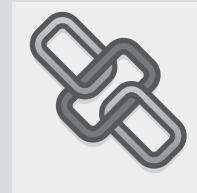
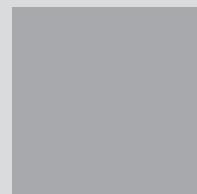
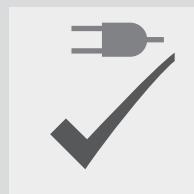
验证引脚分配

Quartus II 软件允许您使用 **Start > Start I/O Assignment Analysis** 命令 (Processing 菜单)，验证引脚分配，包括位置、I/O 块和 I/O 标准分配。可以在设计过程的任何阶段使用此命令来验证分配的准确性，更快地建立最终引出脚。使用此命令无需设计文件，并且可以在设计编译之前验证引出脚。

有关信息	请参阅
导入分配	Altera 网站上 <i>Quartus II Handbook</i> 第2卷“I/O Assignment Planning and Analysis”

第四章

综合



第4章 内容：

简介	62
使用 Quartus II Verilog HDL & VHDL Integrated Synthesis	63
使用其它 EDA 综合工具	66
控制 Analysis & Synthesis	68
使用 Design Assistant 检查设计可靠性	71
使用 RTL Viewer 分析综合结果	73
采用 Technology Map Viewer 分析综合结果	77
进行渐进式综合	78

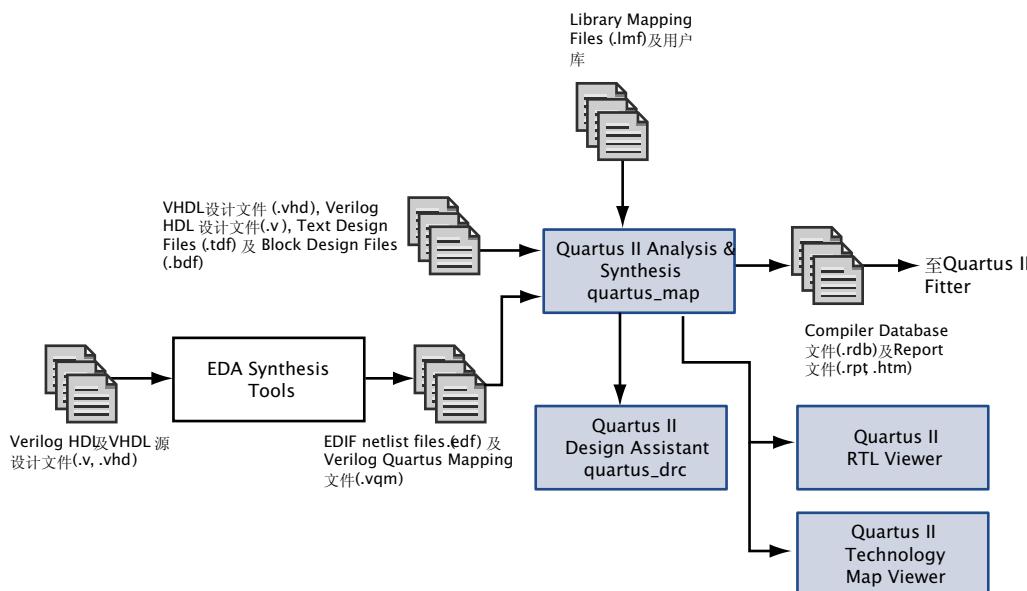
4

简介



您可以使用 Compiler 的 Quartus® II Analysis & Synthesis 模块分析设计文件，建立工程数据库。Analysis & Synthesis 使用 Quartus II Integrated Synthesis 综合 Verilog(.v) 或者 VHDL 设计文件 (.vhd)。根据需要可以使用其它 EDA 综合工具综合 Verilog HDL 或 VHDL 设计文件，然后生成 Quartus II 软件使用的 EDIF 网表文件 (.edf) 或者 Verilog Quartus Mapping File (.vqm)。图 1 是综合设计流程。

图 1. 综合设计流程



您可以在包含 Analysis & Synthesis 模块的 Quartus II 软件中启动完整编译，也可以单独启动 Analysis & Synthesis。Quartus II 软件还允许您在不运行 Analysis & Synthesis 的情况下，进行 Analysis & Elaboration，检查设计的语法错误。

关于启动完整编译或单独启动 Compiler 模块的详细信息，请参阅 第 3 页“图形用户界面设计流程”和第 1 章“设计流程”，第 15 页“命令行设计流程”。

使用 quartus_map 可执行文件

还可以在命令提示符下或在脚本中通过使用 **quartus_map** 可执行文件单独运行 Analysis & Synthesis。如果尚未建立新工程，**quartus_map** 可执行命令将建立新工程。

quartus_map 可执行命令创建可以使用任何文本编辑器查看的独立文本型报告文件。

如果希望获取有关 **quartus_map** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_map -h ↵
quartus_map --help ↵
quartus_map --help=<topic name> ↵
```

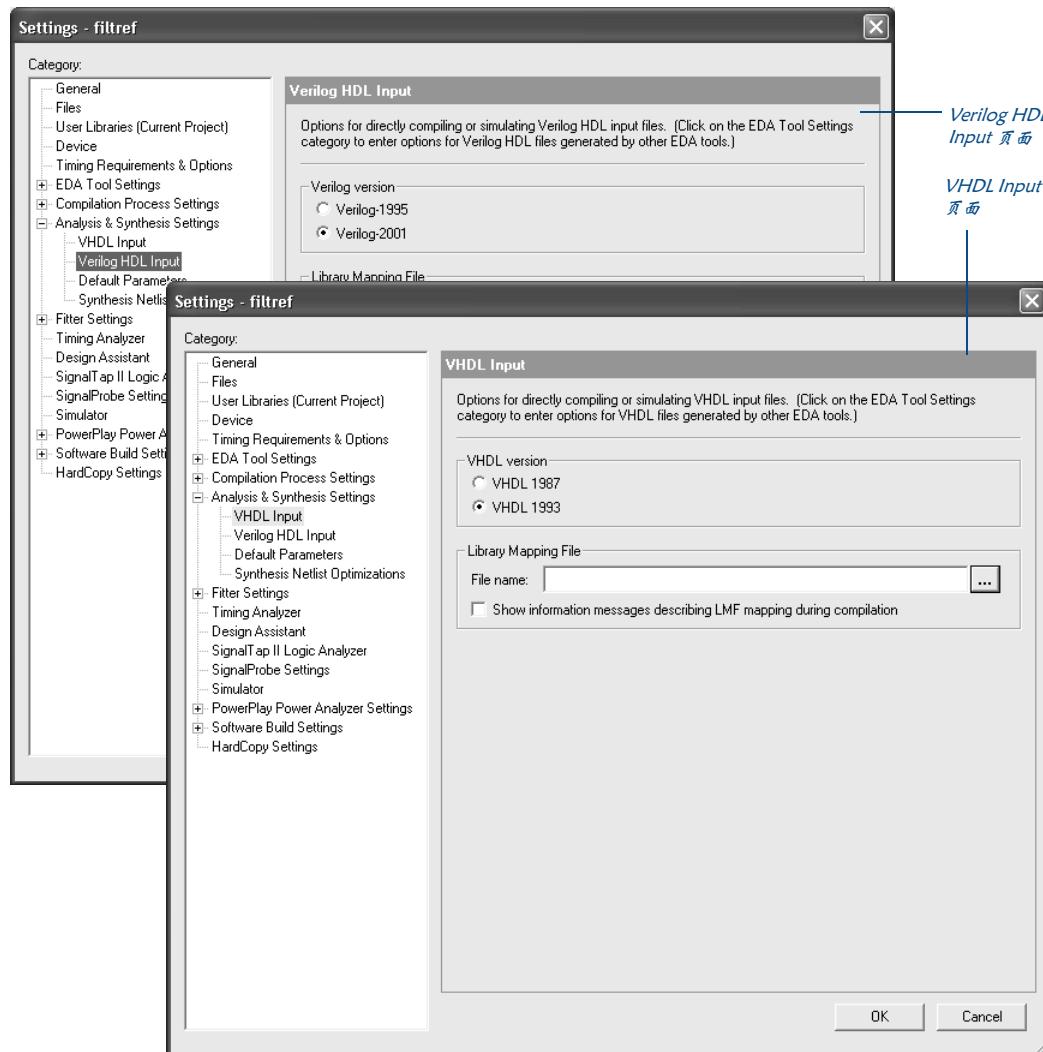
使用 Quartus II Verilog HDL & VHDL Integrated Synthesis



您可以使用 Analysis & Synthesis 分析并综合 Verilog HDL 和 VHDL 设计。Analysis & Synthesis 包括 Quartus II Integrated Synthesis，它完全支持 VHDL 和 Verilog HDL 语言，并提供控制综合过程的选项。

Analysis & Synthesis 支持 Verilog-1995 (IEEE Std. 1364-1995) 和 Verilog-2001 标准 (IEEE Std. 1364-2001)，还支持 VHDL 1987 (IEEE Std. 1076-1987) 和 1993 (IEEE Std. 1076-1993) 标准。可以选择要使用的标准；在默认情况下，Analysis & Synthesis 使用 Verilog-2001 和 VHDL 1993。如果使用另一个 EDA 综合工具，还可以指定 Quartus II 用于将非 Quartus II 功能映射到 Quartus II 功能的 Library Mapping File (.lmf)。您可以指定 Verilog HDL Input 和 VHDL Input 页面的这些选项和其他选项，它们在 Settings 对话框 (Assignments 菜单) 的 Analysis & Synthesis Settings 下。这些页面如 图 2 所示。

图 2. Settings 对话框的 Verilog HDL 和 VHDL Input 页面



大多数 Verilog HDL 和 VHDL 设计可在 Quartus II Integrated Synthesis 和其他 EDA 综合工具中成功实现编译。如果您的设计也要在第三方 EDA 工具中例化 Altera 宏功能模块、参数化模块库 (LPM) 或者知识产权 (IP) 宏功能模块，则需要使用空体文件或者 black-box 文件。但是，在为 Quartus II Integrated Synthesis 例化宏功能模块时，可以不使用 black-box 文件，而直接例化宏功能模块。有关例化参数化功能的详细信息，请参阅 第 46 页 “在 Quartus II 软件中例化宏功能模块” 和 第 2 章 “设计输入”，第 47 页 “在 EDA 工具中例化宏功能模块”。

当您建立 VHDL 和 Verilog HDL 设计时，应将它们添加至工程中。使用 **New Project Wizard** (File 菜单) 或 **Settings** 对话框的 **Files** 页面建立工程时，可以添加设计文件，或者在 Quartus II Text Editor 中编辑文件，保存文件时，系统提示您将其添加至当前工程中。在将文件添加至工程中时，应按照要求 Integrated Synthesis 处理这些文件的顺序来添加。此外，如果您使用 VHDL 设计，可以在 **Files** 页面的 **Properties** 对话框中，指定设计的 VHDL 库。如果不指定 VHDL 库，Analysis & Synthesis 会将 VHDL 实体编译进 **work** 库中。有关将文件添加至工程的详细信息，请参阅第 2 章“设计输入”，第 39 页“建立设计”。

Analysis & Synthesis 构建单个工程数据库，将所有设计文件集成在设计实体或工程层次结构中。Quartus II 软件使用该数据库处理其余工程。其它 Compiler 模块更新该数据库，直到它包含完全优化的工程为止。开始时，该数据库仅包含原始网表；最后，它包含完全优化且适配的工程，用于为时序仿真、时序分析、器件编程等建立一个或多个文件。

建立数据库时，Analysis & Synthesis 的 Analysis 阶段将检查工程的逻辑完整性和一致性，以及边界连接和语法错误。

Analysis & Synthesis 还对设计实体或工程文件的逻辑进行综合和技术映射。它从 Verilog HDL 和 VHDL 中推断触发器、锁存器和状态机。为状态机建立状态分配，并作出能减少资源的选择。此外，它还用 Altera 参数化模块 (LPM) 功能库中的模块替换运算符，例如 + 或 -，该功能已为 Altera 器件做了优化。

Analysis & Synthesis 使用多种算法来减少逻辑门数量，删除冗余逻辑，尽可能地高效利用器件体系结构。可以使用逻辑选项分配来定制综合。Analysis & Synthesis 还应用逻辑综合技术，协助实现工程时序要求，优化设计以满足这些要求。

Messages 窗口和 Report 窗口的 Messages 区域显示 Analysis & Synthesis 生成的任何消息。Status 窗口显示工程编译期间 Analysis & Synthesis 处理的时间。

有关信息	请参阅
Quartus II 软件支持的 Verilog HDL 结构	Quartus II Help 中的“Quartus II Verilog HDL Support”
Quartus II 软件中支持的 VHDL 结构	Quartus II Help 中的“Quartus II VHDL Support”
使用 Quartus II Integrated Synthesis	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷“Quartus II Integrated Synthesis”

使用其它EDA综合工具



可以使用其它EDA综合工具综合VHDL或Verilog HDL设计，然后生成Quartus II软件使用的EDIF网表文件或VQM文件。

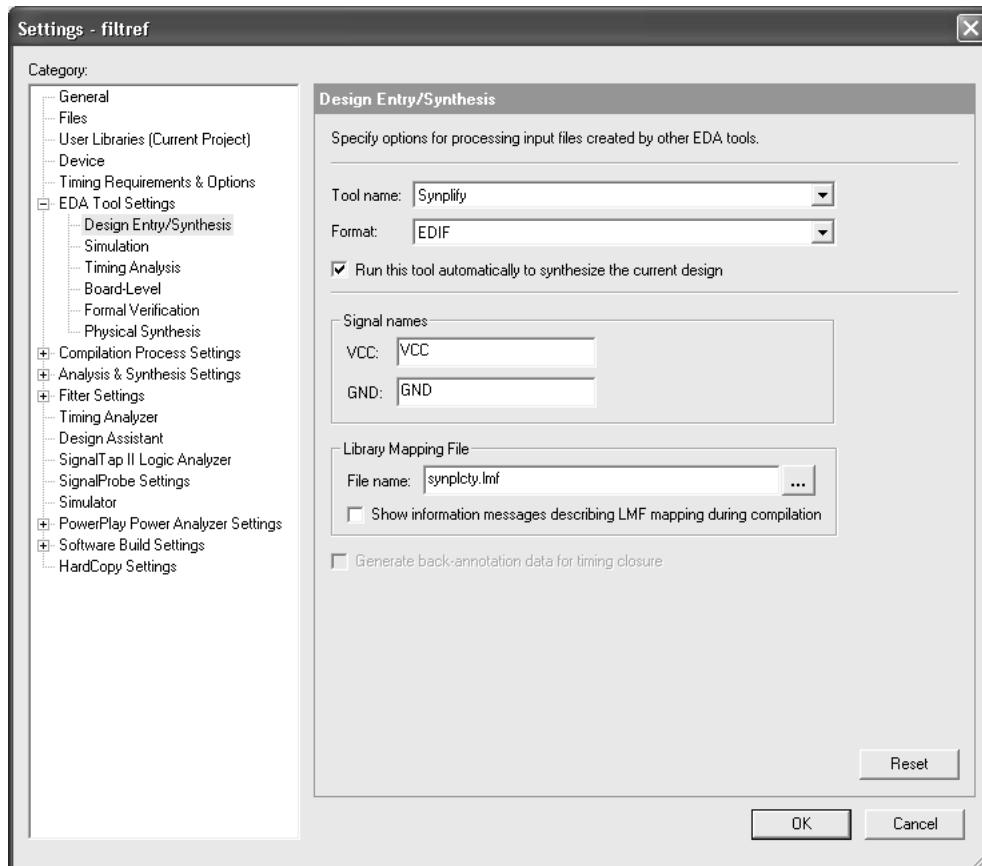
Altera提供多种EDA综合工具使用的库。Altera还为多种工具提供NativeLink®支持。NativeLink技术有助于在Quartus II软件和其它EDA工具之间无缝传递信息，并允许您从Quartus II图形用户界面中自动运行EDA工具。

如果已使用其它EDA工具建立了分配或约束条件，可以使用Tcl命令或脚本将这些约束条件导入到包含您设计文件的Quartus II软件中。许多EDA工具可自动生成分配Tcl脚本。表1列出了Quartus II支持的EDA综合软件。

表1. Quartus II支持的EDA综合工具

综合工具名称	EDIF网表文件(.edf)	Verilog Quartus Mapping文件(.vqm)	NativeLink支持
Mentor Graphics LeonardoSpectrum	✓		✓
Mentor Graphics Precision RTL Synthesis	✓		✓
Synopsys Design Compiler	✓		
Synopsys Design Compiler FPGA	✓		
Synopsys FPGA Compiler II	✓		✓
Synplicity Synplify	✓	✓	✓
Synplicity Synplify Pro	✓	✓	

在Settings对话框(Assignments菜单)EDA Tool Settings下的Design Entry & Synthesis页面中，可以指定将要使用的EDA综合工具，同时还可以指定支持NativeLink的EDA工具是否作为完整编译的一部分在Quartus II软件中自动运行。Design Entry & Synthesis页面还允许您为EDA综合工具指定其它选项。请参阅图3。

图 3. Settings 对话框的 EDA Tool Design Entry & Synthesis 页面

如果您在 **Design Entry & Synthesis** 页面已经指定了一个 EDA 综合工具，则可以通过选择 Quartus II 软件的 **Start > Start EDA Synthesis (Processing 菜单)** 来运行此工具。许多 EDA 工具还允许您从该 EDA 工具的图形用户界面内运行 Quartus II 软件。有关信息，请参阅您的 EDA 工具文档。

有关信息	请参阅
使用 Synplicity Synplify 软件	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Synplicity Synplify 和 Synplify Pro 支持”
使用 Mentor Graphics LeonardoSpectrum 软件	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Mentor Graphics LeonardoSpectrum 支持”
使用 Mentor Graphics Precision RTL Synthesis 软件	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Mentor Graphics Precision RTL Synthesis 支持”
使用 Synopsys FPGA Compiler II 软件	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Synopsys FPGA Compiler II BLIS 和 Quartus II LogicLock 设计流”
使用 Synopsys DC FPGA 软件	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Synopsys Design Compiler FPGA 支持”

控制 Analysis & Synthesis

可以使用以下选项和功能来控制 Quartus II Analysis & Synthesis:

- ◇ Complier 指令和属性
- ◇ Quartus II 逻辑选项
- ◇ Quartus II 综合网表优化选项

使用 Complier 指令和属性

Quartus II 软件支持编译器指令，这些指令也称为编译指示。可以在 Verilog HDL 或 VHDL 代码中包括 translate_on 和 translate_off 等编译器指令作为备注。这些指令不是 Verilog HDL 或 VHDL 命令；但是，综合工具使用它们以特定方式推动综合过程。仿真器等其它工具则忽略这些指令并将它们作为备注处理。

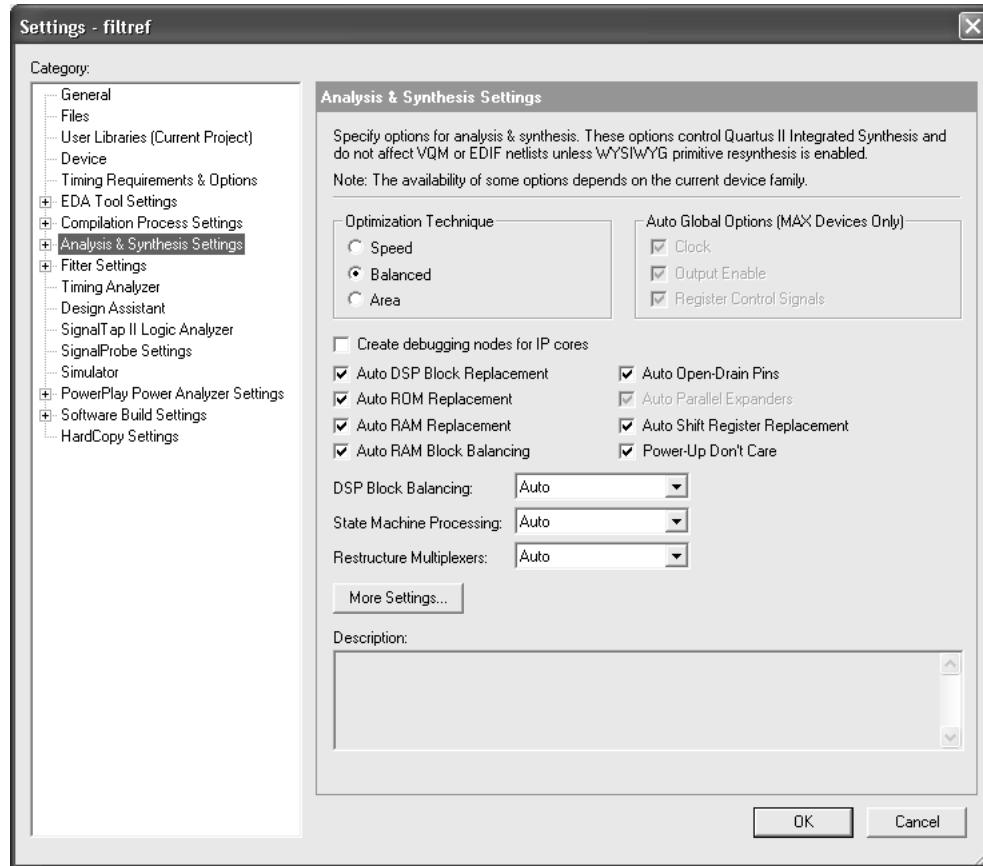
还可以指定属性，这些属性有时称为编译指示或指令，用于推动特定设计单元的综合过程。还提供一些属性，作为 Quartus II 逻辑选项。

有关信息	请参阅
使用编译器指令和属性	QuartusII Help 的“VHDL Language Directives & Attributes”和“Verilog HDL Language Directives & Attributes”
结合 Quartus II Integrated Synthesis 使用 Complier 指令和属性	Altera 网站上 <i>QuartusII Handbook</i> 第 1 卷“Quartus II Integrated Synthesis”

使用 Quartus II 逻辑选项

Quartus II 逻辑选项允许您在不编辑源代码的情况下设置属性。您可以在 Assignment Editor 中分配单独的 Quartus II 逻辑选项，而且可以在 **Settings** 对话框 (Assignments 菜单) **Analysis & Synthesis Settings** 页面中为工程指定全局 Analysis & Synthesis 逻辑选项。请参阅图 4。

图 4. Settings 对话框的 Analysis & Synthesis 设置页面



Analysis & Synthesis Settings 页面的 Quartus II 逻辑选项允许您指定 Complier 应该执行速度优化、面积优化，还是执行“平衡”优化，“平衡”优化努力达到速度和面积的最佳组合。它还提供多种其他选项，例如用来上电的逻辑电平控制选项，删除重复或者冗余逻辑的选项，用 DSP Blocks、RAM、ROM、开漏引脚替换相应逻辑的选项，状态机的编码方式选项，实现多路复用器所需的逻辑单元数量，以及其他影响 Analysis & Synthesis 的选项等。

有关信息	请参阅
使用 Quartus II 逻辑选项控制综合	Quartus II Help 中的“Logic Options”、“Creating, Editing, and Deleting Assignments”和“Specifying Settings for Default Logic Options”
建立逻辑选项分配	Quartus II Tutorial 中的编译部分
使用会影响综合的 Quartus II 综合选项和逻辑选项	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷“Quartus II Integrated Synthesis”

使用 Quartus II 综合网表优化选项

Quartus II 综合优化选项用于在多种 Altera 器件系列的综合阶段优化网表。这些优化选项对标准编译期间出现的优化进行补充，出现在完整编译的 Analysis & Synthesis 阶段。这些优化对综合网表进行更改，通常有利于面积和速度的改善。Settings 对话框 (Assignments 菜单) Analysis & Synthesis Settings 下的 Synthesis Netlist Optimizations 页面，允许您指定网表优化选项，包括以下综合优化选项：

- ◇ 进行 WYSIWYG 基本单元再综合
- ◇ 进行逻辑门级寄存器再定时
- ◇ 允许寄存器再定时，权衡 T_{su}/T_{co} 和 F_{max}

有关综合网表优化选项的详细信息，请参阅 第 9 章“时序逼近”，第 148 页“使用网表优化实现时序逼近”。

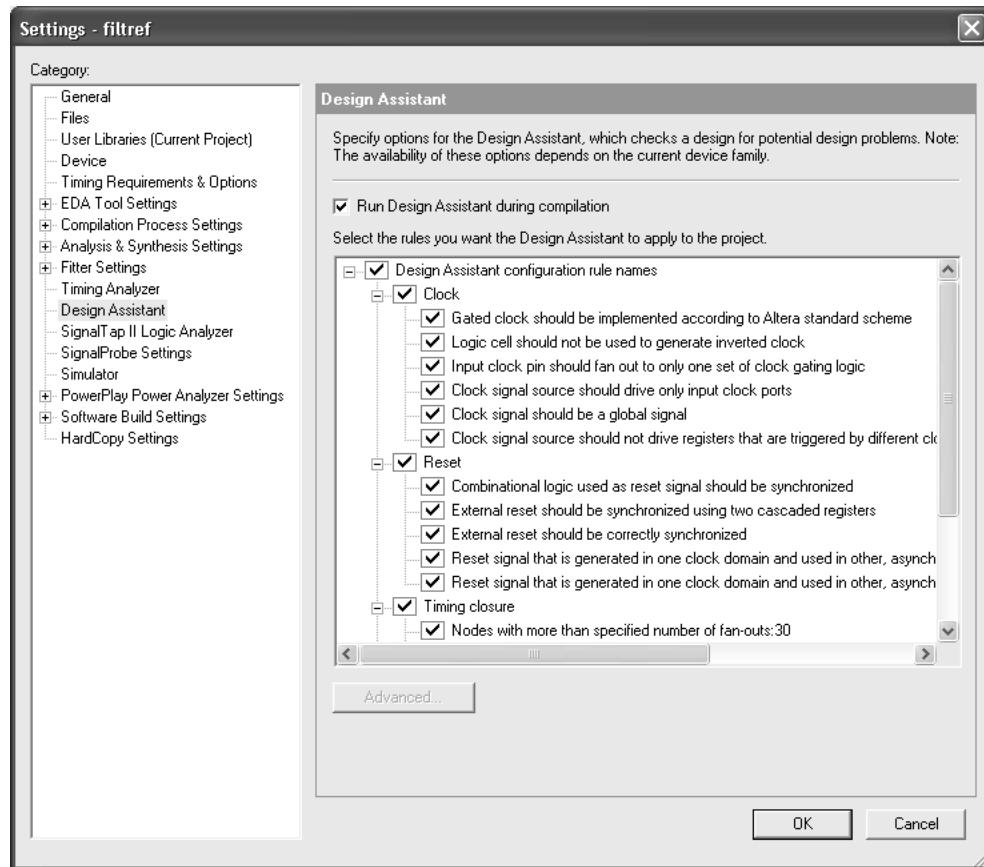
有关信息	请参阅
使用 Quartus II 综合和网表优化选项	Altera 网站上 <i>Quartus II Handbook</i> 第 2 卷“Netlist Optimizations & Physical Synthesis”和“Design Optimization for Altera Devices”。

使用 Design Assistant 检查设计可靠性



Quartus II Design Assistant 依据一组设计规则，检查设计的可靠性。在将设计移植到 HardCopy™ 器件之前，检查设计的可靠性时，Design Assistant 非常有用。Settings 对话框 (Assignments 菜单) 的 Design Assistant 页面用于指定检查设计时所使用的可靠性准则。请参阅 图 5。

图 5. Settings 对话框的 Design Assistant 页面



使用 quartus_drc 可执行文件

还可以在命令提示符下或在脚本中通过使用 **quartus_drc** 可执行文件单独运行 Design Assistant。在运行 Design Assistant 之前，必须先运行 Quartus II Fitter 可执行文件 **quartus_fit**。

quartus_drc 可执行文件建立可以使用任何文本编辑器查看的单独文本型报告文件。

获取有关 **quartus_drc** 可执行文件的帮助信息，请在命令提示符下键入以下命令之一：

```
quartus_drc -h ↵  
quartus_drc -help ↵  
quartus_drc --help=<topic name> ↵
```

还可以遵循较好的同步设计方法和 Quartus II 编码样式准则，提高设计优化。

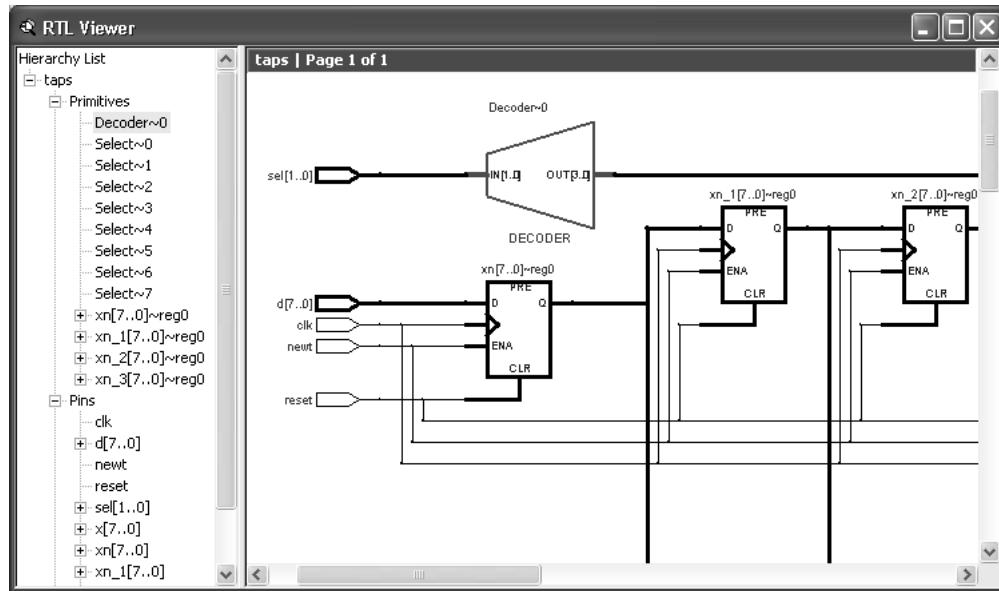
有关信息	请参阅
使用 Quartus II Design Assistant	Quartus II Help 中的“Analyzing Designs with the Design Assistant”和“Overview:Using the Design Assistant”
使用 Quartus II 综合选项，遵循同步设计方法和编码样式准则	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷“Design Recommendations for Altera Devices, Recommended HDL Coding Styles”和“Quartus II Integrated Synthesis”
	Quartus II Help 中的“VHDL, VHDL, and Verilog HDL Style Guide”

使用 RTL Viewer 分析综合结果



Quartus II RTL Viewer 提供您设计的原理示意图。要想为 Quartus II 工程运行 RTL Viewer，您必须首先通过选择 **Start > Start Analysis & Elaboration** (Processing 菜单) 来分析设计。也可以执行 Analysis & Synthesis 或者进行完整编译，因为这些步骤中包括编译流程的 Analysis & Elaboration 阶段。成功执行 Analysis & Elaboration 后，选择 **RTL Viewer** (Tools 菜单) 来显示 RTL Viewer 窗口。RTL Viewer 包括原理图视图，同时也包括层次结构列表，列出整个设计网表的实例、基本单元、引脚和网络。请参阅图 6。

图 6. RTL Viewer 窗口



RTL Viewer 显示 Verilog HDL 或 VHDL 设计和 AHDL Text Design Files(.tdf)、Block Design Files(.bdf)、Graphic Design Files(.gdf)，或者在 Quartus II 软件内进行综合的文件的 Analysis & Elaboration 结果。对于通过其他 EDA 综合工具生成的 VQM Files 或者 EDIF 网表文件，RTL Viewer 显示 WYSIWYG 基本单元基元表征的层次结构。

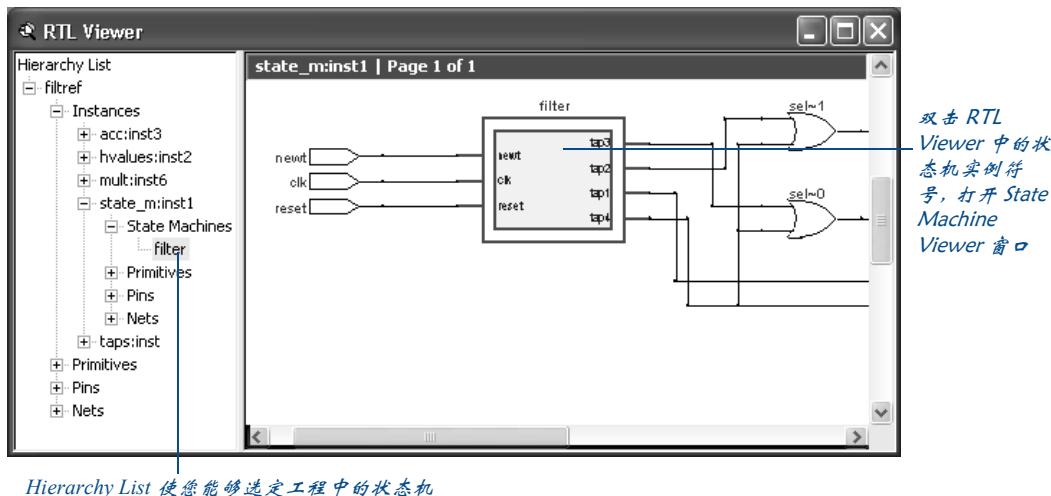
在层次结构列表中，可以选择一个或者多个条目，并在原理图视图中高亮显示，反之亦然。RTL Viewer 允许您调整视图或者放大、缩小视图来集中查看不同层次的细节，通过 RTL Viewer 查找特定名称，在层次结构中上、下移动，或者转到选定网络的来源。如果希望调整扇入和扇出显示，可以将其扩展或者消除。对个别条目，可以使用工具提示来查看节点和源信息。还可以在 RTL Viewer 中选择一个节点，根据该节点的可能位置，在设计文件、Timing Closure 平面布局图、Assignment Editor、Chip Editor、Resource Property Editor、Technology Map Viewer 中找到它。

如果设计规模较大，RTL Viewer 将其分割成多个页面来显示。Options 对话框 (Tools 菜单) 的 **RTL/Technology Map Viewer** 页面允许您指定在每一个页面上控制 RTL Viewer 显示设计数量的选项。通过使用 **Next Page** 和 **Previous Page** 按钮，或者使用 **Go To** 命令 (Edit 菜单)，您可以在 RTL Viewer 中浏览页面。

Filter 命令（右键弹出菜单）允许您过滤视图以便显示所选择节点或网络的驱动源、驱动目标，或者同时显示。还可以过滤视图以显示两个所选节点之间的节点和路径。您选择的每一个过滤都在 RTL Viewer 中建立一个新的过滤页面；可以使用 **Forward** 和 **Back** 按钮浏览设计中的已过滤页面和原始页面。

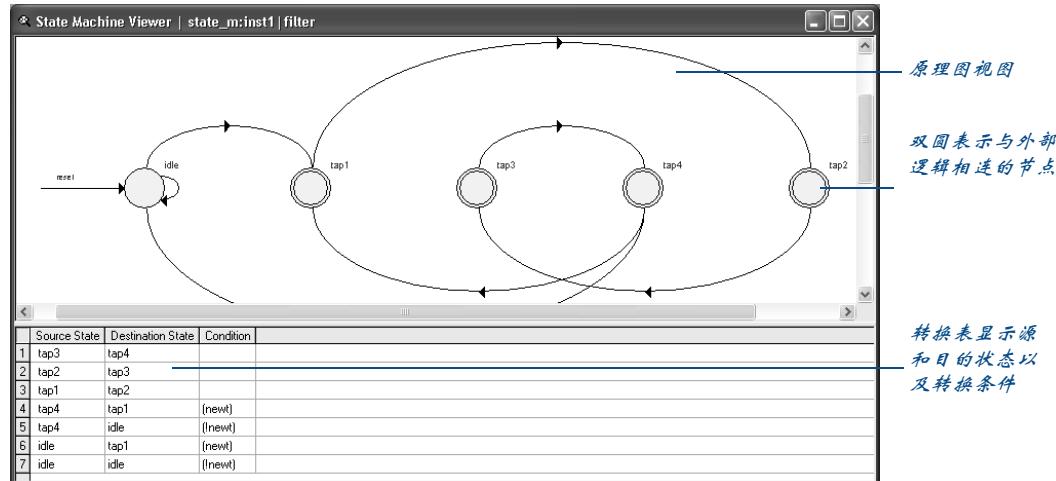
RTL Viewer 提供的 State Machine Viewer 窗口使您能够查看状态机图。如果工程中含有状态机，可以首先打开 RTL Viewer 中的工程，然后双击状态机实例，或者指向状态机实例，选取 **Hierarchy Down**（右键弹出菜单）来打开 State Machine Viewer。请参阅图 7。

图 7. RTL Viewer 窗口中的状态机实例



State Machine Viewer 包括一个原理图视图和一个转换表。参见图 8。

图 8. State Machine Viewer 窗口



在转换表中选择一个单元后，原理图中相应状态或转换高亮显示。同样的，当在原理图中选择一个状态或者转换后，转换表中相应的单元高亮显示。原理视图可以放大和缩小、向上或者向下滚动、高亮显示扇入和扇出。在转换表中，可以将所选单元或者整个表格复制到文本编辑器中。还可以将显示在表栏中的数据对齐、排序。

可以打印包括 State Machine 视图在内的 RTL 视图。如果希望导出 RTL 视图或 State Machine 视图的副本，可以导出 JPEG File Interchange Format 文件 (.jpg) 或 Bitmap 文件 (.bmp) 格式的全部或部分图像副本。还可以将副本保存至 Clipboard，在其他图形或绘图程序中以 Graphics Interchange Format 文件 (.gif)、JPEG 文件或 BMP 文件格式保存；也可以作为 Enhanced Metafile (.emf) 粘贴在 Microsoft Word 文档中。

在采用 RTL Viewer 查看设计后，如果决定修改设计，则应该再次执行 Analysis & Elaboration，在 RTL Viewer 中分析更新后的设计。

有关信息

使用 Quartus II RTL Viewer

请参阅

Altera 网站上 *Quartus II Handbook* 第 1 卷
“Analyzing Designs with the Quartus II RTL
Viewer and Technology Map Viewer”

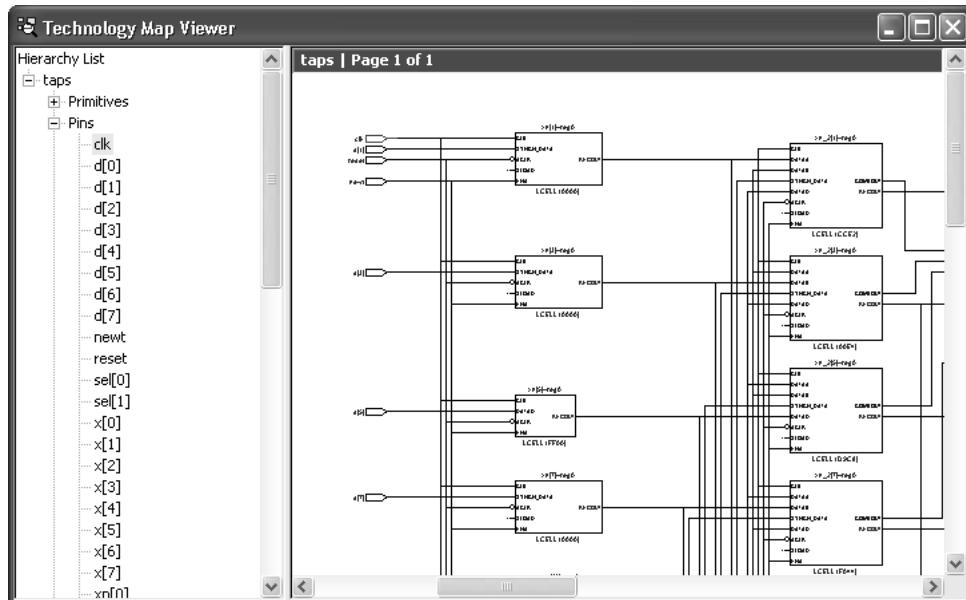
Quartus II Help 中的 “Overview: Viewing RTL
Schematics”

采用 Technology Map Viewer 分析综合结果



Quartus II Technology Map Viewer 提供设计的底级或基元级专用技术原理表征。要为 Quartus II 工程运行 Technology Map Viewer，必须首先进行 Analysis & Synthesis，或者进行完整编译。成功进行 Analysis & Synthesis 之后，可通过选择 Technology Map Viewer (Tools 菜单) 来显示 Technology Map Viewer 窗口。Technology Map Viewer 包括一个原理视图以及一个层次列表，列出整个设计网表的实例、基本单元、引脚和网络。请参阅图 9。

图 9. Technology Map Viewer 窗口



在 Technology Map Viewer 中，可以选择层次列表中的一个或多个条目，高亮显示原理视图，反之亦然。Technology Map Viewer 浏览视图的方式与 RTL Viewer 非常相似；请参阅第 73 页“使用 RTL Viewer 分析综合结果”。Technology Map Viewer 中的工具提示显示等式信息，以及节点和源信息。

Timing Analysis 或者进行包括 Timing Analysis 的完整编译之后，还可以使用 Technology Map Viewer 来查看组成时序路径的节点，该路径包括全部时延和各个节点时延的信息。参见第 8 章“时序分析”，第 138 页“使用 Technology Map Viewer”。

有关信息	请参阅
使用 Quartus II Technology Map Viewer	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Analyzing Designs with the Quartus II RTL Viewer and Technology Map Viewer”
	Quartus II Help 中的 “Overview: Viewing RTL Schematics”

进行渐进式综合

渐进式综合是自上而下渐进式编译流程的组成部分，可以将设计中的实体指定为设计分区，在上面逐渐进行 Analysis & Synthesis，而不会影响工程的其他部分。关于渐进式编译设计流程的详细信息，请参考第 1 章 “设计流程”，第 28 页 “自上而下渐进式编译设计流程”。

可以只进行渐进式综合，也可以进行完整的渐进式编译。关于进行完整渐进式编译的详细信息，请参考第 5 章 “布局布线”，第 83 页 “进行完整的渐进式编译”。

设计完成后，渐进式综合只更新重新综合的设计部分，从而缩短了综合时间，减少了对运行时存储器的占用。功能修改并重新综合设计的一部分而不会影响其他部分意味着在没有改动的部分，已寄存和已结合的节点名称保持不变。成功进行工程及其所有分区的 Analysis & Synthesis 之后，单个分区必须合并到一起，作为完整工程的一部分再次进行编译。

渐进式综合有助于分阶段的分区设计，但是对于需要跨越不同层次边界进行优化的工程则用处不大。

下面的步骤描述了为渐进式综合建立设计的基本流程：

1. 进行 Analysis & Elaboration。
2. 将工程的一个或多个实体指定为分区。请参阅 第 3 章 “约束输入”的第 57 页 “分配设计分区”。
3. 确定为 Incremental compilation 选定了 Design Partitions Window (Assignments 菜单) 中或 Settings 对话框 (Assignments 菜单) Compiler Process Settings 页面的 Incremental Synthesis only。

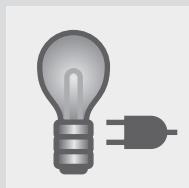
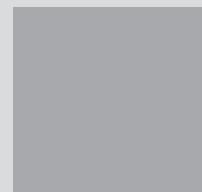
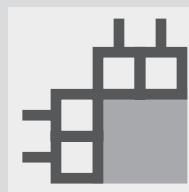
4. 编译工程：

- ✓ 完整编译工程。第一次编译之后，如果对工程进行其他修改，应重新编译工程。重新编译工程时，软件分别对每个改动的分区进行编译，然后自动合并分区，建立一个单一网表。
- ✓ 如果单独运行 Compiler 模块，则进行 Analysis & Synthesis，逐步综合每个分区。当准备将分区合并回整个工程时，在运行 Fitter 和其他 Compiler 模块之前，使用 Start > Start Partition Merge 命令（Processing 菜单）建立一个完整的工程数据库。

有关信息	请参阅
使用 Quartus II Incremental Synthesis 功能	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Quartus II Incremental Compilation”和 “Quartus II Integrated Synthesis”
	Quartus II Help 中的 “Overview: Using Incremental Synthesis”

第五章

布局布线

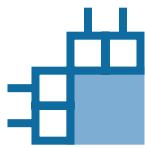


第5章内容:

简介	82
进行完整的渐进式编译	83
分析适配结果	84
优化适配	89
通过反标保留分配	98

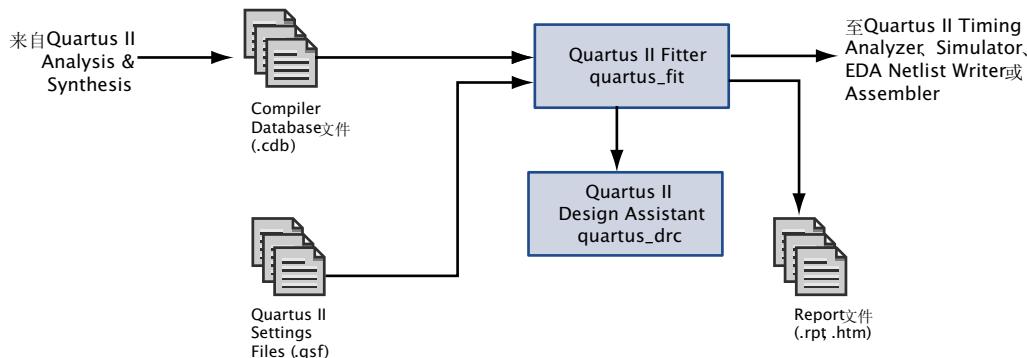
5

简介



Quartus® II Fitter 也称作 PowerFit™ Fitter，执行布局布线功能，在 Quartus II 软件中是指“fitting(适配)”。Fitter 使用由 Analysis & Synthesis 建立的数据仓库，将工程的逻辑和时序要求与器件的可用资源相匹配。它将每个逻辑功能分配给最佳逻辑单元位置，进行布线和时序分析，并选定相应的互连路径和引脚分配。图 1 所示为布局布线设计流程。

图 1. 布局布线设计流程



如果在设计中进行了资源分配，Fitter 尝试将这些资源分配与器件上的资源相匹配，努力满足您已设置的任何其它约束条件，然后试图优化设计中的其余逻辑。如果尚未对设计设置任何约束条件，Fitter 将自动优化设计。如果适配不成功，Fitter 会终止编译，并给出错误信息。

在 **Settings** 对话框 (**Assignments** 菜单) 的 **Compilation Process Settings** 页面，您可以指定是使用正常编译还是智能编译。如果使用“智能”编译，Compiler 将建立详细的数据库，有助于今后的编译更快地运行，但可能会占用额外的磁盘空间。在智能编译之后的重新编译期间，Compiler 将评估自上次编译以来对当前设计所做的更改，然后只运行处理这些更改所需的 Compiler 模块。如果对设计中的逻辑进行任何改动，Compiler 在处理期间将使用所有模块。此选项类似于 MAX+PLUS® II 的 **Smart Recompile** 命令 (**Processing** 菜单)。

可以在包括 Fitter 模块的 Quartus II 软件中启动完整编译，也可以单独启动 Fitter。在单独启动 Fitter 之前，必须成功运行 Analysis & Synthesis。关于运行完整编译的信息，请参考第 1 章“设计流程”，第 3 页“图形用户界面设计流程”。

使用 quartus_fit 可执行文件

还可以在命令提示符下或在脚本中使用 **quartus_fit** 可执行文件来单独运行 Fitter。在运行 Fitter 之前，必须运行 Analysis & Synthesis 可执行文件 **quartus_map**。

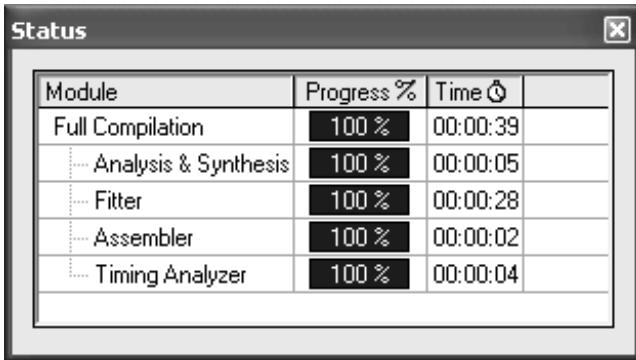
quartus_fit 可执行文件建立了可以使用任何文本编辑器查看的独立文本型报告文件。

获取有关 **quartus_fit** 可执行文件的帮助信息，请在命令提示符下键入以下命令之一：

```
quartus_fit -h ↵  
quartus_fit -help ↵  
quartus_fit --help=<topic name> ↵
```

Status 窗口记录工程编译期间在 Fitter 中处理所花费的时间，以及运行任何其它模块的处理时间。请参阅 图 2。

图 2. Status 窗口



Module	Progress %	Time
Full Compilation	100 %	00:00:39
Analysis & Synthesis	100 %	00:00:05
Fitter	100 %	00:00:28
Assembler	100 %	00:00:02
Timing Analyzer	100 %	00:00:04

进行完整的渐进式编译

完整的渐进式编译使用以前的编译结果，确保只有修改过的设计部分需要进行重新编译，因此能够保持设计性能不变，节省编译时间。完整的渐进式编译是自上而下渐进式编译流程的一部分。更详细的信息，请参阅 第 1 章“设计流程”，第 28 页“自上而下渐进式编译设计流程”。

也可以只进行渐进式综合，或者进行完整的渐进式编译。关于只进行渐进式综合的详细信息，请参阅 第 4 章“综合”，第 78 页“进行渐进式综合”。

以下步骤描述进行完整渐进式编译的基本流程：

1. 进行 Analysis & Elaboration。
2. 将工程的一个或多个实体指定为分区。请参阅第 3 章“约束输入”的第 57 页“分配设计分区”。
3. 选定 Full Incremental compilation 做为 Incremental compilation 模式。
4. 为分区设置合适的 Netlist Type。为保持编译和布局结果，将分区的 Netlist Type 设置为 Post-Fit。
5. 使用 Timing Closure Floorplan 和 LogicLock 分配，为每个分区分配一个器件物理位置。请参阅第 6 章“基于模块的设计”，第 107 页“在自上而下渐进式编译流程中使用 LogicLock 区域”。
6. 进行设置编译，它是设计的一个完整编译。
7. 根据需要，修改设计和设计设置。
8. 再次编译。只有改动过的分区会被重新编译。

有关信息	请参阅
使用 Quartus II 渐进式编译	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Quartus II Incremental Compilation”
	Quartus II Help 中的“Overview: Using Incremental Compilation”

分析适配结果

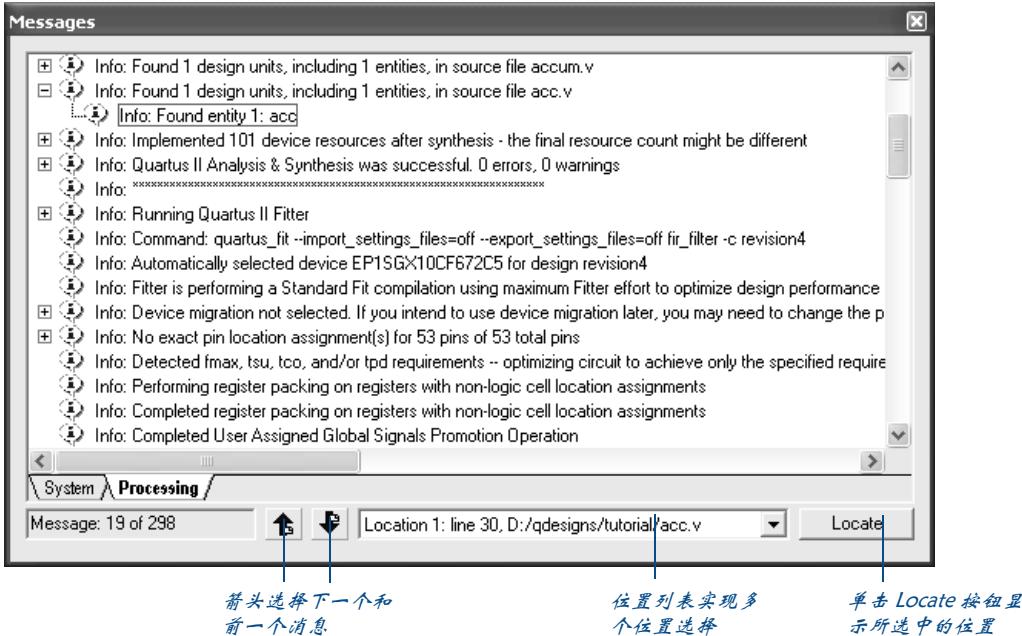
Quartus II 软件提供多种工具来帮助您分析编译和适配结果。Message 窗口和 Report 窗口提供适配结果信息。Timing Closure 平面布局图和 Chip Editor 还可以查看适配结果，进行必要的调整。此外，Design Assistant 根据一组设计规则帮助您检查设计的可靠性。

使用 Messages 窗口查看适配结果



Messages 窗口的 Processing 标签和 Report 窗口或 Report 文件的 Messages 部分显示最近编译或仿真产生的信息。图 3 是 Messages 窗口。

图 3. Messages 窗口



可以从 Messages 窗口的右键弹出菜单中选择 Help 来获取特定消息的帮助。

默认情况下，所有消息类型显示在 Messages 窗口的 Processing 标签下。如果要过滤 Messages 窗口中出现的消息，可以在 Options 对话框（Tools 菜单）Messages 下的 Filtering 标签中设置用于控制警告消息、关键警告消息、信息性消息、额外信息性消息显示的选项。Colors 标签定制每种消息类型的颜色。Options 对话框的 Messages 标签用于指定显示 Processing 标签消息按类型显示的选项：Extra Info、Info、Warning、Critical Warning 以及 Error。消息窗口的右键弹出菜单还提供用于过滤消息和显示不同消息标签的命令。

选定消息，然后从右键弹出菜单中选择一个命令，根据该消息的位置，为 Assignment Editor、Chip Editor、Design File、Resource Property Editor 以及 Timing Closure Floorplan 进行定位。也可以在选取消息之后，从 Message Location 列表中选择一个位置，单击 Locate，定位一个特定位置。

有关信息	请参阅
查看消息	Quartus II Help 中的“Viewing Messages”
定位消息源	Quartus II Tutorial 中的编译部分
	Quartus II Help 中的“Locating the Source of a Message”

使用 Report 窗口或 Report 文件查看适配结果



Report 窗口包含许多部分，可以帮助您对 Fitter 为设计进行布局布线的方式进行分析。它包括多个部分，用于显示资源使用情况。它还列出 Fitter 生成的错误消息，以及正在运行的任何其它模块的消息。

默认情况下，在您运行 Fitter 或其他任何编译或仿真模块时，Report 窗口会自动开启；但是，如果 Options 对话框（Tools 菜单）Processing 页面中适当的 Tool 窗口尚未打开，则在启动处理任务之前，可通过关掉自动开启 Report 窗口，以便指定其不自动开启。同样，如果 Compiler Tool 窗口打开，而 Report 窗口没有自动打开，那么单击每个模块的 Report File 图标则会显示该模块的报告。当 Fitter 正在处理设计时，Report 窗口中的信息将不断更新。如果停止 Fitter，则 Report 窗口将仅显示您停止 Fitter 那一点之前所建立的信息。请参见图 4。

图 4. Report 窗口的 Fitter 部分

Fitter Summary	
Fitter Status	Successful - Tue Mar 01 18:29:25 2005
Quartus II Version	5.0
Revision Name	filtref
Top-level Entity Name	filtref
Family	Stratix II
Device	EP2S15F484C3
Timing Models	Preliminary
Total ALUTs	93 / 12,480 (< 1 %)
Total pins	22 / 343 (6 %)
Total virtual pins	0
Total memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)

根据您在 Options 对话框的 Processing 页面中指定的选项，Quartus II 软件自动生成文本格式和 HTML 格式的 Report 窗口。

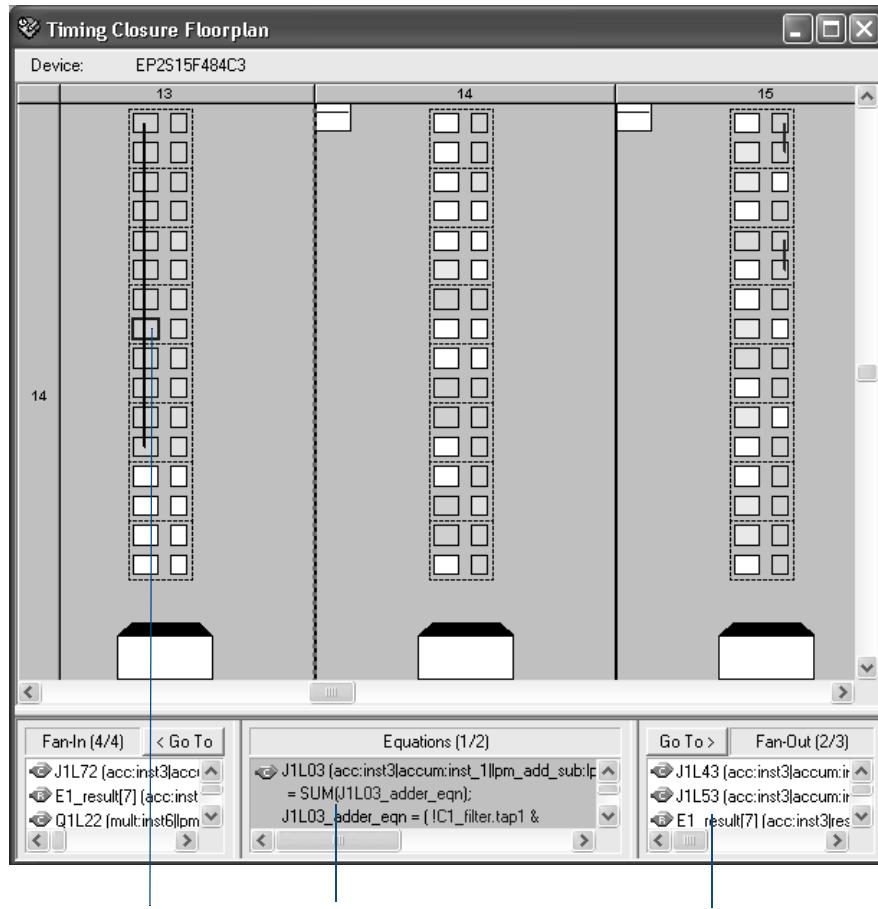
有关信息	请参阅
Report Window 部分	Quartus II Help 中的“Report Window & File Format”
使用 Report Window	Quartus II Help 中的“Overview:Viewing the Results of a Compilation or Simulation in the Report Window”
查看编译报告	Quartus II Tutorial 中的编译部分

使用 Timing Closure Floorplan 分析结果



运行 Fitter 之后，Timing Closure Floorplan 将显示布局布线的结果。此外，可以反标适配结果，以保留上次编译期间的资源分配。由可编辑的 Timing Closure 平面布局图查看 Fitter 以及用户分配执行的逻辑布局，进行 LogicLock™ 区域分配，并查看布线拥塞情况。请参见图 5。

图 5. Timing Closure Floorplan



Timing Closure 平面布局图中的资源使用情况用不同色彩显示。不同颜色代表不同资源，例如，未分配和已分配的引脚和逻辑单元、未布线项、MegaLAB™ 结构、列和行 FastTrack® 扇出。Timing Closure 平面布局图还提供不同的平面布置视图，显示器件的引脚和内部结构。

要在 Timing Closure 平面布局图中编辑分配，可以单击资源分配并将其拖放到新位置。在 Timing Closure 平面布局图中拖放资源时，可以使用橡皮带式生成线显示位置移动所影响的布线资源数量的直观表示。

可以查看设计中的布线拥塞情况，路径的布线延时信息，与指定节点连接的计数。Timing Closure 平面布局图还允许查看特定结构的节点扇出和节点扇入，以及特定节点之间的路径。如有必要，还可以更改或删除资源分配。有关使用 Timing Closure 平面布局图的详细信息，请参阅 第 9 章 “时序逼近”，第 144 页 “使用 Timing Closure Floorplan”。



如果要查看更详细的适配情况，并进行另外的适配调整，Chip Editor 将显示 Timing Closure 平面布局图中没有显示的有关设计布局布线的其它详细信息，并允许使用 Resource Property Editor 和 Chip Editor 进行更改。有关详细信息，请参阅 第 191 页，第 13 章：工程更改管理。

有关信息	请参阅
在 Timing Closure 平面布局图中查看分配和布线信息	Quartus II Help 中的“Overview:Working with Assignments in the Floorplan Editor”和“Overview:Viewing Routing Information”
在 Timing Closure 平面布局图中查看适配	Altera 网站 <i>Quartus II Handbook</i> 第 2 卷“Timing Closure Floorplan”。
在 Chip Editor 中查看适配	Quartus II Tutorial 中的编译器模块 Altera 网站 <i>Quartus II Handbook</i> 第 3 卷“Design Analysis and Engineering Change Management with Chip Editor”。

使用 Design Assistant 检查设计的可靠性



Quartus II Design Assistant 根据一组设计规则检查设计的可靠性，确定是否存在可能影响适配或设计优化的任何问题。Settings 对话框 (Assignments 菜单) 的 Design Assistant 页面用于指定检查设计时要使用的设计可靠性准则。有关详细信息，请参阅 第 4 章 “综合”，第 71 页 “使用 Design Assistant 检查设计可靠性”。

优化适配

运行 Fitter 并分析结果之后，可以试用多种方法来优化适配：

- ◇ 使用位置分配
- ◇ 设置用于控制布局布线的选项
- ◇ 使用 Resource Optimization Advisor
- ◇ 使用 Design Space Explorer

使用位置分配



可以通过使用 Timing Closure 平面布局图或 Assignment Editor 将逻辑分配给器件上的物理资源，例如，引脚、逻辑单元以及逻辑阵列块 (LAB)，以便控制布局布线。您可能想要使用 Timing Closure 平面布局图来编辑分配，因为它为您提供器件及其功能的图形视图。如果要建立新的位置分配，您可能想要使用 **Assignment Editor** 命令 (Assignments 菜单)，因为它允许您同时建立多个特定节点的分配。除了使用 Timing Closure 平面布局图或 Assignment Editor 建立分配外，还可以使用 Tcl 命令。如果要为工程指定全局分配，可以使用 **Settings** 对话框 (Assignments 菜单)。有关指定初始设计约束的详细信息，请参阅 第 51 页，第 3 章：约束输入。

建立分配之后，可以在 Assignment Editor 或 Timing Closure 平面布局图中进行编辑。编译之后，可以使用 Timing Closure 平面布局图来编辑现有资源分配给引脚、逻辑单元、行、列、区域、MegaLAB 结构和 LAB。可以使用 Timing Closure 平面布局图、LogicLock Region 窗口或 LogicLock Region Properties 对话框将节点或实体分配给 LogicLock 区域。

Timing Closure 平面布局图提供器件的不同视图，有助于您对特定位置执行精确的分配。还可以查看等式和布线信息，并可以将分配拖放至 Regions 窗口中的不同区域使分配降级。如果您的设计有太多的约束条件，阻碍了其在器件的适配，您还可以通过删除一些位置分配并让 Fitter 对逻辑进行布局来优化适配。

设置用于控制布局布线的选项

可以设置用于控制 Fitter 并可能影响布局布线的多种选项：

- ◇ Fitter 选项
- ◇ Fitting 优化和物理综合选项
- ◇ 影响适配的个别和全局逻辑选项

设置 Fitter 选项

Settings 对话框 (Assignments 菜单) 的 **Fitter Settings** 页面允许您指定控制时序驱动编译和编译速度的选项。可以指定 Fitter 是否应尽量使用 I/O 单元中的寄存器（而不是使用普通逻辑单元中的寄存器）来满足与 I/O 引脚相关的时序要求和分配。优化设计时，可以指示 Fitter 仅考虑较慢的拐角时序延时，如果优化设计以同时满足所有拐角的时序要求，则同时考虑较快和较慢拐角的时序延时。可指定 Fitter 使用标准适配（它会尽力满足 f_{MAX} 时序要求）、快速适配功能（它可以提高编译速度，但可能降低 f_{MAX} ），还是使用自

动适配功能（在满足时序要求后，可减轻 Fitter 的工作，并能缩短编译时间）。在 **Fitter Settings** 页面中，您可以指定限制 Fitter 仅进行一次尝试（也会降低 f_{MAX} ）。

设置物理综合优化选项

Quartus II 软件允许您设置用来执行物理综合的选项，以便在适配过程中优化网表。在 **Settings** 对话框 (**Assignments** 菜单) **Fitter Settings** 的 **Physical Synthesis Optimizations** 页面中，您可以指定物理综合优化选项。

物理综合优化包括以下选项：

- ◇ 执行组合逻辑的物理综合
- ◇ 寄存器的物理综合：
 - 进行寄存器复制
 - 进行寄存器再定时
- ◇ 物理综合工作级别：
 - 普通（默认工作级别；平均 2 到 3 次编译）
 - 超常（比普通形式花费的编译时间多；可提高性能增益）
 - 快速（比普通形式花费的编译时间少；会降低性能增益）

有关物理综合选项的详细信息，请参阅 第 9 章 “时序逼近”，第 148 页“使用网表优化实现时序逼近”。

有关信息	请参阅
使用 Quartus II 物理综合优化	Altera 网站 <i>Quartus II Handbook</i> 第 2 卷 “Netlist Optimizations & Physical Synthesis”
使用 Quartus II Fitter 优化选项	Quartus II Help 中的 “Using Physical Synthesis”

设置影响布局布线的个别逻辑选项

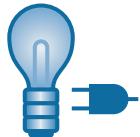
Quartus II 逻辑选项允许您在没有编辑源代码的情况下设置属性。在 **Assignment Editor** (**Assignments** 菜单) 中，您可以为个别节点和实体指定 Quartus II 逻辑选项，而且可以在 **More Fitter Settings** 对话框中指定全局默认逻辑选项，方法是单击 **Settings** 对话框 (**Assignments** 菜单) **Fitter Settings** 页面中的 **More Settings**。例如，可以使用逻辑选项指定信号在器件全局布线路

径上有效，指定 Fitter 应自动建立并行扩展链，指定 Fitter 应自动将寄存器与同一逻辑单元中的组合功能相结合，即所谓的“寄存器组装”，或者限制承载链、串联链和并行扩展链的长度。



使用 Resource Optimization Advisor

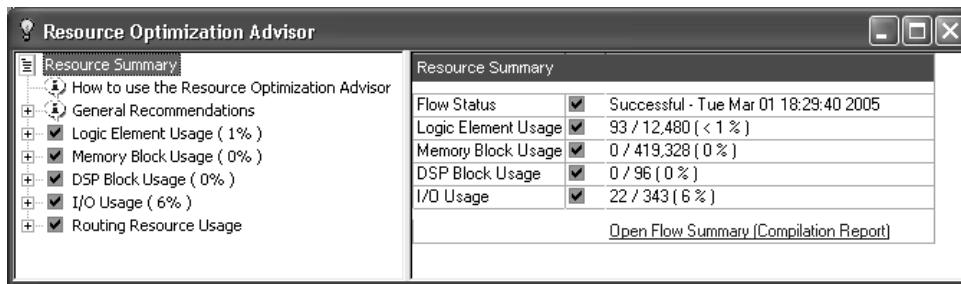
Resource Optimization Advisor 在以下方面为设计资源使用优化提供建议：



- ◇ 逻辑单元使用
- ◇ 存储器块使用
- ◇ DSP 块使用
- ◇ I/O 使用
- ◇ 布线资源使用

在打开的工程中，可以选择 Resource Optimization Advisor (Tools 菜单) 显示 Resource Optimization Advisor。如果还没有编译工程，Resource Optimization Advisor 仅为资源使用优化提供一般建议。工程编译完毕后，Resource Optimization Advisor 在工程信息和当前设置的基础上，可为工程提供特定的建议。图 6 所示为 Resource Optimization Advisor。

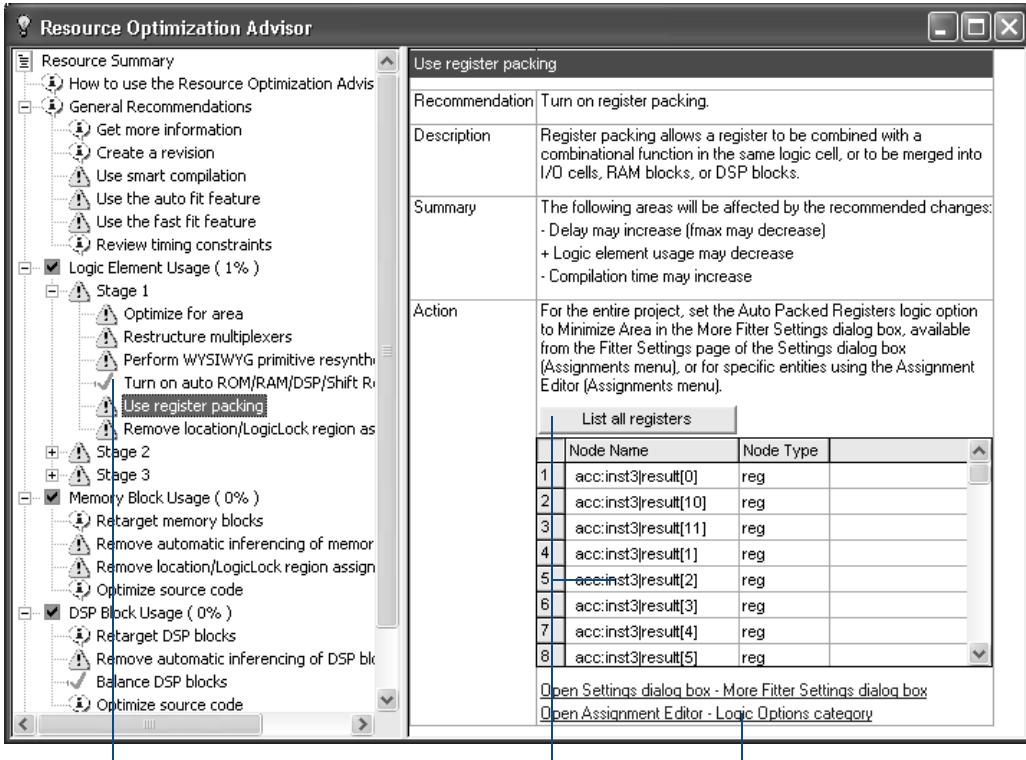
图 6. Resource Optimization Advisor 主页面



Resource Optimization Advisor 的第一页概括了编译后的资源使用情况，指出可能存在问题的区域。Resource Optimization Advisor 左侧显示了问题及其建议的层次列表，图标表示建议是否对当前设计和目标器件系列合适，当

前设计是否已经具有所建议的设置。单击层次列表中的建议时，右侧显示建议的详细说明、概括、当前全局设置，以及一个或多个建议的方法，如图7所示。

图7. Resource Optimization Advisor 建议页面



建议的层次列表——图标表示可能存在问题的区域

一些建议所包含的按钮将提供关于设计的详细信息，比如该列表。

单击建议页面的链接，打开相应的对话框、页面和功能。

如果所建议的方法涉及到修改 Quartus II 设置，Resource Optimization Advisor 的右侧会含有指向 Quartus II 软件相对应对话框、页面和功能的链接，也可能会含有一个提供关于设计更详细信息的按钮，指向 Quartus II Help 或 Altera 网站上其他文档的链接。

如果需要打印当前建议，可以单击右侧，选取 Print Recommendation (右键弹出菜单)；如果需要打印所有建议，则单击左侧，选取 Print All Recommendations (右键弹出菜单)。

如果希望查看建议以提高时序结果，可使用 Timing Optimization Advisor。参见第 9 章“时序逼近”，第 147 页“使用 Timing Optimization Advisor”。

使用 Design Space Explorer



控制 Quartus II 适配的另一种方法是使用 Design Space Explorer (DSE)，它是一个 Tcl 脚本 (*dse.tcl*)，有助于设计优化。DSE 界面可自动搜索一定范围内的 Quartus II 选项和设置，确定要使工程获得最佳结果应使用哪个设置。可以在命令提示符下，运行 `quartus_sh` 可执行文件，来启动 DSE，也可以从 Windows Start 菜单中，选择 `Quartus II <version number> Design Space Explorer` 命令（例如，`Quartus II 5.0 Design Space Explorer` 命令）来启动，还可以在 Quartus II 软件界面中，运行 `Launch Design Space Explorer` 命令来启动 (Tools 菜单)。

可以指定 DSE 对当前工程进行优化设置的努力级别。DSE 界面还允许指定优化目标以及所允许的编译时间。图 8 所示为 Design Space Explorer 的 `Settings` 标签选项。

图 8. Design Space Explorer 的 Settings 标签选项



DSE 提供多个搜索模式，这些模式列在 DSE 窗口的 Exploration Settings 下：

- ◇ 搜索最佳面积
- ◇ 搜索最佳性能（允许您指定努力级别）
- ◇ 高级搜索

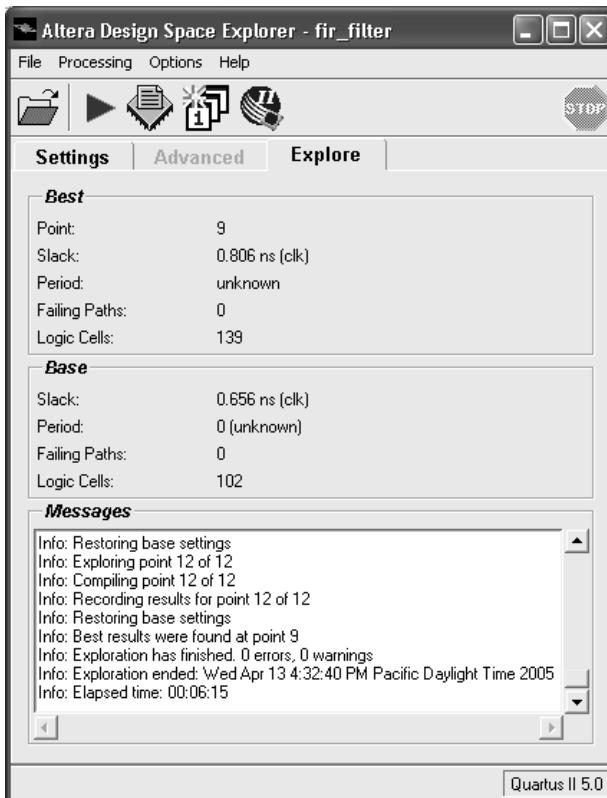
选择 Advanced Search 选项，并打开 Advanced 标签，可指定搜索空间、优化目标和搜索方式的其他选项。图 9 是 Advanced 标签选项。

图 9. Design Space Explorer 的 Advanced 标签选项



指定搜索设置之后，使用 **Explore Space** 命令 (Processing 菜单) 或者按钮来启动搜索。可以在 **Explore** 标签选项下查看搜索结果。图 10 所示为 **Explore** 标签。还可以使用 **View Last DSE Report for Project** 命令 (Processing 菜单) 或者按钮查看文本文件形式的搜索结果。

图 10. Design Space Explorer 的 Explore 选项标签



进行 Design Space Explore

可以在命令提示符下键入以下命令，在图形用户界面模式下运行 DSE:

```
quartus_sh --dse ↵\
```

可以在命令提示符下键入以下命令，在命令行模式下运行带有其他选项的 DSE:

```
quartus_sh --dse -nogui <project name> [-c <revision name>] ↵
```

要获取有关 DSE 选项的帮助，请在命令提示符下键入 quartus_sh --help=dse ↵，或在 DSE 窗口中选择 Show Documentation (Help 菜单)。

许多 Exploration Space 模式允许您指定 DSE 在布局布线和设计中的努力等级；但是，增加努力等级通常要增加编译时间。自定义搜索模式允许指定各种参数、选项和模式，然后搜索它们对设计的影响。

Signature 模式允许搜索单个参数对设计的影响，在 f_{MAX}、延迟、编译时间和面积之间进行权衡。在 Signature 模式下，DSE 测试单个参数对多个种子的影响，然后报告平均值，使您可以评估参数在设计空间内的相互作用。

DSE 同样提供一个 **Optimization Goal** 选项的列表，允许您指定 DSE 是否应该优化面积、速度或者反向迟滞和失败路径。

此外，您可以指定 **Search Method** 选项，对 DSE 在搜索过程中所花费的时间和努力等级提供附加控制。

采样 DSE 完成设计搜索后，可从 DSE 位置建立一个新的修订。然后关闭 DSE，在 Quartus II 软件中以新修订打开工程。



有关信息

请参阅

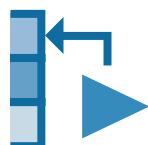
使用 Design Space Explorer

Altera 网站 *Quartus II Handbook* 第 2 卷
“Design Space Explorer”。

优化性能的参数和设置

Altera 网站 *QuartusII Handbook* 第 2 卷
“Design Optimization for Altera Devices”

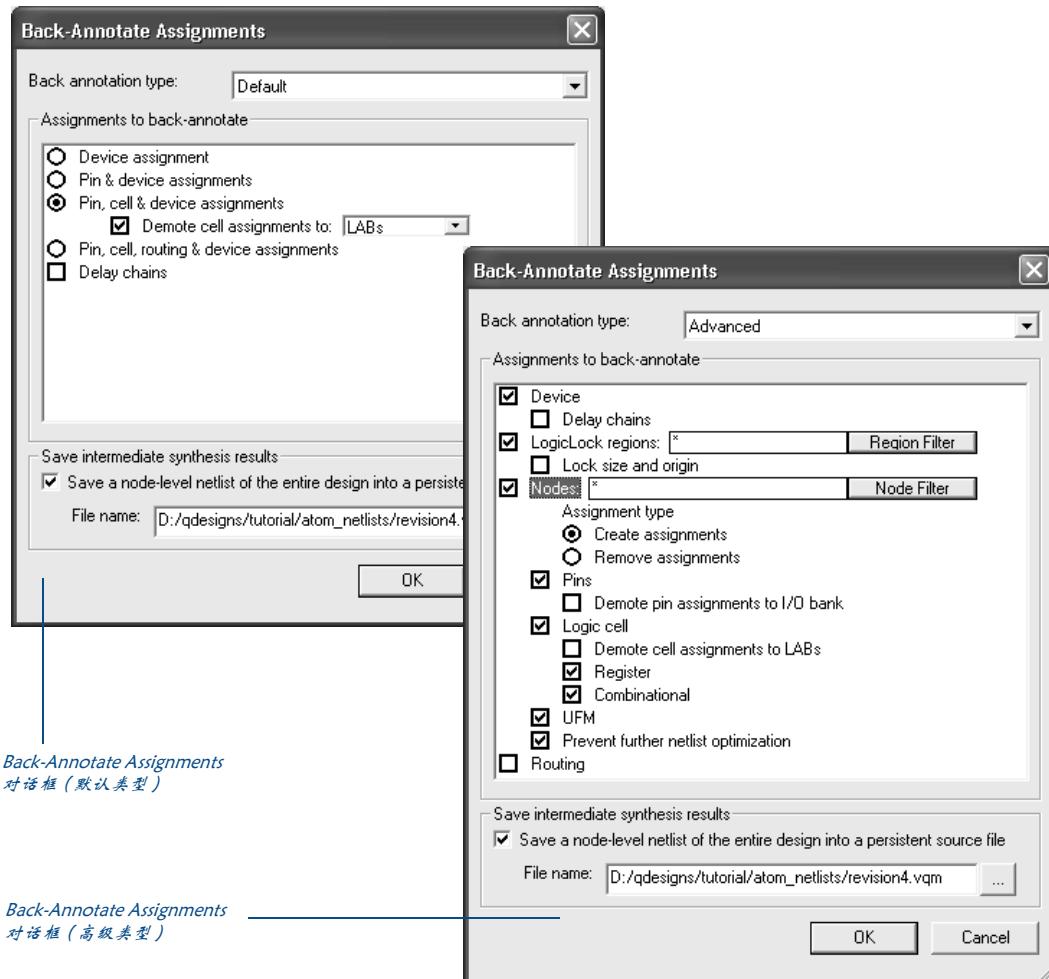
通过反标保留分配



可以通过反标器件资源分配来保留上次编译的资源分配。可以在工程中反标所有资源分配；还可以反标 LogicLock 区域的大小和位置。可以在 **Back-Annotate Assignments** 对话框 (Assignments 菜单) 中指定要反标的分配。

Back-Annotate Assignments 对话框允许选定反标的类型：默认类型或高级类型。请参阅图 11。

图 11. Back-Annotate Assignments 对话框

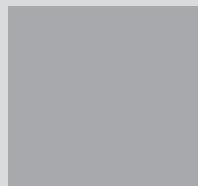
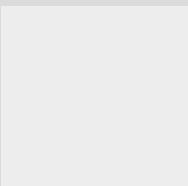
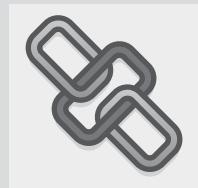


Back-Annotate Assignments (默认类型) 对话框允许将引脚以及逻辑单元分配“降级”为具有较少限制的位置分配，从而使 Fitter 在重新安排布局时具有更多的选择。**Back-Annotate Assignments (高级类型)** 对话框可以执行 Default 反标类型允许的任何操作，并允许反标 LogicLock 区域以及其中的节点和布线（可选）。Advanced 反标类型还提供许多用于根据区域、路径、资源类型等进行过滤的选项，并允许使用通配符。只应使用一种类型的反标，不能两者同时使用。如果不能确定要使用哪种类型，Altera 建议您在大多数情况下都使用 Advanced 反标类型，因为它提供更多的选项，尤其在使用 LogicLock 区域时更是如此。有关使用带有 LogicLock 区域的反标的详细信息，请参阅第 6 章“基于模块的设计”，第 109 页“反标 LogicLock 区域分配”。

有关信息	请参阅
反标位置分配	Quartus II Help 中的“Back-Annotating Assignments for a Project”
反标 LogicLock 区域分配	Quartus II Help 中的“Back-Annotating a LogicLock Region”
反标 LogicLock 布局	Quartus II Tutorial 中的 LogicLock 部分

第六章

基于模块的设计



第 6 章 内容:

简介	102
Quartus II 基于模块的设计流程	102
使用 LogicLock 区域	103
在自上而下渐进式编译流程中使用 LogicLock 区域	107
保存自下而上 LogicLock 流程的中间综合结果	108
LogicLock 与 EDA 工具结合使用	111

6

简介



Quartus® II LogicLock™ 功能支持基于块的设计流程，允许建立模块化设计、单独设计和优化每个模块，然后将每个模块整合到顶层设计中。只要每个模块具有已寄存的输入和输出，这样处理模块便不会影响底层模块的性能。

LogicLock 区域是灵活且可重复使用的约束，能够提高您在目标器件上进行逻辑布局的能力。可以将目标器件上物理资源的任意矩形区域定义为一个 LogicLock 区域。将节点或实体分配给 LogicLock 区域将指示 Fitter 在适配期间将这些节点或实体放置在该区域内。

LogicLock 区域支持面向团队、基于模块的设计，能够单独优化逻辑模块，然后将它们及其布局约束条件导入到规模更大的设计中。LogicLock 方法还能够促进模块的重复使用，因为可以单独开发模块，然后将其约束在 LogicLock 区域之内，供其它设计使用，而不会出现性能劣化，使用户能够充分利用资源，缩短设计周期。

LogicLock 功能还能够将设计分区分配给器件中的物理位置，作为自上而下、渐进式编译流程的一部分。

Quartus II 基于模块的设计流程

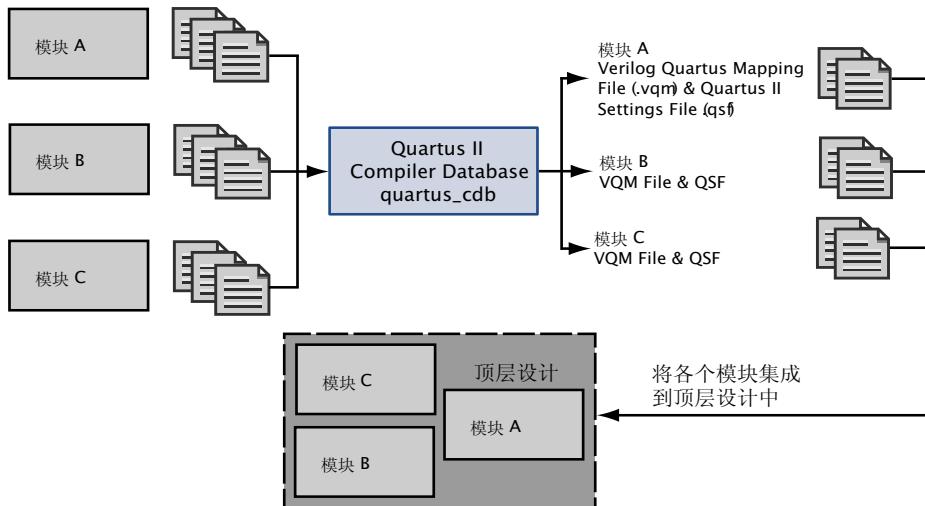
在传统的自上而下的设计流程中，设计只有一个网表。在自上而下的设计流程中，由于每个模块实现方式不同，它们在总体设计中可能具有不同的性能。在自下而上基于模块的设计流程中，每个模块具有单独的网表。这样，设计人员能够建立基于块的设计，每个模块可以单独优化，然后整合到顶层设计中。可以在以下设计流程中使用基于模块的设计：

- ◇ **模块化设计流程：**在模块化设计流程中，将设计划分为对每个子模块进行例化的顶层设计。可以单独开发每个模块，然后将其整合到顶层设计中。布局可以由用户手动决定，也可以由 Quartus II 软件决定。
- ◇ **渐进式编译流程：**在渐进式编译流程中，用户建立并优化系统，然后添加对原始系统性能影响较小或没有影响的后续模块。
- ◇ **团队设计流程：**在团队设计流程中，用户将设计分割为单独的模块，然后在顶层设计中对模块进行例化和连接。其它团队成员单独开发底层模块，为每个模块建立单独的工程，并使用为顶层设计而开发的分配。底层模块完成后，将它们导入到顶层设计中，顶层设计将进行最终编译和验证。

在所有三个设计流程中，均可以通过将设计分割为功能块，保留所在开发层上的性能，将这些功能块按照电路物理结构或关键路径组织起来。图1显示了基于模块的基本设计流程。

图1. 基于模块的设计流程

设计、验证和锁定个别模块



使用 LogicLock 区域

LogicLock 区域按大小（高度和宽度）及其在器件上的位置来定义。可以指定区域的大小和位置，或指示 Quartus II 软件自动建立大小和位置。表1列出了可以在 Quartus II 软件中指定的 LogicLock 区域的主要属性。

表 1. LogicLock 区域属性

属性	取值	行为
状态	浮动或锁定	对于浮动区域，Quartus II 软件可以决定其在器件上的位置。锁定区域的位置由用户定义。在平面布局图中，锁定区域用实线边界线显示，浮动区域的边界线为虚线。锁定区域必须具有固定大小。
大小	自动或固定	自动大小区域允许 Quartus II 软件按区域组成决定其大小。固定区域的形状和大小由用户定义。
保留	开或关	保留属性用于规定 Quartus II 软件是否可以让未分配给区域的实体使用区域中的资源。如果开启保留属性，则只有分配给区域的条目才可以放置在区域的边界内。
软	开或关	软区域更加符合时序要求，它允许一些实体离开某个区域（如果这种离开可以提高总体设计性能的话）。硬区域不允许 Quartus II 软件将内容放置在区域边界之外。
原点	任意平面布局图位置	原点定义 LogicLock 区域在平面布局图中的位置。

采用 LogicLock 设计流程，可以通过声明母区域和子区域来定义一组区域的层次结构。Quartus II 软件将子区域完全放置在母区域的边界内。可以锁定子模块相对于母区域的位置，而无需将母区域限定在器件的锁定位置上。

用户可以使用 Timing Closure 平面布局图、LogicLock Regions Window 命令 (Assignments 菜单)、Project Navigator 的 Hierarchy 选项标签或使用 Tcl 脚本建立和修改 LogicLock 区域。所有 LogicLock 属性和约束条件信息（时钟设置、引脚分配和相对布局信息）均存储在该工程的 Quartus II Settings File (.qsf) 中。

可以使用 Timing Closure 平面布局图建立、编辑 LogicLock 区域分配。可以使用 Create New Region 按钮在 Timing Closure 平面布局图中绘制 LogicLock 区域，然后在平面布局图视图、Node Finder、或 Project Navigator 的 Hierarchy 选项标签中拖放节点。

建立 LogicLock 区域之后，可以使用 LogicLock Regions 窗口查看设计中的所有 LogicLock 区域，包括大小、状态、宽度、高度和原点。还可以编辑和添加新的 LogicLock 区域。请参阅图 2。

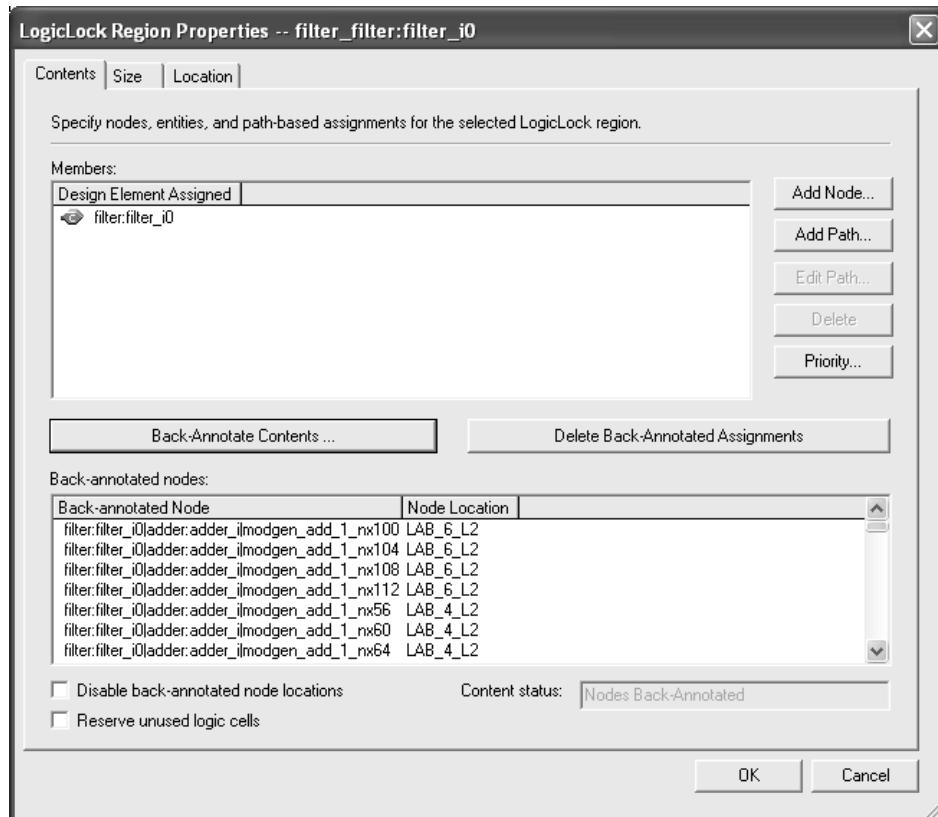
图 2. LogicLock 区域窗口

Region name	Size	State	Width	Height	Origin
LogicLock Regions					
Root_region	Fixed	Locked	36	22	X0_Y0
<<new>>					
state_m:inst1	Fixed	Locked	1	1	LAB_X4_Y5
taps:inst	Auto	Floating	1	1	LAB_X4_Y5
mult:inst6	Fixed	Locked	6	20	LAB_X1_Y1
hvalues:inst2	Fixed	Locked	1	1	LAB_X4_Y5
acc:inst3 accum:inst_1	Fixed	Locked	12	14	LAB_X1_Y1

还可以使用 **LogicLock Regions Properties** 对话框编辑现有 LogicLock 区域，打开 **Back-Annotate Assignments** 对话框，在 LogicLock 区域中反标所有节点，查看设计中 LogicLock 区域的信息，确定包含非法分配的区域。

此外，还可以将基于路径的分配（依据源节点和目标节点）、通配符分配以及基于路径和通配符分配的 Fitter 优先级添加到 LogicLock 区域中。设置优先级用于指定 Quartus II 软件解决基于路径和通配符分配冲突的顺序。可以从 **LogicLock Region Properties** 对话框中打开 Priority 对话框。请参阅图 3。

图 3. LogicLock Region Properties 对话框



在进行分析和详细描述或完整编译之后，Quartus II 软件在 Project Navigator 的 **Hierarchy** 选项标签中显示设计的层次结构。可以在此视图中单击任何设计实体，并由这些实体建立新的 LogicLock 区域，或将它们拖放到 Timing Closure 平面布局图中的现有 LogicLock 区域内。

Altera 还可在命令行或 Quartus II Tcl Console 窗口中提供 LogicLock Tcl 命令，用来分配 LogicLock 区域组成。可以使用所提供的 Tcl 命令建立浮动和自动大小的 LogicLock 区域、在区域中添加节点或层次结构、保留层次结构边界、反标布局结果、导入和导出区域、保存中间综合结果。

有关信息	请参阅
LogicLock 与 Quartus II 软件结合使用	Altera 网站 <i>Quartus II Handbook</i> 第 2 卷 “LogicLock Design Methodology”
	Quartus II Help 中的 “Overview:Using LogicLock Regions”
	Quartus II Tutorial 中的 LogicLock 部分

在自上而下渐进式编译流程中使用 LogicLock 区域

如果要进行完整的渐进式编译，则必须给设计分区分配器件物理位置。可以将设计分区从 Project Navigator 窗口的 **Hierarchy** 标签选项、Design Partitions 窗口或者 Node Finder 中直接拖放至 LogicLock Regions 窗口或者 Timing Closure 平面布局图中，将其分配至 LogicLock 区域。

Altera 建议在设计中，为每个分区建立一个 LogicLock 区域。当这些区域全部达到固定大小、固定位置后，可以逐渐实现最佳性能。理想情况下，应使用 Timing Closure 平面布局图，手动分配 LogicLock 区域，指定器件中的物理位置；也可以通过设置 LogicLock 区域 **Size** 选项为 **Auto**，**State** 选项为 **Floating**，让 Quartus II 软件在一定程度上自动分配 LogicLock 区域至物理位置。如果分区含有许多存储器或 DSP 块，建议将其放置在 LogicLock 区域之外。初次编译之后，应反标 LogicLock 区域属性（不是节点的属性），以确保所有 LogicLock 区域具有固定大小和固定位置。该过程将建立初始平面布局图分配，能够根据需要更方便的进行修改。

初始化或建立编译之后，Altera 建议将 **Size** 设置为 **Fixed**，以产生更好的 f_{MAX} 结果。如果器件利用率较低，增大 LogicLock 区域的大小，使 Fitter 更灵活的进行布局，会产生更好的最终结果。

进行渐进式编译时，适配和综合结果以及设计分区的设置保存在工程数据库中。

关于分配设计分区的详细信息，请参阅第 3 章 “约束输入”的第 57 页 “分配设计分区”。关于渐进式编译的详细信息，请参阅第 1 章 “设计流程”，第 28 页 “自上而下渐进式编译设计流程” 和第 5 章 “布局布线”，第 83 页 “进行完整的渐进式编译”。



有关信息

请参阅

结合 LogicLock 区域使用 Quartus II 渐进式编译

Altera 网站 *Quartus II Handbook* 第 2 卷
“LogicLock Design Methodology”

Altera 网站 *Quartus II Handbook* 第 1 卷
“Quartus II Incremental Compilation”

Quartus II Help 中的 “Overview: Using Incremental Compilation”

保存自下而上 LogicLock 流程的中间综合结果

可以通过为设计中的实体建立 Verilog Quartus Mapping 文件 (.vqm)，结合自下而上的 LogicLock 设计流程，保存各实体的综合结果，以及相应的 QSF，该文件包含实体的 LogicLock 约束信息。



仅为自下而上 LogicLock 设计流程保存中间综合结果

只有使用自下而上的 LogicLock 设计流程时，应将中间综合结果保存至 VQM 文件，而在 LogicLock 区域内使用自上而下的渐进式编译流程时，则不应保存这些结果。自上而下的渐进式编译流程在工程数据库中保存综合和适配结果。

可以设计自定义逻辑模块，或者例化预验证知识产权 (IP) 模块，为该模块做分配、验证功能和性能、锁定该模块以保持其布局和性能不变，然后将要导入的模块导出至另一个设计中。这样，就可以单独设计、测试和优化模块，在将模块集成进规模更大的设计时，其性能可以保持不变。

此外，通过将中间综合结果保存至 VQM 文件，以及导入分配时用工程中的 VQM 文件替换实体，可以确保新工程中被综合的节点名称与导入分配中的节点名称相对应。

以下步骤介绍了保存中间综合结果为 VQM 文件、反标分配以及导出和导入用于设计（包含 LogicLock 区域）的 QSF 的基本流程：

1. 建立 LogicLock 区域。
2. 编译设计。
3. 使用 Back-Annotate Assignments (Advanced 类型) 对话框 (Assignments 菜单)，将逻辑布局锁定在 LogicLock 区域。

4. 通过使用 **Export Assignments** 对话框 (Assignments 菜单)，将 LogicLock 区域分配导出至 QSF。
5. 通过使用 **Import Assignments** 对话框 (Assignments 菜单)，将 VQM 文件中的模块例化进顶层设计，并导入 LogicLock 区域分配。单击 **LogicLock Import File Assignments**，指定含有 LogicLock 区域分配的 QSF 名称、所导入的 QSF 中的实体名称，以及进行分配的设计中的实体名称。

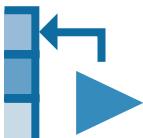
 使用 quartus_cdb 可执行文件

还可以将中间综合结果转存为 VQM 文件，反标分配并在命令提示符下或在脚本中通过 **quartus_cdb** 可执行文件分别导出和导入 LogicLock 区域。

若要获取有关 **quartus_cdb** 可执行文件的帮助信息，请在命令提示符下键入以下命令之一：

```
quartus_cdb -h ↵  
quartus_cdb --help ↵  
quartus_cdb --help=<topic name> ↵
```

反标 LogicLock 区域分配



在导出分配用于顶层设计之前，可以使用 **Back-Annotate Assignments** (Advanced 类型) 命令将逻辑布局锁定在设计的 LogicLock 区域中。在将区域及其分配导入到顶层设计中时，使用反标可以保持 LogicLock 区域的性能不变。

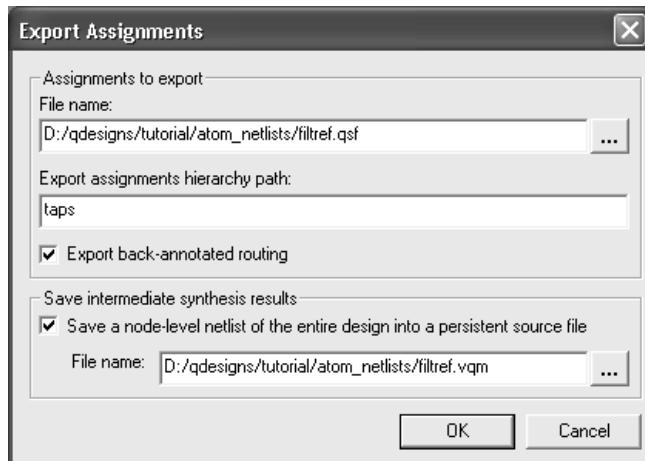
必须使用 **Back-Annotate Assignments** (Advanced 类型) 命令反标 LogicLock 区域分配，也可以使用它反标不包含 LogicLock 区域分配的设计。有关反标分配的详细信息，请参阅 第5章“布局布线”，第98页“通过反标保留分配”。

导出与导入 LogicLock 分配

Export Assignments 和 **Import Assignments** 对话框 (Assignments 菜单) 使您能够使用 LogicLock 区域分配来单独优化实体，并在顶层设计中对这些实体进行例化时保留优化结果。

在导出 LogicLock 区域分配时，Quartus II 软件写入所有 LogicLock 区域分配、其它 QSF 分配和 I/O 标准分配，这些分配应用于在 **Export Assignments** 对话框中指定 QSF 的特定实体实例。默认情况下，Quartus II 软件为整个设计导出 LogicLock 区域分配。可以指定要在 **Export assignments hierarchy path** 对话框中导出的子设计实体。请参见图 4。

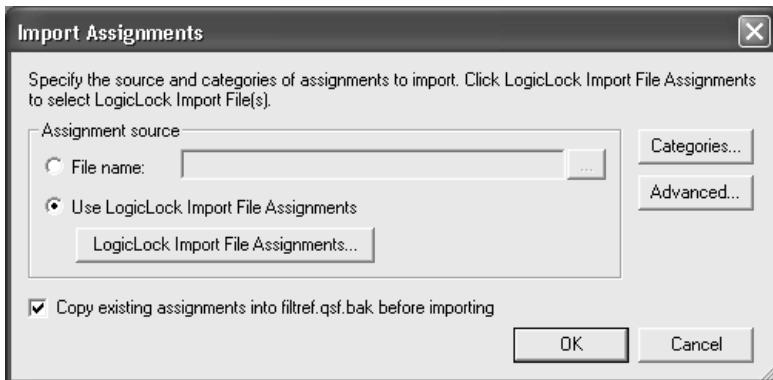
图 4. Export Assignments 对话框



在导入 LogicLock 区域分配时，Quartus II 软件从当前编译项开始，跨越编译层次结构。如果当前工程包含底层实体的多个实例，Quartus II 软件将对为该底层实体而导入的分配进行例化，每个实例进行一次例化。

为防止布局冲突，Quartus II 软件将导入的顶层 LogicLock 区域分配在浮动位置上。但是，它保留所导入子区域相对于母区域的位置。图 5 显示了 Import Assignments 对话框。

图 5. Import Assignments 对话框



导入 LogicLock 区域时，单击 Import Assignments 对话框的 Advanced，可以指定导入分配的属性，其全局或实例等级，以及该分配如何影响当前设计。还可以在导入分配之前，为设计建立当前 QSF 的备份。

有关信息	请参阅
将中间综合结果转存为 VQM 文件、反标注分配以及导出和导入 LogicLock 区域分配	Altera 网站 <i>Quartus II Handbook</i> 第 2 卷 “LogicLock Design Methodology”
	Quartus II Help 中的 “Overview: Saving Intermediate Synthesis Results” 和 “Overview: Using LogicLock Regions”
	Quartus II Tutorial 中的 LogicLock Module

LogicLock 与 EDA 工具结合使用



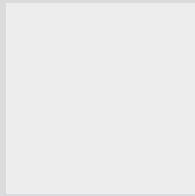
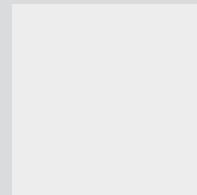
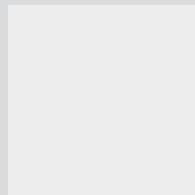
基于模块的 LogicLock 设计流程支持在 EDA 设计输入和综合工具中建立、优化，然后作为单独模块导入到 Quartus II 软件中的模块。使用 EDA 设计输入和综合工具为设计层次结构中的模块建立单独的网表文件 (EDIF Input Files (.edf) 或者 VQM Files)，还能够保持网表文件中逻辑模块不变。然后，可以使用 Quartus II 软件将每个网表文件或网表文件中的模块放入顶层设计中的单独 LogicLock 区域中。一旦进入 Quartus II 软件中，就可以使用 EDA 工具更改、优化、再综合设计中的特定模块，更新设计中的相应部分，而不影响设计中的其它模块。

Mentor Graphics LeonardoSpectrum、Synplicity Synplify、Synopsys FPGA Compiler II 和 Mentor Graphics Precision RTL Synthesis 软件提供定制功能，可以在基于模块的 LogicLock 设计流程中使用这些工具。

有关信息	请参阅
LogicLock 与 EDA 综合工具结合使用	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Synplicity Synplify and Synplify Pro Support”
	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Mentor Graphics LeonardoSpectrum Support”
	Altera 网站 <i>Quartus II Handbook</i> 第 1 卷 “Mentor Graphics Precision RTL Synthesis Support”

第七章

仿真



第7章 内容:	
简介	114
使用 EDA 工具进行设计仿真	115
使用 Quartus II Simulator 进行仿真设计	121

7

简介



可以使用 EDA 仿真工具或 Quartus® II Simulator 对设计进行功能与时序仿真。

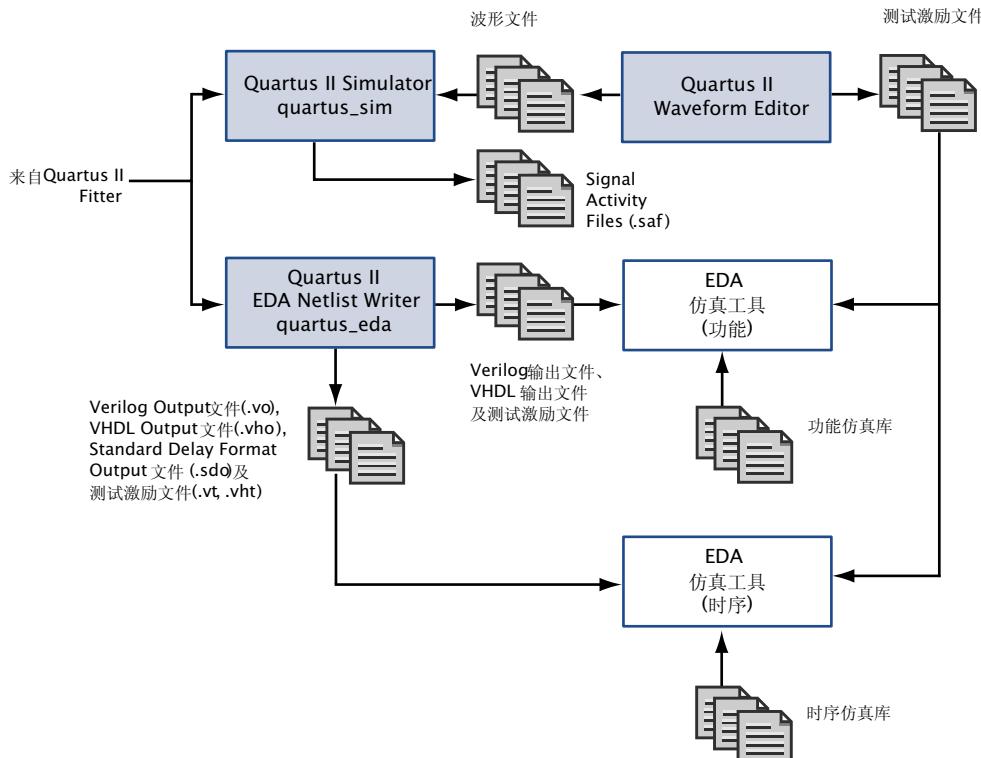


Quartus II 软件提供以下功能，用于在 EDA 仿真工具中进行设计仿真：

- ◇ NativeLink® 集成 EDA 仿真工具
- ◇ 生成输出网表文件
- ◇ 功能与时序仿真库
- ◇ 生成测试激励模板和存储器初始化文件
- ◇ 生成 Signal Activity Files (.saf)

图 1 显示了使用 EDA 仿真工具和 Quartus II Simulator 的仿真流程。

图 1. 仿真流程



使用EDA工具进行设计仿真



Quartus II 软件的 EDA Netlist Writer 模块生成用于功能或时序仿真的 VHDL Output 文件 (.vho) 和 Verilog Output 文件 (.vo)，以及使用 EDA 仿真工具进行时序仿真时所需的 Standard Delay Format Output 文件 (.sdo)。Quartus II 软件生成 Standard Delay Format 2.1 版的 SDF 输出文件。EDA Netlist Writer 将仿真输出文件放在当前工程目录下的特定工具目录中。



此外，Quartus II 软件通过 NativeLink 功能为时序仿真和 EDA 仿真工具提供无缝集成。NativeLink 功能允许 Quartus II 软件将信息传递给 EDA 仿真工具，并具有从 Quartus II 软件中启动 EDA 仿真工具的功能。

表 1 列出了 Quartus II 软件支持的 EDA 仿真工具，并指明哪种工具支持 NativeLink 功能。

表 1. Quartus II 支持的 EDA 仿真工具

仿真工具名称	NativeLink 支持
Cadence Verilog-XL	
Cadence NC-Verilog	✓
Cadence NC-VHDL	✓
Mentor Graphics® ModelSim®	✓
Mentor Graphics ModelSim-Altera	✓
Synopsys Scirocco	✓
Synopsys VCS MX	✓
Synopsys VCS	✓
Synopsys VSS	



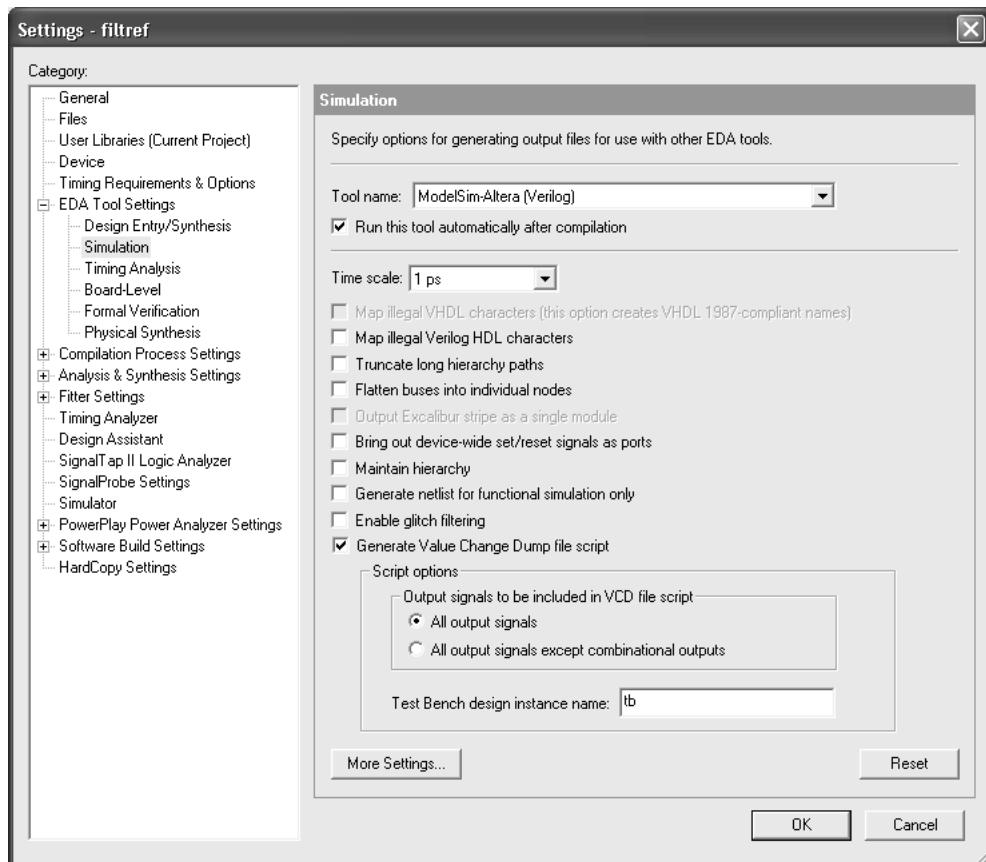
ModelSim-Altera 软件

Altera® 设计软件订购包中含有 Model Technology ModelSim-Altera 软件，支持功能仿真和 HDL 测试激励。

进行 EDA 仿真工具设置

建立一个新工程时，或者在 **Settings** 对话框 (Assignments 菜单) **EDA Tool Settings** 下的 **Simulation** 页面中，可以在 **New Project Wizard** (File 菜单) 中选择 EDA 仿真工具。**Simulation** 页面允许您选择仿真工具并为 Verilog 和 VHDL 输出文件及其对应 SDF 输出文件的生成指定选项，以及功耗分析和 Signal Activity File (.saf) 的选项。图 2 显示了 **Settings** 对话框的 **Simulation** 页面。

图 2. *Settings* 对话框的 *EDA Tool Simulation* 页面



生成仿真输出文件



可以运行 EDA Netlist Writer 模块，并通过指定 EDA 工具设置、编译设计，生成 Verilog 和 VHDL 输出文件。如果已在 Quartus II 软件中编译设计，可以在 Quartus II 软件中指定不同的仿真输出设置（例如，不同的仿真工具），然后使用 Start > Start EDA Netlist Writer 命令（Processing 菜单）重新生成 Verilog 和 VHDL 输出文件。如果您正在使用 NativeLink 功能，也可以使用 Run EDA Simulation Tool 命令（Tools 菜单）进行初始编译后运行仿真。

使用 quartus_eda 可执行文件

还可以在命令提示符下或在脚本中使用 **quartus_eda** 可执行文件单独运行 EDA Netlist Writer。

quartus_eda 可执行文件建立可以使用任何文本编辑器查看的独立文本型报告文件。

若要获取有关 **quartus_eda** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_eda -h ↵  
quartus_eda --help ↵  
quartus_eda --help=<topic name> ↵
```

Quartus II 软件还可以生成以下类型的输出文件，在 EDA 仿真工具中进行功能和时序仿真时使用：

- ◇ **Power Estimation 数据：**您可以使用 EDA 仿真工具执行包括功耗估算数据在内的仿真。可以让 Quartus II 软件将 Verilog HDL 或 VHDL 输出文件中的设计包含功耗估算数据。EDA 仿真工具生成 Power Input File (.pwf)，可以在 Quartus II 软件中使用该文件估算设计的功耗。
- ◇ **Test Bench 文件：**可以使用 Export 命令（File 菜单），由 Quartus II Waveform Editor 的 Vector Waveform File (.vWF) 建立 Verilog Test Bench 文件 (.vt) 和 VHDL Test Bench 文件 (.vht)，用于 EDA 仿真工具。Verilog HDL 和 VHDL Test Bench Files 是测试台模板文件，包含顶层设计文件的例化和来自 Vector Waveform File 的测试向量。如果在 Vector Waveform File 中指定预期值，还可以生成自检测试激励文件。
- ◇ **Memory Initialization 文件：**可以使用 Quartus II Memory Editor 在 Memory Initialization File (.mif) 或者 Hexadecimal (Intel 格式) 文件 (.hex) 中输入内容可寻址存储器 (CAM)、RAM 或 ROM 等存储器模块的初始内容。然后，可以将存储器内容导出为 RAM Initialization File (.rif)，与 EDA 仿真工具一起用于功能仿真。

- ◇ **Signal Activity 文件**：可以生成 Signal Activity Files，与 PowerPlay Power Analyzer 一起使用。Signal Activity File 含有设计的触发速率和静态随机数据。可以对信号有效性周期进行限制，还可以指定进行干扰滤除。

EDA 仿真流程

使用 NativeLink 功能，可以让 Quartus II 软件编译设计，生成相应的输出文件，然后使用 EDA 仿真工具自动进行仿真。也可以在编译之前（功能仿真）或编译之后（时序仿真），在 Quartus II 软件中手动运行 EDA 仿真工具。

功能仿真流程

可以在设计流程中的任何阶段进行功能仿真。以下步骤描述使用 EDA 仿真工具进行设计功能仿真时所需要的基本流程。有关特定 EDA 仿真工具的详细信息，请参阅 Quartus II Help。若要使用 EDA 仿真工具进行功能仿真，请执行以下步骤：

1. 首先在 EDA 仿真工具中设置工程。
2. 建立工作库。
3. 使用 EDA 仿真工具编译相应功能仿真库。
4. 使用 EDA 仿真工具编译设计文件和测试激励文件。
5. 使用 EDA 仿真工具进行仿真。

NativeLink 仿真流程

可以使用 NativeLink 功能，按照以下步骤，使 EDA 仿真工具可以在 Quartus II 软件中自动设置和运行。以下步骤描述将 EDA 仿真工具与 NativeLink 功能结合使用的基本流程：

1. 通过 **Settings** 对话框 (**Assignments** 菜单) 或在工程设置期间使用 **New Project Wizard** (**File** 菜单)，在 Quartus II 软件中进行 EDA 工具设置。
2. 在进行 EDA 工具设置时开启 **Run this tool automatically after compilation**。

3. 在 Quartus II 软件中编译设计。Quartus II 软件执行编译，生成 Verilog HDL 或 VHDL 输出文件以及相应的 SDF 输出文件（如果您正在执行时序仿真），并启动仿真工具。Quartus II 软件指示仿真工具建立工作库；将设计文件和测试激励文件编绎或映射到相应的库中；设置仿真环境；运行仿真。

手动时序仿真流程

如果要加强对仿真的控制，可以在 Quartus II 软件中生成 Verilog HDL 或 VHDL 输出文件以及相应的 SDF 输出文件，然后手动启动仿真工具，进行仿真。以下步骤描述使用 EDA 仿真工具进行 Quartus II 设计时序仿真所需要的基本流程。有关特定 EDA 仿真工具的详细信息，请参阅 Quartus II Help。

1. 通过 **Settings** 对话框 (**Assignments** 菜单) 或在工程设置期间使用 **New Project Wizard** (**File** 菜单)，在 Quartus II 软件中进行 EDA 工具设置。
2. 在 Quartus II 软件中编译设计，生成输出网表文件。Quartus II 软件将该文件放置在特定工具目录中。
3. 启动 EDA 仿真工具。
4. 使用 EDA 仿真工具设置工程和工作目录。
5. 编译或映射到时序仿真库，使用 EDA 仿真工具编译设计和测试激励文件。
6. 使用 EDA 仿真工具进行仿真。

仿真库

Altera 为包含 Altera 专用组件的设计提供功能仿真库，并为在 Quartus II 软件中编译的设计提供基元仿真库。可以使用这些库在 Quartus II 软件支持的 EDA 仿真工具中对含有 Altera 专用组件的设计进行功能或时序仿真。此外，Altera 为 ModelSim-Altera 软件中的仿真提供预编译功能和时序仿真库。

Altera 为使用 Altera 宏功能模块以及参数化模块 (LPM) 功能标准库的设计提供功能仿真库。Altera 还为 ModelSim 软件中的仿真提供 `altera_mf` 和 `220model` 库的预编译版本。[表 2](#) 显示了与 EDA 仿真工具配合使用的功能仿真库。

表2. 功能仿真库

库名称	说明
220model.v	LPM 功能的仿真模型 (220 版)
220model.vhd	
220model_87.vhd	
220pack.vhd	220model.vhd 的 VHDL Component Declarations
altera_mf.v	
altera_mf.vhd	Altera 专用宏功能模块的仿真模型和 VHDL Component Declarations
altera_mf_87.vhd	
altera_mf_components.vhd	
sgate.v	Altera 专用宏功能模块的仿真模型和知识产权功能
sgate.vhd	
sgate_pack.vhd	
stratixgx_mf.v	含有仿真模型的库，用于含有 altgx2 宏功能模块的 Stratix GX 设计。
stratixgx_mf.vhd	
	对于 Verilog 设计，在编译该库之前，必须编译 220model.v 和 sgate.v 仿真模型库（注意顺序）。
	对于 VHDL 设计，在编译这些库之前，必须编译 220pack.vhd, 220model.vhd, sgate_pack.vhd 和 gate.vhd 仿真模型库（注意顺序）。

在 Quartus II 软件中，特定器件体系结构实体和 Altera 专用宏功能模块的信息位于布线后基元时序仿真库中。根据器件系列以及是否使用 Verilog 输出文件或 VHDL 输出文件，时序仿真库文件可能有所不同。对于 VHDL 设计，Altera 为具有 Altera 专用宏功能模块的设计提供 VHDL Component Declaration 文件。

有关信息	请参阅
时序仿真库	Quartus II Help 中的“Altera Postrouting Libraries”
功能仿真库	Quartus II Help 中的“Altera Functional Simulation Libraries”
使用 ModelSim 或 ModelSim-Altera 软件进行仿真	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷“Mentor Graphics ModelSim Support”
使用 VCS 软件进行仿真	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷“Synopsys VCS Support”
使用 NC-Sim 软件进行仿真	Altera 网站 <i>Quartus II Handbook</i> 第 3 卷“Cadence NC-Sim Support”

使用 Quartus II Simulator 进行仿真设计

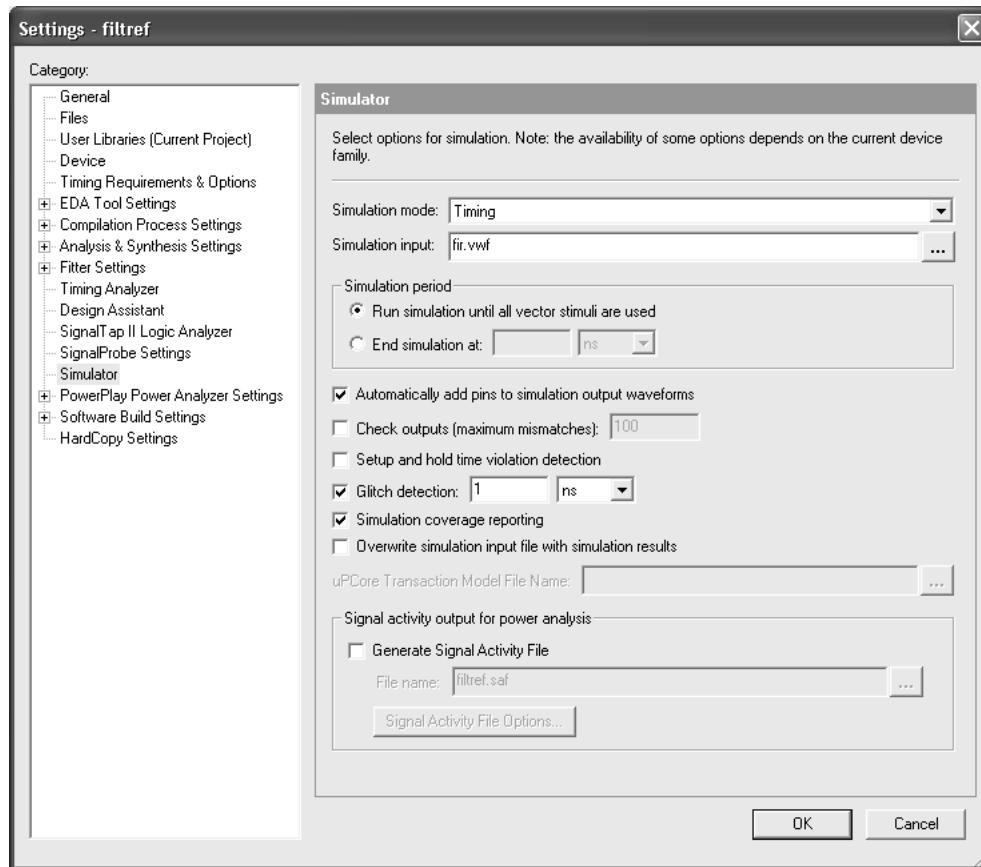


可以使用 Quartus II Simulator 在工程中仿真任何设计。根据所需的信息类型，可以进行功能仿真以测试设计的逻辑功能，也可以进行时序仿真，在目标器件中测试设计的逻辑功能和最坏情况下的时序，或者采用 Fast Timing 模型进行时序仿真，在最快的器件速率等级上仿真尽可能快的时序条件。

Quartus II 软件可以仿真整个设计，也可以仿真设计的任何部分。您可以指定工程中的任何设计实体为顶层设计实体，并仿真顶层实体及其所有附属设计实体。

通过使用 **Settings** 对话框 (Assignments 菜单) 或 **Simulator Tool** 窗口下的 **Simulator** 页面，您可以指定要执行的仿真类型，仿真所需的时间周期，向量激励源，以及其他仿真选项。图 3 所示为 **Simulator** 页面。

图 3. Settings 对话框的 Simulator 页面



开始仿真之前，必须通过为时序仿真编译设计，或者通过选择用于功能仿真的 **Generate Functional Simulation Netlist** 命令（Processing 菜单），来生成合适的仿真网表。此外，必须建立并指定一个向量源文件为仿真输入向量的源。Simulator 使用向量源文件所包含的输入向量，来仿真同一条件下编程器件将要产生的输出信号。

以下步骤说明在 Quartus II 软件中进行功能或时序仿真的基本流程。

1. 指定 Simulator 设置。
2. 如果正在执行功能仿真，则选择 **Generate Functional Simulation Netlist** 命令。如果正在执行时序仿真，则编译设计。
3. 建立并向量源文件。

4. 使用 Start > Start Simulation 命令 (Processing 菜单), Simulator Tool 窗口, 或 quartus_sim 可执行文件运行仿真。

Status 窗口显示仿真进度和处理时间。Report 窗口的 Summary Section 区域显示仿真结果。

使用 quartus_sim 可执行文件

还可以在命令提示符下或在脚本中使用 quartus_sim 可执行文件单独运行 Simulator。

quartus_drc 可执行文件建立可以使用任何文本编辑器查看的独立文本型报告文件。

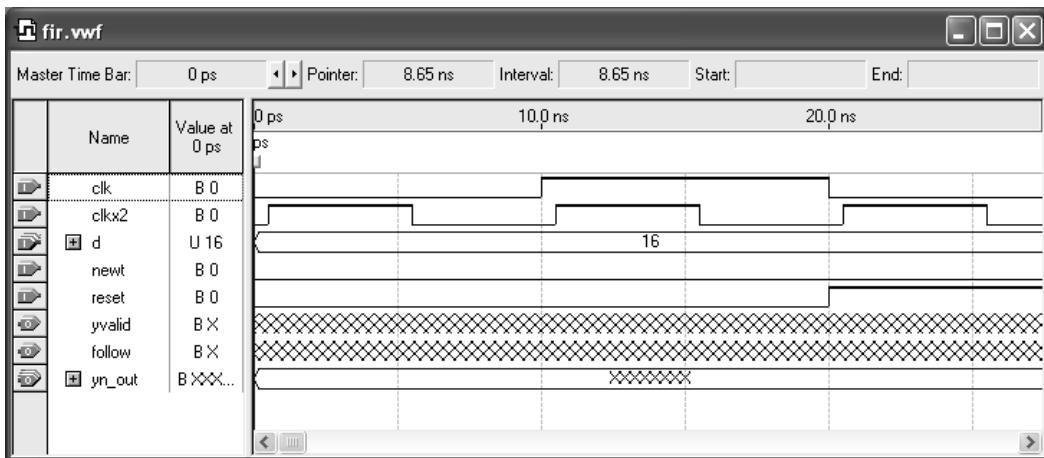
若要获取有关 quartus_sim 可执行文件的帮助信息, 请在命令提示符下键入以下命令之一:

```
quartus_sim -h ↵  
quartus_sim --help ↵  
quartus_sim --help=<topic name> ↵
```

建立波形文件

Quartus II Waveform Editor 可以建立和编辑用于波形或文本格式仿真的输入向量。使用 Waveform Editor, 可以将输入向量添加到波形文件中, 此文件描述设计中的逻辑行为。请参阅图 4。

图 4. Quartus II Waveform Editor

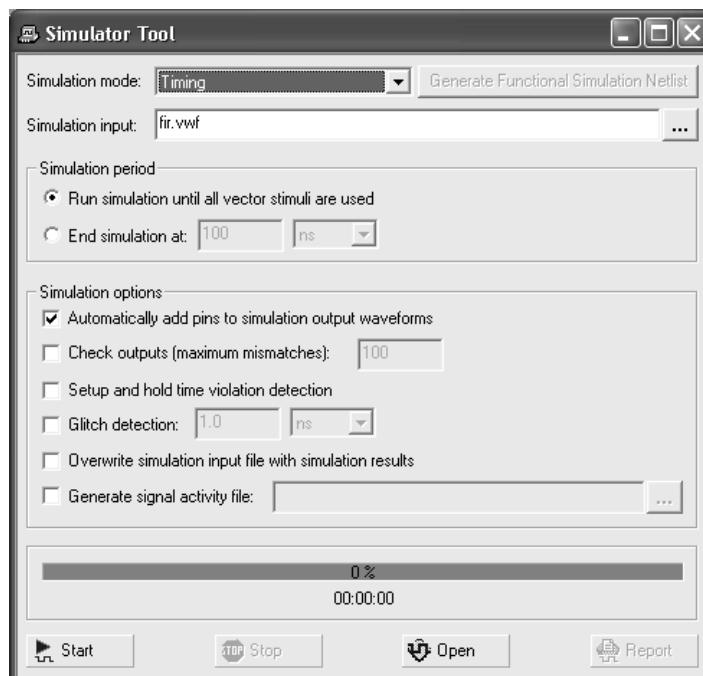


Quartus II 软件支持 Vector Waveform File (.vwf)、Vector Table Output File (.tbl)、Vector File (.vec) 和 Simulator Channel File (.scf) 格式的波形文件。不能在 Waveform Editor 中编辑 Simulator Channel 文件或者 Vector 文件，但可以将其保存为 Vector Waveform 文件。

使用 Simulator Tool

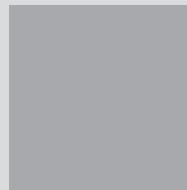
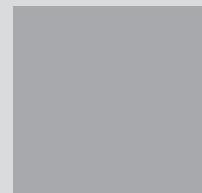
可以使用 **Simulator Tool** 命令 (Tools 菜单) 调整 Simulator 设置，以及启动或者停止 Simulator，为当前工程打开仿真波形。Simulator Tool 窗口的功能与 MAX+PLUS II Simulator 类似。要执行仿真，必须首先在用于功能仿真的 Simulator Tool 中，使用 **Generate Functional Simulation Netlist** 按钮，来生成仿真网表，如果您正在执行时序仿真，则首先要编译设计。图 5 所示为 Simulator Tool 窗口。

图 5. Simulator Tool 窗口



第八章

时序分析



第 8 章 内容：

简介	126
在 Quartus II 软件中进行时序分析	127
进行早期时序估算	133
查看时序分析结果	135
使用 EDA 工具进行时序分析	139

8

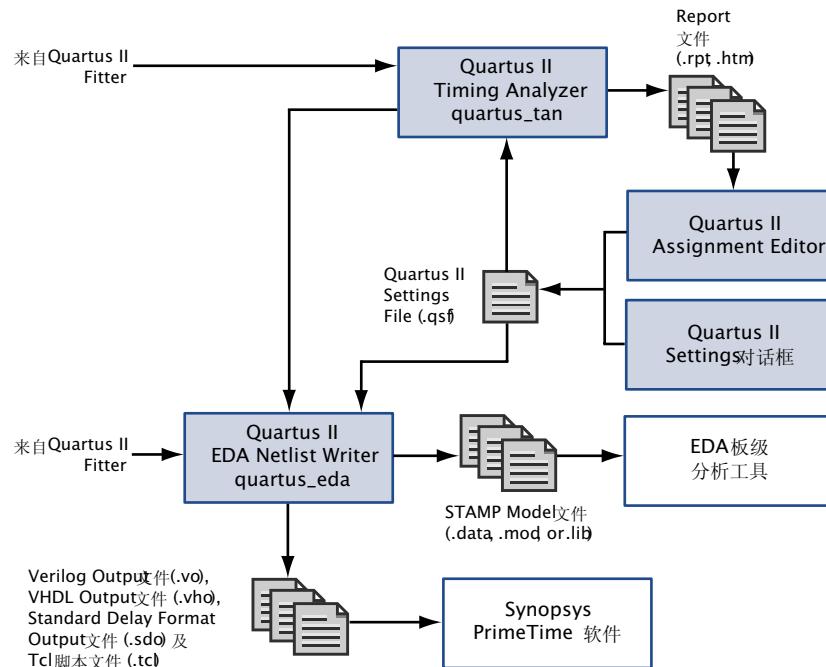
简介



Quartus® II Timing Analyzer 可用于分析设计中的所有逻辑，并有助于指导 Fitter 达到设计中的时序要求。默认情况下，Timing Analyzer 作为完整编译的一部分自动运行，分析、报告时序信息，例如，建立时间 (t_{SU})、保持时间 (t_H)、时钟至输出延时和最小时钟至输出延时 (t_{CO})、引脚至引脚延时和最小引脚至引脚延时 (t_{PD})、最大时钟频率 (f_{MAX})，以及设计的其它时序特性。

当提供时序约束或者默认设置有效时，Timing Analyzer 报告迟滞时间。可以使用 Timing Analyzer 生成的信息分析、调试和验证设计的时序性能。还可以使用快速时序模型，验证最佳情况（最快速率等级的最小延时）条件下的时序。图 1 所示为时序分析流程。

图 1. 时序分析流程



在Quartus II软件中进行时序分析

Timing Analyzer在完整编译期间自动对设计进行时序分析。以下准则描述了使用Quartus II Timing Analyzer可以完成的一些任务：

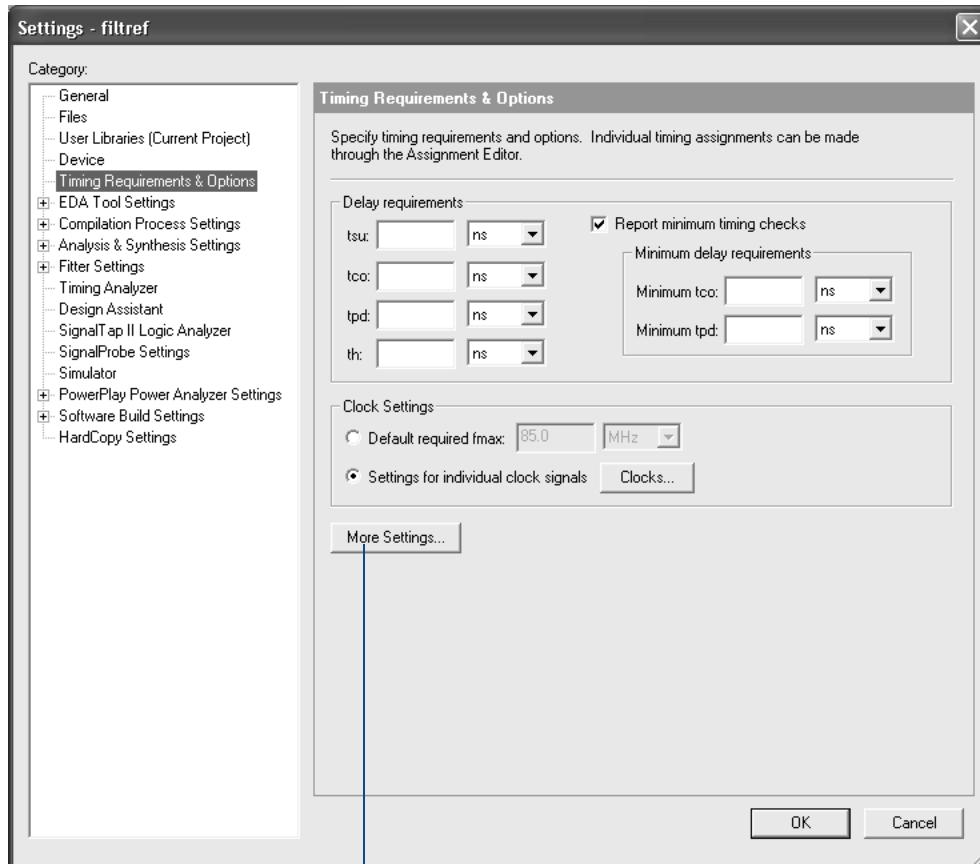
- ◇ 使用Timing Wizard (Assignments菜单)、Settings对话框 (Assignments菜单) 和 Assignment Editor，指定初始工程全局范围和个别的时序要求。
- ◇ 在完整编译期间进行时序分析或在初始编译之后单独进行时序分析。
- ◇ 在部分编译之后，适配完成之前，进行Early Timing Estimate。
- ◇ 使用Report窗口、Timing Closure平面布局图和list_path Tcl命令查看时序结果。

指定时序要求

时序要求允许为整个工程、特定的设计实体或个别实体、节点和引脚指定所需的速度性能。

可以使用Timing向导帮助建立初始工程全局范围时序设置。指定初始时序设置之后，可以再次使用Timing向导或使用Settings对话框的Timing Requirements & Options页面修改设置。图2是Timing Requirements & Options页面。

图2. Settings对话框的Timing Requirements & Options页面



单击 More Settings 按钮，显示 More Timing Settings 对话框，
它含有更多的选项。

您可以使用 Assignment Editor 对个别时序进行设置。指定工程全局范围时序分配以及个别时序分配之后，通过编译设计或在初始编译之后单独运行 Timing Analyzer 来运行时序分析。

如果未指定时序要求设置或选项，Quartus II Timing Analyzer 将使用默认设置运行分析。默认情况下，Timing Analyzer 计算并报告每个寄存器至寄存器延时的 f_{MAX} 、每个输入寄存器的 t_{SU} 和 t_H 、每个输出寄存器的 t_{CO} 、所有引脚至引脚路径间的 t_{PD} 、保持时间、最小 t_{CO} 以及当前设计实体的最小 t_{PD} 。提供约束条件或采用默认设置时，将报告迟滞时间。

您可以通过使用 **Input Maximum Delay**, **Input Minimum Delay**, **Output Maximum Delay**, 或者 **Output Minimum Delay** 分配来指定基于外部器件时序的延时，将这些路径作为时钟分析的一部分，从而指定 I/O 时序要求。也可以使用传统的 **t_{su} requirement**、**t_{co} requirement**，以及 **t_H requirement** 时序分配来指定 I/O 时序要求。这两种类型的 I/O 时序要求通过不同的方式最终都产生相似的结果。

使用 **Settings** 对话框或 **Timing** 向导，可以指定以下时序要求和其它选项：

- ◇ 工程的总体频率要求，以及各个时钟信号的设置
- ◇ 延时要求、最短延时要求和路径剪切选项
- ◇ 报告选项，包括源和目的寄存器的数量，以及排除路径的数量
- ◇ 时序驱动编译选项
- ◇ 建立(恢复)和保持(删除)选项检查具有异步清除、预置或装入信号的时序路径。

进行工程全局范围的时序设置

工程全局范围的时序设置包括最大频率、建立时间、保持时间、时钟至输出延时和引脚至引脚延时以及最小时序要求。还可以设置工程全局范围内的时钟设置和多时钟域、路径剪切选项。

表1. 工程全局范围的时序设置（第1部分，共2部分）

要求	说明
f _{MAX} (最大频率)	在不违反内部建立(t _{su})和保持(t _H)时间要求时，可以达到的最大时钟频率。
t _{su} (时钟建立时间)	触发寄存器的时钟信号在时钟引脚置位之前，经由数据输入或使能端输入进入寄存器的数据必须在输入引脚处出现的时间长度。
t _H (时钟保持时间)	触发寄存器的时钟信号在时钟引脚置位之后，经由数据输入或使能端输入而进入寄存器的数据必须在输入引脚处保持的时间长度。
t _{co} (时钟至输出延时)	时钟信号在触发寄存器的输入引脚上发生跳变之后，寄存器馈送信号输出引脚出现有效输出所需的时间。
t _{PD} (引脚至引脚延时)	输入引脚上的信号通过组合逻辑进行传输并出现在外部输出引脚上所需的时间。

表1. 工程全局范围的时序设置（第2部分，共2部分）

要求	说明
最小 t_{CO} (时钟至输出延时)	时钟信号在触发寄存器的输入引脚上发生跳变之后，寄存器馈送信号的输出引脚出现有效输出所需的最短时间。这个时间总是代表外部引脚至引脚延时。
最小 t_{PD} (时钟至输出延时)	指定可接受的最小引脚至引脚延时，即输入引脚信号通过组合逻辑传输并出现在外部输出引脚上所需的时间。

进行个别时序分配

可以使用 Assignment Editor 对个别实体、节点和引脚分别进行时序分配。个别时序分配超越工程范围要求（如果它们比工程范围要求更加严格的话）。Assignment Editor 支持点对点的时序分配，由通配符在建立分配时识别指定节点，支持使用 **timegroup** 分配以便为节点或节点组建立单独分配。

为引脚和节点输入的时序要求保存在当前层次结构顶层实体的 Quartus II Settings File (.qsf) 中。

可以在 Timing Analyzer 中进行以下类型的个别时序分配：

- ◇ **个别时钟设置**：允许通过定义时序要求和设计中所有时钟信号之间的关系，进行精确的多时钟时序分析。Timing Analyzer 支持单时钟和多时钟频率分析。
- ◇ **时钟不确定性分配**：时钟建立和保持检查时，指定应采用的预期时钟建立或保持不确定性（抖动）。计算保持检查时，在计算建立检查、向数据所需时间加入指定保持不确定性时，Timing Analyzer 在数据所需时间中提取指定建立的不确定性。
- ◇ **时钟延时分配**：指定提前或迟后时钟延时作为等待时间。等待时间会影响时钟斜移，斜移与偏移不同，它影响建立关系。时钟延时代表通过最短路径或最长路径的可能（理想情况下）时钟外部延时。对于建立分析，Timing Analyzer 对每个源采用迟后延迟值，对每个目的寄存器采用提前延迟值，对于保持分析，Timing Analyzer 对每个源使用提前延迟值，对每个目的寄存器采用迟后延迟值。
- ◇ **多周期路径**：需要一个以上时钟周期才能稳定下来的寄存器之间的路径。可以设置多周期路径，指示 Timing Analyzer 放宽度量，并避免不正确地建立或保持时间。

- ◇ **剪切路径：**默认情况下，如果没有设置时序要求或只使用默认的 f_{MAX} 时钟设置，Quartus II软件将切断不相关时钟域之间的路径。如果设置了各个时钟分配，但没有定义时钟分配之间的关系，Quartus II也将切断不相关时钟域之间的路径。还可以定义设计中特定路径的剪切路径。
- ◇ **最大延时要求：**设计中特定节点 t_{SU} 、 t_H 、 t_{PD} 和 t_{CO} 的输入输出最大延时，以及最大时序要求。可以对特定节点或节点组进行这些分配，以超越工程全局范围最大时序要求。
- ◇ **最小延时要求：**特定节点或组 t_H 、 t_{PD} 和 t_{CO} 的输入输出最小延时，以及最小时序要求。可以对特定节点或节点组进行这些分配，以超越工程全局范围最小时序要求。
- ◇ **最大斜移要求：**特定节点或节点组最大时钟和数据到达斜移的时序要求。
- ◇ **时间组分配：**可以在**Time Groups**对话框(Assignments菜单)中定义的高级时序分配；也可以采用Quartus II Tcl Console的Tcl API或者支持Tcl的Quartus II可执行文件之一进行定义。所定义的时间组成员可以是常用节点名称、通配符，以及其他时间组名称。也可以从时间组中排除特定节点、通配符以及其他时间组的名称。然后，这些组可以用在多数时序分配的**From**或**To**区。

进行时序分析

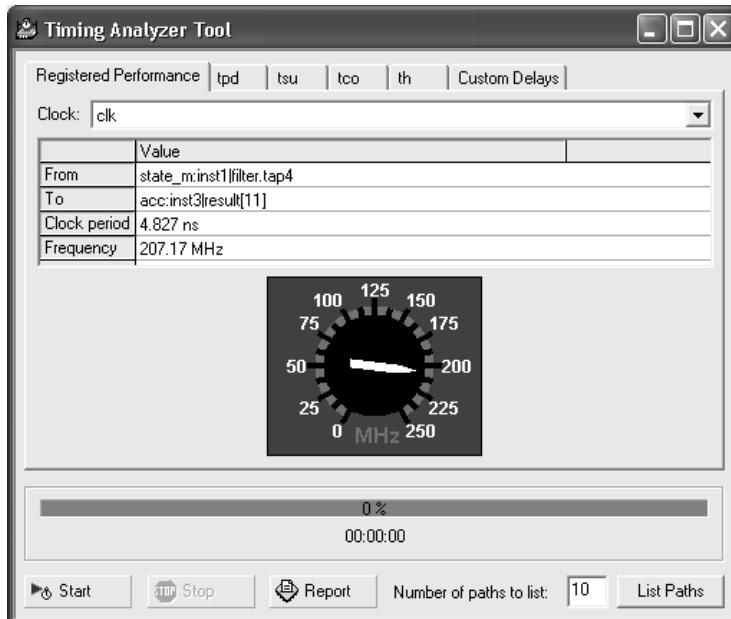
指定时序设置和约束之后，就可以通过完整编译运行Timing Analyzer。

完成编译之后，可以使用**Start > Start Timing Analyzer**命令(Processing菜单)重新单独运行时序分析，通过选择**Start > Start Timing Analyzer (Fast Timing Model)**(Processing菜单)，或者使用**Timing Analyzer Tool**命令(Tools菜单)运行最快时序模型的时序分析。

使用**Start > Start Early Timing Estimate**命令(Processing菜单)，可以在适配完成之前，生成早期时序估算的数据。请参阅第133页“[进行早期时序估算](#)”，了解更详细的信息。

Timing Analyzer Tool窗口提供一个可选界面，用来控制Timing Analyzer。此界面类似于MAX+PLUS II软件中的Timing Analyzer界面。您可以使用Timing Analyzer Tool窗口启动、停止Timing Analyzer，快速查看时序分析结果摘要，或者在Compilation Report中查看详细的时序分析结果。可单击**List Paths**来显示选定路径的传输延时。请参阅图3。

图3. Timing Analyzer Tool



使用quartus_tan可执行文件

还可以在命令提示符下或在脚本中使用**quartus_tan**可执行文件单独运行Timing Analyzer。在运行Timing Analyzer之前，必须先运行Quartus II Fitter可执行文件**quartus_fit**。

quartus_tan可执行文件建立可以使用任何文本编辑器查看的独立文本型报告文件。

还可以通过在命令提示符下键入以下命令，启动**quartus_tan**Tcl脚本shell，运行与时序相关的Tcl命令：

```
quartus_tan -s ↵
```

若要获取有关**quartus_tan**可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_tan --h ↵
quartus_tan --help ↵
quartus_tan --help=<topic name> ↵
```

有关信息	请参阅
在 Quartus II 软件中进行特定的时序设置和时序分析	Quartus II Help 中的“Overview:Using the Timing Analyzer”
	Altera 网站上 <i>Quartus II Handbook</i> 第 3 卷“Quartus II Timing Analysis”。
	Quartus II Tutorial 中的 Timing Analysis 部分
使用 Timing Analyzer Tool 窗口	Quartus II Tutorial 中的 MAX+PLUS II Conversion 部分

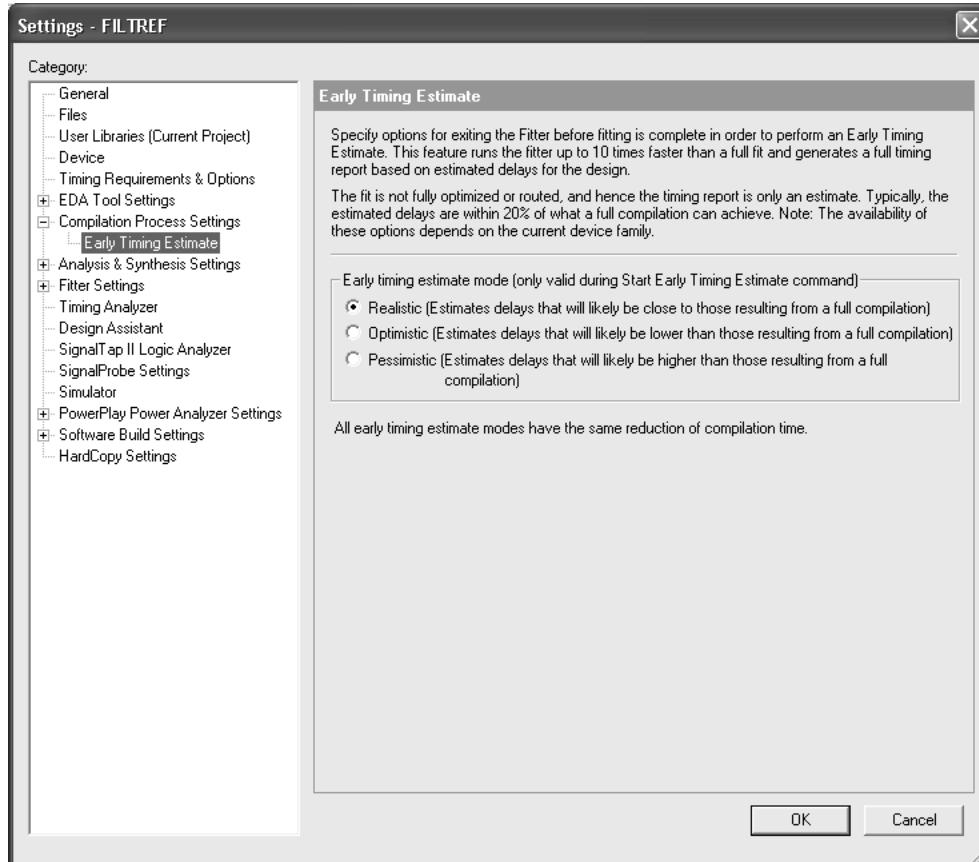
进行早期时序估算

如果希望为早期时序估算生成数据，可以使用 **Start > Start Early Timing Estimate** 命令 (Processing 菜单) 在适配完成之前，指示 Fitter 停止。可以在 **Settings** 对话框 (Assignments 菜单) **Compilation Process Settings** 的 **Early Timing Estimate** 页面为早期时序估算指定选项。可以为 **Early timing estimate mode** 从以下选项中进行选择：

- ◇ Realistic
- ◇ Optimistic
- ◇ Pessimistic

图 4 所示为 Early Timing Estimate 页面。

图 4. *Settings* 对话框的 *Early Timing Estimate* 页面



运行 **Start> Start Early Timing Estimate** 命令时，Compiler 执行包括 Analysis & Synthesis 在内的部分编译，但是在 Fitter 完成之前会停止。然后可以在 Report 窗口的 Timing Analyzer 部分查看早期时序估算，与查看普通的时序分析结果一样；但是要注意，所有的早期时序估算都应看作是初步的结果。关于在 Report 窗口查看时序结果的详细信息，请参考下一节“[查看时序分析结果](#)”。

查看时序分析结果

运行时序分析之后，可以在 Compilation Report 的 Timing Analyzer 文件夹中查看时序分析结果。然后，列出时序路径以验证电路性能，确定关键速度路径以及限制设计性能的路径，并重新进行时序分配。此外，还可以使用 list_path Tcl 命令查找并查看设计中任何延时路径的信息。

也可以使用 Timing Closure 平面布局图 (Project 菜单) 来查看设计中关键路径上的信息，并查看布线拥塞。有关使用 Timing Closure 平面布局图来查看关键路径和布线拥塞的详细信息，请参阅 第9章“时序逼近”，第144页“使用 Timing Closure Floorplan”。

熟悉 MAX+PLUS® II 时序报告的用户可在 Compilation Report 的 Timing Analyzer 部分和 Timing Analyzer Tool 窗口的 Custom Delays 标签中找到时序信息，例如来自 MAX+PLUS II Delay Matrix 的延时信息。

使用报告窗口



Report 窗口的 Timing Analysis 部分列出时钟建立和保持的时序信息； t_{SU} 、 t_H 、 t_{PD} 和 t_{CO} ；最小 t_{PD} 和 t_{CO} 。Timing Analysis Report 窗口部分还列出了最大时钟到达斜移、最大数据到达斜移、最小脉冲宽度要求；在时序分析期间忽略的任何时序约束；以及 Timing Analyzer 生成的任何消息。

Report Window 窗口报告以下类型的时序分析信息：

- ◇ 时序要求的设置
- ◇ 迟滞和最小迟滞
- ◇ 源和目标时钟名称
- ◇ 源和目标节点名称
- ◇ 所需的和实际的点到点时间
- ◇ 实际 f_{MAX}

图 5. Compilation Report 窗口中的 Timing Analysis Results

The screenshot shows the Quartus II software interface with the 'Compilation Report - Timing Analyzer Summary' window open. The left pane displays a tree view of timing analysis results, including Flow Summary, Flow Settings, Flow Elapsed Time, Flow Log, Analysis & Synthesis, Filter, Assembler, and a expanded 'Timing Analyzer' section containing 'Summary', 'Settings', 'Clock Settings Summary', 'Clock Setup: 'clk'', 'Clock Setup: 'clkx2'', 'Clock Hold: 'clk'', 'Clock Hold: 'clkx2'', 'tsu', 'tco', 'th', and 'Ignored Timing Assignments'. The right pane is titled 'Timing Analyzer Summary' and contains a table with the following data:

Type	Slack	Required Time	Actual Time
1 Worst-case tsu	N/A	None	4.187 ns
2 Worst-case tco	N/A	None	6.098 ns
3 Worst-case th	N/A	None	-2.887 ns
4 Clock Setup: 'clk'	1.490 ns	100.00 MHz (period = 10.000 ns)	117.51 MHz (period
5 Clock Setup: 'clkx2'	3.611 ns	200.00 MHz (period = 5.000 ns)	N/A
6 Clock Hold: 'clk'	0.664 ns	100.00 MHz (period = 10.000 ns)	N/A
7 Clock Hold: 'clkx2'	5.168 ns	200.00 MHz (period = 5.000 ns)	N/A
8 Total number of failed paths			

如果希望同时使用最快（最佳情况）时序模型和最慢（最差情况）时序模型来生成时序结果，则需要指定 Timing Analyzer 应同时使用最快和最慢模型分析和报告时序，并在 Timing Analyzer Section 的不同部分显示结果。如果希望查看未约束时序路径列表，可以指定您希望将其包含在 Timing Constraints Check 报告中。

进行分配与查看延时路径

可以从 Report Window 的 Timing Analyzer 部分直接进入 Locate in Assignment Editor、List Paths 和 Locate in Timing Closure Floorplan 命令，从而可以进行个别时序分配，查看延时路径信息。此外，还可以使用 list_path Tcl 命令列出延时路径信息。

可以使用 Locate in Assignment Editor 命令打开 Assignment Editor，在 Timing Analyzer 报告中对任何路径进行个别时序分配。此功能还可以用来方便地在路径上进行点到点分配。

以下步骤描述在 Assignment Editor 中进行个别时序分配的基本流程：

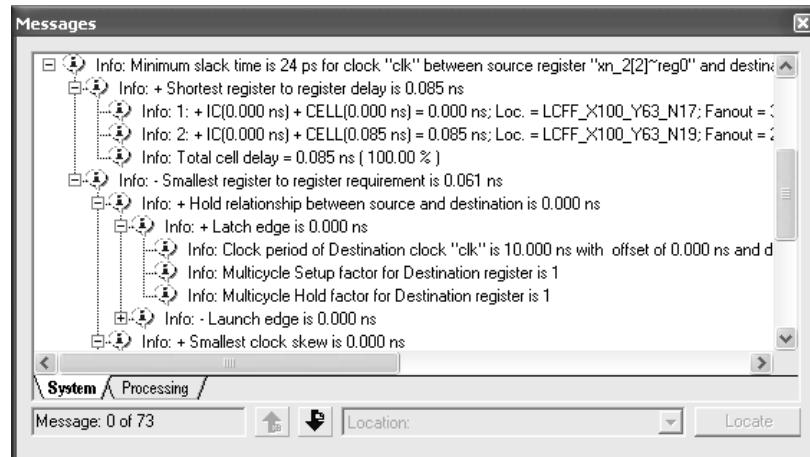
1. 在 Category 栏中，单击 Timing，指明要做的分配类别。
2. 在电子表格中单击 To 单元格并使用 Node Finder 查找节点，或输入希望进行分配的目标节点名称、通配符以及 time group。

3. 在电子表格中单击 **From** 单元格并使用 Node Finder 查找节点，或输入希望进行分配的源节点名称、通配符以及 **time group**。
4. 在电子表格中，双击 **Assignment Name** 单元格并选定要做的时序分配。
5. 可以双击 **Value** 单元格并键入或选择相应的分配值。

还可以使用 **Locate in Timing Closure Floorplan** 命令 (Project 菜单) 在 Timing Closure 平面布局图中查找路径，利用 Timing Closure 平面布局图的功能对特定路径做分配。有关使用 Timing Closure 平面布局图的详细信息，请参阅第 9 章“时序逼近”，第 144 页“使用 Timing Closure Floorplan”。

可以使用 **List Paths** 命令 (右键弹出菜单)，在 Messages 窗口的 Timing Analyzer 报告面板中显示任何路径的中间延时。**List Paths** 命令允许用户查找引脚至引脚、寄存器至寄存器和时钟至输出引脚延时路径，并显示出现在 Report Window 中，设计的任何延时路径信息。请参阅图 6。

图 6. List Paths 命令的输出



list_path Tcl 命令可以在 **quartus_tan** 模块和 Quartus II Tcl Console 中使用，用于指定任何点到点路径，查看延时信息。可以指定要报告的路径数量，路径类型（包括最小时序路径），使用通配符标识源和目标节点。此选项报告信息的方式与 **List Path** 命令相同。请参阅图 7。

图 7. *list_path* 命令的实例输出

```
Path Number: 1
tco from clock clock to destination pin gt1 through register auto_max:auto1street_map[0] is 8.869 ns
-----
+ Longest clock path from clock clock to source register is 2.799 ns^M
  1; + IC(0.000 ns) + CELL(0.619 ns) = 0.619 ns; Loc. = Pin_M2; CLK Node = 'clock'
  2; + IC(1.638 ns) + CELL(0.542 ns) = 2.799 ns; Loc. = LC_X31_Y1_N9; REG Node = 'auto_max:auto1street_map[0]'
  Total cell delay = 1.161 ns
  Total interconnect delay = 1.638 ns
+ Micro clock to output delay of source is 0.156 ns
+ Longest register to pin delay is 5.914 ns
  1; + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = LC_X31_Y1_N9; REG Node = 'auto_max:auto1street_map[0]^'
  2; + IC(0.716 ns) + CELL(0.075 ns) = 0.791 ns; Loc. = LC_X30_Y1_N9; COMB Node = 'rtl^"261'
  3; + IC(0.518 ns) + CELL(0.366 ns) = 1.675 ns; Loc. = LC_X30_Y1_N8; COMB Node = 'rtl^"17'
  4; + IC(1.350 ns) + CELL(2.889 ns) = 5.914 ns; Loc. = Pin_AA13; PIN Node = 'gt1'
  Total cell delay = 3.330 ns
  Total interconnect delay = 2.584 ns
```

使用 Technology Map Viewer



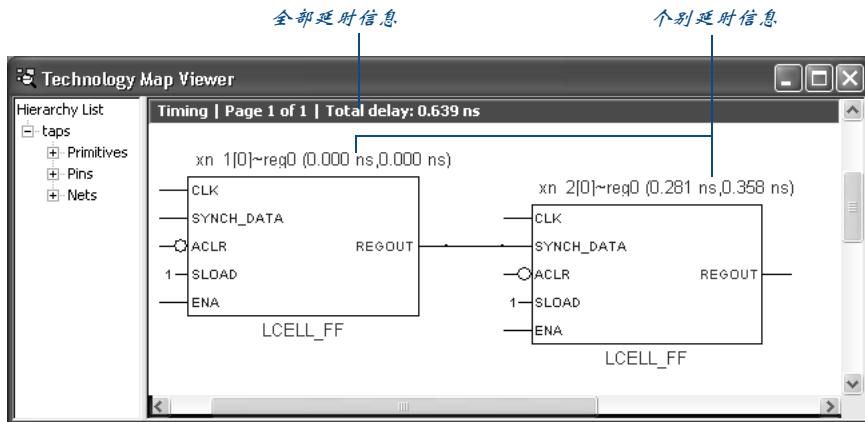
Quartus II Technology Map Viewer 提供设计的底级或者基元级的专用技术原理表征。Technology Map Viewer 包括原理视图，以及层次列表，它列出了实例、基本单元、引脚和整个设计网表的网络图。

进行 Timing Analysis 或者包含 Timing Analysis 的完整编译之后，可以使用 Technology Map Viewer 来查看组成时序路径的节点，包括全部延时和各个节点延时的信息。请参见图 8。

使用以下方法，在进行时序分析之后显示 Technology Map Viewer：

- ◇ 使用 Technology Map Viewer 命令 (Tools 菜单)
- ◇ 在 Report Window 中的 Timing Analyzer 部分，使用 Locate in Technology Map Viewer 命令 (右键弹出菜单)
- ◇ 使用 Messages 窗口 (在使用 List Paths 命令之后) 中的 Location 列表

图 8. Technology Map Viewer 窗口——延时信息



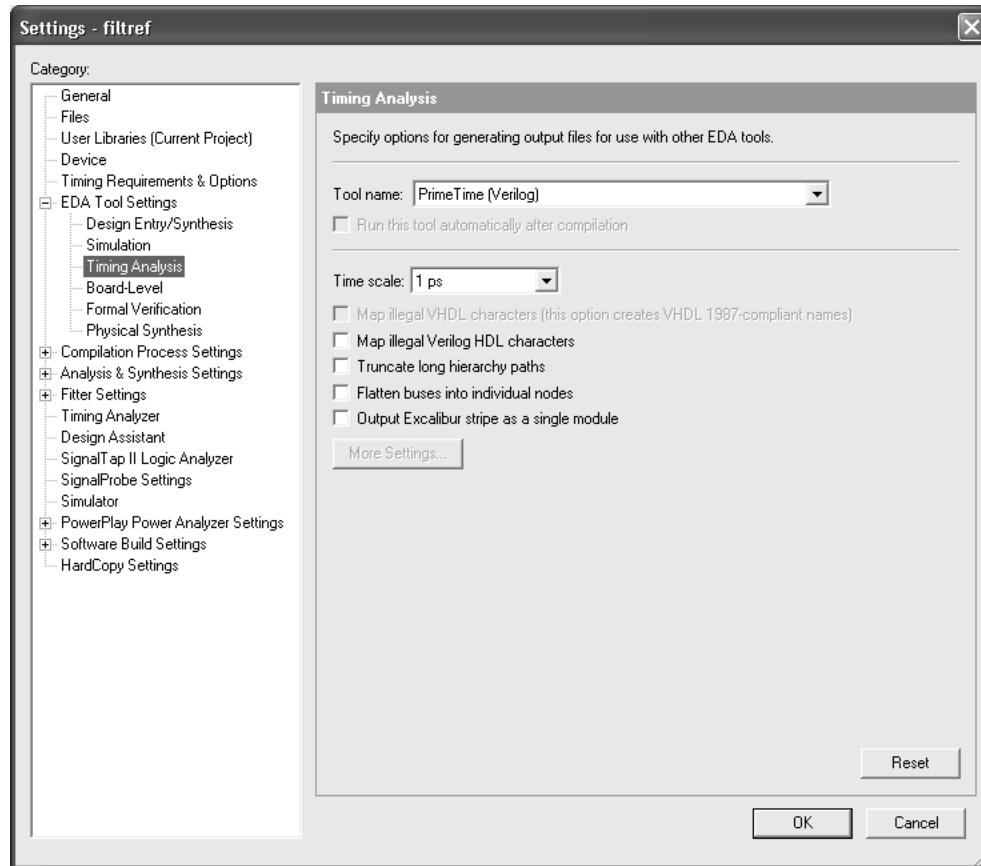
使用 EDA 工具进行时序分析



Quartus II 软件支持在 UNIX 工作站上使用 Synopsys PrimeTime 软件进行时序分析和最小时序分析，并支持使用 Mentor Graphics® Tau 板级验证工具进行板级时序分析。

通过在 **Settings** 对话框 (**Assignments** 菜单) 中的 **EDA Tool Settings** 下的 **Timing Analysis** 和 **Board-Level** 页面中或建立工程时使用 **New Project Wizard** (**File** 菜单) 指定适当的时序分析工具来生成在 EDA 时序分析工具中执行时序分析的必要输出文件，然后执行完整编译。图 9 所示为 **EDA Tool Settings** 下的 **Timing Analysis** 页面。

图 9. Settings 对话框的 EDA Tool Timing Analysis 页面



还可以在初始编译之后，使用 Start > Start EDA Netlist Writer 命令 (Processing 菜单) 生成文件。如果您正在使用 NativeLink® 功能，也可以使用 Run EDA Analysis Tool 命令 (Tools 菜单) 在初始编译后进行时序分析。

使用 quartus_eda 可执行文件

还可以运行 EDA Netlist Writer，在命令提示符下或脚本中使用 **quartus_eda** 可执行文件分别生成所需的输出文件。在运行 EDA Netlist Writer 之前，必须先运行 Quartus II Fitter 可执行文件 **quartus_fit**。

quartus_eda 可执行文件建立可以使用任何文本编辑器查看的独立文本型报告文件。

若要获取有关 **quartus_eda** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_eda -h ↵  
quartus_eda -help ↵  
quartus_eda --help=<topic name> ↵
```

使用 PrimeTime 软件

Quartus II 软件生成 Verilog 或 VHDL 输出文件、包含时序延时信息的 Standard Delay Format Output 文件 (.sdo) 以及设置 PrimeTime 环境的 Tcl 脚本文件。如果正在进行最小时序分析，Quartus II 软件将使用由 Timing Analyzer 在该设计的 SDF 输出文件中生成的最小延时信息。

使用 NativeLink 功能，可以指定 Quartus II 软件从命令行或 GUI 模式启动 PrimeTime 软件。还可以指定 Synopsys Design Constraints 文件 (.sdc)，此文件包含供 PrimeTime 软件使用的时序分配。

以下步骤描述在 Quartus II 软件中编译之后，手动使用 PrimeTime 软件对设计进行时序分析的基本流程：

1. 通过 **Settings** 对话框 (Assignments 菜单) 或在工程设置期间使用 **New Project Wizard** (File 菜单) 进行 EDA 工具设置。
2. 在 Quartus II 软件中编译设计，生成输出网表文件。Quartus II 软件将该文件放置在特定工具目录中。
3. 查找 Quartus II 生成的 Tcl 脚本文件 (.tcl)，用以设置 PrimeTime 环境。
4. 在 PrimeTime 软件中进行时序分析。

使用 Tau 软件

Quartus II 软件生成 STAMP 模型文件，此文件可以被导入到 Tau 软件中，进行板级时序验证。

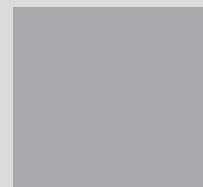
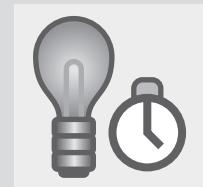
以下步骤描述生成 STAMP 模型文件的基本流程：

1. 通过 **Settings** 对话框 (Assignments 菜单) 或在工程设置期间使用 **New Project Wizard** (File 菜单) 指定 EDA 工具设置。
2. 在 Quartus II 软件中编译设计，生成 STAMP 模型文件。Quartus II 软件将该文件放置在特定工具目录中。
3. 在 Tau 软件中使用 STAMP 模型文件进行板级时序验证。

有关信息	请参阅
Synopsys PrimeTime 软件与 Quartus II 软件结合使用	Altera 网站上 <i>Quartus II Handbook</i> 第 3 卷“Synopsys PrimeTime Support”
Mentor Graphics Tau 软件与 Quartus II 软件结合使用	Quartus II Help 中的 “Overview:Using the Tau Software with the Quartus II Software”

第九章

时序逼近



第 9 章 内容:	
简介	144
使用 Timing Closure Floorplan	144
使用 Timing Optimization Advisor	147
使用网表优化实现时序逼近	148
使用 LogicLock 区域达到时序逼近	150
使用 Design Space Explorer 达到时序逼近	153

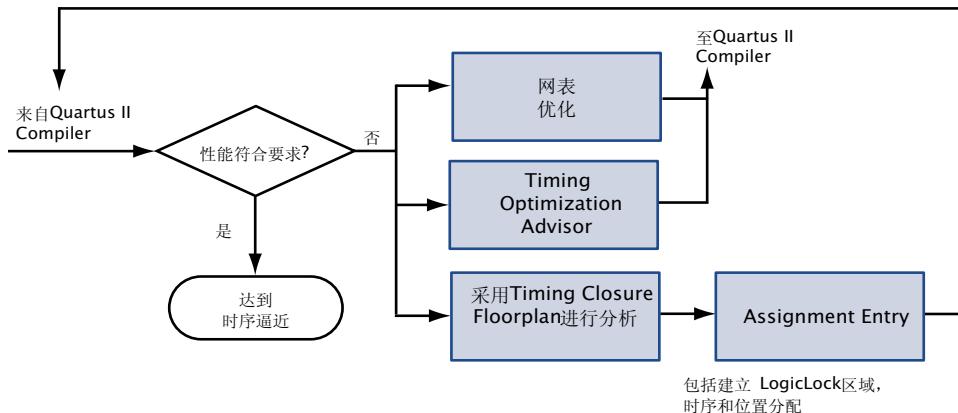
9

简介

Quartus® II 软件提供完全集成的时序逼近流程，该流程通过控制设计的综合和布局布线来达到时序目标。使用时序逼近流程可以对复杂的设计进行更快的时序逼近，减少优化迭代次数并自动平衡多个设计约束。

时序逼近流程可以执行初始编译、查看设计结果，进一步高效地优化设计。在综合之后以及在布局布线期间，可以使用 Timing Closure 平面布局图分析设计并进行分配，使用 Timing Optimization Advisor 查看优化设计时序的建议，使用设计的网表优化，还可以使用 LogicLock™ 区域分配和 Design Space Explorer 进一步优化设计。图 1 所示为时序逼近流程。

图 1. 时序逼近流程



使用 Timing Closure Floorplan



可以使用 Timing Closure 平面布局图查看 Fitter 生成的逻辑布局，查看用户分配和 LogicLock 区域分配，以及设计的布线信息。可以使用这些信息在设计中识别关键路径，进行时序分配、位置分配和 LogicLock 区域分配，达到时序逼近。

可以使用 View 菜单中提供的选项自定义 Timing Closure 平面布局图显示信息的方式。可以按照封装引脚及其功能显示器件；可以按内部 MegaLAB™ 结构、LAB 和单元显示器件；也可以按芯片的区域、按所选信号的名称和位置显示器件。

可以使用 **Field View** 命令 (View 菜单) 在 Timing Closure 平面布局图的高级轮廓视图中显示器件资源的主要分类。在 Field 视图中用彩色区域表示分配，这些彩色区域显示用户已分配数量、已布置的 Fitter 以及器件中每个结构未分配的逻辑。可以使用 Field 视图中的信息进行分配，达到设计的脉序逼近。

查看分配与布线

Timing Closure 平面布局图可同时显示用户分配和 Fitter 位置分配。用户分配是用户在设计中所做的所有位置与 LogicLock 区域分配。Fitter 分配是 Quartus II 软件在上次编译之后布置所有节点的位置。可以使用 **Assignments** 命令 (View 菜单) 显示用户分配和 Fitter 分配。

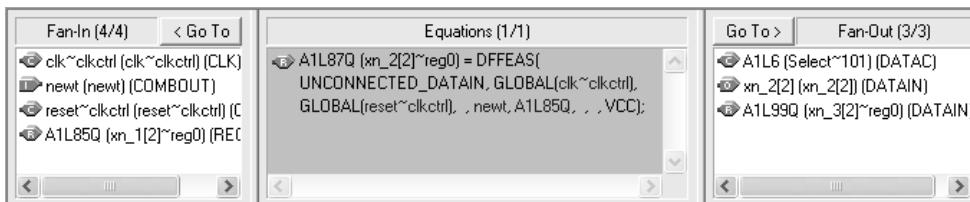
Timing Closure 平面布局图显示器件资源以及所有设计逻辑的相应布线信息。使用 **Routing** 命令 (View 菜单)，可以选择器件资源，查看以下布线信息类型：

- ◇ **节点之间的路径**：显示所选逻辑单元之间、I/O 单元、嵌入式单元和相互连接引脚的路径。
- ◇ **节点扇入和扇出**：显示所选嵌入式单元、逻辑单元、I/O 单元和引脚的节点扇入和扇出布线信息。
- ◇ **布线延时**：显示引至或源自特定逻辑单元、I/O 单元、嵌入式单元、引脚；所选节点之间；或一个或多个关键路径上的布线延时。
- ◇ **连接计数**：显示或隐藏连接至所选对象、从所选对象或所选对象之间连接的数量。
- ◇ **物理脉序估计**：显示从一个物理资源至另一个物理资源的近似延时。选择一个物理资源后，则用可能目标资源的阴影区表示延时（资源的阴影越深，延时越长），可以将鼠标放置在另一个物理资源之上以数字形式显示到达目标资源的延时。
- ◇ **布线拥塞**：显示设计中用图形表示的布线拥塞。阴影越深，布线资源利用率越高。可以选择布线资源，然后指定该资源的拥塞阈值（在器件中以红色区域显示）。
- ◇ **关键路径**：显示设计中的关键路径，包括路径边缘和布线延时。默认关键路径视图显示寄存器至寄存器路径。还可以查看源节点和目标节点之间最差路径的所有组合节点。可以通过指定时钟域、源节点名称和目标节点名称以及要显示的关键路径数和迟滞来指定滤除显示关键路径的标准。

还可以查看设计中 LogicLock 区域的布线信息，包括连接和区域内延时。LogicLock 区域连接显示分配给设计中 LogicLock 区域的实体之间的连接，区域内延时显示 LogicLock 区域（包括其子区域）中源和目标路径之间的最大时间延时。

Equations 窗口显示引脚、I/O 单元、逻辑单元和嵌入式单元分配的布线和等式信息。打开 Equations (View 菜单) 后，在 Timing Closure 平面布局图窗口的底部显示 Equations 窗口。请参见 图 2。

图 2. Equations 窗口



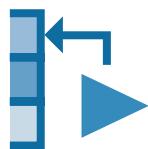
通过在平面布局图中选择一个或多个逻辑单元、嵌入式单元以及引脚分配，可以在 Equations 列表中显示其等式、扇入和扇出，并可以展开或收敛这些条目。Fan-In 列表显示馈送选定逻辑单元、嵌入式单元以及引脚分配的所有节点。Fan-Out 列表显示由选定逻辑单元、嵌入式单元以及引脚分配馈送的所有节点。

进行分配

为便于达到时序逼近，Timing Closure 平面布局图允许直接从布局图进行位置和时序分配。可以在 Timing Closure 平面布局图的自定义区域和 LogicLock 区域中建立和分配节点或实体，编辑对引脚、逻辑单元、行、列、区域、MegaLAB 结构和 LAB 的现有分配。还可以为 Assignment Editor 查找节点或节点组，并在那里进行分配。

可以使用以下方法在 Timing Closure 平面布局图中编辑分配：

- ◇ 剪切、复制和粘贴节点和引脚分配。
- ◇ 启动 Assignment Editor 进行分配。
- ◇ 使用 Node Finder 协助分配工作。
- ◇ 在 LogicLock 区域中建立和分配逻辑。
- ◇ 用鼠标从 Project Navigator 的 Hierarchy 选项标签、LogicLock 区域和 Timing Closure 平面布局图中拖出节点和实体，放到布局图的其它区域中。



在做分配之前，可以使用 **Back-Annotate Assignments** 命令 (Assignments 菜单) 对引脚、逻辑单元、行、列、区域、LAB、MegaLAB 结构和 LogicLock 区域进行反标注分配，保留当前编译的资源分配。有关使用 **Back-Annotate Assignments** 命令的详细信息，请参阅第 5 章“布局布线”，第 98 页“通过反标注保留分配”。

有关信息	请参阅
在 Timing Closure 平面布局图中查看分配和布线	Altera 网站上 <i>Quartus II Handbook</i> 第 2 卷“Timing Closure Floorplan”。
采用 Floorplan Editor 进行分配	Quartus II Help 中的“Overview:Viewing Routing Information”
	Quartus II Help 中的“Overview:Working with Assignments in the Floorplan Editor”
	Quartus II Tutorial 中的 LogicLock 模块

使用 Timing Optimization Advisor



Timing Optimization Advisor 在以下区域中，针对设计时序优化提出建议：

- ◇ 最大频率 (f_{MAX})
- ◇ 建立时间 (t_{SU})
- ◇ 时钟至输出 (t_{CO})
- ◇ 传播延时 (t_{PD})

打开一个工程后，可以通过选择 **Timing Optimization Advisor** (Tools 菜单) 来查看 Timing Optimization Advisor。如果还没有编辑工程，Timing Optimization Advisor 只为时序优化提供一般建议。如果工程已经编译完毕，Timing Optimization Advisor 便能够根据工程信息和当前设置，提供特定的时序建议。图 3 所示为 Timing Optimization Advisor。

图 3. Timing Optimization Advisor 初始页面



Timing Optimization Advisor 的特性与 Resource Optimization Advisor 非常相似；更详细的信息，请参阅 第 5 章 “布局布线”，第 92 页 “使用 Resource Optimization Advisor”。

使用网表优化实现时序逼近



Quartus II 软件包括网表优化选项，用于在综合以及布局布线期间进一步优化设计。网表优化具有按键式特性，它通过修改网表提高性能来改进 f_{MAX} 结果。不管使用何种综合工具，均可应用这些选项。根据设计条件，有些选项可能会比其它选项作用更大一些。

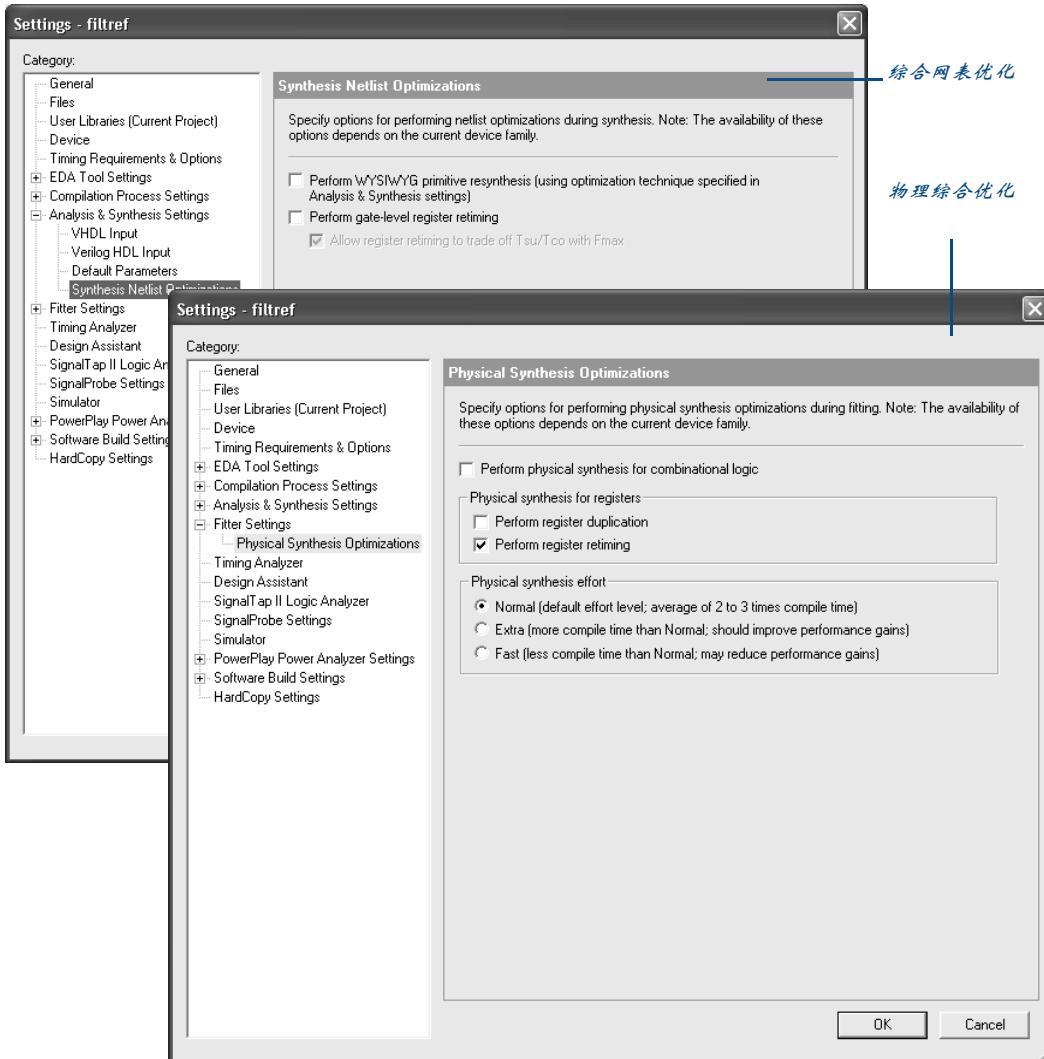
可在 Settings 对话框 (Assignments 菜单) 的 Synthesis Netlist Optimizations 和 Physical Synthesis Optimizations 页面中指定综合和物理综合网表优化选项。请参阅 第 149 页的图 4。

综合网表优化包括以下选项：

- ◇ Perform WYSIWYG primitive resynthesis(执行 WYSIWYG 基本单元再综合): 指示 Quartus II 软件在综合期间取消映射 WYSIWYG 基本单元。打开此选项后，Quartus II 软件将取消基元网表中逻辑单元对逻辑门的映射，并将逻辑门重新映射回 Altera® LCELL 基本单元。该选项允许 Quartus II 软件在重新映射过程期间使用特定器件体系结构的不同技术，使用在 Settings 对话框 Analysis & Synthesis Settings 页面中指定的优化技术 (Area, Balanced 或 Speed)。
- ◇ Perform gate-level register retiming(执行门级寄存器再定时): 允许在组合逻辑间移动寄存器以平衡时序，但不更改当前设计的功能。此选项仅在组合逻辑门间移动寄存器，不会在用户例化的逻辑单元、存储器块、

DSP 块、承载或级联链间移动寄存器，并能够将寄存器从组合逻辑块的输入移动到输出，从而可以将寄存器组合在一起。它还可以从组合逻辑块输出端寄存器开始，在组合逻辑块输入端建立多个寄存器。

图 4. 网表优化



- ◇ Allow register retiming to trade off T_{su}/T_{co} with F_{max} (允许寄存器再定时，平衡 F_{max} 和 T_{su}/T_{co}): 指示 Quartus II 软件在寄存器再定时，平衡 t_{CO} 、 t_{SU} 与 f_{MAX} 时，在与 I/O 引脚相关的寄存器间移动逻辑。此选项开启之后，寄存器再定时可能对馈送 I/O 引脚并由 I/O 引脚馈送的寄存器产生影响。如果未开启此选项，寄存器再定时不会接触与 I/O 引脚相连的任何寄存器。

物理综合网表优化和适配包括以下选项：

- ◇ Perform physical synthesis for combinational logic(进行组合逻辑的物理综合): 指示 Quartus II 软件在适配期间对组合逻辑进行物理综合优化，以提高性能。
- ◇ Perform register duplication(进行寄存器复制): 指示 Quartus II 软件在适配期间使用寄存器复制对寄存器进行物理综合优化，以提高性能。
- ◇ Perform register retiming(进行寄存器再定时): 指示 Quartus II 软件在配合期间使用寄存器再定时对寄存器进行物理综合优化，以提高性能。
- ◇ Physical synthesis effort(物理综合工作等级): 指定 Quartus II 软件进行物理综合时的工作等级 (Normal、Extra 和 Fast)。

Quartus II 软件不能为反标设计中的适配和物理综合进行这些网表优化。此外，如果在设计上使用一个或多个这些网表优化选项，然后对设计进行反标，如果要保存结果，则必须生成 Verilog Quartus Mapping 文件 (.vqm)。在以后的编译中，必须用 VQM 文件替代最初的设计源代码。

有关信息

请参阅

使用网表优化实现时序逼近

Altera 网站上 *Quartus II Handbook* 第 2 卷
“Netlist Optimizations and Physical Synthesis”

使用 LogicLock 区域达到时序逼近



可以使用 LogicLock 区域达到时序逼近，方法是：在 Timing Closure 平面布局图中分析设计，然后将关键逻辑约束在 LogicLock 区域中。LogicLock 区域通常为分层结构，使用户对模块或模块组的布局和性能有更强的控制。可以在个别节点上使用 LogicLock 功能，例如，将沿着关键路径的节点分配给 LogicLock 区域。

要在设计中使用 LogicLock 区域成功地改进性能，需要详细掌握设计的关键路径。一旦实现了 LogicLock 区域并达到所要的性能，就可以对该区域的内容进行反标，以锁定逻辑布局。

软 LogicLock 区域

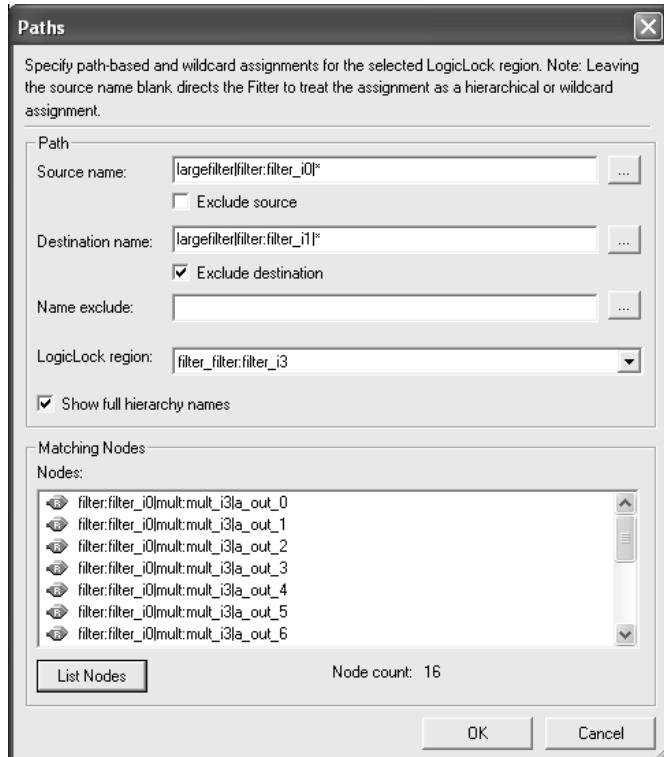
LogicLock 区域具有预定义边界和节点，这些边界和节点分配给一直驻留在边界或 LogicLock 区域范围之内的特定区域。软 LogicLock 区域可以通过删除 LogicLock 区域的固定矩形边界来增强设计性能。启用软区域属性后，Fitter 试图在区域中尽量多布置一些已分配节点并尽可能将它们靠近放置，提高软区域外移动节点的灵活性来满足设计的性能要求。

基于路径的分配

Quartus II 软件可以将特定的源和目标路径分配给 LogicLock 区域，从而可以方便地将关键设计节点组合进一个 LogicLock 区域。通过从 Timing Closure 平面布局图和 Report 窗口的 Timing Analyzer 区域中拖放关键路径至 LogicLock 区域，您可使用 Paths 对话框建立基于路径的分配。

Paths 对话框允许通过识别源节点和目标节点，识别时使用通配符，来指定路径。可单击 List Nodes 来确定会有多少节点分配至 LogicLock 区域。通过单击 Add Path 或双击 LogicLock Region Properties 对话框的 Contents 选项标签，或双击 Timing Closure 平面布局图中的路径，打开该对话框。请参见图 5。

图 5. Paths 对话框



有关信息

使用 LogicLock 方法实现时序逼近

请参阅

Altera 网站上 *Quartus II Handbook* 第 2 卷
“Timing Closure Floorplan”

Altera 网站上 *Quartus II Handbook* 第 2 卷
“LogicLock Design Methodology”

Quartus II Tutorial 的 LogicLock 部分

使用 Design Space Explorer 达到时序逼近



可以使用 Design Space Explorer (DSE) Tcl 脚本、`dse.tcl` 针对时序，来优化设计。DSE 界面可以使用户浏览一定范围内的 Quartus II 选项和设置，自动确定应采用哪种设置以获得工程的最佳可能结果。可以指定允许 DSE 所作修改的级别、优化目标、目标器件和允许的编译时间。

可以使用命令行下的 `quartus_sh` 可执行文件，Quartus II 软件中的 Launch Design Space Explorer 命令 (Tools 菜单)，或者 Windows Start 菜单中 `Quartus II <version number> Design Space Explorer` 命令来运行 Design Space Explorer。关于使用 Design Space Explorer 的详细信息，请参阅 第5章“布局布线”，第94页“使用 Design Space Explorer”

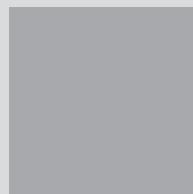
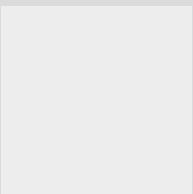
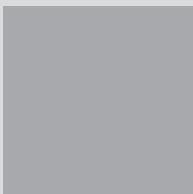
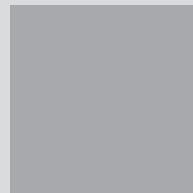
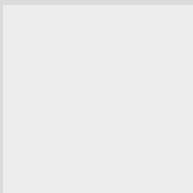
使用渐进式编译达到时序逼近

可以使用渐进式编译，通过分配设计分区、编译设计、处理其他设计分区时，只修改设计中的关键单元布局来达到时序逼近。

关于渐进式编译的详细信息，请参阅 第1章“设计流程”，第28页“自上而下渐进式编译设计流程”和 第5章“布局布线”，第83页“进行完整的渐进式编译”。

第十章

功耗分析



第 10 章 内容:	
简介	156
使用 PowerPlay Power Analyzer 分析功耗	156
指定 Power Analyzer 选项	158
使用 PowerPlay Early Power Estimator	160

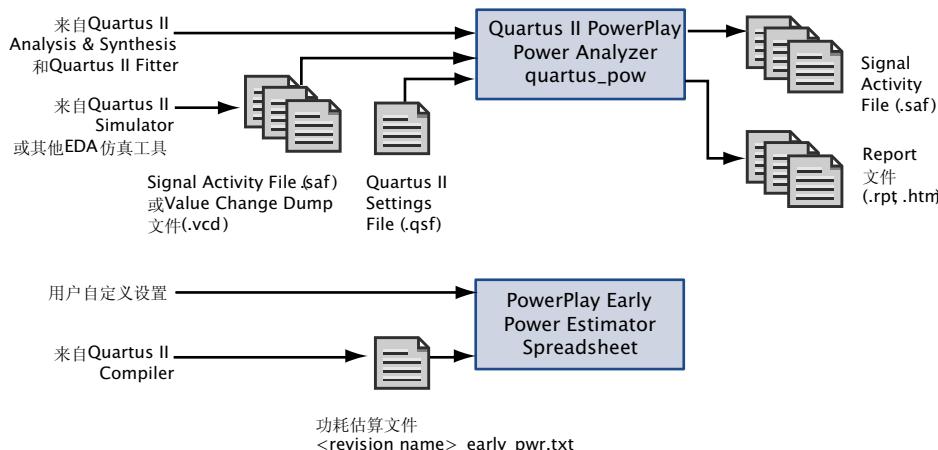
10

简介



Quartus® II PowerPlay Power Analysis Tools 提供的界面使您能够在设计过程中，估算静态和动态功耗。PowerPlay Power Analyzer 进行适配后功耗分析，产生高亮的功耗报告，显示模块类型和实体，以及消耗的功率。Altera® PowerPlay Early Power Estimator 在设计的其他阶段估算功耗，产生估算信息的 Microsoft Excel 电子表格。

图 1. PowerPlay Power Analysis 流程

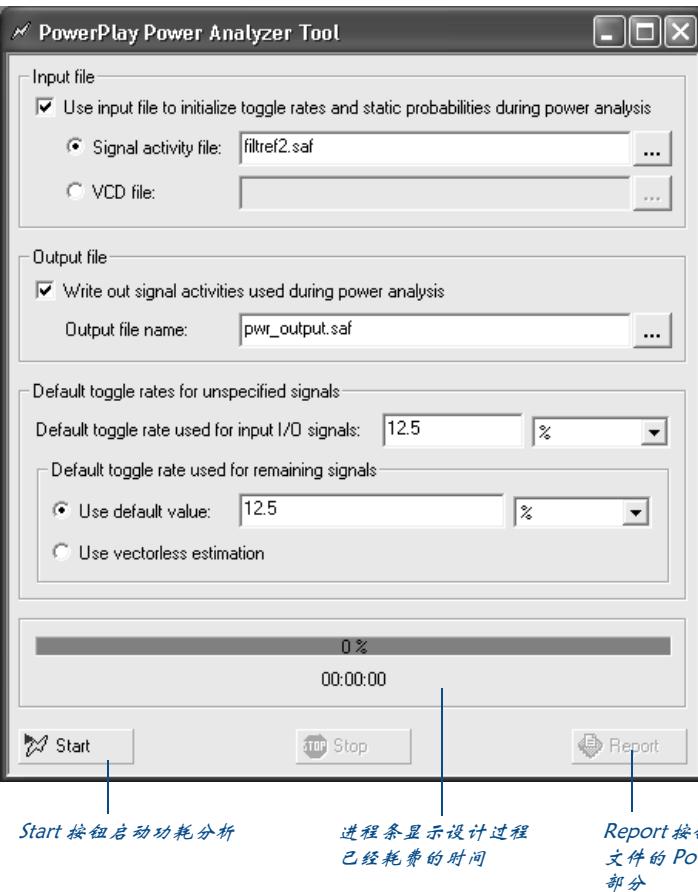


使用 PowerPlay Power Analyzer 分析功耗

成功运行 Analysis & Synthesis 和 Fitter 之后，可以使用 Quartus II PowerPlay Power Analyzer Tool (Tools 菜单)。可以指定使用 Quartus II Simulator 生成的 Signal Activity File (.saf) 等输入文件，还是其他 EDA 仿真工具生成的 Value Change Dump 文件 (.vcd) 作为输入来初始化功耗分析过程中的触发速率和静态几率，以及是否需要将功耗分析过程中使用的信号活动写入到输出文件中。此外，还可以使用 Quartus II 用户界面的用户分配或者在 Quartus II Settings File (.qsf) 中，指定基于实体的触发速率。对于有些器件系列，Quartus II 软件将分析设计拓扑和功能，填补任何丢失的信号活动信息。

然后可以单击 Power Analyzer Tool 窗口中的 Start，启动功耗分析，一个状态条将显示处理时间。功耗分析完成后，单击 Report，显示 Report 文件 (.rpt, .htm)。请参见图 2。

图 2. PowerPlay Power Analyzer Tool



使用 quartus_pow 程序

还可以在命令行或脚本中，使用 **quartus_pow** 可执行文件单独运行 PowerPlay Power Analyzer。在运行 PowerPlay Power Analyzer 之前，必须成功运行 Quartus II Fitter、**quartus_fit**。

quartus_pow 可执行文件建立一个能够由任何文本编辑器查看的独立文本型报告文件。

若要获取有关 **quartus_pow** 可执行文件的帮助，请在命令行提示符下，键入以下命令之一：

```
quartus_pow -h ↵
quartus_pow -help ↵
quartus_pow --help=<topic name> ↵
```



有关信息

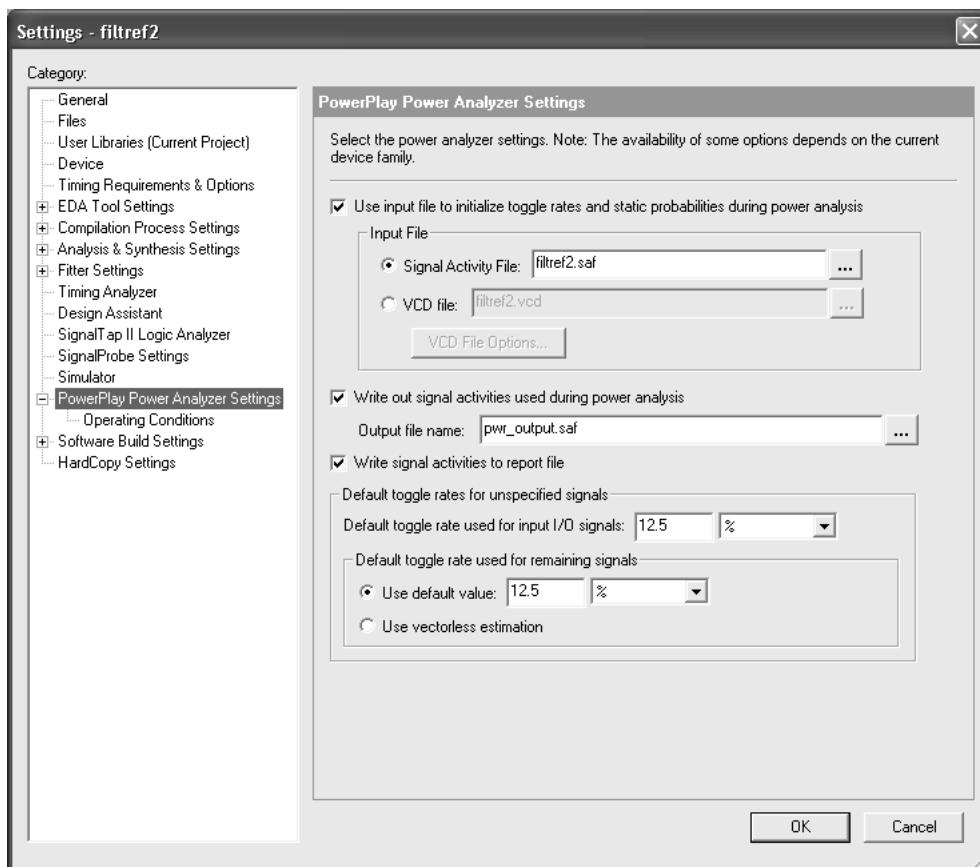
请参阅

使用 Quartus II PowerPlay Power Analyzer
Altera 网站上 *Quartus II Handbook* 第 3 卷
“PowerPlay Power Analyzer”

指定 Power Analyzer 选项

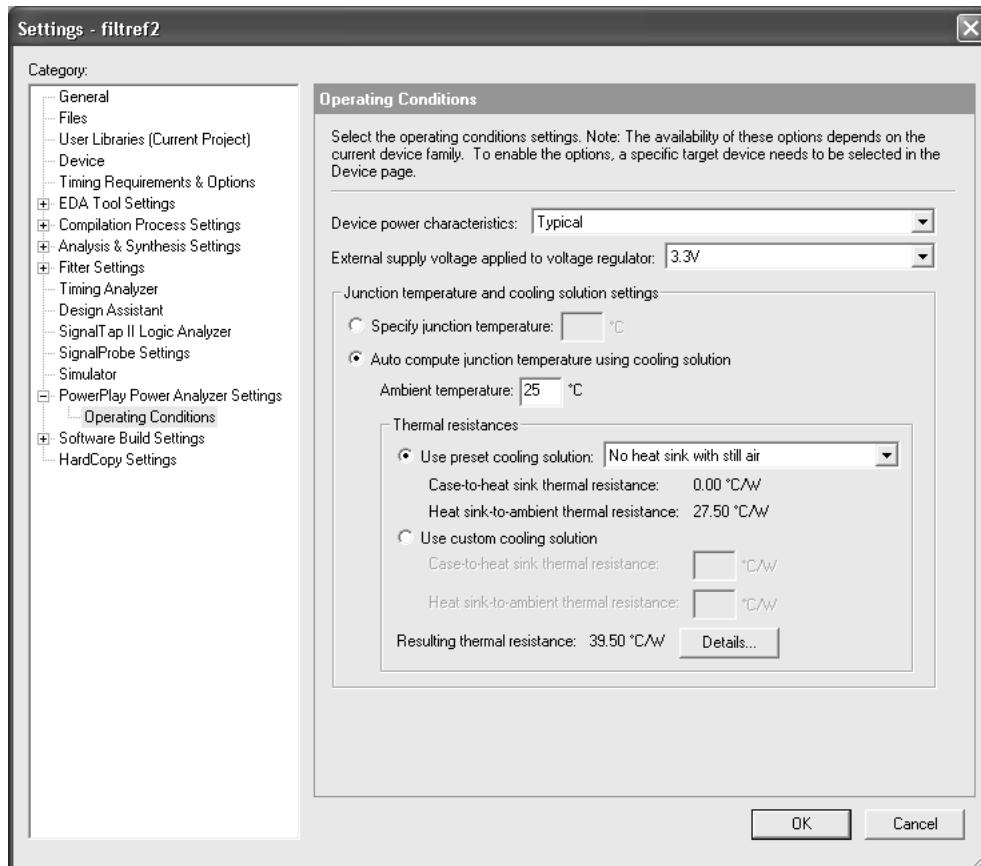
可以在 **Settings** 对话框 (**Assignments** 菜单) **PowerPlay Power Analyzer Settings** 页面中，指定功耗分析的默认设置，包括输入文件类型、写入的输出文件类型、是否将信号活动写入到报告文件中以及默认触发速率的设置等。请参见图 3。

图 3. PowerPlay Power Analyzer Settings 页面



根据不同的目标器件系列，还可以指定功耗分析的默认工作条件。在 Settings 对话框 Operating Conditions 页面中设置结温、散热方案要求和器件特性。请参见图 4。

图 4. Operating Conditions 页面



使用 PowerPlay Early Power Estimator

可以使用 Altera PowerPlay Early Power Estimator 电子表单来计算 Stratix™、Stratix II、Stratix GX、Cyclone™ 以及 MAX® II 器件的功耗，该表单可以从 Altera 网站 <http://www.altera.com/support/devices/estimator/powerplay.html> 的 Power Consumption 部分下载。PowerPlay Early Power Estimator 电子表单基于 Microsoft Excel，专用于当前器件系列。电子表单中的宏功能计算功耗估算，在表单中提供电流 (I_{CC}) 和功耗 (P) 估算。

在设计过程的任何阶段都可以使用 PowerPlay Early Power Estimator 来估算功耗；但是，Altera 建议设计完成后，尽量使用 PowerPlay Power Analyzer，而不是 PowerPlay Power Estimator，以得到最精确的功耗分析。

如果在开始设计之前使用 PowerPlay Early Power Estimator，可以为 PowerPlay Early Power Estimator 指定器件资源、工作频率、触发速率和其他参数。如果在建立设计之后，使用该功能，可以在 Quartus II 软件中编译设计，然后使用 **Generate Early Power Calculator File** 命令（Power 菜单）来生成功耗估算文件，该文件是基于文本的文件，其名称为 **<revision name>_early_pwr.txt**，含有当前器件和设计的功耗信息。然后可以将功耗文件导入到 PowerPlay Early Power Estimator 中。

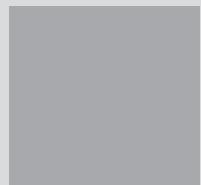
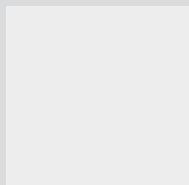
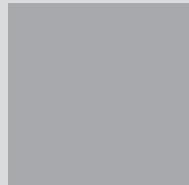
使用 Early Power Estimations

应将 PowerPlay Early Power Estimator 提供的功耗计算看作是功耗的估算，而不是标准。在器件工作期间，应确定验证实际的 ICC，该测量依赖于实际的器件设计和环境工作条件。

有关信息	请参阅
使用 PowerPlay Early Estimator	Altera 网站上的 <i>Power Calculator User Guide</i>
	Altera 网站上 <i>Quartus II Handbook</i> 第 3 卷 “PowerPlay Early Power Estimator”
	Altera 网站上 <i>Application Note 74 (Evaluating Power for Altera Devices)</i>
关于器件要求的信息	Altera 网站上不同器件的手册或数据资料

第十一章

编程和配置



第 11 章 内容:

简介	164
使用 Programmer 对一个或多个器件编程	167
建立辅助编程文件	168
使用 Quartus II 软件通过远程 JTAG 服务器进行编程	174

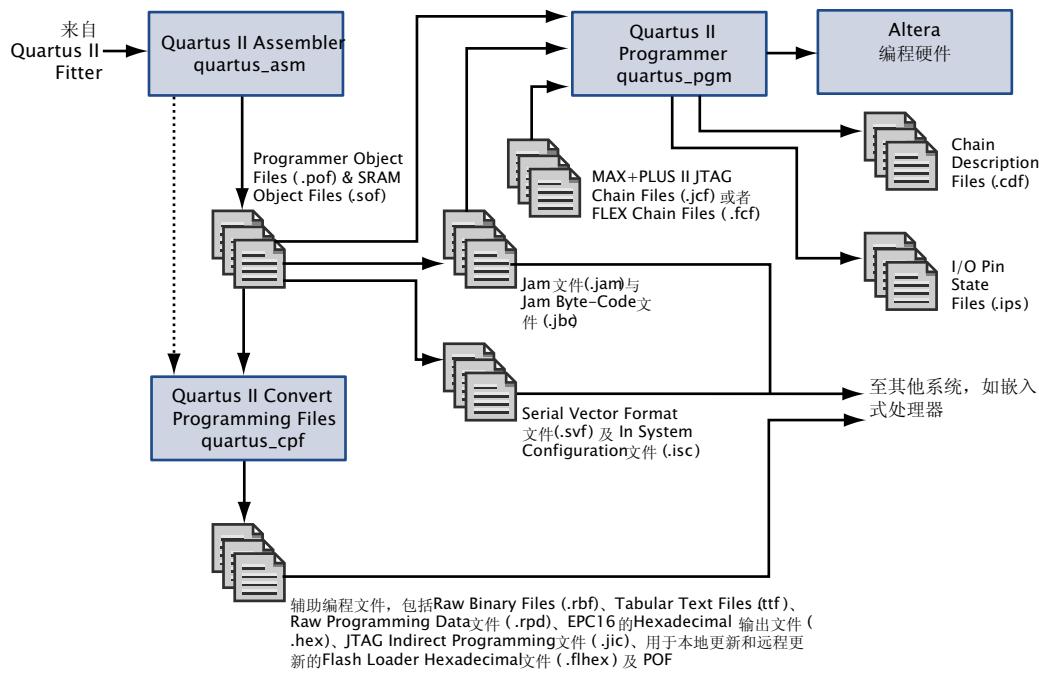
11

简介



使用 Quartus®II 软件成功编译工程之后，就可以对 Altera® 器件进行编程或配置。Quartus II Compiler 的 Assembler 模块生成编程文件，Quartus II Programmer 可以用它与 Altera 编程硬件一起对器件进行编程或配置。还可以使用 Quartus II Programmer 的独立版本对器件进行编程和配置。图 1 所示为编程设计流程。

图 1. 编程设计流程



Assembler 自动将 Fitter 的器件、逻辑单元和引脚分配转换为器件的编程镜像，其形式是目标器件的一个或多个 Programmer Object Files (.pof) 或者 SRAM Object Files (.sof)。

可以在包括 Assembler 模块的 Quartus II 软件中启动完整编译，也可以单独运行 Assembler。

 使用 quartus_asm 可执行文件

还可以在命令提示符下或在脚本中使用 **quartus_asm** 可执行文件单独运行 Assembler。在运行 Assembler 之前，必须成功运行 Quartus II Fitter 可执行文件 **quartus_fit**。

quartus_asm 可执行文件建立可以使用任何文本编辑器查看的独立文本型报告文件。

若要获取有关 **quartus_asm** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_asm -h ↵
quartus_asm -help ↵
quartus_asm --help=<topic name> ↵
```

还可以指示 Assembler 或者 Programmer 通过以下方法之一，以其它格式生成编程文件：

- ◇ **Settings** 对话框 (Assignments 菜单) **Device** 页面的 **Device & Pin Options** 对话框允许指定可选编程文件格式，例如，Hexadecimal (Intel 格式) Output 文件 (.hexout)、Tabular Text Files (.ttf)、Raw Binary Files (.rbf)、Jam 文件 (.jam)、Jam Byte-Code 文件 (.jbc)、Serial Vector Format 文件 (.svf) 和 In System Configuration 文件 (.isc)。
- ◇ **Create/Update > Create JAM、SVF，或者 ISC File** 命令 (File 菜单) 生成 Jam 文件、Jam Byte-Code 文件、Serial Vector Format 文件或 In System Configuration 文件。
- ◇ **Create/Update > Create/Update IPS File** 命令 (File 菜单) 显示 **ISP CLAMP State Editor** 对话框，使您能够建立或更新 I/O Pin State 文件 (.ips)，该文件包括特定器件的引脚状态信息，用于编程期间的引脚状态配置。
- ◇ **Convert Programming Files** 命令 (File 菜单) 将一个或多个设计的 SOF 和 POF 组合并转换为其它辅助编程文件格式，例如，Raw Programming Data 文件 (.rpd)、EPC16 或 SRAM 的 HEXOUT 文件、Local Update 或 Remote Update 的 POF 和 POF、Raw Binary 文件、Tabular Text 文件、JTAG Indirect Configuration 文件 (.jic) 以及 Flash Loader Hexadecimal 文件 (.flhex)。

这些辅助编程文件可用于嵌入式处理器类型的编程环境，而且对于一些 Altera 器件而言，它们还可以由其它编程硬件使用。

Programmer 使用 Assembler 生成的 POF 和 SOF 对 Quartus II 软件支持的所有 Altera 器件进行编程或配置。可以将 Programmer 与 Altera 编程硬件配合使用，例如，MasterBlaster™、ByteBlasterMV™、ByteBlaster™ II 或 USB-Blaster™ 下载电缆；或 Altera Programming Unit(APU)。





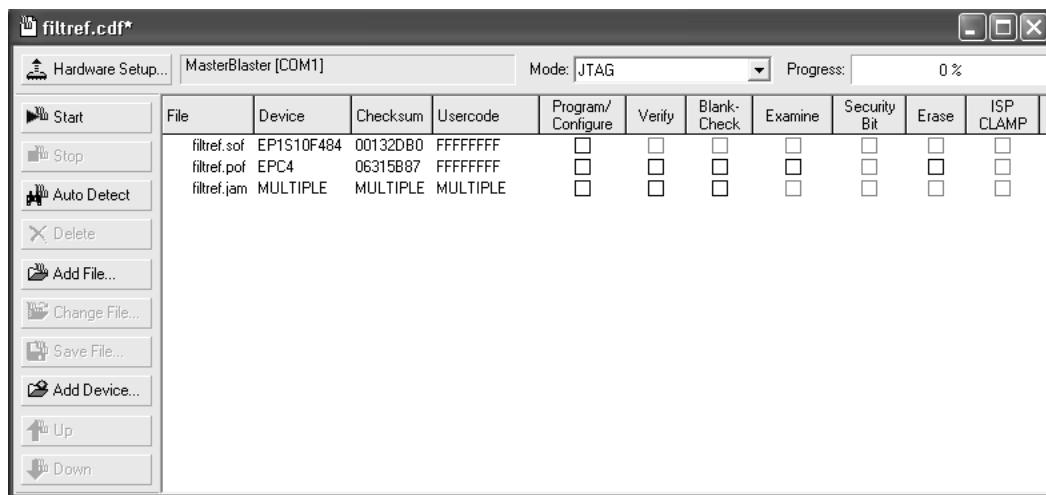
使用独立的 Programmer

如果只想使用 Quartus II Programmer，可以安装 Quartus II Programmer 的独立版本 **quartus_pgmw**，不要安装完整的 Quartus II 软件。

Programmer 允许建立包含设计所用器件名称和选项的 Chain Description File (.cdf)。还可以打开一个 MAX+PLUS II JTAG Chain File (.jcf) 或者 FLEX Chain File (.fcf)，并在 Quartus II Programmer 中保存为 CDF。

对于允许对多个器件进行编程或配置的一些编程模式，CDF 还指定了 SOF、POF、Jam 文件、Jam Byte-Code 文件、设计所用器件的自上而下顺序，以及链中器件的顺序。图 2 所示为 Programmer 窗口。

图 2. Programmer 窗口



使用 quartus_pgm 可执行文件

还可以在命令提示符下或在脚本中使用 **quartus_pgm** 可执行文件，单独运行 Programmer。要在运行 Programmer 之前生成编程文件，可能需要先运行 Assembler 可执行文件 **quartus_asm**。

若要获取有关 **quartus_pgm** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_pgm -h ↵
quartus_pgm -help ↵
quartus_pgm --help=<topic name> ↵
```

Programmer 具有四种编程模式：

- ◇ Passive Serial 模式
- ◇ JTAG 模式
- ◇ Active Serial Programming 模式
- ◇ In-Socket Programming 编程模式

Passive Serial 和 JTAG 编程模式允许使用 CDF 和 Altera 编程硬件对单个或多个器件进行编程。可以使用 Active Serial Programming 模式和 Altera 编程硬件对单个 EPCS1 或 EPCS4 串行配置器件进行编程。可以配合使用 In-Socket Programming 模式与 CDF 和 Altera 编程硬件对单个 CPLD 或配置器件进行编程。

若要使用计算机上没有提供但可通过 JTAG 服务器获得的编程硬件，可以使用 Programmer 指定、连接至远程 JTAG 服务器。

有关信息	请参阅
一般编程信息	Quartus II Help 中的“Programming Files”词汇表定义、“Overview:Working with Chain Description Files”和“Overview:Converting Programming Files”
使用 Programmer	Quartus II Tutorial 中的 Programming 模块
Altera 编程硬件	Altera 网站上的 <i>Altera Programming Hardware Installation Guide</i> , <i>MasterBlaster Serial/USB Communications Cable User Guide</i> , <i>ByteBlaster II Download Cable User Guide</i> , <i>ByteBlasterMV Download Cable User Guide</i> , <i>USB-Blaster Download Cable User Guide</i> 以及 <i>EthernetBlaster Download Cable User Guide</i>
编程硬件安装	<i>Quartus II Installation & Licensing for PCs</i> 以及 <i>Quartus II Installation & Licensing for UNIX and Linux Workstations</i> 手册
专用器件的编程信息	Altera 网站上的 <i>Configuration Handbook</i> 。

使用 Programmer 对一个或多个器件编程

Quartus II Programmer 允许编辑 CDF，CDF 存储器件名称、器件顺序和设计的可选编程文件名称信息。可以使用 CDF，通过一个或多个 SOF、POF 或通过单个 Jam 文件或 Jam Byte-Code 文件对器件进行编程或配置。

以下步骤描述使用 Programmer 对一个或多个器件进行编程的基本流程：

1. 将 Altera 编程硬件与您的系统相连，并安装所需的驱动程序。
2. 进行设计的完整编译，或至少运行 Compiler 的 Analysis & Synthesis、Fitter 和 Assembler 模块。Assembler 自动为设计建立 SOF 和 POF。
3. 打开 Programmer，建立新 CDF。每个打开的 Programmer 窗口代表一个 CDF；可以打开多个 CDF，但每次只能使用一个 CDF 进行编程。
4. 选择编程硬件设置。选择的编程硬件设置将影响 Programmer 中可用的编程模式类型。
5. 选择相应的编程模式，例如，Passive Serial 模式、JTAG 模式、Active Serial Programming 模式或者 In-Socket Programming 模式。
6. 根据不同的编程模式，可以在 CDF 中添加、删除或更改编程文件与器件的顺序。可以指示 Programmer 在 JTAG 链中自动检测 Altera 支持的器件，并将其添加至 CDF 器件列表中。还可以添加用户自定义的器件。
7. 对于非 SRAM 非易失性器件，例如配置器件、MAX 3000 和 MAX 7000 器件，可以指定其他编程选项来查询器件，例如，Verify、Blank-Check、Examine、Security Bit 和 Erase。
8. 如果设计含有 ISP CLAMP State 分配，或者 I/O Pin State File，请打开 ISP CLAMP。
9. 启动 Programmer。

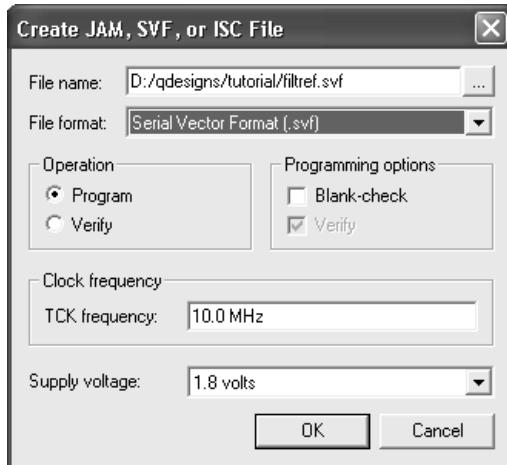
建立辅助编程文件

还可以使用其它格式的文件，如 Jam 文件、Jam Byte-Code 文件、Serial Vector Format 文件、In System Configuration 文件、Raw Binary 文件、Tabular Text 文件或者 I/O Pin State 文件来建立辅助编程文件，供嵌入式处理器等其它系统使用。此外，可以将 SOF 或 POF 转换为其它编程文件格式，例如，远程更新的 POF、本地更新的 POF、EPC16 的 HEXOUT 文件、SRAM 的 HEXOUT 文件或 Raw Programming Data 文件、Tabular Text 文件、JIC 文件和 Flash Loader Hexadecimal 文件。可以使用 **Create/Update > Create JAM, SVF, or ISC File** 命令 (File 菜单)、**Create/Update > Create/Update IPS File** 命令 (File 菜单) 和 **Convert Programming Files** 命令 (File 菜单) 建立这些辅助编程文件。还可以使用 **Settings** 对话框 (Assignments 菜单) **Device** 页面 **Device & Pin Options** 对话框的 **Programming Files** 选项标签，指定在编译期间 Assembler 生成的可选编程文件格式。

建立其它编程文件格式

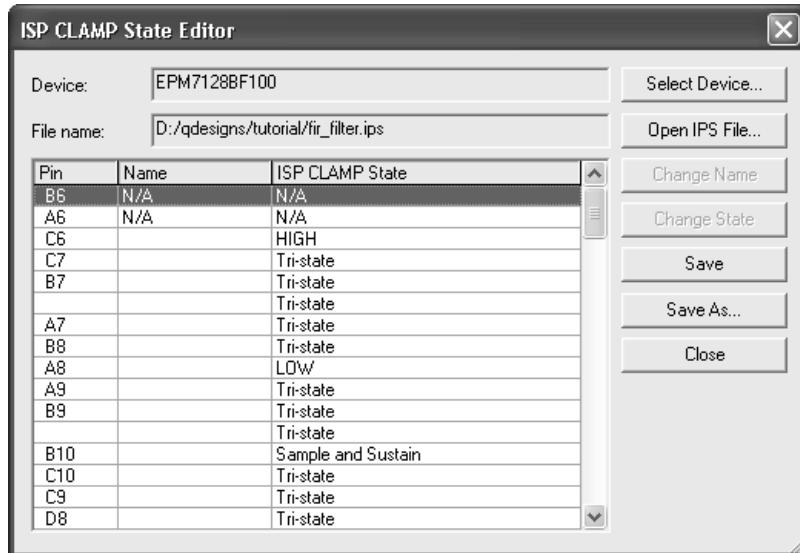
可以使用 **Create/Update > Create JAM, SVF, or ISC File** 命令 (File 菜单) 建立 Jam 文件、Jam Byte-Code 文件、Serial Vector Format 文件或者 In System Configuration 文件。然后，这些文件可以与 Altera 编程硬件或智能主机一起用以配置 Quartus II 软件支持的任何 Altera 器件。还可以将 Jam 文件和 Jam Byte-Code 文件添加至 CDF。请参见图 3。

图 3. *Create JAM, SVF, or ISC File* 对话框



可以使用 **Create/Update > Create/Update IPS File** 命令 (File 菜单) 建立 I/O Pin State 文件，该文件描述编程起始阶段使用的器件引脚 ISP CLAMP 状态。
Create/Update > Create/Update IPS File 命令打开 **ISP CLAMP State Editor** 对话框，如图 4 所示。

图 4. ISP CLAMP State Editor 对话框



以下步骤描述建立 Jam 文件、Jam Byte-Code 文件、Serial Vector Format 文件、In System Configuration 文件或 I/O Pin State 文件的基本流程：

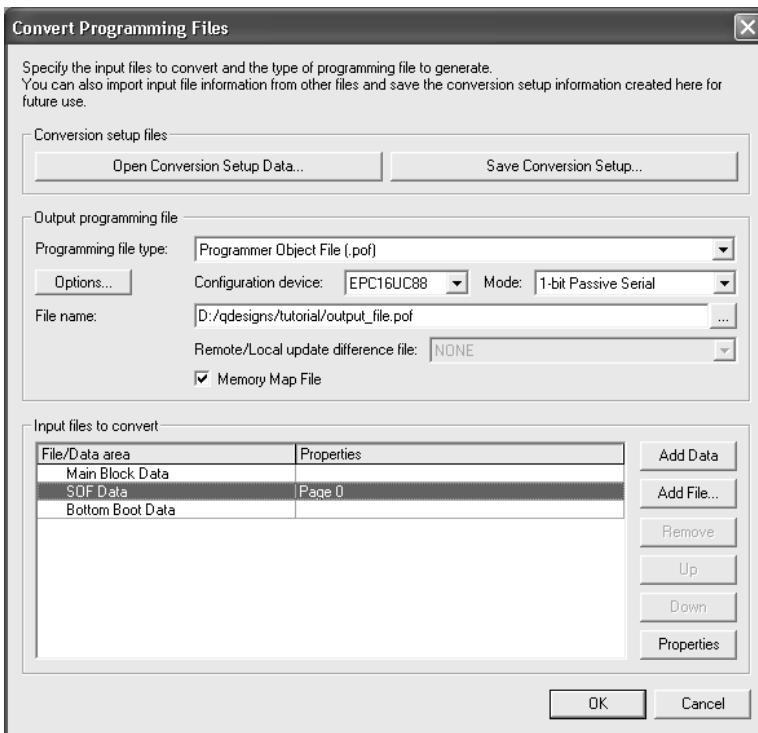
1. 进行设计的完整编译，或至少运行 Compiler 的 Analysis & Synthesis、Fitter 和 Assembler 模块。Assembler 为设计自动建立 SOF 和 POF。
2. 打开 Programmer 窗口，建立新 CDF。
3. 指定 JTAG 模式。
4. 在 CDF 中添加、删除或更改编程文件和器件的顺序。可以指示 Programmer 在 JTAG 链中自动检测 Altera 支持的器件，并将其添加至 CDF 器件列表中。还可以添加用户自定义的器件。
5. 如果要建立 Jam 文件、Jam Byte-Code 文件、Serial Vector Format 文件或者 In System Configuration 文件，请选择 Create/Update > Create Jam, SVF, or ISC File (File 菜单)，指定要建立文件的名称和格式。
6. 如果要建立 I/O Pin State 文件，选择 Create/Update > Create/Update IPS File (File 菜单)，然后在 ISP CLAMP State Editor 对话框 (File 菜单) 中，指定合适的 ISP CLAMP 状态引脚设置和文件的特定名称。

转换编程文件

可以使用 Convert Programming Files 对话框 (File 菜单) 将一个或多个设计的 SOF 或 POF 组合起来并转换为与不同配置方案一起使用的其它编程文件格式。例如，可以将具有远程更新能力的 SOF 添加至远程更新的 POF，用于在远程更新配置模式下对配置器件进行编程，或者可以将 Programmer Object 文件转换为供外部主机使用的 EPC16 HEXOUT 文件。也可以将 POF 转换为与某些配置器件一起使用的 Raw Programming Data 文件。还可以将 SOF 或 POF 转换为 JTAG Indirect Configuration 文件，用于编程 Cyclone™ 器件的配置数据，使其连接至 EPCS1 或 EPCS4 串行配置器件。

可以使用 Convert Programming Files 对话框，对 SRAM 的 HEXOUT 文件、POF、Raw Binary 文件、Tabular Text 文件中存储的 SOF 链进行排列，或指定要在 EPC16 的 HEXOUT 文件中存储的 POF，来设置输出编程文件。在 Convert Programming Files 对话框中指定的设置将保存到 Conversion Setup File (.cof) 中，此文件包含器件和文件名称、器件顺序、器件属性和文件选项等信息。图 5 所示为 Convert Programming Files 对话框。

图 5. Convert Programming Files 对话框



对于 EPC4、EPC8 或 EPC16 配置器件的 POF，还可以指定以下信息：

- ◇ 建立不同的配置比特流，这些比特流存储在配置存储器空间页面中。
- ◇ 在每个页面中建立 SOF 并行链。
- ◇ 排列闪存中存储的 SOF 和 Hexadecimal (Intel 格式) 文件 (.hex) 的顺序。
- ◇ 指定 **SOF Data** 项和 HEX 文件的属性。
- ◇ 在配置存储器空间中添加或删除 **SOF Data** 项。
- ◇ 根据需要，可以建立 Memory Map 文件 (.map)。

对于本地更新的 POF 和远程更新的 POF，可以指定以下信息：

- ◇ 在配置存储器空间中添加或删除具有远程更新能力的 POF 和 SOF。
- ◇ 指定 **SOF Data** 的属性。
- ◇ 添加或删除 **SOF Data** 项。
- ◇ 根据需要，可以建立 Memory Map 文件，生成远程更新差异文件和本地更新差异文件。

还可以使用 **Convert Programming Files** 对话框将多个 SOF 排列和组合为 Active Serial Configuration 模式下的单个 POF。POF 可用于对 EPCS1 或 EPCS4 串行配置器件进行编程，然后，可以用该配置器件通过 Cyclone 器件配置多个器件。



使用 quartus_cpf 可执行文件

还可以在命令提示符下或在脚本中使用 **quartus_cpf** 可执行文件，单独运行 Convert Programming Files 功能。要在运行 Programmer 之前生成编程文件，可能需要先运行 Assembler 可执行文件 **quartus_asm**。

若要获取有关 **quartus_cpf** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_cpf -h ↵  
quartus_cpf -help ↵  
quartus_cpf --help=<topic name> ↵
```

以下步骤描述转换编程文件的基本流程：

1. 运行 Compiler 的 Assembler 模块。Assembler 自动为设计建立 SOF 和 POF。
2. 使用 **Convert Programming Files** 对话框并指定所要建立编程文件的格式和名称。
3. 指定与编程文件配置存储器空间相兼容的配置模式。
4. 为编程文件类型和目标器件指定相应的编程选项。

5. (可选) 通过选择差异文件的类型，使 Programmer 为远程更新的 Programmer Object 文件或本地更新的 Programmer Object 文件生成远程更新差异文件或本地更新差异文件。
6. 添加或删除 **SOF Data** 项并将它们分配给页面。
7. (可选) 添加、删除或更改要为一个或多个 **SOF Data** 项或 **POF Data** 项而转换的 SOF 和 POF 的顺序。
8. (可选) 为 EPC4、EPC8 或 EPC16 配置器件，添加 HEX 文件至 POF 的 **Bottom Boot Data** 或 **Main Block Data** 项中，并指定 **SOF Data** 项、**POF Data** 项和 HEX 文件的其他属性。
9. 保存 **Input files to convert** 列表的当前状态并输出 Conversion Setup 文件中的编程文件设置。
10. 转换文件。根据需要，还可以指定要建立的 Memory Map 文件。

有关信息	请参阅
在系统可编程能力和电路内可重新配置能力	Altera 网站上的 <i>Configuration Handbook</i>
	Altera 网站上的 <i>Application Note 100 (In-System Programmability Guidelines)</i>
	Altera 网站上的 <i>Application Note 95 (In-System Programmability in MAX Devices)</i>
	Altera 网站上的 <i>Application Note 122 (Using Jam STAPL for ISP & ICR via an Embedded Processor)</i>
在系统编程	Quartus II Tutorial 中的 Programming 部分
远程系统配置	Altera 网站上的 <i>Stratix Device Handbook</i> 第 2 卷 “Using Remote System Configuration with Stratix & Stratix GX Devices” 以及 “Using Remote System Configuration with Stratix & Stratix GX Devices”

使用 Quartus II 软件通过远程 JTAG 服务器进行编程

在 Programmer 窗口的 **Hardware** 按钮或 Edit 菜单的 **Hardware Setup** 对话框中，可以添加能够连机访问的远程 JTAG 服务器，这样，就可以使用本地计算机未提供的编程硬件，配置本地 JTAG 服务器设置，让远程用户连接到本地 JTAG 服务器。

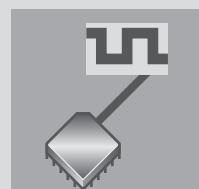
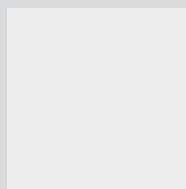
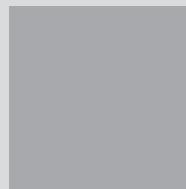
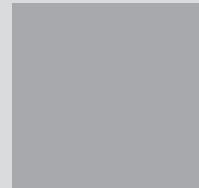
可以在 **Hardware Setup** 对话框 **JTAG Settings** 选项标签下的 **Configure Local JTAG Server** 对话框中指定可以连接至 JTAG 服务器的远程客户端。

在 **Hardware Setup** 对话框 **JTAG Settings** 选项标签 **Add Server** 对话框中指定要连接的远程服务器。连接到远程服务器之后，与远程服务器相连的编程硬件将显示在 **Hardware Settings** 选项标签中。

有关信息	请参阅
使用 Local JTAG Server	Quartus II Help 中的“Configuring Local JTAG Server Settings,” 和 “Adding a JTAG Server”

第十二章

调试



第 12 章 内容:

简介	176
使用 SignalTap II Logic Analyzer	177
使用 SignalProbe	184
使用 In-System Memory Content Editor	187
使用 RTL Viewer 和 Technology Map Viewer	189
使用 Chip Editor	190

12

简介



系统全速运行时，Quartus® II SignalTap® II Logic Analyzer 和 SignalProbe™ 功能可以在系统分析内部器件节点和 I/O 引脚。SignalTap II Logic Analyzer 使用嵌入式逻辑分析仪，根据用户定义的触发条件，将信号数据通过 JTAG 端口送往 SignalTap II Logic Analyzer 或者外部逻辑分析仪、示波器。也可以使用 SignalTap II Logic Analyzer 的单独版本来捕获信号。SignalProbe 功能使用未用器件布线资源上的渐进式布线，将选定信号送往外部逻辑分析仪或示波器。图 1 和图 2 显示了 SignalTap II 和 SignalProbe 调试流程。

图 1. SignalTap II 调试流程

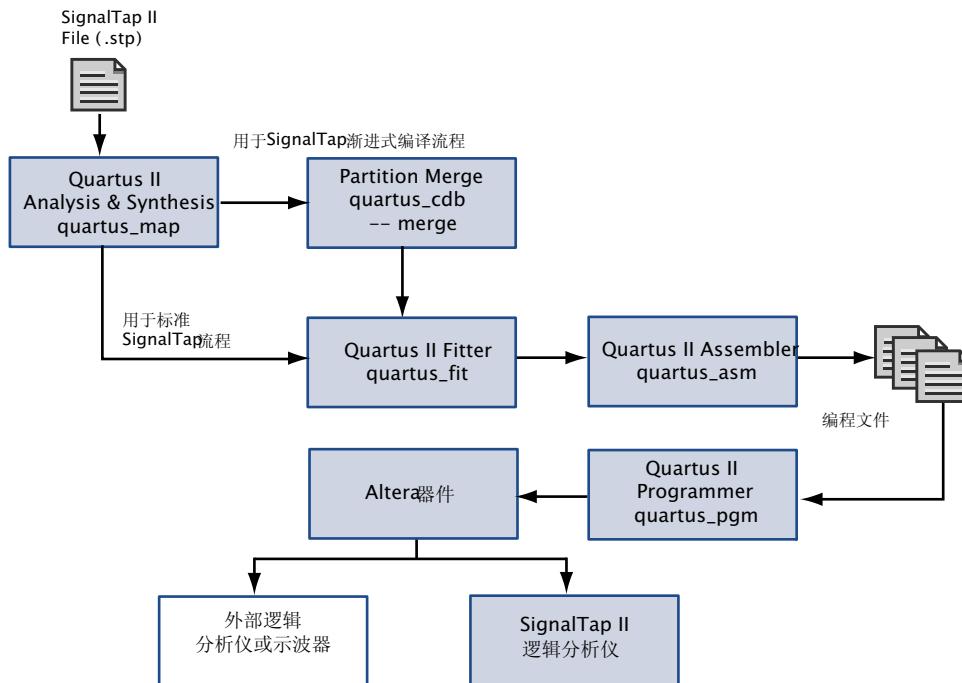
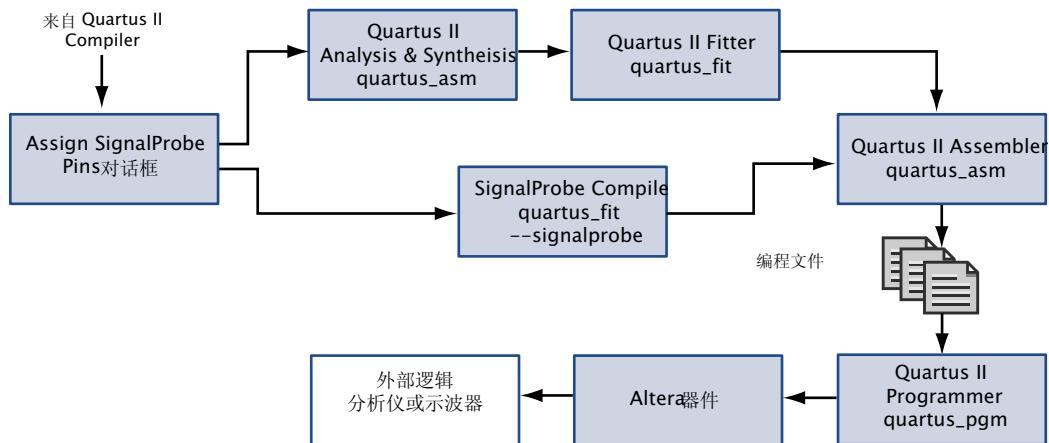


图 2. SignalProbe 调试流程



使用 SignalTap II Logic Analyzer



SignalTap II Logic Analyzer 是第二代系统级调试工具，可以捕获和显示实时信号行为，观察系统设计中硬件和软件之间的相互作用。Quartus II 软件可以选择要捕获的信号、开始捕获信号的时间以及要捕获多少数据样本。还可以选择是将数据从器件的存储器块通过 JTAG 端口传递至 SignalTap II Logic Analyzer，还是至 I/O 引脚以供外部逻辑分析仪或示波器使用。

可以使用 MasterBlaster™、ByteBlasterMV™、ByteBlaster™ II、USB-Blaster™ 或 EthernetBlaster 通信电缆将配置数据下载到器件中。这些电缆还用于将捕获信号数据从器件的 RAM 资源上载至 Quartus II 软件。然后，Quartus II 软件将 SignalTap II Logic Analyzer 采集的数据显示为波形。

设置和运行 SignalTap II Logic Analyzer

使用 SignalTap II Logic Analyzer 之前，必须先建立 SignalTap II 文件 (.stp)，此文件包括所有配置设置并以波形显示所捕获的信号。一旦设置了 SignalTap II 文件，就可以编译工程，对器件进行编程并使用逻辑分析仪采集、分析数据。

每个逻辑分析仪实例均嵌入到器件的逻辑中。SignalTap II Logic Analyzer 在单个器件上支持多达 1,024 个的通道和 128K 采样。

编译之后，可以使用 **Run Analysis** 命令（**Processing** 菜单）运行 SignalTap II 逻辑分析仪。



Using the quartus_stp 可执行文件

还可以通过使用 **quartus_stp** 可执行文件，在命令提示符或 Tcl 或命令行脚本中单独运行 SignalTap II Logic Analyzer。必须首先运行 **quartus_stp** 可执行文件来建立 SignalTap II 文件。编译完毕后，可以运行 SignalTap II Logic Analyzer 来捕获信号。

若要获取 **quartus_stp** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

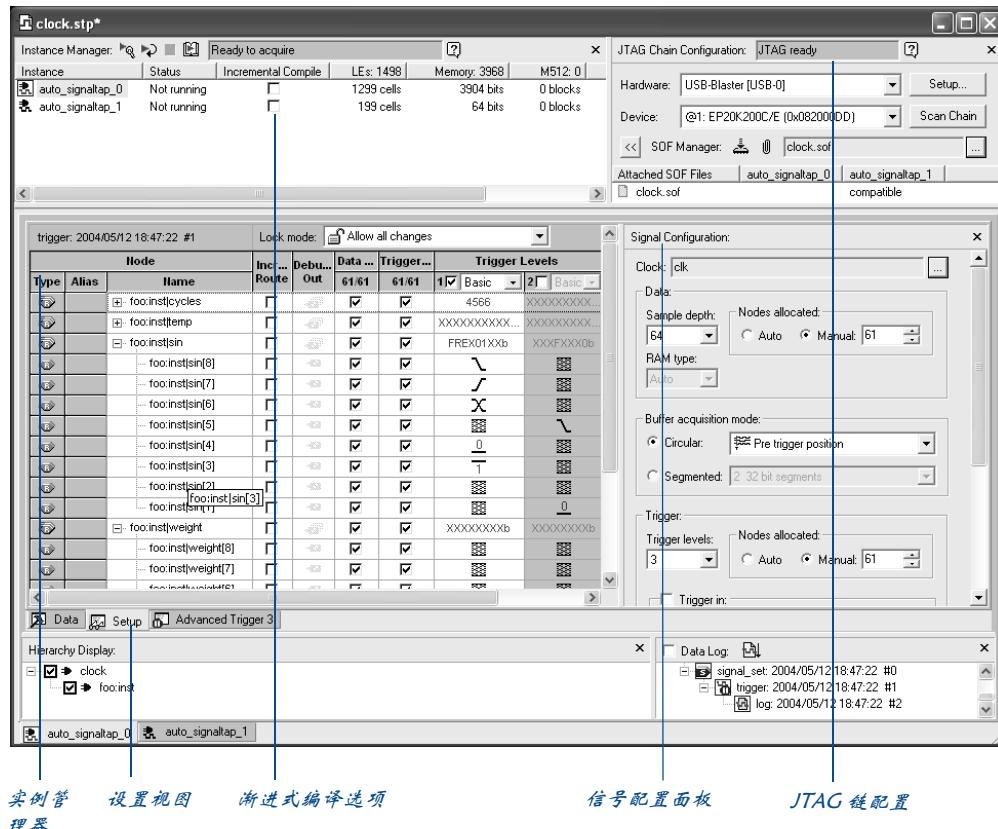
```
quartus_stp -h ↵
quartus_stp -help ↵
quartus_stp --help=<topic name> ↵
```

以下步骤描述设置 SignalTap II 文件和采集信号数据的基本流程：

1. 建立新的 SignalTap II 文件。
2. 向 SignalTap II 文件添加实例，并向每个实例添加节点。可以使用 Node Finder 中的 SignalTap II 滤波器查找所有预综合和适配后的 SignalTap II 节点。
3. 给每个实例分配一个时钟。
4. 设置其它选项，例如采样深度和触发级别，并将信号分配给数据 / 触发输入和调试端口。
5. 根据需要，可指定 Advanced Trigger 条件。
6. 编译设计。
7. 对器件进行编程。
8. 在 Quartus II 软件中或使用外部逻辑分析仪或示波器采集、分析信号数据。

图 3 所示为 SignalTap II Logic Analyzer。

图 3. SignalTap II Logic Analyzer



使用单独的 SignalTap II Logic Analyzer

如果希望仅使用 SignalTap II Logic Analyzer，可以使用单独的图形用户界面版的 SignalTap II Logic Analyzer，quartus_stpw。

可以使用以下功能设置 SignalTap II Logic Analyzer：

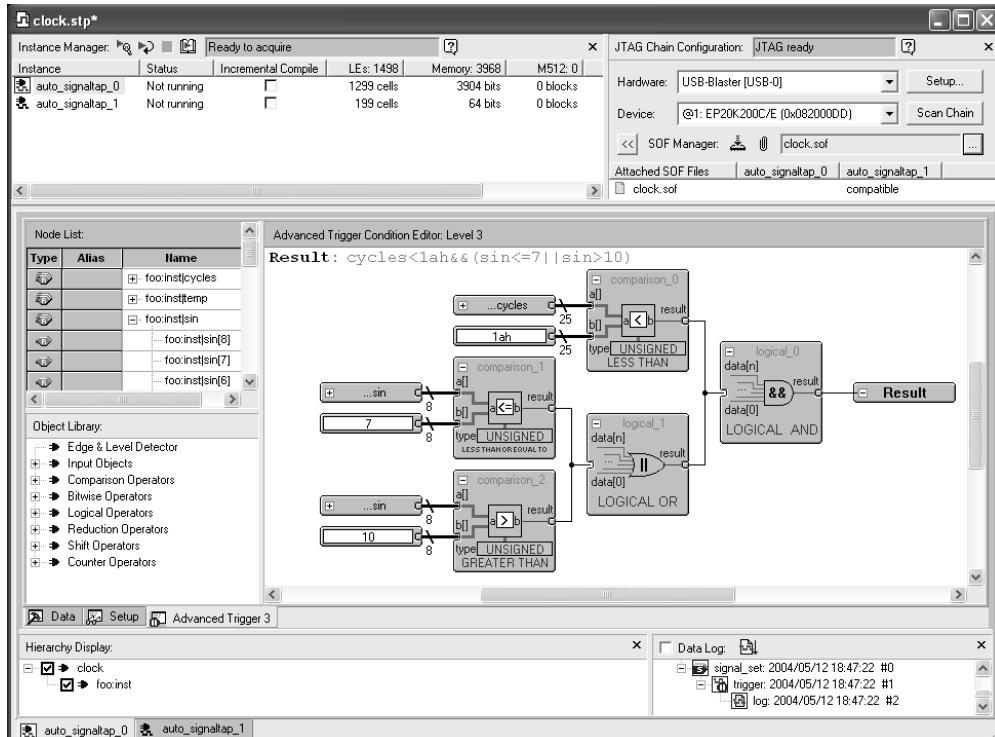
- ◇ **Instance Manager(实例管理器):** Instance Manager 在每个器件中逻辑分析仪的多个嵌入式实例上建立并进行 SignalTap II 逻辑分析。可以使用它在 SignalTap II 文件中对单独和独特的逻辑分析仪实例建立、删除、重命名、应用设置。Instance Manager 显示当前 SignalTap II 文件中的所有实例、每个相关实例的当前状态以及相关实例中使用的逻辑单元和存

储器比特的数量。Instance Manager 可以协助检查每个逻辑分析仪在器件上要求的资源使用量。可以选择多个逻辑分析仪并选择 Run Analysis (Processing 菜单) 来同时启动多个逻辑分析仪。

- ◇ Triggers(触发): 触发是由逻辑电平、时钟沿和逻辑表达式定义的一种逻辑事件模式。SignalTap II Logic Analyzer 支持多级触发、多个触发位位置、多段触发以及外部触发事件。使用 SignalTap II Logic Analyzer 窗口中的 **Signal Configuration** 面板来设置触发选项，并可通过选择 SignalTap II Logic Analyzer 窗口的 **Setup** 选项标签 **Trigger Levels** 列中的 **Advanced** 来指定高级触发。

根据内部总线或节点的数据值，高级触发提供建立灵活的、用户定义逻辑表达式和条件的功能。使用 **Advanced Trigger** 选项标签，可从 **Node List** 和 **Object Library** 中拖放符号来建立逻辑表达式，其中包括逻辑、比较、比特操作、减法、位移运算符以及事件计数器。图 4 显示了 SignalTap II 窗口中的 **Advanced Trigger** 选项标签。

图 4. SignalTap II 窗口的 Advanced Triggers 选项标签



可以给逻辑分析仪配置最多十个触发器级别，帮助您只查看最重要的数据。可以指定四个单独的触发位置：前、中、后和连续。触发位置允许指定在选定实例中，触发之前和触发之后应采集的数据量。

分段模式允许通过将存储器分为不同的时间段，为周期性事件捕获数据，而无需分配大采样深度。

- ◇ **Incremental Routing(渐进式布线):** 渐进式布线功能允许在不执行完整重新编译的情况下分析适配后节点，从而有利于缩短调试过程。

在使用 SignalTap II 渐进式布线功能之前，必须打开 **Settings** 对话框 (Assignments 菜单) SignalTap II Logic Analyzer 页面中的 **Automatically turn on smart compilation if conditions exist in which SignalTap II with incremental routing is used**，进行智能编译。此外，在编译设计之前，必须使用 **Trigger Nodes allocated** 和 **Data Nodes allocated** 框保留 SignalTap II 渐进式布线的触发或数据节点。通过选择 Node Finder 中 Filter 列的 **SignalTap II: post-fitting**，来找到 SignalTap II 渐进式布线源的节点。

当设计不是渐进式编译模式时，可以使用渐进式布线。工程为渐进式编译模式，不进行完整编译而分析适配后节点，则应使用 SignalTap II 渐进式编译。更详细的信息，请参考下一节，**渐进式编译使用 SignalTap II Logic Analyzer**。

- ◇ **Attach Programming File(附加编程文件):** 允许在单个 SignalTap II 文件中采用多个 SignalTap II 配置（触发设置）和相关的编程文件。可以使用 SOF Manager 添加、重命名、删除 SRAM Object Files (.sof)，从 SignalTap II 文件中提取 SOF，也可以对器件进行编程。

渐进式编译使用 SignalTap II Logic Analyzer

渐进式编译功能不必对设计进行完整的编译，使用 SignalTap II Logic Analyzer 逐步分析适配后节点，能够极大的缩短调试处理时间。

与渐进式适配特性不同，渐进式编译功能不需要进行智能编译，可用于渐进式编译模式。

以下步骤描述了渐进式编译进行 SignalTap II 逻辑分析的基本流程：

1. 确定打开 **Settings** 对话框 (Assignments 菜单) Compilation Process Settings 页面的 **Full incremental compilation** 选项。
2. 删除所有设计分区分配。

3. 建立并打开一个 SignalTap II 文件。
4. 打开要逐步编译的实例旁的 SignalTap II Logic Analyzer Instance Manager 中的 **Incremental Compile** 选项。(参见 179 页, 图 3)。

使用渐进式编译时, 必须使用适配后 SignalTap II 节点; 如果节点不是适配后节点, **Incremental Compile** 选项打开后, SignalTap II Logic Analyzer 将其转换为适配后节点。
5. 运行 SignalTap II Logic Analyzer。对于渐进式模式中的实例, 不需要进行完整的工程编译, 任何修改都采用逐步编译。

关于渐进式编译的详细信息, 请参考 第 1 章 “设计流程”, 第 28 页 “自上而下渐进式编译设计流程” 和 第 5 章 “布局布线”, 第 83 页 “进行完整的渐进式编译”。

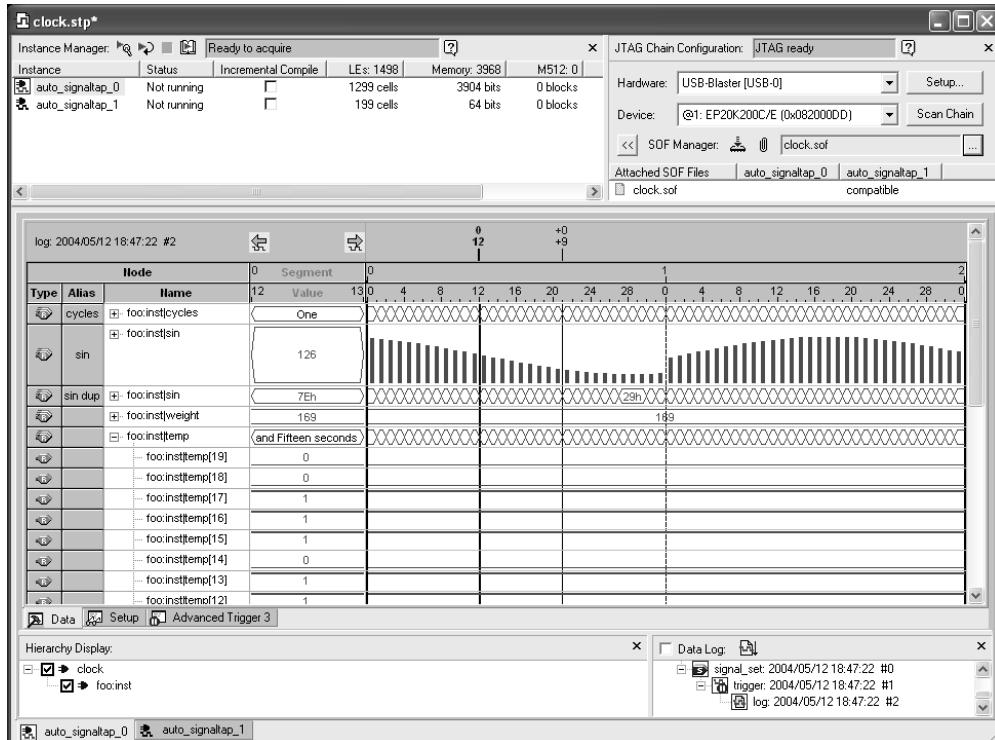
有关信息	请参考
使用 Quartus II 渐进式编译	Altera 网站 <i>Quartus II Handbook</i> , 第 1 卷 “Quartus II Incremental Compilation”
	Quartus II Help 中的 “Overview: Using Incremental Compilation”

分析 SignalTap II 数据

在使用 SignalTap II Logic Analyzer 查看逻辑分析的结果时, 数据存储在器件内部存储器中, 通过 JTAG 端口导入到逻辑分析仪的波形视图中。

在波形视图中, 可以插入时间栏, 对齐节点名称, 复制节点; 建立、重命名总线和取消总线组合; 指定总线值的数据格式; 还可以打印波形数据。数据日志用于建立波形, 此波形显示 SignalTap II Logic Analyzer 采集的数据历史记录。数据以分层方式组织; 使用相同触发器捕获的数据日志将组成一组, 放在 Trigger Sets 中。图 5 显示了波形视图。

图 5. SignalTap II 波形视图

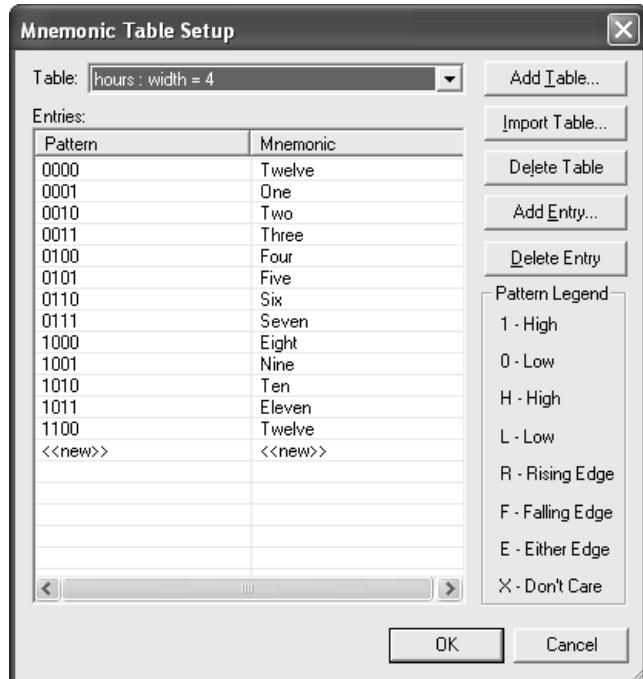


Waveform Export 应用程序允许将捕获的数据导出为 EDA 工具可以使用的以下工业标准格式：

- ◇ Comma Separated Values 文件 (.csv)
- ◇ Table 文件 (.tbl)
- ◇ Value Change Dump 文件 (.vcd)
- ◇ Vector Waveform 文件 (.vwf)
- ◇ Joint Photographic Experts Group 文件 (.jpeg)
- ◇ Bitmap 文件 (.bmp)

还可以配置 SignalTap II Logic Analyzer，为一组信号建立助记表。助记表功能允许将预定义名称分配给一组位模式，使捕获的数据更有意义。请参见图 6。

图 6. Mnemonic Table Setup 对话框



有关信息

使用 SignalTap II Logic Analyzer

请参考

Altera 网站上 *Quartus II Handbook* 第 3 卷
“Design Debugging Using SignalTap II
Embedded Logic Analyzer”

Quartus II Help 中的 “Overview:Using the
SignalTap II Logic Analyzer”

使用 SignalProbe



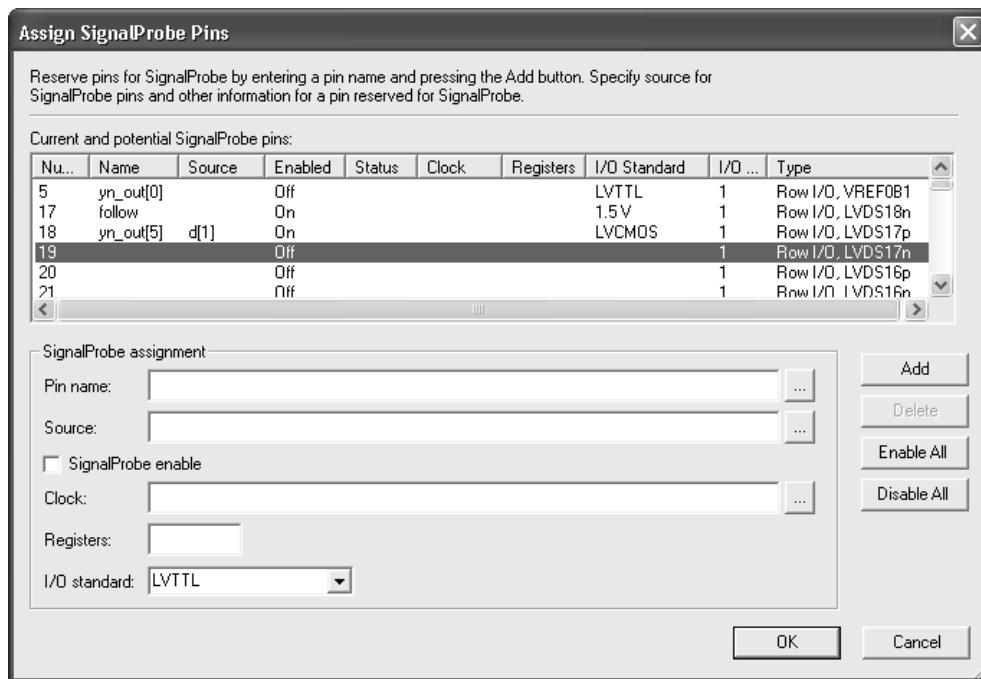
SignalProbe 功能允许在不影响设计中现有适配的情况下，将用户指定信号连接到输出引脚，不需要再进行一次完整编译，就可以调试信号。从一个已经完全布线过的设计开始，可以选择和布线要调试的信号，通过以前保留或当前未使用的 I/O 引脚进行调试。

SignalProbe 功能允许指定设计中要调试的信号，执行一次 SignalProbe 编译，将那些信号与未使用或保留的输出引脚相连，发送信号至外部逻辑分析仪。在分配引脚、查找可用 SignalProbe 源时，可以使用 Node Finder。SignalProbe 编译时间通常为正常编译时间的 20 到 30%。

若要使用 SignalProbe 功能来保留引脚，对设计进行 SignalProbe 编译，请执行以下操作：

1. 进行设计的完整编译。
2. 选择要调试的信号以及信号要通过的 I/O 引脚，打开 **Settings** 对话框 (Assignments 菜单) **SignalProbe Settings** 页面的 **SignalProbe Pins** 对话框的 SignalProbe 功能。请参见 186 页，图 7。
3. 使用 **Start > Start SignalProbe Compilation** 命令 (Processing 菜单) 执行 SignalProbe 编译。SignalProbe 编译在不影响设计适配的情况下编译设计，对 SignalProbe 信号的布线比正常编译速度快。另一种方法是打开 **Settings** 对话框 **SignalProbe Settings** 页面中的 **Automatically route SignalProbe signals during compilation**，然后选择 **Start Compilation**(Processing 菜单)，使完整编译中包含可能影响设计布局布线的 SignalProbe 连接。
4. 给器件配置新编程数据，测试信号。

图 7. Assign SignalProbe Pins 对话框



保留 SignalProbe 引脚时，也可使用寄存器流水线功能来忽略抖动，强制信号状态在时钟边沿上输出，或延时一个信号输出。还可使用寄存器流水线功能同步来自信号总线的多个 SignalProbe 输出，或防止 SignalProbe 布线因为 f_{MAX} 更改而变成关键路径。

如果 SignalProbe 编译期间，Complier 无法对 SignalProbe 信号进行布线，可以通过修改上次适配结果改动来实现 Complier 对 SignalProbe 信号的布线。

还可以将 SignalProbe 功能与 Tcl 配合使用。使用 Tcl 命令，可以添加和删除 SignalProbe 分配和源，对设计进行 SignalProbe 编译，在完整编译中编译已布线的 SignalProbe 信号。



有关信息

使用 SignalProbe 功能

请参考

Altera 网站上 *Quartus II Handbook* 第 3 卷
“Quick Design Debugging Using SignalProbe”

Quartus II Help 中的 “SignalProbe
Introduction”

TCL 命令与 SignalProbe 功能配合使用

Altera 网站上 *Quartus II Handbook* 第 2 卷
“Tcl Scripting”。

使用 In-System Memory Content Editor



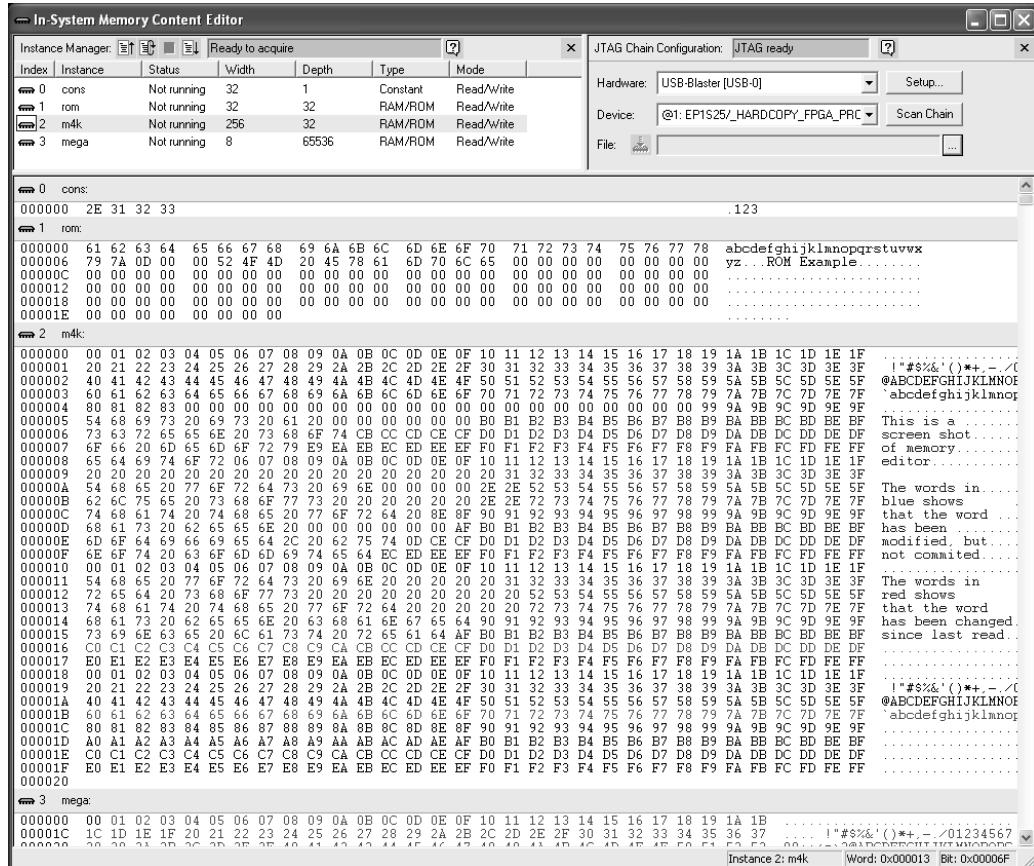
In-System Memory Content Editor 允许在运行时，独立于设计的系统时钟，来查看和修改 RAM、ROM 或者寄存器内容。调试节点使用标准编程硬件，通过一个 JTAG 接口与 In-System Memory Content Editor 进行通信。

可以通过 **MegaWizard® Plug-In Manager** (Tools 菜单)，使用 In-System Memory Content Editor 来设置和例化 `lpm_rom`、`lpm_ram_dq`、`altsyncram` 和 `lpm_constant` 宏功能模块，也可以使用 `lpm_hint` 宏功能参数，直接在设计中例化这些宏功能。

In-System Memory Content Editor (Tools 菜单) 用于捕获和更新器件中的数据。可以导入和导出 Memory Initialization File(.mif)、Hexadecimal (Intel 格式) 文件 (.hex) 和 RAM Initialization File(.rif) 格式。In-System Memory Content Editor 具有以下特性：

- ◇ **Instance Manager(实例管理器)**: 含有存储器实例的列表，包括目录、实例名称、状态、数据宽度、数据深度、类型和模式。Instance Manager 控制哪个存储器模块具有被查看的、卸载的或者更新的数据。Instance Manager 的命令影响全部的所选存储器模块。
- ◇ **JTAG Chain Configuration(JTAG 链配置)**: 用于选择编程硬件和器件，以获取数据或读出数据，并为编程选择 SRAM Object File(.sof)。
- ◇ **HEX Editor(HEX 编辑器)**: 运行时，对在系统存储器进行编辑、保存更改，显示存储器模块中的当前数据，更新或卸载存储器模块的所选部分。可以使用 Go To 命令 (右键弹出菜单) 自动找到特定实例中特定存储器模块的特定数据地址。每个 16 进制数值以空格分割来显示字。存储器地址显示在左列中，ASCII 值 (如果字宽度是 8 的整数倍) 显示在右列中。每个存储器实例在 HEX Editor 中具有不同的窗口视图。图 8 所示为 In-System Memory Content Editor 窗口中的 HEX Editor。

图 8. In-System Memory Content Editor 窗口



有关信息

请参考

使用 In-System Memory Content Editor

Altera 网站上 *Quartus II Handbook* 第 3 卷：
“In-System Editing of Memory and Constants”

Quartus II Help 中的 “Overview: Using the In-System Memory Content Editor”

使用 RTL Viewer 和 Technology Map Viewer



可以使用 RTL Viewer 在进行分析和详细描述后分析设计。RTL Viewer 提供设计的逻辑门级原理图和层次结构列表，列出整个设计网表的实例、基本单元、引脚和网络。可过滤显示在视图上的信息，浏览设计视图的不同页面来检查设计并确定应当作的更改。



Quartus II Technology Map Viewer 提供设计的底级或基元级特定技术原理表。Technology Map Viewer 包括一个原理视图，以及一个层次列表，列出整个设计网表的实例、基本单元、引脚和网络。

关于 RTL Viewer 和 Technology Map Viewer 的详细信息，请参考第 4 章“综合”第 73 和 77 页的“使用 RTL Viewer 分析综合结果”和“采用 Technology Map Viewer 分析综合结果”。

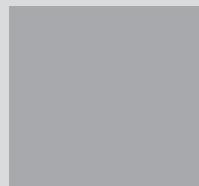
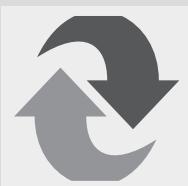
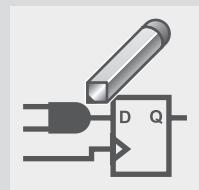
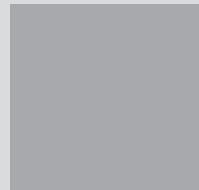
使用 Chip Editor



可以将 Chip Editor 与 SignalTap II 和 SignalProbe 调试工具一起使用，加快设计验证，并逐步修复在设计验证期间未解决的错误。运行 SignalTap II 逻辑分析仪或使用 SignalProbe 功能验证信号之后，可以使用 Chip Editor 来查看编译后布局布线的详细信息。还可以使用 Resource Property Editor 对逻辑单元、I/O 单元或 PLL 基元的属性和参数进行编译后编辑，而无需执行完整的重新编译。有关使用芯片编辑器的详细信息，请参阅下一章，第 13 章“工程更改管理”。

第十三章

工程更改管理



第 13 章 内容:

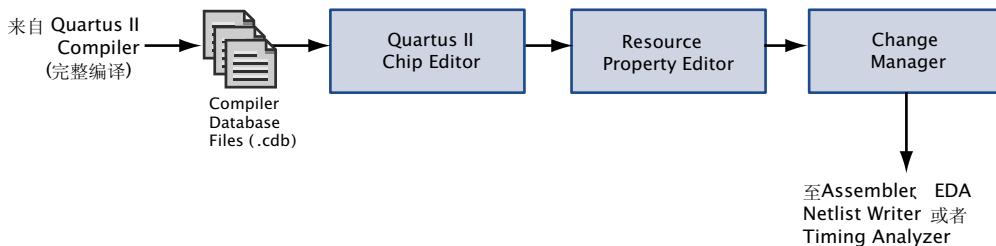
简介	192
使用 Chip Editor 识别延时与关键路径	193
在 Chip Editor 中编辑基元	194
使用 Resource Property Editor 修改资源属性	194
使用 Change Manager 查看和管理更改	196
验证 ECO 更改的效果	198

13

简介

Quartus® II 软件允许在完整编译之后对设计做小的更改，可参考为工程更改纪录 (ECO)。可直接对设计数据库进行 ECO 更改，而不是更改源代码或 Quartus II Settings 和 Configuration 文件 (.qsf)。对设计数据库做 ECO 更改可避免为实施一个更改而运行完整编译。图 1 为工程更改管理设计流程。

图 1. 工程更改管理设计流程



以下步骤概述了 Quartus II 软件中工程更改管理的设计流程。

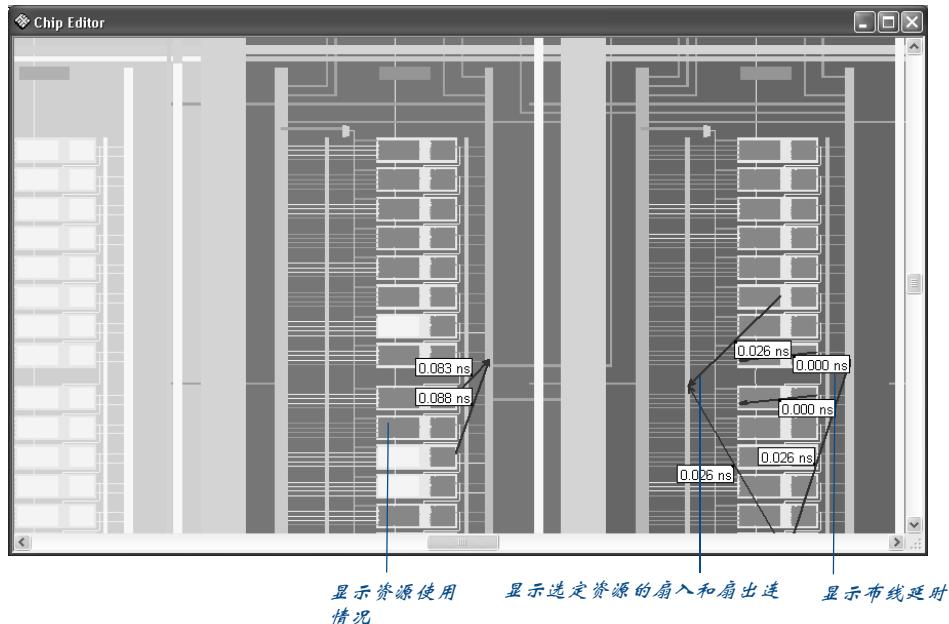
1. 完整编译之后，使用 Chip Editor 查看设计布局布线详细信息，并确定要更改的资源。
2. 建立、移动以及删除 Chip Editor 中的基元。
3. 使用 Resource Property Editor 编辑资源内部属性、编辑或删除连接。
4. 重复步骤 2 和 3，直到完成全部更改。
5. 使用 **Check Resource Properties** 命令 (Edit 菜单) 检查资源更改的合法性。
6. 在 Change Manager 中查看更改的摘要和状态，并控制要对资源属性做什么更改，并保存。还可以添加备注，帮助引用每个更改。
7. 使用 **Check and Save All Netlist Changes** 命令 (Edit 菜单) 检查网表中所有其它资源更改的合法性。
8. 运行 Assembler，生成新的编程文件，或再次运行 EDA Netlist Write，生成新网表。如果要验证时序更改，可以运行 Timing Analyzer。如果要验证仿真是否正确，可以运行 Simulator。

使用 Chip Editor 识别延时与关键路径



可以使用 Chip Editor 查看布局布线的详细信息。Chip Editor 可以显示 Quartus II Timing Closure 平面布局图中不显示的设计布局布线的其它详细信息。它显示完整的布线信息，显示每个器件资源之间的所有可能和使用的布线路径。参见图 2。

图 2. Chip Editor



Chip Editor 显示器件的所有资源，例如，互连和布线连线、逻辑阵列块 (LAB)、RAM 块、DSP 块、I/O、行、列以及块与互连和其它布线连线之间的接口。

可以通过放大或缩小、选择要显示的特定路径以及显示独立的鹰眼视图窗口 (可放大显示器件视图，在普通窗口中指定一个矩形“view-port”，来放大查看视图的局部)，控制 Chip Editor 显示信息的详尽程度。还可以设置控制不同资源显示、扇入和扇出、关键路径、信号延时估计和 Fitter 布局的选项。然后，可以使用此信息确定可能要在 Resource Property Editor 中编辑的属性和设置。可以在 Chip Editor 中选择一个或多个资源，并选择 Locate in Resource Property Editor (右键弹出菜单)，打开 Resource Property Editor，编辑该资源。有关详细信息，请参阅 第 194 页“使用 Resource Property Editor 修改资源属性”。

如果选择了多个单元，Selected Elements Window 命令（右键弹出菜单）允许在 Resource Property Editor 或其他编辑器中查找，根据需要，从选择中删除单元。



有关信息

请参考

工程更改管理和使用 Chip Editor

Altera 网站上 *Quartus II Handbook* 第 1 卷
“Engineering Change Management”

使用 Chip Editor

Altera 网站上 *Quartus II Handbook* 第 3 卷
“Design Analysis and Engineering Change
Management with the Chip Editor”

Quartus II Help 中的 “Overview:Using the Chip
Editor” 和 “Making Post-Compilation Changes
Introduction”

在 Chip Editor 中编辑基元

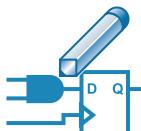


Chip Editor 可以建立新的基元或者将现有基元移动到其他位置。还可以删除基元。这些更改反映在 Change Manager 中。

可以通过在 Chip Editor 窗口中选定资源位置，选择 Create Atom（右键弹出菜单），指定基元的新名称来建立新的基元。可以使用 Locate in Resource Property 命令（右键弹出菜单）修改新基元的属性和连接。更详细的信息，请参阅下一节 “使用 Resource Property Editor 修改资源属性”。

如果要将一个基元移动到新位置，可以选定该基元，将其拖放至新位置。如果要删除一个基元，可以使用 Delete 命令（右键弹出菜单）。Check and Save All Netlist Changes 命令 (Edit 菜单) 保存对基元所作的所有改动。

使用 Resource Property Editor 修改资源属性

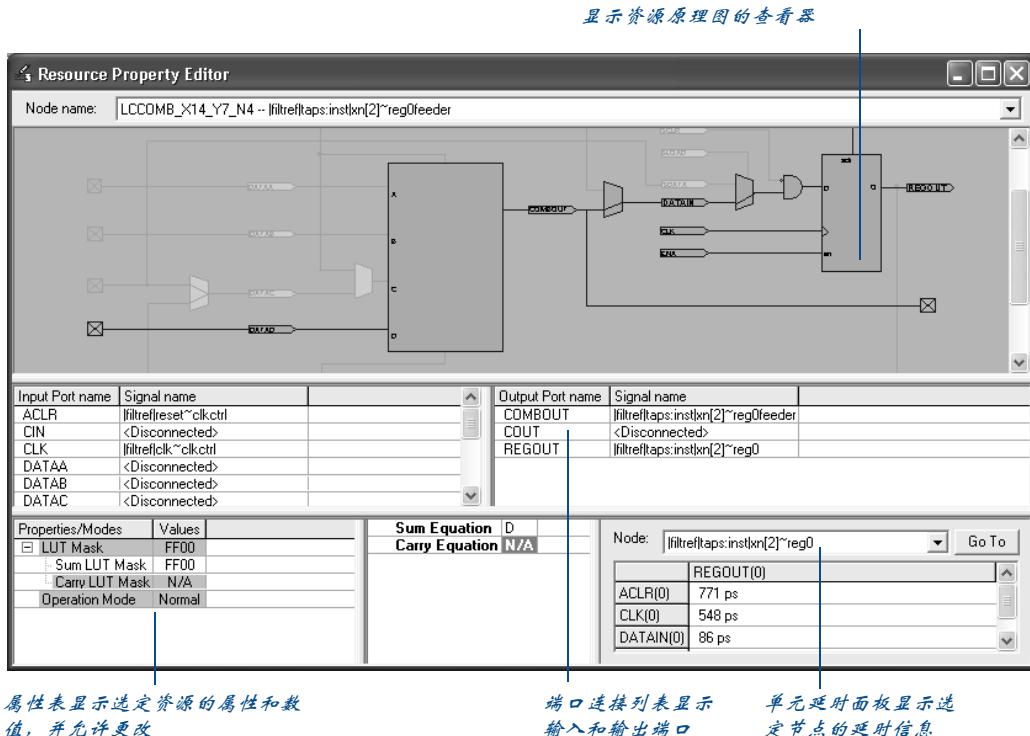


Resource Property Editor 用于对逻辑单元、I/O 单元或 PLL 资源的属性和参数进行编译后编辑，以及编辑或删除个别节点的连接。可以使用工具栏按钮在资源中前后浏览。还可以同时选择和更改多个资源。此外，当用鼠标指向一个资源端口时，Resource Property Editor 高亮显示该端口的扇入和扇出。

Resource Property Editor 含有一个显示正在被修改的资源示意图的查看器，一个列出所有输入和输出端口及其连接信号的端口连接列表，一个显示资源

可用属性和参数的属性列表。如果看不见端口连接或属性列表，可使用 View Port Connections 命令和 View Properties 命令（View 菜单）将其显示出来。图 3 是 Resource Property Editor。

图 3. Resource Property Editor



可以在示意图、端口连接表或属性表中对资源进行更改。如果在端口连接表或属性表中做更改，该更改将自动反映在示意图中。还可以查看等式和单元延时信息。

Resource Property Editor 还允许在示意图或端口连接表中选定节点，并选择 Edit Connection (右键弹出菜单) 来为该连接指定新信号。若要删除该连接，可选定节点并选择 Remove Connection (右键弹出菜单)。在端口连接列表中，可通过选择 Create 或 Remove (右键弹出菜单) 建立或删除输出端口。在示意图中，使用 Remove > Fan-Outs 对话框 (右键弹出菜单)，选定一个节点，然后指定一个或多个要删除的扇出。

一旦做了更改，就可以使用 **Check Resource Properties** 命令 (Edit 菜单) 对资源进行简单的设计规则检查。还可以在 Change Manager 中查看所做更改的摘要。有关详细信息，请参考下一节“[使用 Change Manager 查看和管理更改](#)”。



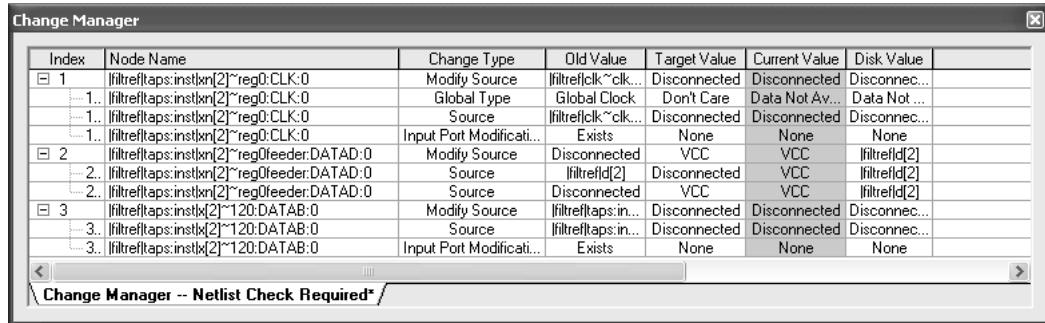
有关信息	请参考
工程更改管理和使用 Resource Property Editor	Altera 网站上 <i>Quartus II Handbook</i> 第1卷 “Engineering Change Management”
使用 Resource Property Editor	Altera 网站上 <i>Quartus II Handbook</i> 第3卷 “Design Analysis and Engineering Change Management with the Chip Editor”
	Quartus II Help 中的“Overview:Using the Resource Property Editor”和“Making Post-Compilation Changes Introduction”

使用 Change Manager 查看和管理更改



Change Manager 窗口列出所做的所有 ECO 更改。它允许在列表中选择每个 ECO 更改，并指定是否要应用或删除更改。它还允许添加备注，以便参考。可通过选取 **Utility Windows > Change Manager** (View 菜单) 打开 Change Manager。请参阅图 4。

图 4. Change Manager



The screenshot shows the Change Manager window with the following data:

Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
1..	lfilterlaps:inst[kn[2]]~reg0:CLK:0	Modify Source	lfilterclk~clk...	Disconnected	Disconnected	Disconnect...
1..	lfilterlaps:inst[kn[2]]~reg0:CLK:0	Global Type	Global Clock	Don't Care	Data Not Av...	Data Not ...
1..	lfilterlaps:inst[kn[2]]~reg0:CLK:0	Source	lfilterclk~clk...	Disconnected	Disconnected	Disconnect...
1..	lfilterlaps:inst[kn[2]]~reg0:CLK:0	Input Port Modificati...	Exists	None	None	None
2..	lfilterlaps:inst[kn[2]]~reg0feeder:DATAD:0	Modify Source	Disconnected	VCC	VCC	lfilterfd[2]
2..	lfilterlaps:inst[kn[2]]~reg0feeder:DATAD:0	Source	lfilterfd[2]	Disconnected	VCC	lfilterfd[2]
2..	lfilterlaps:inst[kn[2]]~reg0feeder:DATAD:0	Source	Disconnected	VCC	VCC	lfilterfd[2]
3..	lfilterlaps:inst[k2]~120:DATAB:0	Modify Source	lfilterlaps:in...	Disconnected	Disconnected	Disconnect...
3..	lfilterlaps:inst[k2]~120:DATAB:0	Source	lfilterlaps:in...	Disconnected	Disconnected	Disconnect...
3..	lfilterlaps:inst[k2]~120:DATAB:0	Input Port Modificati...	Exists	None	None	None

At the bottom of the window, there is a status bar with the text "Change Manager -- Netlist Check Required*".

Change Manager 的日志视图显示每个 ECO 更改的以下信息：

- ◇ Index (目录)
- ◇ Node Name(节点名称)
- ◇ Change Type(更改类型)
- ◇ Old Value(旧值)
- ◇ Target Value(目标值)
- ◇ Current Value(当前值)
- ◇ Disk Value(硬盘值)
- ◇ Comments(备注, ECO 更改的备注)

Current Value 栏中的绿色阴影表明更改已经应用到当前值中。Disk Value 栏中的兰色阴影表明更改已经成功保存在硬盘上。

提交所需更改之后，应选择 Check and Save All Netlist Changes (右键弹出菜单) 以检查网表中所有其它资源更改的合法性。然后，可以使用右键弹出菜单中的命令对列表中 ECO 更改执行以下操作：

- ◇ 将目标值应用于先前和当前更改
- ◇ 将旧值恢复至当前和后续更改
- ◇ 删 除当前和后续更改
- ◇ 导出先前和当前更改
- ◇ 导出所有更改

当选取命令之一进行导出时，可以将导出数据存储为 Tcl 脚本文件 (.tcl)，该文件顺序存放 Chip Editor Tcl 命令，在 Change Manager 日志丢失或损坏时，使用该文件可以回到 Quartus II 软件中，重新产生一组更改。还可以保存一个 Comma-Separated Values 文件 (.csv) 或 Text 文件 (.txt)，这些文件含有用于数据记录的表格。

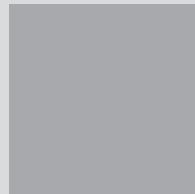
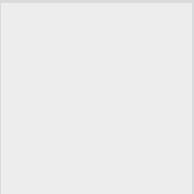
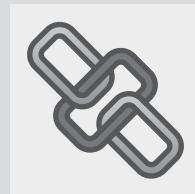
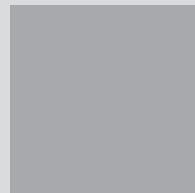
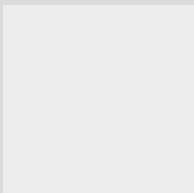
有关信息	请参考
工程更改管理和使用 Change Manager	Altera 网站上 <i>Quartus II Handbook</i> 第 1 卷 “Engineering Change Management”
使用 Change Manager	Altera 网站上 <i>Quartus II Handbook</i> 第 3 卷 “Design Analysis and Engineering Change Management with the Chip Editor”
	Quartus II Help 中的 “Overview:Using the Change Manager” 和 “Making Post-Compilation Changes Introduction”

验证 ECO 更改的效果

做过 ECO 更改后，应运行 Compiler 的 Assembler 模块以建立新的 POF。如果要再次运行 EDA Netlist Writer 来生成新的网表，或再次运行 Timing Analyzer 和 Simulator 来验证相应时序改进中的更改结果，可以使用 Compiler Tool 窗口，或在命令行或脚本中使用 `quartus_asm` 或 `quartus_eda` 以及 `quartus_tan` 可执行文件，单独运行每个模块。但是，执行完整编译将改变 ECO 更改的数值。

第十四章

形式验证



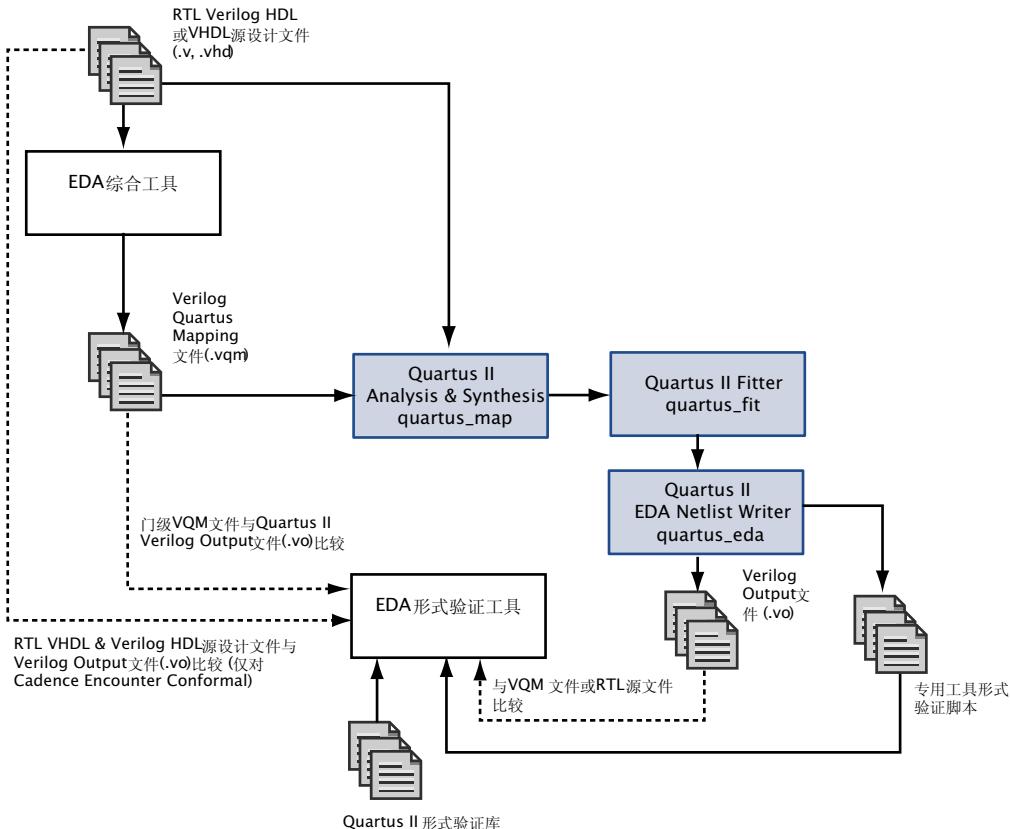
第 14 章 内容:	
简介	200
使用 EDA 形式验证工具	201
指定其他设置	203

14

简介

Quartus II 软件使用形式验证 EDA 工具来验证源设计文件和 Quartus II 输出文件之间的逻辑等价性。图 1 为形式验证流程。

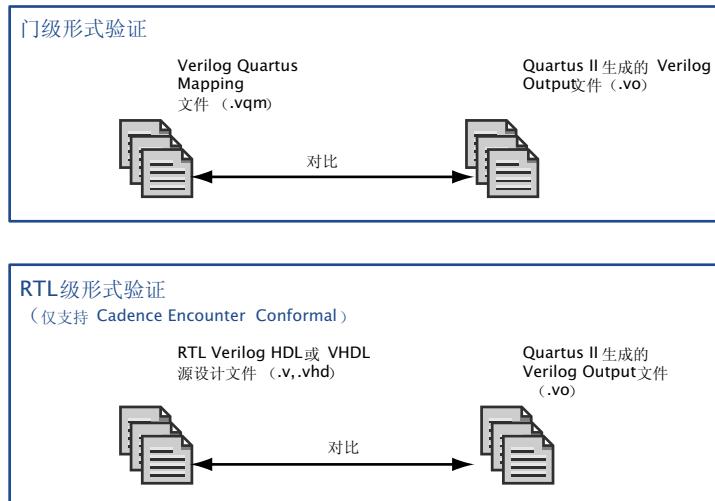
图 1. 形式验证流程



Quartus II 软件支持的形式验证类型为等价性检查，通过使用数学技巧而不是使用测试向量进行仿真，来比较源设计文件和修改后设计文件的功能等价性。等价性检查极大的缩短了验证设计所花费的时间。Quartus II 软件可以验证由 EDA 综合工具生成的综合门级 Verilog Quartus Mapping 文件 (.vqm) 和 Quartus II 软件生成的 Verilog Output 文件之间的逻辑等价性。对于

Cadence Encounter Conformal 软件, Quartus II 同样可以验证 RTL VHDL 设计文件 (.vhd) 或 Verilog HDL 设计文件 (.v) 和 Quartus II 生成的 Verilog Output 文件的逻辑等价性。图 2 所示为形式验证所比较的文件类型。

图 2. 形式验证比较的文件类型



使用 EDA 形式验证工具



可以在 Quartus II 设计中使用 EDA 形式验证工具进行形式验证。形式验证软件在综合和适配期间, 比较 Quartus II 软件是否能够正确解译 VQM 文件以及源 VHDL 或 Verilog HDL 设计文件中的逻辑。

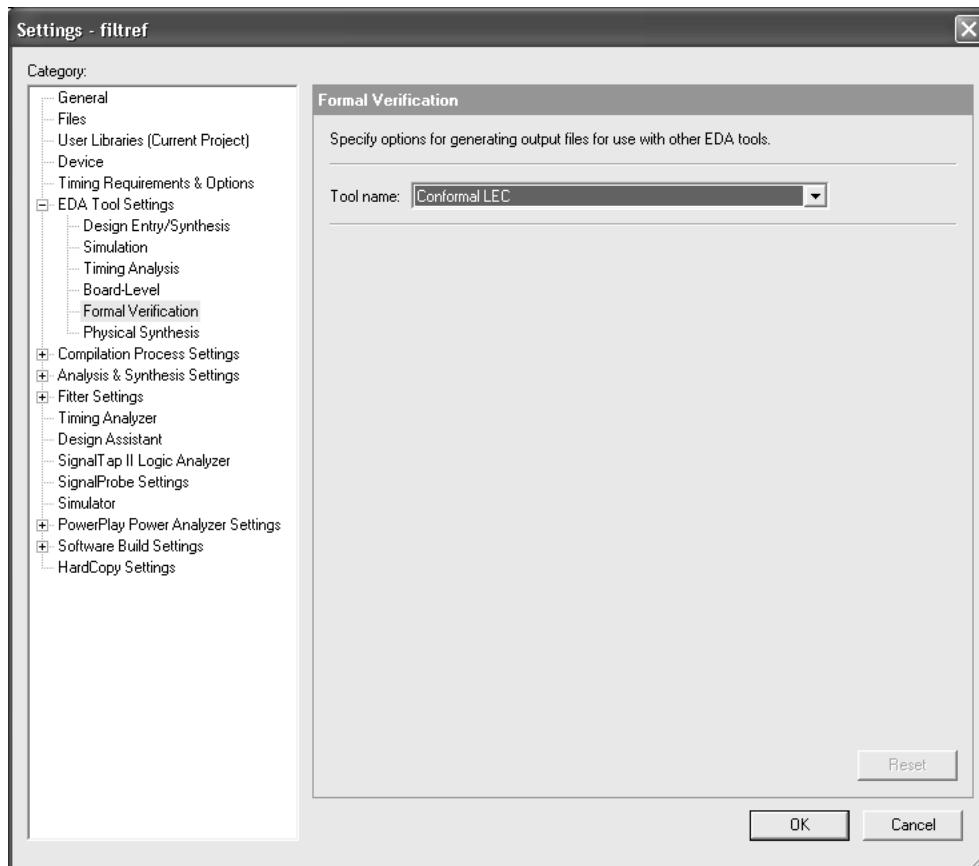
表 1 列出了由 Quartus II 软件支持的 EDA 形式验证工具。

表 1. Quartus II 支持的 EDA 形式验证工具

形式验证工具名称	Verilog Quartus Mapping 文件 (.vqm) 支持	RTL Verilog HDL 或 VHDL Design 文件支持
Cadence Encounter Conformal	✓	✓
Synopsys Formality	✓	

在 Settings 对话框 (Assignments 菜单) EDA Tool Settings 的 Formal Verification 页面中，可以指定能够显示的 EDA 形式验证工具。参见图 3。

图 3. Settings 对话框的 Formal Verification 页面



有关信息

使用 Cadence Encounter Conformal 软件

请参考

Altera 网站上 *Quartus II Handbook* 第 3 卷
“Cadence Encounter Conformal Support”

Quartus II Help 中的 “Overview: Using the
Encounter Conformal Software with the Quartus
II Software”

使用 Synopsis Formality 软件

Altera 网站上 *Quartus II Handbook* 第 3 卷
“Synopsis Formality Support”

指定其他设置

编译工程生成文件，用于形式验证工具时，Altera 强烈建议您关掉以下选项：

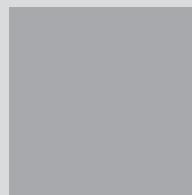
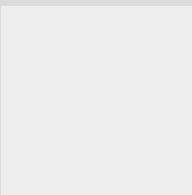
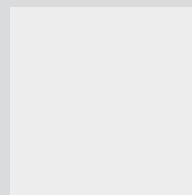
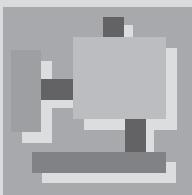
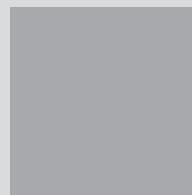
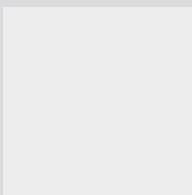
- ◇ 必须关掉 **Settings** 对话框 (**Assignments** 菜单) **Analysis & Synthesis Settings** 下 **Synthesis Netlist Optimizations** 页面的 **Perform gate-level register retiming** 选项。
- ◇ 必须关掉 **Settings** 对话框 **Fitter Settings** 下 **Physical Synthesis Optimizations** 页面的 **Perform register retiming** 选项。

Altera 之所以建议关掉这些选项是因为它们通常会导致关键路径上寄存器的移动与合并，影响形式验证工具用作比较点的逻辑寄存器。

有关信息	请参考
Cadence Encounter Conformal 软件的其他建议和选项	Altera 网站上 <i>Quartus II Handbook</i> 第 3 卷“Cadence Encounter Conformal Support”
使用 Synopsis Formality 软件的其他建议和选项	Quartus II Help 中的 “Overview: Using the Encounter Conformal Software with the Quartus II Software”

第十五章

系统级设计



第 15 章 内容:	
简介	206
使用 SOPC Builder 建立 SOPC 设计	208
使用 DSP Builder 建立 DSP 设计	210

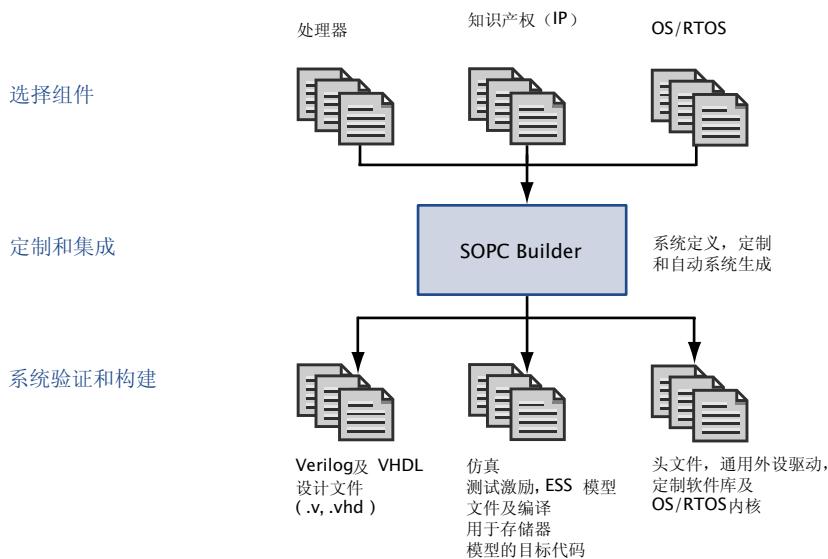
15

简介

Quartus®II 软件支持 SOPC Builder 和 DSP Builder 系统级设计流程。系统级设计流程使工程师能够以更高级的抽象概念快速设计、评估可编程芯片系统 (SOPC) 体系结构和设计。

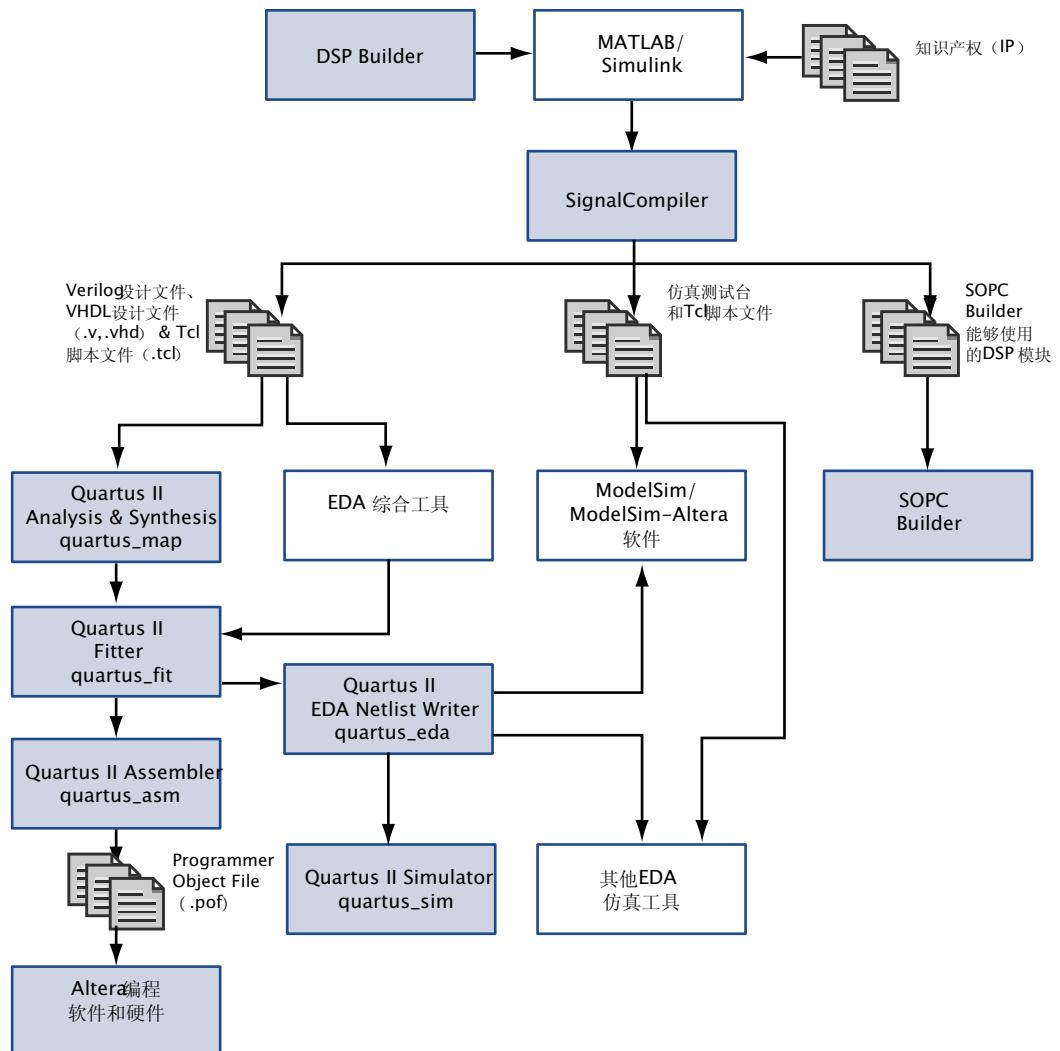
SOPC Builder 是自动的系统开发工具，可以极大简化建立高性能 SOPC 设计的任务。此工具能够在 Quartus II 软件中使 SOPC 开发的系统定义和集成阶段完全实现自动化。SOPC Builder 允许选择系统组件、定义和定制系统，并在集成之前生成和验证系统。图 1 显示了 SOPC Builder 设计流程。

图 1. SOPC Builder 设计流程

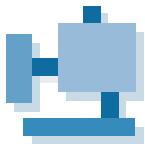


通过使用 VHDL 综合和仿真工具以及 Quartus II 软件结合 MathWorks MATLAB 和 Simulink 系统级设计工具的算法开发、仿真以及验证功能，Altera® DSP Builder 集成了高级算法和 HDL 开发工具。207 页上的图 2 显示了 DSP Builder 的设计流程。

图 2. DSP Builder 设计流程



使用 SOPC Builder 建立 SOPC 设计



SOPC Builder 包含在 Quartus II 软件中，它为建立 SOPC 设计提供标准化的图形环境，SOPC 由 CPU、存储器接口、标准外设和用户自定义的外设等组件组成。SOPC Builder 允许选择和自定义系统模块的各个组件和接口。SOPC Builder 将这些组件组合起来，生成对这些组件进行例化的单个系统模块，并自动生成必要的总线逻辑，将这些组件连接起来。

SOPC Builder 库包括以下组件：

- ◇ 处理器
- ◇ 知识产权 (IP) 和外设
- ◇ 存储器接口
- ◇ 通信外设
- ◇ 总线和接口，包括 AvalonTM 接口
- ◇ 数字信号处理 (DSP) 内核
- ◇ 软件
- ◇ 头文件
- ◇ 通用 C 驱动程序
- ◇ 操作系统 (OS) 内核

可以使用 SOPC Builder 构建包括 CPU、存储器接口和 I/O 外设在内的嵌入式微处理器系统；但是，也可以生成不包括 CPU 的数据流系统。它允许指定具有多个主机和从机的系统拓扑结构。SOPC Builder 还可以导入或提供用户定义逻辑模块的接口，该模块作为定制外设连接到系统上。

建立系统

在 SOPC Builder 中构建系统时，可以选择用户自定义模块或模块集组件库中提供的模块。

SOPC Builder 可以导入或提供用户定义逻辑块的接口。SOPC Builder 系统与用户定义逻辑配合使用时具有以下四种机制：简单 PIO 连接、系统模块内例化、外部逻辑总线接口以及发布本地 SOPC Builder 组件。

SOPC Builder 提供用于下载的库组件（模块），包括 NIOS[®] II 等处理器、UART、定时器、PIO、Avalon 三态桥接器、多个简单存储器接口和 OS/RTOS 内核。此外，还可以从 MegaCore[®] 功能列表中进行选择，包括支持 OpenCore[®] Plus 硬件评估的功能等。

可以使用 SOPC Builder 的 **System Contents** 页面定义系统。可以在模块集中选择库组件，在模块表中显示添加的组件。

可以使用 **System Contents** 页面中的模块表或单独向导中的信息定义以下组件选项：

- ◇ 系统组件和接口
- ◇ 主机和从机连接
- ◇ 系统地址映射
- ◇ 系统 IRQ 分配
- ◇ 共享从机的仲裁优先级
- ◇ 多主机和从机时钟域

生成系统

SOPC Builder 中的每个工程包含系统说明文件 (PTF 文件)，它包含在 SOPC Builder 中输入的所有设置、选项和参数。此外，每个模块具有相应的 PTF 文件。在系统生成期间，SOPC Builder 使用这些文件为系统生成源代码、软件组件和仿真文件。

完成系统定义之后，可以使用 SOPC Builder 的 **System Generation** 页面生成系统。

SOPC Builder 软件自动生成所有必要逻辑，用以将处理器、外设、存储器、总线、仲裁器、IP 功能以及系统外逻辑和存储器接口集成在一起，并建立将组件捆绑在一起的 HDL 源代码。

SOPC Builder 还可以建立软件开发工具包 (SDK) 软件组件，例如，头文件、通用外围设备驱动程序、定制软件库和 OS/real-time 操作系统 (RTOS 内核) 等，在生成系统时提供完整的设计环境。

对于仿真，SOPC Builder 建立 Mentor Graphics®ModelSim® 仿真目录，它包括 ModelSim 工程文件、所有存储器组件的仿真数据文件、提供设置信息的宏文件、别名和总线接口波形初试装置。它还建立仿真测试激励，用于例化系统模块、驱动时钟和复位输入，并可以例化和连接仿真模型。

还可以生成 Tcl 脚本，用于在 Quartus II 软件中设置系统编译所需的所有文件。



有关信息

使用 SOPC Builder

请参阅

Altera 网站上 *Quartus II Handbook* 第 4 卷
“SOPC Builder”

Altera 网站上 *Application Note333 (Developing Peripherals for SOPC Builder)*

Quartus II Help 中的 “Overview:Using SOPC Builder”

使用 DSP Builder 建立 DSP 设计

DSP Builder 在算法友好开发环境中建立 DSP 设计的硬件表征，缩短了 DSP 设计周期。DSP Builder 允许系统、算法和硬件设计人员共享公共开发平台。DSP Builder 是 Altera 提供的可选软件包，它也包含在 DSP 开发套件中。

DSP Builder 采用 SignalTap® II 模块或环路硬件 (HIL) 模块为系统级调试提供支持。可以全部通过 MATLAB/Simulink 界面实现设计综合、编译和下载，然后进行调试。Simulink 模型的环路硬件模块采用物理 FPGA 电路板对 Quartus® II 软件设计进行协同仿真，设计的一部分由该电路板实现。通过建立和编译 Quartus II 工程来定义 FPGA 的组成和功能。一个简单的 JTAG 接口将 Simulink 和 FPGA 电路板连接在一起。

例化功能

可以将现有的 MATLAB 功能和 Simulink 块与 Altera DSP Builder 块和 MegaCore 功能组合在一起（其中包括支持 OpenCore Plus 硬件评估功能），将系统级设计和实现与 DSP 算法开发相连。

要在设计中使用支持 OpenCore Plus 功能的 MegaCore 功能，必须在运行 MATLAB/Simulink 环境之前下载这些功能。

生成仿真文件

在 Simulink 软件中验证设计之后，可以使用 DSP Builder SignalCompiler 模块生成在 EDA 仿真工具中进行设计仿真的文件。

SignalCompiler 模块将 DSP Builder Simulink 模型转换为 VHDL 或 Verilog 模型，生成 Verilog HDL 或 VHDL 测试激励文件，导入 Simulink 输入激励。可在 ModelSim 软件中使用 Tcl 脚本进行自动仿真，或在另一个 EDA 仿真工具中使用 Verilog HDL 或 VHDL 测试激励文件进行仿真。

生成综合文件

DSP Builder 提供两种综合和编译流程：自动和手动。可以在 Quartus II、Mentor Graphics LeonardoSpectrum 或 Synplicity Synplify 软件中采用 DSP Builder SignalCompiler 模块生成的 Tcl 脚本对设计进行综合。如果 DSP Builder 设计是顶层设计，可以使用自动或手动综合流程。如果 DSP Builder 设计不是顶层设计，必须使用手动综合流程。

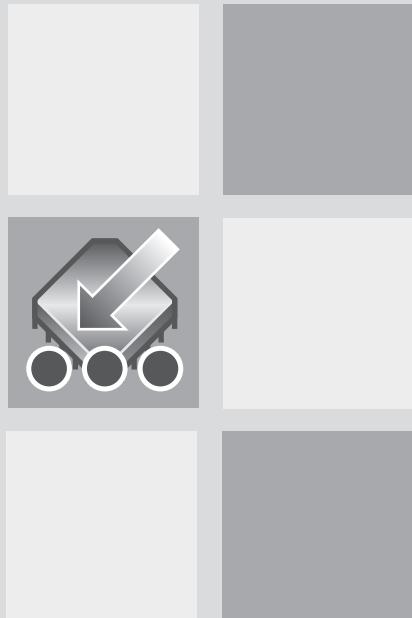
可以使用自动流程在 MATLAB/Simulink 设计环境中控制整个综合和编译流程。SignalCompiler 块可以建立 VHDL 设计文件和 Tcl 脚本，在 Quartus II、LeonardoSpectrum 或 Synplify 软件中进行综合，在 Quartus II 软件中编译设计，还可以选择下载设计到 DSP 开发板上。可以在 Simulink 软件内指定用于设计的综合工具。

在手动流程中，SignalCompiler 模块生成 VHDL 设计文件和 Tcl 脚本，然后，在 EDA 综合工具或 Quartus II 软件中进行手动综合，Quartus II 软件允许指定您自己的综合或编译设置。生成输出文件时，SignalCompiler 模块将每个 Altera DSP Builder 块映射至 VHDL 库。将 MegaCore 功能作为 black-box 处理。

有关信息	请参阅
使用 DSP Builder	Altera 网站上的 <i>DSP Builder User Guide</i>

第十六章

软件开发



第 16 章 内容:

简介	214
在 Quartus II 软件中使用 Software Builder	214
进行软件构建设置	215
生成软件输出文件	215

16

简介



Quartus® II Software Builder 集成编程工具可以将软件源文件转换为用于配置 Excalibur™ 器件的闪存编程文件或被动编程文件，或包含 Excalibur 器件嵌入式处理器带区的存储器初始化数据文件。可以使用 Software Builder 处理 Excalibur 设计的软件源文件，包括使用 SOPC Builder 和 DSP Builder 系统级设计工具建立的设计。

在 Quartus II 软件中使用 Software Builder

Software Builder 使用 ADS Standard Tools 或 GNUPro for ARM® 软件工具集处理 Quartus II Text Editor 或其它 Assembly、C/C++ 语言开发工具建立的软件源文件。可以使用 Software Builder 处理以下软件源文件：

- ◇ 汇编文件 (.s, .asm)
- ◇ C/C++ Include 文件 (.h)
- ◇ C 源文件 (.c)
- ◇ C++ 源文件 (.cpp)
- ◇ 库文件 (.a)

Software Builder 能够在最少帮助下在软件源文件上构建软件，并允许对特定设计进行定制处理。一旦指定软件构建设置，就可以使用 **Start Software Build** 命令 (Processing 菜单) 运行 Software Builder。

还可以使用 Software Builder 在软件构建期间或之后运行命令行命令，在 Quartus II 软件内为 Excalibur 器件运行程序或进程。



使用 quartus_swb 可执行文件

还可以在命令提示符下或在脚本中使用 **quartus_swb** 可执行文件单独运行 Software Builder。

若要获取有关 **quartus_swb** 可执行文件的帮助，请在命令提示符下键入以下命令之一：

```
quartus_swb -h ↵  
quartus_swb --help ↵  
quartus_swb --help=<topic name> ↵
```

进行软件构建设置

在执行软件构建之前，可以使用 **Settings** 对话框 (Assignments 菜单) 的 **Software Build Settings** 页面，指定软件构建设置。

可以使用 **Settings** 对话框指定以下设置：

- ◇ 工程、工具集目录、体系结构和软件工具集、字节顺序、输出文件名、定制构建和后构建命令行命令、编程文件生成选项的软件构建设置名称。
- ◇ **C/C++ Compiler 选项**：优化级别、预处理器定义、包含目录以及命令行命令。
- ◇ **Assembly 选项**：预处理器定义、其他包含目录和命令行命令。
- ◇ **Linker 选项**：对象文件、库文件、库目录、链接类型和命令行命令。

生成软件输出文件

可以通过在 Quartus II 软件中构建软件来处理设计，生成包含存储器初始化数据、被动编程文件和闪存编程文件的文件。还可以使用 **makeprogfile** 应用程序（也由 Quartus II 软件在软件构建期间使用）和独立的 **MegaWizard® Plug-In Manager** 在 Quartus II 软件之外生成被动编程文件和闪存编程文件。

有关使用 **makeprogfile** 应用程序的详细信息，请在命令提示符下键入 **makeprogfile -h ↵**。



使用独立的 MegaWizard Plug-In Manager

可以在命令提示符下键入以下命令，实现在 Quartus II 软件之外使用 **MegaWizard Plug-In Manager**：

```
qmegawiz ↵
```

每当使用 Software Builder 生成闪存编程文件，或使用 Compiler 或 Software Builder 生成被动编程文件时，Software Builder 将自动建立仿真器初始化文件。仿真器初始化文件为 Excalibur 嵌入式处理器带区内存储器区域的每个地址指定初始化数据。**表 1** 列出了 Software Builder 生成的仿真器初始化文件。

表1. 仿真器初始化文件

文件名称	文件内容
memory_REGS	寄存器初始化数据
memory_SRAM0	SRAM0 初始化数据
memory_SRAM1	SRAM1 初始化数据
memory_DPRAM0	DPRAM0 初始化数据
memory_DPRAM1	DPRAM1 初始化数据

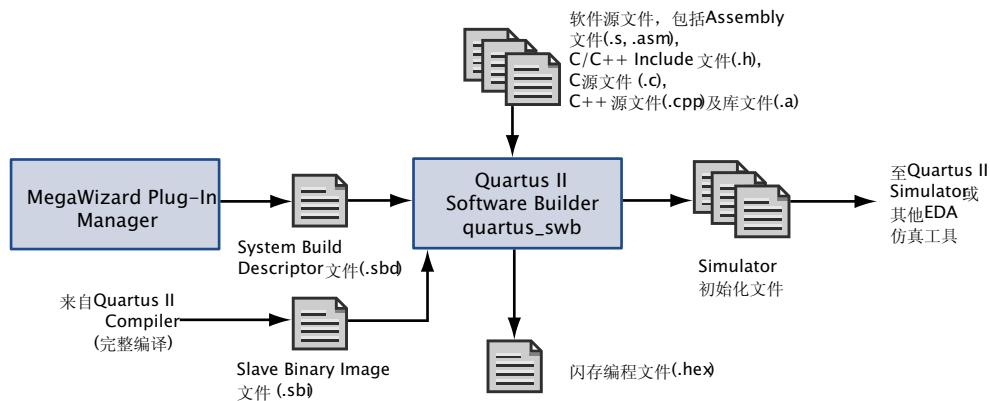
生成闪存编程文件

闪存编程文件是 Hexadecimal(Intel 格式) 文件 (.hex)，它对闪存进行编程，Excalibur 器件从此闪存中加载配置和存储器初始化数据。以下步骤描述使用 Software Builder 建立闪存编程文件的基本流程：

1. 建立软件源文件并将其添加至工程中。
2. 运行 **ARM-based Excalibur MegaWizard Plug-In**，生成 System Build Descriptor 文件 (.sbd)。
3. 若要闪存编程文件包含 Excalibur 器件可编程逻辑器件 (PLD) 部分的配置数据，请编译设计，生成 Slave Binary Image 文件 (.sbi)。
4. 指定工具集目录和软件构建设置。若要生成闪存编程文件，必须指定输出文件类型和文件名称，开启 **Flash memory configuration**，并且如果正在使用 Slave Binary Image 文件，在 **Settings** 对话框 (Assignments 菜单) **Software Build Settings** 页面中指定可选的 Slave Binary Image 文件。
5. 启动软件构建。

图1是使用 Software Builder 建立闪存编程文件的流程。

图1. 闪存编程文件流程



Software Builder 执行以下步骤生成闪存编程文件：

1. 汇编器、C/C++ 编译器、连接器和代码转换器将软件源文件转换为 HEX 文件，它包含 Excalibur 器件的 Excalibur 嵌入式处理器带区存储器初始化数据。
2. 由 HEX 文件、System Build Descriptor 文件和 Slave Binary Image 文件建立启动数据对象文件。
3. 连接器将启动数据文件与二进制启动载入文件相连，建立 Executable 和 Linkable Format 文件 (.elf)。
4. 代码转换器将 Executable 和 Linkable Format 文件转换为名为 *<projector name>_flash.hex* 的闪存编程文件。

然后可以使用 `exc_flash_programmer` 应用程序通过 Expansion Bus Interface 0 (EBIO) 将闪存编程文件中的信息编程至 Excalibur 器件的闪存。

生成被动编程文件

被动编程文件用于使用 Passive Parallel Asynchronous (PPA)、Passive Parallel Synchronous (PPS)、Passive Serial (PS) 配置方案对 Excalibur 器件进行配置。可以使用 Software Builder、`makeprogfile` 应用程序或 Compiler 生成以下被动编程文件：

- ◇ Hexadecimal (Intel 格式) Output 文件 (.hexout)
- ◇ Programmer Object Files (.pof)

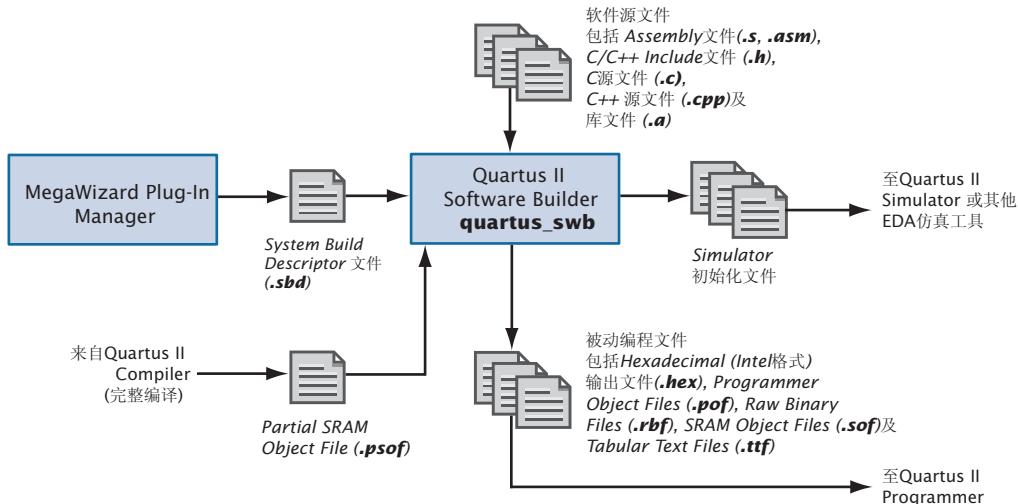
- ◇ Raw Binary Files(.rbf)
- ◇ SRAM Object Files(.sof)
- ◇ Tabular Text Files(.ttf)

以下步骤描述使用 Software Builder 建立被动编程文件的基本流程：

1. 建立软件源文件并将其添加至工程中。
2. 运行 ARM-based Excalibur MegaWizard Plug-In，生成 System Build Descriptor 文件。
3. 编译设计，生成可编程逻辑 Partial SRAM Object File(.psof)。
4. 指定软件工具集目录和软件构建设置。如果要生成闪存编程文件，必须指定输出文件类型和文件名称，开启 Passive configuration，并在 Settings 对话框 (Assignments 菜单) Software Build Settings 页面中指定 PSOF。
5. 启动 Software Builder。

图 2 是使用 Software Builder 建立被动编程文件的流程。

图 2. 被动编程文件流程



Software Builder 执行以下步骤，生成被动编程文件：

1. 汇编器、C/C++ 编译器、连接器和代码转换器将软件源文件转换为 HEX 文件，它包含 Excalibur 器件的 Excalibur 嵌入式处理器带区存储器初始化数据。
2. **makeprogfile** 应用程序处理 HEX 文件、System Build Descriptor 文件和 PSOF，建立一个或多个被动编程文件。

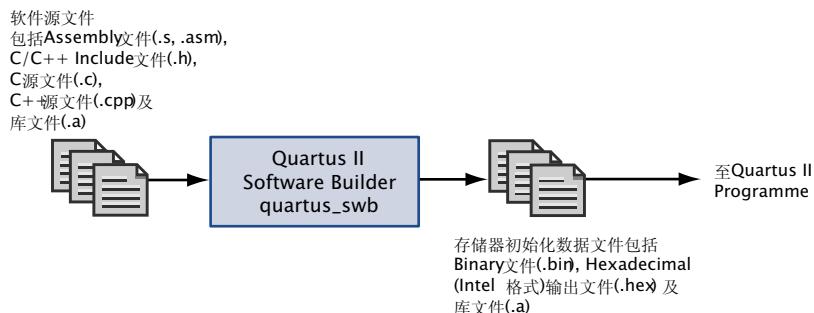
生成存储器初始化数据文件

Binary 文件 (.bin)、HEX 文件和 Library 文件 (.a) 包含 Excalibur 嵌入式处理器带区的存储器初始化数据。以下步骤描述使用 Software Builder 建立 BIN 文件、HEX 文件和 Library 文件的基本流程：

1. 建立软件源文件并将其添加至工程中。
2. 指定软件工具集目录和软件构建设置。使用 **Settings** 对话框 (Assignments 菜单) **Software Build Settings** 页面指定输出文件类型和文件名称。如果在 **Output file format** 列表中选择了 HEX 文件，并且不想生成闪存编程文件或被动编程文件，请选择 **Programming file generation** 下的 **None**。
3. 启动软件构建。

图 3 为使用 Software Builder 生成存储器初始化数据文件的流程。

图 3. 存储器初始化数据文件流程



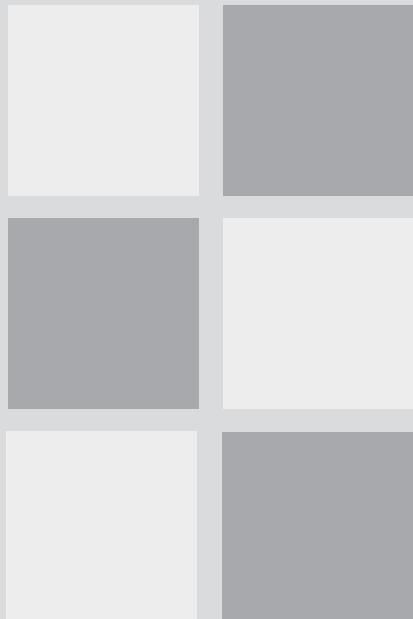
要生成存储器初始化文件，Software Builder 执行以下步骤：

1. 汇编器和 C/C++ 编译器由设计的软件源文件生成中间对象文件。
2. 如果正在生成 BIN 文件或 HEX 文件，则连接器连接对象文件并生成中间 ELF 文件，代码转换器将 ELF 文件转换为 BIN 文件或 HEX 文件。
3. 如果正在生成 Library 文件，则 Software Builder 使用 Software Builder Archiver 处理对象文件，使之进入 Library 文件中。

有关信息	请参阅
进行 Software 构建	Quartus II Help 中的“Overview:Using the Software Builder”
为 POF 和 SOF 生成被动编程文件和可选编程文件	Altera 网站上 <i>Application Note 299 (System Development Tools for Excalibur Devices)</i>
生成 BIN 文件、HEX 文件和 Library 文件并生成调试信息	Quartus II Help 中的“Generating Passive Programming Files”
	Quartus II Help 中的“Generating Binary Files, Hexadecimal (Intel-Format) Files”
	Quartus II Help 中的“Overview:Checking Software Source Files and Output Files”

第十七章

安装、许可和技术支持



第17章 内容:

安装 Quartus II 软件	222
许可 Quartus II 软件	222
获取技术支持	224

17

安装 Quartus II 软件

可以在以下平台上安装 Quartus® II 软件：

- ◇ 运行速度为 400 MHz 或更快的 Pentium PC，并采用以下操作系统之一：
 - Microsoft Windows NT 4.0 (SP4 或更新)
 - Microsoft Windows 2000
 - Microsoft Windows XP
- ◇ 速度为 400 MHz 或更快的 Pentium III 或 4 PC，运行以下 Linux 操作系统之一：
 - Red Hat Linux 7.3 或 8.0
 - Red Hat Linux Enterprise 3
 - 仅对 AMD64/EM64T 的 Red Hat Linux Enterprise 3.0 WS
- ◇ 运行 Solaris 8 或 9 版本或 Solaris 8 或 9 版本 (64 位) 的 Sun Ultra 工作站
- ◇ HP 9000 系列 700/800 工作站，运行 HP-UX 11.0 版，ACE 日期为 1999 年 11 月或更新

有关信息	请参阅
系统要求和安装说明	Altera 网站上 <i>Quartus II Installation & Licensing for PCs</i> 手册
有关磁盘空间和存储器的信息	Altera 网站上 <i>Quartus II Installation & Licensing for UNIX and Linux Workstations</i> 手册
有关新功能、设备支持、EDA 接口支持的最新信息	Quartus II <i>readme.txt</i> 文件
	Altera 网站上的 <i>Quartus II Software Release Notes</i>

许可 Quartus II 软件

要使用 Altera® 提供的软件，需要设置并获取 Altera 订购许可。一个 Altera 订购包括以下软件：

- ◇ Altera Quartus II 软件
- ◇ Mentor Graphics® ModelSim®-Altera 软件

Altera 提供多种类型的软件订购。表 1 所示为能够提供的各种许可和订购选项。

表 1. Altera 许可和订购选项

许可名称	说明
FIXEDPC	与 USB 软件狗或并口软件狗 (T 保护或“道尔芯片”) 绑定的单独 PC 许可
FLOATPC	向具有 PC 或 UNIX 许可服务器的 PC 用户提供浮动网络许可
FLOATNET	向使用 PC、Solaris 或 HP-UX 许可服务器的 PC、Solaris 和 HP-UX 用户提供浮动网络许可
FLOATLNX	向运行 Red Hat Linux 和使用 PC、UNIX 或 Linux 许可服务器的 PC 用户提供浮动网络许可
Quartus II 网络版	Quartus II 软件的免费入门级版本，支持选定器件。 可以从 Altera 网站 www.altera.com 获取 Quartus II 网络版软件

客户在购买选定开发工具包时将收到用于 PC 的 Quartus II 软件免费版本，并得到有关获取该软件许可的帮助。

以下步骤阐述取得软件许可的基本流程：

1. 启动 Quartus II 软件后，如果软件检测不到有效的 ASCII 文本许可文件 **license.dat**，系统将会提示以下内容：
 - **Enable 30-day evaluation period with no license file (no programming file support)**。该选项允许无需编程文件支持，为期 30 天的 Quartus II 软件评估。30 天的过渡期过后，必须从 Altera 网站 www.altera.com/licensing 的 Licensing 上获取有效的许可文件，继续本过程的剩余步骤。
 - **Perform automatic web license retrieval**。选择该项即自动从 Altera 网站请求有效许可文件。如果使用节点锁定 (FIXEDPC) 许可，Quartus II 软件能够从网站成功获取许可文件，则可以跳过该过程的剩余步骤。如果使用网络 (多用户) 许可，或者如果 Quartus II 软件不能获取许可文件，则系统会指导您通过许可过程。
 - **Specify valid license file**。如果您具有有效许可文件，但是没有指定该许可文件的位置，请选择该项来显示 Options 对话框 (Tools 菜单) 的 License Setup 页面。该页面将给出一个选项来指定有效许可

文件 (Specify valid license file) 或使用 LM_LICENSE_FILE 变量 (Use LM_LICENSE_FILE variable)。还可以在 Windows NT、Windows 2000 或 Windows XP 的系统控制面板中或 UNIX 和 Linux 工作站的 .cshrc 文件中指定许可文件或 LM_LICENSE_FILE 变量。如果选择该项，可跳过该过程的剩余步骤。

2. 如果在 Altera 网站的 Licensing 上中请求新的许可文件，请选择适当的许可类型。请参阅 223 页上的表 1。
3. 填写申请信息。
4. 通过电子邮件收到许可文件之后，将其保存至系统的一个目录中。
5. 根据需要，可以修改许可文件。
6. 为您的系统建立和配置 FLEXlm 许可管理器服务器。

有关信息	请参阅
有关 Quartus II 软件许可、修改许可文件和指定许可文件位置的详细信息	Altera 网站上的 <i>Quartus II Installation & Licensing for PCs</i> 手册
有关 Quartus II 许可的一般信息	Altera 网站上的 <i>Quartus II Installation & Licensing for UNIX and Linux Workstations</i> 手册
Altera 软件许可	Quartus II Help 中的“Overview:Obtaining a License File”和“Specifying a License File”
	Altera Altera 网站上的 <i>Application Note 340 (Altera Software Licensing)</i>

获取技术支持

获取技术支持最容易的方法是登录 mySupport 网站并注册一个 Altera.com 帐户。在购买时已经对您的 Quartus II 软件副本进行了登记；但是，要使用 mySupport 网站查看、提交服务请求，还必须注册一个 Altera.com 帐户。注册一个 Altera.com 帐户仅用于 mySupport 网站；但是，有了 Altera.com 帐户，可以更方便地使用 Altera 网站的许多其它功能，例如“Download Center”、“Licensing Center”“Altera Technical Training”的在线课程注册或“Buy On-Line-Altera eStore”功能等。

要注册一个 Altera.com 帐户用户名和密码，请执行以下步骤：

1. 进入 mySupport 网站：

- ✓ 若要在运行 Quartus II 软件时启动网络浏览器并连接到 mySupport 网站，请选择 **Altera on the Web > Quartus II Home Page** (Help 菜单)。

或者

- ✓ 将网络浏览器指向 mySupport 网站
www.altera.com/mysupport。

2. 按照 mySupport 网站上的说明，注册一个 Altera.com 帐户。

即使不是当前 Altera 订购用户，您仍可以注册一个 Altera.com 帐户。

有关其它技术支持资源的详细信息，请参阅 表 2。

表 2. Quartus II 技术支持资源

资源	说明
Altera 网站	www.altera.com Altera 网站提供 Altera 及其所有产品的信息。
支持中心	www.altera.com/support Altera 网站的“支持中心 (Support Center)”部分使您能够访问 mySupport 网站，提供“Altera Find Answers”。此外，它还提供软件和器件支持信息以及可以集成到设计中的设计实例。
mySupport 网站	www.altera.com/mysupport 或在 Quartus II 软件中选择 Altera on the Web > Quartus II Home Page (Help 菜单)。 mySupport 网站允许提交、查看和更新技术支持服务申请。
Altera Find Answers	www.altera.com/answers Altera Find Answers 使用自然语言处理技术 (NLP) 来分析您提问的含义和上下文，并提供答案。简单的搜索引擎只返回与关键词相对应的文档列表，而 Altera Find Answers 做出实际回答。
电话联系	(800) 800-EPLD (太平洋标准时间早上 7:00 至下午 5:00，周一至周五) 拨打热线电话需要使用您的 6 位数 Altera ID。 (408) 544-8767 (太平洋标准时间早上 7:00 至下午 5:00，周一至周五)

第十八章

文档和其他资源



第 18 章 内容:

获取在线帮助	228
使用 Quartus II 在线教程	229
其它 Quartus II 软件文档	230
其他 Altera 文献	230

18

获取在线帮助

Quartus® II 软件含有平台独立的帮助系统，为 Quartus II 软件提供全面的文档和 Quartus II 软件生成的特定消息的详细内容。可以使用以下方法之一查看帮助：

通过帮助主题列表关键词搜索 选择 Index (Help 菜单)，使用 Index 标签选项进行搜索。

通过帮助系统的全文进行搜索 选择 Search (Help 菜单)，使用 Search 标签选项进行搜索。

搜索帮助主题类的概要 选择 Contents (Help 菜单)，查看 Contents 标签选项。

在您的 Favorites 列表中添加主题 打开要将您偏好主题列表添加进去的 Quartus II 帮助主题。单击 Favorites 选项标签，然后单击 Add，将主题添加到您的 Favorites 列表中。

查看有关消息的帮助 选择要获得帮助的消息，并选择 Help (右键弹出菜单)。还可以选择 Messages (Help 菜单)，查看所有消息的滚动列表。

获取菜单命令或对话框的帮助 菜单命令高亮显示时，或者在活动对话框中按 F1，即可获取该项的上下文相关帮助。**查找术语的定义** 选择 Glossary (Help 菜单)，可以看到 Glossary 列表。



使用 Help Topics

要从 Contents 选项标签中打印 Help Topics，请选择要打印的 Help 文件夹或个别 Help 主题，然后选择 Print (右键弹出菜单) 或在工具栏上单击 Print 按钮。如果选择要打印的 Help 文件夹，可以选择打印该文件夹中的所有主题。还可以使用 Print 命令或 Print 按钮打印正在查看的任何 Help 主题。

要在打开的 Quartus II Help 主题中搜索关键字，请按 Ctrl+F，打开 Find 对话框，并键入搜索文本，然后单击 Find Next。



有关信息

请参阅

使用 Quartus II Help

Quartus II Help 中的 “Using Quartus II Help Effectively” 和 “Help Menu Commands”

Quartus II Installation & Licensing for PCs 手册和
Quartus II Installation & Licensing for UNIX and Linux Workstations 手册中的 “Using Quartus II Help”

使用 Quartus II 在线教程

在线教程介绍 Quartus II 设计软件的功能。它指导您如何简单快捷地建立、处理您自己的逻辑设计。Basic 和 optional 教程部分的模块化设计允许您在以下 Quartus II 软件部分中选择学习：

- ◇ Basic 教程部分指导您进行有限脉冲响应 (FIR) 滤波器设计实例（称为 fir_filter）的建立、时序分析、仿真和编程所需采取的步骤。
- ◇ Optional 教程部分关注从 MAX+PLUS® II 软件移植、使用 LogicLock™ 功能以及使用 Stratix™ 器件功能等主题。开始任一 Optional 教程之前无需完成 Basic 教程。

在成功安装 Quartus II 软件之后要启动 Quartus II 教程，请：

- ✓ 选择 Tutorial (Help 菜单)。

启动教程之后，Quartus II 窗口将做调整，使您可以同时看到 Tutorial 窗口和 Quartus II 软件。



有关使用 Quartus II 教程的详细信息

如果要完成 Basic 或 LogicLock 教程，必须安装对 Cyclone™ EP1C6 器件的支持。此外，如果要完成 Optional MAX+PLUS II 转换和 Stratix 教程部分，则必须安装对 MAX® EPM570 和 Stratix EP1S25 器件的支持。

此教程设计用于在线显示。然而，如果您要打印一个或多个教程部分，请单击每个部分开始处的 **Printing Options** 按钮，然后单击链接，打开相应的可打印版本。

其它 Quartus II 软件文档

表1 显示了有关 Quartus II 软件的其它软件文档：

表1. 其它 Quartus II 文档

文档	说明	位置
<i>Quartus II Software Release Notes</i>	提供有关新功能、器件支持、EDA 接口支持的最新信息，以及已知问题和解决方法	在 Altera® 网站上
<i>Quartus II Installation & Licensing for PCs 手册</i>	提供有关软件要求、安装和 PC 许可的详细信息	在 Quartus II 订购包和 Altera 网站上
<i>Quartus II Installation & Licensing for UNIX and Linux Workstations 手册</i>	提供有关软件要求、安装以及 UNIX 和 Linux 工作站许可的详细信息	在 Quartus II 订购包和 Altera 网站上
Quartus II <i>readme.txt</i> 文件	提供有关存储器、磁盘空间和系统要求的信息	在 Quartus II 软件 CD-ROM 中，并与 Quartus II 软件一起安装
<i>Quartus II Scripting Reference Manual</i>	提供关于命令行、Tcl 命令和脚本的信息	在 Altera 网站上
<i>Quartus II Software Quick Start Guide</i>	显示如何为目标器件设置工具、设置时序要求以及编译工具	在 Quartus II 订购包和 Altera 网站上

其他 Altera 文献

Altera 网站 www.altera.com 的 Literature 部分提供许多与 Quartus II 软件相关的主题文档。其中有许多文档可以从 Altera Documentation Library CD 或 Altera Literature Services 中获取。还可以从 ShopAltera 网站 www.shopaltera.com 购买印刷版文档集。

Altera 提供包括以下主题文献：

- ◇ Quartus II 特性和有关设计流程中使用这些特性的帮助向导
- ◇ Altera 器件特性、功能、结构、规范、配置和引出脚
- ◇ 设计解决方案和方法
- ◇ 实现器件功能
- ◇ Altera 编程硬件功能、使用和安装

- ◇ Quartus II 软件与其它 EDA 工具配合使用
- ◇ 使用其它 Altera 软件工具
- ◇ 实现 IP MegaCore® 功能和 Altera 宏功能模块
- ◇ 优化设计和改进性能
- ◇ 综合、仿真和验证帮助向导
- ◇ 产品更新和通知

Altera 网站上的文献是有关 Altera 产品和功能的最新信息；即使在产品发布之后，这些信息也经常更新。Altera 不断添加新文献，提供有关 Altera 工具和器件最新功能的详细信息，按照 Altera 客户要求提供其它信息。



使用 Altera Find Answers 在 Altera Literature 中进行搜索

可以使用 Altera Find Answers（位于 Altera 网站 www.altera.com/answers 的 Support Center 部分），在 Altera 网站的所有文献中进行搜索。Altera Find Answers 使用自然语言处理技术（NLP）来分析您提问的含义和上下文，并提供答案。与返回和关键词相对应文档列表的简单搜索引擎不同，Altera Find Answers 提供实际答案。

索引

A

ADS Standard Tools 软件工具集 214
AHDL 42
AHDL Include Files (.inc) 40
Altera Find Answers 225
Altera Hardware Description Language (AHDL) 42
Altera Megafunction Partners Program (AMPP) 44
Altera on the Web 命令 225
Altera Programming Unit (APU) 165
Altera 网站 225
Altera.com 帐户 225
AMPP 44
Analysis & Elaboration 62
Analysis & Synthesis 4
Analysis & Synthesis Settings 页面 69, 148
APU 165
ARM-based Excalibur MegaWizard Plug-In 216, 218
Assembler 4, 164, 165
Assembly Files (.s, .asm) 214
Assign SignalProbe Pins 对话框 185
Assignment Editor 52, 90, 127
Assignment Editor 命令 90
Avalon 接口 208

B

Back-Annotate Assignments 命令 108, 147
反标 98, 109, 147
批处理文件 20
Binary Files (.bin) 219
black-box 方法 47
Block Design Files (.bdf) 40
Block Editor 40
Block Symbol Files (.bsf) 40, 42
Board-Level 页面 139
ByteBlaster II 下载电缆 165, 177
ByteBlasterMV 下载电缆 165, 177

C

C Source Files (.c) 214
C++ Source Files (.cpp) 214
Chain Description Files (.cdf) 166, 167
Check Resource Properties 命令 196
Chip Editor 89, 190, 193, 194
clear-box 方法 48
命令行可执行文件 15
Comma-Separated Value Files (.csv) 53, 183
编译流程 5, 19
编译, 渐进式 28, 57, 107, 181
Compiler Database Interface 4
Convert MAX+PLUS II Project 命令 38
Convert Programming Files 命令 165, 168
Copy Project 命令 33
Create 命令 195
Create/Update > Create Jam, SVF, or ISC File 命令 165, 168
Create/Update > Create/Update IPS File 命令 165, 168, 169
Create/Update 命令 41, 42
关键路径 145
Customize 对话框 6, 7

D

Design Assistant 4, 71, 89
Design Assistant 页面 71, 72
设计约束 32, 52, 192
设计输入 32
设计分区 57, 107
Design Partitions 窗口 57, 58
Design Space Explorer 94, 153
器件, 编程和配置 164
DSE 94, 153
dse.tcl Tcl 脚本 94, 153
DSP Builder 206, 210
 建立设计 210
 设计流程 207
 生成仿真文件 210
 生成综合文件 211

- 例化功能 210
 SignalCompiler 模块 210
 使用其他 EDA 工具 211
- E**
- Early Timing Estimate 页面 133
 早期时序估算 133
 ECOs 192
 EDA Netlist Writer 4, 115, 117, 140
EDA Tool Settings 页面 13, 66, 202
 EDA 工具
 形式验证 201
 最小时序分析 139
 功耗估算 117
 仿真 115
 指定设置 13, 56, 66, 116, 202
 综合 66
 时序分析 139
 时序仿真 118
 EDIF Input Files (.edf) 40
 EDIF 网表文件 (.edf) 62, 66
Edit Connection 命令 195
 Equations 窗口 146
 EthernetBlaster 下载电缆 165
exc_flash_programmer 应用程序 217
 Executable 和 Linkable Format Files (.elf) 217
Export Assignments 对话框 109
Export 命令 53
Export Database 命令 38
- F**
- Field View 命令 145
Files 页面 33
 Fitter 4, 82
Fitter Settings 页面 90
 适配
 分析 84
 设计流程 82
 优化 89, 150
 Flash Loader Hexadecimal File (.flhex) 165
 内存编程文件 216, 217
 FLEX Chain File (.fcf) 166
- 编译流程 5, 19
 形式验证
 设计流程 200
 指定设置 203
 完整编译 4
 功能仿真
 Quartus II Simulator 121
- G**
- GNUPro for ARM 软件工具集 214
 Graphic Design Files (.gdf) 40
 图形用户界面 3
- H**
- Help 228
 Hexadecimal (Intel 格式) Files (.hex) 172, 216
 Hexadecimal (Intel 格式) Output Files (.hexout) 165, 168, 171, 217
- I**
- I/O Pin State Files (.ips) 165, 168, 169
Import Assignments 命令 59, 109
Import Database 命令 38
 In System Configuration Files (.isc) 165, 168
 渐进式编译 28, 57, 107, 181
 渐进式综合 28, 57, 78
 Integrated Synthesis 63
 Intellectual Property (IP) 功能 44
ISP CLAMP State Editor 对话框 165, 168, 169
- J**
- Jam Byte-Code Files (.jbc) 165, 167, 168
 Jam Files (.jam) 165, 167, 168
 JTAG Chain File (.jcf) 166
 JTAG Indirect Configuration Files (.jic) 165
 JTAG 端口 176
- L**
- Launch Design Space Explorer** 命令 94, 153

- Library Files (.a) 214, 219
 Library Mapping Files (.lmp) 63
 参数化模块库 (LPM) 功能 43
List Paths 命令 137
`list_path` Tcl 命令 137
 LMFs 63
Locate in Timing Closure Floorplan 命令 137
 逻辑选项 69, 91
LogicLock 102, 103
 - 基于模块的设计流程 102
 - 保存中间综合结果 108
 - 使用其他 EDA 工具 111
 - 使用 Tcl 106**LogicLock Region Properties 对话框** 90
LogicLock 区域 103
 - 达到时序逼近 150
 - 导出 109
 - 导入 109
 - 基于路径的分配 151
 - 软 LogicLock 区域 151**LogicLock Regions 窗口** 104
LogicLock Regions Window 命令 104
 LPM 43
- ## M
- makefile 支持 25
makeprogfile 应用程序 215
 MasterBlaster 下载电缆 165, 177
 MAX+PLUS II Assignment & Configuration Files (.acf) 59
 MAX+PLUS II 外观和操作习惯 6
 MAX+PLUS II 快捷菜单 7
 MAX+PLUS II Simulator Channel Files (.scf) 124
 MAX+PLUS II Symbol Files (.sym) 42
 MegaCore 功能 44
 宏功能模块 43
 - 例化 43, 46, 64
 - 在其他 EDA 工具中例化 47, 64**MegaWizard Plug-In Manager** 43, 215
 - `qmegawiz` 可执行文件 17
 - 使用 black-box 方法 47
 - 使用 clear-box 方法 48**Memory Editor** 117
- memory initialization 数据文件 219
Memory Initialization Files (.mif) 117
 Messages 窗口 84
 mySupport 网站 224, 225
- ## N
- NativeLink** 118, 141
 网表优化
 - 达到时序逼近 148
 - 适配 150
 - 物理综合 150
 - 综合 68, 71, 148**New Project Wizard** 33
- ## O
- OpenCore** 45
Operating Conditions 页面 159
- ## P
- 分区 28, 57, 107
 被动编程文件 217
Path 对话框 151
 Perl 脚本 20
Physical Synthesis Optimizations 页面 91, 148
 物理综合, 优化 91, 150
Pin Planner 54
 POFs 164, 167, 168
 功耗分析 156
 - PowerPlay Early Power Estimator 160
 - PowerPlay Power Analyzer 156
 - PowerPlay Power Analyzer Tool 157**Power Input Files (.pwf)** 117
PowerFit Fitter 82
PowerPlay Early Power Estimator 156, 160
PowerPlay Power Analyzer Settings 页面 158
PowerPlay Power Analyzer Tool 156, 157
PowerPlay Power Analyzer Tool 命令 156
Print All Recommendations 命令 93
Print Recommendation 命令 93
Priority 对话框 105

programmable logic Partial SRAM Object Files (.psof) 218

Programmer 164

 quartus_pgm 可执行文件 166

 quartus_pgmw 可执行文件 17

Programmer Object Files (.pof) 164, 167, 168, 217

Programming Files 168

Project Navigator 窗口 33

Q

qmegawiz 可执行文件 17

QSF 33, 104, 130

Quartus II Default Settings Files (.qdf) 33

Quartus II Project Files (.qpf) 33

Quartus II 快捷菜单 7

Quartus II Settings Files (.qsf) 33, 104, 130

Quartus II 软件

 命令行设计流程 15

 EDA 工具设计流程 10, 22

 通用设计流程 2

 GUI 设计流程 3

Quartus II Tutorial 229

Quartus II Workspace Files (.qws) 33

quartus_asm 可执行文件 17, 165

quartus_cdb 可执行文件 17, 38, 109

quartus_cpf 可执行文件 18, 172

quartus_drc 可执行文件 17, 72

quartus_eda 可执行文件 17, 117, 141

quartus_fit 可执行文件 17, 83

quartus_map 可执行文件 17, 63

quartus_pgm 可执行文件 17, 166

quartus_pgmw 可执行文件 17, 166

quartus_pow 可执行文件 17, 157

quartus_sh 可执行文件 18

quartus_sim 可执行文件 17

quartus_stp 可执行文件 18, 178

quartus_stpw 可执行文件 17, 179

quartus_swb 可执行文件 18, 214

quartus_tan 可执行文件 17, 132

R

RAM Initialization Files (.rif) 117

Raw Binary Files (.rbf) 165, 168, 218

Regions 窗口 90

Remove Connection 命令 195

Report 窗口 86, 135

Resource Optimization Advisor 92

Resource Property Editor 190, 194

Revisions 对话框 35

RTL Viewer 189

Run EDA Simulation Tool 命令 117

Run EDA Timing Analysis Tool 命令 140

S

Selected Elements Window 命令 194

Serial Vector Format Files (.svf) 165, 168

Set as Design Partition 命令 57

设置

 Analysis & Synthesis 69

 Compiler 56

 Design Assistant 71

 EDA tools 13, 66, 116, 202

 Fitter 90

 Fitter optimization 150

 形式验证 203

 HardCopy 56

 物理综合优化 91

 PowerPlay Power Analyzer 158

 Quartus II Project Files (.qpf) 33

 Quartus II Settings Files (.qsf) 33

 SignalProbe 185

 SignalTap II Logic Analyzer 178, 181

 Simulator 56, 122

 Software Builder 56, 215, 216, 218, 219

 synthesis optimization 71, 148

 Timing Analyzer 56

 Verilog HDL 输入 63

 VHDL 输入 63

Settings 对话框 56, 90, 127

shell, Tcl 脚本 18

Shop Altera 网站 230

Signal Activity Files (.saf) 114, 118, 156

SignalProbe 特性 176, 184

SignalProbe Settings 页面 185

SignalTap II Files (.stp) 177

SignalTap II Logic Analyzer 176, 177

- S**
- SignalTap II Logic Analyzer 页面 181
 - Simulation 页面 116
 - Simulator 121
 - Simulator 页面 121
 - Simulator Tool 124
 - Slave Binary Image File (.sbi) 216
 - SOFs 164, 167, 168
 - Software Build Settings 页面 215, 216, 218, 219
 - Software Builder 214
 - makeprofile** 应用程序 215
 - 存储器初始化数据文件 219
 - SOPC Builder 206
 - System Contents** 页面 209
 - System Generation** 页面 209
 - SRAM Object Files (.sof) 164, 167, 168, 218
 - Standard Delay Format Output Files (.sdo) 115
 - Start Early Timing Estimate 命令 133
 - Start EDA Netlist Writer 命令 117, 140
 - Start EDA Synthesis 命令 67
 - Start I/O Assignment Analysis 命令 60
 - Start Partition Merge 命令 79
 - Start SignalProbe Compilation 命令 185
 - Start Software Build 命令 214
 - Start Timing Analyzer (Fast Timing Model) 命令 131
 - Start Timing Analyzer 命令 131
 - State Machine Viewer 75
 - Support Center 225
 - Symbol Editor 42
 - Synopsys Design Constraints File (.sdc) 141
 - 综合
 - VHDL 和 Verilog HDL 支持 63
 - Synthesis Netlist Optimizations 页面 71, 148
 - System Build Descriptor Files (.sbd) 216
- T**
- Table Files (.tbl) 183
 - Tabular Text Files (.ttf) 165, 168, 218
 - Tcl 18, 20, 22
 - Technology Map Viewer 77, 138, 189
 - Technology Map Viewer 命令 138
- U**
- Text Design Files (.tdf) 40
 - Text Editor 41
 - Time Groups 对话框 131
 - Timing Analysis 页面 139
 - Timing Analyzer 4, 126
 - Timing Closure 平面布局图 87, 144
 - Timing Optimization Advisor 147
 - Timing Requirements & Options 页面 127
 - 时序仿真
 - EDA 工具 118
 - Quartus II Simulator 121
 - Timing 向导 52, 127
- V**
- USB-Blaster 下载电缆 165, 177
- W**
- Value Change Dump Files (.vcd) 156, 183
 - Vector Files (.vec), 124
 - Vector Table Output Files (.tbl) 124
 - Vector Waveform Files (.vwf) 124, 183
 - Verilog Design Files (.v) 40, 62, 66
 - Verilog HDL 42, 63
 - Verilog HDL Input 页面 63
 - Verilog Output Files (.vo) 115, 200
 - Verilog Quartus Mapping Files (.vqm) 40, 62, 66, 108, 150, 200
 - Verilog Test Bench Files (.vt) 117
 - VHDL 42, 63
 - VHDL Design Files (.vhd) 40, 62, 66
 - VHDL Input 页面 63
 - VHDL Output Files (.vho) 115
 - VHDL Test Bench Files (.vht) 117
 - View Port Connections 命令 194
 - View Properties 命令 194
 - VQM Files 62, 66, 108, 150, 200

