Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский

**университет информационных технологий, механики и оптики»**

**Факультет ПИИКТ**

**Дисциплина: Основы Профессиональной Деятельности**

**Лабораторная работа №7**

**«Основы БЭВМ»**

**Вариант 12999.**

Выполнил: Братчиков Иван

Группа: Р3101

2019 г.

**1.Описание программы:**

1. **ASL M - сдвиг ячейки памяти влево, 0 разряд заполняется значением 0. Признаки N/Z не устанавливать**
2. **Код операции - 7...**
3. **Тестовая программа должна начинаться с адреса 016B16**

**2.Текст исходной программы:**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес МП | Микро-команды | Действие | Комментарии |
| B0 | 0100 | РД → БР | Из РД помещаем содержимое памяти в А |
| B1 | 4005 | БР → A |
| B2 | 0008 | RAL(A) → БР | Сдвиг влево А и установка C |
| B3 | 4045 | БР → A, C |
| B4 | B0B7 | IF BIT(0,AK) = 0 THEN GOTO (B7) | Если нулевой бит А равен 0, то переходим к записи из A в БР, иначе заполняем 0 разряд А нулем. |
| B5 | 1080 | A + COM(0) = A - 1 → БР | Декремент аккумулятора |
| B6 | B0B8 | IF BIT(0,AK) = 0 THEN GOTO (B8) | Переход к записи в РД из БР, если нулевой бит теперь равен 0 |
| B7 | 1000 | A → БР | Записываем результат сдвига в ячейку памяти |
| B8 | 4002 | БР → РД |
| B9 | 0002 | РД → ОП(РА) |
| BA | 8390 | GOTO ПРЕ(90) | Окончание команды |

**3.Текст тестовой программы на языке Ассемблера БЭВМ:**

|  |  |  |
| --- | --- | --- |
| ORG | 000 |  |
| X: | WORD 0022 | ; Число 2216 |
| TEMP\_LAST:  TEMP\_NEGATIVE:  TEMP\_ZERO:  RESULT\_LAST:  RESULT\_NEGATIVE:  RESULT\_ZERO: | WORD 0000  WORD FFFF  WORD 8000  WORD 0000  WORD 0000  WORD 0000 | ; Проверка на 0 бит и рабочий ROL  ; Отрицательное число для проверки флагаN  ; Проверка на отсутствие флаг Z  ; Результат на 0 бит и корректность программы  ; Результат на отсутствие флага N  ; Результат на отсутствия флага Z |
| OVERALL\_SCORE:  MASK: | WORD 0000  WORD FFFE | ; Равно 1, если команда корректна  ; Маска для проверки последнего  ;бита ячейки памяти на 0 |
| ORG  BEGIN: | 16B  CLA  ADD X  MOV TEMP\_LAST  BR CHECK\_LAST | ; Сохранили значение ячейки X  ; Переходим на первый Тест |
| CHECK\_LAST: | CLA | ; Очищаем А и С  ; Инвертируем бит C (т.е. C = 1), чтобы проверить обнуление 0 бита при сдвиге  ; Исполняем команду ASL M  ; Очищаем А и С  ; Добавляет в A сохраненное оригинальное значение X  ; Сдвигаем значение X влево  ; Очищаем бит C  ; Побитово умножаем маску и А  ; Вычитаем получившиеся число от результата ASM M команды  ; Если разница == 0, то переходим в RETURN\_LAST |
|  | CLC  CMC  WORD 7000  CLA  CLC  ADD TEMP\_LAST  ROL  CLC  AND MASK  SUB X  BEQ RETURN\_LAST |
| RETURN\_LAST: | BR OVERALL  CLA  CLC | ; Иначе подводим итог всех тестов  ; Очищаем аккумулятор и устанавливаем метку в единицу (тест прошел успешно)  ; Переходим в следующий тест  ; Очищаем А и С  ; Выполняем команду ASL M над Отрицательным числом  ; Если результат больше 0 то переход в RETURN\_NEGATIVE  ; Иначе подводим итог тестов  ; Очищаем аккумулятор и устанавливаем метку в единицу (тест прошел успешно)  ; Завершение всех тестов, переход к итогу  ; Очищаем А и С  ; Логически умножаем результаты всех тестов  ; Записываем результат в ячейку Result  ; Переход к завершению программы |
|  | INC |
| CHECK\_NEGATIVE:  RETURN\_NEGATIVE:  OVERALL: | MOV RESULT\_LAST  BR CHECK\_NEGATIVE  CLA  CLC  WORD 7002  BPL RETURN\_NEGATIVE  BR OVERALL  CLA  CLC  INC  MOV RESULT\_NEGATIVE  BR OVERALL  CLA  CLC  ADD RESULT\_NEGATIVE  AND RESULT\_LAST  MOV OVERALL\_SCORE  BR END |
| END: | HLT | ; Остановка ЭВМ |

**4.Таблица трассировки цикла исполнения разработанной команды**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Исходные данные: команда 16F ASL M, код команды 7000** | | | | | | | | | | | |
| **СчМК до выборки МК** | **Содержимое памяти и регистров процессора перед циклом исполнения команды** | | | | | | | | | | |
| **РМК** | **СК** | **РА** | **РК** | **РД** | **А** | **С** | **БР** | **N** | **Z** | **СчМК** |
|  | AB1D | 170 | 16F | 7000 | 7000 | 0000 | 0 | 07000 | 0 | 1 | 1D |
| **Исполнение команды** | | | | | | | | | | | |
| 1D | EF2D | 170 | 16F | 7000 | 7000 | 0000 | 0 | 07000 | 0 | 1 | 1E |
| 1E | 0100 | 170 | 16F | 7000 | 7000 | 0000 | 0 | 07000 | 0 | 1 | 1F |
| 1F | 4001 | 170 | 000 | 7000 | 7000 | 0000 | 0 | 07000 | 0 | 1 | 20 |
| 20 | EE27 | 170 | 000 | 7000 | 7000 | 0000 | 0 | 07000 | 0 | 1 | 27 |
| 27 | 0001 | 170 | 000 | 7000 | 0022 | 0000 | 0 | 00000 | 0 | 1 | 28 |
| 28 | AD2B | 170 | 000 | 7000 | 0022 | 0000 | 0 | 07000 | 0 | 1 | 29 |
| 29 | AC43 | 170 | 000 | 7000 | 0022 | 0000 | 0 | 07000 | 0 | 1 | 2A |
| 2A | 83B0 | 170 | 000 | 7000 | 0022 | 0000 | 0 | 00002 | 0 | 1 | B0 |
| B0 | 0100 | 170 | 000 | 7000 | 0022 | 0000 | 0 | 00022 | 0 | 1 | B1 |
| B1 | 4005 | 170 | 000 | 7000 | 0022 | 0022 | 0 | 00022 | 0 | 1 | B2 |
| B2 | 0008 | 170 | 000 | 7000 | 0022 | 0022 | 0 | 00044 | 0 | 1 | B3 |
| B3 | 4045 | 170 | 000 | 7000 | 0022 | 0044 | 0 | 00044 | 0 | 1 | B4 |
| B4 | B0B7 | 170 | 000 | 7000 | 0022 | 0044 | 0 | 00044 | 0 | 1 | B7 |
| B7 | 1000 | 170 | 000 | 7000 | 0022 | 0044 | 0 | 00044 | 0 | 1 | B8 |
| B8 | 4002 | 170 | 000 | 7000 | 0044 | 0044 | 0 | 00044 | 0 | 1 | B9 |
| B9 | 0002 | 170 | 000 | 7000 | 0044 | 0044 | 0 | 00000 | 0 | 1 | BA |
| BA | 8390 | 170 | 000 | 7000 | 0044 | 0044 | 0 | 00002 | 0 | 1 | 90 |

**Вывод:** В ходе выполнения лабораторной работы я столкнулся с рядом проблем, к примеру, мне надо было проверить правильность своей команды, это я решил с помощью команд SUB и BEQ и BPL, также были трудности с написанием микрокоманды БР->А, получив вертикальную команду, пришлось разбираться, как получить из нее вертикальную, что я смог сделать с помощью Синей Библии.