

Quartus® II 简介

ALTERA®

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



QUARTUS® II

Altera、Altera 徽标、FastTrack、MAX、MAX+PLUS、MAX+PLUS II、MegaCore、MegaWizard、NativeLink、Nios、OpenCore、Quartus、Quartus II、Quartus II 徽标和 SignalTap 是 Altera Corporation 在美国和其它国家（或地区）的注册商标。ByteBlaster、ByteBlasterMV、Excalibur、HardCopy、IP MegaStore、Jam、LogicLock、MasterBlaster、MegaLAB、PowerFit 和 SignalProbe 是 Altera Corporation 在美国和其它国家（或地区）的商标和 / 或服务标志。Altera Corporation 使用的产品设计元素和助记键受版权法和 / 或商标法的保护。

Altera Corporation 承认本文档提及的其它组织的产品或服务商标和 / 或服务标志，特别是：Mentor Graphics 和 ModelSim 是 Mentor Graphics Corporation 的注册商标，ModelTechnology 是 Mentor Graphics Corporation 的商标。

Altera 保留更改本文档中所述设备或设备规范的权利，恕不另行通知。Altera 建议客户在下订单之前要取得设备规范的最新版本，以确认客户所依赖的信息是最新的。Altera 保证自己的半导体产品的性能与 Altera 的标准担保一致，符合当前规范。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的测试。如果没有书面协议另做规定，Altera 对于使用本文档所述的半导体设备而引起或有关的 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 也不保证或表示 Altera 的任何专利权、版权或其它知识产权涵盖了或者与可能使用或正在使用此类半导体器件在任何组合、机器或流程中有关。

没有 Altera Corporation 总裁以书面形式明确同意，Altera 产品不得用作生命支持设备或系统中的关键组件。其中：

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

本文档中提及的产品受以下一个或多个美国专利的保护：RE35977； RE37060； 6557094； 6556502； 6549045； 6538470； 6538469； 6535031； 6532170； 6531889； 6526461； 6525678； 6525564； 6515508； 6515507； 6507216； 6492834； 6492833； 6490717； 6490714； 6489817； 6486702； 6485843； 6483886； 6481000； 6480995； 6480028； 6480027； 6480025； 6472903； 6472272； 6470441； 6469553； 6467036； 6467017； 6462597； 6462577； 6462414； 6460148； 6459303； 6457073； 6453382； 6448820； 6442073； 6437650； 6433585； 6433579； 6429681； 6423572； 6417694； 6417692； 6417550； 6414518； 6414514； 6412812； 6411124； 6410379； 6408432； 6407576； 6407450； 6404225； 6401230； 6400635； 6400598； 6400290； 6396304； 6392954； 6392438； 6389558； 6384630； 6384625； 6377069； 6373280； 6373278； 6369624； 6369613； 6367058； 6362956； 6366498； 6366224； 6366121； 6366120； 6366119； 6365929； 6363505； 6362646； 6359469； 6359468； 6356110； 6356108； 6353552； 6353551； 6351152； 6351144； 6347061； 6346827； 6344989； 6344758； 6344755； 6342794； 6342792； 6340897； 6337578； 6335896； 6335636； 6335635； 6335634； 6326812； 6326807； 6323680； 6323677； 6321369； 6321367； 6320411； 6317860； 6317771； 6317367； 6314550； 6311309； 6301694； 6300794； 6300792； 6298319； 6297565； 6295230； 6294928； 6292116； 6292017； 6292016； 6288970； 6286114； 6285211； 6282122； 6281704； 6279145； 6278291； 6278288； 6275065； 6271729； 6271681； 6271680； 6271679； 6269020； 6268623； 6265926； 6265895； 6265746； 6263482； 6263400； 6262933； 6262595； 6259588； 6259272； 6259271； 6255850； 6255846； 6252422； 6252419； 6249149； 6249143； 6247155； 6247147； 6246270； 6246260； 6243304； 6243326； 6242946； 6242941； 6242772； 6239615； 6239612； 6236597； 6236260； 6236231； 6236094； 6232893； 6226201； 6225823； 6225822； 6219785； 6219284； 6218876； 6218860； 6218859； 6215326； 6212668； 6208162； 6205579； 6204688； 6202185； 6201404； 6198303； 6195788； 6195772； 6192445； 6191998； 6191611； 6191608； 6187634； 6185725； 6184710； 6184707； 6184706； 6184705； 6184703； 6182247； 6182200； 6181162； 6181161； 6181160； 6181159； 6180425； 6177844； 6175952； 6173245； 6172900； 6169417； 6167364； 6166559； 6163195； 6163166； 6161211； 6161049； 6157212； 6157210； 6157208； 6154055； 6154055； 6151258； 6150840； 6147511； 6144573； 6137313； 6134707； 6134705； 6134173； 6134166； 6130555； 6130552； 6128692； 6128215； 6127865； 6127846； 6127823； 6127844； 6127217； 6122270； 6122209； 6121790； 6120550； 6118720； 6118302； 6115312； 6114915； 6112020； 6111026； 6108239； 6107854； 6107825； 6107824； 6107822； 6107820； 6104208； 6102964； 6097211； 6094064； 6091258； 6091102； 6085317； 6084427； 6081449； 6080204； 6078521； 6076179； 6075380； 6072358； 6072332； 6069487； 6066960； 6066599； 6060903； 6058452； 6057707； 6052755； 6052327； 6052309； 6049225； 6049223； 6045252； 6043676； 6040712； 6038171； 6037829； 6034857； 6034540； 6034536； 6032159； 6031763； 6031391； 6029236； 6028808； 6028808； 6028787； 6026226； 6025737； 6023439； 6020760； 6020759； 6020758； 6018490； 6018476； 6014334； 6011744； 6011730； 6011406； 6005806； 6005379； 6002182； 5999016； 5999015； 5998295； 5998263； 5996039； 5986470； 5986465； 5983277； 5982195； 5978476； 5977793； 5977791； 5970255； 5969626； 5968161； 5966597； 5963565； 5963069； 5963051； 5963049； 5959891； 5953537； 5949991； 5949710； 5949250； 5949239； 5948570； 5943267； 5942914； 5940852； 5939790； 5936425； 5926036； 5925904； 5923567； 5915756； 5915017； 5914904； 5914509； 5909450； 5909375； 5909126； 5905675； 5904524； 5900743； 5898630； 5898628； 5898318； 5894228； 5893088； 5892683； 5883850； 5883526； 5880725； 5880597； 5880596； 5878250； 5875112； 5873113； 5872529； 5872463； 5870414； 5869980； 5869979； 5861760； 5859544； 5859542； 5850365； 5850152； 5850151； 5848005； 5847617； 5845385； 5844584； 5838628； 5838584； 5835998； 5834849； 5828229； 5825197； 5821787； 5821773； 5821771； 5815726； 5815024； 5815053； 5812479； 5812450； 5809281； 5809344； 5805516； 5802540； 5801541； 5796671； 5796267； 5793246； 5790469； 5787009； 5771264； 5768562； 5768372； 5767734； 5764583； 5764658； 5764080； 5764079； 5761099； 5760624； 5757207； 5757070； 5744991； 5744383； 5740110； 5732020； 5729495； 5717901； 5705939； 5699312； 5699020； 5696455； 5694058； 5693540； 5691653； 5689195； 5680061； 5672985； 5670895； 5668771； 5659717； 5656528； 5650734； 5649163； 5642262； 5642080； 5633830； 5631576； 5621312； 5614840； 5612642； 5608337； 5606276； 5606266； 5604453； 5598109； 5598108； 5592106； 5592102； 5590305； 5583749； 5581501； 5574893； 5572717； 5572148； 5572067； 5557040； 5567177； 5565793； 5563952； 5561757； 5557217； 5555214； 5550842； 5550782； 5548552； 5548228； 5543730； 5541530； 5537341； 5537295； 5537057； 5525917； 5525827； 5523706； 5523247； 5517186； 5498975； 5495182； 5494356； 5493519； 5490266； 5488586； 5487143； 5486775； 5485103； 5485102； 5483178； 5477474； 5473264； 5463328； 5443924； 5438295； 5436575； 5434674； 5434514； 5432467； 5414312； 5399922； 5396452； 5378449； 5376844； 5375086； 5371422； 5369314； 5359243； 5359242； 5353248； 5352940； 5350954； 5349255； 5341308； 5341048； 5341044； 5329487； 5317212； 5317210； 5315172； 5309046； 5301146； 5294975； 5285153； 5280203； 5274581； 5272368； 5268598； 5266307； 5260611； 5260610； 5258668； 5247478； 5247477； 5243233； 5241224； 5237219； 5220533； 5220214； 5218932； 5166604； 5162680； 5144167； 5138576； 5128565； 5121006； 5111423； 5097208； 5091663； 5066873； 5045772； 4991921； 4930107； 4930098； 4930097； 4912342； 4903223； 4899070； 4899067； 4871930； 4864161； 4831573； 4829018； 4785423； 4774421； 4713792； 4677318； 4617479； 4609986 和某些国外专利。还有其它正在申请的专利。

Altera 产品受许多美国和国际专利及正在申请的专利、集成电路光罩权和版权的保护。

版权所有 © 2003 Altera Corporation. 保留所有权利。



前言

您现在阅读的是 *Quartus II 简介* 手册。Altera® Quartus® II 设计软件是适合单芯片可编程系统 (SOPC) 的最全面的设计环境。如果您以前用过 MAX+PLUS® II 软件、其它设计软件或 ASIC 设计软件，并且准备改用 Quartus II 软件，或如果您对 Quartus II 软件有了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。

本手册针对的读者是 Quartus II 软件的初学者，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显示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。

第一章概述了主要图形用户界面、EDA 工具和命令行界面设计流程。接下来的每一章开头都介绍了该章的具体用途，并对每个任务流加以概述。它显示了如何将 Quartus II 软件与现有的 EDA 工具和命令行设计流程集成在一起。另外，手册还向您推荐了有效使用 Quartus II 软件的其它可用资源，例如 Quartus II 联机帮助和 Quartus II 联机教程、应用程序说明、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，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、性能和时间要求。

文件编制约定

Quartus® II 简介手册采用以下约定，便于查找和解释信息。

印刷约定

Quartus II 文档使用以下印刷约定：

视觉提示：

含义：

粗体且首字母大写

命令名称，对话框、页面和选项卡标题以及按钮名称以粗体显示，且首字母大写。例如：**Find Text** 命令，**Save As** 对话框和 **Start** 按钮。

粗体

命令名称、项目名称、磁盘驱动器名称、文件名称、文件名扩展、软件实用程序名称、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** 命令、**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字段和菜单名称的首字母大写。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。

斜体且首字母大写

帮助类别、手册标题、手册中的章节标题、应用程序说明和简短名称以斜体显示，且首字母大写。例如：*FLEX1m 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

必须如显示的内容一样一字不差地键入的任何内容以 Courier 字体显示。例如：

```
\quartus\bin\lmulti lmhostid.
```



Enter 或回车键。



当项目的顺序不重要时，在项目列表中使用项目符号。



脚印指示您要去哪里获取某特定主题的详细信息。



对号指示仅由一个步骤构成的程序。



用手指向需要特别注意的信息。

术语

在 *Quartus II 简介* 手册中使用以下术语：

术语：

“单击”

“双击”

“选取”

“选择”

“打开” / “关闭”

含义：

表示快速按下并释放鼠标左键。

表示连续两次快速的单击。

表示需要使用鼠标或键组合启动一个操作。

表示需要使用键组合或鼠标突出显示文本和 / 或对象或对话框中的选项。选择不会启动操作。例如：选择 **Chain Description File**，然后单击 **OK**。

表示必须单击复选框以打开或关闭一个功能。

目录

前言	iii
文件编制约定	v
第 1 章：可编程逻辑设计流程	1
简介	2
图形用户界面设计流程	3
EDA 工具设计流程	7
命令行设计流程	11
命令行可执行文件	12
使用标准命令行命令和脚本	14
使用 Tcl 命令	16
建立 Makefile 脚本	19
第 2 章：设计输入	23
简介	24
建立工程	25
建立设计	26
使用 Quartus II Block Editor	27
使用 Quartus II Text Editor	28
使用 Quartus II Symbol Editor	28
使用 Verilog HDL、VHDL 与 AHDL	29
使用 Altera 宏功能模块	30
使用知识产权 (IP) 函数	30
使用 MegaWizard Plug-In Manager	31
在 Quartus II 软件中对宏功能模块进行实例化	32
在 Verilog HDL 和 VHDL 中实例化	33
使用端口和参数定义	33
推断宏功能模块	33
在 EDA 工具中实例化宏功能模块	33
使用 black-box 方法	33
按推断进行实例化	34
使用 clear-box 方法	34
指定初始设计的约束条件	36
使用 Assignment Editor	36
使用 Settings 对话框	37
导入分配	38
验证引脚分配	39
设计方法与设计规划	39
从上到下与从下到上的设计方法比较	39
基于块的设计流程	39
设计分割	40

第 3 章：综合	43
简介	44
使用 Quartus II VHDL 及 Verilog HDL 集成综合	45
使用其它 EDA 综合工具	48
控制 Analysis & Synthesis	50
使用编译器指令和属性	50
使用 Quartus II 逻辑选项	50
使用 Quartus II 综合网表优化选项	51
使用 Design Assistant 检查设计可靠性	52
第 4 章：仿真	55
简介	56
使用 EDA 工具进行设计仿真	57
指定 EDA 仿真工具设置	58
生成仿真输出文件	59
仿真流程	60
功能仿真流程	61
NativeLink 仿真流程	61
手动时序仿真流程	62
仿真库	62
使用 Quartus II 仿真器进行仿真设计	64
指定 仿真器设置	64
进行仿真	64
建立波形文件	65
进行 PowerGauge 功耗估算	66
仿真 Excalibur 设计	66
在 Quartus II 软件中仿真 Excalibur 设计	66
总线功能模型与 EDA 工具结合使用	67
全带区模型与 EDA 工具的结合使用	67
ESS 模型与 EDA 工具结合使用	67
第 5 章：布局布线	69
简介	70
分析布局布线结果	71
使用消息窗口查看布局布线结果	71
使用报告窗口或报告文件查看布局布线结果	72
使用 Floorplan Editor 分析结果	74
使用 Design Assistant 检查设计的可靠性	75
优化布局布线	76
使用位置分配	76
设置用于控制布局布线的选项	77
设置 Fitter 选项	77
设置布局布线优化与物理综合选项	77
设置影响布局布线的逻辑选项	78
使用设计空间管理器 (Design Space Explorer)	78

执行增量布局布线.....	81
通过反标保留分配.....	81
第 6 章：基于块的设计	85
简介.....	86
Quartus II 基于块的设计流程.....	86
使用 LogicLock 区域.....	87
保存中间综合结果.....	91
反标 LogicLock Region Assignments	92
导出与导入 LogicLock 分配	92
LogicLock 与 EDA 工具结合使用.....	95
第 7 章：时序分析	97
简介.....	98
在 Quartus II 软件中进行时序分析.....	99
指定时序要求	99
指定工程范围的时序设置	100
指定个别时序分配	101
进行时序分析	101
查看时序分析结果.....	102
使用报告窗口	103
进行分配与查看延时路径	104
使用 EDA 工具进行时序分析	106
使用 PrimeTime 软件.....	106
使用 BLAST 和 Tau 软件.....	107
第 8 章：时序逼近	109
简介.....	110
使用时序逼近布局图.....	110
查看分配与布线	111
执行分配	112
使用网表优化实现时序逼近.....	113
使用 LogicLock 区域实现时序逼近	116
软 LogicLock 区域	116
基于路径的分配	116
第 9 章：编程与配置	119
简介.....	120
使用 Programmer 对一个或多个器件进行编程	123
建立辅助编程文件.....	124
建立其它编程文件格式	124
转换编程文件	126
使用 Quartus II 软件通过远程 JTAG 服务器进行编程.....	129

第 10 章 : 调试	131
简介	132
使用 SignalTap II 逻辑分析器	133
设置和运行 SignalTap II 逻辑分析器	133
分析 SignalTap II 数据	135
使用 SignalProbe	138
使用 Chip Editor	140
第 11 章 : 工程更改管理	141
简介	142
使用 Chip Editor 识别延时与关键路径	143
使用 Resource Property Editor 修改资源属性	145
使用 Change Manager 查看和管理更改	146
验证 ECO 更改的效果	148
第 12 章 : 系统级设计	149
简介	150
使用 SOPC Builder 建立 SOPC 设计	151
建立系统	152
生成系统	153
使用 DSP Builder 建立 DSP 设计	154
实例化功能	154
生成仿真文件	154
生成综合文件	154
第 13 章 : 软件开发	157
简介	158
在 Quartus II 软件中使用 Software Builder	158
指定软件构建设置	159
生成软件输出文件	159
生成闪存编程文件	161
生成无源编程文件	162
生成存储器初始化数据文件	163
第 14 章 : 安装、许可与技术支持	165
安装 Quartus II 软件	166
许可 Quartus II 软件	167
获取技术支持	168
第 15 章 : 文档和其它资源	171
获取联机帮助	172
使用 Quartus II 联机教程	173
其它 Quartus II 软件文档	173
其它 Altera 文献	174
修订历史记录	176
索引	177

第一章

可编程逻辑设计流程



1

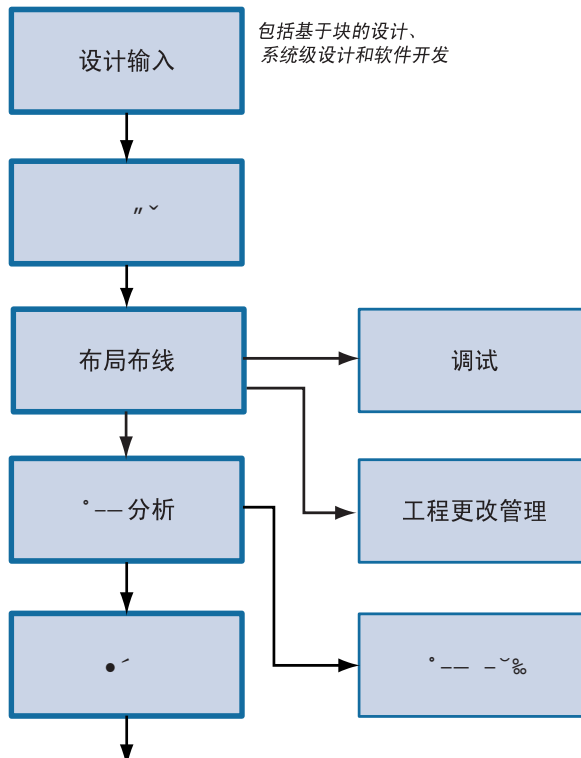
第 1 章内容：

简介	2
图形用户界面设计流程	3
EDA 工具设计流程	7
命令行设计流程	11

简介

Altera® Quartus® II 设计软件提供完整的多平台设计环境，它可以轻易满足特定设计的需要。它是单芯片可编程系统 (SOPC) 设计的综合性环境。Quartus II 软件拥有 FPGA 和 CPLD 设计的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 软件允许您在设计流程的每个阶段使用 Quartus II 图形用户界面、EDA 工具界面或命令行界面。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的选项。本章介绍适用于各个设计流程的选项。本手册的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

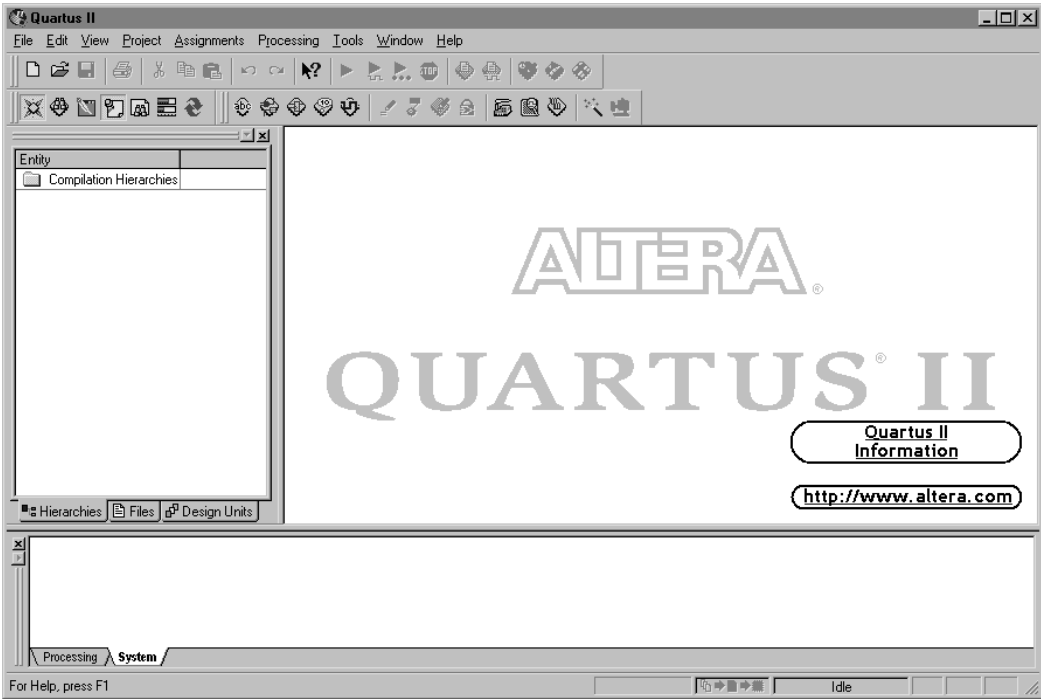
您可以使用 Quartus II 软件完成设计流程的所有阶段；它是完整且易用的独立解决方案。图 2 显示 Quartus II 图形用户界面为设计流程的每个阶段所提供的功能。

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



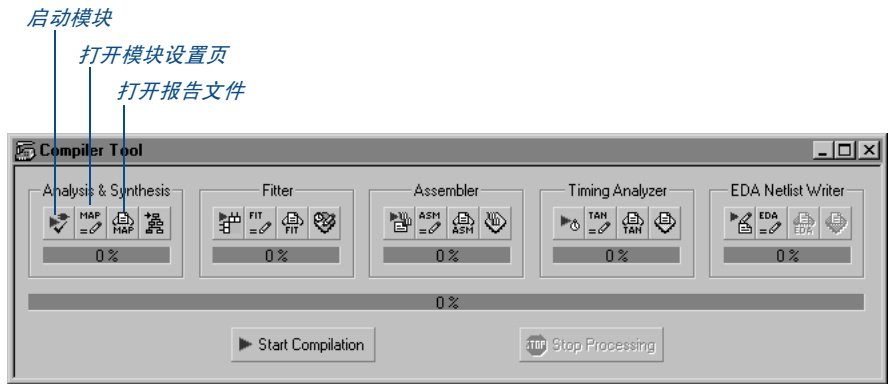
Quartus II 软件包括模块化编译器。编译器包括以下模块（标有星号的模块表示在编译期间可选，具体要视您的设置而定）：

- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
- Design Assistant*
- EDA Netlist Writer*
- Compiler Database Interface*

可以在全编译过程中通过选择 **Start Compilation**（Processing 菜单）来运行所有的编译器模块。若要单独运行各个模块，可以通过选择 **Start**（Processing 菜单），然后从 **Start** 子菜单中为模块选择相应的指令。

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编译器模块。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。

图 4. Compiler Tool 窗口



以下步骤描述 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**（文本编辑器）建立 Verilog HDL、VHDL 或 Altera 硬件描述语言 (AHDL) 设计。您可以使用 **Block Editor**（原理图编辑器）建立流程图或原理图。流程图中可以包含代表其它设计文件的符号。还可以使用 **MegaWizard® Plug-In Manager** 生成宏功能模块和 IP 内核的自定义变量，在设计中将它们实例化。
3. （可选）使用 **Assignment Editor**、**Settings** 对话框（Assignments 菜单）、**Floorplan Editor** 和 / 或 **LogicLock™** 功能指定初始设计的约束条件。
4. （可选）使用 **SOPC Builder** 或 **DSP Builder** 建立系统级设计。
5. （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。
6. 使用 **Analysis & Synthesis** 对设计进行综合。
7. （可选）使用仿真器对设计执行功能仿真。
8. 使用 **Fitter** 对设计执行布局布线。在对源代码进行少量更改之后，还可以使用增量布局布线。
9. 使用 **Timing Analyzer** 对设计进行时序分析。

10. 使用仿真器对设计进行时序仿真。
11. (可选) 使用物理综合、时序底层布局图、LogicLock 功能、Settings 对话框和 Assignment Editor 进行设计优化，实现时序关闭。
12. 使用 Assembler 为设计建立编程文件。
13. 使用编程文件、Programmer 和 Altera 硬件编程器对器件进行编程；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。
14. (可选) 使用 SignalTap® II Logic Analyzer、SignalProbe™ 功能或 Chip Editor 对设计进行调试。
15. (可选) 使用 Chip Editor、Resource Property Editor 和 Change Manager 进行工程更改管理。

EDA 工具设计流程



Quartus II 软件使您能够在设计流程的不同阶段使用您熟悉的 EDA 工具。可以将这些工具与 Quartus II 图形用户界面或 Quartus II 命令行可执行文件一起使用。请参见第 7 页上的图 5。

图 5. EDA 工具设计流程

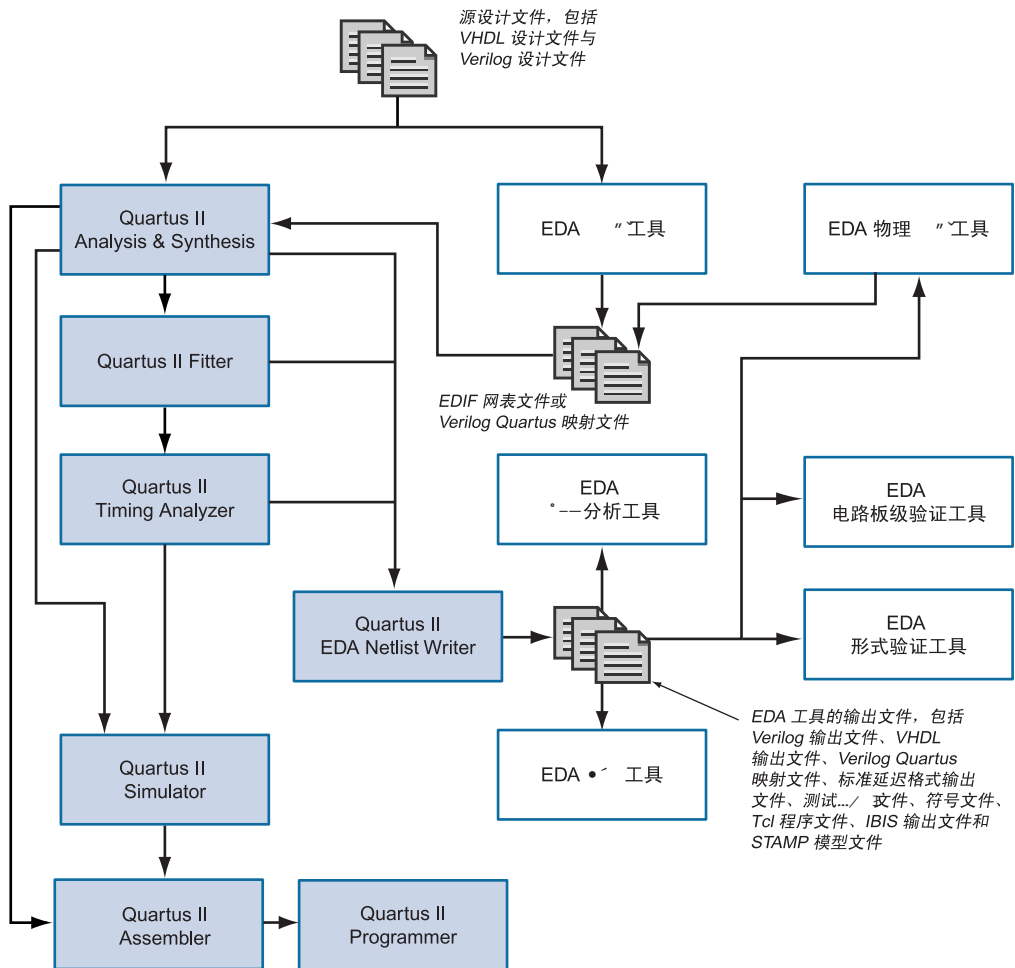


表 1 显示 Quartus II 软件支持的 EDA 工具，并指出哪个 EDA 工具可支持 NativeLink®。NativeLink 技术在 Quartus II 软件和其它 EDA 工具之间无缝地传送信息，并允许在 Quartus II 软件中自动运行 EDA 工具。

表 1. 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
综合	Mentor Graphics Design Architect	
	Mentor Graphics LeonardoSpectrum	✓
	Mentor Graphics Precision RTL Synthesis	✓
	Mentor Graphics ViewDraw	
	Synopsys Design Compiler	
	Synopsys FPGA Express	
	Synopsys FPGA Compiler II	✓
	Synplicity Synplify	✓
	Synplicity Synplify Pro	
仿真	Cadence NC-Verilog	✓
	Cadence NC-VHDL	✓
	Cadence Verilog-XL	
	Model Technology™ ModelSim®	✓
	Model Technology ModelSim-Altera	✓
	Synopsys Scirocco	✓
	Synopsys VSS	
	Synopsys VCS	
时序分析	Mentor Graphics Blast（通过标签）	
	Mentor Graphics Tau（通过标签）	
	Synopsys PrimeTime	✓

表 1. 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
板级验证	Hyperlynx（通过 Signal Integrity IBIS）	
	XTK（通过 Signal Integrity IBIS）	
	ICX（通过 Signal Integrity IBIS）	
	SpectraQuest（通过 Signal Integrity IBIS）	
	Mentor Graphics Symbol Generation (Viewdraw)	
形式验证	Verplex Conformal LEC	
再综合	Aplus Design Technologies (ADT) PALACE	✓
	Synplicity Amplify	

以下步骤描述其它 EDA 工具与 Quartus II 软件配合使用时的基本设计流程。有关可支持的 EDA 工具列表，请参阅第 8 页上的表 1。

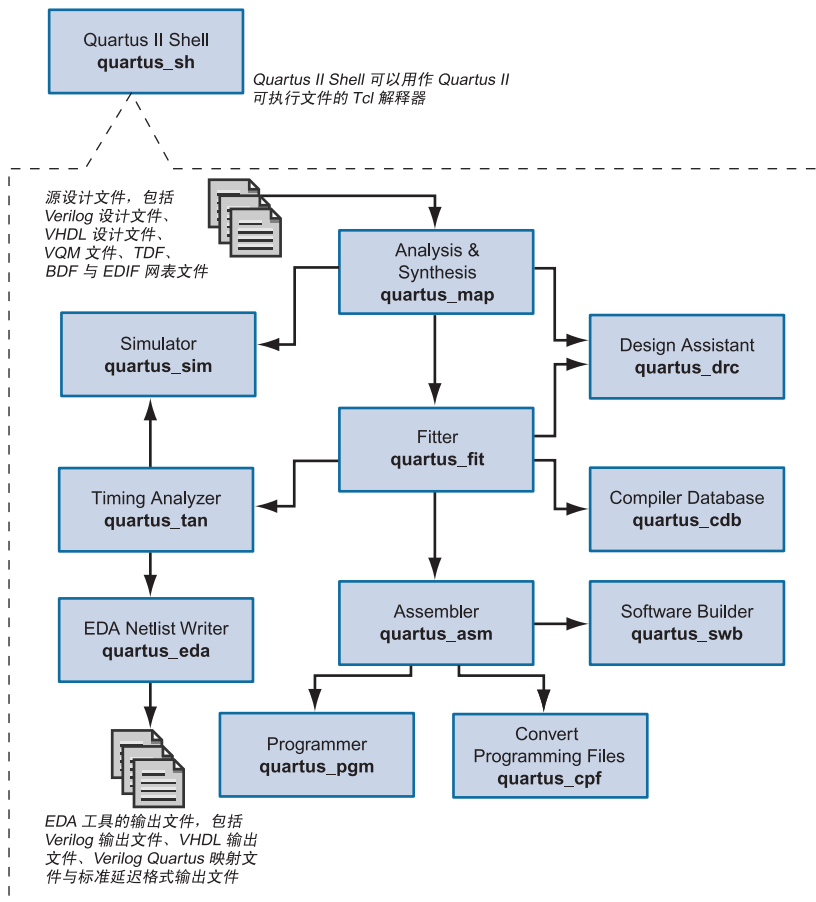
1. 建立新工程并指定目标器件或器件系列。
2. 使用标准文本编辑程序建立 VHDL 或 Verilog HDL 设计文件。若需要的话，可以对运用库函数进行实例化或使用 **MegaWizard Plug-In Manager**（Tools 菜单）建立宏功能模块的自定义变量。
3. 使用其中一个 Quartus II 支持的 EDA 综合工具综合您的设计，并生成 EDIF 网表文件 (.edf) 或 VQM 文件 (.vqm)。
4. （可选）使用其中一个 Quartus II 支持的仿真工具对您的设计执行功能仿真。
5. 在 Quartus II **Settings** 对话框（Assignments 菜单）中，指定选项和库映射文件，以处理使用其它设计输入 \ 综合工具或通过 **MegaWizard Plug-In Manager** 生成的 EDIF 网表文件 (.edf)、VHDL 设计文件 (.vhd)、Verilog 设计文件 (.v) 和 AHDL 文本设计文件 (.tdf)。
6. （可选）在 Quartus II **Settings** 对话框中，为生成 VHDL 输出文件 (.vho)、Verilog 输出文件 (.vo)、标准延时格式输出文件 (.sdo)、标记模型文件、PartMiner XML 格式文件 (.xml) 和 IBIS 输出文件 (.ibs) 指定选项。

7. 使用 Quartus II 软件编译设计并进行布局布线。可以执行全编译，或者分别运行编译器模块：
 - 运行 Analysis & Synthesis，对设计进行综合，并将设计中的函数映射到正确的库模块上。
 - 运行 Fitter，对设计进行布局布线。
 - 运行 Timing Analyzer，对设计进行时序分析。
 - 运行 EDA Netlist Writer，生成与其它 EDA 工具配合使用的输出文件。
 - 运行 Assembler，为您的设计建立编程文件。
8. （可选）使用其中一个 Quartus II 支持的 EDA 时序分析工具对设计进行时序分析。
9. （可选）使用其中一个 Quartus II 支持的 EDA 仿真工具对设计进行时序仿真。
10. （可选）使用其中一个 Quartus II 支持的 EDA 板级验证工具进行板级验证。
11. （可选）使用其中一个 Quartus II 支持的 EDA 形式验证工具进行形式验证，确保 Quartus 布线后网表与综合网表等同。
12. （可选）使用其中一个 Quartus II 支持的 EDA 再综合工具进行板级再综合。
13. 使用编程文件、Programmer 和 Altera 硬件编程器对器件进行编程；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。

命令行设计流程

Quartus II 软件提供完整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 软件和建立 Makefile。有关命令行设计流程的图示说明，请参见图 6。

图 6. 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的单独可执行文件。每个可执行文件仅在运行时才占用内存。这些可执行文件可以与标准的命令行命令和脚本配合使用，与 Tcl 脚本配合使用，还可在 Makefile 脚本中使用。有关所有可用命令行可执行文件的列表，请参阅表 2。



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 界面。

表 2. 命令行可执行文件 (第 1 部分, 共 2 部分)

可执行文件名称	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、综合设计并对工程的设计文件执行技术映射。
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	为编程或配置目标器件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。视所用的选项而定，在运行 EDA Netlist Writer 之前，必须成功运行 Analysis & Synthesis、Fitter 或 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 Database 的 VQM 文件，使它们可以用于反标和 LogicLock 功能。在运行 Compiler Database Interface 之前必须成功运行 Fitter 或 Analysis & Synthesis。
quartus_sim	仿真器	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_pgm	Programmer	对 Altera 器件进行编程。

表 2. 命令行可执行文件（第 2 部分，共 2 部分）

可执行文件名称	标题	功能
quartus_cpf	转换编程文件	将编程文件转换为辅助编程文件格式。
quartus_swb	Software Builder	为 Excalibur 嵌入式处理器处理设计。
quartus_sh	Tcl Shell	为 Quartus II 软件提供 Tcl 脚本外壳。

获取有关 Quartus II 可执行文件的帮助信息

若要获取有关每个 Quartus II 可执行文件的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：

```
<可执行文件名称> -h ←
<可执行文件名称> -h ←
<可执行文件名称> --help=<主题或选项名称> ←
```

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

```
quartus_sh --qhhelp ←
```

既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编译器可执行文件：

```
quartus_sh --flow compile <工程名称>
[-c <编译器设置文件名称>] ←
```

此命令将在全编译过程中运行 **quartus_map**、**quartus_fit**、**quartus_asm** 和 **quartus_tan** 可执行文件。视设置而定，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

若在 Quartus II 软件的以前版本中使用 **quartus_cmd** 可执行文件进行工程编译，由于本版本具有向后兼容性，因此仍可支持该执行文件；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 12 页上的表 2 中列出的可执行文件。如果使用 **quartus_cmd** 可执行文件编译设计，您应键入以下命令和选项：

```
quartus_cmd <工程名称> -c <编译器设置文件名称>.csf ←
```

有些可执行文件可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：

< 工程名称或设置名称 >.< 可执行文件缩略名称 >.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 Analysis & Synthesis，并生成名为 **chiptrip.map.rpt** 的报告文件。



将设置文件与 Quartus II 可执行文件结合使用

Altera 建议在使用 Quartus II 可执行文件时要将设置文件命名为与工程相同的名称。

如果设置文件名称与工程名称不同，您可以使用 **-c** 选项指定要使用的设置文件名称。例如，如果要为具有 **speed_ch** 设置的 **chiptrip** 工程运行 **quartus_map** 可执行文件，可以在命令提示符下键入以下命令：

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

quartus_map 可执行文件进行 Analysis & Synthesis，并生成名为 **speed_ch.map.rpt** 的报告文件。

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。可以对这些脚本进行设计，用以建立新工程或编译现有工程。还可以从命令提示符或控制台上运行可执行文件。

第 15 页上的图 7 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 Analysis & Synthesis、进行布局布线、时序分析以及为 Quartus II 软件中的 **filtref** 教程设计生成编程文件的方法。如果已安装教程设计，则它位于 **/<Quartus II 系统目录>/qdesigns/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 **/<Quartus II 系统目录>/qdesigns/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 7 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 系统目录>/bin 目录（或 UNIX 或 Linux 工作站上的
/<Quartus II 系统目录>/<平台> 目录，其中 <平台> 可以为 **solaris**、
linux 或 **hp_II**）。

图 7. 命令行脚本示例

<code>quartus_map filtref --family=Stratix</code>	建立针对 Stratix 器件系列的新 Quartus II 工程
<code>quartus_fit filtref --part=EP1S10F780C5 --fmax=80MHz --tsu=8ns</code>	为 EP1S10F780C5 器件执行布局布线并制定全局时序要求
<code>quartus_tan filtref</code>	进行时序分析
<code>quartus_asm filtref</code>	生成编程文件

图 8 是一个用于 UNIX 工作站的 **quartus_sh** 命令行脚本样本的摘录。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，并报告任何含语法错误的文件。

图 8. 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`
do
    quartus_map fir_filter --csf=filtref.csf --analyze_file=$filename

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


图 8. UNIX 命令行 Shell 脚本示例（第 2 部分，共 2 部分）

```
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
```

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 命令或建立和运行 Tcl 脚本，执行 Quartus II 工程中的以下任务。Tcl API 函数包括以下类别：

- 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
- 仿真器功能
- 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 Quartus II API for Tcl 中的命令建立 Tcl 脚本。应将 Tcl 脚本另存为 Tcl 脚本文件 (.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 命令使用相同的语法。若要将某现有工程用作另一个工程的基础，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 窗口中或在 **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 命令的帮助信息

Quartus II 软件包括 Quartus II 命令行可执行文件和 Tcl API 帮助浏览器，此浏览器是基于 Tcl 和 Tk 的 GUI，用于浏览命令和 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 ←
```

图 9 是一个 Tcl 脚本示例。

图 9. Tcl 脚本示例（第 1 部分，共 2 部分）

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

#----- Get Actual Fmax data from the Report File -----#

proc get_fmax_from_report {} {
#-----#
    global project_name

    # Load the project report database
    load_report $project_name

    # Find the "Timing Analyzer Summary" panel name containing
    # the Actual Fmax data by traversing the panel names
    # Then set the panel row containing the Actual Fmax
    # information

    set fmax_panel_name "Timing Analyzer Summary"
    foreach panel_name [get_report_panel_names] {
        if { [string match "$fmax_panel_name" "$panel_name"] } {
            # Fmax is sorted so we just need to go to Row 1
            set fmax_row [get_report_panel_row "$panel_name" -row 1]
        }
    }
}
```

图 9. Tcl 脚本示例（第 2 部分，共 2 部分）

```
# Actual Fmax is found on the fourth column
# Index starts at 0
set actual_fmax [lindex $fmax_row 3]

# Now unload the project report database
unload_report $project_name

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 global assignments -----#
set_global_assignment -name family STRATIX
set_global_assignment -name device EP1S10F484C5
set_global_assignment -name fmax_requirement $required_fmax
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 -to clock Pin_M20

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

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

建立 Makefile 脚本

Quartus II 软件支持使用 Quartus II 可执行文件的 Makefile 脚本，此脚本用于将您的脚本与各种脚本语言相融合。图 10 是标准 Makefile 脚本的摘录。

图 10. Makefile 脚本摘录 (第 1 部分, 共 2 部分)

```
#####
# Project Configuration:
#
# Specify the name of the design (project) and Compiler Settings
# File (.csf) 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.quartus chiptrip.psf chiptrip.csf

#####
# Main Targets
#
# all: build everything
# clean: remove output files and database
# clean_all: removes settings files as well as clean.
#####

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

clean:
    rm -rf *.rpt *.chg smart.log *.htm *.eqn *.pin *.sof *.pof db
clean_all: clean
    rm -rf *.ssf *.csf *.esf *.fsf *.psf *.quartus *.qws

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 =
```

图 10. Makefile 脚本摘录 (第 2 部分, 共 2 部分)

```
#####  
# 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 --tcl_eval project_new $(PROJECT) -overwrite  
  
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”

Altera 网站上的 *Application Note 309 (Command-Line Scripting in the Quartus II Software)*

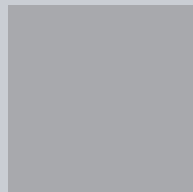
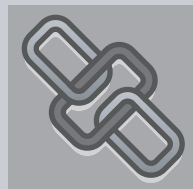
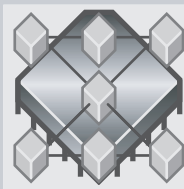
Tcl 命令和 Tcl 脚本

《Quartus II Help》中的“Overview: Using Tcl from the User Interface”、“Overview: Using Tcl Scripting”和“API Functions for Tcl”

Altera 网站上的 *Application Note 195 (Scripting with Tcl in the Quartus II Software)*

第二章

设计输入



第 2 章内容：

简介	24
建立工程	25
建立设计	26
使用 Altera 宏功能模块	30
指定初始设计的约束条件	37
设计方法与设计规划	40

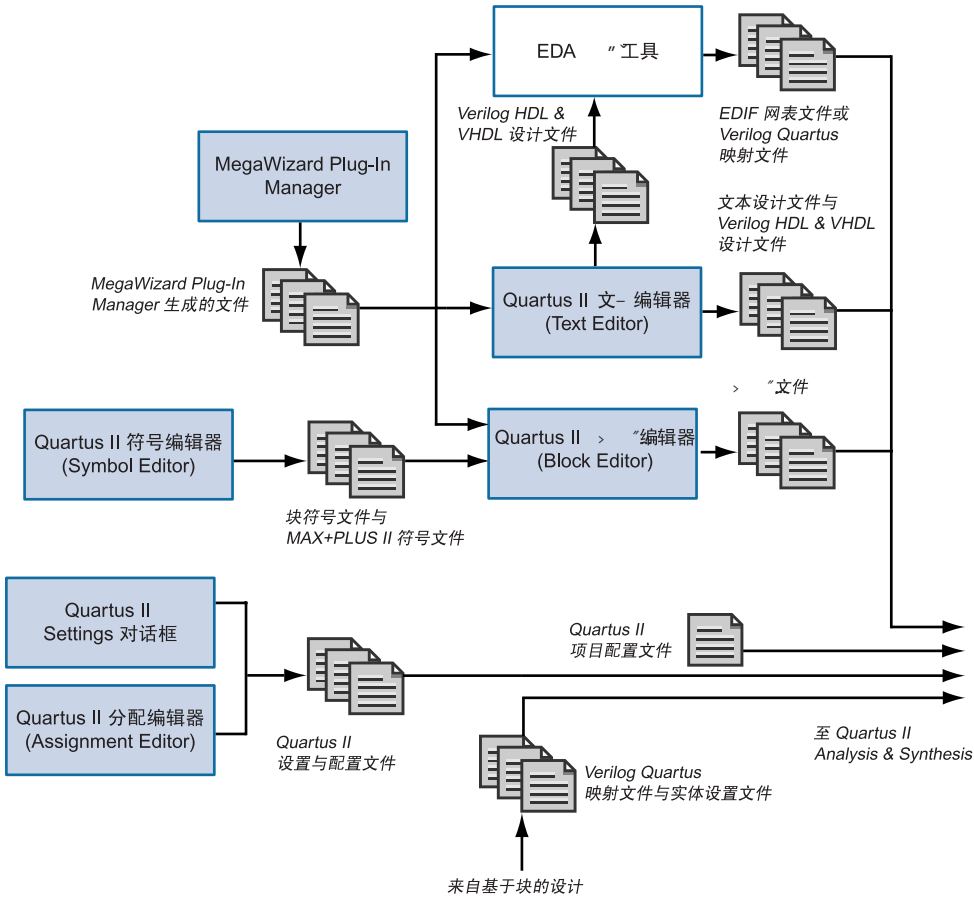
2

简介



Quartus® II 软件中的工程由所有设计文件和与设计有关的设置组成。您可以使用 Quartus II Block Editor、Text Editor、MegaWizard® Plug-In Manager (Tools 菜单) 和 EDA 设计输入工具建立包括 Altera® 宏功能模块、参数化模块库 (LPM) 函数和知识产权 (IP) 函数在内的设计。可以使用 Settings 对话框 (Assignments 菜单) 和 Assignment Editor 设定初始设计约束条件。图 1 显示设计输入流程。

图 1. 设计输入流程



建立工程

Quartus II 软件将工程信息存储在 Quartus II 工程配置文件 (.quartus) 中。它包含有关 Quartus II 工程的所有信息，包括设计文件、波形文件、SignalTap® II 文件、内存初始化文件以及构成工程的编译器、仿真器和软件构建设置。可以使用 **New Project Wizard**（File 菜单）或 **quartus_map** 可执行文件建立新工程。

使用 **New Project Wizard**，可以为工程指定工作目录、分配工程名称以及指定最高层设计实体的名称。还可以指定要在工程中使用的的设计文件、其它源文件、用户库和 EDA 工具，以及目标器件系列和器件（也可以让 Quartus II 软件自动选择器件）。

建立工程后，可以使用 **Settings** 对话框（Assignments 菜单）的 **Add/Remove** 页在工程中添加和删除设计和其它文件。在执行 Quartus II Analysis & Synthesis 期间，Quartus II 软件将按 **Add/Remove** 页中显示的顺序处理文件。

使用 quartus_map 可执行文件

可以在命令提示符下单独使用 Analysis & Synthesis，并使用 **quartus_map** 可执行文件建立新工程或将文件添加至工程中。

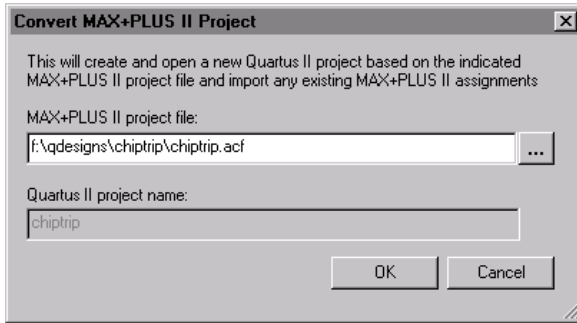
quartus_map 可执行文件可以建立单独的文本型报告文件，此报告文件可以使用任何文本编辑器查看。

若要获取有关 **quartus_map** 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_map -h ↵  
quartus_map -help ↵  
quartus_map --help=<主题名称> ↵
```

如果现有 MAX+PLUS® II 工程，还可以使用 **Convert MAX+PLUS II Project** 命令（File 菜单）将 MAX+PLUS II 分配与配置文件 (.acf) 转换为 Quartus II 工程。Quartus II 软件为工程建立新的 Quartus II 工程配置文件和有关的设置和配置文件。请参见第 26 页上的图 2。

图 2. 转换 MAX+PLUS II Project 对话框



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 设计语言建立设计。

Quartus II 软件还支持采用 EDA 设计输入和综合工具生成的 EDIF 输入文件 (.edf) 或 VQM 文件 (.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，以及生成 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 3 章：合成”中的“第 48 页使用其它 EDA 合成工具”。

可以使用以下设计文件类型在 Quartus II 软件或 EDA 设计输入工具中建立设计。

表 1. 支持的设计文件类型（第 1 部分，共 2 部分）

类型	描述	扩展名
原理图设计文件	使用 Quartus II Block Editor 建立的原理图设计文件。	.bdf
EDIF 输入文件	使用任何标准 EDIF 网表编写程序生成的 2.0.0 版 EDIF 网表文件。	.edf .edif
图形设计文件	使用 MAX+PLUS II Graphic Editor 建立的原理图设计文件。	.gdf

表 1. 支持的设计文件类型（第 2 部分，共 2 部分）

类型	描述	扩展名
文本设计文件	以 Altera 硬件描述语言 (AHDL) 编写的设计文件。	.tdf
Verilog 设计文件	包含使用 Verilog HDL 定义的设计逻辑的设计文件。	.v .vlg .verilog
VHDL 设计文件	包含使用 VHDL 定义的设计逻辑的设计文件。	.vh .vhd .vhdl
VQM 文件	通过 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 用于以原理图和流程图的形式输入和编辑图形设计信息。Quartus II Block Editor 读取并编辑原理图设计文件和 MAX+PLUS II 图形设计文件。可以在 Quartus II 软件中打开图形设计文件并将其另存为原理图设计文件。

每个原理图设计文件包含块和符号，这些块和符号代表设计中的逻辑，Block Editor 将每个流程图、原理图或符号代表的设计逻辑融合到工程中。

可以用原理图设计文件中的块建立新设计文件，可以在修改块和符号时更新设计文件，也可以在原理图设计文件的基础上生成块符号文件 (.bsf)、AHDL 包含文件 (.inc) 和 HDL 文件。还可以在编译之前分析原理图设计文件是否出错。Block Editor 还提供有助于您在原理图设计文件中连接块和基本单元（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、颜色和屏幕元素、缩放以及不同的块和基本单元属性。

Block Editor 的以下功能可以帮助您在 Quartus II 软件中建立原理图设计文件：

- **对 Altera 提供的宏功能模块进行实例化：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 函数在内的宏功能模块。宏功能模块以原理图设计文件中的块表示。请参阅“第 32 页使用 MegaWizard Plug-In Manager”。
- **插入块和基本单元符号：**流程图使用称为块的矩形符号代表设计实体以及相应的已分配信号，在从上到下的设计中很有用。块是用代表相应信号流程的管道连接起来的。可以将流程图专用于代表您的设计，也可以将流程图与图形单元相结合。

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) 函数和其它宏功能模块。
- **从块或原理图设计文件建立文件：**若要层次化设计工程，可以在 Block Editor 中使用 **Create/Update** 命令（File 菜单）从原理图设计文件中的块开始建立其它原理图设计文件、AHDL 包含文件、Verilog HDL 和 VHDL 设计文件以及 Quartus II 块符号文件。还可以从原理图设计文件本身建立 Verilog 设计文件、VHDL 设计文件和块符号文件。

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 软件建立的那些文本文件。

还可以用 Text Editor 将任何 AHDL 语句或节段模板、Tcl 命令或任何支持的 VHDL 或 Verilog HDL 构造模板插入当前文件中。AHDL、VHDL 和 Verilog HDL 模板为您输入 HDL 语法提供了一个简便的方法，可以提高设计输入的速度和准确度。还可以获取有关所有 AHDL 元素、关键字和语句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 函数的库中选择。可以自定义这些块符号文件，然后将这些符号添加到使用 Block Editor 建立的原理图中。Symbol Editor 读取并编辑块符号文件和 MAX+PLUS II 符号文件 (.sym)，并将它们转存为块符号文件。

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文本设计文件、Verilog 设计文件和 VHDL 设计文件，并在分级设计中将这些文件与其它类型的设计文件相结合。



Verilog 设计文件和 VHDL 设计文件可以包含由 Quartus II 支持的构造的任意组合。它们还可以包含 Altera 提供的逻辑函数，包括基本单元和宏功能模块以及用户自定义的逻辑函数。



在 Text Editor 中，使用 **Create/Update** 命令（File 菜单）从当前的 Verilog HDL 或 VHDL 设计文件建立块符号文件，然后将其合并到原理图设计文件中。同样，可以建立代表 Verilog HDL 或 VHDL 设计文件的 AHDL 包含文件，并将其合并到文本设计文件中或另一个 Verilog HDL 或 VHDL 设计文件中。

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 3 章：合成”中的“第 45 页使用 Quartus II VHDL 及 Verilog HDL 综合合成”。



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函数，并完全支持 LPM 函数。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》中的设计输入模块

使用 Altera 宏功能模块



Altera 宏功能模块是复杂或高级构建模块，可以在 Quartus II 设计文件中与门和触发器基本单元一起使用。Altera 提供的可参数化宏功能模块和 LPM 函数均为 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 特定器件的功能，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 函数和 IP 函数，用于 Quartus II 软件和 EDA 设计输入与综合工具中的设计。

表 2. Altera 提供的宏功能模块与 LPM 函数

类型	描述
算术组件	包括累加器、加法器、乘法器和 LPM 算术函数。
门	包括多路复用器和 LPM 门函数。
I/O 组件	包括时钟数据恢复 (CDR)、锁相环 (PLL)、双数据速率 (DDR)、千兆位收发器块 (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、移位寄存器宏模块和 LPM 存储器函数。

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编码。此外，这些函数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伸缩为不同的大小。Altera 还为宏功能模块和 LPM 函数提供 AHDL 包含文件和 VHDL 组件申明。

使用知识产权 (IP) 函数

Altera 提供多种方法来获取 Altera Megafunction Partners Program (AMPP™) 和 MegaCore® 宏功能模块，这些函数经严格的测试和优化，可以在 Altera 特定器件结构中发挥出最佳性能。可以使用这些知识产权的参数化块减少设计和测试时间。MegaCore 和 AMPP 宏功能模块包括应用于通信、数字信号处理 (DSP)、PCI 和其它总线界面以及存储器控制器中的宏功能模块。

使用 OpenCore® 和 OpenCore Plus 功能，可以在获得使用许可和购买之前免费下载和评估 AMPP 和 MegaCore 函数。

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

- **AMPP 程序：**AMPP 程序可以支持第三方供应商，以便建立和分布与 Quartus II 软件配用的宏功能模块。AMPP 合作伙伴提供一系列为 Altera 器件实行优化的现成宏功能模块。

AMPP 函数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 函数。

- **MegaCore 函数：**MegaCore 函数是用于复杂系统级函数的预验证 HDL 设计文件，并且可以使用 **MegaWizard Plug-In Manager** 进行完全参数化。MegaCore 函数由多个不同的设计文件组成：用于实施设计的综合后 AHDL 包含文件和为使用 EDA 仿真工具进行设计和调试而提供的 VHDL 或 Verilog HDL 功能仿真模型。

MegaCore 函数通过 Altera 网站上的 IP MegaStore 提供，或通过 MegaWizard Portal Extension 用于 **MegaWizard Plug-In Manager** 来提供。评估 MegaCore 函数无需许可，而且对评估没有时间限制。

- **OpenCore 评估功能：**OpenCore 宏功能模块是通过 OpenCore 评估功能获取的 MegaCore 函数。Altera OpenCore 功能允许您在采购之前评估 AMPP 和 MegaCore 函数。可以使用 OpenCore 功能编译、仿真设计并验证设计的功能和性能，但不支持编程文件的生成。
- **OpenCore Plus 硬件评估功能：**OpenCore Plus 评估功能通过支持免费 RTL 仿真和硬件评估来增强 OpenCore 评估功能。RTL 仿真支持用于在设计中仿真 MegaCore 函数的 RTL 模型。硬件评估支持用于为包括 Altera MegaCore 函数的设计生成时限编程文件。可以在决定购买 MegaCore 函数的许可之前使用这些文件，进行板级设计验证。

OpenCore Plus 功能支持的 MegaCore 函数包括标准 OpenCore 版本和 OpenCore Plus 版本。OpenCore Plus 许可用于生成时限编程文件，但不生成输出网表文件。

使用 MegaWizard Plug-In Manager

MegaWizard Plug-In Manager 可以帮助您建立或修改包含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这些文件进行实例化。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 函数。**MegaWizard Plug-In Manager** 运行一个向导，帮助您轻松地自定义宏功能模块变量指定选项。该向导用于为参数和可选端口设置数值。可以从 Tools 菜单或从原理图设计文件中打开 **MegaWizard Plug-In Manager**，也可以将它作为独立实用程序来运行。表 3 列出 **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 3. MegaWizard Plug-In Manager 生成的文件

文件名称	描述
< 输出文件 >.bsf	Block Editor 中使用的宏功能模块的符号。
< 输出文件 >.cmp	组件申明文件。
< 输出文件 >.inc	宏功能模块包装文件中模块的 AHDL 包含文件。
< 输出文件 >.tdf	要在 AHDL 设计中实例化的宏功能模块包装文件。
< 输出文件 >.vhd	要在 VHDL 设计中实例化的宏功能模块包装文件。
< 输出文件 >.v	要在 Verilog HDL 设计中实例化的宏功能模块包装文件。
< 输出文件 >_bb.v	Verilog HDL 设计所用宏功能模块包装文件中模块的空体或 black-box 申明，用于在使用 EDA 综合工具时指定端口方向。
< 输出文件 >_inst.tdf	宏功能模块包装文件中子设计的 AHDL 实例化示例。
< 输出文件 >_inst.vhd	宏功能模块包装文件中实体的 VHDL 实例化示例。
< 输出文件 >_inst.v	宏功能模块包装文件中模块的 Verilog HDL 实例化示例。



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 软件中对宏功能模块进行实例化

可以在 Block Editor 中直接实例化、在 HDL 代码中实例化（通过端口和参数定义实例化或使用 **MegaWizard Plug-In Manager** 对宏功能模块进行参数化并建立包装文件），也可以通过界面，在 Quartus II 软件中对 Altera 宏功能模块和 LPM 函数进行实例化。

Altera 建议您使用 **MegaWizard Plug-In Manager** 对宏功能模块进行实例化以及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 中实例化

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 **MegaWizard Plug-In Manager** 建立包含宏功能模块实例的 Verilog HDL 或 VHDL 包装文件，然后，您可以在设计中使用此文件。对于 VHDL 宏功能模块，**MegaWizard Plug-In Manager** 还建立组件申明文件。

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方法调用函数，直接在 Verilog HDL 或 VHDL 设计中对宏功能模块进行实例化。在 VHDL 中，还需要使用组件申明。

推断宏功能模块

Quartus II Analysis & Synthesis 可以自动识别某些类型的 HDL 代码和推断相应的宏功能模块。由于 Altera 宏功能模块已对 Altera 器件实行优化，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。对于一些体系结构特定的功能，例如 RAM 和 DSP 块，必须使用 Altera 宏功能模块。

Quartus II 软件在综合期间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 工具中实例化宏功能模块



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 函数和 IP 函数。可以通过为函数建立 black-box、通过推断或通过使用 clear-box 方法在 EDA 工具中实例化宏功能模块。

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 包装文件。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 包装文件包含参数化函数的端口和参数，可以将其用在最高层设计文件中实例化宏功能模块，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 函数建立 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 函数。
2. 使用 **MegaWizard Plug-In Manager** 生成的 black-box 文件在 EDA 综合工具中实例化函数。
3. 在 EDA 综合工具中进行设计的综合和优化。EDA 综合工具在综合期间将宏功能模块作为 black-box 处理。

按推断进行实例化

EDA 综合工具可自动识别某些类型的 HDL 代码和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实例化。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 函数作为 black-box 处理。因此，EDA 综合工具不能使用 Altera 宏功能模块完全综合和优化设计，因为此工具没有该函数的完整模型或时序信息。使用 clear-box 流程，用户可以使用 **MegaWizard Plug-In Manager** 建立用于 EDA 综合工具的完全可综合 Altera 宏功能模块或 LPM 函数。

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 函数。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 综合工具中对函数进行实例化。
3. 在 EDA 综合工具中进行设计的综合和优化。

由于 clear-box 宏功能模块或 LPM 函数包括较详细的信息（使用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 仿真器没有影响）。此外，clear-box 宏功能模块或 LPM 函数中包括特定的器件详细信息，因此要为设计使用不同的器件，需要为新器件重新生成 clear-box 函数。



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 函数，请参阅 IP 文档。有关 Altera 宏功能模块的信息，请参《Quartus II Help》。

在 EDA 工具中使用 Altera 提供的宏功能模块和 LPM 函数

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

Altera 网站上的 *Application Note 226 (Synplify & Quartus II Design Methodology)*

Altera 网站上的 *应用注解 225 (LeonardoSpectrum 及 Quartus II 设计方法)*

Altera 网站上的 *Application Note 222 (Using Precision RTL Synthesis in the Quartus II Design Methodology)*

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

《Quartus II Tutorial》中的设计输入模块

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

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

MegaCore、OpenCore 和 OpenCore Plus 功能及函数

Altera 网站上的 *Application Note 125 (Evaluating AMPP & MegaCore Functions)*

Altera 网站上的 *Application Note 176 (OpenCore Plus Hardware Evaluation)*

指定初始设计的约束条件



建立工程和设计之后，可以使用 Quartus II 软件中的 **Settings** 对话框、**Assignment Editor** 和 **Floorplan Editor** 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。Quartus II 软件还提供 **Compiler Settings** 向导（**Assignments** 菜单）和 **定时设置向导**（**Assignments** 菜单），协助用户指定初始设计的约束条件。

使用 Assignment Editor

Assignment Editor 是用于在 Quartus II 软件中建立和编辑分配的界面。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** 选择要分配的特定节点和实体；显示有关特定分配的信息；添加、编辑或删除选定节点的分配。还可以向分配添加备注，可以查看出现分配的设置和配置文件。

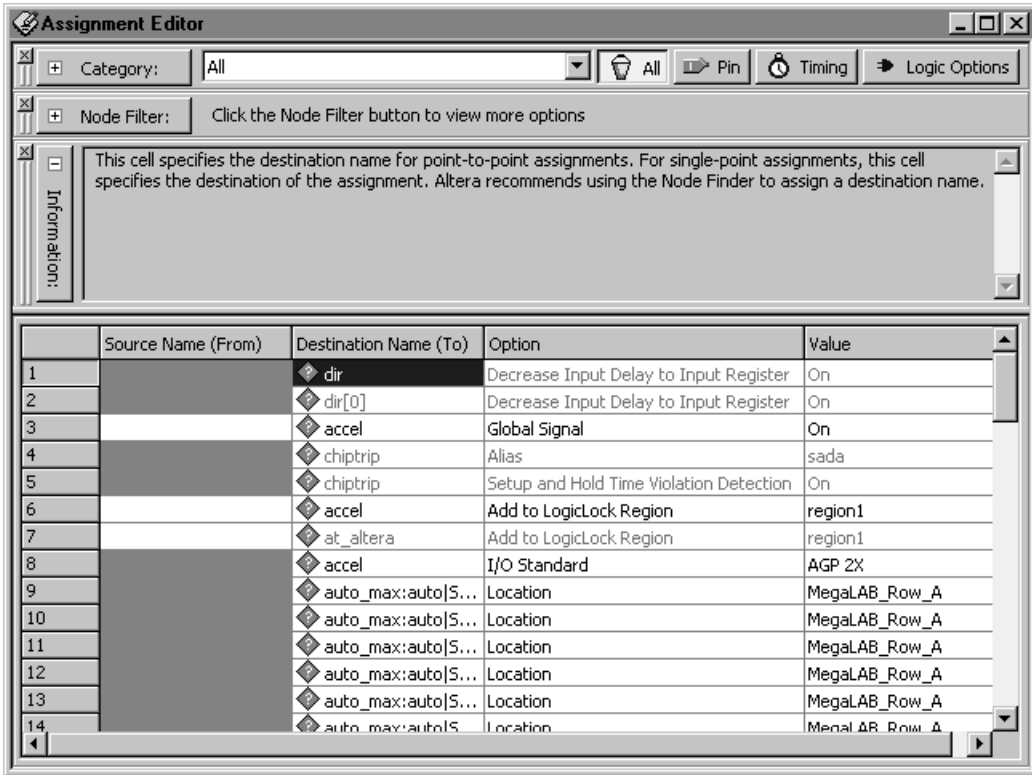
以下步骤描述使用 **Assignment Editor** 进行分配的基本流程：

1. 打开 **Assignment Editor**。
2. 在 **Category** 栏中选择相应的类别分配。
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，或允许您键入分配信息。当您添加、编辑和删除分配时，消息窗口中将出现相应的 Tcl 命令。还可以将数据从 **Assignment Editor** 导出到 Tcl 脚本文件 (.tcl) 或与电子表格兼容的文件中。

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。如果分配或分配值无效，Quartus II 软件不会添加或更新数值，改为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 3. Quartus II Assignment Editor



使用 Settings 对话框

可以使用 **Settings** 对话框（Assignments 菜单）进行编译器、仿真器和软件构件设置、时序设置以及修改工程设置。

使用 **Settings** 对话框可以执行以下类型的任务：

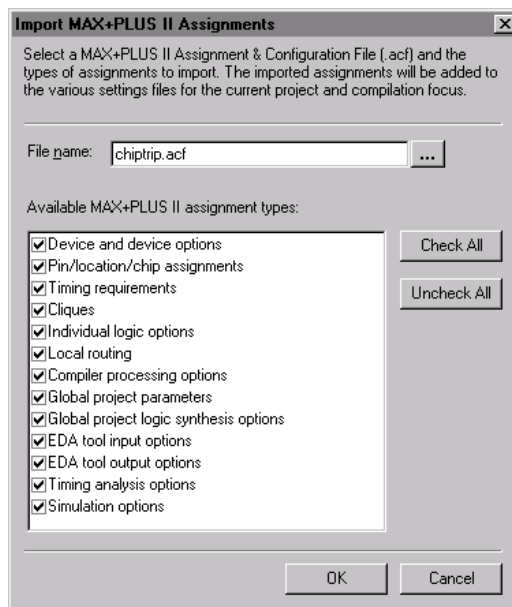
- **修改工程设置：** 在工程中添加和删除文件，指定自定义用户库、工具集目录、EDA 工具设置、默认逻辑选项和参数设置。
- **指定 HDL 设置：** Verilog HDL 和 VHDL 语言版本以及库映射文件 (.lmf)。
- **指时序设置：** 为工程设置默认频率或定义各时钟的设置、延时要求和路径切割选项以及时序分析报告选项。

- **指定编译器设置：** 引脚分配（通过 **Assign Pins** 对话框）、器件选项（封装、引脚计数、速度等级）、迁移器件、编译器注意项、模式、布局布线和综合选项、SignalTap® II 设置、Design Assistant 设置和网表优化选项。
- **指定仿真器设置：** 仿真注意项、模式（功能或时序）以及时间和波形文件选项。
- **指定软件构建设置：** 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 HardCopy 时序设置：** HardCopy™ 时序选项并生成 HardCopy 文件。

导入分配

Quartus II 软件用于将包含 MAX+PLUS II 工程分配和设置的 MAX+PLUS II 分配与配置文件 (.acf) 方便地导入您的 Quartus II 工程中。可以使用 **Import MAX+PLUS II Assignments** 命令（Assignments 菜单）导入特定类型的分配，并将其添加到 Quartus II 软件的设置和配置文件中。还可以使用 Tcl 命令或脚本从其它 EDA 综合工具中导入分配。请参阅图 4。

图 4. Import MAX+PLUS II Assignments 对话框



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 — 位置、I/O 库和 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，实现更快地建立最终引出脚。无需设计文件便可使用此命令，并且可以在设计编译完成之前验证引出脚。

设计方法与设计规划

在建立新设计时，必须考虑 Quartus II 软件提供的设计方法。例如，LogicLock™ 功能提供从上到下或从下到上的设计方法，以及基于块的设计流程。不管您有没有使用 EDA 设计输入和综合工具，都可以使用这些设计流程。

从上到下与从下到上的设计方法比较

在从上到下的设计流程中，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等级结构的优化处理，而且管理起来也比较容易。

在从下到上的设计方法中，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，并在每个模块上应用不同的优化技巧。对单个模块的修改不会影响其它模块的优化。从下到上的设计还有助于在其它设计中重新使用设计模块。

基于块的设计流程

在基于块的从下到上 LogicLock 设计流程中，可以独立设计和优化每个模块，在最高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最高层设计中。在最高层设计中，每个模块都不影响其它模块的性能。一般基于块的设计流程可以在模块化、分级、递增和协作式设计流程中使用。

可以在基于块的设计流程中使用 EDA 设计输入和综合工具，分别设计和综合各个模块，然后将各模块整合到 Quartus II 软件的最高层设计中，也可以在 EDA 设计输入和综合工具中完整设计和综合基于块的设计。

设计分割

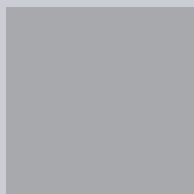
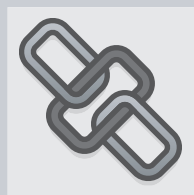
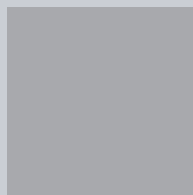
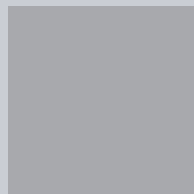
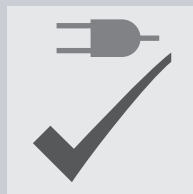
在 Quartus II 软件或其它 EDA 工具中建立分级设计时，设计被分割为单独的模块。在设计规划期间分割设计时需做以下考虑：

- 分割设计的位置
- 分割之间的时钟数和 I/O 连接
- 状态机的放置
- 非关键函数与时序关键函数的分离
- 在分级模块中限制关键路径
- 登记各个模块的输入和输出

有关使用 LogicLock 功能和基于原理图设计的详细信息，请参阅第 85 页中的“[第 6 章：基于块的设计](#)”。

第三章

综合



第 3 章内容：

简介	44
使用 Quartus II VHDL 及 Verilog HDL 集成综合	45
使用其它 EDA 综合工具	48
控制 Analysis & Synthesis	50
使用 Design Assistant 检查设计可靠性	52

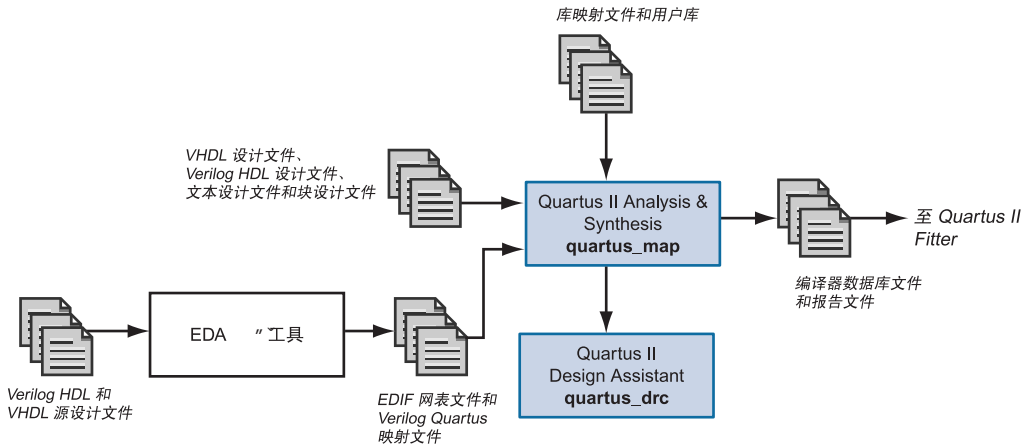
3

简介



可以使用 Compiler 的 Quartus® II Analysis & Synthesis 模块分析设计文件和建立工程数据库。Analysis & Synthesis 使用 Quartus II Integrated Synthesis 综合 VHDL 设计文件 (.vhd) 或 Verilog 设计文件 (.v)。用户喜欢的话，可以使用其它 EDA 综合工具综合 VHDL 或 Verilog HDL 设计文件，然后再生成可以与 Quartus II 软件配合使用的 EDIF 网表文件 (.edf) 或 VQM 文件 (.vqm)。图 1 显示了综合设计流程。

图 1. 综合设计流程



可以在包含 Analysis & Synthesis 模块的 Quartus II 软件中启动全编译，也可以单独启动 Analysis & Synthesis。Quartus II 软件还允许您在不运行 Integrated Synthesis 的情况下执行 Analysis & Elaboration。

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** 可执行文件将建立新工程。

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

若要获取有关 **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=<主题名称> ←
```

使用 Quartus II VHDL 及 Verilog HDL 集成综合

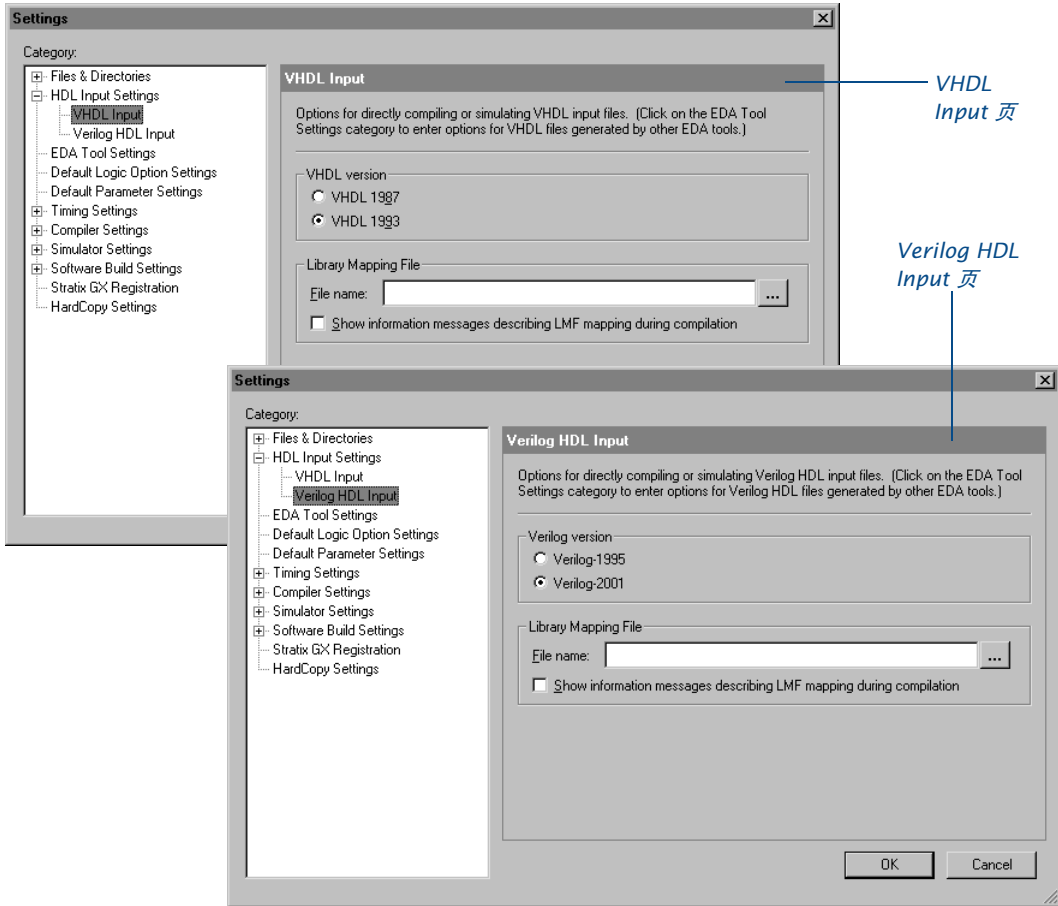


可以使用 Analysis & Synthesis 分析并综合 VHDL 和 Verilog HDL 设计。Analysis & Synthesis 包括 Quartus II Integrated Synthesis，它完全支持 VHDL 和 Verilog HDL 语言，并提供控制综合过程的选项。



Analysis & Synthesis 支持 Verilog-1995 标准（IEEE 标准 1364-1995）和大多数 Verilog-2001 标准（IEEE 标准 1364-2001）构造，还支持 VHDL 1987（IEEE 标准 1076-1987）和 1993（IEEE 标准 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。还可以指定 Quartus II 应用来将非 Quartus II 函数映射到 Quartus II 函数的库映射文件 (.lmf) 上。可以在 **Settings** 对话框（Assignments 菜单）的 **Verilog HDL Input** 和 **VHDL Input** 页中指定这些或其它选项，如第 46 页上的图 2 中所示。

图 2. Settings 对话框的 VHDL Input & Verilog HDL Input 页



尽管大多数的 VHDL 和 Verilog HDL 设计可以在 Quartus II Integrated Synthesis 和其它 EDA 综合工具中成功编译，但如果使用另一个 EDA 工具对这些函数进行实例化，则 Altera 宏功能模块、参数化模块库 (LPM) 函数和知识产权 (IP) 宏功能模块需要使用空体或 black-box 文件。但是，在您为 Quartus II Integrated Synthesis 实例化宏功能模块时，可以不使用 black-box 文件，而直接实例化宏功能模块。有关实例化宏功能模块详细信息，请参阅

“第 2 章：设计输入”中的“第 32 页在 Quartus II 软件中对宏功能模块进行实例化”和“第 33 页在 EDA 工具中实例化宏功能模块”。

当您建立 VHDL 或 Verilog HDL 设计时，您应该将它们添加至工程中。当您使用 **New Project Wizard**（File 菜单）或使用 **Settings** 对话框的 **Add/Remove** 页建立工程时，您可以添加设计文件；或者如果您在

Quartus II Text Editor 中编辑文件，在保存文件时，系统将提示您将其添加至当前工程中。在将文件添加至工程中时，应按您希望 **Integrated Synthesis** 处理这些文件的顺序来添加。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”中的“第 25 页建立工程”。

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 软件用此数据库进行其余工程处理。其它 **Compiler** 模块对该数据库进行更新，直到它包含完全优化的工程。开始时，该数据库仅包含原始网表；最后，它包含完全优化且合适的工程，工程将用于为时序仿真、时序分析、器件编程等建立一个或多个文件。

当它建立数据库时，**Analysis & Synthesis** 的分析阶段将检查工程的逻辑完整性和一致性，并检查边界连接和语法错误。

Analysis & Synthesis 还在设计实体或工程文件的逻辑上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。它为状态机建立状态分配，并作出能减少所用资源的选择。此外，它还用 **Altera** 参数化模块库 (LPM) 函数中的模块替换运算符，例如 + 或 -，而该函数已为 **Altera** 器件做了优化。

Analysis & Synthesis 使用多种算法来减少门的数量，删除冗余逻辑以及尽可能有效地利用器件体系结构。可以使用逻辑选项分配自定义综合。**Analysis & Synthesis** 还应用逻辑综合技术，以协助实施工程时序要求，并优化设计以满足这些要求。

消息窗口和 **Report** 窗口的消息区域显示 **Analysis & Synthesis** 生成的任何消息。**Status** 窗口记录工程编译期间在 **Analysis & Synthesis** 中处理所花的时间。



有关信息

请参阅

Quartus II 软件支持的 Verilog HDL 构造	《Quartus II Help》中的“Quartus II Verilog HDL Support”
Quartus II 软件中支持的 VHDL 构造	《Quartus II Help》中的“Quartus II VHDL Support”
使用 Quartus II Integrated Synthesis	Altera 网站上的 <i>Application Note 238 (Using Quartus II Verilog HDL & VHDL Integrated Synthesis)</i>

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，然后生成可以与 Quartus II 软件配合使用的 EDIF 网表文件或 VQM 文件。

Altera 提供与许多 EDA 综合工具配合使用的库。Altera 还为许多工具提供 NativeLink® 支持。NativeLink 技术有助于在 Quartus II 软件和其它 EDA 工具之间无缝传送信息，并允许您从 Quartus II 图形用户界面中自动运行 EDA 工具。

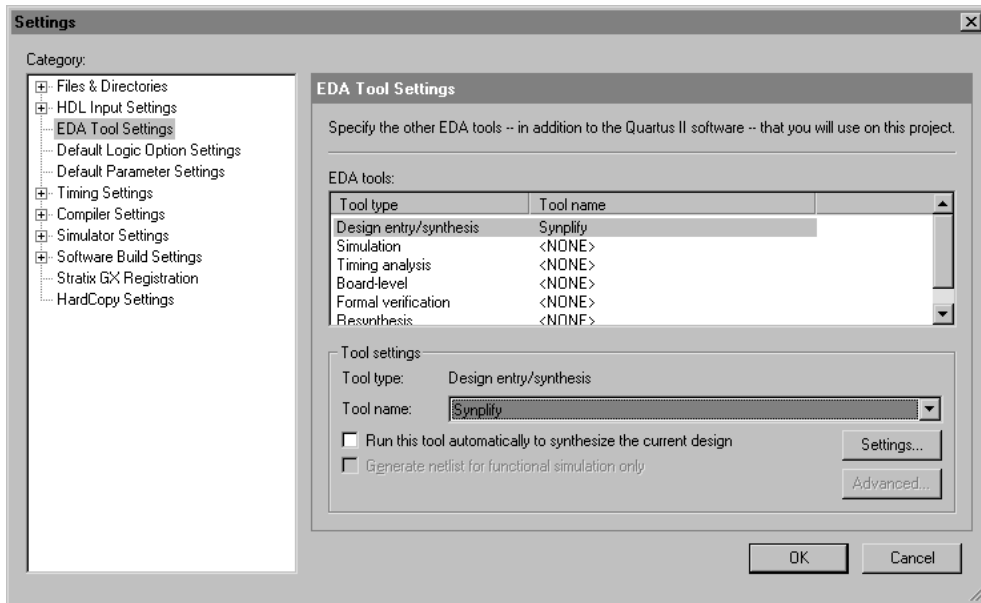
如果已使用其它 EDA 工具建立了分配或约束条件，可以使用 Tcl 命令或脚本将这些约束条件导入包含您的设计文件的 Quartus II 软件中。许多 EDA 工具可自动生成分配 Tcl 脚本。表 1 列出了 Quartus II 支持的 EDA 综合软件。

表 1. Quartus II 支持的 EDA 综合工具

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

可以在 **Settings** 对话框（Assignments 菜单）的 **EDA Tool Settings** 页中，指定是否应在 Quartus II 软件中自动运行具有 NativeLink 支持的 EDA 工具，并使它成为综合设计全编译的一部分。**EDA Tools Settings** 页还允许您为 EDA 工具指定其它选项。请参见图 3。

图 3. Settings 对话框的 EDA Tool Settings 页



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。有关详细信息，请参阅 EDA 工具文档。



有关信息

使用 Mentor Graphics LeonardoSpectrum

使用 Mentor Graphics Precision RTL Synthesis

使用 Synplcity Synplify

请参阅

Altera 网站上的 *Application Note 225 (LeonardoSpectrum & Quartus II Design Methodology)*

Altera 网站上的 *Application Note 168 (Getting Started with the LeonardoSpectrum Software)*

Altera 网站上的 *Application Note 222 (Using Precision RTL Synthesis in the Quartus II Design Methodology)*

Altera 网站上的 *Application Note 226 (Synplify & Quartus II Design Methodology)*

控制 Analysis & Synthesis

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

- 编译器指令和属性
- Quartus II 逻辑选项
- Quartus II 综合网表优化选项

使用编译器指令和属性

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，用于推动特定设计元素的综合过程。它还提供一些属性，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

《Quartus II Help》中的“VHDL Language Directives & Attributes”和“Verilog HDL Language Directives & Attributes”

结合 Quartus II Integrated Synthesis 使用编译器指令和属性

Altera 网站上的 *Application Note 238 (Using Quartus II Verilog HDL & VHDL 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。可以在 Assignment Editor 中指定 Quartus II 逻辑选项。Quartus II 逻辑选项用于保留寄存器、指定通电时的逻辑电平、删除重复或冗余的逻辑、优化速度或区域、控制扇出、设置状态机的编码级别以及控制其它许多选项。



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》中的编译模块
使用会影响综合的 Quartus II 综合选项和逻辑选项	Altera 网站上的 <i>Application Note 238 (Using Quartus II Verilog HDL & VHDL Integrated Synthesis)</i>

使用 Quartus II 综合网表优化选项

Quartus II 综合优化选项用于设置选项，在许多 Altera 器件系列的综合期间优化网表。这些优化选项对标准编译期间出现的优化进行补充，并且是在全编译的 Analysis & Synthesis 阶段出现。这些优化对综合网表进行更改，通常有利于区域和速度的改善。**Settings** 对话框（Assignments 菜单）的 **Netlist Optimizations** 页用于指定网表优化选项，其中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重新定时
- 允许寄存器重新定时，在 T_{su}/T_{co} 和 F_{max} 之间进行取舍

Netlist Optimizations 页还包括 Fitter 网表优化和物理综合选项。有关综合优化选项、Fitter 网表优化选项和物理综合的详细信息，请参阅“第 8 章：定时闭合”中的“第 113 页使用网表优化实现定时闭合”。

还可以在 **Settings** 对话框的 **Synthesis** 页中，指定是否要 Analysis & Synthesis 将综合结果保存至 VQM 文件。



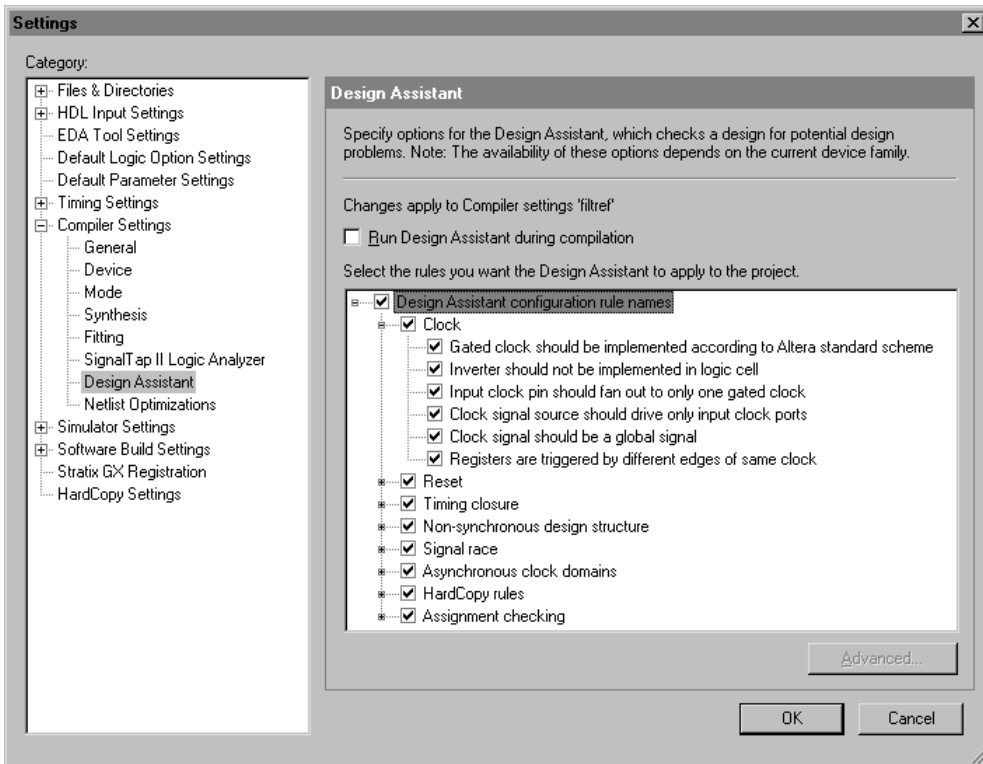
有关信息	请参阅
使用 Quartus II 综合和网表优化选项	Altera 网站上的 <i>Application Note 198 (Timing Closure with the Quartus II Software)</i>
	Altera 网站上的 <i>Application Note 297 (Optimizing FPGA Performance Using the Quartus II Software)</i>

使用 Design Assistant 检查设计可靠性



Quartus II Design Assistant 允许您依据一组设计规则，检查设计的可靠性。在为 HardCopy™ 器件进行迁移设计之前，在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** 页用于指定检查设计时要使用的设计可靠性准则。请参阅图 4。

图 4. Settings 对话框的 Design Assistant 页



 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的文本型报告文件。

若要获取有关 **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=<主题名称> ←
```

还可以遵循较好的同步设计做法和遵照 Quartus II 编码样式准则，改进设计优化。



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”

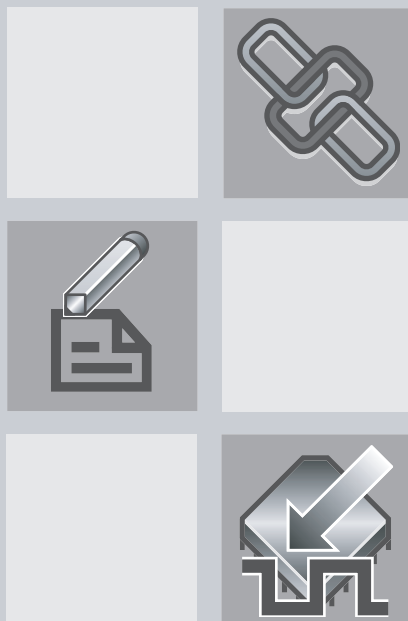
使用 Quartus II 综合选项，遵循同步设计做法和编码样式准则

Altera 网站上的 *Application Note 238 (Using Quartus II Verilog HDL & VHDL Integrated Synthesis)*

《Quartus II Help》中的 “AHDL, VHDL, and Verilog HDL Style Guide”

第四章

仿真



第 4 章内容：

简介	56
使用 EDA 工具进行设计仿真	57
使用 Quartus II 仿真器进行仿真设计	64
仿真 Excalibur 设计	66

4

简介



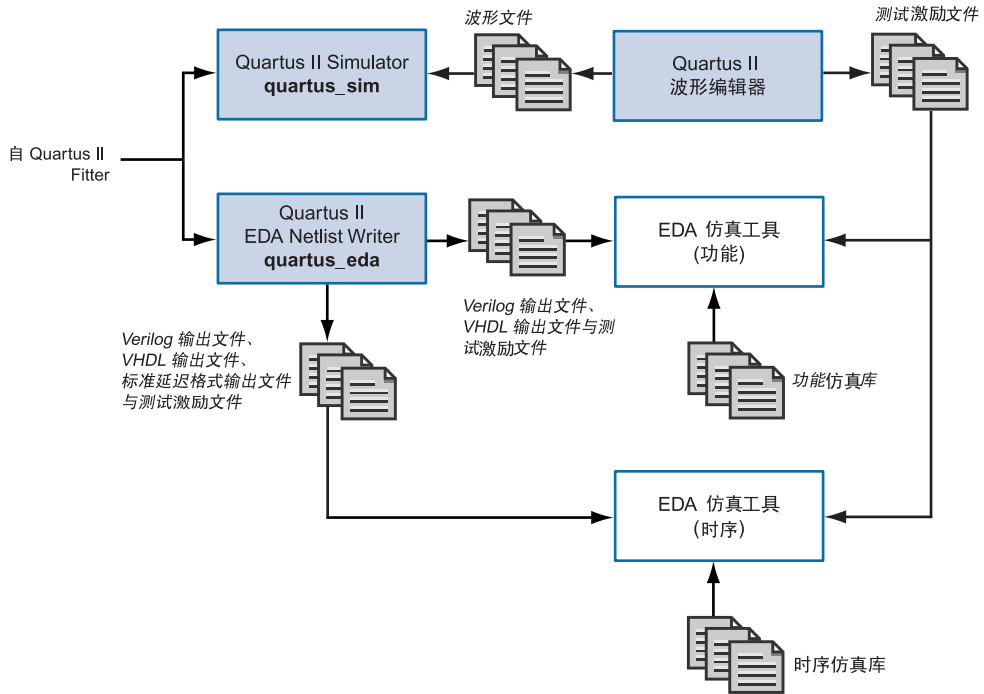
可以使用 EDA 仿真工具或使用 Quartus® II 仿真器进行设计的功能与时序仿真。

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
- PowerGauge™ 功耗估算
- 生成测试台模板和内存初始化文件

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

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 输出文件 (.vho) 和 Verilog 输出文件 (.vo)，以及使用 EDA 仿真工具进行时序仿真所需的标准延时格式输出文件 (.sdo)。Quartus II 软件在 Standard Delay Format 2.1 版中生成 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工具特定目录中。

此外，Quartus II 软件通过 NativeLink 功能使时序仿真与 EDA 仿真工具完美集成。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 NativeLink 功能支持的 EDA 仿真工具。

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	✓
Model Technology™ ModelSim®	✓
Model Technology ModelSim-Altera	✓
Synopsys Scirocco	✓
Synopsys VCS	
Synopsys VSS	

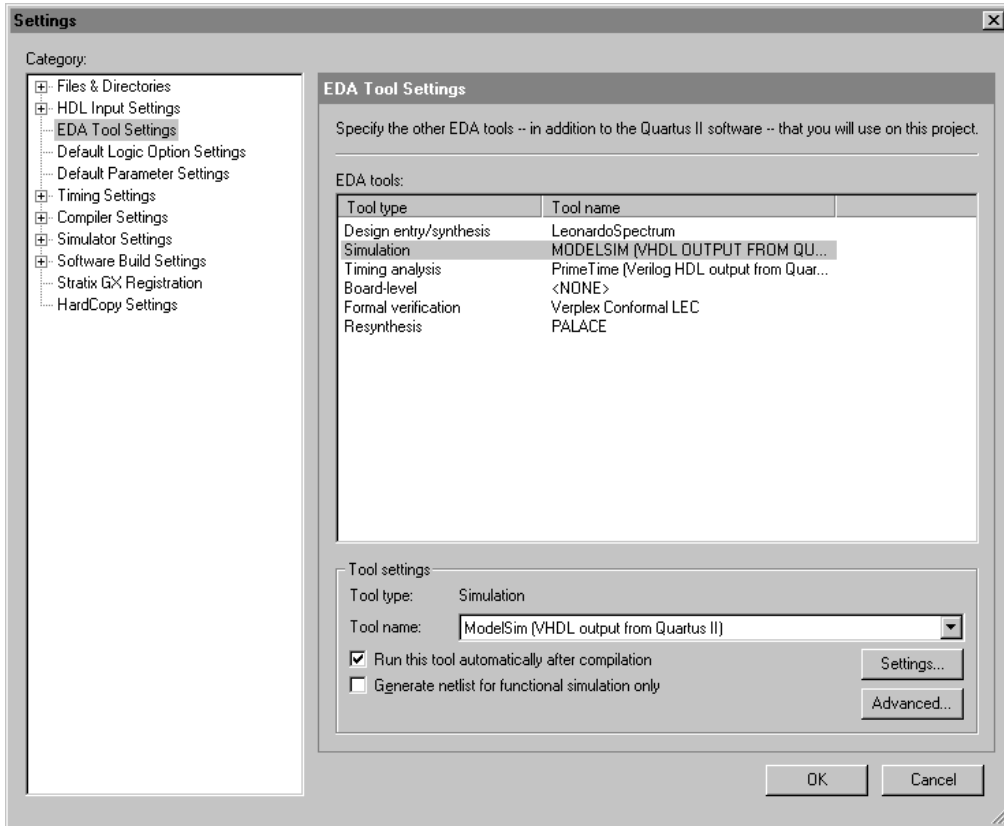
ModelSim-Altera 软件

在用于支持行为仿真和 HDL 测试台的 Altera® 设计软件订购中包括 Model Technology ModelSim-Altera 软件。

指定 EDA 仿真工具设置

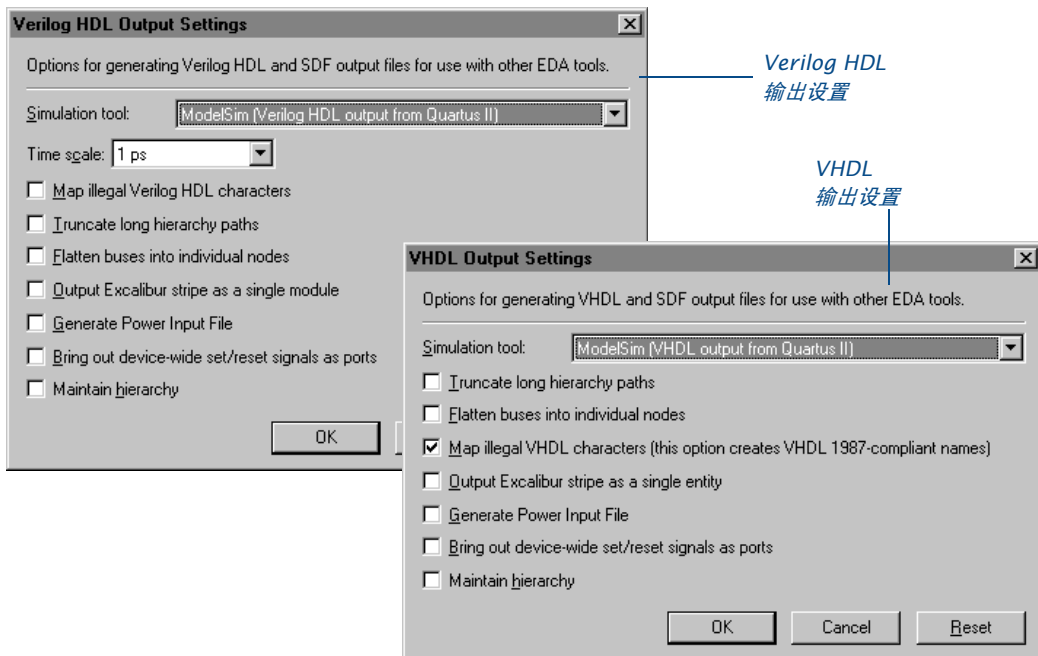
建立新工程时，可以从 **New Project Wizard**（File 名称）中或者从 **Settings** 对话框（Assignments 菜单）的 **EDA Tool Settings** 页中选择 EDA 仿真工具。图 2 显示了 **Settings** 对话框的 **EDA Tool Settings** 页。

图 2. EDA Tool Settings 页



可以在 **Verilog HDL/VHDL Output Settings** 对话框（通过 **EDA Tool Settings** 页进入）中指定用于生成 Verilog 和 VHDL 输出文件和相应 SDF 输出文件的其它选项。第 59 页上的图 3 显示了 **Verilog/VHDL Output Settings** 对话框。

图 3. Verilog HDL/VHDL Output Settings 对话框



生成仿真输出文件



可以运行 EDA Netlist Writer 模块，并通过指定 EDA 工具设置和编译设计，生成 Verilog 输出文件和 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 输出文件或 VHDL 输出文件。在使用 NativeLink 功能时，还可以通过使用 **Run EDA Simulation Tool** 和 **Run EDA Timing Analysis Tool** 命令（Tools 菜单）在初始编译之后运行仿真或时序分析工具。

 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单独的文本型报告文件。

若要获取有关 **quartus_eda** 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_eda -h ←  
quartus_eda --help ←  
quartus_eda --help=<主题名称> ←
```

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

- **功耗估算数据：**可以使用 ModelSim 或 ModelSim-Altera 软件进行包括功耗估算数据在内的仿真。可以指示 Quartus II 软件将 Verilog HDL 或 VHDL 输出文件中设计的功耗估算数据包括在内。ModelSim 软件生成功率输入文件 (**.pwf**)，此文件用于在 Quartus II 软件中估计设计功耗。
- **测试激励文件：**可以从 Quartus II Waveform Editor 的矢量波形文件 (**.vwf**) 建立 Verilog 测试激励文件 (**.vt**) 和 VHDL 测试激励文件 (**.vht**)。Verilog HDL 和 VHDL 测试激励文件是测试台模板文件，其中包含最高层设计文件的实例化和来自矢量波形文件的测试矢量。如果在矢量波形文件中指定预期值，还可以生成自检测试激励文件。
- **存储器初始化文件：**可以使用 Quartus II Memory Editor 输入存储器模块的初始内容，例如，在存储器初始化文件 (**.mif**) 或十六进制 (Intel 格式) 文件 (**.hex**) 中输入可寻址内容存储器 (CAM)、RAM 或 ROM。然后，可以将存储器内容导出为 RAM 初始化文件 (**.rif**)，与 EDA 仿真工具一起用于功能仿真。

仿真流程

使用 NativeLink 功能，可以指示 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。此外，还可以在编译之前（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

可以在设计流程的任何位置上进行功能或行为仿真。以下描述使用 EDA 仿真工具进行设计的功能仿真时所采用的基本流程。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，请执行以下操作：

1. 在 EDA 仿真工具中设置工程。
2. 建立工作库。
3. 使用 EDA 仿真工具编译相应的功能仿真库。
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 行步骤设置，使 EDA 仿真工具可以在 Quartus II 软件中自动设置和运行。以下描述将 EDA 仿真工具与 NativeLink 功能配合使用的基本流程：

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目录；编译或映射到相应的库、设计文件和测试激励文件；设置仿真环境；以及运行仿真。

手动时序仿真流程

如果要加强仿真控制，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。以下描述使用 EDA 仿真工具执行 Quartus II 设计时序仿真所采用的基本流程。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工具特定的目录中。
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. 编译并映射到时序仿真目录，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 特定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于原子时序的仿真库。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 特定组件的任何设计进行功能或时序仿真。此外，Altera 为 ModelSim-Altera 软件中的仿真提供编译前功能与时序仿真库。

Altera 为使用 Altera 宏功能模块以及标准参数化模块库 (LPM) 函数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** 库的编译前版本。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	描述
220model.v 220model.vhd 220model_87.vhd	LPM 函数的仿真模型（2 2 0 版）
220pack.vhd	220model.vhd 的 VHDL 组件申明
altera_mf.v altera_mf.vhd altera_mf_87.vhd altera_mf_components.vhd	Altera 特定宏功能模块的仿真模型和 VHDL 组件申明

在 Quartus II 软件中，特定器件体系结构实体和 Altera 特定的宏功能模块的时序信息位基于原子时序的设置后仿真库中。视器件系列以及是否使用 Verilog 输出文件或 VHDL 输出文件而定，时序仿真库文件可能有所不同。对于 VHDL 设计，Altera 为具有 Altera 特定的参数化函数的设计提供 VHDL 组件申明文件。



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 网站上的 <i>Application Note 204 (Using ModelSim-Altera in a Quartus II Design Flow)</i>
使用 VCS 软件进行仿真	Altera 网站上的 <i>Application Note 239 (Using VCS with the Quartus II Software)</i>

使用 Quartus II 仿真器进行仿真设计



可以使用 Quartus II 仿真器在工程中仿真任何设计。视所需的信息类型而定，可以进行功能仿真以测试设计的逻辑运算，也可以进行时序仿真以在目标器件中测试设计的逻辑运算和最差时序。

Quartus II 软件可以仿真整个设计，或仿真设计的任何部分。可以在工程中将任何设计实体指定为仿真焦点。在仿真设计时，仿真器 仿真焦点实体及其所有附属设计实体。

指定 仿真器设置

通过建立 仿真器设置，指定要仿真的类型、仿真涵盖的时间段、激励向量以及其它仿真选项。可以使用 **Settings** 对话框（**Assignments** 菜单）或仿真器 **Settings Wizard**（**Processing** 菜单）建立 仿真器设置的自定义组，也可以使用每次建立新工程时自动生成的默认 仿真器设置。

进行仿真

以下步骤描述在 Quartus II 软件中进行功能或时序仿真的基本流程。

1. 指定 仿真器设置。
2. 建立并指定矢量源文件。
3. 使用 **Start > Start Simulation** 命令（**Processing** 菜单）或使用 **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仿真器。

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

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：

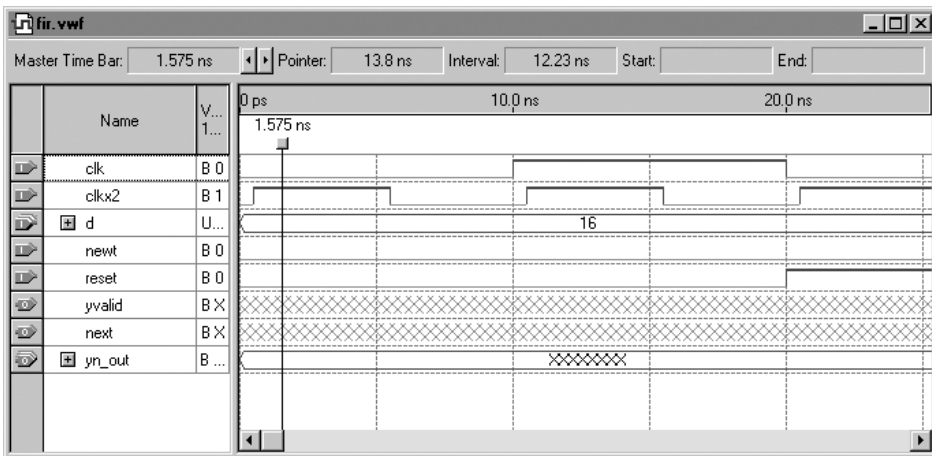
```
quartus_sim -h ←
quartus_sim --help ←
quartus_sim --help=<主题名称> ←
```

建立波形文件

Quartus II Waveform Editor 可以建立和编辑用于波形格式仿真的输入矢量。使用 Waveform Editor，可以将输入矢量添加到波形文件中，此文件描述设计中的逻辑行为。请参见第 65 页上的图 4。

Quartus II 软件支持矢量波形文件 (`.vwf`)、表文件 (`.tbl`)、矢量文件 (`.vec`) 和矢量表输出文件 (`.tbl`) 格式的波形文件。可以在 MAX+PLUS® II 软件中将 MAX+PLUS II 仿真器通道文件 (`.scf`) 另存为表文件，然后使用 Waveform Editor 打开表文件并另存为矢量波形文件。

图 4. Quartus II Waveform Editor



进行 PowerGauge 功耗估算

Quartus II 软件可以估计在时序仿真期间当前设计所消耗的功率。可以指示仿真器以毫瓦 (mW) 为单位计算和报告设计仿真期间所消耗的内功率、I/O 引脚功率和总功率。可以在 Report 窗口中查看 PowerGauge 功耗估算的结果。

仿真 Excalibur 设计

可以使用总线功能模型在 Quartus II 仿真器中进行 Excalibur™ 器件的功能仿真，也可以使用 EDA 仿真工具通过总线功能模型或全带区模型进行 Excalibur 器件的功能或时序仿真。还可以使用 EDA 仿真工具和软件调试器与 Excalibur Stripe 仿真器 (ESS) 模型联合进行功能、时序和硬件协同仿真。

在 Quartus II 软件中仿真 Excalibur 设计

总线功能模型仿效 Excalibur 器件的 Excalibur 嵌入式处理器带区中的 AMBA™ 高性能总线 (AHB) 的行为。它在带区到 PLD 的桥接器上通过带区主端口以及在 PLD 至带区的桥接器上通过带区从端口，仿真 Excalibur 嵌入式处理器带区和 PLD 之间的交互作用。

在编译和综合之前，可以在 Quartus II 软件中进行 Excalibur 设计的功能仿真。总线功能模型验证与 Excalibur 带区桥接器连接的 AHB 主端口或从端口的功能。必须先生成 uPCore 处理模型输入文件 (.mbus_in)。然后，可以使用 Quartus II 仿真器和总线功能模型对设计进行功能仿真，并生成包含总线处理的 uPCore 处理模型输出文件 (.mbus_out)。

以下步骤描述在 Quartus II 软件中对总线功能模型进行功能仿真的基本流程：

1. 建立主端口高级命令文件。
2. 使用 `exc_bus_translate` 实用程序建立 uPCore 处理模型输入文件。
3. 指定仿真器设置。必须在 **Settings** 对话框（**Assignments** 菜单）的 **Options** 页中指定 uPCore 处理模型输入文件的名称。
4. 运行仿真。Quartus II 仿真器仿真设计，并生成 uPCore 处理模型输出文件，该文件显示总线功能模型的仿真结果。

总线功能模型与 EDA 工具结合使用

可以使用总线功能模型通过 EDA 仿真工具进行功能或时序仿真。若要使用总线功能模型，通过其它 EDA 工具进行仿真，需要建立总线功能模型仿真文件，此文件包括带区至 PLD 以及 PLD 至带区的总线处理。

一旦生成了这些文件，就可以设置仿真工具；编译相应的库、设计文件和测试激励文件；以及运行仿真。在仿真期间，每个处理的主命令、地址和数据值将写入输出文件中。

全带区模型与 EDA 工具的结合使用

Excalibur 全带区模型是 Excalibur 嵌入式处理器带区的完整的寄存器传输级别 (RTL) 模型。它包括 Excalibur 嵌入式处理器内核和外围设备（例如，SDRAM 接口、扩展总线接口、UART 接口）。除配置逻辑主端口外，所有带区组件均包括在全带区模型中。

可以使用全带区模型进行功能或时序仿真，验证带区中除配置逻辑以外的所有元素的功能和时序。软件代码还可以与全带区模型结合起来进行联合仿真。可以使用 Software Builder 将软件源文件转换为存储器初始化文件，与 EDA 仿真工具配合使用。

ESS 模型与 EDA 工具结合使用

ESS 模型是快速带区仿真模型，它仿效 Excalibur 嵌入式处理器内核、带区寄存器以及带区至 PLD 和 PLD 至带区总线处理的功能，用以仿真 Excalibur 设计。

ESS 模型包含 ARM 922T 处理器的功能精确的模型、监视器定时器、定时器和中断控制器、嵌入式 UART 以及到 PLD 的接口。它支持从闪存启动，通过载入片上内存的串行文件进行配置，并具有与 ARM 开发套件 (ADS) AXD 和 AWD 软件调试器和 GDB GNU 调试器连接的接口。

可以将 ESS 模型与 ModelSim PE 或 SE 软件配合使用，进行功能硬件仿真，仿真带区至 PLD 和 PLD 至带区桥接器和 PLD 接口，而对于软件和硬件协同仿真，还可以将 AXD 软件调试器（作为 ARM 开发套件的一部分提供）连接至 Excalibur 嵌入式处理器内核，控制软件代码的执行，同时在 ModelSim 软件中仿真硬件设计。

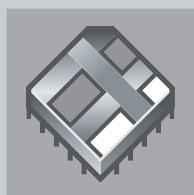
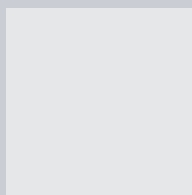
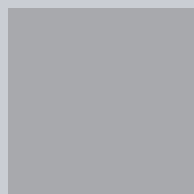
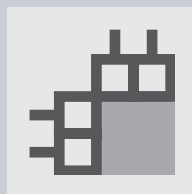
ESS 模型还可以使用 ModelSim PE/SE 和 ModelSim-Altera 软件仿真 Verilog HDL 设计，以及使用 ModelSim SE 软件仿真 VHDL 设计。可以从 PC 上的 AXD、ADW、Mentor Graphics XRAY 和 GNUPro arm-elf-gdb 软件调试器，以及从 Solaris 工作站上的 GNUPro arm-elf-gdb 调试将 ESS 模型作为目标。



有关信息	请参阅
总线功能模型	Altera 网站上的 <i>Excalibur Bus Functional Model User Guide</i> 《Quartus II Tutorial》中的 Excalibur 模块
在 EDA 仿真工具中使用总线功能模型和全带区模型	《Quartus II Help》中的 “Overview: Using the ModelSim Software with the Quartus II Software” Altera 网站上的 <i>Excalibur Hardware Design Tutorial</i> Altera 网站上的 <i>Application Note 240 (Simulating Excalibur Systems)</i>
进行仿真或与 ESS 模型一起联合仿真	《Quartus II Help》中的 “Overview: Using the ModelSim Software with the ESS Model” Altera 网站上的 <i>Excalibur Stripe 仿真器 User Guide</i>

第五章

布局布线

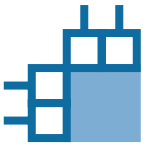


第 5 章内容：

简介	70
分析布局布线结果	71
优化布局布线	76
执行增量布局布线	81
通过反标保留分配	81

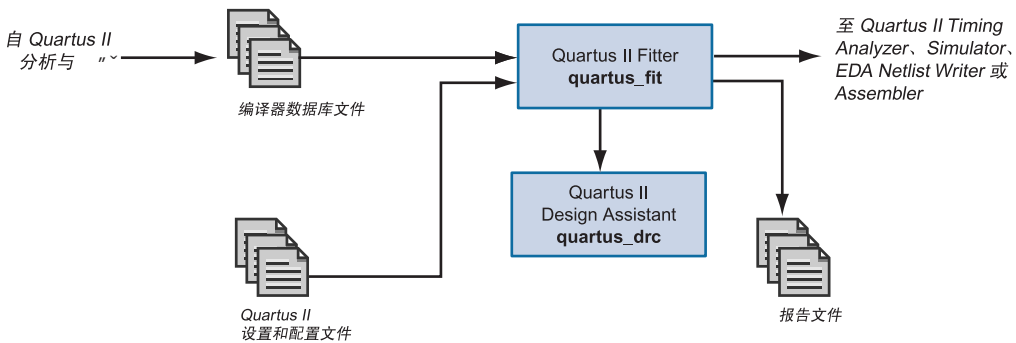
5

简介



Quartus® II Fitter 也称为 PowerFit™ Fitter，执行布局布线，这在 Quartus II 软件中也称为“布局布线”。Fitter 使用由 Analysis & Synthesis 建立的数据库，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好的逻辑单元位置，进行布线和时序，并选择相应的互连路径和引脚分配。图 1 显示布局布线设计流程。

图 1. 布局布线设计流程



如果在设计中执行了资源分配，Fitter 将试图将这些资源分配与器件上的资源相匹配，并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，则 Fitter 将自动优化设计。如果找不到布局布线，Fitter 会终止编译。

在 **Settings** 对话框（**Assignments** 菜单）**Compiler Settings** 项下的 **Mode** 页中，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，有助于将来更快地运行编译，但可能会消耗额外的磁盘空间。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。如果对设计的逻辑做任何更改，Compiler 在处理期间将使用所有模块。此选项类似于 MAX+PLUS® II **Smart Recompile** 命令（**Processing** 菜单）。

可以在包括 Fitter 模块的 Quartus II 软件中启动全编译，也可以单独启动 Fitter。在单独启动 Fitter 之前，必须成功运行 Analysis & Synthesis。

 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**。

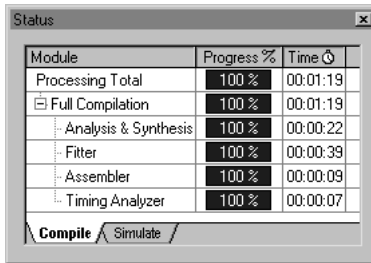
quartus_fit 可执行文件建立可以使用任何文本编辑器查看的单独的文本型报告文件。

若要获取有关 **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=<主题名称> ←
```

Status 窗口记录工程编译期间在 Fitter 中处理所花费的时间，以及您可能运行的任何其它模块的处理时间。请参阅图 2。

图 2. Status 窗口



Module	Progress %	Time
Processing Total	100 %	00:01:19
Full Compilation	100 %	00:01:19
Analysis & Synthesis	100 %	00:00:22
Fitter	100 %	00:00:39
Assembler	100 %	00:00:09
Timing Analyzer	100 %	00:00:07

Compile Simulate

分析布局布线结果

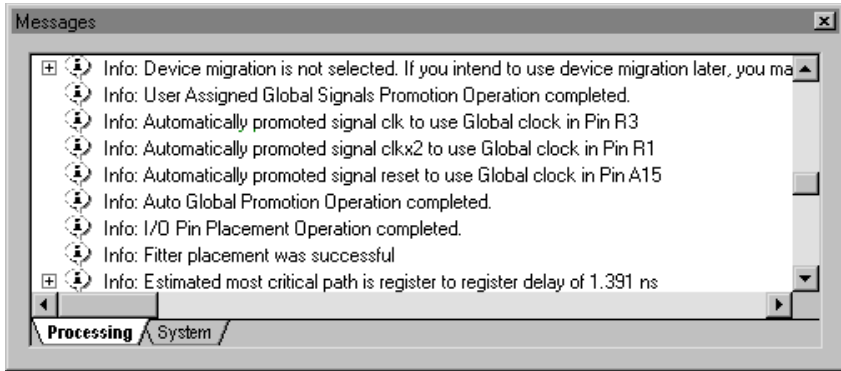
Quartus II 软件提供多个工具帮助您分析编译和布局布线的结果。消息窗口和 Report 窗口提供布局布线结果信息。Floorplan Editor 和 Chip Editor 还允许查看布局布线结果和进行必要的调整。此外，Design Assistant 可以帮助您根据一组设计规则检查设计的可靠性。

使用消息窗口查看布局布线结果



消息窗口的 **Processing** 选项卡以及 Report 窗口消息部分或报告文件将显示 Fitter 生成的消息。请参阅图 3。

图 3. 消息窗口



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

如果要过滤消息窗口中出现的消息，可以在 **Options** 对话框（**Tools** 菜单）的 **Processing** 选项卡中设置用于控制信息和 / 或警告消息显示的选项。消息窗口的右键弹出菜单还提供用于控制警告消息、关键消息、信息性消息和额外信息性消息的显示的命令。

如果消息中有一个源文件在设计中而且该源文件的位置可以找到，则可以在消息上单击鼠标右键，然后选择 **Locate**（右键弹出菜单）。**Utility Windows >Message Locations** 命令（**View** 菜单）也显示消息的源文件。



有关信息

请参阅

查看消息

《Quartus II Help》中的“Viewing 消息”

查找消息的源文件

《Quartus II Tutorial》的编译模块

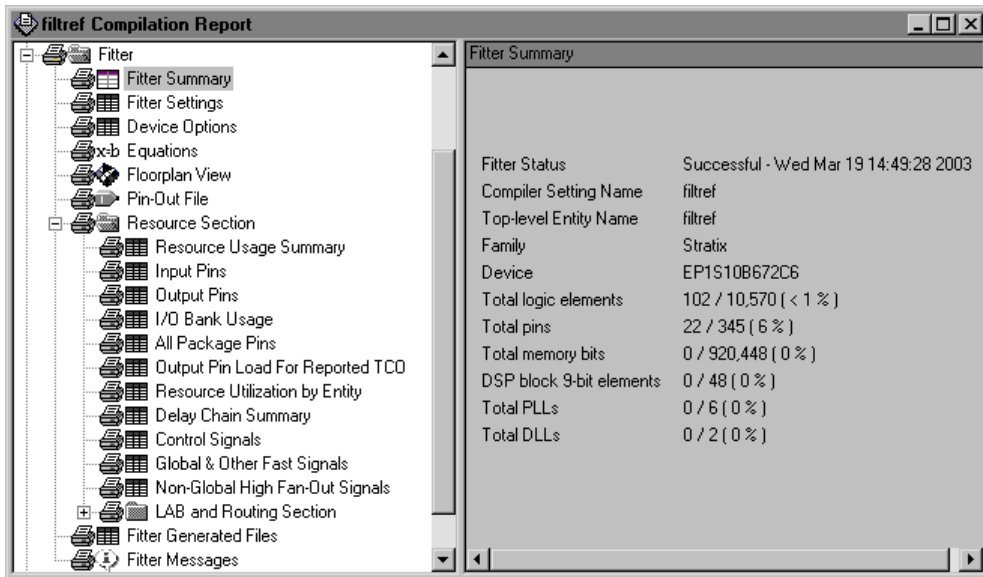
使用报告窗口或报告文件查看布局布线结果



Report 窗口包含许多部分，它们可以帮助您对 Fitter 为设计执行布局布线的方式进行分析。它包括好几个部分，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。

在运行 Fitter 或任何其它编译或仿真模块时，Report 窗口将自动打开；然而，如果 Compiler Tool 窗口已打开，则 Report 窗口不会自动打开，但单击每个模块的报告文件图标将显示该模块的报告。当 Fitter 正在处理设计时，Report 窗口中的信息将不断更新。如果您停止 Fitter，则 Report 窗口将仅显示您停止 Fitter 之前所建立的信息。请参阅第 73 页上的图 4。

图 4. 报告窗口的 Fitter 部分



视您在 **Options** 对话框（Tools 菜单）**Processing** 选项卡中指定的选项而定，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

报告窗口部分

《Quartus II Help》中的“Report Window & File Format”

使用报告窗口

《Quartus II Help》中的“Overview: Viewing the Results of a Compilation or Simulation in the Report Window”

查看编译报告

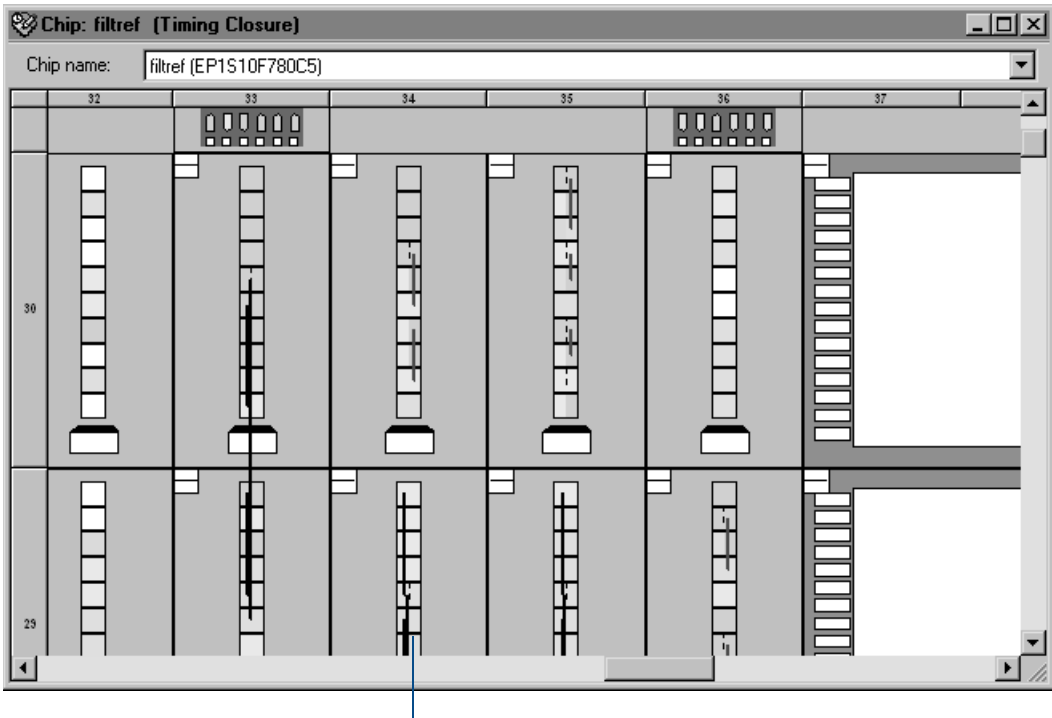
《Quartus II Tutorial》中的编译模块

使用 Floorplan Editor 分析结果



运行 Fitter 之后，Floorplan Editor 将显示布局布线的结果。可以查看不可编辑的（只读）上一个编译布局图，它显示上一次编译期间执行的资源分配和布线。此外，可以反标布局布线结果，以保留上次编译期间执行的资源分配。可编辑的时序逼近布局图允许查看 Fitter 和 / 或用户分配执行的逻辑分配，执行 LogicLock™ 区域分配以及查看布线拥塞情况。请参阅图 5。

图 5. Floorplan Editor



Floorplan Editor 允许查看资源
使用情况和布线

如果编译用于 Excalibur™ 器件的设计，还可以在 Floorplan Editor 中查看 Excalibur 嵌入式处理器带区。该带区位于逻辑单元和引脚之间，并包含与微处理器的嵌入式逻辑以及与双端口 RAM 的接口。

Floorplan Editor 中的资源使用情况用不同的色彩显示。不同的颜色代表不同的资源，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。Floorplan Editor 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 Floorplan Editor 中编辑分配，可以单击资源分配并将其拖放到新位置。在 Floorplan Editor 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的直观表示。

可以查看设计中的布线拥塞情况，查看路径的布线延时信息，并可以查看与特定节点的连接的计数。Floorplan Editor 还允许查看特定结构的节点扇出和节点扇入，或查看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 Floorplan Editor 的详细信息，请参阅“第 8 章：定时闭合”中的“第 110 页使用定时闭合平面布置图”。



如果要查看更详细的布局布线细节，以及进行另外的布局布线调整，Chip Editor 将显示 Floorplan Editor 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 和更改管理器进行更改。有关详细信息，请参阅第 141 页中的“第 11 章：工程更改管理”。



有关信息

请参阅

在 Floorplan Editor 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”

查看上次编译布局图中的布局布线

《Quartus II Tutorial》中的编译器模块

使用 Design Assistant 检查设计的可靠性



Quartus II Design Assistant 用于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布局布线或设计优化的任何问题。**Settings** 对话框（Assignments 菜单）的 **Design Assistant** 页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 3 章：综合”的“第 52 页使用 Design Assistant 检查设计可靠性”。

优化布局布线

运行 Fitter 并分析结果之后，可以试用多种选项来优化布局布线：

- 使用位置分配
- 设置用于控制布局布线的选项
- 使用设计空间管理器 (Design Space Explorer)

使用位置分配



可以通过使用 Floorplan Editor 或 Assignment Editor 将逻辑分配给器件上的物理资源，例如，引脚、逻辑单元或逻辑阵列块 (LAB)，以便控制布局布线。您可能想要使用 Floorplan Editor 编辑分配，因为它为您提供器件及其功能的图形视图。如果您要建立新的位置分配，您可能想要使用 Assignment Editor，因为它允许您同时建立多个节点特定的分配。除了使用 Floorplan Editor 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 2 章：设计输入”中的“第 36 页指定初始设计的约束条件”。

建立分配之后，可以在 Assignment Editor 或 Floorplan Editor 中进行编辑。编译之后，可以使用 Floorplan Editor 编辑分配给引脚、逻辑单元、行、列、区域、MegaLAB 结构和 LAB 的现有资源。可以使用 Floorplan Editor、LogicLock Region 窗口或 **LogicLock Region Properties** 对话框（**Assignments** 菜单）将节点或实体分配给 LogicLock 区域。

Floorplan Editor 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。还可以查看等式和路由信息，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，阻碍了在器件中执行布局布线，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布局布线。

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个选项：

- Fitter 选项
- Fitting 优化和物理综合选项
- 影响布局布线的逻辑选项

设置 Fitter 选项

Settings 对话框（**Assignments** 菜单）的 **Fitting** 页用于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。可以指定是否要 Fitter 使用标准布局布线（它会尽力满足 f_{MAX} 时序约束条件）还是使用快速布局布线功能（它可以提高编译速度但可能降低 f_{MAX} ）或是限制仅进行一次布局布线尝试（它也可能降低 f_{MAX} ）。

设置布局布线优化与物理综合选项

Quartus II 布局布线优化选项允许设置选项，用以在布局布线期间执行物理综合，优化网表。可以在 **Settings** 对话框（**Assignments** 菜单）的 **Netlist Optimizations** 页中指定 Quartus II Fitter 优化选项。

Fitter 优化选项包括以下选项：

- 进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
 - 进行寄存器重新定时

有关布局布线优化和物理综合选项的详细信息，请参阅“第 8 章：定时闭合”中的“第 113 页使用网表优化实现定时闭合”。

设置影响布局布线的逻辑选项

Quartus II 逻辑选项允许在不编辑源代码的情况下设置属性。可以在 Assignment Editor（Assignments 菜单）中为各个节点和实体指定 Quartus II 逻辑选项，并可以在 Settings 对话框（Assignments 菜单）中指定全局默认逻辑选项。例如，可以使用逻辑选项指定应在全局路由路径上的整个器件上提供信号，指定 Fitter 应自动建立并行扩展器链，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“寄存器封装”，或者限制承载链、串联链和并行扩展器链的长度。



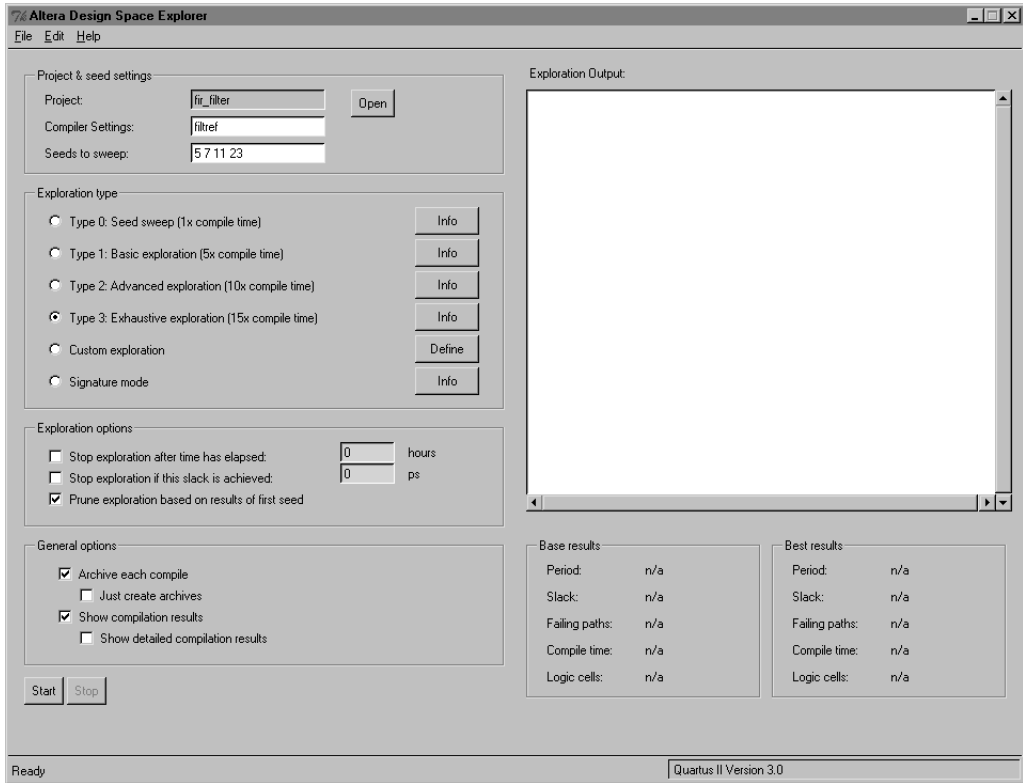
有关信息	请参阅
使用 Quartus II 逻辑选项控制布局布线	《Quartus II Help》中的“Logic Options”、“Creating, Editing & Deleting Assignments”和“Specifying Settings for Default Logic Options”
使用 Quartus II Fitter 优化选项	《Quartus II Help》中的“Optimizing Netlists During Synthesis & Fitting”

使用设计空间管理器 (Design Space Explorer)

控制 Quartus II 布局布线的另一种方法是使用设计空间管理器 (DSE)，它是一个 Tcl 脚本 `dse.tcl`，可以使用 `quartus_sh` 可执行文件从命令行运行它，来优化设计。DSE 界面允许自动探究一系列的 Quartus II 选项和设置，从而确定要使工程获得最好的结果应使用哪个设置。

可以指定允许 DSE 进行更改的程度、优化目标、目标器件以及允许的编译时间。请参阅图 6。

图 6. Design Space Explorer 界面



运行 Design Space Explorer (DSE)

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 DSE：

```
quartus_sh -t dse ↵
```

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）。

不应在 Quartus II 图形用户界面内运行 DSE。

DSE 提供多个探究模式，这些模式列在 DSE 窗口中的 **Exploration type** 项下：

- 类型 0：种子扫描
- 类型 1：基本探究
- 类型 2：高级探究
- 类型 3：详尽探究
- 自定义
- 签名模式

类型 0、类型 1、类型 2、类型 3 和自定义模式是参数扫描模式。这些模式允许指定必须达到的工程目标，例如，一组时序要求，使您可以改进时序。不同的模式均允许指定您要 DSE 为实现目标而在布局布线设计上所做的努力程度；但是，提高努力程度通常会增加编译时间。自定义探究模式允许指定各种参数、选项和模式，然后探究它们对设计的影响。如果不指定目标，DSE 将查找最好的 f_{MAX} 结果。

签名模式允许探究单个参数对设计的影响，以及在 f_{MAX} 、停滞、编译时间和区域之间进行取舍。在签名模式下，DSE 测试单个参数对多个种子的影响，然后报告平均值，使您可以评估该参数如何在设计的空间内进行交互。



有关信息	请参阅
使用设计空间管理器 (Design Space Explorer)	Altera 网站上的 <i>Application Note 198 (Timing Closure with the Quartus II Software)</i>
优化性能的参数和设置	Altera 网站上的 <i>Application Note 297 (Optimizing FPGA Performance Using the Quartus II Software)</i>

执行增量布局布线



如果所做的更改仅影响少数节点，还可以通过使用增量布局布线避免运行全编译。增量布局布线允许以尽量保留以前编译的布局布线结果的模式运行 Compiler 的 Fitter 模块。增量布局布线试图尽可能地再现以前编译的结果，这可以防止时序结果中出现不必要的变化，并且由于它重新使用以前编译的结果，因此所需的编译时间通常比标准布局布线要少。

可以通过选择 **Start > Start Incremental Fitting** 命令（Processing 菜单）开启增量布局布线。



从命令行运行增量布局布线

还可以使用 **quartus_fit** 可执行文件，在命令提示符下键入以下命令来运行增量布局布线：

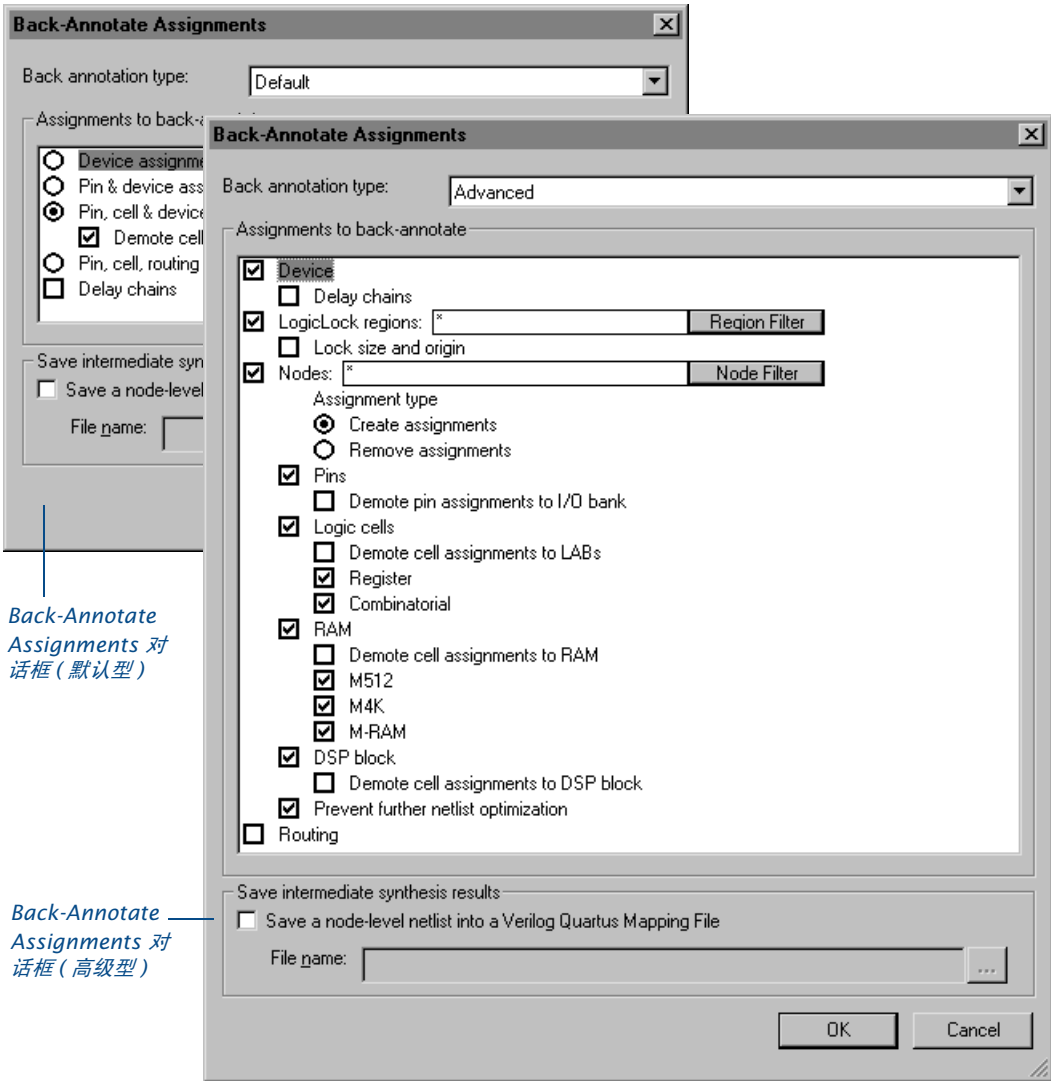
```
quartus_fit <工程名称> --incremental_fitting ←
```

通过反标保留分配

可以通过反标给任何器件资源的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 对话框允许选择反标的类型：默认型或高级型。请参阅图 7。

图 7. Back-Annotate Assignments 对话框



Back-Annotate Assignments 对话框 (默认型)

Back-Annotate Assignments 对话框 (高级型)

Back-Annotate Assignments (默认型) 对话框允许将引脚和 / 或逻辑单元分配“降级”为具有较少限制的位置分配，从而使 Fitter 在重新安排分配中具有更多的自由。**Back-Annotate Assignments** (高级型) 对话框可以执行默认型反标允许的任何操作，并允许反标 LogicLock 区域以及其中的节点和布线 (可选)。高级型反标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，不能两者都使用。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

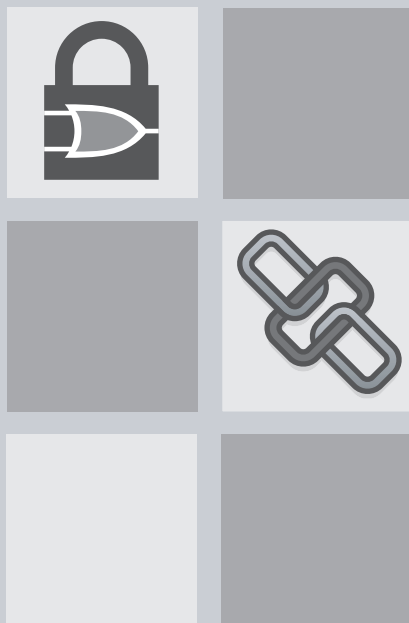
高级型反标，因为它提供更多的选项，尤其在使用 LogicLock 区域时更是如此。有关反标与 LogicLock 区域结合使用的详细信息，请参阅“第 6 章：基于块的设计”中的“第 92 页反向注释 LogicLock Region Assignments”。



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》的 LogicLock 模块

第 六章

基于块的设计



第 6 章内容：

简介	86
Quartus II 基于块的设计流程	86
使用 LogicLock 区域	87
保存中间综合结果	91
LogicLock 与 EDA 工具结合使用	95

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，然后将每个模块融合到最高层设计中。

LogicLock 区域是灵活且可重复使用的约束条件，能够提高您引导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何任意矩形区域定义为一个 LogicLock 区域。将节点或实体分配给 LogicLock 区域就是指示 Fitter 在布局布线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、基于块的设计，能够单独优化逻辑块，然后将它们及其布局约束条件导入更大的设计中。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，可供其它设计使用，不会出现性能损失，使用户能够充分利用资源和缩短设计周期。

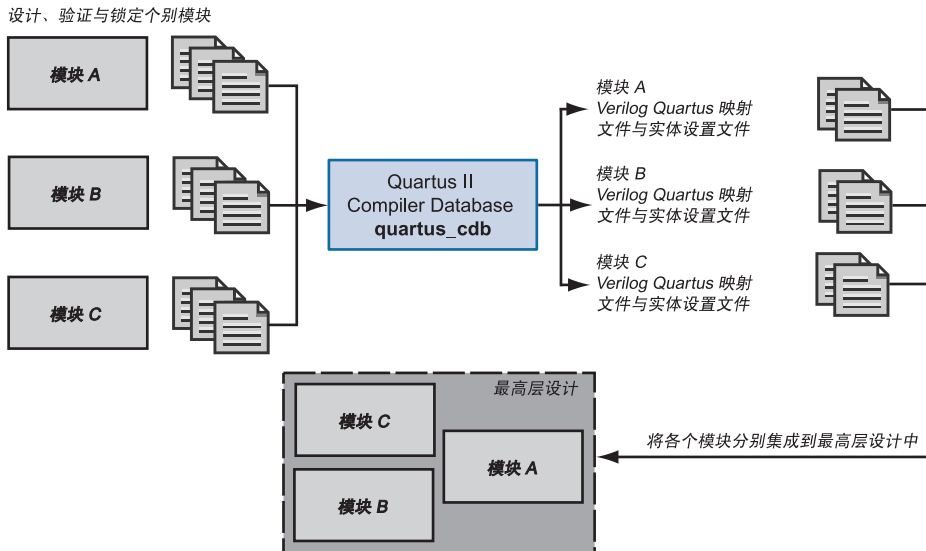
Quartus II 基于块的设计流程

在传统的从上到下的设计流程中，设计只有一个网表。在从上到下的设计流程中，各个设计模块在总体设计中可能具有不同的性能，这与模块的自我实现不同。在从下到上基于块的设计流程中，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，其中的每个模块可以独立优化，然后整合到最高层设计中。可以在以下设计流程中使用基于块的设计：

- **模块化设计流程：**在模块化设计流程中，将设计划分为对单独子模块进行实例化的最高层设计。可以单独开发每个模块，然后将每个模块整合到最高层设计中。布局可以由用户手动决定，也可以由 Quartus II 软件决定。
- **递增设计流程：**在递增设计流程中，用户建立并优化系统，然后添加对原始系统的性能影响较小或没有影响的未来模块。
- **团队型设计流程：**在团队型设计流程中，用户将设计分割为单独的模块，然后在最高层设计中对模块进行实例化和连接。然后，其它团队成员单独开发低层模块，为每个模块建立单独的工程，并使用为最高层设计而制定的分配。低层模块完成后，将它们导入最高层设计中，最高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有开发层上的性能，这些功能块按照电路的物理结构或关键路径组织起来。图 1 显示了基于块的基本设计流程。

图 1. 基于块的设计流程



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（第 1 部分，共 2 部分）

属性	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 软件按给定内容决定区域的大小。固定区域的形状和大小由用户定义。
保留	开或关	保留的属性用于规定 Quartus II 软件是否可以未分配给区域的实体使用区域中的资源。如果保留属性为开启，则只有分配给区域的项才可以放置在区域的边界内。

表 1. LogicLock 区域属性（第 2 部分，共 2 部分）

属性	值	行为
强制	硬或软	软区域较遵循时序约束条件，并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的边界之外。
原点	任意布局图位置	原点定义 LogicLock 区域在布局图中的位置。

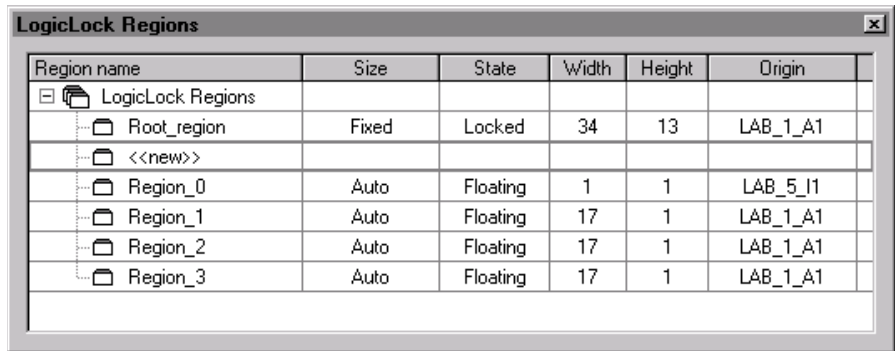
采用 LogicLock 设计流程，可以通过申报母区域和子区域来定义一组区域的层次结构。Quartus II 软件将子区域完全放置在母区域的边界内。可以锁定子模块相对于母区域的位置，而无需将母区域限定在器件的某锁定位置上。

可以使用 Floorplan Editor、**LogicLock Region Properties** 对话框（鼠标右键弹出菜单）、Project Navigator 的 **Hierarchy** 选项卡或使用 Tcl 脚本建立和修改 LogicLock 区域。建立 LogicLock 区域之后，还可以使用 Assignment Editor 将逻辑放置在该区域中。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）均存储在该特定工程的实体设置文件 (.esf) 中。

可以使用 Floorplan Editor 建立和编辑 LogicLock 区域分配。可以使用 **Create New Region** 按钮在时序逼近布局图中绘制 LogicLock 区域，然后在布局图视图、Node Finder、或 Project Navigator 的 **Hierarchy** 选项卡中拖放节点。

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。请参见图 2。

图 2. LogicLock Regions 窗口

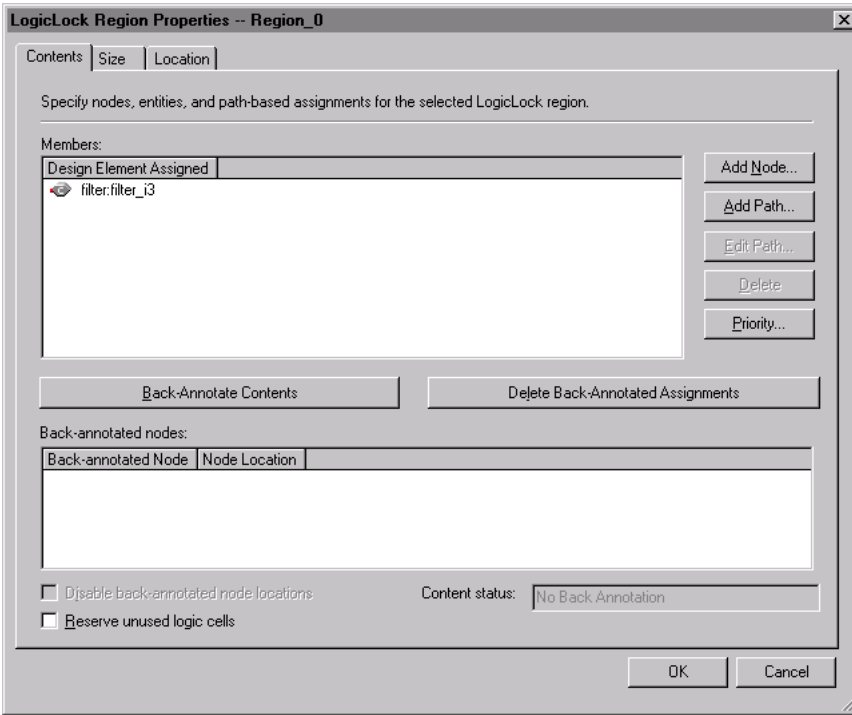


Region name	Size	State	Width	Height	Origin
LogicLock Regions					
Root_region	Fixed	Locked	34	13	LAB_1_A1
<<new>>					
Region_0	Auto	Floating	1	1	LAB_5_I1
Region_1	Auto	Floating	17	1	LAB_1_A1
Region_2	Auto	Floating	17	1	LAB_1_A1
Region_3	Auto	Floating	17	1	LAB_1_A1

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 区域中对所有节点进行反标，查看设计中 LogicLock 区域的信息，以及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的分配和通配符分配的 Fitter 优先级添加到 LogicLock 区域中。设置优先级用于指定 Quartus II 软件解决基于路径的分配和通配符分配的冲突的顺序。可以从 **LogicLock Region Properties** 对话框中打开 **Priority** 对话框。请参见图 3。

图 3. LogicLock Region Properties 对话框



在分析和详细描述或全编译之后，Quartus II 软件在 Project Navigator 的 **Hierarchy** 选项卡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，并从这些实体建立新的 LogicLock 区域，或将它们拖放到 Floorplan Editor 中的现有 LogicLock 区域内。

Altera 还通过命令行或 Quartus II Tcl Console 窗口提供 LogicLock Tcl 命令，用以分配 LogicLock 区域内容。可以使用提供的 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、导入和导出区域以及保存中间综合结果。



有关信息

请参阅

LogicLock 与 Quartus II 软件结合使用

Altera 网站上的 *Application Note 161 (Using the LogicLock Methodology in the Quartus II Design Software)*

《Quartus II Help》中的“Overview: Using LogicLock Regions”

《Quartus II Tutorial》中的 LogicLock 模块

保存中间综合结果

可以通过为设计中的实体建立 VQM 文件 (.vqm)，将各实体的综合结果与基于块的 LogicLock 设计流程一起保存，同时保存的相应实体设置文件包含实体 LogicLock 约束条件信息。

可以设计自定义逻辑块或实例化预验证知识产权 (IP) 块、给该块做分配、验证功能和性能、锁定该块以保持该布局 and 性能，然后将要导入的块导出至另一个设计中。这样，就可以单独设计、测试和优化块，并可以在将块集成为更大的设计时保持块的性能。

此外，通过将中间综合结果保存至 VQM 文件以及用导入分配的工程中的 VQM 文件替换实体，这样可以确保新工程中综合的节点名称与导入的分配中的节点名称相对应。

以下步骤描述为包含 LogicLock 区域的设计保存中间综合结果、反标分配以及导出和导入实体设置文件的基本流程。

1. 通过在 **Settings** 对话框（**Assignments** 菜单）的 **Synthesis** 页中或在 **Back-Annotate Assignments**（高级类型）对话框（**Assignments** 菜单）中指定 VQM 文件的名称，指定 Quartus II 软件应将当前编译注意项的中间综合结果转存为 VQM 文件。
2. 建立 LogicLock 区域。
3. 编译设计，生成 VQM 文件，或者使用 **Start > Start VQM Writer** 命令（**Processing** 菜单），在初始编译之后生成 VQM 文件。
4. 使用 **Back-Annotate Assignments**（高级类型）对话框将逻辑布局锁定在 LogicLock 区域中。

5. 通过使用 **Export LogicLock Regions** 命令（Assignments 菜单）将 LogicLock 区域分配导出至实体设置文件中。
6. 然后，可以将 VQM 文件中的模块实例化为最高层设计，并使用 **Import LogicLock Regions** 命令（Assignments 菜单）导入 LogicLock 区域分配。



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** 可执行文件单独导出和导入 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=<主题名称> ←
```

反标 LogicLock Region Assignments

可以在导出分配用于最高层设计之前，使用 **Back-Annotate Assignments**（高级类型）命令逻辑布局锁定在一个设计的 LogicLock 区域中。在将区域及其分配导入到最高层设计中时，使用反标可以保持 LogicLock 区域的性能。

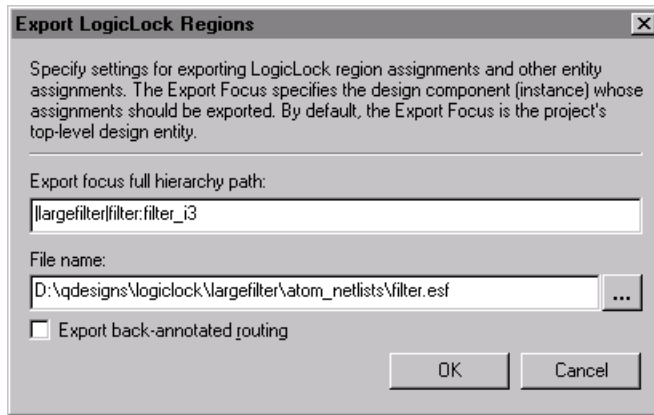
必须使用 **Back-Annotate Assignments**（高级类型）命令反标 LogicLock 区域分配，还可以使用它反标不包含 LogicLock 区域分配的设计。有关反标分配的详细信息，请参阅“第 5 章：布局布线”中的“第 81 页通过反标保留分配”。

导出与导入 LogicLock 分配

在 **Export LogicLock Regions** 和 **Import LogicLock Regions** 对话框（Assignments 菜单）中，可以使用 LogicLock 区域分配单独优化实体，并在最高层设计中对这些实体进行实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、其它实体设置文件分配和 I/O 标准分配，这些分配应用于在 **Export LogicLock Regions** 对话框中指定的实体设置文件的特定实体实例。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 focus full hierarchy path** 对话框中导出的子设计实体。请参见图 4。

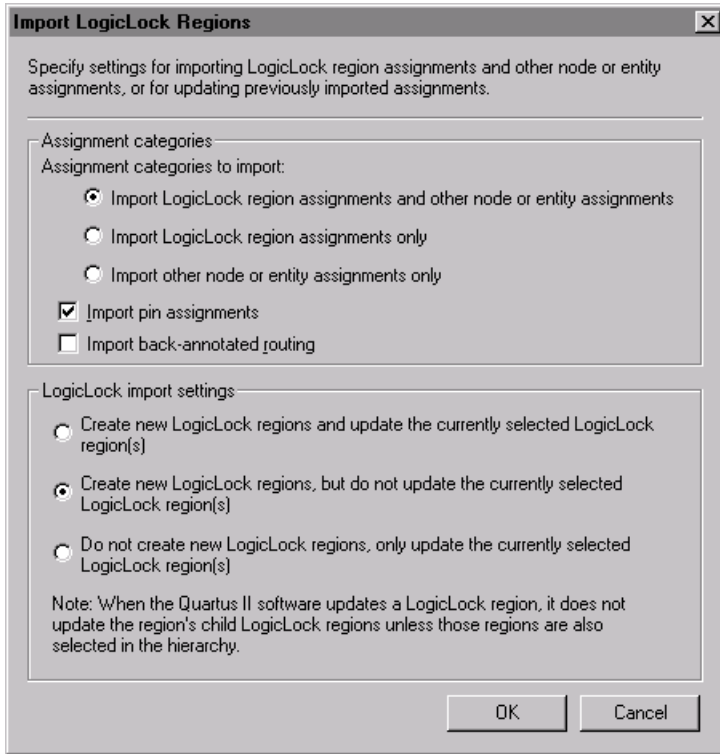
图 4. **Export LogicLock Regions** 对话框



在导入 LogicLock 区域分配时，Quartus II 软件从当前编译注意项开始跨越编译层次结构。如果当前工程包含低层实体的多个实例，Quartus II 软件将对为该低层实体而导入的分配进行实例化，每个实例进行一次实例化。

为防止布局冲突，Quartus II 软件将导入的最高层 LogicLock 区域分配在浮动位置上。但是，它保留导入的子区域相对于母区域的位置。导入 LogicLock 区域时，可以指定分配类别，以导入和指定是否建立新 LogicLock 区域和（或）更新当前所选的 LogicLock 区域。请参阅图 5。

图 5. Import LogicLock Regions 对话框



有关信息

将中间综合结果转存为 VQM 文件、反标分配以及导出和导入 LogicLock 区域分配

请参阅

Altera 网站上的 *Application Note 161 (Using the LogicLock Methodology in the Quartus II Design Software)*

《Quartus II Help》中的“Overview: Saving Intermediate Synthesis Results”

《Quartus II Help》中的“Overview: Using LogicLock Regions”

《Quartus II Tutorial》中的 LogicLock 模块

LogicLock 与 EDA 工具结合使用



基于块的 LogicLock 设计流程支持在 EDA 设计输入和综合工具中建立、优化，然后作为单独模块被导入到 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 输入文件（.edf）或 VQM 文件）。然后，可以使用 Quartus II 软件将每个网表文件放入最高层设计中的单独 LogicLock 区域中。一旦进入了 Quartus II 软件中，就可以使用 EDA 工具更改、优化、再综合设计中的某特定模块，更新相应的网表文件，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 软件具有自定义功能，从而可以在基于块的 LogicLock 设计流程中使用这些工具。



有关信息

LogicLock 与 EDA 综合工具结合使用

请参阅

Altera 网站上的 *Application Note 165 (Synplify and Quartus II LogicLock Design Flow)*

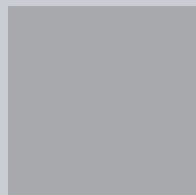
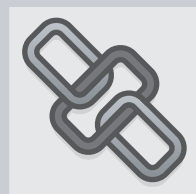
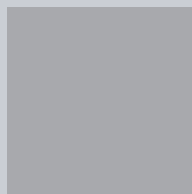
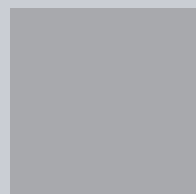
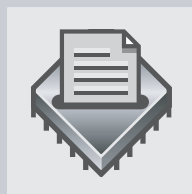
Altera 网站上的 *Application Note 164 (LeonardoSpectrum and Quartus II LogicLock Design Flow)*

Altera 网站上的 *Application Note 171 (FPGA Compiler II BLIS & the Quartus II LogicLock Design Flow)*

Altera 网站上的 *Application Note 222 (Using Precision RTL Synthesis in the Quartus II Design Methodology)*

第七章

时序分析



第 7 章内容：

简介	98
在 Quartus II 软件中进行时序分析	99
查看时序分析结果	102
使用 EDA 工具进行时序分析	106

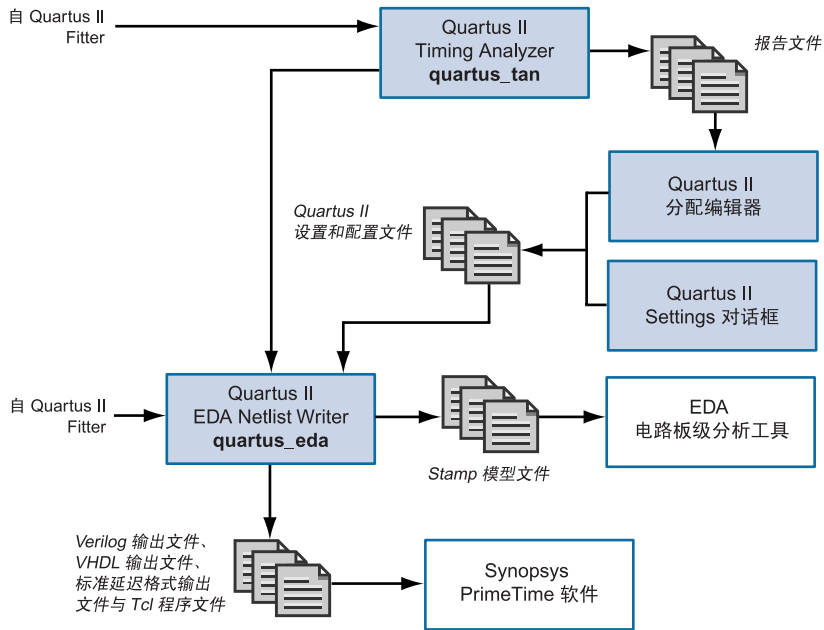
7

简介



Quartus® II Timing Analyzer 允许用户分析设计中所有逻辑的性能，并协助引导 Fitter 满足设计中的时序分析要求。默认情况下，Timing Analyzer 作为全编译的一部分自动运行，它观察和报告时序信息，例如，建立时间 (t_{SU})、保持时间 (t_{H})、时钟至输出延时 (t_{CO})、引脚至引脚延时 (t_{PD})、最大时钟频率 (f_{MAX})、延缓时间以及设计的其它时序特性。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 Quartus II Timing Analyzer 进行最少的时序分析，它报告最佳情况时序结果，验证驱动芯片外信号的时钟至管脚延时。图 1 显示了时序分析流程。

图 1. 时序分析流程



在 Quartus II 软件中进行时序分析

Timing Analyzer 在全编译期间对设计自动进行时序分析。以下准则描述了使用 Quartus II Timing Analyzer 可以完成的一些任务：

- 使用 **定时设置向导**（Assignments 菜单）、**Settings** 对话框（Assignments 菜单）和 Assignment Editor，指定初始工程范围的时序要求和个别时序要求。
- 在全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。
- 使用报告窗口、时序逼近布局图和 list_paths 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 **定时设置向导**（Assignments 菜单）帮助您建立初始工程范围时序设置。指定初始时序设置之后，可以再次使用 **定时设置向导** 或使用 **Settings** 对话框（Assignments 菜单）修改设置。

可以使用 **Assignment Editor**（Assignments 菜单）进行个别时序设置。指定工程范围时序分配和 / 或单个时序分配之后，通过编译设计或在初始编译之后通过单独运行 **Timing Analyzer** 来运行时序分配。

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} 、延缓时间、保持时间、最小 t_{CO} 以及当前设计实体的最小 t_{PD} 。

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

- 工程的总频率要求或各个时钟信号的设置
- 延时要求、最短延时要求和路径切割选项
- 报告选项，包括数字或源以及目标寄存器，且不包括路径
- 时序驱动编译选项

指定工程范围的时序设置

工程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低时序要求。还可以设置工程范围的时钟设置和多个时钟域、路径切割选项和默认外部延时。

表 1. 工程范围的时序设置

要求	描述
f_{MAX} (最大频率)	在不违反内部建立 (t_{SU}) 和保持 (t_H) 时间要求下可以达到的最大时钟频率。
t_{SU} (时钟建立时间)	在触发寄存器计时的时钟信号已经在时钟引脚确立之前, 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出现的时间长度。
t_H (时钟保持时间)	在触发寄存器计时的时钟信号已经在时钟引脚确立之后, 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转换之后, 在由寄存器馈送信号的输出引脚上取得有效输出所需的时间。
t_{PD} (引脚至引脚延时)	输入引脚处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。
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转换之后, 在由寄存器馈送信号的输出引脚上取得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。
最短 t_{PD} (时钟至输出延时)	指定可接受的最少的引脚至引脚延时, 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。

指定个别时序分配

可以使用 **Assignment Editor** 对个别实体、节点和引脚进行个别时序分配。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）。**Assignment Editor** 支持点到点时序分配和通配符，用于在做分配时标识特定节点。

输入的引脚和节点时序要求保存在当前层次结构中最高层实体的实体设置文件 (.esf) 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
- **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** 调整其度量，并避免不当地违反建立或保持时间。
- **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。
- **最少延时要求：**特定节点或组的个别 t_H 、最小 t_{CO} 和最小 t_{PD} 时序要求。可以对特定节点或组进行这些分配，以超越工程范围最小时序要求。
- **外部延时：**指定信号从外部寄存器（器件之外）到达输入引脚的延时。
- 设计中特定节点的个别 t_{SU} 、 t_{PD} 和 t_{CO} 要求。

进行时序分析

指定时序设置和分配之后，就可以通过全编译运行 **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，或通过选择 **Start > Start Minimum Timing Analysis**（**Processing** 菜单）运行最少时序分析。

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单独的文本型报告文件。

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=<主题名称> ←
```

有关信息

特定的时序设置以及在 Quartus II 软件中进行时序分析

请参阅

《Quartus II Help》中的“Overview: Using the Timing Analyzer”

Altera 网站上的 *Application Note 123 (Using Timing Analysis in the Quartus II Software)*

《Quartus II Tutorial》的 Timing Analysis 模块

查看时序分析结果

运行时序分析之后，可以在编译报告的 **Timing Analyzer** 文件夹中查看时序分析结果。然后，可以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，并进行另外的时序分配。此外，还可以使用 `list_paths` Tcl 命令查找并查看设计中任何延时路径的信息。

熟悉 MAX+PLUS® II 时序报告的用户可以查找时序信息，例如，查找来自编译报告中 Timing Analyzer 部分 Delay Matrix 中的延时信息。

使用报告窗口



报告窗口的时序分析部分列出报告的时钟建立和保持的时序信息； t_{SU} 、 t_H 、 t_{PD} 、 t_{CO} ；最小脉冲宽度要求；在时序分析期间忽略的任何时序分配；以及 Timing Analyzer 生成的任何消息。默认情况下，Timing Analyzer 还报告最佳情况最少时钟至输出时间和最佳情况最少点到点延时。

报告窗口包括以下类型的时序分析信息：

- 时序要求的设置
- 停滞和最少停滞
- 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
- 所需的保持关系
- 实际 f_{MAX}

图 2. Report 窗口中的时序分析结果

Timing Analyzer Summary					
Type	Slack	Required Time	Actual Time	Source Name	Destination Name
1	6.412 ns	50.00 MHz (period = ...	73.59 MHz (peri...	state_m.inst1 filter~11	acc:inst3 result[11]
2	8.254 ns	100.00 MHz (period ...	N/A	inst4	inst5[7]
3	N/A	None	4.146 ns	newt	state_m.inst1 filter~8
4	N/A	None	9.694 ns	state_m.inst1 filter~11	next
5	N/A	None	-1.530 ns	d[4]	taps:inst[kn[4]~reg0
6	N/A	None	5.518 ns	inst5[6]	yn_out[6]

进行分配与查看延时路径

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

可以使用 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对路径进行点到点分配。

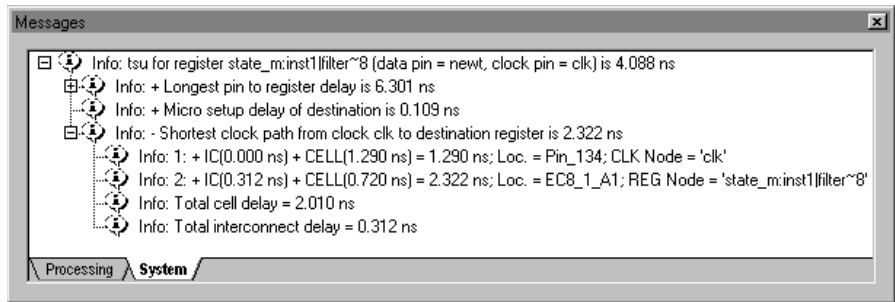
以下步骤描述在 Assignment Editor 中进行个别时序分配的基本流程：

1. 在 **Category** 栏中，单击 **Timing**，以指示您要做的分配类别。
2. 在电子表格中单击 **Destination Name (To)** 单元格并使用 Node Finder 查找节点，或键入标识要分配目标节点的节点名称和 / 或通配符。
3. 在电子表格中单击 **Source Name (From)** 单元格并使用 Node Finder 查找节点，或键入标识要分配源节点的节点名称和 / 或通配符。
4. 在电子表格中，双击 **Option** 单元格并选择您要做的时序分配。对于要求赋值的分配，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）在时序逼近布局图中查找路径，允许利用时序逼近布局图功能对特定路径做分配。有关使用时序逼近布局图的详细信息，请参阅“第 8 章：定时闭合”中的“第 110 页使用定时闭合平面布置图”。

可以使用 **List Paths** 命令（右边按钮弹出菜单），在消息窗口的 Timing Analyzer 报告小窗口中显示任何路径的中间延时。它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，并显示在 Report 窗口中出现的设计的任何延时路径的信息。请参见图 3。

图 3. Output from List Paths 命令



`list_paths` Tcl 命令可以在 `quartus_tan` API 和 Quartus II Tcl 控制台中使用，用于指定任何点到点路径和查看延时信息。可以指定要报告的路径数，路径类型（包括最少时序路径）和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。请参见图 4。

图 4. Sample Output from list_paths 命令

```
-----
Path Number: 1
too 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'^M
  2: + IC(1,638 ns) + CELL(0,542 ns) = 2,799 ns; Loc. = LC_X31_Y1_N9; REG Node = 'auto_max:auto1street_map[0]'^M
  Total cell delay = 1,161 ns^M
  Total interconnect delay = 1,638 ns^M
+ Micro clock to output delay of source is 0,156 ns^M
+ Longest register to pin delay is 5,914 ns^M
  1: + IC(0,000 ns) + CELL(0,000 ns) = 0,000 ns; Loc. = LC_X31_Y1_N9; REG Node = 'auto_max:auto1street_map[0]'^M
  2: + IC(0,716 ns) + CELL(0,075 ns) = 0,791 ns; Loc. = LC_X30_Y1_N9; COMB Node = 'rtl"261'^M
  3: + IC(0,518 ns) + CELL(0,366 ns) = 1,675 ns; Loc. = LC_X30_Y1_N8; COMB Node = 'rtl"17'^M
  4: + IC(1,350 ns) + CELL(2,889 ns) = 5,914 ns; Loc. = Pin_AA13; PIN Node = 'gt1'^M
  Total cell delay = 3,330 ns^M
  Total interconnect delay = 2,584 ns^M
-----
```

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少时序分析，并支持使用 Mentor Graphics® BLAST 或 Tau 板级验证工具进行板级时序分析。

在建立新工程时，可以通过在 **Settings** 对话框（**Assignments** 菜单）或 **New Project Wizard**（**File** 菜单）中指定相应的时序分析工具，生成要在 EDA 时序分析工具中做时序分析所必需的输出文件，然后进行全编译。还可以在初始编译之后，使用 **Start > Start EDA Netlist Writer** 命令（**Processing** 菜单）makefile。

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** 可执行文件单独生成必要的输出文件。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单独的文本型报告文件。

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=< 主题名称 > ←
```

使用 PrimeTime 软件

Quartus II 软件生成 Verilog 输出文件或 VHDL 输出文件、包含时序延时信息的标准延时格式输出文件 (.sdo) 以及设置 PrimeTime 环境的 Tcl 脚本文件。如果正在进行最少时序分析，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少延时信息。

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 设计约束 (SDC) 文件，此文件包含供 PrimeTime 软件中使用的时序分配。

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）指定 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的目录中。
3. 运行 Quartus II 生成的 Tcl 脚本文件 (.tcl)，用以设置 PrimeTime 环境。
4. 在 PrimeTime 软件中做时序分析。

使用 BLAST 和 Tau 软件

Quartus II 软件生成 Stamp 模型文件，此文件可以被导入 BLAST 或 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的目录中。
3. 在 BLAST 或 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 软件配合使用

《Quartus II Help》“Overview: Using the PrimeTime Software with the Quartus II Software”

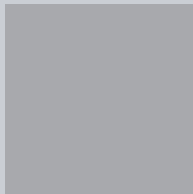
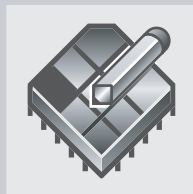
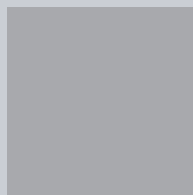
Innoveda BLAST 和 Mentor Graphics Tau 软件与 Quartus II 软件配合使用

《Quartus II Help》中的 “Overview: Using the BLAST Software with the Quartus II Software”

《Quartus II Help》中的 “Overview: Using the Tau Software with the Quartus II Software”

第八章

时序逼近



第 8 章内容：

简介	110
使用时序逼近布局图	110
使用网表优化实现时序逼近	113
使用 LogicLock 区域实现时序逼近	116

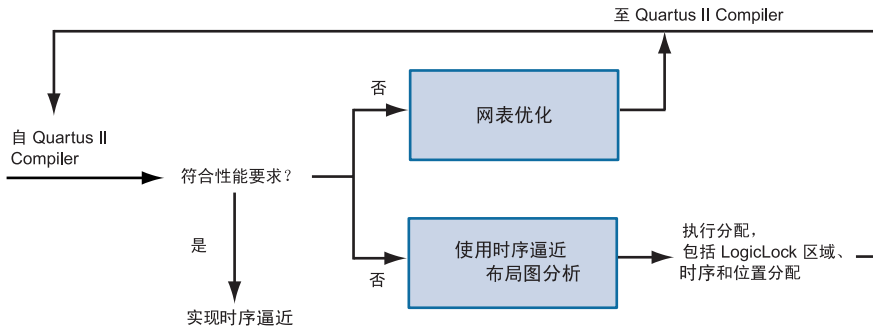
8

简介

Quartus® II 软件提供完全集成的时序逼近流程，可以通过控制综合和设计的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、查看设计结果以及有效地对设计进行进一步优化。在综合之后以及在布局布线期间，可以在设计上使用网表优化，使用时序逼近布局图分析设计并执行分配，以及使用 LogicLock™ 区域分配进一步优化设计。图 1 显示了时序逼近流程。

图 1. 时序逼近流程



使用时序逼近布局图



可以使用时序逼近布局图查看 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。可以使用此信息在设计中标识关键路径，并执行时序分配、位置分配和 LogicLock 区域分配，实现时序逼近。

可以使用 View 菜单中提供的选项自定义时序逼近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 和单元格显示器件；按芯片的区域显示器件；按所选信号的名称和位置显示器件；以及使用 Field View 命令（View 菜单）显示器件。

Field View 命令在 Floorplan Editor 的高级总体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，这些彩色区域显示已分配用户量、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，实现设计的时序逼近。

查看分配与布线

时序逼近布局图可以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 软件在最后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。

时序逼近布局图允许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），可以选择器件资源和查看以下布线信息类型：

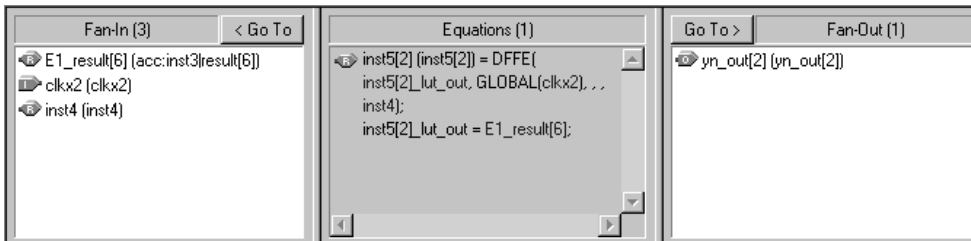
- **节点之间的路径：**显示所选逻辑单元之间的路径、I/O 单元、嵌入式单元和相互馈电的引脚。
- **节点扇入和扇出：**显示所选嵌入式单元、逻辑单元、I/O 单元和引脚的节点扇入和扇出布线信息。
- **布线延时：**显示所选特定逻辑单元之间、引至或源自特定逻辑单元、I/O 单元之间、引至或源自 I/O 单元、嵌入式单元之间、引至或源自嵌入式单元、引脚之间、引至或源自引脚的布线延时；或显示一个或多个关键路径的布线延时。
- **连接计数：**显示或隐藏接至选定对象、接自选定对象或选定对象之间的连接数量。
- **布线统计：**显示一个或多个选定引脚、逻辑单元、嵌入式单元、I/O 单元、LAB 或 MegaLAB 结构的布线统计。
- **物理时序估计：**显示到达器件上任何其它节点或实体的近似延时。如果选择了一个节点或实体，则用潜在目标资源的阴影表示延时（资源的阴影越深，延时越长），可以将鼠标放置在可能目标节点之上来显示到达目标节点的延时。
- **布线拥塞：**显示用图形表示的设计中的布线拥塞。阴影越深，布线资源利用率越大。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。可以使用延时或停滞指标来指定是否要查看关键路径，并可以指时序钟域、源节点名称和目标节点名称以及要显示的关键路径数。

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）中源路径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 Floorplan Editor 窗口的底部显示 Equations 窗口。请参阅第 112 页上的图 2。

通过在布局图中选择一个或多个逻辑单元、嵌入式单元和 / 或引脚分配，可以在 **Equations** 列表中显示其等式、扇入和扇出，并可以扩展或收敛条件。**Fan-In** 列表显示馈送选定逻辑单元、嵌入式单元和 / 或引脚分配的所有节点。**Fan-Out** 列表显示由选定逻辑单元、嵌入式单元和 / 或引脚分配馈送的所有节点。

图 2. 等式窗口



执行分配

为便于实现时序逼近，时序逼近布局图允许直接从布局图进行位置和时序分配。可以在时序逼近布局图的自定义区域和 LogicLock 区域中建立和分配节点或实体，还可以编辑对引脚、逻辑单元、行、列、区域、MegaLAB 结构和 LAB 的现有分配。

可以使用以下方法在时序逼近布局图中编辑分配：

- 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** 选项卡、LogicLock 区域和时序逼近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 区域进行反标分配，从而保留来自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”中的“第 81 页通过反标保留分配”。



有关信息

请参阅

查看和执行分配，并在时序逼近布局图中查看布线

Altera 网站上的 *Application Note 198 (Timing Closure with the Quartus II Software)*

《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

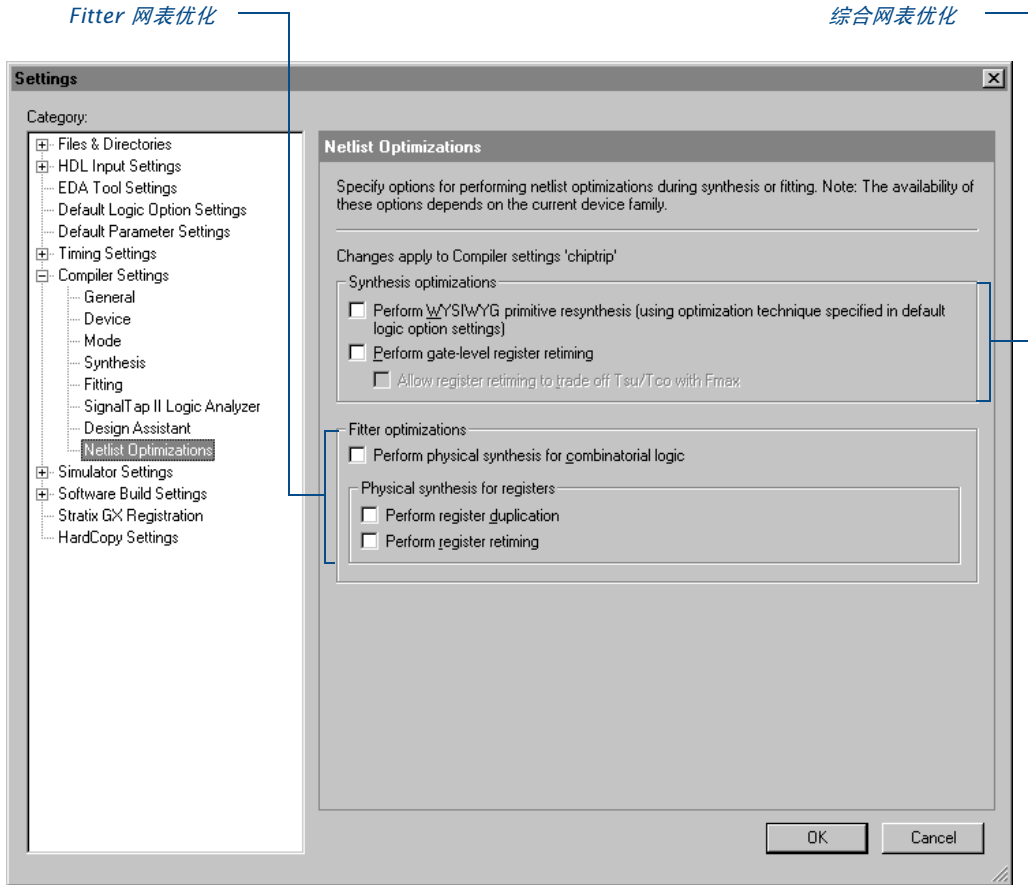
使用网表优化实现时序逼近



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。网表优化是下压按钮功能，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。视设计而定，有些选项可能比其它选项具有更多的效果。

可以在 **Settings** 对话框（Assignments 菜单）的 **Netlist Optimizations** 页中指定 Analysis & Synthesis 和 Fitter 网表优化。请参阅第 114 页上的图 3。

图 3. 网表优化



综合的网表优化包括以下选项：

- **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 软件将取消原子网表中逻辑元素对门的映射，并将该门重新映射回 Altera® LCELL 基本单元。使用此选项，Quartus II 软件能够在重新映射过程中使用特定于器件体系架构的不同技术。
- **执行门电平寄存器重新定时：** 允许在组合逻辑间移动寄存器以平衡时序，但不更改当前的设计功能。此选项仅在组合门间移动寄存器，不在用户实例化的逻辑单元、存储器块、DSP 块或者承载或串联链间移动寄

寄存器，并能够将寄存器从组合逻辑块的输入移到输出，从而可能将寄存器结合在一起。它还可以用组合逻辑块输出端的寄存器在组合逻辑块的输入端建立多个寄存器。

- **允许寄存器重新定时，在 F_{max} 和 T_{su}/T_{co} 之间进行取舍：** 指示 Quartus II 软件在寄存器重新定时，实现对 t_{CO} 和 t_{SU} 与 f_{MAX} 的取舍时，在与 I/O 引脚相关的寄存器间移动逻辑。此选项开启之后，寄存器重新定时可能对馈送 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，寄存器重新定时不会接触与 I/O 引脚相连的任何寄存器。

布局布线和物理综合的网表优化包括以下选项：

- **进行组合逻辑的物理综合：** 指示 Quartus II 软件在布局布线期间对组合逻辑进行物理综合优化，以提高性能。
- **进行寄存器复制：** 指示 Quartus II 软件在布局布线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。
- **进行寄存器重新定时：** 指示 Quartus II 软件在布局布线期间使用寄存器重新定时对寄存器进行物理综合优化，以提高性能。

Quartus II 软件不能为反标设计上的布局布线和物理综合进行这些网表优化。此外，如果在设计上使用这些网表优化中的一个或多个，然后对设计进行反标，如果要保存结果，则必须生成 VQM 文件 (**.vqm**)。在将来编译中，必须用 VQM 文件代替原始设计源代码。



有关信息

请参阅

使用网表优化实现时序逼近

Altera 网站上的 *Application Note 198 (Timing Closure with the Quartus II Software)*

使用 LogicLock 区域实现时序逼近



可以使用 LogicLock 区域实现时序逼近，方法是：在时序逼近布局图中分析设计，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多的控制。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，需要对设计的关键路径有详细的了解。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

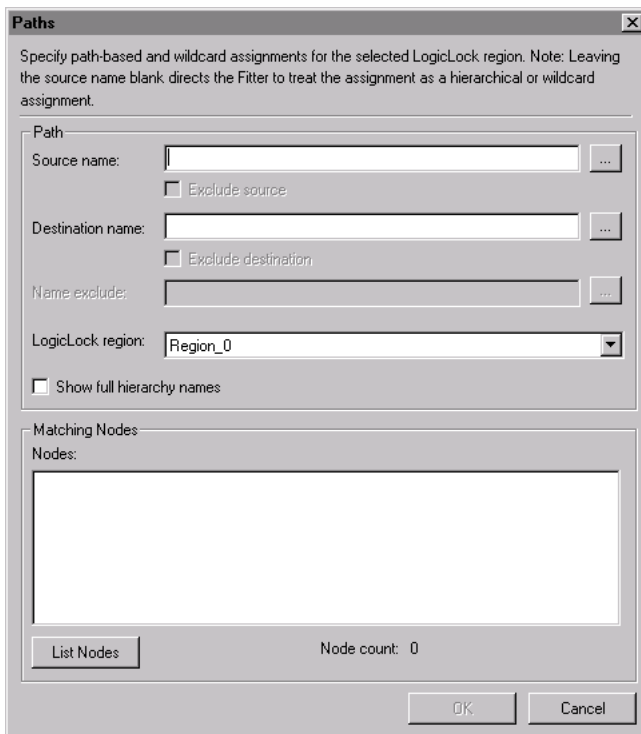
LogicLock 区域具有预定义边界和节点，这些边界和节点分配给始终驻留在边界或 LogicLock 区域范围内的特定区域。软 LogicLock 区域可以通过删除 LogicLock 区域的固定矩形边界来增强设计性能。启用软区域属性后，Fitter 试图在区域中尽量多布置一些已分配节点并尽各种可能将它们紧邻布局，并提高软区域外移动节点的灵活性来满足设计的性能要求。

基于路径的分配

Quartus II 软件可以将特定的源路径和目标路径分配给 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。可以使用 **Path** 对话框（从 **LogicLock Region Properties** 对话框进入）建立基于路径的分配，方法是从 Report 窗口的 Timing Analyzer 区域中拖放路径，以及从时序逼近布局图（通过右键单击时序逼近布局图中的关键路径并选择 **Properties** 而进入）中拖放关键路径。

Paths 对话框允许通过标识源节点和目标节点指定路径，在标识节点时使用通配符，以及单击 **List Nodes** 以确定将要分配给 LogicLock 区域的节点数。请参见图 4。

图 4. Paths 对话框



有关信息

使用 LogicLock 方法实现时序逼近

建立 LogicLock 区域

请参阅

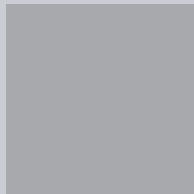
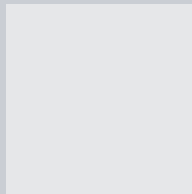
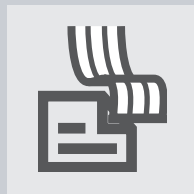
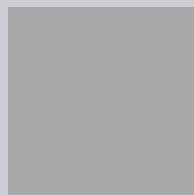
Altera 网站上的 *Application Note 161 (Using the LogicLock Methodology in the Quartus II Design Software)*

Altera 网站上的 *Application Note 198 (Timing Closure with the Quartus II Software)*

《Quartus II Tutorial》的 LogicLock 模块

第九章

编程与配置



第 9 章内容：

简介	120
使用 Programmer 对一个或多个器件进行编程	123
建立辅助编程文件	124
使用 Quartus II 软件 通过远程 JTAG 服务器编程	129

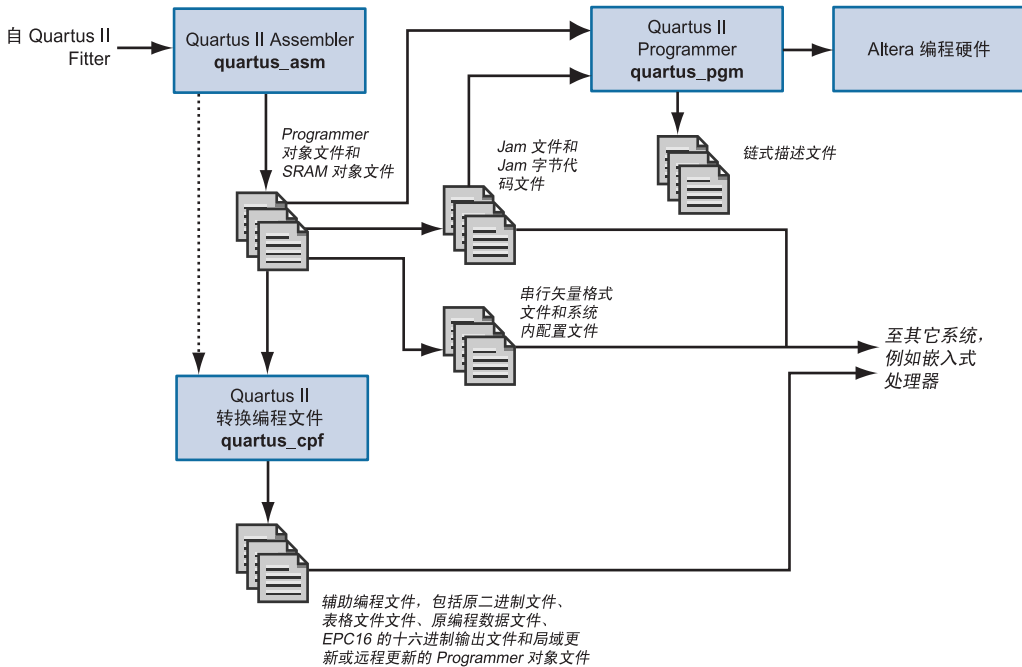
9

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 显示了编程设计流程。

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该器件的编程图像，这些图像以目标器件的一个或多个 Programmer 对象文件 (.pof) 或 SRAM 对象文件 (.sof) 的形式存在。

可以在包括 Assembler 模块的 Quartus II 软件中启动全编译，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` 可执行文件独立运行 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基于文本的单独报告文件。

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=<主题名称> ←
```

还可以指示 Assembler 通过以下方法之一以其它格式生成编程文件：

- **Device & Pin Options** 对话框可以从 **Settings** 对话框（Assignments 菜单）的 **Device** 页进入，允许指定可选编程文件格式，例如，十六进制（Intel 格式）输出文件（`.hexout`）、表格文本文件（`.ttf`）、原二进制制文件（`.rbf`）、Jam™ 文件（`.jam`）、Jam 字节代码文件（`.jbc`）、串行矢量格式文件（`.svf`）和系统内配置文件（`.isc`）。
- **Create/Update > Create JAM, SVF, or ISC File** 命令（File 菜单）生成 Jam 文件、Jam 字节代码文件、串行矢量格式文件或系统内配置文件。
- **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，原编程数据文件（`.rpd`）、EPC16 或 SRAM 的 HEXOUT 文件、POF、局域更新或远程更新的 POF、原二进制文件和表格文本文件。

这些辅助编程文件可以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。



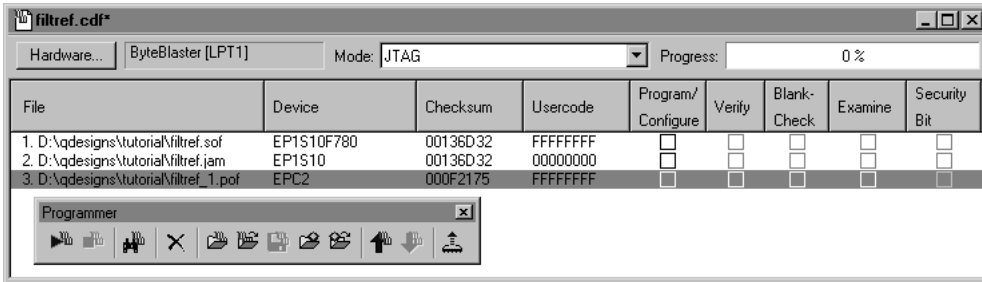
Programmer 使用 Assembler 生成的 POF 和 SOF 对 Quartus II 软件支持的所有 Altera 器件进行编程或配置。可以将 Programmer 与 Altera 编程硬件配合使用，例如，MasterBlaster™、ByteBlasterMV™、ByteBlaster™ II 或 USB-Blaster 下载电缆；或 Altera 编程单元 (APU)。

 使用独立的 Programmer

如果仅想使用 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链式描述文件 (.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 字节代码文件 和设计所用器件的从上到下顺序，以及链中器件的顺序。图 2 显示了 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=< 主题名称 > ←
```

Programmer 具有四种编程模式：

- 被动串行模式
- JTAG 模式
- 主动串行编程模式
- 套接字内编程模式

被动串行和 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。可以使用主动串行编程模式和 Altera 编程硬件对单个 EPCS1 或 EPCS4 串行配置器件进行编程。可以配合使用套接字内编程模式与 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 指定和连接至远程 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 Data Sheet</i> 、 <i>ByteBlaster II Parallel Port Download Cable Data Sheet</i> 、 <i>ByteBlasterMV Parallel Port Download Cable Data Sheet</i> 、 <i>MasterBlaster Serial/USB Communications Cable Data Sheet</i> 或 <i>USB-Blaster Download Cable Data Sheet</i>
编程硬件安装	<i>Quartus II Installation & Licensing for PCs</i> 和 <i>Quartus II Installation & Licensing for UNIX and Linux Workstations</i> 手册
特定器件的编程信息	Altera 网站上 <i>Stratix Device Handbook 手册</i> 的 <i>Configuring Stratix & Stratix GX Devices</i> 和 <i>Using Altera Enhanced Configuration Devices</i> 两章 Altera 网站上的 <i>Application Note 250 (Configuring Cyclone FPGAs)</i>

使用 Programmer 对一个或多个器件进行编程

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 字节代码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，并安装任何必要的启动程序。
2. 进行设计的全编译，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，被动串行模式、JTAG 模式、主动串行编程模式或套接字内编程模式。
6. 视编程模式而定，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 稳定器件，例如配置器件、MAX 3000 和 MAX 7000 器件，可以指定额外编程选项来查询器件，例如，**Verify**、**Blank-Check**、**Examine** 和 **Security Bit**。
8. 启动 Programmer。

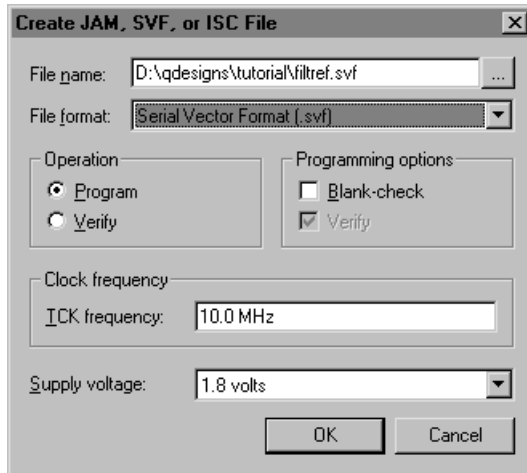
建立辅助编程文件

还可以使用其它格式（例如 Jam 文件、Jam 字节代码文件、串行矢量格式文件、系统内配置文件、原二进制文件或表格文本文件）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、局域更新的 POF、EPC16 的 HEXOUT 文件、SRAM 的 HEXOUT 文件或原编程数据文件。可以使用 **Create/Update > Create JAM, SVF, or ISC 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 **Device & Pin Options** 对话框（可以从 **Settings** 对话框（Assignments 菜单）的 **Device** 页进入）的 **Programming Files** 选项卡，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 字节代码文件、串行矢量格式文件或系统内配置文件。然后，这些文件可以与 Altera 编程硬件或智能主机配合使用，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 字节代码文件添加至 CDF。请参阅第 125 页上的图 3。

图 3. Create JAM, SVF, or ISC File 对话框



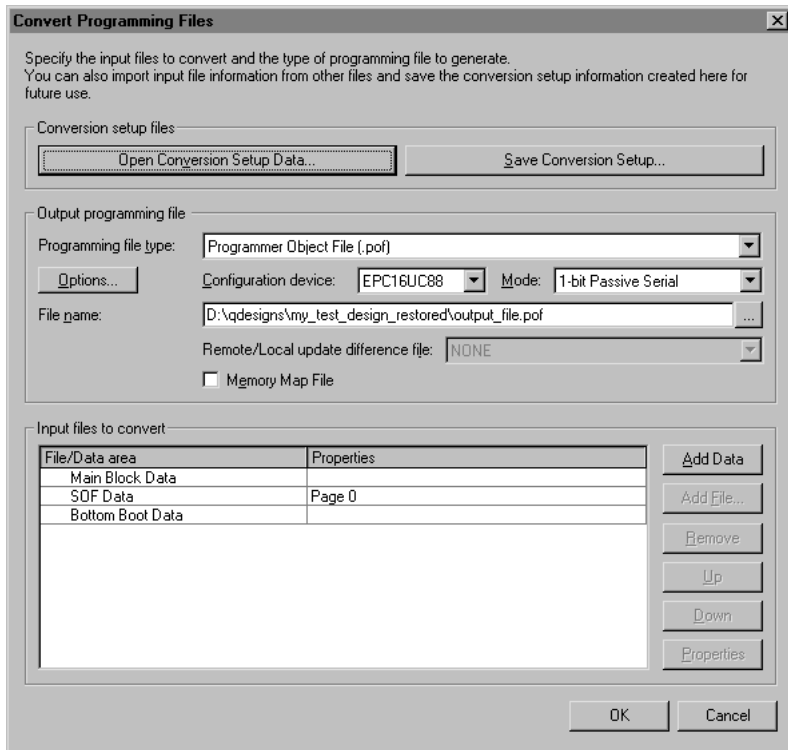
以下步骤描述建立 Jam 文件、Jam 字节代码文件、串行矢量格式文件或系统内配置文件：

1. 进行设计的全编译，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。
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. 选择 **Create/Update > Create Jam, SVF, or ISC File**（File 菜单）并指定要建立文件的名称和文件格式。

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，此 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 对象文件转换为供外部主机使用的 EPC16 的 HEXOUT 文件。也可以将 POF 转换为与某些配置器件一起使用的原编程数据文件。请参阅图 4。

图 4. **Convert Programming Files** 对话框



可以使用 **Convert Programming Files** 对话框，对 SRAM 的 HEXOUT 文件、POF、原二进制文件或表格文本文件中存储的 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转换设置文件 (.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。

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

- 建立不同的配置比特流，这些比特流是存储在配置存储器空间页面中。
- 在每个页面中建立 SOF 的并行链。
- 排列闪存中存储的 SOF 和十六进制（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。
- 若用户希望，也可以建立存储器映射文件（.map）。

对于局域更新的 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** 项的属性。
- 添加或删除 **SOF Data** 项。
- 用户希望的话，可以建立存储器映射文件，并生成远程更新差异文件和局域更新差异文件。

还可以使用 **Convert Programming Files** 对话框将多个 SOF 排列和组合为主动串行配置模式下的单个 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转换编程文件功能。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=<主题名称> ←
```

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（File 菜单）建立转换设置文件，并指定所要建立编程文件的格式和名称。
3. 指定与编程文件的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，为远程更新的 **Programmer** 对象文件或局部更新的 **Programmer** 对象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 数据、POF 数据和 HEX 文件的附加属性。
9. 转换文件。用户希望的话，还可以指定要建立的存储器映射文件。



有关信息

请参阅

系统内可编程性和电路内可重新配置性

Altera 网站上的 *Application Note 100 (In-System Programmability Guidelines)*

Altera 网站上的 *Application Note 95 (In-System Programmability in MAX Devices)*

Altera 网站上的 *Application Note 88 (Using the Jam Language for ISP & ICR via an Embedded Processor)*

Altera 网站上的 *Application Note 122 (Using Jam STAPL for ISP & ICR via an Embedded Processor)*

Altera 网站上的 *Application Note 85 (In-System Programming Times for MAX Devices)*

Altera 网站上的 *Application Note 298 (Reconfiguring Excalibur Devices under Processor Control)*

系统内编程

《Quartus II Tutorial》中的 Programming 模块

远程系统配置

Altera 网站上 *Stratix Device Handbook 的 Using Remote System Configuration with Stratix & Stratix GX Devices* 一章

使用 Quartus II 软件通过远程 JTAG 服务器进行编程

在 **Hardware Setup** 对话框（可从 **Programmer** 窗口的 **Hardware** 按钮或从 **Edit** 菜单进入）中，还可以添加可以联机访问的远程 JTAG 服务器，这样，就可以使用局域计算机未提供的编程硬件，以及配置局域 JTAG 服务器设置等，让远程用户可以连接到局域 JTAG 服务器。

可以在 **Configure Local JTAG Server** 对话框（从 **Hardware Setup** 对话框的 **JTAG Settings** 选项卡进入）中指定远程客户端应该先启用，才可以连接到 JTAG 服务器。

可以在 **Add Server** 对话框（可以从 **Hardware Setup** 对话框的 **JTAG Settings** 选项卡进入）中指定要连接到的远程服务器。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** 选项卡中。



有关信息

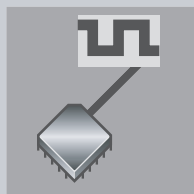
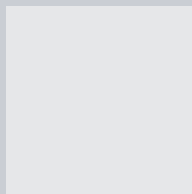
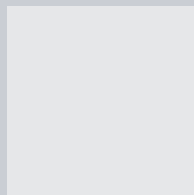
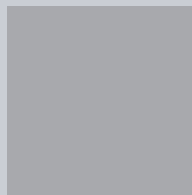
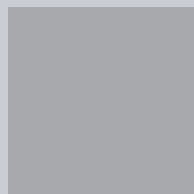
请参阅

使用 Local JTAG 服务器

《Quartus II Help》中的“Configuring Local JTAG Server Settings”和“Adding a JTAG Server”

第十章

调试



第 10 章内容：

简介	132
使用 SignalTap II 逻辑分析器	133
使用 SignalProbe	138
使用 Chip Editor	140

10

简介



Quartus® II SignalTap® II 逻辑分析器和 SignalProbe™ 功能可以分析内部器件节点和 I/O 引脚，同时在系统内以系统速度在运行。SignalTap II 逻辑分析器使用嵌入式逻辑分析器将信号数据通过 JTAG 端口送往 SignalTap II 逻辑分析器或者外部逻辑分析器或示波器。SignalProbe 功能使用未用器件路由资源上的递增式路由，将选定信号送往外部逻辑分析器或示波器。图 1 和图 2 显示了 SignalTap II 和 SignalProbe 调试流程。

图 1. SignalTap II 调试流程

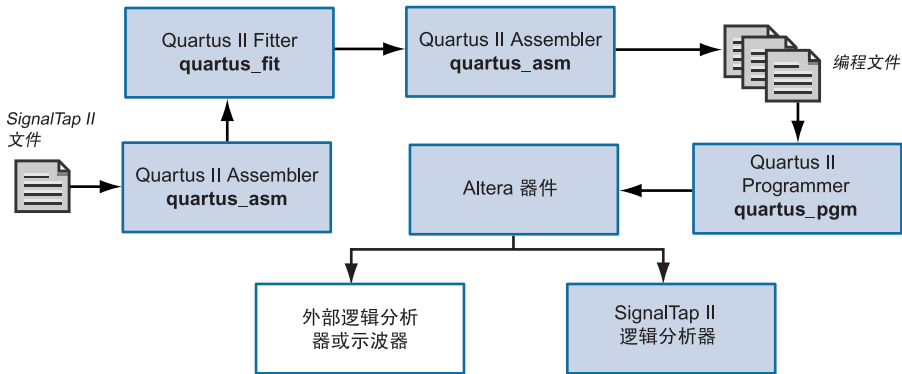
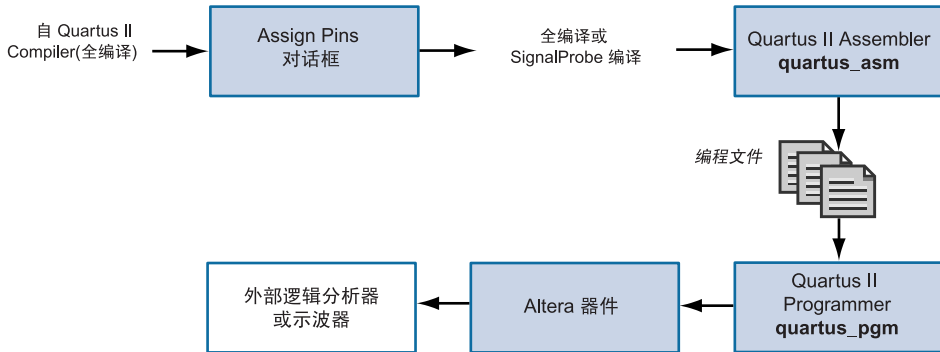


图 2. SignalProbe 调试流程



使用 SignalTap II 逻辑分析器



SignalTap II 逻辑分析器是第二代系统级调试工具，可以捕获和显示实时信号行为，允许观察系统设计中硬件和软件之间的交互作用。Quartus II 软件允许选择要捕获的信号、开始捕获信号的时间以及要捕获多少数据样。还可以选择是将数据从器件的存储器块通过 JTAG 端口路由至 SignalTap II 逻辑分析器，或是至 I/O 引脚以供外部逻辑分析器或示波器使用。

可以使用 MasterBlaster™、ByteBlasterMV™、ByteBlaster™ II 或 USB-Blaster 通信电缆下载配置数据到器件上。这些电缆还用于将捕获的信号数据从器件的 RAM 资源上载至 Quartus II 软件。然后，Quartus II 软件将 SignalTap II 逻辑分析器采集的数据以波形显示。

设置和运行 SignalTap II 逻辑分析器

若要使用 SignalTap II 逻辑分析器，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捕获到的信号。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器采集和分析数据。

每个逻辑分析器实例均嵌入器件上的逻辑之中。SignalTap II 逻辑分析器在单个器件上支持多达 1,024 个通道和 128K 数据样。

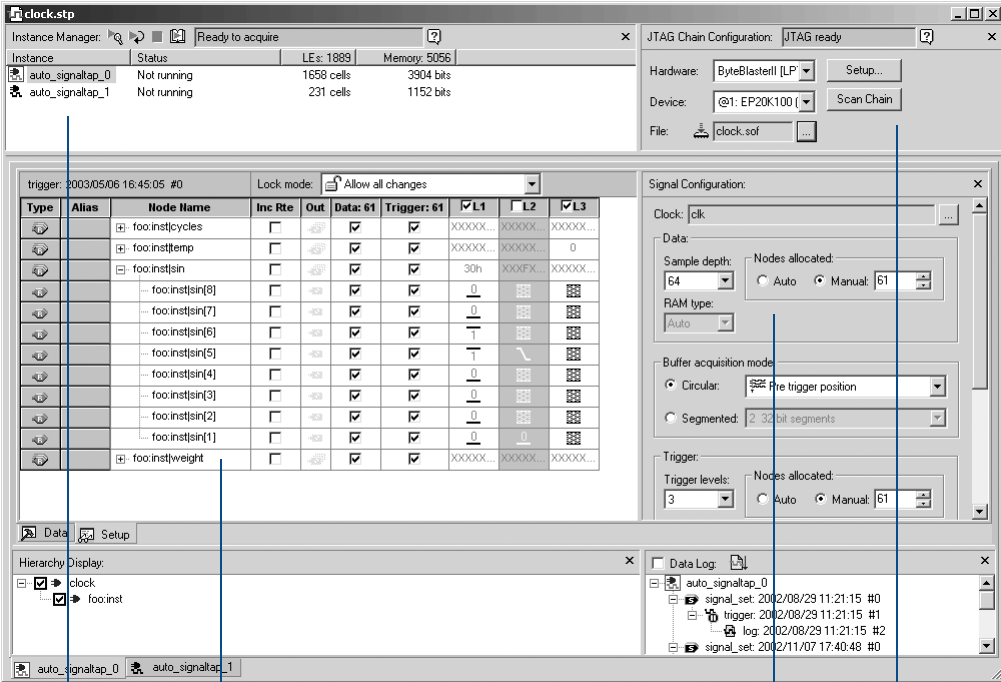
编译之后，可以使用 **Run Analysis** 命令（Processing 菜单）运行 SignalTap II 逻辑分析器。请参阅第 134 页上的图 3。

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布局布线后的 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. 编译设计。
6. 对器件进行编程。

7. 在 Quartus II 软件中或使用外部逻辑分析器或示波器采集和分析信号数据。

图 3. SignalTap II 逻辑分析器



实例管理器 设置视图

信号配置 面板 JTAG 链配置

可以使用以下功能设置 SignalTap II 逻辑分析器：

- **多个逻辑分析器：** SignalTap II 逻辑分析器在每个器件中支持逻辑分析器的多个嵌入式实例。可以使用此功能为器件中的每个时钟域建立单独且唯一的逻辑分析器，并在多个嵌入式逻辑分析器中应用不同的设置。
- **实例管理器：** 实例管理器允许在多个实例上建立并执行 SignalTap II 逻辑分析。可以使用它在 SignalTap II 文件中建立、删除和重命名实例。实例管理器显示当前 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元素和存储器比特的数量。实例管

理器可以协助检查每个逻辑分析器在器件上要求的资源使用量。可以选择多个逻辑分析器以及选择 **Run Analysis**（Processing 菜单）来同时启动多个逻辑分析器。

- **触发器：**触发器是逻辑级别和 / 或逻辑边缘方面的逻辑事件模式。SignalTap II 逻辑分析器支持多级触发、多个触发位置、多个段以及外部触发事件。可以使用 SignalTap II 逻辑分析器窗口中的 **Signal Configuration** 面板设置触发器选项。

可以给逻辑分析器配置最多十个触发器级别，使您可以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在触发器之前和触发器之后应采集的数据量。分段的模式允许通过将存储器分为周密的时间段，为定期事件捕获数据，而无需分配大采样深度。

- **递增路由：**递增路由功能允许在不执行完全重新编译的情况下分析布局布线后节点，从而有利于缩短调试过程。

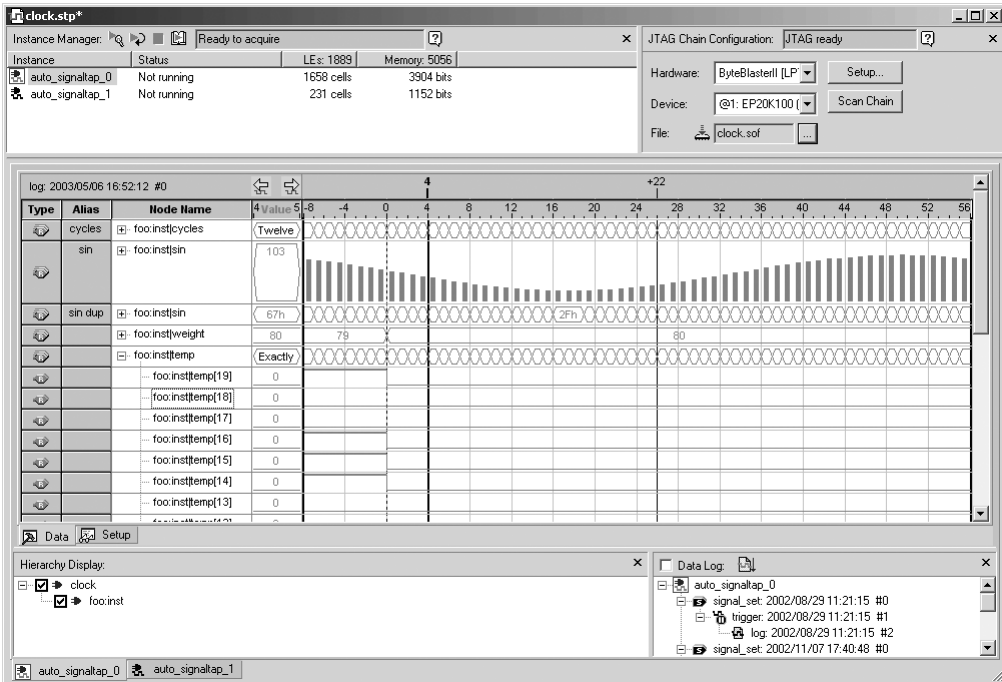
在使用 SignalTap II 递增路由功能之前，必须打开 **Settings** 对话框（Assignments 菜单）**SignalTap II 逻辑分析器页**中若在使用递增路由的 SignalTap II 中具备了条件，则自动开启智能编译来执行智能编译。此外，在编译设计之前，必须使用**已分配触发节点 (Trigger Nodes allocated)** 和**已分配数据节点 (Data Nodes allocated)** 框保留 SignalTap II 递增路由的触发或数据节点。可以在 Node Finder 的 **Filter** 列表中选择 **SignalTap II: post-fitting**，查找 SignalTap II 递增路由源的节点。

分析 SignalTap II 数据

在使用 SignalTap II 逻辑分析器查看逻辑分析的结果时，数据是存储在器件的内部存储器中，然后通过 JTAG 端口导入逻辑分析器的波形视图中。

在波形视图中，可以插入时间栏，对齐节点名称，复制节点；建立总线、重命名总线和取消总线组合；指定总线值的数据格式；以及打印波形数据。使用数据日志建立波形，此波形显示使用 SignalTap II 逻辑分析器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 4 显示了波形视图。

图 4. SignalTap II 波形视图

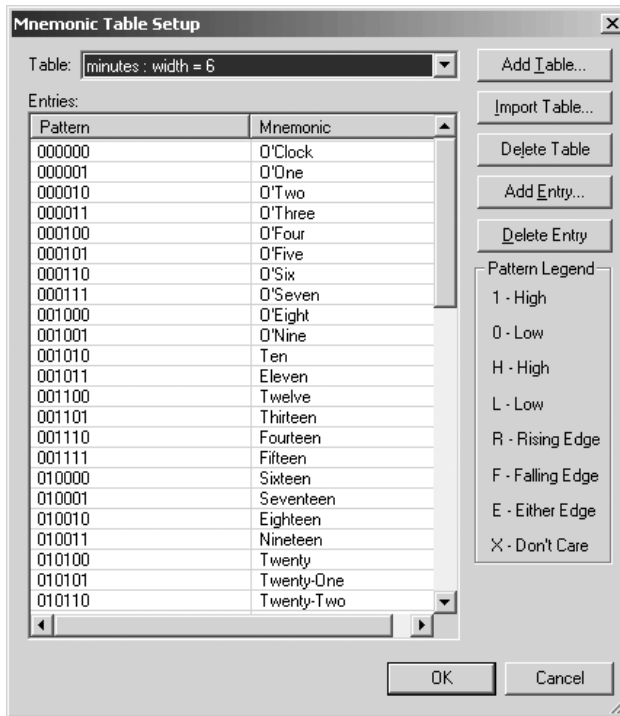


Waveform Export 实用程序允许将捕获的数据导出为 EDA 工具可以使用的以下工业标准格式：

- 逗号分隔值文件 (.csv)
- 表文件 (.tbl)
- 值更改转储文件 (.vcd)
- 矢量波形文件 (.vwf)

还可以配置 SignalTap II 逻辑分析器，为一组信号建立助记表。助记表功能允许将预定名称分配给一组位模式，使捕获的数据更有意义。请参阅图 5。

图 5. Mnemonic Table Setup 对话框



有关信息

使用 SignalTap II 逻辑分析器

请参阅

Altera 网站上的 *Application Note 280 (Design Verification Using the SignalTap II Embedded Logic Analyzer)*

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

使用 SignalProbe



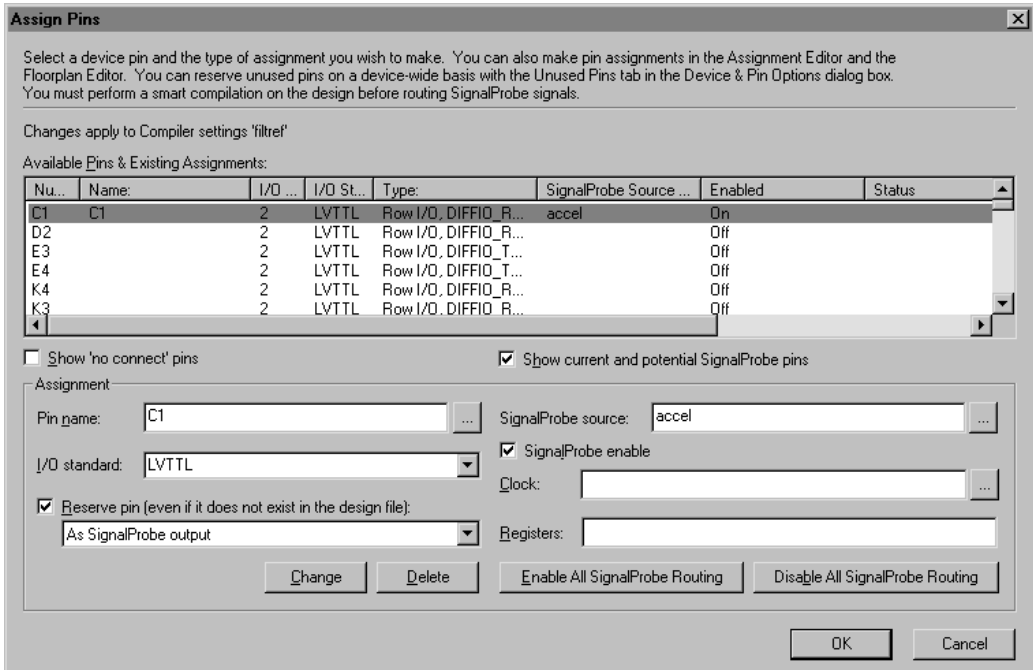
SignalProbe 功能允许在不影响设计中现有布局布线的情况下将用户特定的信号路由至输出引脚，从而无需另做一次全编译，就可以调试信号。从完全路由设计开始，可以选择和路由信号，通过以前保留或当前未使用的 I/O 引脚进行调试。

SignalProbe 功能允许指定设计中要调试的信号，然后执行一次 SignalProbe 编译，使那些信号与未使用或保留的输出引脚相连，再发送信号至外部逻辑分析器。在分配引脚、查找可用 SignalProbe 源时，可以使用 Node Finder。SignalProbe 编译通常大约花费正常编译所需时间的 10%。

若要使用 SignalProbe 功能中的保留引脚和对设计执行 SignalProbe 编译，请执行以下操作：

1. 进行设计的全编译。
2. 选择要调试的信号以及信号要通过的 I/O 引脚，然后打开 **Assign Pins** 对话框（可从 **Settings** 对话框（**Assignments** 菜单）的 **Device** 页访问）中的 SignalProbe 功能。请参阅第 139 页上的图 6。
3. 执行 SignalProbe 编译。另一种方法是打开 **Settings** 对话框 **Mode** 页中的 **编译期间自动路由 SignalProbe 信号 (Automatically route SignalProbe signals during compilation)**，然后选择 **Start Compilation**（**Processing** 菜单），使全编译中包含 SignalProbe 连接。
4. 给器件配置新编程数据，检查信号。

图 6. Assign Pins 对话框



在保留 SignalProbe 引脚时，还可以使用寄存器流水线功能使调试信号与时钟信号同步，从而消除从源信号到输出引脚的路由延时。

可以在调试之后保持或删除所有或部分 SignalProbe 路由。如果在设计中保持 SignalProbe 路由，可以在全编译期间自动进行 SignalProbe 路由选择。

还可以将 SignalProbe 功能与 Tcl 配合使用。可以使用 Tcl 命令，添加和删除 SignalProbe 分配和源，对设计执行 SignalProbe 编译，在全编译中编译已有路由的 SignalProbe 信号。



有关信息

请参阅

使用 SignalProbe 功能

Altera 网站上的 *Technical Brief 82 (SignalProbe Compilation Enables Fast System Debugging with the Quartus II Software)*

《Quartus II Help》中的 “SignalProbe Introduction”

TCL 命令与 SignalProbe 功能配合使用

Altera web 网站上的 *Application Note 195 (Scripting with Tcl in the Quartus II Software)*

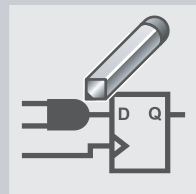
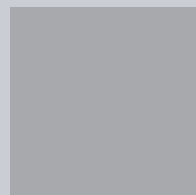
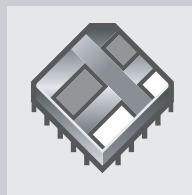
使用 Chip Editor

可以将 Chip Editor 与 SignalTap II 和 SignalProbe 调试工具一起使用，加快设计验证以及递增修复在设计验证期间未解决的错误。运行 SignalTap II 逻辑分析器或使用 SignalProbe 功能验证信号之后，就可以使用 Chip Editor 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 元素或 PLL 原子的属性和参数执行编译后编辑，而无需执行完全的重新编译。

有关使用 Chip Editor 的详细信息，请参阅下一章 “[第 11 章：工程更改管理](#)”

第十一章

工程更改管理



第 11 章内容：

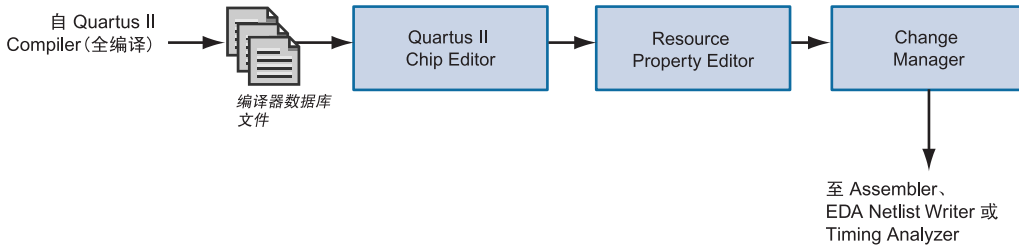
简介	142
使用 Chip Editor 识别延时与关键路径	143
使用 Resource Property Editor 修改资源属性	145
Change Manager 查看和管理更改	146
验证 ECO 更改的效果	148

11

简介

Quartus® II 软件允许在全编译之后对设计进行少量修改，通常称为工程更改纪录 (ECO)。可以直接在设计数据库上做这些 ECO 更改，而不是在源代码或设置和配置文件上做，这样就无需运行全编译来实施这些更改。图 1 显示了工程更改管理设计流程。

图 1. 工程更改管理的设计流程



以下步骤概述了 Quartus II 软件中工程更改管理的设计流程。

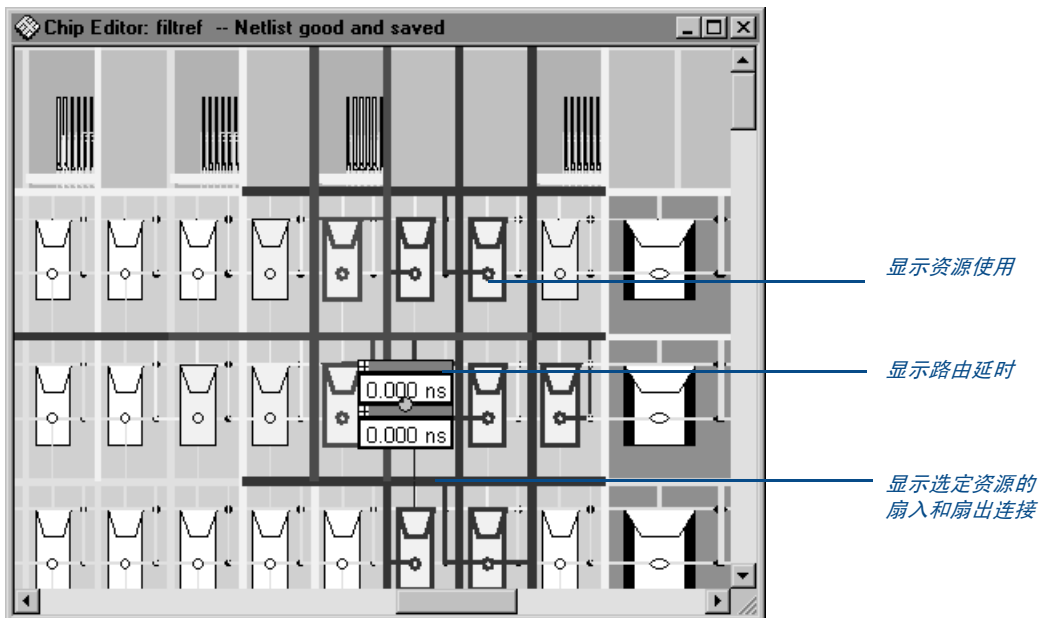
1. 全编译之后，使用 **Chip Editor** 查看设计布局布线详细信息，并确定要更改的资源。如果需要的话，可以使用 **Netlist Explorer** 过滤和高亮显示资源。
2. 使用 **Resource Property Editor** 编辑资源的内部属性。
3. 使用 **Check Resource Properties** 命令（**Edit** 菜单）检查资源更改的合法性。
4. 在更改管理器中查看更改的摘要和状态，并控制要实现和 / 或保存对资源属性的哪些更改。还可以添加备注，帮助您引用每个更改。
5. 使用 **检查和保存所有网表更改 (Check and Save 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。
6. 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**。

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 Floorplan Editor 中不显示的设计布局布线的其它详细信息。它显示完整的路由信息，显示每个器件资源之间的所有可能和使用的路由路径。请参阅图 2。

图 2. Chip Editor

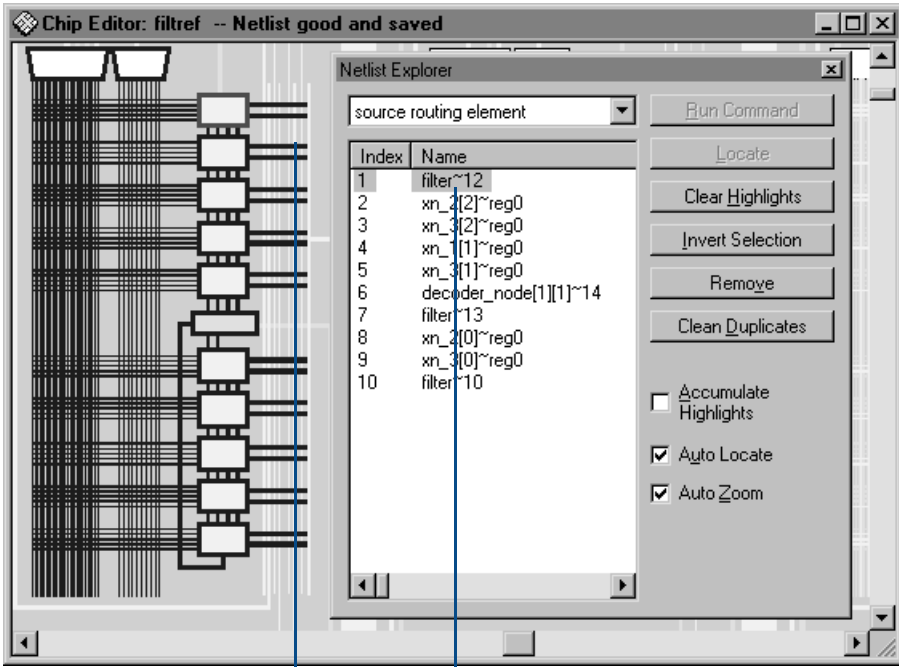


Chip Editor 显示器件的所有资源，例如，互连和路由线路、逻辑阵列块 (LAB)、RAM 块、DSP 块、I/O、行、列以及块、互连和其它路由线路之间的接口。

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（显示器件视图的缩放比例），控制 Chip Editor 显示的信息的详尽水平。还可以设置控制不同资源显示、扇入和扇出、关键路径和信号延时估计的选项。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 中选择资源，并选择 **Locate in Resource Property Editor**（右键弹出菜单），打开 Resource Property Editor 和编辑该资源。有关详细信息，请参阅“第 145 页使用 Resource Property Editor 修改资源属性”。

Chip Editor 还包括可以在 Chip Editor 中高亮显示和选择网表元素的 Netlist Explorer 窗口。请参阅图 3。

图 3. Netlist Explorer



Netlist Explorer 允许在 Chip Editor 中过滤和高亮显示资源

如果在 Chip Editor 中选择元素，它们将显示在 Netlist Explorer 的列表中。然后，可以应用不同的过滤和命令，例如，那些用于查找扇出或路由元素的命令，或用于根据停滞、名称等特定条件过滤列表的选项。此列表将根据所选的选项进行更新和过滤。可以通过重复这些步骤和应用不同的命令来继续对此网表进行“研究”。



有关信息

工程更改管理和使用 Chip Editor

使用 Chip Editor

请参阅

Altera 网站上的 *Application Note 310 (Using the Quartus II Chip Editor)*

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

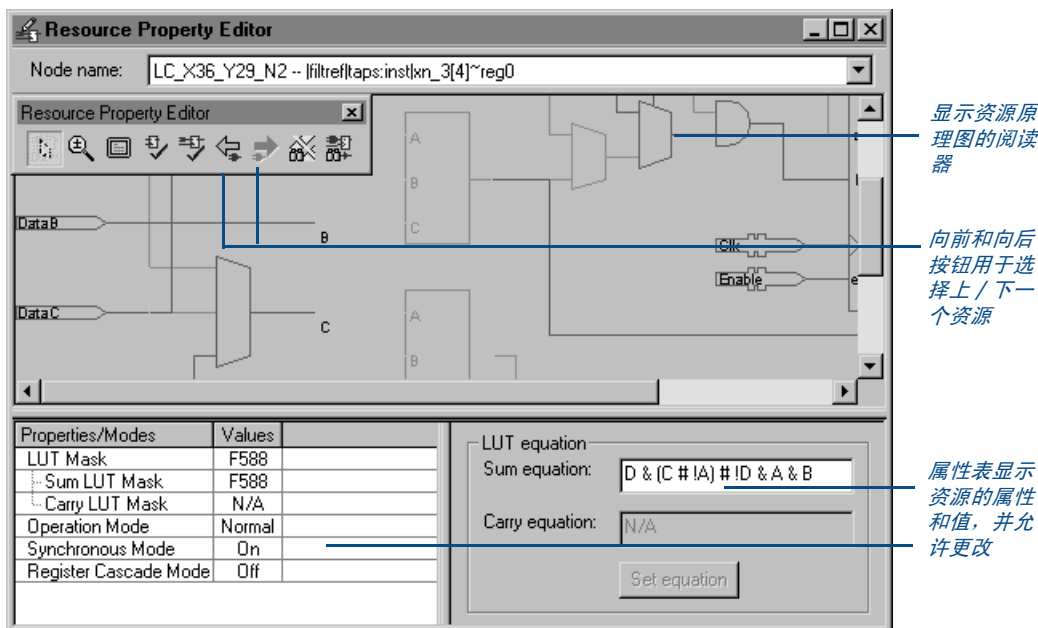
使用 Resource Property Editor 修改资源属性



Resource Property Editor 用于对逻辑单元、I/O 元素或 PLL 资源的属性和参数执行编译后编辑。可以使用工具栏按钮在资源中前后移动。还可以同时选择和更改多个资源。可以跟踪资源的扇入和扇出，并可以在 Resource Property Editor 中查看资源。

Resource Property Editor 包含一个阅读器和一个属性表，阅读器显示正在修改的资源的原理图，属性表显示该资源的可用属性和参数。请参阅第 145 页上的图 4。

图 4. Resource Property Editor



可以在原理图中或属性表中更改资源。如果在属性表中做更改，该更改将自动反映在原理图中。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。还可以在更改管理器中查看所做更改的摘要。有关详细信息，请参阅下一节“使用 Change Manager 查看和管理更改”。



有关信息

工程更改管理和使用 Resource Property Editor

使用 Resource Property Editor

请参阅

Altera 网站上的 *Application Note 310 (Using the Quartus II Chip Editor)*

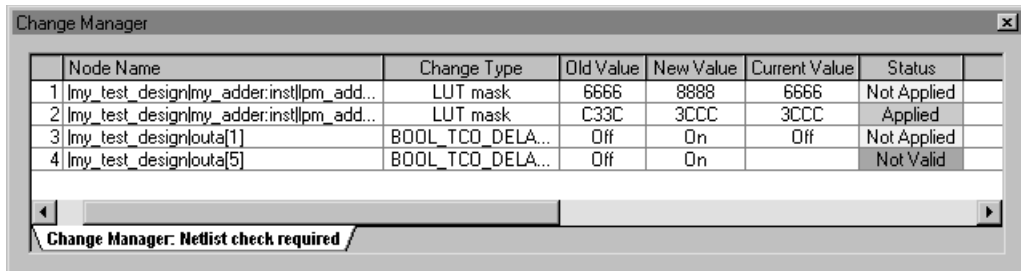
《Quartus II Help》中的“Overview: Using the Resource Property Editor”和“Making Post-Compilation Changes Introduction”

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。请参阅图 5。

图 5. Change Manager



Change Manager 的日志视图显示每个 ECO 更改的以下信息：

- 更改编号
- 节点名称
- 更改类型
- 旧值
- 新值
- 当前值
- 您添加的有关 ECO 更改的备注。
- 状态，可以为以下指示符之一：
 - **待定：**已在 Resource Property Editor 中做了更改，但尚未使用 **Check Resource Properties** 命令（Edit 菜单）对它进行检查，以确保它在单一资源环境中为合法的。
 - **已应用：**已在 Resource Property Editor 中做了更改，并且已使用 **Check Resource Properties** 命令成功地进行了检查。已在网表中输入更改，并且已在编译器数据库文件（.cdb）中保存了数值。
 - **无效：**已在 Resource Property Editor 中做了更改，并且已检查 **Check Resource Properties** 命令，但更改管理器认为更改不再有效，并且无法安全地或不能应用在设计上（例如，如果资源不再存在）。编译器数据库文件中的值与 **Old Value** 和 **New Value** 栏中显示的值不匹配。
 - **未应用：**已在 Resource Property Editor 中做了更改，并且已经使用 **Check Resource Properties** 命令对更改进行了检查，但设计是在 Quartus II 软件之外重新编译或修改。编译器数据库文件中的值与 **Old Value** 栏显示的值相匹配，但与 **New Value** 栏显示的值不匹配。如果更改的状态为**未应用**，仍可以使用 **Apply New Value** 命令（右键弹出菜单）手动应用更改。

应用所需更改之后，应选择 **Check and Save All Netlist Changes**（Edit 菜单）以检查网表中所有其它资源更改的合法性。

如果 ECO 更改要求重新运行 Compiler 的 Assembler 模块，它还将显示消息 POF not up-to-date，表示必须使用 Assembler 生成一个已更新的 Programmer 对象文件（.pof）。

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：

- 应用新值
- 全部应用新值
- 还原旧值
- 删除



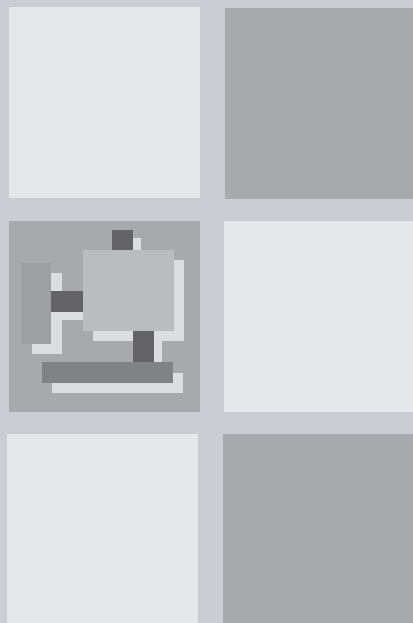
有关信息	请参阅
工程更改管理和使用 Change Manager	Altera 网站上的 <i>Application Note 310 (Using the Quartus II Chip Editor)</i>
使用 Change Manager	《Quartus II Help》中的“Overview: Using the Change Manager”和“Making Post-Compilation Changes Introduction”

验证 ECO 更改的效果

进行 ECO 更改之后，就不必再次运行全编译，尽管您可能需要运行 Compiler 的 Assembler 模块，以便建立新 POF，或者您可能需要再次运行 EDA Netlist Writer，以便生成新网表。您可能还想再次运行 Timing Analyzer 以验证所做的更改是否产生相应的时序改进。可以使用 Compiler Flow 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。

第十二章

系统级设计



12

第 12 章内容：

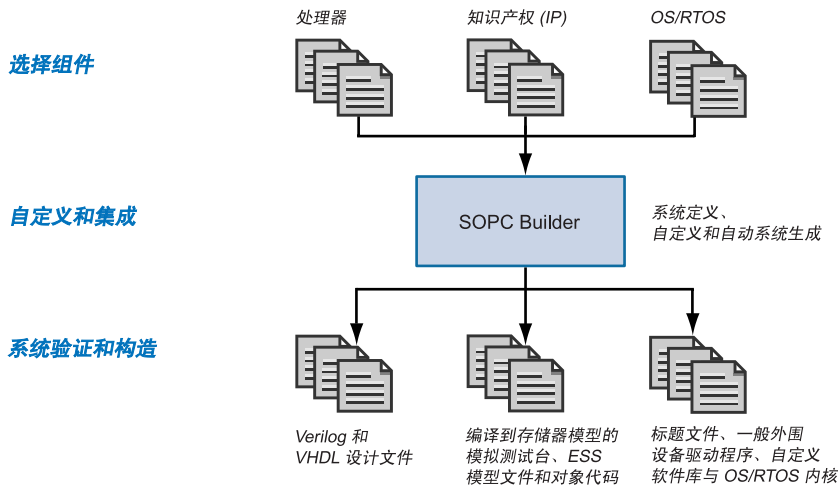
简介	150
使用 SOPC Builder 建立 SOPC 设计	151
使用 DSP Builder 建立 DSP 设计	154

简介

Quartus® II 软件支持 SOPC Builder 和 DSP Builder 的系统级设计流程。系统级设计流程使工程师能够以更高水平的抽象概念快速地设计和评估单芯片可编程系统 (SOPC) 体系结构和设计。

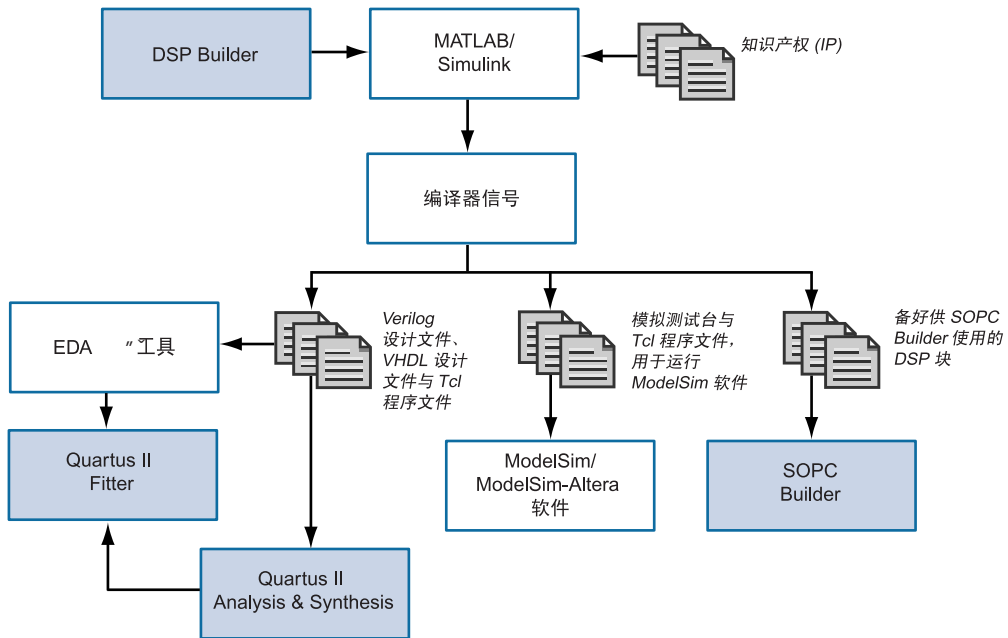
SOPC Builder 是自动化系统开发工具，可以有效简化建立高性能 SOPC 设计的任务。此工具能够完全在 Quartus II 软件中使系统定义和 SOPC 开发的集成阶段实现自动化。SOPC Builder 允许选择系统组件，定义和自定义系统，并在集成之前生成和验证系统。图 1 显示了 SOPC Builder 设计流程。

图 1. SOPC Builder 设计流程



Altera® DSP Builder 通过将 MathWorks MATLAB 和 Simulink 系统级设计工具的算法开发、仿真和验证功能与 VHDL 综合和仿真工具以及 Quartus II 软件相结合，将高级算法和 HDL 开发工具集成在一起。第 151 页上的图 2 显示了 DSP Builder 设计流程。

图 2. DSP Builder 设计流程



使用 SOPC Builder 建立 SOPC 设计



SOPC Builder 与 Quartus II 软件一起提供，它为建立 SOPC 设计提供标准化的图形环境，其中，SOPC 由 CPU、存储器接口、标准外围设备和用户自定义的外围设备等组件组成。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实例化的单个系统模块，并自动生成必要的总线逻辑，以将这些组件连接到一起。

SOPC Builder 库组件包括：

- 处理器
- 知识产权 (IP) 和外围设备
- 存储器接口
- 通信外设
- 总线和接口，包括 Avalon™ 总线和 AMBA™ 高性能总线 (AHB)
- 数字信号处理 (DSP) 内核
- 软件
- 标题文件

- 一般 C 驱动器
- 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 外设的嵌入式微处理器系统；但是，还可以生成不包括 CPU 的数据流系统。它允许指定具有多个主连接和从连接的总线拓扑结构。SOPC Builder 还可以导入或提供到达用户自定义逻辑块的接口，其中，逻辑块作为自定义外设连接到系统上。

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 可以导入或提供到达用户自定义逻辑块的接口。SOPC Builder 系统与用户定义逻辑配合使用时具有以下四种机制：简单的 PIO 连接、系统模块内实例化、到达外部逻辑的总线接口以及发布局域 SOPC Builder 组件。

SOPC Builder 提供用于下载的库组件（模块），包括 Excalibur 嵌入式处理器带区和 NIOS 处理器等处理器、UART、定时器、PIO、Avalon 三态桥接器、多个简单的存储器接口和 OS/RTOS 内核。此外，还可以从一系列的 MegaCore[®]、OpenCore[®] 和 OpenCore Plus 宏功能模块中进行选择。

可以使用 SOPC Builder 的 **System Contents** 页定义系统。可以在模块集中选择库组件，并在模块表中显示添加的组件。可以使用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
- 主连接和从连接
- 系统地址映射
- 系统 IRQ 分配
- 共享从连接的仲裁优先级
- 系统时钟频率

生成系统

SOPC Builder 中的每个工程包含系统描述文件（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。在生成系统期间，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。

完成系统设计之后，可以使用 SOPC Builder 的 **System Generation** 页或使用命令行生成系统。

SOPC Builder 软件自动生成所有必要逻辑，用以将处理器、外围设备、内存、总线、仲裁器、IP 内核及到达系统外逻辑和存储器的接口集成在一起，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，标题文件、一般外围设备驱动程序、自定义软件库和 OS/ 实时操作系统（RTOS 内核），以便在生成系统时提供完整的设计环境。

为了仿真，SOPC Builder 建立了 Model Technology™ ModelSim® 仿真目录，它包含 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、别名和最初的一组总线接口波形。它还建立仿真测试台，可以实例化系统模块、驱动时钟和复位输入，并可以实例化和连接仿真模型。

还生成 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 网站上的 *SOPC Builder Data Sheet*

与 *Quartus II* 软件一起安装的 *SOPC Builder User Guide*

《Quartus II Help》中的“Overview: Using SOPC Builder”

Altera 网站上的 *Application Note 308 (Building Embedded Processor Systems Using SOPC Builder & Excalibur Devices)*

使用 DSP Builder 建立 DSP 设计

DSP Builder 通过帮助您在易于算法应用的开发环境中建立 DSP 设计的硬件表示，缩短了 DSP 设计周期。DSP Builder 允许系统、算法和硬件设计者共享公共开发平台。DSP Builder 是个由 Altera 提供的一个可选软件包，并且 DSP 开发工具包中也包含它。

DSP Builder 还使用 SignalTap® II Logic Analyzer 对系统级调试提供支持。可以完全通过 MATLAB/Simulink 接口综合、编译和下载设计，然后执行调试。

实例化功能

可以将现有的 MATLAB 功能和 Simulink 块与 Altera DSP Builder 块和 MegaCore、OpenCore 和 OpenCore Plus 功能相结合，将系统级设计和实现与 DSP 算法开发相连。

要在设计中使用 MegaCore、OpenCore 和 OpenCore Plus 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

可以使用 Simulink 软件仿真设计，或使用 Simulink 软件中的 SignalCompiler makefile，用于在 EDA 仿真工具中仿真设计。

SignalCompiler 生成 Tcl 脚本和 VHDL 测试激励文件，其中，Tcl 脚本用于在 ModelSim 软件中进行 RTL 仿真，VHDL 测试激励文件用于导入 Simulink 输入激励源。可以在 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 VHDL 测试激励文件进行仿真。

生成综合文件

仿真之后，可以在 Quartus II、Mentor Graphics LeonardoSpectrum 或 Synplicity Synplify 软件中使用自动流程或在其它综合工具中使用手动流程，对 SOPC 设计进行综合。如果 DSP Builder 设计是最高层设计，可以使用自动或手动综合流程。如果 DSP Builder 设计不是最高层设计，必须使用手动综合流程。

可以使用自动流程从 MATLAB/Simulink 设计环境内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。可以从 Simulink 软件内指定用于设计的综合工具。

在手动流程中，SignalCompiler 生成 VHDL 设计文件和 Tcl 脚本，然后，可以用它们在 EDA 综合工具或 Quartus II 软件中进行手动综合，Quartus II 软件还允许指定您自己的综合或编译设置。生成输出文件时，SignalCompiler 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

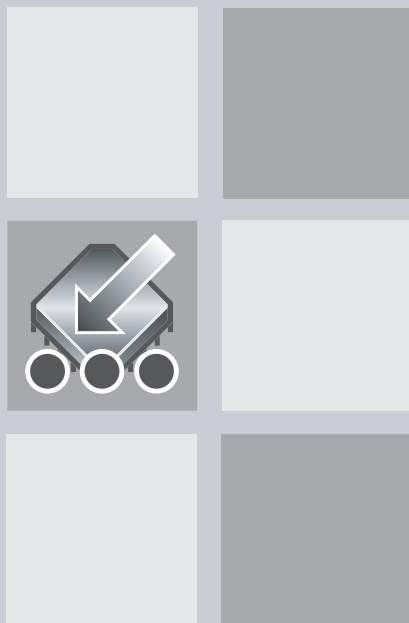
请参阅

使用 DSP Builder

Altera 网站上的 *DSP Builder User Guide*

第十三章

软件开发



第 13 章内容：

简介	158
在 Quartus II 软件中 使用 Software Builder	158
指定软件构建设置	159
生成软件输出文件	160

13

简介



Quartus® II Software Builder 是集成编程工具，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无源编程文件，或包含 Excalibur 器件的嵌入式处理器带区的存储器初始化数据的文件。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® 软件成套工具处理 Quartus II Text Editor 或其它汇编或 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++ 包含文件 (.h)
- C 源文件 (.c)
- C++ 源文件 (.cpp)
- 库文件 (.a)

Software Builder 可以在极少的帮助下在软件源文件上进行软件构建，并允许自定义对特定设计的处理。还可以使用 Software Builder 在软件构建期间或之后运行命令行命令，从 Quartus II 软件内为 Excalibur 器件运行程序或进程。

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，从 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=<主题名称> ←
```

指定软件构建设置

在执行软件构建之前，可以使用 **Software Build Settings** 向导或 **Settings** 对话框（Assignments 菜单）的 **Software Build Settings** 页，指定软件构建设置。

可以使用 **Software Build Settings** 向导或 **Settings** 对话框指定以下设置：

- 工程的软件构建设置名称
- **CPU 选项**：体系结构和软件工具集、字节顺序、输出文件名称、自定义构建和构建后命令行命令以及编程文件生成选项
- **C/C++ 编译器选项**：优化级别、预处理器定义和包含目录以及命令行命令
- **汇编器选项**：预处理器定义、额外包含目录和命令行命令
- **连接器选项**：对象文件、库文件、库目录、链接类型和命令行命令

在指定工具集之前，必须使用 **Settings** 对话框的 **Toolset Directories** 页指定 Software Builder 在软件构建期间使用的工具集目录。

生成软件输出文件

可以通过在 Quartus II 软件中进行软件构建，来处理设计和生成包含存储器初始化数据、无源编程文件和闪存编程文件的文件。还可以使用 **makeprogfile** 实用程序（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 软件之外生成无源编程文件和闪存编程文件。

有关使用 **makeprogfile** 实用程序的详细信息，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 生成无源编程文件时，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的存储器区域中的每个地址指定初始化数据。

表 1. 仿真器初始化文件

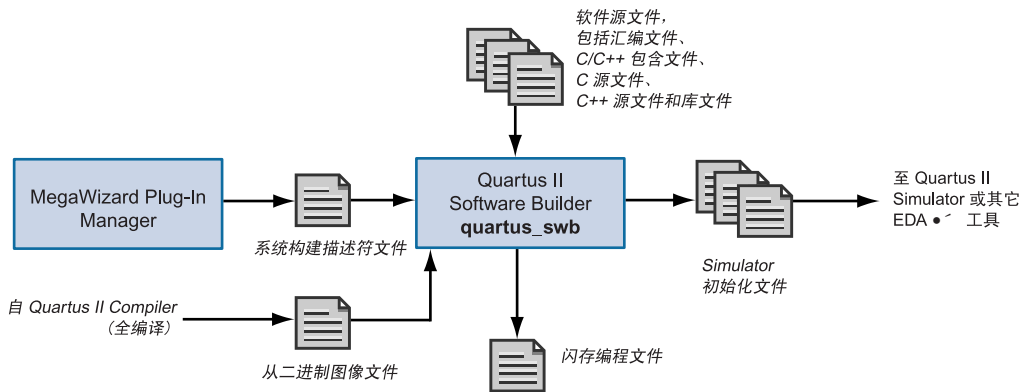
文件名称	文件内容
memory.regs	寄存器初始化数据
memory.sram0	SRAM0 初始化数据
memory.sram1	SRAM1 初始化数据
memory.dpram0	DPRAM0 初始化数据
memory.dpram1	DPRAM1 初始化数据

生成闪存编程文件

闪存编程文件是十六进制（Intel 格式）文件（.hex），它对闪存进行编程，Excalibur 器件就是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**，生成系统构建描述文件（.sbd）。
3. 若要闪存编程文件包含 Excalibur 器件的可编程逻辑器件 (PLD) 部分的配置数据，请编译设计，生成从二进制图像文件（.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**，并且如果正在使用从二进制图像文件，也请在 **Settings** 对话框（**Assignments** 菜单）的 **CPU** 页中指定可选的从二进制图像文件。
5. 启动软件构建。

图 1. 闪存编程文件流程



若要生成闪存编程文件，Software Builder 执行以下步骤：

- 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 嵌入式处理器带区的存储器初始化数据。

- 从 HEX 文件、系统构建描述文件和从二进制图像文件上建立启动数据对象文件。
- 连接器将启动数据文件与二进制启动载入文件相连，建立可执行和可链接格式文件 (**.elf**)。
- 代码转换器将可执行和可链接格式文件转换为名为 **<project name>_flash.hex** 的闪存编程文件。

然后可以使用 **exc_flash_programmer** 实用程序通过扩展总线 0 接口 (EBI0) 将闪存编程文件中的信息编程至 Excalibur 器件的闪存中。

生成无源编程文件

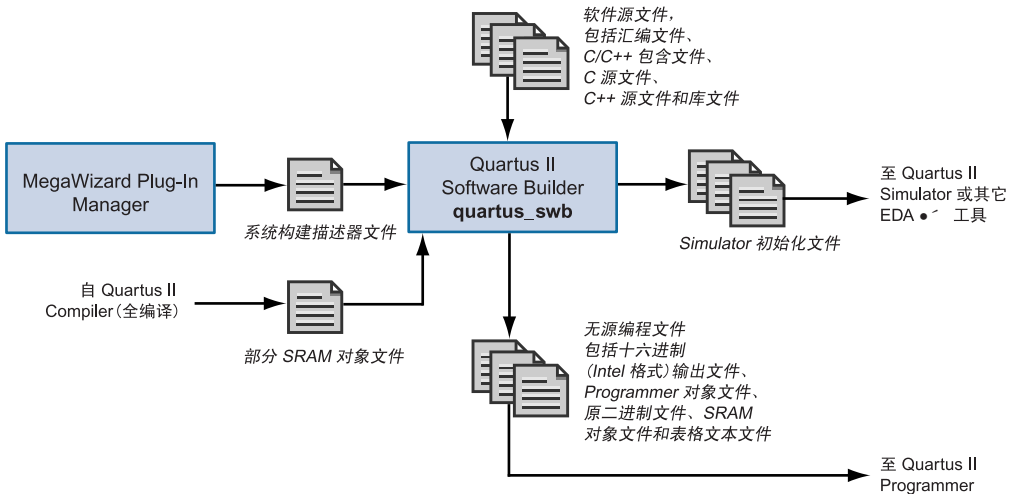
无源编程文件用于使用被动并行异步配置模式 (PPA)、无源并行同步 (PPS) 或被动串行 (PS) 配置方案对 Excalibur 器件进行配置。可以使用 **Software Builder**、**makeprogfile** 实用程序或 **Compiler** 生成以下无源编程文件：

- 十六进制 (Intel 格式) 输出文件 (**.hexout**)
- Programmer 对象文件 (**.pof**)
- 原二进制文件 (**.rbf**)
- SRAM 对象文件 (**.sof**)
- 表格文本文件 (**.ttf**)

以下步骤描述使用 **Software Builder** 建立无源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**，生成系统构建描述文件。
3. 编译设计，生成可编程逻辑部分 SRAM 对象文件 (**.psof**)。
4. 指定软件工具集目录和软件构建设置。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) 的 **CPU** 页中指定 **PSOF**。
5. 启动 **Software Builder**。

图 2. 无源编程文件流程



要生成无源编程文件，Software Builder 执行以下步骤：

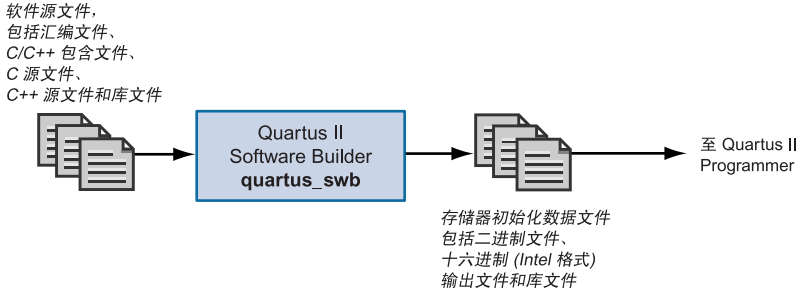
- 汇编器、C/C++ 编译器、连接器和代码转换器将软件源文件转换为 HEX 文件，它包含 Excalibur 器件的 Excalibur 嵌入式处理器带区的存储器初始化数据。
- **makeprogfile** 实用程序处理 HEX 文件、系统构建描述文件和 PSOF，建立一个或多个无源编程文件。

生成存储器初始化数据文件

二进制文件 (.bin)、HEX 文件和库文件 (.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 文件和库文件的基本流程：

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) 的 **CPU** 页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无源编程文件，请选择 **Programming file generation** 项下的 **None**。
3. 开始软件构建。

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：

- 汇编器和 C/C++ 编译器根据设计的软件源文件生成中间对象文件。
- 如果正在生成 BIN 文件或 HEX 文件时，则连接器连接对象文件并生成中间 ELF 文件，而且代码转换器将 ELF 文件转换为 BIN 文件或 HEX 文件。
- 如果正在生成库文件，则 Software Builder 使用 Software Builder Archiver 处理对象文件，使之进入库文件中。



有关信息

请参阅

进行软件构建

《Quartus II Help》中的“Overview: Using the Software Builder”

Altera 网站上的 *Application Note 299 (System Development Tools for Excalibur Devices)*

《Quartus II Tutorial》中的 Excalibur 模块

为 POF 和 SOF 生成无源编程文件和可选编程文件

《Quartus II Help》中的“Generating Passive Programming Files”

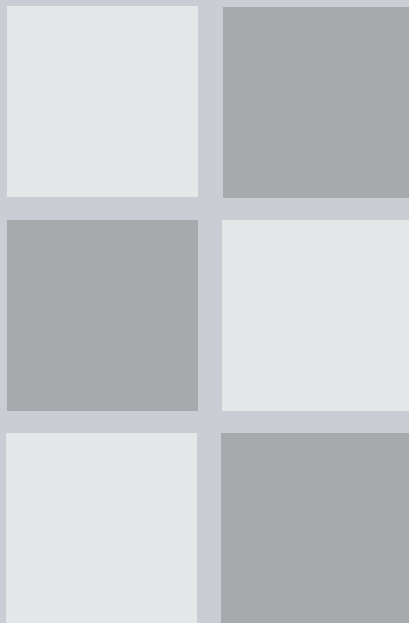
生成 BIN 文件、HEX 文件和库文件并生成调试信息

《Quartus II Help》中的“Generating Binary Files, Hexadecimal (Intel-Format) Files, Library Files & Motorola S-Record Files”

《Quartus II Help》中的“Overview: Checking Software Source Files and Output Files”

第十四章

安装、许可与
技术支持



14

第 14 章内容：

安装 Quartus II 软件	166
许可 Quartus II 软件	167
获取技术支持	169

安装 Quartus II 软件

可以在以下平台上安装 Quartus® II 软件：

- Pentium PC，运行速度为 400 MHz 或更快，并采用以下操作系统之一：
 - Microsoft Windows NT 4.0 版本（Service Pack 4 或更新版本）
 - Microsoft Windows 2000
 - Microsoft Windows XP
- Pentium PC，速度为 400 MHz 或更快，运行 Red Hat Linux 7.1、7.2 或 8.0 版
- 运行 Solaris 7 版本或 8 版本的 Sun Ultra 工作站
- HP 9000 Series 700/800 工作站，运行 HP-UX 11.0 版，ACE 日期为 1999 年 11 月或更后



有关信息

请参阅

系统要求和安装说明

Quartus II Installation & Licensing for PCs 手册

Quartus II Installation & Licensing for UNIX and Linux Workstations 手册

Altera CD 安装指导 (Installation Guide)

有关磁盘空间和存储器的特定信息

Quartus II **readme.txt** 文件

有关新功能、设备支持、EDA 接口支持的最新信息

Altera 网站上的 *Quartus II Software Release Notes*

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。可以订购 Altera 的以下软件：

- Altera Quartus II 软件
- Altera MAX+PLUS® II 软件
- Model Technology™ ModelSim® – Altera 软件

Altera 提供多种类型的软件订购。表 1 显示了提供的不同许可和订购选项。

表 1. Altera 许可和订购选项

许可名称	描述
FIXEDPC	与软件保护（T 保护或“dongle”）捆绑的单独 PC 许可证
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 用户提供的浮动网络许可
FLOATLNX	向运行 Red Hat Linux 和使用 PC、UNIX 或 Linux 许可服务器的 PC 用户提供的浮动网络许可
Quartus II Web 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 Web 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，并获得有关获取该软件许可的指令。

如果没有有效的许可文件，您应请求新的许可文件；然而，您还可以选择 30 天试用版，用以评估 Quartus II 软件，但它没有编程文件支持。要使用 30 天试用版，在您启动 Quartus II 软件后，请选择 **Enable 30-day evaluation period** 选项。30 天试用期结束后，您必须取得有效的许可文件才能使用该软件。

以下步骤描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**，将出现含 **Request updated license file from the web** 选项的提示信息。此选项显示 Altera 网站的“许可”部分，它允许您请求许可文件。

或

如果想稍后请求许可文件，可以进入 Altera 网站 www.altera.com/licensing 的“许可”部分。

2. 选择相应许可类型的链接。请参阅第 167 页上的表 1。
3. 指定请求的信息。
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. 如有必要，可以修改许可文件。
6. 为您的系统设置和配置 FLEXlm 许可管理器服务器。
7. 如果启动 Quartus II 软件，但尚未指定许可文件位置，将出现 **Specify valid license file** 选项。此选项显示 **Options** 对话框（Tools 菜单）的 **License Setup** 选项卡。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。



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	<i>Quartus II Installation & Licensing for PCs</i> 手册 <i>Quartus II Installation & Licensing for UNIX and Linux Workstations</i> 手册
有关 Quartus II 许可的一般信息	《Quartus II Help》中的“Overview: Obtaining a License File”获取许可文件“Specifying a License File”
有关 Altera 软件许可的详细信息	Altera 网站上的 <i>Application Note 205 (Altera Software Licensing)</i>
有关 Altera 软件许可的进一步信息	Altera 网站上的 <i>Application Note 229 (Troubleshooting 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 软件进行了登记；但是，要使用 mySupport 网站查看和提交服务请求，还必须注册一个 Altera.com 帐户。注册一个 Altera.com 帐户只为了使用 mySupport 网站；但是，有了 Altera.com 帐户，可以更简便地使用 Altera 网站的其它许多功能，例如“Download Center”、“Licensing Center”“Altera Technical Training”的在线课程注册或“Buy On-Line-Altera eStore”功能。

要注册一个 Altera.com 帐户用户名称和密码，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
 - ✓ 将网络浏览器指向 mySupport 网站 **www.altera.com/mysupport**。
2. 按照 mySupport 网站上的步骤注册一个 Altera.com 帐户。

即使您不是当前的 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 (第 1 部分, 共 2 部分)

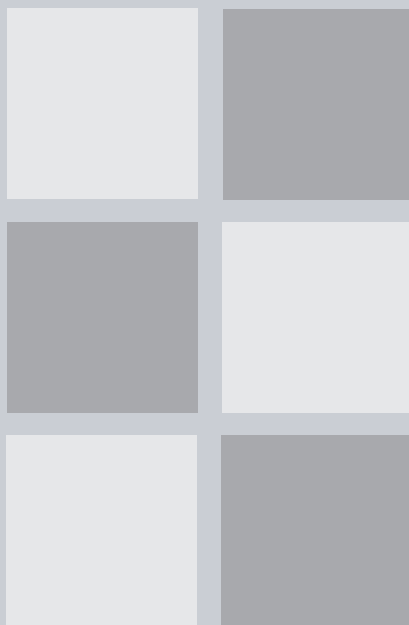
资源	描述
Altera 网站	www.altera.com Altera 网站提供 Altera 及其所有产品的信息。
支持中心	www.altera.com/support Altera 网站的“支持中心”部分允许您访问 mySupport 网站，还提供“知识数据库”。此外，它还提供软件和器件支持信息以及可以集成到设计中的设计示例。

表 2. Quartus II 技术支持资源 (第 2 部分, 共 2 部分)

资源	描述
mySupport 网站	<p>www.altera.com/mysupport 或在 Quartus II 软件中选择 Altera on the Web > Quartus II Home Page (Help 菜单)。</p> <p>mySupport 网站允许提交、查看和更新技术支持服务请求。</p>
知识数据库	<p>www.altera.com/kdb</p> <p>“知识数据库”帮助您寻找技术问题的答案，并允许在解决方案文档和其它 Altera 文献中搜索关键字和短语。</p>
电话	<p>(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。</p> <p>(408) 544-7000 (太平洋标准时间早上 7:00 至下午 5:00，周一至周五)</p>

第十五章

文档和其它资源



15

第 15 章内容：

获取联机帮助	172
使用 Quartus II 联机教程	173
其它 Quartus II 软件文档	174
其它 Altera 文献	175

获取联机帮助

Quartus® II 软件包括不依赖于平台的帮助系统，提供 Quartus II 软件的综合文档以及 Quartus II 软件生成的特定消息的详细信息。可以使用以下方法之一查看帮助：

要通过帮助主题列表搜索 选择 **Index**（Help 菜单），使用 **Index** 选项卡进行搜索。

要在帮助系统的全文中搜索 选择 **Search**（Help 菜单），使用 **Search** 选项卡进行搜索。

要搜索帮助主题类别大纲 选择 **Contents**（Help 菜单），可以查看 **Contents** 选项卡。

要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）。还可以选择消息 **s**（Help 菜单），查看所有消息的滚动列表。

要获取菜单命令或对话框的帮助 从高亮显示的菜单命令或运行对话框中按 F1，即可获取该项的上下文相关帮助。

要查找术语的定义 选择 **Glossary**（Help 菜单），可以看到 **Glossary** 列表。

使用帮助主题

要从 **Contents** 选项卡中打印帮助，请选择要打印的帮助文件夹或个别帮助主题，然后选择 **Print**（右键弹出菜单）或在工具栏上单击 **Print** 按钮。如果选择要打印的帮助文件夹，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帮助主题。

要在打开的 Quartus II 帮助主题中搜索关键字，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 帮助

《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 联机教程

联机教程介绍 Quartus II 设计软件的功能。它指导您如何简单快捷地建立和处理您自己的逻辑设计。基本和高级教程的模块化设计允许您选择要学习的 Quartus II 软件的部分：

- 基本教程教导您进行采样有限脉冲应答 (FIR) 滤波器设计（称为 **fir_filter**）的建立、时序分析、仿真和编程所需采取的步骤。
- 高级教程在基本教程培训的基础上，重点讲述 LogicLock 功能以及 Excalibur 和 Stratix 器件功能。

在成功安装 Quartus II 软件之后要启动 Quartus II 教程：

- ✓ 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。



使用 Quartus II 教程搜索文献

如果要完成基本或 LogicLock 教程，必须安装支持 APEX 20K EP20K100E 设备的软件。此外，如果要完成 Excalibur 和 Stratix 高级教程模块，必须安装支持 Excalibur EPXA10 和 Stratix EP1S25 设备的软件。

此教程是设计用于联机显示。然而，如果您要打印一个或多个教程模块，请单击每个模块开始处的 **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	描述	位置
<i>Quartus II Software Release Notes</i>	提供有关新功能、设备支持、EDA 接口支持的最新信息，以及已知问题和解决方法	Altera® 网站
<i>Quartus II Installation & Licensing for PCs</i> 手册	提供有关软件要求、安装和 PC 许可的详细信息	在 Quartus II 订购包中和 Altera 网站上
<i>Quartus II Installation & Licensing for UNIX and Linux Workstations</i> 手册	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 订购包中和 Altera 网站上
<i>Altera CD Installation Guide</i>	提供 Quartus II 订购包中所有 Altera CD-ROM 的基本安装指令	在 Quartus II 订购包中和 Altera 网站上
Quartus II readme.txt 文件	提供有关内存、磁盘空间和系统要求的信息	在 Quartus II 软件 CD-ROM 上，并与 Quartus II 软件一起安装
<i>Quartus II Software Quick Start Guide</i>	显示如何为目标器件设置工程、设置时序要求以及编译工程	在 Quartus II 订购包中和 Altera 网站上

其它 Altera 文献

Altera 网站 www.altera.com 的“文献”部分提供许多与 Quartus II 软件相关的主题的文档。其中有许多文档还可以从 Altera 文档库 CD 或 Altera 文献服务中获取。还可以从 ShopAltera 网站 www.shopaltera.com 购买印刷版的文档集。

Altera 提供包括以下一些主题文献：

- Quartus II 功能和有关设计流程中使用这些功能的向导
- Altera 器件功能、函数、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、用途和安装
- 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，并依 Altera 客户请求提供其它信息。



在知识数据库中搜索 Altera 文献

可以使用知识数据库（位于 Altera 网站 www.altera.com/kdb 的“支持中心”部分）在 Altera 网站的所有文献中搜索关键字或短语。

修订历史记录

手册 *Quartus II 简介* 3.0 版 1 修订本中包含的信息取代以前版本中发布的信息。

此版本有少量的印刷变动。

索引

A

Add/Remove 页 25
ADS Standard Tools 软件工具集 158
AHDL 29
AHDL 包含文件 (.inc) 27
Altera Megafunction Partners Program (AMPP) 31
Altera on the Web 命令 169
Altera 编程单元 (APU) 121
Altera 网站 169
Altera 硬件描述语言 (AHDL) 29
Altera.com 帐户 169
AMBA 高性能总线 (AHB) 151
AMPP 31
Analysis & Elaboration 44
Analysis & Synthesis 4
 VHDL 和 Verilog HDL 支持 45
 设计流程 44
 使用 EDA 工具执行 48
 网表优化 50
APU 121
ARM-based Excalibur MegaWizard Plug-In 161, 162
Assembler 4, 120, 121
Assign Pins 对话框 38
Assignment Editor 36, 99, 104
Avalon 总线 151

B

Back-Annotate Assignments 命令 91, 113
Block Editor 27
ByteBlaster II 下载电缆 121, 133
ByteBlasterMV 下载电缆 121, 133

C

C 源文件 (.c) 158
C++ 源文件 (.cpp) 158
Compiler
 模块 4

 启动 4
 指定设置 37
 状态 71

Compiler Database Interface 4
Compiler Settings 指南 36
Compiler 的模块 4
Convert Programming Files 命令 124
CPU 页 161, 162, 163
Create Jam, SVF, or ISC File 命令 124
Create/Update 命令 28, 29

D

Design Assistant 4, 52, 75
Design Assistant 页 52
DSE 78
dse.tcl Tcl 脚本 78
DSP Builder 150, 154
 SignalCompiler 154
 建立设计 154
 设计流程 151
 生成综合文件 154
 生成仿真文件 154
 实例化功能 154
 与其它 EDA 工具布局布线使用 154

E

ECO 142
 建立 145
 验证 148
EDA Netlist Writer 4, 57, 59, 106
EDA Tool Settings 页 48, 58
EDA 工具
 时序分析 106
 时序仿真 61, 62
 功率估算 60
 功能仿真 61
 综合 48
 仿真 57
 设置 48
 使用 LogicLock 95

- 支持的工具 8, 48, 57, 106
 - 指定设置 37, 58
 - 最少时序分析 106
 - EDA 界面 7, 16
 - EDIF 输入文件 (.edf) 26
 - EDIF 网表文件 (.edf) 44, 48
 - ESS 模型 67
 - exc_flash_programmer** 实用程序 162
 - Excalibur 带区仿真器 (ESS) 模型 67
 - Excalibur 设计
 - 仿真 66
 - Export LogicLock Regions** 命令 92
- F**
- Field View** 命令 110
 - Fitter 4, 70
- G**
- GNUPro for ARM 软件工具集 158
- I**
- Import LogicLock Regions** 命令 92
 - Import MAX+PLUS II Assignments** 命令 38
 - Integrated Synthesis 45
- J**
- Jam 文件 (.jam) 121, 123, 124
 - Jam 字节代码文件 (.jbc) 121, 123, 124
 - JTAG 端口 132
- L**
- List Paths** 命令 104
 - list_paths Tcl 命令 105
 - LMF 45
 - Locate in Timing Closure Floorplan** 命令 104
 - LogicLock 86, 87
 - 保存中间综合结果 91
 - 使用 Tcl 90
 - 与其它 EDA 工具布局布线使用 95
 - LogicLock Region Properties** 对话框 76, 88, 116
 - LogicLock Regions 窗口 88
 - LogicLock 区域 87
 - 查看连接 112
 - 查看区域内延迟 112
 - 导出 92
 - 导入 92
 - 基于路径的分配 116
 - 软 LogicLock 区域 116
 - 实现时序逼近 116
 - 属性 87
 - LPM 30
- M**
- makeprogfile** 实用程序 159
 - MasterBlaster 下载电缆 121, 133
 - MATLAB/Simulink 环境 154
 - MAX+PLUS II 分配与配置文件 (.acf) 25, 38
 - MAX+PLUS II 符号文件 (.sym) 28
 - MAX+PLUS II 仿真器通道文件 (.scf) 65
 - MegaCore 函数 31
 - MegaWizard Plug-In Manager** 30, 159
 - qmegawiz 可执行文件 12
 - 独立版本 12
 - 使用 black-box 方法 33
 - 使用 clear-box 方法 34
 - Memory Editor 60
 - Messages 窗口 71
 - mySupport 网站 169
- N**
- NativeLink 60, 106
 - Netlist Explorer 144
 - Netlist Optimizations** 页 51, 113
 - New Project Wizard** 25
- O**
- OpenCore Plus 硬件评估功能 31
 - OpenCore 评估功能 31

P

Path 对话框 116
 Perl 程序 14
 POF 120, 123, 124
 PowerFit Fitter 70
Priority 对话框 89
 Programmer 120
 quartus_pgmw 可执行文件 12
 独立版本 12, 121
 Programmer 对象文件 (.pof) 120, 123, 124, 162
Programming Files 选项卡 124

Q

qmegawiz 可执行文件 12
 Quartus II 教程 173
 Quartus II 软件
 EDA 工具设计流程 7, 16
 GUI 设计流程 3
 命令行设计流程 11
 一般设计流程 2
 Quartus II 工程配置文件 (.quartus) 25
quartus_asm 可执行文件 12, 121
quartus_cdb 可执行文件 12, 92
quartus_cpf 可执行文件 13, 127
quartus_drc 可执行文件 12, 53
quartus_eda 可执行文件 12, 60, 106
quartus_fit 可执行文件 12, 71
quartus_map 可执行文件 12, 25, 44
quartus_pgm 可执行文件 12
quartus_pgmw 可执行文件 12
quartus_sh 可执行文件 13
quartus_sim 可执行文件 12, 65
quartus_swb 可执行文件 13, 158
quartus_tan 可执行文件 12, 102

R

RAM 初始化文件 (.rif) 60
 Regions 窗口 76
Run EDA Simulation Tool 命令 59
Run EDA Timing Analysis Tool 命令 59

S

Settings 对话框 37, 76, 99
 Shop Altera 网站 174
 SignalProbe 功能 132, 138
 保留引脚 139
 编译 138
 设计流程 132
 使用 138
 SignalTap II 逻辑分析器 132, 133
 触发器 135
 递增路由 135
 多个分析器 134
 分析数据 135
 设计流程 132
 设置和运行 133
 实例管理器 134
 助记表 136
SignalTap II 逻辑分析器 页 135
 SignalTap II 文件 (.stp) 133
 Simulator 64
 使用 64
 指定设置 37, 64
 SOF 120, 123, 124
Software Build Settings 页 159
Software Build Settings 指南 159
 Software Builder 158
 makeprogfile 实用程序 159
 存储器初始化数据文件 163
 仿真器初始化文件 160
 闪存编程文件 161
 生成输出文件 159
 无源编程文件 162
 指定设置 37, 159
 SOPC Builder 150
 System Contents 页 152
 System Generation 页 153
 建立设计 151
 建立系统 152
 设计流程 150
 生成系统 153
 使用 151
 SRAM 对象文件 (.sof) 120, 123, 124, 162
Start EDA Netlist Writer 命令 59, 106
Start I/O Assignment Analysis 命令 39

Start Minimum Timing Analysis 命令 101
Start Software Build 命令 158
Start Timing Analyzer 命令 101
Start VQM Writer 命令 91
 Symbol Editor 28
 Synopsys 设计约束 (SDC) 文件 106
Synthesis 页 51

T

Tcl 13, 14, 16
 Text Editor 28
 Timing Analyzer 4, 98
Timing 指南 36, 99
Toolset Directories 页 159

U

USB-Blaster 下载电缆 121, 133

V

Verilog HDL 29, 45
Verilog HDL Input 页 45, 46
Verilog HDL Output Settings 对话框 58
 Verilog Quartus 映射文件 (.vqm) 27, 44, 48, 91, 115
 Verilog 测试台文件 (.vt) 60
 Verilog 设计文件 (.v) 27, 44, 48
 Verilog 输出文件 (.vo) 57
 VHDL 29, 45
VHDL Input 页 45, 46
VHDL Output Settings 对话框 58
 VHDL 测试台文件 (.vht) 60
 VHDL 设计文件 (.vhd) 27, 44, 48
 VHDL 输出文件 (.vho) 57
 VQM 文件 44, 48

W

Waveform Editor 60, 65
Waveform Export 实用程序 136

Z

帮助, 获取 172
 保存中间综合结果 91
 报告窗口 72, 103
 编程 120

- 编程硬件 121
- 设计流程 120

 编程文件

- 建立辅助 124
- 转换 124

 编译器指令 50
 标准延迟格式输出文件 (.sdo) 57
 表格文本文件 (.ttf) 121, 124, 162
 表文件 (.tbl) 65, 136
 布局布线

- 另请参阅布局布线
- 增量布局布线 81
- 设计流程 70

 布线 70

- 关键路径 111
- 连接计数 111
- 统计 111
- 延迟 111
- 拥塞 111

 参数化模块 (LPM) 函数库 30
 参数性函数 30

- 实例化 32, 46
- 使用 30
- 推断 33, 34
- 在其它 EDA 工具中实例化 33, 46

 测试台文件 60
 串行矢量格式文件 (.svf) 121, 124
 从二进制图像文件 (.sbi) 161
 存储器初始化数据文件 163, 164
 存储器初始化文件 (.mif) 60
 等式窗口 112
 增量布局布线 81
 时序逼近 109, 110

- 查看布线 111
- 查看分配 111
- 设计流程 110
- 使用 LogicLock 区域 116
- 使用网表优化 113
- 执行分配 112

- 时序逼近布局图 74, 75, 110
- 时序分析 98
 - 查看结果 102
 - 查看延迟路径 104
 - 进行 99, 101
 - 设计流程 98
 - 使用 EDA 工具进行 106
 - 指定设置 37
- 时序仿真
 - EDA 工具 61, 62
 - Quartus II Simulator 64
- 时序要求 99
 - 个别 101
 - 工程范围 100
 - 指定 99
- 逗号分隔值文件 (.csv) 136
- 独立 Programmer 120
- 二进制文件 (.bin) 163
- 反标 81, 92, 113
- 分割 40
- 分配
 - 查看 111
 - 导入 38
 - 基于路径 116
 - 进行 36
 - 位置 76
 - 验证 39
 - 执行 112
- 分配编辑器 76
- 更改管理的设计流程 142
- 工程更改单 *请参阅* ECO
- 功率估算 60, 66
- 功率输入文件 (.pwf) 60
- 功能仿真
 - EDA 工具 61
 - Quartus II Simulator 64
- 关键路径 111
- 综合
 - VHDL 和 Verilog HDL 支持 45
 - 设计流程 44
 - 使用 EDA 工具执行 48
 - 网表优化 50
 - 优化 51, 114
- black-box 方法 33
- 汇编文件 (.s, .asm) 158
- 基于块的设计 39, 85, 87
- 基于路径的分配 116
- 教程 173
- 可编程逻辑部分 SRAM 对象文件 (.psof) 162
- 可编程芯片系统 (SOPC) 150
- 可执行和可链接格式文件 (.elf) 162
- 可执行文件 11
- 库文件 (.a) 158, 163
- 库映射文件 (.lbf) 45
- 块符号文件 (.bsf) 27, 28
- 块设计文件 (.bdf) 26, 27
- 链式描述文件 (.cdf) 122, 123
- 逻辑选项 50, 78
- 命令行可执行文件 11
- 仿真 55
 - 库 62
 - 仿真流程 56
- 仿真器初始化文件 160
- 布局布线
 - 增量布局布线 81
 - 分析 71
 - 设计流程 70
 - 优化 76, 115
- 配置 120
- 批处理文件 14
- 布局图编辑器 74, 75
- 器件, 编程和配置 120
- 全编译 4
- 软件开发 157
 - 另请参阅* Software Builder
- 闪存编程文件 161
- 上次编译布局图 74, 75
- 设计分割 40
- 设计空间管理器 78
- 设计输入 24
- 设计约束条件 36
- 设置
 - Analysis & Synthesis 51
 - Compiler 37
 - Design Assistant 52
 - EDA 工具 48
 - Fitter 优化 115
 - HardCopy 37
 - Simulator 37, 64

- Software Builder 37
- Timing Analyzer 37
- Verilog HDL 输入 45
- VHDL 输入 45
- 综合优化 51, 114
- 生成文件支持 19
- 十六进制 (Intel 格式) 输出文件
(.hexout) 121, 124, 126, 162
- 十六进制 (Intel 格式) 文件 (.hex) 127, 161
- 实体设置文件 (.esf) 88
- 矢量表输出文件 (.tbl) 65
- 矢量波形文件 (.vwf) 65, 136
- 矢量文件 (.vec), 65
- 属性 50
- clear-box 方法 34
- 图形设计文件 (.gdf) 26, 27
- 图形用户界面 3
- 外壳, Tcl 脚本 13
- 网表优化
 - 综合 50, 51, 114
 - 布局布线 115
 - 实现时序逼近 113
 - 物理综合 115
- 位置分配 76
- 文本设计文件 (.tdf) 27
- 文件编制约定 v
- 无源编程文件 162, 163
- 物理时序估算 111
- 物理综合, 优化 115
- 系统调试 131
 - 另请参阅 SignalProbe 功能
 - 另请参阅 SignalTap II 逻辑分析器
- 系统构建描述符文件 (.sbd) 161
- 系统内配置文件 (.isc) 121, 124
- 工程, 建立 25
- 芯片编辑器 75, 140, 143
- 原二进制文件 (.rbf) 121, 124, 162
- 在 63
- 知识产权 (IP) 函数 30
- 值更改转储文件 (.vcd) 136
- 转换 Max+PLUS II Project 命令 25**
- 资源属性编辑器 145
- 总线功能模型 66
- 最少的时序分析 98
- 最少时序分析 101