Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №8 по теме **«Моделирование ассоциативной памяти с системой адресации по разрядным столбцам и по словам»** **по курсу «АОИС»**

Выполнил: Пилипейко В.И гр.921701

Проверил: Захаров В.В

МИНСК

2020

**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.

**Задание**

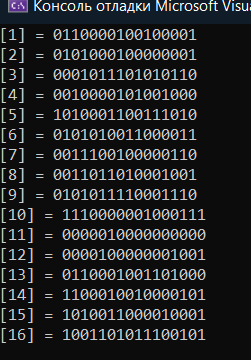
Построить и проверить программную модель ассоциативной памяти с диагональной адресацией на основе сумматора.

**Вариант 19(3)**

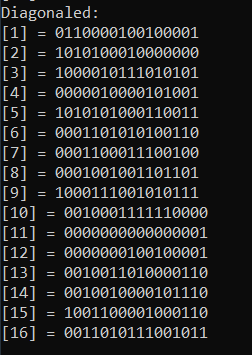
Упорядоченная выборка (сортировка). Арифметические операции над  полями слов  - cложение полей A*j* и B*j*  в словах S*j*, у которых V*j* совпадает с заданным V = 000…111. Логические функции над разрядными столбцами - *f*6 и *f*9, *f*4 и *f*11.

**Ход и результаты работы**

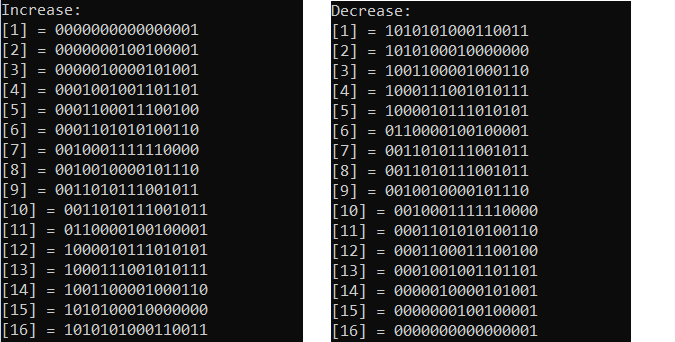
1. Для упрощения демонстрации работы реализованной программы исходная память - массив 16х16 разрядов- будет заполнена псевдослучайным образом.



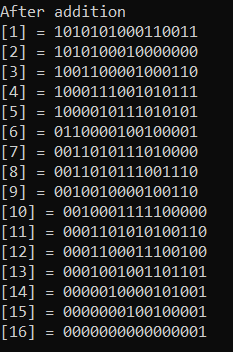
1. Преобразуем структуру адресации элементов памяти в диагональную. Для этого воспользуемся методом сдвига каждого слова (строки массива) на такое количество ячеек вправо, которое равно порядковому номеру слова (строки).



1. Далее произведем сортировку массива соответственно с заданием лабораторной работы методами, описанными в предыдущей лабораторной работе.



1. Произведем сложение полей слов с маской V = 001.



1. Посчитаем сложение первого и шестнадцатого слова памяти функции f6 и f9. Для этого поместим первый аргумент в буфер, и с словом из буфера и вторым аргументом посчитаем значения функции f6, которая имеет вид !x1\*x2 + x1\*!x2 (неравнозначность), запишем результат в буфер. Далее возьмем слово из буфера и второй аргумент, посчитаем значение функции f9 = x1\*x2 + !x1\*!x2 (эквивалентность). Получаем:



1. Посчитаем сложение четвертого и пятого слова памяти функции *f*4 и *f*11. Для этого поместим первый аргумент в буфер, и с словом из буфера и вторым аргументом посчитаем значения функции *f*4, которая имеет вид !*x*1\**x*2 (*запрет 2-го аргумента*), запишем результат в буфер. Далее возьмем слово из буфера и и второй аргумент, посчитаем значение функции *f*11 = *x*1 + !*x*2 (*импликация от 2-го аргумента к 1-му*). Получаем:



**Вывод**

Ассоциативные процессоры, использующие адресацию только по разрядным столбцам, имеют некоторые недостатки: запись данных в АМ необходимо производить по разрядным столбцам, из-за чего все содержимое всего массива памяти должно быть предварительно занесено в буфер; модификация данных предусматривает даже при изменении всего одного слова перезапись содержимого всего массива памяти.

Диагональная адресация призвана решить проблему одномерности считывания/записи в массив. Этот метод адресации предусматривает сдвиг каждого слова (строки массива) на такое количество ячеек вправо, которое равно порядковому номеру слова (строки), а после - транспонирование массива (эквивалентно транспонированию матрицы).

Для выполнения операций (арифметических и логических) используется буфер - массив или вовсе одно слово, который предназначен для временного хранения содержимого обработанного аргумента.

Для логической операции первый из операндов загружается в буферный регистр, заданная логическая операция выполняется между вторым операндом и регистром, после чего результат, заменивший прежнее содержимое регистра, пересылается из этого регистра в позицию результирующего разрядного столбца.