

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ СИСТЕМНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5
курса «Основы профессиональной деятельности»
по теме: «Исследование работы БЭВМ: асинхронный ввод/вывод
с помощью прерываний»
Вариант № 1007

Выполнил студент:
Тюрин Иван Николаевич
группа: Р3110

Преподаватель:
Клименков С. В.,
Ларочкин Г. И.

Санкт-Петербург, 2022 г.

Содержание

Лабораторная работа № 5. Исследование работы БЭВМ: асинхронный ввод/вывод с помощью прерываний	2
1. Задание варианта № 1007	2
2. Описание программы	3
1. Назначение программы	3
2. Область допустимых значений	4
3. Методика проверки	5
4. Программа	6
5. Вывод	6

Лабораторная работа № 5

Исследование работы БЭВМ: асинхронный ввод/вывод с помощью прерываний

1. Задание варианта № 1007

, , ,

По выданному преподавателем варианту разработать и исследовать работу комплекса программ обмена данными в режиме прерывания программы. Основная программа должна изменять содержимое заданной ячейки памяти (X), которое должно быть представлено как знаковое число. Область допустимых значений изменения X должна быть ограничена заданной функцией $F(X)$ и конструктивными особенностями регистра данных ВУ (8-ми битное знаковое представление). Программа обработки прерывания должна выводить на ВУ модифицированное значение X в соответствии с вариантом задания, а также игнорировать все необработываемые прерывания.

1. Основная программа должна уменьшать на 2 содержимое X (ячейки памяти с адресом 015_{16}) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности. ВУ-3 осуществлять вывод результата вычисления функции $F(X) = 3X + 1$ на данное ВУ, а по нажатию кнопки готовности ВУ-2 прибавить утроенное содержимое РД данного ВУ к X , результат записать в X .
3. Если оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в записать максимальное по ОДЗ число.

, , ,

2. Описание программы

2. 1. Назначение программы

Описание программы представлено в таблице 1.1. Программа осуществляет асинхронный ввод с ВУ-2 и вывод на ВУ-3 при помощи прерываний. ВУ-2 производит прерывание при выставлении готовности кнопкой, после чего над переменной в программе производятся определенные операции. ВУ-3 производит прерывание при выставлении готовности кнопкой, после чего над переменной производятся операции и результат выводится на табло этого устройства.

Адрес	Метка	Мнемоника	Описание
0x000		org 0x000	Инициализация вектора прерываний
0x001	v0:	word \$default, 0x180	
0x002	v1:	word \$default, 0x180	
0x003	v2:	word \$int2, 0x180	
0x004	v3:	word \$int3, 0x180	
0x005	v4:	word \$default, 0x180	
0x006	v5:	word \$default, 0x180	
0x007	v6:	word \$default, 0x180	
0x008	v7:	word \$default, 0x180	
0x015	x:	og 0x015 word ?	Параметры программы Переменная (X)
0x016	uplim:	word 0x002a	Верхняя граница ОДЗ
0x017	lowlim:	word 0xffd5	Нижняя граница ОДЗ
0x018	default:	iret	Стандарт. обработка прер.
0x019	int2:	nop	Обработчик прерывания ВУ-2, точка для останова Считывание из ВУ-2 Прибавление утроенного значания РД ВУ-2
0x01A		cla	
0x01B		in 0x4	
0x01C		sxtb	
0x01D		push	
0x01E		asl	
0x01F		add &0	
0x020		add x	
0x021		call _check	
0x022		st x	
0x023		pop	
0x024		ld x	
0x025		nop	
0x026		iret	
0x027	int3:	nop	Обработчик прерывания ВУ-3, точка для останова Арифметическая операция $3x+1$
0x028		ld x	
0x029		asl	
0x02A		add x	

0x02B		inc	Вывод на ВУ-3 Точка для останова
0x02C		out 0x6	
0x02D		nop	
0x02E		iret	
0x02F	_check:	cmp uplim	Подпрограмма для проверки выхода из ОДЗ
0x030		bpl _setmax	
0x031		cmp lowlim	
0x032		bmi _setmax	
0x033	_retcheck:	ret	
0x034	_setmax:	ld uplim	
0x035		jump _retcheck	
0x036	START:	di	Начало программы, инициализация векторов прерывания
0x037		cla	
0x038		out 0x1	
0x039		out 0x3	
0x03A		out 0xb	
0x03B		out 0xf	
0x03C		out 0x13	
0x03D		out 0x17	
0x03E		out 0x1b	
0x03F		out 0x1f	
0x040		ld #0xa	
0x041		out 0x5	
0x042		ld #0xb	
0x043		out 0x7	
0x044		ei	
0x045	main:	di	Главный цикл Уменьшение переменной на 2
0x046		ld x	
0x047		sub #2	
0x048		call _check	
0x049		st x	
0x04A		ei	
0x04B		nop	
0x04C		jump main	

Таблица 1.1: Описание работы программ

2. 2. Область допустимых значений

Пользователю доступны для ввода любые однобайтовые знаковые числа, то есть из диапазона $[-128; 127]$. ОДЗ для переменной X находится из соотношения:

$$\begin{aligned}
 -128 \leq 3x + 1 \leq 127 &\Rightarrow \\
 0xffd5 = -43 \leq x \leq 42 = 0x002a
 \end{aligned}$$

3. Методика проверки

Главный цикл

1. В переменную `x` записать `0xffd8`.
2. Поставить точку останова в цикле `main` на адрес `0x04b`.
3. Загрузить программный комплекс в память БЭВМ.
4. Запустить выполнение программы в автоматическом режиме с адреса `0x036` до остановки.
5. Записать значение аккумулятора, оно должно быть на 2 меньше, чем число которое мы записали в переменную.
6. Запустить выполнение программы с адреса предыдущей остановки до следующей остановки.
7. Записать значение аккумулятора и сверить его с предполагаемым: оно должно быть равно верхней границе ОДЗ `0x002a` (т.к. на новом витке цикла переменная вышла из ОДЗ).

Обработчик прерывания ВУ-2

1. Убрать прежние и поставить точки останова в подпрограмме `int2` на адресах `0x019` и `0x025`.
2. Загрузить программный комплекс в память БЭВМ.
3. Запустить выполнение программы в автоматическом режиме с адреса `0x036`.
4. Установить значение и выставить режим готовности на ВУ-2.
5. Выполнять программу до точки останова.
6. Записать значение аккумулятора.
7. Выполнить программу до следующей точки останова.
8. Записать значение аккумулятора.
9. Рассчитать значение переменной по формуле $x + 3y$, где y - число введенное на ВУ-2, если вычисленное значение не находится в ОДЗ, то должно быть равно `0x02a`.
10. Сравнить полученные значения аккумулятора: второе должно удовлетворять расчетному.

Обработчик прерывания ВУ-3

1. Убрать прежние и поставить точки останова в подпрограмме `int3` на адресах `0x027` и `0x02d`.
2. Загрузить программный комплекс в память БЭВМ.
3. Запустить выполнение программы в автоматическом режиме с адреса `0x036`.
4. Выставить режим готовности на ВУ-3.
5. Выполнять программу до точки останова.
6. Записать значение аккумулятора.
7. Выполнить программу до следующей точки останова.
8. Записать значение аккумулятора.
9. Рассчитать значение переменной по формуле $3x + 1$, если вычисленное значение не находится в ОДЗ, то должно быть равно `0x02a`.
10. Сравнить полученные значения аккумулятора: второе должно удовлетворять расчетному.

4. Программа

Была составлена программа на языке ассемблера. Она представлена в листингах [1.2](#) и [1.4](#).

5. Вывод

Научился работать с прерываниями в БЭВМ, писать код на языке ассемблера БЭВМ для их обработки. Научился делать одновременно несколько важных лаб за ночь и терпению в очереди за день.

```

1 org 0x000
2 v0:      word    $default ,    0x180
3 v1:      word    $default ,    0x180
4 v2:      word    $int2 ,      0x180
5 v3:      word    $int3 ,      0x180
6 v4:      word    $default ,    0x180
7 v5:      word    $default ,    0x180
8 v6:      word    $default ,    0x180
9 v7:      word    $default ,    0x180
10 org 0x015
11 x:       word    ?
12 uplim:   word    0x002a
13 lowlim:  word    0xffd5
14 default:
15         iret
16
17 int2:
18     nop                ; debug
19     cla
20     in      0x4        ; dev-2
21         sxtb
22         push
23         asl
24         add      &0
25     add      x
26     call    _check
27     st      x
28         pop
29     ld      x          ; to see in AC
30     nop                ; debug
31     iret
32
33 int3:
34     nop                ; debug
35     ld      x
36     asl                ; f(x)=3x+1
37     add      x
38         inc
39     out      0x6        ; dev-3
40     nop                ; debug
41     iret

```

Листинг 1.2: Первая часть кода программы на языке ассемблера БЭВМ: параметры программы, переменные, инициализация векторов прерывания, обработчики прерывания


```

1  _check:
2      cmp uplim
3      bpl _setmax
4      cmp lowlim
5      bmi _setmax
6  _retcheck:
7      ret
8  _setmax:
9      ld uplim
10     jump _retcheck
11
12 START:
13     di
14     cla
15     out    0x1      ;deny interruptions
16     out    0x3
17     out    0xb
18     out    0xf
19     out    0x13
20     out    0x17
21     out    0x1b
22     out    0x1f
23     ld     #0xa     ;mr dev-2 (1000v0010)
24     out    0x5
25     ld     #0xb     ;mr dev-3 (1000v0011)
26     out    0x7
27     ei
28
29 main:
30     di
31     ld     x
32     sub    #2
33     call   _check
34     st     x
35     ei
36     nop
37     jump  main

```

Листинг 1.4: Вторая часть кода программы на языке ассемблена БЭВМ: подпрограмма проверки выхода результата вычисления из ОДЗ и главный цикл