ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

PYRAMIC ARRAY

Ultimate Guide

Juan Azcarreta Ortiz
Robin Scheibler

August 29, 2016

1 INTRODUCTION

The Pyramic array is a 48 microphone channels modular system (Figure 1.1), which prominent applications are related with beamforming, acoustic source localization and room shape recognition among others. The system is composed of six arrays of eight microphone channels each. Pyramic array interfaces with an FPGA system from Altera's Cyclone V family [3]. Those devices feature a Hard Processor System (HPS) embedded with the FPGA part in the same chip [2]. Therefore, the FPGA design allows to develop hardware programming for real-time applications, while the HPS presents a more traditional way to implement software applications. This guide sets a ground to start working with the Pyramic array.

The main features of the system are:

- A modular system with 6 linear arrays of 8 microphones each, giving a total of 48 microphones distributed in the edges of a tetrahedron. Each array samples synchronously 8 audio channels at 48 kHz and 16 bits resolution.
- A parallel SPI communication between the Pyramic array and a GPIO port connected to the pins of the FPGA part of an Altera's Cyclone V family.
- An interface between the FPGA and a ARM-9 processor located in the Hard Processor System (HPS). The system runs over a Linux based operating system. The data is stored in the DDR3 memory of DE1-SoC device.
- An Ethernet communication between the Altera's DE1-SoC device and the Pyramic array, which allows internet sharing and control of the system through a WebServer.
- The samples are stored in the DDR3 memory available in the board, in a microSD plugged in the board and is sent through *scp* to a host computer.

In our implementation Altera's DE1-SoC board from Cyclone V family has been chosen. For more information regarding the Pyramic array design and implementation refer to [1].

2 PROJECT STRUCTURE

We introduce the project structure followed In the implementation of the PyraMic array design. In our design the project name is MIC_ARRAY, inside where we can find the following tree of folders shown in Figure 2.1.

The side of hw corresponds to all the files compiled through Quartus and Qsys, i.e. , every hardware design (even in VHDL or Verilog) is located in this folder. Sw corresponds to all the software part of an embedded system. From compiling, downloading an operating system to run an ARM Cortex-0 Processor. The sdcard directory contains all final targets needed to create a valid sdcard from which the DE1-SoC can boot. Let us have a closer look into each folder branch:

- hw: it contains all the hardware-related file of the project. Those are the main folders you should find inside:

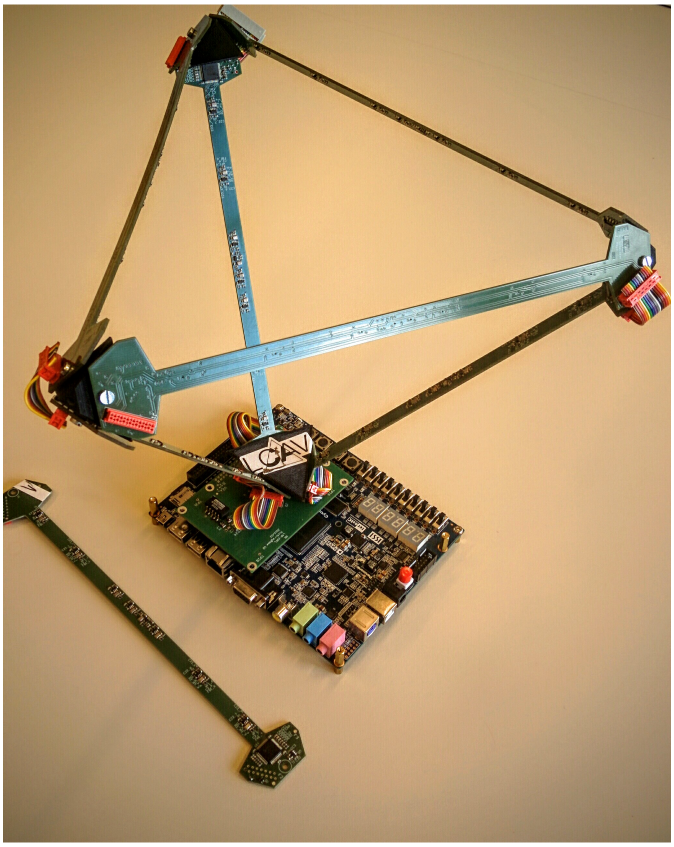

Figure 1.1: Pyramic array

  - Modelsim: Contains all the test-bench and simulation files developed through the project. In our case the file name is called ‘
  - Quartus: Contains all the files related to the project and qsys system, both the created by the user and the generated ones.
  - Hdl: Where all the design files are stored, from the top level entity to a particular component designed in VHDL.
- sw: it contains all the software-related files for the project. Those are the main folders you should find inside:
  - hps: With all the necessary files to launch the preloader in the HPS and capable or running bare-metal or Linux OS applications.
  - Nios2: Not used in this project. All the applications running Nios2 soft-processor should be in this folder.
- Sdcard: Where all the files that should be saved to the SD card are stored.

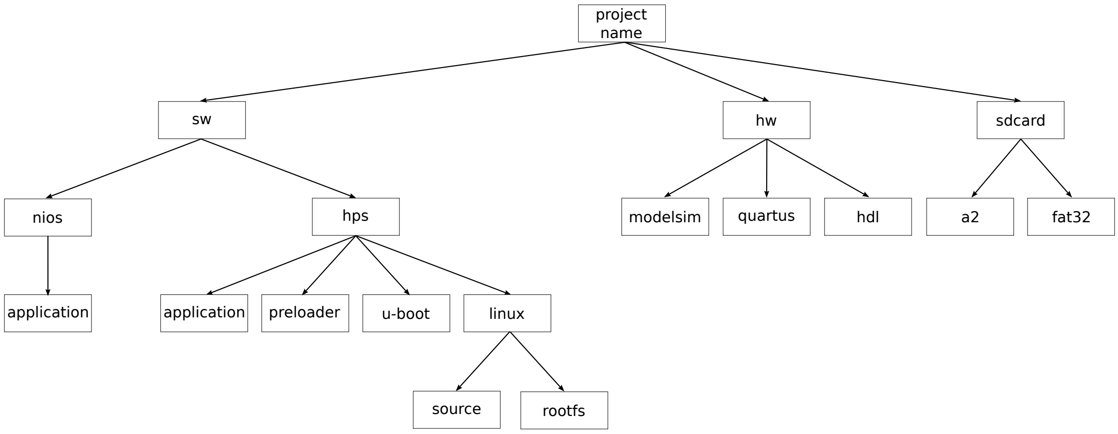

Figure 2.1: Project Structure

3 START THE SYSTEM

First, connect the Altera DE1-SoC board as displayed in Figure 3.1:

- Ethernet connection
- UART connection
- Power
- Pyramic array into the GPIO 0 pins (red rectangle)
- Insert MicroSD card

After connecting everything, power the board pressing the red button. The device will be configured thanks to the device tree files stored in a *fat32* partition of 512 MB in the microSD card. The SD card has a file system *a2* partition of 24 GB where all the directories of the Linux OS system are defined.

- The FPGA and the HPS portions turn on.
- The HPS boots from the device tree *.dtb* stored in the MicroSD.
- The FPGA is configured through the HPS part loading a binary *.rbf* file located in the HPS.
- The MPU subsystem boots from flash devices connected to the HPS pins.

3.1 Set internet sharing

Connect the UART and Ethernet cables to your computer. Open a terminal and start a serial communication (through *Putty* in Windows or *minicom* in Linux) with the Pyramic array. First, we want to share internet with the DE1-SoC device. For those purposes, we need to set a **static IP address** in the DE1-SoC device. In **your computer**, open a terminal and type:

```
$ ifconfig
```

Check the IP address of your laptop for *eth0* connection. That would be the *gateway* address. Now, go to the DE1-SoC *minicom* window opened before and open the following file:

```
$ nano /var/network/interfaces
```

As an example, imagine we found an IP address of *10.42.0.1*. In that case, you should enter the following parameters at the end of the file you just opened. Change

```
iface eth0 inet dhcp
```

by

```
iface eth0 inet static
address 10.42.0.2
netmask 255.255.255.0
gateway 10.42.0.1
```

Note the *address* could be 10.42.0.3 as well, the only conditions is that gateway = addess AND gateway (i.e., just last number can change).

Now, in the board open the following file:

```
$ nano /var/resolv.conf
```

and set the nameserver as follows:

```
$ nameserver 10.42.0.1
```

We have configured *eth0* connection to access the internet through our computer. After this, we need to create a Shared Connection. In Ubuntu go to Network Connections, click Add and chose Ethernet. Edit the network name and in the IPv4 panel change the method to *Shared to other Computers*. Create the network. Then, everytime you connect to this new shared network DE1-SoC board will have access to the internet through Ethernet Connection.

3.2 Running an application

3.2.1 Running from Linux

Open a *ssh* connection by typing the following in your computer:

```
$ ssh root@10.42.0.1
```

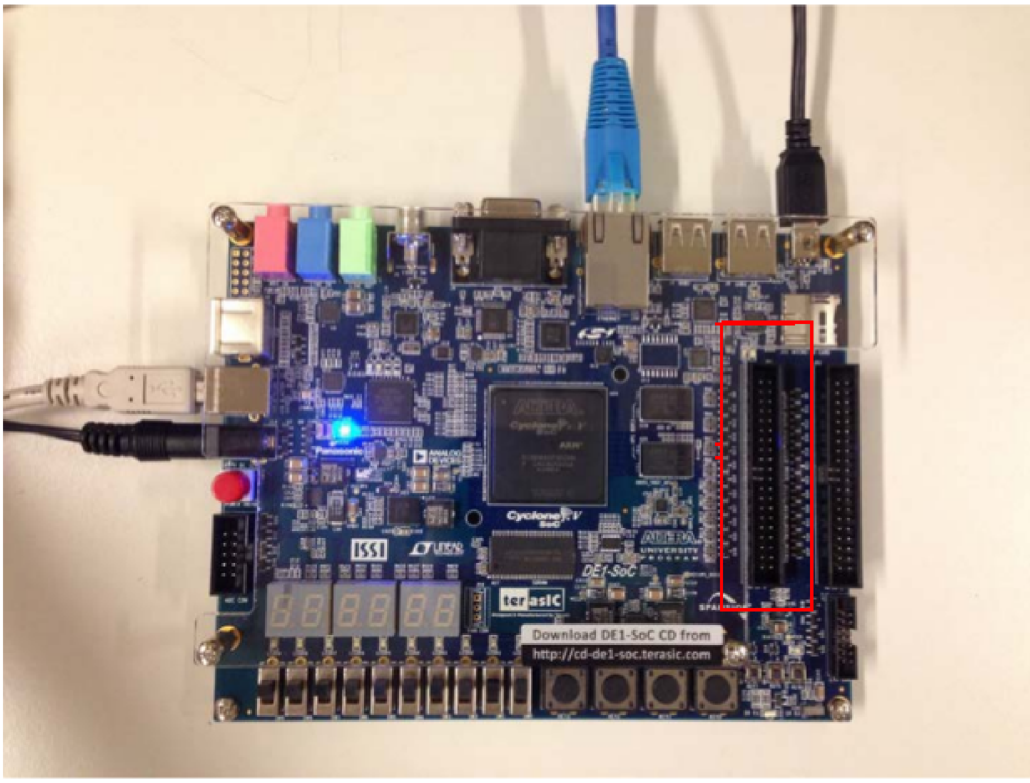

Figure 3.1: Pyramic array overview.

Type the password 1234 and you should be connected to the device through Ethernet. Now, go to the folder where the binary is saved in the board:

```
$ cd /home/lcav
$ Mic_Array_HPS 3 audio_folder
```

where Mic_Array_HPS is the current name of the project, 3 corresponds to the recording duration in seconds and audio_folder the directory where the data will be stored in your computer.

3.2.2 Running from a WebServer

Once the Internet is Shared with the device just type the address of your board in the search bar. As an alternative, you can type the following command in a terminal in your laptop:

```
$ firefox 10.42.0.2
```

Nowadays, the WebServer has input text boxes where the duration (in seconds) and the storage folder can be chosen. When the acquisition is finished the target folder and a flag message will appear on the screen.

Pyramic array uses *APACHE 2* web server, which implements *.php* scripts. To change the current *.php* script go to the following file:

```
$ nano /var/www/html/index.php
```

and you can develop your own web application.

3.2.3 Running from Eclipse DS-5

In your laptop open Eclipse typing the following commands:

```
$ embedded_command_shell
$ eclipse &
```

It is preferred the Working directory not to interference with the directory where your project is. To check the licenses are correct and available go to Tools -> License Manager.

If you need to import a project go to File -> Import -> General -> Existing Project into Workspace. In Select root directory chose the desired project to open (Mic_Array_HPS in our case).

To run the project and download the binary to the DE1-SoC board right click on the Mic_Array_HPS folder in the right panel and chose Debug as -> Debug Configurations.

Follow Section 13.8.2.3 in [4] to create a new debug configuration. Just particularize the application on host to download and the Target working directory to /home/lcav . Check the board has internet access and press Debug to load your design.

4 HARDWARE DEVELOPMENT

If you wish to modify anything from the FPGA desing you will need to open the Quartus project, adapt the project to your needs and recompile it again. If you do so, please, back up a working copy before making any change. After performing the changes and successfully compiling the project, you need to:

- Load the new *.rbf* binary file in the microSD card *fat32* partition
    1. Create the *.rbf* file from the *.sopc*

```
$ quartus_cpf -c /hw/quartus/output_files/
  Pyramic_array.sof sdcard/fat32/socfpga.rbf
```

2. Plug the microSD card in your computer and copy the *sdcard/fat32/socfpga.rbf* file in the *fat32* partition of your SDcard.

- Create the new header file for the Eclipse project

1. Create the header file by typing:

```
$ sopcinfo2swinfo --input=hw/quartus/Pyramic_array.
  sopcinfo
```

```
$ swinfo2header --input=hw/quartus/Pyramic_array.
  swinfo
```

2. After the header files are created, copy the *Pyramic_array.h* and paste it into the *Mic_Array_soc.h* in Eclipse.

Note that if you perform any change into the HPS part (such as adding a new HPS peripheral controller or changing a bus width) you need to create the **.dtb** again. For those purposes you should need to run the *u-boot* script again as in Section 13.2.2 of [4].

REFERENCES

[1] Juan Azcarreta. Pyramic array: An fpga based platform for multi-channel audio acquisition. 2016.

[2] Altera Corporation. *" Cyclone V Device Handbook, Volume 3: Hard Processor System Technical Referene Manual," 31 July 2014*, 2014.

[3] Altera Corporation. *"Cyclone V Device Overview"*, 2016.

[4] René Beuchat Sahand Kashani-Akhavan. *SoC-FPGA Design Guide*.