

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

15.03.06 - Мехатроника и Робототехника

Направление: Искусственный интеллект

Пояснительная записка

Тема работы:

«Rectify»

Булькин Сергей Васильевич,

24942

Морозова Софья Дмитриевна,

24942

Ребриков Никита Сергеевич,

24942

Новосибирск

2025

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ.....	3
2. ЦЕЛЬ ПРОЕКТА.....	3
3. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	3
4. ЗАДАЧИ ПРОЕКТА.....	4
5. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ.....	4
6. ИГРА «RECTIFY».....	5
6.1. ПРАВИЛА ИГРЫ.....	5
6.2. ИГРОВОЙ ПРОЦЕСС.....	5
6.3. УПРАВЛЕНИЕ.....	8
7. ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ.....	9
8. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	
8.1. Cell_controller	
8.2. o_math_i_math	
8.3 display_controller	
8.4. input_contoroller	
8.5. People_controller	
8.6. MCU	
8.7. SPI interface	
8.8. Main	
9. ЗАКЛЮЧЕНИЕ	
10. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ	

1 ВВЕДЕНИЕ

В рамках образовательной программы CS/MR Digital Platforms на учебный год 2024/25 каждый студент должен разработать и внедрить программное обеспечение для игровой системы, работающей на микроконтроллере CdM-8.

Название программы:
Электронная реализация игры «Rectify» на базе микроконтроллера CdM-8.

Rectify - это оригинальная игра, придуманная участниками проекта.

2 ЦЕЛЬ ПРОЕКТА

Разработка электронной реализации игры «Rectify» с использованием программного обеспечения Logisim с применением ассемблера для написания кода, управляющего работой логических элементов и компонентов схемы.

3 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Определение проблемы программы:
Современные образовательные программы требуют практического освоения принципов работы цифровых схем и микроконтроллеров. Данный проект решает задачу создания интерактивной игровой системы, которая:

- Наглядно демонстрирует работу логических элементов;
- Позволяет отработать навыки программирования на ассемблере;
- Служит примером проектирования законченного устройства на базе CdM-8.

Область применения программы:

- Образовательный процесс – изучение архитектуры микроконтроллеров, цифровой схемотехники и ассемблерного программирования.
- Прототипирование электронных устройств – игра может быть использована как основа для более сложных проектов, связанных с управлением периферийными устройствами.

4 ЗАДАЧИ ПРОЕКТА

- Определить основные логические операции и взаимодействия, необходимые для реализации игры.
- Спроектировать основные блоки игры (игровое поле, счетчик очков, контрольные элементы).
- Разработать архитектуру схемы и определить необходимые компоненты Logisim.
- Разработать программное обеспечение для управления логическими элементами.
- Реализовать обработку игровых событий и взаимодействие с пользователем.
- Обеспечить корректную работу всех компонентов схемы.
- Подготовить документацию по разработке, описать логику работы, структуру схемы и функции кода.
- Распределить роли и обязанности внутри команды (схемотехники, программисты, тестировщики, документалисты).
- Обеспечить эффективную коммуникацию и координацию между членами команды.
- Вести общий репозиторий и систему контроля версий для совместной работы над проектом.

5 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

В ходе разработки электронной реализации игры «Rectify» были использованы следующие программы и инструменты:

- Logisim — для проектирования и моделирования логической схемы;
- Среда разработки (sosoIDE с авторскими поправками) — для написания и отладки кода на языке программирования: Ассемблер;
- Sourcetree, GIT — для организации совместной работы над проектом;

6 ИГРА «RECTIFY»

Rectify — это интеллектуальная игра. В ней участвуют два игрока. Они захватывают поле 8 на 8 при помощи очков энергии.

6.1 ПРАВИЛА ИГРЫ

Игровое поле:

Состоит из клеток, включает в себя генератор задач, таблицу для фиксации текущих заданий, систему учета очков прогресса и дополнительной энергии. Взаимодействие с игровым полем осуществляется посредством клавиатуры.

Клетка:

Трехзначное число и дисплей. Каждая клетка изначально имеет в себе 7 щитов и 3 энергии. Дисплей предназначен для идентификации принадлежности клетки к определенному игроку и демонстрации задания. На дисплей для чисел выводиться только количество энергии.

Щиты - броня клетки. Захват клетки невозможен, если у нее есть щиты. Они восстанавливаются после завершения хода обоих игроков.

Энергия - ресурс клетки для захвата. Её можно перемещать.

Задание:

Прямоугольник, размещаемый на свободных клетках и граничащий с клеткой игрока. В генераторе задач отображаются два значения, представляющих длины сторон прямоугольника. Допустимые значения длины сторон варьируются от 1 до 4 единиц.

Задание создается на этапе планирования. Игрок может принять или отклонить его: в первом случае параметры прямоугольника сохраняются в таблице. Максимум игрок может принять пять заданий. Во втором случае игрок переходит в следующую фазу игры.

Для выполнения задания нужно захватить весь прямоугольник, размещенный на поле. Его площадь конвертируется в очки прогресса и в дополнительную энергию.

Очки прогресса: Площадь задания делится на четыре и округляется в большую сторону.

Дополнительная энергия: Площадь задания умножается на два.

Пример: Оранжевый игрок выполняет задание 3 2. Он получает 6 дополнительной энергии и 2 очка прогресса.

6.2 ИГРОВОЙ ПРОЦЕСС

Подготовка:

Игроки выбирают точку старта. Эти клетки красятся в цвет игрока и получают стартовые 10 энергии. (Можно выбирать любую стартовую позицию, но убедитесь, что между выбранными клетками есть минимум две свободные.)

Игра:

Игроки ходят по очереди. Ход каждого игрока состоит из четырех фаз. По окончании этих фаз ход передается другому игроку.

- *Фаза планирования*

Генератор задания создает два числа.

Игрок решает берет ли он задание и проверяет есть ли место в таблице.

Условия размещения задания:

Размещается на свободных клетках и граничит с клеткой игрока.

Прямоугольник должен быть размещен так, чтобы уже захваченные игроками клетки в нём не находились.

Задание можно поворачивать.

Принимая задание, игрок обязуется его выполнить (захватить все клетки выделенного прямоугольника), после выполнения игрок получит вознаграждение.

Игрок 1 может захватить клетку с заданием игрока 2. Для выполнения задания игроку 2 надо захватить клетку игрока 1 со своим заданием.

- *Фаза перемещения*

Игрок планирует перемещение энергии с одной клетки на другую. Можно передавать только ту энергию, которая была на начало хода игрока. Атаковать можно свободную клетку или занятую другим игроком.

Атака по свободной:

У свободной клетки 7 щитов и 3 энергии, чтобы её захватить, нужно потратить 10 энергии.

Атака по занятой:

Для захвата нужно перенести энергию, чтобы пробить 7 щитов, а после перебить значение противника. Когда игрок перенес энергию равную 7 щитам и количеству энергии противника, клетка становится ничейной. Чтобы забрать её себе, нужно перенести хотя бы еще 1 энергию.

Если клетка после боя осталась ничейной, то она больше не вырабатывает энергию. То есть, для захвата надо будет пробить щиты клетки.

- *Фаза зарядки*

Каждая принадлежащая клетка игроку увеличивает свою энергию на 2 (свободные клетки не получают эту энергию). И каждая клетка восстанавливает щиты (7 единиц).

Передача энергии перемещенной игроком за фазу перемещения.

- *Фаза распределения*

Игрок может распределять дополнительную энергию, если она есть. Но можно и не использовать их.

Конец игры:

Есть два варианта определить победителя

- Побеждает игрок, занявший все поле.
- Игрок с наибольшим количеством очков прогресса. (Остановить игру можно только по обоюдному согласию)

6.3 УПРАВЛЕНИЕ

Управление в игре осуществляется с помощью кнопок клавиатуры.

a	Движение курсора влево
d	Движение курсора вправо
w	Движение курсора вверх
s	Движение курсора вниз
ENTER	Многофункциональная кнопка, используемая для подтверждения ввода.
Backspace	Многофункциональная кнопка, используемая для отмены действия.
A	Перемещение энергии влево
D	Перемещение энергии вправо
W	Перемещение энергии вверх
S	Перемещение энергии вниз
-	Позволяет исправлять количество переданной энергии

Последовательности:

- Выбор клетки старта: wasd - выбор клетки для начала, ENTER подтверждения выбора.
- Передача энергии: wasd - выбор клетки откуда игрок возьмет энергию. SHIFT и wasd - для выбора направления. Ввод в клавиатуру значения для передачи. ENTER подтверждения выбора. Можно повторить операцию с '-' для того чтобы уменьшить количество передаваемой энергии.
- Размещение задания: wasd - выбор клетки рядом с которой игрок захочет разместить задания. Выбор брать задание или нет: брать - ENTER, не брать Backspace. wasd - выбор направления размещения задания. Ввод с клавиатуры - перемещает задание в плоскости, перпендикулярной выбранному направлению, относительно курсора.

В любой момент можно сделать шаг назад при помощи Backspace.

7. ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

Программная часть проекта представляется ассемблерным кодом в среде CocolIDE.

Главная задача ассемблерной программы: контролировать игру и реализовывать логически-сложные ситуации, например, при запуске игры, CDM-8 запускается первым и очищает поле, подготавливает его к игре, далее игроку предлагается выбрать стартовую и конечную клетку.

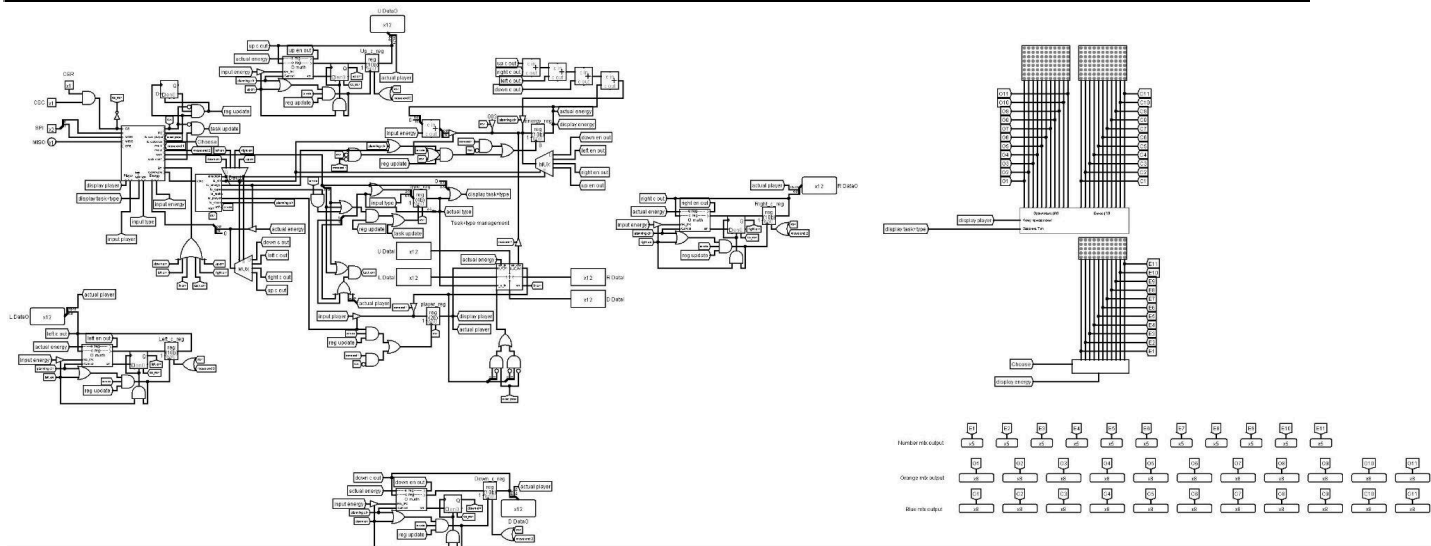
В начале игры код ассемблера отвечает за то, чтобы отправить команду второму контроллеру (Basic_controller) на разрешение игроку выбрать клетку, а затем уже CDM-8 эту клетку покрасит в нужный цвет и выдаст ей нужное количество энергии.

По ходу игры за генерацию заданий отвечает отдельный блок task_manager, ассемблер к нему обращается и запрашивает задание, блок отправляет это задание, а затем ассемблерный код даёт игроку выбор: брать задание или нет. В случае взятия задания, даёт игроку выбор как ставить данное задание. Затем снова ассемблерный код подаёт команду Basic_controller'у о том, что теперь можно перемещать энергию между клетками, после этого игрок начинает фазу перемещения. После завершения хода управление возвращается к CDM-8 (т.е. к ассемблерному коду) и он уже иницирует процедуру завершения хода и проверки выполнения задания. Также ассемблерный код “просыпается”, если мы пытаемся переместить энергию из специальной клетки (это нужно, чтобы, например, защитная клетка атаквала на 50% меньше). В итоге ассемблерный код управляет всем, раздаёт команды и обрабатывает тяжёлые случаи (например, как поместить задание, выполнил ли его игрок или нет) для этого ему помогают множество вспомогательных контроллеров.

8 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Аппаратная часть проекта представлена набором взаимосвязанных цифровых схем, созданных в Logisim.

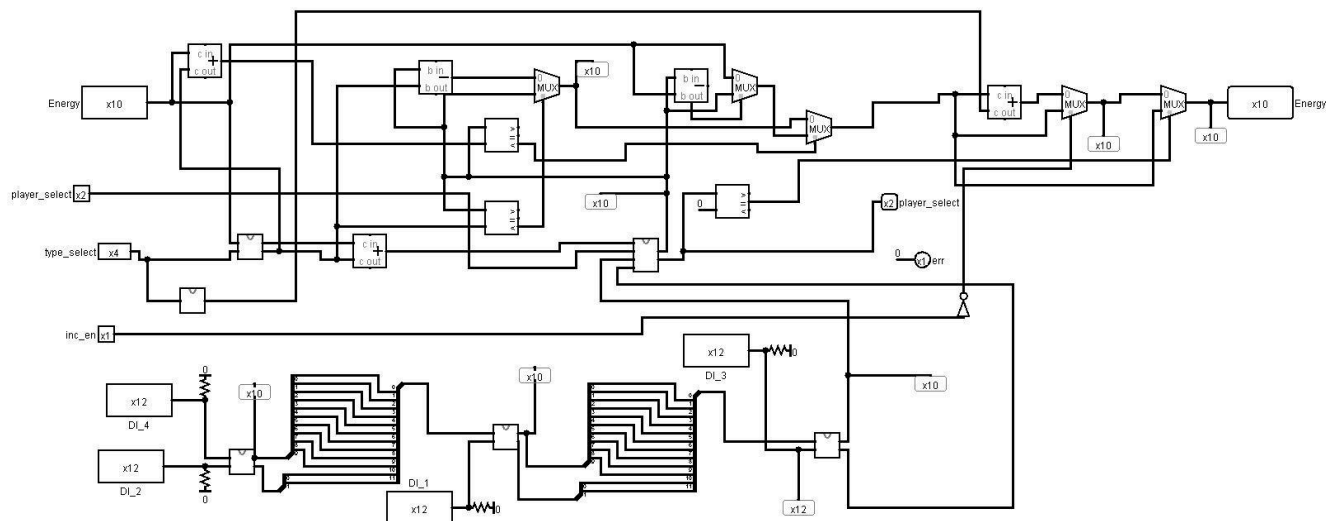
8.1 Cell_controller



Скриншот 1. “Схема cell_controller”

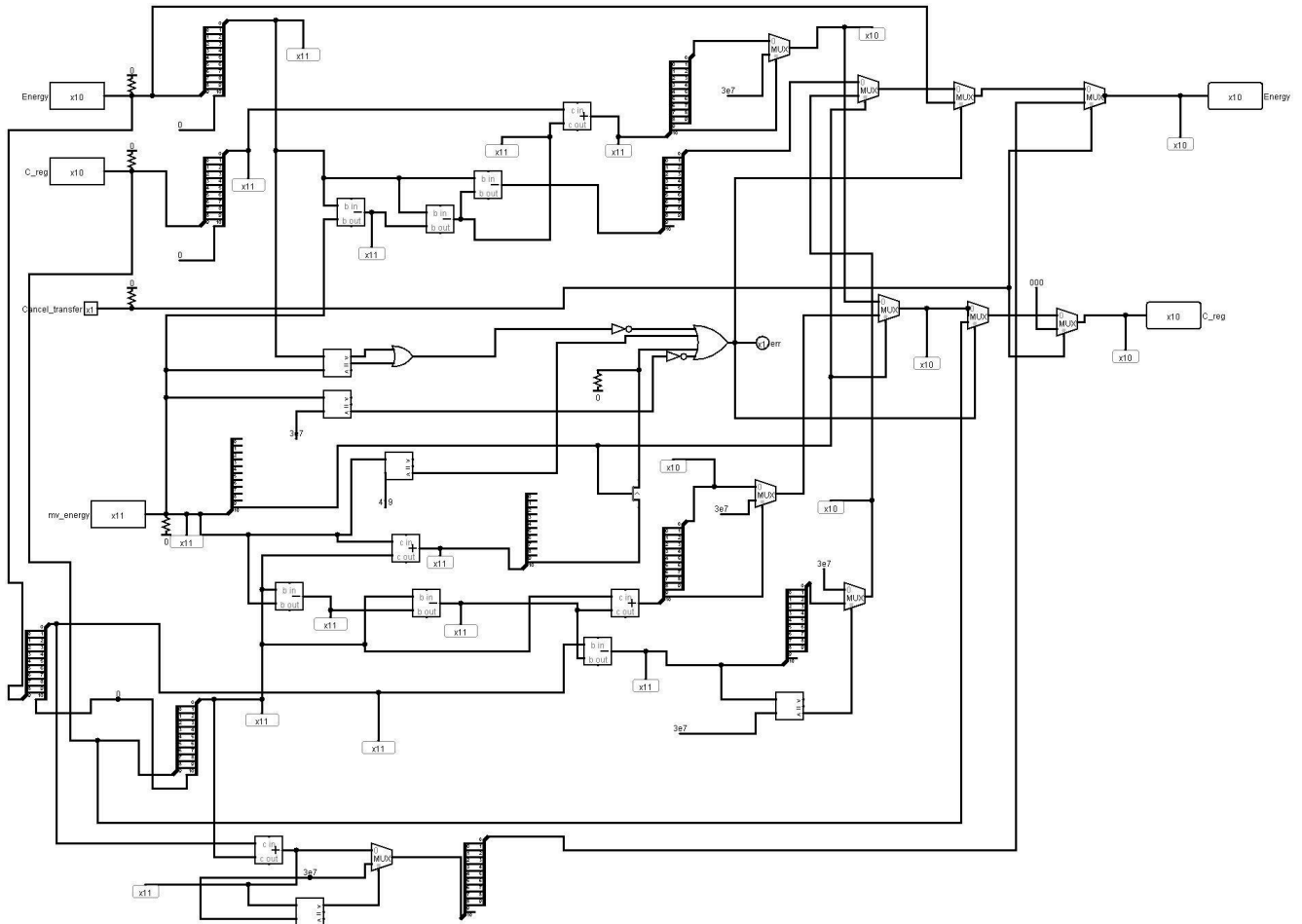
Данная схема позволяет показать тип клетки, количество энергии в ней, то как она взаимодействует с соседними клетками реализовано в подсхеме o_math_i_math

8.2 o_math_i_math



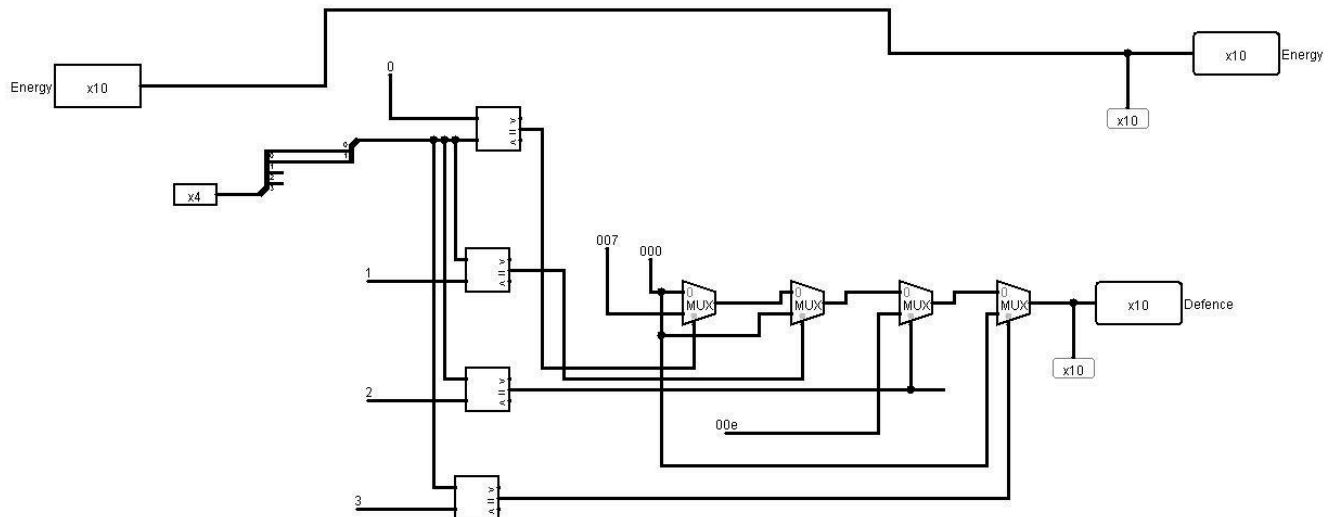
Скриншот 2: “Подсхема i_math”

Данная подсхема реализовывает логику того, как клетка принимает энергию из соседних, а также учитывает, какого она типа: экономическая, защитная, транспортная или обычная.



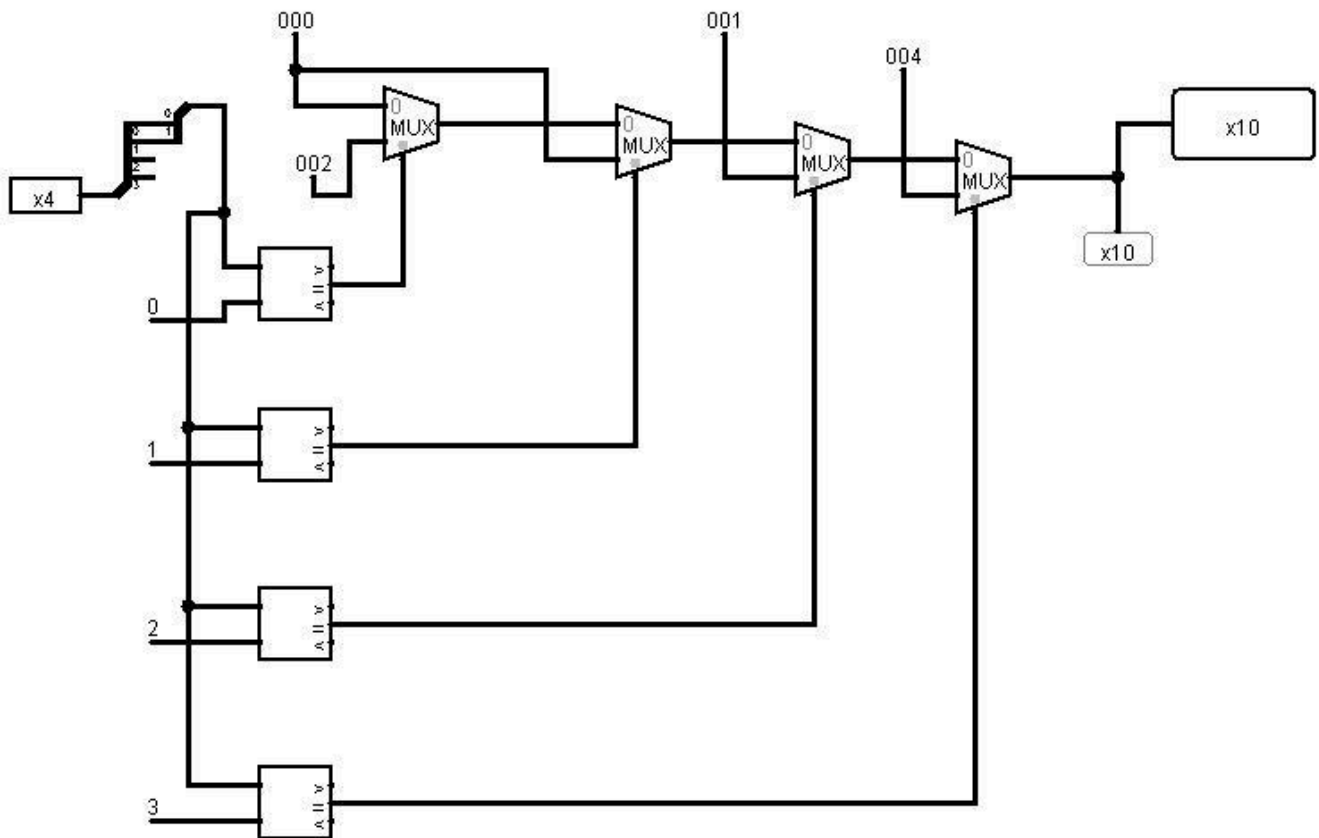
Скриншот 3: “Подсхема o_math”

Данная подсхема реализовывает логику того, как клетка перемещает энергию в соседние ей



Скриншот 4: “Подсхема type_math”

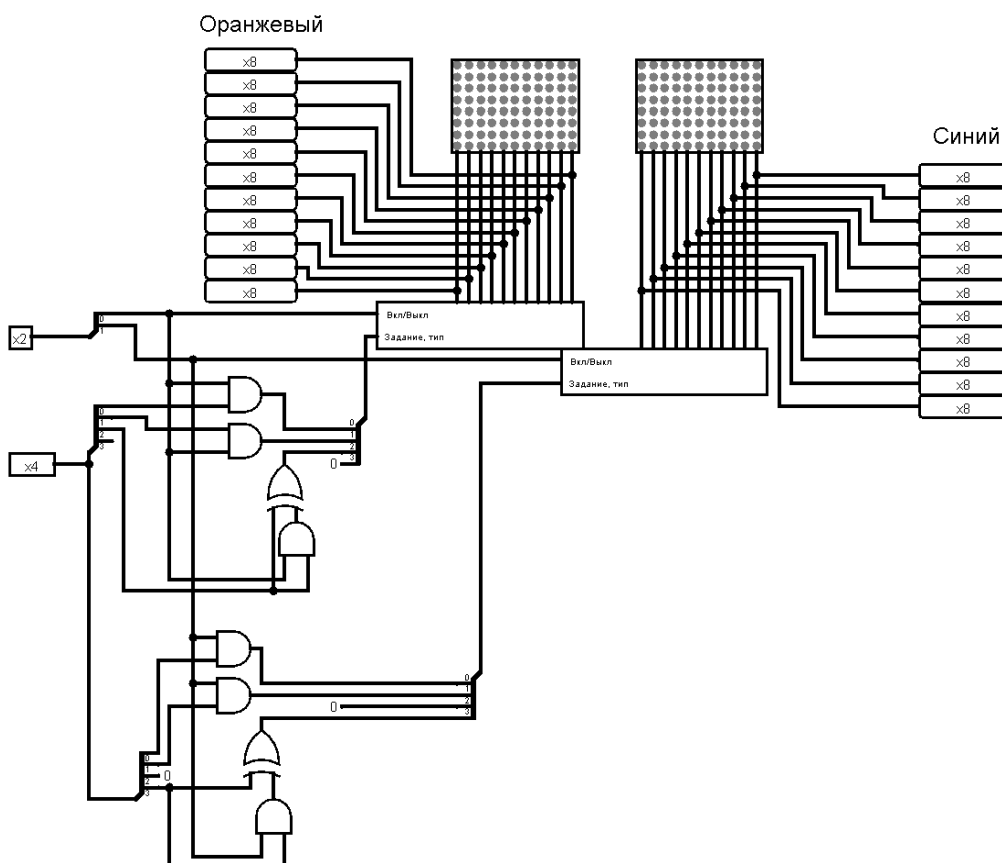
Подсхема реализует то, сколько защиты у клетки в зависимости от её типа



Скриншот 5: “Подсхема increment”

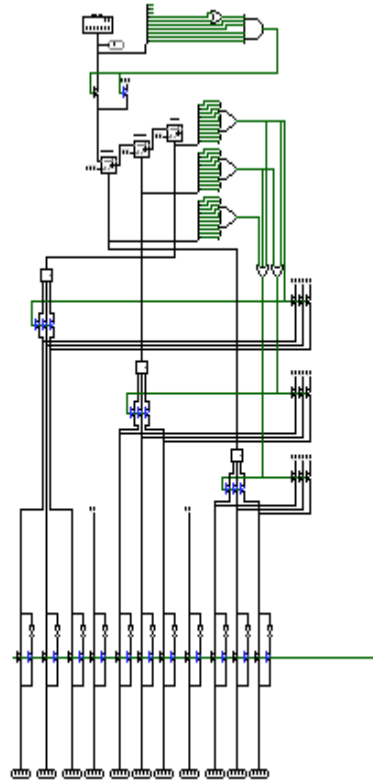
Логика подсхемы заключается в том, что в зависимости от типа клетки на выход выдаётся то количество энергии, которое прибавляется после фазы распределения

8.3. display_controller



Скриншот 6: “Схема Player_type”

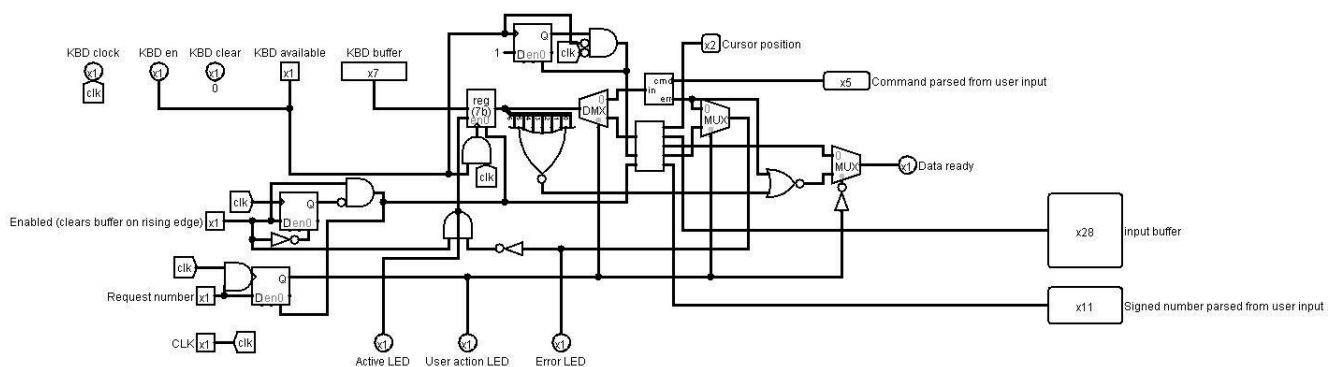
Данная схема реализует то, какому игроку принадлежит клетка: оранжевому или синему, является ли клетка заданием.



Скриншот 7: “Схема Numbers”

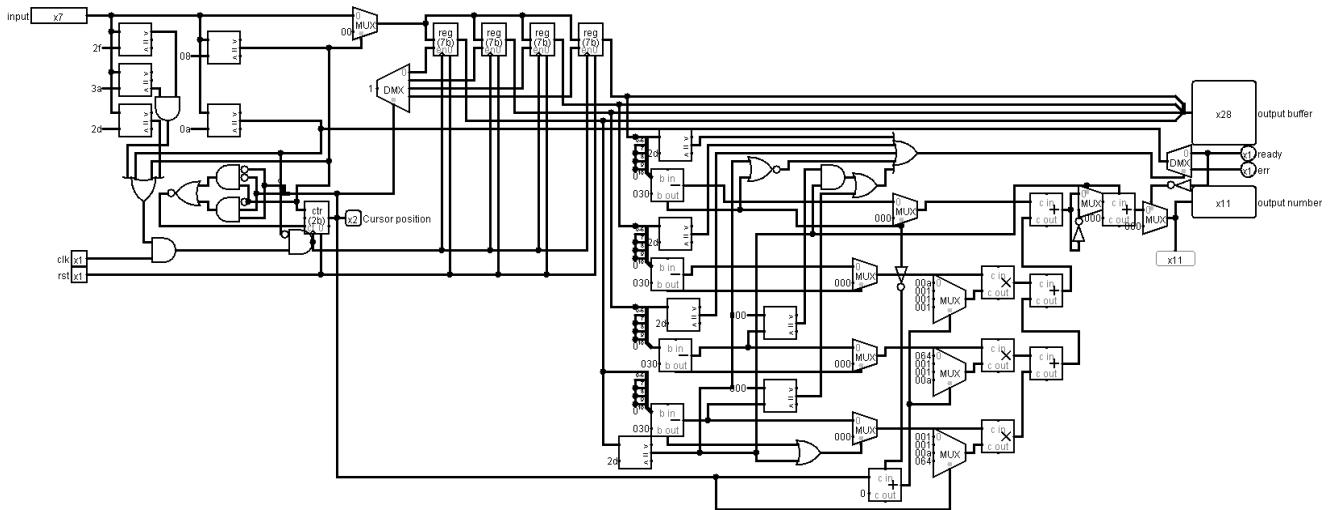
Реализует вывод цифр числа на экран матрицы, чтобы игрок мог знать, сколько энергии в клетке

8.4 input_controller

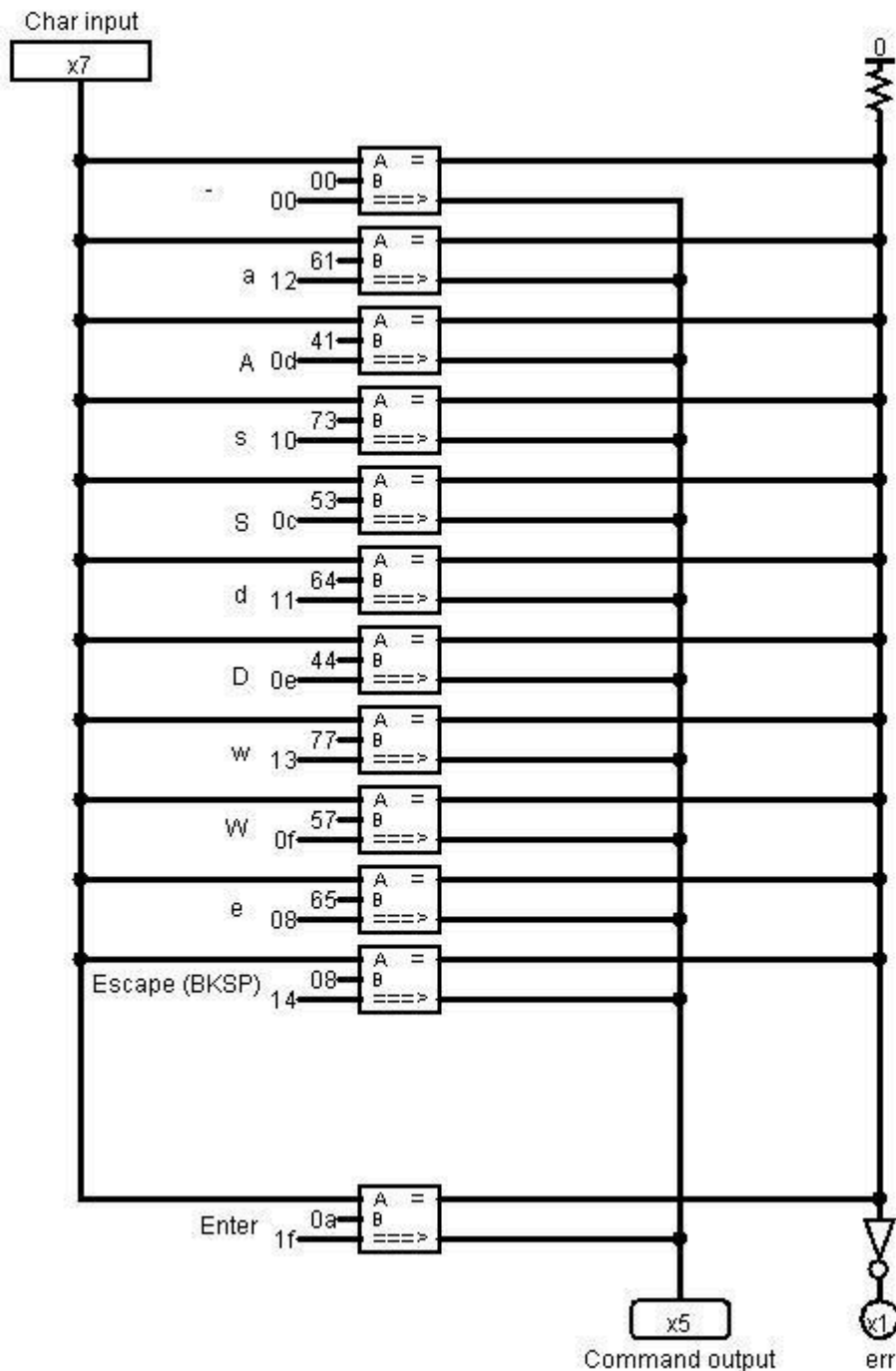


Скриншот 8: “Схема input controller”

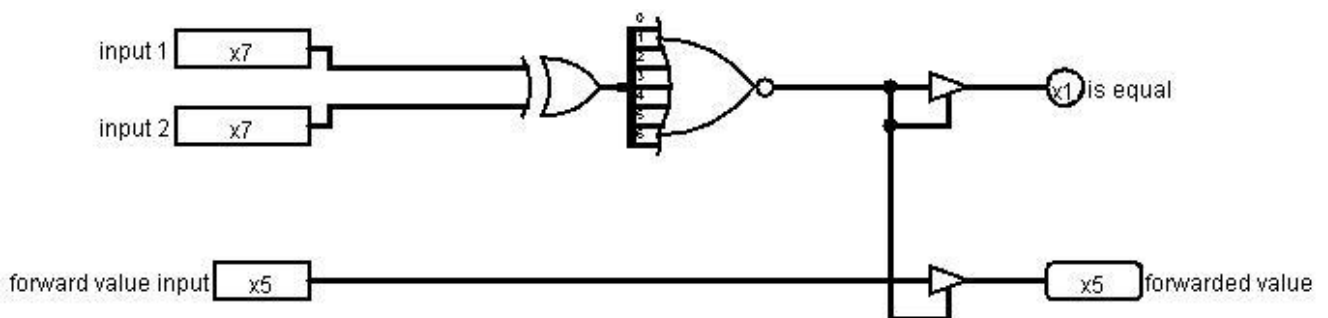
Схема реализует считывание ввода команд игроком с клавиатуры



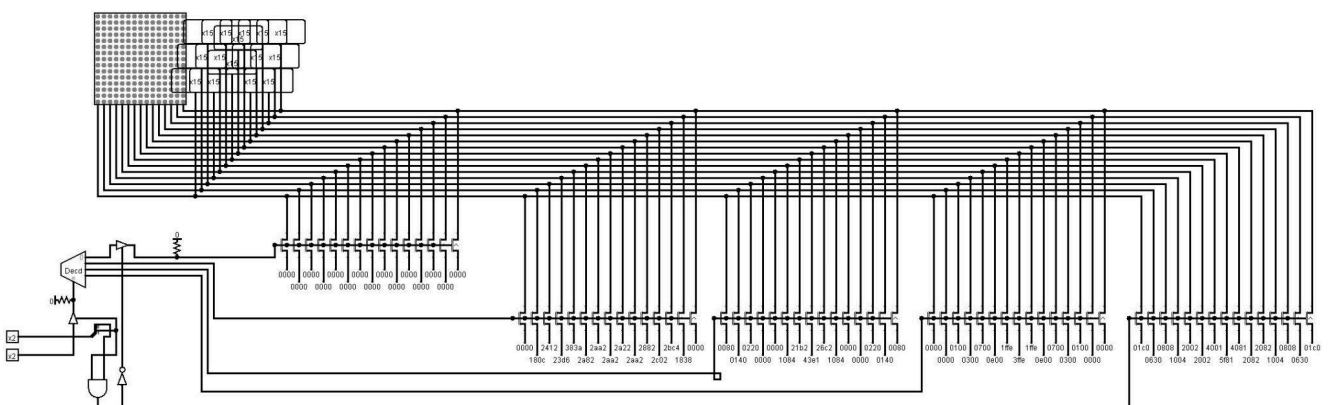
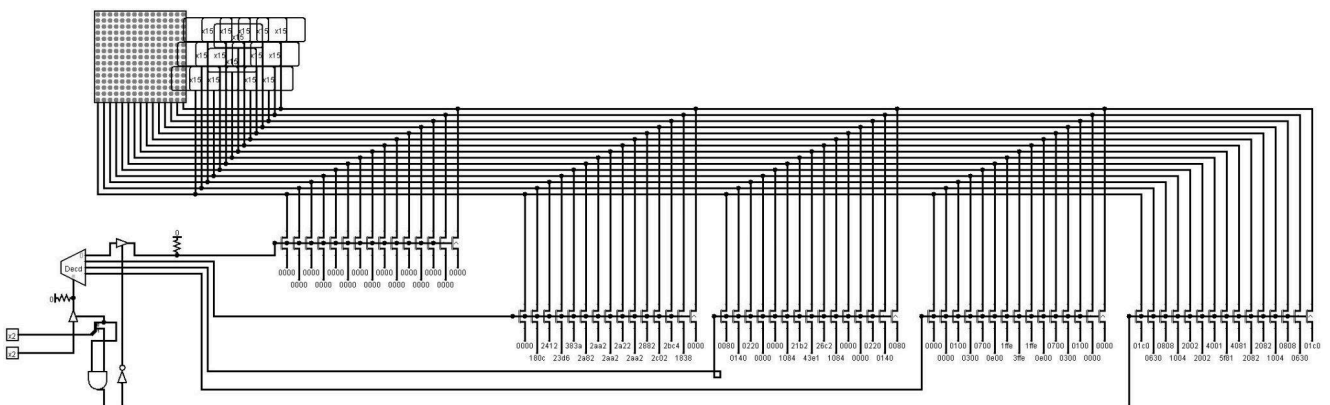
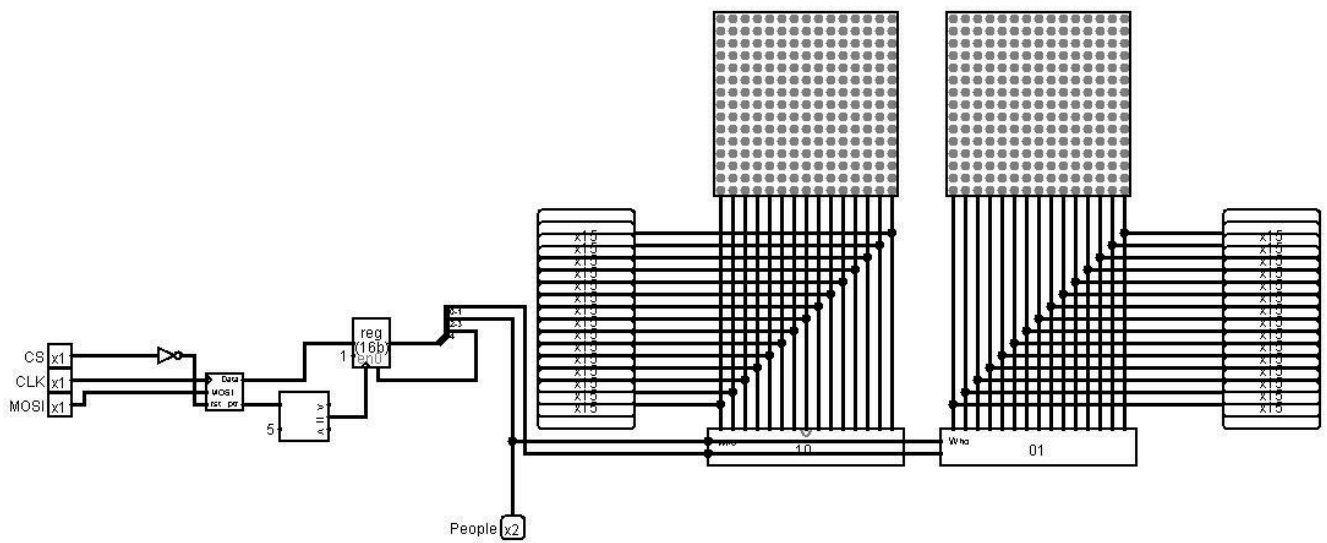
Скриншот 9: “number_parser”



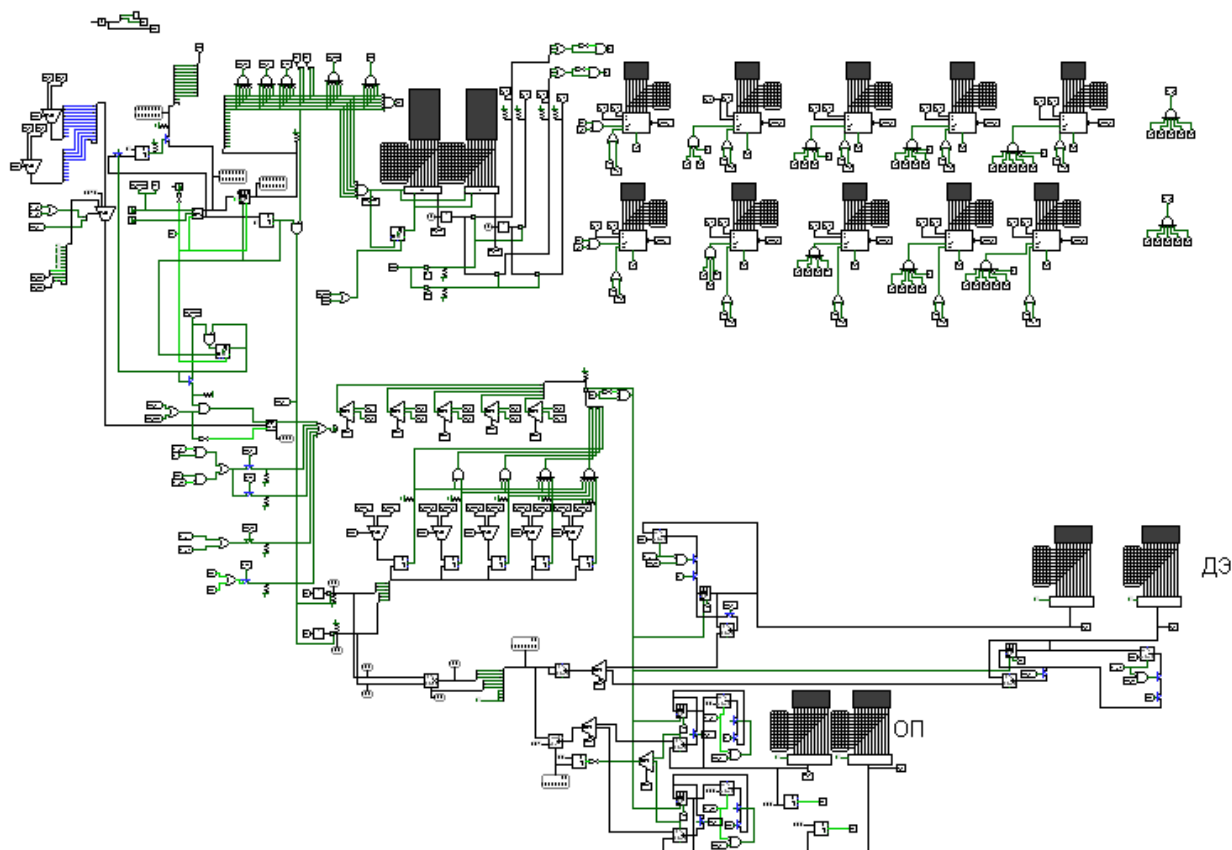
Скриншот 10: “Подсхема command_decoder”



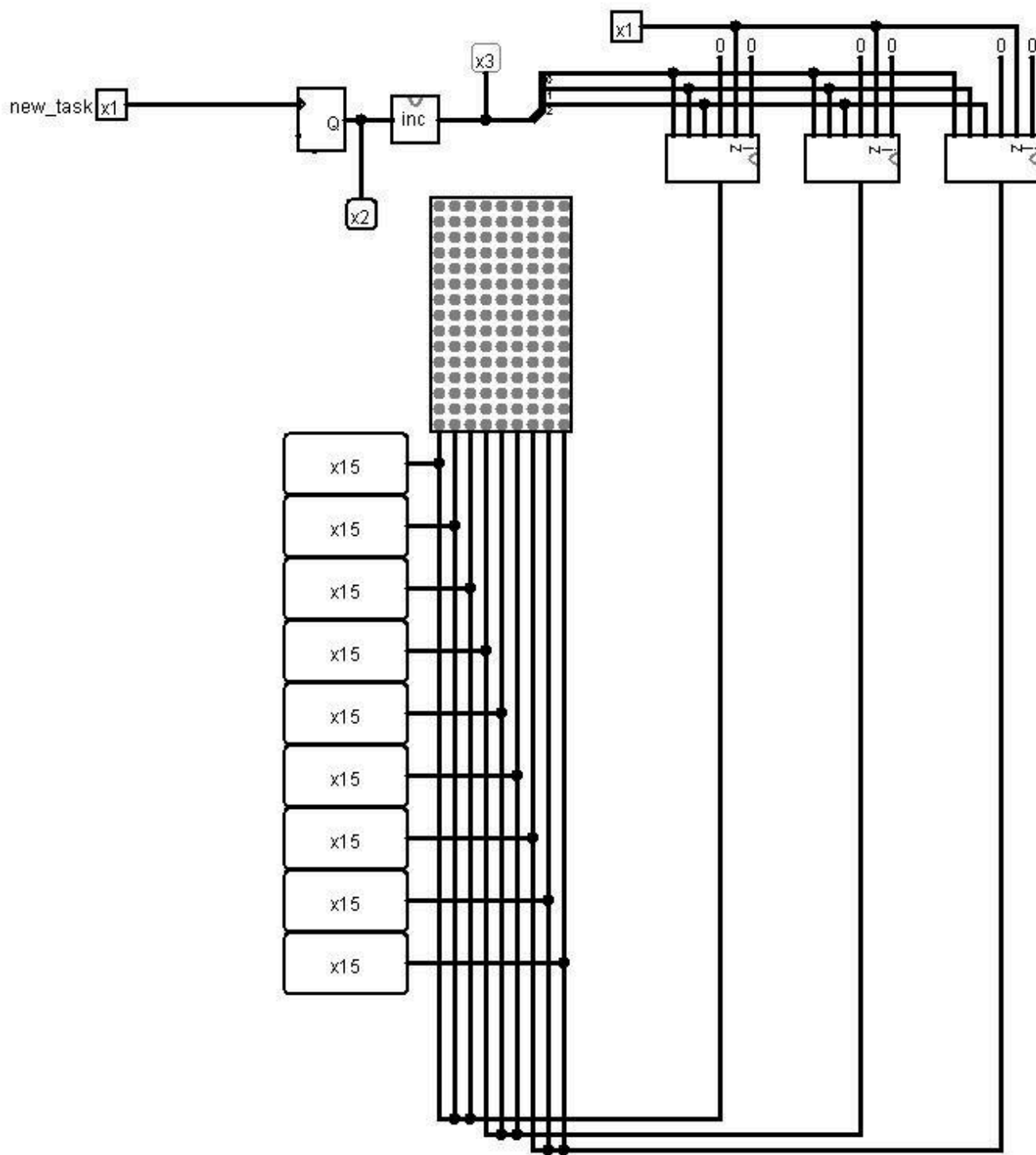
Скриншот 11: “Подсхема internal_comparator”



Скриншот 14: “Подсхема orange”

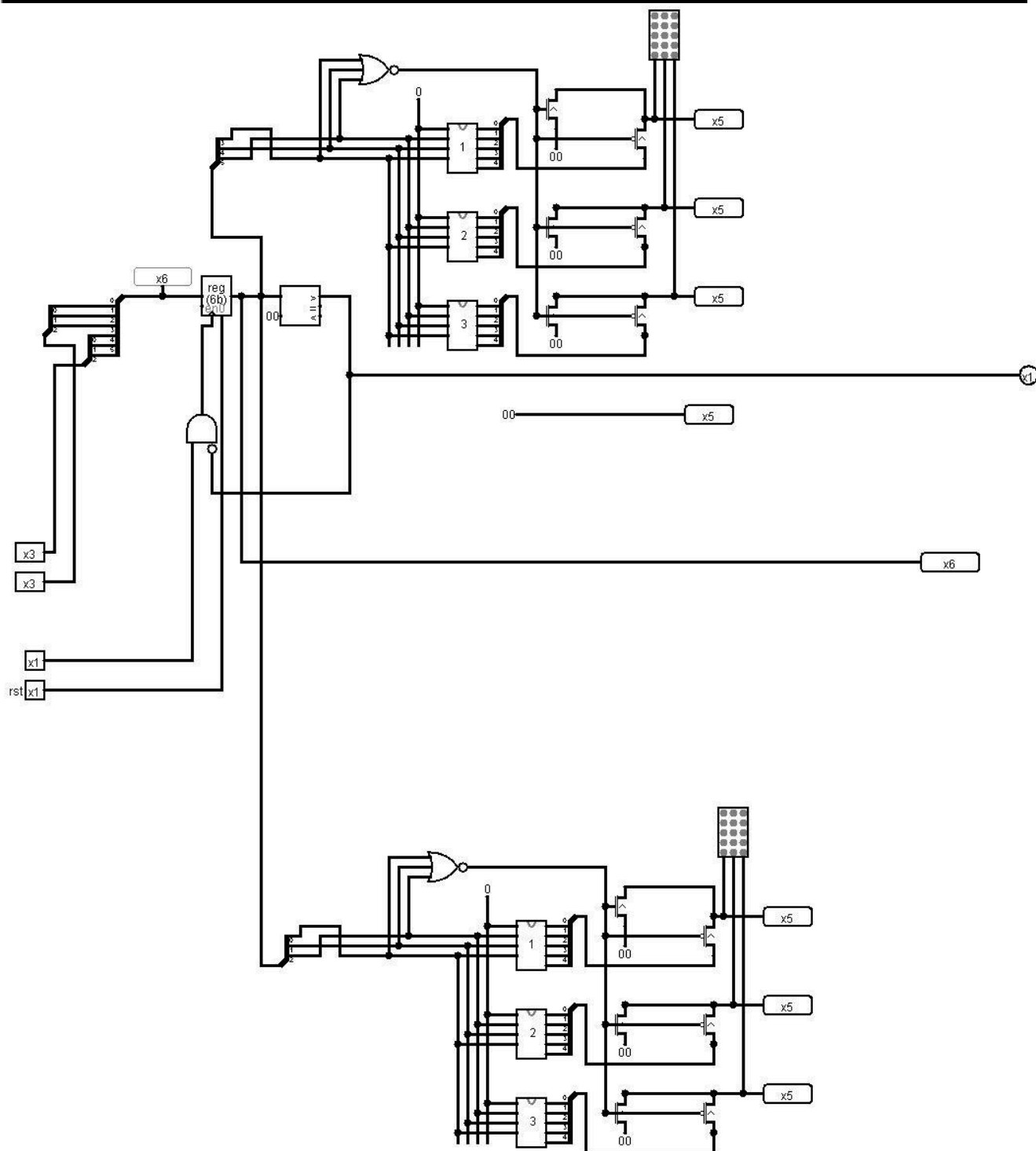


Скриншот 15: “Схема generator”

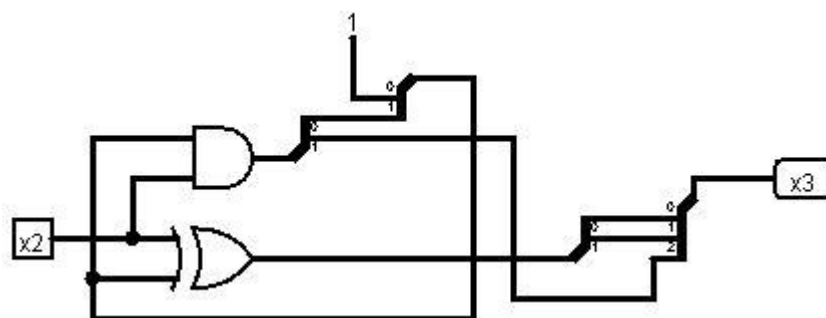


Скриншот 16: “Подсхема Weight”



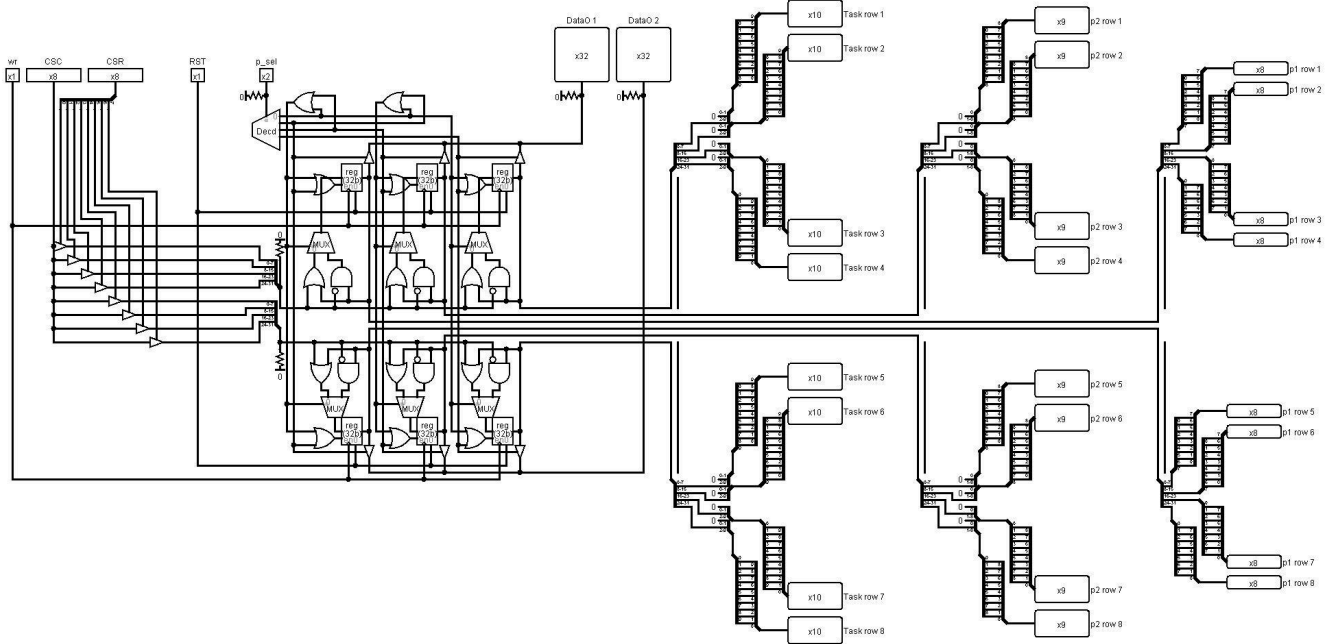


Скриншот 18: "Схема number_2"

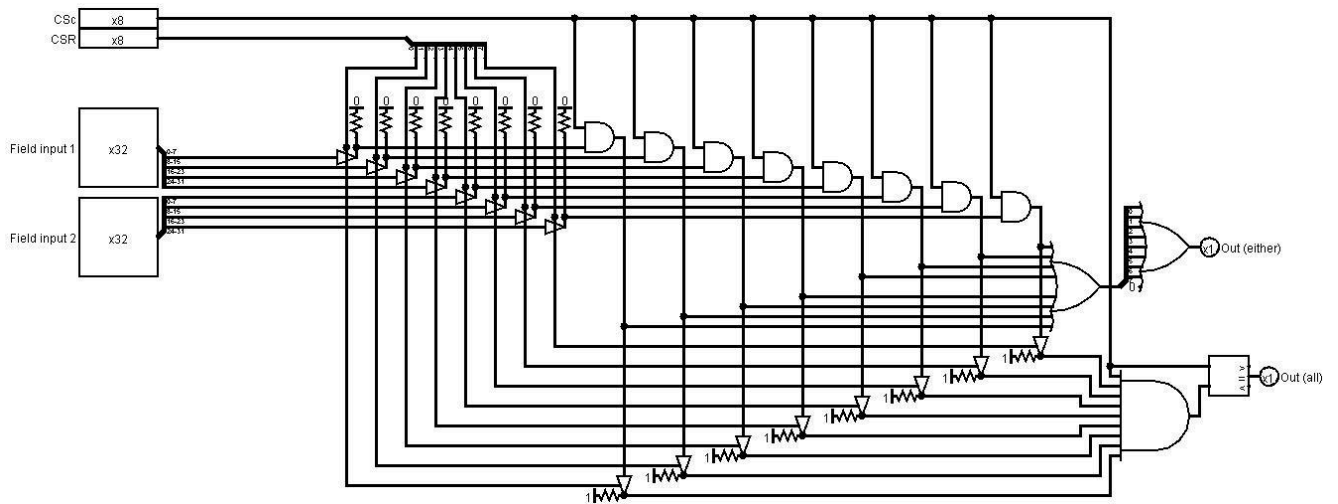


Скриншот 19: "Подсхема inc"

Скриншот 21: “Подсхема Ю dmx”

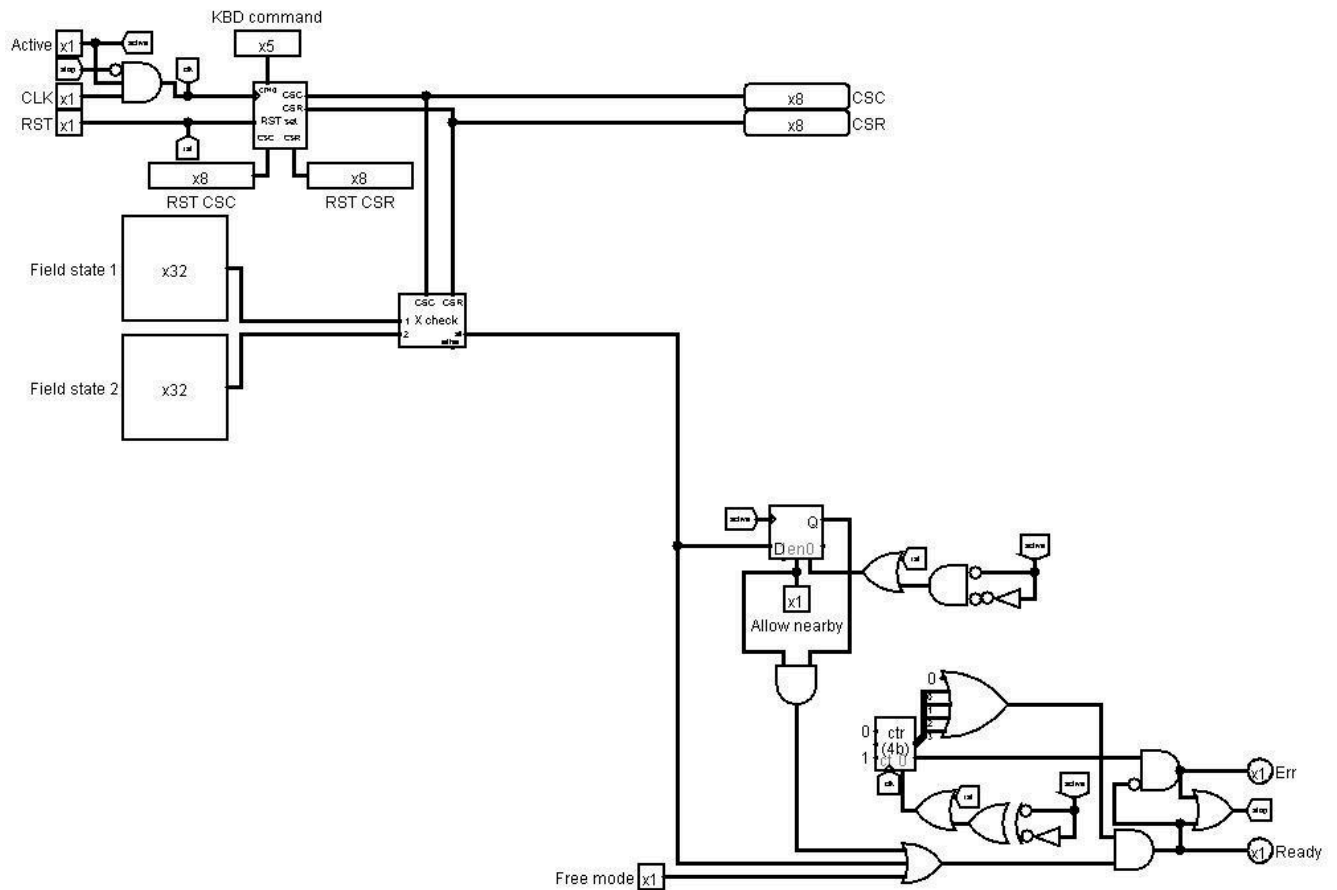


Скриншот 22: “Схема Field sahce”

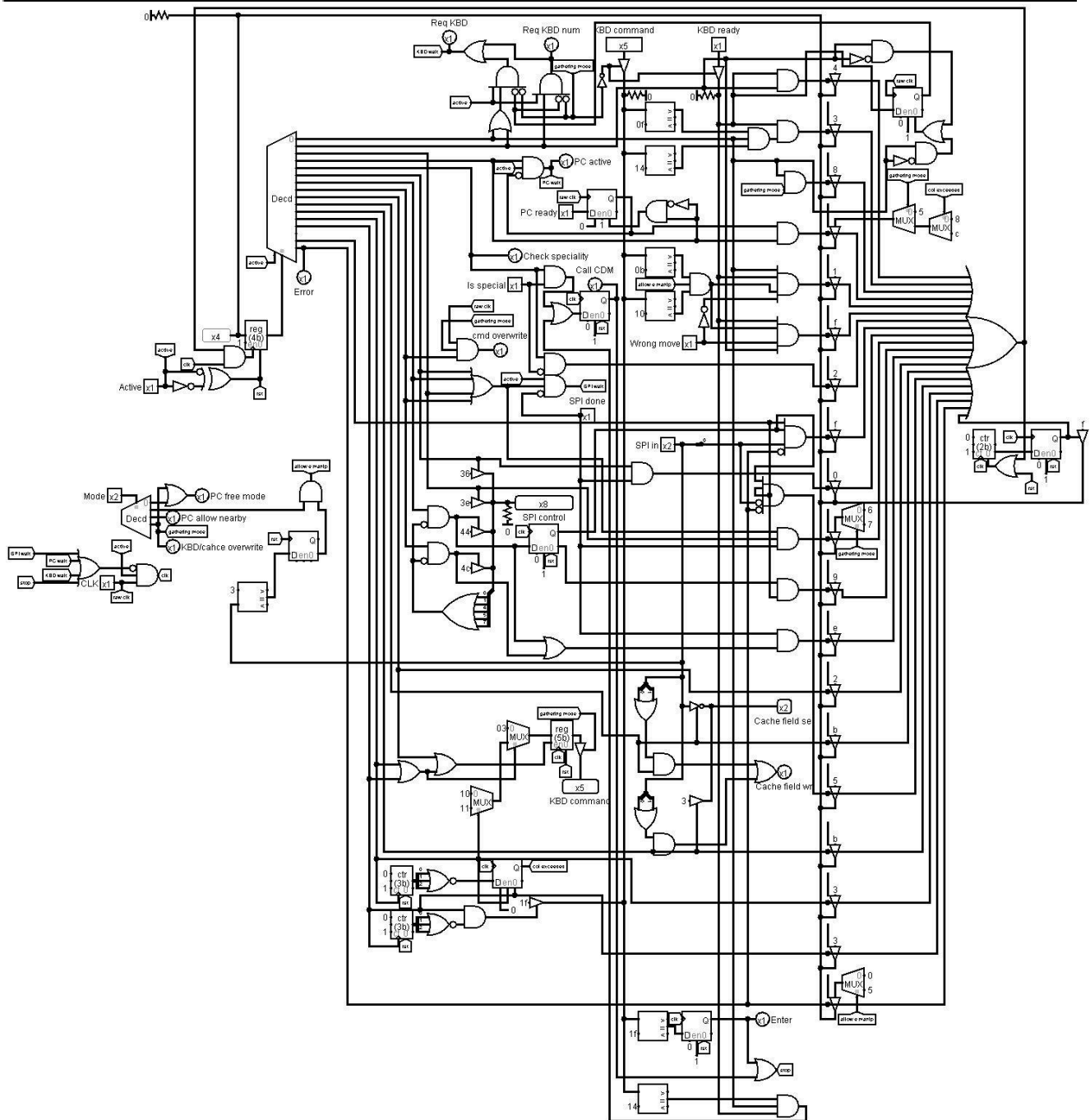


Скриншот 23: “Схема position checker”

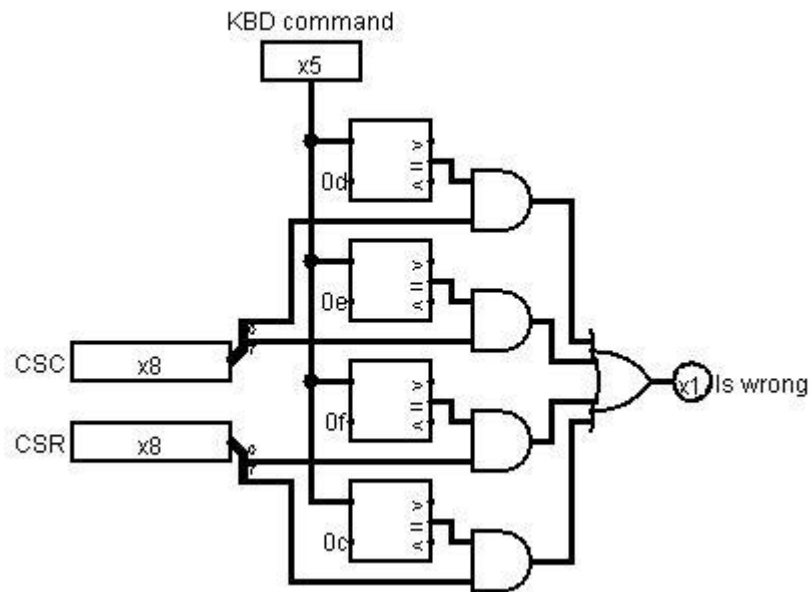




Скриншот 25: “Схема pointer controller”



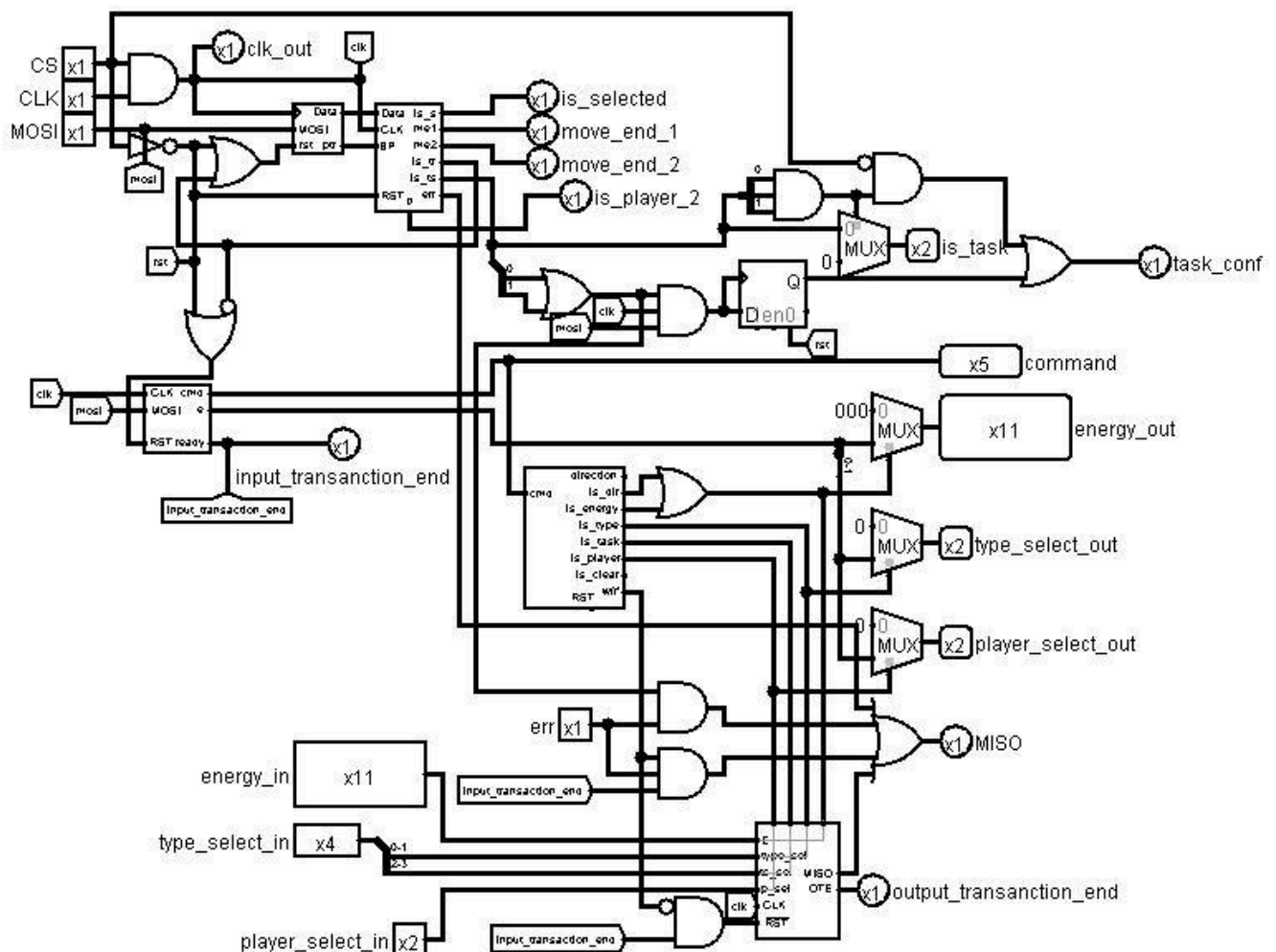
Скриншот 26: “Схема basic controller”



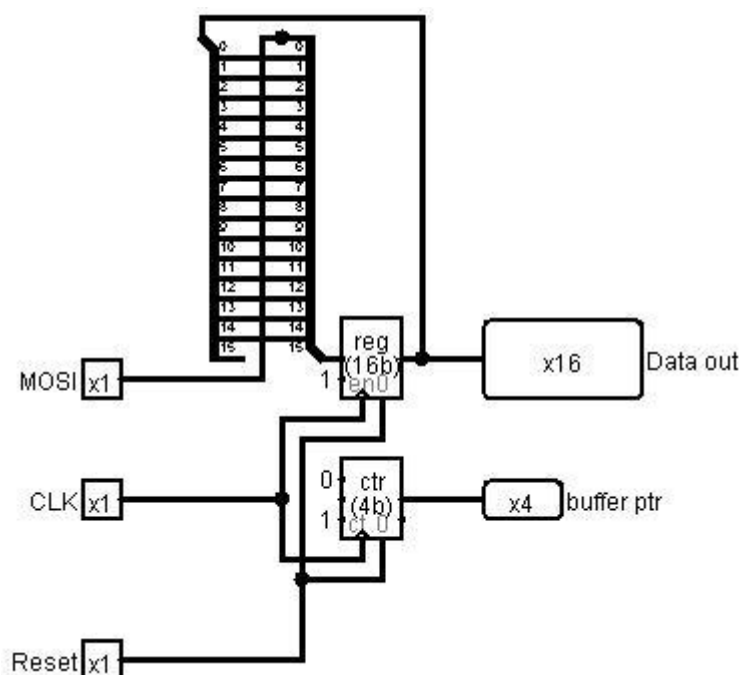
Скриншот 27: “Подсхема move checker”

8.7. SPI interface

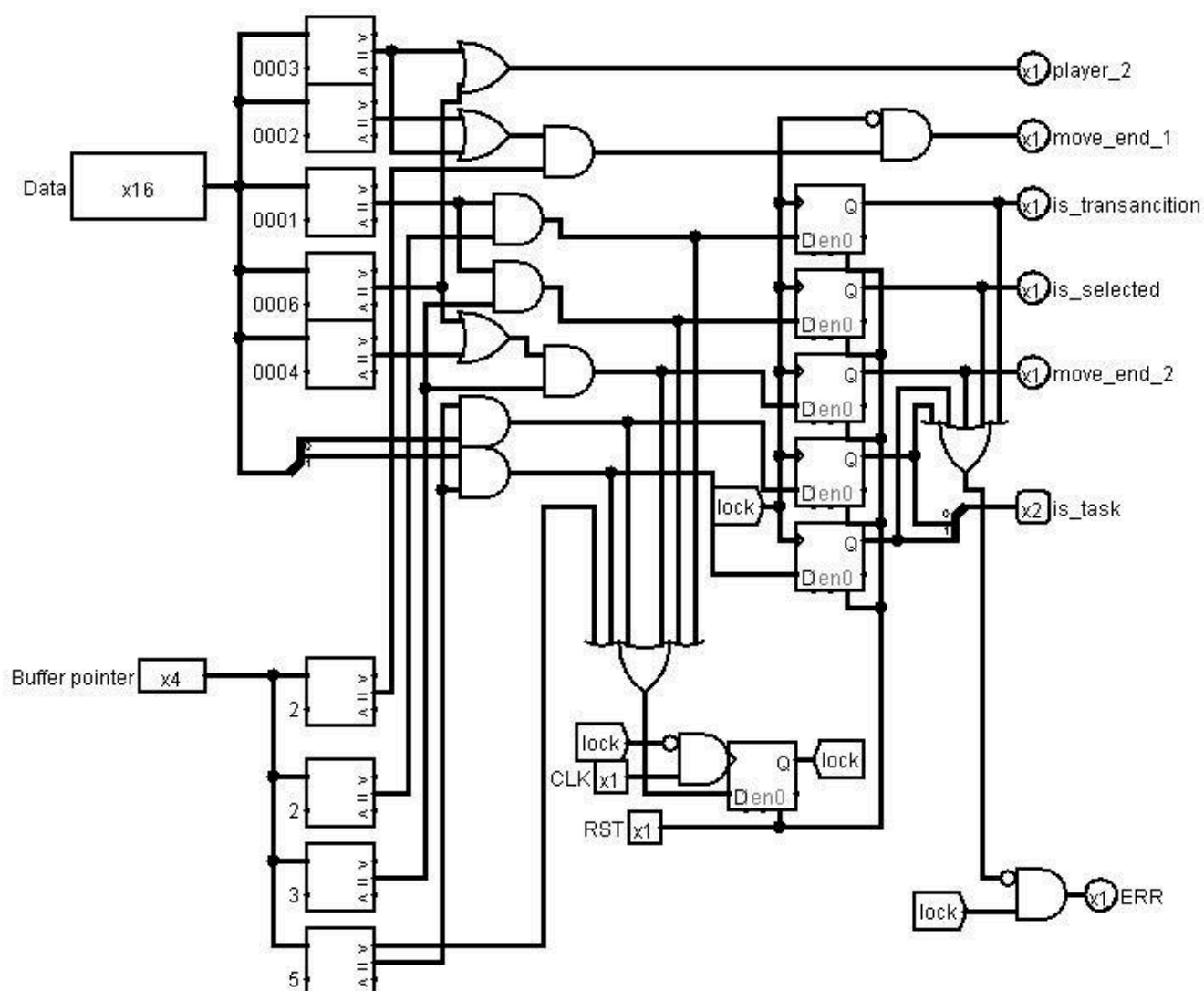
В данной схеме реализован последовательный периферийный интерфейс (SPI) для обеспечения связи между микроконтроллером и клетками



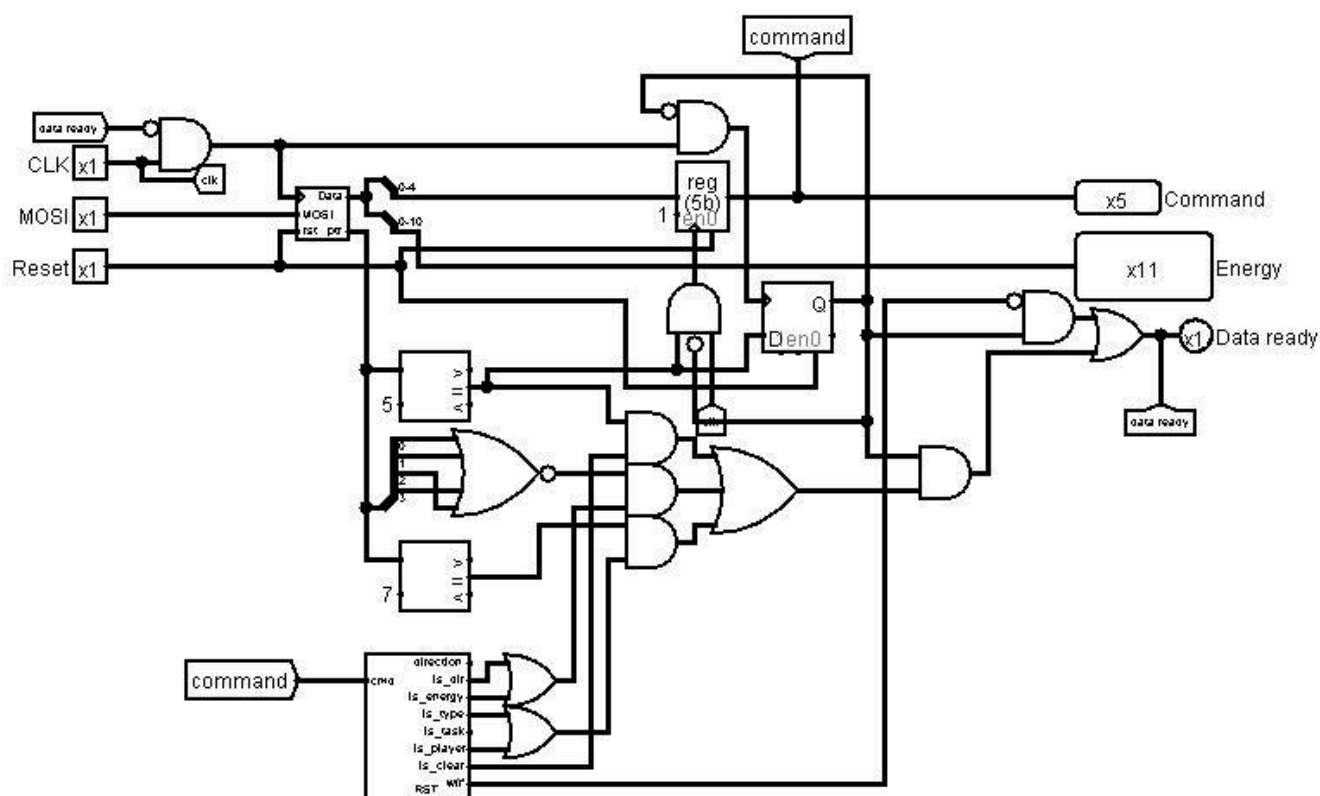
Скриншот 28: “Схема SPI module (cell)”



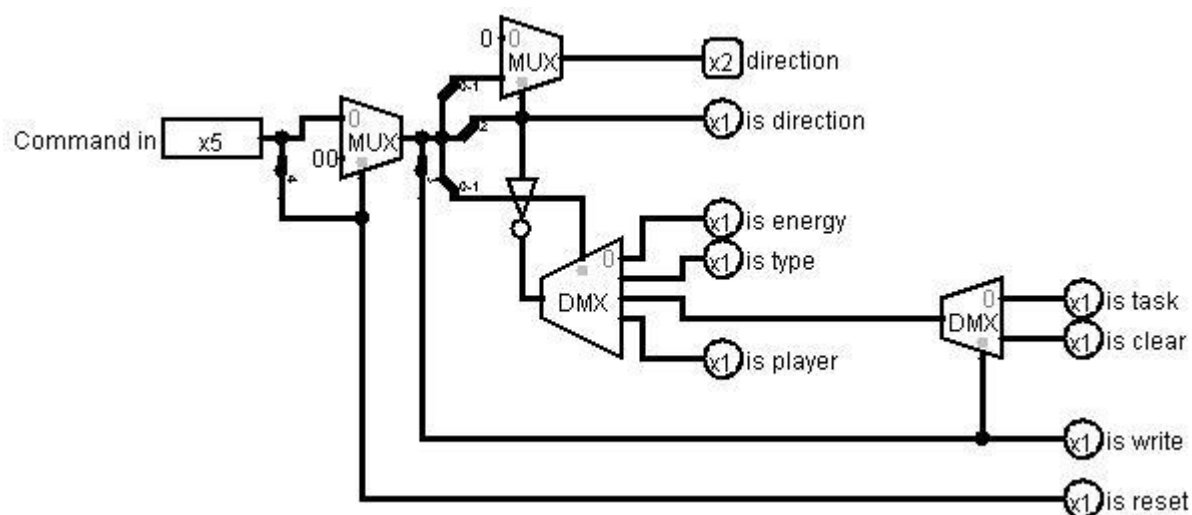
Скриншот 29: “Подсхема input shifter”



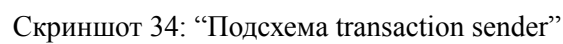
Скриншот 30: “Схема special sequence decoder”

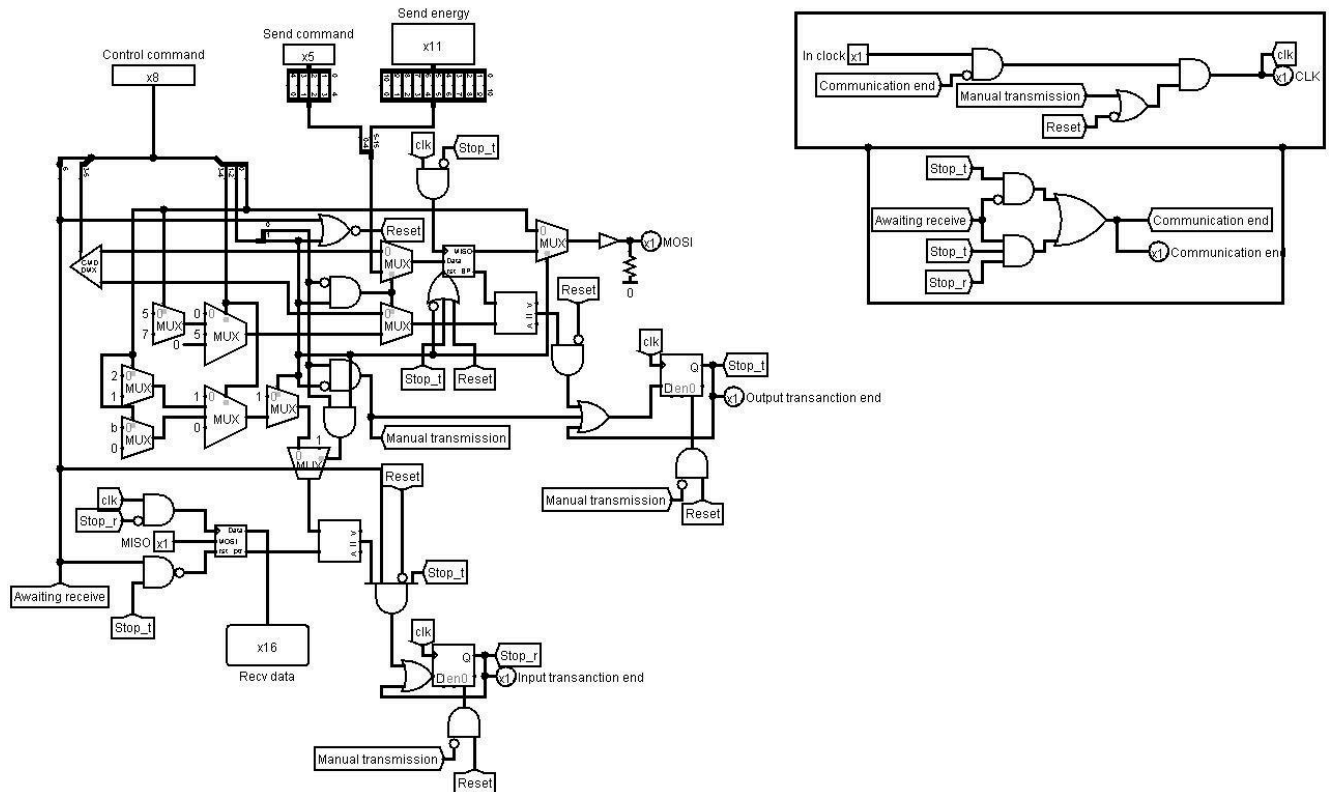


Скриншот 31: “Схема transaction parser”

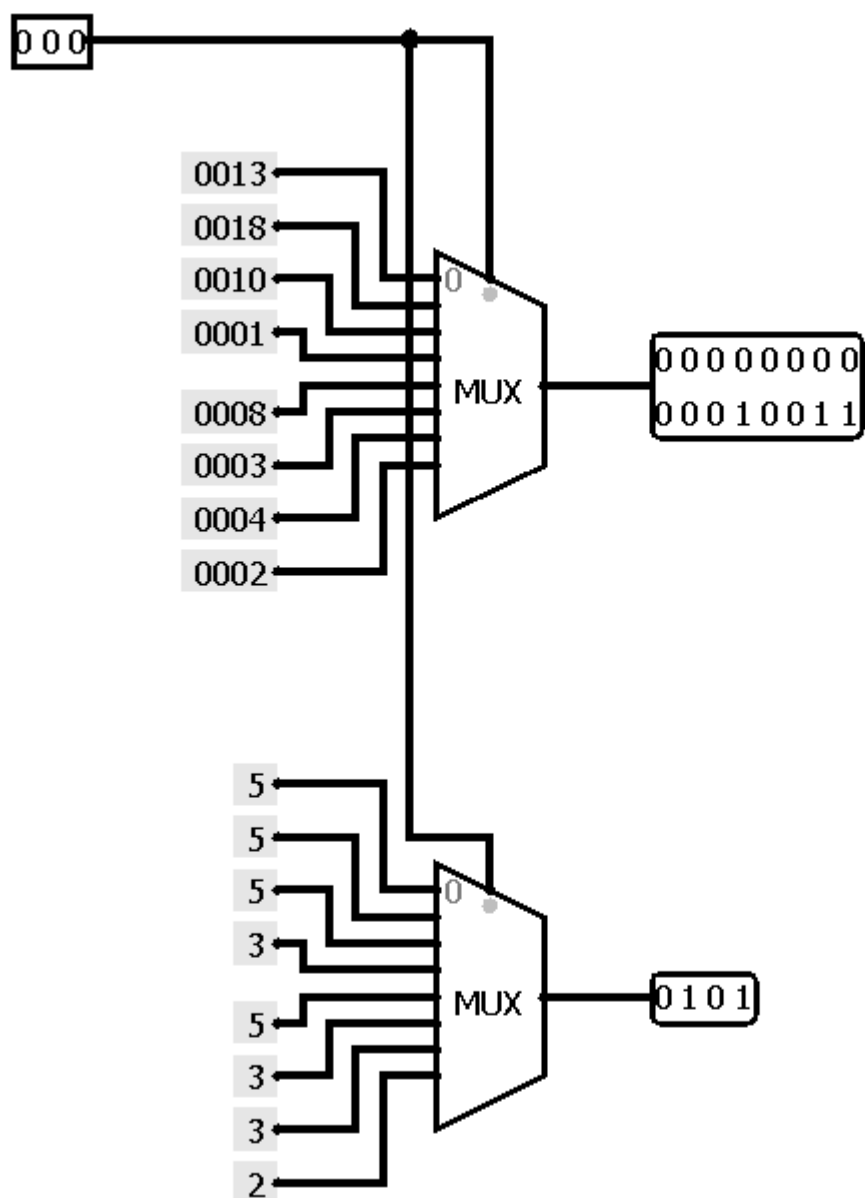


Скриншот 32: “Подсхема command decoder”



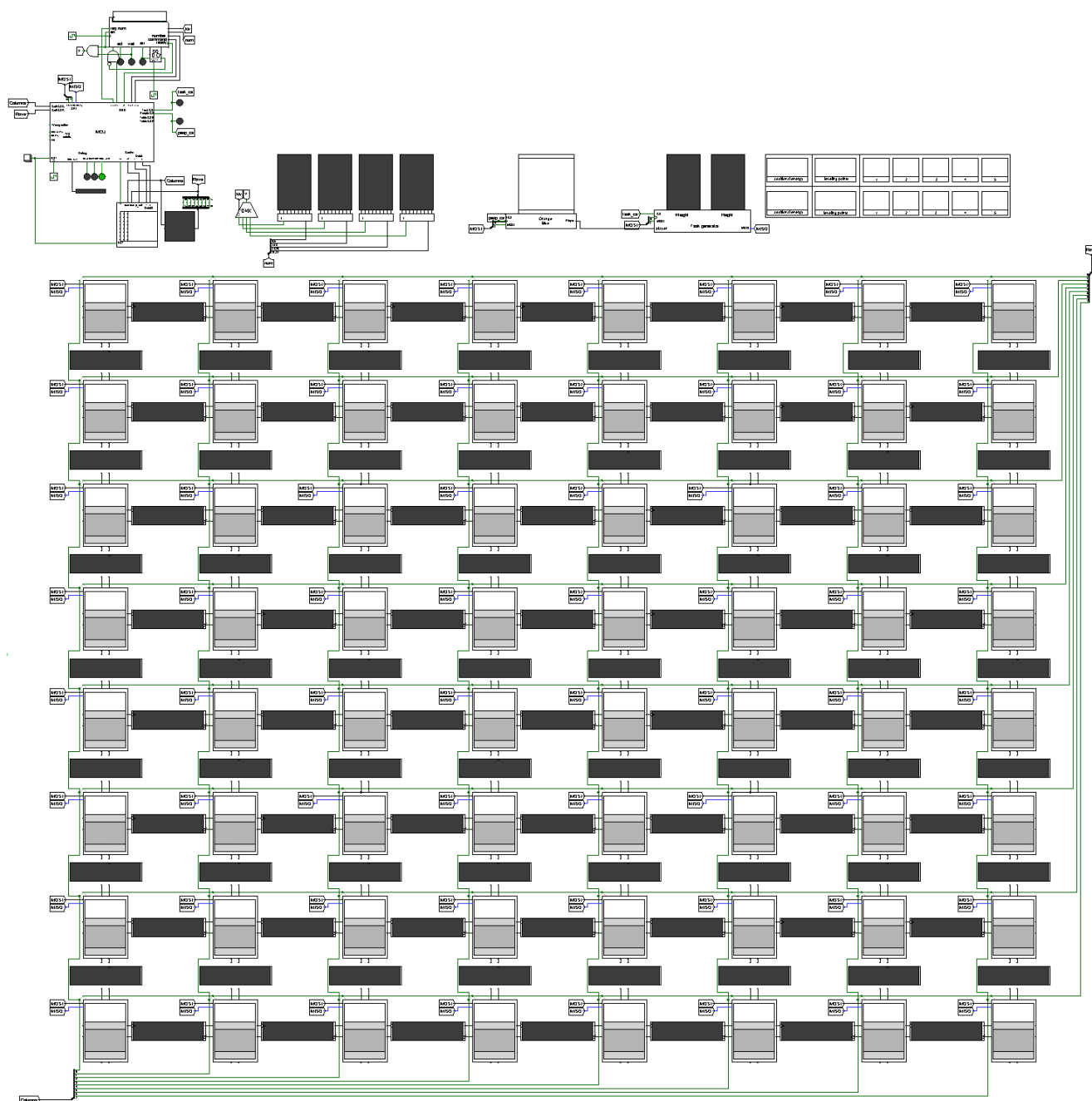


Скриншот 35: “Схема SPI module (master)”



Скриншот 36: “Подсхема predefined command selector”

8.8. Main



Скриншот 37: “Схема Main”

В данной схеме находится сама игра “Rectify”, здесь реализовано поле 8 x 8 клеток, а также взаимодействие игроков с полем

9 ЗАКЛЮЧЕНИЕ

В ходе проекта была создана электронная версия игры "Rectify" на микроконтроллере CdM-8 с использованием Logisim и ассемблера. Успешно реализованы: логическая схема игры, программное обеспечение, тестирование и документация. Проект демонстрирует практическое применение цифровой логики и низкоуровневого программирования, представляя ценность для образовательных целей. Все задачи выполнены, полученный опыт может быть использован в будущих разработках.

10 ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

В работе над проектом использовались материалы курса CS/MR Digital platforms 2024/25, размещенные в Google Classroom.

- tome.pdf