Real Time System for Extraction and Playback of an Instrumental Sound

Laurentiu Mihai Ionescu, Alin Gheorghita Mazare, Ioan Lita, Nadia Belu
Dep. Electronics, Computers and Electrical Engineering
Faculty of Electronics, Communications and Computers,
University of Pitesti
Pitesti, Romania
ioan.lita@upit.ro

Adrian-Ioan Lita
Politehnica University of Bucharest
Bucharest, Romania

***Abstract*— This paper presents an acquisition system for sound recording, typical spectral component identification and extraction of particular instruments from a soundtrack, based on each instrument's spectral components. This system is also capable of sound reproduction (playback) and its main feature consists of minimizing the delay between the recorded sound and the playback sound below 100ms. To achieve all these objectives, the proposed system consists of several components which will operate in parallel: the acquisition module which outputs the digital waveform of the sound and its FFT transform used for spectral representation; the comparison module, composed of associative modules which compares the closeness of the sound sample to certain templates which contain spectral representation of pre-recorded instruments, used to identify whether certain instrument is present; the spectral form extraction module which isolates a certain instrument from the original sound-mix, based on filtering the spectral components that don't belong to the respective instrument; the last module is the inverse FFT module, which converts the sound from spectral domain back to time domain. With the exception of the microphone, the analog-to-digital converter (ADC) and the digital-to-analog converter (DAC) capable of outputting the sound, all other operations are done by an SoC integrated system (Xilinx Zynq 7000).**

Keywords*— System on Chip (SoC), real time, sound recognition, sound playback

I. INTRODUCTION

Analysis and processing of one-dimensional signals (such as sound) is done most often by software processing tools since the variety of algorithms allows flexibility and continuous improvement through development. This usually translates into an off-line sound analysis: the sound is recorded and at a later time is processed by the computer. There are several software solutions designed for sound recognition, but along with existing instruments new techniques for improving sound recognition algorithms are being researched, with applications in a large number of areas [1].

A class of applications require though real-time automated sound recognition. For applications requiring on-line sound analysis in order to identify sound patterns in real-time, hardware analysis tools are needed to be integrated in the area where the sound is captured, resulting a „sound sensing" solution based on microsystems with classic microprocessor (e.g. ARM) [2] or digital signal processors [3]. These solutions have some limitations related to sequential programming and complexity of the calculations. Another solution can be using digital structures from reconfigurable circuits, or using certain solutions which combine many types of circuits, some for spectral processing, and others for identifying certain models [4].

Dedicated algorithms for sound sequences recognition already exist. In principle, they are divided into three main classes: Gaussian Mixture Models, Support Vector Machines and Deep Neural Networks [5]. Each of these imply using a certain amount of resources which must be taken into account when implementation using hardware structures is desired. The overall cost of the system must be taken into account as well.

In the field of analyzing the sound outputted by musical instruments there are already solutions which use classic microprocessor architectures [6]. Their purpose is to determine either the sound quality starting from component isolation and analysis [7] or the overall accuracy of the melody by comparison of certain points to different expected template patterns [8].

This article presents an alternative processing solution capable to identify and re-play certain sound sequences, using both DSP cores, as well as hardware processing structures – a hybrid structure, while keeping in mind a reduced hardware cost. This is possible due to the integration of many components – DSP modules, microprocessor and dedicated logic structures - into a single SoC (System on a Chip).

The overall presented structure uses digital signal processors (called here DSP modules) to calculate and play samples of the recorded sound's spectral representation, and hardware integrated modules which identify certain instruments that are desired to be extracted from the soundtrack. The experiments where done using soundtracks with more instruments and extracting the sound of only a certain instrument. The central component of the system is a System on a Chip – Zynq 7000 from Xilinx, which accommodates under the hood the following modules: a

general purpose RISC processor (running Linux), which is used to select the chosen instrument to be extracted, DSP cores used for calculating FFT and inverse FFT, as well as an FPGA used for implementing the instrument template identifying modules.

Section II will present the overall system architecture, while Section III presents the experimental datasets and results.

II. System Architecture

The overall presented structure uses digital signal processors (called here DSP modules) to calculate and play samples of the recorded sound's spectral representation, and hardware integrated modules which identify certain instruments that are desired to be extracted from the soundtrack. The general architecture is presented in the figure below:

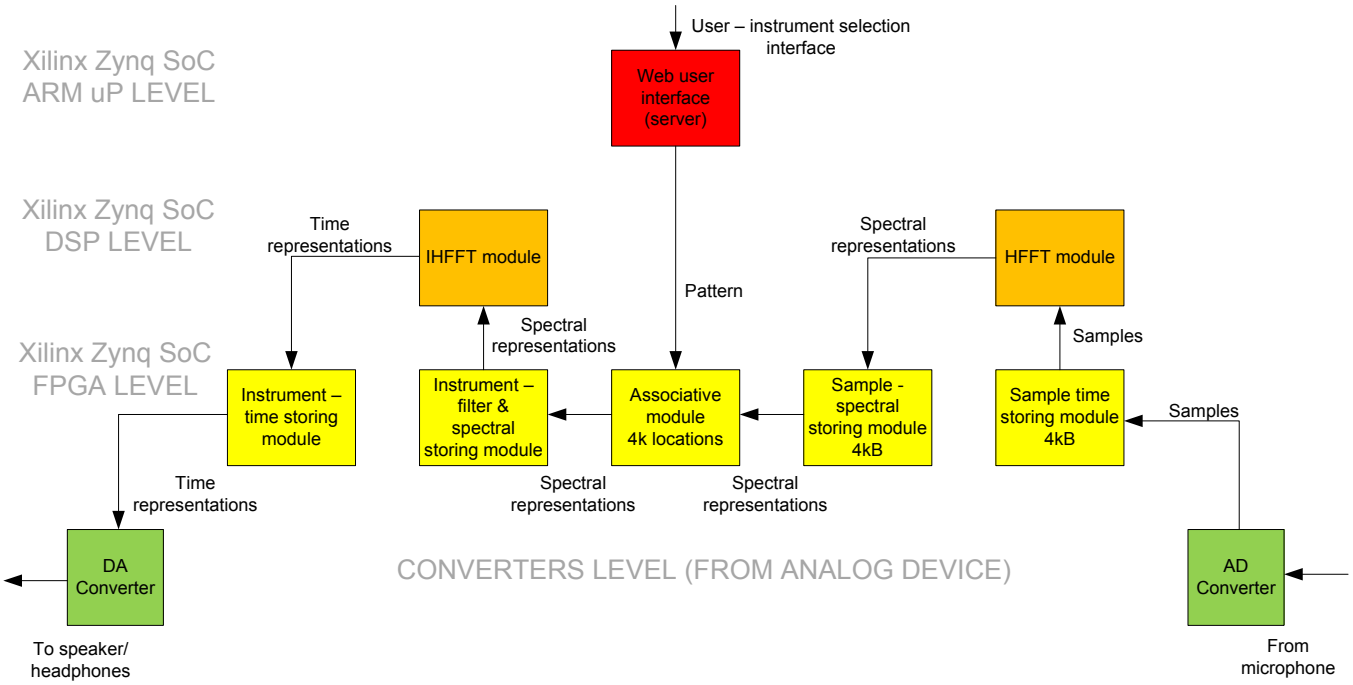


Fig. 1. System block diagram (multilevel hardware architecture)

As figure 1 depicts, the architecture was structured on 4 levels. The lowest level includes the components outside the SoC: the AD and DA converters. Analog Devices manufactures both converters used in this paper. The AD converter is SSM2603, which samples sound at a frequency of 96 KHz. The following levels are build inside the SoC Zynq 7000(XC7Z010):

The first level inside the SoC contains the custom hardware structures. These were implemented using the FPGA (based on Artrix 7 architecture). The custom hardware structures are the automated storing blocks for both the digital samples of sound over time and spectral representation, as well as the module responsible of recognizing patterns of sound.

In order to optimize the Fourier transformer module, memories are organized as 4K x 16. Synchronous high-speed SRAM memory was configured to work in bi-port mode, so that the blocks which need to store data and write the memory (such as the block which reads the AD converter) can work completely independent in terms of system frequency from the blocks which needs to read the memory (for example, the HFFT).

Another level implemented using the FPGA is the associative module. This uses an associative memory based on a Hopfield neural network. The main objectives are fast pattern detection on one side, and fast training. The associative memory is made with a network of comparators and sorting circuits.

The next level is represented by the DSP blocks. These are responsible with calculating the FFT and the IFFT. The DSP blocks already existing in the SoC XC7Z010 can process the transforms of the 4K x 16 samples stored in the internal memory.

The top level and the only level accessible to the user is the microprocessor level. The microprocessor is an ARM Cortex A9 which runs Linux operating system, on which a web based graphical interface application was developed to allow the user to select the musical instrument which is desired to be identified and isolated from the rest of the orchestra.

As figure 1 depicts, the architecture has a horizontal symmetry. The right side is responsible of acquiring and representing data, and the left side is responsible of filtering and outputting the signal. The central post is the instrument pattern recognition module, instrument selected through the web interface.

III. EXPERIMENTAL RESULTS

Experiments were conducted to identify, extract and playback desired instruments from a soundtrack, so that a sound specialist can monitor only a certain instrument at a time. The instrument can be extracted from any desired soundtrack. The experiments were conducted using three instruments: harpsichord, piano and trumpet, while the soundtrack also contains background noise (such as a theater at half break). The processing flow is depicted below:

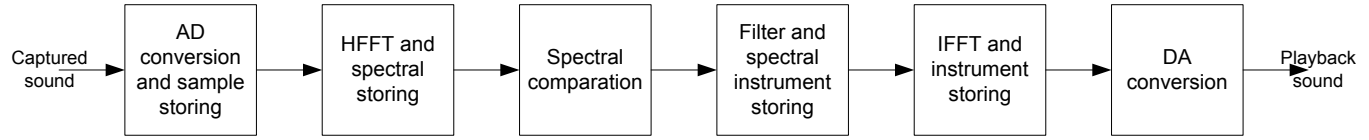


Fig. 2. Processing flow

A. Data set

The data set that was used in the experiments contains three studio recordings (with little to no noise) of the three instruments, from which spectral representation was extracted. The sample files were provided on Stanford University [9] website with instrument sound samples, and the spectral representation processing was done with MATLAB 2011 on a PC.

Using MATLAB, three other files were generated: each two instruments plus maximum 50 dB noise (coresponding to the noise level in a concert hall during break). The dataset consists of 6 files: 3 with original sound instruments and 3 with two instruments and noise mixed together.

B. Experimental results

When the experiments were conducted, three parameters were taken into consideration. The first is the response time to which a certain instrument is outputted as challenged by the input. Determination of the response time was conducted using the analysis and implementation tools of the Xilinx ISE Pack. The second parameter is related to the rate of success in detecting the instrument. The central component which identifies the spectrum of a certain instrument is, as shown above, the associative memory. This identifies the number of common points between the analyzed spectrum and the spectrum of a certain instrument. To experimentally visualize and determine the identification success rate, common spectral points between two instruments with different noise level were outputted. Common spectral points represents an undesired characteristic, which translates that on certain spectral zones two instruments sound similar and are hard to be separated one from another. The third parameter taken into consideration is the overall cost of the system.

The total processing time is maximum 100ms: this means that the playback sound will have a maximum delay of 0.1s from the original soundtrack. The processing time of each step of the flow is presented in the table below:

TABLE I. PROCESSING TIMES FOR EACH MODULE

Step	Time	Step	Time
AD conv. & storing	42.7 ms	Filtering &storing	41 us
HFFT & storing	221 us	H – IFFT & store	250 us
Comparison	700 us	DA conv.	43 ms

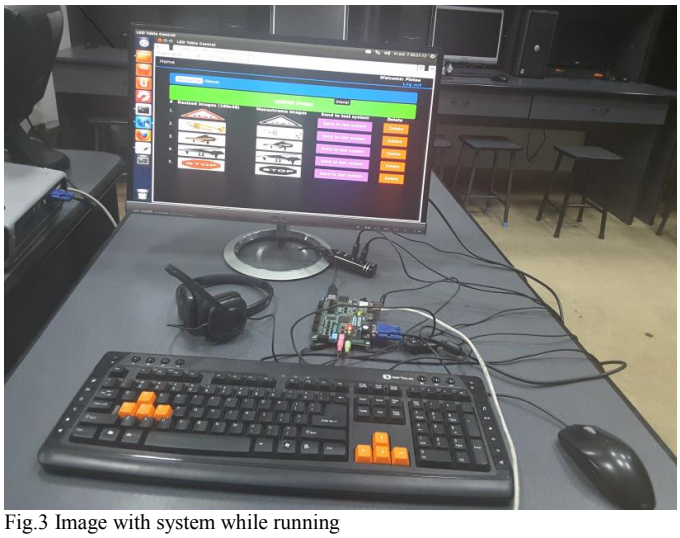


Fig.3 Image with system while running

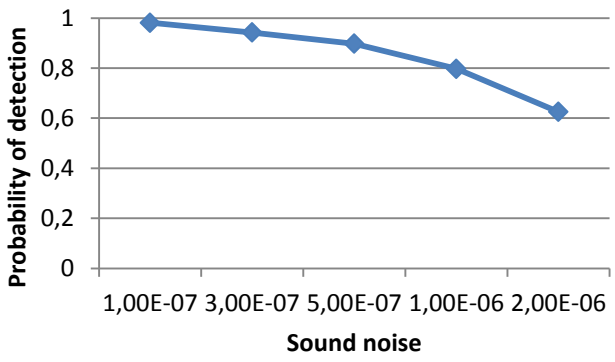
Along with the response time, another important parameter is the identification success rate. This highly depends on the associative memory modules, used for template matching. For soundtracks such as the one presented above, the success rate does not drop below 80%.

The total number of spectral points analyzed by the associative memory is 4096 (all spectral values stored). The analysis outputs the following common points between instruments in respect to the noise level power:

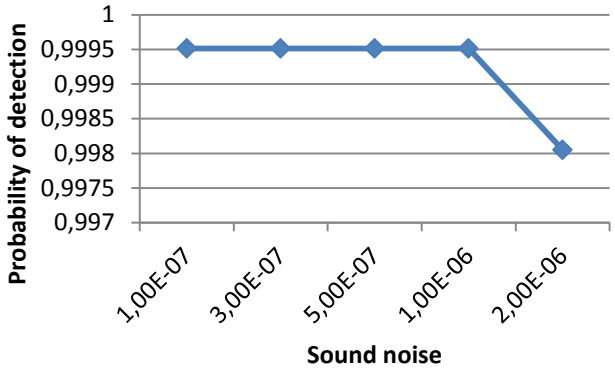
TABLE II. COMMON POINTS BETWEEN INSTRUMENTS WITH DIFFERENT NOISE LEVELS

Noise level (W)	Trumpet-Piano	Trumpet-Harpsichord	Piano-Harpsichord
1.00E-07	76	2	0
3.00E-07	238	2	0
5.00E-07	420	2	0
1.00E-06	832	2	0
2.00E-06	1536	8	6

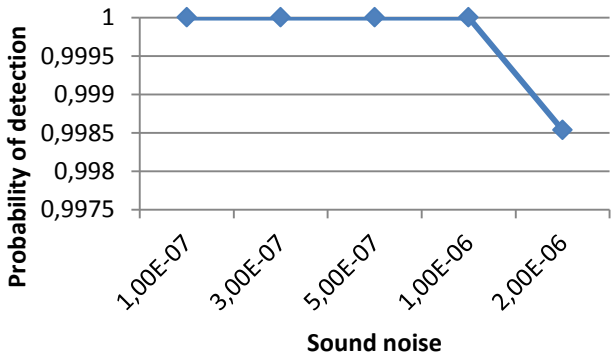
These points determine the detection probability of a certain instrument from a mix of two instruments, as follows:



a) Trumpet – piano mix



b) Trumpet- Harpsichord mix



c) Piano- Harpsichord mix

Fig.4 Probability of detection for different instrument from a mixed sound

The third analyzed parameter refers to the overall cost of the solution. As figure 3 depicts, a Zynq 7000 system development kit provided by Digilent was used. The cost of such a system, which also include the AD and DA converters, as well as the SoC is 189$, which is a reduced cost for a high performance sound sensing system.

IV. Conclusions

The system was designed, implemented and experimented. It is a real time solution for determining the quality of music by separately reproducing certain musical instruments. The small response time allows audition in parallel with the orchestra. The performance of correctly detecting an instrument is very high, allowing high accuracy detection even with external noise (room noise), at a low price.

Further research directions include building a larger instrument library (both in quantity and quality) and conducting experiments using more complex sounds.
In a larger area, the system can be used for identifying certain sound spectrums in other areas, such as security and automotive.

Acknowledgment

The research that led to the results shown here has received funding from the project "Cost-Efficient Data Collection for Smart Grid and Revenue Assurance (CERA-SG)", ID: 77594, 2016-19, ERA-Net Smart Grids Plus.

References

[1] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," IEEE Transactions on Multimedia, vol. 17, no. 10, pp. 1733–1746, 2015

[2] Tauhidur Rahman, Alexander T Adams, Mi Zhang, Erin Cherry, "BodyBeat: A Mobile System for Sensing Non-Speech Body Sounds", MobiSys'14, June 16–19, 2014, Bretton Woods, New Hampshire, USA, 2014

[3] Lianfu Han, Zhengguang Shen, Changfeng Fu, Chao Liu, "Design and Implementation of Sound Searching Robots in Wireless Sensor Networks", Sensors 2016, 16(9), 1550, 2016

[4] Siddharth Sigtia, Adam M. Stark , Sacha Krstulović , Mark D. Plumbley, "Automatic Environmental Sound Recognition: Performance Versus Computational Cost", IEEE/ACM Transactions on Audio, Speech, and Language Processing (Volume: 24, Issue: 11, 2016)

[5] Aris Tjahyanto, Diah Puspito Wulandari, Yoyon K. Suprapto, Mauridhi Hery Purnomo, "Gamelan instrument sound recognition using spectral and facial features of the first harmonic frequency", Acoustical Science and Technology, Vol. 36 (2015) No. 1 P 12-23

[6] T. Fujishima, „Realtime chord recognition of musical sound: A system using common lisp music", Proc. ICMC, pp. 464-467, 1999.

[7] O Romani Picas, H Parra Rodriguez, D Dabiri, „A Real-Time System for Measuring Sound Goodness in Instrumental Sounds", Society Convention 138, 2015

[8] A Arzt, A Widmer, „Real-time music tracking using multiple performances as a reference", Proc. of the International Society for Music, 2015

[9] https://ccrma.stanford.edu/~jos/pasp/Sound_Examples.html