5 МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

## **ЛАБОРАТОРНАЯ РАБОТА №6**

по дисциплине

«Основы профессиональной деятельности»

Вариант № 4

***Выполнил:***

Студент группы P3130

Захра дарабзадех

***Преподаватель:***

Саржевский Иван Анатольевич

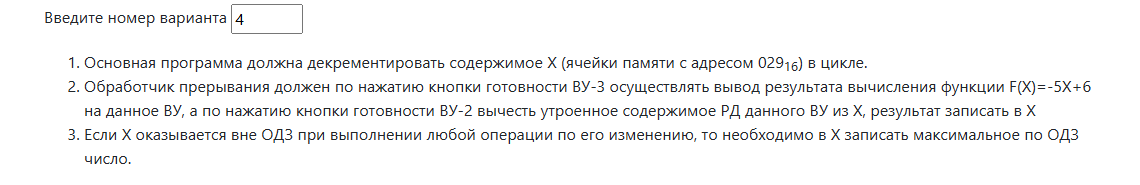
Воронина Дарья Сергеевна

Комлев Игорь Владимирович

Санкт-Петербург, 2025 г.

# Текст задания

### **Лабораторная работа №6**

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.

# Программа на Assembly

|  |
| --- |
| ORG 0x0  V0: WORD $default, 0x180 ; Вектор прерывания #0  V1: WORD $default, 0x180 ; Вектор прерывания #1  V2: WORD $int2, 0x180 ; Вектор прерывания #2  V3: WORD $int3, 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 ; обработка прерывания по умолчанию  ORG 0x029  X: WORD 0x001A ; X = +26 (начальное значение)  max: WORD 0x001A ; max = +26  min: WORD 0xFFE8 ; min = -24  tmp: WORD 0x0 ; временная ячейка  org 0x40  start: DI ; запрет прерываний  cla ; очистка аккумулятора  out 0x1 ; запрет прерываний  out 0x3  out 0xA  out 0xD  out 0x11  out 0x15  out 0x19  out 0x1D ; последний запрет прерываний  ld #0xA ; (1000|0010) = 1010 = 0xA  out 0x5  ld #0xB ; (1000|0011) = 1011 = 0xB  out 0x7  EI  main: DI ; запрет прерываний  ld $X ; загрузка значения X  dec ; X = X - 1  call check  st $X ; сохранение нового значения X  EI ; разрешение прерываний  jump main ; переход в начало цикла  int2: ; Обработка прерывания от ВУ-2  in 0x4 ; Чтение из РД ВУ-2  hlt ; Точка останова  st $tmp ; Сохранение Y во временный регистр  add $tmp ; Y + Y = 2Y  add $tmp ; 2Y + Y = 3Y  neg ; -3Y  add $X ; X = X - 3Y  call check ; Проверка выхода за ОДЗ  st $X ; Сохранение нового значения X  hlt ; Точка останова  iret ; Возврат из обработки прерывания  int3:  ; DI ; запрет прерываний  ; push ; сохранение AC в стек  ld $X ; загрузка X  hlt  neg ; -X  sub $X ; -2X  sub $X ; -3X  sub $X ; -4X  sub $X ; -5X  add #6 ; F(X) = -5X + 6  call check  hlt  out 0x6 ; вывод на ВУ-3  ; pop ; восстановление AC из стека  ; EI ; разрешение прерываний  IRET ; возврат из прерывания  ; корректно  check:  check\_min:  cmp min ; сравнение с min  bpl check\_max ; если >= min → проверить max  ld $max ; если < min → установить max  jump fini  check\_max:  cmp max ; сравнение с max  bmi fini ; если <= max → всё в порядке  ld\_max:  ld $max ; если > max → установить max  ; корректно  fini:  ret ; возврат |

# ОДЗ и Область представления

-128 <= 5x+6 <= 127

-134 <= 5x <= 121

-24 <= x <= 26

-24= 1111.1111.1110.1000 = FFE8

**Число x ϵ [0x FFE8, 0x1A]**

Описание и назначение исходных данных

min и max – границы ОДЗ;

X – условная переменная.

Область преставления исходных данных и результата

min, max и X – 16 битные знаковые числа.

# Методика проверки программы

Проверка обработки прерываний:

Загрузить текст программы в БЭВМ.

Заменить NOP по нужному адресу на HLT , чтобы можно было остановиться на важных этапах.

Запустить программу в режиме РАБОТА.

Установить "Готовность ВУ-3".

Дождаться останова программы.

Записать текущее значение X из памяти БЭВМ:

Запомнить текущее состояние счетчика команд.

Ввести в клавишный регистр значение 0x029.

Нажать «Ввод адреса».

Нажать «Чтение».

Записать значение регистра данных.

Вернуть счетчик команд в исходное состояние.

Записать результат обработки прерывания – содержимое DR контроллера ВУ-3.

Рассчитать ожидаемое значение F(X):

F(X)=−5X+6

Сравнить расчетное значение с фактическим на ВУ-3.

Нажать «Продолжение».

Ввести в ВУ-2 произвольное число Y, записать его.

Установить «Готовность ВУ-2».

Дождаться останова программы.

Записать текущее значение X из памяти БЭВМ, аналогично пункту 6.

Нажать «Продолжение».

Записать текущее значение X из памяти БЭВМ, аналогично пункту 6.

Рассчитать ожидаемое значение переменной X после обработки прерывания:

X=X−3Y

Проверка основной программы:

Загрузить текст программы в БЭВМ.

Записать в переменную X минимальное по ОДЗ значение -24.

Запустить программу в режиме "ОСТАНОВ".

Пройти нужное количество шагов программы.

Убедиться, что при увеличении X (в данном случае это не относится к вашей программе, так как у вас DEC) значение не выходит за пределы ОДЗ.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Прерывание ВУ-3 | | | Прерывание ВУ-2 | | | |
| AC(0...7)  x | Ожидание  -5\*X+6 | DR | AC (0…7)  x | DR  КВУ-2 | AC  (x – 3DR) | Результат AC (0...7) |
| 1416 (20) | FFA2 16 (-94) | A216(-94) | 116 (1) | (127) | FE8416 (-380) | 001A16 (26) |
| FF16 (-1) | 000B16 (11) | 0B16 (11) | 116 (1) |  | FFFE16 (-2) | FFFE16 (26) |
| 1816 (24) | FF8E16 (-114) | 8E16 (-114) | 116 (1) | E116(-31) | 005E16 (94) | 001A16 (26) |

|  |  |  |
| --- | --- | --- |
| Основная программа | | |
| AC X | Ожидание X-1 | AC |
| 1716 (23) | 16(22) | 16(22) |
| 1816 (24) | 17(23) | 17(23) |
| 1916 (25) | 18(24) | 18(24) |

|  |
| --- |
| Start compilation at Sun Jun 22 13:18:13 IRST 2025  Finish compilation at Sun Jun 22 13:18:13 IRST 2025  000| V0: WORD 0x10  001| WORD 0x180  002| V1: WORD 0x10  003| WORD 0x180  004| V2: WORD 0x56  005| WORD 0x180  006| V3: WORD 0x61  007| WORD 0x180  008| V4: WORD 0x10  009| WORD 0x180  00a| V5: WORD 0x10  00b| WORD 0x180  00c| V6: WORD 0x10  00d| WORD 0x180  00e| V7: WORD 0x10  00f| WORD 0x180  010| default: IRET ; type=NONADDR value=0xb00  029| X: WORD 0x1a  02a| max: WORD 0x1a  02b| min: WORD 0xffe8  02c| tmp: WORD 0x0  040| START: DI ; type=NONADDR value=0x1000  041| CLA ; type=NONADDR value=0x200  042| OUT ; type=IO value=0x1301  043| OUT ; type=IO value=0x1303  044| OUT ; type=IO value=0x130a  045| OUT ; type=IO value=0x130d  046| OUT ; type=IO value=0x1311  047| OUT ; type=IO value=0x1315  048| OUT ; type=IO value=0x1319  049| OUT ; type=IO value=0x131d  04a| LD #10 ; type=ADDR value=0xaf0a  04b| OUT ; type=IO value=0x1305  04c| LD #11 ; type=ADDR value=0xaf0b  04d| OUT ; type=IO value=0x1307  04e| EI ; type=NONADDR value=0x1100  04f| main: DI ; type=NONADDR value=0x1000  050| LD $X ; type=ADDR value=0xa029  051| DEC ; type=NONADDR value=0x740  052| CALL check ; type=ADDR value=0xde1a  053| ST $X ; type=ADDR value=0xe029  054| EI ; type=NONADDR value=0x1100  055| JUMP main ; type=ADDR value=0xcef9  056| IN ; type=IO value=0x1204  057| HLT ; type=NONADDR value=0x100  058| ST $tmp ; type=ADDR value=0xe02c  059| ADD $tmp ; type=ADDR value=0x402c  05a| ADD $tmp ; type=ADDR value=0x402c  05b| NEG ; type=NONADDR value=0x780  05c| ADD $X ; type=ADDR value=0x4029  05d| CALL check ; type=ADDR value=0xde0f  05e| ST $X ; type=ADDR value=0xe029  05f| HLT ; type=NONADDR value=0x100  060| IRET ; type=NONADDR value=0xb00  061| LD $X ; type=ADDR value=0xa029  062| HLT ; type=NONADDR value=0x100  063| NEG ; type=NONADDR value=0x780  064| SUB $X ; type=ADDR value=0x6029  065| SUB $X ; type=ADDR value=0x6029  066| SUB $X ; type=ADDR value=0x6029  067| SUB $X ; type=ADDR value=0x6029  068| ADD #6 ; type=ADDR value=0x4f06  069| CALL check ; type=ADDR value=0xde03  06a| HLT ; type=NONADDR value=0x100  06b| OUT ; type=IO value=0x1306  06c| IRET ; type=NONADDR value=0xb00  06d| CMP min ; type=ADDR value=0x7ebd  06e| BPL label=check\_max ; type=BRANCH value=0xf302  06f| LD $max ; type=ADDR value=0xa02a  070| JUMP fini ; type=ADDR value=0xce03  071| CMP max ; type=ADDR value=0x7eb8  072| BMI label=fini ; type=BRANCH value=0xf201  073| LD $max ; type=ADDR value=0xa02a  074| RET ; type=NONADDR value=0xa00 |

# Вывод

В ходе выполнения лабораторной работы я изучил обмен данными с ВУ – 2 и ВУ – 3 в режиме прерываний, также изучил цикл прерывания и циклы исполнения команды IRET