SWEEP 'n' PLAY

Student engineer's from Information Department:
Nuzhah JEETUN nuzhah.jeetun@edu.esiee.fr
Marwa BOUFALGHA marwa.boufalgha@edu.esiee.fr

Student engineer's from Embedded System:
Oualid BACHAOU oualid.bachaou@edu.esiee.fr

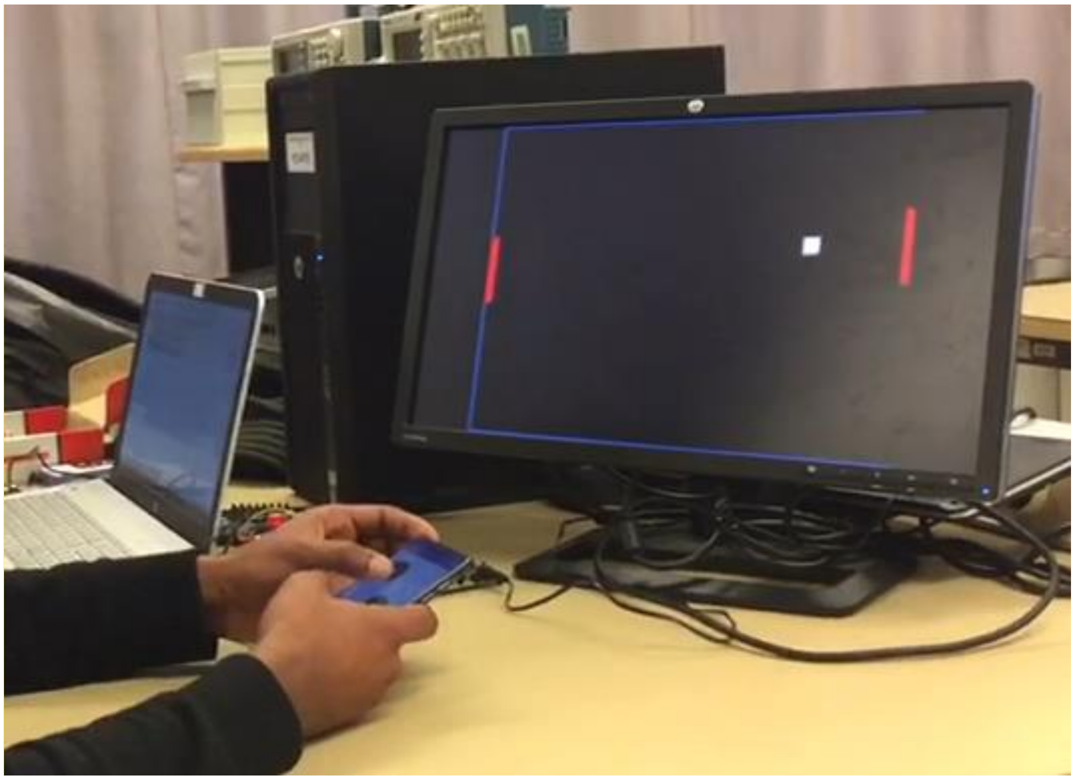
Student engineer's from Electronic Department:
Sylvain BEAU sylvain.beau@edu.esiee.fr
Samuel DUPORT samuel.duport@edu.esiee.fr
Kamel EL MOUSSATI kamel.elmoussati@edu.esiee.fr

May 07th 2018.

Advisor: Anne EXERTIER

ESIEE Paris
Noisy-le-Grand, FRANCE

Submitted for the 2018 Digilent Design Contest Europe



This project was realized as part of the 4th year engineering program at ESIEE Paris by 6 students under the guardianship of Mrs. Anne EXERTIER and Mr. Antoine DHERMIES. They proposed and designed this project to showcase their achievements to the European engineer. The team composed of a different department, blend and develop their teamwork skill and knowledge to develop their project idea. The project name is SWEEP’n’PLAY, a new concept of gaming.

The Team Project

The SWEEP’n’PLAY Marketing Sheet

SWEEP'n'PLAY is a new game concept. The console offers a new way of playing thanks to its smartphone or tablet. It combines the following 5 mains points:

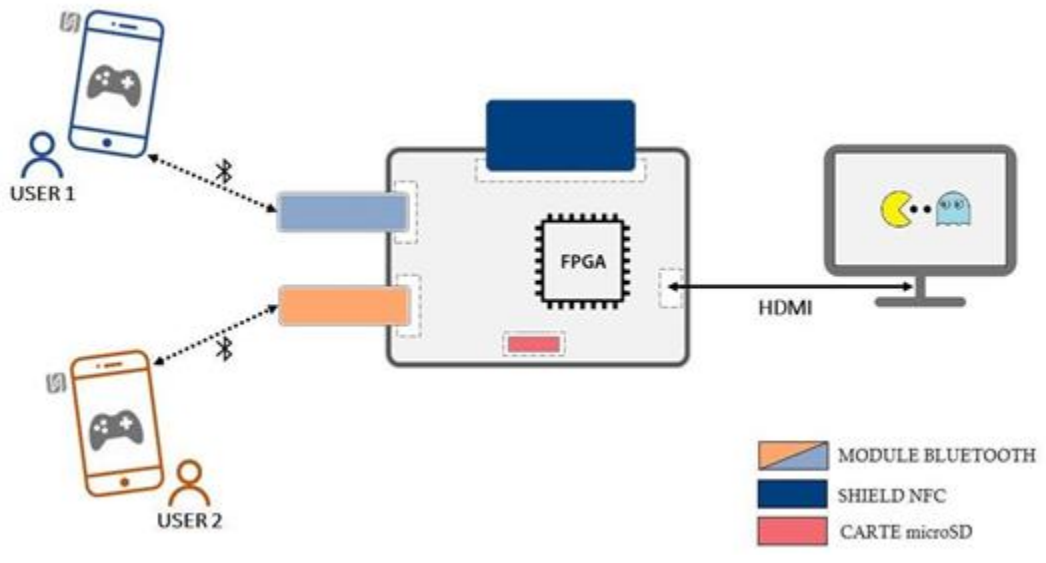
• Requires only a TV and a mobile. Once the console is connected to the television, the user with his smartphone / tablet can start playing.

• The mobile becomes a versatile controller. From the dedicated application, the set of movements made with the mobile terminal (touch event, rotation, ...) will be transcribed in the game.

• Intuitive and fast connection. A simple scan with the mobile over the console is enough to initiate wireless communication between the two terminals.

• A multitude of games. The console has a large library of games.

• Up to 8 players simultaneously. The SWEEP'n'PLAY console supports multiplayer thanks to the technologies it contains.



Introduction

Abstract

The SWEEP'n'PLAY console offers a new way of playing thanks to its smartphone. Once the console is connected to the television, the user mobile becomes a versatile controller; From the dedicated application, the set of movements made will be transcribed in the game. The console has a large library of games and supports up to 7 players.

Objectives

The team is divided into 3 pairs, corresponding to the three sectors (Electronic Systems, Embedded Systems and Information Systems) that the team composed when the project was launched. Together, we have defined the following objectives:

1. For the pair coming from Embedded Systems, the main objectives are:
    - Develop a video game
    - Make a case for the console
    - Manage the image / sound interface of the console

2. For the binomial from the Electronic Systems sector, the main objectives are:
    - Realize the NFC link on FPGA
    - Make the Bluetooth connection on FPGA
    - Manage storage / reading of data

3. For the binomial coming from the Information Systems, the main objectives are:
    - Make the NFC link on a smartphone
    - Realize Bluetooth communication on a smartphone
    - Develop the mobile application
    - Realize the website

Once this list of objectives defined and divided into pairs. We have listed the set of objectives by priority of delays, for more detail the delays of the tasks are listed in the GANTT Chart.

Features-in-Brief

- **Needs Hardware**

Needs	**Solution**
Gaming support	FPGA card
	Smartphone
Transmission control	Bluetooth
ID reader	NFC
Connectors	HDMI (or VGA)
Display	Screen (or overhead projector)

- **Needs Software**

The application we decided to develop is an application allowing a user to play with his smartphone. We will present the work environment as well as the development tools we have used. Then, we describe the operation of the application.

Mobile technologies are gaining more and more place in the market. There are several development environments of an application each with its advantages and disadvantages. Even though it is possible to develop an application using a basic text editor for writing, it is more comfortable to use a development environment, which will make it easier to write, compile, debug the application. As part of the project, we decided to develop an Android application. However, throughout the project, we use Android Studio. Android is based on a Linux 2.6 kernel. Android provides a Software Development Kit (SDK) based on the Java language. It has a library of several basic java classes that will help us create our application.

In order to develop an application, we started by downloading the necessary tools such as the Android Studio platform. It can be downloaded for free from:

https://developer.android.com/studio/index.html

Vivado being the Xilinx product implementation software, it is obvious that we use this one. we used Xilinx Vivado + SDK 2017.4, it can be downloaded from:

https://www.xilinx.com/support/download.html

Project Summary

SWEEP'n'PLAY include several features:

- The mobile application
- The telecommunication peripherals
- The game
- The Graphic Engine
- The Marketing part

Reference Material

➔ For the command transmission, we used an bluetooth PMOD classe II.

The Bluetooth is a technology wireless. The Bluetooth and the Wi-Fi use the same frequency band, 2,4 GHz. The comparison stops here because we don’t use it for the same application in the fact of the Wi-Fi is more performing but use more energy that’s why we have chosen to use Bluetooth technology.
The Bluetooth technology use passing band lower and transmit lower data. There are 3 classes, with the class 1, using 100mW max, you can connect with a system at 100 meters, with the class 2, using 2.5 mW max, and you can send data at 10 meters and for the class 3, using 1 mW max only 1 meter. we have chosen the second class to allow access to public access.

➔ For the connection to the card, we used an NFC shield NXP-OM23221ARD

Near Field Communication, is a technology used for exchanging data under 10cm between 2 systems using this technology. The NFC is on almost all mobile. There are 3 function modes

- Mode emulation card: the mobile function as a contactless card. The SIM of the mobile may use for storing some information and secure it.
- Example: Payment with contactless

- Mode lecture, the equipped of the NFC is able to read tags* in order to get information and launch an application in an automatic way.

o Example: Launching an application close the NFC Shield.

- Mode peer to peer, it's for sending data between 2 mobiles.

o Example: exchanging picture between mobiles.

* A "tag" is an electronic label programmable, using the NFC technology. The interest is to send the same information at the approach of a system.

Security of data with the NFC:
The exchange of the data is secure for 2 reasons:
- The respect of the norm 14443 and FeliCa using algorithm authentication
- The lower distance of communication that extremely decreases the thief of data.

➔ Higt-Definition Multimedia Interface

HDMI is a proprietary audio/video interface for transmitting uncompressed video data and compressed or uncompressed digital audio from an HDMI-compliant source device, such as a display controller, to a compatible computer monitor, video projector, digital television or digital audio device. HDMI is a digital replacement for analog video standards.

HDMI implements the EIA/CEA-816 standards, which define video formats and waveforms, transport of compressed, uncompressed, and LPCM audio, auxiliary data, and implementation of the VESA EDID. CEA-861 signals carried by HDMI are electrically compatible with CEA-861 signals used by the digital visual interface. No signal conversion is necessary, nor is there a loss of video quality when a DVI to HDMI adapter is used. The CEC capability allows HDMI devices to control each other when necessary and allows the user to operate multiple devices with one handheld remote control device.

➔ Arty Z7

the ARTY Z7 allows us enough resources and a processor which can generate a game, it is also equipped with 2 Pmods, a HDMI port and the connector for add the NFC Module.

➔ Smartphone android

➔ HD Screen or projector

Tools Required

- **Software part:**

For the SWEEP'n'PLAY project, we used many tools. For the deployment of the game engine and the graphics engine, we programmed on NotePad++ before proceeding to simulation on Modelsim.

The high HDMI part and the Bluetooth have been done on VIVADO.
The driver is a programme on C which is executed on the processor, the program configures the IP the Bluetooth and transmits the data from the user.

- **Hardware part:**

A Zybo has been used to precede test with an exit by VGA connected to a screen. Before migrating the code on the Arty Z7 with the HDMI successfully.
The low HDMI part has been done in hardware.

Design Status

Concerning the status of the project. We consider that we have done 75% of the project because currently, two applications are ready on Smartphone, Samsung Note 4 and Essential Phone which is able to play with it. The mobiles can connect to their Bluetooth and send data which read from the Game Engine and transmit to the HDMI part before showing it on a screen.

But many things need to be improved or created:

- The NFC Part hasn’t been done and may take a week to programme it.
- The Bluetooth needs to be complete because PMOD Bluetooth is coded as a slave and can’t get many users on its connection and may take a week to improve it.
- The application needs to be improved for the design and complete to make it accessible for all Smartphone.

Background

Why This Project?

Today, many individual hobbies are made for users. Although they are often connected, they highlight the paradox between the connection and the isolation that can set up mobile applications. Technology brings a distance that ultimately impacts society and the very nature of humanity.

That’s why we made this product with an as main goal, create a kind of social net-Game.

That’s how the idea of SWEEP'n'PLAY has been born. A platform that centralizes a multitude of games and a mobile, from the most basic to the most complex. From the most ingenious to the most entertaining. Sweep’n’play can be introduced in those areas for encouraging passenger to play against each other, Always in a competitive atmosphere.

Design: Mobile application

- **Features and specifications**

Concern the application development Android studio gives development kit gives access to examples, documentation, but especially to the system programming API and an emulator to test applications. They have several choices to test an application, either by the emulator that Android offer or otherwise on a real phone. it just uses the appropriate class to perform a function.

- **Graphical interface**

Android uses XML resource files: "Activity_main.xml", to generate a graphical interface. This file is called by the main file "Activity_main.java" by command "setContentView (R.layout.main)". "R" is a resource file of the application containing all the widgets identifiers. It is generated automatically by the SDK.

- **Bluetooth communication**

To play, the user must establish a Bluetooth connection between smartphone and the FPGA card.

The *BluetoothServerSocket* and *BluetoothSocket* classes are used.

- The *BluetoothServerSocket* class is used to establish listening support for initializing a link between a device. To restore recognition, one of the devices acts as a server that listens and accepts incoming requests.
- The *BluetoothSocket* class, it is used to create a new socket client on a server once the connection is established.

Sockets are used by the server and the client to transmit data streams.

- A server socket is used for incoming connection requests that originate from Bluetooth devices. To listen for connection requests, we call the *ListenUsingRfcommWithServiceRecord* method of the Bluetooth Adapter. It is passed a character string "name" to identify our server and UUID. We recover an instance of *BluetoothServerSocket*. To start listening to connections, you must call the *accept* method on the server socket. The server will then block until a client with a matching UUID tries to connect. If an incoming connection request succeeds, accept will return a socket connected to the client. It is used to make data transfers.

- The *BluetoothSocket* class is used on the client to initiate a communication channel from the application to a listening server socket. Create socket clients by calling *createRfcommSocketToServiceRecord* on a *BluetoothDevice* object. This object represents the target remote server. it must have a server socket listening for connection requests

On using an OutputStream to send a string to a remote device.

Design: Peripheral communication

- **Features and Specifications**

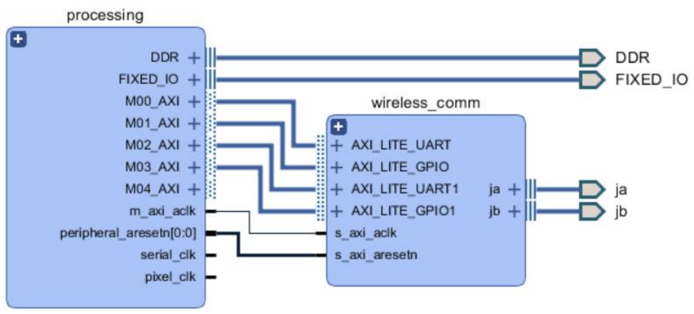
Exchange pieces of information between two devices refer to data communication. Modern world scenario is ever changing; Wireless exchange take a wide part in communication channels. The project refers to wireless data communication in order to transmit players information and commands to the SWEEP ‘n’ PLAY console thanks to the smartphone technologies. Bluetooth and NFC technologies state this point of view :

- Performance (range, broadband and power consumption suitable)
- Scalability (more and more present on a smartphone)
- Reliability (high data encryption)

Only Bluetooth implementation will be described below.

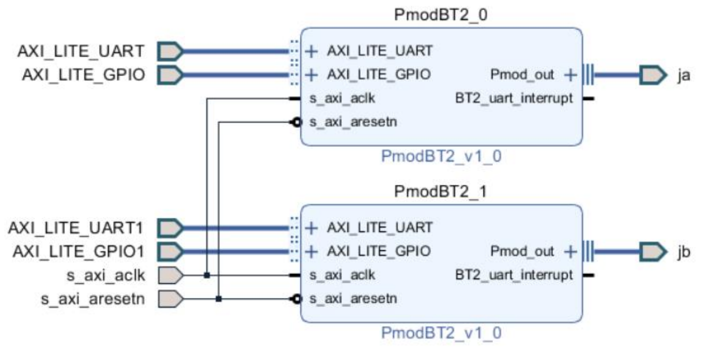
- **Design Overview**

The design contains two parts: A processing part, mainly software, manage the order of transmission and its parameters. The second one, mainly hardware, deals with the lowest layer information transmission.
The processing part contains the processor with the AXI MM bus manager. This bus allows communicating properly data with the logical part. The latter contains Pmod BT2 v1 from Digilent Intellectual Property repository.



- **Detailed Design Description**

This Pmod communicates with the SoC FPGA thanks to UART protocol. The Intellectual Property Pmod BT2 contains, therefore, a UART controller for transmitting information by transmission (Tx) and reception (Rx) pinout. An input/output manager also is integrated into this IP in order to deal with the other pins of the Pmod that allow reconfiguration.
Two players game commands can be handled simultaneously with two IP and therefore two Pmod BT2.



The software part, configure this two IP with the correct baud rate (115200 baud). Once the Bluetooth established, the Pmod transmit all the user commands. The software programme loop capture this commands. The function of the driver BT2_RecvData is placed in an infinite loop (Appendix A)

Design: Mobile application

- **Features and Specifications**

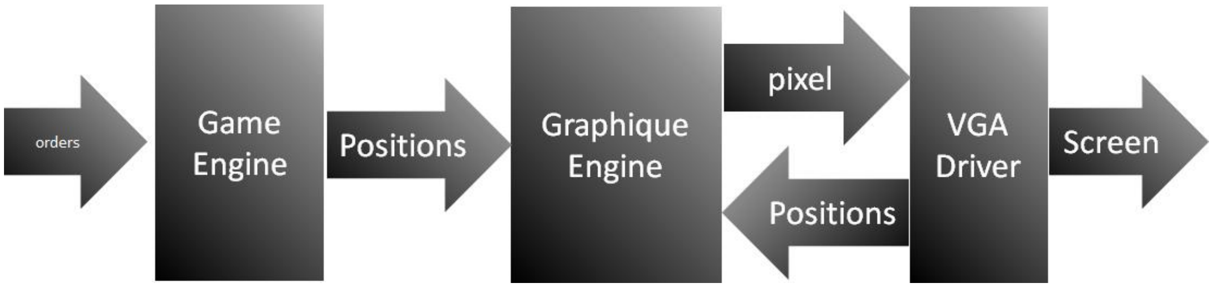
Regarding the game, we set ourselves the objective of making a game easy to program, which will be entirely realized in hardware in order to be in agreement with the specifications and the contest.

Ping-pong, being the first historical game, easy to implement and easily improved, we decided to make this game. Ping-Pong has the calculation of the position of 3 elements via a Cartesian reference in two dimensions. Thus, this game can be easily improved, enriched to make it more attractive. As for example to add a second bullet, accelerate the refresh period of position calculation of the elements.

The second reason we chose this game is that it's a multiplayer game. One of the objectives of the SWEEP'n'PLAY project is to set up a platform to physically connect users, allowing them to occupy their time in a friendly moment with other users who are in the same situation.

- **Design Overview**

The main architecture in this game is the same for any other games that we would like to develop later. It is divided into three distinct parts:
• The first named Game_Engine will detail the algorithm to make the game.
• The second named Graphique_Engine will detail the transcription of the elements of the game.
• The third named Driver will detail the global display on a screen.
As described in the following diagram, these 3 parts allow the interaction between the commands already processed (see chapter communication) is the screen visible to all users.



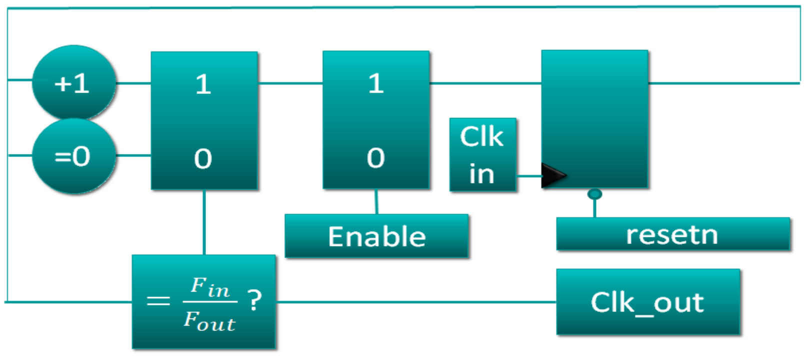
- **Detailed Design Description for Game_engine**

As described in the previous diagram, Game Engine has input user commands and outputs the position of the elements of the game.
The Pong is made of 5 parts:

- A pulse generator that allows pacing the game
- User-provided data processing
- A treatment of the position of the ball
- A treatment for moving the ball
- A treatment for the score counter

- <u>A pulse generator</u>

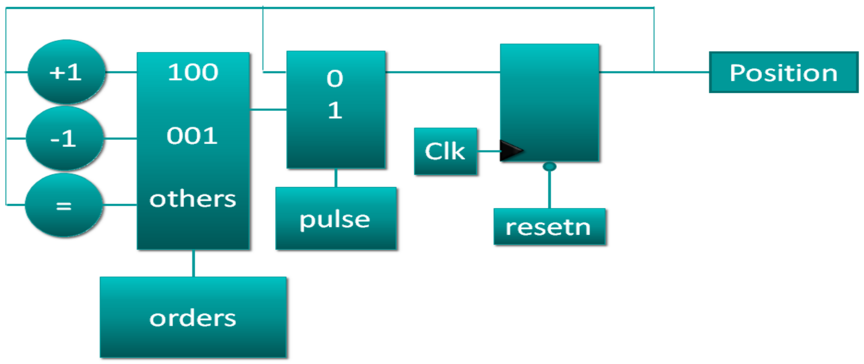
The pulse generator allows counter incrementation at each clock stroke according to an enable signal, the game we have not set up a pause, it is for this reason that the enable signal is blocked at 1 if you would add one.
Once the maximum value has been obtained, the value of clk_out change with a reset of our counter.



- <u>Order processing</u>

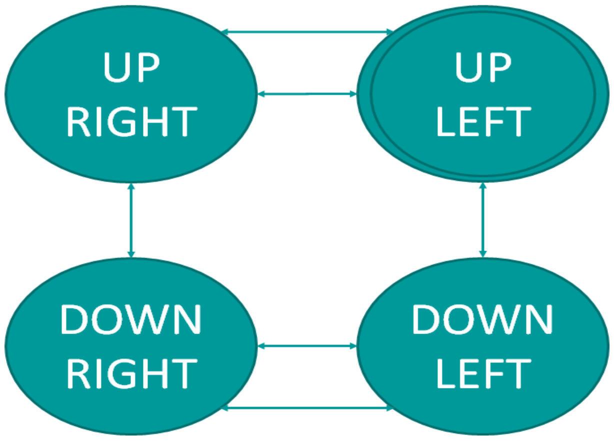
The processing of the user's commands is performed by a method of incrementation and decrementation according to a binary code on 3bits input.

This signal is updated via the enable speed input which is a signal given by the pulse generator. The position is an N-bit signal with N being the upper rounding of the Log in base 2 of the vertical resolution of the screen. This makes it possible to have a position on a number of bits equal to our screen resolution.



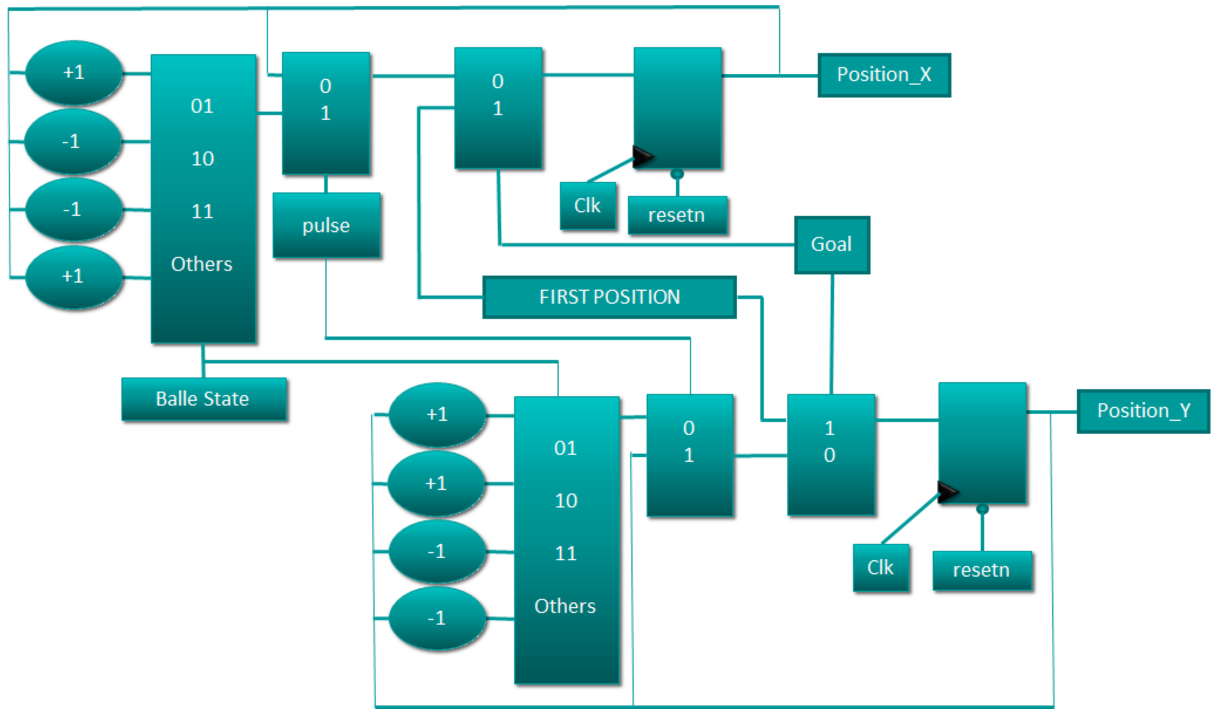
- <u>FSM</u>

direction to follow via a binary code on 2 bits.
Inputs are the ball positions and racket position, which we use to perform the treatment of rebounds via comparators
For rebounds against rackets the position of the ball is compared with 3 values:
• The position of the ball must equal the position of the racket (X-axis)
• The position of the ball must be greater than or equal to the position of the racket minus the half width of the racket (Y-axis)
• The position of the ball must be less than or equal to the position of the ball plus half the width of the racket (Y-axis).
For rebounds against the extremity of the screen, it is sufficient to compare the position of the ball X / Y with the minimum and maximum values of the screen of the resolution defined.



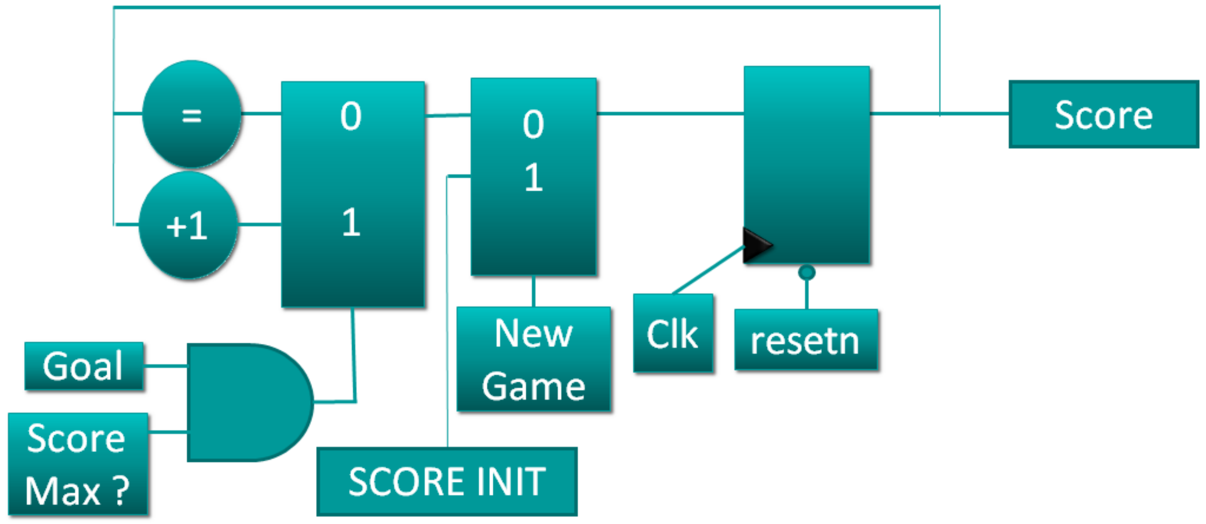
- <u>Treatment of the position of the ball</u>

The treatment for the ball position is controlled by the state machine described above. The user inputs are two signals.
First entry the state of the ball allows us to increment or decrement the X and Y positions.
The second input resets these same X and Y positions in case of a goal.
In output, we have two signals.
• The first calculates the X-axis position with an N-bit 'std_logic_vector' where N is the upper round of the Log in base 2 of the horizontal resolution of the screen.
• The second calculates the Y-axis position with a M-bit 'std_logic_vector' with M being the upper round of the Log in base 2 of the vertical screen resolution.
These signals are updated via the enable speed input which is a signal given by the pulse generator.

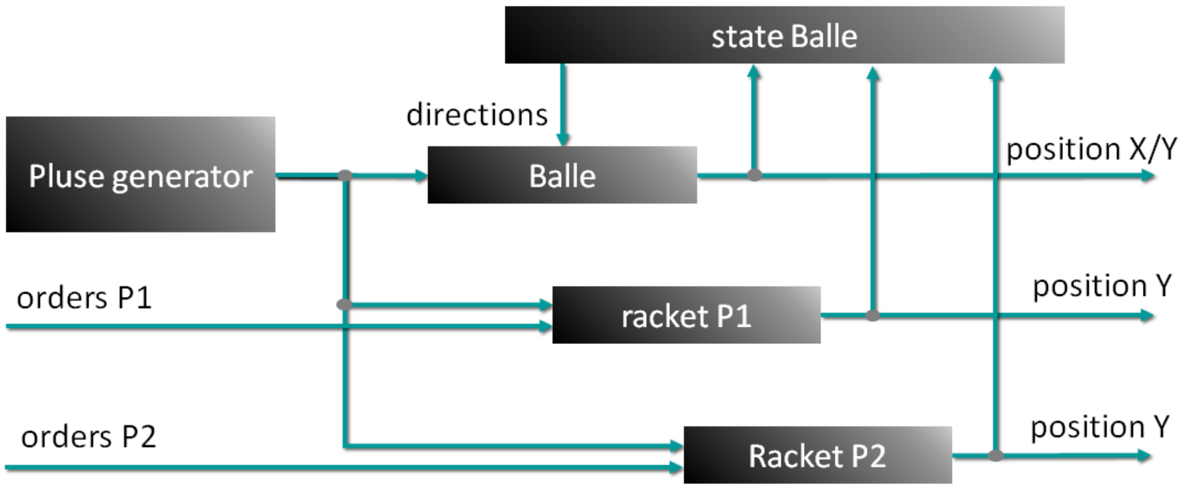


- <u>Treatment for the score counter</u>

Finally, the score is processed by a simple counter up to a max value, the only way to reset the score is to launch a 'NewGame' which is a binary code sent by the user.



with a more general view the game architecture seems to like:



- **Detailed Design Description for Graphique_engine**

they are six inputs, four come from the game and the other tow from the Display module. The four signals corresponding to the location of the ball and the X and Y sliders to which we add a value corresponds to the width and length of the ball and racket. And coming on the Display module the position in X and Y of the pixel to control.

the Display module which can be VGA or HDMI must contain its own process that increments the pixel to be controlled. The signals are incremented at each rising edge of the clock and the game sends signals that change in real time.

The main role of the graphics engine is to make a comparison between the signals from the game and those of the Display module, before sending a signal output. If the signal values are true then the graphics engine will produce an output signal corresponding to the colour of the pixel to be changed.

Discussion

Problems Encountered

Bluetooth

The Bluetooth can connect at his own service 7 different devices thanks to Piconet. In that way, only one Pmod Bluetooth has been bought. The idea was to use our two mobiles for playing.

devices thanks to Piconet. In that way, only one Pmod Bluetooth has been bought. The idea was to use our two mobiles for playing.

In the beginning, we wanted to place the device as a Slave and the Pmod Bluetooth as a Master. But we were stopped when we precede test on the communication. The error was the device was able to send data to the Arty board, but the card didn’t give any data to the device which was bad because we weren’t able to command the FPGA.

In order to solve this problem, we placed the card as Slave and the device as a Master and tried to send data. This solution was successful but by putting the PMOD Bluetooth as a Slave, we can’t connect many devices on it. So, connecting another PMOD Bluetooth was the only solution possible.

Concerning this part of the project, we should correct this default, which is not important currently.

NFC

Currently, The NFC isn’t ready yet because we didn’t have enough time to do it. It was an important phase of the project because by approaching the device on the board, the shield NFC launches a command to the Bluetooth with the information of the device and proceed to the connection. With more time we may finish it.

Video transmits

Coding a video HDMI isn’t easy, more if the sound is coded too. It takes a long time to program it. The game was ready to test but the HDMI wasn’t. To encounter this problem, we implemented a VGA PMOD for proceeding to the test.

The HDMI has finally programmed, and it is successful.

Application
The application isn’t downloadable yet, few mistakes should be correct because it isn’t downloadable all devices.

Engineering Resources Used

ESIEE Paris, engineering school based at Champs-Sur-Marne, is a school composed at 40% seeker teacher. Teacher a fine human resource when we couldn’t find a solution.
In that way, to get some any information about the development of the application we asked Mr.Hammouch REDAH how to solve the problem that we met by doing the Bluetooth.
Mme. Anne EXERTIER and Mr Antoine DHERMIES were present when we had any request about FPGA or when we need component like the PMOD VGA.

Actually, we worked hard in the laboratory of school, each Monday afternoon, on a project for 4 months. The rest of the time we did a lot of test concerning the communication which was not an easy part of the project.

Marketability

The SWEEP’n’PLAY console can be placed in every public place would allow people to stay together connected virtually but also stay gathered in these places. People would be able to use the product to play with other people next to them.

In many situations, potential users are forced to wait for long periods such as at railway stations, airports, cinema and others. Whether it is in a cinema, an Airport, Train station. Everyone has certainly lived, and there is nothing enjoyable. It’s this waiting time that we are obligated to lose with our phone.
The problem that we want to solve is to delete the notion of longtime waste by waiting for the departure, by integrating the FPGA card, known for its speed of execution, with several simple games.

We want to create a cool space where people will be able to play against the other one in order to win point corresponding to his score. With that point, people will be able to personalize his avatar Currently, we did the Pong historic game, but we look forward to implementing several games which will possible to play incorporation with any other player.
Our target is to develop a gaming platform of several vintage games for creating a community of player

Lastly, since the application, the information like score, participant or location can be easily shared online on different social networks.
This product will allow connecting physically and virtually several users. since it's necessary to be at least two to play.

Appendix A: {main_app.c}

```
#include "xil_cache.h"
#include "xparameters.h"
#include "PmodBT2.h"
#include "xuartps.h"
#include "my_game.h"


#define UART_DEVICE_ID XPAR_PS7_UART_0_DEVICE_ID

PmodBT2 myDevice_0,myDevice_1;
XUartPs myUart;

int main(void){
  XUartPs_Config *myUartCfgPtr;
  u8 buf0[1],buf1[1];
  int n,m;

  Xil_ICacheEnable();
  Xil_DCacheEnable();

  myUartCfgPtr = XUartPs_LookupConfig(UART_DEVICE_ID);
  XUartPs_CfgInitialize(&myUart,myUartCfgPtr,myUartCfgPtr->BaseAddress);


         BT2_Begin (
    &myDevice_0,
           XPAR_WIRELESS_COMM_PMODBT2_0_AXI_LITE_UART_BASEADDR,
           XPAR_WIRELESS_COMM_PMODBT2_0_AXI_LITE_GPIO_BASEADDR,
           XPAR_PS7_UART_0_UART_CLK_FREQ_HZ,
    115200
         );
```

```
            BT2_Begin (
        &myDevice_1,
                XPAR_WIRELESS_COMM_PMODBT2_1_AXI_LITE_UART_BASEADDR,
                XPAR_WIRELESS_COMM_PMODBT2_1_AXI_LITE_GPIO_BASEADDR,
                XPAR_PS7_UART_0_UART_CLK_FREQ_HZ,
        115200
    );






    while(1) {
        n = BT2_RecvData(&myDevice_0, buf0, 1);
        if (n != 0) {
                XUartPs_Send(&myUart, buf0, 1);
        }
        m = BT2_RecvData(&myDevice_1, buf1, 1);
       if (m != 0) {
                XUartPs_Send(&myUart, buf1, 1);

        }


    MY_GAME_mWriteReg(XPAR_GAME_MY_GAME_0_S00_AXI_BASEADDR,0,(buf0[0]%48)+(buf1[0]%48)*8);
    }

    Xil_DCacheDisable();
    Xil_ICacheDisable();

     return XST_SUCCESS;
}
```