A DIGITAL PCM FILTER

by

O. Monkewich, B.Eng.

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Applied Science

Department of Electrical Engineering
University of Ottawa
Ottawa, Canada

July 1974

ABSTRACT

A stored-product technique is proposed as a design method for a digital PCM-channel filter. In place of multipliers the filter uses ROM blocks to store the coefficient-signal products. A flexible quantization algorithm is introduced into the processor to keep the number of stored products at an acceptable level. Details of the PCM filter design are worked out to meet the performance and cost requirements for a 24-channel system. Results of computer simulation and error analysis, based on a sinusoidal input with varying amplitudes are given. Hardware implementation compatible with the low-speed MOS technology is described.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. W. Steenaart, Research Director of this project, for his help and advice.

The author also gratefully acknowledges the funding of this project by the Department 1E00 of Bell-Northern Research.

Special appreciation is extended to Margareth James and Carol Mathieu for the typing of the thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 General

In general, every digital filter implementation is designed to perform some sequence of binary multiplications and additions conforming to well defined accuracy considerations. The arithmetic accuracy of a particular filter is characterized by the binary register lengths in which the final and the intermediate results of the operations are stored within the filter. Although it is possible to trade off cost and processing speed for increased accuracy, the number of possible numerical values resulting from the binary operations within the filter must necessarily remain finite.

The sums and products are generated by hardware mechanizations which, within the designed accuracy, are capable of adding or multiplying any two numbers of specified bit length. In a given filter design, many sums and products which fall within the bit length of the processor either never occur or have little effect on the response. In this sense most digital processors are too general and inherently slow.

The following question arises. Is it possible to store all or a part of the arithmetic results, which are certain to occur in a given filter, and access them in some desired sequence? In most applications this would reduce the processing time by a factor greater than 10.

The multiplier is by far the slowest component in any digital filter mechanization. In order to multiply an n-bit number by an m-bit number ($m<n$) using a conventional multiplication scheme, it is necessary to

perform m shifts and m parallel additions requiring not fewer than 2m clock intervals. In addition, some round off algorithm must be implemented before the product is completed. Such an algorithm performs the same round off operation on all products. In contrast, only one clock period is required to retrieve a product from storage and separate round off procedures can be applied to each product.

The effectiveness of a product-storing scheme depends on the amount of storage required to perform the filtering function. In the majority of applications the binary word lengths vary from eight to eighteen binary digits. Thus the number of possible products which may result from the multiplication operation may vary from 256 to 262,144. actual number of bits of storage requires is then 1.6 K-bits to some 5

In some applications it is not necessary to store all 2^n products in a basically n-bit processor. For example, in a typical PCM system the input signal to the filter can assume at most 256 distinct values, normally represented in an 8-bit nonlinear code. The typical dynamic range and resolution, however, require a 13-bit linear code representat All arithmetic operations must be done, of course, in the linear code, however, there will be only 256 13-bit products generated amounting to 3328 bits of storage per multiplier.

When two such products are added the result may not belong to a small set of 256 values. Thus if further multiplications are to be performed, as in recursive filters, some form of quantization must be introduced beforehand. The choice of the type of quantization would depend on the error sensitivity of the filter and the amount of storage that can be tolerated.

1.2 Summary

This work deals with three main aspects: 1) the use of Read Only Memory (ROM) to replace all multipliers in a filter mechanizatio 2) the design of a flexible and fast quantization algorithm for use as a part of the digital processor, 3) the application of these techniques to the practial problem of PCM channel filters.

Chapter II outlines the essential features of a Pulse Code Modulation system relating to channel filtering of voice frequency signals. A method of introducing a digital filter to perform the main filtering function is described. A number of design features are proposed as possible solutions to the problem of speed and accuracy. One of these features is the use of ROM in conjunction with a quantiza algorithm to replace the conventional multipliers.

In Chapter III a discussion of the specific quantization needs is followed by a formulation of a quantization algorithm suitable for the proposed application. Some relations on worst-case error and most likely error are derived for the algorithm. Finally, a hardware implementation of the quantization algorithm is described.

Chapter IV is devoted to the detailed design and analysis of the digital PCM channel filter. All design aspects such as prototype transformation, stability, response, realization and hardware implementation are considered. The design is examined in the light of performance and cost of the existing analog filters.

The performance of the digital filter is simulated by a computer program which includes the effects of all error sources due to quantiza truncation and roundoff. A final combination of digital filter and rec

order analog filter is shown to satisfy the insertion loss requirements over approximately 92% of the dynamic range.

Chapter V describes the method of simulation and analysis used in the preceding chapters. The listings and description of the software developed for this purpose are included.

CHAPTER II

FUNDAMENTALS OF PROPOSED DESIGN

2.1 General

The first part of the chapter gives a brief description of the Pulse Code Modulation system functions relevant to the proposed digital filtering problem. The nature of the filtering function required by the system and the extent to which it can be done by a digital filter are discussed in broad terms.

The remainder of the chapter describes the proposed method of introducing a digital filter into the system and the associated problems relating to processing speed and cost. Finally, the solutions to each of the design problems to be presented in detail in the later chapters are stated and explained.

2.2 Some System Considerations

A Pulse Code Modulation (PCM) system is a digital voice communication system consisting of a transmit channel bank and a receive channel bank. At the transmit channel bank each voice signal is band limited by a low pass filter, sampled, quantized and coded; the coded words are then transmitted as a digital bit stream. At the receive channel bank the code words are decoded into Pulse Amplitude Modulation (PAM) pulses which are fed to a low pass filter to reconstruct the voice signal.[1] Fig. 2.1 shows a simplified block diagram of a PCM system.

PCM channel banks usually consist of 24 voice channels which are sampled at the rate of 8 KHz. In order to achieve low redundancy the quantized values of PAM pulses are coded according to a nonlinear scale. This

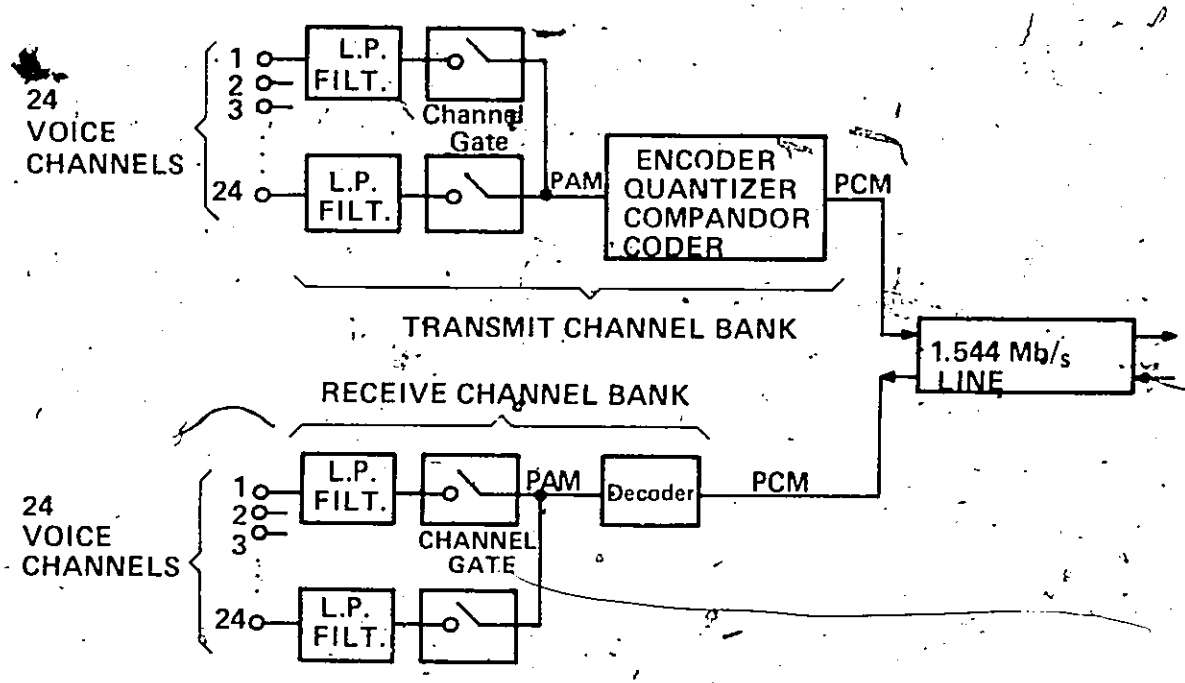


FIG. 2.1 — Simplified block diagram of a PCM system

nonlinear scale is usually a piecewise linear approximation of a logarithmic curve. A practical approximation is the so called μ-law which consists of a total of 15 linear segments each subdivided into 16 equal quantization intervals.[2] This requires 3 code bits to specify the positive segment number and 4 code bits to specify the quantization level within the segment. Together with the sign bit each code word of the nonlinear or compressed code is thus made up of 8 bits.

An 8-bit code can represent at most 2^8 or 256 different values, although the dynamic range of these values on the linear scale can be made arbitrarily large. In order to achieve good resolution over the desired dynamic range a 13-bit equivalent linear code is used. The 13-bit linear code, however, does not exist anywhere in the system; the encoder codes the quantized PAM signals directly into the 8-bit compressed code. Transmission of the 8-bit per sample code for 24 channels sampled at 8 KHz determines the 1.544 Mb/s rate of the 24-channel PCM system.[3]

Before a voice frequency signal is sampled it is band limited to 3400 Hz by a low pass filter. The voice band of interest extends from 300 to 3400 Hz. A given system, therefore, uses 48 low pass analog filters, 24 on the transmit side and 24 on the receive side. In a typical case the channel filtering requirements can be met by a fifth order elliptic filter. It is proposed that the function of these filters be performed, at least for the larger part, by a digital filter.

To make use of the A/D and D/A conversion already available in the system the digital filtering on the transmit side must be done after the

encoding operation. Similarly, on the receive side, the digital filtering must be done prior to decoding. This arrangement is shown in Fig. 2.2.

Some degree of analog low pass filtering must remain in each channel to counter the periodicity of the digital filter pass band at frequencies beyond the half sampling frequency.

2.3 Practical Design Considerations

In developing the proposed digital filtering technique most of the thinking was guided by the filter performance requirements in the frequency domain and the cost in relation to the existing analog method.

The fifth order elliptic LC filter that satisfies the system requirements consists of two inductors and five capacitors which together with the cost of assembly and tuning establish the total cost of the filter. Only about one half of this cost can be considered towards the cost of the digital filter since some analog filtering must remain in the system. The hardware complexity of a digital filter is far greater than that of an analog filter of equal order and the corresponding cost is therefore much higher. It is therefore necessary to multiplex the digital filter with all twenty-four channels.

As a passive device the analog filter draws no power. A fifth or sixth order digital filter, on the other hand, built with high speed logic can draw several watts of power. A potential digital filter design must therefore be capable of being realized entirely in the relatively low speed and low power MOS technology.(4)

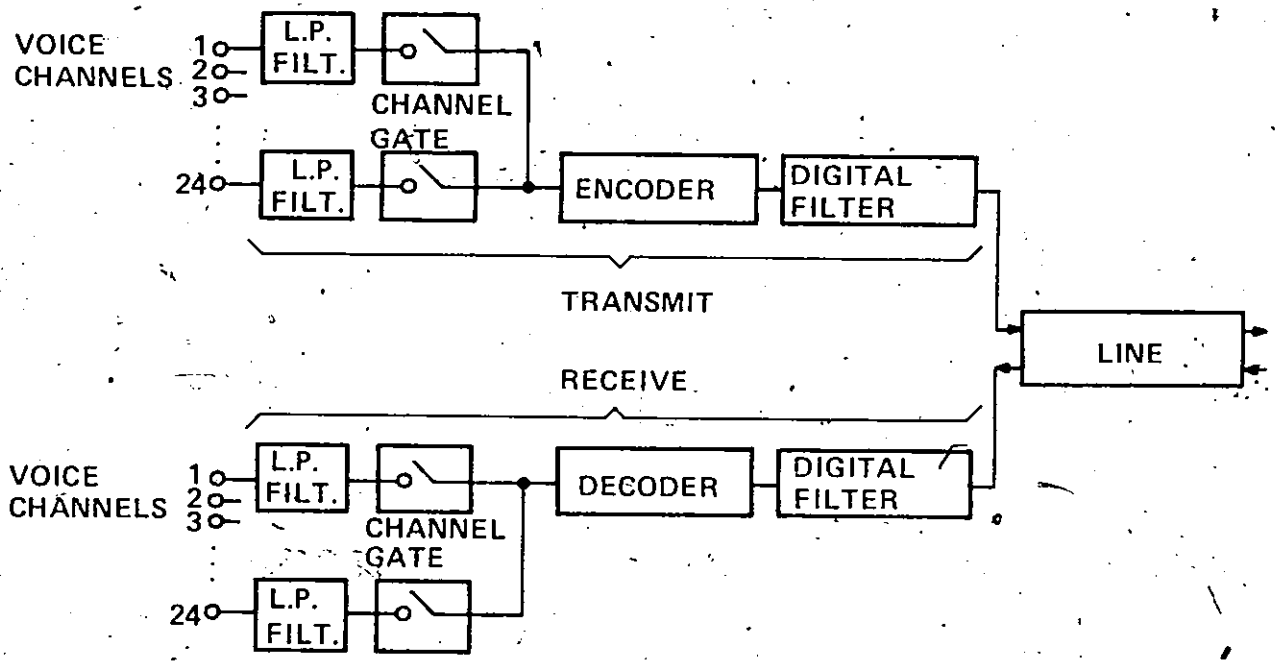


FIG. 2.2 — Digital filter in a PCM system

Some difficulties due to low clock rates can be easily observed. Consider the binary multiplication of two eight-bit numbers. A conventional multiplier must perform eight one-bit shifts and eight additions. Even with simultaneous shifting and parallel addition, requiring considerable hardware complexity, the operation requires eight clock periods to complete. With MOS clock rates of 5 MHz the operation requires 1.6 μsec. A fifth-order filter has a minimum of 10 such multiplications to perform. Disregarding all other operations the filter processing time becomes 16 μsec per channel. If the filter is to be multiplexed with 24 channels using 8 KHz sampling rate the available processing time is only 5.2 μsec. per channel. It will be shown in Chapter IV, however, that a sampling rate of 24 KHz must be used in order to achieve the required stop band characteristic; the time available for processing then reduces to roughly 1.7 μsec. per channel.

In addition to the usual operations of multiplication and addition within the digital filter a code conversion operation must be introduced. The A/D conversion of the analog signal in the system converts the coded signal into the 8-bit nonlinear code. Before arithmetic operations on the code words can be carried out the 8-bit code must be linearized to the 13-bit code. At the output of the digital filter the 13-bit code must be reconverted to the original 8-bit compressed code. The conventional digital methods available to perform this conversion require additional hardware and increase the total processing time of the filter.(5)

2.4 Proposed Design Method

The design aspects presented in the preceding section can be classified as four distinct problems:

1. Meeting the frequency domain performance requirements.
2. Meeting the cost requirements by multiplexing the filter with twenty-four channels.
3. Realizing the digital filter with MOS technology and,
4. Providing code compressions and expansion, without additional hardware or processing time.

Chapter IV is devoted entirely to the problem of meeting the frequency domain performance requirements. An approach to solving the other three problems will be stated in general terms in this section and the details will be worked out in the following chapters.

To reduce the filter processing time the conventional multipliers are replaced by Read Only Memory (ROM) units storing the products of all possible signal values and the multiplier coefficients. With 256 possible signal values and 13-bit products a 3328-bit ROM is required for each multiplier. Since any one of the 256 products can be uniquely addressed by an 8-bit code word no explicit code conversion from the 8-bit nonlinear code is necessary. Fig. 2.3 shows a series of multipliers corresponding to the coefficients A0, A1, A2, A3 and unit delay registers R1, R2, R3 and R4. In each case the input is the 8-bit compressed code representation of the input sample while the output is the 13-bit linear code representation of the corresponding product.

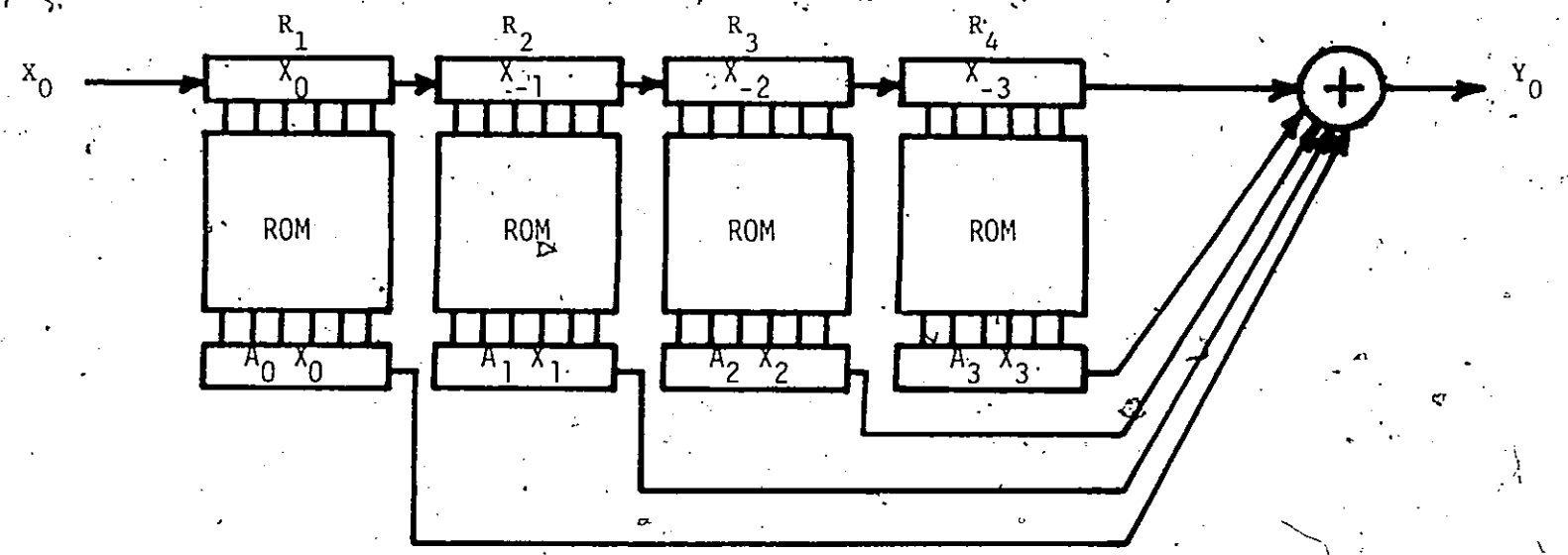


FIG. 2.3 — Multiplier Implementation Using Read Only Memory

This approach to generating products introduces a problem however. Following the addition of the products the resulting sum can take on a large number of values making it necessary to increase the ROM storage in the recursive section to an impractical level. To reduce the number of possible outputs from the adder a quantization stage is introduced as shown in Fig. 2.4.

The next step is to arrive at a digital quantization technique which would add very little to the processing time of the filter and at the same time offer some choice in the selection of the number and distribution of the quantized sums. The solution to this problem is presented in detail in Chapter III including the mechanization of the quantization algorithm.

To illustrate the length of the processing period for a given input sample consider a third order filter. The total processing time t_T can be subdivided into a number of major operation intervals as shown in Fig. 2.5. Define the following quantities.

t_T - total time required to process one sample

t_A - time required to add one product to the accumulator

t_M - time required to access a product in ROM

t_o - time required to convert the output to the 8-bit compressed code.

With a single adder the addition intervals must be added serially without overlapping. With individual ROMs for each coefficient the accessing of the products may be overlapped. The final quanitzed sum can be converted from the 13-bit linear code to the 8-bit compressed code by a table look up method during the interval t_o of the next processing cycle. It follows then that the total processing time can be expressed as

$$t_T = nt_A + t_o$$

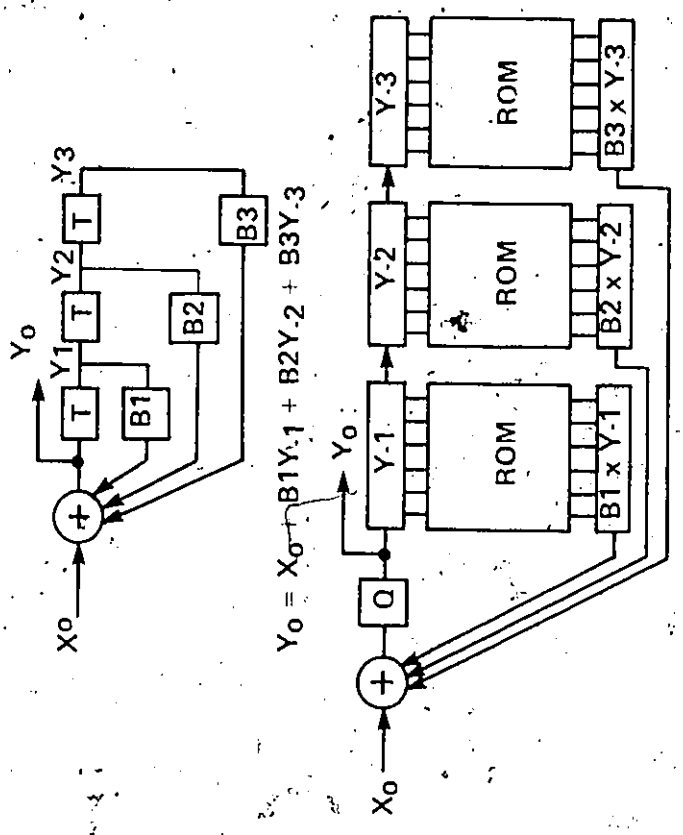
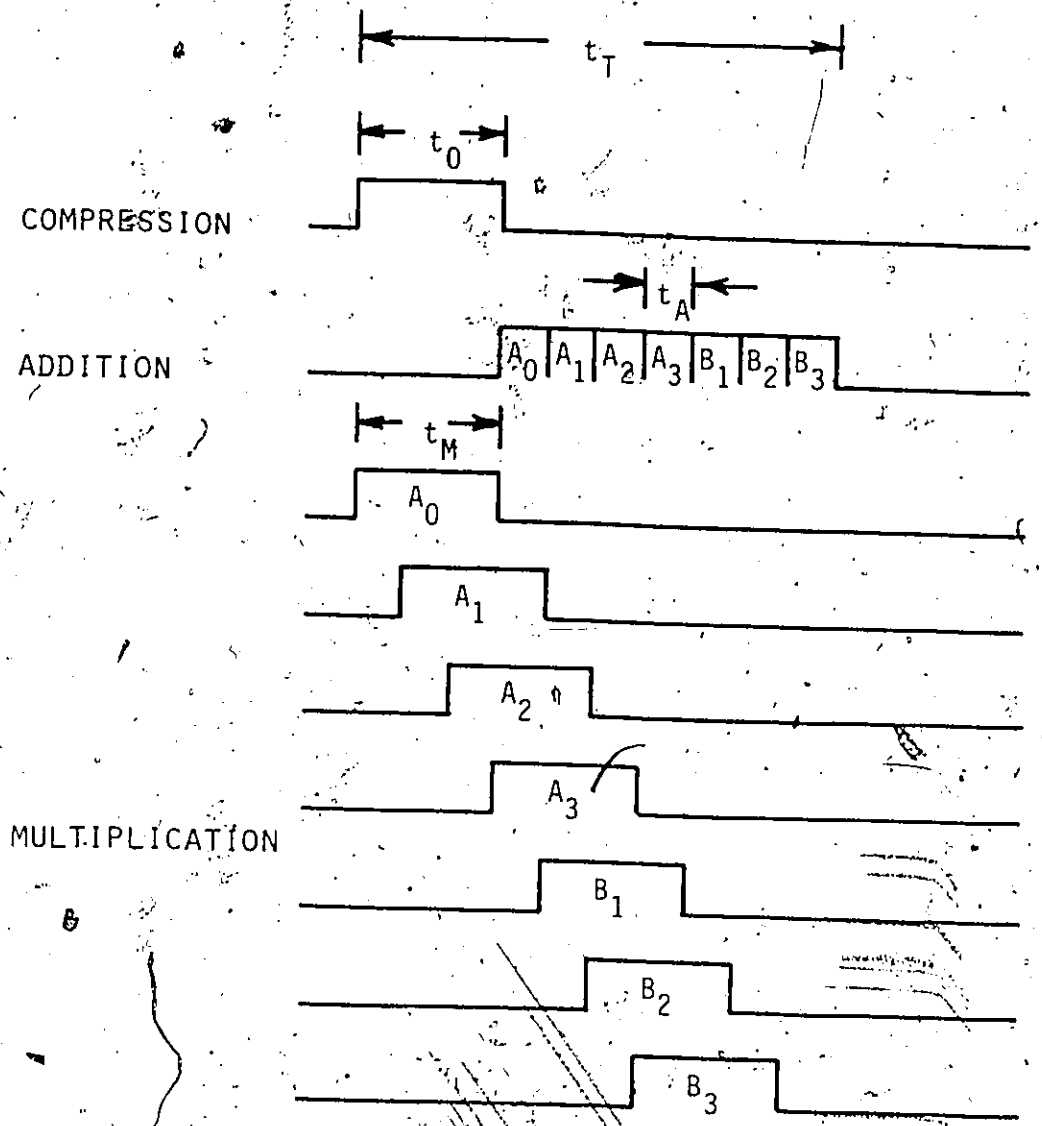


FIG. 2.4 — Quantization in the recursive section



$$t_T = nt_A + t_M$$

for t_A = 50 nsec

t_M = 150 nsec

$$t_T = 350 + 150 = 500 \text{ nsec}$$

FIG. 2.5 — Timing diagram for a third order filter section

TIME AVAILABLE TO THE DIGITAL FILTER TO PROCESS ONE SAMPLE

8 KHz sampling rate

$$\frac{1}{(8 \times 10^3)\ (24)} \approx 5.2\ \mu\text{sec}$$

24 KHz sampling rate

$$\frac{1}{(24 \times 10^3)\ (24)} \approx 1.7\ \mu\text{sec}$$

TIME REQUIRED TO PERFORM A MULTIPLICATION BY CONVENTIONAL MEA

13-bit word x 13-bit word

10 MHz clock rate

13 shifts	1.3 μsec.
13 additions	1.3 μsec.
	2.6 μsec.

FIG. 2.6 Processing Time per Sample and the Speed of Conventional Multiplication

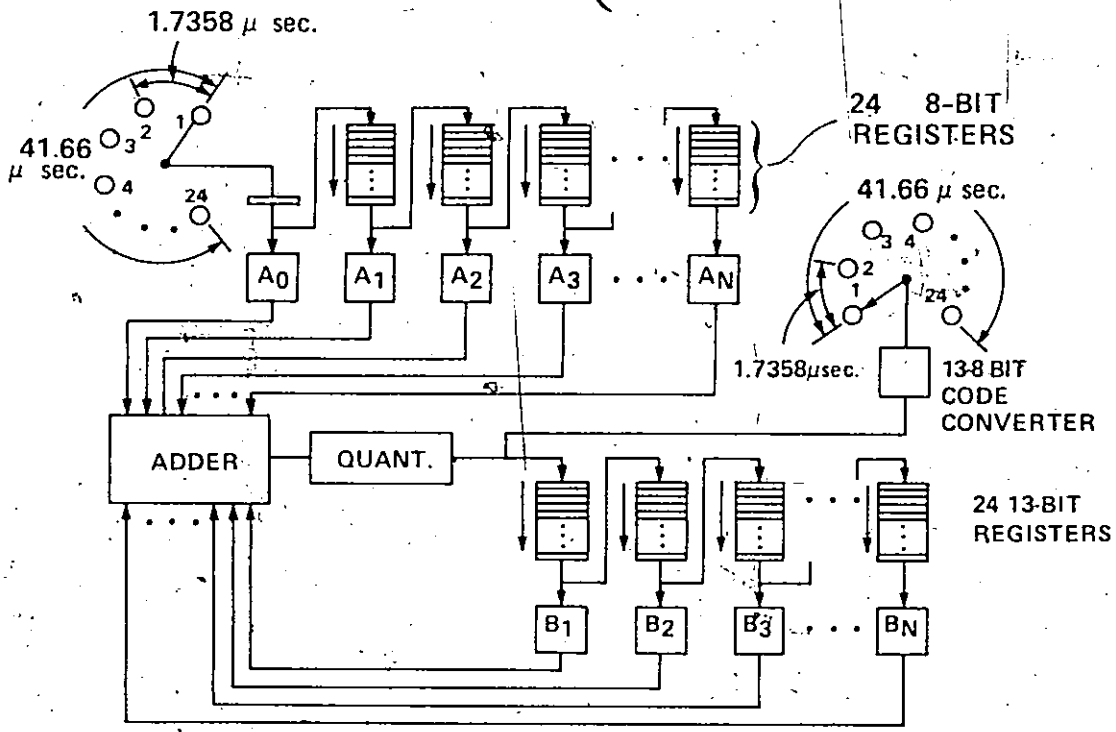


FIG. 2.7 — 24-channel multiplexing arrangement for an N-th order filter

where n is the number of additions during the cycle. Using some typical values

$$t_A = 50 \text{ nsec.}$$

$$t_M = 150 \text{ nsec.}$$

the total processing time works out to be 500 nsec and 1 μsec for a 6th order filter. The total processing time is therefore within the required limit of 1700 nsec. The time available to the digital filter to process one sample is computed in Fig. 2.6 for the 8 KHz and the 24 KHz sampling rates. Within this processing time a sixth order digital filter must perform fourteen multiplications. If conventional multiplication by means of successive shifting and adding is used the time required to complete one multiplication works out to 2.6 μsec for 10 MHz clock rate.

Finally, a multiplexing arrangement must be worked out. A variety of well established techniques is available. Fig. 2.6 is one possible method of multiplexing an N-th order filter with 24 channels.

CHAPTER III

QUANTIZATION

3.1 General

In the context of digital filtering, quantization refers to the inevitable process of rounding or truncation due to the finite register lengths encountered in all digital arithmetic processors. Such truncation and rounding results in uniform quantization with the quantization interval determined by the value of the least significant bit in a given register.[6] Preliminary investigation indicates that no specific effort has been made to introduce some of the quantization principles used in PCM. In this chapter some of these basic principles will be used to arrive at a useful quantization method which offers the flexibility of a tradeoff between the quantization error and the number of products stored.

If quantization is to be used as a functional operation in a digital processor, it is essential that its execution time be kept to a minimum. It will be shown that in this sense, the quantization scheme associated with the well known segmented μ-law is not the best suited. On the other hand, the proposed quantization method is basically a single step operation. That is, using Read Only Memory and a minimum number of address bits, any arbitrary number inside the prescribed dynamic range can be assigned its proper quantization level value within one clock interval.

The method makes use of a piece wise linear approximation of the continuous A-law companding characteristic and thus has a similar signal to quantization noise ratio over the dynamic range of the input signals.[7]

3.2 Necessary Properties for the Quantization Algorithm.

The quantization algorithm will be applied exclusively to numbers in binary form and the mechanization of the algorithm will be a digital logic circuit. Since processing speed is of prime importance the algorithm must be capable of recognizing an n-bit number as belonging to a particular quantization interval in a minimum number of logical steps. This will be referred to as the first property of the algorithm.

A second property of the algorithm must be that which permits the designer to choose between the various degrees of quantization accuracy and the corresponding hardware complexity.

The algorithm must also satisfy some established signal to quantization noise criteria or follow some proven companding characteristic such as the A-law or the μ-law. This will be referr to as the third property of the algorithm.

An algorithm will now be formulated and will be shown to posse all three of the above properties.

A signal voltage can assume both positive and negative values with respect to some reference voltage. We will first deal with th positive values and then generalize to include the negative values. Every system is designed to some dynamic range so that no voltage amplitude exceeds some upper value of V volts. On the other hand the dynamic range of a digital system is determined by the digital registers of some specified bit length N. Thus the largest number that can be represented by a sequence of N bits is 2^N and corresponc to V volts. All other signal voltages can then be represented by

numbers between 0 and 2^N in steps of 1 assuming that such voltages have previously been quantized to the nearest unity in accordance with some approximation criterion.

It is well known in PCM work that it is not necessary to quantize all amplitudes to the same quantization interval. The larger amplitudes can be rounded off more coarsely than the smaller amplitudes and at the same time preserve the same percentage error. Thus the various signal amplitudes are classified into several ranges and within each range a progressively larger quantization interval is assumed. It is therefore worthwhile to examine in detail the various ways in which the 2^N integers can be subdivided into ranges and subranges that would satisfy the quantizati needs and at the same time satisfy the three proposed properties.

Any one of the integers between 0 and r^N can be represented by the polynomial

$$A(r) = r^{n-1} + a_{n-2}r^{n-2} + \ldots + a_1 r + a_0 \qquad (1)$$

where r is the radix

or by the sequence

$$a_{n-1}a_{n-2}a_{n-3}\cdots a_1 a_0 \qquad (2)$$

where $a_i \in \{0, 1\}$, $r = 2$

and $0 \leq n \leq N$

Thus let $S = \{A(r) \mid \deg A(r) \leq N-1\}$

One simple classification of the polynomials A(r) is that accordin to the degree. Thus for A(r) of degree k the corresponding sequence would be

$$1\, a_{k-1}a_{k-2}\cdots a_1 a_0 \qquad (3)$$

The detection of such a class would be simple to implement since the sequence can be recognized by its leading coefficient 1.

So we partition S into N equivalence classes

$$S = \{S_0, S_1, S_2, \ldots, S_{N-1}\} \tag{4}$$

where $S_k = \{A(r) \in S_k \mid \deg A(r) = k\}$

Furthermore, $S_i \cap S_j = \phi$ for $i \neq j$

and $S_0 \cup S_1 \cup S_2 \cup \ldots \cup S_{N-1} = S$.

This partition induces an equivalence relation on S defined by

$$A_a(r) \; R \; A_b(r) \leftrightarrow A_a(r), A_b(r) \in \text{same } S_i$$

Any element of S_k is then represented by

$$A(r) = r^k + a_{k-1} r^{k-1} + a_{k-2} r^{k-2} + \ldots + a_1 r + a_0 \tag{5}$$

or equivalently by the sequence

$$1 \; a_{k-1} \; a_{k-2} \cdots a_1 \; a_0 \tag{6}$$

The integers corresponding to the same equivalence class S_k are now associated with a distinct companding segment k. That is the signal levels which fall within the range of the integers of the class S_k will be uniformly quantized.

The next step is to subdivide all integers corresponding to S_k into quantization intervals called quanta. With each quantum we can then associate a distinct quantization level numerically equal to the mid value of the quantum. When a signal assumes the value of one of the integers in the quantum the signal is assigned the value of the quantization level associated with that quantum.

Consider the sequence

$$1 \; a_{k-1} a_{k-2} a_{k-3} \cdots a_2 a_1 a_0 \tag{7}$$

corresponding to the class S_k. There are a total of 2^k such sequences.

We can thus subdivide S_k into 2^k quanta each containing one integer. If we ignore a_0 then for every such sequence we will have two equal sequences of the form

$$1\ a_{k-1}a_{k-2}\cdots a_2a_1 \tag{8}$$

one for $a_0=0$ and one for $a_0=1$. There are 2^{k-1} such pairs; each pair can be considered as a class or a quantum. If both a_0 and a_1 are ignored then S_k is subdivided into 2^{k-2} groups of sequences, each containing four equivalent sequences

$$\begin{aligned} &1\ a_{k-1}a_{k-2}\cdots a_3a_200\\ &1\ a_{k-1}a_{k-2}\cdots a_3a_201\\ &1\ a_{k-1}a_{k-2}\cdots a_3a_210\\ &1\ a_{k-1}a_{k-2}\cdots a_3a_211 \end{aligned} \tag{9}$$

all representable by the charcteristic sequence $a_{k-1}a_{k-2}\cdots a_3a_2$. The four sequences can be considered as a quantum with the quantum level corresponding to the mid value of $1\ a_{k-1}a_{k-2}\cdots a_3a_201\Delta 1$. If a signal amplitude assumes a value of one of the four integers represented by one of the four sequences the signal is assigned the amplitude value of

$$1\ a_{k-1}a_{k-2}\cdots a_3a_201\Delta 1 \tag{10}$$

In general S_k can be partitioned into 2^{k-m} equivalence classes each of order 2^m, $0\leq m\leq k$ by defining a relation $\sim$ on S_k by

$$A_i(r)\sim A_j(r)\leftrightarrow A_i(r) - A_m(r) = A_j(r) - A_m(r)$$

where $A_i(r)$, $A_j(r) \in S_k$ and i, j and m are the degrees of the respective polynomials. $\sim$ is an equivalence relation because from the definition of $\sim$: $A_i(r) \sim A_i(r)$, $A_i(r) \sim A_j(r) \rightarrow A_j(r) \sim A_i(r)$

and $A_i(r) - A_j(r)$, $A_j(r) - A_e(r) + A_i(r) - A_e(r)$. Hence, partitions S_k into disjoint equivalence classes whose union is all of S_k

$$S_k = \{Q_0, Q_1, Q_2, \ldots, Q_{m-1}\} \tag{11}$$

where $Q_i \cap Q_j = \emptyset$ for $i \neq j$

and $Q_0 \cup Q_1 \cup Q_2 \cup \ldots \cup Q_{m-1} = S_k$

The preceding partitioning of integers in the range 0 to 2^N is compatible with the first and second properties of the sought quantization algorithm. The first property is satisfied because the characteristic sequence associated with each quantum is the minimum number of consecutive bits required to specify all sequences within the quantum. The consecutive nature of the characteristic sequence is particularly important if the sequence is to be used as an address to access the quantization level value in ROM. The second property is satisfied because the quantization step can be selected to be any size in steps 2^m. This permits a trade off between the quantization accuracy and the size of address sequence which determines the number of quanta and the number of levels to be stored.

3.3 Quantization for PCM[8]

Consider two quanta Q_a and Q_b belonging to two successive segments S_k and S_{k+1}, that is, $Q_a \in S_k$ and $Q_b \in S_{k+1}$. Let $o(Q_a)$ and $o(Q_b)$ represent the orders of Q_a and Q_b respectively. From the preceding section it is clear that $\frac{o(Q_b)}{o(Q_a)} = 2^\alpha$ which $\alpha = 0, 1, 2, \ldots$ Thus the quanta of the successively higher segments can be made successively larger

by a proper choice of α. In PCM it is customary to double the size of the quanta belonging to the successive segments and to choose the number of quanta per segment to be 16. Thus consider $\frac{o(Q_b)}{o(Q_a)} = 2$ and m=4. The dynamic range is usually chosen to correspond to a 13-bit linear code in which the 13th bit is used as the sign bit. This defines the input integer range of -4096 to +4906 with N=12.

The sequences corresponding to each segment can be written as

$$
\begin{array}{ll}
k=0 & 0 \\
k=0 & 1 \\
k=1 & 1\ a_0 \\
k=2 & 1\ a_1 a_0 \\
k=3 & 1\ a_2 a_1 a_0 \\
k=4 & 1\ a_3 a_2 a_1 a_0 \\
k=5 & 1\ a_4 a_3 a_2 a_1 a_0 \\
k=6 & 1\ a_5 a_4 a_3 a_2 a_1 a_0 \\
k=7 & 1\ a_6 a_5 a_4 a_3 a_2 a_1 a_0 \\
k=8 & 1\ a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 \\
k=9 & 1\ a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 \\
k=10 & 1\ a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0
\end{array}
\qquad (12)
$$

If we combine the k=0, k=1, k=2 and k=3 into one segment with all other cases representing distinct segments we will obtain what is known as the A-law segment companding characteristic.

Figure 3.1 shows the first six positive segments of the resulting companding law. Each division shown on the X and Y axes corresponds to the partitioning of integers according to segments.

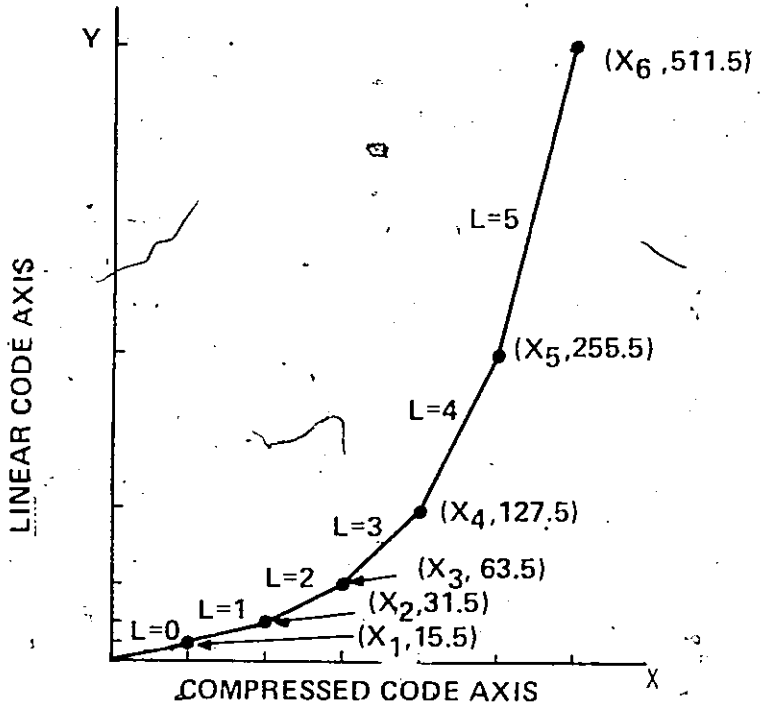


FIG. 3.1 — First six positive segments of the companding law

There are eight such segments in PCM. Each such division is again subdivided into 16 equal quanta. For L>1 any quantization level is defined as being numerically equal to the mid value of its quantum. For L=0 and L=1 each quantum contains one integer hence each quantization level is defined as numerically equal to that integer.

Let q designate the number of a quantum within a segment so that q=0, 1, 2, ... 15 for each segment. It is clear that a quantization level is given by

$$Q(L) = q \quad \text{for} \quad L=0 \tag{13}$$

and

$$Q(L) = q + 16 \text{ for } L=1 \tag{14}$$

Let $\delta(L)$ represent the number of integers in a given quantum then by the assumption that $\frac{\delta(L+1)}{\delta(L)} = 2$ we have

$$\delta(L) = 1 \text{ for } L=0 \tag{15}$$

and

$$\delta(L) = 2^{L-1} \text{ for } L\geq 1 \tag{16}$$

$\delta(L)$ also represents the spacing between quantization levels within a segment except at the segment edges where the spacing is given by

$$\frac{\delta(L)}{2} + \frac{\delta(L+1)}{2}$$

So $Q(L) = Q_{min}(L) + q + \delta(L)$

where $Q_{min}(L) = Q_{min}(1)\delta(L) + \frac{\delta(L)}{2} + \frac{\delta(L)}{2} - 0.5$

$$= \delta(L)\,|16+0.5| - 0.5$$

$$= 2^{L-1}\,(16.5) - 0.5$$

$$Q(L) = 2^{L-1}\,(16.5) - 0.5 + q2^{L-1}$$

$$Q(L) = 2^{L-1}\,(q+16.5) - 0.5 \tag{17}$$

for $L\geq 1$

Using these relations all quantization level values can be generated. Table 3.1 gives the tabulation for eight positive segments.

TABLE 3.1

	L							
q	0	1	2	3	4	5	6	7
0	0	16	32.5	65.5	131.5	263.5	527.5	1055.5
1	1	17	34.5	69.5	139.5	279.5	559.5	1119.5
2	2	18	36.5	73.5	147.5	295.5	591.5	1183.5
3	3	19	38.5	77.5	155.5	311.5	623.5	1247.5
4	4	20	40.5	81.5	163.5	327.5	655.5	1311.5
5	5	21	42.5	85.5	171.5	343.5	687.5	1375.5
6	6	22	44.5	89.5	179.5	359.5	719.5	1439.5
7	7	23	46.5	93.5	187.5	375.5	751.5	1503.5
8	8	24	48.5	97.5	195.5	391.5	783.5	1567.5
9	9	25	50.5	101.5	203.5	407.5	815.5	1631.5
10	10	26	52.5	105.5	211.5	423.5	847.5	1695.5
11	11	27	54.5	109.5	219.5	439.5	879.5	1759.5
12	12	28	56.5	113.5	227.5	455.5	911.5	1823.5
13	13	29	58.5	117.5	235.5	471.5	943.5	1887.5
14	14	30	60.5	121.5	243.5	487.5	975.5	1951.5
15	15	31	62.5	125.5	251.5	503.5	1007.5	2015.5
$\delta(L)$	1	1	2	4	8	16	32	64

3.4 Quantization Error Bounds

Figure 3.2 shows the quantization operation as it can be implemented in the proposed digital filter. Without quantization y_n is given by

$$y_n = B_1 y_{n-1} + B_2 y_{n-2} + \ldots + B_m y_{n-m} \tag{18}$$

With quantization

$$y_n' = B_1 y_{n-1}' + B_2 y_{n-2}' + \ldots + B_m y_{n-m}' \tag{19}$$

where y_{n-1}', y_{n-2}', ..., y_{n-m}' are the quantized versions of y_{n-1}, y_{n-2}, ..., y_{n-m}. The fractional error is then given by

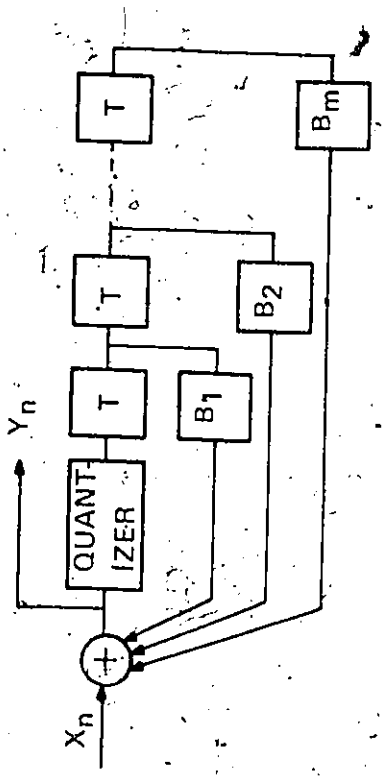


FIG. 3.2 — Quantization in the recursive section

$$\frac{y_n-y'_n}{y'_n} = A_1 \frac{y_{n-1}-y'_{n-1}}{y'_{n-1}} + A_2 \frac{y_{n-2}-y'_{n-2}}{y'_{n-2}} + \ldots + A_m \frac{y_{n-m}-y'_{n-m}}{y'_{n-m}} \quad (20)$$

In Figure 3.3 L_a and L_b are two adjacent segments and q_{ka} and q_{kb} are two corresponding quantization levels in segment L_a and L_b respectively. The fractional error at a given quantization level is

$$e = \frac{q_k-q_{k-1}}{2q_k} \quad (21)$$

Since q_k-q_{k-1} is constant for all k within the same segment we can write

$$e_{(max)a} = \frac{q_k-q_{k-1}}{2q_{oa}} \quad (22)$$

$$e_{(min)a} = \frac{q_k-q_{k-1}}{2q_{15a}} \quad (23)$$

and

$$e_{(avg)a} = \frac{q_k-q_{k-1}}{2q_{8a}} \quad (24)$$

Furthermore, all maximum errors are equal

$$\frac{q_{ka}-q_{(k-1)a}}{2q_{oa}} = \frac{q_{kb}-q_{(k-1)b}}{2q_{ob}} = \frac{q_{kc}-q_{(k-1)c}}{2q_{oc}} = \ldots \quad (25)$$

The same holds for the minimum and the average errors.

If y_{n-i} falls within the kth quantum and $\frac{y_{n-1}-y'_{n-i}}{y'_{n-i}}$ is assumed to be maximum for all i then

$$\frac{y_{n-1}-y'_{n-1}}{y'_{n-1}} = \frac{y_{n-2}-y'_{n-2}}{y'_{n-2}} = \ldots = \frac{y_{n-m}-y'_{n-m}}{y'_{n-m}} = \frac{q_k-q_{k-1}}{2q_o} \quad (26)$$

It follows then that

$$\left(\frac{y_n-y'_n}{y'_n}\right) = (B_1+B_2+\ldots+B_m)\frac{q_k-q_{k-1}}{2q_o} \quad (27)$$

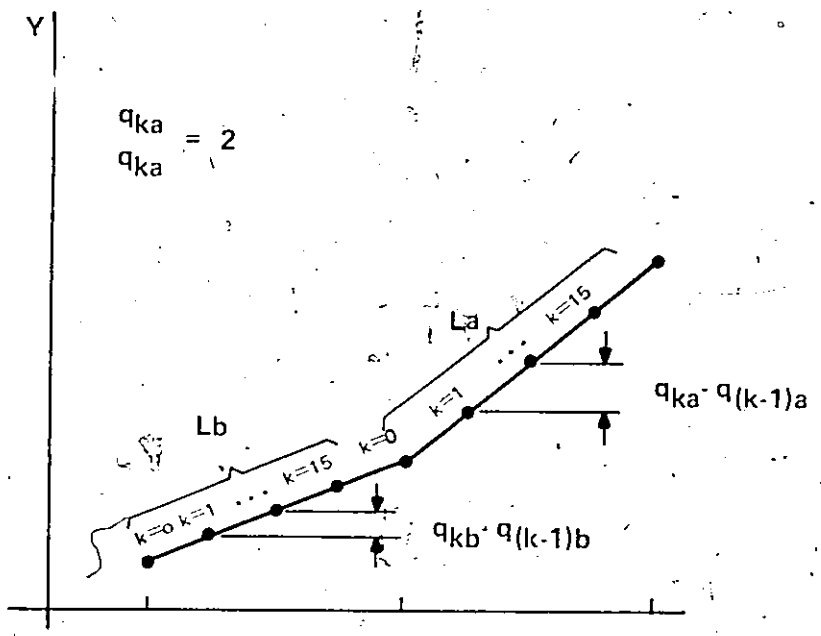


FIG. 3.3 — Relation of quanta in adjacent segments

$$\left(\frac{y_n-y'_n}{y'_n}\right)_{avg} = (B_1+B_2+\ldots+B_m)\,\frac{q_k-q_{k-1}}{2q_8} \tag{28}$$

and

$$\left(\frac{y_n-y'_n}{y'_n}\right)_{min} = (B_1+B_2+\ldots+B_m)\,\frac{q_k-q_{k-1}}{2q_{15}} \tag{29}$$

Specifically for the A-law level assignment the maximum fractional error can be found as follows

$$\frac{\frac{q_k-q_{k-1}}{2.} - 0.5}{q_0} = \frac{8-0.5}{263.5} = 0.0246 \tag{30}$$

$$\left(\frac{y_n-y'_n}{y'_n}\right)_{max} = 0.0246\,(B_1+B_2+\ldots+B_m) \tag{31}$$

3.5 Further Quantization Error Considerations

All positive signal amplitudes between 0 and 2^N can be represented by a continuous variable x. If x is the analog value of the speech amplitude then the probability density function of x is given by

$$p(x) = \frac{1}{\sqrt{2}\,\sigma}\, e^{-\sqrt{2}\,\frac{|x|}{\sigma}} \tag{32}$$

where σ is the RMS speech voltage.(9)

If a finite number of binary digits are used to represent the values of x then only a finite number of values of x spaced by some δ can be represented. Let x_i be any one of these values of x then the probability that the voice signal amplitude x will fall in the interval $x_i-\delta/2 \leq x \leq x_i+\delta/2$ is given by

$$p(x_i) = \int_{x_i-\delta/2}^{x_i+\delta/2} p(x)\,dx \tag{33}$$

Here δ is the maximum value representable by the least significant digit in the finite-length register and is equal to 2^a where "a" is the position of the least significant digit relative to the binary point.

If δ is small relative to 2^N the $p(x)$ can be assumed to be constant over the range $x_i-\delta/2$ to $x_i+\delta/2$. If we let $p(x)=p_i$ in that range then

$$p(x_i) = \int_{x_i-\delta/2}^{x_i+\delta/2} p_i\,dx = p_i\delta \tag{34}$$

If e_i is the instantaneous error $x-x_i$ then the mean squared error can be written as [2]

$$\overline{e_i^2} = \int_{x_i-\delta/2}^{x_i+\delta/2} (x-x_i)^2\, p_i\,dx$$

$$= \frac{\delta^3}{12}\, p_i \tag{35}$$

If we assume $\delta=1$ then $|x_{i+1}-x_i| = 1$ and the entire positive signal amplitude range can be expressed in terms of integers.

There are 2^{L+3} integers in each segment of Fig. 3.1. Hence each quantum contains

$$\frac{2^{L+3}}{2^m} = \frac{2^{L+3}}{2^4} = 2^{L-1} \text{ integers.} \tag{36}$$

If x_i is any integer between 0 and 2^N then

$$2^{L+3} \leq x_i \leq 2^{L+3} + \sum_{j=0}^{L+2} 2^j \tag{37}$$

for L=1, 2, 3, ..., 7

and

$$0 \leq x_i \leq 2^{L+3} \sum_{j=0}^{L+2} 2^j \tag{38}$$

for L=0.

The probability of the signal falling within any one quantum is then

$$\sum_{i=0}^{2^{L-1}} p(x_i) = \sum_{i=0}^{2^{L-1}} \int_{x_i - /2}^{x_i + /2} p(x)\, dx \tag{39}$$

$$= \sum_{i=0}^{2^{L-1}} \delta_i p_i$$

$$= \delta \sum_{i=0}^{2} p_i \qquad \delta_i = \delta \text{ for all } i \tag{40}$$

$$= \delta P_j \qquad P_j = \sum_{i=0}^{2^{L-1}} p_i \tag{41}$$

The smallest integer in the jth quantum is

$$m_j = 2^{L+3} + 2^{L-1} \cdot q \tag{42}$$

The largest integer in the same quantum is

$$M_j = m_j + 2^{L-1} - 1 \tag{43}$$

If x_i falls inside the jth quantum then the error due to quantization is given by

$$e_j = Q_j(L) - x_i \tag{44}$$

and the mean squared error is

$$\overline{e_j^2} = \sum_{i=m_j}^{M_j} |Q_j(L) - x_i|^2 P_j \tag{45}$$

3.6 Hardware Realization of the Quantization Algorithm

Figure 3.4 shows the hardware mechanization of the quantization algorithm. The number to be quantized is deposited in its binary form into the register designated AC. Gates G4 and G5 detect the number of leading zeros and the position of the most significant non-zero binary digit. The outputs of these gates enable the inputs to the ROM which stores the quantized value of the number being processed. The inputs to the remaining ROMs are simultaneously disabled by the same gates G4 and G5. The arrival of a clock pulse at the ROM input produces the 13-bit quantized result.

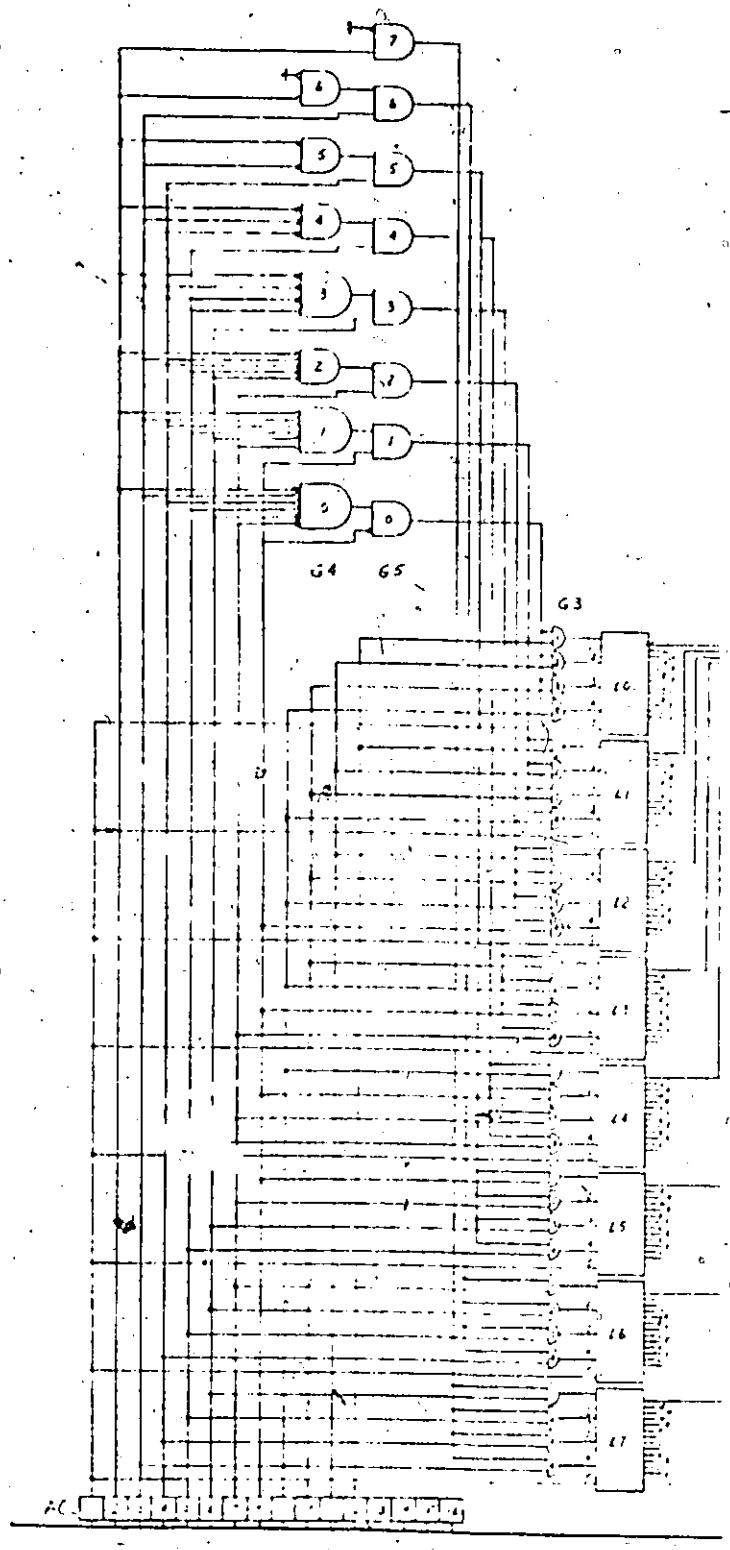


FIG. 3.4 — Mechanization of the quantization algorithm

CHAPTER IV

A DIGITAL PCM CHANNEL FILTER DESIGN

4.1 General

In this chapter the techniques of the preceding two chapters will be applied to the design of a digital PCM channel filter. The design will center on the transmit filter which will later be extended to the receive filter. The main objective will be to achieve the insertion loss performance of the existing analog filters at a competitive cost per channel.

Fifth and sixth order digital filters are examined in combination with various designs of second and third order analog filters. The effect of using the sampling rates of 16 KHz, 24 KHz and 32 KHz is also considered.

The evaluation of each case is performed on the basis of arithmetic simulation of the processors with all sources of error due to quantization, truncation and roundoff properly accounted for. The dependence on signal amplitude is also examined. All analyses are based on the sampled sinusoidal signal input. A number of design, analysis and simulation programs are written in Fortran IV for this purpose.

Of the cases considered, the most effective design combination is the third order analog polynomial filter with a third order recursive digital filter. In this arrangement each sample is made to pass through the digital filter twice, making the filter effectively a sixth order filter.

4.2 Existing Analog PCM Transmit Filter

Figure 4.1a shows the schematic diagram of the fifth order analog filter which satisfies the requirements of a channel filter in a typical PCM system.[10] Figure 4.1b shows the insertion loss characteristic of the filter and the corresponding specification contours set by the system requirements.

The p-plane transfer function of this filter is given by

$$H(p) = K \frac{(p^2+\omega_1^2)(p^2+\omega_2^2)}{(p-p_1)(p-p_2)(p-p_2^*)(p-p_3)(p-p_3^*)} \qquad (1)$$

where ω_1 and ω_2 are the attenuation pole frequencies in radians/sec., p_1 is the real attenuation zero on the negative real axis, p_2 and p_3 are complex attenuation zeros and p_2^* and p_3^* are their complex conjugates all expressed in radians per second. For the particular design of Figure 4.1a we have

$\omega_1 = 1.6169977$

$\omega_2 = 2.4377100$

$p_1 = -1.095684287$

$p_2 = -0.1608781146 - j\ 1.196356047$

$p_2^* = -0.1608781146 + j\ 1.196356047$

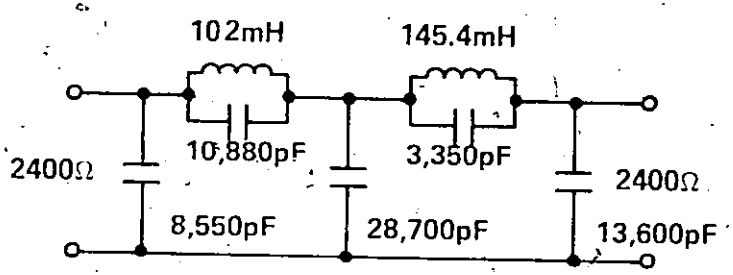
$p_3 = -0.6399974725 - j\ 0.9633257457$

$p_3^* = -0.639974725 - j\ 0.9633257457$

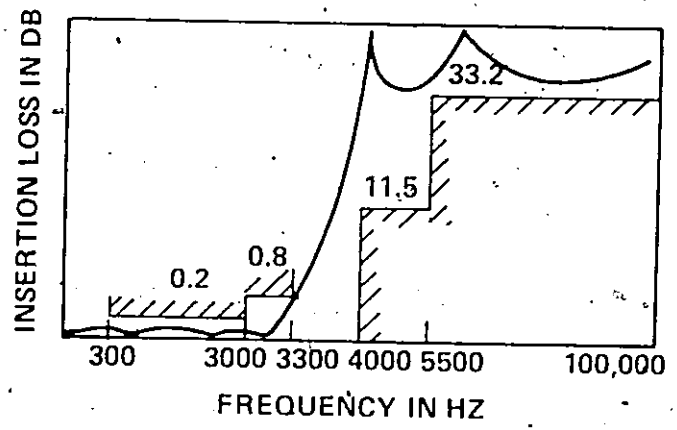
These are normalized with respect to the pass band edge at 2960 Hz.

The insertion loss of the filter is given by

$$IL(dB) = 20 \log_{10} \left| \frac{1}{H(p)} \right|_{p=j\omega} \qquad (2)$$



(a)



(b)

FIG. 4.1 — Fifth order analog PCM channel filter and response

The constant K can be determined from the relation

$$|H(p)|_{p=0} = 1 \tag{3}$$

4.3 The Bilinear Transform

All designs considered in this chapter are based on the bilinear transform method. This makes it possible to make use of the well established p-plane approximation and realization routines already available.[11]

If in the expression for H(p) we let $p = \frac{2}{T}\frac{Z-1}{Z+1}$, where T is a constant and Z a complex variable, then after some manipulation we can get

$$H(Z) = \frac{K_5Z^5+K_4Z^4+K_3Z^3+K_2Z^2+K_1Z+K_0}{L_5Z^5+L_4Z^4+L_3Z^3+L_2Z^2+L_1Z+L_0} \tag{4}$$

H(Z) is then the Z-domain transfer function of the corresponding digital filter. If we define T as the spacing between samples, in seconds, and let $Z = e^{j\Omega T}$ then the insertion loss of the digital filter will be equal to that of the prototype analog filter except for a warping in the frequency scale. For the analog filter

$$IL_A = 20 \log_{10} \left|\frac{1}{H(p)}\right|_{p=j\omega} \tag{5}$$

The warping of the frequency scale can be described as follows. If at some frequency ω_1 IL_A = a dB then there exists a frequency Ω_1 at which IL_D = a dB. ω_1 and Ω_1 are related by

$$\omega_1 = \frac{2}{T} \tan \frac{\Omega_1 T}{2} \tag{6}$$

or

$$\Omega_1 = \frac{2}{T} \tan^{-1} \frac{\omega_1 T}{2} \tag{7}$$

The frequency scale of the analog transfer function H(p) can be prewarped so that $IL_A = IL_D$ at some desired frequency ω_1. That is, if we want IL_D = a dB to occur at $\Omega_1 = \omega_1$, we can modify the frequency scale of H(p) by replacing p by

$$\frac{p}{\frac{2}{T}\tan\frac{\omega_1 T}{2}}. \tag{8}$$

This will shift ω_1 to $\omega_1' = \frac{2}{T}\tan\frac{\omega_1 T}{2}$. Then, when the bilinear transform is applied to $H\left(\frac{p}{\frac{2}{T}\tan\frac{\omega_1 T}{2}}\right)$, the IL_D = a dB point will fall at

$$\frac{2}{T}\tan^{-1}\frac{\omega_1 T}{2} = \frac{2}{T}\tan^{-1}\left(\frac{T}{2}\times\frac{2}{T}\tan\frac{\Omega_1 T}{2}\right) \tag{9}$$

$$= \frac{2}{T}\tan^{-1}\left(\tan\frac{\Omega_1 T}{2}\right) = \frac{2}{T}\times\frac{\Omega_1 T}{2} = \Omega_1 \tag{10}$$

4.4 Computation of the Ideal Insertion Loss

For a transfer function of the complex frequency p

$$H(p) = \frac{K(p^2+\omega_1^2)(p^2+\omega_2^2)\ \ldots}{(p-p_1)(p-p_2)(p-p_2^*)(p-p_3)(p-p_3^*)\ \ldots} \tag{11}$$

we can write

$$H(p) = \frac{K}{p-p_1}\times\frac{p^2+\omega_1^2}{p^2-2a_1p+a_1^2+b_1^2}\times\frac{p^2+\omega_2^2}{p^2-2a_2p+a_2^2+b_2^2}\times\ldots \tag{12}$$

$$= H_0 \times H_1 \times H_2 \times \ldots \tag{13}$$

where p_1 is real and $p_2=a_1+jb_1$, $p_2^*=a_1-jb_1$, $p_3=a_2+jb_2$, $p_3^*=a_2-jb_2$. The insertion loss is given by

$$20\log\left|\frac{1}{H(p)}\right|_{p=j\omega} = 20\log\left|\frac{1}{H_0}\right|_{p=j\omega} + 20\log\left|\frac{1}{H_1}\right|_{p=j\omega} + 20\log\left|\frac{1}{H_2}\right|_{p=j\omega} + \ldots \tag{14}$$

If we apply the prewarping transformation

$$p \rightarrow \frac{p}{\frac{2}{T}\tan\frac{\omega_1 T}{2}} \tag{15}$$

to H(p) followed by the bilinear transform $p = \frac{2}{T}\frac{Z-1}{Z+1}$ the result will be equivalent to the combined transformation of

$$p = \frac{1}{\tan\frac{\omega_1 T}{2}}\cdot\frac{Z-1}{Z+1} \tag{16}$$

In all designs that follow ω_1 will be made equal to the cut off frequency of the analog filter. Thus $\omega_1=\omega_c=1$ and following the prewarping transformation $\Omega_c=\omega_c=1$. Let $g = \frac{1}{\tan T/2}$, then $p = g\frac{Z-1}{Z+1}$.

$$H_o(Z) = \frac{K}{g\frac{Z-1}{Z+1} - p_1} = \frac{KZ+K}{(g-p_1)Z + (-g-p_1)} \tag{17}$$

$$H_1(Z) = \frac{g^2\frac{(Z-1)^2}{(Z+1)^2} + \omega_1^2}{g^2\frac{(Z-1)^2}{(Z+1)^2} - 2a_1 g\frac{(Z-1)}{(Z+1)} + a_1^2 + b_1^2} \tag{18}$$

$$= \frac{(g^2+\omega_1^2)Z^2 + (-2g^2+2\omega_1^2)Z + g^2 + \omega_1^2}{(g^2-2a_1g+a_1^2+b_1^2)Z^2 + (-2g^2+2a_1^2+2b_1^2)Z + (g^2+2a_1g+a_1^2+b_1^2)}$$

$$= \frac{R_2Z^2 + R_1Z + R_o}{Q_2Z^2 + Q_1Z + Q_o} \tag{19}$$

Letting $Z = e^{j\omega T}$ and using the relation $e^{jn\omega T} = \cos n\omega T + j\sin n\omega T$ we get for the magnitude-squared of the first order factor

$$|H_o(e^{j\omega T})|^2 = \frac{(K\cos\omega T + K)^2 + (K\sin\omega T)^2}{|(g-p_1)\cos\omega T - g - p_1|^2 + |(g-p_1)\sin\omega T|^2} \tag{20}$$

and for the magnitude squared of the second order factor

$$|H_1(e^{j\omega T})|^2 = \frac{(R_2\cos 2\omega T + R_1\cos\omega T + R_o)^2 + (R_2\sin 2\omega T + R_1\sin\omega T)^2}{(Q_2\cos 2\omega T + Q_1\cos\omega T + Q_o)^2 + (Q_2\sin 2\omega T + Q_1\sin\omega T)^2} \quad (21)$$

The corresponding insertion loss expressions are given by

$$IL_o = 10 \log_{10} \frac{1}{|H_o(e^{j\omega T})|^2} \text{ dB} \quad (22)$$

$$IL_1 = 10 \log_{10} \frac{1}{|H_1(e^{j\omega T})|^2} \text{ dB} \quad (23)$$

The total insertion loss is then

$$IL = IL_o + IL_1 + \ldots \quad (24)$$

K is obtained from $\frac{1}{H(p)}\Big|_{p=o}$ hence

$$K = \frac{-p_1(a_1^2+b_1^2)(a_2^2+b_2^2)\ \ldots}{\omega_1^2\omega_2^2\ \ldots} \quad (25)$$

These steps are generallized and given in the form of a FORTRAN IV program in APPENDIX.

As a specific example consider the bilinear transformation of the fifth order analog filter of Fig. 4.1. Using the sampling rates of $f_s = \frac{1}{T}$ = 16 KHz and 32 KHz we obtain the insertion loss vs. frequency characteristics of Fig. 4.2 and Fig. 4.3 respectively.

Three general features of the insertion loss characteristic should be observed.

1) The stop bands are symmetric about $(2n+1)f_s/2$ n=0, 1, 2, ...

2) The pass bands repeat periodically and are centered at nf_s.

3) The insertion loss for frequencies $f \leq f_s/2$ is the same as that of the

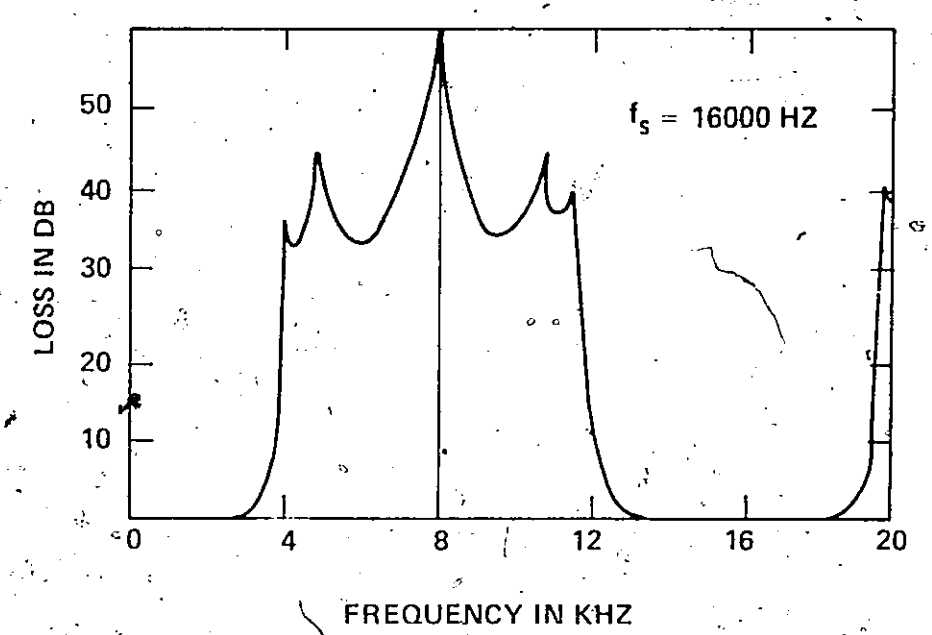


FIG. 4.2 — Insertion loss characteristic of a fifth order digital filter with 16KHz sampling rate

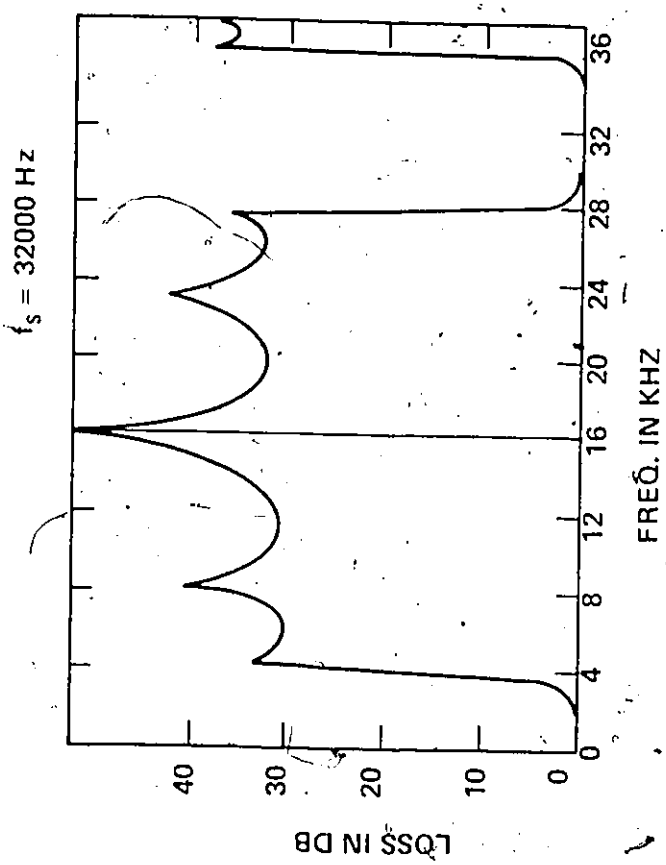


FIG. 4.3 — Insertion loss characteristic of a fifth order digital filter with 32KHz sampling rate

analog prototype transfer function H(p) within the warping factor $\tan \frac{\omega_c T}{2}$. In addition it may be noted that the frequency $f_s/2$ corresponds to the analog frequency of infinity.

Fig. 4.4 and Fig. 4.5 show the pass band plotted on an expanded loss scale for the sampling rates of 16 KH and 32 KHz respectively. The contours shown in these figures represent the PCM system requirements in the pass band. It can be seen that H(Z), derived from H(p) of the actual analog PCM channel filter through bilinear transformation, easily meets the insertion loss requirements of the system.

4.5 Some General Design Considerations

The insertion loss behaviour shown in Fig. 2 through Fig. 5 inclusive is ideal in that the effects of quantization, truncation and roundoff have not been introduced into the computation. Once the errors due to the finite arithmetic are introduced the pass band ripple becomes considerably larger. The effects of finite arithmetic on the insertion loss characteristic will be the subject of a large part of this chapter.

In addition to the effects of the finite arithmetic there is the problem of the periodically repeating pass bands. The system requirements specify that the average stop band attenuation of 33.2 dB be maintained at all frequencies above 5500 Hz. It is therefore necessary to introduce a low order analog filter to supply the attenuation at the frequencies at which the repeated pass bands occur. Such a filter must satisfy three requirements.

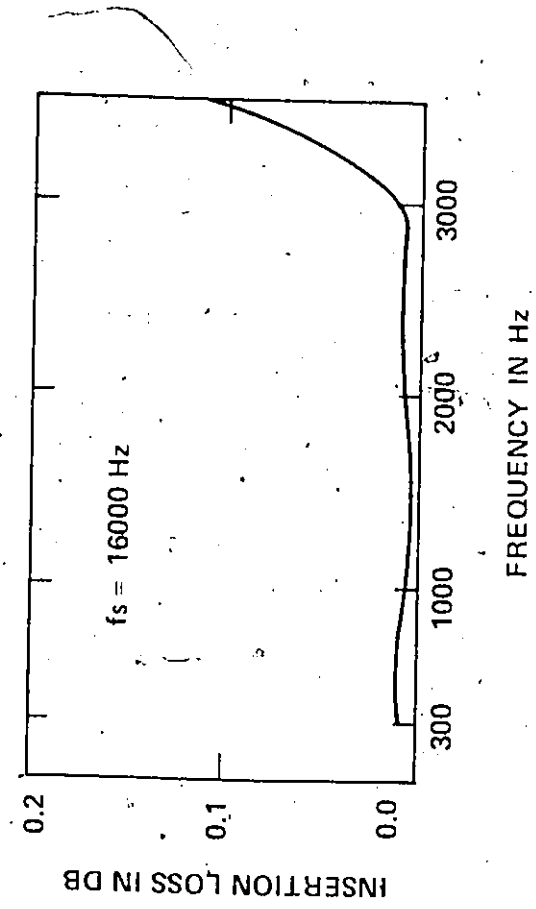


FIG. 4.4 — Pass band behaviour of a fifth order digital filter with 16KHz sampling rate

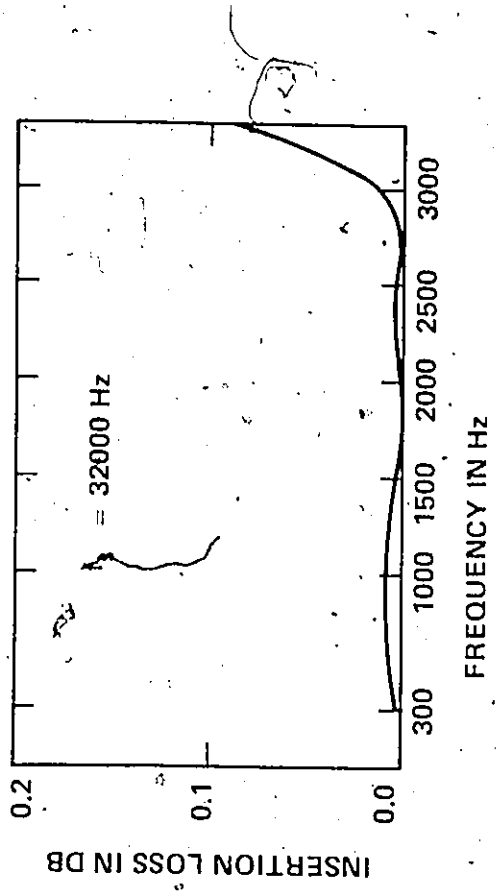


FIG. 4.5 — Pass band behaviour of a fifth order digital filter with 32KHz sampling rate

1) Its insertion loss must rise fast enough to supply the required attenuation in the second pass band of the digital filter.
2) It must not contribute significantly to the overall ripple in the pass band.
3) It must be realizable in an RC form.

Fig. 4.2 and Fig. 4.3 show that for higher sampling rates the nearest unwanted pass band occurs at progressively higher frequencies. For sufficiently high sampling rates a relatively slow rising insertion loss characteristic and a corresponding low order analog filter can be used. Increasing the sampling rate, however, increases the sensitivity of the digital processor to the arithmetic roundoff errors. The high sampling rates also require higher processing speeds. The processing speed in turn determines the number of channels that can be multiplexed as discussed in Section 2.4.

4.6 Realization of H(Z)

A variety of forms of realization of H(Z) in terms of unit delays, multipliers and adders exist.[12] Only the direct form and the cascade connection of direct forms will be considered here.

Applying the bilinear transformation $p = g\frac{Z-1}{Z+1}$ of Section 4.4 to the analog transfer function H(p) we obtain the digital filter transfer function H(Z). For a filter of order N H(Z) can be expressed as

$$H(Z) = \frac{A_o + A_1Z^{-1} + A_2Z^{-2} + \ldots + A_NZ^{-N}}{1 + B_1Z^{-1} + B_2Z^{-2} + \ldots + B_NZ^{-N}} \tag{26}$$

$$= \frac{\sum_{i=0}^{N} A_iZ^{-i}}{1 + \sum_{i=1}^{N} B_iZ^{-i}} \tag{27}$$

If X(Z) and Y(Z) are the Z-transforms of the input and output sequences respectively, then

$$Y(Z) = H(Z)\, X(Z) \tag{28}$$

and the corresponding difference equation is

$$y(nT) = \sum_{i=0}^{N} A_i\, x(nT-iT) - \sum_{i=1}^{N} B_i\, y(nT-iT) \tag{29}$$

where A_i and B_i are real coefficients computed in Section 4.4 and T is the sample spacing in seconds. $x(nT)$ is the input sequence of the signal samples to the filter and $y(nT)$ is the output sequence of the processed signal samples. The arithmetic operations corresponding to the difference equation are shown in Fig. 4.6. This realization of the difference equation is known as the direct form realization of $H(Z)$.

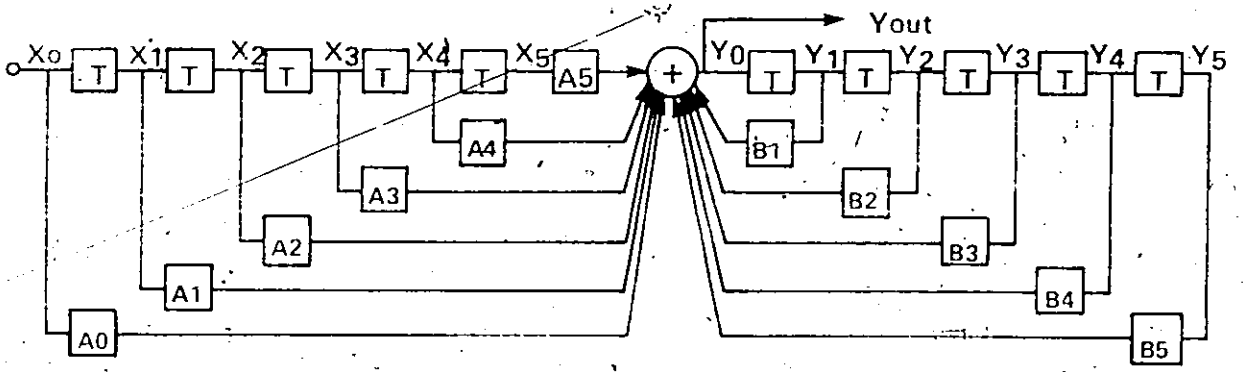
For high order filters $H(Z)$ can always be expressed as a product of some convenient number of lower order factors

$$H(Z) = \frac{\sum_{i=0}^{k} a_i Z^{-i}}{1 + \sum_{i=1}^{k} b_i Z^{-i}} \times \frac{\sum_{i=0}^{\ell} c_i Z^{-i}}{1 + \sum_{i=1}^{\ell} d_i Z^{-i}} \times \frac{\sum_{i=0}^{m} g_i Z^{-i}}{1 + \sum_{i=1}^{m} h_i Z^{-i}} \times \ldots \tag{30}$$

where $k+\ell+m+ \ldots = N$

Each factor can then be realized in the direct form as a lower order section and $H(Z)$ can be realized as a tandem connection of the sections. Fig. 4.7 shows a 5th order filter realization made up of a 3rd and 2nd order sections connected in tandem.

The direct form has a higher sensitivity to the coefficient accuracy than the cascade form for the same order of the filter, but requires only one adder.[13]



$$Yout = A_0X_0 + A_1X_1 + A_2X_2 + A_3X_3 + A_4X_4 + A_5X_5 - B_1Y_1 - B_2Y_2 - B_3Y_3 - B_4Y_4 - B_5Y_5$$

FIG. 4.6 — The direct form realization of a fifth order digital filter

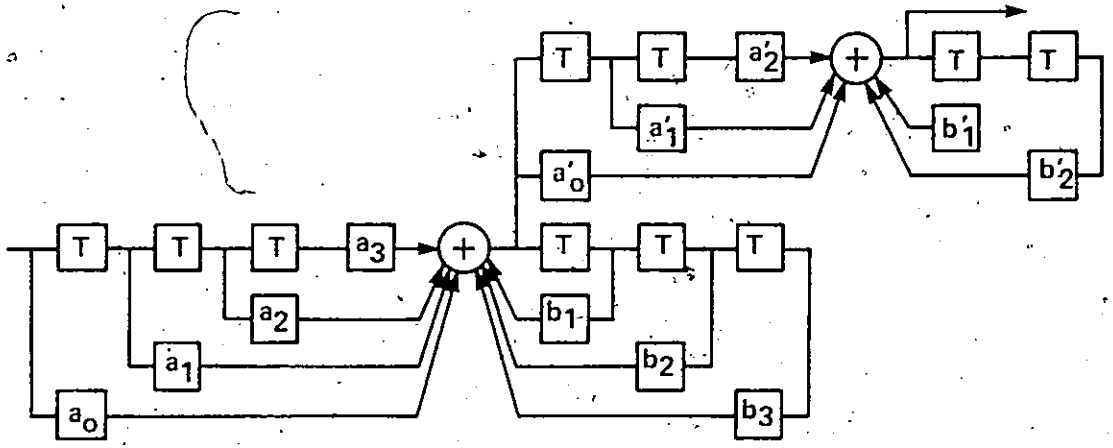


FIG. 4.7 — The cascade realization of a fifth order digital filter

In the proposed PCM channel filter application using the stored-product method, a quantization operation is performed on the output of the adder to limit the number of stored products. A realization with many adders would require equally many stages of quantization resulting in a large cumulative quantization error. A careful examination must therefore be given to the balance between the quantization error and coefficient accuracy.

4.7 Some Stability Considerations

The system equation relating the filter H(Z) to the input X(Z) and the output Y(Z) is

$$Y(Z) = H(Z)\, X(Z) \tag{31}$$

The signal quantization operation will affect the accuracy of X(Z) and Y(Z) only, and thus will contribute to the distortion of the output without affecting the stability of the filter. The coefficient roundoff, however, will affect H(Z) and therefore contribute to the filter instability and output error. Of the two error sources the quantization noise constitutes the larger contribution to the output error.

The errors affecting H(Z) will be considered first. H(Z) was obtained using the bilinear Z transform which maps the imaginary axis of the p-plane onto the unit circle in the Z-plane with the left half of the p-plane mapping into the interior of the unit circle. The bilinear Z transform preserves the order and stability of H(p). The resulting H(Z) can be expressed as

$$H(Z) = \frac{\sum_{k=0}^{N} a_k Z^{-k}}{1 + \sum_{}^{N} b_k Z^{-k}} = \frac{\sum_{k=0}^{N} a_k Z^{N-k}}{Z^N + \sum_{}^{N} b_k Z^{N-k}} \tag{32}$$

The critical values of p corresponding to the roots of the denominator of H(p) are mapped into the unit circle in the Z-plane. The unit circle is therefore a boundary of stability enclosing the singularities of the polynomial $Z^N + \sum_{k=0}^{N} b_k Z^{N-k} = D(Z)$. J.F. Kaiser[14] has worked out the bounds on the accuracy of the coefficients b_k to ensure that the zeros of the denominator polynomial of H(Z) all lie within the unit circle. He has shown that if any of the b_k are changed by the amount $1 + \sum_{k=1}^{N} b_k$ then D(Z) will have a zero at Z=1. Any further change in the magnitudes of any combination of the b_k in such a manner as to cause D(Z) to change sign will result in an unstable digital filter.

In this light we examine the stability of the fifth order digital filter obtained in Section 4.4 when realized in the direct form. For comparison we also introduce a third order digital filter which satisfies the same pass band requirements. In each case three different sampling rates are considered, 16 KHz, 24 KHz and 32 KHz. Table 4.1 lists the b_k coefficients for the various cases followed by the value of $1 + \sum_{k} b_k$. Table 4.2 and Table 4.3 give the Z-plane roots and p-plane roots of the denominator polynomials of H(Z) and H(p) respectively. Fig. 4.8 and Fig. 4.9 show the Z-plane pole positions in relation to the unit circle.

In this application each multiplier will be replaced by a 4096-bit Read Only Memory (ROM). The signal-coefficient products will be stored as fixed-point numbers in fixed length ROM registers with all binary points aligned. Once the register length is specified the degree of the coefficient roundoff will be determined by the largest product that must be stored. The products will be stored in 13-bit

TABLE 4.1

f_s	5TH ORDER $b_0Z^5+b_1Z^4+b_2Z^3+b_3Z^2+b_4Z+b_5$		3RD ORDER $b_0Z^3+b_1Z^2+b_2Z+b_3$	
16 KHz	b_0	1.000000	b_0	1.000000
	b_1	-0.9162497	b_1	-0.06027823
	b_2	1.338713	b_2	0.6093505
	b_3	-0.5904635	b_3	0.06739420
	b_4	0.2985120		
	b_5	-0.03823003		
	$1 + \sum b_k =$	1.092282	$1 + \sum b_k =$	1.6164665
24 KHz	b_0	1.000000	b_0	1.0000000
	b_1	-2.375177	b_1	-0.9877751
	b_2	2.953626	b_2	0.7787396
	b_3	-2.000437	b_3	-0.09407682
	b_4	0.7689281		
	b_5	-0.1240241		
	$1 + \sum b_k =$	0.222916	$1 + \sum b_k =$	0.7068877
32 KHz	b_0	1.000000	b_0	1.000000
	b_1	-3.096391	b_1	-1.487411
	b_2	4.306952	b_2	1.051284
	b_3	-3.200346	b_3	-0.1965227
	b_4	1.265985		
	b_5	-0.2093802		
	$1 + \sum b_k =$	0.066821	$1 + \sum b_k =$	0.367350

TABLE 4.2

	Z-PLANE ROOTS	
	5TH ORDER	3RD ORDER
f_s	$b_0Z^5+b_1Z^4+b_2Z^3+b_3Z^2+b_4Z+b_5$	$b_0Z^3+b_1Z^2+b_2Z+b_3$
16 KHz	0.1629735 + j 0.0 0.1748777 + j 0.5234066 0.1748777 - j 0.5234066 0.2017604 + j 0.8541495 0.2017604 - j 0.8541495	-0.1074241 + j 0.0 0.08385116 + j 0.7876133 0.08385116 - j 0.7876133
24 KHz	0.3820344 + j 0.0 0.4453800 + j 0.4505452 0.4453800 - j 0.4505452 0.5511913 + j 0.7106704 0.5511913 - j 0.7106704	0.1253721 + j 0.0 0.4312015 + j 0.6961922 0.4312015 - j 0.6961922
32 KHz	0.5063910 + j 0.0 0.5859537 + j 0.3835051 0.5859537 - j 0.3835051 0.7090462 + j 0.5834051 0.7090462 - j 0.5834051	0.2730670 + j 0.0 0.6071720 + j 0.5924769 0.6071720 - j 0.5924769

TABLE 4.3

P-PLANE ROOTS	
5TH ORDER	3RD ORDER
-1.095684 + j 0.0	-2.024649 + j 0.0
-0.160878 + j 1.196356	-0.338753 + j 1.432000
-0.160878 - j 1.196356	-0.338753 - j 1.432000
-0.639997 + j 0.963326	
-0.639997 - j 0.963326	

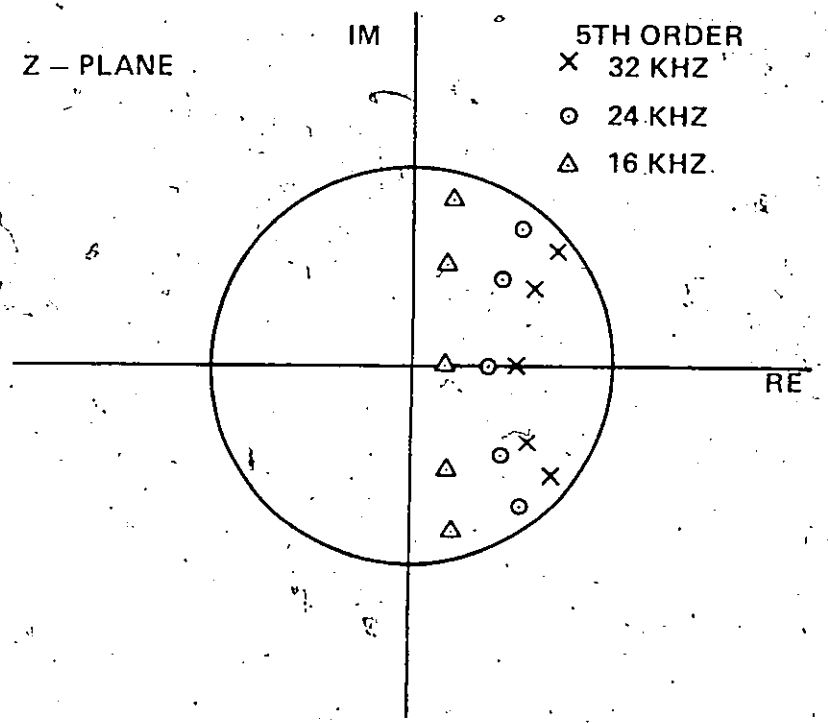


FIG. 4.8 — Z-plane pole positions and the unit circle for the fifth order digital filter

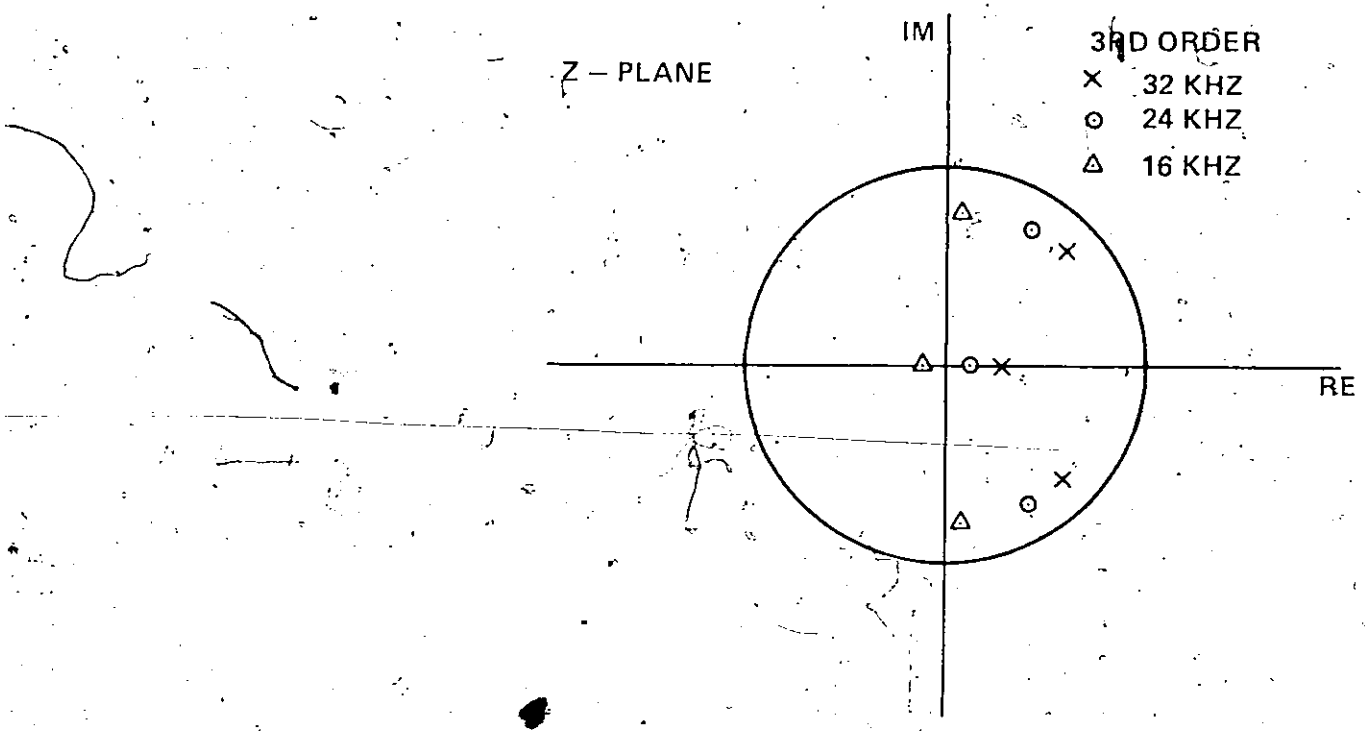


FIG. 4.9 — Z-plane pole positions and the unit circle for the third order digital filter.

registers including a sign bit and one fraction bit. This means that in the worst case each product will be rounded to the nearest 0.5 decimal accuracy. We now consider if the stability requirements can be met. Here the third order case with 24 KHz sampling rate will be singled out as an important example.

For large signals the stability criterion is easily met. Consider the largest signal value 2015.5 and the coefficient -0.08407682 which is most sensitive to roundoff. The product 0.08407682 x 2015.5 = 169.45683 after rounding becomes 169.5. If we now recompute the coefficient as

$$\frac{169.5}{2015.5} = 0.08409823$$

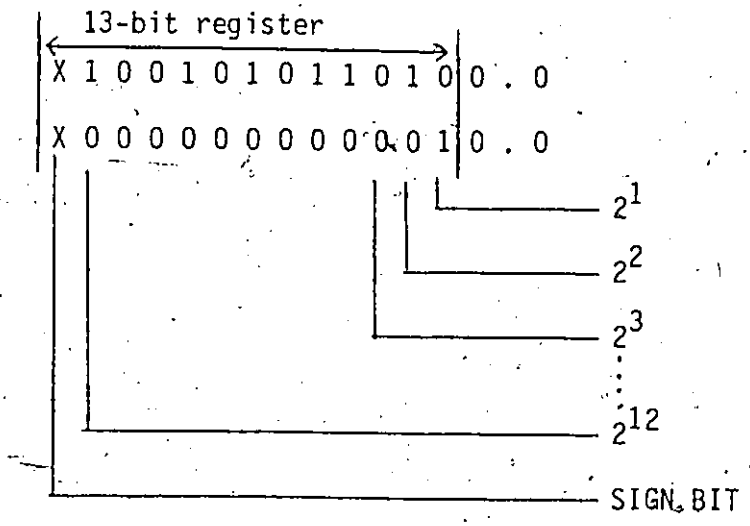
we see that the effective change in the coefficient is 0.00002141. The smallest non zero signal corresponds to 1.0 and represents the worst case in which the coefficients themselves must be rounded to the nearest 0.5. The following changes in the coefficients will take place.

Original Coefficient	Rounded Coefficient	Change
0.9877751	1.000000	0.0122249
0.7787396	1.000000	0.2212604
0.08407682	0.000000	0.08407682
	SUM =	0.3175621

The sum is less than 0.7068877 obtained for $1 + \sum_k b_k$. In this case it is clear that the stability conditions can be met with 4096-bit memory per multiplier.

Of the three sets of fifth order filter coefficients given in Table 4.I only the set corresponding to the 16 KHz sampling rate satisfies the stability requirement with the 13-bit product representation.

It can be seen that the value of $1 + \sum_k b_k$ decreases for the higher sampling rates. The absolute value of the b_k coefficients, however, increases. Since the dynamic range of the signal-coefficient products is restricted to a 13-bit fixed point representation the roundoff error must also increase. As an example consider $|b_1| = 2.375177$. For the largest signal the product to be stored is $2015.5 \times 2.375177 = 4787.0691435$, which rounded to one binary fractional digit is 1001010110100.0. For the smallest signal the product is $1 \times 2.375177 = 2.375177$ which after similar rounding becomes 10.1. If these are then stored in a 13-bit field with the binary points aligned, then with allowance for the sign bit we get



In this case the effective coefficient roundoff is 0.375177 which is greater than the 0.222916 value for $1 + \sum_k b_k$.

The roundoff error becomes worse for the higher sampling rates while the sensitivity increases. This eliminates the use of the 24 KHz and 32 KHz sampling rates in conjunction with the fifth order direct form realization. The stability criteria can always be met, however, by extending the product accuracy and memory storage.

4.8 The Effect of Coefficient Accuracy on the Insertion Loss

The direct form realization consists of a recursive part characterized by the coefficients b_k and a nonrecursive part characterized by the coefficients a_k. The accuracy used to represent the b_k coefficients was shown to affect the stability of the filter. The deviation of the actual insertion loss response of the filter from the ideal is on the other hand affected by both a_k and b_k coefficients. Table 4.4 summarizes the coefficient values for the nonrecursive sections.

In the preceding section it was sufficient to examine the worst case to determine if the stability criteria are satisfied. In this section, the best ratio of the number of fractional bits to the product register length will be determined in order to maximize the dynamic range and minimize the insertion loss ripple in the pass band

Two filter realizations will be considered. The first will be the fifth order direct form realization of Fig. 4.6 with the coefficients corresponding to the 16 KHz sampling rate. The second will be a sixth order filter consisting of two identical third order sections connected in tandem as shown in Fig. 4.10 with coefficients corresponding to the 24 KHz sampling rate.

TABLE 4.4

f_s	5TH ORDER		3RD ORDER	
16 KHz	a_0	0.08950067	a_0	0.3673946
	a_1	0.1788376	a_1	0.4408383
	a_2	0.2778023	a_2	0.4408383
	a_3	0.2778023	a_3	0.3673946
	a_4	0.1788376		
	a_5	0.08950067		
24 KHz	a_0	0.04616759	a_0	0.2726230
	a_1	0.009372354	a_1	0.0808208
	a_2	0.05591779	a_2	0.0808208
	a_3	0.05591779	a_3	0.2726230
	a_4	0.009372354		
	a_5	0.04616759		
32 KHz	a_0	0.03174654	a_0	0.2233105
	a_1	-0.02710262	a_1	-0.3963513
	a_2	0.02876602	a_2	-0.3963513
	a_3	0.02876602	a_3	0.2233105
	a_4	-0.02710262		
	a_5	0.03174654		

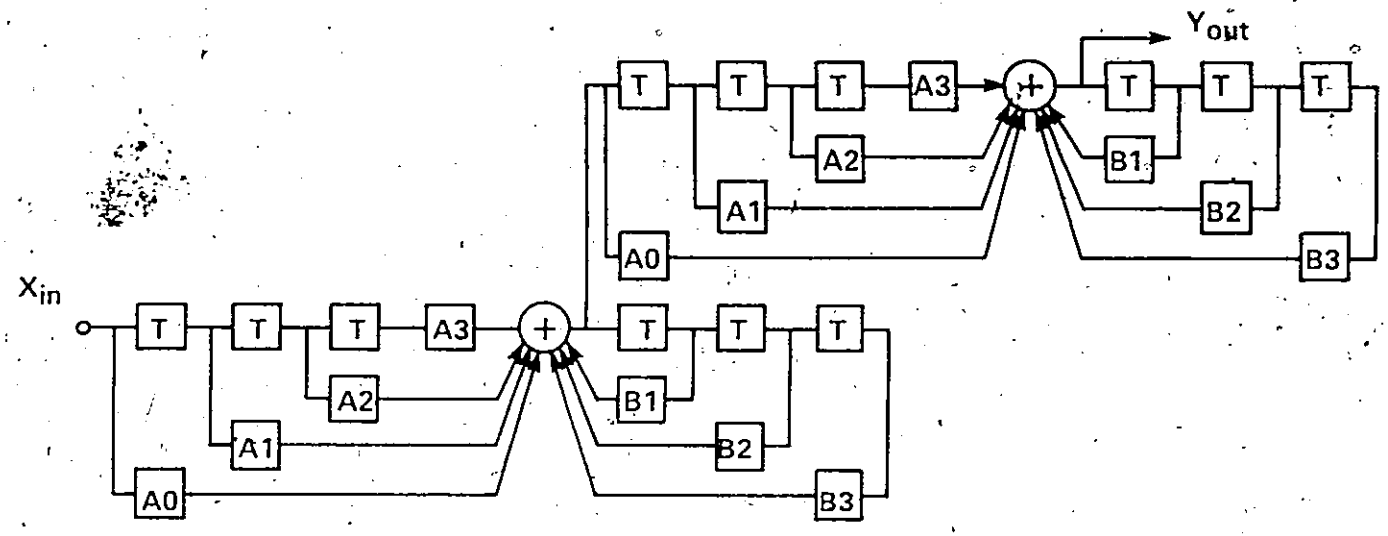


FIG. 4.10 — Sixth order filter with two identical third order sections

The effect of coefficient roundoff on the insertion loss will be studied by simulating the arithmetic process of the respective digital filters. A simulation program which computes insertion loss for various conditions of arithmetic accuracy will be described in Chapter V.

In the proposed fixed point, stored-product processor the degree of coefficient roundoff varies with the magnitude of the product stored.[16] The exact degree of coefficient roundoff can only be computed for a specific coefficient and a given signal amplitude. For a given set of coefficients and some signal amplitude an average roundoff error can be computed. A sample computation to determine the average coefficient roundoff error is illustrated in Table 4.5. This case corresponds to the fifth order filter and a signal amplitude of about 3% of the maximum. Fig. 4.11 shows the dependence of the coefficient roundoff on the amplitude of the signal for the fifth and sixth order filters. The sixth order filter represented by curve (1) suffers lower coefficient roundoff errors for the same values of signal amplitude.

Next, the pass band ripple is obtained for several values of coefficient roundoff error. The insertion loss ripple is plotted as a function of the roundoff error in Fig. 4.12. The lower slope of the curve corresponding to the sixth order filter indicates a lower sensitivity to the coefficient error in comparison to the fifth order case. The sixth order filter was designed to have a higher pass band ripple in its ideal response and so initially curve (1) falls above curve (2). Fig. 4.13 and Fig. 4.14 are sample plots of the

TABLE 4.5

COEF	SIGNAL	PRODUCT	ROUNDED PRODUCT	ROUNDED COEF	CHANGE IN COEF	FRACTIONAL COEF ERROR
C	S	C x S	$(CS)_R$	$C_R = (CS)_{R/S}$	$C - C_R$	$\frac{C - C_R}{C}$
.08950067	60.	5.37004	5.5	.09166666	.002166	.0242
.1788376	60.	10.730256	10.5	.175000	.0038376	.02146
.2778023	60.	16.668138	16.5	.275000	.0028023	.01009
.2778023	60.	16.668138	16.5	.275000	.0028023	.01009
.1788376	60.	10.730256	10.5	.175000	.0038376	.02146
.08950067	60.	5.37004	5.5	.0916666	.002166	.0242
0.9162497	60.	54.974982	55.0	0.916666	.000417	.000455
1.338713	60.	80.32278	80.5	1.341667	.002954	.002206
0.5904635	60.	35.42781	35.5	0.591666	.0012032	.002038
0.298512	60.	17.91072	18.0	0.30000	.001488	.004985
0.03823003	60.	2.2938018	2.5	0.0416666	.003437	.08989

Sum = .211074

Average Fractional Coefficient Error = .019188

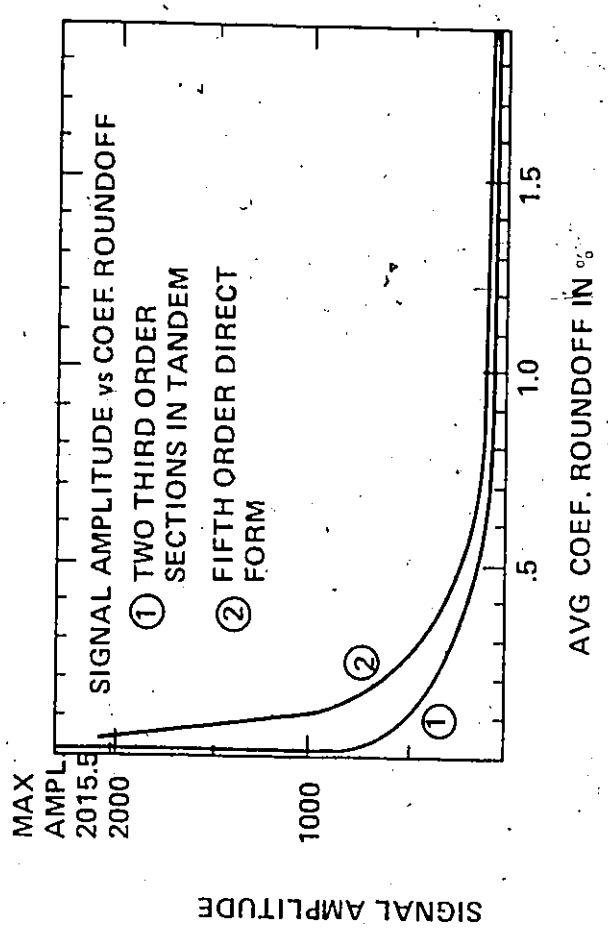


FIG. 4.11 — Dependence of the average coefficient roundoff error on signal amplitude in fixed point arithmetic

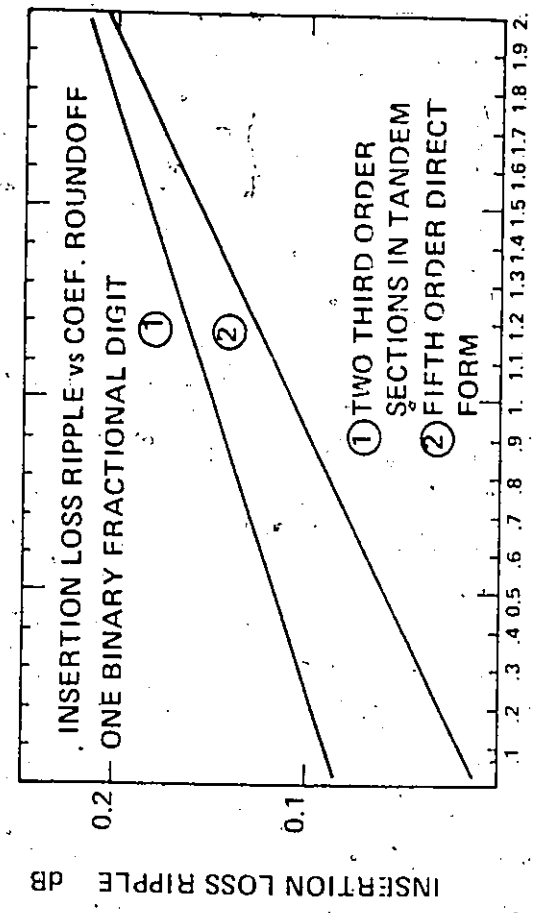


FIG. 4.12 — Pass band ripple as a function of coefficient roundoff

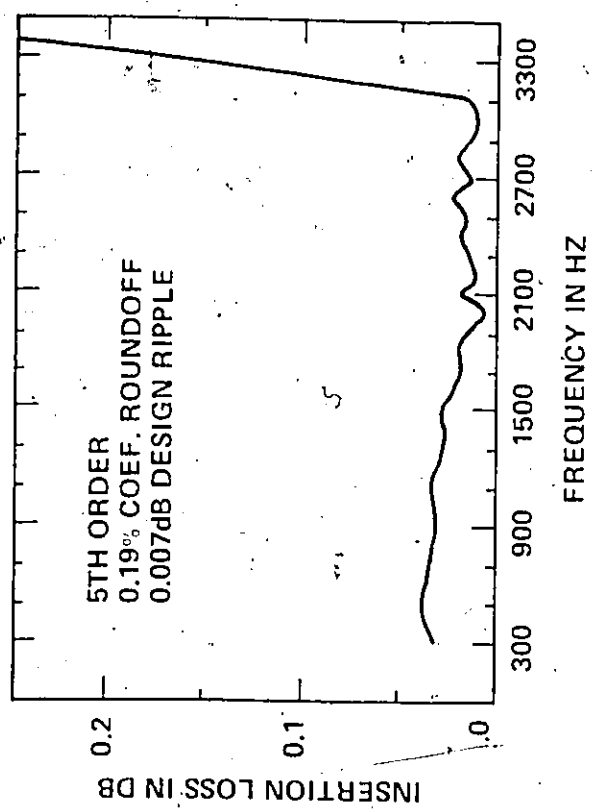


FIG. 4.13 — Pass band ripple for a given coefficient accuracy

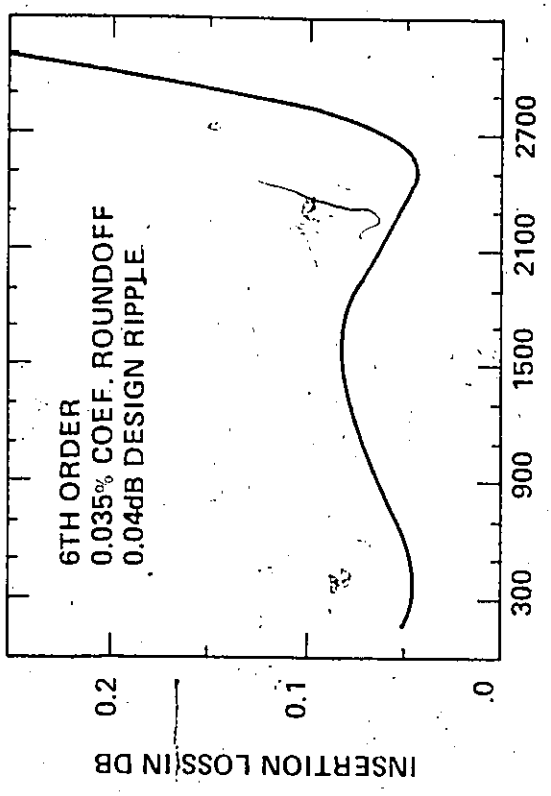


FIG. 4.14 — Pass band ripple for a given coefficient accuracy

pass band insertion loss characteristic for the fifth and sixth order filters respectively.

The main feature brought out in Fig. 4.11 is that for signals with amplitudes below 3% of the maximum the coefficient roundoff error becomes greater than 2%. Fig. 4.12 shows that coefficient roundoff error of more than 2% corresponds to pass band ripple of more than 0.2 dB. Since the contribution to the ripple due to quantization noise and the analog filter have not been included so far the coefficient roundoff error of 2% or more can be considered unsatisfactory.

Since the products corresponding to the different coefficients are stored in different ROM blocks the location of the binary point can be established independently for each block. Thus for low coefficient values more fractional bits can be used. Table 4.6 summarizes the binary point locations for each of the five distinct ROM blocks for the sixth order filter. The leading bit position represented by X is reserved for the sign bit. The table shows the fractional bits for 13-bit and 16-bit register lengths. Since there are 256 products to be stored in each 4096-bit ROM each product can be stored as 4096/256-bit or 16-bit number. However, if 13-bit lengths are used then a greater number of products can be stored making finer step quantization possible. The latter case will be shown to have the greater advantage. In either case, however, following the summation of the products at the adder the resulting sum must be rounded to the nearest unity before quantization is performed.

TABLE 4.6

ROM BLOCK	COEFFICIENT	MAXIMUM SIGNAL	MAXIMUM PRODUCT	MAXIMUM PRODUCT IN BINARY REPRESENTATION	
1	0.272623	2015.5	549.4716	X1000100101 1 X	XXX
2	0.0808208	2015.5	162.858	X0100011 1 XXXX	XXX
3	0.9877751	2015.5	1990.86	X1111100011 1 X	XXX
4	0.7787396	2015.5	1569.55	X1100010000 1 X	XXX
5	0.08407682	2015.5	169.456	X0101001 1 XXXX	XXX

Table 4.7 summarizes the results plotted in Fig. 4.15 through Fig. 4.18 inclusive. These results correspond to the signal amplitude of 10% of the maximum value. For signal amplitudes below 3% of the maximum the ripple becomes greater than 0.2 dB due to the coefficient roundoff. As the signal amplitude is increased up to its maximum value the ripple in the pass band approaches the design value of 0.04 dB, however, the quantization noise becomes the main contribution to the ripple and will be discussed next.

4.9 The Effect of Quantization

In this section the effect of quantization on the pass band ripple of the sixth order filter will be considered. A comparison will be made between the 16-bit product representation in conjunction with the quantization level assignment of Table 3.1 and the 13-bit product representation in conjunction with an enlarged quantization level assignment of Table 4.8. Since the output of the filter must be in the form of the standard A-law the quantization according to Table 4.8 will only be performed between the two third-order sections. Doubling the number of quantization levels in the upper three segments of the companding curve is aimed at improving the pass band ripple for the high amplitude signals.

In order to observe the effect of the smaller quantization intervals corresponding to L=5, 6 and 7 a signal amplitude of 90% of the maximum is applied to the input of the sixth order filter. Fig. 4.19 shows the resulting pass band response when the 16-bit product representation is used with the level assignment of Table 3.1. For comparison Fig. 4.20 shows the pass band response corresponding to the 13-bit product representation and the level assignment of Table 4.8.

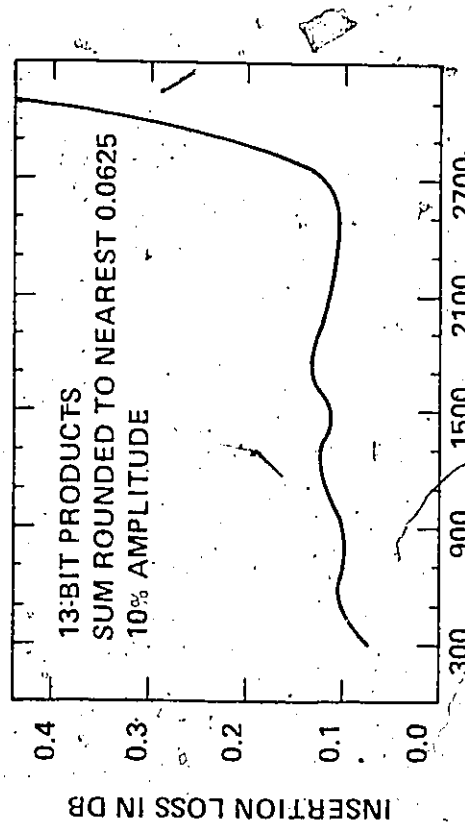


FIG. 4.15 — Pass band behaviour with four practical bits in each 13-bit product

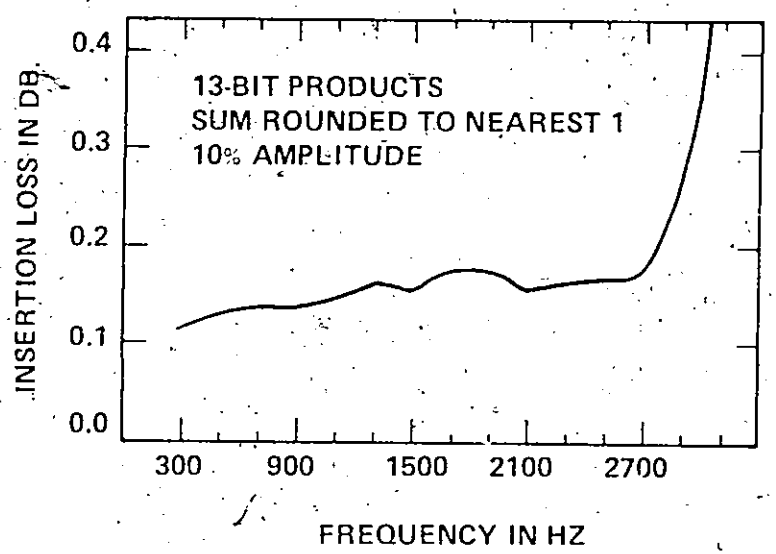


FIG. 4.16 — Pass band behaviour with no fractional bits in each 13-bit product

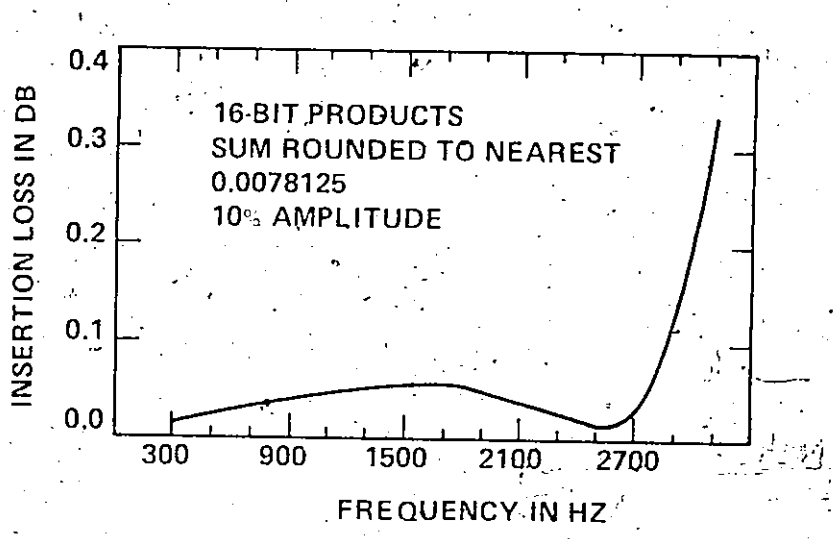


FIG. 4.17 — Pass band behaviour with seven fractional bits in each 16-bit product

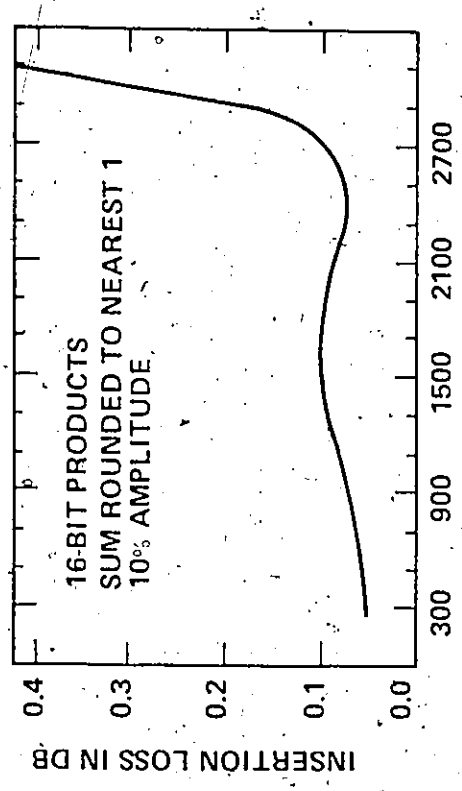


FIG. 4.18 — Pass band behaviour with no fractional bits in each 16-bit product

TABLE 4.7

	16-BIT PRODUCTS		13-BIT PRODUCTS	
	SUM ROUNDED TO NEAREST 0.0078125	SUM ROUNDED TO NEAREST 1	SUM ROUNDED TO NEAREST 0.0625	SUM ROUNDED TO NEAREST 1
RIPPLE	0.088 dB	0.124 dB	0.125 dB	0.152 dB

10% Signal Amplitude

TABLE 4.8

q	L								q
	0	1	2	3	4	5	6	7	
0	0	16	32.5	65.5	131.5	259.5	519.5	1039.5	0
1	1	17	34.5	69.5	139.5	267.5	535.5	1071.5	1
2	2	18	36.5	73.5	147.5	275.5	551.5	1103.5	2
3	3	19	38.5	77.5	155.5	283.5	567.5	1135.5	3
4	4	20	40.5	81.5	163.5	291.5	583.5	1167.5	4
5	5	21	42.5	85.5	171.5	299.5	599.5	1199.5	5
6	6	22	44.5	89.5	179.5	307.5	615.5	1231.5	6
7	7	23	46.5	93.5	187.5	315.5	631.5	1263.5	7
8	8	24	48.5	97.5	195.5	323.5	647.5	1295.5	8
9	9	25	50.5	101.5	203.5	331.5	663.5	1327.5	9
10	10	26	52.5	105.5	211.5	339.5	679.5	1359.5	10
11	11	27	54.5	109.5	219.5	347.5	695.5	1391.5	11
12	12	28	56.5	113.5	227.5	355.5	711.5	1423.5	12
13	13	29	58.5	117.5	235.5	363.5	727.5	1455.5	13
14	14	30	60.5	121.5	243.5	371.5	743.5	1487.5	14
15	15	31	62.5	125.5	251.5	379.5	759.5	1519.5	15
						387.5	775.5	1551.5	16
						395.5	791.5	1583.5	17
						403.5	807.5	1615.5	18
						411.5	823.5	1647.5	19
						419.5	839.5	1679.5	20
						427.5	855.5	1711.5	21
						435.5	871.5	1743.5	22
						443.5	887.5	1775.5	23
						451.5	903.5	1807.5	24
						459.5	919.5	1839.5	25
						467.5	935.5	1871.5	26
						475.5	951.5	1903.5	27
						483.5	967.5	1935.5	28
						491.5	983.5	1967.5	29
						499.5	999.5	1999.5	30
						507.5	1015.5	2031.5	31
δ(L)	1	1	2	4	8	8	16	32	

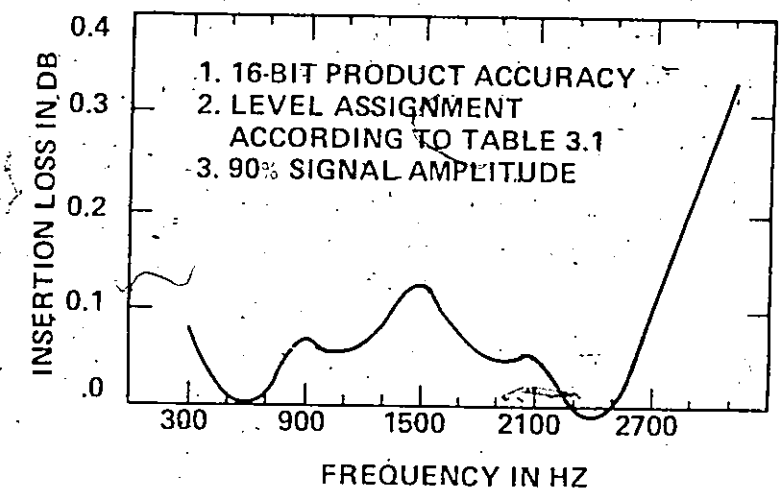


FIG. 4.19 — Pass band behaviour with 16-bit product accuracy and standard level assignment

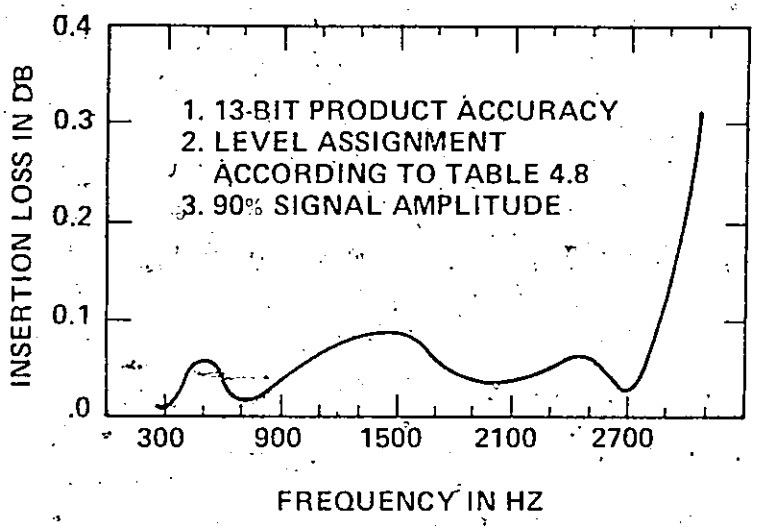


FIG. 4.20 — Pass band behaviour with 13-bit product accuracy and finer level assignment

The final design consisting of two identical third order stages will implement the 13-bit product representation with the quantization at the output of the first stage according to Table 4.8 and the quantization at the output of the second stage according to Table 3.1.

4.10 The Analog Filter

As was stated earlier the purpose of the analog filter is to complement the digital filter insertion loss response in the stop band. The second pass band of the digital filter begins at 20.6 KHz but the stop band loss drops below the 33.4 dB level at about 19 KHz. The analog filter must make up the loss difference at that point and maintain the 33.4 dB loss through out the second pass band. At the same time the filter must not contribute significantly to the ripple of the first pass band.

An analog filter of this kind must be included in each of the 24 channels in a manner in which the existing 5th order analog filters are used at present. To make the use of a digital filter as advantageous as possible the order of the analog filter must be kept to a minimum. The digital filter then serves to reduce the order of the analog filters presently used in the PCM system. If an analog filter of order higher than three must be used the use of a digital filter cannot be justified over the existing totally analog method.

Fig. 4.21 shows the loss characteristics of several low-order analog filters plotted over the insertion loss of the sixth order digital filter for comparison. 2B and 3B represent the second and

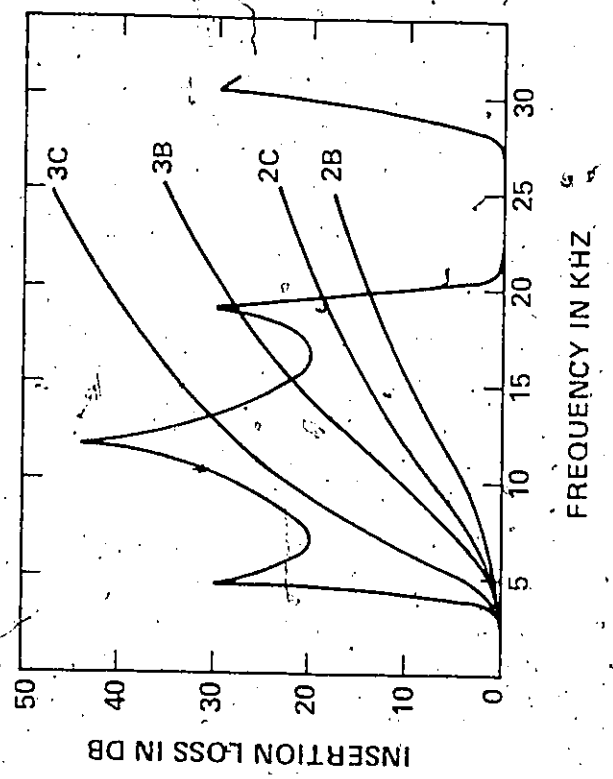


FIG. 4.21 — Stop band loss of a sixth order digital filter and of several low-order analog filters

third order Butterworth filters while 2C and 3C designate the second and third order Chebyshev filters respectively. The third order Chebyshev response is chosen as a satisfactory case. All other cases fail to provide sufficient loss in the 5 to 10 KHz range.

In all four cases the analog filter was designed to yield a minimum distortion of the primary pass band. A second order filter might be made sufficient by introducing a substantial slope in the first pass band of the digital filter. An attempt can then be made to equalize out the slope by adjusting the digital filter coefficients. This approach was not attempted within the time available.

4.11 The Composite Response

In this section the total insertion loss response of the final filter is examined and compared to the system requirements. All sources of error are included with the exception of the effects of temperature and aging on the analog filter.

Fig. 4.22 and Fig. 4.23 illustrate the stop band performance for low and high amplitude signals respectively. The specification contour beyond the 5.5 KHz point represents an average loss requirement assuming an even loss in the stop band. In marginal cases a more precise method of computing the requirement can be used. If this method is used the insertion loss may be allowed to fall below the average of 33.4 dB over some frequency bands if over other bands there exists a proper excess loss. Because of its complexity such a computation was not done in this case but it is estimated that the dip in insertion loss in the region of 6 KHz is acceptable.

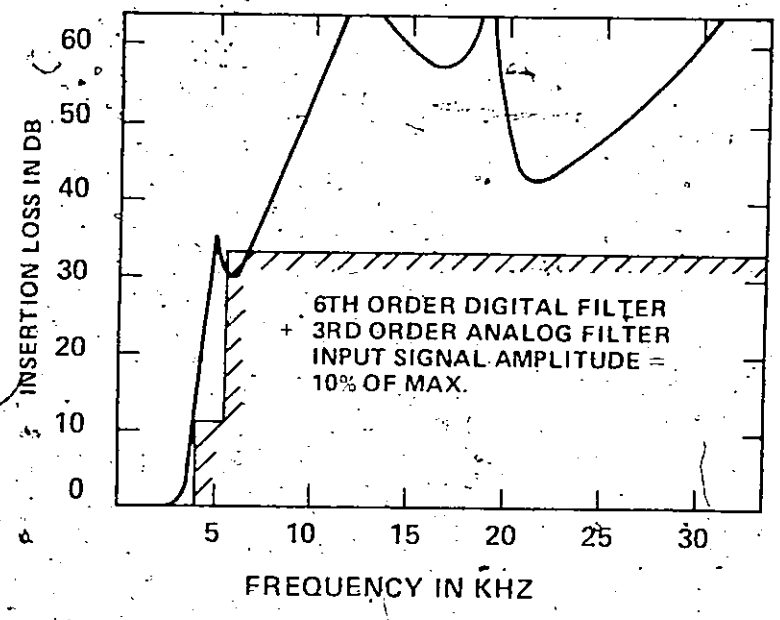


FIG. 4.22 — Combined stop band loss for low amplitude signals

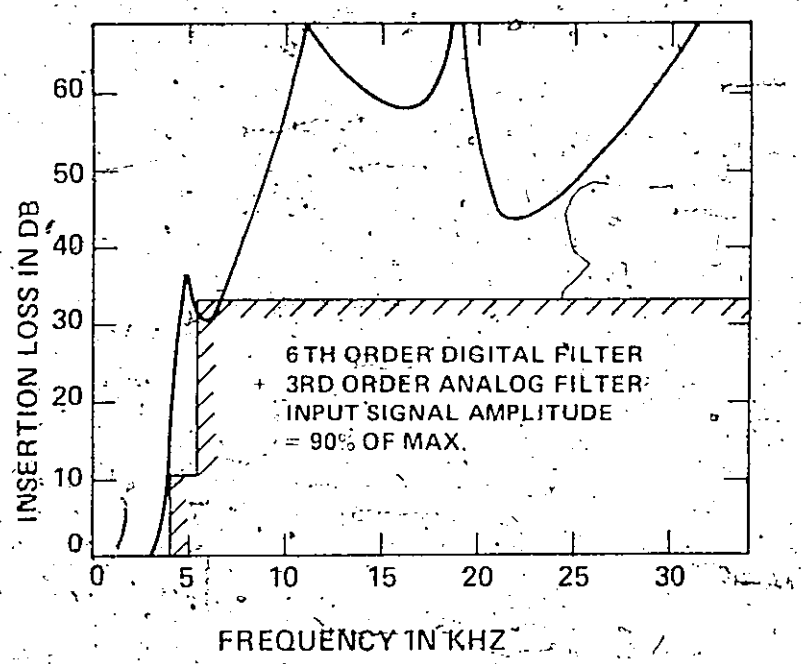


FIG. 4.23 — Combined stop band loss for high amplitude signals

Fig. 4.24 and Fig. 4.25 show the pass band ripple corresponding to the high and the low signal amplitudes respectively. In both cases the largest contribution to the pass band ripple appears in the form of rolloff at 3000 Hz. This rolloff can be reduced by designing the digital filter for a somewhat wider pass band and a lower stop band loss. Although an attempt was made to obtain a reasonable trade off between these variables the final design discussed here was not optimized in the strict sense.

When the input signal amplitude is less than 5% or greater than 95% of the maximum value the pass band ripple exceeds the specified limit. The degradation of the pass band for the upper 5% of the signal amplitudes is due to the particular algorithm used to prevent overflow in the adder accumulator and may be entirely removed by improving the method of treating the overflow. The performance for the lower 5% of the signal amplitudes can only be improved by introducing finer quantization steps within the digital filter.

4.12 Mechanization

Since the proposed sixth order digital filter consists of two identical third order sections only one third order section need be realized in hardware if the input signal is made to pass through the section twice. The schematic diagram of Fig. 4.26 shows the implementation of one such third order section.

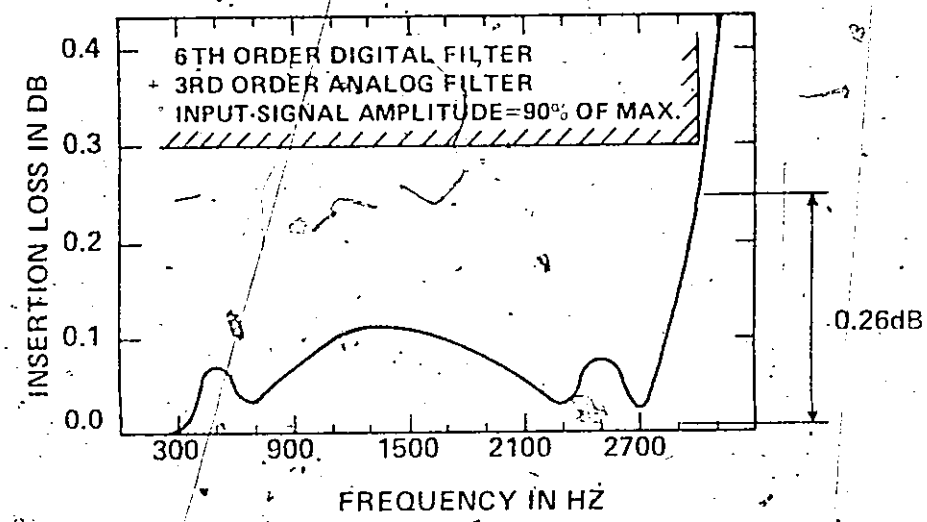


FIG. 4.24— Combined pass band ripple for high amplitude signals

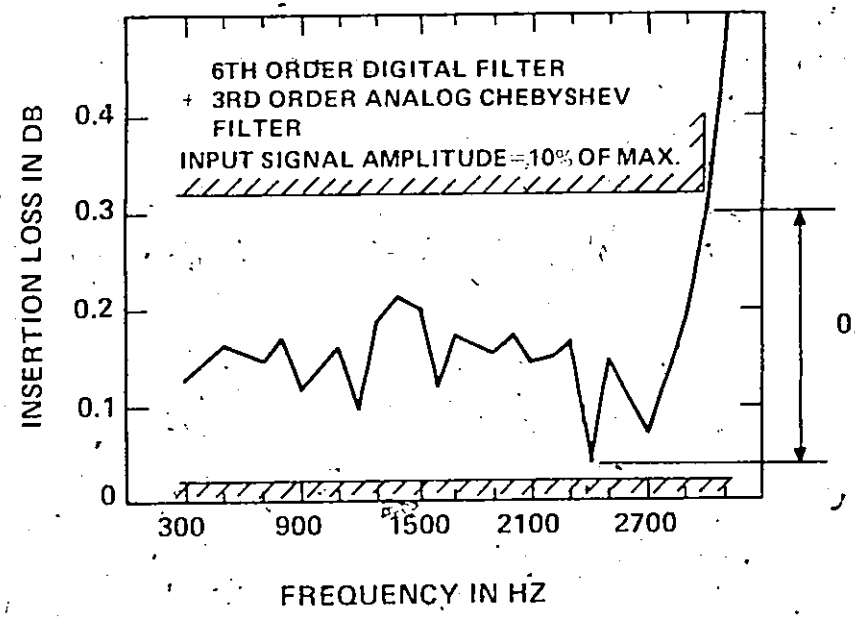


FIG. 4.25 — Combined pass band ripple for low amplitude signals

The products of the input signal and the four nonrecursive coefficients are stored in the ROM blocks designated A0, A1, A2 and A3. Similarly the products of the output signal and the three recursive coefficients are stored in the ROM blocks designated B1, B2 and B3. Each of the seven ROMs are of 4096-bit capacity. Blocks designated L0 through L7 inclusive are ROM's storing quantization level values of Table 3.1 and Table 4 Blocks L0 to L4 inclusive each have 416-bit capacity while blocks L4 to L7 each have 1248-bit capacity. Block D represents a 3328-bit ROM which simply converts the 13-bit linear code to the 8-bit compressed code.

R1 through R4 and R6 through R8 are 13-bit storage registers with parallel transfer capability. AC and FA are the adder accumulator and a set of full adders respectively; together with the incident register R5 they make up a 16-bit parallel adder. The binary point separating the fractional bits is represented by the symbol ⊥. The 13-bit outputs from each of the seven multipliers are connected to the OR gates G1 such that the binary points ⊥ are properly aligned. All other gates, G2, G3, G4 and G5 together with the eight ROMs L0 through L7 make up the quantization network whose operation was described in Chapter III.

The operation of the third order section when it is used as the first stage of the sixth order filter differs slightly from its operation as the second stage. During the first stage operation the quantization level assignment of Table 4.8 is used.

FIG. 4.26 — Hardware implementation of a third order section

In this mode of operation each of the ROMs L5, L6 and L7 are provided with one additional input carrying the next least significant bit. For example, for L7 the 7th bit of the accumulator AC is used as the extra input. This extra input is gated through an AND gate which is enabled during the first stage operation and disabled during the second stage operation. For simplicity Fig. 4.26 shows the second stage operation.

The sequence of events during the computation of one output sample can be described as follows. The contents of the accumulator AC are cleared at the end of the sequence so that we start the next cycle with all bits in the accumulator set to zero excep for bit 13 which is set to 1 to produce rounding. Simultaneously with the clearing of the accumulator the contents of the unit delay registers are advanced: R1→R2, R2→R3, R3→R4, R6→R7, R7→R8. This is done in the time slot of 150 nsec. while the 13-bit linear code is being converted to the 8-bit compressed code at the output. The sequence begins with the new input sample entered in the register R1. The products of the signal values stored in R1, R2, R3, R4, R6, R7 and R8 and the corresponding coefficients A0, A1, A2, A3, B1, B2 and B3 are retrieved from the ROMs in the above order at intervals of 50 nsec. There will be a period of 150 nsec before the first product is available but the remaining products will follow 50 nsec apart. The interval of 50 nsec is established by the speed of the parallel adder. Thus the final sum will be available 500 nseconds from the arrival of the input sample. If bit #1 of the accumulator is zero the quantization

step is carried out next. If bit #1 is one the contents of the accumulator bits #2 through #12 is negated in accordance with two's complement.

The process of negation and quantization takes 215 nsec. Gates G3, G4 and G5 determine the quantization segment while the correct quantization level value is retrieved from the appropriate ROM designated L0 through L7. Together with the 13-bit to 8-bit code conversion which follows the total cycle time is 865 nsec. The entire 6th order filter process is completed in 1730 nsec. This corresponds to the time available to process 24 channels at the sampling rate of 24 KHz.

CHAPTER V

SIMULATION AND ANALYSIS

5.1 General

The first essential part of this chapter deals with the basic method used in the simulation and analysis of the digital filter. The analysis extends to include the effects of quantization, truncation and roundoff.

The latter part of the chapter lists and describes the main programs and subroutines developed for this purpose.

5.2 Simulation

Simulation here refers to the simulation of all arithmetic operations performed by the digital filter in the exact sequence of occurance in an actual physical unit. The input samples and the results of each arithmetic operation are quantized, rounded or truncated as required by the register lengths and arithmetic accuracy.

The operations necessary for the third order section are shown in Fig. 5.1. The operations of addition, multiplication and unit delay are ideal and introduce no error. In addition the following operations are performed:

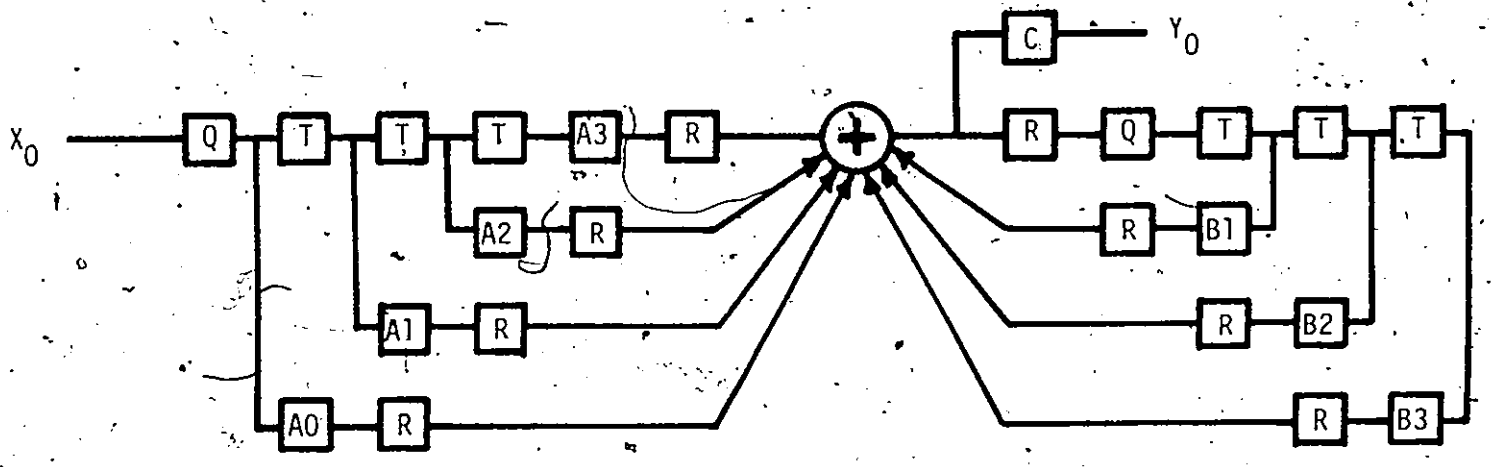
(1) Quantization of the input samples X_0.

(2) Roundoff of the products.

(3) Roundoff of the sum.

(4) Quantization of the output samples Y_0.

The operation designated by C in the diagram is the code compression operation from the 13-bit to the 8-bit code and introduces no additional error.



R — roundoff
Q — quantization
C — code compression
T — unit delay
Ai,Bi — multiplication

FIG. 5.1 — Error sources in the third order section

The filter is specified in the data file in terms of its exact coefficients. A discrete sinusoidal input signal is introduced by specifying the frequency, amplitude and the sampling rate. The input samples can be quantized or left in the continuously variable amplitude form. If quantization is required it can be done according to the level assignment of Table 3.1 or Table 4.8.

The degree of truncation or roundoff is specified in terms of the number of binary fractional digits. The actual arithmetic is done in the decimal system but the rounding and truncation are carried out to the decimal equivalent of the binary digit specified.

5.3 Analysis

The main property of the digital filter considered in the analysis is its insertion loss behaviour with frequency. To obtain the insertion loss of the filter the RMS values of the samples of the input and output sinusoids are computed. The insertion loss is then given by

$$I.L.(dB) = 20 \log_{10} \frac{(V_o)_{RMS}}{(V_{IN})_{RMS}} \qquad (1)$$

In computing the insertion loss by this method care must be taken to include all possible sample values. This can be done by averaging over the correct number of cycles of the input and output signals as dictated by the ratio of the signal frequency to the sampling frequency. If F1 is the signal frequency and F is the sampling frequency then any integral multiple of F/F1 cycles can be used.

To establish the steady state condition several cycles of input are processed but not included in the computation of the insertion loss. For the third order section 300 samples were found sufficient to reach the steady state at all frequencies of interest. Table 5.1 gives the comparison between the insertion loss responses computed by two methods. LOSS1 is computed directly from the transfer function $H(Z)$, $Z=e^{j\omega T}$ and LOSS2 is computed by processing the sampled sinusoidal input followed by the conversion of the input and output signals to their RMS values.

TABLE 5.1

FREQ. Hz	LOSS1 dB	LOSS2 dB
500	.003	.003
1000	.006	.006
2500	.003	.003
3000	.013	.013
3500	4.897	4.896
4000	28.755	28.755
4500	33.959	33.960
5000	44.853	44.853
6000	34.572	34.572
7000	36.048	36.047
8000	150.957	150.956

Sampling Frequency F = 16 KHz

Phase and delay as well as the transient response are also available from the simulation and analysis program but were not used in the evaluation of the filter.

5.4 The Programs

The listings of the main programs and subroutines are given at the end of this section in the following order:

SIM4

SIGNAL

QUANT

QUANTI

ROUND

BNRND

CHEB1

DIGITAL

SIM4 is the main simulation program while the remaining programs with the exception of DIGITAL are subroutines called by SIM4. The SIGNAL subroutine generates a sampled sine wave of specified frequency, amplitude and sampling rate. Subroutines QUANT and QUANTI quantize a number of any value according to the level assignment of Table 3.1 and Table 4.8 respectively. ROUND and BNRND provide decimal and binary roundoff to a specified number of fractional digits. CHEB1 computes the insertion loss response of a third order Chebyshev low pass filter for a given pass band ripple. This subroutine is called by SIM4 when the insertion loss of the digital and the analog filters is to be added for a combined response output.

The program DIGITAL computes and prints out the ideal coefficients and insertion loss response of a direct form digital filter designed on the basis of the bilinear transform. The input to the program includes the order of the analog prototype filter, its critical frequencies and the sampling rate.

APPENDIX

FILE: SIM4 FORTRAN A BELL-NORTHERN RESEARCH

```
C     TWO THIRD ORDER FILTERS IN TANDEM
C     THIS PROGRAM PRINTS OUT THE INSERTION LOSS AS A FUNCTION OF
C     FREQUENCY.  IT DOES NOT PRINT OUT THE SAMPLES IN AND SAMPLES
C     OUT FOR A PARTICULAR FREQUENCY F1.  THE INPUT DATA
C     MUST SPECIFY THE FREQUENCY INCREMENTS: FMIN,FDEL,FMAX.
C     THIS PROGRAM ADDS THE RESPONSE OF A 3RD ORD CHEB FILTER.
      DIMENSION A(10),IOP(80),S(20)
      NINP=1
      NOUT=8
      NDATA=7
      CALL FREES(A,IOP,NINP,NDATA)
      INPUT=A(1)
      IPROD=A(2)
      BPR1=A(3)
      BPR2=A(4)
      ISUM=A(5)
      BP=A(6)
      IOUT=A(7)
      WRITE(NOUT,11) INPUT,IPROD,BPR1,BPR2,ISUM,BP,IOUT
11    FORMAT(2I2,2F8.0,I2,F3.0,I2)
      NDATA=4
      CALL FREES(A,IOP,NINP,NDATA)
      AMPL=A(1)
      F=A(2)
      M=A(3)
      WRITE(NOUT,1) AMPL,F,M
1     FORMAT(F14.7,F10.0,I5)
      NDATA=3
      CALL FREES(A,IOP,NINP,NDATA)
      FMIN=A(1)
      FDEL=A(2)
      FMAX=A(3)
      WRITE(NOUT,12) FMIN,FDEL,FMAX
12    FORMAT(3F10.0)
      NDATA=6
      CALL FREES(A,IOP,NINP,NDATA)
      A0=A(1)
      A1=A(2)
      A2=A(3)
      A3=A(4)
      A4=A(5)
      A5=A(6)
      NDATA=6
      CALL FREES(A,IOP,NINP,NDATA)
      B1=A(1)
      B2=A(2)
      B3=A(3)
      B4=A(4)
      B5=A(5)
      WRITE(NOUT,2) A0,A1,A2,A3,A4,A5
2     FORMAT(6E14.7)
      WRITE(NOUT,3) B1,B2,B3,B4,B5
3     FORMAT(5E14.7)
      F1=FMIN
77    X7=0.
```

FILE: SIM4 FORTRAN A BELL-NORTHERN RESEARCH

```
      X6=0.
      X5=0.
      X4=0.
      X3=0.
      X2=0.
      X1=0.
      X0=0.
      Y7=0.
      Y6=0.
      Y5=0.
      Y4=0.
      Y3=0.
      Y2=0.
      Y1=0.
      Y0=0.
      ENT=0.
      V1=0.
      V2=0.
      N=0
      G=0.
5     SUM=0.
      SUM2=0.
      EN=N
      IF(INPUT.EQ.0) GO TO 6
      CALL SIGNAL(AMPL,N,F1,F,XX)
      X0=XX
      GO TO 7
6     X0=AMPL*COS(6.2831853*F1*EN/F)
7     S(1)=A0*X0
      S(2)=A1*X1
      S(3)=A2*X2
      S(4)=A3*X3
      S(5)=-B1*Y1
      S(6)=-B2*Y2
      S(7)=-B3*Y3
      IF(IPROD.NE.1) GO TO 9
      DO 8 I=1,7
      X=S(I)
C     BPR1 IS THE NUMBER OF BINARY FRACTIONAL DIGITS TO WHICH
C     EACH PRODUCT IS ROUNDED OFF TO BEFORE ADDITION.
      BPR=BPR1
      CALL BNRND(BPR,X)
8     S(I)=X
9     DO 10 I=1,7
      IF(ISUM.NE.1) GO TO 10
      X=SUM
C     BP IS THE NUMBER OF BINARY FRACTIONAL DIGITS TO WHIC
C     EACH PARTIAL SUM IS TRUNCATED AFTER EACH PRODUCT IS ADDED
C     TO THE CUMULATIVE SUM IN THE ADDER ACCUMULATOR.
      CALL BNTRNC(BP,X)
      SUM=X
10    SUM=SUM+S(I)
      IF(ISUM.NE.1) GO TO 14
C     BPR2 IS THE NUMBER OF BINARY FRACTIONAL DIGITS TO WHICH
C     THE FINAL SUM IS ROUNDED OFF TO AFTER ALL ELEVEN
```

FILE: SIM4 FORTRAN A BELL-NORTHERN RESEARCH

```
C       PRODUCTS ARE ADDED.
        BPR=BPR2
        X=SUM
        CALL BNRND(BPR,X)
        SUM=X
14      IF(IOUT.EQ.0) GO TO 15
        Y=SUM
        CALL QUANT(Y)
        SUM=Y
15      Y0=SUM
        X3=X2
        X2=X1
        X1=X0
        Y3=Y2
        Y2=Y1
        Y1=Y0
C       THE SECOND FILTER
        X4=Y0
        S(8)=A0*X4
        S(9)=A1*X5
        S(10)=A2*X6
        S(11)=A3*X7
        S(12)=-B1*Y5
        S(13)=-B2*Y6
        S(14)=-B3*Y7
        IF(IPROD.NE.1) GO TO 109
        DO 108 I=8,14
        X=S(I)
        BPR=BPR1
        CALL BNRND(BPR,X)
108     S(I)=X
109     DO 110 I=8,14
        IF(ISUM.NE.1) GO TO 110
        X=SUM2
        CALL BNTRNC(BP,X)
        SUM2=X
110     SUM2=SUM2+S(I)
        IF(ISUM.NE.1) GO TO 114
        BPR=BPR2
        X=SUM2
        CALL BNRND(BPR,X)
        SUM2=X
114     IF(IOUT.EQ.0) GO TO 115
        Y=SUM2
        CALL QUANT(Y)
        SUM2=Y
115     YOUT=SUM2
        Y7=Y6
        Y6=Y5
        Y5=YOUT
        X7=X6
        X6=X5
        X5=X4
        N=N+1
        ENT=ENT+1./F
```

FILE: SIM4 FORTRAN A BELL-NORTHERN RESEARCH

```
      IF(N.LE.500) GO TO 5
      G=G+1.
20    FORMAT(3E14.7)
C     NEXT TWO IF STATEMENTS DETERMINE WHETHER TO NORMALIZE X0 AND
C     YOUT WITH RESPECT TO .A  FROM SUBROUTINE SIGNAL TO IMPROVE
C     THE ACCURACY OF COMPUTING RMS VALUES OF X0 AND YOUT.
      AA=AMPL*2015.5/2.5
      IF(INPUT.NE.1) GO TO 25
      X00=X0/AA
      V1=V1+X00**2
      GO TO 27
25    V1=V1+X0**2
27    IF(IOUT.NE.1) GO TO 26
      Y00=YOUT/AA
      V2=V2+Y00**2
      GO TO 28
26    V2=V2+YOUT**2
28    IF(G.LT.M*F/F1) GO TO 5
      RMSV1=SQRT(V1/G)
      RMSV2=SQRT(V2/G)
30    FORMAT(2E14.7)
      DB=20.*ALOG10(RMSV1/(RMSV2))
      CALL CHEB1(F1,DB1)
      DB=DB+DB1
      WRITE(NOUT,40) F1,DB
40    FORMAT(F10.0,F10.3)
      F1=F1+FDEL
      IF(F1.LT.FMAX+FDEL) GO TO 77
      STOP
      END
```

FILE: SIGNAL FORTRAN A BELL-NORTHERN RESEARCH

```
C       THIS ROUTINE COMPUTES THE N'TH QUANTIZED SAMPLE OF A
C       COSINE OF SPECIFIED AMPLITUDE=AMPL AND FREQUENCY FI
C       FOR A SPECIFIED SAMPLING FREQUENCY OF F HZ.
C       QUANTIZATION IS ACCORDING TO A-LAW AND A MAXIMUM SIGNAL OF
C       VOLTS PEAK-TO-PEAK
C
        SUBROUTINE SIGNAL(AMPL,N,FI,F,XX)
        A=AMPL*2015.5/2.5
        TPI=6.2831853
        T=1./F
        EN=N
        XX=A*COS(TPI*FI*EN*T)
        X=XX
        CALL QUANT(X)
        XX=X
        RETURN
        END
```

FILE: QUANT FORTRAN A BELL-NORTHERN RESEARCH

```
      SUBROUTINE QUANT(Y)
      K=0
      IF(Y.GE.0.) GO TO 5
      K=-1
      Y=-Y
5     IF(Y.GE.1000.) GO TO 11
      IF(Y.GE.100.) GO TO 12
      IF(Y.GE.10.) GO TO 13
      IF(Y.GE..5) GO TO 14
      IF(Y.LT.5) Y=0.
11    SF=4.
      GO TO 30
12    SF=3.
      GO TO 30
13    SF=2.
      GO TO 30
14    SF=1.
30    X=Y
      CALL ROUND(SF,X)
      Y=X
      L=0
      IF(Y.GE.15.5) L=L+1
      IF(Y.GE.31.5) L=L+1
      IF(Y.GE.63.5) L=L+1
      IF(Y.GE.127.5) L=L+1
      IF(Y.GE.255.5) L=L+1
      IF(Y.GE.511.5) L=L+1
      IF(Y.GE.1023.5) L=L+1
      IF(Y.LT.2047.5) GO TO 35
      IF(Y.GE.2047.5) Y=2015.5
      GO TO 55
35    IF(L.GT.0) GO TO 43
      IF(L.EQ.0) V=Y
      GO TO 44
43    V=(Y+.5)/(2.**(L-1))-16.5
44    IF(V.GE.9.5) GO TO 45
      IF(V.GE.1.) GO TO 46
      IF(V.GE..5) GO TO 47
      IF(V.GE.0.) GO TO 48
45    SF=2.
      GO TO 50
46    SF=1.
      GO TO 50
47    V=1.
      GO TO 50
48    V=0.
50    X=V
      CALL ROUND(SF,X)
      V=X
      IF(L.GT.0) GO TO 53
      IF(L.EQ.0.0) Y=V
      GO TO 54
53    Y=(2.**(L-1))*(V+16.5)-.5
54    IF(K.EQ.0.0) GO TO 55
      K=0
```

```
      Y=-Y
55    RETURN
      END
```

FILE: QUANT1 FORTRAN A BELL-NORTHERN RESEARCH

```
      SUBROUTINE QUANT1(Y)
      K=0
      IF(Y.GE.0.) GO TO 5
      K=-1
      Y=-Y
5     IF(Y.GE.1000.) GO TO 11
      IF(Y.GE.100.) GO TO 12
      IF(Y.GE.10.) GO TO 13
      IF(Y.GE..5) GO TO 14
      IF(Y.LT.5) Y=0.
11    SF=4.
      GO TO 30
12    SF=3.
      GO TO 30
13    SF=2.
      GO TO 30
14    SF=1.
30    X=Y
      CALL ROUND(SF,X)
      Y=X
      L=0
      IF(Y.GE.15.5) L=L+1
      IF(Y.GE.31.5) L=L+1
      IF(Y.GE.63.5) L=L+1
      IF(Y.GE.127.5) L=L+1
      IF(Y.GE.255.5) L=L+1
      IF(Y.GE.511.5) L=L+1
      IF(Y.GE.1023.5) L=L+1
      IF(Y.GE.2047.5) L=L+1
      IF(Y.LT.4095.5) GO TO 35
      IF(Y.GE.4095.5) Y=4031.5
      GO TO 55
35    IF(L.GT.0) GO TO 43
      IF(L.EQ.0) V=Y
      GO TO 44
43    V=(Y+.5)/(2.**(L-1))-16.5
44    IF(V.GE.9.5) GO TO 45
      IF(V.GE.1.) GO TO 46
      IF(V.GE..5) GO TO 47
      IF(V.GE.0.) GO TO 48
45    SF=2.
      GO TO 50
46    SF=1.
      GO TO 50
47    V=1.
      GO TO 60
48    V=0.
50    X=V
      CALL ROUND(SF,X)
      V=X
      IF(L.GT.0) GO TO 53
      IF(L.EQ.0) Y=V
      GO TO 54
53    Y=(2.**(L-1))*(V+16.5)-.5
54    IF(K.EQ.0) GO TO 55
```

FILE: QUANT1 FORTRAN A BELL-NORTHERN RESEARCH

```
      K=0
      Y=-Y
55    RETURN
      END
```

FILE: ROUND FORTRAN A BELL-NORTHERN RESEARCH

```
      SUBROUTINE ROUND(SF,X)
      IF(X.GE.10000.) GO TO 11
      IF(X.GE.1000.) GO TO 12
      IF(X.GE.100.) GO TO 13
      IF(X.GE.10.) GO TO 14
      IF(X.GE.1.) GO TO 15
      IF(X.GE..1) GO TO 16
      IF(X.GE..01) GO TO 17
      IF(X.GE..001) GO TO 18
      IF(X.LT..001) GO TO 100
11    XX=10.**(5-SF)
      GO TO 2
12    XX=10.**(4-SF)
      GO TO 2
13    XX=10.**(3-SF)
      GO TO 2
14    XX=10.**(2-SF)
      GO TO 2
15    XX=10.**(1-SF)
      GO TO 2
16    XX=10.**(0-SF)
      GO TO 2
17    XX=10.**(-1-SF)
      GO TO 2
18    XX=10.**(-2-SF)
2     IX=X/XX+.5
      X=XX*IX
100   RETURN
      END
```

FILE: BNRND FORTRAN 4 BELL-NORTHERN RESEARCH

```
      SUBROUTINE BNRND (BPR,X)
      R=2.**BPR
      IX=X*R+.5
      X=IX/R
      RETURN
      END
```

FILE: CHEB1 FORTRAN A BELL-NORTHERN RESEARCH

```
C       THIS SUBROUTINE COMPUTES THE INSERTION LOSS OF A THIRD ORDER
C       CHEBYSHEV POLYNOMIAL FILTER FOR RIPL=.05DB AND CUTOFF
C       FREQUENCY F0=2900 HZ AT ANY SPECIFIED FREQUENCY OF F1 HZ.
        SUBROUTINE CHEB1(F1,DB1)
        RIPL=.05
        F0=2900.
        F=F1/F0
        REPS2=(10.)**(RIPL/10.)-1.
        CN=4.*(F**3)-3.*F
        CN2=CN**2
        H2=1./(1.+REPS2*(  2)
        DB1=10.*ALOG10(1./H2)
        RETURN
        END
```

FILE: DIGITAL FORTRAN A BELL-NORTHERN RESEARCH

```
      DIMENSION X(10),IOP(80)
      NINP = 1
      NOUT = 8
      NDATA = 3
      CALL FREES(X,IOP,NINP,NDATA)
      FMIN = X(1)
      FDEL = X(2)
      FMAX = X(3)
      WRITE(NOUT,1) FMIN, FDEL, FMAX
1     FORMAT(3F10.1)
      NDATA=2
      CALL FREES(X,IOP,NINP,NDATA)
      F0=X(1)
      FS=X(2)
      WRITE(NOUT,5) F0,F
5     FORMAT(F10.1,F10.1)
      NDATA=1
      CALL FREES(X,IOP,NINP,NDATA)
      IORD=X(1)
      WRITE(NOUT,3) IORD
3     FORMAT(I1)
      NDATA = 1
      CALL FREES(X,IOP,NINP,NDATA)
      W1 = X(1)
      WRITE(NOUT,2) W1
2     FORMAT(F10.6)
      NDATA = 1
      CALL FREES(X,IOP,NINP,NDATA)
      P1 = X(1)
      WRITE(NOUT,2) P1
      NDATA = 2
      CALL FREES(X,IOP,NINP,NDATA)
      A1 = X(1)
      B1 = X(2)
      WRITE(NOUT,4) A1,B1
4     FORMAT(2F10.6)
      IF(IORD.EQ.3) GO TO 10
      NDATA=1
      CALL FREES(X,IOP,NINP,NDATA)
      W2=X(1)
      WRITE(NOUT,102) W2
102   FORMAT(F10.6)
      NDATA=2
      CALL FREES(X,IOP,NINP,NDATA)
      A2=X(1)
      B2=X(2)
      WRITE(NOUT,104) A2,B2
104   FORMAT(2F10.6)
10    F = FMIN
6     CALL DRES(W1,P1,A1,B1,F0,FS,F,DB,CK,R0,R1,R2,R3,Q0,Q1,Q2,Q3
     *,W2,A2,B2,RR0,RR1,RR2,QQ0,QQ1,QQ2,IORD)
      WRITE(NOUT,7) F,DB
7     FORMAT(F10.0,F10.3)
      F = F + FDEL
      IF(F.LT.FMAX) GO TO 6
```

FILE: DIGITAL FORTRAN A BELL-NORTHERN RESEARCH

```
      IF(IORD.EQ.5) GO TO 78
      A0=CK*R3/Q3
      A1=CK*R2/Q3
      A2=CK*R1/Q3
      A3=CK*R0/Q3
      B0=1.
      B1=Q2/Q3
      B2=Q1/Q3
      B3=Q0/Q3
      WRITE(NOUT,77) A0,A1,A2,A3
      WRITE(NOUT,77) B0,B1,B2,B3
77    FORMAT(4E14.7)
78    TPI=6.283185
      W0=TPI*F0
      P1=W0*P1
      W1=W0*W1
      W2=W0*W2
      A1=W0*A1
      A2=W0*A2
      B1=W0*B1
      B2=W0*B2
      CK=W0*CK
      T=TPI/(FS/F0)
      G=W0/(TAN(T/2.))
      BB1=G-P1
      BB0=G-P1
      G2=G*G
      AB1=A1**2+B1**2
      AB2=A2**2+B2**2
      R2=G2+W1**2
      R1=-2.*G2+2.*W1**2
      R0=G2+W1**2
      RR2=G2+W2**2
      RR1=-2.*G2+2.*W2**2
      RR0=G2+W2**2
      Q2=G2-2.*A1*G+AB1
      Q1=-2.*G2+2.*AB1
      Q0=G2+2.*A1*G+AB1
      QQ2=G2-2.*A2*G+AB2
      QQ1=-2.*G2+2.*AB2
      QQ0=G2+2.*A2*G+AB2
      R4=R2*RR2
      R3=R2*RR1+R1*RR2
      R2=R2*RR0+R1*RR1+R0*RR2
      R1=R1*RR0+R0*RR1
      R0=R0*RR0
      Q4=Q2*QQ2
      Q3=Q2*QQ1+Q1*QQ2
      Q2=Q2*QQ0+Q1*QQ1+Q0*QQ2
      Q1=Q1*QQ0+Q0*QQ1
      Q0=Q0*QQ0
      R5=CK*R4
      R4=CK*(R3+R4)
      R3=CK*(R2+R3)
      R2=CK*(R1+R2)
```

REFERENCES

(1) F. Meyer, "Design of a Multiplexing System for a Pulse Code Modulated Telephone Carrier System". Report to the Association of Professional Engineers of the Province of Ontario, Sept. 1972.

(2) K. W. Cattermole, "Principles of Pulse Code Modulation", New York, American Elsevier, 1969.

(3) Northern Electric Co. Ltd., "LD-4 Long Haul Digital Coaxial Cable System", Bulletin TB312

(4) W. M. Penney, "MOS Integrated Circuits, Theory, Fabrication, Design and Applications to MOS LSI", New York, Van Nostrand Reinhold, 1972.

(5) H. Kaneko, "A Unified Formulation of Segment Companding Laws and Synthesis of Codecs and Digital Compandors", Bell Syst. Tech. J., Vol. 47, No. 7, Sept. 1970.

(6) B. Liu, "Effect of Finite Word Length on the Accuracy of Digital Filters -- A Review", IEEE Trans. Circuit Theory, Vol. CT-18, Nov. 1971.

(7) H. M. Bender and O. P. Mahajan, "PCM Compandor Evaluations", Tech. Memo TM8336-1-69, Northern Electric Research and Development Laboratories, June 1969.

(8) M. R. Aaron and K. H. Kaneko "Synthesis of Digital Attenuators for Segment-Companded PCM Codes", IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, Dec. 1971.

(9) Members of the Technical Staff, Bell Telephone Laboratories, "Transmission Systems for Communications", Bell Telephone Laboratories, Inc., Fourth Edition.

(10) J.A.C. Bingham, "Specifications and Design of Channel Filters for PCM", IEEE Conf. on Commun., 1968.

(11) J. F. Kaiser, "Some Practical Considerations in the Realization of Linear Digital Filters", Proc. 3rd Annu. Allerton Conf. Circuit System Theory, 1963.

(12) C. M. Rader and B. Gold, "Digital Processing of Signals", New York, McGraw-Hill, 1969.

(13) L. B. Jackson, "An Approach to Implementation of Digital Filters", IEEE Trans. Audio Electroacoust., Vol. AU-16, Sept. 1968.

(14) J. F. Kaiser, "Some practical Considerations in the Realization of Linear Digital Filters", Proc. 3rd Annu. Allerton Conf. Circuit System Theory, 1963.

(15) L. B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters:, Bell Syst. Tech. J., Vol. 49, 1970.

(16) L. B. Jackson, "Roundoff-noise Analysis for Fixed-Point Digital Filters Realized in Cascade or Parallel Form", IEEE Trans. Audio Electroacoust., Vol. AU-18, June 1970.

(17) W. N. Carr and J. P. Mize, "MOS/LSI Design and Application", Texas Instruments Electronics Series, McGraw-Hill, 1972.

(18) W. Neu and A. Kundig, "Project for a Digital Telephone Network", IEEE Trans. Commun. Technol., Vol. EOM-16, No. 5, Oct. 1968.

(19) A. Kundig, "Digital Filtering in PCM Telephone Systems", IEEE Trans. Audio and Electroac., Vol. AU-18, No. 4, December 1970.

(20) W. L. Montgomery, "Digitally Linearizable Compandors with Comments on "Project for a Digital Telephone Network"", IEEE Trans. on Communic. Tech., Vol. COM-18, No. 1, Feb. 1970.

(21) G. A. Maley, "Manual of Logic Circuits", Prentice-Hall, 1970.

(22) RCA, "COS/MOS Digital Integrated Circuits", Databook Series SSD-203, 1972.

(23) R. J. Schwarz and B. Friedland, "Linear Systems", McGraw-Hill, 1965.

FILE: DIGITAL FORTRAN A BELL-NORTHERN RESEARCH

```
      IF(IORD.EQ.5) GO TO 78
      A0=CK*R3/Q3
      A1=CK*R2/Q3
      A2=CK*R1/Q3
      A3=CK*R0/Q3
      B0=1.
      B1=Q2/Q3
      B2=Q1/Q3
      B3=Q0/Q3
      WRITE(NOUT,77) A0,A1,A2,A3
      WRITE(NOUT,77) B0,B1,B2,B3
77    FORMAT(4E14.7)
78    TPI=6.2831853
      W0=TPI*F0
      P1=W0*P1
      W1=W0*W1
      W2=W0*W2
      A1=W0*A1
      A2=W0*A2
      B1=W0*B1
      B2=W0*B2
      CK=W0*CK
      T=TPI/(FS/F0)
      G=W0/(TAN(T/2.))
      BB1=G-P1
      BB0=G-P1
      G2=G*G
      AB1=A1**2+B1**2
      AB2=A2**2+B2**2
      R2=G2+W1**2
      R1=-2.*G2+2.*W1**2
      R0=G2+W1**2
      RR2=G2+W2**2
      RR1=-2.*G2+2.*W2**2
      RR0=G2+W2**2
      Q2=G2-2.*A1*G+AB1
      Q1=-2.*G2+2.*AB1
      Q0=G2+2.*A1*G+AB1
      QQ2=G2-2.*A2*G+AB2
      QQ1=-2.*G2+2.*AB2
      QQ0=G2+2.*A2*G+AB2
      R4=R2*RR2
      R3=R2*RR1+R1*RR2
      R2=R2*RR0+R1*RR1+R0*RR2
      R1=R1*RR0+R0*RR1
      R0=R0*RR0
      Q4=Q2*QQ2
      Q3=Q2*QQ1+Q1*QQ2
      Q2=Q2*QQ0+Q1*QQ1+Q0*QQ2
      Q1=Q1*QQ0+Q0*QQ1
      Q0=Q0*QQ0
      R5=CK*R4
      R4=CK*(R3+R4)
      R3=CK*(R2+R3)
      R2=CK*(R1+R2)
```