Embedded System Design - Advanced Course
Project Proposal

(EDAN85)

Javier Macotela Lopez, fr6577ma-s@student.lu.se
Sergio Benages Montolio, se5602be-s@student.lu.se

September 4, 2017

1 Introduction

The Embedded system design - Advanced Course is a project-based course, a game is proposed due to the mix between software and hardware that will be implemented. The game aimed to replicate is Duck Hunt, a famous Nintendo game developed in the early 80's.

The reason why this game was selected is the gameplay that offers, clearly different from the rest. Although the graphics will differ, the experience is what will be intended to be copied. This game used to work with a special controller called Nintendo zapper. Figure 1 shows a screen-shot of how the game looks like for reference purposes.

Figure 1: Duck Hunt game

2 System Architecture

In this section, the project architecture will be explained in broad terms since they can be committed to changes during the project's development. Figure 2 shows the general structure that the architecture will have.

The main tool will be a FPGA Nexys 4, but not all its modules will be used. For output a display will be needed, and for input a custom made zapper will be used. The zapper communicates through XADC to the MicroBlaze, sending data in an asynchronous way. The MicroBlaze will connect through an AXI bus to the VGA controller and this will communicate to a BRAM to load the scenarios. Finally the VGA controller will send the output to a display so the user can play.

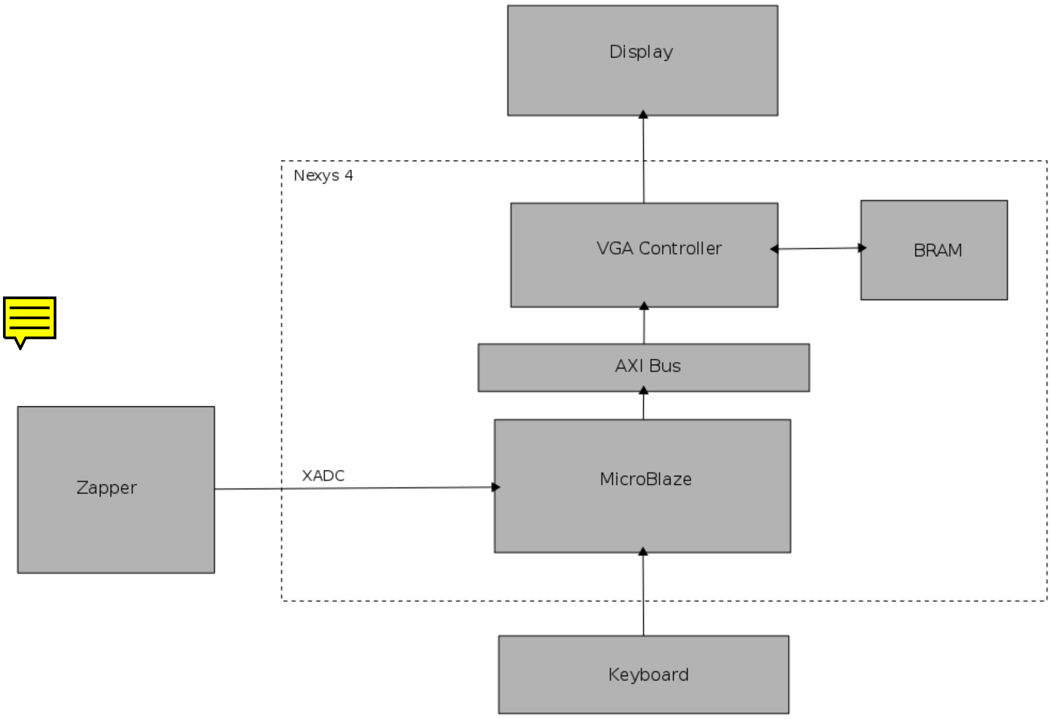

Figure 2: System overview

2.1 IO

In this section, the input and outputs will be explained, this is the most important part of the project because if they do not work correctly the gameplay and experience will fail.

2.1.1 Input

As mentioned before, the system will have a Nintendo zapper imitation as its main input, but as well a keyboard or a mouse might be necessary for different features in the game, like setup for example. For explanatory purposes an original Nintendo zapper is shown in Figure 3.

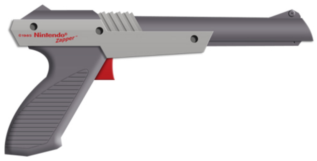

Figure 3: Nintendo zapper

The zapper is compound mainly by a photo-transistor and a button, that when pressed, it closes the circuit and lets the photo-transistor detect if the target was hit. To detect if the target was hit, the photo-transistor has to be aiming at a white area while the non-target area will be black, this way the zapper will be able to detect a voltage change when the shot is successful. This will use the XADC port in the FPGA as shown in Figure 2.

2.1.2 Output

The output will be shown mainly in a display using VGA.

2.2 FPGA

The rest of the game will be handled in the FPGA which will be a Nexys 4 development board. It has a MicroBlaze and a VGA controller. Here, a mix of software and hardware implementation will be done in order to get a functional project.

2.2.1 VGA controller

The VGA controller will be necessary to generate the graphics, here is where the most work will be focused, to set scenarios and scores, the RAM will be necessary as well.

2.2.2 XADC

The FPGA includes analog input ports that will make the detection of the zapper lectures and allow this project to work properly, some calibration will be needed but overall this protocol will make the communication very simple.

2.3 Software

The software will be done using C++ and will control the input and outputs of the game as well as hardware interactions. It will be implemented in the MicroBlaze.

3 Time Planning

The time planning will be divided between the 2 members and are expected to be adequate for each part of the project which will be divided into stages as well. Figure 4 shows the time planning and work division.

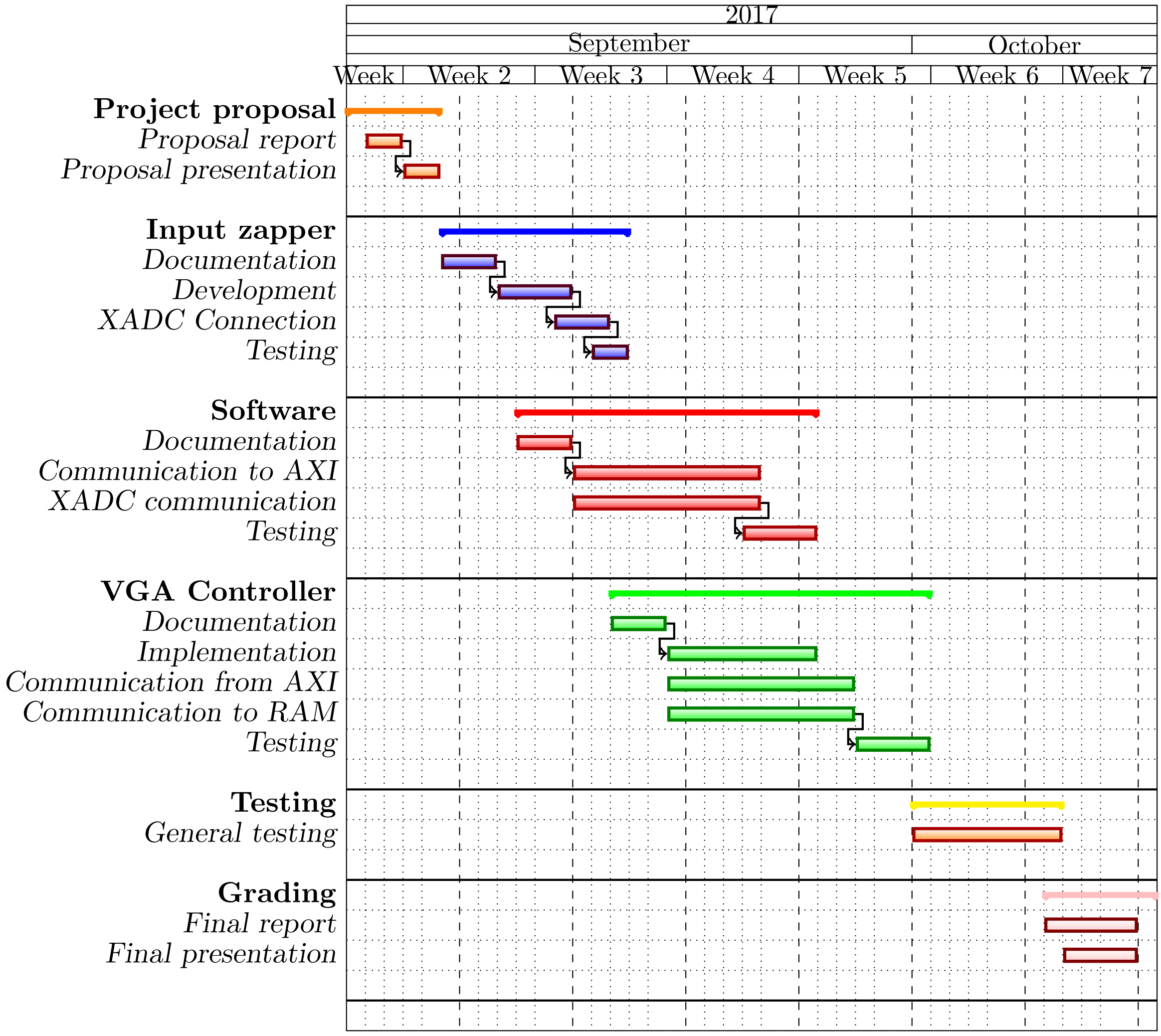

Figure 4: Project time planning