МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1

по дисциплине «Организация памяти ЭВМ»

Вариант 10

Выполнил студент группы ИВТ-32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Куваев А.С./

Киров 2017

1. Задание на лабораторную работу

Целью

1. Исследовать работу стеков типа LIFO и FIFO в режимах загрузки и чтения стека.

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

3. На основе двухпортового ОЗУ организовать стек типа FIFO для очереди команд с возможностью параллельного пополнения очереди команд через каждые 4 считанные из очереди команды.

1. LIFO и FIFO

Необходимо разработать микропрограмму для пред-дектементного LIFO, начинающегося по адресу 0D и имеющего размерность 8. Необходимо разработать микропрограмму для пост-инкрементного FIFO, начинающегося по адресу 09.

Функциональная схема LIFO представленная на рисунке 1. Функциональная схема FIFO представленная на рисунке 2.

ГСА операций Push и Pop для LIFO представлена на рисунке 3, а для LIFO – на рисунке 4.

Текст микропрограммы представлен на рисунке 5.



Рисунок 1 – Функциональная схема LIFO



Рисунок 2 – Функциональная схема FIFO



Рисунок 3 – ГСА операций Push и Pop для LIFO



Рисунок 4 – ГСА операций Push и Pop для FIFO

|  |
| --- |
| ┌──┬──────┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬────────────────────┐  │№ │Данные│~WR│~RD│CRI│SP+│SP-│~WR│~RD│CRI│WR+│RD+│ Комментарии │  ├──┼──────┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼────────────────────┤  │00│ FF │ 1 │ 1 │ 1 │ 0 │ 1 │ 1 │ 1 │ 1 │ 0 │ 0 │Записываем в регистр│  │01│ 00 │ 0 │ 1 │ 1 │ 0 │ 0 │ 0 │ 1 │ 0 │ 1 │ 0 │В стек │  │02│ │ │ │ │ │ │ │ │ │ │ │ │  │03│ │ │ │ │ │ │ │ │ │ │ │ │  │04│ 00 │ 1 │ 0 │ 0 │ 0 │ 0 │ 1 │ 0 │ 0 │ 0 │ 0 │Читаем │  │05│ 00 │ 1 │ 1 │ 0 │ 1 │ 0 │ 1 │ 1 │ 0 │ 0 │ 1 │Очищаем │  │06│ │ │ │ │ │ │ │ │ │ │ │ │  │07│ │ │ │ │ │ │ │ │ │ │ │ │  └──┴──────┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴────────────────────┘ |

Рисунок 5 – текст микропрограммы

1. Двух-портовое ОЗУ

Функциональная схема двухпортового ОЗУ представлена на рисунке 6. ГСА операций push и pop для стека LIFO построенного на основе ОЗУ представлены на рисунке 7. Код программы представлен на рисунке 8

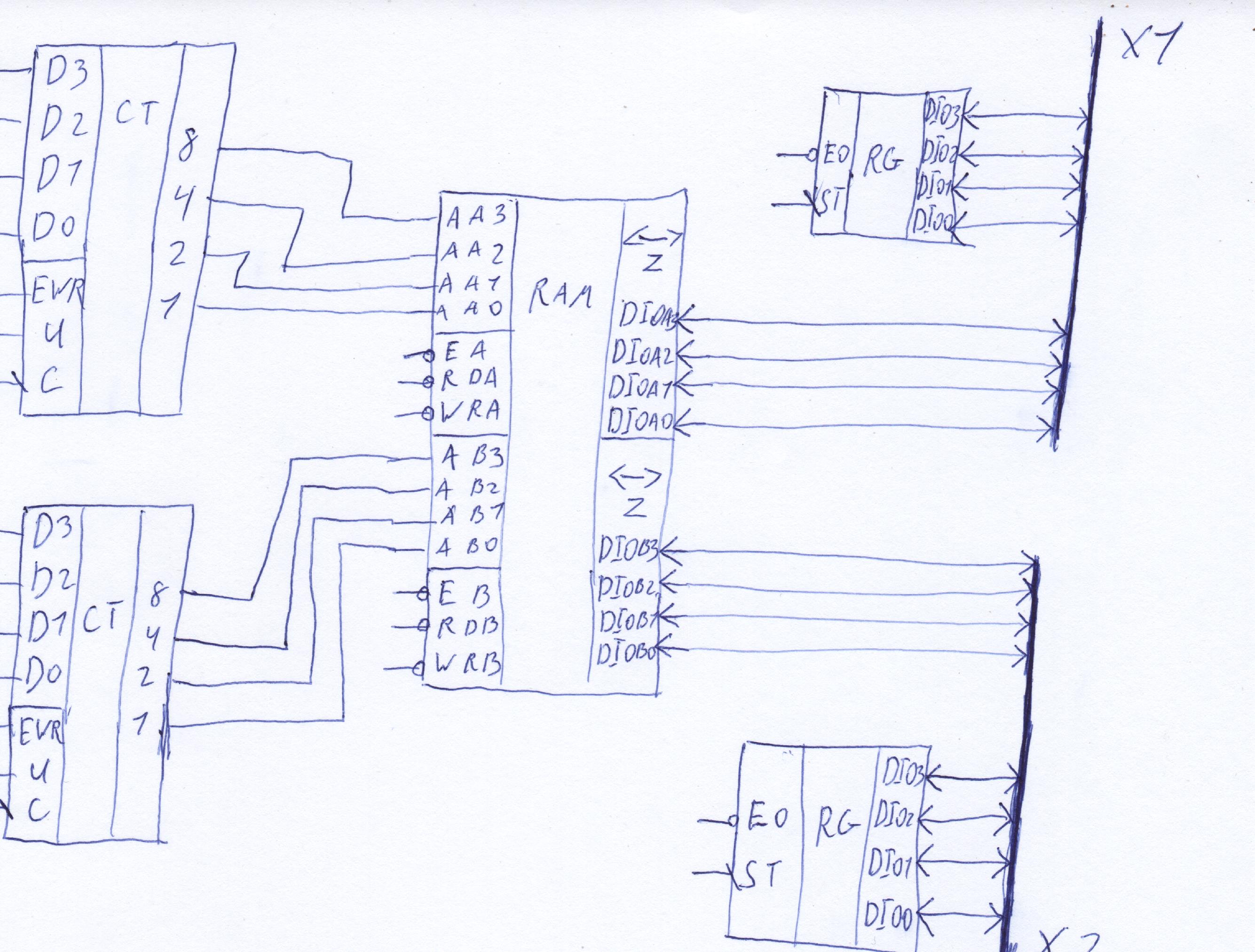


Рисунок 6 – функциональная схема двухпортовой ОЗУ



Рисунок 7 – ГСА операций Push и pop

|  |
| --- |
| ┌──┬────┬────┬─────────┬───────┬─────┬────┬────┬─────────┬───────┬─────┬───────────┐  │№ │Адр.│DIOA│~EA~RA~WA│EWR U C│ST~EO│Адр.│DIOB│~EB~RB~WB│EWR U C│ST~EO│Комментарии│  ├──┼────┼────┼─────────┼───────┼─────┼────┼────┼─────────┼───────┼─────┼───────────┤  │00│ │ │ │ │ │ │ │ │ │ │ │  │01│ │ │ │ │ │ │ │ │ │ │ │  │02│ 00 │ 07 │ 1 1 1 │ 0 0 0 │ 1 1 │ 01 │ 00 │ 1 1 1 │ 0 0 0 │ 1 1 │Готовим │  │03│ 00 │ FF │ 0 1 0 │ 0 0 1 │ 0 0 │ 01 │ FF │ 1 1 1 │ 0 0 0 │ 0 0 │Пишем │  │04│ │ │ │ │ │ │ │ │ │ │ │  │05│ 00 │ 00 │ 1 1 1 │ 0 0 0 │ 0 0 │ 00 │ FF │ 0 0 1 │ 0 0 1 │ 1 1 │Читаем │  │06│ │ │ │ │ │ │ │ │ │ │ │  │07│ 00 │ 02 │ 1 1 1 │ 0 0 0 │ 1 1 │ 00 │ 00 │ 1 1 1 │ 0 0 0 │ 1 1 │ │  │08│ 00 │ FF │ 0 1 0 │ 0 0 1 │ 0 0 │ 00 │ FF │ 0 0 1 │ 0 0 1 │ 1 1 │WR and R │  │09│ │ │ │ │ │ │ │ │ │ │ │  │0A│ 00 │ 00 │ 1 1 1 │ 1 0 1 │ 0 0 │ 00 │ 00 │ 1 1 1 │ 1 0 1 │ 0 0 │Сброс │  └──┴────┴────┴─────────┴───────┴─────┴────┴────┴─────────┴───────┴─────┴───────────┘ |

Рисунок 8 – Текст программы

1. Ассоциативное запоминающее устройство

Функциональная схема LS1 представлена на рисунке 9. Код программы представлен на рисунке 10.

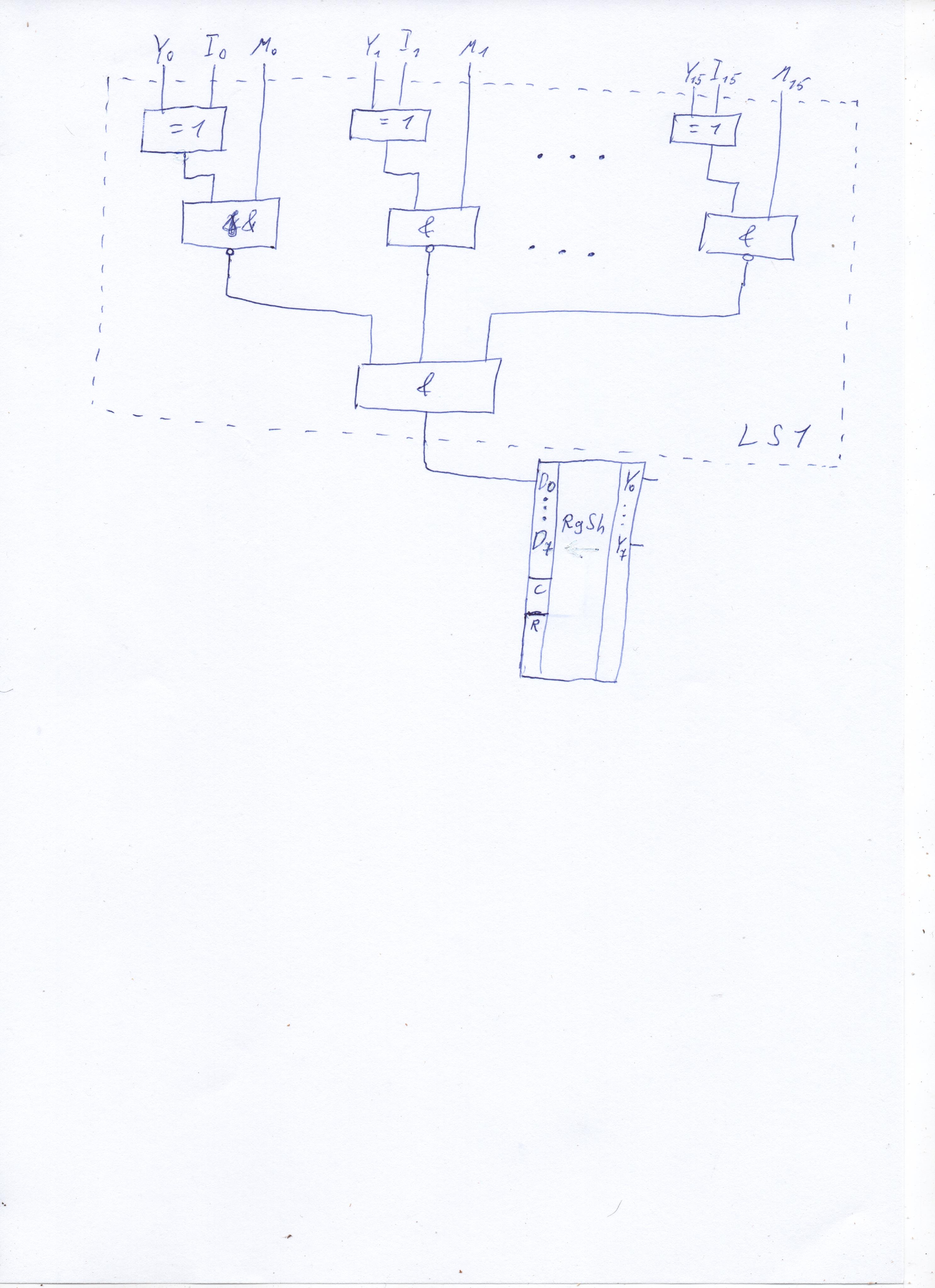


Рисунок 9 – функциональная схема LS1

|  |
| --- |
| ┌──┬────┬──────┬──┬──┬──┬───┬───┬───┬────────────────────────────┐  │№ │Адр.│Данные│CM│CI│CA│~EO│~WR│~RD│ Комментарии │  ├──┼────┼──────┼──┼──┼──┼───┼───┼───┼────────────────────────────┤  │00│000 │ 0001 │0 │1 │1 │ 1 │ 0 │ 1 │Подготовка к записи │  │01│001 │ 0002 │0 │1 │1 │ 1 │ 0 │ 1 │Запись │  │02│010 │ 0003 │0 │1 │1 │ 1 │ 0 │ 1 │Запись │  │03│011 │ 0004 │0 │1 │1 │ 1 │ 0 │ 1 │Запись │  │04│100 │ 0005 │0 │1 │1 │ 1 │ 0 │ 1 │Запись │  │05│101 │ 0006 │0 │1 │1 │ 1 │ 0 │ 1 │ │  │06│110 │ 0007 │0 │1 │1 │ 1 │ 0 │ 1 │ │  │07│000 │ 0007 │0 │1 │0 │ 1 │ 1 │ 1 │ │  │08│000 │ FFFF │1 │0 │0 │ 1 │ 1 │ 0 │Запрос и маска │  │09│000 │ 0000 │0 │0 │0 │ 0 │ 1 │ 0 │Чтение │  │0A│000 │ 0000 │0 │0 │0 │ 0 │ 1 │ 0 │Чтение │  │0B│000 │ 0000 │0 │0 │0 │ 0 │ 1 │ 0 │Чтение │  │0C│000 │ 0000 │0 │0 │0 │ 0 │ 1 │ 0 │Чтение │  └──┴────┴──────┴──┴──┴──┴───┴───┴───┴────────────────────────────┘ |

Рисунок 10 – код программы

1. Вывод

В ходе выполнения лабораторной работы были изучены структуры LIFO и FIFO, двухпортовая ОЗУ и ассоциативное запоминающее устройство.

Были реализованы подпрограммы push и pop для LIFO с пред-декрементом и для FIFO с пост-инкрементом. Основная особенность пред-декремента заключается в том, что перед записью в ОЗУ производится декремент указателя SP, а в операции чтения инкремент производится только после считывания данных. В пост-инкременте FIFO запись производится наоборот: сначала записываются данные по указателю SP, а затем производится инкремент. Операция чтения полностью совпадает с LIFO.

На базе двухпортового ОЗУ были реализованы так же подпрограммы push и pop для FIFO. В ходе экспериментов с портами А и В было выяснено, что операции чтения и записи по порту А выполняются быстрее, чем по порту В. На основании этого была разработана подпрограмма параллельной записи и чтения, в которой запись производилась по порту А, а чтение по порту В. Таким образом, если FIFO на момент параллельного чтения и записи был пустой, то данные, которые считает порт В, во время параллельной операции, будут достоверными.

В ходе изучения ассоциативного устройства, была разработана программа, которая позволяет находить в АЗУ четные числа. Данная операция была реализована при помощи регистра запросов, регистра маски и логической схемы на основе «исключающего или» и «и».