实验4 优化Y86-64流水线处理器性能

班级： 07112002 学号： 1120201316 姓名： 金雅各

一、实验目的

本次作业作为第4章处理器体系结构的配套实验，以Y86-64顺序和流水线处理器仿真器为基础，对Y86-64处理器进行性能优化，巩固同学们在课堂上所学的知识。

二、实验内容

第一部分（Part A）：编写Y86-64简单程序，熟悉Y86-64工具。

第二部分（Part B）：扩展SEQ模拟器支持新的指令

第三部分（Part C）：在前两部分的基础上，优化Y86-64基准测试程序和流水线处理器设计

三、实验步骤

Part A:

编写并模拟以下三个 Y86-64 程序。这些程序所需的行为由 examples.c 中的示例C 函数定义。

修改misc文件夹下的Makefile 文件

第2行改为: CFLAGS=-Wall -O1 -g -fcommon

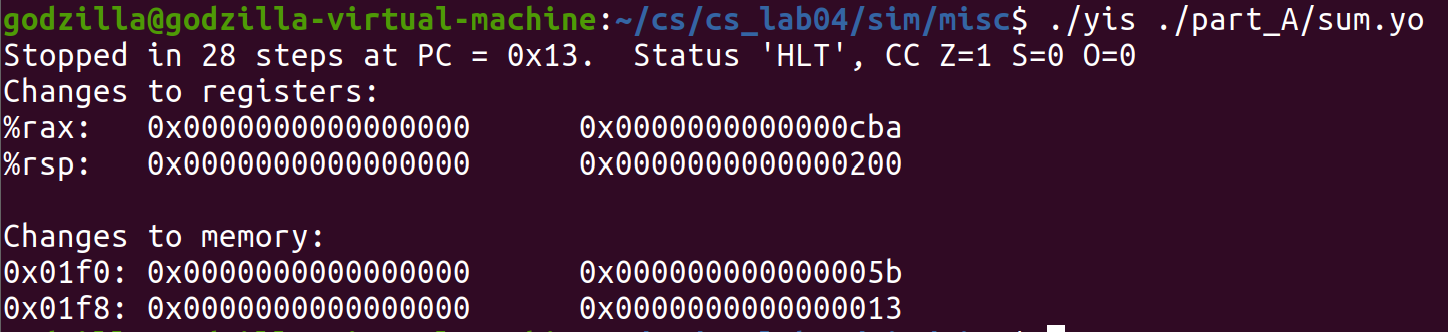
第3行改为: LCFLAGS=-O1 -fcommon

然后输入: make clean; make

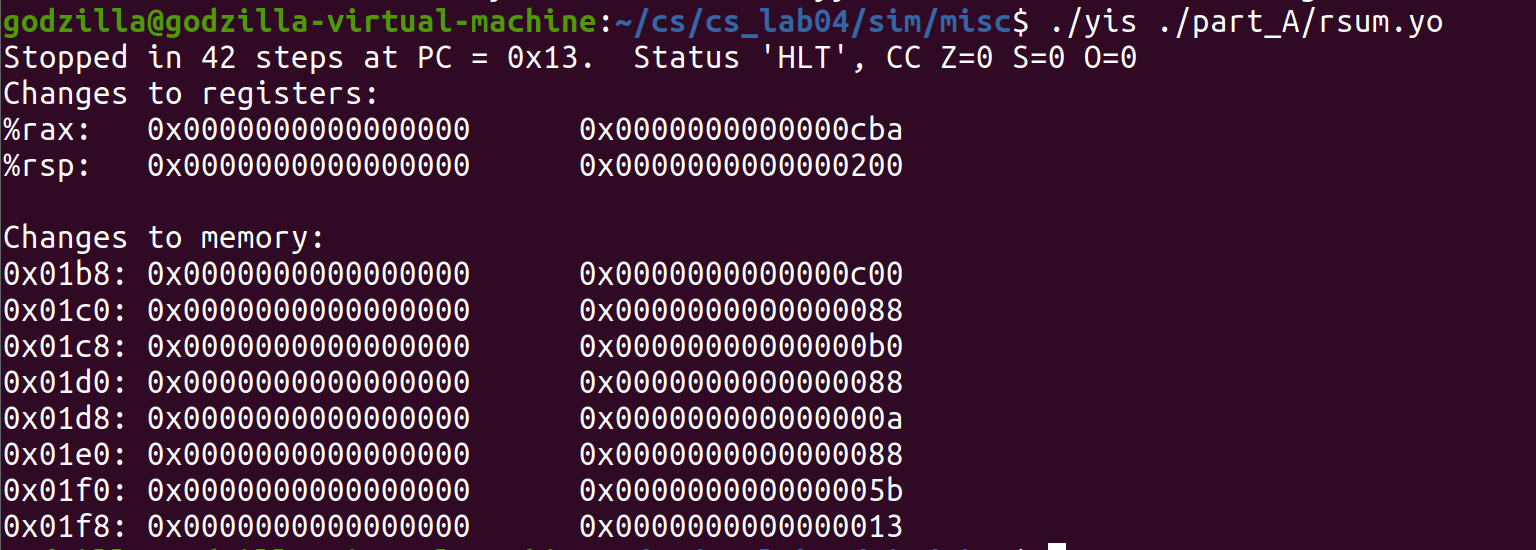
按照example.c 文件中的三个函数，编写sum.ys rsum.ys copy.ys

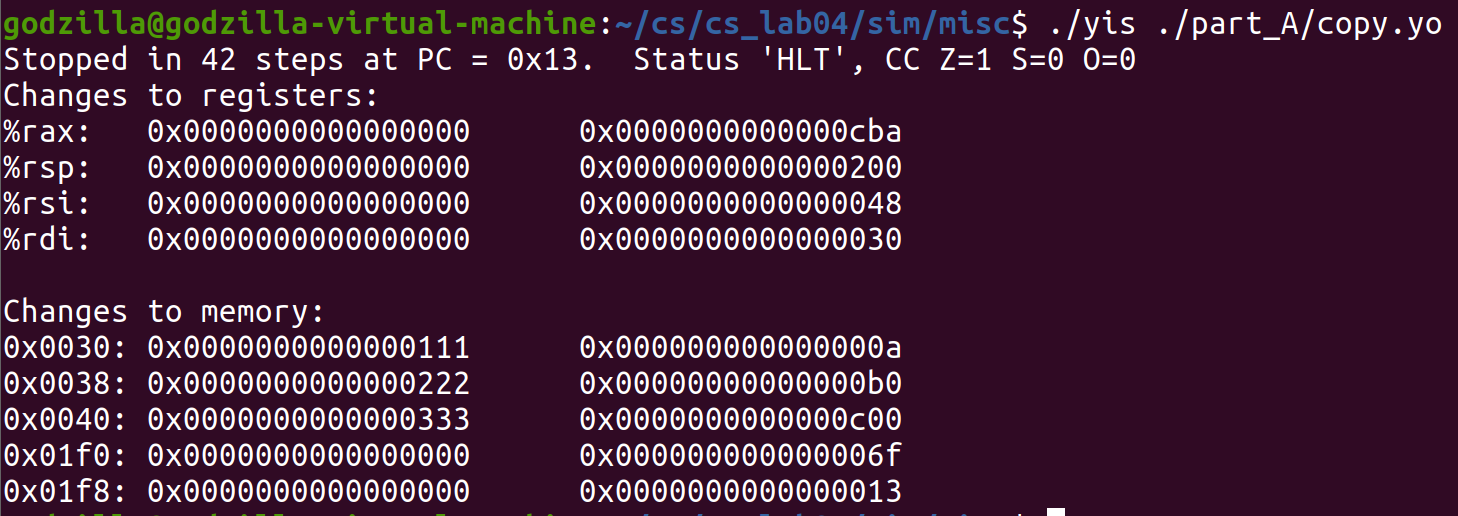
使用yas进行编译，再使用yis运行

运行结果结果如图所示

Sum.ys:

Rsum.ys:

Copy.ys:



Part B:

更改seq下Makefile配置文件

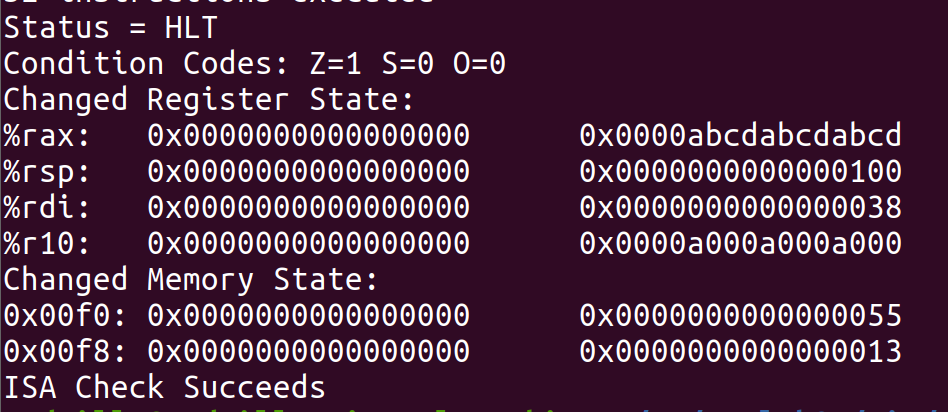
第20行: TKINC=-isystem /usr/include/tcl8.6

第26行: CFLAGS=-Wall -O2 -DUSE\_INTERP\_RESULT

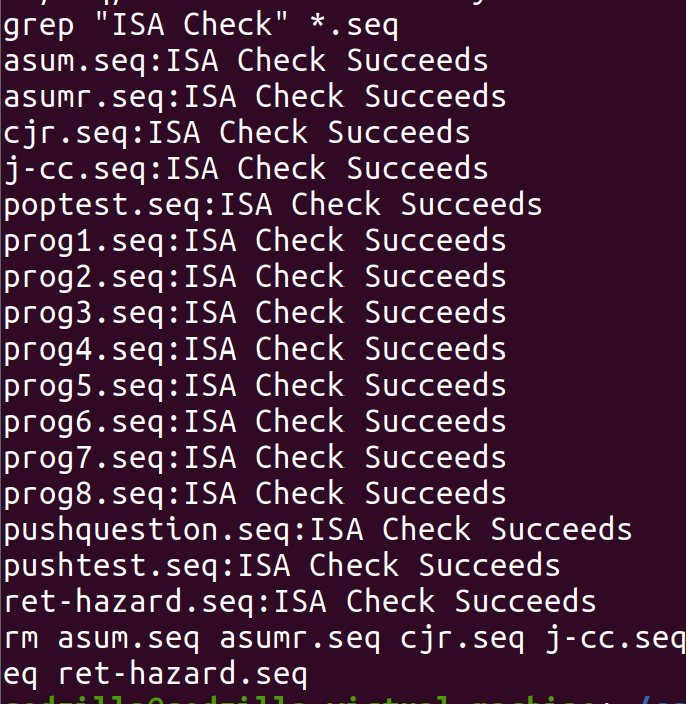
注释掉/sim/pipe/psim.c 806、807 line和/sim/seq/ssim.c 844、845 line。

修改seq-full.hcl 文件，加入iaddq执行令

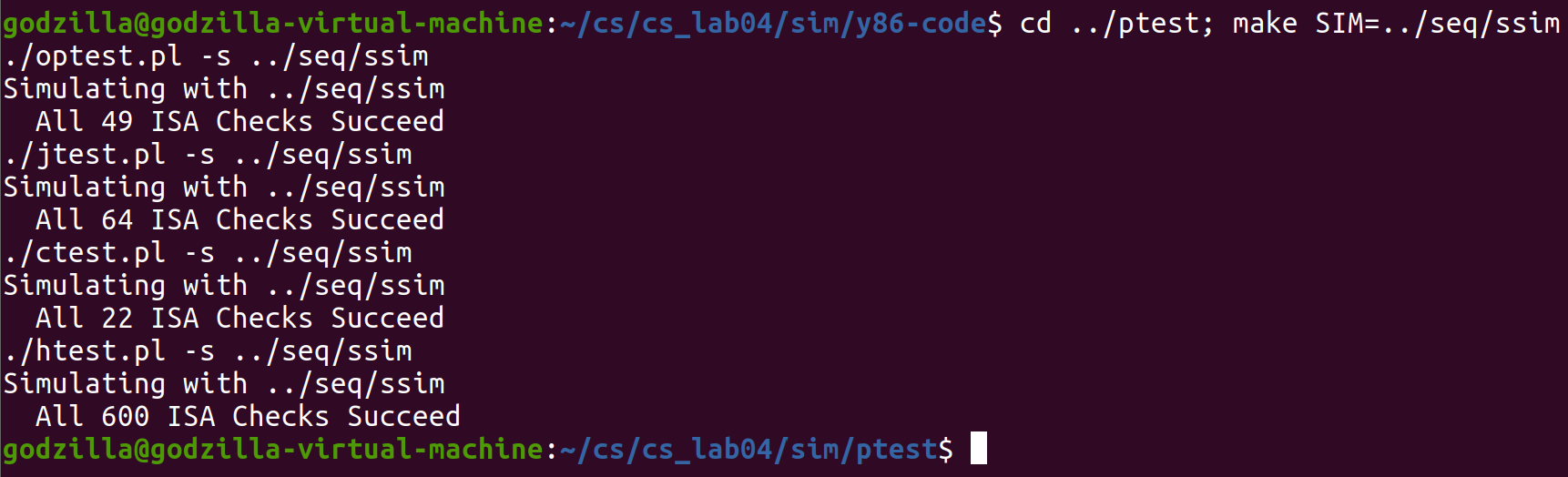
编译ssim: make VERSION=full

初始测试：./ssim -t ../y86-code/asumi.yo

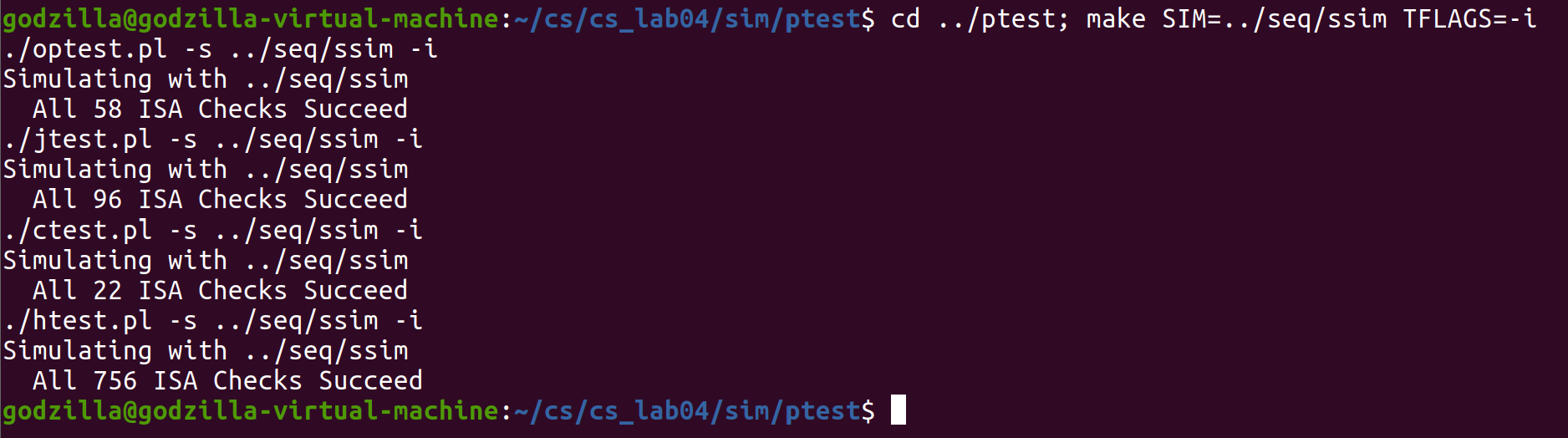
Y86-64 基准程序测试： cd ../y86-code; make testssim



回归测试：cd ../ptest; make SIM=../seq/ssim



测试 iaddq 的实现：cd ../ptest; make SIM=../seq/ssim TFLAGS=-i



Part C:

进入pipe文件夹

修改Makefile文件

第20行: TKINC=-isystem /usr/include/tcl8.6

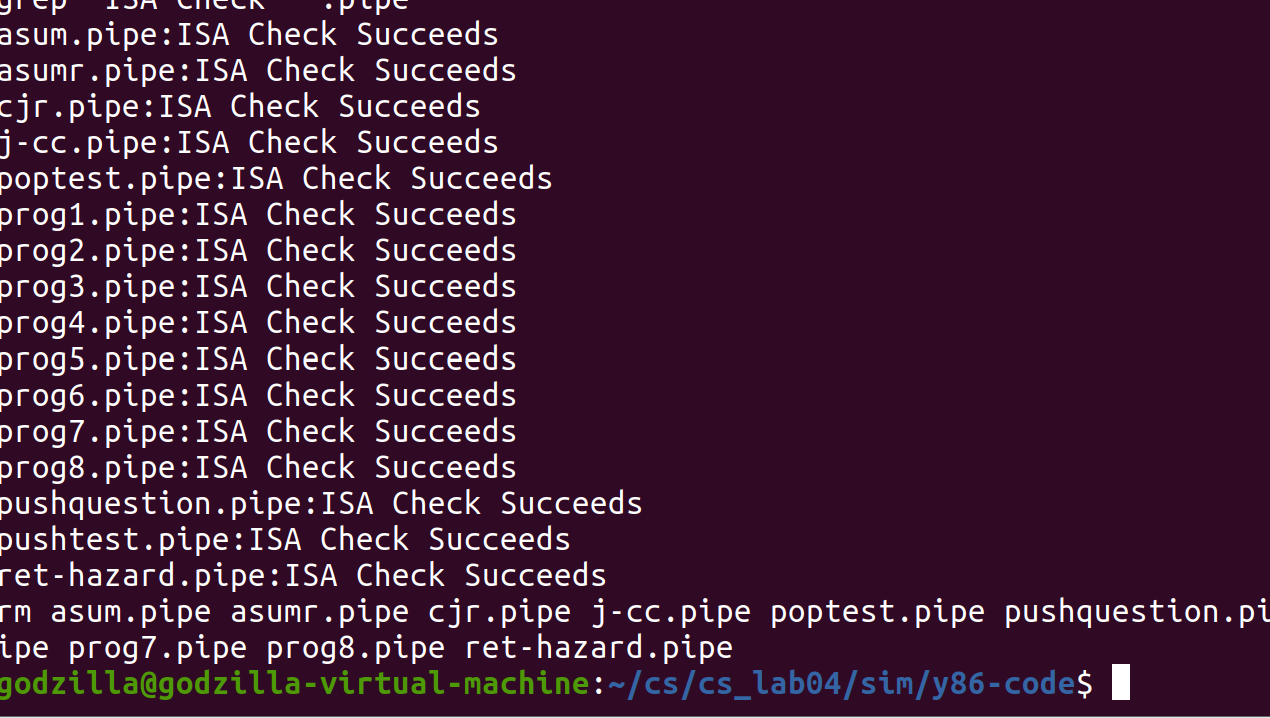
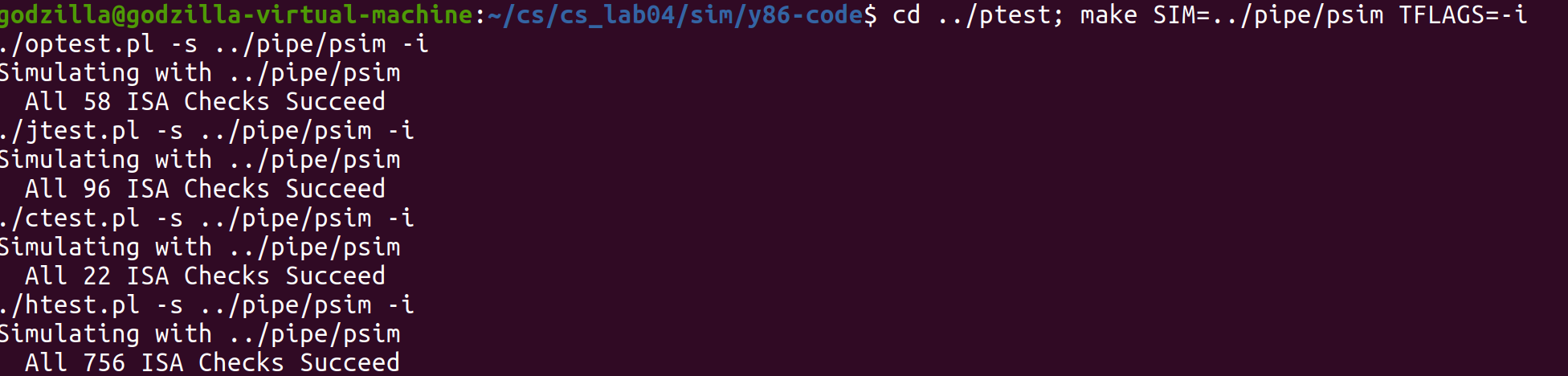
第26行: CFLAGS=-Wall -O2 -DUSE\_INTERP\_RESULT

修改pipe-full.hcl 文件，加入iaddq指令

编译psim: make psim VERSION=full

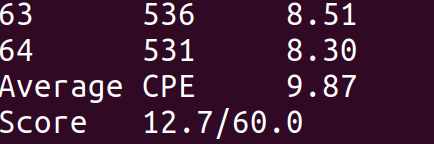
构建驱动程序: make drivers

Y86-64 基准程序进行测试：cd ../y86-code; make testpsim

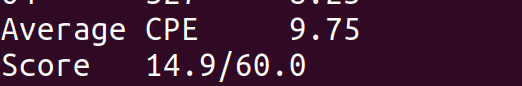
回归测试：cd ../ptest; make SIM=../pipe/psim TFLAGS=-i

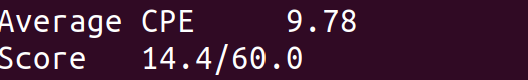
编写4X4循环扩展的ncopys函数

评估结果：./correctness.pl -p

改为6X6循环扩展

评估结果: ./correctness.pl -p

结果有所提升，改为8X8循环展开：

结果下降，故选择6X6循环展开的形式，考虑在其他方面改善程序性能

仔细观战ncopy.ys代码，发现

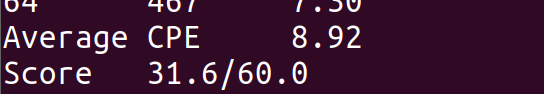
mrmovq (%rdi), %r8

rmmovq %r8, ()%rsi

两条指令之间存在数据相关，

故交换指令的顺序，先读入6个数据到寄存器r8-r13中再将6个寄存器的值写回内存，同时使用条件传送指令代替条件指令

结果有明显提升：

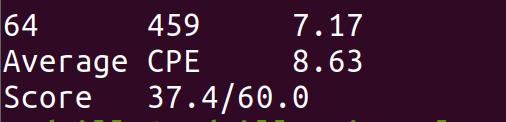
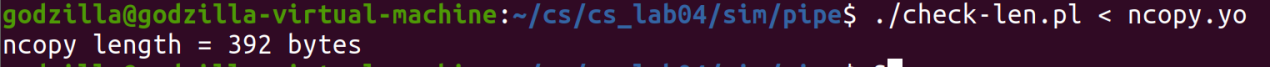
由于条件语句默认选择分支，且只有少本分数组个数在6一下，故交换6X6与1X1模块的位置，初始指令改为：

xorq %rax, %rax # rax 统计正整数的个数

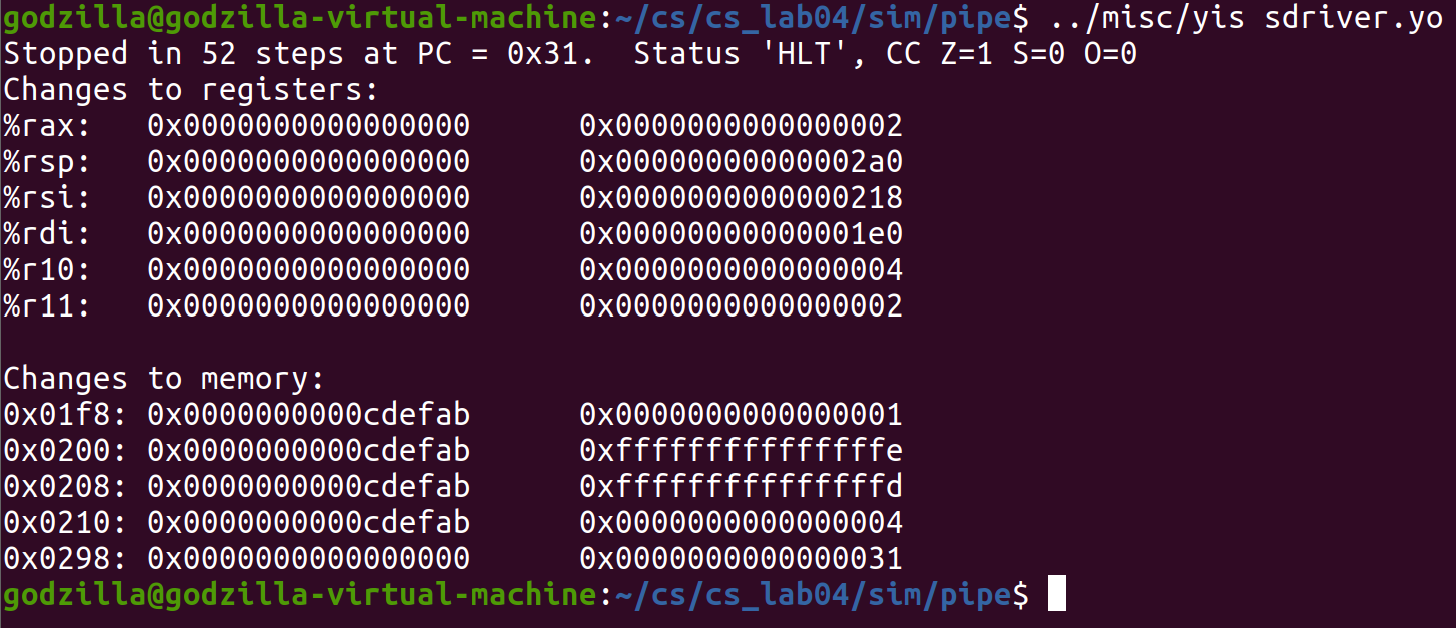
iaddq $-5 , %rdx # limit=len-5

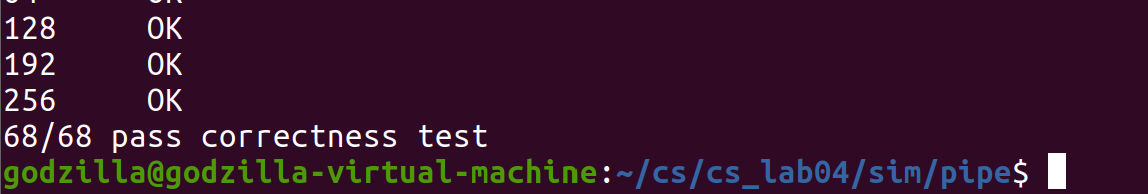
jg Loop6\_6

结果有所提升：

Ncopy.ys 的汇编版本长度为： ./check-len.pl < ncopy.yo

确保您的 ncopy.ys 函数与 YIS 一起正常工作： ../misc/yis sdriver.yo

使用 ISA 模拟器在一系列块长度上测试代码： ./correctness.pl



四、实验收获与体会

通过本次实验，通过理论与实践的结合，我对于处理器优化有了更深的了解，

其中循环展开。数据相关以及条件语句对于程序的性能有较大的影响。