

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5

По Основам Профессиональной Деятельности

Вариант №14013

Выполнил:

Ступин Тимур Русланович

Группа № Р3108

Проверил:

Вербовой Александр Александрович

Санкт-Петербург 2024

Содержание

Задание.....	3
Текст программы	4
Описание программы	5
Расположение данных в памяти	5
Область допустимых значений.....	5
Трассировка на выданных данных.....	5
Дополнительной задание	7
Вывод.....	7

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 $0CB_{16}$. Размещаемая строка находится по адресу $61A_{16}$.
3. Стока 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 0D (CR). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

Текст программы

Адрес	Метка	Команда ассемблера	Аргумент	Описание
		ORG	0x0CB	Адрес начала программы
0CB	BEGIN:	WORD	0x61A	Адрес начала строки
0CC	CUR:	WORD	?	Адрес текущего элемента строки
0CD	MASK:	WORD	0x0D	Стоп-символ
0CE	TMP:	WORD	?	Буферная переменная
0CF	FLAG:	WORD	0x0	Флаг на 1/2 символ
0D0	START:	LD	BEGIN	Загрузка адреса первого символа
0D1		ST	CUR	
0D2	READ:	CLA		Начало чтения
0D3	CHECK:	IN	5	Ожидание готовности ВУ2
0D4		AND	#0x40	
0D5		BEQ	CHECK	
0D6		IN	4	Чтение символа
0D7		ST	TMP	Сохранение введенного символа для дальнейшей проверки на стоп
0D8		LD	FLAG	Проверка флага
0D9		BEQ	S1	
0DA		LD	TMP	Сохранение второго символа
0DB		ADD	(CUR)	
0DC		ST	(CUR)+	
0DD		JUMP	SW	Переход на переключение флага
0DE	S1:	LD	TMP	Сохранение первого символа
0DF		SWAB		
0E0		ST	(CUR)	
0E1	SW:	LD	FLAG	Переключение флага
0E2		NOT		
0E3		ST	FLAG	
0E4		LD	TMP	Проверка на окончание ввода
0E5		CMP	MASK	
0E6		BEQ	STOP	
0E7		JUMP	READ	Переход на чтение символов
0E8	STOP:	HLT		Окончание работы программы

Описание программы

Программа осуществляет ввод данных с ВУ-2

Расположение данных в памяти

0x0CB – адрес начала программы

0x0E8 – адрес конца программы

0x61A – адрес начала массива строки

0x0CC – адрес текущей ячейки в массиве

Область допустимых значений

Адрес начала массива строки должен располагаться после программы, значит он может принимать значения $\in [0xE9, 0x7FF]$

Каждый символ к кодировке ISO-8859-5 $\in [0x21, 0x7E] \cup [0xA1, 0xFF]$

Трассировка на выданных данных

Слово для трассировки: СИНХРОНИЧНОСТЬ

ISO-8859-5: C1 B8 BD C5 C0 BE BD B8 C7 BD BE C1 C2 CC

UTF-8: D0A1 D098 D09D D0A5 D0A0 D09E D09D D098 D0A7 D09D D09E D0A1 D0A2 D0AC

UTF-16: 0421 0418 041D 0425 0420 041E 041D 0418 0427 041D 041E 0421 0422 042C

Выполняемая команда		Содержание регистров в процессоре после выполнения команды								Ячейка, содержимое которой изменилось после выполнения команды	
Адр	Знчн	IP	CR	AR	DR	SP	BR	AC	NZVC	Адр	Знчн
0D0	AEFA	0D1	AEFA	0CB	061A	000	FFFA	061A	0000		
0D1	EEFA	0D2	EEFA	0CC	061A	000	FFFA	061A	0000	0CC	061A
0D2	0200	0D3	0200	0D2	0200	000	00D2	0000	0100		
0D3	1205	0D4	1205	0D3	1205	000	00D3	0040	0100		
0D4	2F40	0D5	2F40	0D4	0040	000	0040	0040	0000		
0D5	F0FD	0D6	F0FD	0D5	F0FD	000	00D5	0040	0000		

0D6	1204	0D7	1204	0D6	1204	000	00D6	00C1	0000		
0D7	EEF6	0D8	EEF6	0CE	00C1	000	FFF6	00C1	0000	0CE	00C1
0D8	AEF6	0D9	AEF6	0CF	0000	000	FFF6	0000	0100		
0D9	F004	0DE	F004	0D9	F004	000	0004	0000	0100		
0DE	AEEF	0DF	AEEF	0CE	00C1	000	FFEF	00C1	0000		
0DF	0680	0E0	0680	0DF	0680	000	00DF	C100	1000		
0E0	E8EB	0E1	E8EB	61A	C100	000	FFEB	C100	1000	61A	C100
0E1	AEED	0E2	AEED	0CF	0000	000	FFED	0000	0100		
0E2	0280	0E3	0280	0E2	0280	000	00E2	FFFF	1000		
0E3	EEE8	0E4	EEE8	0CF	FFFF	000	FFEB	FFFF	1000	0CF	FFFF
0E4	AEE9	0E5	AEE9	0CE	00C1	000	FFE9	00C1	0000		
0E5	7EE7	0E6	7EE7	0CD	000D	000	FFE7	00C1	0001		
0E6	F001	0E7	F001	0E6	F001	000	00E6	00C1	0001		
0E7	CEEA	0D2	CEEA	0E7	00D2	000	FFEA	00C1	0001		
0D2	0200	0D3	0200	0D2	0200	000	00D2	0000	0101		
0D3	1205	0D4	1205	0D3	1205	000	00D3	0040	0101		
0D4	2F40	0D5	2F40	0D4	0040	000	0040	0040	0001		
0D5	F0FD	0D6	F0FD	0D5	F0FD	000	00D5	0040	0001		
0D6	1204	0D7	1204	0D6	1204	000	00D6	00B8	0001		
0D7	EEF6	0D8	EEF6	0CE	00B8	000	FFF6	00B8	0001	0CE	00B8
0D8	AEF6	0D9	AEF6	0CF	FFFF	000	FFF6	FFFF	1001		
0D9	F004	0DA	F004	0D9	F004	000	00D9	FFFF	1001		
0DA	AEF3	0DB	AEF3	0CE	00B8	000	FFF3	00B8	0001		
0DB	48F0	0DC	48F0	61A	C100	000	FFF0	C1B8	1000		
0DC	EA8F	0DD	EA8F	61A	C1B8	000	FFEF	CAB8	1000	61A	C1B8
0DD	CE03	0E1	CE03	0DD	00E1	000	0003	C1B8	1000		
0E1	AEED	0E2	AEED	0CF	FFFF	000	FFED	FFFF	1000		
0E2	0280	0E3	0280	0E2	0280	000	00E2	0000	0100		
0E3	EEE8	0E4	EEE8	0CF	0000	000	FFEB	0000	0100	0CF	0000
0E4	AEE9	0E5	AEE9	0CE	00B8	000	FFE9	00B8	0000		
0E5	7EE7	0E6	7EE7	0CD	000D	000	FFE7	00B8	0001		

0E6	F001	0E7	F001	0E6	F001	000	00E6	00B8	0001		
0E7	CEEA	0D2	CEEA	0E7	00D2	000	FFEA	00B8	0001		

Дополнительной задание

- На одно ВУ (по вашему выбору) передать значение даты в формате (MM.dd) (пример: 03.13) а на принтер выводить буквенное значение даты (Тринадцатое Марта)

В качестве устройства ввода я выбрал цифровую клавиатуру (ВУ-9)

Для принтера была использована кодировка КОИ8-Р

[Ссылка на код](#)

Вывод

В ходе работы я изучил принцип работы ввода и вывода в БЭВМ, разработал программу асинхронного обмена с внешним устройством, осуществил ввод информации с устройства ввода ВУ-2. Также я изучил принципы написания программ для БЭВМ на языке Assembler и использовал этот язык для более быстрого написания кода.