

# ModelSim 使用指南

Mentor 公司的 ModelSim 是业界最优秀的 HDL 语言仿真软件，它能提供友好的仿真环境，支持 Verilog HDL、VHDL 以及两者的混合仿真。可以将整个程序分步执行，在程序执行的任何步骤任何时刻都可以查看任意变量的当前值，并可以查看某一单元或模块的输入输出的连续变化等。ModelSim 分几种不同的版本：SE、PE、LE 和 OEM，其中 SE 是最高级的版本。而集成在 Actel、Atmel、Altera、Xilinx 以及 Lattice 等 FPGA 厂商设计工具中的均是其 OEM 版本。本节针对初学者，以分频器仿真为例介绍 ModelSim SE 10.1a 的基本使用步骤。

第 1 步：打开 ModelSim 软件，启动画面如图 B-1 所示，进入 ModelSim 主窗口如图 B-2 所

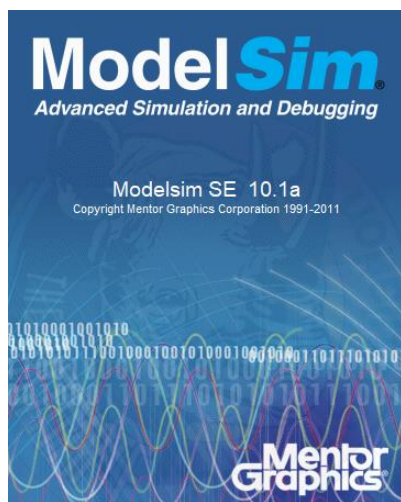


图 B-1 软件的启动画面

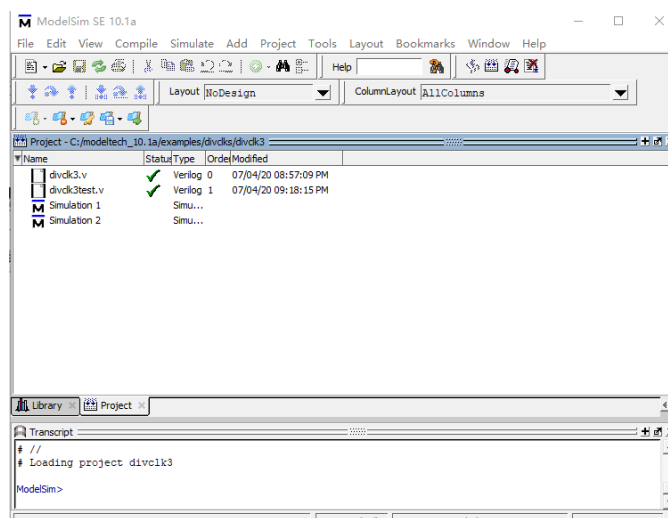


图 B-2 ModelSim 主窗口

示。

第 2 步：工作目录的转换。选择菜单 File→Change Directory，在弹出的 Choose directory 对话框中转换工作目录路径，本例设为 C:/Verilog/counter，单击确定按钮完成工作目录的转换。

第 3 步：新建仿真工程项目，增加仿真文件。选择 File 菜单下的 "New→Project"，新建一个工程，在弹出的对话框中，给该工程命名并指定一个存放的路径，在这里，工程名和顶层文件名保持一致是推荐的做法。默认的库名就是 "work"，这个无需更改，如图 B-3 所示，单击 "OK" 按钮完成新工程项目的创建。此时会弹出图 B-4 所示的对话框，提示添加文件到当前项目，如果仿真文件未准备，则选择 Create New File 选项，新建仿真文件。这里新建两个仿真文件，一个是 counter.v，一个是 counter\_test.v，前者为原始的设计文件(源代码)，后者是其相应的仿真测试文件（测试代码）。首先建立 counter.v，如图 B-5 所示，在对话框中填写

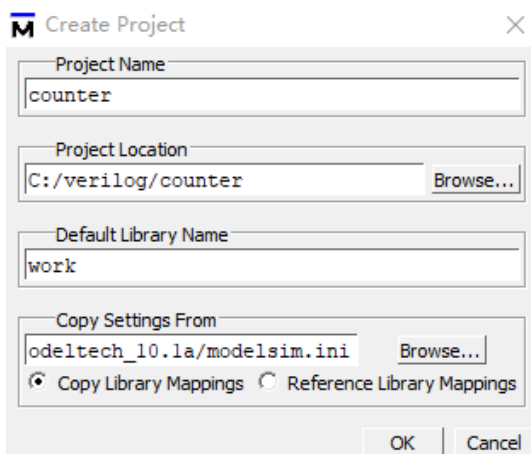


图 B-3 新建工程项目

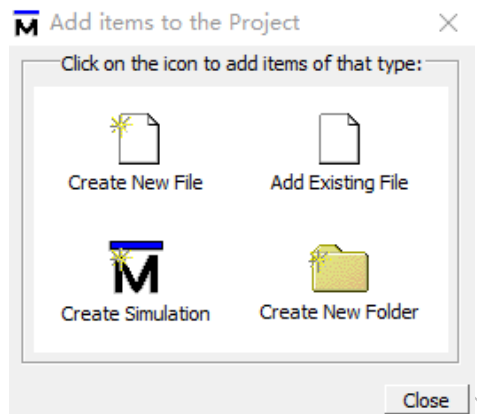


图 B-4 新建仿真文件

文件名为 counter，选择文件类型为 Verilog，单击 OK 按钮，此时，Project 页面中会出现 counter.v 的图标，双击图标，在出现的程序编辑区或者记事本中编写代码。类似步骤创建 counter\_test.v 文件，这里省略具体操作。注意图 B-4 所示的 Add items to the Project 窗口中的 Create New File 等功能均可在 project 选项卡中单击鼠标右键，通过弹出的菜单实现。当图 B-4 所示的窗口不再需要时，可以手动关闭它，点击 close。

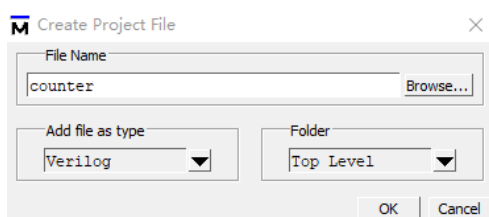


图 B-5 新建项目文件

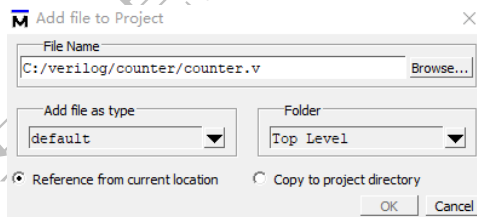


图 B-6 加入已存在的仿真文件到工程中

以下给出两个仿真文件的代码：

源代码 counter.v:

```
`timescale 1ns/1ps //时间单位/时间精度
module counter(
input      clk,
input      rst,
output reg[3:0] cnt, //时间计数器
output     div_2,    //2 分频
output     div_4,    //4 分频
output     div_8     //8 分频
);
always @ (posedge clk or posedge rst)
begin
if(rst) begin
cnt<=4'h0;
end
else begin
cnt<=cnt+1'b1;
end
end
```

---

```

        assign div_2=cnt[0];
        assign div_4=cnt[1];
        assign div_8=cnt[2];
    endmodule
测试代码 counter_test.v:
`timescale 1ns/1ps //仿真时间单位/时间精度
module counter_test();
    reg      clk;
    reg      rst;
    wire[3:0] cnt;
    wire      div_2;
    wire      div_4;
    wire      div_8;

    parameter clk_cycle=10; //20M 时钟
    parameter clk_hcycle=5;
    counter dut( //实例化待测试模块
        .clk(clk),
        .rst(rst),
        .cnt(cnt),
        .div_2(div_2),
        .div_4(div_4),
        .div_8(div_8)
    );
    initial begin
        clk=1'b1;
        end
    always #clk_hcycle clk=~clk; //产生时钟信号
    initial begin //产生复位信号
        rst=1'b1;
        #10 //延时 10ns 即 10 个时间单位后,rst 从 1 变为 0
        rst=1'b0;
        end
    initial begin
        $monitor($time,,clk,,rst,,cnt,,div_2,,div_4,,div_8);
        #10000 $stop;
        end
    endmodule

```

当然新建这两个文件的步骤可以放在全部工作开始之前进行，无需等到第 3 步开始的时候再进行。这时选择 Add Existing File 选项，将已存在的仿真文件加入当前工程，如图 B-6 所示。通过新建文件或者添加已有的文件两种方式，最终都是在该工程路径下建立好了两个文件，如图 B-7 所示。因为还没有编译文件，所以 Status 一栏显示的是两个问号。

Project - C:/verilog/counter/counter					
Name	Status	Type	Order	Modified	
counter_test.v	?	Verilog	1	07/05/20 10:53:08 AM	
counter.v	?	Verilog	0	07/05/20 10:40:43 AM	

图 B-7 Project 区域状态

第 4 步：编译仿真文件到 work 工作库。ModelSim SE 是编译型仿真器，所以在仿真前必须对仿真文件进行编译，并加载到 work 工作库。

在 Project 区域单击鼠标右键，选择"Compile→Compile All"，ModelSim 会对 counter.v 和 counter\_test.v 两个文件进行编译，如果编译通过，则会在文件旁边的 Status 一栏中显示两个绿色的√，否则显示×，并在命令中出现错误信息提示，双击错误信息可自动定位到代码中的错误出处，修改后，重新编译，直到通过为止。编译完成后，选择 Library 标签页，会发现在 work 工作库中出现了 counter 和 counter\_test 图标，这是刚才编译的结果，如图 B-8 所示。

Library			
Name	Type	Path	
work	Library	C:/verilog/counter/work	
counter	Module	C:/verilog/counter/counter.v	
counter_test	Module	C:/verilog/counter/counter_test.v	
floatfixlib	Library	\$MODEL_TECH/./floatfixlib	
mc2_lib (empty)	Library	\$MODEL_TECH/./mc2_lib	
mtiAvm	Library	\$MODEL_TECH/./avm	
mtiOvm	Library	\$MODEL_TECH/./ovm-2.1.2	
mtiPA	Library	\$MODEL_TECH/./pa_lib	
mtiUPF	Library	\$MODEL_TECH/./upf_lib	
mtiUvm	Library	\$MODEL_TECH/./uvm-1.1a	
sv_std	Library	\$MODEL_TECH/./sv_std	
vital2000	Library	\$MODEL_TECH/./vital2000	
ieee	Library	\$MODEL_TECH/./ieee	
modelsim_lib	Library	\$MODEL_TECH/./modelsim_lib	
std	Library	\$MODEL_TECH/./std	
std_developerskit	Library	\$MODEL_TECH/./std_developerskit	
synopsys	Library	\$MODEL_TECH/./synopsys	
verilog	Library	\$MODEL_TECH/./verilog	

图 B-8 编译仿真文件到 work 工作库

第 5 步：仿真配置。

编译通过之后，在 Project 区域鼠标右键点击"Add to Project → Simulation Configuration"，如图 B-9 所示：

Project - C:/verilog/counter/counter					
Name	Status	Type	Order	Modified	
counter_test.v	✓	Verilog	1	07/05/20 02:21:25 PM	
counter.v	✓	Verilog	0	07/05/20 02:21:55 PM	

Edit	
Execute	
Compile	
Add to Project	New File...
Remove from Project	Existing File...
Close Project	Optimization Configuration...
Update	Simulation Configuration...
Properties...	Folder...
Project Settings...	

图 B-9 仿真配置

在出现的 Add Simulation Configuration 对话框如图 B-10 所示，单击右下角 Optimization Options，打开后切换到 Options 选项卡页面，在 Optimization Level 中选择 Disable Optimizations，如图 B-11 所示：

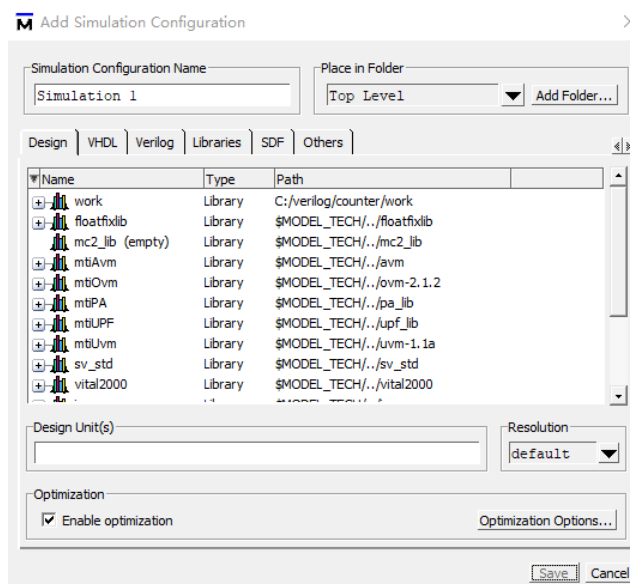


图 B-10 添加仿真配置对话框

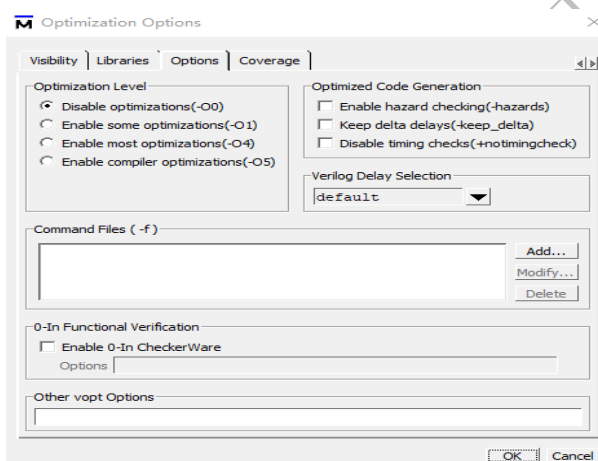


图 B-11 关闭优化选项 1

单击 OK 返回 Add Simulation Configuration 对话框，在 Optimization 栏中关闭 Enable Optimization，再展开 work 目录，选中 Test Bench 文件 test\_counter8，之后 save 保存。如图 B-12 所示：

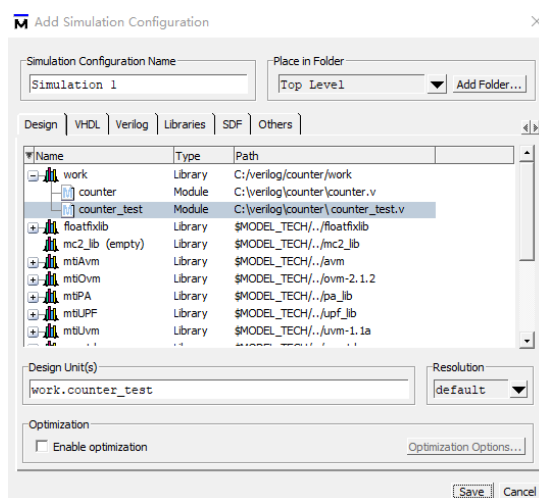
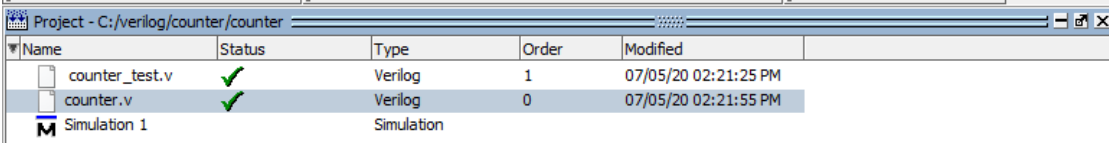


图 B-12 关闭优化选项 2

此时会在 Project 区域出现一个仿真配置文件：Simulation 1，如图 B-13 所示，双击它就能进入仿真了，在重启 ModelSim 之后，还可以双击它进入仿真，比较方便。



Name	Status	Type	Order	Modified
counter_test.v	✓	Verilog	1	07/05/20 02:21:25 PM
counter.v	✓	Verilog	0	07/05/20 02:21:55 PM
Simulation 1		Simulation		

图 B-13 生成仿真配置文件 Simulation 1

注意：如果不关闭优化选项的话，有时候 ModelSim 软件会报错导致不能正常进行仿真。

第 6 步：加载设计。

双击"Simulation 1"后进入仿真波形界面。在工作区中出现 Sim 页面时，如图 B-14 所示圆圈处，说明装载成功。

第 7 步：加载信号到 Wave 窗口中。

在 Object 区域鼠标右键选择"Add → To Wave → Signals in Region"，把待仿真的信号添加入 Wave 窗口，如图 B-14 所示。也可以将 Objects 窗口中出现的信号用鼠标左键拖到 Wave 窗口中（不想观察的信号则不需要拖）。

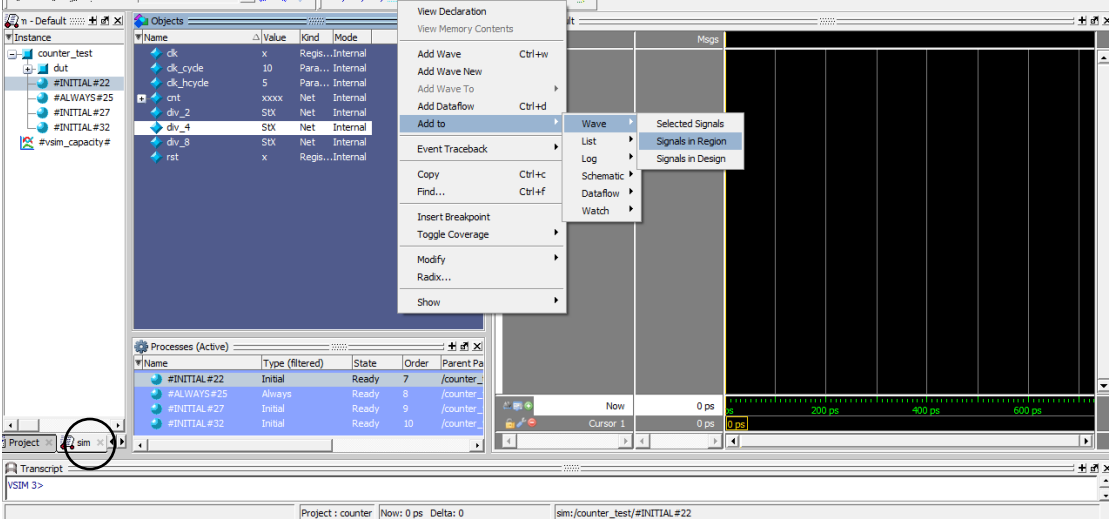


图 B-14 待仿真的信号添加入 Wave 窗口

如何修改信号值显示类型，如把 wave 窗口中的 div\_4 改成无符号数显示，在 div\_4 上单击鼠标右键，按照图 B-15 的方法修改即可。

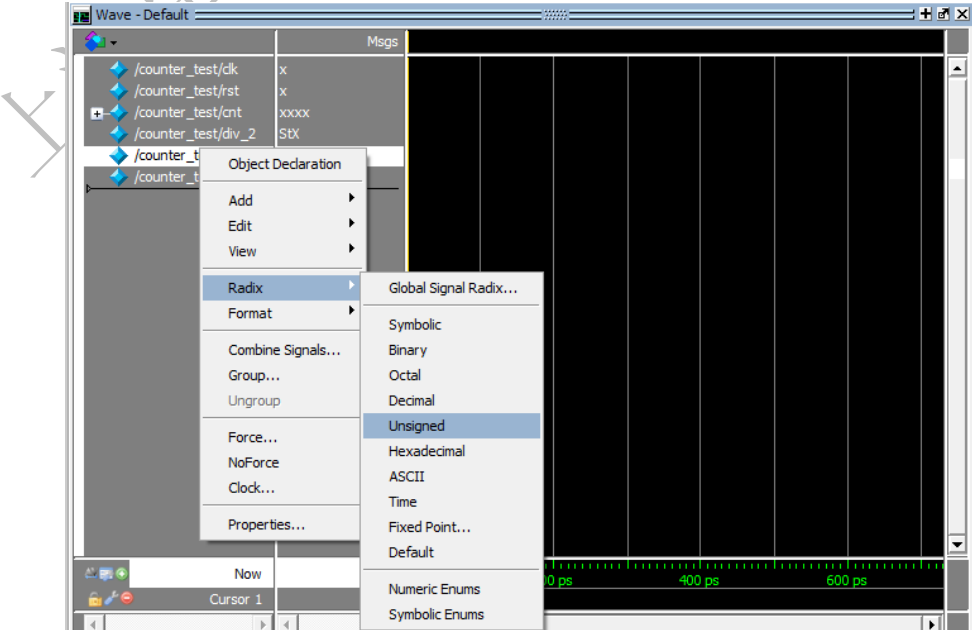


图 B-15 无符号数设置示例

第 8 步：仿真设计文件

设置仿真运行时间，假设置为 10ms。可以通过主窗口菜单 Simulate->Runtime Options 设置运行时间等参数，或者在空白仿真波形窗口中调整工具栏中的运行时间长度（Run

Length）。10ms 为设定每次运行的时间长度 Run Length；表示运行 Run；表示

继续运行（continueRun）；Run -All 表示一直运行，直到单击 Break 中断。在最下面的命令行窗口输入“run”，或者选中 Simulate->Run-All，如图 B-16 所示，得到功能仿真结果如图 B-17 所示。

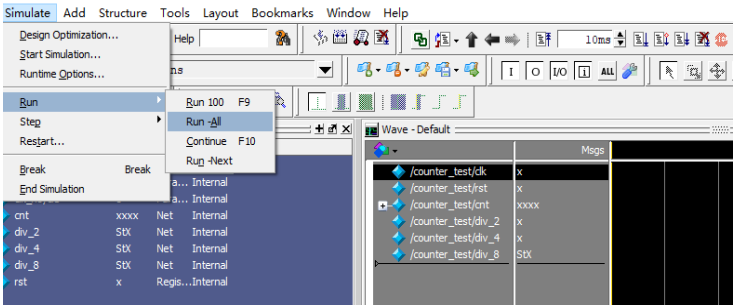


图 B-16 仿真运行命令

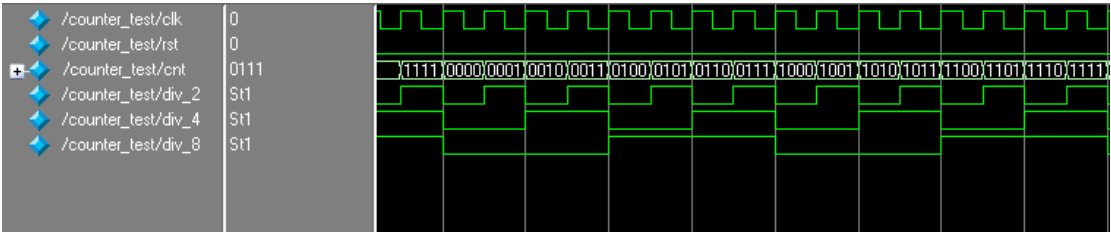


图 B-17 功能仿真波形