

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический университет
Петра Великого»

Институт компьютерных наук и кибербезопасности
Направление: 02.03.01 Математика и компьютерные науки

Архитектура суперкомпьютеров
ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Моделирование кэш-памяти прямого отображения

Вариант 11

Студент,
группы 5130201/30002

_____ Михайлова А. А.

Преподаватель

_____ Чуватов М. В.

«_____» _____ 2025 г.

Санкт-Петербург, 2025

Содержание

Введение	3
1 Первая часть	4
1.1 Постановка задачи	4
1.2 Особенности реализации	4
1.3 Результаты работы	7
2 Вторая часть	10
2.1 Постановка задачи	10
2.2 Особенности реализации	10
2.3 Результаты работы	11
Введение	12
Список использованной литературы	13

Введение

Кэш-память прямого отображения — это один из типов кэш-организации, при котором каждая строка основной памяти может быть размещена только в одном строго определенном месте кэш-памяти. Такое отображение реализуется за счёт использования части адреса — индекса, который однозначно указывает на положение строки в кэше. Оставшаяся часть адреса — тэг — используется для проверки принадлежности данных нужной области памяти.

При обращении к данным процессор сначала проверяет кэш: по индексу выбирается соответствующая строка из теговой директории, после чего сравнивается её тэг с тэгом запрашиваемого адреса. Если они совпадают и строка помечена как действительная (валидная), происходит кэш-попадание, и данные берутся непосредственно из кэша. В противном случае — кэш-промах, и данные загружаются из основной памяти, после чего записываются в кэш для последующего использования.

Целью данной работы является моделирование простой кэш-памяти прямого отображения в среде Logisim, а также изучение базовых механизмов кэширования, управления состоянием строк и организации взаимодействия между процессором, кэшем и основной памятью.

1 Первая часть

1.1 Постановка задачи

Требуется изготовить простую модель кэш-памяти прямого отображения в Logisim. Конкретнее, будет моделироваться ситуация когда центральный процессор выполняет операции чтения из основной памяти по указанному оператором адресу.

Определены следующие начальные условия:

1. Побайтовая адресация;
2. Строковый доступ к основной памяти;
3. Состояние строки кэша задаётся одним битом действительности;
4. Каждое чтение из основной памяти должно кэшироваться;
5. Байт, читаемый из памяти по указанному адресу, должен записываться в отдельный регистр разрядностью 1 байт.

Используются 4 строки кэш-памяти, длина строки 16 байт, адресуемая память 512 байт.

Схема должна функционировать следующим образом:

1. Оператор указывает адрес нужного ему слова (байта) в основной памяти;
2. Кэш-контроллер анализирует адрес и проверяет наличие соответствующей строки в кэш-памяти;
3. В случае попадания запрошенное слово (байт) загружается из кэш-памяти в регистр-получатель;
4. В случае промаха соответствующая строка читается из основной памяти, записывается в кэш-память, запрошенное слово (байт) загружается из кэш-памяти в регистр-получатель.

Тэговая директория, содержащая таблицу тэгов и состояний строк кэш-памяти, должна быть реализована в виде набора регистров (регистр для хранения тэга и состояния строки), разрядности элементов и их количества должны точно соответствовать структуре адреса (тэг-индекс-смещение). Кэш-память также должна быть выполнена в виде набора регистров, хранение строки может быть реализовано как в одном регистре, так и в нескольких регистрах.

Размер адреса 9 бит, где 3 бита тэг, 2 бита индекс, 4 бита смещение.

1.2 Особенности реализации

В начале входной адрес разбивается на тэг, индекс и смещение – рис. 1.

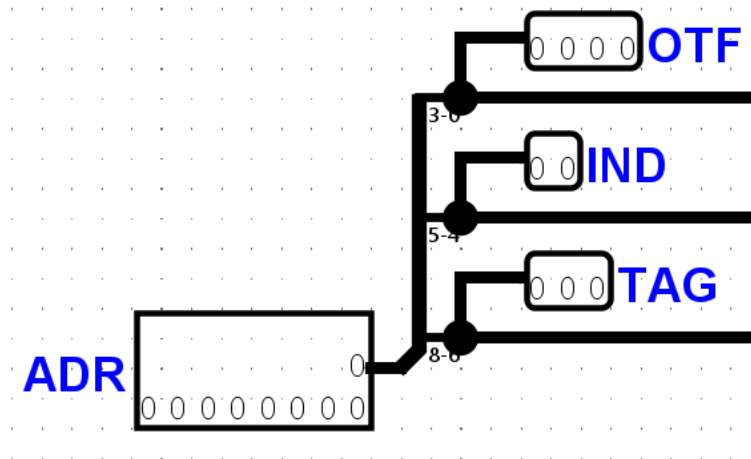


Рис. 1: Разбиение на тэг, индекс и смещение

Индекс идет на декодер для выбора строки кэша, с которой будут считываться данные или в которую будет записаться данные из памяти – рис. 2.

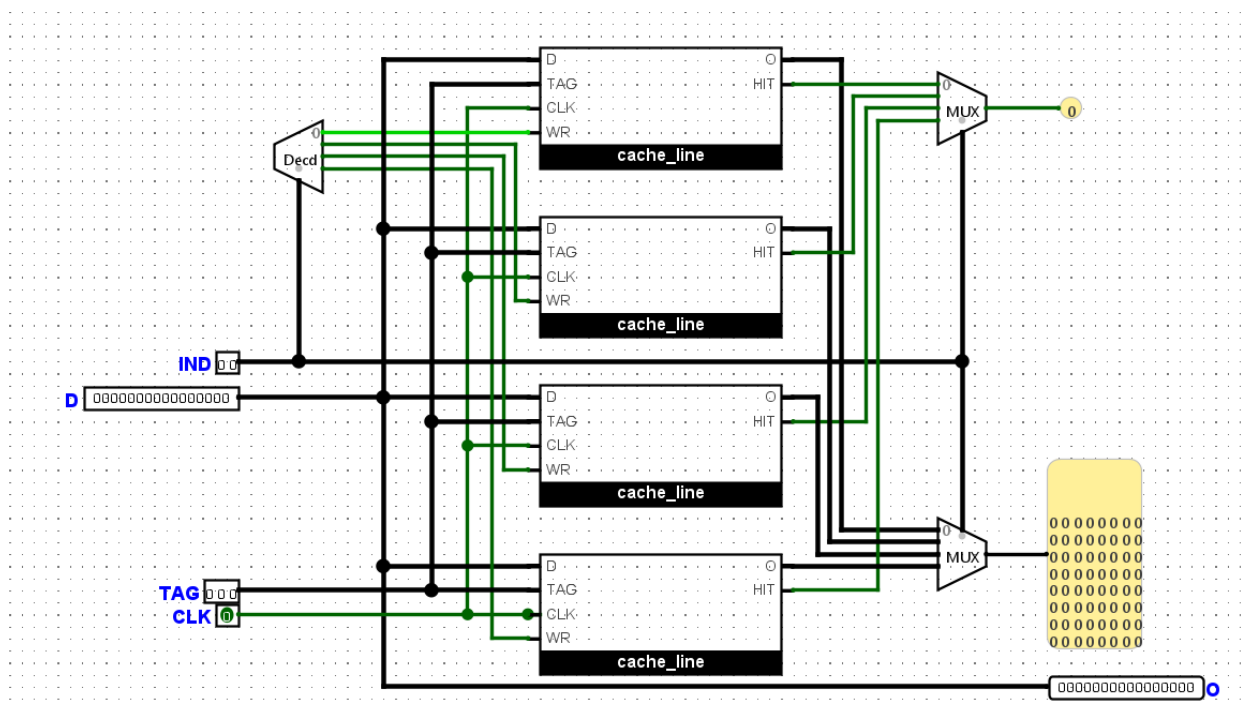


Рис. 2: Соединение индекса с декодером

Строка кэша выделена в отдельную структуру, внутри неё присутствует регистр для хранения данных, регистр для хранения тега и регистр для хранения бита валидности. Является ли строка валидной вычисляется путем сравнения входного тега и того, который уже записан в строку. На вход в нее поступает значение для перезагрузки строк, значение тактового генератора, выбирающий сигнал, значение строки и тег – рис. 3.

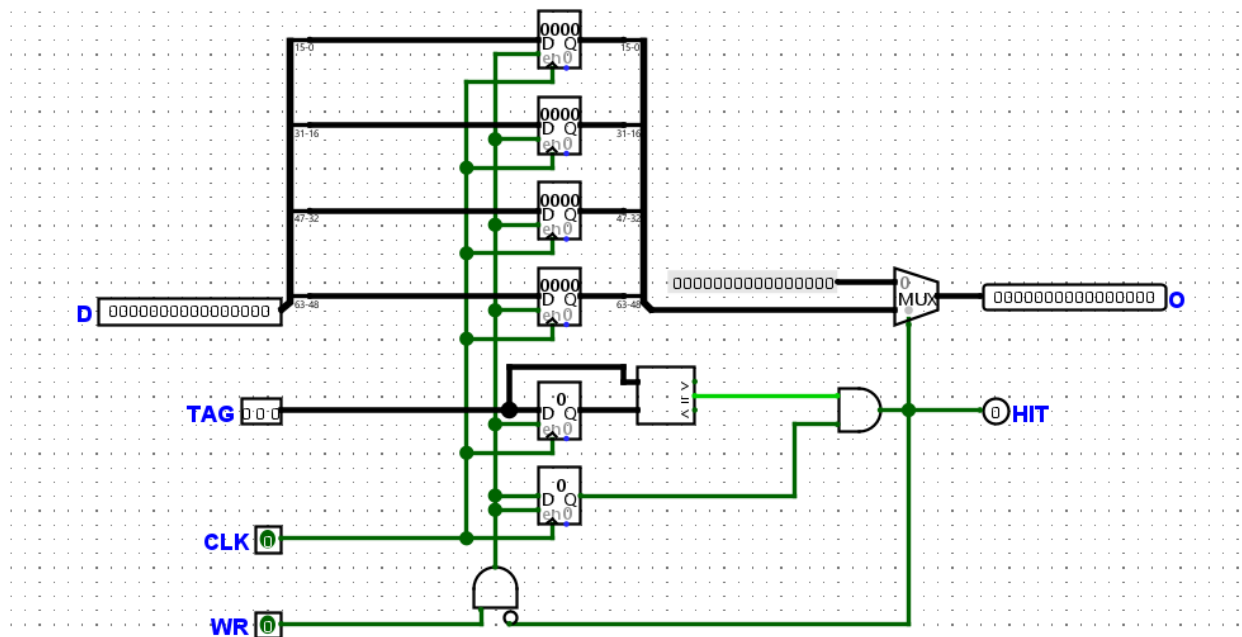


Рис. 3: Строка кеша

Основными ячейками памяти являются 2 ОЗУ. Оно имеет разрядность адреса 5 битов (тег + индекс) и разрядность данных 64 бита – рис 4.

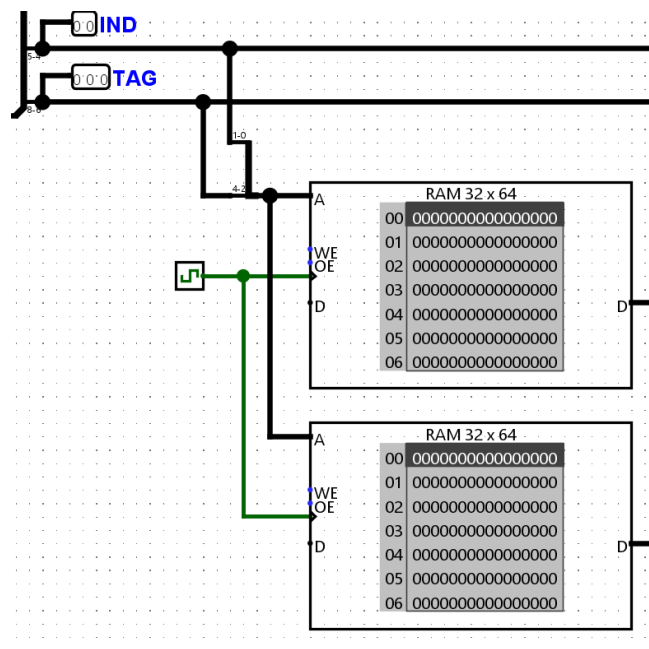


Рис. 4: ОЗУ

Далее строка из памяти выходит на мультиплексор, чтобы считать данные с нужной строки с нужным индексом – рис. 5.

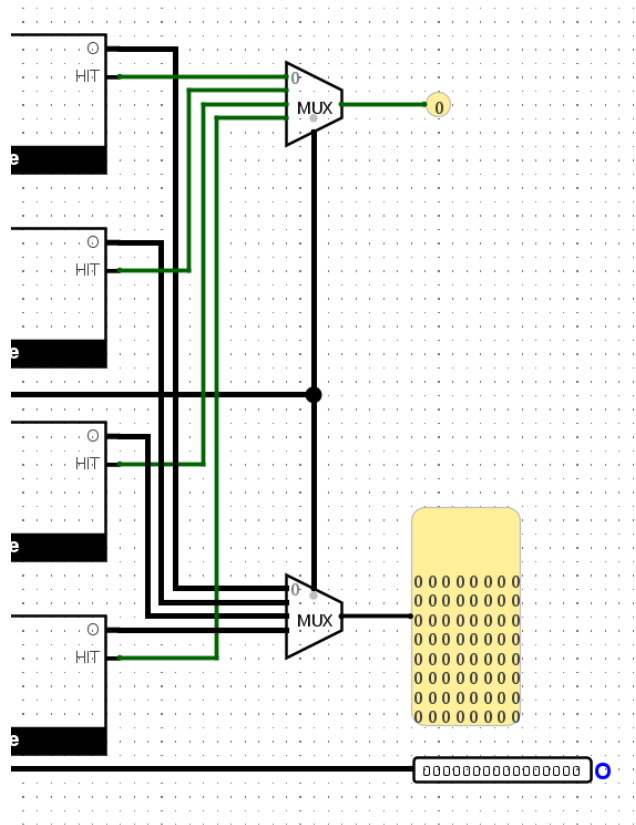


Рис. 5: Считывание данных с нужной строки

Далее мультиплексор делит строку 64 бита на 8 слов и отображает результат в регистре – рис. 6.

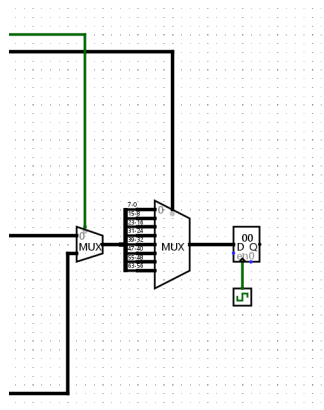


Рис. 6: Деление строки

1.3 Результаты работы

В ходе выполнения первой части лабораторной работы была реализована простая модель кеш-памяти прямого отображения в Logisim. Центральный процессор выполняет операции чтения из основной памяти по указанному оператором адресу.

Используется побайтовая адресация, строковой доступ к основной памяти, состояние строки кеша задается одним битом действительности, каждое чтение из основной памяти кешируется, байт, читаемый из памяти по указанному адресу записывается в отдельный регистр разрядностью 1 байт.

Схема функционирует в соответствии с поставленной задачей. Модель работает правильно и соответствует принципам работы кэш-памяти прямого отображения.

Итоговая схема представлена на рис. – 7, 8, 9.

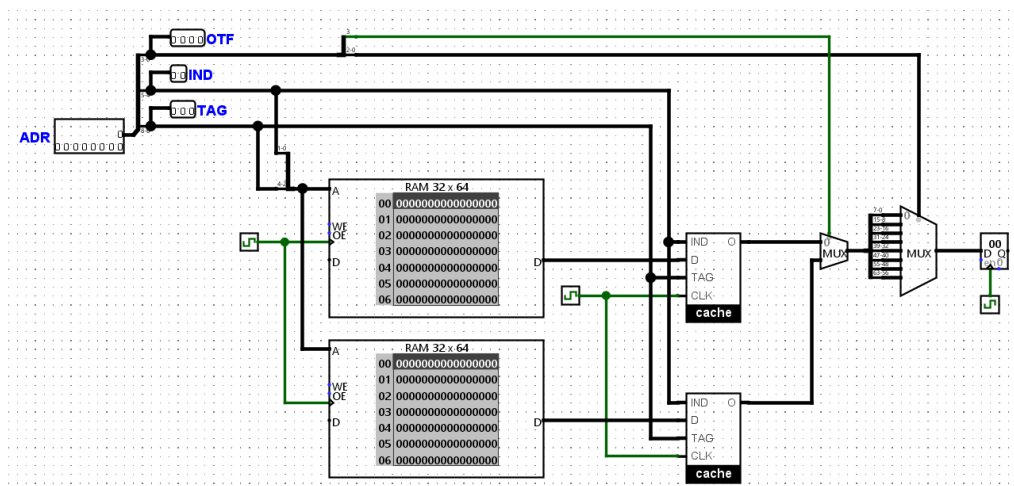


Рис. 7: main

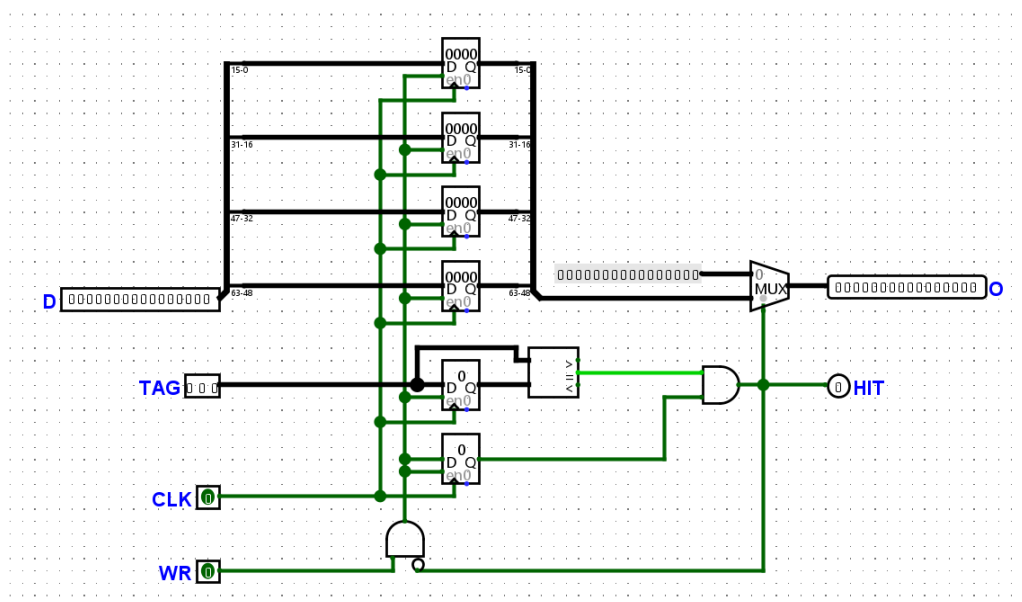


Рис. 8: cache_line

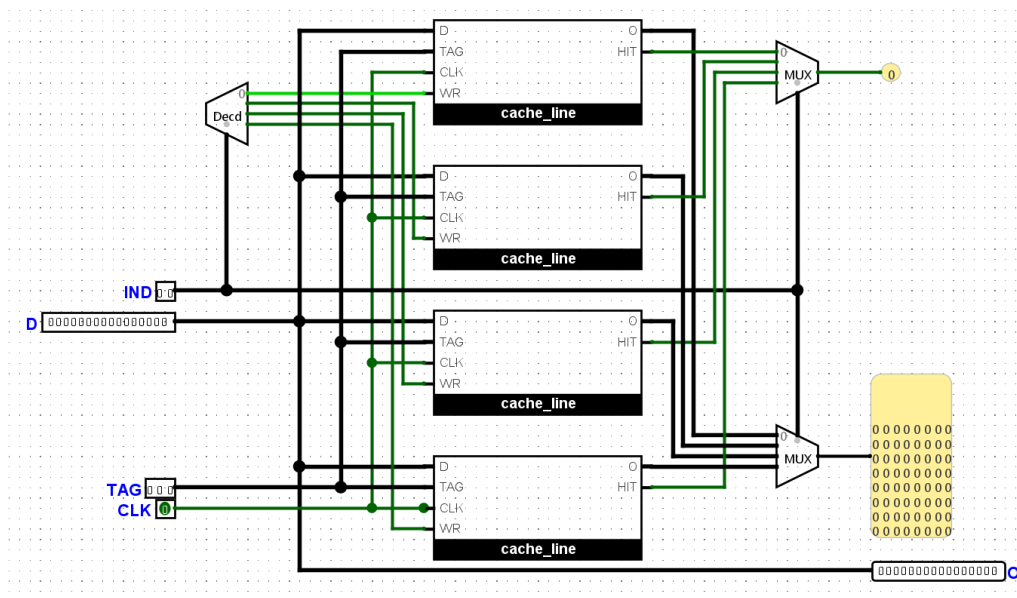


Рис. 9: cache

2 Вторая часть

2.1 Постановка задачи

Во второй части лабораторной работы требуется модифицировать ранее созданную модель. Необходимо разделить линию тактирования, замедлив в 4 раза те ответвления, которые подключены к элементам ОЗУ (основная память). При выполнении операции чтения слова по указанному адресу в целевой регистр сначала выполнять проверку наличия строки в кэше и в случае попадания читать слово из строки кэша, но при промахе продолжать обращение в основной памяти, читать строку оттуда, записывая в кэш, а запрошенное слово в целевой регистр.

2.2 Особенности реализации

Для замедления работы ОЗУ относительно кэша в 4 раза была реализована схема, использующая счетчик. В ней при каждом такте счетчик увеличивается на 1 и когда значение становится равным трём, то тактовый сигнал подается на замедленную сеть. То есть на вход чтения для ОЗУ. Также внутри счетчика было учтена ситуация, когда значения адреса меняется, в таком случае счетчик обнуляется. Также для корректной обработки значения до первого срабатывания замедленной линии была добавлена схема обнуляющая целевой регистр с момента начала работы и до первого чтения из ОЗУ – рис. 10.

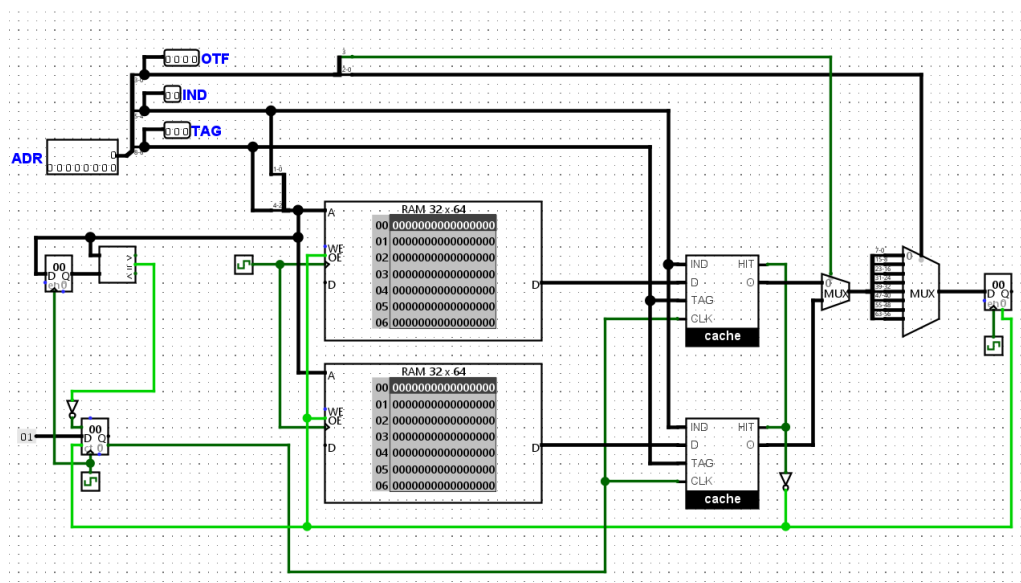


Рис. 10: Замедление работы ОЗУ

2.3 Результаты работы

В данной части лабораторной работы была модифицирована схема подачи тактирующего сигнала к основной памяти и тактирующего сигнала для обновления значения в кэш-памяти. Линия тактирования, идущая к основной памяти и к кэшу, была замедлена в 4 раза. Для этого была реализована схема, которая при наличии строки в кэш-памяти, соответствующей выбранному оператором адресу, проводит сигнал тактирования за 1 такт, а при отсутствии такой строки выполняет подсчёт количества проведённых тактов и при значении счётчика равном «3» проводит такт к основной памяти, что позволяет реализовать доступ к данным из основной памяти за 4 такта. Также была продумана ситуация, когда оператор изменяет адрес выбранных данных – в этом случае счётчик обнуляется, что делает доступ к данным из основной памяти всегда за 4 такта.

Заключение

В результате выполнения обеих частей лабораторной работы была реализована полностью функционирующая модель кэш-памяти прямого отображения, в которой реализована запись и чтение данных в кэш. Также в данной работе реализовано замедление подачи тактового сигнала к основной памяти для получения данных из неё в четыре раза, что позволяет симитировать быстродействие кэша по сравнению с обычной памятью. Данная работа позволила глубже понять принципы работы кэш памяти и управление состоянием строк в памяти.

Список литературы

- [1] Logisim evolution URL: <https://github.com/logisim-evolution/logisim-evolution>
- [2] Руководство «Как стать пользователем Logisim» URL: <https://cburch.com/logisim/docs/2.7/ru/html/guide/index.html>