Санкт-Петербургский Национальный Исследовательский  
Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №7**

По “Основы профессиональной деятельности”

Вариант 1781

*Выполнил*:

Студент группы P3117

Васильченко Роман Антонович

*Преподаватель:*

Ткешелашвили Нино Мерабиевна



Санкт-Петербург

2022

Оглавление

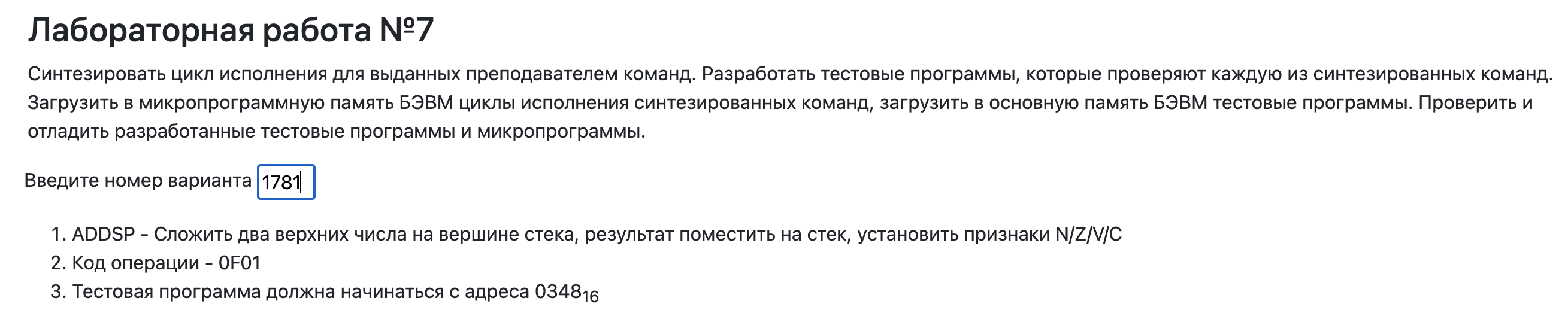
[Задание 2](#_Toc1)

[Программа 2](#_Toc2)

[Описание программы 4](#_Toc3)

[Методика проверки 5](#_Toc4)

## Задание



**Исходный код синтезируемой команды**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес МП | Микрокоманда | Описание | Комментарий |
| BB | 81F0014002 | 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 | 0001F09021 | BR + DR → DR, N, Z, V, C | Сумма двух верхних чисел, знаки N, Z, V, C |
| F6 | 0088009208 | ~0 + SP -> SP, AR | Результат выражения |
| F7 | 0200000000 | DR -> MEM(AR) | в стек |
| F8 | 80C4101040 | GOTO INT @ C4 | Переход к циклу прер. |

### Описание команды

1. ADDSP - Сложить два верхних числа на вершине стека, результат поместить на стек, установить признаки N/Z/V/C
2. Код операции - 0F01
3. Тестовая программа должна начинаться с адреса 034816

## Программа

|  |  |  |
| --- | --- | --- |
| ORG | 0x0348 |  |
| ARG1: | WORD | 0x0084 |
| ARG2: | WORD | 0x0044 |
| ARG3: | WORD | 0x6D60 |
| ARG4: | WORD | 0x12AF |
| ARG5: | WORD | 0x8000 |
| ARG6: | WORD | 0x8000 |
| CHECK1: | WORD | 0x0 |
| CHECK2: | WORD | 0x0 |
| CHECK3: | WORD | 0x0 |
| FINAL: | WORD | 0x0 |
| RES1: | WORD | 0x0 |
| RES2: | WORD | 0x0 |
| RES3: | WORD | 0x0 |
| START: | CLA |  |
| CALL | TEST1 |  |
| HLT |  |  |
| CLC |  |  |
| CALL | TEST2 |  |
| HLT |  |  |
| CLC |  |  |
| CALL | TEST3 |  |
| HLT |  |  |
| CLC |  |  |
| LD | #0x1 |  |
| CLC |  |  |
| AND | CHECK1 |  |
| AND | CHECK2 |  |
| AND | CHECK3 |  |
| ST | FINAL |  |
| STOP: | HLT |  |
| TEST1: | LD | ARG1 |
| PUSH |  |  |
| ST | RES1 |  |
| LD | ARG2 |  |
| PUSH |  |  |
| ADD | RES1 |  |
| ST | RES1 |  |
| CMP1: | WORD | 0x0F01 |
| 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 |  |  |
| ST | RES2 |  |
| LD | ARG4 |  |
| PUSH |  |  |
| ADD | RES2 |  |
| ST | RES2 |  |
| CMP2: | WORD | 0x0F01 |
| POP |  |  |
| CMP | RES2 |  |
| BEQ | DONE2 |  |
| ST | RES2 |  |
| POP |  |  |
| POP |  |  |
| CLA |  |  |
| ST | CHECK2 |  |
| RET |  |  |
| DONE2: | ST | RES2 |
| POP |  |  |
| POP |  |  |
| LD | #0x1 |  |
| ST | CHECK2 |  |
| CLA |  |  |
| RET |  |  |
| TEST1: | LD | ARG5 |
| PUSH |  |  |
| ST | RES3 |  |
| LD | ARG6 |  |
| PUSH |  |  |
| ADD | RES3 |  |
| ST | RES3 |  |
| CMP0: | WORD | 0x0F01 |
| POP |  |  |
| CMP | RES3 |  |
| BEQ | DONE3 |  |
| ST | RES3 |  |
| POP |  |  |
| POP |  |  |
| CLA |  |  |
| ST | CHECK3 |  |
| RET |  |  |
| DONE1: | ST | RES3 |
| POP |  |  |
| POP |  |  |
| LD | #0x1 |  |
| ST | CHECK3 |  |
| CLA |  |  |
| RET |  |  |

### Описание программы

1. Программа тестируем синтезированную команду на трех разных тестах и записывает +1, если результат сходится и +0, если нет.

**Область допустимых значений**

Значение ARG(s) должны находиться в промежутке (-1)

Однако для специфичных случаев (Тестирование NZVC флагов) допускается

\*

* Стоит отметить, что в данном случае требуется понимать, что числа могут перепрыгивать из + в – и обратно, так как появляются C и V флаги.

### Методика проверки (Подготовка к проверке)

1. Скачать код с Github:   
     
   $ wget [https://raw.githubusercontent.com/RomanVassilchenko/ITMOProjects/main/ОПД/1%20курс%20%7C%202%20семестр/ЛабораторныеРаботы/lab7/main.asm](https://raw.githubusercontent.com/RomanVassilchenko/ITMOProjects/main/ОПД/1 курс %7C 2 семестр/ЛабораторныеРаботы/lab6/main.asm)
2. Открыть БЭВМ в формате cli или dual “java –Dmode=dual –jar bcomp-ng.jar”
3. Открыть help “?”
4. На основе help и таблицы микрокоманд перенести нужные микрокоманды в БЭВМ
5. Открыть режим ввода Assembler “asm”
6. Загрузить команды Assembler в БЭВМ
7. Написать после кода Assembler END и нажать Enter

### **Методика проверки синтезированной программы:**

1. Нажать на Run (F9)
2. Запустить основную программу с адреса 0x0348 (F7)
3. Дождаться остановы.
4. Удостовериться, что значение RES1 (35216) = (ARG1 + ARG2) = 00C816 и CHECK1 (34E16) = 000116
5. Продолжить.
6. Дождаться остановы.
7. Удостовериться, что значение RES2 (35316) = (ARG3 + ARG4) = 800F16 и CHECK2 (34F16) = 000116
8. Продолжить.
9. Дождаться остановы.
10. Удостовериться, что значение RES3 (35416) = (ARG5 + ARG6) = 000016 и CHECK3 (35016) = 000116
11. Продолжить.
12. Удостовериться, что значение FINAL (35116) = 000116

### Сопоставление полученного и теоретического результата

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ячейка с результатом** | | **Первое число** | **Второе число** | **Теоретический результат** | **Полученный результат** |
| RES1 | 0x346 | 0084 | 0044 | 00C8 (N=0, Z=0, V=0, C=0) | 00C8 (N=0, Z=0, V=0, C=0) |
| RES2 | 0x347 | 6D60 | 12AF | 800F (N=1, Z=0, V=1, C=0) | 800F (N=1, Z=1, V=1, C=0) |
| RES3 | 0x348 | 8000 (N=1) | 8000 | 0000 (N=0, Z=1, V=0, C=1) | 0000 (N=0, Z=1, V=0, C=1) |

### Вывод

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