

高效开发 VerilogHDL 项目

V1.0@2014.11.22

作者：刘乾@北航计算机学院

编辑：高小鹏@北航计算机学院

写在开头

我一直很想整理出一份这样的类似解决方案的文档来帮助大家对 ISE 进行更快的入门与不必要纠结一些很无脑的错误。

现在，我把一些经验总结后分享给大家，让更多的人对这款软件更快地上手。

同时，我也非常希望有更多的同学能分享你的经验！

引言：关于 ISE 的问题库的使用说明

首先声明这是 ISE 自带的帮助库：<http://www.xilinx.com/support.html>

从 ISE 里进入的方式是：Help → Xilinx on the Web → Support and Services

一、ISE 常见问题

很多同学表示从来没有用过这个帮助库，这个库是用来做什么的呢？

这个库是用来 search 一些常见的系统错误的。下面是我利用 ISE 问题库解决了的一些系统错误。

Q1：

错误提示：

ERROR:Simulator:904 –

Unable to remove previous simulation file A:/mips/mips_tb_isim_beh.exe.

Please check if you have another instance of this simulation running on your system, terminate it and then recompile your design.

System Error Message: boost::filesystemremove: %Ü¼ø·ÃÊ¡£:

"A:\mips\mips_tb_isim_beh.exe"

A1：这种错误出现是由于同时开启了对同一模块的仿真，关闭上一次仿真后的界面即

可，如果界面找不到了，请启动任务管理器，从进程中关闭 isimgui.exe 和

mips_tb_isim_beh.exe 后即可重新仿真

Q2 :

错误提示:

FATAL ERROR: PrivateChannel: Error connecting to server socket

ERROR: The simulation failed to launch for the following reason:

Failed to communicate with child process.

Please shut down ISim and retry the simulation. If the problem persists, please contact Xilinx support.

Time resolution is 1 fs

No active Database

Unable to execute live simulation command.

A2 : 这种错误一般是在高版本 14.6 中仿真时产生，并且是在 win8 里出现的。如果你是 win8 用户，可以尝试在 win8 开始的磁贴页面找到应用程序里的 ISE Design Suite 32 bit （ISE 的桌面快捷方式是默认为 64bit 的）

如果你无法使用 32 位版本的，只能使用 64 位版本的，使用如下教程可以使你在打开工程时不崩溃：

找到程序安装路径下的这两个文件夹

X:\Xilinx\14.6\ISE_DS\ISE\lib\nt64

X:\Xilinx\14.6\ISE_DS\common\lib\nt64

首先在第 1 个文件夹中，重命名 libPortability.dll 为 libPortability.dll.orig，然后复制 libPortabilityNOSH.dll 的一个副本并重命名为 libPortability.dll，这样你就又有一个 libPortability.dll 文件了。

然后在第 2 个文件夹，将之前得到的新的 libPortability.dll 覆盖到这个文件夹中。

但此解决方案只能使你打开工程时不至于崩溃或闪退，仿真时有可能依旧出现 bug。

最终解决的方案仍旧没找到，只能是尝试性的测试。希望有解决了的同学可以将解决方案 email 给我，以便我汇总在一起。

Q3 : 打不开 Isim

A3 : 同样可能是因为我们的版本的问题，请从 32 位版本快捷方式打开

还有一个解决方案如下：

打开 ISE 的目录，找到 14.6\ISE_DS\ISE\gnu 请从 MINGW 的官网下载最新版的 MINGW 覆盖该文件

Q4 :

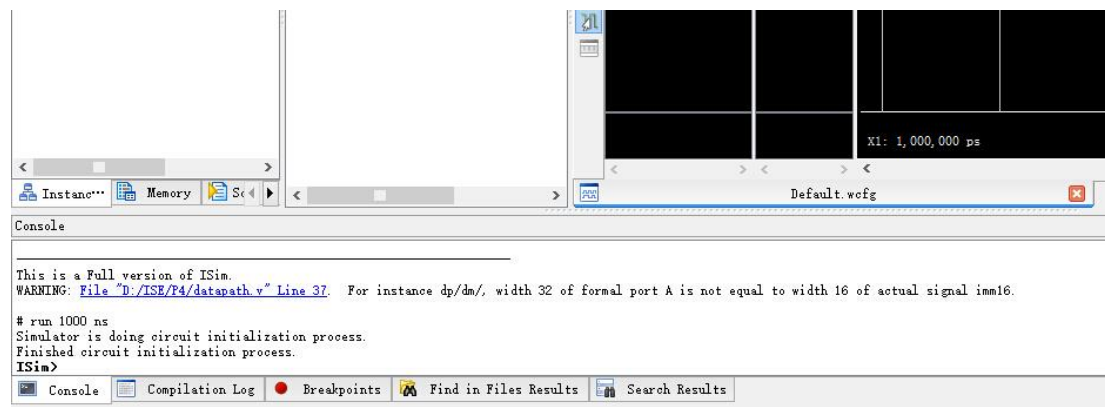
错误提示:

WARNING: File "D:/ISE/P4/datapath.v" Line 37. For instance dp/dm/, width 32 of formal

port A is not equal to width 16 of actual signal imm16.

A4：这是一个典型的位宽不匹配的错误，我们从中可以看到，在 datapath.v 文件里的 dm 模块中模块内部的定义变量 A 和外部传入的变量 imm16 的位宽是不一致的，这个虽然是警告，但是有可能就是造成波形图错乱的关键原因之一，请记住，Behavioral Check 只能检查出模块内部的显性的位宽不正确的错误，在模块中实例化子模块的时候的错误在仿真前有时候是检查不出来的！

这个常见错误原因这一的截图如下：



请一定要在仿真后看一下下面的提示信息并修改位宽。

另外注明一点，位宽的错误有可能是模块参数的顺序出错导致的！请注意模块顺序，好的代码风格应该是在实例化模块时如下：

```
Mips U_MIPS(.clk(clk),.reset(reset))
```

详情可翻阅 Verilog 教程资料进行查询。

二、仿真调试技巧

以下是我调试的过程中总结的一些小技巧，如果你觉得还有一些需要补充的请发邮件给我，邮箱在文章最后会附上，谢谢！

1. 如何把中间变量的值加入波形图？

打开仿真界面后，我们可以看到三栏，最左侧一栏下方有三个选项，分别是

Instances and Processes ,Memory,Source Files

其中我们点到第一个选项，可以看到顶层目录，点击前面的+号，可以展开目录，如果想添加某个中间变量的值入波形图的话，请点击中该变量，右键第一个选项可以将其加入波形图中。


2. 一个值出错了，我该如何知道是在哪里出错了呢？

我的思路是这样的：

如果有一个值是非预期值（比如出现 XXXX 或者一些不是你想要的值），那么我们可以这样推理，首先寻找到输出为该变量（注意是直接输出该变量）的器件，这里的器件包括

多路选择器,比如我有一个多路选择器是根据 MemtoReg 信号的值选择写入 GPR 的数据,那么这时候如果我的 **WriteData** 是 **XXX**, 那么我们就考虑有以下三种情况:



- 1) **DM** 的 **readdata** 值是 **XXX**;
- 2) **ALU** 的结果是 **XXX**;
- 3) 控制信号 **MemtoReg** 是 **XXX**

请通过小技巧 1 的方式将相关的变量值加入波形图, 点击 , 或者通过菜单栏里的 Simulation→Restart 再次仿真, 然后注意观察刚才出现错误的指令的相应位置的值哪个是错误的, 如果我们观察到 ALU 的结果是 XXX, 那我们继续考虑是否是 ALU 的输入或者控制信号有错误, 溯源以后可以发现源头的错误值是在哪里产生的。

源头错误的产生原因往往有以下几点:

- 1) **未能成功传参**。请检查在你的顶层模块实例化模块的模块参数变量名是否全部写对? 请尤其注意错别字母和大小写的区分! 如果要查找某一变量名所在位置的话, 可以使用快捷键 **ctrl+F** 进行查找。
- 2) **位宽不匹配**。确实存在一种情况是在仿真和 check 时都不会报错的位宽检查, 比如我定义了一个 **wire [31:0] aluout** 当其要传入一个模块时, 该模块本来相应位置的位宽应该为 32 位, 但是我们如下定义 **ALU my_alu(XX..., aluout[31:2], XXX)** 本来 alu 的输出就应该是 32 位的, 我们定义也是 32 位的, 但在使用时错误地使用了 **aluout** 的 31:2 位作为变量参数, **这时候我们的 ISE 不会报错!** 在我电脑上是不报错的, 其他电脑测过两台, 也是不会报错的, 所以请格外注意!
- 3) **未定义该变量**。请注意检查你的变量是否在模块的开始定义过, 如果定义过, 请检查是否存在语法错误, 常见的初级语法错误将在后面进行说明。

3. 合理设置仿真时间

请多使用该符号  而不要使用 , 第一个代表的是执行一个你的 runtime, runtime 默认是 1us, 可在仿真时进行调整, 请注意把 testbench 里的 clk 保持不变的延迟时间和 runtime 维持比例不要过小! 否则将一次跳完很多指令或一次, 不利于检查你的错误!

三、调试过程中常见问题

1. clk 的值为 xx, 线为黄色, 无波形图

这种状态很多情况下都有可能会导致, 请尝试以下方法进行改进看是否解决你的问题:

- 1) **观察你是否使用了 assign 连续赋值了 reg 定义的变量**

请记住: wire 对应于连续赋值, 如 **assign**; reg 对应于过程赋值, 如 **always**, **initial**

2) 观察是否在组合逻辑电路中使用了 `always@(posedge clk)` 类似的描述, 为了确保不是该问题导致的原因, 建议所有的组合逻辑建模过程都使用 `assign!` (高老板也强调过这一点)

实际上我使用 `always@(*)` 也可以描述组合逻辑, 但是会有一些风险, 如果不使用 `*` 的话请注意要把赋值表达式右端的变量全部加入敏感列表内!

关于风险的问题我自己的一点想法:

这个问题涉及到一个 `always@(*)` 和 `assign` 表示组合逻辑电路时的不定态问题。举个例子:

```
wire a;
reg b;
assign a = 1'b0;
always@(*)
    b = 1'b0;
```

在这种情况下, 做仿真时 `a` 将会正常为 0, 但是 `b` 却是不定态。这是为什么? `verilog` 规定, `always@(*)` 中的 `*` 是指该 `always` 块内的所有输入信号的变化为敏感列表, 也就是仿真时只有当 `always@(*)` 块内的输入信号产生变化, 该块内描述的信号才会产生变化, 而像

```
always@(*) b = 1'b0 ;
```

这种写法由于 `1'b0` 一直没有变化, 所以 `b` 的信号状态一直没有改变, 由于 `b` 是组合逻辑输出, 所以复位时没有明确的值 (不定态), 而又因为 `always@(*)` 块内没有敏感信号变化, 因此 `b` 的信号状态一直保持为不定态。

3) 是否是因为是初始态的原因, 没有进行 `run` (如果 `run` 之后 `clk` 发生变化就是该原因)

2. 在 ISE 仿真时我的 `pc` 始终为 XXXXXX 或 `instruction` 始终为 XXXXXXXX

A2: 最有可能的情况是 `PC` 未进行初始化, 建议在测试文件里 `reset` 一下或者可以在 `pc` 模块中对 `pc` 的值进行初始化。

如果你在你的 `pc` 模块中发现如此定义且 `reset` 后还是不可以, 那么请注意你是否有如下类似语句:

```
assign PC = _____ 或者
PC <= _____
```

如果赋值给 `PC` 不是一个 `input` 变量而是 `中间变量`, 那么需要对这些中间变量进行初始化! 完成以上内容, 一般 `PC` 的初始值会变成 0, 即仿真的第一步完成了。

3. 如果我在 `PC` 值为 0 时下一步我的 `PC` 值没有像正常的 `PC+1` 而是变成了不定值?

A3: 对于这种问题 请注意在顶层设计模块中你是否有传往 `PC` 的 `NPC` 的控制信号是否

进行了初始化？（一般初始化控制信号可以解决这个问题）

4. 我的控制信号是对的，输入也是对的，case 语句写的也是对的，为什么最后结果不对？

A4: 遇到这种情况，请检查下列是否出错：

- 1) 语句的==后面的常量值是否写清楚了位数？如果是类似于 if(memtoreg==01) 的语句，请将它改成 if(memtoreg==2'b01) 的格式以确保正确。
- 2) 请检查多个 if 条件判断是否会出现矛盾。

四、VerilogHDL 代码编写建议

以下是关于编写 VerilogHDL 代码的建议。

1. 避免引入锁存器！

当你用 always 建模组合逻辑时，为了避免引入锁存器，必须做到：

- 每一个 if 描述语句都有对应的 else 语句。
- 每一个 case 语句都对应一个 default 语句。

2. 在 always 块中是顺序执行的，所以请注意赋值顺序！

3. 在“always”块内被赋值的每一个信号都必须定义成 reg 型。

- ### 4. always 语句由于其不断重复执行的特性，只有和一定的时序控制结合在一起才有用。
- 如果一个 always 语句没有时序控制，则这个 always 语句将会发成一个仿真死锁。见下例：

```
always areg = ~areg;
```

这个 always 语句将会生成一个 0 延迟的无限循环跳变过程，这时会发生仿真死锁。

所以请注意，请一定不要单独使用 always！

5. 请注意一个模块的输入不可以为 reg 类型！

- ### 6. 输入端口可以由 wire/reg 驱动，但输入端口只能是 wire；输出端口可以是 wire/reg 类型，输出端口只能驱动 wire。

写在末尾：

本文会持续不定量地出更新版，请一定及时下载！

邮箱：372899855@qq.com

请把你所遇到的没有涉及到的一些困扰多时的问题打包发给我，我尽量反馈给高老师。

如果还有同学愿意当计组助教的，请联系我！

项塔兰 致