

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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

Вариант 8777

Выполнил:

Горин Семён Дмитриевич

Группа Р3108

Проверил:

Вербовой Александр Александрович

Санкт-Петербург 2025

Содержание

<i>Задание.....</i>	<i>3</i>
<i>Текст исходной программы на языке Ассемблера БЭВМ.....</i>	<i>3</i>
<i>Таблица трассировки</i>	<i>4</i>
<i>Выводы</i>	<i>7</i>

Задание

По выданному преподавателем варианту(представлен на рисунке 1) 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 вывод данных на ВУ-3
2. Программа начинается с адреса 494_{16} . Размещаемая строка находится по адресу $5AC_{16}$.
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.

Рисунок 1

Текст исходной программы на языке Ассемблера БЭВМ

Заданное слово	Коды символов слова в кодировке Windows-1251
Луна	CB F3 ED E0

; Запись строки в память

ORG 0x5AC

STRING: WORD 0xCBF3

WORD 0xEDE0

; Программа

ORG 0x492

MASK: WORD 0xFF

POS: WORD \$STRING ; Указатель на текущую пару символов

START: CLA

BEGIN: IN 7 ; Чтение регистра состояния ВУ-3

AND #0x40 ; Проверка готовности

BEQ BEGIN ; Спин - луп

LD (POS) ; Загрузка в АС первых 2 байтов строки

SWAB ; Меняем местами 1 и 2 байт строки

AND MASK ; Оставляем только младший байт

BEQ STOP ; Если младший байт равен 00 то Останов

OUT 6 ; Запись в DR ВУ-3 первого символа

CHECK: IN 7 ; Чтение регистра состояния ВУ-3

AND #0x40 ; Проверка готовности
 BEQ CHECK ; Спин - луп
 LD (POS)+ ; Загрузка первых 2 байтов строки и увеличение указателя на 1
 AND MASK ; Оставляем только младший байт
 BEQ STOP ; Если младший байт равен 00 то Останов
 OUT 6 ; Запись в DR ВУ-3 первого символа
 JUMP BEGIN ; Повторяем цикл вывода на ВУ-3

 STOP: HLT ; Останов

Таблица трассировки

Таблица 1

Выполняемая команда		Содержимое регистров процессора после выполнения команды								Ячейка, содержащее которой изменилось после выполнения команды	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый Код
494	200	495	200	494	200	0	494	0	0100	-	-
495	1207	496	1207	495	1207	0	495	40	0100	-	-
496	2F40	497	2F40	496	40	0	40	40	0000	-	-
497	F0FD	498	F0FD	497	F0FD	0	497	40	00	-	-
498	A8FA	499	A8FA	5AC	CBF3	0	FFFA	CBF3	1000	-	-
499	680	49A	680	499	680	0	499	F3CB	1000	-	-
49A	2EF7	49B	2EF7	492	00FF	0	FFF7	00CB	0000	-	-
49B	F009	49C	F009	49B	F009	0	049B	00CB	0000	-	-
49C	1306	49D	1306	49C	1306	0	049C	00CB	0000	-	-
49D	1207	49E	1207	49D	1207	0	049D	40	0000	-	-
49E	2F40	49F	2F40	49E	40	0	40	40	0000	-	-
49F	F0FD	4A0	F0FD	49F	F0FD	0	049F	40	0000	-	-
4A0	AAF2	4A1	AAF2	5AC	CBF3	0	FFF2	CBF3	1000	493	05AD

4A1	2EF0	4A2	2EF0	492	00FF	0	FFF0	00F3	0000	-	-
4A2	F002	4A3	F002	4A2	F002	0	04A2	00F3	0000	-	-
4A3	1306	4A4	1306	4A3	1306	0	04A3	00F3	0000	-	-
4A4	CEF0	495	CEF0	4A4	495	0	FFF0	00F3	0000	-	-

Дополнительное задание

```

ORG 0x100;
; Переменные
PST: WORD 0x500;
POS: WORD 0x500;

APOS: WORD 0x4FF;
AIND: WORD 8;
AOI: WORD 0;

CD: WORD ?;

; Программа

STOP: HLT;
; Очистка дисплея
START:
LD #0x0F
OUT 0x14
LD #0x1F
OUT 0x14
LD #0x2F
OUT 0x14
LD #0x3F
OUT 0x14
LD #0x4F
OUT 0x14
LD #0x5F
OUT 0x14
LD #0x6F
OUT 0x14
LD #0x7F
OUT 0x14
LD #0x8F
OUT 0x14
; Очистка переменных
LD #8;
ST AIND;
LD #0;
ST AOI;
LD PST;
ST POS;
DEC
ST APOS;
; Считывание слова с Numpad'a
INPUT: IN 0x1D
AND #0x40
BEQ INPUT

IN 0x1C

CMP #0xF
BEQ CALC

```

```

ST (POS)+
JUMP INPUT

; Тело программы

CALC: LD POS
CMP PST
BEQ FORMAT
DEC
CMP PST
BEQ FORMAT
DEC
ST CD
ST POS

; Подсчет часов

; Если в текущем разряде < 6 десятков минут, то занимаем из старшего
COUNT: LD (POS)
CMP #0x6
BMI BORROW
; Иначе вычитаем 6 из разряда десятков минут
SUB #6
ST (POS)
; Загружаем текущий разряд ответа, если он равен 9, то зануляем текущий
разряд, сдвигаем указатель на 1 и записываем в новый разряд 1.
LD (APOS)
CMP #9
BEQ SHIFT
; Иначе разряд +1
INC
ST (APOS)
; Цикл обновляется
JUMP COUNT

; Зануляем разряд и сдвигаем указатель
SHIFT: CLA
ST (APOS)
; проверяем следующий разряд на переполнение
LD -(APOS)
CMP #9;
BEQ SHIFT
INC
ST (APOS)
; Переход на младший разряд
LD PST
DEC
ST APOS
; Снова считать часики...
JUMP COUNT

; Подпрограмма заема из более старшего разряда
BORROW:
; Проверка что указатель разряда не превысил длину числа, иначе все разряды
числа нулевые, а значит - вывод
LD CD
CMP PST
BEQ FORMAT

; Проверка что разряд не нулевой иначе заем из следующего
LD -(CD)
BEQ BORROW

```

```

; Вычитаем 1 из старшего разряда, сдвигаем указатель разряда
DEC
ST (CD)+
; Прибавляем к младшему разряду 10
LD (CD)
ADD #0xA
ST (CD)
LD POS
ST CD
JUMP COUNT

; Форматирование полученных цифр для вывода на индикатор
FORMAT:
; Загружаем позицию цифры
LD AOI
; сдвигаем чтобы позиция занимала старшую тетраду младшего байта
ASL
ASL
ASL
ASL
; Записываем номер разряда для вывода
OR (APOS)
ST (APOS)
LD (AOI)+
OR -(APOS)
LOOP AIND
JUMP FORMAT
OR (APOS)+
; Вывод
OUTPUT: IN 0x15
AND #0x40
BEQ OUTPUT
LD (APOS)
OUT 0x14
CLA
ST (APOS)+
LD APOS
CMP PST
BEQ STOP
JUMP OUTPUT

```

Выводы

Во время выполнения лабораторной работы я изучил конструкции языка Ассемблер БЭВМ, познакомился со структурой некоторых ВУ, узнал о том как БЭВМ может с ними взаимодействовать.