1.下载群文件中的nscscc2021\_group\_v0.01.7z到本地并解压，密码：nscscc2021

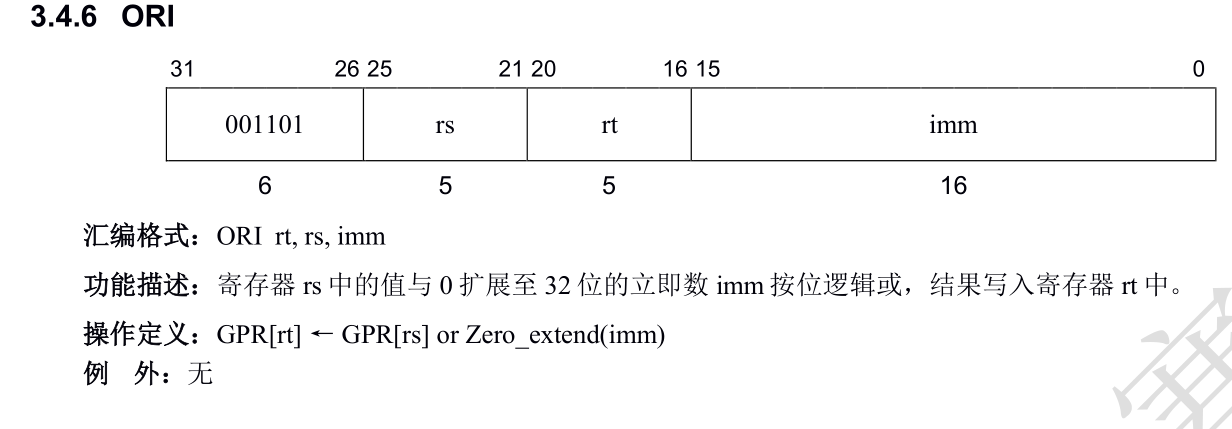
2.clone本项目到本地

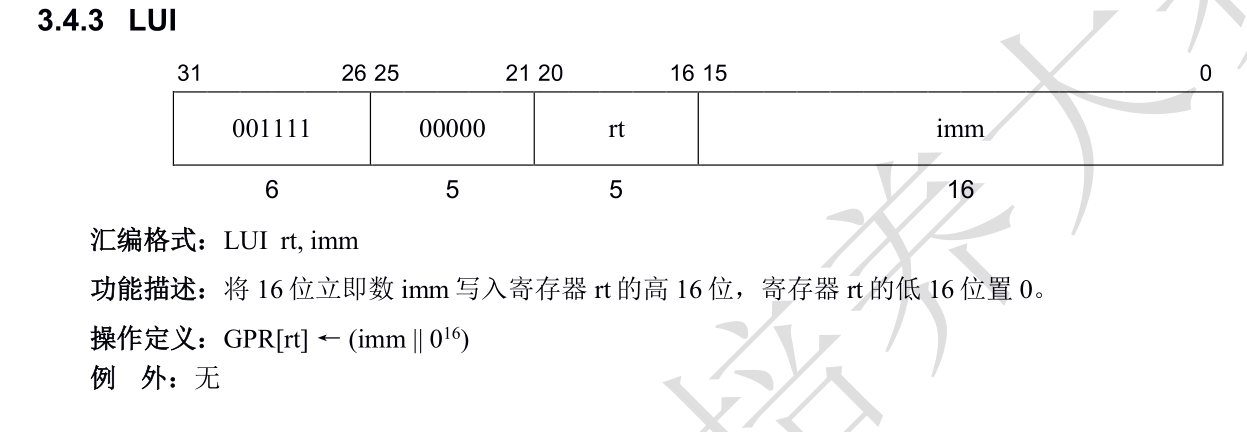
3.启动路径 nscscc2021\_group\_v0.01\func\_test\_v0.01\soc\_sram\_func\run\_vivado\mycpu\_prj1 下的 mycpu.xpr项目

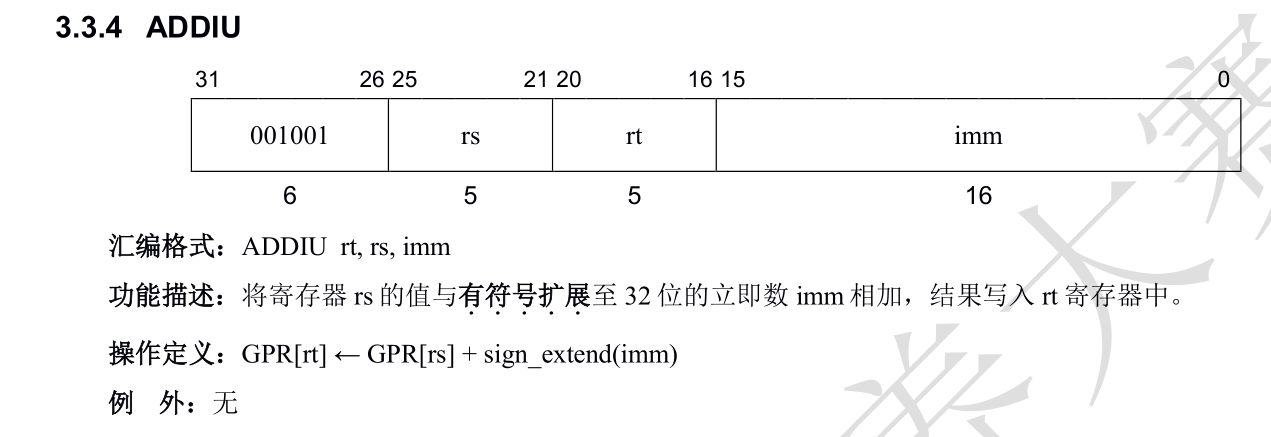
4.在vivado中添加源文件，把这个项目的.v文件都导入（包括lib文件夹下）。

5.点击simulation进行仿真，第一次仿真时会对项目使用的ip核进行综合，可能需要等待10-25分钟不等（视电脑性能而定）。

6.点击播放进行仿真，如果停止可在vivado下方的控制台看到提示信息，这是龙芯实验平台提供的比对机制会告诉你当前在哪条 指令出现错误。







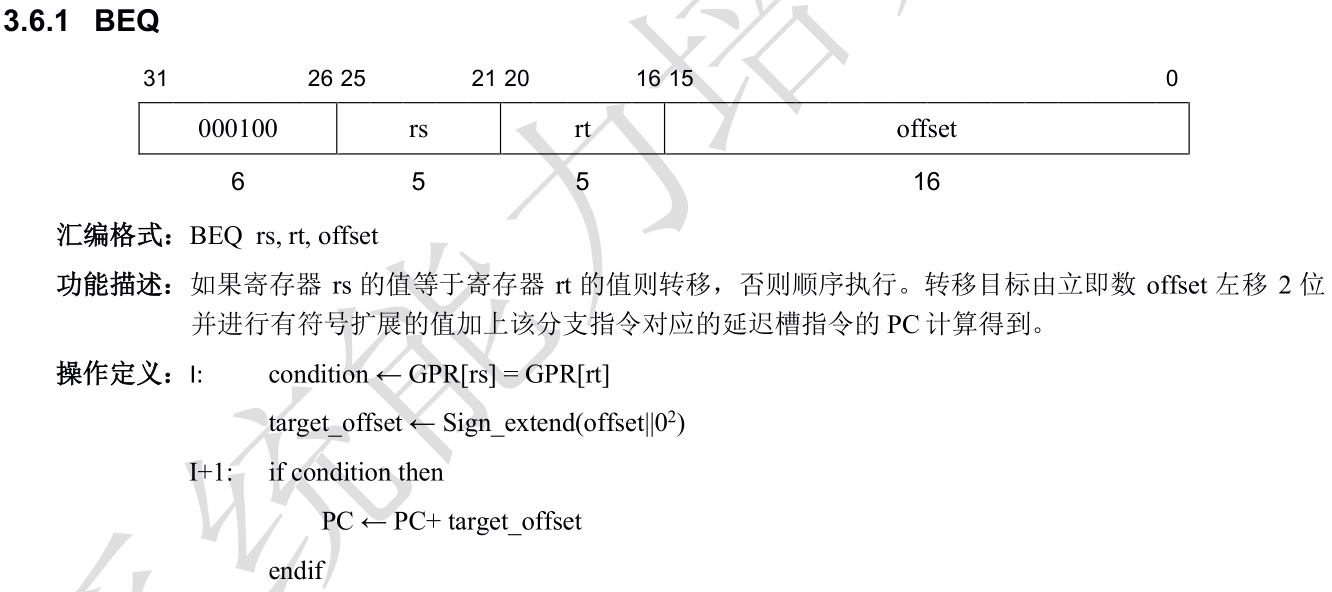


图1 初始程序能实现的4条指令格式

查看console（控制台）中的trace比对机制的提示，记下提示的PC值，并猜测可能发生的错误。

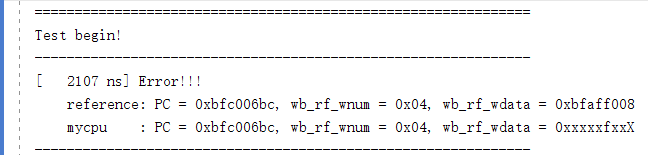
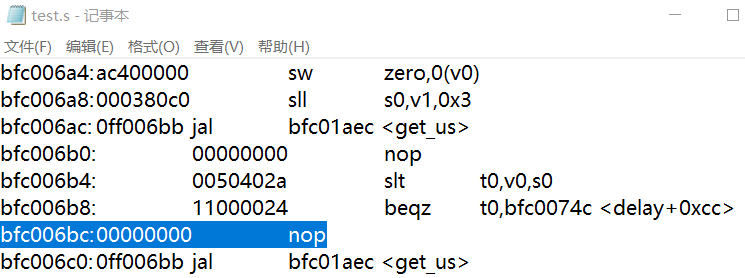


图2 初始运行错误截图

7.打开test.s文件，使用提示的PC值进行查找，查看该PC值对应的指令是什么，判断是否添加该指令。



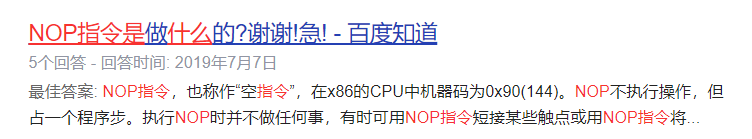
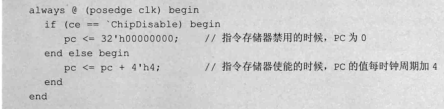
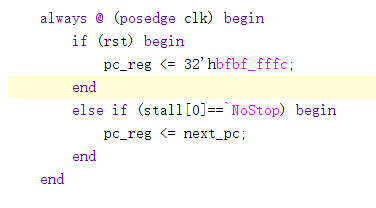


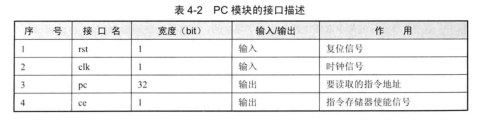
图3 寻找报错的指令

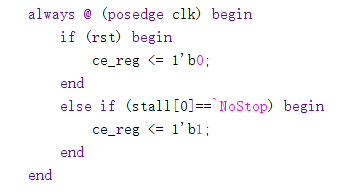
IF.v:

PC赋值分类讨论：

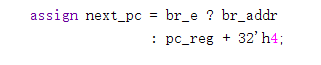


指令寄存器使能信号赋值分类讨论：

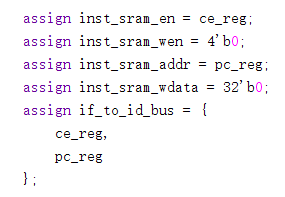




讨论是分支还是顺序执行：

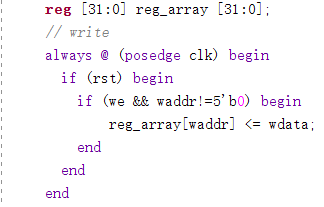


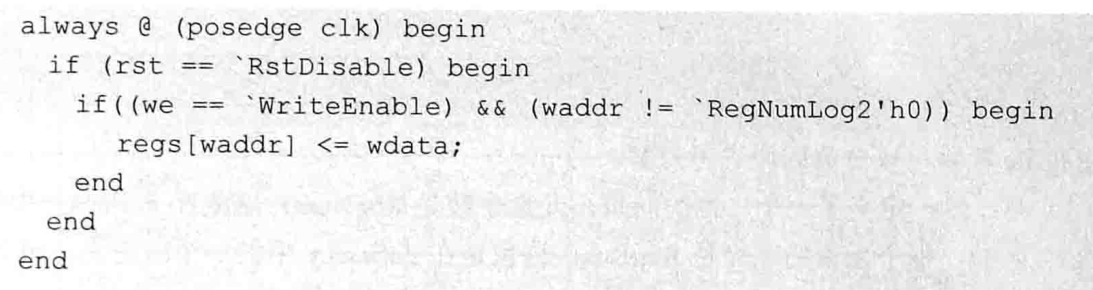
IF/ID:



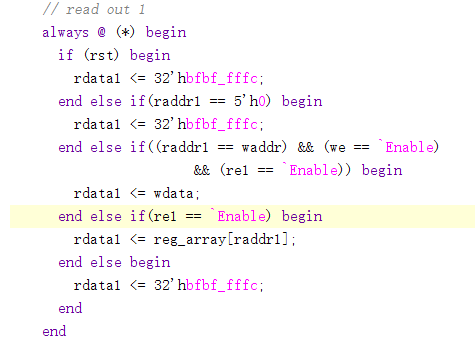
regfile.v:

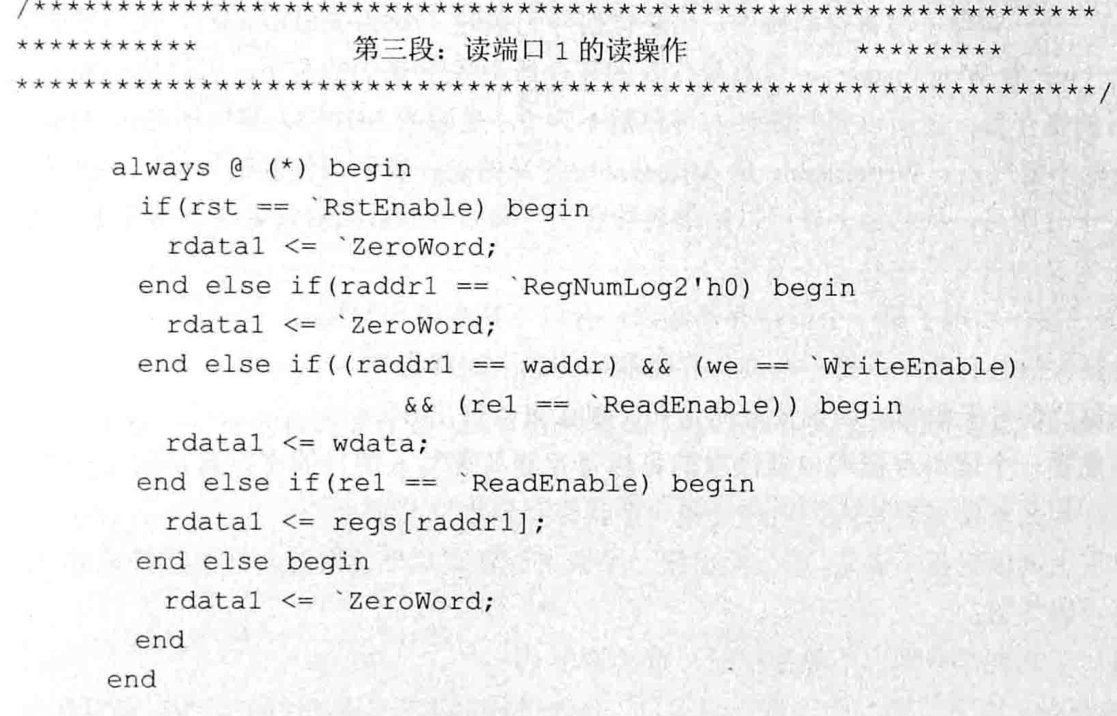
第二段：写操作：

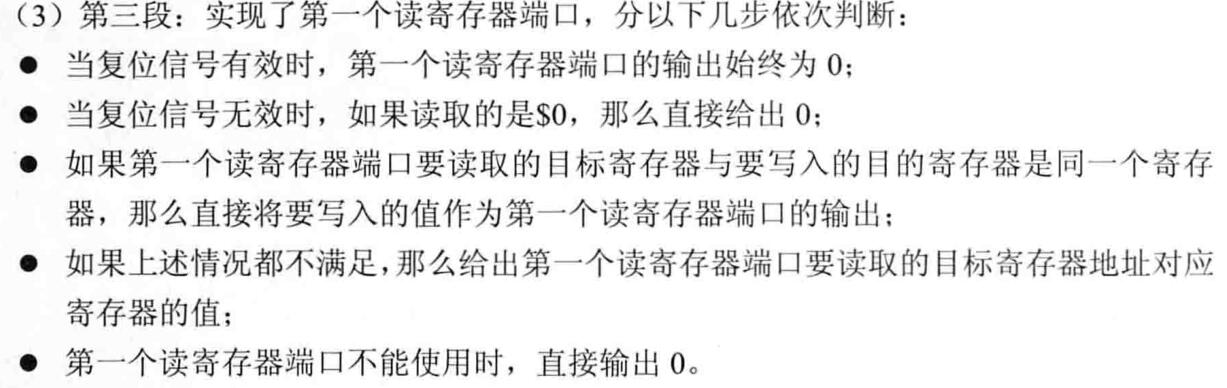




第三段：

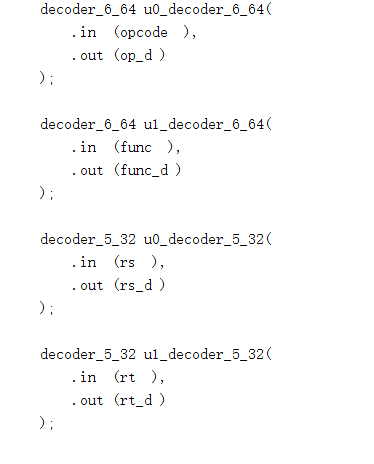






ID.v:

ID段译码和读source/target register:



  
这里应该是选择信号用于确定指令是RR型，RI型还是offset型