Good afternoon. Let me introduce our team. These are Bespalov Sergey, Grischenko Alexander and I am Solopov Ilya. The aim of our presentation is to give you information about our project. If you have any questions, feel free to ask us at the end of the presentation.

Shoot 'em up is a type of video game in which the player controls a spaceship with the aim of destroying enemies with gunfire. Enemies in such games are usually various aliens or monsters that attack the player by shooting or otherwise.

The aim of this project was to design and build a Shoot 'em up game on logical circuits using the CdM-8 processor and its assembly language.

And it was necessary to solve the problems. You can see them on the screen.

Let’s turn to the next slide. The functional requirements are presented on the screen. They were implemented by combining hardware and software. We would like to describe how it was made. Let’s go through the hardware part.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Player movement**

A decoder is used to move the player. The value from the decoder is shifted to the left and to the right, then with the help of the “OR” operation it is fed to output 0. The value without shifts is fed to output 1.

**Player’s shooting**

A decoder is also used for shooting. Its value is given in a column depending on the position of the cannon. The value fed to the decoder is incremented every clock.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Enemy movement**

Enemy movement is implemented with counter. Depending on the value of the counter, lines of enemies are passed through shifters.

**Monster shooting**

Monster shooting is similar to the player's shooting. The coordinate of the shot is calculated from the height of the enemies and a random number processed by the processor.

**Destruction of the monsters**

Let’s define a layer term.

Layer is a 1024-bit value used to display game elements on the screen.

Bullet impact on aliens is a logical "AND" between the rows of the player's bullet layer and the monster layer. The result and the alien status byte are sent to CdM-8.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Win and lose conditions**

The player needs to destroy all enemies to win. If an enemy bullet hits the user, then this is a loss. Also, if the aliens reach the walls, then this is a loss.

Let’s switch to the software part.

**Wall destruction.**

This is an algorithm that implements the initialization and destruction of protective walls when monster bullets hit them. The inputs are a string with the current state of the walls and a string of a bullet layer. The output data is a string with the state of the walls after hit.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Monster shooting**

This is an algorithm that implements the calculation of the coordinate in which the monster shoots from the random number of the shooting enemy.

**Hit**

This is an algorithm that implements the initialization of the state of the enemies and processing a new enemy state byte from the old one and the player's bullet layer string.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

As a result of the work done, we created the game "Rush space" in the Shoot 'em up genre. The problems were successfully completed. We were able to implement all the indicated functional requirements. Creating the project, we got knowledge in the field of making logical circuits, working with a processor and its programming, writing project documentation and experience in working in a team.

Let’s move on to demonstration our game.

Thank you for your attention. Have you got any questions?