Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №7**

По дисциплине

«Основы профессиональной деятельности»

Вариант 1418

Выполнил:

Петров Вячеслав Маркович

Группа P3108

Принял:

Вербовой Александр Александрович

Санкт-Петербург 2024

Оглавление

[**Текст задания** 3](#_Toc165715757)

[**Исходный код синтезируемой команды** 3](#_Toc165715758)

[**Трассировка микропрограммы** 4](#_Toc165715759)

[**Код программы проверки команды на языке ассемблера** 4](#_Toc165715760)

[**Описание тестовых программ** 7](#_Toc165715761)

[**Подготовка к проверке** 8](#_Toc165715762)

[**Методика проверки тестов** 8](#_Toc165715763)

[**Вывод** 8](#_Toc165715764)

# **Текст задания**

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.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

# **Исходный код синтезируемой команды**

|  |  |  |
| --- | --- | --- |
| **Адрес ячейки** | **новый код МК** | **Комментарий** |
| 3D | 81E0104002 | if CR(12) = 1 then GOTO RESERVED E0 // Команда 9ХХХ теперь обрабатывается микрокомандой с адресом @E0 |
| **Цикл исполнения команды MADC(E0—E4)** | | |
| E0 | 80E2011040 | if PS(C) = 0 then GOTO E2 // Если флаг С не выставлен, то |
| E1 | 0001009401 | DR + 1 ? DR // Инкрементируем DR без установки флагов |
| E2 | 0001009011 | AC + DR ? DR // Сложение чисел и запись результата в регистр DR без установки флагов |
| E3 | 0200000000 | DR ? MEM(AR) // Запись результата в ячейку памяти |
| E4 | 80C4101040 | GOTO INT @ C4 // Завершение цикла выполнения команды, переход к циклу прерываний |

# **Трассировка микропрограммы**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MP до выборки МК | Содержимое памяти и регистров процессора после выборки и исполнения команды | | | | | | | | | |
| MR | IP | CR | AR | DR | SP | BR | AC | NZVC | MP (СчМК) |
| E0 | 80E2011040 | 011 | 9501 | 501 | 020D | 000 | 0010 | 0000 | 0100 | E2 |
| E2 | 0001009011 | 011 | 9501 | 501 | 020D | 000 | 0010 | 0000 | 0100 | E3 |
| E3 | 0200000000 | 011 | 9501 | 501 | 020D | 000 | 0010 | 0000 | 0100 | E4 |
| E4 | 80C4101040 | 011 | 9501 | 501 | 020D | 000 | 0010 | 0000 | 0100 | C4 |

# **Код программы проверки команды на языке ассемблера**

ORG 0x0

TT1: WORD 0x0; Тест 1 - Проверка корректного результата при С = 0

TT2: WORD 0x0; Тест 2 - Проверка на отсутствие изменения NZVC

TT3: WORD 0x0; Тест 3 - Проверка на корректный результат при С = 1 + проверка на сам флаг

TT4: WORD 0x0; Тест 4 – Аналогично 1ому тесту, только с другой адресацией

TT6: WORD 0x0; Тест 6 – Крайний случай

MEM: WORD 0x7C1

ORG 0x40F

START: CALL $TEST1; **Вызов** первого теста

LD $TT1; Загрузка результата 1ого теста в AC

NOP; Проверка

CALL $TEST2; **Вызов** второго теста

LD $TT2; Загрузка результата 2ого теста в АС

NOP ; Проверка

CALL $TEST3; **Вызов** третьего теста

LD $TT3; Загрузка результата 3его теста

NOP ; Проверка

CALL $TEST4; **Вызов** четвертого теста

LD $TT4; Загрузка результата 4его теста

NOP ; Проверка

CALL $TEST5; **Вызов** пятого теста

NOP ; Проверка

CALL $TEST6; **Вызов** шестого теста

LD $TT6; Загрузка результата 6ого теста

NOP ; Проверка

HLT

ORG 0x500

A1: WORD 0xA234 ; Первое слагаемое

B1: WORD 0x020D ; Второе слагаемое / результат сложения

RES1: WORD ?; Результат сложения командной ADD

**TEST1:** CLA

CLC

LD A1

ADC B1

ST RES1; Запись результата A1 + B1 -> RES1

LD A1

WORD 0x9501; Выполнение команды MADC

LD B1; Запись результата MADC в AС

CMP RES1 ; Проверка результатов

BNE ERR1

LD #0x1 ; Запись работы теста

ST $TT1 ; при корректной работе

RET

ERR1: LD #0x0 ; Запись работы теста

ST $TT1 ; при некорректной работе

RET

ORG 0x600

A2: WORD 0xA234; Первое слагаемое

B2: WORD 0xFFFF ; Второе слагаемое / результат сложения двух слагаемых

**TEST2**: CLA

CLC

LD A2

WORD 0x9601; Вызов MADC

BLO ERR2 ; Carry Flag выставляться не должен

LD #0x1 ; Запись работы теста

ST $TT2 ; при корректной работе

RET

ERR2: LD #0x0 ; Запись работы теста

ST $TT2 ; при некорректной работе

RET

ORG 0x700

A3: WORD 0xA234; Первое слагаемое

B3: WORD 0xFFFF; Второе слагаемое / результат сложения командой MADC с С = 1

RES3: WORD ?; Результат команды ADC для A3 и B3

**TEST3:** CLA

CLC

CMC

LD A3

ADC B3

ST $RES3; ADC + также выставляется сам CF

LD A3

WORD 0x9701; Выполнение MADC с выставленным флагом С

LD B3

BHIS ERR3; Сравнение С с 0, проверка на то, был ли сброшен флаг

CMP RES3; Сравнение с командой ADC,

BNE ERR3; результаты должны совпадать

LD #0x1; Запись работы теста

ST $TT3; при корректной работе

RET

ERR3: LD #0x0; Запись работы теста

ST $TT3; при некорректной работе

RET

ORG 0x750

A4: WORD 0xA234; Первое слагаемое

B4: WORD 0x0234; Второе слагаемое / результат сложения командой MADC

RES4: WORD ?; Результат команды ADC для A4 и B4

**TEST4:** CLA

CLC

LD A4

ADC B4

ST $RES4; ADC

LD A4

WORD 0x9EF6; Выполнение MADC с адресацией

LD B4

CMP RES4; Сравнение с командой ADC,

BNE ERR4; результаты должны совпадать

LD #0x1; Запись работы теста

ST $TT4; при корректной работе

RET

ERR4: LD #0x0; Запись работы теста

ST $TT4; при некорректной работе

RET

ORG 0x7A0

A5: WORD 0xA234; Первое слагаемое

RES5: WORD ?; Результат команды ADC для A4 и B4

**TEST5:** CLA

CLC

LD A4

ADC #0x33; то есть WORD 0x5F33

ST $RES5; ADC

LD A4

ORG 0x7C0

A6: WORD 0x0000; Первое слагаемое

B6: WORD 0xFFFF; Второе слагаемое / результат сложения командой MADC с С = 1

RES6: WORD ?; Результат команды ADC для A3 и B3

**TEST6:** CLA

CLC

CMC

LD A6

ADC (MEM)

ST $RES6; ADC + также выставляется сам CF

LD A3

WORD 0x9805; Выполнение MADC с выставленным флагом С

LD B6

CMP RES6; Сравнение с командой ADC,

BNE ERR6; результаты должны совпадать

LD #0x1; Запись работы теста

ST $TT6; при корректной работе

RET

ERR3: LD #0x0; Запись работы теста

ST $TT6; при некорректной работе

RET

# **Описание тестовых программ**

1. Первый тест проверяет команду MADC без CF: значение сравнивается с результатом команды ADD тех же чисел. Если тест работает правильно, то в переменную TT1 записывается 1, если тест неверный, то в ТТ1 записывается 0. После выполнения TEST1 результат ТТ1 также выводится в АС, перед выполнением следующего теста.
2. Второй тест проверяет отсутствие выставления знаков: при корректной работе в переменную TT2 записывается 1, при некорректной — 0. Аналогично первому тесту перед переходом к выполнению третьего теста значение TT2 выводится на АС.
3. Третий тест проверяет результат сложения ADD двух чисел + 1 с результатом MADC (с выставленным CF), они должны совпадать. Также флаг С не должен сбрасываться. Аналогично: корректный результат – 1 в TT3, некорректный – 0 в TT3. К концу программы результат теста выводится в АС.
4. Четвертый тест аналогичен первому, но использует относительную адресацию
5. В пятом тесте показывает, что будет, если мы напрямую загрузим значение второго слагаемого, то есть не будем иметь нужный нам AR
6. Шестой тест рассматривает крайний случай, когда складываются 0xFFFF и 0x0000, Carry Flag = 1

# **Подготовка к проверке**

1. Открыть БЭВМ в формате cli или dual “java –Dmode=dual –jar bcomp-ng.jar”
2. Открыть help “?”
3. На основе help и таблицы микрокоманд перенести нужные микрокоманды в БЭВМ
4. Открыть режим ввода Assembler “asm”
5. Загрузить команды Assembler в БЭВМ
6. Заменить везде NOP на HLT.
7. Написать после кода Assembler END и нажать Enter

# **Методика проверки тестов**

1. Запустить программу в режиме “РАБОТА” (адрес начала программы 0х40F).
2. Дождаться останова. Записать значение из АС в результат первого теста **ТТ1**.
3. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
4. Дождаться останова. Записать значение из АС в результат второго теста **ТТ2**.
5. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
6. Дождаться останова. Записать значение из АС в результат третьего теста **ТТ3**.
7. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
8. Дождаться останова. Записать значение из АС в результат третьего теста **ТТ4**.
9. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
10. Дождаться останова. Посмотреть, куда записался результат MADC.
11. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
12. Дождаться останова. Записать значение из АС в результат третьего теста **ТТ6**.
13. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
14. Дождаться останова.
15. Удостовериться, что все результаты тестов равны 0x1.

# **Вывод**

В ходе выполнения лабораторной работы я изучил алгоритм синтеза собственной команды БЭВМ с помощью микропрограмм и методику проверки сделанной программы