Национальная научно-образовательная корпорация ИТМО

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине

«ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

Вариант № 1160

Выполнил:

Студент группы P3111

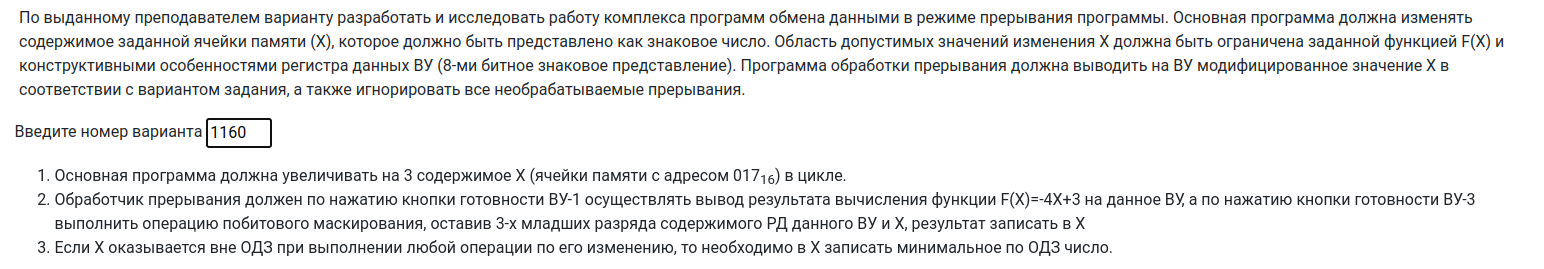
Наземцев Сергей Дмитриевич

Преподаватель:

Ткешелашвили Нино Мерабиевна

Санкт-Петербург, 2023

Задание



Выполнение

ORG 0x0017

X: WORD ?

MIN: WORD 0xFFE1 ; левая граница ОДЗ = -31

MAX: WORD 0x0020 ; правая граница ОДЗ = 32

ORG 0x0 ; Инициализация векторов прерывания

V0: WORD $DEFAULT,0x180 ; Вектор прерывания #0

V1: WORD $INT1,0x180 ; Вектор прерывания #1

V2: WORD $DEFAULT,0x180 ; Вектор прерывания #2

V3: WORD $INT3,0x180 ; Вектор прерывания #3

V4: WORD $DEFAULT,0x180 ; Вектор прерывания #4

V5: WORD $DEFAULT,0x180 ; Вектор прерывания #5

V6: WORD $DEFAULT,0x180 ; Вектор прерывания #6

V7: WORD $DEFAULT,0x180 ; Вектор прерывания #7

DEFAULT:

IN 4

IN 0x18

IN 0x1C

IRET ; просто возврат

ORG 0x0F0 ; загрузка начальных векторов прерывания

START: DI ; запрет прерываний

CLA

LD #8

OUT 0x5 ; MR КВУ-2 на вектор 0

OUT 0x19 ; MR КВУ-8 на вектор 0

OUT 0x1D ; MR КВУ-9 на вектор 0

LD #0

OUT 0x1 ; MR КВУ-0 на вектор 0

OUT 0xB ; MR КВУ-4 на вектор 0

OUT 0xD ; MR КВУ-5 на вектор 0

OUT 0x11 ; MR КВУ-6 на вектор 0

OUT 0x15 ; MR КВУ-7 на вектор 0

LD #0x9 ; разрешить прерывания и вектор №1

OUT 3 ;

LD #0xB ; разрешить прерывания и вектор №2

OUT 7 ;

JUMP $PROG

PROG: EI ; разрешили прерывания

CLA

SPINLOOP: DI

LD $X

NOP

ADD #0x3 ; увеличили на 3

CMP $MIN ; сравнили с

BLT CHECK ; левой границей

CMP $MAX ; сравнили с

BGE CHECK ; правой границей

ST $X ; сохранили Х

EI

JUMP SPINLOOP ; продолжили бесконечный цикл

CHECK: LD $MIN ; загрузили левую границу ОДЗ

ST $X ; сохранили Х

JUMP SPINLOOP ; продолжили бесконечный цикл

INT1:

LD $X ; загрузили Х

NOP

ASL ; умножили на 2

ASL ; умножили на 4

NEG ; умножили на -4

ADD #3

OUT 2 ; вывели на ВУ-1

NOP ; проверка работоспособности ВУ-1

IRET ; возврат из обработки прерывания

INT3: NOP

IN 6 ; загрузили РДВУ-3

AND #0x07

AND $X

CMP $MIN ; сравнили с

BLT CHECK1 ; левой границей

CMP $MAX ; сравнили с

BGE CHECK1 ; правой границей

ST $X ; сохранили Х

NOP ; проверка работоспособности ВУ-3

IRET ; возврат из обработки прерывания

CHECK1: LD $MIN ; загрузили левую границу ОДЗ

ST $X ; сохранили Х

NOP ; проверка работоспособности ВУ-3

IRET ; возврат из обработки прерывания

# **Назначение программы**

По команде готовности ВУ-1 вычисляет результат функции y = -4x+3 и выводит результат на ВУ-1

По команде готовности ВУ-3 выполняет операцию побитового маскирования, оставив 3 младших разряда содержимого РД данного ВУ и Х, и сохраняет результат в переменную Х.

# **Расположение в памяти БЭВМ данных программы:**

0x0-0xF – векторы прерываний

0x17 - 0x19 – данные программы (переменные)

0x0F0 - 0x0FF – инициализация векторов прерывания

0x100 - 0x10E – основная программа

0x10F - 0x117 – обработка прерывания от ВУ-1

0x118 - 0x123 – обработка прерывания от ВУ-3

0x124 - 0x127 – проверка на ОДЗ

# **Область представления:**

X, left, right - знаковые 16-разрядные целые числа

# **Область допустимых значений:**

-128 <= -4\*x + 3 <= 127

-131 <= -4x <= 124

-31 <= x <= 32

# **Методика проверки программы:**

1. Загрузить текст программы в БЭВМ.
2. Заменить NOP на HLT в той части программы, которую хотите проверить (то есть в метке SPINLOOP для основной программы, в метке INT1 для теста ВУ-1 и в метке INT3 для теста ВУ-3, можно заменить сразу везде для теста всего кода).
3. Скомпилировать программу
4. Запустить программу в режиме РАБОТА

Для проверки основной программы:

1. Дождаться останова
2. Записать значение аккумулятора (1)
3. Нажать «Продолжение»
4. Дождаться останова и записать значение аккумулятора (2)
5. Из (2) вычесть (1) и посмотреть на разницу. Если она равна 3 в 10СС, то программа выполнилась корректно
6. Для дополнительной проверки крайнего случая с ОДЗ: продолжать идти по пунктам 1-5 до момента, пока число в АС не превысит максимально допустимое, пронаблюдать изменение аккумулятора в этот момент, записать находящееся в нем значение, сравнить с минимальным по ОДЗ числом.

Для проверки ВУ-1:

1. Открыть КВУ-1 и установить готовность
2. Дождаться останова
3. Записать значение АС (переменную Х)
4. Нажать «Продолжение»
5. Дождаться останова
6. Записать значение АС (результат выполнения функции -4\*x+3)
7. Рассчитать ожидаемое значение функции для текущего Х и сверить правильность
8. В случае успеха нажать «Продолжение»

Для проверки ВУ-3:

1. Открыть КВУ-3 и ввести туда какое-нибудь число
2. Установить готовность ВУ-3
3. Дождаться останова
4. Записать значение АС (переменная Х)
5. Нажать «Продолжение»
6. Дождаться останова
7. Записать значение АС (результат выполнения функции DR & 0x7 & X)
8. Рассчитать ожидаемое значение и сверить с фактическим

|  |  |  |
| --- | --- | --- |
| Основная программа | | |
| Старое значение AC(0..7) | Ожидаемое значение | Новое значение AC(0..7) |
| 0x3 (3) | 0x6(6) | 0x6(6) |
| 0xF (15) | 0x12 (18) | 0x12 (18) |
| 0x21 (33) | 0xFFE1 (-31) | 0xFFE1 (-31) |

|  |  |  |
| --- | --- | --- |
| Проверка ВУ-1 | | |
| Значение AC (0..7) | Ожидаемое значение -4x+3 | Результат АС (0..7) |
| 0x9 (9) | 0xFFDF (-33) | 0xFFDF |
| 0x3 (3) | 0xFFF7 (-9) | 0xFFF7 (-9) |
| 0x12 (18) | 0xFFBB (-69) | 0xFFBB (-69) |

|  |  |  |
| --- | --- | --- |
| Проверка ВУ-3 | | |
| Значение АС(0..7) | DR КВУ-3 | AC (DR & 0x7 & X) |
| 0x9 (0000 1001) | 0x88 (136) (1000 1000) | 0x0 |
| 0x9 (0000 1001) | 0x89 (137) (1000 1001) | 0x1 (1) |
| 0x1B (0001 1011) | 0xAF (1010 1111) | 0x3 (0000 0011) |

**Вывод**

В ходе выполнения лабораторной работы я попрактиковался в работе с различными ВУ, изучил обмен с прерываниями на ВУ-1 и ВУ-3, изучил циклы прерывания, а также потренировался в написании программ на Ассемблере.