**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет Программной Инженерии и Компьютерной Техники**

**Отчет по лабораторной работе №5:**

**Вариант 34054**

Студент:

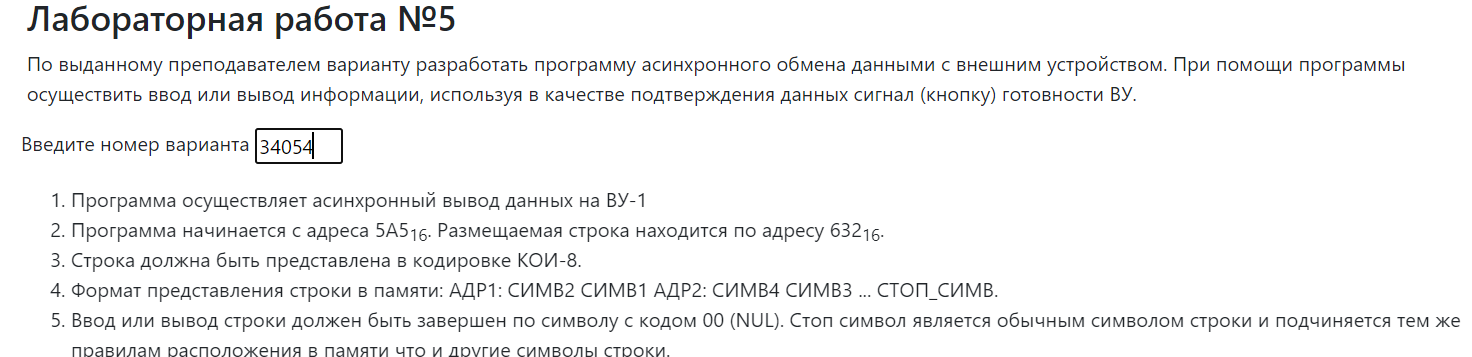
Иванов Илья Андреевич

Р3117

Преподаватель:

Блохина Елена Николаевна

Санкт-Петербург, 2024



**Исходное слово:** УраН&

**Кодировки исходного слова:**

UTF-8: d0a3 d180 d0b0 d09d 26

UTF-16: 0423 0440 0430 041d 0026

КОИ-8: f5 d2 c1 ee 26

Необходимо доработать программу следующим образом:

Ввод осуществляется с ВУ-3, вывод на ВУ-7

Все остальное согласно первоначальному заданию

Xxxx xxxx xxxxxxxx

КОП

0001 0010 0000 0101

На индикатор надо вывести 12345678

**Что вводить на ву-3:**

0111 0001 на позицию **7** цифру **1**

0110 0010 на позицию **6** цифру **2**

0101 0011 на позицию **5** цифру **3**

0100 0100 на позицию **4** цифру **4**

0011 0101 на позицию **3** цифру **5**

0010 0110 на позицию **2** цифру **6**

0001 0111 на позицию **1** цифру **7**

0000 1111 на позицию **0** цифру **8**

0000 0000 **стоп символ**

**Программа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии | |
| 5AA | 0200 | CLA | Очистка акк-ма |
| 5AB | AEF9 | LD (IP-6) | Загрузка адреса первого символа |
| 5AC | EEF9 | S T (IP-6) | Установка адреса в переменную |
| 5AD | AAF8 | LD (IP-7)+ | Загрузка адреса следующего символа |
| 5AE | EEF8 | ST (IP-7) | Установка 8 бит в переменную |
| 5AF | 0680 | SWAB | Смена первых 8 бит со вторыми |
| 5B0 | EEF7 | ST (IP-8) | Установка 8 бит в переменную | |
| 5B1 | 1203 | IN 3 | Ожидание готовности SR (#3)  (SR(#3) -> AC) | |
| 5B2 | 2F40 | AND #0x40 | M&AC-> AC  Проверяет нажата ли кнопка «готов»  Смотря в 6й бит | |
| 5B3 | F0FD | BEQ (IP-3) | Если Z==1 то IP-3+1->IP  Так называемый «спин-луп» | |
| 5B4 | AEF2 | LD (IP-13) | Сохранение в переменную первого символа | |
| 5B5 | 2EF3 | AND (IP-12) | Сравнение с «стоп» символом | |
| 5B6 | F009 | BEQ (IP+9) | Если Z==1 то IP+9+1->IP | |
| 5B7 | 1302 | OUT 2 | Вывод 1го символа  Младшие 8 бит в ВУ-1 | |
| 5B8 | 1203 | IN 3 | Ожидание готовности SR (#3)  (SR(#3) -> AC) | |
| 5B9 | 2F40 | AND #0x40 | M&AC-> AC  Проверяет нажата ли кнопка «готов»  Смотря в 6й бит | |
| 5BA | F0FD | BEQ (IP-3) | Если Z==1 то IP-3+1->IP  Так называемый «спин-луп» | |
| 5BB | AEEC | LD (IP-20) | Сохранение в переменную второго символа | |
| 5BC | 2EEC | AND (IP-20) | Сравнение с «стоп» символом | |
| 5BD | F002 | BEQ (IP+2) | Если Z==1 то IP+2+1->IP | |
| 5BE | 1302 | OUT 2 | Вывод 2го символа  Младшие 8 бит в ВУ-1 | |
| 5BF | CEED | JUMP (IP-19) | Переход в ячейку | |
| 5C0 | 0100 | HLT | Остановка программы | |

**Описание программы:**

Программа осуществляет вывод данных на ВУ-1

**ОДЗ:**

Допустимое расположение символов в памяти:

Так как в ТЗ написано, что начало должно быть в ячейке 632, то начиная с нее и до 7FF

**Адрес первой и последней команд:**

Первая: 5AA

Последняя: 5C0

**Адреса переменных программы:**

5A5-5A9

**Расположение строки:**

632-634

**Код программы на Ассемблере:**

ORG 0x5A5

BEGIN: WORD 0x632 ; адрес начала строки

CUR: WORD ? ; текущий символ

FIRST: WORD ? ; первый символ

SECOND: WORD ? ; второй символ

FINISH: WORD 0x00FF ; символ выхода

START: CLA

LD BEGIN

ST CUR

READ: LD (CUR)+

ST FIRST

SWAB

ST SECOND

L1: IN 3

AND #0x40

BEQ L1

LD FIRST

AND FINISH

BEQ STOP

OUT 2

L2: IN 3

AND #0x40

BEQ L2

LD SECOND

AND FINISH

BEQ STOP

OUT 2

JUMP READ

STOP: HLT

**Код допа:**

res:WORD 0x632; Ссылка на результат

finish:WORD 0x0; Стоп-символ

temp:WORD ? ; Ячейка для записи нечетных символов

START: CLA

s1:IN 7

AND #0x40;

BEQ s1;

IN 6

ST (res)

ST temp;

CMP finish

BEQ exit

CLA

s2:IN 7

AND #0x40

BEQ s2

IN 6

SWAB;Перемещаем четный символ в старший байт

OR temp; Совмещаем с 1-м символом

ST (res); Сохраняем в память по ссылки

SUB temp; Вычитаем 1-й символ

SWAB; Перемещаем четный символ в младший байт

CMP finish

BEQ exit

LD (res)+

CLA

JUMP s1

exit: LD (res)+

BEGIN: WORD 0x632 ; адрес начала строки

CUR: WORD ? ; текущий символ

FIRST: WORD ? ; первый символ

SECOND: WORD ? ; второй символ

FINISH: WORD 0x00FF ; символ выхода

CLA

LD BEGIN

ST CUR

READ: LD (CUR)+

ST FIRST

SWAB

ST SECOND

L1: IN 0x15

AND #0x40

BEQ L1

LD FIRST

AND FINISH

BEQ STOP

OUT 0x14

L2: IN 0x15

AND #0x40

BEQ L2

LD SECOND

AND FINISH

BEQ STOP

OUT 0x14

JUMP READ

STOP: HLT