

# Synplify pro综合

西安交大SOC设计中心：沈云红

Email: [socman\\_shen@163.com](mailto:socman_shen@163.com)

<http://aiar.xjtu.edu.cn>

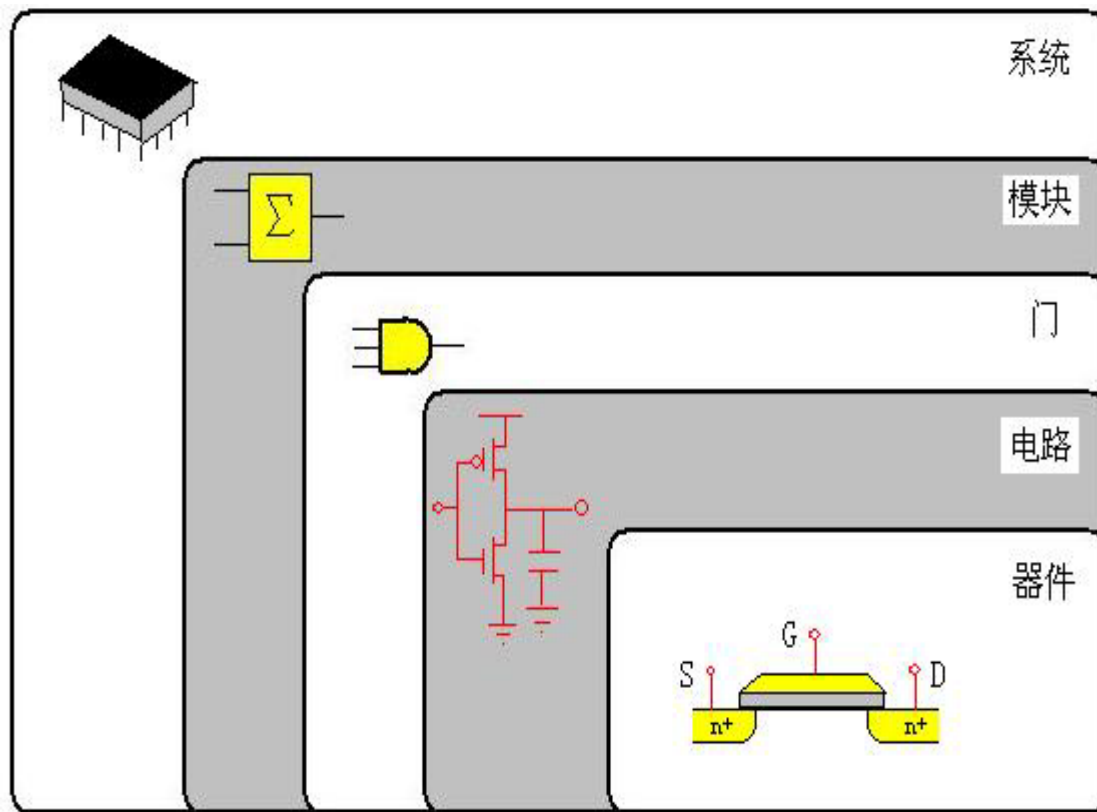
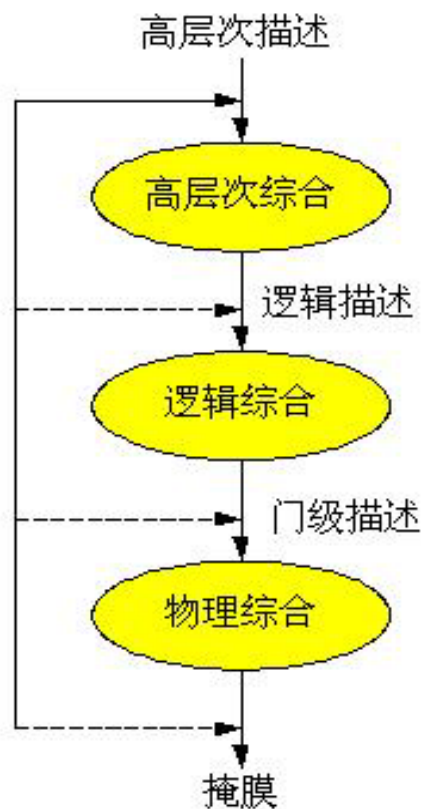
# 什么是综合？

## • 设计描述

- 高层次描述：对整个系统的数学模型描述，它试图在系统设计的初始阶段，通过对系统行为描述的仿真来发现系统设计中的问题，此时考虑更多的是系统结构和工作过程能够达到设计规格的要求，而与具体的工艺和器件无关
- 逻辑描述：对设计进行RTL级描述，导出逻辑表达式，并用功能仿真工具进行仿真
- 门级描述：设计的门级实现，相当于系统的逻辑原理图
- 物理描述：或称版图描述，ASIC设计中表现为GDSII 文件

## • 综合指不同设计描述之间的转换

# IC设计中的综合



# IC设计中的综合

- **高层次综合**：将系统算法层的行为描述转化为寄存器传输层的结构描述
- **逻辑综合**：将寄存器传输层的的结构描述转化为逻辑层的结构描述，以及将逻辑层的结构描述转化为电路的结构描述
- **物理综合**：将系统电路层的结构描述转化为版图层的物理描述

# 综合约束

- **综合约束**

- 设计所期望达到的性能和面积的目标，在综合实现阶段附加约束，是为了便于控制综合实现过程，使设计满足运行速度，引脚位置等方面的要求

- **附加合理的约束**

- 达到面积，速度，性能的良好折中

- **约束原则**

- 面积和速度是设计效果最终的评定标准，在两者冲突时，采用速度优先

# 综合的策略

## • 综合策略

- 自顶向下的层次化综合处理：适合于中小规模设计
  - 优点：1.子模块的约束由综合工具自动进行，综合约束文件简单易行；2.能够全面考虑各个子模块之间的相互关系，从整体上进行时序优化，综合效果比较好
  - 缺点：1.综合时间长，消耗机器内存大；2.子模块的修改影响到整个设计，牵一发而动全身；3.子模块中包含多个时钟或有内部产生时钟，综合不能对每个时钟域进行约束和优化，综合效果差
- 基于时间预算的综合处理：适合由成熟IP构成的设计
  - 优点：1.容易对整个设计进行综合；2.子模块的改变后只须对该模块重新综合；3.能很好的处理多时钟；4.不同子模块综合目标不一致，综合结果最优
  - 缺点：1.需要处理多个综合批处理文件；2.可能忽视全局的关键路径，因为它在子模块也许不是关键路径；3.综合前要得到比较精确的子模块时序信息比较困难

# 综合的策略（续）

- 自低向上的综合处理：适合大规模设计
  - 优点：1.站用机器内存小，并可以在多个机器上并行处理各个子模块；2.可以对各个子模块进行不同目标的优化，修改后只须对该模块重新综合；
  - 缺点：综合过程需要多次迭代直到时序“收敛”；

## • 综合策略的选择

- 取决于设计的特点，设计的规模，设计的总体结构
- 综合的目标

# Synplify pro综合

## • FPGA的综合

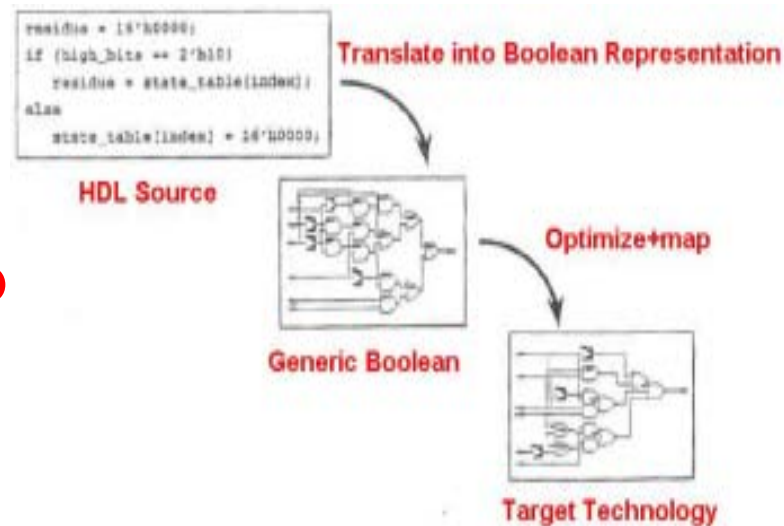
- 针对具体的FPGA器件，将RTL级的设计描述转化成门级描述
- Synthesis=Translation+Optimization+Mapping
  - 对HDL源代码进行编译与逻辑层次上的优化，消除冗余逻辑
  - 对编译的结果进行逻辑映射与结构层次上的优化，生成逻辑网表

## • 特点

- 综合速度快
- 综合效果比较好

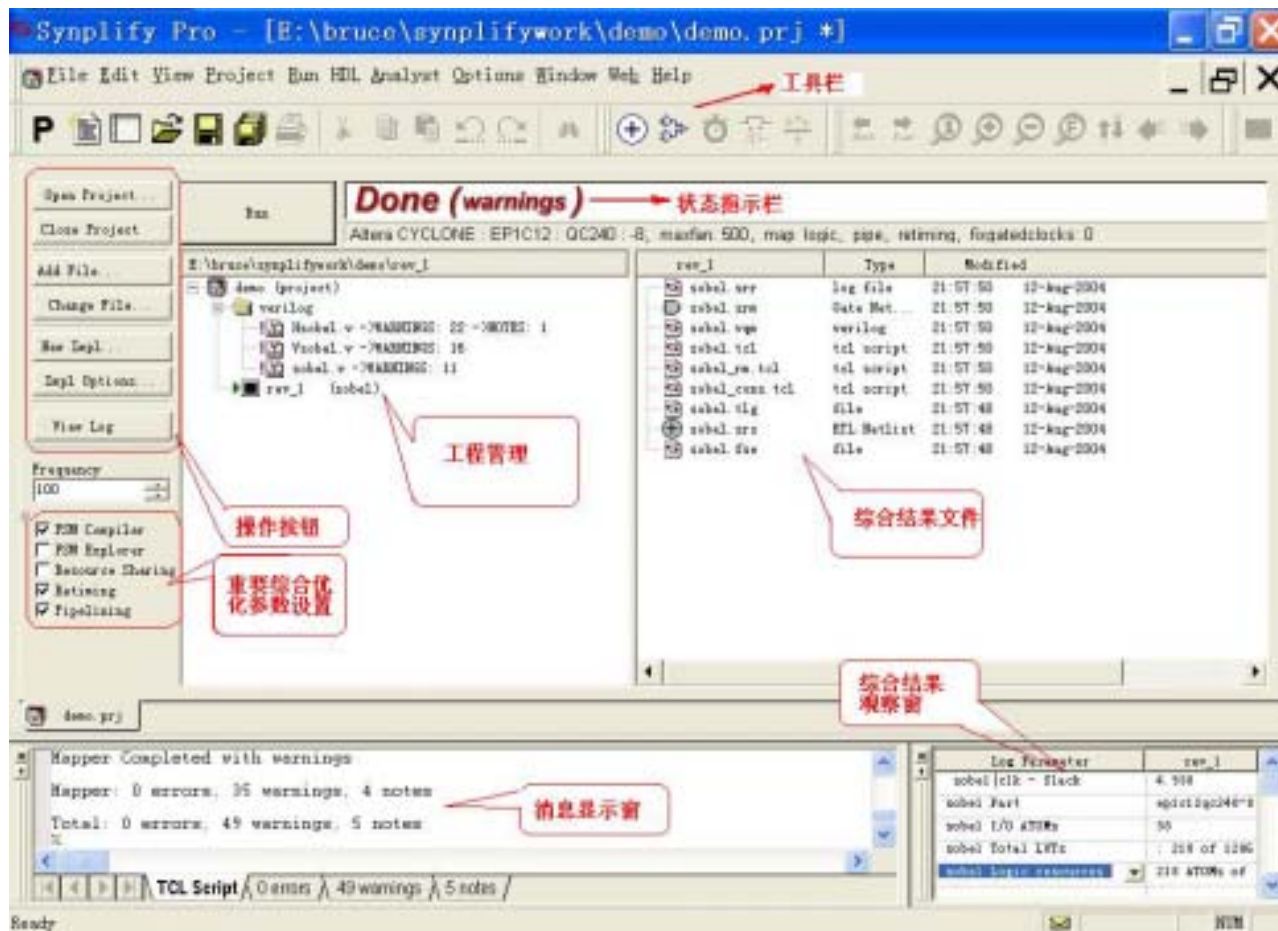
## • Synplify与Synplify pro

- Synplify简装版，后者包含了前者所有功能
- 综合原理和机制完全相同

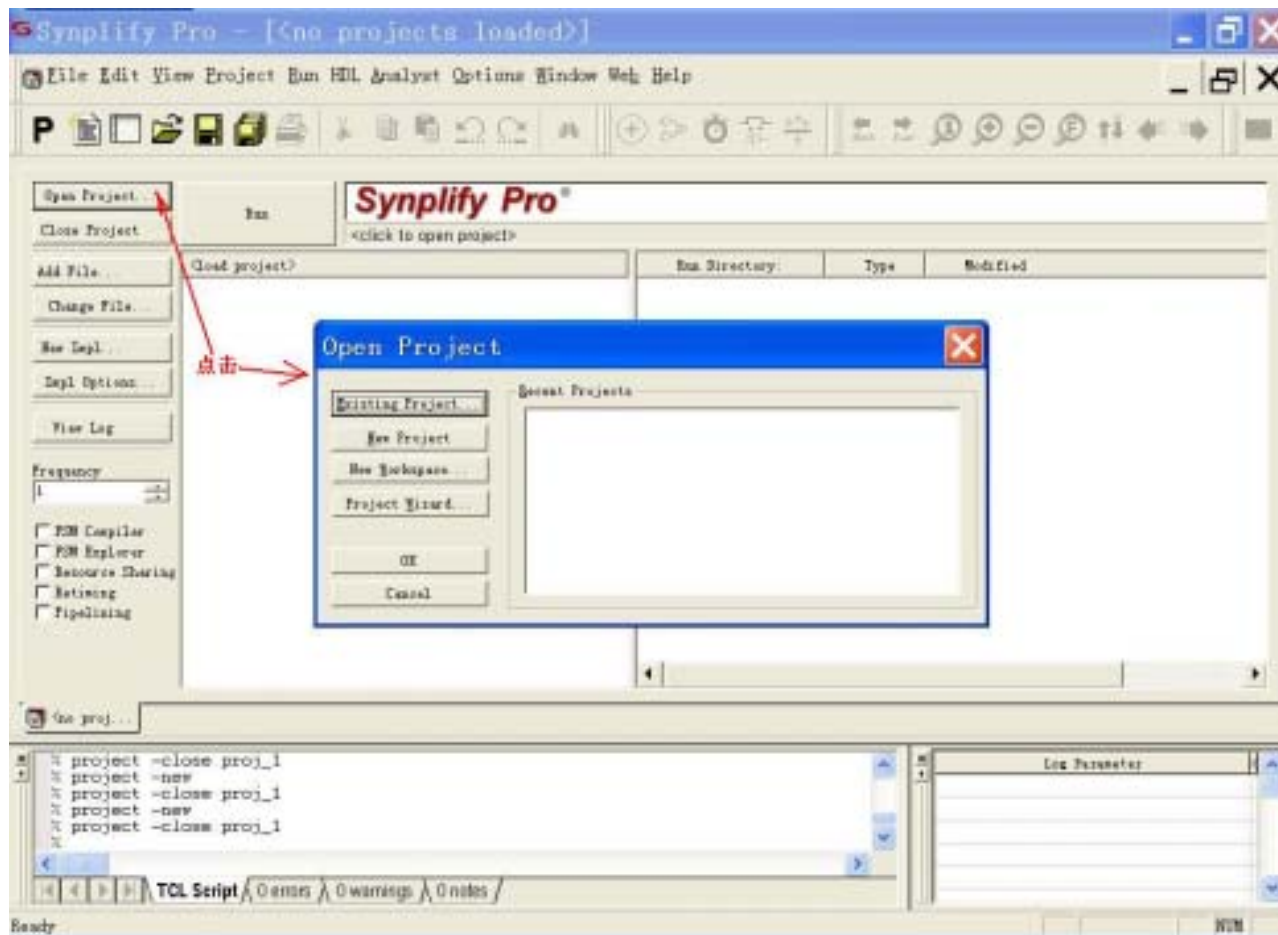




# 界面



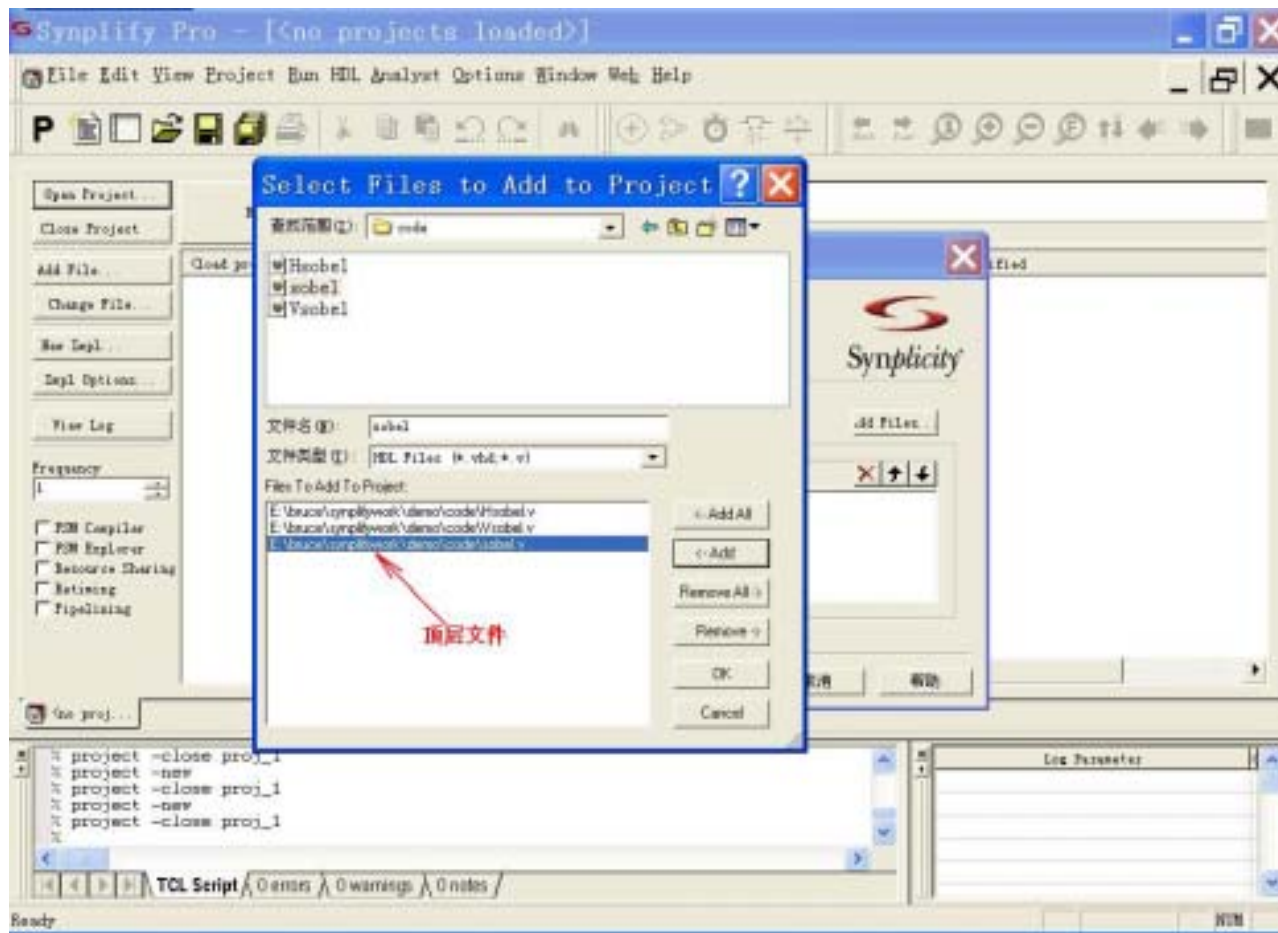
# 新建工程 (1)



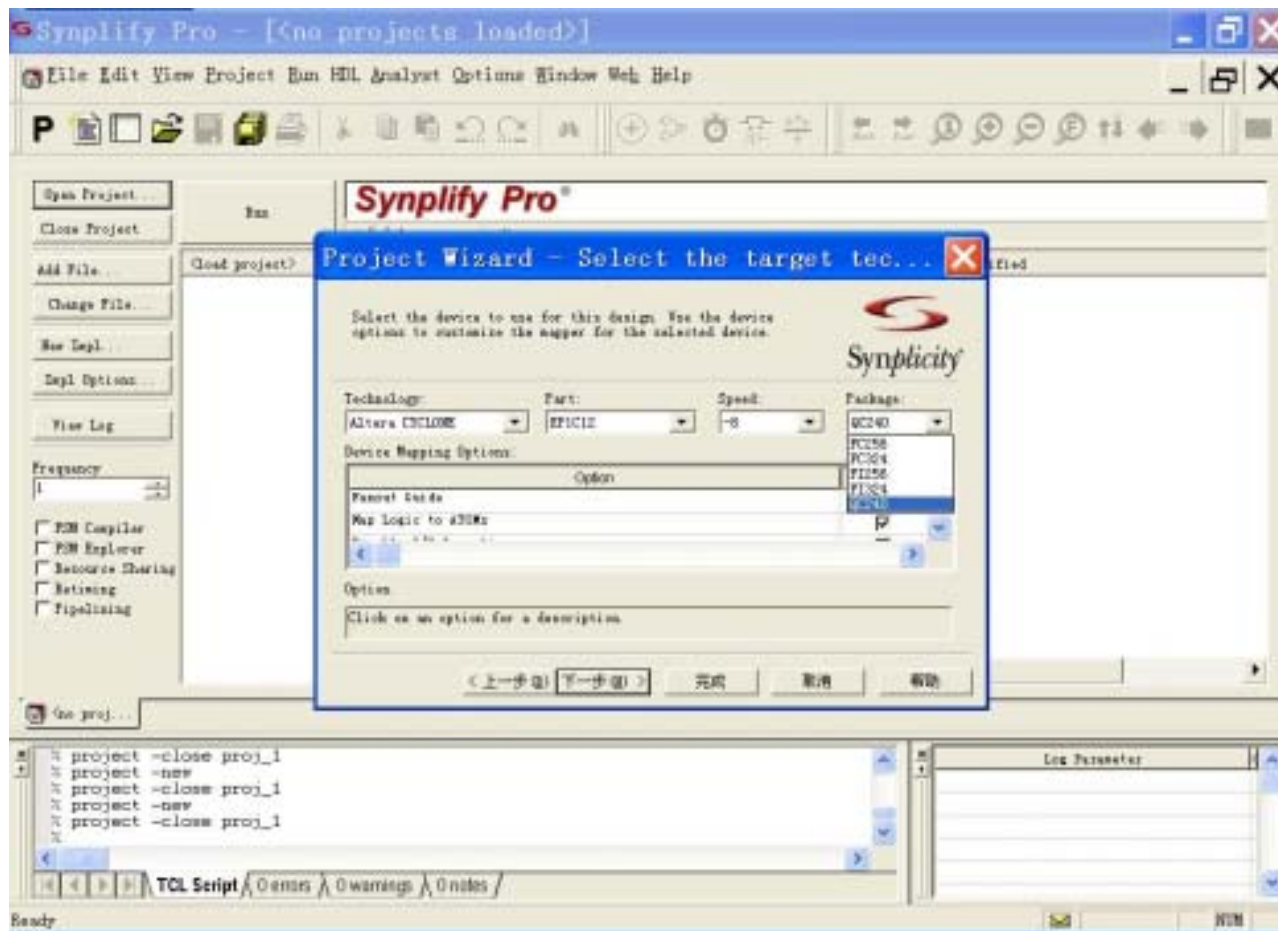
# 新建工程 (2)



# 新建工程 (3)



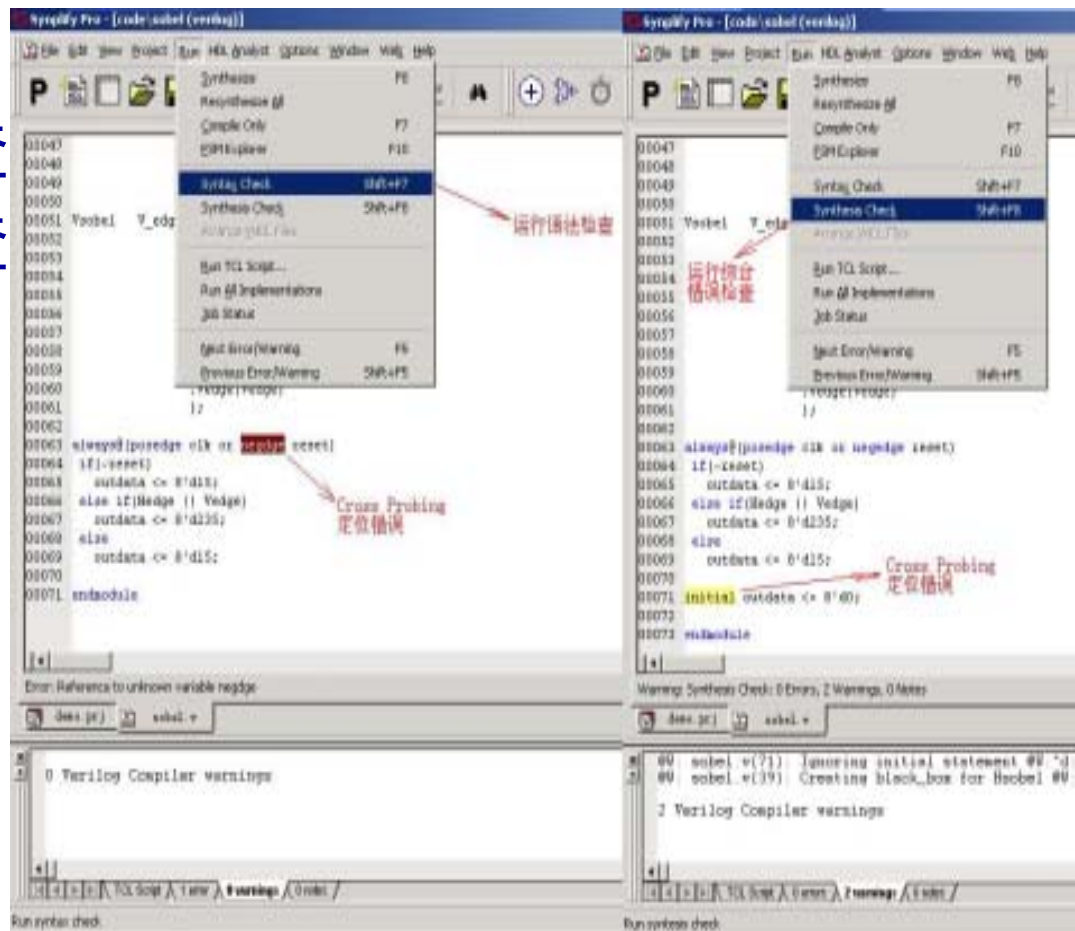
# 新建工程 (4)





# 代码错误检查

- 包括两个层次
  - 语法错误检查
  - 综合错误检查
- 推荐在代码调试阶段使用代码错误检查工具，及时发现问题



# 时序优化工具

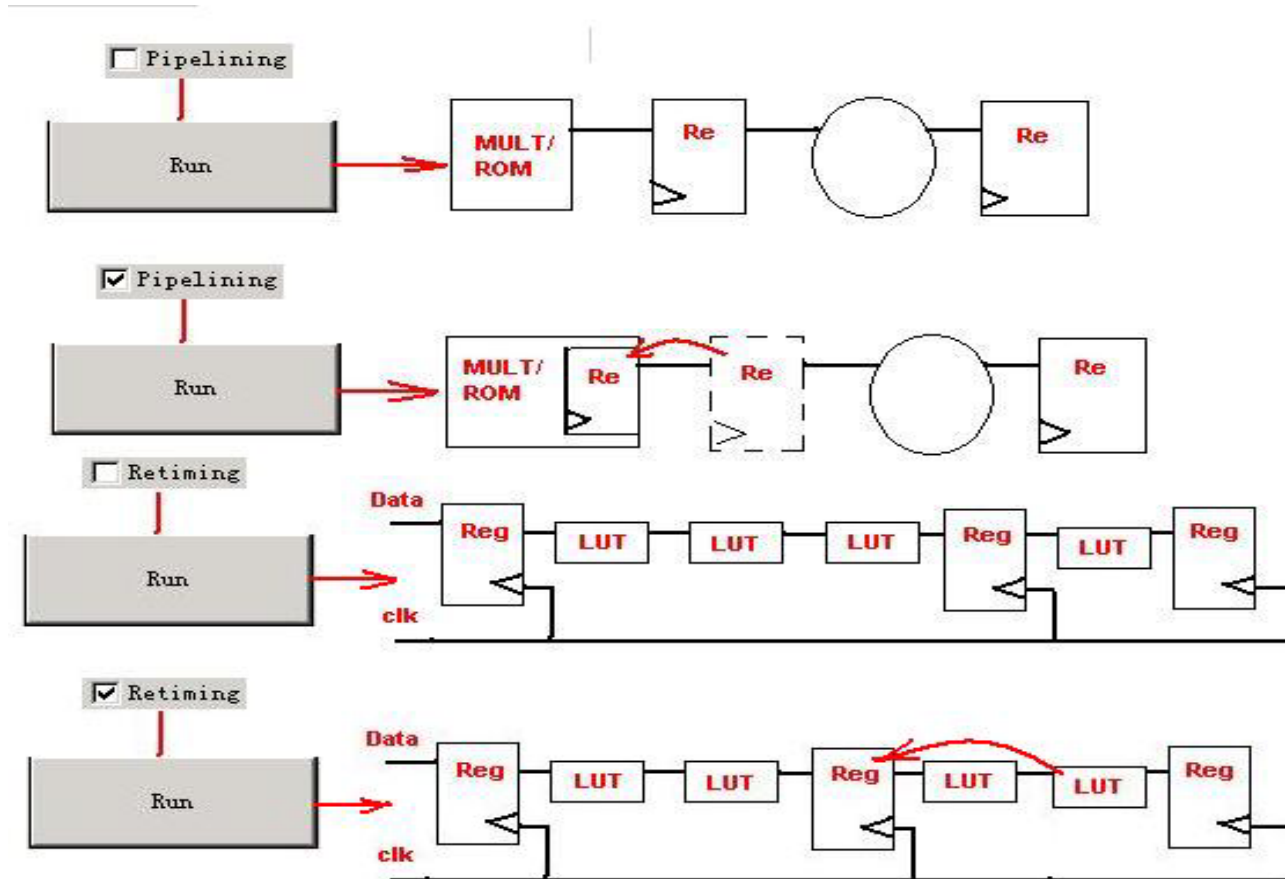
- **Pipelining**

- 将较大的组合逻辑用寄存器分割成若干较小的逻辑，减少从输入到输出的时延
- 自动优化乘法器，ROM等结构，提高工作频率

- **Retiming**

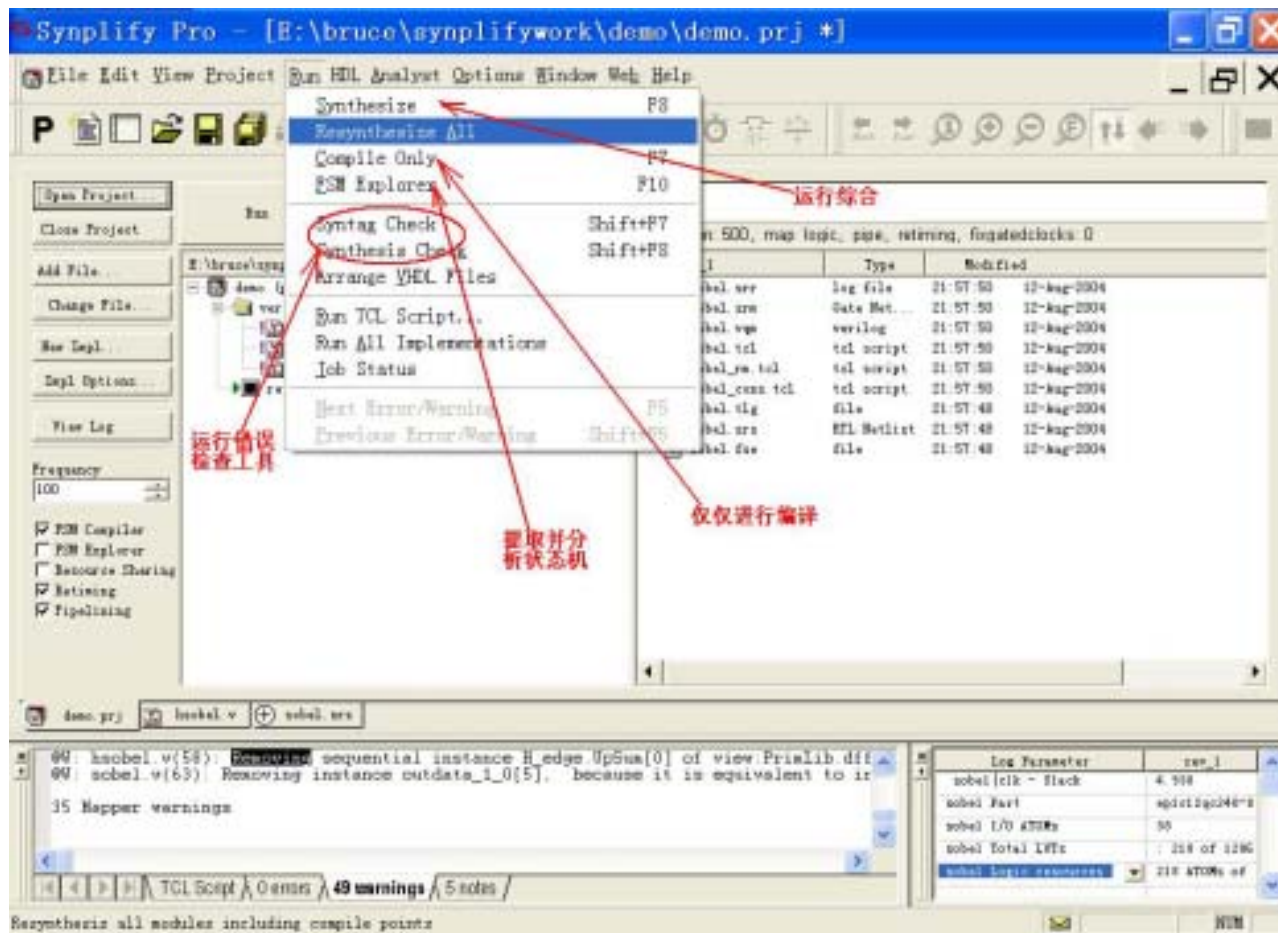
- 在不改变逻辑功能的前提下，自动用寄存器分割组合逻辑，在组合电路中插入平衡时延，提高芯片工作频率
- 本质是寄存器在宏观上的移动，不影响整体的寄存器级数
- Retiming的功能比Pipelining更强大，选定了Retiming就一定会自动选上Pipelining

# Retiming和Pipelining





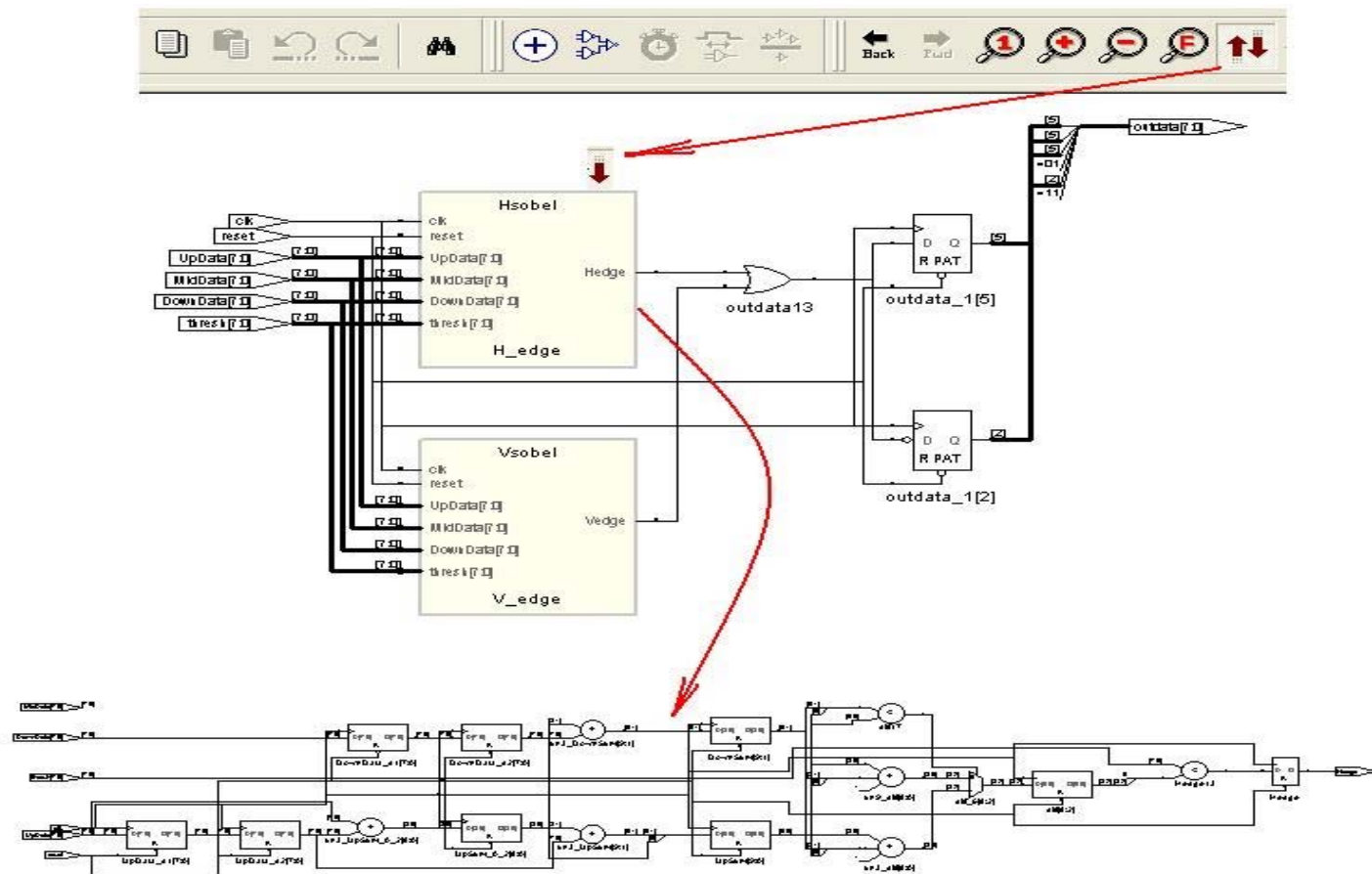
## 运行



## 分析运行结果

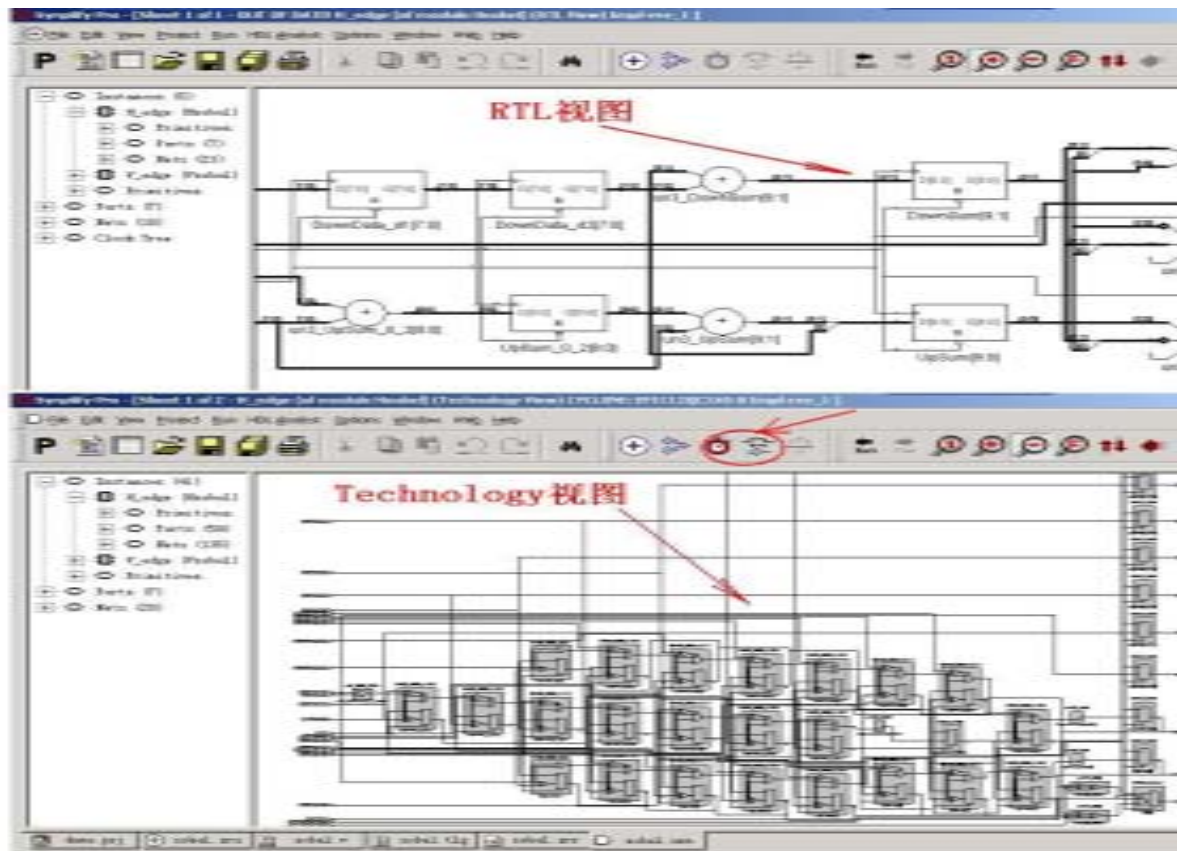


# PUSH/POP功能



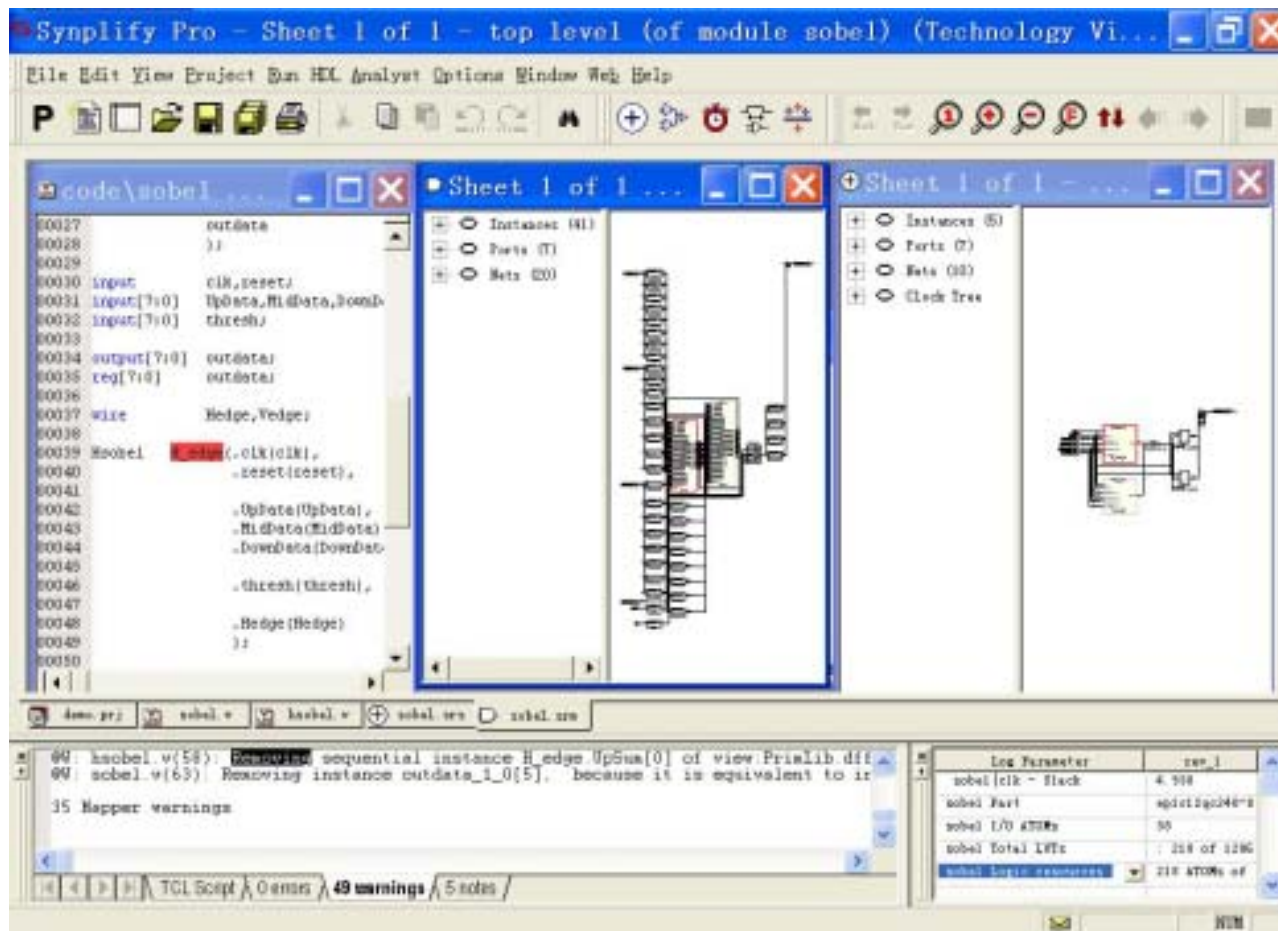
# RTL与Technology view

- Technology View是RTLView向具体器件进行结构映射的结果

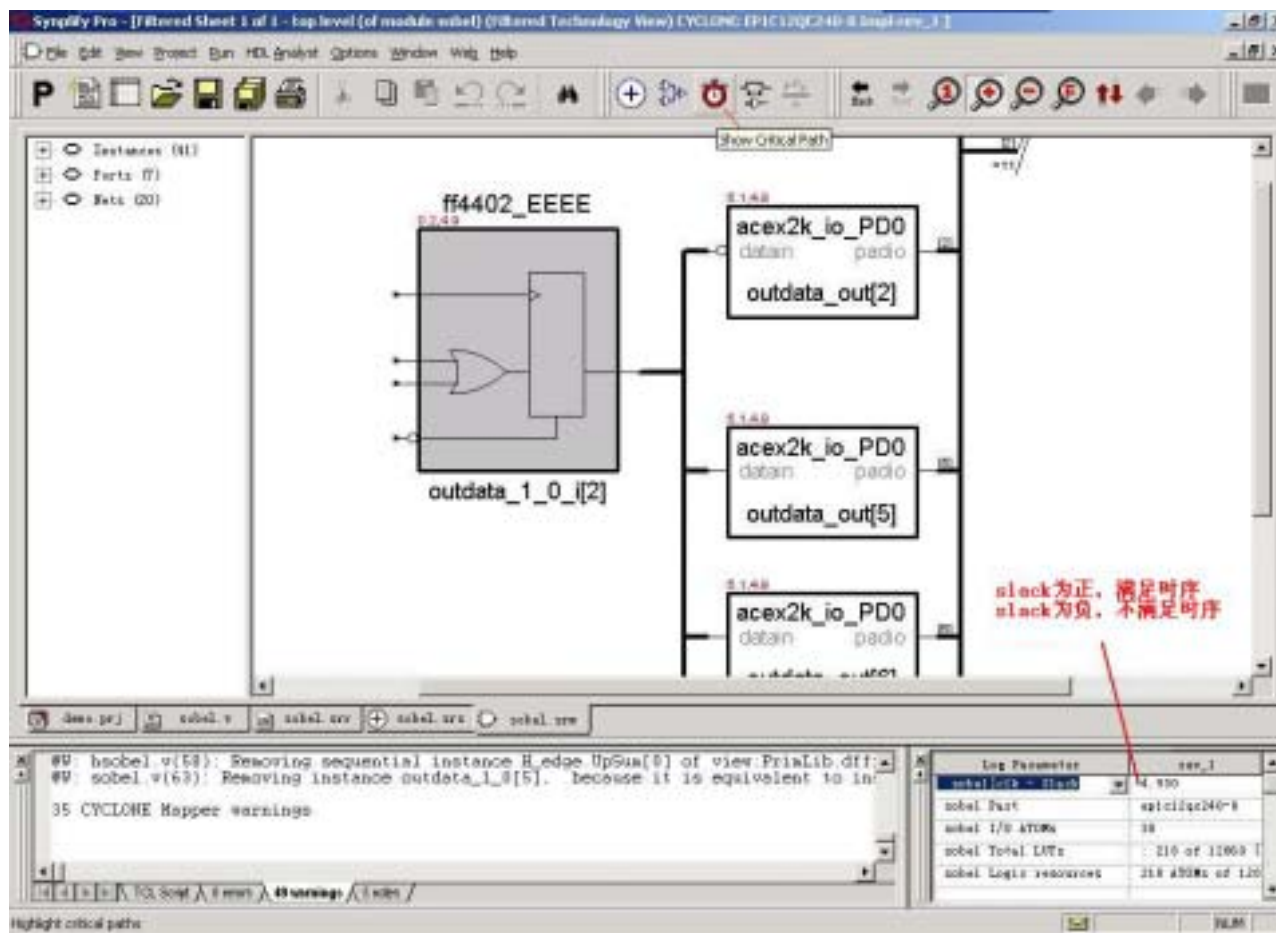




# Cross Probing功能



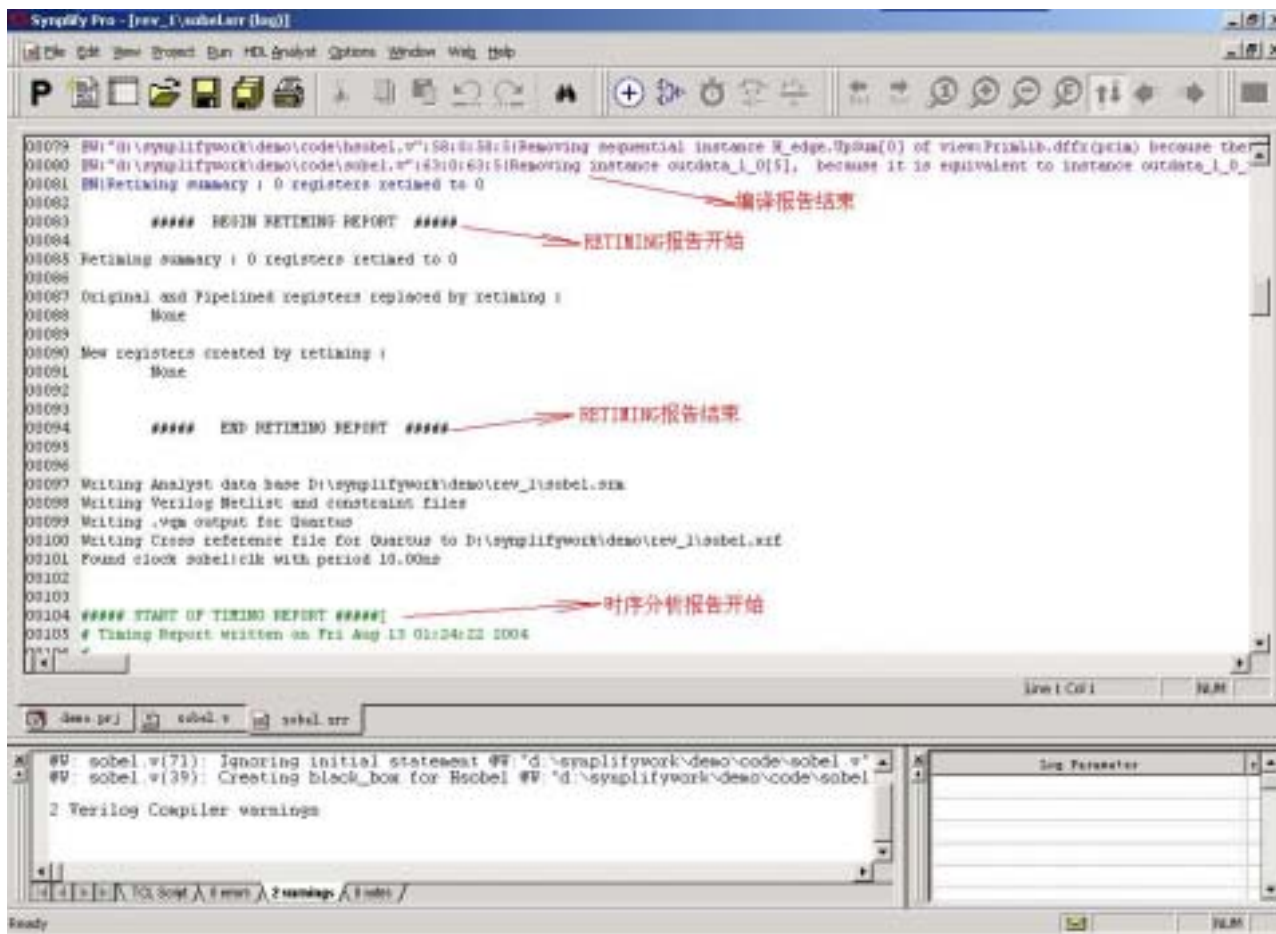
# 显示分析关键路径



# 查看综合报表

- 综合报表所含的信息

- 基本编译信息
- 有限状态机编译信息
- 时序约束信息
  - 总体性能概括 ( Performance Summary )
  - 时钟参数信息 ( Clock Relationships )
  - 输入输出引脚信息 ( Input Ports and Output Ports )
  - 较差时钟路径的起点和终点信息 ( Starting Points and Ending Points with Worst Slack )
- 组合电路时序优化信息
- 综合频率信息
- 面积信息 ( 资源消耗信息 )



The screenshot displays the Synplify Pro software interface. The main window shows a log of compilation and timing reports. Red arrows point to specific lines in the log, indicating key stages of the process:

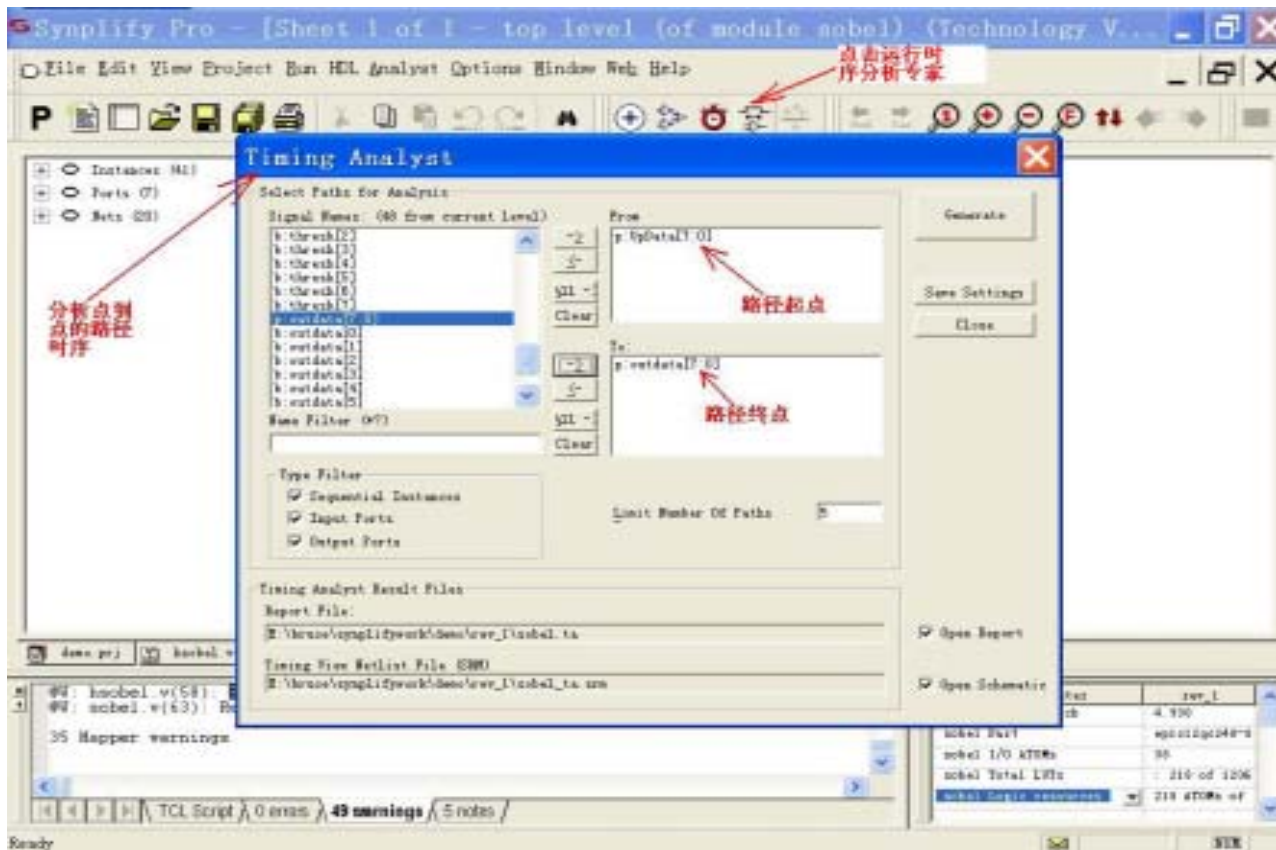
- 编译报告结束** (Compilation report ended) points to line 01000: "Removing instance outdata\_1\_0[5], because it is equivalent to instance outdata\_1\_0\_".
- RETIMING报告开始** (Retiming report started) points to line 01003: "#### BEGIN RETIMING REPORT ####".
- RETIMING报告结束** (Retiming report ended) points to line 01094: "#### END RETIMING REPORT ####".
- 时序分析报告开始** (Timing analysis report started) points to line 01104: "#### START OF TIMING REPORT ####".

The bottom status bar indicates "2 warnings / 1 notes /".

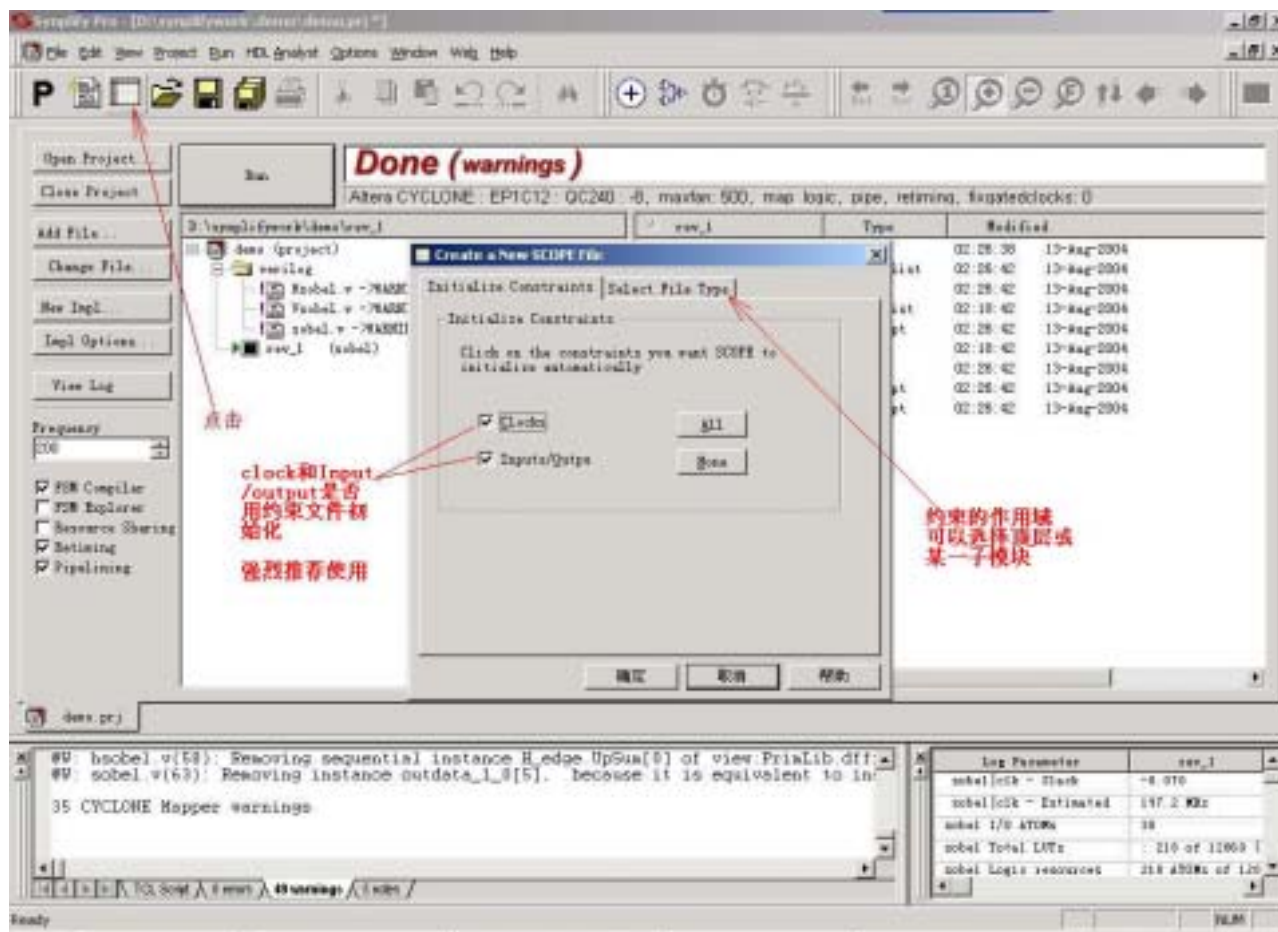


# 时序分析专家

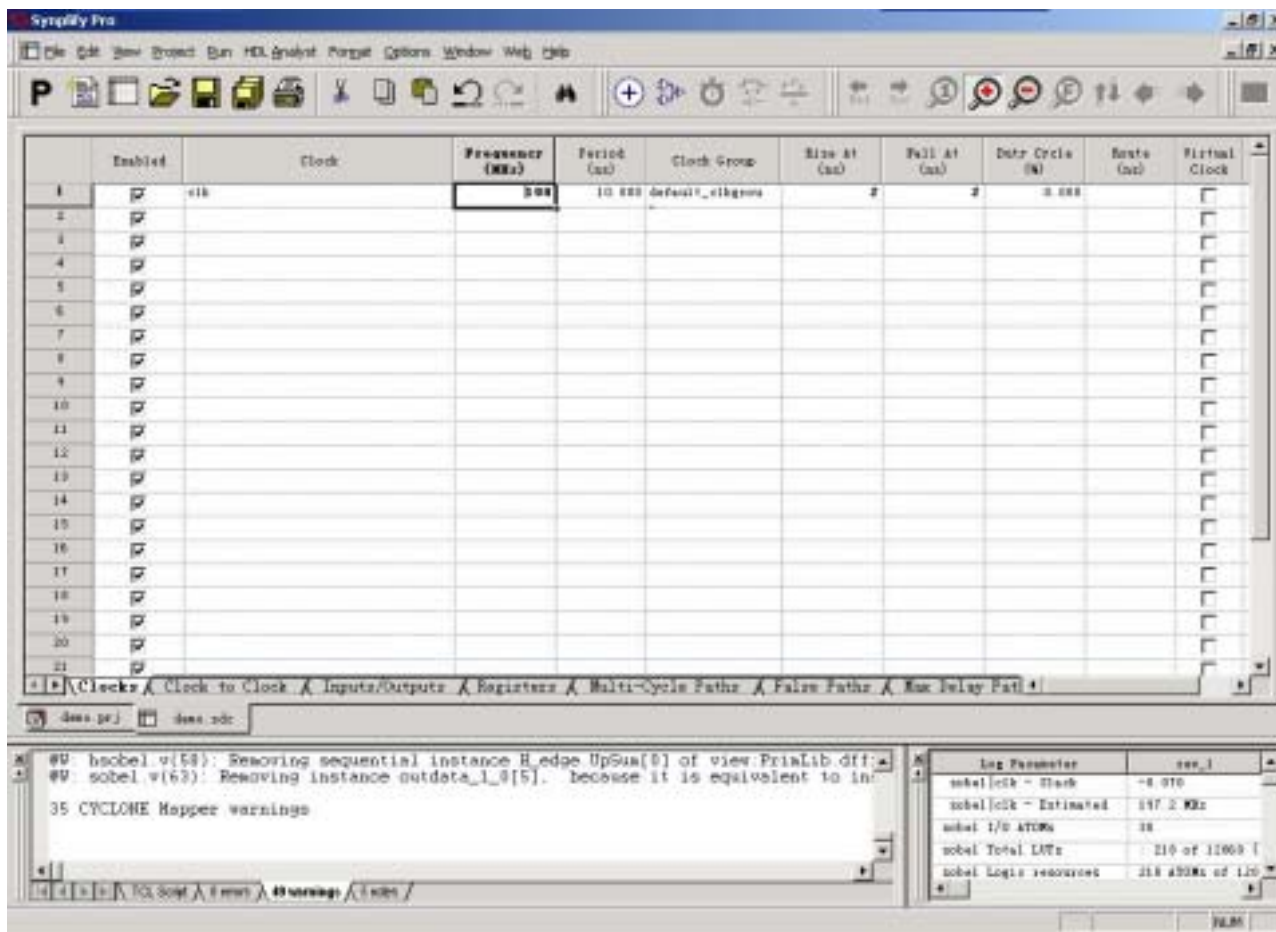
- 点到点的路径时序分析



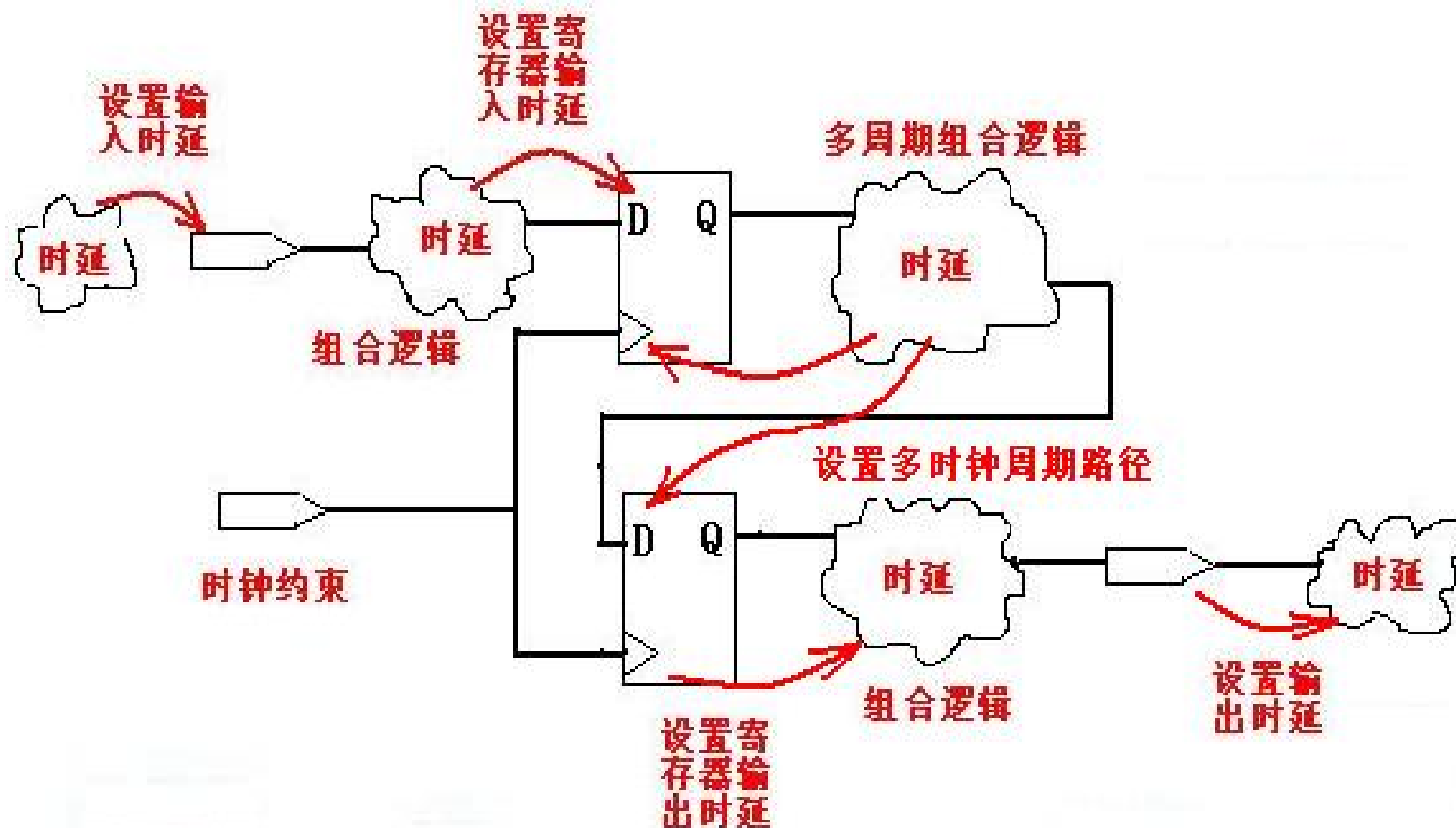
# 为设计添加约束 (SDC)



# 添加约束参数

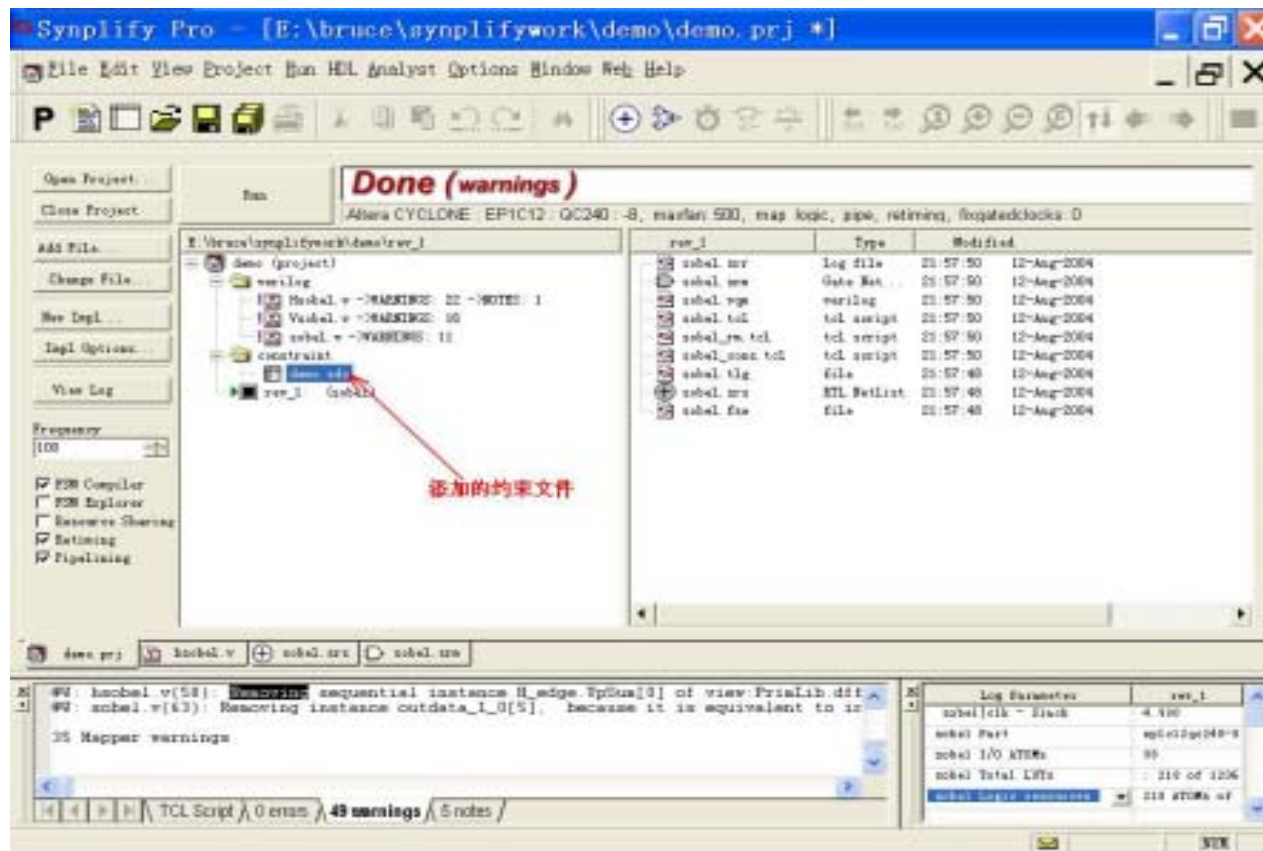


# 时间约束



# 添加的约束文件

- 保存SCOPE界面设置的参数，并选择将约束加于工程



*Thank you!*