

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

**Отчёт**

**По лабораторной работе №5**

**«Асинхронный обмен данными с ВУ»**

по дисциплине «Основы профессиональной деятельности»

**Вариант: 11501**

Работу выполнил:

Поленов Кирилл Александрович

Группа Р3113

Работу приняла:

Ткешелашвили Нино Мерабиевна

г. Санкт-Петербург 2024

# Оглавление

Задание .....	3
Реализация задания на ассемблере БЭВМ .....	3
ОП и ОДЗ исходных данных и результата .....	4
Трассировка программы .....	5
Дополнительное задание.....	6
Реализация дополнительного задания на ассемблере БЭВМ.....	7
Выводы .....	10

# Задание

## Лабораторная работа №5

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 4B6<sub>16</sub>. Размещаемая строка находится по адресу 5A5<sub>16</sub>.
3. Строка должна быть представлена в кодировке Windows-1251.
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.

## Реализация задания на ассемблере БЭВМ

### Описание программы

Адрес	Содержимое	Мнемоника	Описание
4B6	05A5	ADR	Ячейка для инкрементирования адреса результата
4B7	0200	CLA	0000 -> AC
4B8	1205	IN 5	Ввод содержимого SR BY2 в 6й бит AC
4B9	2F40	AND #40	Условие «6й бит AC == 1»
4BA	F0FD	BEQ (IP - 3)	Переход на 0x4B8, если Z == 1
4BB	1204	IN 4	Ввод содержимого DR BY2 в младший байт AC
4BC	7F00	CMP #00	Сравнение AC со «стоп символом»
4BD	F00B	BEQ (IP + 11)	Переход на 0x4C9, если Z == 1
4BE	0680	SWAB	Свап младшего и старшего байта AC между собой
4BF	E8F6	ST (IP - 10)	Сохранение 1-ого символа по адресу из ячейки 0x4B6
4C0	1205	IN 5	Ввод содержимого SR BY2 в 6й бит AC
4C1	2F40	AND #40	Условие «6й бит AC == 1»
4C2	F0FD	BEQ (IP - 3)	Переход на 0x4C0, если Z == 1
4C3	A8F2	LD (IP - 14)	Загрузка содержимого ячейки по адресу из ячейки 0x4B6
4C4	1204	IN 4	Ввод содержимого DR BY2 в младший байт AC
4C5	7F00	CMP #00	Сравнение AC со «стоп символом»
4C6	F002	BEQ (IP + 2)	Переход на 0x4C9, если Z == 1
4C7	EAEЕ	ST (IP - 18)+	Сохранение 2-ого символа по адресу из ячейки 0x4B6 с увеличением адреса внутри этой ячейки на 1
4C8	CEEF	JUMP (IP - 17)	Переход по адресу 0x4B8
4C9	E8EC	ST (IP - 20)	Сохранение «стоп символа» по адресу из ячейки 0x4B6
4CA	0100	HLT	Конец программы
5A5	0000	RES	Ячейка для сохранения символов слова

Таблица 1

# ОП и ОДЗ исходных данных и результата

## Область представления:

- RES – 16-разрядная ячейка для хранения 2х символов. Старший байт – код первого символа, младший байт – код второго символа
- ADR – 11-разрядное беззнаковое число. Ячейка для хранения адреса начала символов слова

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	<u>NUL</u> 0000	<u>STX</u> 0001	<u>SOT</u> 0002	<u>ETX</u> 0003	<u>EOT</u> 0004	<u>ENQ</u> 0005	<u>ACK</u> 0006	<u>BEL</u> 0007	<u>BS</u> 0008	<u>HT</u> 0009	<u>LF</u> 000A	<u>VT</u> 000B	<u>FF</u> 000C	<u>CR</u> 000D	<u>SO</u> 000E	<u>SI</u> 000F
10	<u>DLE</u> 0010	<u>DC1</u> 0011	<u>DC2</u> 0012	<u>DC3</u> 0013	<u>DC4</u> 0014	<u>NAK</u> 0015	<u>SYN</u> 0016	<u>ETB</u> 0017	<u>CAN</u> 0018	<u>EM</u> 0019	<u>SUB</u> 001A	<u>ESC</u> 001B	<u>FS</u> 001C	<u>GS</u> 001D	<u>RS</u> 001E	<u>US</u> 001F
20	<u>SP</u> 0020	<u>!</u> 0021	<u>"</u> 0022	<u>#</u> 0023	<u>\$</u> 0024	<u>%</u> 0025	<u>&amp;</u> 0026	<u>'</u> 0027	<u>(</u> 0028	<u>)</u> 0029	<u>*</u> 002A	<u>+</u> 002B	<u>,</u> 002C	<u>-</u> 002D	<u>.</u> 002E	<u>/</u> 002F
30	<u>0</u> 0030	<u>1</u> 0031	<u>2</u> 0032	<u>3</u> 0033	<u>4</u> 0034	<u>5</u> 0035	<u>6</u> 0036	<u>7</u> 0037	<u>8</u> 0038	<u>9</u> 0039	<u>:</u> 003A	<u>;</u> 003B	<u>&lt;</u> 003C	<u>=</u> 003D	<u>&gt;</u> 003E	<u>?</u> 003F
40	<u>@</u> 0040	<u>A</u> 0041	<u>B</u> 0042	<u>C</u> 0043	<u>D</u> 0044	<u>E</u> 0045	<u>F</u> 0046	<u>G</u> 0047	<u>H</u> 0048	<u>I</u> 0049	<u>J</u> 004A	<u>K</u> 004B	<u>L</u> 004C	<u>M</u> 004D	<u>N</u> 004E	<u>O</u> 004F
50	<u>P</u> 0050	<u>Q</u> 0051	<u>R</u> 0052	<u>S</u> 0053	<u>T</u> 0054	<u>U</u> 0055	<u>V</u> 0056	<u>W</u> 0057	<u>X</u> 0058	<u>Y</u> 0059	<u>Z</u> 005A	<u>[</u> 005B	<u>\</u> 005C	<u>]</u> 005D	<u>^</u> 005E	<u>_</u> 005F
60	<u>`</u> 0060	<u>a</u> 0061	<u>b</u> 0062	<u>c</u> 0063	<u>d</u> 0064	<u>e</u> 0065	<u>f</u> 0066	<u>g</u> 0067	<u>h</u> 0068	<u>i</u> 0069	<u>j</u> 006A	<u>k</u> 006B	<u>l</u> 006C	<u>m</u> 006D	<u>n</u> 006E	<u>o</u> 006F
70	<u>p</u> 0070	<u>q</u> 0071	<u>r</u> 0072	<u>s</u> 0073	<u>t</u> 0074	<u>u</u> 0075	<u>v</u> 0076	<u>w</u> 0077	<u>x</u> 0078	<u>y</u> 0079	<u>z</u> 007A	<u>{</u> 007B	<u> </u> 007C	<u>}</u> 007D	<u>~</u> 007E	<u>DEL</u> 007F
80	<u>Ъ</u> 0402	<u>Ѓ</u> 0403	<u>Ѕ</u> 201A	<u>Ї</u> 0453	<u>Љ</u> 201E	<u>Њ</u> 2026	<u>Ћ</u> 2020	<u>Ќ</u> 2021	<u>Ў</u> 20AC	<u>а</u> 2030	<u>б</u> 0409	<u>в</u> 2039	<u>г</u> 040A	<u>д</u> 040C	<u>е</u> 040B	<u>ж</u> 040F
90	<u>Ђ</u> 0452	<u>Ї</u> 2018	<u>Ѕ</u> 2019	<u>Ї</u> 201C	<u>Љ</u> 201D	<u>Њ</u> 2022	<u>Ћ</u> 2013	<u>Ќ</u> 2014	<u>Ў</u> 2122	<u>а</u> 0459	<u>б</u> 203A	<u>в</u> 045A	<u>г</u> 045C	<u>д</u> 045B	<u>е</u> 045E	<u>ж</u> 045F
A0	<u>Њ</u> 00A0	<u>Ѓ</u> 040E	<u>Ѕ</u> 045E	<u>Ї</u> 0408	<u>Љ</u> 00A4	<u>Њ</u> 0490	<u>Ћ</u> 00A6	<u>Ќ</u> 00A7	<u>Ў</u> 0401	<u>а</u> 00A9	<u>б</u> 0404	<u>в</u> 00AB	<u>г</u> 00AC	<u>д</u> 00AD	<u>е</u> 00AE	<u>ж</u> 0407
B0	<u>°</u> 00B0	<u>±</u> 00B1	<u>І</u> 0406	<u>і</u> 0456	<u>Г</u> 0491	<u>μ</u> 00B5	<u>¶</u> 00B6	<u>·</u> 00B7	<u>ё</u> 0451	<u>№</u> 2116	<u>є</u> 0454	<u>»</u> 00BB	<u>ј</u> 0458	<u>Ѕ</u> 0405	<u>Ѕ</u> 0455	<u>і</u> 0457
C0	<u>А</u> 0410	<u>В</u> 0411	<u>В</u> 0412	<u>Г</u> 0413	<u>Д</u> 0414	<u>Е</u> 0415	<u>Ж</u> 0416	<u>З</u> 0417	<u>И</u> 0418	<u>Й</u> 0419	<u>К</u> 041A	<u>Л</u> 041B	<u>М</u> 041C	<u>Н</u> 041D	<u>О</u> 041E	<u>П</u> 041F
D0	<u>Р</u> 0420	<u>С</u> 0421	<u>Т</u> 0422	<u>У</u> 0423	<u>Ф</u> 0424	<u>Х</u> 0425	<u>Ц</u> 0426	<u>Ч</u> 0427	<u>Ш</u> 0428	<u>Щ</u> 0429	<u>Ъ</u> 042A	<u>Ы</u> 042B	<u>Ь</u> 042C	<u>Э</u> 042D	<u>Ю</u> 042E	<u>Я</u> 042F
E0	<u>а</u> 0430	<u>б</u> 0431	<u>в</u> 0432	<u>г</u> 0433	<u>д</u> 0434	<u>е</u> 0435	<u>ж</u> 0436	<u>з</u> 0437	<u>и</u> 0438	<u>й</u> 0439	<u>к</u> 043A	<u>л</u> 043B	<u>м</u> 043C	<u>н</u> 043D	<u>о</u> 043E	<u>п</u> 043F
F0	<u>р</u> 0440	<u>с</u> 0441	<u>т</u> 0442	<u>у</u> 0443	<u>ф</u> 0444	<u>х</u> 0445	<u>ц</u> 0446	<u>ч</u> 0447	<u>ш</u> 0448	<u>щ</u> 0449	<u>ъ</u> 044A	<u>ы</u> 044B	<u>ь</u> 044C	<u>э</u> 044D	<u>ю</u> 044E	<u>я</u> 044F

## Область определения:

- 8-ричный код символа для ввода  $\in [0x21; 0xFF] \setminus \{0x7F, 98, 0xA0\}$   
(Исключены служебные символы)
- Максимально возможное количество символов для ввода = 1204  
( $2047 (0x7FF) - 1445 (0x5A5)) * 2 = 602 * 2 = 1204$ )

# Трассировка программы

Слово для трассировки: РЫЦАРЬ

Слово в кодировке windows-1251: D0 DB D6 C0 D0 DC

Трассировка производится для первых двух символов (D0 DB)

## Таблица трассировки

Выполняемая команда		Содержимое регистров после выполнения команды								Ячейка, содержимое которой изменилось после выполнения команды	
Адрес	Содержимое	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Содержимое
xxx	xxxx	xxx	xxxx	xxx	xxxx	xxx	xxxx	xxxx	xxxx	xxx	xxxx
4B7	0200	4B8	0200	4B7	0200	000	04B7	0000	0100	-	-
4B8	1205	4B9	1205	4B8	1205	000	04B8	0040	0100	-	-
4B9	2F40	4BA	2F40	4B9	0040	000	0040	0040	0000	-	-
4BA	F0FD	4BB	F0FD	4BA	F0FD	000	04BA	0040	0000	-	-
4BB	1204	4BC	1204	4BB	1204	000	04BB	00D0	0000	-	-
4BC	7F00	4BD	7F00	4BC	0000	000	0000	00D0	0001	-	-
4BD	F00B	4BE	F00B	4BD	F00B	000	04BD	00D0	0001	-	-
4BE	0680	4BF	0680	4BE	0680	000	04BE	D000	1001	-	-
4BF	E8F6	4C0	E8F6	5A5	D000	000	FFF6	D000	1001	-	-
4C0	1205	4C1	1205	4C0	1205	000	04C0	D040	1001	-	-
4C1	2F40	4C2	2F40	4C1	0040	000	0040	0040	0001	-	-
4C2	F0FD	4C3	F0FD	4C2	F0FD	000	04C2	0040	0001	-	-
4C3	A8F2	4C4	A8F2	5A5	D000	000	FFF2	D000	1001	-	-
4C4	1204	4C5	1204	4C4	1204	000	04C4	D0DB	1001	-	-
4C5	7F00	4C6	7F00	4C5	0000	000	0000	D0DB	1001	-	-
4C6	F002	4C7	F002	4C6	F002	000	04C6	D0DB	1001	-	-
4C7	EAEE	4C8	EAFE	5A5	D0DB	000	FFEE	D0DB	1001	4B6	5A6
										5A5	D0DB
4C8	CEEF	4B8	CEEF	4C8	04B8	000	FFEF	D0DB	1001	-	-
4B8	1205	4B9	1205	4B8	1205	000	04B8	D040	1001	-	-
4B9	2F40	4BA	2F40	4B9	0040	000	0040	0040	0001	-	-
4BA	F0FD	4BB	F0FD	4BA	F0FD	000	04BA	0040	0001	-	-

4BB	1204	4BC	1204	4BB	1204	000	04BB	0000	0001	-	-
4BC	7F00	4BD	7F00	4BC	0000	000	0000	0000	0101	-	-
4BD	F00B	4C9	F00B	4BD	F00B	000	000B	0000	0101	-	-
4C9	E8EC	4CA	E8EC	5A6	0000	000	FFEC	0000	0101	5A6	0000
4CA	0100	4CB	0100	04CA	0100	000	04CA	0000	0101	-	-

*Таблица 2*

## Дополнительное задание

По нажатию на готовность ВУ-2 программа случайным образом рисует на ВУ-6 (бегущая строка) одно из трёх изображений: камень, ножницы или бумагу. Выпадение всех фигур должно быть равновероятным.

## Реализация дополнительного задания на ассемблере БЭВМ

```
1  ORG 0x011
2  COUNTER: WORD 0x0000
3  MAX:     WORD 0x0003
4
5  START:  LD COUNTER
6          INC
7
8          CMP #01
9          BEQ PRINTSTONE
10 PRINTSTONE: PUSH
11             CALL $STONE
12             POP
13
14          CMP #02
15          BEQ PRINTSCISSORS
16 PRINTSCISSORS: PUSH
17             CALL $SCISSORS
18             POP
19
20          CMP #03
21          BEQ PRINTPAPER
22 PRINTPAPER: PUSH
23             CALL $PAPER
24             POP
25
26          CMP MAX
27          BEQ RESET
28          JUMP START
29
30 RESET:  LD#00
31          ST COUNTER
32
```

```
34  ORG 0x050
35  SCISSORS: LD#0x82
36          OUT 0x10
37          LD#0x45
38          OUT 0x10
39          LD#0x2A
40          OUT 0x10
41          LD#0x10
42          OUT 0x10
43          LD#0x2A
44          OUT 0x10
45          LD#0x45
46          OUT 0x10
47          LD#0x82
48          OUT 0x10
49          LD#0x00
50          OUT 0x10
51          LD#0x00
52          OUT 0x10
53          RET
54
```



```
55  ORG 0x065
56  STONE: LD#0x07
57         OUT 0x10
58         LD#0x0F
59         OUT 0x10
60         LD#0x1F
61         OUT 0x10
62         LD#0x7F
63         OUT 0x10
64         LD#0xFF
65         OUT 0x10
66         LD#0xFF
67         OUT 0x10
68         LD#0xFF
69         OUT 0x10
70         LD#0x7F
71         OUT 0x10
72         LD#0x07
73         OUT 0x10
74         LD#0x00
75         OUT 0x10
76         LD#0x00
77         OUT 0x10
78         RET
79
```

```

80      ORG 0x07E
81      PAPER:  LD#0x0E
82              OUT 0x10
83              LD#0x1E
84              OUT 0x10
85              LD#0x3E
86              OUT 0x10
87              LD#0x7E
88              OUT 0x10
89              LD#0x7E
90              OUT 0x10
91              LD#0x7E
92              OUT 0x10
93              LD#0x7C
94              OUT 0x10
95              LD#0x78
96              OUT 0x10
97              LD#0x70
98              OUT 0x10
99              LD#0x00
100             OUT 0x10
101             LD#0x00
102             OUT 0x10
103             LD#0x00
104             OUT 0x10
105             RET

```

## Выводы

В ходе данной лабораторной работы я:

- Познакомился с реализацией подпрограмм в БЭВМ
- Познакомился с такой структурой данных, как стек
- Закрепил знания о режимах адресации в БЭВМ