Higher college of Informatics

21931

“Tetris”

Notes

Mihail Peshkov

Vyacheslav Chelondaev

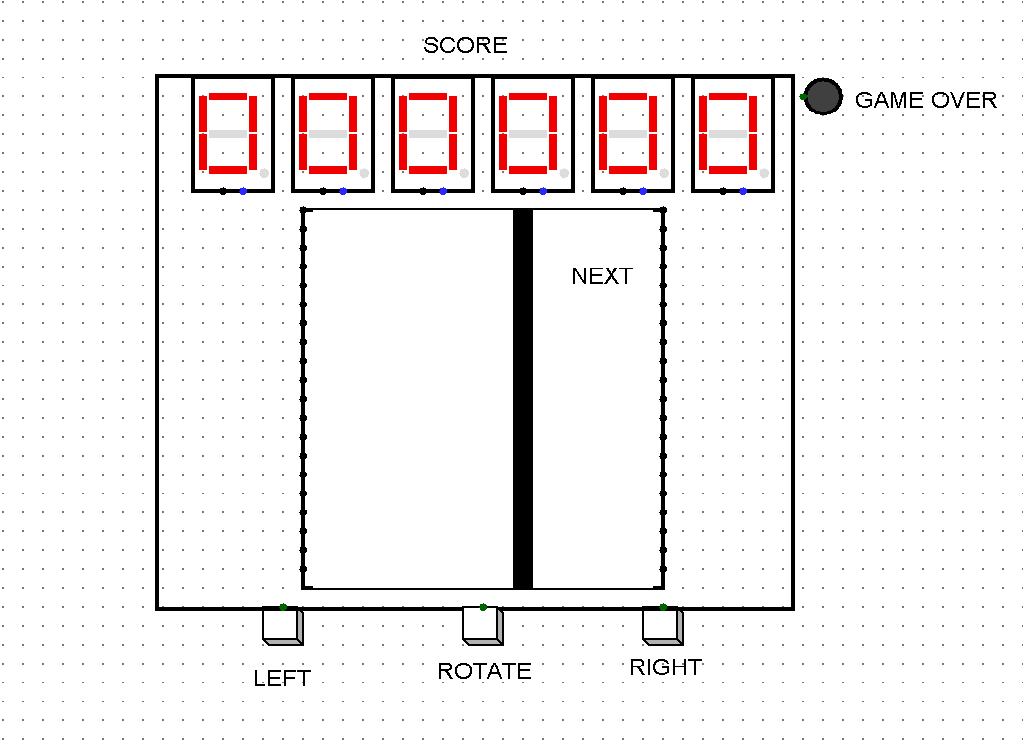
Sayan Radnaev

2022

Overview

For this project we were implemented a Tetris game.

Were Implemented a 17x20 video display, 3 buttons for control falling figures and scoreboard (picture 1).



picture 1

The display has been divided into 2 areas.

First: 10x20, this is the playing field itself, figures fall out and remain here.

Second: 7x20, that area shows the next falling figure.

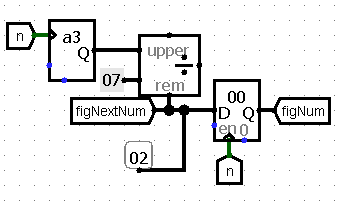
3 buttons for control: to left, rotate, to right

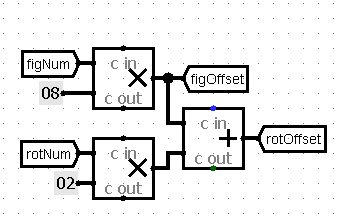
Rules

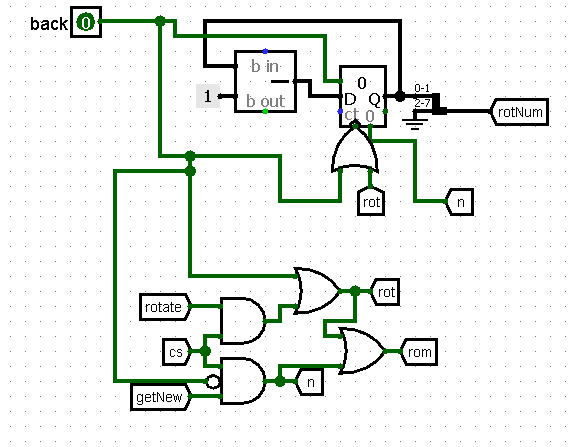
Random Tetrimino figures fall from above onto the field. While flying, the player can rotate the figure 90° and move it horizontally. The figurine flies until it hits another figurine or hits the bottom of the field. If at the same time a horizontal row of 10 cells is filled, it disappears and everything above it falls by one cell. Additionally, a figure is shown that will follow the current one - this is a hint that allows the player to plan actions. The game ends when the new figure cannot fit on the playing field. The player gets points for each filled row, so his task is to fill the rows without filling the glass itself (vertically) for as long as possible in order to get as many points as possible in this way.

Scoring

When scoring points for lines, the number of points depends on how many lines are cleared at a time. 1 line - 100 points, 2 lines - 300 points, 3 lines - 700 points, 4 lines - 1500 points.

 рандом фигур

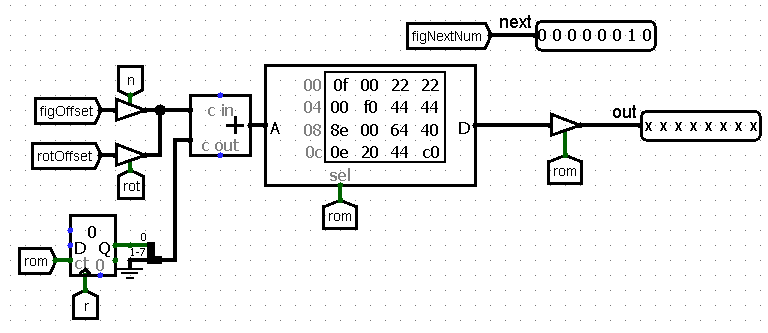
 отступ к нужной памяти для чтения данных (фигуры)



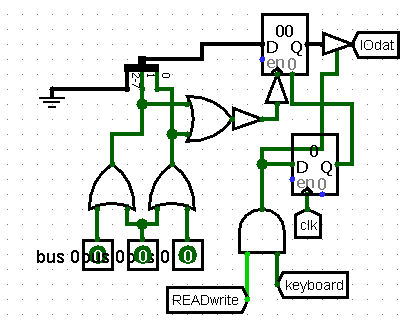
счетчик, для поворота фигуры

регистр с текущим номером спрайта

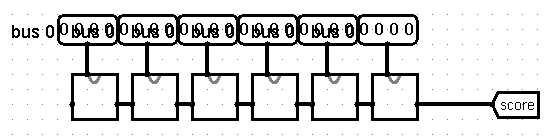
нижняя часть раздает инструкции в зависимости от поданных сигналов( ротате)



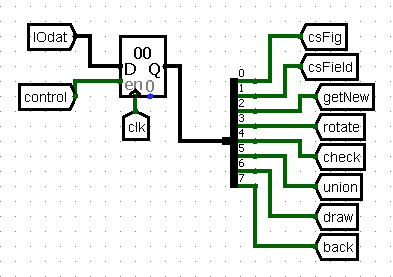
хранилка спрайтов



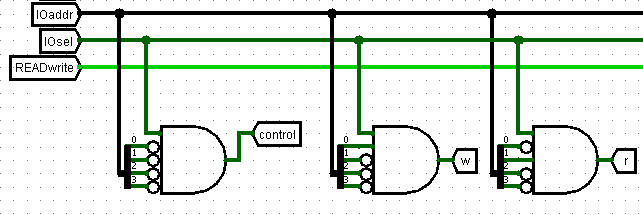
реализация кнопок



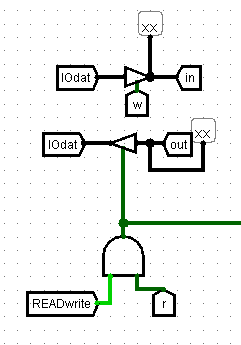
перевод счета в десятичный



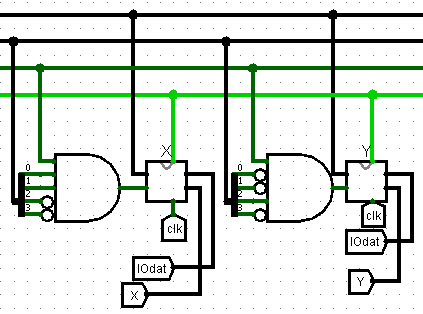
регистр контроля



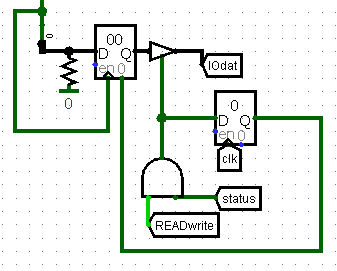
разрешение на чтение и запись \\ тактовая частота



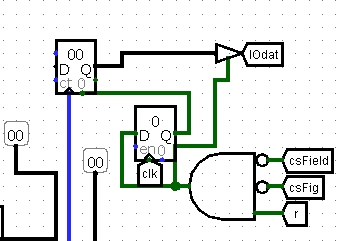
чтение и запись в основном, тактовая частота чисто по фану



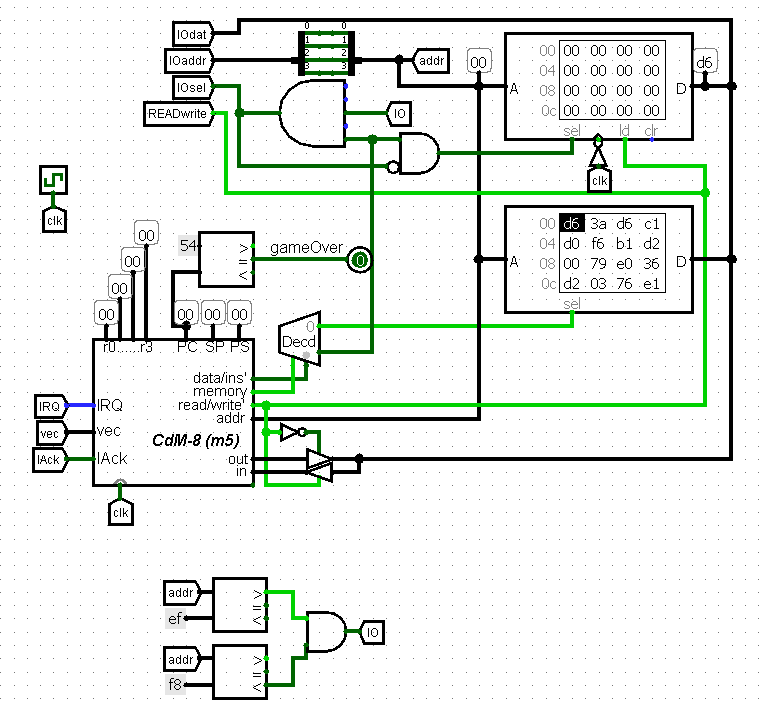
хранение координат



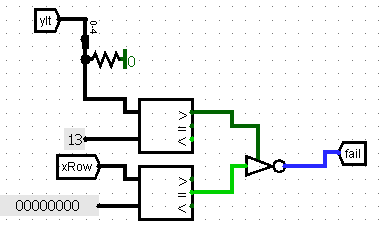
хранится была ошибка или нет, после чтения становится 0



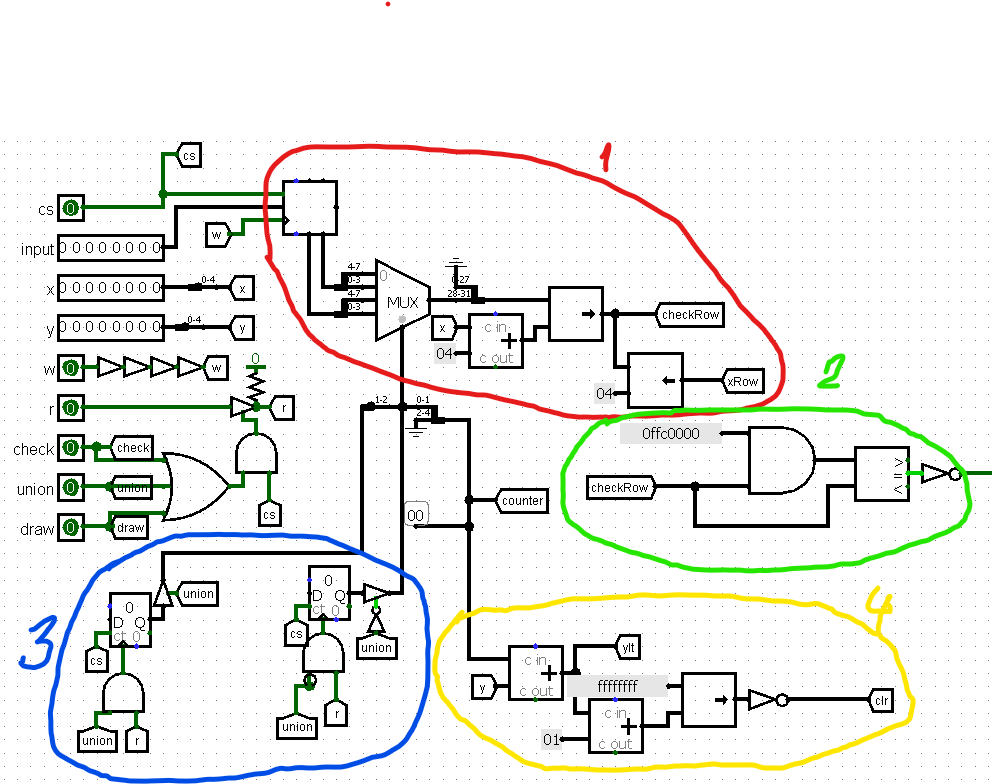
количество заполненных строк и прибавление очков ( очки добавляются в программе) и очистка регистра после чтения



использование CdM-8 (m5)



заставляет фигуру остановиться на дне



Зона 1

одна из строк конечного изображения фигуры

Зона 2

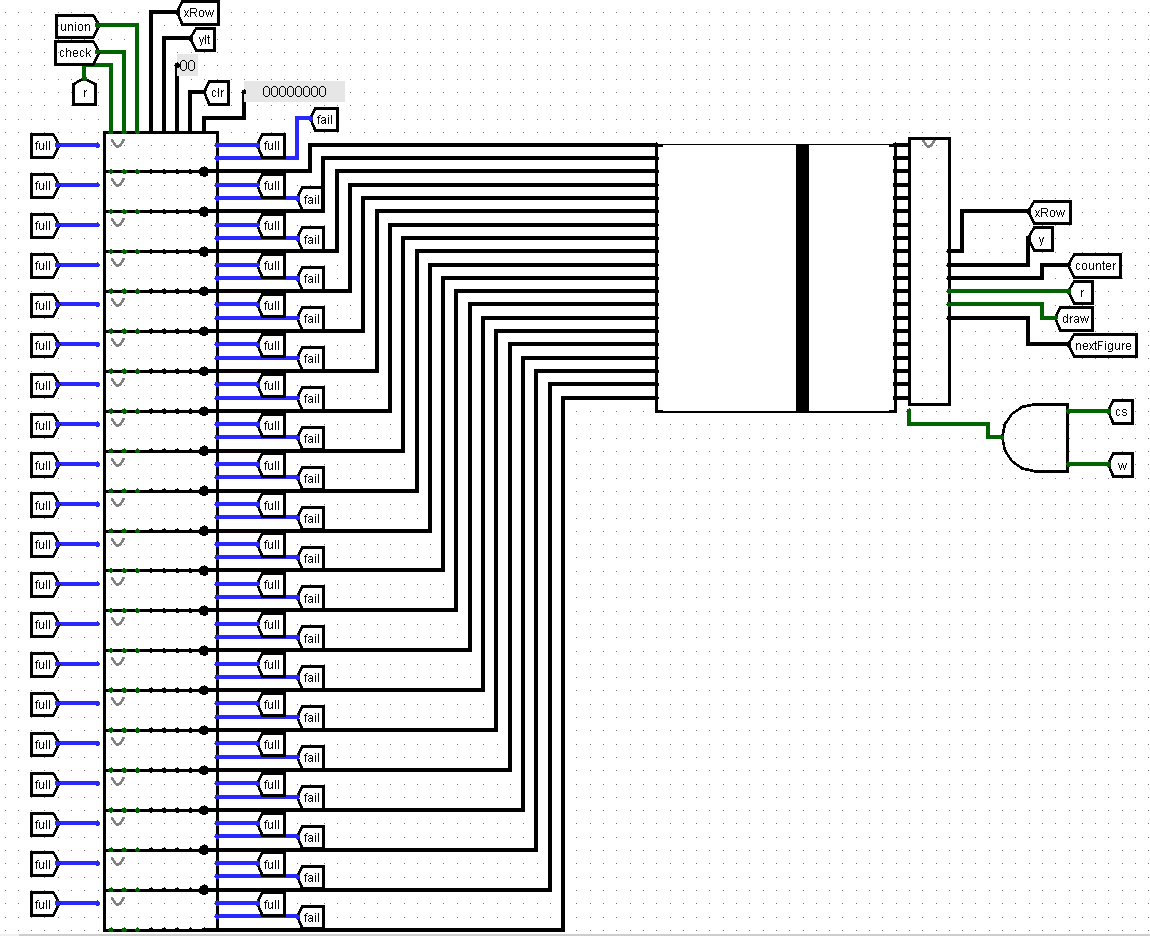
проверка не выхода за пределы битовой маски

Зона 3 ( добавить внутрь перекресток, он урезает сигнал до 2х битного)

Счетчики когда подаётся юнион, то 3х битный счётчик, если нет, то 2х битный

Зона 4

Битовая маска очищаемых полей



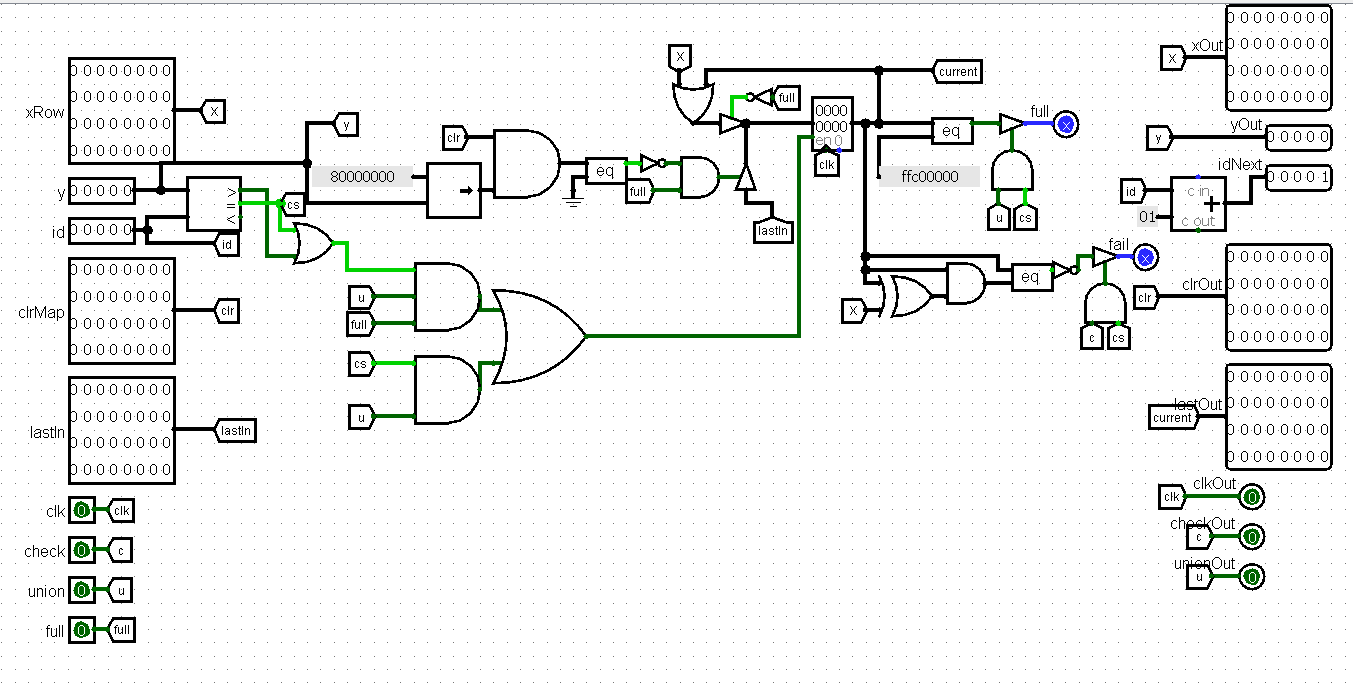
С левой стороны

частичная проверка соблюдений правил игр, проверка на пересечение фигуры и поля, заполненность строки

(про беспроводной дисплей)

С правой стороны

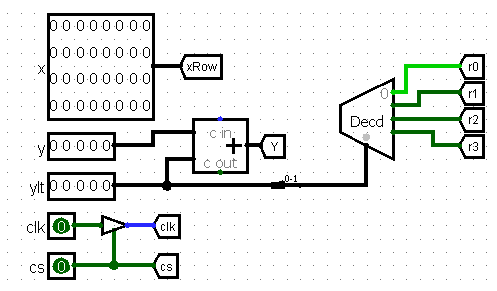
отрисовка линии разделения полей, сама фигура и следующая фигура



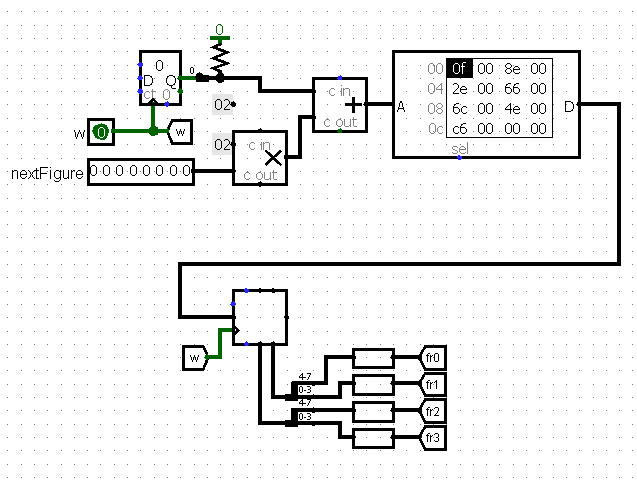
Он сравнивает у и id если, y < id, то при условии, что поданы сигналы full and u, он очищает строку y, если y = id

если регистр совпадает с битовой маской и подняты u and cs, то поднимает full, после поднятия full запускает ещё раз и тогда удаляет строку и сдвигает выше лежащие строки вниз

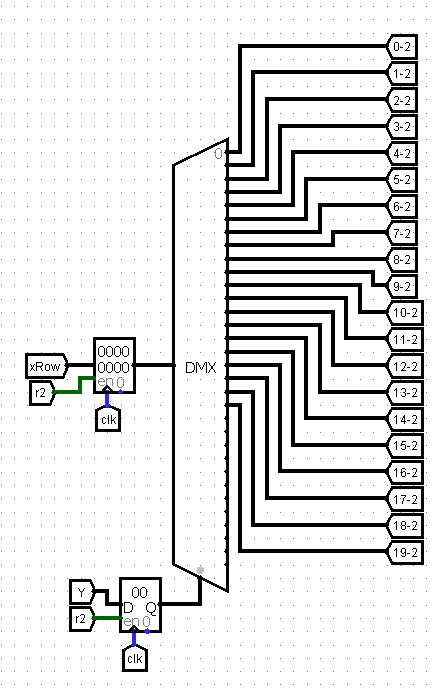
clr очищает в соответствии с битовой маской, так как больше некуда двигаться



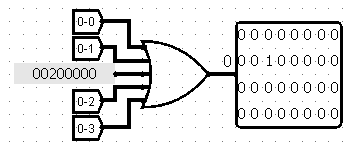
поочередный выбор отрисовки строки фигуры



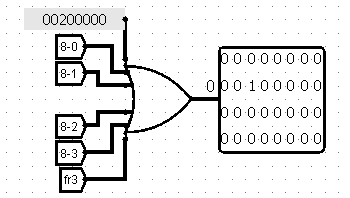
чтение строк следующей фигуры



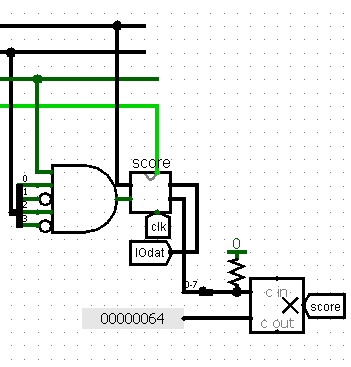
отрисовка определенной строки фигуры на определенной строке поля



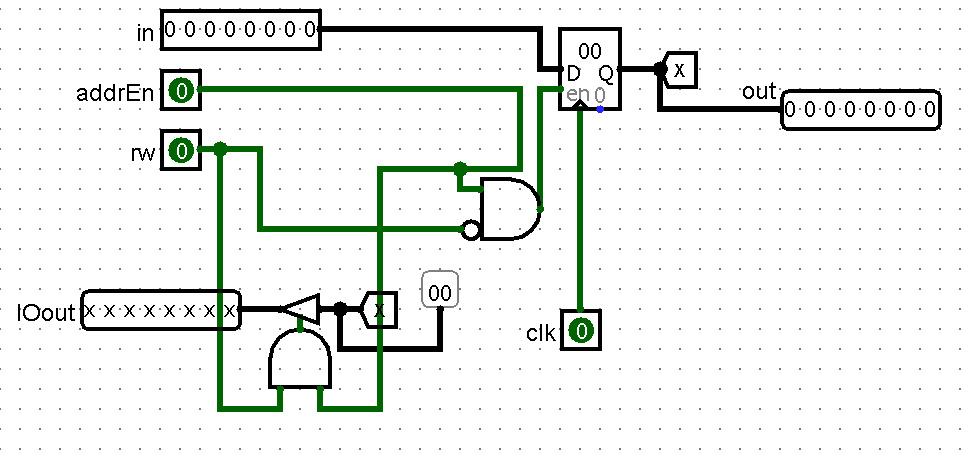
объединяет все получившиеся строки



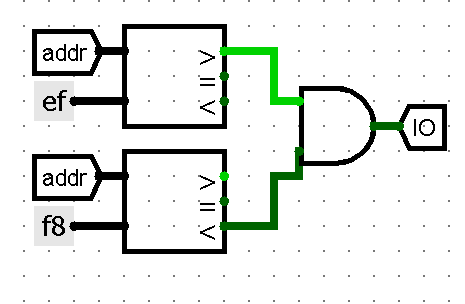
также объединяет получившиеся строки и рисует строку от следующей фигуры



домножает на 100 счет выдаваемый программой



эта схема необходима для связи с CdM-8, она выдает ответ от (автоматки) IOout работает когда происходит чтение, out работает всегда



сверяет адреса, является ли это вводом/выводом