***Department of Electric & Electronic Engineering,***

***Boğaziçi University***

**FPGA BASED SPACECRAFT GAME**

Mücahit Ekinci: 2015401231

Ömer Avcı: 2016401153

Şenol Mutlu

30.05.2018

***Table of Contents***

[1 INTRODUCTION 3](#_Toc119830505)

[2 PROBLEM STATEMENT 3](#_Toc119830506)

[3 RELATED BACKGROUND 3](#_Toc119830507)

[4 DESIGN 3](#_Toc119830508)

[4.1 Definitions, Acronyms and Abbreviations 3](#_Toc119830509)

[4.2 Subsection\_1 3](#_Toc119830510)

[4.2.1 Subsubsection\_1 3](#_Toc119830511)

[4.2.2 Subsubsection\_2 3](#_Toc119830512)

[5 RESULTS 4](#_Toc119830513)

[6 CONCLUSION 4](#_Toc119830514)

[7 REFERENCES 4](#_Toc119830515)

[A READ ME 5](#_Toc119830516)

[B USERS’ MANUAL 5](#_Toc119830517)

# INTRODUCTION

In the project,a spaceship tries to escape from meteors and destroy them by a beam weapon. The number of meteors destroyed is counted by the seven segments of FGPA. Furthermore, when the spaceship collides with any meteors, the game finishes.

# PROBLEM STATEMENT

First of all, we try to construct a spaceship. To do it, we think the screen as two dimensional analytic space and define an interval by parabolic inequalities. Beam weapon would be continuous laser so creating it is straightforward when the user pushes the button.

Secondly, meteors move to left from right and speed of meteors is increasing with time. There are many problems related to meteors initially. First problem is about creating moving meteors. Second problem is about destroying meteors. In the problem, on the initial demo, meteors are destroyed only the firing button is pushed. When the button is released, the meteors turn back. Third problem is counting how many meteors are destroyed. In the problem, on the initial demo, seven segments rise more than one when beam weapon button is pushed to destroy any meteors. Forth problem is guaranteeing that any meteors are never entered back and forward porches. Fifth problem is about hardness about the game. Hardness of the game should increase.

# RELATED BACKGROUND

The project is required to know basic VHDL computer language and become familiar with FPGA.

# DESIGN

## Movement and Creation of Spaceship and Meteors

To create all objects, the screen is considered as two-dimensional analytic space and all objects are defined by mathematical inequalities.

To create spacecraft, a rectangle is constructed and inside the rectangle, the spaceship is defined by parabolic inequalities. On the other hand, there are three type of meteor shape which are rectangular, circular, and like a bullet. Rectangular shape is defined by two vertical and two horizontal line, circular shape is defined by a center point and radius on two dimensional analytic space, and meteor like bullet is defined by two circle equation and one rectangular equation.

Spacecraft and meteors are moving objects. Spaceship moves just vertically, meteors move only horizontally. Vertical boundary conditions of the spaceships have two parameters which are “upper\_y” and “lower\_y”. The parameters increase or decrease the same amount when the user pushes up or down buttons. On the other hands, location and spend of meteors do not directly depend on buttons. Location of meteors are initialized by programmer. The game starts with low speed meteors. In time, the speed can be three times of initial speed to make the game harder. Additionally, when speed of meteors become two times of initial meteor speed, new meteors set come into screen to surprise the player; that is to say, to make the game harder.

#### 

*Figure 1: Movement of Meteors and Spacecraft*

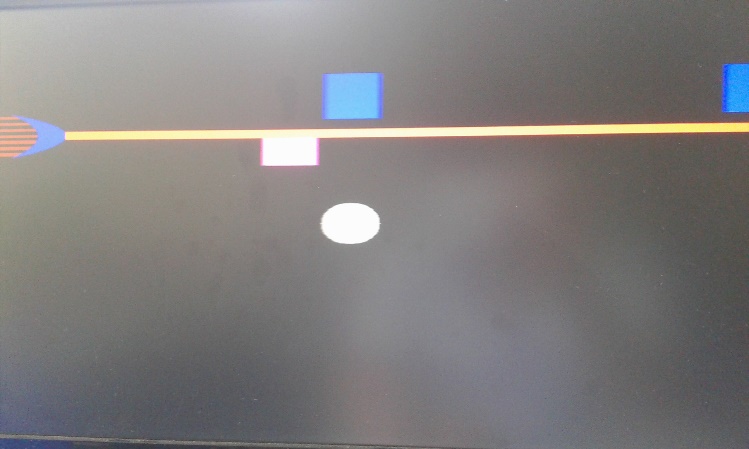
## Beam Weapon, Destroying Meteors and Counting The Number of Destroyed Meteors

Beam weapon is a continuous line and it is activated when the related button is pushed.

The main problem about destroying meteors is that meteors are destroyed only when beam gun is activated. In other words, when beam gun is deactivated, destroyed meteors come back into screen again. To solve the problem, programmers define a special parameters for each meteor. The all parameters set into zero initially. When beam gun activated on proper location, the associated parameter value increases and by the change of parameter, meteors can disappear when beam weapon is deactivated.

Algorithm of counting the number of destroyed meteors is like destroying meteors.

First of all, there are two four bit signal vector on related code. Each signal vector is used to determine related seven segment value. The main problem is that beam gun is fired on suitable location, the vectors rise more than one. To overcome the problem, special parameters are defined for each meteor and all the parameters are initialized as zero. Counting occur only related parameter is zero and if counting emerges then related parameters increases. Therefore counting occurs only once for each meteors if player shot that meteor.



*Figure2: Fired Beam Gun*

### End of The Game

The game finishes only when any meteor strikes the spacecraft. On the end of the game, “GAME OVER” is written into screen. “GAME OVER” is created by rectangles which are defined by mathematical equations. To finish the game, a parameter is used, “finisher”. Initially, finisher value is zero and all codes except codes of “GAME OVER” will work properly only if value of finisher is zero. Losing condition is mathematically defined and when a meteor hits spaceship, finisher value increases. Therefore, game over written into screen.



*Figure3: The End of Game*

### Subsubsection\_2

# RESULTS

We designed a game which you play as a spaceship. There are meteors incoming from right to left. You can move vertically also you can fire a laser beam to destroy incoming meteors. Your score is how much meteor you destroyed.

# CONCLUSION

Our game was making a spaceship game with a laser beam. We achieved that goal. In our game player controls a spaceship, tries to escape from meteors and destroy them with ship’s laser weapon. Meteors are coming as waves. While waves are increasing the speed of meteors are also increasing so the game is hardening. When making the game we encountered optimization problem. The game got bigger and more complex in making process. Recently the program’s synthesis become too long so it enables us to make much update in short time. At some point of project we took an error which says that we are beyond the limits of FPGA.

So we will optimize the program for making new development. In future, we think to add more meteors and features to make the game more fun to play. We actually made meteors which don’t disappear when shot by beam. It provides a hardness in game because player must avoid it by moving. Then we removed it since it is not suitable for proposal of project.

So in future we will try to add more features to game for increasing its gaming performance.

# REFERENCES

W. Press, S. .Teukolsky, W. Vetterling, B. Flannery. Numerical Recipes in C, Cambridge University Press, 1994.