

Федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Отчет
по лабораторной работе №6
«Обмен данными с ВУ по прерыванию»
по дисциплине «Основы профессиональной деятельности»

вариант 1413

Выполнил: Векшин А.И.,
группа Р3116
Преподаватель: Афанасьев
Д.Б.

Санкт-Петербург

~ 2023 ~

Содержание

Отчет	1
Содержание	2
Текст задания	3
Текст исходной программы	4
Описание программы	6
Назначение	6
Описание и назначение исходных данных и результата	6
Расположение в памяти	6
Методика проверки	7
Проверка прерывания ВУ-3	7
Проверка прерывания ВУ-2	7
Проверка выполнения основного цикла программы	8
Таблица результатов проверки	8
Доп	9
Вывод	11

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 (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 необрабатываемые прерывания.

1. Основная программа должна декрементировать содержимое X (ячейки памяти с адресом 027_{16}) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции $F(X)=3X+5$ на данное ВУ, а по нажатию кнопки готовности ВУ-2 прибавить утроенное содержимое РД данного ВУ к X , результат записать в X
3. Если X оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в X записать максимальное по ОДЗ число.

Текст исходной программы

```
                ORG 0x0                ; Инициализация векторов прерывания
V0:             WORD $DEFAULT, 0x180   ; Вектор 0 - стандартный обработчик
V1:             WORD $DEFAULT, 0x180   ; Вектор 1 - стандартный обработчик
V2:             WORD $INT2, 0x180      ; Вектор 2 - INT2
V3:             WORD $INT3, 0x180      ; Вектор 3 - INT3
V4:             WORD $DEFAULT, 0x180   ; Вектор 4 - стандартный обработчик
V5:             WORD $DEFAULT, 0x180   ; Вектор 5 - стандартный обработчик
V6:             WORD $DEFAULT, 0x180   ; Вектор 6 - стандартный обработчик
V7:             WORD $DEFAULT, 0x180   ; Вектор 7 - стандартный обработчик
```

```
DEFAULT: IRET                ; Возврат из прерывания
```

```
                ORG 0x27
X:              WORD 0x0
max:            WORD 0x0028
min:            WORD 0xFFD4
START: DI                ; Запрет прерываний
                LD #0xA
                OUT 0x5        ; Запись MR для ВУ-2
                LD #0xB
                OUT 0x7        ; Запись MR для ВУ-3
                CLA            ; Запись MR для других ВУ
                OUT 0x1
                OUT 0x3
                OUT 0xB
                OUT 0xE
                OUT 0x12
                OUT 0x16
                OUT 0x1A
                OUT 0x1E
```

```
BG:             DI                ; начало итерации
                LD X
                DEC
                CALL SAVE
                EI                ; конец итерации
                NOP                ; точка отладки #0
                JUMP BG           ; переход в начало цикла
```

```
SAVE:           CMP max            ; сохранение X
                BGE newX
                CMP min
                BGE store
newX:           LD max
store:          ST X
                RET
```

INT2:	NOP	; точка отладки #1
	PUSH	; Сохраняем исходный AC в стек
	CLA	
	IN 0x4	; чтение с РДВУ-2
	SXTB	
	PUSH	; сохранение N в стек
	LD X	; $X + 3*N$
	ADD (SP+0)	
	ADD (SP+0)	
	ADD (SP+0)	
	POP	; удаление N из стека
	CALL SAVE	; сохранение полученного значения
	NOP	; точка отладки #2
	LD (SP+0)	; Возвращаем исходный AC
	POP	
	IRET	; возврат к основной программе
INT3:	NOP	; точка отладки #3
	PUSH	; Сохраняем исходный AC в стек
	LD X	; загружаем текущее значение X
	ADD X	; $3*X + 5$
	ADD X	
	ADD #5	
	OUT 0x6	; вывод в РДВУ-3
	NOP	; точка отладки #4
	LD (SP+0)	; Возвращаем исходный AC
	POP	
	IRET	; возврат к основной программе

Описание программы

Назначение

Основная программа декрементирует содержимое X (ячейки памяти с адресом 027_{16}) в цикле.

По нажатию кнопки готовности ВУ-3 обработчик прерывания осуществляет вывод результата вычисления функции $F(X)=3X+5$ на данное ВУ.

По нажатию кнопки готовности ВУ-2 прибавляет утроенное содержимое РД данного ВУ к X , результат записывается в X

Если X оказывается вне ОДЗ при выполнении любой операции по его изменению, то в X записывается максимальное по ОДЗ число.

Описание и назначение исходных данных и результата

	Адрес	Назначение	ОПИ	ОДЗ
X	0x027	Ячейка памяти, значение которой увеличивается в цикле (исходные данные)	знаковое 16-ти разрядное число	$-0x2C \leq X \leq 0x28$
N		Аргумент (вводится на ВУ-2)	знаковое 8-ти разрядное число	$0x0 \leq X \leq 0xFF$
R		Результат (вывод на ВУ-3)	беззнаковое 8-ти разрядное число	$-0x7F \leq X \leq 0x7F$

Расположение в памяти

Программа: от 0x027 до 0x059

Вектора прерывания: от 0x000 до 0x00F

Константы:

0x028 (0x0028)) – максимальное значение X

0x029 (0xFFD4) – минимальное значение X

Исходные данные: 0x027

Результат: 0x027

Первая исполняемая команда: 0x02A

Методика проверки

Проверка прерывания ВУ-3

1. Загрузить комплекс программ в память БЭВМ
2. Изменить значения точек отладки №3 и №4 по адресам 0x052 и 0x059 соответственно на HLT
3. Запустить программу в автоматическом режиме с адреса 0x02A
4. Установить «Готовность ВУ-3»
5. Дождаться Останова
6. Записать содержимое АС в момент останова программы
7. Записать текущее значение IP
8. Ввести в клавишный регистр значение 0000 0000 0010 0111
9. Нажать кнопку «Ввод адреса»
10. Нажать кнопку «Чтение»
11. Записать значение ячейки 0x027
12. Ввести в клавишный регистр ранее записанное значение IP
13. Нажать кнопку «Ввод адреса»
14. Вычислить значение по формуле $F(X) = 3X+5$, где X соответствует значению, записанному в пункте 11, оставив при этом только 8 младших разрядов
15. Продолжить выполнение программы
16. Дождаться Останова
17. Убедиться в равенстве значений АС и записанного в пункте 6
18. Сравнить значение регистра данных ВУ-3 со значением, полученным в пункте 14. Убедиться в правильности вычисленного значения

Проверка прерывания ВУ-2

1. Загрузить комплекс программ в память БЭВМ
2. Изменить значения точек отладки №1 и №2 по адресам 0x046 и 0x050 соответственно на HLT
3. Запустить программу в автоматическом режиме с адреса 0x02A
4. Ввести значение в регистр данных ВУ-2
5. Установить «Готовность ВУ-2»
6. Дождаться Останова
7. Записать содержимое АС в момент останова программы
8. Записать текущее значение IP
9. Ввести в клавишный регистр значение 0000 0000 0010 0111
10. Нажать кнопку «Ввод адреса»
11. Нажать кнопку «Чтение»
12. Записать значение ячейки 0x027
13. Ввести в клавишный регистр ранее записанное значение IP
14. Нажать кнопку «Ввод адреса»
15. Утроить значение регистра данных ВУ-2 и добавить к записанному значению АС
16. Продолжить выполнение программы
17. Дождаться Останова
18. Убедиться в равенстве значений АС и записанного в пункте 7
19. Сравнить значение ячейки памяти с адресом 0x027 (для его получения можно повторить пункты 7-13) со значением, полученным в пункте 15. Убедиться в правильности вычисленного значения

Санкт-Петербург

~ 2023 ~

Проверка выполнения основного цикла программы

1. Загрузка комплекса программ в память БЭВМ
2. Изменить значение точки отладки №0 по адресу 0x03d на HLT
3. Запустить программу в автоматическом режиме с адреса 0x02A
4. Дождаться Остановка
5. Записать текущее значение IP
6. Ввести в клавишный регистр значение 0000 0000 0010 0111
7. Нажать кнопку «Ввод адреса»
8. Нажать S «Чтение»
9. Записать значение ячейки 0x027
10. Ввести в клавишный регистр ранее записанное значение IP
11. Нажать кнопку «Ввод адреса»
12. Продолжить выполнение программы
13. Дождаться Остановка
14. Сравнить значение ячейки памяти с адресом 0x027 (для его получения можно повторить пункты 5-11) со значением, полученным в пункте 9. Новое значение должно быть либо на 1 больше записанного, либо равняться минимальному значению X по ОДЗ в случае, если записанное значение больше – 0x28

Таблица результатов проверки

№	Прерывание ВУ-3			Прерывание ВУ-2			Основной цикл		
	Исходное	Подсчитанное	Полученное	Исходное	Подсчитанное	Полученное	Исходное	Подсчитанное	Полученное
1	X: 0x0026	0x74	0x74	РДВУ-2: 0xD5 X: 0x0006	0xFFD5	0xFFD5	X: 0x0027	0x0025	0x0025
2	X: 0xFFE7	0xBA	0xBA	РДВУ-2: 0xF0 X: 0xFFD4	0xFFFF0	0xFFFF0	X: 0xFFD4	0x0027 (из-за ОДЗ)	0x0027 (из-за ОДЗ)
3	X: 0x0006	0x14	0x14	РДВУ-2: 0x0F X: 0xFFD7	0x000F (из-за ОДЗ)	0x000F (из-за ОДЗ)	X: 0x0028	0x0026	0x0026

Доп

```
                ORG 0x0                ; Инициализация векторов прерывания
V0:             WORD $DEFAULT, 0x180   ; Вектор 0 - стандартный обработчик
V1:             WORD $DEFAULT, 0x180   ; Вектор 1 - стандартный обработчик
V2:             WORD $INT, 0x180       ; Вектор 2 - INT2
V3:             WORD $INT, 0x180       ; Вектор 3 - INT3
V4:             WORD $DEFAULT, 0x180   ; Вектор 4 - стандартный обработчик
V5:             WORD $DEFAULT, 0x180   ; Вектор 5 - стандартный обработчик
V6:             WORD $DEFAULT, 0x180   ; Вектор 6 - стандартный обработчик
V7:             WORD $DEFAULT, 0x180   ; Вектор 7 - стандартный обработчик
```

```
DEFAULT:IRET    ; Возврат из прерывания
```

```
                ORG 0x27
X:             WORD 0x0
max:          WORD 0x0028
min:          WORD 0xFFD4
```

```
START: DI       ; Запрет прерываний
        LD #0xA
        OUT 0x5 ; Запись MR для ВУ-2
        LD #0xB
        OUT 0x7 ; Запись MR для ВУ-3
        CLA    ; Запись MR для других ВУ
        OUT 0x1
        OUT 0x3
        OUT 0xB
        OUT 0xE
        OUT 0x12
        OUT 0x16
        OUT 0x1A
        OUT 0x1E
```

```
BG:      DI       ; начало итерации
        LD X
        DEC
        CALL SAVE
        EI       ; конец итерации
        NOP      ; точка отладки #0
        JUMP BG  ; переход в начало цикла
```

```
SAVE:    CMP max   ; сохранение X
        BGE newX
        CMP min
        BGE store
newX:    LD max
```

```

store:  ST X
        RET

INT:     NOP
        PUSH                ; Сохраняем исходный AC в стек
        CLA
        IN 0x7              ; чтение с РДБУ-2
        AND #0x40           ;
        BEQ IEX             ; проверяем готовность ВУ-2
        CLA                 ; Если готовность есть:
        IN 0x4              ; чтение с РДБУ-2
        SXTB
        PUSH                ; сохранение N в стек
        LD X                ;  $X + 3 \cdot N$ 
        ADD (SP+0)
        ADD (SP+0)
        ADD (SP+0)
        POP                 ; удаление N из стека
        CALL SAVE           ; сохранение полученного значения
        NOP                 ; точка отладки #2
        IN 0x5              ; вывод на ВУ-3
        AND #0x40           ;
        BEQ IEX             ; проверяем готовность ВУ-3
        LD X                ; загружаем текущее значение X
        ADD X               ;  $3 \cdot X + 5$ 
        ADD X
        ADD #5
        OUT 0x6             ; вывод в РДБУ-3
        NOP                 ; точка отладки #4

IEX:     LD (SP+0)
        POP
        IRET

```

Вывод

.Λ_Λ

(·ω·.)ツ—☆·*。

⊂ √ ·° +.

じ—J °。+ *''')

·'·'·*''')

(,·' (,·'* ☆ Нужно читать методичку