Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Дисциплина «Основы профессиональной деятельности»

**Лабораторная работа №7**

По дисциплине

**«Основы профессиональной деятельности»**

Вариант: 3175

Выполнил

Колмаков Дмитрий Владимирович,

Группа Р3131

Преподаватель

Перцев Тимофей Сергеевич

г. Санкт-Петербург, 2023 г

Оглавление

[Задание 3](#_Toc135874694)

[Ход работы 4](#_Toc135874695)

[Назначение программы 4](#_Toc135874696)

[ОДЗ 4](#_Toc135874697)

[Расположение данных в памяти 4](#_Toc135874698)

[Область представления 4](#_Toc135874699)

[Код программы на ассемблере 5](#_Toc135874700)

[Вывод 7](#_Toc135874701)

# Задание

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. ANDSP - Логическое И двух верхних чисел на вершине стека, результат поместить на стек, установить признаки N/Z
2. Код операции - 0F10
3. Тестовая программа должна начинаться с адреса 042716

# Ход работы

## Исходный код синтезируемой команды

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Микрокоманда | Описание | Комментарий |
| E0 | 0180009008 | SP → AR; MEM(AR) → DR | Первое число из стека → BR |
| E1 | 0020009001 | DR → BR |
| E2 | 0180009408 | (SP+1) → AR; MEM(AR) → DR | Второе число из стека → DR |
| E3 | 0001809821 | (BR & DR) → DR, N, Z | Логическое И двух верхних чисел на вершине стека, флаги N, Z |
| E4 | 0288009208 | (SP + ~0) → SP, AR; DR → MEM(AR) | SP = SP - 1, результат операции на стек |
| E5 | 80C4101040 | GOTO INT @ C4 | Переход к циклу прерываний |

## Текст тестовой программы

ORG 0x010

ARG1: WORD 0xFFFF

ARG2: WORD 0x0000

ARG3: WORD 0x1234

ARG4: WORD 0x1234

ARG5: WORD 0xF91A

ARG6: WORD 0x3C78

CHECK1: WORD 0x0

CHECK2: WORD 0x0

CHECK3: WORD 0x0

FINAL: WORD 0x0

RES1: WORD 0x0000

RES2: WORD 0x1234

RES3: WORD 0x3818

TESTRES1: WORD 0x0

TESTRES2: WORD 0x0

TESTRES3: WORD 0x0

START: CLA

CALL TEST1

CALL TEST2

CALL TEST3

LD #0x1

AND CHECK1

AND CHECK2

AND CHECK3

ST FINAL

STOP: HLT

TEST1: LD ARG1

PUSH

LD ARG2

PUSH

CLA

WORD 0x0F10 ; ANDSP

POP

ST TESTRES1

CMP RES1

BEQ DONE1

ERROR1: POP

POP

CLA

RET

DONE1: POP

POP

LD #0x1

ST CHECK1

CLA

RET

TEST2: LD ARG3

PUSH

LD ARG4

PUSH

CLA

WORD 0x0F10 ; ANDSP

POP

ST TESTRES2

CMP RES2

BEQ DONE2

ERROR2: POP

POP

CLA

RET

DONE2: POP

POP

LD #0x1

ST CHECK2

CLA

RET

TEST3: LD ARG5

PUSH

LD ARG6

PUSH

CLA

WORD 0x0F10 ; ANDSP

POP

ST TESTRES3

CMP RES3

BEQ DONE3

ERROR3: POP

POP

CLA

RET

DONE3: POP

POP

LD #0x1

ST CHECK3

CLA

RET

## Таблица трассировки МК

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МР до выборки МК | Содержимое памяти и регистров процессора после выборки микрокоманды | | | | | | | | | |
| **MR** | **IP** | **AR** | **AR** | **DR** | **SP** | **BR** | **AC** | **NZVC** | **СчМК** |
| E0 | 0180009008 | 04E | 0F10 | 7FD | 3C78 | 7FD | 004D | 0000 | 0100 | E1 |
| E1 | 0020009001 | 04E | 0F10 | 7FD | 3C78 | 7FD | 3C78 | 0000 | 0100 | E2 |
| E2 | 0180009408 | 04E | 0F10 | 7FE | F91A | 7FD | 3C78 | 0000 | 0100 | E3 |
| E3 | 0001809821 | 04E | 0F10 | 7FE | 3818 | 7FD | 3C78 | 0000 | 0000 | E4 |
| E4 | 0288009208 | 04E | 0F10 | 7FC | 3818 | 7FC | 3C78 | 0000 | 0000 | E5 |
| E5 | 80C4101040 | 04E | 0F10 | 7FC | 3818 | 7FC | 3C78 | 0000 | 0000 | C4 |
| C4 | 80DE801040 | 04E | 0F10 | 7FC | 3818 | 7FC | 3C78 | 0000 | 0000 | C5 |
| C5 | 8001401040 | 04E | 0F10 | 7FC | 3818 | 7FC | 3C78 | 0000 | 0000 | 01 |

# Методика проверки

1. Загрузить комплекс разработанных микропрограмм в микропрограммную память БЭВМ;
2. Загрузить тестовую программу в память базовой ЭВМ;
3. Запустить основную программу с адреса в режиме работа;
4. Дождаться останова;
5. Проверить результат в AC и флаги N, Z;
6. Нажать продолжить;
7. Перейти к шагу 4 еще два раза;
8. Проверить значение ячейки памяти FINAL с номером 0x019, если значение 0x0001 – все тесты выполнены успешно.

Комментарии к методике

* Для проверки используется три пары значений: 0xFFFF и 0x0000, 0x1234 и 0x1234, 0xF91A и 0x3C78
* Данные значения показывают правильную работу программы логического умножения числа на ноль, числа на само число, двух разных ненулевых чисел
* В ходе проверки флаги N, Z выставляются корректно
* Результат каждого теста записывается в соответствующую ячейку TESTRES, значение ячейки RES 0x0001 означает успешное выполнение. 0x0000 – ошибку при выполнении.
* При успешном выполнении всех тестов значение FINAL станет 0x0001, иначе 0x0000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ячейка с результатом** | | **Первое число** | **Второе число** | **Теоретический результат** | **Полученный результат** |
| TESTRES1 | 0x01D | 0xFFFF (N=0, Z=1) | 0x0000 (N=0, Z=1) | 0x0000 (N=0, Z=1) | 0x0000 (N=0, Z=1) |
| TESTRES2 | 0x01E | 0x1234 (N=0, Z=0) | 0x1234 (N=0, Z=0) | 0x1234 (N=0, Z=0) | 0x1234 (N=0, Z=0) |
| TESTRES3 | 0x01F | 0xF91A (N=1, Z=0) | 0x3C78 (N=0, Z=0) | 0x3818 (N=0, Z=0) | 0x3818 (N=0, Z=0) |

# Вывод

В ходе выполнения лабораторной работы я изучил алгоритм синтеза собственной команды БЭВМ с помощью горизонтальных микрокоманд, обратился к знаниям из курса дискретной математики и реализовал на практике алгоритм умножения со сдвигом СЧП вправо, разработал методику проверки сделанной программы, а также получил полное понимание устройства БЭВМ и желание разработать новые команды и модули для уже существующей БЭВМ, либо создать свою с нуля.