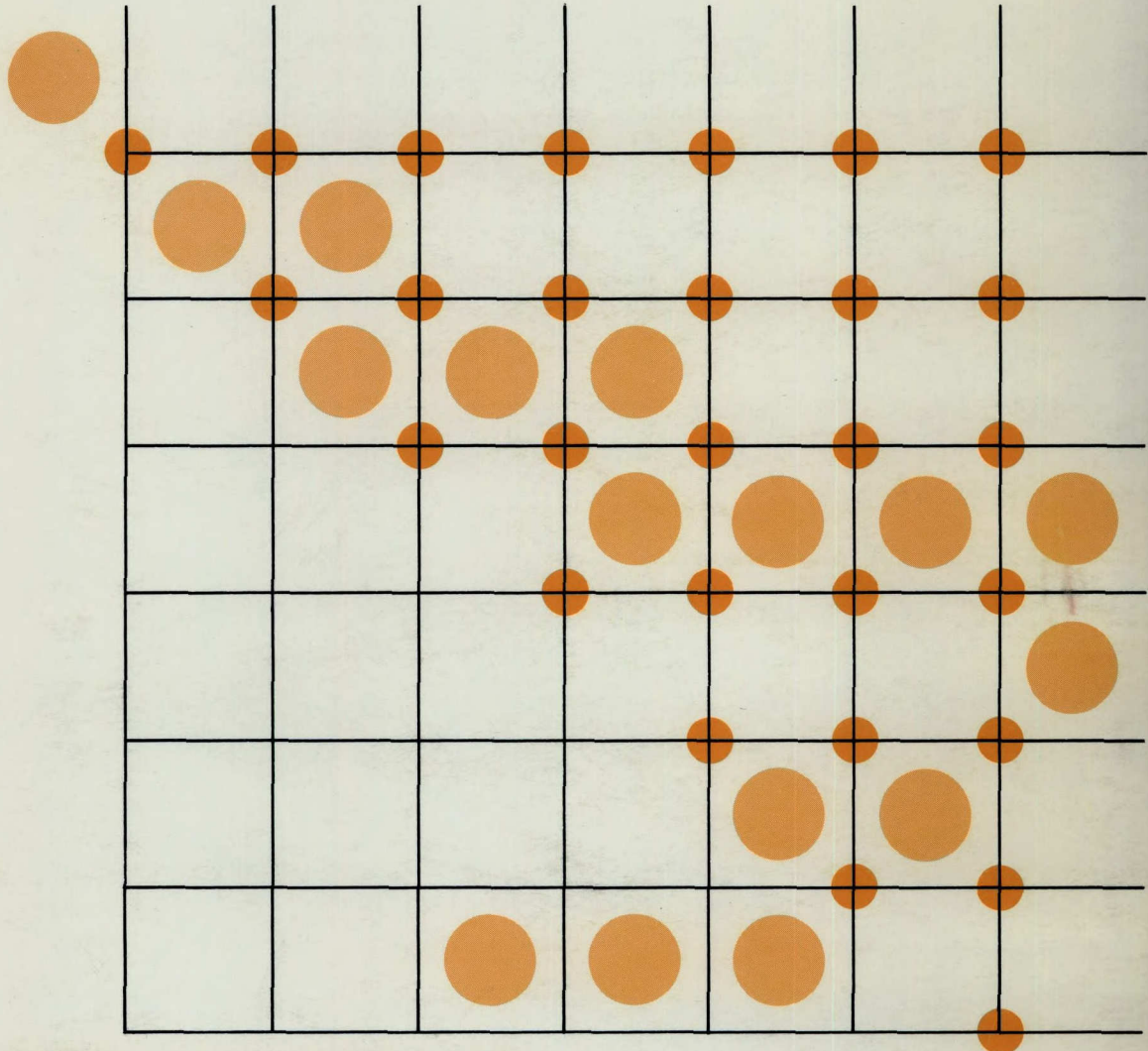


APPLICATION OF WAVE DIGITAL
FILTER STRUCTURES TO SUB-BAND
VOICE CODER DESIGN

FINAL REPORT
APRIL 1986

DSS Contract No. OST85-00192



IC

LKC
P
91
.C654
M42
1986

Department of Electrical Engineering
The University of Manitoba
Winnipeg, Manitoba, Canada
R3T 2N2

Government of Canada Gouvernement du Canada

Department of Communications
Ministère des Communications

DOC CONTRACTOR REPORT

DOC-CR-RI-86-001

DEPARTMENT OF COMMUNICATIONS - OTTAWA - CANADA
MINISTÈRE DES COMMUNICATIONS - OTTAWA - CANADA

TITLE: Application of Wave Digital Filter Structures to Sub-Band Voice Coder Design

AUTHORS: Dr. G.O. Martens, Mark Jarmasz & Victor Shkawrytko
AUTEUR(S):

ISSUED BY CONTRACTOR AS REPORT NO:

PREPARED BY: Dept. of Electrical Engineering
PREPARE PAR: The University of Manitoba
Winnipeg, Manitoba, R3T 2N2

Industry Canada
Library - Queen
AOUT 16 2012
AUG
Industrie Canada
Bibliothèque - Queen

DEPARTMENT OF SUPPLY AND SERVICES CONTRACT NO: 6ST85-00192

DEPARTMENT OF COMMUNICATIONS REQUISITION NUMBER: 36001-05-0082

DOC SCIENTIFIC AUTHORITY: Dr. Mike Sablatash

CLASSIFICATION: Releasable

COMMUNICATIONS CANADA
CRC
AOUT 15 1986
LIBRARY - BIBLIOTHÈQUE

This report presents the views of the author(s). Publication of this report does not constitute DOC approval of the report's findings or conclusions.

DATE: April 1986

RECEIVED
MAY 19 1966

9)
C1658
M 3309
1966

ORIGINAL DOCUMENTS REVIEW AND PUBLICATION RECORD

SECTOR DDGRI	BRANCH DIP	DATE August 1986
-----------------	---------------	---------------------

PURPOSE This form is for use during review of the DOC-CR contractor reports. It is designed to: record decisions for classification, record reasons for classification and cautionary marking provide for indexing requirements.

INSTRUCTIONS * 1 copy of the completed form must accompany the contractor report package submitted to the CRC Library.

* Complete the following items as applicable.

1. DOC-CR NO. DOC-CR-RI-86-001	2. DSS CONTRACT NO. 36001-05-0082
3. TITLE: Application of wave digital filter structures to sub-band voice coder design	4. DATE April 1986
5. CONTRACTOR University of Manitoba	
6. SCIENTIFIC AUTHORITY Dr. Mike Sablatash	7. LOCATION DIP/CRC
8. TEL. NO. 998-2061	
9. CONTRACTOR REPORT CLASSIFICATION: RELEASABLE <input checked="" type="checkbox"/> CONDITIONALLY RELEASABLE <input type="checkbox"/> NON-RELEASABLE <input type="checkbox"/>	

* REASONS FOR CLASSIFICATION:

10. NO. OF COPIES SUBMITTED TO LIBRARY:

EXECUTIVE SUMMARY FINAL REPORT 5 copies

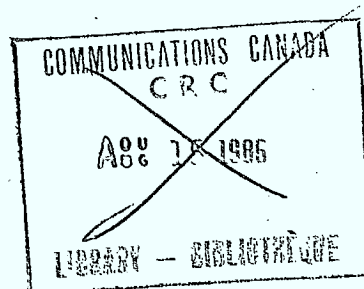
.....
Scientific Authority's Signature

.....
Date

This form is not official therefore it is not signed.

TABLE OF CONTENTS

Acknowledgement	ii
Abstract	1
 <u>SECTION</u>	
I. Introduction	1
II. Polyphase FIR and IIR Structures for Decimators and Interpolators with Integer Changes in Sampling Rate	5
III. Polyphase Structures for DFT and QMF Filter Banks	10
IV. Design of Finite and Infinite Response Filters for a QMF Bank	16
V. Wave Digital Filters for QMF Banks	21
VI. Wave Digital Four-Band Filter Bank Design	29
VII. Real-Time Evaluation System Configuration	37
VIII. Real-Time Implementation of Wave Digital QMF Banks	41
Two-Band WD Design from Section V	41
Four-Band WD Design from Section VI	44
IX. Conclusions	46
References	47
 <u>APPENDIX</u>	
A Two-Band Wave Digital Filter Bank	50
B Four-Band Wave Digital Filter Bank	55



(ii)

ACKNOWLEDGEMENT

The authors acknowledge the interest and encouragement of Dr. Mike Sablatash for the work described in this report.

ABSTRACT

In this report we address the problem of designing an efficient digital filter bank whose function is to split a broad-band signal (usually speech) into sub-bands. These sub-bands are subsequently individually encoded using coders that are optimized to the statistics of the input signal, i.e., preferential coding weight is given to the spectrum components that are perceptually most significant. The resulting coder/decoder arrangement output signal is, for a given bit-rate, subjectively better than an equivalent broad-band waveform encoded signal.

Wave digital (WD) filters are known to be highly efficient structures with a number of superior properties. The fact that these structures inherently provide two output signals makes them particularly suitable for sub-band coders. It is shown that a bireciprocal, symmetric WD lattice structure is equivalent to a critically sampled, uniform, two-band discrete Fourier transform (DFT) filter bank which, with properly selected polyphase filters, is referred to as the quadrature mirror filter (QMF) bank. This equivalence is established to emphasize the common conceptual framework of the filter bank designs.

In addition, we present designs of two- and four-band WD filter banks for sub-band coders. Since these structures, in a back-to-back arrangement, inherently provide an allpass response irrespective of the values of coefficients [24], we direct our efforts toward minimization of the overall group delay distortion. We show that comparisons with existing designs [21], [23] are overwhelmingly in favour of the WD designs. In fact, the digital filter banks in the existing designs account for most of the computational load, whereas the reverse is true when WD structures are used. Also, important details concerning real-time implementation of the WD designs on the MC68000 microprocessor are given.

I. INTRODUCTION

Man-machine communication by voice has been and continues to be an area of great interest and research activity [1]. Voice Input/output is considered to be an integral component of the proposed fifth generation computers of the near future [2]. It is difficult to imagine a computer of the future which could not interact with man in the most natural mode of human communication: voice.

The area of voice output is a continually expanding field, mostly due to the growth of digital technology. Higher speed microprocessors with larger memory capabilities, reduced instruction set processors with co-processing capabilities and specialty high speed processors with custom architectures have allowed the implementation of complex computing algorithms in efficient, often highly parallel and comparatively inexpensive ways. An especially great impact has been made by the specialty digital signal processors such as the Bell Labs DSP, the NEC μ PD7720, and the Texas Instruments TMS320-10.

With the advent of faster processing hardware, more complex speech processing algorithms are being exploited to reduce the transmission and storage requirements of speech. As well, higher fidelity speech and music transmission is being achieved at reduced bit-rates through the application of these more complex algorithms.

The requirement for wider-band speech transmission through the telephone and integrated services digital networks of the future was recognized by CCITT in the early 1980s, and resulted in the establishment of standards for voice transmission of bandwidths from 5 to 7 kHz [3].

It has also been suggested that high quality voice output for Voice I/O terminal applications in the home environment (such as a Voice I/O Telidon terminal) would become a requirement for general acceptability of the service [4]. Furthermore, as semiconductor memory prices continue to drop, and with the installation of optical fibre communication networks, the need for minimal bandwidth (or bit-rate) encoding schemes is to some extent relaxed.

Of the wider-band speech coding approaches which have been popularized throughout the 70s and early 80s, sub-band coding has been recognized as an efficient, high quality coding scheme of moderate complexity [5], [6].

The objective of any speech coding scheme is to maximize output speech quality while minimizing the coder output bit-rate. Most typically, speech quality is degraded in a coding system due to some perceived coding distortion. This distortion is a direct result of the coding process and in order to minimize the perception of this distortion several strategies can be employed. These include exploitation of the auditory system through aural noise masking, the relative phase insensitivity of the ear and the variant frequency sensitivity of the ear [7]. Sub-band coding employs the noise masking principle to maintain high quality speech.

When a signal is quantized, the quantizer introduces errors into the signal since only discrete amplitude values are assigned to the input signal samples. This quantization error manifests itself as a flat noise spectrum superimposed upon the signal spectrum and extends evenly across the bandwidth of the signal. The perception of

this added noise can be greatly decreased by masking the noise with speech spectral energy. This can be achieved by shaping the noise spectrum to match (on average) the shape of the input speech spectrum.

Coding speech in sub-bands achieves such control over the noise spectrum, although crudely. Speech is partitioned into several sub-bands and, by encoding each band with a particular number of bits, one can vary the amount of quantizing noise in each sub-band. This effectively shapes the noise spectrum. This approach also exploits the variant frequency sensitivity of the ear to some extent, through the judicious selection of sub-band widths and centre frequencies.

The main design considerations in sub-band coder design are [8]:

1. the selection of the number of sub-bands and their bandwidths,
2. the implementation of the band-pass filters,
3. selection of the type of band encoders and the number of bits per band.

Various implementations of the above three coder components have been described in the literature [9-13]. Representative of the most common sub-band coder implementations are the split-band ADPCM coder of Johnston and Crochiere [12] and the four and five-band coders of Crochiere et al [13]; the latter being a tree-type extension of the former structure. In both cases, tree structured band-splitting filters define the sub-band bandwidths and the bands are encoded using ADPCM.

The purpose of our research is to design and implement digital filters that perform the band splitting operation. More specifically, we choose the class of wave digital (WD) filters which are known to possess superior properties. In Sections II and III we present the theoretical framework for designing multirate digital filters for sub-band coders. In Section II we derive the finite (FIR) and infinite (IIR) impulse response polyphase structures for decimators and interpolators with integer changes in the sampling rate. These structures provide the main building blocks of sub-band coders and are responsible for most of the computational load. Hence, economical designs of band splitting filters is of great importance. In Section III the polyphase structures are extended to the discrete Fourier transform (DFT) and quadrature mirror filter (QMF) banks. It is shown there that the QMF bank can be simply derived from the 2-channel DFT filter bank by selecting an appropriate synthesis filter. This derivation is conceptually simpler than the original one given by Estaban and Galand [20].

In Section IV specific FIR [21] and IIR [23] filter designs for a QMF bank are presented. The FIR design subsequently serves as the benchmark design. In Section V, we present the results from a very recent publication [24] in which is derived an arrangement of a 2-port WD filter and its transpose such that the overall filtering

effect is that of an allpass. If the WD filter is chosen to be the symmetric lattice structure with birciprocal characteristic function, then the above arrangement becomes a QMF bank. This observation is not explicitly stated in [24]. Also in Section V, we present an alternative design to the one given in [24] that simultaneously minimizes the overall group delay distortion and satisfies the FIR benchmark specifications. The reduction in the overall group delay distortion is even more pronounced in the 4-channel WD filter bank design, which is presented in Section VI.

In Sections VII and VIII, we describe the real-time implementation of the WD filter bank designs on the Motorola MC68000 microprocessor. The fact that a general purpose microprocessor, such as the MC68000, is sufficient for the real-time implementation of the designs from Sections V and VI demonstrates the remarkable efficiency of the WD approach. Real-time operation of the WD filter algorithms is achieved without the need of a special signal processing architecture or a hardware multiplier. Other approaches requiring longer coefficient word-lengths and/or numerous multiplications and storage locations do not allow the use of a general purpose microprocessor.

The first part of Section VII contains a description of the tools used to implement the 2- and 4-band WD filter banks on the microprocessor. Both hardware and software aspects of the development/evaluation system are described. In Section VIII, the implementation details for the two WD designs are presented. Also included is the comparison of the computational requirements of each design. Based on an obvious extension of the two WD designs, a relationship is given which relates the minimum microprocessor clock rate required to the number of bands in the filter bank and the desired sampling rate. This provides an estimate of the computational requirements of a WD design with an arbitrary number of bands. Also in Section VIII, a comparison of the 2-band WD filter bank operating on the MC68000 is made with the FIR benchmark filter from Section IV operating on two other signal processors. The MC68000, due to its general purpose architecture, requires a much higher clock-rate in order to be comparable to the specialty signal processors, but otherwise compares favourably, primarily due to the efficiency of the WD filtering algorithm.

II. POLYPHASE FIR AND IIR STRUCTURES FOR DECIMATORS AND INTERPOLATORS WITH INTEGER CHANGES IN SAMPLING RATE

One of the main steps in the design of sub-band coders is the design of efficient decimators and interpolators. One such design is based on the concept of the polyphase structure [15]. To derive this structure, consider the block diagram of a decimator in Fig. 1.

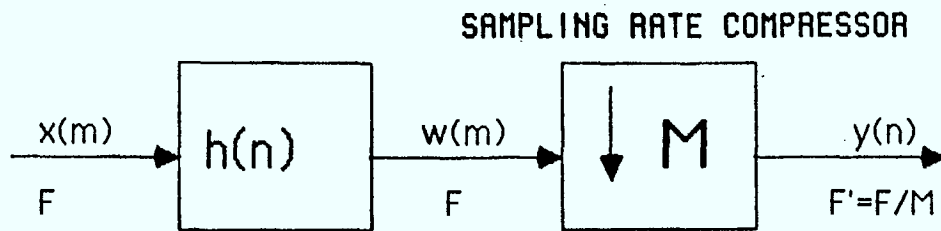


Fig. 1 Block diagram of a decimator with an integer decimation factor M.

It can be shown [16, Sec. 2.3.2] that in Fig. 1 we have

$$y(n) = w(Mn) = \sum_{k=-\infty}^{\infty} h(k) x(Mn - k). \quad (1)$$

To derive the polyphase structure, the summation in (1) is split into two summations by letting [16, Sec. 3.3.2]

$$k = rM + \rho, \quad \rho = 0(1)M-1, \quad r = 0, \pm 1, \pm 2, \dots \quad (2)$$

Substituting (2) into (1) yields

$$y(n) = \sum_{\rho=0}^{M-1} \sum_{r=-\infty}^{\infty} h(rM + \rho) x((n - r)M - \rho). \quad (3)$$

Next, we define

$$p_{\rho}(r) = h(rM + \rho), \quad \rho = 0(1)M - 1 \quad (4a)$$

$$x_{\rho}(m) = x(mM - \rho) \quad (4b)$$

and (3) can be rewritten as

$$y(n) = \sum_{\rho=0}^{M-1} \sum_{r=-\infty}^{\infty} p_{\rho}(r) x_{\rho}(n-r) = \sum_{\rho=0}^{M-1} p_{\rho}(n) * x_{\rho}(n) \quad (5)$$

where $*$ denotes discrete convolution. Thus, the output $y(n)$ is a sum of outputs from M linear time-invariant filters with impulse responses given by (4a). Also, from (4b) and (5) it is seen that each linear filter operates on a different input sequence. This fact suggests a structure with a parallel set of branches, with each branch computing one discrete convolution. Such a structure is derived in [15] and is shown in Fig. 2.

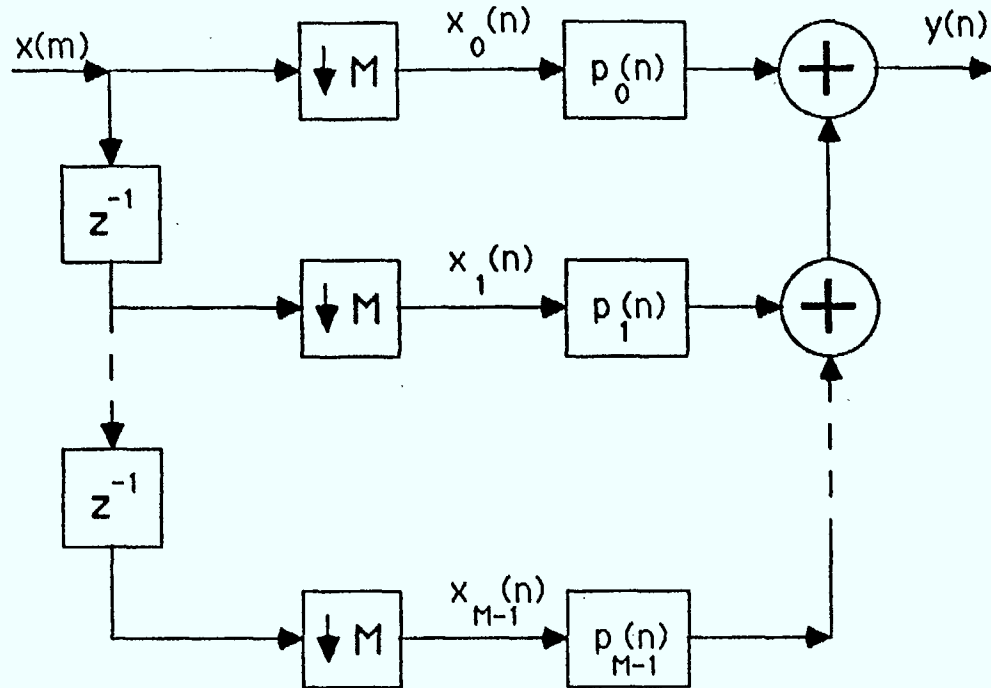


Fig. 2 Polyphase structure for an M-to-1 decimator.

In Fig. 2, the delays together with the sampling rate compressors ensure that each filter receives a different input sequence $x_p(n)$ at the rate F/M . An equivalent structure that uses a commutator switch to implement the same function is shown in Fig. 3. The commutator rotates counterclockwise and distributes M samples of the input sequence, $x(n)$, among the M polyphase branches. Next, the polyphase filter outputs are computed and summed to give the output sample of $y(n)$. If the polyphase structure is to be implemented in software, then the commutator model is much more practical.

An equivalent polyphase structure based on a clockwise commutator model can be derived by replacing p with $-p$ in the above derivation. In this case, the

polyphase filters are defined by

$$p_{\rho}(r) = h(rM - \rho) \quad , \quad \rho = 0(1)M - 1 \quad (6)$$

and the input sequences on which they operate are given by

$$x_{\rho}(n) = x(nM + \rho) \quad , \quad \rho = 0(1)M - 1 \quad (7)$$

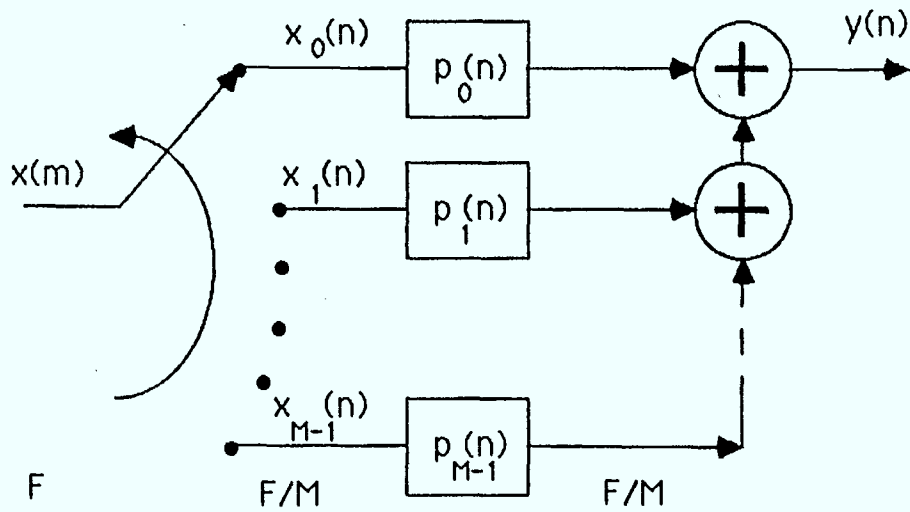


Fig. 3 Counter-clockwise commutator model for an M-to-1 decimator.

The set of filters $p_{\rho}(r)$ in (4a) make up all the samples of $h(n)$. It can be shown [16, Sec. 3.4.1] that the z-transform of $h(n)$ can be expressed as a function of the z-transforms of the polyphase filters $p_{\rho}(r)$, i.e.

$$H(z) = \sum_{\rho=0}^{M-1} z^{-\rho} P_{\rho}(z^M). \quad (8)$$

Conversely, since $p_{\rho}(r)$ is a sampled and delayed version of $h(n)$, it can be shown [16, Sec. 4.2.1] that

$$P_{\rho}(z) = \frac{1}{M} \sum_{m=0}^{M-1} e^{-j2\pi m \rho / M} z^{\rho / M} H(e^{-j2\pi m / M} z^{1/M}) \quad (9)$$

For decimators that use IIR filters, it is desirable from the implementation point of view that $P_\rho(z)$ have the form of a ratio of polynomials in z , i.e.

$$P_\rho(z) = \frac{N_\rho(z)}{D_\rho(z)}, \quad \rho = 0(1)M-1. \quad (10)$$

This form imposes a constraint on the $H(z)$ in (9), i.e., the denominator polynomial of $H(z)$ must be a polynomial in z^M (with a possible factor of z^{M-1}).

The frequency response of each polyphase filter can be obtained from (9) by letting $z = e^{j\omega}$, which yields

$$P_\rho(e^{j\omega}) = \frac{1}{M} \sum_{m=0}^{M-1} e^{j(\omega-2\pi m)\rho/M} H(e^{j(\omega-2\pi m)/M}). \quad (11)$$

Ideally, the frequency response of the decimation filter $h(m)$ in Fig. 1 (this filter is really an anti-aliasing filter) is given by the ideal lowpass characteristic

$$\tilde{H}(e^{j\omega}) = \begin{cases} 1, & 0 \leq \omega \leq \pi/M \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

Substituting (12) into (11) yields the ideal characteristic for the polyphase filters, i.e.,

$$\tilde{P}(e^{j\hat{\omega}}) = \frac{1}{M} e^{j\hat{\omega}\rho/M}, \quad \rho = 0(1)M-1 \quad (13)$$

where $\hat{\omega} = \omega M$ denotes the frequency at the output sampling rate. Thus, the polyphase filters should approximate allpass functions, with each value of ρ , $\rho = 0(1)M-1$, corresponding to a different phase shift (constant group delay). These phase shifts associated with each branch of the network account for the term *polyphase network*.

As stated earlier, the desirable form in (10) imposes a constraint on the denominator polynomial of $H(z)$. An allowable transfer function $H(z)$ can be expressed as

$$H(z) = \frac{\sum_{r=0}^{N-1} b_r z^{-r}}{1 - \sum_{r=1}^R a_r z^{-rM}} = \frac{N(z)}{D(z)} \quad (14)$$

where $D(z)$ is a polynomial in z^{-M} . An algorithm that finds an optimum set of coefficients for the above transfer function is presented in [17], [18]. The authors claim that these filters are more efficient in terms of multiplications per second than linear phase FIR filters for transition band values of less than 0.02 (with the sampling frequency normalized to 1).

An interpolator realized as a polyphase structure can be derived from Fig. 3 by applying the operation of transposition [16], [19]. The resulting network is shown in Fig. 4, where, for each input sample $x(m)$, L output samples are computed (L is the interpolation factor). The polyphase filters are the same as in (4a). Note that in both Fig. 2 and Fig. 3 the polyphase filters are operated at the low sampling rate of F/M and F/L for decimators and interpolators, respectively. This results in substantial computational savings and makes the polyphase structure the best choice for many multirate applications.

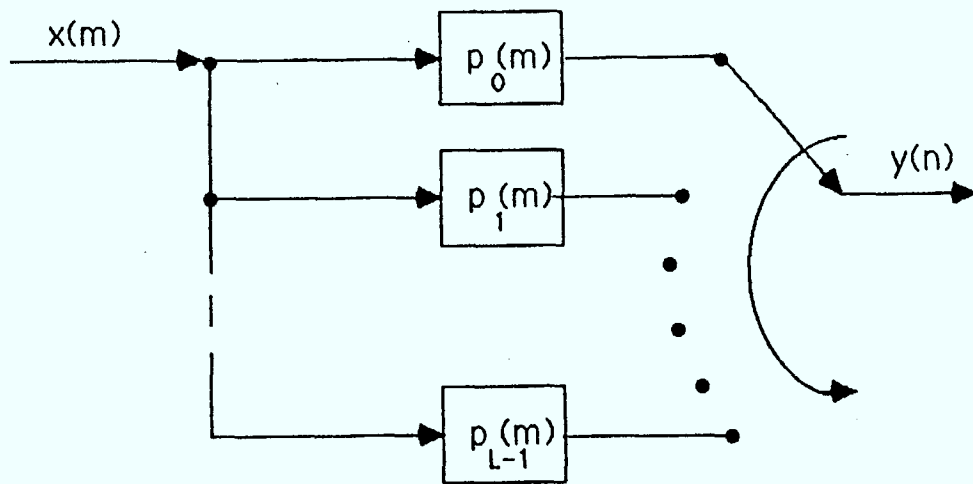


Fig. 4 Counter-clockwise commutator model for a 1-to-L interpolator.

III. POLYPHASE STRUCTURES FOR DFT AND QMF FILTER BANKS

One of the most important classes of filter banks is the discrete Fourier transform (DFT) filter bank. In addition to providing a unique representation of signals in terms of the eigenfunctions of linear time-invariant systems (i.e. complex exponentials), the DFT can be computed very efficiently via the Fast Fourier Transform (FFT) algorithm [19]. In this section, we will show that the DFT filter bank for two bands reduces to the QMF bank [20] by letting the synthesis polyphase filter be the same as the analysis filter, and by delaying the output sequence with respect to the input sequence by one sample. In the derivation of the DFT filter bank we use the counterclockwise commutator model for the decimator on the analysis side and the clockwise model for the interpolator on the synthesis side. This choice is opposite to the one in [16, Sec. 7.2.3], and is made to ease the transition from the DFT to the QMF bank. This renders an independent derivation of the QMF bank, as is done in [16, Sec. 7.7.1], unnecessary.

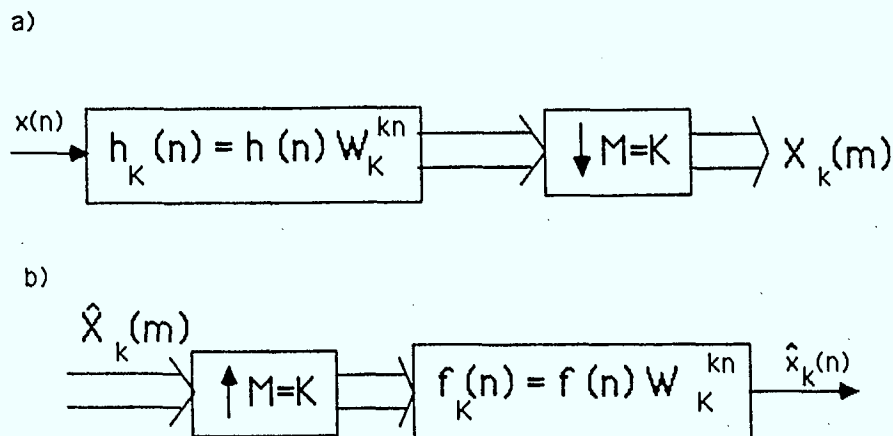


Fig. 5 Integer-band complex bandpass filter model of the k^{th} channel of the DFT filter bank: a) analyzer and b) synthesizer.

As a starting point, we use the complex bandpass filter model for the k^{th} channel of the critically sampled (the decimation factor M is equal to the number of channels K) and uniform (all filters are frequency translations of the baseband filter) DFT filter bank, as shown in Fig. 5 [16, Sec.7.2.2]. In Fig. 5, the double line denotes complex quantities and $W_K = e^{j2\pi/K}$. The derivation that follows is essentially the same as in Section II. First, the unit sample responses for the polyphase branches of the bandpass filters $h_k(n)$ are defined as

$$p_{\rho,k}(m) = h_k(mM + \rho) \quad , \quad \rho = 0(1)M-1 \quad (15)$$

where, from Fig. 5,

$$h_k(n) = h(n) W_M^{kn} \quad . \quad (16)$$

Substituting (16) into (15) yields

$$p_{\rho,k}(m) = h(mM + \rho) W_M^{k\rho} = p_{\rho}(m) W_M^{k\rho} \quad . \quad (17)$$

It is seen from (17) that the form of the bandpass filters is separable into a product, one term of which is independent of the filter bank channel index k . The branch input signals are defined as

$$x_{\rho}(m) = x(mM - \rho) \quad , \quad \rho = 0(1)M-1 \quad . \quad (18)$$

Applying (1) to Fig. 5a gives

$$X_k(m) = \sum_{n=-\infty}^{\infty} h(n) W_M^{kn} x(mM - n) \quad . \quad (19)$$

As in Section II, we split the above summation into two by letting

$$n = rM + \rho \quad , \quad \rho = 0(1)M-1 \quad (20)$$

which, when substituted into (19), results in

$$\begin{aligned} X_k(m) &= \sum_{\rho=0}^{M-1} \sum_{r=-\infty}^{\infty} h(rM + \rho) W_M^{k\rho} x((m-r)M - \rho) \\ &= \sum_{\rho=0}^{M-1} W_M^{k\rho} \left\{ p_{\rho}(m) * x_{\rho}(m) \right\} \quad , \quad k = 0(1)M-1 \quad . \quad (21) \end{aligned}$$

The summation with respect to ρ corresponds to the inverse DFT scaled by M . Note also that the discrete convolution in (21) is independent of k . Hence, all channels can share the same set of polyphase filters which reduces the computation by a factor of M . This reduction is a consequence of the separation property in (17). The resulting polyphase structure for the M channel DFT filter bank analyzer is easily obtained from (21), and is shown in Fig. 6.

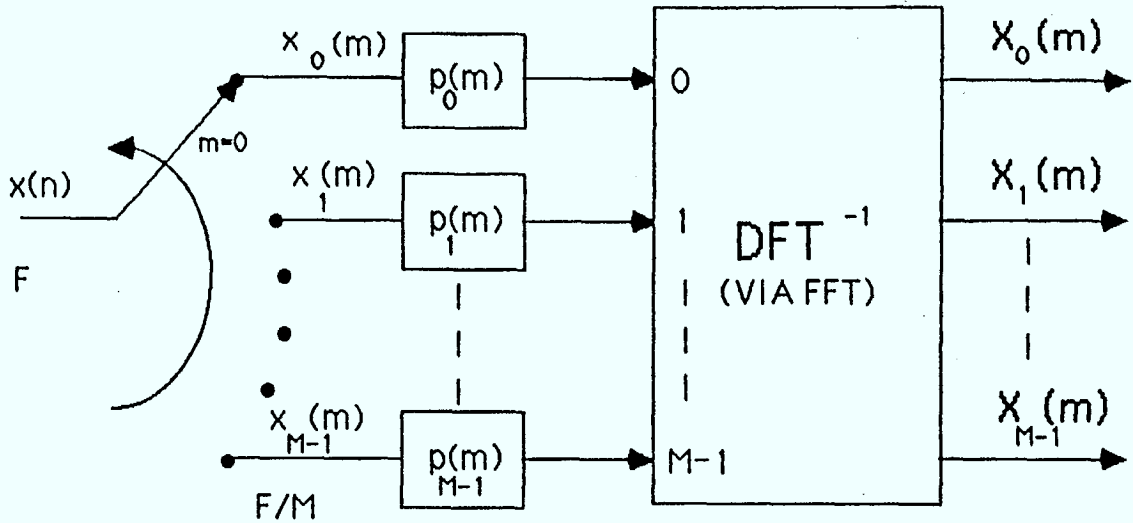


Fig. 6 Analyzer polyphase DFT filter bank.

The polyphase structure for the synthesizer is derived in a similar manner by using the clockwise commutator model for the polyphase structure. In Fig. 5b the unit sample responses of the bandpass filters $f_k(n)$ are defined as

$$\bar{q}_{\rho,k}(m) = f_k(mM - \rho) \quad (22)$$

where

$$f_k(n) = f(n) W_M^{kn} \quad (23)$$

The overbar notation in (22) is used to distinguish clockwise formulation from counterclockwise formulation. Substituting (23) into (22) gives

$$\bar{q}_{\rho,k}(m) = f(mM - \rho) W_M^{-k\rho} = \bar{q}_{\rho}(m) W_M^{-k\rho} \quad (24)$$

It can be shown [16, Sec. 7.2.2] that in Fig. 5b

$$\hat{x}_k(n) = \sum_{m=-\infty}^{\infty} \hat{X}_k(m) f_k(n - mM) \quad (25)$$

and the overall output signal is given by

$$\hat{x}(n) = \frac{1}{M} \sum_{k=0}^{M-1} \hat{x}_k(n) \quad (26)$$

The output sequence is partitioned into M sets by letting

$$n = rM - \rho, \quad \rho = 0(1)M - 1 \quad (27)$$

which yields

$$\begin{aligned} \hat{x}_\rho(r) &= \hat{x}(rM - \rho) \\ &= \frac{1}{M} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} \hat{X}_k(m) f((r-m)M - \rho) W_M^{-k\rho} \end{aligned} \quad (28)$$

Making appropriate substitutions from (24) and interchanging summations yields

$$\hat{x}_\rho(r) = \sum_{m=-\infty}^{\infty} \bar{q}_\rho(r-m) \frac{1}{M} \left\{ \sum_{k=0}^{M-1} \hat{X}_k(m) W_M^{-k\rho} \right\} \quad (29)$$

This form suggests the polyphase synthesis structure shown in Fig. 7, where, for convenience, the factor 1/M is ignored.

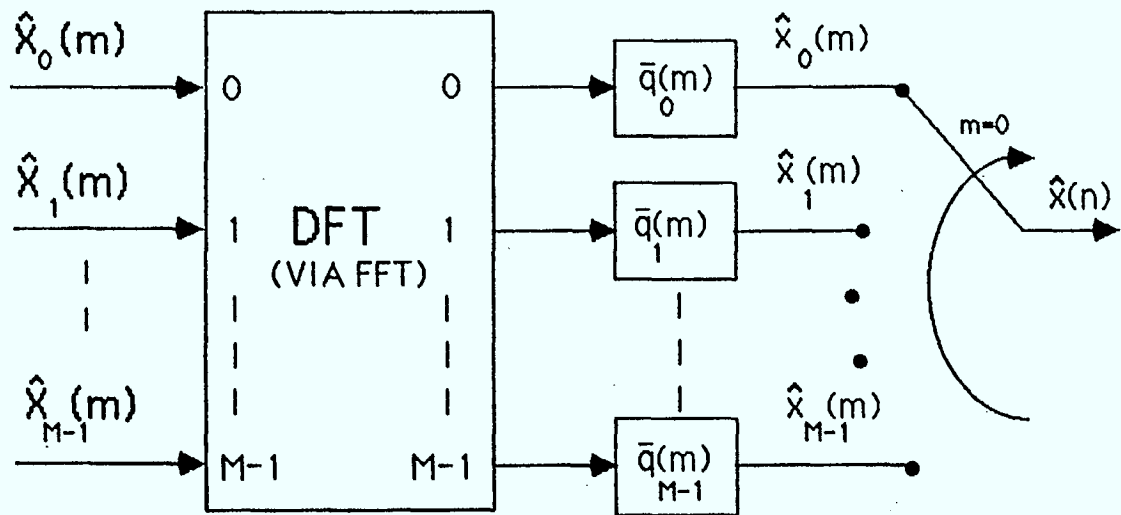


Fig. 7 Synthesizer polyphase DFT filter bank.

If the polyphase analysis/synthesis structure is connected back-to-back, then it can easily be shown by combining (21) and (29) that

$$\hat{x}(r) = \bar{q}_\rho(r) * p_\rho(r) * x_\rho(r) \quad , \quad \rho = 0(1)M-1 \quad . \quad (30)$$

Thus, for exact resynthesis, it is required that

$$\bar{q}_\rho(r) * p_\rho(r) = \begin{cases} 1, & r=0 \\ 0, & \text{otherwise} \end{cases} \quad (31)$$

In order to derive the QMF bank ($M=2$), we propose that it is sufficient to set

$$f(n) = h(n+1) \quad . \quad (32)$$

From (17), (24), and (32) we obtain

$$\begin{aligned} \bar{q}_\rho(m) &= f(mM - \rho) \\ &= h(mM + (1 - \rho)) \\ &= p_{1-\rho}(m) \quad , \quad \rho = 0, 1 \quad . \end{aligned} \quad (33)$$

The resulting QMF bank is shown in Fig. 8. The time displacement between input and output, which is introduced by (32), is realized in Fig. 8 by moving the starting position of the clockwise commutator back one sample. Thus, a one sample delay is inherent in Fig. 8 as a consequence of (32).

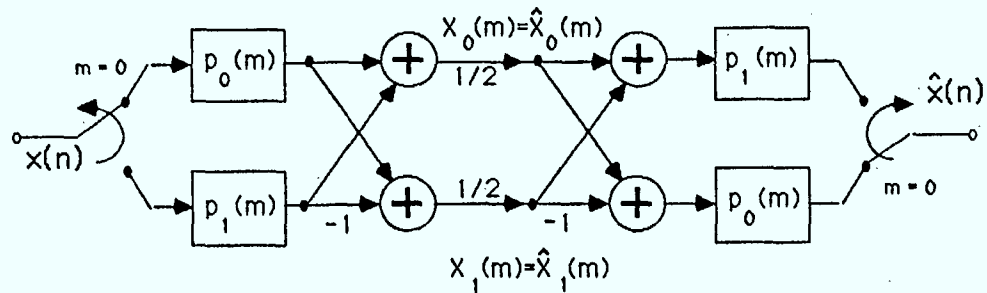


Fig. 8 Polyphase realization of the QMF bank derived from the 2-channel DFT filter bank.

From (16), (23), and (32) we have, for $M=2$

$$h_0(n) = h(n) \Leftrightarrow H_0(e^{j\omega}) = H(e^{j\omega}) \quad (34)$$

$$h_1(n) = (-1)^n h(n) \Leftrightarrow H_1(e^{j\omega}) = H(e^{j(\omega-\pi)})$$

$$f_0(n) = f(n) = h(n+1) \Leftrightarrow F_0(e^{j\omega}) = e^{j\omega} H(e^{j\omega})$$

$$f_1(n) = (-1)^n f(n) = -(-1)^{n+1} h(n+1) \Leftrightarrow F_1(e^{j\omega}) = -e^{j\omega} H(e^{j(\omega-\pi)})$$

Thus, all four filters are derived from a common design characterized by $h(n)$. The term *quadrature mirror* refers to the fact that the pair of filters $h_0(n)$ and $h_1(n)$ in (34) have amplitude responses that are mirror images about the frequency $\omega = \pi/2$, and their phase responses differ by $\pi/2$ radians for $\omega = \pi/2$. Applying (4a) to $h(n)$ yields

$$p_0(m) = h(2m) \quad , \quad p_1(m) = h(2m+1) \quad . \quad (35)$$

Also, from (31) and (33) we have

$$p_1(m) * p_0(m) = u_0(m) \quad \text{or} \quad P_1(e^{j\omega}) P_0(e^{j\omega}) = 1 \quad (36)$$

as the condition for exact resynthesis. Applying (11) to (36) yields the requirement

$$|H^2(e^{j\hat{\omega}}) - H^2(e^{j(\hat{\omega}-\pi)})| = 1 \quad (37)$$

where $\hat{\omega} = \omega/2$. Thus, the lowpass filter $H(z)$ must simultaneously approximate the ideal lowpass characteristic in (12) as well as the condition for exact resynthesis in (37). If an FIR filter is used to implement $H(z)$, then linear phase is possible, but (37) is only approximately true, whereas the WDF approach satisfies (37) exactly, but introduces phase distortion. The correct choice between the two approaches depends on the application.

The above derivation of the QMF bank is much simpler than the one in [16, Sec. 7.7.3], as well as the frequency domain derivation in [16, Sec. 7.7.1]. The distinguishing feature of the QMF bank is that the aliasing due to the decimation in the analysis structure is exactly canceled by the imaging due to interpolation in the synthesis structure. Also, the structure in Fig. 8 can be readily extended to a tree-type arrangement, thus allowing a filter bank with nonuniform bands.

IV. DESIGN OF FINITE AND INFINITE IMPULSE RESPONSE FILTERS FOR A QMF BANK

If linear phase response (i.e. constant group delay) is desired, then a symmetric FIR filter is the appropriate choice [20]. The symmetry property implies that (for even N)

$$h(n) = h(N-1-n) \quad , \quad n = 0(1)N-1 \quad (38)$$

It can be shown [19] that the Fourier transform of (38) can be expressed as

$$\begin{aligned} H(e^{j\omega}) &= e^{-j\omega(N-1)/2} \sum_{n=0}^{N/2-1} 2h(n) \cos \left[\omega \left(n - \frac{N-1}{2} \right) \right] \\ &= H_r(\omega) e^{-j\omega(N-1)/2} \end{aligned} \quad (39)$$

where $H_r(\omega)$ is real. The term $e^{-j\omega(N-1)/2}$ contributes a constant group delay of $(N-1)/2$ samples to the response. The exact resynthesis condition in (37) can be restated in terms of $H_r(\omega)$ as

$$H_r^2(\omega) + H_r^2(\omega + \pi) = 1 \quad (40)$$

An FIR filter can only approximate (40) and, consequently, a number of optimization algorithms have been developed for finding sets of coefficients $h(n)$ that simultaneously approximate the ideal lowpass response and the exact resynthesis condition in (40) [12, 13, 21]. An example of one such filter is shown in Fig. 9 (thin line). This is a 32-tap filter ($N=32$) which we will use as a benchmark. The attenuation characteristic, $A(f)$, of this filter satisfies

$$A(f) \geq 38 \text{ dB} \quad \text{for} \quad f/f_s \geq 0.293 \quad (41)$$

where f_s is the sampling frequency. For even-length, symmetric filters it can be readily shown that

$$p_0(m) = p_1(N/2 - 1 - m) \quad (42)$$

i.e., the polyphase impulse responses are time reversed versions of each other. The coefficients for the FIR benchmark filter were obtained from [21] and are given in Table 1. The reconstruction error (40) for the above set of coefficients is shown in Fig. 9b (thin line), and is within $\pm 0.025 \text{ dB}$. This filter bank (see Fig. 8) performs 32 multiplications per sampling interval.

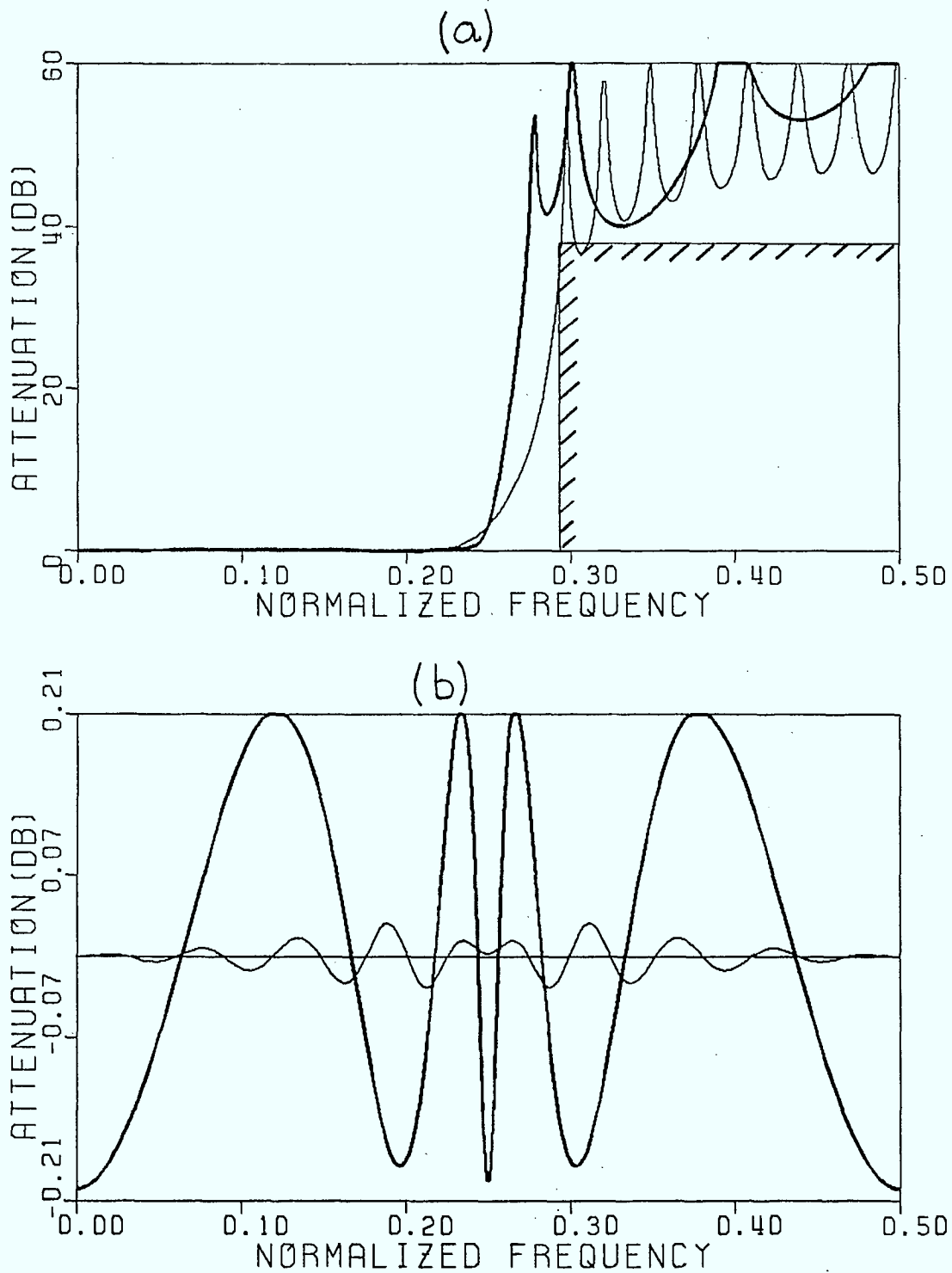


Fig. 9 (a) Lowpass responses of the FIR benchmark (thin line) and IIR (thick line) filters and (b) overall responses given by (37).

Table 1			
FIR Benchmark Filter Coefficients			
$p_0(0) = p_1(15) =$	0.002245139	$p_0(8) = p_1(7) =$	0.4636741
$p_0(1) = p_1(14) =$	-0.001969672	$p_0(9) = p_1(6) =$	-0.09933859
$p_0(2) = p_1(13) =$	0.000842683	$p_0(10) = p_1(5) =$	0.05481213
$p_0(3) = p_1(12) =$	0.00206947	$p_0(11) = p_1(4) =$	-0.0349644
$p_0(4) = p_1(11) =$	-0.007961731	$p_0(12) = p_1(3) =$	0.02270415
$p_0(5) = p_1(10) =$	0.01947218	$p_0(13) = p_1(2) =$	-0.01422899
$p_0(6) = p_1(9) =$	-0.04452423	$p_0(14) = p_1(1) =$	0.008181941
$p_0(7) = p_1(8) =$	0.1329725	$p_0(15) = p_1(0) =$	-0.003971152

If modest phase distortion is acceptable, then the polyphase filters in Fig. 8 can also be realized as IIR filters. In this case, the denominator of the overall transfer function $H(z)$ is a polynomial in z^2 . This approach is described by Barnwell [22]; however, he restricts himself to polyphase filters with equal denominator polynomials, a restriction which, as will be seen, is relaxed in the wave digital approach. Barnwell concluded that the computational savings inherent in IIR filters cannot be fully utilized because the overall group delay distortion reduces the quality of the speech signal to an unacceptable level. He reports that when the variation in the group delay exceeds about 50 samples, the speech signal begins to sound reverberant. Furthermore, in the four and five band designs (the QMF bank is cascaded in a tree structure), the IIR filters could only be used in the outer two bands. Because of this, the resulting computational savings are very modest as compared to an equivalent FIR design.

Another approach is to realize the filters in (34) directly. The resulting QMF bank is shown in Fig. 10 [20]. In this case, the restriction on the denominator polynomial, due to the polyphase requirement, no longer applies. Also, the single delay between input and output is unnecessary, and from (34) we have

$$F_0(e^{j\omega}) = H(e^{j\omega}) \quad , \quad F_1(e^{j\omega}) = -H(e^{j(\omega-\pi)}) \quad . \quad (43)$$

Some designs, with the common lowpass filter $H(z)$ realized as an IIR filter are presented by Millar [23]. Millar restricts himself to a certain class of factors whose product is a transfer function $H(z)$ with the property

$$\arg [H(e^{j\omega})] = \arg [H(e^{j(\omega-\pi)})] + \pi/2 \quad (44)$$

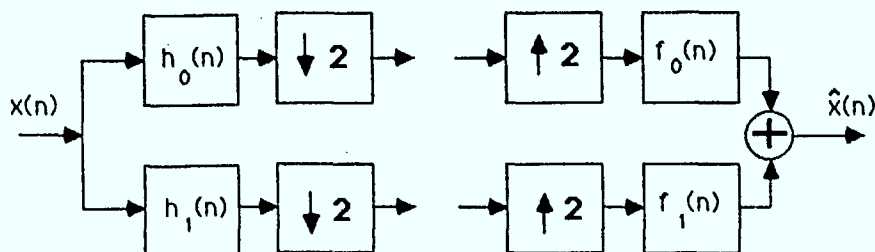


Fig. 10 Two-band analysis/synthesis framework for a QMF design.

i.e., for all frequencies, the phases of signals from filter $H(e^{j\omega})$ are in quadrature with those from filter $H(e^{j(\omega-\pi)})$. This property ensures that the signals from the two channels will be in phase at the recombination node. In Fig. 9 we have plotted the response of one such filter (solid curve) as well as the overall response for the two channel QMF bank. The transfer function for this filter is given by [23]

$$H(z) = \frac{S_c (z+1) (a_0z+1) (z^2+A_1z+1) (z^2+A_2z+1) (z^2+A_3z+1)}{z (z-a_0) (z^2+a_1z+b_1) (z^2+a_2z+b_2) (z^2+a_3z+b_3)} \quad (45)$$

where

$$\begin{array}{llll} a_0 = 0.2, & A_1 = 0.344, & a_1 = -0.065, & b_1 = 0.442 \\ S_c = 0.058146, & A_2 = 1.605, & a_2 = 0.065, & b_2 = 0.442 \\ & A_3 = 0.634, & a_3 = 0.0, & b_3 = 0.871. \end{array}$$

The poles and zeros for this filter are displayed in Fig. 11. The transfer function for the highpass filter $H(z)$ can be obtained from (45) by replacing z with $-z$. The complete system (see Fig. 10) exhibits a single peak group delay distortion of 42 samples, with a minimum delay of 5 samples. It is reported in [23] that single peak delay variations of less than 20 msec (160 samples for a sampling frequency of 8 kHz), and periodic delay variations of less than 10 msec (80 samples) in 500 Hz do not cause discernible distortion of the speech signal. These estimates are somewhat higher than the ones reported in [22], and Millar suggests that this discrepancy is probably due, in part, to the overall system amplitude ripple of the structures used in [22]. The

structure in Fig. 10 requires an average of 22 multiplications per input sample, which is a substantial reduction from the 32 required by an equivalent FIR structure. Also, the complete system requires 36 temporary storage locations as compared to 62 for the FIR system. Hence, for speech processing, the IIR structure seems to offer substantial reduction in the computational requirements. In the next section, we will show how these savings can be increased even further by employing wave digital filters. In addition to these savings, the wave digital approach offers a simple scheme for the suppression of all zero-input parasitic oscillations and low sensitivity to coefficient quantizations.

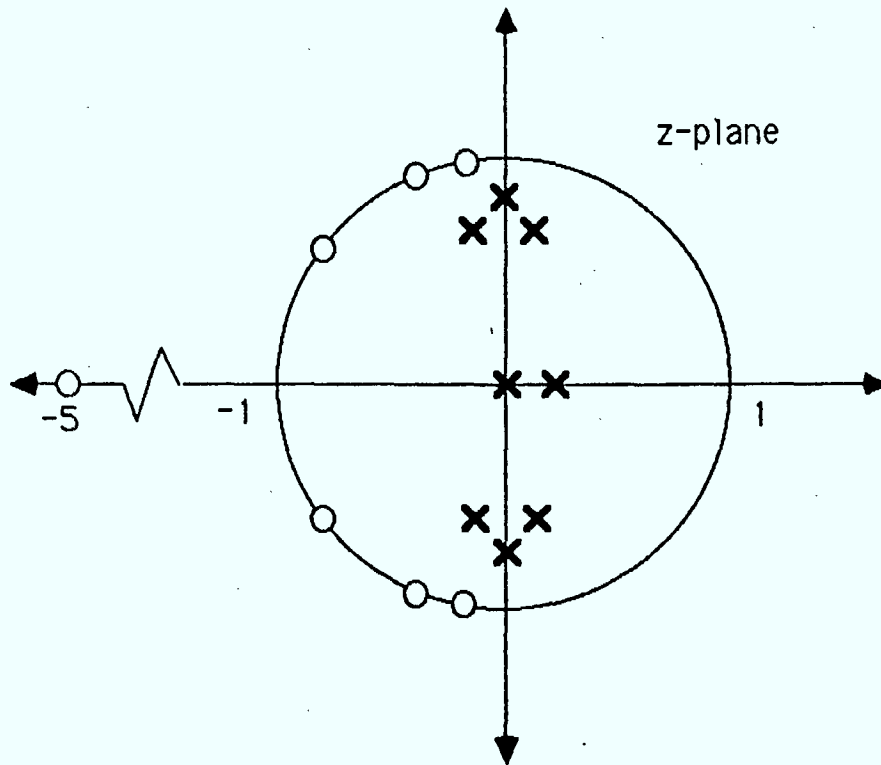


Fig. 11 Pole/zero plot of the filter in (45).

V. WAVE DIGITAL FILTERS FOR QMF BANKS

In a very recent publication, Fettweis et al. [24] derived an arrangement of a two-port wave digital filter (WDF) N and its transpose N^T , such that the overall filtering effect is that of an allpass, and this irrespective of any selectivity requirements of the filter N . This arrangement is shown in Fig. 12a.

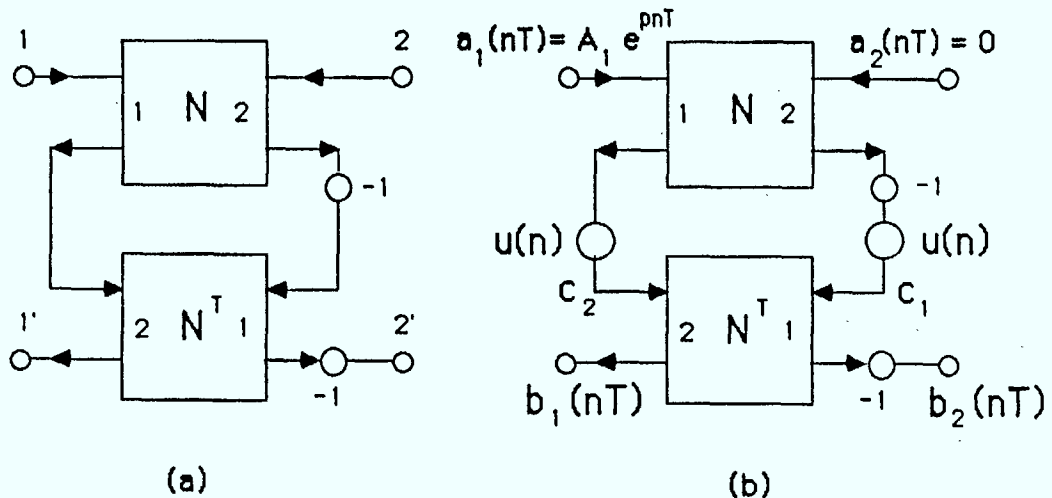


Fig. 12 (a) Arrangement with the overall S given by (48) and (b) with $b_1(nT)$ given by (51).

If the two-port scattering matrix is given by

$$\mathbf{S} = \begin{bmatrix} S_{11}(\psi) & S_{12}(\psi) \\ S_{21}(\psi) & S_{22}(\psi) \end{bmatrix} \quad (46)$$

where

$$\psi = \tanh(pT/2) = \frac{z-1}{z+1}, \quad z = e^{pT} \quad (47)$$

then it is shown in [24] that the two-port scattering matrix for the arrangement in Fig. 12a is given by

$$\mathbf{S} = \begin{bmatrix} \det \mathbf{S} & 0 \\ 0 & \det \mathbf{S} \end{bmatrix}, \quad \det \mathbf{S} = S_{11}(\psi) S_{22}(\psi) - S_{12}(\psi) S_{21}(\psi) \quad (48)$$

It is well known that for N lossless, $\det \mathbf{S}$ is an allpass function [25]. It follows that the net effect of the network in Fig. 12a is the introduction of allpass phase shifts.

Next, Fettweis et al. introduce a multiplier

$$u(n) = [1 + (-1)^n]/2 \doteq \frac{1}{2} (1 + e^{jn\pi}) \quad (49)$$

in both output branches of N , as shown in Fig. 12b. This multiplier effectively reduces the sampling rate at the interconnecting ports by a factor of two. Steady-state analysis shows that if \mathbf{S} is such that

$$S_{21}(\psi) S_{12}\left(\frac{1}{\psi}\right) = \pm S_{11}(\psi) S_{22}\left(\frac{1}{\psi}\right) \quad (50)$$

then, in Fig. 12b,

$$b_1(nT) = \frac{1}{2} \det \mathbf{S} a_1(nT) \quad (51)$$

Again, except for an allpass phase shift, the input sequence $a_1(nT)$ is fully recoverable at the output port #2 of N^T . The condition in (50) is referred to as the bireciprocal property [27].

For real, lossless and symmetric filters, we have

$$S_{11} = S_{22} \quad , \quad S_{12} = S_{21} \quad (52)$$

In this case, the two transfer functions can be expressed in terms of the two allpass reflectances, S_1 and S_2 , i.e.,

$$S_{11} = \frac{1}{2} (S_1 + S_2) \quad , \quad S_{21} = \frac{1}{2} (S_2 - S_1) \quad (53)$$

where the two reflectances correspond to the two canonic lattice impedances of N [25]. The condition in (50) can be restated as

$$S_1(\psi) S_2\left(\frac{1}{\psi}\right) = - S_1\left(\frac{1}{\psi}\right) S_2(\psi) \quad (54)$$

which is satisfied if

$$S_1\left(\frac{1}{\psi}\right) = +/- S_1(\psi) \quad , \quad S_2\left(\frac{1}{\psi}\right) = -/+ S_2(\psi) \quad (55)$$

where either both upper signs or both lower signs hold. According to (47), the replacement $\psi \rightarrow \frac{1}{\psi}$ corresponds to the replacement $z \rightarrow -z$. It follows from (55) that if we choose the upper signs, then S_1 must be even in z and S_2 odd, and vice versa if the other choice is made. From the first choice we have

$$S_1\left(\psi = \frac{z-1}{z+1}\right) = \hat{S}_1(z^2) \quad , \quad S_2\left(\psi = \frac{z-1}{z+1}\right) = z^{-1} \hat{S}_2(z^2) \quad (56)$$

A realization of a symmetric WD lattice filter in terms of the canonic reflectances S_1 and S_2 is shown in Fig. 13 [26].

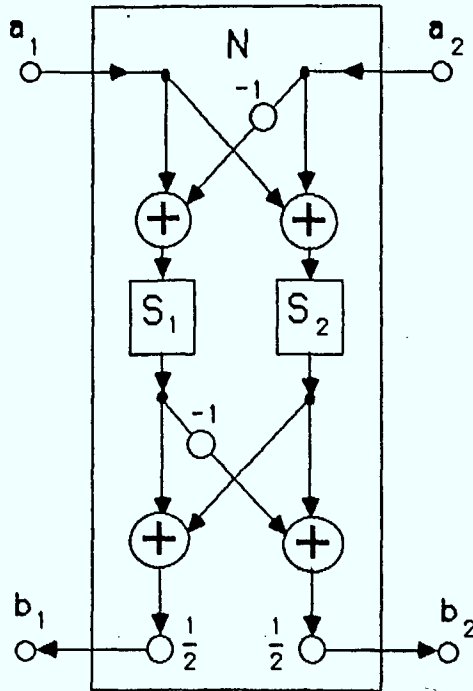


Fig. 13 Wave digital lattice filter.

When the structure in Fig. 13 is connected to its transpose according to Fig. 12b and the substitutions from (56) are made, then the resulting complete arrangement is as shown in Fig. 14.

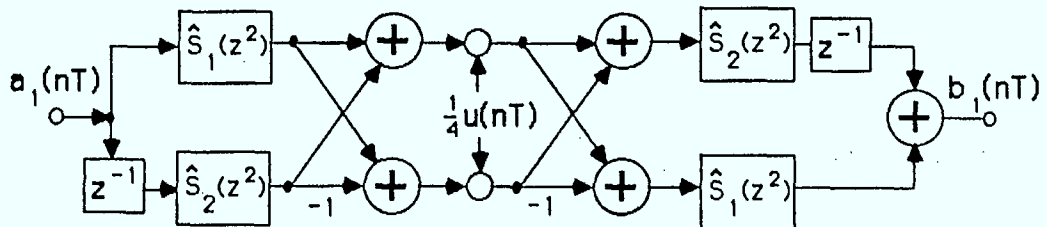


Fig. 14 Arrangement corresponding to Fig. 12b when N is a symmetric WD lattice filter.

The arrangement in Fig. 14 is essentially the same as for the polyphase QMF bank in Fig. 8. In fact, the combination of the single delay and the multiplier $u(nT)$ in Fig. 14 can be realized using a commutator switch (see Figs 2 and 3 and the discussion in Section II). This is also a consequence of the fact that \hat{S}_1 and \hat{S}_2 are functions of z^2 . The resulting structure is shown in Fig. 15, and it is seen that for

$$\hat{S}_1(z) = P_0(z) \quad , \quad \hat{S}_2(z) = P_1(z) \quad (57)$$

this structure is equivalent to the polyphase QMF bank in Fig. 8. This equivalence is not made in [24]. This curious omission is probably due to the authors of [24] wanting to emphasize the general nature of the arrangement in Fig. 12b, and not wishing to commit themselves completely to polyphase structures.*

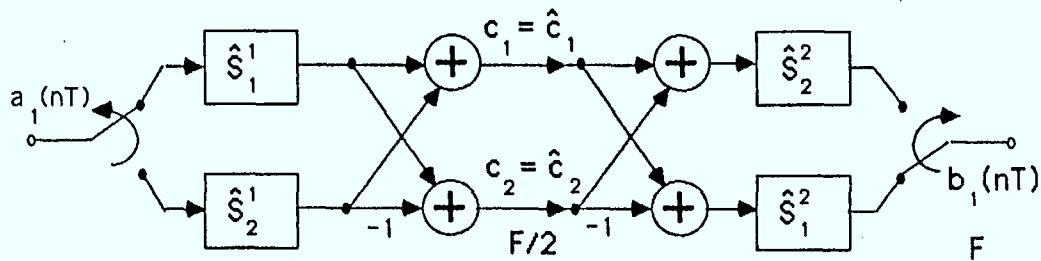


Fig. 15 Wave digital arrangement equivalent to the QMF bank in Fig. 8.

The details of designing WD lattice filters with the bireciprocal property (55) are given in [27, 28, 29]. Here we present the design from [24], as shown in Fig. 16. Note that, in Fig. 16, both allpass sections are operated every $2T$ sec. Hence, the $2T$ delays in Fig. 16 require only one storage location each. Because the decimation factor is two, the above structure is operated within a $2T$ time frame. For the complete system in Fig. 15 the sequence of operations is as follows:

*In the most recent publication by Fettweis, this equivalence is in fact made explicit [36].

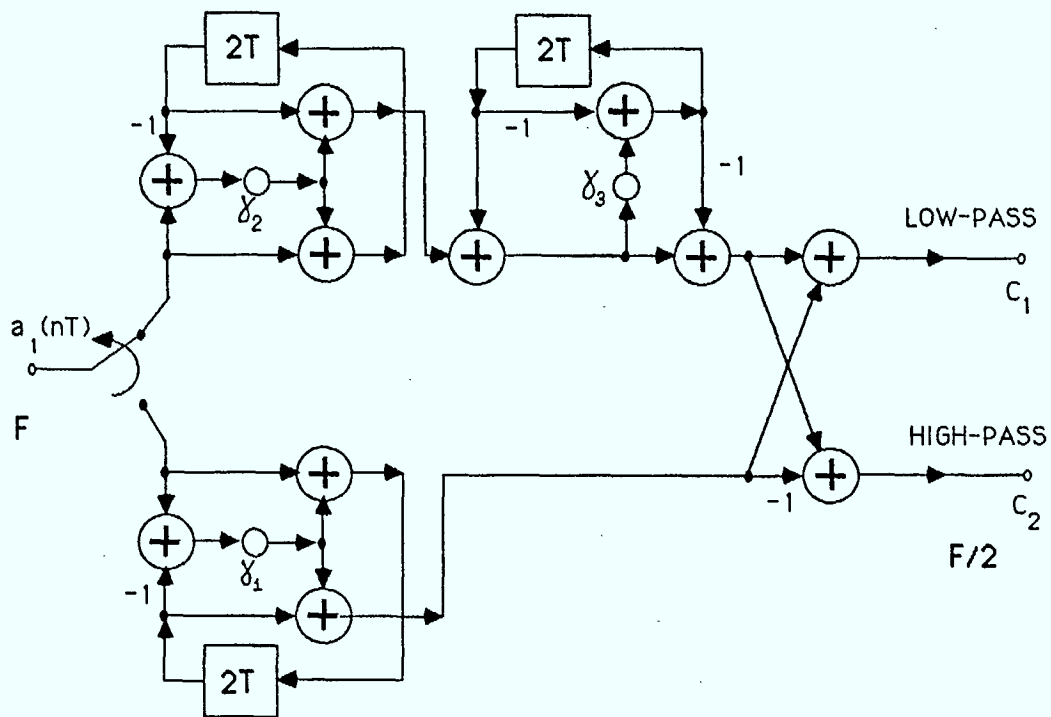


Fig. 16 A 7th order WD lattice filter bank analyzer consisting of only 3 multipliers.

Cycle 0:

Analyzer: evaluate \hat{S}_1^1 with $a_1(nT)$ as input and compute c_1 and c_2 .

Synthesizer: evaluate \hat{S}_1^2 with $(\hat{c}_1 - \hat{c}_2)$ as input and take out $b_1(nT)$.

Cycle 1:

Analyzer: evaluate \hat{S}_2^1 with $a_1((n+1)T)$ as input.

Synthesizer: evaluate \hat{S}_2^2 with $(\hat{c}_1 + \hat{c}_2)$ as input and take out $b_1((n+1)T)$.

The two phase cycle is now repeated for the next two input samples, and so on. Note that in cycle 0 two evaluations of the reflectance $\hat{S}_1(z)$ are required, and since, as shown in Fig. 16, each is a cascade of two first-order allpass sections, a total of 4 allpass sections must be evaluated. On the other hand, during cycle 1 two $\hat{S}_2(z)$ reflectances, each consisting of one allpass section (see Fig. 16), must be evaluated, thus bringing the total to two allpass sections. This discrepancy in the computational loads between the two cycles is undesirable since it leads to under utilization of the signal processor during cycle 1, and cycle 0 imposes a lower sampling frequency than is possible with more even distribution of the overall computational load. We propose a method of circumventing this problem by setting the output switch in Fig. 15 to its upper position, which effectively introduces a one sample delay between the input and output sequences. Now, during each cycle, only one $\hat{S}_1(z)$ and one $\hat{S}_2(z)$ (total of 3 allpass sections) must be evaluated. Thus, a saving of one allpass evaluation for the worst path is incurred. Further details concerning this design are given in Section VIII.

The structure in Fig. 15, together with the coefficient set

$$\gamma_1 = \frac{1}{2} \quad , \quad \gamma_2 = \frac{5}{32} = \frac{1}{8} + \frac{1}{32} \quad , \quad \gamma_3 = \frac{21}{128} = \frac{1}{8} + \frac{1}{32} + \frac{1}{128} \quad (58)$$

satisfies the specifications given by (41). The undecimated lowpass response is shown in Fig. 17a (thin line), and the overall group delay response for the structure in Fig. 15 is given in Fig. 17b (thin line). This design compares remarkably well with the designs in Section IV. From Figs. 15 and 16, it is seen that the structure performs 3 multiplications per input sample and requires 6 temporary storage locations for the overall structure. Moreover, it is reported in [23] that the magnitude distortions become noticeable for the FIR and IIR structures if the coefficients are quantized to less than 12 and 10 bits, respectively. But, the allpass response of the configuration in Fig. 15 (with appropriate substitutions from Fig. 16) is structurally inherent, and is not dependent on coefficient quantizations! This remarkable property in itself is sufficient to make the WD approach the best choice. Also, the three multiplications in Fig. 16 are of a very simple nature (see (58)) and do not require a multiply routine for their implementation. This latter property is due to the low sensitivity of WD filters to parameter quantizations.

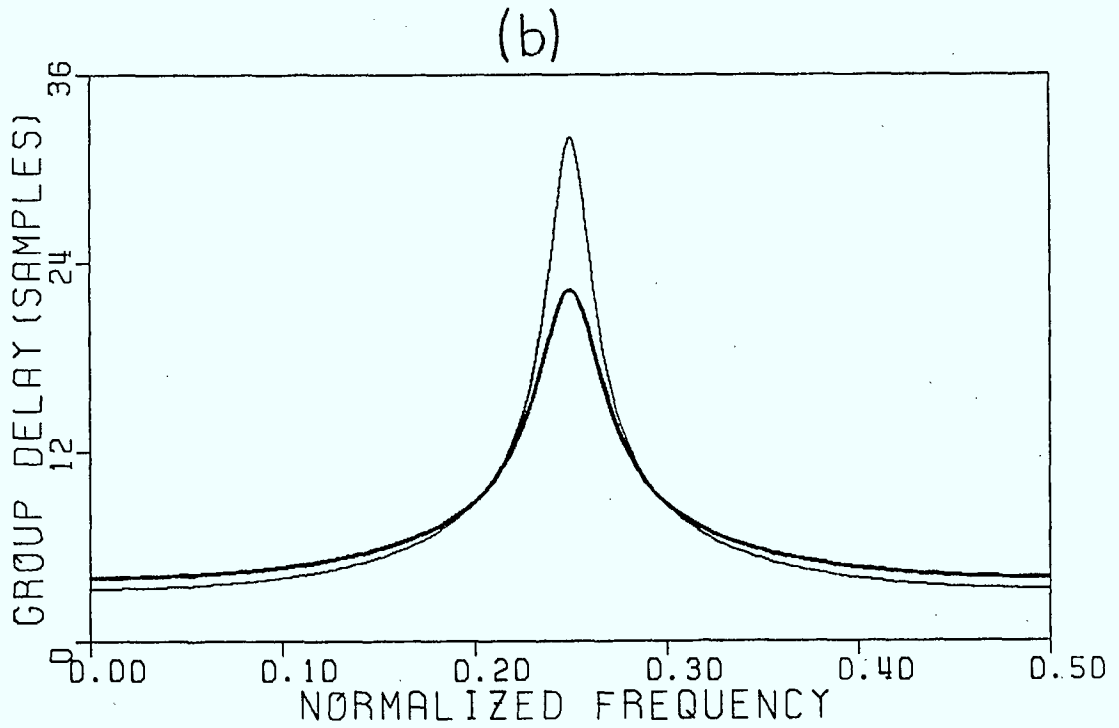
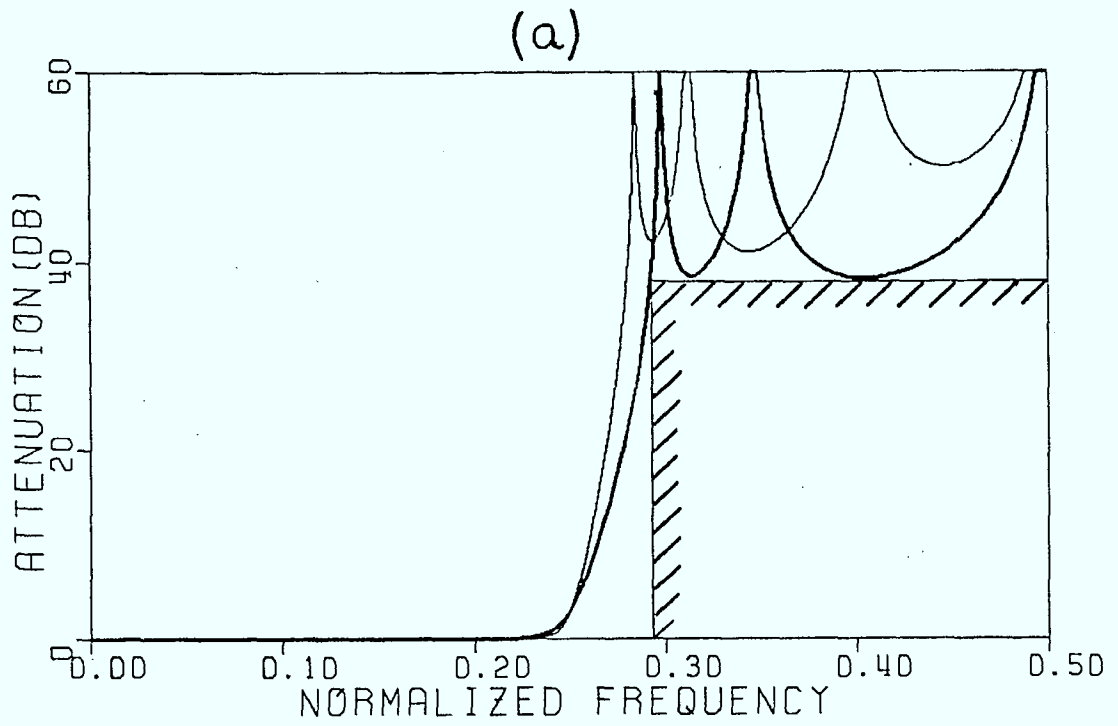


Fig. 17 Wave digital (a) lowpass responses and (b) overall group delay responses for the coefficient sets (58) (thin line) and (64) (thick line).

It is seen from Fig. 17a that the WD filter offers substantial margin with respect to the benchmark specifications. We propose that the available margin in the magnitude response be exchanged for improved overall group delay response. This can be accomplished by supplying an appropriate objective function to a nonlinear optimization algorithm, which then finds a new coefficient set. The objective function is derived as follows. From Fig. 16, we have

$$\hat{S}_2(z^2) = \frac{\gamma_1 z^2 + 1}{z^2 + \gamma_1}, \quad \hat{S}_1(z^2) = \frac{(\gamma_2 z^2 + 1)((1 - \gamma_3)z^2 + 1)}{(z^2 + \gamma_2)(z^2 + (1 - \gamma_3))} \quad (59)$$

The lowpass and highpass transfer functions are given by (see Fig. 14)

$$H_{LP}(z) = \left[\hat{S}_1(z^2) + z^{-1} \hat{S}_2(z^2) \right] \quad \text{and} \quad H_{HP}(z) = \left[\hat{S}_1(z^2) - z^{-1} \hat{S}_2(z^2) \right] \quad (60)$$

respectively. The overall transfer function for the 2-band back-to-back arrangement in Fig. 14, or equivalently Fig. 15, is given by

$$\begin{aligned} H_2(z) &= H_{LP}^2(z) - H_{HP}^2(z) \\ &= 4 z^{-1} \hat{S}_1(z^2) \hat{S}_2(z^2) \\ &= 4 z^{-1} \prod_{i=1}^3 \frac{\hat{\gamma}_i z^2 + 1}{z^2 + \hat{\gamma}_i} \end{aligned} \quad (61)$$

where $\hat{\gamma}_1 = \gamma_1$, $\hat{\gamma}_2 = \gamma_2$, and $\hat{\gamma}_3 = 1 - \gamma_3$. It is seen from (61) that $H_2(z)$ is a product of allpass sections. Thus, the coefficient values do not affect the magnitude response but affect the overall group delay response given by

$$\tau_2(\omega) = 1 + 2 \sum_{i=1}^3 \frac{1 - \hat{\gamma}_i^2}{\hat{\gamma}_i^2 + 2\hat{\gamma}_i \cos(2\omega) + 1} \quad (62)$$

It can be readily shown that, in the range $0 \leq \omega < \pi$, the above function is maximum at $\omega = \pi/2$ and minimum at $\omega = 0$. We define an objective function as the difference between the maximum and minimum values of (62); the result is given by (for the overall system)

$$f(\hat{\gamma}) = 8 \sum_{i=1}^3 \frac{\hat{\gamma}_i}{1 - \hat{\gamma}_i^2} \quad (63)$$

This objective function is minimized subject to the constraint that the magnitude response of $H_{LP}(z)$ satisfy the specification in (41). A new set of quantized coefficients has been found and is given by

$$\gamma_1 = \frac{11}{32} = \frac{1}{4} + \frac{1}{16} + \frac{1}{32}, \quad \gamma_2 = \frac{1}{16}, \quad \gamma_3 = \frac{121}{512} = \frac{1}{4} - \frac{1}{64} + \frac{1}{512} \quad (64)$$

The magnitude and overall group delay responses for the new filter are shown in Fig. 17 (solid line). Note that the lowpass response is within the allowable specification and the group delay nonlinearity, given by (63), has been decreased by 10.5 samples. This is a substantial reduction considering that the multipliers in (64) require only one extra addition for their implementation.

Both sets of coefficients in (58) and (64) yield designs with nonlinear group delays that, for speech signals, fall within the allowable range as discussed in Section IV. The advantage of using the new set becomes more pronounced in the four-band design, which we present next.

VI. WAVE DIGITAL FOUR-BAND FILTER BANK DESIGN

One simple way of deriving a 4-band structure is by extending the 2-band design from Fig. 15 to a two-level tree structure, as shown in Fig. 18. The specifications for the 4-band design are obtained from the 2-band design in (41) by simple extensions, e.g., the lowpass channel stopband corner is $\frac{1}{2}$ the original value and the first stopband corner of the first bandpass channel is $\frac{1}{2}$ the value of the original highpass (mirror image of the lowpass) stopband corner, etc. These extensions stem from the fact that, as will be shown shortly, for the second level of the tree structure we have the mapping $z \rightarrow z^2$. It follows that the critical frequencies for the second level, say f/f_s , are obtained from $\frac{1}{2}f_1/f_s$, which is the normalized frequency variable of the first level. The complete set is given below:

$$\begin{aligned}
 \text{lowpass channel} \quad A(f) &\geq 38 \text{ dB} \quad \text{for} \quad f/f_s \geq 0.1465 & (65) \\
 \text{bandpass \#1 channel} \quad A(f) &\geq 38 \text{ dB} \quad \text{for} \quad 0.1035 \geq f/f_s \geq 0.293 \\
 \text{bandpass \#2 channel} \quad A(f) &\geq 38 \text{ dB} \quad \text{for} \quad 0.2070 \geq f/f_s \geq 0.3965 \\
 \text{highpass channel} \quad A(f) &\geq 38 \text{ dB} \quad \text{for} \quad 0.3535 \geq f/f_s \geq 0.0
 \end{aligned}$$

where f_s is the sampling frequency. A critical aspect of the 4-band design is the selection of initial positions of the commutator switches (see Fig. 18) which ensure proper routing and timing of the signals throughout the structure. The overall decimation factor of the channel signals is 4 and, consequently, the structure is computed in a 4T time frame. With reference to Fig. 18, the computation is sequenced as follows:

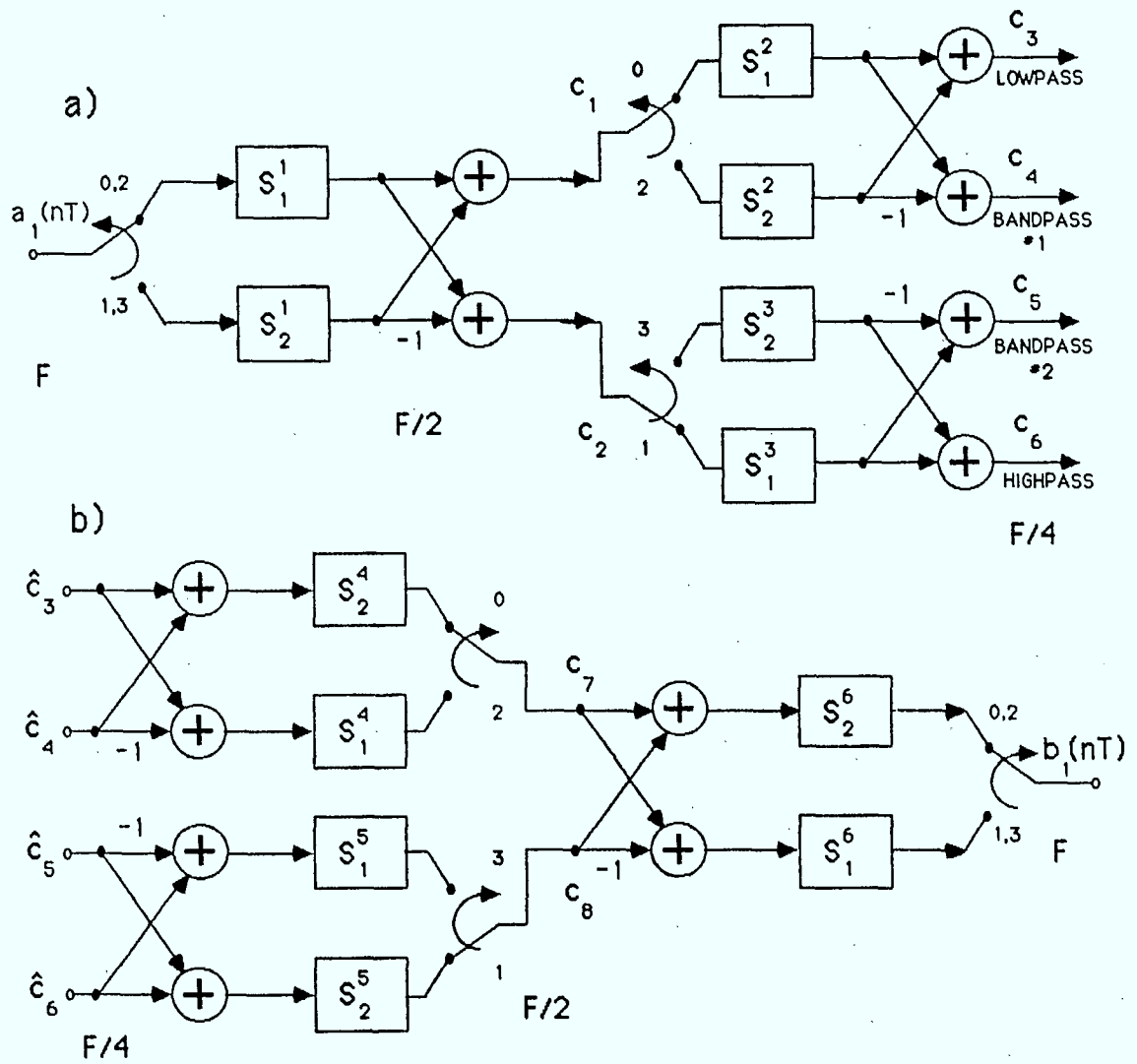


Fig. 18 Four-channel WD filter bank: (a) analyzer and (b) synthesizer.

In the analyzer:

Cycle 0:

Evaluate \hat{S}_1^1 with $a_1(nT)$ as input and compute c_1 .

Evaluate \hat{S}_1^2 with c_1 as input and compute c_3 .

Cycle 1:

Compute c_2 .

Evaluate \hat{S}_2^1 with $a_1((n+1)T)$ as input.

Evaluate \hat{S}_1^3 with c_2 as input and compute c_4 .

Cycle 2:

Evaluate \hat{S}_1^1 with $a_1((n+2)T)$ as input and compute c_1 .

Evaluate \hat{S}_2^2 with c_1 as input and compute c_5 .

Cycle 3:

Compute c_2 .

Evaluate \hat{S}_2^1 with $a_1((n+3)T)$ as input.

Evaluate \hat{S}_2^3 with c_2 as input and compute c_6 .

In the synthesizer:

Cycle 0:

Evaluate \hat{S}_2^6 with $(c_7 + c_8)$ as input.

Evaluate \hat{S}_2^4 with $(\hat{c}_3 + \hat{c}_4)$ as input and set c_7 to its output.

Obtain \hat{c}_3 from channel and take out $b_1(nT)$.

Cycle 1:

Evaluate \hat{S}_2^5 with $(\hat{c}_6 + \hat{c}_5)$ as input and set c_8 to its output.

Evaluate \hat{S}_1^6 with $(c_7 - c_8)$ as input.

Obtain \hat{c}_4 from channel and take out $b_1((n+1)T)$.

Cycle 2:

Evaluate \hat{S}_2^6 with $(c_7 + c_8)$ as input.

Evaluate \hat{S}_1^4 with $(\hat{c}_3 - \hat{c}_4)$ as input and set c_7 to its output.

Obtain \hat{c}_5 from channel and take out $b_1((n+2)T)$.

Cycle 3:

Obtain \hat{c}_6 from channel.

Evaluate \hat{S}_1^5 with $(\hat{c}_6 - \hat{c}_5)$ as input and set c_8 to its output.

Evaluate \hat{S}_1^6 with $(c_7 - c_8)$ as input and take out $b_1((n+3)T)$.

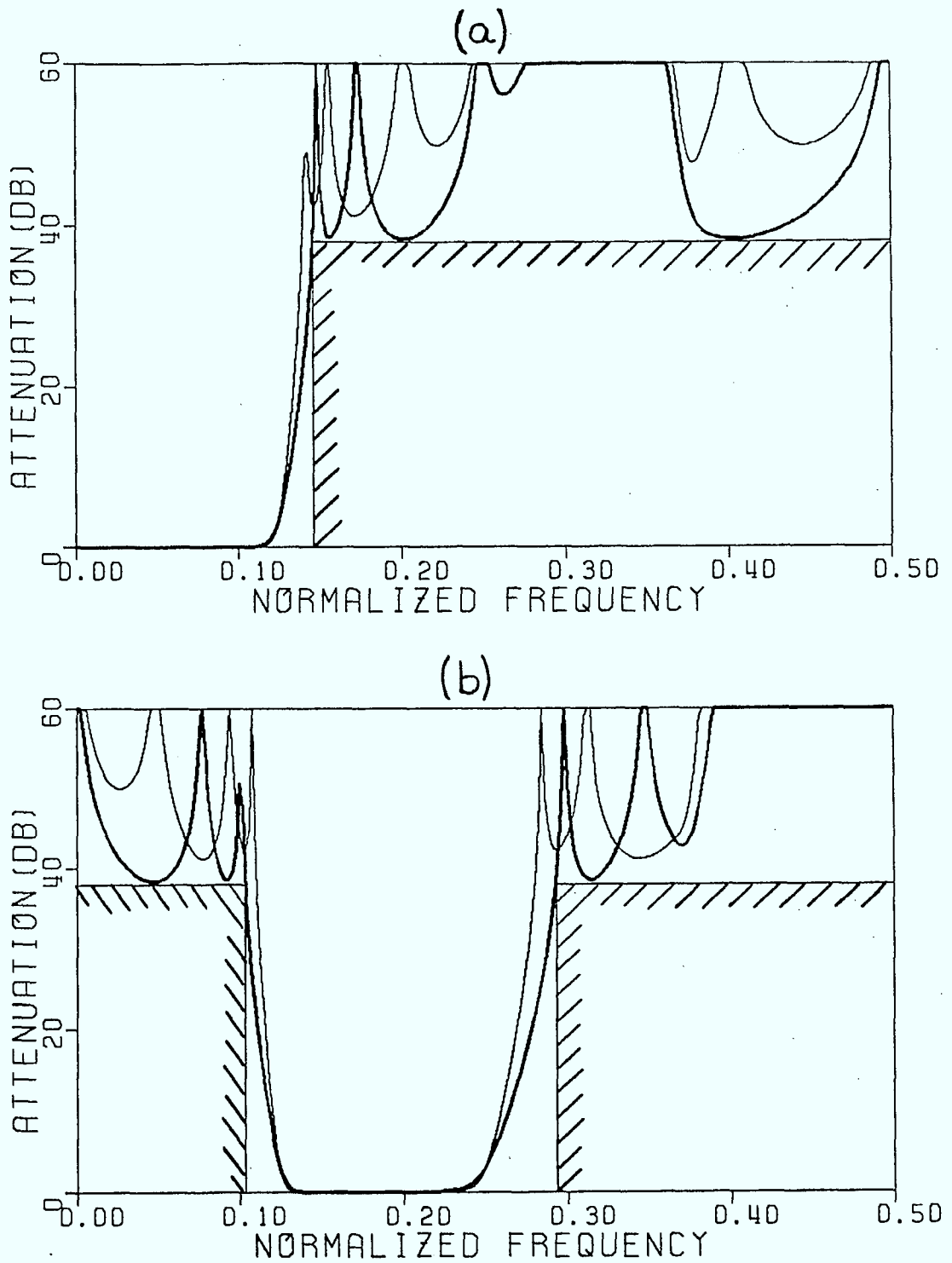


Fig. 19 Frequency responses of the 4-channel WD filter bank: (a) lowpass and (b) bandpass #1 channels for coefficient sets (58) (thin line) and (64) (thick line), respectively.

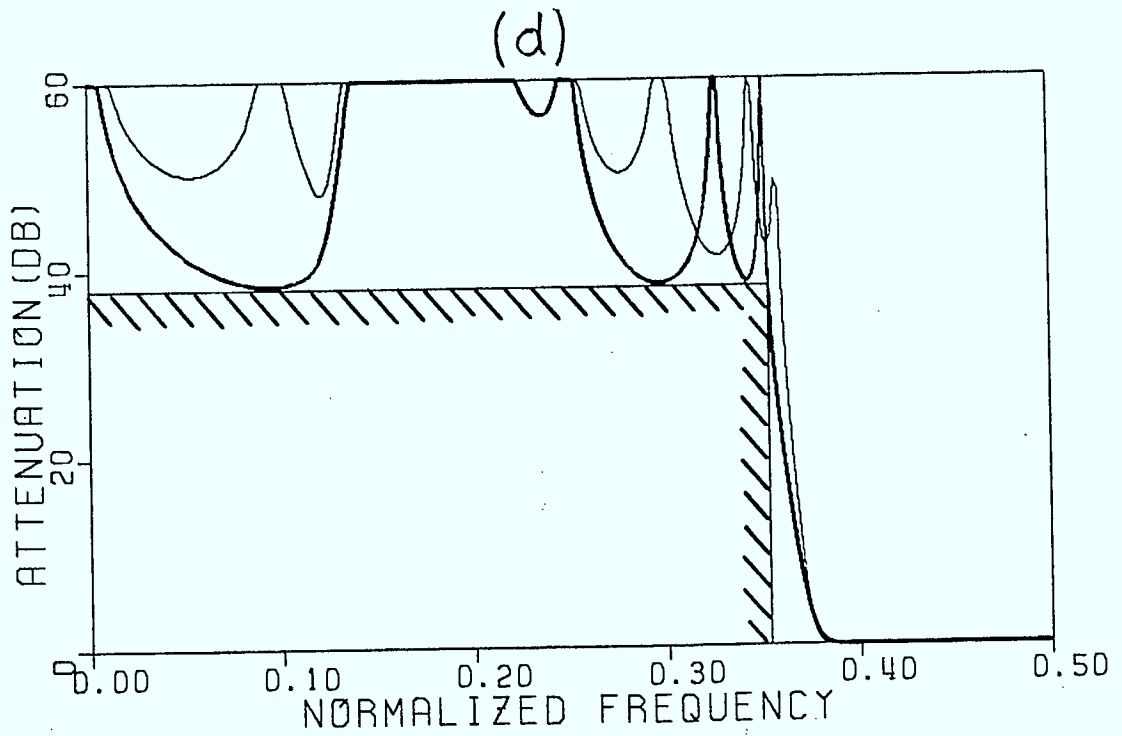
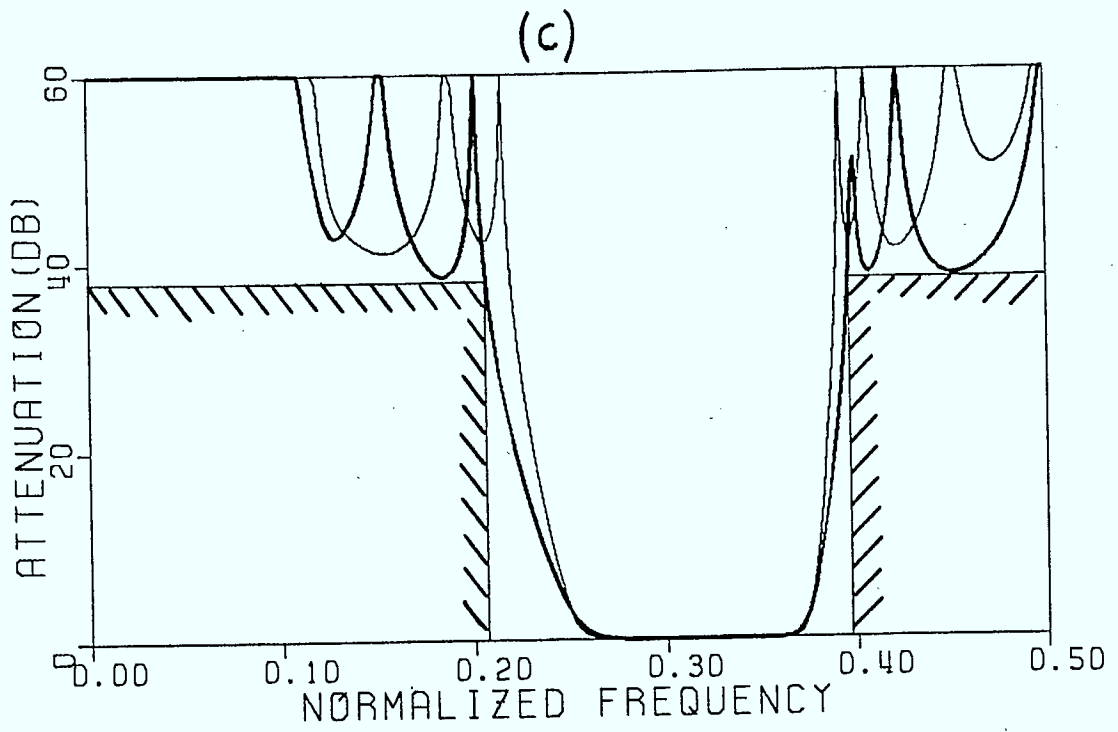


Fig. 19 cont. (c) Bandpass #2 and (d) highpass channels.

The initial switch positions in Fig. 18 were selected such that the computation load during each cycle is about the same, i.e., two $\hat{S}_1(z)$ and two $\hat{S}_2(z)$ must be evaluated during each cycle, for a total of 6 allpass sections. This selection introduces an additional 3 sample delay between the input and output sequences, and is analogous to the 2-band case. The worst path in a suboptimum design would require 8 allpass evaluations. Note also that the channel signals, $c_1, c_2, c_3,$ and $c_4,$ are computed in the analyzer sequentially. The corresponding signals in the synthesizer are also obtained sequentially. This sequencing corresponds to the way these signals, in a realistic situation, are sent and received. Additional details pertaining to this structure are given in Section VIII.

The back-to-back arrangement of the structure in Fig. 18 performs 6 multiplications per input sample and the complete system requires 18 storage locations. These numbers are still well below the requirements of the two-band designs in Section IV! For the coefficient set in (58), the magnitude responses of the four channels are shown in Fig. 19 (thin line). Again, the available margin in the magnitude response can be traded for improved overall group delay response. In the 4-band case, the nonlinearity in the group delay is given by (62) plus another term introduced by the second stage in Fig. 18. To derive the second term, consider the identity in Fig. 20 [16].

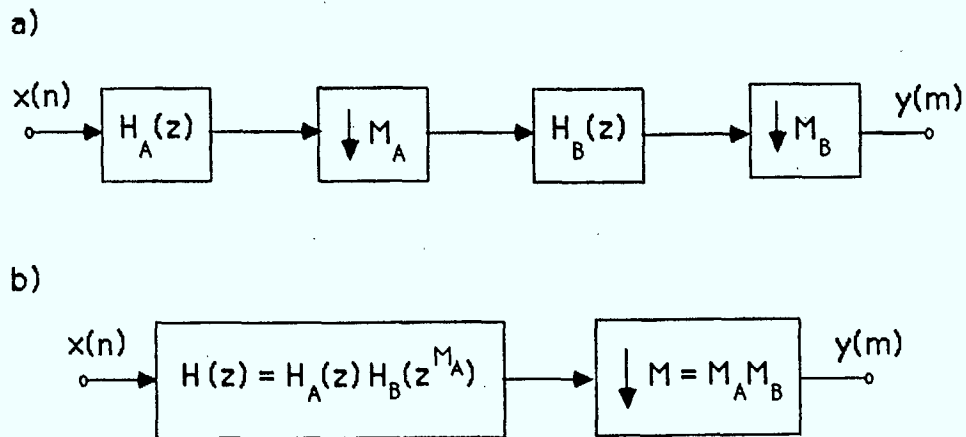


Fig. 20 (a) A two-stage decimator and (b) its single stage identity system. Applying the identity in Fig. 20 to the 4-channel analyzer in Fig. 18a yields 4 transfer functions (before decimation):

$$\begin{aligned}
 H_{LP}(z) &= \left[\hat{S}_1(z^2) + z^{-1} \hat{S}_2(z^2) \right] \left[\hat{S}_1(z^4) + z^{-2} \hat{S}_2(z^4) \right] \\
 H_{BP_1}(z) &= \left[\hat{S}_1(z^2) + z^{-1} \hat{S}_2(z^2) \right] \left[\hat{S}_1(z^4) - z^{-2} \hat{S}_2(z^4) \right] \\
 H_{BP_2}(z) &= \left[\hat{S}_1(z^2) - z^{-1} \hat{S}_2(z^2) \right] \left[\hat{S}_1(z^4) - z^{-2} \hat{S}_2(z^4) \right] \\
 H_{HP}(z) &= \left[\hat{S}_1(z^2) - z^{-1} \hat{S}_2(z^2) \right] \left[\hat{S}_1(z^4) + z^{-2} \hat{S}_2(z^4) \right] .
 \end{aligned} \tag{66}$$

The overall transfer function of the 4-band back-to-back arrangement in Fig. 18 is given by

$$\begin{aligned}
 H_4(z) &= H_{LP}^2(z) - H_{BP_1}^2(z) + H_{BP_2}^2(z) - H_{HP}^2(z) \\
 &= H_2(z) \left[4 z^{-2} \prod_{i=1}^3 \frac{\hat{\gamma}_i z^4 + 1}{z^4 + \hat{\gamma}_i} \right] .
 \end{aligned} \tag{67}$$

The group delay response of $H_4(z)$ is given by

$$\tau_4(\omega) = \tau_2(\omega) + \left[2 + 4 \sum_{i=1}^3 \frac{1 - \hat{\gamma}_i^2}{\hat{\gamma}_i^2 + 2\hat{\gamma}_i \cos(4\omega) + 1} \right] \tag{68a}$$

where the term in brackets is contributed by the bracketed term in (67). We define the objective function as the difference between the first maximum (which is at $\omega \cong \pi/4$) and the first minimum (which is at $\omega = 0$) of the group delay response in (68a); i.e.,

$$f(\hat{\gamma}) = 4 \sum_{i=1}^3 \frac{\hat{\gamma}_i}{1 + \hat{\gamma}_i} \left[\frac{1 - \hat{\gamma}_i}{\hat{\gamma}_i^2 + 1} + \frac{4}{1 - \hat{\gamma}_i} \right] . \tag{68b}$$

A quantized coefficient set that minimizes (68) turns out to be the same as for the two-band case, and is given in (64). This is not too surprising in view of the fact that both levels of the tree structure use the same set of filters. The overall group delay nonlinearity is decreased by 21 samples, as shown in Fig. 21 (thick line). The magnitude responses of the 4 channels for the new set of coefficients are shown in Fig. 19a-d (thick lines).

The group delay distortion for both designs falls within the allowable range as specified in [23]. However, the coefficient set in (58) yields a 4-band design with overall group delay distortion that falls outside the allowable range, as specified in [22]. In fact, our informal listening tests of processed speech showed no discernable difference between the two designs. Hence, if the additional group delay distortion

is of no importance, then the simpler coefficient set given in (58) should be used in the 4-band design.

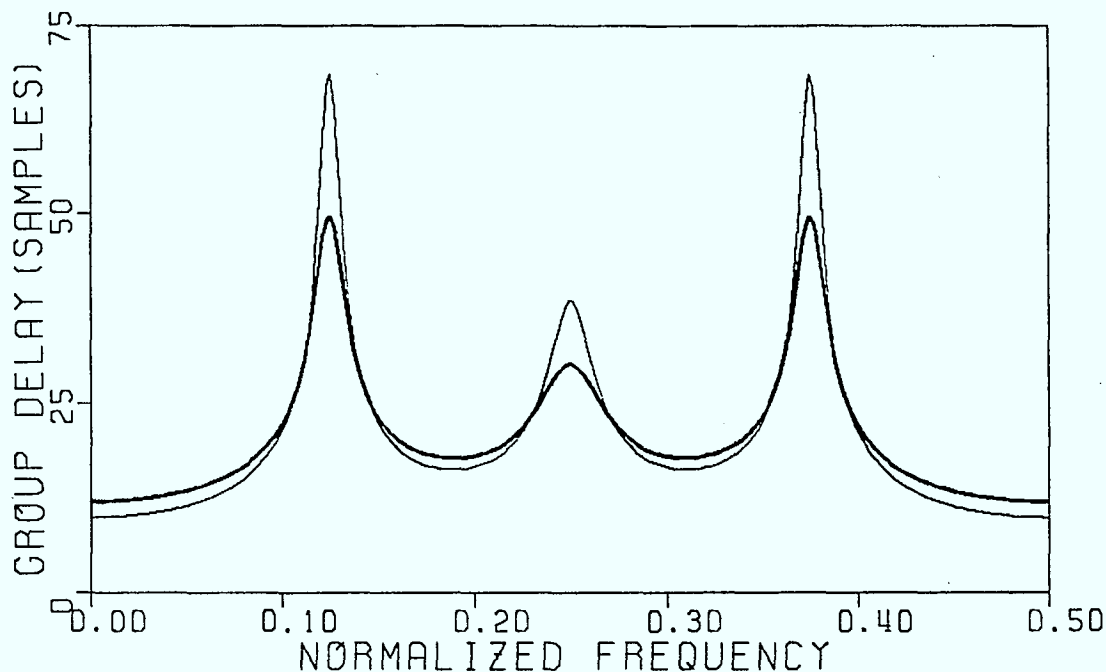


Fig. 21 Overall group delay responses of the 4-channel WD filter bank for coefficient sets (a) (58) (thin line) and (b) (64) (thick line).

Two more important properties of bireciprocal WD lattice filters should be mentioned. It has been shown in [29] that a simple 2's complement truncation scheme is sufficient for ensuring freedom from zero-input parasitic oscillations, i.e., nonlinear stability is ensured without having to employ a more complex magnitude truncation scheme [30]. Furthermore, it has been pointed out in [28] that the internal signals at all the ports in the bireciprocal WD lattice filter do not exceed the input level at steady-state conditions for any frequency. This means that, for sinusoidal excitation, WD filters are inherently scaled in the best way possible. In the next two Sections we present the details concerning real-time implementation^{of} the above WD designs on the MC68000 microprocessor.

VII. REAL-TIME EVALUATION SYSTEM CONFIGURATION

The filter structures described in Sections V and VI were implemented in real time on the Motorola MC68000 microprocessor. A general purpose microprocessor such as the MC68000 was selected since it clearly demonstrates the efficiency of the WD filtering algorithms; in a dedicated signal processor, the architecture can compensate for an otherwise inefficient filter implementation. As well, software development tools and a MC68000 microcomputer board were easily obtainable and facilitated the implementation of the filter structures.

The WD filter structures were evaluated using a microcomputer/microprocessor based system. A block diagram of the evaluation system hardware is shown in Fig. 22. The system consists of a personal computer, a Motorola MC68000-based microprocessor board with 8-bit A/D and D/A converters, a signal conditioning subsystem and an audio input/output subsystem. Each system component is described individually below:

- (1) The personal computer used in this system is of an IBM-PC/XT compatible type running at an 8 MHz clock speed. The system includes one 400 kbyte floppy disc drive and a 20 Mbyte hard disc drive. The hard disc is necessary to store large speech data files used in off-line filter simulations. The computer is also equipped with an Intel 8087 math co-processor enhancement which decreases the execution times of programs significantly. This also facilitates the simulation of multi-band filter structures and the off-line processing of speech data (acquired using the MC68000 microprocessor board). Communication between the personal computer and the remainder of the system is through the serial asynchronous communication port of the computer.
- (2) A Motorola MC68000 microprocessor was selected for implementing the multi-band WD filter bank structures. The MC68000 has an advanced general purpose architecture consisting of eight 32-bit data registers and eight 32-bit address registers [31]. A multiply and a divide instruction are also available. Although a single-instruction multiply is available, it is unnecessary for our purposes, and the more computationally efficient 'shift-and-add' method will be used instead. The added efficiency is partly due to the flexibility offered by the number and type of internal registers on the MC68000. This, combined with the high clock speed (up to 20 MHz), makes the MC68000 a viable choice for the evaluation microprocessor subsystem.

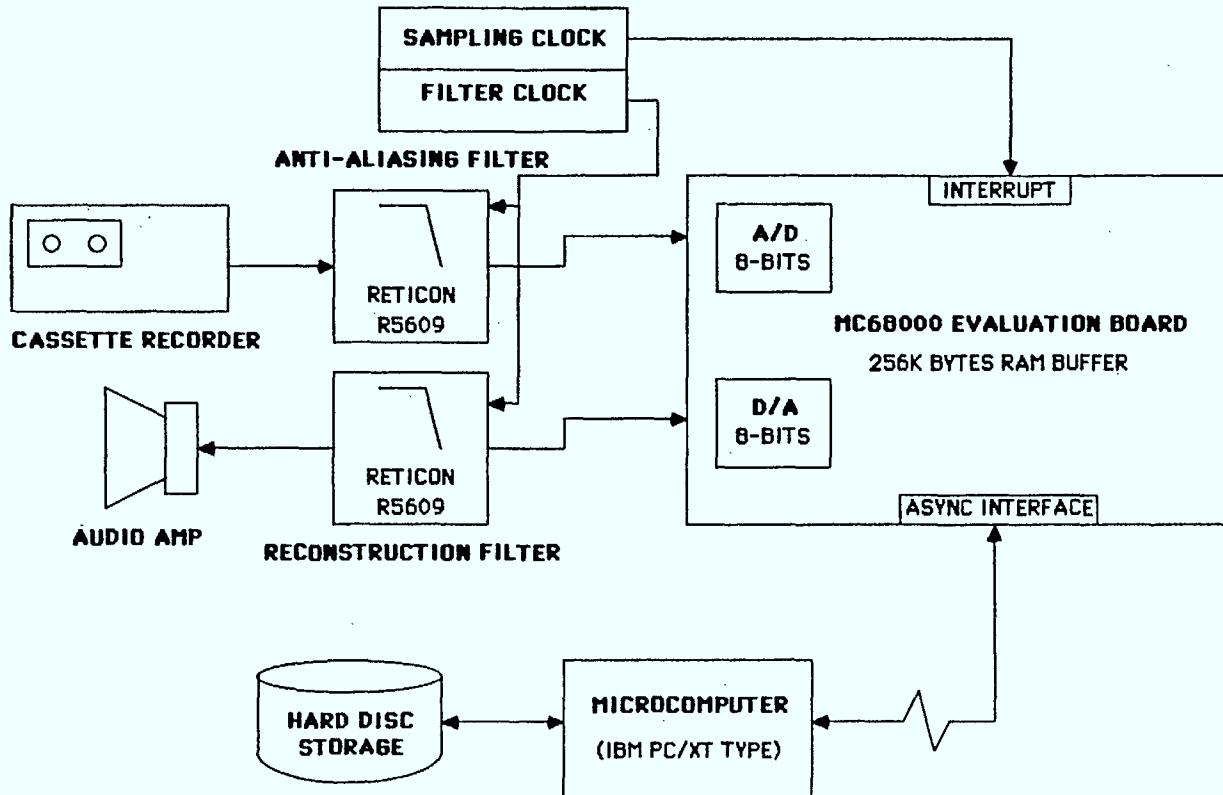


Fig. 22. Evaluation system hardware block diagram.

The particular microprocessor board used for the evaluation subsystem is an enhanced version of the Motorola MC68000 Educational computer board [32]. The enhancements include an 8 MHz MC68000, the addition of 8-bit A/D and D/A convertors, and the expansion of the user memory to greater than 256 kbytes of RAM. The board contains two serial communication ports used as a terminal port and a host data transfer port. The board also contains a monitor program which facilitates various utility functions such as in-line assembly, data transfers and single-step operation of the CPU. An on-board timer device allows the generation of periodic CPU interrupts, which can serve as a sampling clock.

- (3) Prior to digitization, a speech signal to be processed must be amplified and band-limited. Similarly, after D/A conversion, the output signal must be reconstructed through low pass filtering. The signal conditioning subsystem consists of an input voltage amplifier followed by a low pass switched-capacitor filter (Reticon R5609 7th order elliptic low pass). The reconstruction filter is an identical Reticon switched-capacitor filter. The desired filter cutoff frequency is determined by an externally applied clock signal of 100 times the desired filter cutoff.
- (4) The audio input/output subsystem consists of a hi-fidelity cassette tape recorder as a speech source and an audio output amplifier for audio playback of the processed speech. A list of standard Harvard sentences [33] recorded by one male and one female speaker was used as the input speech material.

A block diagram of the software incorporated into the evaluation system is shown in Fig. 23. The evaluation system is used in two ways:

- i) simulation of multi-band filters in high level language (Pascal) with off-line speech processing, and
- ii) interaction with the microprocessor board for real-time filter implementations.

The filter structures simulated on the personal microcomputer are written in the Pascal programming language. The DFT of the unit sample response sequence is used to verify proper simulation operation. A typical simulation of the WD designs from Sections V and VI proceeds as follows:

- i) A flow graph of the filter structure is translated into Pascal instructions and stored on disc via the editor.
- ii) The program is compiled and executed using the run-time facility of the Pascal compiler.
- iii) The DFT of the unit sample response is displayed on the screen and is compared to the desired response.
- iv) Speech data acquired using the microprocessor board is then applied to the simulation filter and output values are stored on disc.
- v) Processed speech is then transferred back to the microprocessor board memory, to be played back through the D/A and audio output subsystem.

Several utility routines for formatting speech data are required to support data transfers to and from the microprocessor board. These routines were also written in Pascal and executed on the personal computer.

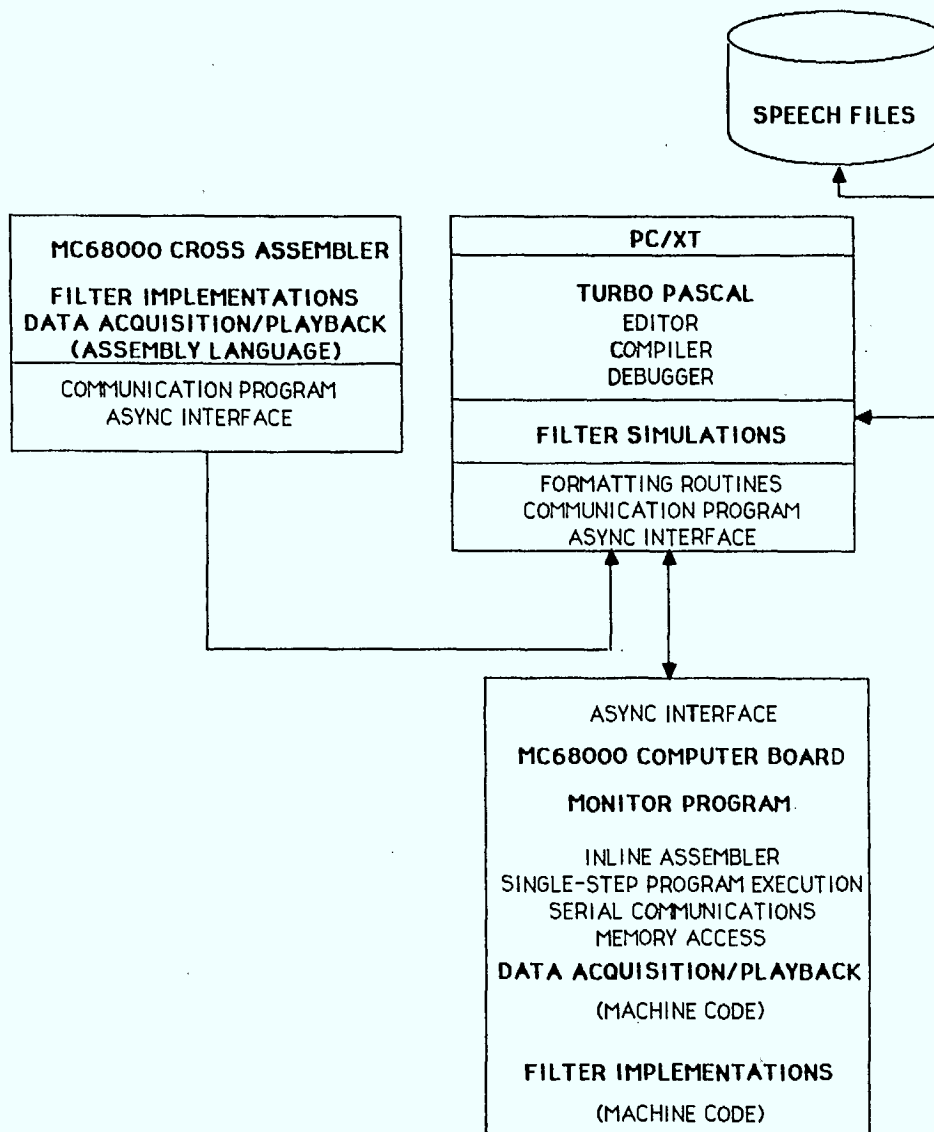


Fig. 23. Evaluation system software block diagram.

Assembly language programs must be written for the MC68000 in order to implement filter algorithms in real-time. Assembly capability for the MC68000 was available on a Z-80 based personal computer assembled at the University of Manitoba. The computer runs the CP/M operating system and is equipped with a cross-assembler for the MC68000. A program written and assembled on the CP/M (using a standard text editing program) is subsequently transferred to the PC/XT computer and down to the microprocessor board. This program is then executed in a single step fashion to ensure that it is an accurate implementation of the filtering algorithm. Unit sample response values are used for verification at this level. Finally, the program is executed in real-time where speech material is processed continuously and audio output is available for informal listening tests.

VIII. REAL-TIME IMPLEMENTATION OF WAVE DIGITAL QUADRATURE MIRROR FILTER BANKS

In this section we present the details and results concerning real-time implementation of the 2- and 4-band WD filter bank designs from Sections V and VI, respectively.

Two-Band WD Design From Section V

A typical application of a multi-band filter in sub-band coding is the two-band or split-band ADPCM sub-band coder operating at 14 kHz [11, 12, 13]. Such a coder has been applied to commentary grade (7 kHz bandwidth) speech and music encoding [10, 11]. An altered 2-band WD filter bank design from Fig. 15 is shown in Fig. 24.

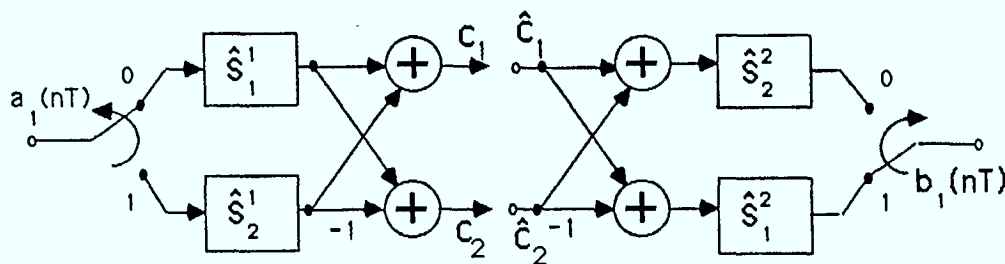


Fig. 24 Altered version of the 2-band WD design from Fig. 15.

As mentioned in Section V, we start the output switch in Fig. 24 in its upper position. This is done so that the computational load is distributed evenly between the two cycles, and its consequence is the introduction of one additional sample delay between the input and output sequences. We estimate that without this preventive measure about 180 machine cycle discrepancy between the two cycles would have resulted. With reference to Fig. 24, the sequence of operations to be transcribed into machine language is as follows:

Cycle 0:

Analyzer: Evaluate \hat{S}_1^1 with $a_1(nT)$ as input and compute c_1 .

Synthesizer: Evaluate \hat{S}_2^2 with $(\hat{c}_1 + \hat{c}_2)$ as input and take out $b_1(nT)$. Obtain \hat{c}_1 from channel.

Cycle 1:

Analyzer: Compute c_2 first and then evaluate \hat{S}_2^1 with $a_1((n+1)T)$ as input.

Synthesizer: Obtain \hat{c}_2 from channel, evaluate \hat{S}_1^2 with $(\hat{c}_1 - \hat{c}_2)$ as input and take out $b_1((n+1)T)$.

We would like to point out that, contrary to what might seem reasonable, the retrieval of \hat{c}_1 from the channel during cycle 0 cannot occur before the evaluation of \hat{S}_2^2 . If this is not observed the results are disastrous. The signal flowgraphs of $\hat{S}_1(z)$ and $\hat{S}_2(z)$ are shown in Fig. 16. Henceforth, we assume that both analysis and synthesis functions are located on one processor, i.e., full duplex operation with two sources communicating over a common channel is considered. This means that the signal sequence $c_1(n), c_2(n), c_1(n+1), c_2(n+1) \dots$ is different from $\hat{c}_1(n), \hat{c}_2(n), \hat{c}_1(n+1), \hat{c}_2(n+1) \dots$. However, we equate the two sequences for the purposes of analysis and verification of the overall configuration. In other applications, where either the analyzer or the synthesizer is to be implemented, the computational load is roughly halved.

From Section V we have two sets of coefficients: one (58) was obtained from [24], and the other set (64) was optimized with respect to the overall group delay distortion. Both sets were assessed using the off-line procedure described in Section VII, and from our informal listening tests, no discernable difference was found between them. Since the first set (58) requires one addition and 5 shifts fewer, we elected to implement this set on the microprocessor. We would like to point out, however, that if the application requires phase equalization (flat group delay), then the second set, as can be seen from Fig. 17b, offers a substantial reduction of $10\frac{1}{2}$ samples in the overall delay.

The WD property of low sensitivity to coefficient quantizations implies that, in the majority of cases, one can find simple coefficients that do not require a multiply routine for their implementation. Instead, the multiplications are realized using the 'shift-and-add' method which, when applied to the coefficient set in (58), yields

$$\gamma_1 x_1 = \frac{x_1}{2}, \quad \gamma_2 x_2 = \frac{1}{8} \left(x_2 + \frac{x_2}{4} \right), \quad \gamma_3 x_3 = \frac{1}{8} \left(x_3 + \frac{1}{4} \left(x_3 + \frac{x_3}{4} \right) \right) \quad (69)$$

where the x_i are some arbitrary values. The nesting in (69) is used so that the underflow bits are part of the exact result and, consequently, round-off error is

minimized. The longest multiplication in (69) requires 17% fewer machine cycles than an equivalent internal multiply on the MC68000.

The assembly language program written for the MC68000 that implements the 2-band WD design from Fig. 24 is listed in Appendix A. This program was written so that future extensions to the 4-band case can be done with ease. A known unit sample response sequence (generated by the high-level-language simulation) was used to verify proper operation of the algorithm. All arithmetic operations and memory transfers were performed using 16 bits. Although 32 bit operations could have been used, 16 bits is more than sufficient for speech and audio signals, and results in a substantial reduction in the total number of machine cycles required. The machine cycle counts for the 2-band design are given in Table 2. It is seen that cycle 1 is the worst path. It requires 478 machine cycles which, for a 14 kHz sampling rate, imposes a minimum clock rate of 6.7 MHz. Thus, an MC68000 running at 8 MHz is sufficient for implementing a full duplex 2-band WD design.

Table 2				
Computational Requirements of Two- and Four-Band WD Filter Bank Designs				
2-Band ($f_s = 14 \text{ kHz}$) 4-Band ($f_s = 8 \text{ kHz}$)	Machine Cycle Count			Microprocessor Minimum Clock Rate
	Analyzer	Synthesizer	Total	
2-Band Cycle 0	252	190	442	
2-Band Cycle 1	180	298	478	6.7 MHz
4-Band Cycle 0	428	266	694	
4-Band Cycle 1	366	368	734	5.9 MHz
4-Band Cycle 2	344	374	718	
4-Band Cycle 3	264	466	730	

In addition to the band-splitting and reconstructing operations a sub-band coder includes the functions of encoding and decoding. These functions are most often realized using the adaptive differential pulse-code modulation (ADPCM) scheme [10], [13], and it is generally concluded that their implementation is computationally less intensive than the band-splitting and reconstructing operations. This conclusion, however, does not hold in the case of the WD filter banks from Sections V and VI. Instead, the remarkable efficiency of these structures forces us to conclude that it is the functions of encoding and decoding that are the more computationally intensive! Our conservative estimates show that, based on the design from [35], the implementations of the ADPCM encoder and decoder on the MC68000 would require about 700

and 500 machine cycles, respectively. Thus, for the 2-band design, a total of about 1680 machine cycles would be required, of which, only 30% is needed for the band-splitting and reconstructing operations. In this case, the full duplex operation is not possible since, for $f_s = 14 \text{ kHz}$, the minimum clock rate required is 23.5 MHz. This value is greater than the maximum, currently available MC68000 clock speed which is 20 MHz. It follows that the operations of analysis and synthesis have to be distributed between two microprocessors which would require a minimum clock rate of 13.3 MHz and 11.2 MHz, respectively. An interesting comparison can be made with the 2-band sub-band coder that uses the FIR benchmark filter bank from Section IV. This coder was implemented on the Bell Laboratories digital signal processing (DSP) integrated circuit [10] which has been optimized for signal processing applications. It was reported in [10] that two DSP chips are required for implementing the analysis and synthesis algorithms. However, in contrast to the MC68000, the Bell DSP is not commercially available.

Four-Band WD Design From Section VI

A sub-band coder with 4 bands operating at an 8 kHz sampling rate is the next level of complexity in the implementation of sub-band coders [13]. A 4-band coder offers greater bit rate compression than the 2-band coder. The WD filter bank for the 4-band coder is shown in Fig. 18. The initial switch positions were chosen to allow even distribution of the overall computational load among the 4 cycles. This choice introduces an additional 3 sample delay, but it eliminates a 360 machine cycle discrepancy between the two extreme paths that otherwise would have resulted. As in the 2-band case, the two sets of coefficients, given by (58) and (64), were assessed using the off-line procedure described in Section VII. Our informal listening tests showed no discernable difference between them and, as a result, we elected to implement the simpler set given by (58). If circumstances require phase equalization, then the second set (64) should be used since it offers (see Fig. 21) a reduction of 21 samples in the overall delay.

The assembly language program that implements the 4-band WD filter bank from Fig. 18 is listed in Appendix B. This program is a straightforward 2-level tree-type extension of the program for the 2-band case. The machine cycle counts are given in Table 2, and it is seen there that cycle 1 accounts for the longest delay of 734 machine cycles which imposes a minimum clock rate of 5.9 MHz. If the functions of encoding and decoding are included, then the longest path requires about 1930 machine cycles which imposes a minimum clock rate of 15.5 MHz. As in the 2-band case, the functions of encoding and decoding account for most of the computational load (62%). This fact notwithstanding, the entire 4-band sub-band encoder/decoder

algorithm can be run in real-time on a single MC68000 microprocessor with a 20 MHz clock rate! A structure similar to the one in Fig. 18 (actually a 5-band design) required two Bell Labs DSP chips for its implementation [13].

The data points in Table 2 allow us to derive an estimate of the computational requirement for an arbitrary number of bands N . A conservative straight line extrapolation of the two points from Table 2 yields an estimate for the minimum clock rate, and is given by

$$f_{CLOCK} \approx f_s (256x + 222) \quad , \quad \text{for } 2^{(x-1)} < N \leq 2^x \quad (70)$$

where f_s is the desired sampling frequency and 2^x is the maximum decimation factor which equals the number of cycles over which the computation must be distributed. Note that the above equation does not include the computational requirements of the encoding and decoding functions. As an example, for $N=5$ and $f_s = 14kHz$, the minimum clock rate from (70) would be 7.9 MHz without and 17.5 MHz with the encoder/decoder, respectively. The $N=5$ design is in fact the one implemented on the two DSP chips [13]. The same design using the WD approach would require only one MC68000 microprocessor running at 20 MHz. Also, it is seen from (70) that as N grows exponentially, the computational load grows only linearly!

Table 3		
Comparative Performance of Three Signal Processors		
Processor	Clock Speed (MHz)	Percent Real-Time ($f_s = 14kHz$)
TMS320-10	5	40
MC68000	20	67
Bell Labs DSP	5	78

Table 3 shows the performance of the 2-band WD sub-band coder implemented on the MC68000 as compared to the benchmark QMF bank from Section IV implemented on two dedicated signal processing devices. The TMS320-10 result is taken from an actual implementation [34], and the DSP result was published in [13]. The comparison is made in terms of the percent of real-time capability of performing the band-splitting and encoding functions with $f_s = 14 kHz$. Not surprisingly, the MC68000 is only comparable to the TMS320-10 and the DSP if the ratio of clock speeds is 8 and 4, respectively. This is due to the highly efficient, signal processing oriented architectures of both the TMS320-10 and the DSP. As seen from Table 3, the TMS320-10 is twice as efficient as the Bell Labs DSP.

IX. CONCLUSIONS

The main conclusion from our investigation is that the WD filter bank, whether band-splitting or reconstruction, outperforms the FIR and other IIR filter banks by a large margin. As a comparison of the 2-band filter banks, the latter require 10 and 7 times more multiplications per input sample, respectively! Moreover, the symmetric WD lattice with a bireciprocal characteristic function has the following superior properties:

- (1) this structure is equivalent to a critically sampled, uniform 2-band DFT polyphase filter bank, thus establishing an important common conceptual framework;
- (2) the bireciprocal property reduces the number of required allpass sections, and therefore multipliers, by a factor of more than 2 (the 7th-order filter has only 3 multipliers);
- (3) two output ports are available and provide mirror-image responses;
- (4) the back-to-back arrangement of the structure inherently provides an overall allpass response, regardless of the values of parameters;
- (5) the low sensitivity to parameter quantizations and the small total number of parameters allows a thorough search to be conducted in the discrete space for parameters that do not require a multiply routine for their implementation (i.e., the 'shift-and-add' procedure is sufficient);
- (6) a simple 2's complement truncation scheme is sufficient to ensure complete suppression of all zero-input parasitic oscillations (this property is analogous to the stability property of FIR filters);
- (7) steady-state analysis shows that the internal signals at all the ports do not exceed the input level for all frequencies, which means the structure is inherently scaled in the best possible way;
- (8) in many instances the available margin in the magnitude response can be traded for improved group delay response (this is greatly facilitated by the existence of closed-form formulae for the group delay response); and
- (9) the small number of parameters allows an arbitrary starting point for a suitable nonlinear optimization algorithm, thus circumventing the need of an initial design step (the order of the filter can be estimated from the specifications).

The above properties combine to make the WD approach to the design of filter banks the correct choice for sub-band coders and other similar multi-rate applications. The relative simplicity of the resulting algorithm makes its implementation possible using an off-the-shelf general purpose microprocessor.

REFERENCES

- [1] J. L. Flanagan, "Computers that talk and listen: man-machine communication by voice", *Proc. IEEE*, vol. 64, No. 4, pp. 405-415, April 1976
- [2] E.A. Feigenbaum and P. McCorduck, *The Fifth Generation*, Reading, Mass: Addison-Wesley, 1983.
- [3] X. Maitre, and T. Aoyama, "Speech coding activities within CCITT: status and trends", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Paris, 1982, vol. 2, pp. 954-959.
- [4] G.O. Martens and V.N. Shkawrytko, "Voice Input/Output incorporation into the existing Telidon/Videotex system concept", Final Report for DSS Contract No. OSU82-00217, July 1, 1982 - June 30, 1983.
- [5] J.L. Flanagan, et al, "Speech coding", *IEEE Trans. Commun.*, vol. COM-27, No. 4, April, 1979, pp. 710-737.
- [6] J.N. Holmes, "A survey of methods for digitally encoding speech signals", *The Radio Electronic Engineer*, vol. 52, No. 6, pp. 267-276, June 1982.
- [7] T.P. Barnwell III, et al, "A real-time speech subband coder using the TMS320-10", presented at Southcon 1984.
- [8] R.E. Crochiere, "On the design of sub-band coders for low-bit-rate speech communication", *Bell Syst. Tech. J.*, vol. 56, no. 5, pp. 747-770, May-June 1977.
- [9] R.E. Crochiere, et al, "Digital coding of speech in sub-bands", *Bell Syst. Tech. J.*, vol. 55, no. 8, pp. 1069-1085, Oct. 1976.
- [10] R.E. Crochiere, "Sub-band coding", *Bell Syst. Tech. J.*, vol. 60, no. 7, pp. 1633-1653, Sept. 1981.
- [11] J.D. Johnston and D.J. Goodman, "Digital transmission of commentary-grade (7 kHz) audio at 56 or 64 kbits/s", *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 136-138, Jan. 1980.
- [12] J.D. Johnston and R.E. Crochiere, "An all-digital "commentary grade" subband coder", *J. Audio Eng. Soc.*, vol. 27, no. 11, pp. 855-865, Nov. 1979.
- [13] R.E. Crochiere, R.V. Cox and J.D. Johnston, "Real-time speech coding", *IEEE Trans. Commun.*, vol. COM-30, no. 4, pp. 621-634, April 1982.
- [14] 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.

- [15] M.G. Bellanger, G. Bonnerot and M. Goudreuse, "Digital filtering by polyphase network: application to sample-rate alteration and filter banks", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, no. 2, pp. 150-156, Apr. 1978.
- [16] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1983.
- [17] H. G. Martinez and T.W. Parks, "Design of recursive digital filters with optimum magnitude and attenuation poles on the unit circle", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, no. 2, pp. 150-156, Apr. 1978.
- [18] H. G. Martinez and T.W. Parks, "A class of infinite-duration impulse response digital filters for sampling rate reduction", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, no. 2, pp. 154-162, Apr. 1979.
- [19] A.V. Oppenheim and R.W. Schaffer, *Digital Signal Processing*, Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1975.
- [20] D. Esteban and C. Galand, "Application of quadrature mirror filters to split band voice coding schemes", *Proc. 1977 IEEE Int. Conf. ASSP*, Hartford, Conn., pp. 191-195, May 1977.
- [21] J.D. Johnston, "A filter family designed for use in quadrature mirror filter banks", *Proc. 1980 IEEE Int. Conf. ASSP*, pp. 291-294, Apr. 1980.
- [22] 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.
- [23] P.C. Millar, "Recursive quadrature mirror filters - criteria specification and design method", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 2, pp. 413-420, Apr. 1985.
- [24] A. Fettweis, J.A. Nossek and K. Meerkötter, "Reconstruction of signals after filtering and sampling rate reduction", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 4, pp. 893-902, Aug. 1985.
- [25] V. Belevitch, *Classical Network Theory*, San Francisco, CA: Holden-Day, 1968.
- [26] A. Fettweis, A. Sedlmeyer and H. Levin, "Lattice wave digital filters", *Int. J. Circ. Th. Appl.*, vol. 1, pp. 5-10, Jan. 1985.
- [27] W. Wegener, "Wave digital directional filters with reduced number of multipliers and adders", *Arch. Elektr. Ubert.*, vol. 33, pp. 239-243, Jun. 1979.
- [28] L. Gazsi, "Explicit formulae for lattice wave digital filters", *IEEE Trans. Circ. Th. Syst.*, vol. CAS-32, pp. 68-88, Jan. 1985.

- [29] J.A. Nossek and H.D. Schwartz, "Wave digital lattice filters with applications in communication systems ", *Proc. IEEE Int. Symp. Circuits Syst.*, Newport Beach, CA, May 1983, vol. 2, pp. 845-848.
- [30] A. Fettweis and K. Meerkötter, "Suppression of parasitic oscillations in wave digital filters", *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 239-246, Mar. 1975 and p. 575, June 1975.
- [31] *Motorola Microprocessors Data Manual*, Austin, TX: Motorola, Inc., 1981.
- [32] *MC68000 Educational Computer Board User's Manual*, Motorola, Inc., 1982.
- [33] "IEEE recommended practice for speech quality measurements", *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 225-246, Sept. 1969.
- [34] W. A. Lavoie and R. Nesbitt, "A split-band ADPCM voice coder using a TMS320-10 signal processor", BSc Thesis, Dept. of E. E. University of Manitoba, 1985-86.
- [35] J.R. Boddie et al, "Digital Signal Processor: ADPCM Coding", *Bell Syst. Tech J.*, vol. 60, no. 7, pp. 1547-1561, Sept. 1981, part 2.
- [36] A. Fettweis, "Wave digital filters: theory and practice", *Proc. IEEE*, vol. 74, no. 2, pp. 270-327, Feb. 1986.

APPENDIX A

TWO-BAND WAVE DIGITAL FILTER BANK

The following is a listing of the assembly language program developed for the Motorola MC68000 microprocessor which implements a 2-band polyphase wave digital filter bank. This program was assembled on a CP/M based cross assembler which is relies on the publication "MC68000 16-Bit Microprocessor User's Manual", Third Edition, edited by Karen Scoble and published by Prentice-Hall of Englewood Cliffs, New Jersey, 07632.

NEWFLTR IDNT 1,0 TWO-CHANNEL POLYPHASE WDF BANK

* DEFINE SYSTEM DEPENDENT ADDRESS LOCATIONS

TCR EQU \$00010021
TIVR EQU \$00010023
CPRH EQU \$00010027
CPRM EQU \$00010029
CPRL EQU \$0001002B
TSR EQU \$00010035
ACIA EQU \$00010042
ATOD EQU \$00030001
DTOA1 EQU \$00020000
DTOA2 EQU \$00020001
TOFF EQU \$A6
TON EQU \$A7
HI EQU \$00
MID EQU \$00
LO EQU \$01
INTVEC EQU \$100

* DEFINE INTERRUPT VECTOR VALUES FOR CYCLE 0 AND CYCLE 1

CYC0 EQU \$1000
CYC1 EQU \$2000

* DEFINE MEMORY LOCATION LABELS

C1CHAN EQU \$3000
C2CHAN EQU \$3010

* ANALYZER MEMORY ALLOCATION

ORG \$3100
AADDR DS.L \$20

* SYNTHESIZER MEMORY ALLOCATION

ORG \$3200
SADDR DS.L \$20

ORG \$100
IVECTOR DC.L \$00001000

* MEMORY ALLOCATION FOR INTERMEDIATE VALUES

* TABLE OF VARIABLE/REG DEFN:

D0 - VAR A0 - MEM PTR
D1 - VAR A1 - N/U
D2 - VAR A2 - N/U
D3 - VAR A3 - N/U
D4 - VAR A4 - N/U
D5 - VAR A5 - N/U
D6 - O/P A6 - N/U
D7 - VAR A7 - STACK PTR

* ORIGIN

ORG \$0900

•
• MACRO DEFINITIONS
•
• S2 REFLECTANCE CALCULATION
•

```
S2  MACRO
    MOVE.W    D0,D1
    MOVE.W    (A0),D2
    SUB.W    D2,D0
    ASR.W    #01,D0
    ADD.W    D0,D1
    ADD.W    D2,D0
    MOVE.W    D1,(A0)
    ENDM
```

•
•
• S1 REFLECTANCE CALCULATION
•

```
S1  MACRO
    MOVE.W    D0,D1
    MOVE.W    (A0),D2
    MOVE.W    4(A0),D3
    SUB.W    D2,D0
    MOVE.W    D0,D4
    ASR.W    #02,D4
    ADD.W    D4,D0
    ASR.W    #03,D0
    ADD.W    D0,D1
    ADD.W    D2,D0
    MOVE.W    D1,(A0)
    ADD.W    D3,D0
    MOVE.W    D0,D1
    ASR.W    #02,D0
    ADD.W    D1,D0
    ASR.W    #02,D0
    ADD.W    D1,D0
    ASR.W    #03,D0
    SUB.W    D3,D0
    MOVE.W    D0,4(A0)
    SUB.W    D0,D1
    MOVE.W    D1,D0
    ENDM
```

•
• INITIALIZATION
•
• - SET 68231 FOR EXTERNAL CLOCK INPUT
•

```
INIT:
    MOVE.B    #TOFF,TCR
    MOVE.B    #HI,CPRH
    MOVE.B    #MID,CPRM
    MOVE.B    #LO,CPRL
    MOVE.B    #$40,TIVR
    MOVE.W    SR,D0
    ANDI     #$F8FF,D0
    OR       #$0100,D0
    MOVE     D0,SR
```

•
START:

ZERO OUT INTERNAL REGISTERS

```
MOVE.W    #$00,D0
MOVE.W    #$00,D1
MOVE.W    #$00,D2
MOVE.W    #$00,D3
MOVE.W    #$00,D4
MOVE.W    #$00,D5
MOVE.W    #$00,D6
MOVE.W    #$00,D7
MOVE.W    D0,A0
MOVE.W    D0,A1
MOVE.W    D0,A2
MOVE.W    D0,A3
MOVE.W    D0,A4
MOVE.W    D0,A5
MOVE.W    D0,A6
MOVE.B    D0,ATOD
MOVE.B    #TON,TCR
```

```
LOOP:     BRA    LOOP        WAIT FOR INTERRUPT
```

```
.
* INTERRUPT ROUTINES
.
* CYCLE 0
.
```

```
ORG    $01000
ANAL0: CLR.W D0
        MOVE.B    ATOD,D0
        MOVE.B    D6,DTOA1
        LEA.L    AADDR,A0
        S1
        MOVE.W    D0,8(A0)
        ADD.W 24(A0),D0
        MOVE.W    D0,12(A0)
        MOVE.W    D0,C1CHAN
```

```
SYNTH0: LEA.L    SADDR,A0
        MOVE.W    D0,4(A0)
        ADD.W 8(A0),D0
        S2
        ASR.W #2,D0
        MOVE.W    D0,D6
        MOVE.W    C1CHAN,4(A0)
        MOVE.L    #CYC1,INTVEC
        MOVE.B    #$01,TSR
        RTE
```

```
.
* CYCLE 1
.
```

```
ORG    $2000
ANAL1: CLR.W D0
        MOVE.B    ATOD,D0
        MOVE.B    D6,DTOA1
        LEA.L    AADDR,A0
        MOVE.W    8(A0),D1
        SUB.W 24(A0),D1
        MOVE.W    D1,16(A0)
        MOVE.W    D1,C2CHAN
```

```
LEA.L      AADDR+20,A0
S2
MOVE.W     D0,4(A0)
SYNTH1:
LEA.L      SADDR,A0
MOVE.W     C2CHAN,8(A0)
MOVE.W     4(A0),D0
SUB.W      8(A0),D0
LEA.L      SADDR+16,A0
S1
ASR.W      #2,D0
MOVE.W     D0,D6
MOVE.L     #CYC0,INTVEC
MOVE.B     #S01,TSR
RTE
END

RESET INTERRUPT FLAG
RETURN
```

APPENDIX B

FOUR-BAND WAVE DIGITAL FILTER BANK

The following is a listing of the assembly language program developed for the Motorola MC68000 microprocessor which implements a 4-band polyphase wave digital filter bank. This program was assembled on a CP/M based cross assembler which is relies on the publication "MC68000 16-Bit Microprocessor User's Manual", Third Edition, edited by Karen Scoble and published by Prentice-Hall of Englewood Cliffs, New Jersey, 07632.

FLTR4 IDNT 1,0 FOUR-CHANNEL POLYPHASE WDF BANK

* DEFINE SYSTEM DEPENDENT ADDRESS LOCATIONS

TCR EQU \$00010021
TIVR EQU \$00010023
CPRH EQU \$00010027
CPRM EQU \$00010029
CPRL EQU \$0001002B
TSR EQU \$00010035
ACIA EQU \$00010042
ATOD EQU \$00030001
DTON1 EQU \$00020000
DTON2 EQU \$00020001
TOFF EQU \$A6
TON EQU \$A7
HI EQU \$00
MID EQU \$00
LO EQU \$01
INTVEC EQU \$100

* DEFINE MEMORY LOCATION LABELS

C3CHAN EQU \$3000
C4CHAN EQU \$3004
C5CHAN EQU \$3008
C6CHAN EQU \$300C

* ANALYZER MEMORY ALLOCATION

ORG \$3100
AADDR DS.L 100

* SYNTHESIZER MEMORY ALLOCATION

ORG \$3500
SADDR DS.L 100

ORG \$100
IVECTOR DC.L \$00001000

* MEMORY ALLOCATION FOR INTERMEDIATE VALUES

* TABLE OF VARIABLE/REG DEFN:

D0 - VAR A0 - MEM PTR
D1 - VAR A1 - N/U
D2 - VAR A2 - N/U
D3 - VAR A3 - N/U
D4 - VAR A4 - N/U
D5 - VAR A5 - N/U
D6 - O/P A6 - N/U
D7 - VAR A7 - STACK PTR

* ORIGIN

ORG \$0900

* MACRO DEFINITIONS

* S2 REFLECTANCE CALCULATION

S2 MACRO
 MOVE.W D0,D1
 MOVE.W (A0),D2
 SUB.W D2,D0
 ASR.W #\$01,D0
 ADD.W D0,D1
 ADD.W D2,D0
 MOVE.W D1,(A0)
 ENDM

* S1 REFLECTANCE CALCUALTION

S1 MACRO
 MOVE.W D0,D1
 MOVE.W (A0),D2
 MOVE.W 4(A0),D3
 SUB.W D2,D0
 MOVE.W D0,D4
 ASR.W #\$02,D4
 ADD.W D4,D0
 ASR.W #\$03,D0
 ADD.W D0,D1
 ADD.W D2,D0
 MOVE.W D1,(A0)
 ADD.W D3,D0
 MOVE.W D0,D1
 ASR.W #\$02,D0
 ADD.W D1,D0
 ASR.W #\$02,D0
 ADD.W D1,D0
 ASR.W #\$03,D0
 SUB.W D3,D0
 MOVE.W D0,4(A0)
 SUB.W D0,D1
 MOVE.W D1,D0
 ENDM

* INITIALIZATION

- SET 68231 FOR EXTERNAL CLOCK INPUT

INIT:

 MOVE.B #TOFF,TCR
 MOVE.B #HI,CPRH
 MOVE.B #MID,CPRM
 MOVE.B #LO,CPRL
 MOVE.B #\$40,TIVR
 MOVE SR,D0
 ANDI #\$F8FF,D0
 OR #\$0100,D0
 MOVE D0,SR

START:

 MOVE.W #\$00,D0
 MOVE.W #\$00,D1
 MOVE.W #\$00,D2
 MOVE.W #\$00,D3

ZERO OUT INTERNAL REGISTERS

```
MOVE.W    #$00,D4
MOVE.W    #$00,D5
MOVE.W    #$00,D6
MOVE.W    #$00,D7
MOVE.W    D0,A0
MOVE.W    D0,A1
MOVE.W    D0,A2
MOVE.W    D0,A3
MOVE.W    D0,A4
MOVE.W    D0,A5
MOVE.W    D0,A6
MOVE.B    D0,ATOD
MOVE.B    #TON,TCR
```

```
LOOP:     BRA    LOOP        WAIT FOR INTERRUPT
```

```
*
* INTERRUPT ROUTINES
*
* CYCLE 0
*
```

```
ORG    $01000
```

```
ANAL0:
```

```
CLR.W D0
MOVE.B    ATOD,D0
MOVE.B    D6,DTOA1
LEA.L    AADDR,A0
S1
MOVE.W    D0,8(A0)
ADD.W 24(A0),D0
LEA.L    AADDR+28,A0
S1
MOVE.W    D0,8(A0)
ADD.W 24(A0),D0
MOVE.W    D0,C3CHAN
```

```
SYNTH0:
```

```
LEA.L    SADDR,A0
MOVE.W    8(A0),D0
ADD.W 12(A0),D0
S2
ASR.W #02,D0
MOVE.W    D0,D6
MOVE.W    16(A0),D0
ADD.W 20(A0),D0
LEA.L    SADDR+4,A0
S2
MOVE.W    D0,4(A0)
MOVE.W    C3CHAN,12(A0)
MOVE.L    #ANAL1,INTVEC
MOVE.B    #S01,TSR
RTE
```

```
*
* CYCLE 1
*
```

```
ANAL1:
```

```
CLR.W D5
MOVE.B    ATOD,D5
MOVE.B    D6,DTOA1
LEA.L    AADDR,A0
MOVE.W    8(A0),D0
```



```
SUB.W 24(A0),D0
LEA.L      AADDR+56,A0
S1
MOVE.W     D0,8(A0)
MOVE.W     D5,D0
LEA.L      AADDR+20,A0
S2
MOVE.W     D0,4(A0)
MOVE.W     16(A0),D0
SUB.W 32(A0),D0
MOVE.W     D0,C4CHAN
```

SYNTH1:

```
LEA.L      SADDR,A0
MOVE.W     24(A0),D0
ADD.W 28(A0),D0
LEA.L      SADDR+32,A0
S2
MOVE.W     D0,-20(A0)
MOVE.W     -24(A0),D1
SUB.W D0,D1
MOVE.W     D1,D0
LEA.L      SADDR+36,A0
S1
ASR.W #02,D0
MOVE.W     D0,D6
MOVE.W     C4CHAN,-16(A0)
MOVE.L     #ANAL2,INTVEC
MOVE.B     #S01,TSR
RTE
```

•
• CYCLE 2
•

ANAL2:

```
CLR.W D0
MOVE.B     ATOD,D0
MOVE.B     D6,D0A1
LEA.L      AADDR,A0
S1
MOVE.W     D0,8(A0)
ADD.W 24(A0),D0
LEA.L      AADDR+48,A0
S2
MOVE.W     D0,4(A0)
MOVE.W     16(A0),D0
SUB.W 24(A0),D0
MOVE.W     D0,C5CHAN
```

SYNTH2:

```
LEA.L      SADDR,A0
MOVE.W     8(A0),D0
ADD.W 12(A0),D0
S2
ASR.W #02,D0
MOVE.W     D0,D6
MOVE.W     16(A0),D0
SUB.W 20(A0),D0
LEA.L      SADDR+48,A0
S1
MOVE.W     D0,-40(A0)
MOVE.W     C5CHAN,-24(A0)
```

MOVE.L #ANAL3,INTVEC
MOVE.B #01,TSR
RTE

•
• CYCLE 3
•

ANAL3:

CLR.W D5
MOVE.B ATOD,D5
MOVE.B D6,DTOA1
LEA.L AADDR+68,A0
MOVE.W -4(A0),D0
ADD.W 4(A0),D0
MOVE.W D0,C6CHAN
MOVE.W -60(A0),D0
SUB.W -44(A0),D0
S2
MOVE.W D0,4(A0)
MOVE.W D5,D0
LEA.L AADDR+20,A0
S2
MOVE.W D0,4(A0)

SYNTH3:

LEA.L SADDR,A0
MOVE.W C6CHAN,28(A0)
MOVE.W 28(A0),D0
SUB.W 24(A0),D0
LEA.L SADDR+56,A0
S1
MOVE.W D0,-44(A0)
MOVE.W D0,D1
MOVE.W -48(A0),D0
SUB.W D1,D0
LEA.L SADDR+36,A0
S1
ASR.W #02,D0
MOVE.W D0,D6
MOVE.L #ANAL0,INTVEC
MOVE.B #01,TSR
RTE
END

LKC
 P91 .C654 M42 1986
 Application of wave digital
 filter structures to
 sub-band voice coder design

P
 91
 C654
 M369
 1986

DATE DUE
 DATE DE RETOUR

CRC LIBRARY/BIBLIOTHEQUE CRC
 P91.C654 M369 1986

INDUSTRY CANADA / INDUSTRIE CANADA



208212



cover design: Diane Weselake, Instructional Media Centre