МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №6**

по дисциплине

«ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

Вариант № 65412

***Выполнил:***

Студент группы P3118

Рамеев Тимур

Ильгизович

***Преподаватель:***

Осипов Святослав

Владимирович

**Содержание**

[Задание 3](#_Toc137777552)

[Выполнение работы 4](#_Toc137777553)

[Определение текста программы 4](#_Toc137777554)

[Описание программы 6](#_Toc137777555)

[Трассировка 7](#_Toc137777556)

[Вывод 8](#_Toc137777557)

# Задание

По выданному преподавателем варианту разработать и исследовать работу комплекса программ обмена данными в режиме прерывания программы. Основная программа должна изменять содержимое заданной ячейки памяти (Х), которое должно быть представлено как знаковое число. Область допустимых значений изменения Х 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 Х в соответствии с вариантом задания, а также игнорировать все необрабатываемые прерывания.

1. Основная программа должна уменьшать на 3 содержимое X (ячейки памяти с адресом 03B16) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=6X+2 на данное ВУ, a по нажатию кнопки готовности ВУ-2 вычесть содержимое РД данного ВУ из Х, результат записать в X
3. Если Х оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в Х записать максимальное по ОДЗ число.

# Выполнение работы

## Определение текста программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды/Данные** | **Мнемоника** | **Комментарий** |
| 000 | 0100 | WORD $EXDEVICE1 | Адрес подпрограммы для обработки прерывания с ВУ-1 |
| 001 | 0180 | WORD 0x180 | Регистр PS для проги выше |
| 002 | 0100 | WORD $EXDEVICE2 | Адрес подпрограммы для обработки прерывания с ВУ-2 |
| 003 | 0180 | WORD 0x180 | Регистр PS для проги выше |
|  |  |  |  |
| 03B | 0000 | X: WORD 10 | Ячейка памяти (X) |
| 03C | FFFD | DECREMENT: WORD -3 | Декремент для основной программы |
|  |  |  |  |
| 010 | 0200 | START: CLA | Блок для запрета |
| 011 | 1301 | OUT 0x1 | вызова прерываний для всех |
| 012 | 1303 | OUT 0x3 | ВУ |
| 013 | 1305 | OUT 0x5 | (вдруг моя программа – |
| 014 | 1307 | OUT 0x7 | это подпрограмма другой |
| 015 | 130B | OUT 0xB | программы) |
| 016 | 130E | OUT 0xE |  |
| 017 | 1312 | OUT 0x12 |  |
| 018 | 1316 | OUT 0x16 |  |
| 019 | 131A | OUT 0x1A |  |
| 01A | 131E | OUT 0x1E |  |
| 01B | AF08 | LD #0x8 | Говорим, что нулевой вектор прерывания будет для ВУ-1 |
| 01C | 1303 | OUT 0x3 |  |
| 01D | AF09 | LD #0x9 | Говорим, что 1-й вектор прерывания будет для ВУ-2 |
| 01E | 1305 | OUT 0x5 |  |
| 01F | 1000 | MAIN: DI | Начало главной подпрограммы / запрещаем прерывание |
| 020 | AE1A | LD X |  |
| 021 | 4E1A | ADD DECREMENT | Отнимаем 3 |
| 022 | 7FEB | CMP #-21 | Проверяем на минимальное значение |
| 023 | F301 | BNC PROVERKA1 |  |
| 024 | AF14 | LD #20 | Максимальное число по ОДЗ |
| 025 | EE15 | PROVERKA1: ST X |  |
| 026 | 1100 | EI | Разрешаем прерывание |
| 027 | CEF7 | JUMP MAIN | Спин луп |
| 028 | 0100 | HLT |  |
|  |  |  |  |
| 100 | 0C00 | EXDEVICE1: PUSH | Первая подпрограмма для ВУ-1 / сохраняем текущий аккумулятор в стеке |
| 101 | A03B | LD $X | Выполняем функцию |
| 102 | 0500 | ASL | F(x) = 6x + 2 |
| 103 | 0500 | ASL |  |
| 104 | 403B | ADD $X |  |
| 105 | 403B | ADD $X |  |
| 106 | 4F02 | ADD #2 |  |
| 107 | 1302 | OUT 0x2 | Выводим значение в ВУ-1 |
| 108 | 0800 | POP | Возвращаем аккумулятор из стека |
| 109 | 0B00 | IRET | Возврат из подпрограммы |
|  |  |  |  |
| 200 | 0C00 | EXDEVICE2: PUSH | Вторая подпрограмма для ВУ-2 / сохраняем текущий аккумулятор в стеке |
| 201 | 0200 | CLA |  |
| 202 | 1204 | IN 0x4 | Получаем данные из ВУ-2 |
| 203 | 0780 | NEG | Выполняем функцию |
| 204 | 403B | ADD $X | X = X – DR(ВУ-2) |
| 205 | 7FE5 | CMP #-27 | Проверяем на минимальное значение |
| 206 | F300 | BNC PROVERKA2 |  |
| 207 | AF14 | PROVERKA2: LD #20 | Максимальное число по ОДЗ |
| 208 | E03B | ST $X | Сохраняем результат |
| 209 | 0800 | POP | Возвращаем аккумулятор из стека |
| 20A | 0B00 | IRET | Возврат из подпрограммы |

## Описание программы

1. Исходные данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес в памяти | Назначение, описание | Присвоенное обозначение | Область представления | Область допустимых значений |
| 03B | Переменная программы | X | Знаковые числа длиной 1 байт | -21 ≤ X ≤ 20 |

1. Назначение программы и функция:

Основная программа должна уменьшать на 3 содержимое X (ячейки памяти с адресом 03B16) в цикле.

Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=6X+2 на данное ВУ, a по нажатию кнопки готовности ВУ-2 вычесть содержимое РД данного ВУ из Х, результат записать в X

1. Адрес первой исполняемой команды – 0x10, адрес последней исполняемой команды – 0x28. (без учета подпрограмм)
2. Вычисление ОДЗ для X:
   1. Так как у нас есть 8 бит один из которых под знак, поэтому имеем промежуток [-128 : 127]
   2. Это будет областью значения для функции f(x) = 6x + 2, потому что именно она выводится ВУ-2
   3. -128 ≤ 6x + 2 ≤ 127

-130 ≤ 6x ≤ 125

-21 ≤ x ≤ 20

## Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров после выполнения программы** | | | | | | | | **Изменения в памяти** | |
| Адрес | Значение | IP | CR | AR | DR | SP | BR | AC | NZVC | Адр | Знчн |

# Вывод

При выполнении работы я познакомился с вводом-выводом по прерыванию. Я написал программу, реализующую прерывание от двух ВУ.