

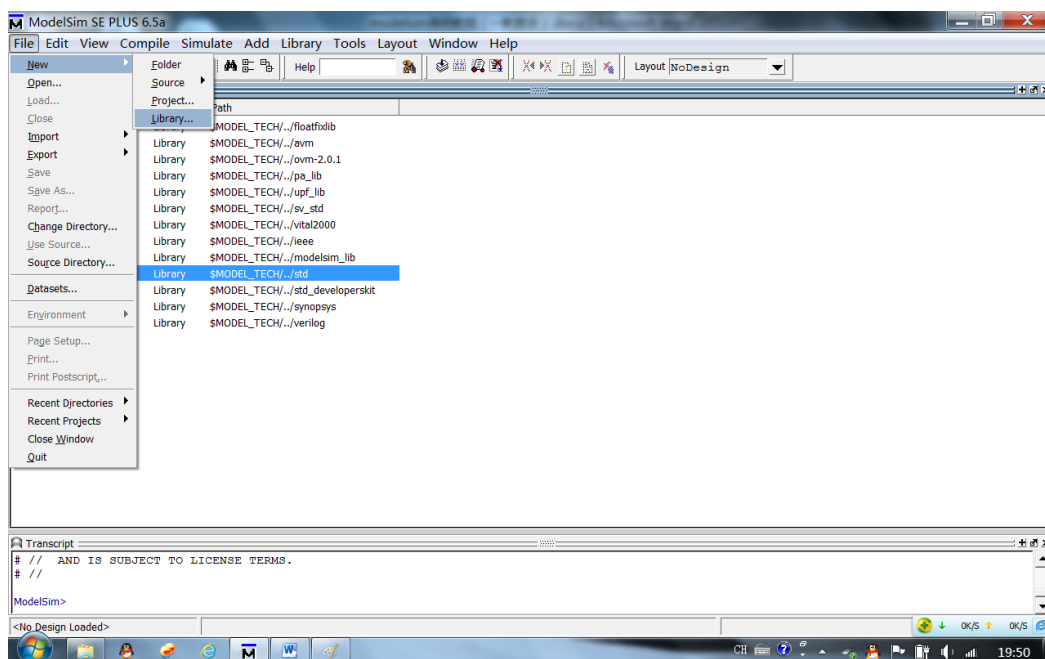
Modelsim 详细使用方法

很多的 modelsim 教程中都讲得很丰富，但忽视了对整个仿真过程的清晰解读，而且都是拿 counter 范例举例子，有些小白就不会迁移了。这里我们着眼于能顺利的跑通一个自己写的程序，一步一步的讲解，如果你是一个初学者，这再适合你不过了，虽然貌似字写得比较多，那是因为写得相当的详细，一看就会啦 O(∩_∩)O~

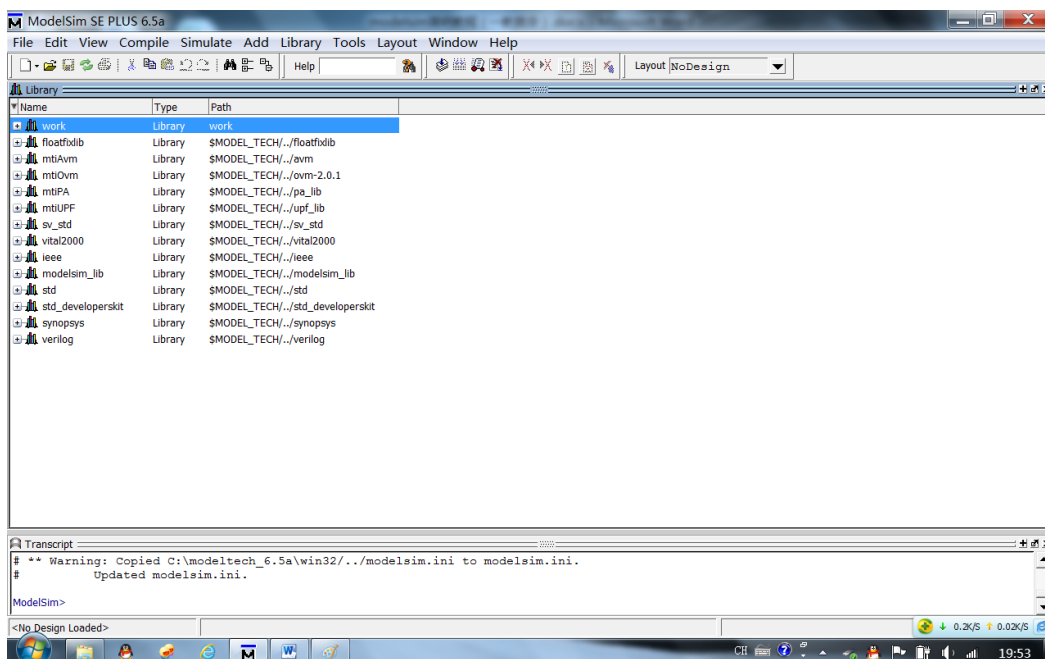
一、 建立工程

- 1、在建立工程（project）前，先建立一个工作库（library），一般将这个 library 命名为 work。尤其是第一次运行 modelsim 时，是没有这个“work”的。但我们的 project 一般都是在这个 work 下面工作的，所以有必要先建立这个 work。

File→new→library

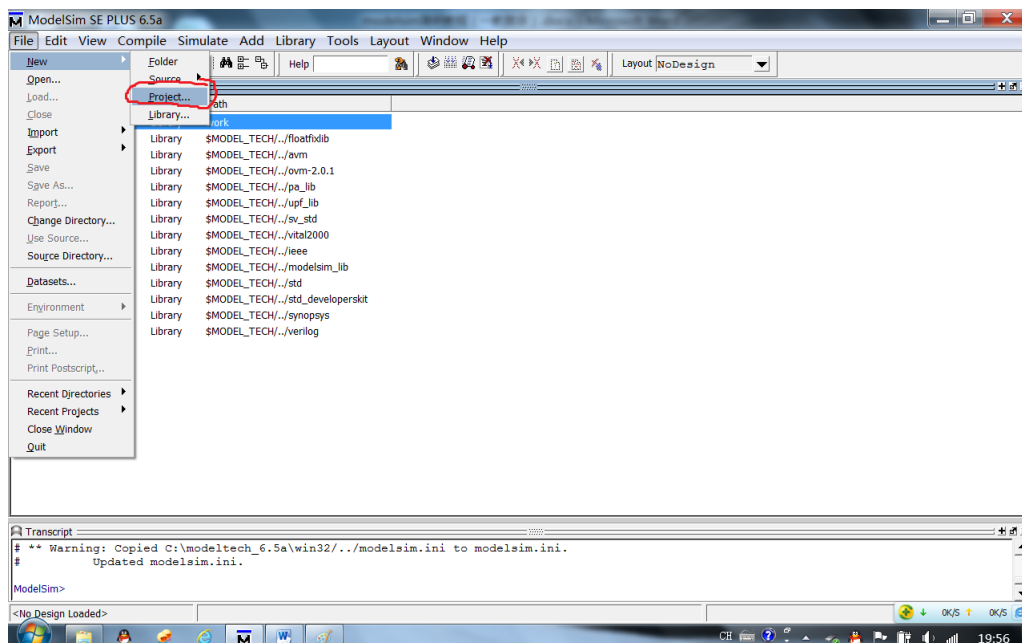


点击 library 后会弹出一个对话框，问是否要创建 work，点击 OK。就能看见 work.

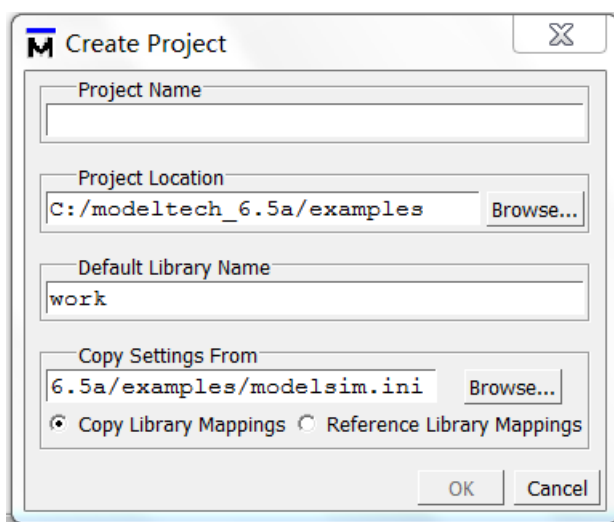


2、如果在 library 中有 work，就不必执行上一步骤了，直接新建工程。

File→new→project

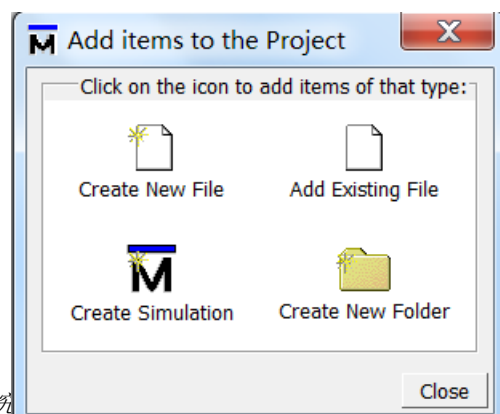


会弹出



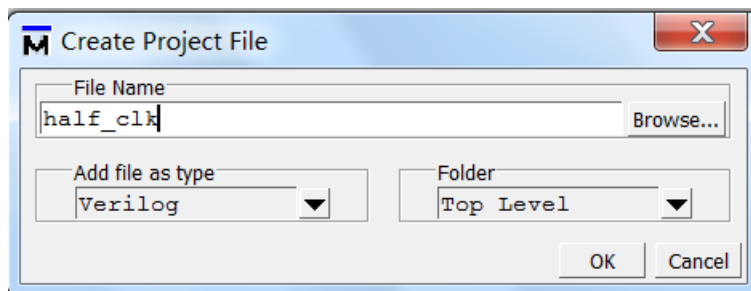
在 Project Name 中写入工程的名字，这里我们写一个二分频器，所以命名 half_clk,然后点击 OK。

会出现

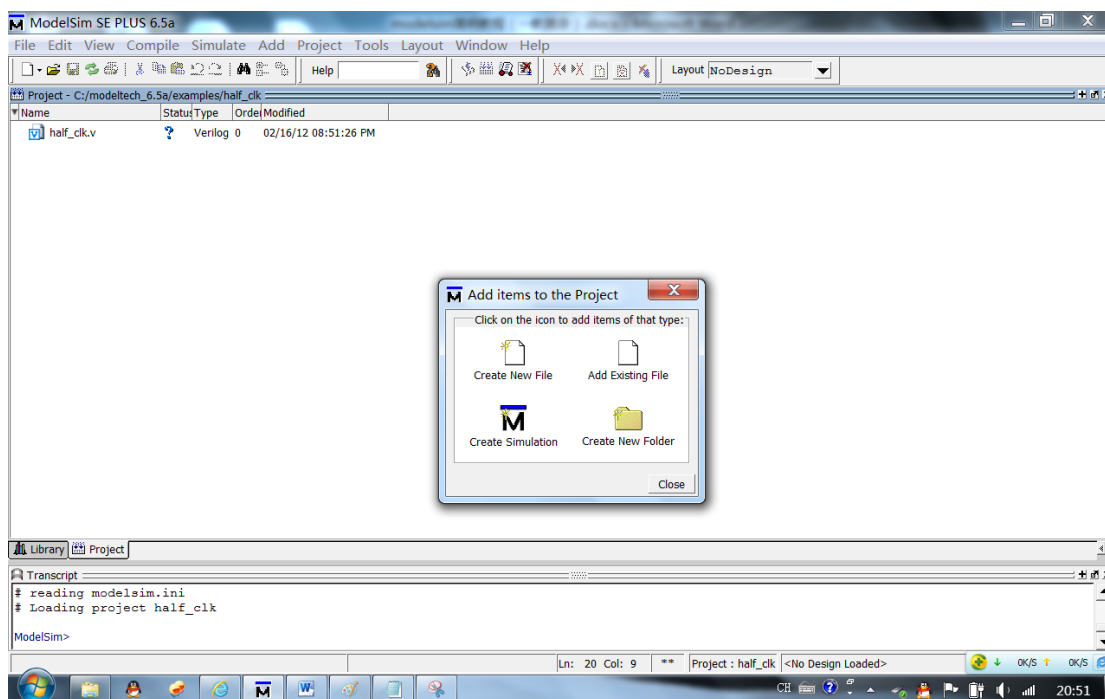


版权所有，翻录必究

由于我们是要仿一个自己写的程序，所以这里我们选择 Create New File。



在 File Name 中写入文件名（这里的 file name 和刚刚建立的 project name 可以一致也可以不一致）。注意 Add file as type 要选择成 Verilog（默认的是 VHDL），然后 OK。



发现屏幕中间的那个对话框没有自己消失，我们需要手动关闭它，点 close。

并且在 project 中出现了一个 half_clk.v 的文件，这个就是我们刚刚新建的那个 file。

这样工程就建立完毕了。

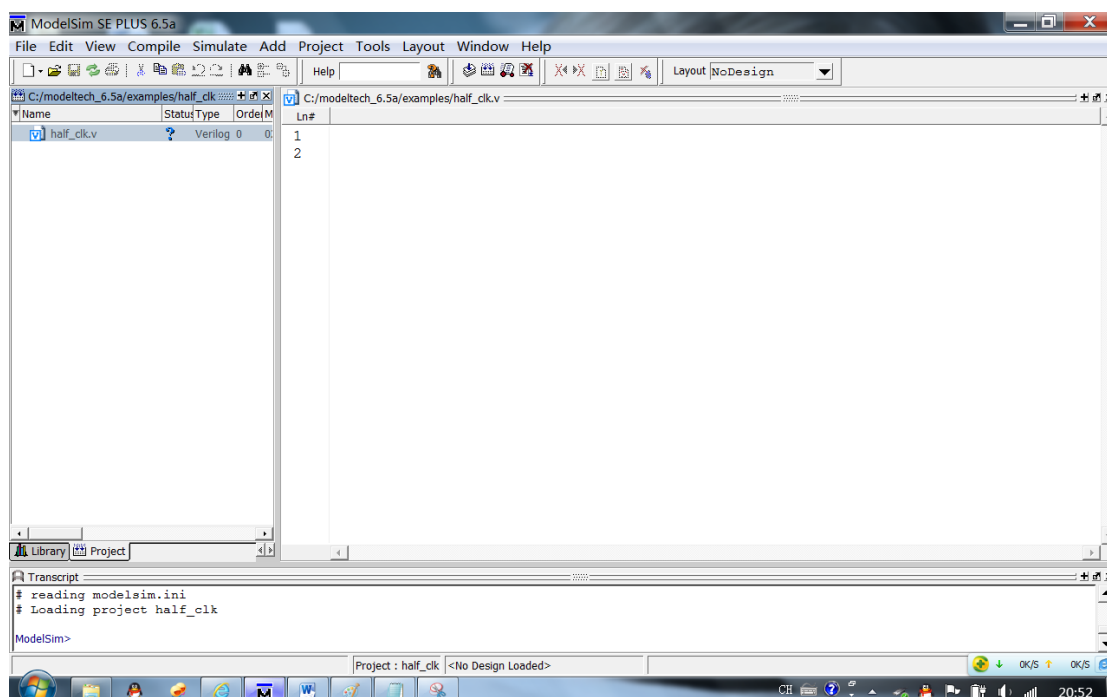
二、 写代码：

- 1、 写主程序：双击 half_clk.v 文件会出现程序编辑区，在这个区间里写好自己的程序，这里我们写一个简单的二分频的代码：

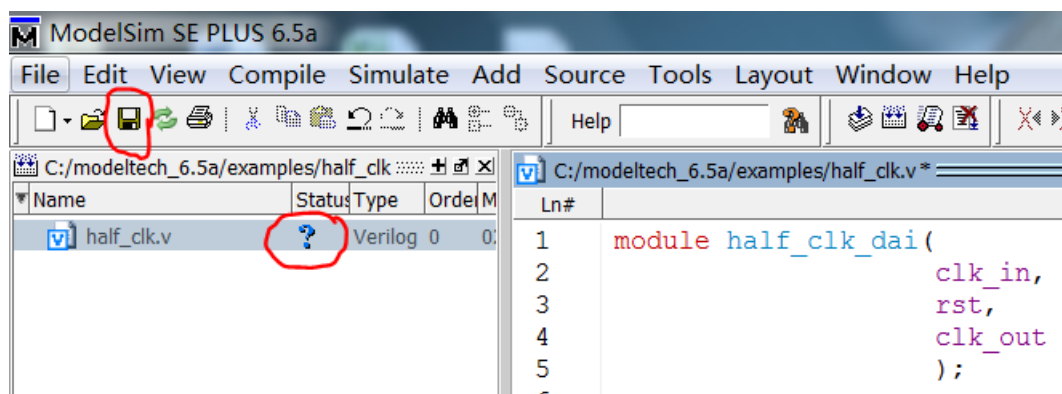
```
module half_clk_dai(  
    clk_in,  
    rst,  
    clk_out  
);  
  
input clk_in;  
input rst;
```

```
output clk_out;  
reg clk_out;  
  
always @(posedge clk_in or negedge rst)  
begin  
    if(!rst)  
        clk_out<=0;  
    else  
        clk_out<=~clk_out;  
    end  
end
```

endmodule

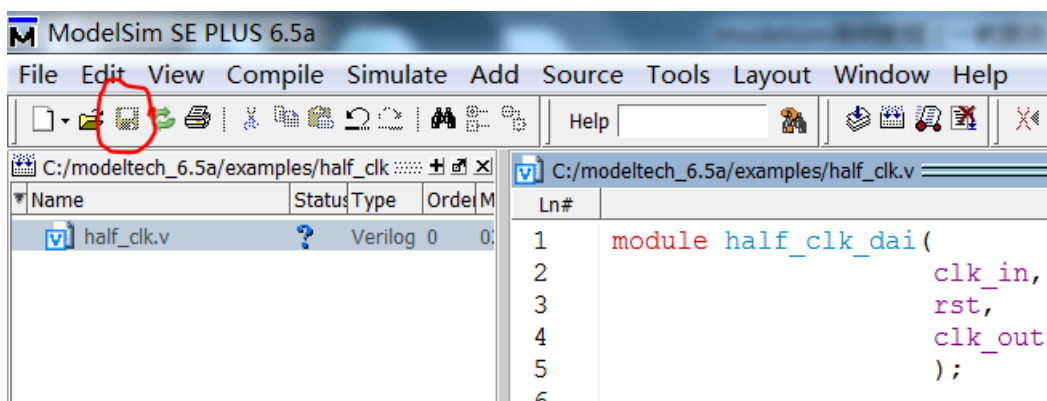


写完代码后，不能马上就编译，要先保存，否则，编译无效。我们会看到“保存”的图标是两色的，而且 half_clk.v 后的“？”还存在，这说明你没有保存文件，



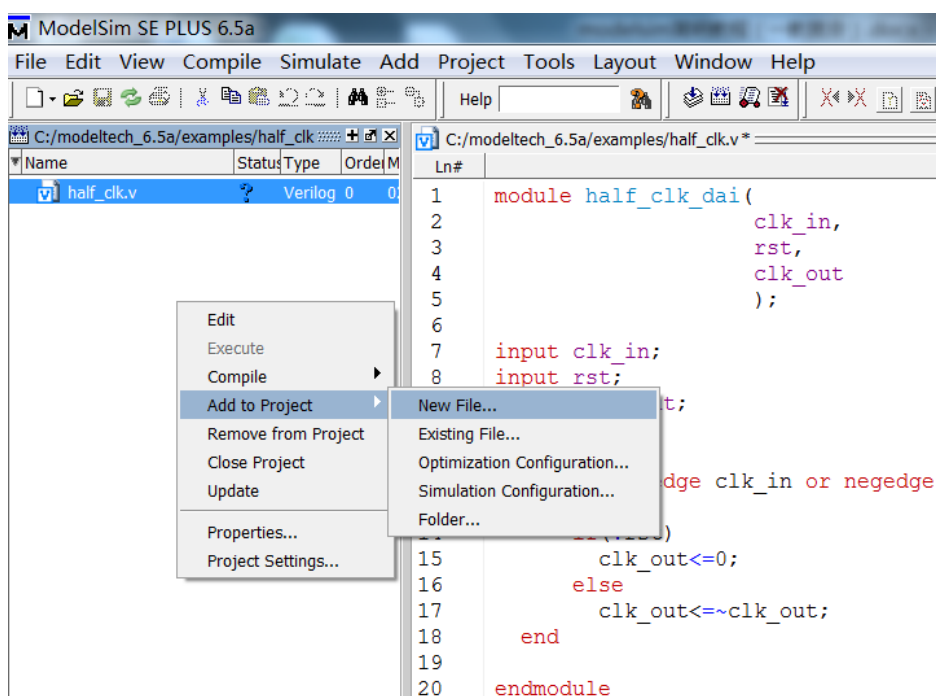
这时，只需要点击保存，就会发现“保存”的图标变成了阴影，这样编译才有效，而就是这么不起眼但至关重要的一步“保存”，往往被很多初学者忽视，最终怀疑程序和

软件问题。其实只是没保存。

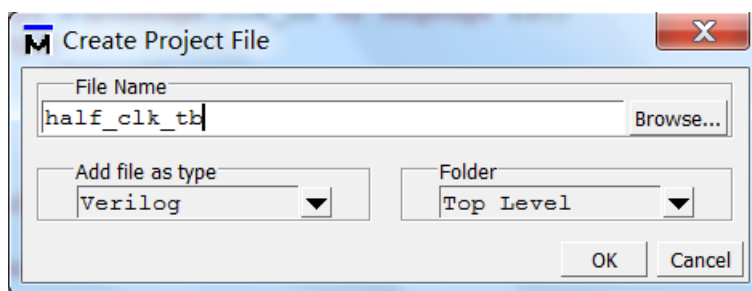


2、写测试程序 (testbench): 每一个主程序 (实现我们需要的某种功能的程序), 都要配套的编写一个测试程序, 为什么? 这里不打算细说, testbench 是给主程序提供时钟和信号激励, 使其正常工作, 产生波形图, 具体请参看 verilog 教课书。

在 half_clk.v 下方的空白区域内右键, 选择 Add to Project → New File



出现下框, 写入测试程序的名字 half_clk_tb, tb 是 testbench 的意思, 注意选 verilog, Ok。



这样就把 half_clk_tb.v 加载到了 project 中, 双击 half_clk_tb.v 在右边的程序编辑区中编写代码:

```
`timescale 1ns/1ns
```

```
module half_clk_top;

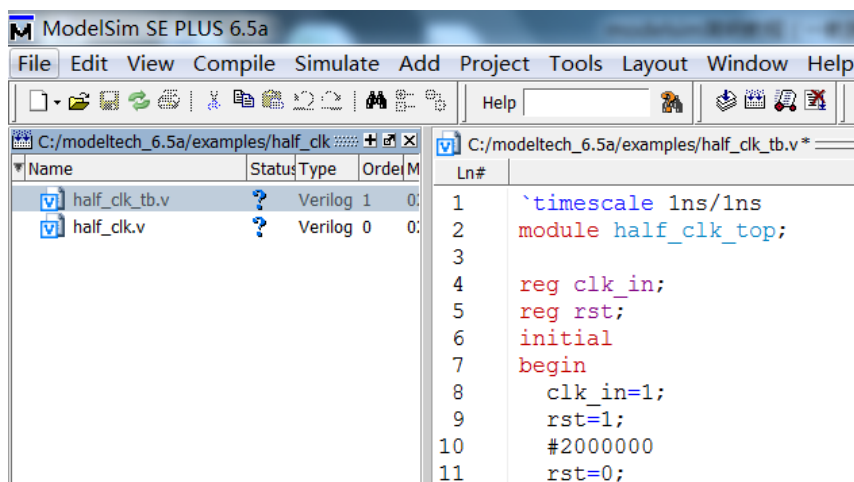
reg clk_in;
reg rst;

initial
begin
    clk_in=1;
    rst=1;
    #1000
    rst=0;
    #1000
    rst=1;
end

always #200 clk_in=~clk_in;

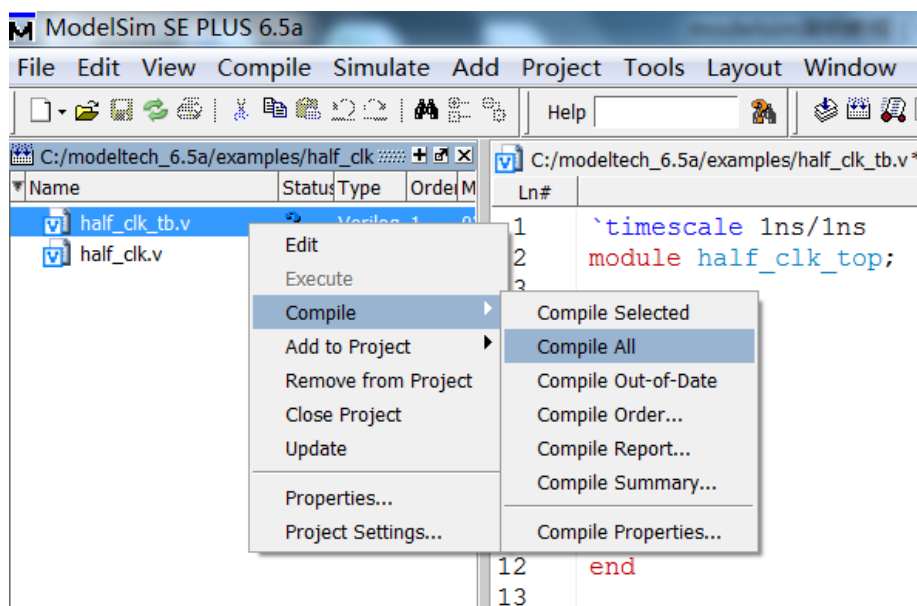
half_clk_dai dai1(
    .clk_in(clk_in),
    .rst(rst),
    .clk_out(clk_out)
);

endmodule
```

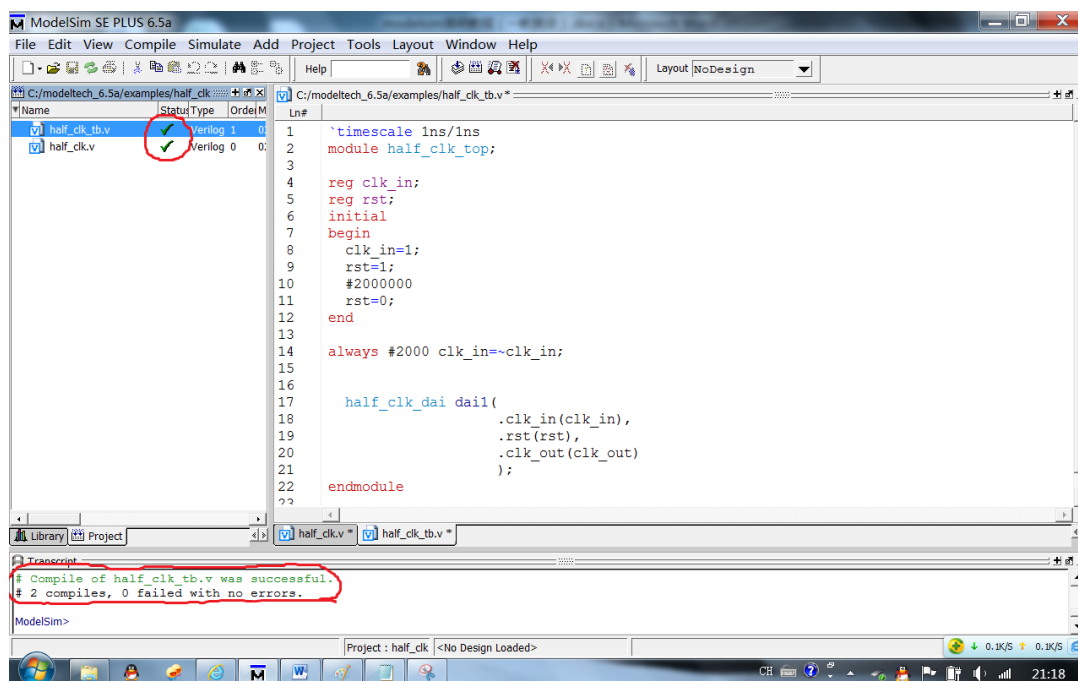


三、编译代码

在 half_clk.v 的文件上点右键，选择 Compile，至于是 Compile All 还是 Compile Selected 都可以，就看自己选择了。然后点击。

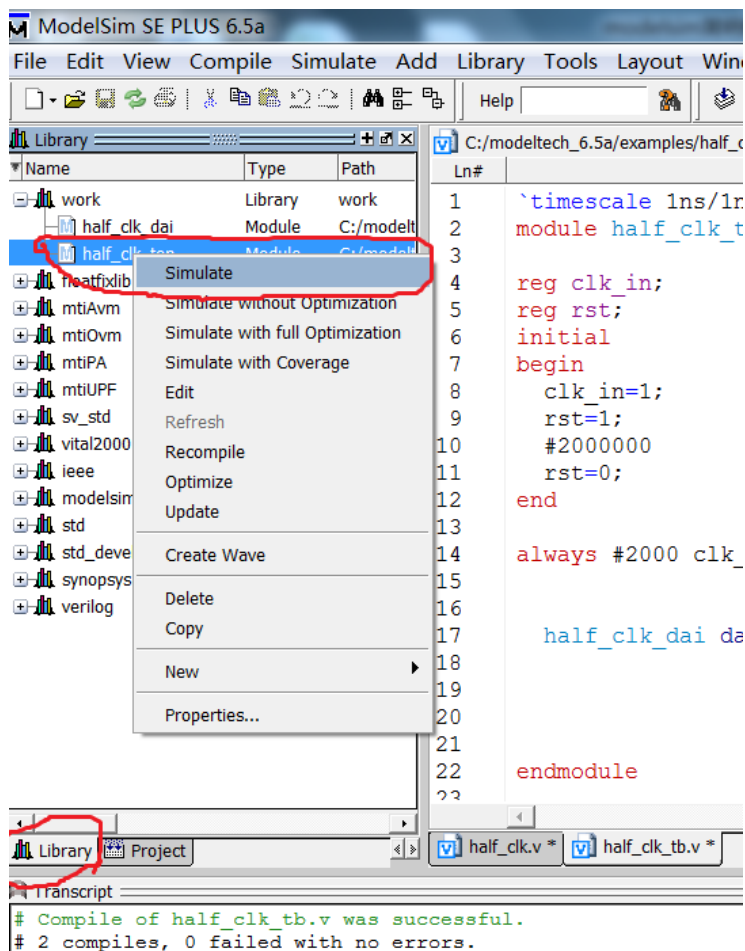


编译成功后，half_clk.v 和 half_clk_tb.v 后面的? 变成了对勾，并且在最下方的 Transcript 栏中出现了 successful 字样，说明编译成功，否则会报错，就要回到程序中修改，只有编译成功后，才能往后面进行。

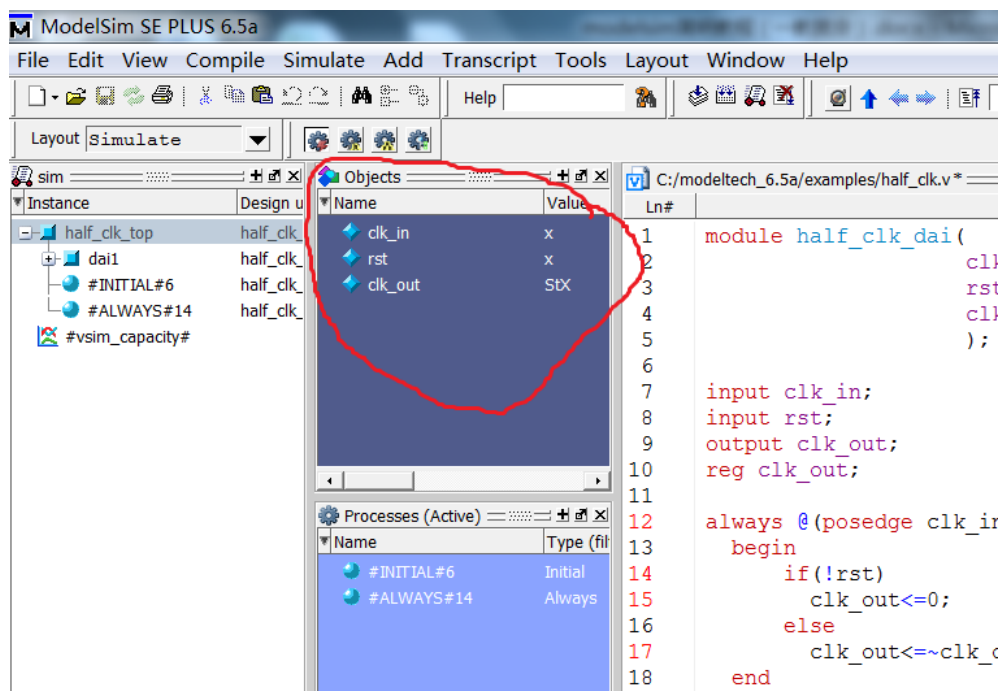


四、 仿真

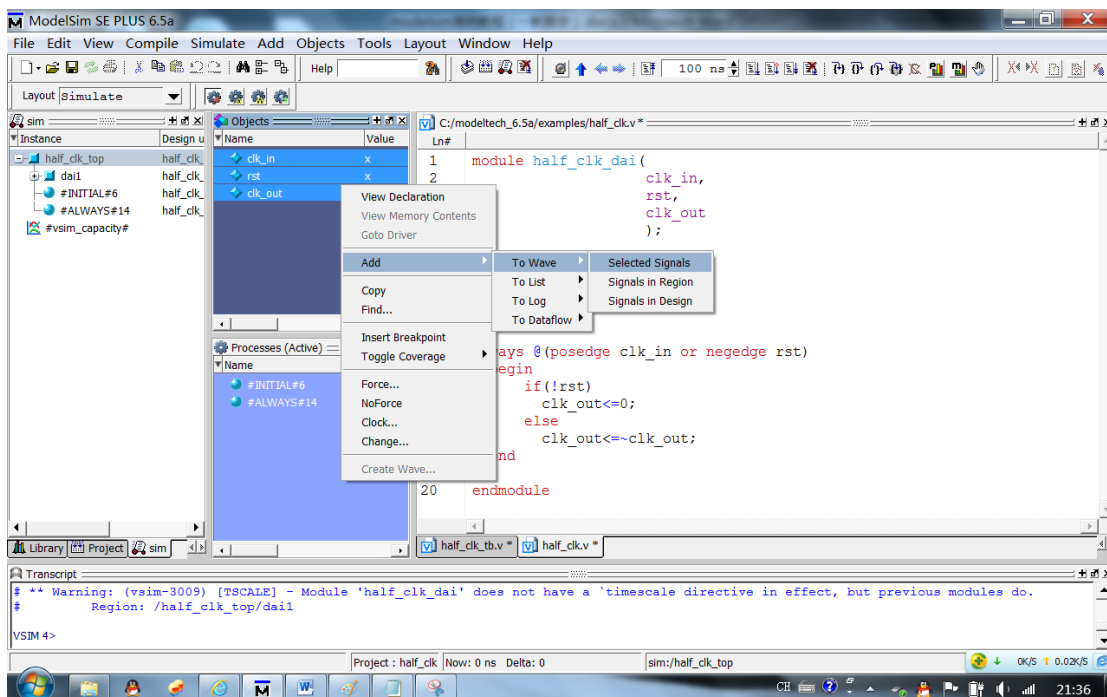
在屏幕左下角的位置有一个 library 和 project 的切换窗口，点击 library，再点击 work 前的“+”号，将其展开，会看到两个文件，文件名是我们刚刚写的 half_clk.v 和 half_clk_tb.v 两个文件中的 module 名。仿真不用两个文件都 simulate，只需 simulate 测试文件即可，我们选择 half_clk_top 右键，选择 simulate。



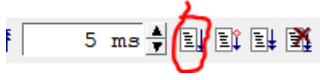
之后会出现 objects 框:

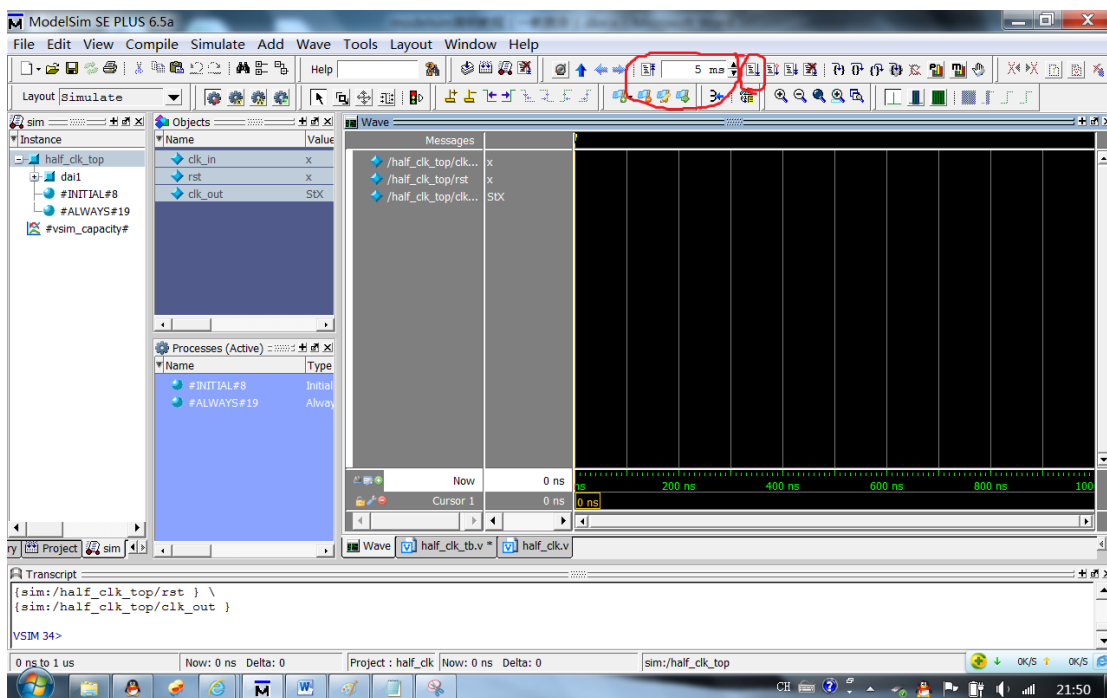


按住 Ctrl 键选中 clk_in,clk_out,rst 三个信号, 右键选择 Add→To Wave→Selected Signals

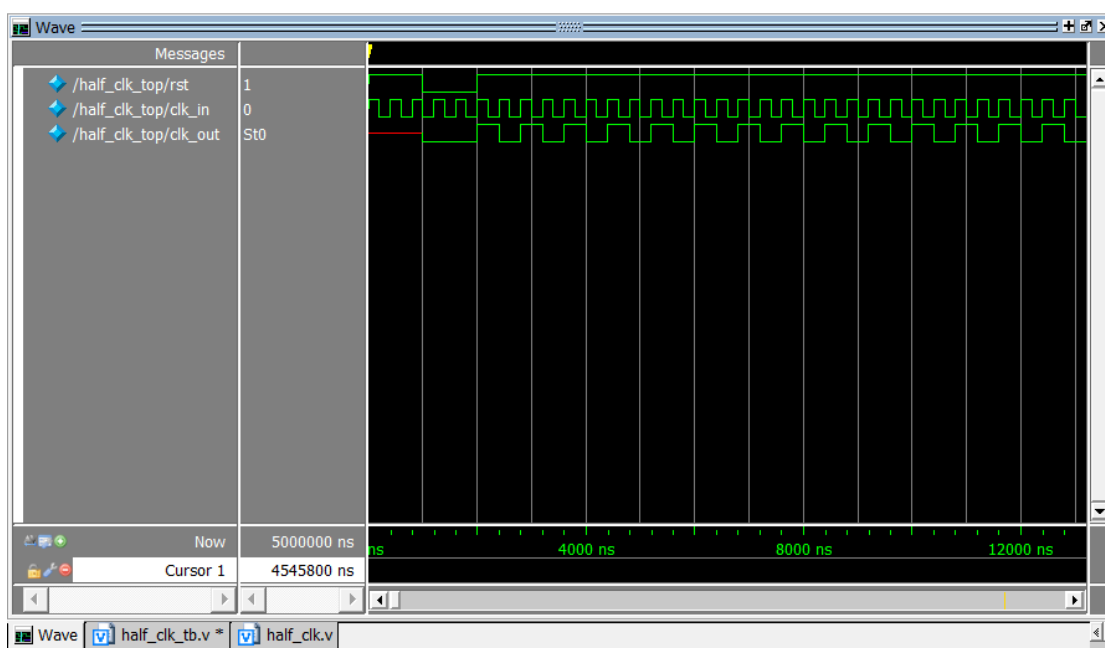


这样，看波形的窗口就会出现，将红色圈圈中的仿真时间步改成 5ms,然后按旁边的运

行按键， 波形就出现了。按住 ctrl 滚动鼠标滑轮可以缩放波形。

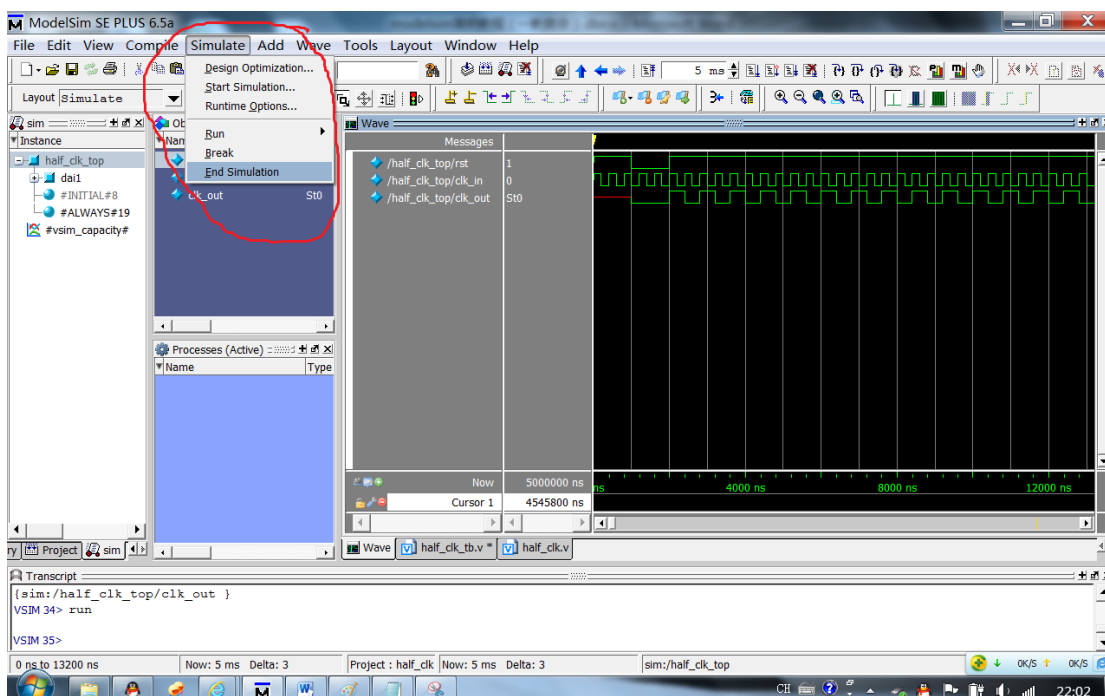


看下面的波形可观察，在 rst 复位并置高之后，clk_out 就出现了，并且周期是 clk_in 的两倍，也就是输出频率为输入频率的一半，达到了二分频的效果。



五、 停止仿真

当 modelsim 在仿真中的时候, 修改程序、编译等都是无效的, 也不能强行关闭软件, 这就需要手动停止仿真, 以便进行其他操作。选择菜单栏中的 simulate→End Simulation 即可。



总结:

建立工程→编写主程序和测试程序→编译→仿真→观察波形。

写了很多，但是应该讲得很详细了吧，希望对大家的学习有所帮助。