On the Designing of Spikes Band-Pass Filters for FPGA

M. Domínguez-Morales, A. Jimenez-Fernandez, E. Cerezuela-Escudero,
R. Paz-Vicente, A. Linares-Barranco, and G. Jimenez[*]

Departamento de Arquitectura y Tecnología de Computadores,
ETS Ingeniería Informática - Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012-Sevilla, Spain
{mdominguez,ajimenez,ecerezuela,rpaz,
alinares,gaji}@atc.us.es

Abstract. In this paper we present two implementations of spike-based band-pass filters, which are able to reject out-of-band frequency components in the spike domain. First one is based on the use of previously designed spike-based low-pass filters. With this architecture the quality factor, Q, is lower than 0.5. The second implementation is inspired in the analog multi-feedback filters (MFB) topology, it provides a higher than 1 Q factor, and ideally tends to infinite. These filters have been written in VHLD, and synthesized for FPGA. Two spike-based band-pass filters presented take advantages of the spike rate coded representation to perform a massively parallel processing without complex hardware units, like floating point arithmetic units, or a large memory. These low requirements of hardware allow the integration of a high number of filters inside a FPGA, allowing to process several spike coded signals fully in parallel.

Keywords: Spike processing, neuromorphic engineering, band-pass filter.

1 Introduction

Neuromorphic engineers work in the study, design and development of neuro-inspired systems, like aVLSI chips for sensors [1][2], neuro-inspired processing, filtering or learning [3][4][5][6], neuro-inspired control pattern generators (CPG) , neuro-inspired robotics [7][8][15] and so on. Spiking systems are neural models that mimic the neuron layers of the brain for processing purposes. Signals in the spike-domain are composed of short pulses in time, called spikes [9]. Information is carried by the spike frequency or rate [11], following a Pulse Frequency Modulation (PFM) scheme, and from other point of view, in the inter-spike-time (ISI) [6]. Spike rate coded information can be processed in a continuous way, without codifying information into discrete samples. Spike processing hardware can be easily replicated performing a massively parallel information processing [12].

In this context, previous work presented a set of building blocks, to process spike coded signals in real time [17], mimicking elementary analogical components to

[*] This work has been supported by the Spanish project VULCANO (TEC2009-10639-C04-02).

design spike filters, equivalent to classical analogical filters but filtering spikes in time (Spike Signal Processing, SSP). In [17] it was thoroughly explained how to design and tune spike-based low-pass (SLPF) and high-pass filters (SHPF). This paper presents in detail how we can design spike band-pass filters (SBPF), using the elements presented in [17]. So a short description and operation theory of previous elements for SSP is presented in section 2. In section 3, a low quality, Q, factor SBPF is designed using simply SLPF. In section 4, inspired in the topology of multi-feedback (MFB) band-pass filters, we design and analyze a high Q factor SBPF, were Q can be ideally infinite. Both filters topologies have been simulated with the help of MATLAB Simulink and the addition Xilinx SystemGenerator, which allows integrating, among many other things, VHDL files with Simulink common elements.

2 Building Blocks for SSP

In this section we will make a short introduction to basic building blocks and full gain adjustable SLPF [17], because this information is necessary to understand SBPF.

2.1 Spike Synthetic Generator (SSG)

This element generates a spikes sequence from a discrete digital number, and will generate the stimulation of spike-based filters simulations. We use in particular the Reverse Bit-wise Synthetic Spike Generator (RB-SSG) found in [14]. RB-SSG is based in digital counters, and it does not use probabilistic techniques, providing deterministic and well characterized spike rates or ISI. RB-SSG generates a spike stream from an N-bits signed discrete number, being the output spike rate proportional to the input value, X, as (1) denotes. RB-SSG gain, $k_{RB\text{-}SSG}$ can be calculated according to (2).

$$RB - SSG_{SpikeRate} = k_{RB-SSG} * X \tag{1}$$

$$k_{RB-SSG} = \frac{F_{CLK}}{2^{N-1}(genFD + 1)} \tag{2}$$

F_{CLK} is the FPGA clock frequency, N input signal number of bits, and genFD is a clock divider used to obtain accurate values of $k_{RB\text{-}SSG}$. RB-SSG is able to generate positive and negative spikes, so two 1-bit signals are necessary to propagate generated spikes, positive and negative by different channels.

2.2 Spikes Integrate and Generate (SI&G)

SI&G performs the temporal integration operation, which means to provide a stream of spikes, with a spike rate proportional to the integration of input spikes. Fig. 1 shows internal SI&G blocks, where input spikes enter the left block and output integrated spikes are generated by the right block. SI&G has a digital counter connected to a RB-SSG. In order to integrate spikes we can use an N-bit digital up/down counter (whit saturation), where a positive spike increases by one counter value, and a negative spike decreases in the same quantity. Spike counter output provides a digital N-bit number; therefore it is necessary to convert spikes integration to a spike stream. For this task a RB-SSG is used.

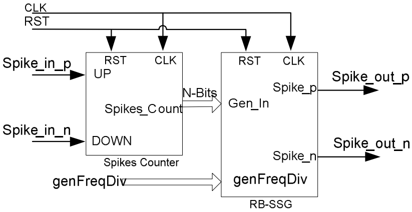

Fig. 1. SI&G internal blocks: Counter and RB-SSG

The RB-SSG included inside the SI&G will provide a spike stream, with a spike rate proportional to spikes counter, or integration. SI&G operation is equivalent to a continuous integration, where SI&G gain, $k_{SI\&G}$ is equivalent to $k_{RB\text{-}SSG}$ and can be calculated using (2). Making an analogy to continuous systems, we can calculate equivalent SI&G transfer function applying Laplace Transform, there is a result:

$$SI\&G(s) = \frac{F_{SI\,\&G}(s)}{F_{inputSpikes}(s)} = \frac{k_{SI\&G}}{s} = \frac{F_{CLK}}{2^{N-1}(genFD+1)} * \frac{1}{s} \tag{3}$$

2.3 Spikes Hold and Fire (SH&F)

In this section we present an element designed to subtract signed spikes, that have been designed and used successfully for spike filter designs [17] and spike-based closed-loop robotic controllers [15]. Subtracting a spike-based input signal (f_U) to another (f_Y), will yield a new spike signal whose spike rate ($f_{SH\&F}$) will be the difference between both input spike rates (4):

$$f_{SH\&F} = f_U - f_Y \tag{4}$$

The procedure of the SH&F is to hold the incoming spikes a fixed period of time while monitoring the input evolution to decide output spikes: holding, canceling, or firing spikes according to input spike ports and signs.

2.4 Spike Frequency Divider (SFD)

This block will divide the spike rate of an input signal by a constant. SFD have implemented inspired in the way that RB-SSG works. Fig. 3 shows SFD internal components. In SFD *spikesDiv* behaves like the constant to divide. Output buffers only enable output spikes according to an N-bit divider, *spikesDiv,* homogenously in time. SFD transfer function can be calculated using (6), where N represents the SFD number of bits. SFD transfer function is equivalent to a gain block, k_{SFD}, with a value in the range of [0, 1], with 2^N possible steps. SFD accuracy can also be adjusted with N (N=16 bits in Fig.5) getting an accuracy of 2^{-N}.

$$k_{SFD} = \frac{F_{outSpikes}}{F_{inputSpikes}} = \frac{spikesDiv}{2^N} \tag{5}$$

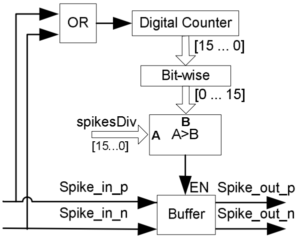

Fig. 3. SFD internal components

2.4 Spike Low-Pass Filters (SLPF)

A SLPF will attenuate high frequency components of spikes rate signals. In order to build a SLPF, a SI&G has been feedbacked using a SH&F and two SFD, as it is shown in Fig. 4. The idea is to integrate input spikes, but subtracting output integrated spikes from the integrator, with the help of a SH&F. Two SFD have been included in order to provide an adjustable gain. Taking into account SLPF components topology and transfer functions shown in (3), (4) and (5), SLPF transfer function can be calculated as shown in (6):

$$F_{SLPF}(s) = \frac{F_{outSpikes}(s)}{F_{inputSpikes}(s)} = \frac{k_{SFDOut} * SI\&G(s)}{1 + k_{SFDFB} * SI\&G(s)} = \frac{k_{SFDOut} * k_{I\&G}}{s + k_{SFDFB} * k_{I\&G}} \tag{6}$$

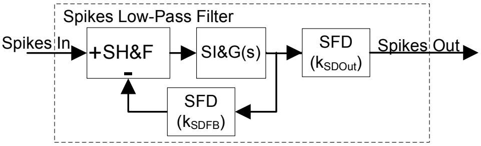

Fig. 4. SLPF blocks topology

SLPF is equivalent to a first order low-pass filter, with adjustable cut-off frequency ($\omega_{cut\text{-}off}$ in rads/sec) and gain (k_{SLPF}):

$$\omega_{cut-off} = k_{SFDFB} * k_{I\&G}\ , \quad k_{SLPF} = \frac{k_{SFDOut}}{k_{SFDFB}} \tag{7) (8}$$

3 Low Q-Factor Spikes Band-Pass Filter (LQ-SBPF)

In order to design a SBPF, our first approximation was using SLPF. The idea is to use two SLPF, with different cut-off frequencies, subtracting to the highest cut-off frequency SLFP output spikes, the low cut-off frequency SLPF spikes, as it is shown in Fig. 5. LQ-SPBF transfer function can be calculated from SLPFs transfer functions:

$$F_{SBPF}(s) = F_{SLPF_{HF}}(s) - F_{SLPF_{LF}}(s) = \\ \frac{k_{spikesDivHF} * k_{BWSpikesGenHF}}{s + k_{spikesDivFBHF} * k_{BWSpikesGenHF}} - \frac{k_{spikesDivLF} * k_{BWSpikesGenLF}}{s + k_{spikesDivFBLF} * k_{BWSpikesGenLF}} \tag{9}$$

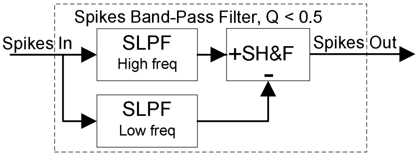

Fig. 5. LQ-SBPF subtracting two SLPF with different cut-off frequencies

If both SLPF gain are equal to 1, $k_{spikesDivHF}$= $k_{spikesFBHF}$= $k_{spikesDivLF}$= $k_{spikesFBLF}$, (9) is simplified, can be compared with ideal second order continuous band-pass filter, in (10). So adjusting SLPF cut-off frequencies, we can design LQ-SBPF with a desired mid band pass frequency ω_{mid} (11), Q factor (12) and SBPF gain (13). Since both SLPF poles are real, equivalent SBPF poles are also real, so Q≤0.5.

$$F_{Ideal\ BPF}(s) = k_{BPF} \frac{s * \omega_{mid}/Q}{s^2 + s * \omega_{mid}/Q + \omega_{mid}{}^2} \tag{10}$$

$$\omega_{mid} = \sqrt{\omega_{HF} * \omega_{LF}}\,, \qquad Q = \frac{\sqrt{\omega_{HF} * \omega_{LF}}}{\omega_{HF} + \omega_{LF}}, \qquad k_{BPF} = \frac{\omega_{HF} - \omega_{LF}}{\omega_{HF} + \omega_{LF}} \qquad (11)\ (12)\ (13)$$

We have simulated LQ-SBPF using a simulation framework described in [10]. First simulation of LQ-SBPF can be found in Fig. 6. We have set the right parameters to obtain a LQ-SBPF with a mid frequency of 3kHz, and a Q factor of 0.4. We can see the input spike rate frequency, in blue, composed of 3 tones, 300Hz, 3kHz, and 30kHz with different amplitudes. In red, the response of an ideal BPF with same characteristics, and in black, the reconstruction of the LQ-SBPF output spike frequency. Fig. 6 evidences how the low and high frequency tones are removed, remaining only the spikes of the 3kHz tones. For analyzing the LQ-SBPF frequency response, we have designed a set of LQ-SBPFs and simulated each of them for different tone frequencies. Making a frequency sweep, and annotating LQ-SBPF output spike rate for each tone. Using this we can calculate a Bode diagram. Fig. 7 shows Bode diagram for LQ-SBPF together with their ideal response, in discontinuous traces. Simulated filters have two mid-frequencies, 500Hz and 5kHz, and different Q factors, from 0.3 to 0.5. LQ-SBPF Bode diagram denotes that the filters have a correct behavior according to the theoretical model, rejecting out of band components with a slope of 20dB/decade.

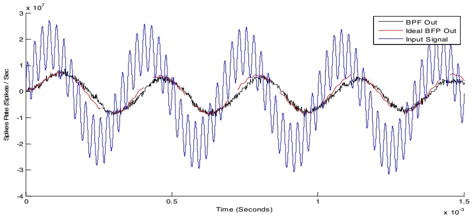

Fig. 6. LQ-SBPF 3-tone temporal simulation

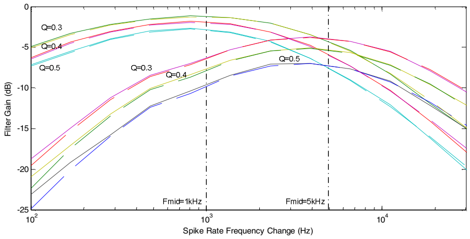

Fig. 7. Diverse LQ-SBF bode diagram

4 High Q-Factor Spike Band-Pass Filter (HQ-BPF)

One common topology used to build high Q BPF, is the multi-feedback (MFB) topology. Applying the principles of MFB filters, the HQ-BPF topology is shown in Fig. 8. This filter is basically a SLPF, with the difference that a second SI&G has been added in the feedback loop. Spikes coming from this SI&G and the SFD, also present in SLPF, are added using an SH&F that implements the multi-feedback.

We can calculate HQ-BPF equivalent transfer function applying basic systems theory. First of all, the transfer function of the elements placed at the feedback can be:

$$FB(s) = k_{SFD} + SI\&G(s)_{FB} = \frac{k_{SFD} * s + k_{SI\&G\_FB}}{s} \tag{14}$$

Using (14) we are able to calculate HQ-BPF equivalent transfer function:

$$BPF(s)_{HQ} = \frac{I\&G(s)}{1 + FB(s) * I\&G(s)} = \frac{k_{SI\&G} * s}{s^2 + k_{SI\&G} * k_{SFD} * s + k_{SI\&G} * k_{SI\&G\_FB}} \tag{15}$$

If we assume that both SI&G have the same gain, $k_{SI\&G}$=$k_{SI\&G\_FB}$, we can extract filter features from (15), being the mid-frequency equivalent to $k_{SI\&G}$, in (16) and the Q factor, the inverse of SFD gain(k_{SFD}), in (17). As theoretical equations denote, HQ-SBPF is tunable using only two parameters: SI&G gain and SFD gain.

$$\omega_{mid} = \sqrt{k_{SI\&G}{}^2} = k_{SI\&G} \quad Q = \frac{k_{SI\&G}}{k_{SI\&G} * k_{SFD}} = \frac{1}{k_{SFD}} \qquad (16)\ (17)$$

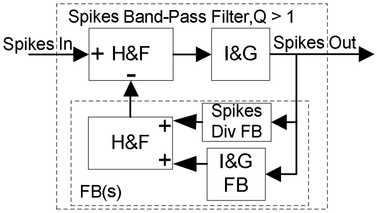

Fig. 8. Spikes-based building blocks topology for HQ-SBPF

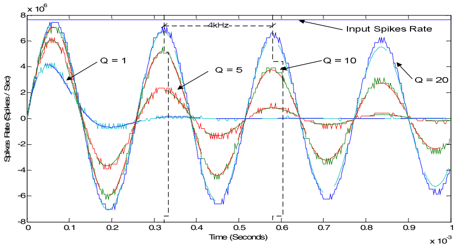

Fig. 9. HQ-SBPF with diverse Q factors, step response

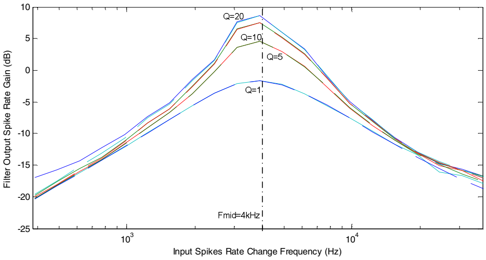

Fig. 10. Diverse Q factor HQ-SBF bode diagram

Similar simulations presented with LQ-BPF have been made for HQ-BPF. Fig. 9 shows step response of several HQ-BPF with equal mid-frequencies, at 4kHz, but different Q factors. When a step is present, only the frecuencial component of the mid-frequency can go through the HQ-BPF. However, this filter presents two complex poles and a sub-damped response, introducing an oscillation to output frequency. Frecuencial simulations have been made in to determine the Bode diagram of HQ-SBPF, which can be found in Fig. 10. For this figure, we have fixed the HQ-SBPF mid-frequency at 4 kHz, and modified the Q factor. These filters have resonance frequencies placed at $k_{SI\&G}$ (rad/sec), and the Q factor can be controlled using k_{SFD}. This figure also includes ideal BPF responses, these are very accurate with HQ-SBPF output, as predicted from equations (15), (16) and (17).

5 Conclusions

This paper has presented two new elements for Spike Signal Processing (SSP), previously existent elements implement basic arithmetic operations and low/high-pass spike-based filters. Based on these basic elements and filters we have designed two spike band-pass filters, one with different ranges of Q factors. SBPF are being used nowadays in our labs for many practical applications, for example a Synthetic AER Cochlea that uses LQ-SBPF, spike-based frequency-domain modulation for transmitting a lot of spike-coded information, as it is made in radio communications.

References

1. Lichtsteiner, P., et al.: A 128×128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor. IEEE Journal on Solid-State Circuits 43(2) (2008)
2. Chan, V., et al.: AER EAR: A Matched Silicon Cochlea Pair With Address Event Representation Interface. IEEE TCAS I 54(1) (2007)
3. Serrano-Gotarredona, R., et al.: On Real-Time AER 2-D Convolutions Hardware for Neuromorphic Spike-Based Cortical Processing. IEEE TNN 19(7) (2008)
4. Oster, M., et al.: Quantifying Input and Output Spike Statistics of a Winner-Take-All Network in a Vision System. In: IEEE International Symposium on Circuits and Systems, ISCAS (2007)
5. Hafliger, P.: Adaptive WTA with an Analog VLSI Neuromorphic Learning Chip. IEEE Transactions on Neural Networks 18(2) (2007)
6. Indiveri, G., et al.: A VLSI Array of Low-Power Spiking Neurons and Bistables Synapses with Spike-Timing Dependant Plasticity. IEEE Trans. on Neural Networks 17(1) (2006)
7. Gomez-Rodriguez, F., et al.: AER Auditory Filtering and CPG for Robot Control. In: IEEE International Symposium on Circuits and Systems, ISCAS 2007 (2007)
8. Linares-Barranco, A., et al.: Using FPGA for visuo-motor control with a silicon retina and a humanoid robot. In: IEEE International Symposium on Circuits and Systems, ISCAS 2007 (2007)
9. Shepherd, G.: The Synaptic Organization of the Brain. Oxford University Press, Oxford (1990)
10. Jimenez-Fernandez, A., et al.: Simulating Building Blocks for Spikes Signals Processing. In: International WorkShop in Artificial Neural Networks , IWANN 2011 (2011)
11. Mahowald, M.: VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function. PhD. Thesis, California Institute of Technology Pasadena, California (1992)
12. Serrano-Gotarredona, R., et al.: CAVIAR: A 45k-neuron, 5M-synapse AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking. IEEE Trans. on Neural Networks 20(9) (2009)
13. Gomez-Rodriguez, F., Paz, R., Miro, L., Linares-Barranco, A., Jimenez, G., Civit, A.: Two hardware implementations of the exhaustive synthetic AER generation method. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 534–540. Springer, Heidelberg (2005)
14. Paz-Vicente, R., et al.: Synthetic retina for AER systems development. In: International Conference on Computer Systems and Applications, AICCSA 2009 (2009)
15. Jimenez-Fernandez, A., et al.: AER-based robotic closed-loop control system. In: IEEE International Symposium on Circuits and Systems, ISCAS 2008 (2008)
16. Jimenez-Fernandez, A., et al.: AER and dynamic systems co-simulation over Simulink with Xilinx System Generator. In: IEEE Int.Symp. on Circuits and Systems, ISCAS 2008 (2008)
17. Jimenez-Fernandez, A., et al.: Building Blocks for Spike-based Signal Processing. In: IEEE International Joint Conference on Neural Networks, IJCNN 2010 (2010)