**Национальный исследовательский университет компьютерных технологий, механики и оптики**

**Факультет ПИиКТ**

**Лабораторная работа №6**

**«Асинхронный обмен данными с ВУ»**

Вариант №3072

Работу выполнил: Кулаков Никита

Группа: P3130

Преподаватель: Перминов И. В.

Город: Санкт-Петербург

2021 год

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.

Текст варианта:

1) Основная программа должна увеличивать на 3 содержимое X (ячейки памяти с адресом 03F16) в цикле.

2) Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=-2X-1 на данное ВУ, a по нажатию кнопки готовности ВУ-3 вычесть утроенное 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 минимальное по ОДЗ число.

Описание программы:

|  |  |
| --- | --- |
| Назначение программы и реализуемая ею функция (формула): | Значение увеличивается в цикле на 3.  ВУ-1: , print()  ВУ-3:  Если число не входит в ОДЗ, то |
| Область представления: | Числа представлены в дополнительным коде:  16-разрядов, где 1 старший бит выделяется под знак |
| Область допустимых значений: | [-64;63] – Ограничение от функции .  [-128;127] – ОДЗ для .  [-128;127] – ВУ-3 DR.  Остальные числа представлены в ДК. |
| Расположение в память ЭВМ программы, исходных данных и результата: | 0x716 – переменная X. |
| Адреса первой и последней выполняемой команд программы: | 0x020 – начало.  Конец на любом HLT |

Программа на языке Ассемблер:

ORG 0x0

V0: WORD $DEFAULT, 0x180

V1: WORD $INT1, 0x180

V2: WORD $DEFAULT, 0x180

V3: WORD $INT3, 0x180

V4: WORD $DEFAULT, 0x180

V5: WORD $DEFAULT, 0x180

V6: WORD $DEFAULT, 0x180

V7: WORD $DEFAULT, 0x180

DEFAULT:IRET

ORG 0x020

START:

DI

CLA

OUT 1

OUT 5

LD #9

OUT 3

LD #0xB

OUT 7

BR PROG

PROG:

EI

LD $X

INCLP:

ADD #0x03

BR INCLP

ORG 0x040

INT1:

ST $X

NOP ;Запись числа X

ASL

INC

NEG

NOP ;Образ X до проверки на валидность.

CALL CHECKER

OUT 2

LD $X

IRET

ORG 0x050

INT3:

PUSH

NOP ;Запись числа X

IN 6

ST $X ;Сюда записывается промежуточный результат с ВУ-3.

ASL;

ADD $X

NEG

ST $X

POP

ADD $X

ST $X

NOP ;Образ X до проверки на валидность.

CALL CHECKER

IRET

ORG 0x070

CHECKER:

DEC

CMP #0x7F

NOP ;Проверка ОДЗ: F(x) < 128 (Если F(x) >= 128, то GOTO OVERFLOW).

BGE OVERFLOW

INC

CMP #0x80

NOP ;Проверка ОДЗ 2: F(x) >= -128 (Если F(x) < -128, то GOTO OVERFLOW)

BLT OVERFLOW

RET

OVERFLOW:

LD #0xC0

NOP ;Изменение числа X на -64.

ST $X

RET

ORG 0x716

X: WORD 0x00

|  |  |
| --- | --- |
| Исходные данные | Значения­ |
|  |  |
|  |  |
|  |  |

Вывод:

В ходе выполнения данной лабораторной работы я познакомился с организацией прерываний и тем, как они устроены в БЭВМ-NG. Узнал, для чего нужны векторы прерывания и как писать таблицы векторов. Выучил регистры состоянии PS, а также сигналы шины. Также разобрался с логикой управления при определении ВУ. Научился пользоваться командами IRET, DI, EI.

При выполнении данной лабораторной работы возникли трудности с определением, как "зайти" в подпрограмму, но в дальнейшем она была решена.