Санкт-Петербургский политехнический университет Петра Великого
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Дисциплина: Низкоуровневое программирование

Тема: Программирование RISC-V

Вариант: 13

Выполнил студент гр. 3530901/90002 _____ Д. Е. Бакин
(подпись)

Преподаватель _____ Д. С. Степанов
(подпись)

“__” _____2021 г.

Санкт-Петербург

2021

1 Задачи работы

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

2 Ход Работы

Функциональность, определенная вариантом задания:

Реализовать вычисление определенного полинома, с целыми коэффициентами, в точке x0 (x0 - int) при помощи схемы Горнера.

Схема Горнера:

Задан многочлен:

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_n x^n, \quad a_i \in \mathbb{R}.$$

Требуется вычислить значение данного многочлена при фиксированном значении $x = x_0$. Представим многочлен $P(x)$ в следующем виде:

$$P(x) = a_0 + x(a_1 + x(a_2 + \cdots x(a_{n-1} + a_n x)\ldots)).$$

Коэффициенты a_0, a_1… a_n задаются в виде массива, где первый элемент — это старший коэффициент a_n, где n – степень полинома.

Также в памяти нужно задать длинну массива, и сам x_0.

Предполагается что массив состоит минимум из 2 элементов, иначе полином равен константе или нулю и ничего вычислять не требуется.

2.1 Основная программа

```
.rodata                 # секция неизменяемых данных
array_length:           # создание переменной длинны массива (степень многочлена + 1)
.word 2                 # минимум 2 элемента (иначе, полином равен константе)
x:                      # создание переменной X0
.word 1
array:
.word 1, 227            # сам массив (состоит из коэффициентов многочлена)

.text                   # директива размещения инструкций в секции кода
__start:                # метка "точки" программы
.globl __start          # эта директива старта выполнения программы

addi t0, zero, 1        # i = 1
la t2, array_length
lw t2, 0(t2)            # t2 = array.size
la t1, array            # t1 = array[0]/address
lw a2, 0(t1)            # a2 = array[0]
la t3, x
lw t3, 0(t3)            # t3 = x

j checkLoop             # goto checkLoop
loop:
slli t4, t0, 2          # t4 = i*4 (смещение по адресу)
add t5, t1, t4          # t5 = array[i]/address
lw t5, 0(t5)            # t5 = array[i]
mul a2, a2, t3          # a2 *= x
add a2, a2, t5          # a2 += array[i]
addi t0, t0, 1          # i++
checkLoop:
bgtu t2,t0, loop        # if(array.size>i) goto loop

li a0, 1                # a0 = 1
mv a1, a2               # a1 = a4
ecall                   # системный вызов, печать результата

li a0, 10
ecall
```

Рис. 1 Код программы.

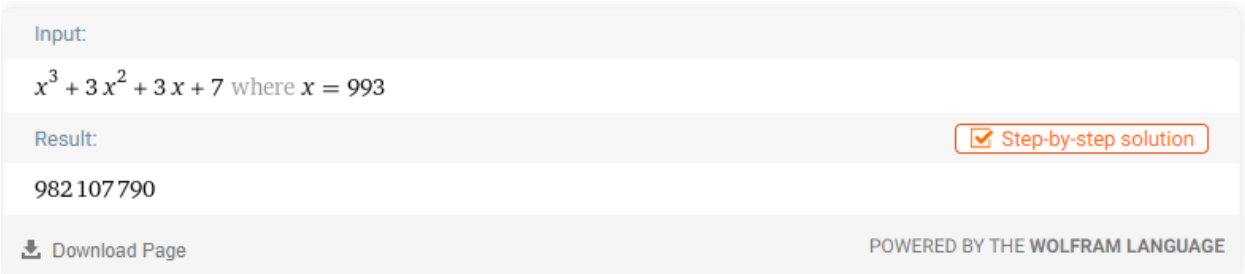
В поля с директивой .rodata записывются входные данные.

Программа на выходе печатает значение регистра a2, где хранится результат.

Примеры работы программы:

С полиномом 227x + 1, при x = 1, программа выдала 228, что соответствует действительному значению полиному в этой точке.

С более сложным полиномом $1*x^3 + 3x^2+3x+7$, в точке 993, программа выдала 982107790, что также является верным ответом:



2.2 Программа в виде подпрограммы

```
.rodata                 # секция неизменяемых данных
array_length:           # создание переменной длинны массива (степень многочлена + 1)
.word 2                 # минимум 2 элемента (иначе, полином равен константе)
x:                      # создание переменной X0
.word 2
array:
.word 1, 226            # сам массив (состоит из коэффициентов многочлена)
array_length2:
.word 3
array2:
.word 1, 3, 5
msg: .string " "

.text                   # директива размещения инструкций в секции кода
__start:                # метка "точки" программы
.globl __start          # эта директива старта выполнения программы

la a1, array_length
lw a1, 0(a1)            # a1 = array.size
la a2, array            # a2 = array[0]/address
la a3, x
lw a3, 0(a3)            # a3 = x
jal ra, fun             # вызов функции

li a0, 4                # a0 = 1
la a1, msg              # a1 = " "
ecall

la a1, array_length2
lw a1, 0(a1)            # a1 = array.size
la a2, array2           # a2 = array[0]/address
la a3, x
lw a3, 0(a3)            # a3 = x
jal ra, fun             # вызов функции

li a0, 10
ecall

fun:
mv t3, a3               # t3 = x
mv t2, a1               # t2 = array.size
mv t1, a2               # t1 = array[0]/address
lw a2, 0(t1)            # a2 = array[0]
addi t0, zero, 1        # i = 1

j checkLoop             # goto checkLoop
loop:
slli t4, t0, 2          # t4 = i*4 (смещение по адресу)
add t5, t1, t4          # t5 = array[i]/address
lw t5, 0(t5)            # t5 = array[i]
mul a2, a2, t3          # a2 *= x
add a2, a2, t5          # a2 += array[i]
addi t0, t0, 1          # i++
checkLoop:
bgtu t2,t0, loop        # if(array.size>i) goto loop

li a0, 1                # a0 = 1
mv a1, a2               # a1 = a4
ecall                   # системный вызов, печать результата
ret                     # return
```

Рис. 2 Код программы с подпрограммой.

Здесь мы из основной программы два раза вызываем нашу программу с разными даннымия, и результат выводится в консоль через пробел.

Вызов подпрограммы с полномами $5*x^3+5x+1$ и $6*x^3+6x+5$, в точке x=1.

Программа выдала, верный результат:

3 Вывод

В данной работе мы ознокомились с ISA процессора RISC-V. Была написана программа и подпрограмма, заданная по варианту.

При разработке использовался эмулятор:

https://github.com/andrescv/Jupiter/releases/tag/v3.1