

**Министр науки и высшего образования Российской
Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Домашняя работа № 3

Программирование обмена данными с внешними устройствами

Выполнил студент группы № М3101

Михеев Артем Романович

Подпись:



Проверил:

Бабич Мария Сергеевна

Санкт-Петербург
2020

Цель работы

Написать комплекс программ, рассчитывающих разные математические выражения, использующие изменяющуюся переменную, в зависимости от того, с какого ВУ был послан сигнал. Обмен данными с ВУ должен происходить в режиме прерывания, а вывод данных в асинхронном режиме. Переменная должна увеличиваться циклом на 1 при каждом проходе, кол-во инструкций в цикле должно быть не больше 3.

Задание, вариант 6

Составить методику проверки правильности выполнения разработанного комплекса на БЭВМ, получить заданное количество результатов для возможных запросов прерывания от имеющихся ВУ (ВУ-1, ВУ-2, ВУ-3).

Вариант 6:

По запросу ВУ-1 вывести $(5X-1)/2$, а по запросу ВУ-3 вывести $(X/2)-6$. Оба результата выводить на ВУ-3.

Решение

1. Исходный код всей программы получился таким:

```
ORG 0000
INTERRUPT_RET: WORD 0000
BR HANDLE_INTERRUPT           # при прерывании мы переходим к другой функции

ORG 0020
BEGIN:                        # основная программа
    EI
LOOP_ACCUMULATE:             # цикл в 3 команды изменяющий нашу переменную
    ISZ ACCUMULATOR
    NOP
    BR LOOP_ACCUMULATE

ORG 0040
HANDLE_INTERRUPT:
    TSF 1                     # если прерывание от не ВУ-1, то проверить дальше
    BR CHECK_IN_3
HANDLE_IN_1:
    CLA
    ADD MULTIPLY_ACC_BY_2_ADDR # для функции CALL_FN_ANY_SIGN загрузим адрес
                                вызываемой функции (умн. на 2)
    MOV CALL_FN_ADDR

    CLA
    ADD ACCUMULATOR
    JSR CALL_FN_ANY_SIGN      #  $X*2=2X$ 
    JSR CALL_FN_ANY_SIGN      #  $2X*2=4X$ 
    ADD ACCUMULATOR          #  $4X+X=5X$ 
```

INC	# 5X+1
MOV TEMPORARY	
CLA	
ADD DIVIDE_ACC_BY_2_ADDR	# загрузим адрес функции деления на 2
MOV CALL_FN_ADDR	
CLA	
ADD TEMPORARY	
JSR CALL_FN_ANY_SIGN	# (5X+1)/2
OUT 3	# вывод, очистка флага у ВУ-1, возвращение
CLF 1	
BR RETURN	
CHECK_IN_3:	
TSF 3	# иначе проверим если это запрос от ВУ-3
BR RESET_2	
HANDLE_3:	
CLA	
ADD DIVIDE_ACC_BY_2_ADDR	# X / 2
MOV CALL_FN_ADDR	
CLA	
ADD ACCUMULATOR	
JSR CALL_FN_ANY_SIGN	
SUB CONST_SIX	# X/2-6
OUT 3	# вывод, очистка флага у ВУ-3, возвращение
CLF 3	
BR RETURN	
RESET_2:	
CLF 2	# иначе запрос от ВУ-2, просто очистим у него флаг
RETURN:	
EI	
BR (INTERRUPT_RET)	
ORG 0080	
CALL_FN_ADDR: WORD FFFF	
CALL_FN_ANY_SIGN: WORD 0000	
CALL_FN_ANY_SIGN_START:	# функция которая при отриц. арг. сначала переведет его в полож., потом вызовет нужную функцию и переведет обратно
BPL POSITIVE_CASE	
JSR CONVERT_ACC_SIGN	
JSR (CALL_FN_ADDR)	
JSR CONVERT_ACC_SIGN	

```

    BR (CALL_FN_ANY_SIGN)
POSITIVE_CASE:
    JSR (CALL_FN_ADDR)
    BR (CALL_FN_ANY_SIGN)

```

```

ORG 0090
CONVERT_ACC_SIGN: WORD 0000
CONVERT_ACC_SIGN_START:      # работает как для полож. так и для отриц.
    CMA
    INC
    BR (CONVERT_ACC_SIGN)

```

```

ORG 00A0
DIVIDE_ACC_BY_2: WORD 0000
DIVIDE_ACC_BY_2_START:
    CLC
    ROR
    BR (DIVIDE_ACC_BY_2)

```

```

ORG 00B0
MULTIPLY_ACC_BY_2: WORD 0000
MULTIPLY_ACC_BY_2_START:
    CLC
    ROL
    BR (MULTIPLY_ACC_BY_2)

```

data

```

ORG 0100
ACCUMULATOR: WORD 0000
CONST_SIX: WORD 0006
TEMPORARY: WORD 0000

```

```

FUNCTION_TABLE:                # адреса функций для использования
                                CALL_FN_ANY_SIGN

DIVIDE_ACC_BY_2_ADDR: WORD 00A0
MULTIPLY_ACC_BY_2_ADDR: WORD 00B0

```

2. Теперь составим методику проверки корректности исполнения программы:
 - 1) Перевести предоставленный выше исходный код программы в их обозначения в виде машинных слов (ассемблировать), после чего занести их в память БЭВМ.
 - 2) Запустить основную программу с адреса 0020 в автоматическом режиме.
 - 3) Подождать некоторое время для более наглядных результатов, установить “Готовность ВУ-1”
 - 4) Дождаться сброса “Готовность ВУ-1”, в ВУ-3 к этому моменту будет записано значение выражения $(5X+1)/2$ на момент установления “Готовность ВУ-1”.
 - 5) Установить “Готовность ВУ-2”

- 6) Дождаться сброса “Готовность ВУ-2”, убедиться в том, что данные в ВУ-3 не изменились, так как при сигнале от ВУ-2 программа не должна ничего высчитывать.
 - 7) Установить “Готовность ВУ-3”
 - 8) Дождаться сброса “Готовность ВУ-3”, в ВУ-3 к этому моменту будет записано значение выражения $(X/2)-6$ на момент установления “Готовность ВУ-3”.
 - 9) Для проверки правильности результатов установить меньшую тактовую частоту, что позволит понять, какое значение X было перед установкой Готовности какого-либо из ВУ, для того чтобы подтвердить, что после сброса соотв. Готовности, в ВУ-3 записан правильный результат (проделать шаги 3-8 снова, но теперь проверять значение).
3. Проверив сначала правильность программы по шагам 1-8 предоставленной методики, я убедился, что написанная программа корректно реагирует на необходимые прерывания. Для проверки же правильности именно результатов вычислений для начала просмотрю значение X и значение выражения при прерывании ВУ-1, а потом значение X и значение выражения при прерывании ВУ-3:
- | | |
|---|----------------------------------|
| ВУ-1 – $X = 7$, $(5X+1) / 2 = 00010010 = 18$ | $[(5*7+1) / 2 = 18$, правильно] |
| ВУ-3 – $X = 16$, $(X/2)-6 = 00000010 = 2$ | $[16 / 2 - 6 = 2$, правильно] |

Выводы

При решении этого домашнего задания потребовались как ранее полученные навыки написания подпрограмм для БЭВМ, так и новые знания о различных режимах взаимодействия с ВУ. Использовался как режим прерываний, так и асинхронный (для вывода) режим взаимодействия с ВУ. Всё это позволит в будущем гораздо проще понять, как работают аналогичные вещи в настоящих современных системах.