# 5.1 ModelSim 软件的主要结构

首次启动 ModelSim 6.4a 软件,可以看到 ModelSim 的主窗口,包 括菜单栏、工具栏、工作区和命令行操作区,如图 5.1 所示。



图 5.1 ModelSim 主窗口

在工作区可以根据操作显示 Project 标签、Library 标签、Sim 标签 (显示 Load Design、Hierarchical Structure)以及 Files 标签;在命令行 操作区,可以用命令提示符的方式进行编译、仿真设计,同时打开其他 窗口。

## 5.2 ModelSim 的简要使用方法

#### 1. 建立工程

使用 ModelSim 建立工程的操作步骤如下: (1) 第一次打开 ModelSim 会出现 Welcome to ModelSim 对话

第 5 章 Quartus II软件第三方工 具

框,选择 Create a Project,或者启动 ModelSim 后选择 File→New→Project...命令,都会 打开 Create Project 对话框,如图 5.2 所示。

Project Name	
test	
Project Locatio	n
C:/altera/90/	modelsim_ae/examples Browse.
work	
	From
Copy Settings F	
Copy Settings H 90/modelsim_a	e/modelsim.ini Browse

图 5.2 Create Project 对话框

(2) 在 Create Project 对话框中,填写 test 作为 Project Name; 在 Project Location 栏中选择 Project 文件的存储目录; 保留 Default Library Name 的设置为 work。

(3) 单击 OK 按钮确认,在 ModelSim 软件主窗口的工作区中即增加了一个空的 Project 标签,同时弹出一个 Add items to the Project 对话框,如图 5.3 所示。

ModelSim ALTERA 6.4a - Custom Altera Version File Edit View Compile Simulate Add Project Tools Layout Window Help	
[□-☞■●● ※唱電⊇⊇ #點點點兩 ]Heb [ 點   多幽翼壓   Contains	3
X X 🖸 🗄 🀐 Layout NoDesign 💌	
Workspace       Image: Control of the intervence of the interv	
Transcript	± =========
# Loading project test	1
ModelSim> A Transcript	

图 5.3 Add items to the Project 对话框

(4) 添加包含设计单元的文件。直接单击 Add items to the Project 对话框中的 Add Existing File 或 Create New File 选项可以在工程中加入已经存在的文件或建立新文件。本节我们选择 Add Existing File。

也可以单击 Close 按钮关闭 Add items to the Project 对话框以后,在 ModelSim 软件中选择 File→Add to Project→Existing Files 命令,弹出 Add file to Project 对话框,如

-- EDA技术与SoC设计应用

图 5.4 所示。

File Name	
F:/PWM/pwm.v	Browse
Add file as type	Folder
default 🗾	Top Level

图 5.4 Add file to Project 对话框

单击 Add file to Project 对话框中的 Browse 按钮,打开 ModelSim 安装路径中的 examples 目录,选取 counter. v 和 tcounter. v 文件(注意,同时选取两个文件时,文件名 直接用空格隔开),再打开对话框下面的 Reference from current location 选项,然后单击 OK 按钮。

(5) 在工作区的 Project 选项卡中可以看到新加入的文件,单击右键,选取 Compile → Compile All 命令对加入的文件进行编译,如图 5.5 所示。



图 5.5 编译软件

(6)两个文件编译完成后,用鼠标单击 Library 标签栏。在 Library 选项卡中,用鼠标单击 work 库前面的加号"+",展开 work 库,将会看到两个编译了的设计单元(如果看不到,需要把 Library 的工作域设为 work),如图 5.6 所示。

(7) 导入一个设计单元。双击 Library 选项卡中的 pwm,在工作区中将会出现 sim 标签,其中显示了 pwm 设计单元的结构,如图 5.7 所示。

到这一步通常就开始运行仿真、分析以及调试设计了,不过这些工作将在后面的讲述中完成。现在,结束仿真并关闭工程。选择 Simulate→End Simulation 命令,当提示是

Ten Ten Te	ompile Sin	nulate A <u>d</u> d Libr	rary Tools Layout Window Help
D-68858		122   <b>M</b> E	: 🖥 🖬 📙 Help 🛛 🖓 🖓 🖓 🕅 🖧 💁 Contains 🔍 🧷
X4 X 🗅 🖻 🐐	Layout No	Design 💌	
orkspace		+ # ×	Transcript # d
Name	Type	Path _	# Reading C:/altera/90/modelsim_ae/td/vsim/pref.td
work	Library	work	# Loading project test
-M pwm	Module	F:/PWM/pwm.	ModelSim>
220model	Library	\$MODEL_TECH	
220model_ver	Library	\$MODEL_TECH	
alt_ver	Library	\$MODEL_TECH	
alt_vtl	Library	\$MODEL_TECH	
- altera	Library	\$MODEL_TECH	
⊕– <b>∭</b> altera_mf	Library	\$MODEL_TECH	
] altera_mf_ver	Library	\$MODEL_TECH	
- altera_ver	Library	\$MODEL_TECH	Let Transcript 31
- altgxb	Library	\$MODEL_TECH	
- altgxb_lib	Library	\$MODEL_TECH	
altgxb_ver	Library	\$MODEL_TECH	
apex20k	Library	\$MODEL_TECH	
apex20k_ver	Library	\$MODEL_TECH	Model Sim.
)- 🏨 apex20ke	Library	\$MODEL_TECH	
		····· ····	

图 5.6 编译后的设计单元

NA ModelSim ALTERA 6.4a	
File Edit View Compile Simulate Add	Transcript Tools Layout Window Help
The Edit View Comple Simulate Aga	
▋▋・ਫ਼₽ਫ਼₽₽₽₽₽₽	Al 5 15 M    Help   Mai   50 🖽 🗱
	🕅 🔁 🗗 👔 🏫 🖉 Contains 🔍 📿 🔀 🕅 🕅 🦓
Layout Simulate	
Workspace H d X	Transcript 🔤 👬 🖬 🗗
▼ Instance Design unit Design	ModelSim> vsim work.pwm
pwm pwm Module     #ALWAYS#29 pwm Proces     #ALWAYS#71 pwm Proces     #ASSIGN#93 pwm Proces	# visim work.pwm # // ModelSim ALTERA 6.4a Oct 22 2008 # // Copyright 1991-2008 Mentor Graphics Corporation # // Copyright 1991-2008 Mentor Graphics Corporation # // All Rights Reserved.
	# // THIS WORK CONTAINS TRADE SECRET AND # // PROPRIETARY INFORMATION WHICH IS THE PROPERTY # // OP MENTOR GRAPHICS CORPORATION OR ITS LICENSORS # // AND IS SUBJECT TO LICENSE TERMS, # // # Loading work.pwm
	VSIM 2>
-	ि Transcript
	Model Sim .
Project : test Now: 0 ns Delta: 0	sim:/pwm

图 5.7 导入设计单元

否退出仿真时选择"是",然后选择 File→Close→Project 命令,并确定关闭当前工程。 前面的步骤在工作目录下建立了一个名为 tset.mpf 的工程文件,该文件中包含了建 立工程过程中的所有信息。在任何时间,可以选择 File→Open→Project 命令打开该工

215

第5章 Quartus II 软件第三方工具

程文件。

2. 基本 VHDL 仿真

1) 准备仿真

在进行仿真之前还应进行以下准备工作:

(1) 为本次练习新建一个目录,然后复制< ModelSim 安装目录>\Modeltech\_6.4a\ examples 目录中所有的 vhd 文件到该目录下。

(2) 启动 ModelSim 软件,选择 File→Change Directory 命令,在弹出的 Choose folder 对话框中设置该目录为当前工作目录。

(3) 在编译任何 HDL 代码前,要建立一个设计库 来存放编译结果。在 ModelSim 主窗口中,选择 File→ New→Library 命令建立一个新的设计库,如图 5.8 所示。

在 Create a New Library 对话框中,选择 Create 栏中的 a new library and a logical mapping to it 选项, 并在 Library Name 栏中输入新的设计库名称,如 work,单击 OK 按钮确定。

Create	20.000.0000
a new library a	and a logical mapping to it
C a map to an ex	xisting library
Library Name:	
NOT N	
Library Physical Na	me:
ork	
	OK Can

图 5.8 建立新的设计库

这就在当前目录中建立了一个子目录,即用户的设计库。ModelSim 在这个目录中 保存了名为\_info 的特殊文件。

也可以直接在 ModelSim 的命令行操作区或 DOS/UNIX 的命令行中输入下面的命令完成设计库的建立和逻辑映射:

vlib work(回车) vmap work work (回车)

注意:不要直接在 Windows 或 UNIX 的文件夹管理器中建立设计库目录,因为这样 建立的目录中无法得到 ModelSim 需要的特殊文件\_info。

(4) 选择 Compile→Compile 命令,将弹出 Compile Source Files 对话框,如图 5.9 所示。

M Compile Source Files	×
Library: work	
查找范围(I): 🕼 work	·●∎™≣▼
名称	修改日期
will counter.vhd	2008/7/2 9:03
۰ <u>ا</u>	,
文件名(N): counter.vhd	Compile
文件类型(T): HDL Files (*.v;*.vl;*.vhd;*.vh	dl;*.v 🔻 Done
Compile selected files together Default Opti	ons Edit Source

图 5.9 Compile Source Files 对话框

从文件列表中选取要编译的 VHDL 文件,如 counter. vhd,并单击图 5.9 中的 Compile 按钮,编译完成后单击 Done 按钮。也可以从文件列表中同时选择多个文件进 行编译,按照设计的需要依次选取并进行编译。

还可以直接在 ModelSim 的命令行操作区或 DOS/UNIX 的命令行中输入下面的命 令完成设计文件的编译:

vcom couter.vhd(回车)

(5) 在 ModelSim 主窗口工作区的 Library 选项卡中单击 work 库前面的加号"+"展 开 work 库,从中选择 counter 设计单元并双击鼠标左键,如图 5.10 所示。

M ModelSim ALTERA 6.4a	Tarle Levent Window Hale
D. Compile Simulate Age Struct	uure igois tayout window nep Br M Hep Br M Ø ≝ Ø B
	0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Layout Simulate	
	Copying C:\altera\90\modelsim_ae\vin32aloem//modelsim.ini to modelsim.ini     # Modfrying modelsim.ini     # Warning: Copied C:\altera\90\modelsim_ae\vin32aloem//modelsim.ini to modelsim.ini.     # Updated modelsim.ni.     vcom reportprogress 300 -work work C:/altera\90\modelsim_ae/examples/work/counter.vhd     # Model Technology Modelsim ALTERA vcom 6.4a Compler 2008.08 Oct 22 2008     # - Compling architecture only of counter     # - Compling architecture only of counter     # vaim work.counter     # vaim work.counter     # uoading std.standard     # Loading work.counter     # Loading work.counter     # Loading work.counter     # Some modelsim and the counter of the complex of the complex of the counter of the counter of the complex of the counter     # Some work.counter     # Loading work.counter     # Some work.counter     # Deading work.counter     # Some work.counter     # Loading work.counter     # Loading work.counter     # Loading work.counter     # Loading work.counter     # Deading work.counter     # Deading work.counter     # Loading work.counter     # Deading work.counter     # Deading work.counter     # Deading work.counter     # Loading work.counter     # Loading work.counter     # Loading work.counter     # Deading wor
Now: 0 ps Delta: 0 sim:/counter	1.

图 5.10 导入设计单元仿真

也可以直接在 ModelSim 的命令行操作区或 DOS/UNIX 的命令行中输入下面的命令行:

```
vsim counter(回车)
```

(6) 在 ModelSim 主窗口中选择 View 打开所有 ModelSim 窗口,包括 dataflow 窗口、list 窗口、local 窗口、message viewer 窗口、objects 窗口、process 窗口、call stack 窗口、workspace 窗口以及 wave 窗口。

也可以直接在 ModelSim 的命令行操作区输入下面的命令打开所有窗口:

```
view * (回车)
```

(7) 从 Objects 窗口中选择 Add→To Wave→All items in region 命令,如图 5.11 所示,该命令将设计中的顶层(top-level)信号加入 wave 窗口中,如图 5.12 所示。

也可以直接在 ModelSim 的命令行操作区输入下面的命令将顶层信号加入 wave 窗

#### -----EDA技术与SoC设计应用

:	on	To <u>W</u> ave	Objects
	on		
		To <u>L</u> ist	Contains 🔎
Mode	gn 👘	To Log	Name
Out		To Dataflow	
In		0	dk ∕
In			🔩 reset
		0	🔩 dk 🔩 reset

图 5.11 signals 窗口

wave - default File Edit View Add Fo	ormat Tools V	Vindow	×
0.000	₩ <b>@</b> 221	▲ ま ┺ │ ◈ 緇 綛 函	ď
100 p		14155555555555555555555555555555555555	B. B. B. B. B.
N G 34   BX 3+   4	∰   <b>Q, Q, Q</b> ,		· · · · · · · · · · · · · · · · · · ·
Messages			
C /counter/count /counter/clk /counter/reset	0000000 0 0		
E ■ Now	0 ps	os 500 ps 1000 ps 1500 ps 2000 p	s 2500 ps
É Cursor 1	0 ps	0 ps	
K D	4	•	Þ
0 ps to 2667 ps	Now: 0 ps	Delta: 0	1.

图 5.12 wave 窗口

口中:

add wave /counter/\*(回车)

(8) 在 Objects 窗口中选择 Add→To List→All items in region 命令,该命令在 list 窗口中加入设计中所有顶层信号,如图 5.13 所示。

💮 list		- • ×
<u>File Edit View</u>	<u>Add T</u> ools <u>W</u> indow	
0-6883	6   % BB 22	M = ~ ~ K
ps-↓ delta-↓	/counter/count- /counter/clk- /counter/reset-	<u> </u>
0 +0	00000000 0 0	
1 line	1	× }

图 5.13 list 窗口

也可以直接在 ModelSim 的命令行操作区输入下面的命令将顶层信号加入 list 窗口中:

add list /counter/\*(回车)

2) 运行仿真

通过输入连续的时钟信号开始仿真。

(1) 在 ModelSim 主窗口的命令行操作区中,在 VSIM 提示符下输入下面的命令:

force clk 1 50,0 100 - repeat 100(回车)

ModelSim 解释 force 命令如下:

从当前时间开始,在其后 50ns 时置 clk 为 1;

当前时间后 100ns 时置 clk 为 0;

每隔 100ns 重复该 clk 周期。

也可以通过菜单操作定义时钟输入信号:在 signals窗口中选中 clk 信号,然后选择 Objects 窗口的 Edit→Clock 命令,弹出如图 5.14 所示的时钟定义 (Define Clock)对话框,在此对话框中进行相应的设置。

(2)执行主窗口或 wave 窗口工具条按钮的两个不同的 Run 功能(Run 功能只在主窗口和波形(wave)窗口中定义,即这两个窗口中有 Run 功能)。首先选取 Run 按钮,运行完成之后选取 Run-All 按钮。



图 5.14 仿真时钟定义对话框

Run:运行仿真,在100ns后停止。也可以在 ModelSim 的命令行操作区输入 run 100 后按回车键,或在主窗口中选择 Simulate→Run→Run 100ns 命令。

Run-All: 一直运行仿真,直到单击 Break 按钮为止。也可以在 ModelSim 的命令行 操作区输入 run-all 后按回车键,或在主窗口中选择 Simulate→Run→Run-All。

(3) 选取主窗口或波形窗口的 Break 按钮来中断仿真,一旦仿真到达一个可接受的 停止点,它就停止运行。

在源文件窗口中,箭头指向下一条将被执行的 HDL 语句。如果暂停发生时仿真器 不是在评测一个过程,则没有箭头显示在源文件窗口上。

(4) 在函数内部设置一个断点,本节在 counter. vhd 文件的第 18 行函数内部设置一 个断点:移动鼠标到源文件窗口,在 21 行上单击设置断点,可以看到紧挨着行号有一个 红点,可以用鼠标单击切换断点的使能功能,断点禁止后看到的是一个小的红色圆环。 可以在断点上单击鼠标右键,选取 Remove BreakPoint 21 来取消断点,如图 5.15 所示。

也可以在 ModelSim 的命令行操作区输入下面的命令设置断点:

bp counter. vhd 21(回车)

注意:断点只能设置在蓝色标号的文本行上。

(5)选取连续运行快捷按钮(Continue Run)恢复中断了的运行,ModelSim 会碰上断点,通过源文件中的一个箭头或是在主窗口中的一条中断信息显示出来。

该操作也可以通过在 ModelSim 的命令行操作区输入 run-continue 后按回车键来完

#### EDA技术与SoC设计应用

H C:/alt	tera/90/modelsim_ae/examples/work/counter.vhd	×
<u>File</u> <u>E</u> c	dit <u>V</u> iew <u>T</u> ools <u>W</u> indow	
0-0	ਡ 🖬 🌣 🖨   🌡 🖻 🏥 Ω Ը   🐴 計 🐘 🛛 🕹 🏛 👰 🍱 🗍 💥 🗷 🖻 🕉 👘	ď
Ln#		-
14	architecture only of counter is	
15	constant tpd reset to count : time := 10 ns;	
16	constant tpd_clk_to_count : time := 5 ns;	
17		
18	function increment(val : bit_vector) return bit_vector	
19	is	
20	normalize the indexing	
210	alias input : bit_vector(val'length downto 1) is val;	
22	<pre>variable result : bit_vector(input'range) := input;</pre>	
23	variable carry : bit := '1';	
24	begin	0.0
25	for i in input'low to input'high loop	
26	result(i) := input(i) xor carry;	
27	carry := input(i) and carry;	
28	exit when carry = '0';	
29	end loop;	
30	return result;	-1
	Ln: 18 Col: 55 Ri	EAD //

图 5.15 设置或取消断点

成,或在 ModelSim 主窗口中选择 Simulate→Run→Continue 命令。

(6) 单击 Step 按钮可以单步执行仿真,注意 Locals 窗口中值的变化,如图 5.16 所示。可以持续单击 Step 按钮执行单步仿真。

M Locals		
<u>File Edit View Add</u>	Window	
Locals		: 1 M X
Da # 2 - Ca		
▼ Name	Value	
only		
tpd_reset_to_count		
tpd_clk_to_count	5 ns	
ctr		

图 5.16 Locals 窗口

该操作也可以通过在 ModelSim 的命令行操作区输入 run-step 后按回车键来完成, 或在 ModelSim 主窗口中选择 Simulate→Run→Step 命令。

(7) 输入下面的命令结束仿真:

```
quit-force(回车)
```

该命令不需要确认就直接结束仿真并退出 ModelSim。

3. 基本 Verilog 仿真

1) 准备仿真

在进行基本 Verilog 仿真之前要进行以下准备工作:

(1) 为本练习新建一个目录,复制< ModelSim 安装目录>\ Modeltech\_6.4a \

examples 目录中所有的 Verilog(.v)文件到该目录下。

(2)设置该目录为当前工作目录,通过从该目录直接调用 ModelSim 或启动 ModelSim 软件后选择 File→Change Directory 命令来完成。

(3) 在编译 Verilog 文件前,需要在新目录下生成一个设计库来保留编译结果。如果读者仅仅熟悉解释性 Verilog 仿真器,诸如 Cadence Verilog-XL,那么对于用户来说这是一个新的方法。因为 ModelSim 是一个编译性 Verilog 仿真器,对于编译它需要一个目标设计库。如果需要,ModelSim 能够编译 VHDL 和 Verilog 代码到同一个库中。

(4) 在 ModelSim 主窗口中,选择 File→New→Library 命令建立一个新的设计库,如图 5.17 所示。

Compile Source Files	
Library: work	
宣抚氾围(L):  Jaltera	
名称	修改日期
퉬 work	2008/7/2 11:12
Counter.v	2008/7/2 11:14
tcounter.v	2008/7/2 11:16
7件名(N): 「counter y」「tcounter y」	Comila
文件类型(T): HDL Files (*. v;*. vl;*. vhd;*. vh	hdl;*.v V Done
Compile selected files together Default Opt	tions Edit Source

图 5.17 Compile Source Files 对话框

在 Create a New Library 对话框中,选择 Create 栏中的 a new library and a logical mapping to it 选项,并在 Library Name 栏中输入新的设计库名称,如 work,单击 OK 按钮确定。

这就在当前目录中建立了一个子目录,即用户的设计库。ModelSim 在这个目录中 保存了名为\_info 的特殊文件。

也可以直接在 ModelSim 的命令行操作区或 DOS/UNIX 的命令行中输入下面的命令完成设计库的建立和逻辑映射:

```
vlib work(回车)
vmap work work (回车)
```

注意:不要直接在 Windows 或 UNIX 的文件夹管理器中建立设计库目录,因为这样 建立的目录中无法得到 ModelSim 需要的特殊文件\_info。

(5)编译 Verilog 设计。这个设计例子由两个 Verilog 源文件组成,每一个都包含一个唯一的模块。文件 counter.v包含一个名为 counter 的模块,它执行一个简单的八位二进制加法计数器。另一个文件 tcounter.v是一个测试台模块(test\_counter),通常用来检验 counter。在仿真中,可以看到这两个文件被层次化配置,它们使用了被测试台实例化的 counter 模块的一个简单实例(实例名为 dut)。后面将有机会看到这个代码的结构,现在,需要编译两个文件到 work 设计库中。

(6) 选择 Compile → Compile 命令编译 counter. v 和 tcounter. v 两个文件。在如图 5.17 所示的 Compile Source Files 对话框中,使用 Ctrl 键+鼠标左键操作同时选择工作目录中的 counter. v 和 tcounter. v 文件,然后单击 Compile 按钮,完成后单击 Done 按钮。

也可以直接在 ModelSim 的命令行操作区输入下面的命令完成编译:

vlog counter.v tcounter.v(回车)

注意: ModelSim 只有在载入设计以后才检查 Verilog 模块的顺序,因此,在这里不用考虑编译 counter. v 和 tcounter. v 文件的先后顺序。

(7) 在主窗口工作区的 Library 选项卡中,单击 work 库前面的加号"+"展开该库,可以看到 counter 和 test\_counter 两个设计单元,如图 5.18 所示。

在图 5.18 中,用鼠标左键双击 test\_counter 载入设计单元,同时在工作区中出现一个新的 sim 选项卡,如图 5.19 所示。



图 5.18 展开 work 库

也可以直接在 ModelSim 的命令行操作区输入下面的命令完成设计单元的载入:

vsim test\_counter(回车)

(8) 在主窗口命令行操作区的 VSIM 提示符下输入下面的命令,调出 signals、list 和 wave 窗口:

view signals list wave(回车)

也可以通过选择 View→<窗口名>打开需要的窗口。

(9)向 wave 窗口添加信号。在 signals 窗口中,选择 Edit→Select All 命令,选择所 有信号,然后拖动所有信号到 wave 窗口的路径名或数值窗格中,如图 5.20 所示。

HDL条目也能够从一个窗口复制到另一个窗口(或者是在 wave 和 list 窗口内部),

Image: Second secon	ModelSim ALTERA 6.4a File Edit View Compile Simulate A	dd Transcript Tools Lavout Window Help
↑       +       →       E	D-288081189822	♠ ∰ ∰ M    Help 🌇 🆓 🕮 🐼 🕅
Contains       Contains <td< th=""><th>▲</th><th>± 24 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18</th></td<>	▲	± 24 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18
Workspace       Image: State of the state o	Contains 🔍 📝 🕅	🔁 🕅 🀐 🛛 Layout Simulate 🔍
Instance Design unit Desi     Instance Design unit Desi     Instance Design unit Desi     Instance Design unit Desi     Instance Itest_counter Mod     Instance Mod     Instance Troc     Instance Troc	Workspace 🛛 👬 📶 📩	Transcript ::::::::::::::::::::::::::::::::::::
	test_counter test_counter Mod     function for the set_counter Mod     function for the set_counter Mod     function for the set_counter for test_counter	Loading work.counter     Loading work.counter     Loading work.counter     To the test counter (ISCALE] - Module 'counter' does not hav     e a 'timescale directive in effect, but previous modules do.     # Region: /test_counter/dut     VSIM 13>
	· ·	hC:/altera/tcounter.v + ₫ ≥ Ln#

图 5.19 载入设计单元



图 5.20 向 wave 窗口拖入信号

这可通过选择 Edit→Copy 和 Edit→Paste 命令完成。通过 Edit→Delete 命令也能删除 选取的条目。

(10) 导入设计的时候会在工作区打开一个新的 sim 标签。在 sim 选项卡中单击加 号"+"展开设计层次结构,可以看到本节实例的 test\_counter、dut(counter)和名为 increment 的函数(如果在 sim 选项卡中没有显示 test\_counter,则当前仿真的是 counter, 而不是 test\_counter),如图 5.21 所示。

(11) 单击其中的 increment 函数,可以注意到其他窗口自动更新。明确地说,source 窗口显示了在 sim 选项卡层次结构中所选层次的 Verilog 代码,signals 窗口显示了对应

第5章

Quartus II 软件第三方工具

ModelSim ALTERA 6.4a	• ×
<u>File Edit View Compile Simulate Add Structure Tools Layout Window Help</u>	
X → X D B K Layout Simulate ▼ 2 2 2 2 2 2 5 4 4 4 4 4 4 4 4 4 4 4 4 4	
Workspace Transcript	16 H =
VInstance       Design unit       Design unit       Design unit         Image: test_counter       test_counter       Module         Image: test_counter       Process         Image: test_counter </td <td>- </td>	- 
	+ 4 ×
Now: 0 ns Delta: 0 sim:/test_counter	//

图 5.21 层级结构显示

信号。在这种方式下使用层次结构类似于解释性 Verilog 仿真器中的 Scoping 命令。 现在,单击 sim 选项卡中的顶层行,保证 test\_counter 模块显示在 source 窗口中,如

图 5.22 所示。

h C:/a	ltera/tcounter.v	×
<u>File</u>	dit <u>V</u> iew <u>T</u> ools <u>W</u> indow	
0-0	≥ 🛛 ॐ ቆ   Հ № ቈ Չ Ը   А Է ╚     ♦ ఝ 🎗 🕅	¥
J X∢ ▶)	K 🗅 🖻 🍂	
Ln#		-
1	`timescale 1ns/1ns	
2	module test counter;	
3		
4	reg clk in;	
5	reg rst;	
6	initial	
7	begin	
8	clk in=1;	
9	rst=1;	
10	#20000 rst=0;	
11	#20000 rst=1;	-
	4	+
	"tcounter.v" was modified after it was Ln: 2 Col: 15	1.

图 5.22 source 窗口

2) 运行仿真

(1) 单击主窗口工具条上的 Run 按钮启动仿真,默认仿真长度为 100ns; 也可以在 ModelSim 的命令行操作区输入 run(回车),或在主窗口中选择 Simulate→Run→Run 100ns 命令。

(2) 设置运行长度(Run Length)为 500ns,然后单击 Run 按钮,如图 5.23 所示。

现在仿真运行了 600ns(默认的 100ns 加上设置的 500ns),在工作区底部状态栏可以 看到这些信息。

ModelSim ALTERA 6.4a	
Image: Second state       Image: Second state<	Indow (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
Image: Simple	odel <i>Sim</i> 。

图 5.23 设置运行长度

(3)上面的设置使仿真器前进了 500ns,也可以通过下面的命令设置仿真器推进的 时间:

run @ 3000(回车)

(4) 单击主窗口工具条上的 Run-All 快捷按钮,让仿真器连续运行,直到停止在 tcounter.v 模块中为止,如图 5.24 所示。



图 5.24 仿真停止在 tcounter. v 模块

也可以单击 Break 按钮中断运行,在 source 窗口中查看中断时执行的语句。

3) 调试仿真

(1) 在 list 窗口中选取/test\_counter/count,如图 5.25 所示。从 list 窗口菜单条中选择 View→Signal Properties,弹出 List Signal Properties 对话框,如图 5.26 所示。

第5章

Quartus II 软件第三方工具



图 5.25 在 list 窗口中选择/test\_counter/count



在 Radix 栏为信号 counter 选取十进制(Decimal),相应的 list 窗口的输出也发生改变,成为十进制数,而不是缺省的二进制了。

(2) 选取主窗口工作区中 sim 选项卡层次结构中的 dut: counter,然后在 source 窗 口打开的 counter.v 中的第 30 行(这里包含一个 Verilog 功能增量的调用)设置断点,如 图 5.27 所示。

(3) 单击 Restart 按钮,重载设计组件并重置仿真时间为零。也可以在 ModelSim 的 命令行操作区输入 restart(回车),或在主窗口中选择 Simulate→Run→Restart 命令。

(4) 确认 Restart 对话框中所有条目被选中,如图 5.28 所示,然后单击 Restart 按钮。



图 5.27 在 source 窗口中设置断点

图 5.28 Restart 对话框

(5) 在主窗口工具条中选择 Run-All 快捷按钮,恢复执行仿真。中断后观察 source 窗口。

(6)当中断到达后,可以观察一个或多个信号的值,通过以下几种方法可以检测这些 值查看显示在 signals 窗口中的值;在 source 窗口中,在变量上单击鼠标右键,从弹出的 右键菜单中选择 Examine 命令; 在命令行操作区输入 examine 命令,可以输出变量值, 如 examine count。

(7) 执行单步命令 Step, 遍历 Verilog 源函数。

(8) 结束仿真,其执行命令为 quit-force。

4. VHDL/Verilog 混合仿真

1) 准备仿真

(1) 建立一个新的工作目录,复制< ModelSim 安装目录>\Modeltech\_5.8d\ examples\mixedHDL目录中所有的VHDL(.vhd)文件和Verilog(.v)文件到该目录下。

(2) 启动 ModelSim 软件,选择 File→Change Directory 命令,将新建目录设置为当前工作目录。

(3) 在主窗口中选择 File→New→Library 命令,在新目录下建立一个设计库来保留 编译结果。

(4)编译文件。在主窗口中选择 Compile → Compile 命令,打开 Compile Source Files 对话框,逐个编译 Verilog 文件,如 cache.v、memory.v和 proc.v。也可以在主窗口 命令行操作区输入下面的命令行完成 Verilog 文件编译:

vlog cache.v memory.v proc.v(回车)

(5) 依赖于设计,VHDL的编译次序是特定的。在这个例子中,top.vhd 文件必须最后编译。按照下面的顺序编译文件:

util.vhd set.vhd top.vhd

或在主窗口命令行操作区输入下面的命令行完成 VHDL 文件编译:

vcom util.vhd set.vhd top.vhd(回车)

(6) 编译完成后,单击 Compile Source Files 对话框中的 Done 按钮。

2) 运行仿真

(1) 在主窗口工作区的 Library 选项卡中单击 work 库前面的加号"+"展开该库,可 以看到所有设计单元。

(2) 用鼠标左键双击 work 库中的 top 实体,在工作区中出现一个新的 sim 选项卡。 也可以直接在 ModelSim 的命令行操作区输入下面的命令完成 top 实体的载入:

vsim top(回车)

(3) 在 ModelSim 主窗口中选择 View→All Windows 命令,打开所有仿真窗口。也可以直接在命令行操作区输入下面的命令:

view \* (回车)

(4) 在 signals 窗口中,选择 Add→Wave→Signals in Region 命令,向 wave 窗口添加 信号;选择 Add→List→Signals in Region 命令,向 list 窗口添加信号。

也可以直接在 ModelSim 命令行操作区输入下面的命令实现向 wave 窗口和 list 窗

EDA技术与SoC设计应用

口添加信号:

```
add wave *(回车)
add list * (回车)
```

(5)观察 ModelSim 工作区 sim 选项卡中的层次结构。注意设计中两者的层次混合,VHDL 层级用一个蓝色方框前缀指示,Verilog 层级用一个天蓝色的圆形前缀指示。

(6) 在工作区 sim 选项卡中,单击层次结构中的 c: cache 模块,它的源代码出现在 source 窗口中。

(7) 在 source 窗口中,选择 Edit→Find 命令,用查找功能在 cache. v 文件中定位到 cache\_set 的声明位置。

(8) 找到 cache\_set 声明后可以发现, cache\_set 是 cache. v 文件内实例化的 VHDL 实体。

(9) 在 ModelSim 工作区 sim 选项卡的层次结构中单击 c: cache 前面的加号"+"展 开,单击 c: cache 下一级的 s0: cache\_set(only),则 source 窗口显示了 cache\_set 实体的 VHDL 代码。

(10) 结束仿真,其执行命令为 quit-force。

5. 调试 VHDL 仿真

1) 准备仿真

(1) 建立一个新的工作目录,复制< ModelSim 安装目录>\Modeltech\_6.4a\ examples 目录中的 gates.vhd、adder.vhd 和 testadder.vhd 文件到该目录下。

(2) 启动 ModelSim 软件,选择 File→Change Directory 命令,将新建目录设置为当前工作目录。

(3) 在主窗口中选择 File→New→Library 命令,在新目录下建立一个设计库 library\_2 来保留编译结果,或在命令行中输入 vlib library\_2 命令。

(4) 在 ModelSim 主窗口命令行操作区或 UNIX/DOS 命令行中,输入下面的命令将 源文件编译到新的设计库中:

vcom - work library\_2 gates.vhd adder.vhd testadder.vhd(回车)

(5)映射新库到工作库。可以直接编辑 modelsim. ini 文件来生成映射,或者用 vmap 命令生成一个逻辑库名字来完成:

vmap work library\_2(回车)

ModelSim 自动修改 modelsim. ini 文件。

(6) 在 ModelSim 主窗口工作区的 Library 选项卡中,单击 work 库前面的加号"+" 展开,可以看到所有设计单元。

(7) 用鼠标左键双击 work 库中名为 test\_adder\_structural 的配置,在工作区中出现 一个新的 sim 选项卡,如图 5.29 所示;也可以直接在命令行操作区输入下面的命令:

```
vsim-tnswork.test_adder_structural(回车)
```

ModelSim ALTERA 6.4a File Edit View Compile Simulate Add TI	ranscript T <u>o</u> ols Layo <u>u</u> t <u>W</u> indow <u>H</u> elp	• ×
□・☞ □ ∞ ●   ※ № @ □ □   4           ↑ ← →   目           100 ps → 目           X+X           100 ps → 目           1 Layout           Simulate	E B M   Heb B B   20 H 20 M B D D D D D D D D D D D D D D D D D D D	
Workspace       ⇒ ± d at x         VInstance       Design unit         Testbench       testbench( Architectur)         Test       test         Uut       addern(stru Architectur)         Test       testbench( Process)         standard       standard         Package       std_logic_1164         gates       gates	Transcript ::::::::::::::::::::::::::::::::::::	
	ModelSim.	

图 5.29 载入配置

(8) 在主窗口中选择 View→All Windows 命令,或输入 view \* 命令,打开所有仿真窗口。

(9) 在 signals 窗口中选择 Edit→Select All 命令,选中所有信号,然后将它们拖动到 list 窗口中。

该操作与在 signals 窗口中选择 Add→List→Signals in Region 命令或输入 add list \* 命令的结果相同。

(10) 以同样的方法,把信号加到 wave 窗口中。

(11) 在 ModelSim 主窗口的运行时间长度中设置 1000ns。

2) 运行调试仿真

(1)选取 Run 命令,运行仿真。主窗口中的一条消息将通报有一个判断错误,如 图 5.30 所示。执行后面的步骤查找错误。

ModelSim ALTERA	6.4a mpile <u>S</u> imulate A <u>d</u> d S	tṟucture Tools Layou॒t Window <u>H</u> elp	- • ×
<b>□•☞■</b> ◎◎ <b>↑</b> ◆ ⇒   IF [ 10	· ※ 哈德 오오   A   · ※ 哈德 오오   A	E= B• ₩   Hep 8   @ ₩ ₩ ₩ ₩ ₩ ₩   P• 0• 0• 10 10 0 0 0 0 0 0 0 0 0 0 0 0 0	a
X4 ▶X ⊡ ⊡ Ka	Layout Simulate	Transcript	;±a
v Instance v Instance v I testbench v I uut v test standard std_logic_1164 gates	Design unit Design unit testbench( Architectun addem(stru Architectun testbench( Process standard Package std_logic_1 Package gates Package	VSIM 20> run # *** Error: Sum is 00000111. Expected 00001000 # Time: 600 ns Iteration: 0 Instance: /testbench # ** Note: There were ERRORS in the test. # Time: 1 us Iteration: 0 Instance: /testbench VSIM 21> R Transcript	ـــــــــــــــــــــــــــــــــــــ
·	Memories	ModelSim.	
Now: 1,000,200 ps Delta:	0 sim:/testbench		

图 5.30 运行消息提示

第

13章 Quartus II软件第三方工具

- EDA技术与SoC设计应用

(2) 改变仿真判断选项,在主窗口中选择 Simulate→Runtime Options 命令,弹出如 图 5.31 所示的 Runtime Options 对话框。

Immediate Assertion Break Severity	No Message Display	For	
C Fatal	VHDL	Verilog	
C Failure	Failure	Fatal	
C Error	Error	Error	
C Warning	Warning	Warning	
C Note/ Info	Note	Info	

图 5.31 Runtime Options 对话框

(3) 选取 Assertions 选项卡。在 Break on Assertion 栏中改变选择为 Error,并单击 OK 按钮。该选项将使仿真停在 HDL 判断语句上。

(4) 选取 Restart 重新开始仿真。确定 Restart 对话框中的所有条目被选,然后单击 Restart 按钮。

(5) 选取 Run 命令,可以看到 source 窗口中的箭头指向判断语句,在 variables 窗口 中可以看到 i=6,这表示仿真停留在测试模式环路的第六次重复中,如图 5.32 所示。

(6) 在 variables 窗口中单击加号"+"展开名为 test\_patterns 的变量,也要展开排列 在 test\_patterns 下的第六次记录,如图 5.33 所示。

M Locals	
<u>File Edit View Ad</u>	d <u>W</u> indow
Locals	₩₩ ¥ ¥ X
• # # E %	
▼ Name	Value
adder8	
🔶 n	8
test_patterns	{{0000000} {000000
∎-<>> to_char	UX01ZWLH-
— test —	
+ vector	{00000101} {0000000
found_error	false
— loop —	
	6

M Locals		- • ×
<u>File Edit View A</u>	dd <u>W</u> indow	
Locals	9000	
B # 5 B		
▼ Name	Value	-
	{00000001} {0000000 {00001010} {0000001 {00000011} {0000001 {00000101} {00000000 00000101 {00000000 00000101 1 000001000 0	
□> (7)	{00000011} {1111110 {00000011} {1111110	•

图 5.32 调试中的 variables 窗口

图 5.33 展开第六次记录

判断表明了 signals 窗口中的 sum 不等于 variables 窗口中的 sum 字段。输入 a、b 和 cin 的和应该等于输出 sum,但是在测试向量内有一个错误。为了改正这个错误,需要 重新仿真且修改测试向量的初始值。

- (7) 执行 restart-f 命令。参数-f 使 ModelSim 不出现确认对话框就重新仿真。
- (8) 在 process 窗口中选取 test/testbench 过程更新 variables 窗口,如图 5.34 所示。

<u>File Edit View V</u>	<u>W</u> indow				
Processes (Active) ===					년 4월 1
A A A C	iontains 🔎	a			
Name	Type (filtered)	State	Order	Parent Path	
test	VHDL Process	Active	1	/testbench	
🎱 p1	VHDL Process	Ready	2	/testbench/uut/g	
🎱 p1	VHDL Process	Ready		/testbench/uut/g	
🌒 p1	VHDL Process	Ready		/testbench/uut/g	
p1	VHDL Process	Ready		/testbench/uut/g	
🌒 p1	VHDL Process	Ready		/testbench/uut/g	
p1	VHDL Process	Ready		/testbench/uut/g	
p1	VHDL Process	Ready		/testbench/uut/g	
🌒 p1	VHDL Process	Ready		/testbench/uut/g	
🌒 p1	VHDL Process	Ready		/testbench/uut/a	

图 5.34 process 窗口

(9) 再次展开 variables 窗口中的 test\_patterns 和 test\_patterns[6]。单击变量名字, 高亮显示.sum 记录,然后选择 variables 窗口中的 Edit→Change...命令,弹出 Change Selected Variable 对话框,如图 5.35 所示。

Variable Name:	/testbench/test/test_patterns(6).sum		
Value:	00001000	_	

图 5.35 修改 variables 窗口中记录值

(10) 在 Change Selected Variable 对话框中,把 Value 中的数值最后四位(1000) 替换为 0111,并单击 Change 按钮。(这只是暂时编辑,必须用文本编辑器永久地改变源代码。)

(11) 选取 Run 命令运行仿真。

这样,仿真运行时就不会报错了。

6. 运行批处理模式(batch-mode)仿真

批处理模式仿真必须运行在 DOS 或 UNIX 提示符下。除非特殊说明,该部分所提到的命令都是在 DOS 或 UNIX 命令提示符下输入的。

(1) 建立一个新目录,复制< ModelSim 安装目录>\Modeltech\_5.8d\examples\ counter.vhd 文件到该目录下。

(2) 将新建目录设置成当前工作目录。

(3) 生成一个新的设计库。在新建目录的 DOS 或 UNIX 命令提示符后输入 vlib work 后按回车键。

(4) 映射库:在 DOS 或 UNIX 命令提示符后输入 vmap work work 后按回车键。

(5) 编译源文件: 在 DOS 或 UNIX 命令提示符后输入 vcom counter. vhd 后按回 车键。

(6)使用宏文件为计数器提供激励。这里我们使用 ModelSim 提供的宏文件,复制

第5章

Quartus II 软件第三方工具

----- EDA技术与SoC设计应用

example\stim. do 文件到当前工作目录中。

(7) 使用编辑器建立批处理文件,内容为

```
add list - decimal *
do stim.do
write list counter.lst
quit - f
```

保存批处理文件到当前目录,命名为 yourfile。 (8)执行下面的命令,运行批处理模式仿真:

```
vsim-do yourfile-wlf saved.wlf counter-c(回车)
```

在名为"counter"的设计单元调用 vsim 仿真器。

通过-wlf这个可选项通知仿真器在名为 saved. wlf 的日志文件中保存仿真结果。

运行 yourfile 批处理设置:值以十进制的方式列示出来;执行名为 stim. do 的激励,

并将结果写到名为 counter.lst 的文件中。默认设计名为 counter。

(9) 浏览保存在 saved. wlf 文件中的仿真结果:

```
vsim-view saved.wlf(回车)
```

(10) 在 ModelSim 软件的主窗口中选择 View 菜单打开 signals、list 和 wave 窗口, 或在 ModelSim 命令行操作区输入下面的命令:

view signals list wave(回车)

(11) 在窗口中放置信号,在 ModelSim 命令行操作区输入下面的命令:

```
add wave *(回车)
add list *(回车)
```

(12) 在 Variables 窗口中检验保存的仿真结果。

(13) 结束仿真,其执行命令为

quit -f(回车)

### 5.3 在 ModelSim SE 中指定 Altera 的仿真库

图 5.36 所示为 ModelSim SE 6.4a 版的启动画面。

以下讲述如何在 ModelSim SE 中创立 Altera 的仿真库(在 Quartus Ⅱ里提取库)。

(1) 启动 ModelSim SE 仿真工具,在主窗口中选择 File→Change Directory 命令,将 工作目录改变到想要存放仿真库的目录下,如图 5.37 所示,单击 OK 按钮确定。

(2) 在主窗口中选择 File→New→Library 命令,弾出 Create a New Library 窗口,将 Create 选项设置为 a new library and a logical mapping to it,输入库的名字(例如 Altera\_ song),如图 5.38 所示,单击 OK 按钮确定。

这个操作过程实质上相当于在 ModelSim 主窗口的命令操作区中输入了 vlib 和 vmap 命令。



图 5.36 ModelSim SE 6.4a 版的启动画面

-

图 5.37 指定工作目录

Create	
	a new library and a logical mapping to it
C	a map to an existing library
Libr	ary Name:
lter	a_song
Libr	ary Physical Name:
Alter	a_song

图 5.38 Create a New Library 对话框

(3)选中主窗口 Library 选项卡中的 Altera\_song,选择 Compile → Compile 命令,弹 出如图 5.39 所示的对话框。将查找范围指定到< Quartus II 安装目录>\eda\sim\_lib 文 件夹下,对以下 8 个文件分两次进行编译: 220model. v,220model. vhd,220model\_87. vhd, 220pack. vhd, altera \_ mf. v, altera \_ mf. vhd, altera \_ mf \_ 87. vhd, altera \_ mf \_ components. vhd。编译顺序为 220pack. vhd 和 altera\_mf\_components. vhd 先编译,其他 文件后编译。单击 Compile 按钮编译,完成后单击 Done 按钮。

(4) 如果要后仿真,就把要用的系列库再编译。例如用 Altera 的 Cyclone 系列,就再编译 cyclone\_components. vhd、cyclone\_atoms. vhd 和 cyclone\_atoms. v 三个文件。

(5)将 ModelSim SE 根目录下的配置文件 modelsim. ini 的属性由只读改为可写,这

第5章

Quartus II 软件第三方工具

-----EDA技术与SoC设计应用

Library: Altera_song	
查找范围 (I): 📗 Altera_song	▼■℃⊡▼
名称 ^	修改日期 ^
altera_mf.v	2009/2/26 0:44
altera_mf.vhd	2009/2/26 0:45
altera_mf_87.vhd	2009/2/26 0:45
altera_mf_components.vhd	2009/2/26 0:42 -
	*
【件名谜): 「.v" "altera_mf.vhd" "altera_m	f_87. vhd" Compile
(件类型(I):  HDL Files (*.v;*.vl;*.vhd;*.v	hdl;*.v 💌 Done

个操作是为了使软件可以记录仿真库建立的路径以及映射关系。以后每次启动 ModelSim 仿真工具时,软件都会根据 ini 文件中的配置寻找仿真库,并形成映射关系。 (6) 保存文件并退出。以后对 Altera 的设计进行仿真都不需要再进行库的处理了。

图 5.39 选择 Altera 编译库