

Porting and Optimization of ITU-T G.729.1 codec on SC3850 DSP core

Reshmi S¹, Sreenesh Shasidharan²

¹(M.Tech Student, Department of Electronics & Communication,
Federal Institute of Science & Technology, Ankamaly)

²(Assistant Professor, Department of Electronics & Communication,
Federal Institute of Science & Technology, Ankamaly)

Abstract—This paper presents a methodology towards the details of optimization for the real-time implementation of ITU's G.729.1 codec on SC3850 DSP core running at 1 GHz clock speed. The main aim is the optimized fixed point implementation of G.729.1 speech coding algorithm on SC3850 DSP core. Optimizations were done in C language to reduce the execution speed of the codec by exploiting the architectural features of the target device. First the reference code provided by ITU-T's published documents was modified to meet the requirements in C language. Next by exploiting the features of SC3850 core, the critical parts of our code that consumes more cycles were modified in order to reduce their execution time. In each stage, the correctness of the implementation was verified by testing our codes against testing vectors provided by ITU-T. More than 90 percent improvement over the baseline codec was achieved by the systematic implementation of these optimization techniques for G.729.1 codec on SC3850 core.

Keywords: *bitrate, G.729.1 codec, MCPS, optimization, SC3850*

I. INTRODUCTION

Nowadays, the speech communication services demanded by the current tech-savvy globe are ever-increasing. The numbers of requests for services augment with number of users. Hence the most straight forward way for the designers of communication systems to fulfill this increasing need is providing transmission canals with lower bandwidth to each user. Compression is the most efficient technique for that. The objective of speech compression is to represent a digitized speech signal using reduced bit-rate as possible so that the reconstructed speech signal maintains an optimum level of perceptual quality.

VoIP is nowadays facing new challenges in terms of quality of service and efficiency in networks. Wideband audio codecs are extensively used to meet these challenges by allowing a flat transition of voice quality from narrowband (300-3400 Hz) to wideband (50-7000 Hz). They are capable of providing most lifelike conversations with higher level of fidelity and

quality by exploiting the full range of human speech. G.729.1 codec- an 8-32 Kbit/s scalable wideband speech and audio codec was standardized by ITU-T in May 2006 as a part of standardization activity held in Jan 2004. This is the former speech codec with an embedded scalable structure and is backward compatible with existing ITU G.729 speech coding standard. As bit rate increases quality of speech will increase proportionally due to the scalable structure of the codec. Scalable coding technology used by the G.729.1 codec allows the proper selection of bit-rate during transmission by any communication component. This is done by simply truncating the bitstream. Coder operates in many modes and high-quality speech communication is obtained with the low-delay mode. The main applications of this codec are: VoIP (IP telephony) including IP phones, VoIP handsets, voice recording equipments audio/video conferencing, media servers/gateways, call center equipment voice messaging servers. G.729 Annex J and G.729EV is another name for G.729.1 where EV denotes Embedded Variable (bit rate).

StarCore's variable length execution set (VLES) architecture processors are largely used in codecs due to lower power consumption, efficient compilability, and very compact code density. Merging of VLES and DSP technologies on to a single core/system has mitigated the ever increasing requirements of signal processing applications on several platforms. Here we are going to discuss the implementation of ITU's G.729.1 fixed point standard on SC3850 digital signal processor.

II. OVERVIEW OF G.729.1 CODEC

G.729.1 codec is an 8-32 kbit/s scalable wideband speech and audio coding algorithm interoperable with G.729, G.729A and G.729B[2]. The input which is an analog audio signal is sampled at 8 KHz or 16 KHz (default) sampling frequency. The resulting signal is then quantized and converted to 16-bit linear PCM which serves as the input to the encoder. Decoder output is also 16-bit linear PCM. The output bitstream of encoder is scalable and is structured in 12 embedded layers. This layers corresponds to 12 available bit rates from 8 to 32 kbit/s with core layer interoperable with G.729. The output bandwidth of G.729.1 is 50-4000 Hz at 8 and 12 kbit/s and 50-7000 Hz from 14 to 32 kbit/s (per 2 kbit/s steps). The G.729.1 coder operates on

20 milliseconds frames of input speech, corresponding to 320 samples at a sampling rate of 16000 samples per second. In-order to consistent with G.729 using 10 ms frame and 5 ms subframes [3], two 10 ms CELP frames are processed per 20 ms frame. Hence 20ms frames are referred to as superframes.

A. Encoding principle

Encoding is done in such a way that the input speech samples are analyzed in order to extract the parameters of the speech synthesis model. Transmitting these parameters as opposed to compressed speech samples saves the bandwidth and reduces the noise. At the receiving side parameters are used to synthesize and reconstruct speech.

The G.729.1 encoder is illustrated in Figure 1. The encoder operates at the maximal bit-rate of 32 kbit/s. The coder is organized in three-stages: embedded Code-Excited Linear-Prediction (CELP) coding, Time-Domain Bandwidth Extension (TDBWE) and predictive transform coding [2]. Layers 1 and 2 is generated by the embedded CELP stage which gives a narrowband output(50-4000 Hz) at 8 and 12 kbit/s. Layer 3 is generated by the TDBWE stage producing a wideband output (50-7000 Hz) at 14 kbit/s. Layers 4 to 12 are generated by the TDAC stage operates in the Modified Discrete Cosine Transform (MDCT) domain at 14 to 32 kbit/s.

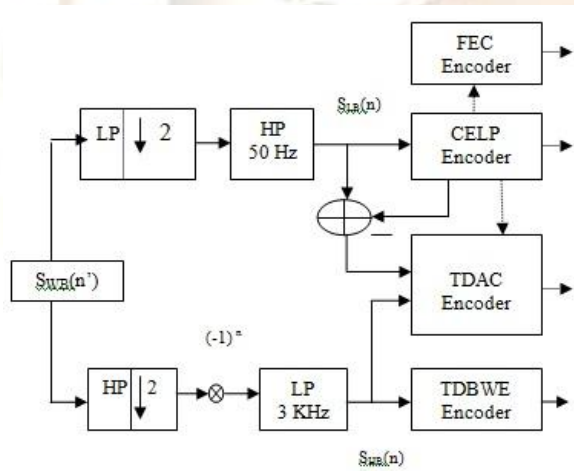


Figure 1. Block diagram of encoder

A 64-coefficient quadrature mirror filterbank (QMF) divides the input signal $S_{WB}(n')$ into a higher band and lower band. The input signal at lower-band and higher-band is decimated by a factor of 2. In order to remove the frequency components below 50Hz the lower-band decimated signal is passed through an elliptical high-pass filter (HPF) with cut-off frequency 0.05 KHz. The resulting signal is encoded by a narrowband embedded CELP encoder [3]. The input signal at higher-band is spectrally folded to get more exact representation and is passed through an elliptic low-pass filter (LPF) with 3 kHz cutoff frequency to remove the frequency components below 3 KHz. The resulting signal is then encoded by parametric time-domain bandwidth extension (TDBWE) encoder.

Time-domain aliasing cancellation (TDAC) encoder which is a transform-based coder jointly encodes the lower-band CELP difference signal and the higher-band signal $S_{HB}(n)$. In addition, parameter-level redundancy in the bitstream is introduced by the frame erasure concealment (FEC) encoder by transmitting the signal classification information, energy information, and phase information [2].

B. Decoding principle

G.729.1 decoder is illustrated in Figure 2. At the receiver, the transmitted parameters are decoded. The decoder operation is based on the received bit-rate or actual number of received layers. If the received bit rates are 8 or 12 kbit/s the CELP decoder does the job of decoding by reconstructing a lower-band signal (50-4000 Hz). This is then post-filtered and pre-processed by a high-pass filter to increase the subjective quality and reduces the coding error. The resulting signal is upsampled to 16 kHz using the QMF synthesis filterbank. If the received bit rate is 14 kbit/s, the TDBWE decoder reconstructs a higher-band signal $S_{BWE}(n)$ which is upon merging with narrowband enhancement layer (12 kbit/s) produces a wide-band output of 50-7000 Hz. If the received bit rate is between 16 to 32 kbit/s, the TDAC decoder reconstructs both a lower-band difference signal and a higher-band signal. In order to mitigate the pre/postecho artefacts due to transform coding the resultant signal is shaped in time domain. CELP output is combined with modified TDAC lower-band signal. The quality of the speech output can be improved by using modified TDAC higher-band signal instead of TDBWE output for the whole frequency range. The signals are then upsampled to 16 KHz and combined in QMF filterbank.

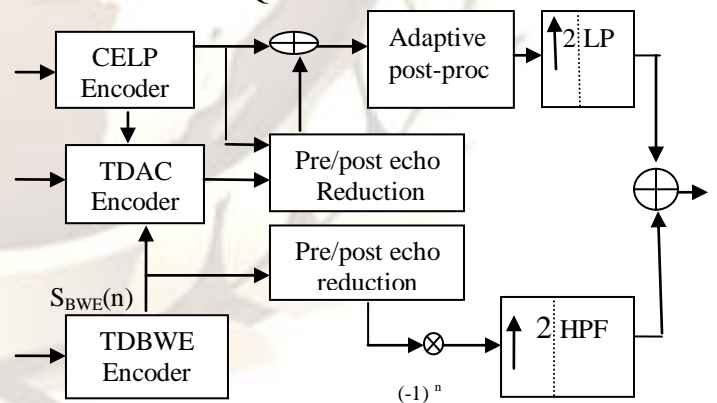


Figure 2. Block diagram of decoder

III. FIXED POINT IMPLEMENTATION

A. Architecture

The StarCore SC3850 DSP core used in mainstream DSP applications, such as wireless and wireline communications, on both the infrastructure and the subscriber sides. The SC3850 core is having a variable-length execution set (VLES) execution

model, which utilizes maximum parallelism by allowing multiple address generation and data arithmetic logic units to execute multiple instructions in a single clock cycle[4]. Target markets for the SC3850 architecture includes wireless and wireline base stations, wireless and wireline infrastructure, broadband wireless etc. Key features of the SC3850 DSP core include the following [4]:

- Main core resources
 - 4 data ALU execution units
 - 2 integer and address generation units
 - Sixteen 40-bit data register
 - Sixteen 32-bit address registers
- Instruction set
 - 16-bit instruction set, expandable to 32- and 48-bit instructions
 - Sixteen 40-bit data register
 - Sixteen 32-bit address registers
 - Very high execution parallelism
 - Up to six instructions executed in a single clock cycle, statically scheduled
 - Up to 4 data ALU instructions and 2 memory access/integer instructions per cycle
- Data type support
 - Byte (8-bit), word (16-bit), and long (32-bit) data widths, supported by instructions and memory moves.

B. Software

C language was used for high level implementation. The encoder and decoder parameters were dealt efficiently with the help of structures. SC3850 assembly language was used for low level modifications. Firstly the encoder and decoder tester applications were developed using Visual Studio 2008 express edition by modification of reference code. Later the code is loaded to Code Warrior for SC3850 specific implementation.

IV. OPTIMIZATION

The methodology for reforming a software system in all aspects to make it work more efficiently utilizing fewer resources is known as optimization. Optimization can be done for speed and memory. Here we concentrated on speed optimization for superior performance.

The optimization tool used in this project was Code Warrior 10.2.9. It is latest version from the CodeWarrior family provides a graphical user interface for managing software development projects. CodeWarrior 10.2.9 can be used to develop C, C++, and assembly language code targeted at many processors.

A. Optimization strategy

The strategy we followed in the optimization is as follows

1. Initial MCPS of the code is calculated by running the project with worst-case and all the optimization parameters enabled.
2. Profile all the functions in the implementation. Identify the critical functions that takes the more

number of processor cycles by analyzing the profiler information.

3. Apply generic optimization techniques to improve the execution speed in C.
4. Analyze the disassembly the code (assembly generated by the compiler) to find the areas where better assembly can be written.
5. After each optimization MCPS is calculated and note down the value.
6. If not satisfied with the performance of codes, go back to step 2.

B. Profiling

Profile information of the reference code gives an precise estimate about the relative contribution of different modules to the total computational complexity. Profiling information helped to identify critical portions of the code, which were optimized to improve the performance. Codewarrior's simulator configuration is used for profiling.

C. MCPS Calculation

MCPS (million cycles per second) is a performance metric. It is the total no. of cycles taken to process all the frames It is obtained by multiplying the no. of cycles required to execute one frame by no. of frames per second as in (1).

$$MCPS = \frac{C \times fs}{S \times N} \quad (1)$$

where, C is total no. of cycles for executing one frame, fs is the sampling rate, S is the total no. of samples / frame and N is the total no. of frames .

D. Optimization Techniques

We followed both target dependent and target independent optimization techniques in this project. Target independent techniques are generalized techniques applied at high level optimization. Target dependent optimization involves techniques utilizing architectural features of SC3850. The compiler specific optimization level was set at -o3[5]. Some of the techniques followed are explained as follows[5]

1) Making use of intrinsics

An intrinsic function is a way to instruct the compiler to use a specific assembly language instruction [5]. Many functions in the reference code were replaced by intrinsic after studying its exact functionality. The replacement was done with proper care to maintain bit-exactness of the original ITU-T implementation.. The StarCore C/C++ compiler substitutes the intrinsic function call with a set of designated assembly language instructions. Intrinsics improves performance of generated code and exploits the architectural features of SC3850 that can't be done by normal C language.

2)Function inlining

Function inlining is the process of substituting the called function code in the place of functional call thereby eliminating functional call overhead .Inlining can be done by instructing the StarCore compiler using special #pragma directives. This technique improves execution time at the expense of larger code size. Small functions that are frequently-called are the best candidates for inlining in the C code. The criteria for the selecting the best candidate depends upon the number and type of parameters the function passed, the data type of the returned value and register allocation in the resulting assembly code.

3)Fast 32-bit DPF format operations

The ITU-T G.729EV reference code was originally designed for 16-bit processors which do not support 32-bit operations. Thus the 32-bit operations were done using a 32-bit double precision format in which data is represented with a precision of 1 in 2^{31} . Some functions which originally took the 32-bit parameters in two higher and lower 16-bit half words and perform 16-bit operations on them were replaced with single 32-bit DPF parameter. Instead of using two separate DALU registers for two 16-bit half words now only a single DALU register for a 32-bit word is used. Thus, functions that originally received two pointers to two 16-bit arrays could now operate with pointer to a single 32-bit array. Modifications are implemented in DPF format to maintain bit-exactness of the original ITU-T implementation. This optimization makes use of the processors 32-bit capabilities and got 50% reduction in MCPS and memory moves since the parameters passed to function reduced by half.

4)Loop unrolling

Looping overhead in software which increases the cycle count can be reduced by loop unrolling. Unrolling the loop N times can be done by duplicates the loop body N times and decreasing the loop counter by a factor of N. Since the architecture of SC3850 contains 4 ALU's unrolling the bigger loops (higher loop count) by a factor of N=4 gave better performance. This significantly reduces the MCPS, since it maximizes the efficient usage of multiple ALU execution units simultaneously and multiple register moves. Unrolling the inner loops reduces the function looping overhead significantly, as loop counter needs to be updated less often and fewer branches are executed.

5)Loop merging

Combining two or more loops of same loop count into a single loop loads the ALU more efficiently. Loop fusion is a process of combining multiple loops, which reuse the same data, into one loop. The advantage of this is that data can be reused, reducing the memory access, and loop overhead is reduced.

6)Use of SIMD2 instructions

This optimization will make efficient use of MAC registers and enables packet data movement results in

reduced memory moves with high degree of parallelism.

7)Making use of overflow flag

SC3850 Exception and mode register (EMR) has an overflow flag called DOVF which is set if saturation results due to some of the arithmetic instructions. Overflow check can be done efficiently by monitoring this flag and this can be directly accessed by making use of assembly routines.

V. VERIFICATION AND TESTING

At each stage of optimization we verified the correctness of our implementation using the test vectors included as a part of G.729.1 recommendation. Each of these test vectors includes an input and output file. For each input file, the output file of an implementation should match the prescribed output for the specified input in a bit exact manner. By doing this testing phase we became assured that our implementation was fully compliant with pseudo codes provided by CCITT.

VI. RESULTS AND CONCLUSION

In this paper we presented fixed-point optimized implementation of G.729.1 codec (annex J) on SC3850 DSP core. First the details of implementation of this algorithm in C language were discussed. Next using our C written codes, optimization was done for real-time implementation. We suggested a strategy for optimization process which can be used in other similar researches. Using the proposed strategy, a set of different techniques in order to reduce consuming cycles of G.729.1 algorithm were introduced. These techniques included some methods based on SC3850 architecture and some methods specifically for optimization of G.729.1 algorithm. . The codec can be easily used for real-time applications. The speed of operation of codec is improved by 90% and it's a great success.

Table.1 shows the MCPS consumption of the codec before and after optimization for a sampling rate of 16 KHz .

| G.729.1 Codec | Initial MCPS | After adding Intrinsic (MCPS) | Final MCPS | % improvement |
|----------------------|---------------------|--------------------------------------|-------------------|----------------------|
| Encoder | 400 | 245 | 28 | 93 |
| decoder | 240 | 215 | 22 | 90 |

ACKNOWLEDGMENT

I gratefully acknowledge the contributions of Federal Institute of science and technology, ankamaly for the support and assistance for the successful completion of this project .

REFERENCES

- [1] Wai c Chu Speech Coding Algorithms Foundation and Evolution of Standardized Coders- Mobile Media Laboratory DoCoMo USA Labs San Jose, California.
- [2] ITU-T Rec. G.729.1, "G.729 based Embedded Variable bit-rate coder - An 8-32kb/s scalable wideband coder bitstream interoperable with G.729" May 2006.
- [3] ITU-T Rec. G.729, "Coding of Speech at 8 kbit/s using Conjugate Structure Algebraic Code Excited Linear Prediction (CSACELP)," March 1996
- [4] MSC8156 Reference Manual Six Core Digital Signal Processor MSC8156RM Rev 2, June 2011
- [5] C Code Optimization Examples for the StarCore SC3850 Core, Rev 0, April 2010