FPGA Implementation of a Feature Extraction Technique based on Fourier Transform

Mohammed Bahoura
Department of Engineering,
Université du Québec à Rimouski,
300, allée des Ursulines, Rimouski, Qc, Canada.

Hassan Ezzaidi
Department of Applied Sciences,
Université du Québec à Chicoutimi,
555, boul. de l'Université, Chicoutimi, Qc, Canada.

***Abstract*— In this paper, a feature extraction method based on short-time Fourier transform (STFT) was implemented on FPGA for real-time pattern recognition. The proposed technique was implemented using Xilinx System Generator (XSG) in MATLAB/SIMULINK environment. The response signals obtained during feature extraction process of blue whale call are represented. Also, the classification performance based on the fixed-point XSG implementation is compared to that based on the floating-point MATLAB one using isolated blue whale calls.**

I. Introduction

The feature extraction is an important step in pattern recognition systems. It transforms originally high-dimensional patterns into lower dimensional vectors by capturing the essential of their characteristics [1]. Various feature extraction techniques have been proposed in the literature for different signal applications. In speech and speaker recognition, they are essentially based on Fourier transform, cepstral analysis, autoregressive modeling, and wavelet transform. These feature extraction techniques were also used in the recognition of musical instruments, biomedical signals, marine mammal vocalizations, etc.

Automatic recognition systems were firstly proposed and evaluated using software platform such as MATLAB. However, their hardware implementation remains a great challenge that requires a tradeoff between complexity, computation speed and efficiency of these systems. Most of the hardware-based architectures are proposed for speech and speaker recognition using digital signal processor (DSP) and field programmable gate array (FPGA) [2]–[4].

In this paper, we propose an FPGA-implementation of a feature extraction technique for real-time passive acoustic monitoring (PAM) system that can be used to identify and localize underwater mammals.

II. Feature Extraction

Fig. 1 represents the spectrograms of three typical blue whale calls followed by the used feature extraction technique. The vocalization signal is split into successive frames of N samples, $s(m,n)$, where m is the frame index and n is the sample time index within the given frame. Then, a short-time Fourier transform is applied to each frame to extract D-dimensional feature vector x_m. Feature extraction can be seen as a mapping $f : \mathbb{R}^N \rightarrow \mathbb{R}^D$, where $D \ll N$.

A. Short-time Fourier transform

The short-time Fourier transform (STFT) of a given frame $s(m,n)$ is a Fourier transform performed in successive frames:

$$S(m,k) = \sum_n s(m,n) e^{-j2\pi nk/N} \tag{1}$$

where $s(m,n) = s(n)w(n-mL)$, and $w(n)$ is a windowing function of N samples. This function is located at mL, where L is the shift-time step in samples. N represents also the number of discrete frequencies that is usually chosen to be a power-of-2 for using the fast Fourier transform (FFT). The overlap ratio between successive frames is $(N-L)/N$. The power spectrum density (PSD) is then computed [5], [6]

$$P_s(m,k) = \frac{1}{N}|S(m,k)|^2 \tag{2}$$

At the sampling frequency f_S, each windowed segment (frame) is represented by N-points PSD covering the frequency range $[-\frac{f_S}{2}, \frac{f_S}{2}[$. As power spectrum is symmetric, it can be described by only $N/2$ discrete frequencies. Each frequency index k represents a discrete frequencies $f_k = kf_S/N$, where $0 \leq k < N/2$.

B. Feature vector extraction

As shown in Fig. 1, this method consists in extracting features from two subbands, (15.137-20.508 Hz) and (38.574-84.961 Hz) corresponding to the AB and D calls frequency ranges, respectively. The first six components of the feature vector $\mathbf{x}_m$ are obtained by averaging PSD points between $P_s[m,31]$ and $P_s[m,42]$ by bins of 2 points. The last six features are similarly obtained for the PSD interval from $P_s[m,79]$ to $P_s[m,174]$ but with bins of 16 points. For a given frame m, the feature vector components are defined by

$$x_{m,n} = \begin{cases} \dfrac{1}{2} \displaystyle\sum_{k=29+2n}^{30+2n} P_s(m,k) & 1 \leq n \leq 6 \\ \dfrac{1}{16} \displaystyle\sum_{k=63+16(n-6)}^{78+16(n-6)} P_s(m,k) & 7 \leq n \leq 12 \end{cases} \tag{3}$$

Hence, a 12-dimensional feature vector is constructed, $\mathbf{x}_m = [x_{m,1}, x_{m,2}, \ldots, x_{m,12}]^T$.

978-1-4673-5292-5/12/$31.00 ©2012 Crown

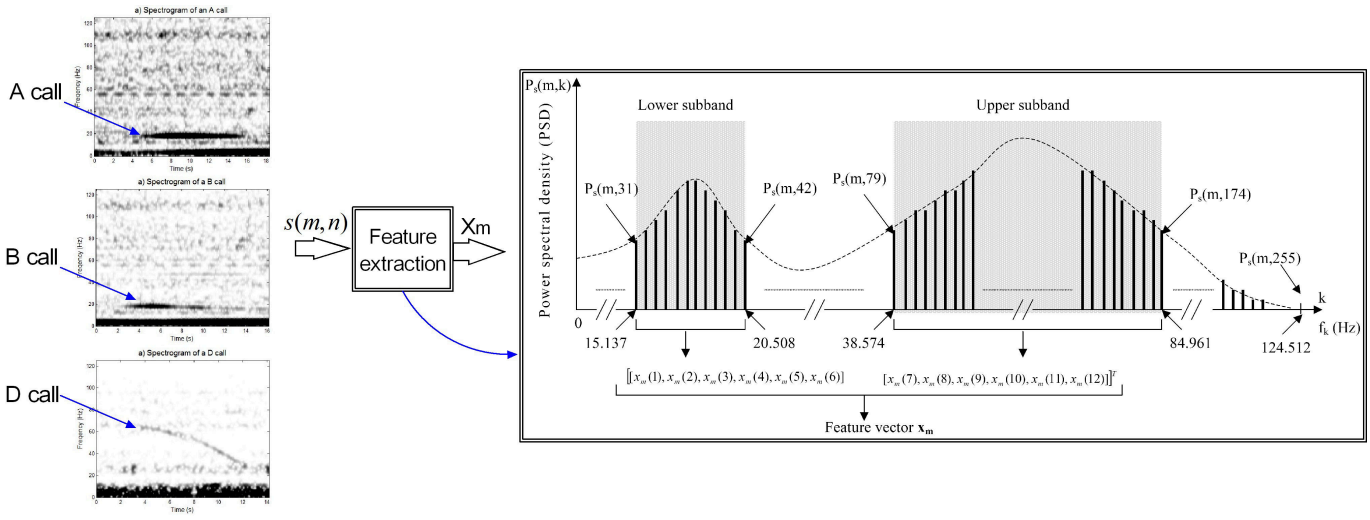

Fig. 1. Feature extraction process using short-time Fourier transform (STFT). Spectrograms of A, B, and D calls are given in the left followed by the computation of the feature vector from spectrum, which is described in the right [5]. For each signal frame $s(m,n)$, the feature vector $\mathbf{x}_m$ is constructed by extracting six components from lower subband (15.1375-20.508 Hz) and six components from upper subband (38.574-84.961 Hz). The discrete frequencies are obtained with a sampling frequency of 250 Hz and frame a length of 512 samples.

III. Classification

A multi-layer perceptron (MLP) neural network implemented on MATLAB is used to compare performances of the fixed-point architecture to the floating-point one. It is characterized by 12 inputs, 25 hidden neurons and 3 output neurons. We use an *hyperbolic tangent* function for hidden layer and a *logistic sigmoid* function for output layer [5]. The used database contains 100 isolated calls per class, where 90 calls are used for training and 10 calls for testing.

IV. FPGA implementation

The described feature extraction technique was implemented on FPGA using Xilinx System Generator (XSG) and the Virtex-6 FPGA ML605 Evaluation Kit. Fig. 2 presents the proposed architecture that is mainly based on the algorithm described in section II.

A. Signal windowing

The windowed signal $s(m,n)$ was obtained by multiplying $s(n)$ by Hamming window $w(n)$ of $N = 512$ samples stored in a ROM block driven by a cyclic modulo-N counter (Fig. 2).

B. Sort-time Fourier transform

The STFT of a given input frame, $s(m,n)$, is computed using the Xilinx FFT block.

$$S(m,k) = \sum_{n=0}^{N-1} s(m,n) e^{-jnk2\pi/N} \tag{4}$$

where N is the transform size, k is the frequency bin index ($0 \leq k \leq N-1$), and $j = \sqrt{-1}$. Pipelined streaming input/output option has been chosen to achieve continuous computation of the short-time Fourier transform. The Xilinx FFT block provides two outputs corresponding to real, $S_r(m,k)$, and imaginary, $S_i(m,k)$, parts of $S(m,k)$.

C. Power Spectrum

The power spectrum block computes power spectrum using the real and imaginary parts provided by the Xilinx FFT block.

$$P_s(m,k) = \left(S_r^2(m,k) + S_i^2(m,k)\right)(1/N) \tag{5}$$

where N is the frame length of 512 samples.

D. Feature extraction

For each frame $s(m,n)$, the feature extraction block in Fig. 2 computes its 12 feature components according to Eq. (3). It can be noted that division by 2 and 16 and replaced by multiplication by $1/2$ and $1/16$, respectively. Each feature component block uses a register enabled by the frequency index (k) to save it. A second register enabled by the "done" signal of the FFT block is used to synchronize the feature component values at the processing end of each frame.

E. One value per frame

An optional subsystem in Fig. 2 based on the standard Simulink blocks is used to take one feature vector per frame rather than repeated N identical vectors.

Table I gives the resource requirement and the maximum operating frequency as reported by Xilinx ISE tool.

TABLE I

Resource utilization and maximum operating frequency of the Virtex-6 XC6VLX240T chip using CLB logic structure.

Resource utilization	
Flip Flops (301,440)	13,2385 (4.4%)
LUTs (150,720)	12,014 (8.0%)
Bonded IOBs (600)	306 (51.0%)
RAMB18E1s (832)	62 (0.2%)
DSP48E1s (768)	9 (1.2%)
Maximum Operating Frequency	66.061 MHz

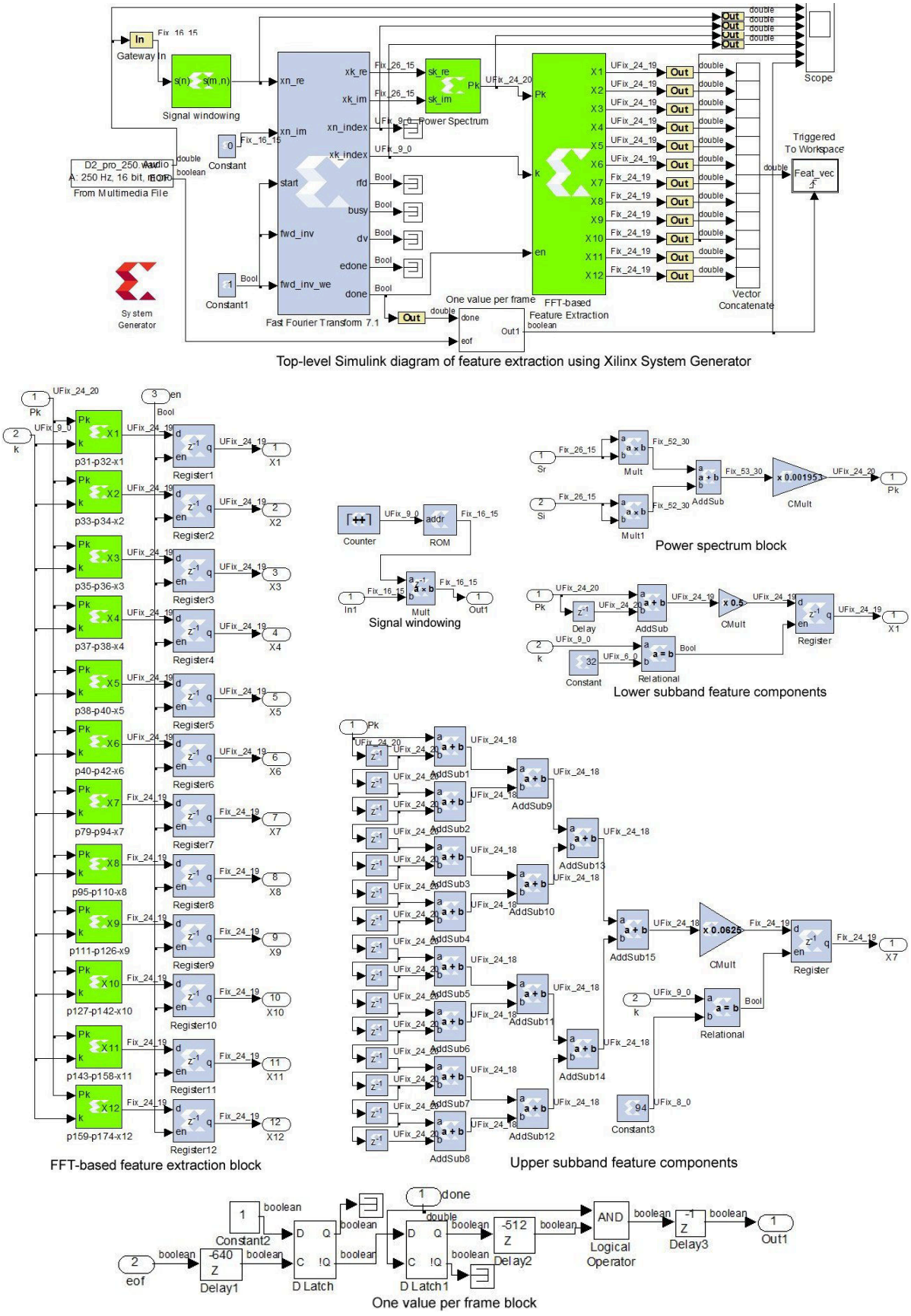

Fig. 2. Feature extraction technique implemented on FPGA using Xilinx System Generator. The top-level Simulink diagram is given on the top followed by details of different subsystems. The green blocks are designed using the XSG blocks (blue). The white blocks are the standard Simulink blocks.

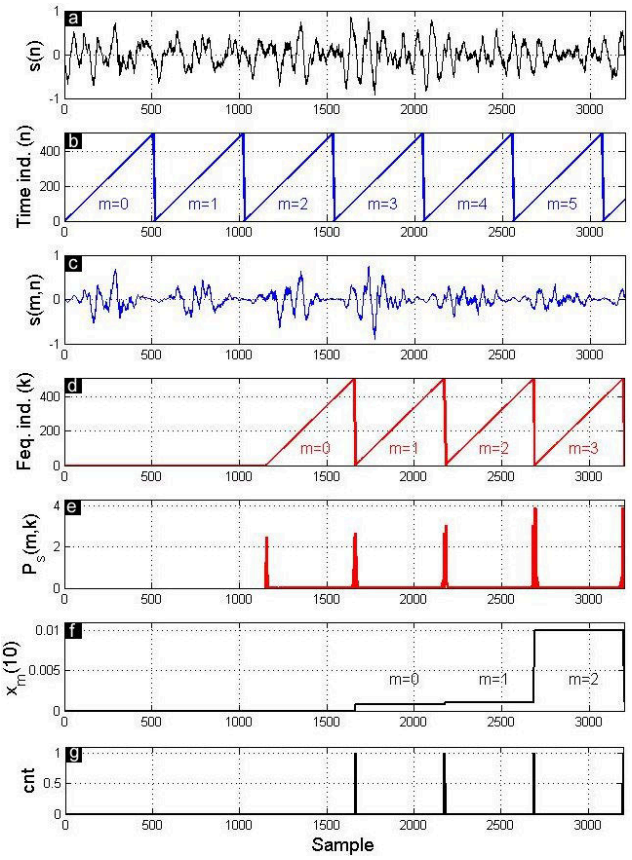

Fig. 3. Response signals obtained during feature extraction process of a D call. a) input signal $s(n)$, b) Time index (n) that allows to delimit successive frames $s(m,n)$, c) windowed signal $s(m,n)$, d) Frequency index (k) that allows to delimit successive power spectra $P_s(m,k)$, e) Power spectra $P_s(m,k)$, f) 10th component of the feature vector, and g) control signal that send one feature vector per frame to workspace.

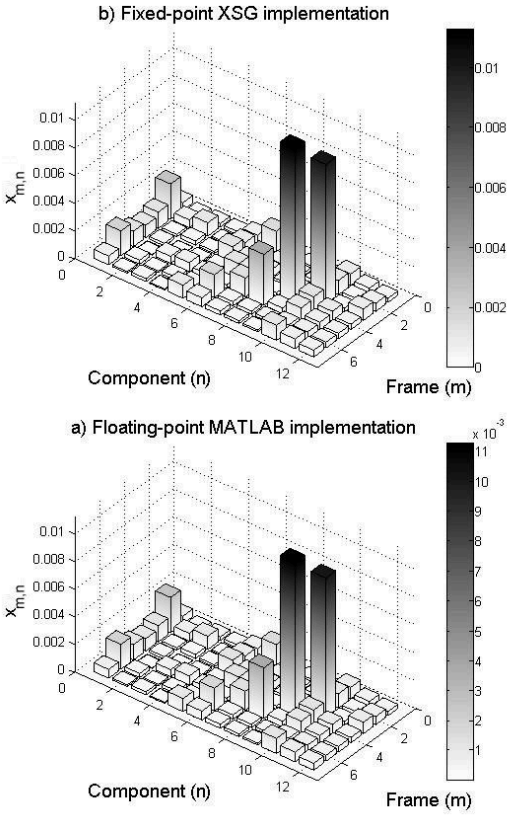

Fig. 4. Example of feature vectors obtained from a D call using fixed-point XSG implementation (a), and floating-point MATLAB implementation (b).

V. Results

Fig. 3 presents response signals at different levels of this architecture. It can be seen the FFT computing latency of 1150 samples that represents the delay between the first sample of the input frame $s(m,n)$ and the first sample of its corresponding spectrum $P_s(m,k)$. As pointed previous in subsection IV-D, the feature components are available just from the end of their corresponding power spectrum $P_s(m,k)$. Fig. 4 shows that fixed-point XSG implementation gives the same performance as floating-point MATLAB implementation. Table II gives the confusion matrix of the true classes versus assigned classed for fixed-point and floating point implementations of the STFT-based feature extraction method. These results present the averaged values of 50 repeated tests.

TABLE II
Confusion matrix of MATLAB and XSG based implementations.

True class	Assigned class (XSG)			Assigned class (Matlab)		
	A	B	C	A	B	C
A	6.72	3.02	0.26	6.72	3.02	0.26
B	0.24	8.20	1.56	0.24	8.20	1.56
D	0.00	0.00	10.00	0.00	0.00	10.00

VI. Conclusion

In this paper, feature extraction technique based on Fourier transform has been implemented on FPGA using Xilinx System Generator. The fixed-point XSG implementation provides similar performances than the floating-point MATLAB one. As future work, a classifier as a neural network will be added to this architecture to implement complete pattern recognition system on FPGA.

References

[1] M. Bahoura, "Pattern recognition methods applied to respiratory sounds classification into normal and wheeze classes," *Computers in Biology and Medicine*, vol. 39, no. 9, pp. 824–843, 2009.

[2] J.-C. Wang, J.-F. Wang, and Y.-S. Weng, "Chip design of MFCC extraction for speech recognition," *Integration, the VLSI Journal*, vol. 32, pp. 111–131, 2002.

[3] M. Staworko and M. Rawski, "FPGA implementation of feature extraction algorithm for speaker verification," in *17th International Conference "Mixed Design of Integrated Circuits and Systems", MIXDES 2010*, 2010, pp. 557–561.

[4] R. Ramos-Lara, M. López-García, E. Cantó-Navarro, and L. Puente-Rodriguez, "Real-time speaker verification system implemented on reconfigurable hardware," *Journal of Signal Processing Systems*, pp. 1–15. [Online]. Available: http://dx.doi.org/10.1007/s11265-012-0683-5

[5] M. Bahoura and Y. Simard, "Blue whale calls classification using short-time fourier and wavelet packet transforms and artificial neural network," *Digital Signal Processing*, vol. 20, no. 4, pp. 1256–1263, 2010.

[6] ——, "Serial combination of multiple classifiers for automatic blue whale calls recognition," *Expert Systems with Applications*, vol. 39, no. 11, pp. 9986–9993, 2012.