

Федеральное государственное автономное образовательное учреждение
высшего образования “Национальный исследовательский университет ИТМО”

Факультет Программной Инженерии И Компьютерной Техники

Лабораторная работа №7

Вариант 507

Выполнила:

Абдуллаева София Улугбековна

Группа Р3108

Проверил:

Вербовой Александр Александрович

Санкт-Петербург 2025

Оглавление

Задание	3
Исходный код синтезируемой команды	3
Таблица трассировки микропрограммы	3
Тестовая программа	4
Описание тестовых программ.....	6
Подготовка к проверке и 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. ADCSP – Сложить два верхних числа на вершине стека с учетом переноса, результат поместить на стек, установить признаки N/Z/V/C
2. Код операции – 0F10
3. Тестовая программа должна начинаться с адреса 00E6₁₆

Исходный код синтезируемой команды

Адрес ячейки	Микрокоманда	Комментарий
E0	0080009008	SP -> AR ; поместить вершину стека в AR
E1	0100000000	MEM(AR) -> DR ; ячейка памяти попадает в DR
E2	0020009001	DR -> BR ; первое значение стека поместить в BR
E3	0080009408	SP + 1 -> AR ; второе значение стека поместить в AR
E4	0100000000	MEM(AR) -> DR ; ячейка памяти попадает в DR
E5	80E8011040	if PS(C) = 0 then GOTO E8 ; если перенос не выставлен, то переход на E8
E6	0001E09421	BR + DR + C -> DR , N, Z, V, C ; сложение первого, второго значений стека и C и установка флагов
E7	80E9101040	GOTO E9 ; переход на E9
E8	0001E09021	BR + DR -> DR , N, Z, V, C ; сложение первого и второго значений стека и установка флагов
E9	0088009208	~0 + SP -> SP, AR ; уменьшить стек на 1 и записать в SP и AR
EA	0200000000	DR -> MEM(AR) ; результат операции на стек
EB	80C4101040	GOTO INT @ C4 ; переход к циклу прерывания

Таблица трассировки микропрограммы

Возьмём значения двух чисел из второго теста, в 7FE находится FDDD и в 7FD - 4022

MP до выборки МК	Содержимое памяти и регистров процессора после выборки микрокоманды									
	MR	IP	CR	AR	DR	SP	BR	AC	NZVC	СчМК
E0	0080009008	0E7	0F10	7FE	0000	7FE	0E6	FDDD	1001	E1
E1	0100000000	0E7	0F10	7FE	FDDD	7FE	0E6	FDDD	1001	E2

E2	0020009001	0E7	0F10	7FE	FDDD	7FE	FDDD	FDDD	1001	E3
E3	0080009408	0E7	0F10	7FD	FDDD	7FD	FDDD	4022	0001	E4
E4	0100000000	0E7	0F10	7FD	4022	7FD	FDDD	4022	0001	E5
E5	80E8011040	0E7	0F10	7FD	4022	7FD	FDDD	4022	0001	E6
E6	0001E09421	0E7	0F10	7FD	3E00	7FD	FDDD	4022	0001	E7
E7	80E9101040	0E7	0F10	7FD	3E00	7FD	FDDD	4022	0001	E9
E9	0088009208	0E7	0F10	7FC	3E00	7FC	FDDD	4022	0001	EA
EA	0200000000	0E7	0F10	7FC	3E00	7FC	FDDD	4022	0001	EB
EB	80C4101040	0E7	0F10	7FC	3E00	7FC	FDDD	4022	0001	C4

Тестовая программа

```

ORG 0x0DA
T1: WORD 0x0 ; тест 1 без выставления флага C
T2: WORD 0x0 ; тест 2, в котором V=0 и C=1
T3: WORD 0x0 ; тест 3, в котором V=1 и C=1
T4: WORD 0x0 ; тест 4, сравниваем сложение 2 чисел без флага C и INC со
сложением 2 чисел, где C=1

ORG 0x0E6
START: CALL $TEST1 ; вызываем 1 тест
      LD $T1
      HLT
      CALL $TEST2 ; вызываем 2 тест
      LD $T2
      HLT
      CALL $TEST3 ; вызываем 3 тест
      LD $T3
      HLT
      CALL $TEST4 ; вызываем 4 тест
      LD $T4
      HLT

ORG 0x100
X1: WORD 0x0112
Y1: WORD 0x0014
RES1: WORD 0x0126
TEST1: LD X1
      PUSH
      LD Y1
      PUSH
      WORD 0x0F10 ; команда ADCSP
      HLT
      POP ; снять со стека результат команды
      CMP RES1 ; сравнить его с RES1
      BNE ERROR1

```

```

        POP            ; снять со стека 2 число
        POP            ; снять со стека 1 число
        LD #0x01
        ST $T1         ; записать 1 при корректном результате
        RET
ERROR1: POP            ; снять со стека 2 число
        POP            ; снять со стека 1 число
        LD #0x00
        ST $T1         ; записать 0 при некорректном результате
        RET

ORG 0x150
X2: WORD 0xFDDD
Y2: WORD 0x4022
RES2: WORD 0x3E00
TEST2: CLC            ; C=0
        CMC            ; ^C=1
        LD X2
        PUSH
        LD Y2
        PUSH
        WORD 0x0F10 ; команда ADCSP
        HLT
        POP            ; снять со стека результат команды
        CMP RES2       ; сравнить его с RES2
        BNE ERROR2
        POP            ; снять со стека 2 число
        POP            ; снять со стека 1 число
        LD #0x01
        ST $T2         ; записать 1 при корректном результате
        RET
ERROR2: POP            ; снять со стека 2 число
        POP            ; снять со стека 1 число
        LD #0x00
        ST $T2         ; записать 0 при некорректном результате
        RET

ORG 0x200
X3: WORD 0xC087
Y3: WORD 0xA074
RES3: WORD 0x60FC
TEST3: CLC            ; C=0
        CMC            ; ^C=1
        LD X3
        PUSH
        LD Y3
        PUSH
        WORD 0x0F10 ; команда ADCSP
        HLT
        POP            ; снять со стека результат команды
        CMP RES3       ; сравнить его с RES3
        BNE ERROR3
        POP            ; снять со стека 2 число
        POP            ; снять со стека 1 число
        LD #0x01
        ST $T3         ; записать 1 при корректном результате
        RET
ERROR3: POP            ; снять со стека 2 число
        POP            ; снять со стека 1 число
        LD #0x00
        ST $T3         ; записать 0 при некорректном результате

```

```

RET
ORG 0x250
X4: WORD 0x3013
Y4: WORD 0x3F7A
RES4: WORD 0x0
TEST4: CLC          ; C=0
        LD X4
        PUSH
        LD Y4
        PUSH
        WORD 0x0F10 ; команда ADCSP
        HLT
        POP          ; снять со стека результат сложения 2 чисел
        INC          ; инкрементирование результата на 1
        ST $RES4     ; положить в RES4, чтобы потом сравнить со 2 результатом
        POP          ; снять со стека 2 число
        POP          ; снять со стека 1 число
        CLC          ; C=0
        CMC          ; ^C=1
        LD X4
        PUSH
        LD Y4
        PUSH
        WORD 0x0F10 ; команда ADCSP
        HLT
        POP          ; снять со стека результат команды
        CMP RES4     ; сравнить его с RES4
        BNE ERROR4
        POP          ; снять со стека 2 число
        POP          ; снять со стека 1 число
        LD #0x01
        ST $T4       ; записать 1 при корректном результате
        RET
ERROR4: POP          ; снять со стека 2 число
        POP          ; снять со стека 1 число
        LD #0x00
        ST $T4       ; записать 0 при некорректном результате
        RET

```

Описание тестовых программ

1. Первый тест проверяет выполнение операции сложения двух верхних чисел на вершине стека при $C=0$
2. Второй и третий тесты проверяют выполнение сложения двух верхних чисел на вершине стека при $C=1$, при этом во 2 тесте $V=0$ и в 3 тесте $V=1$ ($V = C_{14} + C_{\text{new}}$ на выходе коммутатора)
3. В четвёртом тесте сбрасывается флаг C в 0, сначала вычисляется сложение двух верхних чисел на вершине стека, снимаем результат со стека и используем команду `INC`, результат записывается в `RES4`. Далее устанавливается флаг $C=1$, снова выполняется команда `ADCSP`, сравниваем второй результат с `RES4`, они должны совпадать

Подготовка к проверке и методика проверки программы

1. Запустить БЭВМ в режиме Dual с помощью команды java - Dmode=dual -jar bcomp-ng.jar
2. Ввести последовательность команд в терминале:
ma
mw 0080009008
mw 0100000000
mw 0020009001
mw 0080009408
mw 0100000000
mw 80E8011040
mw 0001E09421
mw 80E9101040
mw 0001E09021
mw 0088009208
mw 0200000000
mw 80C4101040
mdecodeall
3. Загрузить тестовую программу в БЭВМ
4. Запустить БЭВМ в режиме “Работа”, нажать на “Продолжение”, дождаться останова
5. После каждого теста в АС будет 1, тогда тесты выполнены корректно

Вывод

Во время выполнения лабораторной работы я изучила, как синтезировать собственную команду в БЭВМ с помощью микропрограмм. Также научилась делать трассировку микропрограммы и тестировать разработанные микропрограммы.