

*Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №5 по дисциплине «Основы профессиональной
деятельности»
Вариант 10035

Выполнил:

Студент группы Р3112

Медведев Ярослав Александрович

Преподаватель:

Блохина Елена Николаевна

г. Санкт-Петербург
2024

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.

Введите номер варианта

1. Программа осуществляет асинхронный ввод данных с ВУ-2
2. Программа начинается с адреса 3A1₁₆. Размещаемая строка находится по адресу 60E₁₆.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... 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.

Слово: Йогурт:

ISO-8859-5: B9, DE, D3, E3, E0, E2, 3A

UTF-16: 0419, 043E, 0433, 0443, 0440, 0442, 003A

UTF-8: D099, D0BE, D0B3, D183, D180, D182, 003A

Доп:

Ввод осуществляется с клавиатуры (ВУ-8)

вывод осуществляется на текстовый принтер

Выводятся только нечетные символы

все остальное согласно первоначальному заданию

XXXX XXXX XXXX XXXX

СТОП-символ (если это будет необходимо) можно поменять (стоп-символ - /)

Текст подпрограммы:

```
ORG 0x3A1 ;
I: WORD 0x060E ;
RES1: WORD 0x0000 ;
RES2: WORD 0x0000 ;
MASK: WORD 0xFF2F;
START: CLA ;
S1: IN 0x19;
AND #0x40 ;
BEQ S1 ;
CLA ;
IN 0x18 ;
ST RES1;
AND MASK;
CMP RES1;
BEQ EXIT;
LD RES1;
PRINT: IN 0x0D ;
AND #0x40;
BEQ PRINT;
LD RES1;
OUT 0x0C;
```

```
S2:  IN 0x19;
      AND #0x40 ;
      BEQ S2 ;
      CLA ;
      IN 0x18 ;
      ST RES2;
      LD RES1;
      SWAB ;
      OR RES2;
      ST RES1;
      AND MASK;
      CMP RES1;
      BEQ EXIT;
      LD RES1;
      ST (I)+;
      JUMP START;
EXIT: LD RES1;
      ST (I)+;
      HLT ;
```

Ход работы

Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарии
3A5	0200	CLA	0 -> AC
3A6	1205	IN 5	5 -> МЛ. БАЙТ AC
3A7	2F40	AND #40	40 & AC -> AC
3A8	F0FE	BEQ (IP - 2)	Переход если равенство (Z==1)
3A9	0200	CLA	0 -> AC
3AA	1204	IN 4	4 -> МЛ. БАЙТ AC
3AB	EEF7	ST (IP - 9)	AC -> (IP - 9)
3AC	2EF8	AND (IP - 8)	(IP - 8) & AC -> AC
3AD	7EF5	CMP (IP - 11)	УСТАНОВИТЬ ФЛАГИ ПО РЕЗУЛЬТАТУ AC - (IP - 11)
3AE	F011	BEQ (IP + 17)	Переход если равенство (Z==1)
3AF	1205	IN 5	5 -> МЛ. БАЙТ AC
3B0	2F40	AND #40	40 & AC -> AC
3B1	F0FE	BEQ (IP - 2)	Переход если равенство (Z==1)
3B2	0200	CLA	0 -> AC
3B3	1204	IN 4	4 -> МЛ. БАЙТ AC
3B4	EEEE	ST (IP - 17)	AC -> (IP - 17)
3B5	AEED	LD (IP - 19)	(IP - 19) -> AC
3B6	0680	SWAB	AC0...AC7 <-> AC8...AC15
3B7	3EEE	OR (IP - 20)	(IP - 20) AC -> AC

3B8	EEEE	ST (IP - 22)	AC -> (IP - 22)
3B9	2EEB	AND (IP - 21)	(IP - 21) & AC -> AC
3BA	7EE8	CMP (IP - 24)	УСТАНОВИТЬ ФЛАГИ ПО РЕЗУЛЬТАТУ AC - (IP - 24)
3BB	F004	BEQ (IP + 4)	Переход если равенство (Z==1)
3BC	AEF6	LD (IP - 26)	(IP - 26) -> AC
3BD	EAE4	ST (IP - 28)+	AC -> (IP - 28)+
3BE	CEE9	JUMP (IP - 25)	(IP - 25) -> IP
3BF	AEE3	LD (IP - 29)	(IP - 29) -> AC
3C0	EAF1	ST (IP - 31)+	AC -> (IP - 31)+
3C1	0100	HLT	ОСТАНОВ

Текст программы на ассемблере

```

ORG 0x3A1 ;
I: WORD 0x060E ; Ссылка на ячейку для записи информации
RES1: WORD 0x0000 ; Первый введенный символ/конечный результат
RES2: WORD 0x0000 ; Второй введенный символ
MASK: WORD 0xFF00; Маска для проверки на стоп-символ
START: CLA ; Очистка аккумулятора
S1: IN 5 ; Получение данных из регистра состояния ВУ-2
    AND #0x40; Проверка на наличие введенного символа
    BEQ S1 ; Нет - "Спин-луп"
    CLA ; Очистка аккумулятора
    IN 4 ; Получение данных из дата регистра ВУ-2
    ST RES1; Сохранение первой буквы в RES1
    AND MASK; Применение маски
    CMP RES1; Проверка на стоп-символ
    BEQ EXIT; Да - переход к завершению программы
S2: IN 5 ; Получение данных из регистра состояния ВУ-2
    AND #0x40 ; Проверка на наличие введенного символа
    BEQ S2 ; Нет - "Спин-луп"
    CLA ; Очистка аккумулятора
    IN 4 ; Получение данных из дата регистра ВУ-2
    ST RES2; Сохранение второй буквы в RES2
    LD RES1; Загрузка первой буквы
    SWAB ;
    OR RES2; Подготовка данных для записи
    ST RES1; Сохранение конечного результата в RES1

```

AND MASK; Применение маски
 CMP RES1; Проверка на стоп-символ
 BEQ EXIT; Да - переход к завершению программы
 LD RES1;
 ST (I)+; Загрузка в память конечного результата
 JUMP START; Переход к началу программы
 EXIT: LD RES1;
 ST (I)+; Сохранение в память конечного результата
 HLT ; Прекращение работы программы

Описание программы

Программа осуществляет асинхронный ввод данных с ВУ-2, предоставляя строку в кодировке ISO-8859-5 и записывая ее в память в формате: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП_СИМВ. Концом передачи информации служит символ 00.

Область представления данных

I - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$.
 RES1 - беззнаковое 16-разрядное число, диапазон - $[2^{15}, 2^{15} - 1]$.
 RES2 - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$.
 MASK - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$.

Область допустимых значений

I - [f60E; 07FF];
 RES1 - [0000, FFFF];
 RES2 - [00, FF];
 MASK - константа (FF00).

Расположение в памяти ЭВМ

Расположение программы - [3A5, 3BF].
 3A1 - исходная переменная I - адрес ячейки для записи символов.
 3A2 - переменная, хранящая первую букву и конечный результат.
 3A3 - переменная, хранящая вторую букву.
 3A4 - исходная константа, выполняющая роль маски.

Программа:

Адрес первой выполняемой команды - 3A5, адрес последней выполняемой команды - 3BF.

Трассировка

Выполняемая команда	Содержимое регистров процессора после выполнения команды	Ячейка, содержимое которой изменилось после выполнения команды

[illegible]

Вывод

В ходе выполнения лабораторной работы я познакомился с асинхронным вводом-выводом в БЭВМ, более подробно рассмотрел взаимодействие с ВУ-2 и научился писать программы ввода-вывода на языке Ассемблера БЭВМ.