

Федеральное государственное автономное образовательное учреждение высшего  
образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

---

Лабораторная работа №6 по дисциплине  
«Основы профессиональной деятельности»  
Вариант 1361

Выполнил:

Студент группы Р3113

Иванов Евгений Дмитриевич

Преподаватель:

Ткешелашвили Нино Мерабиевна

г. Санкт-Петербург

2020

Условие варианта(вариант 1361):

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 необработываемые прерывания.

Введите номер варианта

1361

1. Основная программа должна уменьшать на 3 содержимое X (ячейки памяти с адресом 024<sub>16</sub>) в цикле.

2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=-2X-5 на данное ВУ, а по нажатию кнопки готовности ВУ-3 выполнить операцию побитового маскирования, оставив 4-х младших разряда содержимого РД данного ВУ и 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.

Код программы:

```
ORG 0x000
V0: WORD $VU1
WORD 0x180
V1: WORD $VU3
WORD 0x180
V2: WORD $DEF
WORD 0x180
V3: WORD $DEF
WORD 0x180
V4: WORD $DEF
WORD 0x180
V5: WORD $DEF
WORD 0x180
V6: WORD $DEF
WORD 0x180
V7: WORD $DEF
WORD 0x180
DEF: IRET
```

Задание всех векторов прерывания, в программе используется два ВУ (ВУ-1 и ВУ-3), поэтому для них инициализируются вектора прерывания (для каждого отдельный, так как обработка различная): V0 и V1, для остальных векторов обработки нет (мгновенный возврат IRET)

```
ORG 0x024
X: WORD ?
LT: WORD 0xFFBE
RT: WORD 0x003D
```

Задание ячейки с переменной X, а также

- Левой границы(нижней) ОДЗ = LT
- Правой границы(верхней) ОДЗ = RT

```
ORG 0x100
D: WORD 0x5
VU1: LD $X
NEG
ASL
SUB D
OUT 2
NOP
IRET
```

Обработка прерывания по вектору 0, назначенный на ВУ-1  
На РДВУ-1 выводится значение формулы F(X) = -2X-5

Вывод на регистр данных, сброс флага  
Метка для отладки  
Возврат из прерывания

```
ORG 0x200
A: WORD ?
MASK:WORD 0x0F
VU3: LD $X
```

Обработка прерывания по вектору 1, назначенному на ВУ-3  
Данное значение X  
Маска, оставляющая последние 4 бита  
Загрузка X

ST	A	Сохранение X в локальную переменную
IN	6	Взять значение из РДВУ-3, сброс флага
AND	A	Побитовое умножение с значением X
AND	MASK	Маскирование результата
ST	\$X	Сохранение результата в значение X
NOP		Метка для отладки
IRET		Возврат из прерывания
ORG	0x300	
B: WORD	0x3	Вычитаемое из X значение.
START:CLA		Начало программы. 0=>AC
OUT	0x05	Присваивание запрета прерываний на все ВУ, кроме
OUT	0x0B	используемых(ВУ-1 и ВУ-3)
OUT	0x0D	
OUT	0x11	
OUT	0x15	
OUT	0x19	
OUT	0x1D	
LD	#0x8	Установка вектора прерывания V0 на ВУ-1
OUT	3	
LD	#0x9	Установка вектора прерывания V1 на ВУ-3
OUT	7	
CLA		
WHILE:EI		Начало работы программы, разрешение прерываний
DI		Запрет прерываний, так как начинается обработка X
LD	\$X	Загрузка переменной
SUB	B	Уменьшение значения на B
CMP	\$LT	Проверка на выход за ОДЗ снизу, если переменная X
BLT	NEWX	находится вне ОДЗ, то переход к NEWX метке
CMP	\$RT	Проверка на выход за ОДЗ сверху, если переменная X
BGE	NEWX	находится вне ОДЗ, то переход к NEWX метке
ST	\$X	Сохранение результата в переменную
JUMP	WHILE	Переход к следующей итерации цикла.
NEWX:LD	\$RT	Загрузка максимального по ОДЗ значения X
ST	\$X	в переменную X.
JUMP	WHILE	Переход к следующей итерации цикла.
HLT		Остановка программы(никогда не выполняется)

### Описание программы:

- Основная программа уменьшает значение переменной X(ячейки памяти с адресом 0x024) на 3 каждую итерацию в цикле.
- Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции  $F(X) = -2X - 5$  на данное ВУ, а по нажатию кнопки готовности ВУ-3 выполнить операцию побитового маскирования, оставив 4-х младших разряда содержимого РД данного ВУ и X, результат записать в X.
- Если X оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в X записать максимальное по ОДЗ число.

### Поиск ОДЗ:

Так как происходит вывод на ВУ-1, то знаковое число в 8-ми разрядах находится в диапазоне от -128 до 127. Это значит что  $F(X)$  лежит в этом интервале, найдем крайние значения переменной X:

$-2x - 5 = -128 \Leftrightarrow -2x = -123 \Leftrightarrow x \leq 61 = 00111101_2 = 3D_{16}$  — верхняя

граница. (0x3D)

$-2x - 5 = 127 \Leftrightarrow -2x = -132 \Leftrightarrow x \geq -66 = 10111110_2 = BE_{16}$  — нижняя граница.

(FxBE)

Таким образом,  $x \in [-66; 61]$

Значения границ значений переменной **X** хранятся в ячейках **LT** и **RT**.

Несложно заметить, что изменяться **X** может не только в основной программе, но и при обработке прерывания по ВУ-3, однако в этом случае числу **X** может быть присвоено любое значение от 0000 до 1111 (так как происходит маскирование по 4 младшим битам), поэтому этот случай всегда будет входить в ОДЗ.

### Расположение программы в памяти:

- Инициализация векторов прерывания 0x00 — 0x10
- Результат (переменная X): 0x24
- Вспомогательные данные об ОДЗ: 0x25 — 0x26
- Обработка вектора прерывания V0: 0x100 — 0x107
- Обработка вектора прерывания V1: 0x200 — 0x209
- Основная программа: 0x300 — 0x318, адрес старта программы: 0x301

### Методика проверки работы основной программы:

- 1) Не выставлять никаких флагов готовности ВУ
- 2) Записать в переменную X по адресу 0x024 минимальное по ОДЗ число, равное  $FFBE_{16} = -66_{10}$
- 3) Запустить выполнение программы в режиме “ОСТАНОВ”
- 4) Пройти нужное количество шагов программы, убедиться, что при повторном прохождении адреса 30D (когда IP = 30D), значение переменной X по адресу 0x024 равняется  $003D_{16} = 61_{10}$ .

### Методика проверки работы прерываний для ВУ:

- 1) Загрузить комплекс программ в память базовой ЭВМ.
- 2) Изменить значения точки остановок (NOP) на HLT. Они находятся по адресам: 0x106 (для обработки прерывания от ВУ-1) и 0x208 (для обработки прерывания ВУ-3)
- 3) Запустить программу в режиме “РАБОТА”.
- 4) Поставить флаг готовности ВУ-1
- 5) Дождаться остановки работы программы.
- 6) Записать текущий IP.
- 7) Записать результат обработки прерывания: значение РДВУ-1.
- 8) Записать значение переменной X из памяти БЭВМ по адресу 0x024, для этого:
  - Ввести в клавишный регистр значение 0x024.
  - Нажать «ВВОД АДРЕСА».
  - Нажать «ЧТЕНИЕ».
  - Записать содержимое DR.
- 9) Рассчитать ожидаемое значение обработки прерывания и записать его.
- 10) Восстановить значение регистра IP, ввести записанное значение в IR и после нажать «ВВОД АДРЕСА».
- 11) Продолжить работу программы в режиме “РАБОТА”, для этого необходимо нажать «ПРОДОЛЖЕНИЕ».

- 12) Ввести в РДВУ-3 произвольное число, его записать отдельно (в таблице это "РДВУ-3")
- 13) Поставить флаг готовности ВУ-3.
- 14) Дождаться остановки работы программы.
- 15) Записать текущий IP.
- 16) Записать значение переменной X до обработки прерывания из памяти БЭВМ по адресу 0x200, для этого:
  - Ввести в клавишный регистр значение 0x200.
  - Нажать «ВВОД АДРЕСА».
  - Нажать «ЧТЕНИЕ».
  - Записать содержимое DR как "X(старое)".
- 17) Рассчитать ожидаемое значение обработки прерывания.
- 18) Записать фактическое значение результата обработки прерывания, то есть новое значение переменной X, для этого надо повторить пункт 8.
- 19) Повторить пункты 10-11.
- 20) Сравнить результаты: ожидаемые значения должны совпадать с фактическими.

IP	X	РДВУ-1 (фактическое)	РДВУ-1 (ожидаемое)	IP	X (старое)	РДВУ-3	X-новое (ожидаемое)	X-новое (фактическое)
$107_{16}$	$-9_{10}$	$13_{10}$	$13_{10}$	$209_{16}$	$FFEB_{16}$	$66_{16}$	$0002_{16}$	$0002_{16}$
$107_{16}$	$-21_{10}$	$37_{10}$	$37_{10}$	$209_{16}$	$FFCD_{16}$	$B7_{16}$	$0005_{16}$	$0005_{16}$

### Выводы:

Во время выполнения лабораторной работы я изучил механизм прерываний в БЭВМ и обмен данными между ВУ и БЭВМ при работе с прерываниями.