Xilinx Open Hardware 2023 Report: text to speech

By Paolo Russo, Team: xohw23-170

1 Introduction

Advancements in open hardware and software have revolutionized various fields, enabling individuals and communities to explore innovative solutions collaboratively. This report presents an open hardware project developed on the Zybo ZYNQ-7020 development board, utilizing its record and play function in conjunction with MATLAB. The objective of this project is to generate audio sentences using prerecorded Italian phonemes, which have been modified and manipulated using MATLAB.

The Zybo ZYNQ-7020 development board, with its powerful processing capabilities and flexible programmability, provides an ideal platform for prototyping and implementing diverse applications. By combining this hardware resource with MATLAB, a popular programming environment that can be used also for audio processing, we aim to create a system that can generate synthetic Italian audio sentences based on user-defined text inputs.

The project leverages the concept of phonemes, which are the smallest distinct units of sound in a language. In the Italian language, phonemes form the foundation of pronunciation and articulation. By pre-recording and modifying Italian phonemes, we can assemble them dynamically to generate natural-sounding sentences, offering a versatile tool for various applications such as language learning, speech synthesis, and linguistic research.

MATLAB, a widely-used software environment, provides a rich set of tools and functions for audio signal processing. By utilizing MATLAB's signal processing capabilities, we can manipulate and modify the prerecorded phonemes to match the desired text inputs. This allows us to generate coherent and contextually relevant audio sentences in Italian, tailored to specific requirements.

Throughout this report, we will delve into the technical details of the open hardware project, discussing the Zybo ZYNQ-7020 development board's features, the recording and playback capabilities it offers, and the integration with MATLAB. We will explore the implementation steps, including the preprocessing of prerecorded phonemes, the generation of sentences based on user-defined text inputs, and the playback of the synthesized audio. We envision this project to serve as a foundation for researchers, developers, and enthusiasts to build upon and explore new frontiers in audio synthesis and language processing in Italian, a complex and particular language.

In the digital age, text-to-speech systems have become increasingly relevant due to their potential in various fields such as telecommunications, education, and accessibility. While sophisticated text-to-speech systems have seen significant advancements, the importance of simple, resource-friendly systems cannot be overlooked. These systems can be invaluable in contexts where resources are limited or the primary requirement is for basic sentence construction and spelling.

In response to this need, our project embarked on the development of a rudimentary text-to-speech system using the Zybo ZYNQ-7020 development board, an open hardware platform. We chose to focus on the Italian language, given its rich phonetic diversity. This choice provided an engaging range of sounds for our explorations and offered us a unique opportunity to delve into the complexities of manipulating phonemes within the confines of a simplified text-to-speech system.

A key goal of this project was the creation of a compact dataset for the system. Striving for simplicity and efficiency, we prioritized limiting the size of the dataset. To achieve this, we utilized pre-recorded Italian phonemes, adjusted and manipulated using MATLAB. As a result of our focus on a smaller, more manageable dataset, the system currently delivers a lower quality of output. Despite this, it remains a functional tool for applications requiring only basic speech capabilities.

While this project is still in its early stages and the current system's output is limited in quality, it provides a foundation that can be built upon. Further modifications, including the addition of filters and post-processing steps, are planned to enhance the output quality. The long-term goal is to refine this rudimentary system, improving its functionality and performance while maintaining the compactness of the dataset. Despite its simplicity, this project highlights the potential of open hardware and software platforms as tools for addressing real-world needs.

2 Data: Italian Phonemes

Firstly, we should address the creation and manipulation of Italian phonemes utilized in our project. To maintain simplicity, phonemes were recorded using built-in functions on our PCs. Rather than capturing the phonemes themselves, we focused on the pronunciation of individual letters in Italian. It is essential to note that certain vowels such as "e" and "o", along with some consonants like "c", "g", "s", and "z", and complex sounds such as "gl", "gn", and "sc", have multiple pronunciations in Italian. To ensure our work remains accessible to those with basic knowledge of Italian speech, we chose to avoid the introduction of phonology notation. Once recorded, these sounds were imported into MATLAB and recognized as a double vector with a known sampling frequency (fs) of 48kHz. Additionally, a generic beep sound was incorporated to test the Zybo board after all the data had been manipulated. The first step in manipulation involved normalizing each data piece using the MATLAB function "normalize". Subsequently, we removed silent parts before and after each phoneme to facilitate their combination into words. We used a for-loop to apply a threshold to trim each data piece appropriately, verifying the accuracy of our cut by plotting the resulting data. Before further data modification, we reduced the sampling frequency to make the sounds less storage-intensive without sacrificing quality. A MATLAB script was employed to derive a factor for reducing the fs from 48kHz to 12kHz. This was achieved using a for-loop with the "rem" function, which calculates the remainder of a division. Note that both 48kHz and 12kHz are supported by the audio microcontroller within the Zybo ZYNQ-7020. The next challenge was to make the data readable by the Zybo board. According to the datasheet, sound data is stored in a 24-bit fixed-point notation with 1 bit for the sign and 23 bits for the fractional part. To achieve this, each piece of data was converted to this notation using a MATLAB script. This script utilized the MATLAB function "round", which was provided with data multiplied by 2^{23} as input. The final step involved the generation of a C language header file containing every sound as a constant int32_t vector. This header file was named "soundsdata.h". To create this, we authored a simple script that wrote the syntax of these vectors (including the necessary square brackets and curly brackets) and the respective data numbers into a .txt file. Subsequently, these individual .txt files were merged into a single header file, readying us for the next stage of the project.

3 Zybo ZYNQ-7020 and Implementation

Among all the possible function to implement with the Zybo ZYNQ-7020 we chose the Record&Play function. For our purpose only two files are necessary: *audio.c* and *demo.c*. Here we are not counting the header files of these. The first one, *audio.c* is a set of function that programs the audio microcontroller and allows us to change the frequency among a list of possible ones only varying some bits. In our case the combination of bits necessary to obtain a frequency of 12kHz is "001". In *demo.c* we have the main function that gives us the graphical interface and tells which button on the board we have to press in order to active some functions. We have modified this by deactivating the record functions and activating only the playback one by inserting with our file *soundsdata.h* the set of phonemes. With some macros at the beginning of the main function we have defined the audio sampling rate by setting to 12000. There is an other macro that allows to extend the duration of our audio playback if necessary, by default set to 5. Modyfing the switch case inside the main function of *demo.c* we got the function that is now visible. Other modifications have been made to fnAudioPlay function that have been replaced by fnAudioPlay2 that takes one more input with respect to the original one and it is the address of the vector to playback. Any adding or modifications of the functions have been reported inside the *audio.h* file. Here is proposed a block diagram of the project and how it works:

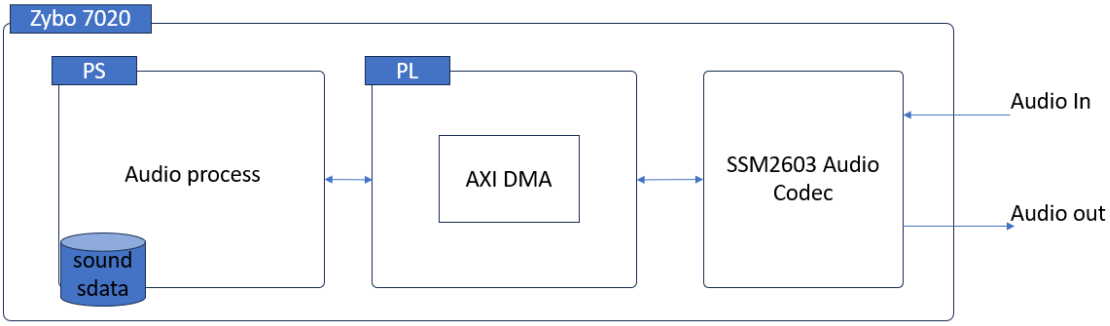

Figure 1: Block diagram of the project

The Zybo ZYNQ-7020 board is a critical component of our project, as depicted in the block diagram in Figure 1. It houses various integral elements such as the memory for storing the dataset, the ARM processor, and the Programmable Logic (PL) that includes the audio decoder. MATLAB plays a role outside of the board by generating the dataset. The dataset, comprised of all the processed Italian phonemes and an additional beep sound, is created using MATLAB and stored in the Zynq's onboard memory. The phonemes, as the basic elements of our text-to-speech system, allow the construction of words and sentences. The ARM processor within the Zynq system manages the operations. It's responsible for forming words and sentences by organizing the necessary phonemes stored in memory. Once a sentence is assembled, it is transferred via Direct Memory Access (DMA) to the Programmable Logic (PL) portion of the Zynq system. DMA is used for its efficiency and speed, making it ideal for real-time applications like ours. The PL section of the Zybo ZYNQ-7020 board integrates an audio decoder, specifically designed to handle audio data. This decoder takes the fixed-point phoneme data transferred via DMA and decodes it back into audio signals. The Zybo board uses a SSM2603 audio codec for this purpose, which interfaces with the rest of the system through an I2S protocol. After decoding, the audio signals are output through the Zybo board's audio out jack, allowing the synthesized speech to be heard through a connected speaker or set of headphones. In testing the system, we attempted to synthesize the sentence "Ciao sono Paolo" (English translation: "Hello, I'm Paolo"). However, due to the distinct manner in which each phoneme was recorded, the output gave a robotic impression. This highlighted the need for further refinement and post-processing to improve the naturalness and fluidity of the synthesized speech. Such enhancements will form an integral part of future iterations of this project.