

Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники



**Лабораторная работа по основам профессиональной
деятельности №6**

Вариант №2462

Группа: Р3132

Студент: Гиниятуллин А. Р.

Преподаватель: Перминов И. В.

г. Санкт-Петербург

2025

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 (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 уменьшать на 2 содержимое X (ячейки памяти с адресом 033_{16}) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции $F(X)=4X-8$ на данное ВУ, а по нажатию кнопки готовности ВУ-3 вычесть X из содержимого РД данного ВУ, 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.

Ассемблерный код:

Метка	Команда	Аргумент	Комментарий
	ORG	0x0	
V0:	WORD	\$DEFAULT, 0x180	Вектор прерывания #0
V1:	WORD	\$INT1_HANDLER, 0x180	Вектор прерывания #1
V2:	WORD	\$DEFAULT, 0x180	Вектор прерывания #2
V3:	WORD	\$INT3_HANDLER, 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:	IRET		
	ORG	0x020	
MIN_ODZ_X:	WORD	-30	-30 decimal for X
MAX_ODZ_X:	WORD	33	33 decimal for X (This is the value to write if X goes out of bounds)
CONST_2:	WORD	2	
CONST_8:	WORD	8	
TEMP_STORAGE:	WORD	0	Temporary storage DR_VU3
X:	ORG	0x33	
	WORD	0	Initial value of X (e.g., 0 for testing)
	ORG	0x050	

START:	DI	Disable interrupts during setup
	CLA	
	OUT 0x1	MR for VU-0
	OUT 0x5	MR for VU-2
	OUT 0x9	MR for VU-4
	OUT 0xB	MR for VU-5
	OUT 0xD	MR for VU-6
	OUT 0xF	MR for VU-7
	OUT 0x11	MR for VU-8
	OUT 0x13	MR for VU-9
	LD #0x9	0x9 = 1001_binary (Enable interrupt AND Vector #1)
	OUT 0x3	Write to MR of VU-1
	LD #0xB	0xB = 1011_binary (Enable interrupt AND Vector #3)
	OUT 0x7	Write to MR of VU-3
	EI	Enable interrupts now that setup is done
MAIN_LOOP:	DI	--- Start of ATOMIC operation for X = X - 2 ---
	LD X	Load current X into AC
	SUB CONST_2	AC = X - 2
	CALL CHECK_X_ODZ	Check if AC (new X candidate) is within ODZ, correct AC if not
	ST X	Store modified (and possibly corrected) X from AC
	EI	--- End of ATOMIC operation ---
	JUMP MAIN_LOOP	
INT1_HANDLER:	DI	Disable further interrupts
	PUSH	Save current Accumulator
	LD X	Load current X into AC
	ASL	AC = 4*X - 8
	ASL	
	SUB CONST_8	
	OUT 0x2	Output F(X) result to VU-1 Data Register
	POP	Restore original Accumulator
	EI	Re-enable interrupts before returning
	IRET	
INT3_HANDLER:	DI	Disable further interrupts
	PUSH	Save current Accumulator
	IN 0x6	Read from VU-3 Data Register
	SXTB	Sign-extend AC from 8-bit to 16-bit
	ST TEMP_STORAGE	Store this value temporarily
	LD TEMP_STORAGE	
	SUB X	AC = (sign-extended DR_VU3) - X
	CALL CHECK_X_ODZ	Check if AC (the new X candidate) is within ODZ, correct AC if not
	ST X	Store new (and possibly corrected) value into X
	POP	Restore original Accumulator
	EI	Re-enable interrupts before returning
	IRET	

CHECK_X_ODZ:	CMP BMI CMP BEQ BPL	MIN_ODZ_X LOAD_MAX_VALUE MAX_ODZ_X ODZ_EXIT LOAD_MAX_VALUE	Compare AC with MIN_ODZ_X. Sets N if AC < MIN_ODZ_X If AC < MIN_ODZ_X (N=1), it's out of bounds, load MAX_ODZ_X Compare AC with MAX_ODZ_X If AC == MAX_ODZ_X, it's in bounds. If AC > MAX_ODZ_X (AC - MAX_ODZ_X is positive, N=0), it's out of bounds, load MAX_ODZ_X BPL (Branch if PLus) branches if N=0.
ODZ_EXIT:	RET		AC is within ODZ or was equal to MAX_ODZ_X
LOAD_MAX_VALUE:	LD RET	MAX_ODZ_X	Load MAX_ODZ_X into AC because original AC was out of bounds

Назначение программы (Program Purpose):

- **Основная программа:** Циклически и атомарно уменьшает значение знаковой 16-битной переменной X, хранящейся в памяти по адресу 0x033, на 2. При этом осуществляется проверка на выход значения X за пределы допустимых значений (ОДЗ).
- **Обработчик прерывания от ВУ-1 (вектор прерывания #1):** Активируется по нажатию кнопки готовности ВУ-1. Вычисляет функцию $F(X) = 4X - 8$. Результат вычисления, представляемый как 8-битное знаковое число, выводится в регистр данных (РД) ВУ-1 (порт 0x2).
- **Обработчик прерывания от ВУ-3 (вектор прерывания #3):** Активируется по нажатию кнопки готовности ВУ-3. Считывает 8-битное знаковое значение из регистра данных (РД) ВУ-3 (порт 0x6), выполняет знаковое расширение этого значения до 16 бит, затем вычитает из него текущее значение переменной X. Полученный результат записывается обратно в ячейку X.
- **Контроль ОДЗ:** Если в результате любой операции (в основной программе или в обработчике прерывания от ВУ-3) значение переменной X выходит за пределы установленного ОДЗ, то X принудительно устанавливается в максимальное допустимое значение по ОДЗ.

- **Обработка прочих прерываний:** Все прерывания от других внешних устройств или по другим векторам, не задействованным в задании, игнорируются. Для этого в таблице векторов прерываний для них установлен адрес стандартного обработчика (DEFAULT_INT_HANDLER), который выполняет только инструкцию возврата из прерывания (IRET). Кроме того, при инициализации программы прерывания от неиспользуемых ВУ отключаются путем записи нуля в их регистры управления (MR).

Область представления и область допустимых значений исходных данных и результата:

- **a. Область представления (Data Representation):**
 - Переменная X: знаковое 16-битное целое число. Диапазон значений: [-32768, 32767].
 - Константы MIN_ODZ_X, MAX_ODZ_X, CONST_2, CONST_8: знаковые 16-битные целые числа.
 - Значение из регистра данных ВУ-3 (РДВУ-3): знаковое 8-битное целое число. Диапазон значений: [-128, 127].
 - Результат функции F(X) для вывода в ВУ-1: знаковое 8-битное целое число. Диапазон значений: [-128, 127].
- **b. Область допустимых значений (ОДЗ) для переменной X:**
ОДЗ для переменной X определяется ограничением, накладываемым функцией $F(X) = 4X - 8$, результат которой должен помещаться в 8-битный знаковый регистр данных ВУ-1 (от -128 до +127):
 - $-128 \leq 4X - 8 \leq 127$
 - $-128 + 8 \leq 4X \leq 127 + 8$
 - $-120 \leq 4X \leq 135$
 - $-120 / 4 \leq X \leq 135 / 4$
 - $-30 \leq X \leq 33.75$

Поскольку X является целым числом, ОДЗ для X составляет: **[-30, 33]**.

 - $\text{MIN_ODZ_X} = -30$ (FFEAh или -1Eh в 16-ричном виде)
 - $\text{MAX_ODZ_X} = +33$ (0021h в 16-ричном виде)

При выходе значения X за эти пределы, в X записывается MAX_ODZ_X (т.е. 33).

Расположение в памяти ЭВМ программы, исходных данных и результатов:

- Инициализация векторов прерывания: [0; 00F]
- Исходные данные программы: [02B; 031]
- Программа: [032; 036]
- Подпрограмма обработки прерывания ВУ-1: [037; 03E]
- Подпрограмма обработки прерывания ВУ-2: [03F; 04F]
- Подпрограмма проверки числа во вхождение в Одз: [050; 055]
- Подпрограмма обработки прерываний для непредусмотренных ВУ: [059; 064]

Методика проверки

Убедиться, что выставлен режим работы “Run” и запустить программу, нажав на кнопку “Start”

1. Основная программа

1. Загрузить текст программы в БЭВМ.
2. Записать в переменную X максимальное по ОДЗ значение (41)
3. Запустить программу в режиме остановки.
4. Пройти нужное количество шагов программы, убедиться, что при уменьшении X на 1, до того момента, когда он равен -43, происходит сброс значения в макс. по ОДЗ.

2. Обработка прерывания ВУ-1

- 1.1. Открыть ВУ, нажав кнопку “CDev 1” и выставить готовность ВУ нажав кнопку “Ready”
- 1.2. Дождаться остановки программы
- 1.3. Из аккумулятора (AC) записать значение X в таблицу
- 1.4. Продолжить выполнение программы, нажав кнопку “Continue”
- 1.5. Дождаться остановки программы
- 1.6. Открыть ВУ, нажав кнопку “CDev 1” и записать посчитанное значение F(X) в таблицу
- 1.7. Посчитать ожидаемый результат функции по формуле $3x + 2$ и записать результат в таблицу
- 1.8. Нажать кнопку “Continue”, чтобы основная программа продолжила выполняться

Обработка прерывания ВУ-2

- 1.9. Записать число для подсчета логической операции в таблицу
- 1.10. Открыть ВУ, нажав кнопку “CDev 2” и ввести число, которое ранее было введено в таблицу
- 1.11. Выставить готовность ВУ нажав кнопку “Ready”
- 1.12. Дождаться остановки программы
- 1.13. Из аккумулятора (AC) записать значение X в таблицу
- 1.14. Продолжить выполнение программы, нажав кнопку “Continue”
- 1.15. Дождаться остановки программы
- 1.16. Из аккумулятора (AC) записать значение “Исключающее ИЛИ-НЕ” в таблицу
- 1.17. Посчитать “Исключающее ИЛИ-НЕ” введенного в ВУ значения и значения X (если результат выходит за границы [-43, 41], то результат считать равным 41)
- 1.18. Нажать кнопку “Continue”, чтобы основная программа продолжила выполняться

Таблица для тестирования программного комплекса

ВУ-1	
Значение X	0000 0000 0010 0010 (34)
Посчитанное программой значение F(X)	0000 0000 0110 1000 (104)

Посчитанное пользователем значение F(X)	0000 0000 0110 1000 (104)
Значение X	0000 0000 0010 1011 (13)
Посчитанное программой значение F(X)	0000 0000 0010 1001 (41)
Посчитанное пользователем значение F(X)	0000 0000 0010 1001 (41)
ВУ-2	
Введенное значение	1111 1111 1001 1100 (-100)
Значение X	0000 0000 0001 1001 (15)
Результат, посчитанный программой	0000 0000 0010 1100 (44)
Результат, посчитанный пользователем	0000 0000 0010 1100 (44)
Введенное значение	0000 0000 0100 1101 (77)
Значение X	1111 1111 1101 1000 (-40)
Результат, посчитанный программой	0000 0000 0010 1100 (44)
Результат, посчитанный пользователем	0000 0000 0010 1100 (44)

Вывод:

Изучил организацию процесса управления программой при управляемом прерываниями вводе-выводе.