**ФЕДЕРАЛНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИСЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: «Основы Профессиональной Деятельности»

**ОТЧЁТ**

по лабораторной работе №7  
Вариант №18003, 18004

Выполнил:

Студент группы P3110

Федоров Евгений Константинович

Проверил:

Перцев Тимофей Сергеевич

Санкт-Петербург 2024

**Оглавление**

[**Задание №1** 3](#_Toc167185474)

[**Выполнение** 3](#_Toc167185475)

[**Тестовая программа на ассемблере** 4](#_Toc167185476)

[**Методика проверки** 6](#_Toc167185477)

[**Задание №2** 7](#_Toc167185478)

[**Тестовая программа на ассемблере** 8](#_Toc167185479)

[**Методика проверки** 12](#_Toc167185480)

[**Вывод** 13](#_Toc167185481)

# **Задание №1**

Синтезировать цикл исполнения для выданных преподавателем команд. Разработать тестовые программы, которые проверяют каждую из синтезированных команд. Загрузить в микропрограммную память БЭВМ циклы исполнения синтезированных команд, загрузить в основную память БЭВМ тестовые программы. Проверить и отладить разработанные тестовые программы и микропрограммы.

1. BGC ADDR - переход к команде, расположенной по адресу, на которую указывает адресная часть команды, если аккумулятор содержит число, большее чем 4095
2. Код операции - FDXX
3. Тестовая программа должна начинаться с адреса 01B616

# **Выполнение**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес МП | Микрокоманда | Описание | Комменатрий |
| E0 | 80C4084002 | if CR(11) = 0 then GOTO INT @ C4 | Проверка на то, что мы работаем с FD |
| E1 | 80C4044002 | if CR(10) = 0 then GOTO INT @ C4 |
| E2 | 81C4024002 | if CR(9) = 1 then GOTO INT @ C4 |
| E3 | 80C4014002 | if CR(8) = 0 then GOTO INT @ C4 |
| E4 | 81C4804010 | if AC(15) = 1 then GOTO INT @ C4 | Проверка на то, чтобы число было неотрицательным |
| E5 | 81F8404010 | if AC(14) = 1 then GOTO E8 | Проверка на то, чтобы число в акумуляторе было больше, чем 4095 |
| E6 | 81F8204010 | if AC(13) = 1 then GOTO E8 |
| E7 | 80C4104010 | if AC(12) = 0 then GOTO INT @ C4 |
| E8 | 805C101040 | GOTO BR @ 5C | Переход на цикл исполнения безусловного перехода (IP + XX) |

# **Тестовая программа на ассемблере**

ORG 0x1AF

test1\_number: WORD 0x0FFF ;4095

test2\_number: WORD 0x1000 ;4096

test1\_result: WORD 0

test2\_result: WORD 0

ORG 0x1B6

START: LD $test1\_number

WORD 0xFD03 ; if command works wrong it will go to test\_exception

right\_execution:

LD #0x1

ST $test1\_result

JUMP second\_test

test\_exception:

CLA

ST $test1\_result

JUMP second\_test

second\_test: LD $test2\_number

WORD 0xFD03 ; if command works right it will go to the second\_part

second\_right\_execution:

CLA

ST $test2\_result

JUMP check\_status

second\_part:

LD #0x1

ST $test2\_result

CLA

JUMP check\_status

check\_status:

LD $test1\_result

AND $test2\_result

CMP #0x1

BEQ succed

LD #0xF

HLT

succed:

LD #0x1

HLT

# **Методика проверки**

1. Запустить БЭВМ через терминал в режиме Dual при помощи команды java –Dmode=dual –jar bcomp-ng.jar
2. В терминале ввести следующую последовательность команд:

ma

mw 80C4084002

mw 80C4044002

mw 81C4024002

mw 80C4014002

mw 81C4804010

mw 81E8404010

mw 81E8204010

mw 80C4104010

mw 805C101040

1. Загрузить тестовую программу в БЭВМ. Для этого перейти во вкладку «Ассемблер», вставить скопированную программу и нажать кнопку «Компилировать»
2. Переключить тумблер «Работа/Останов» в режим «Работа»
3. Нажать кнопку пуск
4. Дождаться завершения программы
5. Посмотреть число в регистре AC. Если оно равно 1 (0000 0000 0000 0001 в 2-чной системе счисления), то оба теста прошли успешно. Иначе, один из тестов был провален. Для того, чтобы зафиксировать какой из, сделать следующую последовательность действий:
   * + 1. Ввести адрес 0x1B1 (0000 0001 1011 0001) в клавишный регистр IR и нажать кнопку «Чтение»
       2. Посмотреть на регистр DR. Если его значение равно 1 (0000 0000 0000 0001), то тест на **переход при числе меньшем либо равным, чем 4095** был пройден успешно. Если его значение равно 0 (0000 0000 0000 0000), то тест был провален.
       3. Повторить последовательность действий из пункта 7.1 для адреса 0x1B2 (0000 0001 1011 0010)
       4. Посмотреть на регистр DR. Если его значение равно 1 (0000 0000 0000 0001), то тест на **переход при числе большем, чем 4095** был пройден успешно. Если его значение равно 0 (0000 0000 0000 0000), то тест был провален.

# **Задание №2**

Синтезировать цикл исполнения для выданных преподавателем команд. Разработать тестовые программы, которые проверяют каждую из синтезированных команд. Загрузить в микропрограммную память БЭВМ циклы исполнения синтезированных команд, загрузить в основную память БЭВМ тестовые программы. Проверить и отладить разработанные тестовые программы и микропрограммы.

1. LDNEG M - загрузка с изменением знака. Записать в аккумулятор содержимое ячейки, с измененным знаком, на которую указывает адресная часть команды. Установить признаки N/Z
2. Код операции - 9...
3. Тестовая программа должна начинаться с адреса 016716

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес МП | Микрокоманда | Описание | Комменатрий |
| 3D | 81F0104002 | if CR(12) = 1 then GOTO F0 | Переход в F0, для команды с кодом 9 |
| F0 | 0010809501 | ~DR + 1 ? AC, N, Z | Перевод значения DR в доп. код , запись результата в AC, выставление флагов N,Z |
| F1 | 80C4101040 | GOTO INT @ C4 | Переход на цикл прерывания |

# 

# **Тестовая программа на ассемблере**

ORG 0x0159

test\_number1: WORD 0

test\_number2: WORD 0xFBCD

test\_number3: WORD 0x5

test1\_res: WORD 0

test2\_res: WORD 0

test3\_res: WORD 0

test4\_res: WORD 0

test5\_res: WORD 0

testbuf1: WORD 0

testbuf2: WORD 0x0433

testbuf3: WORD 0xFFFB

testbuf4: WORD 0xFFFF

testbuf5: WORD 0xFF8E

ORG 0x0167

START:

WORD 0x9159

CMP $testbuf1

zero\_test:

BEQ first\_success

CLA

ST $test1\_res

JUMP positive\_test

positive\_test:

WORD 0x915A

CMP $testbuf2

BEQ second\_success

CLA

ST $test2\_res

JUMP negative\_test

negative\_test:

WORD 0x915B

CMP $testbuf3

BEQ third\_success

CLA

ST $test3\_res

JUMP e\_adress\_test

WORD 0x1

e\_adress\_test:

WORD 0x9EFE

CMP $testbuf4

BEQ fourth\_success

CLA

ST $test4\_res

JUMP stack\_test

stack\_test:

LD #0x72

PUSH

WORD 0x9C00

CMP $testbuf5

BEQ fifth\_success

CLA

ST $test5\_res

JUMP check\_res

check\_res:

CLA

LD $test1\_res

AND $test2\_res

AND $test3\_res

AND $test4\_res

AND $test5\_res

BEQ incorrect\_res

HLT

incorrect\_res:

LD #0x0

HLT

first\_success:

LD #0x1

ST $test1\_res

JUMP positive\_test

second\_success:

LD #0x1

ST $test2\_res

JUMP negative\_test

third\_success:

LD #0x1

ST $test3\_res

JUMP e\_adress\_test

fourth\_success:

LD #0x1

ST $test4\_res

JUMP stack\_test

fifth\_success:

LD #0x1

ST $test5\_res

JUMP check\_res

# **Методика проверки**

1. Запустить БЭВМ через терминал в режиме Dual при помощи команды java –Dmode=dual –jar bcomp-ng.jar
2. В терминале ввести следующую последовательность команд:

3d

ma

mw 81F0104002

f0

ma

mw 0010809501

mw 80C4101040

1. Загрузить тестовую программу в БЭВМ. Для этого перейти во вкладку «Ассемблер», вставить скопированную программу и нажать кнопку «Компилировать»
2. Переключить тумблер «Работа/Останов» в режим «Работа»
3. Нажать кнопку пуск
4. Дождаться завершения программы
5. Посмотреть число в регистре AC. Если оно равно 1 (0000 0000 0000 0001 в 2-чной системе счисления), то все 5 тестов прошли успешно. Иначе, некоторые из тестов были провалены. Для того, чтобы зафиксировать какой из, сделать следующую последовательность действий:
6. Ввести адрес 0x15C (0000 0001 0101 1100) в клавишный регистр IR и нажать кнопку «Чтение»
7. Посмотреть на регистр DR. Если его значение равно 1 (0000 0000 0000 0001), то тест на правильный перевод в дополнительный код был пройден успешно. Если его значение равно 0 (0000 0000 0000 0000), то тест был провален.
8. Повторить последовательность действий из пунктов 7.1 и 7.2 для ячеек 0x15D, 0x15E, 0x15F, 0x160

# **Вывод**

В ходе выполнения лабороторной работы научился синтезировать команды в БЭВМ. Разобрался в работе ОМК и УМК.