ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

15.03.06 - Мехатроника и робототехника
Профиль: Искусственный интеллект

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Тема работы: **‘DINOSAUR’**

Гриценко Тимофей Сергеевич, 24942
Сальникова Ксения Васильевна, 24942
Смирнова Анастасия Павловна, 24942

Новосибирск
2025

Ошибка! Закладка не определена.

1 ВВЕДЕНИЕ 3
2 ЦЕЛЬ И ОБЛАСТЬ ПРИМЕНЕНИЯ 3
3 ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ 3
4 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 5
4.1 interface 6
4.2 main 6
4.3 nickname 8
4.4 cdm-8 9
4.5 magic-printer 10
4.6 counter 11
4.7 bcd-counter 12
4.8 bcd-add 13
4.9 timer 14
4.10 6-counter 14
4.11 10-counter 15
4.12 level 16
4.13 cmp 17
4.14 speed 18
4.15 speeddino 18
4.16 cactus 18
4.17 big-cactus 19
4.18 small-cactus 20
4.19 wide-cactus 21
4.20 obstacles 21
4.21 bird-wings-up 22
4.22 bird-wingsdouwn 23
4.23 birds 23
4.24 random-spawn 24
4.25 shiftreg 25
4.26 keyboard-handler 26
4.27 dino-jump 27
4.28 dino 28
4.29 pause 29
4.30 gameover 29
4.31 invertor 30
5 ЗАКЛЮЧЕНИЕ 30
6 ИСТОЧНИКИ 30

1 ВВЕДЕНИЕ

В рамках курса CS/MR Digital platforms 2024/25, студентам необходимо разработать игру, используя логические схемы и процессор CdM-8.

«DINOSAUR» — это проект, целью которого является воссоздание знаменитой игры Dino из интернет-браузера Google Chrome. Dino – это браузерная игра, в которой игрок управляет бегущим динозавриком, перепрыгивая появляющиеся на пути препятствия. Цель игры – набрать как можно большее количество очков.

Особенность реализации в том, что игра создаётся с помощью Logisim — образовательного инструмента для моделирования цифровых схем. Главная задача проекта — показать, как на базе базовых логических элементов и цифровых платформ можно построить полноценную интерактивную игру.

2 ЦЕЛЬ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Программа «DINOSAUR» решает задачу моделирования известной игры средствами цифровой логики. Основная идея — продемонстрировать работу логических элементов и принципов цифровых платформ через знакомый игровой процесс.

Данная программа будет полезна в образовательных целях — для изучения цифровой логики, разработки цифровых схем и наглядной демонстрации работы интерактивных систем. Она представляет интерес для студентов и преподавателей курсов по цифровым платформам, а также для всех, кто хочет увидеть практическое применение теоретических знаний в создании игровых проектов с использованием Logisim.

3 ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

Для реализации программной части нашей игры мы использовали язык ассемблера процессора CdM-8 в интегрированной среде разработки CocoIDE, которая специально предназначена для разработки кода, который будет выполняться этим процессором.

Процессор используется для сортировки рекордов игрока.

Процессор ждет введенного от пользователя никнейма (до 4 символов). После открывает доступ к игре. По завершении игры, очки делятся на два 8-битных числа и передаются в процессор. После этого процессор выполняет сортировку пузырьком – в зависимости от значений очков сортируются адреса, указывающие на начало соответствующего никнейма. В конце выводится отсортированный список по возрастанию, обновляются все поля со счетчиками, и процессор снова ждет введенного от пользователя никнейма.

Ниже представлена подробная блок-схема алгоритма.

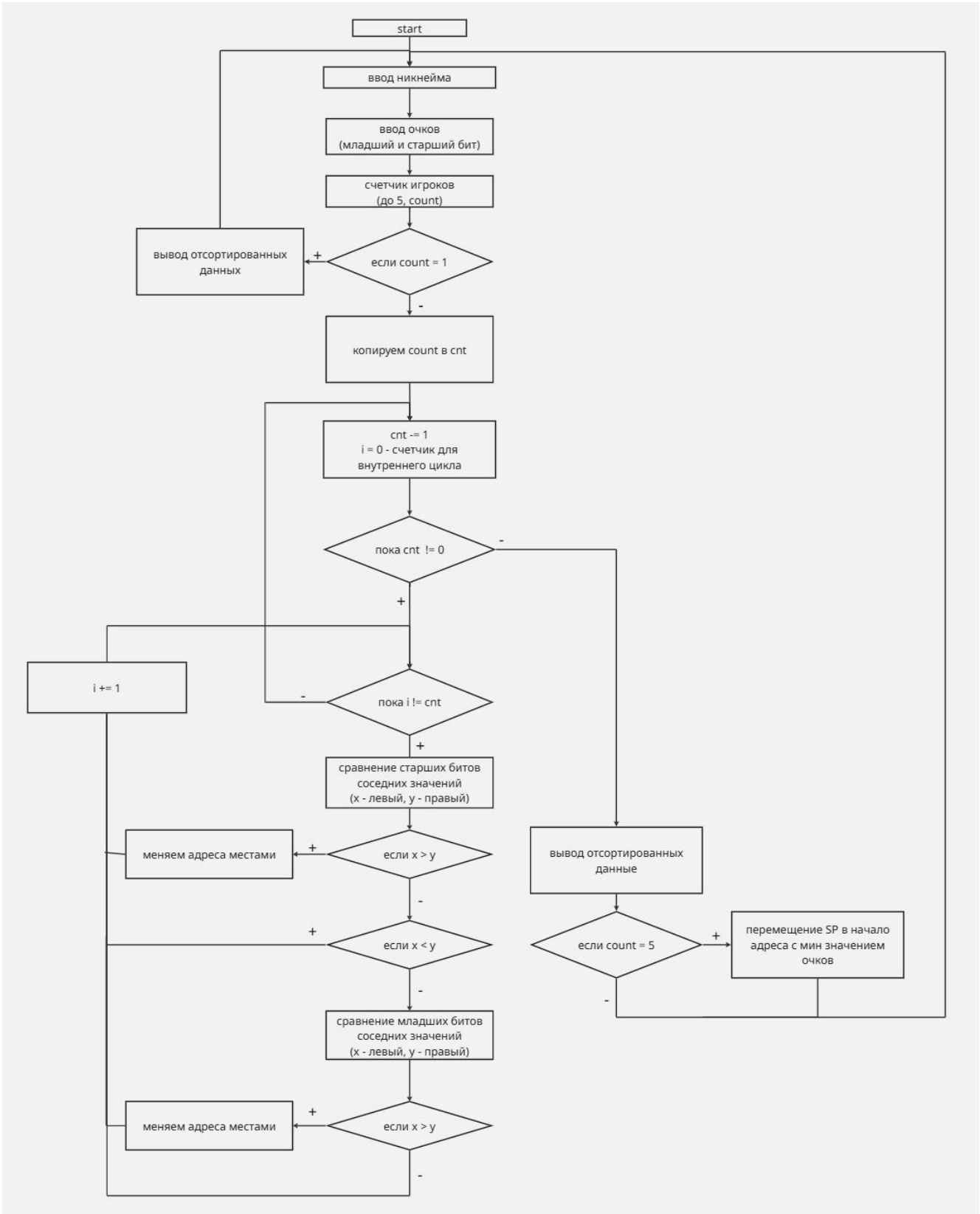


Схема 1. «Алгоритм сортировки»

4 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Аппаратная часть проекта представлена набором взаимосвязанных цифровых схем, созданных в Logisim. Ниже расположена диаграмма связей между различными подсхемами проекта «DINOSAUR».

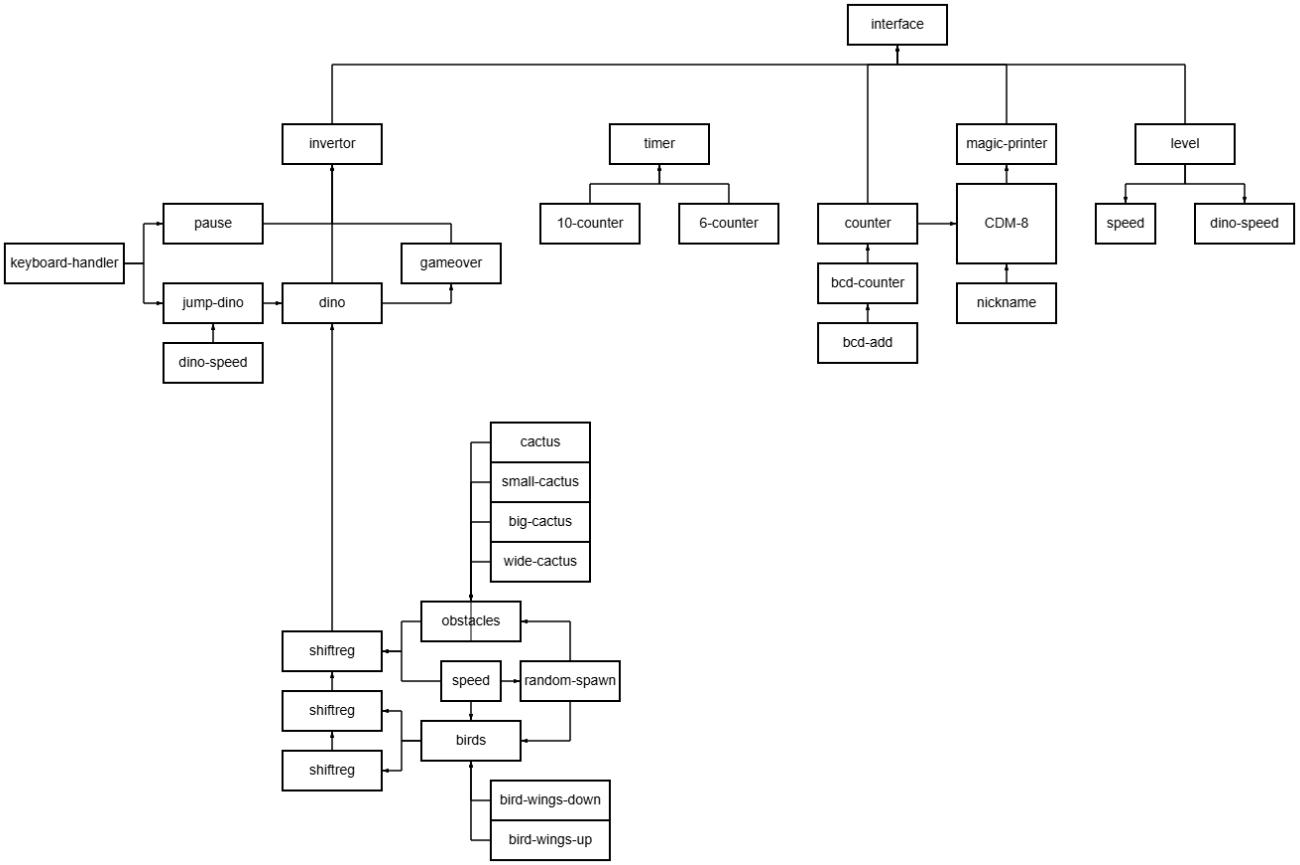
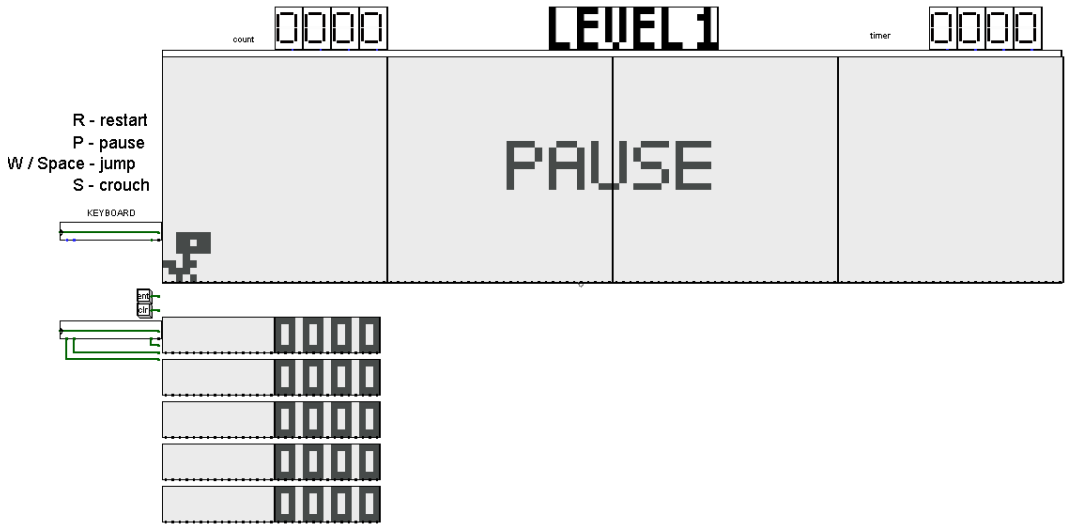


Схема 2. «Связи подсхем проекта»

4.1 interface



Скриншот 1. «Схема interface»

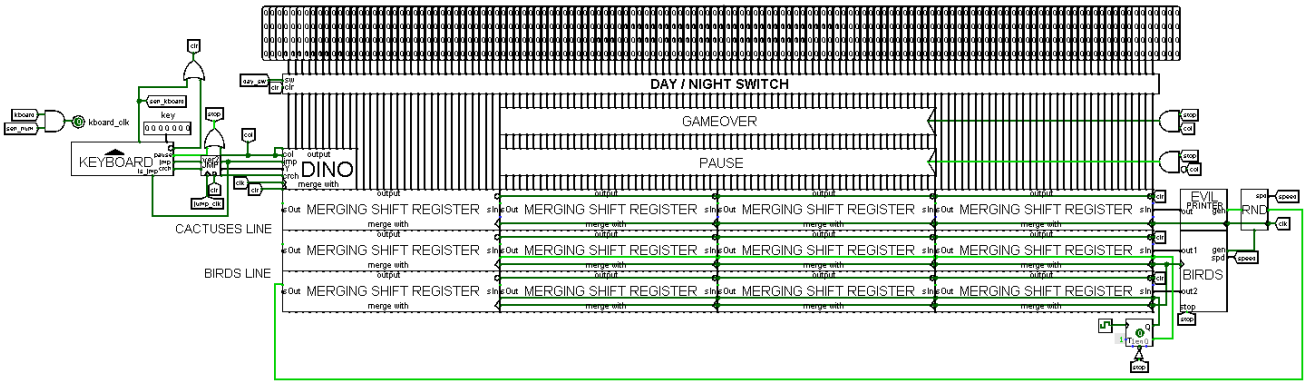
Данная схема служит интерфейсом проекта. С ней напрямую взаимодействует пользователь. Она состоит из следующих элементов:

1. Дисплей из 4-х матриц 32x32.
2. 5 семисегментных индикаторов, показывающих текущее количество очков.
3. Матрица 6x24 для отображения текущего уровня.
4. 4 семисегментных индикатора, отображающих время игры в виде MM:CC.
5. Клавиатура, предназначенная для управления динозавриком.
6. Клавиатура, предназначенная для ввода никнейма.
7. Кнопка enter для ввода никнейма и кнопка clear для возможности очистки.
8. Дисплей из 5-ти матриц 5x16 и 5-ти матриц 5x15.

4.2 main

Схема main состоит из нескольких ключевых частей.

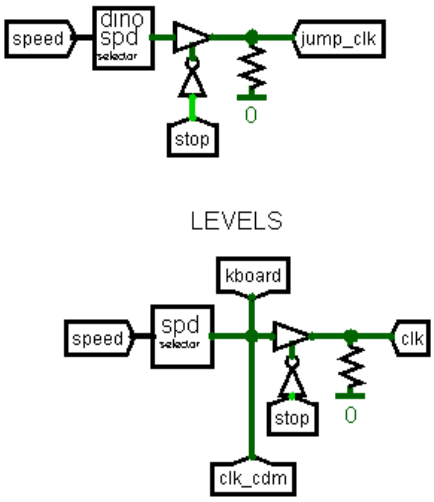
1. Блок отрисовки и управления персонажем.



Скриншот 2. «Схема main: блок отрисовки и управления персонажем»

В этой части схемы происходит генерация и отрисовка препятствий (подсхемы obstacles и birds – для генерации, shiftreg – для отрисовки и перемещения), рандомизация появления препятствий (подсхема random-spawn), а также управление персонажем и его отрисовка (подсхемы keyboard-handler, dino-jump, dino). Помимо этих подсхем, здесь также применены подсхемы pause и gameover для отрисовки соответствующих надписей, и подсхема invertor для переключения дня и ночи.

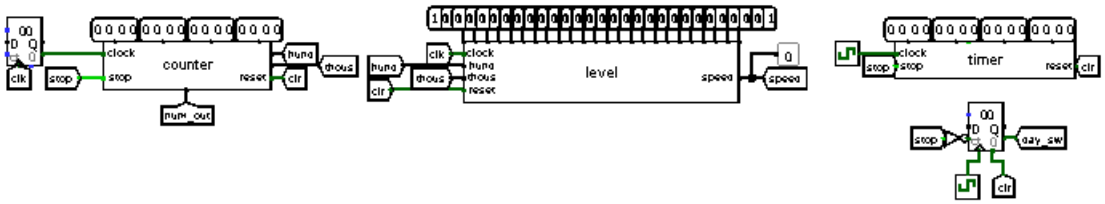
2. Блок управления скоростью игры.



Скриншот 3. «Схема main: блок управления скоростью игры»

Здесь посредством подсхем speed и speeddino происходит выбор тактовой частоты для изменения скорости передвижения объектов и скорости прыжка.

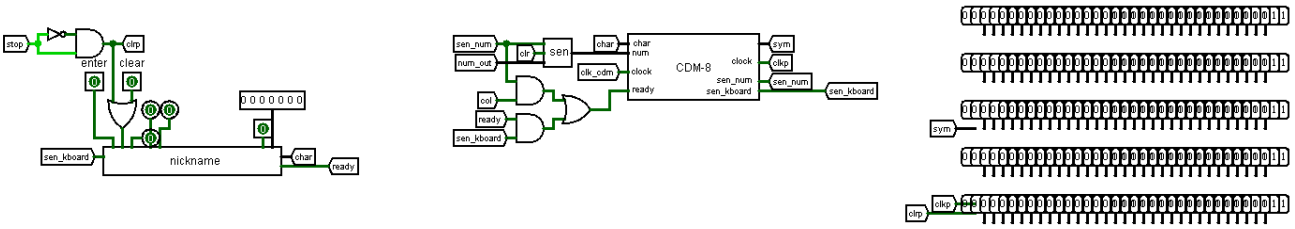
3. Блок со счётчиками.



Скриншот 4. «Схема main: блок со счётчиками»

Состоит из подсхем counter (для подсчёта очков), level (для отображения текущего уровня сложности) и timer (счёт времени игры в минутах и секундах). В этой части также реализован расчёт времени переключения дня и ночи.

4. Блок с процессором



Скриншот 5. «Схема main: блок с процессором»

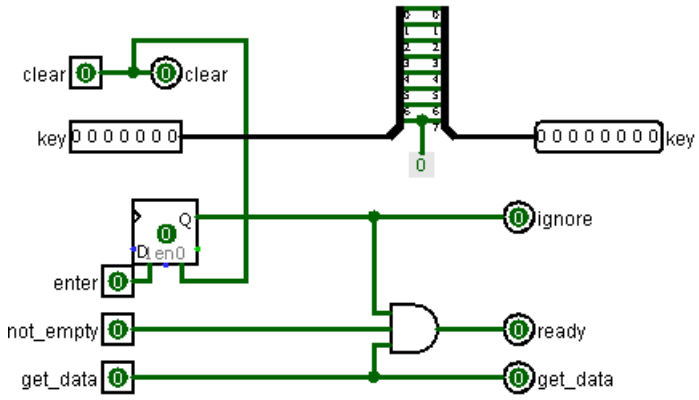
Состоит из подсхем nickname (для ввода никнейма), cdm-8 (процессор с сортировкой) и magic_printer (для вывода турнирной таблицы).
Входы схемы main:

- 2 7-битных числа – ключи клавиш внешних клавиатур.
- 1-битный контакт для очистки клавиатуры ввода никнейма.
- 1-битный контакт – кнопка завершения ввода никнейма.
- 1-битный контакт – наличие символов в клавиатуре ввода никнейма.

Выходы схемы main:

- 128 32-битных чисел – данные для заполнения матриц дисплея.
- 5 4-битных чисел – данные для семисегментных индикаторов счётчика очков.
- 24 6-битных числа – данные для заполнения матрицы отображения текущего уровня сложности.
- 4 4-битных числа – данные для семисегментных индикаторов таймера.
- 2 1-битных контакта частоты опрашивания внешних клавиатур.
- 1-битный контакт разрешения на ввод следующего никнейма.
- 1-битный контакт для очистки клавиатуры ввода никнейма.
- 155 5-битных числа – данные для заполнения матриц таблицы рекордов.

4.3 nickname



Скриншот 6. «Схема nickname»

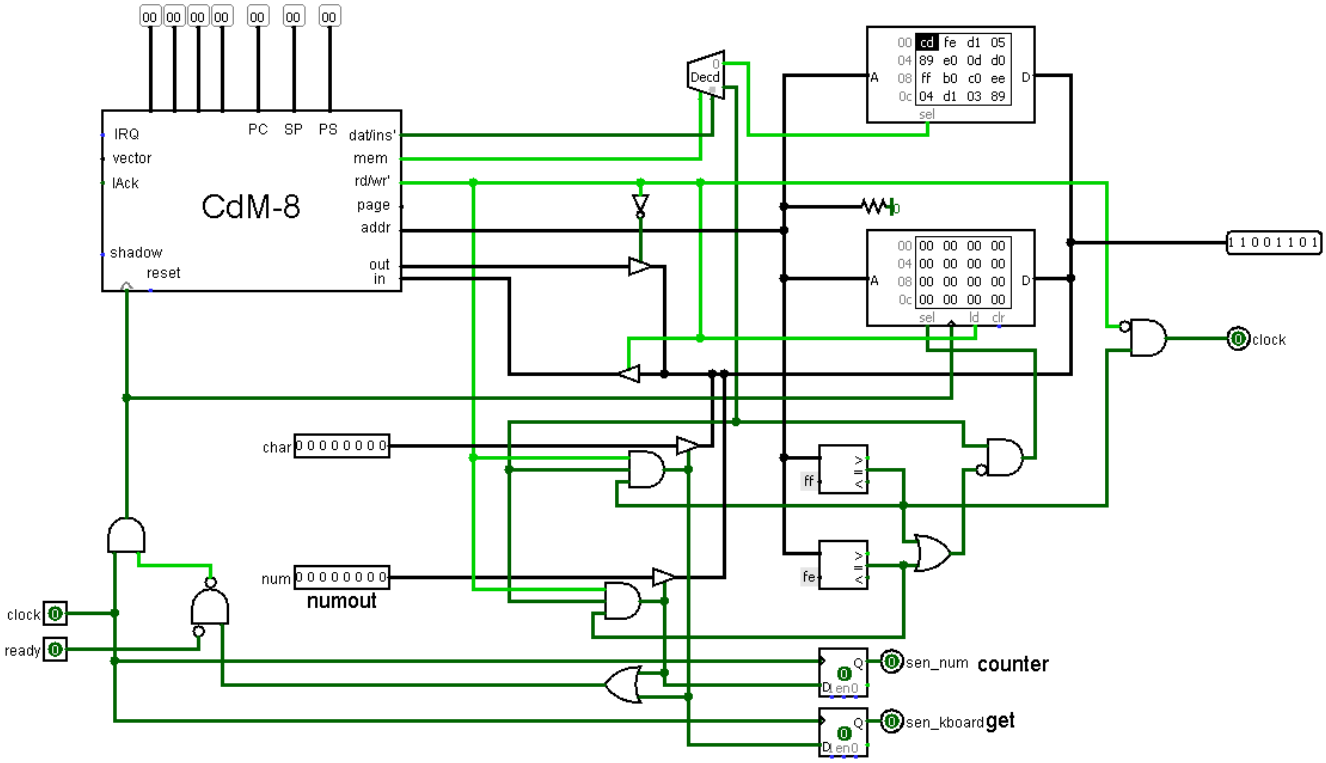
Схема для обработки ввода с клавиатуры ввода никнейма. Приводит ввод в более удобный для обработки процессором вид.
Входы схемы nickname:

- 1-битный контакт clear – для очистки внешней клавиатуры. Транслирует значение в одноимённый выход. Также устанавливает значение триггера enter на 0.
- 7-битный контакт key – принимает ключ символа с внешней клавиатуры.
- 1-битный контакт enter – принимает сигнал с кнопки enter в схеме interface.
- 1-битный контакт not_empty – принимает сигнал о наличии символа с внешней клавиатуры.
- 1-битный контакт get_data – готовность процессора к записи данных.

Выходы схемы nickname:

- 1-битный контакт clear – для очистки внешней клавиатуры.
- 1-битный контакт ignore – для игнорирования тактового входа внешней клавиатуры.
- 1-битный контакт ready – готовность клавиатуры к передаче данных процессору.
- 1-битный контакт get_data – считать символ с клавиатуры.

4.4 cdm-8



Скриншот 7. «Схема cdm-8»

Это схема сортирует в RAM (ОЗУ) полученные данные и поочередно выводит их.

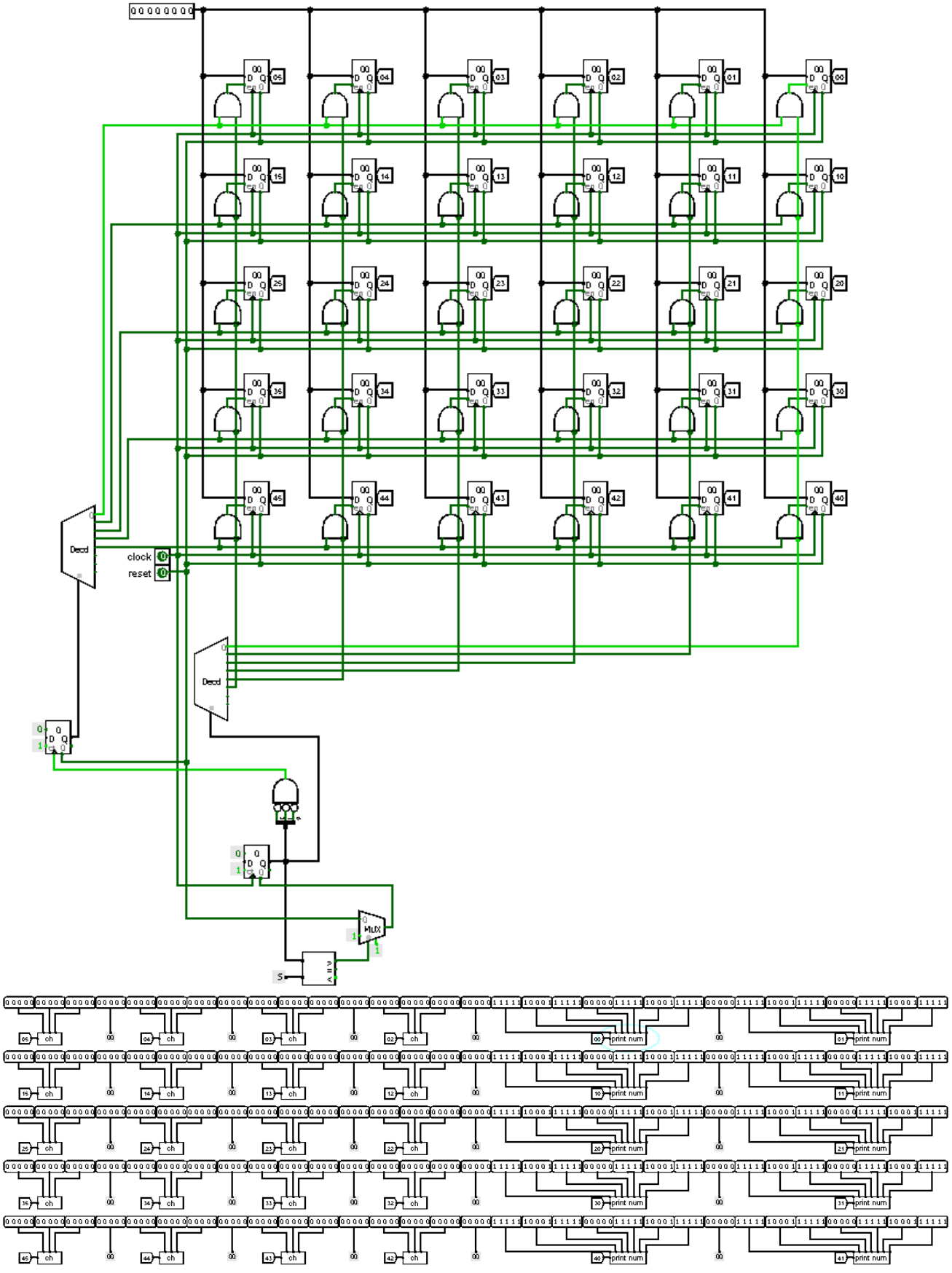
Входы схемы cdm-8:

- 1-битный контакт clock – тактовый генератор.
- 1-битный контакт ready – клавиатура готова передавать символы.
- 8-битное число char – для передачи символа никнейма.
- 8-битное число num – для передачи количества очков.

Выходы схемы cdm-8:

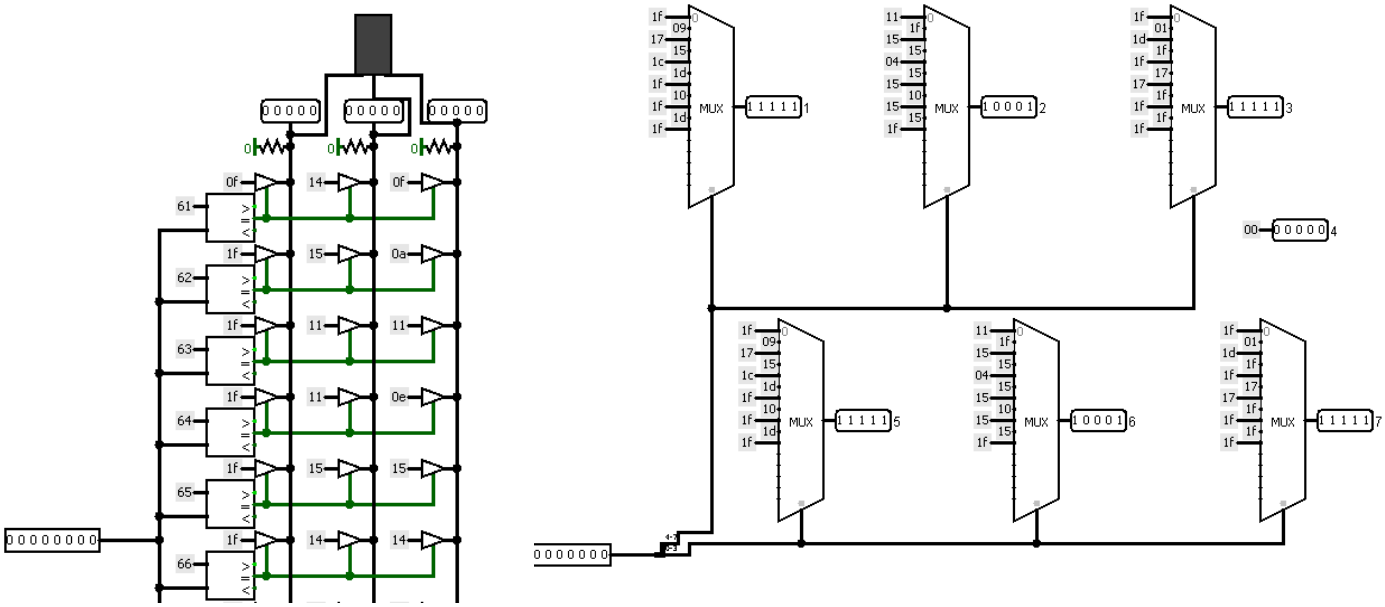
- 8-битный контакт – для вывода отсортированных данных.
- 1-битный контакт – для разрешения вывода данного значения.

4.5 magic-printer



Скриншот 8. «Схема magic_printer»

С помощью двух счетчиков вычисляются координаты, по которым будет выводится символ. Для вывода используются схемы char_out (для вывода букв) и num_out (для вывода чисел). Эти две схемы представляют собой просто отрисованные символы.



Скриншот 9. «Схема char_out» - слева, «схема num_out» - справа

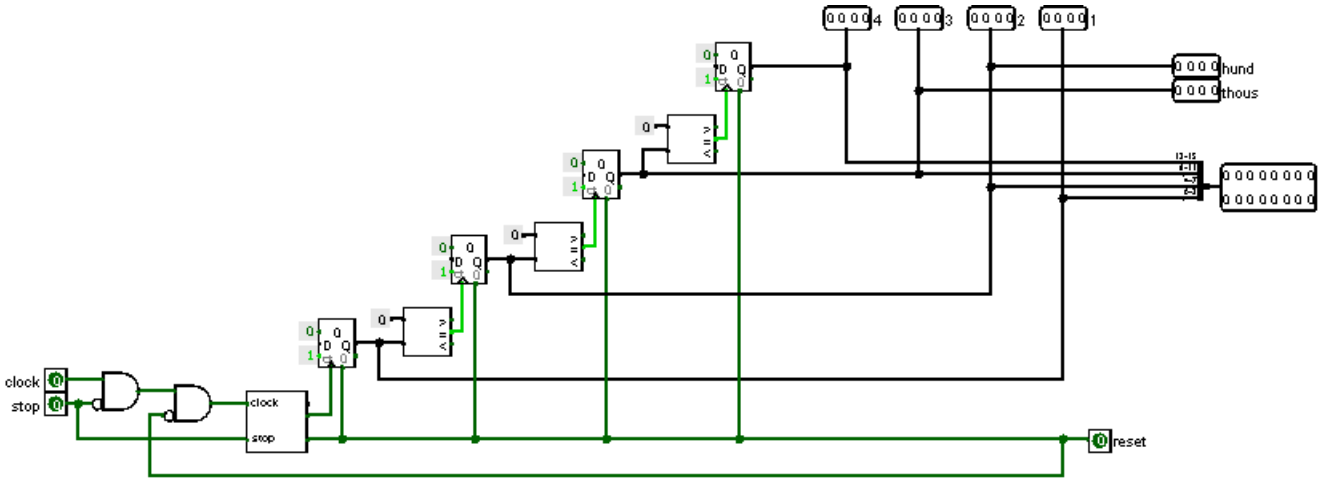
Входы схемы magic_printer:

- 1-битный контакт clock – тактовый генератор.
- 1-битный контакт reset – для очистки данных.
- 8-битное число – ключ символа, который нужно вывести.

Выходы схемы magic_printer:

- 5-битное число – для отрисовки символов.

4.6 counter



Скриншот 10. «Схема counter»

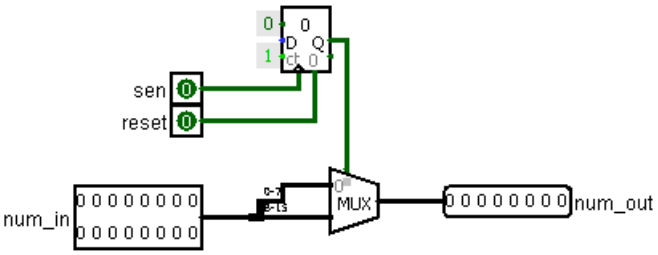
Состоит из bcd-counter и счетчиков, которые считают десятки, сотни и тысячи. На входе использует тактовый генератор для синхронизации счета, сигнал стоп и очистки. Выводит сотни и тысячи для смены уровня и 16-битное представление очков, которое с помощью схемы number_sen разделяется на два 8-битных числа и поочередно выводятся.

Входы схемы counter:

– 1-битный контакт clock – тактовый генератор.
– 1-битный контакт stop – для остановки счётчика.
– 1-битный контакт reset – для сброса счётчика.

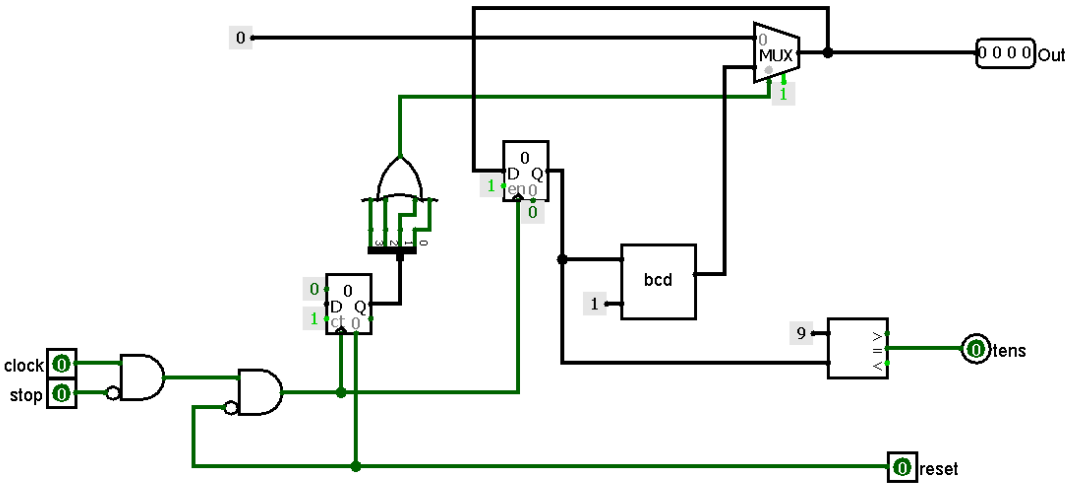
Выходы схемы counter:

– 4 4-битных числа – счёт в представлении BCD.
– 2 4-битных числа – значение разрядов сотен и тысяч в представлении BCD.
– 1 16-битное число – конкатенация всех чисел счётчика.



Скриншот 11. «Схема number_sen»

4.7 bcd-counter



Скриншот 12. «Схема bcd-counter»

В этой части схемы происходит работа счетчика очков, который использует на входе тактовый генератор для синхронизации счета, сигнал стоп и остановки. Сам счетчик реализован за счет bcd-add с инкрементом на единицу. Также при достижении значения 9 происходит установка флага, отвечающего за количество десятков, что позволяет отслеживать переход от одной десятки к следующей.

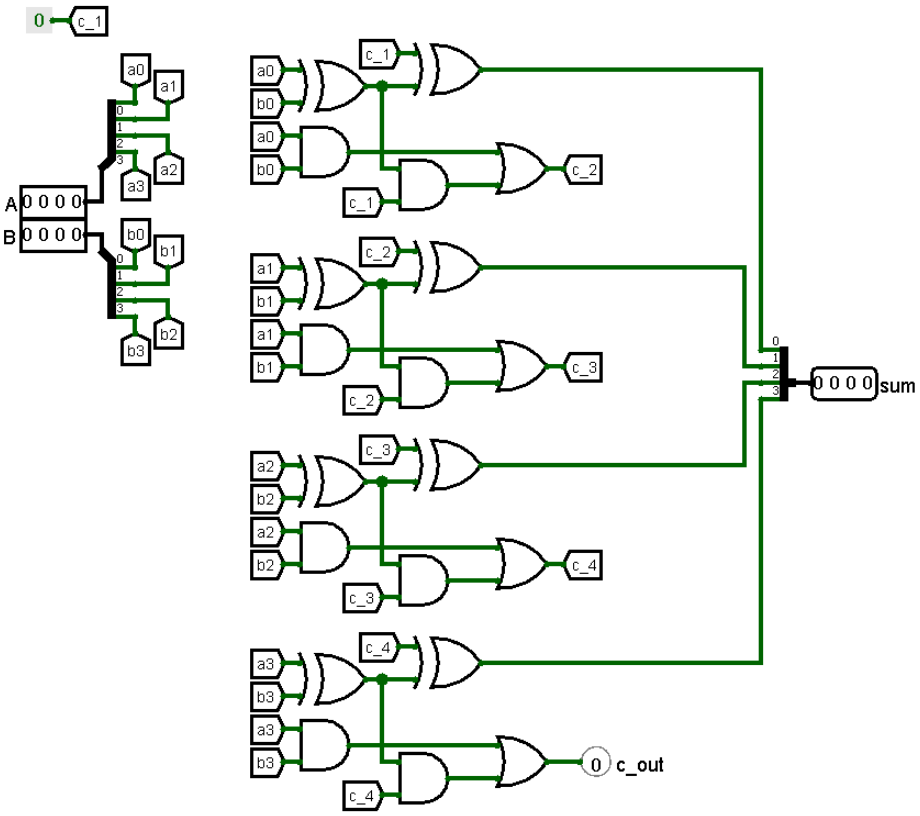
Входы схемы bcd-counter:

– 1-битный контакт clock – тактовый генератор.
– 1-битный контакт stop – для остановки счётчика.
– 1-битный контакт reset – для сброса счётчика.

Выходы схемы bcd-counter:

– 1 4-битное число – цифра в представлении BCD.
– 1-битный контакт – поднят, если счётчик равен 9.

4.8 bcd-add



Скриншот 13. «Схема bcd-add»

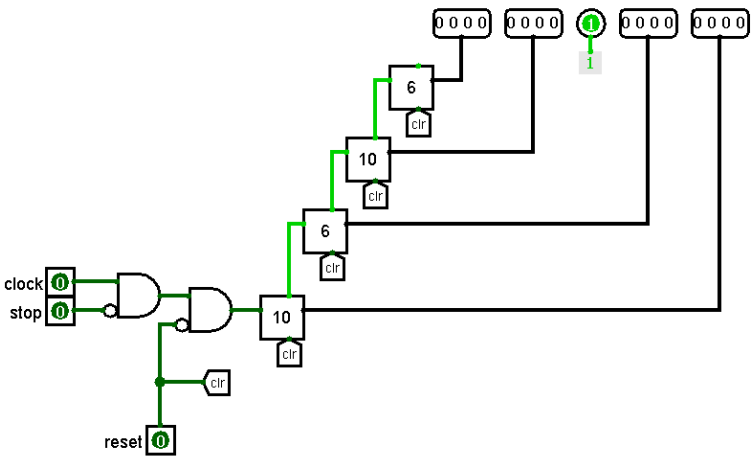
BCD сумматор получает две цифры и с учетом переноса выводит результат сложения.
Входы схемы bcd-add:

- 2 4-битных числа – цифры в представлении BCD.

Выходы схемы bcd-add:

- 1 4-битное число – результат сложения двух цифр на входе.

4.9 timer



Скриншот 14. «Схема timer»

Состоит из 2-х 6-counter и 2-х 10-counter, которые считают минуты и секунды. На входе использует тактовый генератор для синхронизации счета, сигнал стоп и остановки. Выводит минуты, разделитель, секунды.

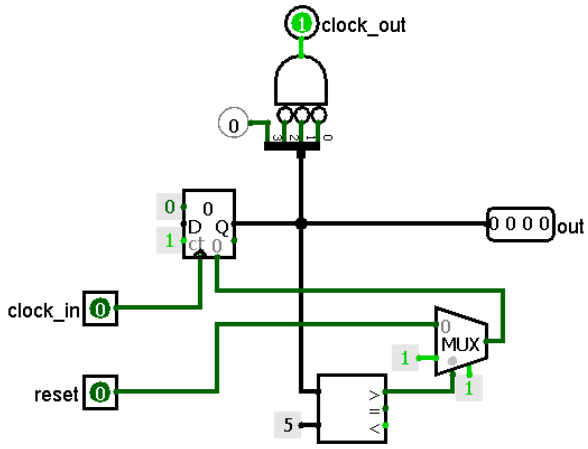
Входы схемы timer:

- 1-битный контакт clock – тактовый генератор.
- 1-битный контакт stop – для остановки таймера.
- 1-битный контакт reset – для сброса таймера.

Выходы схемы timer:

- 4 4-битных числа – цифры таймера в формате BCD.
- 1-битный контакт – константная единица (для удобства подключения).

4.10 6-counter



Скриншот 15. «Схема 6-counter»

Эта схема представляет собой счетчик до 6, который на вход получает тактовый генератор и сигнал очистки, а выводит значение счетчика и сигнал тактового генератора, который будет использоваться в следующих к нему подключенных счетчиках.

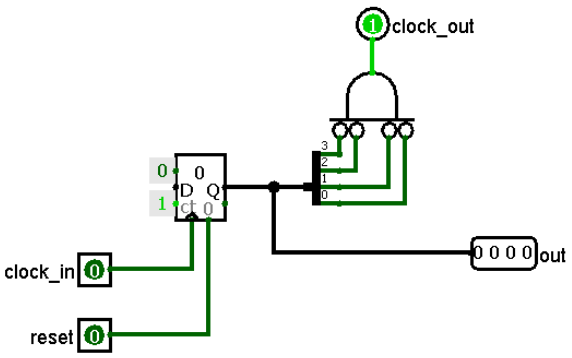
Входы схемы 6-counter:

- 1-битный контакт clock_in – тактовый генератор.
- 1-битный контакт reset – для сброса счётчика.

Выходы схемы 6-counter:

- 4-битное число – значение счётчика.
- 1-битный контакт – редуцированный тактовый генератор.

4.11 10-counter



Скриншот 16. «Схема 10-counter»

Эта схема представляет собой счетчик до 10, который на вход получает тактовый генератор и сигнал очистки, а выводит значение счетчика и сигнал тактового генератора, который будет использоваться в следующих к нему подключенных счетчиках.

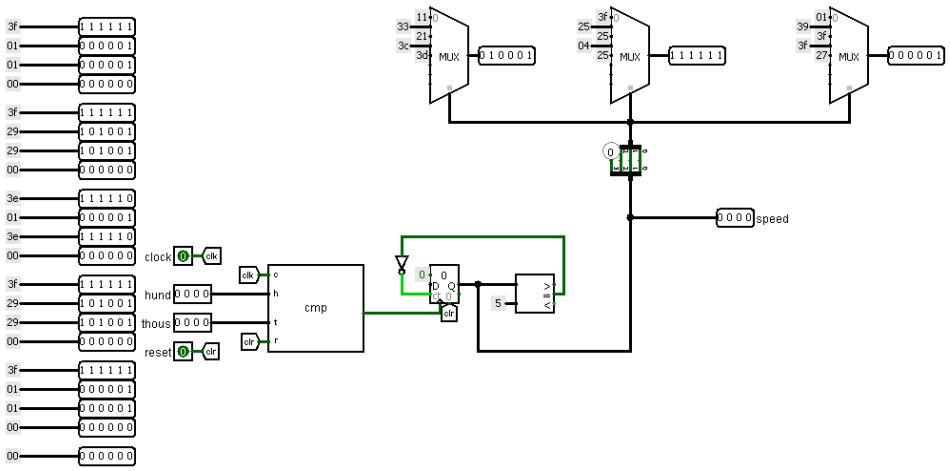
Входы схемы 10-counter:

- 1-битный контакт clock_in – тактовый генератор.
- 1-битный контакт reset – для сброса счётчика.

Выходы схемы 10-counter:

- 4-битное число – значение счётчика.
- 1-битный контакт – редуцированный тактовый генератор.

4.12 level



Скриншот 17. «Схема level»

Состоит из константного вывода надписи «level» и числа. На входе принимает тактовый генератор, сигнал очистки, сотни и тысячи, по которым происходит отслеживание необходимых набранных очков для смены уровня (для этого используется «схема cmp»).

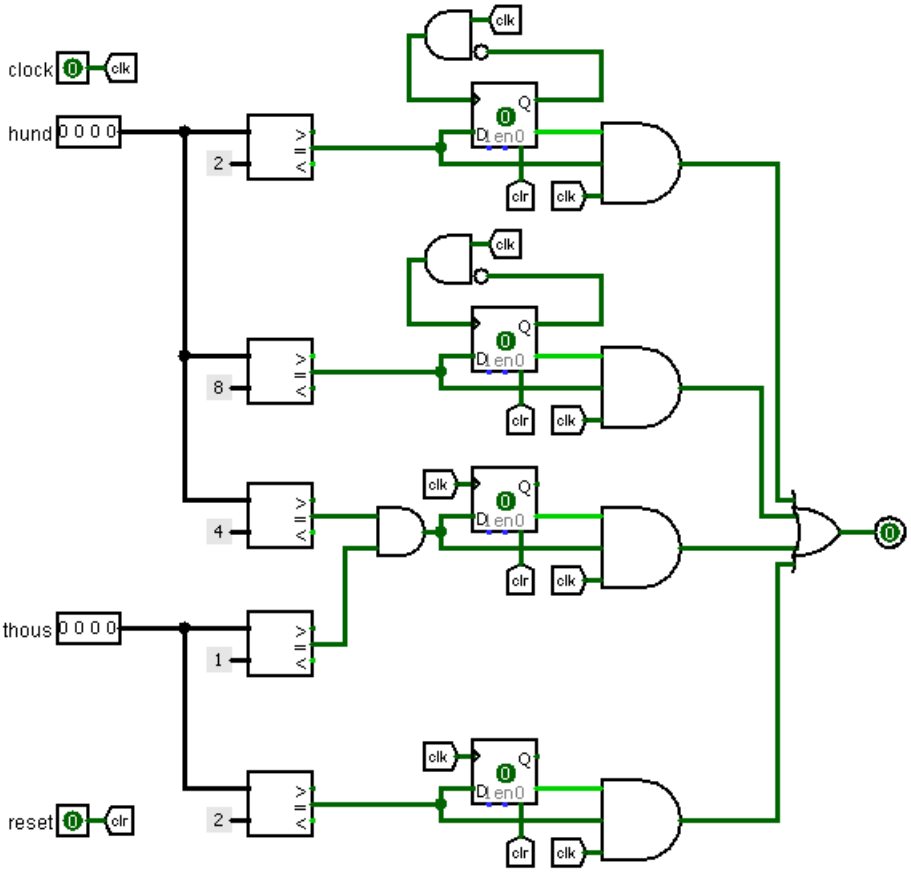
Входы схемы level:

- 1-битный контакт clock – тактовый генератор.
- 1-битный контакт reset – для сброса счётчика.
- 2 4-битных числа – значение разрядов сотен и тысяч в представлении BCD, получаемых со схемы counter.

Выходы схемы level:

- 21 6-битное число – константы для вывода надписи «level» на матрице.
- 3 6-битных числа – для выводы текущего уровня на матрице.
- 4-битное число – текущий уровень.

4.13 cmp



Скриншот 18. «Схема cmp»

Сигнал смены уровня поднимается, когда на счетчике определенное количество очков (200-800-1400-2000). Схема cmp принимает на вход сотни и тысячи и сравнивает их, после первого вхождения той или иной цифры сигнал поднимается.

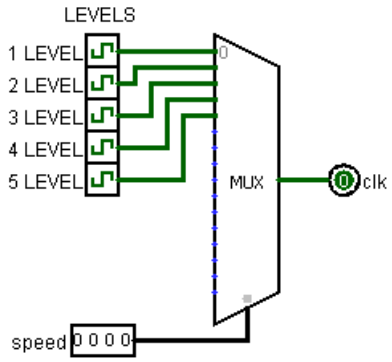
Входы схемы cmp:

- 1-битный контакт clock – тактовый генератор.
- 1-битный контакт reset – для сброса триггеров.
- 2 4-битных числа – значение разрядов сотен и тысяч в представлении BCD, получаемых со схемы counter.

Выходы схемы cmp:

- 1-битный контакт – поднимается по достижении определённого количества очков.

4.14 speed



Скриншот 19. «Схема speed»

Схема выбирает тактовый генератор определённой частоты в зависимости от числа, поданного на вход. Необходима для ускорения встречных объектов по мере увеличения количества очков.

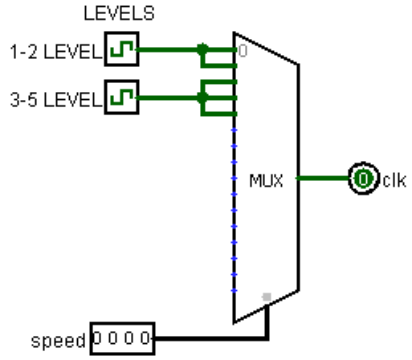
Входы схемы speed:

– 4-битное число speed, значением от 1 до 5 – текущий уровень скорости объектов.

Выходы схемы speed:

– 1-битный контакт clk – частота выбранного тактового генератора.

4.15 speeddino



Скриншот 20. «Схема speeddino»

Аналогичная предыдущей схема для управления прыжком динозаврика.

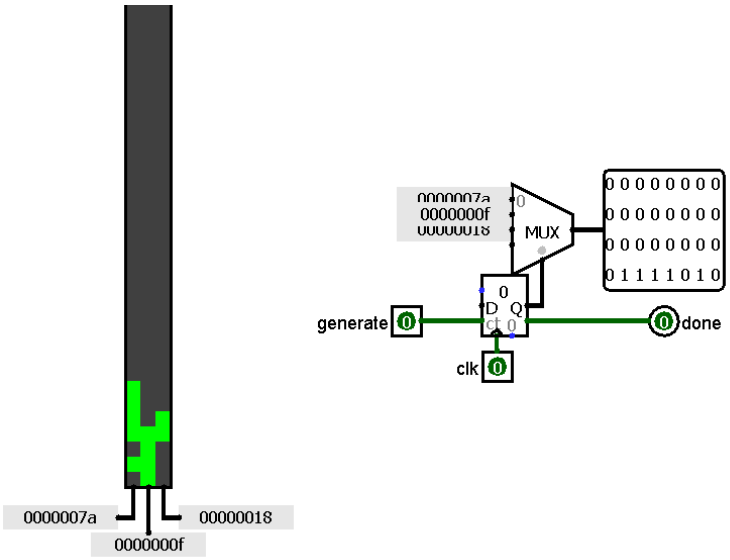
Входы схемы speeddino:

– 4-битное число speed, значением от 1 до 5 – текущий уровень скорости объектов.

Выходы схемы speeddino:

– 1-битный контакт clk – частота выбранного тактового генератора.

4.16 cactus



Скриншот 21. «Схема cactus»

Типовая схема генерации объектов. Объект хранится в константах. Передача для отрисовки происходит по столбикам. Счётчик пробегает все столбцы, мультиплексор передаёт на выход соответствующие 32-битные значения.

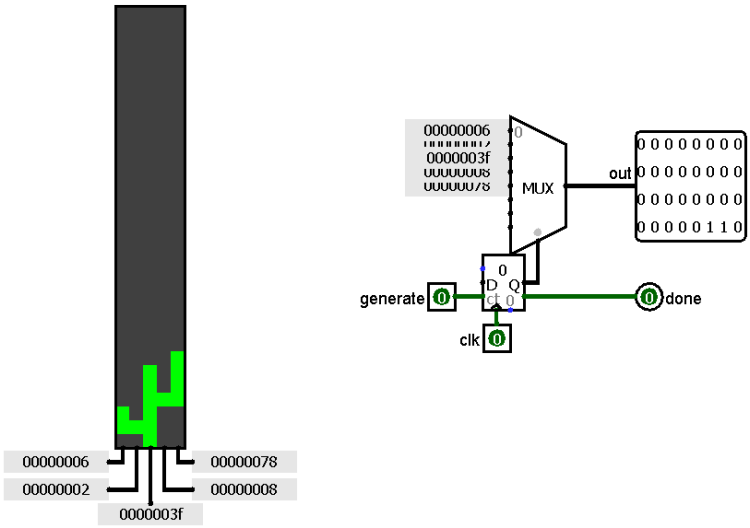
Входы схемы cactus:

- 1-битный контакт generate – флаг начала отрисовки.
- 1-битный контакт clk – частота.

Выходы схемы cactus:

- 1-битный контакт done – флаг готовности работы схемы, поднят, когда генерация объекта выполнена.

4.17 big-cactus



Скриншот 22. «Схема big-cactus»

Принцип работы аналогичен предыдущей схеме.

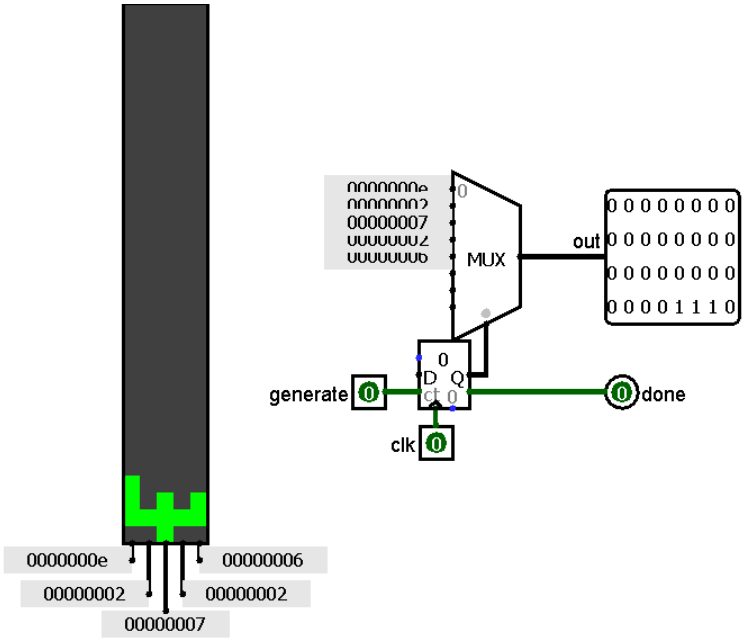
Входы схемы big-cactus:

- 1-битный контакт generate – флаг начала отрисовки.
- 1-битный контакт clk – частота.

Выходы схемы big-cactus:

- 1-битный контакт done – флаг готовности работы схемы, поднят, когда генерация объекта выполнена.

4.18 small-cactus



Скриншот 23. «Схема small-cactus»

Принцип работы аналогичен предыдущей схеме.

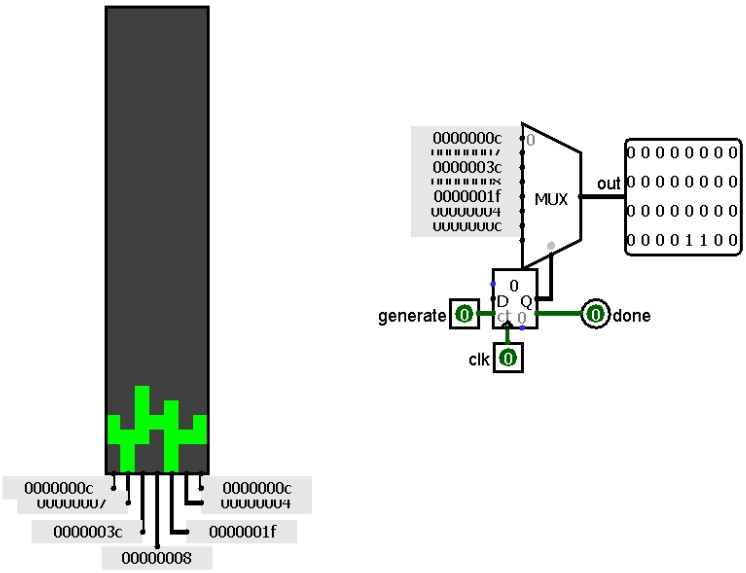
Входы схемы small-cactus:

- 1-битный контакт generate – флаг начала отрисовки.
- 1-битный контакт clk – частота.

Выходы схемы small-cactus:

- 1-битный контакт done – флаг готовности работы схемы, поднят, когда генерация объекта выполнена.

4.19 wide-cactus



Скриншот 24. «Схема wide-cactus»

Принцип работы аналогичен предыдущей схеме.

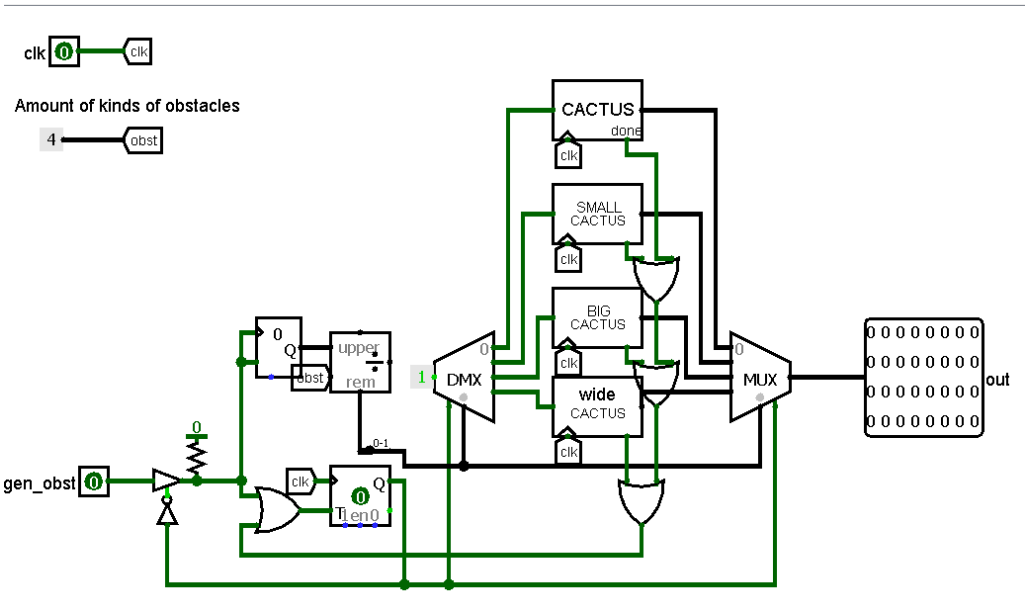
Входы схемы wide-cactus:

- 1-битный контакт generate – флаг начала отрисовки.
- 1-битный контакт clk – частота.

Выходы схемы wide-cactus:

- 1-битный контакт done – флаг готовности работы схемы, поднят, когда генерация объекта выполнена.

4.20 obstacles



Скриншот 25. «Схема obstacles»

Данная схема по требованию выбирает случайное препятствие для отрисовки с помощью встроенного генератора случайных чисел и операции остатка от деления. Далее, так же, как и в предыдущих схемах, передаёт их по столбикам для отрисовки. Выбор передаваемого столбца реализован с помощью ранее упомянутых схем с кактусами и мультиплексоров/демультиплексоров. Пока отрисовка препятствия не окончена, новые генерироваться не будут. Это ограничение достигается благодаря управляемому буферу.

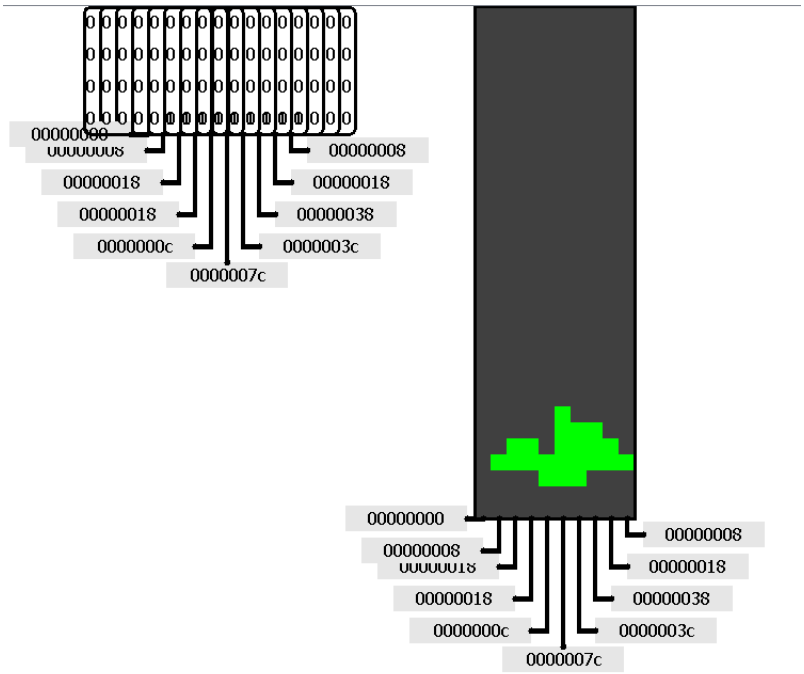
Входы схемы obstacles:

- 1-битный контакт gen_obst – флаг генерации случайного препятствия и начала отрисовки.
- 1-битный контакт clk – частота.

Выходы схемы obstacles:

- 32-битное число – столбец, отправляемый для отрисовки.

4.21 bird-wings-up



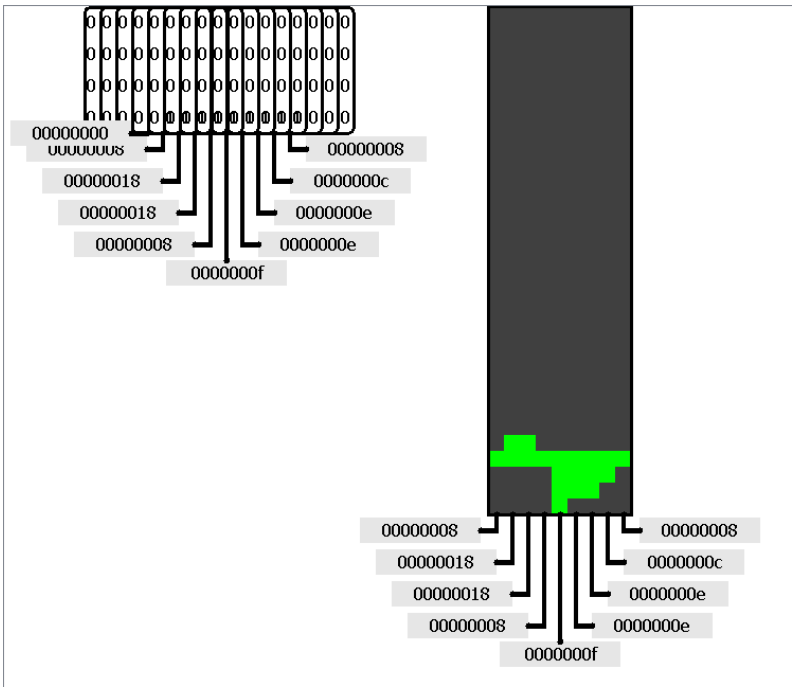
Скриншот 26. «Схема bird-wings-up»

Схема представляет собой не более чем набор констант для отрисовки птицы с поднятыми крыльями.

Выходы схемы bird-wings-up:

- 10 32-битных чисел – столбцы для отрисовки птицы.

4.22 bird-wingsdouwn



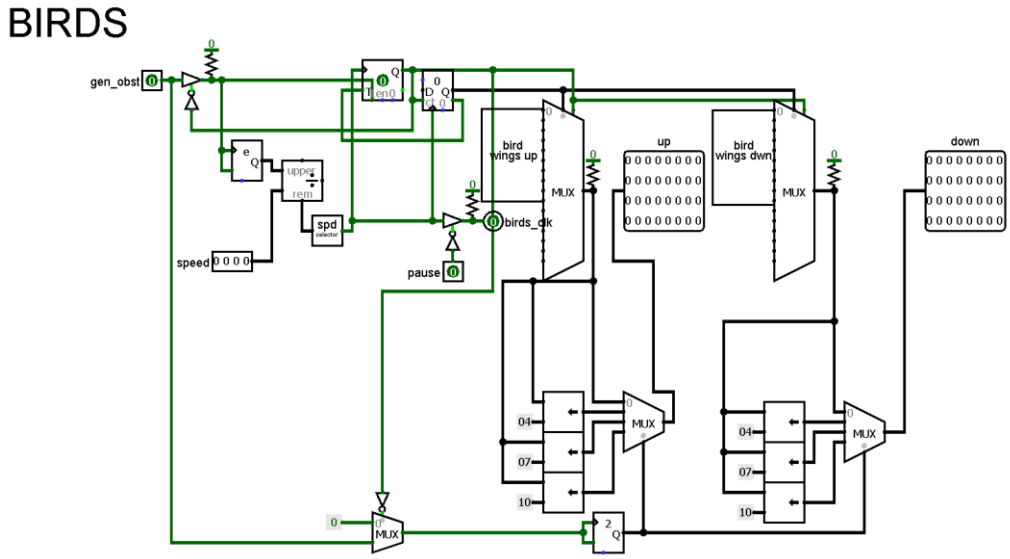
Скриншот 27. «Схема bird-wingsdown»

Схема, аналогичная предыдущей, но для птицы с опущенными крыльями.

Выходы схемы bird-wingsdown:

– 10 32-битных чисел – столбцы для отрисовки птицы.

4.23 birds



Скриншот 28. «Схема birds»

Текущая схема предназначена для генерации птиц. Это препятствие отличается тем, что у него есть анимация полета в два кадра. Также, скорость птиц может отличаться от скорости остальных препятствий. Птицы отрисовываются сразу в двух вариациях: с крыльями вниз и вверх

(в основной схеме main эти варианты чередуются), рандомайзер с делителем, как ранее в схеме obstacles, выбирает скорость полета выбранной птицы. Другой рандомайзер выбирает на какой высоте полетит птица: высоко, на уровне кактусов (необходимо перепрыгнуть) или «в лоб» (необходимо пригнуться).

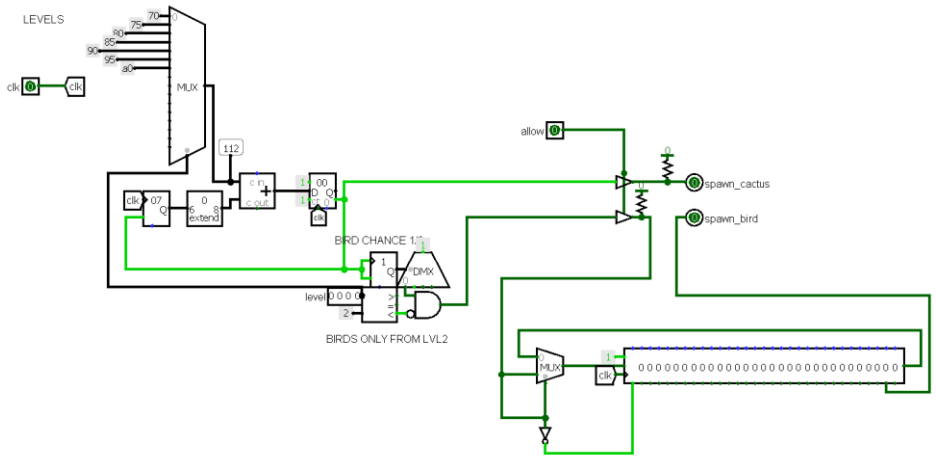
Входы схемы birds:

- 1-битный контакт gen_obst – флаг начала отрисовки птицы.
- 1-битный контакт pause – пауза.
- 4-битное число speed – текущая скорость движения остальных объектов.

Выходы схемы birds:

- 2 32-битных числа – столбцы, отправляемые для отрисовки птиц в двух вариациях: с поднятыми и опущенными крыльями.
- 1-битный контакт – частота (скорость) птицы, которая полетит.

4.24 random-spawn



Скриншот 29. «Схема random-spawn»

Схема реализует генератор случайного появления объектов и отдаёт схемам birds и obstacles команды для начала генерации препятствий. В зависимости от уровня меняется значение насколько близко друг другу появляются препятствия. Птицы появляются с шансом 12,5% и только со второго уровня, к тому же на определенном расстоянии от предыдущего объекта.

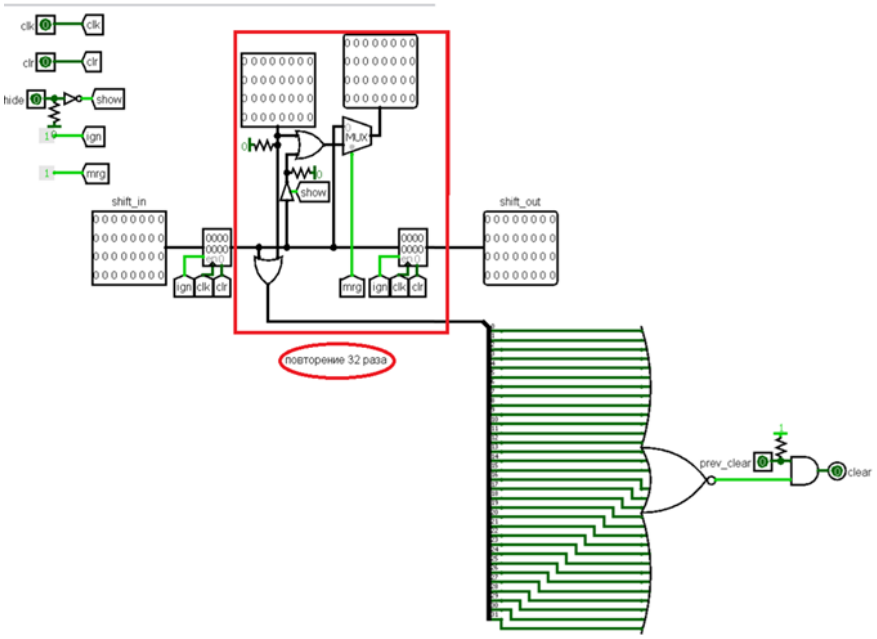
Входы схемы random-spawn:

- 1-битный контакт clk – частота.
- 1-битный контакт allowed – флаг, разрешающий/запрещающий генерацию.
- 4-битное число level – текущий уровень (скорость объектов).

Выходы схемы random-spawn:

- 1-битный контакт spawn_cactus – генерация кактуса.
- 1-битный контакт spawn_bird – генерация птицы.

4.25 shiftreg



Скриншот 30. «Схема shiftreg»

Данная схема приводит в движение и компонует все препятствия. По своей сути, она является модифицированным 32-битным сдвиговым регистром, способным сливать (операция OR) и подавать на выход своё внутреннее значение со значением, приходящим извне. Такая архитектура позволяет налагать один регистр на другой для отрисовки разных классов объектов, даже с различной скоростью. Другая модификация заключается в наличии флага hide, который позволяет, наоборот, игнорировать внутреннее значение регистра и работать как мост. Такая модификация является основой для анимации полёта птиц.

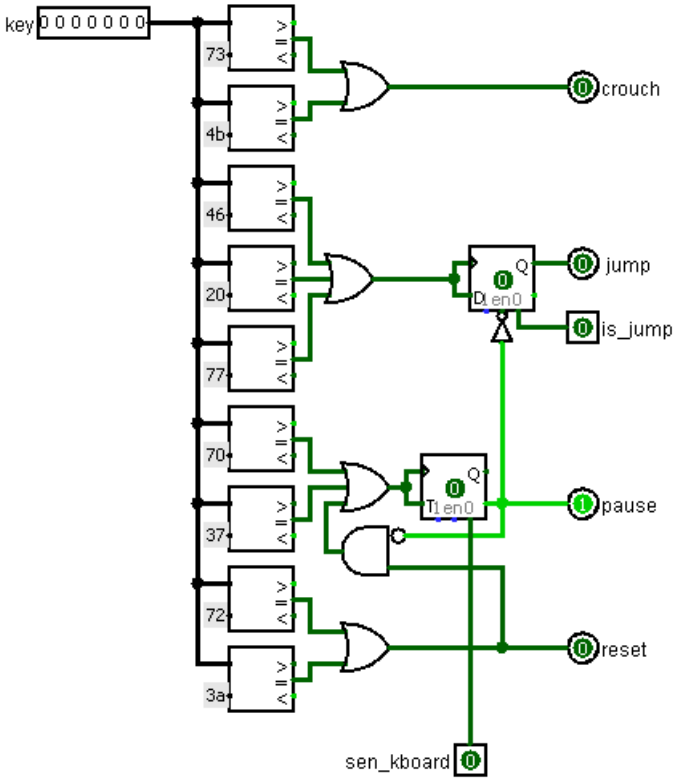
Входы схемы shiftreg:

- 1-битный контакт clk – частота.
- 1-битный контакт clr – флаг для очистки всех значений.
- 1-битный контакт prev_clear – информация о заполненности другого shiftreg, подключенного последовательно с текущим (опциональный параметр). Необходим для быстрого получения информации о том, пуста ли последовательность shiftreg. Используется для определения возможности сгенерировать птицу (если всё поле пусто).
- 1-битный контакт hide – скрывает хранящиеся внутри регистра значения, если флаг поднят.
- 32 32-битных значения – входящие значения для слияния. Не влияют на данные, хранящиеся в самом регистре, но влияют на исходящие значения.
- 32-битное число shift_in – задвигаемое в регистр значение.

Выходы схемы shiftreg:

- 1-битный контакт clear – является ли shiftreg пустым.
- 32 32-битных контакта – выходные значения.
- 32-битное число shift_out – значение, выдвинутое из shiftreg.

4.26 keyboard-handler



Скриншот 31. «Схема keyboard-handler»

Схема предназначена для обработки данных с клавиатуры. Обрабатывает и русскую, и английскую раскладку клавиатуры, а также пробел для прыжка. Приседание переключаемое (нажать, чтобы присесть, нажать повторно, чтобы встать). Пока динозаврик в прыжке повторные нажатия кнопки прыжка не учитываются.

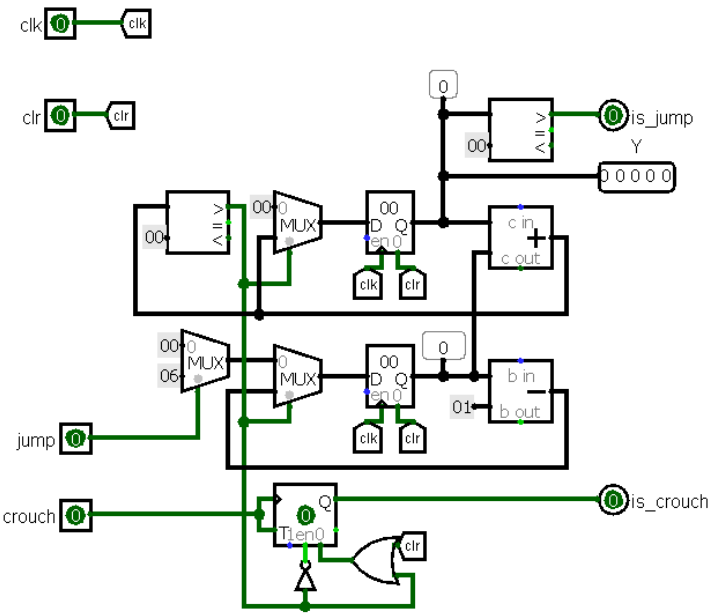
Входы схемы keyboard-handler:

- 7-битное число key – ключ нажатой клавиши на клавиатуре.
- 1-битный контакт is_jump – находится ли динозаврик в прыжке.
- 1-битный контакт sen_kboard – ждет ли клавиатура символов.

Выходы схемы keyboard-handler:

- 1-битный контакт crouch – была ли нажата кнопка приседания.
- 1-битный контакт jump – была ли нажата кнопка прыжка.
- 1-битный контакт pause – была ли нажата кнопка паузы.
- 1-битный контакт reset – была ли нажата кнопка рестарта.

4.27 dino-jump



Скриншот 32. «Схема dino-jump»

Схема, реализующая физическую формулу вертикального движения объекта:

$$v = v_0 - gt, \qquad s = s_0 + vt$$

где v_0 = 6, а g = 1 в нашем случае. Расчёт, происходит, соответственно, каждый тик clk. Присед прерывается прыжком. Прыжок считается завершённым, когда координата Y снова становится нулём.

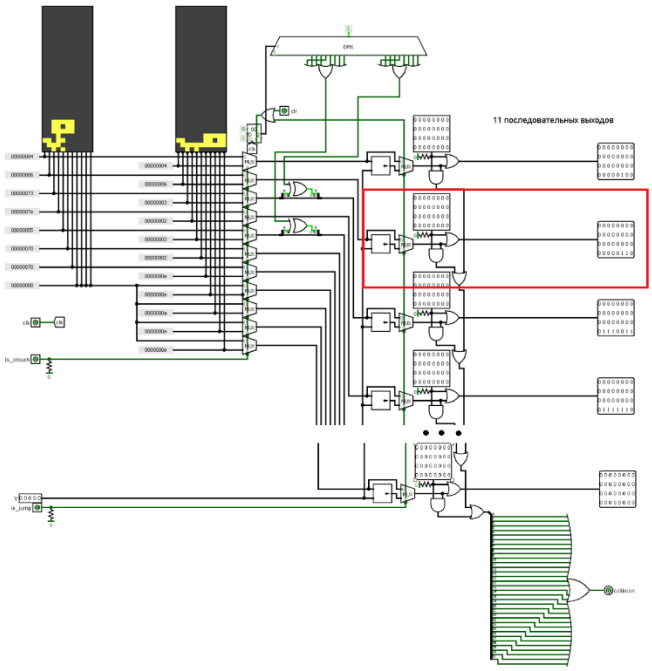
Входы схемы dino-jump:

- 1-битный контакт clk – частота.
- 1-битный контакт clr – очистка всех регистров и триггеров, если поднят.
- 1-битный контакт jump – команда начала прыжка.
- 1-битный контакт crouch – команда включения/отключения приседа.

Выходы схемы dino-jump:

- 1-битный контакт is_crouch – крадётся ли динозаврик
- 1-битный контакт is_jump – находится ли динозаврик в прыжке.
- 5-битный контакт Y – координата вертикального положения динозаврика.

4.28 dino



Скриншот 33. «Схема dino»

Схема реализует:

- Анимацию и отображение персонажа в двух состояниях — стоя и присев, с учётом координаты вертикального положения. Перемещение по вертикали происходит с помощью битовых сдвигателей.
- Объединение спрайтов персонажа и препятствий, чтобы корректно визуализировать игровую сцену. Принцип объединения тот же, что и в схеме shiftreg.
- Обнаружение столкновений (коллизий) между персонажем и препятствиями на пиксельном уровне. С помощью логических И (AND) с входящими данными. Если ИЛИ (OR) над получившимися числами даёт нам не ноль – произошла коллизия.
- Управление выводом игровых данных для дальнейшей обработки или отображения на экране.

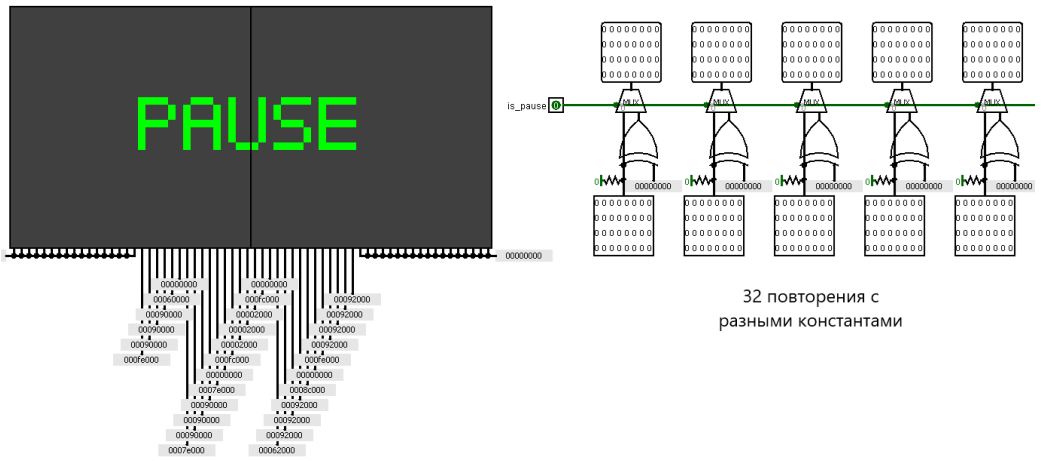
Входы схемы dino:

- 1-битный контакт clk – частота.
- 1-битный контакт clr – очистка всех регистров и триггеров, если поднят.
- 1-битный контакт is_jump – находится ли динозаврик в прыжке.
- 1-битный контакт is_crouch – крадётся ли динозаврик.
- 1-битный контакт cls – произошла ли коллизия (нужно для отрисовки анимации проигрыша)
- 5-битное число Y – координата вертикального положения спрайта.
- 11 32-битных чисел – столбцы игрового поля в месте отрисовки динозаврика.

Выходы схемы dino:

- 11 32-битных чисел – столбцы игрового поля с отрисованным поверх динозавриком.
- 1-битный контакт collision – поднят, если произошла коллизия.

4.29 pause



Скриншот 34. «Схема pause»

Схема реализует наложение соответствующей надписи на игровое поле в случае паузы посредством XOR (для лучшей читабельности, если позади надписи окажется объект).

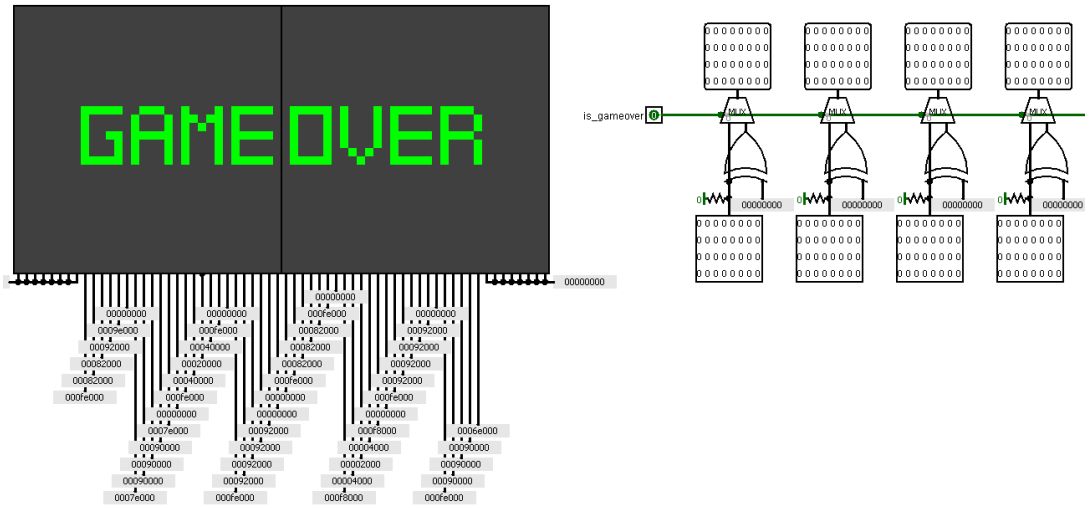
Входы схемы pause:

- 1-битный контакт is_pause – нажата ли в данный момент пауза.
- 64 32-битных контакта – игровое поле позади надписи.

Выходы схемы pause:

- 64 32-битных числа – столбцы игрового поля с отрисованной поверх надписью.

4.30 gameover



Скриншот 35. «Схема gameover»

Схема аналогична предыдущей, но для надписи «gameover».

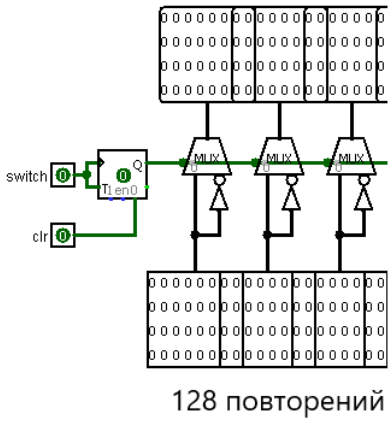
Входы схемы gameover:

- 1-битный контакт is_gameover – закончилась ли игра.
- 64 32-битных контакта – игровое поле позади надписи.

Выходы схемы gameover:

– 64 32-битных числа – столбцы игрового поля с отрисованной поверх надписью.

4.31 invertor



Скриншот 36. «Схема invertor»

Схема смены дня и ночи. По своей сути представляет из себя инвертор для всего игрового поля.

Входы схемы invertor:

– 1-битный контакт switch – переключить день на ночь или наоборот.
– 1-битный контакт clr – сброс (установить день).
– 128 32-битных контакта – игровое поле.

Выходы схемы invertor:

– 128 32-битных числа – игровое поле с изменённым временем суток.

5 ЗАКЛЮЧЕНИЕ

В этом проекте мы создали систему, которая управляет игровым процессом — от рисования врагов и препятствий до сортировки турнирной таблицы. Алгоритм сортировки был успешно реализован и адаптирован для процессора CdM-8.

Система работает стабильно, умеет динамически создавать врагов, а таблица игроков обновляется в реальном времени и всегда остаётся упорядоченной.

В будущем можно улучшить сортировку, чтобы она быстрее работала с большим количеством данных, добавить новых врагов, более плавные анимации, поддержку цвета и различные декоративные спрайты и элементы.

6 ИСТОЧНИКИ

COMPUTING PLATFORMS / A. SHAFARENKO, S.P. HUNT. – 2015.