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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №6

Дисциплина: Основы профессиональной деятельности

Вариант № 14266

Выполнил: Васильев Никита Алексеевич, студент группы P3108

Преподаватель: Вербовой Александр Александрович

Санкт-Петербург 2024

Текст задания

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 2 содержимое X (ячейки памяти с адресом 02816) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=4X-3 на данное ВУ, a по нажатию кнопки готовности ВУ-2 прибавить утроенное содержимое РД данного ВУ к 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.

Программа на ассемблере

ORG 0x0 ; Инициализация векторов прерывания

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

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

V2: WORD $INT2, 0x180 ; Вектор прерывания #2

V3: WORD $DEFAULT, 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

ORG 0x28

X: WORD ? ; Переменная

MIN: WORD 0xFFE1 ; Минимальное значение X {-31}

MAX: WORD 0x0020 ; Максимальное значение X {32}

DEFAULT: IRET ; Обработка прерываний по умолчанию

ORG 0x35

START:

DI ; Запрет прерываний на период загрузки

CLA ; Заносим 0 в AC

OUT 0x1 ; MR КВУ-0 на вектор 0

OUT 0x7 ; MR КВУ-3 на вектор 0

OUT 0xB ; MR КВУ-4 на вектор 0

OUT 0xE ; MR КВУ-5 на вектор 0

OUT 0x12 ; MR КВУ-6 на вектор 0

OUT 0x16 ; MR КВУ-7 на вектор 0

OUT 0x1A ; MR КВУ-8 на вектор 0

OUT 0x1E ; MR КВУ-9 на вектор 0

LD #0x9 ; Разрешить прерывания на вектор 1

OUT 0x3 ; (1000 | 0001 = 1001) в MR КВУ-1

LD #0xA ; Разрешить прерывания на вектор 2

OUT 0x5 ; (1000 | 0010 = 1010) в MR КВУ-2

EI ; Разрешение прерываний

; Основная программа

PROG:

DI ; Запрет прерываний для безопасного доступа к памяти

LD X ; Загрузка переменной

EI ; Разрешение прерываний

SUB #2 ; Уменьшение AC

CALL CHECK ; Проверка принадлежности ОДЗ

DI ; Запрет прерываний для безопасного доступа к памяти

ST X ; Сохранение переменной

EI ; Разрешение прерываний

JUMP PROG ; Циклический переход

; Обработка прерывания ВУ-1

INT1:

CALL CHECK ; Проверка принадлежности ОДЗ

PUSH ; Сохранили AC в стек

NOP ; Отладочная точка останова

ASL

ASL

SUB #0x3

OUT 0x2 ; Вывод результата на ВУ-1

NOP ; Отладочная точка останова

POP ; Возврат значения AC

IRET ; Возврат из прерывания

; Обработка прерывания ВУ-2

INT2:

CALL CHECK ; Проверка принадлежности ОДЗ

PUSH ; Сохранили AC в стек

NOP ; Отладочная точка останова

CLA ; Очистка AC

IN 0x4 ; Прочитали данные с ВУ-2

SXTB ; Расширение знака для корректной работы

NOP ; Отладочная точка останова

PUSH ; Сохранили DR в стек

ASL ; Удвоенное значение DR

ADD (SP+0) ; Добавляем значение DR

ADD (SP+1) ; Складываем с X

NOP ; Отладочная точка останова

ST (SP+1) ; Сохранили результат в X

NOP ; Отладочная точка останова

POP ; Убрали DR из стека

POP ; Вернули АС

IRET ; Возврат из прерывания

; Проверка принадлежности ОДЗ

; Проверяет значения из AC

; Если значение не принадлежит ОДЗ то оно приводиться к нему

CHECK:

CMP $MIN ; Проверка на минимум

BGE CHECK\_MAX ; x >= MIN, переход на проверку максимума

LD $MAX ; Сохранение максимума

JUMP RETURN ; Выход из функции

CHECK\_MAX:

CMP $MAX ; Сравнение с максимумом

BLT RETURN ; Возврат если x < MAX

LD $MAX ; Сохранение максимального значения

RETURN: RET

Область представления и область допустимых значений

Область представления:

* X, MIN, MAX – знаковое 16-разрядное целое число
* DR (регистр данных) ВУ-1 и ВУ-2 – знаковое 8-разрядное целое число

Область допустимых значений:

Расположение в памяти ЭВМ

000 – 00F – расположение векторов прерываний;

028 – 030 – расположение переменных;

035 – 049 – расположение основной программы;

050 – 060 – расположение подпрограммы обработки прерываний с ВУ-1;

061 – 079 – расположение подпрограммы обработки прерываний с ВУ-2;

080 – 088 – расположение подпрограммы обработки для проверки ОДЗ.

Методика проверки программы

1. Загрузить комплекс программ в память БЭВМ
2. Заменить NOP на HLT по нужным адресам
3. Запустить БЭВМ в режиме РАБОТА
4. Ввести в регистр данных ВУ-1 тестовые данные
5. Установить готовность ВУ-1
6. Дождаться остановки программы
7. Запомнить текущее значение AC (считаем, что это Х)
8. Нажать кнопку ПРОДОЛЖЕНИЕ
9. Дождаться остановки программы
10. Сверить значение в аккумуляторе со значением введенным на ВУ-1
11. Нажать кнопку ПРОДОЛЖЕНИЕ
12. Дождаться остановки программы
13. Записать значение из аккумулятора, сравнив его с регистром данных ВУ-1
14. Нажать кнопку ПРОДОЛЖЕНИЕ
15. Дождаться остановки программы
16. Рассчитать ожидаемы результат вычисления выражения (DR ВУ-1) – 3X и сравнить его с полученным в аккумуляторе
17. Нажать кнопку ПРОДОЛЖЕНИЕ
18. Дождаться остановки программы
19. Проверить корректность приведения вычисленного значения в ОДЗ
20. Нажать кнопку ПРОДОЛЖЕНИЕ
21. Установить готовность ВУ-2
22. Дождаться остановки программы
23. Записать значение переменной X из аккумулятора
24. Нажать кнопку ПРОДОЛЖЕНИЕ
25. Дождаться остановки программы
26. Рассчитать ожидаемый результат вычисления функции F(X) и сравнить его с полученным в аккумуляторе и на ВУ-2

Проверка основной программы

1. Загрузить комплекс программы в память БЭВМ
2. Запустить БЭВМ в режиме РАБОТА
3. Дождаться достижения крайних значений аккумулятора для проверки корректности приведения к ОДЗ

|  |  |  |
| --- | --- | --- |
| Прерывание на ВУ-1 | | |
| *X до* | *X после* | *X ожидаемое* |
| 000A (10) | 003D (61) | 003D (61) |
| 0018 (24) | 005D (93) | 005D (93) |
| FFF8 (-8) | FFDD (-35) | FFDD (-35) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Прерывание на ВУ-2 | | | | |
| DR ВУ-2 | *X до* | *X без ОДЗ* | *X после* | *X ожидаемое* |
| 0A (10) | FFFA (-6) | 0018 (24) | 0018 (24) | 0018 (24) |
| 06 (6) | FFFA (-6) | 000C (12) | 000C (12) | 000C (12) |
| 26 (38) | FFFC (-4) | 006E (110) | 0020 (32) | 0020 (32) |

Вывод:

При выполнении данной лабораторной работы я познакомился с обменом данными в режиме прерываний и циклами прерываний, командами EI, DI, IRET.