**Национальный исследовательский университет информационных технологий, механики и оптики**

**Факультет Программной Инженерии и Компьютерной Техники**



**Вариант № 1152**

**Лабораторная работа №5**

**По дисциплине:**

**«Основы профессиональной деятельности»**

Работу выполнила:

Студентка группы P3112

Сенина Мария Михайловна

Преподаватель:

Перминов Илья Валентинович

Санкт-Петербург

2021

**ТЕКСТ ЗАДАНИЯ**

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-1
2. Программа начинается с адреса. Размещаемая строка находится по адресу .
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП\_СИМВ.
5. Ввод или вывод строки должен быть завершён 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.

**Текст программы**

|  |  |  |  |
| --- | --- | --- | --- |
| Метка | Мнемоника | Параметр | Описание |
|  | ORG | 0x061 |  |
| CUR: | WORD | 0x590 | Текущий элемент в массиве |
| STOPW: | WORD | 000A | Стоп-слово |
| LITTLE: | WORD | 0000 | Младший бит текущего элемента |
| BIG: | WORD | 0000 | Старший бит текущего элемента |
| START: | CLA |  | Начало программы |
|  | LD | (CUR)+ | Загрузка в Little младшего байта |
|  | ST | LITTLE |
|  | SWAB |  | Загрузка в Big старшего байта |
|  | ST | BIG |
| S1: | IN | 3 | Ожидание готовности ВУ к выводу младшего байта |
|  | AND | #0x40 |
|  | BEQ | S1 |
|  | LD | LITTLE | Проверка, что младший байт не стоп-слово |
|  | SXTB |  |
|  | CMP | STOPW |
|  | BEQ | FINISH |
|  | OUT | 2 | Вывод младшего байта |
| S2: | IN | 3 | Ожидание готовности ВУ к выводу старшего байта |
|  | AND | #0x040 |
|  | BEQ | S2 |
|  | LD | BIG | Проверка, что старший байт не стоп-слово |
|  | SXTB |  |
|  | CMP | STOPW |
|  | BEQ | FINISH |
|  | OUT | 2 | Вывод старшего байта |
|  | JUMP | START | Переход к новому элементу массива |
| FINISH: | HLT |  | Останов |

**Назначение программы и реализуемые ею функции (формулы)**

Вывод текста сохранённого в массиве в формате АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП\_СИМВ

**Область представления и область допустимых значений исходных данных и результата**

**Область Представления**

Ячейка 0x61 – 11-ти разрядное целое беззнаковое числа с фиксированной запятой.

Ячейка 0x62 - 0x64 – символ строки в кодировке КОИ-8

Ячейки 0x590−(0x590 + – 1): два символа строки в кодировке КОИ-8

**Область Допустимых Значений**

Все ячейки кроме 0x61 могут иметь любые значения. Ячейка 0х61 хранит адрес первого элемента массива, значит массив должен лежать .

Т.е. если – это длинна строки: =>

**Расположение в памяти ЭВМ программы, исходных данных и результатов**

Ячейки 0x61-0x7B – код программы

Ячейка CUR 0x61 – адрес первого символа строки

Ячейка STOPW 0x62 – код символа окончания строки

Ячейка LITTLE 0x63 – переменная (значение младшего байта)

Ячейка BIG 0x64 – переменная (значение старшего байта)

Ячейки 0x590−(0x590 + – 1) – выводимая строка, если – её длина в шестнадцатеричной системе счисления.

**Адреса первой и последней выполняемых процессором команд**

Первая команда: 0x65 последняя команда: 0x7B

**Вывод**

В этой лабораторной я познакомилась с тем, как подключаются к БЭВМ внешние устройства и тем, как вводить и выводить данные с их помощью.

**Выданная строка**

|  |  |  |  |
| --- | --- | --- | --- |
| Символ | КОИ-8 | UTF-8 | UTF-16 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Первое слово: JJHH

**Трассировка с выданными строками**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AC | BR | NZVC | IR | DR | CR | IP | SP | AR | АДРЕС | Новое значение |
| 0000 | 0065 | 0100 | 065 | 0200 | 0200 | 066 | 0000 | 065 |  |  |
| JJHH | FFFA |  | 065 | JJHH | AAFA | 067 | 0000 | 590 |  |  |
| JJHH | FFFB |  | 065 | JJHH | EEFB | 068 | 0000 | 063 | 063 | JJHH |
| HHJJ | 0068 |  | 065 | 0680 | 0680 | 069 | 0000 | 068 |  |  |
| HHJJ | FFFA |  | 065 | HHJJ | EEFA | 06A | 0000 | 064 | 064 | HHJJ |
| HH40 | 006A |  | 065 | 1203 | 1203 | 06B | 0000 | 06A |  |  |
| 0040 | 0040 | 0000 | 065 | 0040 | 2F40 | 06C | 0000 | 06B |  |  |
| 0040 | FFFD | 0000 | 065 | F0FD | F0FD | 06D | 0000 | 06C |  |  |
| JJHH | FFF5 |  | 065 | JJHH | AEF5 | 06E | 0000 | 063 |  |  |
| 00HH | 006E |  | 065 | 0600 | 0600 | 06F | 0000 | 06E |  |  |
| 00HH | FFF2 |  | 065 | 000A | 7EF2 | 070 | 0000 | 062 |  |  |
| 00HH | 0009 |  | 065 | F009 | F009 | 071 | 0000 | 070 |  |  |
| 00HH | 0071 |  | 065 | 1302 | 1302 | 072 | 0000 | 071 |  |  |
| 0040 | 0072 |  | 065 | 1203 | 1203 | 073 | 0000 | 072 |  |  |
| 0040 | 0040 | 0000 | 065 | 0040 | 2F40 | 074 | 0000 | 073 |  |  |
| 0040 | 0074 | 0000 | 065 | F0F | F0FD | 075 | 0000 | 074 |  |  |
| HHJJ | FFEE |  | 065 | HHJJ | AEEE | 076 | 0000 | 064 |  |  |
| 00JJ | 0073 |  | 065 | 0600 | 0600 | 077 | 0000 | 076 |  |  |
| 00JJ | FFEA |  | 065 | 000A | 7EEA | 078 | 0000 | 077 |  |  |
| 00JJ | 0078 |  | 065 | F002 | F002 | 079 | 0000 | 078 |  |  |
| 00JJ | 0079 |  | 065 | 1302 | 1302 | 07A | 0000 | 079 |  |  |
| 00JJ | FFEA |  | 065 | 0065 | CEEA | 07B | 0000 | 07A |  |  |