

# An Approach to Sparse Reconfigurable Hardware for LPC of Speech for VoIP

Edet Bijoy K, Shaniba Asmi P

## ABSTRACT

Linear Prediction Coding (LPC) plays a vital role in speech communication. LPC algorithms are most commonly used for voice coders. In this paper we present a new approach to implement a reconfigurable hardware for sparse LPC algorithms for VoIP applications. The motivation behind this is that the sparser the feature the better would be the bit rate constraint. The necessity of a reconfigurable system is scalability and transcoding. The computational cost is expected to be very low for such a hardware which is capable of solving least square problems.

## Keywords

Linear Prediction Coding, Reconfigurable Computing, VoIP.

## 1. INTRODUCTION

As the speech communication services provided on VoIP are now becoming more popular the new generation of such services requires new speech coding algorithms designed with emphasis more on wider frequency band width lower delay and lower complexity. The coder G.711.1 was designed to handle wideband speech when compared to G.711 which is limited to narrowband speech. The necessity of a reconfigurable system is scalability and transcoding. Many wideband speech coding standards were established without scalability with G.711. Also transcoding between codecs of two different bit streams is much more complex in computation. Also the quality degradation is also very important which cannot be ignored.

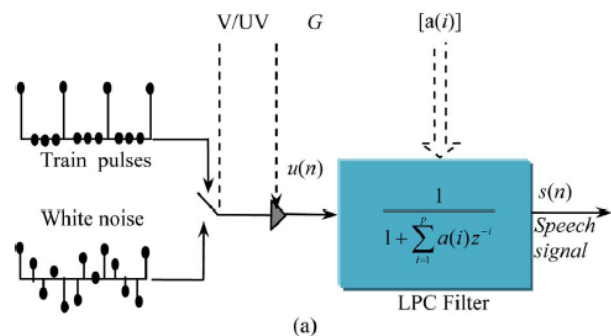
The Voice over Internet Protocol (VoIP) has revolutionized the point to point communication in this era. The idea of coding the speech segments into data packets can be counted as a technology of the decade. The speech communication have seen dynamic transitions through the traditional telephony known as Plain old telephony service, to what we see as the internet telephony [1]. Each and every transmission calls for modification in the processing strategy of the speech signals. The height of this can be seen when we begin to see the speech signals as packets. The crucial part of the processing is the design of the speech coders also known as vocoders which should least compromise on parameters like efficiency, intelligibility, communication delay, computational complexity, power consumption and robustness to packet loss and noise [1]. The quality in general is measured as factor of Mean-Opinion Score (MOS) ranging from bad to excellent.

The VoIP system currently uses the internet Low Bit-rate Codec (iLBC), which eliminates the dependency on previous frames because each packet is treated independently to achieve a better response to packet loss, delay or packet jitter. It is suitable for real time

communications such as telephony and video conferencing, streaming audio, archival, and messaging. It is commonly used for applications such as Skype, Yahoo Messenger and Google Talk [2].

## 2. HARDWARE IMPLEMENTATION OF LPC ALGORITHM

In this section we discuss about the hardware implementation of LPC algorithm for speech coding application. To process the speech signals the production of speech is very important. The human speech production system is mathematically modeled as given in fig 1.



Vocal tract	⇔	H(Z) (LPC Filter)
Air	⇔	u(n) (Innovation)
Vocal Cord	⇔	V(voiced)
Vocal Cord	⇔	T (pitch period)
Fricatives	⇔	UV(unvoiced)
Air Volume	⇔	u(n) (Innovation)

Fig 1, LPC Model

The relation of input and output is given by

$$s(z) = \frac{G \cdot u(z)}{A(z)}$$

The human speech is produced in the vocal tract as a combination of vocal cords (in glottis) interacting with articulators. It can be modeled by the passage of an excitation signal through a filter. This model is called linear predictive coding (LPC). The LPC “Linear Predictive Coding” algorithm is a widely used technique for voice coder. Applications such as cell phones, hearing aids, and digital audio devices have constraints on area, speed and power consumption.

This implementation can be addressed by Systems on Programmable Chip (SoPC). Modern Field Programmable Gate Arrays (FPGAs) contain many resources that support DSP applications such as embedded multipliers, Multiply Accumulate (MAC) units and processor cores. Implementation approach is based on the benefits to implement the blocks on hardware rather than software and also on the reusability of blocks in other applications.

The LPC block input is assumed to be a 16 bits PCM signal. Two pre-processing functions are applied before the encoding process signal scaling, and high-pass filtering. The scaling consists of dividing the input by a factor 2 to reduce the

possibility of overflows in the fixed-point implementation. The high-pass filter serves as a precaution against undesired low-frequency components. The Linear Prediction (LP) is performed once per speech frame using the autocorrelation method with a 30ms asymmetric window. Every 80 samples (10ms), the autocorrelation coefficients of windowed speech are computed and converted to the LP coefficients using the Levinson algorithm [6].

The architecture makes use of an autocorrelation core that computes the autocorrelation coefficients and windowing core in order to reduce the generation of side lobes in the frequency spectrum. For windowing two blocks are implemented: multiplier block and a 256 word 16 bit ROM as shown in Figure 2.

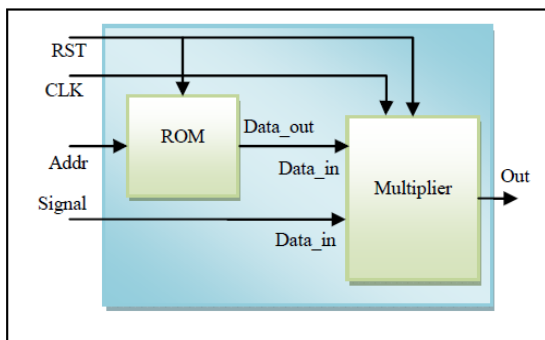


Fig 2. Windowing Block Architecture.

Xilinx's CORE generator has been used to produce the simple arithmetic operators and memory required by this windowing block in order to reduce the development time. The CORE generator can be used to produce devices ranging in complexity from simple arithmetic operators and delay elements to complex building blocks such as digital signal decoders, processing filters, multiplexers, transformers, FIFOs, and memories. The final system built around the windowing core and the autocorrelation computation which was implemented using the Spartan 3 Xilinx board is shown in figure 3.

The RTL specification is made compatible with most of FPGA design tools for an efficient component reuse. The specification can be parameterized according to the window size and the data width (word length). It is also parameterized according to the value of index  $k$  of the  $r(k)$  autocorrelation coefficient equation.

The results of pure hardware implementation have significant speedup and less resource utilization over the software implementation and rather HW/SW version.

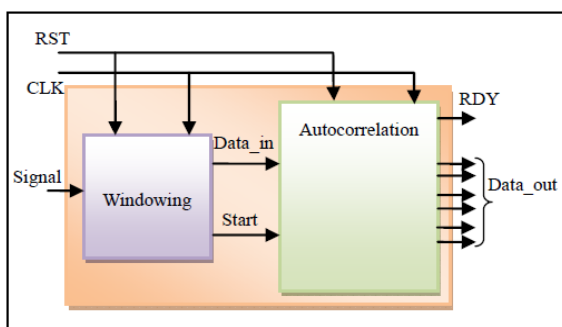


Fig 3. Final hardware architecture.

### 3. RECONFIGURABLE PERIPHERAL PROCESSOR FOR SPEECH

FPGA co-processor can realize function of peripheral processor, implementing hardware platform interface with specific external equipment in audio signal processing tasks. The DISC (Dynamic Instruction Set Computer) Core Processor in figure 4 presents a static component of the system. The main tasks of system are managing I/O operations, initiating of application specific instructions, and providing the reconfiguration sequences in so-called Instruction Space [5].

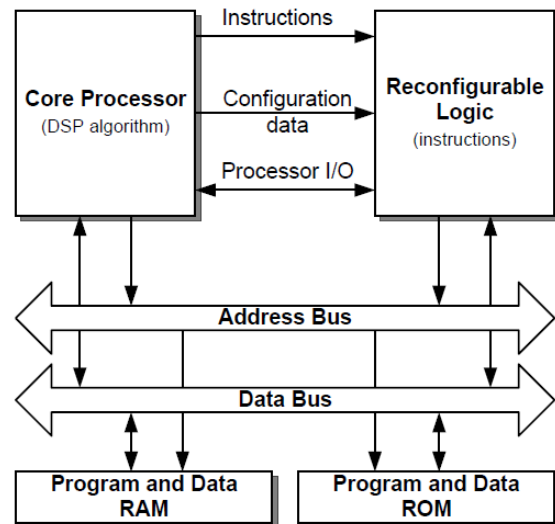


Fig 4. DISC-processor structure

The instruction space is a dynamic part of the system usually built on reconfigurable FPGA(s). Configurations implementing the circuitry needed by the DSP algorithm are being loaded into the instruction space. These predefined configurations (instructions from Core Processor) individually respond to the initiation signals that depend on the DSP algorithm that being implemented by the processor. This DISC-approach in various DSP applications boosts the performance of the currently used algorithm. The structure of the DISC-processor remains unchanged, that yields a significant flexibility particularly suited for embedded processors.

The peripheral processor functions are realized by FPGA Xilinx Spartan-3 XC3S200. The peripheral processor meant for hardware platform external interface realization, pre- and post-data processing, supplying with necessary interfaces data formats, external interface packets forming and disassembling as well as digital signal processing algorithms implementation with the object of DSP unload. The external memory and peripheral processor (FPGA) are connected to the TMS320C6713 through the 32-Bit EMIF. The TLV320AIC23 codec is used for stereo audio signal input and output purposes. It is wired to the DSP via two serial channels: McBSP0 (control channel), McBSP1 (receive/transmit channel). The synchronize clock frequency can be adjusted by the Digital Clock Manager (DCM) blocks (FPGA) as in figure 5. For system external interface implementation, FPGA's I/O ports are used. The Spartan-3 family FPGAs supports 26 I/O standards 8 of which are differential (LVDS, LVPECL etc.). The I/O ports can be used as one or several parallel data buses of necessary capacity

as well as serial lines. In addition to logic cells, FPGA includes triggers, memory blocks, matrix multipliers and phased-locked loop schemes. Twelve matrix multipliers blocks with 18-bits operands and 36-bits result can be used to implement digital signal processing functions on the peripheral processor. Realization of peripheral processor project can be done with the help of CAD Xilinx ISE® . In this case, FPGA configuration sequence is generated by means of iMPACT utility and loaded through JTAG interface into the platform's flash-memory. Every time after power up FPGA reads this sequence from flash-memory.

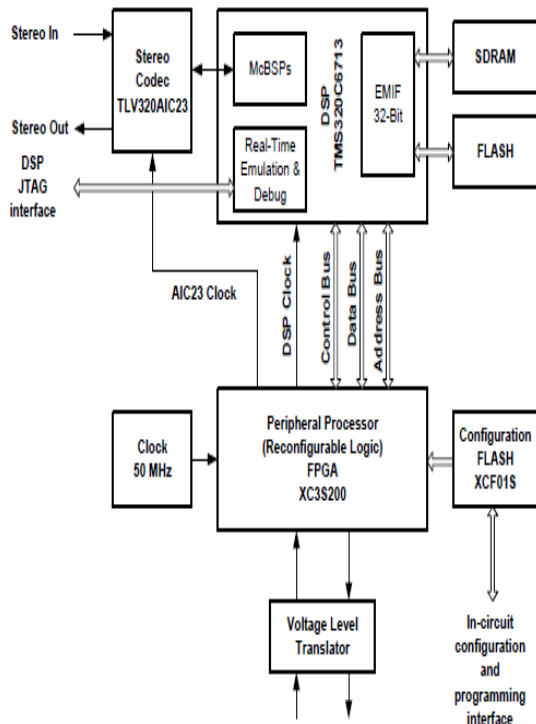


Fig 5. Module structure

In future peripheral processor can be implemented with reconfigurable structure of arithmetical and logical units to meet the requirements of the current algorithm. Therefore, overall performance and flexibility of the system can be significantly improved.

#### 4. SPARSE LINEAR PREDICTIONS

In this section we discuss on the problem of determining short-term predictor. As the speech samples  $x(n)$  is represented as linear combination of  $K$  past samples:

$$x(n) = \sum_{k=1}^K a_k x(n-k) + r(n)$$

Where  $\{a_k\}$  are the prediction coefficients and  $r(n)$  is the prediction error. The optimization problem is defined to find the prediction coefficient vector  $a \in \mathcal{R}^K$  from the set of observed real samples  $x(n)$  for  $n = 1, \dots, N$ , in matrix form:

$$x = Xa + r$$

The problem becomes

$$a = \arg \min_a \|x - Xa\|_p^p + \gamma \|a\|_k^k$$

Where

$$x = \begin{bmatrix} x(N_1) \\ \vdots \\ x(N_2) \end{bmatrix}, X = \begin{bmatrix} x(N_1-1) & \cdots & x(N_1-K) \\ \vdots & & \vdots \\ x(N_2-1) & \cdots & x(N_2-K) \end{bmatrix}$$

The prediction coefficient vector  $a$  is found such that the resulting residual is sparse. The optimization problem is defined by setting  $p=1$  and  $\gamma=0$ , then

$$\min_a \|x - Xa\|_1$$

The compressive sensing formulation is particularly relevant in sparse recovery problems. By introducing sparsity in residual it is reasonable assume that only a small portion of the residual samples are sufficient to reconstruct the speech signal with high accuracy.

#### 5. SPARSE HARDWARE

In normal LPC analysis the hardware components used are for windowing and auto correlation functions which require the basic Multiplier and Accumulator (MAC). This can be done easily by using a DSP architecture or high end FPGAs with inbuilt DSPs. But in sparse implementation of LPC the basic design consideration is the sparse matrix.

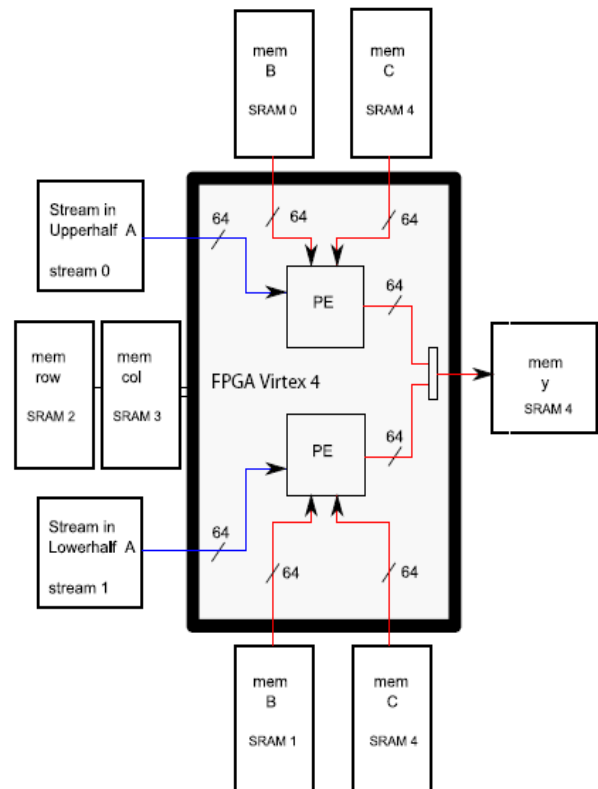


Fig 6. Organization of Memories for SGI RASC Implementation.

Georgi Kuzmanov of Delft University of Technology have designed a new hardware for the implementations of compressive sensing applications [7]. The memory organization of the system is shown in figure 6. The design is basically a sparse/dense Matrix-Vector Multiplier which supports double precision floating-point matrix-vector multiplication. The design is composed of multiple processing elements and is optimized for FPGAs.

Considering the matrix vector product  $c = Ab$ , with dimensions of  $\mathbf{A} M \times N$ ,  $\mathbf{b}$  and  $\mathbf{c} N \times 1$  the memory  $\mathbf{A}$  is used to store the row elements of matrix  $\mathbf{A}$ , memory  $\mathbf{B}$  is used to store the elements of vector  $\mathbf{b}$  and memory  $\mathbf{C}$  is used for the initial values of vector  $\mathbf{c}$ . This architecture well suits for the prediction of coefficients in sparse Linear Prediction.

## 6. CONCLUSION

As the speech communication is the widely used technology in the world today, the challenges that researches overcome is the low bit rate transmission of speech packets through VoIP. Many hardware is been implemented for various speech coding algorithms including LPC. As sparse linear predictions outperform the normal LPC it is very essential to implement reconfigurable hardware for sparse LPC considering the scalability. Thus we have discussed about a new approach that could be implemented for sparse linear prediction which could be capable for solving the least squares problems without much complexity and in low cost.

## 7. REFERENCES

- [1] Tokunbo Ogunfunmi and Madihally J. (Sim) Narasimha "Speech over VoIP Networks: Advanced Signal Processing and System Implementation", 22 May 2012 IEEE CIRCUITS AND SYSTEMS MAGAZINE pp.35 – 55, 2012.
- [2] D. Giacobello, M. G. Christensen, M. N. Murthi, S. Jensen and M. Moonen, "Sparse linear prediction and its applications to speech processing," Audio, Speech, and Language Processing, IEEE Transactions, 2010.
- [3] Russel Tessier, Wayne Burleson, "Reconfigurable Computing for Digital Signal Processing: A Survey", Journal of VLSI Signal Processing 28, 7-27, 2001.
- [4] J. Makhoul, "Linear prediction: A tutorial review," Proceedings of the IEEE, vol. 63, pp. 561-580, 1975.
- [5] Michael Livshitz, Alexey Petrovsky, Andrey Stankevich, Mikhail Kachinshy, Alexander Petrovsky, "Embedded Reconfigurable Hardware Platform for Real-Time Audio and Speech Signal Processing", The Fifth International Conference on Neural Networks and Artificial Intelligence 2010 May 27-30, Minsk, Belarus.
- [6] Mohamed Atri, Fatma Sayadi, Wajdi Elhamzi, Rached Tourki, "Efficient Hardware/Software Implementation of LPC Algorithm in Speech Coding Applications", Journal of Signal and Information Processing, 2012, 3, 122-129
- [7] Kuzmanov, Georgi, and Mottaqiallah Taouil. "Reconfigurable sparse/dense matrix-vector multiplier." Field-Programmable Technology, 2009. FPT 2009. International Conference on. IEEE, 2009.