

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 4

Выполнил студент группы ИВТ-32 _____/Рзаев А. Э./
Проверил преподаватель _____/Куваев А.С./

Киров 2018

1 Задание на лабораторную работу

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.

2 LIFO и FIFO

Необходимо разработать микропрограмму для прединкрементного LIFO, начинающегося по адресу 0x04 и имеющего размерность 0x08. Необходимо разработать микропрограмму для прединкрементного FIFO, начинающегося по адресу 0x03.

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 FIFO – на рисунке 4.

Текст микропрограммы представлен на рисунке 5.

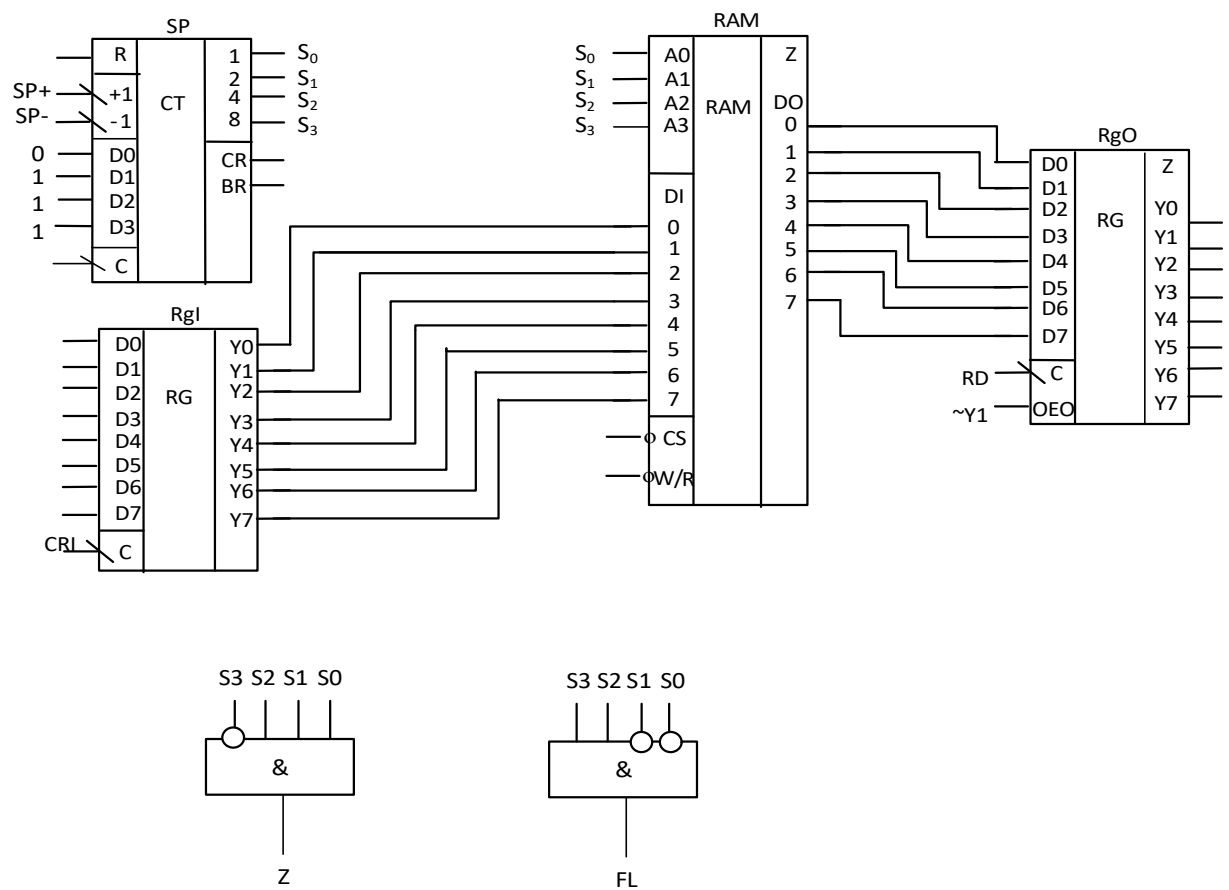


Рисунок 1 – Функциональная схема LIFO

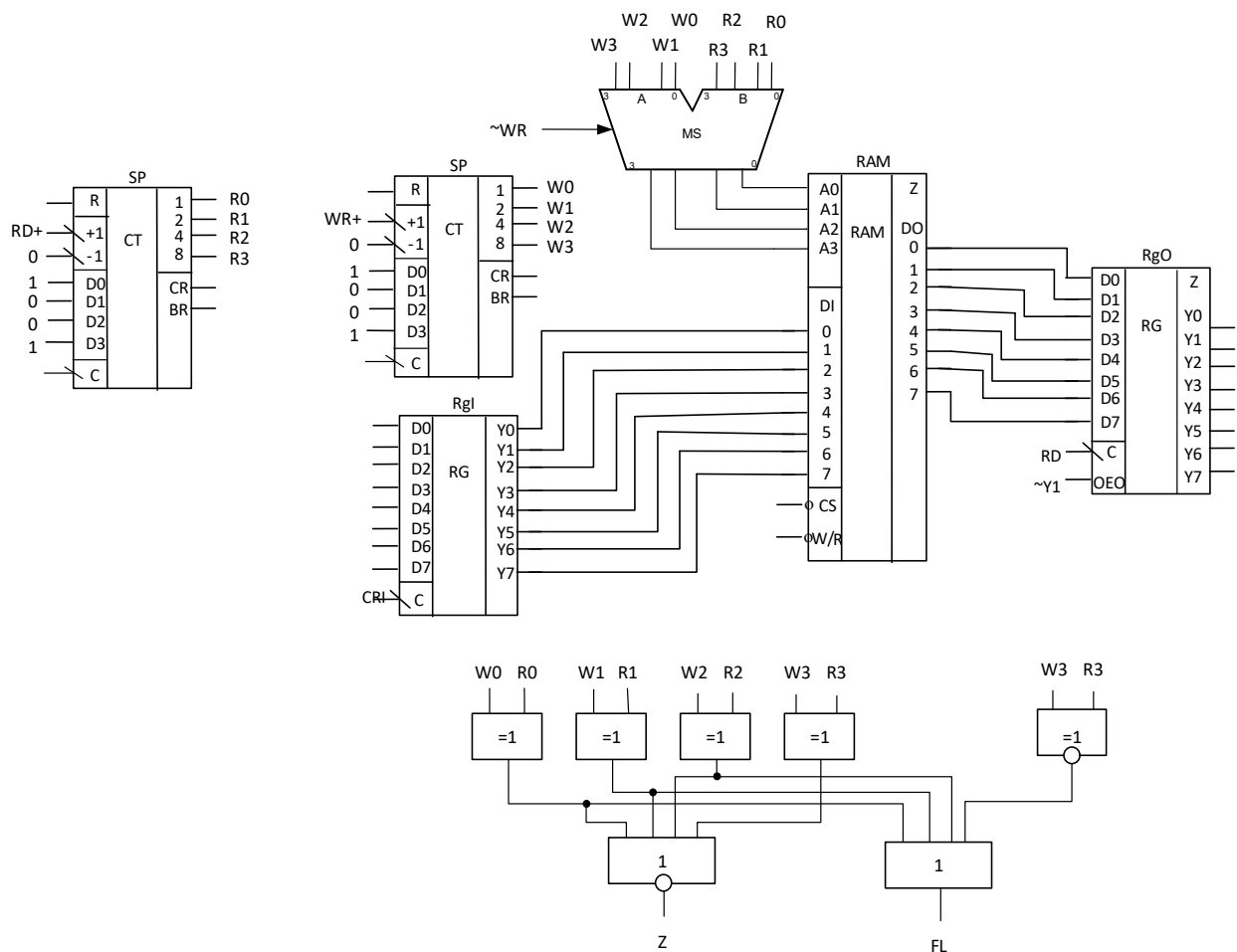


Рисунок 2 – Функциональная схема FIFO

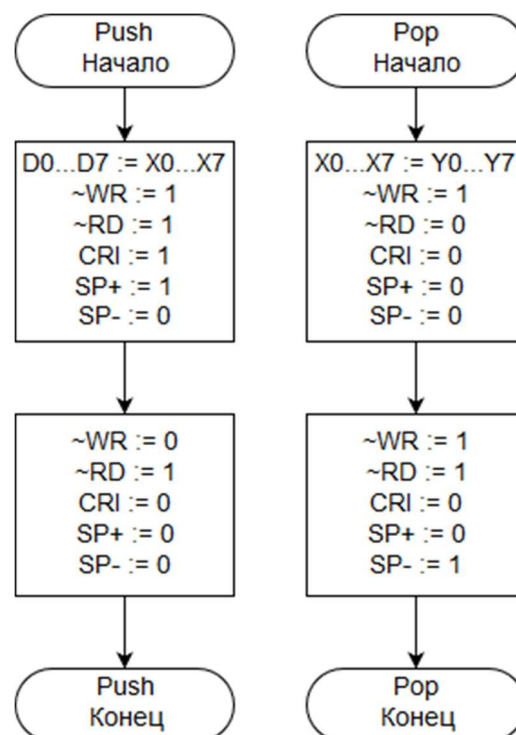


Рисунок 3 – ГСА операций Push и Pop для LIFO

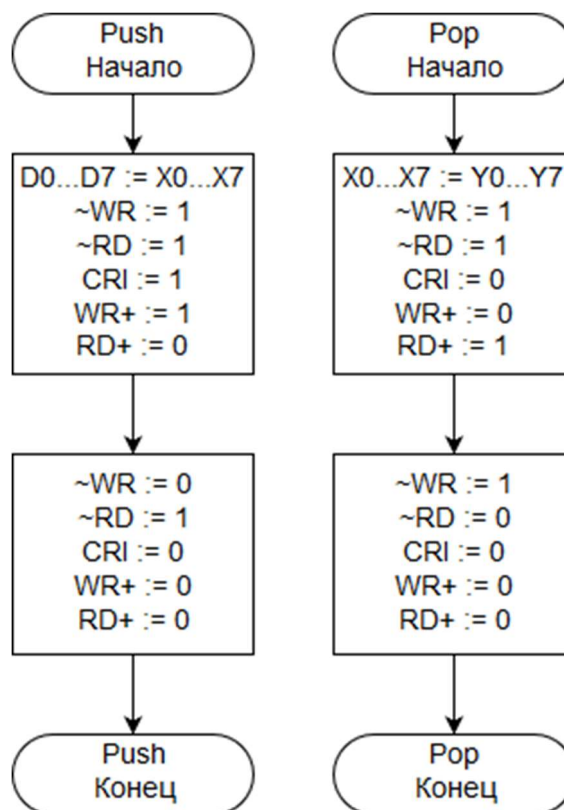


Рисунок 4 – ГСА операций Push и Pop для FIFO

№	Данные	~WR	~RD	CRI	SP+	SP-	~WR	~RD	CRI	WR+	RD+	Комментарии
00	00001111	1	1	1	1	0	1	1	1	1	0	RgDI=D; SP++; PUSH
01	00000000	0	1	0	0	0	0	1	0	0	0	RAM[A]=RgDI
02	00000000	1	0	0	0	0	1	1	0	0	1	RgDO=RAM[A]; POP
03	00000000	1	1	0	0	1	1	0	0	0	0	SP--

Рисунок 5 – Текст микропрограммы

3 Двухпортовое ОЗУ

Функциональная схема двухпортового ОЗУ представлена на рисунке 6. ГСА операций Push и Pop для стека FIFO построенного на основе ОЗУ представлены на рисунке 7. Текст микропрограммы представлен на рисунке 8

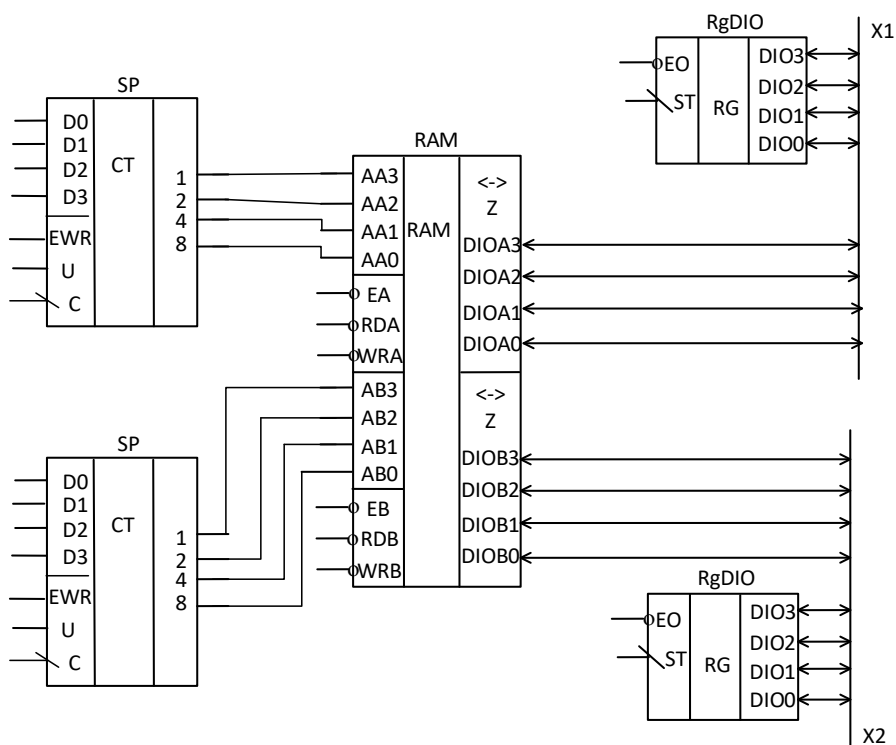


Рисунок 6 – Функциональная схема двухпортовой ОЗУ

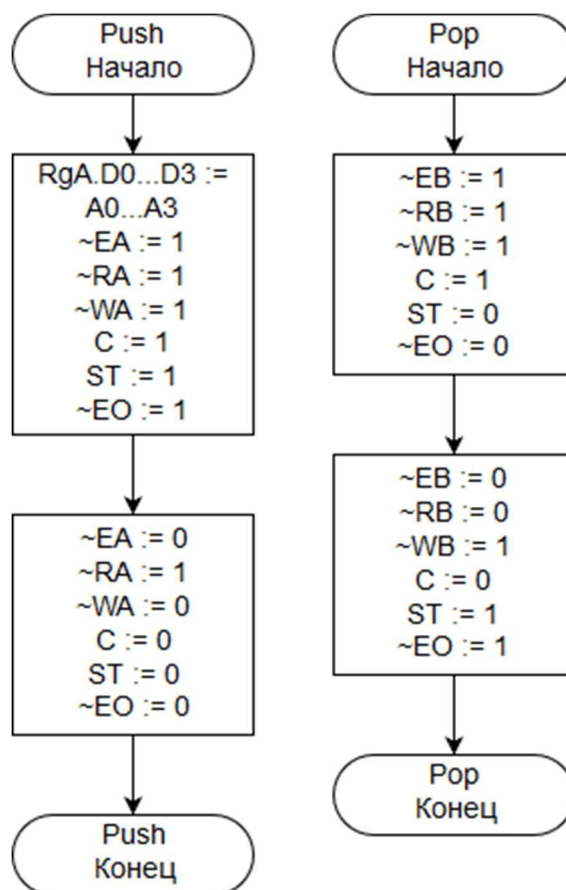


Рисунок 7 – ГСА операций Push и Pop

№	Адр.	DIOA	~EA~RA~WA	EWR U C	ST ~EO	Адр.	DIOB	~EB~RB~WB	EWR U C	ST ~EO	Комментарии
00	0000	0000	1 1 1	1 0 1	0 0	0000	0000	1 1 1	1 0 1	0 0	RgA:=0;RgB:=0 PUSH POP
01	0000	0000	1 1 1	0 0 1	1 1	0000	0000	1 1 1	0 0 0	0 0	
02	0000	1111	0 1 0	0 0 0	0 0	0000	0000	1 1 1	0 0 0	0 0	
11	0000	0000	1 1 1	0 0 0	0 0	0000	0000	1 1 1	0 0 1	0 0	
12	0000	0000	1 1 1	0 0 0	0 0	0000	1111	0 0 1	0 0 0	1 1	

Рисунок 8 – Текст микропрограммы

4 Ассоциативное запоминающее устройство

Функциональная схема LS1 представлена на рисунке 9. Текст микропрограммы представлен на рисунке 10.

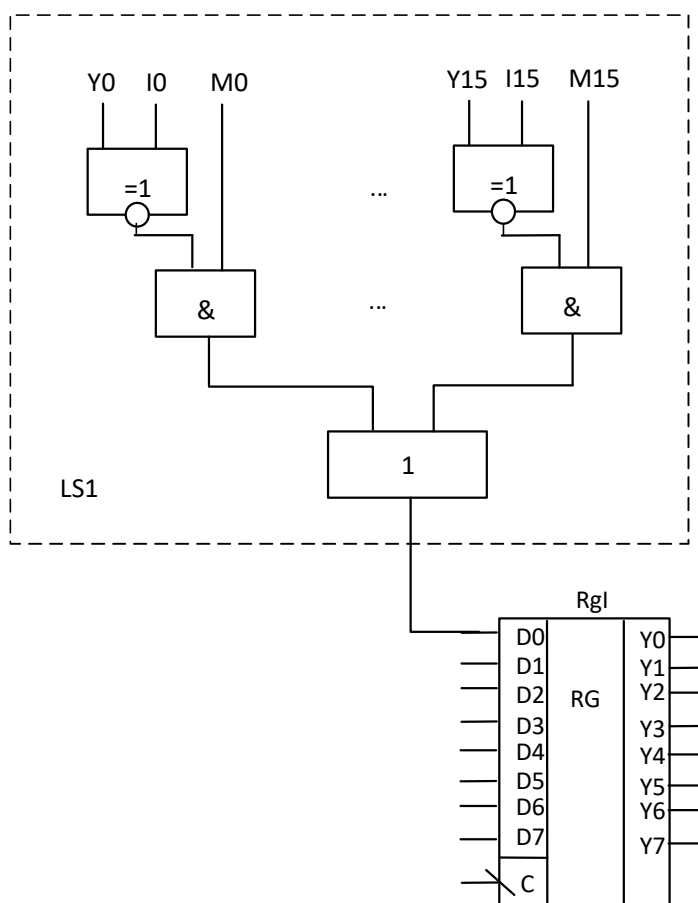


Рисунок 9 – Функциональная схема LS1

№	Адр.	Данные	CM	CI	CA	~EO	~WR	~RD	Комментарии
00	0000	00000000 11111111	0	1	1	1	1	1	M[0000]:=D
01	0000	00000000 00000000	0	0	0	0	0	1	
02	0001	00000000 11111110	0	1	1	1	1	1	M[0010]:=D
03	0000	00000000 00000000	0	0	0	0	0	1	
04	0010	00000000 11111101	0	1	1	1	1	1	M[0010]:=D
05	0000	00000000 00000000	0	0	0	0	0	1	
06	0011	00000000 11111011	0	1	1	1	1	1	M[0010]:=D
07	0000	00000000 00000000	0	0	0	0	0	1	
08	0100	00000000 11110111	0	1	1	1	1	1	M[0010]:=D
09	0000	00000000 00000000	0	0	0	0	0	1	
0A	0000	00000000 11110000	0	1	0	1	1	1	RgI:=D
0B	0000	11110000 11110000	1	0	0	1	1	1	RgM:=D
0C	0000	00000000 00000000	0	0	0	1	1	0	RgO:=M[RgA]

Рисунок 10 – Текст микропрограммы

5 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 и FIFO с прединкрементом. Основная особенность прединкремента заключается в том, что перед записью в ОЗУ производится инкремент указателя SP, а в операции чтения декремент производится только после считывания данных.

На базе двухпортового ОЗУ были реализованы так же подпрограммы push и pop для FIFO.

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