

## 一、安装 vivado 2019.2

正常安装 vivado 即可, [Vivado 2019 WebPACK\(免费版\) 安装教程与避雷](#) [vivado webpack-CSDN博客](#)。  
我安装的就是 WebPACK 免费版, 能满足备赛需要。

vivado还可以到杨建磊老师的ftp服务器上获取, 这里面还有些关于fpga, Xilinx的学习资源

IP: 10.130.147.228

用户名: ftpuser

密码: buaaspinact

Linux系统安装vivado后会产生桌面快捷方式, 不要管这个, 我们用终端打开vivado

```
harry@harry-virtual-machine:~/tools/xilinx/Vivado/2019.2/bin$ ./vivado
```

可能发现缺少库。 `couldn't load file "librdi_commontasks.so": libtinfo.so.5: cannot open shared object file: No such file or directory`

解决: `sudo apt-get install libtinfo5`







打开vivado后, 仿真任意工程测试, 我这边有报错 `ERROR: [XSIM 43-3409] Failed to compile generated C file xsim.dir/tb_top_behav/obj/xsim_4.c.`

解决: `sudo apt-get install libncurses5`

## 二、安装 VCS

VCS是synopsys系列软件之一, 下载链接在下面这个博客里, 也可以在[芯王国](#)公众号上找到

[Vivado 2021.2版本与VCS/Verdi 2018版本联合仿真 verdi2018 linux下载-CSDN博客](#)

- ☐  vcs\_vO-2018.09-SP2
  - ☐  verdi-2018.9
  - ☐  vcs\_mx\_vO-2018.09-SP2
  - ☐  scl\_v2018.06
  - ☐  synopsysinstaller\_v5.0.rar
  - ☐  scl\_keygen\_2030.zip
- CSDN @FPGA干货店

或者到北航云盘

<https://bhpan.buaa.edu.cn/link/AAE0F74DF381804CD0BCB7D7C0B89EC350>

文件夹名: IC\_EDA\_pack

有效期限: 2024-10-01 16:02

我们需要的只是这里的后四个

vcs\_mx\_v0-2018.09-SP2 就是 VCS，但这个版本（就是vcs后面这一串）很重要，其他版本可能无法编译 vivado ip 库，但是 vivado 的版本似乎不一定是2019.2  
scl\_v2018.06 这个在破解时需要用到  
synopsysinstaller\_v5.0.rar 这是一个安装程序，用来安装前两个程序。也就是先在某个目录下将它解压，运行它提供的脚本来安装前两个软件  
scl\_keygen\_2030.zip 这个是生成破解需要的License的工具。与前三个都放在ubuntu系统中不同，这个工具需要运行在windows系统中。

## 具体步骤

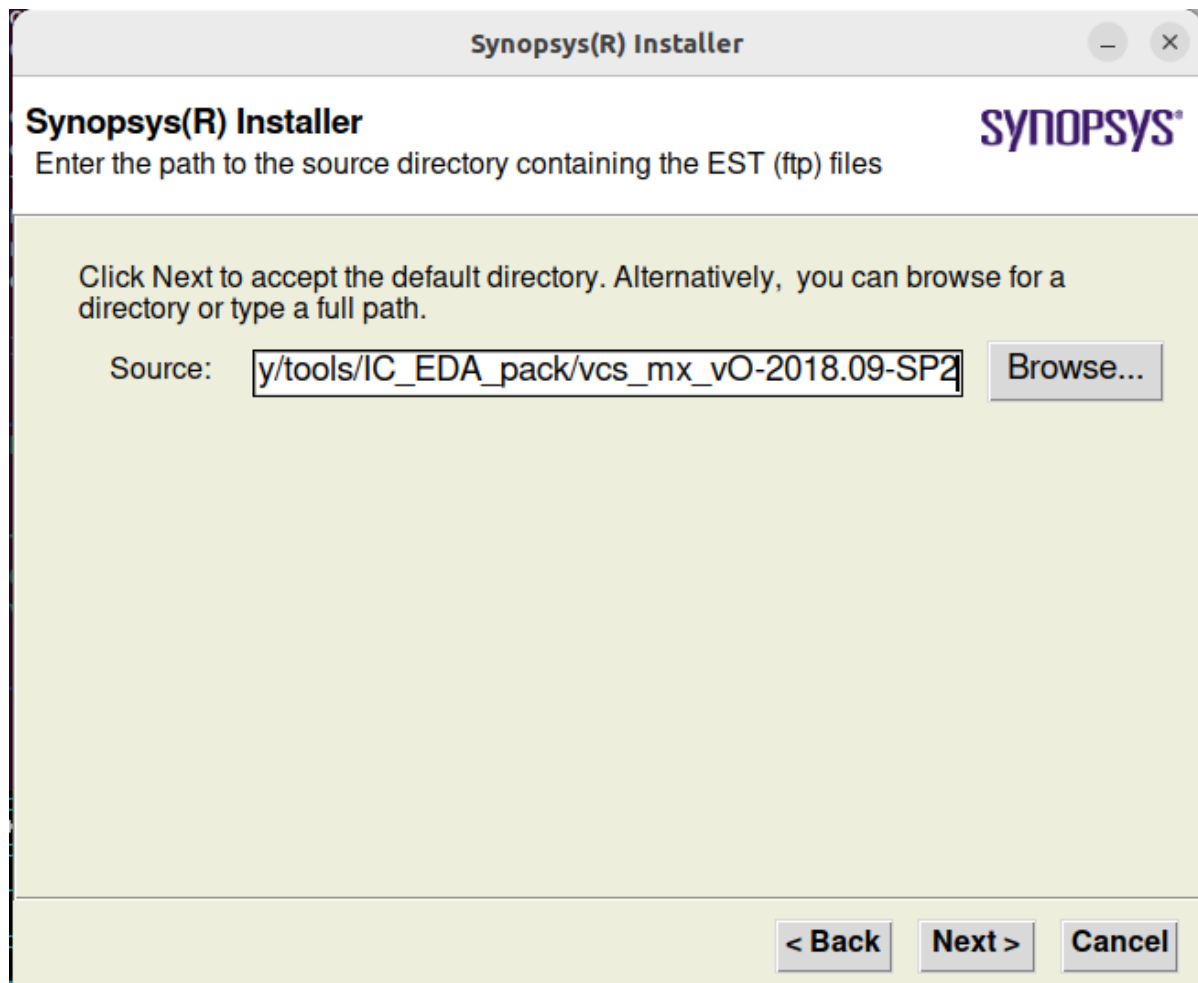
先解压synopsysinstaller\_v5.0.rar到某个目录中，进入synopsysinstaller\_v5.0目录运行 `./SynopsysInstaller_v5.0.run`，安装位置直接回车选择默认。安装后是这样的

```
harry@harry-virtual-machine:~/tools/IC_EDA_pack/synopsysinstaller_v5.0$ ls
batch_installer      installer_INSTALL_README.txt
checksum_info.txt    installer.log
container            setup.sh
container_setup.sh   snps_container_INSTALL_README.txt
doc                  snps_container_v1.0_common.spf
install_bin          SynopsysInstaller_v5.0.run
installer
```

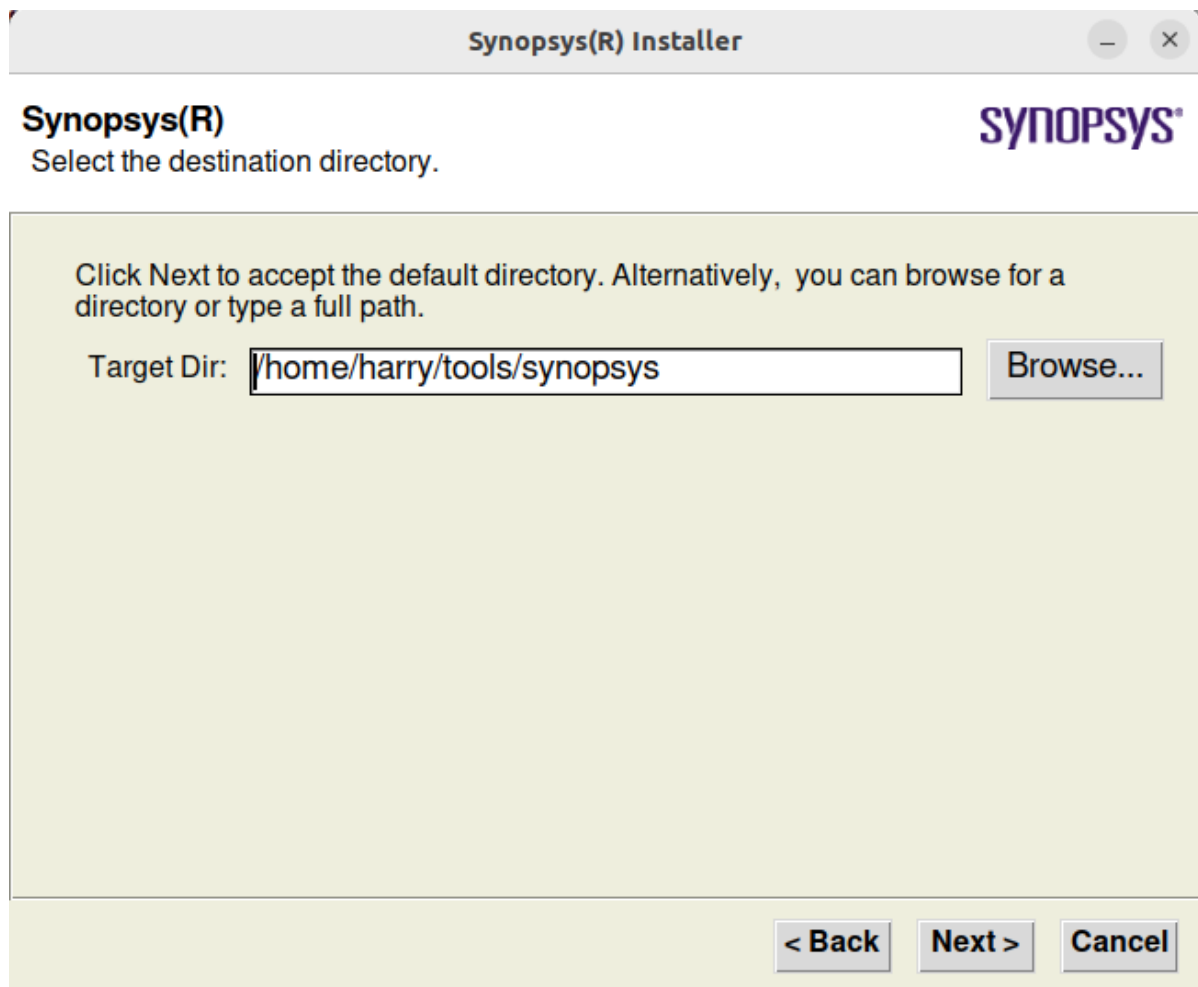
下面安装 vcs\_mx 和 scl。提前创建一个 synopsys 目录作为安装目标位置。回到 synopsysinstaller\_v5.0目录，执行 `. setup.sh`

先后填3个设置，其他直接next， `vcs_mx` || `scl` 是同一套操作

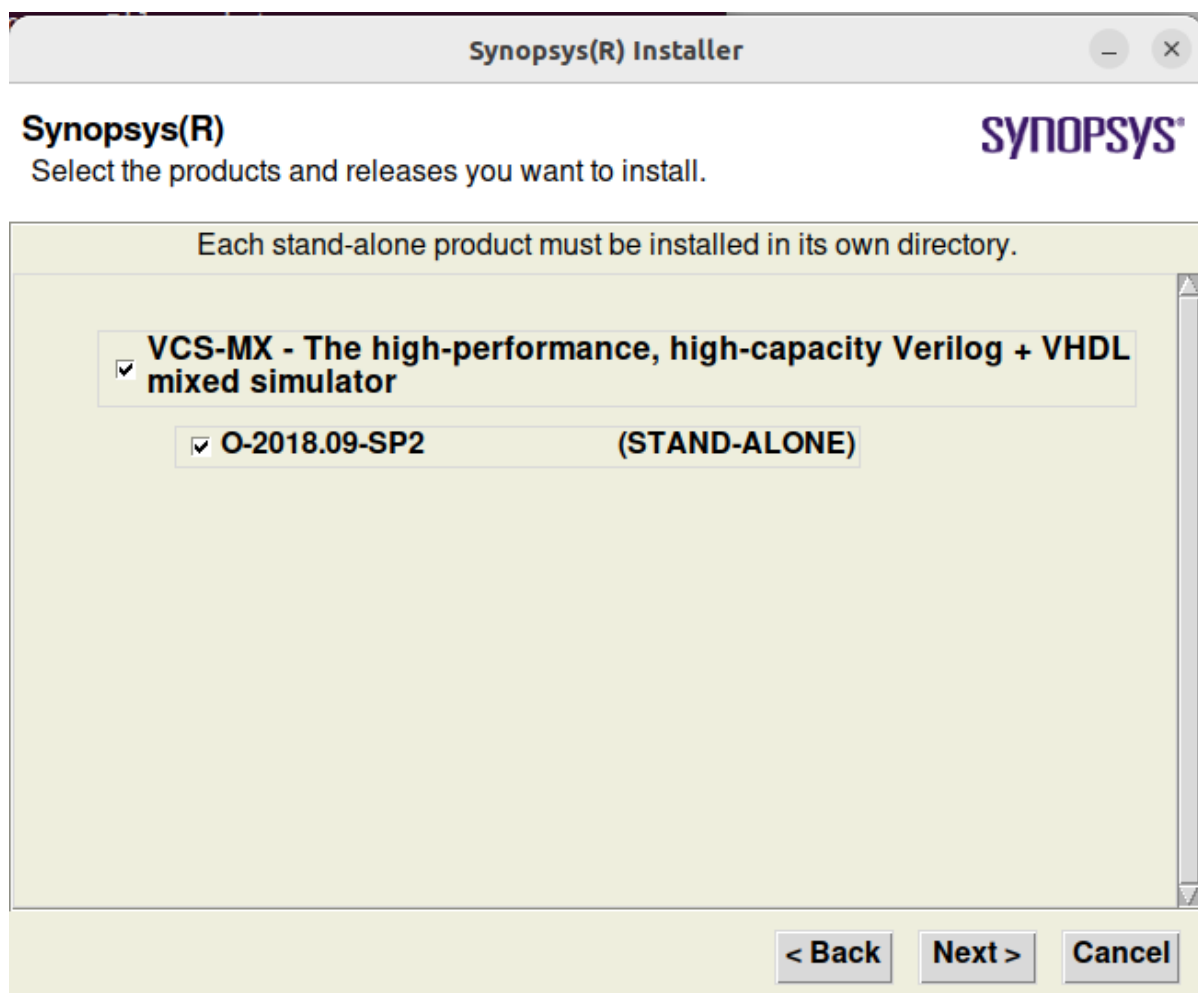
选择安装哪个软件（选vcs\_mx那个文件夹）



选择安装到哪里（刚创建的synopsys目录）



这里全选就行



这里安装完vcs似乎会显示failed，我没管，后面没影响

安装完 vcs 后，随时可以测试 vcs 是否可用，可以按这个博客来（只看 4. 测试 后面的就行，这个博客主要提供一个sv文件用来测试）测试vcs失败时的报错可以大概引导你下一步要做什么，当你跟着这个教程发现情况不一样时。另外，测试vcs单独仿真通过后，再去进行第三步

[VCS和Verdi安装教程，测试成功程序员IT虾米网\(itxm.cn\)](#)

如果你现在就测试VCS仿真，它会告诉你没有license，解决方法见博客（看 3 激活 部分就行，它用 verdi测试激活是否成功，我们用vcs测试就行）

[Ubuntu18.04 安装 synopsys2018 EDA工具 \(vcs、verdi\) 记录 synopsys安装包-CSDN博客](#)

激活最终就是执行 `/edaTools/synopsys/scl/amd64/bin/lmgrd -c`

`/edaTools/synopsys/scl/admin/license/Synopsys.dat` 这个命令，lmgrd在我们之前安装scl里，Synopsys.dat是前面说的破解程序生成的。

## 1. 每次开机都要激活 2. 先激活，在另起一个终端打开 vivado

使用.bashrc添加一些环境变量，就是把下面这段复制到.bashrc的末尾。具体的路径和hostname要改成自己的。

```
#####  
#####  
#  
#VCS2018+VERDI2018  
#  
#####  
#synosys tolls list  
  
export VCS_HOME=/home/harry/tools/synopsys/vcs-mx/0-2018.09-SP2  
export SCL_HOME=/home/harry/tools/synopsys/scl/2018.06  
  
#dev  
export PATH=$PATH:$VCS_HOME/gui/dve/bin  
alias dve="dve"  
  
#vcs  
export PATH=$PATH:$VCS_HOME/bin  
alias vcs="vcs"  
  
#scl  
export PATH=$PATH:$SCL_HOME/linux64/bin  
export VCS_ARCH_OVERRIDE=linux  
  
#LICENCE  
export LM_LICENSE_FILE=27000@harry-virtual-machine  
alias lmg_vcs='lmgrd -c  
/home/harry/tools/synopsys/scl/2018.06/admin/license/Synopsys.dat'
```

## 激活后，测试VCS时可能遇到的问题

问题1：vcs后面加个 `-full64` 选项就可以解决

```
harry@harry-virtual-machine:~$ vcs test.v
Error-[VCS_COM_UNE] Cannot find VCS compiler
VCS compiler not found. Environment variable VCS_HOME
(/home/harry/tools/synopsys/vcs-mx/0-2018.09-SP2/linux) is selecting a
directory in which there isn't a compiler
'/home/harry/tools/synopsys/vcs-mx/0-2018.09-SP2/linux/bin/vcs1' for a
machine of this type 'linux'.
Please check whether 'VCS_HOME' is incorrect; if not, see below.

Perhaps vcs hasn't been installed for machine of type "linux".
Or the installation has been damaged.
To verify whether vcs0-2018.09 supports machine of type "Linux 6.5.0-14-generic",
please look at ReleaseNotes for more details.
We determine the machine type from uname; maybe uname is incorrect.
You can fix installation problems by reinstalling from CDROM
or downloading it from the Synopsys ftp server.
For assistance, please contact vcs technical support
at vcs_support@synopsys.com or call 1-800-VERILOG
```

问题2：下面这个经典老图里体现了两个问题

```
harry@XTW-20230205FCE: ~$ cat /home/harry/tools/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so
ORIGIN'/simv.daidir//scsim.db.dir -rdynamic -Wl,-rpath=/home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib -L/home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib obj/amcQw_d.o _74184_archive_1.so SIM_l.o rmapats_mop.o rmapats.o rmar.o rmar_nd.o rmar_llvm_0_1.o rmar_llvm_0_0.o -lzerosoft_rt_stubs -lvirsim -lerrorinf -lsnpsmalloc -lvfs -lvcsnew -lsimprofile -luclinate /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/vcs_tls.o -Wl,-whole-archive -lvcsucli -Wl,-no-whole-archive /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/vcs_save_restore_new.o -ldl -lc -lm -lpthread -ldl
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/vcs_save_restore_new.o: in function `SNPSee_907e539fb4bc8f7101eac356af4ce4ed':
../OBJ/lib/saverestoreobj/obj-linux64/kernel.o:(.text+0xc41): undefined reference to `pthread_yield'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs_set_dir_map'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `snpsReallocFunc'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `snps_mem_get_info'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `snpsCheckStrdupFunc'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `ZsExecuteNBAs'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs_fread'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `snpsGetMemBytes'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs::getDirMap()'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs::fclose(vfs::CFile*)'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs_stat'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsucli.so: undefined reference to `isUcliEngineStarted'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs::vfsGetNextVirtualFile(VFILE_LIST*, bool)'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `ZsAddBorderInstance'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `snpsMemGroupSetThreadPrivate'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsucli.so: undefined reference to `setEnvPair'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `snpsMemGroupEnableLocks'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsucli.so: undefined reference to `waitForUserInput'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsnew.so: undefined reference to `vfs_fseek'
/usr/bin/ld: /home/synopsys/vcs-mx/0-2018.09-SP2/linux64/lib/libvcsucli.so: undefined reference to `initNativeUcli'
```

首先是 `../OBJ/lib/saverestoreobj/obj-linux64/kernel.o:(.text+0xc41): undefined reference to pthread_yield` 这一行，解决见：

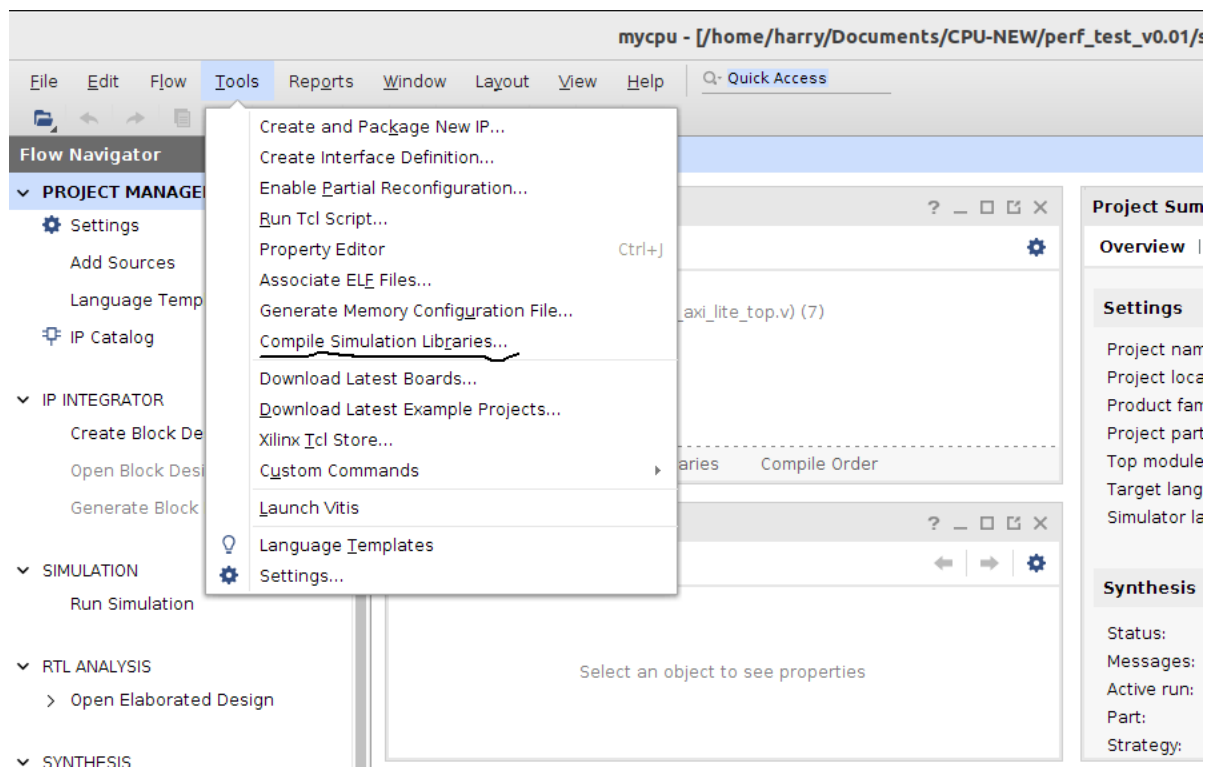
[OBJ/lib/saverestoreobj/obj-linux64/kernel.o:\(.text+0xc41\): undefined reference to `pthread\\_yield'-CSDN博客](#)

其次是下面的那些 `undefined reference to`，这里应该是gcc版本的问题，我的解决方法是在 `vcs` 命令后再加入 `-LDFLAGS -Wl,--no-as-needed`。也就是按照这个博客。（但也有说法是g++版本的问题，不懂）

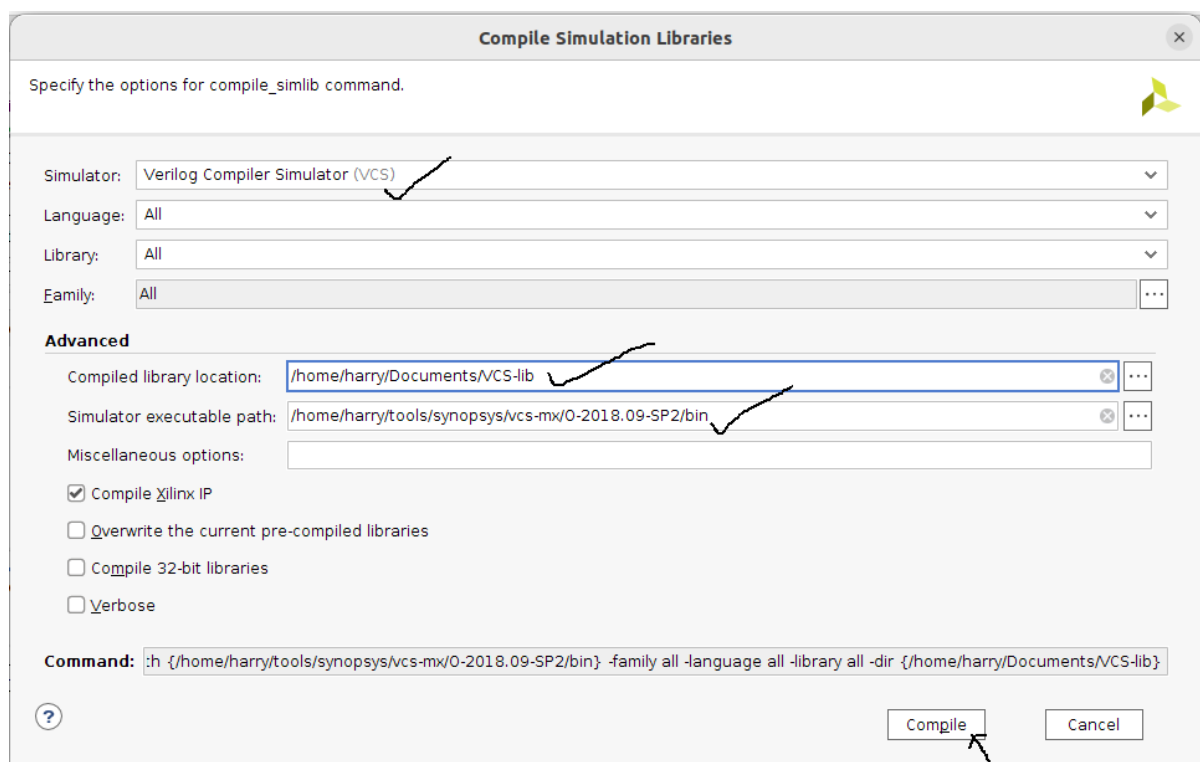
[libvcsnew.so: undefined reference to /opt/synopsys/vcs-mx/o-2018.09-sp2/linux64/lib/lib-CSDN博客](#)

### 三、使用VCS编译vivado ip库

测试好vcs可以仿真后，在vivado中编译仿真库



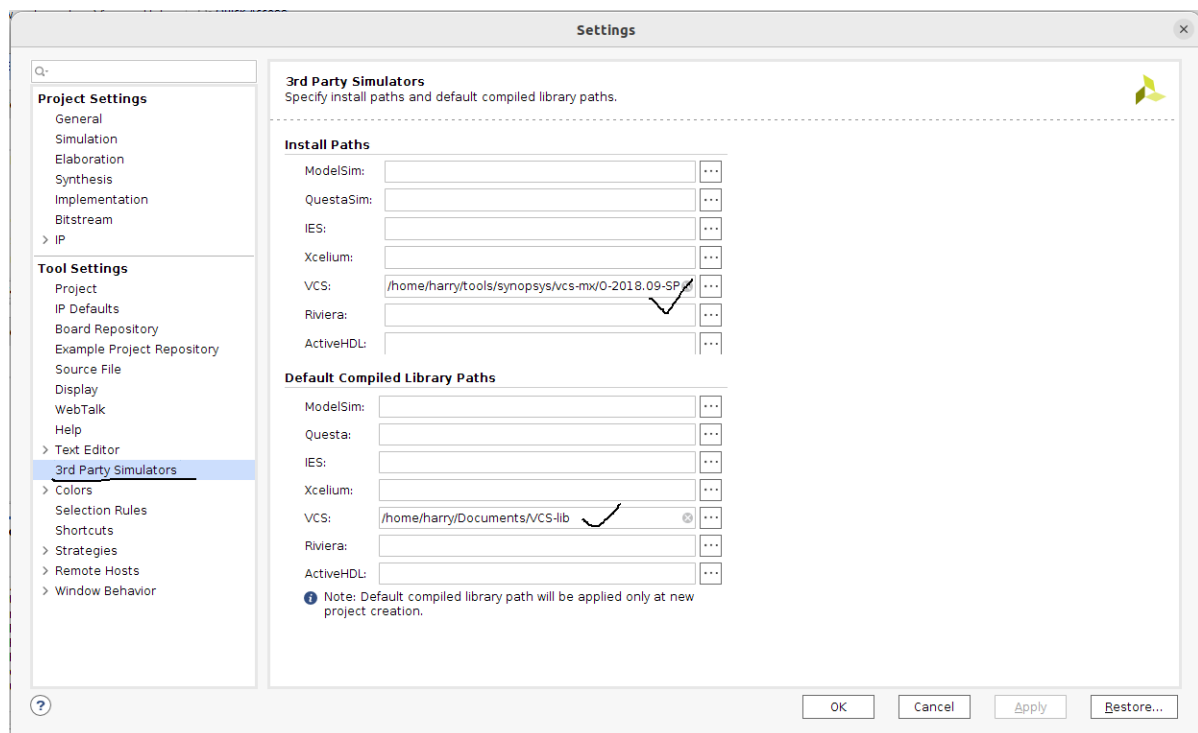
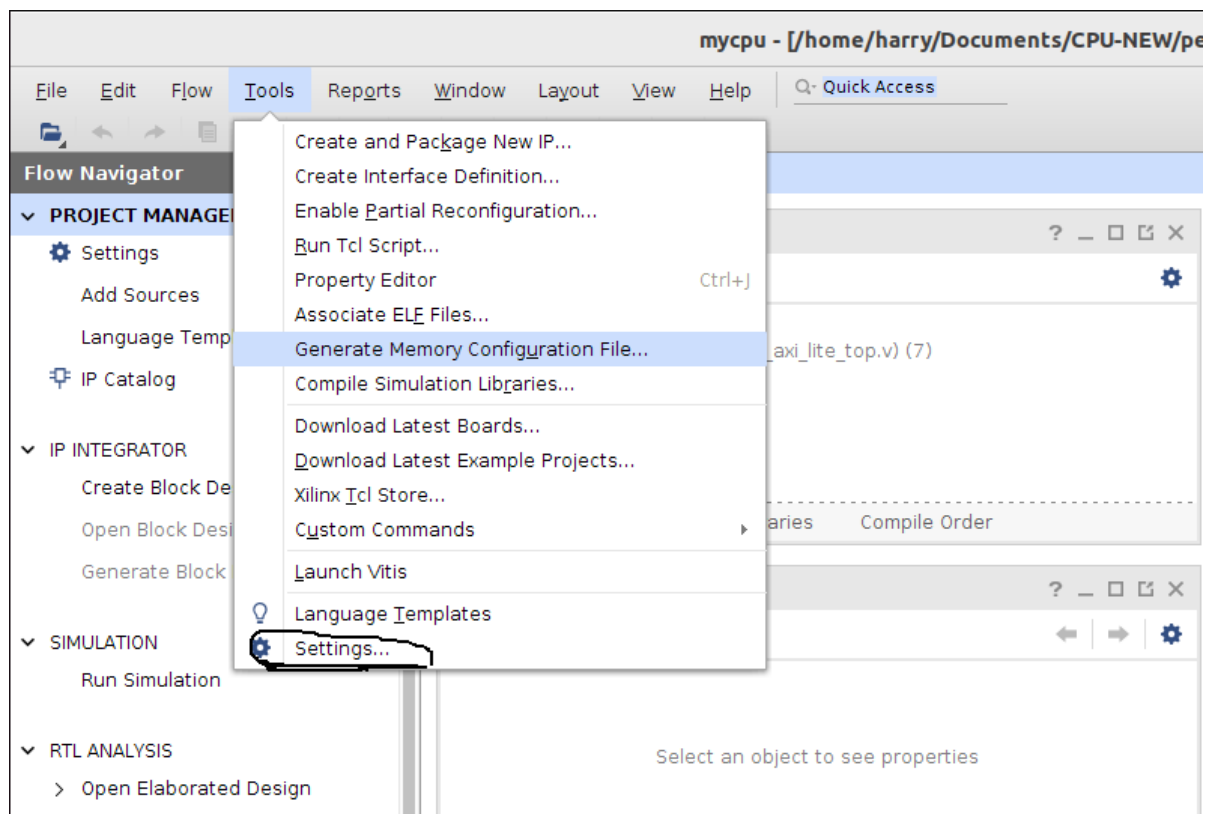
中间的Compiled library location要自己创建一个目录



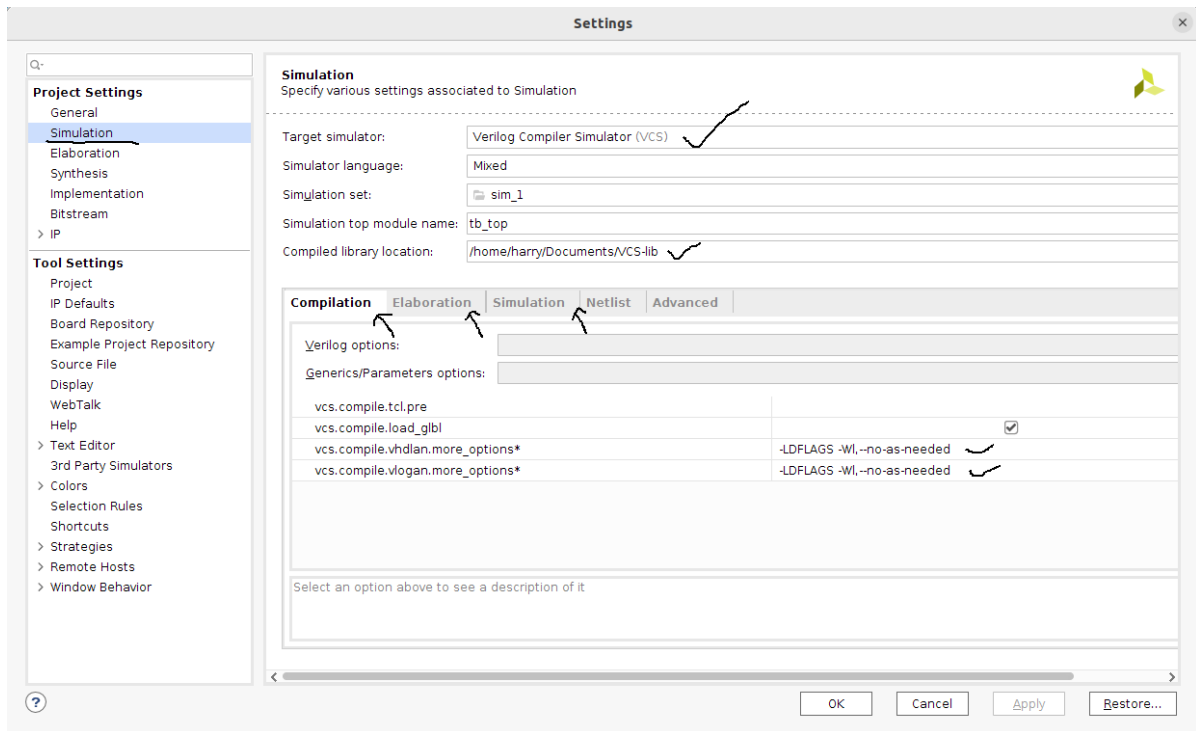
这个大概5mins左右

## 四、设置 vivado + vcs 联合仿真

按照图中的做就行

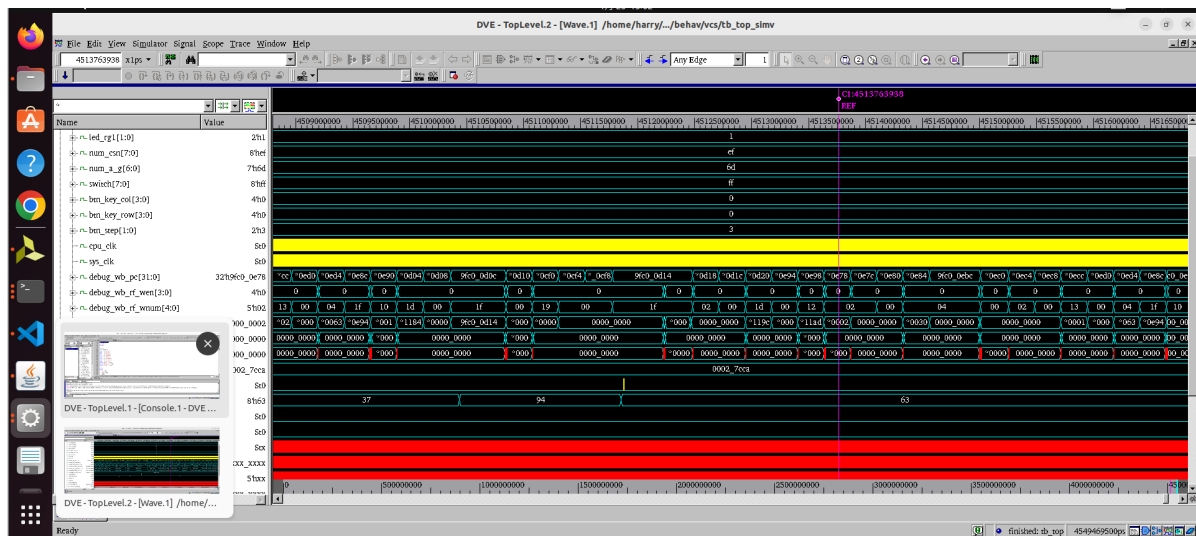


这里下面的两个more\_options\*, 可能需要配置也可能不需要, 这个跟前面测试vcs出现的那一群 undefined reference是一个问题。我这里把三个箭头指向的步骤中的所有more\_options\*都设置成了 `-LDFLAGS -Wl,--no-as-needed`。



## 五、使用vcs仿真

经过上面的配置后，采用同样的方式进行仿真，vivado就会调用vcs。



最后：

vcs仿真对verilog语法的要求更加严格，有助于减少上板错误，但一开始仿真需要照着报错逐个修改。

```
file copy -force ../../../../soft/perf_func/obj/stream_copy/axi_ram.mif
/mycpu.sim/sim_1/behav/xsim/axi_ram.mif
```

在进行功能&性能测试时，需要把上面命令中的xsim改成vcs。