МИНИИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2

по дисциплине

«Организация памяти ЭВМ»

Вариант 4

Разработал студент группы ИВТб-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Козюков М.Н. /

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

Киров 2020

**Задание на лабораторную работу**

Исследовать работу Ассоциативного ЗУ. Необходимо 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 для различных значений эталонов в RgI и RgM.

Исследовать работу Двухпортового ЗУ. Необходимо разработать подмикропрограмму, выполняющую следующие функции:

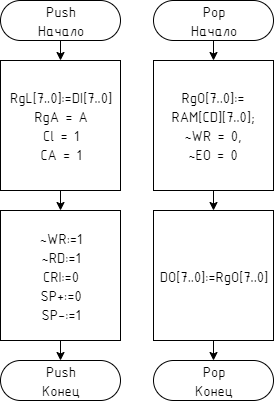
1. Исследовать ОЗУ в режиме произвольного доступа при записи и чтении
   1. Выполнить запись данных во все ячейки ОЗУ в режимах:
      1. Записи одновременно по порту А и В
      2. Раздельной записи по одному из портов А и В
   2. Выполнить сочетание процедур чтения и записи одновременно по портам А и В:
      1. Порт А чтение, порт В запись
      2. Порт В чтение, порт А запись
      3. Порт В чтение, порт А чтение
   3. Выполнить попытку записи по портам А и В в одну и ту же ячейку и сделать выводы
2. На основе ОЗУ организовать стек типа FIFO для очереди команд с возможностью параллельного пополненния очереди команд через каждые 4 считанные из очереди команды:
   1. Запись 8-х чисел
   2. Чтение 4-х чисел
   3. Запись 4-х чисел с параллельным считыванием из очереди
   4. Запись 4-х чисел с параллельным считыванием из очереди
   5. Сброс очереди команд

**Выполнение лабораторной работы**

**Задание 1**

Запись числа: Во входной регистр RgDI записать данные с входной шины при помощи подачи управляющего сигнала CI. В регистр адресов RgA записать адрес, куда будет записываться значение в память с регистра RgDI, при помощи подачи управляющего сигнала СА. Записать данные из регистра RgDI в ячейку памяти по адресу RgA.

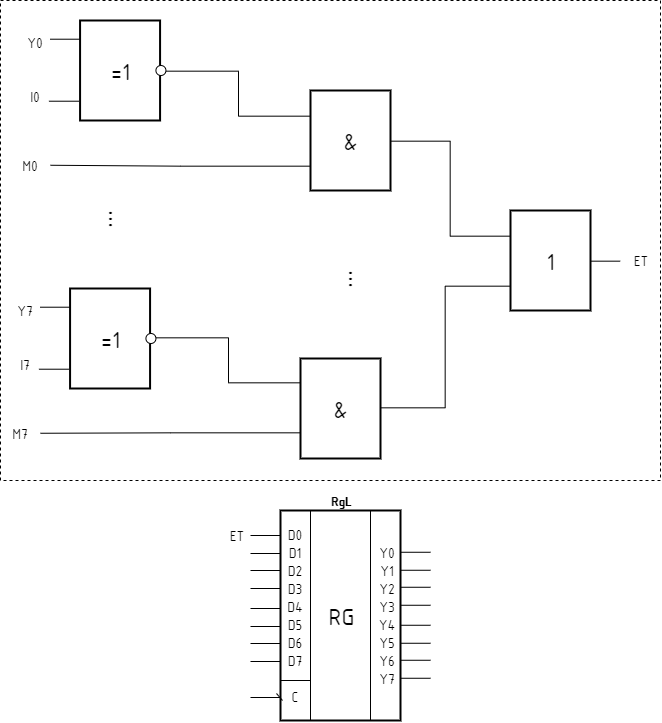
Чтение числа: Подать сигнал ~RD = 0, тем самым данные, находящиеся в ячейке памяти по адресу CD, формируемого из регистра RgSH, запишутся в выходной регистр RgDO Подать сигнал ~E0 = 0, тем самым выдав содержимое регистра RgDO на выходную шину данных.



*Рисунок 1 – ГСА операции записи и чтения для AZU*

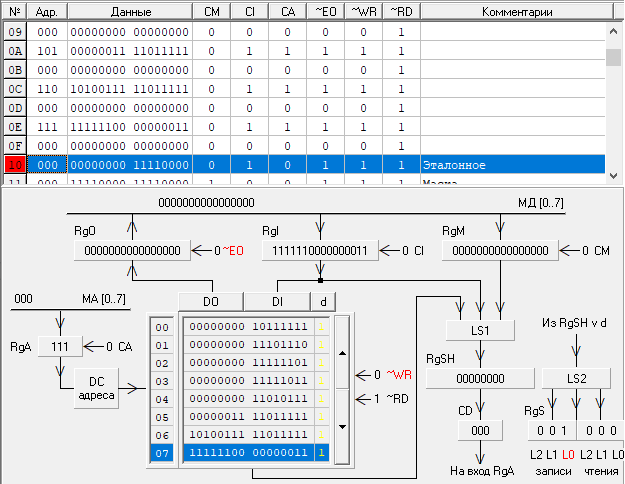
*Таблица 1 – Текст микропрограммы*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *№* | *Адр.* | *Данные* | *CM* | *CI* | *CA* | *~EO* | *~WR* | *~RD* |
| *00* | *000* | *00000000 11111111* | *0* | *1* | *1* | *1* | *1* | *1* |
| *01* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *0* | *1* |
| *02* | *001* | *00000000 11111110* | *0* | *1* | *1* | *1* | *1* | *1* |
| *03* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *0* | *1* |
| *04* | *010* | *00000000 11111101* | *0* | *1* | *1* | *1* | *1* | *1* |
| *05* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *0* | *1* |
| *06* | *011* | *00000000 11111011* | *0* | *1* | *1* | *1* | *1* | *1* |
| *07* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *0* | *1* |
| *08* | *100* | *00000000 11110111* | *0* | *1* | *1* | *1* | *1* | *1* |
| *09* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *0* | *1* |
| *0A* | *000* | *00000000 11110000* | *0* | *1* | *0* | *1* | *1* | *1* |
| *0B* | *000* | *11110000 11110000* | *1* | *0* | *0* | *1* | *1* | *1* |
| *0C* | *000* | *00000000 00000000* | *0* | *0* | *0* | *1* | *1* | *0* |
| *0D* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *1* | *0* |
| *0E* | *000* | *00000000 00000000* | *0* | *0* | *0* | *0* | *1* | *0* |

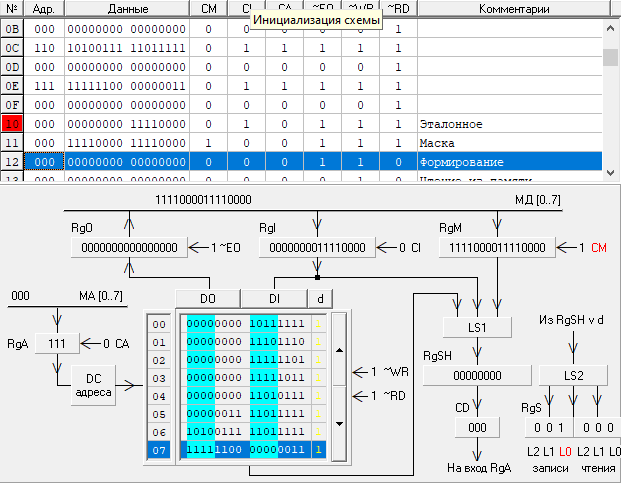
**

*Рисунок 2 – Функциональная схема лонической схемы LS1*

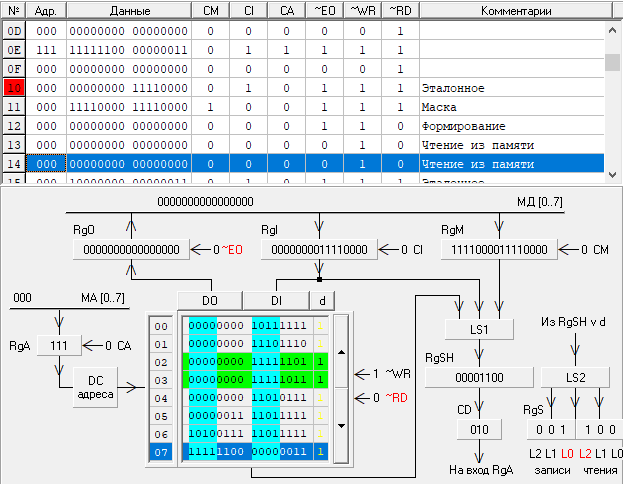
**Задание 2**



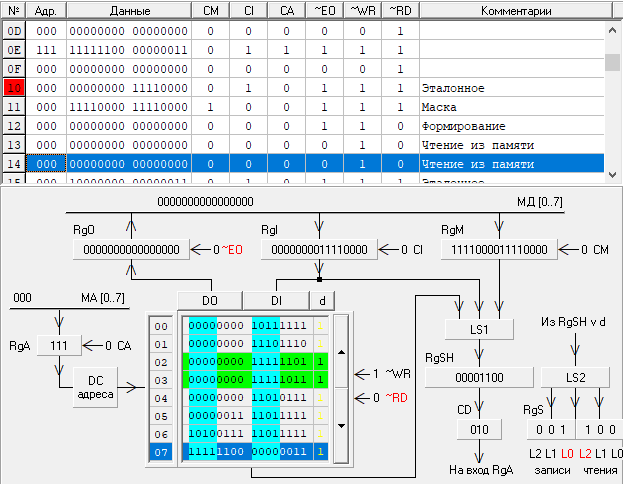
*Рисунок 3 – Запись 8-ми чисел*



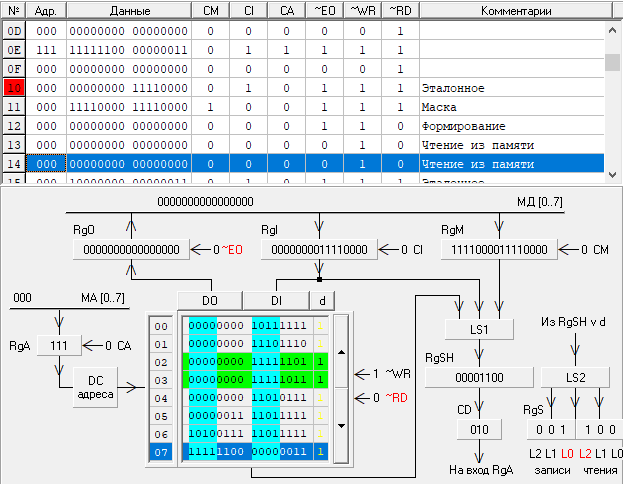
*Рисунок 4 – Запись компаранта(RgI) и маски(RgM)*



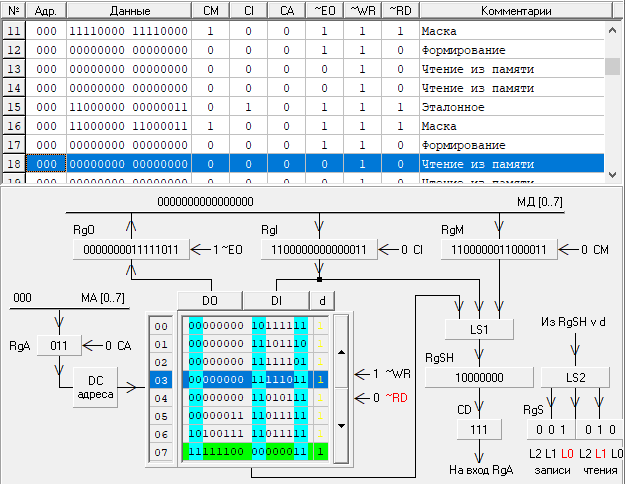
*Рисунок 5 – Формирование LS2*



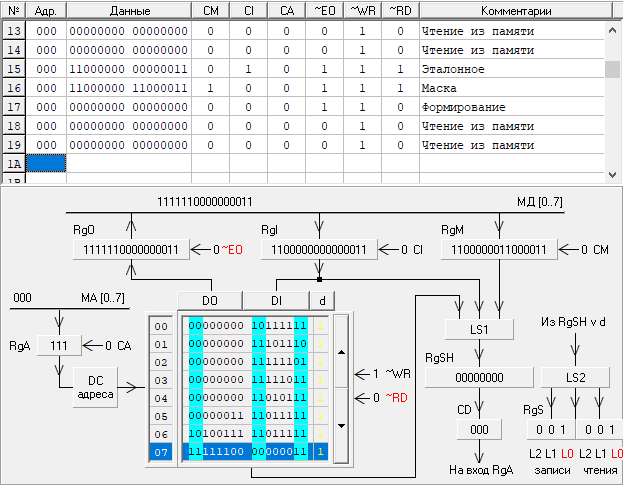
*Рисунок 6 – Считывание совпавших ячеек*



*Рисунок 7 – Запись компаранта(RgI) и маски(RgM)*



*Рисунок 8 – Формирование LS2*

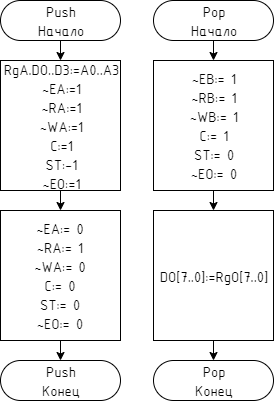


*Рисунок 9 – Считывание совпавших ячеек*

**Задание 3**

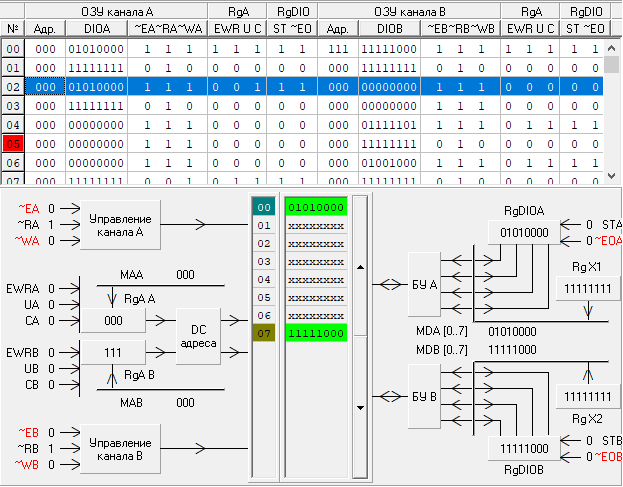
Запись числа: ОЗУ канала нужно перевести в Z-состояние(режим хранения) при помощи подачи единиц на входы ~EA, ~RA, ~WA. Во входной регистр RgDIOX записать данные с входной шины при помощи подачи единицы на входы ST и ~EO. В регистр адресов RgAX записать адрес, куда будет записываться значение в память с регистра RgDIOX, при помощи подачи единицы на входы EWR, С. Записать данные из регистра RgDIOX в ячейку памяти по адресу RgAX при помощи подачи ноля на входы ~EA и ~WA и единицы на вход ~RA. Регистр/счётчик адреса находится в режиме состояния.

Чтение числа: ОЗУ канала нужно перевести в Z – состояние ( режим хранения) при помощи подачи единиц на входы: ~EA, ~RA, ~WA. В регистр адресов RgAX записать адрес, откуда будет считываться значение из памяти в регистр RgDIOX, при помощи подачи единицы на входы: EWR, C. Подать нули на входы ~EA, ~RA и единицу на вход ~WA, тем самым выдав содержимое на выходную шину данных и сохранить операнд в RgDIOX

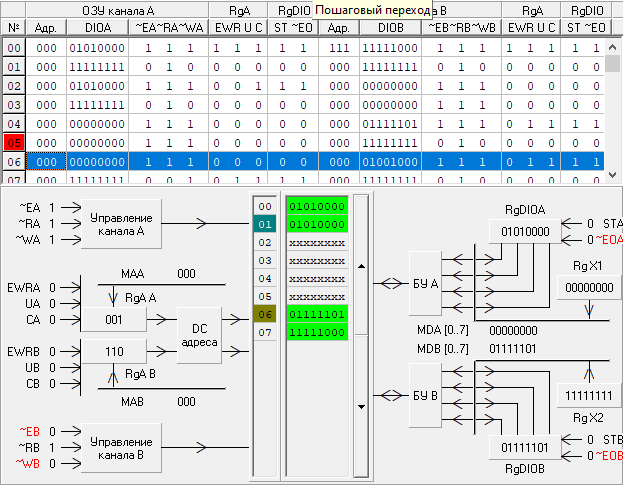


*Рисунок 10 – ГСА операции записи и чтения для двухпортового запоминающего устройства*

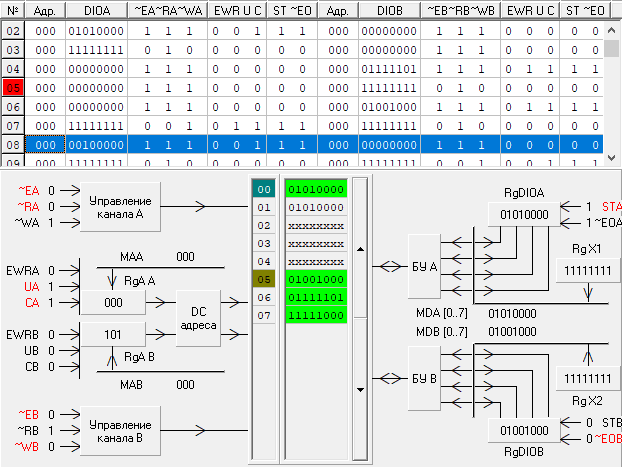
**Задание 4**



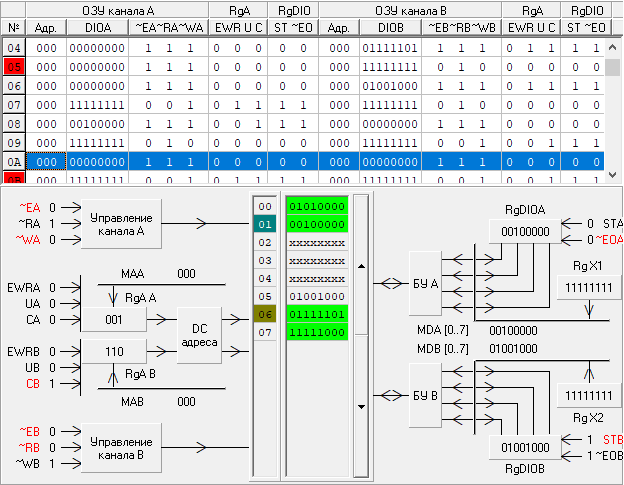
*Рисунок 11 – Запись одновременно по порту А и В*



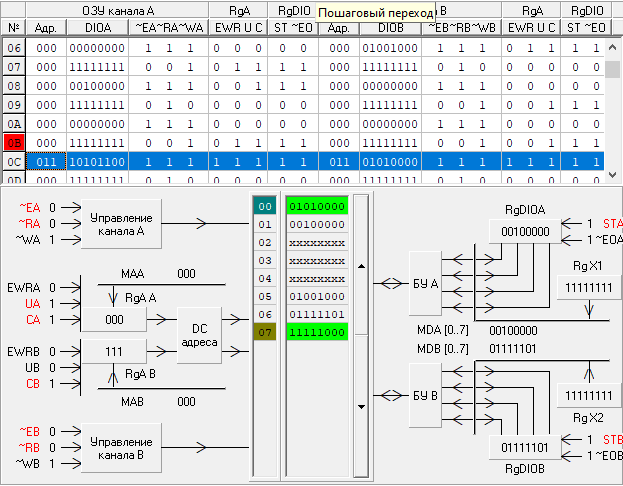
*Рисунок 12 – Раздельная запись по одному из портов А и В*



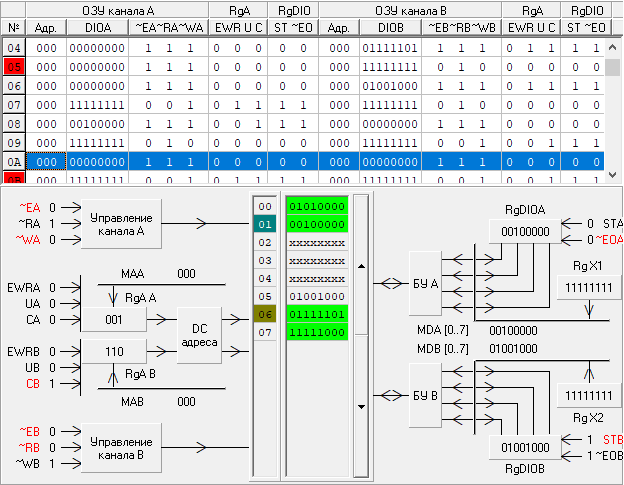
*Рисунок 13 – Порт А чтение, порт В запись*



*Рисунок 14 – Порт А запись, порт В чтение*

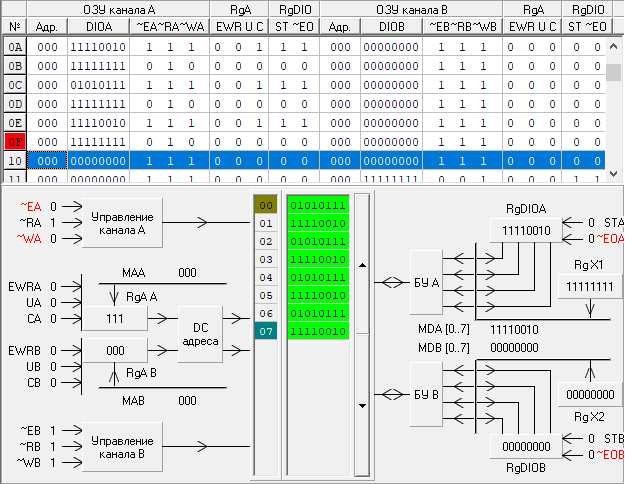


*Рисунок 15 – Порт А чтение, порт В чтение*

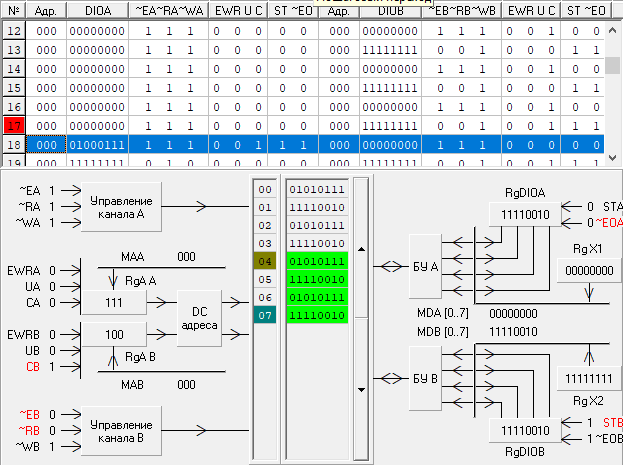


*Рисунок 16 – Попытка записи по портам А и В в одну и ту же ячейку*

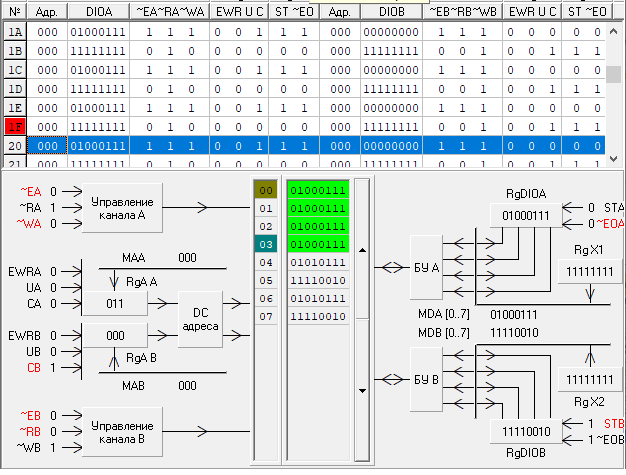
**Задание 5**



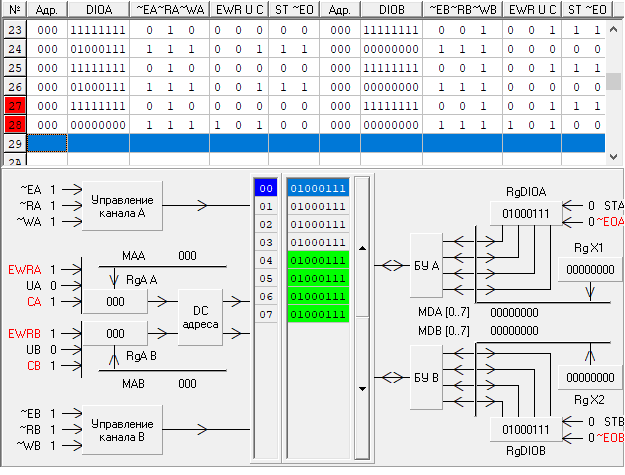
*Рисунок 17 – Запись 8-ми чисел*



*Рисунок 18 – Чтение 4-х чисел*



*Рисунок 19 – Запись 4-х чисел с параллельным считыванием из очереди*



*Рисунок 20 – Сброс очереди команд(команда БП)*

**Вывод**

В ходе выполнения лабораторной работы была изучена структура ассоциативного запоминающего устройства. Была разработана программа, которая позволяет находить в АЗУ данные по заданным маскам и компарантам. Данная операция была реализована при помощи регистра запросов, регистра маски и логической схемы на основе «исключающего или», «или», «и». Также изучили структуру двухпортового запоминающего устройства. На базе двухпортового запоминающего устройства были реализованы так же программы записи и чтения для FIFO/