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 №1

по курсу «Организация памяти ЭВМ»

Вариант 7

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

Проверил доцент кафедры ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Мельцов В.Ю./

Киров 2023

1. Задание на лабораторную работу

Необходимо разработать микропрограмму для прединкрементного LIFO, начинающегося по адресу 0х03h и имеющего глубину 9 ячеек. Разработать микропрограмму для преинкрементного FIFO, начинающегося по адресу 0х06h и имеющего глубину 7 ячеек.

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.

2. 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 в случае работы с установкой и (BP)- 1+глубина стека в случае работы с реальной памятью.

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

Функциональная схема LIFO приведена на рисунке 1.

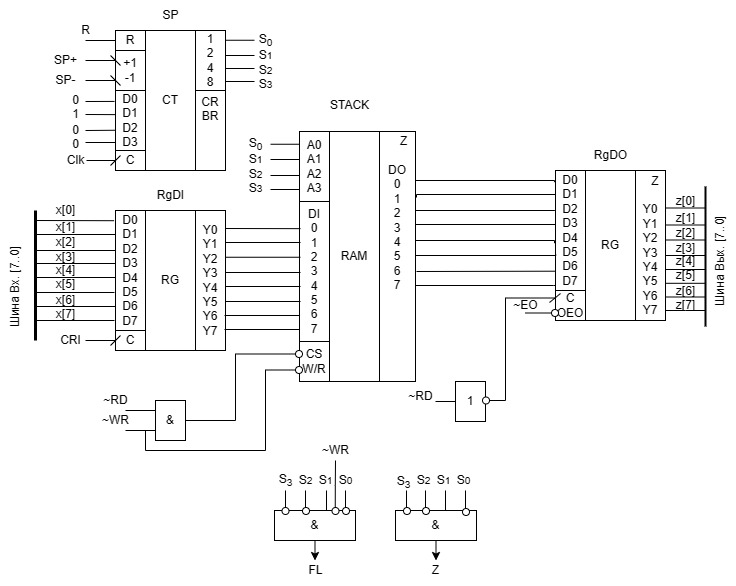


Рисунок 1 – Функциональная схема LIFO

2.1. Запись

Для записи данных в стек необходимо:

Во входной регистр (RgDI) записать данные с входной шины данных (ШД) при помощи подачи управляющего сигнала cri = 1. В этом же такте необходимо увеличить значение указателя SP на 1 при помощи подачи управляющего сигнала SP+ = 1, тем самым выполнив инкремент перед записью в ячейку памяти (прединкремент).

Записать данные из входного регистра RgDI в ячейку памяти по адресу SP при помощи подачи сигнала ~WR = 0.

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

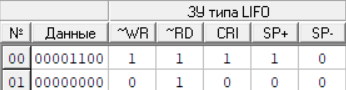


Рисунок 2 – Микропрограмма записи в стек

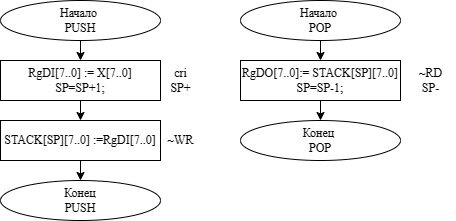


Рисунок 3 – ГСА операции PUSH для LIFO

2.2. 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) при окончании импульса сигнала. 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. Подача сигнала SP- в этом же такте возможна лишь в том случае, когда подача происходит после окончания импульса сигнала на чтение. Временная диаграмма с указанием управляющих сигналов представлена на рисунке 4.

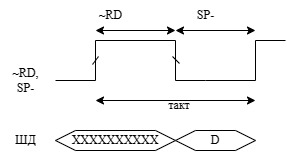


Рисунок 4 – Временная диаграмма

Подмикропрограмма чтения из стека представлена на рисунке 5, граф-схема алгоритма чтения из стек представлена на рисунке 6.

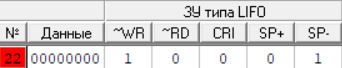


Рисунок 5 – Микропрограмма чтения из стека

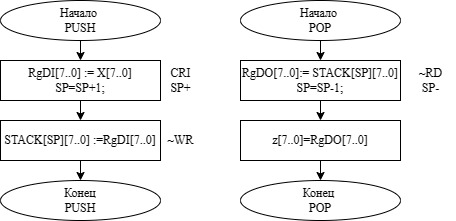


Рисунок 6 – ГСА операции POP для LIFO

2.3. Выполнение последовательности операций

2.3.1. Запись 4х чисел

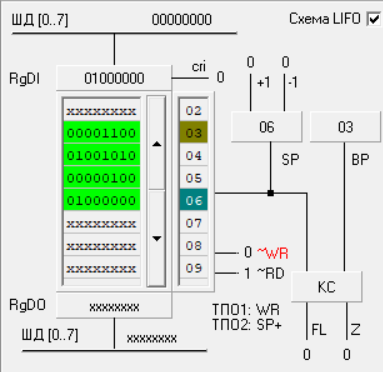


Рисунок 7 – Запись 4х чисел

2.3.2. Чтение 2х чисел

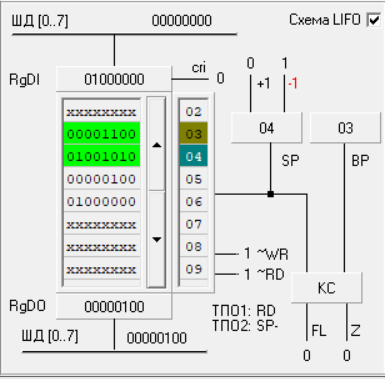


Рисунок 8 – Чтение 2х чисел

2.3.3. Запись 2х чисел

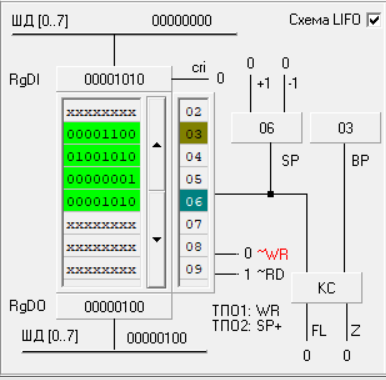


Рисунок 9 – Запись 2х чисел

2.3.4. Чтение 3х чисел

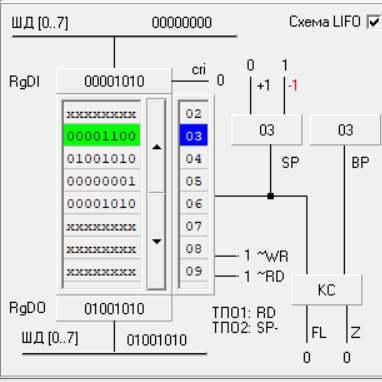


Рисунок 10 – чтение 3х чисел

2.3.5. Запись до полного стека

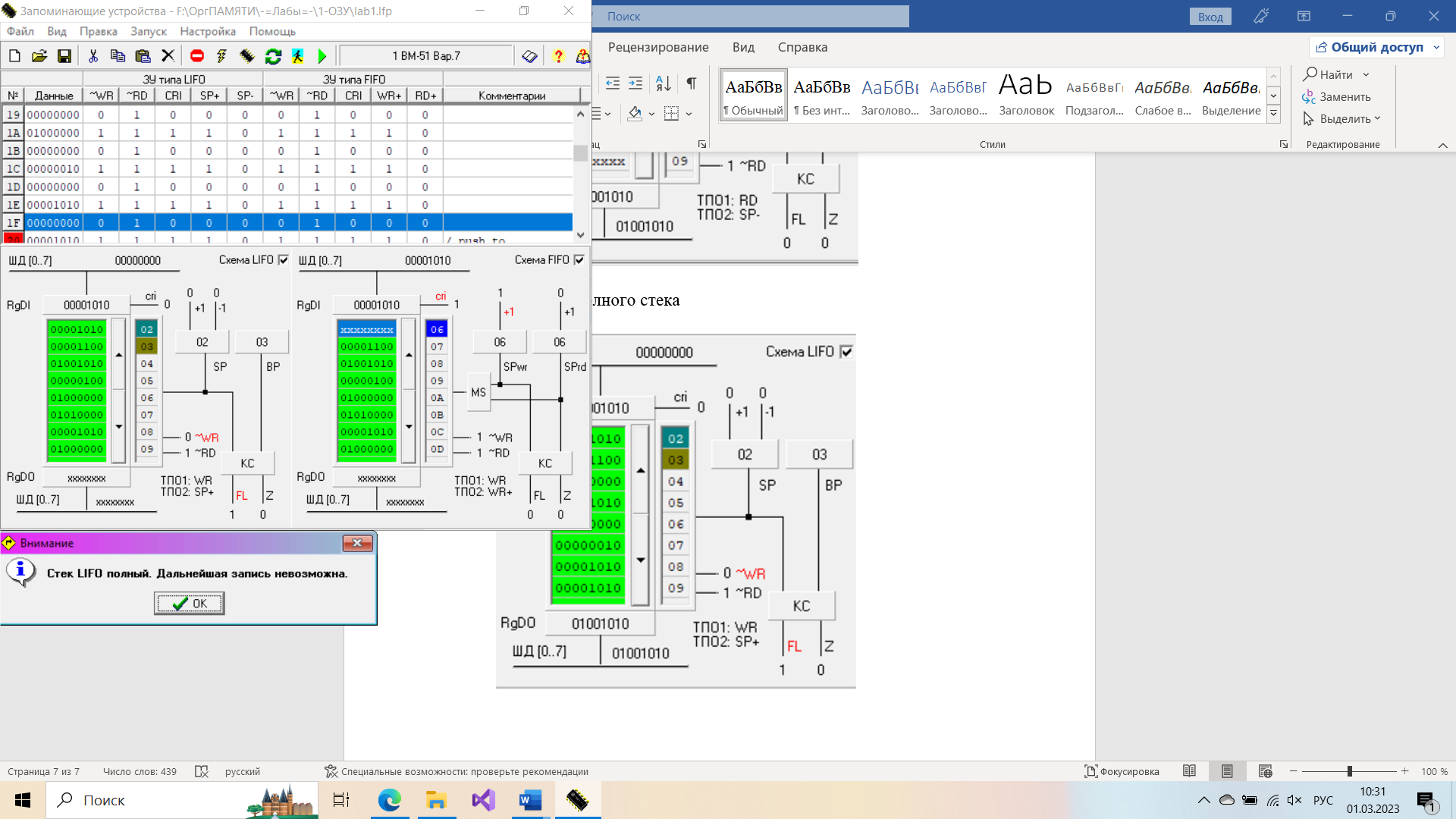
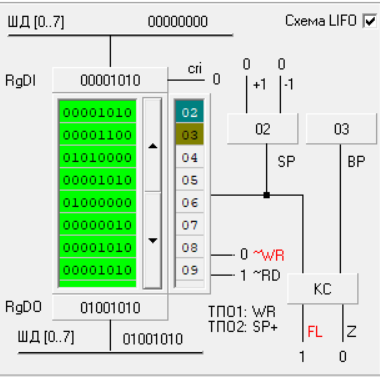


Рисунок 10 – Запись до полного стека

2.3.6. Чтение до пустого стека

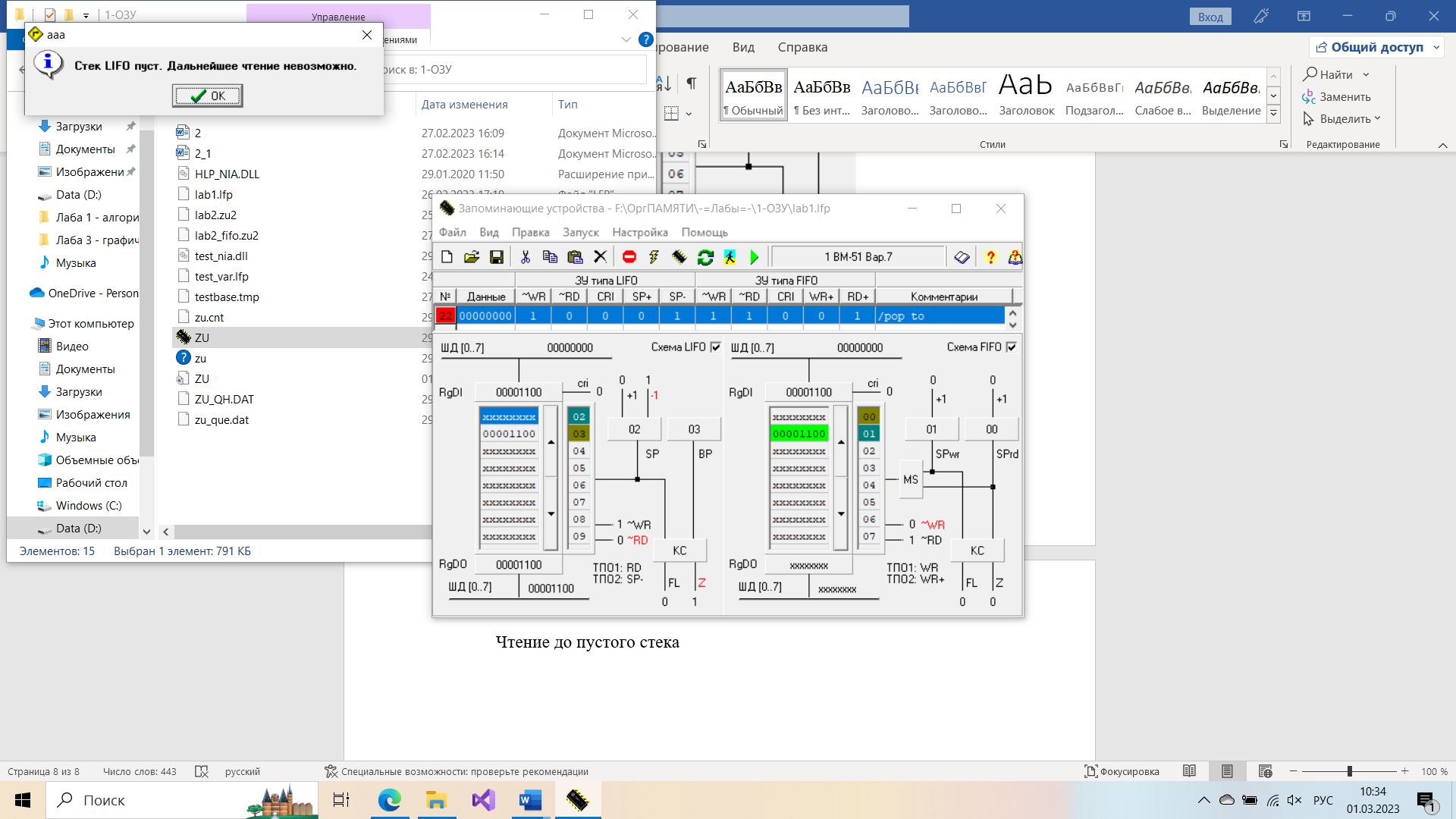
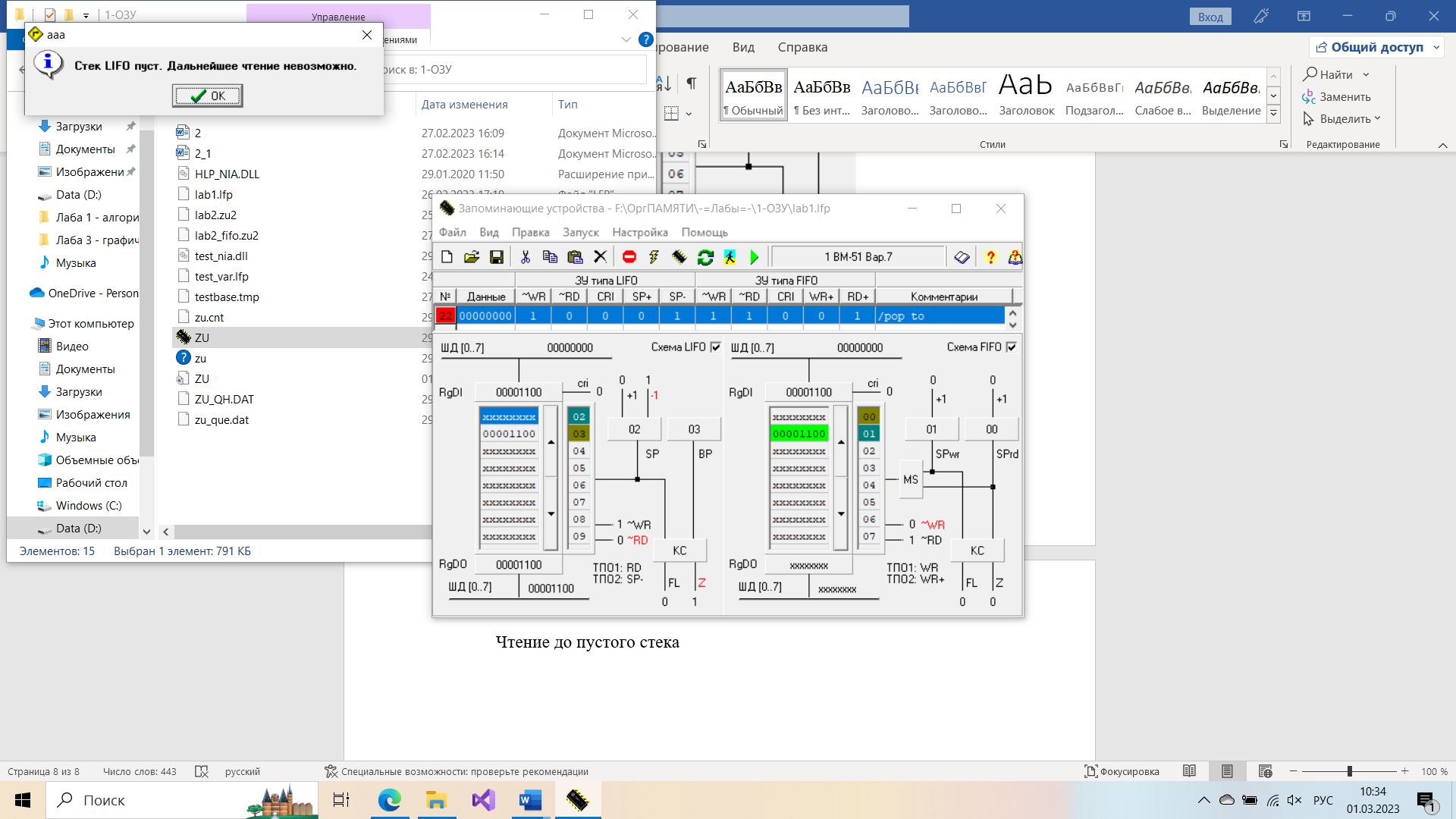


Рисунок 11 – Чтение до пустого стека

3. 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

Функциональная схема FIFO приведена на рисунке 12.

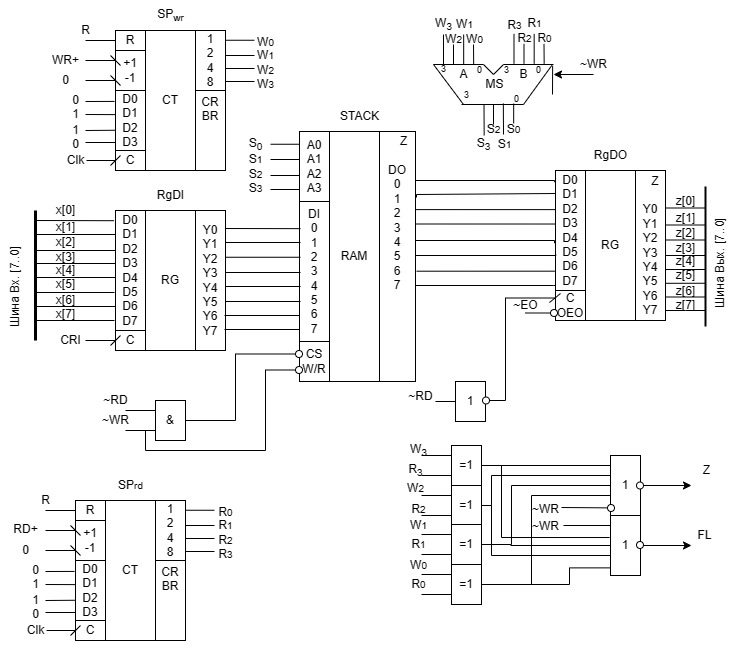


Рисунок 12 – Функциональная схема FIFO

3.1. Запись

Для записи данных в стек необходимо:

Во входной регистр (RgDI) записать данные с входной шины данных (ШД) при помощи подачи управляющего сигнала cri = 1. В этом же такте необходимо увеличить значение указателя SPwr на 1 при помощи подачи управляющего сигнала WR+ = 1, тем самым выполнив инкремент перед записью в ячейку памяти (прединкремент).

Записать данные из входного регистра RgDI в ячейку памяти по адресу SPwr при помощи подачи сигнала ~WR = 0.

Подмикропрограмма записи в стек представлена на рисунке 13, граф-схема алгоритма записи в стек представлена на рисунке 14.

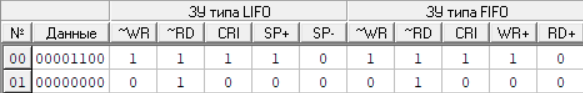
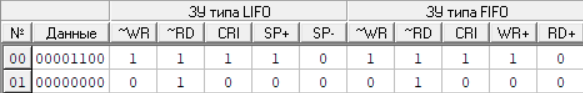


Рисунок 13 – Микропрограмма записи в стек

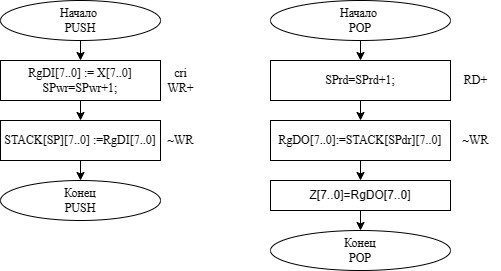


Рисунок 14 – ГСА операции PUSH для FIFO

3.2. Чтение

Для чтения данных из стека необходимо:

Увеличить значение указателя SPrd на 1 при помощи подачи управляющего сигнала RD+ = 1

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 15, граф-схема алгоритма чтения из стек представлена на рисунке 16.

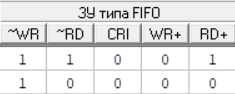


Рисунок 15 – Микропрограмма записи из стека

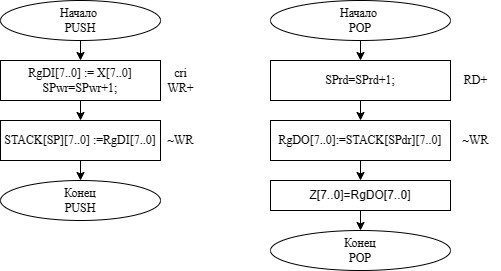


Рисунок 16 – ГСА операции POP для FIFO

3.3. Выполнение последовательности операций

3.3.1. Запись 4х чисел

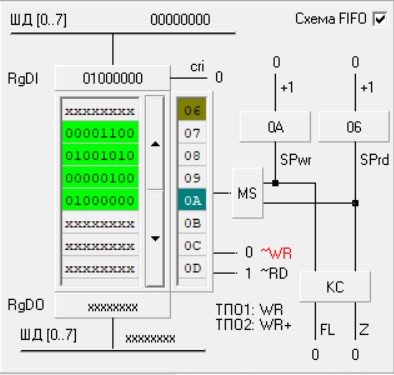


Рисунок 17 – Запись 4х чисел

3.3.2. Чтение 2х чисел

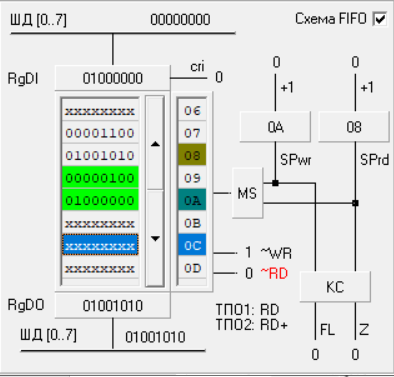


Рисунок 18 – Чтение 2х чисел

3.3.3. Запись 2х чисел

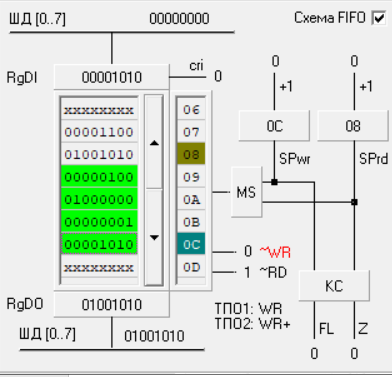
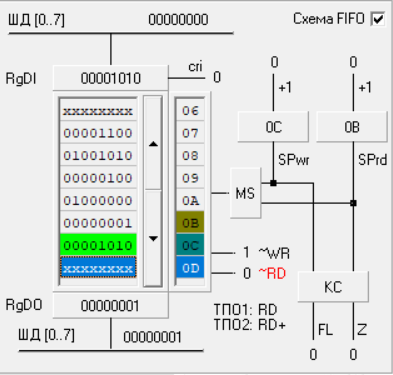


Рисунок 19 – Запись 2х чисел

3.3.4. Чтение 3х чисел

  
Рисунок 20 – Чтение 3х чисел

3.3.5. Запись до полного стека

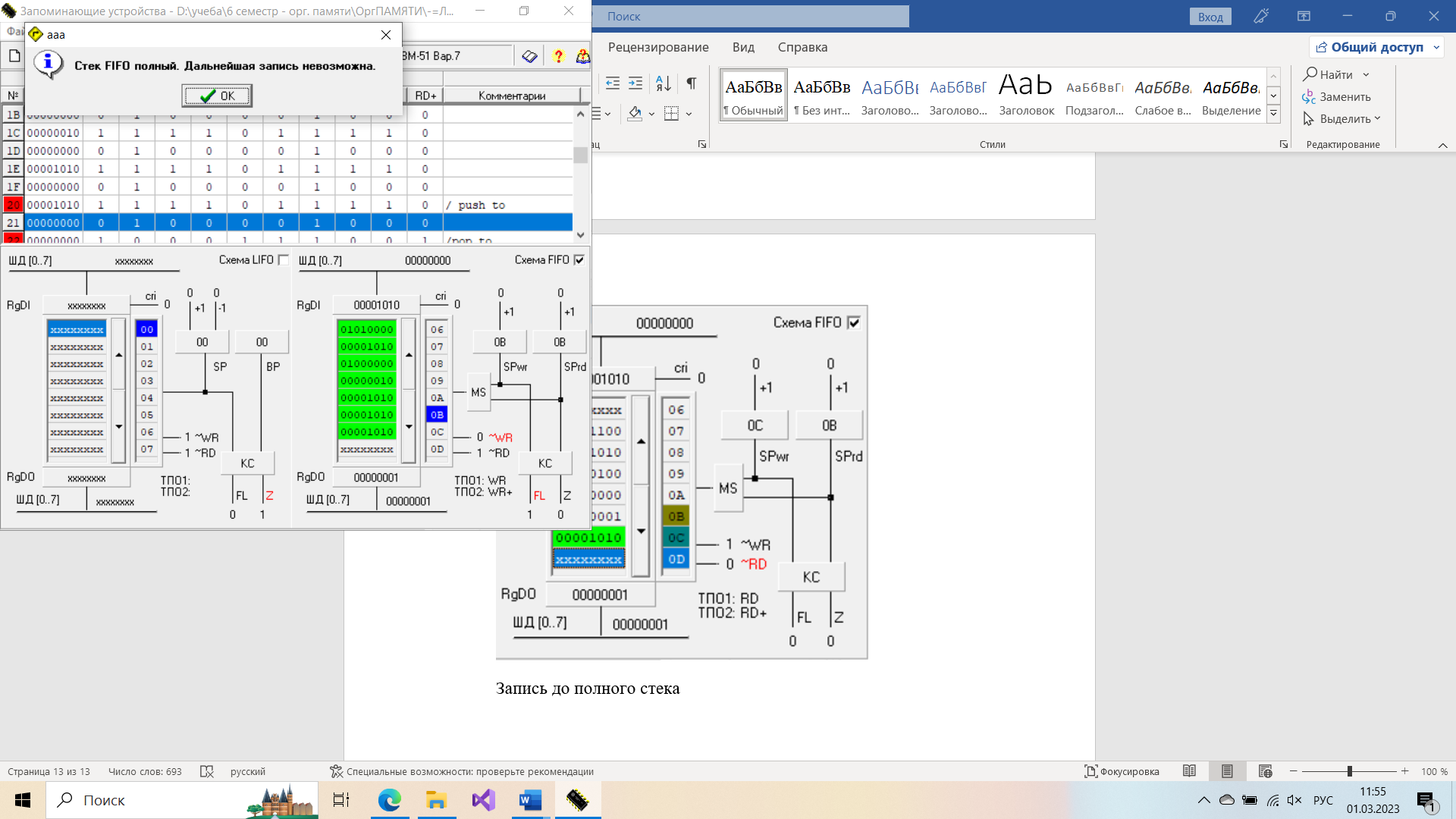
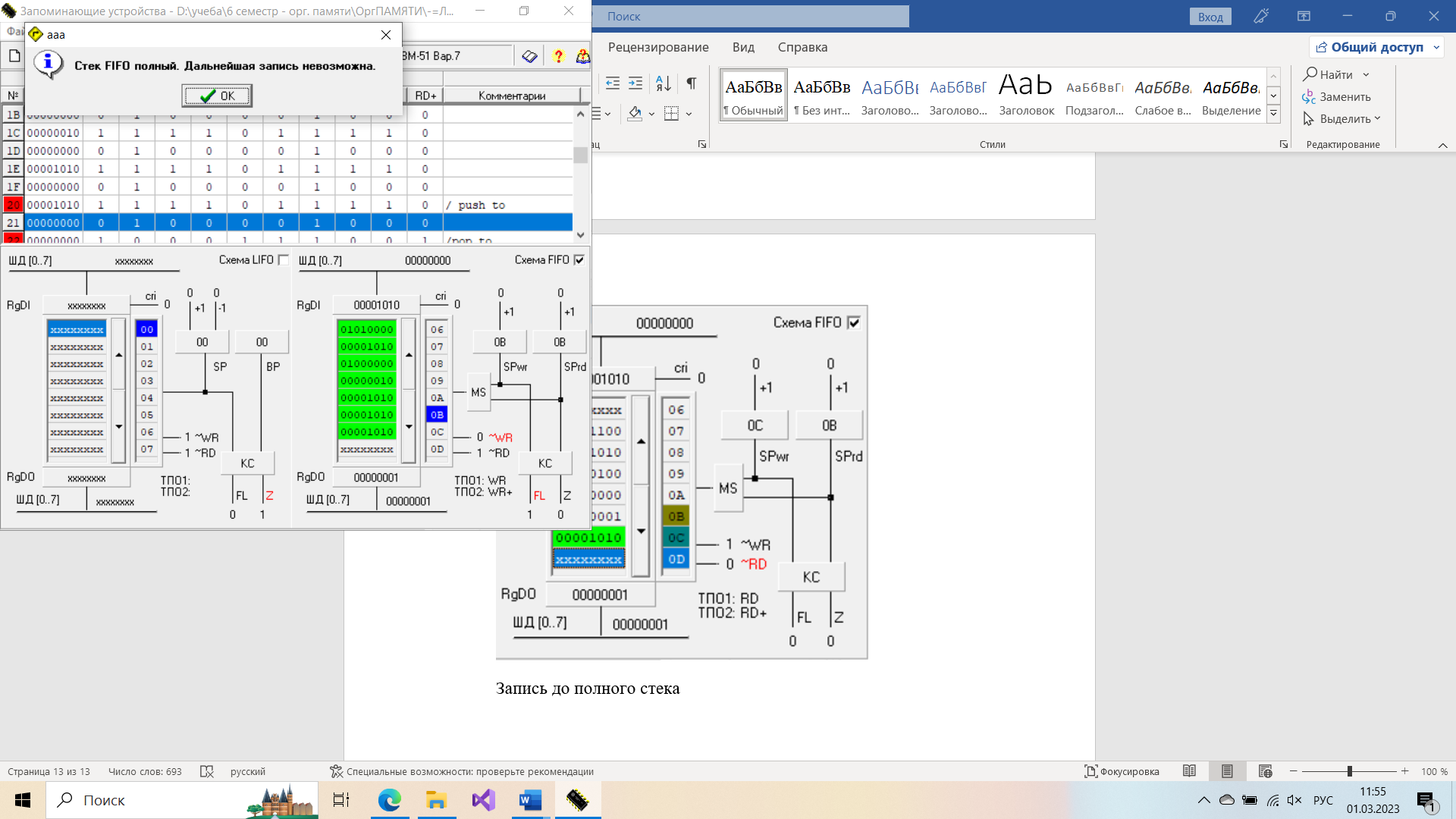


Рисунок 21 – Запись до полного стека

3.3.6. Чтение до пустоты

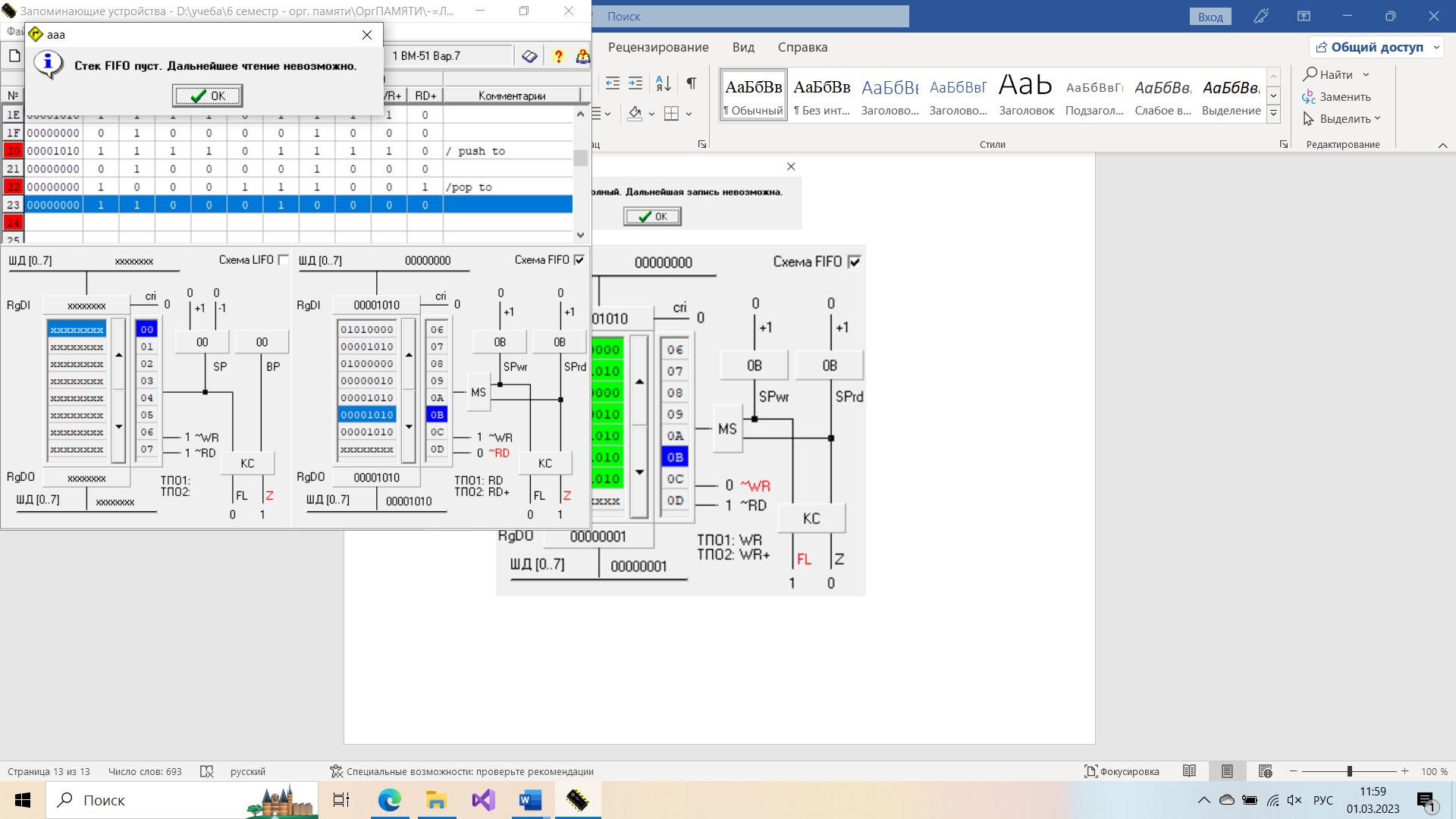
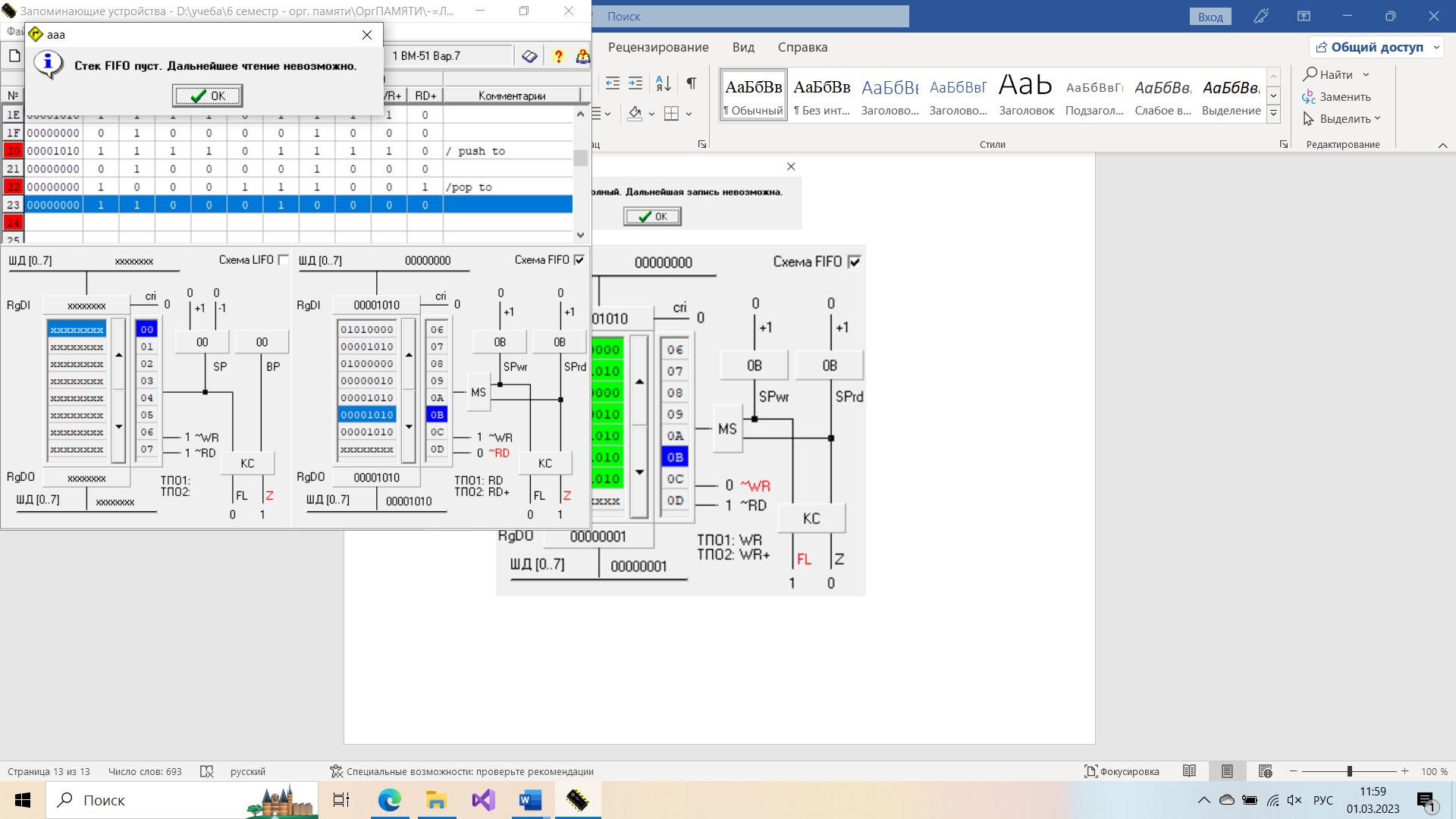


Рисунок 22 – Чтение до пустоты

4. 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.