Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
  
Факультет программной инженерии и компьютерной техники  
  
Направление подготовки 09.03.04 «Программная инженерия» – Системное и прикладное программное обеспечение

**Отчёт  
По лабораторной работе №7**

По дисциплине «Основы профессиональной деятельности»  
**Вариант: 131313475, 11666**

Выполнил:  
Студент 1 курса  
Васильев Артём Евгеньевич

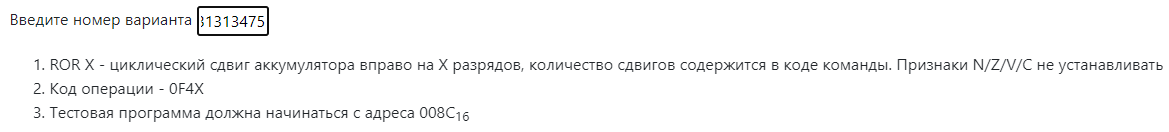
Группа: 3119

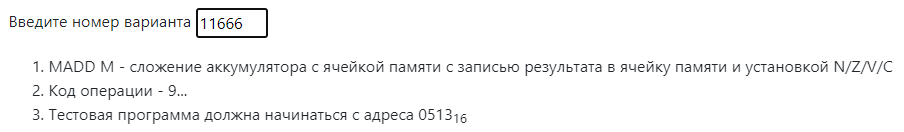
Преподаватель:

Перцев Тимофей Сергеевич

2024 г.

# Задание





# Основные этапы вычисления

**Задание 1**

# **Выполнение**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес МП | Микрокоманда | Описание | Комменатрий |
| E0 | 81E1041002 | if CR(6) = 0 then GOTO INT @ C4 | Проверяем МК |
| E1 | 0020002002 | LTOH(CR) -> BR | Находим X из OF4X через перенос в старшую часть и сдвигом старшей тетрады за границы числа |
| E2 | 0020020020 | SHL(BR) -> BR |
| E3 | 0020020020 | SHL(BR) -> BR |
| E4 | 0020020020 | SHL(BR) -> BR |
| E5 | 0020020020 | SHL(BR) -> BR |
| E6 | 0020004020 | HTOL(BR) -> BR | Переносим в младшую часть и возвращаем в нормальное состояние сейчас число выглядит как X0 |
| E7 | 0020080020 | ASR(BR) -> BR |
| E8 | 0020080020 | ASR(BR) -> BR |
| E9 | 0020080020 | ASR(BR) -> BR |
| EA | 0020080020 | ASR(BR) -> BR |
| EB | 0020009120 | BR + ~0 -> BR | Декремент X |
| EC | 81C4804020 | if BR(15) = 1 then GOTO INT @ C4 | Если X < 0, то на цикл прерывания |
| ED | 0010380010 | ROR(AC) -> AC, C | Циклический сдвиг вправо |
| EE | 80EB101040 | GOTO EB | Повторяем, пока X >= 0 |

**Тестовая программа**

ORG 0x0079

test\_number1: WORD 0xFBCD

test\_number2: WORD 0x1

test\_number3: WORD 0x9999

res\_num1: WORD 0

res\_num2: WORD 0

res\_num3: WORD 0

res1: WORD 0

res2: WORD 0

res3: WORD 0

all\_res: WORD 0

curr\_res1: WORD 0xFBCD ;для сдвига 0

curr\_res2: WORD 0x0 ;для сдвига 1

curr\_res3: WORD 0x3329 ;для сдвига 12

ORG 0x008С

START:

CALL test1

CALL test2

CALL test3

CALL check\_all\_res

JUMP stop

test1:

LD test\_number1

WORD 0x0F40

ST res\_num1

CMP curr\_res1

CALL check

ST res1

CLC

RET

test2:

LD test\_number2

WORD 0x0F41

ST res\_num2

CMP curr\_res2

CALL check

ST res2

CLC

RET

test3:

LD test\_number3

WORD 0x0FFC

ST res\_num3

CMP curr\_res3

CALL check

ST res3

CLC

RET

check: BZS setYes

LD #0x0

RET

setYes: LD #0x1

RET

check\_all\_res: LD res1

AND res2

AND res3

ST all\_res

RET

stop: HLT

**Методика проверки:**

1. Ввести тестовую программу в БЭВМ

2. Запустить программу в режиме «Работа»

3. Дождаться конца выполнения программы.

4. Проверить число в регистре АС, или в ячейке 0x082, если оно равно 0x1, значит, программа выполнилась корректно.

Если при выполнении программы в одном и тестов возникла ошибка, нужно ввести в **IR**

0x07F нажать кнопку «Ввод адреса», затем кнопку «Чтение»,

проверить значения ячеек

0x07F – корректное выполнение первого теста

0x080 – корректное выполнение второго теста,

0x081 – корректное выполнение третьего теста.

5. Для проверки правильности вычислений ввести в **IR** 0x079 , нажать кнопку «Ввод адреса», затем кнопку «Чтение» и посмотреть содержимое этих ячеек. В каждой из них будут результаты вычислений.

**Задание 2**

# **Выполнение**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес МП | Микрокоманда | Описание | Комменатрий |
| 3D | 81F0104002 | if CR(12) = 1 then GOTO F0 | Переход в F0, для команды с кодом 9 |
| F0 | 0201E09011 | AC + DR -> DR, N, Z, V, C; DR -> MEM(AR) | Складываем АС с данными из ячейки |
| F1 | 80C4101040 | GOTO INT @ C4 | Переход на цикл прерывания |

**Тестовая программа**

ORG 0x04EB

test\_number1: WORD 0x0537

test\_number2: WORD 0xFBCD

test\_number3: WORD 0x5

test\_number4: WORD 0x5

num1: WORD 0x2

num2: WORD 0x5

num3: WORD 0xFFFB

res1: WORD 0

res2: WORD 0

res3: WORD 0

res4: WORD 0

all\_res: WORD 0

curr\_res1: WORD 0x0539 ; тест для прямой относительной (+num1)

curr\_res2: WORD 0xFBD2 ; тест для прямой абсолютной (+num2)

curr\_res3: WORD 0 ; тест для прямой относительной (+num3)

curr\_res4: WORD 0xA ; тест для стека (+num2)

ORG 0x0513

START:

JUMP test1

test1:

LD $num1

WORD 0x94EB

LD test\_number1

CMP curr\_res1

CALL check

ST res1

JUMP test2

test2:

LD $num2

WORD 0x94EC

LD test\_number2

CMP curr\_res2

CALL check

ST res2

JUMP test3

test3:

LD $num3

WORD 0x9EC9

LD test\_number3

CMP curr\_res3

CALL check

ST res3

JUMP test4

test4:

LD $test\_number4

PUSH

LD $num2

WORD 0x9C00

POP

ST test\_number4

CMP curr\_res4

CALL check

ST res4

JUMP check\_all\_res

check: BZS setYes

LD #0x0

RET

setYes: LD #0x1

RET

check\_all\_res: LD res1

AND res2

AND res3

AND res4

ST all\_res

JUMP stop

stop: HLT

**Методика проверки:**

1. Ввести тестовую программу в БЭВМ

2. Запустить программу в режиме «Работа»

3. Дождаться конца выполнения программы.

4. Проверить число в регистре АС, или в ячейке 0x4F6, если оно равно 0x1, значит, программа выполнилась корректно.

Если при выполнении программы в одном и тестов возникла ошибка, нужно ввести в **IR**

0x4F2 нажать кнопку «Ввод адреса», затем кнопку «Чтение»,

проверить значения ячеек

0x4F2 – корректное выполнение первого теста

0x4F3 – корректное выполнение второго теста,

0x4F4 – корректное выполнение третьего теста,

0x4F5 – корректное выполнение четвёртого теста,

5. Для проверки правильности вычислений ввести в **IR** 0x04EB , нажать кнопку «Ввод адреса», затем кнопку «Чтение» и посмотреть содержимое этих ячеек. В каждой из них будут результаты вычислений, поскольку по условию число загружается из ячейки, суммируется и загружается обратно в ячейку. (в ячейку 4ЕЕ был записан результат при помощи ST, поскольку в 4 тесте используется адресация относительно SP)

# Заключение

В ходе выполнения данной лабораторной работы я познакомился с ВУ в БЭВМ. Научился с ними работать, выводить значения, а также писать код на ассемблере.