

0. Write-Ahead

0.0 学生信息

姓名： (因上传github仓库已隐藏)

学号： (因上传github仓库已隐藏)

0.1 baseline的解释

首先，如何理解整个项目的运行流程，参考这篇非常好的博文，当时看完瞬间解决了我的疑惑，也和本个baseline内容非常契合，博客链接如下。

[modelsim仿真中do文件的写法技巧](#)

使用verilog语言描述处理器并进行仿真的流程如下（注意本次试验只需要Modelsim软件即可，不需要像本科时下载Quartus，Modelsim可以写代码，编译和仿真，虽然我是用vscode写的）

1. 使用编辑器编写verilog代码
2. 编译verilog代码
3. 使用modelsim进行仿真

对于第一步，使用modelsim自带的编辑器或者vscode甚至文本文档都可以。

对于第二步，具体分为建立工程和工程库（vlib命令和vmap命令），加载设计文件和编译源文件（vlog命令）。这些步骤可以使用modelsim的图形化界面进行，也可以直接使用命令行执行（命令行窗口打开路径：view->Transcript），baseline非常贴心的写好了所有步骤的命令，并且放在了do文件中，只需要在命令行中执行 do D:/pe_exp/sim/build_0.do，就完成了第二步的所有步骤。

下面简单讲讲仿真的步骤。首先我们要对一个设计进行仿真呢，我们一般需要进行以下几个步骤：

- ①创建一个工程和工程库；
- ②加载设计文件（包括你编写好的testbench）；
- ③编译源文件；
- ④运行仿真，并查看结果；
- ⑤最后进行工程调试。

而do文件，就是把上述的步骤①---④用tcl脚本语言来编写出来，让Modelsim来运行该do文件宏命令，并自动执行仿真的步骤。这种好处也许在小设计中没怎么表现，但是如果在一个大的工程中，常常需要对一个设计单元进行反复的调试和仿真，但是仿真时的设置是不变的，这时如果使用了do文件，把仿真中使用到的命令都保存下来了，就可以节省大量的人力，提高了工作效率。

编写名为**counter.do**的文件，其内容为下：

vlib work (对应仿真步骤①：新建work库。该命令的作用是在当前目录下建立一个work目录，请注意不要直接在windows中新建一个work的文件夹，因为用操作系统建立的work文件夹并没有ModelSimSE自动生成的_info文件。)

vmap work work (对应仿真步骤①：该命令的作用是将目前的逻辑工作库work和实际工作库work映射对应。也可以直接用指令“vmap work”表示将work库映射到当前工作目录下。)

vlog counter.v counter_tb.v (对应仿真步骤②③：编译counter.v和counter_tb.v文件，默认编译到work库下。该命令的作用是编译这些文件，要注意的是文件可以单独分开编译，但是一定要先编译被调用的文件。假如是VHDL文件，只需要把指令vlog换成vcom即可。)

vsim work.counter_tb -t 1ns (对应仿真步骤④：仿真work库中名为counter_tb的模块，最小时间单位为1ns。)

add wave/counter_tb/* (该命令的作用是将testbench文件camera_tb.v中模块camera_tb下所有的信号变量加到波形文件中去，注意在“*”前要加空格。这时候你也可以看到wave文件被打开。当然也可以单个信号的添加，例如添加时钟：addwave clk 等等。)

run 2000 (该命令的作用是运行2000个单位时间的仿真。也可以用**run-all**命令来一直仿真下去。)

这时候就可以在wave窗口文件中看到你的仿真结果。当然也可以观察其它窗口的结果，用**view***命令显示。

view*命令可以观察包括signals、wave、dataflow等窗口文件，也可以分别打开。例如用**view signals**来观察信号变量。

对于第三步，同样是可以使用modelsim的图形化界面进行，也可以直接使用命令行执行，当然我们选择命令行执行，使用vsim命令，baseline的**sim_run_x.do**中写好了参数，直接执行do文件更方便更快捷。

在wave中添加需要观察的变量，在命令行窗口中执行**run -all**即可。

其次，浅浅解释一下**tp_top_2.v**，即testbench生成输入信号和读取输入数据和测试模块。老师使用了**\$random()**来随机控制是否信号valid。（后面懒得写了，基本就是四套测试数据，weight和neru的ready控制读入下一行向量，inst的ready控制读入下一个矩阵，也即下一个测试数据）

最后，总的实验流程：

1. 使用vscode完成代码
2. 打开modelsim，找到transcript窗口，命令行输入**do D:/pe_exp/sim/build_0.do**
3. 找到transcript窗口，命令行输入**do D:/pe_exp/sim/sim_run_0.do**
4. 在sim窗口选择要观察的模块，然后去object窗口选择该模块要观察的波形
5. 找到transcript窗口，命令行输入**run -all**
6. 缩放波形，观察结果

0.2 baseline的问题

baseline是无法直接运行的，有一些错误需要更改。

1. **sim_run_x.do**中，**vsim**指令的tp_top表示仿真顶层模块名字，但是**tp_top_x.v**文件中的顶层模块名字并没有数字后缀，所以需要把三个**tp_top_x.v**文件中的模块名字加上数字后缀（当然也可以改三个**sim_run_x.f**文件中参数，等效）。
2. **compile_2.f**中，添加**D:/pe_exp/src/matrix_pe/matrix_pe.v**，该文件也是实验三的文件，但是却没有写。

3. 群里同学说data文件夹下的数据有问题，需要重新生成。我没有测试是否有问题，是直接运行了一下 `data_gen.py`

0.3 实验手册的问题

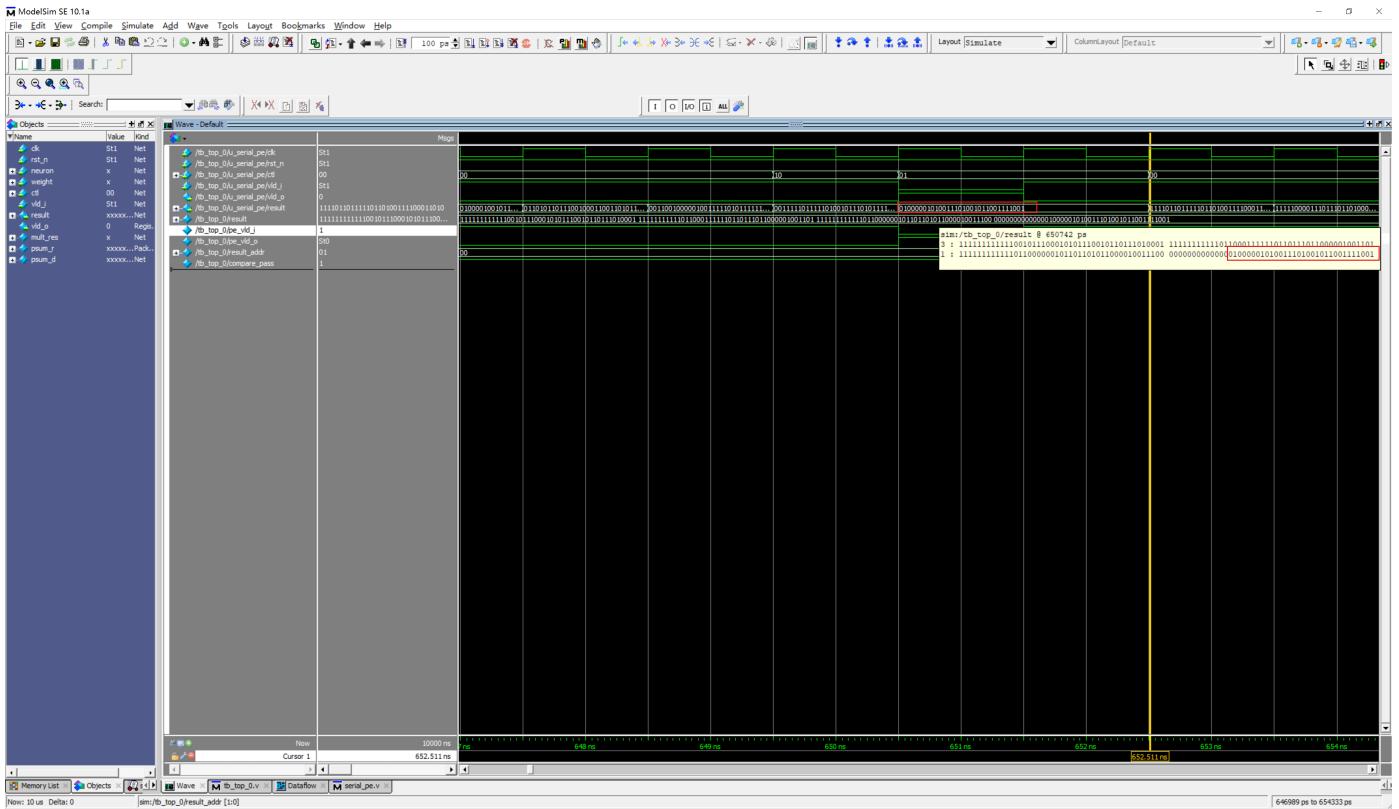
1. 6.5.3.1中，"do path to build/build.do"，其中path to build不是指令，而是一种说明，坑了我好久，应该是 `do <path store your build>/build.do`
2. 6.5.3.4中，运行仿真应该是 `run -all`，而不是 `run all`（虽然后面写的是正确的），运行后，会有提示框，询问是否finish，一定要点否，否则会关闭整个modelsim软件。

1. Experiment

实验结果不再进行过多说明，只展示每个实验结果检查正确的截图，和对应波形图位置。

1.0 exp0

```
# ** Warning: (vlib-34) Library already exists at "D:/pe_exp/sim/work".
#
# Modifying modelsim.ini
# Model Technology ModelSim SE vlog 10.1a Compiler 2012.02 Feb 22 2012
# -- Compiling module tb_top_0
# -- Compiling module serial_pe
#
# Top level modules:
#   tb_top_0
VSIM 15> do D:/pe_exp/sim/sim_run_0.do
# vsim +nowarnTSCALE -lib work -c -novopt tb_top_0
# Refreshing D:/pe_exp/sim/work.tb_top_0
# Loading work.tb_top_0
# Refreshing D:/pe_exp/sim/work.serial_pe
# Loading work.serial_pe
add wave \
sim:/tb_top_0/WIDTH \
sim:/tb_top_0/clk \
sim:/tb_top_0/rst_n \
sim:/tb_top_0/inst \
sim:/tb_top_0/neuron \
sim:/tb_top_0/weight \
sim:/tb_top_0/result \
sim:/tb_top_0/inst_addr \
sim:/tb_top_0/iter \
sim:/tb_top_0/neuron_addr \
sim:/tb_top_0/weight_addr \
sim:/tb_top_0/pe_inst \
sim:/tb_top_0/pe_weight_line \
sim:/tb_top_0/pe_neuron_line \
sim:/tb_top_0/pe_weight \
sim:/tb_top_0/pe_neuron \
sim:/tb_top_0/pe_ccl \
sim:/tb_top_0/pe_vld_i \
sim:/tb_top_0/pe_result \
sim:/tb_top_0/pe_vld_o \
sim:/tb_top_0/result_addr \
sim:/tb_top_0/compare_pass
VSIM 17> run -all
# INFO: num.0 result is correct.
# INFO: num.1 result is correct.
# INFO: num.2 result is correct.
# INFO: num.3 result is correct.
# ** Note: $finish : D:/pe_exp/sim/tb_top_0.v(23)
#   Timer 10 us Iteration: 0 Instance: /tb_top_0
# 1
# Break in Module tb_top_0 at D:/pe_exp/sim/tb_top_0.v line 23
VSIM 18>
```

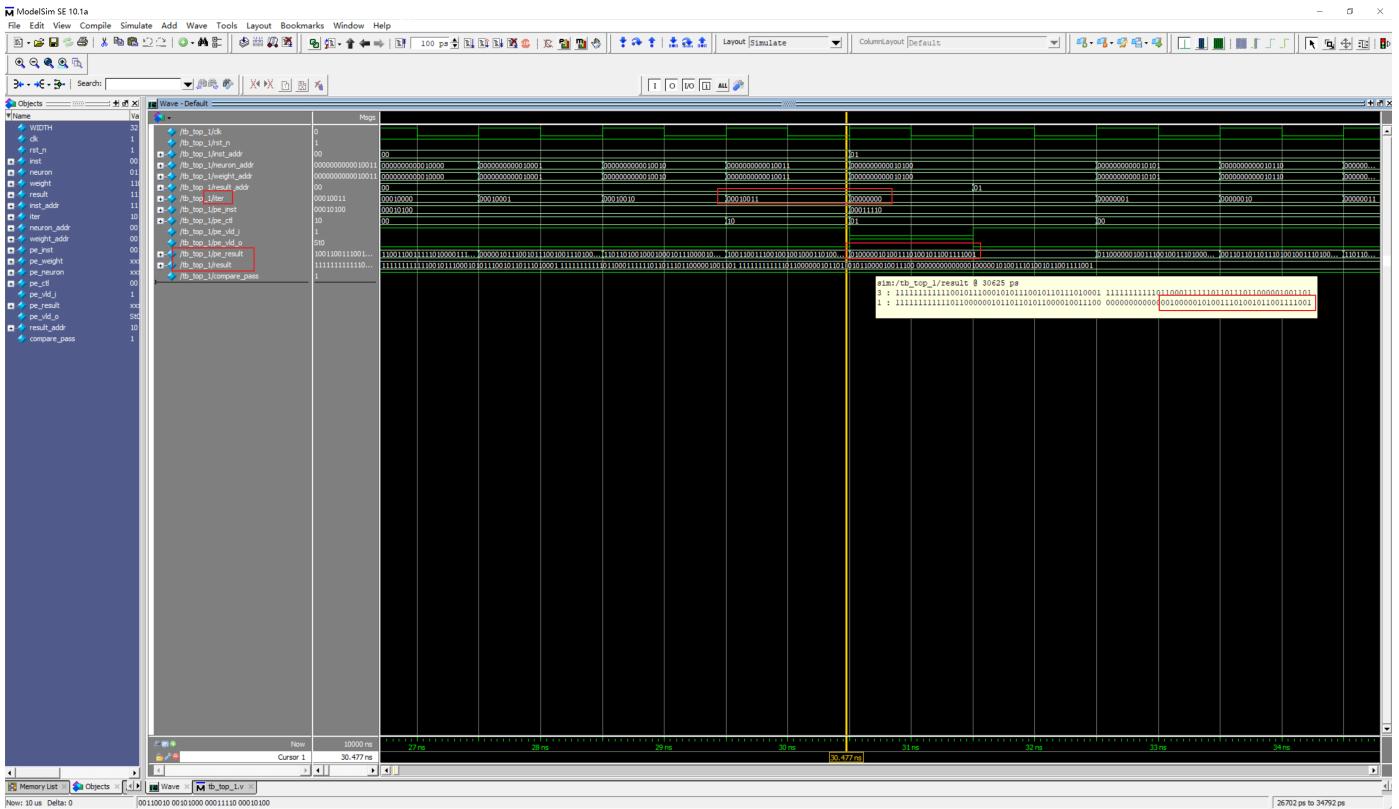


1.1 exp1

```

Transcript
File Edit View Bookmarks Window Help
Transcript
Reading D:/Modelsim/tcl/vsim/pref.tcl
// ModelSim SE 10.1a Feb 22 2012
//
// Copyright 1991-2012 Mentor Graphics Corporation
// All Rights Reserved.
//
// THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION
// WHICH IS THE PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS
// LICENSORS AND IS SUBJECT TO LICENSE TERMS.
//
ModelSim> do D:/pe_exp/sim/build_1.do
D:/pe_exp/sim
** Warning: (vlib-34) Library already exists at "D:/pe_exp/sim/work".
#
Modifying modelsim.ini
Model Technology ModelSim SE vlog 10.1a Compiler 2012.02 Feb 22 2012
-- Compiling module tb_top_1
-- Compiling module parallel_pe
-- Compiling module pe_mult
-- Compiling module pe_acc
#
Top level modules:
# tb_top_1
ModelSim> do D:/pe_exp/sim/sim_run_1.do
# vsim +nowarnTSCALE -lib work -6 -novopt tb_top_1
# Refreshing D:\pe_exp\sim\work\tb_top_1
# Loading work.tb_top_1
# Refreshing D:\pe_exp\sim\work\parallel_pe
# Loading work.parallel_pe
# Refreshing D:\pe_exp\sim\work\pe_mult
# Loading work.pe_mult
# Refreshing D:\pe_exp\sim\work\pe_acc
# Loading work.pe_acc
VSIM3> run -all
# INFO: num_0 result is correct.
# INFO: num_1 result is correct.
# INFO: num_2 result is correct.
# INFO: num_3 result is correct.
** Note: $finish : D:/pe_exp/sim/tb_top_1.v(23)
# Time: 10 us Iteration: 0 Instance: /tb_top_1
# 1
# Break in Module tb_top_1 at D:/pe_exp/sim/tb_top_1.v line 23
VSIM4>

```



1.2 exp2

```

Transcript
File Edit View Bookmarks Window Help
Transcript
File Edit View Bookmarks Window Help
D:\pe_exp\sim
** Warning: (vlib-34) Library already exists at "D:\pe_exp\sim\work".
#
# Modifying modelsim.ini
# Model Technology ModelSim SE vlog 10.1a Compiler 2012.02 Feb 22 2012
-- Compiling module tb_top_2
-- Compiling module parallel_pe
-- Compiling module pe_mult
-- Compiling module pe_acc
-- Compiling module matrix_pe
#
# Top level modules:
# tb_top_2
VSIM13> do D:\pe_exp\sim\sim_run_2.do
# vsim -nowarnIFSCALE -lib work -c -novopt tb_top_2
# Refreshing D:\pe_exp\sim\work\tb_top_2
# Loading work\tb_top_2
# Refreshing D:\pe_exp\sim\work\matrix_pe
# Loading work\matrix_pe
# Refreshing D:\pe_exp\sim\work\parallel_pe
# Loading work\parallel_pe
# Refreshing D:\pe_exp\sim\work\pe_mult
# Loading work\pe_mult
# Refreshing D:\pe_exp\sim\work\pe_acc
# Loading work\pe_acc
add wave sim:/tb_top_2/*
** Warning: (vsim-WLF-5000) WLF file currently in use: vsim.wlf
#
# File in use by: DELL Hostname: DESKTOP-9539SJ5 ProcessID: 17184
#
# Attempting to use alternate WLF file ".\wlft884mqb".
** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
#
# Using alternate file: .\wlft884mqb
#
add wave sim:/tb_top_2/u_matrix_pe/*
add wave -position end sim:/tb_top_2/result
VSIM17> run -all
# INFO: num_0 result is correct.
# INFO: num_1 result is correct.
# INFO: num_2 result is correct.
# INFO: num_3 result is correct.
# ** Note: ofinish : D:\pe_exp\sim\tb_top_2.v(23)
# Time: 10 us Iteration: 0 Instance: /tb_top_2
#
# Break in Module tb_top_2 at D:\pe_exp\sim\tb_top_2.v line 23
# Causality operation skipped due to absence of debug database file
add wave -position end sim:/tb_top_2/u_matrix_pe/pe_vid_i
VSIM18>

```

