

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
«Синтез команд БЭВМ»
По дисциплине «Основы профессиональной деятельности»
Вариант: 8756

Выполнил:
Ясаков Артем Андреевич

Группа: Р3113

Преподаватель:
Ермаков Михаил Константинович

Санкт-Петербург 2025 г.

Оглавление

Задание	3
Исходный код синтезируемой команды	3
Трассировка микрокоманды	4
Тестовая программа	4
Описание тестовых программ.....	6
Подготовка к проверке.....	7
Методика проверки программы.....	7
Вывод.....	7

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.

Введите номер варианта

1. MADC M - сложение с учетом переноса аккумулятора с ячейкой памяти с записью результата в ячейку памяти и без установки N/Z/V/C
2. Код операции - 9...
3. Тестовая программа должна начинаться с адреса 00A4₁₆

Исходный код синтезируемой команды

Адрес ячейки	новый код МК	Комментарий
E0	81E1104002	if CR(12) = 1 then GOTO RESERVED E1 ; Команда 9XXX теперь обрабатывается микрокомандой с адресом @E1
Цикл исполнения команды MADC(E1—E5)		
E1	80E3011040	if PS(C) = 0 then GOTO E2 ; Если флаг C не выставлен, то перейти на E3
E2	0001009401	DR + 1 → DR ; Инкрементируем DR без установки флагов
E3	0001009011	AC + DR → DR ; Сложение чисел и запись результата в DR без установки флагов
E4	0200000000	DR → MEM(AR) ; Запись результата в ячейку памяти
E5	80C4101040	GOTO INT @C4 ; Завершение цикла выполнения команды, переход к циклу прерываний

Трассировка микрокоманды

MP до выборки МК	Содержимое памяти и регистров процессора после выборки и исполнения команды									
	MR	IP	CR	AR	DR	SP	BR	AC	NZVC	MP
E1	80E4011040	10A	9101	101	0135	7FE	0109	B345	1000	E4
E4	0001009011	10A	9101	101	B47A	7FE	0109	B345	1000	E5
E5	0200000000	10A	9101	101	B47A	7FE	0109	B345	1000	E6
E6	80C4101040	10A	9101	101	B47A	7FE	0109	B345	1000	C4

Тестовая программа

ORG 0x0

T1: WORD 0x0; Тест 1 - Проверка корректного результата при C = 0

T2: WORD 0x0; Тест 2 - Проверка на отсутствие изменения NZVC

T3: WORD 0x0; Тест 3 - Проверка на корректный результат при C=1

T4: WORD 0x0; Тест 4 - Крайний случай

ORG 0x0A4

```

START:  CALL $TEST1; Вызов тестов
        LD $T1
        NOP
        CALL $TEST2
        LD $T2
        NOP
        CALL $TEST3
        LD $T3
        NOP
        CALL $TEST4
        LD $T4
        HLT
    
```

ORG 0x100

```

A1:     WORD 0xB345
B1:     WORD 0x0135
RES1:   WORD ?
TEST1:  CLA
        CLC
    
```

LD A1
ADC B1
ST RES1

LD A1
WORD 0x9101; Выполнение команды MADC
LD B1
CMP RES1; Проверка результатов
BNE ERR1

LD #0x1
ST \$T1
RET

ERR1: LD #0x0
ST \$T1
RET

ORG 0x200

A2: WORD 0xB345
B2: WORD 0xFFFF
TEST2: CLA

CLC
LD A2
WORD 0x9201; Выполнение команды MADC
BHS ERR2; CF выставляться не должен

LD #0x1
ST \$T2
RET

ERR2: LD #0x0
ST \$T2
RET

ORG 0x300

A3: WORD 0xB345
B3: WORD 0xFFFF
RES3: WORD ?

TEST3: CLA
CLC
CMC
LD A3
ADC B3
ST \$RES3

LD A3
WORD 0x9301; Выполнение MADC с выставленным флагом C
LD B3

```
BLO ERR3
CMP RES3
BNE ERR3
```

```
LD #0x1
ST $T3
RET
```

```
ERR3:    LD #0x0
          ST $T3
          RET
```

```
ORG 0x400
```

```
A4:      WORD 0x0000
B4:      WORD 0xFFFF
RES4:    WORD ?
```

```
TEST4:   CLA
          CLC
          CMC
          LD A4
          ADC B4
          ST $RES4
```

```
LD A4
WORD 0x9401; Выполнение MADC с выставленным флагом C
LD B4
CMP RES4
BNE ERR4
```

```
LD #0x1
ST $T4
RET
```

```
ERR4:    LD #0x0
          ST $T4
          RET
```

Описание тестовых программ

1. Проверка команды MADC без CF.
2. Второй тест проверяет отсутствие выставления знаков NZVC.
3. Проверка результата сложения ADC двух чисел и CF=1 с результатом MADC (с выставленным CF), они должны совпадать. Также флаг C не должен сбрасываться.
4. Рассматриваем крайний случай, когда складываются 0xFFFF и 0x0000, CF = 1

Если какой-либо тест # работает правильно, то в переменную T# записывается 1, иначе записывается 0. После выполнения теста результат выводится в АС, перед выполнением следующего теста.

Подготовка к проверке

1. Открыть БЭВМ в формате cli или dual “java -Dmode=dual -jar bcomp-ng.jar”
2. Ввести микрокоманды через консоль

```
ma
mw 81E1104002
mw 80E3011040
mw 0001009401
mw 0001009011
mw 0200000000
mw 80C4101040
```
3. Открыть режим ввода Assembler “asm”
4. Загрузить команды Assembler в БЭВМ
5. Заменить везде NOP на HLT.
6. Написать после кода Assembler END и нажать Enter
7. Удостовериться что после прогона всех тестов в аккумуляторе лежит 0x1

Методика проверки программы

1. Запустить программу в режиме “РАБОТА” (адрес начала программы 0x0A4).
2. Дождаться останова. Записать значение из АС в результат первого теста **T1**.
3. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
4. Дождаться останова. Записать значение из АС в результат второго теста **T2**.
5. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
6. Дождаться останова. Записать значение из АС в результат третьего теста **T3**.
7. Нажать кнопку “ПРОДОЛЖЕНИЕ”.
8. Дождаться останова. Записать значение из АС в результат третьего теста **T4**.
9. Удостовериться, что все результаты тестов равны 0x1.

Вывод

В ходе данной лабораторной работы я познакомился с МПУ БЭВМ и синтезировал свою команду и изучил методику проверки сделанной программы