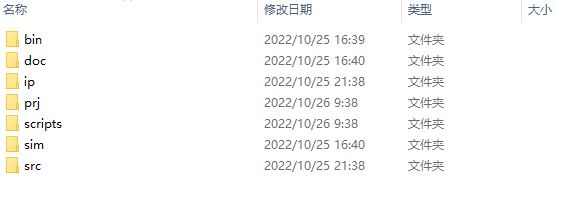
FPGA 平台管理和git使用说明

为了方便管理维护和共同开发FPGA,对整个FPGA 工程结构以及使用做了简单的文档描述。

1. FPGA工程目录

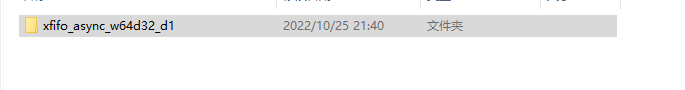


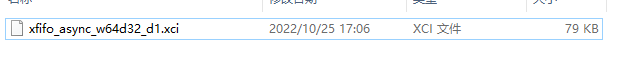
bin: 生成烧录的bin文件(bin文件有fsbl.elf fpga.bit app.elf 通过命令打包生成后面可以补充)。生成的版本文件可以做归档。

Doc: 项目开发相关文档（寄存器手册,逻辑架构等）

Ip:用来存放生成的ip



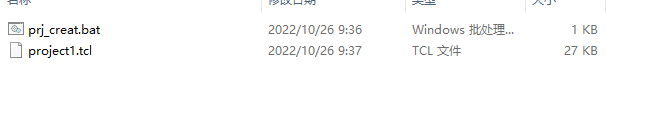




相关的ip存放相应的目录下，BD文件比较特殊，这个会特殊处理。上传或者更新的时候只需要更新.xci文件其余的不用上传。BD文件上传HDL文件夹和相应的\*.bd文件。

Prj: 文件夹由脚本生成的工程，不用上传。

Scripts ：存放脚本的文件夹，后面有ip的改动或者工程文件的增加或删除要及时更新相关脚本si



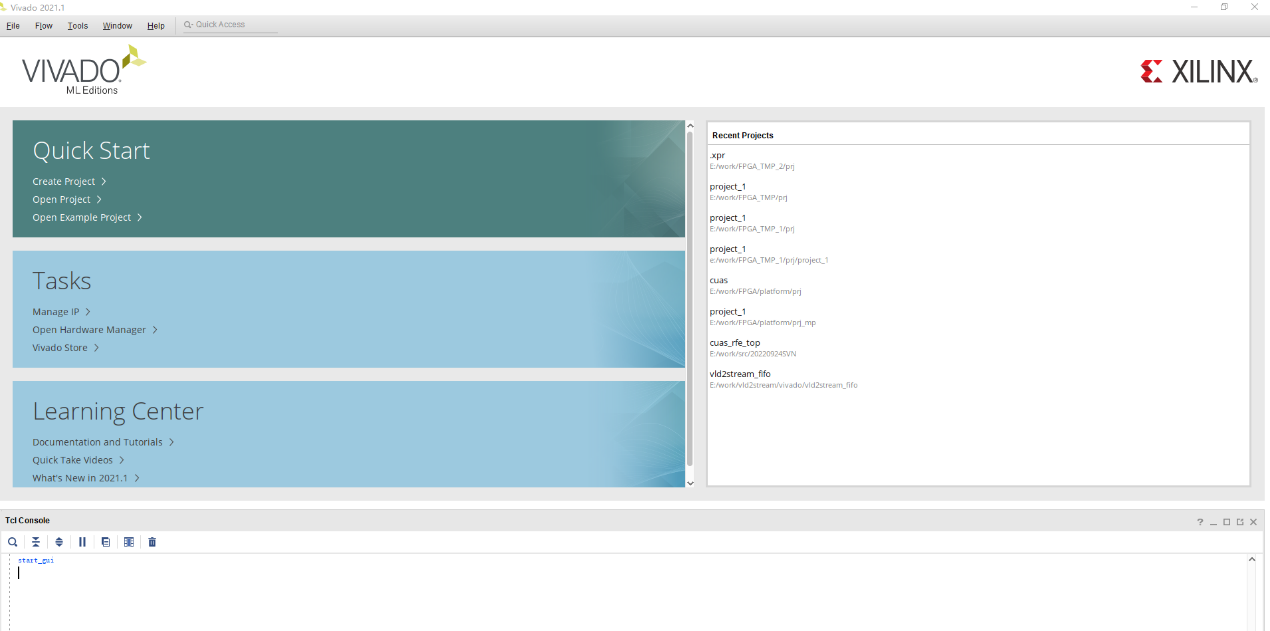
Sim : 存放相关仿真的文件

Src:

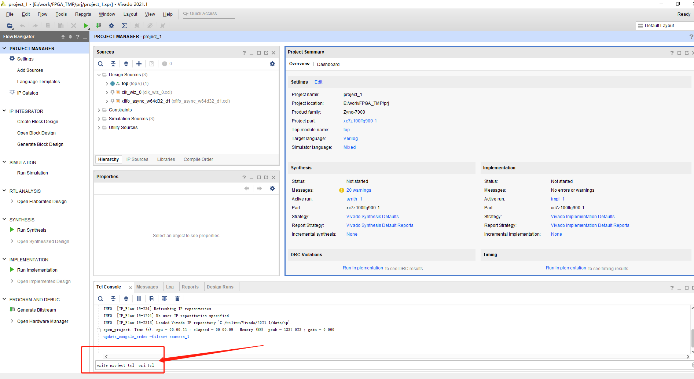


存放verilog代码的路径。

1. 现有工程代码平台整理说明：
2. 按照FPGA工程目录对代码进行整理,ip和verilog放到相应的文件夹下。
3. 创建工程

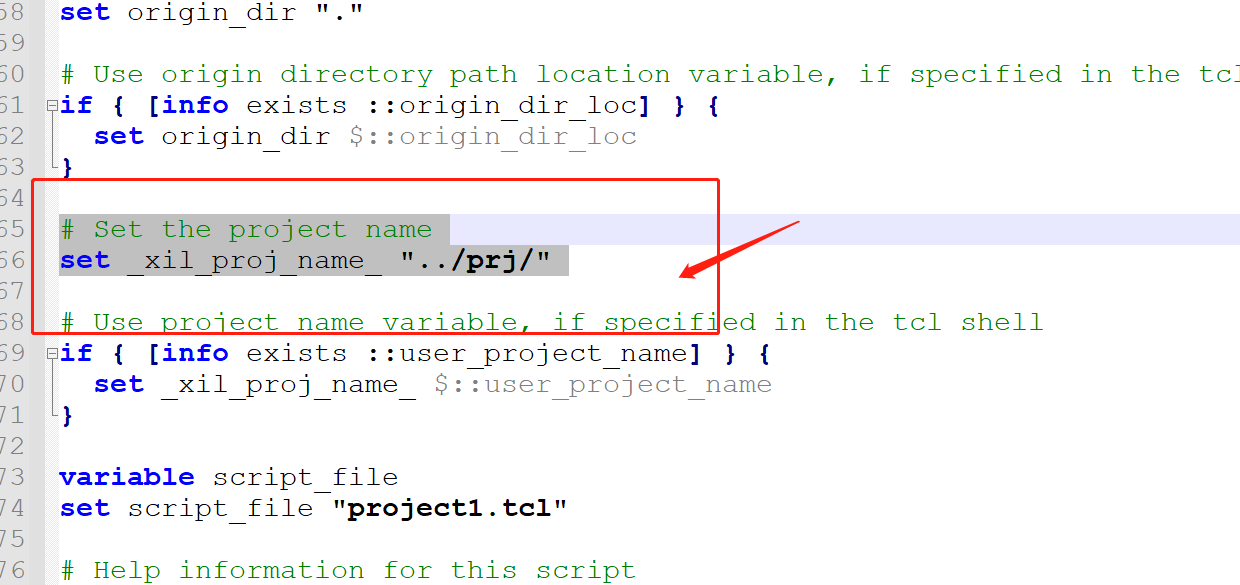


1. 添加相应的代码（src,ip，约束等）
2. 跑个工程，确保工程的正确
3. 以上都没有问题，就可以生成脚本

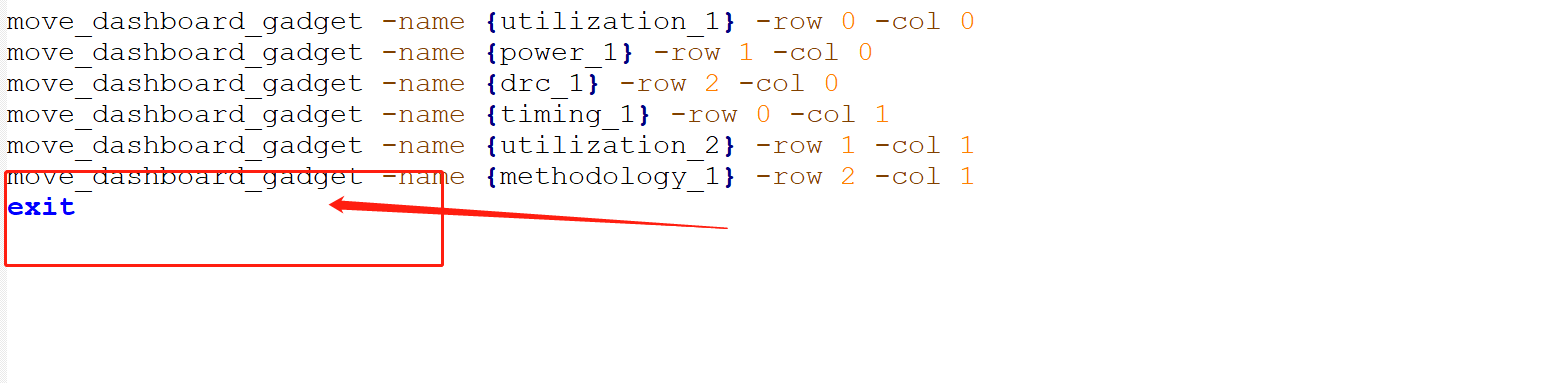


在Tcl Console 控制台下使用命令write\_project\_tcl prj.tcl生成tcl脚本。

1. 对tcl文件进行修改

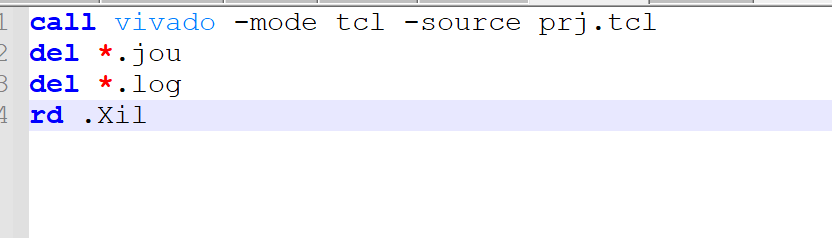


这里的修改会在FPGA工程目录下生成prj文件。



文件末尾加上exit 配合.bat脚本生成工程。

1. 将tcl文件拷到scripts文件下上传git
2. .bat脚本



调用vivado生成工程，这里是调用的命令行的模式生成的工程。确保vivado已经添加到环境变量。后面会删除一些log日志。

1. 后面ip路径只保留.xci文件。BD文件保留.bd文件和HDL文件夹。
2. 整个工程整理完成，上传git
3. Git使用说明：（）