МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА – Российский технологический университет»

Институт информационных технологий

Кафедра вычислительной техники

Отчет

По лабораторному практикуму

дисциплины:

«Архитектура вычислительных машин и систем»

Исполнитель: Желудков Данила Михайлович

Ладыгин Сергей Алексеевич

студенты группы ИКБО-07-19

Руководитель: Скрябин Юрий Михайлович

ассистент кафедры ВТ

Москва 2020

**Лабораторная работа №1**

**ПРОЦЕССОР С РЕГИСТРОВОЙ ПАМЯТЬЮ**

*Цель работы*: изучение алгоритма работы процессоров, использующих регистры для хранения операндов и результатов арифметических и логических операций.

*Задание к работе*: Описать на ассемблере работу одноадресного 32-битного процессора, при следующих операциях с одномерным массивом из 32-битных чисел: а) Найти сумму элементов массива б) Произвести сортировку массива по четности (12 вариант)

Работу выполнил:

Практическая часть

%include "io64.inc"

section .data

vec dd 5, 5, 4, 3, 2, 1

iter dd 0

iterTop dd 0; индекс текущего элемента

iterEax dd 0; индекс последнего элемента

elem dd 0; текущий элем

section .text

global CMAIN

CMAIN:

mov rbp, rsp; for correct debugging

xor rax, rax

xor rdx, rdx

xor rcx, rcx

;сумма

mov ecx, [vec] ;сохраняем адрес нулевого элемента

mov edx, vec + 4 ; передаем первый элемент

l1:add eax, [edx], edx;суммируем каждый эелемент

add edx, 4, edx; получаем адрес следующего элемента

loop l1

;сортировка

xor rax, rax

xor rdx, rdx

xor rcx, rcx

xor ebx, ebx

xor esi, esi

;mov ebx, [vec +4]; хранит первый элемент

mov ecx, [vec]; хранит нулевой элемент (количество элементов массива)

;mov esi, [vec]

for\_i:

mov edx, 1 ; индекс начинаем с единицы

for\_j:

mov eax, 4; eax выступает в роли текущего элемента

mov [iter], edx; Помещаем в iter количество пройденных элементов

mul edx; перемножаем eax на edx и тем самым получаем индекс текщего элемента, при этом edx обнуляется

mov edx, [iter];Возвращаем значение edx

mov [iterTop], eax;Запоминаем значение eax чтобы потом узнать индекс первого числа

mov eax, 4

mul ecx; умножаем eax на количество элементов, тем самым узнаём индекс последнего

mov [iterEax], eax; сохраняем инд последнего элемента

mov ebx, [iterTop]; помещаем индекс текущего элемента

mov eax, [vec + ebx]; помещаем текущий элемент

; mov [elem], eax ; запомнили текущий элемент

test eax, 1; проверили на четность

jz no\_change; если четное, то не меняем

mov edx, [iterEax]; поместили инд последнего

mov ebx, [vec + edx]; поместили последний

xchg eax, ebx; поменяли местами значения

mov edx, [iterTop]; поместили инд текущего

mov [vec + edx], eax; записали в место текущего элемента последний

mov edx, [iterEax]; поместили индекс последнего

;mov [vec + 20], ebx

no\_change:

mov edx, [iter]

cmp edx, ecx

jae exit\_for\_j

;jbe no\_swap

mov eax, edx

;no\_swap:

inc edx

jmp for\_j

exit\_for\_j:

loop for\_i

ret