МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №7**

по дисциплине

«Основы профессиональной деятельности»

Вариант № 3100

***Выполнил:***

Студент группы P3131

Дворкин Борис Александрович

***Преподаватель:***

Клименков Сергей

Викторович

Санкт-Петербург, 2023 г.

Содержание

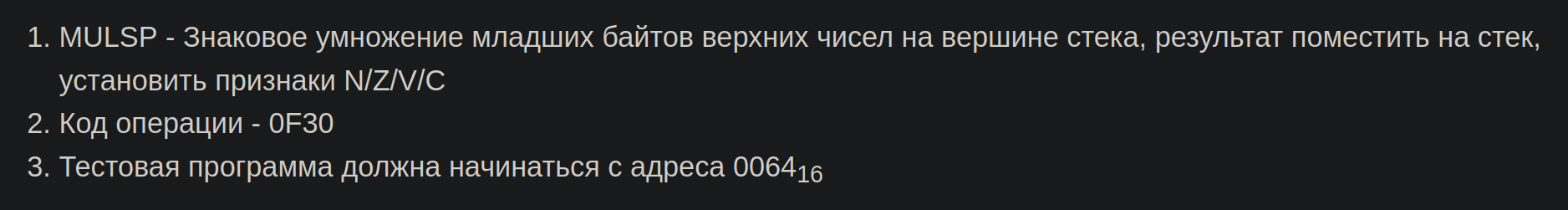
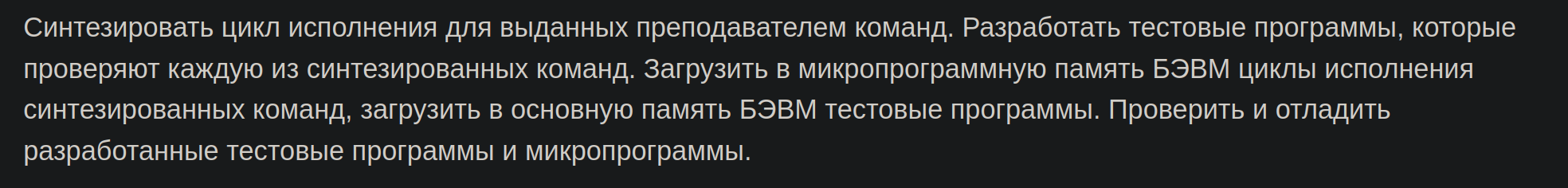
[Текст задания 3](#_Toc127641422)

[Описание программы 3](#_Toc127641423)

[Вывод 6](#_Toc127641424)

[Таблица трассировки 7](#_Toc127641425)

Текст задания



Описание программы

**Назначение программы**

1. MULSP - Знаковое умножение младших байтов двух верхних чисел на вершине стека, результат поместить на стек, установить признаки N/Z/V/C
2. Код операции - 0F30
3. Тестовая программа должна начинаться с адреса 006416

**Текст микропрограммы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес МП** | **Микрокоманда** | **Описание** | **Комментарий** |
| BB | 81F0304002 | if CR(8) = 1 then GOTO @ F0 | Исправленная ячейка интерпретатора |
| F0 | 0080009008 | SP -> AR | Первое значение стека |
| F1 | 0100000000 | MEM(AR) -> DR | в DR |
| F2 | 0020009001 | DR -> BR | Первое значение в BR |
| F3 | 0080009408 | SP + 1 -> AR | Второе значение стека |
| F4 | 0100000000 | MEM(AR) -> DR | в DR |
| F5 | 0000800002 | BR & 0x00FF -> BR | Сохранить младший байт первого числа |
| F6 | 0001800002 | DR & 0x00FF -> DR | Сохранить младший байт второго числа |
| F7 | 0004009001 | DR -> IP | Первое число в IP для удобства умножения |
| F8 | 0000000000 | No operations | Сброс CR |
| F9 | 0008009001 | DR -> SP | Инициализировать результат умножения в SP |
| FA | 0002001002 | LTOL(CR) → CR | Если второе число равно 0, завершить |
| FB | 0008009401 | DR + 1 -> SP | Добавить первое число к результату |
| FC | 0001808008 | HTOH(SP) → DR, N, Z | Уменьшить второе число на 1 |
| FD | 0002001002 | LTOL(CR) → CR | Если второе число не равно 0, продолжить умножение |
| FE | 0008009001 | DR -> SP | Перенести результат умножения в SP |
| FF | 000180982F | DR -> DR; N, Z, V, C | Установить флаги N, Z, V, C |
| 100 | 0088009208 | ~0 + SP -> SP, AR | Результат выражения |
| 101 | 0200000000 | DR -> MEM(AR) | Результат умножения в стек |
| 102 | 80C4101040 | GOTO INT @ C4 | Переход к циклу прерывания |

**Текст программы**

|  |
| --- |
| ORG 0x058  ARG1: WORD 0x0000  ARG2: WORD 0x0003  ARG3: WORD 0x0004  ARG4: WORD 0xAAAA  ARG5: WORD 0x8342  CHECK1: WORD 0xFFFF  CHECK2: WORD 0xFFFF  CHECK3: WORD 0xFFFF  FINAL: WORD 0xFFFF  RES1: WORD 0x0000  RES2: WORD 0x000С  RES3: WORD 0x0006  START: CLA  CALL TEST1  HLT  CALL TEST2  HLT  CALL TEST3  HLT  LD #0x1  CLC  AND CHECK1  AND CHECK2  AND CHECK3  ST FINAL  STOP: HLT    TEST1: LD ARG1  PUSH  LD ARG2  PUSH  WORD 0x0F30  HLT  POP  CMP RES1  BEQ DONE1  ST RES1  POP  POP  CLA  ST CHECK1  RET  DONE1: ST RES1  POP  POP  LD #0x1  ST CHECK1  CLA  RET    TEST2: LD ARG3  PUSH  LD ARG4  PUSH  WORD 0x0F30  HLT  POP  CMP RES2  BEQ DONE2  ST RES2  POP  POP  CLA  ST CHECK2  RET  DONE2: ST RES2  POP  POP  LD #0x1  ST CHECK2  CLA  RET    TEST3: LD ARG5  PUSH  LD ARG5  PUSH  WORD 0x0F30  HLT  POP  CMP RES3  BEQ DONE3  ST RES3  POP  POP  CLA  ST CHECK3  RET  DONE3: ST RES3  POP  POP  LD #0x1  ST CHECK3  CLA  RET |

Таблица трассировки микропрограммы

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МР до выборки МК | Содержимое памяти и регистров процессора после выборки микрокоманды | | | | | | | | | |
| **MR** | **IP** | **AR** | **AR** | **DR** | **SP** | **BR** | **AC** | **NZVC** | **СчМК** |
| E0 | 0080009008 |  |  |  |  |  |  |  |  | E1 |
| E1 | 0100000000 |  |  |  |  |  |  |  |  | E2 |
| E2 | 0020009001 |  |  |  |  |  |  |  |  | E3 |
| E3 | 0080009408 |  |  |  |  |  |  |  |  | E4 |
| E4 | 0100000000 |  |  |  |  |  |  |  |  | E5 |
| E5 | 0003809821 |  |  |  |  |  |  |  |  | E6 |
| E6 | 0088009208 |  |  |  |  |  |  |  |  | E7 |
| E7 | 0200000000 |  |  |  |  |  |  |  |  | E8 |
| E8 | 80C4101040 |  |  |  |  |  |  |  |  | C4 |

Методика проверки

1. Загрузить тестовую программу в память базовой ЭВМ.
2. Запустить основную программу с адреса 0x058 в режиме работа.
3. Дождаться останова.
4. Проверить значение ячейки памяти FINAL с номером 0x060, если значение 0x0001 – все тесты выполнены успешно.

Комментарии к методике

* Для проверки используется три пары значений: 0000 & 0000, 0003 & 0004, AAAA & 8342
* Данные значения показывают правильную работу программы с отрицательными, нулевыми и положительными числами.
* В ходе проверки флаги N, Z, V, C меняются с 0 на 1 и с 1 на 0 в двух разных случаях, что говорит о правильном выставлении флагов.
* Результат каждого теста записывается в соответствующую ячейку RES, значение 0x0001 означает успешное выполнение. Любое другое – результат выполнения синтезированной команды
* При успешном выполнении всех тестов значение FINAL станет 0x0001, иначе 0x0000

Сопоставление полученного и теоретического результата

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ячейка с результатом** | | **Первое число** | **Второе число** | **Теоретический результат** | **Полученный результат** |
| RES1 | 0x401 | 0000 | 0000 | 0000 (N=0, Z=1, V=0, C=0) |  |
| RES2 | 0x402 | 0003 | 0004 | 000C (N=0, Z=0, V=0, C=0) |  |
| RES3 | 0x403 | AAAA | 8342 | (N=0, Z=0) |  |

1010 1010 1010 1010 = - 0101 0101 0101 0110 = -21846

1000 0011 0100 0010 = - 0111 1100 1011 1110 = -31934

# Вывод

В ходе выполнения лабораторной работы я изучил обмен данными с ВУ-1 и ВУ-3 в режиме прерываний, также изучил цикл прерывания и циклы исполнения новых команд. Также закрепил знания в написании программ на ассемблере БЭВМ.

# Методика проверки программы

Проверка обработки прерываний:

1. Загрузить текст программы в БЭВМ.

2. Заменить NOP по нужному адресу на HLT.

3. Запустить программу в режиме РАБОТА.

4. Установить «Готовность ВУ-1».

5. Дождаться останова.

6. Записать текущее значение X из памяти БЭВМ:

1. Запомнить текущее состояние счетчика команд.

2. Ввести в клавишный регистр значение 0x010

3. Нажать «Ввод адреса».

4. Нажать «Чтение».

5. Записать значение регистра данных.

6. Вернуть счетчик команд в исходное состояние.

7. Нажать «Продолжение» 2 раза.

8. Записать результат обработки прерывания – полученное значение F(x) из DR контроллера ВУ-1.

9. Нажать «Продолжение».

10. Ввести в ВУ-3 произвольное число, записать его.

11. Установить «Готовность ВУ-3».

12. Дождаться останова.

13. Записать текущее значение X из памяти БЭВМ:

1. Запомнить текущее состояние счетчика команд.

2. Ввести в клавишный регистр значение 0x010

3. Нажать «Ввод адреса».

4. Нажать «Чтение».

5. Записать значение регистра данных.

6. Вернуть счетчик команд в исходное состояние.

14. Нажать «Продолжение» 2 раза.

15. Записать результат обработки прерывания – DR + X из DR контроллера ВУ-3.

16. Рассчитать ожидаемое значение переменной X после обработки прерывания и сравнить.

Проверка основной программы:

1. Загрузить текст программы в БЭВМ.

2. Записать в переменную X максимальное по ОДЗ значение (22)

3. Запустить программу в режиме останова.

4. Пройти нужное количество шагов программы, убедиться, что при уменьшении X на 1, до после момента, когда он равен 2-21, происходит сброс значения в минимальное по ОДЗ.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Прерывание ВУ-1 | | | Прерывание ВУ-3 | | | | |
| AC  (0...7) | Ожидание  -6\*X+3 | DR | AC (0…7) | DR  КВУ-3 | AC  (DR + X) | Результат AC (0...7) |
| 1016 (16) | FFA316 (-93) | FFA316 (-93) | 116 (1) | (127) | 5D16 (126) | 1616 (22) |
| FF16 (-1) | FD16 (-3) | FD16 (-3) | 116 (1) |  | 0016 (0) | 016 (0) |
| 1816 (25) | 7F16 (127) | 7F16 (127) | 116 (1) | (-31) | E216 (-32) | EВ16 (-21) |