Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №6

по курсу «Организация ЭВМ и систем»

**Вариант 9**

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Клюкин В.Л./

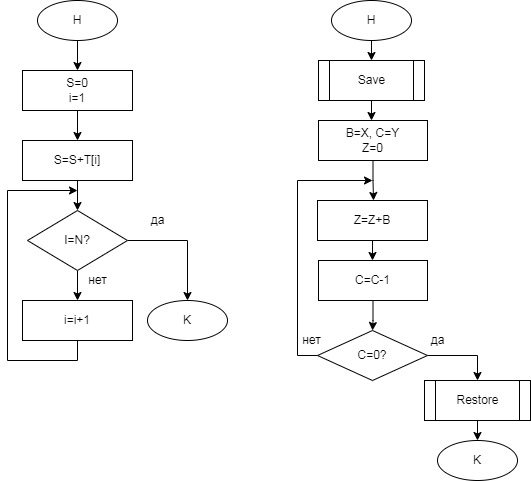
Киров 2022

1. Задание

Выполнить умножение положительных чисел с фиксированной запятой методом "младшими разрядами вперед". Произведение представлено 16-разрядным, а множители 8-разрядными двоичными числами

2. Определение архитектуры и программирование

2.1. Схема алгоритма



2.2. Форматы данных

По заданию множители представлены 8-разрядными числами, произведение – 16-разрядными числами.

2.3. Программно-доступные регистры

|  |  |  |
| --- | --- | --- |
| r0 | X | Множитель 1 |
| r1 | Y | Множитель 2 |
| r2 | Z | Произведение |
| r3 | S | Сумма |
| r4 | N | Число повторений |
| r5 | AM | Указатель массива |
| r6 | PC | Программный счетчик |
| r7 | RP (PZ) | Регистр признаков |
| r8 | rSP | Регистр указателя стека |
| r9 | rATI | Регистр адреса таблицы прерываний |

2.4. Система команд

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Мнемоника | Описание | Изменение признака Z |
| Суммирование | ADD r r\* | r:=r+r\*, IP=IP+1 | + |
| Вычитание | SUB r r\* | r:=r-r\*, IP=IP+1 | + |
| Добавление С | AD r C | r:=r+C, IP=IP+1 | + |
| Вычитание С | SB r C | r:=r-C, IP=IP+1 | + |
| Чтение в регистр | LD r A | r:=M[A], IP=IP+1 | - |
| Запись регистра | MV r A | M[a]:=r, IP=IP+1 | - |
| Чтение в регистр с индексацией | LDI r r\* | r:=M[r\*], IP=IP+1 | - |
| Запись в стек | PUSH r (SP) | M[SP]:=r, SP:=SP-1; IP:=IP+1 | - |
| Чтение из стека | POP r (SP) | SP:=SP+1, r:=M[SP], IP:=IP+1 | - |
| Переход | JMP A | IP:=A | - |
| Переход, если нуль | JZ A | Если Z=1, то IP:=A  иначе IP:=IP+1 | - |
| Обращение к подпрограмме | CALL (SP) A | M[SP]:=IP, SP:=SP-1 IP:=IP+1 | - |
| Возврат из подпрограммы | RET (SP) | SP:=SP+1, IP:=M[SP] | - |
| Остановка | HLT A | IP:=A, остановка | - |
| Загрузка маски | LM A | RM=M[A], IP:=IP+1 | - |

3. Кодирование программы и распределение памяти программ и данных

3.2. Коды операций

|  |  |  |
| --- | --- | --- |
| Название | Мнемоника | Код операции |
| Суммирование | ADD r r\* | 01 |
| Вычитание | SUB r r\* | 02 |
| Добавление С | AD r C | 9 |
| Вычитание С | SB r C | A |
| Чтение в регистр | LD r A | B |
| Запись регистра | MV r A | C |
| Чтение в регистр с индексацией | LDI r r\* | 0E |
| Запись в стек | PUSH r (SP) | 07 |
| Чтение из стека | POP r (SP) | 06 |
| Переход | JMP A | 03 |
| Переход, если нуль | JZ A | 04 |
| Обращение к подпрограмме | CALL (SP) A | D |
| Возврат из подпрограммы | RET (SP) | 05 |
| Остановка | HLT A | 00 |
| Загрузка маски | LM A | 0F |

3.3. Распределение памяти программ и данных

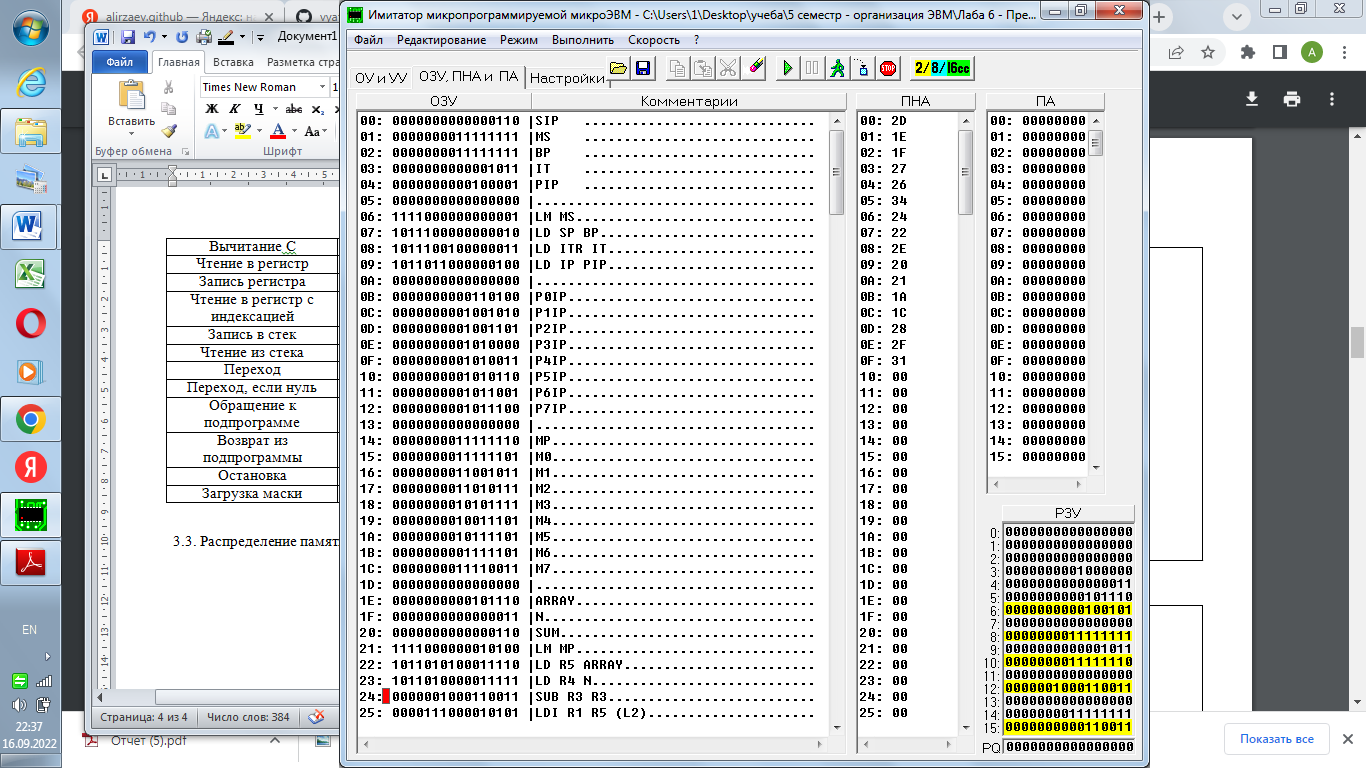


Рисунок 1 – Служебная программа и таблица прерываний

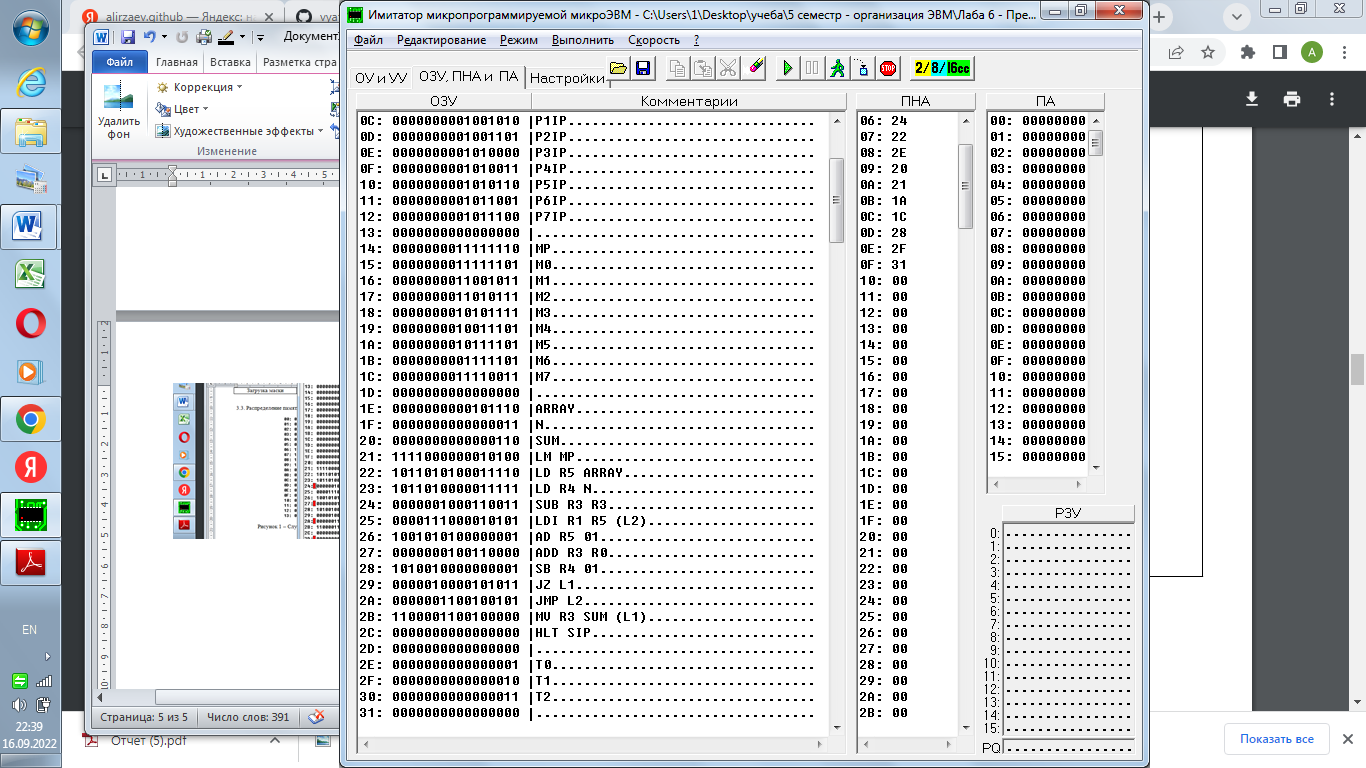


Рисунок 2 – маски и основная программа

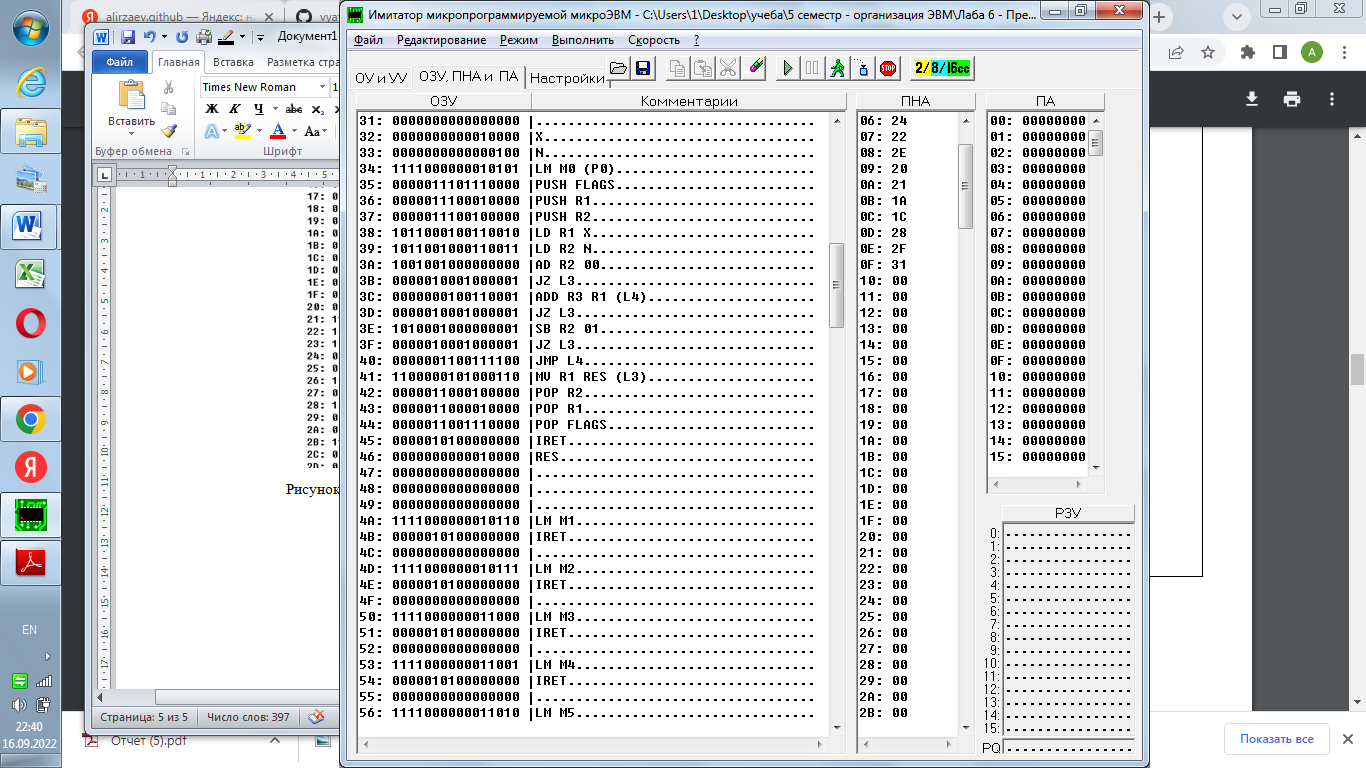


Рисунок 3 – программа P0

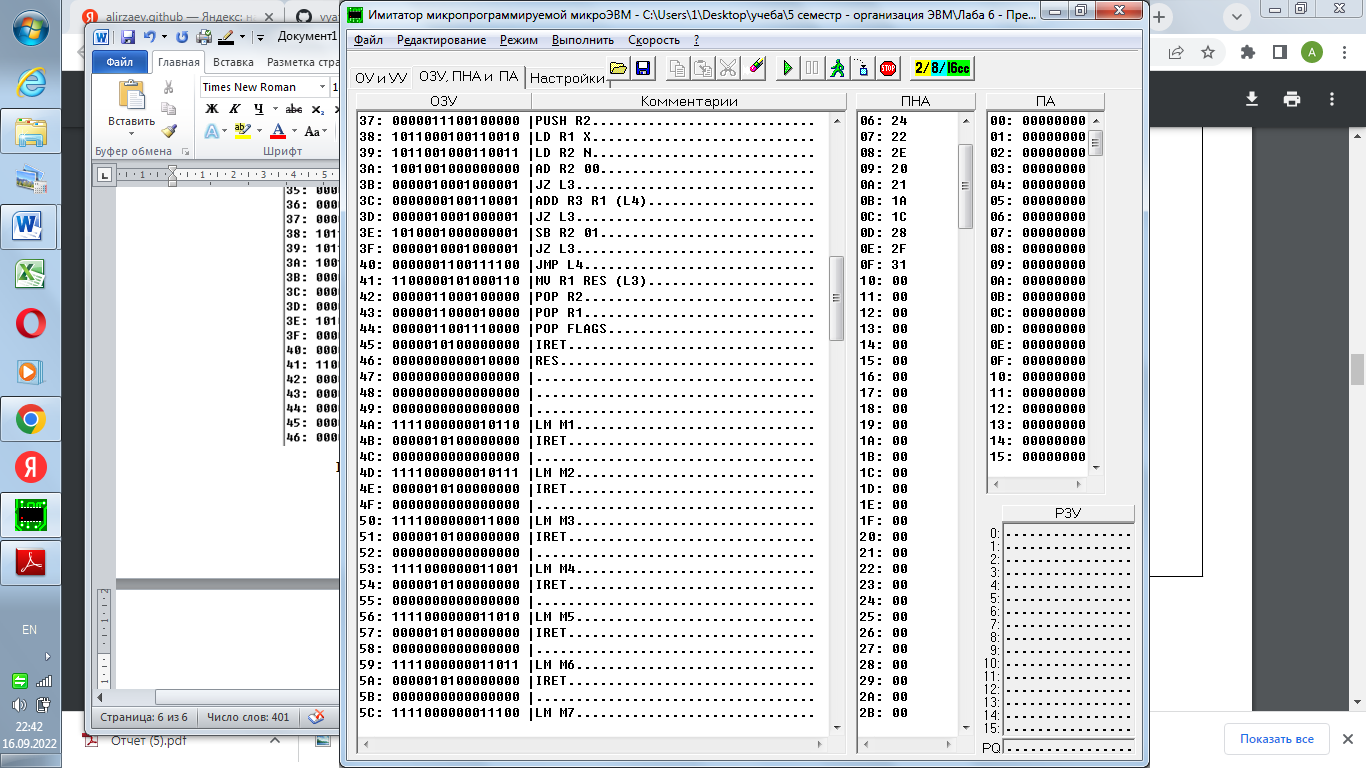


Рисунок 4 – программы P1-P5

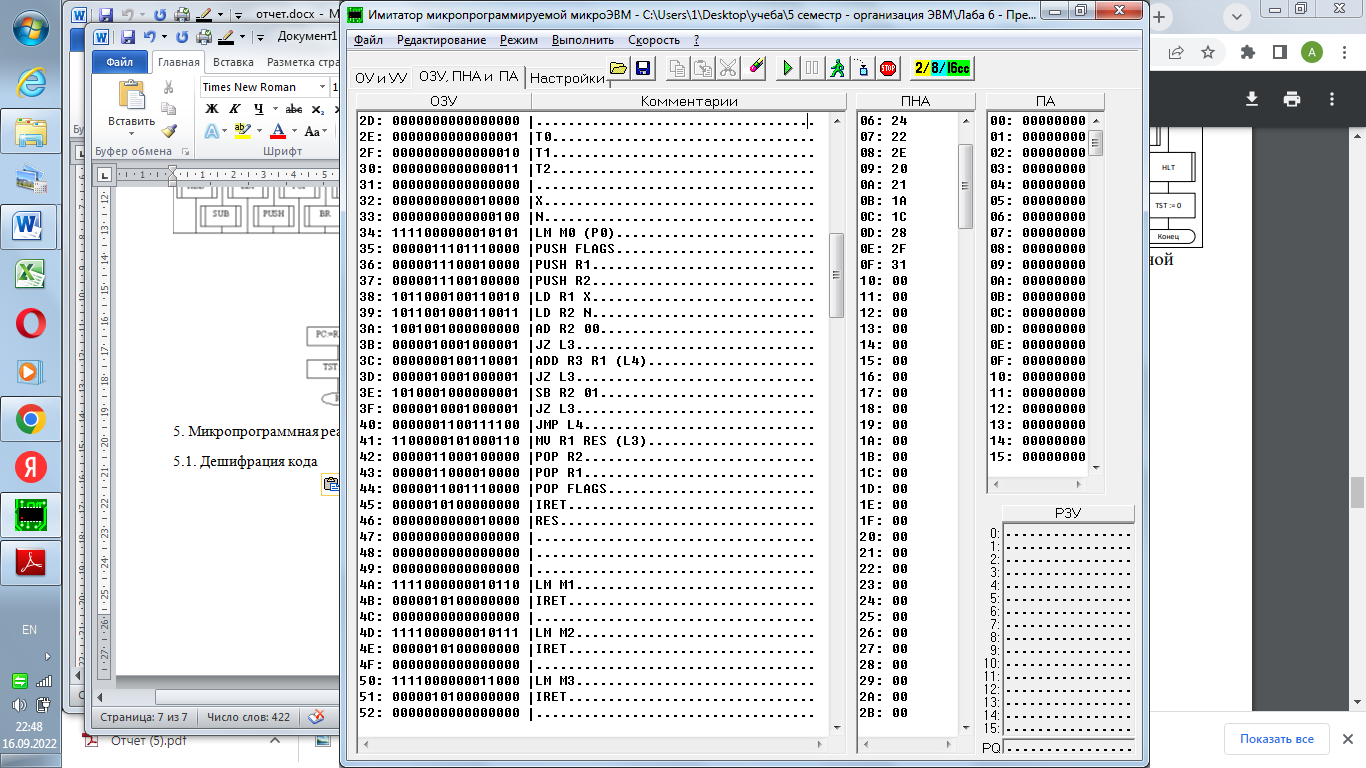
4. Разработка алгоритма и структуры программы

4.1. Алгоритм работы ЭВМ



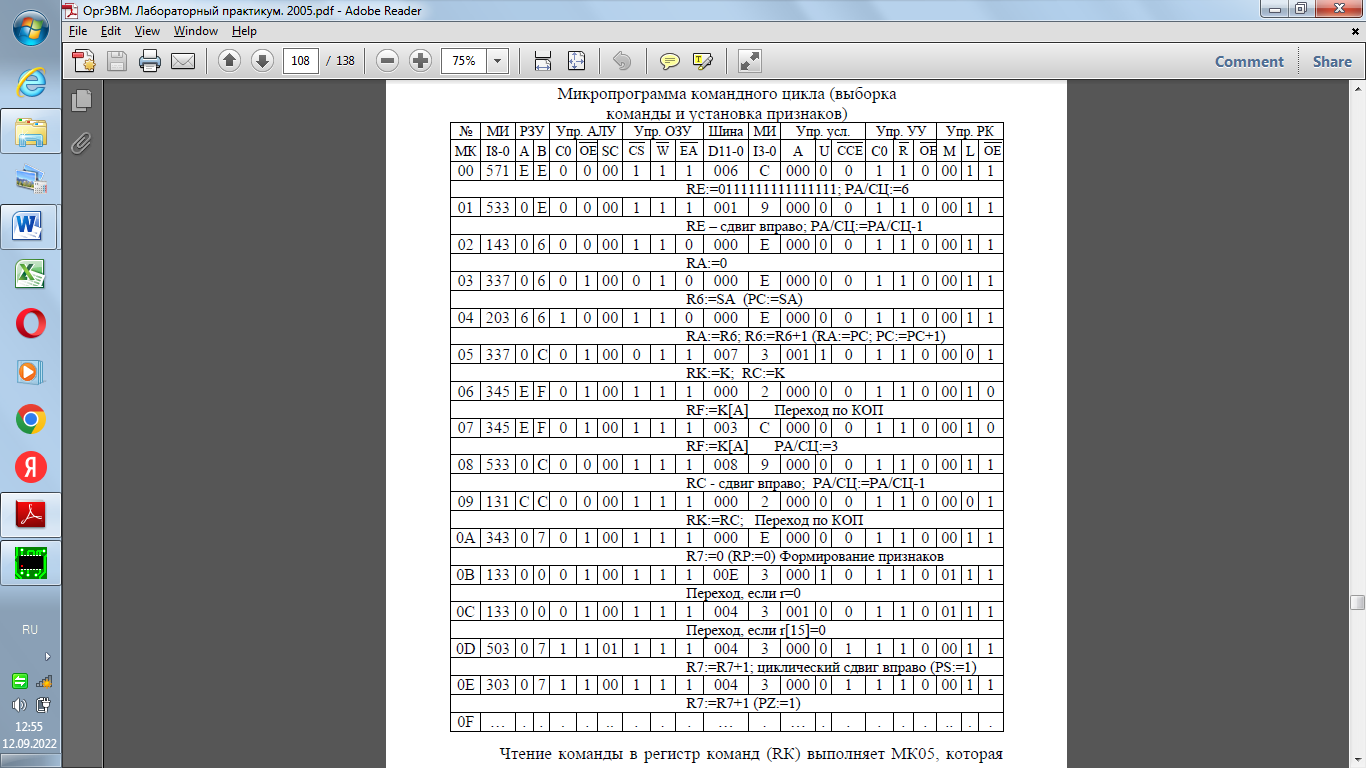
5. Микропрограммная реализация ЭВМ

5.1. Дешифрация кода

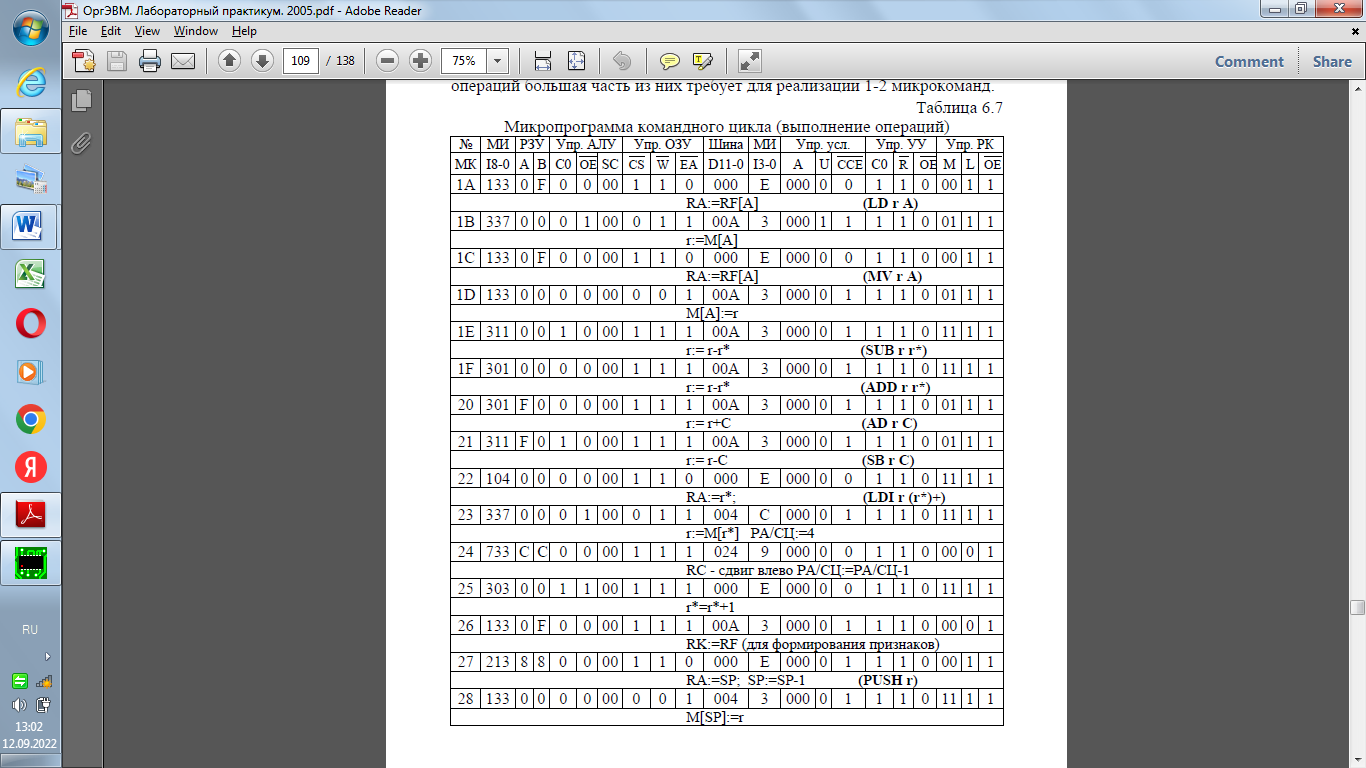


5.3. Микропрограммная реализация

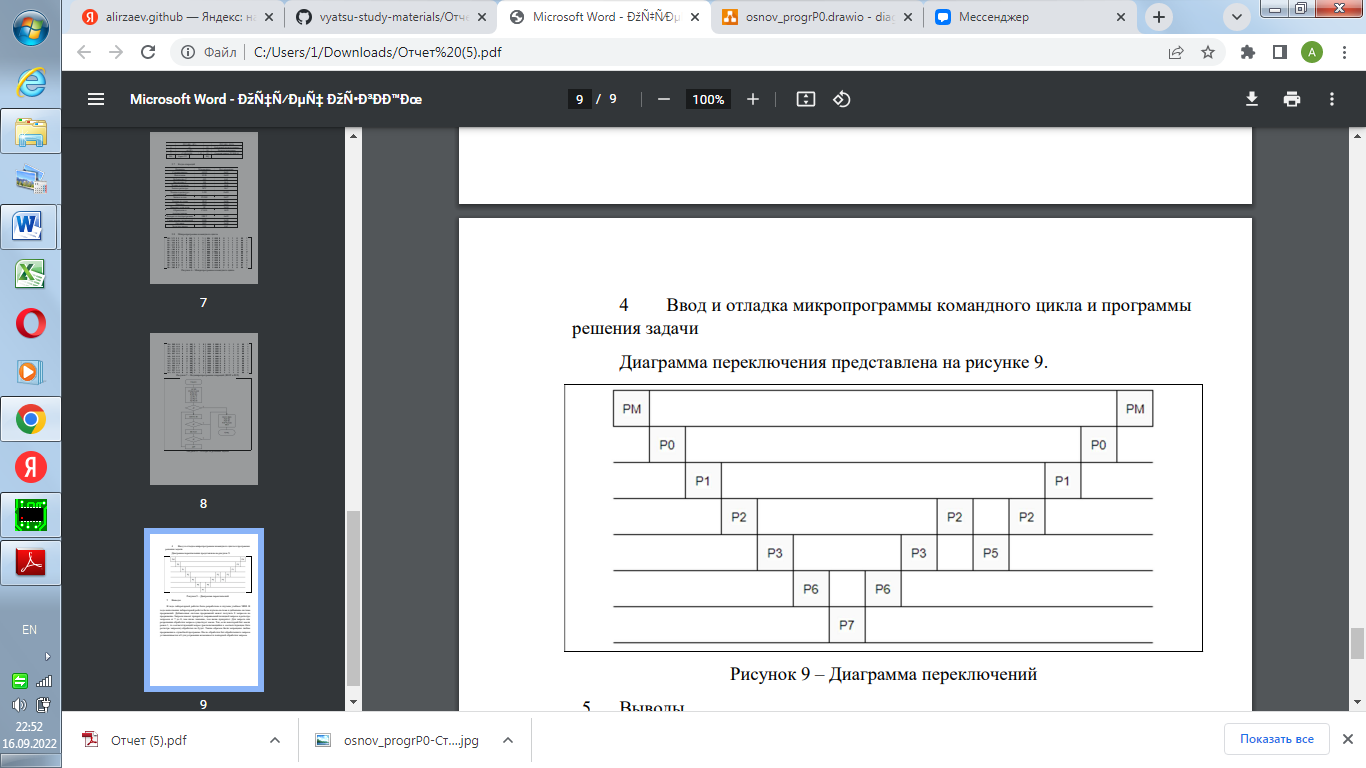
5.3.1 Микропрограмма командного цикла (выборка команды и установка признаков)



5.3.2. Микропрограмма командного цикла (выполнения операций)



6. Построение диаграммы прерываний



7. Вывод

В ходе лабораторной работы была разработана и изучена учебная ЭВМ. В ходе выполнения лабораторной работы была изучена система и добавлена система прерываний. Добавленная система прерываний может получать 8 запросов на прерывание. Запросы имеют приоритет, выражаемый позицией запроса в регистре запросов от 7 до 0, чем ниже значение, тем выше приоритет. Для запрета или разрешения обработки запроса существует маска. Так, если некоторый бит маски равен 1, то соответствующий запрос (располагающийся в соответствующем бите регистра запросов) обработан не будет. Таким образом были запрещены любые прерывания в служебной программе. После обработки бит обработанного запроса устанавливается в 0 для устранения возможности повторной обработки запроса.