ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 7, July 2021

Design and Implementation of Sinusoidal Model Based Codec2 Speech Decoder

[1] Sowjanya.K, [2]Dr. Kalpana.A.B
[1] Post graduate student, Dept of ECE, Bangalore Institute of Technology, Bangalore, India
[2] Associate Profesor,Dept of ECE, Bangalore Institute of Technology, Bangalore, India

Abstract: There are different optimization criteria in the design and development of speech compression algorithms in that Codec2 vocoder algorithm is one of an efficient sinusoidal coding with very high compression rate down to 450 bit/s. In this paper, to increase the performance of voice decoding process, efficient hardware architecture of Codec2 low bit-rate speech decoder is designed. The decoding processes of codec2 is based on sinusoidal model .Although the sinusoidal decoding algorithm is complicated with many arithmetic operations such as the arithmetic of complex numbers, FFT, FIR filter, division, trigonometry, exponential and logarithm functions, several techniques were explored to optimize and parallelize a data path of the proposed hardware.

The proposed hardware implementation is done using verilog (hardware description language) and synthesized on Xilinx Artix-7 FPGA.

Index Terms— speech compression algorithm, vocoder, sinusoidal coding, codec2, Linear Prediction Coding.

1. INTRODUCTION

Human speech is a common signal that is transmitted in a variety of applications in radio communications. Depending on the compression of the speech signal, there are trade-offs for quality of speech versus the associated communication signal size. For example, signal size can be quantified in terms such as bits per sample and sampling rate in a digital communication sense. Larger signals require higher capacity communication links, more time for transmission, more power consumption to transmit the signal, and more memory capacity at both the transmission and receiving nodes. Compressing a signal reduces many of these factors but comes with a computational cost to perform the signal processing, a number of audio compression codecs (Coder and Decoder) have been created for this purpose, and the focus of this work is on Codec2.

Linear predicting coding LPC-10 is vocoder developed by one of the early standards [1] that is US Department of Defense. Later some speech coding algorithms are developed such as: CELP, MELP and Codec2. CELP employ long term and short term liner prediction model to improve the encoding processes.[2]. Later MELP created to solve the problems of speech coding [3]. This algorithm features less complexity than the CELP algorithm with comparable output speech quality at half 2.4 Kbps[4].

Codec2 is an open-ended and license-free and audio codec that operates at a squashed lowest bit rate [5]. The current implementation can be run on microprocessors [6] and is in C. The key interest of this core is that it has been advanced without any existing commercial ties, and researchers and developers can use this codec for any appeal without restriction. This paper implements a RTL version of Codec2 on a Field Programmable Gate Array using Verilog HDL. The interrogation for this exercise are to see what trade-offs in terms of reckoning speed and quality a hardware execution of the Codec2 decoder on FPGA results in compared to a software implementation running on a microprocessor. Hypothesis is that a hardware execution can be faster than a software execution on a microprocessor due to custom parallel implementation capabilities possible on FPGA. Also, the formation of this hardware core will allow future chip designers to test and implement Codec2 on other FPGAs and even ASIC versions of this core (if the market demands ever is high enough to justify the high cost of manufacturing ASICs). To test the hypothesis, it is implemented and tested a Codec2 decoder in Verilog and mapped it to Artix 7 FPGA environment. It is observe and report the speed and area utilization of this execution and compare it to the Codec2 running on a Raspberry Pi, counting its ARM processor.

To digitalize speech or audio signal in form of binary bits, codec2 algorithm is used and mean while

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 7, July 2021

maintaining the quality of signal. Codec 2 is a low bit rate speech coding algorithm and is based on sinusoidal coding [7]. Specialized for human speech. Bit rate of 3200 to 450bit/s is successfully created [8].

Main aim of the project is to design hardware architecture of codedc2 speech decoder with the parallelism and pipelining of speech signal processing. Codec2 decoder is proposed to increase the performance of voice decoding process and reduce comprehensive tasks from a host processor

The audio is regenerated by modeling oration as sum of sine wave with unconventional of amplitude called LS pairs, on the top of set on fundamental frequency of speaker voice.LS pairs (Linear spectrum pairs) are used to represent LPC coefficients for transmission over the channel. The LSP coefficients shows the Linear Predictive coding (LPC) model in the frequency domain.

The technique of speech coding, used to represent audio signal processing or speech processing in spectral envelope form of digital signal of speech in compressed form is know as LPC.

linear predictive coding

LPC is used in audio and speech signal processing and it is represented by spectral envelop (envelop of amplitude of spectrum means it describe one point at a time) of digital signal of speech in compressed form. Linear prediction is a mathematical operation where future values of a discrete-time signal are estimated as a linear function of previous samples. Linear Predictive Coding (LPC) is one of the speech analysis techniques, and is one of the useful methods for encipher good quality speech at a low bit rate. It provides accurate speech parameters, and is relatively efficient for computation. The most important thing of LPC is the linear predictive filter which determine the value of the next sample by previous samples of linear combination In Linear Predictive Coding (LPC) analysis the short-time spectrum and it can be represent based on deriving linear prediction coefficients. LPC analysis is an effective method to estimate the main parameters of speech signals. LPC starts with the assumption that in speech signal buzzer is produced at the end of a tube for voiced sounds, and added hissing and popping sounds for voiceless sounds. The glottis (the space between the vocal folds) produces the buzz, which is characterized by its intensity (loudness) and frequency (pitch). The speech signals vary with time, this process is done on short chunks of the speech signal, which are called frames; generally, 30 to 50 frames per second give an speech with good compression.

The Lpc Coding Function

As mentioned above the LPC coding function will take the speech audio signal and divide it info 30mSec frames. These frames start every 20mSec. Thus each frame overlaps with the previous and next frame. Shown in the figure below:

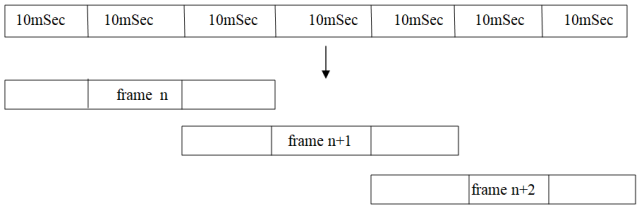

Fig 1: Frame overlaps with the previous and next frame

After the frames have been separated, the LPC function will take every frame and extract the necessary information from it. This is the voiced/unvoiced, gain, pitch, and filter coefficients information. To determine if the frame is voiced or unvoiced we need to find out if the frame has a dominant frequency. If it does, the frame is voiced. If there is no dominant frequency the frame is unvoiced. If the frame is voiced we can find the pitch. The pitch of an unvoiced frame is 0. The pitch of a voiced frame is in fact the dominant frequency in that frame. One way of finding the pitch is to cross correlate the frame. This will strengthen the dominant frequency components and cancel out most of the weaker ones. If the 2 biggest data point magnitudes are within a 100 times of each other, it means that there is some repetition and the distance between these two data points is the pitch.

Linear Spectral Pairs

Linear spectral pairs used represent the coefficients of the LPC To improve the performance of the low bit rate speech coder, a transformation of LPC parameters to more efficient representation know as Line Spectrum Pairs (LSP) or Line Spectral Frequencies (LSF) is used. LSP is the mathematic model which is introduced by Itakura as an alternative representation to LPC parameters. The basic idea of the linear prediction parameters is that the next sample speech signal can be predicted by a linear combination of the past values of the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 7, July 2021

sample signal at time. Real vocal has both poles and zeros. By this reason, LSP technique is map the A (z) into other equivalent polynomials to represent the all pole and zero in that transfer function. A(z) is decomposed into both symmetric and an asymmetric polynomial by adding and subtracting the time reversed system function as follows: Linear predictive is polynomial

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$$

Can be expressed in terms of

$$A(z) = 0.5[P(z) + Q(z)]$$

Where:

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1})$$
$$P(z) = A(z) + z^{-(p+1)}A(z^{-1})$$

P is a symmetric polynomial and Q an antisymmetric polynomial; physically P (z) corresponds to the vocal tract with the glottis closed (the space between vocal folds which produces the vowels and voiced constraints) and Q(z) with glottis opened. There are three important and interesting properties of P(z) and Q(z) listed as follows:

1. All zeros of P(z) and Q(z) are on the unit circle.
2. Zeros of P(z) and Q(z) are interlaced with each other
3. Minimum phase property of A (z) P is easily preserved after quantization of zeros of P(z) and Q(z).

The first two properties are useful for finding the zeros of P(z) and Q(z) and the third property ensures the stability of the synthesis filter. Since zeros of P(z) and Q(z) are on the unit circle, they can be expressed as e^{jw} and hence ω's are called line spectrum frequencies (LSF). Lines spectrum pairs (LSP) can be represent the Linear prediction coefficients (LPC) since they can give better interpolation properties and robustness to quantization noise.

Two IIR filters can be implemented to retrieve the LPC from LSP samples as shown in Fig.2.Ap(n) and Aq(n) are the input samples for LSP

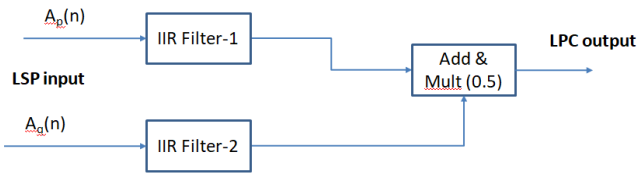

Fig 2 : LSP to LPC conversion

IIR Filter is as shown in the figure(3) Delay block is implemented using D – flip flop .For Full adder & Subtract modules, Ripple carry adder based adder & subtractor are implemented

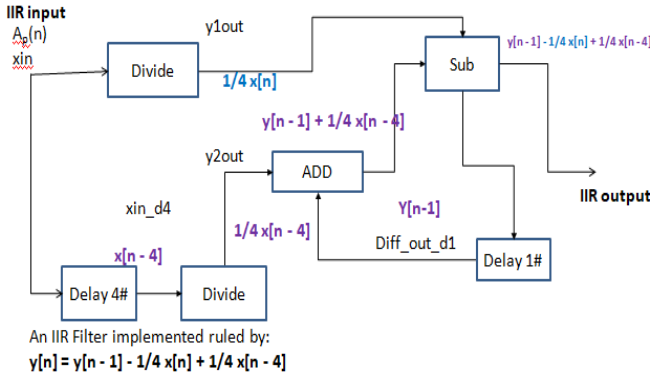

Fig 3: Block Diagram of IIR filter.

Audio compression

Audio compression techniques attempt to shrink the signal size via lossy or lossless compression techniques. Lossless compression techniques find redundant data in the encoding of the signal at the transmission side and are completely recoverable at the receiver side. A well-known algorithm in this domain that can be applied to digital data is the Lempel-Ziv algorithm [9]. Lossy compression, on the other hand, attempts to remove data that is less useful, and thus makes the signal less like the original signal, but the signal contains most of the relevant information. For example, an original speech signal includes characteristics that would allow a listener to identify who the speaker is based on the acoustics, but lossy compression can remove these qualities to only transmit the information, also known as the words spoken. A well-known algorithm for lossy compression is MP3 encoding for music.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 7, July 2021

codec2 speech decoder implimentation

Codec2 uses the sinusoidal property of the speech signals. By considering the fundamental frequency of sinusoidal nature, LPC coefficients are generated for the Lines spectrum pairs .The LPC is given to FFT core to transform the LPC into frequency transformation.The FFT outputs are passed to an FIR filter to remove any unwanted harmonics. Filtered power spectrum is multiplied with Encoded speech samples.The multiplied version of speech signal with power spectrum is passed to IFFT (Inverse FFT) block to reproduce the raw speech signals. Below Figure(4) show block diagram of codec2 speech decoder.

A Fast fourier transform

Fourier transform is used to convert a signal, which is in the time or space domain into its spectrum in the frequency domain. The time and frequency domains are just alternative ways of representing signals and the Fourier transform is the mathematical relationship between these two representations. Discrete Fourier Transform (DFT) is similar like Fourier transform and used with digitized signals. The Fast Fourier Transform (FFT) is an implementation of the DFT which produces almost the same results as the DFT, and the main difference between these two is FFT is more efficient and much faster which often reduces the computation time significantly. FFT can reduce the Number of multiplication and addition so it is faster than DFT.

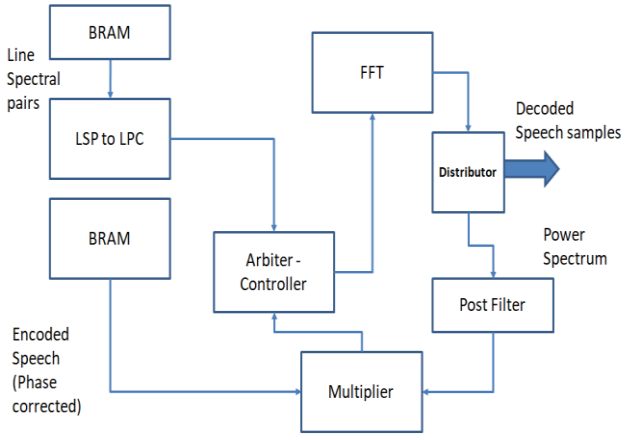

Fig 4: Block diagram of codec2 speech decoder

A DFT can be performed as in order of O (N^2), whereas FFT reduces the time complexity in the order of O (NlogN). The output of the FFT is containing information about the frequency content of the signal. The magnitude tells about the strength of the frequency components. The phase tells about how all the frequency components align in time. The DFT is type Fourier transform of discrete signal x(n) and transfer function is given by

$$X(w) = \sum_{n=-\infty}^{\infty} x(n)\, e^{jwn}$$

Assuming the recorded voice signal x(n) is a sequence which consists of complex values, such as x(n)=R+I, where R stands for the real part of the value, and I stands for the imaginary part of the value. Since the exponent factor is:

$$e^{jwn} = \cos(wn)+j.\sin(wn)$$

The module performs the FFT & IFFT based on the control signal. I (In phase) & Q (Quadrature phase) data are input to FFT module fft_mode is the control signal which decides the FFT or IFFT operation Radix 2 decomposition (butterfly) is used .as

The N- point DFT of an input sequence x[n] is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}, k = 0,1,,,,N-1$$

Where:

$$W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$$

Use the FFT IP from the Xilinx IP [10]. The IP provides the best performance in the FPGA platform. Same IP core can be used to perform FFT as well as Inverse FFT (IFFT). The core details are mentioned in the "Fast Fourier Transform v9.1 – Logic ore IP Product Guide" from Xilinx.

To use the FFT core, create a Vivado project with targeted FPGA platform. Once project is created, FFT core can be opened and used. In the Vivado project, click "IP catalogue" and search for FFT. In the search result, select "Fast Fourier Transform" IP.

B. Post Filter – FIR Filter

A Post filter removes the unwanted harmonics from the power spectrum got from the FFT of LPC coefficients.A standard FIR filter is implemented to act as a post filter.8 bit input is stored in 16 different registers. The 16:1 Mux selects the delayed versions one by one based on the counter based select lines. The Mux output is then multiplied with FIR filter coefficients. These

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 7, July 2021

coefficients are stored in a ROM.

The output from the multiplier is accumulated in the accumulator. The accumulator output will contain the filtered values. ROM_FIRcoeff is the Read only memory module implemented to hold the FIR coefficients. The contents are initialized with FIR coefficients. Based on the address, this module will output the stores coefficients.

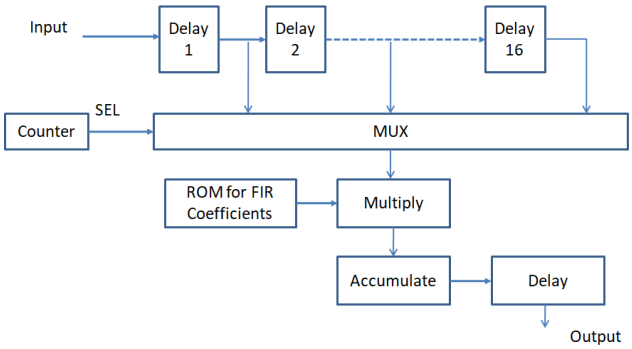

Fig 5: Block diagram of post filter.

C. Accumulator IP

The Accumulator module can generate adder-based, subtracter-based and adder/subtracter-based accumulators operating on signed or unsigned data. The function can be implemented in a single DSP slice or LUTs (but currently not a hybrid of both). Pipelining is available for both implementations[11].

D . *Multiplier IP*

Generates fixed-point parallel multipliers and constant-coefficient multipliers for two's complement signed or unsigned data. Supports inputs ranging from 1 to 64 bits wide and outputs ranging from 1 to 128 bits wide with any portion of the full product selectable. Configurable latency for all multiplier variants. Supports symmetric rounding to infinity when using the DSP Slice [12].

PERFORMANCE AND IMPLEMENTATION RESULTS

The proposed decoder was implemented on targeted Xilinx FPGA Artix-7 XC7A200T and simulated by taking the Encoded audio bit streams.

The optimized design is verified at each block level and finally the decoded /raw audio outputs are seen. It is shown in figure (6) and figure(7) shows floor plan for decoder.

Table1 shows resource utilization of the design. It shows that, the number of DSPs reduced. The size of circuits depends on look up table. LUTs are used around 1.5%.

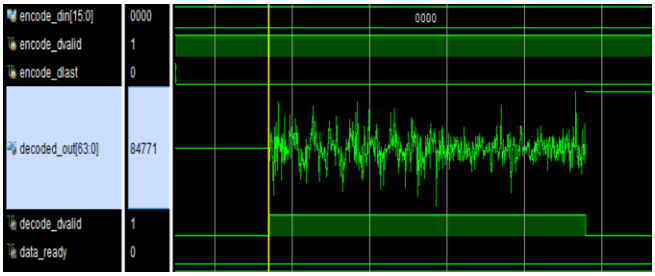

Fig 6; The decoded output waveform

TABLE 1 FPGA resource utilization

Resource	Utilization	Available	Utilization%
LUT	1978	133800	1.48
LUTRAM	168	46200	0.36
FF	3208	267600	1.20
BRAM	2	365	0.55
DSP	4	740	0.54
IO	51	285	17.89
BUFG	2	32	6.25

TABLE 2 Resource utilization comparsion

Resource	Reference design resources			First design resources			Final design resources		
	Utilization	Available	Utilization%	Utilization	Available	Utilization%	Utilization	Available	Utilization%
LUT	3158	63400	4.98%	2320	133800	1.73%	1978	133800	1.48%
FFs	627	126800	0.49%	3956	267600	1.48%	3208	267600	1.20%
BRAM	72,064 bit	4,976,640 bit	1.45%	49,152 bit	5,980,160 bit	0.82%	32,768 bit	5,980,160 bit	0.55%
DSPs	5	240	2.08%	6	740	0.81%	4	740	0.54%

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 7, July 2021

TABLE 3 Resource utilization – Module level

Name	LUT as Logic (134600)	LUT as Memory (46200)	LUT Flip Flop Pairs (134600)	Block RAM Tile (365)	DSPs (740)	E
audio_decode_top	1810	168	1192	2	4	
LSP_buffer (blk_mem...	0	0	0	0.5	0	
lsp_lpc_inst (lsp_lpc)	109	16	55	0	0	
post_filter (FIRtop)	196	0	52	0	0	
mult_overlapp (mult_o...	286	0	32	0	0	
vio_inst (vio_0)	330	0	241	0	0	
dbg_hub (dbg_hub_CV)	432	24	295	0	0	
FFT_inst (xfft_0)	428	128	480	1.5	4	

Table 1 shows over all resource utilization of the design. Table 3 shows resource utilization at module level

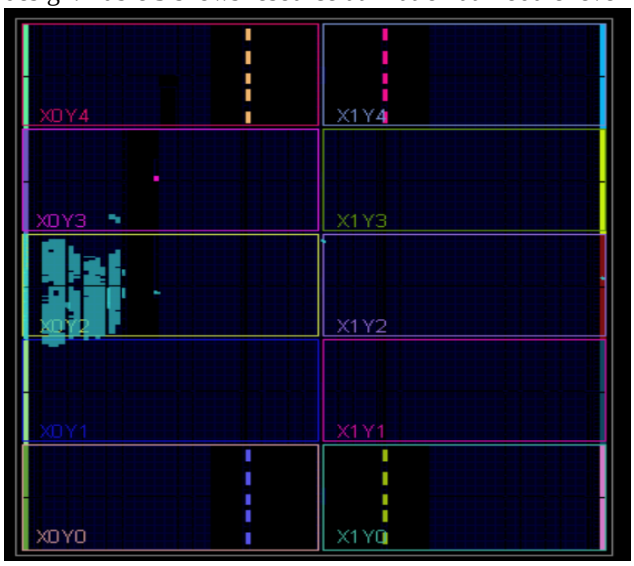

Fig 7: Implementation Results of decoder

The Design was compiled with positive slack for 100MHz clock constraint. Since the slack is positive, The Design can work more than 100MHz (10ns) speed. Results show that the maximum speed of clock is up to 131.99MH, This is better than the reference paper [13].

CONCLUSION

The design of codec2 decoder is capable of decoding speech signal with high performance, at Maximum clock frequency 131.99MHz. The efficient design of Codec2 speech decoder was developed by optimizing various dataflow of each processing step. Many hardware parallelisms and pipelining schedule are carefully organized to handle whole hardware resources with high utilization as much as possible.

Acknowledgment

I Would like to thank Dr. A.B. Kalpana Associate professor, department of electronics and communication engineering. BIT, the organization and my parent, friends. For the guidance and extended support for the work.

REFERENCES

1. J. Srinonchat, "New Technique to Reduce Bit Rate of LPC-10 Speech Coder," IEEE Region 10 Conference, Hong Kong, November 2006, pp.1-4
2. M. Sadek, S.E. Ramly, A. Tawfik, "A new CELP speech coder for wireless communication," 14th International Conference on Digital Signal Processing Proceedings, Greece, Vol.2, pp.997-1000
3. C. Wu, H. Jiang, B. Li, "An Improved MELP Speech Coder," International Conference on Information Technology and Computer Science, Ukraine, August 2009, Vol.2, pp.130-133
4. T.T. Teo, E.C. Tan, "Implementation of 2400 bps MELP vocoder on TMS320C44," Fourth International Conference on Signal Processing, China, October 1998, pp.576-579
5. D. Rowetel, "Codec2", http://www.rowetel.com/?page_id==452
6. D. Rowetel, "Codec2 Speech Coding Reference C Source Code," https://svn.code.sf.net/p/freetel/code/codec2/branches
7. D. Rowetel, "Techniques for Harmonic Sinusoidal Coding," Ph.D. Thesis, 1997
8. https://www.tapr.org/pdf/DCC2011-Codec2- VK5DGR.pdf
9. W.C. Chu, "Speech Coding Algorithms Foundation and Evolution of Standardized Coders," Wiley-Interscience, 2003
10. https://www.xilinx.com/support/documentation/ip_documentation/xfft/v9_0/pg109-xfft.pdf
11. https://www.xilinx.com/support/documentation/ip_documentation/accum/v12_0/pg119-c-accum.pdf
12. https://www.xilinx.com/support/documentation/ip_documentation/mult_gen/v12_0/pg1

08-mult-gen.pdf

13. An Efficient Hardware Architecture of Codec2 Low Bit-rate Speech Decoder | IEEE Conference Publication | IEEE Xplore