A Real Time Speech Subband Coder Using the TMS32010

APPLICATION REPORT: SPRA167

Authors: T. Barnwell, R. Schafer, R. Mersereau, and D. Smith
Atlanta Signal Processors Incorporated

Digital Signal Processing Solutions
1989

TEXAS INSTRUMENTS

TRADEMARKS

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

A Real Time Speech Subband Coder Using the TMS32010

Abstract

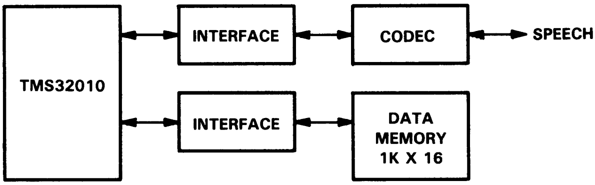
This report describes a potential candidate for the 16-kbit/s transcoder, which is the subband coder. A reprinted article on the subject is included, which describes the theory, TMS320120 implementation requirements, and performance for the 16-kbit/s subband coder. The complete code for the coder is available for licensing from Atlanta Signal Processors Inc. (ASPI). Interested readers should contact ASPI for further information.

Product Support on the World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

PREFACE

A full-duplex 16-kbit/s subband coder can be implemented using a single TMS32010 (without external program memory) with a 1K-word off-chip RAM data buffer. The compression algorithm achieves toll-quality speech with a DRT (Diagnostic Rhyme Test) score of 94 percent. Because of the compactness of the subband coding, it only requires 80 percent of the TMS32010 CPU utilization, thus allowing the processor to perform other tasks in addition to speech coding. A simplified block diagram of the coder is shown in the following figure:



Simplified Block Diagram of the 16-kbit/s Subband Coder

The subband coder can be used in many applications, such as telecommunications and computers. In a transcoder design for digital telephony, the subband coder is valuable because of the following features: relative insensitivity to transmission errors, small bandwidth requirement, toll-quality speech, and simple, economical hardware configuration. In a voice-mail system, speech data rate directly corresponds to the storage required in the memory. The 16-kbit/s subband coder provides an efficient tradeoff between speech quality and data rate. In addition, the use of a single TMS32010 coder allows a cost-effective system implementation.

The following reprinted article details the theory and realization of the subband coder. The TMS32010 source code and documentation of the 16-kbit/s coder is available for licensing from Atlanta Signal Processors Incorporated. For more information, contact:

Atlanta Signal Processors Incorporated
770 Spring Street, Suite 208
Atlanta, Georgia 30308
(404) 892-7265

INTRODUCTION

Over the past several years, the *Subband Coder* has taken its place as one of the highest quality and most cost effective approaches to the coding of speech at medium bit rates. Table 1 illustrates the relationship of the subband coder to a number of other popular approaches to speech coding. Fundamentally, of course, any coding operation must distort the coded signal in some way. The goal of a good coding technique is to constrain the distortion so as to have minimum impact on the overall system performance. In most speech communications systems and voice response systems, this translates into minimizing the perceived differences between the coded and uncoded speech signals.

Modern speech coding systems typically use three classes of features in order to minimize the perceived coding distortion: the characteristics of the auditory system; the characteristics of the vocal tract; and the characteristics of language and individual talkers. Of these, only the first two can be effectively used in medium bit rate systems such as subband coders. Typically, three separate characteristics of the auditory system can be used to advantage. These are (1) the aural noise masking effect, (2) the frequency variant sensitivity of auditory perception (critical bands), and (3) the relative phase insensitivity of the ear (within critical bands). Likewise, a vocal tract model can either be used in a long term statistical sense, or in a short-time stationary time-varying model. If the short-time stationary model is used, then it, in turn, may be applied in three ways: (1) by using syllabic energy variations; (2) by using an explicit slowly time-varying vocal tract model; or (3) by using the pseudo-periodicity (pitch) in voiced sounds.

The simplest common speech coding technique, linear Pulse Code Modulation (PCM), makes essentially no assumptions (except as to dynamic range) about the characteristics of the signal being coded or the eventual use of the coded signal. As a result, linear PCM systems require the greatest bit rate to generate toll quality speech. Such systems have the advantage, however, that since they show no preference toward any particular class of signals, they may be used to code signals which are neither spoken nor heard (such as data signals). Companded Pulse Code Modulation systems, such as the 64 Kbps mu-law companded system used in telephone switching networks, make direct use of the ear's noise masking characteristics in its most basic form. In the resulting systems, the noise is correlated with the signal, which is good, but noise also spreads throughout the frequency range with no regard to the presence of signal energy, which is bad. Companded PCM systems are extremely simple and inexpensive to implement, but the rate at which they must operate to achieve toll quality speech is still relatively high.

Systems such as Differential Pulse Code Modulation (DPCM) and Delta Modulation (DM) make direct use of a long-term stationary statistical model for speech production. Adaptive Differential Pulse Code Modulation (ADPCM) and Adaptive Delta Modulation (ADM) also make use of the slowly varying nature of the short-time energy, causing the noise to be heavily correlated with the speech signal, and causing a dramatic drop in the idle channel noise. Various forms of both the Adaptive Transform Coder (ATC) and the Adaptive Predictive Coder (sometimes called the residual excited vocoder) make use of all of the auditory and vocal tract features, and such systems are capable of generating excellent quality speech at medium to low bit rates. However, such systems are also generally both very complex and quite sensitive to both background noise and transmission errors. Pitch-excited vocoders in general, and LPC vocoders in particular, are capable of operating at low to very low bit rates, but they generally can never achieve toll quality and do not perform well on either noisy speech or non-speech signals.

With this perspective, it is easy to see the advantages and disadvantages of the subband coder as compared to other speech coding systems. Since the subband coder uses only the characteristics of the auditory system in reducing the perceived distortion, it is capable of coding both speech and non-speech signals with good perceived fidelity. Similarly, subband coders can handle multiple simultaneous voices with no increase in degradation. In addition, since subband coders do not include the use of a time-varying vocal tract model, they are far less complex and far less sensitive to transmission bit errors than APC, ATC, or pitch-excited vocoders. This makes the subband coder a very good choice for real communications and telephony systems where the background noise is not controlled and where high transmission error rates might occasionally be expected. Likewise, the low complexity of subband coder realizations make them well matched to implementations on such signal processing micro-computers as the Texas Instruments TMS320 and the NEC 7720. The obvious applications for subband coders for speech include mobile radio, mobile telephony, and voice switching networks.

SUBBAND CODER THEORY

The concept of subband coding for speech was first introduced by Crochiere et al. [1-2]. In a subband coding system, the speech signal is divided into bands by a filter bank (Fig. 1), and then the outputs of the individual bands are first decimated, typically at their nominal Nyquist rate, and then coded for transmission. At the receiver, the coded signals from the individual bands are first decoded, and then padded with zeros and passed through a set of interpolation filters before being summed to form an estimate of the input signal. Such systems can be thought of as exhibiting two classes of distortion: analysis-reconstruction distortion, which is defined as the distortion which would be present if the subband signals were not coded; and coding distortion, which is the distortion due directly to the coding operations themselves. The analysis-reconstruction system alone can exhibit three separate types of distortion: interband aliasing distortion; frequency domain magnitude distortion (which will be called "frequency distortion"); and phase distortion. In the original subband coding systems, infinite impulse response (IIR) filters were used, and the resulting coding systems exhibited all four types of distortion: interband aliasing; frequency distortion; phase distortion; and coding distortion. In later work, Esteban et al. [3] introduced the concept of quadrature mirror filters, which can be used to realize two-band analysis-reconstruction systems which exhibit no interband aliasing. Further, Esteban showed that if equal length linear phase filters are used for the band splitting, then the overall analysis-reconstruction transfer function is a linear phase. Hence, quadrature mirror filters allow for analysis-reconstruction systems which exhibit only frequency distortion.

Based on these results, Esteban et al. [4] and Crochiere et al. [5-6] developed subband speech coding systems based on octave band tree structures of quadrature mirror filters. The octave band structure has the advantage that it can simultaneously approximate the critical band structure of the ear while also utilizing quadrature mirror filters. Such systems exhibit an overall system response which is linear phase, and as such they can be used to code data signals as well as speech signals. The filters used by Crochiere et al. [5] were designed by Johnston [7] using an iterative approach which sought to minimize the frequency distortion in the analysis-reconstruction system. Another characteristic of these later subband coders is that the band-splitting filters were realized using a two-band polyphase filter bank [8] to improve the computational efficiency. In very recent work, Smith and Barnwell [9-10] have presented a technique for designing tree-structured analysis-reconstruction filter banks which allow for exact reconstruction. Such systems, which allow for the design of filter banks of arbitrary quality, are very attractive since they exhibit only coding distortions. However, they sometimes require slightly more computation since they cannot always use the polyphase structure directly in implementing the filter banks.

The octave band subband coders used in speech coding applications are typically formed as a tree structure of analysis-reconstruction pairs based on quadrature mirror or exact reconstruction filters. Fig. 2(a) shows the block diagram for such a two-band system. For the lower channel, the input signal, $x(n)$, is first passed through a half-band low-pass filter with impulse response $h_0(n)$, and then decimated at a 2-to-1 rate. For the upper channel, the signal is passed through a half-band high-pass filter with the impulse response $h_1(n)$, followed by another 2-to-1 decimator. The output of these two channels, $y_0(n)$ and $y_1(n)$, are the two signals which would be coded in a two-band subband coding system. For the reconstruction, these two signals are padded with zeros at a 1-to-2 rate, passed through the two interpolation filters with impulse responses $g_0(n)$ and $g_1(n)$, respectively, and the outputs are summed to form the estimation of the original input, $\hat{x}(n)$.

Let us define the transfer functions of the two analysis filters as

$$H_0(z) = \sum_{n=-\infty}^{\infty} h_0(n)z^{-n} \tag{1a}$$

$$H_1(z) = \sum_{n=-\infty}^{\infty} h_1(n)z^{-n} \tag{1b}$$

and the Fourier transform of these two transfer functions by

$$H_0[\omega] = H_0(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_0(n)e^{-j\omega} \tag{2a}$$

$$H_1[\omega] = H_1(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_1(n)e^{-j\omega} \tag{2b}$$

Then, for quadrature mirror filters, the high-pass filter transfer function can be defined in terms of the low-pass filter by

$$H_1(z) = H_0(-z) \tag{3}$$

or equivalently by

$$H_1[\omega] = H_0[\omega-\pi] \tag{4}$$

In comparison, the Smith-Barnwell exact reconstruction filters are related by

$$H_1(z) = H_0(-1/z) \tag{5}$$

or equivalently by

$$H_1[\omega] = H_0[-\omega-\pi] \tag{6}$$

For quadrature mirror filters, this leads to the relationship

$$h_1(n) = (-1)^n h_0(n) \tag{7}$$

and for the Smith-Barnwell filters, the relationship becomes

$$h_1(n) = (-1)^n h_0(-n) \tag{8}$$

Fig. 2(b) gives the equivalent polyphase filter structure [8] for the two-band analysis-reconstruction system of Fig. 2(a). For the quadrature mirror filters, the transfer function for the low-pass filter can be expressed as

$$H_0(z) = P_0(z^2)+P_1(z^2)z^{-1} \tag{9}$$

and the polyphase filters can be defined as

$$p_0(n) = h_0(2n) \tag{10a}$$

$$p_1(n) = h_0(2n+1) \tag{10b}$$

This is the well known polyphase result that the impulse response of the polyphase filters for a two-band system is formed by taking every second sample of the impulse response of the prototype low-pass filter. The polyphase filter structure reduces the required number of multiplies by a factor of two over the filter structure of Fig. 2(a). The polyphase filter structure can also be used with Smith-Barnwell exact reconstruction filters under the condition that

$$h_0(n) = h_0(-n) \tag{11}$$

or, equivalently, $h_0(n)$ must be a zero (or equivalently, a linear) phase filter. Although this condition can sometimes be met for Smith-Barnwell filters, the filter design problem for this special case is not well understood, and no realizations of this type have yet been demonstrated [9-10]. For those cases which are well understood (including the ones discussed below), this condition is not met and polyphase realizations are not possible.

In the traditional quadrature mirror reconstruction, the reconstruction filters can be defined in terms of their polyphase filters as

$$q_0(n) = p_1(n) \tag{12a}$$

$$q_1(n) = p_0(n) \tag{12b}$$

From Fig. 2(c), it is obvious that this leads to an overall system transfer function given by

$$\hat{X}(z) = C(z)X(z) = P_0(z^2)P_1(z^2)X(z) \tag{13}$$

What this shows is that for the case of quadrature mirror filters, the entire analysis-reconstruction system behaves like a linear filter.

For the Smith-Barnwell exact reconstruction filters, the equivalent characterization of the reconstruction system is given by

$$G_0(z) = H_0(1/z) = H_1(-z) \tag{14a}$$

$$G_1(z) = H_1(1/z) = H_0(-z) \tag{14b}$$

This, of course, leads to exact reconstruction, or equivalently, a linear system C(z) which is distortionless. In particular, if the analysis and reconstruction filters chosen are both length L FIR filters, then C(z) is an ideal delay given by

$$C(z) = z^{-2L} \tag{15}$$

This condition is also approximately true for the Johnston quadrature mirror filters. The primary point here in terms of the realization of subband coding algorithms is that, even under the best of conditions, a two-band analysis-reconstruction system still behaves as an ideal delay. This implies the need for additional storage for delay memory in tree structured realizations.

TREE STRUCTURED SUBBAND CODERS

Two-band subband coders based on the two-band analysis-reconstruction systems of the type we have been discussing are of relatively little interest for speech coding themselves, since they offer little gain over full-band ADPCM. However, such two-band systems can be combined in tree structures to produce effective coding structures for speech. This tree-structuring procedure has the advantage that it can take direct advantage of the properties of half-band quadrature mirror filters and Smith-Barnwell filters in filter structures more appropriate to the critical band nature of aural perception.

The basic structure of an octave band subband coder for speech is illustrated in Fig. 3 where the band-splitters and band-mergers are typically implemented using the polyphase forms shown in Fig. 4. A key issue in the octave band structure is the compensation filters for bands which have not been split. It has been shown [11] that for both the quadrature mirror filters and for Smith-Barnwell filters, the compensation filters can be adequately realized using ideal delays. The basic problem is that in a tree structured environment, these delays "stack", and the required delay memory size can become quite large. In particular, if there are N band-splitters in an octave band subband coder, then the compensation delay required by the q^{th} band is given by

$$C_q(z) = z^{-d(q)} \tag{16}$$

where d(q) is given by

$$d(q) = \sum_{p=1}^{N-q} L(p+q)2^p \tag{17}$$

and where L(p) is the length of the FIR analysis and reconstruction filters in the p^{th} band-splitting operation. This delay storage requirement can be quite large if a large number of octave bands is used. This translates into a total delay storage requirement for the entire system of

$$D = \sum_{p=1}^{N} L(p)2^p \tag{18}$$

TMS320 REALIZATIONS

In designing subband coder implementations for a fast signal processing microprocessor such as the TMS320, it is important to determine exactly which of the microcomputer resources (data memory, program memory, or real-time) constrains the form of the implementations. Since the TMS320 has a relatively large program address space (4K sixteen bit words) and since the subband coder algorithms are quite compactly programmed, it is clear that program address space is not an issue of concern. However, this is certainly not true of data memory, which is in rather short supply (144 words total), and real-time might also be an issue. A program for a subband coder can be functionally divided into five parts: input-output operations; analysis and reconstruction filter implementations; delay compensation implementation; channel quantization implementation; and control.

The main loop in a subband coder program must operate on frames of input samples where the frame length is such as to result in an integer number of data points being coded by the deepest branch coder in the tree structure. In order to compute the number of cycles per sample required by the subband coder main loop, it is hence correct to compute the number of cycles required by the entire main loop and then to divide by the frame length. To compute the actual number of samples required by the entire realization, the number of cycles per sample which are required by the interrupt based I/O buffering routines must also be included.

The minimum number of sample points which is required for a data frame in order to meet the above constraint is given by

$$M = 2^N \tag{19}$$

where, as before, N is the number of band-splitters in the coder structure. In an interrupt driven implementation, at least M data locations must be available for I/O buffering. In addition, the total number of data locations necessary to support the filter and delay structure is given by

$$K = \sum_{p=1}^{N} L(p)2^p \tag{20}$$

This includes both the delay storage needed in the filter implementations and the delay storage needed for the compensation delays. In addition to this storage, our current implementations use two storage locations per APCM coder, two storage locations for the interrupt routine, and four temporary storage locations for various utility operations. Hence, the total number of storage locations required can be approximated as

$$NN = M + K + 2N + 6 \tag{21}$$

Table 2 gives a tabulation of NN versus N for a "typical" FIR filter structure given by $L(1) = 32$, $L(2) = L(3) = L(4) = 16$, and $L(5) = 8$ for both full duplex and half duplex realizations. Since the TMS320 has only 144 data storage locations, clearly all implementations which use more than 144 locations must be realized through the addition of a read and write capable off-chip random access memory. This, of course, adds considerably to the cost of the implementation. In addition, this "external program" RAM must be accessed using the Table Read and Table Write TMS320 operations, which require three cycles per access and which also require the use of the arithmetic accumulator. Hence, this memory can only be accessed at a comparatively low rate. Table 2 shows that a three band (two band-splitters) system is the largest subband coder which can be realized in full duplex without external program RAM, and a four band system (three band-splitters) is the largest subband coder which can be realized in half duplex without external program RAM. It is clear from this analysis that data memory is a major constraining factor in subband coder realizations.

The real-time constraints in any subband coder realization, of course, are highly dependent on the sampling rate at which the coder is implemented. For toll quality speech, a sampling rate of eight kilohertz is appropriate, whereas for lower bit rate systems, sampling rates as low as 6.4 kilohertz might well be used. Since access to external program RAM, if used, must be performed by means of Table Reads and Table Writes, the programs must run much slower if more than 144 data storage locations are required. For the purposes of this paper, we will first compute the approximate number of cycles per sample point required for all of the operations in the subband coder excluding the data exchanges needed to support external program RAM. This will show approximately how much real-time remains for the implementation of the external program RAM data transfers. From this analysis and an analysis of the data transfer rates required by the Table Read and the Table Write operations, it will be obvious that essentially any reasonable subband coder for speech can be implemented in real-time on a single TMS320 equipped with adequate external program RAM.

For a subband coder environment, the total number of cycles per sample point necessary to realize the arithmetic operations of the analysis and reconstruction filters is simply computed as

$$MM = \sum_{p=1}^{N} L(p)2^{1-p} \tag{22}$$

current realization requires 24 cycles, while the control overhead per sample requires less than 50 cycles per sample per band-splitter. In addition, the APCM coders require approximately 60 cycles for both the coder and the decoder (120 cycles total for full duplex operation), while the delay functions require less than 10 cycles per sample. Hence, the approximate number of cycles per sample can be written as

$$LL = MM + 120N + 24 + 60 \sum_{p=1}^{N} 2^{1-p} \tag{23}$$

Table 3 gives a tabulation for the approximate percent of real-time required as a function of N excluding the time required to access external program RAM for a sampling rate of eight kilohertz. As would be expected, the real-time requirement is nowhere near linear with N, and there is still over 25% of real-time available even at $N=5$. Now let us consider the additional requirement imposed by the slow access of the external program RAM. The first point to note is that the vast majority of the required delay memory is not used in the filter computations, but is used to implement the compensation delays. The number of memory locations required by the filters is given by

$$MA = \sum_{p=1}^{N} 2L(p) \tag{24}$$

Since, for the example shown, MA has a maximum value of 176, 84 for the receiver and 84 for the transmitter, then at most two data context switches must be computed for a full duplex realization and no data context switches must be performed for half duplex realizations. Since the data delay operations can all be performed using only one data memory location and paired Table Read and Table Write operations and since only block accesses are required, then no additional data memory is required and the total number of memory locations accessed per frame is given by

$$MB = \sum_{p=1}^{N} 2^{p-1} \tag{25}$$

Since these operations must be done once per frame, and they can be performed in six cycles per data point, the total number of additional cycles is given by

$$NA = [2MA + 2MB]6/2^{N} \tag{26}$$

As can be seen in Table 4, all of the example subband coders can still be implemented in real-time.

CONCLUSIONS

Based on the above analyses and the associated realizations, it is clear that a TMS320 is well matched to the subband coder implementation problem for voice applications. For the subband coder realization, it was found that the delay compensation memory requirements dominated the realization, and reasonable implementations always required external program RAM. However, when this is made available, virtually any subband coder of interest can be implemented using the same hardware environment.

REFERENCES

1. R. E Crochiere, S. A. Webber, and J. L. Flanagan, "Digital coding of speech in sub-bands," *Bell Syst. Tech. J.*, vol. 55, pp. 1069-1085, Oct. 1976.
2. R. E. Crochiere, "On the design of sub-band coders for low-bit-rate speech communications," *Bell Syst. Tech. J.*, vol. 56, pp. 747-770, May-June, 1977.
3. A. Croisier, D. Esteban, and G. Galand, "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques," presented at the 1976 Int. Conf. Inform. Sci. Syst., Patras, Greece, 1976.
4. D. Esteban and C. Galand, "Applications of quadrature mirror filters to split band or voice coding schemes," *Proc. 1977 Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, May 1977, pp. 191-195.
5. A. J. Barabell and R. E. Crochiere, "Sub-band coder design incorporating quadrature mirror filters and pitch detection," *Proc. 1979 Int. Conf. Acoust., Speech, Signal Processing*, Washington, D.C. Apr. 1979.
6. R. E. Crochiere, "A novel approach for implementing pitch prediction in sub-band coders," *Proc. 1979 Int. Conf. Acoust., Speech, Signal Processing*, Washington, D.C. Apr. 1979.
7. J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," *Proc. 1980 Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, Apr. 1980.
8. M. G. Bellanger, G. Bonnerst, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 109-114, Apr. 1976.
9. M. J. T. Smith and T. P. Barnwell III, "Exact reconstruction techniques for tree-structured subband coders," accepted for publication, *IEEE Trans. Acoust., Speech, Signal Processing*.
10. M. J. T. Smith and T. P. Barnwell III, "A procedure for designing exact reconstruction filter banks for tree-structures," *Proc. 1984 Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, Mar. 1984.
11. T.P. Barnwell III, "Subband coder design incorporating recursive quadrature filters and optimum ADPCM coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 751-765, Oct. 1982.

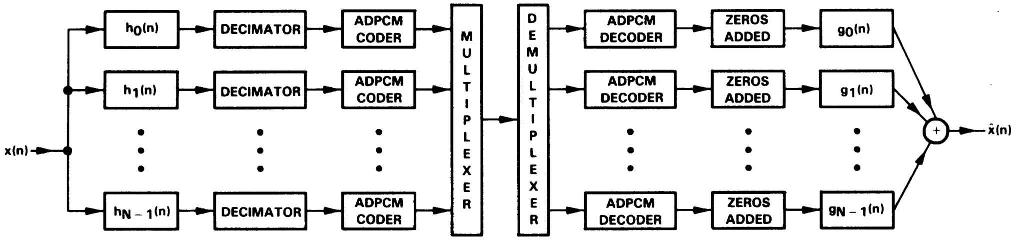


Fig. 1. N-band analysis-reconstruction system for speech coding

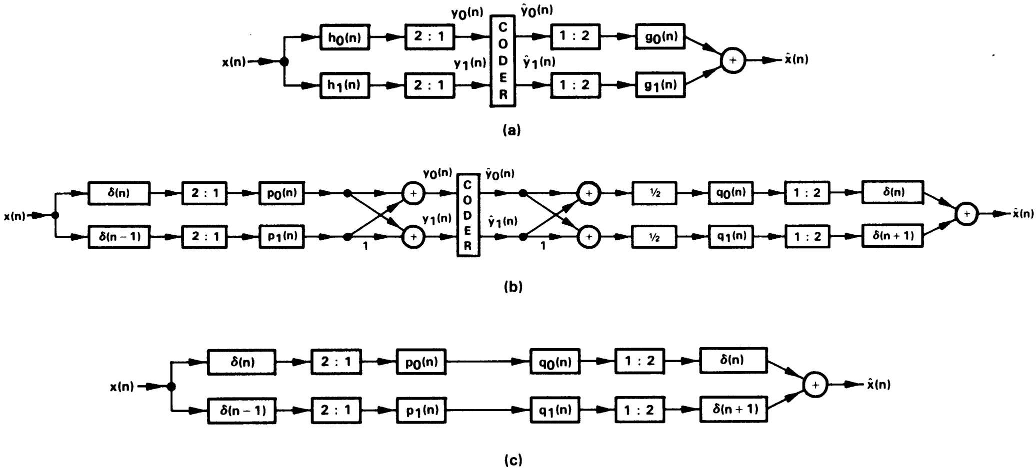


Fig. 2. (a) Two-band analysis-reconstruction band splitter
(b) Equivalent polyphase structure for a two-band analysis-reconstruction band-splitter
(c) Equivalent polyphase structure for analyzing the reconstruction

TABLE 1
Summary of Speech Coding Techniques

Coding Technique	Effective Bit Rates	Aural Noise Masking	Aural Frequency	Aural Phase	Syllabic Energy	Vocal Tract Model	Short Time Stationarity	Pitch
Linear PCM	120 Kbps-80 Kbps	No	No	No	No	No	No	No
Companded PCM	100 Kbps-50 Kbps	Yes	No	No	No	No	No	No
Delta Modulation	80 Kbps-50 Kbps	Yes	No	No	No	Yes	No	No
DPCM	80 Kbps-40 Kbps	Yes	No.	No	No	Yes	No	No
ADM	40 Kbps-16 Kbps	Yes	No	No	Yes	Yes	No	Maybe
ADPCM	40 Kbps-16 Kbps	Yes	No	No	Yes	Yes	No	Maybe
Subband Coder	32 Kbps-10 Kbps	Yes	Yes	Maybe	Yes	No	No	Maybe
ATC	32 Kbps-7 Kbps	Yes	Yes	Maybe	Yes	Yes	Yes	Yes
APC	32 Kbps-4 Kbps	Yes	No	No	Yes	Yes	Yes	Maybe
LPC Vocoder	2.4 Kbps-.6 Kbps	Yes	No	Yes	Yes	Yes	Yes	Yes

TABLE 2
Storage Requirements For Typical Subband Coders

Number of Band-Splits	Locations (Full Duplex)	Locations (Half Duplex)	I/O Buffer (M)	Delay Memory (K)
1	76	52	2	64
2	144	77	4	128
3	278	144	8	256
4	536	275	16	512
5	818	364	32	768

TABLE 3
Total Cycles of TMS320 and Percent Real-Time Excluding External Program RAM Access Time at Eight Kilohertz Sampling Rate

Number of Bands	Cycles Per Samples	Percent Real-Time
1	268	43
2	374	59
3	427	68
4	454	72
5	467	74

TABLE 4
Total Cycles of TMS320 and Percent Real-Time Including External Program RAM Access Time at Eight Kilohertz Sampling Rate

Number of Bands	Cycles Per Samples	Percent Real-Time
1	268	43
2	374	59
3	617	98
4	584	93
5	540	87

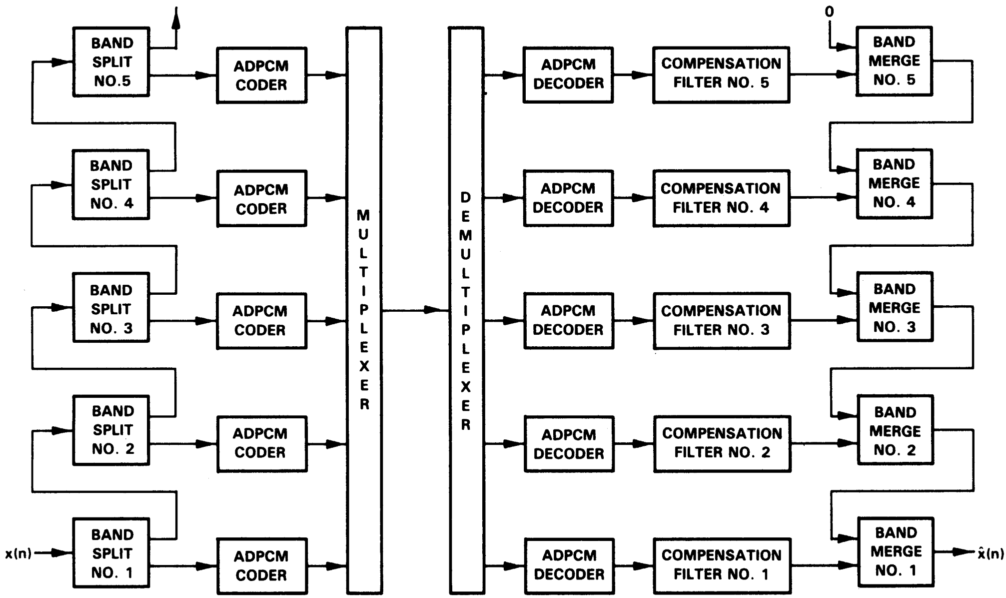
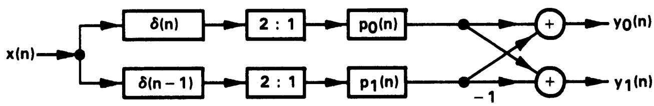
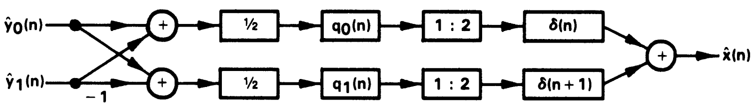


Fig. 3. Octave band subband coding system



(a)



(b)

Fig. 4. (a) Band-splitting module for octave band subband coder
(b) Band merging module for octave band subband coder