ModelSim仿真入门

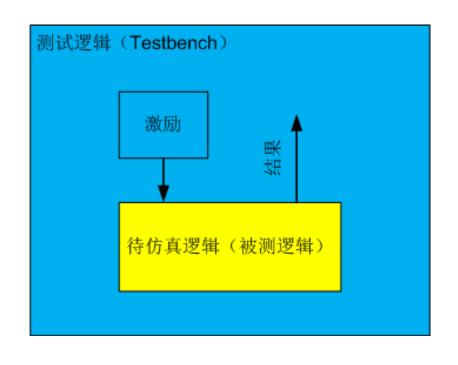
华中科技大学电工电子科技创新中心 电工电子工程基础Ⅱ

内容

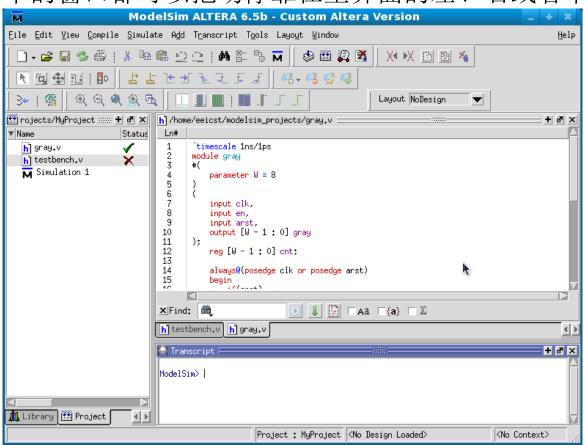
- 使用ModelSim仿真Verilog HDL
- 编写仿真平台(Testbench)
- 编译和仿真
- 查看仿真结果

ModelSim介绍

- 为什么要仿真
 - 复杂的设计几乎不可能写完功能就是对的,因此需要在投入使用前仿真验证
 - 仿真分为功能仿真(一般即RTL级)和时序仿真(一般即Gate级)
- 仿真的概念
 - 通过产生一定的激励信号 送给被测逻辑,并观察被 被测逻辑的行为(包括输 出和内部行为),来判断 被测逻辑是否满足功能需 求以及时序需求的过程。



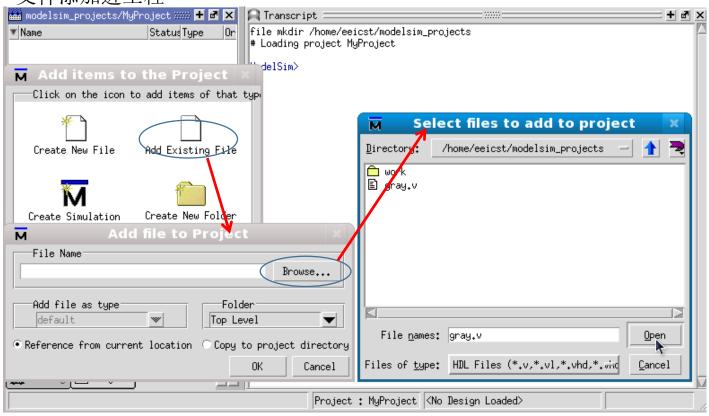
- ModelSim的界面
 - 界面中的窗口都可以拖动停靠在主界面的左、右或者下方。



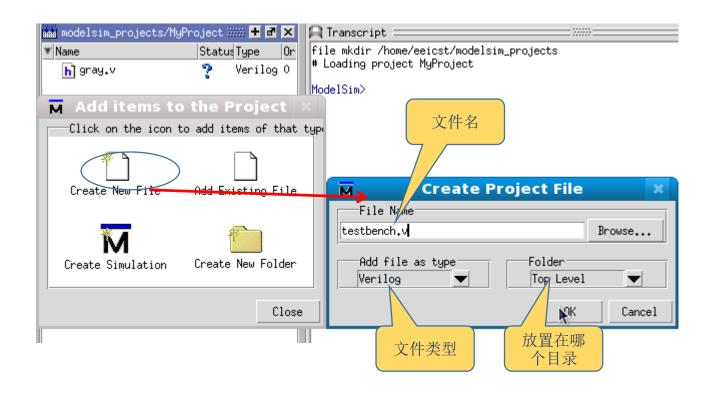
- 建立工程
 - 新建工程
 - File New Project...



- 为工程添加文件
 - 将待仿真的文件复制进刚刚建立的目录内,并"Add Existing File",将待仿真的文件添加进工程

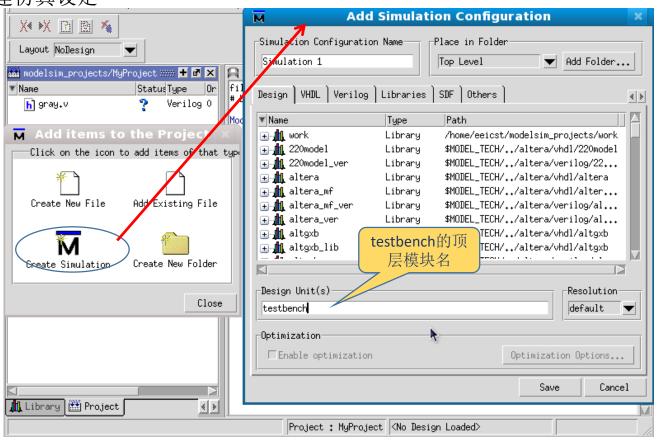


- 为工程添加文件
 - 创建用于书写Testbench的源文件,"Create New File"



• 为工程添加文件

• 创建仿真设定



编写Testbench

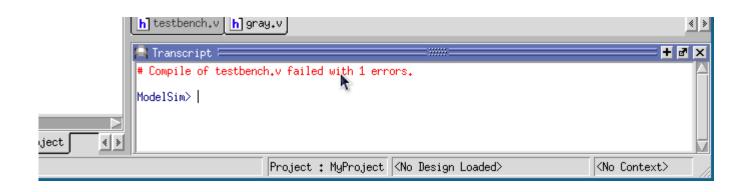
- 编写testbench
 - Testbench中可以使用全部的verilog语法,包括:
 - #(延时)
 - initial
 - 各种系统函数, \$stop, \$readmemh等等
 -
 - 在ModelSim中做功能仿真时,需要注意:
 - 所有寄存器型变量(如reg、integer)的仿真初始值均为不确定值"X",如果你的模块中含有不会在运行期间被赋予确定值的reg,那么在ModelSim中它会一直保持为"X",这是与实际器件不符的,因为实际器件中不会出现"X"。

这些语法是不可 综合进FPGA的

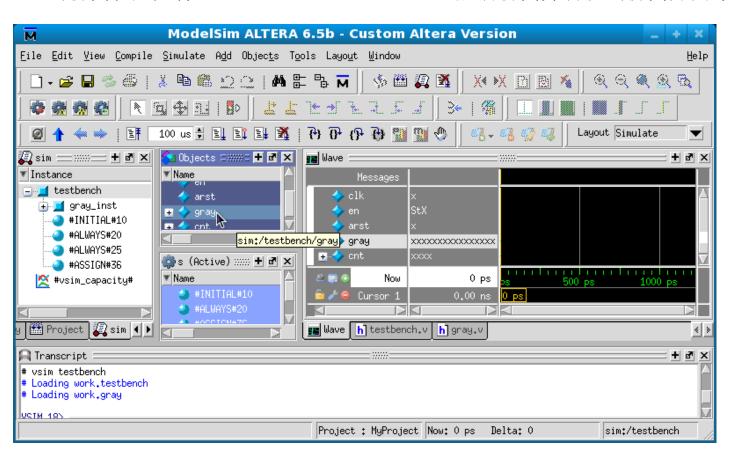
编写Testbench

```
个Testbench解析
timescale 1ns/1ps
                                             时间单位/时间精度
module testbench:
   reg clk, arst;
                                              仿真的顶层模块
   wire en;
   wire [15 : 0] gray;
                                           申明需要的寄存器和线网
   reg [3 : 0] cnt;
   initial begin
                                               Initial, 初始化
       gray_inst.cnt <= 1'b0;</pre>
       clk <= 1'b1;
                                          初始化gray inst中的寄存器
       arst <= 1'b0;
                                          在1234ns处产生一个宽度为
       cnt <= 1'b0;
       #1234 arst <= 1'b1;
                                              1234ns的arst信号
       #1234 arst <= 1'b0;
   end
   always begin
                                            产生周期为20ns的时钟
       #10 clk \leftarrow ~clk;
   end
   always@(posedge clk) begin
                                         模为10的计数器cnt,用于驱
       if(cnt < 4'd9) begin
           cnt <= cnt + 1'b1;</pre>
                                          动en信号每10个周期有效
       end else begin
           cnt <= 1'b0;
       end
   end
   assign en = (cnt == 4'd0);
                                             调用被测试的模块
   gray #(.W(16)) gray_inst(
       .clk(clk),
       .en(en),
       .arst(arst),
       .gray(gray));
endmodule
```

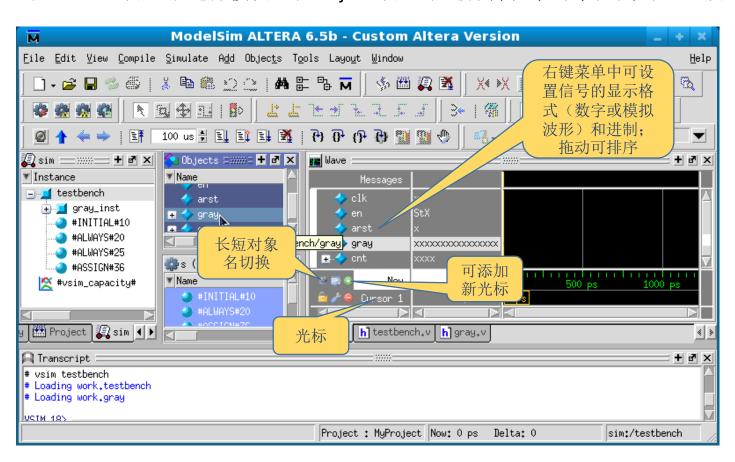
- 编译
 - 在project视图中右键单击,选择菜单 Compile Compile All
 - 如果编译有错误,会在脚本窗口中出现红色提示,双击它可以定位错误



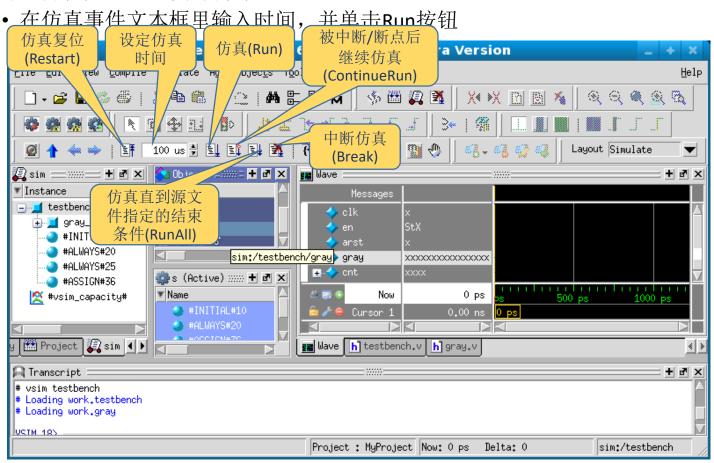
- 启动仿真
 - 双击仿真设定文件 "Simulation 1", ModelSim启动仿真并切换到仿真界面布局



- 添加待观察信号
 - 在Instance窗口中选择模块,在Objects窗口中选择待观察对象拖拽到Wave窗口中。

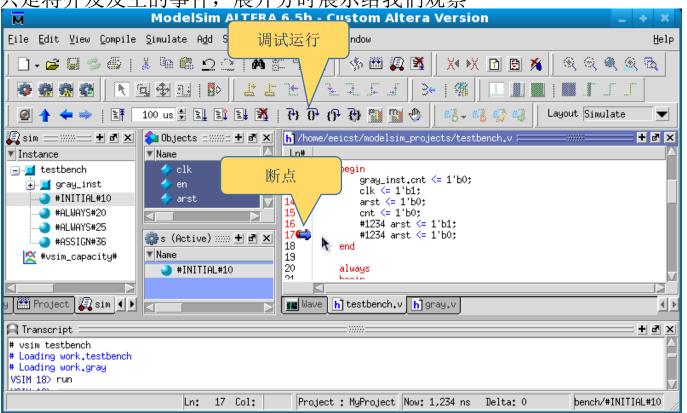


• 设定仿真时间,并仿真



• 调试

- 可在源文件中设定断点, 仿真至断点时, 会暂停
- · 需要注意: HDL描述的逻辑是并发的,虽然ModelSim可以单步,但应理解为,它只是将并发发生的事件,展开分时展示给我们观察



查看结果

- 仿真开始之后,便可以看到Wave窗口中出现波形
- 如果发现错误需要修改源文件:

