

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 5

по дисциплине
‘Основы профессиональной деятельности’

Вариант № 8611

Выполнил:

Студент группы Р3115

Собитов Анвархон А.

Преподаватель:

Абузов Ярослав А.



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2023

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 ввод данных с ВУ-3
2. Программа начинается с адреса 467_{16} . Размещаемая строка находится по адресу 590_{16} .
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 завершен по символу с кодом 00 (NUL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

Код на Assembler

```
ORG 0x467 ; Адрес начала программы
res: WORD 0x590 ; Ссылка на результат
finish: WORD 0x00 ; Стоп-символ
temp: WORD ? ; Ячейка для записи нечетных символов

START: CLA ; Очистка аккумулятора
s1: IN 7 ; Ожидание ввода нечетного символа
    AND #0x40 ; Проверка на наличие введенного символа
    BEQ s1 ; Нет - "Спин-луп"
    IN 6 ; Вывод байта в АС
    ST (res) ; Сохраняем символ в результат
    ST temp ; Сохраняем символ во временную переменную
    CMP finish ; Проверяем на стоп-символ
    BEQ exit ; Если стоп-символ - выход
    CLA ; Очистка аккумулятора
s2: IN 7 ; Ожидание ввода четного символа
    AND #0x40 ; Проверка на наличие введенного символа
    BEQ s2 ; Нет - "Спин-луп"
    IN 6 ; Вывод байта в АС
    SWAB ; Перемещаем четный символ в старший байт
    OR temp ; Совмещаем с 1-м символом
    ST (res) ; Сохраняем в память по ссылке
    SUB temp ; Вычитаем 1-й символ
    SWAB ; Перемещаем четный символ в младший байт
    CMP finish ; Проверяем на стоп-символ
    BEQ exit ; Если стоп-символ - выход
    LD (res)+ ; Инкрементируем ссылку на результат
    CLA ; Очистка аккумулятора
    JUMP s1 ; Возвращаемся в начало цикла

exit: HLT ; Остановка программы
```

Область представления

- res – 11-разрядная ячейка со ссылкой на результат.
- finish – 16-разрядная константа.
- temp – 16-разрядная ячейка для временного хранения введенных символов.
- 590- ? – 16-разрядные ячейки, хранящие в себе по два символа в кодировке Windows-1251.

Расположение данных в памяти

- 46A-483 – команды;
- 467, 468, 469– исходные данные;
- 590 - ? – результат.

Адреса первой и последней выполняемой команды

- Адрес первой команды: 46A
- Адрес последней команды: 483

Область допустимых значений

- res (указатель на ячейки массива, хранящий результат ввода) $\in [562; 2047]$
- temp (ячейка для записи нечетных символов) $\in [0; 255]$, т.к. в нее записывается только 1 символ из 8 бит.
- Введенный символ: $[00; FF]$

Адрес первого элемента массива равен 590 по условию. Т.к. $2047 - 590 = 1457$ – кол-во ячеек, которые могут использоваться для записи результата $\Rightarrow 1457 * 2 = 2914$ – максимально возможное кол-во введенных символов (т.к. в данной кодировке символ занимает 1 байт), включая обязательный стоп-символ \Rightarrow Кол-во введенных символов $\in [1; 2970]$.

Вывод

При выполнении данной лабораторной работы я познакомился с асинхронным вводом-выводом данных в БЭВМ, узнал о внешних устройствах, их регистрах и принципах работы. Также, я познакомился с представлением данных в различных кодировках и попрактиковался с вводом данных на ВУ-3.

