* 实验介绍：

archlab分为Part A, Part B，Part C三部分。在Part A中，你要编写Y86-64程序并熟悉Y86-64工具。在Part B中，你要使用新的指令扩展SEQ模拟器。A和B部分是C的基础，在Part C中，你需要优化Y86-64基准测试程序和处理器设计。

* PART A:

在sim/misc文件里，有examples.c文件，里面有三个函数，sum,rsum,copy。编写这三个函数的y86-64 code.

**注意：要在每个函数之前加上注释写上你的姓名学号。**

sum.ys：迭代求和链表元素

编写一个Y86-64程序sum.ys，以迭代方式求和链表的元素。

rsum.ys：递归求和链表元素

编写一个Y86-64程序rsum.ys，该程序递归求和链表的元素。

copy.ys：将源块复制到目标块

编写程序（copy.ys），将一个单词块从内存的一部分复制到另一部分（非

重叠区域）的内存区域，计算所有复制单词的校验和（xor）。

调试工具：

**YAS**：Y86-64汇编器，输入.ys结尾的y86 code，输出.yo结尾的object code。

**YIS**：Y86-64指令模拟器，输入.yo结尾的object code，输出执行结果。

**上传文件：学号-sum.ys, 学号-rsum.ys, 学号-copy.ys**

* PART B:

在sim/seq文件夹里，修改**seq-full.hcl**文件，添加**iaddq**指令，可以参考作业4.51和4.52。

**注意：你提交的hcl文件开头添加注释，写上你的学号和姓名；**

**有关iaddq指令所需的计算的描述。可以参考课本图4.18中的描述。**

Build and test：

第一步：在TTY模式下运行诸如asumi.yo（测试iaddq）之类的简单程序，并将结果与​​ISA仿真进行比较：

Unix> ./ssim -t ../y86-code/asumi.yo

如果ISA测试失败，则应通过单步执行模拟器来调试实现。

第二步：在../y86代码中的Y86-64基准程序上对其进行测试。

Unix>（cd ../y86-code; make testssim）

第三步：在../ptest中运行大量的回归测试。

Unix>（cd ../ptest;使SIM = .. / seq / ssim）

**上传文件：学号-seq-full.hcl**

* Part C

任务：在sim/pipe文件夹里修改ncopy.ys和pipe-full.hcl.尽所能提高ncopy.ys性能。

**注意：**

**提交两个文件pipe-full.hcl和ncopy.ys。 每个文件应以带有以下信息的标题注释开头：**

**1. 你的姓名和学号。**

**2. 对代码的描述。在每种情况下，请描述修改代码的方式和原因。**

Build and test:

每次修改ncopy.ys程序时，都可以通过键入unix> make drivers来重建驱动程序。

每次修改pipe-full.hcl文件时，都可以通过键入unix> make psim VERSION = full来重建模拟器。

**上传文件：学号-ncopy.ys, 学号-pipe-full.hcl**

**注：上传文件需要加上学号，如012321313-ncopy.ys**

关于本次实验调试和评分的更多信息，请看archlab.pdf。