Real-Time Implementation of G.723.1 Speech Codec on a 16-bit DSP Processor

Sheng-Kai Chang and Jwu-Sheng Hu

Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C

Abstract

This thesis presents a full-duplex, real-time implementation of ITU-T G.723.1 [1] speech coder on a 16-bit fixed-point TI's DSP chip, TMS320C5402. This thesis is divided into two parts. In the first part of the thesis, a brief introduction of G.723.1 speech encoder and decoder is presented. In the second part, the fixed-point operations and optimization methods are proposed in order to reduce the total cycle times consumed in real-time implementation. After the optimization, the total code size is 26.6k words and the computation complexities are 37 and 39 MIPS in the dual bit rates of 5.3k and 6.3kbps respectively, which are less than 40% of 100 MIPS DSP. In addition, a real-time demo using Pipe mechanism is also presented in this thesis.

I. Introduction

Most of these applications require that the speech signal is in digital format so that it can be processed, stored, or transmitted. But it is also associated with a high data rate and requires more transmission bandwidth and storage. Speech Coding or Speech Compression is the field concerned with achieving compact digital representations of voice signals for the purpose of efficient transmission or storage.

The performance of speech coding is evaluated based on four attributes: bit rate, delay, complexity, and quality [2]. There is a strong interaction between all these attributes and that they can be trade-off against each other. In the past ten years, ITU-T (International Telecommunication Union Telecommunication Standardization Sector) sets up different speech standard coders for each application demand. The ITU-T G series are especially for transmission systems and media, digital systems and networks.

G.723.1 is intended to standardize telephony/videoconferencing over public telephone (POTS) networks and is a part of the overall ITU-T H.324 standard. It is also recommended as the low bit rate speech technology for the ITU-T H.323 audio and video standard. This algorithm is applicable for real-time video and teleconferencing application where reduced bandwidth and very high quality voice is required. However, the performance of the original ITU-T source code is not high enough. Because DSP could perform complicated mathematical operations on real-time signals, and DSP processors are specially architected to

accelerate the execution of repetitive, numerically intensive calculations. Due to the large computational time required for the encoder and decoder, digital signal processor (DSP) is suitable for implementation in real-time.

This thesis is divided into two parts. In the first part of the thesis, a brief introduction of G.723.1 speech encoder and decoder is presented. In the second part, the optimization methods are proposed.

II. G.723.1 Coder Principle

The block diagram of G.723.1 speech encoder is shown in Fig. 1. The G.723.1 coder is based on the principles of linear prediction analysis-by- synthesis (ABS) coding [3][4] and attempts to minimize a perceptually weighted error signal. The G.723.1 coder also belongs to the Code-Excited Linear-Prediction (CELP) model [3][4]. This coder operates on a speech of 30 ms, which corresponds to 240 samples at a sampling rate of 8kHz. For every 30-ms frame, the speech signal is analyzed to extract the CELP parameters. These parameters are the Linear-Prediction (LP) filter coefficients, adaptive and fixed-codebook [5] indices and gains. Note that the 30-ms speech frame is subdivided into four 7.5-ms subframes. All the parameters except Line Spectrum Pairs (LSPs) [3][6] are encoded for each subframe. The reason for splitting into four subframes is to prevent the spectral transition from one frame to another from being abrupt and hence improving the speech quality.

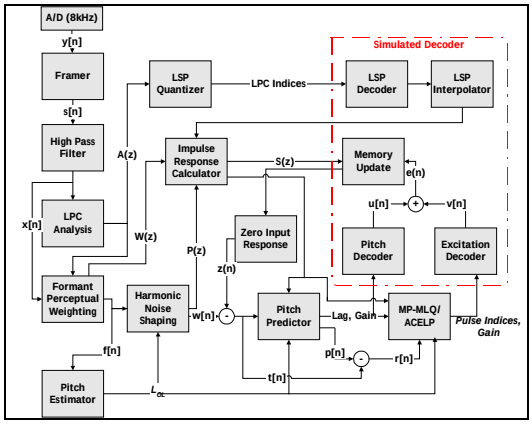

Fig. 1 G.723.1 Encoder

At the decoder, the parameters are used to retrieve the excitation and synthesis filter parameters. Speech is reconstructed by filtering this excitation through the short-term synthesis filter shown is Fig. 2. The short-term synthesis filter is based on a tenth-order LP filter. The long-term synthesis filter is implemented using the so-called adaptive-codebook approach. After computing the reconstructed speech, it is further enhanced by a postfiler. Table 1 summarizes the coding parameters for the G.723.1 speech coder.

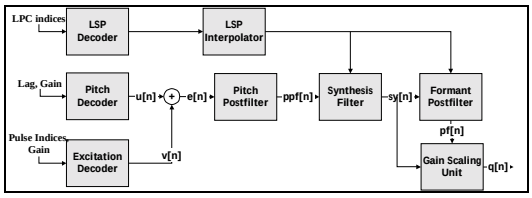

Fig. 2 G.723.1 Decoder

Table 1 Coding parameters and bit allocation for G.723.1

Parameters coded	6.3kbps	5.3kbps
LPC indices	24	24
Adaptive codebook lags	18	18
All the gains combined	48	48
Pulse positions	73	48
Pulse signs	22	16
Grid index	4	4
Total	158	158

III. Architecture of DSPC5402

For the real-time implementation of speech

coding, the Texas Instrument (TI) Development Starter Kit (DSK) with DSP C5402 chip is used.

The C5402 device key features are as follow [7]: The clock rate is 100 MHz; the cycle time is 10ns. A 40-bit Arithmetic Logic Unit (ALU) including a 40-bit barrel shifter and two independent 40-bit accumulators, for single-cycle instructions with shifting. A 17-× 17-bit parallel multiplier coupled to a 40-Bit dedicated adder for non-pipelined single-cycle Multiply/Accumulate (MAC) operation as shown in Fig 3. There are 4K x 16-bit on-chip ROM and 16K x 16-bit dual-access on-chip RAM. The advanced multibus architecture with three separate 16-bit data memory busses, one program memory bus, and four address buses, for efficient data access. Two address generators with eight auxiliary registers and two auxiliary register arithmetic units, which facilitate multiple data operand operations. The C54x DSP architecture supports single-cycle direct addressing of data memory.

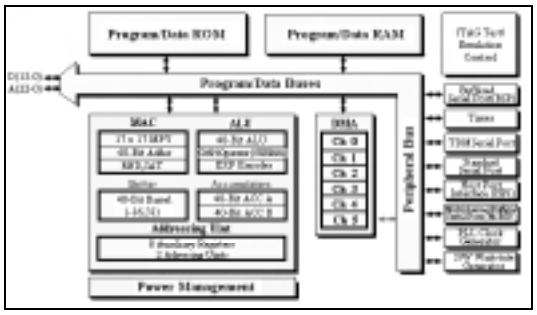

Fig. 3 Architecture of TMS320 C54x DSP

IV. Implementation

The whole codec process per frame has to be completed within 30 ms so that a decoding speech signal should not be broken. Before Optimization, this thesis analyzes this coder computation. The computations on DSP are 86 and 69 million cycles at 6.3 and 5.3kbps respectively. Consequently, in order to realize a real-time codec with the utilization of this 16-bit DSP processor, which the cycle time is 10ns, the total computation at least have to improve 29 and 23 times at 6.3 and 5.3 kbps, respectively.

$$\begin{matrix} \frac{86\,million\,cycles\times 10ns}{30ms} \approx 29\,times \\ \frac{69\,million\,cycles\times 10ns}{30ms} \approx 23\,times \end{matrix}. \qquad (1)$$

The optimizations of the G.723.1 codec are divided into three parts: fixed-point operation, assembly-level optimization [9] and C-level optimization [8]. This thesis lists some techniques of optimization using in the implementation.

Fixed-point Operation

The fixed-point DSP can handle all mathematics in fixed-point operation. We should pay attention to the dynamic range of all mathematics, scaling operation and overflow case in fixed-point operations.

◆ Pitch Estimation

In order to increase the precision of the input signal, $f[n]$ with Q15 format, vector normalization function is used before the pitch estimation. And then Q11 format is used for this signal, $f[n]$. The normalized autocorrelation criterion [6], $C_{OL}(j)$, are defined as follow:

$$C_{OL}(j)=\frac{\left(\sum_{n=0}^{119} f[n]\cdot f[n-j]\right)^2}{\sum_{n=0}^{119} f[n-j]\cdot f[n-j]},\ 18\le j\le 142 \qquad (2)$$

$$C = \sum_{n=0}^{119} f[n] \cdot f[n-j] \quad E = \sum_{n=0}^{119} f[n-j] \cdot f[n-j]$$

The worst case of the correlation output C and E are 120 when $f[n]$=1 for any values n. Generally, this will cause the overflow situation.

After the multiplications of two Q11 format data, the result still reserve 7-bit (128) to handle the worst-case value, 120 as shown in Fig. 4. Vector normalization function could prevent the overflow case from the auto-correlation or cross-correlation computation.

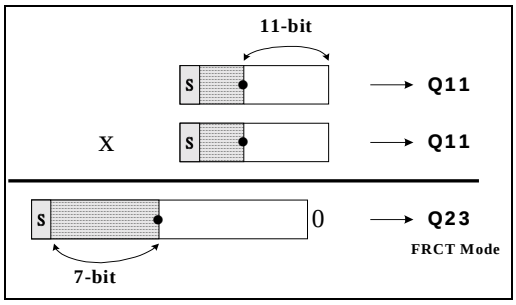

Fig. 4 Multiplying Two Q11 Numbers

In this coder, this thesis totally uses the Q0 (integer), Q15 (fraction) and Q11 (mixed integer-fraction) formats depend on the dynamic ranges of each signal.

Assembly-level Optimization

To analyze the G.723.1 C codes, we find that the original C codes are based on European Telecommunications Standards Institute (ETSI) functions [8]. All math operations defined in basop.c in the reference C program can be replaced by TMS320C54x intrinsics, if possible. But in view of the entirety of efficiency, this thesis chose to code directly in assembly language [10] instead of intrinsics. Besides, the memory management was applied for further reduction of cycle and program size.

C-Level Optimization

TI code generation tools have been designed to achieve the best optimization possible for the entire application, not for specific kernels. Since the tools look at the entire code, not selected pieces, we may see inefficiencies in a certain kernel of code that reflect efficient code generation in another section of code.

Here, this thesis presents some C-coding style guidelines to improve the efficiency of the TMS320C54x C-compilers [11].

(1) Allocate most often used elements of a structure, array or bit-fields in the first element, the lowest address or LSB respectively. (2) When initializing different variables with the same constant, rearrange the code. (3) Use memcpy when copying an array variable into another. (4) Rearrange all like data declarations together, and listing 16-bit data first.

Benchmark Results

The real-time implementation of G.723.1 on DSPC54x has passed all test vectors given by ITU-T. The total code size is 26.6k words as shown in Table 1. After Optimization, the computation loadings are 39 and 37 MIPS for full-duplex implementation as shown in Table 2.

Table 1 Code Size of Implementation

	Program Memory	Data Memory	Total
Original	36.0K	14.7K	50.7K
Optimized	12.1K	14.5K	26.6K
Improved ratio	66.4%	1.3%	48.2%

Table 2 Performances at Each Bit rate

5.3kbps	Encoder	Decoder	Total Computation (MIPS)
6.3	1096506.7	99848.5	39
5.3	1012922.8	98323.1	37

Speech Quality Assessment

Spectral distortion (SD) [5] measures in the frequency domain are defined as distortion between the input and output whole speech spectra:

$$SD = \sqrt{\frac{1}{\pi}\int_0^{\pi} \{10\log_{10} s_x(\omega) - 10\log_{10} s_y(\omega)\}^2 d\omega} \quad (3)$$

Where $S_x(w)$ and $S_y(w)$ are the power spectra of the input and output speech signal respectively, and w is the signal frequency band. The smaller SD indicates the better objective speech quality.

Table 4 SD Values at Each Bit Rate

SD	Floating-point (dB)	Optimized (dB)
5.3kbps	4.15	4.39
6.3kbps	4.03	4.05

Real-time Implementation

This program has demonstrated the real-time implementation of G.7231.1 codec using DSP/BIOS [11]-[14] APIs for scheduling data transfer between the hardware I/O peripherals and the target DSP as shown in Fig. 5. From the result obtained, the reconstructed speech was near toll-quality except some interference. The interferences might be attributed to the noisy environment of the DSK.

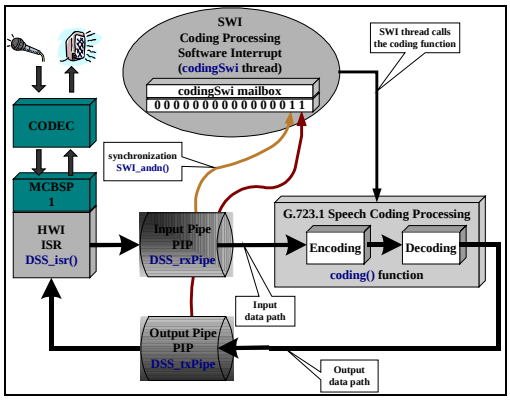

Fig. 5 Block Diagram of Pipe Mechanism

V. Conclusion and Future Work

Speech coding technique has been applied extensively to the communications over Internet and. It is efficient to deliver and store speech data with the digitalize process of the speech. Because of the lower bit rate and higher quality, we choose the G.723.1 speech coding as our research.

The goal of this thesis is to investigate the properties of the G.723.1 speech standard coding and their real-time implementations on a 16-bit DSP processor. In the first part of this thesis, we introduce the theory of G.723.1 both on encoder and decoder. Due to the original coding algorithm is too complex, and costs too much computation; we propose some techniques to optimize. The second part of this thesis is its implementation.

Efficient implementation of ITU-T G.723.1 speech coding using a TI TMS320C5402 DSP chip is proposed. Optimization process went through two steps. Initially, the ITU-T C-code was rewritten to assembly-code. Second, the new memory management was applied for further reduction of cycle and program size. This thesis makes good use of the lower-cost DSP processor

that performs well enough G.723.1 codec implementation. Finally, this thesis also gives a demo to show the truly real-time implementation. This implementation can be combined the image and speech over Internet to Video Over Internet Phone (VOIP).

References

[1] ITU-T Recommendation G.723.1, "Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s", Mar 1996.

[2] Cox, R.V.; Kroon, P., "Low bit-rate speech coders for multimedia communication", IEEE Communications Magazine, Volume: 34 Issue: 12 , Dec. 1996 Page(s): 34 –41.

[3] Kondoz A.M., "Digital Speech Coding for Low Bit Rate Communication Systems", John Wiley & Sons, 1994.

[4] *Spanias, A.S.,"* Speech coding: a tutorial review", Proceedings of the IEEE, Volume: 82 Issue: 10, Oct. 1994, Page(s): 1541 –1582.

[5] Sadaoki Furui and M. Mohan Sondhi, "Advances in Speech Signal Processing", Marcel Dekker, 1992.

[6] Randy, Goldberg and Lance Riek, "A Practical Handbook of Speech Coders", CRC Press, 2000.

[7] "TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals", Texas Instruments, 2000.

[8] "TMS320C54x Optimizing C Compiler User's Guide", Texas Instruments, 1999.

[9] "TMS320C54x Assembly Language Tools User's Guide", Texas Instruments, 1999.

[10] 胡竹生, 賴鴻志, 張勝凱, "DSP TMS320C54xx 晶片原理與應用", 全華圖書, 2000.

[12] Karen Baldwin and Rosemarie Piedra, "Generating Efficient Code with TMS320 DSPs: Style Guidelines", Texas Instruments, 1997.

[11] "TMS320C54x DSP/BIOS User's Guide", Texas Instruments, 2000.

[12] "TMS320C54x DSP/BIOS Application Programming Interface (API) Ref Guide", Texas Instruments, 2000.

[13] Jack Greenbaum, "Writing Flexible Device Drivers for DSP/BIOS", Texas Instrument, 2000.

[14] Andy The' David W. Dart, "How to Get Started With DSP/BIOS II", Texas Instrument, 200