

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

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.01

Информатика и Вычислительная Техника

(Компьютерные системы и технологии)

Дисциплина «Основы профессиональной деятельности»

Отчет

По лабораторной работе №5

Вариант №3107

Студент

**Карташев Владимир Сергеевич,
группа Р3131**

г. Санкт-Петербург, 2023 г.

Задание:

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-2
2. Программа начинается с адреса $54B_{16}$. Размещаемая строка находится по адресу 577_{16} .
3. Строка должна быть представлена в кодировке Windows-1251.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП_СИМВ.
5. Ввод или вывод строки должен быть завершён по символу с кодом 0A (NL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

Описание программы:

Передаваемое сообщение: “Ералаш”

В кодировке КОИ-8R: E5 D2 C1 CC C1 DB

В кодировке UTF-8: D095 D180 D0B0 D0BB D0B0 D188

В кодировке UTF-16: 0415 0440 0430 043B 0430 0448

В кодировке **WINDOWS-1251**: C5 F0 E0 EB E0 F8

В кодировке ISO-8859-5: B5 E0 D0 DB D0 E8

Текст программы на языке ассемблера PDP-8 (PDP-11):

ORG	0x54B	Адрес начала программы
result:	WORD 0x577	Ссылка на результат
stop:	WORD 0x0A	Стоп-слово
var:	WORD	Ячейка для временной записи нечетных символов
start:	CLA	Очистка аккумулятора
s1:	IN 5 AND #0x40 BEQ s1 IN 4 SWAB ST (result) ST var SWAB CMP stop BEQ exit CLA	Ожидание ввода нечётного символа Бит 6 SR == 0 (“Готов” нажата?) Нет – “Спин-луп” Ввод чётного символа в АС Перемещение нечётного символа в старший байт АС Сохраняем символ по адресу результата Сохраняем символ во временную ячейку Перемещение символа обратно в младший байт АС Проверяем на стоп-слово Если стоп-слово, то выходим Очистка аккумулятора
s2:	IN 5 AND #0x40 BEQ s2 IN 4 OR var ST (result) SUB var CMP stop BEQ exit LD (result)+ CLA JUMP s1	Ожидание ввода нечётного символа Бит 6 SR == 0 (“Готов” нажата?) Нет – “Спин-луп” Ввод чётного символа в АС Совмещаем с 1-м (последним нечётным) символом Сохраняем символ по адресу результата Вычитаем из АС нечётный символ Проверяем на стоп-слово Если стоп-слово, то выходим Инкремент текущего адреса результата Очистка аккумулятора Возвращаемся в начало цикла
exit:	LD (result)+ HLT	Инкремент текущего адреса результата Остановка программы

Текст исходной программы:

Адрес	Код команды	Мнемоника	Описание
54B	0577	result	Ссылка на адрес текущего результата
54C	000A	stop	Стоп-слово
54D	BEEF	var	Ячейка для временной записи нечётных символов
54E	0200	CLA	Очистка аккумулятора, старт программы
54F	1207	IN 5	Чтение регистра состояния (SR) ВУ-2
550	2F40	AND 40	Проверка на готовность ВУ (Бит 6 SR == 0?)
551	F0FD	BEQ (IP-3)	Не готов – «Спин-луп»
552	1206	IN 4	Чтение регистра данных (DR) ВУ-2 в АС
553	0680	SWAB	Обмен старшего и младшего байтов
554	E8F6	ST (IP-10)	Сохраняем символ в результат по ссылке
555	EEF7	ST IP-9	Сохраняем символ во временную ячейку
556	0680	SWAB	Обмен старшего и младшего байтов
557	7EF4	CMP IP-12	Проверяем на стоп-символ (флаги по рез-ту АС-stop)
558	F000	BEQ IP+13	Если стоп-символ (Z==1), то выход (переход в 567)
559	0200	CLA	Очистка аккумулятора
55A	1207	IN 7	Чтение регистра состояния (SR) ВУ-2
55B	2F40	AND 40	Проверка на готовность ВУ (Бит 6 SR == 0?)
55C	F0FD	BEQ IP-3	Не готов – «Спин-луп»
55D	1206	IN 6	Чтение регистра данных (DR) ВУ-2 в АС
55E	3EEE	OR IP-18	Логическое или (temp АС) -> АС
55F	E8EB	ST (IP-21)	Сохраняем символы в результат по ссылке
560	6EEC	SUB IP-20	Вычитание АС – temp -> АС
561	7EEA	CMP IP-22	Проверяем на стоп-символ (флаги по рез-ту АС-stop)
562	F003	BEQ IP+3	Если стоп-символ (Z==1), то выход (переход в 567)
563	AAE7	LD (IP-25)+	Инкрементируем ссылку на результат
564	0200	CLA	Очистка аккумулятора

565	CEE9	JUMP IP-23	Возвращаемся в начало цикла
566	AAE4	LD (IP-28)+	Инкрементируем ссылку на результат
567	0100	HLT	
577	0000	string	Строки результата, начиная с адреса 577

Описание:

Назначение программы:

Программа осуществляет посимвольный асинхронный ввод данных с ВУ-2, посимвольно записывает их в память (по 2 символа в 1 ячейке, в старшем байте нечетный символ, в младшем – чётный символ). Программа будет получать символы до тех пор, пока на ВУ-2 не будет введён стоп-символ с кодировкой 0x0A, который она запишет в память и прекратит своё выполнение.

Область представления:

result – 11-разрядная адресная ячейка со ссылкой на результат

stop – 16-разрядная постоянная из символов

var – 16-разрядная переменная, ячейка для временного хранения вводимых символов

string – 16-разрядные ячейки, хранящие в себе по 2 символа закодированного слова

Адреса первой и последней выполняемой инструкции программы:

Адрес первой команды: 54В

Адрес последней команды: 267

Область допустимых значений:

result (ссылка на адрес ячеек памяти, в которых хранятся вводимые символы) $\in [577, 7FF]$
 *меньше быть не может, т.к. в программе только увеличиваем res (инкрементируем)

var (ячейка для временной записи нечётных символов) $\in [0100_{16}, FF00_{16}]$ *т.к. в неё из АС записываются только 1 символ из 8 бит, находящийся при этом в старшем байте АС

Введённый символ $\in [00, FF]$ * т.к. состоит из 8 бит, вводится в младший байт аккумулятора

Количество введенных символов $\in [0, 1298]$ адрес первого элемента со строкой результата 577 по условию. Т.к. $7FF - 577 + 1(16 \text{ CC}) = 2047 - 1399 + 1(10 \text{ CC}) = 649$ – количество ячеек памяти, которые можно использовать для записи вводимых символов, значит $649 * 2 = 1298$ – максимально возможное количество введенных символов (т.к. в 1 ячейке могут быть максимум 2 символа)

Таблица трассировки:

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
54E	0200	54F	0200	54E	0200	000	054E	0000	0100		
54F	1205	550	1205	54F	1205	000	054F	0040	0100		
550	2F40	551	2F40	550	0040	000	0040	0040	0000		
551	F0FD	552	F0FD	551	F0FD	000	0551	0040	0000		
552	1204	553	1204	552	1204	000	0552	00C5	0000		
553	0680	554	0680	553	0680	000	0553	C500	1000		
554	E8F6	555	E8F6	577	C500	000	FFF6	C500	1000	577	C500
555	EEF7	556	EEF7	54D	C500	000	FFF7	C500	1000	54D	C500
556	0680	557	0680	556	0680	000	0556	00C5	0000		
557	7EF4	558	7EF4	54C	000A	000	FFF4	00C5	0001		
558	F000	559	F000	558	F000	000	0558	00C5	0001		
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
559	0200	55A	0200	55A	0200	000	055A	0000	0101		
55A	2F40	55B	2F40	55A	0040	000	0040	0000	0101		
55B	F000	55C	F000	55B	F000	000	0000	0000	0101		
55C	1204	55D	1204	55C	1204	000	055C	00F0	0101		
55D	3EEF	55E	3EEF	54D	C500	000	3A0F	C5F0	1001		
55E	E8EC	55F	E8EC	577	C5F0	000	FFEC	C5F0	1001	577	C5F0
55F	6EED	560	6EED	54D	C500	000	FFED	00F0	0001		
560	7EEB	561	7EEB	54C	000A	000	FFEB	00F0	0001		
561	F000	562	F000	561	F000	000	0561	00F0	0001		
562	AAE8	563	AAE8	577	C5F0	000	FFE8	C5F0	1001		
563	0200	564	0200	563	0200	000	0563	0000	0101		

564	CEEA	54F	CEEA	564	054F	000	FFEA	0000	0101		
54F	1205	550	1205	54F	1205	000	054F	0040	0101		
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
550	2F20	551	2F40	550	0040	000	0040	0040	0001		
551	F0FD	552	F0FD	551	F0FD	000	0551	0040	0001		
552	1204	553	1204	552	1204	000	0552	00E0	0001		
553	0680	554	0680	553	0680	000	0553	E000	1001		
554	E8F6	555	E8F6	578	E000	000	FFF6	E000	1001	578	E000
555	EEF7	556	EEF7	54D	E000	000	FFF7	E000	1001	54D	E000
556	0680	557	0680	556	0680	000	0556	00E0	0001		
557	7EF4	558	7EF4	54C	000A	000	FFF4	00E0	0001		
558	F000	559	F000	558	F000	000	0558	00E0	0001		
559	0200	55A	0200	559	0200	000	0559	0000	0101		
55A	2F40	55B	2F40	55A	0040	000	0040	0000	0101		
55B	F000	55C	F000	55B	F000	000	0000	0000	0101		
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
55C	1204	55D	1204	55C	1204	000	055C	00EB	0101		
55D	2EEF	55E	2EEF	54D	E000	000	1F14	E0EB	1001	54D	E0EB
55E	E8EC	55F	E8EC	578	E0EB	000	FFEC	E0EB	1001	578	E0EB
55F	6EED	560	6EED	54D	E000	000	FFED	00EB	0001		
560	7EEB	561	7EEB	54C	000A	000	FFEB	00EB	0001		
561	F000	562	F000	561	F000	000	0561	00CB	0001		
562	AAE8	563	AAE8	578	E0EB	000	FFE8	E0EB	1001		
563	0200	564	0200	563	0200	000	0563	0000	0101		
564	CEEA	54F	CEEA	564	054F	000	FFEA	0000	0101		
54F	1205	550	1205	54F	1205	000	054F	0040	0101		
550	2F20	551	2F40	550	0040	000	0040	0040	0001		
551	F0FD	552	F0FD	551	F0FD	000	0551	0040	0001		
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
552	1204	553	1204	552	1204	000	0552	00E0	0001		

553	0680	554	0680	553	0680	000	0553	E000	1001		
554	E8F6	555	E8F6	579	E000	000	FFF6	E000	1001	579	E000
555	EEF7	556	EEF7	54D	E000	000	FFF7	E000	1001	54D	E000
556	0680	557	0680	556	0680	000	0556	00E0	0001		
557	7EF4	558	7EF4	00E0	000A	000	FFF4	00E0	0001		
558	F000	559	F000	558	F000	000	0558	00E0	0001		
559	0200	55A	0200	559	0200	000	0559	0000	0101		
55A	2F40	55B	2F40	55A	0040	000	0040	0000	0101		
55B	F000	55C	F000	55B	F000	000	0000	0000	0101		
55C	1204	55D	1204	55C	1204	000	055C	00F8	0101		
55D	3EEF	55E	3EEF	54D	E000	000	1F07	E0F8	1001		
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
55E	E8EC	55F	E8EC	579	E0F8	000	FFEC	E0F8	1001	579	E0F8
55F	6EED	560	6EED	54D	E000	000	FFED	00F8	0001		
560	8EEB	561	8EEB	54C	000A	000	AAEB	00F8	0001		
561	F000	562	F000	561	F000	000	0561	00F8	0001		
562	AAE8	563	AAE8	579	E0F8	000	FFE8	E0F8	1001		
563	0200	564	0200	563	0200	000	0563	0000	0101		
564	CEEA	54F	CEEA	564	054F	000	FFEA	0000	0101		
54F	1205	550	1205	54F	1205	000	054F	0040	0101		
550	2F40	551	2F40	550	0040	000	0040	0040	0001		
551	F0FD	552	F0FD	551	F0FD	000	0551	0040	0001		
552	1204	553	1204	552	1204	000	0552	000A	0001		
553	0680	554	0680	553	0680	000	0553	0A00	0001		
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
554	E8F6	555	E8F6	57A	0A00	000	FFF6	0A00	0001	57A	0A00
555	EEA8	556	EEA8	54D	0A00	000	FFF8	000A	0001	54C	000A
567	0100	568	0100	567	0100	000	FFF9	000A	0001		

