МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Стеки типа LIFO и FIFO

Отчет по лабораторной работе №1 дисциплины

«Организация памяти ЭВМ»

Вариант 5

Выполнила студентка группы ИВТ-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кашина В.С./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В.Ю./

Киров 2022

**Задание на лабораторную работу**

* Исследовать стеки типа LIFO и FIFO:

• УС указывает на свободную ячейку с постдекрементом для LIFO;

• УС указывает на свободную ячейку с постинкрементом для FIFO;

• Начальный адрес ячейки ОЗУ, с которой начинается стек LIFO – 0A;

• Начальный адрес ячейки ОЗУ, с которой начинается стек FIFO – 4;

• Глубина стека – 9.

* Составить подмикропрограммы операций загрузки данных в стек и извлечения из стека.
* Для каждого стека выполнить последовательность следующих операций:
* запись 4-х чисел
* чтение 2-х чисел
* запись 2-х чисел
* чтение 3-х чисел
* запись пока стек не будет полон
* чтение пока стек не будет пуст.

1. **Исследование стека типа LIFO**

Для работы со стеком типа LIFO нужен указатель (SP), указывающий на последнюю занятую ячейку, а также указатель на начальный адрес стека (BP)

Стек оказывается полным, когда при очередной записи в стек, указатель SP будет указывать на адрес (BP)+1

Стек оказывается пустым, когда при очередном чтении из стека, указатель SP будет указывать на адрес (BP)+1

Схема взаимодействия стека типа LIFO приведена на рисунке 1.

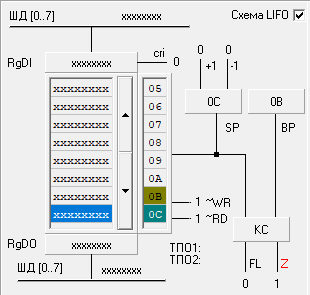


Рисунок 1 – схема взаимодействия стека типа LIFO

* **LIFO**

Запись

Для записи данных в стек необходимо:

1. Во входной регистр (RgDI) записать данные с входной шины данных (ШД) при помощи подачи управляющего сигнала CRI = 1.
2. Записать данные из входного регистра RgDI в ячейку памяти по адресу SP при помощи подачи сигнала ~WR = 0. В этом же такте необходимо уменьшить значение указателя SP на 1 при помощи подачи управляющего сигнала SP- = 1, тем самым выполнив декремент после записи в ячейку памяти (постдекремент).

Подмикропрограмма записи в стек представлена на рисунке 2, граф-схема алгоритма записи в стек представлена на рисунке 3.

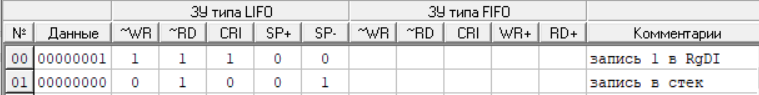


Рисунок 2 – подмикропрограмма записи в стек

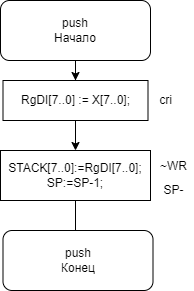


Рисунок 3 – граф-схема алгоритма записи в стек

Чтение

Для чтения данных из стека необходимо:

Подать сигнал чтения ~RD = 0, тем самым данные, находящиеся в ячейке памяти по адресу SP запишутся в выходной регистр RgDO и будут доступны на выходной шине данных (ШД). В этом же такте необходимо увеличить значение указателя SP на 1 при помощи подачи управляющего сигнала SP+ = 1, для того, чтобы SP указывал на последнюю занятую ячейку после чтения.

Подмикропрограмма чтения из стека представлена на рисунке 4, граф-схема алгоритма чтения из стек представлена на рисунке 5.

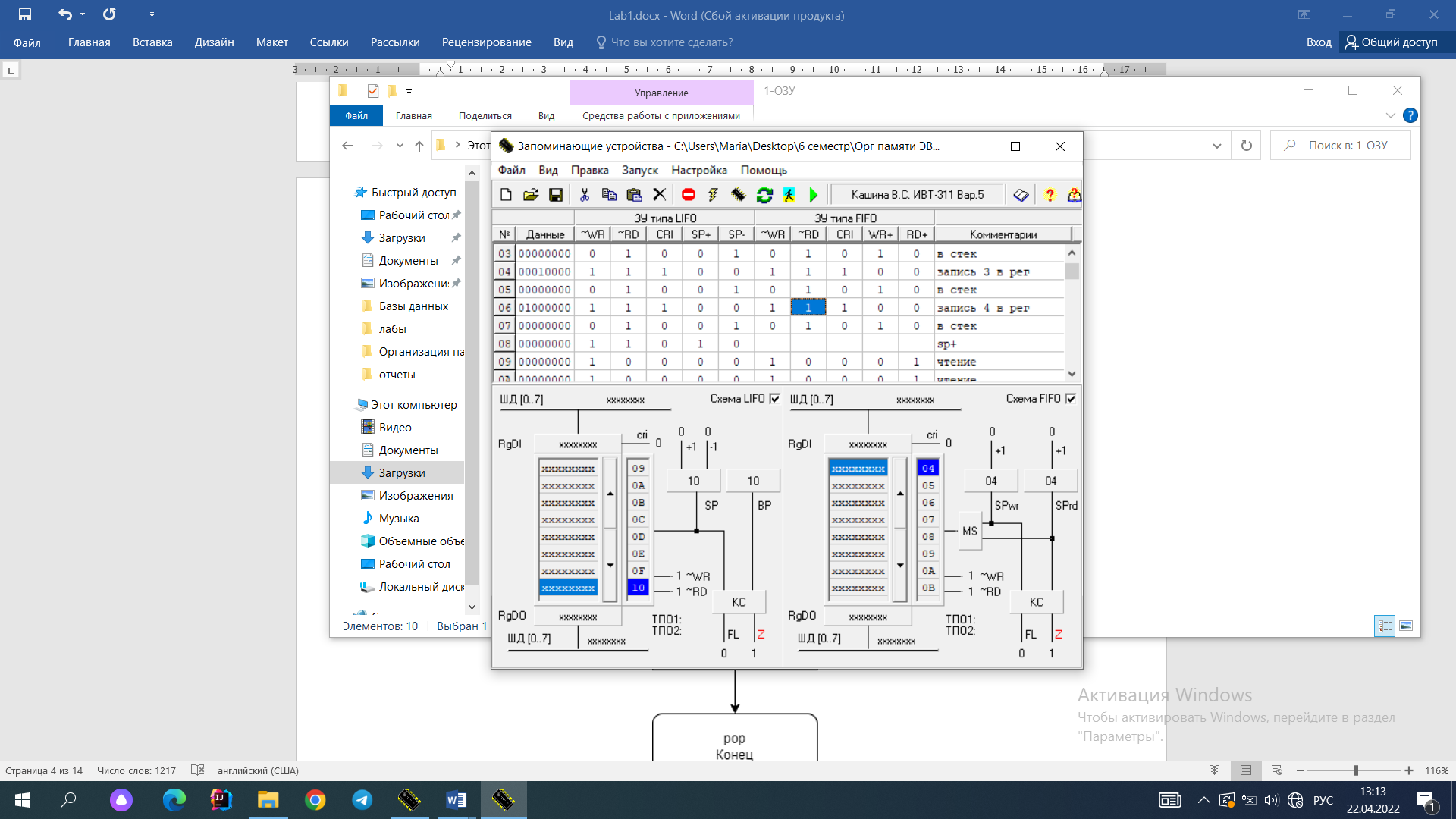


Рисунок 4 – подмикропрограмма чтения из стека

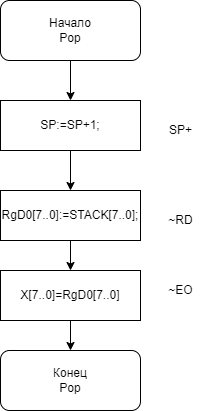
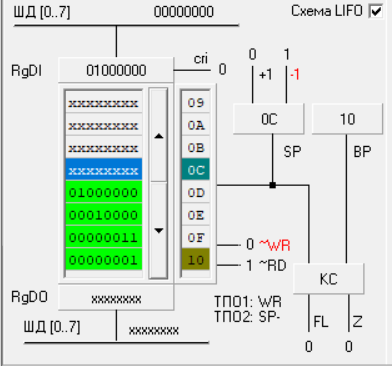


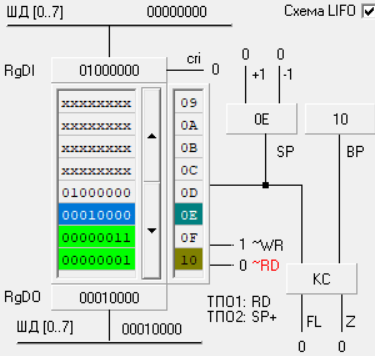
Рисунок 5 – граф-схема алгоритма чтения из стека

Последовательность выполнения команд:

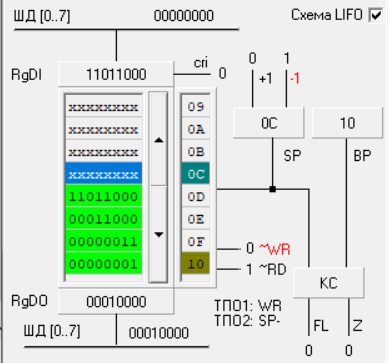
* запись 4-х чисел



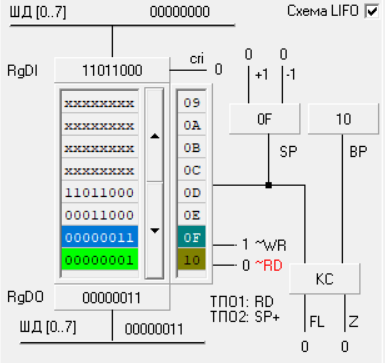
* чтение 2-х чисел



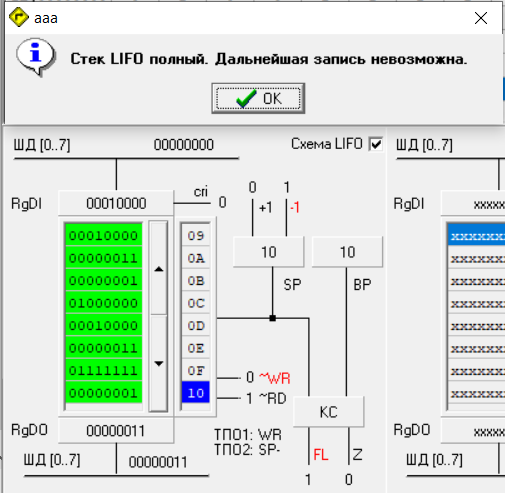
* запись 2-х чисел



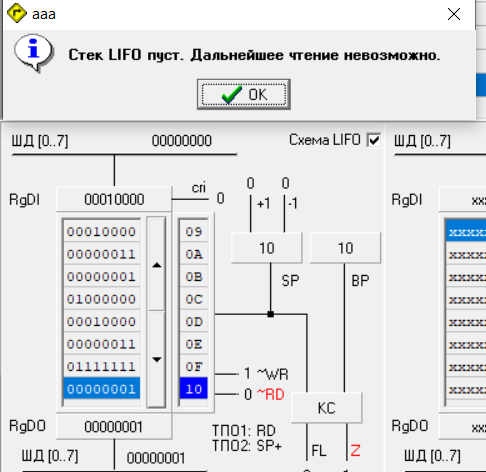
* чтение 3-х чисел



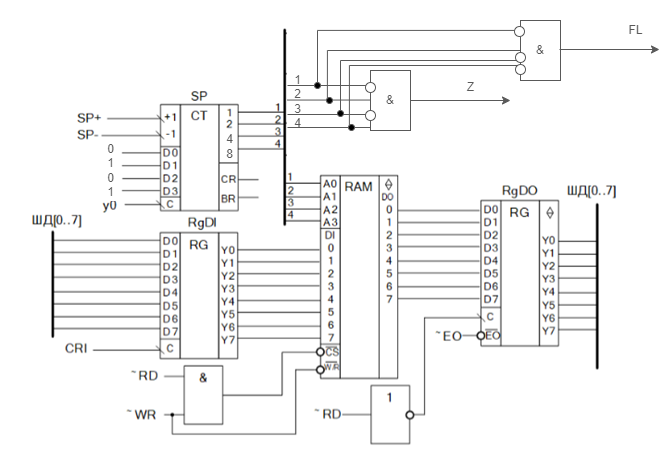
* запись пока стек не будет полон



* чтение пока стек не будет пуст



**Функциональная схема стека типа LIFO**



|  |  |
| --- | --- |
| Осведомительные сигналы:  Z = 1 – стек пуст;  FL = 1 – стек полон. | Управляющие сигналы:  y0 – запись адреса в SP;  ~EO – выдача данных с RgDO;  SP- – декремент SP;  SP+ – инкремент SP;  CRI – запись данных в RgDI;  ~RD – чтение из памяти;  ~WR – запись в память; |

**Исследование стека типа FIFO**

Для работы со стеком типа FIFO нужен указатель SPrd, указывающий на начало стека, и указатель SPwr, указывающий на конец стека.

Стек оказывается полным, когда при очередной записи в стек, указатель SPrd будет указывать на адрес SPwr.

Стек оказывается пустым, когда при очередном чтении из стека, указатель SPrd будет указывать на адрес SPwr.

Схема взаимодействия стека типа FIFO приведена на рисунке 7.

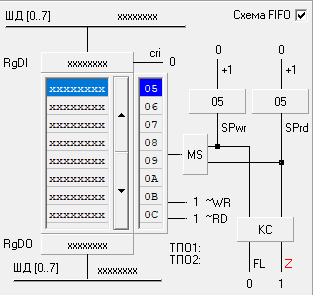


Рисунок 7 – схема взаимодействия стека типа FIFO

* **FIFO**

Запись

Для записи данных в стек необходимо:

1. Во входной регистр (RgDI) записать данные с входной шины данных (ШД) при помощи подачи управляющего сигнала CRI = 1.
2. Записать данные из входного регистра RgDI в ячейку памяти по адресу SPwr при помощи подачи сигнала ~WR = 0. В этом же такте необходимо увеличить значение указателя SPwr на 1 при помощи подачи управляющего сигнала WR+ = 1, тем самым выполнив инкремент после записи в ячейку памяти (постинкремент).

Подмикропрограмма записи в стек представлена на рисунке 8, граф-схема алгоритма записи в стек представлена на рисунке 9.

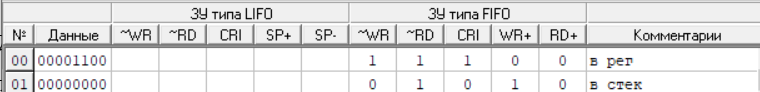


Рисунок 8 – подмикропрограмма записи в стек

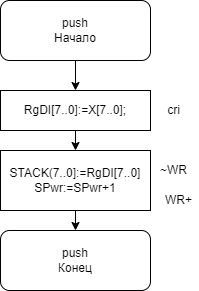


Рисунок 9 – граф-схема алгоритма записи в стек

Чтение

Для чтения данных из стека необходимо:

1. Увеличить значение указателя SPrd на 1 при помощи подачи управляющего сигнала RD+ = 1
2. Подать сигнал чтения ~RD = 0, тем самым данные, находящиеся в ячейке памяти по адресу SPrd запишутся в выходной регистр RgDO и будут доступны на выходной шине данных (ШД).

Подмикропрограмма чтения из стека представлена на рисунке 10, граф-схема алгоритма чтения из стек представлена на рисунке 11.

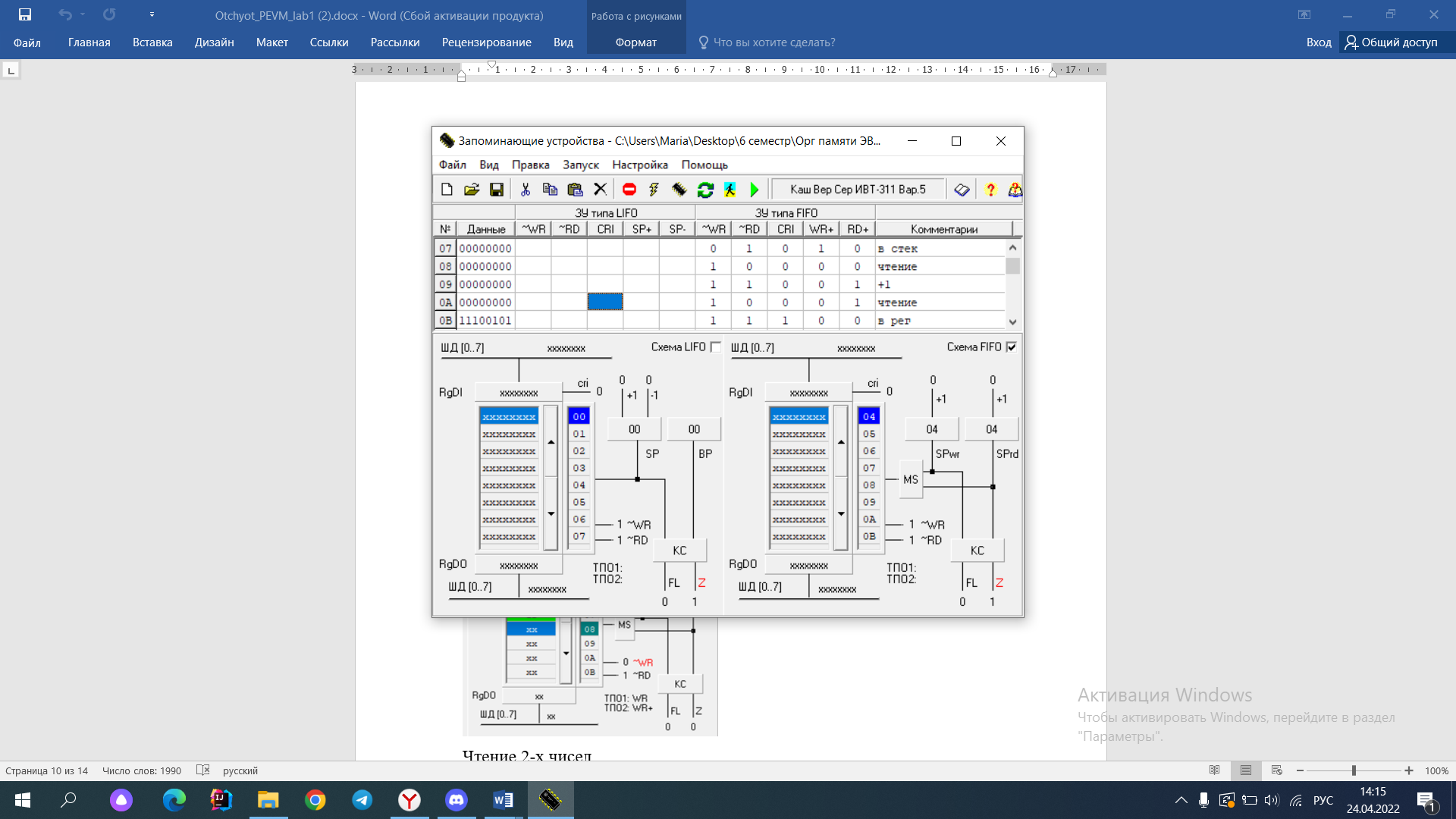


Рисунок 10 – подмикропрограмма чтения из стека

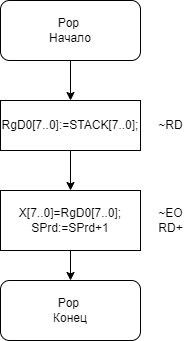


Рисунок 11 – граф-схема алгоритма чтения из стека

**Функциональная схема стека типа FIFO**

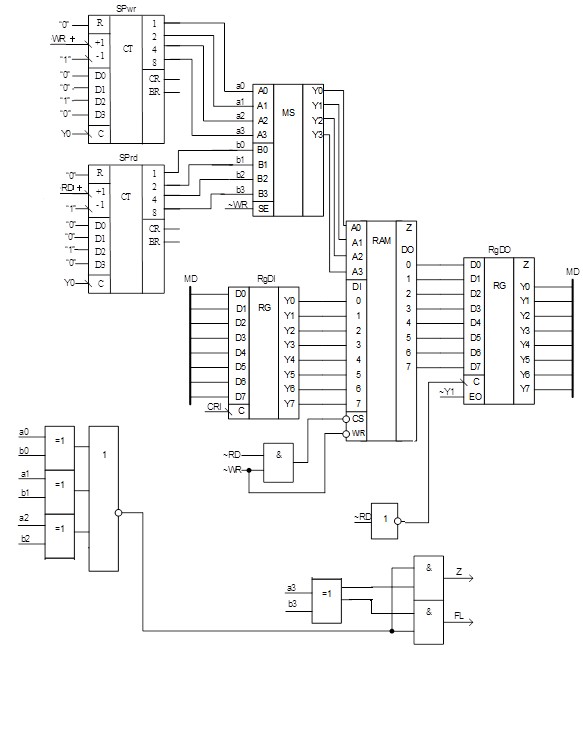


Рисунок 12 - Функциональная схема стека типа FIFO

|  |  |
| --- | --- |
| Осведомительные сигналы:  Z = 1 – стек пуст;  FL = 1 – стек полон. | Управляющие сигналы:  Y0 – запись адреса в SP;  ~Y1 – выдача данных с RgDO;  WR+ – инкремент SPwr;  RD+ – инкремент SPrd;  CRI – запись данных в RgDI;  ~RD – чтение из памяти;  ~WR – запись в память; |

**Вывод**

В ходе выполнения лабораторной работы были изучены принципы работы стеков типа LIFO и FIFO

Стек типа LIFO – стек, в котором последние записанные в стек данные считаются первыми. Для работы со стеком данного типа потребовался перемещаемый указатель, который по условиям задания, всегда указывает на последний занятый элемент памяти. Помимо перемещаемого указателя, для работы со стеком LIFO потребовался адрес начала стека, который всегда принимает одно статическое значение. Стек оказывается полным, если при очередной записи в стек, а именно при подаче сигнала декремента перемещаемого указателя происходит ситуация, при которой адрес перемещаемого указателя на единицу больше адреса начала стека, при этом формируется сигнал FL = 1. Стек оказывается пустым, если при очередном чтении из стека, а именно при подаче сигнала инкремента перемещаемого указателя происходит ситуация, при которой адрес перемещаемого указателя на единицу больше адреса начала стека, при этом формируется сигнал Z = 1.

Стек типа FIFO – стек, в котором последние записанные в стек данные считаются последними, то есть имеет место быть очереди данных. Для работы со стеком данного типа потребовался перемещаемый указатель на начало очереди, с помощью которого производится извлечение из стека. Помимо перемещаемого указателя на начало очереди, необходим указатель на конец очереди, с помощью которого производится вставка в стек, и который, по условиям задания, всегда указывает на последние вошедшие в очередь данные. Стек оказывается полным, если при очередной записи в стек, а именно при подаче сигнала инкремента указателя на конец очереди происходит ситуация, при которой адреса указателей начала и конца очереди совпадают, при этом формируется сигнал FL = 1. Стек оказывается пустым, если при очередном чтении из стека, а именно при подаче сигнала инкремента указателя на начало очереди происходит ситуация, при которой адреса указателей начала и конца очереди совпадают, при этом формируется сигнал Z = 1.