MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER EDUCATION
"NOVOSIBIRSK NATIONAL RESEARCH UNIVERSITY
STATE UNIVERSITY"
(NOVOSIBIRSK STATE UNIVERSITY, NSU)

09.03.01 - Informatics and Computer Engineering
Focus (profile): Software Engineering and Computer Science

TERM PAPER

Job topic: **"The Game of TV-Tennis"**

Komarov Roman
Pronin Nikolay
Zhao Song Tao

Novosibirsk
2024

1 INTRODUCTION

Our project was developed in 2024 as part of the Digital Platforms course at the Novosibirsk State University. It is a TV tennis game created using Logisim hardware and CDM-8 software.

2 PURPOSE AND AREA OF APPLICATION

The TV-Tennis Game project serves as an educational tool to demonstrate the integration of hardware and software components in creating digital entertainment applications. Utilizing Logisim for hardware simulation and CDM-8 for software development, the project showcases the interdisciplinary approach in designing interactive games.

3 FUNCTIONAL CHARACTERISTICS

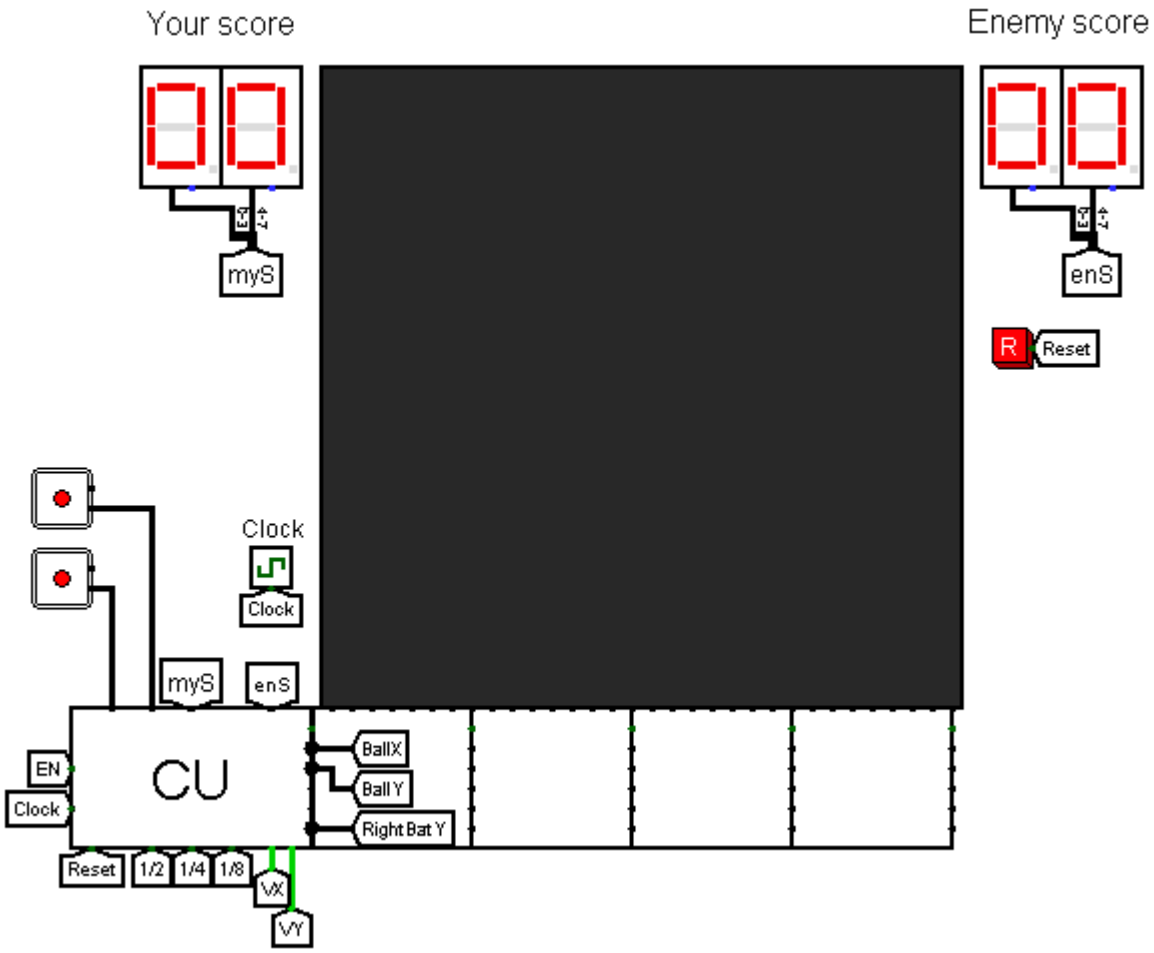
Display

The display is a square with 1024 pixels arranged in a 32x32 grid. Each column of pixels corresponds to a 32-bit input pin, each bit representing the state of the pixel (on or off). To display two bats and a ball on the screen, we need to know the ballX and ballY coordinates, as well as the Y coordinates of both bats. Each bat consists of three pixels: one center pixel, one above the center pixel, and one below the center pixel.

The video system is controlled by 32 video chips divided into 4 sections of 8 chips each. Each video chip is responsible for controlling the pixels in its assigned column, numbered from 0 to 31. The chips are numbered sequentially: the first chip is assigned the identifier 0b00000 and subsequent chips are incremented by 1. In addition, the input of each chip receives signals indicating whether the program is enabled, the X and Y coordinates of the ball, the Y coordinates of the left and right bits, and the ID number (No ID is sent to the zero video chip, it becomes zero with the help of a pull resistor). Special columns in the sections correspond to the position of the bit: column 3 in section 0 indicates the left bit, and column 4 in section 3 indicates the right bit. Both the ball and the bat can occupy these special columns at the same time.

The scoring panels display the players scores from the control unit.

There is also a Reset button next to the screen. It resets the points and puts the bats with the ball in the initial position.



Control unit

The Control Unit (CU) is the most important part of our circuit. This is where such processes as ball movement, ball reflection, bat movement and score counting take place. The Control Unit

receives input data like EN (indicating whether the game is On or Off), the clock and 3 ball speed settings, which we will discuss later.

Bat Movement

First, our bats consist of 4 bits. They track the movement with a joystick. If their value is greater than 8, the racket will move in the other direction, and if it is less, it will move in the other direction. Then we translate to 5 bits, since the bat can take 32 possible positions (2^5 = 32). Also for the left bat a system is implemented where the racket comes back when hitting the upper or lower wall. We also implemented a system for the left bat, when it hits the upper or lower wall, it moves back. We have implemented it this way: if the joystick does not move, the bat stands still as always. If a bat moves, we look if its coordinates coincide with the coordinates of the upper or lower wall and if they do, we either subtract the coordinate if we hit the upper wall or add it if we hit the lower wall. This is how we move the bat. For the right bat this is implemented using CDM-8.

Ball Movement

The position of the ball is changed iteratively. Each time we subtract 1 from BallX and add 1, keeping these values separate. Then we look at the value of VX (the direction of the ball's flight along the X axis). If Vx = 1, it means that the ball is moving to the right and we write the ball coordinate + 1 to the register. If Vx = 0, then the ball moves to the left, and we write the ball coordinate - 1 to the register. Situation with BallY is similar, only the ball moves up or down.

Ball Reflection

We also need to change the values of VX and VY when the ball collides with a wall or bat. So how do we do this? In the case of the wall, we simply compare the coordinate of the ball with the coordinate of the wall (when comparing by X coordinate, the coordinate of the left wall is 0, the coordinate of the right wall is 1f. The Y coordinate comparison is similar) and in case they are equal, we change their value.

With the reflection of the ball from bats, things are a bit more complicated. From the algorithm of calculating the coordinate of the ball we need to take the value of the ball before it is written to the register, so that the ball bounces exactly from the bat and not from inside it. We divide this value by X and Y coordinates into 32 bits, take the coordinates of the centers of the left and right bat, also divide them into 32 bits and construct full-fledged rackets from them using two shifts - left and right. Then we compare the row and the column they are in, and if they coincide, then reflection happens.

Score Count

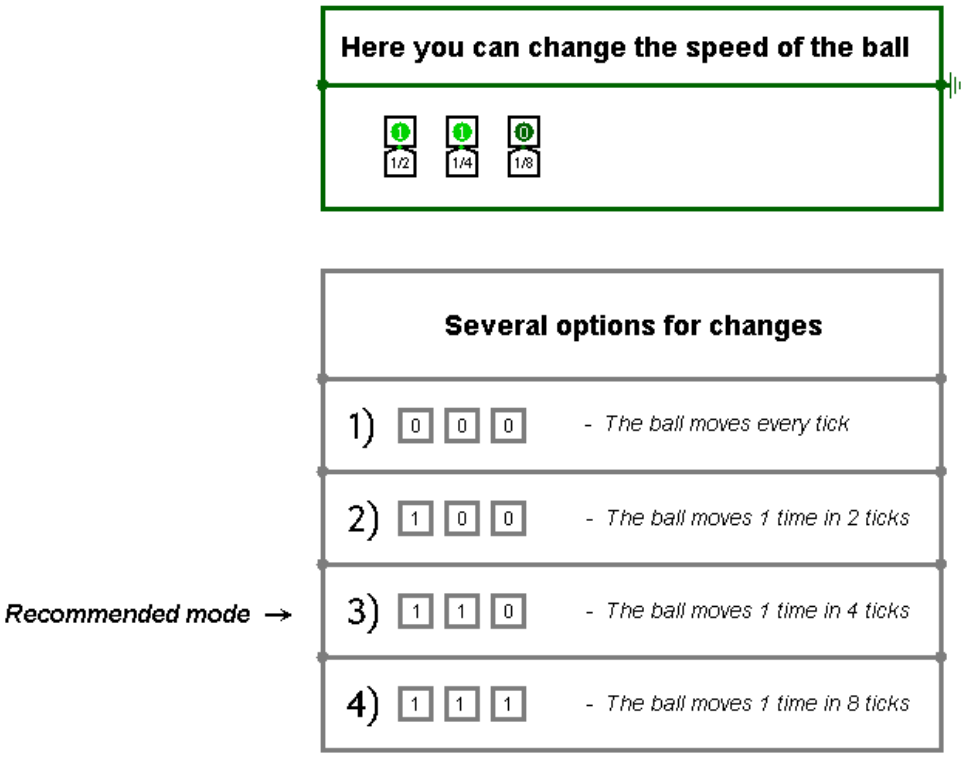
A score is added to the player when the ball hits the wall opposite to his bat. We have implemented it in the same way that when the ball bounces against the wall, we compare the coordinate of the ball with the coordinate of the wall. Then we need to check the value of VX. If it is positive, the ball will definitely hit the right wall (if the racket does not touch the ball) because it moves to the right, and if it is negative, it will hit the left wall because it moves to the left. Therefore, we add points to ourselves when VX is positive and the coordinate of the ball coincides with the coordinate of the right wall, and we add points to the enemy when VX is negative and the coordinate of the ball coincides with the coordinate of the left wall.

Ball Speed

To improve our project, we have implemented a customizable ball speed. That is, the player can set the ball speed with which he will be comfortable to play. The player has 4 variants of ball speed:

1. The ball moves every tick.
2. The ball moves once in 2 ticks
3. The ball moves once per 4 ticks.
4. The ball moves once per 8 ticks.

This setting with a memo of all possible ball speeds is located on the main screen in a convenient menu.



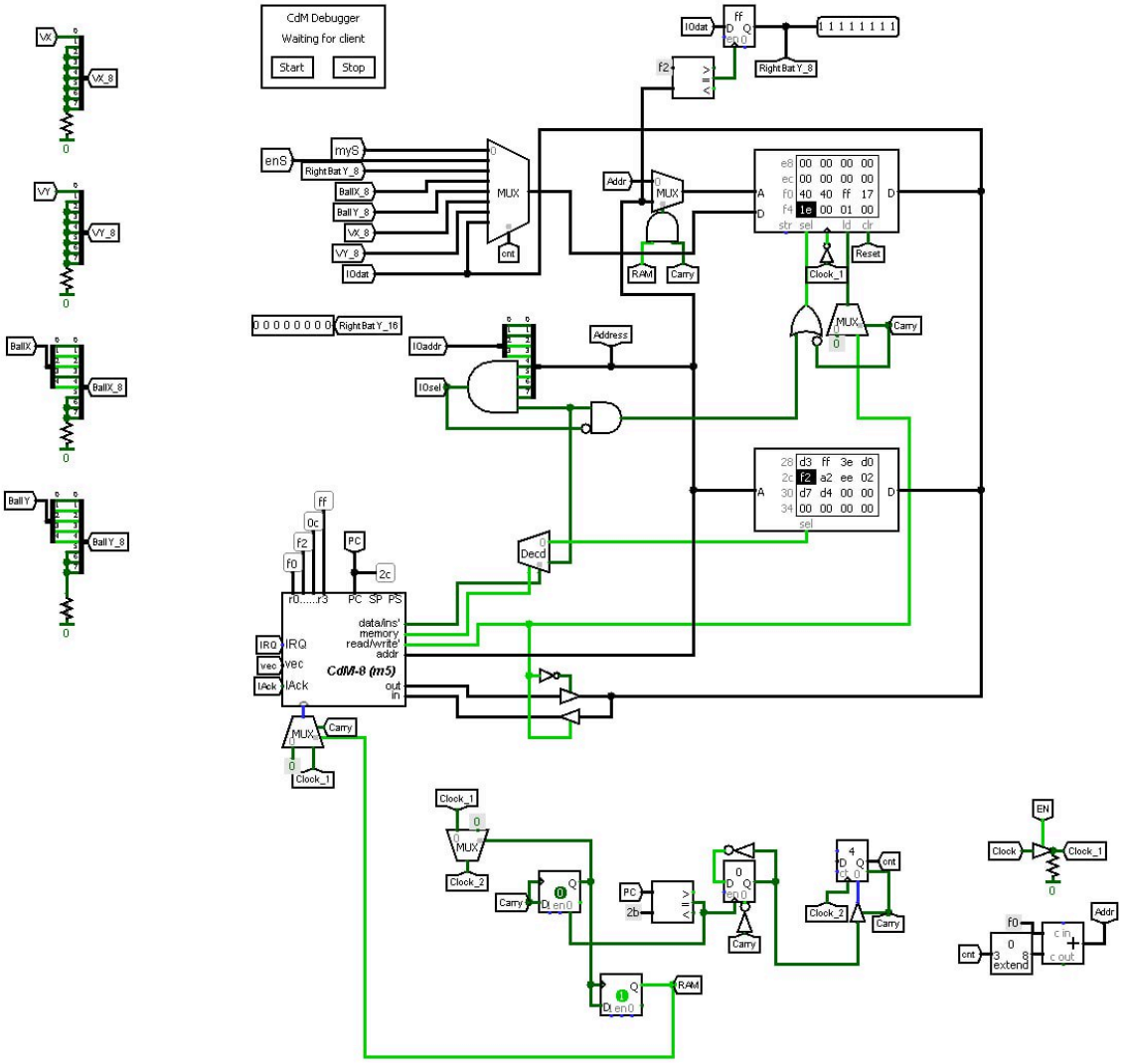
We used D Flip-Flop to implement the change of ball delay. When the user presses in sequence, each time the delay is multiplied by 2, because each time another D Flip-Flop is connected.

Video chip

Now let's explain how we display all of this on the screen. From the Control Unit we send to the video chip the EN value, the x and y coordinates of the ball and the y coordinates of the left and right bats. In the Video chip we construct our bats also with the help of shifts. We display them only if the Id number is 0 or 3 for the left and right bats respectively. We also consider the EN value so that the bats are not displayed when the game is off. The ball is also only displayed on one chip. To find out which chip the ball is in, compare the last 2 bits of the ball with the ID. The first 3 bits are responsible for which column of the video chip to display the ball in. Depending on this, we use the y-coordinate of the ball and display it in the corresponding column. In column 3 and 4 we also output the bat.

4 TECHNICAL CHARACTERISTICS

Software



We created a bot player to control the right bat. It runs on a CDM-8 processor using Harvard connectivity architecture.
We transfer the value of the counters, the coordinates of our bat, the coordinates of the ball and the direction of its movement to the data bus. Depending on the direction of the ball's movement, we read or write data.

5. Conclusion

In conclusion, our TV-Tennis Game project represents the successful integration of hardware and software components to create a quality interactive gaming experience. Every aspect of the game has been carefully designed and implemented to provide players with an enjoyable gaming experience.

There is potential for further development and improvement of the TV-Tennis Game in the future. This could be moving the bat with the keyboard, changing the angle of the ball bounce depending on where the ball touched the bat, small edits to increase the performance of the project, etc. We believe that we were able to fulfill the requirements for this project and also managed to improve it.

6 SOURCES USED IN DEVELOPING

- "Computing platforms" (tome.pdf) - A.Shafarenko and S.P.Hunt.
- http://www.cburch.com/logisim/docs/2.6.0/ru/libs/index.html - Website with description of all elements and tools in Logisim.
- http://ccfit.nsu.ru/~fat/Platforms/ - All materials for the course "Digital Platforms", including lectures.