

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по
Основам профессиональной деятельности №5
Вариант №2000

Работу выполнил:

Агаев Х. Р.

Группа:

P3234

Санкт-Петербург,

2023

СОДЕРЖАНИЕ

Стр.

1	Текст задания	3
2	Текст исходной программы	4
3	Описание программы	8
	3.1 Реализуемая функция	8
	3.2 Область допустимых значений	8
	3.3 Расположение программы в памяти	8
4	Трассировка программы	9
5	Дополнительное задание №1: ввод с калькулятора, вывод в текстовый принтер	12
	ЗАКЛЮЧЕНИЕ	16

1 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.

Задания, соответствующие варианту:

1. Программа осуществляет асинхронный ввод данных с ВУ-3.
2. Программа начинается с адреса $2BF_{16}$. Размещаемая строка находится по адресу $5FD_{16}$.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти АDR0 ДЛИНА АDR1 СИМВ1 СИМВ2 АDR2 СИМВ3 СИМВ4 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Ввод строки начинается со ввода количества символов (1 байт), и должен быть завершен по вводу их необходимого количества.

2 Текст исходной программы

Для выполнения данного задания была написана программа на языке ассемблера, выполняющая взаимодействие с ВУ-3, используя асинхронный обмен данными. Код данной программы и ее описание представлены ниже.

```
ORG 0x2BF
```

```
CURRENT_LEN: & WORD 0x0
```

```
PTR: WORD 0x5FD
```

```
START:
```

```
    CLA
```

```
GET_LEN:
```

```
    IN 7
```

```
    AND #0x40
```

```
    BEQ GET_LEN
```

```
    IN 6
```

```
    ST 0x5FD
```

```
    ST CURRENT_LEN
```

```
    CMP (PTR)+
```

```
    CLA
```

```
    ST (PTR)
```

```
CLEAR:
```

```
    CLA
```

```
ADD_SYMBOL:
```

```
    IN 7
```

```
    AND #0x40
```

```
    BEQ ADD_SYMBOL
```

```
    LD (PTR)
```

```
    BNE IS_SECOND
```

```
    IN 6
```

```
SWAB
ST (PTR)
JUMP NEXT
```

```
IS_SECOND:
    ST (PTR)
    CMP (PTR)+
    CLA
    ST (PTR)
```

```
NEXT:
    LOOP CURRENT_LEN
    JUMP CLEAR
    HLT
```

```
ORG 0x5FD
STRING: WORD 0
```

Адрес	Код команды	Мнемоника	Комментарий
2BF (CURRENT_LEN)	0000	-	Кол-во мест для символов в массиве
2C0 (PTR)	05FD	-	Указатель на текущий элемент массива
2C1 (START)	0200	CLA	0 -> AC
2C2 (GET_LEN)	1207	IN 7	Получение SR БУ-3
2C3	2F40	AND 0x40	Если SR = 1, то AC != 0
2C4	F0FD	BEQ GET_LEN	Если AC == 0, перемещение в GET_LEN
2C5	1206	IN 6	Получение DR из БУ-3
2C6	E5FD	ST 0x5FD	AC -> M(5FD) (сохранение длины)
2C7	EEF7	ST CURRENT_LEN	AC -> CURRENT_LEN
2C8	7AF7	CMP (PTR)+	Увеличение PTR
2C9	0200	CLA	0 -> AC
2CA	E8F5	ST (PTR)	AC -> (PTR) (0 -> (PTR))
2CB (CLEAR)	0200	CLA	0 -> AC
2CC (ADD_SYMBOL)	1207	IN 7	Получение SR БУ-3
2CD	2F40	AND 0x40	Если SR = 1, то AC != 0
2CE	F0FD	BEQ ADD_SYMBOL	Если AC == 0, перемещение в ADD_SYMBOL
2CF	A8F0	LD (PTR)	(PTR) -> AC

2D0	F104	BNE IS_SECOND	Если данное слово непусто, то в нем уже лежит одна буква, AC != 0, перемещение в IS_SECOND
2D1	1206	IN 6	Получение DR из ВУ-3
2D2	0680	SWAB	Меняем местами старший и младший байт AC
2D3	E8EC	ST (PTR)	AC -> (PTR) (загрузка символа в массив)
2D4	CE05	JUMP NEXT	Переход в NEXT
2D5 (IS_SECOND)	1206	IN 6	Получение DR из ВУ-3
2D6	E8E9	ST (PTR)	AC -> (PTR) (загрузка символа в массив)
2D7	7AE8	CMP (PTR)+	Увеличение PTR (так как уже в одной ячейке лежат два символа)
2D8	0200	CLA	0 -> AC
2D9	E8E6	ST (PTR)	AC -> (PTR) (0 -> (PTR))
2DA (NEXT)	8EE4	LOOP CURRENT_LEN	CURRENT_LEN - 1 -> CURRENT_LEN, Если CURRENT_LEN >= 0, иначе IP + 1 -> IP
2DB	CEEF	JUMP CLEAR	Переход в CLEAR
2DC	0100	HLT	Останов
...
5FD	-	-	Начало строки

3 Описание программы

3.1 Реализуемая функция

Данная программа принимает ввод с ВУ-3 и записывает полученные данные в память.

3.2 Область допустимых значений

Так как при передаче строки первым делом вводится ее длина, которая кодируется 1 байтом, размер строки не может быть больше 255 (или меньше 0).

Все символы кодируются в соответствии со стандартом кодировки ISO-8859-5, в которой каждый символ также кодируется 1 байтом, то есть может принимать 256 различных значений.

3.3 Расположение программы в памяти

2C1 - 2DC - Расположение программы в памяти, 2BF, 2C0 - длина строки и указатель на ее элемент, 5FD - начало строки (массив).

4 Трассировка программы

Положим в программу значения, пусть длина равна 2, вводим строку "Да" в кодировке ISO-8859-5.

Адрес	Код	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адрес	Новый код
2BF	0000	2BF	0000	000	0000	000	0000	0000	004	0100		
2BF	0000	2C0	0000	2BF	0000	000	02BF	0000	004	0100		
2C0	05FD	2C1	05FD	2C0	05FD	000	02C0	0000	004	0100		
2C1	0200	2C2	0200	2C1	0200	000	02C1	0000	004	0100		
2C2	1207	2C3	1207	2C2	1207	000	02C2	0000	004	0100		
2C3	2F40	2C4	2F40	2C3	0040	000	0040	0000	004	0100		
2C4	F0FD	2C2	F0FD	2C4	F0FD	000	FFFD	0000	004	0100		
2C2	1207	2C3	1207	2C2	1207	000	02C2	0040	004	0100		
2C3	2F40	2C4	2F40	2C3	0040	000	0040	0040	000	0000		
2C4	F0FD	2C5	F0FD	2C4	F0FD	000	02C4	0040	000	0000		
2C5	1206	2C6	1206	2C5	1206	000	02C5	0002	000	0000		
2C6	E5FD	2C7	E5FD	5FD	0002	000	02C6	0002	000	0000	5FD	0002
2C7	EEF7	2C8	EEF7	2BF	0002	000	FFF7	0002	000	0000	2BF	0002
2C8	7AF7	2C9	7AF7	5FD	0002	000	FFF7	0002	005	0101	2C0	05FE
2C9	0200	2CA	0200	2C9	0200	000	02C9	0000	005	0101		
2CA	E8F5	2CB	E8F5	5FE	0000	000	FFF5	0000	005	0101	5FE	0000
2CB	0200	2CC	0200	2CB	0200	000	02CB	0000	005	0101		
2CC	1207	2CD	1207	2CC	1207	000	02CC	0000	005	0101		

2CD	2F40	2CE	2F40	2CD	0040	000	0040	0000	005	0101		
2CE	F0FD	2CC	F0FD	2CE	F0FD	000	FFFD	0000	005	0101		
2CC	1207	2CD	1207	2CC	1207	000	02CC	0040	005	0101		
2CD	2F40	2CE	2F40	2CD	0040	000	0040	0040	001	0001		
2CE	F0FD	2CF	F0FD	2CE	F0FD	000	02CE	0040	001	0001		
2CF	A8F0	2D0	A8F0	5FE	0000	000	FFF0	0000	005	0101		
2D0	F104	2D1	F104	2D0	F104	000	02D0	0000	005	0101		
2D1	1206	2D2	1206	2D1	1206	000	02D1	00B4	005	0101		
2D2	0680	2D3	0680	2D2	0680	000	02D2	B400	009	1001		
2D3	E8EC	2D4	E8EC	5FE	B400	000	FFEC	B400	009	1001	5FE	B400
2D4	CE05	2DA	CE05	2D4	02DA	000	0005	B400	009	1001		
2DA	8EE4	2DB	8EE4	2BF	0001	000	0000	B400	009	1001	2BF	0001
2DB	CEEF	2CB	CEEF	2DB	02CB	000	FFEF	B400	009	1001		
2CB	0200	2CC	0200	2CB	0200	000	02CB	0000	005	0101		
2CC	1207	2CD	1207	2CC	1207	000	02CC	0040	005	0101		
2CD	2F40	2CE	2F40	2CD	0040	000	0040	0040	001	0001		
2CE	F0FD	2CF	F0FD	2CE	F0FD	000	02CE	0040	001	0001		
2CF	A8F0	2D0	A8F0	5FE	B400	000	FFF0	B400	009	1001		
2D0	F104	2D5	F104	2D0	F104	000	0004	B400	009	1001		
2D5	1206	2D6	1206	2D5	1206	000	02D5	B4C0	009	1001		
2D6	E8E9	2D7	E8E9	5FE	B4C0	000	FFE9	B4C0	009	1001	5FE	B4C0
2D7	7AE8	2D8	7AE8	5FE	B4C0	000	FFE8	B4C0	005	0101	2C0	05FF

2D8	0200	2D9	0200	2D8	0200	000	02D8	0000	005	0101		
2D9	E8E6	2DA	E8E6	5FF	0000	000	FFE6	0000	005	0101	5FF	0000
2DA	8EE4	2DC	8EE4	2BF	0000	000	FFFF	0000	005	0101	2BF	0000
2DC	0100	2DD	0100	2DC	0100	000	02DC	0000	005	0101		

5 Дополнительное задание №1: ввод с калькулятора, вывод в текстовый принтер

ORG 0x2BF

CURRENT_LEN: WORD 0x0

PTR: WORD 0x5FD

START:

CLA

GET_LEN:

IN 0x1D

AND #0x40

BEQ GET_LEN

IN 0x1C

ST 0x5FD

ST CURRENT_LEN

CMP (PTR)+

CLA

ST (PTR)

CLEAR:

CLA

ADD_SYMBOL:

IN 0x1D

AND #0x40

BEQ ADD_SYMBOL

LD (PTR)

BNE IS_SECOND

IN 0x1C

CALL ISO_8859_5

SWAB

```
ST (PTR)
JUMP NEXT
```

```
IS_SECOND:
IN 0x1C
CALL ISO_8859_5
ST (PTR)
CMP (PTR)+
CLA
ST (PTR)
```

```
NEXT:
LOOP CURRENT_LEN
JUMP CLEAR
```

```
PREPARING_FOR_PRINT:
LD STRING_ADDR
ST PTR
LD 0x5FD
ST CURRENT_LEN
CLA
```

```
PRINT_1ST:
IN 0xD
AND #0x40
BEQ PRINT_1ST
CMP (PTR)+
LD (PTR)
SWAB
OUT 0xC
CMP -(CURRENT_LEN)
PRINT_2ND:
IN 0xD
```

```
AND #0x40
BEQ PRINT_2ND
LD (PTR)
OUT 0xC
LOOP CURRENT_LEN
JUMP PRINT_1ST
HLT
```

```
ISO_8859_5:
PUSH
AND BYTE_TWO
CMP #0xA
BEQ MINUS
CMP #0xB
BEQ PLUS
CMP #0xC
BEQ DIV_
CMP #0xD
BEQ MUL_
CMP #0xE
BEQ POINT
CMP #0xF
BEQ EQUAL
ADD #0x30
JUMP EXIT_
MINUS:
LD #0x2D
JUMP EXIT_
PLUS:
LD #0x2B
JUMP EXIT_
DIV_:
LD #0x2F
```

```

JUMP EXIT_
MUL_:
LD #0x2A
JUMP EXIT_
POINT:
LD #0x2E
JUMP EXIT_
EQUAL:
LD #0x3D
JUMP EXIT_
EXIT_:
ST TEMP
POP
AND BYTE_ONE
ADD TEMP
RET

BYTE_ONE: WORD 0xFF00
BYTE_TWO: WORD 0x00FF
TEMP: WORD 0x0
STRING_ADDR: WORD 0x5FD

ORG 0x5FD
STRING: WORD 0x0

```

ЗАКЛЮЧЕНИЕ

При выполнении данной лабораторной работы были изучены и реализованы следующие концепции:

- устройства ввода-вывода, ВУ;
- контроллеры ВУ, работа с ними.