

**ALTERA.**

# Quartus II 软件使用教程

**Cytech-XA**

Vincent Song

Q2 2008



# Programmable Logic Families

- **Structured ASIC**
    - HardCopy® II & HardCopy Stratix
  - **High & medium density FPGAs**
    - Stratix III, Stratix II & Stratix
  - **Low-cost FPGAs**
    - Cyclone III, Cyclone II & Cyclone
  - **FPGAs w/ clock data recovery**
    - Stratix II GX & Stratix
  - **Low-cost 90-nm FPGAs for PCI Express, Gigabit Ethernet, and Serial RapidIO up to 2.5 Gbps**
    - Arria GX
- CPLDs**
- MAX II, MAX 7000 & MAX 3000
- **Configuration devices**
  - Serial (EPCS) & enhanced (EPC)



**HardCopy® II**



**Stratix® III**



**Cyclone® III**  
65-nm low-cost FPGAs



**Arria™ GX**



**Stratix® II GX**

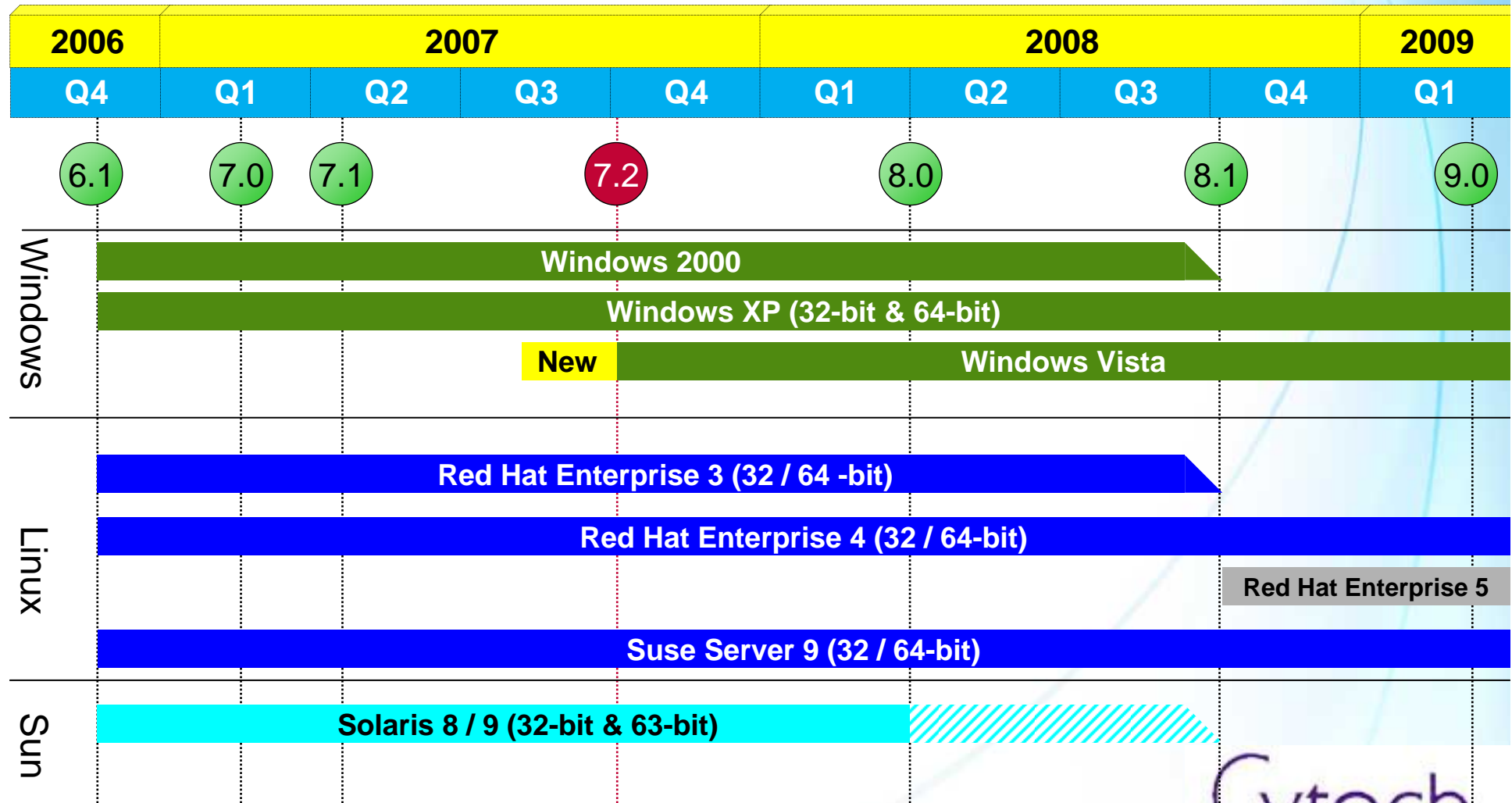


**MAX® II**



**Cytech**  
Technology

# QuartusII软件发布RoadMap



# 从QII 6.1开始支持多核处理器和64位OS

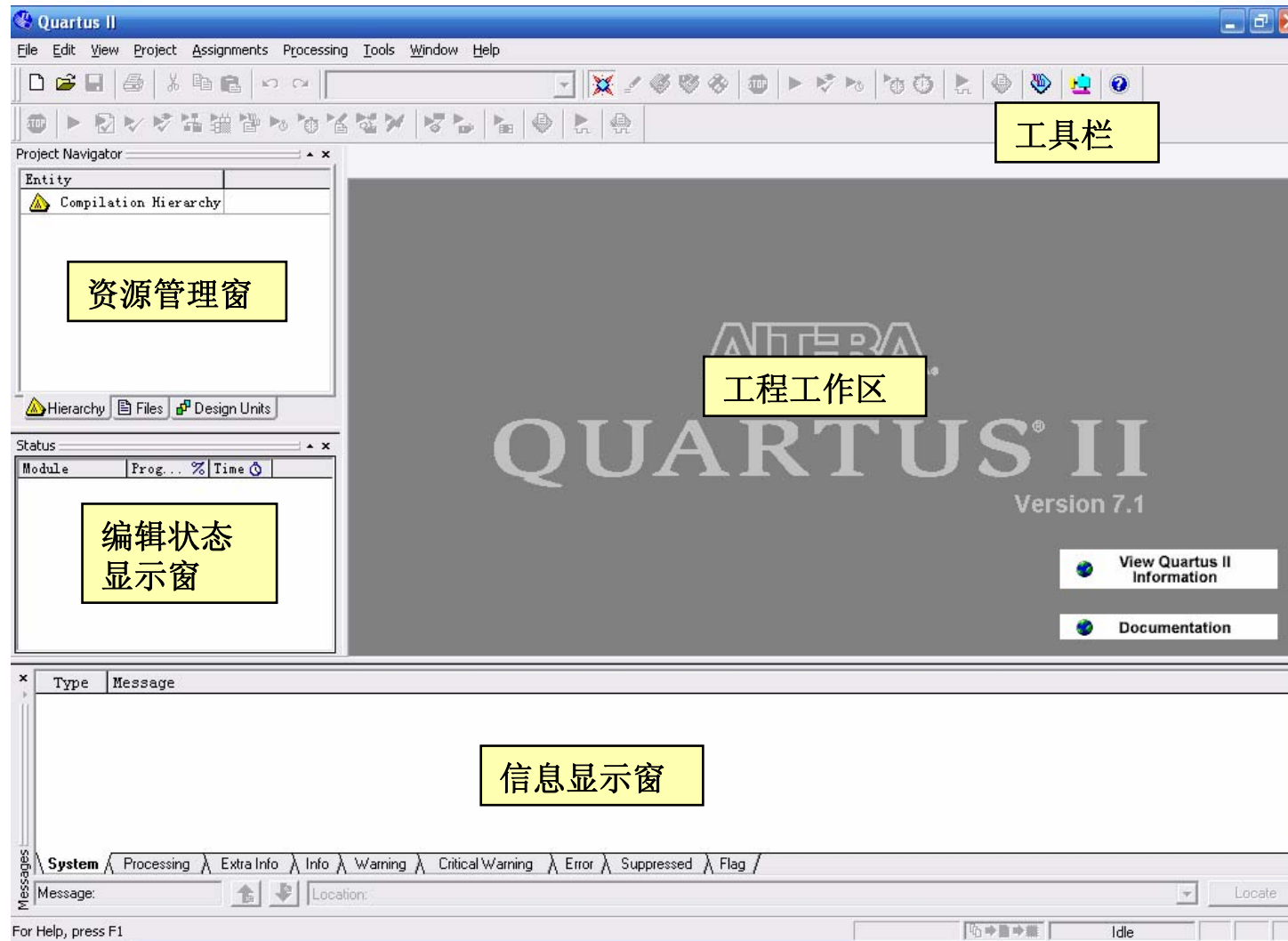
- Multi-processor cores now mainstream
  - Benefit → faster compile times



- 64-Bit O/S – moving mainstream
  - Benefit → access to more than 2 GB of memory



# Quartus II开发环境



# 主要快捷键

The image shows the Quartus II IDE interface with several callout boxes pointing to specific features:

- Settings**: Points to the 'Tools' menu.
- Assignment Editor**: Points to the 'Assignments' menu.
- Pin Planner**: Points to the 'Pin Planner' icon in the toolbar.
- Chip Planner (Floorplan & Chip Editor)**: Points to the 'Floorplan' icon in the toolbar.
- Execution controls**: Points to a group of icons including 'Run', 'Stop', and 'Refresh'.
- Compilation report**: Points to the 'Compilation Report' icon in the toolbar.
- Programmer**: Points to the 'Programmer' icon in the toolbar.

A red box highlights a set of icons in the toolbar, including 'Stop', 'Run', 'Refresh', 'Compile', 'Open Project', 'Save', 'Print', 'Find', 'Find Next', 'Find Previous', 'Find All', 'Find in Files', 'Find in Project', 'Find in Package', 'Find in Library', 'Find in Database', 'Find in System', 'Find in User', 'Find in Workspace', 'Find in Recent', 'Find in History', 'Find in Favorites', 'Find in Recent Projects', 'Find in Recent Files', 'Find in Recent Packages', 'Find in Recent Libraries', 'Find in Recent Databases', 'Find in Recent Systems', 'Find in Recent Users', 'Find in Recent Workspaces', 'Find in Recent Recent Projects', 'Find in Recent Recent Files', 'Find in Recent Recent Packages', 'Find in Recent Recent Libraries', 'Find in Recent Recent Databases', 'Find in Recent Recent Systems', 'Find in Recent Recent Users', 'Find in Recent Recent Workspaces'.

The 'Customize' dialog box is open, showing the 'Toolbars' tab. The 'Standard Quartus II' toolbar is selected. The 'Processing' checkbox is checked. The 'Show tooltips' checkbox is checked. The 'Borderless look' checkbox is checked. The 'Large buttons' checkbox is unchecked. The 'New...' button is visible. The 'Reset' button is visible. The 'Reset All' button is visible.

**To open step by step compilation flow:**  
1. Tools ⇒ Customize ⇒ Toolbars  
2. Select "Processing" Check Box



# Agenda

- 设计流程概要
- 建立工程
- 设计输入
- 编译
- 综合
- 使用Synplify Pro做综合
- 布局布线
- Assignment Editor
- 管脚分配
- 仿真
- 器件编程
- 时序约束
- SignalTap II 逻辑分析仪

**ALTERA.**

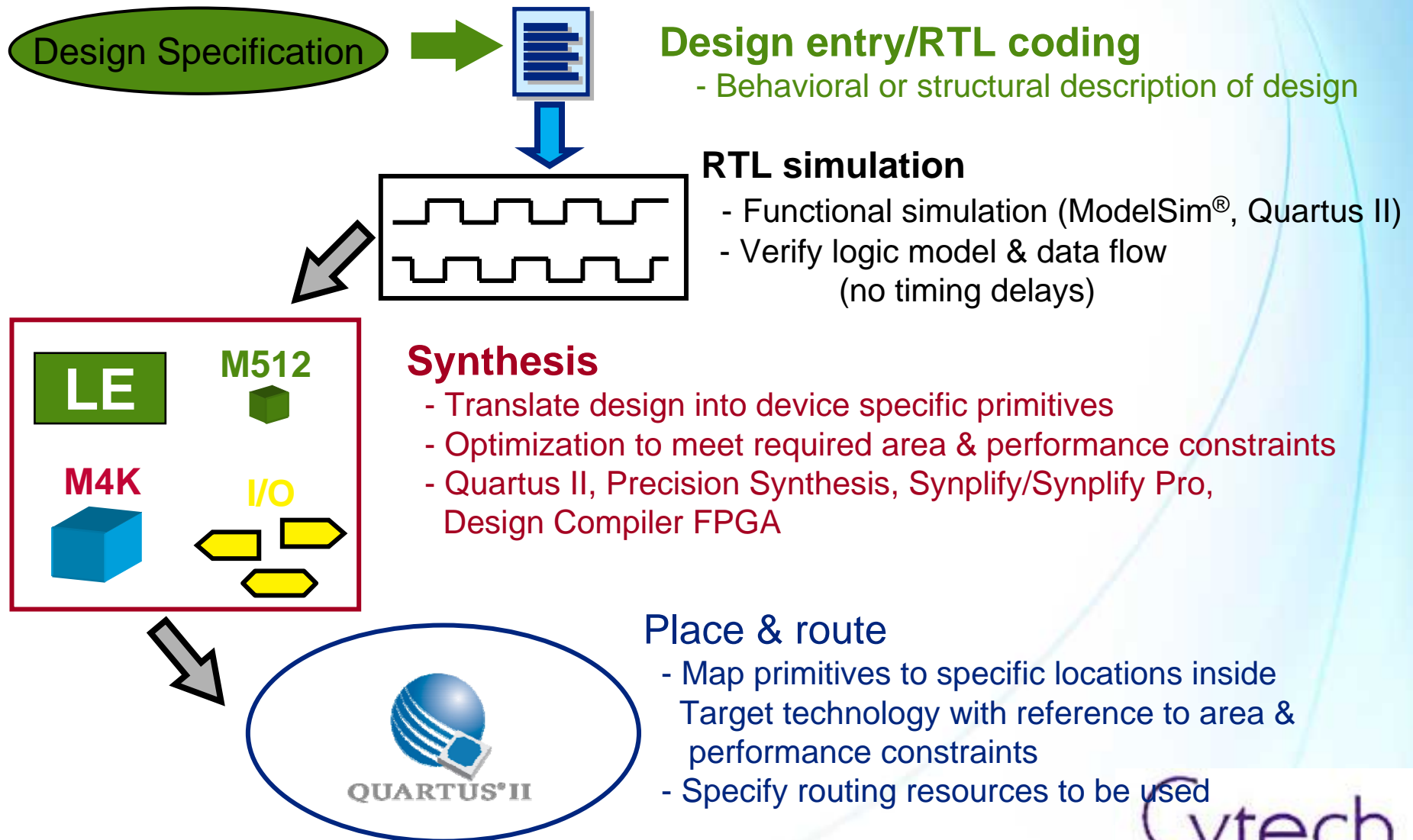
# Quartus II 软件使用教程

设计流程概要

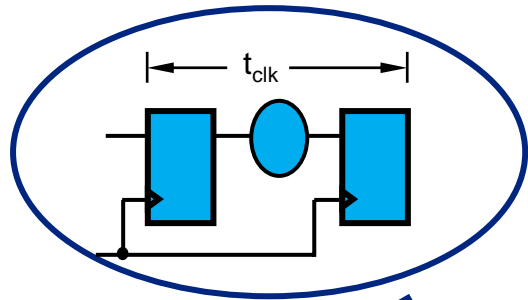




# Typical PLD Design Flow

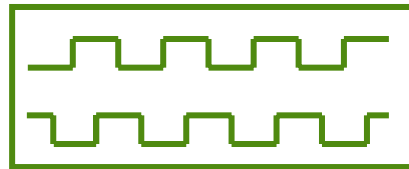


# Typical PLD Design Flow



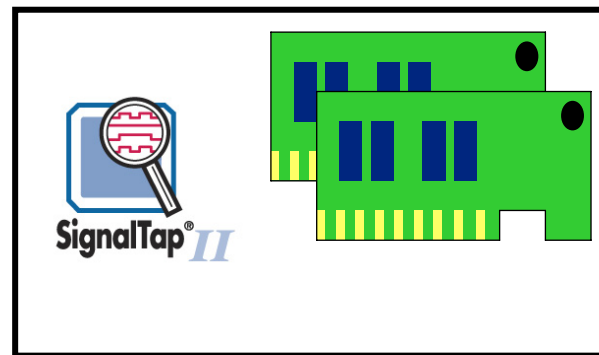
## Timing analysis

- Verify performance specifications were met
- Static timing analysis



## Gate level simulation

- Timing simulation
- Verify design will work in target technology



## PC board simulation & test

- Simulate board design
- Program & test device on board
- Use **SignalTap II** for debugging

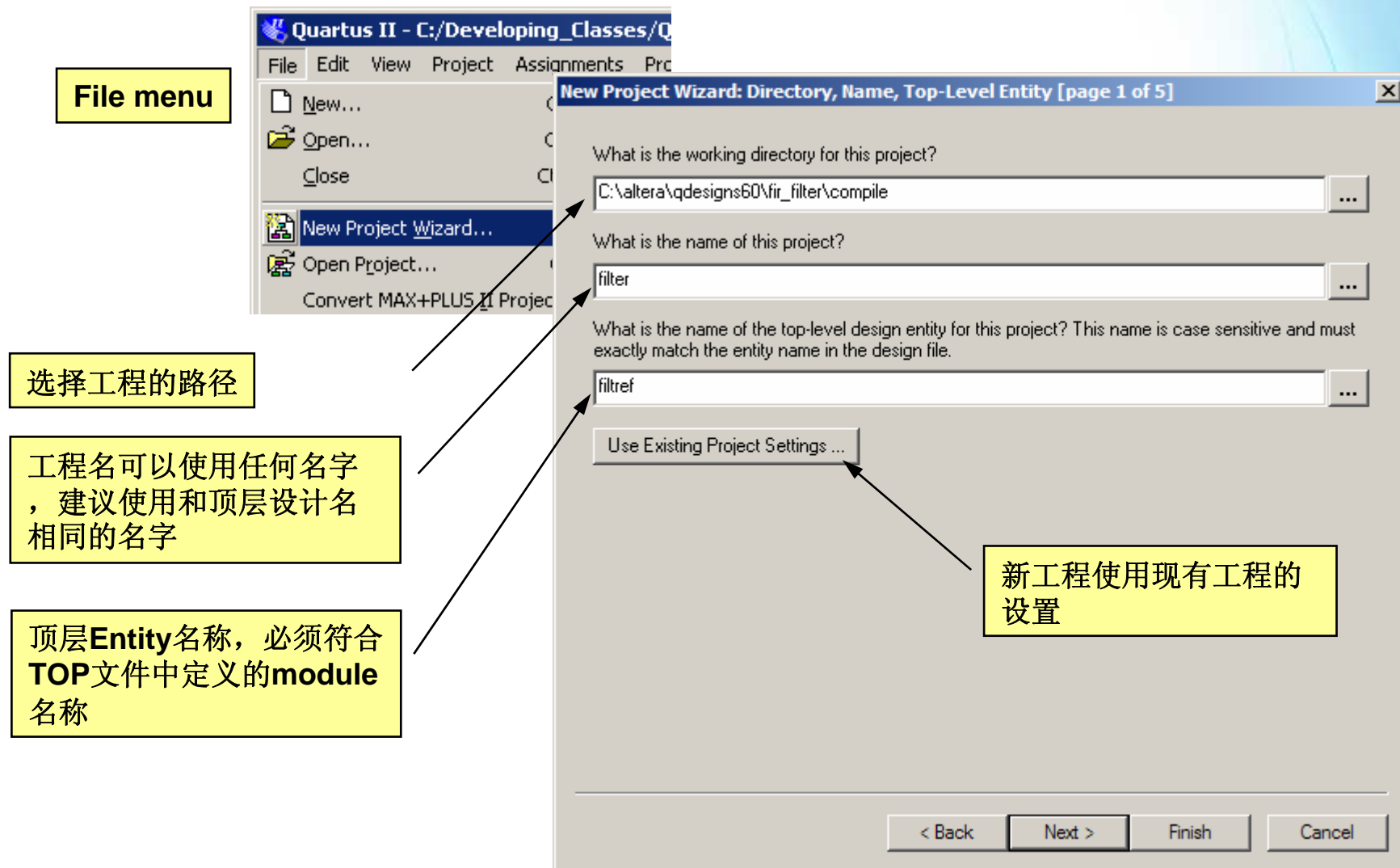
**ALTERA.**

# Quartus II 软件使用教程

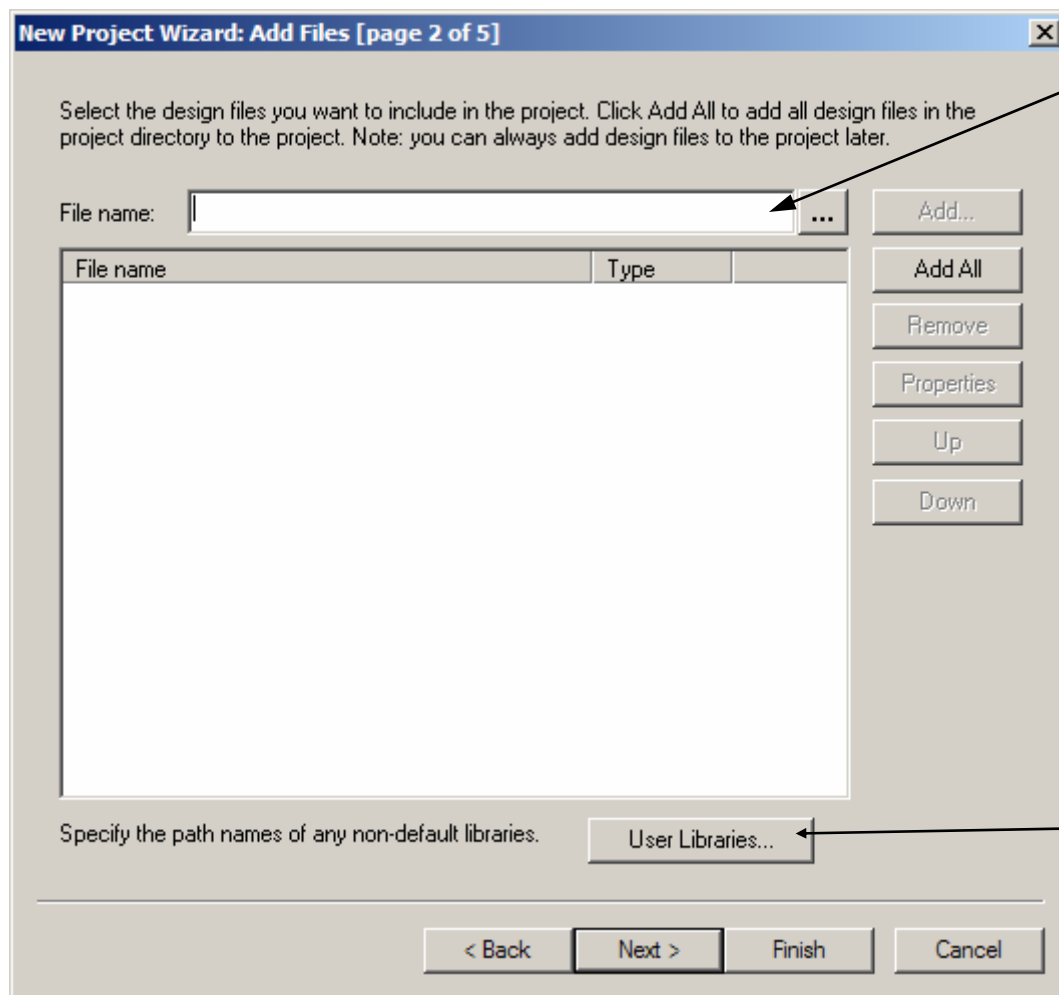
建立工程



# 设计新工程使用New Project Wizard比较方便



# 添加源文件（这一步骤可以跳过）



## Add design files

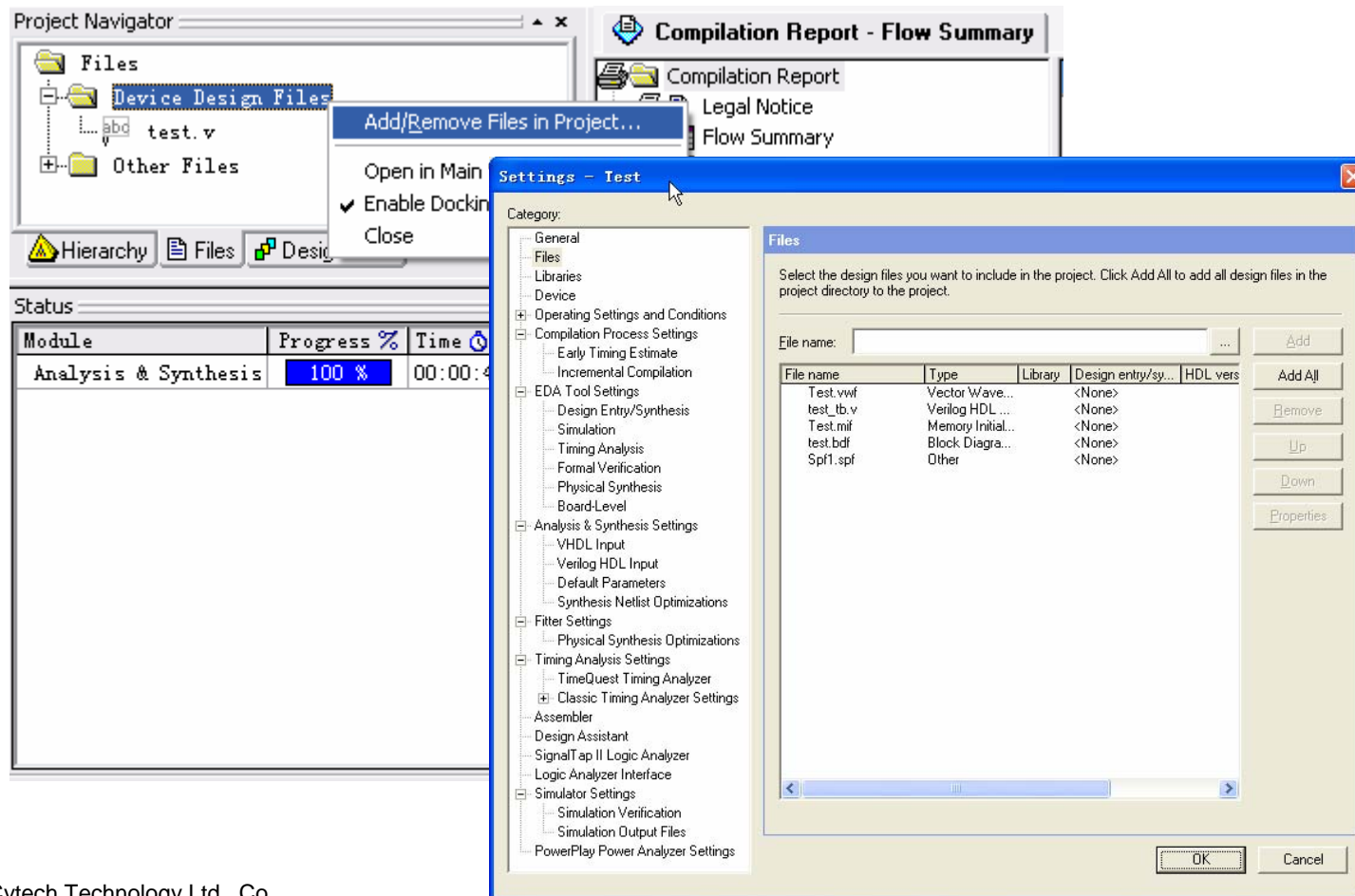
- Graphic (.BDF, .GDF)
- AHDL
- VHDL
- Verilog
- EDIF
- VQM

## Add user library pathnames

- User libraries
- MegaCore®/AMPP<sup>SM</sup> libraries
- Pre-compiled VHDL packages



如果跳过新建向导的**Add File**，可以在工程生产完毕之后，在导航界面的**File**下点击“**Device Design Files**”，右键弹出菜单选择“**Add/Remove Files in Project**”



# 选择器件

New Project Wizard: Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Family:

Target device

☐ Auto device selected by the Filter

☒ Specific device selected in 'Available devices' list

Show in 'Available device' list

Package:

Pin count:

Speed grade:

Core voltage: 1.2V

☒ Show Advanced Devices

Available devices:

Name	ALUTs	Memory Bits	DSP	PLL	DLL
EP2S15F484C3	12480	419328	12	6	2
EP2S15F484C4	12480	419328	12	6	2
EP2S15F484C5	12480	419328	12	6	2
EP2S15F484I4	12480	419328	12	6	2
EP2S30F484C3	27104	1369728	16	6	2
EP2S30F484C4	27104	1369728	16	6	2
EP2S30F484C5	27104	1369728	16	6	2
EP2S30F484I4	27104	1369728	16	6	2

Companion device

HardCopy II:

☒ Limit DSP & RAM to HardCopy II device resource

< Back Next > Finish Cancel

选择器件系列

**Package**可以选择器件的封装，**Pin count**可以选择器件的引脚数，**Speed grade**可以选择器件的速度等级，这些选项可以缩小可用器件列表的范围，以便快速找到需要的目标器件。

# EDA 工具设置

选择综合、仿真、时序分析等第三方工具

New Project Wizard: EDA Tool Settings [page 4 of 5]

Specify the other EDA tools -- in addition to the Quartus II software -- used with the project.

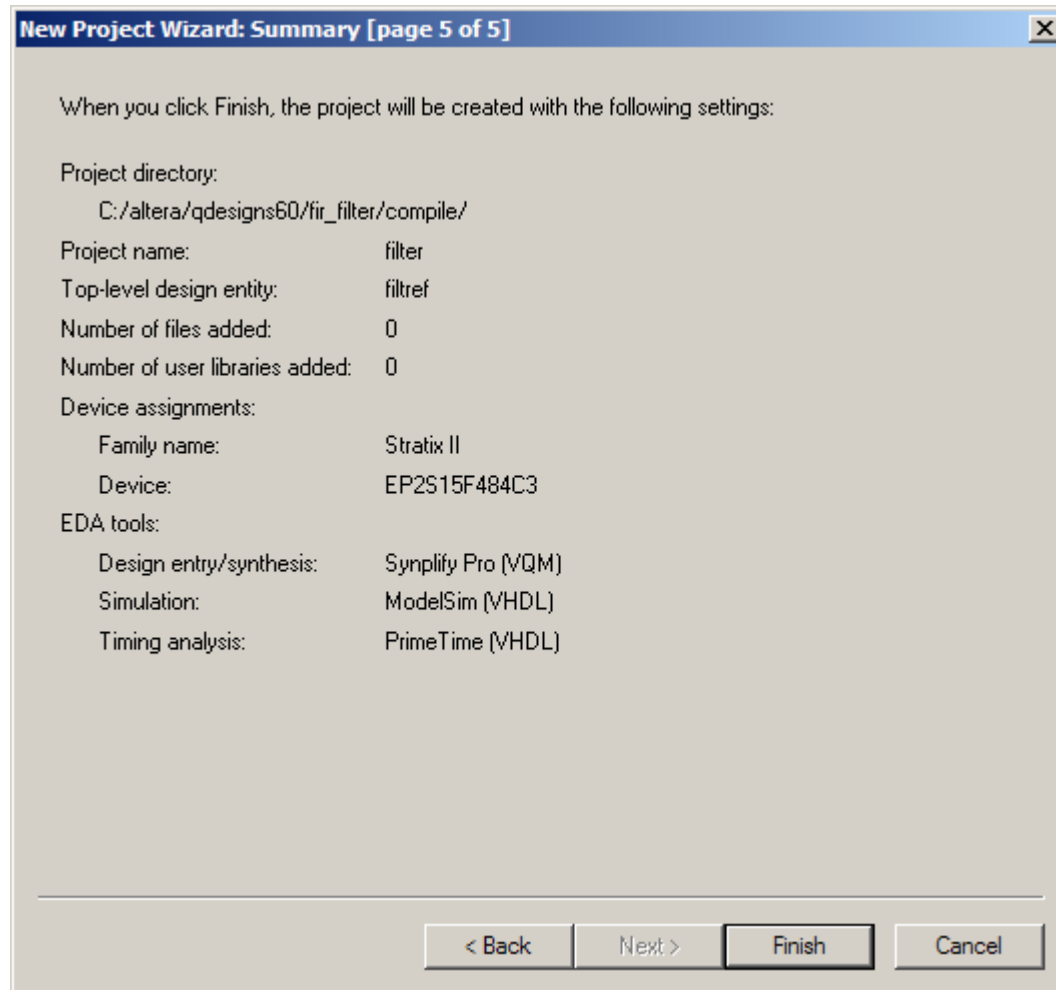
☒ EDA design entry/synthesis tool: Synplify Pro  
Format: VQM  
☐ Run this tool automatically to synthesize the current design

☒ EDA simulation tool: ModelSim  
Format: Verilog  
VHDL  
Verilog  
☐ Run this tool automatically after compilation

☒ EDA timing analysis tool: PrimeTime  
Format: VHDL  
☐ Run this tool automatically after compilation

< Back Next > Finish Cancel

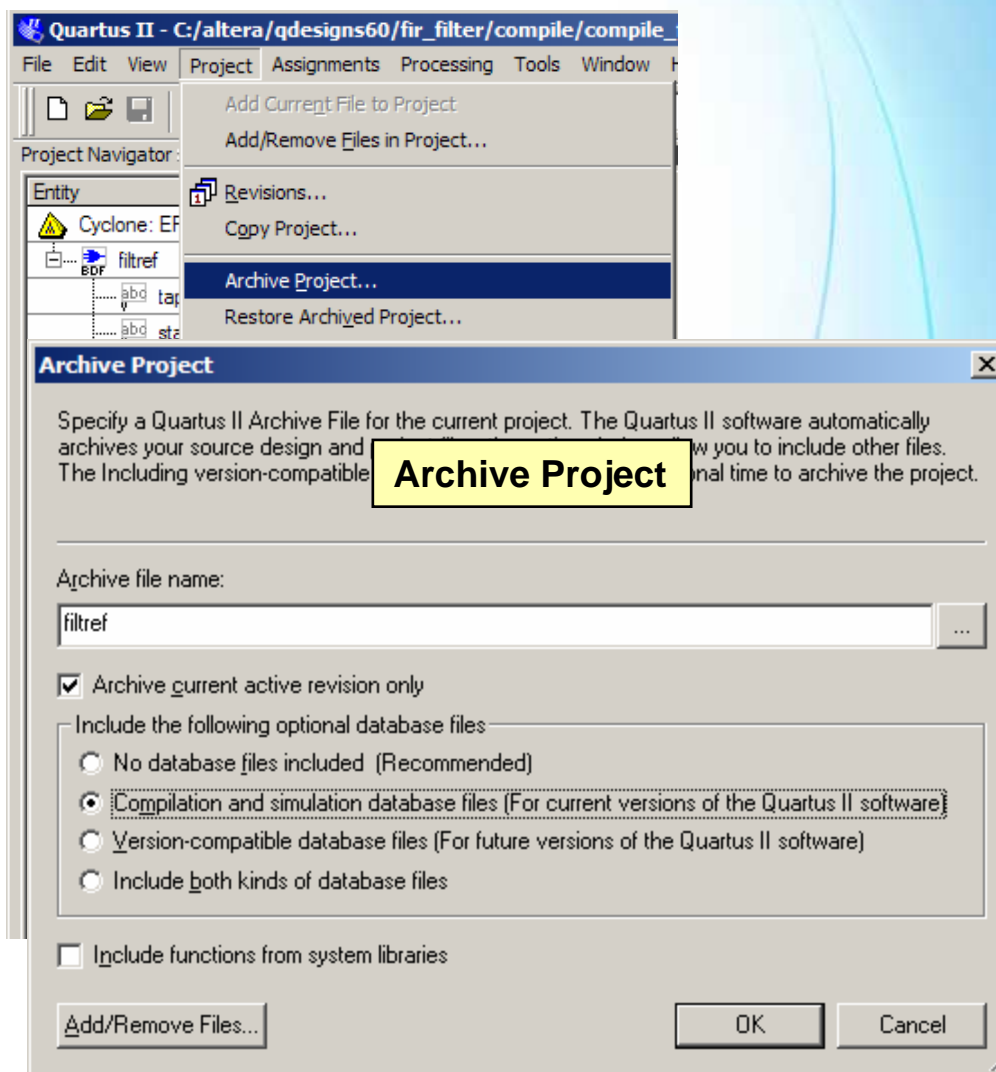
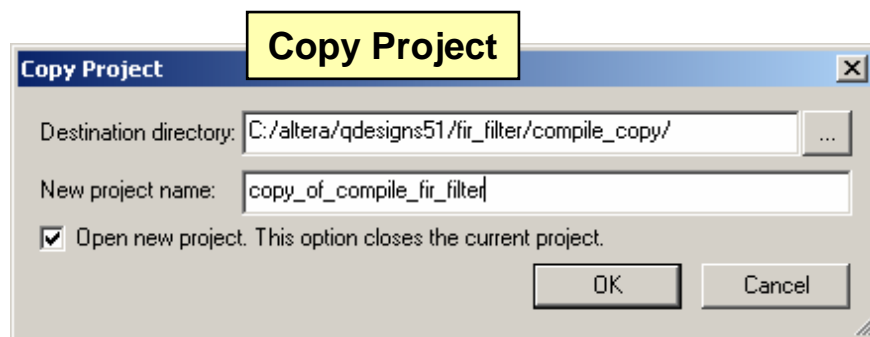
# 完成!



确认全部参数设置，  
若无误则单击**Finish**  
按钮，完成工程的创  
建；若有误，可单击  
**Back**按钮返回，重新  
设置。

# 工程管理

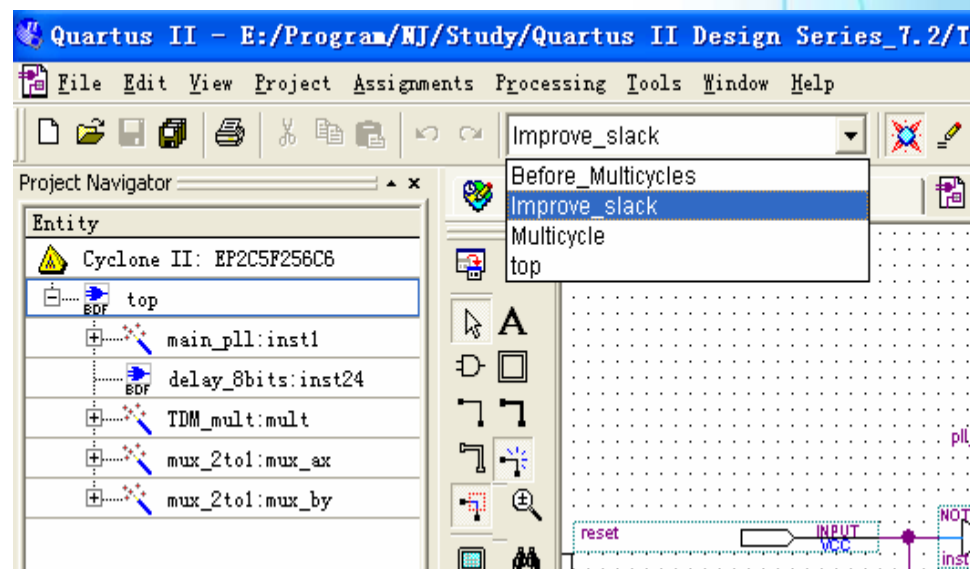
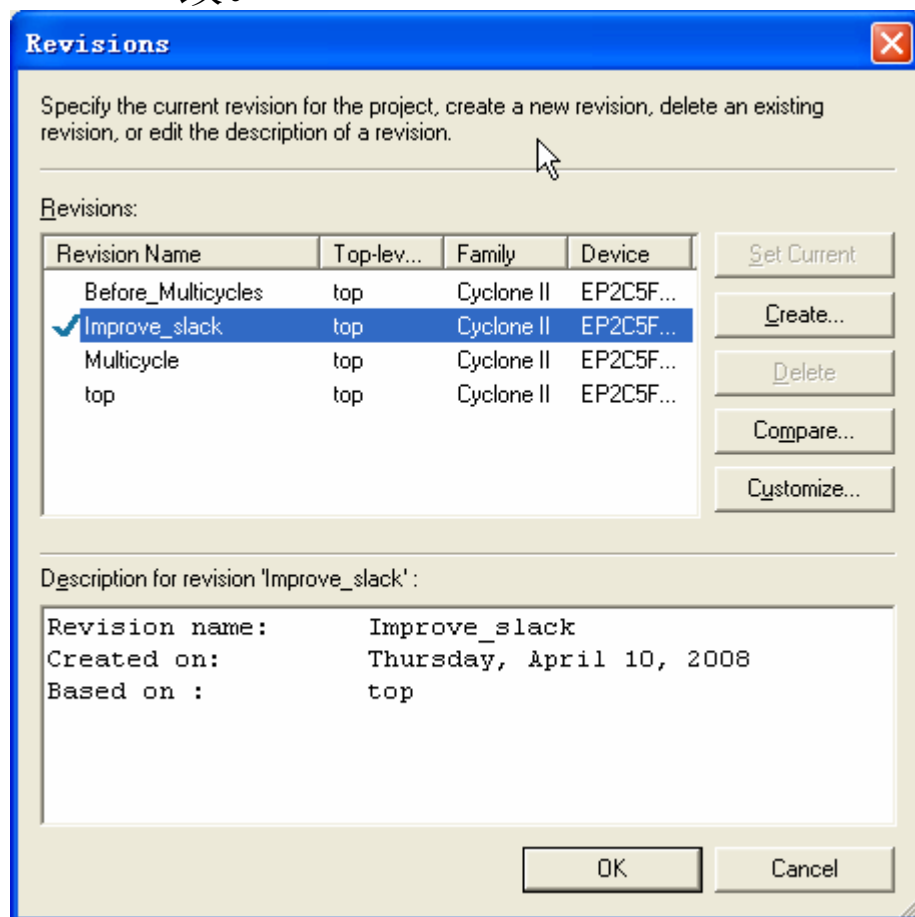
- 工程打包
  - 生成.qar文件
- 工程复制





# 版本管理

- 通过菜单**Project -> Revisions**打开版本管理窗口，可以在原工程的基础上建立多个版本，并且可以比较，方便设计。
- 注意：不同的版本只能对约束做更改，如果更改原设计，则所有版本均会更改。



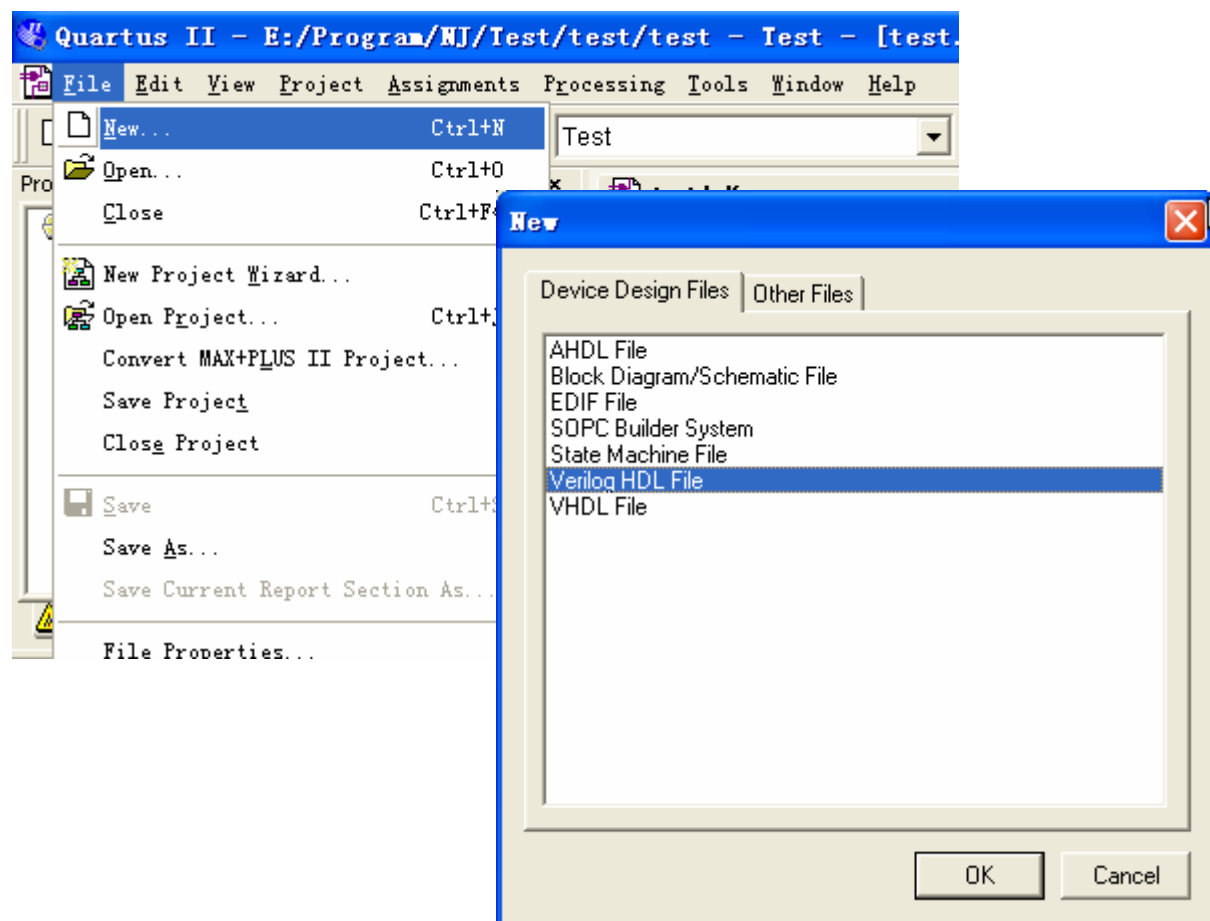
**ALTERA.**

# Quartus II 软件使用教程

设计输入



# 新建一个设计文件



选择要创建的文件  
类型

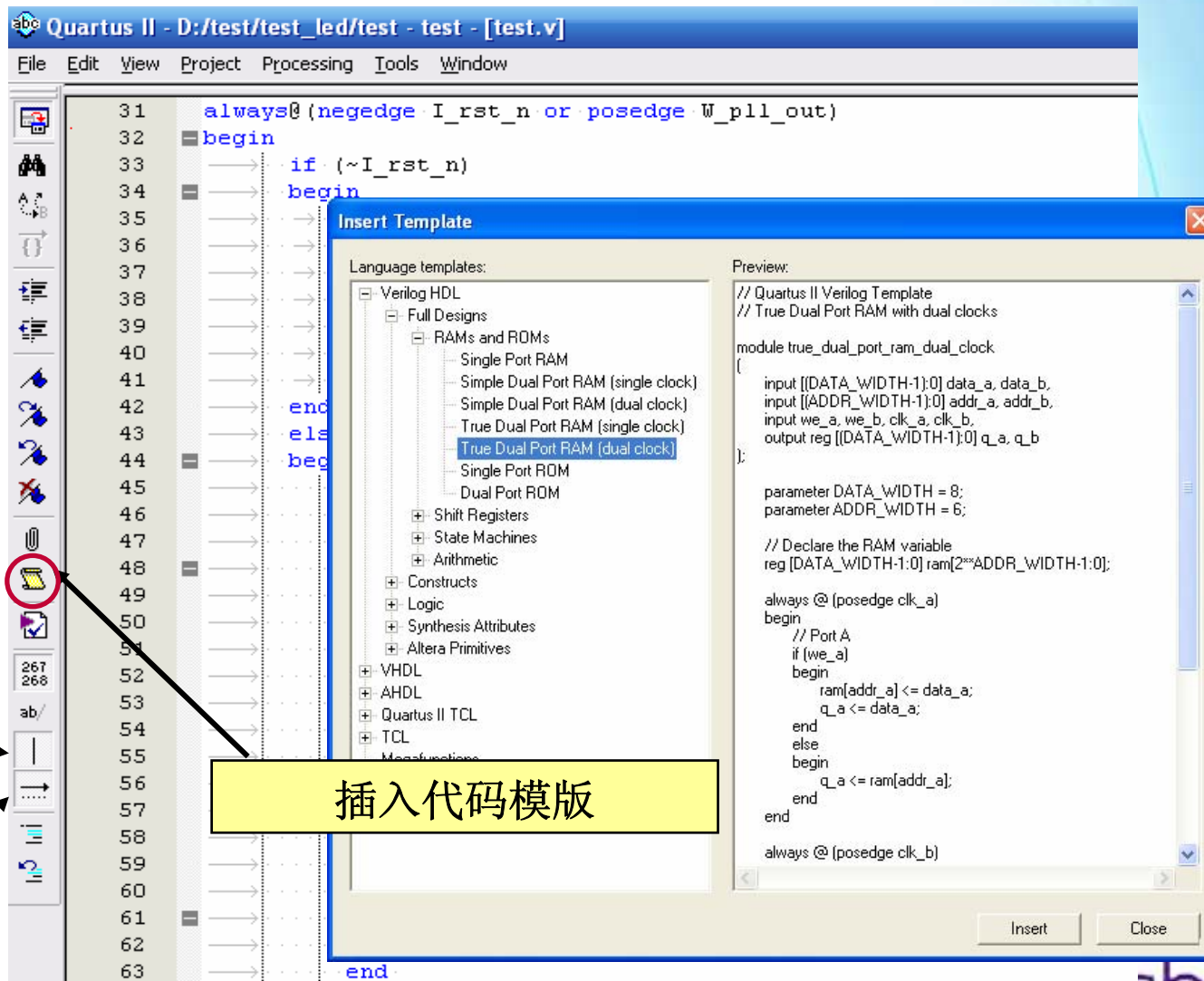
# QII7.1文本编辑器

独立/整合窗口切换

“Alt”键实现列操作的切换

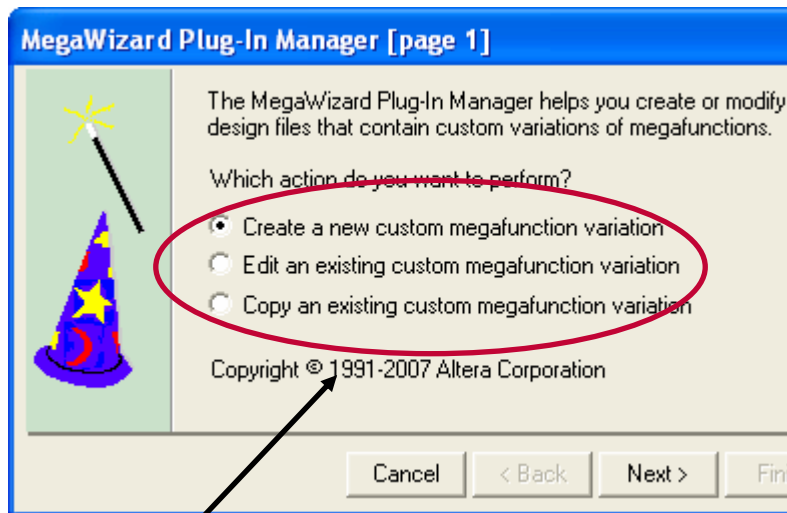
列对齐显示标记

行对齐显示标记



# 使用MegaWizard Plug-in Manager调用宏功能模块

Tools ⇒ MegaWizard Plug-In Manager



可以创建一个新的IP文件，也可以编辑已有的IP文件，或者拷贝已创建的文件。



选择megafunction 或IP



# MegaWizard示例

察看本机和互联网上帮助文档

MegaWizard Plug-In Manager - RAM: 2-PORT [page 3 of 12]

**RAM: 2-PORT**  
Version 7.1

About Documentation

1 Parameter Settings 2 EDA 3 Summary

General Widths/Blk Type Clks/Rd, Byte En Regs/Clkens/Aclrs Output1 Mem Init

Currently selected device family: Stratix II

Block Type: AUTO

Resource Usage  
256 ram\_bits (AUTO)

How will you be using the dual port ram?

- ☒ With one read port and one write port
- ☐ With two read/write ports

How do you want to specify the memory size?

- ☒ As a number of words
- ☐ As a number of bits

Cancel < Back Next > Finish

# MegaWizard示例

MegaWizard Plug-In Manager - RAM: 2-PORT [page 10 of 10] -- Summary

**RAM: 2-PORT**

1 Parameter Settings 2 EDA 3 Summary

data[7..0] wraddress[4..0] wren rdaddress[4..0] clock

32 Word(s) RAM

q[7..0]

Block Type: AUTO

Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory:  
E:\Program\NJ\Test\test\

File	Description
<input checked="" type="checkbox"/> ram.v	Variation file
<input type="checkbox"/> ram.inc	AHDL Include file
<input checked="" type="checkbox"/> ram.cmp	VHDL component declaration file
<input checked="" type="checkbox"/> ram.bsf	Quartus II symbol file
<input checked="" type="checkbox"/> ram_inst.v	Instantiation template file
<input checked="" type="checkbox"/> ram_bb.v	Verilog HDL black-box file
<input checked="" type="checkbox"/> ram_waveforms.html	Sample waveforms in summary
ram_wave*.jpg	Sample waveform file(s)

Cancel < Back Next > Finish

- 默认
  - HDL 源文件
  - symbol文件(.bsf)
- 可选
  - 器件声明文件 (.cmp)
  - 例化模型文件(\_int.v)
  - 黑盒子文件(\_bb.v)
  - 示例波形 (.html)

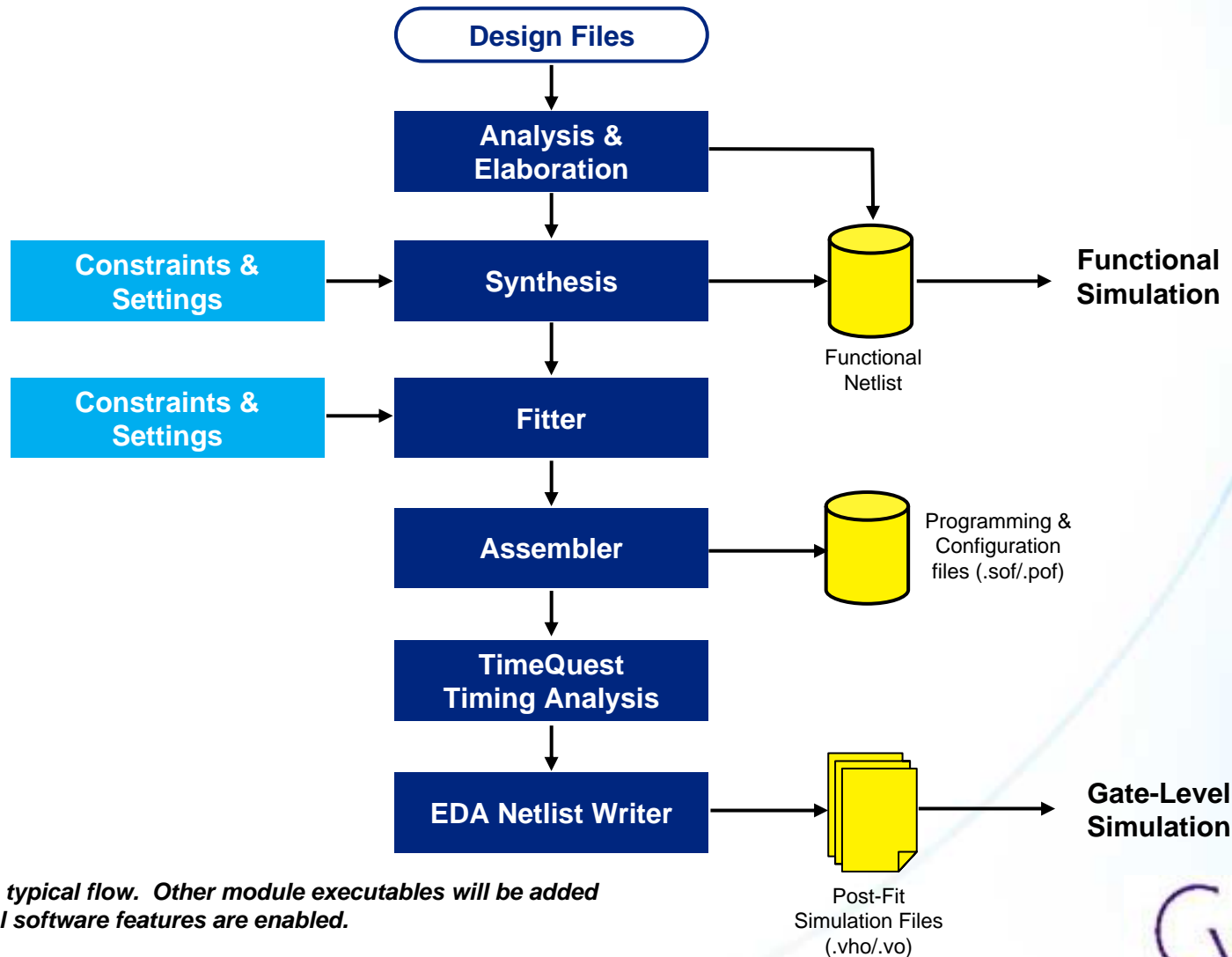
**ALTERA.**

# Quartus II 软件使用教程

编译



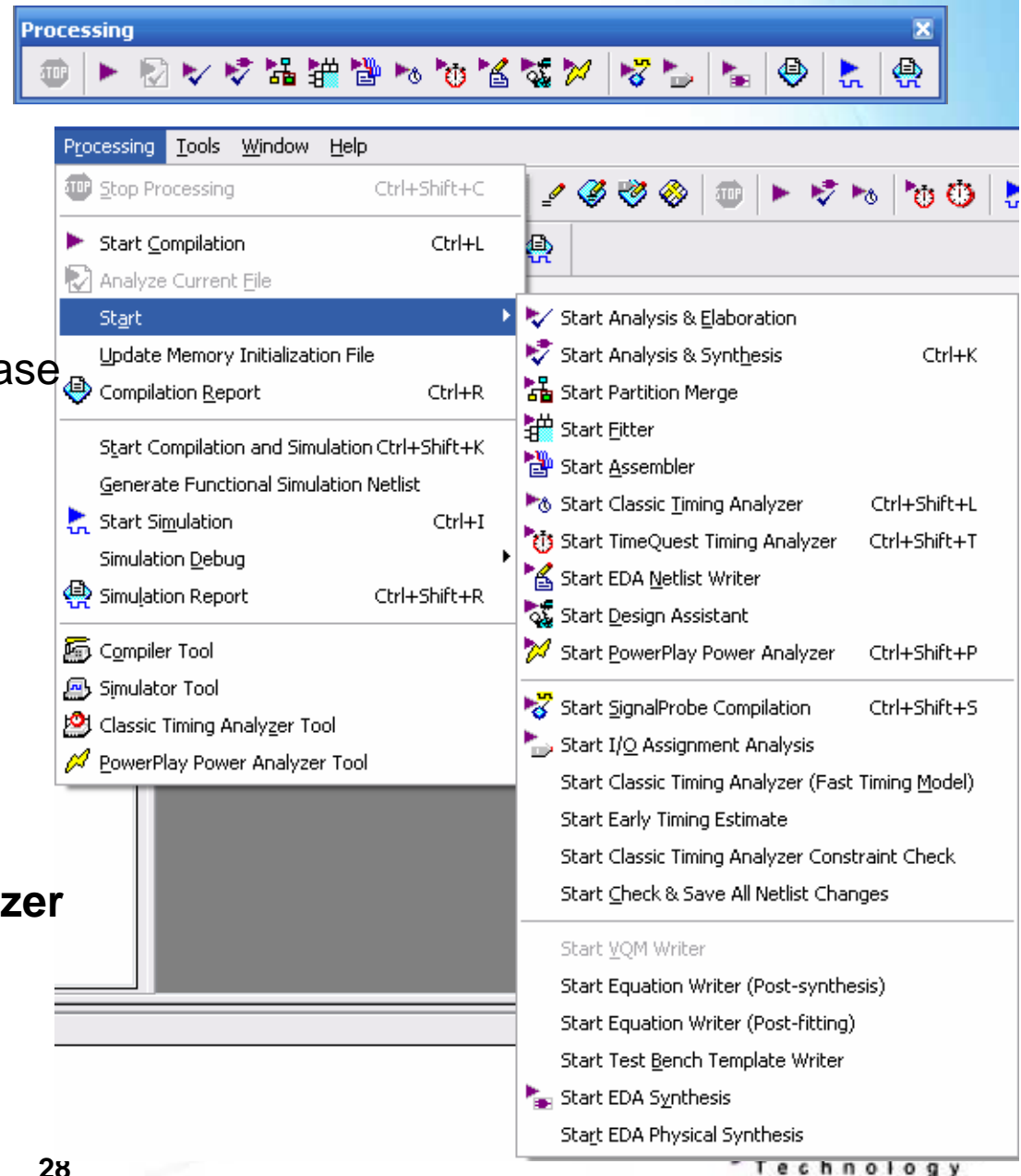
# Qusrtus II全编译流程



*\*This is the typical flow. Other module executables will be added if additional software features are enabled.*

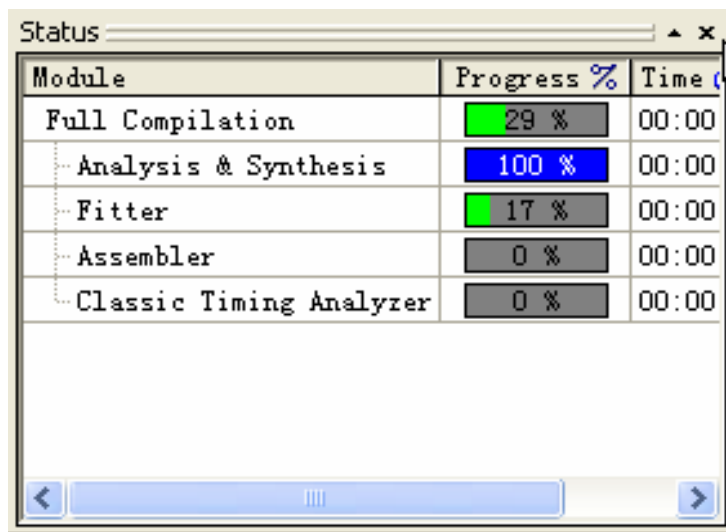
# Processing选项

- **Start Compilation**
  - Performs full compilation
- **Start Analysis & Elaboration**
  - Checks syntax & builds database only
  - Performs initial synthesis
- **Start Analysis & Synthesis**
  - Synthesizes & optimizes code
- **Start Fitter**
  - Places & routes design
  - Generates output netlists
- **Start Assembler**
  - Generate programming files
- **Start TimeQuest Timing Analyzer**
- **Start I/O Assignment Analysis**
- **Start Design Assistant**





# Status & Message Windows



The screenshot shows a 'Status' window with a table of modules and their progress. The 'Analysis & Synthesis' module is highlighted in blue and shows 100% progress. The 'Fitter' module shows 17% progress. The 'Full Compilation', 'Assembler', and 'Classic Timing Analyzer' modules show 0% progress. The time for all modules is 00:00.

Module	Progress %	Time
Full Compilation	29 %	00:00
Analysis & Synthesis	100 %	00:00
Fitter	17 %	00:00
Assembler	0 %	00:00
Classic Timing Analyzer	0 %	00:00

- **Analysis & Synthesis**完成综合的功能
- **Fitter**是对设计进行布局布线
- **Assembler**为编程或配置目标器件建立一个或多个编程文件，包括.sof和.pof。
- **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、保持时间、时钟至输出延时、引脚至引脚延时、最大时钟频率、延缓时间以及设计的其它时序特性。

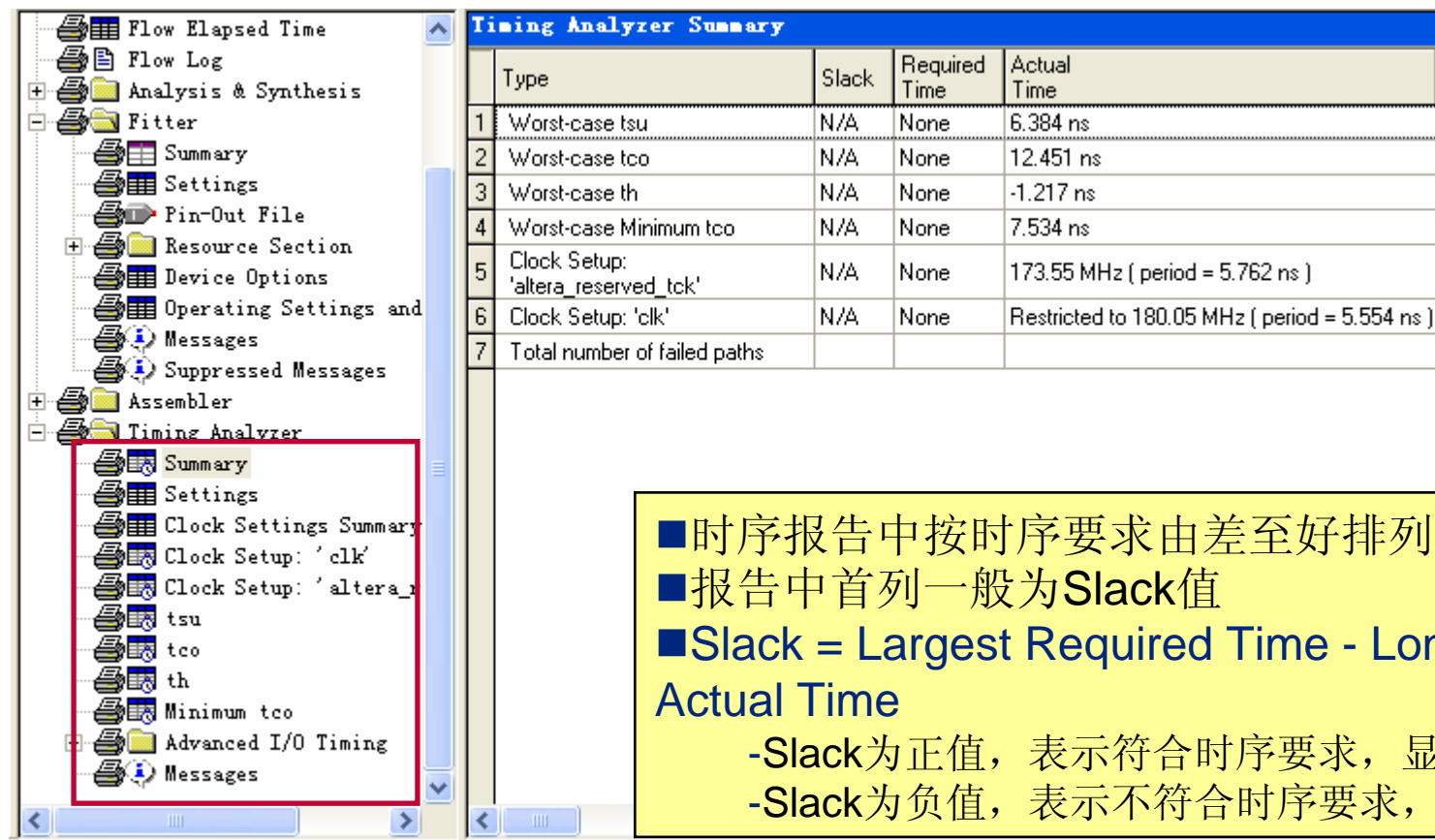
# 编译报告-资源报告

The screenshot shows the 'Fitter Summary' window in the Xilinx ISE software. The left pane displays a tree view of the compilation report, with the 'Resource Section' highlighted. The right pane shows the 'Input Pins' table, which lists the pins used in the design.

	Name	Pin #	I/O Bank	X coordinate	Y coordinate	Z coordinate	Combinational Fan-Out	Registered Fan-Out	Global	Input Register	Pc Hi
1	addr[0]	129	8	16	24	21	2	0	no	no	no
2	addr[1]	132	8	13	24	14	2	0	no	no	no
3	addr[2]	128	8	16	24	14	2	0	no	no	no
4	addr[3]	52	3	16	0	7	2	0	no	no	no
5	addr[4]	119	7	23	24	0	2	0	no	no	no
6	addr[5]	124	7	18	24	14	2	0	no	no	no
7	addr[6]	55	4	18	0	14	2	0	no	no	no
8	addr[7]	120	7	23	24	7	2	0	no	no	no
9	addr[8]	126	7	16	24	0	2	0	no	no	no
10	addr[9]	99	6	34	17	14	2	0	no	no	no
11	clk	22	1	0	11	0	1	0	yes	no	no

资源的详细信息

# 编译报告-时序报告



	Type	Slack	Required Time	Actual Time
1	Worst-case tsu	N/A	None	6.384 ns
2	Worst-case tco	N/A	None	12.451 ns
3	Worst-case th	N/A	None	-1.217 ns
4	Worst-case Minimum tco	N/A	None	7.534 ns
5	Clock Setup: 'altera_reserved_tck'	N/A	None	173.55 MHz ( period = 5.762 ns )
6	Clock Setup: 'clk'	N/A	None	Restricted to 180.05 MHz ( period = 5.554 ns )
7	Total number of failed paths			

- 时序报告中按时序要求由差至好排列
- 报告中首列一般为Slack值
- $\text{Slack} = \text{Largest Required Time} - \text{Longest Actual Time}$ 
  - Slack为正值，表示符合时序要求，显示为黑色
  - Slack为负值，表示不符合时序要求，显示为红色

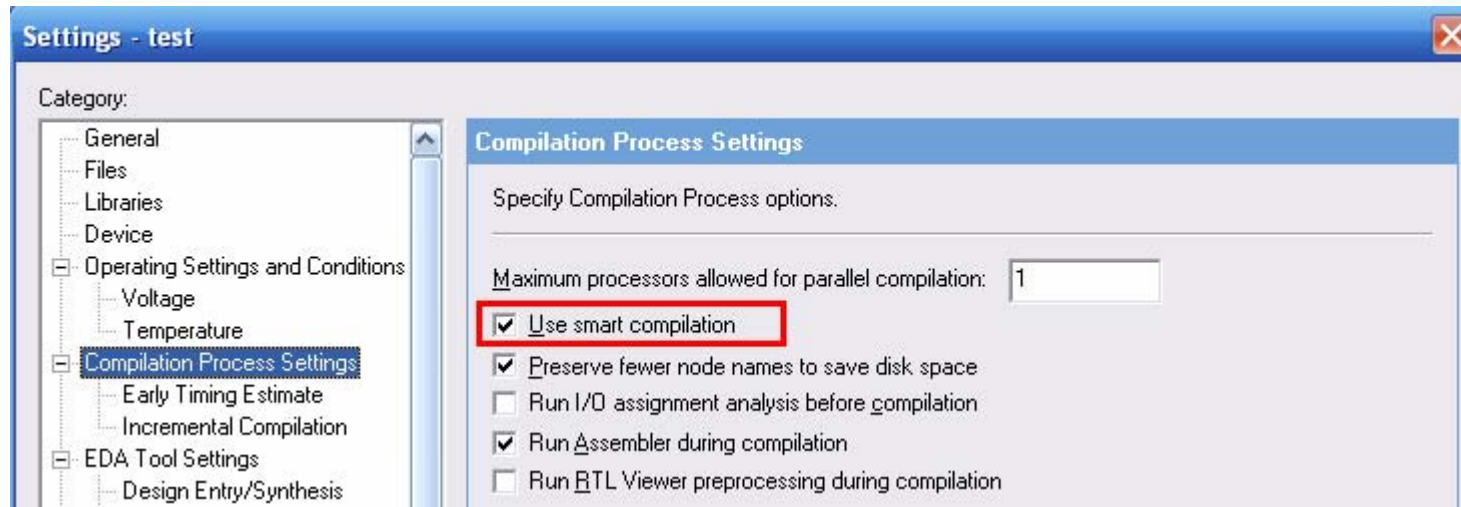
**ALTERA.**

# Quartus II 软件使用教程

综合



# 与Synthesis相关的设置(1)

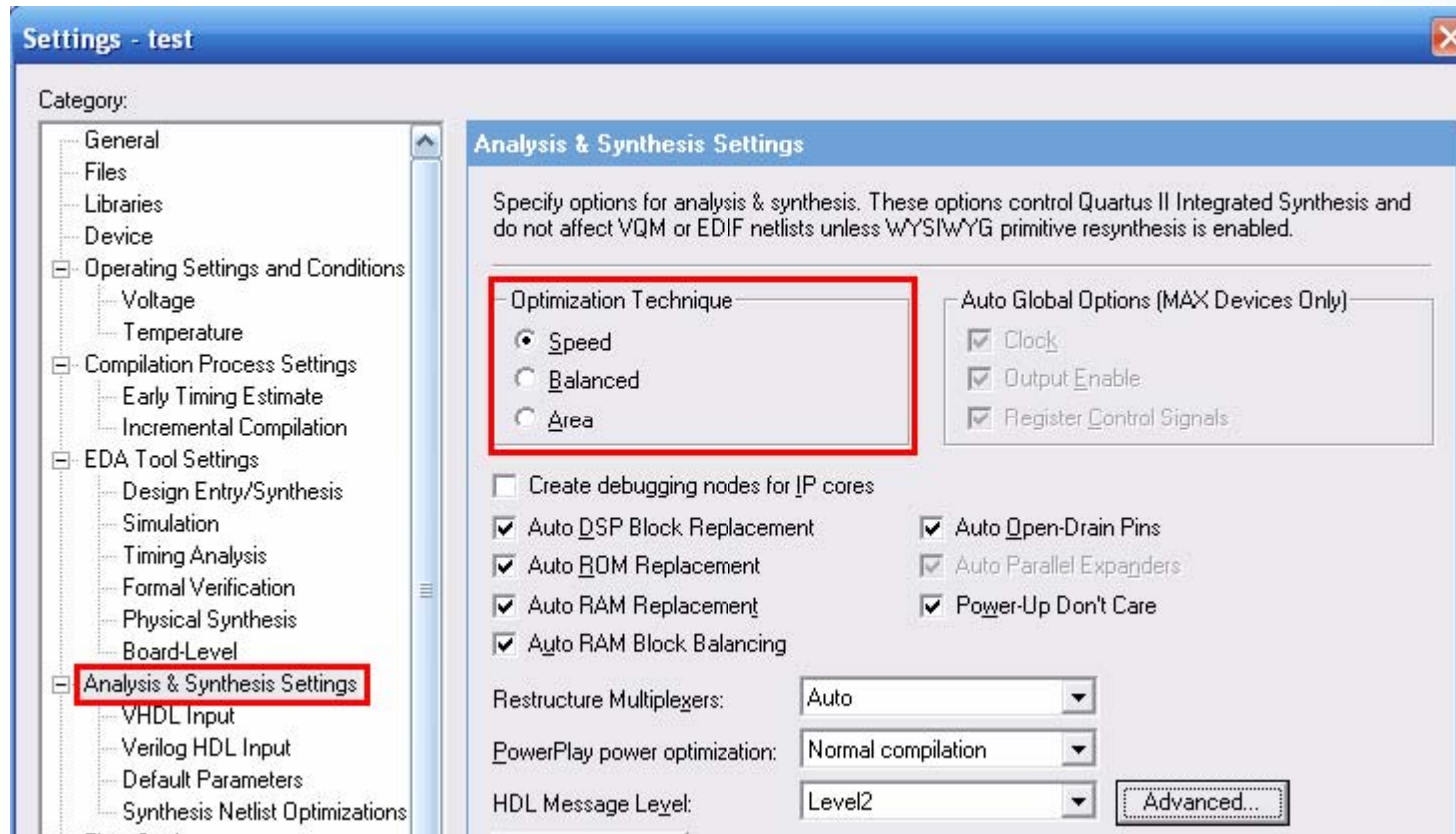


■ 如果选中，可以减少工程的编译时间。比如在工程中没有改动源代码而只是对约束进行了修改，使用了**Smart Compilation**选项，则进行全编译（▶ 按钮）过程中，软件会自动跳过“**Analysis & Synthesis**”步骤。

■ **QII**默认设置为关闭，建议打开。



## 与Synthesis相关的设置(2)

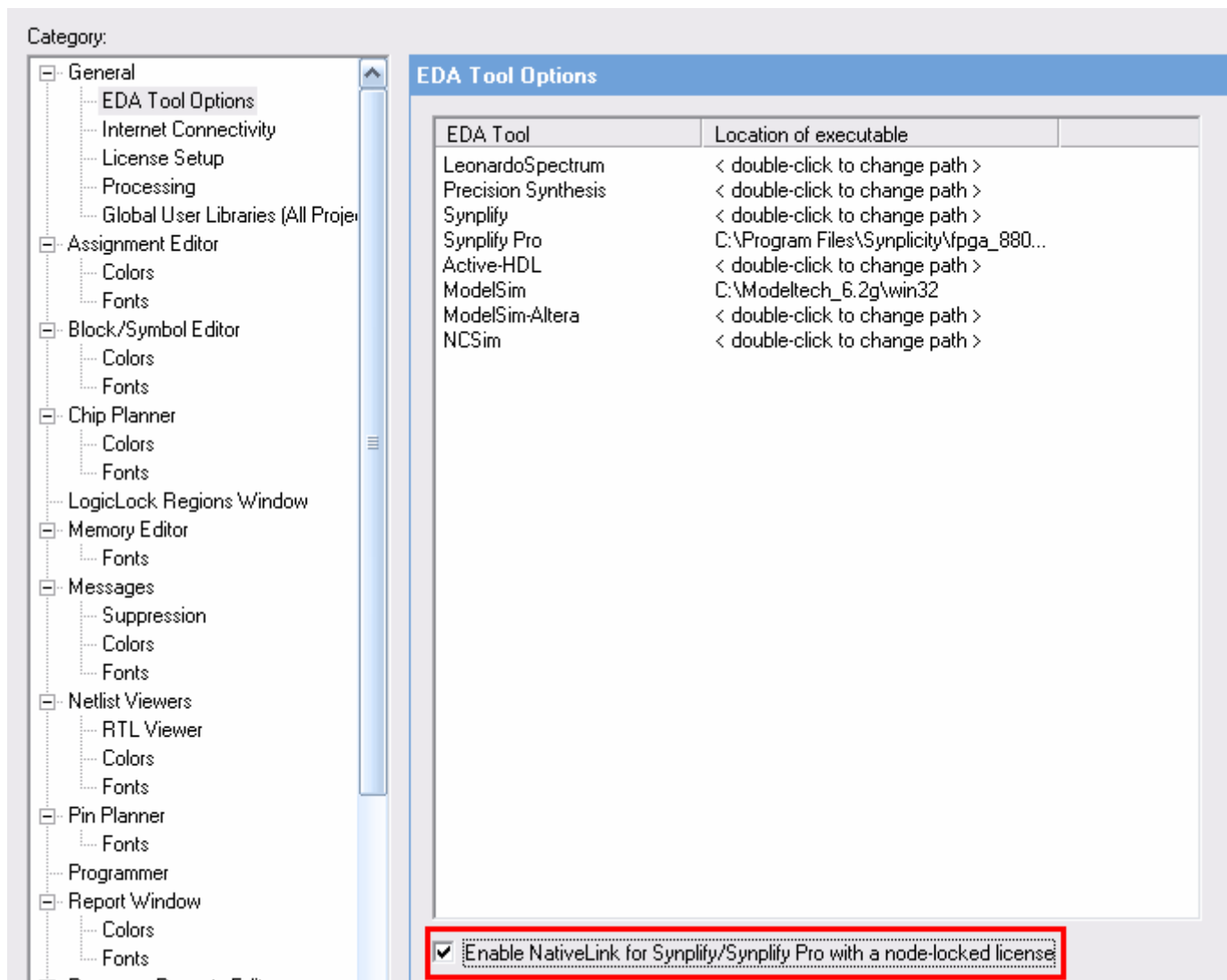


- 优化目标：速度，面积和平衡，默认设置为平衡
- 一般是优化工程设计的第一步





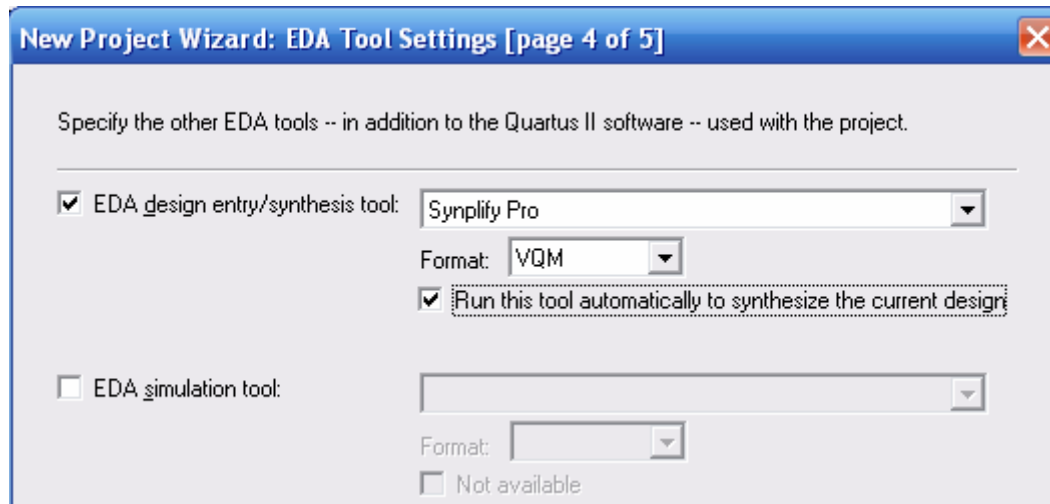
# 第三方综合器Synplify Pro嵌入(1)



## Step1:

Tools菜单Options->General->EDA Tool Options指定SynplifyPro的安装路径，选中“Enable NativeLink for Synplify/Synplify Pro with a node-locked license”。如果不选，QII是无法直接调用Synplify进行综合的。

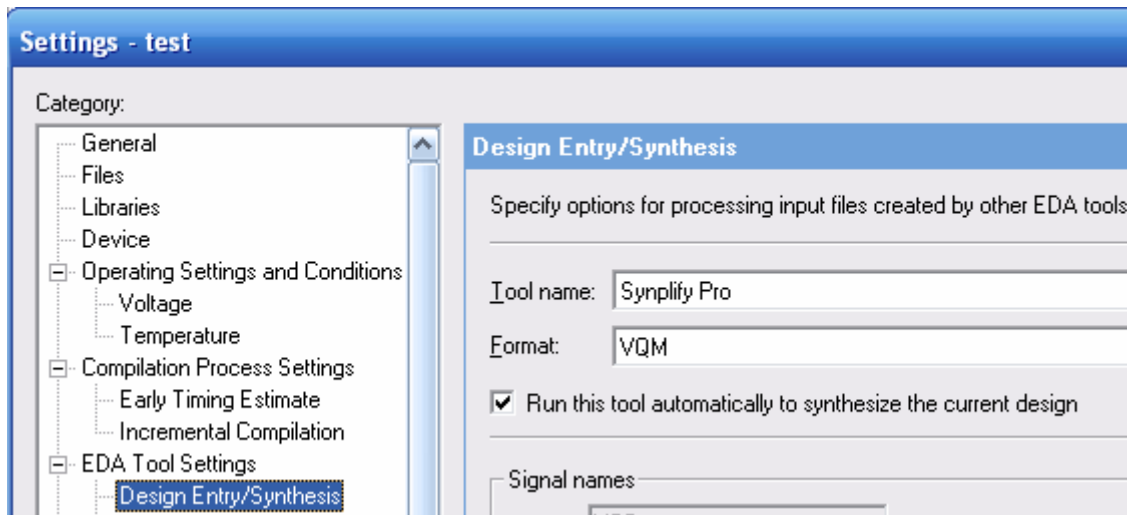
## 第三方综合器Synplify Pro嵌入(2)



### Step2:

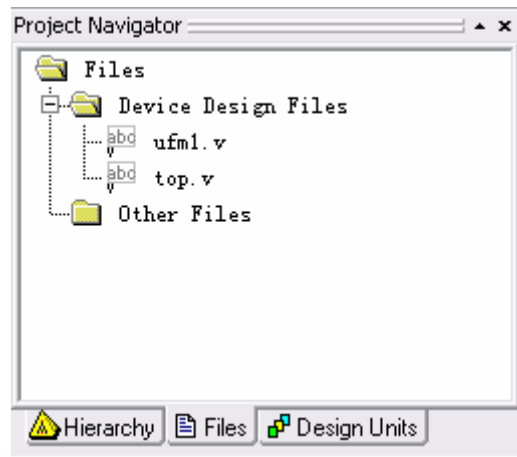
-如果是新建工程，需要在 EDA Tool Settings中设置，并选中“Run this tool automatically to synthesize the current design”

-如果是现有工程，Settings->EDA Tool Settings同样设置



## 第三方综合器Synplify Pro嵌入(3)

- 如果使用IPcore，QII导航界面可以看到的是IPcore生成的.v文件，这样是无法进行直接调用SynplifyPro的，软件会报错
- 只需要将ufm1.v文件删除，用ufm1\_bb.v文件替代，就可以直接调用Synplify Pro
- QII会自动在工程目录下生成synplify\_xxx\_work目录，目录下有供Synplify Pro可以直接打开的工程文件



ALTERA.

# Quartus II 软件使用教程

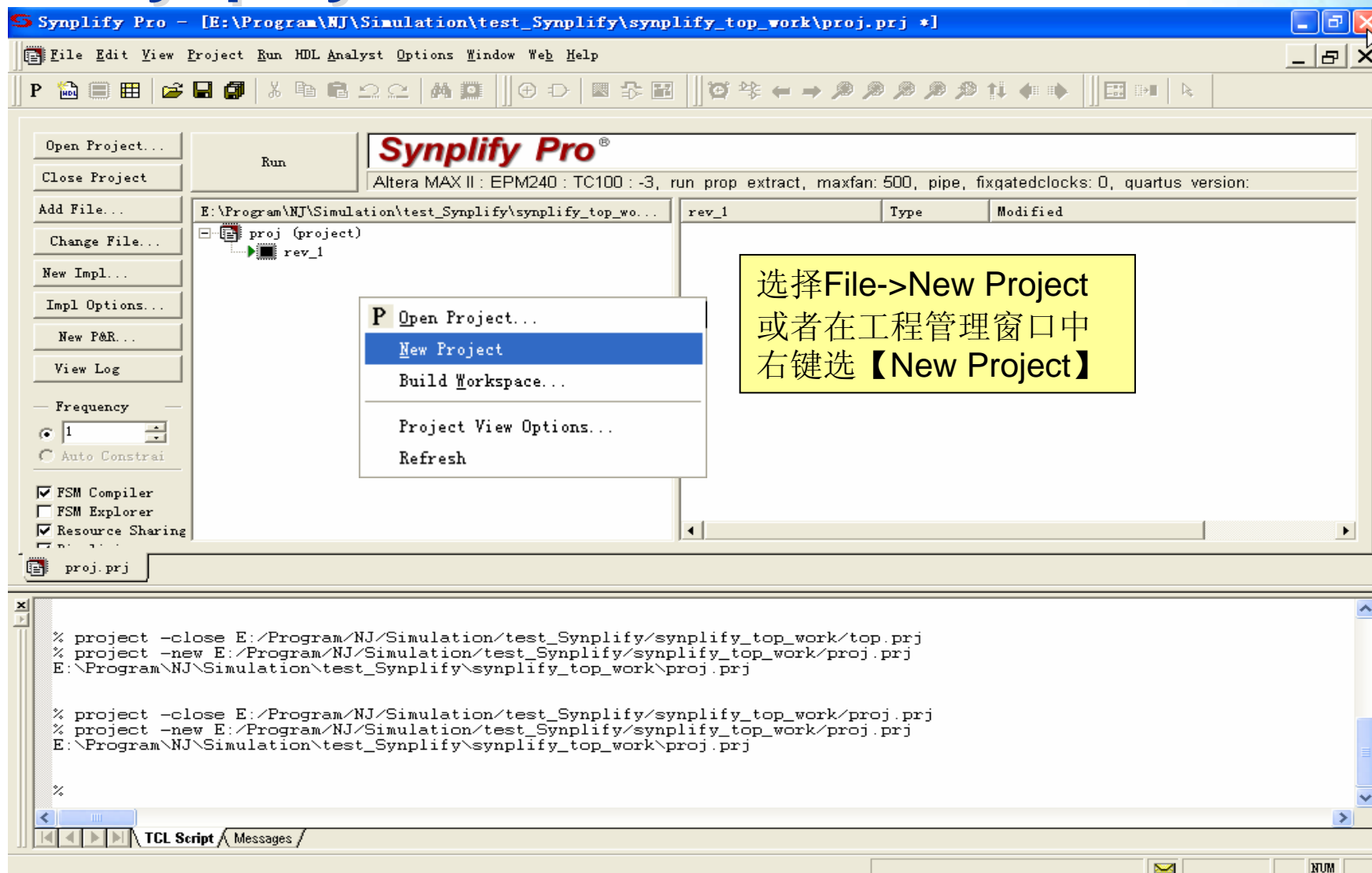
使用Synplify Pro做综合



## 使用Synplify Pro做综合

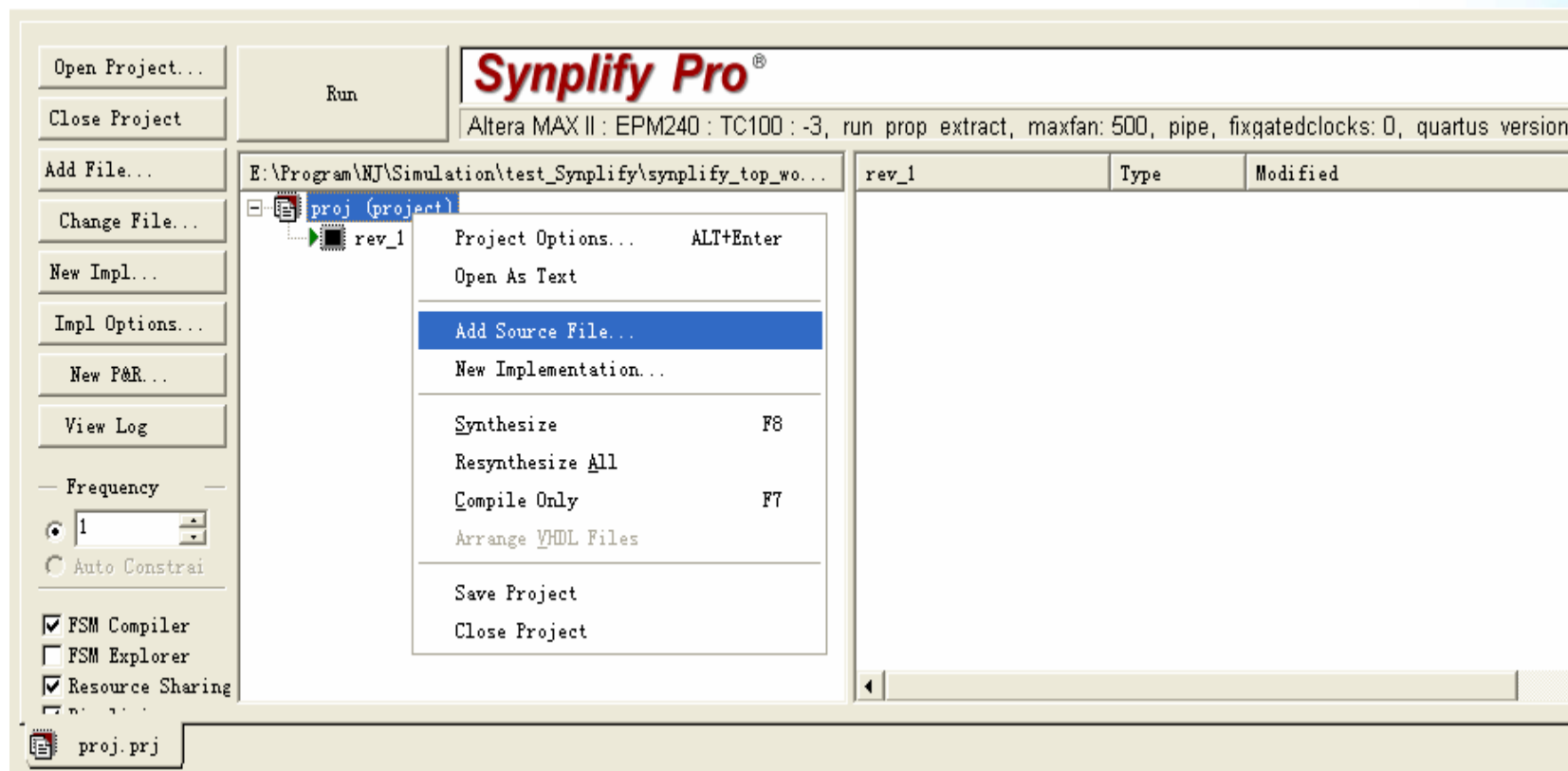
- 通常我们综合时，使用的是Quartus II自带的综合工具，实际工作中，许多设计人员都习惯于使用专业综合工具Synplify Pro。
- 正常情况下，正版的Quartus II和Synplify Pro可以实现无缝链接，过程同ISE差不多。但是大多数用户使用时，Quartus II直接调用Synplify Pro往往是有问题的，因此我们最好将两者分离开来操作。

# 在Synplify Pro下建立工程





# 添加源文件



## 添加源文件时注意:

The screenshot shows the Synplify Pro interface. On the left is a sidebar with buttons: Open Project..., Close Project, Add File..., Change File..., New Impl..., Impl Options..., New P&R..., View Log, Frequency, Auto Constrai, Physical Synthes, FSM Compiler, and FSM Explorer. The main window displays a project structure tree for 'top (project)' under the path 'E:\Program\NJ\Simulation\test\_Synplify\synplify\_top\_work'. The tree includes a 'Verilog' folder with files 'main\_pll\_bb.v', 'mux\_2to1\_bb.v', 'TDM\_mult\_bb.v', and 'top.v', each showing warning and note counts. Below it is a 'Constraint' folder containing 'synplify\_top\_work'. To the right, a file list table shows various files and their properties.

如果调用的是Quartus II 软件中生成的IP模块, 则只需要加入\_bb.v的黑盒文件 (bb.v文件是使用MegaWizard向导产生的)

顶层文件要放在文件结构树的最下面或者是最后一个加到工程里面。

File Name	Type	Time	Date
top.tap	Timing A...	15:12:16	27-Mar-2008
top.tcl	tcl script	15:12:16	27-Mar-2008
top.map	Mapper A...	15:12:16	27-Mar-2008
top.vm	file	15:12:16	27-Mar-2008
top.vqm	verilog	15:12:16	27-Mar-2008
top.xrf	file	15:12:16	27-Mar-2008
rpt_top_areasrr	Hierarch...	15:12:16	27-Mar-2008
top_cons.tcl	tcl script	15:12:16	27-Mar-2008
top.fit.rpt	file	15:27:50	27-Mar-2008
top.pin	Altera P...	15:27:50	27-Mar-2008

# 设置工程属性

Open Project... Close Project Add File... Change File... New Impl... Impl Options... New P&R... View Log

Frequency: 1 Auto Cons

Physical Synthes FSM Compiler FSM Explorer

Run

**Done (warnings)**

Altera CYCLONE II : EP2C5 : FC256 : -6, run prop extract, maxfan: 30, pipe, fixgatedclocks: 3, quartus version: 5.1

E:\Program\NJ\Simulation\test\_Synplify\top (project)

- Verilog
  - main\_pll\_bb.v
  - mux\_2to1\_bb.v
  - TDM\_mult\_bb.v
  - top.v -> WARN
- Constraint
- synplify\_top\_work**

**Options for implementation: top : synplify\_top\_work**

Timing Report | Place and Route | Verilog | Netlist Restructure

Device | Options | Constraints | Implementation Results

Implementation: synplify\_top\_work

Results Directory: E:\Program\NJ\Simulation\test\_Synplify\synplify\_top\_work

Result File Name: top.vqm

Optional Output Files

- ☒ Write Mapped Verilog Ne
- ☐ Write Mapped VHDL Netli
- ☒ Write Vendor Constraint
- ☒ Write Verification Interface Format

Quartus: Quartus II 5.1

Quartus II 4.0  
Quartus II 4.1  
Quartus II 4.2  
Quartus II 5.0  
Quartus II 5.1

Implementations: synplify\_top\_work

Synplicity

取消 帮助

双击一个实现

确认生成的VQM网表

设置对应的Quartus II版本

# 综合设计

点击Run，对源文件进行综合

Done (warnings)

Altera CYCLONE II: EP2C5: FC256: -6, run prop extract, maxfan: 30, pipe, fixgatedclocks: 3, quartus version: 5.1

Run

Files

synplify\_top\_work

File	Type	Modified
TDM_mult_bb.v	file	09:53:23 25-Apr-2008
TDM_mult_bb.vqm	verilog	09:53:23 25-Apr-2008
top.prd	file	09:54:34 25-Apr-2008
top.prj	Project	09:54:34 25-Apr-2008
top.sap	Annotate...	
top.fse	Synplific...	
top.map	Mapper A...	
top.srd	Synplific...	
top.tap	Timing A...	
top.tcl	tcl script	
top.srm	Gate Net...	09:54:36 25-Apr-2008
top.v	file	09:54:36 25-Apr-2008
top.vqm	verilog	09:54:36 25-Apr-2008
top.xrr	file	09:54:36 25-Apr-2008
AutoConstraint...	constraint	09:54:36 25-Apr-2008
top.cons.tcl	tcl script	09:54:36 25-Apr-2008

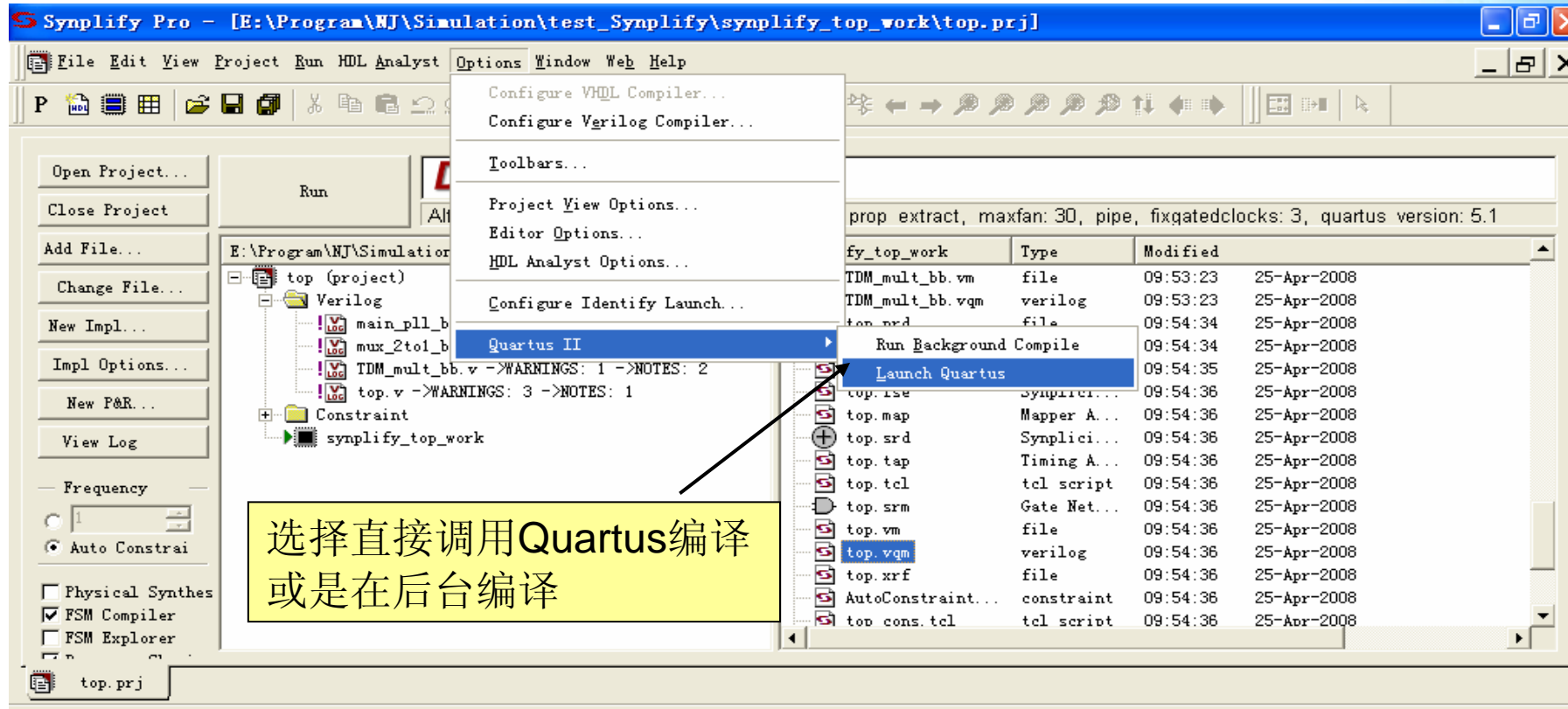
Messages

Type	ID	Message	Source Location	Log Location	Time	Report
Warning	BN191	Writing property annotation file E:\Program\N...	-	top.srr (103)	09:54:...	Mapper Report
Warning	BN225	Writing default property annotation file E:\P...	-	top.srr (106)	09:54:...	Mapper Report
Warning		Gated clock conversion enabled	-	top.srr (51)	13:50:...	Mapper Report
Warning	FA174	The following device usage report estimates p...	-	top.srr (536)	13:50:...	Resource Utilization
Warning	MF249	Running in 32-bit mode.	-	top.srr (50)	13:50:...	Mapper Report
Warning	MT195	This timing report estimates place and route ...	-	top.srr (141)	13:50:...	Timing Report
Warning	MT197	Clock constraints cover only FF-to-FF paths a...	-	top.srr (143)	13:50:...	Timing Report
Warning	MT206	Autoconstrain Mode is ON	-	top.srr (54)	13:50:...	Mapper Report

信息栏将显示警告或者错误

文件列表可以看到生成的vqm网表

# 调用Quartus II编译工程



需要注意的是，在Synplify Pro工程下用到的IP模块调用的是\_bb.v的文件，但进入到Quartus II中则需要的是由MegaWizard向导生成的.v文件，所以最好让Quartus II工程文件与IP生成的.v文件放在同一个目录下。

**ALTERA.**

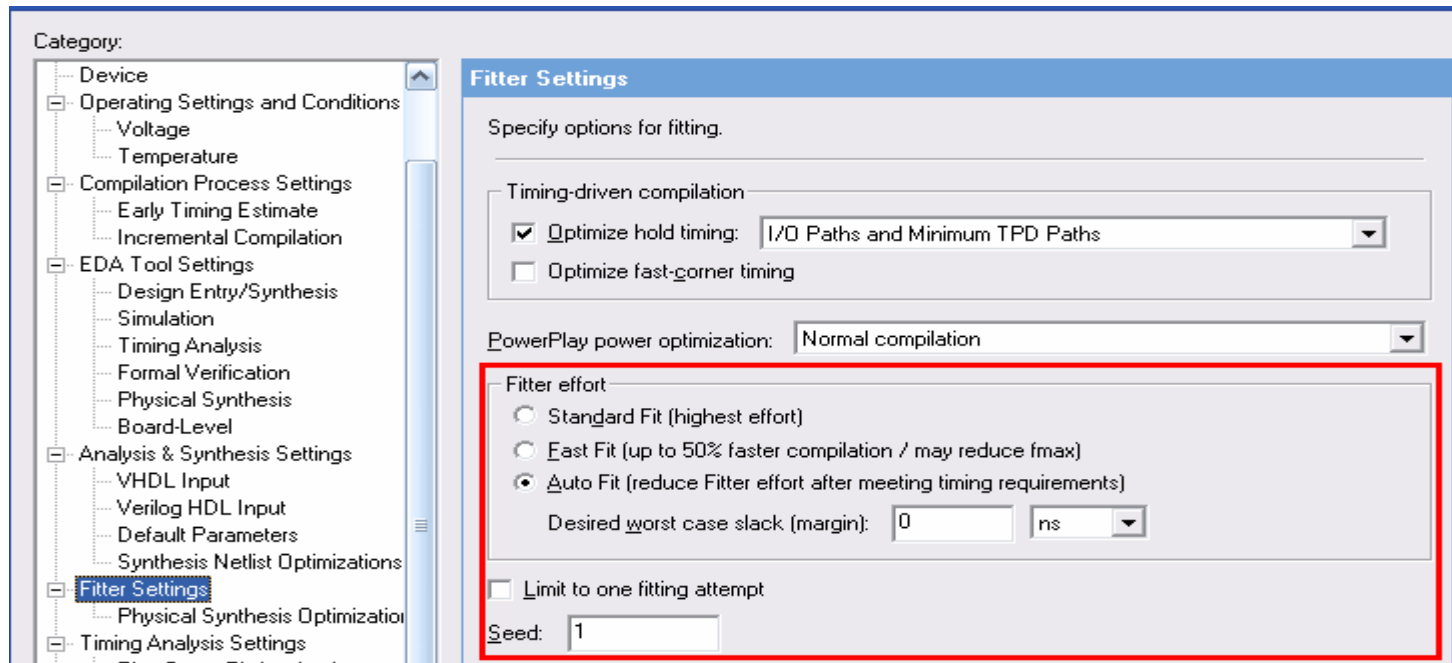
# Quartus II 软件使用教程

布局布线





# 与Fitter相关的设置(1)



## -Standard Fit

编译效果最好，时间最长

## -Fast Fit

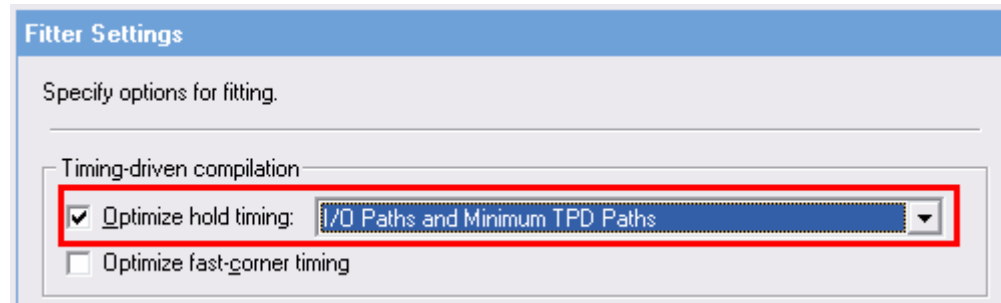
编译时间减少50%，牺牲Fmax作代价

## -Auto Fit

一旦满足时序要求，QII会停止优化，从而节省编译时间

**-One fitting attempt**（不同的种子导致编译结果小幅度变动，波动范围在±5%）

## 与Fitter相关的设置(2)



### ■默认情况下为“IO to register and min Tco”

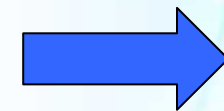
-表示以IO到寄存器的Th约束、从寄存器到IO的最小Tco约束和从IO或寄存器到IO或寄存器的最小TPD约束为优化目标

### ■设置为“All Paths”

-除了IO路径和最小TPD路径为优化目标外，增加了寄存器到寄存器的时序约束优化

-尽可能的使用同步设计以避免时序问题

### ■对于时序报告中Hold 时间的不满足，可以尝试设置“All Paths”



**ALTERA.**

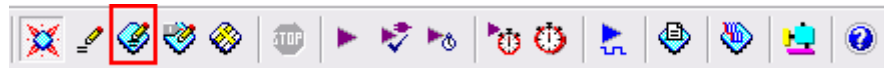
# Quartus II 软件使用教程

*Assignment Editor*

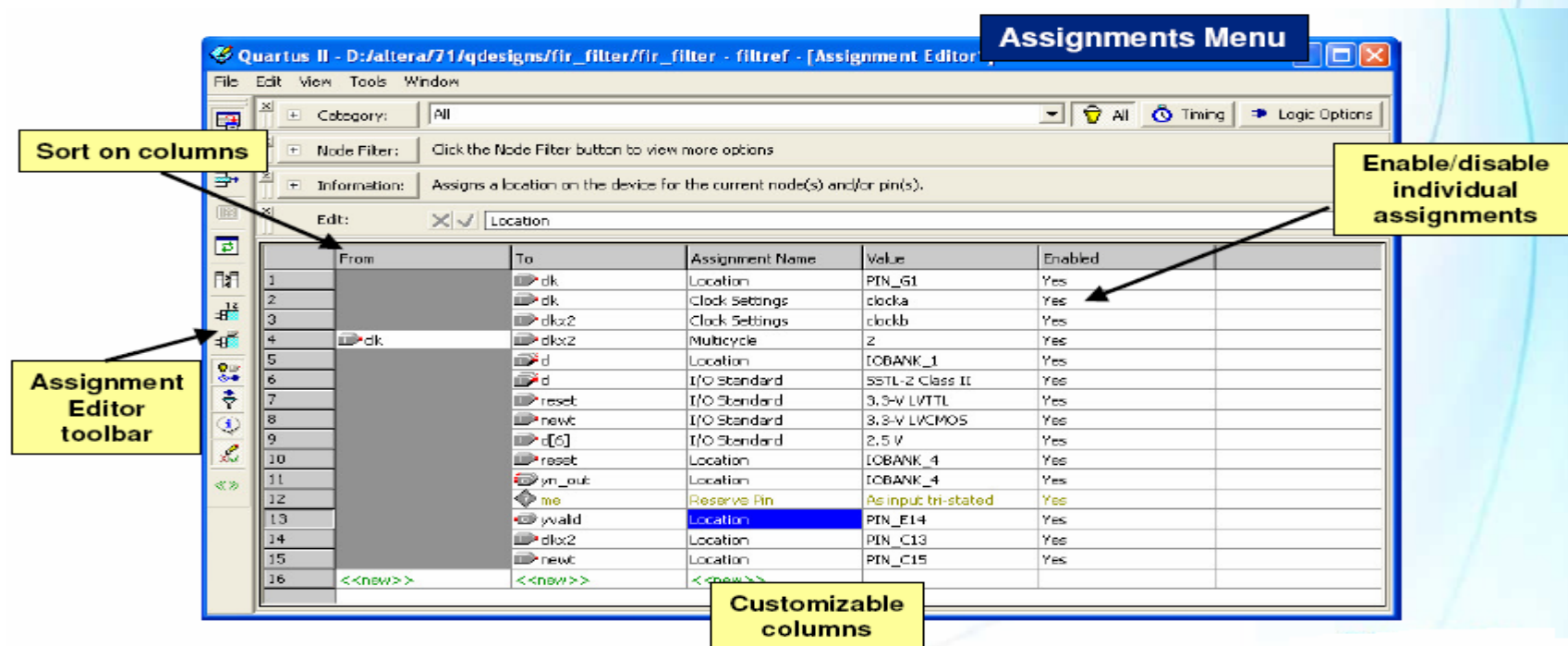


# Assignment Editor (AE)

- 选择菜单Assignments→Assignment Editor或者直接点击按钮



- 通过AE生成的各种约束都会保存在.QSF文件中

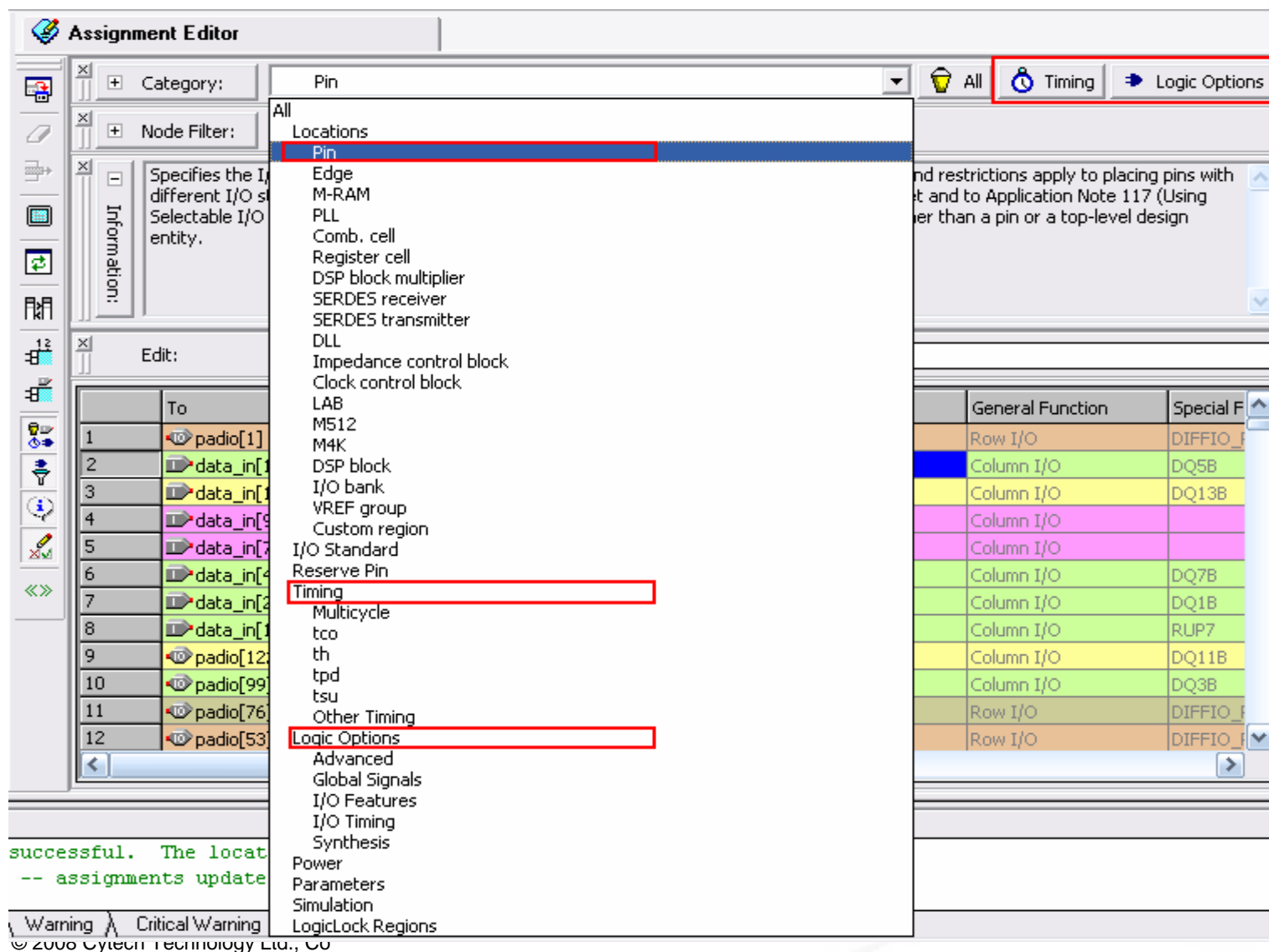


The screenshot shows the 'Assignment Editor' window in Quartus II. The window title is 'Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filterf - [Assignment Editor]'. The 'Assignments Menu' is visible at the top right. The window contains a table of assignments with columns: From, To, Assignment Name, Value, and Enabled. The table is sorted by the 'From' column. Annotations point to various features:

- Sort on columns**: Points to the 'From' column header.
- Assignment Editor toolbar**: Points to the toolbar on the left side of the window.
- Enable/disable individual assignments**: Points to the 'Enabled' column header.
- Customizable columns**: Points to the bottom of the table where new columns can be added.

	From	To	Assignment Name	Value	Enabled
1		dk	Location	PIN_61	Yes
2		dk	Clock Settings	clocka	Yes
3		dkx2	Clock Settings	clockb	Yes
4	dk	dkx2	Multicycle	2	Yes
5		d	Location	IOBANK_1	Yes
6		d	I/O Standard	55TL-2 Class II	Yes
7		reset	I/O Standard	3.3-V LVTTTL	Yes
8		newtc	I/O Standard	3.3-V LVC MOS	Yes
9		d[6]	I/O Standard	2.5 V	Yes
10		reset	Location	IOBANK_4	Yes
11		ym_out	Location	IOBANK_4	Yes
12	me		Reserve Pin	As input tri-stated	Yes
13		yvalid	Location	PIN_E14	Yes
14		dkx2	Location	PIN_C13	Yes
15		newtc	Location	PIN_C15	Yes
16	<<new>>	<<new>>	<<new>>		

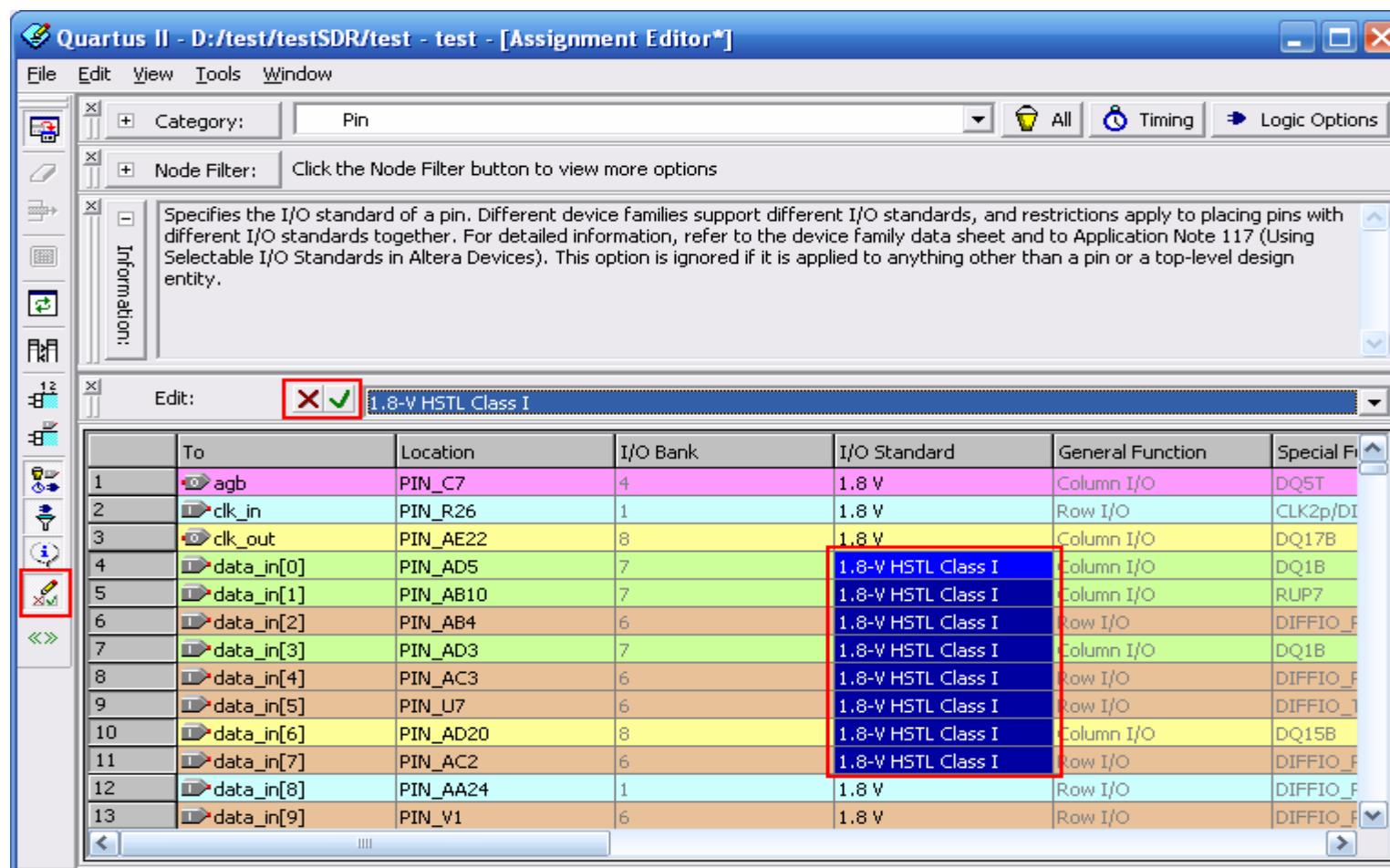
# Using AE



Category下拉选择，一般常用的是Pin，Timing和Logic Options，可以分别独立设置

# 编辑多个约束

- Use Edit bar, auto-fill, copy & paste



鼠标选择多个需改动的约束，在Edit Bar中下拉或直接复制粘贴选择新的约束。当Edit Bar中的“√”变成灰色表示修改成功



**ALTERA.**

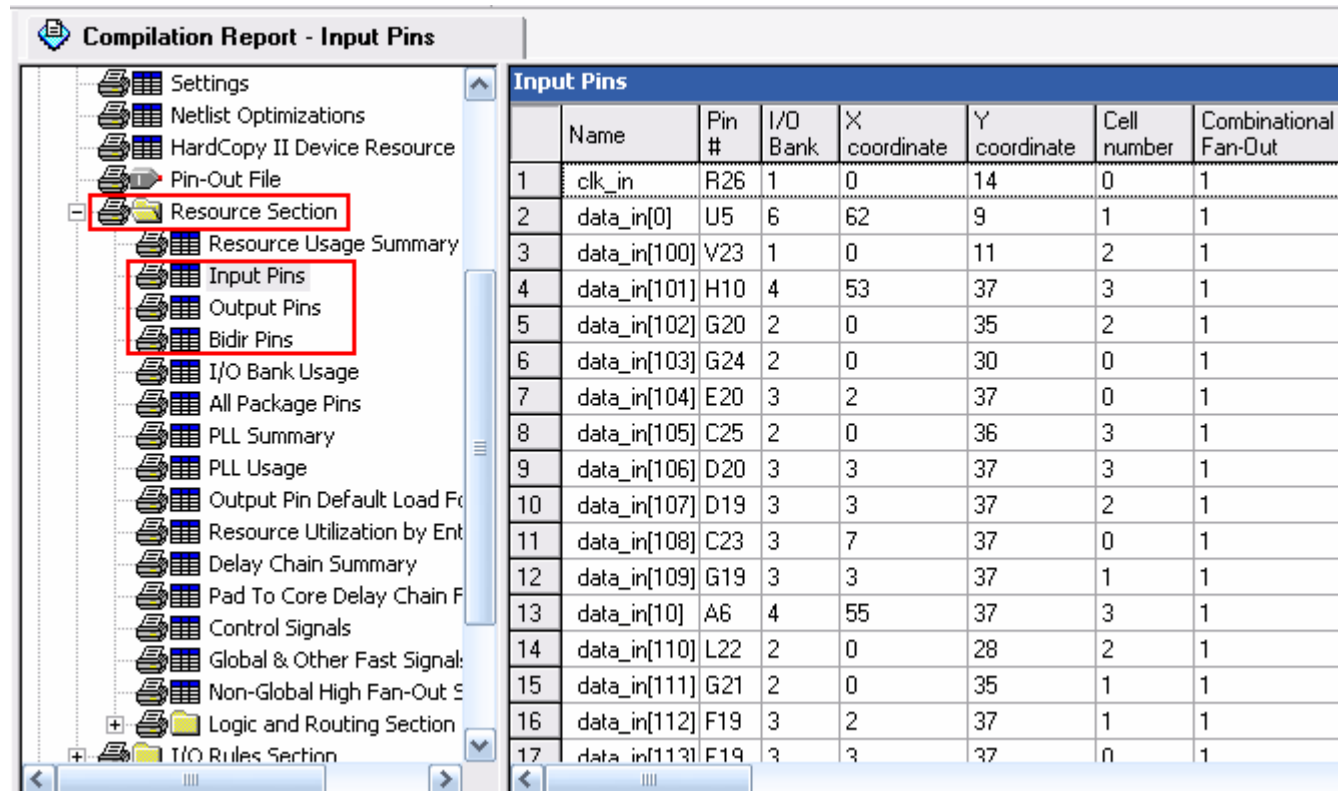
# Quartus II 软件使用教程

管脚分配



# Pin的约束(1)

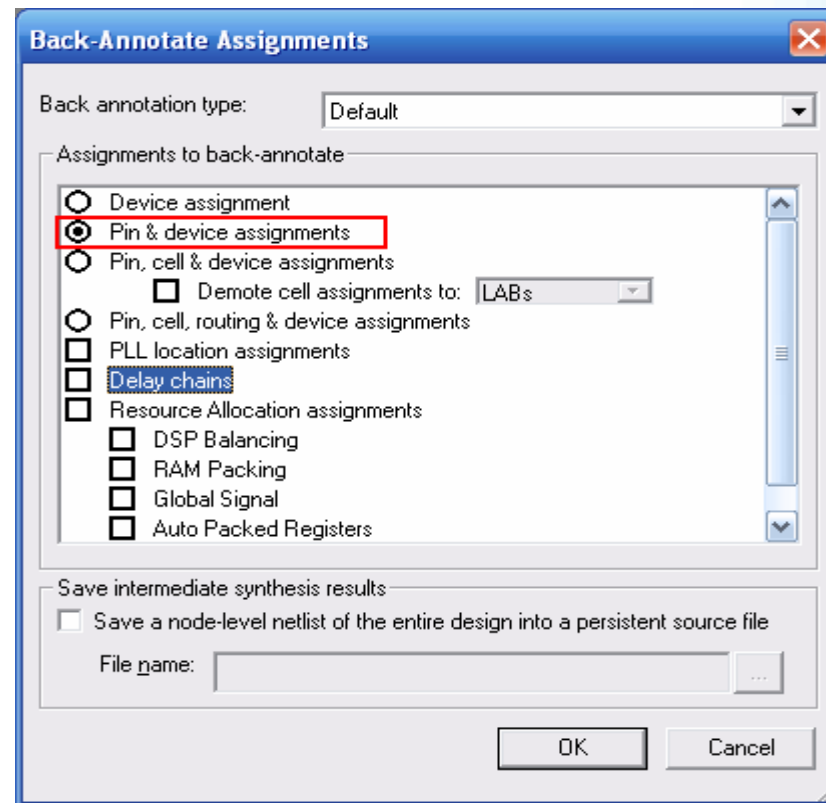
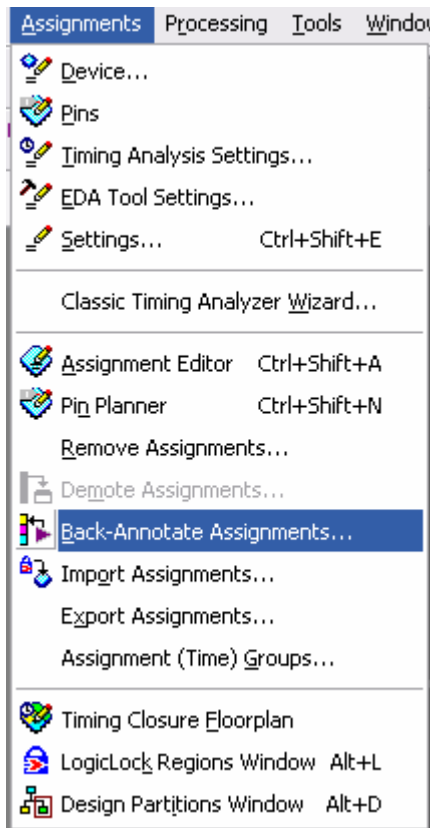
- 一个新建Project是没有约束的，AE显示为空
- 如果在没有约束的情况下直接编译Project，QII会自动分配引脚，显示在编译报告中(Compilation Report->Fitter->Resource Section->Input Pins, Output Pins or Bidir Pins)



Input Pins							
	Name	Pin #	I/O Bank	X coordinate	Y coordinate	Cell number	Combinational Fan-Out
1	clk_in	R26	1	0	14	0	1
2	data_in[0]	U5	6	62	9	1	1
3	data_in[100]	V23	1	0	11	2	1
4	data_in[101]	H10	4	53	37	3	1
5	data_in[102]	G20	2	0	35	2	1
6	data_in[103]	G24	2	0	30	0	1
7	data_in[104]	E20	3	2	37	0	1
8	data_in[105]	C25	2	0	36	3	1
9	data_in[106]	D20	3	3	37	3	1
10	data_in[107]	D19	3	3	37	2	1
11	data_in[108]	C23	3	7	37	0	1
12	data_in[109]	G19	3	3	37	1	1
13	data_in[10]	A6	4	55	37	3	1
14	data_in[110]	L22	2	0	28	2	1
15	data_in[111]	G21	2	0	35	1	1
16	data_in[112]	F19	3	2	37	1	1
17	data_in[113]	F19	3	3	37	0	1

## Pin的约束(2)

- 如果需要手工输入Pin约束，除直接编辑QSF文件之外，还有两种相对来说比较方便的方法
- 方法一：反标约束，让QII自动生成约束



反标时只选择“Pin & device assignments”，AE立刻显示反标过的Pin约束

# Pin的约束(3)

## ■ 方法二： Pin Planner图形化操作

直接拖动引脚到约束的位置上

Node Name	Direction	Location
clk_in	Input	
clk_out	Output	
data_in[143]	Input	PIN_T3
data_in[142]	Input	
data_in[141]	Input	
data_in[140]		
data_in[139]		
data_in[138]		
data_in[137]		
data_in[136]		
data_in[135]		
data_in[134]		

## Pin的约束(4)

- 需要注意的是，在反标FPGA时会多出两三个信号，这与FPGA的配置模式有关。为避免麻烦，建议直接删除

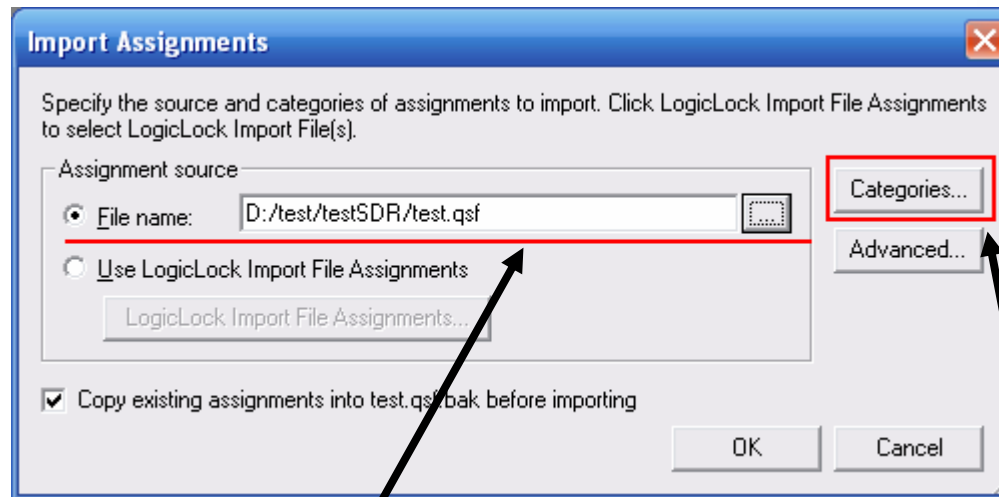
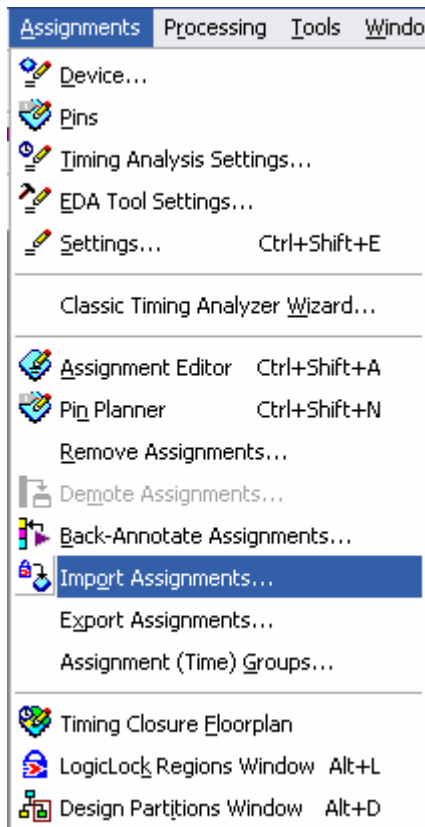
	From	To ▲	Assignment Name
292		reset	Location
293		?	Partition Hierarchy
294		~ASD0~	Location
295		~DATA0~	Location
296		~nCS0~	Location
297	<<new>>	<<new>>	<<new>>

在Category下拉选择Pin，直接修改Location栏下的Pin值就可以了。比如新约束为A22，点击“PIN\_A18”，手工敲入“A22”即可

Category:		Pin		
Node Filter:		Click the Node Filter button to view more options		
Information:		Assigns a location on the device for the current node(s) and/or pin(s).		
Edit:		PIN_A18		
	To	Location	I/O Bank	I/O Standard
1	agb	PIN_A18	3	1.8 V
2	clk_in	PIN_R26	1	1.8 V
3	clk_out	PIN_B18	3	1.8 V
4	data_in[0]	PIN_U5	6	1.8 V
5	data_in[1]	PIN_B3	4	1.8 V
6	data_in[2]	PIN_Y2	6	1.8 V
7	data_in[3]	PIN_C2	5	1.8 V
8	data_in[4]	PIN_D4	4	1.8 V
9	data_in[5]	PIN_D3	5	1.8 V
10	data_in[6]	PIN_F18	3	1.8 V
11	data_in[7]	PIN_K8	5	1.8 V

## Pin的约束(5)

- 如果先前已有.QSF文件，可以直接导入新的Project中，AE立刻显示新的约束



选择需导入的文件

选择需导入的约束类型



**ALTERA.**

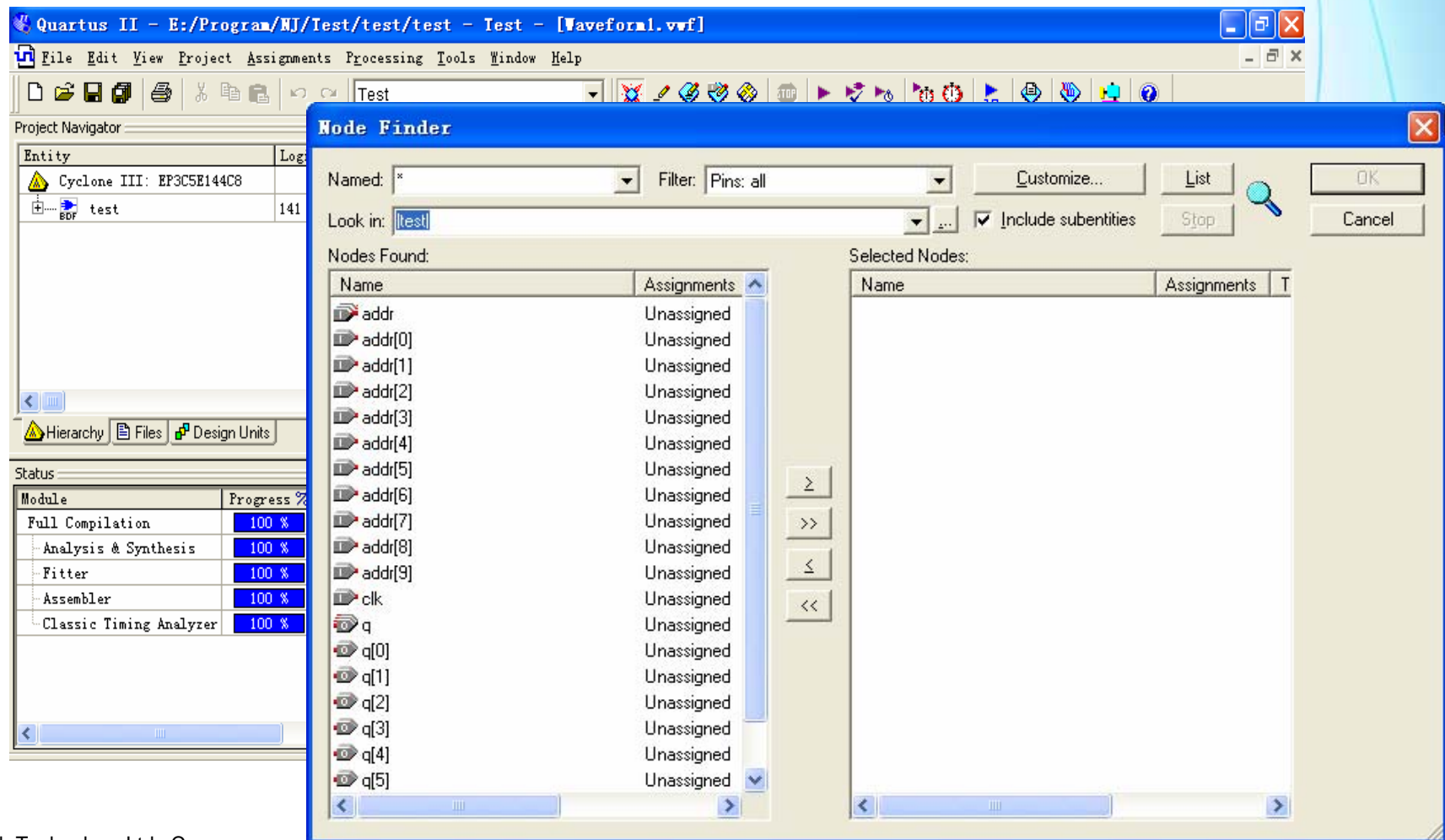
# Quartus II 软件使用教程

仿真

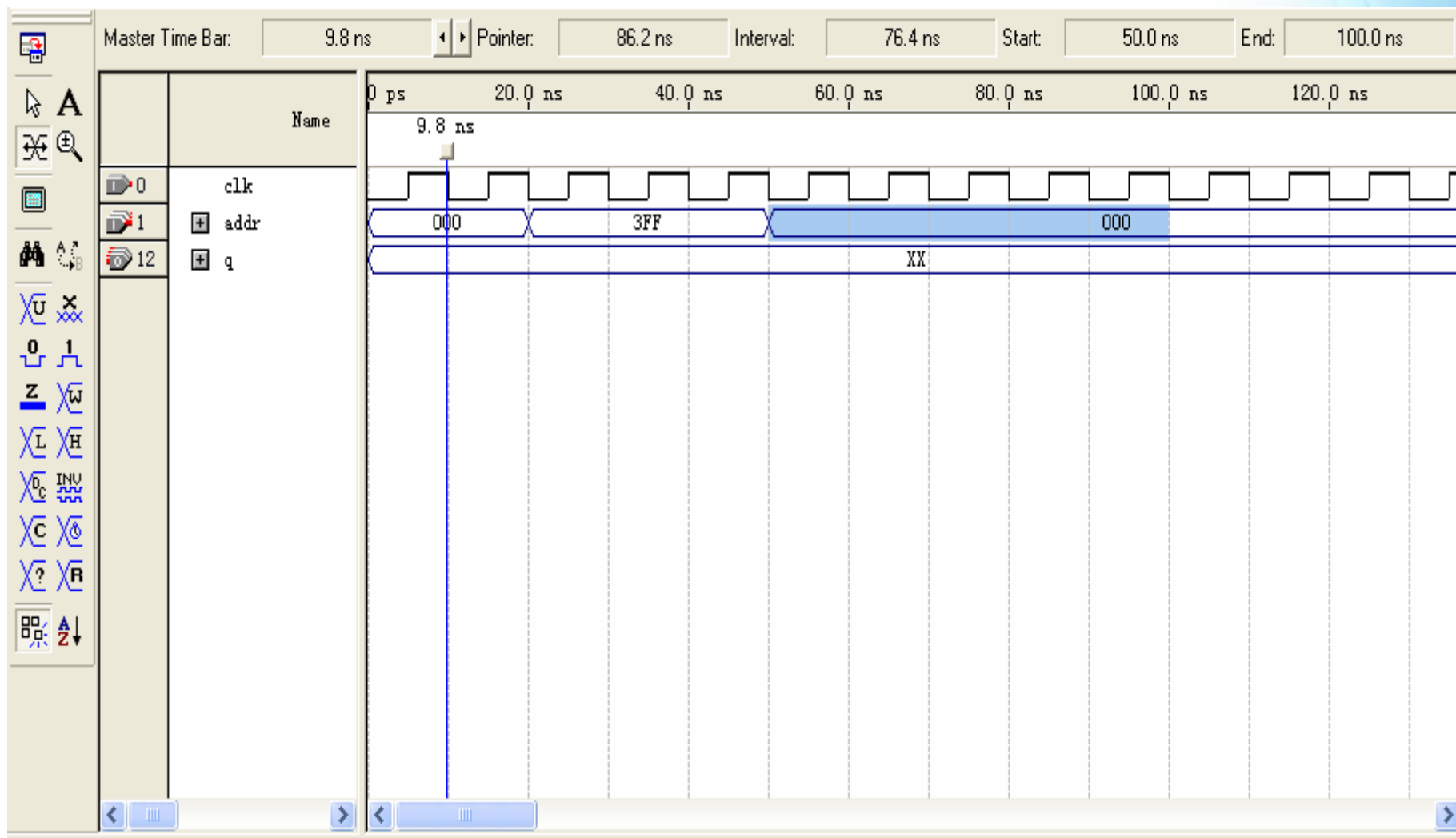


# 建立仿真文件

选择File菜单下的New->Other Files->Vector Waveform File



# 编辑波形



© 2008 Cytech Technology  
 : Minimum tco from clock "clk" to destination pin "q[6]  
 : Quartus II Classic Timing Analyzer was successful. 0

Arbitrary Value...  
 Random Values...

Ctrl+Alt+B  
 Ctrl+Alt+R

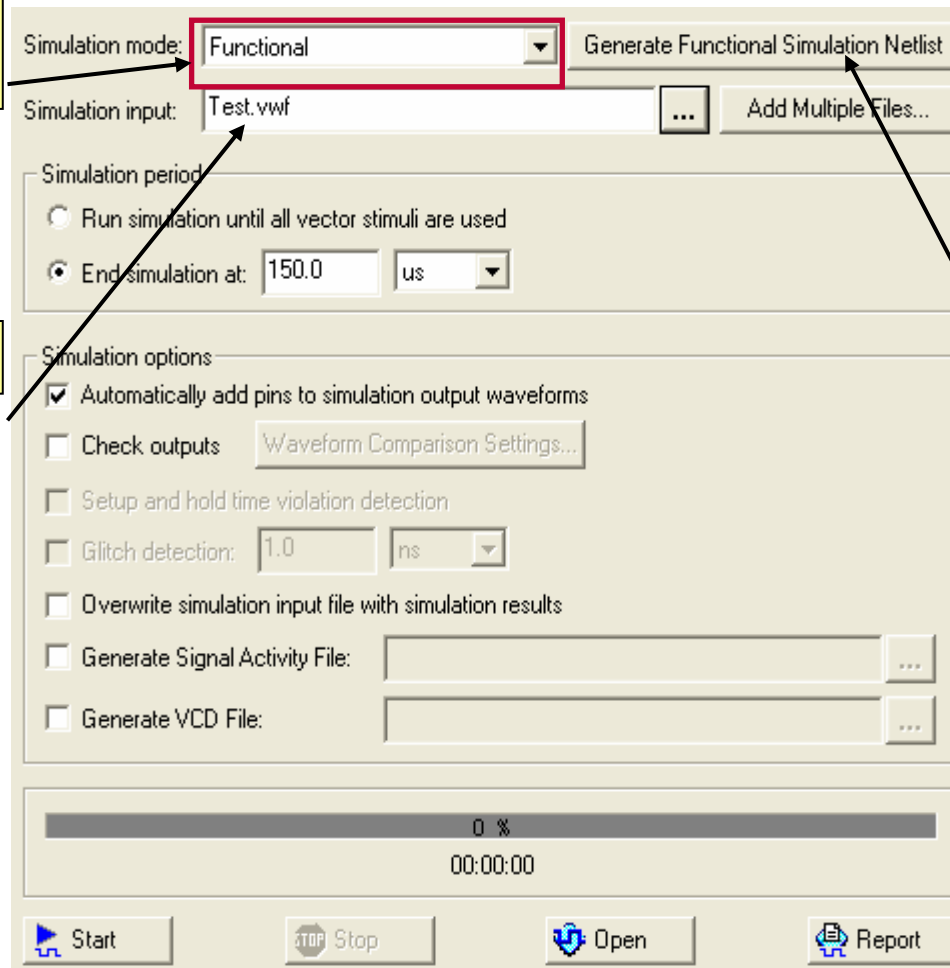
ecn  
 Technology

# 用Quartus II进行功能/时序仿真

## ■ 从Processing菜单打开simulation tools

选择做功能仿真  
还是时序仿真

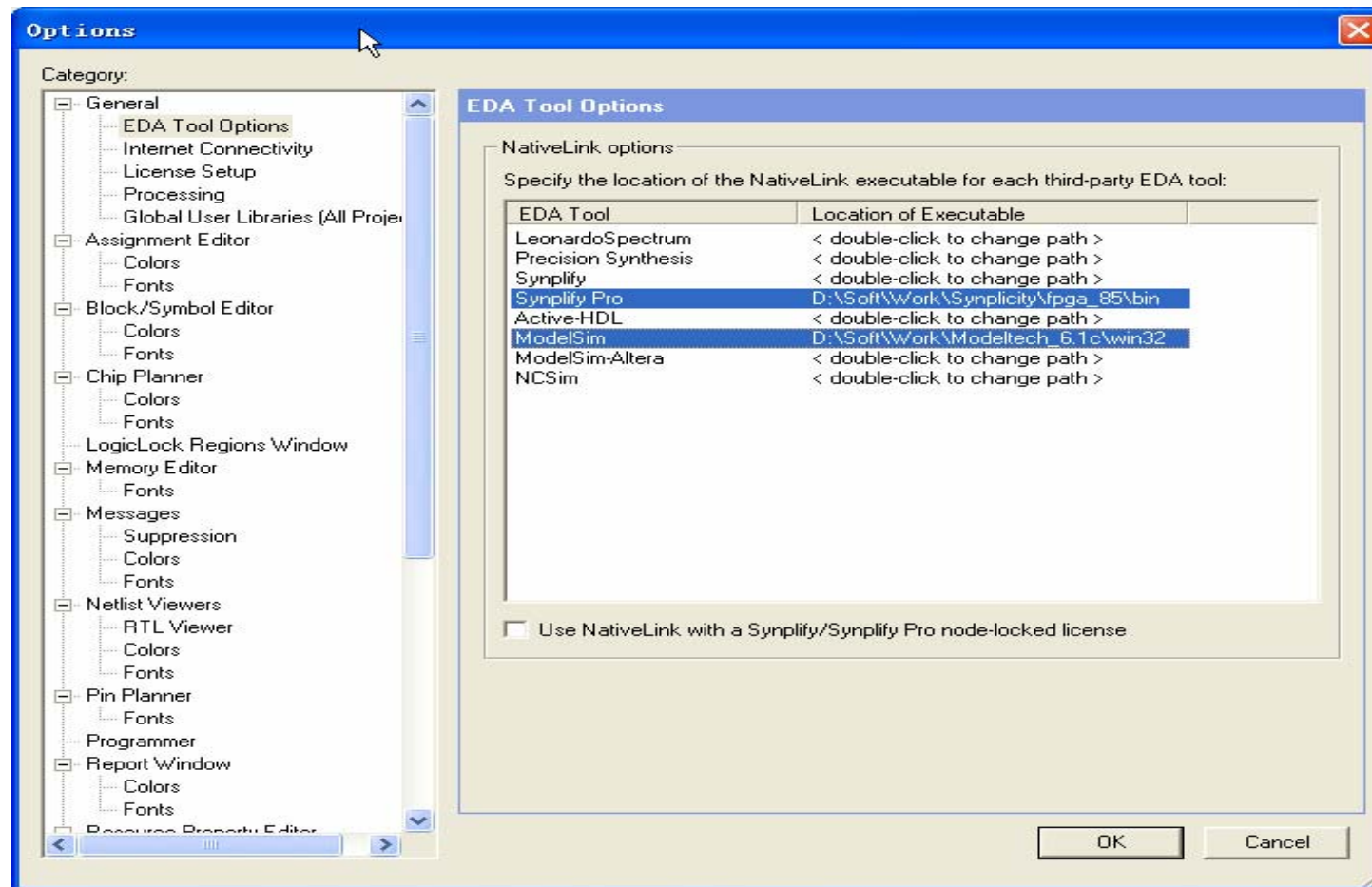
选择激励文件



缺省情况下，QII产生的是  
timing netlist，因此在做功能  
仿真前，需要先生成功能仿真  
网表文件

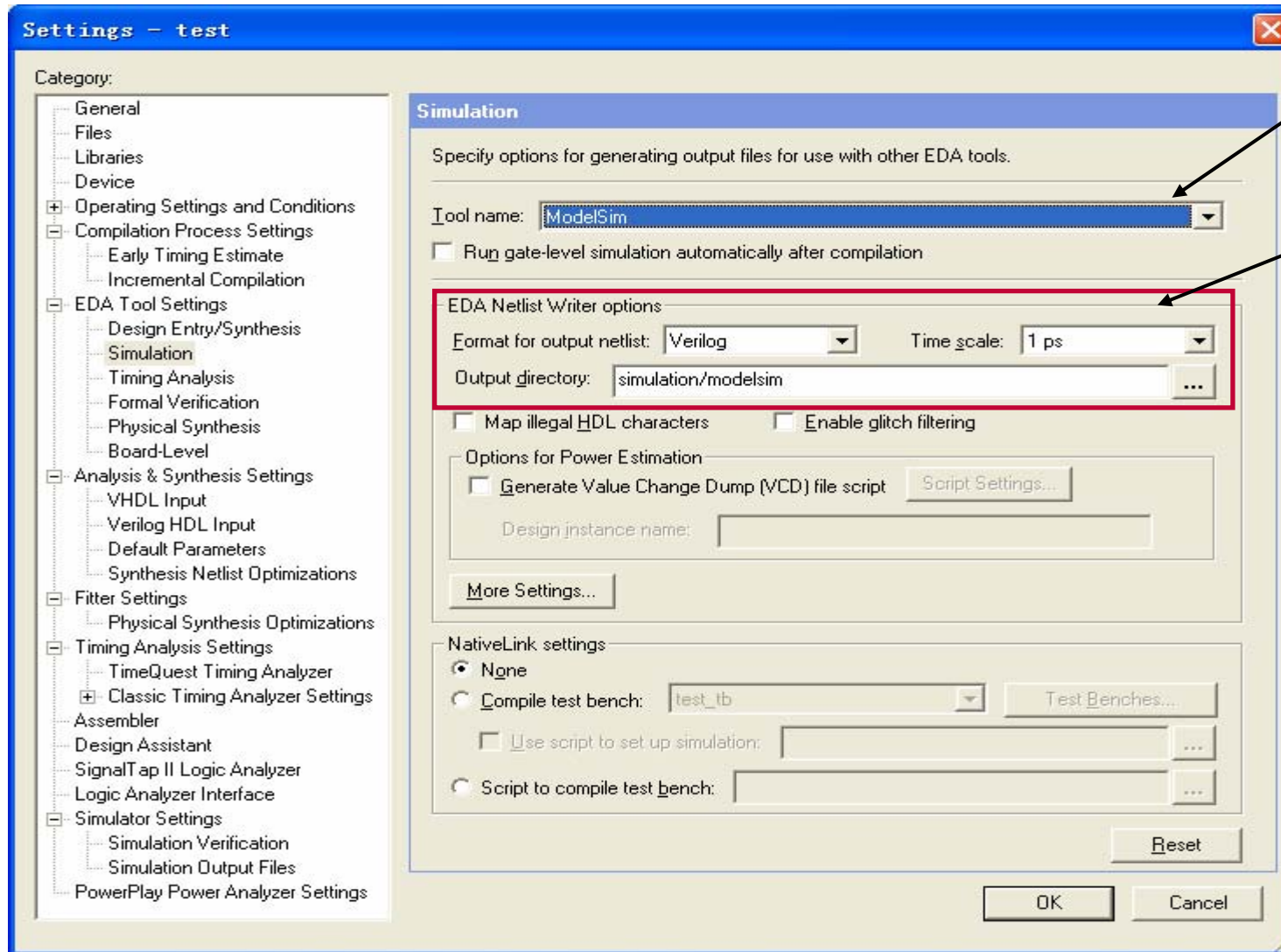
# 调用ModelSim-SE进行功能/时序仿真

- 1、选择Tools->Options, 点击【General】/【EDA Tool Options】, 设置ModelSim执行文件的安装路径（Synplify Pro也在此设置）。





## 2、选择EDA Tools Settings下的Simulation栏，设置仿真工具。

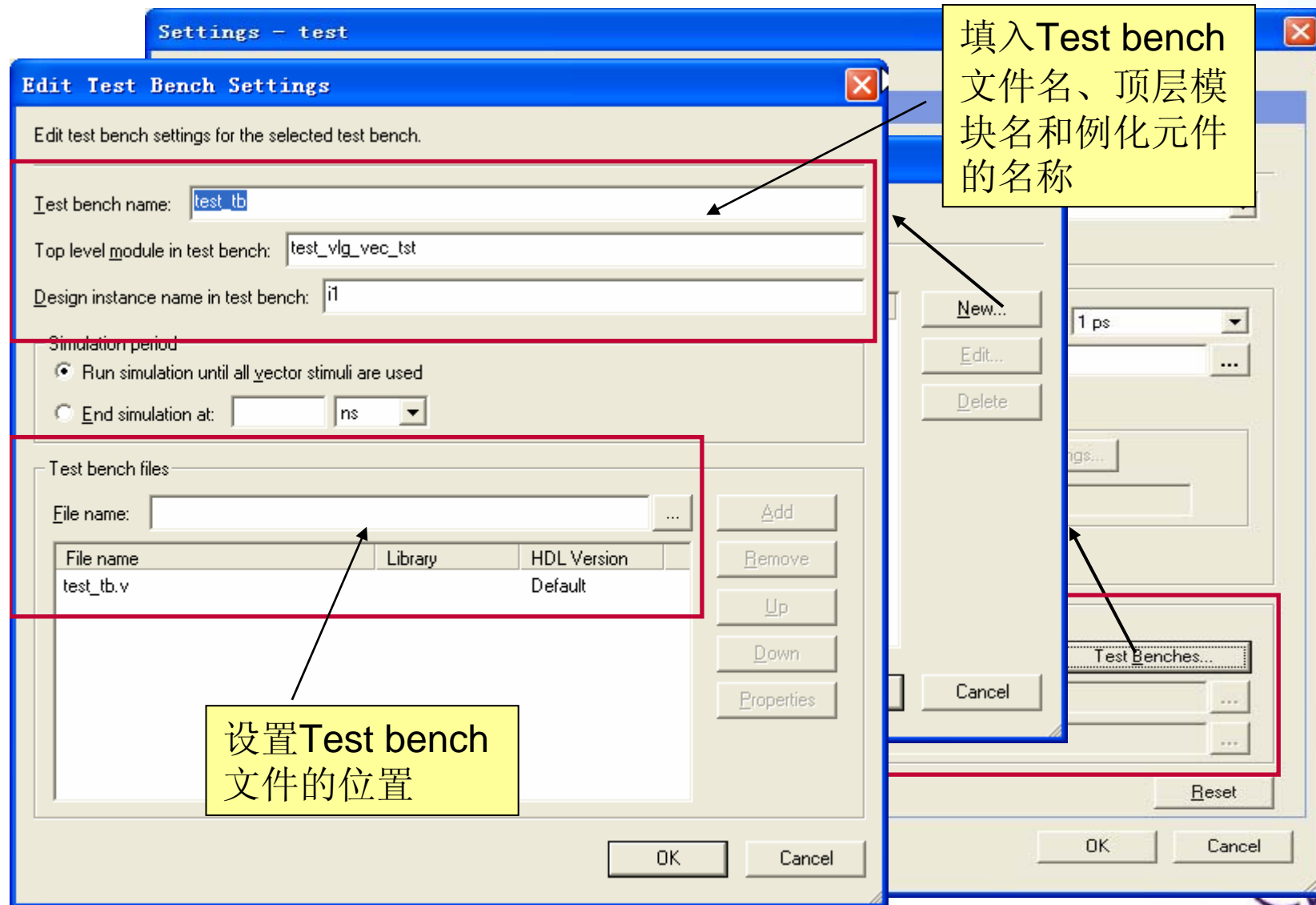


选择仿真工具


设置生成的  
仿真文件的  
语言和所存  
的路径



### 3、重点，设置TestBench!!!



## 4、运行仿真

- 点击  “Start Compilation”按钮编译工程，完成之后在当前的工程目录下可以看到一个名为“Simulation”的新文件夹，下面的“ModelSim”文件夹下包括仿真需要的.vo网表文件和包含延迟信息的.sdo文件。
- 如果之前在“Settings → EDA Tools Setting → Simulation”出现的设置栏中选中了“Run this tool automatically after compilation”，编译完成后Quartus会自动调用ModelSim进行门级时序仿真。
- 如果没选，则选择菜单Tools → EDA Simulation Tool → Run EDA Gate Level Simulation，Quartus便会调用ModelSim进行门级时序仿真。
- 选择Run EDA RTL Simulation则进行行为级仿真。
- 用户也可以单独在ModelSim中通过【Tools】/【Execute Macro】运行Quartus II自动生成的\*\_run\_msim\_gate\_verilog.do文件进行时序仿真或者\*\_run\_msim\_rtl\_verilog.do文件进行功能仿真。

**ALTERA.**

# Quartus II 软件使用教程


器件编程

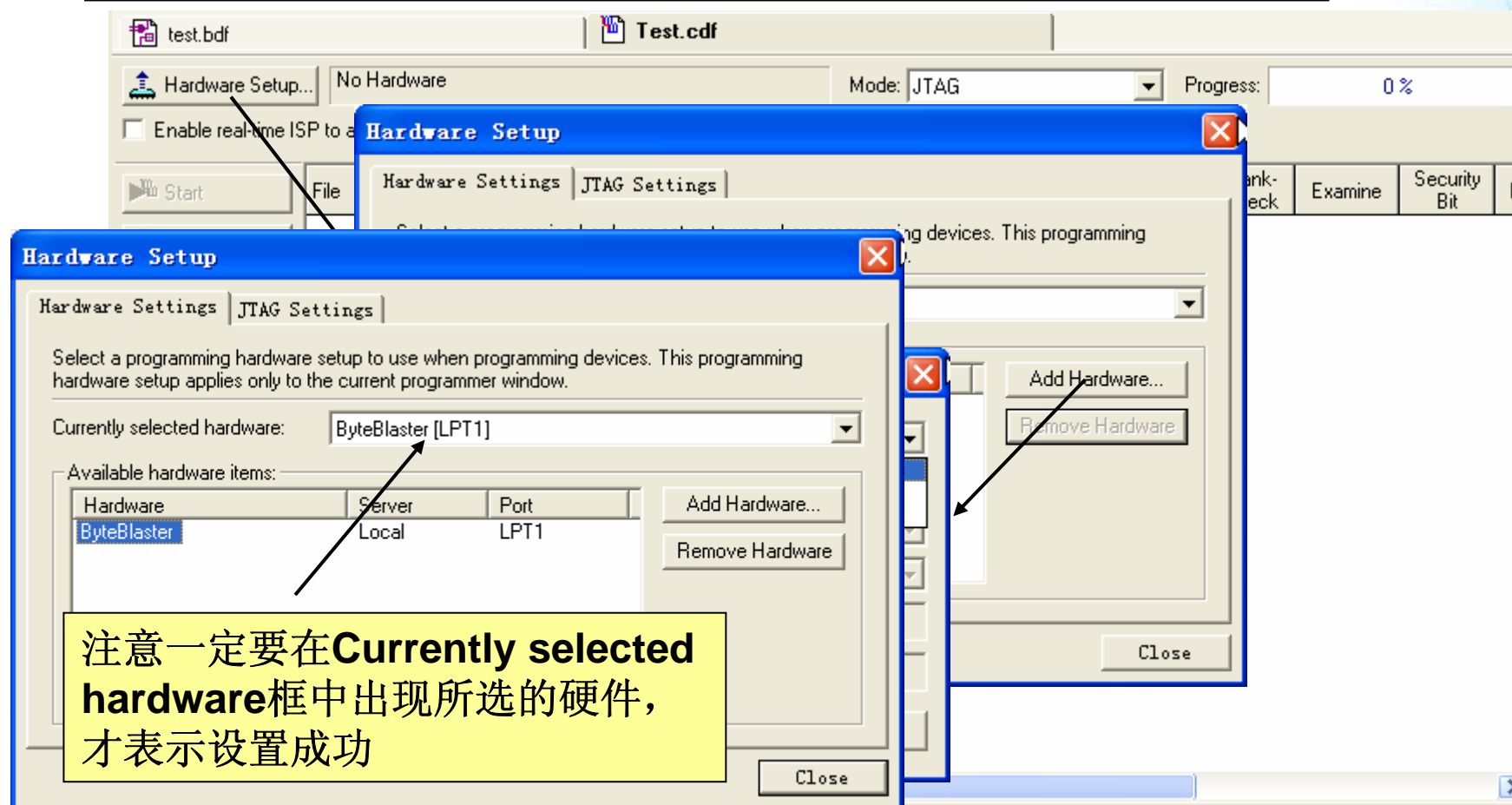


## 器件编程

- Alrera编程器硬件包括MasterBlaster™、ByteBlasterMV™、 ByteBlaster™ II、 USB-Blaster和 Ethernet Blaster下载电缆，或 Altera 编程单元 (APU)。
- Quartus II软件编程器具有四种编程模式：
  - 被动串行模式（Passive Serial mode）；
  - JTAG模式；
  - 主动串行编程模式（Active Serial Programming mode）；
  - 套接字内编程模式（In-Socket Programming mode）。

# 打开编程器窗口

选择Tools-> Programmer或者单击快捷图标，打开编程器窗口



# 设置编程选项

选择编程模式

点击Start按钮

进度条显示编程进度，100%时提示编程完毕

添加pof文件，Mode栏选择Passive serial Programming（对应EPC配置器件）或者Active Serial Programming（主动串行，对应EPCS配置器件），由于是对配置器件编程，因此编程选项中选择多种操作，如Program/Configure、Verify、Blank-Check、Examine。

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit
test.pof	EPCS4	06738333	00000000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page_0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



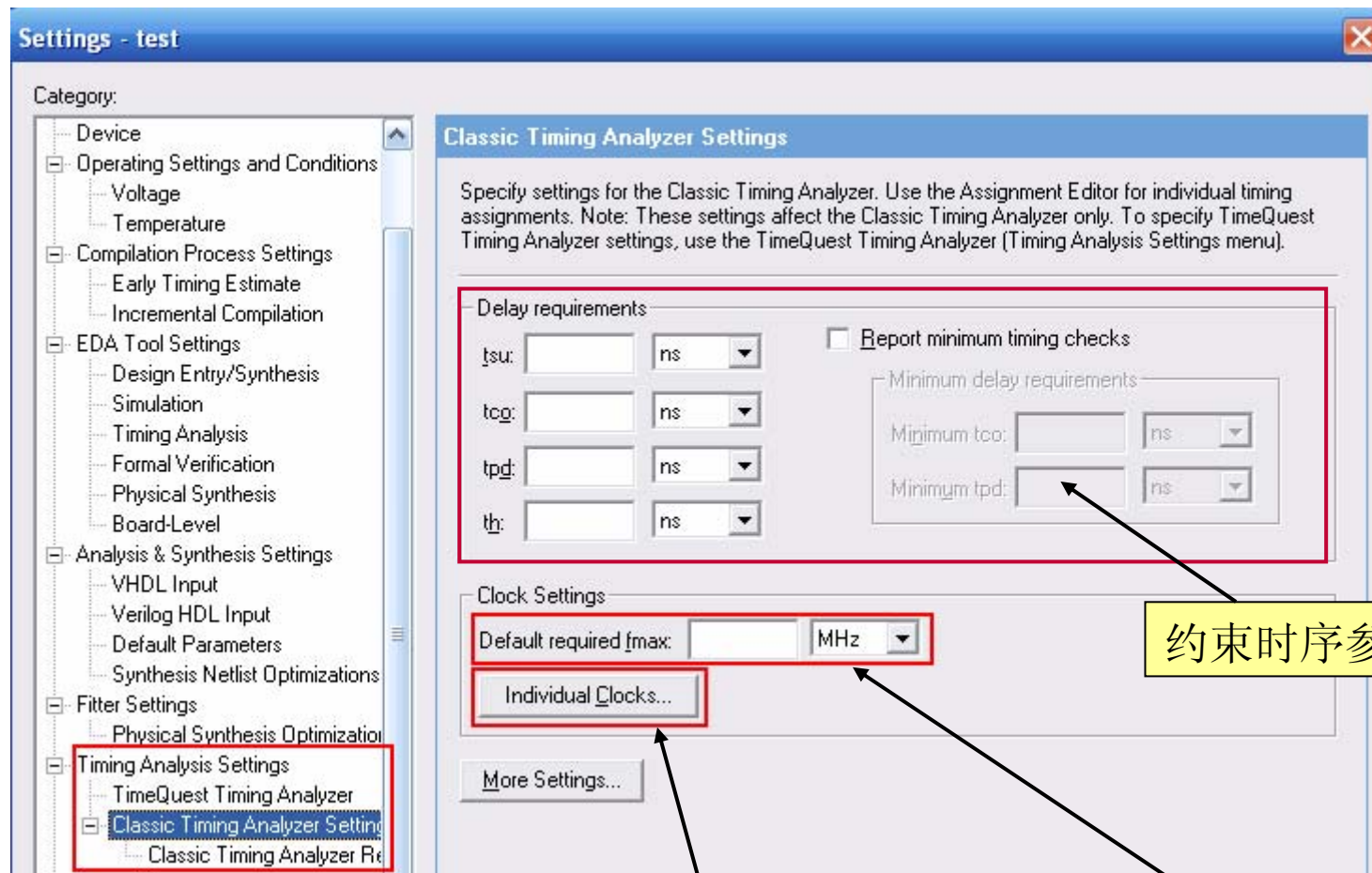
**ALTERA.**

# Quartus II 软件使用教程

时序约束



# 时序约束设置—时钟频率设置



约束时序参数

多时钟设计中的全局时钟约束

单时钟设计中的全局时钟约束

-如果设计中的时钟都由PLL产生，QII会自动约束

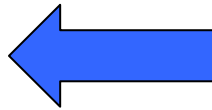
© 2008 Cytech Technology Ltd., Co

# 时序分析及优化-最基本的方法

- 时序优化(Fmax优化)最根本、最有效的方式还是对设计代码的优化
- 常用几种速度优化的技巧(Coding Style):
  - 增加流水级
  - 组合逻辑平衡
  - 复制高扇出结点
  - 用户状态机设计
  - 模块边界输入输出寄存
- QII软件也为工程师提供了很多方便设计优化的选项

# 时序分析及优化-优化首选及保持时间优化

- 通过时序分析报告发现时序存在的问题
  - 结合List Path和Locate功能
- 不可以把所有优化选项一起加上，需要对症下药
- 首选第一步是设置综合优化选项，选择优化目标的优先原则
- 对于保持时间告警的尝试解决，设置保持时间的优化选项





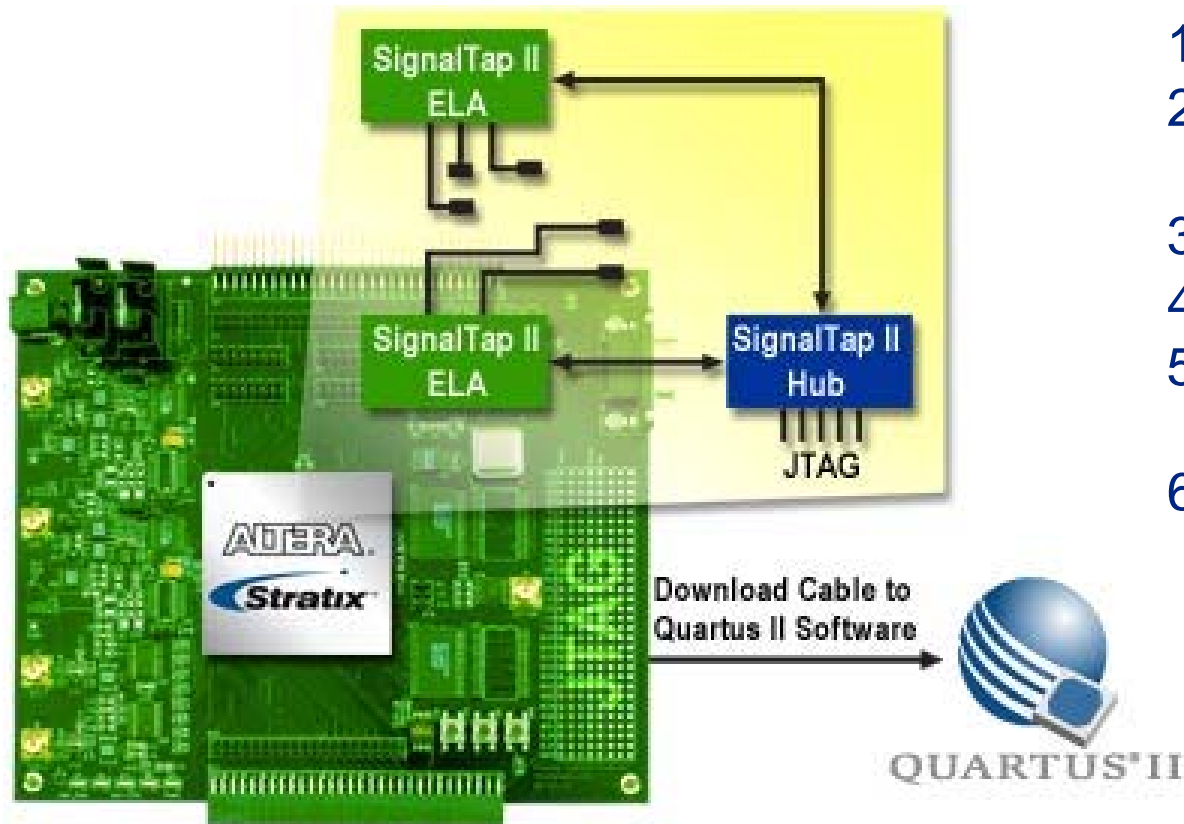
**ALTERA.**

# Quartus II 软件使用教程

*SignalTap II 逻辑分析仪*



# SignalTap II如何工作?



1. 配置ELA
2. 将ELA和原有设计一起下载到FPGA中
3. 启动 ELA
4. 定义触发条件
5. 采样, 并将数据存储在FPGA内部剩余RAM中
6. 通过JTAG口将采样数据传递给Quartus II软件

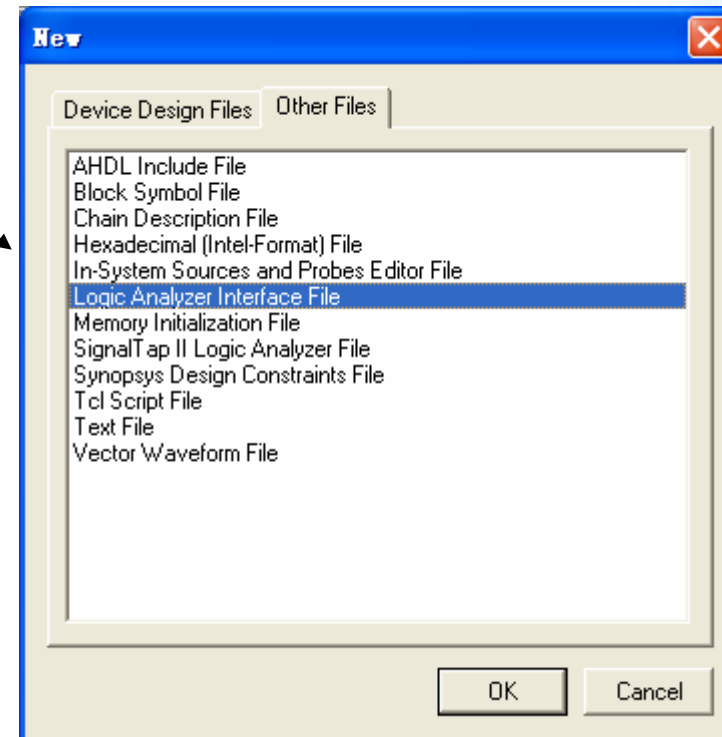


# 特性

Feature	Benefit
支持多个SignalTap II 核	支持在单个芯片中同时存在多个不同时钟域或者不同功能模块的逻辑分析仪IP函数
支持采样时钟超过200MHz	使用户可以实时观察信号
支持增量编译	允许用户添加/编辑逻辑分析仪的属性，而不影响现有设计的布局布线
最大1024个数据通道/ 每个通道最大128K采样点	允许用户观察大量大量的采样数据
最大10级触发条件	为设置复杂的触发条件提供了足够的灵活性
支持外部触发	允许用户用外部信号触发逻辑分析仪或者输出一个触发信号
支持基本或高级触发功能	支持信号电平、固定值、复杂算数逻辑或者状态机多种触发模式
支持多种文件格式存储数据	使得采样数据可以被第三方验证工具读入、显示和分析

# 1) 创建一个新的 .STP 文件

- 方法1
  - 选择菜单 **Tools** → **SignalTap II Embedded Logic Analyzer**
- 方法2
  - 选择菜单 **File** → **New**
- 默认文件名为 stp1.stp



# .STP 文件的界面

The screenshot shows the Quartus II .STP file interface. The main window title is "Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filtref - [stp1.stp\*]". The menu bar includes File, Edit, View, Project, Processing, Tools, and Window. The toolbar contains icons for opening, saving, and compiling. The Instance Manager panel at the top left shows a table of instances:

Instance	Status	Incremental Compilation	LEs: 651	Memory: 65536	M512/LU
auto_signaltap_0	Not running	<input checked="" type="checkbox"/>	651 cells	65536 bits	

The JTAG Chain Configuration panel at the top right shows "No device is selected". The Signal Configuration panel on the right shows settings for "clk" as the clock, "4 K" sample depth, "M4K/M9K" RAM type, and "Circular" buffer acquisition mode. The main table in the center lists nodes and their configurations:

Type	Alias	Node Name	Data Enable	Trigger Enable	Trigger Levels
		d[0..7]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXb XXXXXXb XXXXXXb
		d[0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[1]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[2]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[3]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[4]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[5]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[6]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		d[7]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		taps:inst[xn]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXb XXXXXXb XXXXXXb
		taps:inst[xn[0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

The Hierarchy Display panel at the bottom left shows the design hierarchy. The Data Log panel at the bottom right shows the data log for "auto\_signaltap\_0".

实体管理

JTAG 链配置

节点列表

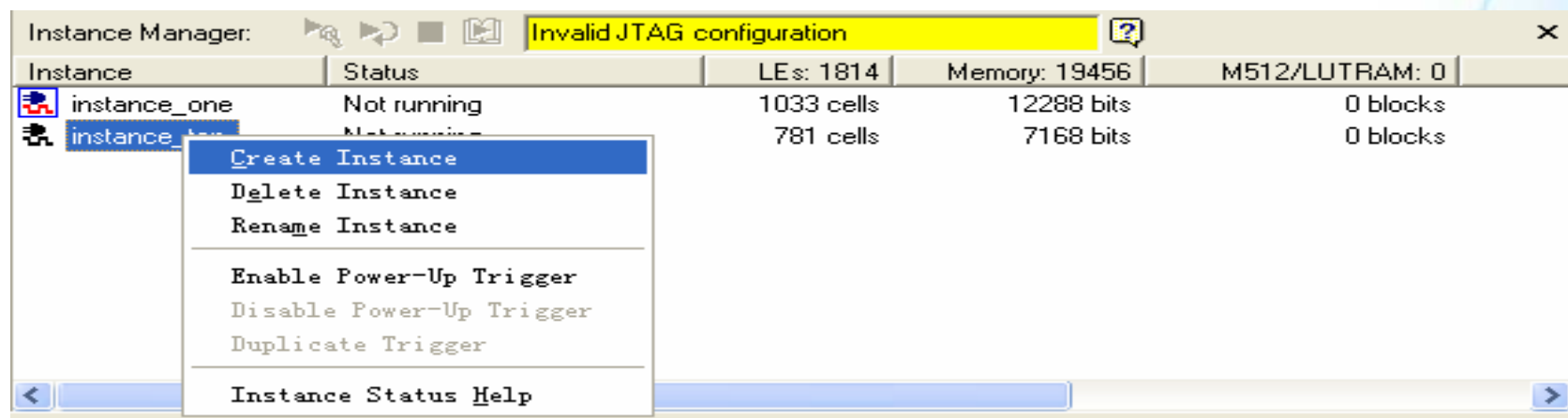
采样信号配置

设计层次

存储日志 (记录采样设置和结果)

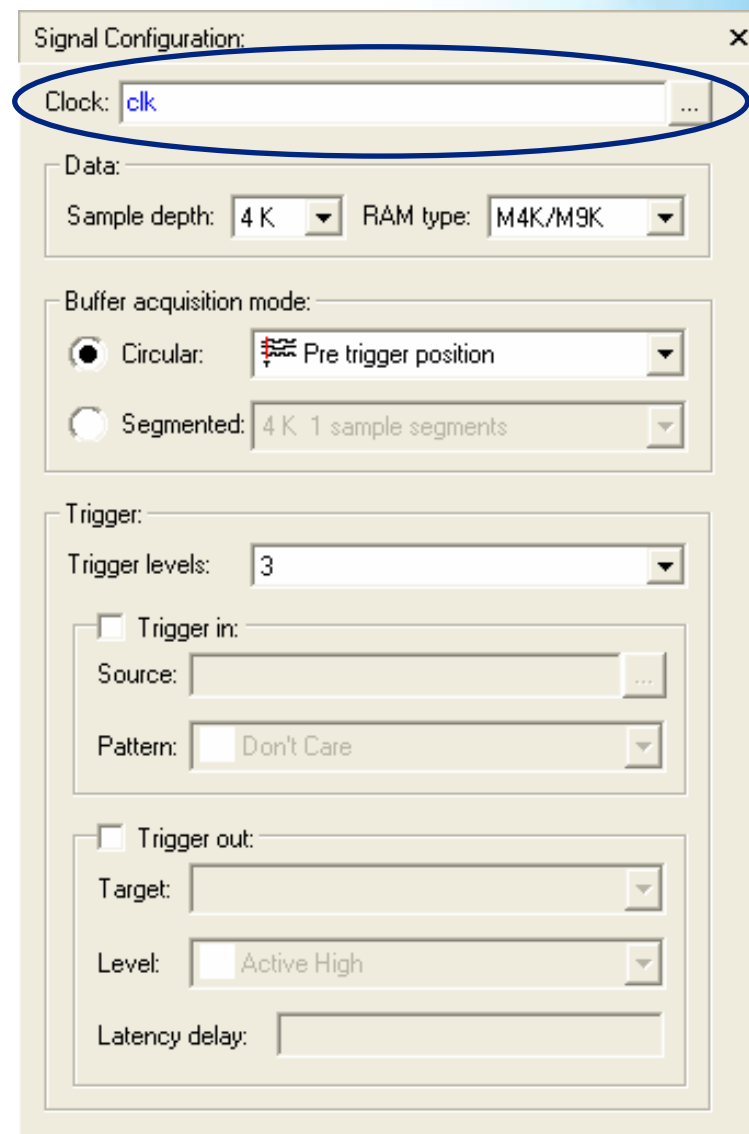
# 实体管理

- 增加/删除.STP文件中的实体模块
- 切换对哪个实体进行操作
- 显示ELA占用的资源
- 运行和控制实体



# 设置采样时钟

- Altera推荐使用全局时钟，而不要使用门控时钟
- 在每个采样时钟上升沿将被测信号存储到缓存
- 如果没有分配采样时钟，软件自动产生一个外部引脚名
  - **auto\_stp\_external\_clock**
  - ELA 建议此外部信号连接到专用时钟脚上 (用Pin Planner分配)



# 指定采样深度和RAM类型

- 采样深度
  - 设置每个信号的采样点数
  - 0 to 128K 采样深度
- SignalTap II所能显示的被测信号波形的时间长度为Tx，计算公式如下：
  - $T_x = N \times T_s$
  - N为缓存中存储的采样点数，Ts为采样时钟的周期
- 选择RAM类型
  - 选择适当的RAM有利于节省RAM资源

The image shows a 'Signal Configuration' dialog box with the following settings:

- Clock:** clk
- Data:**
  - Sample depth:** 4 K
  - RAM type:** M4K/M9K
- Buffer acquisition mode:**
  - Circular:** ☒ Pre trigger position
  - Segmented:** ☐ 4 K 1 sample segments
- Trigger:**
  - Trigger levels:** 3
  - Trigger in:**
    - ☐ Source:
    - ☐ Pattern: Don't Care
  - Trigger out:**
    - ☐ Target:
    - ☐ Level: Active High
    - ☐ Latency delay:



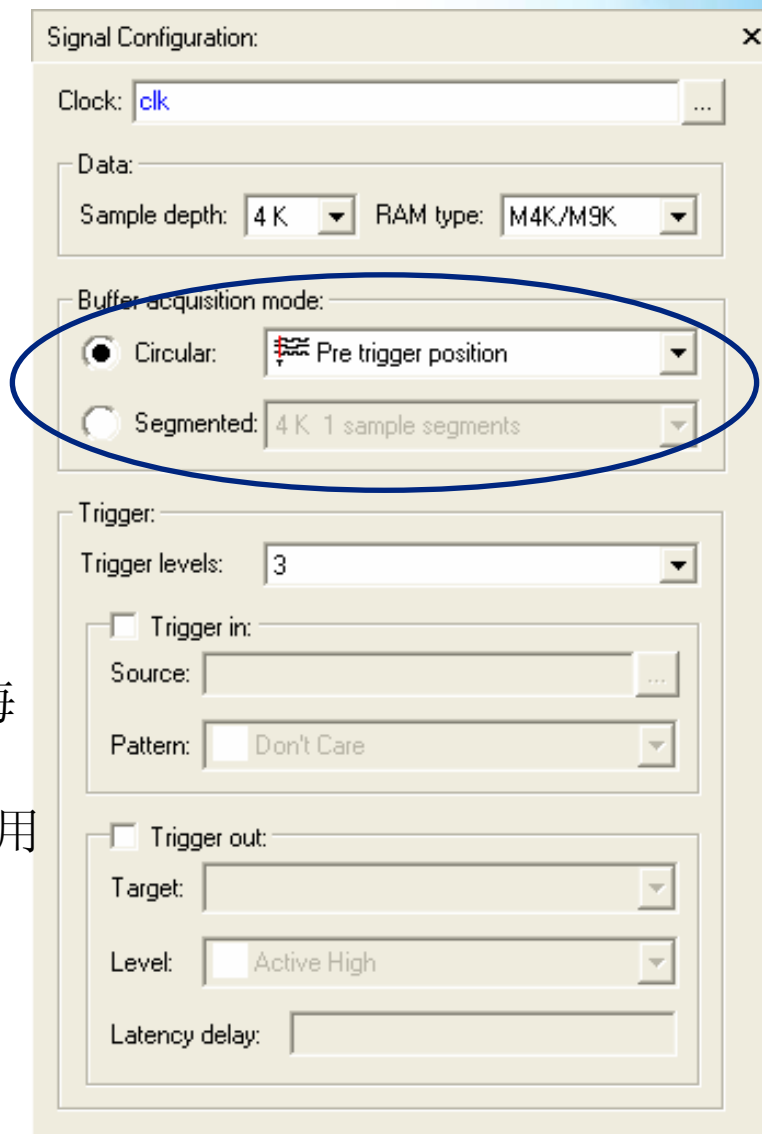
# 数据获取模式

## ■ 环形存储

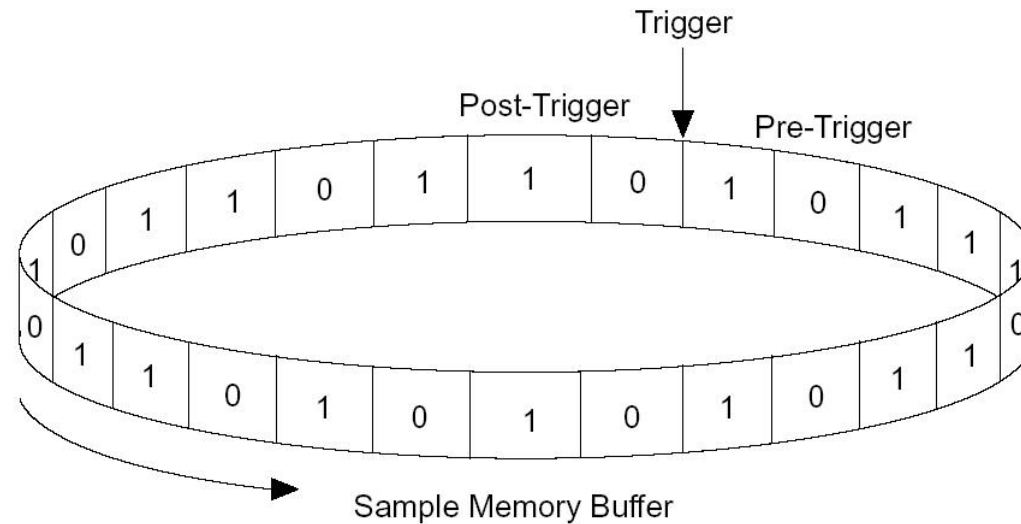
- 指定trigger位置
  - Pre (12% before trigger, 88% after)
  - Center (50% before, 50% after)
  - Post (88% before, 12% after)
  - Continuous

## ■ 分段存储

- 将整个缓存分成多个片段(segment)，每当触发条件满足时就捕获一段数据。
- 可以去掉无关的数据，使采样缓存的使用更加灵活

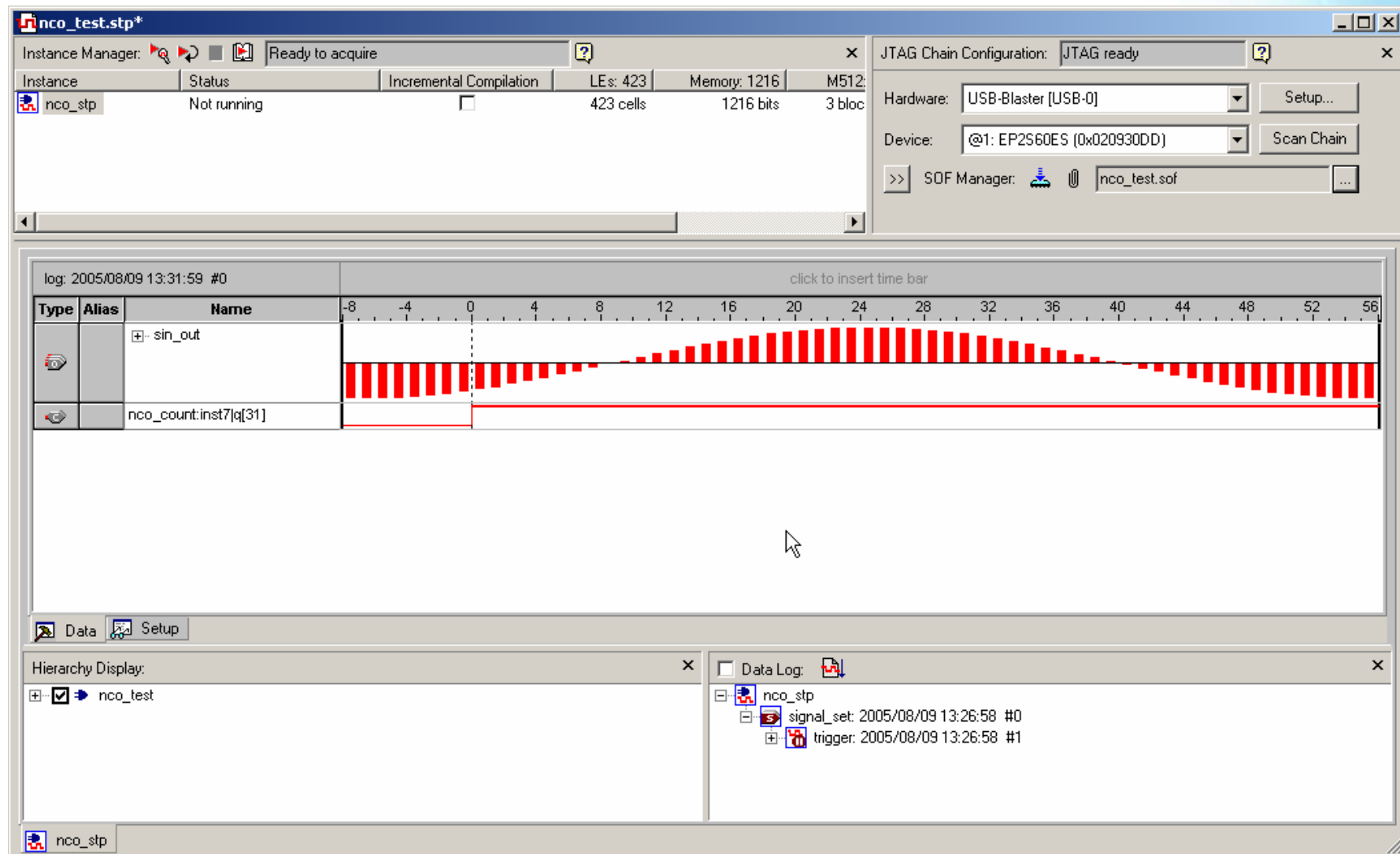


# 环形Buffer

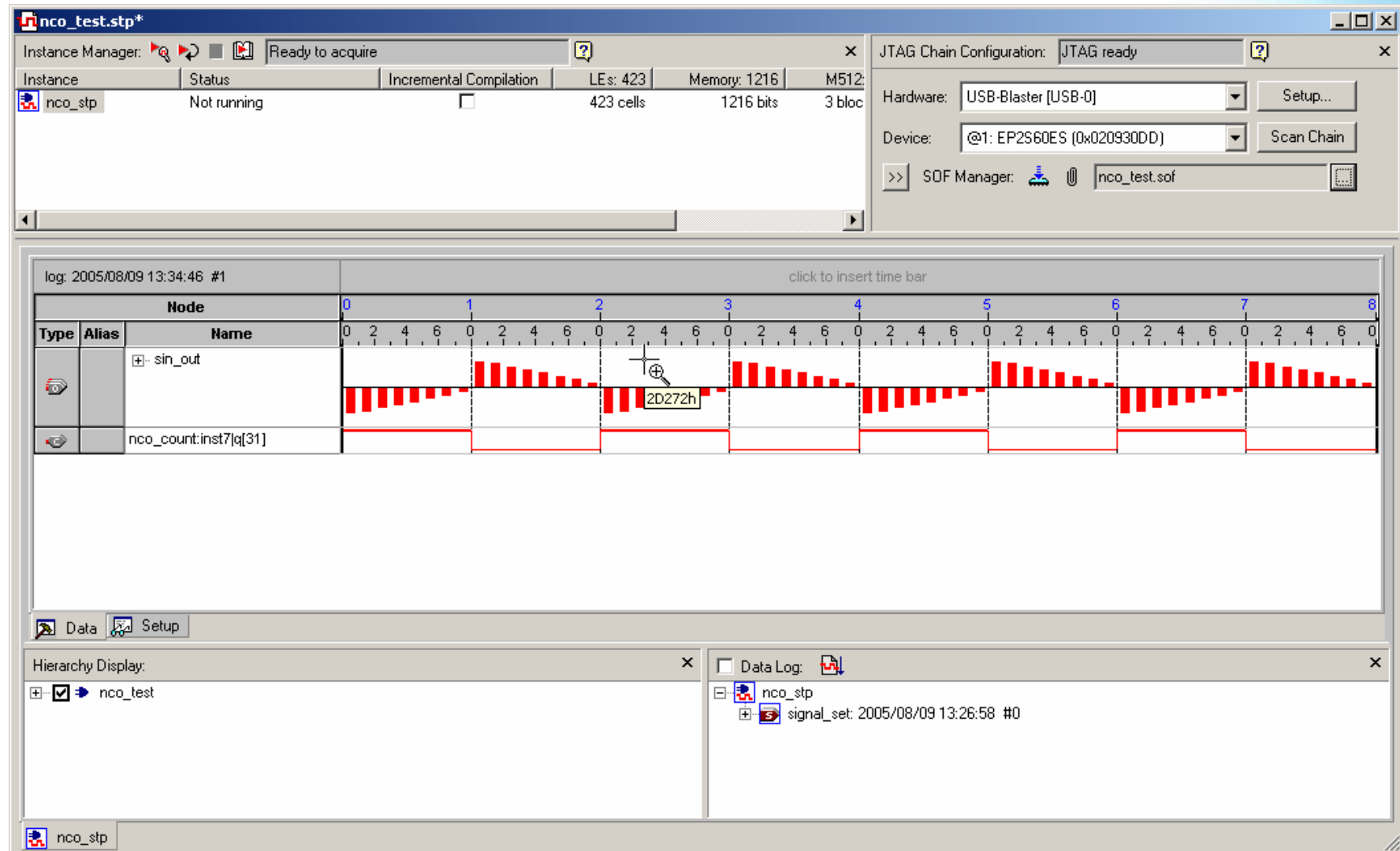


- 触发前，采样数据循环填入一个环形数据缓冲区中
- 触发后，软件采集post-trigger的数据直到填满Buffer

# 环形Buffer示例



# 分段Buffer示例



# 触发

## ■ 触发级别

- 支持 10 级触发条件
- 所有事件发生后才开始获取数据

## ■ Trigger-in

- 任何I/O脚都可以触发分析仪
- Behaves like trigger level “0”
- 自动创建`auto_stp_trigger_in_n`脚，在Pin Planner中进行分配

## ■ Trigger-out

- 当触发条件满足时，输出一个信号表示触发
- 自动创建`auto_stp_trigger_out_n`脚，在Pin Planner中进行分配
  - Latency delay表示从触发到信号输出的延迟时钟数

Signal Configuration:

Clock: `clk`

Data:

Sample depth: `4 K` RAM type: `M4K/M9K`

Buffer acquisition mode:

☒ Circular: `Pre trigger position`

☐ Segmented: `4 K 1 sample segments`

Trigger:

Trigger levels: `3`

☒ Trigger in:

Source: `auto_stp_trigger_in_0`

Pattern: `Rising Edge`

☒ Trigger out:

Target: `auto_stp_trigger_out_0`

Level: `Active High`

Latency delay:

# 节点列表 & 波形窗

The image displays two windows from the Quartus II software, illustrating the 'Node List' and 'Waveform' views.

**Left Window (Node List):** The 'Node List' window shows a table of nodes and their aliases. The 'Setup' tab is selected, indicated by an arrow pointing to the 'Setup' button in the bottom right corner.

Type	Alias	Name	Data Enable	Tri
		COUNTER:u1 ONE_LED_OUT	7	
		...R:u1 ONE_LED_OUT[0]		
		...R:u1 ONE_LED_OUT[1]		
		...R:u1 ONE_LED_OUT[2]		
		...R:u1 ONE_LED_OUT[3]		
		...R:u1 ONE_LED_OUT[4]		
		...R:u1 ONE_LED_OUT[5]		
		...R:u1 ONE_LED_OUT[6]		

**Right Window (Waveform):** The 'Waveform' window displays a timing diagram for the same nodes. The 'Data' tab is selected, indicated by an arrow pointing to the 'Data' button in the bottom right corner. The waveform shows the signal levels for the nodes over time, with a time axis ranging from -6 to 10. The signal for 'ONE\_LED\_OUT' is shown as a red line, and the signal for 'ONE\_LED\_OUT[0]' through 'ONE\_LED\_OUT[6]' are shown as blue lines. The waveform is labeled 'eight' and 'nine' at the top, and 'zero\_ones' at the bottom.

Setup tab    Data tab



# 添加节点

trigger: 2008/04/28 09:31:58 #1

Allow all changes

Type	Alias	Name
		COUNTER:u1 CNT_ONE
		COUNTER:u1 CNT_ONE_EN
		ONE_SEG

Node Finder

Named: \* Filter: SignalTap II: post-fitting

Look in: |top\_counter

Nodes Found:

Name	SignalTap II: post-fitting
~QIC_CREATED_GND~1	Unassigned
COUNTER:u1 Add0~103	Unassigned
COUNTER:u1 Add0~104	Unassigned
COUNTER:u1 Add1~103	Unassigned
COUNTER:u1 CNT_ONE	Unassigned
COUNTER:u1 CNT_ONE[0]	Unassigned
COUNTER:u1 CNT_ONE[1]	Unassigned
COUNTER:u1 CNT_ONE[2]	Unassigned
COUNTER:u1 CNT_ONE[3]	Unassigned
COUNTER:u1 CNT_ONE[3]~410	Unassigned
COUNTER:u1 CNT_ONE[3]~411	Unassigned
COUNTER:u1 CNT_ONE_ENABLE	Unassigned
COUNTER:u1 CNT_TEN	Unassigned
COUNTER:u1 CNT_TEN[0]	Unassigned
COUNTER:u1 CNT_TEN[1]	Unassigned
COUNTER:u1 CNT_TEN[1]~880	Unassigned
COUNTER:u1 CNT_TEN[2]	Unassigned

SignalTap II: post-fitting

SignalTap II: pre-synthesis

SignalProbe

Include subentities

Assignments

Customize... List Stop OK Cancel

双击空白处 设置是否采样该信号

设置触发信号

Data Setup

# 基本触发

所有信号都满足时开始获取数据

选择是否使能该级触发条件

trigger: 2007/05/25 17:04:34 #0

Allow all changes

Node			Data Enable	Trigger Enable	Trigger Levels			
Type	Alias	Name	7	7	1	2	3	4
		COUNTER:u1 ONE_LED_OUT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 <input checked="" type="checkbox"/> Basic	2 <input checked="" type="checkbox"/> Basic	3 <input checked="" type="checkbox"/> Basic	4 <input checked="" type="checkbox"/> Basic
		...R:u1 ONE_LED_OUT[0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	zero_ones	one	two	three
		...R:u1 ONE_LED_OUT[1]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	0	0
		...R:u1 ONE_LED_OUT[2]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1	1	1
		...R:u1 ONE_LED_OUT[3]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1	0	0
		...R:u1 ONE_LED_OUT[4]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	0	0
		...R:u1 ONE_LED_OUT[5]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	0	0
		...R:u1 ONE_LED_OUT[6]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1	0	0

Don't Care

0 Low

Falling Edge

Rising Edge

1 High

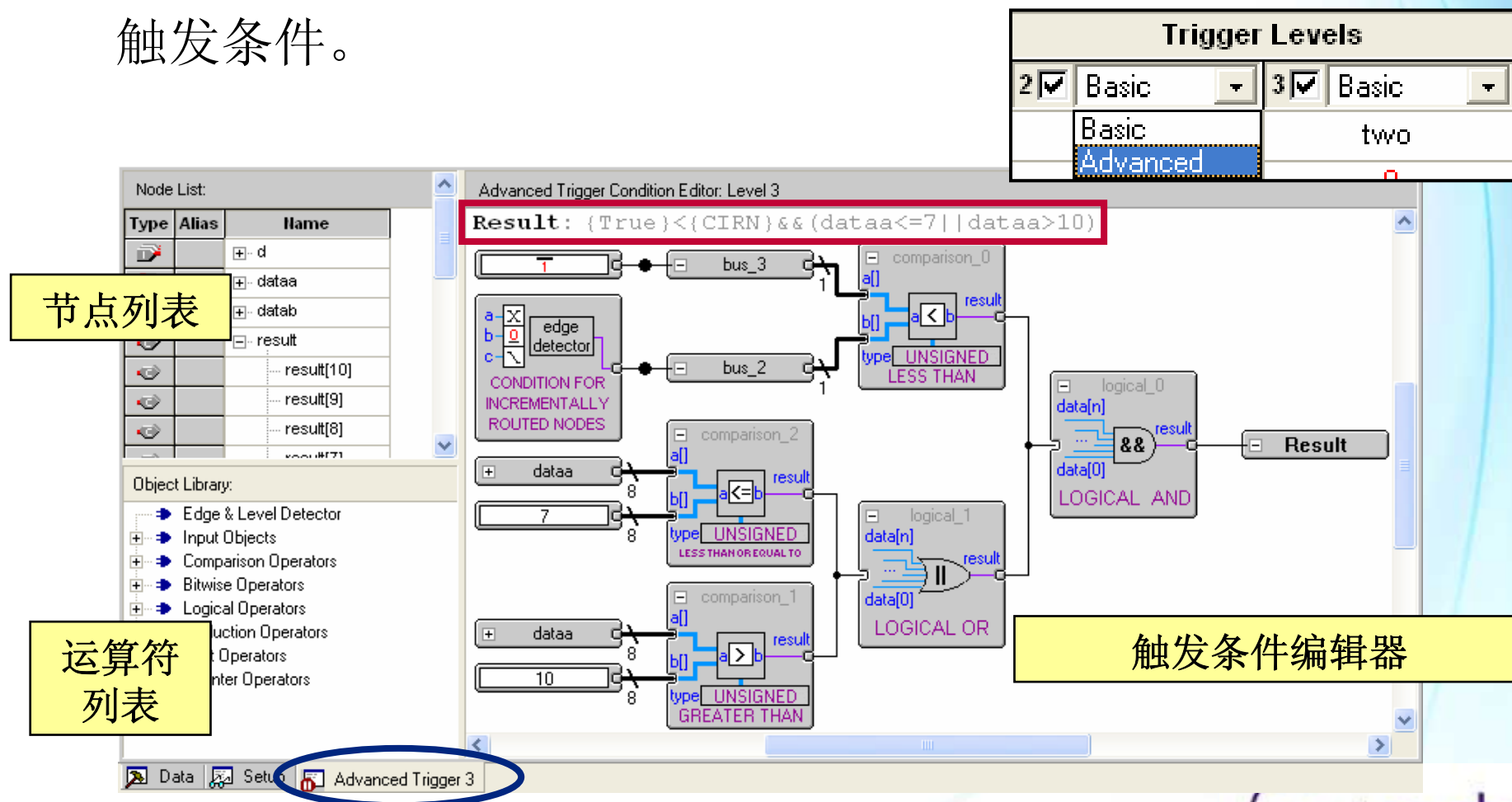
X Either Edge

Insert Value...

右键设置触发值

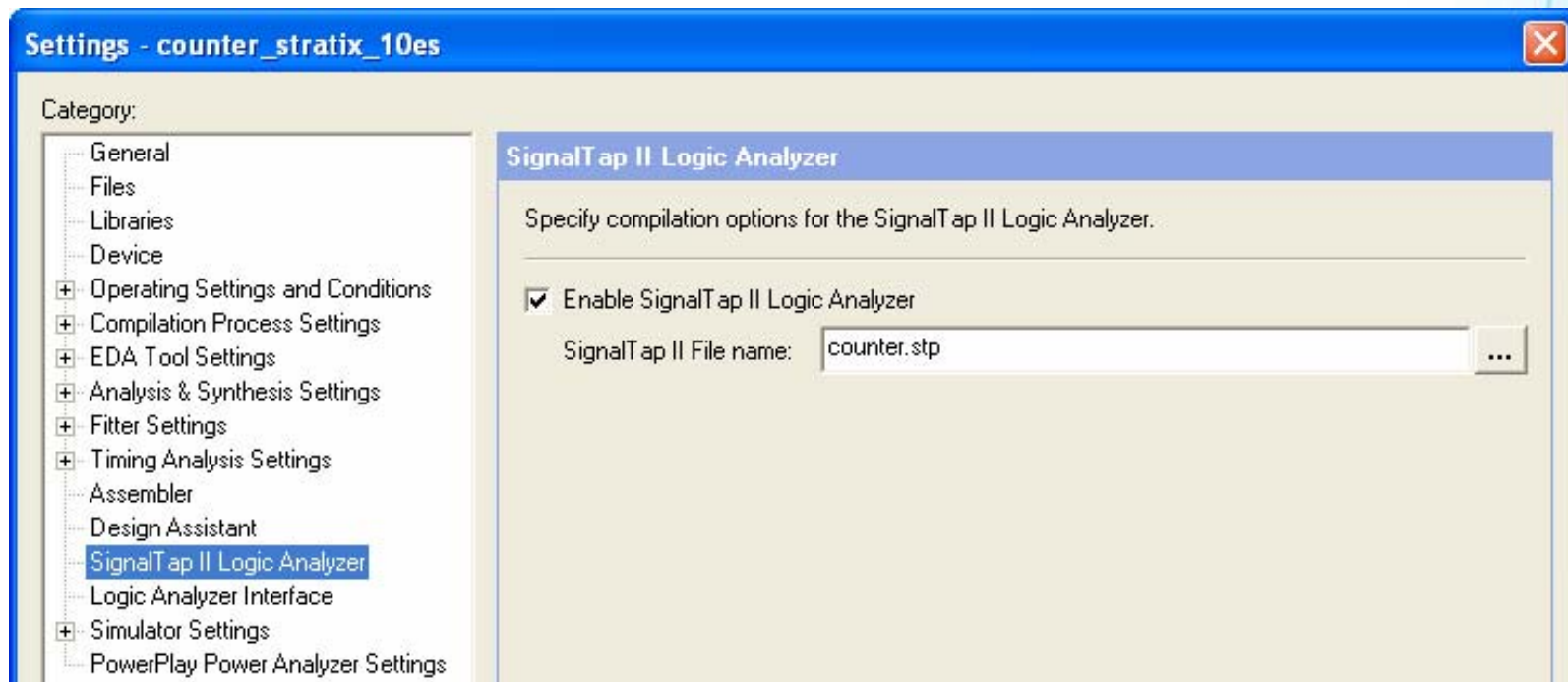
# 高级触发

- 触发类型选择Advanced时，可以在图形界面中创建复杂的触发条件。



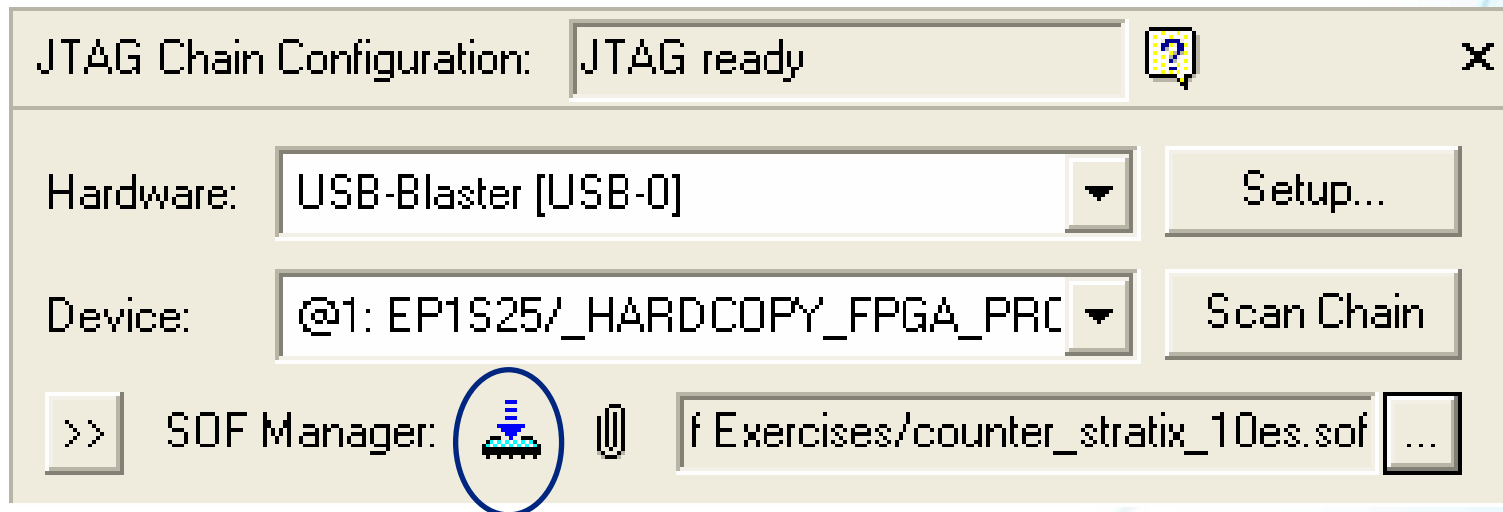
## 2) 保存.STP 文件 & 编译

- 选择菜单 Assignments → Settings
  - 指定当前使用的 SignalTap II 文件

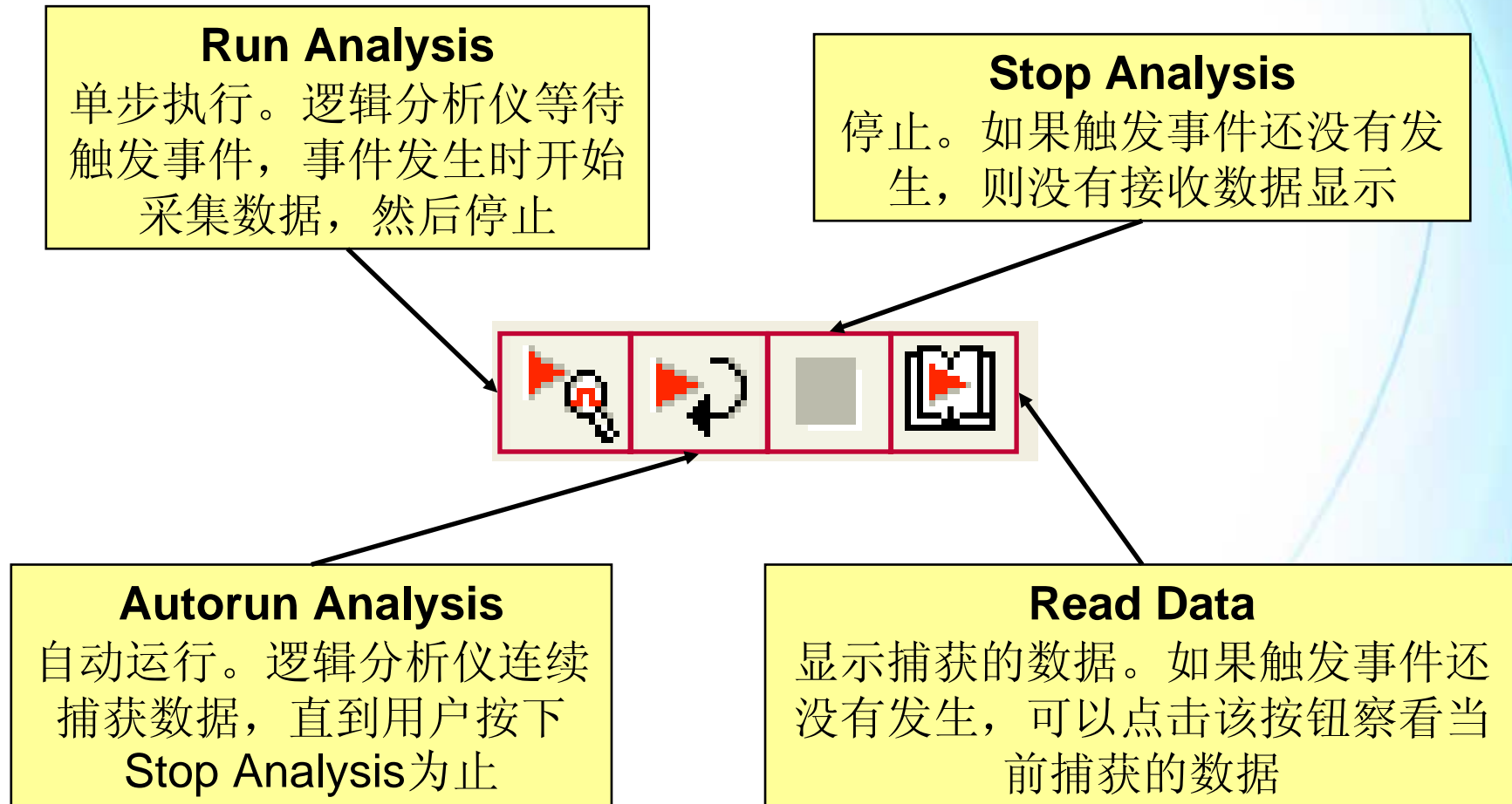


### 3) 器件编程

- 使用Quartus II 编程器或者 SignalTap II 分析仪中的编程界面
  - SignalTap II界面中的编程按钮只能对JTAG链中的当前器件进行编程
  - 使用Quartus II 编程器对多个FPGA进行编程
    - 可以为JTAG链上的每个FPGA单独创建一个 SignalTap II 文件



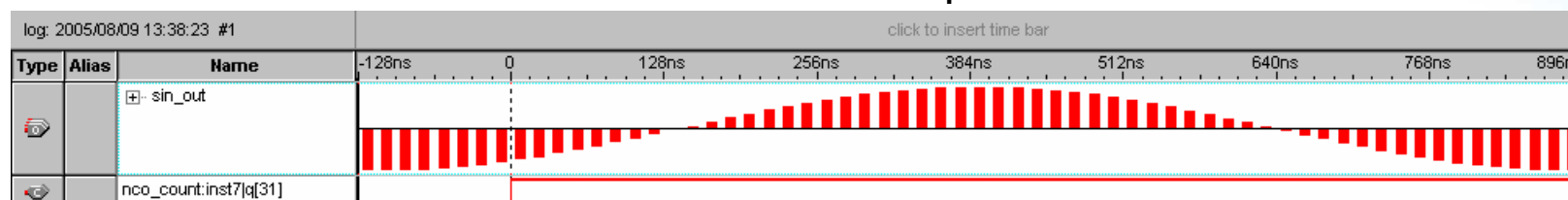
## 4) 获取数据



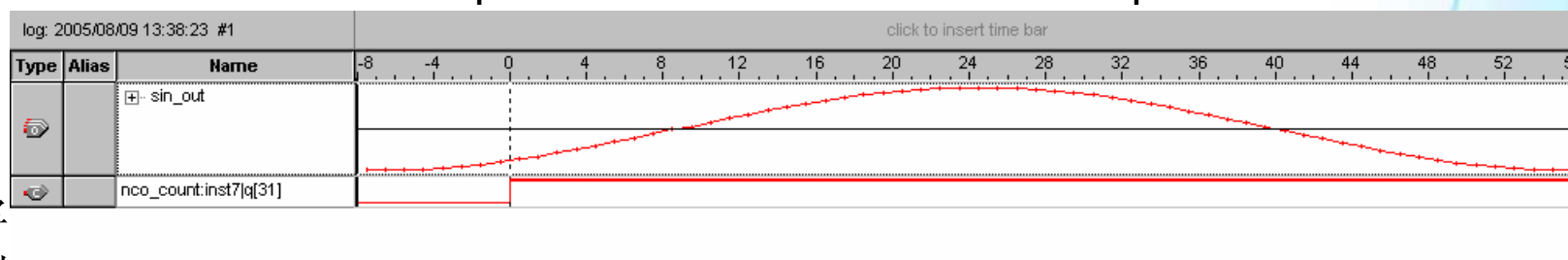


# 显示获取的数据


## Time Formatted Bar Graph



## Sample Number Formatted Line Graph



- 将
- 创建信号节点的列表文件(菜单File → Create/Update)
- 输出结果供其它验证工具显示和分析 (菜单File → Export)
  - 创建 .VWF, .TBL, .CSV, .VCD, .JPG or .BMP 文件



**谢谢!**

**THANK YOU**