Университет ИТМО

Факультет ФПИ и КТ

**Лабораторная работа №5**

По Основам профессиональной деятельности

Вариант 1754

Выполнил: Кочнев Р.Д.

Группа: Р3117

Преподаватель: Ткешелашвили Н.М.

Санкт-Петербург

2022

Оглавление

[Задание 2](#_Toc101225793)

[Ход работы 2](#_Toc101225794)

[Текст исходной программы 2](#_Toc101225795)

[Описание программы 3](#_Toc101225796)

[Доп задание 4](#_Toc101225797)

[Трассировка с данными числами 6](#_Toc101225798)

[Вывод 6](#_Toc101225799)

## 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 вывод данных на ВУ-3
2. Программа начинается с адреса 3D816. Размещаемая строка находится по адресу 62E16.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Вывод строки начинается со вывода количества символов (1 байт), и должен быть завершен по выводу их необходимого количества.

## Ход работы

### Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| Метка | Мнемоника | Параметр | Описание |
|  | ORG | 0x62E | Начало массива |
|  | WORD | 0x8421 | Элементы массива |
|  | WORD | 0x1248 |
|  | WORD | 0x6336 |
|  | ORG | 0x3D8 |  |
| CUR: | WORD | 0x62E | Адрес текущего элемента в массиве |
| COUNT: | WORD | 0x0002 | Количество элементов в массиве |
| ZERO: | WORD | 0x0000 | Константа 0 |
| LITTLE: | WORD | 0x0000 | Младший байт текущего элемента |
| BIG: | WORD | 0x0000 | Старший байт текущего элемента |
| MASK: | WORD | 0x00FF | Маска для выделения младшего байта |
| START: | CLA |  | Начало программы |
| S0 | IN | 7 | Вывод количество символов |
|  | AND | #0x40 |
|  | BEQ | S0 |
|  | LD | COUNT |
| START1 | OUT | 6 | Начало вывода символов |
|  | LD | (CUR)+ | Загрузка в Little младшего байта |
|  | ST | LITTLE |
|  | SWAB |  | Загрузка в Big старшего байта |
|  | ST | BIG |
| S1: | IN | 7 | Ожидание готовности ВУ к выводу старшего байта |
|  | AND | #0x40 |
|  | BEQ | S1 |
|  | LD | BIG | Вывод старшего байта |
|  | OUT | 6 |
|  | LD | COUNT | Уменьшение количество элементов на 1 |
|  | DEC |  |
|  | ST | COUNT |
|  | AND | MASK | Проверка, что остались элементы |
|  | CMP | ZERO |
|  | BEQ | FINISH |
| S2: | IN | 7 | Ожидание готовности ВУ к выводу младшего байта |
|  | AND | #0x40 |
|  | BEQ | S2 |
|  | LD | LITTLE | Вывод младшего байта |
|  | OUT | 6 |
|  | JUMP | START | Переход к новому элементу массива |
| FINISH: | HLT |  | Остановка |

### Описание программы

**Назначение программы: Вывод текста, сохранённого в массиве в формате:**

АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП\_СИМВ

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

Ячейки 0x3В8-0x3FB – код программы

Ячейка CUR 0x3D8 – адрес первого символа строки

Ячейка LITTLE 0x3DB – переменная (значение младшего байта)

Ячейка BIG 0x3DC – переменная (значение старшего байта)

Ячейка MASK 0x3DD – битовая маска для выделения младшего байта

Ячейки 0x62E−(0x55B + COUNT) – выводимая строка, если

**Область представления:**

Ячейка 0x3D8 – 11-ти разрядное целое беззнаковое число с фиксированной запятой.

Ячейка 0x3D9 - 0x3DC – символ строки в кодировке КОИ-8

Ячейка 0x3DD – набор из 16 логических значений

Ячейки 0x62E−(0x55B + COUNT): два символа строки в кодировке КОИ-8

**Область допустимых значений**

Все ячейки кроме CUR 0x3D8, ZERO 0x000 и константы MASK 0x2FC могут иметь любые значения. Ячейка 0х3D8 хранит адрес первого элемента массива, значит массив должен лежать [0x000, 0𝑥3D7] и [0x3FC, 0х7𝐹𝐹].

Т.е. если L16 – это длинна строки:

### Доп задание

|  |  |  |  |
| --- | --- | --- | --- |
|  | ORG | 0x000 |  |
| ZERO: | WORD | 0x0000 | Константа 0 |
| LITTLE: | WORD | 0x0000 | Младший байт текущего элемента |
| BIG: | WORD | 0x0000 | Старший байт текущего элемента |
| MASK: | WORD | 0x00FF | Маска для выделения младшего байта |
| BMASK: | WORD | 0xFF00 | Маска для выделения старшего байта |
| STOPSIGN: | WORD | 0x2E; | Символ остановки |
| BS: | WORD | 0x8 | Символ backspace |
|  |  |  |  |
| START: | CLA |  | Очистка принтера |
|  | OUT | 0xC |
| S1: | IN | 0x19; | Ожидание ВУ к вводу старшего байта |
|  | AND | #0x40; |
|  | BEQ | S1; |
|  | IN | 0x18; | Ввод символа |
|  | AND | MASK; | Проверка на стоп символ и переход на вывод данных |
|  | CMP | STOPSIGN; |
|  | BEQ | FINISH2; |
|  | CMP | BS; | Проверка на backspace и удаление последнего символа |
|  | BEQ | FRSTBS; |
|  | SWAB; |  | Переворот строки формата 00FF в FF00 и сохранение |
|  | ST | BIG; |
|  |  |  |  |
| S2: | IN | 0x19; | Проверка ВУ на готовность к вводу старшего байта |
|  | AND | #0x40; |
|  | BEQ | S2; |
|  | IN | 0x18; | Ввод символа |
|  | AND | MASK; | Проверка на стоп символ |
|  | CMP | STOPSIGN; |
|  | BEQ | FINISH1; |
|  | CMP | BS; | Проверка на backspace и удаление последнего символа |
|  | BEQ | S1; |
|  | ADD | BIG; | Сохранение двух символов в стек |
|  | PUSH |  |
|  | JUMP | S1; | Переход к началу ввода |
|  |  |  |  |
| FRSTBS: | POP; |  | Вызов, если backspace первый байт |
|  | PUSH |  |  |
|  | AND | MASK; | Проверка на первый символ в стеке и игнор его |
|  | CMP | LITTLE |
|  | BEQ | S1; |
|  | POP |  |  |
|  | AND | BMASK | Второй символ в строке удаление ячейки из стека |
|  | ST | BIG |
|  | JUMP | S2 |
|  |  |  |  |
| FINISH1: | ADD | BIG; | Если стоп символ второй байт в строке |
| FINISH2: | PUSH |  | Если стоп символ первый байт в строке |
|  |  |  |  |
| PRINTSTART: | CLA; |  | Начало вывода |
|  |  |  |  |
| BB: | IN | 0xD; | Ожидание готовности принтера |
|  | AND | #0x40; |
|  | BEQ | BB; |
|  |  |  |  |
| PRINT: | POP; |  |  |
|  | PUSH |  |  |
|  | AND | MASK; | Проверка на конец стека |
|  | CMP | LITTLE |
|  | BEQ | FINISH; |
|  | POP |  | Вывод младшего байта строки |
|  | OUT | 0xC |
|  | PUSH |  |
|  |  |  |  |
| BC: | IN | 0xD; | Ожидание готовности принтера |
|  | AND | #0x40; |
|  | BEQ | BC; |
|  | POP |  | Не пытайтесь понять этот костыль. Если много баловаться с стеркой, то стоп символ не будет выведен. Это все ради него |
|  | PUSH |  |
|  | SWAB |  |
|  | AND | MASK; |
|  | CMP | ZERO |
|  | BEQ | AB |
|  |  |  |  |
|  | OUT | 0xC | Вывод старшего байта |
|  | POP |  |
|  | JUMP | BB |
|  |  |  |  |
| AB: | POP |  | Как правильно выбрать костыли, как подобрать костыли подмышечные и с опорой  на руки |
|  | CLA |  |
|  | JUMP | BB |
|  |  |  |  |
| FINISH: | HLT |  | Конец :) |

### Трассировка с данными числами

## Вывод

В процессе выполнения лабораторной работы был получен опыт работы с подпрограммами и стеком, разобрался, каким образом реализован стек в БЭВМ, изучил принцип действия команд PUSH, POP, CALL, RET