

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

Реализация записи на массив RAID-5 в среде Logisim
Вариант 11

Студент,
группы 5130201/30002

Михайлова А.А.

Руководитель

Чуватов М. В.

Санкт-Петербург - 2025 г.

Содержание

Введение	3
1. Постановка задачи	4
2. Ход работы.....	5
2.1. Разбиение сообщения на блоки.....	5
2.2. Подсчёт избыточности.....	5
2.3. Механизм чередования блоков сообщения и избыточности. Запись сообщения на диски .	6
3. Проверка записи данных	8
4. Результаты работы.....	10
Список источников	11

Введение

RAID — это метод объединения нескольких физических дисков в единый логический блок, что позволяет повысить надежность, скорость работы или оба параметра одновременно. Различные уровни RAID разработаны для решения специфических задач, таких как защита данных от потерь и улучшение производительности системы.

Одним из наиболее востребованных уровней является RAID 5, который обеспечивает баланс между отказоустойчивостью и производительностью. В RAID 5 используется метод чередования данных с добавлением блоков проверки четности. Эти блоки позволяют восстановить информацию в случае отказа одного из дисков в массиве. Благодаря распределению данных и блоков четности по всем дискам, RAID 5 обеспечивает ускорение операций чтения/записи за счет параллельной работы нескольких дисков, а также защищает данные от потерь при сбое любого из них.

1. Постановка задачи

В среде Logisim необходимо разработать модель дискового массива, реализующую запись данных, формируемых оператором по выбранному им же адресу, с учётом следующих особенностей:

- 1) уровень массива — RAID 5;
- 2) количество дисков в массиве – 5;
- 3) адрес выбирается оператором в диапазоне от 0 до 255 с помощью 8-разрядного контакта;
- 4) размер блока данных – 24 бит;
- 5) вычисление блока избыточности необходимо выполнять по алгоритму $R = (M1 + M2) \text{ XOR } (M3 + M4)$;
- 6) порядок перемещения блока избыточности по дискам массива в зависимости от адреса по возрастанию;

Вариант чередования «По возрастанию» демонстрирует следующий пример:

адрес 0: M1 M2 R

адрес 1: M1 R M2

адрес 2: R M1 M2

адрес 3: M1 M2 R

адрес 4: M1 R M2

...

- 7) в качестве замены дисковых накопителей в модели необходимо использовать элемент «ОЗУ». Разрядность ячеек ОЗУ определяется как 24 бита в сообщении / (5 дисков - 1 избыточность), и того 6 бит в сообщении, а также 1 дополнительный бит для записи избыточности. А значит необходимо будет использовать ячейки памяти для ОЗУ размером 7 бит.

2. Ход работы

2.1.Разбиение сообщения на блоки

Для начала сообщение длиной 24 бита требуется разделить на 5 дисков. С учетом избыточности, которая занимает 1 диск делим сообщение на 4 части. Для этого используем разветвители (рис.1).

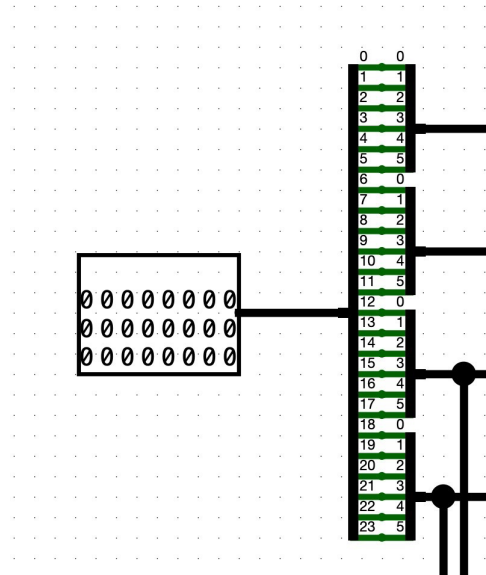


Рис.1. Разделение сообщения из 24 битов на 4 6-битных сигнала

2.2.Подсчёт избыточности

Для удобства, выносим подсчёт избыточности в отдельный модуль “R”. Подсчет идет по формуле

$R = (M1 + M2) \text{ XOR } (M3 + M4)$ – где XOR – исключающее ИЛИ. Из – за суммирования возможна ситуация выхода за пределы 6 бит сообщения, расширим длину до 7, дополнив спереди незначащими нулями. Результат представлен на рис.2.

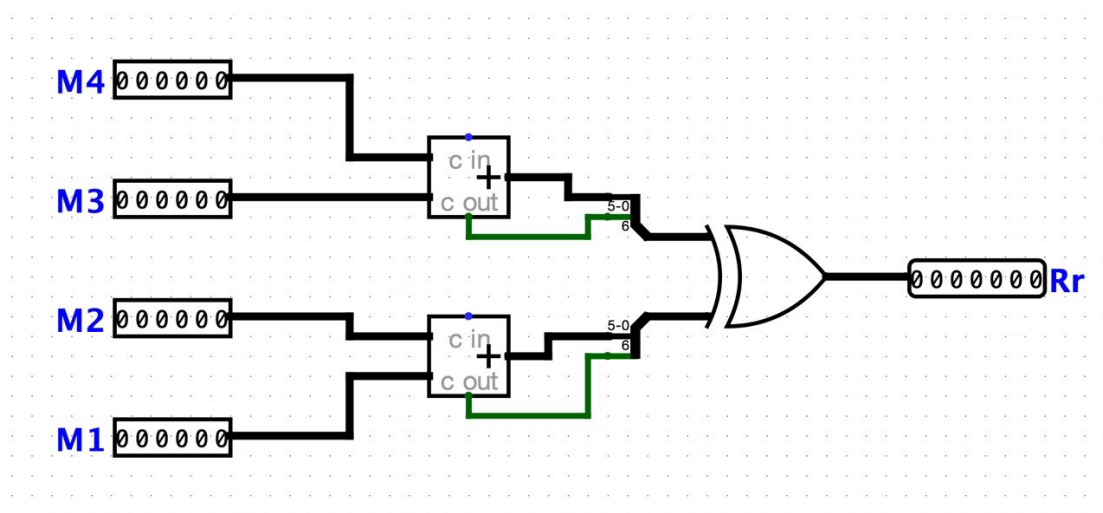


Рис.2. Вычисление блока избыточности

Во избежании конфликтов при записи на диск поступим также с M1, M2, M3, M4. (рис.3)

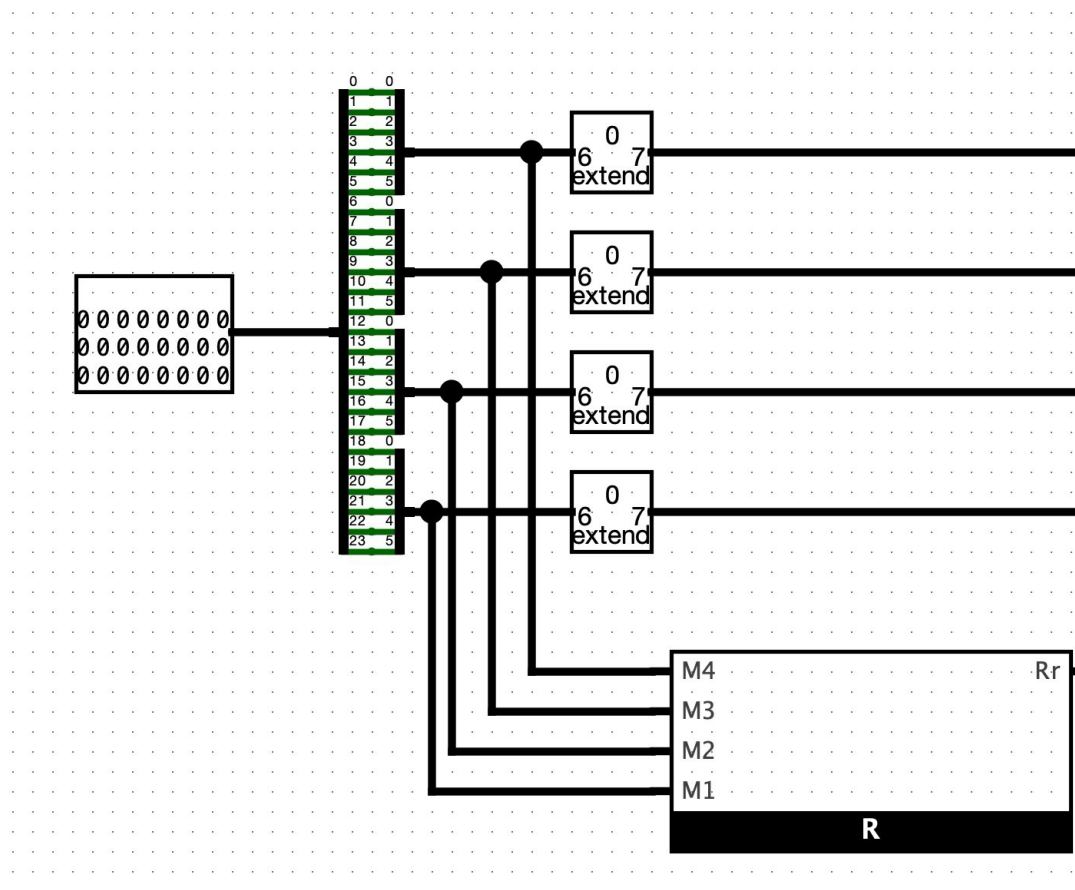


Рис.3. Расширение M1, M2, M3, M4 до 7 бит, внедрение модуля избыточности

2.3.Механизм чередования блоков сообщения и избыточности. Запись сообщения на диски

Теперь необходимо реализовать механизм чередования блоков M1, M2, M3, M4, R по дискам в зависимости от адреса.

адрес 0: M1 M2 M3 M4 R

адрес 1: M1 M2 M3 R M4

адрес 2: M1 M2 R M3 M4

адрес 3: M1 R M2 M3 M4

адрес 4: R M1 M2 M3 M4

С помощью элемента делитель получаем остаток от деления адреса на количество состояний. Тогда состояния будут циклично сменяться. С помощью демультимплексоров каждый блок сообщений и избыточность направляем на соответствующий диск.

Далее записываем сообщения на диски ОЗУ с выбором режима ячейки памяти на 7 бит. Вводим 3 переключателя. Writing осуществляет запись на диск только при состоянии “логическая единица”. Тактовый генератор отвечает за запись в память при переключении. Поле adr – позволяет выбрать любой 8 разрядный адрес.

Полная схема с механизмами чередования блоков и записи сообщения на диски представлена на рис. 4.

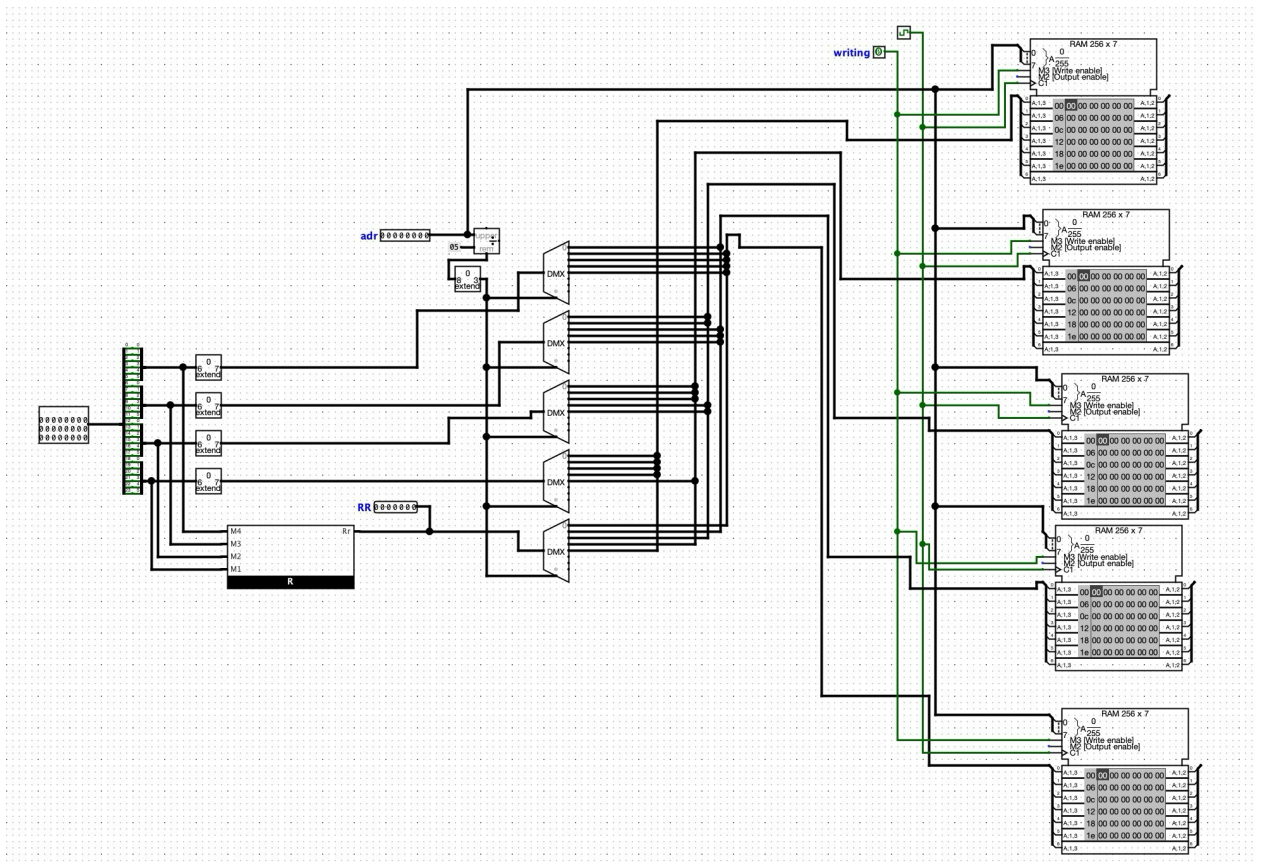


Рис. 4. Модель RAID-5 массива

3. Проверка записи данных

Для проверки данных выберем избыточность 1111111(рис. 4), что соответствует значению на диске 7f и воспользуемся тактовым сигналом на адрес 00000000. В результате избыточность на 5 диске на адресе 00000000. Далее меняем адрес на 00000001 и снова пускаем тактовый сигнал. Теперь наше максимальное значение избыточности на 4 диске на адресе 00000001(рис.5). Повторим это с адресами 00000010, 00000011, 00000100, 00000101. В результате мы видим, что избыточность 7f корректно записывается на диски, чередуясь в порядке возрастания(рис.6).

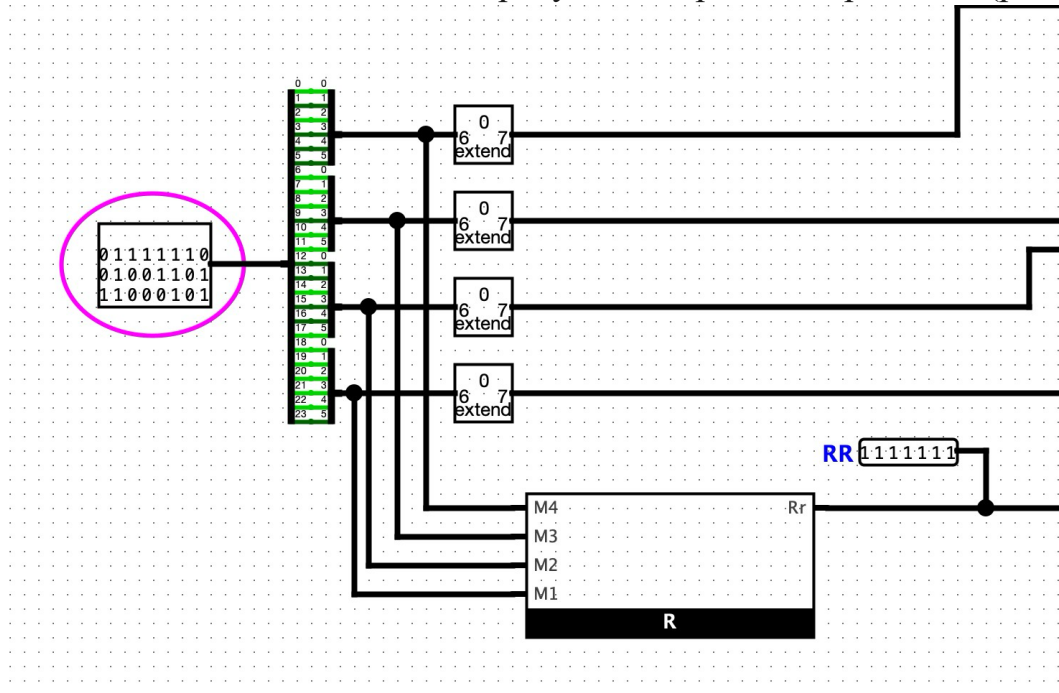


Рис. 4. Максимальное значение избыточности

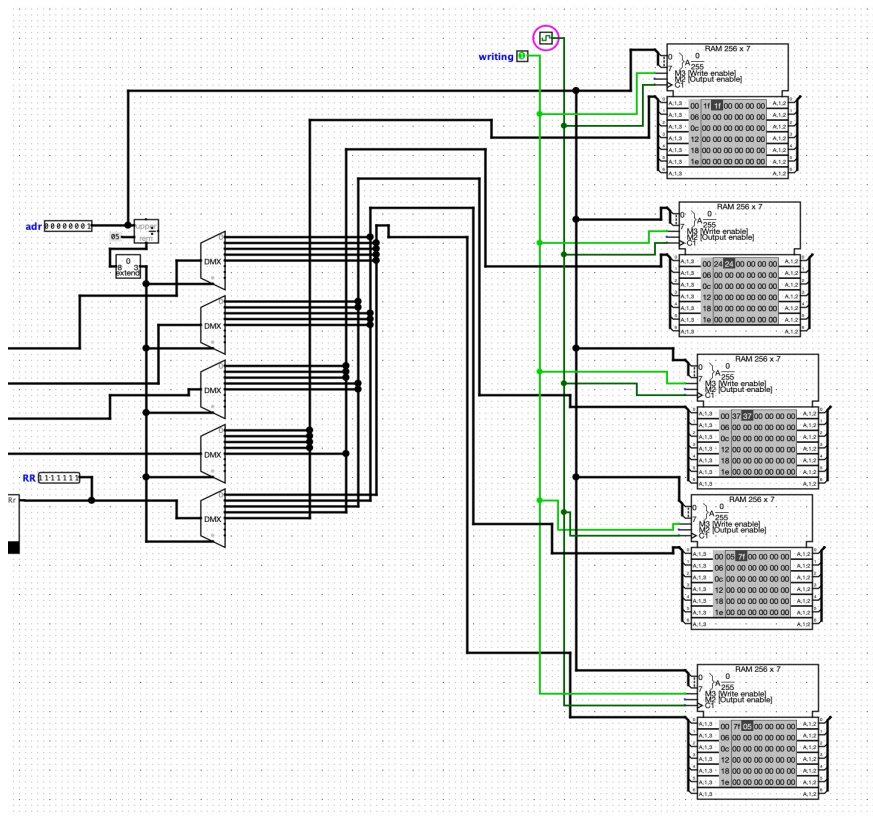


Рис. 5 Значение избыточности при втором такте

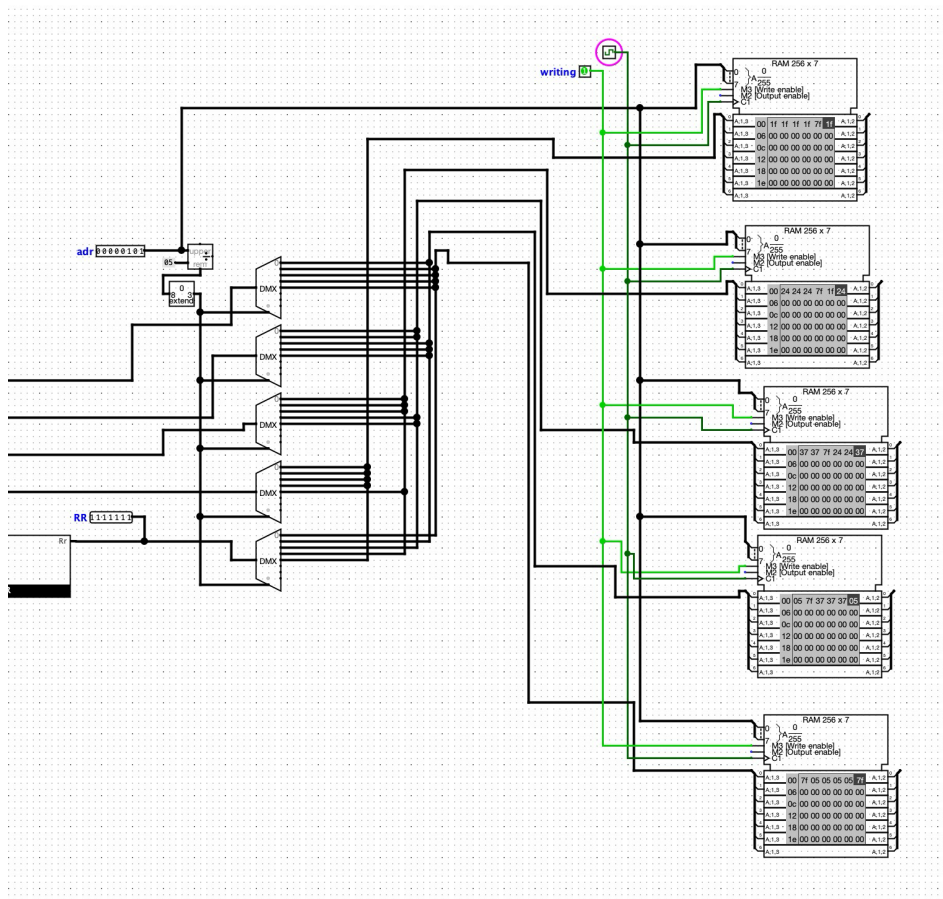


Рис. 6 Результат проверки по 6 адресам

Вводя переключатель “writing” в состояние “логический ноль” запись перестает осуществляться (рис. 7).

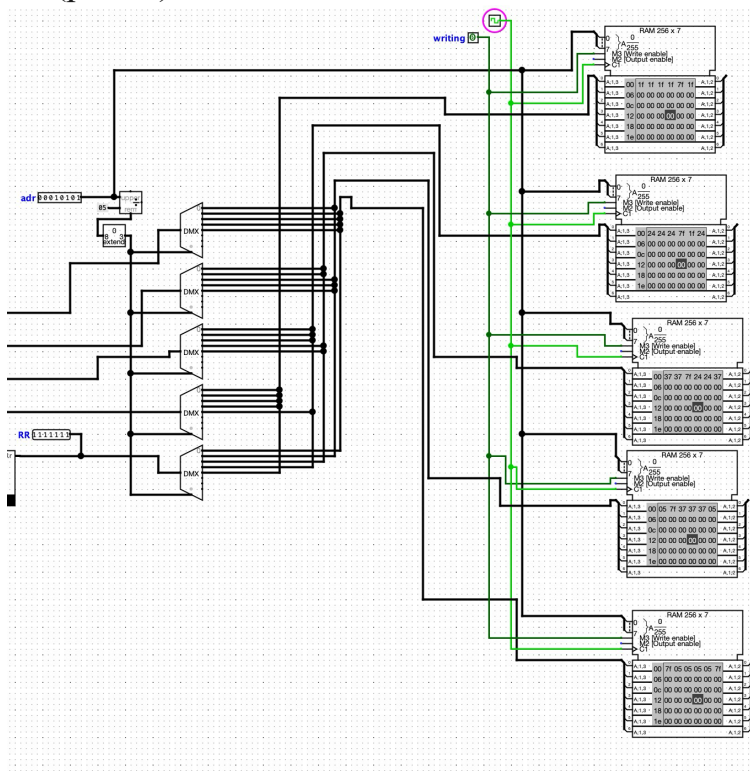


Рис. 7 Переключатель “writing” в состоянии “логический ноль”

4. Результаты работы

В результате работы была разработана модель дискового массива в среде Logisim Evolution, реализующая запись данных, формируемых оператором по выбранному им же адресу, уровень массива – RAID 5, в массиве 5 дисков, адрес выбирается оператором в диапазоне от 0 до 255 с помощью 8-разрядного контакта, размер блока данных 24 бита, вычисление блока избыточности выполняется по алгоритму $R = (M1 + M2) \text{ XOR } (M3 + M4)$, порядок перемещения блока избыточности по дискам массива в зависимости от адреса по возрастанию.

После реализации модели была проведена проверка записи данных. Схема работает в соответствии с принципами RAID – 5.

Список источников

- 1) Logisim-evolution URL: <https://github.com/logisim-evolution/logisim-evolution>
- 2) RAID // Википедия URL: <https://ru.wikipedia.org/wiki/RAID>
- 3) Руководство “Как стать пользователем Logisim” URL:
<https://cburch.com/logisim/docs/2.7/ru/html/guide/index.html>