

Министерство высшего образования и науки Российской Федерации
Национальный научно-исследовательский университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №6
по дисциплине
«Основы профессиональной деятельности».

Вариант №696.

Работу выполнил:
Афанасьев Кирилл Александрович,
Студент группы Р3106.
Преподаватель:
Афанасьев Дмитрий Борисович.

Санкт-Петербург, 2023

Оглавление

<i>Задание</i>	3
<i>Текст исходной программы</i>	3
<i>Описание программы.....</i>	5
<i>Методики проверки</i>	6
Проверка основного цикла	6
Проверка прерывания ВУ-2.....	7
Проверка прерывания ВУ-3.....	7
Сводная таблица результатов проверок по соответствующим методикам.....	8
<i>Вывод</i>	8

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.

Вариант 696:

1. Основная программа должна увеличивать на 2 содержимое X (ячейки памяти с адресом 031_{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 изменить знак содержимого РД данного ВУ и записать в X
3. Если X оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в X записать минимальное по ОДЗ число.

»

Текст исходной программы

Таблица 1: Текст исходной программы.

Готовность ВУ-3: $F(X) = 3X-5 \rightarrow \text{РДВУ-3}$, Готовность ВУ-2: $(-\text{РДВУ2}) \rightarrow X$. Основной цикл: $X + 2 \rightarrow X$.		
	ORG 0x0	; Инициализируем векторы прерываний.
V0:	WORD \$DEFAULT, 0x180	; Вектор 0 – стандартный обработчик.
V1:	WORD \$DEFAULT, 0x180	; Вектор 1 – стандартный обработчик.
V2:	WORD \$INT2, 0x180	; Вектор 2 установлен на обр. INT2.
V3:	WORD \$INT3, 0x180	; Вектор 3 установлен на обр. 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	; Стандартный обработчик: просто ; возвращает из прерывания.
START:	ORG 0x020 DI	; На время инициализации векторов. ; запретим какие-либо прерывания.
	LD #0xA	; Инициализируем прерывание ВУ-2.
	OUT 0x5	; на вектор 2 (и разрешим его).
	LD #0xB	; Инициализируем прерывание ВУ-3.
	OUT 0x7	; на вектор 3 (и разрешим его).
	CLA	; Остальные прерывания ВУ будут.
	OUT 0x1	; назначены на вектор 0.
	OUT 0x3	; (и вообще запрещены на уровне КВУ).

	OUT 0xB	;
	OUT 0xE	;
	OUT 0x12	;
	OUT 0x16	;
	OUT 0x1A	;
	OUT 0x1E	;
	JUMP \$PROG	; Векторы назначены. Двигаемся в ; основной цикл...
ADDR_X:	ORG 0x030	
X:	WORD 0x031	; Указатель на X.
PROG:	WORD ?	; Основная переменная программы.
	EI	; Разрешаем прерывания.
	LD ADDR_X	; Загрузим адрес X.
	PUSH	; И положим адрес в стек.
INCLP:	CALL \$AT_INC	; Вызываем атомарную операцию +2.
	JUMP INCLP	; Это вся суть основного цикла.
TEMP_ADDR:	ORG 0x040	
AT_INC:	WORD ?	; Врем. ячейка для адреса аргумента.
	DI	; Атомарное увел. яч. памяти на 2.
	LD &1	; Загрузим адрес операнда.
	ST TEMP_ADDR	; Сохраним во временную ячейку.
	LD (TEMP_ADDR)	; Загрузим операнд.
	NOP	; Точка отладки 1.
	INC	; Увеличим его на 2. Прерывания
	INC	; были запрещены до этого.
	CALL \$AAV_CHECK	; Сверим с ОДЗ полученный результат.
	ST (TEMP_ADDR)	; Сохраним его в ячейку памяти.
	NOP	; Точка отладки 2.
	EI	; Снова разрешим прерывания.
	RET	; И вернем логику управления.
INT2:	ORG 0x050	; Обработчик вектора прерываний 2
	NOP	; Точка отладки 3.
	PUSH	; Сохраним состояние АС.
	IN 0x4	; Произведем чтение с ВУ-2.
	SXTB	; Расширим знак пришедшего числа.
	NEG	; Изменим знак числа.
	ST \$X	; Сохраним в X.
	POP	; Вернем состояние АС.
	NOP	; Точка отладки 4.
	IRET	; Выйдем из текущего прерывания.
INT3:	ORG 0x060	; Обработчик вектора прерываний 3.
	NOP	; Точка отладки 5.
	PUSH	; Сохраним состояние АС.
	LD \$X	; Загрузим X в АС.
	ASL	; Арифметический сдвиг влево ($X*2$).
	ADD \$X	; $2*X + X \rightarrow AC$ ($3*X$).
	SUB #5	; Из $3*X$ вычитаем 5.
	OUT 6	; Итоговый результат $F(X)$ на ВУ-3.

	POP	; Вернем состояние АС.
	NOP	; Точка отладки 6.
	IRET	; Выйдем из текущего прерывания.
	ORG 0x070	
AAV_CHECK:	CMP MAX_VAL	; Проверим верхнюю границу ОДЗ.
	BEQ AAV_RET	; Они равны? Да – на выход.
	BGE RESET	; Результат больше? Сбрасываем его.
	CMP MIN_VAL	; Проверим нижнюю границу ОДЗ.
	BGE AAV_RET	; Результат больше? На выход.
RESET:	LD MIN_VAL	; Иначе сбросим до минимального числа.
AAV_RET:	RET	; Тот самый выход!
MIN_VAL:	WORD 0xFFD5	; Нижняя граница ОДЗ.
MAX_VAL:	WORD 0x002A	; Верхняя граница ОДЗ.

Окончание таблицы.

Описание программы

- Назначение основной программы: увеличение значения ячейки памяти на 2. Нажатие на кнопку «Готов» на ВУ-2 обновит значение ячейки памяти инвертированным значением с РДВУ-2. Нажатие на кнопку «Готов» на ВУ-3 выведет в РДВУ-3 значение $F(X) = 3X - 5$.
- Описание исходных данных:
 - X – основная ячейка памяти.
 - ОПИ:
 - X – знаковое, 8-разрядное число.
 - ОДЗ:
 - $-0x2B \leq X \leq 0x2A$
- Расположение в памяти ЭВМ программы и исходных данных:
 - Программный комплекс располагается в следующих ячейках памяти:
 - Векторы прерываний: между ячейками 000 и 00F включительно
 - Стандартный обработчик прерываний: между ячейками 010 и 01F включительно
 - Инициализация векторов прерываний: между ячейками 020 и 02E включительно
 - Основной цикл программы: между ячейками 030 и 036 включительно
 - Подпрограмма атомарного увеличения ячейки на 2: между ячейками 040 и 04C включительно
 - Обработчик прерываний вектора 2: между ячейками 050 и 05A включительно
 - Обработчик прерываний вектора 3: между ячейками 060 и 06B включительно
 - Подпрограмма проверки на вхождение ячейки в ОДЗ: между ячейками 070 и 078 включительно

- Исходные данные должны располагаться в ячейках памяти:
 - X – 031
- Результат работы программы должен располагаться в ячейке памяти 031.
- В программе используются следующие неизменяемые значения (константы):
 - По адресу 000 – значение 0x0010
 - По адресу 001 – значение 0x0180
 - По адресу 002 – значение 0x0010
 - По адресу 003 – значение 0x0180
 - По адресу 004 – значение 0x0050
 - По адресу 005 – значение 0x0180
 - По адресу 006 – значение 0x0060
 - По адресу 007 – значение 0x0180
 - По адресу 008 – значение 0x0010
 - По адресу 009 – значение 0x0180
 - По адресу 00A – значение 0x0010
 - По адресу 00B – значение 0x0180
 - По адресу 00C – значение 0x0010
 - По адресу 00D – значение 0x0180
 - По адресу 00E – значение 0x0010
 - По адресу 00F – значение 0x0180
 - По адресу 077 – значение 0xFFD5
 - По адресу 078 – значение 0x002A
- В программе также используются вспомогательные ячейки, находящиеся по адресам 030 и 040.
- Первая команда располагается в ячейке по адресу 020.

Методики проверки

Проверка основного цикла

1. Загрузить комплекс программ в память Базовой ЭВМ.
2. Изменить значение отладочной точки 1 и отладочной точки 2 по адресам 045 и 04A на HLT.
3. Переключить тумблер в режим «РАБОТА», отключить потактовое исполнение, нажать кнопку «ПУСК».
4. Дождаться остановки работы ЭВМ.
5. Записать текущее значение счетчика команд (IP).
6. Ввести в клавишный регистр (IR) значение 0000.0000.0011.0001 (0x0031).
7. Нажать кнопку «ВВОД АДРЕСА».
8. Нажать кнопку «ЧТЕНИЕ».
9. Записать значение регистра данных (DR).
10. Ввести в клавишный регистр (IR) ранее записанное значение счётчика команд (IP).
11. Нажать кнопку «ВВОД АДРЕСА».
12. Не меняя состояние тумблеров, нажать кнопку «ПРОДОЛЖИТЬ».

13. Дождаться остановки работы ЭВМ.
14. Повторить пункты 5–11 включительно.
15. Сравнить полученные 2 записанных значения.
 - а. Второе значение либо должно быть больше первого на 2.
 - б. Либо равняться минимальному значению согласно ОДЗ исходных данных, в случае если первое значение было больше, чем 0x28.

Проверка прерывания ВУ-2

1. Загрузить комплекс программ в память Базовой ЭВМ.
2. Изменить значение отладочной точки 3 и отладочной точки 4 по адресам 051 и 058 на HLT.
3. Переключить тумблер в режим «РАБОТА», отключить потактовое исполнение, нажать кнопку «ПУСК».
4. Придумать любое число и записать его в РДВУ-2.
5. Установить «Готовность ВУ-2».
6. Дождаться остановки работы ЭВМ.
7. Записать текущее значение аккумулятора (АС).
8. Не меняя состояние тумблеров, нажать кнопку «ПРОДОЛЖИТЬ».
9. Дождаться остановки работы ЭВМ.
10. Записать текущее значение аккумулятора (АС). Проверить, что оно совпало с тем значением, что было получено в пункте 7.
11. Ввести в клавишный регистр (IR) значение 0000.0000.0011.0001 (0x0031).
12. Нажать кнопку «ВВОД АДРЕСА».
13. Нажать кнопку «ЧТЕНИЕ».
14. Вычислить дополнительный код числа, полученного в пункте 4 и сравнить его с младшим байтом DR. Они должны быть равны.

Проверка прерывания ВУ-3

1. Загрузить комплекс программ в память Базовой ЭВМ.
2. Изменить значение отладочной точки 5 и отладочной точки 6 по адресам 061 и 069 на HLT.
3. Переключить тумблер в режим «РАБОТА», отключить потактовое исполнение, нажать кнопку «ПУСК».
4. Установить «Готовность ВУ-3».
5. Дождаться остановки работы ЭВМ.
6. Записать текущее значение аккумулятора (АС).
7. Ввести в клавишный регистр (IR) значение 0000.0000.0011.0001 (0x0031).
8. Нажать кнопку «ВВОД АДРЕСА».
9. Нажать кнопку «ЧТЕНИЕ».
10. Записать значение младшего байта регистра данных (DR).
11. Ввести в клавишный регистр (IR) ранее записанное значение счётчика команд (IP).
12. Нажать кнопку «ВВОД АДРЕСА».
13. Не меняя состояние тумблеров, нажать кнопку «ПРОДОЛЖИТЬ».

14. Дождаться остановки работы ЭВМ.
15. Записать текущее значение аккумулятора (АС). Проверить, что оно совпадает с тем значением, что было получено в пункте 6.
16. Вычислить функцию $F(X) = 3X - 5$ от значения, полученного в пункте 10. Сравнить его с текущим значением РДВУ-3. Убедиться, что значения равны.

Сводная таблица результатов проверок по соответствующим методикам

Таблица 2: Результаты проверки работы программного комплекса.

№	Основной цикл			Прерывание ВУ-2			Прерывание ВУ-3		
	Исходное	Подсчитанное	Полученное	Исходное	Подсчитанное	Полученное	Исходное	Подсчитанное	Полученное
1	DR: 0x0000	0x0002	0x0002	AC: 0x000A	0x000A	0x000A	AC: 0x001A	0x001A	0x001A
1				DRED- 2: 0x1F	0xE1	0xE1	X: 0x001A	0x0049	0x0049
2	DR: 0x0028	0x002A	0x002A	AC: 0xFFE5	0xFFE5	0xFFE5	AC: 0x001E	0x001E	0x001E
2				DRED- 2: 0x54	0xAC	0xAC	X: 0x001E	0x55	0x55
3	DR: 0x002A	0xFFD5	0xFFD5	AC: 0xFFDD	0xFFDD	0xFFDD	AC: 0xFFE1	0xFFE1	0xFFE1
3				DRED- 2: 0x80	0x80	0x80	X: 0xFFE1	0x9E	0x9E

Окончание таблицы.

Вывод

Во время выполнения данной лабораторной работы я изучил процесс прерывания программы и исследовал функционирование Базовой ЭВМ при обмене данными в режиме прерывания программы, а также научился писать методики проверки программных комплексов и следовать им.