FEDERAL STATE BUDGETARY EDUCATIONAL INSTITUTION OF HIGHER PROFESSIONAL EDUCATION "NOVOSIBIRSK NATIONAL RESEARCH STATE UNIVERSITY"

DEPARTMENT OF INFORMATION TECHNOLOGIES

Group Project B

The discipline "Digital platforms"

**“The Game of TV-Tennis”**

Completed:

Alekhin Nikita

(1st year, group 21216)

Muraviev Dmitry

(1st year, group 21216)

Novosibirsk, 2022

Contents

[Contents 2](#_Toc103525208)

[I Introduction 3](#_Toc103525209)

[II Hardware 3](#_Toc103525210)

[1) Video subsystem 3](#_Toc103525211)

[a) Matrix 3](#_Toc103525212)

[b) Video adapter 4](#_Toc103525213)

[2) Kinematic controller 5](#_Toc103525214)

[a) Input data 5](#_Toc103525215)

[b) Imitation of a human game 6](#_Toc103525216)

[c) Creation of output data for the processor 6](#_Toc103525217)

[d) Displacement of the ball 7](#_Toc103525218)

[e) Bounce from the rackets 7](#_Toc103525219)

[f) Bounce 2.0 8](#_Toc103525220)

[g) Changing velocity 9](#_Toc103525221)

[h) “Unpredictable start” 10](#_Toc103525222)

[i) Score 11](#_Toc103525223)

[3) CDM-8 12](#_Toc103525224)

[III Software 14](#_Toc103525225)

[IV Result 14](#_Toc103525226)

# Introduction

Our team consisting of Dmitry Muraviev and Nikita Alekhin decided to carry out a group project based on the knowledge of development and the ability to model digital electrical circuits using Logisim software, as well as the ability to write code in programming language (Assembler) for the CDM-8 processor.

We have chosen a project in which we need to recreate one of the early arcade video games released in 1972 by Atari. It is a Pong. It was this game that could interest us, because it has an inexhaustible potential for various improvements and improvements.

For the best efficiency of our work, we decided to distribute our responsibilities.

Nikita volunteered to model a digital circuit in Logisim because this topic was close and interesting to him. Dmitry must written records, check the circuit for errors and write software.

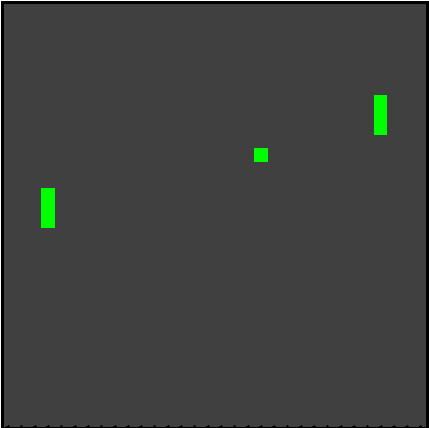
# Hardware

## Video subsystem

### Matrix

First of all we decided to start with a simple one, namely by connecting the display matrix and creating a video adapter.

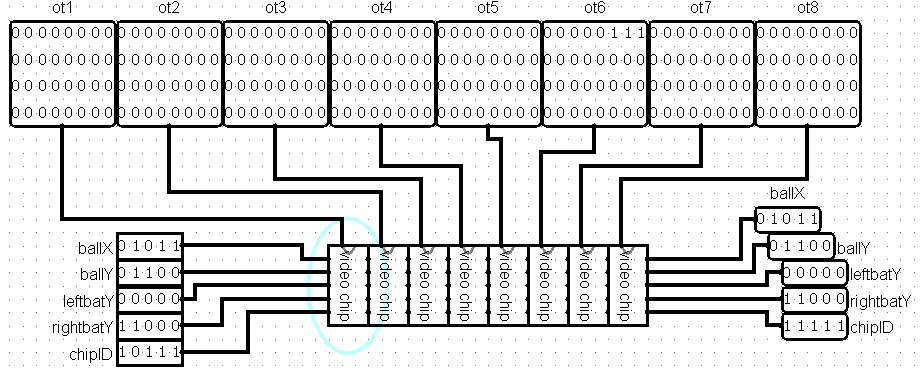
Unfortunately, the maximum possible supported resolution of the matrix in Logisim is 1024 pixels. Because of such a small resolution, there is a problem of "granularity", so the ball is presented in the form of 1 pixel. After consulting, we decided that the best size for rackets at which there will be a better game experience is 3 pixels. The matrix consists of 32 columns; each of them contains 32 pixels. Each column has a 32-bit input. Each bit corresponds to its own pixel.



Picture

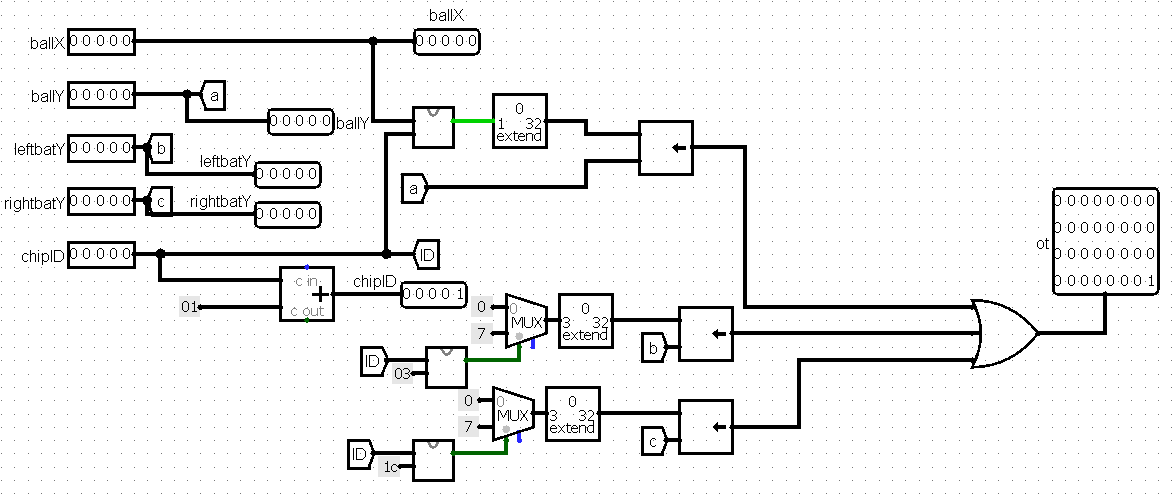
### Video adapter

To correctly display the picture, it was necessary to create a video adapter that accepts four 5-bit inputs: ballX, ballY, leftbatY, rightbatY. Video adapter consists of 32 video chips, arranged in series, and having a connection to 4 inputs.



Picture 2

In order to be able to select the desired column, we also had to connect one more 5-bit wire (ChipID) to the video chips, which on each video chip increases by one. Thus, we were able to number each video chip. If the value of ballX matches the number of the video chip, then the image will appear in the right place.



Picture 3

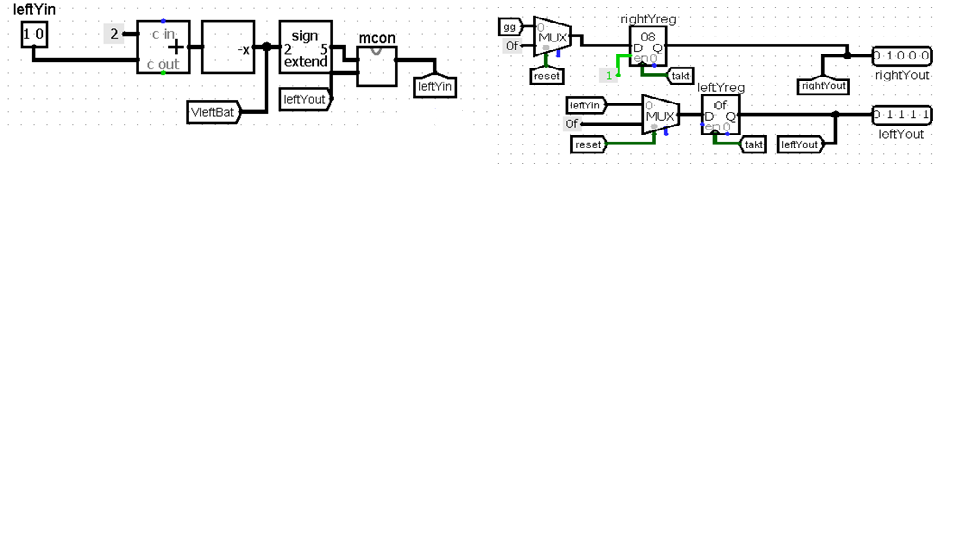
## Kinematic controller

For the correct operation of the system, it was necessary to assemble a circuit of the kinematic controller. The kinematic controller is one of the most complex technical details of our project, so to get to know it better, you need to look at it in more detail. For convenience, his work was divided into several blocks.

1. Input data
2. Imitation of a human game
3. Creation of output data for the processor
4. Displacement of the ball
5. Bounce from the rackets
6. Bounce 2.0
7. Changing velocity
8. “Unpredictable start”
9. Score

### Input data

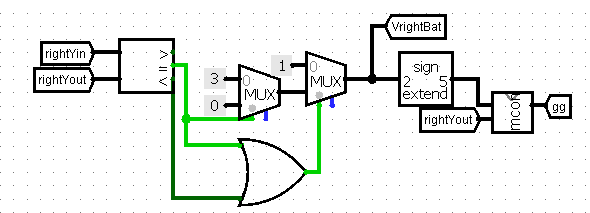
You also need to convert the values given by the joystick and enter the starting values for the position of the ball when the game starts. For the function of the latter is responsible for the scheme shown in the second picture.



Picture

### Imitation of a human game

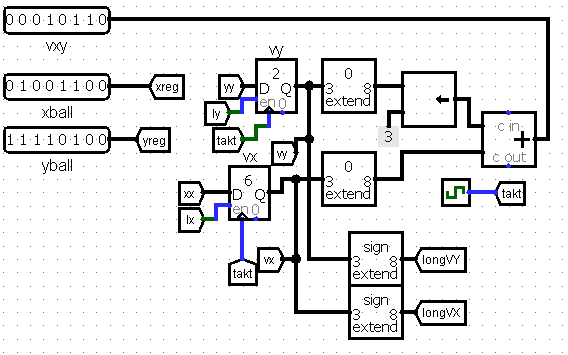
The kinematic controller takes the coordinate of the future position of the ball, to which it is necessary to move the racket in advance. Initially, we wanted to connect directly to the video adapter, and then it turned out that the right racket was "teleported" on the screen, which is why the player considered the game dishonest. Therefore, a scheme was added that could slow down the racket, for smoother movement on the screen, like a living person.



Picture

### Creation of output data for the processor

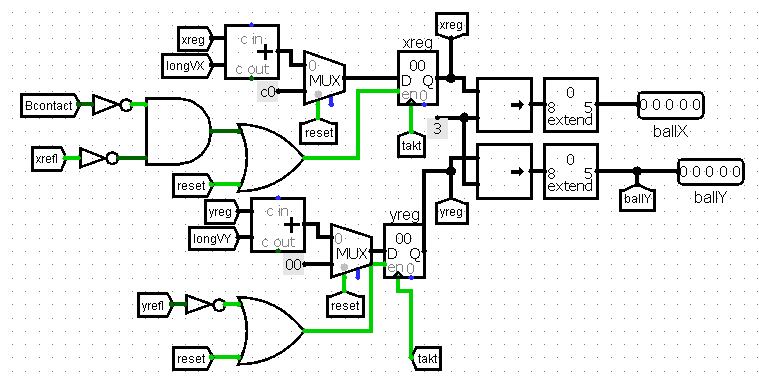
Also initially all projections of the ball velocity are 3-bit values that need to be stored, which registers successfully handle, and converted to 8-bit values that will be needed later. The ball has 16 possible directions. We decided to stop at this number of directions, but if necessary, it can be increased. In the picture below, you can see that another 8-bit value called "vxy" is also created. First-third bits (right to left) mean a vertical change in the position of the ball. Fourth-sixth bits mean a horizontal change of position. These values are necessary for the correct calculation of the future position of the ball.



Picture

### Displacement of the ball

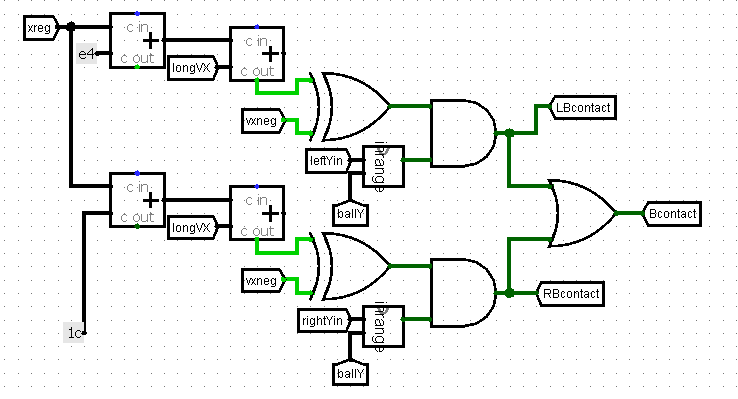
The following scheme, at first glance, seems very complex and cumbersome, but in fact everything is extremely simple. On it, the values of the position of the ball and the projection of its velocity are added, thus, the displacement of the ball in the direction of its velocity is obtained, followed by the transformation of its coordinates relative to the X and Y axes into 5-bit signals for correct display in the video adapter.



Picture

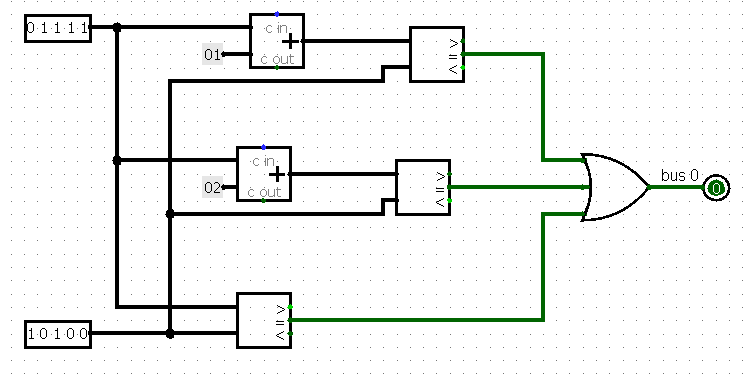
### **Bounce from the rackets**

A new scheme has been added to properly bounce the ball off the rackets, which accepts "xreg", which is an 8-bit value of the ball's position along the X axis, to which is added the column number of each of the rackets. Overflow means that the ball is within reach of each racket. Then the condition checking the equality of the Y coordinates of the ball and the racket. When it is executed, the signal “LBcontact” or “RBcontact” is transmitted, which allows you to understand that the ball indicates contact with the racket.



Picture 8

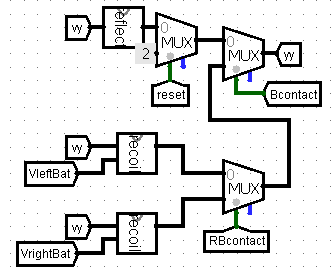
Since the position of the racket is represented as a single pixel coordinate, it was necessary to add an "inrange" scheme that allow the ball to bounce off the entire surface of the racket. The implementation of "inrange" is shown below:



Picture

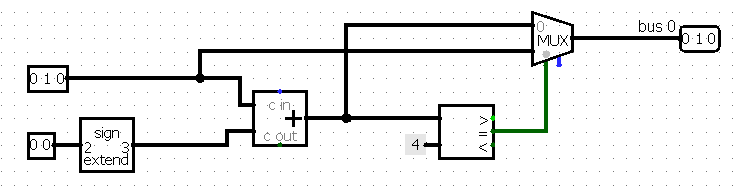
### Bounce 2.0

For a more interesting and varied game, an innovation has been added that should change the velocity of the ball relative to the Y axis when bouncing off a moving racket.

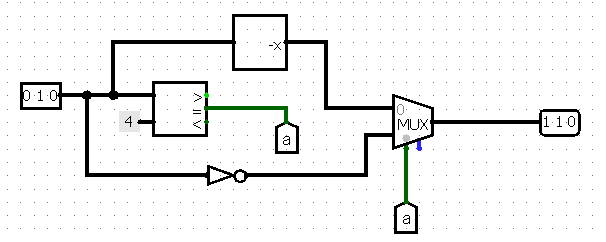


Picture 10

For this, circuits were assembled, marked in the picture as “recoil” and “reflector”, respectively.



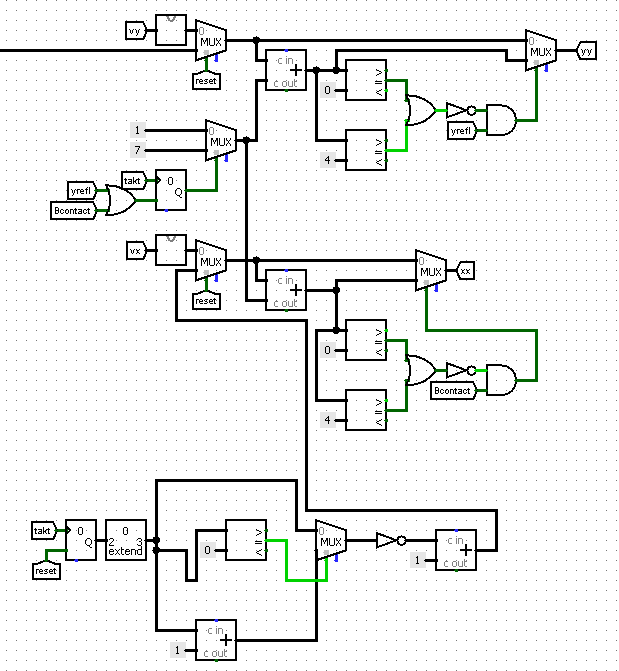
Picture



Picture

### Changing velocity

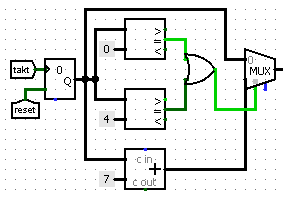
At this stage of development, the change in the speed of the ball occurred only when it was hitting the edge of the racket. These cases were quite rare, so the system needed to be improved. For a more interesting and exciting game, the idea of an additional change in the speed of the ball when it is hit from the racket or the vertical border of the screen was proposed. Dmitry was initially against adding this mechanic because he thought the game would no longer be realistic, but after trying it a few times, he agreed that the game became more interesting and less predictable. Therefore, a random number generator was added, which may add or subtract 1 from speed of the ball when it hit top or bottom of the game field or when it hit the racket, and some conditions that may not change speed.



Picture

### “Unpredictable start”

For even more exciting game play, a new “Unpredictable start” mechanic has also been introduced. From the name you can guess about the function. With is each new round starts with a random speed of the ball. This mechanic allows you to make the game even more unpredictable for the player.

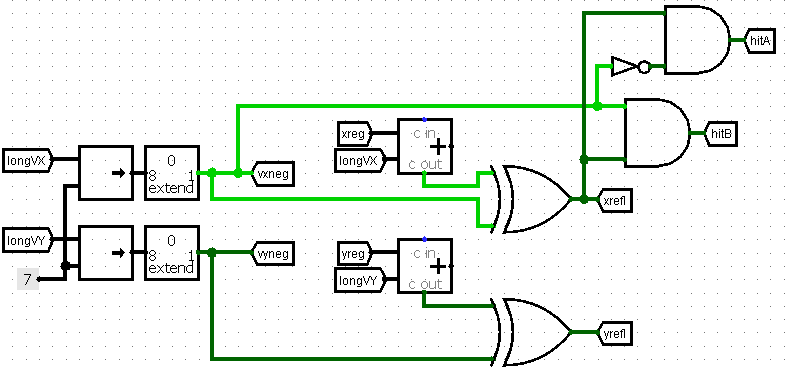


Picture

### Score

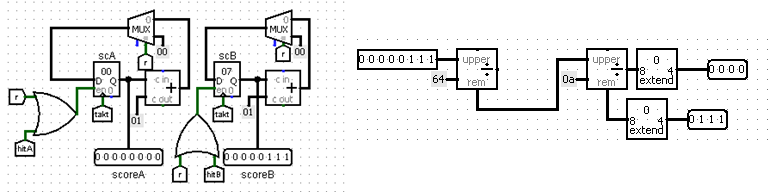
It is also necessary to take into account the condition of a goal scored. Initially, we made a scheme that does not restart after a goal is scored. But it was decided to abandon this game mechanic, because people who tried to play in this way claimed that the game became boring and very simple.

Therefore, it was decided to change the implementation of this concept by adding an automatic level restart.



Picture 15

The kinematic controller also has a scoring function. There is nothing complicated in this scheme. Registers were added to remember the current account and small arithmetic calculations were made to correctly display the result in decimal notation instead of hexadecimal notation.



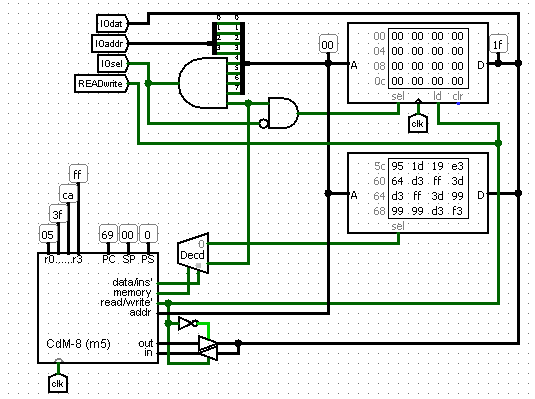
Picture

## CDM-8

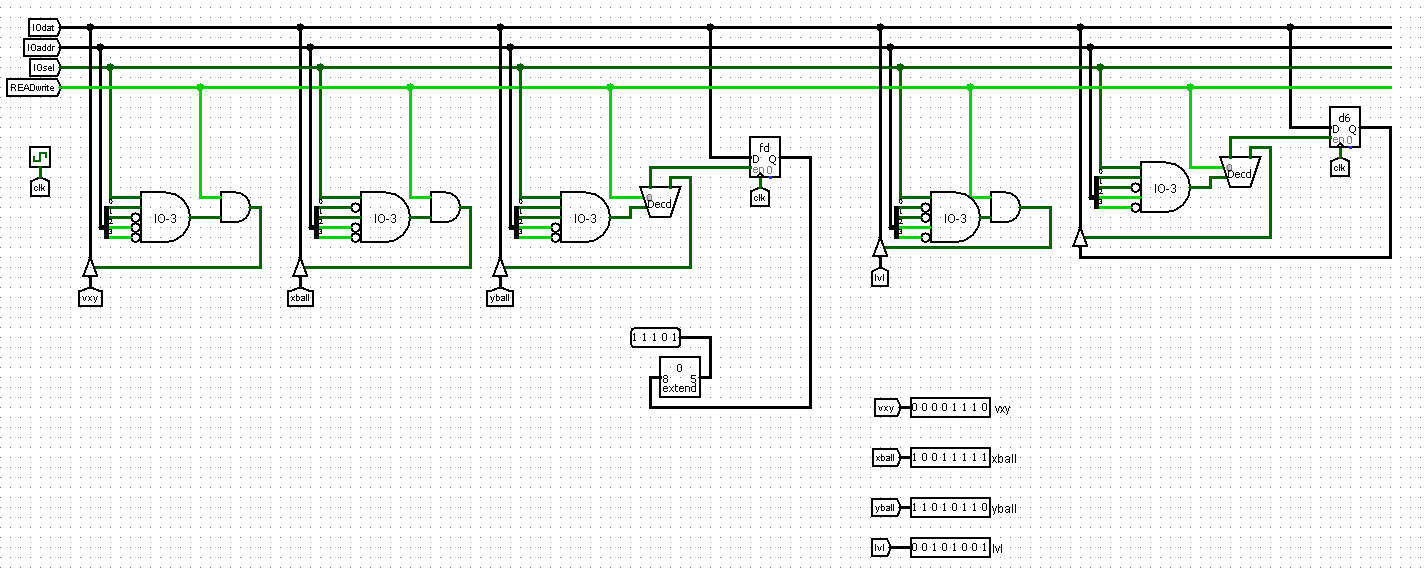
For the full-fledged operation of the game, only a device was missing that could calculate the future position of the ball, for its subsequent reflection with a racket.

The CDM-8 mark 5 was chosen as the processor. This is a rather old processor, the processing power of which was enough for us. There was also the question of choosing von Neumann (Manchester) or Harvard architecture. Their main difference is the way data is stored. The Manchester architecture uses only 1 memory bank, which contains all the values for calculation. The Harvard architecture has 2 memory banks. One stores instruction data, and in the other all the other necessary elements. We used the Harvard architecture. Although it was more difficult to connect, we had a big gain in the amount of available memory, which could be useful in further improvements to our project.

As a result, we got the following scheme:



Picture



Picture 18

The necessary inputs to the processor are

* “vxy” is a value containing the ball speed values
* the ball coordinates are “xball” and “yball”
* “lvl” is a value that allows you to switch the difficulty levels of the game.

Difference between difficulty levels is more accurate calculation of the new position of the ball the new position of the ball.

When you press the top button next to the screen, a light next to it lights up, which indicates that a more difficult level of the game is on. The player has the ability to switch difficulty levels at any time.

# Software

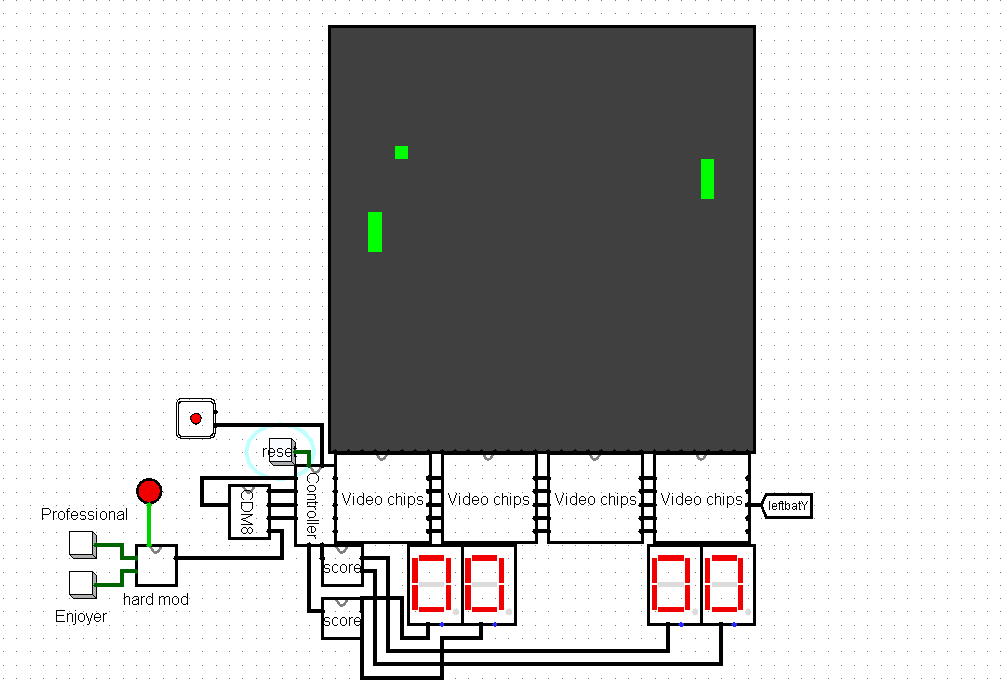
The final step in our project was to write software that could correctly predict the future position of the ball.

We have a closed loop which reads all the input values. Depending on the level of difficulty, the time is calculated, for which the ball from the current position will reach the right racket. After that, the vertical coordinate of the ball is calculated, on which the racket must be located in advance. For counting at positive and negative vertical speed, 2 cases were provided. If the vertical velocity of the ball is positive, then the product of the ball's vertical velocity and time is added to the current vertical coordinate. If the received value is less than 255 (the maximum X and Y coordinate), then the received values ​​will simply be transmitted. Otherwise it will overflow and the resulting value will be equal to the difference between the largest coordinate of the field and the required coordinate. Therefore, in this case, the required value will be equal to the difference between 255 and the received value. If the vertical speed of the ball is negative, then we "flip" the field along the horizontal axis and perform the same operations, if the speed were positive. This trick allowed us to save a lot of memory and reject of writing the same type of code.

# Result

In the end, we managed to simulate the game “TV-tennis” in the Logisim program, in which the following were implemented:

1. A video subsystem consisting of 32 video chips that allows you to correctly display a digital signal on a screen matrix.
2. The kinematic controller, which converts the signal of the joystick controlled by the player, limits the movement of rackets, so that they do not have the ability to “teleport” when they hit the screen. A system was implemented that allows you to support the movement of the ball in 16 directions, the “Unpredictable start” mechanics, changing the speed of the ball when bouncing off the edges of the racket or the boundaries of the field, imitation of the human game by the bot, entering the score of the points received.
3. A computing module was connected, allowing a person to play against a computer, with the ability to select the level of difficulty of the game.



picture