**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

**Отчет**

**По лабораторной работе №8**

По дисциплине: Аппаратные основы интеллектуальных систем

На тему: Моделирование ассоциативной памяти с системой адресации по разрядным столбцам и по словам.

Выполнил:                                   Щур А. А.

Группа 821703

Проверил:                                    Захаров В.В.

**Минск 2019**

**Цель работы:** освоение навыков построения и верификации модели ассоциативной памяти, обеспечивающей адресное считывание и запись по разрядным столбцам и по словам и выполнение логических операций над столбцами, поисковых операций и арифметических операций над полями слов.

**Вариант 28**

Построить и проверить программную модель ассоциативной памяти с диагональной

адресацией на основе сумматора. Размер массива памяти не более 16х16 двоичных

разрядов (битов).

Разработанная программная модель должна уметь выполнять операции считывания и

записи любых задаваемых разрядных столбцов и слов, а также выполнять поиск по

соответствию, и логические операции f5 и f10, f0 и f15 над разрядными столбцами и

арифметические операции над полями слов.

**Ход работы:**

В ходе написания лабораторной работы был реализован класс asoc\_memo, содержащий в

себе поля:

1) tbl – массив слов, хранимый в памяти

2) buff – буфер для временного хранения слов



Класс содержит 9 методов:

1) addition() – метод сложения двоичных слов, реализованный в лабораторной работе №1.

2) asoc\_memo() – конструктор класса. Инициализирует tbl 16-ю 16-разрядными

словами, каждый разряд которого – “ложь”.

3) print\_tbl() – вывод содержимого tbl на экран

4) add\_el() – добавление слова в tbl. Слова добавляются согласно диагональной

системе адресации.

5) get\_el() – возвращает по запросу номера строки нулевого разряда слова данное

слово

6) F5\_and\_F10() – выполнение логических функций F5&F10 над словами.

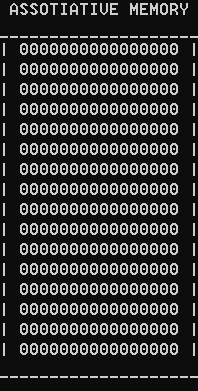
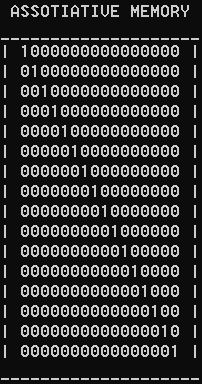
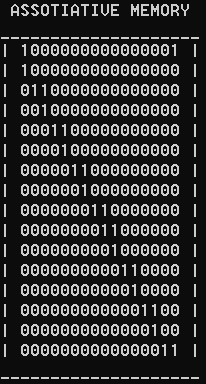
7) F0\_and\_F15() – выполнение логических функций F0& F15 над словами.

8) matching\_search() – поиск по соответствию

9) ariphmetics() – выполнение сложения полей Aj и Bj в словах Sj, у которых Vj

совпадает с заданным V= 000-111.

1. Добавление слов 1111111111111111 и 1010101010101010

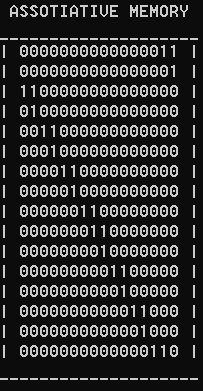


2) Вывод на экран слова, 0-ой символ которого содержится в 0-ом символе первой

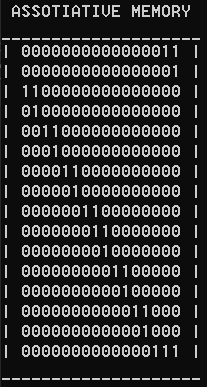
строки



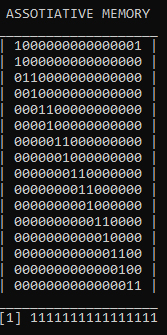
3) Выполнение функций F5&F10



4) Выполнение функций F0&F15



5) Поиск по соответствию (ключ – “111111111”)



6) Выполнение арифметических операций



**Выводы:**

Ассоциативная память может обеспечивать адресное считывание и запись по разрядным столбцам и по словам и выполнение логических операций над столбцами, поисковых операций и арифметических операций над полями слов.

Ввиду использования адресации только по разрядным столбцам ассоциативные процессоры имеют некоторые недостатки, такие как наличие буфера для временного хранения слов, что обеспечивает возможность записи и модификации данных в ассоциативный массив, невозможность осуществить считывание слов по заданному адресу (просмотр целиком всего слова, определенная часть которого совпала с аргументом поиска).

Для исправления этих недостатков используется так называемая диагональная система адресации. При данной системе адресации в каждый i-тый столбец помещается i-ый разряд j-того слова, причем в каждом последующем столбце порядок слов смещается на единицу, что вызывает “диагональную” запись разрядов слов.

В данном случае был использован квадратный массив 16x16 бит. Для получения массива с большим объемом как правило используется объединение нескольких таких квадратных массивов.

Данная система адресации требует большей аппаратной базы по сравнению с АП базовой конструкции, однако она предоставляет возможность адресного считывания и записи как по словам, так и по разрядным столбцам