Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №5

По дисциплине

«Основы Профессиональной Деятельности»

Вариант № 18987

Выполнил студент группы P3118:

Богданова Мария Михайловна

Преподаватель:

          Осипов Святослав Владимирович

Санкт-Петербург

2023 г.

**Оглавление**

[Задание 2](#_Toc87370215)

[Ход работы 3](#_Toc87370216)

[Текст исходной программы 3](#_Toc87370217)

[Описание программы 3](#_Toc87370218)

[Трассировка 4](#_Toc87370219)

[Вариант программы с меньшим количеством команд 4](#_Toc87370220)

[Трассировка с данными числами 5](#_Toc87370221)

[Вывод 5](#_Toc87370222)

## **Задание**

По выданному преподавателем варианту разработать программу асинхронного обмена данными с внешним устройством. При помощи программы осуществить ввод или вывод информации, используя в качестве подтверждения данных сигнал (кнопку) готовности ВУ.

1. Программа осуществляет асинхронный ввод данных с ВУ-2
2. Программа начинается с адреса 5A816. Размещаемая строка находится по адресу 5C416.
3. Строка должна быть представлена в кодировке Windows-1251.
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.

## **Ход работы**

|  |  |
| --- | --- |
| ORG 0x5A8 |  |
| RES: word 0x5C4 | Адрес текущей ячейки сохранения |
| STOPSYM: word 0x000A | Символ окончания ввода |
| RIGHT\_MASK: word 0x00FF | Маска для выявления младших байтов |
| START: CLA |  |
| S1: IN 5 | Ожидание ввода первого символа (Готов?) Проверка готовности ВУ-2  (SR == 0 -> переход) |
| AND #0x40 |
| BEQ S1 |
| IN 4 | Ввод первого символа  Сохранение |
| ST (RES) |
| CMP STOPSYM | Проверка на стоп-символ |
| BEQ FINISH |
| SWAB | Перемещение 1-го символа в ст. часть. Подготовка к вводу второго символа |
| ST (RES) |
| CLA |
| S2: IN 5 | Ожидание ввода второго символа  Проверка готовности ВУ-2  (Не готов -> переход) |
| AND #0x40 |
| BEQ S2 |
| LD (RES) | Сохранить 2 символа в текущую ячейку сохр.  Ввод второго символа в младшие 8 разрядов  Сохранение |
| IN 4 |
| ST (RES) |
| AND RIGHT\_MASK | Проверка на стоп-символ |
| CMP STOPSYM |
| BEQ FINISH |
| LD (RES)+ | Перейти к следующей ячейке сохранения |
|  |
| JUMP START | Возврат в начало цикла |
| FINISH: HLT | Остановка |

### **Описание программы**

**Расположение в памяти БЭВМ программы, исходных данных и результатов:**

5A9 – 5AA - Переменные

5AB – 5C1 – Программа

5C4+ – результат

**Адреса первой и последней выполняемой команды:**

Начало работы – 5AB  
Остановка программы – 5С1

### Трассировка с данными числами

Ы@k

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адр | Знач | IP | CR | AR | DR | SP | BR | AC | NZVC | Адр | Знач |
| 5AB | 0200 | 5AC | 0200 | 5AB | 0200 | 000 | 05AB | 0000 | 0100 |  |  |
| 5AC | 1205 | 5AD | 1205 | 5AC | 1205 | 000 | 05AC | 0040 | 0100 |  |  |
| 5AD | 2F40 | 5AE | 2F40 | 5AD | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 5AE | F0FD | 5AF | F0FD | 5AE | F0FD | 000 | 05AE | 0040 | 0000 |  |  |
| 5AF | 1204 | 5B0 | 1204 | 5AF | 1204 | 000 | 05AF | 00DB | 0000 |  |  |
| 5B0 | E8F7 | 5B1 | E8F7 | 5C4 | 00DB | 000 | FFF7 | 00DB | 0000 | 5C4 | 00DB |
| 5B1 | 7EF7 | 5B2 | 7EF7 | 5A9 | 000A | 000 | FFF7 | 00DB | 0001 |  |  |
| 5B2 | F00E | 5B3 | F00E | 5B2 | F00E | 000 | 05B2 | 00DB | 0001 |  |  |
| 5B3 | 0680 | 5B4 | 0680 | 5B3 | 0680 | 000 | 05B3 | DB00 | 1001 |  |  |
| 5B4 | E8F3 | 5B5 | E8F3 | 5C4 | DB00 | 000 | FFF3 | DB00 | 1001 | 5C4 | DB00 |
| 5B5 | 0200 | 5B6 | 0200 | 5B5 | 0200 | 000 | 05B5 | 0000 | 0101 |  |  |
| 5B6 | 1205 | 5B7 | 1205 | 5B6 | 1205 | 000 | 05B6 | 0040 | 0101 |  |  |
| 5B7 | 2F40 | 5B8 | 2F40 | 5B7 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 5B8 | F0FD | 5B9 | F0FD | 5B8 | F0FD | 000 | 05B8 | 0040 | 0001 |  |  |
| 5B9 | A8EE | 5BA | A8EE | 5C4 | DB00 | 000 | FFEE | DB00 | 1001 |  |  |
| 5BA | 1204 | 5BB | 1204 | 5BA | 1204 | 000 | 05BA | DB40 | 1001 |  |  |
| 5BB | E8EC | 5BC | E8EC | 5C4 | DB40 | 000 | FFEC | DB40 | 1001 | 5C4 | DB40 |
| 5BC | 2EED | 5BD | 2EED | 5AA | 00FF | 000 | FFED | 0040 | 0001 |  |  |
| 5BD | 7EEB | 5BE | 7EEB | 5A9 | 000A | 000 | FFEB | 0040 | 0001 |  |  |
| 5BE | F002 | 5BF | F002 | 5BE | F002 | 000 | 05BE | 0040 | 0001 |  |  |
| 5BF | AAE8 | 5C0 | AAE8 | 5C4 | DB40 | 000 | FFE8 | DB40 | 1001 | 5A8 | 05C5 |
| 5C0 | CEEA | 5AB | CEEA | 5C0 | 05AB | 000 | FFEA | DB40 | 1001 |  |  |
| 5AB | 0200 | 5AC | 0200 | 5AB | 0200 | 000 | 05AB | 0000 | 0101 |  |  |
| 5AC | 1205 | 5AD | 1205 | 5AC | 1205 | 000 | 05AC | 0040 | 0101 |  |  |
| 5AD | 2F40 | 5AE | 2F40 | 5AD | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 5AE | F0FD | 5AF | F0FD | 5AE | F0FD | 000 | 05AE | 0040 | 0001 |  |  |
| 5AF | 1204 | 5B0 | 1204 | 5AF | 1204 | 000 | 05AF | 006B | 0001 |  |  |
| 5B0 | E8F7 | 5B1 | E8F7 | 5C5 | 006B | 000 | FFF7 | 006B | 0001 | 5C5 | 006B |
| 5B1 | 7EF7 | 5B2 | 7EF7 | 5A9 | 000A | 000 | FFF7 | 006B | 0001 |  |  |
| 5B2 | F00E | 5B3 | F00E | 5B2 | F00E | 000 | 05B2 | 006B | 0001 |  |  |
| 5B3 | 0680 | 5B4 | 0680 | 5B3 | 0680 | 000 | 05B3 | 6B00 | 0001 |  |  |
| 5B4 | E8F3 | 5B5 | E8F3 | 5C5 | 6B00 | 000 | FFF3 | 6B00 | 0001 | 5C5 | 6B00 |
| 5B5 | 0200 | 5B6 | 0200 | 5B5 | 0200 | 000 | 05B5 | 0000 | 0101 |  |  |
| 5B6 | 1205 | 5B7 | 1205 | 5B6 | 1205 | 000 | 05B6 | 0040 | 0101 |  |  |
| 5B7 | 2F40 | 5B8 | 2F40 | 5B7 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 5B8 | F0FD | 5B9 | F0FD | 5B8 | F0FD | 000 | 05B8 | 0040 | 0001 |  |  |
| 5B9 | A8EE | 5BA | A8EE | 5C5 | 6B00 | 000 | FFEE | 6B00 | 0001 |  |  |
| 5BA | 1204 | 5BB | 1204 | 5BA | 1204 | 000 | 05BA | 6B0A | 0001 |  |  |
| 5BB | E8EC | 5BC | E8EC | 5C5 | 6B0A | 000 | FFEC | 6B0A | 0001 | 5C5 | 6B0A |
| 5BC | 2EED | 5BD | 2EED | 5AA | 00FF | 000 | FFED | 000A | 0001 |  |  |
| 5BD | 7EEB | 5BE | 7EEB | 5A9 | 000A | 000 | FFEB | 000A | 0101 |  |  |
| 5BE | F002 | 5C1 | F002 | 5BE | F002 | 000 | 0002 | 000A | 0101 |  |  |
| 5C1 | 0100 | 5C2 | 0100 | 5C1 | 0100 | 000 | 05C1 | 000A | 0101 |  |  |
| 5C2 | 0000 | 5C3 | 0000 | 5C2 | 0000 | 000 | 05C2 | 000A | 0101 |  |  |
| 5C3 | 0000 | 5C4 | 0000 | 5C3 | 0000 | 000 | 05C3 | 000A | 0101 |  |  |
| 5C4 | DB40 | 7FF | DB40 | 7FF | 05C5 | 7FF | FFFF | 000A | 0101 | 605 | FFFF |
|  |  |  |  |  |  |  |  |  |  | 7FF | 05C5 |
| 7FF | 05C5 | 000 | 05C5 | 7FF | 05C5 | 7FF | 07FF | 0005 | 0000 |  |  |

## Вывод

В ходе выполнения лабораторной работы я изучила устройство ввода и вывода в БЭВМ и работу БЭВМ с устройствами по сигналам готовности.

16815

ORG 0x5A8

RES: WORD 0x5C4

## Список литературы

1. В.В Кириллов, А.А Приблуда, С. В. Клименков, Д.Б. Афанасьев “Методические указания к лабораторным по курсу Основы профессиональной деятельности ” <https://se.ifmo.ru/documents/10180/38002/Методические+указания+к+выполнению+лабораторных+работ+и+рубежного+контроля+БЭВМ+2019+bcomp-ng.pdf/d5a1be02-ad3f-4c43-8032-a2a04d6db12e>

ORG 0x5A8

RES: WORD 0x5C4

STOPSYM: WORD 0x000A

RIGHT\_MASK: WORD 0x00FF

START: CLA

S1: IN 5

AND #0x40

BEQ S1

IN 4

ST (RES)

CMP STOPSYM

BEQ FINISH

SWAB

ST (RES)

CLA

S2: IN 5

AND #0x40

BEQ S2

LD (RES)

IN 4

ST (RES)

AND RIGHT\_MASK

CMP STOPSYM

BEQ FINISH

LD (RES)+

JUMP START

FINISH: HLT  
  
**Доп. Задание:**  
Программа должна считывать ключ ВУ-2, затем записывать первый и второй символ последовательно и выполнять XOR с ключом, записывая результат.  
Сначала вводим 8 битный ключ, затем вводим 8 битные слова, которые XORрим с ключом и сохраняем в ячейки (слово1 слово2).

ORG 0x200

RES: WORD 0x400

STOPSYM: word 0x030

KEY: WORD 0x00

LOCATION: WORD 0x400

TEMP: WORD 0x0000

MASK: WORD 0x00FF

B: WORD 0x00C0

START: CLA

RKEY: IN 0x019

AND #0x40

BEQ RKEY

IN 0x018

AND MASK

ST KEY

S1: CLA

IN 0x019

AND #0x40

BEQ S1

IN 0x018

ST (RES)+

CMP STOPSYM

BEQ O1

JUMP S1

O1:

IN 0xD

AND #0x40

BEQ O1

LD (LOCATION)+

CMP STOPSYM

BEQ FINISH

SUB B

PUSH

NOT

AND KEY

PUSH

LD KEY

NOT

AND &1

OR &0

SWAP

POP

SWAP

POP

OUT 0xC

JUMP O1

FINISH: HLT