

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 5

"Асинхронный обмен данными с ВУ"

по дисциплине

«ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

Вариант №5122

Выполнил:

Студент группы Р3118

Шипунов Илья

Михайлович

Преподаватель:

Перминов Илья

Валентинович

Санкт-Петербург

2022

Задание и основные этапы выполнения

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.

1. Программа осуществляет асинхронный ввод данных с ВУ-3
2. Программа начинается с адреса 230_{16} . Размещаемая строка находится по адресу 611_{16} .
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ1 СИМВ2 АДР2: 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.

1. Программа.

Метка	Мнемоника	Параметр	Описание
	ORG	0x230	
START	CALL	READ	Блок начала работы. Производится чтение с внешнего устройства, а затем младший байт считанного значения, равного количеству символов, предстоящих для чтения, помещается в ячейку-счётчик (AMT) и в первую ячейку для хранения результата (CUR).
	ST	AMT	
	ST	(CUR)+	
	BEQ	FINISH	
S1	CALL	READ	Блок чтения и записи первого символа для каждой следующей ячейки хранения результата (CUR). Производится чтение с внешнего устройства, а затем младший байт считанного значения, равный первому символу, расположенному по текущему адресу, помещается в старший байт аккумулятора и записывается в память по текущему адресу для хранения результата (CUR), после чего производится проверка количества предстоящих итераций, если значение ячейки-счётчика (AMT) == 0 => переход к блоку завершения программы.
	SWAB		
	ST	(CUR)	
	LOOP	(AMT)	
	JUMP	S2	
	JUMP	FINISH	
S2	CALL	READ	Блок чтения и записи второго символа для каждой следующей ячейки хранения результата (CUR). Производится чтение с внешнего устройства, а затем младший байт считанного значения, равный первому символу, расположенному по текущему адресу, помещается в предварительно загруженный в аккумулятор младший байт значения по текущему адресу ячейки хранения результата (CUR) и записывается в память по её адресу, после чего производится проверка количества предстоящих итераций, если значение ячейки-счётчика (AMT) == 0 => переход к блоку завершения программы. Иначе переход к блоку чтения и записи первого символа.
	OR	(CUR)	
	ST	(CUR)+	
	LOOP	(AMT)	
	JUMP	S1	
FINISH	HLT		Блок завершения программы.
READ	CLA		Блок чтения содержимого, переданного через регистры ВУ. Осуществляется циклическая проверка готовности ВУ, после подтверждения которой происходит чтение из ВУ и запись содержимого в аккумулятор.
	IN	7	
	AND	#0x40	
	BEQ	READ	
	IN	6	
	RET		
AMT	WORD	0x610	Ячейка-счётчик
CUR	WORD	0x611	Первая ячейка для хранения результата

2. Описание программы.

Программа производит чтение из регистра ВУ-3 и запись считанных символов в массив, первый элемент которого находится по адресу 611.

Расположение в памяти БЭВМ программы, исходных данных и результатов:

0x230 – 0x23F – инструкции основной программы.

0x240 – 0x245 – инструкции подпрограммы для чтения содержимого регистра ВУ-3.

0x246 – адрес ячейки-счётчика

0x247 – адрес первого элемента массива.

0x610 – ячейка-счётчик количества символов к чтению.

0x611 – 0x7FF – массив, для хранения результатов считывания.

Адреса первой и последней выполняемой инструкции программы.

0x230 – адрес первой инструкции.

0x23F – адрес последней инструкции.

Область представления для исходных данных:

0x610 – знаковое 16-разрядное число.

Область представления для результата:

0x611 – знаковое 16-разрядное число.

0x612 – 0x7FF – два символа (в старшем и младшем байтах) в кодировке КОИ – 8.

Нахождение области допустимых значений для исходных данных и результата:

Ввиду отсутствия переменных и арифметических операций можно сделать вывод о том, что все данные в программе могут иметь любые возможные значения, кроме адреса размещения элементов массива.

С учётом расположения программы по адресам 0x230 – 0x247 и максимально возможного размера массива для хранения символов, найдём ОДЗ для адреса размещения этого массива.

Максимальное количество символов в вводимой строке = 255 => количество элементов в массиве = $\text{ceil}(255/2) = 128$, с учётом двух дополнительных ячеек для хранения количества символов имеем массив максимальным размером 130. =>

$$x \in [0; 560 - 130) \cup (583; 2047 - 130) \Rightarrow x \in [0; 430) \cup (583; 1917]$$

Учитывая то, что в программе задействован стек, а максимальное количество одновременно размещаемых в нём элементов = 1 => имеем:

Значения массива могут быть размещены в следующих промежутках: [0; 430) и (583; 1917);

3. Трассировка

Данные: "Группа P3118";

Представление в формате KOI-8:

E7 D2 D5 D0 D0 C1 20 50 33 31 31 38

Представление в формате UTF-8:

D093 D180 D183 D0BF D0BF D0B0 0020 0050 0033 0031 0031 0038

Представление в формате UTF-16:

0413 0440 0443 043F 043F 0430 0020 0050 0033 0031 0031 0038

Выполняемая команда		Содержимое регистров процессора после выполнения команды.									Ячейка, содержимое которой изменилось после выполнения команды.	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адрес	Новый код
230	DE0F	240	DE0F	7FF	0231	7FF	0240	0000	004	0100	7FF	0231
240	0200	241	0200	240	0200	7FF	0240	0000	004	0100		
241	1207	242	1207	241	1207	7FF	0241	0040	004	0100		
242	2F40	243	2F40	242	0040	7FF	0040	0040	000	0000		
243	F0FC	244	F0FC	243	F0FC	7FF	0243	0040	000	0000		
244	1206	245	1206	244	1206	7FF	0244	000C	000	0000		
245	0A00	231	0A00	7FF	0231	000	0245	000C	000	0000		
231	E814	232	E814	610	000C	000	0014	000C	000	0000	610	000C
232	EA14	233	EA14	611	000C	000	0014	000C	000	0000	247	0612
											611	000C
233	F00B	234	F00B	233	F00B	000	0233	000C	000	0000		
234	DE0B	240	DE0B	7FF	0235	7FF	0240	000C	000	0000	7FF	0235
240	0200	241	0200	240	0200	7FF	0240	0000	004	0100		
241	1207	242	1207	241	1207	7FF	0241	0040	004	0100		
242	2F40	243	2F40	242	0040	7FF	0040	0040	000	0000		
243	F0FC	244	F0FC	243	F0FC	7FF	0243	0040	000	0000		
244	1206	245	1206	244	1206	7FF	0244	00E7	000	0000		
245	0A00	235	0A00	7FF	0235	000	0245	00E7	000	0000		
235	0680	236	0680	235	0680	000	0235	E700	008	1000		
236	E810	237	E810	612	E700	000	0010	E700	008	1000	612	E700
237	880E	238	880E	610	000B	000	000A	E700	008	1000	610	000B
238	CE01	23A	CE01	238	023A	000	0001	E700	008	1000		

23A	DE05	240	DE05	7FF	023B	7FF	0240	E700	008	1000	7FF	023B
240	0200	241	0200	240	0200	7FF	0240	0000	004	0100		
241	1207	242	1207	241	1207	7FF	0241	0040	004	0100		
242	2F40	243	2F40	242	0040	7FF	0040	0040	000	0000		
243	F0FC	244	F0FC	243	F0FC	7FF	0243	0040	000	0000		
244	1206	245	1206	244	1206	7FF	0244	00D2	000	0000		
245	0A00	23B	0A00	7FF	023B	000	0245	00D2	000	0000		
23B	380B	23C	380B	612	E700	000	182D	E7D2	008	1000		
23C	EA0A	23D	EA0A	612	E7D2	000	000A	E7D2	008	1000	247	0613
											612	E7D2
23D	8808	23E	8808	610	000A	000	0009	E7D2	008	1000	610	000A
23E	CEF5	234	CEF5	23E	0234	000	FFF5	E7D2	008	1000		
234	DE0B	240	DE0B	7FF	0235	7FF	0240	E7D2	008	1000	7FF	0235

4. Вывод

В ходе выполнения лабораторной работы я изучил принцип работы БЭВМ с внешними устройствами, ознакомился с различными кодировками, а также научился азам работы с ассемблерными командами.