**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

**Отчет**

По дисциплине: Аппаратные основы интеллектуальных систем

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

Выполнила:                                   Рабушка А. А.

Группа 021703

Проверил:                                    Захаров В.В.

**Минск 2021**

**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.

**Задача:**

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 f5иf10, f0иf15 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:

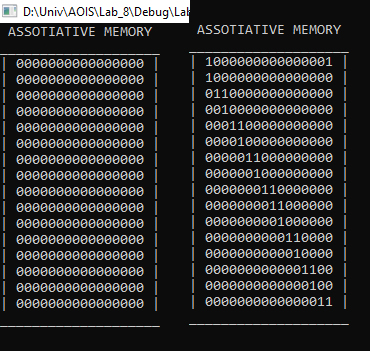
1. tbl – массив слов, хранимый в памяти
2. buff – буфер для временного хранения слов

Класс содержит 9 методов:

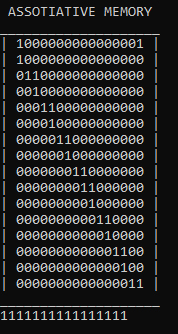
1. addition() – метод сложения двоичных слов, реализованный в лабораторной работе №1.
2. asoc\_memo() – конструктор класса. Инициализирует tbl 16-ю 16-разрядными словами, каждый разряд которого – “ложь”.
3. print\_tbl() – вывод содержимого tbl на экран
4. add\_el() – добавление слова в tbl. Слова добавляются 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. Стоит отметить, что результат будет заведомо равен слову 0000’0000’0000’0000 (x \* !x = 0)
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

Далее соответственно списку методов будут приведены скриншоты результата их выполнения. На левой половине изображения будет отображено состояние памяти до выполнения функции, на правой – после (либо вывод иной информации на экран).

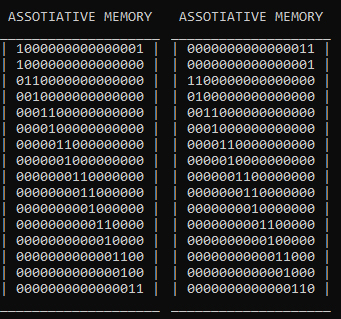
1. Добавление слов 1111111111111111 и 1010101011010101



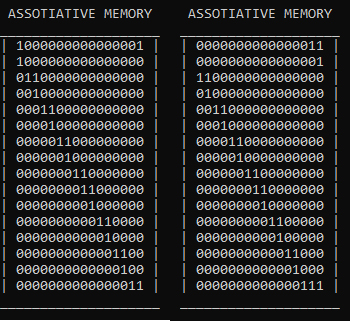
1. Вывод на экран слова, 0-ой символ которого содержится в 0-ом символе первой строки



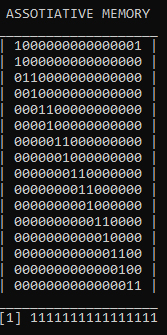
1. Выполнение логических функций F5&F10 (операнды 0, 1)



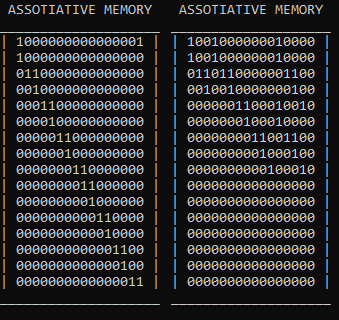
1. Выполнение логических функций F0&F15 (операнды 0, 1)



1. Поиск по соответствию (ключ – “111111111”)



1. Выполнение арифметических действий



**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.

Известно, что ассоциативные процессоры (АП), использующие

адресацию только по разрядным столбцам, имеют ряд недостатков, в

частности:

а) запись данных в ассоциативный массив (АМ) приходится производить

по разрядным столбцам, из-за чего все содержимое массива должно быть

предварительно занесено в некоторый буфер;

б) большие трудности вызывает также модификация данных

(изменение лишь одного слова требует перезаписи содержимого всего

массива);

в) в рамках базовой конфигурации АМ практически невозможно

осуществить считывание слов по заданному адресу (эта операция

необходима для того, чтобы просмотреть целиком все слово, определенная

часть которого совпала с аргументом поиска).

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

Диагональная адресация: - разрядные столбцы записываются по диагоналям, образованным запоминающими ячейками;

- слова размещаются по столбцам.

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.