Cochlear Implant Stimulation Strategy Based on Wavelet Transform: Toward DSP Real Time Implementation

Amira Derbel, Mohamed Ghorbel, Mounir Samet and Ahmed Ben Hamida
Technologies de l'Information & Electronique Médicale
National Engineers School of Sfax
Sfax – Tunisia
Amira.derbel@yahoo.fr

Jean Tomas
Laboratoire IMS, CNRS UMR 5218,
Bordeaux, France
Jean.tomas@ims-bordeaux.fr

***Abstract*—This paper discuses the real-time implementation of a cochlear implant strategy based on wavelet transform (WT). The cochlear implant converts in real time, sounds perceived from the outside, into electrical pulses that stimulate nerve endings in the cochlea. These electrical stimuli are interpreted by the brain as sounds. To achieve a real-time electrical stimulation strategy, optimization steps have been taken. These steps consist of using dynamic link libraries, making efficient memory allocation and performing fixed-point arithmetic. Using these steps, we have been able to reduce the processing time to approximately 10.49µs on a 720MHz TMS320C6416 DSP board. So, we can note that the execution time represents only about 21% of the minimum biphasic stimulation pulse width (25µs/phase).**

I. INTRODUCTION

Prosthetic devices such as cochlear implants are used to restore partial hearing in profoundly deaf people or in patients suffering from nerve deafness [1]. It delivers an electrical signal to the cochlea through a series of electrodes placed inside it. It uses a speech processor which converts the sound produced by a microphone into electrical signals [2].

The main function of the speech processor is to decompose the input signal into its frequency components, in the same way as the healthy cochlea which analyzes the input signal into its frequency components. Much of the success of cochlear implants can be attributed to the development of sophisticated a real-time signal-processing algorithms implemented in the speech processor. These algorithms are designed to some extent to mimic the function of a healthy cochlea [2], [9].

Two important factors need to be considered when implementing in real-time signal-processing algorithms for cochlear implant applications. First, the frequency decomposition, which can be done using either a filter bank or a fast Fourier transform (FFT), needs to yield adequate frequency resolution similar to that of the healthy ear. Fine spectral resolution would be highly desirable for transmission of fine spectral cues ("fine structure") to the implant recipients. Fine resolution is also needed for the implementation of current steering or current focusing stimulation strategies. Second, and perhaps most important, the incurred signal delay needs to be small [8]. This factor may consider different aspects such as speed, reliability, power consumption and functional units [5].

In this work, we explore a TMS320C6416 DSP real time implementation of a continuous wavelet filtering module based strategy for cochlear implant processors. To guarantee that, the time-consuming components would be sufficient for processing in real time, we profiled the execution of this algorithm with optimizer assembly code.

This paper includes five sections which are organized as follows. Section I is an introduction. Section II is an over view of stimulation strategy architecture. Section III covers a real time implementation and the optimization steps taken to reach a real time working solution on a TMS320C6416 digital signal processor (DSP) board. The results and discussions are presented in Section IV. Finally, the conclusions are provided in Section V.

II. COCHLEAR IMPLANT STRATEGY FORMALISM

Stimulation Strategy Principle

The function of the signal processor in a cochlear implant primarily consists of dividing an input speech signal into a number of frequency bands (12-22) in order to extract the signal strength in each band for exciting the implanted electrodes accordingly. In other words, the signal processor is programmed to emulate the functioning of the inner ear. The most common strategies used in commercial cochlear implants are Continuous Interleaved Sampling (CIS) and Advanced Combination Encoder (ACE). Both of these strategies can be realized using either a filter bank approach [3]. When using the filter bank approach, a set of bandpass filters is used to divide the signal into a number of frequency bands or channels. Then, the filtered output is fullwave rectified and passed through lowpass filters to extract channel envelopes based on which, electrodes pulses are generated (figure 1).

978-1-61284-137-3/11/$26.00 ©2011 IEEE

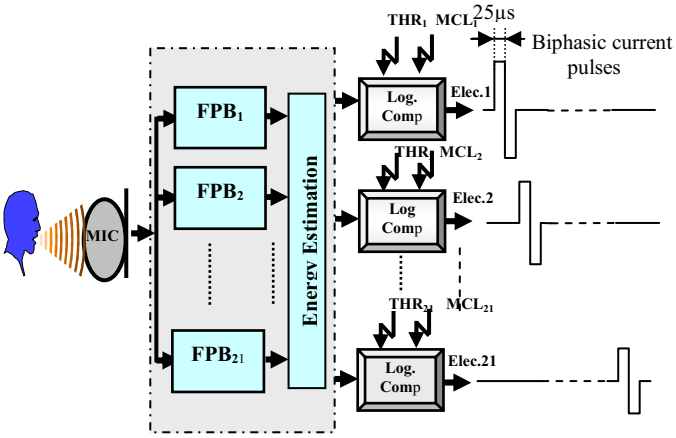


Figure 1. Bloc architecture of the proposed stimulation strategy

Wavelet approach, which is similar to the perception process of the auditory system, is used to decompose the input speech signal into a number of frequency bands. Basing on human auditory system, our considered filter bank spectrum would be divided according to ERB critical bands (table1). With 16-kHz sampling frequency, our band-pass filter module was composed of 21 filters which correspond to 21 energy [5].

TABLE 1. ERB CRITICAL BANDS

N	Center Freq. (f_0) [Hz]	inferior limit [Hz]	superior limit [Hz]
1	50	20	100
2	150	100	200
3	250	200	300
4	350	300	400
5	450	400	510
6	570	510	630
7	700	630	770
8	840	770	920
9	1000	920	1080
10	1170	1080	1270
11	1370	1270	1480
12	1600	1480	1720
13	1850	1720	2000
14	2150	2000	2320
15	2500	2320	2700
16	2900	2700	3150
17	3400	3150	3700
18	4000	3700	4400
19	4800	4400	5300
20	5800	5300	6400
21	7000	6400	7700
22	8500	7700	9500
23	10500	9500	12000
24	13500	12000	15500

B. Continuous Wavelet Transform

The continuous wavelet transform CWT is used to design a digital filter banc for a speech processing strategies used for the external part of cochlear prosthesis system [13]. The time-domain definition of CWT of f(t) is given by equation 1:

$$CWT_f(a,b) = \int_{-\infty}^{+\infty} \psi^*_{a,b}(t) f(t) dt \qquad (1)$$

This analysis uses a family of functions ψa,b(t) based on a mother wavelet ψ [5]. ψa,b(t) is given by the following equation:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi(\frac{t-b}{a}) = \frac{1}{\sqrt{a}} \tilde{\psi}(\frac{t-b}{a}) e^{-iw_0 t} \qquad (2)$$

,a∈ $\Re^+$, b∈ $\Re$

Where 'a' is a dilatation parameter, 'b' is translation parameter and $\tilde{\psi}(t)$ is the envelope of $\psi(t)$.

The commonly used mother wavelet in the CWT is Morlet and Marr wavelet [5]. Due to the easy selection of its center frequency and its exponential form, Morlet wavelet is chosen as the mother wavelet to decompose the audible spectrum for the cochlear bank filter. Morlet wavelet is defined as:

$$\psi(t) = \frac{1}{\sqrt{2\pi}} e^{\frac{-t^2}{2}} e^{-iw_0 t} \qquad (3)$$

Basically, the idea of Auditory Model Based Wavelet Transform is to make the temporal envelope of the mother time varying according to the characteristics of the target signal by introducing the elementary paving. This last will adjust the Morlet wavelet to emulate the ERB model of the auditory system in the following way:

$$\psi_{EPB}(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-t^2}{2\sigma^2}} e^{-iw_0 t} \qquad (4)$$

The first sigma is just the scaling factor to insure the energy is the same for every mother wavelet and the second sigma adjusts the envelope of the mother wavelet ψ (t) without adjusting its center frequency.

In order to share the considered audible spectrum and fix the width of each band corresponding to ERB model, an interested study presented in [5] and finished by a final expression of a auditory model based mother wavelet that given by :

$$\psi(t) = \sqrt{\frac{\pi}{1.4}}(25 + 75(1+1.4 f_0^2)^{0.69}) * e^{\frac{-[25+75(1+1.4 f_0^2)^{0.69}]^2 \pi^2 t^2}{1.4}} e^{-i2\pi f_0 t} \qquad (5)$$

III. REAL-TIME DSP IMPLEMENTATION

The main goal of this work is to process the speech signal in real time and to deliver stimulation information to the implant. Therefore, a specific platform is used to allow the experimenter to implement the process at run time. This platform is composed of two parts: a PC and a TMS320C6416 digital signal processor (DSP) board.

In order to create a high performance C6416 implementation design, ours has followed three steps for algorithm implementation. We are used first with C programming language. Then, a linear assembly code [4]. Finally, in this paper, we optimize the algorithm and the assembly code in order to achieve the necessary performance for real-time processing [9].

A. TMS320C6416 DSP Board Overview

In this project, we used the TMS320C6416 digital signal processor (DSP) board. C6416 is a recent version of the TMS320 family of DSPs provided by Texas Instrument. It uses VelociTI, high-performance and a very long instruction-word (VLIW) architecture. The C6416 register file extends this by additionally supporting packed 8-bit types and 64-bit fixed point data types. There are two general-purpose register files, A and B, which have 32-bit registers in total [10].

B. Optimization Techniques

Our initial C6416 implementation was based on language C on with Linear Assembly code. In this initial version, we achieved an execution time on the order of thirty micro-seconds for the necessary processing [4].

To guarantee the sufficiency of the time-consuming components for processing in real time, we profiled the execution of this algorithm with hand-optimized Assembly code. The principal techniques used to optimize our algorithm are explained in the following parts and resumed in figure 2.

1) Circular addressing

C6416 processor supports "circular addressing," which allows the processor to access a block of data sequentially and then automatically it wraps around to the beginning address [5], exactly the pattern used to access coefficients in CWT filtering. By circular addressing, the memory addressing cycle changes its starting point [7].

In fact, at real-time processing, for each sample input placed at its specified level in the "N" sized buffer (shown in figure 2), one is obliged to begin reading the buffer by the level where sample input is placed. Here, a problem arises with linear addressing. We can avoid that by employing the circular addressing feature of the C6416 in our assembly code.

2) Register Files Partition

Knowing that in C6416, there are two general-purpose register files data paths (A and B), each of these files contains 32 (32-bit) registers (A0-A31 for file A and B0-B31 for file B). The general-purpose registers could be used for data, data address pointers, or condition registers [10]. The processing of instructions occurs in each of the two data paths (A and B). So, we can optimize the time consumption to 50% of the total time processing.

3) Fast Multipliers

The CWT filter is mathematically based on $\sum X * Psi$, where X is a vector of input data, and Psi is a vector of filter coefficients. For each "tap" of the filter, a data sample is multiplied by a filter coefficient, with the result added to a running sum for all of the taps [4].

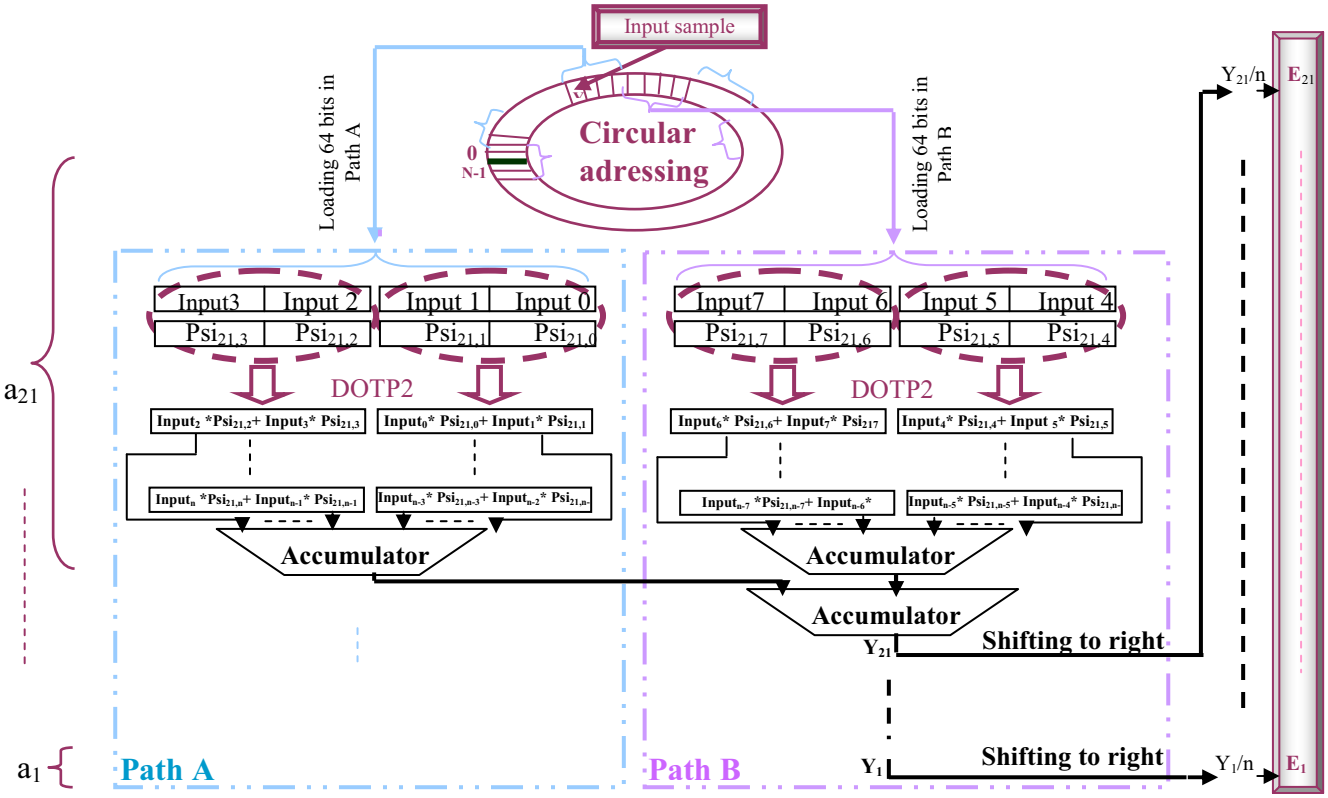


Figure 2. Architecture of code optimization algorithm of Continuous wavelet transform based strategy

Hence, the main component of the CWT filter algorithm is a dot product: multiply and add, multiply and add. These operations are not unique to the CWT filter algorithm. In fact, multiplication (often combined with accumulation of products) is one of the most common operations performed in signal processing [6].

So, in our algorithm, for each of the two concerned buffers (input and transfer function), a 32-bit fixed-point data can be loaded through the same packet and passed to the step of multiplication. In this step, Dot Product, Signed Packed 16-Bit (DOTP2) [7] instruction that returns the dot-product between two pairs of signed, packed 16-bit values, takes an important role. In fact, the product of the lower half-words of the two loaded data is added to the product of the upper half-words and then accumulated to the concerning destination [10]. This process is repeated another and another time with the two channels A and B in order to reach the register that includes the latest packets.

4) Enhanced Loading Data

DSP processor architects who want to improve performance beyond the gains afforded by faster clock speeds and modest hardware improvements must find a way to get significantly more useful DSP work out of every clock cycle. One approach is to extend conventional DSP architectures by adding parallel execution units, typically a second multiplier and adder.

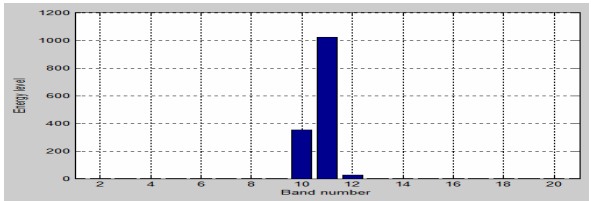
Knowing that the C6416 register file extends by supporting packed 64-bit fixed-point data types, the use of input data loading with Non-Aligned Double Word (LDNDW) helps us to optimize our algorithm. In this respect, we achieved a speedup of four times in terms of the total number of instructions executed in each path A and B [10]. With this increased parallelism, enhanced-conventional DSP processors can execute significantly much work per clock cycle. For example: two DOTP2 per cycle instead of one in each path, so four DOTP2 per cycle.

IV. Results and Discussions

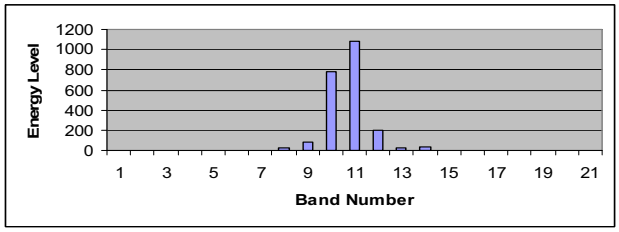
A. Parameterization Results

To validate the implementation of our hand-optimized algorithm, we choose to generate some harmonic signals with Matlab that makes it possible to activate and test not only each electrode alone but also real speech signals extracted from TIMIT database.

Figure 3 represents the parameterization result of a harmonic at 1.3 kHz with (a) Matlab implementation, (b) hand-optimized based TMSC6416 implementation.



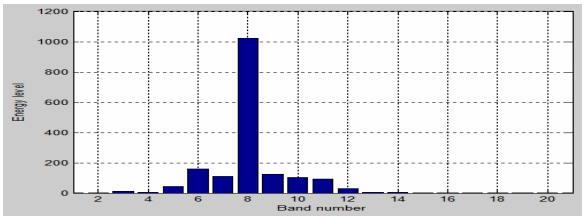
(a)



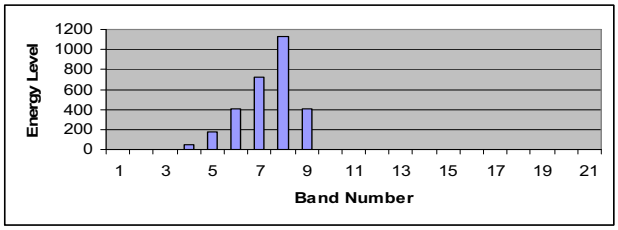
(b)

Figure 3. A 1.3 kHz harmonic parameterizations with (a) Matlab implementation, (b) hand-optimized based TMSC6416 implementation.

Figure 4 represents the parameterization result of a phoneme “aa” extracted from TIMIT database with (a) Matlab implementation, (b) hand-optimized based TMSC6416 implementation.



(a)



(b)

Figure 4. Phoneme “aa” extracted from TIMIT parameterizations with (a) Matlab implementation, (b) hand-optimized based TMSC6416 implementation.

As for the Matlab implementation results, the experimental results of this hand-optimized algorithm were satisfactory. The cochlear filter bank makes it possible to locate the harmonic in its corresponding frequency. Usually, harmonic energies are located in their corresponding frequency. The minor difference is caused by the fixed edge of input and output data necessary in C6416 based implementation (fixed point).

Moreover, some experiments show that various analyzed phonemes keep their frequencies positions. The maxima of the parameters keep the same frequency position for each phoneme. For example, in figure 5, we notice, that the phoneme “aa” guards its maximum in the eighth band for the two implementation types: Matlab implementation and final version of TMSC6416 implementation.

B. Processing Times

In order to prove the advantage of each optimization step of the CWT based, we calculated the necessary computing time related to the simulation results obtained with TMSC6416 implementation. Figure 7 shows the

results time consumption obtained for different optimization steps. The compiler optimizer is capable to reduce the processing time from 47.28 to 40.60 µsec just by using built-in optimizations supplied by –o0, -o1, -o2 and -o3 compiler options. Added to that, the Linear Assembly Decoder required 32.50 µsec of total processing time. Finally, with hand-optimization techniques used in our algorithm, we have been able to reduce the processing time to approximately 10.49µs

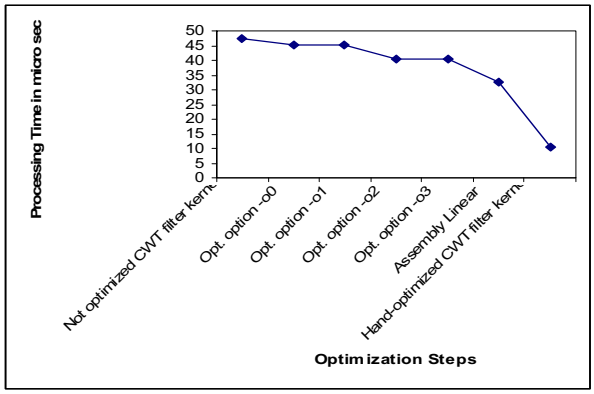


Figure 7. Performance C6416 based Optimizer Assembly Decoder implementation

In summary, the design conversion resulted in the following performance: The hand-optimized Assembly Decoder based C6416 implementation required 10.49 µsec of total processing time (this includes all of the outputs). This enhancement equals 77.81% of the original DSP processing which makes a better processing performance.

V. CONCLUSIONS

In this paper, we are interested in cochlear implant research which is actually a developing field regrouping researchers in different disciplines.

We aim to achieving a real-time implementation on DSP of auditory model based strategy using continuous wavelet transform speech processing. We presented different optimization steps that can be used to optimize any signal processing algorithm for real-time execution on DSPs. In order to validate the code hand-optimized algorithm of our strategy in real time, we have chosen to generate some harmonic signals with Matlab and to extract some real speech signals from TIMIT database.

This current implementation meets the real-time processing requirements. The time consumed with optimizer Assembly code is 10.49 µs, so with 16 KHz as sampling frequency. Thus, only occupied 21% of the processing time DSP was occupied.

REFERENCES

[1] National Institutes of Health, Cochlear Implants in Adults and Children, NIH Consensus Statement Online, 1995.

[2] P. Loizou, “Mimicking the human ear,” IEEE Signal Processing Magazine, vol. 15, pp. 101-130, 1998.

[3] V. Hamacher, J. Chalupper, J. Eggers, U. Kornagel, H. Puder, and U. Rass, “Signal Processing in High-End Hearing Aids: State of the Art, Challenges, and Future Trends,” Hindawi Publishing Corporation, EURASIP Journal on Applied Signal Processing,18, 2915–2929, 2005.

[4] A. Derbel, M. Ghorbel, M. Samet, and A. Hamida, “Implementation of strategy based on auditory model based wavelet transform speech processing on DSP dedicated to cochlear prosthesis,” in Proc. 4th Int. Symp.Comput. Intell. Intell. Inf., pp. 143–148, 2009.

[5] D.Amira, , K.Fathi, , B.Ahmed, and E.Noureddine, “Wavelet-Based Parametrisation of Speech Signal Dedicated to Cochlear Prosthesis”, Fourth International Multi-Conference on Systems, Signals and Devices, 2007.

[6] E. Jennifer and B. Jeff, “The Evolution of DSP Processors”, Berkeley Design Technology, Inc. (BDTI), 2000

[7] G.Farah, “Optimizing modems using code composer studio and TI resources”, Embedded Edge, 2001.

[8] B.H.Bonham and L.M.Litvak, “Current focusing and steering : Modeling, physiology, and psycophysics”, Hear. Res., vol 242, pp. 141-153, 2008.

[9] V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou, “A Recursive Wavelet-Based Strategy for Real-Time Cochlear Implant Speech Processing on PDA Platforms” IEEE transactions on Biomedical Engeneering, vol 57, pp. 2053–2063, 2009.

[10] Texas Instruments. Inc, “TMS320C64X User’s Guide”, Houston, TX, 1999.

[11] D. Amira, K. Fathi, B. Ahmed and S. Mounir, “Wavelet Filtering Based on Mellin Transform Dedicated to Cochlear Prostheses” 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC07), 2007.

[12] N.Tönder, R.Hartmann, and R.Klinke, “A Versatile system for the generation and the development of speech coding strategies in cochlear implants” IEEE Transactions on Biomedical Engineering, vol 45, pp.773–782, 1998.

[13] O. Rioul, .P Duhmel, “Fast Algorithm to discrete and continous Wavelet transforms”, IEEE Transaction on signal processing, vol 38, pp. 569-586, 1992.