Integer-based digital processor for the estimation of phase synchronization between neural signals.

James Brian Romaine, Manuel Delgado-Restituto, Juan Antonio Leñero-Bardallo and Ángel Rodríguez-Vázquez
Institute of Microelectronics of Sevilla and University of Sevilla
Avda. Americo Vespucio s/n, 41092-Seville, SPAIN
Email: {romaine, mandel, juanle, angel}@imse-cnm.csic.es

***Abstract*—This paper reports a low area, low power, integer-based neural digital processor for the calculation of phase synchronization between two neural signals. The processor calculates the phase-frequency content of a signal by identifying the specific time periods associated with two consecutive minima. The simplicity of this phase-frequency content identifier allows for the digital processor to utilize only basic digital blocks, such as registers, counters, adders and subtractors, without incorporating any complex multiplication and or division algorithms. The low area and power consumptions make the processor an extremely scalable device which would work well in closed loop neural prosthesis for the treatment of neural diseases.**

I. Introduction

Epilepsy is a neurological disorder, defined as the re-occurrence of two or more unprovoked seizures separated by a minimum of 24 hours. This disorder affects to approximately 50 million people. Epilepsy is successfully eradicated by means of anti-convulsive medication or invasive surgery in up to 70% of persons treated [1]. This however leaves a 30% gap of 15 million patients who are unsuccessfully treated by any existing method. Recently neural prosthesis have been introduced in the form of closed loop feedback systems which detect specific bio-markers in neural signals during the pre-ictal and or ictal stages. These feedback systems stimulate zones of the brain, with the goal of disrupting the excitability of the neurons around specific regions [2]. This stoppage of neural firing could prevent focal to generalized spreading of seizures.

One bio-marker which could hold the key to early detection of seizures was proposed in [3]. This bio-marker is detectable as a large drop in synchronization between neural signals during the pre-ictal and ictal stages, i.e., prior to a seizure. To detect such bio-marker, neural processors such as [2] have been designed. However, they utilize complex algorithms in the form of the Hilbert transform (HT) [4] to extract the instantaneous phase of the signals and use complex quantitative measures of indexing, such as the phase locking value (*PLV*). The HT and *PLV* calculations incorporate many complex operations such as multiplication and division [5] and hence increases power and area consumption. This makes multichannel calculation difficult as scalability is reduced.

978-1-5090-0493-5/16/$31.00

The processor proposed in this paper uses a combination of signal smoothing, minima detection and error accumulation in order to calculate the phase-frequency differences between two neural signals. The processor is integer-based and incorporates only power of two divisions and multiplications. This processor has an accuracy comparable to that of more complex algorithms and the small area consumption and small power consumption could be exploited for multichannel calculations.

The paper is organized as follows. Sec. II reviews some of the methods most commonly used for quantifying the phase synchronization between signals and introduces the proposed algorithm which is referred to as Discrete Distance Approximation (DDA). Then, Sec. III presents the digital realization of the different functional blocks required for DDA computation. Afterward, Sec. IV describes the FPGA implementation of the processor and Sec. V presents the obtained experimental results. Finally, Sec. VI gives some concluding remarks.

II. Synchronization Algorithms

Methods for identifying and quantifying the phase relationship between signals stretch far and wide [6]. In the midst of which lie several common steps for the detection of synchronization. In most cases the first step is to filter the input signals into a desired band. In the case of neurophisiological signals obtained, for instance, by EEG or ECoG methods, these bands range from almost zero to 100Hz and can be separated as follows. $\alpha\sim\ 8-12Hz, \beta\ \sim\ 12-30Hz$, $\theta\ \sim\ 4-7Hz$, $\delta\ \sim\ 0.1-3Hz$, $\gamma\ \sim\ 30-100Hz$ [7]. The next step is to estimate the phase angle φ of both filtered signals. This is calculated in most cases by extracting the instantaneous phase, although more simplistic estimation methods do exist. The third step involves calculating the phase angle error, $\varphi_{sig1}-\varphi_{sig2}$, between the two signals. Finally the amount of error between the phase angles need to be quantified over a trial period and indexed. If detection is required then smoothing the resultant data can provide more accuracy to detection schemes.

A. Hilbert Transform and Phase Locking Value

The Hilbert transform is a common method for the detection of instantaneous phase angles. This approach creates a 90°phase shifted version of an original signal yielding both real and imaginary parts finally leading to the equation. $A(t)e^{j\varphi(t)}$ where $\varphi(t)$ is the instantaneous phase [4]. This

phase can be extracted using the $\arctan(\cdot)$ function. The phase locking value usually accompanies the HT

as an index for quantification. It is calculated as:

$$PLV = |\frac{1}{N}\sum_{j=0}^{N-1} \exp i\triangle\varphi(j)| \quad (1)$$

This index provides a mean phase coherence quantity bounded from 0 to 1, where 1 is perfect synchronization and 0 is no synchronization [8].

B. Discrete Distance Approximation and Synchronization Index

The discrete distance approximation algorithm proposed herein relies on the detection of minimum to minimum transition periods in order to identify large changes in phase frequency content between two signals. The transition period is an integer count every clock cycle between consecutive minima i.e, the number of samples between consecutive minima at the clock sampling frequency. The transition period can be noted as T_S^n where S is the signal to which the transition period belongs and n is the transition index e.g., $T_1^{n1}$for signal 1, transition $n1$ and T_2^{n2} for signal 2 transition $n2$.

The DDA algorithm, represented by the block diagram of Fig. 1, works as follows:

1) Pre-process signals (smoothing) (block A for signal 1 and block B for signal 2).
2) Utilizing a minimum detection scheme, search for minima in both signals. If minimum is detected in signal S, output counter value to corresponding register in block C and restart counter. The value stored in the register corresponds to T_S^n (blocks A and B).
3) If a minimum was found in signal 1, the associated register will hold the period T_1^{n1}. Next, check to see if a transition period from signal 2, T_2^{n2}, has been stored in the corresponding register. If true, calculate the absolute difference (error) between the two, i.e. $\triangle T_{diff}(n) = \left|T_1^{n1} - T_2^{n1}\right|$. If T_2^{n2} is yet to be stored, continue searching for transition periods in signal 1, overwriting the values in the register until T_2^{n2} is stored. Note that the opposite is true if T_2^{n2} is stored and $T_1^{n1}$is yet to be stored (block C).
4) Accumulate errors and calculate the Synchronization Index (*SI*) defined as:

$$SI = \sum_{n=0}^{K-1} \frac{\triangle T_{diff}(n)}{\triangle T_{max}} \quad (2)$$

where $\triangle T_{max} = f_s/f_{min} - f_s/f_{max}$ rounded to power of 2, fs is the sampling rate of the input signals, $fmin$ and $fmax$ denote the lower and upper limits of the frequency band in which signals 1 and 2 are allocated, and K is the number of samples to accumulate before outputting value, rounded to nearest power of 2.
Smooth the output (block D).

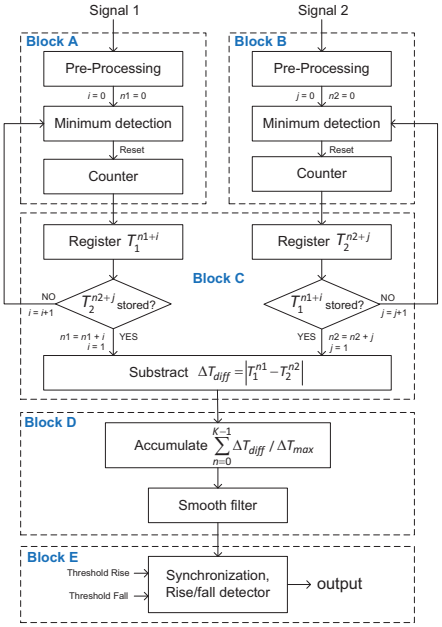

Fig. 1. Hardware/algorithm flow diagram DDA processor.

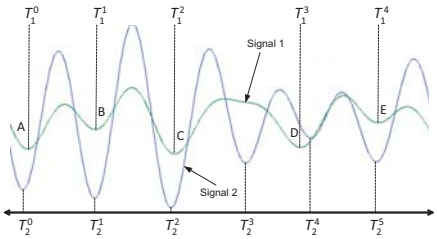

Fig. 2. Time segment of two neural signals.

5) Detect rise or falls in synchronization using thresholds based on selected frequency band (block E).

As an example, in Fig. 2 we compare a time segment of two neural signals. Each time stamp in the figure represents the number of samples from a previous minimum. e.g T_1^1 is an integer value representing the numnber of samples counted from point T_1^0.

At point A the calculation would therefore be $\triangle T_{diff}(0) = |T_1^0 - T_2^0|$. At point B we have $T_1^1$and T_2^1 and hence the calculation is $\triangle T_{diff}(1) = |T_1^1 - T_2^1|$. At point C the calculation $\triangle T_{diff}(2) = |T_1^2 - T_2^2|$ at point D we calculate $\triangle T_{diff}(3) = |T_1^3 - T_2^3|$. At point E because T_2^5 is detected before T_1^4, we replace T_2^4 with $T_2^5$and the calculation is hence, $\triangle T_{diff}(4) = |T_1^4 - T_2^5|$. Therefore, the synchronization index for this segment will be:

$$SI = \frac{|T_1^0 - T_2^0| + |T_1^1 - T_2^1| + |T_1^2 - T_2^2| + |T_1^3 - T_2^3| + |T_1^4 - T_2^5|}{\triangle Tmax} \quad (3)$$

III. Digital Design

The digital design begins with a pre-processing stage (blocks A and B in Fig. 1) which involves a simple smoothing filter. Afterward, the minima detection scheme works via the pair wise comparison of a current sample $S(n)$ and a previous

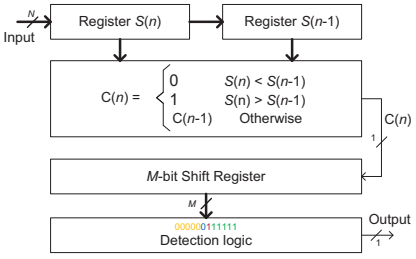

Fig. 3. Minimum detection scheme.

sample $S(n-1)$, as shown in Fig. 3. A shift register of $N \times 2$ bits holds the current sample $S(N)$ and the previous sample $S(n-1)$, hence updating the values every cycle. The two samples are then compared using an $N \times 2$ bit comparator as per the constraints shown in Fig. 3. The comparison results are stored in a shift register with an even number M of memory elements, where the most recent comparison is stored in the M-th position. A minimum is identified if the majority of stored values in the range $[0,\ M/2]$ of the register is filled with logic '0' and the majority of stored values in the range $[(M/2)+1,\ M]$ is filled with '1'. However due to non ideal signals the detection logic allows for one outlier either side. Assumming $M = 10$, in the example of Fig. 4, the detection logic would see the value "0000011111" and hence identify this as a minimum. In the example of Fig. 5, although the bit 0 and bit 7 are inverted, "1000011011" the one outlier rule from the detection logic also detects this as a minimum. Due to the pair-wise approach this minima detection scheme utilizes little hardware such as comparators, registers and some basic logic.

Counters running at the sample frequency of signals 1 and 2 measure corresponding transitions periods. Every time a minimum is detected the counter is reset. Following the procedure in Fig. 1, the values stored in the counters are not dumped into the registers in block C until the transition periods of signal 1 and signal 2 are available. Note that some transition periods may be skipped in this operation. Once the registers are filled, a substractor and some multiplexors for the calculation of the absolute value are used for deriving the equivalet phase difference between both signals, $\triangle T_{diff}$.

In block D, the logic used for the calculation of the synchronization index (*SI*) incorporates an accumulator, which sums up the $\triangle T_{diff}(n)$ values, and a counter to control the *K* value, i.e. the number of additions. This block also contains a hardware friendly exponential smoothing filter which allows for more accurate detection in the final block (block E). Such block uses two comparators in order to compare the value from block D with two constant threshold values "Rise" and "Fall". If the input to block E is less than the"Rise" threshold a counter gets +1. If the next *P* consecutive values are also less than "Rise" (the counter = *P*), then a rise in synchronization is said to be detected. The opposite is true is *P* consecutive values are higher than the "Fall" threshold.

IV. Tools and Methods

The digital design was realized in the *VIVADO* design environment using a *XILINX* ARTIX 7 AC701xc7a200tfbg676-

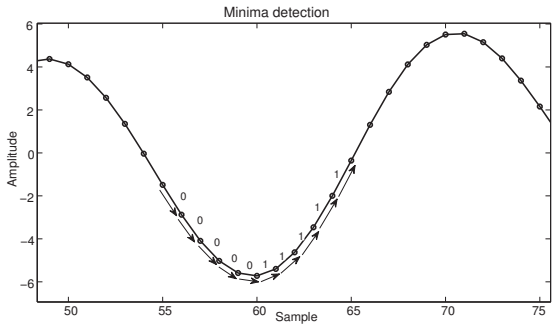

Fig. 4. Example ideal signal.

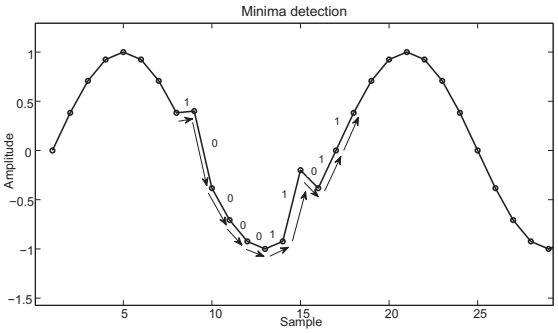

Fig. 5. Example non ideal signal.

2 evaluation board. For ease of testing, a hardware Co-Simulation of the DDA algorithm was ran through the *MATLAB DSP SIMULINK* environment along withe a *MATLAB* based calculation of the HT and *PLV* index. The signals used came from the *Freiburg, Epilepsiae project [9].* This database is a world leading database on epilepsy, containing 275 patients of both scalp and intracranial recordings. The signals (10-bit long sampled at 1024 Hz) were intracranial recordings first filtered into the beta band and read into *SIMULINK* before being transferred to the FPGA device via a JTAG connection. The data from the FPGA was also collected via the JTAG connection and plotted in *MATLAB*. Fig. 6 shows a general overview of the setup.

V. FPGA Implementation Results

Using the *VIVADO* default implementation strategy, Table. V shows the resources utilized for the entire digital design. The results shows that only fractions of a percent of the

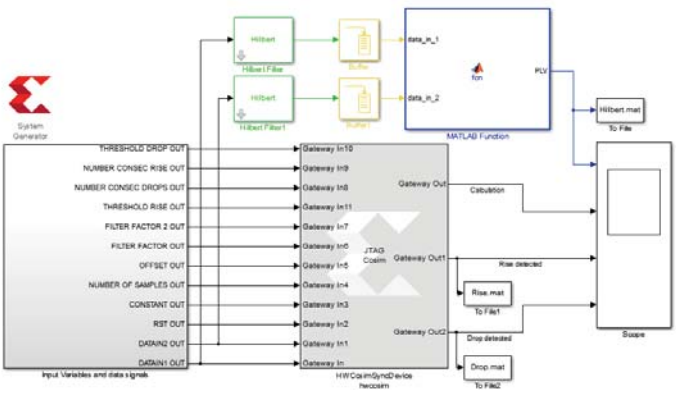

Fig. 6. *VIVADO* DSP generator setup.

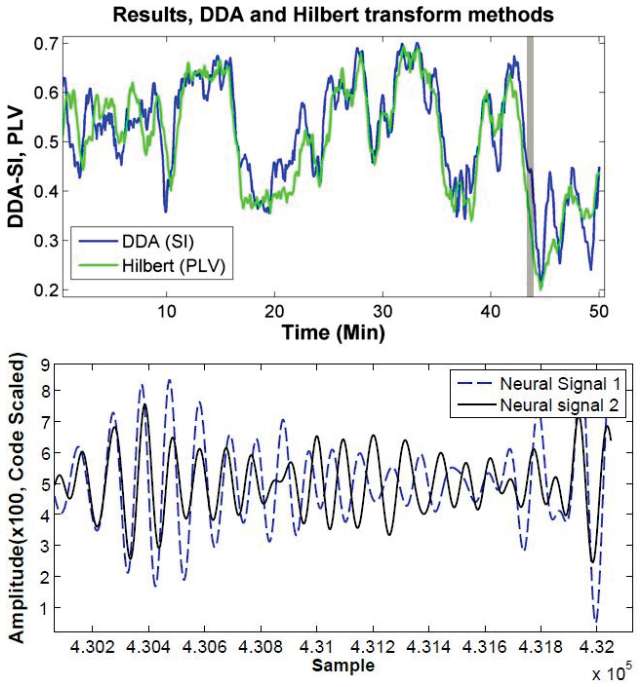

Fig. 7. Hardware Co-simulation results for DDA(*SI*) processor and HT(*PLV*) algorithm (top), (bottom) example neural segment shows how the neural signals change in synchronicity, these corresponds to a small section of the shaded area.

Table I
UTILIZATION REPORT VIVADO.

Resource	Utilization DDA	Utilization [10]
Slice LUTs	211 (0.15%)	1513
Slice Registers	283 (0.10%)	N/A
Slice	92 (0.27%)	917
LUT as Logic	211 (0.15%)	N/A
LUT FF pairs	288 (0.21%)	1614
IO	40 (10%)	116
Clocking	1 (3.12%)	1

available resources are being utilized to realize this processor and is much more favorable than other designs for HT such as that in [10].

Fig. 7 shows the results from the Co-simulation, for both the DDA(*SI*) processor and the HT(*PLV*) for a pair of neural signals. From the results we can note that both the DDA(*SI*) and HT(*PLV*) methods share a common shape, where most large drops and or rises in synchronicity are detected. This close relationship between both the HT(*PLV*) and DDA(*SI*) methods can best be seen in Fig. 8 which shows a mean correlation of approximately 0.9 between the obtained results. The most interesting feature in Fig. 7 (top) is the large drop in synchronization at approximately 43 minutes (indicated with a shaded line). The large drop passes the threshold fall value and triggers the drop detector. This drop could be an indicative sign of an upcoming seizure event. Fig. 7 (bottom) shows how the neural signals couple and de-couple over a segment of the shaded area.

VI. CONCLUSION

This paper proposes an algorithm, validated with a FPGA implementation, for quantifying the phase synchronization between signals. The processor produces very comparable results to other algorithms which are much more computationally hungry and area in-efficient. The processor is fully capable of detecting the crucial bio-markers for early prediction with very little hardware and power cost. This makes for an extremely scalable device capable of multichannel calculations, in turn increasing the likelihood of detection.

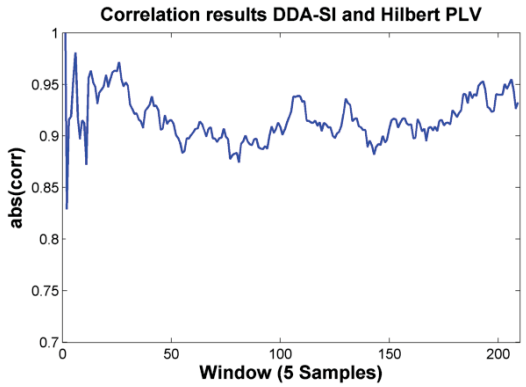

Fig. 8. Absolute correlation between synchronization results from Fig. 7(top).

ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Science & Innovation under grant TEC2012-33634, the ONR under grant N000141110312 and the FEDER Program.

REFERENCES

[1] A. Giuliano *et al.*, "Epilepsy Care in the World WHO 2005. Epilepsy Atlas, 2005." 2005.

[2] K. Abdelhalim *et al.*, "A phase synchronization and magnitude processor VLSI architecture for adaptive neural stimulation," in *2010 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Nov. 2010, pp. 5–8.

[3] F. Mormann *et al.*, "Epileptic seizures are preceded by a decrease in synchronization," *Epilepsy Research*, vol. 53, no. 3, pp. 173–185, Mar. 2003.

[4] M. Le Van Quyen *et al.*, "Comparison of Hilbert transform and wavelet methods for the analysis of neuronal synchrony," *Journal of Neuroscience Methods*, vol. 111, no. 2, pp. 83–98, Sep. 2001.

[5] D. E. Troncoso Romero *et al.*, "Digital FIR Hilbert Transformers: Fundamentals and Efficient Design Methods," in *MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1*, V. Katsikis, Ed. InTech, Sep. 2012.

[6] M. Rosenblum *et al.*, "Chapter 9 Phase synchronization: From theory to data analysis," in *Handbook of Biological Physics*, F. M. a. S. Gielen, Ed. North-Holland, 2001, vol. Volume 4, pp. 279–321.

[7] NeuroSkyInc, "Brain Wave Signal (EEG) of NeuroSky, Inc." Tech. Rep., Dec. 2009.

[8] F. Mormann *et al.*, "Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy patients," *Physica D: Nonlinear Phenomena*, vol. 144, no. 3,4, pp. 358–369, Oct. 2000.

[9] "The freiburg EEG database, url:http://epilepsy.uni-freiburg.de/freiburg-seizure-prediction-project/eeg-database."

[10] A. Das *et al.*, "Design and implementation of FPGA based linear all digital phase-locked loop," in *2012 Annual IEEE India Conference (INDICON)*, Dec. 2012, pp. 280–285.