Abstract

Title of Thesis: An Implementation of Binomial-QMF for Subband Coding of Speech

Feng Chen, Master of Science in Electrical Engineering, 1991

Thesis directed by: Professor Ali N. Akansu

A real-time subband speech codec is implemented by the author on TMS320C25 digital signal processor based hardware for speech compression down to 0.25 bit per sample.

The Binomial Quadrature Mirror Filters (B-QMFs) are employed as the basic block for the hierarchical, dynamic tree, analysis/synthesis filter bank. After the analysis, available bits are optimally allocated among subbands. Band statistics are updated to overcome the nonstationarities of speech signal.

The energy compaction performance of this codec is compared with different subband and DCT in simulations for speech signals and AR(1) source models.

It is shown that the speech codec implemented provides an intelligible speech quality even at 0.25 bits per sample.

An Implementation of Binomial-QMF for Subband Coding of Speech

by
Feng Chen

Thesis submitted to the Faculty of the Graduate School of the New Jersey Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

1991

APPROVAL SHEET

Title of Thesis: An Implementation of Binomial-QMF for Subband Coding of Speech

Name of Candidate: Feng Chen
Master of Science in Electrical Engineering, 1991

Thesis and Abstract Approved:

Dr. A. Akansu — Date
Assistant Professor
Department of Electrical Engineering

Dr. N. Ansari — Date
Assistant Professor
Department of Electrical Engineering

Dr. Z. Siveski — Date
Assistant Professor
Department of Electrical Engineering

VITA

Name: Feng Chen

Permanent address:

Degree and date to be conferred: M. Sc., Dec 20, 1991.

Date of birth:

Place of birth:

Secondary education: High School Attached to Fudan University, Shanghai, China

Collegiate institutions attended:	Date	Degree	Date of Degree
New Jersey Institute of Technology	9/89-12/91	M. Sc.	Dec 1991
Fudan University	9/84-7/88	B. Sc.	July 1988

Major: Electrical and Computer Engineering.

To my wife and my parents

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data compression plays an important role in data transmission and storage. Considering speech signal as an example. The speech signal is first lowpass filtered for anti-aliasing and then it is sampled at the rate of f_s, usually at a resolution of 12 bits/sample. In order to transmit it without any compression, the required channel capacity is at least $12 \times f_s bits/sec$. The limited channel capacity brings the need of data compression to reduce the signal bandwidth while maintaining a required quality.

There are mainly three statistical characteristics which are used to compress signals,

- The histogram of a source. Analog signal values span a voltage range from $-V_{cc}$ to V_{cc}. After it is sampled at 12bits/sample, the values range from -2^{11} to $2^{11}-1$. Fortunately these values are not evenly distributed, leaving us a huge room for an efficient quantization.

- Inter-sample Correlation. Signal correlation among adjacent signal samples indicates redundancies between the samples. This redundancy can be exploited for source compression. The well-known source coding methods to exploit inter-sample correlation include delta modulation, differential pulse code modulation(DPCM), linear predictive coding(LPC), transform coding, and subband coding.

- Energy distribution in frequency domain. For many real sources signal energy tends to accumulate at the lower part of the frequency spectrum. This provides the basis for subband coding. By splitting the signal spectrum into several subbands, we observe that the subbands contain different amount of information, leading to the assignment of different bit rates for them. We further observe that the subband signals have different characteristics so that we can apply different coding methods to different subbands.

In this thesis, Subband signal decomposition and scalar quantization are used for speech coding. The thesis consists of two main parts, simulation and implementation. In the simulation part, an adaptive algorithm [7] is applied to define the irregular subband tree structure of a given signal source. For evaluation of irregular tree performance, a performance criterion developed by Akansu and Liu [7] is employed. We compare the energy compaction performance of irregular subband tree structure vs. the industry standard, DCT. In the implementation part, a fixed half-band tree structure is realized by using TI TMS320C25 general purpose DSP chip. The optimum bit allocation is calculated for each subband and the scalar quantization is applied to compress the bit rate. In both parts, Binomial Perfect Reconstruction(PR) Quadrature Mirror Filter(QMF) banks derived by Akansu et al.[5] have been employed for subband coding of speech.

This thesis is organized as follows, Chapter 2 covers the Perfect Reconstruction Quadrature Mirror Filter (PR-QMF) theory. Chapter 3 introduces sub-band coding and describes an adaptive irregular tree structuring algorithm. Chapter 4 introduces a benchmark for the proposed algorithm. Chapters 5 and 6 discuss the bit allocation and quantization issues respectively. Chapter 7 explains in detail, the implementation of subband speech codec. This chapter also examines the performance of this real-time system. Chapter 8 compares the energy compaction performance of subband decomposition with DCT. Discussions and conclusions are given in Chapter 8.

Chapter 2

Perfect Reconstruction Quadrature Mirror Filter Bank

2.1 Two Channel QMF Bank

The structure of a two channel QMF bank is shown in Fig.(2.2). It consists of two analysis (anti-aliasing) filters and two synthesis (interpolation) filters, down-samplers and up-samplers. Since the sampling rates throughout the system are not the same, the QMF bank is also known as a multirate digital filter bank. The following sections describe the down- and up-samplers and the perfect reconstruction requirements for the analysis and synthesis filters.

2.1.1 Down-sampler and Up-sampler

The down-samplers used in a half-band QMF are two-fold, i.e. the down-sampler extracts one out of two consecutive input samples and thus forms the output. Therefore the output of the two-fold down-sampler is a compressed version of the input. The input-output time relation of a down-sampler for the rate of 2 is

$$y(n) = x(2n) \tag{2.1}$$

Fig.(2.1a) shows the structure of an N-fold down-sampler. It is obvious that a down-sampler is not a time-invariant device. Thus, it cannot be represented by a transfer function. As we know, a compression in time-domain corresponds to an

$$y(n) = \begin{cases} x(\frac{n}{N}), & n = \text{multiple of } N \\ 0, & otherwise \end{cases}$$

(a) (b)

Figure 2.1: N fold down- and up-samplers

expansion in frequency domain as expressed

$$Y(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) + X(-e^{j\omega})] \tag{2.2}$$

It is clear that if the input signal is not band limited to half-band (either lower half or upper half), the down-sampling will cause aliasing at the output.

A two-fold up-sampler inserts a zero between two adjacent input samples. It causes the length of the output sequence to be doubled, and the power spectral density of the signal to be compressed by one half. The frequency domain input-output expression of a two-fold interpolator is given by

$$Y(e^{j\omega}) = X(e^{j2\omega}), \qquad or \qquad Y(z) = X(z^2) \tag{2.3}$$

An N-fold up-sampler is shown in Fig.(2.1b).

Before we discuss the perfect reconstruction requirements of the analysis and synthesis filters, it is desirable to spend some time on their fundamental use. As discussed earlier in this section, the down-sampler expands the spectrum of the input. This will cause an aliasing in the output if the input spectrum is not band limited to half of the original band (either lowband or highband). This is not

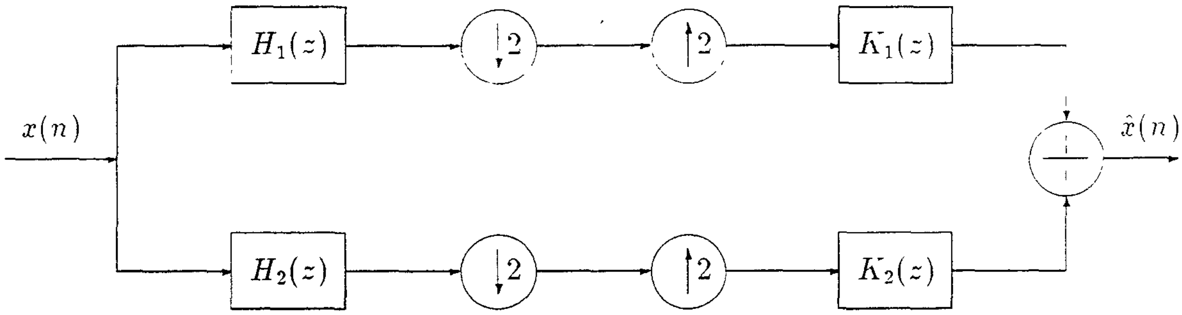


Figure 2.2: Two channel QMF bank

the case for many physical sources, including speech. The analysis filters serve the purpose of limiting the input signal to a lowpass and a highpass subsignals so that the aliasing effect is confined into a tolerable degree. The two-fold down sampling produces an image of the original spectrum which is to be filtered out by the interpolation filter, or the synthesis filter.

Perfect reconstruction Requirements

The perfect reconstruction conditions of half-band PR FIR filter bank were originally developed by Smith and Barnwell[4]. The input-output relation of 2-band filter bank can be easily followed with the help of Fig.(2.2). The reconstructed signal $\hat{x}(n)$, or its counterpart in Z, $\hat{X}(z)$, can be expressed as,

$$\hat{X}(z) = T(z)X(z) + S(z)X(-z) \tag{2.4}$$

where

$$\begin{aligned} T(z) &= \frac{1}{2}[H_1(z)K_1(z) + H_2(z)K_2(z)] \\ S(z) &= \frac{1}{2}[H_1(-z)K_1(z) + H_2(-z)K_2(z)] \end{aligned} \tag{2.5}$$

Perfect reconstruction requires that,

$$a. \quad S(z) = 0 \quad \text{for all } z$$

$$b. \quad T(z) \;=\; cz^{-n_0} \tag{2.6}$$

where c is a constant and n_0 is an integer.

If we choose

$$K_1(z) = -H_2(-z)$$

and

$$K_2(z) = H_1(-z)$$

The term $S(z)$ is set to zero.

For the requirement $b.$ with N odd, we choose

$$H_2 = z^{-N} H_1(-z^{-1}) \tag{2.7}$$

Therefore $T(z)$ becomes,

$$T(z) = \frac{1}{2} z^{-N} [H_1(z) H_1(z^{-1}) + H_1(-z) H_1(-z^{-1})] \tag{2.8}$$

The perfect reconstruction requirement of a half-band QMF now reduces to finding $H_1(z)$ such that

$$\begin{aligned} Q(z) &= H_1(z) H_1(z^{-1}) + H_1(-z) H_1(-z^{-1}) = \text{constant} \\ &= R(z) + R(-z) \end{aligned} \tag{2.9}$$

This selection implies that all four filters are causal whenever $H_1(z)$ is causal.

2.2 The Binomial-Hermite Family

The PR-QMF banks employed in this thesis are designed based on Binomial-Hermite sequences[2], which will effectively decrease the number of time-consuming multiplications.

The Binomial-Hermite sequences[2] are a family of finite duration discrete polynomials weighted by a Gaussian-like binomial envelope. They are generated by

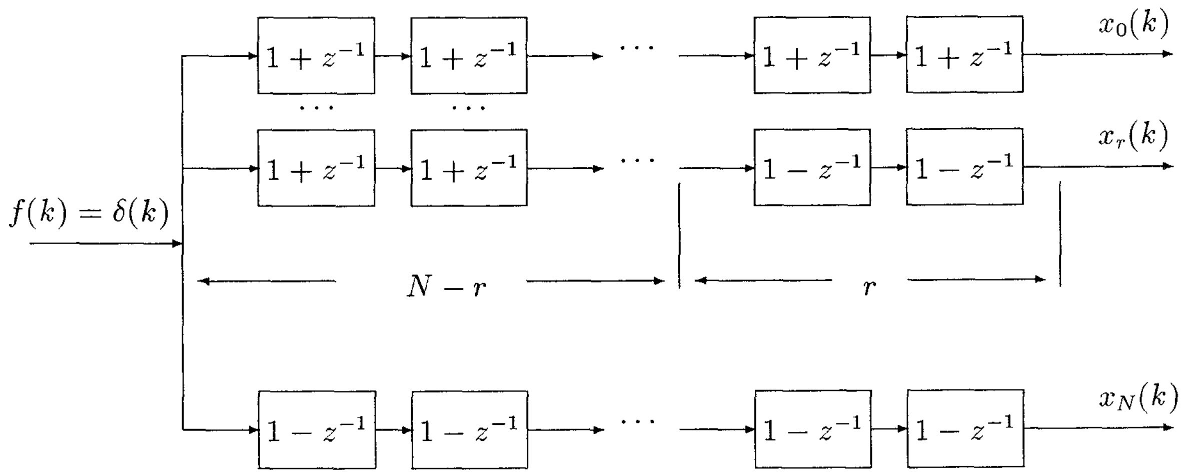


Figure 2.3: Block diagram of binomial filter structure

successive differencing of the binomial sequence defined as

$$x_0(k) = \begin{cases} \begin{pmatrix} N \\ k \end{pmatrix} = \dfrac{N!}{(N-k)!k!} & 0 \le k \le N \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

The general expression of binomial sequences is then

$$\begin{aligned} x_r(k) &= \nabla^r \begin{pmatrix} N-r \\ k \end{pmatrix} \\ &= \begin{pmatrix} N \\ k \end{pmatrix} \sum_{\nu=0}^{r} (-2)^\nu \begin{pmatrix} r \\ \nu \end{pmatrix} \frac{k^{(\nu)}}{N^{(\nu)}} \\ &= \begin{pmatrix} N \\ k \end{pmatrix} H_r(k) \qquad\qquad k = 0, 1, \cdots, N \end{aligned} \tag{2.11}$$

where $k^{(\nu)}$ is a polynomial in k of degree ν,

$$k^{(\nu)} = \begin{cases} k(k-1)\ldots(k-\nu+1) & \nu > 1 \\ 1 & \nu = 1 \end{cases} \tag{2.12}$$

The structure of the binomial filters are shown in Fig.(2.3). Note that the filter coefficients are either 1 or -1. Therefore no multiplication is needed in this part.

2.3 The Binomial QMF

To form the perfect reconstruction Binomial QMF [5], we observe that the first half of the binomial family spans the low frequency while the second half spans the high part of the spectrum. We therefore take the linear combination of the first half to form the low-pass filter and the second half the high-pass filter. Since the second half of the binomial family is simply the mirror image of the first half [5], it is only necessary to give the coefficients of the low-pass filter. We take as the half band filter

$$h_n = \sum_{r=0}^{\frac{N-1}{2}} \theta_r x_r(n)$$

or

$$H(z) = \sum_{r=0}^{\frac{N-1}{2}} \theta_r (1+z^{-1})^{N-r}(1-z^{-1})^r = (1+z^{-1})^{(N+1)/2} F(z) \tag{2.13}$$

and impose perfect reconstruction conditions Eq.(2.6) on it. Take $\theta_0 = 1$ for convenience.The perfect reconstruction condition gives a set of $\frac{N-1}{2}$ nonlinear algebraic equations, in the $\frac{N-1}{2}$ unknowns $\theta_1, \theta_2, \cdots, \theta_{\frac{N-1}{2}}$. From these, we can obtain the corresponding Binomial PR-QMFs.

These filters are implemented using FIR structure, as is the case in this thesis. The structure is shown in Fig.(2.4) for $N = 7$, or 8-tap filter. Wherein both low-pass and high-pass filters are simultaneously realized. Coefficient θ_0 can be set to unity, leaving θ_1 through θ_3 as tap weights. These are the only multiplications needed when using the Binomial network as the half-band QMF rather than the eight weights in a transversal structure. The obtained weight coefficients θ_r of 4, 6, 8-tap filters are shown in Table.(2.1)[5]

4-tap		
θ_r	set 1	set 2
θ_0	1	1
θ_1	$\sqrt{3}$	$-\sqrt{3}$

6-tap		
θ_r	set 1	set 2
θ_0	1	1
θ_1	$\sqrt{2\sqrt{10}+5}$	$-\sqrt{2\sqrt{10}+5}$
θ_2	$\sqrt{10}$	$\sqrt{10}$

8-tap				
θ_r	set 1	set 2	set 3	set 4
θ_0	1	1	1	1
θ_1	4.9892	-4.9892	1.0290	-1.0290
θ_2	8.9461	8.9461	-2.9705	-2.9705
θ_3	5.9160	-5.9160	-5.9160	5.9160

Table 2.1: Binomial weight coefficients θ_r for 4-, 6- and 8-tap Binomial-QMFs

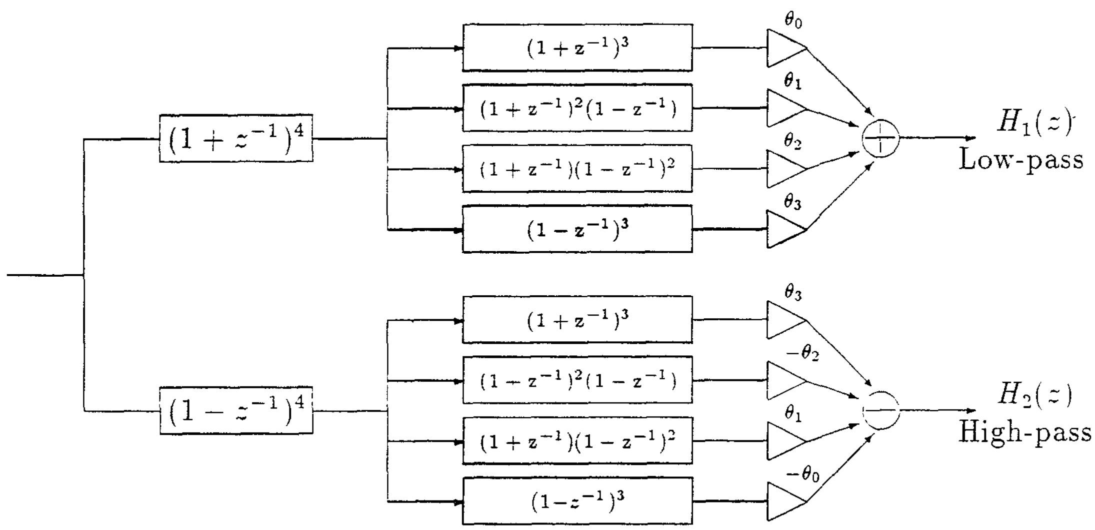


Figure 2.4: 8-tap low-pass and high-pass Binomial-QMFs

Chapter 3

Subband Tree Structures and Frequency Split

The basic idea of subband tree structuring is to split the signal into low and high frequency bands, then perform the similar splitting on subbands, until the desired frequency split is achieved. Different encoding method is applied to each subband signal according to their characteristics, and available bits are optimally allocated among bands.

This idea came from the unevenly distributed nature of signal energy. For example in speech signals, as shown in Fig.(3.1), most of signal energy spans only in the lower frequency band. On the other hand, one can observe that the higher frequency band of the signal is nearly flat, suggesting a semi-white noise to which little coding gain can be achieved. Therefore it is unnecessary to treat the upper half spectrum the same as the lower half, i.e. we can split the signal into lower and higher bands and encode them separately.

In this thesis, the perfect reconstruction quadrature mirror filter banks which were introduced in Chapter 2 are employed in hierarchical subband structures for speech coding.

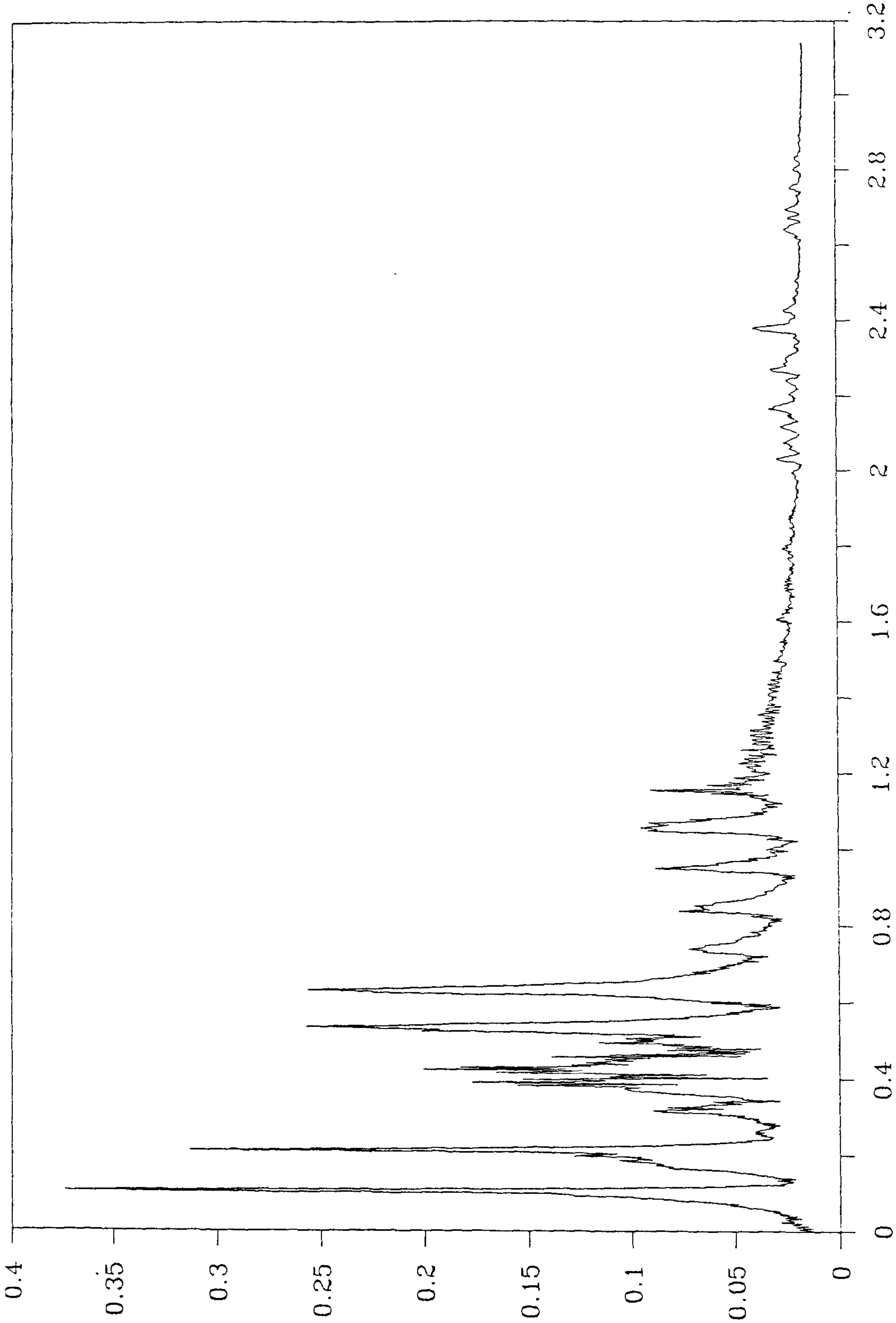


Figure 3.1: Power spectral density function of speech signal “child”

3.1 Sub-Band Coding

In sub-band coding the input signal spectrum is divided into several sub-bands by passing the signal through a bank of bandpass filters (with low- and high-pass filters as the special case). Each sub-band is then down-sampled to fit its Nyquist rate and encoded for transmission. At the receiver side, the decoded subband signals are first interpolated and then passed through a set of interpolation filters to remove the imaging effects. Then they are reassambled to form the output signal, which is expected to be a close replica of the original signal. Fig.(3.2) illustrates a block diagram of the subband encoder. The encoder consists of a bank of M bandpass filters, followed by sub-band encoders and a multiplexer. The receiver has the reverse stages of demultiplexing,decoding and bandpass filtering prior to sub-band additions.

The employment of perfect reconstruction(PR) quadrature mirror filter(QMF) banks eliminates the reconstruction error between the input and the output provided that there is no quantization and transmission (channel) errors. The block diagram of sub-band coding with PR-QMF banks is shown in Fig.(2.1).

Sub-band encoding offers significant advantages over the time domain algorithms. By subband tree structuring, the redundancy of the input signal is removed , and the subband signals are uncorrelated. By allocating the bits in subbands, the number of quantizer levels of each subband can be independently controlled according to their statistics. Moreover, the selection of the quantizer for a specific band can also vary.

3.2 Irregular Trees

After having the concept of regular tree structure, let us take a look again at speech signals. The typical examples shown here are in Fig.(3.1). We can observe that

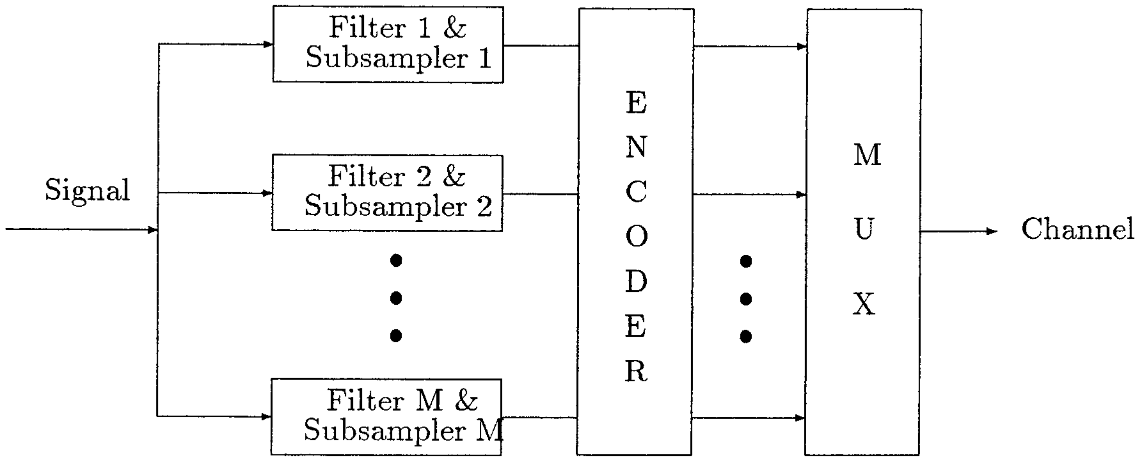


Figure 3.2: M-band subband encoder

most part of the signal energy is located in low band. Furthermore, even in the low frequency band, the energy is not equally distributed. In general, most of the practical signals have a bandlimited frequency spectrum, or have significant portion of their energy concentrated within several frequency bands, or packets. This fact reminds that full tree structure is not always necessary. Here we take into account a new algorithm introduced in [7] to perform irregular tree structuring according to input signal statistics.

The proposed algorithm is based on energy compaction criterion which will be described in Chapter 4. This algorithm assumes ideal filter banks and half-band frequency split all the time. If one considers an input signal with the known power spectral density $P_{xx}(\omega)$ and splits it into low and high half-bands with ideal filters, the corresponding energy compaction gain is given as

$$G_{SBC} = \frac{\sigma_x^2}{[\sigma_l^2 \sigma_h^2]^{\frac{1}{2}}} \tag{3.1}$$

where σ_x^2, σ_l^2, and σ_h^2 are average variance, low-band variance, and high-band variance, respectively.

In orthonormal signal decomposition, band variance is preserved. i.e., the average variances of each level of full tree structure are the same. In this case, σ_x^2 is given by

$$\sigma_x^2 = \frac{1}{2}(\sigma_l^2 + \sigma_h^2) \tag{3.2}$$

The low and high band variances can be calculated directly by Parseval's Energy Theorem,

$$\sigma_l^2 = \frac{1}{\pi}\int_0^{\pi/2} P_{xx}(\omega)d\omega \tag{3.3}$$

and

$$\sigma_h^2 = \frac{1}{\pi}\int_{\pi/2}^{\pi} P_{xx}(\omega)d\omega \tag{3.4}$$

The two band energy ratio is defined as

$$\eta = \frac{\sigma_h^2}{\sigma_l^2} \tag{3.5}$$

It can be seen that for a uniform power spectrum or white noise $\eta = 1$, $G_{SBC} = 1$.

The tree structuring algorithm for a given maximum frequency resolution $BW_{min} = \frac{\pi}{2^L}$ can be summarized as follows[7],

a) Measure the power spectral density, P_{xx} of the given ZERO MEAN source.

b) Calculate σ_l^2, σ_h^2 and find η. Compare η with a predefined threshold T:

If $\eta \leq T$, stop

If $\eta > T$, band split, obtain low and high bands.

c) Calculate $\sigma_{ll}^2, \sigma_{lh}^2, \sigma_{hl}^2, \sigma_{hh}^2$ band variances and the corresponding η_l and η_h. Check with the threshold T:

If $\eta_l \leq T$, stop

If $\eta_l > T$, band split, obtain ll and lh bands.

If $\eta_h \leq T$, stop

If $\eta_h > T$, band split, obtain hl and hh bands.

d) Repeat the procedure until the allowed maximum frequency resolution BW_{min} is reached. The corresponding tree structure based on input statistics is obtained.

The proposed algorithm is signal dependent and the regular tree of 2^L equal bands is adapted to the signal power spectrum within M unequal bands, $2^L > M$. Whenever $2^L = M$, irregular tree becomes a regular tree. It is observed that an irregular tree is a better practical choice. It is obvious that regular tree provides the upper compaction bound for known L.

Chapter 4

Performance Measure

To measure the goodness of the orthogonal transform or subband tree structuring, the introduction of a benchmark is necessary. In this thesis, an energy compaction measure, namely gain of subband coding over pulse code modulation, G_{SBC}, is applied to fulfill this requirement. The proposed measure is a common tool used to compare orthonormal signal decomposition techniques. It is first introduced to measure orthogonal transforms. This measure can be easily extended to multi-level multi-band regular tree as well as irregular tree structures.

An N_1 band orthonormal transform implies the variance preservation condition

$$\sigma_x^2 = \frac{1}{N_1} \sum_{k=0}^{N_1-1} \sigma_k^2 \tag{4.1}$$

where σ_x^2 is the input signal variance with zero mean and $\{\sigma_k^2\}$ are the band variances. If one assumes that all the bands and the input signal have the same *pdf* type, the distortion ratio of subband coding over PCM at the same bit rate can be easily obtained as

$$G_{SBC} = \frac{D_{PCM}}{D_{SBC}} = \frac{\sigma_x^2}{[\prod_{k=0}^{N_1} \sigma_k^2]^{\frac{1}{N_1}}} \tag{4.2}$$

If each band of the first level decomposition tree goes through an N_2 band decomposition, G_{SBC} can be easily extended for this case. The input variance now is connected to the variances of $N_1 \times N_2$ bands as

$$\sigma_x^2 = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sigma_{k_1 k_2}^2 \tag{4.3}$$

An orthonormal transform assures that the average of the analysis bands' quantization errors is equal to the reconstruction error of the signal after the synthesis operations and shown as

$$\sigma_r^2 = \sigma_q^2 = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sigma_{qk_1 k_2}^2 \tag{4.4}$$

To compare the energy compaction ratio of subband coding with PCM, we assume the same *pdf* type for the input signal of PCM and get the corresponding distortion for PCM at the same bit rate as

$$\sigma_{qPCM}^2 = \epsilon^2 2^{-2B} \sigma_x^2 \tag{4.5}$$

Applying optimum bit allocation in SBC the distortion gain or ratio of the orthogonal transform is follows[7]

$$max\{G_{SBC}\} = \frac{\sigma_{qPCM}^2}{min\{\sigma_q^2\}} = \frac{\frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sigma_{k_1 k_2}^2}{[\prod_{k_1=0}^{N_1-1} \prod_{k_2=0}^{N_2-1} \sigma_{k_1 k_2}^2]^{\frac{1}{N_1 N_2}}} \tag{4.6}$$

It can be shown that if the process repeats regularly for L levels with the number of bands in each decomposition as $N_i, \quad i = 1, 2, \cdots, L$, this expression is generalized for $\prod_{i=1}^{L} N_i$ bands as follows[7]

$$max\{G_{SBC}\} = \frac{[\prod_{i=1}^{L}]^{-1} \{ \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \cdots \sum_{k_L=0}^{N_L-1} \sigma_{k_1 k_2 \quad k_L}^2 \}}{[\prod_{k_1=0}^{N_1-1} \prod_{k_2=0}^{N_2-1} \cdots \prod_{k_L=0}^{N_L-1} \sigma_{k_1 k_2 \quad k_L}^2]^{[\prod_{i=1}^{L} N_i]^{-1}}} \tag{4.7}$$

This expression is valid for any regular tree structure of orthogonal filter banks or block transforms.

Similar expression for the irregular tree structures can be obtained. The case considered here assumes an N_1-band orthogonal decomposition in the first level of the tree and only bank p decomposed into N_2 bands again in the second level of the tree. Since any middle level node in the tree means an orthonormal decomposition, overall average distortion of the scenario considered can be similarly written as before

$$\sigma_q^2 = \frac{1}{N_1} \sum_{k_1=0,k_1\neq p}^{N_1-1} \sigma_{qk_1}^2 + \frac{1}{N_1 N_2} \sum_{k_2=0}^{N_2-1} \sigma_{qpk_2}^2 \tag{4.8}$$

and the band distortion terms with the same type of *pdf* assumption

$$\sigma_{qk_1}^2 = \epsilon^2 2^{-2B_{k_1}} \sigma_{k_1}^2 \qquad \begin{array}{rcl} k_1 & = & 0, 1, \cdots, N_1 - 1 \\ k_1 & \neq & p \end{array} \tag{4.9}$$

and

$$\sigma_{qpk_2}^2 = \epsilon^2 2^{-2B_{pk_2}} \sigma_{pk_2}^2 \qquad k_2 = 0, 1, \cdots, N_2 - 1 \tag{4.10}$$

When the bit allocation among the bands are optimally distributed, one get the average distortion as

$$min\{\sigma_q^2\} = \epsilon^2 2^{-2B} [(\prod_{k_2=0}^{N_2-1})^{\frac{1}{N_2}} (\prod_{k_1=0,k_1\neq p}^{N_1-1} \sigma_{k_1}^2)]^{\frac{1}{N_1}} \tag{4.11}$$

Assuming the same *pdf* type again for the original signal provides

$$max\{G_{SBC}\} = \frac{\sigma_{qPCM}^2}{min\{\sigma_q^2\}} = \frac{\sigma_x^2}{[(\prod_{k_2=0}^{N_2-1})^{\frac{1}{N_2}} (\prod_{k_1=0,k_1\neq p}^{N_1-1} \sigma_{k_1}^2)]^{\frac{1}{N_1}}} \tag{4.12}$$

This result can be extended to any arbitrary tree structure and the regular tree is regarded as a special case.

Chapter 5

Optimum Bit Allocation

The ultimate goal of subband coding scheme is to reduce the bandwidth of the transmitted data. Due to uneven subband energies, it is desirable to allocate available bits accordingly. In this way, the sub-bands containing more energy are allocated with more bits so that the majority of the signal energy is retained. The procedure of quantization is two-fold:

- Determination of the bit allocation among the sub-bands.
- Quantization of sub-band signals.

The determination of bit allocation and the selection of quantizers are based on the statistics of sub-band signals, i.e., the mean and variance of sub-band signals. This chapter introduces bit allocation and its application on irregular trees.

5.1 Bit Allocation

To minimize the introduced reconstruction error the bitrates of subband signals have to be allocated according to their variances assuming an overall constant bitrate. This leads to the concept of adaptive bit allocation. If each sub-band has the same bandwidth, the average bitrate is given by

$$\bar{R} = \frac{1}{N} \sum_{k=0}^{N-1} R_k \tag{5.1}$$

in which $\bar{R}$ is preassigned. Here R_k stands for the bitrate of subband k. The goal of the bit allocation is to adjust the values of R_k. Take the simple case as $N = 2$, the problem is to minimize[3]

$$\sigma_r^2 = \frac{1}{2}(\sigma_{q0}^2 + \sigma_{q1}^2) \tag{5.2}$$

The quantization error introduced by a single quantizer can be approximated by [3]

$$\sigma_{qk}^2 = \epsilon^2 2^{-2R_k} \sigma_k^2 \tag{5.3}$$

Thus, given that $\bar{R} = \frac{1}{2}(R_0 + R_1)$,

$$\sigma_r^2 = \frac{1}{2}\epsilon^2(2^{-2R_0}\sigma_0^2 + 2^{2R_0 - 4\bar{R}}\sigma_1^2) \tag{5.4}$$

Differentiating Equation (5.4) with respect to R_0 and set it equal to zero, the following is obtained.

$$-\sigma_0^2 2^{-2r_0} + \sigma_1^2 2^{2R_0 - 4\bar{R}} = 0 \tag{5.5}$$

Solving Equation (5.5), and using the fact that $\bar{R} = \frac{1}{2}(R_0 + R_1)$ for R_0 and R_1

$$\begin{aligned} R_0 &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_0}{\sigma_1} \\ R_1 &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_1}{\sigma_0} \end{aligned} \tag{5.6}$$

We further obtain from Eq.(5.5) that

$$\sigma_0^2 2^{-2R_0} = \sigma_1^2 2^{-2R_1} \tag{5.7}$$

with the expression in Eq.(5.3) we can get

$$\sigma_{q0}^2 = \sigma_{q1}^2 \tag{5.8}$$

which means that each quantizer contributes equally to the reconstruction error.

For larger values of N the bit allocation requires the use of Lagrange multiplier method to obtain a criterion for optimum bit allocation which leads to the following expression [3]

$$R_k = \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_k^2}{[\prod_{j=0}^{N-1} \sigma_j^2]^{1/N}} \tag{5.9}$$

In order to evaluate the performance of an irregular tree structure, it is useful to compare the signal-to-noise ratio with PCM at the same bitrate. Assuming identical quantizer performance, the reconstruction error for PCM is given by

$$\sigma_{r,PCM}^2 = \epsilon^2 2^{-2R} \sigma_x^2 \tag{5.10}$$

In this case the average reconstruction error is given by Eq.(5.2)

$$\sigma_r^2 = \frac{1}{2}(\sigma_{q0}^2 + \sigma_{q1}^2) = \sigma_{q0}^2 \approx \epsilon^2 2^{-2R_0} \sigma_0^2 \tag{5.11}$$

since σ_{q0}^2 and σ_{q1}^2 are identical as stated in Eq.(5.8). Using Eq.(5.6) we get

$$\sigma_{r,SBC}^2 = \epsilon^2 2^{-2\bar{R}} \sigma_0 \sigma_1 \tag{5.12}$$

Thus the coding gain of half-band one step band tree over PCM is

$$G_{SBC} = \frac{\sigma_{r,PCM}^2}{\sigma_{r,SBC}^2} = \frac{\epsilon^2 2^{-2\bar{R}} \sigma_x^2}{\epsilon^2 2^{-2\bar{R}} \sigma_0 \sigma_1} = \frac{\sigma_x^2}{\sqrt{\sigma_0^2 \sigma_1^2}} \tag{5.13}$$

For equal length sub-bands with band number of $N+1$, The above equation can be generalized as follows[3]

$$G_{SBC} = \frac{\sigma_x^2}{[\prod_{k=0}^{N} \sigma_k^2]^{1/(N+1)}} \tag{5.14}$$

where

$$\sigma_x^2 = \frac{1}{N+1} \sum_{k=0}^{N} \sigma_k^2$$

5.2 Modification For Irregular Trees

In the real-time subband codec implementation, we are using a 5-band tree structure, which is shown in Fig.(5.1). Since the derived sub-bands are not of equal length, the bit allocation expression derived in Section 5.1 should be modified.

Because of the downsampling in each analysis level, the length of the subband signal is $\frac{1}{2}$ of the signal in the last level. They contribute differently to the average bitrate. Therefore in our case here, the expression of average bit allocation is

$$\begin{aligned}\bar{R} &= \frac{1}{2}(R_h + \frac{1}{2}(R_{lh} + \frac{1}{2}(R_{llh} + \frac{1}{2}(R_{lllh} + R_{llll})))) \\ &= \frac{1}{2}R_h + \frac{1}{4}R_{lh} + \frac{1}{8}R_{llh} + \frac{1}{16}R_{lllh} + \frac{1}{16}R_{llll} \end{aligned} \tag{5.15}$$

Applying Lagrange multiplier technique, we find that the bit allocations of subbands

$$\begin{aligned}
R_h &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_h^2}{(\sigma_h^2)^{\frac{1}{2}}(\sigma_{lh}^2)^{\frac{1}{4}}(\sigma_{llh}^2)^{\frac{1}{8}}(\sigma_{lllh}^2\sigma_{llll}^2)^{\frac{1}{16}}} \\
R_{lh} &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_{lh}^2}{(\sigma_h^2)^{\frac{1}{2}}(\sigma_{lh}^2)^{\frac{1}{4}}(\sigma_{llh}^2)^{\frac{1}{8}}(\sigma_{lllh}^2\sigma_{llll}^2)^{\frac{1}{16}}} \\
R_{llh} &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_{llh}^2}{(\sigma_h^2)^{\frac{1}{2}}(\sigma_{lh}^2)^{\frac{1}{4}}(\sigma_{llh}^2)^{\frac{1}{8}}(\sigma_{lllh}^2\sigma_{llll}^2)^{\frac{1}{16}}} \\
R_{lllh} &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_{lllh}^2}{(\sigma_h^2)^{\frac{1}{2}}(\sigma_{lh}^2)^{\frac{1}{4}}(\sigma_{llh}^2)^{\frac{1}{8}}(\sigma_{lllh}^2\sigma_{llll}^2)^{\frac{1}{16}}} \\
R_{llll} &= \bar{R} + \frac{1}{2}\log_2 \frac{\sigma_{llll}^2}{(\sigma_h^2)^{\frac{1}{2}}(\sigma_{lh}^2)^{\frac{1}{4}}(\sigma_{llh}^2)^{\frac{1}{8}}(\sigma_{lllh}^2\sigma_{llll}^2)^{\frac{1}{16}}}
\end{aligned} \tag{5.16}$$

These expressions are readily enough for the adaptive determination of band bitrates if the average bitrate is not too low and the band energies do not differ too much. When these prerequisites are not satisfied, the band bitrate may become negative for some subbands. Since negative bitrate is meaningless in quantization, we have to reallocate the bitrates while setting the negative bitrates to zero.

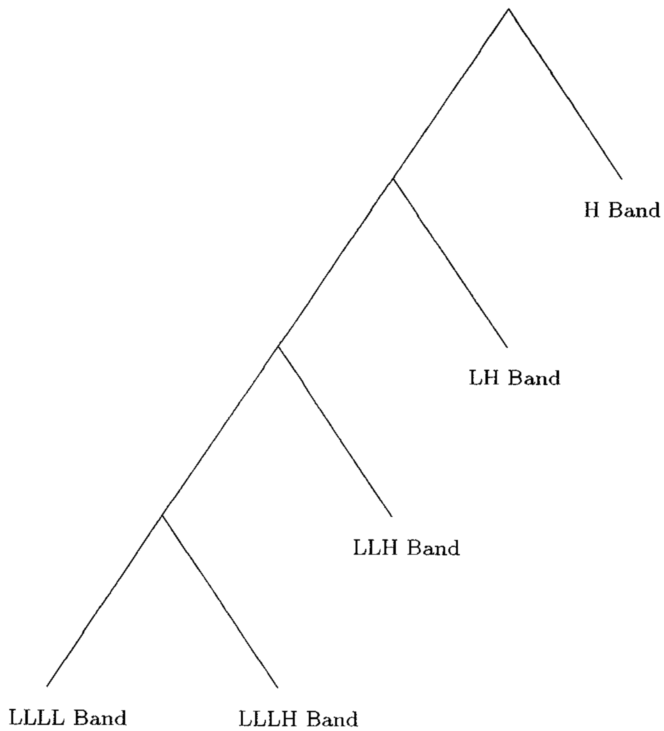


Figure 5.1: Implementation of 5 band subband tree structure

Chapter 6

Quantization

6.1 Introduction

After the optimal bit allocation of sub-bands is calculated, the quantization is performed. Quantization is the most common form of data compression and is fundamental to any digitization scheme or lossy data compression system. Quantization introduces some degree of information loss, and therefore it is not reversible.

The basic idea of quantization is to map a continuous variable u into a discrete counterpart $\dot{u}$, which belongs to a finite set of possible values $\{r_1, \cdots, r_L\}$. During transmission, the indices $1, 2, \cdots, L$ of quanta values are sent instead of the value itself. This procedure significantly reduces the transmission bandwidth.

When a mapping or a quantizer is defined, the input signal u is compared with a set of increasing threshold values $t_k, k = 1, 2, \cdots, L$ with t_1 as the minimum and t_L the maximum values of the interval. If u lies within the interval $[t_k, t_{k+1})$, then it is mapped onto r_k, the k^{th} reconstruction or quanta level.

In this thesis, we consider only zero memory (scalar) quantizers, which depend only on the current input to generate the output, with no consideration of any previous or subsequent inputs. The quantizer of each sub-band operates at a bitrate allocated to it. The assigned quantizer should, at the given bitrate, minimize the error between the input u and the output $\dot{u}$.

6.2 The Optimum Mean Square Quantizer

The criterion of quantizer performance we discuss here is the mean square error between the input signal and its quantized version. Defining u as a random variable with continuous density function $p_u(u)$, the mean square error is defined as [3]

$$\epsilon \stackrel{\text{def}}{=} E[(u - \dot{u})^2] = \sum_{i=1}^{L} \int_{t_i}^{t_i+1} (u - r_i)^2 p_u(u) du \tag{6.1}$$

where $L \equiv$ resolution of the quantizer. Differentiating the above equation with respect to t_k and r_k and equating the result to zero, we obtain

$$\begin{aligned} \frac{\partial \epsilon}{\partial t_k} &= (t_k - r_{k-1})^2 p_u(t_k) - (t_k - r_k)^2 p_u(t_k) = 0 && (6.2) \\ \frac{\partial \epsilon}{\partial r_k} &= 2 \int_{t_k}^{t_{k+1}} (u - r_k) p_u(u) du = 0 \qquad 1 \leq k \leq L && (6.3) \end{aligned}$$

Using the fact that $t_{k-1} \leq t_k$, the following equations for the threshold and reconstruction levels are obtained[3]

$$\begin{aligned} t_r &= \frac{r_k + r_{k+1}}{2} && (6.4) \\ r_k &= \frac{\int_k^{k+1} u p_u(u) du}{\int_{t_k}^{t_{k+1}} p_u(u) du} && (6.5) \end{aligned}$$

These results state that the optimum threshold levels lie halfway between the optimum reconstruction levels, which, in turn, lie at the center of mass of the probability density in the interval levels. Both Eq.(6.4) and Eq.(6.5) are nonlinear equations that have to be solved simultaneously.

The optimum mean square quantizer has several interesting properties as:

1. The output of the quantizer is an unbiased estimate of the input, i.e.,

$$E[\dot{u}] = E[u] \tag{6.6}$$

2. The quantization error is orthogonal to the quantizer output, i.e.,

$$E[(u - \dot{u})\dot{u}] = 0 \tag{6.7}$$

3. The variance of the quantizer output is reduced by the factor of $1 - f(B)$, where $f(B)$ denotes the mean square distortion of the $B-$bit quantizer for unity variance inputs:

$$\sigma_{\dot{u}}^2 = [1 - f(B)]\sigma_u^2 \tag{6.8}$$

Since this thesis involves real-time implementation. The time consuming iteration for the solution of Eqs.(6.4) and (6.5) is nearly impossible. However, for common probability density functions with zero mean and unit variance, tables for the optimum design values for different resolutions are readily available in the literature.

The commonly used quantizers for speech and image models are Gaussian and Laplacian quantizers, which correspond to Gaussian and Laplacian distributions, respectively. The Laplacian probability density function is defined as

$$p_u = \frac{\alpha}{2} exp(-\alpha|u - \mu|) \tag{6.9}$$

where μ is the mean of u, and the parameter α is defined as

$$\sigma^2 = \frac{2}{\alpha} \tag{6.10}$$

The Gaussian distribution is defined as

$$p_u = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(u - \mu)^2}{2\sigma^2}) \tag{6.11}$$

After studying the band histograms of several test speech, we choose Gaussian quantizer for LLLL and LLLH bands and Laplacian quantizer for the other bandpass subbands. The band histograms of h-band and llll-band in speech "one" for are presented in Figs. (6.1) and (6.2) for justification.

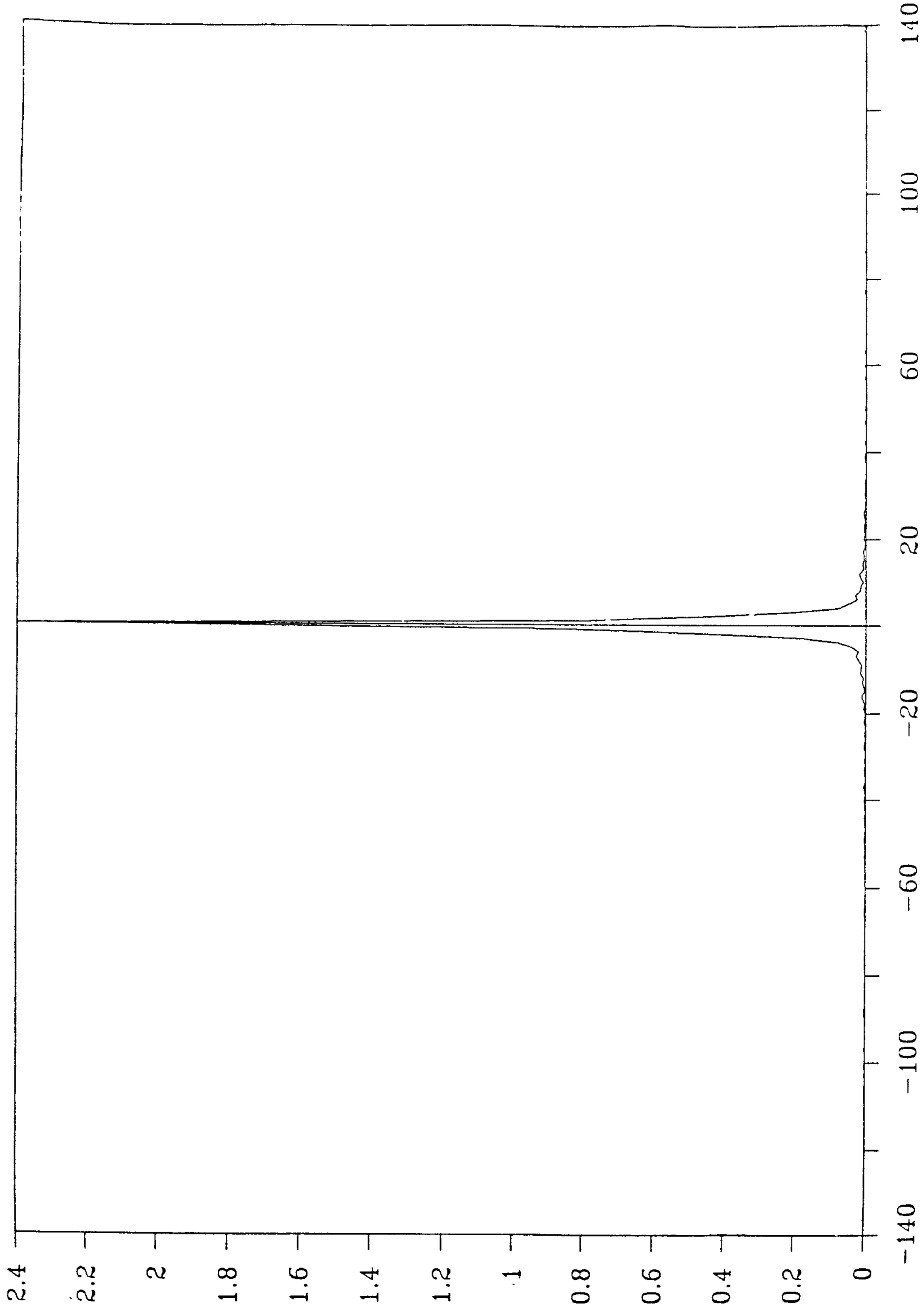


Figure 6.1: Band histogram of h-band in speech "one"

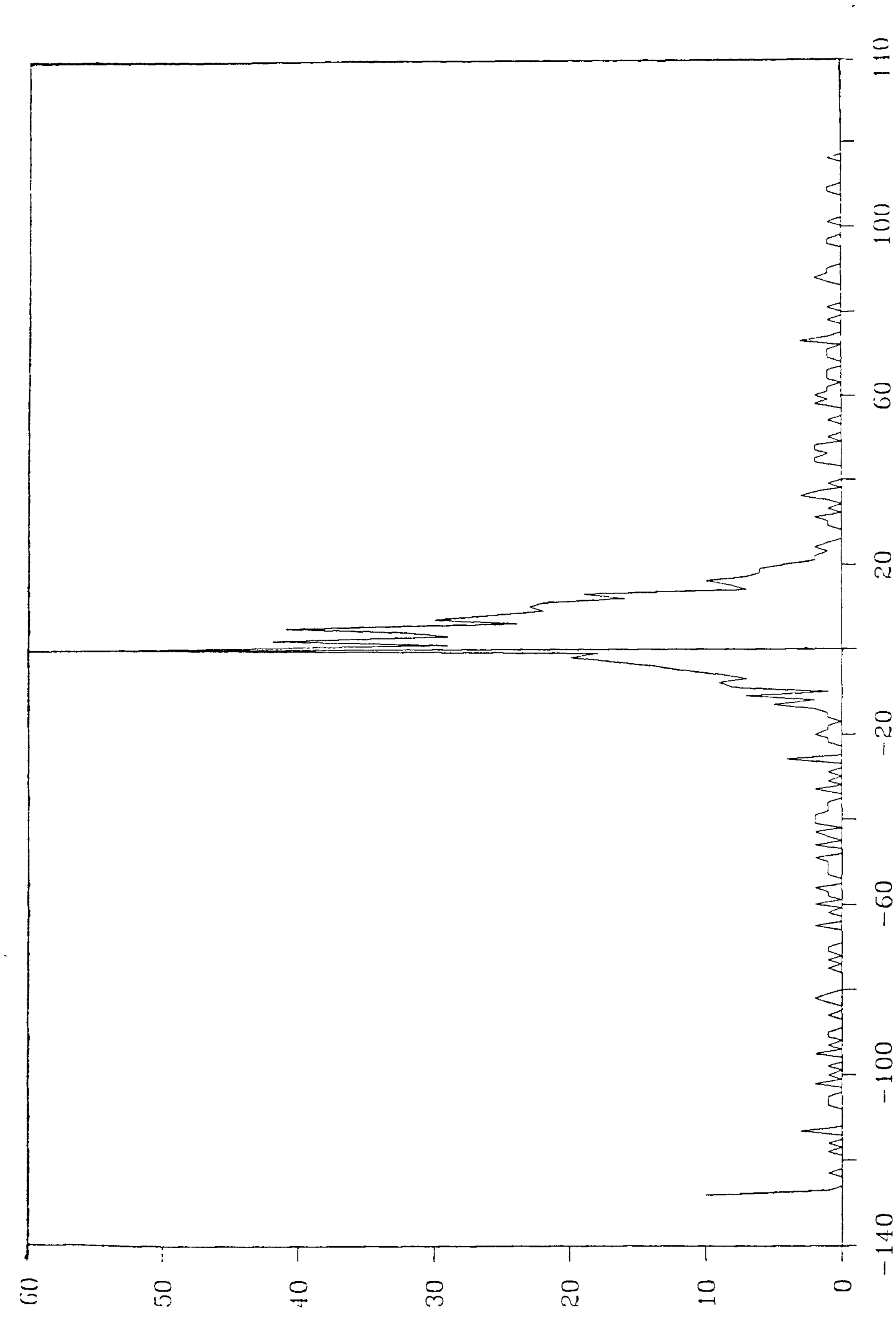


Figure 6.2: Band histogram of llll-band in speech "one"

Chapter 7

Implementation Issues on TMS320C25

This chapter describes in detail the real-time realization of a subband speech codec on TI's TMS320C25 DSP chip.

7.1 Introduction to TMS320C25

The digital signal processor used in this thesis is TMS320C25.It is a 32-bit CMOS single-chip digital signal processor. It features a Harvard-type architecture that maximizes processing power by maintaining two separate memory bus structure and the hardware-intensive instruction set. It provides an execution speed of 12.8 MIPS (million instructions per second).[9]

The advantage of Harvard-type structure over the traditional Von Neuman structure is mainly due to its feature of two separate buses, i.e., data bus and program bus. This enables TMS320C25 to fetch the opcode and data simultaneously. This cuts down the memory accessing time considerably.

In digital signal processing, the calculation of convolution, correlation and filtering require the same "multiplication-then-summation" operation. This is supported by TMS320C25 saliently by providing a multiplier-accumulator structure, which enables the multiplication and summation to be done at the same time. Taking the instruction MAC as an example, it simultaneously reads in two values, multiplies them, while adding the previous result to the accumulator. All is done in a single operation.

For the purpose of Fourier Transform, TMS320C25 also provides a new structure called bit-reversed indexed-addressing mode, which eliminates the need to rearrange the data after an FFT butterfly operation.

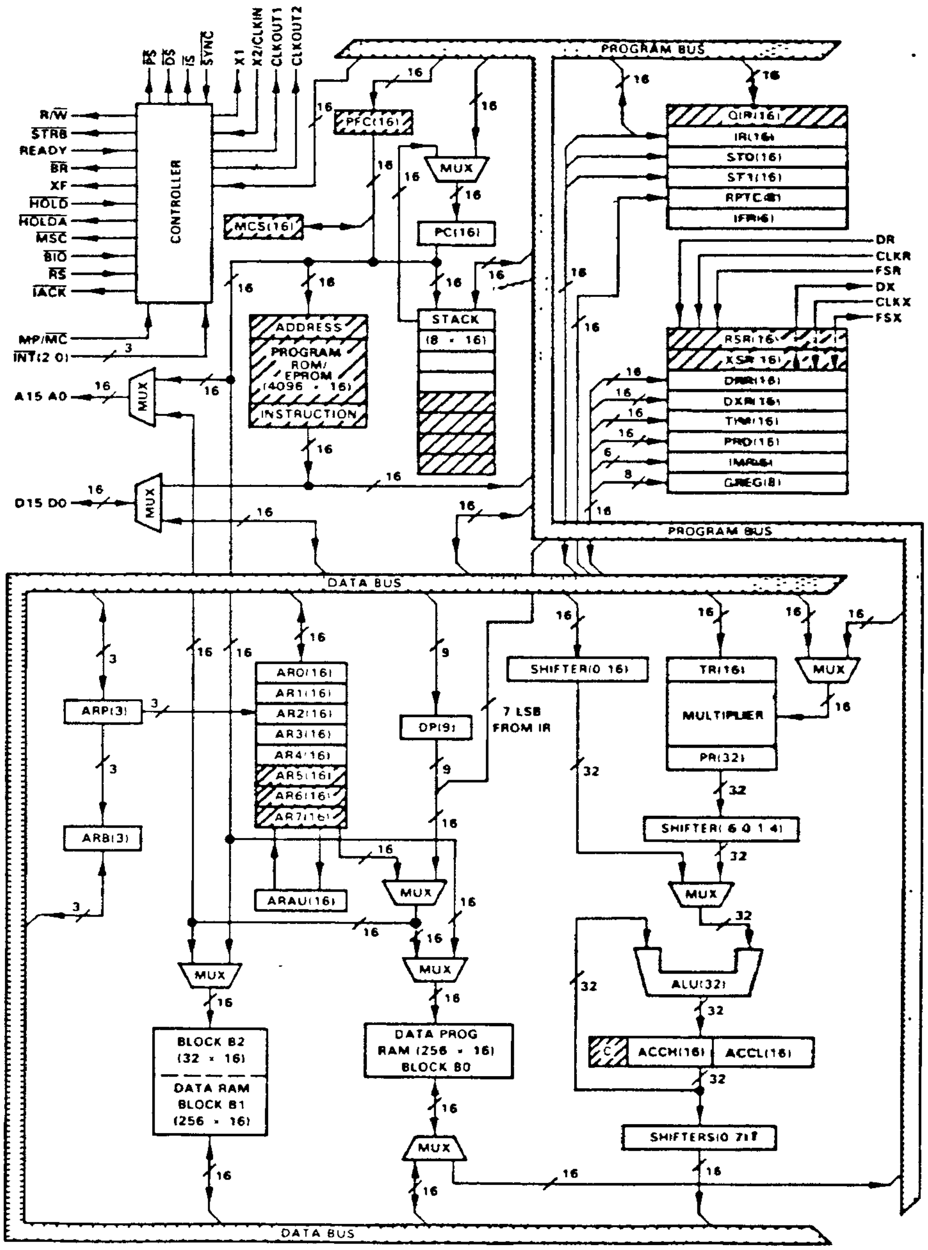
As indicated in Fig.(7.1), the TMS320C25 performs two's-complement arithmetic using 32-bit ALU/accumulator and multiplier. The ALU is a general-purpose arithmetic unit that operates using 16-bit words taken from data RAM or derived from immediate instructions or using the 32-bit result of the multiplier's product register. The accumulator stores the output from the ALU and is the second input to the ALU. The accumulator is of length 32 bits and is divided into a high-order (bits 31 to 16) and a low-order (bits 15 to 0) word. These two words can be accessed separately or as a whole. The multiplier performs 16×16-bit multiplication and stores the result in a 32-bit P reg (production register) in a single instruction cycle.

The TMS320C25 scaling shifter has a 16-bit input connected to the data bus and a 32-bit output connected to the ALU. The scaling shifter produces a left-shift of 0 to 16 bits on the input data. The LSBs of the output are filled with zeros, and the MSBs are sign-extended in the sign-extension mode.

The hardware stack has 8 16-bit words for saving the contents of the program counter during interrupts and subroutine calls. The corresponding PUSH and POP instructions can also save the contents of the accumulator when invoked.

TMS320C25 also features a set of 8 16-bit auxiliary registers AR0 to AR7 used for indirect memory addressing. The auxiliary registers are so well organized that they also provide array operations. These auxiliary registers are pointed to by an auxiliary register pointer (ARP) which activates one AR at a time.[9]

Architecture - Block Diagram



†Shifters on TMS32020 (0, 1, 4)

NOTE Shaded areas are for TMS320C25 and TMS320E25

Figure 7.1: TMS320C25 Block Diagram

7.2 The Subband Analysis/Synthesis Implementation

The implementation of subband Analysis/synthesis structure is divided into 3 major parts

1. Subband analysis of input signal
2. Quantization of subband signals
3. Synthesis from the quantized subband signals into output signal

Due to the time constraint stated above, it is impossible to apply adaptive algorithm to tree structuring. Here we use a fixed tree structure. We go four levels down on the analysis, which brings about $2^4 = 16$ bands if a full tree is applied. After studying the power spectral density functions of various speech signals, we found that the major parts of speech energy tend to gather into lower bands. Therefore the following fixed tree is used which contains only five bands: LLLL, LLLH, LLH, LH, H. The notation here, L stands for low band and H for high band, and the first symbol represents the first level in tree, second symbol for second level, and so forth. LLH, for example, means that the subsignal is formed by first low pass filtering the input and then down sample by 2, low pass and down sample again, then high pass and down sample. The block structure used in this implementation is shown in Fig.(2.3).

7.2.1 Analysis Stage

The filtering is implemented by using 8-tap perfect reconstruction Binomial quadrature mirror filters[5].

The decimation issue is carried out by introducing a set of even/odd indicators which are toggled every time they are accessed.

In order to get updated statistics, i.e., mean and variance, of each subband for quantization use, a reasonable length of subband data should be saved. In order to simplify computation in computing statistics, a length of an integer power of 2 is used for subband data buffer. For LLLL and LLLH bands, 8 words are reserved for each of them. Since the buffer length is doubled for an upper level, the length for LLH, LH and H band are 16, 32, 64 words, respectively.

For the PR-QMF used, although a perfect reconstruction is achieved, the delay of the output is inevitable. According to the discussion in Chapter 2, the delay is 7 samples for each level. There is no problem if we are dealing with full trees. But for the 5-band octave-band structure, that is different. Let us consider 2 levels for simplicity. The input goes to L and H bands, then l band is further divided into LL and LH. In the synthesis the LL and LH are added to form $\tilde{L}$, which is a delayed version of L band, and the delay is 7 samples. Now we address the problem. We cannot just combine $\tilde{L}$ and H to form an output. Instead, we must delay the H band by 7 samples to keep the same step with $\tilde{L}$. Then we can add them as we did to LL and LH bands to form the output, which is the perfect replica of the input, except delayed by 14 samples.

To overcome this fact, separate read and write pointers are assigned to LLH, LH and H bands. For LLH band, a 7-ample delay is introduced, so the write pointer is 7 samples behind the read pointer of LLH band. For LH and H bands, 14 and 21 delays are set for the write pointers.

Consider buffer length together with the delay issue. We get the final format of the buffer for each subband: LLLL and LLLH bands are of length 8 each; LLH band requires a buffer of $16 + 7 = 23$; LH band requires $32 + 14 = 46$; and the buffer length of H band is $64 + 21 = 85$ words. Each time the buffer is accessed, the corresponding pointer is increased by 1. After writing to the LH band, for example, the write pointer for the LH band is increased by one. After reading out a data from the LH band, the read pointer for the LH band is increased. When the pointer goes back to the bottom of the buffer, it is adjusted to point to the first address of the buffer. The extended buffer length (7 for LLH, 14 for LH and 21 for H) is based on the updating procedure.

7.2.2 Synthesis Stage

The synthesis stage is a multi-entry structure. The basic elements of this stage are upsampling by 2 and filtering or interpolation. Filtering in the synthesis stage is almost the same as in analysis.

To introduce the multi-entry structure for the synthesis stage, let us consider what the analysis and synthesis stages do when the signal is applied. The first sample is fetched from A/D to memory. Then, it is fed into the low and high pass analysis filters. Then the first level indicator A1 is checked. If A1 is zero, the filter outputs are discarded and zeros are fed into the synthesis filters to get the system output. Note that this time no operation below the level 1 is taken. If A1 is not zero, the high pass filter output is saved while the low pass filter output is fed into the second level analysis filter. This time the program checks A2 instead of A1 to decide whether to jump to synthesis level 2 with 0 fed or go further to analysis level 3. We can see that the entries of the synthesis stage are not the same when A1 is zero or non-zero. The similar case is also valid for A2, A3, and A4.

Another consideration of the synthesis stage is the realization of the 1:2 interpolator. For convenience, we use another set of 4 even/odd indicators, S1, S2, S3, and S4, for the synthesis levels 1, 2, 3, 4, respectively. The functions of these indicators are similar to A1 through A4 in the analysis stage. When S1, for example, is 0, the input of the synthesis filter is set to zero, otherwise, the input is the output of the corresponding analysis level. A table is made to illustrate the behavior of these 8 indicators and the operations followed. (See Table (7.1))

It is obvious from the table that the values of the analysis and synthesis indicators of the same level are identical. So we can use only 4 indicators, one for each level, in order to save memory.

7.2.3 Quantization

Up to this point, the reconstruction is succeeded perfectly, i.e., without any quantization. Now we include the quantization step, which serves for the ultimate goal of bit reduction.

The quantization routine consists of two steps, which will be discussed separately: bit allocation and quantization of subband signals. Both of these two steps require the statistics of subband data, i.e. mean and variance of LLLL, LLLH, LLH, LH and H band signals. First we talk about the computation of the subband mean and variance.

Time Averaging

The updated statistics of a signal $x(n)$ are obtained by calculating the time averages of $x(n)$ and $x^2(n)$. The time averages of $x(n)$ and $x^2(n)$ are given by

$$\begin{aligned} < x(kN) > &= \frac{1}{N}\sum_{i=0}^{N-1} x(n-i) \qquad k = \pm 0, 1, \cdots \\ < x^2(kN) > &= \frac{1}{N}\sum_{i=0}^{N-1} x^2(n-i) \qquad k = \pm 0, 1, \cdots \end{aligned} \tag{7.1}$$

and

$$\sigma_x^2 =< x^2 > - < x >^2 \tag{7.2}$$

To get a reasonable averaging length, two factors are considered. The first is that the averaging durations for all five bands are not the same. Actually, LLH band has the length of twice the length of LLLL or LLLH bands. LH band doubles the length of LLH band while H band doubles LH. Because of the intrinsic nonstationarities of speech signal, the stationary assumption can only be made when the length of the block is less then a glottal period, which is approximately 100 ms. Assuming a sampling period of 8000Hz, this length corresponds to 80 to 120 samples. Note that

Cycle	A1	A2	A3	A4	S1	S2	S3	S4	synthesis entry
1	0	0	0	0	0	0	0	0	4
2	1	0	0	0	1	0	0	0	3
3	0	1	0	0	0	1	0	0	4
4	1	1	0	0	1	1	0	0	2
5	0	0	1	0	0	0	1	0	4
6	1	0	1	0	1	0	1	0	3
7	0	1	1	0	0	1	1	0	4
8	1	1	1	0	1	1	1	0	1
9	0	0	0	1	0	0	0	1	4
10	1	0	0	1	1	0	0	1	3
11	0	1	0	1	0	1	0	1	4
12	1	1	0	1	1	1	0	1	2
13	0	0	1	1	0	0	1	1	4
14	1	0	1	1	1	0	1	1	3
15	0	1	1	1	0	1	1	1	4
16	1	1	1	1	1	1	1	1	1

Table 7.1: Updating of even/odd indicators and entry of synthesis stage during processing cycles .

the signal length is twice H band length, and one can get relationship for LH, LLH, LLLH, and LLLL bands. Secondly, the averaging length should not be too short. Say the length of 4 is not a proper length for statistics. Compromising the two requirements, with the consideration of easy realization, the statistical updating durations of LLLL and LLLH bands is chosen to be 8. The others are then LLH of length 16, LH of 32 and H of 64. The choice of 8, 16, 32, and 64 is justified because they are the integer powers of 2, so it can be achieved without any operation except shift. To divide by 8, for example, is simply right shift by 3.

To ensure the maximum sampling rate, the computation load should be divided evenly to each sampling period. Consequently, we use the pipeline mode in the computation of statistics. i.e., each time a subband sample is saved, it is added to the calculation of $\sum_{i=0}^{N-1} x_i$ and $\sum_{i=0}^{N-1} x_i^2$. A block counter is checked there after. If the counter expires, which means the N new samples are ready, the calculations of σ_x^2 is performed, then the bit allocation routine is executed, and the counter is reset.

Bit Allocation

The bit allocation routine allocates bitrates for each subband. It carries out this task through look-up table and iterative comparison of real overall average bitrate with the expected bitrate, and correction is made to the subband bit allocations if they do not match.

The bitrate for subband k, with k the possible selection of llll, lllh, llh, lh, or h , is shown bellow,

$$\begin{aligned} B_k &= \bar{B} + \frac{1}{2}\log_2 \frac{\sigma_k^2}{(\sigma_h^2)^{\frac{1}{2}}(\sigma_{lh}^2)^{\frac{1}{4}}(\sigma_{llh}^2)^{\frac{1}{8}}(\sigma_{lllh}^2)^{\frac{1}{16}}(\sigma_{llll}^2)^{\frac{1}{16}}} \\ &= \bar{B} + \frac{1}{2}\log_2 \sigma_k^2 - \frac{1}{4}\log_2 \sigma_h^2 - \frac{1}{8}\log_2 \sigma_{lh}^2 - \frac{1}{16}\log_2 \sigma_{llh}^2 \\ &\quad -\frac{1}{32}\log_2 \sigma_{lllh}^2 - \frac{1}{32}\sigma_{llll}^2 \end{aligned} \tag{7.3}$$

To save computation time, the logarithm is realized by a look-up table[8]. To form a table with a reasonable size, the variances are scaled down by 2^3, which corresponds to $2^{15-3} = 4096$ entries. Since the TMS320C25 deals only with integers, the logarithms are premultiplied by 2^8 to ensure maximum dynamic range. The expected average bitrate is also premultiplied by 2^4 to ensure bitrate resolution of $2^{-4} = 0.0625$. The modified expression of subband bitrate is then

$$\begin{aligned} B_k &= 2^{-4}(2^4\bar{B}) + \frac{1}{2}[2^8\log_2\frac{\sigma_k^2}{2^3}] \times 2^{-8} - \frac{1}{4}[2^8\log_2\frac{\sigma_h^2}{2^3}] \times 2^{-8} \\ &\quad -\frac{1}{8}[2^8\log_2\frac{\sigma_h^2}{2^3}] \times 2^{-8} - \frac{1}{16}[2^8\log_2\frac{\sigma_{llh}^2}{2^3}] \times 2^{-8} \\ &\quad -\frac{1}{32}[2^8\log_2\frac{\sigma_{lllh}^2}{2^3}] \times 2^{-8} - \frac{1}{32}[2^8\log_2\frac{\sigma_{llll}^2}{2^3}] \times 2^{-8} \\ &= 2^{-4}\{[16\bar{B}] + 2^{-5}[2^8\log_2\frac{\sigma_k^2}{2^3}] - M\} \end{aligned} \tag{7.4}$$

where

$$M = 2^{-9}\{2^3[2^8\log_2\frac{\sigma_h^2}{2^3}] + 2^2[2^8\log_2\frac{\sigma_{lh}^2}{2^3}] + 2[2^8\log_2\frac{\sigma_{llh}^2}{2^3}] + [2^8\log_2\frac{\sigma_{lllh}^2}{2^3}] + [2^8\log_2\frac{\sigma_{llll}^2}{2^3}]\}$$

Since the bitrate for each subband can only be integer, the actual average bitrate is not necessarily the desired average bitrate. Furthermore, if the variances of the five subbands differ considerably, the allocated bits for some bands may become negative. The occurrence of negative allocations would require a recalculation of the bitrates with those negative variances set to zero. If the average bitrate is still incorrect, the rates are modified according to the following scheme

1. If the bitrate is larger than the expected value, the bitrates are decremented by 1, one after another starting from h band to llll band.

2. If the bitrate is less than the expected value, the bitrates are incremented by 1, starting from llll band to h band.

The scheme is according to the fact that the energy of speech signal is mostly contained in the lower frequency part, which is represented by low subbands.

Even with the above scheme, there still exists two problems. Firstly, the routine may become a deadlock with bitrate fluctuation around the expected value without converging to it. Secondly, the weights of subbands are not the same. The average bitrate is represented by

$$B_{ave} = \frac{1}{2}B_h + \frac{1}{4}B_{lh} + \frac{1}{8}B_{llh} + \frac{1}{16}(B_{llll} + B_{lllh}) \tag{7.5}$$

which implies that a change in l band is equivalent to 8 times the change in llll band. This will result with inaccuracy of final bit allocation.

These two factors creat a dilemma for the programing: if one wants to override the deadlock by setting indicators for an increment or decrement, which prevents the decrement after increment, or the opposite way, the program may come out with overall bitrate mismatch of as high as 0.5 bit. To avoid this, a much more complicated scheme should be used. This will result in more iteration and therefore more execution time, which is not desired in the real-time operation. Fortunately, the simulation results show that a deficit of 0.5 bit rarely happens. On the other hand, even if it happens, it will last only one block, corresponding to some 0.1 sec.

The overall bitrate still converges to the expected one.

Quantizer Routine

Actual quantization is performed each time after the corresponding subband data is obtained. The quantizer routine consists of the following steps:

1. Normalization of subband data. Since the quantization and reconstruction tables are set with respect to zero mean and unit variance signal only, the data to be quantized should first be normalized to its zero mean, unity variance counterpart. For a given variance of

$$\sigma_x^2 = E[(x - \eta_x)^2]$$

it follows that

$$E[(\frac{x - \eta_x}{\sigma_x})^2] = 1$$

Therefore the normalized sample, $\tilde{x}$ is found by

$$\tilde{x} = \frac{x - \eta_x}{\sigma_x} \tag{7.6}$$

2. Quantization table look-up. The normalized sample is then compared with the thresholds in the quantization tables. An appropriate threshold level i is assigned to the sample and the corresponding reconstruction value is also given for synthesis. Therefore the transmission of index i is enough for the reconstruction. (surely the statistics and bit allocation are to be transmitted at the beginning of each block.)

3. Denormalization of the reconstruction value. This is the opposite operation of step 1.

The bitrate reduction is achieved by transmitting the index $\imath(x)$ instead of 16 bit subband signal itself.

The quantization involves the calculation of square roots and division. These operations are achieved by their opposite operations, square and multiplication. The iterations are inevitable, but made simple if we choose initial increment value as half the maximum possible value, and with increments of half the last increment value. If the trial value is too big, the increment is discarded, if the trial value is too small, the increment is added to the trial value. In both cases iteration then goes to a smaller increment value unless the increment value reaches zero. In the case of 16 bit operation, 14 iterations are required before the correct result is reached.

Chapter 8

Comparisons and Conclusions

As mentioned in Chapter 4, the energy compaction measure, G_{TC} for transform coding and G_{SBC} for subband orthogonal filter banks, is the common tool in the literature for comparison of orthogonal signal decomposition schemes. In this chapter we are going to apply this measure to compare the energy compaction performance of subband irregular tree with the industry standard, the discrete cosine transform. For simulation, AR(1) source is assumed as input.

8.1 The Autoregressive Model

For the performance evaluation of the signal processing algorithms, the introduction of signal model is an inevitable prerequisite. In the simulation, the AR(Autoregressive) process is often used as the model for real world signal. The general expression is given by

$$\frac{Y(z)}{X(z)} = \frac{1}{1 + r_1 z^{-1} + \cdots + r_p z^{-p}} \tag{8.1}$$

For the simplest case, with $p = 1$, the above expression becomes

$$\frac{Y(z)}{X(z)} = \frac{1}{1 + r z^{-1}} \tag{8.2}$$

which is called the AR(1) model. The expression can also be written as difference equation as

$$y(n) = ry(n-1) + x(n) \tag{8.3}$$

where $y(n)$ is the output at the n^{th} interval, and $x(n)$ is a white noise with zero mean and unit variance. The AR(1) source can be easily produced by recursively adding a white noise to its previous value.

One of the advantages of the autoregressive models are their predictable statistics. The autocorrelation of a discrete real process $y(n)$ is given by

$$R_y(k) = E[y(n)y(n-k)] \qquad k = 0, \pm 1, \pm 2, \cdots \tag{8.4}$$

for an AR(1) source $y(n)$, the autocorrelation $R_y(n)$ is recursively acquired by the following steps:

Firstly, $R_y(0)$ is computed by

$$R_y(0) = E\{[ry(n-1) + x(n)]^2\} = r^2 R_y(0) + R_x(0) \tag{8.5}$$

where $R_x(0)$ is the autocorrelation of the normalized white noise, and is often denoted by $R_x(0) = \sigma_x^2$.

The above equation can also be written as

$$R_y(0) = \frac{\sigma_x^2}{1 - r^2} \tag{8.6}$$

Secondly, repeating the above procedure for $k = 1$ and $k =$ 2n and so forth,

$$\begin{aligned} R_y(1) &= E\{y(n)y(n+1)\} = E\{y(n)[ry(n) + x(n+1]\} \\ &= rR_y(0) + E\{y(n)x(n+1)\} \\ R_y(2) &= rR_y(1) + E\{y(n)x(n+2)\} \\ \cdots &= \\ R_y(k) &= rR_y(k-1) + E\{y(n)x(n+k)\} \qquad k = 1, 2, \cdots \end{aligned} \tag{8.7}$$

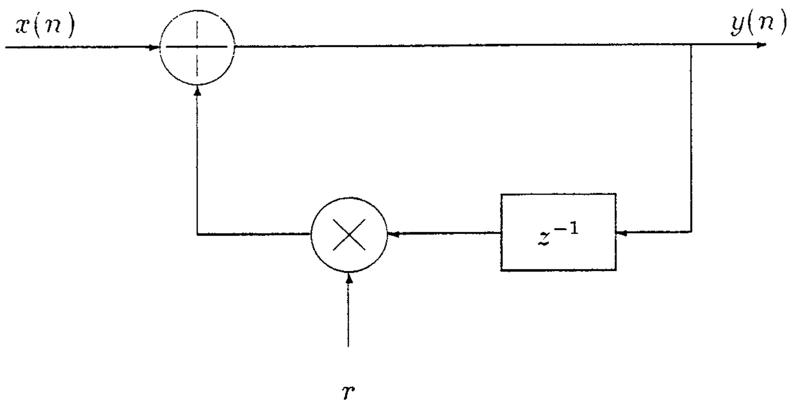


Figure 8.1: AR(1) network

since $y(n)$ depends only on $x(n), x(n-1), \cdots$, and $x(n+1), x(n+2), \cdots$ are uncorrelated with the values before and including interval n, we can draw the conclusion that

$$E\{y(n)x(n+k)\} = 0 \qquad k = 1, 2, \cdots \tag{8.8}$$

so here comes the general expression of $R_y(k)$ as

$$R_y(k) = r^k R_y(0) = r^k \frac{\sigma_x^2}{1-r^2} \tag{8.9}$$

where the stable process requires $-1 < r < 1$. Fig.(8.1) illustrates the AR(1) network. Figs.(8.2) shows the autocorrelation function and power spectral density function for $r = 0.6$.

The correlation coefficient r is derived by

$$r = \frac{R_y(0)}{R_y(1)} \tag{8.10}$$

The transfer function of an AR(1) network is derived from Eq.(8.2) as

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{1 - re^{j\omega T_s}} \tag{8.11}$$

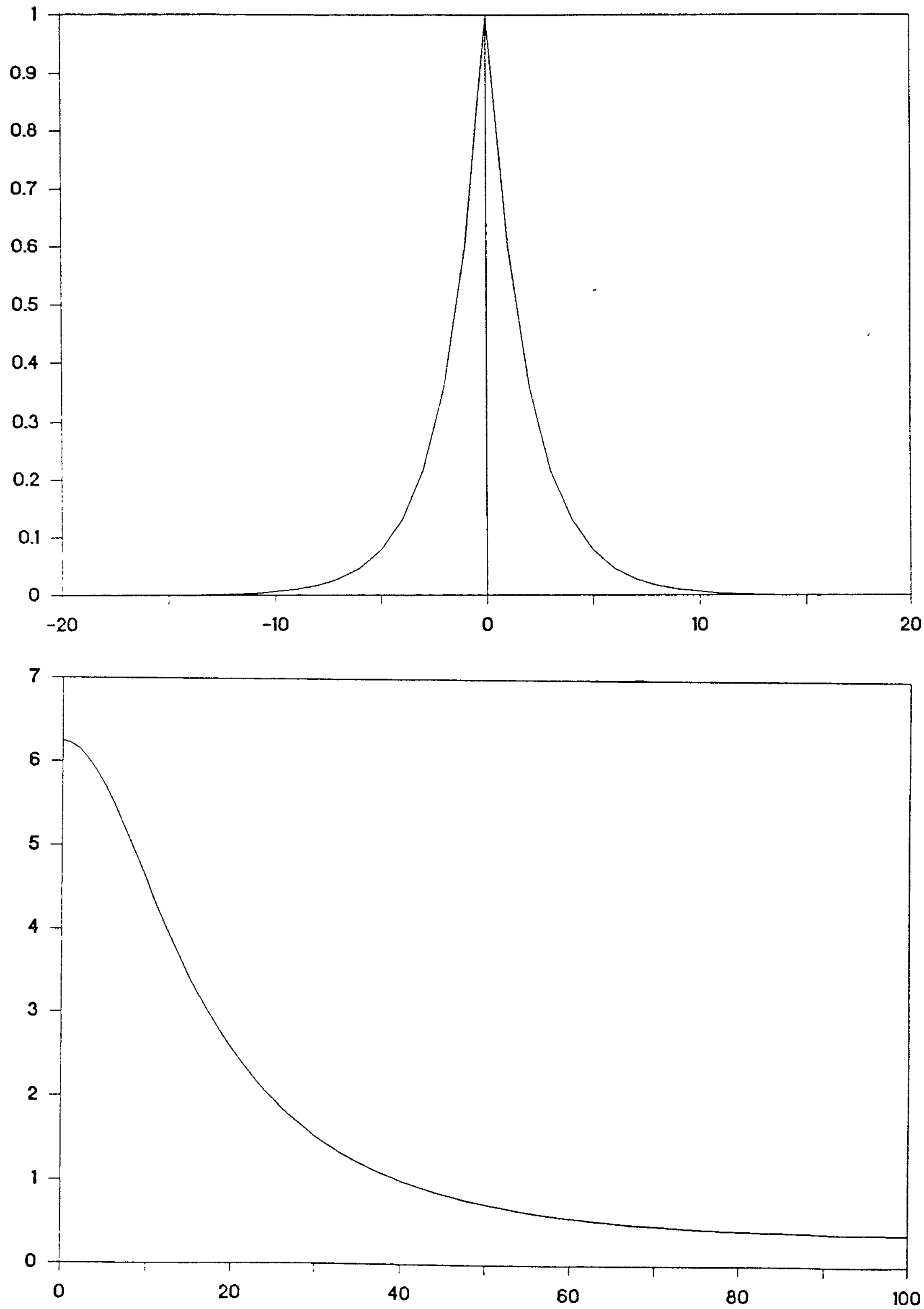


Figure 8.2: Autocorrelation and power spectral density function of AR(1) model for $r = 0.6$

Assuming a constant input power spectral density σ_x^2 the output power spectral density is

$$S_y(\omega) = |H(\omega)|^2\sigma_x^2 = \frac{\sigma_x^2}{(1+r^2) - 2rcos(\omega T_s)} \tag{8.12}$$

It can be seen from Eq.(8.12) that with larger value of r, the frequency energy distribution tend to gather into lower frequencies. In other words, highly-correlated signals have their energy concentrated more in the lower frequencies. For the simulation of speech, the AR(1) model with correlation coefficients of $r = 0.6$ through 0.9 are good approximations.

8.2 The Discrete Cosine Transform

Since we are going to measure the performance of the speech codec in comparison to discret cosine transform (DCT) in terms of energy compaction gain, the basics of DCT are briefly stated here.

The DCT is a suboptimum transform for AR(1) sources with a transform matrix independent to the input. The kernel of DCT is given by

$$C_r(k) = \sqrt{\frac{2}{N+1}\alpha(r)cos\frac{(2k+1)r\pi}{2(N+1)}}, \qquad \begin{array}{rcl} r &=& 0,1,\cdots,N \\ k &=& 0,1,\cdots,N \end{array} \tag{8.13}$$

where $\alpha(0) = 1/\sqrt{2}$ and $\alpha(k) = 1, r \neq 0$. The transform coefficients are then derived as

$$\theta_r = \sqrt{\frac{2}{N+1}}\alpha(r)\sum_{k=0}^{N} cos\frac{(2k+1)r\pi}{2(N+1)}, \qquad r = 0,1,\cdots,N \tag{8.14}$$

The inverse transform is

$$x(n) = \sqrt{\frac{2}{N+1}}\sum_{r=0}^{N}\alpha(r)cos\frac{(2n+1)r\pi}{2(N+1)} \tag{8.15}$$

8.3 Performance Evaluation

The performance of the speech codec is evaluated in three different ways with different emphases. Firstly, it is measured by energy compaction gain over pulse code modulation(PCM) with the scheme given in Chapter 4. The energy compaction gains of discret cosine transform and regular-tree subband coding are employed for comparison. Secondly, the performance of the codec is evaluated by signal-to-noise ratio at each section (SNRSEC). Finally, the subjective appraisal is given which shows that the codec output is intelligible even at the rate of 0.25 bit per sample.

8.3.1 Energy Compaction Performance

The signal decomposition schemes, namely subband and DCT are compared here in several ways.

The AR(1) input signal is employed first for the comparison. The AR(1) sources of correlation coefficients between 0.65 and 0.9 generally cover the speech model. However, the AR(1) model of speech signals "one", which is with $\rho = 0.837108$, and "child", with $\rho = 0.711487$ are used as input since these two signals are used as real input later in experiments.

Tables (8.1) and (8.2) gives energy compaction results of several different subband tree structures and 16 band DCT for the AR(1) models of "one" and "child", respectively. These tables assume the full tree size of 16 bands. We can observe from the tables that 8-tap Binomial-QMF generally provides higher energy compaction performance than 16 band DCT. Even with 5 bands the gain of 8-tap filter is almost the same as 16×16 DCT. For 6-tap filter, 16 band energy compaction gains are also higher than 16 band DCT for both the AR(1) model of "one" and "child".

Tables (8.3) and (8.4) display the energy compaction results of 4-tap, 6-tap, 8-tap binomial QMF based filter banks compared with 16 and 8 band DCT for the speech signals "one" and "child", respectively. From these tables we can see that although the Binomial QMF based filter banks do not perform much better than DCT as they did in AR(1) models, the 8-tap filter banks still show better

	4-tap	6-tap	8-tap
16 bands	2.98097	3.13428	3.19984
6 bands	2.89901	3.03301	3.09326
5 bands	2.87836	3.00843	3.06792
DCT (16×16)	3.06848		

Table 8.1: Energy compaction performance of several 4-, 6-, 8-tap Binomial-QMFs and 16 band DCT for an AR(1) source corresponding to speech signal "one"

	4-tap	6-tap	8-tap
16 bands	1.87092	1.93696	1.96597
6 bands	1.82320	1.88088	1.90771
5 bands	1.81472	1.87145	1.89811
DCT(16×16)	1.91830		

Table 8.2: Energy compaction performance of several 4-, 6-, 8-tap Binomial-QMFs and 16 band DCT for an AR(1) source corresponding to speech signal "child"

compaction performance. For comparison, the 8-tap filter also outperform 8 band DCT in full bands.

For further comparison, Tables (8.5) through (8.8) are presented which list the compaction of full tree 4-, 6-, 8-tap filters and the 8 and 16 band DCT for the signals "one","child" and their AR(1) model approximations.

The energy compaction gain gives the coding gain of the system, but it does not evaluate the performance of the quantization. To further measure the performance

	4-tap	6-tap	8-tap
16 bands	4.777	6.932	8.659
6 bands	3.386	4.536	5.103
5 bands	3.305	4.458	5.013
DCT (16×16)	7.558		
DCT (8×8)	5.688		

Table 8.3: Energy compaction performance of several 4-, 6- 8-tap Binomial-QMFs and 16 and 8 band DCT for speech signal "one"

	4-tap	6-tap	8-tap
16 bands	2.088	2.329	2.437
6 bands	1.734	1.916	2.026
5 bands	1.700	1.894	2.006
DCT (16×16)	2.220		
DCT (8×8)	1.946		

Table 8.4: Energy compaction performance of several 4-, 6-, 8-tap Binomial-QMFs and 16 and 8 band DCT for speech signal "child"

band	2	4	8	16
4-tap	2.82059	3.19890	4.28881	4.77715
6-tap	3.62849	4.41595	6.01510	6.92895
8-tap	4.19690	5.55909	7.62958	8.65944
DCT			5.68774	7.558

Table 8.5: Energy compaction of full tree SBC vs. DCT for speech signal "one".

band	2	4	8	16
4-tap	1.60028	1.75313	1.89214	2.08785
6-tap	1.69848	1.94545	2.12778	2.32911
8-tap	1.75835	2.04507	2.26209	2.43703
DCT			1.94592	2.22

Table 8.6: Energy compaction of full tree SBC vs. DCT for speech signal "child"

band	2	4	8	16
4-tap	2.01683	2.66122	2.90712	2.98097
6-tap	2.07373	2.77675	3.05257	3.13428
8-tap	2.09996	2.82716	3.11512	3.19984
DCT				3.06848

Table 8.7: Energy compaction of full tree SBC vs. DCT for AR(1) source corresponding to signal "one"

band	2	4	8	16
4-tap	1.52611	1.77298	1.84954	1.87092
6-tap	1.56053	1.82953	1.91384	1.93696
8-tap	1.57702	1.85497	1.94220	1.96597
DCT				1.91830

Table 8.8: Energy compaction of full tree SBC vs. DCT for AR(1) source corresponding to signal "child"

of the implemented speech codec, it is necessary to employ signal-to-noise evaluation on the system. The subjective performance measure is also used to determine the minimum bitrate needed to give an intelligible output.

8.3.2 SNRSEG Evaluation and Subjective Measure

The signal-to-noise ratio is applied to measure the overall performance of the speech codec. It should be pointed out that since speech signal has a nonstationary nature, it is worthwhile to apply the SNR measure to each segment of the signal. This measure is known as SNRSEG. Figs.(8.3) and (8.4) show the codec performance in terms of SNRSEG at bitrates of 0.5 and 2 bits/sample. The segment length is 128 samples, which is in accordance with a glottal period of human voice at a sampling rate of 8000Hz.

The output of the real-time speech codec is judged by human ear to give the subjective performance measure. For evaluation, The codec is set to operate with AM radio input at different bitrates ranging from 0.25 to 4 bits per sample. The results of the evaluation are given below.

- The quality of the codec output is very good at a bitrate of 1.5 bits per sample and up. The transmission gives perfect intelligibility of a speaker.

- With bitrate between 0.5 and 1 bit per sample, the output of the codec becomes noisy but it is still possible to understand every word.

- Bitrates less than 0.5 result in a considerable noise, but with concentration one is still able to understand messages at bitrate of 0.25 bit per sample.

8.4 Conclusions

A real-time subband speech codec is implemented on TI's TMS320C25 digital signal processor chip. The codec takes a 5-band irregular tree structure according to the power spectral density function of speech signals.

The performance of the codec is compared with DCT and it shows that the energy compaction gain of the codec is comparable to that of 8×8 DCT, although the former one is easier to implement due to less bands.

The subjective evaluation obtains that the output of the codec is intelligible at bitrate as low as 0.25 to 0.5 bit per sample. This reduces the transfer rate to 2 to 4 kbps at the sampling rate of 8kHz comparing to 96kbps original rate.

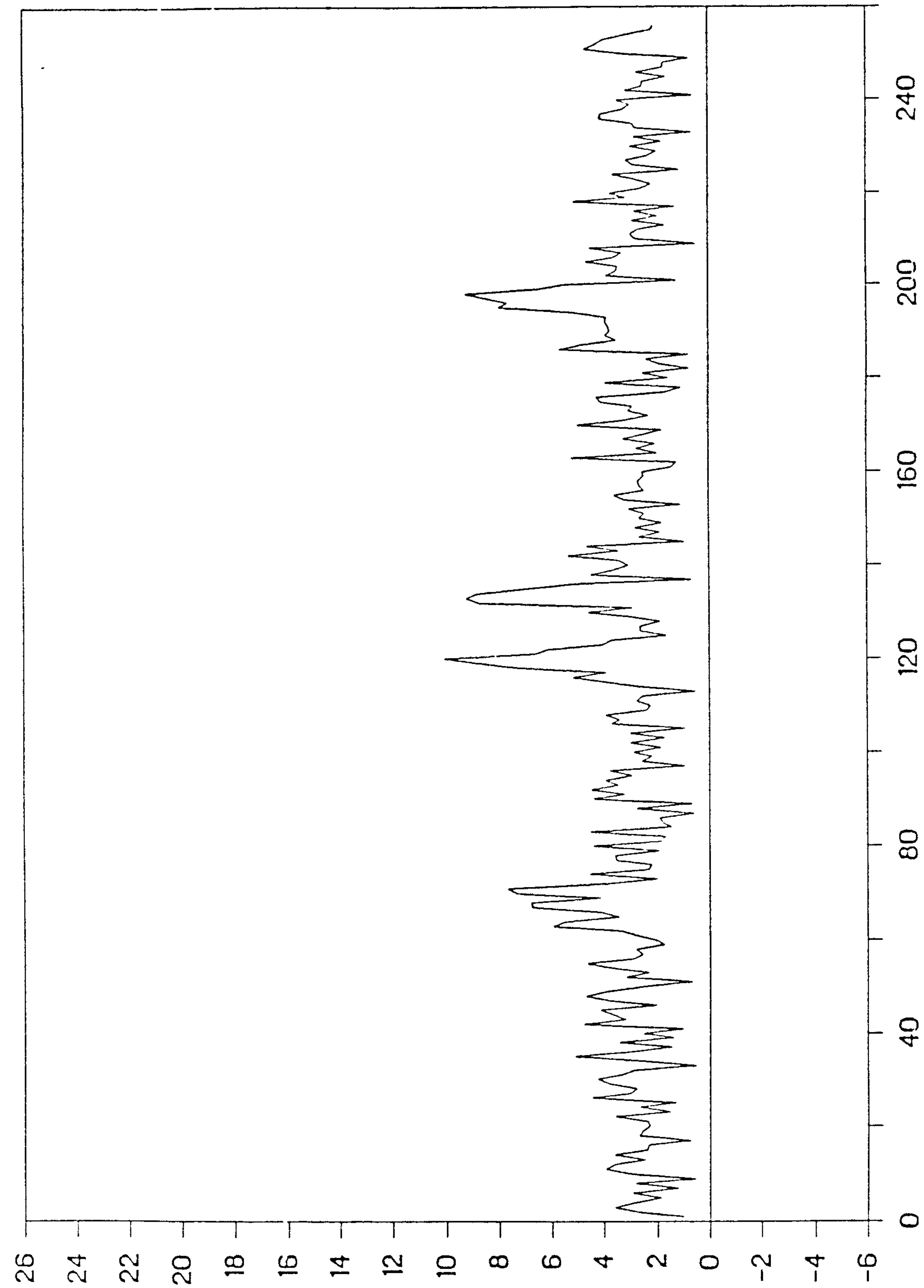


Figure 8.3: SNRSEG of the 5-band speech codec at 0.5 bit per sample

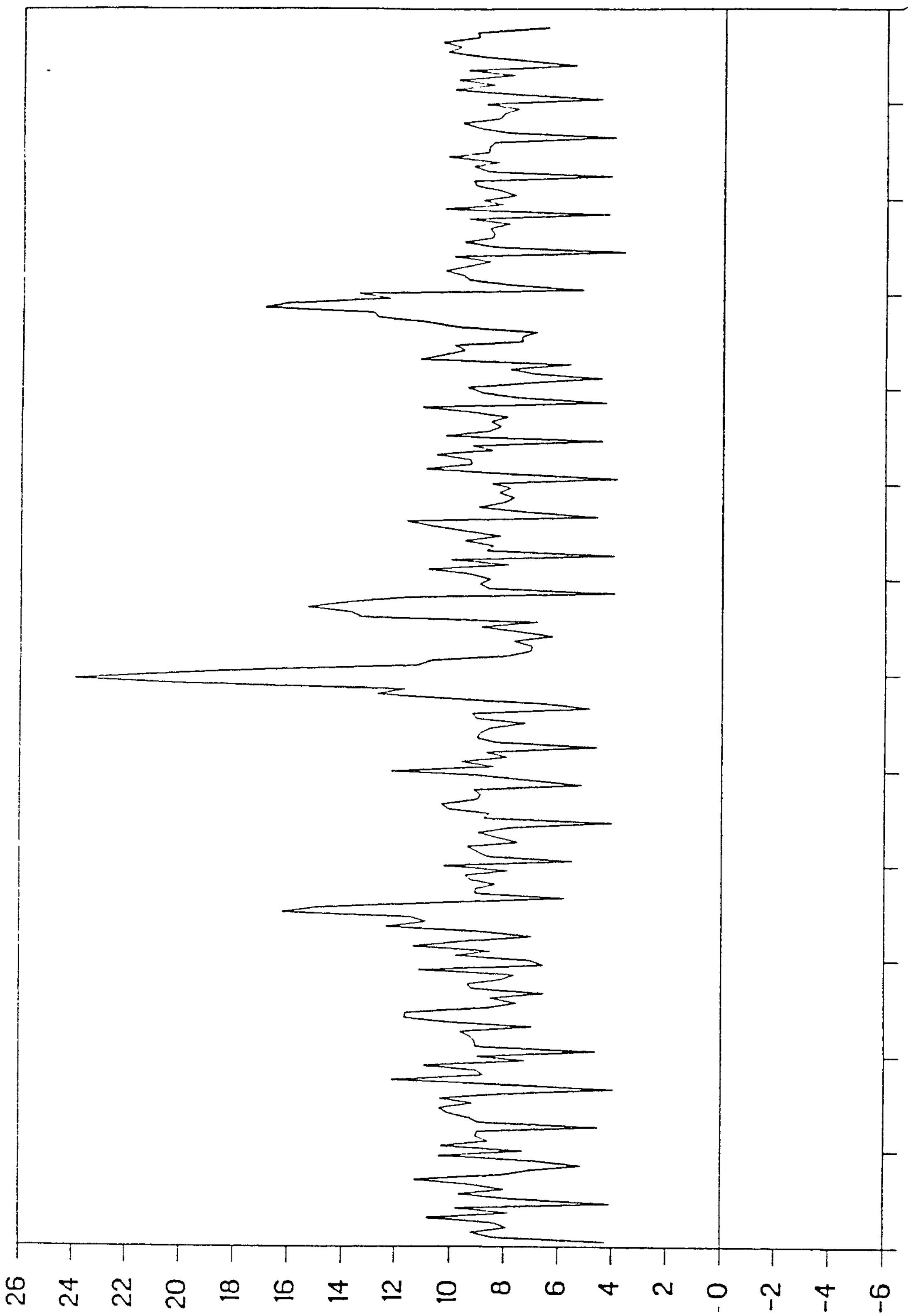


Figure 8.4: SNRSEG of 5-band speech codec at 2 bits per sample

Appendix A

Source Code of the Speech Codec

```
.GLOBAL  LOGTABLE1
.GLOBAL START_1P
.GLOBAL LENGTH_1P
.GLOBAL RECON_1P
.GLOBAL START_2P
.GLOBAL LENGTH_2P
.GLOBAL RECON_2P
.GLOBAL START_3P
.GLOBAL LENGTH_3P
.GLOBAL RECON_3P
.GLOBAL START_4P
.GLOBAL LENGTH_4P
.GLOBAL RECON_4P
.GLOBAL START_5P
.GLOBAL LENGTH_5P
.GLOBAL RECON_5P
.GLOBAL START_6P
.GLOBAL LENGTH_6P
.GLOBAL RECON_6P
.GLOBAL START_7P
.GLOBAL LENGTH_7P
.GLOBAL RECON_7P
*-----------------------------------------
* CONSTANTS
*-----------------------------------------
AVGBIT .SET 5*4 ; AVG BITRATE, PREMULTIPLIED BY 4(2^2)
RATE .SET 1
LENGTH1 .SET 65+21 ; BUFFER LENGTH OF LEVEL 1
LENGTH2 .SET 33+14 ; BUFFER LENGTH OF LEVEL 2
LENGTH3 .SET 17+7 ; BUFFER LENGTH OF LEVEL 3
LENGTH4 .SET 9 ; BUFFER LENGTH OF LEVEL 4
MAXBIT .SET 5 ; MAXIMUM QUANTIZER RESOLUTION BY BIT
INPUTPORT .SET 2
OUTPUTPORT .SET 2
CNTRLPORT .SET 1
;-----------------------------------
; VARIABLES
;-----------------------------------
.BSS A1,1 ; EVEN/ODD INDICATORS FOR LEVEL 1
```

```
.BSS A2,1 ; EVEN/ODD INDICATORS FOR LEVEL 2
.BSS A3,1 ; EVEN/ODD INDICATORS FOR LEVEL 3
.BSS A4,1 ; EVEN/ODD INDICATORS FOR LEVEL 4
.BSS SAMPLE,9 ; NEW INPUT VALUE
.BSS OUTVALUE,1 ; OUTPUT VALUE
.BSS OUT1,1 ; L-BAND SYNTHSIZE OUTPUT
.BSS OUT2,1 ; H-BAND SYNTHESIZE OUTPUT
.BSS TEMPH,1 ; BUFFER
.BSS TEMPL,1 ; BUFFER
.BSS TEMP,1 ; BUFFER
.BSS BUFFER,1 ; BUFFER
.BSS TEMPS,1 ; BUFFER
.BSS  WH,1 ; WRITE INDICATOR FOR H-BAND
.BSS  RH,1 ; READ INDICATOR FOR H-BAND
.BSS  WLH,1 ; WRITE INDICATOR FOR LH-BAND
.BSS  RLH,1 ; READ INDICATOR FOR LH-BAND
.BSS  WLLH,1 ; WRITE INDICATOR FOR LLH-BAND
.BSS  RLLH,1 ; READ INDICATOR FOR LLH-BAND
.BSS  WLLLH,1 ; WRITE INDICATOR FOR LLLH-BAND
.BSS  RLLLH,1 ; READ INDICATOR FOR LLLH-BAND
.BSS  WLLLL,1 ; WRITE INDICATOR FOR LLLL-BAND
.BSS  RLLLL,1 ; READ INDICATOR FOR LLLL-BAND
.BSS LENGTH,1 ; FOR INDICATOR ADJUSTMENT USE
.BSS ADDR,1 ; FOR INDICATOR ADJUSTMENT USE
*******************************************************
******  DEFINITION FOR QUANTIZATION PART ***************
*******************************************************
; FOR BIT RE-ALLOCATION USE
.BSS INC_INDI,1
.BSS INC_LLLL,1
.BSS INC_LLLH,1
.BSS INC_LLH,1
.BSS INC_LH,1
.BSS DEC_INDI,1
.BSS DEC_LLLH,1
.BSS DEC_LLH,1
.BSS DEC_LH,1
.BSS DEC_H,1
.BSS LOOPCOUNTER,1 ; COUNTS BLOCK LENGTH
; FOR BIT ALLOCATION USE
.BSS BASE,1
.BSS LOG_H,1
.BSS LOG_LH,1
.BSS LOG_LLH,1
.BSS LOG_LLLH,1
.BSS LOG_LLLL,1
.BSS SUMH,2
.BSS SUMH2,2
.BSS SUMLH,2
.BSS SUMLH2,2
.BSS SUMLLH,2
.BSS SUMLLH2,2
.BSS SUMLLLH,2
.BSS SUMLLLH2,2
```

```
.BSS SUMLLLL,2
.BSS SUMLLLL2,2
.BSS QRH,1 ; H-BAND POINTER FOR QUANTIZATION DATA
.BSS QRLH,1 ; LH-BAND POINTER FOR QUANTIZATION DATA
.BSS QRLLH,1
.BSS QRLLLH,1
.BSS QRLLLL,1
.BSS BITH,1 ; BITRATE FOR H-BAND
.BSS BITLH,1 ; BITRATE FOR LH-BAND
.BSS BITLLH,1
.BSS BITLLLH,1
.BSS BITLLLL,1
.BSS MH,1 ; MEAN OF H-BAND
.BSS MLH,1 ; MEAN OF L-BAND
.BSS MLLH,1
.BSS MLLLH,1
.BSS MLLLL,1
.BSS SIGMA_H,1 ; VARIANCE OF H-BAND
.BSS SIGMA_LH,1 ; VARIANCE OF LH-BAND
.BSS SIGMA_LLH,1
.BSS SIGMA_LLLH,1
.BSS SIGMA_LLLL,1
.BSS SIGNAREA,1 ; POSITIVE/NEGATIVE INDICATOR USED IN QUAN
LOGTABLE .USECT "LOGAREA",0800h ; LOG TABLE STARTING ADDRESS
; FOR QUANTIZATION USE
START_1 .USECT "ABC",1
LENGTH_1 .USECT "ABC",1
RECON_1 .USECT "ABC",1
START_2 .USECT "ABC",2
LENGTH_2 .USECT "ABC",1
RECON_2 .USECT "ABC",2
START_3 .USECT "ABC",4
LENGTH_3 .USECT "ABC",1
RECON_3  .USECT "ABC",4
START_4 .USECT "ABC",8
LENGTH_4 .USECT "ABC",1
RECON_4 .USECT "ABC",8
START_5 .USECT "ABC",16
LENGTH_5 .USECT "ABC",1
RECON_5 .USECT "ABC",16
START_6 .USECT "ABC",32
LENGTH_6 .USECT "ABC",1
RECON_6  .USECT "ABC",32
START_7 .USECT "ABC",64
LENGTH_7 .USECT "ABC",1
RECON_7 .USECT "ABC",64
*********************************************************
*******  THE FOLLOWING BUFFERS ARE IN PAGE 7  ***********
*********************************************************
; FOR QUANTIZATION USE
H_START .USECT "PAGE7",1
H_LENGTH .USECT "PAGE7",1
RECONH .USECT "PAGE7",1
LH_START .USECT "PAGE7",1
```

```
LH_LENGTH .USECT "PAGE7",1
RECONLH .USECT "PAGE7",1
LLH_START .USECT "PAGE7",1
LLH_LENGTH .USECT "PAGE7",1
RECONLLH .USECT "PAGE7",1
LLLH_START .USECT "PAGE7",1
LLLH_LENGTH .USECT "PAGE7",1
RECONLLLH .USECT "PAGE7",1
LLLL_START .USECT "PAGE7",1
LLLL_LENGTH .USECT "PAGE7",1
RECONLLLL .USECT "PAGE7",1
TH .USECT "PAGE7",1
TLH .USECT "PAGE7",1
TLLH .USECT "PAGE7",1
TLLLH .USECT "PAGE7",1
TLLLL .USECT "PAGE7",1
SAMPLE1 .USECT "PAGE7",1
; FOR SQUARE ROOT CALCULATION
RT_TEMP .USECT "PAGE7",1
RT_TEMPH .USECT "PAGE7",1
RT_OUT .USECT "PAGE7",1
RT_TRY .USECT "PAGE7",1
RT_MED .USECT "PAGE7",1
; INTERMEDIATE BUFFER FOR SUBBANDS
ADDRH .USECT "PAGE7",65+21
ADDRLH .USECT "PAGE7",33+14
ADDRLLH .USECT "PAGE7",17+7
ADDRLLLH .USECT "PAGE7",9
ADDRLLLL .USECT "PAGE7",9

TABLE_START .USECT "TABLEB",21

*********************************************************
**** THE FOLLOWING BUFFERS ARE INDIRECTELY ADDRESSED ****
*********************************************************
; FOR CONVOLUTION USE
AL .SET 0200h
AH .SET AL+9
SL .SET AH+9
SH .SET SL+9
ALL .SET SH+9
ALH .SET ALL+9
SLL .SET ALH+9
SLH .SET SLL+9
ALLL .SET SLH+9
ALLH .SET ALLL+9
SLLL .SET ALLH+9
SLLH .SET SLLL+9
ALLLL .SET SLLH+9
ALLLH .SET ALLLL+9
SLLLL .SET ALLLH+9
SLLLH .SET SLLLL+9

.DATA
```

```
; FILTER COEFFICIENTS
; PREMULTIPLIED BY 2^15
ANALOW .WORD 7549,23424,20673,-917,-6129,1011,1078,-347
ANAHIGH .WORD 347,1078,-1011,-6129,917,20673,-23424,7549
SYNLOW .WORD -347,1078,1011,-6129,-917,20673,23424,7549
SYNHIGH .WORD 7549,-23424,20673,917,-6129,-1011,1078,347

.TEXT
B START
.SPACE 254*16 ; PROGRAMS START AT 100HEX
*-----------------------------------
* BEGINING OF SOURCE CODE
*-----------------------------------
**** INITIALIZATION ****
START CNFD ; CONFIGURE ON-CHIP BLOCK 0 AS DATA MEM
SSXM ; SET SIGN-EXTENSION MODE
SOVM ; SET OVERFLOW MODE
SPM 0 ; SET P-REG SHIFT TO 0
LDPK 6 ; LOAD DATA PAGE COUNTER TO 6TH PAGE
CALL INIT,1 ; INITIALIZE RAM
**** ANALOG BOARD SETUP ****
SETRATE
LALK 0FD8BH,0 ; INITIALIZE AIB
SACL OUTVALUE,0 ; STORE FOR OUTPUT
OUT OUTVALUE,3 ; SETUP SAMPLING RATE
**** INPUT A SAMPLE FROM A/D CONVERTER ****
WAIT BIOZ LOOP ; SYNCHRONIZATION
B WAIT ; WAIT UNTIL AIB INTERUPT
LOOP IN SAMPLE,INPUTPORT ; INPUT A SAMPLE
LAC SAMPLE,0 ; LOAD ACC WITH SAMPLE VALUE
B ANALYZE ; BRANCH TO TREE STRUCTURING
**** OUTPUT A SAMPLE TO D/A CONVERTER ****
OUTPUT SACL OUTVALUE,0 ; STORE VALUE FOR OUTPUT
LRLK AR1,OUTVALUE ; AR1 POINTS TO OUTPUT VALUE
OUT *,OUTPUTPORT ; OUTPUT RECONSTRUCTED SAMPLE
B WAIT ; TO GET A NEW SAMPLE
**** ANALYZE INPUT SIGNAL ****
ANALYZE
LEVEL_1 LARP AR1 ; AR1 ACTIVE
LRLK AR1,SAMPLE+7 ; ASSUME AR1 ACTIVE
ZAC ; ZERO ACC FOR CONVOLUTION
MPYK 0 ; ZERO P-REG
RPTK 7 ; REPEAT FOR 8 TIMES
MAC ANALOW,*- ; CONVOLUTION
APAC ; ADD THE LAST PRODUCT
LRLK AR1,AL ; POINT TO L-BAND BUFFER
SACH *,1 ; STORE
LRLK AR1,SAMPLE+7 ; H-BAND ANALYSIS, SIMILAR TO L-BAND
ZAC
MPYK 0
RPTK 7
MACD ANAHIGH,*-
APAC
; CHECK EVEN/ODD INDICATOR TO DECIDE
```

```
; WHETHER TO STORE OR DISCARD CURRENT
; FILTER OUTPUT
LRLK AR1,AH
SACH *,1
SACH TEMPH,1
LARP AR5
LAR AR5,A1
BANZ STORE1,*,AR4 ; STORE IF EVEN/ODD INDICATOR NOT ZERO
B SYNTH_4 ; DISCARD IF E/O INDICATOR ZERO
STORE1
LAR AR4,WH ; ACC CONTAINS H-BAND VALUE
SACH *+,1 ; STORE VALUE FOR QUANTIZATION
LALK LENGTH1
SACL LENGTH ; STORE BUFFER LENGTH FOR POINTER ADJUST
LALK ADDRH ; LOAD BUFFER STARTING ADDR
; FOR POINTER ADJUSTMENT
CALL ADJUST4,AR4 ; ADJUST POINTER
SAR AR4,WH ; STORE ADJUSTED POINTER
QUAN_1 LAR AR4,RH ; STATISTICS OF H-BAND GOES HERE
LAC *+,AR1 ; LOAD A SAMPLE FROM H-BAND
LRLK AR1,SH ; AR1 POINTS TO SYNTHESIS BUFFER
SACL *,AR4 ; GET AN H-BAND SAMPLE FOR STATISTICS
LALK ADDRH
CALL ADJUST4,AR4 ; ADJUST POINTER
SAR AR4,RH ; STORE NEW POINTER VALUE
; ====================================================
; ----------------------------------------
; MEAN AND VARIANCE CALCULATION , HIGH BAND LEVEL 4
; MEAN CALCULATION
ZALH SUMH ;
ADDS SUMH+1 ; LOAD ACC WITH 32-BIT SUMATION VALUE
ADD TEMPH ; ADD TO CURRENT CURRENT SAMPLE
SACH SUMH ; STORE BACK
SACL SUMH+1 ;
; VARIANCE CALCULATION
LRLK AR1,SUMH2 ;
LARP AR1 ;
SQRA TEMPH ; LOAD 32-BIT SUMATION OF SAMPLE^2
ZALH *+ ; AND ADD TO CURRENT SAMPLE^2
ADDS *-
APAC
SACH *+
SACL *,AR4 ; STORE BACK
; ---------------------------------------
; THE END OF STAT CALCULATION OF LEVEL 1 HIGH BAND
; START OF QUANTIZE LEVEL 1 HIGH BAND
LAR AR4,QRH ; LOAD QUANTIZATION VALUE POINTER
LAC *+ ; LOAD VALUE TO BE QUANTIZED
SACL BUFFER ;
LALK ADDRH ;
CALL ADJUST4 ; ADJUST POINTER
SAR AR4,QRH ; STORE ADJUSTED POINTER
LARP AR0 ; AR0 ACTIVE
LAR AR0,BITH ; LOAD AR0 CURRENT BIT
```

```
BANZ QUAN4,* ; IF BITRATE IS NOT ZERO, QUANTIZE
PUT04 ZAC ; BITRATE ZERO, NO QUANTIZATION
B TACKLE41,*,AR1
QUAN4 LAC BUFFER ; LOAD  SAMPLE TO BE QUANTIZED
SUB MH ; NORMALIZATION
BGEZ T4 ; IF SAMPLE GREATER THAN ZERO, SIGN IS 0
LARK AR7,1 ; IF NEGOTIVE, SET SIGN TO 1
SAR AR7,SIGNAREA
NEG ; INVERT NEGATIVE VALUE FOR QUANTIZATION
T4 LDPK 7 ; QUANTIZATION OPERATES ON PAGE 7
LRLK AR7,SIGMA_H ; AR7 POINTS TO H-BAND VARIANCE
SACL SAMPLE1 ;
LARP AR7
LAR AR6,H_START ; LOAD LOOKUP TABLE BEGIN ADDRESS
LAR AR5,H_LENGTH ; LOAD LEVEL LENGTH
LT *,AR6
LOOP4 LAC SAMPLE1,11 ; LOAD NORMALIZED SAMPLE
MPY *+,AR5
SPAC ; COMPARE WITH THRESHOLD
BLEZ FOUND4 ; FOUND
BANZ LOOP4,*-,AR6 ; TRY NEXT LEVEL UNTIL THE END REACHED
MAR *+ ; ADJUST THRESHOLD INDEX
FOUND4 LARP AR6 ;
LAR AR0,H_START ; FOUND
MAR *0- ;
MAR *- ;
MAR *- ; AR6 NOW CONTAINS THE THRESHOLD INDEX
SAR AR6,TH ; STORE LEVEL NUMBER TO TH
RECONSTRUCTION4
LAR AR6,TH ; RECEIVER SIDE
LAR AR0,RECONH ; LOAD BASE ADDRESS OF
; RECONSTRUCTION TABLE
MAR *0+ ; GET RECONSTRUCTION ENTRY
LAC *,AR1 ; GET RECONSTRUCTION VALUE
LDPK 6 ; BACK TO PAGE 6
LAR AR1,SIGNAREA ; CHECK IF THE QUANTIZER INPUT IS NEG
MAR *-
BANZ TACKLE4,* ; POSITIVE, NO INVERT
NEG ; NEGATIVE, INVERT VALUE
TACKLE4 SACL TEMP ;
LT TEMP ;
MPY SIGMA_H ; DENOMALIZATION
PAC
SACH TEMP,5 ; SCALE
LAC TEMP
ADD MH ; GET DENORM'D VALUE
TACKLE41
LRLK AR1,SH
SACL * ; STORE FOR SYNTHESIS
ZAC
SACL SIGNAREA ; RESET SIGN INDICATOR
LEVEL_2 LARP AR1 ; SIMILAR TO LEVEL 1
LRLK AR1,AL+7
ZAC
```

```
MPYK 0
RPTK 7
MAC ANALOW,*-
APAC
LRLK AR1,ALL
SACH *,1
LRLK AR1,AL+7 ; LH-BAND
ZAC
MPYK 0
RPTK 7
MACD ANAHIGH,*-
APAC
LRLK AR1,ALH
SACH *,1
SACH TEMPH,1
LARP AR5
LAR AR5,A2
BANZ STORE2,*,AR4
B SYNTH_3
STORE2 LAR AR4,WLH
SACH *+,1
LALK LENGTH2
SACL LENGTH
LALK ADDRLH
CALL ADJUST4,AR4
SAR AR4,WLH
QUAN_2 LAR AR4,RLH
LAC *+,AR1
LRLK AR1,SLH
SACL *,AR4
LALK ADDRLH
CALL ADJUST4,AR4
SAR AR4,RLH
; ============================================================
; ----------------------------------------
; MEAN AND VARIANCE CALCULATION , HIGH BAND LEVEL
ZALH SUMLH
ADDS SUMLH+1
ADD TEMPH
SACH SUMLH
SACL SUMLH+1
LRLK AR1,SUMLH2
LARP AR1
SQRA TEMPH
ZALH *+
ADDS *-
APAC
SACH *+
SACL *,AR4
; ----------------------------------------
; THE END OF STAT CALCULATION OF LEVEL 2 HIGH BAND
; START OF QUANTIZE LEVEL 2 HIGH BAND
LAR AR4,QRLH
LAC *+
```

```
SACL BUFFER
LAR AR4,QRLH
MAR *+
LALK ADDRLH
CALL ADJUST4
SAR AR4,QRLH
LARP AR0
LAR AR0,BITLH
BANZ QUAN3,*
PUT03 ZAC
B TACKLE31,*,AR1
QUAN3 LAC BUFFER
SUB MLH
BGEZ T3
LARK AR7,1
SAR AR7,SIGNAREA
NEG
T3 LDPK 7
LRLK AR7,SIGMA_LH
SACL SAMPLE1
LARP AR7
LAR AR6,LH_START
LAR AR5,LH_LENGTH
LT *,AR6
LOOP3 LAC SAMPLE1,1
MPY *+,AR5
SPAC
BLEZ FOUND3
BANZ LOOP3,*-,AR6
MAR *+
FOUND3 LARP AR6
LAR AR0,LH_START
MAR *0-
MAR *-
MAR *-
SAR AR6,TLH
RECONSTRUCTION3
LAR AR6,TLH
LAR AR0,RECONLH
MAR *0+
LAC *,AR1
LDPK 6
LAR AR1,SIGNAREA
MAR *-
BANZ TACKLE3,*
NEG
TACKLE3 SACL TEMP
LT TEMP
MPY SIGMA_LH
PAC
SACH TEMP,5
LAC TEMP
ADD MLH
TACKLE31
```

```
LRLK AR1,SLH
SACL *
ZAC
SACL SIGNAREA
LEVEL_3 LARP AR1 ; SIMILAR TO LEVEL 1
LRLK AR1,ALL+7
ZAC
MPYK 0
RPTK 7
MAC ANALOW,*-
APAC
LRLK AR1,ALLL
SACH *,1
LRLK AR1,ALL+7
ZAC
MPYK 0
RPTK 7
MACD ANAHIGH,*-
APAC
LRLK AR1,ALLH
SACH *,1
SACH TEMPH,1
LARP AR5
LAR AR5,A3
BANZ STORE3,*,AR4
B SYNTH_2
STORE3 LAR AR4,WLLH
SACH *+,1
LALK LENGTH3
SACL LENGTH
LALK ADDRLLH
CALL ADJUST4,AR4
SAR AR4,WLLH
QUAN_3 LAR AR4,RLLH
LAC *+,AR1
LRLK AR1,SLLH
SACL *,AR4
LALK ADDRLLH
CALL ADJUST4,AR4
SAR AR4,RLLH
; ==========================================================
; ---------------------------------------
; MEAN AND VARIANCE CALCULATION , HIGH BAND LEVEL 3
ZALH SUMLLH
ADDS SUMLLH+1
ADD TEMPH
SACH SUMLLH
SACL SUMLLH+1
LRLK AR1,SUMLLH2
LARP AR1
SQRA TEMPH
ZALH *+
ADDS *-
APAC
```

```
SACH *+
SACL *,AR4
; ---------------------------------------
; THE END OF STAT CALCULATION OF LEVEL 3 HIGH BAND
; START OF QUANTIZE LEVEL 3 HIGH BAND
LAR AR4,QRLLH
LAC *+
SACL BUFFER
SAR AR4,TEMP
LAC TEMP
SBLK ADDRLLH
BGEZ RT2
CALL START
RT2 SUB LENGTH
BLEZ NO2
LRLK AR4,ADDRLLH
NO2 SAR AR4,QRLLH
LARP AR0
LAR AR0,BITLLH
BANZ QUAN2,*
PUT02 ZAC
B TACKLE21,*,AR1
QUAN2 LAC BUFFER
SUB MLLH
BGEZ T2
LARK AR7,1
SAR AR7,SIGNAREA
NEG
T2 LDPK 7
LRLK AR7,SIGMA_LLH
SACL SAMPLE1
LARP AR7
LAR AR6,LLH_START
LAR AR5,LLH_LENGTH
LT *,AR6
LOOP2 LAC SAMPLE1,11
MPY *+,AR5
SPAC
BLEZ FOUND2
BANZ LOOP2,*-,AR6
MAR *+
FOUND2 LARP AR6
LAR AR0,LLH_START
MAR *0-
MAR *-
MAR *-
SAR AR6,TLLH
RECONSTRUCTION2
LAR AR6,TLLH
LAR AR0,RECONLLH
MAR *0+
LAC *,AR1
LDPK 6
LAR AR1,SIGNAREA
```

```
MAR *-
BANZ TACKLE2,*
NEG
TACKLE2 SACL TEMP
LT TEMP
MPY SIGMA_LLH
PAC
SACH TEMP,5
LAC TEMP
ADD MLLH
TACKLE21
LRLK AR1,SLLH
SACL *
ZAC
SACL SIGNAREA
LEVEL_4 LARP AR1
LRLK AR1,ALLL+7
ZAC
MPYK 0
RPTK 7
MAC ANALOW,*-
APAC
LRLK AR1,ALLLL
SACH *,1
SACH TEMPL,1
;  LLLH-BAND
LRLK AR1,ALLL+7
ZAC
MPYK 0
RPTK 7
MACD ANAHIGH,*-
APAC
LRLK AR1,ALLLH
SACH *,1
SACH TEMPH,1
LAR AR5,A4
LARP AR5
BANZ STORE4,AR4
B SYNTH_1
STORE4 LAR AR4,WLLLH
LAC TEMPH
SACL *+
LALK LENGTH4
SACL LENGTH
LALK ADDRLLLH
CALL ADJUST4,AR4
SAR AR4,WLLLH
LAC TEMPL
LAR AR4,WLLLL
SACL *+
LAR AR4,WLLLL
MAR *+
LALK ADDRLLLL
CALL ADJUST4,AR4
```

```
SAR AR4,WLLLL
QUAN_4 LAR AR4,RLLLH
LAC *+,AR1
LRLK AR1,SLLLH
SACL *,AR4
LALK ADDRLLLH
LAR AR4,RLLLH
MAR *+
CALL ADJUST4,AR4
SAR AR4,RLLLH
LAR AR4,RLLLL
LAC *+,AR1
LRLK AR1,SLLLL
SACL *,AR4
LALK ADDRLLLL
CALL ADJUST4,AR4
SAR AR4,RLLLL
; ==========================================================
; ---------------------------------------
; MEAN AND VARIANCE CALCULATION , HIGH BAND LEVEL 4
ZALH SUMLLLH
ADDS SUMLLLH+1
ADD TEMPH
SACH SUMLLLH
SACL SUMLLLH+1 ; MEAN CALCULATION
LRLK AR1,SUMLLLH2
LARP AR1
SQRA TEMPH
ZALH *+
ADDS *-
APAC
SACH *+
SACL *,AR4
; ---------------------------------------
; THE END OF STAT CALCULATION OF LEVEL 1 HIGH BAND
; START OF QUANTIZE LEVEL 1 HIGH BAND
LAR AR4,QRLLLH
LAC *+
SACL BUFFER
LAR AR4,QRLLLH
MAR *+
LALK ADDRLLLH
CALL ADJUST4
SAR AR4,QRLLLH
LARP AR0
LAR AR0,BITLLLH
BANZ QUAN1,*
PUT01 ZAC
B TACKLE11,*,AR1
QUAN1 LAC BUFFER
SUB MLLLH
BGEZ T1
LARK AR7,1
SAR AR7,SIGNAREA
```

```
NEG
T1 LDPK 7
LRLK AR7,SIGMA_LLLH
SACL SAMPLE1
LARP AR7
LAR AR6,LLLH_START
LAR AR5,LLLH_LENGTH
LT *,AR6
LOOP1 LAC SAMPLE1,11
MPY *+,AR5
SPAC
BLEZ FOUND1
BANZ LOOP1,*-,AR6
MAR *+
FOUND1 LARP AR6
LAR AR0,LLLH_START
MAR *0-
MAR *-
MAR *-
SAR AR6,TLLLH
RECONSTRUCTION1
LAR AR6,TLLLH
LAR AR0,RECONLLLH
MAR *0+
LAC *,AR1
LDPK 6
LAR AR1,SIGNAREA
MAR *-
BANZ TACKLE1,*
NEG
TACKLE1 SACL TEMP
LT TEMP
MPY SIGMA_LLLH
PAC
SACH TEMP,5
LAC TEMP
ADD MLLLH
TACKLE11
LRLK AR1,SLLLH
SACL *
ZAC
SACL SIGNAREA
; ---------------------------------------
; LOW BAND , LEVEL 4
ZALH SUMLLLL
ADDS SUMLLLL+1
ADD TEMPL
SACH SUMLLLL
SACL SUMLLLL+1
LRLK AR1,SUMLLLL2
LARP AR1
SQRA TEMPL
ZALH *+
ADDS *-
```

```
APAC
SACH *+
SACL *,AR4
; THE END OF STAT CALULATION OF LEVEL 4, LOW BAND
; START OF QUANTIZE LEVEL 4, LOW BAND
LAR AR4,QRLLLL
LAC *+
SACL BUFFER
SAR AR4,TEMP
LAC TEMP
SBLK ADDRLLLL
BGEZ RT1L
CALL START
RT1L SUB LENGTH
BLEZ NO1L
LRLK AR4,ADDRLLLL
NO1L SAR AR4,QRLLLL
LARP AR0
LAR AR0,BITLLLL
BANZ QUAN1L,*
PUT01L ZAC
B TACKLE1L1,*,AR1
QUAN1L LAC BUFFER
SUB MLLLL
BGEZ T1L
LARK AR7,1
SAR AR7,SIGNAREA
NEG
T1L LDPK 7
LRLK AR7,SIGMA_LLLL
SACL SAMPLE1
LARP AR7
LAR AR6,LLLL_START
LAR AR5,LLLL_LENGTH
LT *,AR6
LOOP1L LAC SAMPLE1,11
MPY *+,AR5
SPAC
BLEZ FOUND1L
BANZ LOOP1L,*-,AR6
MAR *+
FOUND1L LARP AR6
LAR AR0,LLLL_START
MAR *0-
MAR *-
MAR *-
SAR AR6,TLLLL
RECONSTRUCTION1L
LAR AR6,TLLLL
LAR AR0,RECONLLLL
MAR *0+
LAC *,AR1
LDPK 6
LAR AR1,SIGNAREA
```

```
MAR *-
BANZ TACKLE1L,*
NEG
TACKLE1L
SACL TEMP
LT TEMP
MPY SIGMA_LLLL
PAC
SACH TEMP,5
LAC TEMP
ADD MLLLL
TACKLE1L1
LRLK AR1,SLLLL
SACL *
ZAC
SACL SIGNAREA
; ---------------------------------------
;  THE END OF QUANTIZATION
; CHECK IF BLOCK END REACHED
; IF YES, UPDATE BITRATE FOR EACH SUBBAND
LAC LOOPCOUNTER ; LOAD SEGMENT COUNTER
ADDK 1 ; INCREASE BY 1
SACL LOOPCOUNTER ; STORE
SUBK 8 ; SEGMENT END REACHED?
BNZ SYNTH_1 ; NO
SACL LOOPCOUNTER ; YES, DO BITRATE UPDATION
; START OF BITRATE UPDATION
;  CALCULATE SIGMA'S
ZALH SUMH ;
ADDS SUMH+1 ; GET 32-BIT SUMMATION
RPTK 6-1 ; SCALE TO GET MEAN
SFR
SACL MH ; SUMH RIGHT SHIFTED 6 BITS
; TO FORM MEAN OF H-BAND SAMPLES
SQRA MH ; MEANH^2 -> P REG
ZALH SUMH2 ; LOAD AC WITH SUM OF (H-BAND SAMPLE)^2
ADDS SUMH2+1
RPTK 6-1 ;
SFR
; RIGHT SHIFT 6 BITS
; TO FORM MEAN OF SAMPLE^2
; CALCULATE VARIANCE OF H BAND
SPAC ; ACC = <H-BAND^2>-<H-BAND>^2
; WHICH IS VARIANCE^2 OF H-BAND
CALL SQRROOT ; TO GET VARIANCE ITSELF
SACL SIGMA_H ;
; ------------------------------------
; REPEAT WITH LH-BAND
ZALH SUMLH
ADDS SUMLH+1
RPTK 5-1
SFR
SACL MLH
SQRA MLH
```

```
ZALH SUMLH2
ADDS SUMLH2+1
RPTK 5-1
SFR
SPAC
CALL SQRROOT
SACL SIGMA_LH
; ------------------------------------
; REPEAT WITH LLH-BAND
ZALH SUMLLH
ADDS SUMLLH+1
RPTK 4-1
SFR
SACL MLLH
SQRA MLLH
ZALH SUMLLH2
ADDS SUMLLH2+1
RPTK 4-1
SFR
SPAC
CALL SQRROOT
SACL SIGMA_LLH
; REPEAT WITH LLLH-BAND
ZALH SUMLLLH
ADDS SUMLLLH+1
RPTK 3-1
SFR
SACL MLLLH
SQRA MLLLH
ZALH SUMLLLH2
ADDS SUMLLLH2+1
RPTK 3-1
SFR
SPAC
CALL SQRROOT
SACL SIGMA_LLLH
; REPEAT WITH LLLL-BAND
ZALH SUMLLLL
ADDS SUMLLLL+1
RPTK 3-1
SFR
SACL MLLLL
SQRA MLLLL
ZALH SUMLLLL2
ADDS SUMLLLL2+1
RPTK 3-1
SFR
SPAC
CALL SQRROOT
SACL SIGMA_LLLL
; ------------------------------------
BITALLOCATION
LRLK AR0,LOGTABLE ; AR0 POINTS TO THE BEGIN
; ADDRESS OF THE LOGARITHM TABLE
```

```
LRLK AR7,SIGMA_H ; AR7 POINTS TO SIGMA
LRLK AR2,LOG_H ;
CALL FINDLOG,AR7 ; GET LOGARITHM
LRLK AR7,SIGMA_LH ; REPEAT WITH LH-BAND
LRLK AR2,LOG_LH
CALL FINDLOG,AR7
LRLK AR7,SIGMA_LLH ; REPEAT WITH LLH-BAND
LRLK AR2,LOG_LLH
CALL FINDLOG,AR7
LRLK AR7,SIGMA_LLLH ; REPEAT WITH LLLH-BAND
LRLK AR2,LOG_LLLH
CALL FINDLOG,AR7
LRLK AR7,SIGMA_LLLL ; REPEAT WITH LLLL-BAND
LRLK AR2,LOG_LLLL
CALL FINDLOG,AR7
; -----------------------------------
LAC LOG_H,9 ; CALCULATE OPTIMUM BIT ALLOCATION
ADD LOG_LH,8
ADD LOG_LLH,7
ADD LOG_LLLH,6
ADD LOG_LLLL,6
SACH BASE ; STORE ACC/32 TO BASE
; ==================================================
LRLK AR7,LOG_H ; ALLOCATE BIT FOR H-BAND
LRLK AR2,BITH
CALL ALLOCATION,AR7
LRLK AR7,LOG_LH ; ALLOCATE BIT FOR LH-BAND
LRLK AR2,BITLH
CALL ALLOCATION,AR7
LRLK AR7,LOG_LLH ; ALLOCATE BIT FOR LLH-BAND
LRLK AR2,BITLLH
CALL ALLOCATION,AR7
LRLK AR7,LOG_LLLH ; ALLOCATE BIT FOR LLLH-BAND
LRLK AR2,BITLLLH
CALL ALLOCATION,AR7
LRLK AR7,LOG_LLLL ; ALLOCATE BIT FOR LLLL-BAND
LRLK AR2,BITLLLL
CALL ALLOCATION,AR7
; FINAL ADJUSTMENT FOR BAND BITRATE
CHECK LAC BITH,3 ; CALCULATE ACTUAL BITRATE
ADD BITLH,2
ADD BITLLH,1
ADD BITLLLH
ADD BITLLLL
SUBK AVGBIT*4 ; AVGBIT PRE-MULTIPLIED BY 4
; ACC NOW = 4*DELTA
CHECK1 BGZ DECBIT1 ; IF TOO BIG, DECREASE BIT
BLZ INCBIT1 ; IF TOO SMALL, INCREASE BIT
CONCLUDE
ZAC ; BIT ALLOCATION CORRECT
SACL INC_INDI ; RESET INC AND DEC INDICATORS
SACL DEC_INDI
SACL INC_LLLL
SACL INC_LLLH
```

```
SACL INC_LLH
SACL INC_LH
SACL DEC_H
SACL DEC_LH
SACL DEC_LLH
SACL DEC_LLLH
B CREAT_TABLE_ADDR; GOTO CREAT TABLE ADDRESSES
DECBIT1 LAC INC_INDI ; CHECK IF LAST TIME IS INCREASE
BZ TRYH ; IF NOT GOTO DECREASE
B CONCLUDE ; IF YES, POSSIBLY FLUCTUATING, NO DECREASE
TRYH LAC DEC_H ; CHECK IF H-BAND DECREASED LAST TIME
BNZ TRYLH ; IF YES, CHANGE LH-BAND
LAC BITH ; IF NO, CHANGE H-BAND
BZ TRYLH ; IF H-BAND BITRATE ZERO, GOTO CHANGE LH-BAND
SUBK 1 ; NOT ZERO, DECREASE BY 1
SACL BITH ;
B CHECK ; GO BACK TO CHECK OVERALL BITRATE
TRYLH LAC DEC_LH ; SIMILAR TO H-BAND
BNZ TRYLLH
LAC BITLH
BZ TRYLLH
SUBK 1
SACL BITLH
B CHECK
TRYLLH LAC DEC_LLH ; SIMILAR TO H-BAND
BNZ TRYLLLH
LAC BITLLH
BZ TRYLLLH
SUBK 1
SACL BITLLH
B CHECK
TRYLLLH LAC DEC_LLLH ; SIMILAR TO H-BAND
BNZ TRYLLLL
LAC BITLLLH
BZ TRYLLLL
SUBK 1
SACL BITLLLH
B CHECK
TRYLLLL LAC BITLLLL
BZ CONCLUDE
SUBK 1
SACL BITLLLL
DECRESET
ZAC
SACL DEC_LLLH
SACL DEC_LLH
SACL DEC_LH
SACL DEC_H
B CHECK
INCBIT1 LACK 1 ; SET INCREASE INDICATOR
SACL INC_INDI
INCLLLL LAC INC_LLLL ; CHECK IF LLLL HAS EVER BEEN INCREASED
BNZ INCLLLH ; YES GOTO NEXT BAND
LAC BITLLLL ; NO, LOAD BAND BITRATE
```

```
SUBK MAXBIT ; CHECK IF IT REACHES MAXIMUM BITRATE
BGEZ INCLLLH ; IF YES, GOTO NEXT BAND
ADDK 1+MAXBIT ; IF NO, INCREASE BITRATE BY 1
SACL BITLLLL ;
B CHECK ; CHECK NEW BITRATE
INCLLLH LAC INC_LLLH ; SIMILAR TO LLLL-BAND
BNZ INCLLH
LAC BITLLLH
SUBK MAXBIT
BGEZ INCLLH
ADDK 1+MAXBIT
SACL BITLLLH
B CHECK
INCLLH LAC INC_LLH ; SIMILAR TO LLLL_BAND
BNZ INCLH
LAC BITLLH
SUBK MAXBIT
BGEZ INCLH
ADDK 1+MAXBIT
SACL BITLLH
B CHECK
INCLH LAC INC_LH ; SIMILAR TO LLLL_BAND
BNZ INCH
LAC BITLH
SUBK MAXBIT
BGEZ INCH
ADDK 1+MAXBIT
SACL BITLH
B CHECK
INCH LAC BITH ; SIMILAR TO LLLL_BAND
SUBK MAXBIT
BGEZ CONCLUDE
ADDK 1+MAXBIT
SACL BITH
INCRESET
ZAC
SACL INC_LLLL
SACL INC_LLLH
SACL INC_LLH
SACL INC_LH
B CHECK
CREAT_TABLE_ADDR
; ASSIGN TABLE BEGIN ADDRESSES
; ACCORDING TO BAND BITRATE
LRLK AR7,H_START ; H-BAND
LRLK AR2,BITH
CALL CREAT,AR2 ; CREAT QUAN/RECON TABLE ADDR
LRLK AR7,LH_START ; SIMILAR TO H-BAND
LRLK AR2,BITLH
CALL CREAT,AR2
LRLK AR7,LLH_START ; SIMILAR TO H-BAND
LRLK AR2,BITLLH
CALL CREAT,AR2
LRLK AR7,LLLH_START ; SIMILAR TO H-BAND
```

```
LRLK AR2,BITLLLH
CALL CREAT,AR2
LRLK AR7,LLLL_START ; SIMILAR TO H-BAND
LRLK AR2,BITLLLL
CALL CREAT,AR2
RES ZAC ; RESET STATISTICS RAM
SACL SUMH
SACL SUMH+1
SACL SUMH2
SACL SUMH2+1
SACL SUMLH
SACL SUMLH+1
SACL SUMLH2
SACL SUMLH2+1
SACL SUMLLH
SACL SUMLLH+1
SACL SUMLLH2
SACL SUMLLH2+1
SACL SUMLLLH
SACL SUMLLLH+1
SACL SUMLLLH2
SACL SUMLLLH2+1
SACL SUMLLLL
SACL SUMLLLL+1
SACL SUMLLLL2
SACL SUMLLLL2+1
*==========================================================================
; SYNTHESIS STAGE
SYNTH_1 LAC A4 ; CHECK EVEN/ODD INDICATOR
BNZ NZERO_1 ; NOT ZERO, FEED WITH RECONSTRUCTED DATA
LACK 1 ; ZERO, FEED WITH 0
SACL A4 ; TOGGLE E/O INDICATOR
ZAC ;
LRLK AR4,SLLLL ;
SACL * ; STORE TO LLLL-BAND SYNTHESIS BUFFER
LRLK AR4,SLLLH ;
SACL * ; STORE TO LLLH-BAND SYNTHESIS BUFFER
B SYS_1 ; GOTO SYNTHESIS FILTERING
NZERO_1 ZAC ;
SACL A4 ; TOGGLE E/O INDICATOR
SYS_1 LARP AR1 ; AR1 ACTIVE
LRLK AR1,SLLLL+7 ; AR1 POINTS TO LLLL-BAND
ZAC
MPYK 0 ; CLEAR REGISTERS
RPTK 7 ; CONVOLUTION CALCULATION
MACD SYNLOW,*-
APAC
SACH TEMPL,1 ; STORE OUTPUT
LRLK AR1,SLLLH+7 ; LLLH-BAND
ZAC
MPYK 0
RPTK 7
MACD SYNHIGH,*-
APAC
```

```
SACH TEMPH,1
LAC TEMPL
ADD TEMPH
LRLK AR1,SLLL
SACL *
SYNTH_2 ; SIMILAR TO SYNTH_1
LAC A3
BNZ NZERO_2
LACK 1
SACL A3
ZAC
LRLK AR4,SLLL
SACL *
LRLK AR4,SLLH
SACL *
B SYS_2
NZERO_2 ZAC
SACL A3
SYS_2 LARP AR1
LRLK AR1,SLLL+7
ZAC
MPYK 0
RPTK 7
MACD SYNLOW,*-
APAC
SACH TEMPL,1
LRLK AR1,SLLH+7 ; LLH-BAND
ZAC
MPYK 0
RPTK 7
MACD SYNHIGH,*-
APAC
SACH TEMPH,1
LAC TEMPL
ADD TEMPH
LRLK AR1,SLL
SACL *
SYNTH_3 ; SIMILAR TO SYNTH_1
LAC A2
BNZ NZERO_3
LACK 1
SACL A2
ZAC
LRLK AR4,SLL
SACL *
LRLK AR4,SLH
SACL *
B SYS_3
NZERO_3 ZAC
SACL A2
SYS_3 LARP AR1
LRLK AR1,SLL+7
ZAC
MPYK 0
```

```
RPTK 7
MACD SYNLOW,*-
APAC
SACH TEMPL,1
LRLK AR1,SLH+7 ; LH-BAND
ZAC
MPYK 0
RPTK 7
MACD SYNHIGH,*-
APAC
SACH TEMPH,1
LAC TEMPL
ADD TEMPH
LRLK AR1,SL
SACL *
SYNTH_4 ; SIMILAR TO SYNTH_1
LAC A1
BNZ NZERO_4
LACK 1
SACL A1
ZAC
LRLK AR4,SL
SACL *
LRLK AR4,SH
SACL *
B SYS_4
NZERO_4 ZAC
SACL A1
SYS_4 LARP AR1
LRLK AR1,SL+7
ZAC
MPYK 0
RPTK 7
MACD SYNLOW,*-
APAC
SACH OUT1,1
LRLK AR1,SH+7 ; H-BAND
ZAC
MPYK 0
RPTK 7
MACD SYNHIGH,*-
APAC
SACH OUT2,1
LAC OUT1
ADD OUT2
B OUTPUT
**** SUBROUTINE ADJUST4 ****
**** ADJUSTS CURRENT AUXILIAY REGISTER AR4 ****
**** ACCORDING TO ITS CORRESPONDING DATA AREA ****
**** IF REGISTER VALUE EXCEEDS BOTTOM OF ****
**** THE DATA AREA, ADJUSTMENT IS MADE ****
**** DATA AREA LENGTH IS IN "LENGTH" ****
**** DATA AREA BEGIN ADDRESS IS IN ACC ****
ADJUST4
```

```
SAR AR4,TEMP ; STORE INDICATOR TO BE REVISED
SACL ADDR ; STORE BEGIN ADDRESS OF THE BUFFER
LAC TEMP ; LOAD ACC WITH INDICATOR VALUE
SUB ADDR ; CHECK IF IT IS LESS THAN BEGIN ADDR
BLZ ERROR ; IF YES, AND ERROR OF ADDRESSING
; HAS ACCURED. GOTO ERROR HANDLING
SUB LENGTH ; NO, CHECK IF POINTER EXCEEDS END
BLZ NOADJUST ; NO, NO ADJUST
LAR AR4,ADDR ; YES, ADJUST TO BEGIN ADDR
NOADJUST
RET
ERROR B START ; ON ERROR, RESTART THE PROGRAM
**** SUBROUTINE INITIALIZATION ****
INIT ; INITIALIZE RAM
ZAC
LARP AR1
LRLK AR1,A1 ; SET EVEN/ODD INDICATORS TO ZERO
RPTK 9
SACL *+
LRLK AR1,ADDRH ; SET SUBBAND BUFFERS TO ZERO
RPTK 65+21-1
SACL *+
LRLK AR1,ADDRLH
RPTK 33+14-1
SACL *+
LRLK AR1,ADDRLLH
RPTK 17+7-1
SACL *+
LRLK AR1,ADDRLLLH
RPTK 9-1
SACL *+
LRLK AR1,ADDRLLLL
RPTK 9-1
SACL *+
LRLK AR1,SAMPLE ; CLEAR CONVOLUSION BUFFERS
RPTK 8-1
SACL *+
LRLK AR1,ALLLL
RPTK 8-1
SACL *+
LRLK AR1,AH
RPTK 8-1
SACL *+
LRLK AR1,ALH
RPTK 8-1
SACL *+
LRLK AR1,ALLH
RPTK 8-1
SACL *+
LRLK AR1,ALLLH
RPTK 8-1
SACL *+
LRLK AR1,SL
RPTK 8-1
```

```
SACL *+
LRLK AR1,SLL
RPTK 8-1
SACL *+
LRLK AR1,SLLL
RPTK 8-1
SACL *+
LRLK AR1,SLLLL
RPTK 8-1
SACL *+
LRLK AR1,SH
RPTK 8-1
SACL *+
LRLK AR1,SLH
RPTK 8-1
SACL *+
LRLK AR1,SLLH
RPTK 8-1
SACL *+
LRLK AR1,SLLLH
RPTK 8-1
SACL *+
****
LALK ADDRH ; INITIALIZE POINTERS
SACL QRH
ADDK 64
SACL RH
ADDK 21
SACL WH ; H-BAND WRITE ADDR IS 21 AHEAD OF READ
LALK ADDRLH
SACL QRLH
ADDK 32
SACL RLH
ADDK 14
SACL WLH ; LH-BAND WRITE IS 14 SAMPLES AHEAD
LALK ADDRLLH
SACL QRLLH
ADDK 16
SACL RLLH
ADDK 7
SACL WLLH ; LLH-BAND WRITE IS 7 SAMPLES AHEAD
LALK ADDRLLLH
SACL QRLLLH
ADDK 8
SACL RLLLH
SACL WLLLH
LALK ADDRLLLL
SACL QRLLLL
ADDK 8
SACL RLLLL
SACL WLLLL
; MOVE TABLES FROM PROGRAM MEMORY
; TO DATA MEMORY FOR EASY ACCESSING
LRLK AR1,START_1
```

```
LALK START_1P
TBLR *
LRLK AR1,LENGTH_1
LALK LENGTH_1P
TBLR *
LRLK AR1,RECON_1
LALK RECON_1P
TBLR *
LRLK AR1,START_2
LALK START_2P
RPTK 1
TBLR *+
LRLK AR1,LENGTH_2
LALK LENGTH_2P
TBLR *
LRLK AR1,RECON_2
LALK RECON_2P
RPTK 1
TBLR *+
LRLK AR1,START_3
LALK START_3P
RPTK 3
TBLR *+
LRLK AR1,LENGTH_3
LALK LENGTH_3P
TBLR *
LRLK AR1,RECON_3
LALK RECON_3P
RPTK 3
TBLR *+
LRLK AR1,START_4
LALK START_4P
RPTK 7
TBLR *+
LRLK AR1,LENGTH_4
LALK LENGTH_4P
TBLR *
LRLK AR1,RECON_4
LALK RECON_4P
RPTK 7
TBLR *+
LRLK AR1,START_5
LALK START_5P
RPTK 15
TBLR *+
LRLK AR1,LENGTH_5
LALK LENGTH_5P
TBLR *
LRLK AR1,RECON_5
LALK RECON_5P
RPTK 15
TBLR *+
LRLK AR1,START_6
LALK START_6P
```

```
RPTK 31
TBLR *+
LRLK AR1,LENGTH_6
LALK LENGTH_6P
TBLR *
LRLK AR1,RECON_6
LALK RECON_6P
RPTK 31
TBLR *+
LRLK AR1,START_7
LALK START_7P
RPTK 63
TBLR *+
LRLK AR1,LENGTH_7
LALK LENGTH_7P
TBLR *
LRLK AR1,RECON_7
LALK RECON_7P
RPTK 63
TBLR *+

LRLK AR1,TABLE_START
LALK START_1
SACL *+,AR2
LRLK AR2,LENGTH_1
LAC *+,AR1
SACL *+
LALK RECON_1
SACL *+
LALK START_2
SACL *+,AR2
LRLK AR2,LENGTH_2
LAC *+,AR1
SACL *+
LALK RECON_2
SACL *+
LALK START_3
SACL *+,AR2
LRLK AR2,LENGTH_3
LAC *+,AR1
SACL *+
LALK RECON_3
SACL *+
LALK START_4
SACL *+,AR2
LRLK AR2,LENGTH_4
LAC *+,AR1
SACL *+
LALK RECON_4
SACL *+
LALK START_5
SACL *+,AR2
LRLK AR2,LENGTH_5
LAC *+,AR1
```

```
SACL *+
LALK RECON_5
SACL *+
LALK START_6
SACL *+,AR2
LRLK AR2,LENGTH_6
LAC *+,AR1
SACL *+
LALK RECON_6
SACL *+
LALK START_7
SACL *+,AR2
LRLK AR2,LENGTH_7
LAC *+,AR1
SACL *+
LALK RECON_7
SACL *
LRLK AR1,LOGTABLE
LALK LOGTABLE1
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0100h
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0200h
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0300h
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0400h
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0500h
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0600h
RPTK 0100h-1
TBLR *+
LALK LOGTABLE1+0700h
RPTK 0100h-1
TBLR *+
ZAC
SACL LOOPCOUNTER ; RESET SEGMENT COUNTER
; RESET STATISTICS BUFFERS
SACL SUMH
SACL SUMH+1
SACL SUMH2
SACL SUMH2+1
SACL SUMLH
SACL SUMLH+1
SACL SUMLH2
SACL SUMLH2+1
SACL SUMLLH
```

```
SACL SUMLLH+1
SACL SUMLLH2
SACL SUMLLH2+1
SACL SUMLLLH
SACL SUMLLLH+1
SACL SUMLLLH2
SACL SUMLLLH2+1
SACL SUMLLLL
SACL SUMLLLL+1
SACL SUMLLLL2
SACL SUMLLLL2+1
RET
***********************************
**** SUBROUTINE SQRROOT
**** INPUT IN ACC
**** OUTPUT IN ACC
***********************************
SQRROOT ; OUTPUT = INPUT^0.5
LDPK 7
SACH RT_TEMPH
SACL RT_TEMP ; TEMP HOLDS -INPUT
ZAC
SACL RT_OUT ; INITIALIZE OUTPUT BUFFER
LACK 4000H
SACL RT_TRY ; STORE FIRST INCREMENT VALUE
; DECREASED BY HALF IN EVERY LOOP
RT_LOOP SACL RT_MED ; STORE TRY VALUE
SQRA RT_MED ; P HOLDS SQUARE OF TRY VALUE  X
ZALH RT_TEMPH
ADDS RT_TEMP
SPAC
BLZ RT_NOADD
RT_ADD LAC RT_MED ; X^2 <= INPUT
SACL RT_OUT ; STORE TRY VALUE TO OUTPUT BUFFER
LAC RT_TRY,15 ;
SACH RT_TRY ;
LAC RT_TRY ; LOAD ACC WITH HALVED INCREMENT VALUE
BZ RT_END
ADD RT_MED ; ADD NEW INCREMENT TO LAST TRY VALUE
B RT_LOOP ; COMPARE AGAIN
RT_NOADD
LAC RT_OUT ;
SACL RT_MED ; PUT LAST (SMALLER) TRY VALUE BACK
LAC RT_TRY,15 ;
SACH RT_TRY ;
LAC RT_TRY ; LOAD ACC WITH HALVED INCREMENT VALUE
BZ RT_END
ADD RT_MED ; ADD NEW INCREMENT TO THE RESTORED
; TRY VALUE
B RT_LOOP ; COMPARE AGAIN
RT_END LAC RT_OUT
LDPK 6
RET
*******************************
```

```
***** SUBROUTINE FINDLOG
***** ASSUME AR7 ACTIVE
***** USE AR0,AR1,AR2,AR7
*******************************
FINDLOG LAC *,14,AR1
SACH TEMP
LAR AR1,TEMP
MAR *0+
LAC *,AR2
SACL *
RET
*******************************
***** SUBROUTINE ALLOCATION
***** ASSUME AR7 ACTIVE
***** USE AR2,AR7
*******************************
ALLOCATION
LAC *,10,AR2 ; LOAD ACC WITH LOG
SACH BUFFER ; LOG*2^(-6) -> BUFFER
LAC BUFFER ; LOAD ACC WITH BUFFER
SUB BASE ; GET BITRATE OFFSET
ADDK AVGBIT ; AVGBIT PREMUTIPLIED BY 2^2
SFR
SFR ; GET ACTUAL BITRATE
SUBK MAXBIT
BGZ SATU
ADDK MAXBIT
B ZB
SATU LACK MAXBIT
SACL *
RET
ZB BGEZ STOREB ; IF BITRATE LESS THAN "0", SET TO "0"
ZAC
STOREB SACL *
RET
*****************************
***** SUBROUTINE CREAT
***** ASSUME AR2 ACTIVE
***** USE ACC, AR1,AR2,AR7
*****************************
CREAT LAC * ; LOAD ACC WITH BAND BITRATE
BZ DONE   ; IF ZERO, NO NEED TO CREAT
ADD *,1,AR1 ; ACC CONTAINS 3*(BITH)
ADLK TABLE_START-3 ; ADD TO TABLE_START TO GET ENTRY
SACL TEMP ; STORE TO TRANSFER TO AR1
LAR AR1,TEMP ; LOAD AR1 WITH ENTRY
LAC *+,AR7 ; LOAD ACC WITH ENTRY CONTANT
SACL *+,AR1 ; SEE TABLE ADDRESS ARRANGEMENT SCHEME
LAC *+,AR7
SACL *+,AR1
LAC *,AR7
SACL *+,AR1
DONE RET
.END
```

Bibliography

[1] P. P. Vaidyanathan, "Quadrature Mirror Filter Banks, M-Band Extensions and Perfect Reconstruction Techniques," *IEEE ASSP Magazine*, pp.4–20, July 1987.

[2] R. A. Haddad, "A Class of Orthogonal Nonrecursive Binomial Filters," *IEEE Trans. Audio and Electroacoustics*, pp.296–304, Dec. 1971.

[3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1984.

[4] M. Smith and T. P. Barnwell, "Exact Reconstruction Techniques for Tree Structured Subband Coders," *IEEE Trans. ASSP*, pp.434-441,1986.

[5] A. N. Akansu, R. A. Haddad and H. Caglar, "The Binomial QMF-Wavelet Transform for Multiresolution Signal Decomposition," *Submitted to IEEE Trans. ASSP.*

[6] R. A. Haddad and A. N. Akansu, "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing," *IEEE Trans. ASSP*, pp.723-727, March 1991.

[7] A. N. Akansu and Y. Liu, "On Signal Decomposition Techniques," *Optical Engineering*, pp.912-920, July 1991.

[8] P. Battenschlag, *An Implementation of MHT Algorithm for Adaptive Transform Coding of Speech*, M.S. Thesis, NJIT, May 1990.

[9] *Second-Generation TMS320 User's Guide*, Texas Instruments, 1989.