

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’

Вариант: 668

Выполнил:

Студент группы Р3113

Кулинич Ярослав Вадимович

Преподаватель:

Афанасьев Дмитрий Борисович



Санкт-Петербург, 2020

Содержание

1	Задание	3
1.1	Условие	3
2	Текст программы	4
3	Описание программы	9
3.1	Назначение программы	9
3.2	Область представления и область допустимых значений исходных данных и результата .	9
3.2.1	Область представления	9
3.2.2	Область допустимых значений	9
3.3	Расположение в памяти ЭВМ программы, исходных данных и результатов	9
3.3.1	Исходные данные и результат	9
3.3.2	Программа	9
3.4	Адреса первой и последней исполняемой команд.	9
4	Методика проверки	10
4.1	Основная программа	10
4.2	Обработчик ВУ-2	12
4.3	Обработчик ВУ-3	13
4.4	Обработчик по умолчанию	14
5	Вывод	14

1 Задание

1.1 Условие

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.

Enter task variant

668

1. Основная программа должна увеличивать на 2 содержимое X (ячейки памяти с адресом $01D_{16}$) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции $F(X)=3X-1$ 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 минимальное по ОДЗ число.

2 Текст программы

```
                                ORG 0x0          ; Блок инициализации векторов
VECTOR_0: WORD $DEF_HAND, 0x180 ; прерывания
VECTOR_1: WORD $DEF_HAND, 0x180
VECTOR_2: WORD $INT2, 0x180
VECTOR_3: WORD $INT3, 0x180
VECTOR_4: WORD $DEF_HAND, 0x180
VECTOR_5: WORD $DEF_HAND, 0x180
VECTOR_6: WORD $DEF_HAND, 0x180
VECTOR_7: WORD $DEF_HAND, 0x180

DEF_HAND: PUSH ; Дефолтный обработчик прерываний
ED1_HAND: IN 3 ; для ВУ-1 и ВУ-4
          AND #0x40
          BEQ ED4_HAND
          CLA
          OUT 2

ED4_HAND: IN 0xA
          AND #0x40
          BEQ END_HAND
          IN 9

END_HAND: POP
          IRET

                                ORG 0x1D ; Блок инициализации начального,
X:        WORD 0 ; макс. и мин. значений X
MIN_X:    WORD 0xFFD6
MAX_X:    WORD 0x2A

START:    DI ; Начало программы

INIT_MR:  ; Блок привязывания MR всех
          LD #8 ; КВУ к векторам прерывания
          OUT 3

          LD #0xA
          OUT 5

          CLA
          OUT 6

          LD #0xB
          OUT 7

          LD #9
          OUT 0xB

EXEC:     EI ; Начало основного цикла
          NOP ;

OPER:     DI ; Блок операции увеличения X на 2
          LD X ; и сохранения результата по адресу X,
          ADD #0x2 ; если X превысит MAX_X,
          CMP MAX_X ; то X будет присвоено MIN_X
```

```

        ST TMP_X_MID ; Сохранение промежуточных
        PUSHF        ; X и PS
        LD &O
        ST TMP_PS_MID
        LD TMP_X_MID
        POPF

        BLT SAVE_X
        BEQ SAVE_X
        LD MIN_X

SAVE_X:   ST X
        PUSHF
        LD &O
        ST TMP_PS
        POPF ; 3F

        LD MAIN_LEN ; Блок сохранения промежуточных и
        BEQ NEXT    ; конечных X и PS в таблицу

        LD TMP_X_MID ; Сохранение промежуточного X
        ST (MAIN_CUR)+

        LD TMP_PS_MID ; Сохранение промежуточного
        ST (MAIN_CUR)+ ; содержимого PS

        LD X
        ST (MAIN_CUR)+ ; Сохранение конечного X

        LD TMP_PS ; Сохранение конечного
        ST (MAIN_CUR)+ ; содержимого PS

        LOOP MAIN_LEN
        JUMP NEXT
        HLT ; БРЕЙКПОИНТ, для оставки после
            ; заполнения всей отладочной таблицы

NEXT:    EI
        NOP ; БРЕЙКПОИНТ
        JUMP OPER ; Зацикливание увеличения X на 2

TMP_X_MID: WORD 0
TMP_PS_MID: WORD 0
TMP_PS:    WORD 0

INT2:    PUSH ; Блок обработчика прерывания ВУ-2
        LD X ; Осуществляет операцию побитового И
        PUSH ; содержимого РДВУ-2 и X,
            ; а результат присваивает X

```

```

IN 0x4
PUSH
AND X
ST X

CLA
IN 5
PUSH

PUSHF

LD INT2_LEN ; Сохранение отладочной информации
BEQ NEXT_INT2

LD &3
ST (INT2_CUR)+ ; Сохранение начального X

LD &2 ; Сохранение пользовательского
ST (INT2_CUR)+ ; ввода в РДВУ-2

LD X
ST (INT2_CUR)+ ; Сохранение нового значения X

LD &0
ST (INT2_CUR)+ ; Сохранение содержимого PS

LD &1
ST (INT2_CUR)+ ; Сохранение содержимого РСВУ-2

OR -(INT2_LEN)

NEXT_INT2: POPF
POP
POP
POP
POP
POP
IRET

INT3: PUSH ; Блок обработчика прерывания ВУ-3
LD X ; Осуществляет подсчет функции  $F(X)=3X-1$ 
PUSH ; И выводит в РДВУ-3
ASL
ADD X
DEC
OUT 0x6
PUSH

CLA
IN 7
PUSH

PUSHF

```

```

LD INT3_LEN      ; Сохранение отладочной информации
BEQ NEXT_INT3

LD &3
ST (INT3_CUR)+   ; Сохранение переменной X

LD &2
ST (INT3_CUR)+   ; Сохранение результата вычисления F(X)

LD &0
ST (INT3_CUR)+   ; Сохранение содержимого PS

LD &1
ST (INT3_CUR)+   ; Сохранение содержимого PCBV-3

OR -(INT3_LEN)

NEXT_INT3:      POPF
                POP
                POP
                POP
                POP
                IRET

                ORG 0x90
MAIN_CUR:       WORD $MSTEP_1_X_MID
MAIN_LEN:       WORD 10 ; Указывает количество сохранений в таблицу
MSTEP_1_X_MID:  WORD 0; Начало блока отладочных ячеек
MSTEP_1_PS_MID: WORD 0; для основного цикла программы
MSTEP_1_X:      WORD 0
MSTEP_1_PS:     WORD 0
MSTEP_2_X_MID:  WORD 0
MSTEP_2_PS_MID: WORD 0
MSTEP_2_X:      WORD 0
MSTEP_2_PS:     WORD 0
MSTEP_3_X_MID:  WORD 0
MSTEP_3_PS_MID: WORD 0
MSTEP_3_X:      WORD 0
MSTEP_3_PS:     WORD 0
MSTEP_4_X_MID:  WORD 0
MSTEP_4_PS_MID: WORD 0
MSTEP_4_X:      WORD 0
MSTEP_4_PS:     WORD 0
MSTEP_5_X_MID:  WORD 0
MSTEP_5_PS_MID: WORD 0
MSTEP_5_X:      WORD 0
MSTEP_5_PS:     WORD 0
MSTEP_6_X_MID:  WORD 0
MSTEP_6_PS_MID: WORD 0
MSTEP_6_X:      WORD 0
MSTEP_6_PS:     WORD 0
MSTEP_7_X_MID:  WORD 0
MSTEP_7_PS_MID: WORD 0
MSTEP_7_X:      WORD 0
MSTEP_7_PS:     WORD 0
MSTEP_8_X_MID:  WORD 0
MSTEP_8_PS_MID: WORD 0
MSTEP_8_X:      WORD 0
MSTEP_8_PS:     WORD 0
MSTEP_9_X_MID:  WORD 0
MSTEP_9_PS_MID: WORD 0
MSTEP_9_X:      WORD 0
MSTEP_9_PS:     WORD 0

```

```

MSTEP_10_X_MID: WORD 0
MSTEP_10_PS_MID: WORD 0
MSTEP_10_X: WORD 0; Конец блока отладочных
MSTEP_10_PS: WORD 0; ячеек для основного цикла программы

INT2_CUR: WORD $INT2STEP_1_X_OLD; Начало блока отладочных ячеек
INT2_LEN: WORD 5; для обработчика прерываний ВУ-2
INT2STEP_1_X_OLD: WORD 0
INT2STEP_1_SX: WORD 0
INT2STEP_1_X_NEW: WORD 0
INT2STEP_1_PS: WORD 0
INT2STEP_1_SR: WORD 0
INT2STEP_2_X_OLD: WORD 0
INT2STEP_2_SX: WORD 0
INT2STEP_2_X_NEW: WORD 0
INT2STEP_2_PS: WORD 0
INT2STEP_2_SR: WORD 0
INT2STEP_3_X_OLD: WORD 0
INT2STEP_3_SX: WORD 0
INT2STEP_3_X_NEW: WORD 0
INT2STEP_3_PS: WORD 0
INT2STEP_3_SR: WORD 0
INT2STEP_4_X_OLD: WORD 0
INT2STEP_4_SX: WORD 0
INT2STEP_4_X_NEW: WORD 0
INT2STEP_4_PS: WORD 0
INT2STEP_4_SR: WORD 0
INT2STEP_5_X_OLD: WORD 0
INT2STEP_5_SX: WORD 0
INT2STEP_5_X_NEW: WORD 0
INT2STEP_5_PS: WORD 0; Конец блока отладочных ячеек
INT2STEP_5_SR: WORD 0; для обработчика прерываний ВУ-2

INT3_CUR: WORD $INT3STEP_1_X; Начало блока отладочных ячеек
INT3_LEN: WORD 5; для обработчика прерываний ВУ-3
INT3STEP_1_X: WORD 0
INT3STEP_1_SX: WORD 0
INT3STEP_1_PS: WORD 0
INT3STEP_1_SR: WORD 0
INT3STEP_2_X: WORD 0
INT3STEP_2_SX: WORD 0
INT3STEP_2_PS: WORD 0
INT3STEP_2_SR: WORD 0
INT3STEP_3_X: WORD 0
INT3STEP_3_SX: WORD 0
INT3STEP_3_PS: WORD 0
INT3STEP_3_SR: WORD 0
INT3STEP_4_X: WORD 0
INT3STEP_4_SX: WORD 0
INT3STEP_4_PS: WORD 0
INT3STEP_4_SR: WORD 0
INT3STEP_5_X: WORD 0
INT3STEP_5_SX: WORD 0
INT3STEP_5_PS: WORD 0; Конец блока отладочных ячеек
INT3STEP_5_SR: WORD 0; для обработчика прерываний ВУ-3

```


3 Описание программы

3.1 Назначение программы

Увеличение переменной X на 2 в основной программе в бесконечном цикле. Если переменная X превысит MAX_X , то установить X как MIN_X . При готовности ВУ-3 вывести на РДВУ-3 значение $F(X) = 3X - 1$. При готовности ВУ-2 выполнить операцию побитового И содержимого РДВУ-2 и X , а результат сохранить в переменную X .

3.2 Область представления и область допустимых значений исходных данных и результата

3.2.1 Область представления

X, MAX_X, MIN_X - 8-разрядные знаковые числа с фиксированной запятой. Диапазон значений формата $-2^7 \dots 2^7 - 1$

Содержимое РДВУ-2 - набор из 8 логических значений 1 или 0.

3.2.2 Область допустимых значений

Для вывода результата $F(X)$ используется 8-битный РДВУ-3, поэтому область значений функции равен $-2^7 \dots 2^7 - 1$

Или

$$\begin{aligned} -2^7 &\leq 3X - 1 \leq 2^7 - 1 \\ \frac{-2^7 + 1}{3} &\leq X \leq \frac{2^7}{3} \end{aligned}$$

Теперь в нужном формате:

$$D6 \leq X \leq 2A$$

3.3 Расположение в памяти ЭВМ программы, исходных данных и результатов

3.3.1 Исходные данные и результат

$X(1D)$ - переменная

$90 \dots B9$ - отладочная таблица для основного цикла программы

$BA \dots D4$ - отладочная таблица для обработчика прерываний ВУ-2

$D5 \dots EA$ - отладочная таблица для обработчика прерываний ВУ-3

3.3.2 Программа

$0 \dots F$ - блок инициализации векторов прерывания

$10 \dots 1B$ - обработчик прерываний по умолчанию

$MIN_X(1E), MAX_X(1F)$ - константы

$20 \dots 4F$ - основная программа

$50, 51, 52$ - локальные переменные

$53 \dots 70$ - обработчик прерываний ВУ-2

$71 \dots 8D$ - обработчик прерываний ВУ-3

3.4 Адреса первой и последней исполняемой команд.

20 - адрес первой исполняемой команды

4 Методика проверки

4.1 Основная программа

Исходные данные: $X = 20$

Ожидаемое выходное значение X : *FFDE*

- Загрузить комплекс программ в БЭВМ
- Загрузить в ячейку 1D значение 20 (присвоение $X = 20$)
- Загрузить в ячейку 91 значение A (присвоение $MAIN_LEN = A$)
- Проверить наличие команды HLT по адресу 4C
- Запустить программу и дождаться ее остановки
- Если программа слишком долго работает, то из-за неопределенной внутренней ошибки произошло заикливание и проверка провалена
- Если после остановки программы IP не равен 4D, то проверка провалена
- Обратиться к ячейкам памяти 92...B9 и заполнить таблицу ниже. При наличии расхождений, проверка провалена. Если все сошлось, проверка была осуществлена успешно.

Имя ячейки	Адрес ячейки	Значение после проверки	Ожидаемое значение
MSTEP_1_X_MID	92		0022
MSTEP_1_PS_MID	93		0188
MSTEP_1_X	94		0022
MSTEP_1_PS	95		0188
MSTEP_2_X_MID	96		0024
MSTEP_2_PS_MID	97		0188
MSTEP_2_X	98		0024
MSTEP_2_PS	99		0188
MSTEP_3_X_MID	9A		0026
MSTEP_3_PS_MID	9B		0188
MSTEP_3_X	9C		0026
MSTEP_3_PS	9D		0188
MSTEP_4_X_MID	9E		0028
MSTEP_4_PS_MID	9F		0188
MSTEP_4_X	A0		0028
MSTEP_4_PS	A1		0188
MSTEP_5_X_MID	A2		002A
MSTEP_5_PS_MID	A3		0185
MSTEP_5_X	A4		002A
MSTEP_5_PS	A5		0185
MSTEP_6_X_MID	A6		002C
MSTEP_6_PS_MID	A7		0181
MSTEP_6_X	A8		FFD6
MSTEP_6_PS	A9		0189
MSTEP_7_X_MID	AA		FFD8
MSTEP_7_PS_MID	AB		0189
MSTEP_7_X	AC		FFD8
MSTEP_7_PS	AD		0189
MSTEP_8_X_MID	AE		FFDA
MSTEP_8_PS_MID	AF		0189
MSTEP_8_X	B0		FFDA
MSTEP_8_PS	B1		0189
MSTEP_9_X_MID	B2		FFDC
MSTEP_9_PS_MID	B3		0189
MSTEP_9_X	B4		FFDC
MSTEP_9_PS	B5		0189
MSTEP_10_X_MID	B6		FFDE
MSTEP_10_PS_MID	B7		0189
MSTEP_10_X	B8		FFDE
MSTEP_10_PS	B9		0189

4.2 Обработчик ВУ-2

Исходные данные: $X = 20$

Значение 1: FF

Значение 2: BA

Значение 3: E9

Значение 4: DE

Значение 5: AD

- Загрузить комплекс программ в БЭВМ
- Загрузить в ячейку 1D значение 20 (присвоение $X = 20$)
- Загрузить в ячейку 91 значение A (присвоение $MAIN_LEN = A$)
- Загрузить в ячейку BB значение 5 (присвоение $INT2_LEN = 5$)
- Проверить наличие команды HLT по адресу 4C
- Запустить программу
- Записать в ВУ-2 "Значение 1" и установить "Готовность ВУ-2"
- Дождаться отмены "готовности ВУ-2"
- Повторить предыдущие 2 шага еще 4 раза, каждый раз используя новое "Значение"
- Дождаться остановки программы
- Если программа слишком долго работает, то из-за неопределенной внутренней ошибки произошло зацикливание и проверка провалена
- Если после остановки программы IP не равен 4D, то проверка провалена
- Обратиться к ячейкам памяти BC...D4 и заполнить таблицу ниже. При наличии расхождений, проверка провалена. Ячейки с прочерком подразумевают трудно предсказуемый результат, поэтому проверяющему нужно самому осуществить операцию побитового "И" X_OLD и SX, а после сравнить с X_NEW. Если SR не равно ожидаемому значению, то можно предположить что проверяющий слишком часто устанавливал "Готовность ВУ-2".

Имя ячейки	Адрес ячейки	Значение после проверки	Ожидаемое значение
INT2STEP_1_X_OLD	BC		—
INT2STEP_1_SX	BD		00FF
INT2STEP_1_X_NEW	BE		—
INT2STEP_1_PS	BF		—
INT2STEP_1_SR	C0		0000
INT2STEP_2_X_OLD	C1		—
INT2STEP_2_SX	C2		00BA
INT2STEP_2_X_NEW	C3		—
INT2STEP_2_PS	C4		—
INT2STEP_2_SR	C5		0000
INT2STEP_3_X_OLD	C6		—
INT2STEP_3_SX	C7		00E9
INT2STEP_3_X_NEW	C8		—
INT2STEP_3_PS	C9		—
INT2STEP_3_SR	CA		0000
INT2STEP_4_X_OLD	CB		—
INT2STEP_4_SX	CC		00DE
INT2STEP_4_X_NEW	CD		—
INT2STEP_4_PS	CE		—
INT2STEP_4_SR	CF		0000
INT2STEP_5_X_OLD	D0		—
INT2STEP_5_SX	D1		00AD
INT2STEP_5_X_NEW	D2		—
INT2STEP_5_PS	D3		—
INT2STEP_5_SR	D4		0000

4.3 Обработчик ВУ-3

Исходные данные: $X = 20$

- Загрузить комплекс программ в БЭВМ
- Загрузить в ячейку 1D значение 20 (присвоение $X = 20$)
- Загрузить в ячейку 91 значение A (присвоение $MAIN_LEN = A$)
- Загрузить в ячейку D6 значение 5 (присвоение $INT3_LEN = 5$)
- Проверить наличие команды HLT по адресу 4C
- Запустить программу
- Установить "Готовность ВУ-3"
- Дождаться отмены "готовности ВУ-3"
- Повторить предыдущие 2 шага еще 4 раза
- Дождаться остановки программы
- Если программа слишком долго работает, то из-за неопределенной внутренней ошибки произошло заикливание и проверка провалена
- Если после остановки программы IP не равен 4D, то проверка провалена
- Обратиться к ячейкам памяти $D7 \dots EA$ и заполнить таблицу ниже. При наличии расхождений, проверка провалена. Ячейки с прочерком подразумевают трудно предсказуемый результат, поэтому проверяющему нужно самому подсчитать значение $F(X)$ для X и сравнить со значением в SX . Если SR не равно ожидаемому значению, то можно предположить что проверяющий слишком часто устанавливал "Готовность ВУ-3".

Имя ячейки	Адрес ячейки	Значение после проверки	Ожидаемое значение
INT3STEP_1_X	D7		—
INT3STEP_1_SX	D8		—
INT3STEP_1_PS	D9		—
INT3STEP_1_SR	DA		0000
INT3STEP_2_X	DB		—
INT3STEP_2_SX	DC		—
INT3STEP_2_PS	DD		—
INT3STEP_2_SR	DE		0000
INT3STEP_3_X	DF		—
INT3STEP_3_SX	E0		—
INT3STEP_3_PS	E1		—
INT3STEP_3_SR	E2		0000
INT3STEP_4_X	E3		—
INT3STEP_4_SX	E4		—
INT3STEP_4_PS	E5		—
INT3STEP_4_SR	E6		0000
INT3STEP_5_X	E7		—
INT3STEP_5_SX	E8		—
INT3STEP_5_PS	E9		—
INT3STEP_5_SR	EA		0000

4.4 Обработчик по умолчанию

- Загрузить комплекс программ в БЭВМ
- Проверить наличие команды HLT по адресу 4C
- Запустить программу
- Установить "Готовность ВУ-1"
- Дождаться сброса "Готовность ВУ-1" если этого не происходит длительное время, программа за-
циклена, проверка провалена.

Проверить ВУ-4 не представляется возможным, ибо при нажатии на соответствующую кнопку ВУ-4 не открывается.

5 Вывод

В ходе выполнения лабораторной работы я изучил организацию ввода-вывода при помощи прерываний в БЭВМ, разработал программу, в которой реализовал тривиальные обработчики прерываний, а также разработал методику проверки работоспособности своей работы.