各位同学大家好，很高兴与大家下学期一起从事risc-v架构的cpu开发的相关工作。我们要研究的risc-v架构的cpu是由苏黎世瑞士联邦理工学院开发的zero-riscy。项目中可能会用到的相关资料和链接如下，有什么不懂的问题大家可以跟我多多沟通~

一、项目资料

1. zero-riscy项目github网址：

<https://github.com/lowRISC/ibex>

1. zero-riscy的使用说明文档：

<https://ibex-core.readthedocs.io/en/latest/index.html>

1. zero-riscy的相关论文：

<https://ieeexplore.ieee.org/document/8106976>

<https://ieeexplore.ieee.org/document/7864441>

1. vivado下载地址（推荐安装vivado2017.4）

<https://china.xilinx.com/support/download/index.html/content/xilinx/zh/downloadNav/vivado-design-tools/archive.html>

选择该文件下载，具体的安装过程可参考我所给的pdf的18、19章。Vivado lic（许可证）的生成方式参考csdn。

1. risc-v指令集介绍文档

<http://crva.ict.ac.cn/documents/RISC-V-Reader-Chinese-v2p1.pdf>（中文版）

<https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf> （英文版）

1. risc-v架构cpu设计方法

书籍《计算机组成与设计：硬件/软件接口（RISC-V版）》

二、学习流程

1. 首先阅读相关文献，了解risc-v架构是什么？有哪些指令集？risc-v架构的cpu如何设计？

2. 阅读论文，了解zero-riscy和其兄弟riscy是如何设计的？区别是什么？其结构是什么样的？

3. 在linux系统中把该cpu工程给编译出来，根据其论文中的结构图分析其代码结构，弄清楚每个模块的功能和想实现的目的。

4. 全部完成之后学习一下基于该核的SOC项目pulpino。并把它工程编译出来。了解其外设和该cpu是如何交互的，了解其运行机制。

5. 将该工程上板验证，尝试调用其串口等。

三、工程实现流程：

Zero-riscy安装和编译流程：

第一步：安装好Ubuntu18.04（虚拟机或双系统均可）、vivado2017.4。

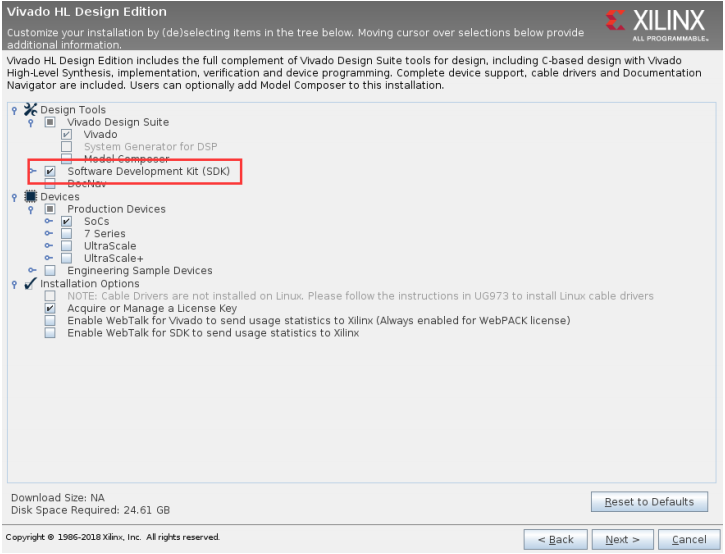
第二步：从GitHub上用git clone 指令将ibex项目下载下来，或者用ssh命令下载。请勿从windows系统中将该工程中压缩包下载然后拷贝到linux环境中，很容易缺少重要文件导致无法编译。

第三步：仔细阅读ibex中的readme，根据步骤编译该cpu工程。

第四步：将tb文件放入该工程中，检测编译结果的正确性。

温馨提醒：

1. 要给电脑留出至少150GB的空间安装双系统或者虚拟机。因为vivado工程本身很大。
2. 这一步里面所有都选上，以免后续报缺少元件库的错误。



1. 这一步切记一定要存在该路径下，否则后续编译很麻烦

