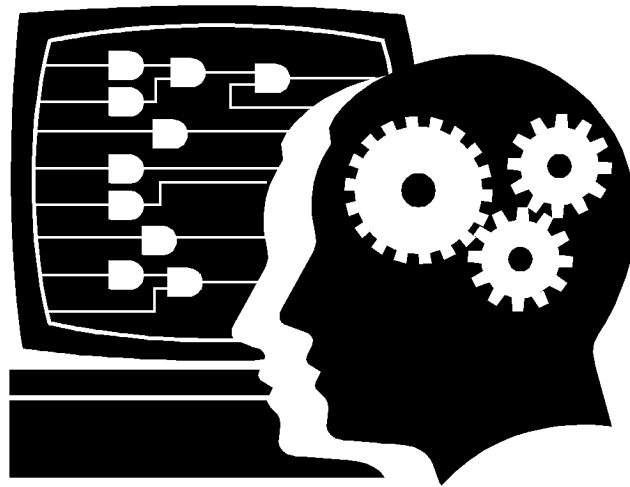
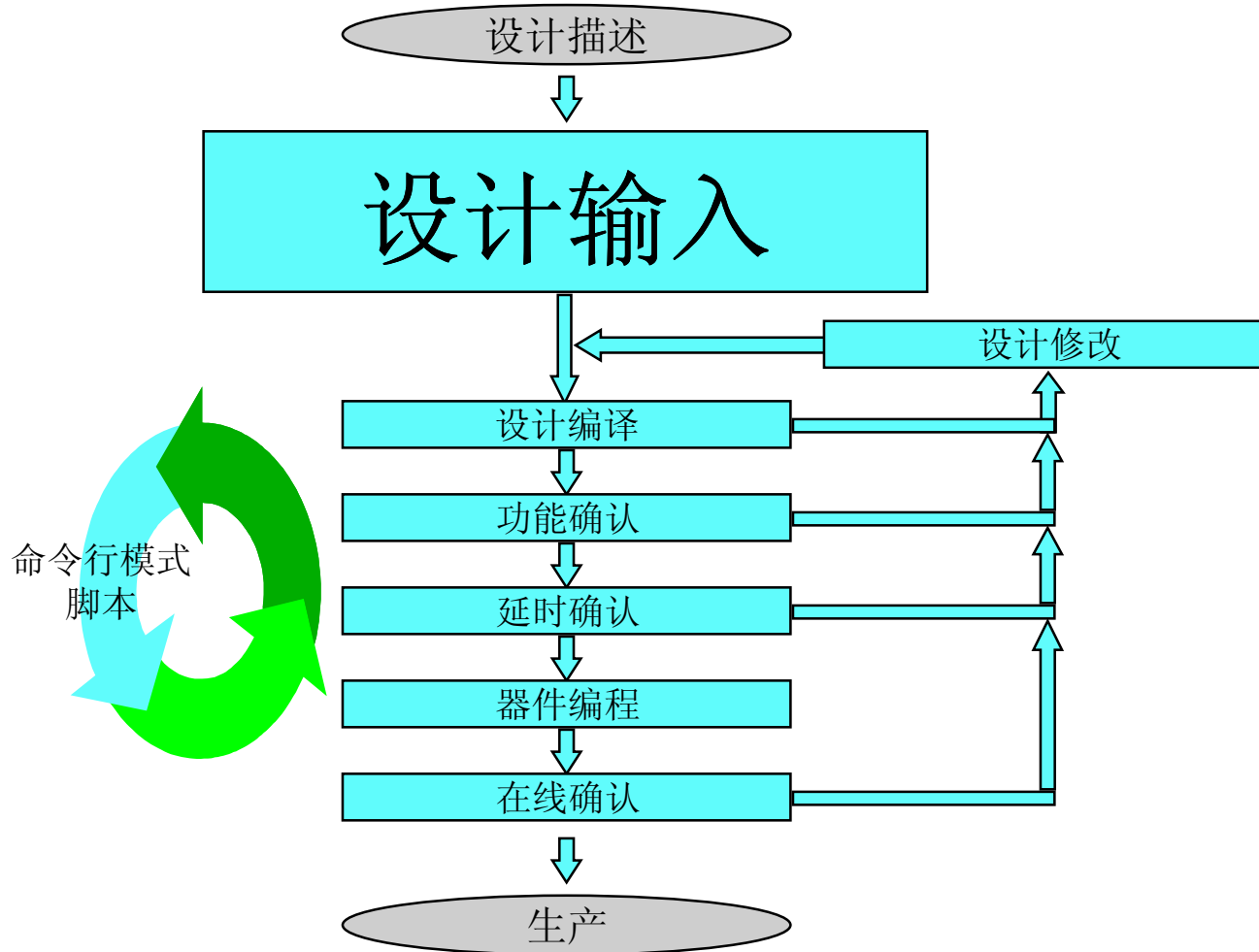


Quartus II 用户指南





设计输入

■ 多种设计输入方法

— Quartus II

- 原理图式图形设计输入
- 文本编辑
 - AHDL, VHDL, Verilog
- 内存编辑
 - Hex, Mif

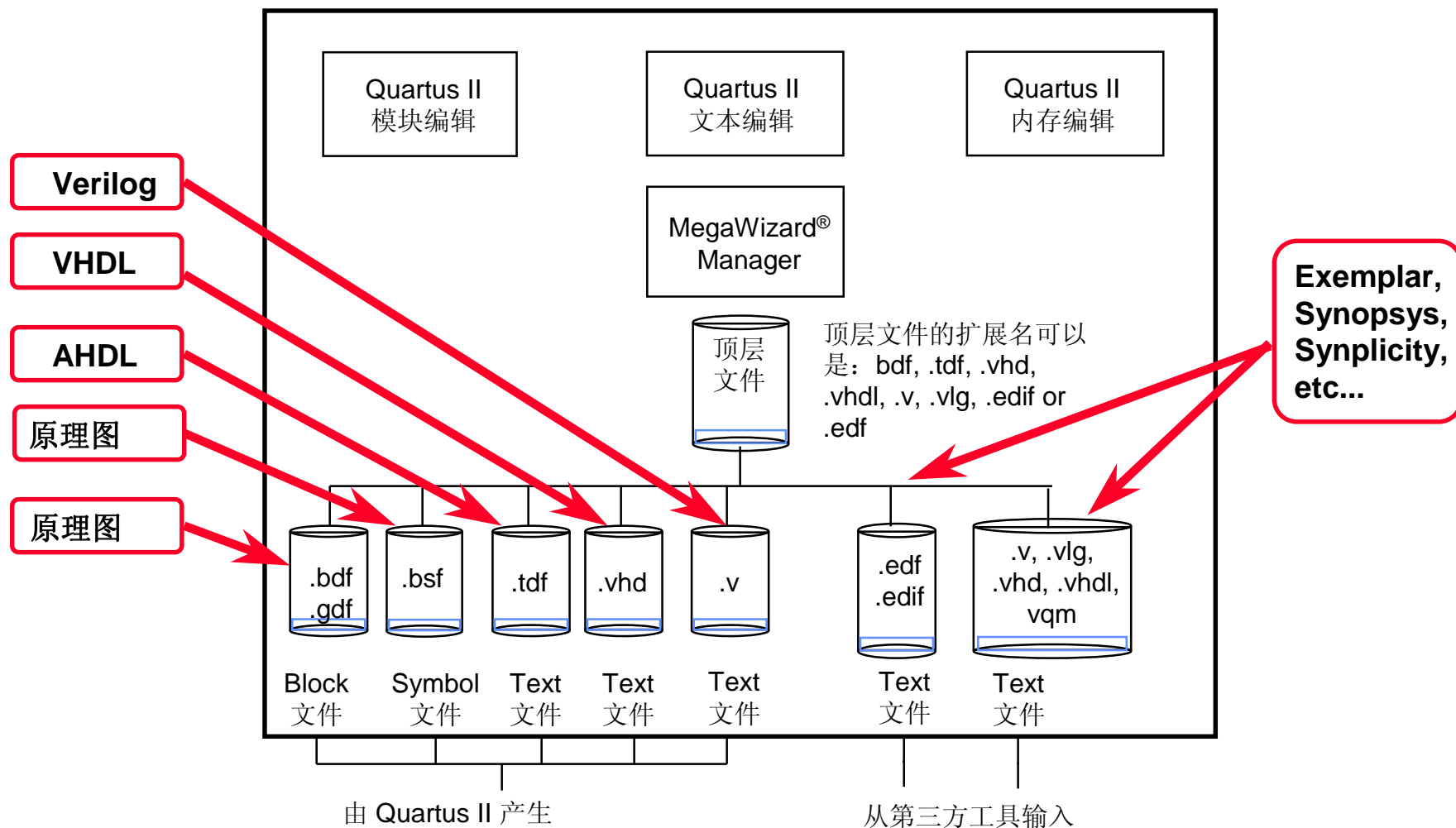
— 第三方工具

- EDIF
- HDL
- VQM

— 或采用一些别的方法去优化和提高输入的灵活性:

- 混合设计格式
- 利用LPM和宏功能模块来加速设计输入

设计输入文件



图表和原理图编辑器

该编辑器既可以编辑图表模块，又可以编辑原理图

- 图表模块编辑是主要的顶层设计的主要方法
- 原理图编辑是传统的设计输入方法
- 用户可以利用加入Quartus II 提供的LPMs，宏功能等函数以及用户自己的库函数来设计
- 提供“智能”的模块链接和映射

图表模块 – 设计流程

- 产生一个新的模块设计文件
 - 画出图表模块或输入设计单元符号
 - 输入接口和参数信息
 - 连接各个设计单元(利用单连线，总线等)
- 保存设计
 - 文件的后缀名为 .bdf
- 顶层模块可以是用户产生的HDL文件或图形编辑文件
- 从顶层设计可以产生设计单元，头文件，或转化成Verilog/VHDL文件

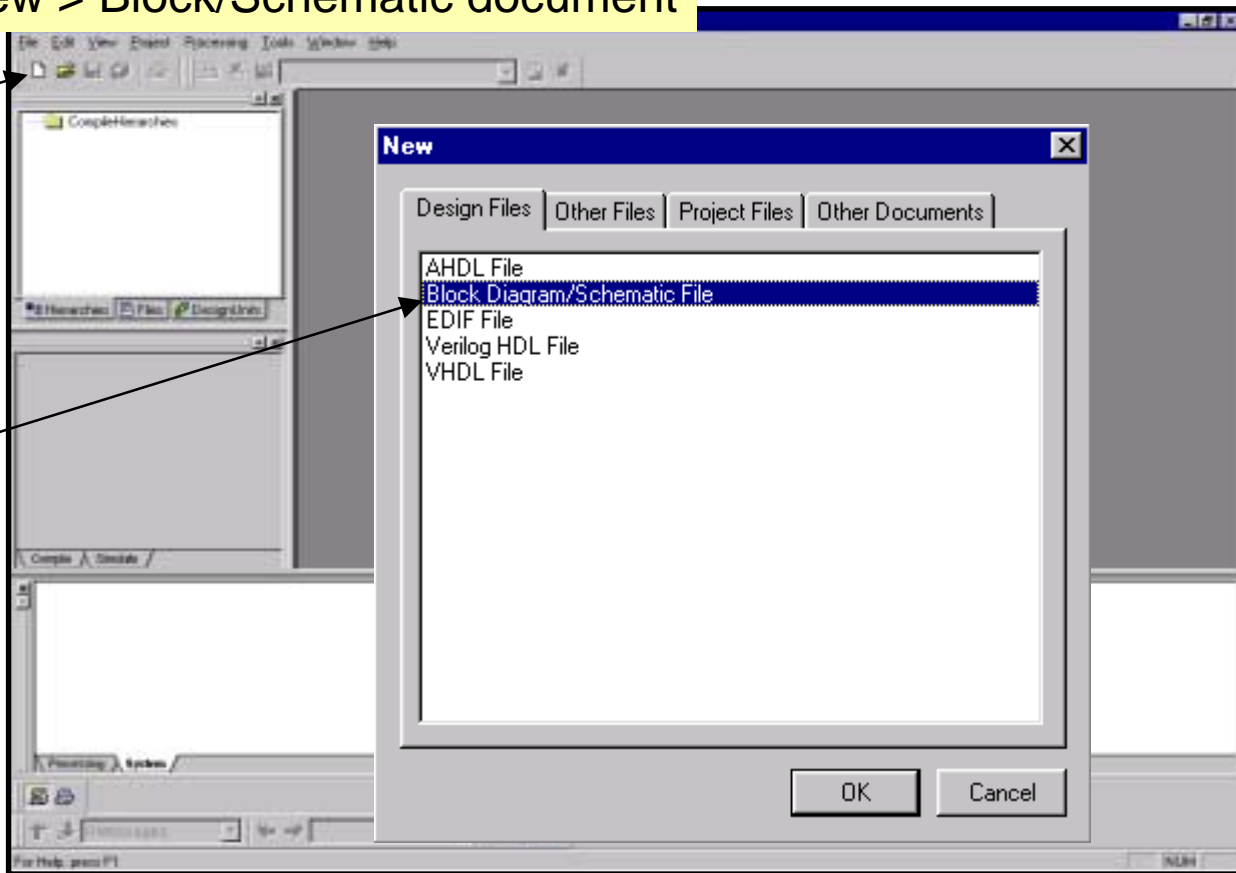
模块编辑器 – 产生一个新的文件

■ 产生一个新的图表模块/原理图文件

菜单: File > New > Block/Schematic document

新建一个新的文件

选择模块/原理图文档

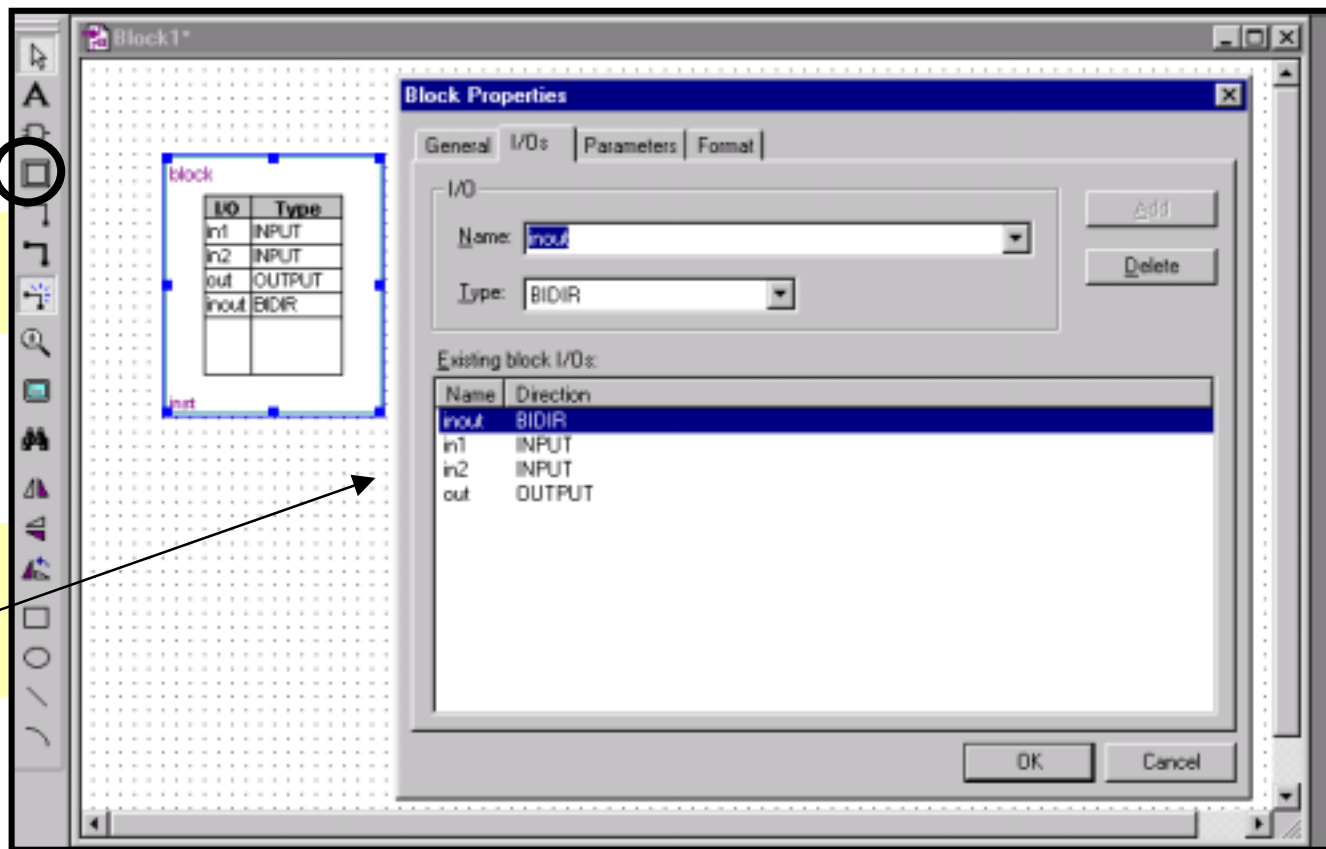


模块编辑器- 设计模块

■ 从工具栏中产生模块和输入端口

点击工具栏中的“模块”来画一个图表

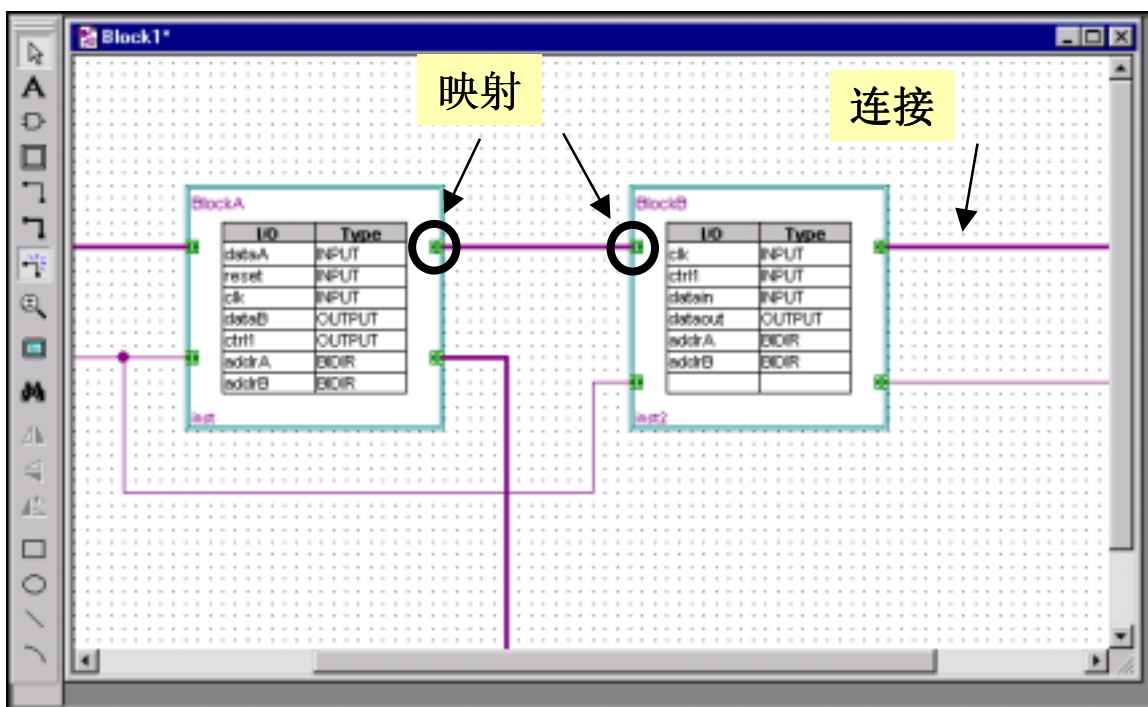
右键点击模块。选择属性，从弹出的对话框中输入端口信息。



模块编辑器-“智能”连接

■ Quartus II 有“智能”模块连接和映射

- 如果连接不同模块时，两边端口的名字相同的话就不用标注出来
- 一个管道可以连接模块之间所有的普通 I/O

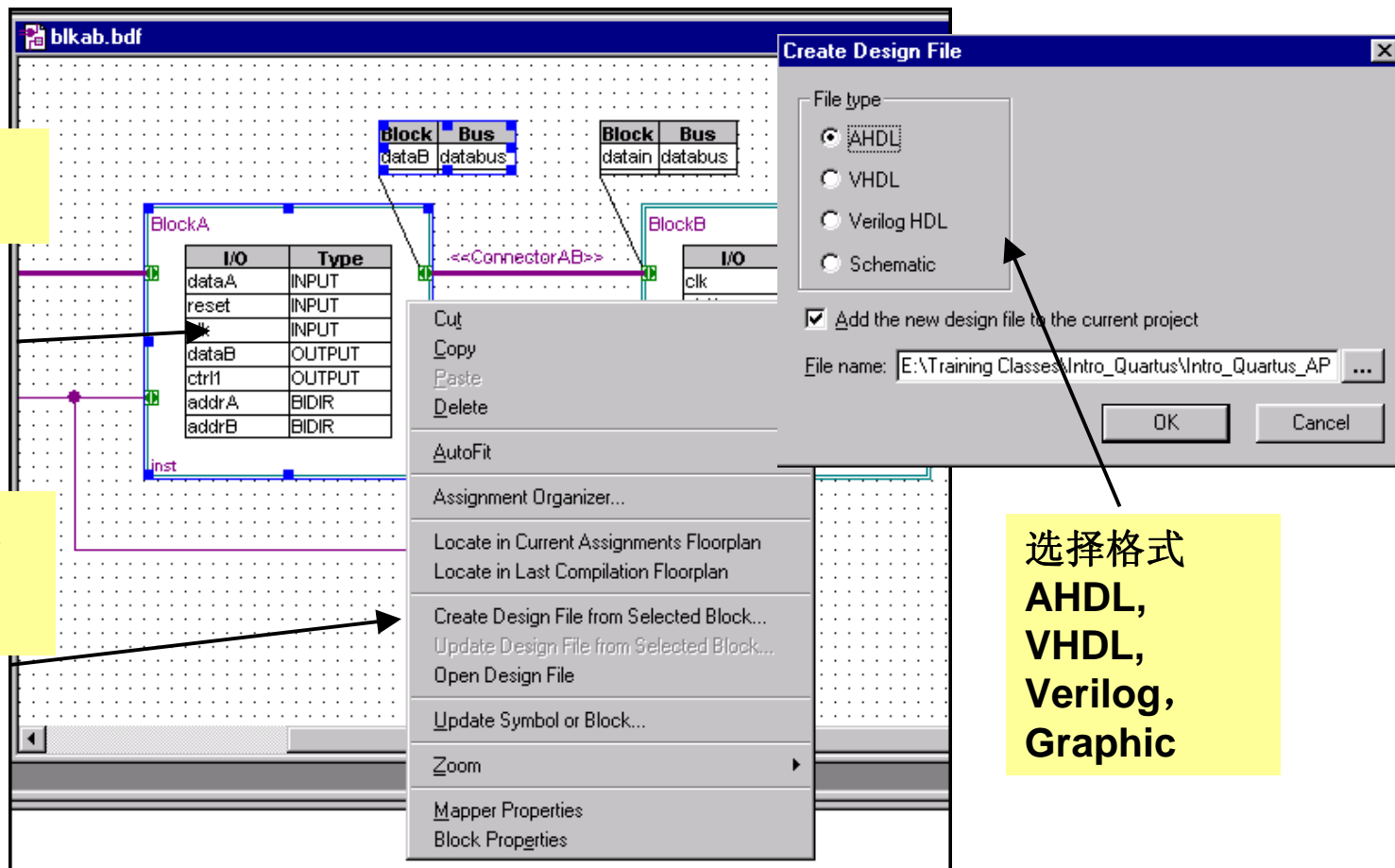


模块编辑器- 产生设计文件

■ 给模块定义并产生HDL或图形设计文件

右键点击，
弹出对话框

从菜单中
选择产生设计
文件



选择格式
AHDL,
VHDL,
Verilog,
Graphic

支持的第三方工具

综合工具

Leonardo Spectrum*
Design Compiler
FPGA Compiler II
Standard* &
Altera Edition*
FPGA Express*
Synplicity Synplify*

仿真工具

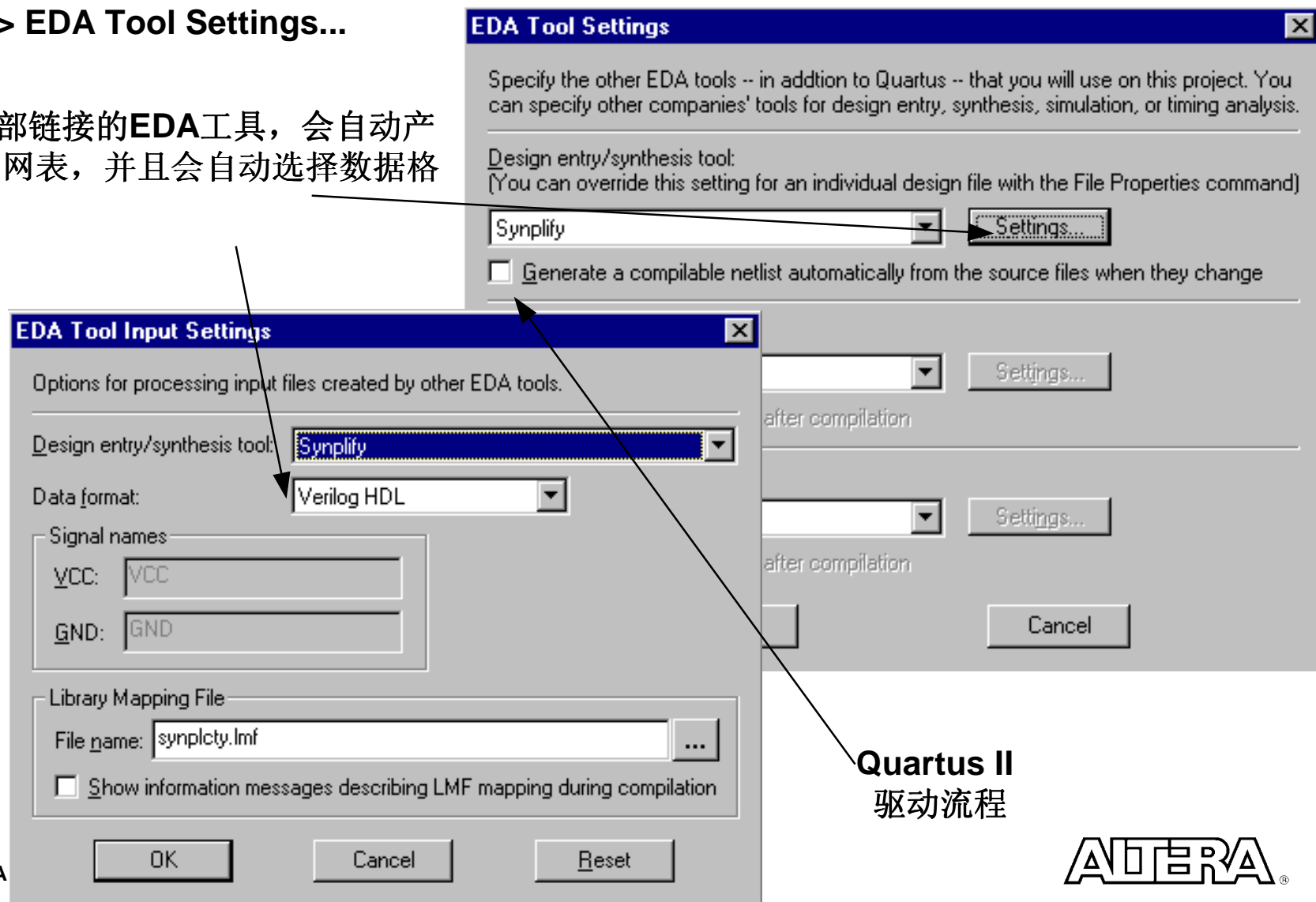
- **ModelSim***
- **ModelSim OEM***
- **Cadence Verilog-XL**
- **Innoveda BLAST**
- **PrimeTime***
- **Synopsys VCS**
- **Synopsys VSS**

注意 * 表示支持内部链接

Quartus II 驱动流程

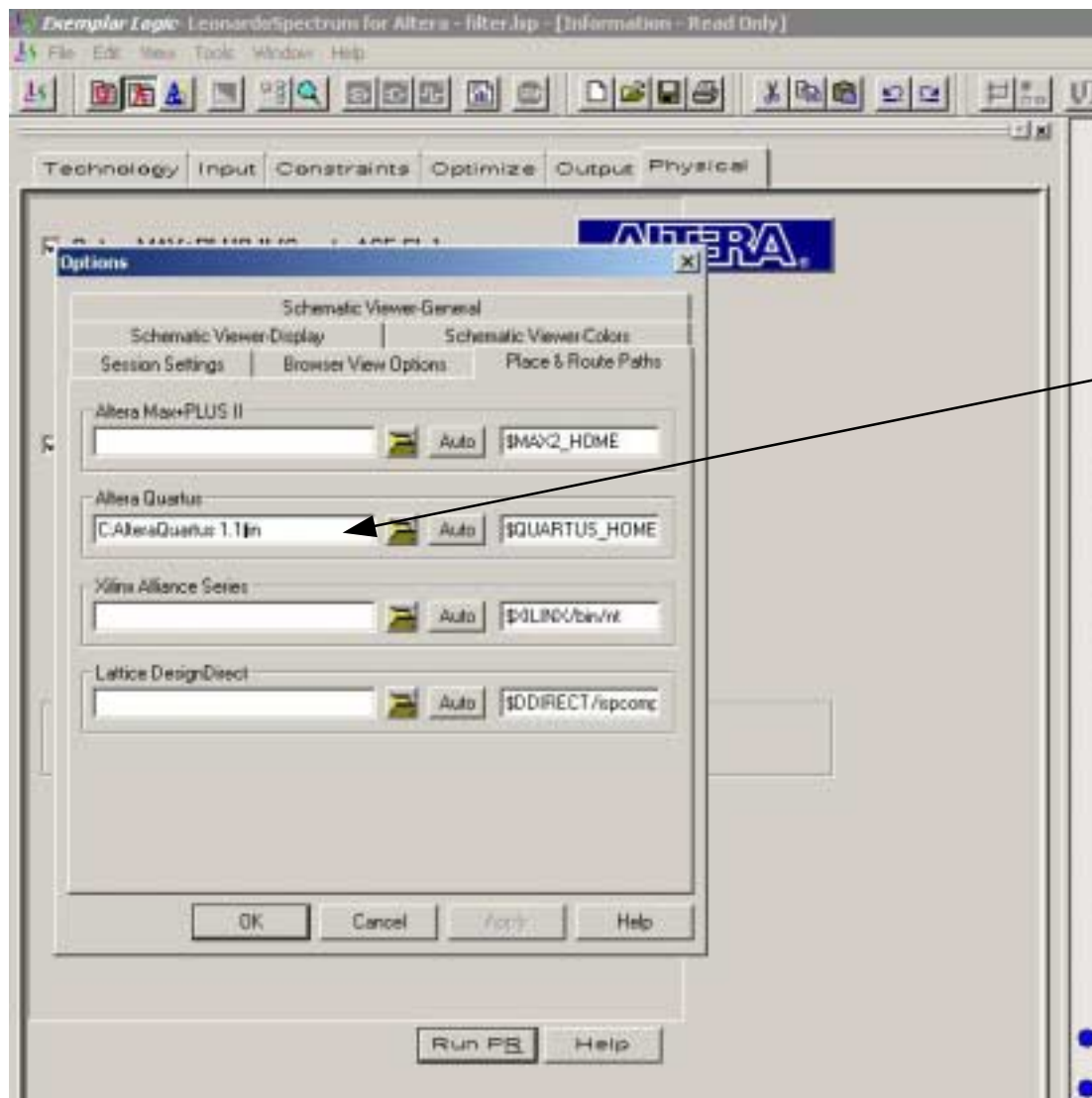
Project > EDA Tool Settings...

当选择内部链接的**EDA**工具，会自动产生**ATOM** 网表，并且会自动选择数据格式



Quartus II
驱动流程

EDA 驱动流程



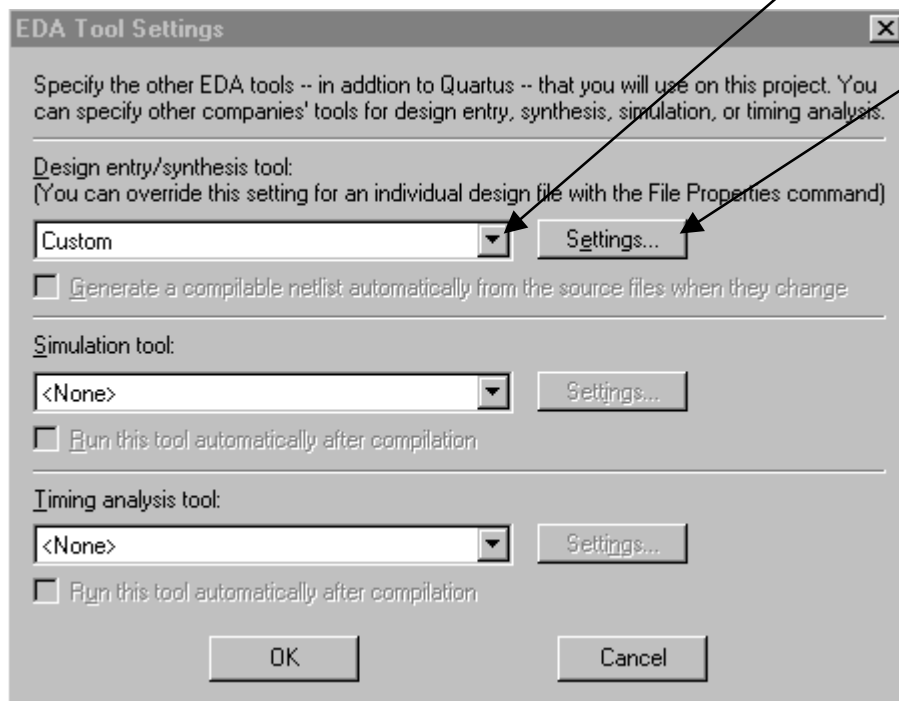
EDA 工具驱动
Quartus II

ALTERA®

文件驱动流程: 非内部链接

如果用一个非内部链接的 EDA 工具产生 VHDL, Verilog, EDIF 文件, 就需要阐明 .lmf 文件来给定文件格式

菜单 **Project > EDA Tool Settings...**

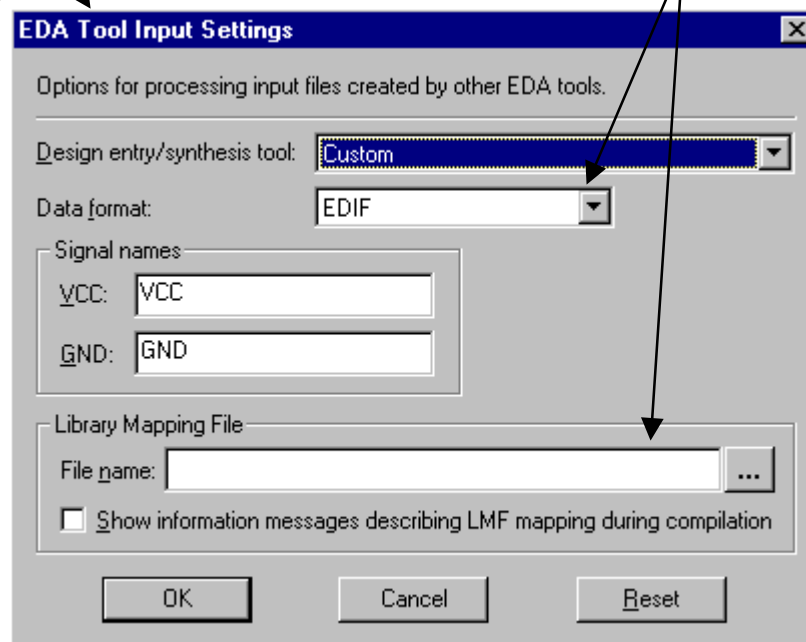


选择自定义

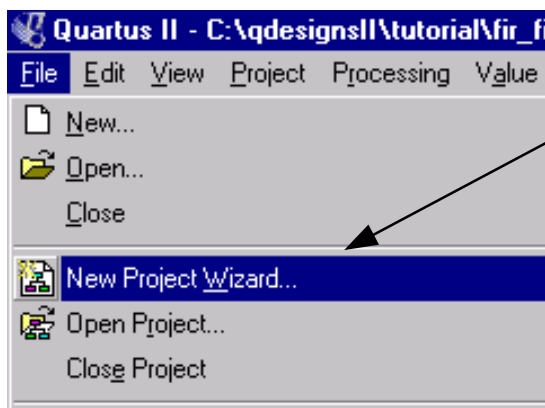
选择设置...

可以是 EDIF,
VHDL, or Verilog

选择库函数映射文件 (lmf)



新建项目指南

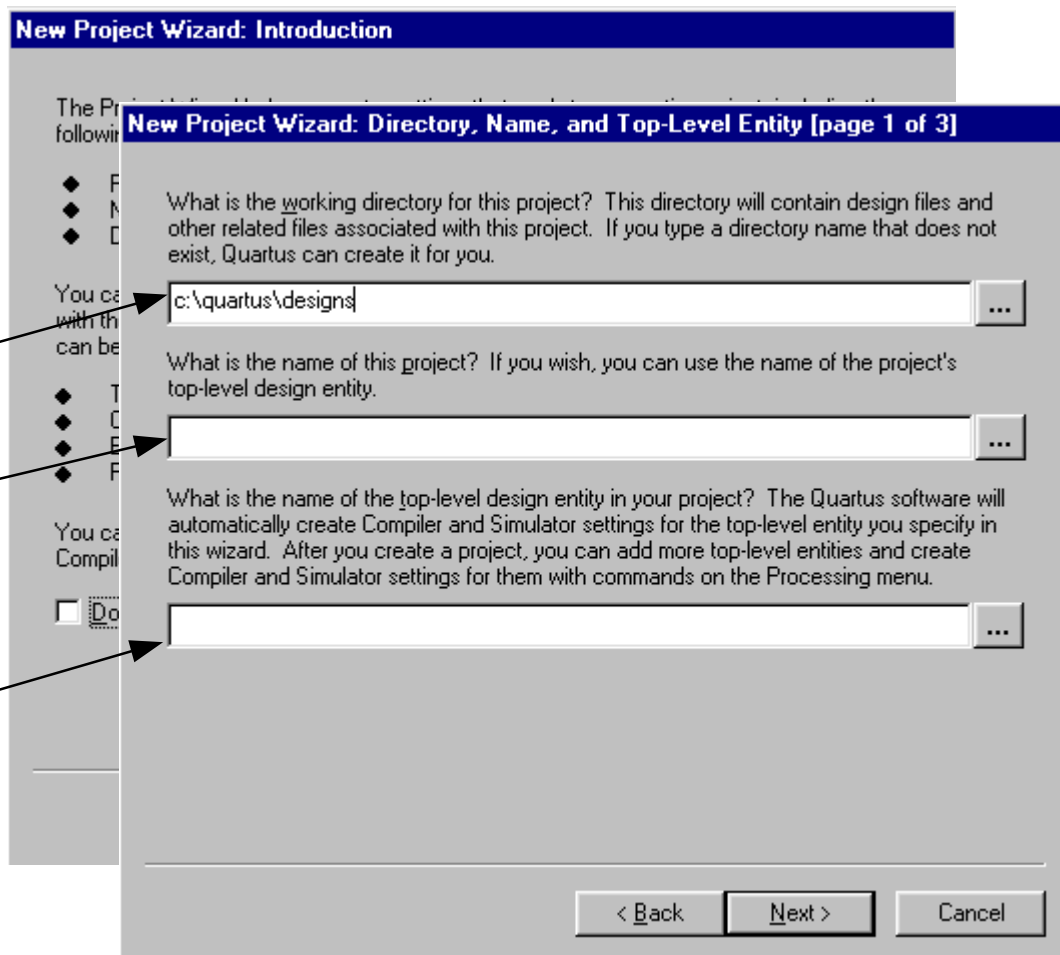


1. 打开新建项目指南

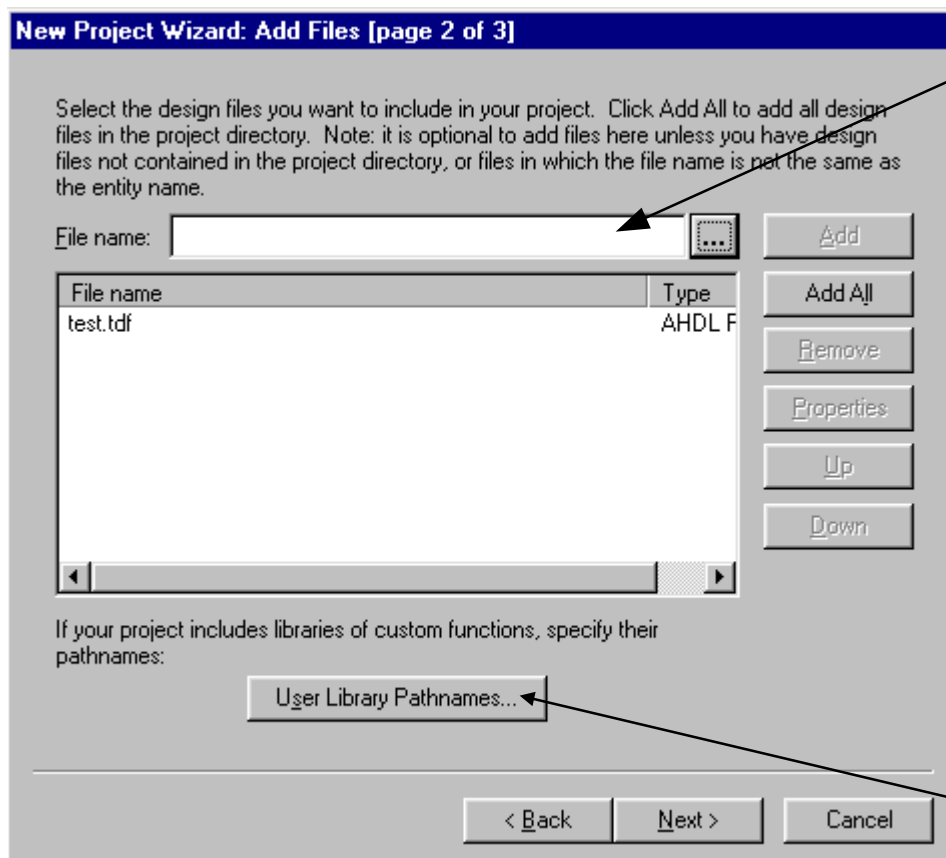
2. 选择工作目录

3. 项目名称，可以是任何名字。
推荐使用和顶层设计名相同的名字。

4. 顶层设计
.名字必须和顶层文件名字相同



新建项目指南



5. 加入设计文件

- Graphic (.BDF, .GDF)
- AHDL
- VHDL
- Verilog
- EDIF

注意:

- 并不是该项目的目录下的所有文件都要被加入
- 假如顶层设计和顶层文件的名字不一样的话，一定要加入顶层文件的名字

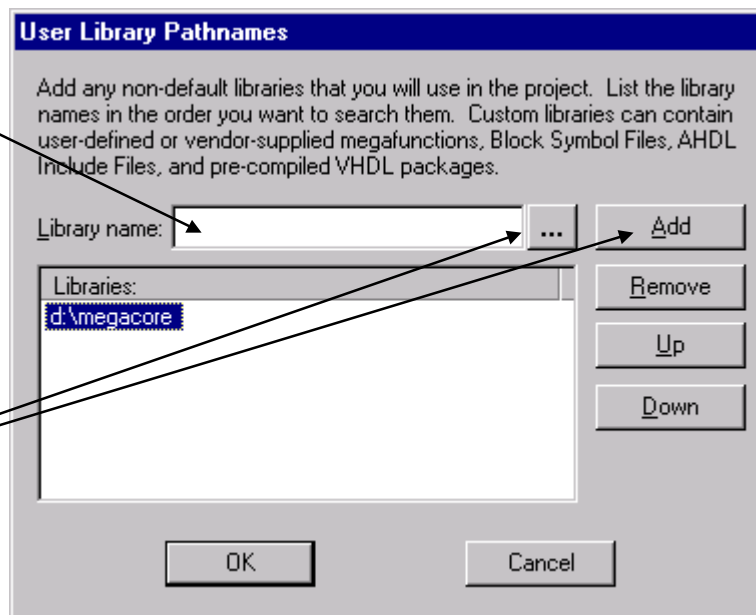
6. 加入用户自己定义的库函数的路径和文件名

新建项目指南

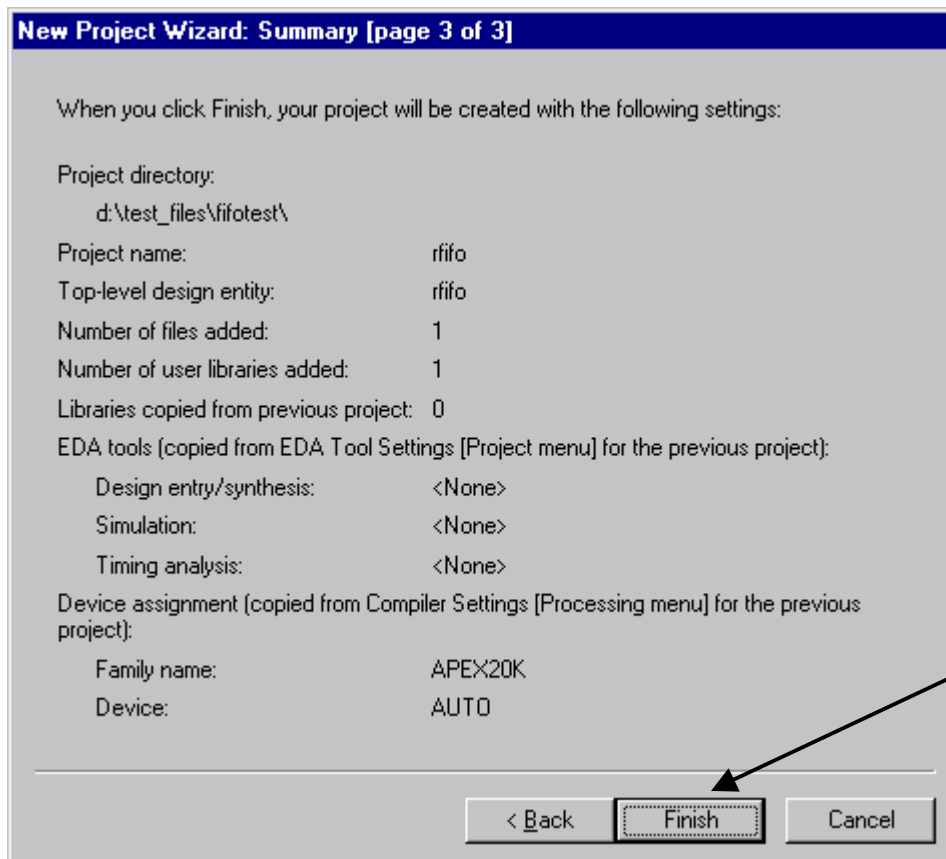
6(继续.) 加入用户定义的库函数的目录和文件名

- 用户定义的库函数 (例如. **MegaWizard** 功能)
- **MegaCore®/AMPPSM** 库
- 预编译 VHDL 文件

浏览并找到文件,
然后再加入



新建项目指南



列出项目名和目录名字

7. 检查结果然后按结束

项目菜单

■ 编辑项目的设置t

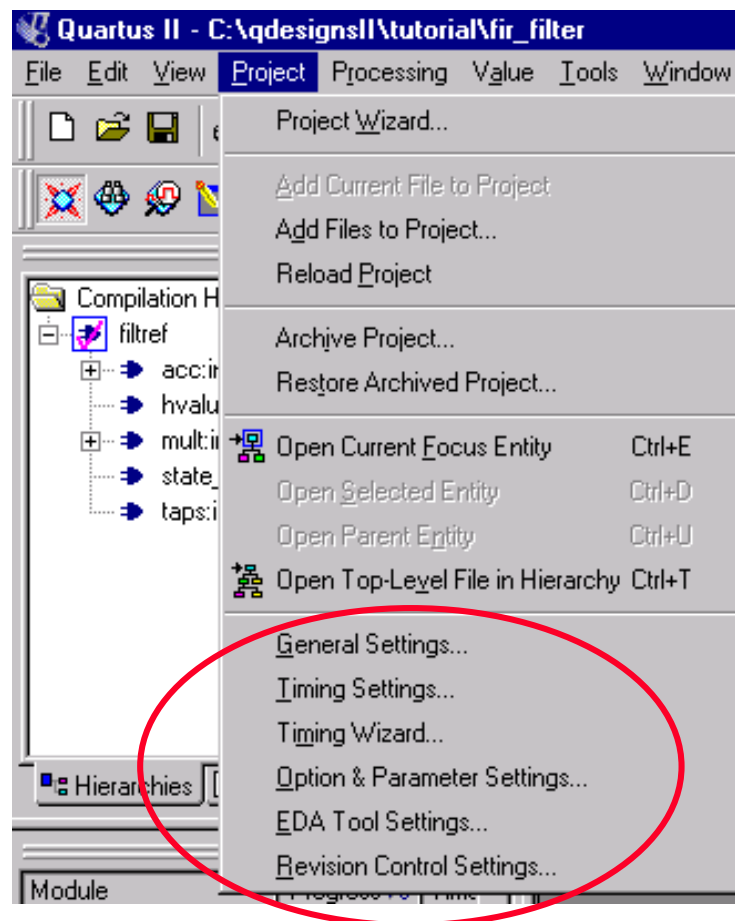
- 增加/去除文件或库函数

■ 项目设置

- HDL 类型界面
- 第三方EDA工具设计流程
- 定时设置
- 版本控制

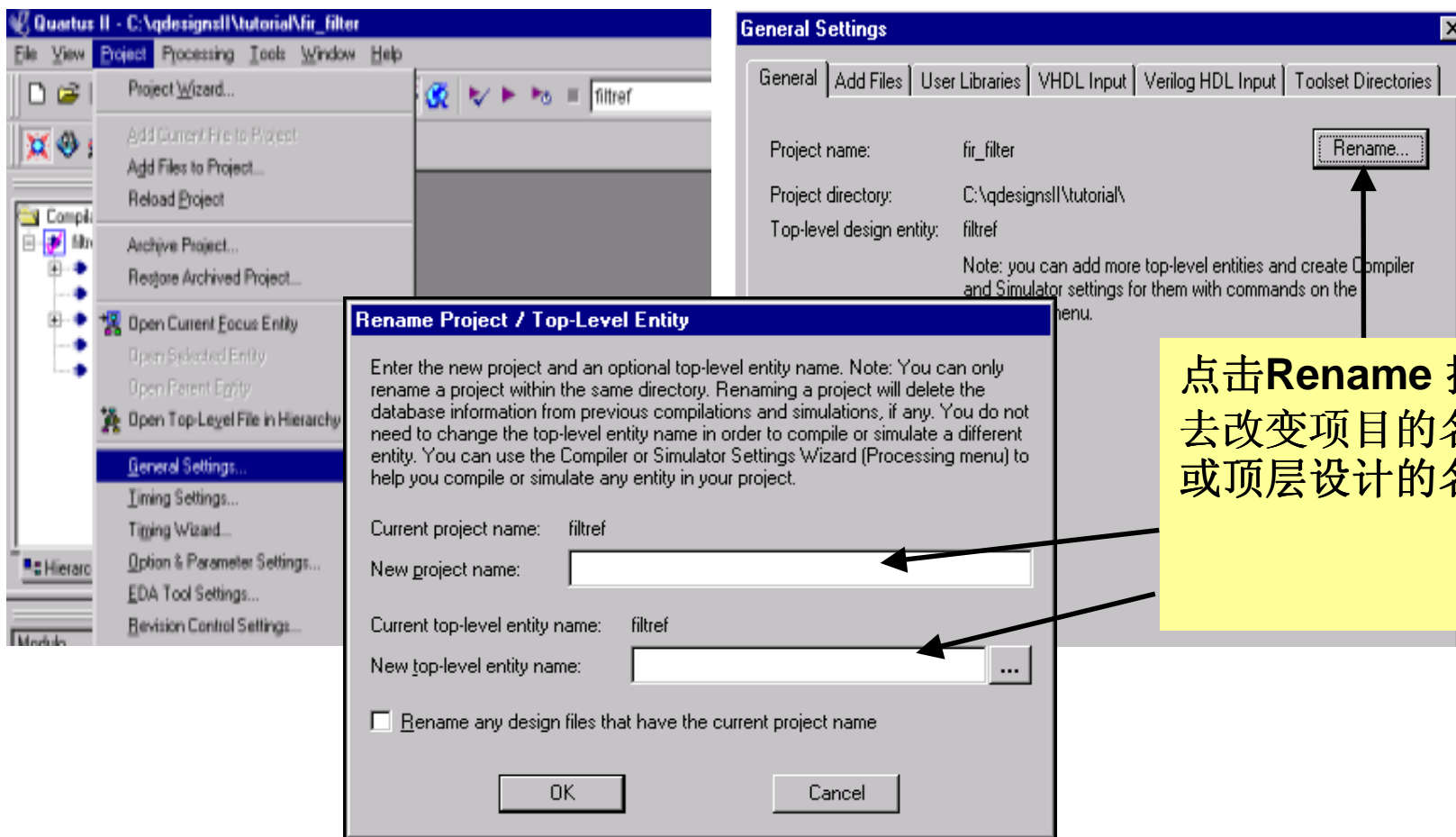
注意:

所有的项目设置中, 除了项目名和顶层设计名不一样之外, 其它的设置都保留上一个项目的设置

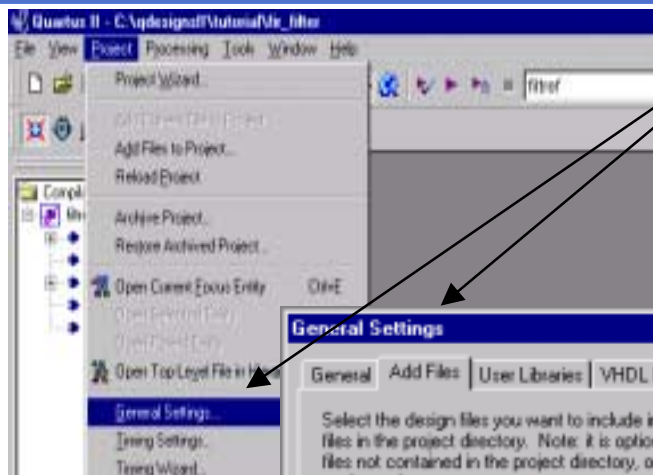


编辑项目设置

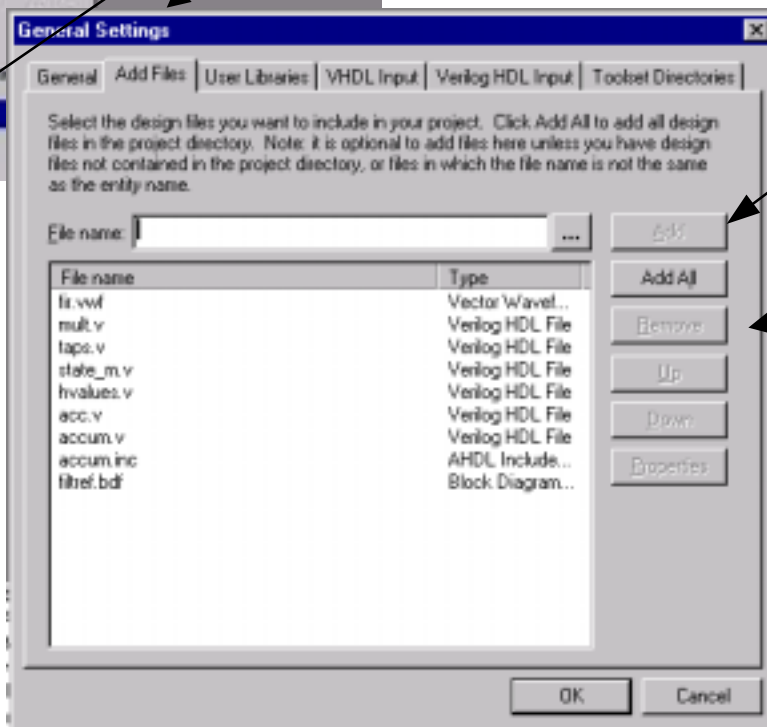
■ 改变项目的名字或顶层设计的名字



编辑项目设置



打开项目基本设置对话框



增加/去除项目文件

增加

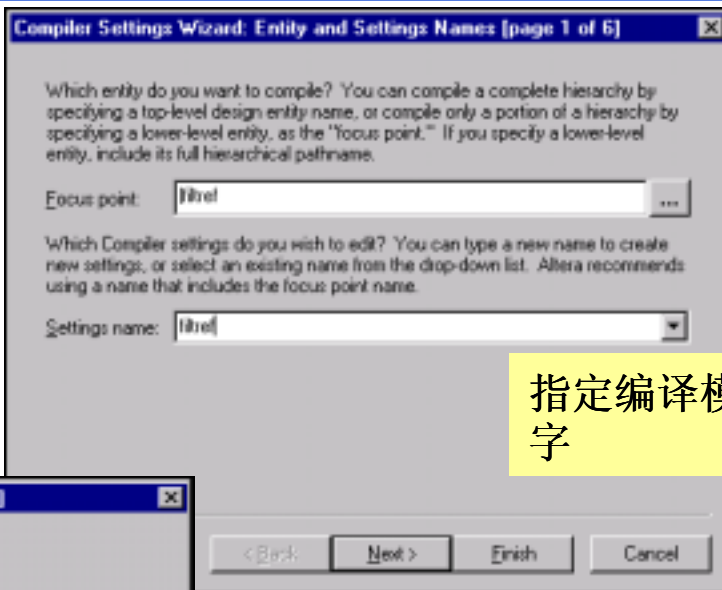
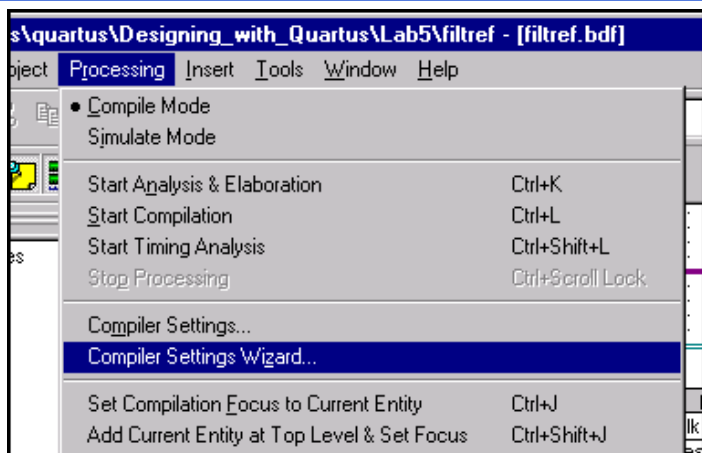
- 找到文件
- 增加

去除

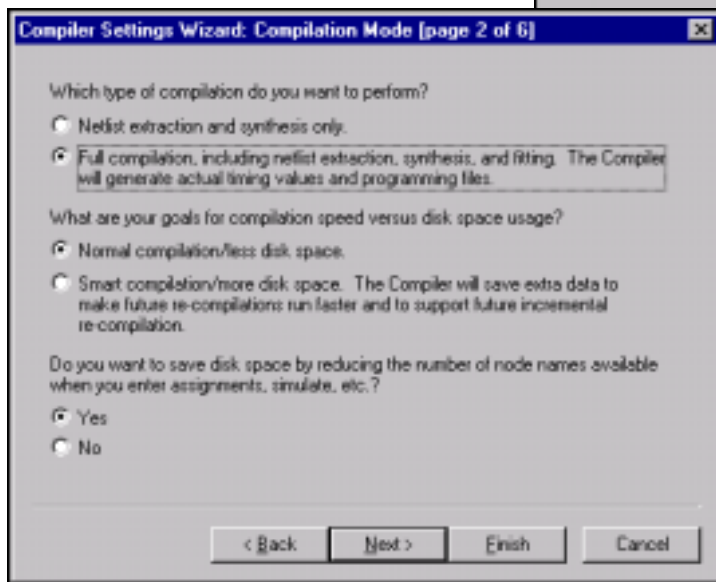
- 从列表中选择
- 去除

Quartus II 编译

编译设置指南



指定编译模块及设定名字

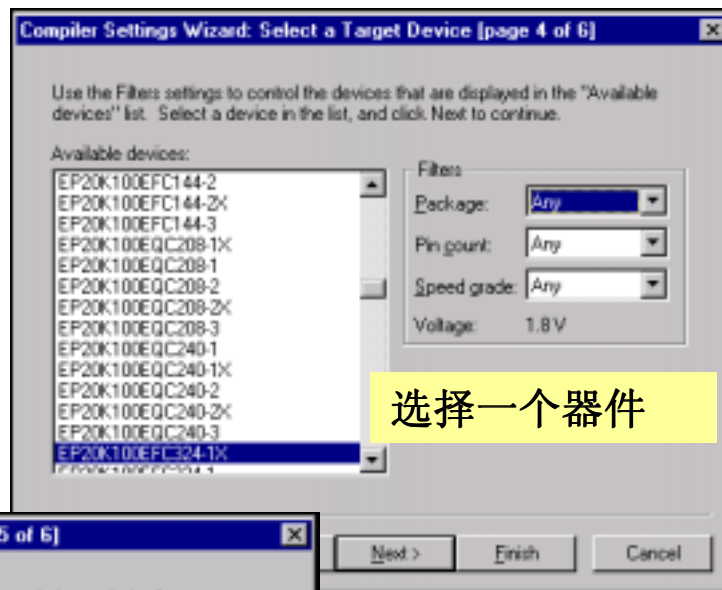


- 网表输出或完全编译？
- 优化速度或资源利用率？
- 削减接点名字？

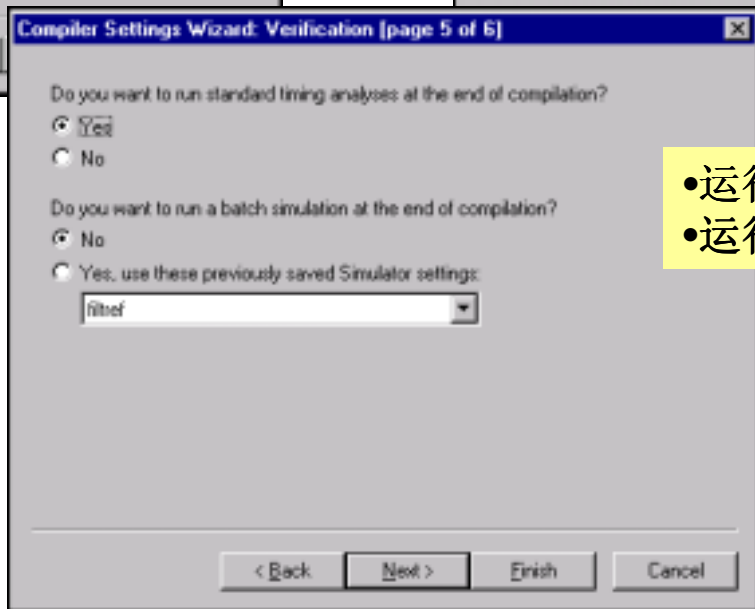
编译设置指南



自动选择或给定器件?



选择一个器件



- 运行延时分析?
- 运行仿真?

编译定义

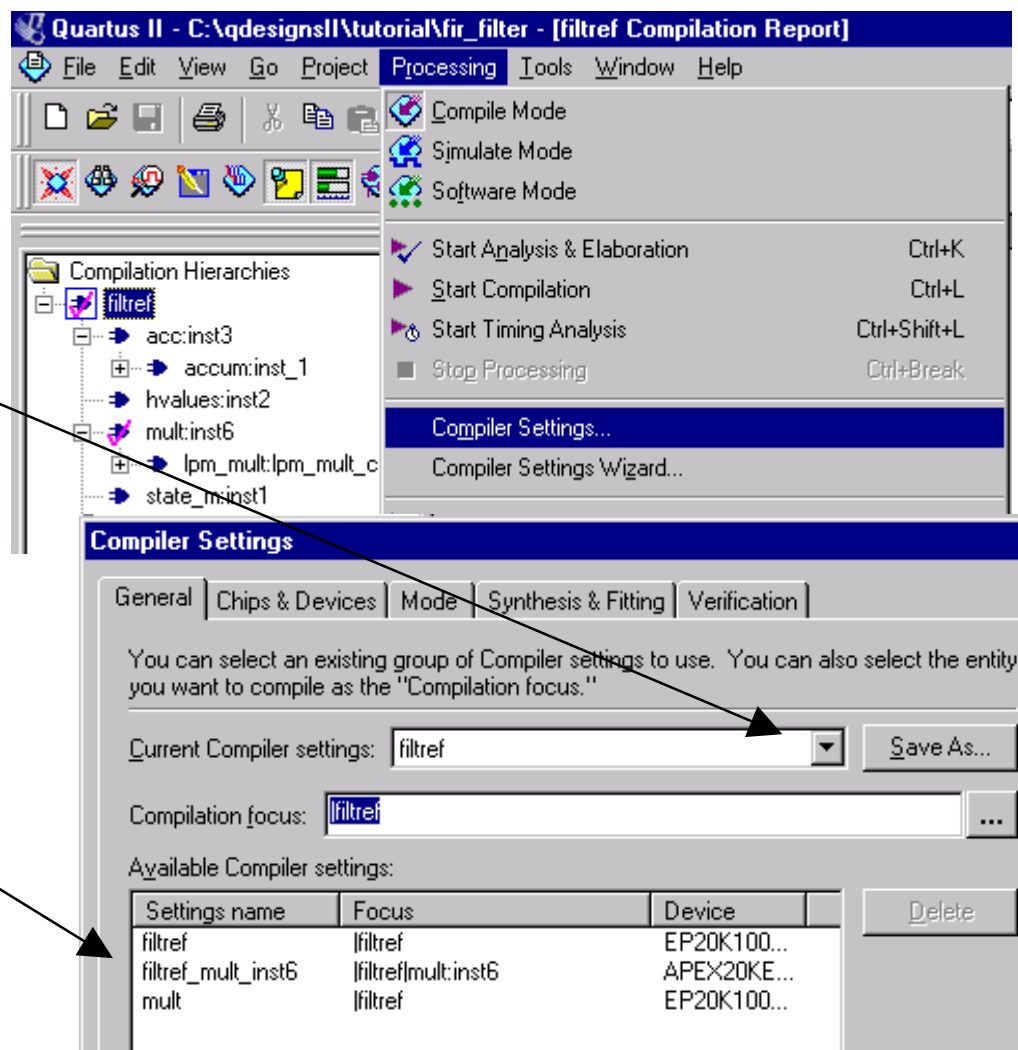
■ 编译类型

- 只是综合并输出网表
 - 编译设计文件，综合产生门级代码
 - 编译器只运行到综合这步就停止了
 - 编译器只产生估算的延时数值
- 完全的编译，包括编译，网表输出，综合，配置器件
 - 编译器除了完成以上的步骤，还要将设计配置到**ALTERA**的器件中去
 - 编译器根据器件特性产生真正的延时时间和给器件的配置文件

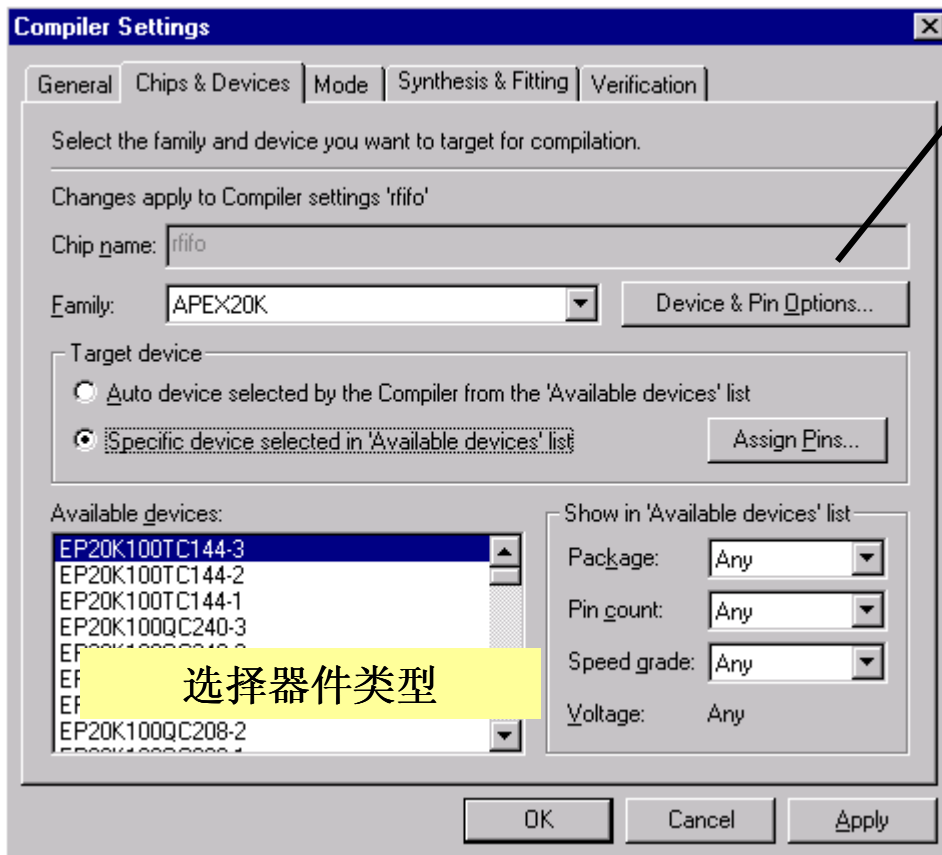
编辑编译选项

从下拉菜单中选择编译设置

显示所有给编译模块的编译设置



编译设置 - 芯片和器件



说明器件和引脚选项

- 概要
- 配置
- 下载文件
- 不用的引脚
- 双种用途的引脚
- 电压值

编译设置 - 芯片和器件

1. 选择引脚号

2. 调用引脚查找器去查找引脚名或类型

3. 增加到指定列表

Pin Assignments

Select a device pin and the type of assignment you wish to make. You can also make pin assignments in the Assignment Organizer and the Floorplan Editor. You can reserve unused pins on a device-wide basis with the Unused Pins tab in the Device & Pin Options dialog box.

Changes apply to Compiler settings: **T1rel'**

Available Pins & Existing Assignments:

Nu...	Name	I/O ...	I/O St...	Type
2		3	LVTTL	Row I/O
3		3	LVTTL	Row I/O
6		3	LVTTL	Row I/O
7		3	LVTTL	Row I/O
8		3	LVTTL	Row I/O
9		3	LVTTL	Row I/O
10		3	LVTTL	Row I/O

☐ Show 'no connect' pins

Assignment

Pin name:

I/O standard: **LVTTL**

☐ Reserve pin (even if it does not exist in the design file):
As input tri-stated

Node Finder

Named: Filter: Design Entry (all names)

Look in: Lock/unlock subtrees

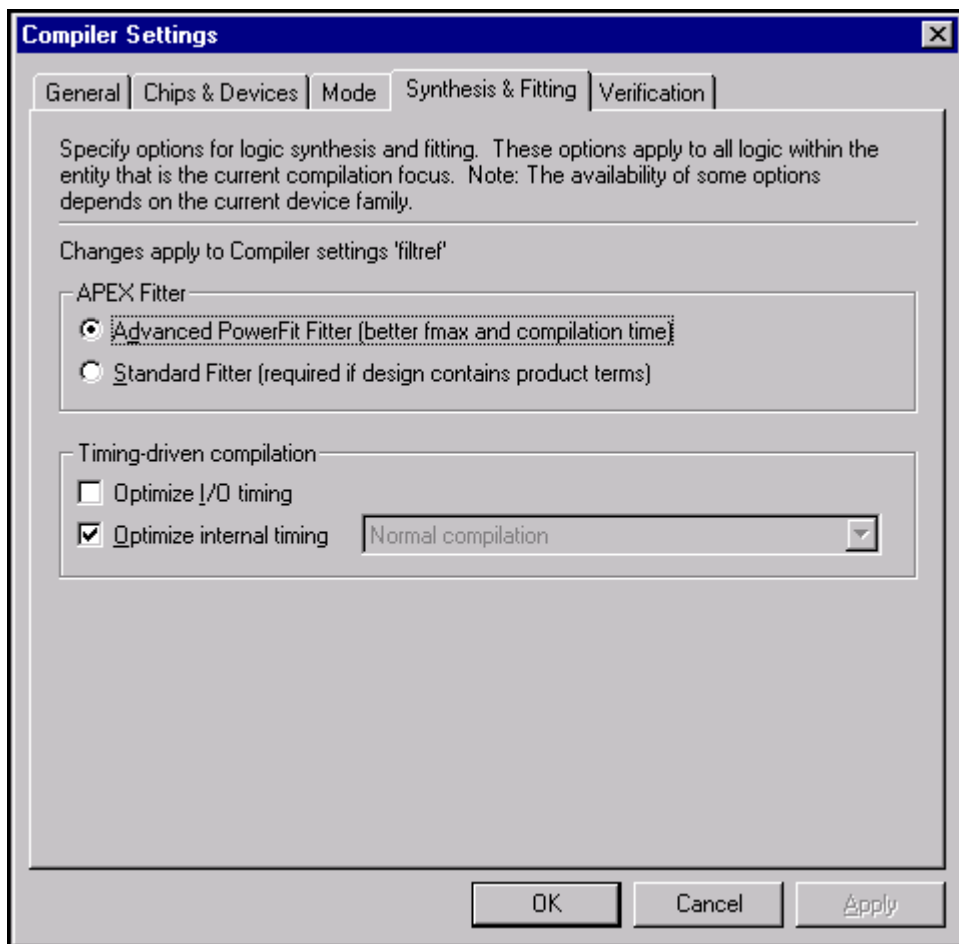
Nodes Found:

Name	Assignments
@ack	Unassigned
@almost_empty	Unassigned
@almost_full	Unassigned
@async_readnfile	Unassigned
@datain	Unassigned
@datain[0]	Unassigned
@datain[1]	Unassigned
@datain[2]	Unassigned
@datain[3]	Unassigned
@datain[4]	Unassigned
@datain[5]	Unassigned
@datain[6]	Unassigned
@datain[7]	Unassigned
@dataout	Unassigned
@dataout[0]	Unassigned
@dataout[1]	Unassigned
@dataout[2]	Unassigned
@dataout[3]	Unassigned
@dataout[4]	Unassigned

Selected Nodes:

Name	Assignments
@ack	Unassigned

编译设置 - 综合与配置



APEX™ 配置器:

- **PowerFit**配置器
 - 更好的**Fmax**和编译时间
 - 对打包 (**Cliques**) 有限制
- 标准配置器
 - **PTERM** 模式

定时驱动式编译:

- 优化 **I/O** 引脚的延时
- 优化 内部的延时

Quartus II 编译界面(开始)

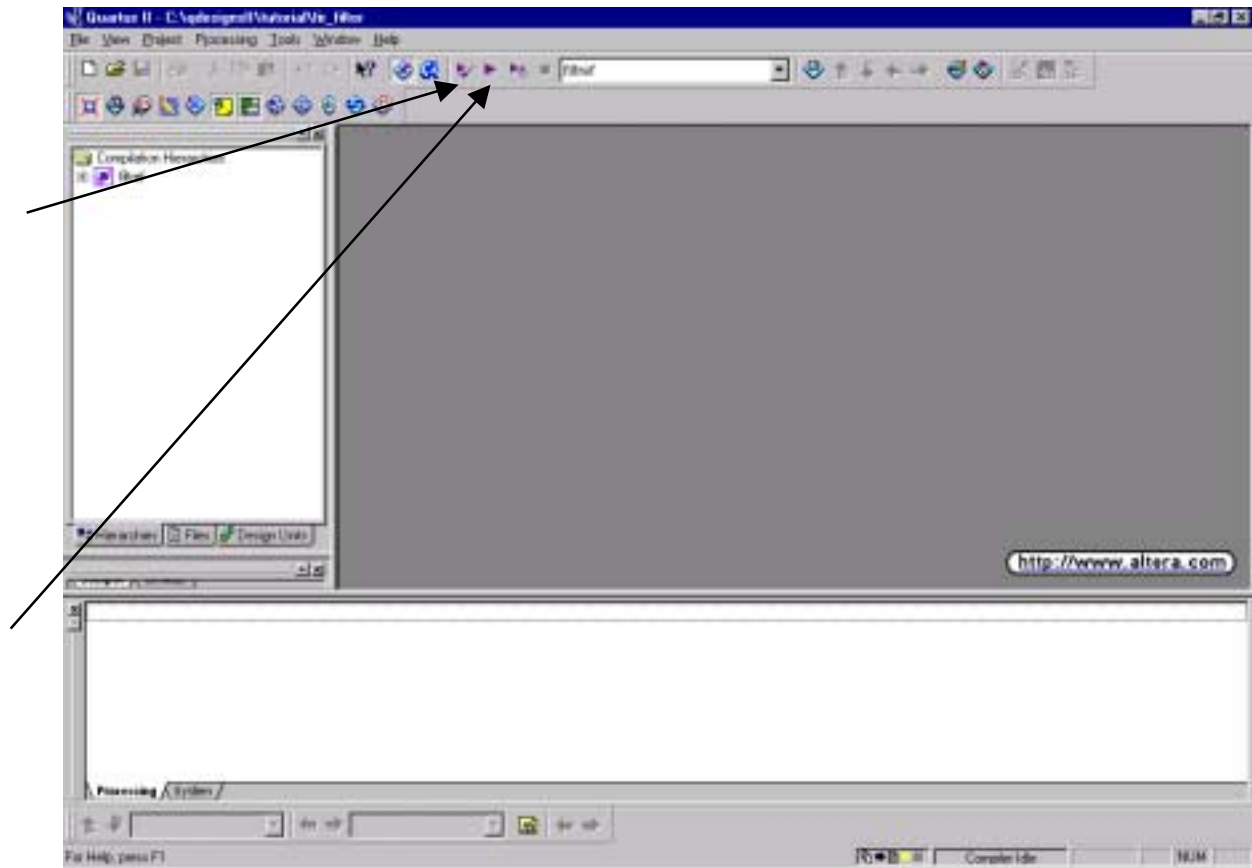
1

检查并解析 (保存 & 校验)



- 执行语法检查

执行完全的编译
(基于所有的编译设置)

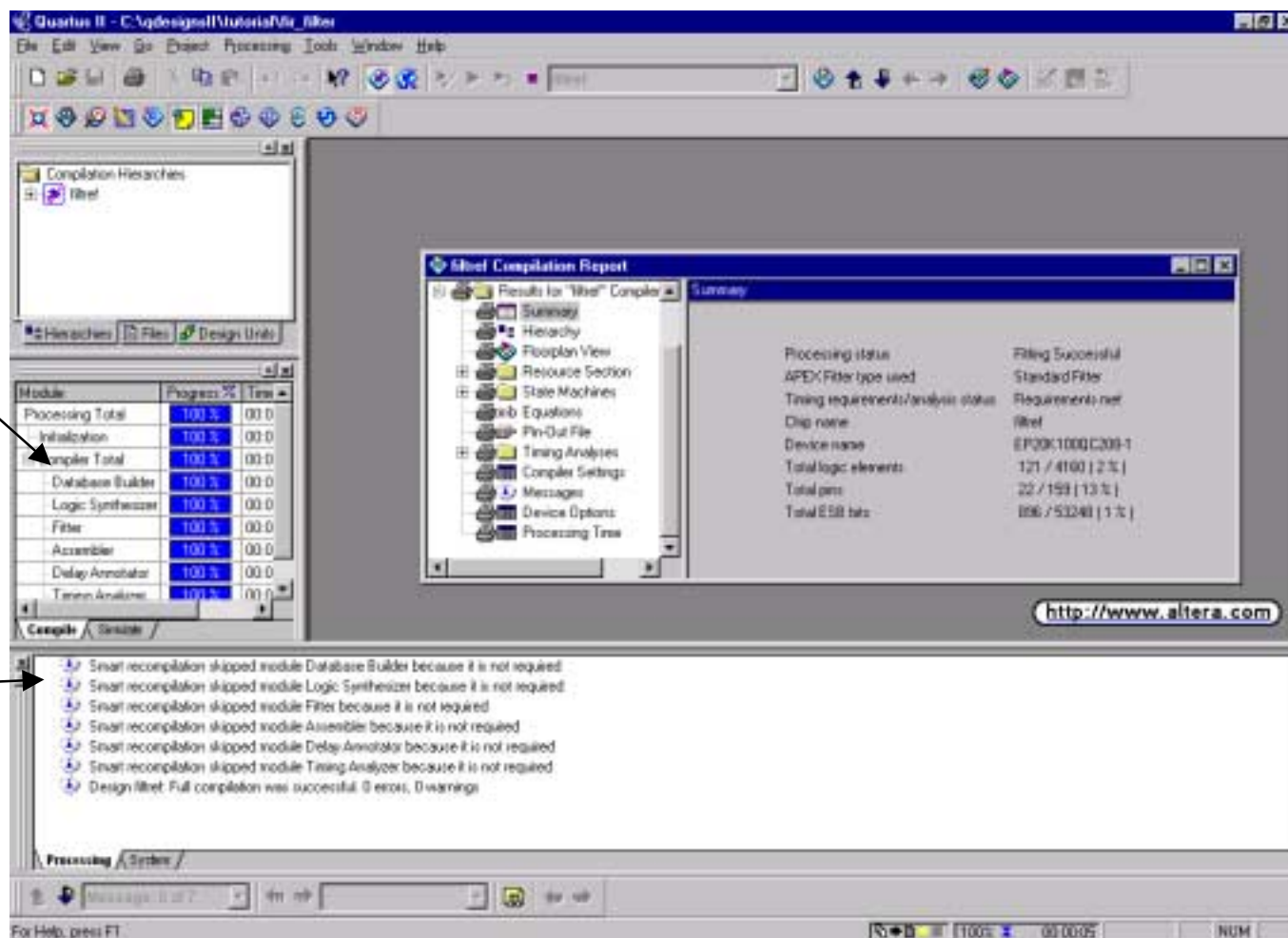


Quartus II 编译

2

状态栏

编译产生的信息



配置组织器

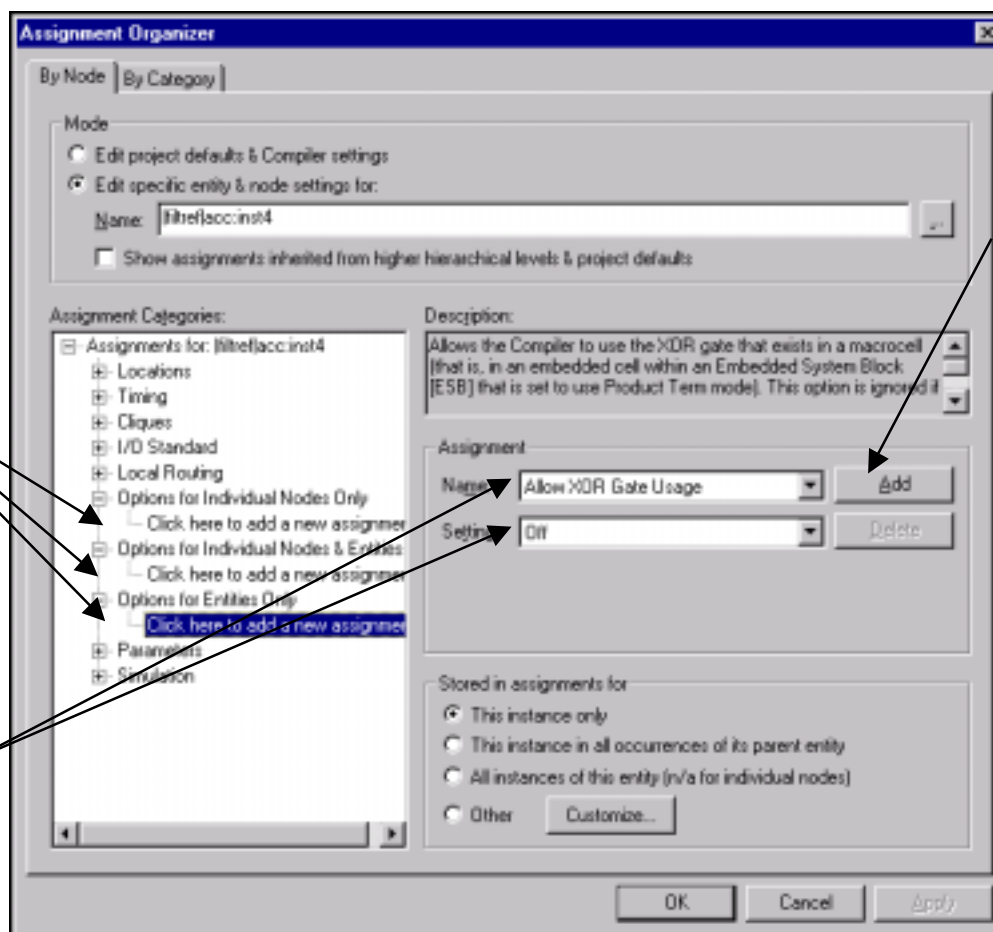
3种类型的逻辑选项

- 只给单独的接点的选项
- 给接点和模块的选项
- 只给模块的选项

1. 点击，增加新的配置

2. 选择一个已设定的配置

3. 点击 **Add** 去增加一项新的设置

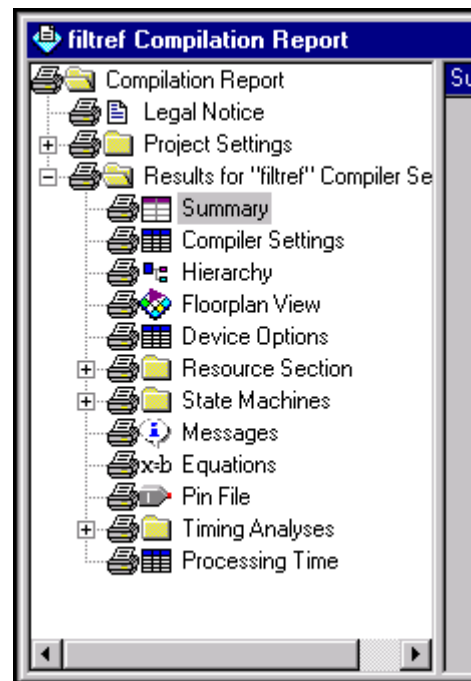


配置组织器的构成

- 定位
- 定时
 - 增加定时设置来提高特性
- 打包
 - 将选择好的逻辑模块放在一起来提高性能
- I/O 标准
- 局部走线
- 单独的逻辑选项
 - 利用Quartus II 的综合/配置的特殊选项，或ALTERA的器件特性s
 - 3 种类型
- 有参数限制的仿真

编译报告

- 包含了怎样将一个设计放到一个器件中的所有信息
 - 器件使用统计
 - 编译设置
 - 底层显示
 - 器件资源利用率
 - 状态机的实现
 - 方程式
 - 延时分析结果
 - CPU 使用资源
- 这是一个只读的窗口



Quartus II 的延时分析



特点

- Quartus II 支持对单个时钟或多个时钟的延时分析
- 单个时钟的延时分析
 - Fmax (最大时钟频率)
 - Tsu, Th, Tco (建立时间, 保持时间, 时钟到输出时间)
 - 整个系统的Fmax (包括引脚上的输入输出延时)
- 多个时钟的延时分析
 - 可以支持客户分析由不同时钟控制的寄存器之间的延时
 - 运用Slack进行分析
- 组合逻辑电路检测
 - Quartus II 会自动检测组合逻辑电路

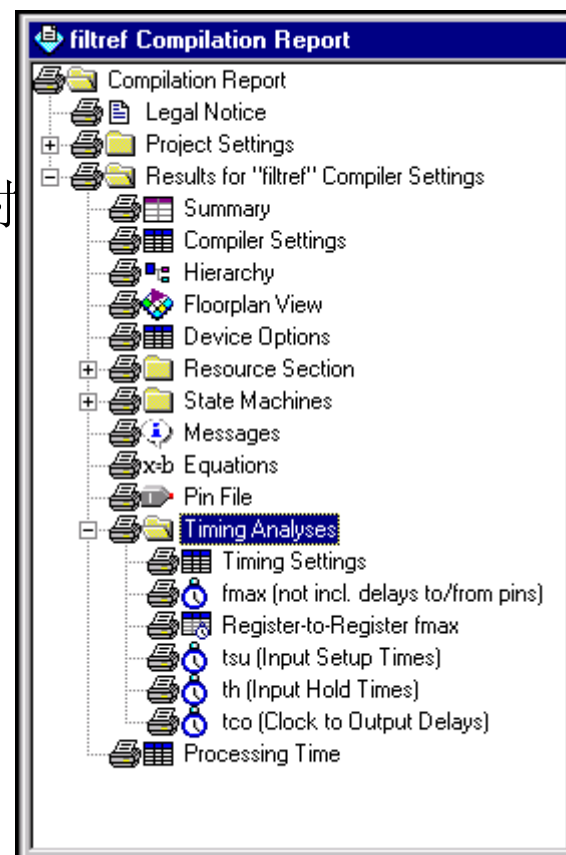
特点

- 不同类型的延时信息 (请参考编译部分)
 - 没有布局步线的延时信息
 - 经过布局步线的延时信息
 - 混合的树状层次型设计
- 在缺省情况下，在编译之后一般自动调用延时分析
 - 也可以禁止调用
- 生成的延时信息也可以以VHDL，Verilog 或 标准延时文件（SDF）的格式输出到第三方的EDA工具中

报告延时分析

■ 所有的延时分析信息都包含在编译报告中

- 总结
- 延时分析
- 内部fmax (不包括引脚上的输入输出延时)
系统fmax (包括引脚上的输入输出延时)
- 寄存器到寄存器的延时报表
- tsu (输入建立时间)
- th (输入保持时间)
- tco (时钟到输出时间)
- tpd (引脚到引脚的延时)



例如:Fmax (包含输入输出引脚的延时)

在缺省情况下最坏的 fmax 放在最上面

选择 fmax (包含输入输出引脚的延时)

Fmax 值

Clock Name	Required fmax	Actual fmax (period)
clk	None	37.66 MHz period = 26.551 ns
next	None	37.66 MHz period = 26.551 ns
yn[0]	None	38.55 MHz period = 25.939 ns
yn[1]	None	38.55 MHz period = 25.939 ns
	None	38.55 MHz period = 25.939 ns
	None	38.56 MHz period = 25.932 ns
tapsinst1ben[4]"seg0	None	73.35 MHz period = 13.634 ns
tapsinst1ben[1]"seg0	None	75.27 MHz period = 13.286 ns
tapsinst1ben[2]"seg0	None	75.53 MHz period = 13.239 ns
tapsinst1ben[3]"seg0	None	75.55 MHz period = 13.237 ns

外部的延时可以在信息窗口中列出来

Processing Total 100% 00:0
Initialization 100% 00:0
Compiler Total 100% 00:0
Database Builder 100% 00:0
Logic Synthesizer 100% 00:0

Micro setup delay of destination is 0.159 ns
Clock clk has System fmax of 37.66 MHz between source register state_minst2filter~19 and destination pin next (period= 26.551 ns)
Longest clock path from clock clk to source register is 2.770 ns
Micro clock to output delay of source is 0.276 ns
Longest register to pin delay is 3.505 ns
External delay of pin is 20.0 ns

Quartus II 仿真

支持多种仿真方法

■ 波形方式输入

- .vwf (向量波形文件) – 是 Quartus II 中最主要的波形文件
- .vec (向量文件) – 是MAX+PLUS II 中的文件. 主要是为了向下兼容
- .tbl (列表文件) – 用来将MAX+PLUS II 中的.scf 文件输入到 Quartus II 中.

■ 支持Testbench

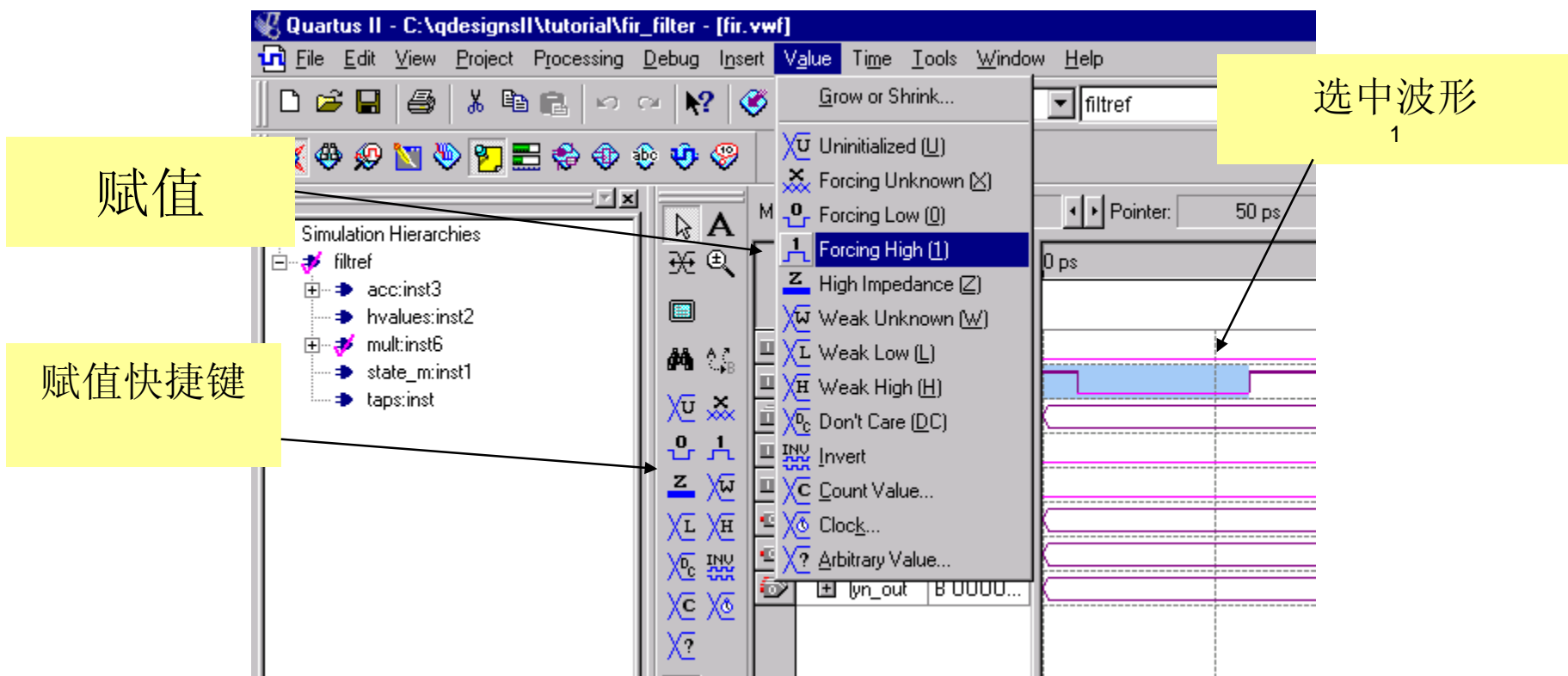
- Tcl/TK 脚本文件

■ 第三方的仿真工具

- Verilog/VHDL Testbench

描绘仿真波形

- 先选中准备要赋值的部分
- 添入初始值

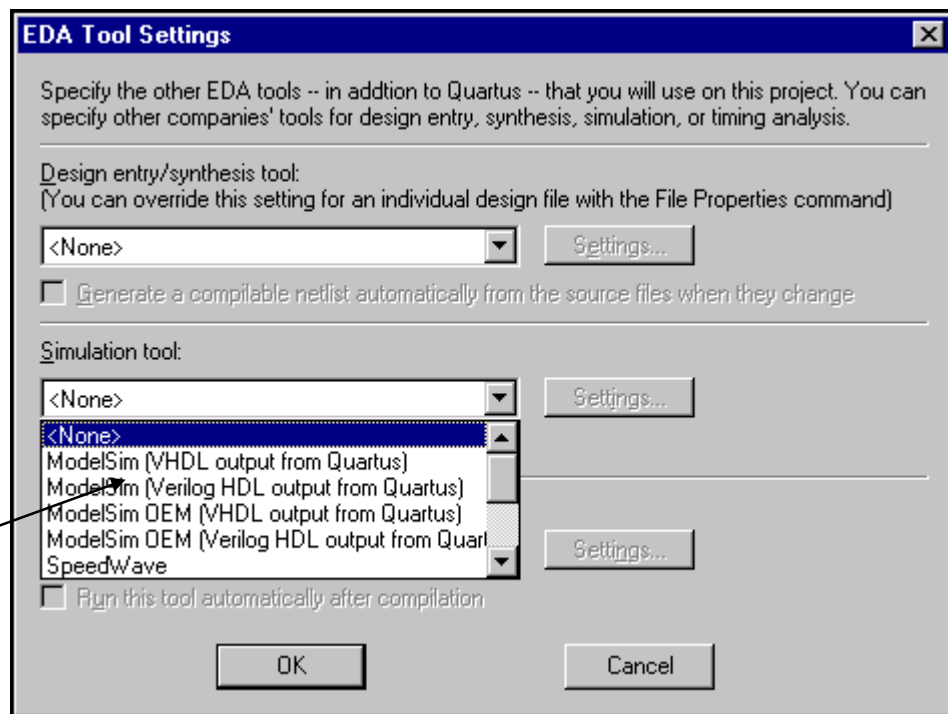


利用第三方的仿真工具进行仿真

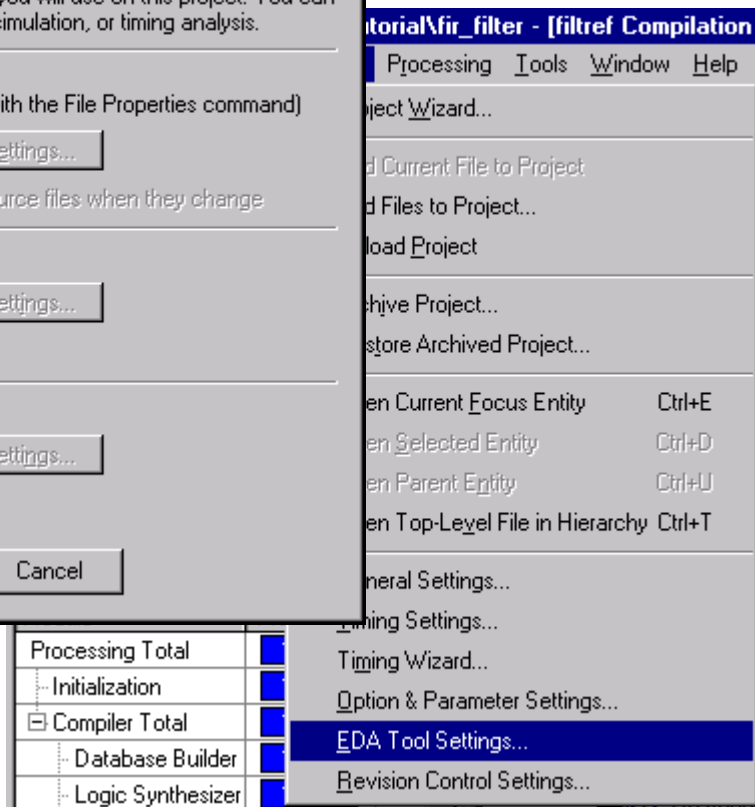
- Model Technology (ModelSim)
- Cadence (VERILOG-XL)
- Synopsys (VCS)
- Synopsys (VSS)

阐明仿真器

■ 从project 菜单中选择 EDA 工具设置



选择仿真工具



第三方仿真工具

■ VHDL 仿真器

- 利用 Quartus II 产生.VHO 和 .SDO 文件
- 利用在sim_lib 目录中的APEX20K_ATOMs.VHD 和 APEX20K_COMPONENTS.VHD 文件

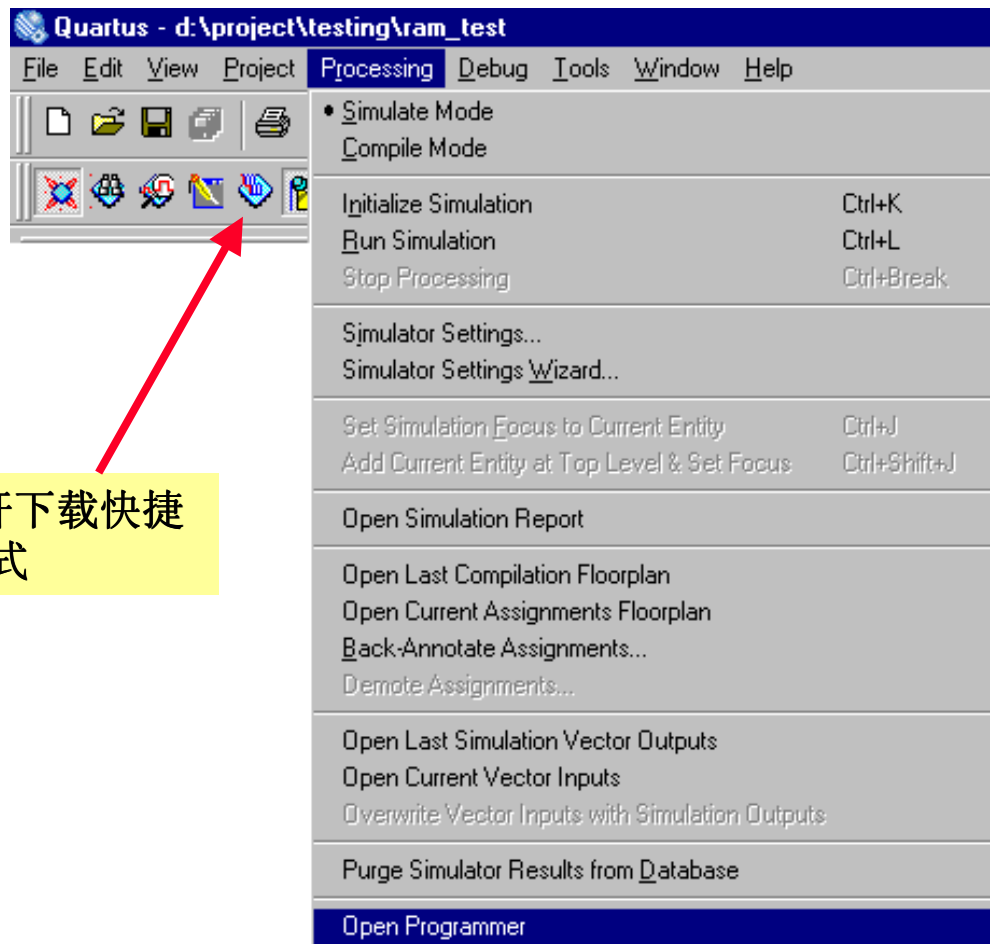
■ Verilog 仿真器

- 利用 Quartus II 产生.VO 和 .SDO
- 利用在sim_lib APEX20K_ATOMS.VO

Quartus II 下载

打开下载窗口

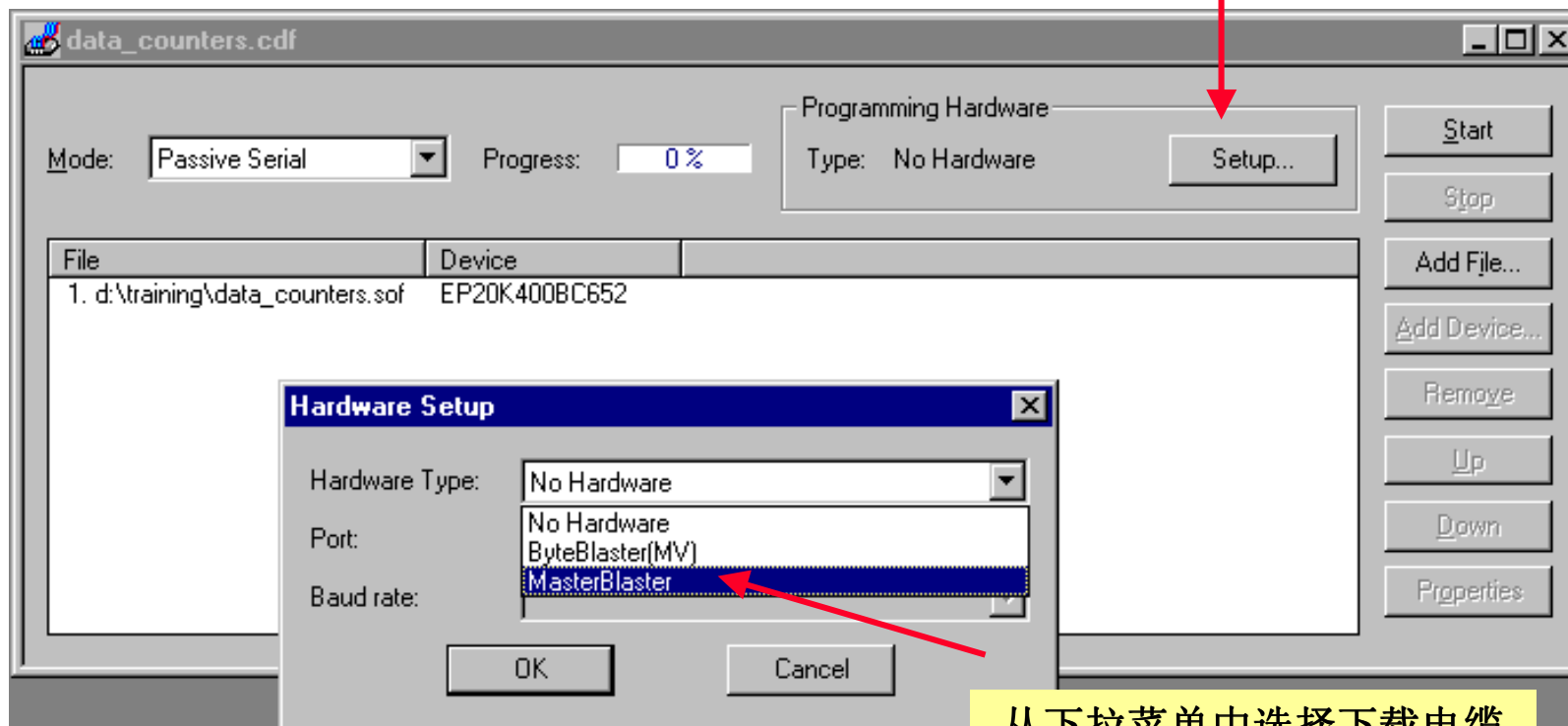
■ Processing → Open Programmer



或 使用打开下载快捷方式

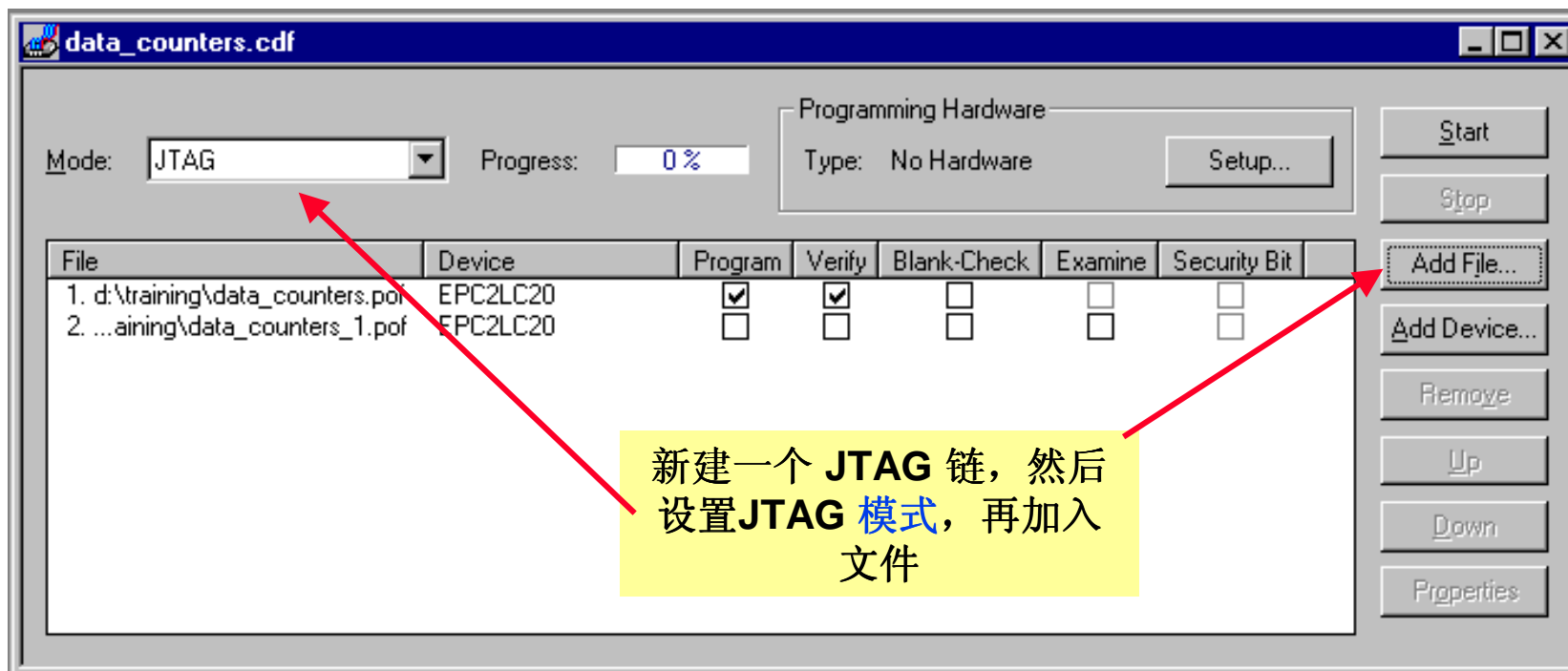
设置下载电缆

选择setup进行设置



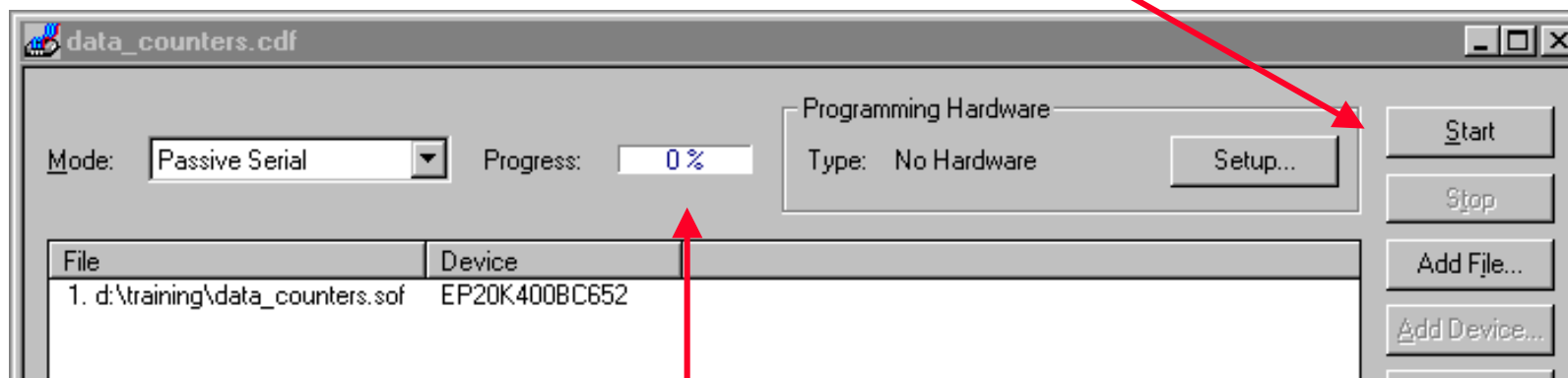
从下拉菜单中选择下载电缆

设置 JTAG 链



开始下载

当你的下载链中的文件与下载电缆已经设置好了之后，点击**开始**按钮开始下载



进度表中显示目前完成下载的比率