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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5 по дисциплине «Основы профессиональной деятельности»

Вариант 10035

Выполнил: Студент группы Р3112 Медведев Ярослав Александрович Преподаватель: Блохина Елена Николаевна

г. Санкт-Петербург 2024

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.

Введите номер варианта 10035

- 1. Программа осуществляет асинхронный ввод данных с ВУ-2
- 2. Программа начинается с адреса $3A1_{16}$. Размещаемая строка находится по адресу $60E_{16}$.
- 3. Строка 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 00 (NUL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

Слово: Йогурт:

ISO-8859-5: B9, DE, D3, E3, E0, E2, 3A

UTF-16: 0419, 043E, 0433, 0443, 0440, 0442, 003A UTF-8: D099, D0BE, D0B3, D183, D180, D182, 003A

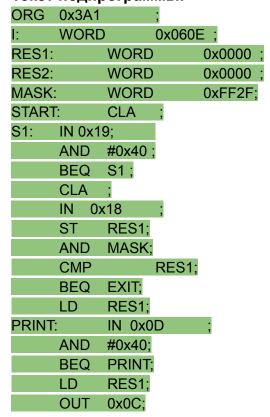
Доп:

Ввод осуществляется с клавиатуры (ВУ-8) вывод осуществляется на текстовый принтер Выводятся только нечетные символы все остальное согласно первоначальному заданию

XXXX XXXX XXXX XXXX

СТОП-символ (если это будет необходимо) можно поменять (стоп-символ - /)

Текст подпрограммы:



```
S2: IN 0x19;
    AND #0x40;
    BEQ S2;
    CLA ;
    IN 0x18 ;
    ST RES2;
    LD RES1;
    SWAB ;
    OR RES2;
    ST
        RES1;
    AND MASK;
    CMP RES1;
    BEQ EXIT;
    LD
        RES1;
    ST (I)+;
   JUMP START;
EXIT: LD RES1;
   ST (I)+;
    HLT ;
```

Ход работы

Текст исходной программы

текст исходной программы							
Адрес	Код команды	Мнемоника	Комментарии				
3A5	0200	CLA	0 -> AC				
3A6	1205	IN 5	5 -> МЛ. БАЙТ АС				
3A7	2F40	AND #40	40 & AC -> AC				
3A8	F0FE	BEQ (IP - 2)	Переход если равенство (Z==1)				
3A9	0200	CLA	0 -> AC				
ЗАА	1204	IN 4	4 -> МЛ. БАЙТ АС				
ЗАВ	EEF7	ST (IP - 9)	AC -> (IP - 9)				
3AC	2EF8	AND (IP - 8)	(IP - 8) & AC -> AC				
3AD	7EF5	CMP (IP - 11)	УСТАНОВИТЬ ФЛАГИ ПО РЕЗУЛЬТАТУ АС - (IP - 11)				
3AE	F011	BEQ (IP + 17)	Переход если равенство (Z==1)				
3AF	1205	IN 5	5 -> МЛ. БАЙТ АС				
3B0	2F40	AND #40	40 & AC -> AC				
3B1	F0FE	BEQ (IP - 2)	Переход если равенство (Z==1)				
3B2	0200	CLA	0 -> AC				
3B3	1204	IN 4	4 -> МЛ. БАЙТ АС				
3B4	EEEF	ST (IP - 17)	AC -> (IP - 17)				
3B5	AEED	LD (IP - 19)	(IP - 19) -> AC				
3B6	0680	SWAB	AC0AC7 <-> AC8AC15				
3B7	3EEE	OR (IP - 20)	(IP - 20) AC -> AC				

3B8	EEEA	ST (IP - 22)	AC -> (IP - 22)		
3B9	2EEB	AND (IP - 21)	(IP - 21) & AC -> AC		
ЗВА	7EE8	CMP (IP - 24)	УСТАНОВИТЬ ФЛАГИ ПО РЕЗУЛЬТАТУ АС - (IP - 24)		
ЗВВ	F004	BEQ (IP + 4)	Переход если равенство (Z==1)		
3BC	AEF6	LD (IP - 26)	(IP - 26) -> AC		
3BD	EAE4	ST (IP - 28)+	AC -> (IP - 28)+		
3BE	CEE9	JUMP (IP - 25)	(IP - 25) -> IP		
3BF	AEE3	LD (IP - 29)	(IP - 29) -> AC		
3C0	EAF1	ST (IP - 31)+	AC -> (IP - 31)+		
3C1	0100	HLT	ОСТАНОВ		

Текст программы на ассемблере

ORG 0x3A1 ;

I: WORD 0x060E; Ссылка на ячейку для записи информации

RES1: WORD 0x0000 ; Первый введенный символ/конечный результат

RES2: WORD 0x0000 ; Второй введенный символ

MASK: WORD 0xFF00; Маска для проверки на стоп-символ

START: CLA ; Очистка аккумулятор

S1: IN 5 ; Получение данных из регистра состояния ВУ-2

AND #0x40; Проверка на наличие введенного символа

BEQ S1; Нет - "Спин-луп"

CLA ; Очистка аккумулятора

IN 4 ; Получение данных из дата регистра ВУ-2

ST RES1; Сохранение первой буквы в RES1

AND MASK; Применение маски

СМР RES1; Проверка на стоп-символ

BEQ EXIT; Да - переход к завершению программы

S2: IN 5 ; Получение данных из регистра состояния ВУ-2

AND #0x40 ; Проверка на наличие введенного символа

BEQ S2; Heт - "Спин-луп"

CLA ; Очистка аккумулятора

IN 4 ; Получение данных из дата регистра ВУ-2

ST RES2; Сохранение второй буквы в RES2

LD RES1; Загрузка первой буквы

SWAB ;

OR RES2; Подготовка данных для записи

ST RES1; Сохранение конечного результата в RES1

AND MASK; Применение маски

CMP RES1; Проверка на стоп-символ

BEQ EXIT; Да - переход к завершению программы

LD RES1;

ST (I)+; Загрузка в память конечного результата JUMP START; Переход к началу программы

EXIT: LD RES1;

ST (I)+; Сохранение в память конечного результата

HLT ; Прекращение работы программы

Описание программы

Программа осуществляет асинхронный ввод данных с ВУ-2, предоставляя строку в кодировке ISO-8859-5 и записываю ее в память в формате: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП_СИМВ. Концом передачи информации служит символ 00.

Область представления данных

I - знаковое 16-разрядное число, диапазон - [-2^15, 2^15 - 1].

RES1 - беззнаковое 16-разрядное число, диапазон - [2^15, 2^15 - 1].

RES2 - знаковое 16-разрядное число, диапазон - [-2^15, 2^15 - 1].

MASK - знаковое 16-разрядное число, диапазон - [-2^15, 2^15 - 1].

Область допустимых значений

I - [f60E; 07FF];

RES1 - [0000, FFFF];

RES2 - [00, FF];

MASK - константа (FF00).

Расположение в памяти ЭВМ

Расположение программы - [3A5, 3BF].

3А1 - исходная переменная I - адрес ячейки для записи символов.

3А2 - переменная, хранящая первую букву и конечный результат.

3А3 - переменная, хранящая вторую букву.

3А4 - исходная константа, выполняющая роль маски.

Программа:

Адрес первой выполняемой команды - 3A5, адрес последней выполняемой команды - 3BF.

Трассировка

Выполняем ая команда	Содержимое регистров процессора после выполнения команды содере кото измень постоя выполя ком
-------------------------	---

Адр ес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адр ес	Нов ый код
3A5	0200	3A6	0200	3A5	0200	0000	03A5	0000	0100	-	-
3A6	1205	3A7	1205	3A6	1205	0000	03A6	0040	0100	-	-
3A7	2F40	3A8	2F40	3A7	0040	0000	0040	0040	0000	-	-
3A8	F0FD	3A9	F0FD	3A8	F0FD	0000	03A8	0040	0000	-	-
3A9	0200	3AA	0200	3A9	0200	0000	03A9	0000	0100	-	-
3AA	1204	3AB	1204	3AA	1204	0000	03AA	00B9	0100	-	-
3AB	EEF6	3AC	EEF6	3A2	00B9	0000	FFF6	00B9	0100	3A2	00B9
3AC	2EF7	3AD	2EF7	3A4	FF00	0000	FFF7	0000	0100	-	-
3AD	7EF4	3AE	7EF4	3A2	00B9	0000	FFF4	0000	1000	-	-
3AE	F010	3AF	F010	3AE	F010	0000	03AE	0000	1000	-	-
3AF	1205	3B0	1205	3AF	1205	0000	03AF	0000	1000	-	-
3B0	2F40	3B1	2F40	3B0	0040	0000	0040	0000	0100	-	-
3B1	F0FD	3AF	F0FD	3B1	F0FD	0000	FFFD	0000	0100	-	-
3AF	1205	3B0	1205	3AF	1205	0000	03AF	0000	0100	-	-
3B0	2F40	3B0	0000	000	0000	0000	0000	0000	0100	-	-
3B0	2F40	3B1	2F40	3B0	0040	0000	0040	0000	0100	-	-
3B1	F0FD	3AF	F0FD	3B1	F0FD	0000	FFFD	0000	0100	-	-
3AF	1205	3B0	1205	3AF	1205	0000	03AF	0040	0100	-	-
3B0	2F40	3B1	2F40	3B0	0040	0000	0040	0040	0000	-	-
3B1	F0FD	3B2	F0FD	3B1	F0FD	0000	03B1	0040	0000	-	-
3B2	0200	3B3	0200	3B2	0200	0000	03B2	0000	0100	-	-
3B3	1204	3B4	1204	3B3	1204	0000	03B3	00DE	0100	-	-
3B4	EEEE	3B5	EEEE	3A3	00DE	0000	FFEE	00DE	0100	3A3	00DE
3B5	AEEC	3B6	AEEC	3A2	00B9	0000	FFEC	00B9	0000	-	-
3B6	0680	3B7	0680	3B6	0680	0000	03B6	B900	1000	-	-
3B7	3EEB	3B8	3EEB	3A3	00DE	0000	4641	B9DE	1000	-	-
3B8	EEE9	3B9	EEE9	3A2	B9DE	0000	FFE9	B9DE	1000	3A2	B9DE
3B9	2EEA	3BA	2EEA	3A4	FF00	0000	FFEA	B900	1000	-	-
3ВА	7EE7	3BB	7EE7	3A2	B9DE	0000	FFE7	B900	1000	-	-
3BB	F003	3BC	F003	3BB	F003	0000	03BB	B900	1000	-	-
3BC	AEE5	3BC	0000	000	0000	0000	0000	0000	0100	-	-
3BC	AEE5	3BD	AEE5	3A2	B9DE	0000	FFE5	B9DE	1000	-	-
3BD	EAE3	3BE	EAE3	60E	B9DE	0000	FFE3	B9DE	1000	3A1	060F
										60E	B9DE

Вывод

В ходе выполнения лабораторной работы я познакомился с асинхронным вводом-выводом в БЭВМ, более подробно рассмотрел взаимодействие с ВУ-2 и научился писать программы ввода-вывода на языке Ассемблера БЭВМ.