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

Отчет по лабораторной работе №1 дисциплины

«Организация памяти ЭВМ»

Вариант 10

Выполнил студент группы ИВТб-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Миночкин Г.О./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Июдин И.Д./

Киров 2019

**Задание на лабораторную работу**

1. Исследовать стеки типа LIFO и FIFO:

• УС указывает на занятую ячейку памяти с преддекрементом;

• Начальный адрес ячейки ОЗУ, с которой начинается стек LIFO – 13;

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

• Глубина стека – 8.

* 1. Составить подмикропрограммы операций загрузки данных в стек и извлечения из стека.
  2. 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. Исследовать АЗУ.

Разработать подмикропрограмму, выполняющую следующие функции:

* 1. Записать в ячейки АЗУ произвольные константы в любые 6-7 ячеек.
  2. Загрузить в регистр маски RgM маску по тем разрядам, по которым будет осуществляться ассоциативный поиск (от 3 до 5 бит).
  3. Загрузить во входной регистр RgI эталонное значение для выполнения ассоциативного поиска.
  4. Выполнить чтение из АЗУ. При первом чтении введен дополнительный такт для наблюдения числа совпадений при поиске.
  5. Количество чтений необходимо выполнить столько раз, пока в регистре сдвига RgSH не будет установлен код 0.
  6. Дозагрузить свободные ячейки АЗУ данными и повторить выполнение п.2.4-2.5 для различных значений эталонов в RgI и RgM.

1. Исследовать двухпортовое ОЗУ.
   1. Исследовать ОЗУ в режиме произвольного доступа при записи и чтении:

а) выполнить запись данных во все ячейки ОЗУ в режимах:

* записи одновременно по порту А и В;
* раздельной записи по одному из портов А и В.

б) выполнить сочетание процедур чтения и записи одновременно по портам А и В:

* порт А чтение, порт В запись;
* порт В чтение, порт А запись;
* порт В чтение, порт А чтение;
* раздельное чтение по порту А или В.

в) выполнить попытку записи по портам А и B в одну и ту же ячейку и сделать выводы.

* 1. На основе ОЗУ организовать стек типа FIFO для очереди команд с возможностью параллельного пополнения очереди команд через каждые 4 считанные из очереди команды:
* запись 8-х чисел
* чтение 4-х чисел
* запись 4-х чисел с параллельным считыванием из очереди
* сброс очереди команд (команда БП)

**Выполнение задания**

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.

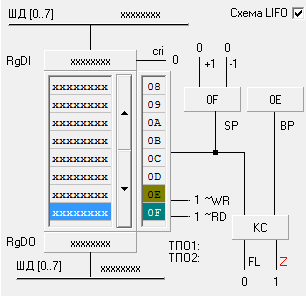


Рисунок 1

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. В этом же такте необходимо уменьшить значение указателя SP на 1 при помощи подачи управляющего сигнала SP- = 1, тем самым выполнив декремент перед записью в ячейку памяти (преддекремент).
2. 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.

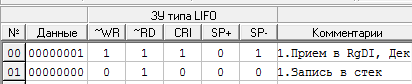


Рисунок 2

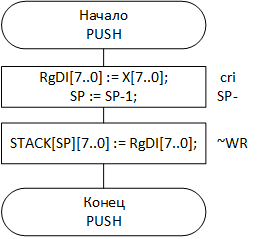


Рисунок 3

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

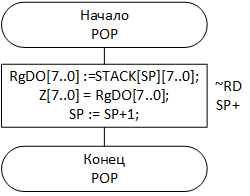


Рисунок 5

Выполнение последовательности операций:

Запись 4-х чисел

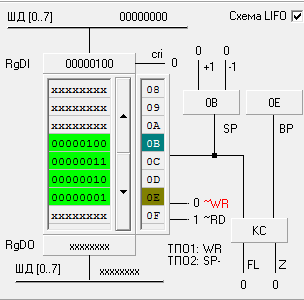


Рисунок 6

Чтение 2-х чисел

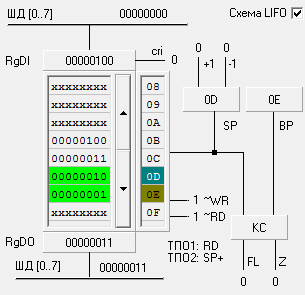


Рисунок 7

Запись 2-х чисел

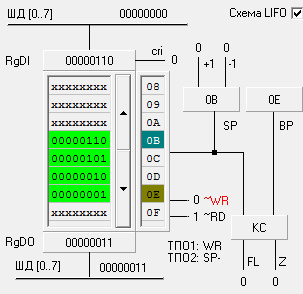


Рисунок 8

Чтение 3-х чисел

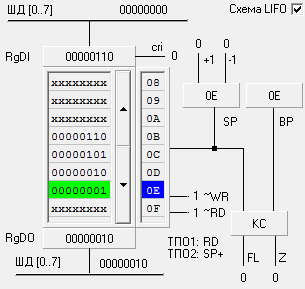


Рисунок 9

Запись до переполнения

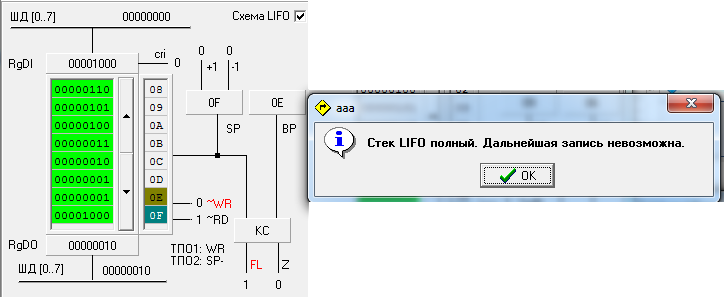


Рисунок 10

Чтение до пустоты

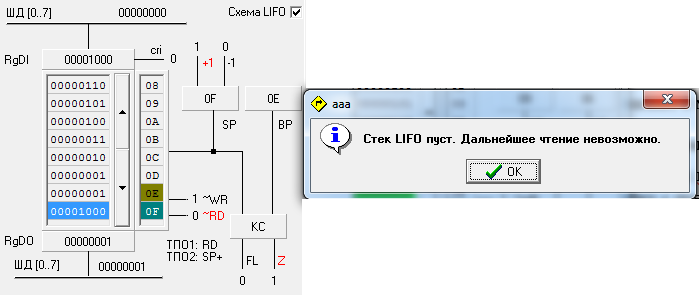


Рисунок 11

**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 12.

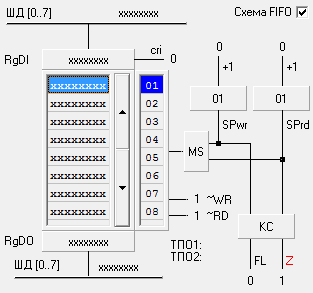
****

Рисунок 12

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. 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).
2. 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.

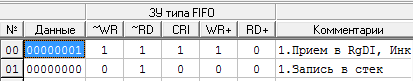


Рисунок 13



Рисунок 14

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

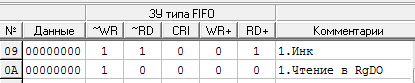


Рисунок 15



Рисунок 16

Выполнение последовательности операций:

Запись 4-х чисел

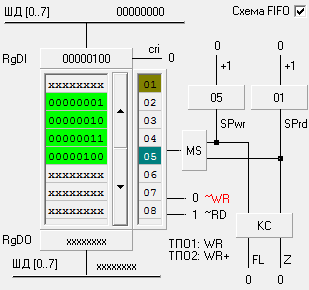


Рисунок 17

Чтение 2-х чисел

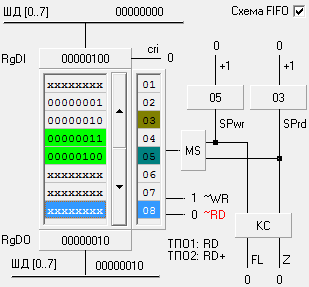


Рисунок 18

Запись 2-х чисел

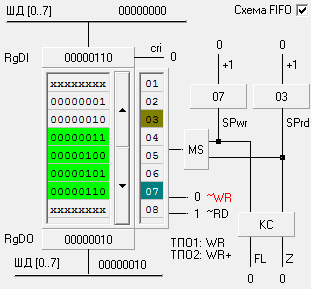


Рисунок 19

Чтение 3-х чисел

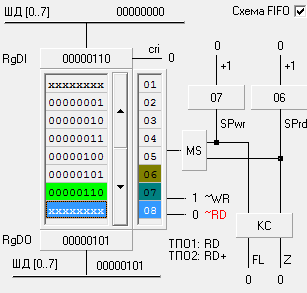


Рисунок 20

Запись до переполнения

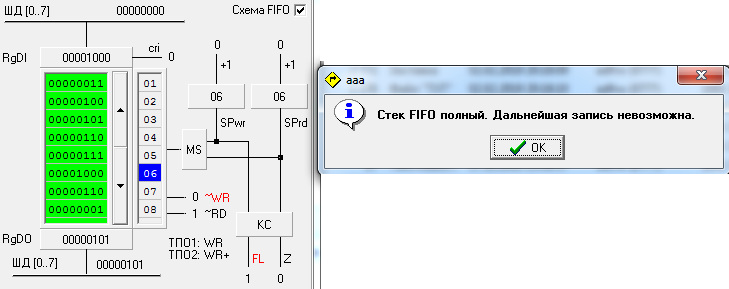


Рисунок 21

Чтение до пустоты

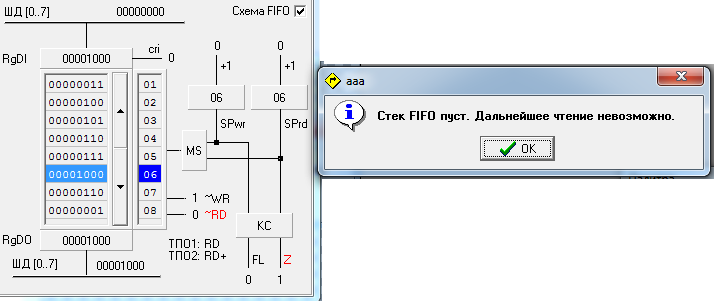


Рисунок 22

1. **Исследование АЗУ**

Схема взаимодействия с АЗУ представлена на рисунке 23.

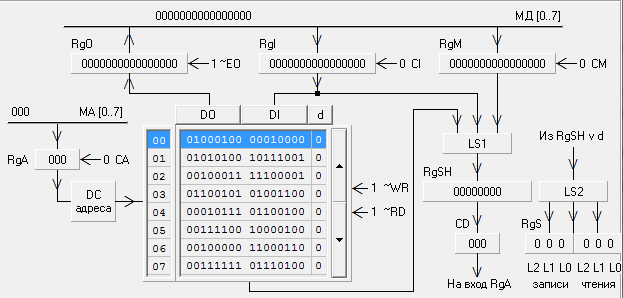


Рисунок 23

Запись:

Для записи данных в память АЗУ необходимо:

* Записать с шины данных (МД) во входной регистр (RgI) данные при помощи подачи управляющего сигнала CI = 1;
* Записать с шины данных (МД) в регистр адреса (RgA) адрес ячейки памяти при помощи подачи управляющего сигнала CA = 1;
* Записать данные из RgI в АЗУ по адресу из RgA при помощи подачи сигнала ~WR = 0.

Чтение:

Для поиска данных по признаку необходимо:

* Записать с шины данных (МД) в регистр маски (RgM) маску, которая будет указывать на то, какие разряды будут проверяться, при помощи подачи сигнала CM = 1;
* Записать с шины данных (МД) во входной регистр (RgI) эталонное значение, с которым будут проверяться данные, при помощи подачи сигнала CI = 1;
* Выполнить поиск при помощи подачи сигнала ~RD = 0, при этом в регистре RgS сформируются флаги результата поиска:

L2 – найдено более одного совпадения,

L1 – найдено одно совпадение,

L0 – не найдено совпадений.

Если найдено более одного совпадения, то производится выдача всех данных через регистр RgO, которые удовлетворяют замаскированному эталонному значению.

Подмикропрограмма взаимодействия с АЗУ представлена на рисунке 24, граф-схема алгоритмов записи и поиска представлена на рисунке 25.

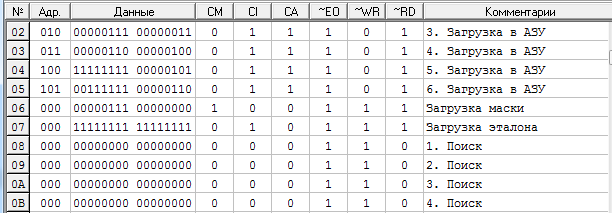


Рисунок 24



Рисунок 25

Выполнение последовательности операций:

Запись произвольных констант в ячейки АЗУ

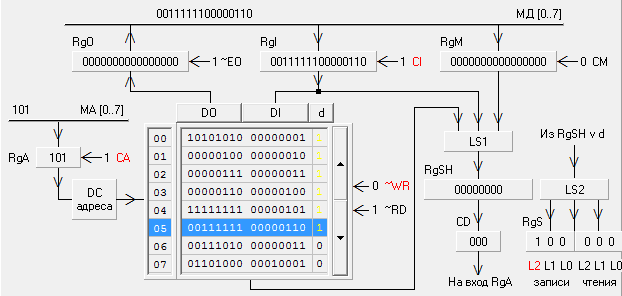


Рисунок 26

Загрузка маски и эталона

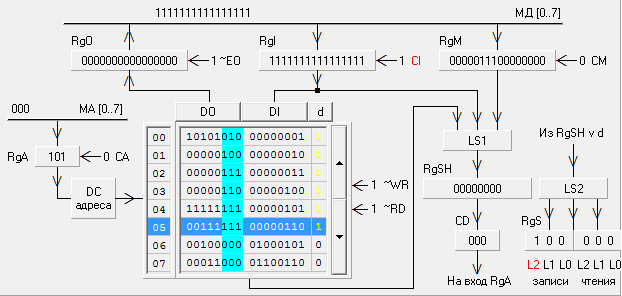


Рисунок 27

Выполнение чтения из АЗУ

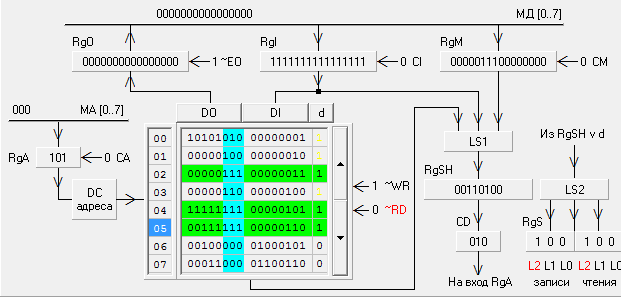


Рисунок 28

После выполнения первого чтения появился сигнал L2 чтения, говорящий о нахождении более одного совпадения.

Чтение первого совпадения по адресу 010

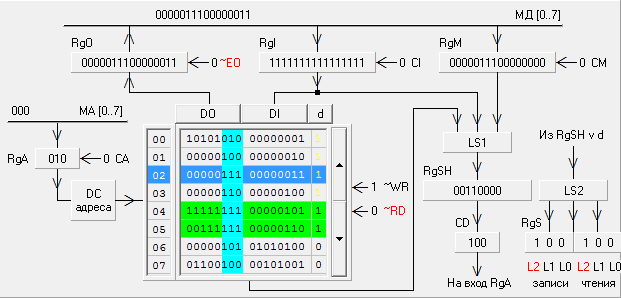
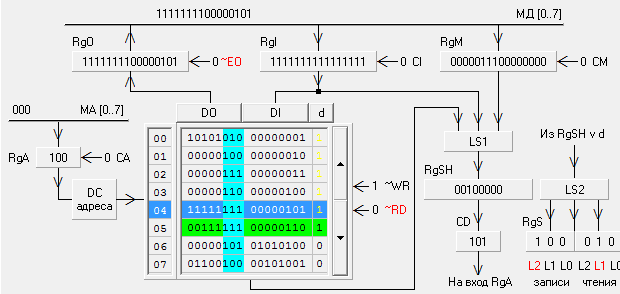


Рисунок 29

L2 чтения – более одного совпадения, продолжаем чтение

Чтение второго совпадения по адресу 100

 Рисунок 30

L1 чтения – одно совпадение, продолжаем чтение

Чтение третьего совпадения по адресу 101

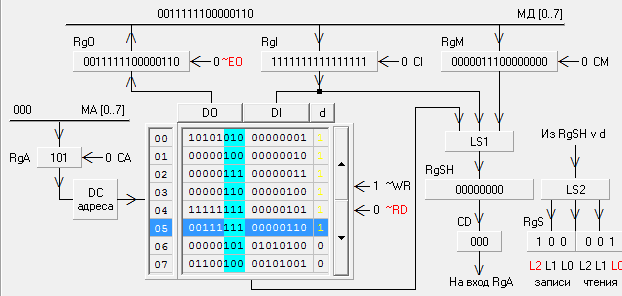
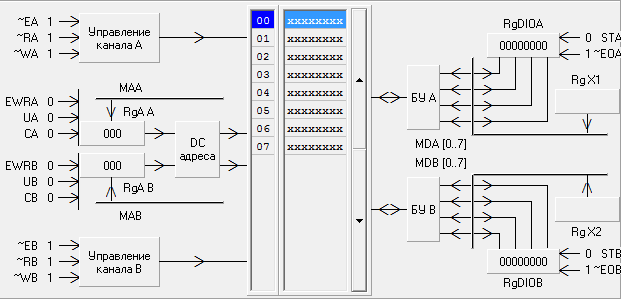


Рисунок 31

L0 чтения – ни одного совпадения, конец чтения

1. **Исследование двухпортового ЗУ**

Схема взаимодействия с двухпортовым ЗУ представлена на рисунке 32.

*Рисунок 32*

Запись:

Для записи входных данных в память двухпортового ЗУ необходимо:

1. Во входной регистр канала А (RgDIOA) и/или канала B (RgDIOB) записать данные с шины данных (ШД) при помощи подачи управляющих сигналов STA = 1 и/или STB = 1 соответственно. В этом же такте нужно в регистр адреса канала А (RgAA) и/или канала В (RgAB) записать адрес ячейки памяти ЗУ, в которую будет осуществляться запись, при помощи подачи управляющих сигналов EWRA = 1 и/или EWRB = 1 соответственно
2. Записать данные в память ЗУ по адресу RgAA и/или RgAB из RgDIOA и/или RgDIOB соответственно, при помощи подачи управляющих сигналов ~WA = 0, ~EA = 0 и/или ~WB = 0, ~EB = 0 соответственно.

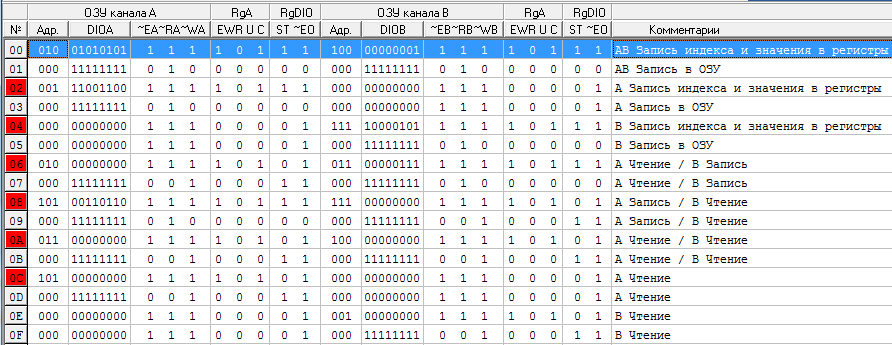
Если запись осуществляется в одну ячейку памяти, то результатом будет конъюнкция данных, находящихся во входных регистрах.

Чтение:

Для чтения данных из памяти двухпортового ЗУ необходимо:

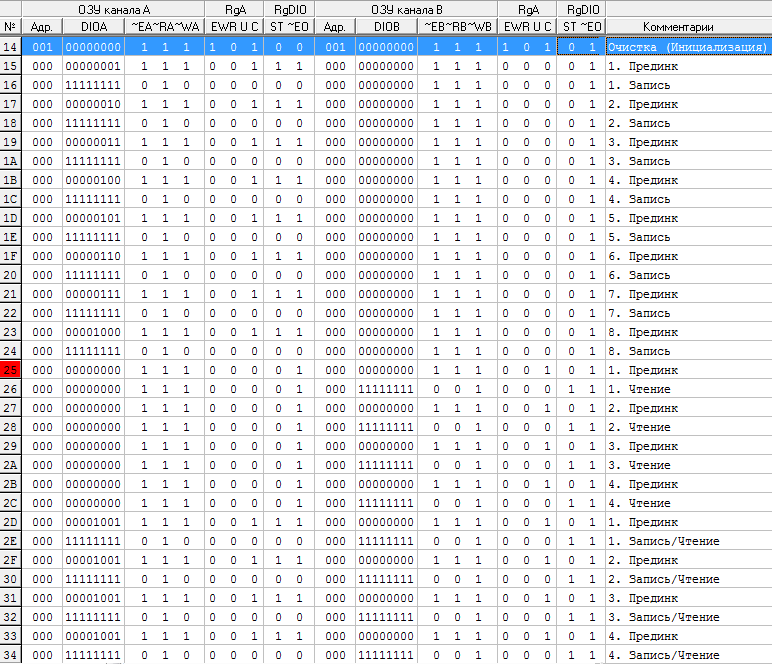
1. В регистр адреса канала А (RgAA) и/или канала В (RgAB) записать адрес ячейки памяти ЗУ, из которого будет производиться чтение, при помощи подачи управляющих сигналов EWRA = 1 и/или EWRB = 1 соответственно
2. Выполнить чтение данных из ЗУ по адресу RgAA и/или RgAB в регистр RgDIOA и/или RgDIOB соответственно, при помощи подачи управляющих сигналов ~RA = 0, ~EA = 0 и/или ~RB = 0, ~EB = 0 соответственно.

Подмикропрограмма исследования двухпортового ЗУ в режиме чтения, записи представлена на рисунке 33.



*Рисунок 33*

Подмикропрограмма исследования двухпортового ЗУ в режиме стека типа FIFO представлена на рисунке 34, граф-схема алгоритма приведена на рисунке 35.



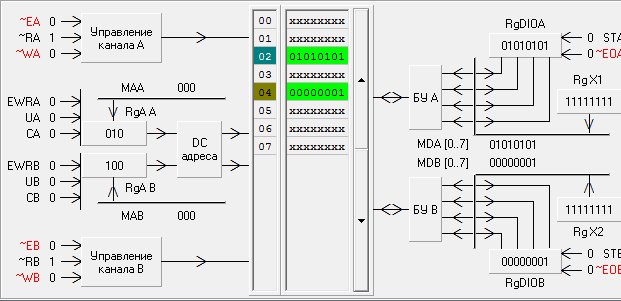
*Рисунок 34*



*Рисунок 35*

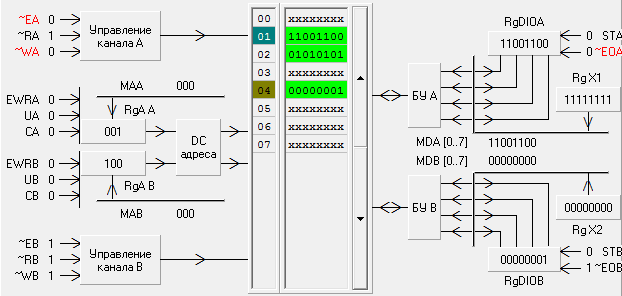
Выполнение последовательности операций:

* запись одновременно по порту А и В

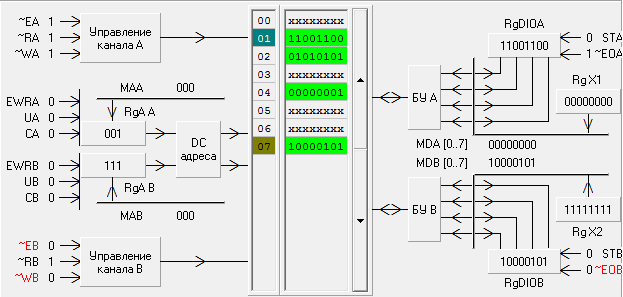


*Рисунок 36*

* раздельная запись по одному из портов А и В

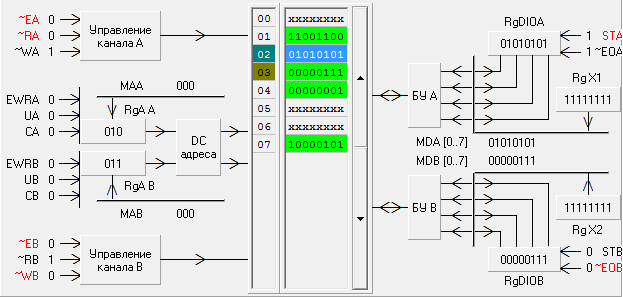


*Рисунок 37*



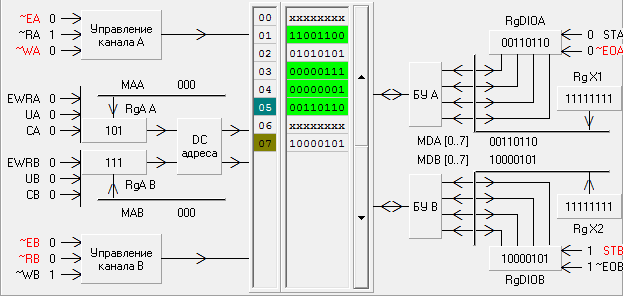
*Рисунок 38*

* порт А чтение, порт В запись



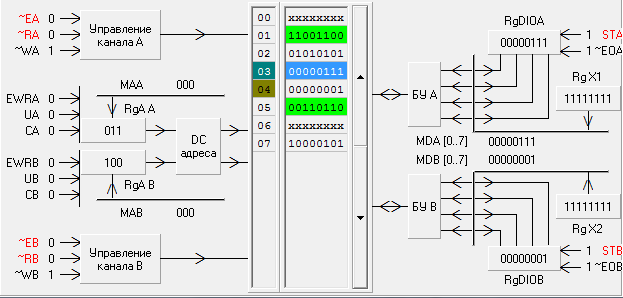
*Рисунок 39*

* порт В чтение, порт А запись



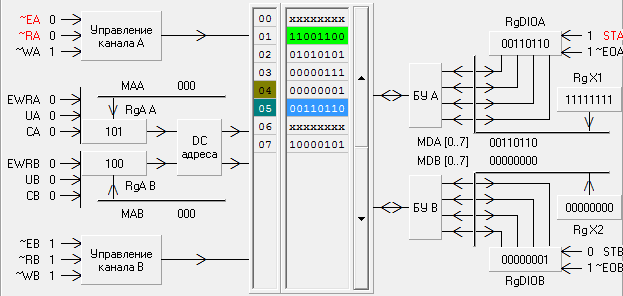
*Рисунок 40*

* порт В чтение, порт А чтение

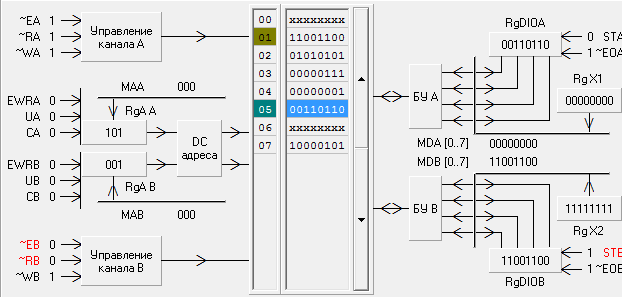


*Рисунок 41*

* раздельное чтение по порту А или В

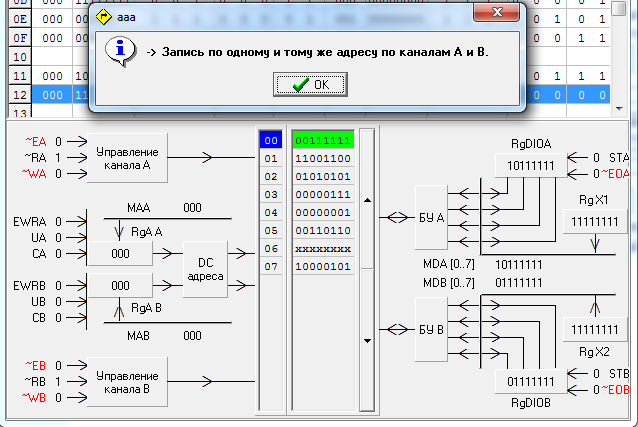


*Рисунок 42*



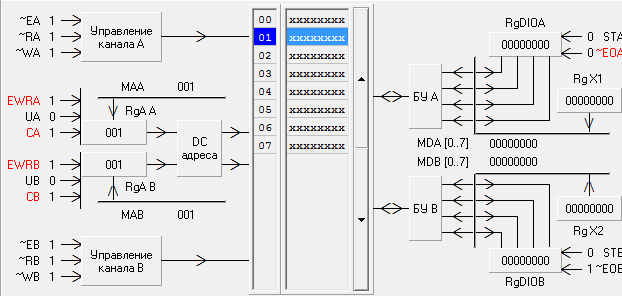
*Рисунок 43*

* запись по портам А и В в одну и ту же ячейку



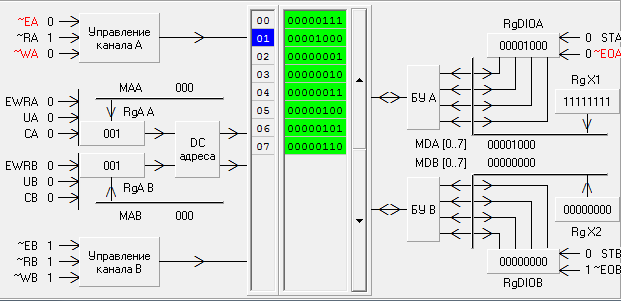
*Рисунок 44*

* сброс очереди (инициализация)



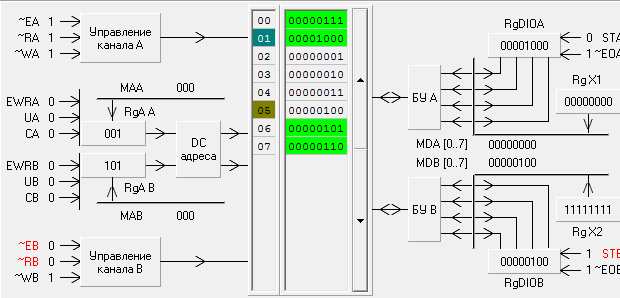
*Рисунок 45*

* запись 8-и чисел в очередь



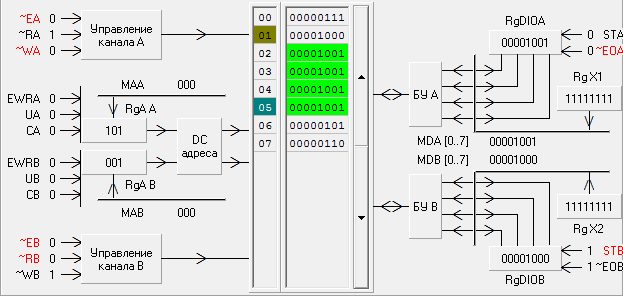
*Рисунок 46*

* чтение 4-х чисел из очереди



*Рисунок 47*

* запись 4-х чисел с параллельным считыванием из очереди



*Рисунок 48*

**Вывод**

В ходе выполнения лабораторной работы были изучены принципы работы стеков типа LIFO и FIFO, ассоциативного запоминающего устройства и двухпортового запоминающего устройства.

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.

Ассоциативное запоминающее устройство позволяет осуществлять поиск в памяти по тегам. Для этого в маске ставятся единицы в те биты, которые необходимо сравнить, а в эталонном значении указывается, то, что должно находиться в соответствующих замаскированных битах. Если после выполнения поиска найдено одно значение, то производится вывод этого значения. Если возникает ситуация нахождения нескольких значений, то выдаются все найденные значения поочередно.

Двухпортовое запоминающее устройство позволяет производить операции по двум портам одновременно (запись и запись, чтение и чтение, запись и чтение, чтение и запись). При наличии двух портов можно взаимодействовать с двумя ячейками памяти за один такт, а не за два как в случае однопостовых запоминающих устройств. Это позволяет ускорить обращение к запоминающему устройству и распараллеливать некоторые операции. Если будет произведена запись в одну и ту же ячейку памяти с двух портов одновременно, то результатом будет конъюнкция записываемых данных с этих портов.