



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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”
Тема: «Обмен данными с ВУ по прерыванию».

Вариант: 1366.

выполнил:
Студент группы Р3130
Птицын Максим Евгеньевич
Преподаватель
Ткешелашвили Нино Мерабиевна

г. Санкт-Петербург
2022 г.

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.

Введите номер варианта

1. Основная программа должна инкрементировать содержимое X (ячейки памяти с адресом $02E_{16}$) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции $F(X)=3X-5$ на данное ВУ, а по нажатию кнопки готовности ВУ-2 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 минимальное по ОДЗ число.

2 Программа

2.1 Assembler

```

                                ORG      0x0
V0:  WORD    $DEF, 0x180
V1:  WORD    $INT1, 0x180
V2:  WORD    $INT2, 0x180
V3:  WORD    $DEF, 0x180
V4:  WORD    $DEF, 0x180
V5:  WORD    $DEF, 0x180
V6:  WORD    $DEF, 0x180
V7:  WORD    $DEF, 0x180
                                ORG      0x2E
X:   WORD    0x0
LB:  WORD    0xFFD7
HB:  WORD    0x2C
DEF:  IRET
START: DI
      CLA
      OUT     0x1
      OUT     0x7
      OUT     0xB
      OUT     0xF
      OUT     0x13
      OUT     0x17
      OUT     0x1B
      OUT     0x1F
      LD      #0x9
      OUT     0x3
      LD      #0xA
      OUT     0x5
      EI
MAIN: DI
      LD      $X
      INC
      CALL    $CHECK
      ST      $X
      EI
      JUMP    $MAIN
CHECK: CMP    $LB
      BPL     PART2
      JUMP    LOAD
PART2: CMP    $HB
      BMI     RETN
LOAD:  LD     LB
RETN:  RET
INT2:  DI
      NOP
      IN      4
      AND     #0xF
      AND     $X
      ST      $X
      NOP
      EI
      IRET
INT1:  DI
      NOP
      LD      $X
      ASL
      ADD     $X
      SUB     #0x5
      OUT     2
      LD      $X
      NOP
      EI
      IRET
```

2.2 Описание программы:

Вывод текста сохранённого в массиве в формате АДР0: ДЛИНА АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ...

(выводит сначала количество символов, а потом символы в порядке возрастания: СИМВ1, СИМВ2, СИМВ3)

3 Область представления данных и область допустимых значений

3.1 Область представления:

В ячейке 360 беззнаковое 11тиразрядное 16теричное число (адрес ячейки).

В ячейках 362-363 символ строки в кодировке ISO-8859-5.

В ячейке 361, 5A1 беззнаковое 8миразрядное 16теричное число.

В дальнейших ячейках массива - беззнаковые 16теричные числа, с закодированными символами в младшем и старшем байте по одному в каждом.

3.2 ОДЗ

3.2.1 ADDR:

$$0_{16} \leq ADDR \leq 7FF_{16}$$

3.2.2 LEN:

$$0_{16} \leq LEN \leq FF_{16}$$

(на самом деле там одз нет, потому что мы выделяем маской значащие биты)

3.3 ADDR+LEN:

$$\begin{cases} 0_{16} \leq ADDR + \frac{LEN}{2} \leq 360_{16} , \\ 382_{16} \leq ADDR + \frac{LEN}{2} \leq 7FF_{16} \end{cases}$$

3.3.1 M_i :

$$20_{16} \leq M_i \leq FF_{16}$$

(имеется ввиду ограничение на младший и старший байт элементов массива)

4 Расположение программы в памяти БЭВМ:

Программы - 360_{16} - 381_{16} .

Выводимая строка - $5A1-(5A1+\frac{LEN}{2}-1)$.

5 Исполнение.

5.1 Выводимая строка:

Symbol	ISO-8859-5	UTF-8	UTF-16
В	0xB2	0xD092	0x412
Е	0xB5	0xD095	0x415
Т	0xC2	0xD0A2	0x422
В	0xB2	0xD092	0x412
Ь	0xCC	0xD0AC	0x42C

5.2 Трассировка:

Адр	Знач	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адр	Знач
364	AAFB	364	0	0	0	0	0	0	4	100		
364	AAFB	365	AAFB	5A1	5	0	FFFB	5	0	0	360	05A2
365	2FFF	366	2FFF	365	FFFF	0	FFFF	5	0	0		
366	EEFA	367	EEFA	361	5	0	FFFA	5	0	0	361	5
367	1207	368	1207	367	1207	0	367	40	4	100		
368	2F40	369	2F40	368	40	0	40	40	0	0		
369	F0FD	36A	F0FD	369	F0FD	0	369	40	0	0		
36A	AEF6	36B	AEF6	361	5	0	FFF6	5	0	0		
36B	1306	36C	1306	36B	1306	0	036B	5	0	0		
36C	200	36D	200	36C	200	0	036C	0	4	100		
36D	AAF2	36E	AAF2	5A2	B2B5	0	FFF2	B2B5	8	1000	360	05A3
36E	EEF3	36F	EEF3	362	B2B5	0	FFF3	B2B5	8	1000	362	B2B5
36F	680	370	680	36F	680	0	036F	B5B2	8	1000		
370	EEF2	371	EEF2	363	B5B2	0	FFF2	B5B2	8	1000	363	B5B2
371	1207	372	1207	371	1207	0	371	B540	8	1000		
372	2F40	373	2F40	372	40	0	40	40	0	0		
373	F0FD	374	F0FD	373	F0FD	0	373	40	0	0		
374	AEED	375	AEED	362	B2B5	0	FFED	B2B5	8	1000		
375	1306	376	1306	375	1306	0	375	B2B5	8	1000		
376	8EEA	377	8EEA	361	4	0	3	B2B5	8	1000	361	4
377	CE01	379	CE01	377	379	0	1	B2B5	8	1000		
379	1207	37A	1207	379	1207	0	379	40	4	100		
37A	2F40	37B	2F40	37A	40	0	40	40	0	0		
37B	F0FD	37C	F0FD	37B	F0FD	0	037B	40	0	0		
37C	AEE6	37D	AEE6	363	B5B2	0	FFE6	B5B2	8	1000		
37D	1306	37E	1306	37D	1306	0	037D	B5B2	8	1000		
37E	8EE2	37F	8EE2	361	3	0	2	B5B2	8	1000	361	3
37F	CEEC	36C	CEEC	37F	036C	0	FFEC	B5B2	8	1000		

Адр	Знач	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адр	Знач
36C	200	36D	200	36C	200	0	036C	0	4	100		
36D	AAF2	36E	AAF2	5A3	C2B2	0	FFF2	C2B2	8	1000	360	05A4
36E	EEF3	36F	EEF3	362	C2B2	0	FFF3	C2B2	8	1000	362	C2B2
36F	680	370	680	36F	680	0	036F	B2C2	8	1000		
370	EEF2	371	EEF2	363	B2C2	0	FFF2	B2C2	8	1000	363	B2C2
371	1207	372	1207	371	1207	0	371	B240	8	1000		
372	2F40	373	2F40	372	40	0	40	40	0	0		
373	F0FD	374	F0FD	373	F0FD	0	373	40	0	0		
374	AEED	375	AEED	362	C2B2	0	FFED	C2B2	8	1000		
375	1306	376	1306	375	1306	0	375	C2B2	8	1000		
376	8EEA	377	8EEA	361	2	0	1	C2B2	8	1000	361	2
377	CE01	379	CE01	377	379	0	1	C2B2	8	1000		
379	1207	37A	1207	379	1207	0	379	40	4	100		
37A	2F40	37B	2F40	37A	40	0	40	40	0	0		
37B	F0FD	37C	F0FD	37B	F0FD	0	037B	40	0	0		
37C	AEE6	37D	AEE6	363	B2C2	0	FFE6	B2C2	8	1000		
37D	1306	37E	1306	37D	1306	0	037D	B2C2	8	1000		
37E	8EE2	37F	8EE2	361	1	0	0	B2C2	8	1000	361	1
37F	CEEC	36C	CEEC	37F	036C	0	FFEC	B2C2	8	1000		
36C	200	36D	200	36C	200	0	036C	0	4	100		
36D	AAF2	36E	AAF2	5A4	00CC	0	FFF2	00CC	0	0	360	05A5
36E	EEF3	36F	EEF3	362	00CC	0	FFF3	00CC	0	0	362	00CC
36F	680	370	680	36F	680	0	036F	CC00	8	1000		
370	EEF2	371	EEF2	363	CC00	0	FFF2	CC00	8	1000	363	CC00
371	1207	372	1207	371	1207	0	371	CC40	8	1000		
372	2F40	373	2F40	372	40	0	40	40	0	0		
373	F0FD	374	F0FD	373	F0FD	0	373	40	0	0		
374	AEED	375	AEED	362	00CC	0	FFED	00CC	0	0		
375	1306	376	1306	375	1306	0	375	00CC	0	0		
376	8EEA	378	8EEA	361	0	0	FFFF	00CC	0	0	361	0
378	CE07	380	CE07	378	380	0	7	00CC	0	0		
380	100	381	100	380	100	0	380	00CC	0	0		

6 Доп. задание.

6.1 Задание:

Ввод двух знаковых чисел с ВУ-9 (цифровая клавиатура), вывод максимального из них на ВУ-2

```
ORG      0x0
NUM1:    WORD      0x0
NUM2:    WORD      0x0
X:        WORD      0x0
SYMBOL:  WORD      0x0
FLAG:    WORD      0x0
START:   CALL      $INIT
FIRST:   CLA
        CALL      $READ
        CALL      $SIGNFLAG
        LD        $FLAG
        BNE       CYCLE
        LD        $SYMBOL
        ST        $X
CYCLE:   CALL      $READ
        CALL      $SEPCHECK
        CALL      $ENDCHECK
        CALL      $SAVE
        JUMP      $CYCLE
FINAL:   LD        $NUM1
        CMP       $NUM2
        BMI       N2
        PUSH
        CALL      $SOUT
        POP
        JUMP      $STOP
N2:      LD        $NUM2
        PUSH
        CALL      $SOUT
        POP
        JUMP      $STOP
SOUT:    IN        0x7
        AND       #0x40
        BEQ       SOUT
        LD        (SP+1)
        OUT       0x6
        RET
STOP:    HLT
INIT:    ST        $NUM1
        ST        $NUM2
        ST        $X
        ST        $SYMBOL
        ST        $FLAG
        RET
READ:    IN        0x1D
        AND       #0x40
        BEQ       READ
        IN        0x1C
        ST        $SYMBOL
SIGNFLAG: LD        $SYMBOL
        CMP       #0xA
        BNE       FRET
        LD        #0x1
        ST        $FLAG
FRET:    RET
SEPCHECK: LD        $SYMBOL
        CMP       #0xC
        BNE       SEPRET
        LD        $X
        PUSH
        CALL      $SIGN
        POP
        ST        $NUM1
        POP
        JUMP      $FIRST
SEPRET:  RET
ENDCHECK: LD        $SYMBOL
        CMP       #0xF
        BNE       ERET
        LD        $X
        PUSH
        CALL      $SIGN
        POP
        ST        $NUM2
        POP
        JUMP      $FINAL
ERET:    RET
SAVE:    LD        $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $X
        ADD       $SYMBOL
        ST        $X
        RET
SIGN:    LD        $FLAG
        BEQ       STX
        LD        (SP+1)
        NEG
        ST        (SP+1)
        CLA
        ST        $FLAG
STX:     ST        $X
        RET
```