**Current Development in Theory and Applications of Wavelets**
Volume 5, Number 1, 2011, Pages 1-34
Published Online: August 31, 2011
This paper is available online at http://pphmj.com/journals/cdtaw.htm


# HARDWARE IMPLEMENTATION OF WAVELET TRANSFORMS FOR REAL-TIME DETECTION AND COMPRESSION OF BIOPOTENTIALS IN NEURAL IMPLANTS

**BENOIT GOSSELIN[1,2], SAIED HOSSEINI-KHAYAT[3], ADAM QUOTB[1] and MOHAMAD SAWAN[1]**

[1]Polystim Neurotechnologies Laboratory
Department of Electrical Engineering
École Polytechnique de Montréal
Montréal, QC H3C 3A7, Canada

[2]Department of Electrical and Computer Engineering
Université Laval
Québec, QC G1V 0A6, Canada

[3]Department of Electrical Engineering
Ferdowsi University of Mashhad
Iran

**Abstract**

Progress in the area of neural signal processing has enabled rapid advances in neuroscience and brain-machine interfacing. Wavelets have already been studied for detection and compression of neural action potentials. Among the major challenges in realizing a practical brain-machine interface is the design of ultra-low power electronic circuits. This paper is a survey of our efforts at Polystim Neurotechnologies Lab in designing wavelet transform (WT) processors for neural recording

2010 Mathematics Subject Classification: 42C40.

Keywords and phrases: discrete and continuous wavelet, neural signals, hardware implementation, brain-machine interface, implantable circuits and systems.

Received December 29, 2010

implantable devices. We describe the issues and possible solutions in designing such processors. In particular, we elaborate on the design of discrete and continuous WT processors using digital and analog components, respectively. Our results show that a folded polyphase architecture can produce a low-area low-power digital implementation. Also, it is shown that a low-power implementation of continuous WT can highly benefit from log-domain filter design and CMOS transistors operated in the weak inversion regime.

## 1. Introduction

Recording extracellular neural biopotentials from many sites in the cortex is becoming a necessity for research in neuroscience. This approach allowed striking advances, such as linking intents of moving specific limbs with the neural activity recorded from a patient's brain [1, 2]. Wirelessly controlled neural recording implants are a key component in the development of new research tools and emerging treatments based on this experimental method [3, 4]. Such miniature devices highly benefit from the progress in micromachining techniques and integrated circuits technology. In fact, there is a growing effort to build high-fidelity signal recording microsystems providing increased resolution and dense parallelism [5-11]. Wavelet signal processing plays a key role in such application for the detection and the compression of neural waveforms.

Our team is currently building a dedicated neural recording implant that includes custom micro-powered neural amplifiers [12], low-power data acquisition and processing circuits [13-15], advanced data management processors, and dedicated controllers [16, 17]. Low-power telemetry is designed to supply the implanted circuits, and enabling bidirectional communications for control of the implant and transferring the recorded signals outside the body, for off-chip treatment and analysis. Wireless operation of such implanted interface will reduce complexity of the recording apparatus and avoids external leads, allowing full mobility of implanted subjects and low risk of infection.

However, the level of power and data rates transmitted over today's wireless links is severely limited due to (1) safety and (2) low data transfer efficiency. This in turn restricts the maximum number of recording channels served by a single wireless link. For example, a 100-channel neural implanted device operating at a sufficient sampling rate generates bit rates on the order of tens of Mbits/s, whereas present low-power telemetry links are able to transmit only 10% of such data rate.

Therefore, suitable mechanisms for data compression must be provided in implants in order to extract relevant information from raw neural signals and to reduce the data rate to be transmitted. Low-complexity custom integrated circuits have been used for real-time data compression in implants [6-8]. However, low-complexity schemes usually have the disadvantage of yielding poor signal integrity. For instance, low-complexity waveform detectors present rather high error rates since they present high sensitivity to noise. Noise has the adverse effects of boosting the data rate with false positives. Thus, high-fidelity compression techniques are needed in multi-channel recording implants to perform data compression while improving data integrity for yielding faithful results in neuroscience applications.

The Wavelet Transform (WT) is an attractive approach to overcoming these limitations. WT processing is a very effective tool for time-frequency analysis, non-stationary signal analysis and fast transients analysis [18]. It has been extensively used for various signal processing tasks such as generalized filtering, data compression, signal detection, feature extraction, sorting, and signal to noise ratio enhancement. Moreover, it offers substantial data compression factors and superior waveform integrity [19].

Designing a hardware WT processor dedicated to implantable devices presents very difficult challenges related to power consumption and chip area as both have to be minimized tremendously. A WT processor can either be implemented with digital building blocks or by means of analog circuits. A digital implementation realizes a discrete wavelet transform (DWT), whereas an analog one realizes a continuous-time wavelet transform (CWT). Although the digital form is often preferred for its simplicity of implementation, its analog counterpart is not to be neglected for its numerous advantages. The digital form is assumed to give superior precision and better reliability, whereas analog implementation is chosen for its reduced size and lower power consumption. However, there are several possible realizations of a DWT or a CWT processor, each presenting a different trade-off.

This paper describes the requirements, issues and solutions in the design of DWT and CWT processors dedicated to implantable neural recording devices. In the next section, an overview of neural signals and their compression using WT is presented. Section 3 provides a comparison study of different DWT processors and presents an efficient architecture for our application. Section 4 describes the issues and solutions for designing a filter bank devoted to a CWT processor. Section 5 gives the conclusions on the presented work.

## 2. Neural Signal Compression with WT

The principal mechanism of information transmission in the cortex is the generation of *biopotentials waveforms* (also called *neural action potentials* (APs), or "*spikes*" for short) by the neurons. The rate of generation of these action potentials conveys information to neighboring neurons. Spikes are short-lived waveforms of a few millisecond duration whose rate of occurrence ranges between 10 to a few hundred per second. Such neural waveforms originate from intracellular depolarization of neurons membranes. As shown in Figure 1, the depolarization can be divided into two periods. During the first, the electric potential experiences a sharp reversal reaching in approximately 1 ms to a potential value close to 1V. In the second period, it falls down to below its resting potential. During this period, the neuron cannot issue another potential, hence the name of "refractory period." The peak to peak amplitude of the intracellular AP is from 90 to 100 mV, duration of 2 to 3 ms and frequency content between 100 Hz and 10 kHz. The extracellular AP is similar to an echo of the intracellular potentials, but presents different shapes from a neuron to another, and has smaller voltage range (100 µV to 1 mV).

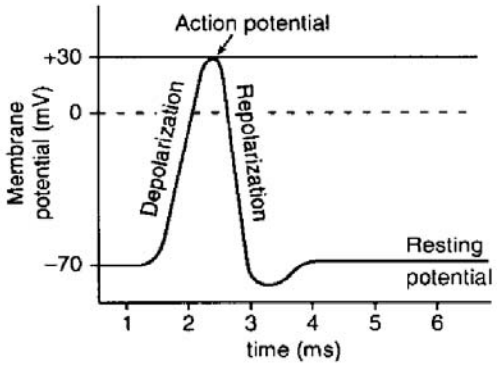



**Figure 1.** A typical intracellular action potential.

Data compression is used to decrease the volume of data that must be transmitted by low-power telemetry circuits in neural implants. Such procedure must extract relevant contents from the raw signal by means of real-time waveform detection and rejecting non relevant parts of the signal. This scheme is highly advantageous in multi-channel neural recording devices because of the discontinuous nature of the neural signal and its low duty cycle. Spike detection prior to data compression allows for omitting those parts of raw signals that bear no

information content. Figure 2 illustrates a simple spike compression scheme where only the time of occurrence of the detected spikes are captured.

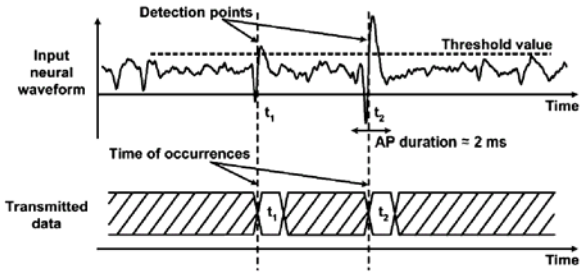



**Figure 2.** A simple neural spikes compression scheme in which the time of occurrence of each spike is detected and sent to the remote host.

An efficient compression procedure must be joined with an efficient spike detection algorithm to allow for a faithful reconstruction of neural signals based on partial representation of the biopotentials. The WT provides the required accuracy for improving the detection rate and the data compression factor in neural recording implants [20].

In a simple compression scheme, the WT produces *approximation* and *detail* coefficients that give a sparse representation of an original signal. The coefficients whose values are below a specified threshold are ignored, reducing the number of bits to be transmitted to a remote base station. The remaining non-zero WT coefficients give an approximate representation of the original signal with a reduced volume of data. Most of the spectral power of neural spikes is located in the lower frequency bands, thus WT coefficients in higher bands can be removed as they contribute insignificantly to the reconstruction. The quadratic error increases in the reconstructed signal as we eliminate more coefficients. WT coefficient thresholding also serves for denoising as high frequency WT coefficients are removed and zeroed before signal reconstruction. Besides, the coefficients in higher frequency bands can be used to characterize the noise in the channel and to calculate the detection thresholds that yield the best performances.

Similar to other circuits and systems dedicated to implantable devices, the design of a spike compressor presents major challenges in term of power consumption and chip area. However, WT-based compression is perhaps more demanding than a low-complexity scheme. Therefore, careful considerations must be made in the design phase. A trade-off exists among accuracy, circuit complexity,

chip area, and power consumption subject to the given requirements. For instance, the choice of a working frequency along with the targeted frequency bands will have an impact on the implementation. Since most of the neural signal energy resides between 100 Hz and 6 kHz, this requires overcoming some challenges in integrated circuit design. Also, the choice of a suitable wavelet basis is critical to reaching the required compression ratio. In particular, the chosen wavelet basis must resemble as much as possible the targeted waveforms in order to produce low error rate [17].

The discrete wavelet transforms lend themselves more easily to hardware implementation than continuous wavelet transforms because digital circuit design techniques have reached a high level of automation and reliability through the use of hardware description languages and powerful hardware simulators and synthesizers. However, analog implementations of CWTs are usually preferred where ultra low-power operation is the primary concern.

Several architectures for hardware implementation of DWT have been proposed in the past [18]. Most of the published results have focused on improving the processing speed in computation-intensive applications such as image processing and pattern recognition. Similarly, analog implementations of CWT can benefit from several circuit blocks including filters, amplifiers, or mixers, operated either in the voltage mode or in the current mode. In the next sections, we will elaborate on digital and analog WT processor architectures that were designed for data compression in low-power neural recording implants.

## 3. Implementation of Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT), based on the idea of sub-band coding, yields a fast computation of WT which is also easy to implement in software and hardware. In CWT, the signals are analyzed using a set of basis functions which relate to each other by simple scaling and translation. In the case of DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales.

The power consumption and silicon area of a digital processor are related to several factors such as the complexity of the implemented algorithm (i.e., number of processing elements and arrangement of the building blocks), the working frequency, the routing complexity, the sensitivity to quantization, and the finite word

length. As a result, the usual metrics including the number of processing elements (multipliers and adders), and the operating frequency gives only a rough approximation of the real power consumption and the area of a chosen architectures [18]. Consequently, an implementation based comparative study is proposed in order to find the best suited DWT architecture for this implantable application.

Among the well-known wavelet bases, the Daubechie basis is the most suited to neural signal processing, as it is appropriate for monitoring discontinuities (i.e., the spikes) in an input signal. Based on our previous experimental work with real neural signals recorded for the visual cortex of rats [12], we have chosen the fourth order Daubechie wavelet basis (*db*4). The *db*8 basis can yield less reconstruction error but requires more computational resources.

The DWT has numerous possible implementation forms [18]. Mallat [21] showed that the approximation coefficients of a DWT can recursively be processed by a two-channel filter bank in order to perform a multilevel wavelet decomposition. However, a straightforward circuit implementation of this scheme leads to tremendous inefficiency (low hardware utilization). Indeed, if the first decomposition level is clocked at frequency *f*, the decomposition level *j* would be clocked at frequency $f/2^{j-1}$, wasting a considerable amount of clock cycles in the preceding stages. Careful consideration of this structure has led to the design of efficient DWT architectures.

Two principal types of architectures enable to compute the DWT:

1. convolution-based structure

2. lifting-based structure

The first approach uses a two-channel filter bank in its *direct* form (Figure 3) or in its *polyphase* form. Most convolution-based architectures aim at minimizing the number of required processing elements (mostly multipliers and adders). However, the required control schemes, and the routing complexity are often significant. A careful balance of parallelism and folding can be used to optimize such architecture [22]. A parallel multilevel DWT architecture, in which a progressive folding optimizes each level, can achieve a hardware utilization of 100%.

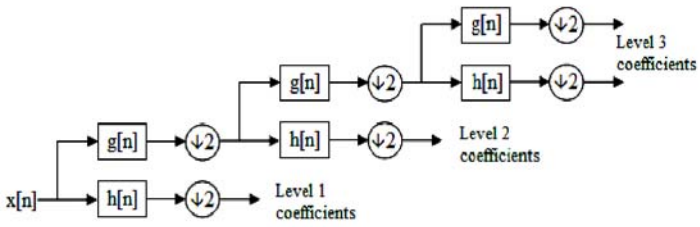



**Figure 3.** Direct form convolution-based structure.

The second approach, the lifting scheme, is based on the factorization of the two-channel filter bank equations and gives a set of recursive equations that define a ladder structure, illustrated in Figure 4. Highly folded architectures based on this structure can lead to up to 100% hardware utilization. For example, a highly folded 1-D lifting-based DWT can be implemented taking advantage on the specific symmetries in the 5/3 and 9/7 Daubechies wavelet bases [23]. This approach can reach 100% hardware utilization, but is not applicable to all wavelet bases. Besides, other types of implementation schemes such as systolic array-based architectures have been proposed [24]. A general method to realize folded lifting-based DWT is available to implement most wavelets bases [25]. This approach is systematic and offers decent hardware utilization (90% for 5 levels). In general, the lifting scheme can decrease the DWT computation complexity by up to 50%.

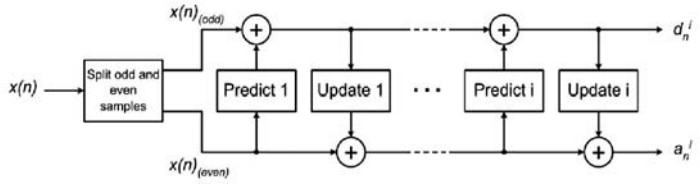



**Figure 4.** Block diagram of a two-channel filter bank using the lifting scheme.

At first glance, the lifting-based DWT may appear more appealing than the convolution-based DWT for low-power and small-chip area implementations. But, other important parameters, such as the computation complexity and precision have to be considered to find the best-suited architecture. A trade-off between these two issues must be made. For instance, fixed-point representations are usually less area and power consuming than floating-point ones. Simulink® models of generic polyphase and lifting structures can be implemented to assess the effect of finite precision on an implementation. The literature usually refers to input scaling, filter

quantization, and internal signals round-off as the principal causes of output signal distortion. The effect of each one of these parameters can successively be studied with the models. To quantify the distortion, an input signal is processed with the models, and reconstructed with an ideal inverse DWT. The distortion is assessed using to common indicators: the root mean square error (RMSE), and the signal to noise ratio (SNR). The RMSE is

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x(n) - r(n))^2}, \quad (1)$$

where $x(n)$ denotes the original signal, $r(n)$ the reconstructed signal, whereas the SNR is

$$SNR = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right), \quad (2)$$

where $\sigma_x^2$ denotes the variance of the original signal, and $\sigma_e^2$ the variance of the error. Table 1 summarizes the performances of such models for a fixed-point implementation. We can see that the polyphase implementation performs slightly better than the lifting one. The lifting scheme requires 9 bits for its coefficients representation, compared to 8 bits for the polyphase scheme.

**Table 1.** Parameters and performance of the DWT fixed-point models [18]

| Algorithm / Parameter | Polyphase | | | Lifting | | |
|---|---|---|---|---|---|---|
| | **Setting** | **RMSE** | **SNR** | **Setting** | **RMSE** | **SNR** |
| Coef. quantization | (8, 7)[a] | 2.28 | 52.7 | (9, 6) | 5.75 | 48.2 |
| Internal round-off | (*X*[b], 7) | | | (*X*, 2) | | |
| Output scaling | (8, 0) | | | (8, 0) | | |

a. $[a, b]$ is such as a indicates the total number of bits and *b* the binary point

b. *X* indicates that the integer part is chosen in the architecture to avoid overflow

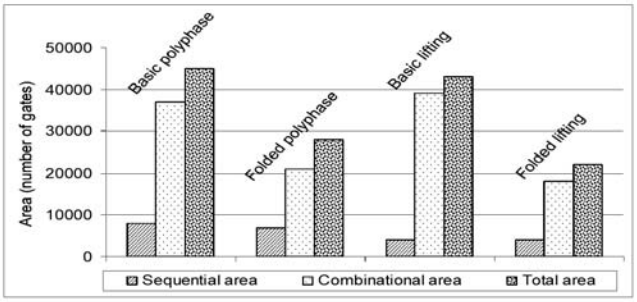



**Figure 5.** Estimated silicon area for different DWT structures [18].

The chip-area and the power consumption of the polyphase and the lifting DWT structures can be compared. Four different DWT processors were implemented in a TSMC CMOS 0.18 μm process and simulated at the gate level with the Synopsys power compiler®. The implemented processors include two fully parallel structures, and two folded structures (one parallel and one folded implementation for each type), all realizing 3-levels Daubechies-4 DWTs. The fully parallel architectures implement the basic structures depicted in Figure 3 and in Figure 4. The two folded architectures improve the parallel ones by avoiding the inefficient down sampling task. The polyphase folded processor is based on [22] and the lifting folded one is based on [25]. These processors present hardware utilizations of 100% and 87.5%, respectively. The four DWT processors were synthesized with Synopsys using the Artisan CMOS 0.18 μm technology library, provided by CMC Microsystems (Kingston, ON, Canada). The number of gates yielded by each implementation is reported in Figure 5 for comparison. As shown, despite the lower computational complexity of the lifting-based architecture, both processors yield similar gate area. It ranges from 45,000 gates for the parallel architectures to 20,000 gates for the folded ones. This can be explained by the trade-offs for a reduced hardware complexity versus good precision. A slightly higher number of bits for the lifting scheme implies more processing elements and larger internal bus width for the lifting architectures.

Several design parameters must be taken into account for an accurate evaluation of the static and dynamic power consumption in each processor: the supply voltage, the transistors characteristics (threshold voltage, mobility, etc.), gate loads and parasitic elements, the propagation delays, the operating frequency, and the switching rate at the different nodes. A gate-level simulation can capture the main design features with an accuracy ranging from 10% to 25% [26].

Since, the power consumption of a digital processor is closely related to the processed data, real neural spike recorded from a past experiment were used for this evaluation. A bank of 20 test neural spikes was constructed from previously recorded waveforms obtained from the visual cortex of rats. Spikes were isolated and randomly distributed along the time axis according to a Poisson distribution, with an average rate of 60 spikes per second.

A background neural noise was generated and adjusted to yield a SNR of 6 dB for each signal. Waveforms were of 1-second duration, and represented on 8 bits. The power consumptions yielded by these 20 tests signals were averaged for each DWT processor. Figure 6 shows the power consumption of the four architectures. It is shown that the lifting processors consume more power than the polyphase implementations. The static power consumption is almost equivalent, because of their similar gate area. But, the dynamic power consumption is higher for the lifting scheme. This high dynamic power consumption may result from the fact that both liftingbased processors are not pipelined, which implies long critical paths. These are known to be a major source of glitches. These unwanted transitions of logic states can account for 20% to 40% of the dynamic power dissipation [26]. Nevertheless, pipelining can considerably increase the size of a lifting-based DWT processor. Therefore, these results suggest that the polyphase scheme is better-suited to implantable biomedical devices.

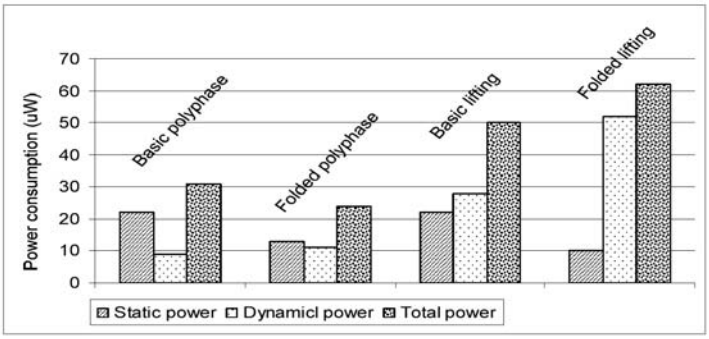



**Figure 6.** Power estimation for the six DWT processors [18].

### 3.1. Processor architecture

An implantable solution must minimize the power consumption and circuit size by means of a dedicated architecture and simplified building blocks. A 32-channels DWT processor based on a polyphase structure was designed and implemented in

VHDL and prototyped using a Xilinx FPGA platform. The prototyping platform features a Virtex 2 (XCV2000E) FPGA and communicates with a remote software driver programming through a serial link. A block diagram of the implemented WT processor is depicted in Figure 7. The DWT processor uses multiplexing of four channels toward each DSP module in order to reduce their number and increasing the overall data processing efficiency. The processor extracts the neural signal recorded by an implant, and transmits it serially to a remote station for reconstruction and further processing.

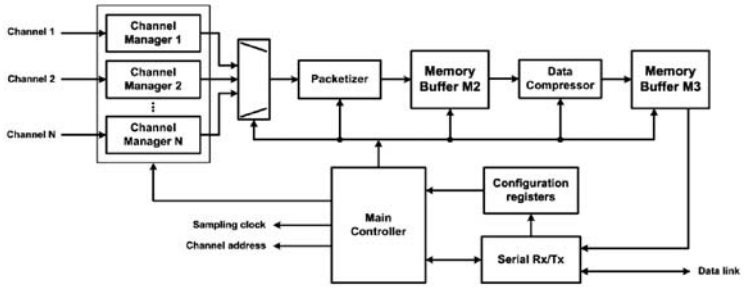



**Figure 7.** Block diagram of the DWT processor.

A dynamically reconfigurable custom controller (Figure 7) is used for data management in the DSP core. This controller enables bidirectional data communications between the implant and an external host and manages the data path from the detector output to the data packetization module [20]. It includes one Channel Manager (CM) per channel (Figure 8), a packetizer module for the whole system, Random Access Memory (RAM) blocks, configuration registers, and a packet dispatcher including a transceiver for serial data transfer. A packet based protocol allows bidirectional communications between the implemented transceiver and the remote host trough a serial data link. The processor includes 32 CMs, each sharing 8 wavelet DSP cores. DSP cores enable feature extraction for spike detection, compression of the data, and denoising of neural input signals.

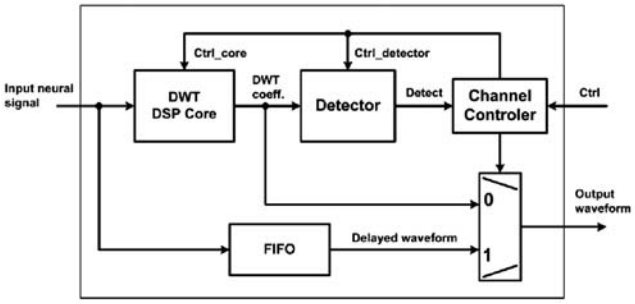



**Figure 8.** Block diagram of a Channel Manager.

The DSP processor performs a Daubechie-4 wavelet with three levels of decomposition on a polyphase structure, giving 4 output frequency bands. The coefficients of each output bands are compared with a detection threshold. The noise level is evaluated with the output coefficients of the higher frequency bands, so the detection threshold can be set above this noise level. Figure 9 is showing WT denoising of a pre-recorded neural signal for SNRs of 10 dB, 5 dB, and 1 dB respectively, for a compression rate of 5%. It is seen that the reconstructed spike is degraded at very low SNRs.

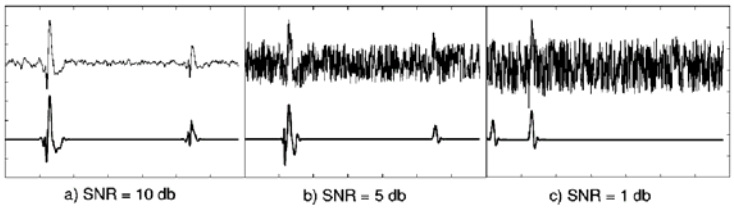



**Figure 9.** Neural spike denoising at different SNR levels.

The controller module can be configured through a serial bus. It allows for individual channel configuration by writing to corresponding registers upon reception of a configuration command along with the channel address. It also includes a 32 bytes FIFO implementing a time-delay which is required to align the detected spikes and allows for a better waveform reconstitution. The data processing delay is removed thanks to this delay line. This data memory buffer allows triggering in the middle of a detected spike window, which leads to superior alignment and accuracy in downstream signal processing tasks. The RAM memory M1 included in each CM (Figure 8) is required for holding the detected waveform before sending it serially. The memories M2 and M3 serve as temporary buffers in the data compression and data transfer processes. The communication module manages the

bidirectional data link interface to access the registers and implement the communication protocol. The critical path has been optimized to avoid undetected spikes. The packet forming module can handle every detected waveforms occurring at least a refractory period after an event. Otherwise the detected event is considered as noise and is rejected.

Each CM includes a FIFO that implements data buffering in the detection process. The packetizer adds a header to the detected spikes samples. This header is composed of a channel identification number, a timestamp and control tags. The 1-kbyte memory M2, fed by the packetizer, implements a data buffer between the packetizer and the packet dispatcher. To overcome limitations due to burst-firing neurons and simultaneous channel firing, the critical path must be optimized to insure that each detected spike can be processed within the refractory period of a neuron (around 5 ms to 10 ms). The processing delay in the controller is expressed as follows:

$$Delay = \left[ M - \frac{M}{N_S} \right] \left[ \frac{N_S + H}{B} \right] < T_R, \tag{3}$$

where $N$ is the number of channels, $M$ is the memory buffer size, NS is the number of raw data samples per packet, $B$ is the data link bandwidth, $H$ is the header size and $T_R$ is the refractory period of a neuron. The time required to process all channels is bounded by the number of channels and by the available bandwidth on the data link. Figure 10 illustrates the worst case delay in the controller with respect to the number of channels included and the available bandwidth, for a buffer size of $M = 1024$ bytes, a packet size of $N_S = 128$ samples, and a header of $H = 4$ bytes.

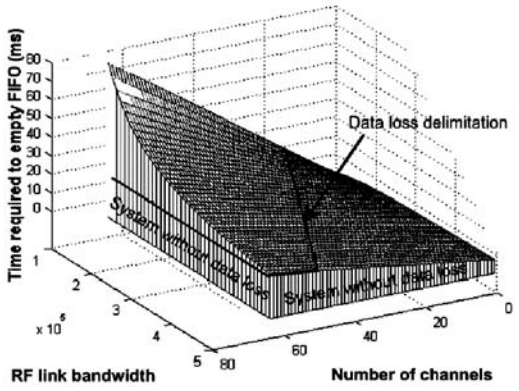



**Figure 10.** Delay in controller with respect to number of channels and RF link bandwidth [17].

A processing delay higher than the refractory period may result in spike loss. Figure 10 shows that a system without spike loss is obtained for the portion of the curve that is below the solid delimitation. According to (1), the presented 32-channel processor requires a data link with data rate of at least 634 kbit/s to avoid spike loss, considering a typical refractory period of 5 ms.

Data compression is performed in order to minimize the output data rate of the WT processor. The data compression module selects coefficients whose magnitude that are greater than a pre-selected threshold and forwards them to the memory block M3. These data are then packetized and sent to an external receiver. The custom reconfigurable controller manages the register accesses and interrupts, and allows remote configuration of each CM through a serial bus master. Such design is highly suitable for channel expansion in future versions. The same bidirectional data link enables dynamic reconfiguration of the device controller and transmission of the data through the same port. This port is designed as half duplex with a higher priority given to register accesses over data. Incoming control packets need negligible processing time compared to a data transmission procedure. However, the WT processor cannot transfer data at full speed when device reconfiguration is performed because some overhead is required for control accesses. The average data rate and the maximum number of channels simultaneously open depend on the average neural activity (spike occurrences rate) in the recorded data. Direct register accesses to configuration memory allows for individual channel configuration, whereas indirect accesses enable to modify all the channels simultaneously.

A multithreading software application running on a host PC sends commands to the WT processor and receives data while it manages the communication protocol. The host application initiates a connection with the controller, and then performs an initial device configuration. A first raw data recording phase must be made in order to measure the characteristics of the noise on each channel and to adjust the WT thresholds. The thresholds are configured by writing back the configuration to the appropriate register. Afterwards, the WT processor can perform neural processing with WT detection and compression enabled at maximal bandwidth utilization.

### 3.2. Implementation results

Table 2 presents the resources utilized for the FPGA prototype implementation. The CMs are by far the type of building blocks that require the largest amount of logical modules. This is because they include the DSPs which require multipliers and adders, consuming a large amount of resources. Moreover, each CM also

includes one 32-bytes FIFO per channel. Another important aspect to note from Table 2 is the number of RAM blocks used by the FPGA. Even if resource utilization reports 36 RAM blocks for this implementation, they are highly underused, and are available for system expansion towards more processing channels using the same prototyping platform. The bottleneck would arise at the logic modules in the WT processor.

**Table 2.** Resources usage for 32 channels implemented in a Xilinx XCV2000E FPGA [20]

| Logic Modules | | | RAM Memory | | |
|---|---|---|---|---|---|
| Building Block | REGs | LUTs | Module | Bytes | 4-Bytes Blocks |
| Channel Manager | ≈0% | ≈0% | | | |
| Detection/Compressor | ≈0% | ≈0% | M1 | 64 | 32 |
| Packetization | 1% | 1% | M2 | 1 k | 2 |
| Transceiver | 36% | 10% | M3 | 1 k | 2 |
| Total | 71% | 23% | Total | 4096 | 36/160 |

We used pre-recorded neural signals to test the DWT processor prototype. The processor output produces raw DWT coefficients which allow recovering the raw input signal with a smaller quantity of data. The signals shown in Figure 11 and Figure 12 were obtained by keeping only eight WT coefficients from the transformed original waveform, buffered with a window size of 64 samples. This gives a raw compression rate of 8 for this window size at a sampling rate of 32 ks/s. Figure 11 shows a spike that is well detected by the DWT processor, whereas Figure 12 depicts a detected spike that is poorly centered within the detection window. The thresholds must be well adjusted in the detection module to yield less false positive events as possible and to minimize the number of lost spikes. If the detection module is too permissive, the implant would require more bandwidth, but would also presents better signal integrity at reconstruction. The threshold must be set according to the available bandwidth at the communication link. The thresholds are usually adjusted within 3 to 5 standard deviations of the input signals. However, for optimal results, fine tuning must often be performed online to adjust the detection thresholds according to a targeted neural signal integrity and bandwidth utilization.

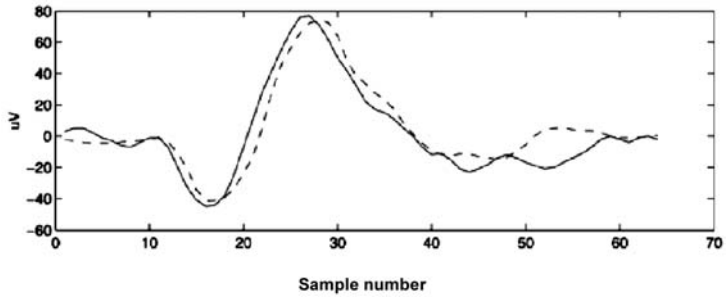



**Figure 11.** A 64-sample signal. Solid line is original signal. Dashed line is reconstructed from 8 coefficients. The reconstructed signal is acceptable.

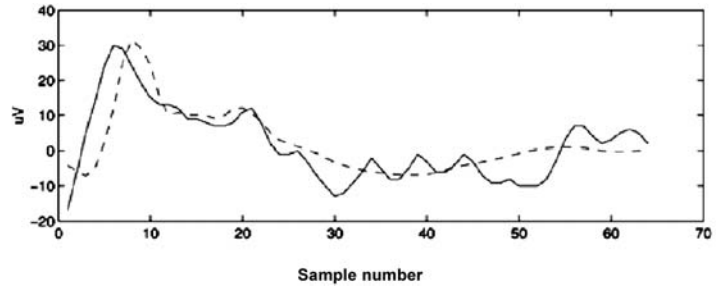



**Figure 12.** A 64-sample signal. Solid line is original signal. Dashed line is reconstructed from 8 coefficients. The reconstructed signal is unacceptable.

### 4. Implementation of the Continuous Wavelet Transform

A general CWT is described by the following equation

$$X(a,\ b) = \frac{1}{\sqrt{|\ a\ |}} \int x(t)\psi\left(\frac{t-b}{a}\right)dt, \tag{4}$$

where $x(t)$ is the signal to be analyzed, $\psi(t)$ is called the mother wavelet or the basis function, $b$ is the translation parameter and $a$ is the scale parameter. All the wavelet functions used in the transformation are derived from the mother wavelet by translating and scaling operations. A mother wavelet must satisfy two conditions: (i) zero mean condition, (ii) admissibility condition. The definition of the former condition is selfevident, and the latter is described by the following equation:

$$\int_{-\infty}^{\infty} \frac{|\ \Psi(\omega)\ |_t^2}{|\ \omega\ |} d\omega < \infty, \tag{5}$$

where $\Psi(\omega)$ is the Fourier transform of the mother wavelet. The admissibility condition guarantees that the original signal can be reconstructed by the inverse wavelet transform. It also implies that the Fourier transform of the wavelet must have a zero component at zero frequency. Hence, the wavelets are inherently band-pass filters in the Fourier domain, defined as wavelet filters. Any function that has finite energy is square integrable and satisfies the wavelet admissibility condition can be a wavelet.

The wavelet analysis has become a very useful tool in neural spike processing, as it can nicely separates low frequency spikes from a high frequency background noise. Although, the wavelet transforms are most often implemented in digital forms (software or hardware), an analog implementation may be an attractive design choice when low-power operation is of primary concern. Approaches for implementing the WT within analog circuits have been proposed in the literature [27], [28], and [29]. A very effective method consists of implementing an approximated wavelet basis with the impulse response of a linear filter [28]. This approach allows implementing a wavelet processor with very low-power and small chip area by means of low-voltage analog filters.

An analog WT implementation begins by choosing a suitable wavelet basis. The chosen basis must present low complexity and be easily approximated by a fraction of low order polynomials. We chose a bi-orthogonal Gaussian wavelet basis because it has a high degree of similarity with bi-phasic extracellular spike signals, and because such basis provides a very good time-frequency resolution. Moreover, the Gaussian basis can easily be implemented with analog filters since band-pass filters naturally exhibit a Gaussian-like response [28]. By taking such an approach, the analog circuits complexity will depends on the accuracy of the wavelet transform approximation. We discuss the translation of the chosen wavelet basis into a rational transfer function in the next section.

### 4.1. Wavelet basis approximation

To implement the wavelet filter, one must first determine a linear differential equation whose impulse response resembles closely the intended mother wavelet (in our case, a Gaussian mother wavelet). However, starting from a given impulse response $\psi(t)$ does not necessarily lead to a physically realizable and stable system. Therefore, one is forced to start by transforming the mother wavelet $\psi(t)$ to its Laplace-domain representation, $\Psi(s)$, and then using an approximation method that

yields a low-order rational function $H(s)$ for serving as the filter's frequency response. Two methods have been used in the literature: (1) Padé approximation [30] and 2) L-2 approximation [31]. Each of these methods has its own advantages and disadvantages. The former is computationally fast but produces less accurate results than the latter [31]. On the other hand, the quality of L-2 approximation significantly depends on the starting point of its optimization algorithm and local optima can prevent the algorithm from convergence to the global optimum. Besides, the Padé approximation may produce an unstable system depending on its expansion point $s_0$.

For its ease of use, we employed Padé approximation to directly produce a low order rational polynomial function. The procedure is as follows. Suppose a rational function $H(s)$ is desired that has a numerator of order *m* and *a* denominator of order *n*. The Padé method allows one to approximate the original function $\Psi(s)$ up to order $m + n + 1$. Suppose that we have the Taylor series expansion of $\Psi(s)$ around some point, e.g., $s = 0$, as follows:

$$\Psi(s) = c_0 + c_1 s + c_2 s^2 + \cdots + c_k s^k + O(s^{k+1}), \tag{6}$$

where $k = m + 1 + n$. Let $\tilde{\Psi}(s)$ be the truncated Taylor series given by the first $k + 1$ terms of Equation (6). The Padé approximation of $\Psi(s)$ is obtained by setting $\tilde{\Psi}(s)$ equal to a rational function $H(s)$

$$\tilde{\Psi}(s) = H(s) = \frac{P(s)}{Q(S)} = \frac{p_0 + p_1 s + \cdots + p_m s^m}{q_0 + q_1 s + \cdots + q_n s^n}. \tag{7}$$

The coefficients $p_i$ and $q_i$ are now to be determined. This is done by solving a set of linear equations which result from equating the corresponding coefficients on both sides of the following equation: $\tilde{\Psi}(s).\ Q(s) = P(s)$. The method is detailed in [30] and [28].

The best approximation we found using a Padé approximation is a 6th order rational function (Equation (8)) of the mother wavelet shown in Figure 13. Mother wavelet to be approximated, and it meets the required admissibility condition (Equation (5)) for a wavelet transform. The time and frequency responses of the approximation are presented in Figure 14. (a) Frequency response and (b) impulse

response of

$$H(s) = \frac{0.4786s^3}{s^6 + 2.058s^5 + 3.139s^4 + 2.537s^3 + 1.569s^2 + 0.5145s + 0.125}. \quad (8)$$

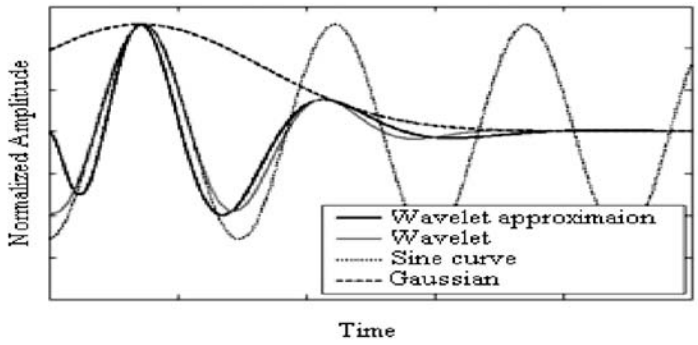



**Figure 13.** Mother wavelet to be approximated.

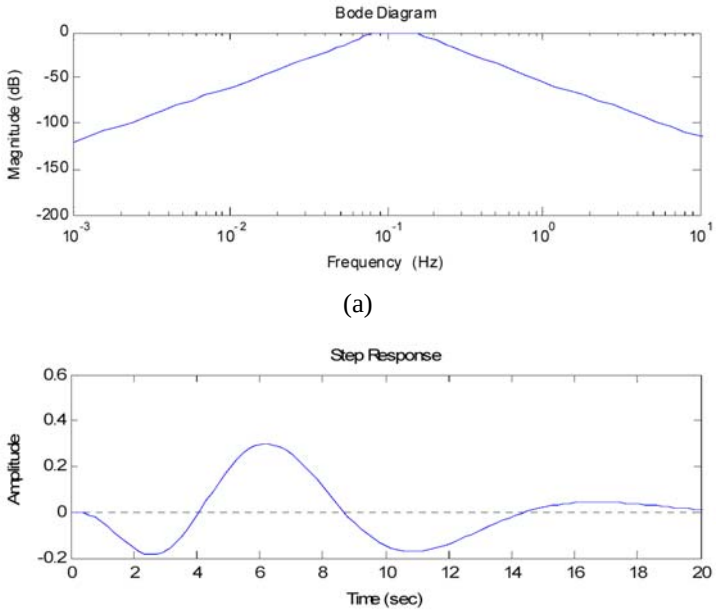



**Figure 14.** (a) Frequency response and (b) impulse response of $H(s)$.

The magnitude of $\Delta\Omega\Delta t$ of the approximating function is 0.41113 while that of the approximated function is 0.42259; this is only 2.8% higher, which is an

acceptable accuracy. The realization of the transfer function is one of the difficult steps because one can easily simulate it in software, but it is not easy to implement as an analog circuit. We have approximated the CWT using a bank of band-pass filters which covers the frequency range that spans a typical spike waveform. The architecture of the filter bank is discussed in the next section.

**4.2. Wavelet filter bank design**

We implemented the approximated Gaussian wavelet basis with six *Q*-constant 6th order band-pass filters whose respective center frequencies are logarithmically spaced to cover most of the biopotentials frequency range (100 Hz to 7 kHz). This filter bank decomposes the input signal into a set of frequency bands by passing it through the six band-pass filters. The transfer function of each band-pass filter implements a 6th order Gaussian wavelet basis generated in the previous section. The band-pass filters in the filter bank have their central frequencies located at $f_{0\psi}/a$ and their bandwidth equal to $\Delta\Omega/a$, where *a* is a variable scale factor, $f_{0\psi}$ is the central frequency, and $\Delta\Omega$ is the bandwidth of the mother wavelet $\psi$, respectively. The quality factor *Q* of each filter is given by

$$Q = \frac{f_0}{BW} = \frac{f_{0\psi}}{W\Omega}. \tag{9}$$

Each 6th order band-pass filter is composed of three 2nd order band-pass filters. A straightforward implementation would require eighteen 2nd order filters. But as depicted in Figure 15. Band-pass filter bank we have employed a parallel topology that shares 2nd order filters between each 6th order filters in order to minimize circuit area and power consumption. The 6th order filters are now organized into thirteen interleaving 2nd order band-pass filters. The resulting transfer function approximates the chosen Gaussian wavelet basis. The six outputs of the filter bank are each related to a specific frequency band as well to a corresponding center frequency. A threshold function can then be applied to one or several combined outputs of the wavelet filter bank to efficiently detect spike waveforms.

Given that the difference between filter components of a CWT processor is based only on factor *a* and does not depend on *Q*, therefore, the bank of filters to be designed is constant-*Q* filters [22]. The cutoff frequency of filter *i*, a band-pass filter centered at $f_{0i}$ with a band with of $BW_i$ is given by Equation (10). In order for the sum of all filters to cover the full desired bandwidth, we must have the upper cutoff

frequency of filter $i(f_{H_i})$ to be equal to the lower cutoff frequency of filter $i+1(f_{L_i})$,

$$f_{H_i} = \frac{f_{0i}}{2Q}(\sqrt{4Q^2+1}+1) = f_{L_{i+1}} = \frac{f_{0i+1}}{2Q}(\sqrt{4Q^2+1}+1). \quad (10)$$

The filter described by $H(s)$ in Equation (8) has a quality factor of 1.47. Therefore, the ratio between the center frequencies of two adjacent filters is given by

$$\frac{f_{0i}}{f_{0i+1}} = \frac{\sqrt{4Q^2+1}-1}{\sqrt{4Q^2+1}+1} = 0.51. \quad (11)$$

Hence, each filter has a frequency about two times greater than the previous one, as shown in Table 3. Therefore, six filters are sufficient to cover the band of interest.

**Table 3.** Filter bank characteristics

| *Filter* | *Center Frequency* (Hz) | *Lower Cut-off Frequency* (Hz) | *Higher Cut-off Frequency* (Hz) |
|---|---|---|---|
| F1 | 5000 | 3581 | 6980 |
| F2 | 2500 | 1791 | 3490 |
| F3 | 1250 | 895 | 1745 |
| F4 | 625 | 448 | 873 |
| F5 | 313 | 224 | 437 |
| F6 | 156 | 112 | 218 |

To reduce the implementation area, we have used transitional filter techniques [19] with transfer function manipulations which permits sharing of the 2nd order filters between each adjacent 6th order filter. As seen from Figure 15, each 6th order filter is composed of three 2nd order filters in parallel. Thus, we have covered the entire frequency range of a neural action potential using 13 filters only. Figure 16 shows the frequency response and time response of the filter bank. As seen in the Bode diagram, the pass band of the filter starts at 100 Hz and ends around 7 kHz.

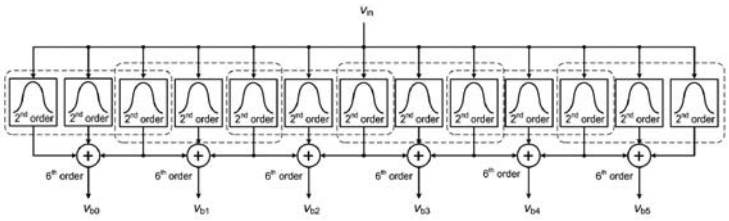
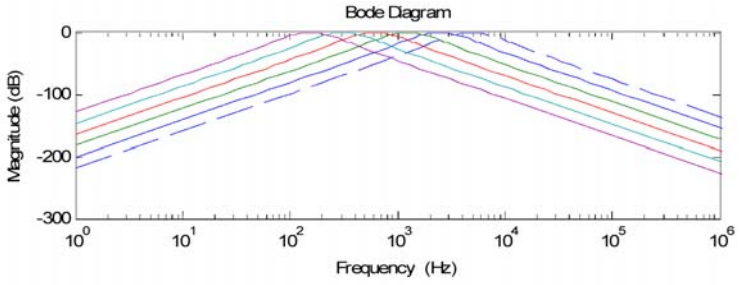
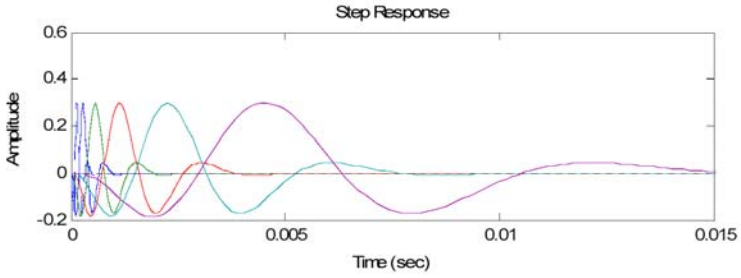



**Figure 15.** Band-pass filter bank.




(a)




(b)

**Figure 16.** (a) Frequency and (b) time response of designed filter bank.

### 4.3. Circuit implementation

In general, a circuit that implements a given transfer function has many possible state space descriptions and many different realizations. Therefore a designer has the freedom to find a circuit that fits his requirements best. Among the low-power and small chip area requirements, a filter bank for WT should have large dynamic range and high linearity. Also, because we adopted a synthesis technique that exclusively uses integrators, special techniques need to be employed to achieve very long time constants in integrators operating in the very low frequency range of neural action potentials [29].

There is a multitude of methods for implementing analog filters. MOS-C filters are based on standard active circuits whose resistances have been replaced by MOS transistors operating in triode. Gm-C filters operates in current-mode and use transconductors and capacitor as integrators, however, they show low linearity and high sensitivity to process variations [32]. Switched capacitor filters (SC) sample a continuous signal using switches controlling the flow of charges. But, sampling makes these circuits prone to charge injection and clock feed-through [32]. Besides, log-domain filters [33] present high dynamic range and consume very little energy, especially when implemented in CMOS technology. Therefore, we chose a log-domain circuit implementation for this wavelet filter bank. The filtering process takes place in the logarithmic domain which is highly nonlinear. CMOS log-domain filters exploit the exponential characteristic of MOSFET devices biased in the subthreshold regime. They inherently exhibit high dynamic range because they do not require any linearization, as other analog filters do. Log-domain filters use current-mode input signals that are internally compressed as a function of voltages in the filter integrator node, as illustrated in Figure 17.

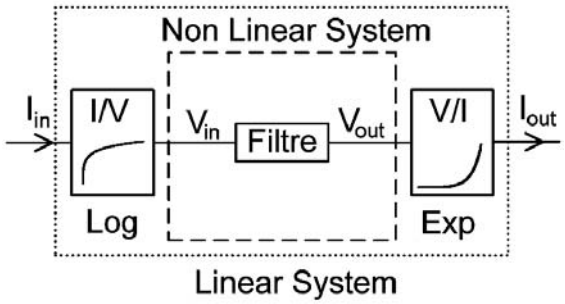



**Figure 17.** General structure of a log-domain filter.

The usage of CMOS transistors biased in weak inversion allows achieving the very low-power consumption needed for an implantable device. Log-domain filters logarithmically compress an input current into internal voltages that are processed before being expanded back; therefore they are well suited to low-voltage circuits. A straightforward implementation of the state space model derived from the transfer function results in circuit realizations that are often impossible to realize because of the wide range of bias current and capacitance size. We chose the set of filter coefficients so as to make sure that the biasing currents are either equal or at least in the same range. This enables sharing of biasing circuits among elements.

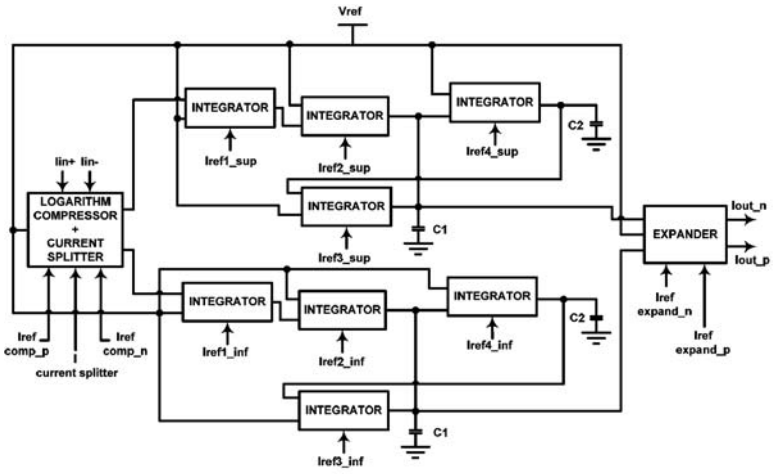



**Figure 18.** Schematic of the implemented 2nd order log-domain filter.

Figure 18 shows the schematic of our 2nd order class AB log-domain band-pass filter obtained from our state space models. It is composed of eight integrators along with a logarithmic compressor and an exponential expander. We have integrated the current splitter to the logarithmic compressor to make the schematic more comprehensible. The presence of the compressor and expander allows the circuit to work in the log-domain mode at very low voltage supply. The inputs Iin+and Iin- correspond respectively to the *p*-type input and *n*-type input of our filter. The Vref potential along with the currents Iref i_sup/inf, Iref comp_p/n and Iref expand_p/n are employed to bias the circuit. In addition, using the currents Iref i_sup/inf, and capacitors C1 and C2, it is possible to precisely set the central frequency of the filters.

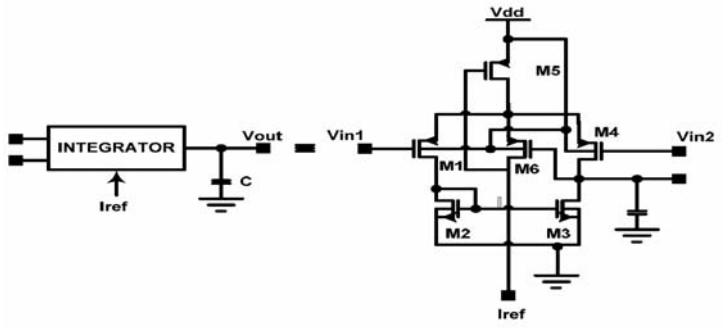



**Figure 19.** Schematic of the implemented integrator circuit.

Figure 19 shows the configuration of our integrator which is similar to a $G_m - C$ integrator made from a differential pair [34]. The difference is that the current in transistor M5 is not constant but increases with the output voltage. Given the level of compatibility between outputs and inputs, this kind of log-domain integrator can be easily cascaded in order to make high-order transfer functions [34]. This property is used for our 6th order filters. In addition, this integrator has low input noise levels thanks to the filter configuration. The noise current of M6 and M5 is equally separated through M1 and M4 before being inverted by the current mirror and added at the capacitor node. Assuming perfect matching between those currents, the noise is cancelled. As a result, for small signals, the noise of the current mirror is insignificant, and only the noise of the two inputs transistors must be taken into consideration [34].

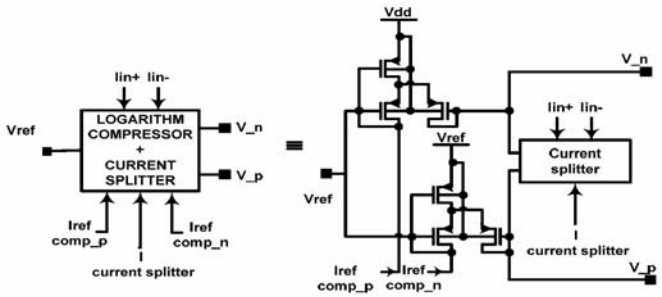



**Figure 20.** Schematic of the implemented logarithmic compressor circuit.

Figure 20 shows our class AB log-domain compressor [34]. It compresses the input current and transforms it into a voltage. It is composed of six pMOS transistors, as well of a current splitter that splits the current to be directed towards the class AB filter configuration. Terminals Iin+ and Iin- are the inputs and V_n and V_p the outputs of this building block.

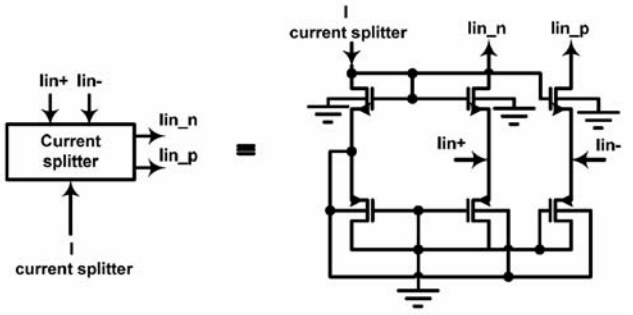



**Figure 21.** Schematic of the implemented current splitter circuit.

The role of the current splitter (Figure 21) is to split the input current into two currents going to the two branches of the class AB 2nd order filter [35]. The two outputs currents are always positive.

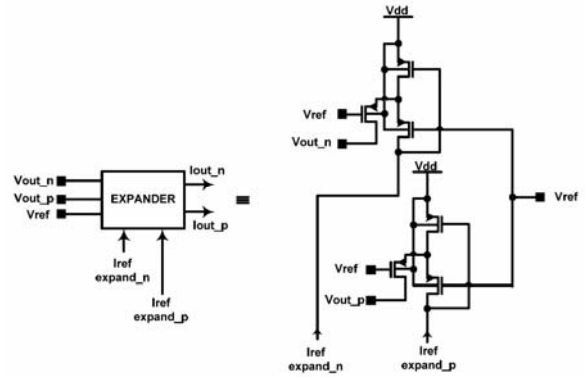



**Figure 22.** Schematic of the implemented expander circuit.

The circuit in Figure 22 shows our expander [34]. It performs the inverse of our compressor circuit by expanding the voltage signal and transforming it into current. Figure 23 shows the overall schematic of our class AB log-domain 6th order band-pass filter. There are three 2nd order band-pass filters, arranged in series, in order to realize a 6th order filter. The six biasing circuits for the integrators, the compressors and the expanders are located at the bottom of the schematic. The three circuits at the top of the figure are the biasing circuits for the current splitter. The circuit outline on the right of the figure is a current adder that sum up the *p*-type and *n*-type output phases to recover the total output from the class AB circuit.

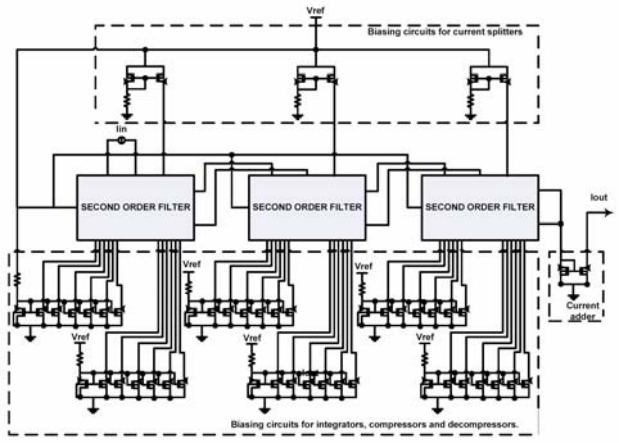



**Figure 23.** Schematic of the implemented sixth order band-pass filter.

### 4.4. Implementation results

We verified our log-domain filter design in a 0.18µm CMOS technology kit using the Spectre® circuit simulator. The results of the 2nd order filters are presented first. We have simulated the AC performance of the circuit depicted in Figure 18. The frequency response is plotted in Figure 24. The filter shows a slope of -60 dB per decade and a central frequency of approximately 1 kHz.

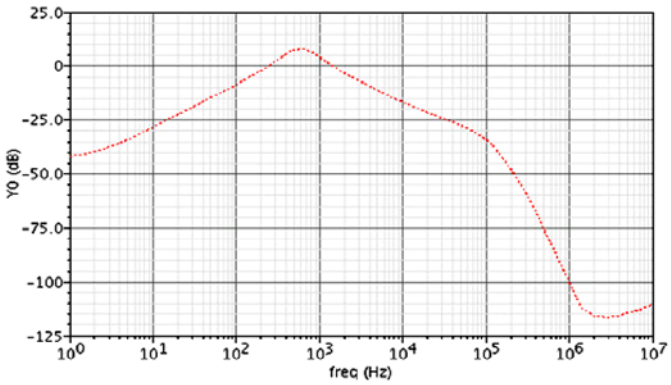



**Figure 24.** Frequency response of 2nd order band-pass filter.

**Table 4.** Power consumption and center frequency of the 6th order filters of the filter bank

| Filter | Power (nW) | Center Frequency (Hz) |
|---|---|---|
| F1 | 559 | 5000 |
| F2 | 285 | 2500 |
| F3 | 230 | 1250 |
| F4 | 184 | 625 |
| F5 | 181 | 313 |
| F6 | 159 | 156 |

The power consumption for each of the 6th order filter was estimated. The filter F1 corresponds to the highest frequency filter. As seen from Table 4, power consumption tends to decrease with the frequency of the filter as it is proportional to the current over the integrating capacitor (I/C). In order to observe the functioning limits of our circuits, the total harmonic distortion (THD) was taken into

consideration. With a THD lower than 0.1%, we obtain the currents shown in Table 5. It is concluded that the input currents must be lower than 1 nA for the circuits to work at the best conditions. The supply voltage of the filter is 1.8 V. A parametric analysis showed that the design is robust against the variation of the supply voltage.

**Table 5.** Filter input range

| Filter | Maximal Input Current (nA) |
|---|---|
| F1 | 1 |
| F2 | 0.9 |
| F3 | 0.6 |
| F4 | 0.6 |
| F5 | 0.6 |
| F6 | 0.5 |

Overall, a total power consumption of 5.6 μW was estimated for the entire filter bank which is indeed very low and quite acceptable for our application. The filter bank occupies 0.25 mm of chip area, which is also good. An integrated circuit was fabricated in TSMC CMOS technology 0.18 ìm to validate our design. The IC mask layout is shown in Figure 25.

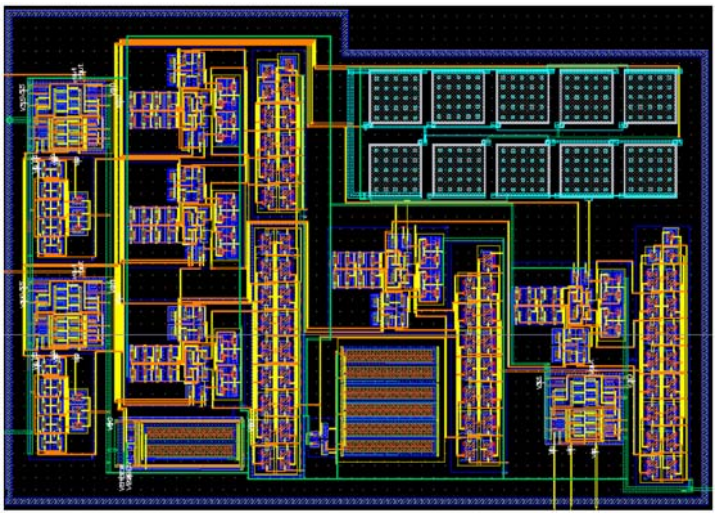


**Figure 25.** Layout of the proposed wavelet filter.

## 5. Conclusion

Wavelet signal processing plays a key role in neural interface applications since the reliability and performance of a multi-channel neural recording implant directly depends on the efficient detection and compression of the neural waveforms. The DWT and the CWT have been extensively used for bioelectric signal processing, and their superior detection and compression efficiency compared to other methods is already known [36-41]. We have developed digital and analog building blocks for a low-power implementation of neural spike detectors and compressors.

In digital implementation, two approaches, namely the polyphase and the lifting structures were compared in terms of power consumption and chip area for the implementation of a DWT. The polyphase structure outperforms the lifting structure in terms of RMSE and SNR for fixed-point implementations. Both structures can be implemented using either fully parallel or folded architecture. Between the two architectures, the latter is recommended as it offers better chip area and higher power savings. Between the basic polyphase structure and the folded polyphase structure, the latter offers lower static power consumption (at identical CMOS technologies). This result is significant because as the implementations are pushed further towards deep submicron technologies, static power consumption become increasingly important.

Then, the CWT was implemented using an analog filter bank comprising high–order log-domain filters. The proposed structure allows fitting the design into the low power and area budgets. The key design steps for such circuit structure were presented along with the simulation and implementation results. The combination of logdomain technique and weak inversion that we have used for our wavelet filter bank enables an ultra-low power implementation of a full WT processor. An analog design can offer very small chip area and achieve high power savings. Indeed, a digital implementation operating in subthreshold regime may produce the same level of power saving in addition to ease of design and inherent flexibility. This question of interest will be investigated in our future work.

As shown in this paper, the power and bandwidth limitations in neural implants can be efficiently addressed by employing on-chip processing cores. The proposed approach employs real-time wavelet signal processing to achieve better signal detection and compression. In fact, automatic detection and compression of biopotentials signals considerably reduces the amount of data to be handled by a wireless transmitter, leading to great reduction in power consumption.

Although tremendous progress has been made in the area of neural interfacing, there are still critical needs for increased performance. Wireless operation of implants is necessary because it enables for battery-less operation and avoids percutaneous connections, which reduce the risk of infection and provide more flexibility. There is an urgent need for more efficiency in wireless powering interfaces as well as in on-chip signal processors before we can benefit from the several emerging clinical applications in this field.

## Acknowledgements

The present work is supported by a Canada Research Chair on smart medical devices. Also acknowledgements are due to NSERC, Canada for the financial support, CMC Microsystems for the design tools, and the graduate students of Polystim Neurotechnologies Laboratory, more particularly Virginie Simard, Jean-François Roy and Cyprien Dumortier for their contributions.

## References

[1] L. R. Hochberg et al., Neuronal ensemble control of prosthetic devices by a human with tetraplegia, Nature 442 (2006), 164-171.

[2] J. Wessberg and M. A. Nicolelis, Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys, J. Cogn. Neurosci. 16 (2004), 1022-1035.

[3] J. K. Chapin, Using multi-neuron population recordings for neural prosthetics, Nat. Neurosci. 7 (2004), 452-455.

[4] M. A. Lebedev and M. A. Nicolelis, Brain-machine interfaces: past, present and future, Trends Neurosci. 29 (2006), 536-546.

[5] B. Gosselin et al., A mixed-signal multichip neural recording interface with bandwidth reduction, IEEE Trans. Biomedical Circuits and Systems 3 (2009), 129-141.

[6] R. R. Harrison et al., A low-power integrated circuit for a wireless 100-electrode neural recording system, IEEE J. Solid-State Circuits 42 (2007), 123-133.

[7] Y. Perelman and R. Ginosar, An integrated system for multichannel neuronal recording with spike/LFP separation, integrated A/D conversion and threshold detection, IEEE Trans. Biomedical Engineering 54 (2007), 130-137.

[8] A. M. Sodagar, K. D. Wise and K. Najafi, A fully integrated mixed-signal neural processor for implantable multichannel cortical recording, IEEE Trans. Biomedical Engineering 54 (2007), 1075-1088.

[9] Y.-K. Song et al., A brain implantable microsystem with hybrid rf/ir telemetry for advanced neuroengineering applications, 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 2007, pp. 445-448.

[10] T. Stieglitz, M. Schuetter and K. P. Koch, Implantable biomedical microsystems for neural prostheses, IEEE Engineering in Medicine and Biology Magazine 24 (2005), 58-65.

[11] S. Yoon-Kyu et al., Development of a chipscale integrated microelectrode/ microelectronic device for brain implantable neuroengineering applications, IEEE Trans. Neural Systems and Rehabilitation Engineering 13 (2005), 220-226.

[12] B. Gosselin, M. Sawan and C. A. Chapman, A low-power integrated bioamplifier with active low-frequency suppression, IEEE Trans. Biomedical Circuits and Systems 1 (2007), 184-192.

[13] B. Gosselin and M. Sawan, Circuits techniques and microsystems assembly for intracortical multichannel ENG recording, IEEE Custom Integrated Circuits Conference (CICC), Invited paper, San Jose, California, USA, 2009.

[14] B. Gosselin and M. Sawan, An ultra low-power CMOS automatic action potential detector, IEEE Trans. Neural Systems and Rehabilitation Engineering 17 (2009), 346-353.

[15] B. Gosselin, M. Sawan and E. Kerherve, Linear-phase delay filters for ultra-low-power signal processing in neural recording implants, IEEE Trans. Biomedical Circuits and Systems 4 (2010), 171-180.

[16] P.-Y. Robert and M. Sawan, An independent-component-analysis-based time-space processor for the identification of neural stimulation sources, 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 2007, pp. 3876-3879.

[17] J. F. Roy and M. Sawan, A fully reconfigurable controller dedicated to implantable recording devices, IEEE-NEWCAS Conference, 2005, The 3rd International, 2005, pp. 303-306.

[18] C. Dumortier, B. Gosselin and M. Sawan, Wavelet transforms dedicated to compress recorded ENGs from multichannel implants: comparative architectural study, 2006 IEEE International Symposium on Circuits and Systems (ISCAS), 2006, pp. 2129-2132.

[19] V. Simard, B. Gosselin and M. Sawan, An analog wavelet processor for a fully implantable cortical signals recording system, 8th Vienna International Workshop on Functional Electrical Stimulation (IFESS), Vienna, Austria, 2004, pp. 164-167.

[20] J.-F. Roy, Unité de contrôle et de compression de données pour la partie implantable d'un électroneurogramme, Génie électrique, Master Thesis, École Polytechnique de Montréal, Montreal, 2006, p. 127.

[21] S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Analysis and Machine Intelligence 11 (1989), 674-693.

[22] F. Marino, D. Guevorkian and J. T. Astola, Highly efficient high-speed/low-power architectures for the 1-D discrete wavelet transform, IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing [see also IEEE Trans. Circuits and Systems II: Express Briefs] 47 (2000), 1492-1502.

[23] L. Chung-Jr, C. Kuan-Fu, C. Hong-Hui and C. Liang-Gee, Lifting based discrete wavelet transform architecture for JPEG 2000, Circuits and Systems, ISCAS 2001, The 2001 IEEE International Symposium, Vol. 2, 2001, pp. 445-448.

[24] H. Chao-Tsung, T. Po-Chih and C. Liang-Gee, Efficient VLSI architectures of lifting-based discrete wavelet transform by systematic design method, IEEE International Symposium on Circuits and Systems, ISCAS 2002, Vol. 5, 2002, pp. V-565-V-568.

[25] L. Hongyu, M. K. Mandal and B. F. Cockburn, Efficient architectures for 1-D and 2-D lifting-based-wavelet transforms, IEEE Trans. Signal Processing [see also IEEE Trans. Acoustics, Speech, and Signal Processing] 52 (2004), 1315-1326.

[26] A. Bellaouar and M. Elmasry, Low-power Digital VLSI Design: Circuits and Systems, Springer, 1995.

[27] T. Edwards and G. Cauwenberghs, Analog VLSI Processor Implementing the Continuous Wavelet Transform, NIPSS, 1995.

[28] S. A. P. Haddad, S. Bagga and W. A. Serdijn, Log-domain wavelet bases, IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications 52 (2005), 2023-2032.

[29] S. A. P. Haddad and W. A. Serdijn, Ultra Low-power Biomedical Signal Processing: An Analog Wavelet Filter Approach for Pacemakers, Springer, 2009.

[30] K. L. Su, Time-domain Synthesis of Linear Networks, Englewood Cliffs, Prentice-Hall, N.J., 1971.

[31] J. M. H. Karel, R. L. M. Peeters, R. L. Westra, S. A. P. Haddad and W. A. Serdijn, An L2-based approach for wavelet approximation, 44th IEEE Conference on Decision and Control European Control Conference (CDC-ECC'05), 2005, pp. 7882-7887.

[32] G. W. Roberts and V. W. Leung, Design and Analysis of Integrator-based Log-domain Filter Circuits, Springer, 2000.

[33] F. Serra-Graells, A. Rueda and J. L. Huertas, Low-voltage CMOS Log Companding Analog Design, Springer, 2003.

[34] D. Python and C. C. Enz, A micropower class-AB CMOS log-domain filter for DECT applications, IEEE J. Solid-State Circuits 36 (2001), 1067-1075.

[35] D. R. Frey, Log-domain filtering: an approach to current-mode filtering, IEE Proceedings on Circuits, Devices and Systems 140 (1993), 406-416.

[36] A. Diedrich, W. Charoensuk, R. J. Brychta, A. C. Ertl and R. Shiavi, Analysis of raw microneurographic recordings based on wavelet de-noising technique and classification algorithm: wavelet analysis in microneurography, IEEE Trans. Biomedical Engineering 50 (2003), 41-50.

[37] K. Kyung Hwan and K. Sung June, A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio, IEEE Trans. Biomedical Engineering 50 (2003), 999-1011.

[38] K. Dohoon, C. Joon Hwan and K. Taejeong, A wavelet (MZ-DWT) based action potential detector at low signal-to-noise ratios, 2nd International IEEE EMBS Conference on Neural Engineering 2005, 2005, pp. 301-303.

[39] C. Joon Hwan, J. Hae Kyung and K. Taejeong, A new action potential detector using the MTEO and its effects on spike sorting systems at low signal-to-noise ratios, IEEE Trans. Biomedical Engineering 53 (2006), 738-746.

[40] K. G. Oweiss, A systems approach for data compression and latency reduction in cortically controlled brain machine interfaces, IEEE Trans. Biomedical Engineering 53 (2006), 1364-1377.

[41] A. Salmanpour, L. J. Brown and J. K. Shoemaker, Detection and classification of raw action potential patterns in human Muscle Sympathetic Nerve Activity, 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 2928-2931.