Санкт-Петербургский политехнический университет Петра Великого Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 2** Дисциплина: Низкоуровневое программирование Тема: Программирование RISC-V

Вариант: 13

Выполнил студент гр. 3530901/00002 А.Д. Чешев

(подпись)

Принял преподаватель Д.С. Степанов

(подпись)

“ ” 2021 г.

Санкт-Петербург

# Постановка задачи:

# 1. Изучить методические материалы, доступные на сайте курса.

# 2. Разработать программу на языке ассемблера RISC-V, реализующую определенную вариантом задания функциональность, отладить программу в симуляторе VSim/Jupiter. Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.

# 3. Выделить определенную вариантом задания функциональность в подпрограмму, организованную в соответствии с ABI, разработать использующую ее тестовую программу. Адрес обрабатываемого массива данных и другие значения передавать через параметры подпрограммы в соответствии с ABI. Тестовая программа должна состоять из инициализирующего кода, кода завершения, подпрограммы main и тестируемой подпрограммы.

# Вариант:

# Расчет значения многочлена по схеме Горнера с «длинным» результатом (переполнение игнорируется).

# 2. Код программы:

# .text

# start:

# .globl start

# la a2, array #addres of first array element

# lw a3, array\_size

# lw a4, x

# li a5, 0 #result

# lw t1, 0(a2)

# li a7, 1 #a7 = 1

# loop:

# mul a5, a5, a4

# add a5, a5, t1

# addi a2, a2, 4

# lw t1, 0(a2)

# sub a3, a3, a7

# beqz a3, exit

# jal zero, loop

# exit:

# li a0, 17

# li a1, 1 # код завершения 1 ~ exit(1)

# ecall

# 

# .rodata

# x:

# .word 3

# .data

# array\_size:

# .word 6

# .rodata

# array:

# .word 3, 2, 6, 1, 4, 15

# Данная программа вычисляет значение многочлена: при x = 3. Таким образом получается, что y = 1089. Вывод программы осуществлен в ячейку a5.

# 3. Код программы, тестовой программы и подпрограммы :

# Программа:

# .text

# start:

# .globl start

# call subprog\_main

# exit:

# li a0, 17

# li a1, 1 # код завершения 1 ~ exit(1)

# ecall

# Тестовая программа:

# .text

# subprog\_main:

# .globl subprog\_main

# la a2, array #addres of first array element

# lw a3, array\_size

# lw a4, x

# li a5, 0 #result

# addi sp, sp, -16

# sw ra, 12(sp)

# call subprog

# lw ra, 12(sp)

# addi sp, sp, 16

# ret

# 

# .rodata

# x:

# .word 3

# .data

# array\_size:

# .word 6

# .rodata

# array:

# .word 3, 2, 6, 1, 4, 15

# Подпрограмма:

# .text

# subprog:

# .globl subprog

# lw t1, 0(a2)

# li a7, 1 #a7 = 1

# loop:

# mul a5, a5, a4

# add a5, a5, t1

# addi a2, a2, 4

# lw t1, 0(a2)

# sub a3, a3, a7

# beqz a3, exit

# jal zero, loop

# exit:

# ret

# Вывод:

# В ходе лабораторной работы были изучены принципы программирования на RISC-V ассемблер, получены практические навыки построения программ.

# Список используемых источников:

# <https://github.com/riscv-non-isa/riscv-asm-manual/blob/master/riscv-asm.md>

# http://kspt.icc.spbstu.ru/course/lowlevelprog