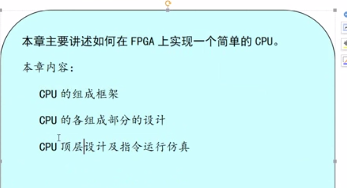
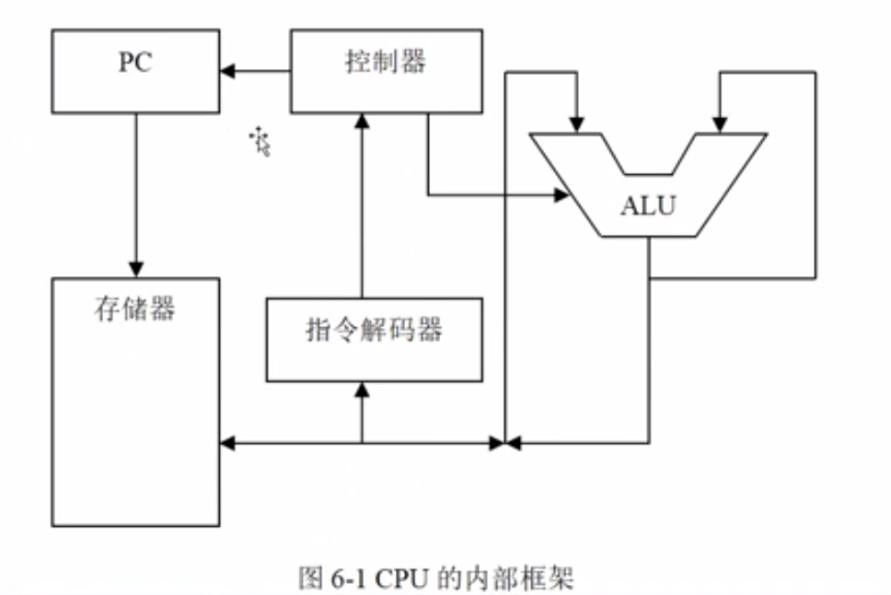
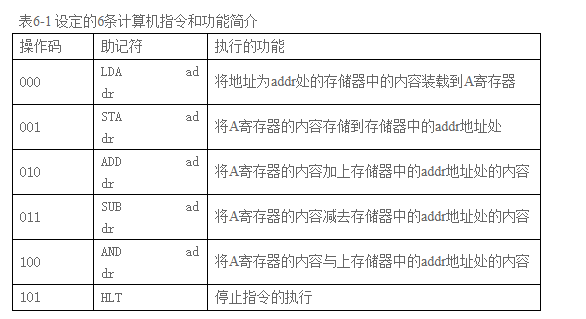
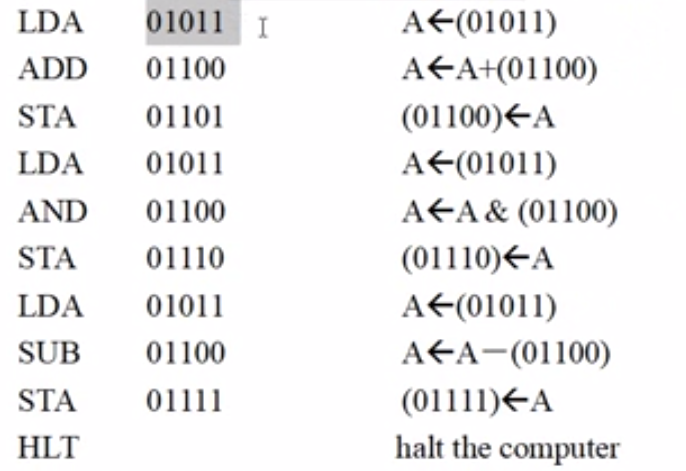
# 概述

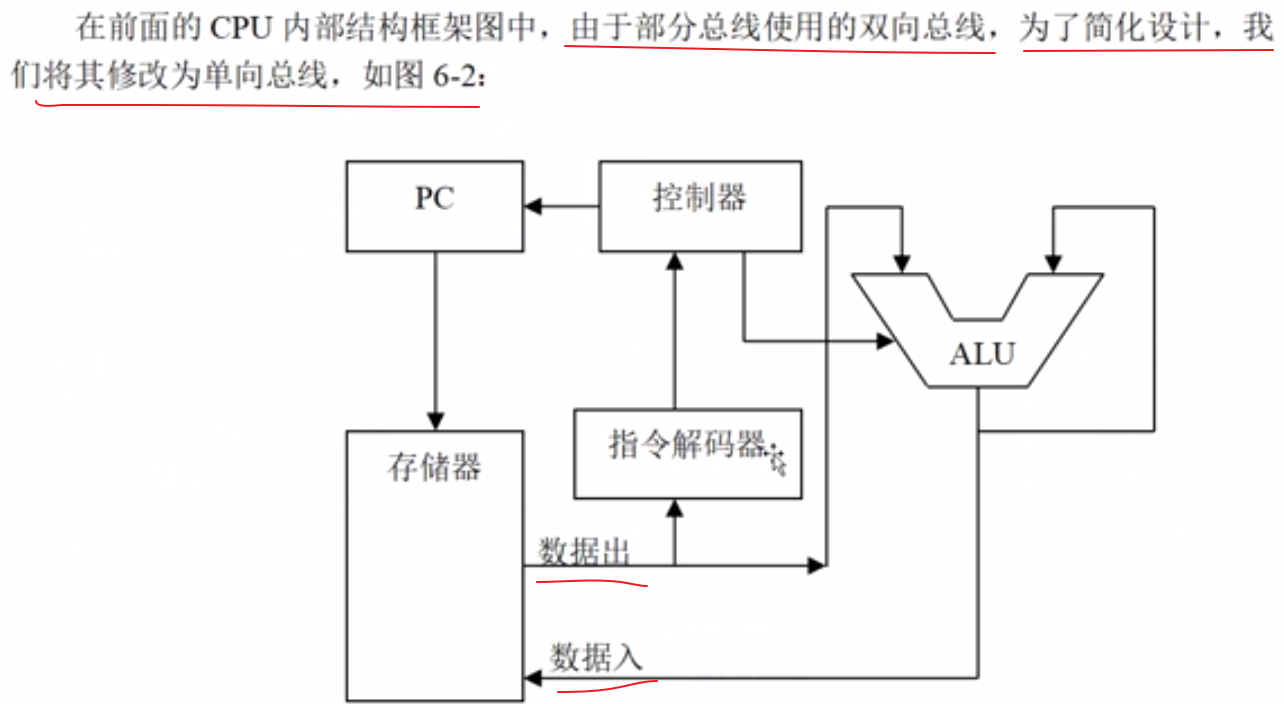


# CPU总体组成：





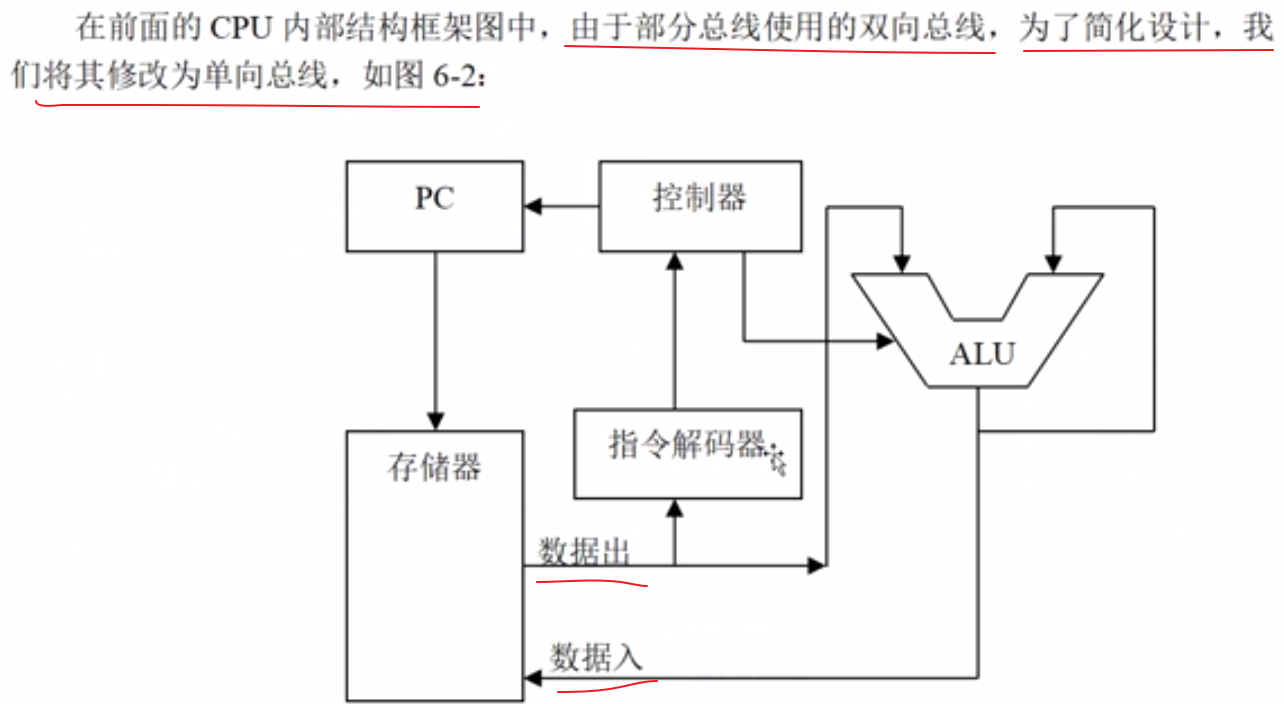


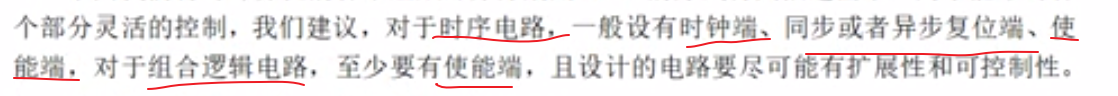


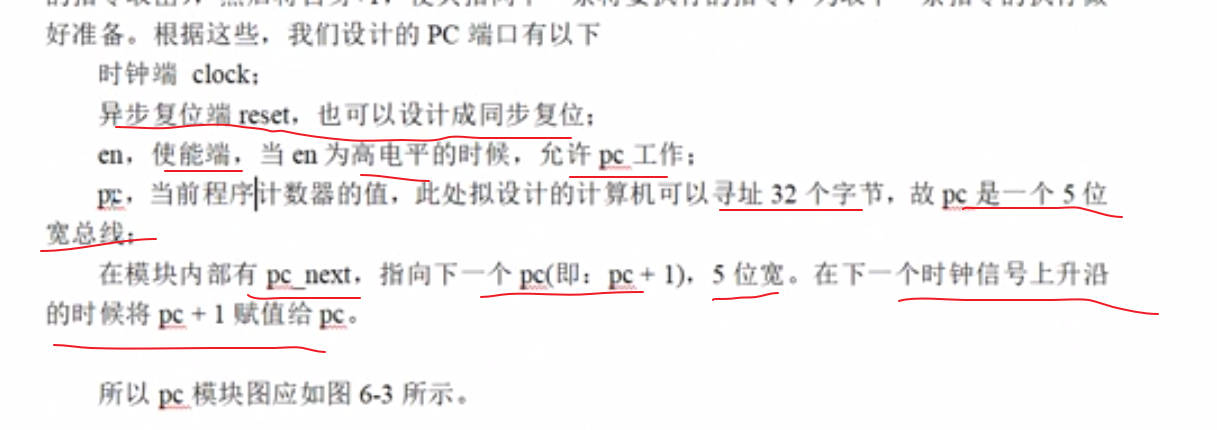
# CPU具体部件设计：

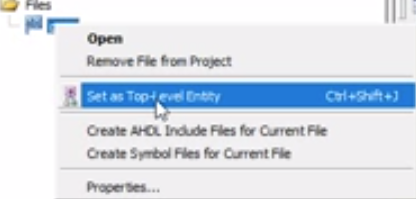
## PC:

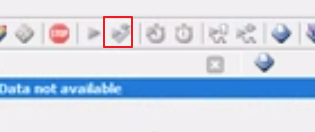
### 程序编译

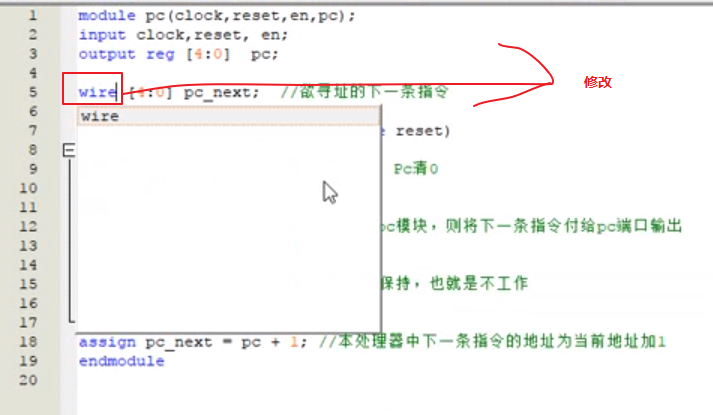




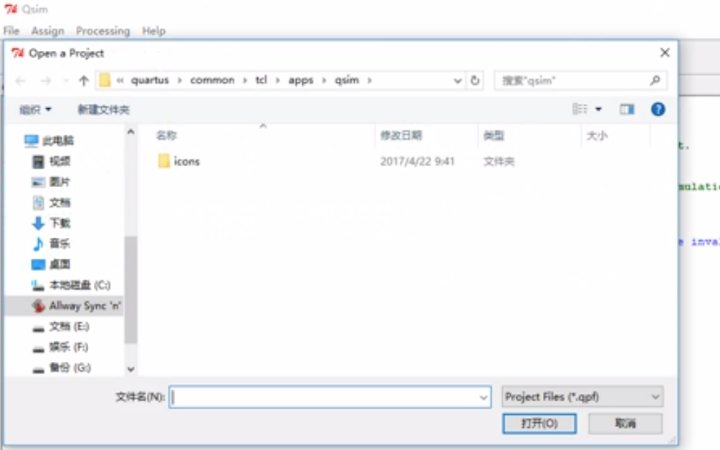


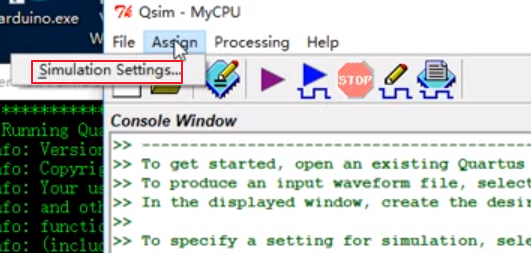


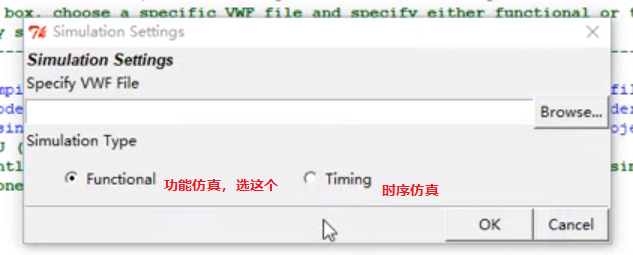


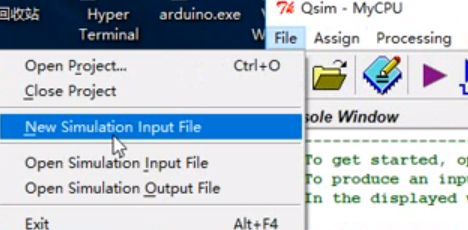


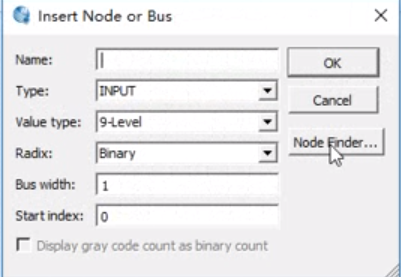
### 仿真

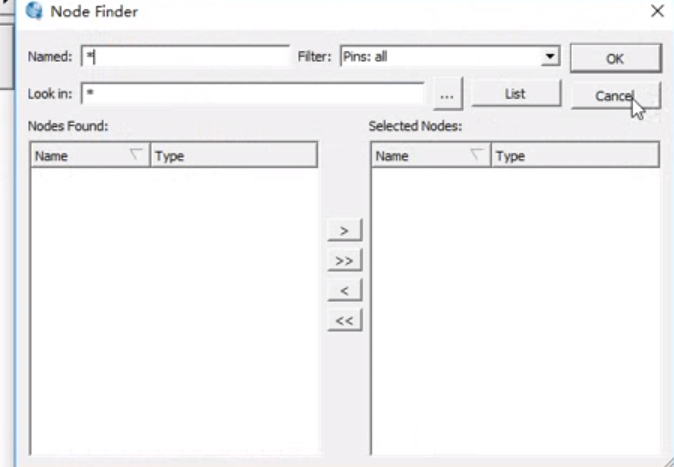


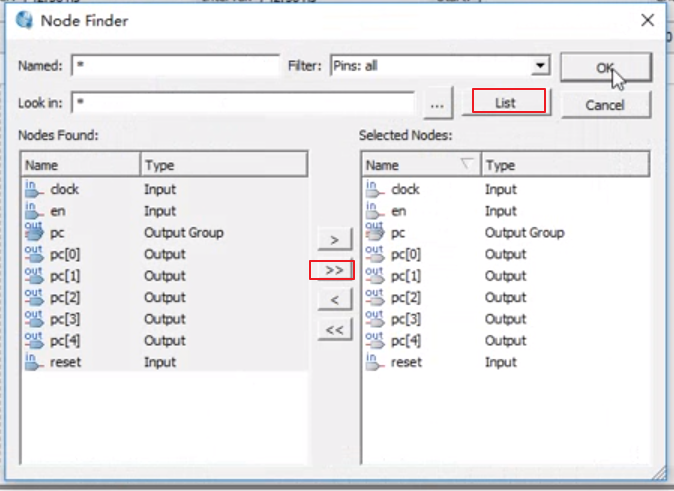


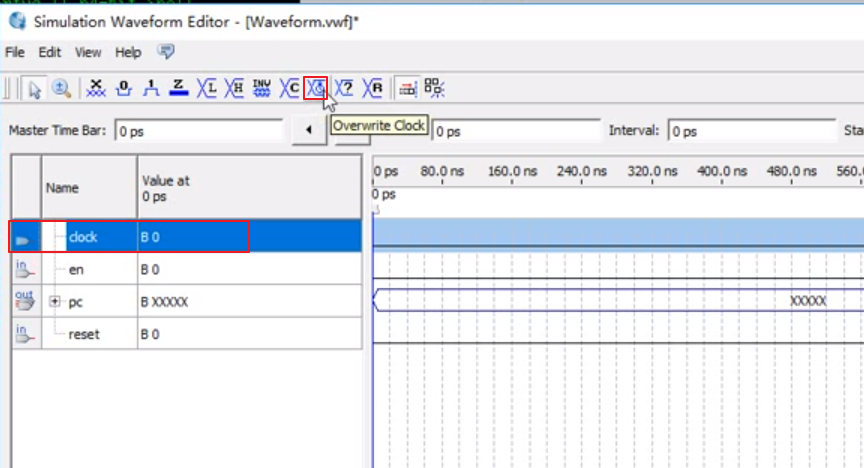


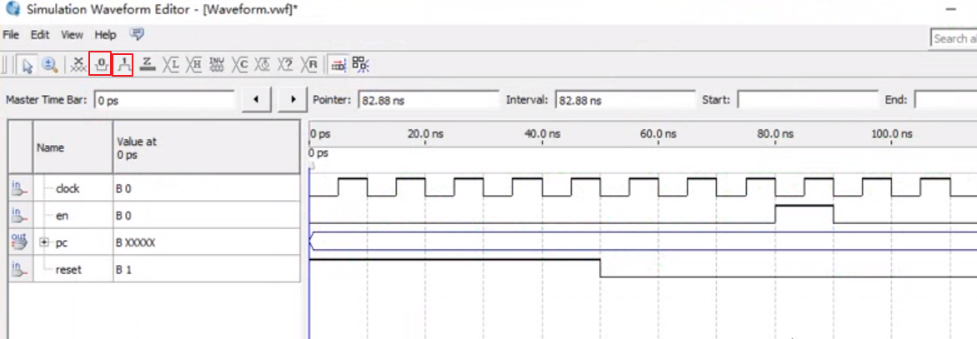


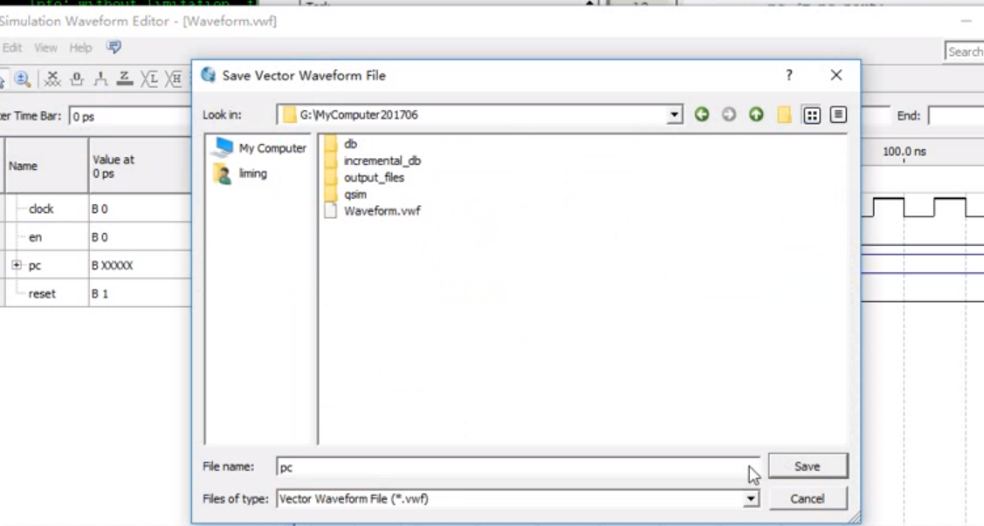


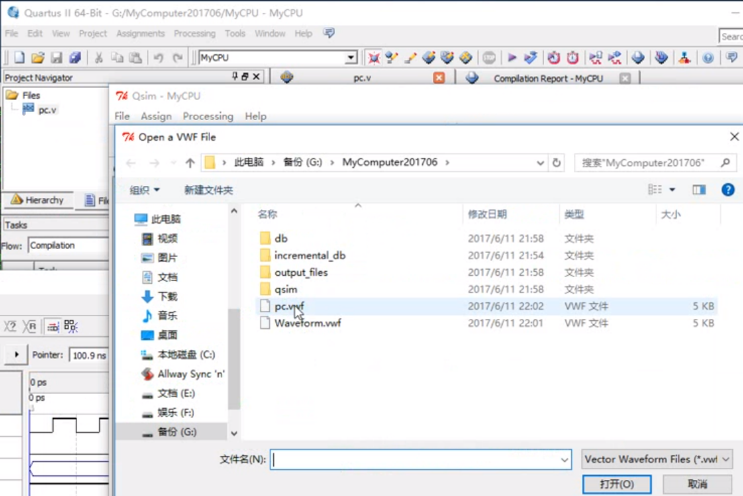


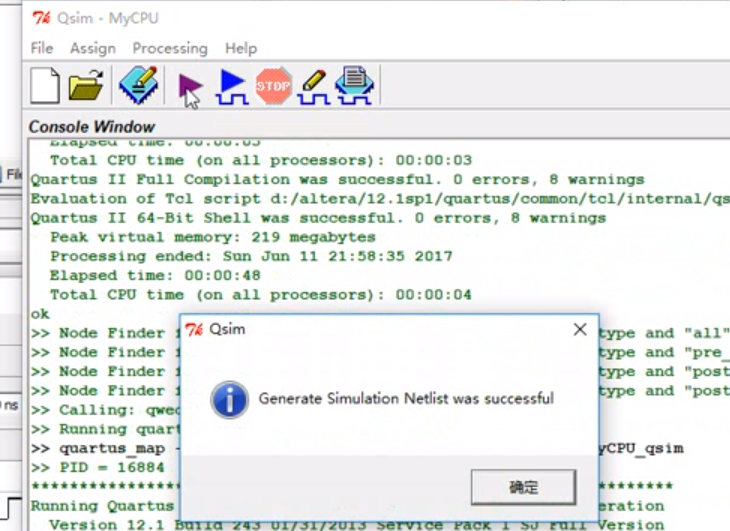


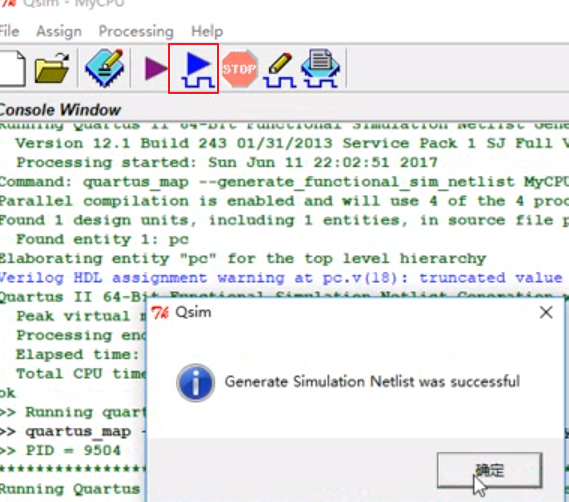




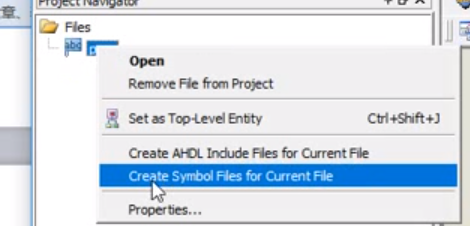


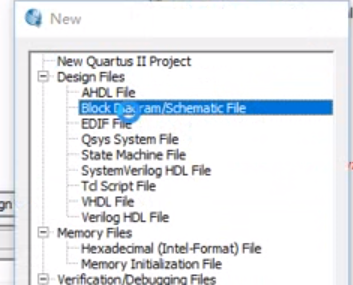


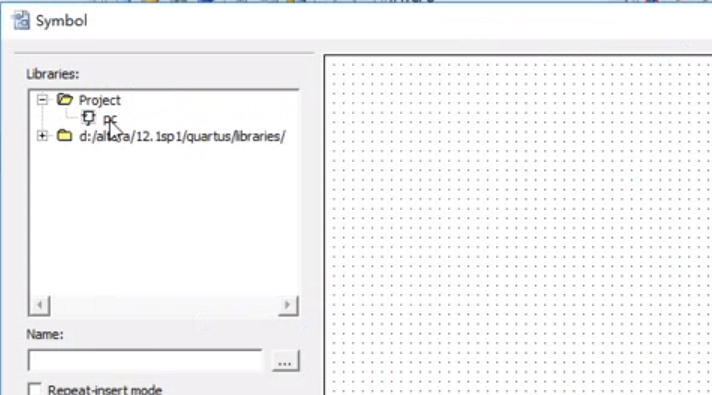




### 模块图

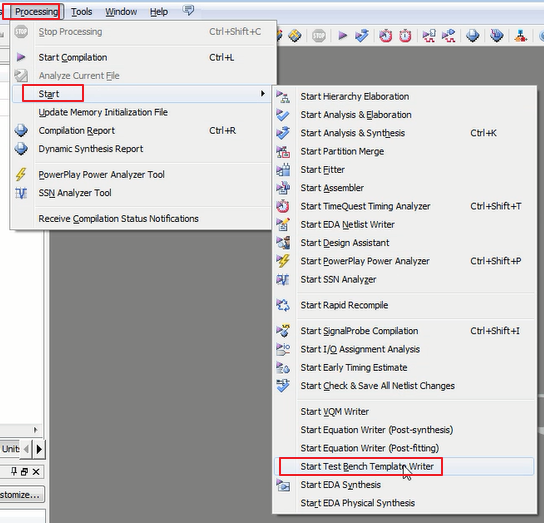






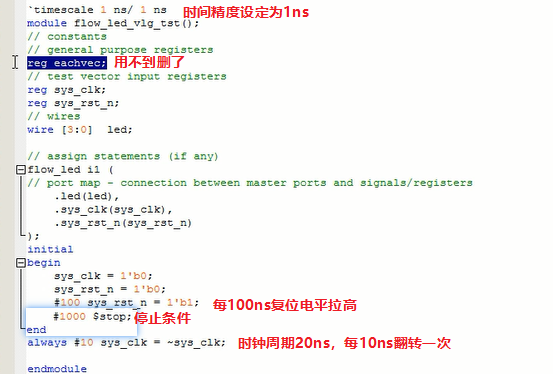
# Modelsim联合仿真

## 生成TestBench模板(.vt)



## 打开模板文件(.vt)编辑

根据console提示来就行。

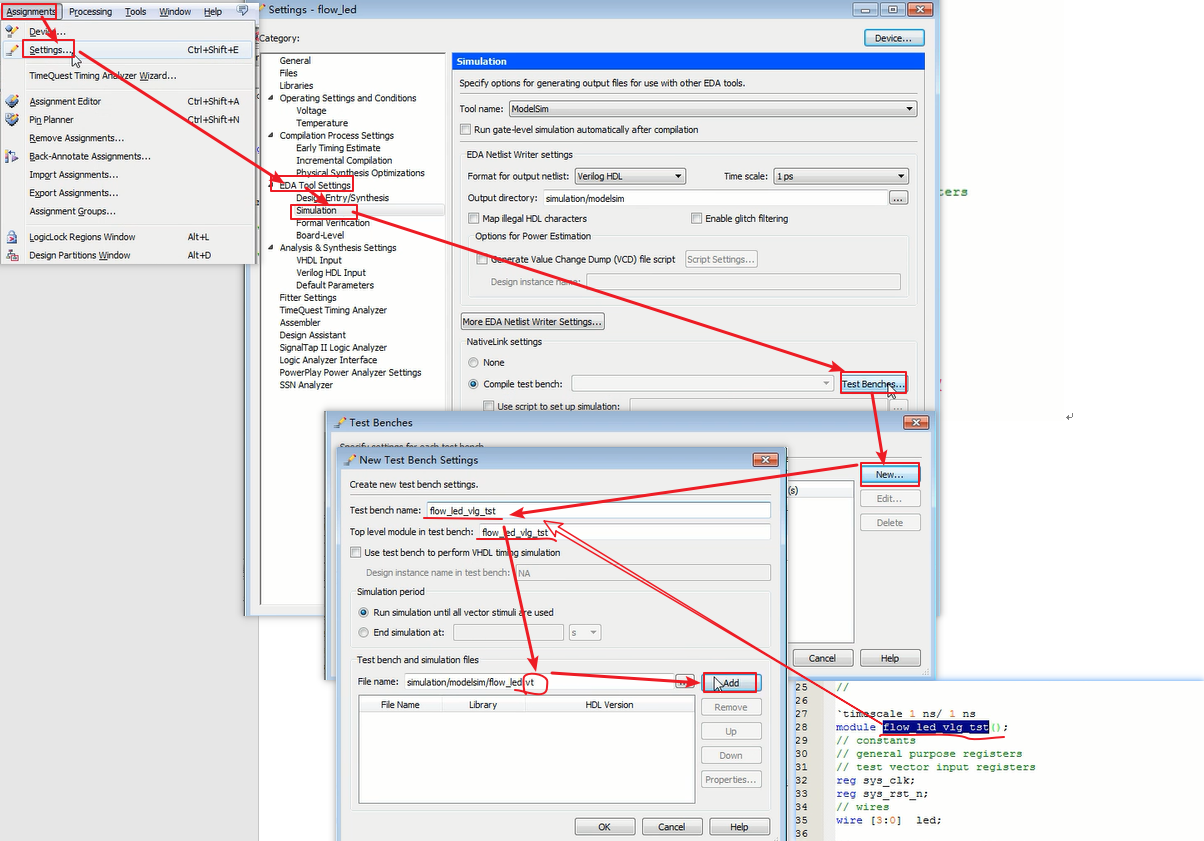


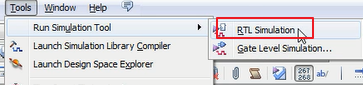
## 打开顶层代码（.v）看情况修改

……

判断时间可以忽略不计，复制要占用1个时钟周期。

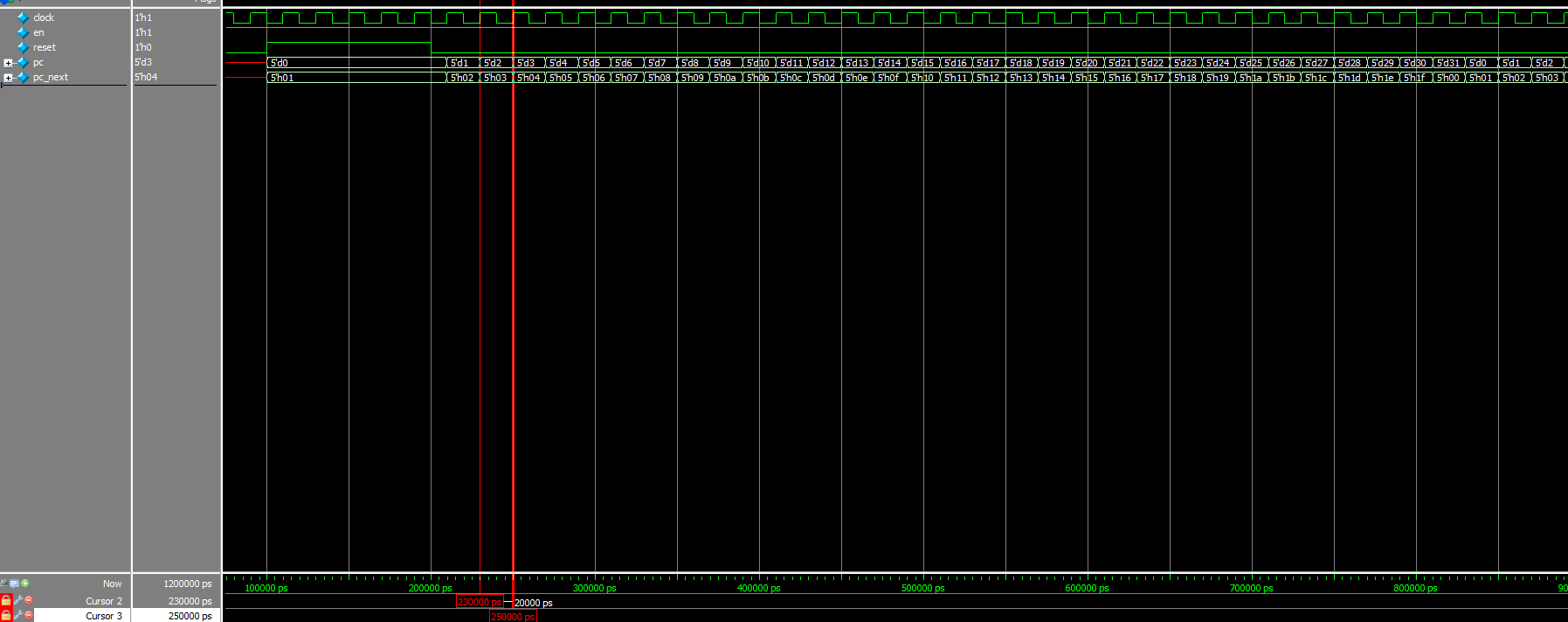
## 载入TestBench文件并启动仿真



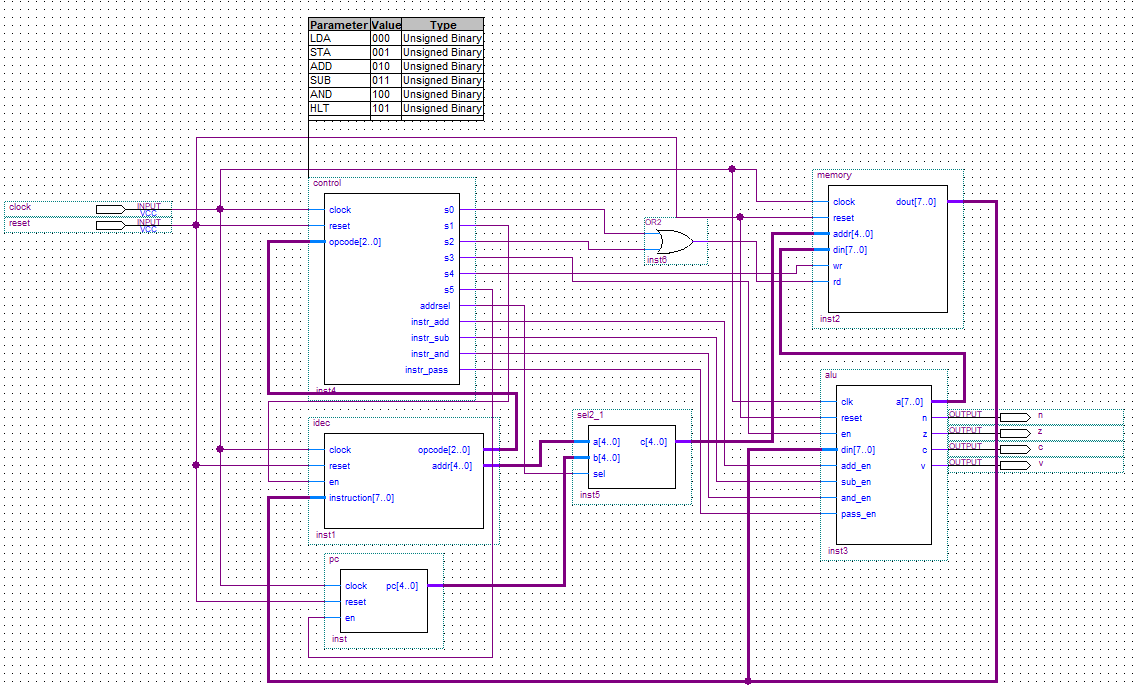


# 实验仿真记录

## PC



## 总体框图：



仿真要看的管脚和观察顺序：(暂时不要将重心放在使能端)

inst1-pc

inst5-c

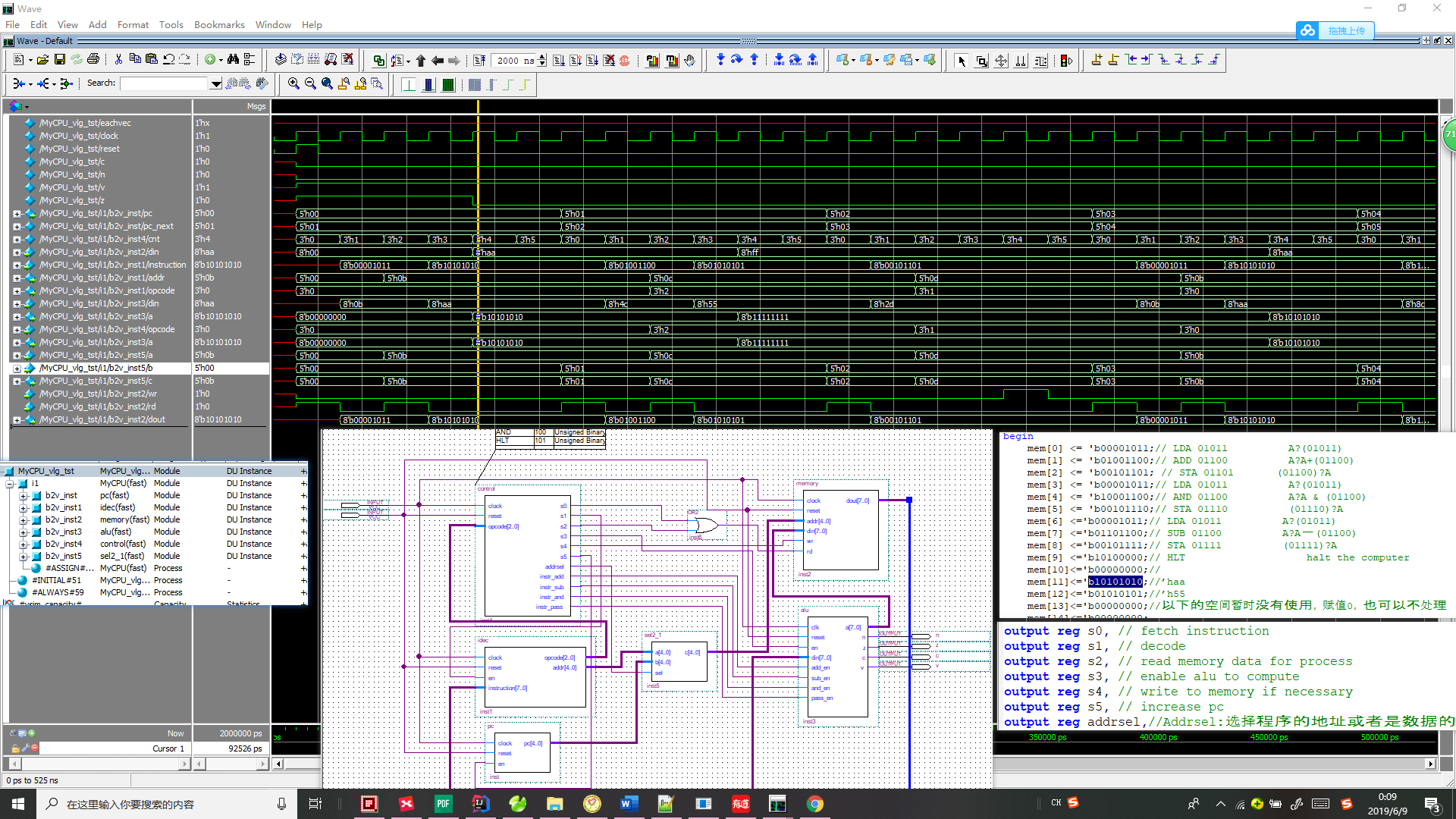
inst2-addr

inst2-dout

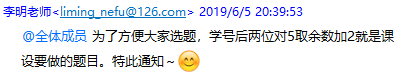
inst1- opcode\addr

inst4- s0\s1\s2\s3\s4\s5->>>???(或者先从s1,s2,s3,s4,s5看起？？？)

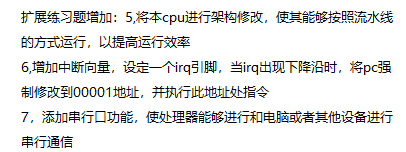
inst5-c----circle



## 总体仿真



# 课设要求



第1题必做，我的学号对应第二题。

## 思路

给pc加两个输入端call[4:0] 和encall,

当encall生效时，pc[4:0]<=call[4:0],pc\_next=pc+1

当encall不生效时,pc=pc\_next;按en生效正常行使功能即可。

addr[4..0]就和call[4..0]相连。

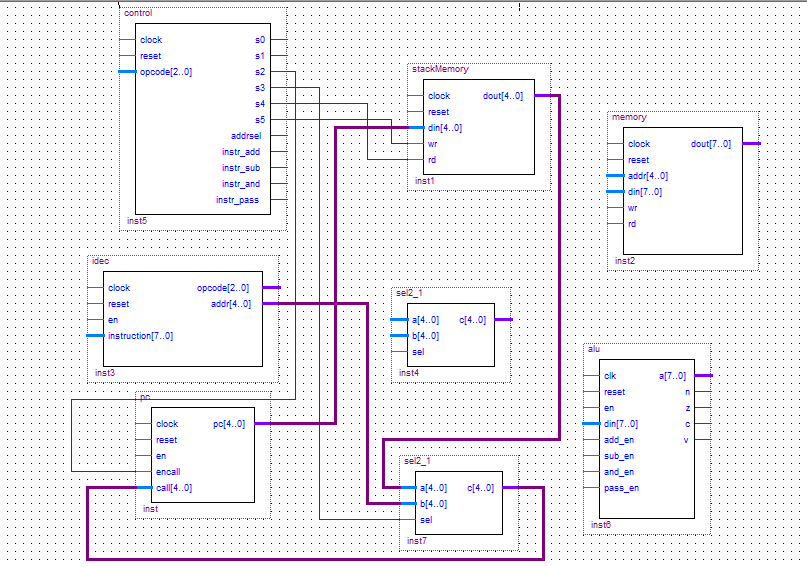
encall和control的s6相连？新加一个。

如果opcode==110，就使能encall(或者说instr\_call)。

但是与此同时还要压栈，也就是向存储器写数据。

使能s3

## 草图



## 全图

