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 №65156

Выполнил:
студент группы P3119
Бардин Петр Алексеевич

Преподаватель:
Осипов Святослав Владимирович

Санкт-Петербург
2023

Содержание

1 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 декрементировать содержимое X (ячейки памяти с адресом 03716) 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-2 выполнить операцию побитового 'И' 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.

Введите номер варианта 65156

1. Основная программа должна декрементировать содержимое X (ячейки памяти с адресом 037_{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-2 выполнить операцию побитового 'И' 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.

2 Анализ

Вся информация по лабораторной размещена в системе контроля версий Git на сервисе Github: `https://github.com/BardinPetr/itmo-labs/tree/main/opd/year_1/lab_6`.

2.1 Текст программного комплекса на ассемблере БЭВМ

```
ORG 0x0
V0: WORD $OUTPUT_INT, 0x180
V1: WORD $AND_INT, 0x180
V2: WORD $DEFAULT_INT, 0x180
V3: WORD $DEFAULT_INT, 0x180
V4: WORD $DEFAULT_INT, 0x180
V5: WORD $DEFAULT_INT, 0x180
V6: WORD $DEFAULT_INT, 0x180
V7: WORD $DEFAULT_INT, 0x180

ORG 0x37
X: WORD ?

START:
  DI

  ; disable ints for dev 0,3,4
  CLA
  OUT 1
  OUT 5
  OUT 7

  LD #8   ; enable | interrupt vector #0 (OUTPUT_INT)
  OUT 0x3 ; dev 1

  LD #9   ; enable | interrupt vector #1 (AND_INT)
  OUT 0x5 ; dev 2

  EI

; func MAIN: decreases $X by 1 and sets it to 65 when not in range of [-62, 65]
; X = X-1 in [-62, 65] ? X-1 : 65
; begin func MAIN
MAIN:
  LD #65
  ST $X
MAIN_LOOP:
  ; disable interrupts to make sure
  ; that interrupt for IOdev 2 won't change X value before it is saved here
  ; otherwise will lose value updated by user
  DI

  LD $X
  DEC

  CALL WRITE_CHECKED ; check X bounds and write

  EI ; restore interrupts

  BR MAIN_LOOP
; begin func MAIN


; func DEFAULT_INT: for no-operation int vector
; begin func
DEFAULT_INT:
  IRET
; end func OUTPUT_INT
```

```
; func OUTPUT_INT: outputs value of f(AC) to IOdev #1
; begin func
OUTPUT_INT:
  DI ; disable interrupts to prevent X being changed in progress
  PUSH ; save AC
  _dbg_intv0_begin: NOP

  ; calculate formula and output
  LD $X
  CALL CALC
  _dbg_intv0_calc: NOP
  OUT 2

  POP ; restore AC
  EI  ; restore interrupts
  IRET
; end func OUTPUT_INT


; func AND_INT: reads value from IOdev 2 and sets X to (X & io)
; affects: AC
; begin func
AND_INT:
  ; disable interrupts to prevent X from being changed
  ; somewhere else after it is loaded but not updated
  DI
  PUSH ; save AC
  _dbg_intv1_begin: NOP

  ; clear AC reading is only to lower byte
  CLA
  ; read from device 1 data byte to lower byte of AC and extend sign
  IN 4
  SXTB

  ; update X
  AND $X
  CALL WRITE_CHECKED ; check X bounds and write

  _dbg_intv1_end: NOP

  POP ; restore AC
  EI ; restore interrupts
  IRET
; end func AND_INT


; func CALC: f(x) = 4*x-8
; params: AC: X
; return: AC: f(X)
; begin func CALC
CALC:
  ASL
  ASL
  SUB #8
  RET
; end func CALC


; func WRITE_CHECKED: checks that AC in range [-62, 65] and writes it to X
```

```
; params: AC: input X
; return: mem(X)
; begin func
WRITE_CHECKED:
  ; begin if
  CMP #66
  BGE INVALID_VAL
  CMP #-62
  BLT INVALID_VAL
  JUMP WRITE
  ; begin branch AC not in [-62, 65]
  INVALID_VAL:
    LD #65
  ; begin branch AC in [-62, 65]
  WRITE:
    ST $X
  ; end if

  RET
; end func WRITE_CHECKED
```

2.2 Описание алгоритма

- MAIN
  В основном цикле загружаем значение X, декрементируем, затем проверяем, что оно в требуемом диапазоне и записываем. На время этой операции придется отключить прерывания, так как если в любой момент от чтения до записи обновленного значения пользователь запросит изменение X, то при возвращении X будет восстановлен на состояние до прерывания, а следовательно, значение обновленное пользователем будет потеряно. Аналогично придется запретить прерывания в AND_INT, так как иначе сразу после операции IN пользователь сможет снова установить готовность ВУ2, а значит произойдет прерывание и аналогично его результат затрется результатом прерывания, вызванного первым по очереди.
- OUTPUT_INT
  Прерывание на вывод значения X в ВУ-1.
- AND_INT
  Прерывание по вводу с ВУ-2. Реализует функцию X = X & D, где D - введенное пользователем значение в ВУ-2. Производится расширение знака, так как вся программа оперирует однобайтовыми числами. На время выполнения блокирует прерывания.
- WRITE_CHECKED
  Производит проверку переданного значения на принадлежность ОДЗ, затем записывает в X. Если проверка не пройдена, записывает в X константу 65 по условию.
- CALC - реализация функции по заданию

2.3 Область допустимых значений и область представления F(X)

Входные данные:

1. X - 1 байт знаковое число в 2 байтовом слове
2. $4X - 8 \in [-256, 255]$
   $4X \in [-248, 263]$
   $X \in [-62, 65]$

Выходные данные:

1. 2 байт знаковое число
2. $F(X) \in [-256, 252]$

2.4 Область допустимых значений и область представления на главную программу

- Вывод значения на ВУ-1:
  1 байт, знаковое число [-256, 252]
- Ввод значение в ВУ-2:
  8 бит логических значений. Перед применением логического И с числом X производится расширение знака.

3 Трассировка

Последовательность действий:

1. Ожидание до начала цикла декремента (в трассировке с адр 046)
2. Готовность ВУ-1
3. Точка останова _dbg_intv0_begin с AC=X (в обработчике прерывания OUTPUT_INT)
4. Точка останова _dbg_intv0_end с AC=4*X-8
5. Ввод ВУ-2 + готовность
6. Точка останова _dbg_intv1_begin с AC=IO2_DR (в обработчике прерывания AND_INT)
7. Точка останова _dbg_intv1_end с AC=X & IO2_DR
8. Готовность ВУ-1
9. Точка останова _dbg_intv0_begin с AC=X (в обработчике прерывания OUTPUT_INT)
10. Точка останова _dbg_intv0_end с AC=4*X-8

∞ Выполнение трассировки

Выполняемая команда			Значения регистров после исполнения								Ячейки памяти	
Адрес	Код	Команда	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
038	1000	DI	039	1000	038	1000	000	0038	0000	0000		
039	0200	CLA	03A	0200	039	0200	000	0039	0000	0100		
03A	1301	OUT	03B	1301	03A	1301	000	003A	0000	0100		
03B	1305	OUT	03C	1305	03B	1305	000	003B	0000	0100		
03C	1307	OUT	03D	1307	03C	1307	000	003C	0000	0100		
03D	AF08	LD	03E	AF08	03D	0008	000	0008	0008	0000		
03E	1303	OUT	03F	1303	03E	1303	000	003E	0008	0000		
03F	AF09	LD	040	AF09	03F	0009	000	0009	0009	0000		
040	1305	OUT	041	1305	040	1305	000	0040	0009	0000		
041	1100	EI	042	1100	041	1100	000	0041	0009	0000		
042	AF41	LD	043	AF41	042	0041	000	0041	0041	0000		
043	E037	ST	044	E037	037	0041	000	0043	0041	0000	037	0041
045	1000	DI	046	1000	045	1000	000	0045	0041	0000		

Выполняемая команда			Значения регистров после исполнения							Ячейки памяти	
Адрес	Код	Команда	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
_dbg_intv0_begin			04E	0100	04D	0100	7FD	004D	003E	0000	7FF 037 7FE 7FD	004A 003E 01E0 003E
_dbg_intv0_calc			052	0100	051	0100	7FD	0051	00F0	0001	7FC	0050
_dbg_intv1_begin			059	0100	058	0100	7FD	0058	000F	0000	7FF 037 7FE 7FD	004A 003B 01E0 003B
_dbg_intv1_end			05B	0100	05A	0100	7FD	005A	000B	0000		
_dbg_intv0_begin			04E	0100	04D	0100	7FD	004D	0009	0000	7FC 037 7FF 7FE 7FD	005C 0009 004A 01E0 0009
_dbg_intv0_calc			052	0100	051	0100	7FD	0051	001C	0001	7FC	0050

4 Вывод

В ходе работы были рассмотрены принципы работы с управляемым по прерыванию вводом-выводом. Изучены возможности по управлению прерываниями, настройке контроллеров ВУ и процессора, а также рассмотрены схемы устройства части прерываний в подсистеме ввода-вывода. Изучены принципы оформления векторов прерывания, и решения основных сложностей при работе с множественными прерываниями.