DLD Project Proposal

Ali Asghar Kerai (ak06857)
Areeb Adnan Khan (ak06865)
Hamad Abdul Razzaq (hr06899)
Lakshman Maheshwari (lm06435)

Registered for: Section T-1

1 Project Idea A

1.1 Overview

For this course, our team proposes to make an Air Hockey Game. This game will be played by 2 players, each of them controlling their paddles with the help of a joystick. The paddles and the puck (a circular disk) and the whole game environment would be displayed on the screen of a computer, laptop or any other displaying device. Joysticks will enable the players to control their paddles and the movement of joystick will be synchronized with the paddle movement on the screen.

1.2 Input Peripheral(s)

In order to play the game and make it user interactive, the players will be using the joysticks to control their virtual/remote paddles on the screen. Our paddle on the screen will shift depending on the direction in which the joystick is moved. Moreover, we may also add a feature in which, the faster the joystick is moved, the faster the paddle will be moved on the screen and the stronger the strike to the puck will be made.

1.3 Output Peripheral(s)

For our output, as mentioned in the overview, we will be displaying the whole game environment on the screen (i.e. VGA Display). And this will not be a fixed display rather it will be motioned displayed (animation) in which the paddles and the puck will be moving in accordance with the input (moving joystick in a certain direction) by the players.

1.4 Block Diagram

The Block diagram for our project idea is as follows:

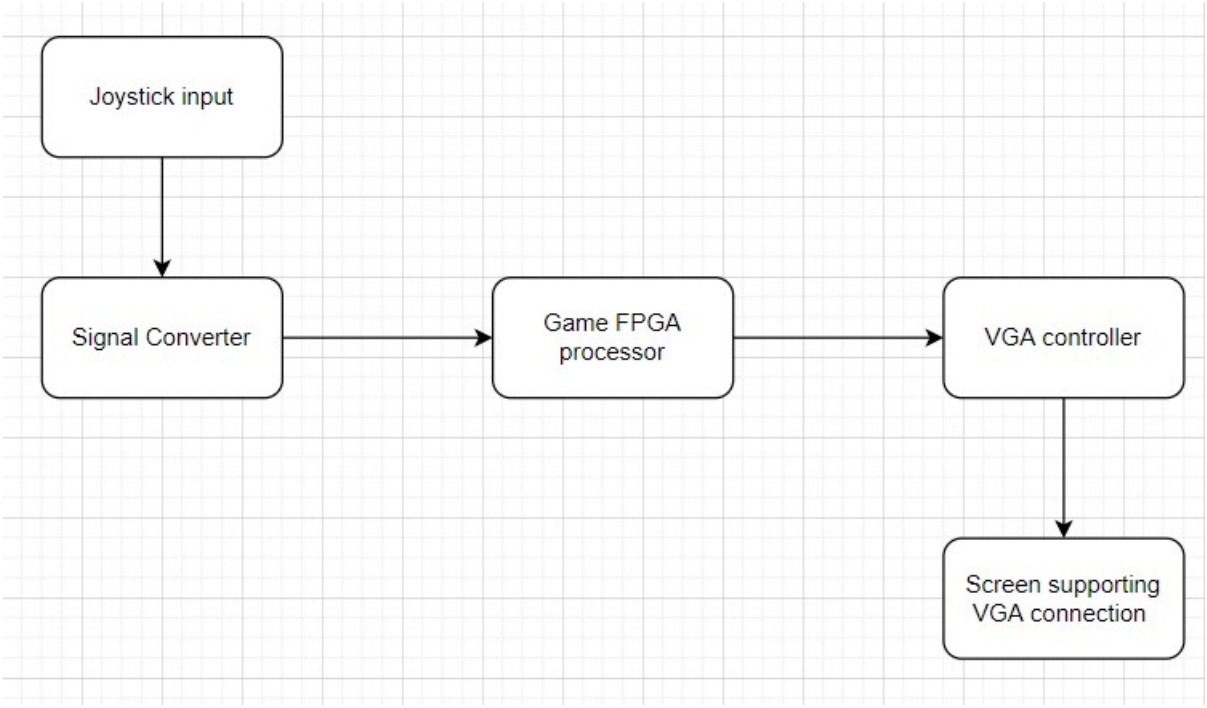

Figure 1: Block Diagram for our design

The details of each block is as follows:

1.4.1 Joystick Input

This is the part of our Digital interaction where the user will be interacting with our implemented design. Here, both the players will use their own respective joysticks to control the paddles on screen.

1.4.2 Signal Converter

As we know, the input taken from the joystick will most likely be an analog signal. In order to work on it (using programmable logic), we have to convert it into a digital signal. For this purpose we will be using an appropriate converter(ADCs for example) that will get this job done for us. We may also be programming a Finite State Machine to make the signal digital.

1.4.3 Game FPGA Processor

This block of our design is the core block which will be doing the work of programming logic in accordance with how the game is played. As an overview, the game works such that both the players have paddles with the help of which they will strike the puck. The players will keep on playing until the puck goes into the goal of one of the players and the goal scorer will gain a point. This logic will be implemented here in game FPGA processor. It will include FPGA, Finite State Machine(s), etc. for the implementation of synchronous sequential logic.

1.4.4 VGA Controller

This block in our design is responsible for the VGA Display. It'll generate the VGA logic for us (not on screen but the logical output). Initially we will set the logic such that the paddles are at the bottom center of either side of the playing area and the puck is at the center. Then in accordance

with the direction in which the joysticks are moved, our logic will be implemented in such a way that it synchronously updates the screen.

1.4.5 Screen Supporting VGA connection

In this block, the VGA logic generated from "VGA controller" block will be connected to an appropriate screen supporting VGA (LED, LCD, etc.) which will display the VGA output. Thus the GUI will be displayed with the help of the screen that will be used in this block.

1.5 Proposed Model & Sample Prototype

Following images gives a rough idea of how our Design will look like. To be more precise, the GUI that can be seen in Figure 2 demonstrates the basic functioning of our virtual programmed interface. The basic structure of our gaming screen will be same but the design, theme, size of paddles and puck etc. may be changed. Similarly, referring to Figure 3, The joystick would be similar to this but then again, based on the commercial availability and budget constraints, our joystick may change but it will function in the same way as the one shown in figure and demonstrated above.

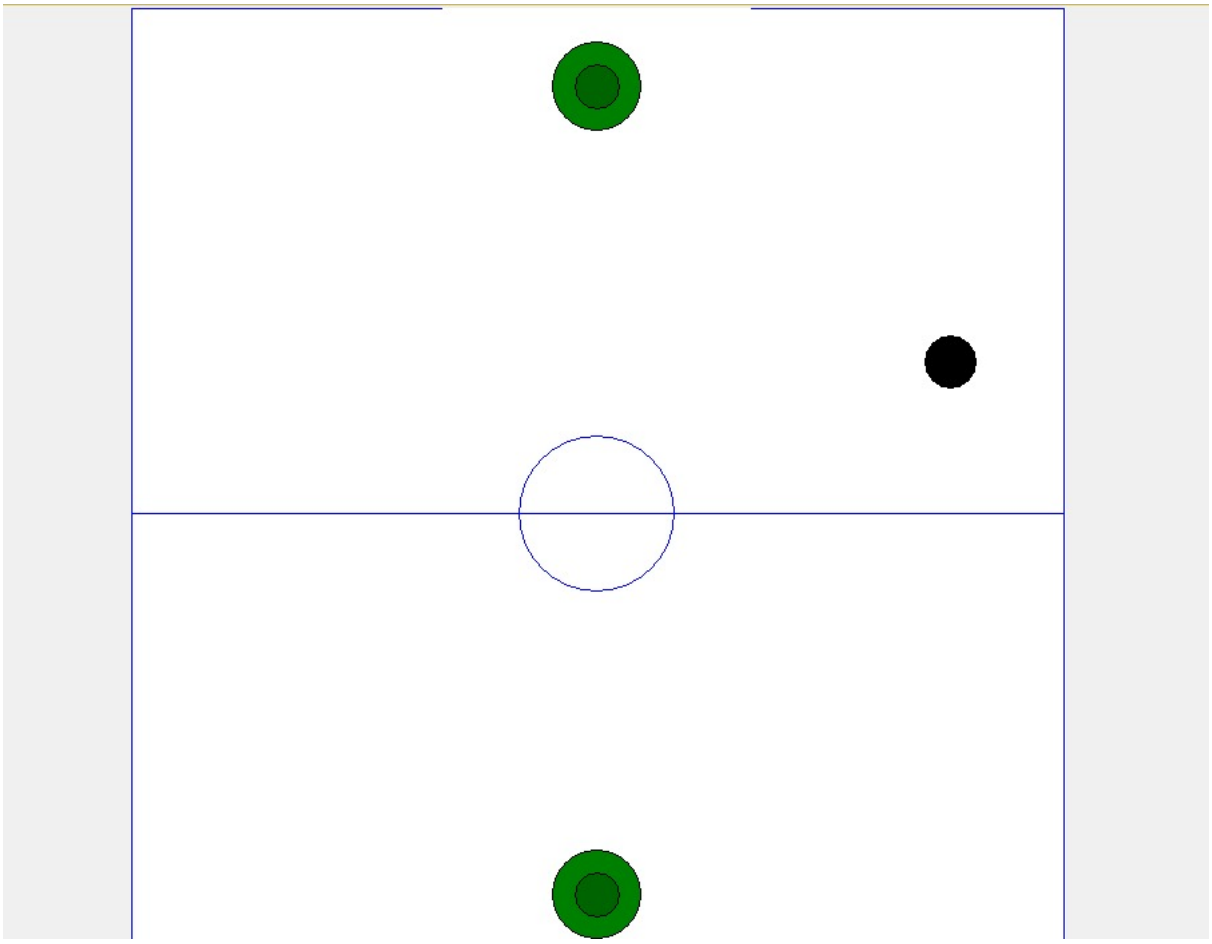

Figure 2: Software Interface for our game

Figure 3: A Sample Joystick (for controlling Paddles)

1.6 Additional Features

Based on the availability of time, we may add the following features:

- Displaying the scores of both the players on the 7-Segment Displays of FPGA.
- Making a more visually appealing GUI for the game.
- Generating sounds when the paddle hits the puck, when the puck hits the corner or when the goal is scored.

2 Project Idea B

2.1 Overview

We will develop a user friendly and interactive game which will entail all the characteristics and elements of a real PACMAN game. So our project will basically be the implementation of PACMAN arcade game but is modified in the sense that we have one ghost operation and other PACMAN-bot so essentially what we would be doing is a PACMAN-type bot that will be moving across the specified maze and eating up all the points and it will also have some bonus extra points and there will be a ghost that will always be trying to catch the PACMAN-bot. The game will be won the by the PACMAN bot when all the points are eaten up or by the ghost if it had caught the PACMAN early enough in the game.

2.2 Input Peripheral(s)

Now, we can see that in order for this Pac-man to work we need two inputs, now first of all this would make this game more interactive in the process like we would use two keyboards for this or we can use one keyboard and one slide switches on the FPGA board by programming it in such a way that it makes mimics the functions of left, right, up and down. So in this way our input (basically the movement of pac-man and the ghost can be taken).
In order for this PACMAN to work we would be using two controllers now we can use two keyboards for this connected to FPGA board or we can improvise and make use of the switches on the FPGA board. In this way, both the players will have their own controls so that they modify the movement of either the pacman or the ghost.

2.3 Output Peripheral(s)

So as PACMAN has the whole arcade box in the olden days we would be taking that idea and implement that in the screen through the use of VGA display. Now in this display the pacman will be moving and so will the ghost. It will also be able to record the scores of the pacman eating the points so in the end we can keep a check and balance of who is winning.

2.4 Block Diagram

Following Figure illustrates the block diagram for this idea:

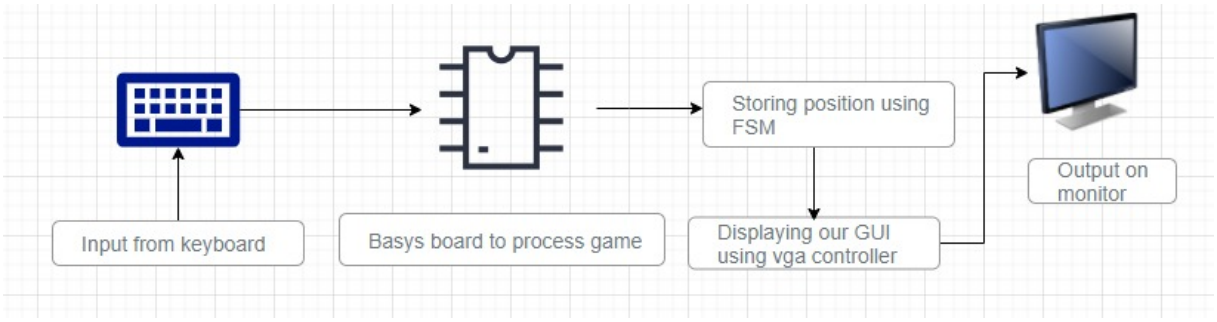

Figure 4: Block Diagram for the game

The details of each block in this diagram are as follows:

2.4.1 Input from Keyboard

We will take input from keyboard through arrow keys which will help the user to navigate. The user can either play as pacman or ghost. We will use a FSM here to store the input in memory.

2.4.2 BASYS board for processing game

This is our basys 3 board which will perform all tasks. It will receive input through keyboard or its built-in switches and generate a transition in our clock signal corresponding to the inputs. As soon as their is a transition the FSM will help the user move. When the pacman has the same position as of ghost game will end.
This will be the brain of our system as this would process most of the important tasks and will execute the operations, as previously stated we will take input from the keyboard through arrow keys which will help the user to navigate. The user can either play as an or ghost. We will use an FSM here to store the input in memory. Now the FSM can work on a positive clock cycle, sending the signal and when it is positive edge and stores the data until the next input appears on the next positive edge cycle. Now our next task will be to move the ghost and the PACMAN, for this

we would take help from the FSMs. The bots moving will have 4 choices (thou not all the time) Left, Right, Up and Down now depending upon our stored inputs the bots will execute the motion until the signal changes that the input has changed thus creating a continuous moving Pacman and ghost that will move continuously until the next input appears and based upon that it will make the decision.

2.4.3 Position Storage Using FSM

Finite State Machine(s) would be used here for multiple purposes regarding game programming logic implementation. For example, one of the FSMs would be used to store the postion of thee points until they are eaten up by Pacman

2.4.4 VGA Controller

The next task that we have is to display the Pacman position and the current score of the Pacman on the screen now this can be done using the sequential circuits involving flip-flops that can store the previous position so that every time our position changes we can see that on the VGA monitor screen.
This block in our design is responsible for the VGA Display. It'll generate the VGA logic for us (not on screen but the logical output). Initially, our Pacman and the ghost will start at their predefined positions and there will be points all over the place. This will be the initial logic that will be generated by our VGA Display. Now, as the players will be moving the Pacman and the ghost, our screen will be synchronized in accordance with the inputs received. We will link our VGA controller with monitor which display the current position of pacman. One of our FSMs here might be used to store the previous position of pacman.

2.4.5 Output on Monitor

This block will contain a screen which supports VGA Display and with the help of our VGA controller block, we will be displaying our implemented logic for the game on the screen.

2.5 Proposed Model & Sample Prototype

The following figure shows the sample model of our design. The keyboards as we all know are all almost similar, thus we will be using any of them in our design. The Pacman GUI that can be seen below is a standard GUI. The Basic structure of our game will be similar to it but some minor changes may be made. These include different type of Maze, theme, appearance of objects, shape of Pacman and the ghost, etc.

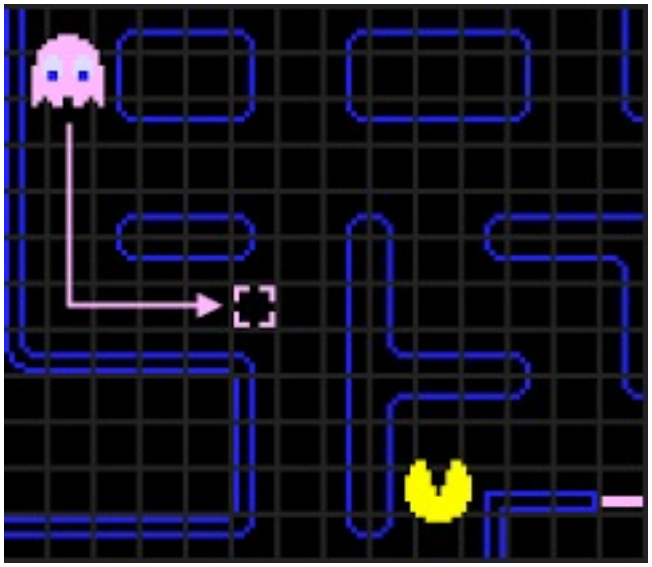

Figure 5: A sample GUI for the proposed game

Figure 6: A sample Keyboard (used by players to control Pacman and the ghost)