Университет ИТМО

Факультет ПИиКТ

Основы профессиональной деятельности

Лабораторная работа №5

Вариант 1353

Выполнил

Студент P3130 Лысенко Артём

Преподаватель

Ткешелашвили Нино Мерабиевна

Г. Санкт-Петербург, 2022

**1) Задание**

1. Программа осуществляет асинхронный вывод данных на ВУ-3
2. Программа начинается с адреса 5C916. Размещаемая строка находится по адресу 60716.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП\_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу c кодом 0A (NL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

|  |  |  |  |
| --- | --- | --- | --- |
| адрес | Код | Мнемоника | Комментарии |
| 5C8 | 0000 | ELEMENT | Адрес текущего элемента массива. |
| 5C9 | +0200 | CLA | Очистка аккумулятора. |
| 5CA | 1207 | IN 7 | Проверка готовности ВУ-3 к записи. |
| 5CB | 2F40 | AND 40 |
| 5CC | F0FD | BEQ IP-3 |
| 5CD | A8FA | LD (IP-6) | Загрузка элемента массива. |
| 5CE | 0680 | SWAB | Меняем байты местами, чтобы получить первый символ. |
| 5CF | 0600 | SXTB |
| 5D0 | 7F0A | CMP #0A | Проверка на СТОП\_СИМВ. |
| 5D1 | F00А | BEQ IP+B | Конец программы, если символ равен СТОП\_СИМВ |
| 5D2 | 1306 | OUT 6 | Вывод первого символа |
| 5D3 | 1207 | IN 7 | Проверка готовности ВУ-3 к записи. |
| 5D4 | 2F40 | AND 40 |
| 5D5 | F0FD | BEQ IP-3 |
| 5D6 | A8FA | LD (IP-6) | Загрузка элемента массива. |
| 5D7 | 0600 | SXTB | Расширение знака символа |
| 5D8 | 1306 | OUT 6 | Вывод второго символа |
| 5D9 | 7F0A | CMP #0A | Проверка на СТОП\_СИМВ. |
| 5DA | F002 | BEQ IP+2 | Конец программы, если символ равен СТОП\_СИМВ |
| 5DB | AAED | LD (IP-20)+ | Получаем следующий элемент массива. |
| 5DC | CEEE | JUMP IP-19 | Делаем все то же самое. |
| 5DD | 0100 | HLT | Останов. |
| ------ | ------- | -------------- | ------------------------------------------------------------ |
| 607 | 1122 | X1X2 | Массив символов |
| 608 | 3344 | X3X4 |
| … | … | …. |

**2) Описание программы**

Программа берет элементы массива, которые представляют из себя 2 символа, и выводит их на ВУ-3.

Исходные данные – 5С8.

Адрес первой команды – 5C9

Адрес последней команды – 5DC

Исходные данные – начинается с 607.

**Область представления**

5С8 – адрес текущего элемента массива – 16-разрядное беззнаковое число.

Массив символов – 16-разрядные символы.

**ОДЗ**

Адрес начала строки [607;7FF] U [0;5C7]

Длина строки может быть любой, но не может заходить на тело программы. Максимальная длина строки равна количеству ячеек памяти в БЭВМ.

Допустимые коды символов – [20;FF]

**3) Программа на ассамблере**ORG 0x5C6

ELEMENT: WORD $ARRAY

START: CLA

SPIN1: IN 7

AND #0x40

BEQ SPIN1

LD (ELEMENT)

SWAB

SXTB

OUT 6

CMP #0x0A

BEQ BREAK

SPIN2: IN 7

AND #0x40

BEQ SPIN2

LD (ELEMENT)+

SXTB

OUT 6

CMP #0x0A

BEQ BREAK

JUMP SPIN1

BREAK: HLT

ORG 0x607

ARRAY: WORD 0xF7E1, 0xFAE1, 0x0A00

Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров после выполнения команды | | | | | | | | Ячейка, содержимое которой поменялось | | |
| адрес | код | IP | CR | AR | DR | SP | BR | AC | NZVC | | Адрес | Новый код |
| 5С9 | 2F40 | 5CA | 2F40 | 5C9 | 0040 | 000 | 0040 | 0000 | 0100 | |  |  |
| 5СA | F0FD | 5C8 | F0FD | 5CA | F0FD | 000 | FFFD | 0000 | 0100 | |  |  |
| 5C8 | 1207 | 5C9 | 1207 | 5C8 | 1207 | 000 | 05C8 | 0040 | 0100 | |  |  |
| 5C9 | 2F40 | 5CA | 2F40 | 5C9 | 0040 | 000 | 0040 | 0040 | 0000 | |  |  |
| 5CA | F0FD | 5CB | F0FD | 5CA | F0FD | 000 | 05CA | 0040 | 0000 | |  |  |
| 5CB | A8FA | 5CC | A8FA | 607 | F7E1 | 000 | FFFA | F7E1 | 1000 | |  |  |
| 5CC | 0680 | 5CD | 0680 | 5CC | 0680 | 000 | 05CC | E1F7 | 1000 | |  |  |
| 5CD | 0600 | 5CE | 0600 | 5CD | 0600 | 000 | 05CD | FFF7 | 1000 | |  |  |
| 5CE | 7F0A | 5CF | 7F0A | 5CE | 000A | 000 | 000A | FFF7 | 1001 | |  |  |
| 5CF | F00A | 5D0 | F00A | 5CF | F00A | 000 | 05CF | FFF7 | 1001 | |  |  |
| 5D0 | 1306 | 5D1 | 1306 | 5D0 | 1306 | 000 | 05D0 | FFF7 | 1001 | |  |  |
| 5D1 | 1207 | 5D2 | 1207 | 5D1 | 1207 | 000 | 05D1 | FF40 | 1001 | |  |  |
| 5D2 | 2F40 | 5D3 | 2F40 | 5D2 | 0040 | 000 | 0040 | 0040 | 0001 | |  |  |
| 5D3 | F0FD | 5D4 | F0FD | 5D3 | F0FD | 000 | 05D3 | 0040 | 0001 | |  |  |
| 5D4 | AAF1 | 5D5 | AAF1 | 607 | F7E1 | 000 | FFF1 | F7E1 | 1001 | | 5C6 | 0608 |
| 5D5 | 0600 | 5D6 | 0600 | 5D5 | 0600 | 000 | 05D5 | FFE1 | 1001 | |  |  |
| 5D6 | 7F0A | 5D7 | 7F0A | 5D6 | 000A | 000 | 000A | FFE1 | 1001 | |  |  |
| 5D7 | F002 | 5D8 | F002 | 5D7 | F002 | 000 | 05D7 | FFE1 | 1001 | |  |  |
| 5D8 | 1306 | 5D9 | 1306 | 5D8 | 1306 | 000 | 05D8 | FFE1 | 1001 | |  |  |
| 5D9 | CEEE | 5C8 | CEEE | 5D9 | 05C8 | 000 | FFEE | FFE1 | 1001 | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В | А | З | А |
| КОИ-8 | F7 | E1 | FA | E1 |
| UTF-16 | 412 | 410 | 417 | 410 |
| UTF-8 | D092 | D090 | D097 | D090 |

**ДОП ЗАДАНИЕ**

ORG 0x5C6

ELEMENT: WORD $ARRAY

NUMBER: WORD $NUM

ADDR: WORD $NUM

START: CLA

LD #0x0B

ST ZAGL

GETNUM: IN 5

AND #0x40

BEQ GETNUM

IN 4

ST (ELEMENT)

CMP #0x0A

BEQ BREAK

JUMP ONE

TODRAW:

LD -(NUMBER)

LD -(NUMBER)

JUMP DRAW

AGAIN:

LD -(NUMBER)

LD -(NUMBER)

LD -(NUMBER)

LD -(NUMBER)

JUMP GETNUM

THREE: LD (ELEMENT)+

AND #0x07

ST (NUMBER)

JUMP TODRAW

TWO: LD (ELEMENT)

ROR

ROR

ROR

AND #0x07

ST (NUMBER)+

JUMP THREE

ONE: LD (ELEMENT)

ROR

ROR

ROR

ROR

ROR

ROR

AND #0x03

ST (NUMBER)+

JUMP TWO

DRAW:

LD (NUMBER)+

CMP #0x0B

BEQ AGAIN

JUMP DRAWELEMENT

BREAK: HLT

DRAWELEMENT:

CMP #0x0

BEQ DRAW0

CMP #0x1

BEQ DRAW1

CMP #0x2

BEQ DRAW2

CMP #0x3

BEQ DRAW3

CMP #0x4

BEQ DRAW4

CMP #0x5

BEQ DRAW5

CMP #0x6

BEQ DRAW6

CMP #0x7

BEQ DRAW7

DRAW0:

LD #0xFF

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x81

OUT 0x10

LD #0x81

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW1:

LD #0xFF

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW2:

LD #0x8F

OUT 0x10

LD #0x89

OUT 0x10

OUT 0x10

LD #0xF9

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW3:

LD #0x89

OUT 0x10

OUT 0x10

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW4:

LD #0xF8

OUT 0x10

LD #0x08

OUT 0x10

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW5:

LD #0xF9

OUT 0x10

LD #0x89

OUT 0x10

OUT 0x10

LD #0x8F

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW6:

LD #0xFF

OUT 0x10

LD #0x89

OUT 0x10

OUT 0x10

LD #0x8F

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

DRAW7:

LD #0x80

OUT 0x10

OUT 0x10

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x0

OUT 0x10

OUT 0x10

JUMP DRAW

ORG 0x607

ARRAY: WORD ?

ORG 0x5B0

NUM: WORD ?

ORG 0x5B3

ZAGL: WORD ?

Вывод: В ходе выполнения лабораторной работы я познакомился с  режимами передачи данных, различными ВУ ввода-вывода, изучил команды ввода-вывода, написал программу на Ассемблере.