8- AND 16-kb/s APC-AB VOICE CODEC USING A SINGLE CHIP DSP

Kazumi SATOH*, Hideaki KURIHARA*, and Shigeyuki UNAGAMI*
Masanori KAJIHARA** and Yoshihiro TOMITA**

*Fujitsu Laboratories Ltd.
**Fujitsu Limited
1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan

ABSTRACT

This paper describes the implementation of an APC-AB codec using a single-chip DSP. To handle the simultaneous 8- or 16-kb/s encoding and decoding with a single-chip DSP, we precisely evaluated arithmetic bit accuracy, simplifying the DSP algorithm and sharing subroutines. We designed a dedicated DSP with a 16E8 floating-point data format and a novel DMA control circuit. The 100 X 68 mm codec module contains 2 K words of RAM, a 12-ms echo canceler, and two interface LSI chips. An SNR of 39 dB for 16-kb/s coding and 36 dB for 8-kb/s coding were obtained using method 2 of CCITT recommendation G.712. The module consumes less than 1.0 watt.

1. INTRODUCTION

Medium-band speech coding techniques for 8-16 kb/s are coming into increasing use in applications such as intracompany and mobile communications[1]. Recent advances in digital signal processing and LSI techniques will make it possible for smart implementation of a complicated algorithm.

Adaptive predictive coding with adaptive bit allocation (APC-AB) is a promising algorithm for medium-band speech coding[2]. We studied an 8-kb/s and 16-kb/s APC-AB digital signal processing to implement a single-chip DSP. We evaluated the arithmetic operation and control based on the general-purpose floating-point DSP (MB86232), and developed a candidate DSP (MB87528) for the APC-AB codec.

2. ALGORITHM

The band splitting filter (Figure 1) divides the input signal into 0-1 kHz, 1-2 kHz, and 2-4 kHz subbans. The predictor consists of a 4th-order short-term predictor and an 8th-order long-term predictor. The transfer function of short-term predictor is shown in Equation (1), and the transfer function of long-term predictor is in Equation (2).

\[ A(z) = \sum_{j=1}^{4} a_j z^{-j} \quad (1) \]

\[ B(z) = \sum_{j=1}^{8} b_j z^{-j-m} \quad (2) \]

\[ m = \left[ Tp \cdot W \right] - \left( N-1 \right) \cdot W/2 \]

Figure 1. APC-AB coding algorithm
Short-term predictor coefficients are converted to LSP parameters. The pitch period (Tp) is determined by

$$C_i = \sum_{j=1}^{2^{n-1}} X_j \cdot X_{j+1} \quad 4 \leq i \leq 30 \quad (3)$$

$$T_p = \text{arg max } (C_i) \quad (4)$$

where

- $X_j$: Lower subband signals
- $n$: Number of samples of the lower subband

Bits are adaptively allocated in both frequency and time domains. Allocation in the frequency domain is determined by the residual power in each subband, then that in the time domain is done using the residual power in each subinterval. Subintervals are defined by the pitch period (Tp) and pitch position (Td) (Figure 2). The residual signal in Tp is divided into four equal subintervals. Td is defined as the first position in the frame at which a residual signal level becomes maximum [3]. A adaptive bit allocation for subintervals is derived from Equation (5) [4],

$$R_i = R \times (1/2) \log \left( \frac{U_i}{\sum_{j=1}^{n} U_j} \right) \quad (5)$$

where

- $R_i$: Number of allocated bits in the i-th subinterval
- $R$: Number of average residual signal bits in a frame
- $U_i$: Residual power in the i-th subinterval
- $C_j$: Ratio of the subinterval width to Tp
  
  (in this case, $C_j = 0.25$)
- $n$: Number of subintervals

The log of the residual power ($U_i$) in the subinterval and total residual power are K-L transformed and converted data is quantized.

The residual signal of each subband is quantized based on adaptive bit allocation. A code word consists of the pitch period, pitch position, pitch gain, LSP parameters and residual power. Table 1 lists code word. In 8-kb/s coding, LSP parameter and residual power quantization bits are allocated fewer than that in 16-kb/s coding to increase the number of bits of the residual code. The number of residual code bits are 1.34 bits per sample for 16-kb/s coding and 0.74 bits per sample for 8-kb/s coding.

Table 2 shows the main APC-AB algorithm features.

3. DSP PROGRAM FEATURES

3.1 Optimized for DSP Processing

We studied the DSP program required to efficiently process a simultaneous operation of 8- or 16-kb/s APC-AB encoding and decoding. Figure 3 shows the DSP program flow.

The main differences between 8-kb/s coding and 16-kb/s coding are the number of data items sampled per frame, the number of bits allocated for the code word and the table data for parameters. The number of loops processed in adaptive predictive coding and the access for LSP parameter data memory are determined based on the selected coding bit rate. The quantization data of the residual signal power and the average number of bits for adaptive allocation are selected by table look-up.

Common processing in both encoding and decoding such as LSP parameters calculation, K-L transformation and the adaptive bit allocation are handled as subroutines. And then also, band splitting filter and long-term prediction filter coefficients, and the quantization data for both LSP parameters and the residual signal power are used in common both encoding and decoding.

3.2 Considering for synchronous timing

APC-AB coding scheme is processed frame-by-frame. A coding timing chart is shown in Figure 4. The transmitting and receiving frame clock phases are not synchronized. We developed the following processing to handle simultaneous encoding and decoding in real time. Decoding is always done after encoding. If decoding is interrupted before $T_s$ in Figure 4, decoder uses the current received
3.3 Processing Capacity

With the new DSP, the total number of algorithm steps is about 3.6 K steps for encoding and decoding (Table 3). The DSP processes 320 K cycles during the 32-ms frame for 8-kb/s coding, and 160 K cycles during the 16-ms frame for 16-kb/s coding. The APC-AB codec operates within these cycles.

Figure 3. DSP program flow

Figure 4. Coding timing chart

Table 3. Memory requirements and processing cycles

<table>
<thead>
<tr>
<th>Program steps (words)</th>
<th>General-purpose DSP (MB86232)</th>
<th>Special-purpose DSP (MB87528)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 kb/s</td>
<td>3.3 k</td>
<td>3.6 k</td>
</tr>
<tr>
<td>16 kb/s</td>
<td>6.4 k</td>
<td>4.3 k</td>
</tr>
</tbody>
</table>

Table 4. Main DSP features (MB87528)

<table>
<thead>
<tr>
<th>Technology</th>
<th>1.2 mm, CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip size</td>
<td>12.8 mm square</td>
</tr>
<tr>
<td>Package</td>
<td>88-pin PGA</td>
</tr>
<tr>
<td>Program ROM</td>
<td>4096 words × 32 bits</td>
</tr>
<tr>
<td>RAM</td>
<td>256 words × 24 bits (Internal)</td>
</tr>
<tr>
<td>Multiplier</td>
<td>16 E8 × 16 E8 ⇒ 16 E8</td>
</tr>
<tr>
<td>Instruction cycle</td>
<td>100 nsec (Floating-point 200 nsec)</td>
</tr>
<tr>
<td>I/O</td>
<td>Serial 2</td>
</tr>
</tbody>
</table>

5. HARDWARE IMPLEMENTATION

We fabricated the APC-AB codec module (Figures 5 and 6) using the MB87528 and several customized LSI chips.

1) APC DSP (MB87528): Special-purpose APC-AB DSP. Input and output speech data are transferred under DMA control.
2) EC INF (MB606608): PCM codec interface LSI supporting linear to nonlinear conversion.
3) EC (MB87064): Echo canceler rejecting 12-ms echo delay.
4) LINE INF (MB660615): Line interface LSI chip for detecting and synchronizing frames.

The module consumes less than 1.0 watt and is 100 X 68 mm.
6. EVALUATION

We compared the results of 24E8 format computer simulation and 16E8 format hardware emulation with the results of 8-kb/s and 16-kb/s transmission tests. We confirmed the coincidence of the transmission codes for sine wave data and male and female voice data. The SNR characteristics were measured based on CCITT G.712 Methods 1 (narrow-band noise) is shown in Figure 7 and 2 (sine wave) is shown in Figure 8.

7. CONCLUSION

We developed an APC-AB codec module which processes either 8-kb/s or 16-kb/s coding by using an optimized algorithm and an advanced special-purpose floating-point DSP. The SNR performance for 16-kb/s coding satisfied the CCITT G.712 specifications.

ACKNOWLEDGMENTS

We like to thank Mr. M. Taka of NTT Laboratories and Dr. H. Takanashi and Dr. K. Murano of Fujitsu Laboratories Ltd. for their guidance.

REFERENCES