CANADIAN THESES ON MICROFICHE

I.S.B.N.

THESES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL-339 (r. 82/08)

Canada

A STORED-PRODUCT DIGITAL FILTER WITH LOGARITHMIC SIGNAL QUANTIZATION

by

O. Monkewich, B. Eng., M.A.Sc.

A thesis submitted to the Faculty of Graduate Studies in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Electrical Engineering
University of Ottawa
Ottawa, Ontario
December, 1981

ABSTRACT

The Stored-Product Digital Filter with internal logarithmic signal quantization was shown to have the highest speed-to-power ratio of all major, digital filter implementations using conventional multipliers. The design details for a fully functional implementation of the filter were worked out in order to determine its LSI implementation potential. The effect of the logarithmic signal quantizer on limit cycles and output noise were derived and shown to be within the design limits of useful applications. The process of stored product rounding in lieu of separate coefficient and product rounding was shown to have accuracy advantages for some design methods. Sufficient simulation and measurement data were obtained to conclude that the filter can meet PCM voice channel specifications with potential 24-channel multiplexing capability.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Willem Steenaart, research director of this project, for his technical guidance and years of encouragement. The author also wishes to thank his advisory committee, Dr. S.S. Stuchly and Dr. M. Goldberg.

The author gratefully acknowledges the financial assistance of the National Sciences and Engineering Council under the Grant No. A8572 and the Federal Department of Communications educational program which made the completion of this project possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.01 General

In the early sixties it became practical to represent analog signals in terms of digital codes and to perform spectrum filtering by arithmetic manipulation of these codes using the already well established techniques of digital computers. Unlike analog and sampled data filters, filters implemented in digital hardware have very few performance limitations; however, typical signal processing requirements can only be achieved at the cost of considerable computing time. For many applications, such as tone separation and speech analysis, characterized by either simple requirements or non-real time processing, the long computation time is not a serious limitation. A far greater potential for digital filters lies in the communications industry where currently thousands of analog filters are being used as part of real-time voice signal processing apparatus (12).

Filters of voice channel quality are characterized by high-order transfer functions which require hardware complexities of the order of several thousand logic gates to implement. This high complexity results from the need to compute the filter transfer function to a high arithmetic accuracy during each sampling interval and the need to accomodate high sampling rates associated with good quality channels. Hardware duplication and the associated power dissipation are usually tolerated in order to achieve the desired processing speeds. There is an obvious advantage in finding a filter implementation which for a given accuracy offers a large speed improvement over the conventional methods while reducing the overall power dissipation. This work describes such a filter implementation and demonstrates that its signal processing qualities are as good as and in some aspects superior to those of conventional structures.

1.02 Speed-Power Requirements

Every digital filter implementation is designed to perform a sequence of binary multiplications and additions conforming to well defined accuracy considerations. By definition, multiplication is repeated addition and therefore is the slower of the two operations. The overall speed-power performance of a digital filter is, therefore, determined by the choice of the multiplication algorithm and the nature of its implementation. A currently available 12-bit by 12-bit multiplier on a chip performs a full multiplication

according to the Booth algorithm in 675ns and dissipates 564 mWatts. A 12-bit addition, however, takes on the average 42ns with power dissipation of about 224 mWatts.

Using the above figures it is possible to form a reasonable estimate of the total power dissipation and hardware complexity needed to satisfy voice channel requirements. Typical voice quality filtering can be achieved with a fifth-order transfer function which, in general, can be represented by eleven multiplications and eleven additions computed to a 12-bit accuracy. To compute such a function using conventional methods would require 8.44us and a power dissipation of 3.6 Watts. Even at a minimum sampling rate of 24KHz, five such filters would be required in a 24-channel system, resulting in the unacceptably high power dissipation of over 18 Watts.

Although there exists a certain amount of freedom in trading off power against speed, the speed/power ratio remains substantially constant for all conventional implementations.

1.03 Stored-Product Digital Filter

The alternative proposed by the author is the Stored-Product Digital Filter (SPDF). SPDF is the author's invention and

is described in the Canadian Patent No. 1,050,619 issued on March 13. 1979. The principle of SPDF is based on storing of products in Read Only Memory (ROM) and accessing the required product by using its multiplicand as the address. This approach is made practical by recognizing that in most digital systems not all of the possible products are needed to maintain a specified performance. For example, in PCM, only 256 distinct multiplicands are present. Thus, for a given multiplier in a 12-bit system, all products can be stored in a 3072-bit ROM. Such ROM is currently available in low-power CMOS with access time of under 150ns and power dissipation of about 0.180 Watts.

The requirements of a 24-channel PCM system can be satisfied by a tendem connection of two identical third-order filter sections, only one of which must be implemented in hardware. Such a sixth-order transfer function can be implemented with only five distinct coefficients and can be computed in about 1.26us with a total power dissipation of about 0.20 Watts. This results in a speed/power ratio of 3.87×10^6 sec/sample/Watt. The corresponding speed/power ratio for the conventional implementation is 0.068×10^6 sec/sample/Watt. The SPDF, therefore, represents a speed/power ratio improvement of nearly 57 times.

The SPDF architecture leading to the improved speed/power ratio and its importance in communication are presented in detail in Chapter 3.

1.04 Further Contributions

In addition to the speed/power ratio improvement, the author's contributions lie in the areas of digital filter architecture, the use of logarithmic quantization of signals in digital filters, the study of limit cycles due to logarithmic signal quantization in the recursive loop, analysis of error effects due to rounding of store andproducts, implementation of a time-shared digital filter for PCM application and improvements in digital filters leading to easier LSI implementation.

The use of Read Only Memory for the replacement of multipliers introduces important architectural flexibilities into digital filter implementation. A very important class of transfer functions derived by means of the bilinear z-transform are characterized by several non-distinct coefficients. Products corresponding to all equal coefficients may be stored in a single ROM table allowing further savings in hardware and power dissipation.

In SPDF filters the products corresponding to all coefficients may be accessed simultaneously. This means that further speed improvements are possible by introducing more adders into the filter architecture to process the available products. Conventional multiplier filters, however, must introduce additional multipliers as well as additional adders in order to increase the processing speed.

These architectural aspects have been published in references (1) and (21) and are presented in Chapter 2 and Chapter 3.

The use of special signal quantization techniques, such as the segmented A-law, was introduced by the author in order to reduce the total ROM capacity of the filter. In Chapter 4 the digital implementation of the adapted A-law characteristic is analysed and completely described in terms of mathematical expressions suitable for digital implementation. The related work was published in references (17) and (18).

For certain ranges of coefficients, quantization of the signal in the recursive loop results in the effect known as "deadband effect". This effect leads to limit cycle oscillations. In many communications applications limit cycles are undesirable because they result in a signal output after the input signal is removed. Chapter 5 presents the investigation into the nature of limit cycles in the presence of the logarithmic quantizer in the recursive loop of the filter. It shows that large amplitude limit cycles are restricted to a very small range of coefficient values and that high quality filters can be designed with coefficients outside this range. The contents of Chapter 5 were published in reference (22).

In digital filter implementations using conventional multi-

pliers, the coefficient rounding and product rounding are performed as separate operations, governed by different criteria and contribute to roundoff noise as two independent effects. In SPDF filter implementation the coefficient accuracy remains arbitrarily high when the products are computed. Product rounding prior to ROM programming remains as the only source of roundoff noise. This represents a definite noise advantage. For example, the average roundoff noise for a system in which the product and signal accuracy are 12 bits while the coefficient accuracy is 20 bits is lower than for the same system in which the coefficient accumacy is also 12 bits. The details of this result were published in reference (24) and are presented in Chapter 6.

Although a great deal of theoretical and experimental work has been done to date in digital filters, most of the implementations have been restricted to the laboratory studies. With only a few minor exceptions the digital filters do not exist as products in industrial communications systems. This is primarily due to the hardware complexity and high power dissipation associated with such devices and not due to any signal processing limitations. The author's contribution in this area has been the adaptation of the proposed SPDF filter to the PCM application. Chapter 7 presents the practical results associated with building and testing of the PCM filter. Some of the results of Chapter 7 were published in (18) and (21).

Chapter 7 also discusses the LSI implementation of the filter reported in references (2), (19) and (20).

1.05 Summary of Conclusions

The proposed SPDF filter implementation offers the highest speed/power ratio of all known conventional implementations using multipliers. At the same time, SPDF signal processing characteristics remain compatible with the high quality requirements of the voice channel filters. Specifically, SPDF is capable of satisfying the PCM filtering requirements for a 24-channel system at a cost lower than that of the currently used technologies.

CHAPTER 2

DIGITAL MULTIPLIERS IN DIGITAL FILTERS

2.01 General

The architecture of the conventional multipliers has been evolving with the progress in digital computers and is well documented by G.A. Blaauw (4). The multipliers are divided into two major categories, clocked multipliers and array multipliers. The clocked multipliers operate on the add-shift principle and usually make use of the Booth's algorithm for negative numbers (14). The fastest multipliers consist of a two-dimensional array of one-bit adders and are refered to as the array multipliers. The speed of the multipliers in both categories is increased by adding look-ahead logic in the adders and in effect doubling the complexity. The main drawback of the conventional multipliers is that they are structured to generate all possible products within the accuracy range of the operands whether or not all such products actually occur in a specific application. A second disadvantage is that separate

arithmetic rounding algorithms cannot be applied to the individual products generated by the same multiplier.

This chapter presents several important multiplication algorithms used in digital multiplier implementations and describes how they are used in digital filter architectures. A speed/power ratio is defined to provide a basis for comparing the various implementations. Although the processing speed and power dissipation of a given implementation can be traded off one against the other, the speed/power ratio remains essentially unchanged unless innovative improvements are introduces.

2.02 Clocked Multipliers

In the majority of prominent multiplication algorithms the operands are treated as integers represented in two's complement notation. Given a binary integer, X, and its two's complement, $\overline{X}$, the defining property of two's complement arithmetic can be stated as

$$X + \overline{X} = 0 + \text{overflow} \qquad 2.01$$

Formally, two's complement representation of a binary integer, X, is expressed as

$$X = -2^n x_n + \sum_{i=0}^{n-1} 2^i x_i \qquad 2.02$$

where $x_i \in \{0,1\}$

Booth's algorithm for multiplication makes use of the two's complement notation to eliminate the extra complementing cycles required for sign conversions. In this way it increases multiplication speed and reduces multiplier hardware complexity. If one operand, Y, initially in the form of Equation 2.02, is rearranged into the form

$$Y = 2^m(y_{m-1} - y_m) + \sum_{j=0}^{m-1} 2^j(y_{j-1} - y_j) \qquad 2.03$$

then the product XY can be represented by

$$XY = 2^m(y_{m-1}-y_m)X + 2^{m-1}(y_{m-2}-y_{m-1})X + \ldots + 2(y_0-y_1)X \qquad 2.04$$

The "shift-and-add" algorithm can now be recognized in Equation 2.04. Each term in parentheses can assume the value of 0,1, or -1; the power of 2, associated with each term, shifts the term by the number of bits equal to the value of the exponent. The product XY is, therefore, obtained by repeatedly shifting and adding the operand X. The sign of the product is implicitly calculated in the process.

A simplified schematic for hardware implementation of Equation 2.04, to process 5-bit operands, is shown in Fig.2.01. Either the positive or the negative value of the

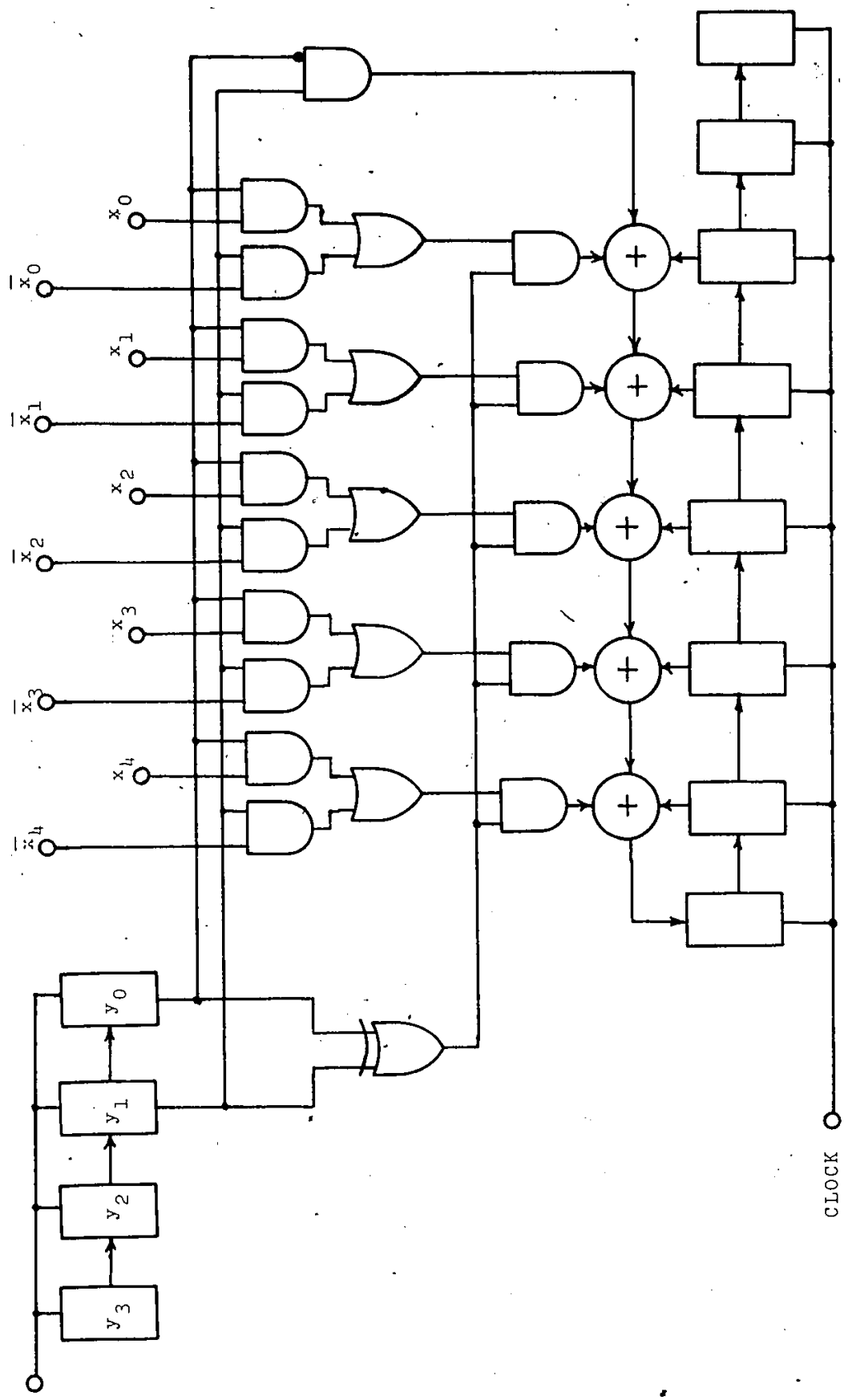


Fig. 2.01 Multiplier based on Booth's algorithm.

operand X is selected by the multiple OR gates according to the next (y_{m-1}-y_m) term. The XOR gate operating on the same term sets X=0 when y_{m-1}-y_m. Each 5-bit value of X is then added cumulatively to the contents of the output register and the sum shifted one bit.

2.03 Array Multipliers

A multiplier that is constructed to a large extent from a network of identical cells is called an array multiplier. A cellular construction of full adders can be used to implement the so called "pencil-and-paper" multiplication algorithm. Each bit of each intermediate product is generated by a separate full adder in the array. The adders are interconnected to add the intermediate products in a pattern that resembles the manual multiplication procedure. The implementation is illustrated schematically in Fig.2.02.

In contrast to the clocked multipliers in which intermediate events are initiated by a clock pulse, array multipliers allow the operand bits to propagate through the cells at the speed governed by the propagation delay time of each gate. The high speed is possible because of the high degree of hardware duplication permitting parallel processing.

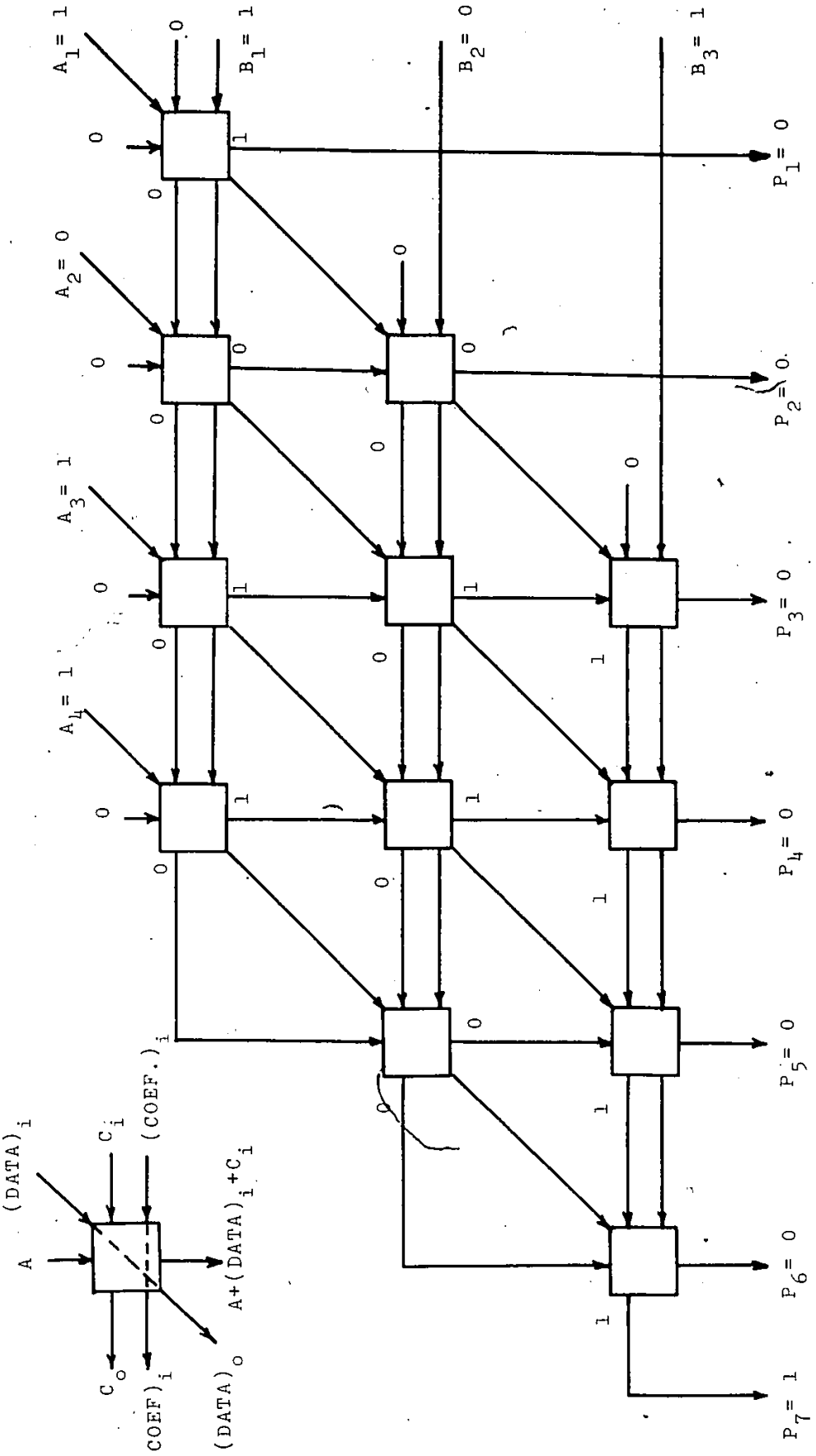


Fig.2.02 A network of cells in an array multiplier.

2.04 Peled-Liu Method

This method uses the shift-and-add principle but computes the entire second-order difference equation rather than the individual products (23). The usual form of the second-order difference equation is

$$y(nT)=a_0x(nT)+a_1x(nT-T)+a_2x(nT-2T)-b_1y(nT-T)-b_2y(nT-2T) \quad 2.05$$

If, however, the signal sampes, x and y, are available in binary form then intermediate products between each coefficient and the j-th bit of every signal sample can be formed. Gathering the j-th bit terms we obtain

$$y(nT) = \sum_{j=0}^{N-1} 2^j [a_0x_j(nT)+a_1x_j(nT-T)+a_2x_j(nT-2T) - b_1y_j(nT-T)-b_2y_j(nT-2T)] \quad 2.06$$

where N is the number of bits in the signal sample representation. Since x and y can only be 0 or 1, Equation 2.06 may be rewritten as

$$y(nT) = \sum_{j=0}^{N-1} 2^j [a_{0j} + a_{1j} + a_{2j} - b_{1j} - b_{2j}] \quad 2.07$$

where the particular a or b is either the value of the corresponding filter coefficient or zero, depending on the j-th bit. There can be 32 possible sums of the form

$$S_j = a_{0j} + a_{1j} + a_{2j} - b_{1j} - b_{2j} \quad 2.08$$

The 32 sums are stored in ROM to any desired precision and asccessed using the j-th bit from each of the five signal samples stored in the filter. The ROM address would have the form

$$A_j = x_j(nT)x_j(nT-T)x_j(nT-2T)y_j(nT-T)y_j(nT-2T) \qquad 2.09$$

The implementation block diagram for this method is given in Fig.2.03.

2.05 Multipliers in Digital Filters

There are two basic forms of structural units for implementing difference equations of the type given by Equation 2.05: the direct form, shown in Fig.2.05, and the cannonic form, shown in Fig.2.06. Sensitivity considerations often lead to cascade or parallel interconnections of low-order sections of either form to make up higher order filters. Another class of low-sensitivity structures known as wave digital filters can be obtained by using special synthesis procedures resembling the classical filter synthesis methods (1). These structures, however, do not rely heavily on conventional digital multipliers for high bit-accuracy operands and will not be considered here.

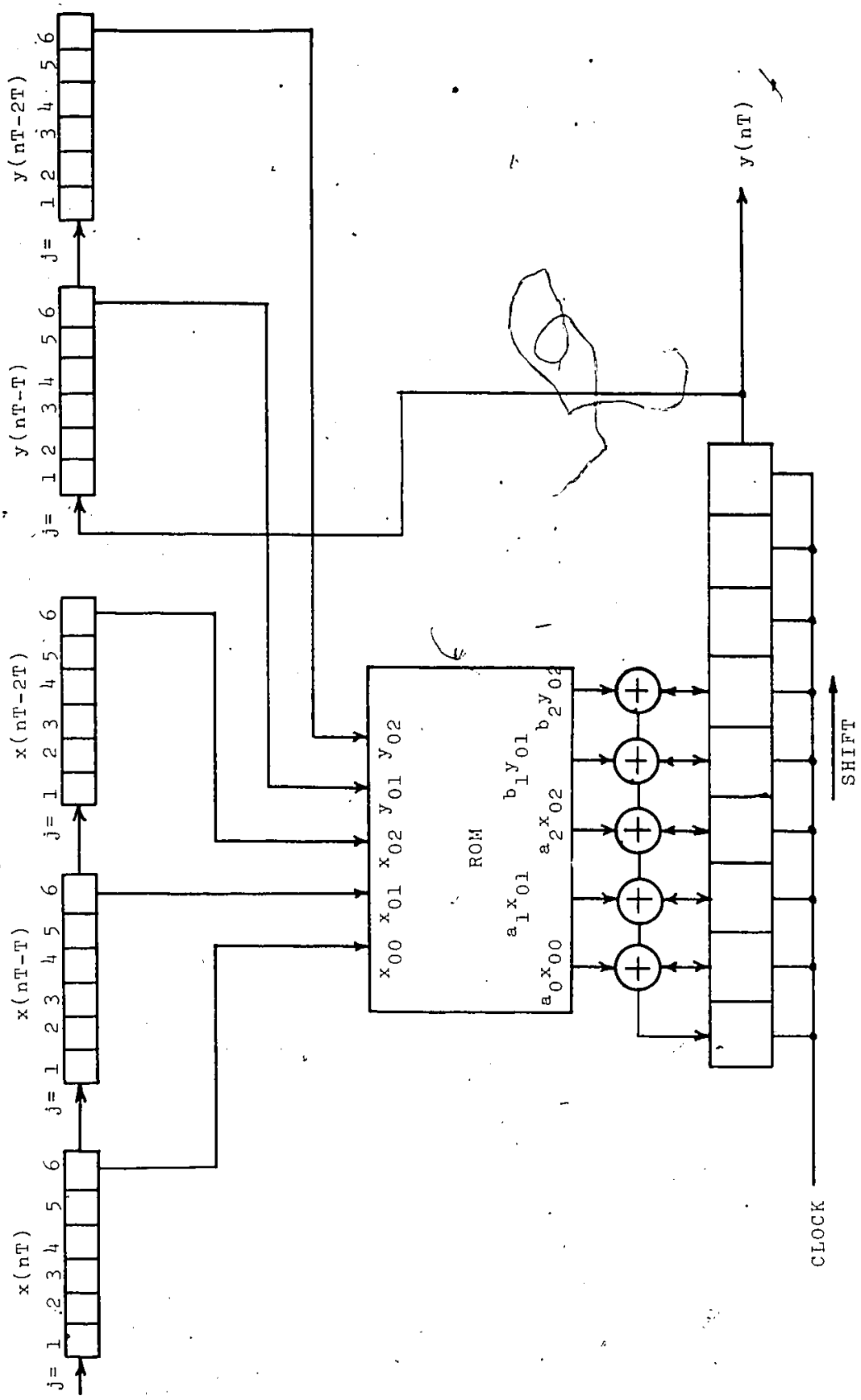


Fig. 2.03 The Peled-Liu method for a second-order system.

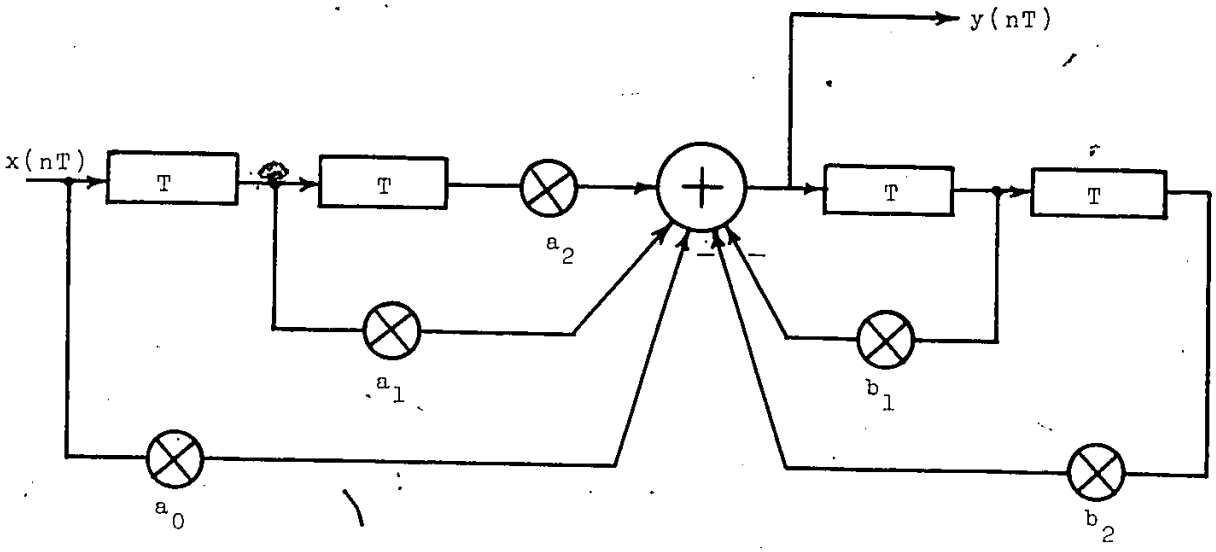


Fig. 2.05 The direct form implementation of a second-order section.

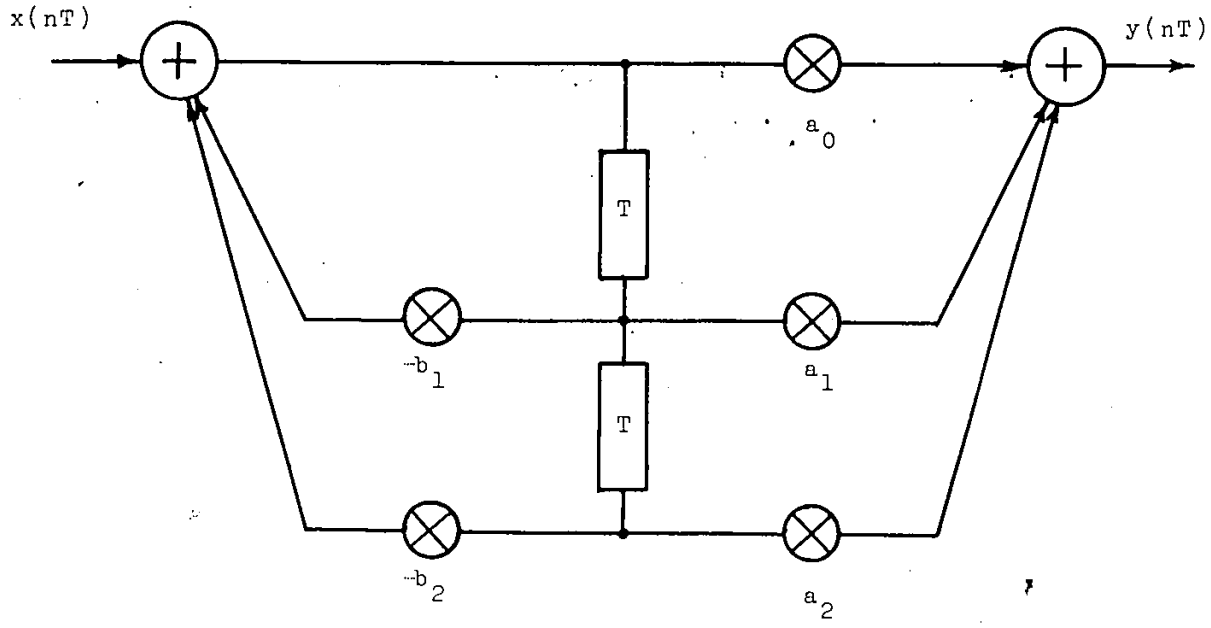


Fig.2.06 The canonic form implementation of a second-order section.

All digital filter structures, therefore, are made up of three basic circuit elements: the adder, the multiplier and the delay element. These occur in approximately the same ratios as those of the basic units shown in Fig.2.05 and Fig.2.06. The delay element is a register of length equal to at most the bit-length of the signal sample representation. The adder complexity may vary from a few gates for serial operation to nearly two hundred gates for a 16-bit parallel operation. By definition multiplication is repeated addition and, therefore, represents the largest portion of filter complexity. This consideration forces many designs to implement only one multiplier and use it sequentially with each of the filter coefficients.

2.06 Speed/Power Ratio

High speed and low power consumption in digital circuits have a reciprocal relation: a circuit can be made to operate faster by designing it to perform several tasks simultaneously in repeated hardware or equivalently, by applying more power. The ratio of processing speed to the power consumption is an important figure of merit for a digital circuit. The performance of the digital filter which is capable of computing output signal samples at some maximum rate while dissipating a known amount of power can be quantified by defining a speed/power ratio as

$$SPR = \frac{\text{time required to process one sample}}{\text{power dissipation}} \qquad 2.10$$

SPR represents the energy required to compute one sample of output to a specified accuracy. The remainder of the chapter will be devoted to assigning a numerical value of the ratio to the second-order filters based on each of the multiplier implementations presented.

TABLE 2.01 summarizes the speed and power parameters for the various circuits from which digital filters can be built. There are a dozen or so new logic families with overlapping specifications now available making it difficult to decide which family comes closest to satisfying a given set of requirements. TTL and CMOS appear to offer the desired variety of speed and power trade-offs to satisfy the majority of digital filter implementations (13), (14), (15) and (25).

Using these two logic families the implementations of the main digital filter functions were grouped into second-order direct form filters. In each case, the speed/power ratio was calculated in order to obtain a first-order comparison between the filters based on the multiplication schemes detailed in this chapter. The composition of each filter was as follows:

TABLE 2.01

Device delay and power dissipation for main digital filter sub-circuits

DEVICE	DELAY (nanoseconds) (LS)TTL	CMOS	POWER (mWatts) (LS)TTL	CMOS
GATE	10	20	2	.3
1-BIT SHIFT REGISTER	20	40	8	.6
12-BIT SHIFT REGISTER	240	480	96	7.2
1-BIT PARALLEL ACCESS REGISTER	20	40	14	2.1
12-BIT PARALLEL ACCESS REGISTER	20	40	168	25.2
1-BIT FULL ADDER	12	140	22	3.3
12-BIT PARALLEL ADDER	42	140	288	43.2
CONTROL CIRCUIT	20	40	224	33.6
Am25LS14 BOOTH MULTIPLIER	675	-	564	-
MPY-12HJ ARRAY MULTIPLIER	80	-	2000	-
TBP18S030 256-BIT ROM	25	-	400	-
IDT6116 16,384-BIT ROM	-	150	-	180

BOOTH (with one multiplier)

- five 12-bit Registers
- one Full Adder
- one 12-bit Accumulator Register
- one Booth's Multiplier
- one Control Circuit

BOOTH (with two multipliers)

- five 12-bit Registers
- three Full Adders
- one 12-bit Accumulator Register
- two Booth's Multipliers
- one Control Circuit

ARRAY

- five 12-bit Parallel Access Registers
- one 12-bit Parallel Adder
- one 12-bit Accumulator Register
- one 12-bit Array Multiplier
- one Control Circuit

PELED-LIU

- five 12-bit Registers
- one 256-bit ROM
- five Full Adders
- one 12-bit Parallel Access Register
- one Control Circuit

A simple method was used for estimating the speed/power ratio for each of the above cases. The time required to compute one output sample was taken as the time required to compute all five products of the second-order transfer function. Implementing all five multipliers to generate all products simultaneously is not advantageous for any of the filters considered. The dissipated power was taken as the total power dissipated by all sub-circuits making up the filter.

TABLE 2.02 summarizes the speed and power parameters for each of the filters. In filters using array multipliers it was assumed that only one multiplier per second-order section would be implemented in hardware. The multiplier in this case represents most of the hardware in the section, therefore, introducing a second multiplier to double the speed would also double the hardware, leaving the speed/power ratio unchanged.

In the next chapter it will be shown that by storing selected products in Read Only Memory (ROM) high speed with low power dissipation can be obtained. Using logarithmic quantization of the signal the total number of products that need to be stored in a 12-bit system can be reduced from 4096 to 256 per multiplier. A second-order section based on this principle would be possible to implement on one LSI

TABLE 2.02

Speed/Power ratio for several second-order filters

FILTER	TIME REQUIRED TO COMPUTE ONE SAMPLE (nanoseconds)	POWER DISSIPATION (Watts)	SPEED/POWER RATIO (sec/sample/Watt)
DIRECT FORM WITH ONE BOOTH MULTIPLIER	3375	1.450	0.203×10^6
DIRECT FORM WITH TWO BOOTH MULTIPLIERS	1688	2.066	0.287×10^6
DIRECT FORM WITH ONE ARRAY MULTIPLIER	400	3.520	0.710×10^6
PELED-LIU 2ND-ORDER MULTIPLIERS	1482	1.140	0.592×10^6
SPDF WITH LOGARITHMIC SIGNAL QUANTIZATION	420	0.205	11.60×10^6

chip using one of several present-day technologies. The corresponding speed/ power ratio would be improved to nearly 12×10^6 sec/sample/Watt.

CHAPTER 3

STORED-PRODUCT DIGITAL FILTERS

3.01 General

Stored-Product Digital Filter (SPDF) architecture with internal logarithmic signal quantization is proposed as a technique for achieving the highest possible processing speed per unit of dissipated power. The architecture makes use of Read Only Memory (ROM) to store a reduced number of signal-coefficient products. The ROM is organized into independently addressable sections, one for each distinct coefficient.

In some applications many of the signal-coefficient products which fall within the bit length of the processor can be omitted without significant effect on the performance quality of the processor. In such cases only those products which are essential to the desired performance need to be stored in ROM. Therefore, a product selection rule in the form of a signal quantizer is defined and implemented as

part of the processor. Specifically, for the PCM application, a product selection rule based on the segmented A-law compression characteristic is used as a practical example.

A filter implemented using these principles is shown to have the highest speed/power ratio of all leading conventional methods.

3.02 SPDF Architecture

The main hardware features of the proposed SPDF implementation are illustrated in Fig.3.01. The discussion of the proposed filter architecture and the associated quantitative analysis will largely deal with the second-order section. This will be done in part, to make the treatment more manageable as well as to provide a common basis for comparison with other current implementations. It is well known that stable high-order filter configurations are possible using a cascade of second-order sections.

In the proposed structure, the multipliers are replaced by blocks of independently addressable ROMs whose outputs are connected to a parallel adder. Quantizers, Q1 and Q2, in the form of digital coder logic are introduced to limit the number of possible sample amplitudes by quantization.

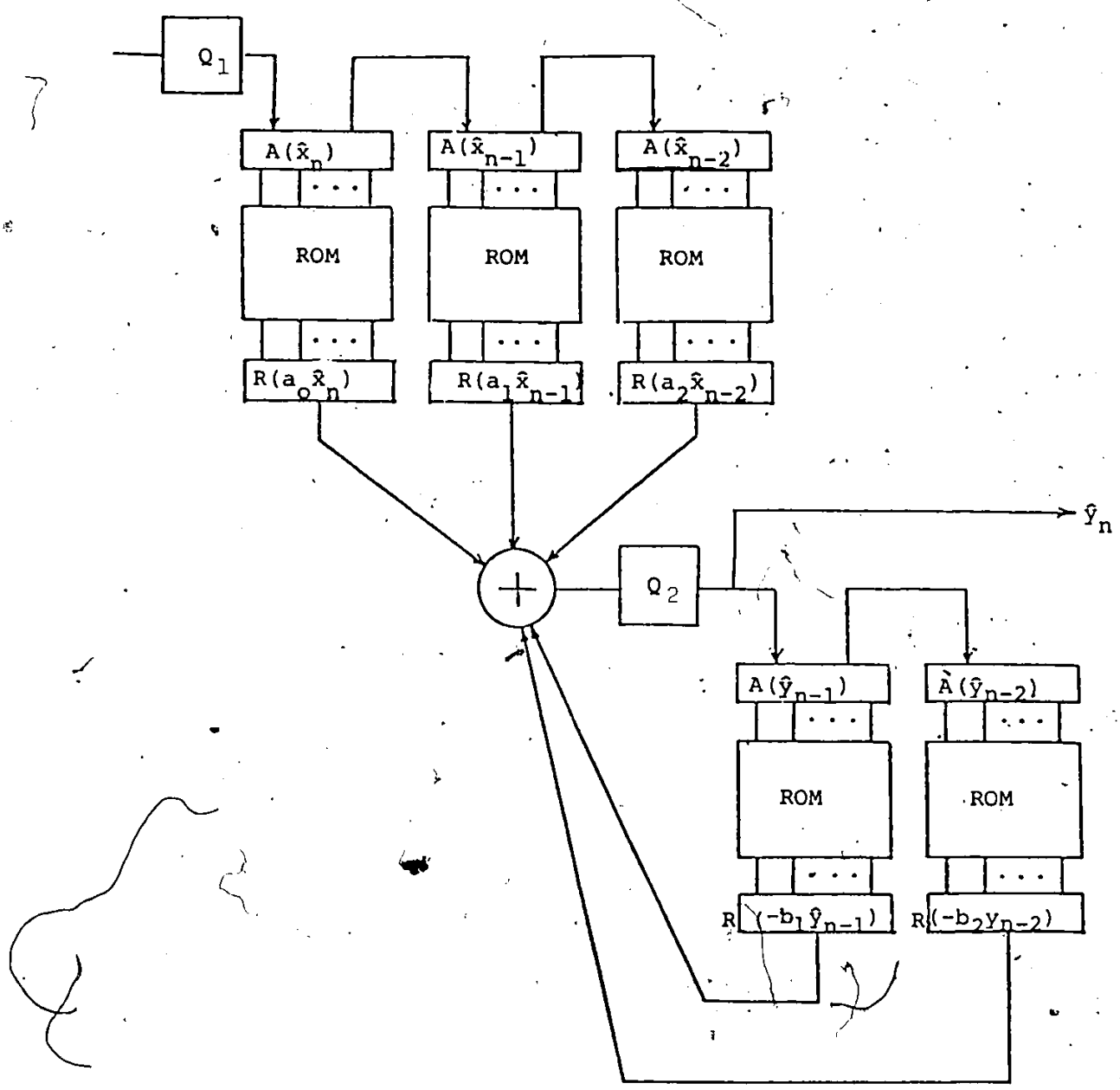


Fig.3.01 Stored-Product Digital Filter implementation with internal logarithmic signal quantization.

The products are stored as N-bit linear binary words in order to make the subsequent addition of the products possible. All bits of a product are accessed in parallel using the compressed-code form of the product's multiplicand as the address. As the products corresponding to each coefficient become available they are added sequentially by an N-bit parallel adder to produce the next output sample.

In general, a quantizer is required following each adder whose output must be used as a ROM address. Q2 in the recursive section, in Fig.3.01 is an exampole. The input quantizer, Q1, is used to bring the available system code into alignment with the nonlinear code used internally in the filter. In many systems, such as the PCM sytems, in which the signal code is identical to the internal filter code, the input quantizer, Q1, is not required.

Although the introduction of the quantizers does not restrict the number of possible filter structures, it is desirable to minimize the number of quantizers in order to keep the quantization noise as low as possible. The direct form structure accomplishes this and was selected for that reason.

The entire structure of Fig.3.01 can be implemented on a single custom LSI chip. The ROM multipliers shown as separate blocks of memory simply represent a particular ROM

organization. Likewise, the quantizer is not a separate entity; it is specified as part of the usual ROM address decode logic. The five signal registers, one parallel adder and the control logic are easily included on the same chip.

3.03 Main SPDF Characteristics

Regardless of the technological developments the basic unit of memory, the ROM cell, will require fewer components to implement than the NAND gate, the basic unit of random logic. On this basis the SPDF architecture attempts to minimize the amount of random logic in favour of arrayed logic of read-only memory. This permits the hardware implementation of all multipliers in the filter without an excessive penalty of high power dissipation.

Several advantages of the implementation follow. With all multipliers in the circuit, the products corresponding to the various coefficients can be made available at very short intervals of time. The intervals would be determined by the number of adders implemented in order to add the products and on the speed of the adders. Although initially one ROM access period is required to initiate the process, in the steady-state the ROM access times would be overlapped to accommodate the speed of addition. This can be done until the upper bound on the processing speed, equal to one ROM

access time, is reached. Therefore, SPDF can compute one output sample in time T, given by

$$T = nt_a \qquad 3.01$$

where t_a is the addition time for two N-bit products and n is the number of levels of addition required to compute the output sample. The value of n can be reduced by introducing concurrent addition operations. For the direct-form, second-order section, the minimum value of n is 3 and the maximum value is 5. The maximum processing speed per sample that can be achieved by a second-order SPDF is given by

$$T = 3t_a \qquad 3.02$$

However, the basic condition

$$t_p < nt_a \qquad 3.03$$

where t_p is the ROM access time, must also be satisfied.

The second important feature that can only be achieved with the SPDF architecture is the ability to perturb or change entirely any one of the products on individual basis. This introduces a new degree of freedom into the design of digital filters not available with any other structure.

The total ROM storage capacity required to implement the second-order section is determined by the number of products stored and the accuracy of their representation. The number of products is controlled directly by the quantization algorithm. For the remainder of this work the segmented A-law characteristic will be assumed for the quantizers Q1 and Q2. Also, a 12-bit signal dynamic range will be used to match the PCM application requirements.

Without quantization, in a 12-bit system, 4096 products must be stored for each coefficient of the transfer function. For the 12-bit product accuracy the ROM capacity works out to be 49,152 bits per coefficient. The complete second-order filter with five coefficients would require a total ROM capacity of 245,760 bits. It is not likely that a filter with this storage requirement can be implemented on one chip in the near future. By introducing A-law compression the number of products is reduced to 256 per coefficient and the ROM capacity to 3072 bits per coefficient. The entire ROM capacity of the second-order filter is reduced to 15,360 bits. This is well within the capabilities of several LSI technologies available today.

3.04 Speed/Power Ratio

The Speed of the SPDF is determined by the Equation 3.01, with the upper bound given by 3.03. Power dissipation will

be determined by the particular choice of technology or a combination of technologies. Currently several proprietary versions of advanced CMOS, using multi-level processes, have achieved densities of 1.1 square mils per memory cell. Using such a technology Integrated Device Technology designed 16,384-bit RAMs with maximum access times of 70, 90 and 120 nanoseconds and power dissipation of 180 mWatts. The total chip area used for the memory is 34,200 square mils. This technology, as well as several others offer a power consumption of about 6 microwatts per memory cell. A two-input NAND gate requires approximately four times as much area to implement compared to a memory cell. A NAND gate would, therefore, occupy an area of 4.4 square mils and dissipate about 24 microwatts of power. If five registers, a control circuit and four 12-bit parallel adders are included the equivalent gate count would be approximately 1200 gates. Allowing 20% area loss to interconnection of gates the additional chip area required to accommodate the random logic would be 6336 square mils. With 6 microwatts of power dissipation per 1.1 square mils, the additional logic would represent a power dissipation of 35 mWatts. The total chip area required for the entire second-order SPDF would be 40,500 square mils, well within the limits of manufacturing economics. The power dissipation would be about 205 mWatts (9),(13),(29).

architecture the maximum signal processing speed of Equation 3.02 can be achieved. Using the add-time of 140 nanoseconds given in TABLE 2.01, the time needed to compute one output sample becomes 420 nanoseconds. The speed/power ratio is then 11.6×10^6 sec/sample/Watt. This value is more than 16 times better than the best value listed in TABLE 2.02.

CHAPTER 4

QUANTIZED SIGNAL REPRESENTATION

4.01 General

The main purpose for introducing signal compression into the digital filter structure is to provide a functional control of the number of possible signal values and hence, the total product storage capacity required. In order to ensure compatibility with current digital communications systems and to simplify hardware implementation, the compression characteristic was chosen to correspond closely to the segmented form of the A-law logarithmic compression curve. Thus, the additive effects of the resulting quantization on the processed signal will correspond to the well established signal-to-quantization noise properties. Other algorithms which will reduce the number of stored products are also possible, however, the A-law will be used throughout to underline a practical communications application and to provide a known basis for comparison.

This chapter presents the definition of the quantization algorithm and the derivation of working algebraic expressions required for the signal processing calculations and implementation (22).

4.02 Quantizer Characteristic

The high speed operation of the SPDF is achieved, in part, by the use of fixed-point arithmetic. In the fixed-point binary form each signal sample value may be thought of as an integer represented in two's complement. In practical voice communications systems the acceptable signal dynamic range is 72dB which can be achieved by a 12-bit system. The sample value, x, processed by the quantizer appears, therefore, as a 12-bit binary integer at the output of the adder and is converted to an 8-bit nonlinear code by the quantizer (16).

In order to formulate the quantization algorithm the entire range of integers, $-2048 < x < 2048$, is partitioned into subranges to correspond to the segments and quanta of the A-law compression characteristic (22). It is sufficient to deal with the positive range only and make the necessary sign adjustment after the quantization process is complete. Thus, the first 16 of 2048 positive integers are identified with the lowest segment designated by S=0. The next 16 integers are identified with the next higher segment, s=1; however, the value of x falling within these two segments remains unaltered by the quantizer. The next 32 integers

correspond to s=2 and are subdivided into 16 groups of 2 integers; the next 64 integers correspond to s=3 and are subdivided into 16 groups of 4 integers. The doubling process continues until the last 1024 integers corresponding to s=7 are subdivided into 16 groups of 64 integers. In all there are 8 segments, s=0,1,2...,7; the 16 groups within each segment are refered to as quanta and are numbered q=0,1,2...,15. The numerical values of the break-points separating the various segments are given in TABLE 4.01.

TABLE 4.01

Partitioning of integers x according to segments

Segment Number	Range of Integers
s = 0	0 < x < 15.5
s = 1	15.5 < x < 31.5
s = 2	31.5 < x < 63.5
s = 3	63.5 < x < 127.5
s = 4	127.5 < x < 255.5
s = 5	255.5 < x < 511.5
s = 6	511.5 < x < 1023.5
s = 7	1023.5 < x < 2047.5

Signal compression results when the magnitude of a sample, expressed as an integer $x(s,q,)=(x;s=p,q=k)$ in a given quantum, is represented by the average $\hat{x}(s,q)$ of the integers

in that quantum. For s=0 and s=1 the quanta contain only one integer and so $x=\hat{x}(s,q)$. For higher values of s, $\hat{x}(s,q)$ can be obtained from the lower break-point, B(s), and the number of integers, n(s,q), in the quantum:

$$\hat{x}(s,q) = B(s) + qn(s,q) + \frac{n(s,q)}{2} \qquad 4.01$$

Figure 4.01 shows the segment and quantum parameters for s=3. The validity of Equation 4.01 can be established by inspection of the interrelationship of these parameters. It can be seen that the lower break-point B(s) is given by

$$B(s) = 2^{s+3} - 0.5 \qquad 4.02$$

and the number of integers in a quantum by

$$n(s,q) = 2^{s-1} \qquad 4.03$$

Substituting Equationa 4.02 and 4.03 into Equation 4.01 we obtain the expression for the quantized value of x,

$$\hat{x}(s,q) = 2^{s-1}(q + 16.5) - 0.5 \qquad 4.04$$

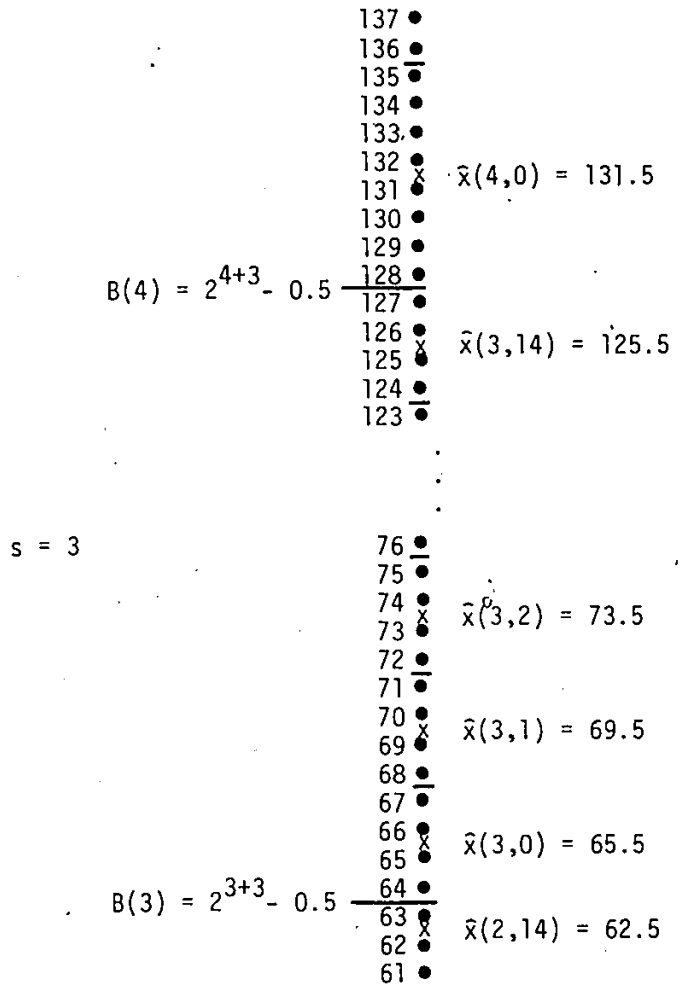


Fig. 4.01 Segment and quantum parameters for s=3.

$\hat{x}(s,q)$ is referred to as quantization level for the range of integers defined by the particular values of s and q. The entire set of positive quantization levels can be generated using the various combinations of s and q in Equation 4.04. The numerical values for the levels are tabulated in TABLE 4.02. The same values with a negative sign complete the full dynamic range of the quantizer.

TABLE 4.02

Quantization levels for integers between 0 and 2048

q	s=0	s=1	s=2	s=3	s=4	s=5	s=6	s=7
0	0	16	32.5	65.5	131.3	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

4.03 Quantization Algorithm

In order to implement the operation of the quantizer, it is necessary to specify its algorithm such that $\hat{x}(s,q)$ is uniquely determined from an arbitrary value of x. This can be done by first determining s and q from x and then using Equation 4.04 to find $\hat{x}(s,q)$. For $x<15.5$ we have simply $s=0$ and $q=x$; similarly, for $15.5<x<31.5$, $s=1$ and $q=x$. In fact, for $s=0$ and $s=1$ $\hat{x}(s,q)=x$. For $x>31.5$ we refer to Figure 4.01. Here it can be seen that any x falling within a given segment s exceeds the value of the segment's lower break-point B(s),

$$B(s) < x \qquad 4.05$$

Substituting Equation 4.02 into 4.05 and rearranging, we obtain

$$2^s < \frac{x + 0.5}{8} \qquad 4.06$$

or

$$s < \log_2[\frac{x + 0.5}{8}] \qquad 4.07$$

The correct value of s corresponds to the highest integer value obtained in the expression 4.07. That is,

$$s = I[\log_2(\frac{x + 0.5}{8})] \qquad 4.08$$

where I is the operator which retains the ineger value of the argument in brackets.

This definition of s provides the proper bounds for the integers falling within the segment's range. Consider the integer $x=B(s_o)-0.5$, in the segment s_1, and immediately outside the lower bound of the segment s_o. Using 4.08 we obtain

$$s_1 = I[\log_2(2^{s_0} - \tfrac{1}{16})] \qquad 4.09$$

but

$$\log_2(2^{s_0} - \tfrac{1}{16}) < s_0 \qquad 4.10$$

and so $s_1 < s_0$. Similarly, consider the integer $x=2B(s_0)+1$, in segment s_2, and immediately outside the upper bound of the segment s_0. Using 4.08 we then obtain

$$s_2 = I[\log_2(2^{s_0+1} + \tfrac{1}{16})] \qquad 4.11$$

but

$$\log_2(2^{s_0+1} + \tfrac{1}{16}) > s_0 + 1 \qquad 4.12$$

therefore, $s_2 > s_0$

Having found s in terms of x it remains to find q corresponding to that value of x.

To determine q we observe in Figure 4.01 that for an arbitrary value of x

$$q < \frac{x - B(s)}{n(s,q)} \qquad 4.13$$

Using Equations 4.02 and 4.03 and rearranging, we get

$$q < \frac{2x + 1}{2^s} - 16 \qquad 4.14$$

As for s, we chose the value of q which corresponds to the highest integer allowed by the expression 4.14. That is,

$$q = I[\frac{2x + 1}{2^s} - 16] \qquad 4.15$$

where I is the same operator as that defined in Equation 4.08.

The same method as that used for the segments can be used to show that Equation 4.15 satisfies the requirement that the integers falling within the respective quanta form disjoint sets.

The expressions derived in this chapter define the quantization algorithm for integers. The quantities $\hat{x}(s,q)$ are essential in generating the signal-coefficient products. SPDF implementated in hardware does not compute the values

of $\hat{x}(s,q)$ directly; instead the products of the form $a\hat{x}(s,q)$, where a is a coefficient of the transfer function, are stored in ROM and are accessed during the operation of the filter.

In order to locate the correct signal-coefficient product $a\hat{x}(s,q)$ in ROM the hardware implemented quantizer synthesizes the corresponding 8-bit memory address code A(s,q). A(s,q) consists of a sign bit (the most significant bit), the line segment number s (3 next most significant adjacent bits) and the quantum number q (the last 4 adjacent bits). The address code establishes a one-to-one correspondence with the products having the same s and q parameter values.

Example

Given that the output of the adder is equal to 492, what is the quantized output of the filter, expressed in the form of the 8-bit address code?

Solution:

$$x = 492$$

$$s = I[\log_2(\frac{492+0.5}{8})]$$

$$s = I[5.943965431]$$
$$= 5$$
$$q = I\left[\frac{492\times 2 + 1}{2^5} - 16\right]$$
$$= I[14.78125]$$
$$= 14$$

$$\hat{x}(5,14) = 2^{5-1}(14 + 16.5) - 0.5$$
$$= 487.5$$

To find A(s,q) we express s and q in binary as follows,

$$s = 101$$
$$q = 1110$$
$$\text{sign} = 0$$
$$A(s,q) = 01011110$$

CHAPTER 5

LIMIT CYCLES IN SPDF

5.01 General

The natural modes of the filter transfer function are realized by the recursive architectural structure utilizing one or more feedback loops. Transfer function coefficients for such structures are calculated under the constraints of the usual stability criteria. For example, when transformed to the z-plane, the natural modes must fall inside the unit circle for the exact values of the coefficients. In the second-order case this condition restricts the recursive coefficient pairs, (b_1,b_2), to a triangular region in the $b_1 x b_2$ plane. However, even those coefficient pairs which initially fall inside the stability triangle can lead to constant amplitude oscillations and DC outputs due to the effects of finite arithmetic of the digital implementation.

L.B. Jackson (10) and others have partitioned the area inside the stability triangle into regions according to order,

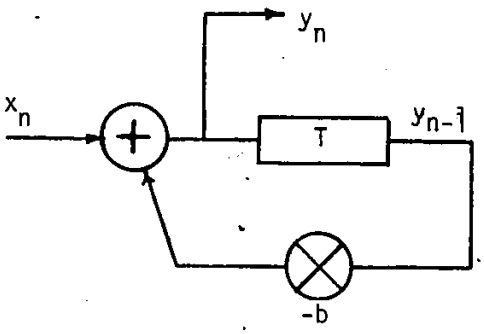
amplitude and frequency of the limit cycles resulting from the usual arithmetic rounding of the coefficients. Introduction of the logarithmic quantization of the recursive signal indirectly contributes to the corresponding quantization effects of the coefficients. This leads to a special limit cycle behaviour.

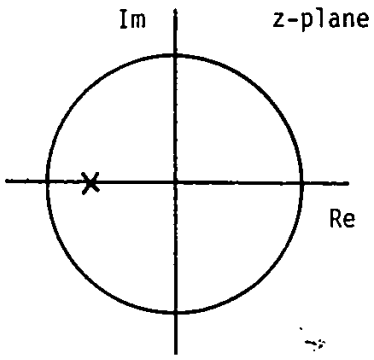
A detailed examination of the nature of the limit cycles resulting from the logarithmic quantization of the recursive signal is presented in this chapter. It is shown that such limit cycles are only possible for coefficients that fall inside narrow strips along the boundaries of the stability triangle in the coefficient plane. The limit cycle behavoiur for coefficients outside these strips is identiacal to that originally described by Jackson. Furthermore, it is shown that high quality digital filters can be designed with coefficients falling outside these strips.

5.02 Basic Stability Considerations

A digital filter is said to be stable when its natural modes are situated inside the unit circle, $|z| = 1$, of the z-plane. The basic parameters of a first-order system are illustrated in Fig. 5.01. Under zero input condition the output of the filter is calculated according to the equation

$$y_n = -by_{n-1} \qquad 5.01$$





$$y_n = x_n - by_{n-1}$$

$$H(z) = \frac{1}{z + b}$$

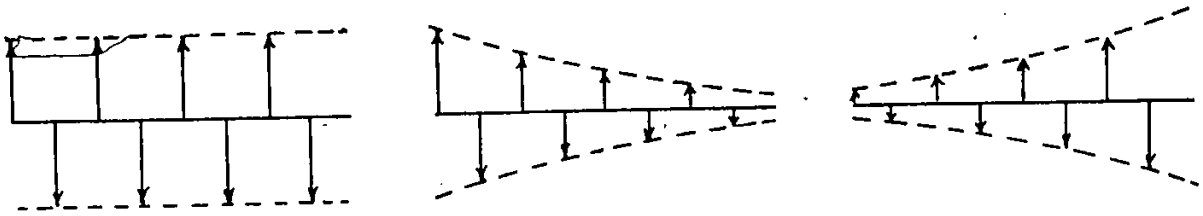


Fig. 5.01 First-order system parameters and response.

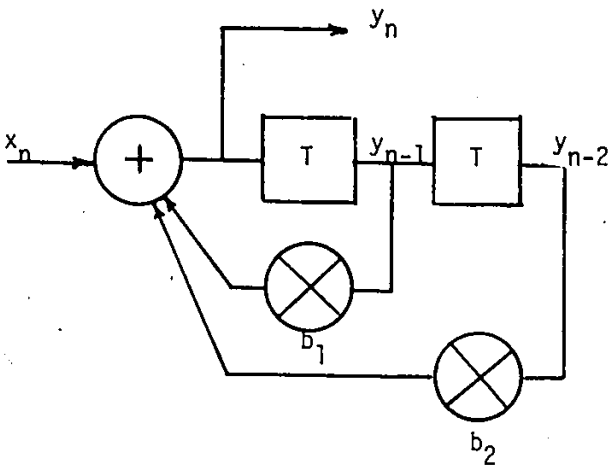
For a stable system, b is a fraction ($-1<b<1$), so that each time the signal circulates through the feedback loop its magnitude is reduced by the multiplying factor, b, and eventually becomes arbitrarily small.

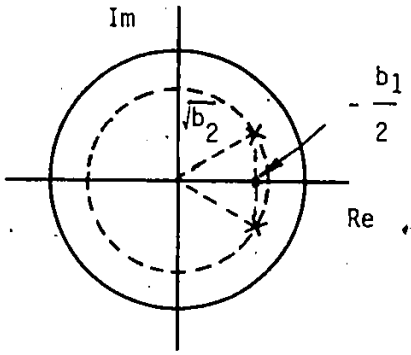
In a system using finite arithmetic, however, if b falls within a certain range of values, the effect of rounding the product may counteract this reduction. In the critical case in which the possible reduction is less than or equal to the incremental change due to rounding, the output signal will remain unchanged. The filter is then critically stable with b effectively equal to 1; the output is then said to have entered a limit cycle, and the corresponding coefficient values are said to fall inside a deadband.

The basic parameters of a second-order system are illustrated in Fig. 5.02. the coefficient pairs, (b1,b2), corres- ponding to every stable pole position can be conveniently represented in the b1xb2 parameter space as the area within the triangle shown in Fig. 5.03. The boundaries defining the triangle follow from the stability condition

$$|z| < 1 \qquad 5.02$$

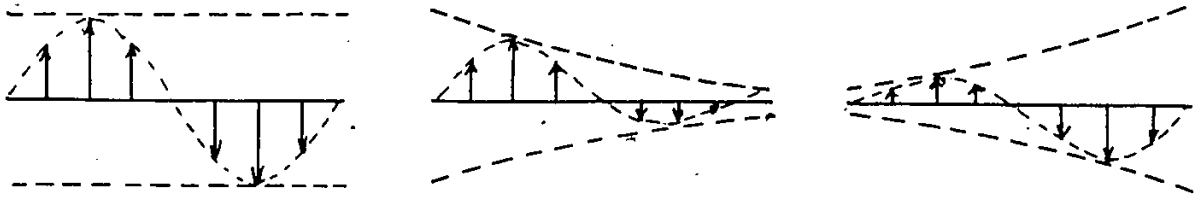
evident from inspection of the unit circle. In the z-plane





$$y_n = x_n - b_1 y_{n-1} - b_2 y_{n-2}$$

$$H(z) = \frac{1}{z^2 + b_1 z + b_2}$$



$b_2 = 1$ $\quad$ $b_2 < 1$ $\quad$ $b_2 > 1$

Fig. 5.02 Second-order system parameters and response.

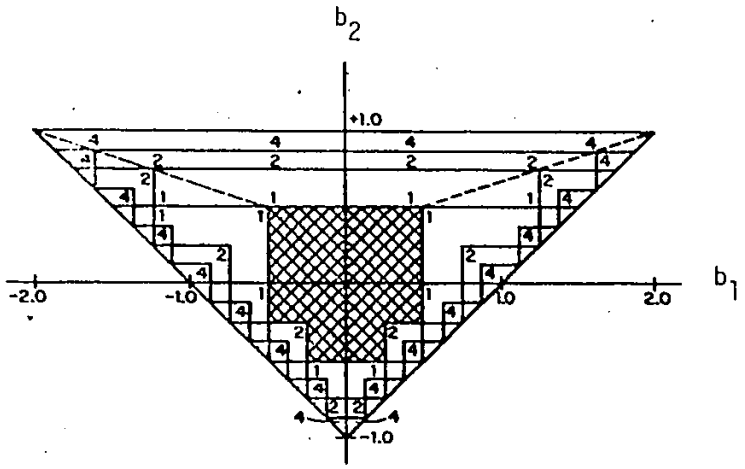


Fig. 5.03 Triangle of coefficient pairs corresponding to stable pole positions $D(z)=z^2 + b_1 z+b_2$.

the transfer function poles are given by the complex values of z

$$z = \tfrac{1}{2}b_1 \pm \tfrac{1}{2}\sqrt{b_1^2 - 4b_2} \qquad 5.03$$

For the real-valued poles, the stability condition 5.02 assumes the extreme values, z=+1. When this bound is applied to 5.03 the right and left sides of the stability triangle are obtained as

$$b_2 \pm b_1 = 1 \qquad 5.04$$

For complex pole pairs we see from Fig. 5.02 that

$$|z| = \sqrt{b_2} \qquad 5.05$$

Applying the bound of 5.02 the third side of the triangle is determined as

$$b_2 = 1 \qquad 5.06$$

The area within the triangle encloses all those coefficient pairs which for exact signal representations yield stable filters. Arithmetic rounding of the output signal, however, introduces coefficient perturbations which can lead to poles on or outside the unit circle. This results in limit cycles whose amplitudes are determined by the particular

coefficient pairs. L.B. Jackson and others have partitioned the area inside the stability triangle into regions according to limit cycle amplitudes. Fig. 5.03 illustrates such partitioning for the normal arithmetic rounding to the nearest unity. Logarithmic quantization of the output signal introduces additional regions inside the triangle. The derivation of these regions will be the main subject of the next two sections (22).

5.03 First-Order Limit Cycles with Logarithmic Signal Quantization

In SPDF implementation the products are stored as integers computed according to

$$R(b_i y_{n-i}) = \mathrm{sgn}(b_i y_{n-i}) I[|b_i y_{n-i}| + 0.5] \qquad 5.07$$

where I, as before, performs decimal truncation on the quantity in brackets. The rounding process, R, applied to the individual products in a first-order system produces the same effect as the rounding of the output signal discussed in the previous section. Therefore, when the quantization algorithm of Chapter 4 is introduced as shown in Fig. 5.04, the limit cycle behaviour of the system remains unchanged for s=0 and s=1. The logarithmic quantizer only contributes to the generation of new limit cycles when the initial signal amplitude falls within the range of the upper segments (s>1).

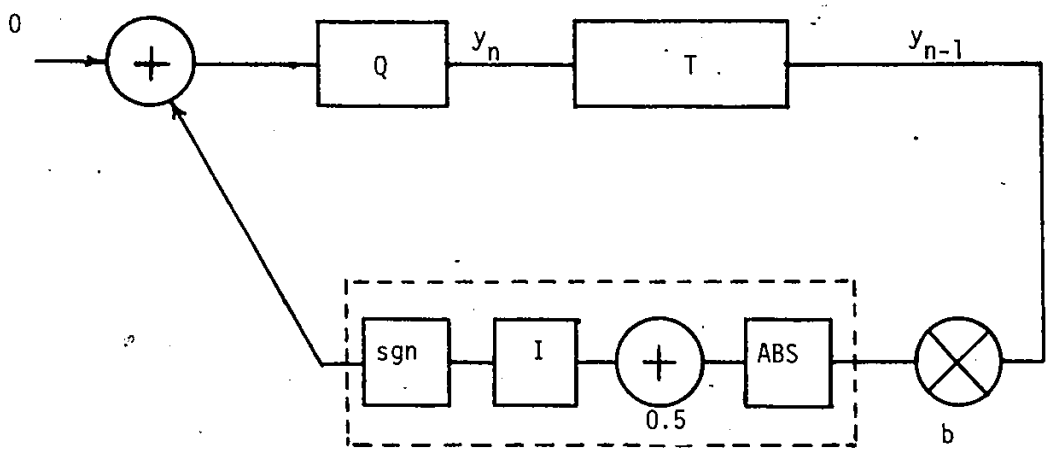


Fig. 5.04 A first-order recursive section with nonlinearly quantized output and product rounding.

the first-order arithmetic process of Fig. 5.04 can be expressed as

$$\hat{y}_n(s_1,q_1) = -Q\{R[b\hat{y}_{n-i}(s_2,q_2)]\} \qquad 5.08$$

where Q represents the quantization operation. Only real poles can be realized with this equation and, therefore, marginal stability corresponds to the z-plane pole positions of $z=\pm 1$. In order to produce a limit cycle, the quantization operation must effectively force $|b| = 1$. Therefore, for the stable values of b, $-1<b<1$, the limit cycle condition is

$$|b\hat{y}_{n-1}(s,q)| < |\hat{y}_{n-1}(s,q)| \qquad 5.09$$

As the signal, $\hat{y}_{n-1}(s,q)$, passes through the multiplier it is reduced by the multiplying factor, b. If this reduction is smaller than one half of the corresponding quantum, as in

$$|\hat{y}_{n-1}(s,q)| - |b\hat{y}_{n-1}(s,q)| < 2^{s-2} \qquad 5.10$$

or

$$1 - |b| < \frac{2^{s-2}}{|\hat{y}_{n-1}(s,q)|} \qquad 5.11$$

then no reduction of the signal will take place and the filter will enter a limit cycle of amplitude $|\hat{y}_{n-1}(s,q)|$. Otherwise, the signal will reduce to a value of the next

lower quantum. To ensure this reduction the recursive coefficient, b, must be calculated according to

$$|b| < 1 - \frac{2^{s-2}}{2^{s-1}(q + 16.5) + 0.5} \qquad 5.12$$

This equation follows from 5.11 and 4.04.

Once the signal amplitude is reduced to within the range of the two lower linear segments (s=0 and s=1), the limit cycle behaviour will be as though no quantizer were present. The lower bound for the value of b, below which the logarithmic quantizer has no contribution to the limit cycles, can be obtained using s=2 and q=0 in Equation 5.12. The result is

$$|b| < 1 - \frac{1}{32.5} \qquad 5.13$$

Example

Suppose that in a first-order system the signal in register T, of Fig. 5.04, is $\hat{y}(4,4)=163.5$ at the instant when the input signal is turned off. What is the largest value of b for which a quantum reduction in the signal, to $\hat{y}(4,3)=155.5$ will take place?

Solution

According to 5.12, the required value of b is the largest value satisfying

$$|b| < 1 - \frac{2^{s-2}}{2^{s-1}(q + 16.5) + 0.5}$$

For s=4 and q=4

$$b < 1 - \frac{4}{163.5}$$

$$b < 0.9755352$$

Therefore, assuming 7-figure accuracy, the largest value of b for which the quantum transition will take place is 0.9755351.

As a check we perform the following operations

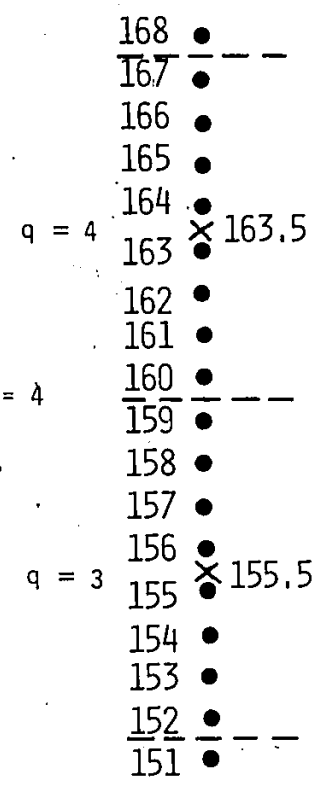
$$R[b\hat{y}_n(4,4)] = R[0.9755351 \times 163.5]$$

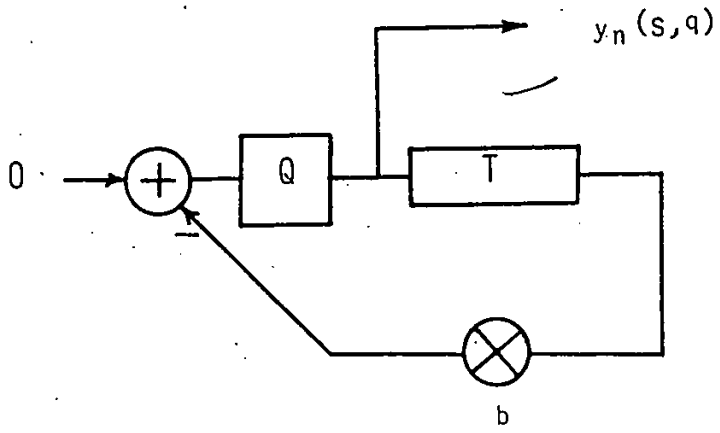
$$= R[159.4999...]$$

$$= 159$$

$$Q[159] = 155.5$$

$$= \hat{y}_n(4,3)$$





For b = 0.9755352

$$R[b\hat{y}_n(4,4)] = R[159.50001]$$

$$= 160.$$

$$Q[160] = 163.5$$

$$= \hat{y}_n(4,4)$$

TABLE 5.01 lists all positive values of the quantized signal according to the various segment and quantum numbers. In each case the lowest value of b (designated by B2), for which no reduction in signal amplitude will take place, is given. The ranges of b falling between any two tabulated values are referred to as deadbands.

If the initial signal is $\hat{y}$=2015.5 and b=0.969683, the output will reduce until it reaches the value $\hat{y}$=1055.5. However, if b=0.969682, it will reduce to $\hat{y}$=527.5. Ultimately, when b=0.969231, the output will reduce all the way to $\hat{y}$=32.5. For smaller values of b the quantizer has no contribution to the limit cycles.

5.04 Second-Order Limit Cycles with Logarithmic Signal Quantization - Real Poles

The second-order recursive SPDF section is shown in Fig. 5.05; its arithmetic process is completely specified by the expression

$$\hat{y}_n(s,q)=-Q\{R[b_1\hat{y}_{n-1}(s_1,q_1)]+R[b_2\hat{y}_{n-2}(s_2,q_2)]\} \qquad 5.14$$

where Q, as in the previous section, is the quantization operation described in Chapter 4. With a proper choice of

TABLE 5.01

Deadbands and limit cycle amplitudes.

S=7	Q=15	Y=2015.5	B2=.984123	S=4	Q=15	Y=251.5	B2=.984095
S=7	Q=14	Y=1951.5	B2=.983602	S=4	Q=14	Y=243.5	B2=.983573
S=7	Q=13	Y=1887.5	B2=.983046	S=4	Q=13	Y=235.5	B2=.983015
S=7	Q=12	Y=1823.5	B2=.982451	S=4	Q=12	Y=227.5	B2=.982418
S=7	Q=11	Y=1759.5	B2=.981813	S=4	Q=11	Y=219.5	B2=.981777
S=7	Q=10	Y=1695.5	B2=.981127	S=4	Q=10	Y=211.5	B2=.981087
S=7	Q=9	Y=1631.5	B2=.980386	S=4	Q=9	Y=203.5	B2=.980344
S=7	Q=8	Y=1567.5	B2=.979585	S=4	Q=8	Y=195.5	B2=.97954
S=7	Q=7	Y=1503.5	B2=.978716	S=4	Q=7	Y=187.5	B2=.978667
S=7	Q=6	Y=1439.5	B2=.97777	S=4	Q=6	Y=179.5	B2=.977716
S=7	Q=5	Y=1375.5	B2=.976736	S=4	Q=5	Y=171.5	B2=.976676
S=7	Q=4	Y=1311.5	B2=.9756	S=4	Q=4	Y=163.5	B2=.975535
S=7	Q=3	Y=1247.5	B2=.974349	S=4	Q=3	Y=155.5	B2=.974277
S=7	Q=2	Y=1183.5	B2=.972962	S=4	Q=2	Y=147.5	B2=.972881
S=7	Q=1	Y=1119.5	B2=.971416	S=4	Q=1	Y=139.5	B2=.971326
S=7	Q=0	Y=1055.5	B2=.969683	S=4	Q=0	Y=131.5	B2=.969582
S=6	Q=15	Y=1007.5	B2=.984119	S=3	Q=15	Y=125.5	B2=.984064
S=6	Q=14	Y=975.5	B2=.983598	S=3	Q=14	Y=121.5	B2=.983539
S=6	Q=13	Y=943.5	B2=.983042	S=3	Q=13	Y=117.5	B2=.982979
S=6	Q=12	Y=911.5	B2=.982447	S=3	Q=12	Y=113.5	B2=.982379
S=6	Q=11	Y=879.5	B2=.981808	S=3	Q=11	Y=109.5	B2=.981735
S=6	Q=10	Y=847.5	B2=.981121	S=3	Q=10	Y=105.5	B2=.981043
S=6	Q=9	Y=815.5	B2=.98038	S=3	Q=9	Y=101.5	B2=.980296
S=6	Q=8	Y=783.5	B2=.979579	S=3	Q=8	Y=97.5	B2=.979487
S=6	Q=7	Y=751.5	B2=.978709	S=3	Q=7	Y=93.5	B2=.97861
S=6	Q=6	Y=719.5	B2=.977762	S=3	Q=6	Y=89.5	B2=.977654
S=6	Q=5	Y=687.5	B2=.976727	S=3	Q=5	Y=85.5	B2=.976608
S=6	Q=4	Y=655.5	B2=.975591	S=3	Q=4	Y=81.5	B2=.97546
S=6	Q=3	Y=623.5	B2=.974338	S=3	Q=3	Y=77.5	B2=.974194
S=6	Q=2	Y=591.5	B2=.97295	S=3	Q=2	Y=73.5	B2=.972789
S=6	Q=1	Y=559.5	B2=.971403	S=3	Q=1	Y=69.5	B2=.971223
S=6	Q=0	Y=527.5	B2=.969668	S=3	Q=0	Y=65.5	B2=.969466
S=5	Q=15	Y=503.5	B2=.984111	S=2	Q=15	Y=62.5	B2=.984
S=5	Q=14	Y=487.5	B2=.98359	S=2	Q=14	Y=60.5	B2=.983471
S=5	Q=13	Y=471.5	B2=.983033	S=2	Q=13	Y=58.5	B2=.982906
S=5	Q=12	Y=455.5	B2=.982437	S=2	Q=12	Y=56.5	B2=.982301
S=5	Q=11	Y=439.5	B2=.981797	S=2	Q=11	Y=54.5	B2=.981651
S=5	Q=10	Y=423.5	B2=.98111	S=2	Q=10	Y=52.5	B2=.980952
S=5	Q=9	Y=407.5	B2=.980368	S=2	Q=9	Y=50.5	B2=.980198
S=5	Q=8	Y=391.5	B2=.979566	S=2	Q=8	Y=48.5	B2=.979381
S=5	Q=7	Y=375.5	B2=.978695	S=2	Q=7	Y=46.5	B2=.978495
S=5	Q=6	Y=359.5	B2=.977747	S=2	Q=6	Y=44.5	B2=.977528
S=5	Q=5	Y=343.5	B2=.97671	S=2	Q=5	Y=42.5	B2=.976471
S=5	Q=4	Y=327.5	B2=.975573	S=2	Q=4	Y=40.5	B2=.975309
S=5	Q=3	Y=311.5	B2=.974318	S=2	Q=3	Y=38.5	B2=.974026
S=5	Q=2	Y=295.5	B2=.972927	S=2	Q=2	Y=36.5	B2=.972603
S=5	Q=1	Y=279.5	B2=.971377	S=2	Q=1	Y=34.5	B2=.971014
S=5	Q=0	Y=263.5	B2=.969639	S=2	Q=0	Y=32.5	B2=.969231

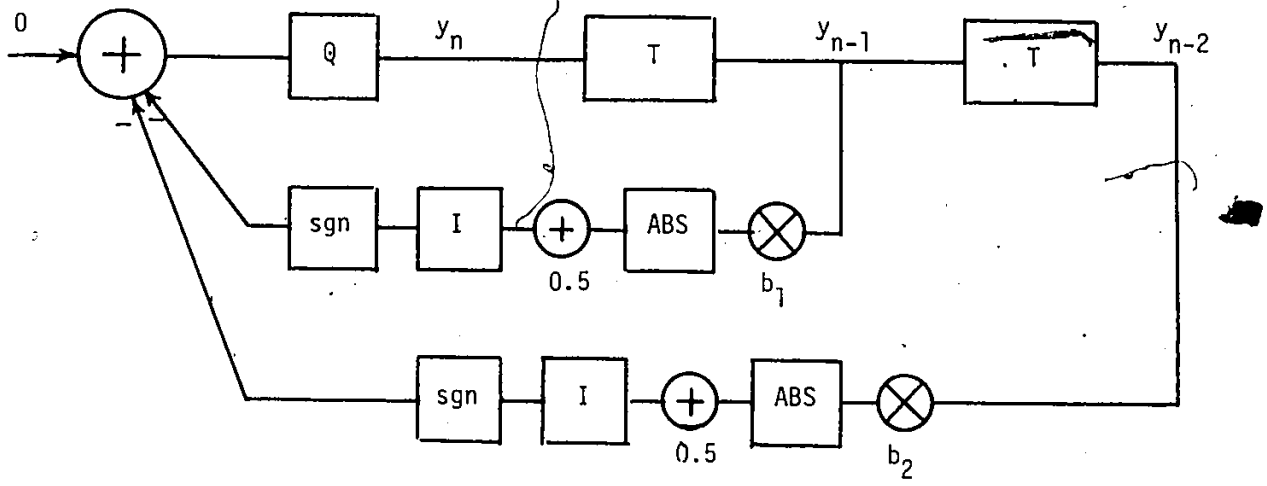


Fig. 5.05 A second-order recursive filter with nonlinear quantization of output.

coefficients b_1 and b_2, it is possible to realize both real and complex-conjugate poles. For some choices of (b_1,b_2) coefficient pairs, however, the quantization operation, Q, forces the poles to change positions and even move out of the unit circle. Complex-conjugate poles which fall on the unit circle result in sinusoidal limit cycles with constant peak amplitudes. Real poles on the unit circle produce DC limit cycles.

Refering to Fig. 5.01 it can be seen that the real pole locations for the marginally stable case lie at $z=\pm 1$. The resulting DC limite cycles represent the signal condition

$$|\hat{y}_n| = |\hat{y}_{n-1}| = |\hat{y}_{n-2}| \qquad 5.15$$

Applying this to Equation 5.14 we obtain

$$|\hat{y}_n| = -Q\{\pm R[b_1|\hat{y}_n|] + R[b_2|\hat{y}_n|]\} \qquad 5.16$$

where the $\pm$ is asociated with b_1 in order to account for both positive and negative pole locations. For the signal to undergo a downward quantum transition, it is necessary to satisfy

$$|\hat{y}_n| - 2^{s-2} < \pm R[b_1|\hat{y}_n|] - R[b_2|\hat{y}_n|] \qquad 5.17$$

where 2 is a half-quantum step. 5.17 can be rearranged to form

$$\frac{R[b_2|\hat{y}_n|]}{|\hat{y}_n|} < \pm\frac{R[b_1|\hat{y}_n|]}{|\hat{y}_n|} - 1 + \frac{2^{s-2}}{|\hat{y}_n|} \qquad 5.18$$

This equation can be simplified if we assume

$$R[b|\hat{y}_n|] \simeq b|\hat{y}_n| \qquad 5.19$$

where b represents either b_1 of b_2. The largest possible error this assumption would introduce into the two terms in 5.18 is .5/32.5; it is shown in Chapter 6 (TABLE 6.02) that on the average this error would be an order of magnitude smaller than that. In the worst case the assumption could lead to a sample amplitude difference of at most one quantum step. A compensation for the error can easily be made since we seek a bound rather than an exact expression. Therefore, substituting 5.19 into 5.18 we get a bounding equation

$$b_2 = \pm b_1 - 1 + \frac{2^{s-2}}{|\hat{y}_n(s,q)|} \qquad 5.20$$

For a specified pair of parameters, s and q, Equation 5.20 represents two straight lines with slopes of +1 and -1, and a common negative b_2-axis intercept given by

$$b_2 = -1 + \frac{2^{s-2}}{|\hat{y}_n(s,q)|} \qquad 5.21$$

These lines are parallel to two of the sides of the stability triangle and fall inside the area of stable coefficients. In fact, Equation 5.20 represents a family of straight lines with running parameters s and q. The two innermost lines corresponding to s=2 and q=0 are given by

$$b_2 = \pm b_1 - 1 + \frac{1}{32.5} \qquad 5.22$$

and form a bound on the coefficient pairs for which limit cycles due to logarithmic quantizer are possible. These are shown in Fig.5.08 as narrow strips along the sloped sides of the triangle and enclose a very small portion of useful coefficients.

A similar bound can be derived for the complex-conjugate poles.

5.05 Second-order Limite Cycles with Logarithmic Signal Quantization - Complex-conjugate Poles

When the input signal to the filter of Fig.5.05 is suddenly removed, the complex-conjugate poles located on the unit circle may sustain a sinusoidally varying output signal of constant amplitude and frequency. The amplitude of the signal would be determined by the amplitude of the preceding signal and the filter coefficients. The frequency would

be determined by the frequency of the poles. In order to derive the desired bounds on the amplitudes of such output signals several related parameters will first have to be defined.

Fig.5.06 shows the relationship of the essential parameters to the unit circle and the filter coefficients b_1 and b_2. The points on the unit circle correspond to the points on the jw-axis of the s-plane. If F_s is used to represent the signal sampling frequency, in Hz, then the singal frequency F, also in Hz, can be expressed as

$$F = \frac{F_s \omega T}{2\pi} \qquad 5.23$$

where T is the time interval between samples, in seconds. The number of samples per cycle, N, defined by

$$N = \frac{F_s}{F} \qquad 5.24$$

can then be expressed as

$$N = \frac{2\pi}{\omega T} \qquad 5.25$$

However,

$$\omega T = \tan^{-1} \frac{Y}{X} \qquad 5.26$$

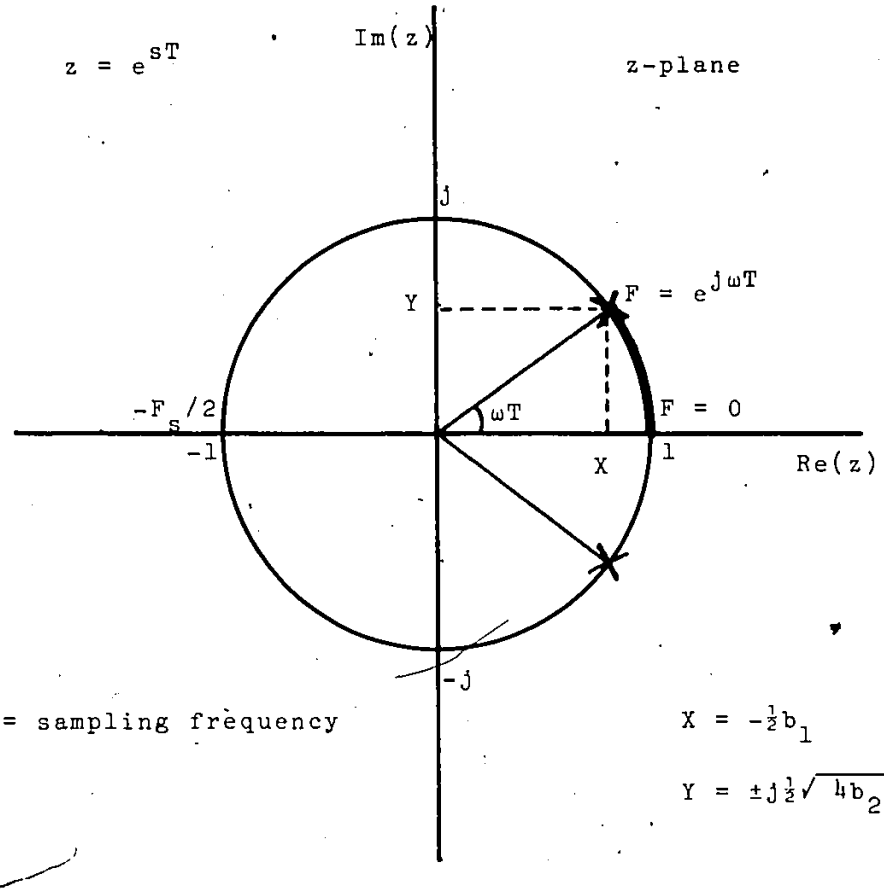


Fig. 5.06 Complex-conjugate poles and filter coefficients in the z-plane.

which leads to

$$N = \frac{2\pi}{\tan^{-1}\left(\frac{\sqrt{4b_2 - b_1^2}}{b_1}\right)} \qquad 5.27$$

As before, effective coefficient rounding may force the poles to migrate to the unit circle. If this happens the resulting limit cycle will have the obvious property that every N-th sample of the output will be equal. Conversely, if not limit cycle occurs and the output decays to zero then for every successive output cycle, k, we will have

$$|\hat{y}(nT)| < |\hat{y}(nT+kT)| \qquad 5.28$$

where $k=1,2,3,\ldots$. This property is illustrated in TABLE 5.02.

In the polar co-ordinate form the position of a complex-conjugate pole is given by

$$z = |\sqrt{b_2}|e^{j\omega T} \qquad 5.29$$

By rounding b_2, a pole initially inside the unit circle can effectively be placed on the unit circle. In TABLE 5.02 the coefficient b_1 was fixed at the value - 1.359570771 while the coefficient b_2 assumed two values, 0.974 and 0.980. For $b_2=0.974$ the output samples spaced by N=8 were progressively smaller in amplitude. For $b_2=0.980$, the samples spaced by

Table 5.02

The effect of rounding b_2 on the output signal.

b_1=-1.39570771

Output Signal for b_2=0.974	Output Signal for b_2=0.980
211.5	243.5
503.5	591.5
503.5	591.5
211.5	243.5
-195.5	-243.5
-471.5	-591.5
-471.5	-591.5
-195.5	-243.5
187.5	243.5
455.5	591.5
455.5	591.5
195.5	243.5
-171.5	-243.5
-423.5	-591.5
-423.5	-591.5
-179.5	-243.5
163.5	243.5
407.5	591.5
407.5	591.5
171.5	243.5
-155.5	-243.5
-391.5	-591.5
-391.5	-591.5
-163.5	-243.5

N=8 remained unchanged. In general, limit cycles due to the quantization algorithm described in Chapter 4, will occur with varying amplitudes as b_2 varies over the range $0.974<b_2<0.985$. Above this range no reduction of output amplitude is possible. Below this range the amplitude of the output will reduce until the effect of rounding of the products to the nearest unity becomes dominant.

Although Equation 5.27 indicates that N is a function of b_2, over the range $0.969<b_2<0.985$ the value of N remains essentially unchanged for a fixed value of b_1. However, b_1 varies over the range $-2<b_1<2$ and determines N and, according to 5.24, the frequency F.

To get an intuitive insight into the second-order process we rewrite Equation 5.14 as

$$-y_n = R[b_1\hat{y}_{n-1}] + R[b_2\hat{y}_{n-2}] \qquad 5.30$$

where the quantzation process is temporarily ignored. Keeping in mind that for a lowpass filter with passband below Fs/4 the coefficient b_1 is negative, the Equation 5.30 can be represented for one frequency cycle as shown in Fig.5.07. Each output sample at instant nT is computed from the two preceding products at instants nT-T and nT-2T. The critical calculation ocurs when the output sample at nT+N

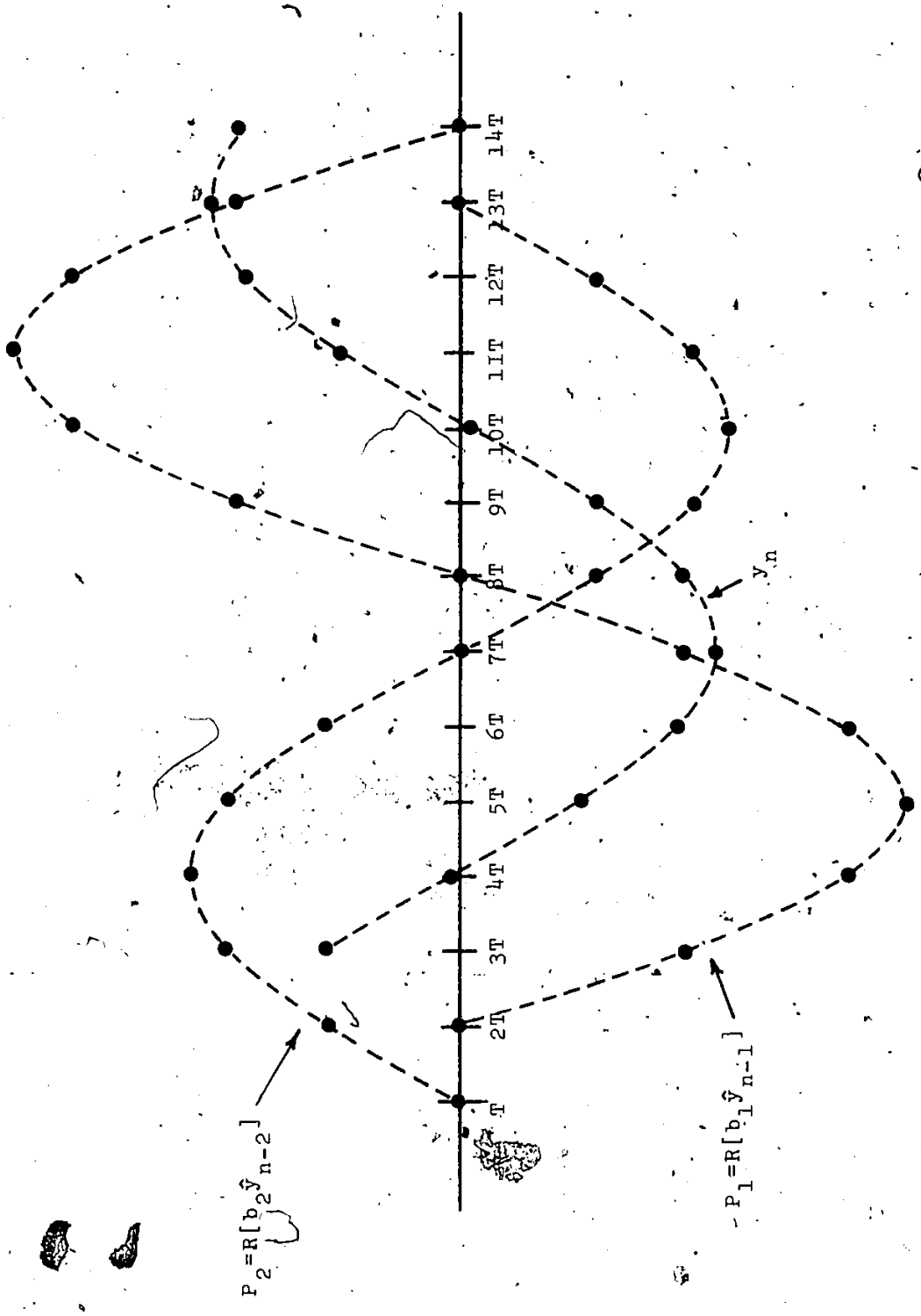


Fig. 5.07 Signal at the input and the output of the adder of the second-order recursive filter.

is computed. If the condition

$$|y(nT+N)| = |y(nT)| \qquad 5.31$$

where

$$y(nT) = P_1(nT-T) + P_2(nT-2T) \qquad 5.32$$

is satisfied then a limit cycle has occured. However, if

$$|y(nT+N)| < |y(nT)| \qquad 5.33$$

then a reduction in the limit cycle amplitude has taken place. By a proper choice of b_2 the amplitudes of P_1 and P_2 curves of Fig.5.07 can be reduced so that Equation 5.33 is satisfied. However, the quantization process may reverse the reduction and the equality condition of 5.31 will again hold. It is proposed that if every point on the P_2 curve is reduced in amplitude by the amount equal to one half the quantum step, 2 or more, the quantization process will not be able to counteract the effect of b_2. Although this is not proved analytically, the validity of the statement has been supported by extensive computer simulation. The result is particularly valid in establishing a lower bound for b_2 below which no limit cycles resulting from the effects of the logarithmic signal quantization are possible.

To ensure that the output signal amplitude undergoes a quantum jump to the next lower quantum level it is, therefore, necessary to satisfy

$$|\hat{y}_{n-2}(s,q)| - b_2|\hat{y}_{n-2}(s,q)| > 2^{s-2} \qquad 5.34$$

from which we obtain the condition which b_2 must satisfy

$$b_2 < 1 - \frac{2^{s-2}}{2^{s-1}(q + 16.5) - 0.5} \qquad 5.35$$

If b_2 is chosen according to Equation 5.35 and Equation 5.14 is expressed as

$$\hat{y}_n(s,q) = -Q\{R[b_1\hat{y}_{n-1}(s_1,q_1)] + \operatorname{sgn}[\hat{y}_{n-2}(s_2,q_2)][|\hat{y}_{n-2}(s_2,q_2)| - 2^{s-2}]\} \qquad 5.36$$

then the amplitude of each output sample will reduce every N-th sample until all succeeding sample amplitudes fall below the quantum defined by the s and q in Equation 5.35. The lower bound we seek for b_2 corresponds to s=2 and q=0, or

$$b_2 < 1 - \frac{1}{32.5} \qquad 5.37$$

For this value of b_2 the output samples will continue to reduce every N-th sample until all succeeding samples fall within the range of the two lower segments, s=0 and s=1. In these two segments the limit cycles are dependent only on

the effects of rounding the products to the nearest unity. The results in this range have been studied by Jackson and others.

The bound corresponding to the Equation 5.37 is represented in Fig.5.08 by a horizontal line just inside the stability triangle. This completes the set of coefficient bounds resulting from the effects of logarithmic signal quantization.

To illustrate the relative magnitude of the bounds, coefficient pairs for two second-order filter sections satisfying PCM filtering requirements are plotted inside the stability triangle. The exact coefficients appear as co-ordinates alongside each of the two points. These points fall well outside the specified bounds.

5.06 Conclusions

Limit cycles resulting from the use of the segmented A-law quantization of the recursive signal in first and secon order filters do not represent a problem. Such limit cycles are only possible for coefficients falling inside narrow strips along the boundaries of the stability triangle in the coefficient plane. Useful filters can be designed for which the coefficients fall outside these boundaries.

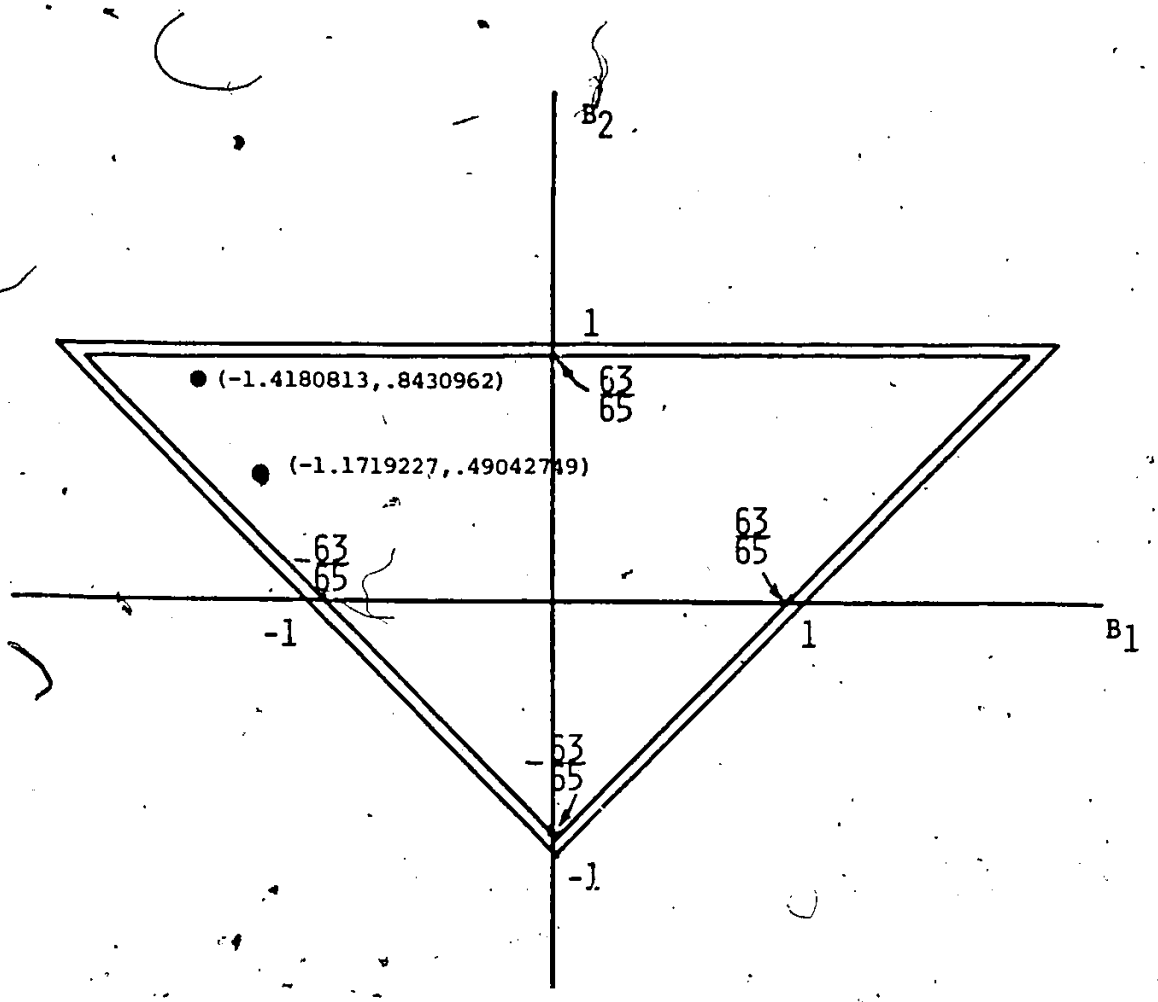


Fig.5.08 Bounds on coefficients for which limit cycles as the result of A-Law quantization of the recursive signal are possible.

CHAPTER 6

NOISE DUE TO FINITE ARITHMETIC

6.01 General

The entire function of a digital filter is to compute the numerical value of the next output sample by performing binary arithmetic calculations involving transfer function coefficients and other signal samples. The finite nature of digital implementations imposes round-off and truncation errors on the numerical quantities involved, leading to output signal distortion. Although the accuracy of the input signal representation is also necessarily finite, the resulting distortion cannot be corrected by any improvements in the filter design techniques. Coefficient rounding and product rounding remain as the sources of error under the filter designer's control.

Coefficient round-off errors introduce a deterministic perturbation in the transfer function, resulting in a predictable error in the frequency response. Product rounding

errors lead to random output noise characterized by the signal sample probability distribution function.

In SPDF the coefficients do not appear explicity but are contained within the products as multipliers of the individual signal samples. The procedure for performing a multiplication in SPDF is, therefore, different. At the design stage of the filter, all possible products of an input signal and a highly accurate coefficient are evaluated using a general purpose computer. Each rounded or truncated value of the product is stored as a word in ROM to be addressed by the corresponding quantized value of the signal. Since the products are stored in fixed point form the product round-off error depends on the magnitude of the product and therefore on the probability distribution of the signal samples. The SPDF is characterized by accurate coefficient values and average round-off errors determined by the signal probability distribution function.

Product rounding is the only mechanism by which SPDF introduces signal distortion as the result of finite arithmetic. This chapter presents a detailed study of the noise introduced by the product rounding process. The effect of noise on the response characteristic is examined in the light of filter first-order sensitivities and the Laplacian speech probability density function. It is shown that with the noise distortion of the response, the filter meets the specifications of the current PCM systems.

6.02 Rounding of Stored Products

The procedure used for the computation of SPDF products is shown in Fig.6.01, (24). The coefficient round-off error may be thought of as being arbitrarily small by attributing the round-off noise effects to product rounding only, with statistical dependence on the signal. The noise effects due to the product round-off can be evaluated in the following way. We define the error, E^2, in the energy of the signal at the output of a filter as

$$E^2 = \sum_{n=0}^{\infty} [y_\infty(n) - y(n)]^2 \qquad 6.01$$

where $y(n)$ and $y_\infty(n)$ are the rounded and the ideal output signals of the filter, repspectively. A computational technique based on the above definition can then be represented as in Fig.6.02, (11). In this figure, $H_\infty(z)$ is the ideal transfer function of the filter while $H_R(z)$ is the same transfer function with the coefficients rounded to L_C bits and the products rounded to L_S bits. The SPDF behaviour will be approached when L_C approaches the 36-bit wordlength.

This method was applied to the second order transfer function with coefficient values;

$a_0 = a_2 = 0.86597443274$
$a_1 = -1.2246727875$
$b_1 = -1.70822940427$
$b_2 = 0.89710660898$

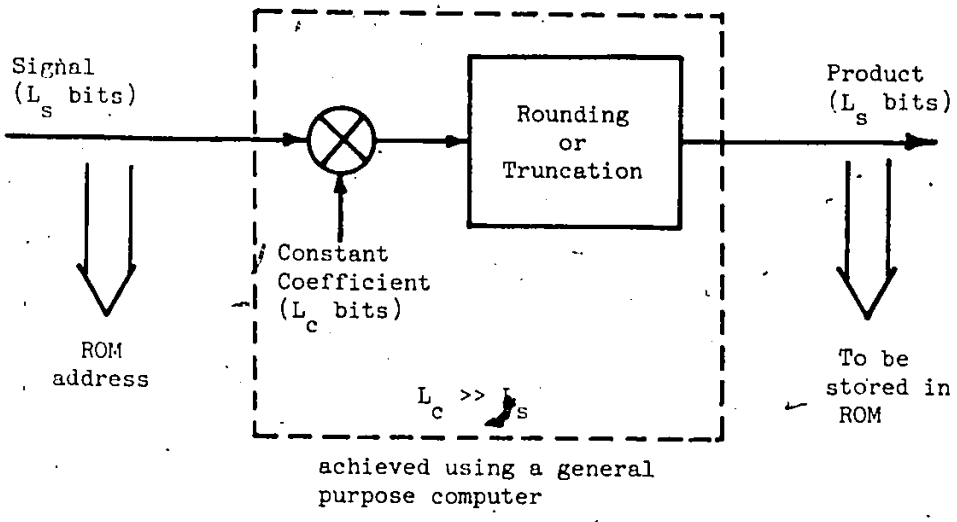


Fig.6.01 Multiplication procedure used for programming ROM.

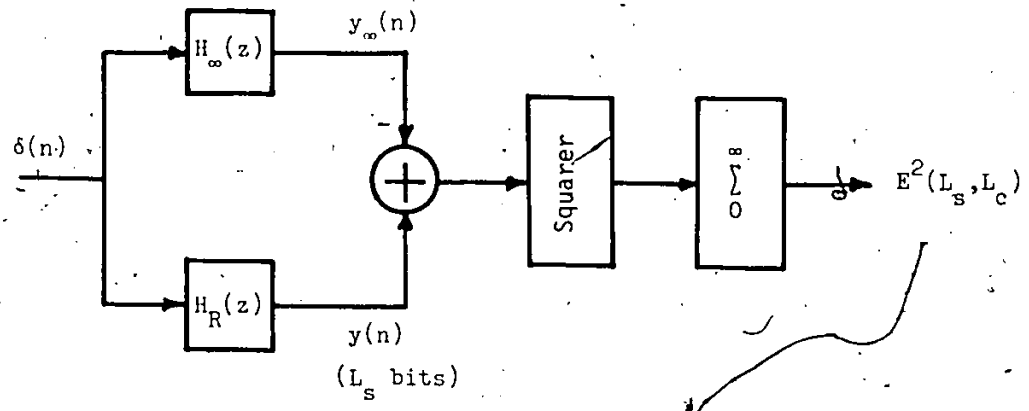


Fig.6.02 Technique for measuring the error due to rounding.

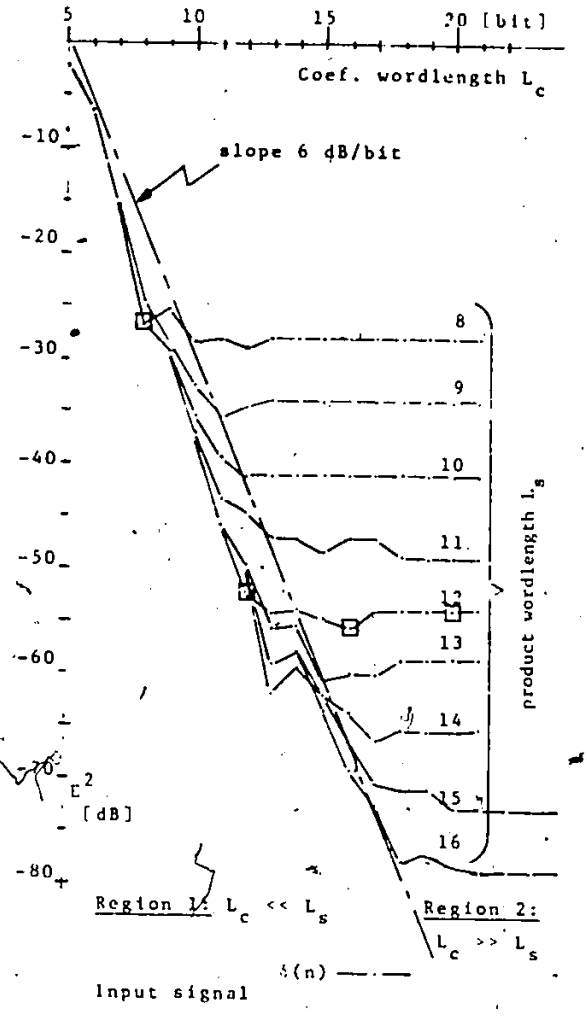


Fig.6.03 Error E^2 in the impulse response as a function of L_C

The base-10 logarithm of the resulting E^2 error, 20Log (E), for several values of L_C and L_S is plotted in Fig.6.03.

Four cases of filter insertion loss response corresponding to the 8-bit, 12-bit, 16-bit and 20-bit coefficient wordlength and 12-bit product wordlength were also computed for comparison. The loss deviations from the ideal insertion loss response over the filter passband frequencies are given in TABLE 6.01. The RMS deviations for each of the four cases were computed and plotted as squares on the L_S=12 curve of Fig.6.03. It can be seen that the vertical separations between these points are close to the expected values of 26dB, 8dB and 0dB (which correspond to the ratios 0.169/0.00794, 0.00794/0.00308 and 0.00308/0.00306).

TABLE 6.01

Inserion loss as a function of coefficient wordlength L_C

FREQ.	(Ideal) Insertion Loss	Deviation in dB from the ideal loss due to coefficient rounding			
		8 bits	12 bits	16 bits	20 bits
400	8.75	.11	3E-03	-1E-03	0
600	8.985	.114	3E-03	0	0
800	9.312	.121	3E-03	-1E-03	-1E-03
1000	9.78	.139	1E-03	-2E-03	-2E-03
1200	10.411	.142	4E-03	-1E-03	1E-03
1400	11.243	.158	4E-03	-1E-03	1E-03
1600	12.421	.174	2E-03	1E-03	-2E-03
1800	14.047	.214	2E-03	-2E-03	-1E-03
2000	15.999	.216	2E-03	-2E-03	0
2200	17.466	.047	-5E-03	0	1E-03
2400	16.312	-.251	-.016	-1E-03	-8E-03
2500	14.156	-.293	-.018	-2E-03	-2E-03
2600	11.911	-.257	-.012	-2E-03	2E-03
2700	9.676	-.23	-9E-03	-3E-03	0
2800	7.441	-.217	-.01	-3E-03	-4E-03
2900	5.529	-.169	-9E-03	-1E-03	0
3000	3.639	-.151	-.011	-6E-03	-7E-03
3100	1.621	-.153	-1E-03	.012	5E-03
3200	- .379	-.115	-.012	3E-03	-3E-03
3300	-2.255	-.102	-5E-03	-5E-03	-9E-03
3400	-4.417	-.117	-.017	-1E-03	-1E-03
RMS of total deviation:		0.169	7.94E-03	3.08E-03	3.06E-03

In Fig.6.03 each curve is dominated by two asymptotes. In Region 1, where L_c is less than L_s, the error is mainly due to the coefficient quantization. The mean slope of the curves in this region is 6dB per bit. In Region 2, where L_c is greater than L_s, the error is mainly due to the signal quantization and product round-off. One-bit increase in L_c in this region will increase E^2 by 6dB, the distance between two horizontal curves. Digital filters employing short wordlengths (L_c less than L_s) are operated in Region 1; while SPDFs are operated in Region 2. In Region 2 the choice of L_s for the product wordlength determines the minimum error E^2. In Region 1, changes in product wordlength for constant coefficient accuracy do not result in error reduction. In Region 2, the overall error can be meaningfully reduced by an increase of the product wordlength. This property is unique to stored-product digital filters. It allows to disregard the deterministic change in the pole locations and the loss characteristic and consider only the effects of product rounding.

6.03 The Effect of Product Rounding on the Loss Characteristic

In the present application, product rounding to the accuracy of the signal will be used. The filter will, therefore, operate on or to the right of the 6dB line separating Region 1 and Region 2. The effect of product rounding on the filter loss characteristic can be established by way of an assumed coefficient roundoff, i.e. if the coefficients had

to be re-established form the stored products, the effective coefficient error could be determined. The average coefficient error magnitude can be used to compute the average insertion loss deviatiation of the filter from the ideal loss characteristic. In this way a probabilistic bound on the loss characteristic can be established. The computational operations of the SPDF are usually done in fixed-point arithmetic; the products are stored as binary integers expressed in two's complement form. Thus, in an N-bit system the maximum product magnitutde is 2^N, where N is typically 12 bits. Given a signal sample value, x, and a coefficient, a, the exact product is computed as

$$P = a\hat{x} \qquad 6.02$$

When rounded to the nearest unity the product can be expressed as

$$P' = R(a\hat{x}) \qquad 6.03$$

This can be interpreted as rounding the coefficient to the new value given by

$$a' = \frac{R(a\hat{x})}{\hat{x}} \qquad 6.04$$

Assuming that the accuracy of the coefficient has not been restricted in any way other than by the product register length, it can be shwon that the resulting fractional error

in the coefficient is equal to the fractional error in the corresponding rounded product. If ΔP represents the product round-off error, then

$$\frac{\Delta P}{P} = \frac{a\hat{x} - R(a\hat{x})}{a\hat{x}} \qquad 6.05$$

$$= \frac{a - \frac{R(a\hat{x})}{\hat{x}}}{a} \qquad 6.06$$

Using equation 6.03 the fractional product error can be expressed as

$$\frac{\Delta P}{P} = \frac{a - a'}{a} \qquad 6.07$$

or

$$\frac{\Delta P}{P} = \frac{\Delta a}{a} \qquad 6.08$$

where Δa is the effective coefficient round-off error resulting from product rounding. This effective coefficient rounding appears as a probabilistic effect which differs from the deterministic effect of coefficient rounding of conventional multiplier digital filters. The magnitude of a nonzero stored product in a 12-bit system will vary between the minimum value of 000000000001 (unity), and the maximum of 011111111111 (2047). These values correspond to R(ax). Given that both products are generated by the same coefficient it can be seen that the effective coefficient

will be represented to a different degree of accuracy over the signal dynamic range.

Consider the positive signal sample values of Table 4.02. Using the coefficient a_0=0.272623 as an example, the corresponding ROM contents are given in Table 6.02 below.

TABLE 6.02

Positive ROM contents corresponding to a0=0.272623

q	s=0	s=1	s=2	s=3	2=4	s=5	s=6	s=7
0	0	4	9	18	36	72	144	288
1	0	5	9	19	38	76	153	305
2	1	5	10	20	40	81	161	323
3	1	5	10	21	42	85	170	340
4	1	5	11	22	45	89	179	358
5	1	6	12	23	47	94	187	375
6	2	6	12	24	49	98	196	392
7	2	6	13	25	51	102	205	410
8	2	7	13	27	53	107	214	427
9	2	7	14	28	55	111	222	445
1	3	7	14	29	58	115	231	462
11	3	7	15	30	60	120	240	480
12	3	8	15	31	62	124	248	497
13	4	8	16	32	64	129	257	515
14	4	8	16	33	66	133	266	532
15	4	8	17	34	69	137	275	549

A similar table of negative values would complete the set of products for the coefficient a_0. The complete binary representation of these products is given in Appendix 7C of page 7C-2. When each product of Table 6.02 is divided by the corresponding sample value of Table 4.02 we obtain a set of a_0 values perturbed by product rounding. These are given in Table 6.03

TABLE 6.03

Perturbed values of a_0=0.272623 resulting from product rounding

q	s=0	s=1	s=2	s=3	s=4	s=5	s=6	s=7
0	.00000	.25000	.27692	.27481	.27376	.27324	.27299	.27286
1	.00000	.29412	.26087	.27338	.27240	.27191	.27346	.27244
2	.50000	.27778	.27400	.27211	.27119	.27411	.27219	.27292
3	.33333	.26316	.25974	.27097	.27010	.27287	.27265	.27255
4	.25000	.25000	.27160	.26994	.27523	.27176	.27307	.27297
5	.20000	.28571	.28235	.26901	.27405	.27365	.27200	.27263
6	.33333	.27273	.26966	.26816	.27298	.27260	.27241	.27232
7	.28571	.26087	.27957	.26738	.27200	.27164	.27279	.27270
8	.25000	.29167	.26804	.27692	.27110	.27331	.27313	.27241
9	.22222	.28000	.27723	.27586	.27027	.27239	.27223	.27276
10	.30000	.26923	.26667	.27488	.27423	.27155	.27257	.27249
11	.27273	.25926	.27523	.27397	.27335	.27304	.27288	.27280
12	.25000	.28571	.26549	.27313	.27253	.27222	.27208	.27255
13	.30769	.27586	.27350	.27234	.27176	.27359	.27239	.27285
14	.28571	.26667	.26446	.27160	.27105	.27282	.27268	.27261
15	.26667	.25806	.27200	.27092	.27435	.27210	.27295	.27239

It can be seen that the average coefficient magnitude and the average error magnitude can be computed for a given statistical distribution of the sample amplitudes.

The total error in the filter loss characteristic is determined not only by the rounding errors but also by the sensitivity of the characteristic to those errors. In the next section the probability of occurrence of each sample will be found which will lead to the average magnitude of the coefficient error. In the following section the filter sensitivity to each coefficient error will be derived. This will complete the tools necessary for arriving at the average deviation of the filter characteristic from the ideal.

6.04 Probability Distribution of Quantized Samples

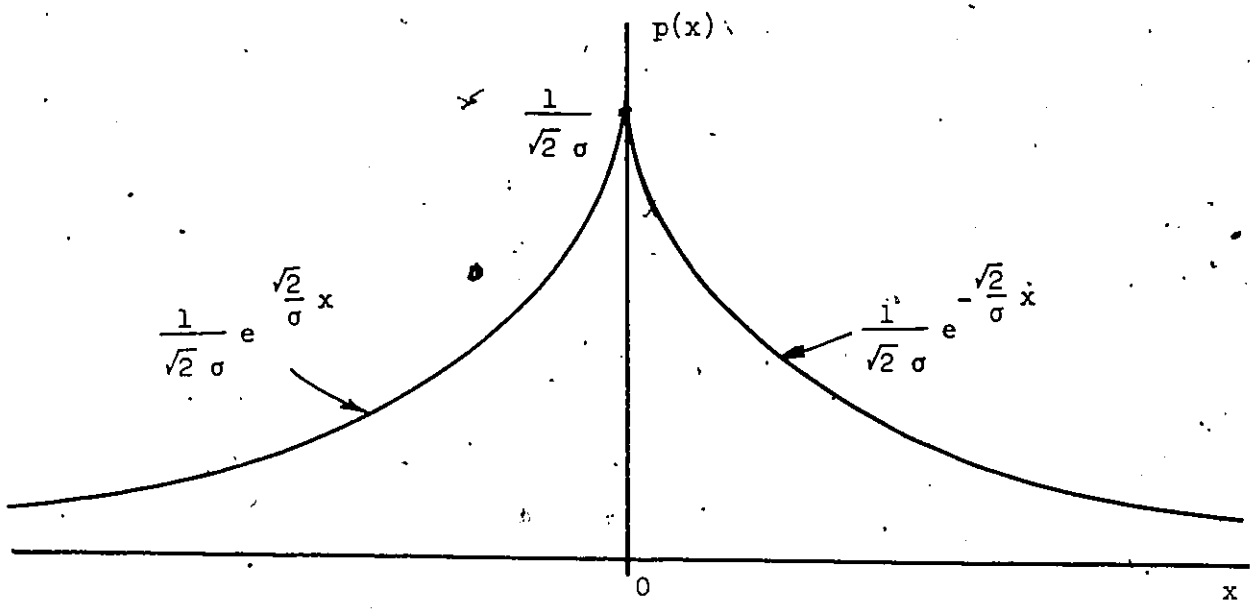
In this section the probability of occurrence of each quantized sample of the signal will be computed in order to determine the average coefficient error. To be consistent with the main application example of PCM, the probability density function corresponding to the speech signal will be assumed. In such applications industrial designers ususally assume the Laplacian amplitude probabiliy density function of the form (26)

$$p(x) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}}{\sigma}|x|} \qquad 6.09$$

where σ is the RMS speech voltage. The continuous form of the function is shown in Figure 6.04. To compute the average coefficient error, however, discrete values of probability, $p(\hat{x}(s,q))$, corresponding to each sample amplitude $\hat{x}(s,q)$ must be found. For this purpose only the positive values listed in TABLE 4.02 need to be considered.

Figure 6.05 illustrates the x-axis subdivision into segments, quanta and integers according to the quantization characteristic discussed in Chapter 4. $p(\hat{x}(s,q))$ is simply the area strip subtended by the particular quantum defined by s and q. Thus,

$$p\{\hat{x}(s,q)\} = \frac{1}{\sqrt{2}\sigma} \int_{L_2}^{L_1} e^{-\frac{\sqrt{2}}{\sigma}x} dx \qquad 6.10$$



Fit. 6.04. Amplitude probability density function for speech

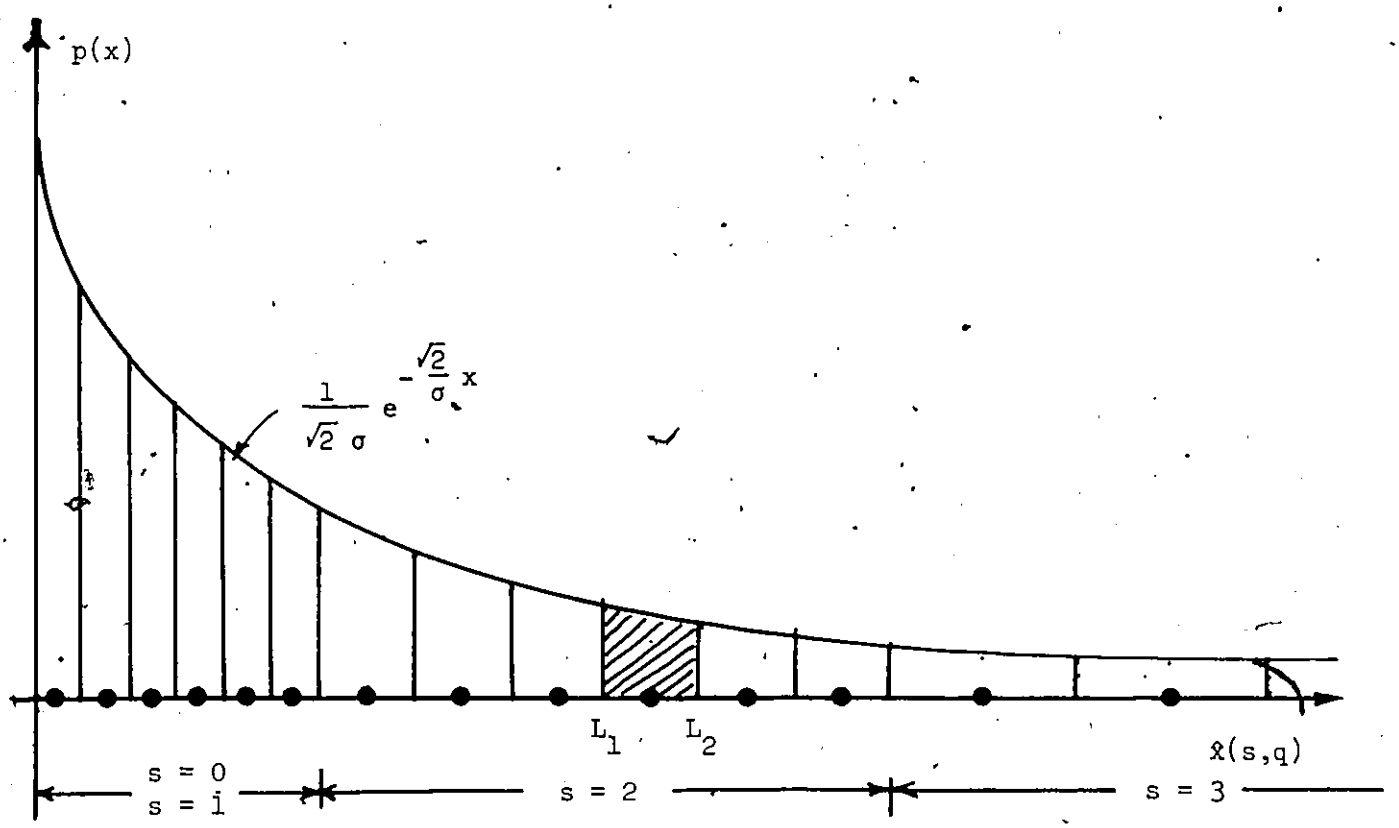


Fig.6.05 Probability density function for quantized samples.

where L1 and L2 represent the bounds defining the corresponding quantum. For s=0 and s=1 these bounds are given by

$$L_1 = \hat{x}(s,q) - 0.5 \qquad 6.11$$

and

$$L_2 = \hat{x}(s,q) + 0.5 \qquad 6.12$$

For s>1 the bounds are

$$L_1 = \hat{x}(s,q) - 2^{s-2} \qquad 6.13$$

and

$$L_2 = \hat{x}(s,q) + 2^{s-2} \qquad 6.14$$

Consider the general property of probability density functions,

$$\int_{-\infty}^{\infty} p(x)dx = 1 \qquad 6.15$$

Approximately 98% of the area under the Laplacian probability density function is included between the 3σ points. For the quantized signals, 3σ will be set at 2048 with σ=683 and

$$\int_{0}^{2048} p(x)dx = 0.49 \qquad 6.16$$

Using integral tables we can perform the following substitution

$$\frac{1}{\sqrt{2}\sigma}\int_{L_2}^{L_1} e^{-\frac{\sqrt{2}}{\sigma}x}\,dx = \tfrac{1}{2}\left[e^{-\frac{\sqrt{2}}{\sigma}L_1} - e^{-\frac{\sqrt{2}}{\sigma}L_2}\right] \qquad 6.17$$

The required working equations are then obtained from 6.10 and 6.16 as

$$p\{\hat{x}(s,q)\} = \tfrac{1}{2}\left[e^{-\frac{\sqrt{2}}{\sigma}L_1} - e^{-\frac{\sqrt{2}}{\sigma}L_2}\right] \qquad 6.18$$

and

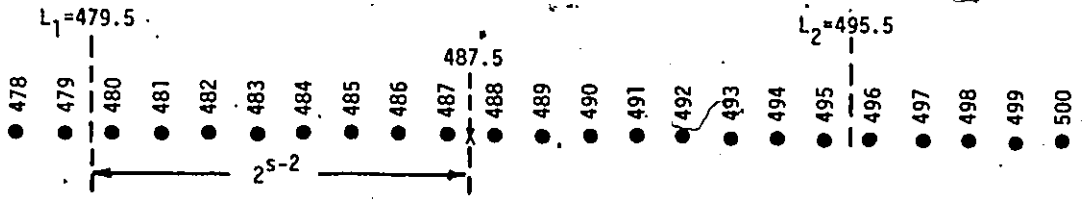
$$\sum_{s,q} p\{\hat{x}(s,q)\} = \tfrac{1}{2}\left[e^{-\frac{\sqrt{2}}{\sigma}L_1} - e^{-\frac{\sqrt{2}}{\sigma}L_2}\right] \qquad 6.19$$

Example

Compute the probability of occurrence of the signal sample defined by s=5 and q=14, assuming σ=1439.5.

Solution:

A representation of the integers belonging to the quantum corresponding to s=5 and q=14 is shown below. L1 and L2 define the boundaries of the quantum.



$$\hat{x}(5,14) = 487.5$$

$$L_1 = 487.5 - 2^{s-2} = 479.5$$

$$L_2 = 487.5 + 2^{s-2} = 495.5$$

$$p\{487.5\} = \tfrac{1}{2}\left[e^{-\frac{\sqrt{2}}{1439.5}479.5} - e^{-\frac{\sqrt{2}}{1439.5}495.5}\right]$$

$$= 0.0048685$$

The values of σ which satisfy 6.19 for several values of the signal dynamic range, are given in TABLE 6.04.

TABLE 6.04
Maximum and RMS sample amplitudes

x(max)	σ(RMS)
8063.5	2731
4031.5	1365
2015.5	683
1007.5	341
503.5	171
251.5	85

Each doubling of the dynamic range allows the addition of one bit to the product word length and requires the corresponding doubling of the total ROM Capacity. Extension of the product word length by one bit simply extends the accuracy of the coefficient by one bit as supported by Equation 6.08.

The individual probabilities for eah of the sample amplitudes can now be computed using Equation 6.18. The case corresponding to $\hat{x}(max)=2015.5$ of the 12-bit system is of interest in the PCM application. The probabilities for this case have been computed for all 128 sample values and are given in the Appendix 6A.

6.05 Average Coefficient Error

A given coefficient, a, when stored as part of a product $R(a\hat{x}(s,q))$ undergoes a rounding process which results in the coefficient error $e(s,q)$. The probability $p(e(s,q))$ depends on the product involved which in turn depends on the probability $p(\hat{x}(s,q))$,

$$p\{e(s,q)\} = p\{\hat{x}(s,q)\} \qquad 6.20$$

The average coefficient round-off error can then be calculated from the usual averaging formula

$$\overline{|\Delta a|} = \sum_{s,q} p\{e(s,q)\} \times |e(s,q)| \qquad 6.21$$

Substituting 6.20 and 6.06 into 6.21 we obtain the complete expression

$$\overline{|\Delta a|} = \sum_{s,q} p\{x(s,q)\} \times \left\{ \frac{\left| a - \frac{R[a\hat{x}(s,q)]}{\hat{x}(s,q)} \right|}{|a|} \right\} \qquad 6.22$$

Given the numerical vaue for a coefficient and the set of sample probabilities, the average coefficient round-off error can be computed directly from 6.22.

The coefficients of interest are those of the third-order transfer function,

$$H(z) = \frac{0.2726230+0.0808208z^{-1}+0.0808208z^{-2}+0.2726230z^{-3}}{1-0.9877751z^{-1}+0.7787396z^{-2}-0.08407628z^{-3}} \qquad 6.23$$

This transfer function is designed to satisfy the PCM requirements and will be further discussed in Chapter 7. With the aid of a simple computer program the average coefficient error was computed for each of the coefficients. The results are given in TABLE 6.05 below.

TABLE 6.05

Average coefficient error.

Coef.	Value	Average Error
a_0	0.27262300	0.00164795085
a_1	0.08082080	0.00148833027
a_2	0.08082080	0.00148833027
a_3	0.27262300	0.00164795085
b_1	0.98777510	0.00178338404
b_2	0.77873960	0.00166771600
b_3	0.09407628	0.00149247878

To achieve the deciminal resolution of 0.0016, at least 0000000001 in binary is required. If the sign bit is included then at least 11-bit coefficient representation is required in order not to exceed the typical coefficient error of TABLE 6.05. In a system with 12-bit signal representation a filter using conventional multipliers would have to perform an 11x12-bit multiplication before product rounding is carried out.

It should be noted that the number of bits required to represent the coefficients is dependent on the speech probability density function parameters. For example, with σ=2048 the same probability function approximates the equiprobable case for which the average coefficient error is approximately 0.0006332. The corresponding coefficient accuracy can be achieved with a 12-bit representation.

6.05 Filter Sensitivity to Roundoff Error

In the previous section the average coefficient error, Δa, due to arithmetic rounding was determined. It is now possible, as the next step, to examine the effect of this rounding on the filter transfer function as a whole. Once the resulting perturbation in the transfer function is known as a function of frequency it will be possible to predict the corresponding deviation of the filter loss characteristic from its designed value.

The transfer function of interest is that given by 6.23; in its general form it can be expressed as

$$H(z) = \frac{a_0+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}}{1+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}} \qquad 6.24$$

The aim is to determine a general expression for the fractional transfer function perturbation, $\Delta H(z)/H(z)$, in terms of the coefficient perturbations $\Delta ai/ai$ and $\Delta bi/bi$. If there are N coefficients in H(z) and $\Delta Hi(z)$ is the perturbation in H(z) due to the perturbation in the i-th coefficient, then the toal fractional perturbation in H(z) is given by (5)

$$\frac{\Delta H(z)}{H(z)} = \sum_{i=1}^{N} \frac{\Delta H_i(z)}{H(z)} \qquad 6.25$$

If the sensitivity function, $S(H,a_i)$, is defined as the ratio of the fractional change in H(z) to the fractional change in the coefficient a_i, then

$$S(H,a_i) = \frac{\frac{\Delta H_i(z)}{H(z)}}{\frac{\Delta a_i}{a_i}} \qquad 6.26$$

Equation 6.25 may now be rewritten to include 6.26,

$$\frac{\Delta H(z)}{H(z)} = \sum_{i=1}^{N} \frac{\frac{\Delta H_i(z)}{H(z)}}{\frac{\Delta a_i}{a_i}} \times \frac{\Delta a_i}{a_i} \qquad 6.27$$

or

$$\frac{\Delta H(z)}{H(z)} = \sum_{i=1}^{N} S(H,a_i) \frac{\Delta a_i}{a_i} \qquad 6.28$$

It can be seen from TABLE 6.03 that the coefficient perturbations of interest are well below 1%; it will be shown later that $\Delta H(z)$ is also correspondingly small. This permits the use of the approximation

$$\frac{\Delta H_i}{\Delta a_i} \simeq \frac{\partial H_i}{\partial a_i} \qquad 6.29$$

where the argument, z, has been dropped to simplify the notation. Applying the approximation of 6.29 to 6.26 we obtain

$$S(H,a_i) = \frac{a_i}{H} \frac{\partial H_i}{\partial a_i} \qquad 6.30$$

This equation leads to the derivation of a closed-form expression for the sensitivity function. If the transfer function is expressed as

$$H(a_i,b_i) = \frac{H_N(a_i)}{H_D(b_i)} \qquad 6.31$$

then by applying the rules of partial differentiation of rational expressions we obtain

$$S(H,a_i) = \frac{a_i}{H_D}\frac{\partial H_N}{\partial a_i} \qquad 6.32$$

and

$$S(H,a_i) = -\frac{b_i}{H_D}\frac{\partial H_D}{\partial b_i} \qquad 6.33$$

We observe that the partial derivative of a polynomial with respect to the i-th coefficient involves only the i-th term. Therefore,

$$\frac{\partial H_N}{\partial a_i} = z^{-i} \qquad 6.34$$

and

$$\frac{\partial H_D}{\partial b_i} = z^{-i} \qquad 6.35$$

Finally, the sensitivity functions may be expressed as

$$S(H,a_i) = \frac{a_i z^{-i}}{H_N(a_i)} \qquad 6.36$$

and

$$S(H,b_i) = \frac{-b_i z^{-i}}{H_D(b_i)} \qquad 6.37$$

These expressions are functions of frequency and can be expressed in terms of jw by making the substitution

$$z = e^{j\omega T} \qquad 6.38$$

where $j = \sqrt{-1}$. Substituting and dividing by $e^{j\omega T}$, we get

$$S(H,a_i) = \frac{a_i}{a_0 e^{j\omega i T} + a_1 e^{j\omega(i-1)T} + a_2 e^{j\omega(i-2)T} + a_3 e^{j\omega(i-3)T}} \qquad 6.39$$

and

$$S(H,b_i) = \frac{-b_i}{e^{j\omega i T} + b_1 e^{j\omega(i-1)T} + b_2 e^{j\omega(i-2)T} + b_3 e^{j\omega(i-3)T}} \qquad 6.40$$

In each case the denominator can be expressed in terms of real and imaginary parts

$$S(H,a_i) = \frac{a_i}{R_{a_i} + jI_{a_i}} \qquad 6.41$$

and

$$S(H,b_i) = \frac{-b_i}{R_{b_i} + jI_{b_i}} \qquad 6.42$$

The loss-frequency characteristic we are seeking is specified by the absolute value, $|H(e^{j\omega T})|$, of the transfer function H(z). Therefore, it is the sensitivity of $|H(e^{j\omega T})|$ to the various coefficients that must be found.

$$S(|H(e^{j\omega T})|,a_i) = \mathrm{Re}\{S(|H(e^{j\omega T})|,a_i)\} \qquad 6.43$$

Using 6.43 we obtain

$$\mathrm{Re}\{\frac{a_i}{R_{a_i} + jI_{a_i}}\} = \frac{a_i R_{a_i}}{R_{a_i}^2 + I_{a_i}^2} \qquad 6.44$$

Hence, 6.41 and 6.42 become

$$S(|H|,a_i) = \frac{a_i R_{a_i}}{R_{a_i}^2 + I_{a_i}^2} \qquad 6.45$$

and

$$S(|H|,b_i) = \frac{-b_i R_{b_i}}{R_{b_i}^2 + I_{b_i}^2} \qquad 6.46$$

It remains only to find the expressions for Ra_i, Ia_i, Rb_i, and Ib_i for i=0,1,2 and 3.

Applying the identity

$$e^{j\omega T} = \cos(\omega T) + j\sin(\omega T) \qquad 6.47$$

to the denominators of 6.39 and 6.40 and grouping the terms

$$R_{a_i} = a_0\cos(i)\omega T + a_1\cos(i-1)\omega T + a_2\cos(i-2)\omega T + a_3\cos(i-3)\omega T \qquad 6.48$$

$$I_{a_i} = a_0\sin(i)\omega T + a_1\sin(i-1)\omega T + a_2\sin(i-2)\omega T + a_3\sin(i-3)\omega T \qquad 6.49$$

$$R_{b_i} = \cos(i)\omega T + b_1\cos(i-1)\omega T + b_2\cos(i-2)\omega T + b_3\cos(i-3)\omega T \qquad 6.50$$

$$I_{b_i} = \sin(i)\omega T + b_1\sin(i-1)\omega T + b_2\sin(i-2)\omega T + b_3\sin(i-3)\omega T \qquad 6.51$$

Using the above four expressions in 6.45 and 6.46 it is a straightforward matter to compute the sensitivities of the filter frequency response, $|H(e^{j\omega T})|$, to each of the seven coefficients. Note that $b_o=1$ and, therefore, can be implemented without error; consequently it is not included in the calculations that follow.

Next, the sensitivities are calculated for the transfer function specified in 6.23 The implementation of this transfer function is the subject of Chapter 7. Using the sampling frequency Fs=24000Hz and

$$T = \frac{1}{F_s} \qquad 6.52$$

we arrive at the results of TABLE 6.06, which are also plotted in Fig.6.06.

TABLE 6.06

Sensitivity to coefficients as a function of frequency

F(Hz)	a_0	a_1	a_2	a_3	b_1	b_2	b_3
200	.3854	.1146	.1146	.3854	1.4001	-1.1006	.1182
400	.3847	.1153	.1153	.3847	1.4084	-1.0975	.1158
600	.3834	.1166	.1166	.3834	1.4224	-1.0922	.1118
800	.3817	.1183	.1183	.3817	1.4424	-1.0842	.1062
1000	.3793	.1207	.1207	.3793	1.4687	-1.0732	.0988
1200	.3763	.1237	.1237	.3763	1.5018	-1.0583	.0895
1400	.3726	.1274	.1274	.3726	1.5423	-1.0384	.0780
1600	.3681	.1319	.1319	.3681	1.5907	-1.0118	.0642
1800	.3626	.1374	.1374	.3626	1.6477	-0.9760	.0475
2000	.3559	.1441	.1441	.3559	1.7133	-0.9272	.0275
2200	.3478	.1522	.1522	.3478	1.7873	-0.8594	.0035
2400	.3379	.1621	.1621	.3379	1.8673	-0.7637	-0.0255
2600	.3258	.1742	.1742	.3258	1.9476	-0.6257	-0.0608
2800	.3106	.1894	.1894	.3106	1.0147	-0.4235	-0.1035
3000	.2914	.2086	.2086	.2914	2.0397	-0.1240	-0.1547
3200	.2665	.2335	.2335	.2665	1.9657	-0.3155	-0.2129
3400	.2330	.2670	.2670	.2330	1.6959	0.9300	-0.2707
3600	.1960	.3140	.3140	.1860	1.1165	1.6852	-0.3089
3800	.1157	.3843	.3843	.1157	0.2173	2.3876	-0.2993
4000	.0000	.5000	.5000	.0000	-0.7617	2.7398	-0.2310

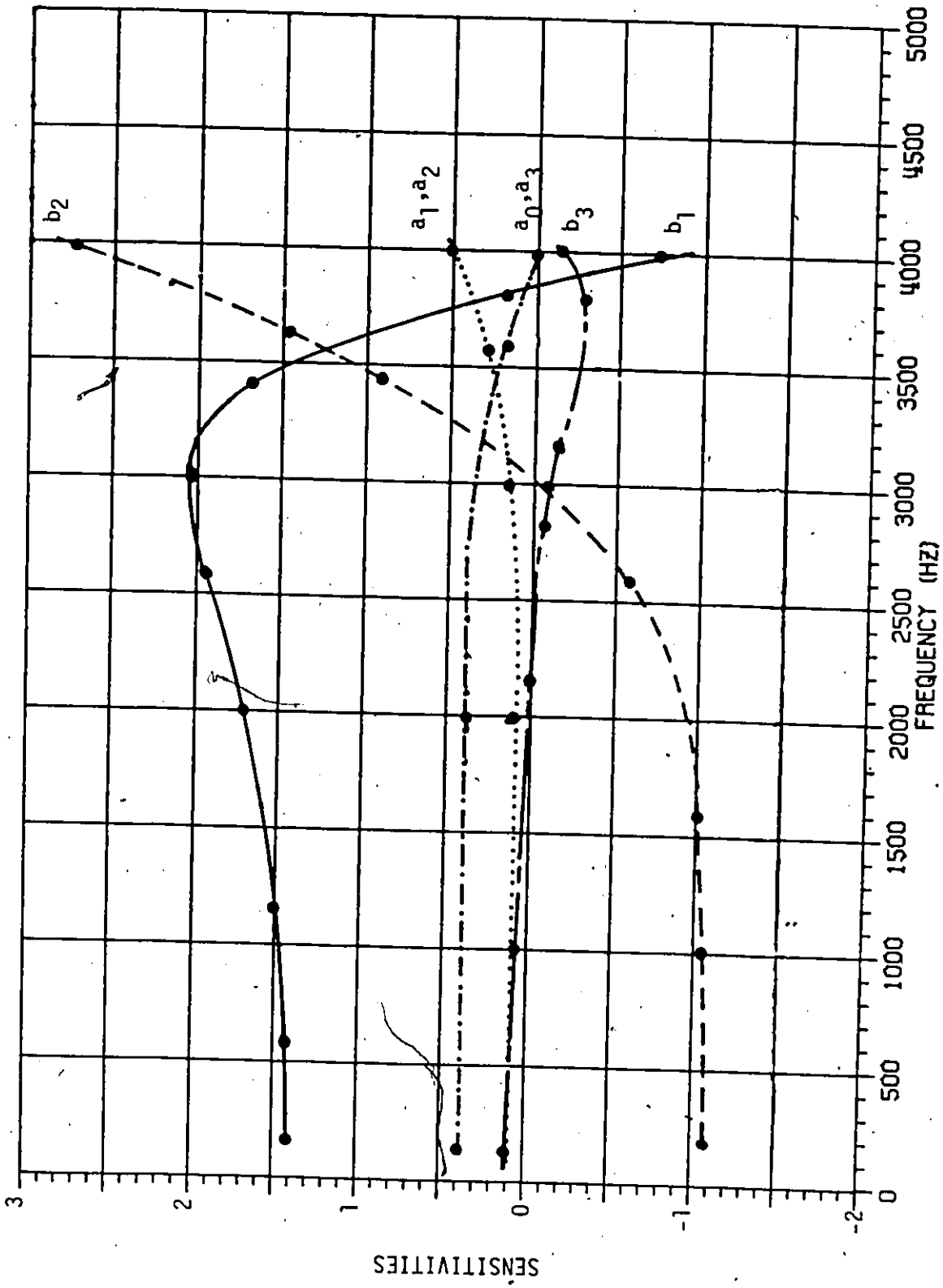


Fig.6.06 Coefficient sensitivities for the third-order SPDF.

The sensitivities to the recursive coefficients b_1, b_2 and b_3 are particularly important because of their relationship to the natural modes which control the inband ripple and filter stability. The sensitivities to the numerator coefficients influence primarily the stopband of the filter. Usually, however, the design margins in this area are less stringent and the stability of the filter is not affected.

6.07 Loss Distortion due to Roundoff Error

Filter designer's objectives center around the requirement to maintain the frequency-loss behaviour of the filter within tight limits set by the overall system considerations. In analog systems variation in the system loss performance come about as the result of many diverse effects such as manufacturing tolerances, temperature-humidity changes and short and long term aging. An effective systems design procecedure can only be implemented if loss variations due to all factors can be predicted in mathematical and statistical terms.

SPDF has two sources which contribute to the loss distortion. The effect of logarithmic signal quantization on the loss characteristic was discussed in Chapter 4. Loss distortion due to product rounding will be discussed in this section.

In Section 6.05 the average coefficient error due to rounding of stored products was determined; in Section 6.06 the filter frequency response sensitivity to coefficient perturbations was derived. These results can now be combined to express the average change in the frequency response due to product rounding,

$$\overline{\frac{\Delta|H|}{|H|}} = \sum_{i=0}^{N} S(H, a_i)\frac{\overline{\Delta a_i}}{a_i} + \sum_{i=1}^{N} S(H, b_i)\frac{\overline{\Delta b_i}}{b_i} \qquad 6.53$$

where the horizontal bar indicates an average quantity, and N the order of the filter. The average loss distortion, in dB, is then simply

$$\overline{\Delta L} = 20 \log_{10}\left(1 + \overline{\frac{\Delta|H|}{|H|}}\right) \qquad 6.54$$

$\overline{\Delta L}$ for the transfer function of 6.23 is given in TABLE 6.07 and plotted in Fig. 6.07. In the PCM application two such filter sections would be used in tandem. The loss distortion given in the table must, therefore, be doubled. However, refering to the loss specifications for the PCM filter (Chapter 7), it can be seen that the loss distortion is well within the required limits.

TABLE 6.07

Average loss deviation from designed value
(3rd-order filter)

Freq.(Hz)	Loss Error (dB)
200	.016
400	.017
600	.018
800	.019
1000	.020
1200	.022
1400	.024
1600	.027
1800	.031
2000	.035
2200	.041
2400	.048
2600	.057
2800	.069
3000	.086
3200	.110
3400	.141
3600	.180

The dependence of the computed average loss deviations on the frequency reflects the frequency dependent nature of the filter sensitivity function. Although, the average coefficient error in TABLE 6.05 is relatively uniform, the resulting error in the loss characteristic exhibits a significant roll-off at the passband edge.

In Fig. 6.07 the values from TABLE 6.07 are plotted and joined by a solid line. For comparison, the measured values of loss deviations are also plotted. An estimate of the measured average is shown as a dotted line. The measured curve shows a roll-off property similar to that of the theoretically derived result.

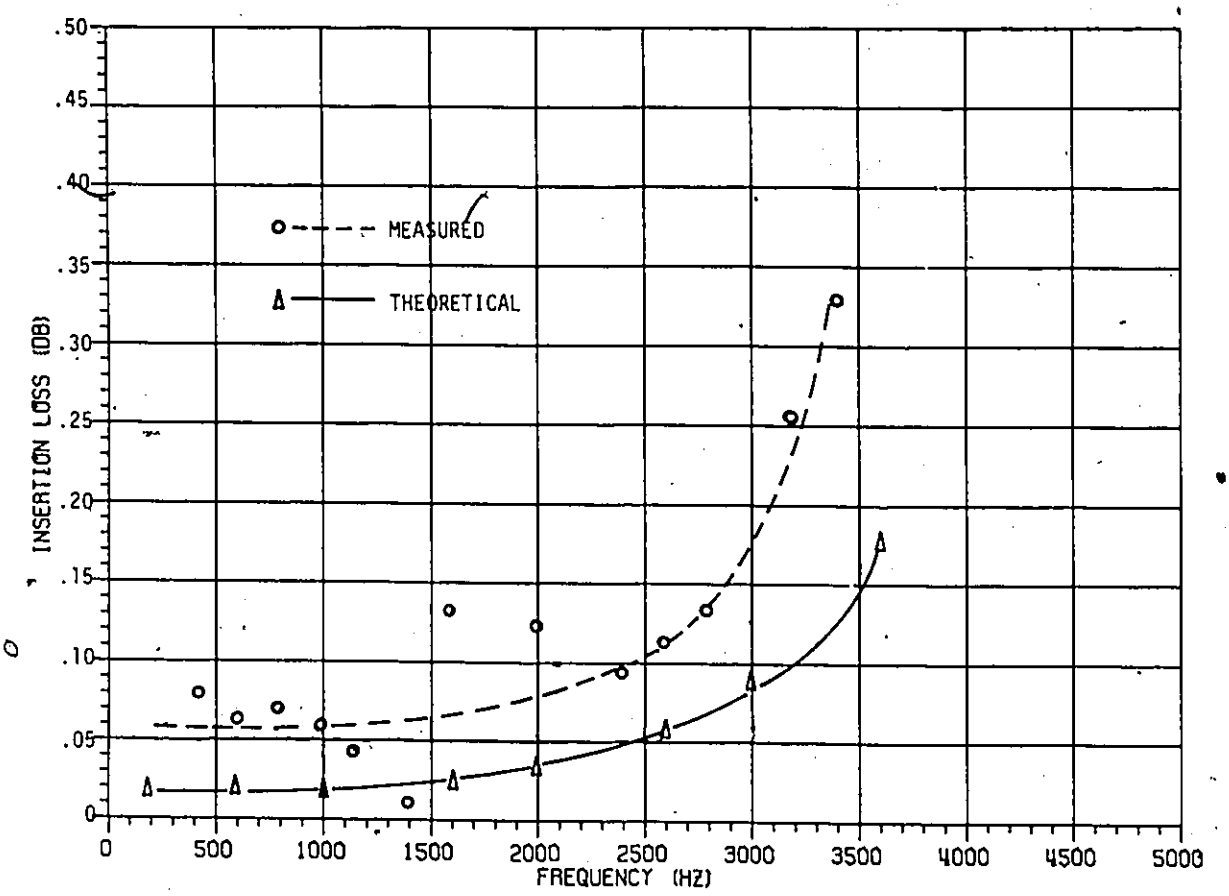


Fig. 6.07 Average loss due to product round-off (third-order).

Two major factors contributing to the differences between the two curves can be identified. To accommodate the available automated ROM burn-in facility the products stored in the test filter were truncated rather than rounded. The greater error resulting from truncation would be expected to increase the steepness of the roll-off at the passband edge. This effect is observed in Fig. 6.07. Furthermore, the measurements were done using analog test equipment with A/D and D/A circuits included as part of the test filter.

Although attempts were made to calibrate out the effects of A/D and D/A conversion, some loss mismatch is suspected. This could account for the vertical translation of the measured curve.

It might be noted that the loss specification contours of Fig. 7.01 are computed using an averaging formula. The average loss deviation of TABLE 6.07 can, therefore, be directly compared to the loss specifications.

CHAPTER 7

EXPLORATORY DEVELOPMENT OF SPDF

7.01 General

The SPDF concept was further developed in the laboratory with the aim of demonstrating technical and economic feasibility of using SPDF in such industrial applications as PCM channel banks and remote concentrator systems. To make SPDF attractive to industrial applications, the work was focused on solving the problems of size, cost and power dissiptation. The problem of size was solved by choosing filter architecture compatible with custom LSI methods. The problem of cost was solved by utilizing the speed advantage of SPDF to multiplex the filter hardware over twenty-four channels. Finally, the power dissiptation was solved by implementing a major part of the filter in low power CMDS technology.

This chapter presents the results obtained on a laboratory model of SPDF PCM filter built to operate directly on the

8-bit PCM code with a channel sample rate of 24KHz. The single channel operation of the filter was verified using linear 12-bit A/D and D/A converters. The filter's analog input-output characteristic was recorded using a spectrum analyser. The filter operation was extended to 12 channels. All relevant control waveforms were recorded to demonstrate the feasibility of extending the processing speed to the full 24-channel operation.

The overall outcome of the study identified the digital filter hardware for custom LSI specification and evaluated the filter in the DE-3 PCM system environment.

7.02 An Overview of PCM Requirements

Pulse Code Modulation (PCM) systems are widely used in telephone communications where channel filters represent a major factor in the systems cost. The terminal equipment in a PCM system is usually organized into groups of 24 voice channels multiplexed for transmission into a single digital bit stream of 1.544Mb/s. At the transmit channel bank each voice signal is band limited by a lowpass filter, then sampled, quantized and coded. At the receive channel bank the signals are decoded into Pulse Amplitude Modulation (PAM) pulses which are passed through a lowpass filter to reconstruct the analog voice signal. Given that a digital

filter with sufficiently high speed capability can be designed to process 24 voice channels, a reduction in per-channel filtering cost should be possible. To ensure a cost advantage it is essential that the increased power dissiptation of the digital filter over the passive filters is kept to a minimum.

The voice band of interest extends from 300 to 3400Hz and is band-limited by a fifth-order elliptic filter function. It was shown by the author in his previous work (16), that filter loss specifications cannot be met with sampling rates below 24KHz. But, in order to filter 24 voice signals, each sampled at 24KHz, the digital filter would be required to process one sample every 1.736×10^{-6} seconds. In Chapter 3 it was shown that a second-order SPDF section is capable of processing one signal sample every 0.420×10^{-6} seconds and, therefore, a sixth-order filter satisfying PCM specifications, could process one sample in 1.26×1^{-6} seconds. By repeatedly passing the signal through the same second-order hardware the power dissiptation may be kept to as low as 205 mWatts or about 8.5 mWatts per channel. Active filters used in newer channel banks dissipate approximately 67 mWatts per channel. Clearly, 8.5 mWatts is well within the accepted limits.

Within these economic constraints the PCM channel filter must perform its main function as a frequency selective

device whose loss characteristic must fall within tightly specified limits. The insertion loss contours defining these limits are shown in Fig. 7.01. The insertion loss characteristic of a fifth-order elliptic transfer function which meets the specifications is also shown. The insertion loss deviation due to all sources must remain small enough so that the loss characteristic remains inside the specified contours under all operating conditions.

7.03 SPDF for PCM

The insertion loss requirements of Fig. 7.01 can also be satisfied by the product of two identical third-order transfer functions. The combined sixth-order transfer function can then be realized by implementing one third-order section in hardware and processing each signal sample twice using the same section. This approach accomplishes two desired results: it reduces the total hardware and power dissiptation and uses only two stages of internal quantization.

The desired third-order transfer function is given by

$$H(z) = \frac{0.2726230+0.0808208z^{-1}+0.0808208z^{-2}+0.2726230z^{-3}}{1-0.9877751z^{-1}+0.7787396z^{-2}-0.08407628z^{-3}} \quad 7.01$$

The numerical values for the coefficients were determined by

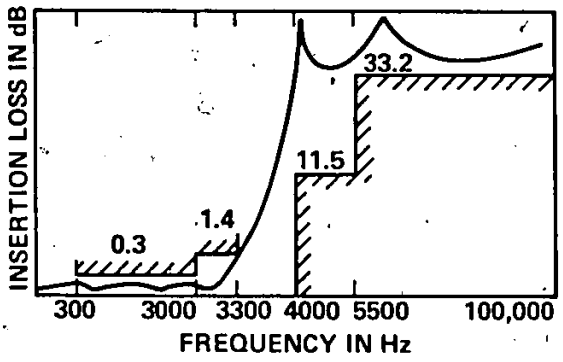


Fig. 7.01 PCM channel filter loss specifications.

applying the bilinear z-transform to the appropriately pre-distorted s-plane transfer function. The details of the derivation for Equation 7.01 are given in the aughor's previous work (16).

The sixth-order stopband insertion loss characteristic corresponding to the product of two identical transfer functions of Equation 7.01 is shown in Fig. 7.02. The insertion loss characteristics for the second and third-order Butterworth and Chebyshev transfer functions are also given. A pre-filter of this type will be required in order to suppress the multiple passbands resulting from the sampling process. The choice of the pre-filter will be determined by the amount of suppression available from the effect of sin(x)/x function and the attenuation foll-off of such system components as microphone transducers and system amplifiers. For the BNR DE-3 PCM system environment, the response (2C) in Fig. 7.02 was chosen and implemented as an active device.

The passband response of the sixth-order digital filter was measured and simulated and the results are shown in Fig. 7.03. These responses correspond to the complete sixth- order filter and include the effects of product rounding and logarithmic signal quantization.

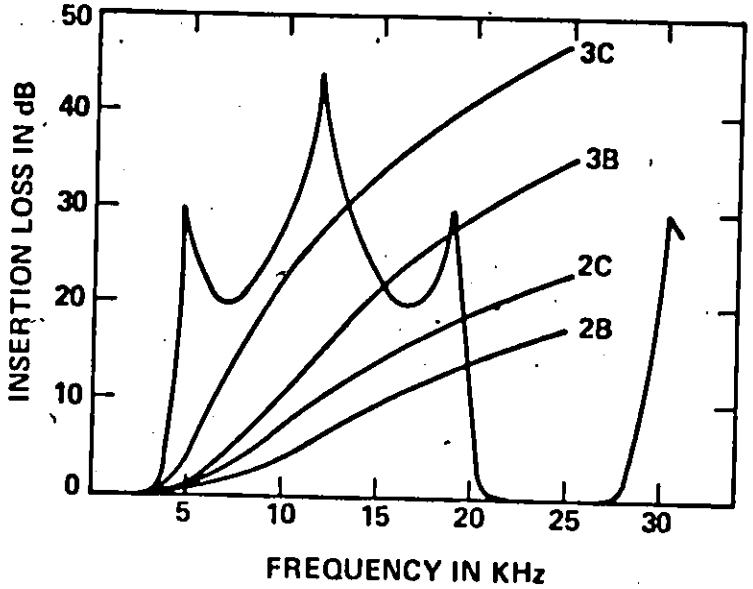


Fig. 7.02 Stopband loss characteristic of the sixth-order digital filter.

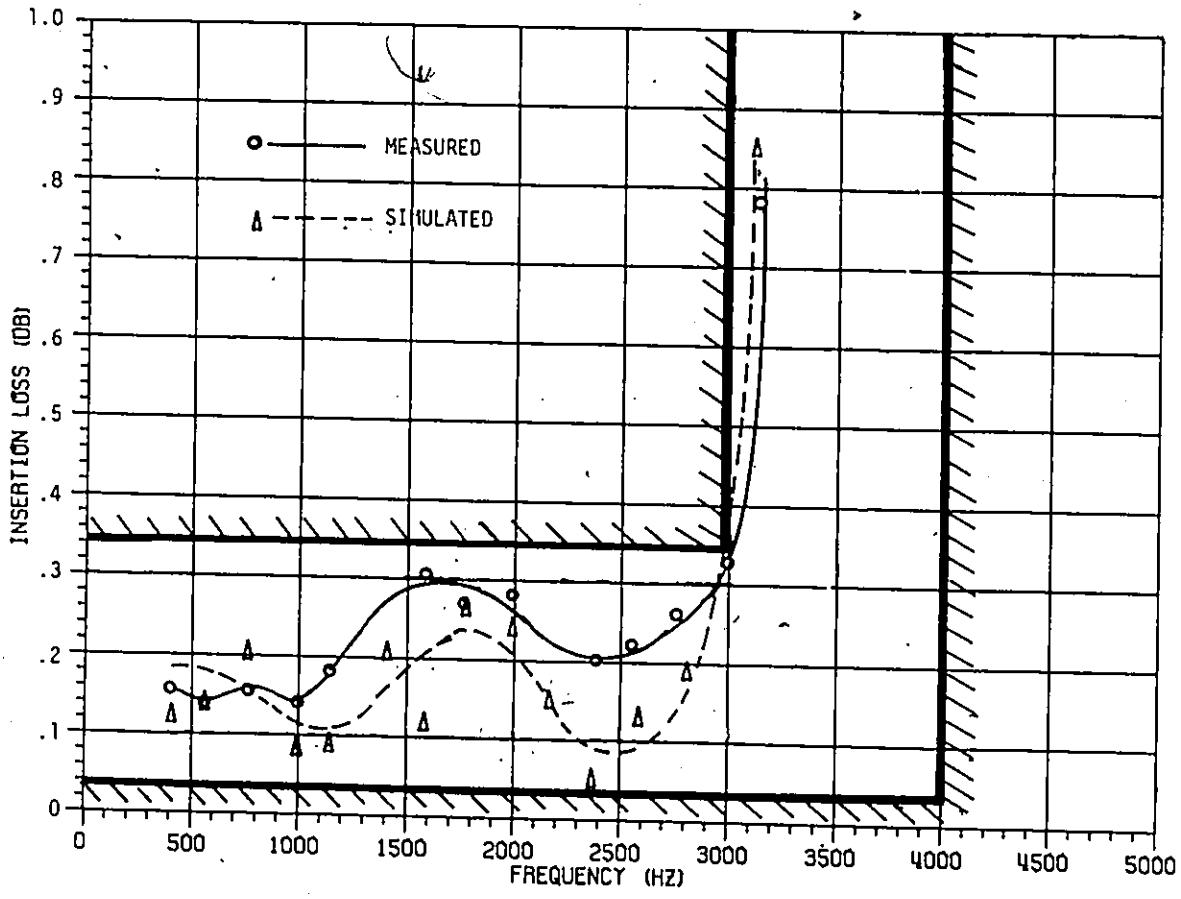


Fig. 7.03 Passband loss characteristic of the sixth-order digital filter.

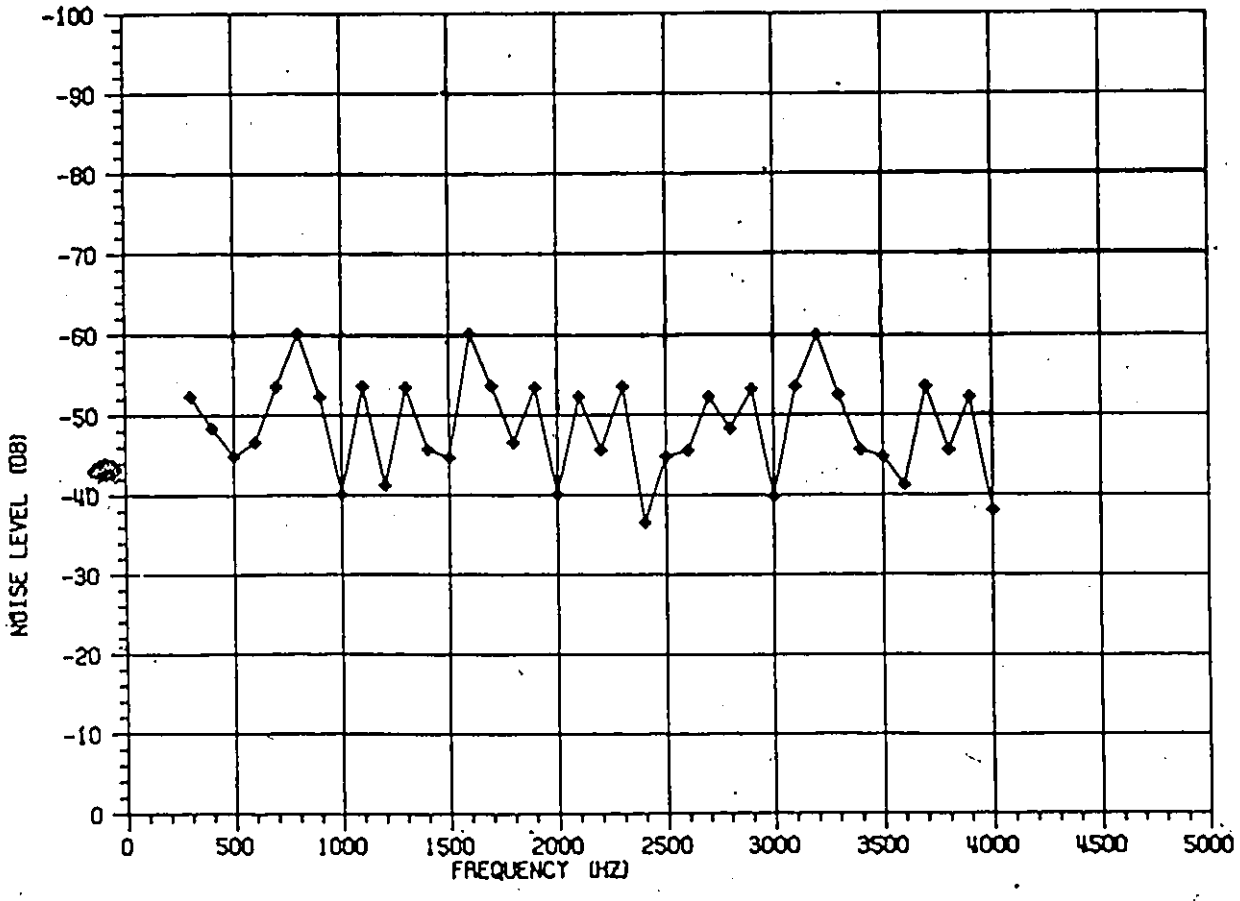


Fig. 7.04 Quantization noise due to the 128-level A-law.

The effects of product rounding on insertion loss have been discussed in detail. Next, the effect of the logarithmic signal quantization on insertion loss will be examined briefly. For this purpose simulation will be used. A sampled sinewave with 12-bit sample accuracy is applied to the input of the logarithmic quantization process. The corresponding quantized samples are then observed at the output. For each frequency of the signal, the RMS value,

V_i, and the RMS value, V_o, corresponding to the input and output samples, respectively, were computed and averaged over several signal periods. The quantization noise-to-signal level was then computed using the equation

$$QNOISE = 20\log\left|\frac{V_o - V_i}{V_i}\right| \qquad 7.02$$

QNOISE over the passband frequency range is plotted in Fig. 7.04. There exists a certain amount of correlation between the spread of the simulated data points of Fig. 7.03 and those of Fig. 7.04. The quantization noise appears to be independent of frequency as would be expected.

7.04 SPDF Implementation

The laboratory model of the third-order SPDF was built on twelve printed circuit boards mounted on BNR's VF-300 shelf. This was done to facilitate experimentation and interchange of filter sections. The multipliers were implemented in 1702AL2 PROMs whose contents could easily be altered using burn-in equipment. The remainder of the hardware was implemented in LS TTL. Certain amount of redundancy was built in for laboratory work; the redundant hardware would not appear as part of the gate count for LSI implementation. The full set of schemeatics for the laboratory model is given in Appendix 7A.

7.05 Hardware Simulation

A powerful interactive digital logic simulation program, F/LOGIC, was used to simulate and analyze the SPDF hardware implementation. The program was designed for use as a design and analysis aid capable of eliminating the need for bread-boarding. The results of the simulation and the associated circuit coding information were used as part of the input to the LSI specialist group to obtain chip design estimates.

A small segment of the circuit description is shown in Fig. 7.05. The coding for the entire filter is given in Appendix 7B. The circuit description is stored in the program as tables which define each gate or memory cell and its interconnections according to manufacturer's specifications. When a gate is switched, the changing signal is traced through the fanout list of the gate. A time base is provided which allows a realistic time scale on which all timing is controlled by the time delays, including rise and fall times, of the individual gates.

ROM elements are also defined in terms of input/output connections and access time. The ROM contents are specified as binary words in an associated data file using any desired addressing scheme. The ROM contents corresponding to the coefficient a_0=0.272623 are shown in Fig. 7.06; the complete set of ROM files are attached in Appendix 7C.

```
FILE: FILTER   LOGIC

MACRO=RR,T,T,T,T,T,T,F,F,X01,X02,X03,X04,X05,X06,/
X07,X08,INCK,T
A0,ROM,2500,A0CLK,12,X08,X07,X06,X05,X04,X03,X02,X01
MACRO=R12,A0:1,A0:2,A0:3,A0:4,A0:5,A0:6,A0:7,A0:8,/
A0:9,A0:10,A0:11,A0:12,A01Q,A02Q,A03Q,A04Q,A05Q,A06Q,/
A07Q,A08Q,A09Q,A010Q,A011Q,A012Q,A0CK,A0CL
A0CL,BUF,1,A0CLR
*
MDATA,A0,A0
*
$
MACRO=RR,X01,X02,X03,X04,X05,X06,X07,X08,X11,X12,X13,X14,X15,X16,/
X17,X18,INCK,T
A1,ROM,2500,A1CLK,12,X18,X17,X16,X15,X14,X13,X12,X11
MACRO=R12,A1:1,A1:2,A1:3,A1:4,A1:5,A1:6,A1:7,A1:8,/
A1:9,A1:10,A1:11,A1:12,A110,A120,A130,A140,A150,A160,/
A170,A180,A190,A1100,A1110,A1120,A1CK,A1CL
A1CL,BUF,1,A1CLR
*
MDATA,A1,A1
*
$
MACRO=RR,X11,X12,X13,X14,X15,X16,X17,X18,X21,X22,X23,X24,X25,X26,/
X27,X28,INCK,T
A2,ROM,2500,A2CLK,12,X28,X27,X26,X25,X24,X23,X22,X21
MACRO=R12,A2:1,A2:2,A2:3,A2:4,A2:5,A2:6,A2:7,A2:8,/
A2:9,A2:10,A2:11,A2:12,A210,A220,A230,A240,A250,A260,/
A270,A280,A290,A2100,A2110,A2120,A2CK,A2CL
A2CL,BUF,1,A2CLR
*
MDATA,A2,A2
*
$
MACRO=RR,X21,X22,X23,X24,X25,X26,X27,X28,X31,X32,X33,X34,X35,X36,/
X37,X38,INCK,T
A3,ROM,2500,A3CLK,12,X38,X37,X36,X35,X34,X33,X32,X31
MACRO=R12,A3:1,A3:2,A3:3,A3:4,A3:5,A3:6,A3:7,A3:8,/
A3:9,A3:10,A3:11,A3:12,A310,A320,A330,A340,A350,A360,/
A370,A380,A390,A3100,A3110,A3120,A3CK,A3CL
A3CL,BUF,1,A3CLR
*
MDATA,A3,A3
*
$
MACRO=RR,ORA,ORB,ORC,ORD,ORE,ORF,ORG,ORH,Y11,Y12,Y13,Y14,Y15,Y16,/
Y17,Y18,INCK,T
B1,ROM,2500,B1CLK,12,Y18,Y17,Y16,Y15,Y14,Y13,Y12,Y11
MACRO=R12,B1:1,B1:2,B1:3,B1:4,B1:5,B1:6,B1:7,B1:8,/
B1:9,B1:10,B1:11,B1:12,B110,B120,B130,B140,B150,B160,/
B170,B180,B190,B1100,B1110,B1120,B1CK,B1CL
B1CL,BUF,1,B1CLR
*
MDATA,B1,B1
*
$
```

Fig. 7.05 A segment of filter description code for F/LOGIC.

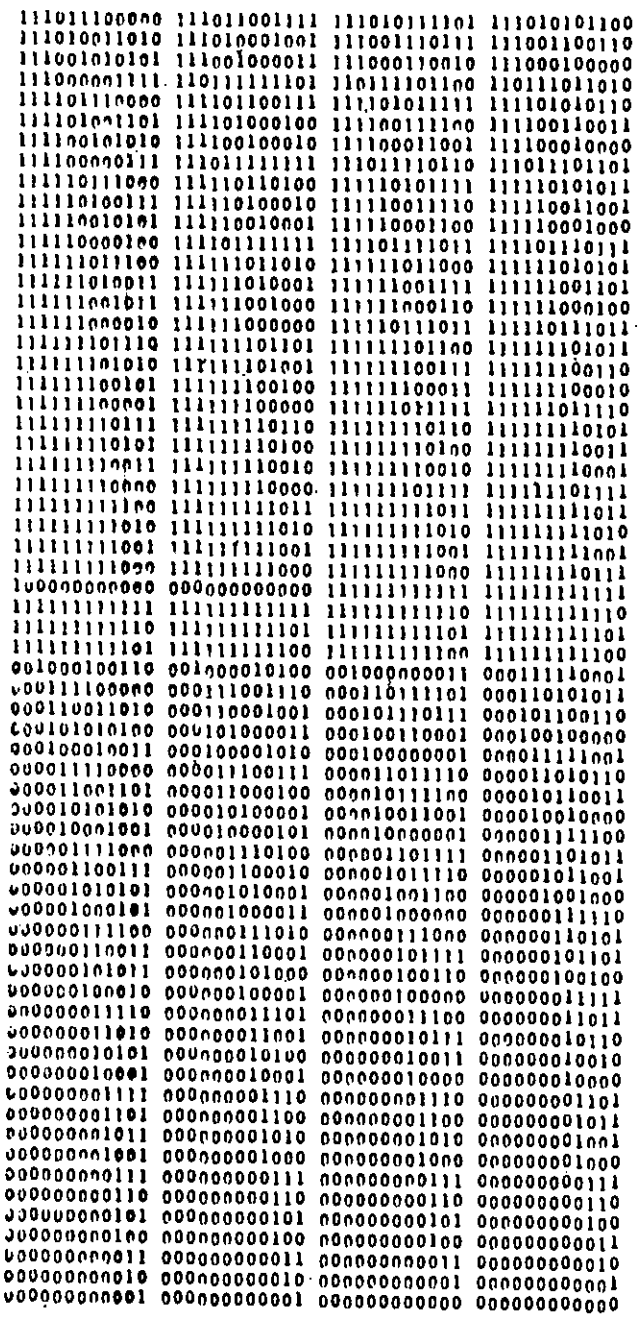


Fig. 7.06 ROM contents for the coefficient a_0=0.272623.

This simulation technique was used to complement the testing work done in the laboratory. To verify the proper operation of the filter, binary-coded sinusoidal signals were applied to the inputs of the simulation and laboratory models and the respective outputs were compared. The 8-bit PCM code corresponding to the 1500-Hz sinewave, sampled at 24KHz, and peak amplitude of 1007.5 (see TABLE 4.02) is given in Fig. 7.07. the bits of each sample were applied to the input terminals X08,X07,X06,X05,X04,X03,X02,X01 of the simulation model (see Appendix 7B), where 08 represents the most significant bit and 01 the least significant bit. The resulting 12-bit signal appearing at the output terminals Z12,Z11,Z10,...,Z2,Z1 of the adder was printed out as shown in Fig. 7.08 (and Appendix 7D). It can be seen that stabilized output bits become available every 19.6 microseconds. This value was used to accommodate the conversion equipment available in the laboratory. In the intervals between stable outputs the output levels are indeterminate due to rise and fall times of the circuit elements.

The procedure was duplicated using the laboratory test setup of Fig. 7.09. The adder output was displayed on the HP1600A Logic State Analyzer and recorded as shown in Fig. 7.10. The small differences between the measured and the simulated results are explained by the slightly different ROM contents in the two cases. The simulation model was based on rounded

MSB

0 1 1 0 1 1 1 1

0 1 1 0 1 1 0 1

0 1 1 0 0 1 1 0

0 1 0 1 1 0 0 0

0 0 0 0 0 0 0 0

1 1 0 1 0 1 1 1

1 1 1 0 1 0 0 1

1 1 1 0 0 0 1 0

1 1 1 0 0 0 0 0

1 1 1 0 0 0 1 0

1 1 1 0 1 0 0 1

1 1 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 1 0 1 1 0 0 0

0 1 1 0 0 1 1 0

0 1 1 0 1 1 0 1

Fig. 7.07 A sequence of PCM samples representing a sinewave of 50% peak amplitude.

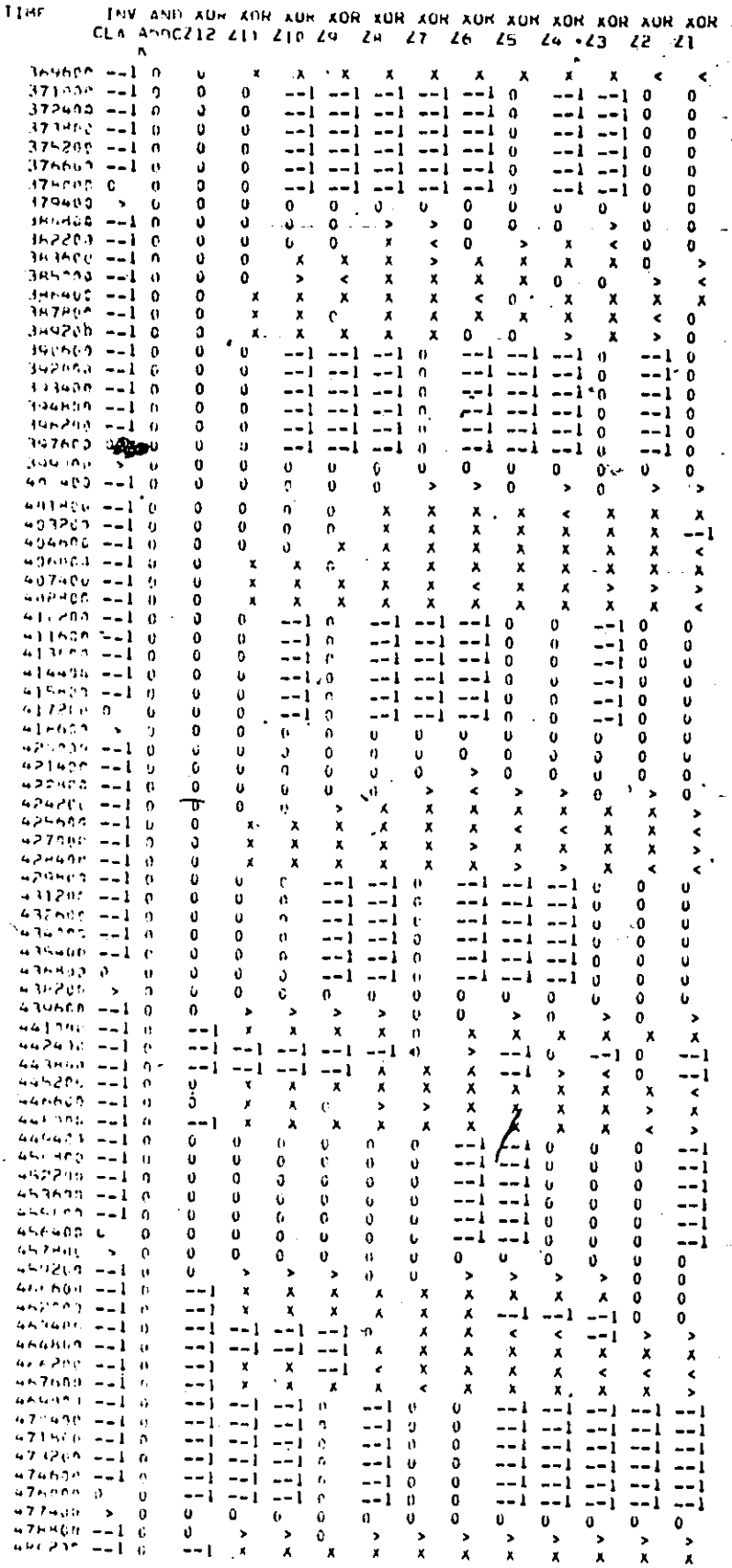


Fig. 7.08 F/LOGIC simulation of the adder output.

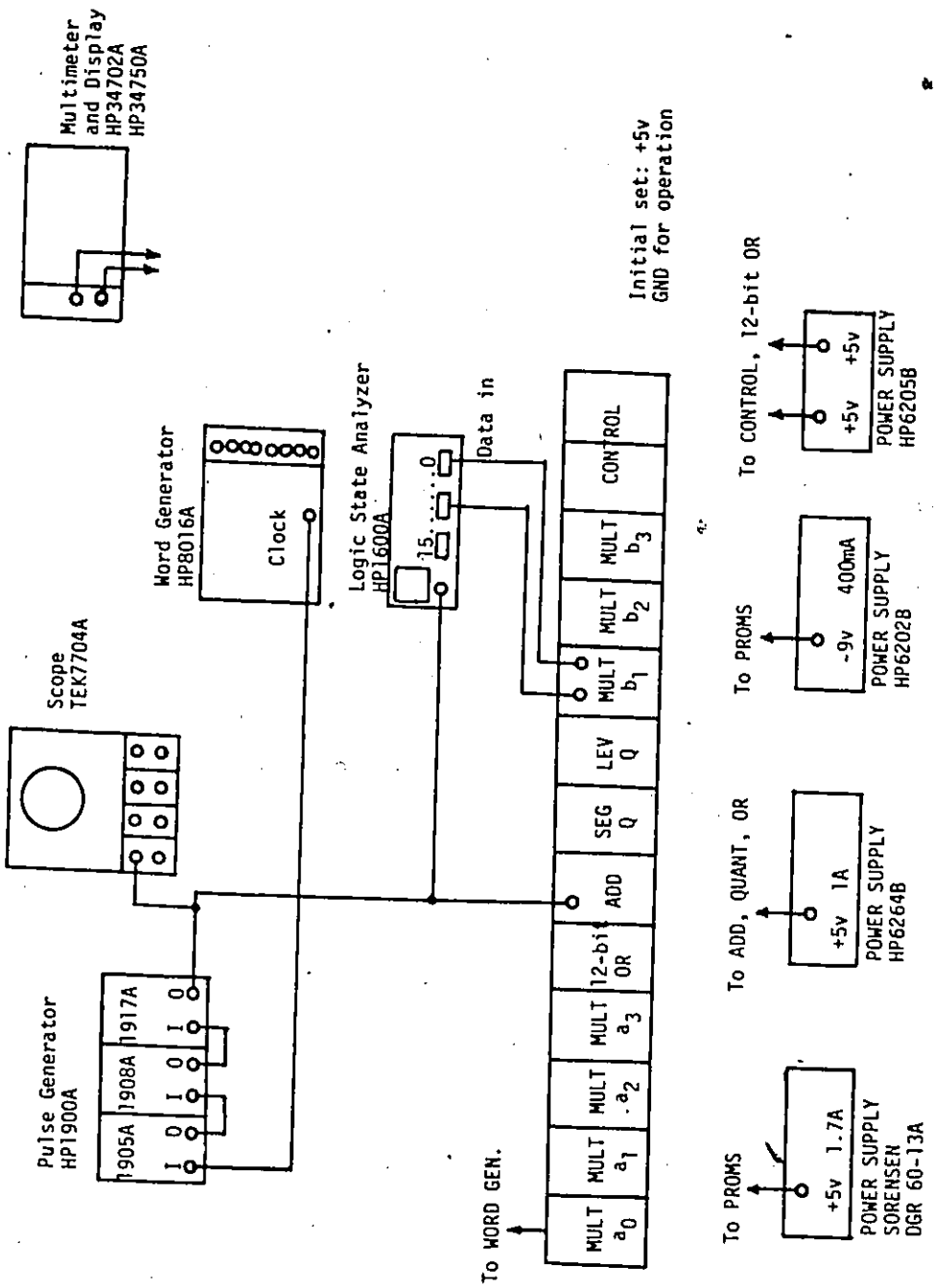


Fig.7.09 Laboratory instrumentation for filter measurements.

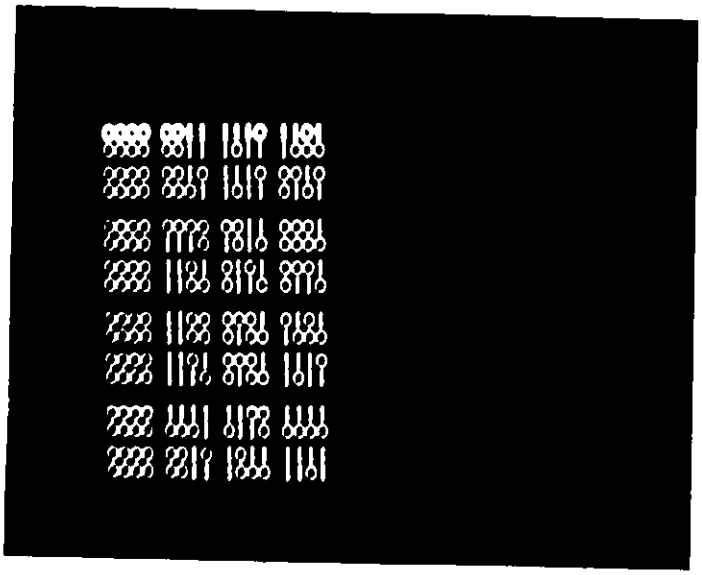


Fig. 7.10 Measured 12-bit signal at the adder output.

products while the hardware model used truncated products.

The signal of Fig. 7.10 was then displayed as a function of time using the TEK7704A oscilloscope. With the time scale set to 20 microseconds per division to correspond to the simulation scale of 19.6 microseconds between samples, the signal was recorded in Fig. 7.11. The logarithmic spacing of the sample amplitudes, resulting from the quantization process, can also be observed.

7.06 Transmission Measurements

Analog-to-analog operation of the digital filter was achieved using INTECH A-856-16 A/D converter and two MOTOROLA MC1406L six-bit D/A converters in each channel. Fig. 7.12 shows some of the control pulses required to operate the filter. The wide pulses represent the memory read intervals for three of the seven ROMs, during which ROM outputs are available. The seven narrow pulses initiate the seven additions required to complete the calculation of the output.

The insertion loss response of the filter was displayed using HP3571A Tracking Spectrum Analyzer and the HP3330B Automatic Synthesizer. The response is shown in Fig. 7.13. The notch that extends below -50dB corresponds to the half-sampling point which in this case is 1.2KHz. The

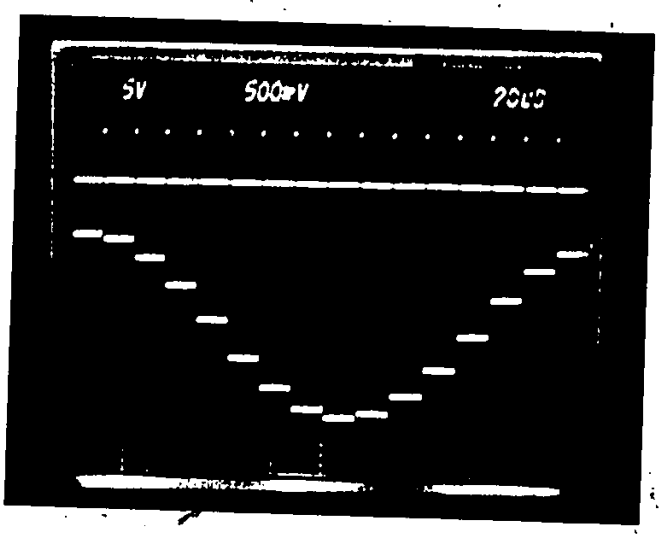


Fig. 7.11 The output of the filter as a function of time.

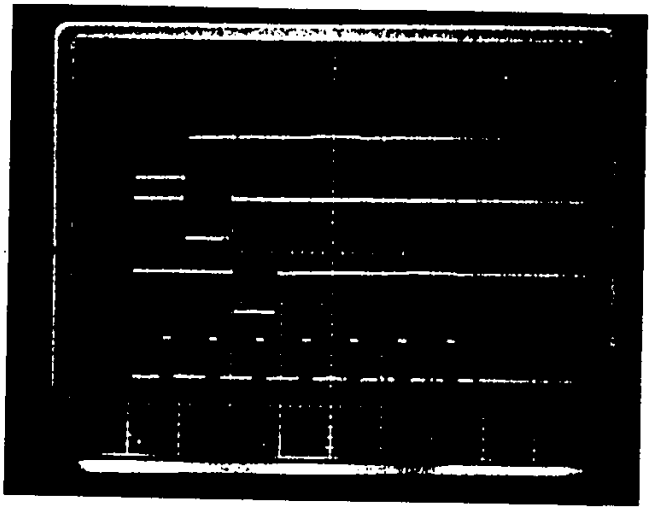


Fig. 7.12 Control pulses required to operate the filter.

SCALE

Vertical	5V/Div
Horizontal	1μs/Div

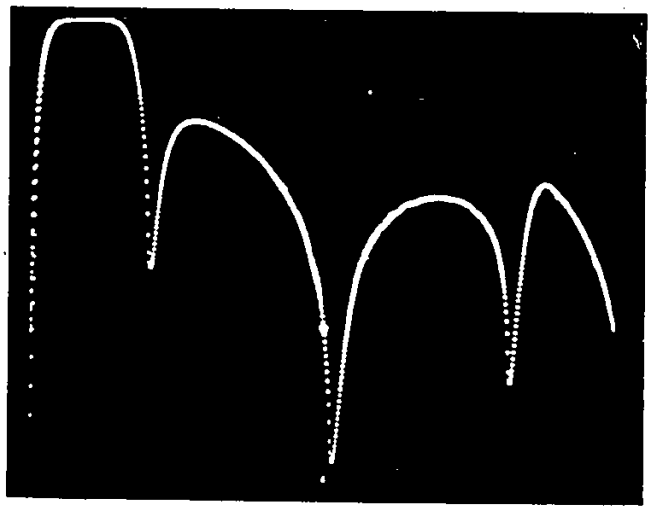


Fig. 7.13 Insertion loss response of the filter.

SCALE

Vertical	5dB/Div
Horizontal	2.3KHz/Div

smaller notch, located at 4.62KHz represents the finite attenuation pole of the third-order transfer function. The computed amplitude response over the same frequency band is plotted in Fig. 7.14. There exists an excellent correlation between the paper design and measured results. TABLE 7.01 shows several insertion loss measurements taken at specific frequency points and compared to the computed values. The computed values were based on impulse sampling and do not take into account the sin(x)/x roll-off in the frequency domain. Also, product rounding was used in computed data while product truncation was used in measurements.

7.07 Summary

Sufficient data was obtained in the exploratory development stage to establish technical feasibility of SPDF and to determine the circuit complexity and interconnection data for LSI study.

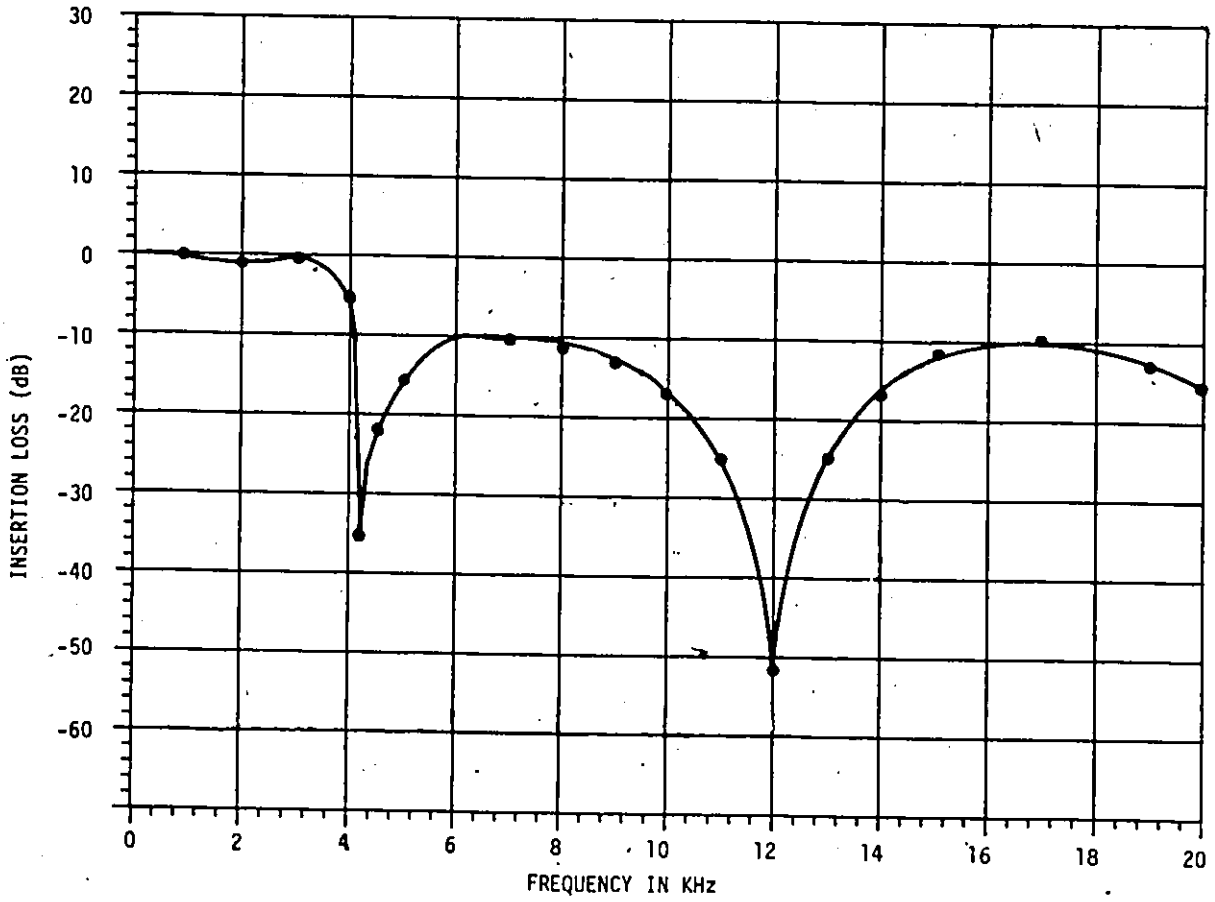


Fig. 7.14 Computed loss response for the 3rd order SPDF.

TABLE 7.01

Comparison of measured and computed results.

	24-KHz SAMPLING RATE		
$\frac{F}{Fs}$ RATIO	FREQUENCY IN HZ	LOSS IN dB	
		MEASURED	COMPUTED
0.0625	1500	0	0.093
0.125	3000	0.28	0.057
0.1875	4500	21.8	22.5
0.25	6000	11.97	10.7

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

1.01 Summary and Conclusions.

The results of this work indicate that the proposed SPDF filter implementation offers the highest speed/power ratio of the major conventional implementations using multipliers. At the same time, SPDF signal processing characteristics remain compatible with the high quality requirements of the voice channel filters. Specifically, SPDF is capable of satisfying the PCM filtering requirements for a 24-channel system at a cost lower than that of the currently used methods.

In Chapter 2, three main conventional multipliers were examined and speed/power ratios for second-order filter sections based on each of these multipliers were estimated. In addition, a second-order section based on the special technique due to Peled and Liu, was also examined. It turned

out that the filter based on the array multiplier (parallel arithmetic) had the highest speed/power ratio (0.710×10^6 sec/sample/Watt) of the four cases.

In Chapter 3 the structure and main characteristics of the stored-product technique with internal logarithmic quantization were presented. It was shown that a second-order SPDF section can be integrated onto a single LSI chip capable of operating with a speed/power ratio of 11×10^6 sec/sample/Watt which corresponds to an improvement by a factor of 16 to 54 times over conventional implementations.

In Chapter 4, A-law quantization algorithm, defined for use as part of the SPDF architecture, was described. This algorithm can be replaced by any suitable compression technique making SPDF widely applicable. Computational expressions relevant to the SPDF operation and ROM programming were derived.

In Chapter 5 the important question of stability and limit cycles, resulting from the introduction of the logarithmic signal quantization, was considered. It was shown that limit cycles, resulting from using the quantizer as part of SPDF architecture, are only possible for coefficients falling inside very narrow strips along the boundaries of the stability triangle in the coefficient plane. It was also shown that useful filters can be designed for which the coefficients fall outside these strips.

Chapter 6 was devoted to the study of signal noise contributions due to internal processes of the SPDF. It was shown that product rounding is the only noise source in SPDF in addition to the logarithmic quantizer. By assuming sufficiently accurate coefficient representation, an equivalence between product rounding and coefficient rounding can be established. Such equivalence was used to compute the average insertion loss deviation due to product rounding in a third-order section. The effects of noise on the loss characteristic were within the design limits of PCM system requirements.

In Chapter 7 the results of computer simulation of hardware and the results of laboratory measurements were presented. The work was aimed at identifying the PCM filter hardware for LSI implementation. Although the full sixth-order, 24-channel operation was not achieved due to budgetary constraints, sufficient confidence was established that such operation is achievable without structural changes. Furthermore, sufficient design detail was made available for immediate LSI implementation of the PCM filter.

1.02 Topics for Further Study

On the topic of limit cycles there are two potential areas for further study. One area deals with the important topic

of limit cycle suppression. Suppression of limit cycles in conventional filters relies on algorithms which treat all products equally over the entire signal dynamic range. Random rounding and truncation are often used. This reduces the signal processing speed of the digital filter by some finite amount. In SPDF implementations, arbitrary algorithms may be applied independently to the individual products prior to storing them in ROM. No real time signal processing speed or power are sacrificed in this operation. Research may be done to develop and test alogorithms to perturb signal-coefficient products for the purpose of suppression of limit cycles. There have been no publications on such an approach by current investigators.

Recursive coefficients are often chosen in order to produce limit cycles intentionally. Such techniques are used in digital oscillators to generate new waveforms. SPDF architecture with ditital logarithmic quantization in the recursive loop generates limit cycles which are not possible with the ususal arithmetic rounding of coefficients and signals. A research activity may be carried out to investigate the nature of the waveforms that can be generated by this mthod.

A-law signal compression technique was found to be particularly useful for voice communications and PCM application. There exist other signal compression techniques that may be useful in other signal processing applications such as

radar or digital waveform generation. Research in this area may lead to some useful results.

REFERENCES

(1) A. Antoniou, "Digital Filters: Analysis and design", McGraw-Hill Book Company, 1979.

(2) P. Benedek, "Digital filter for DMS-1", Dept. 1E81 Memorandum, Bell-Northern Research, June 1978.

(3) J.A.C. Bingham, "Specification and design of channel filters for PCM", IEEE Conference on Communications, pp. 239-244, 1968.

(4) G.A. Blaauw,"Digital System Implementation", Prentice-Hall, Inc., New Jersey, 1976.

(5) M.L. Blostein, "Sensitivity analysis of linear systems in the frequency domain", Short Course Notes on Computer Aided Network Analysis and Design, IEEE International Convention and Exposition, New York, N.Y., March 1972.

(6) L.T. Bruton, "Digital filter synthesis, design and implementation", Technical Report TR-1E80-76-07, Bell-Northern Research, February, 1976.

(7) D. Dubois, O. Monkewich and W. Steenaart, "Minimum Coefficient Quantization Effects in Stored Product Digital Filter Response", submitted for publication, IEEE Transac. CAS.

(8) D. Dubois and W. Steenaart, "High Speed Stored Product Digital Filtering", submitted for publication, IEEE Transac. CAS.

(9) M.I. Elmasry, "LSI realization of a digital signal processor for PCM channels", Dept. 5E10 Report, Bell-Northern Research, December 1976.

(10) L.B. Jackson "An Analysis of Limit Cycles Due to Multiplication Rounding in Recursive Digital (Sub)filters" Proc. 7th Annual Allerton Conference on Circuit System Theory, pp. 69-78, 1969.

(11) J.B. Knowles and E.M. Olcayto, "Coefficient accuracy and digital filter response", IEEE Trans. on Circuit Theory, Vol. CT-15, pp. 31-44, March 1978.

(12) A. Kundig, "Digital filtering in PCM telephone systems", IEEE Trans. Audio and Electroacoustics, Vol. AU-18, No. 4, December 1970.

(13) B. LeBoss, "16-K C-MOS static RAM is fast", Electronics, McGraw-Hill, pp. 169-170, June 2, 1981.

(14) G.W. McIver, R.W. Miller and T.G. O'Shaughnessy, "A monolitic 16x16 digital multiplier", Digest of Technical Papers, IEEE International Solid-State Circuits Conference, pp. 54-55, 1974.

(15) J.R. Mick and J. Springer, "Single-chip multiplier expands digital role in signal processing", Electronics, McGrawHill, pp. 103-108, May 1974.

(16) O. Monkewich, "A digital PCM filter", M.A.Sc. Thesis, University of Ottawa, 1974.

(17) O. Monkewich and W. Steenaart, "Companding for digital filters", IEEE Proc. Int. Symp. on Circuits and Systems, pp. 68-71, 1975.

(18) O. Monkewich and W. Steenaart, "Stored product digital filtering with nonlinear quantization", IEEE Proc. Int. Symp. on Circuits and Systems, pp. 157-160, 1976.

(19) O. Monkewich "A digital filter for PCM"; Dept. 1E80 Report, Case IK7609 Bell-Northern Research, July 21, 1976.

(20) O. Monkewich, "24-Channel Digital PCM Filter", Technical Report TR-1E81-77-25, Bell-Northern Research, June, 1977.

(21) O. Monkewich, "Digital filters", Canadian Patent No. 1050619, March 13, 1979.

(22) O. Monkewich and W. Steenaart, "Deadband effects and limit cycles in Stored-Product Digital Filters", IEEE Proc. Int. Symp. on Circuits and Systems, pp. 813-816, 1981.

(23) A. Peled and B. Liu, "A new hardware realization of digital filters", IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-22, pp. 456-462, Dec. 1974.

(24) W. Steenaart, D. Dubois and O. Monkewich, "Stored-Product Digital Filtering, Structures, Potential and Applications", European Circuit Theorty and Design, The Hague, pp. 118-126, August 25-27, 1981.

(25) TRW LSI Products, "16x16-bit multiplication in only 100 nsec", Electronic Engineering Times, July 21, pp. 47, 1980.

(26) Technical Staff of Bell Telephone Laboratories, "Transmission Systems for Communications", Fourth Edition, p. 578, 1970.

APPENDICES

APPENDIX 6A

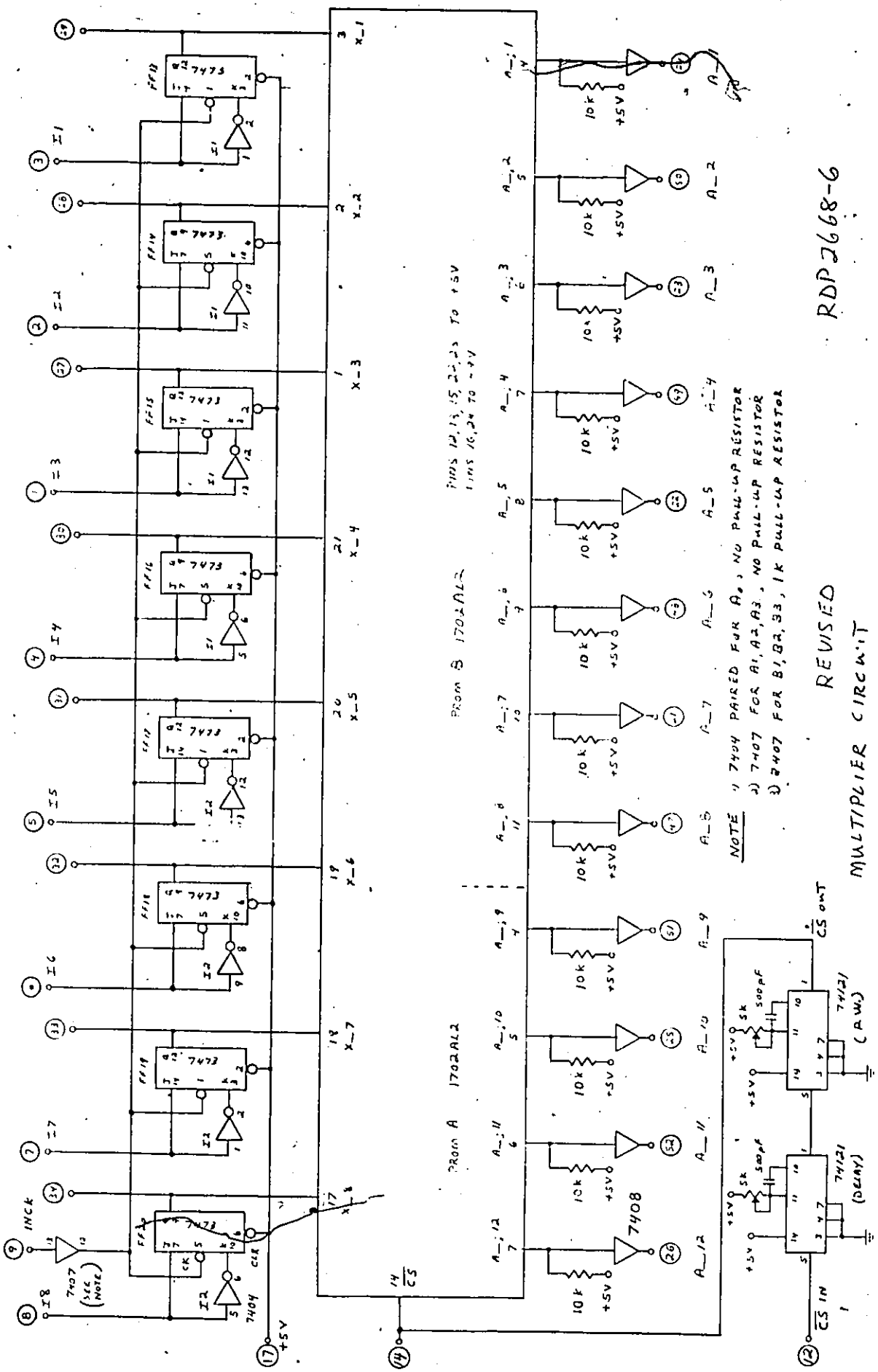
This appendix lists the individual probabilities for each of the quantized sample amplitudes corresponding to a sampled voice signal. The probabilities were computed using the Laplacian probability density function.

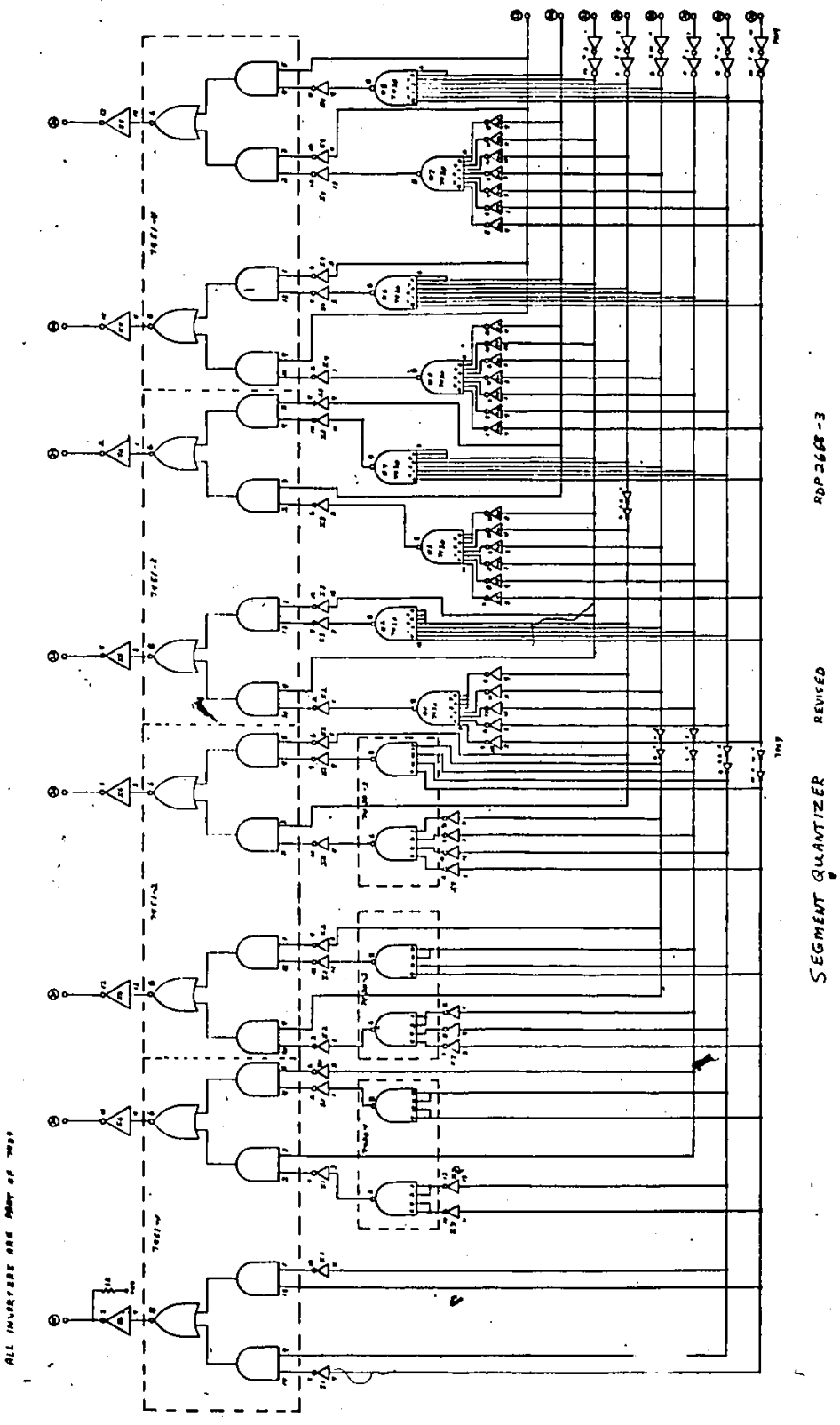
SIGMA=683

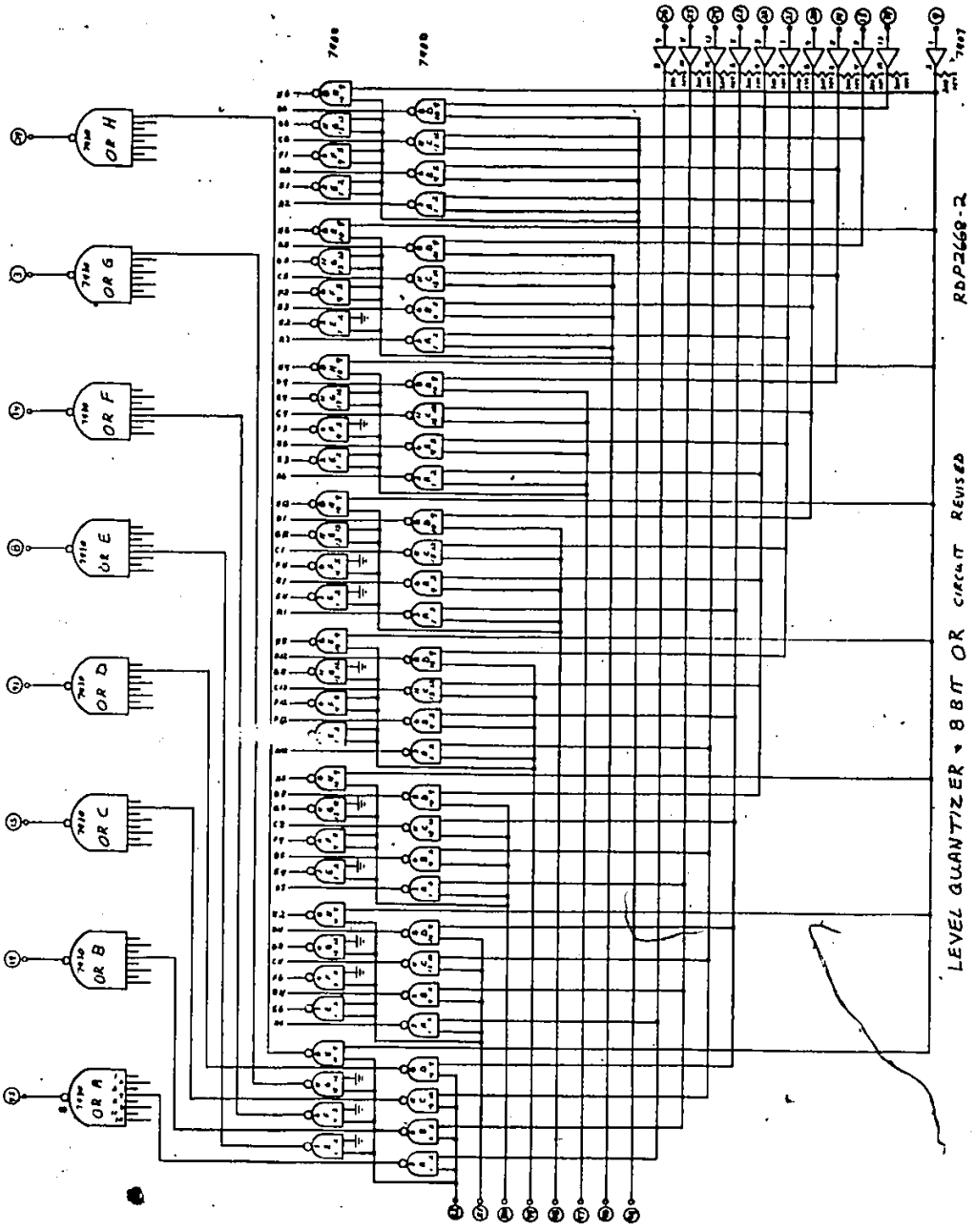
SAMPLE	PROBABILITY	SAMPLE	PROBABILITY	SAMPLE	PROBABILITY
0	1.03539553E-03	65.5	3.61597072E-03	527.5	.0111158161
1	1.03315412E-03	69.5	3.58614563E-03	559.5	.0104031647
2	1.03101724E-03	73.5	3.55656658E-03	591.5	9.73620249E-03
3	1.02888446E-03	77.5	3.52723137E-03	623.5	9.11200019E-03
4	1.02675634E-03	81.5	3.49813821E-03	655.5	8.52781643E-03
5	1.0246325E-03	85.5	3.46928509E-03	687.5	7.9810855E-03
6	1.02251315E-03	89.5	3.44066982E-03	719.5	7.46940632E-03
7	1.02039819E-03	93.5	3.4122906E-03	751.5	6.99053167E-03
8	1.01828744E-03	97.5	3.38414552E-03	783.5	6.54235837E-03
9	1.01618125E-03	101.5	3.35623244E-03	815.5	6.12291813E-03
10	1.01407933E-03	105.5	3.32854982E-03	847.5	5.73036879E-03
11	1.01198171E-03	109.5	3.30109543E-03	879.5	5.36298638E-03
12	1.00988852E-03	113.5	3.27386741E-03	911.5	5.01915743E-03
13	1.00779961E-03	117.5	3.24686405E-03	943.5	4.69737184E-03
14	1.00571495E-03	121.5	3.22008333E-03	975.5	4.3962164E-03
15	1.00363468E-03	125.5	3.1952367E-03	1007.5	4.11436848E-03
16	1.00155878E-03	131.5	6.30824253E-03	1055.5	7.45431346E-03
17	9.99487208E-04	139.5	6.20460884E-03	1119.5	6.5291387E-03
18	9.97419661E-04	147.5	6.10267773E-03	1183.5	5.71878987E-03
19	9.9535654E-04	155.5	6.00242119E-03	1247.5	5.00901559E-03
20	9.93297687E-04	163.5	5.90381176E-03	1311.5	4.38733329E-03
21	9.9124317E-04	171.5	5.80682222E-03	1375.5	3.84280964E-03
22	9.89192893E-04	179.5	5.71142608E-03	1439.5	3.36586826E-03
23	9.8714675E-04	187.5	5.61759717E-03	1503.5	2.94812137E-03
24	9.85104861E-04	195.5	5.52530966E-03	1567.5	2.58222215E-03
25	9.83067258E-04	203.5	5.43453823E-03	1631.5	2.26173567E-03
26	9.81033848E-04	211.5	5.3452581E-03	1695.5	1.98102562E-03
27	9.79004606E-04	219.5	5.25744473E-03	1759.5	1.73515523E-03
28	9.76979503E-04	227.5	5.17107395E-03	1823.5	1.51980049E-03
29	9.74958729E-04	235.5	5.08612199E-03	1887.5	1.33117399E-03
30	9.72942039E-04	243.5	5.00256575E-03	1951.5	1.16595843E-03
31	9.70929649E-04	251.5	4.92038212E-03	2015.5	1.02124821E-03
32.5	1.93583848E-03	263.5	9.59959188E-03		
34.5	1.9278383E-03	279.5	9.28677309E-03		
36.5	1.91987134E-03	295.5	8.98414796E-03		
38.5	1.9119372lE-03	311.5	8.69138435E-03		
40.5	1.9040359E-03	327.5	8.40816095E-03		
42.5	1.89616721E-03	343.5	8.13416687E-03		
44.5	1.88833112E-03	359.5	7.86910131E-03		
46.5	1.88052738E-03	375.5	7.61267341E-03		
48.5	1.87275585E-03	391.5	7.36460166E-03		
50.5	1.86501655E-03	407.5	7.12461367E-03		
52.5	1.85730906E-03	423.5	6.89244612E-03		
54.5	1.84963353E-03	439.5	6.66784415E-03		
56.5	1.84198974E-03	455.5	6.4505613E-03		
58.5	1.83437746E-03	471.5	6.24035886E-03		
60.5	1.82679675E-03	487.5	6.03700624E-03		
62.5	1.81924713E-03	503.5	5.84028026E-03		

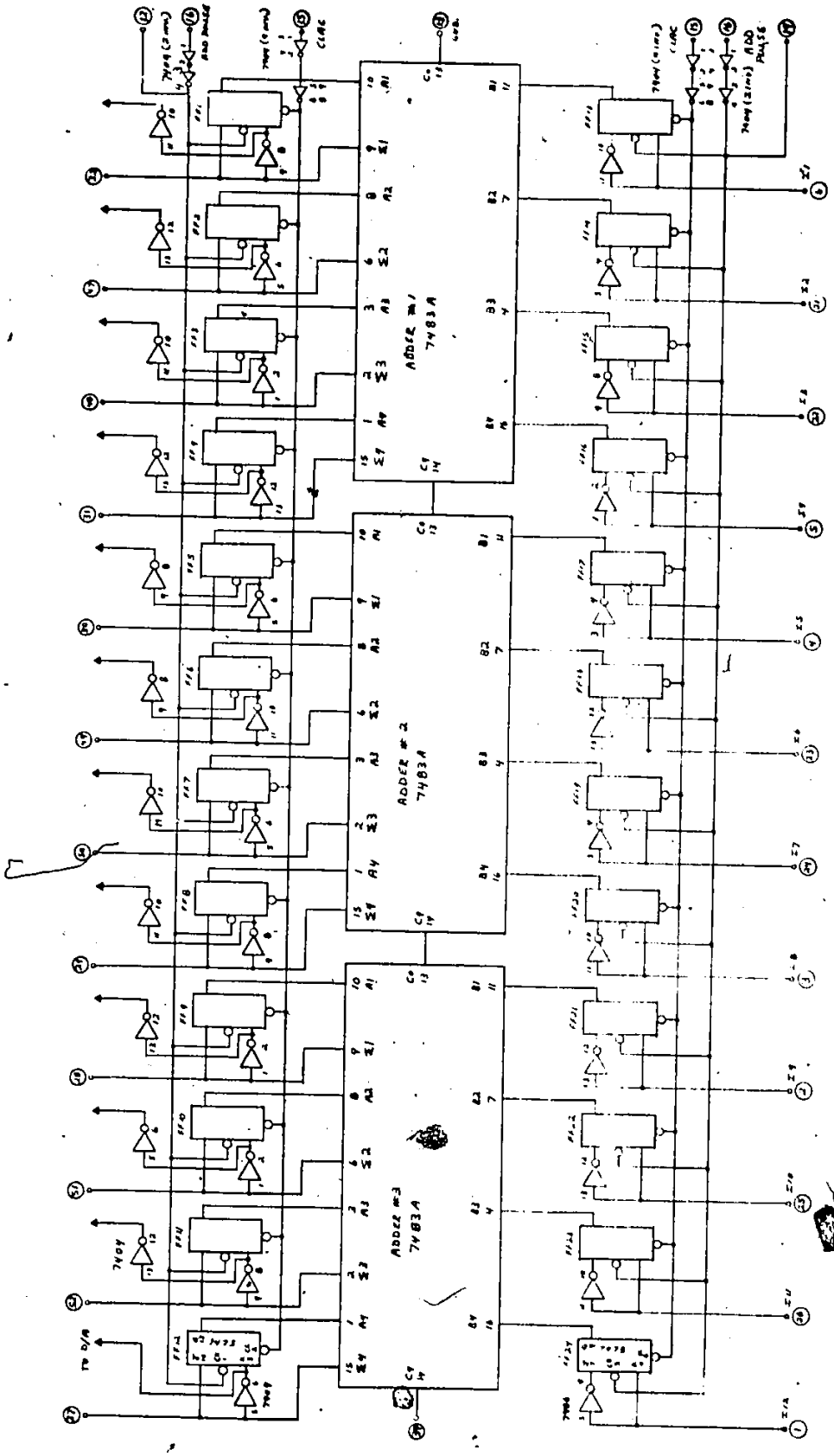
APPENDIX 7A

This appendix includes the complete set of schematic diagrams describing the hardware implementation of the third-order SPDF section.


RDP 2668-6
REVISED
MULTIPLIER CIRCUIT
NOTE
1) 7404 PAIRED FOR A_0, NO PULL-UP RESISTOR
2) 7407 FOR A1, A2, A3, NO PULL-UP RESISTOR
3) 2407 FOR B1, B2, B3, 1K PULL-UP RESISTOR
INCK
7407 (SEE NOTE)
7404
7408
7473
7474
74121
(DELAY)
(A.W.)
+5V
10k
1K
5K
500pF
From A 1702AL2
From B 1702AL2
Pins 14, 15, 25 and 26 to +5V
Pins 16, 24 to -7V
$\overline{CS}$ IN
$\overline{CS}$ OUT

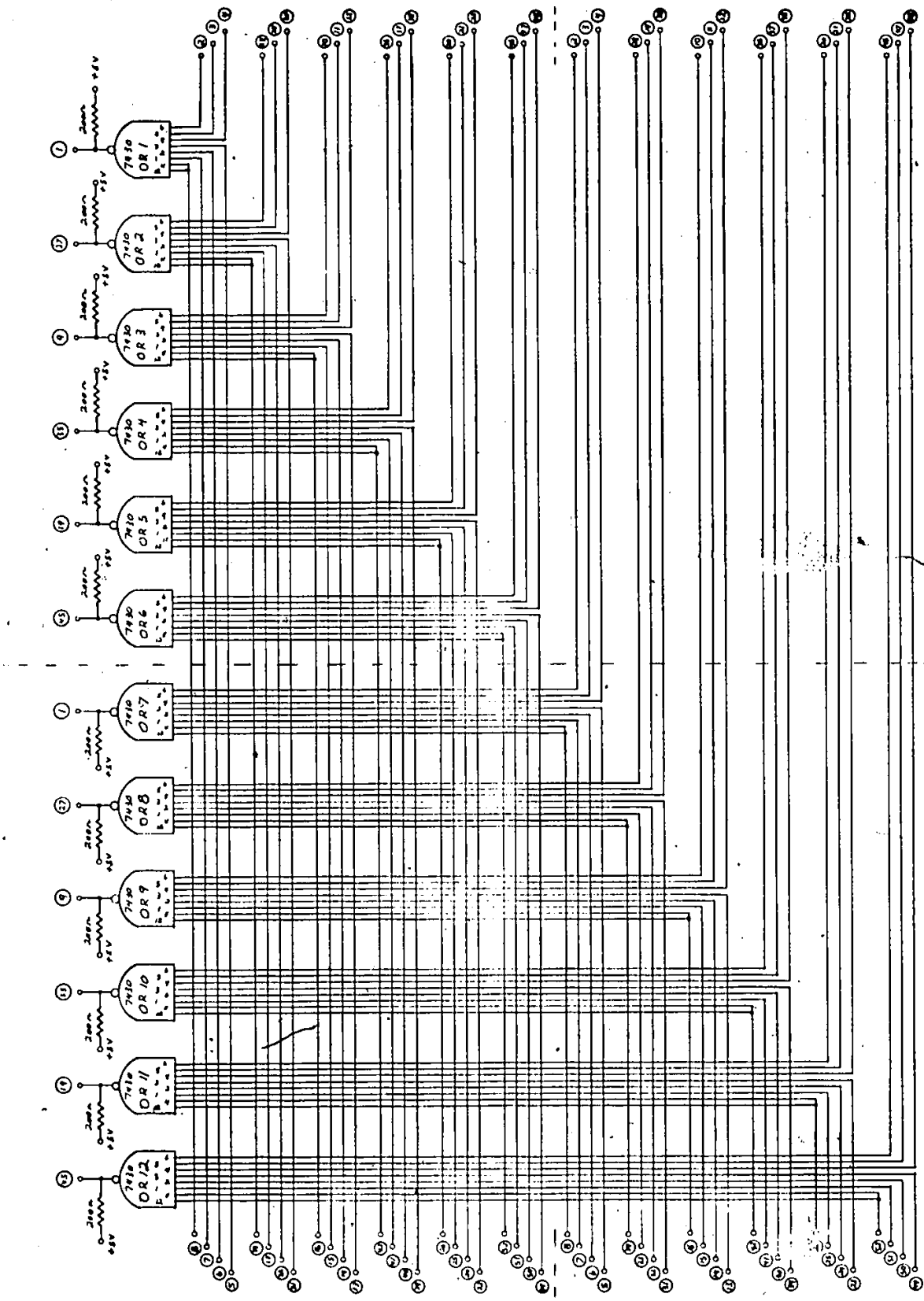

SEGMENT QUANTIZER
REVISED
RDP 26A7-3
ALL INVERTERS ARE PART OF 7451


OR H
OR G
OR F
OR E
OR D
OR C
OR B
OR A
RDP2668-2
LEVEL QUANTIZER - 8 BIT OR CIRCUIT REVISED



ADDER CIRCUIT REVISED

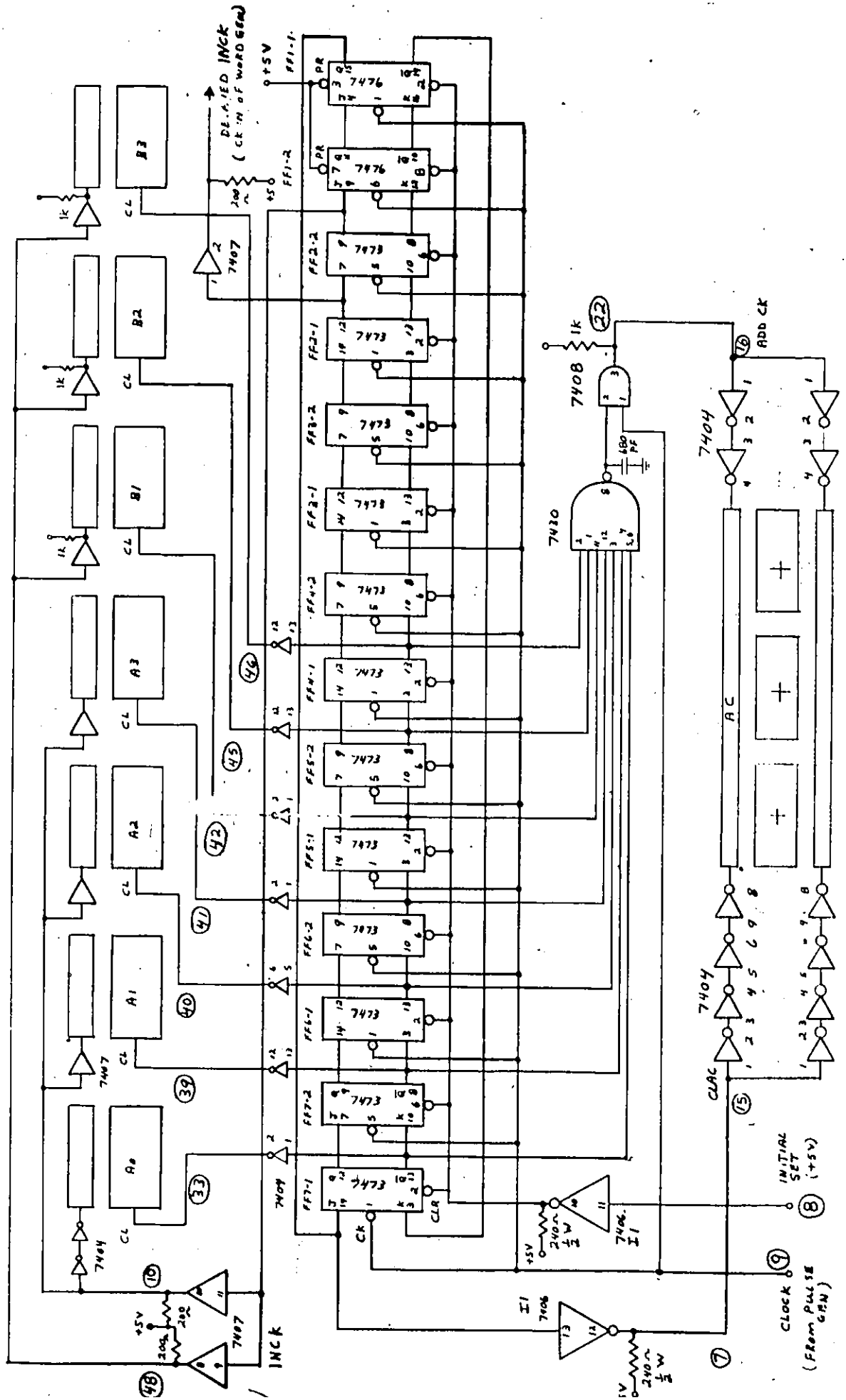
RDP26687/



12 BIT OR CIRCUIT

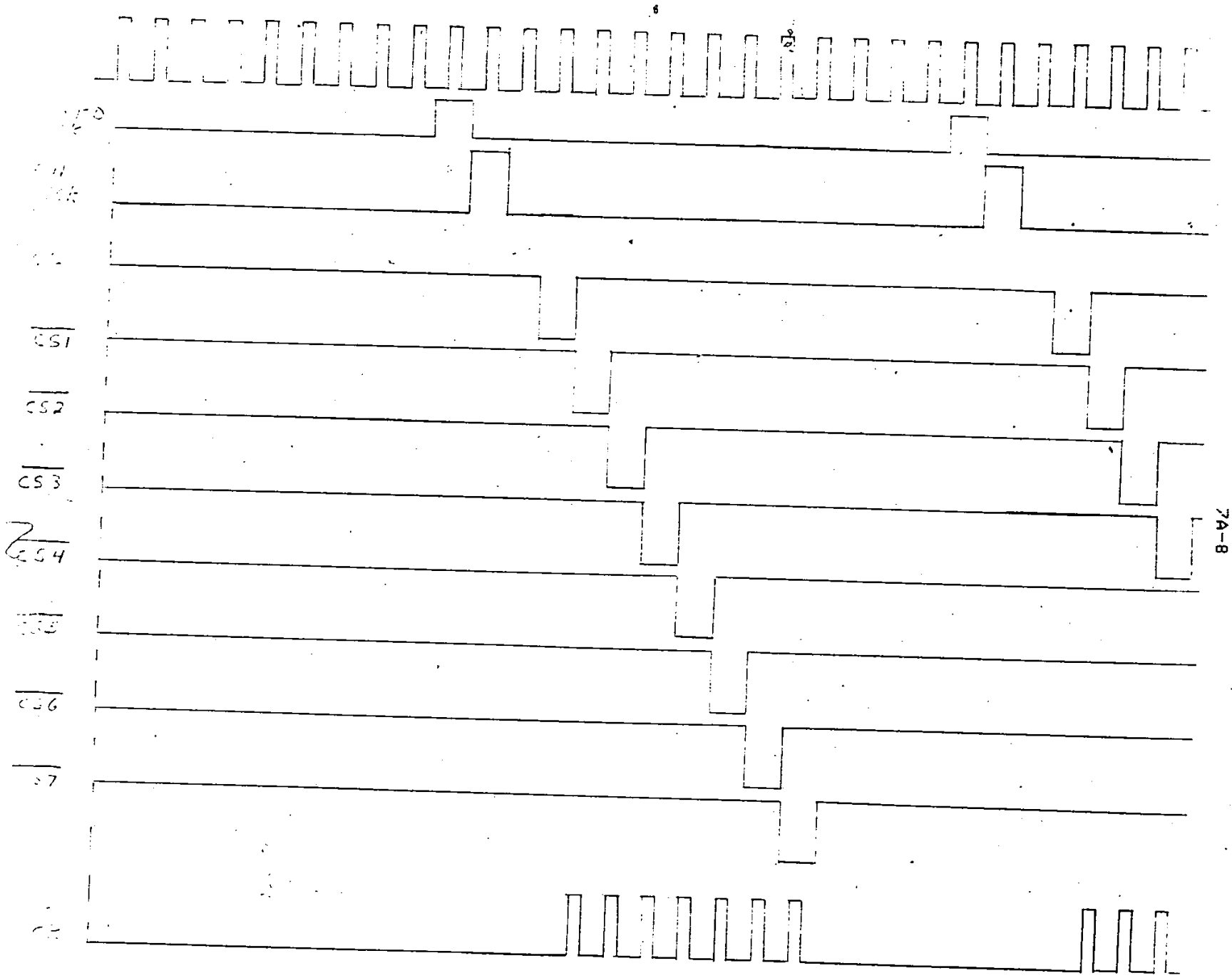
REVISED RDP 2668-4-1

RDP 2668-4-2



CONTROL CIRCUIT REVISED

RDP 2668-5


CS1
CS2
CS3
CS4
CS6
CS7

APPENDIX 7B

This appendix lists the complete F/LOGIC code describing the third-order SPDF section.

```
MACRO=R8,T,T,T,T,T,T,F,F,X01,X02,X03,X04,X05,X06,/
X07,X08,INCK,T
A0,ROM,2500,A0CLK,12,X08,X07,X06,X05,X04,X03,X02,X01
MACRO=R12,A0:1,A0:2,A0:3,A0:4,A0:5,A0:6,A0:7,A0:8,/
A0:9,A0:10,A0:11,A0:12,A01Q,A02Q,A03Q,A04Q,A05Q,A06Q,/
A07Q,A08Q,A09Q,A010Q,A011Q,A012Q,A0CK,A0CL
A0CL,BUF,1,A0CLR
*
MDATA,A0,A0
*
$
MACRO=R8,X01,X02,X03,X04,X05,X06,X07,X08,X11,X12,X13,X14,X15,X16,/
X17,X18,INCK,T
A1,ROM,2500,A1CLK,12,X18,X17,X16,X15,X14,X13,X12,X11
MACRO=R12,A1:1,A1:2,A1:3,A1:4,A1:5,A1:6,A1:7,A1:8,/
A1:9,A1:10,A1:11,A1:12,A11Q,A12Q,A13Q,A14Q,A15Q,A16Q,/
A17Q,A18Q,A19Q,A110Q,A111Q,A112Q,A1CK,A1CL
A1CL,BUF,1,A1CLR
*
MDATA,A1,A1
*
$
MACRO=R8,X11,X12,X13,X14,X15,X16,X17,X18,X21,X22,X23,X24,X25,X26,/
X27,X28,INCK,T
A2,ROM,2500,A2CLK,12,X28,X27,X26,X25,X24,X23,X22,X21
MACRO=R12,A2:1,A2:2,A2:3,A2:4,A2:5,A2:6,A2:7,A2:8,/
A2:9,A2:10,A2:11,A2:12,A21Q,A22Q,A23Q,A24Q,A25Q,A26Q,/
A27Q,A28Q,A29Q,A210Q,A211Q,A212Q,A2CK,A2CL
A2CL,BUF,1,A2CLR
*
MDATA,A2,A2
*
$
MACRO=R8,X21,X22,X23,X24,X25,X26,X27,X28,X31,X32,X33,X34,X35,X36,/
X37,X38,INCK,T
A3,ROM,2500,A3CLK,12,X38,X37,X36,X35,X34,X33,X32,X31
MACRO=R12,A3:1,A3:2,A3:3,A3:4,A3:5,A3:6,A3:7,A3:8,/
A3:9,A3:10,A3:11,A3:12,A31Q,A32Q,A33Q,A34Q,A35Q,A36Q,/
A37Q,A38Q,A39Q,A310Q,A311Q,A312Q,A3CK,A3CL
A3CL,BUF,1,A3CLR
*
MDATA,A3,A3
*
$
MACRO=R8,ORA,ORB,ORC,ORD,ORE,ORF,ORG,ORH,Y11,Y12,Y13,Y14,Y15,Y16,/
Y17,Y18,INCK,T
B1,ROM,2500,B1CLK,12,Y18,Y17,Y16,Y15,Y14,Y13,Y12,Y11
MACRO=R12,B1:1,B1:2,B1:3,B1:4,B1:5,B1:6,B1:7,B1:8,/
B1:9,B1:10,B1:11,B1:12,B11Q,B12Q,B13Q,B14Q,B15Q,B16Q,/
B17Q,B18Q,B19Q,B110Q,B111Q,B112Q,B1CK,B1CL
B1CL,BUF,1,B1CLR
*
MDATA,B1,B1
*
$
```

```
MACRO=R8,Y11,Y12,Y13,Y14,Y15,Y16,Y17,Y18,Y21,Y22,Y23,Y24,Y25,Y26,/
Y27,Y28,INCK,T
B2,ROM,2500,B2CLK,12,Y28,Y27,Y26,Y25,Y24,Y23,Y22,Y21
MACRO=R12,B2:1,B2:2,B2:3,B2:4,B2:5,B2:6,B2:7,B2:8,/
B2:9,B2:10,B2:11,B2:12,B210,B220,B230,B240,B250,B260,/
B270,B280,B290,B2100,B2110,B2120,B2CK,B2CL
B2CL,BUF,1,B2CLR
*
MDATA,B2,B2
*
$
MACRO=R8,Y21,Y22,Y23,Y24,Y25,Y26,Y27,Y28,Y31,Y32,Y33,Y34,Y35,Y36,/
Y37,Y38,INCK,T
B3,ROM,2500,B3CLK,12,Y38,Y37,Y36,Y35,Y34,Y33,Y32,Y31
MACRO=R12,B3:1,B3:2,B3:3,B3:4,B3:5,B3:6,B3:7,B3:8,/
B3:9,B3:10,B3:11,B3:12,B310,B320,B330,B340,B350,B360,/
B370,B380,B390,B3100,B3110,B3120,B3CK,B3CL
B3CL,BUF,1,B3CLR
*
MDATA,B3,B3
*
$
$ OR GATES
$
OR1,OR,80/150/60/150,A010,A110,A210,A310,B110,B210,B310,LD=20/50/20/60
OR2,OR,80/150/60/150,A020,A120,A220,A320,B120,B220,B320,LD=20/50/20/60
OR3,OR,80/150/60/150,A030,A130,A230,A330,B130,B230,B330,LD=20/50/20/60
OR4,OR,80/150/60/150,A040,A140,A240,A340,B140,B240,B340,LD=20/50/20/60
OR5,OR,80/150/60/150,A050,A150,A250,A350,B150,B250,B350,LD=20/50/20/60
OR6,OR,80/150/60/150,A060,A160,A260,A360,B160,B260,B360,LD=20/50/20/60
OR7,OR,80/150/60/150,A070,A170,A270,A370,B170,B270,B370,LD=20/50/20/60
OR8,OR,80/150/60/150,A080,A180,A280,A380,B180,B280,B380,LD=20/50/20/60
OR9,OR,80/150/60/150,A090,A190,A290,A390,B190,B290,B390,LD=20/50/20/60
OR10,OR,80/150/60/150,A0100,A1100,A2100,A3100,B1100,B2100,B3100,/
LD=20/50/20/60
OR11,OR,80/150/60/150,A0110,A1110,A2110,A3110,B1110,B2110,B3110,/
LD=20/50/20/60
OR12,OR,80/150/60/150,A0120,A1120,A2120,A3120,B1120,B2120,B3120,/
LD=20/50/20/60
$
$ ADDER
$
MACRO=R12,OR1,OR2,OR3,OR4,OR5,OR6,OR7,OR8,OR9,OR10,OR11,OR12,/
130,140,150,160,170,180,190,200,210,220,230,240,ADDCK,T
*MACLIB,N7400
MACRO=7483,40,Z3,30,150,V,Z2,Y40,20,Z1,10,130,G,F,C5,Z4,160
MACRO=7483,80,Z7,70,190,V,Z6,180,60,Z5,50,170,G,C5,C9,Z8,200
MACRO=7483,120,Z11,110,230,V,Z10,220,100,Z9,90,210,G,C9,C13,Z12,240
MACRO=R12,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11,Z12,10,20,30,40,50,/
60,70,80,90,100,110,120,ADDCK,CLA
ADDCK,BUF,1,CLOCK
$
$QUANTIZER
$
MACRO=QUANT,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11,Z12,/
```

```
ORA,ORB,ORC,ORD,ORE,ORF,ORG,ORH
$
$ CONTROL
$
MACRO=CONTROL,CLOCK,SET,CLA,A0CLK,A1CLK,A2CLK,A3CLK,B1CLK,B2CLK,/
B3CLK,A0CLR,A1CLR,A2CLR,A3CLR,B1CLR,B2CLR,B3CLR,A0CK,A1CK,A2CK,A3CK,/
B1CK,B2CK,B3CK,INCK,SETB
SET,SW
CLOCK,CLOC,1400,1000,100
*
LPKE,ALL
SET,SET,0,0,100,1,250,0
$DISP,25000,37000,50,ADDCK,OR12,OR11,OR10,OR9,OR8,OR7,OR6,OR5,/
$OR4,OR3,OR2,OR1
$DISP,6500,10000,50,A0CL,A0CK,A012Q,A011Q,A010Q,A09Q,A08Q,/
$A07Q,A06Q,A05Q,A04Q,A03Q,A02Q,A01Q
$DISP,7500,10000,50,A1CL,A1CK,A112Q,A111Q,A110Q,A19Q,A18Q,/
$A17Q,A16Q,A15Q,A14Q,A13Q,A12Q,A11Q
$DISP,2000,5000,50,A1CLK,A1CL,A1CK,A1:12,A1:11,A1:10,/
$A1:9,A1:8,A1:7,A1:6,A1:5,A1:4,A1:3,A1:2,A1:1
$DISP,2000,5000,50,INCK,A1CLK,X18,X17,X16,X15,X14,X13,X12,X11
$DISP,0,5000,50,INCK,A0CLK,X08,X07,X06,X05,X04,X03,X02,X01
$DISP,28800,35000,50,B1CL,B1CK,B112Q,B111Q,B110Q,B19Q,/
$B18Q,B17Q,B16Q,B15Q,B14Q,B13Q,B12Q,B11Q
$DISP,0,78000,100,INCK,ADDCK,ORH,ORG,ORF,ORE,ORD,ORC,ORB,ORA
$DISP,0,78000,100,CLA,ADDCK,Z12,Z11,Z10,Z9,Z8,Z7,Z6,Z5,Z4,Z3,Z2,Z1
$DISP,69000,75000,50,ADDCK,A01Q,A02Q,A11Q,A12Q,A21Q,A22Q,A31Q,A32Q,/
$B11Q,B12Q,B21Q,B22Q,B31Q,B32Q
$DISP,0,41000,100,CLOCK,SET,INCK,CLA,A0CLK,A1CLK,A2CLK,A3CLK,B1CLK,/
$B2CLK,B3CLK,A0CLR,A1CLR,A2CLR,A3CLR
DISP,0,41000,100,CLOCK,B1CLR,B2CLR,B3CLR,A0CK,A1CK,A2CK,A3CK,/
B1CK,B2CK,B3CK,INCK,SETB
CLOCK,SET,CLA,A0CLK,A1CLK,A2CLK,A3CLK,B1CLK,B2CLK,B3CLK,/
A0CLR,A1CLR,A2CLR,A3CLR,B1CLR,B2CLR,B3CLR,A0CK,A1CK,A2CK,A3CK,/
B1CK,B2CK,B3CK,INCK,SETB
SETB,INV,80/220/50/150,SET
MACRO=C7473,CLOCK,SETB,14QB,V,CLOCK,SETB,A0CLK,2QB,A1CLK,1QB,/
G,A0CLK,1QB,INCK
MACRO=C7473,CLOCK,SETB,2QB,V,CLOCK,SETB,A2CLK,4QB,A3CLK,3QB,/
G,A2CLK,3QB,A1CLK
MACRO=C7473,CLOCK,SETB,4QB,V,CLOCK,SETB,B1CLK,6QB,B2CLK,5QB,/
G,B1CLK,5QB,A3CLK
MACRO=C7473,CLOCK,SETB,6QB,V,CLOCK,SETB,B3CLK,8QB,A3CLR,7QB,/
G,B3CLK,7QB,B2CLK
MACRO=C7473,CLOCK,SETB,8QB,V,CLOCK,SETB,B1CLR,10QB,B2CLR,9QB,/
G,B1CLR,9QB,A3CLR
MACRO=C7473,CLOCK,SETB,10QB,V,CLOCK,SETB,B3CLR,12QB,12Q,11QB,/
G,B3CLR,11QB,B2CLR
MACRO=X7476,CLOCK,T,SETB,12Q,V,CLOCK,SETB,T,13Q,14QB,INCK,13QB,/
G,13QB,13Q,12QB
A0CK,BUF,80/280/50/150:100,B1CLK
A1CK,BUF,80/280/50/150:100,B2CLK
A2CK,BUF,80/280/50/150:100,B3CLK
A3CK,BUF,80/280/50/150:100,A3CLR
B1CK,BUF,80/280/50/150:100,B1CLR
B2CK,BUF,80/280/50/150:100,B2CLR
B3CK,BUF,80/280/50/150:100,B3CLR
A0CLR,BUF,1,B1CLK
A1CLR,BUF,1,B2CLK
A2CLR,BUF,1,B3CLK
CLA,INV,80/220/50/150,A3CLK
```

```
$QUANTIZER
Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11,Z12,ORA,ORB,ORC,ORD,ORE,ORF,ORG,ORH
NGA0,AND,80/220/50/150,Z12,Z11,Z10,Z9,Z8,Z7,Z6,LD=30/75/30/90
PGA0,AND,80/220/50/150,-Z12,-Z11,-Z10,-Z9,-Z8,-Z7,-Z6,LD=30/75/30/90
NGB0,AND,80/220/50/150,NGA0,Z5,LD=30/75/30/90
PGB0,AND,80/220/50/150,PGA0,-Z5,LD=30/75/30/90
ORL0,OR,80/220/50/150,NGB0,PGB0,LD=30/75/30/90
NGA1,AND,80/220/50/150,Z12,Z11,Z10,Z9,Z8,Z7,Z6,LD=30/75/30/90
PGA1,AND,80/220/50/150,-Z12,-Z11,-Z10,-Z9,-Z8,-Z7,-Z6,LD=30/75/30/90
NGB1,AND,80/220/50/150,NGA1,-Z5,LD=30/75/30/90
PGB1,AND,80/220/50/150,PGA1,Z5,LD=30/75/30/90
ORL1,OR,80/220/50/150,NGB1,PGB1,LD=30/75/30/90
NGA2,AND,80/220/50/150,Z12,Z11,Z10,Z9,Z8,Z7,LD=30/75/30/90
PGA2,AND,80/220/50/150,-Z12,-Z11,-Z10,-Z9,-Z8,-Z7,LD=30/75/30/90
NGB2,AND,80/220/50/150,NGA2,-Z6,LD=30/75/30/90
PGB2,AND,80/220/50/150,PGA2,Z6,LD=30/75/30/90
ORL2,OR,80/220/50/150,NGB2,PGB2,LD=30/75/30/90
NGA3,AND,80/220/50/150,Z12,Z11,Z10,Z9,Z8,LD=30/75/30/90
PGA3,AND,80/220/50/150,-Z12,-Z11,-Z10,-Z9,-Z8,LD=30/75/30/90
NGB3,AND,80/220/50/150,NGA3,-Z7,LD=30/75/30/90
PGB3,AND,80/220/50/150,PGA3,Z7,LD=30/75/30/90
ORL3,OR,80/220/50/150,NGB3,PGB3,LD=30/75/30/90
NGA4,AND,80/220/50/150,Z12,Z11,Z10,Z9,LD=30/75/30/90
PGA4,AND,80/220/50/150,-Z12,-Z11,-Z10,-Z9,LD=30/75/30/90
NGB4,AND,80/220/50/150,NGA4,-Z8,LD=30/75/30/90
PGB4,AND,80/220/50/150,PGA4,Z8,LD=30/75/30/90
ORL4,OR,80/220/50/150,NGB4,PGB4,LD=30/75/30/90
NGA5,AND,80/220/50/150,Z12,Z11,Z10,LD=30/75/30/90
PGA5,AND,80/220/50/150,-Z12,-Z11,-Z10,LD=30/75/30/90
NGB5,AND,80/220/50/150,NGA5,-Z9,LD=30/75/30/90
PGB5,AND,80/220/50/150,PGA5,Z9,LD=30/75/30/90
ORL5,OR,80/220/50/150,NGB5,PGB5,LD=30/75/30/90
NGA6,AND,80/220/50/150,Z12,Z11,LD=30/75/30/90
PGA6,AND,80/220/50/150,-Z12,-Z11,LD=30/75/30/90
NGB6,AND,80/220/50/150,NGA6,-Z10,LD=30/75/30/90
PGB6,AND,80/220/50/150,PGA6,Z10,LD=30/75/30/90
ORL6,OR,80/220/50/150,NGB6,PGB6,LD=30/75/30/90
NGB7,AND,80/220/50/150,Z12,-Z11,LD=30/75/30/90
PGB7,AND,80/220/50/150,-Z12,Z11,LD=30/75/30/90
ORL7,OR,80/220/50/150,NGB7,PGB7,LD=30/75/30/90
L0A,AND,80/220/50/150,ORL0,Z1,LD=30/75/30/90
L0B,AND,80/220/50/150,ORL0,Z2,LD=30/75/30/90
L0C,AND,80/220/50/150,ORL0,Z3,LD=30/75/30/90
L0D,AND,80/220/50/150,ORL0,Z4,LD=30/75/30/90
L0E,AND,80/220/50/150,ORL0,F,LD=30/75/30/90
L0F,AND,80/220/50/150,ORL0,F,LD=30/75/30/90
L0G,AND,80/220/50/150,ORL0,F,LD=30/75/30/90
L0H,AND,80/220/50/150,ORL0,Z12,LD=30/75/30/90
L1A,AND,80/220/50/150,ORL1,Z1,LD=30/75/30/90
L1B,AND,80/220/50/150,ORL1,Z2,LD=30/75/30/90
L1C,AND,80/220/50/150,ORL1,Z3,LD=30/75/30/90
L1D,AND,80/220/50/150,ORL1,Z4,LD=30/75/30/90
L1E,AND,80/220/50/150,ORL1,T,LD=30/75/30/90
L1F,AND,80/220/50/150,ORL1,F,LD=30/75/30/90
L1G,AND,80/220/50/150,ORL1,F,LD=30/75/30/90
```

```
L1H,AND,80/220/50/150,ORL1,Z12,LD=30/75/30/90
L2A,AND,80/220/50/150,ORL2,Z2,LD=30/75/30/90
L2B,AND,80/220/50/150,ORL2,Z3,LD=30/75/30/90
L2C,AND,80/220/50/150,ORL2,Z4,LD=30/75/30/90
L2D,AND,80/220/50/150,ORL2,Z5,LD=30/75/30/90
L2E,AND,80/220/50/150,ORL2,F,LD=30/75/30/90
L2F,AND,80/220/50/150,ORL2,T,LD=30/75/30/90
L2G,AND,80/220/50/150,ORL2,F,LD=30/75/30/90
L2H,AND,80/220/50/150,ORL2,Z12,LD=30/75/30/90
L3A,AND,80/220/50/150,ORL3,Z3,LD=30/75/30/90
L3B,AND,80/220/50/150,ORL3,Z4,LD=30/75/30/90
L3C,AND,80/220/50/150,ORL3,Z5,LD=30/75/30/90
L3D,AND,80/220/50/150,ORL3,Z6,LD=30/75/30/90
L3E,AND,80/220/50/150,ORL3,T,LD=30/75/30/90
L3F,AND,80/220/50/150,ORL3,T,LD=30/75/30/90
L3G,AND,80/220/50/150,ORL3,F,LD=30/75/30/90
L3H,AND,80/220/50/150,ORL3,Z12,LD=30/75/30/90
L4A,AND,80/220/50/150,ORL4,Z4,LD=30/75/30/90
L4B,AND,80/220/50/150,ORL4,Z5,LD=30/75/30/90
L4C,AND,80/220/50/150,ORL4,Z6,LD=30/75/30/90
L4D,AND,80/220/50/150,ORL4,Z7,LD=30/75/30/90
L4E,AND,80/220/50/150,ORL4,F,LD=30/75/30/90
L4F,AND,80/220/50/150,ORL4,F,LD=30/75/30/90
L4G,AND,80/220/50/150,ORL4,T,LD=30/75/30/90
L4H,AND,80/220/50/150,ORL4,Z12,LD=30/75/30/90
L5A,AND,80/220/50/150,ORL5,Z5,LD=30/75/30/90
L5B,AND,80/220/50/150,ORL5,Z6,LD=30/75/30/90
L5C,AND,80/220/50/150,ORL5,Z7,LD=30/75/30/90
L5D,AND,80/220/50/150,ORL5,Z8,LD=30/75/30/90
L5E,AND,80/220/50/150,ORL5,T,LD=30/75/30/90
L5F,AND,80/220/50/150,ORL5,F,LD=30/75/30/90
L5G,AND,80/220/50/150,ORL5,T,LD=30/75/30/90
L5H,AND,80/220/50/150,ORL5,Z12,LD=30/75/30/90
L6A,AND,80/220/50/150,ORL6,Z6,LD=30/75/30/90
L6B,AND,80/220/50/150,ORL6,Z7,LD=30/75/30/90
L6C,AND,80/220/50/150,ORL6,Z8,LD=30/75/30/90
L6D,AND,80/220/50/150,ORL6,Z9,LD=30/75/30/90
L6E,AND,80/220/50/150,ORL6,F,LD=30/75/30/90
L6F,AND,80/220/50/150,ORL6,T,LD=30/75/30/90
L6G,AND,80/220/50/150,ORL6,T,LD=30/75/30/90
L6H,AND,80/220/50/150,ORL6,Z12,LD=30/75/30/90
L7A,AND,80/220/50/150,ORL7,Z7,LD=30/75/30/90
L7B,AND,80/220/50/150,ORL7,Z8,LD=30/75/30/90
L7C,AND,80/220/50/150,ORL7,Z9,LD=30/75/30/90
L7D,AND,80/220/50/150,ORL7,Z10,LD=30/75/30/90
L7E,AND,80/220/50/150,ORL7,T,LD=30/75/30/90
L7F,AND,80/220/50/150,ORL7,T,LD=30/75/30/90
L7G,AND,80/220/50/150,ORL7,T,LD=30/75/30/90
L7H,AND,80/220/50/150,ORL7,Z12,LD=30/75/30/90
ORA,OR,80/220/50/150,L0A,L1A,L2A,L3A,L4A,L5A,L6A,L7A,LD=30/75/30/90
ORB,OR,80/220/50/150,L0B,L1B,L2B,L3B,L4B,L5B,L6B,L7B,LD=30/75/30/90
ORC,OR,80/220/50/150,L0C,L1C,L2C,L3C,L4C,L5C,L6C,L7C,LD=30/75/30/90
ORD,OR,80/220/50/150,L0D,L1D,L2D,L3D,L4D,L5D,L6D,L7D,LD=30/75/30/90
ORE,OR,80/220/50/150,L0E,L1E,L2E,L3E,L4E,L5E,L6E,L7E,LD=30/75/30/90
ORF,OR,80/220/50/150,L0F,L1F,L2F,L3F,L4F,L5F,L6F,L7F,LD=30/75/30/90
ORG,OR,80/220/50/150,L0G,L1G,L2G,L3G,L4G,L5G,L6G,L7G,LD=30/75/30/90
ORH,OR,80/220/50/150,L0H,L1H,L2H,L3H,L4H,L5H,L6H,L7H,LD=30/75/30/90
```

FILE: R12 MACRO

```
J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,Q1,Q2,Q3,Q4,Q5,Q6,Q7,/
Q8,Q9,Q10,Q11,Q12,CK,CL
MACRO=X7404,J1,K1,J2,K2,J3,K3,G,K4,J4,K5,J5,K6,J6,V
MACRO=X7404,J7,K7,J8,K8,J9,K9,G,K10,J10,K11,J11,K12,J12,V
MACRO=X7473,CK,CL,K1,V,CK,CL,J2,2QB,Q2,K2,G,Q1,1QB,J1
MACRO=X7473,CK,CL,K3,V,CK,CL,J4,4QB,Q4,K4,G,Q3,3QB,J3
MACRO=X7473,CK,CL,K5,V,CK,CL,J6,6QB,Q6,K6,G,Q5,5QB,J5
MACRO=X7473,CK,CL,K7,V,CK,CL,J8,8QB,Q8,K8,G,Q7,7QB,J7
MACRO=X7473,CK,CL,K9,V,CK,CL,J10,10QB,Q10,K10,G,Q9,9QB,J9
MACRO=X7473,CK,CL,K11,V,CK,CL,J12,12QB,Q12,K12,G,Q11,11QB,J11
DUMMY,AND,1,1QB,2QB,3QB,4QB,5QB,6QB,7QB,8QB,9QB,10QB,11QB,12QB
```

FILE: R8 MACRO

```
J1,J2,J3,J4,J5,J6,J7,J8,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,CK,CL
MACRO=X7404,J1,K1,J2,K2,J3,K3,G,K4,J4,K5,J5,K6,J6,V
MACRO=X7404,J7,K7,J8,K8,T,K9,G,K10,T,K11,T,K12,T,V
MACRO=X7473,CK,CL,K1,V,CK,CL,J2,2QB,Q2,K2,G,Q1,1QB,J1
MACRO=X7473,CK,CL,K3,V,CK,CL,J4,4QB,Q4,K4,G,Q3,3QB,J3
MACRO=X7473,CK,CL,K5,V,CK,CL,J6,6QB,Q6,K6,G,Q5,5QB,J5
MACRO=X7473,CK,CL,K7,V,CK,CL,J8,8QB,Q8,K8,G,Q7,7QB,J7
DUMMY,AND,1,1QB,2QB,3QB,4QB,5QB,6QB,7QB,8QB,K9,K10,K11,K12
```

FILE: C7473 MACRO

```
1CK,1CL,1K,V,2CK,2CL,2J,2QB,2Q,2K,G,1Q,1QB,1J
Q11,JK,100,1CK,1J,1K,F,1CL:-1
1Q,BUF,1/100/1/150,Q11
1QB,INV,1/100/1/150,Q11
Q12,JK,100,2CK,2J,2K,F,2CL:-1
2Q,BUF,1/100/1/150,Q12
2QB,INV,1/100/1/150,Q12
```

FILE: X7404 MACRO

```
I1,01,I2,02,I3,03,G,04,I4,05,I5,06,I6,V
01,INV,80/150/50/150,I1,LD=30/75/30/90
02,INV,80/150/50/150,I2,LD=30/75/30/90
03,INV,80/150/50/150,I3,LD=30/75/30/90
04,INV,80/150/50/150,I4,LD=30/75/30/90
05,INV,80/150/50/150,I5,LD=30/75/30/90
06,INV,80/150/50/150,I6,LD=30/75/30/90
```

FILE: X7476 MACRO

```
1CK,1S,1R,1J,V,2CK,2S,2R,2J,2QB,2Q,2K,G,1QB,1Q,1K
QI1,JK,100,1CK,1J,1K,1S:-1,1R:-1
1Q,BUF,1/100/1/150:200,QI1
1QB,INV,1/100/1/150:200,QI1
QI2,JK,100,2CK,2J,2K,2S:-1,2R:-1
2Q,BUF,1/100/1/150:200,QI2
2QB,INV,1/100/1/150:200,QI2
```

APPENDIX 7C

This appendix lists the contents of the ROM multipliers for each of the seven coefficients:

$a_o = a_3 = 0.2726230$ 7C-2

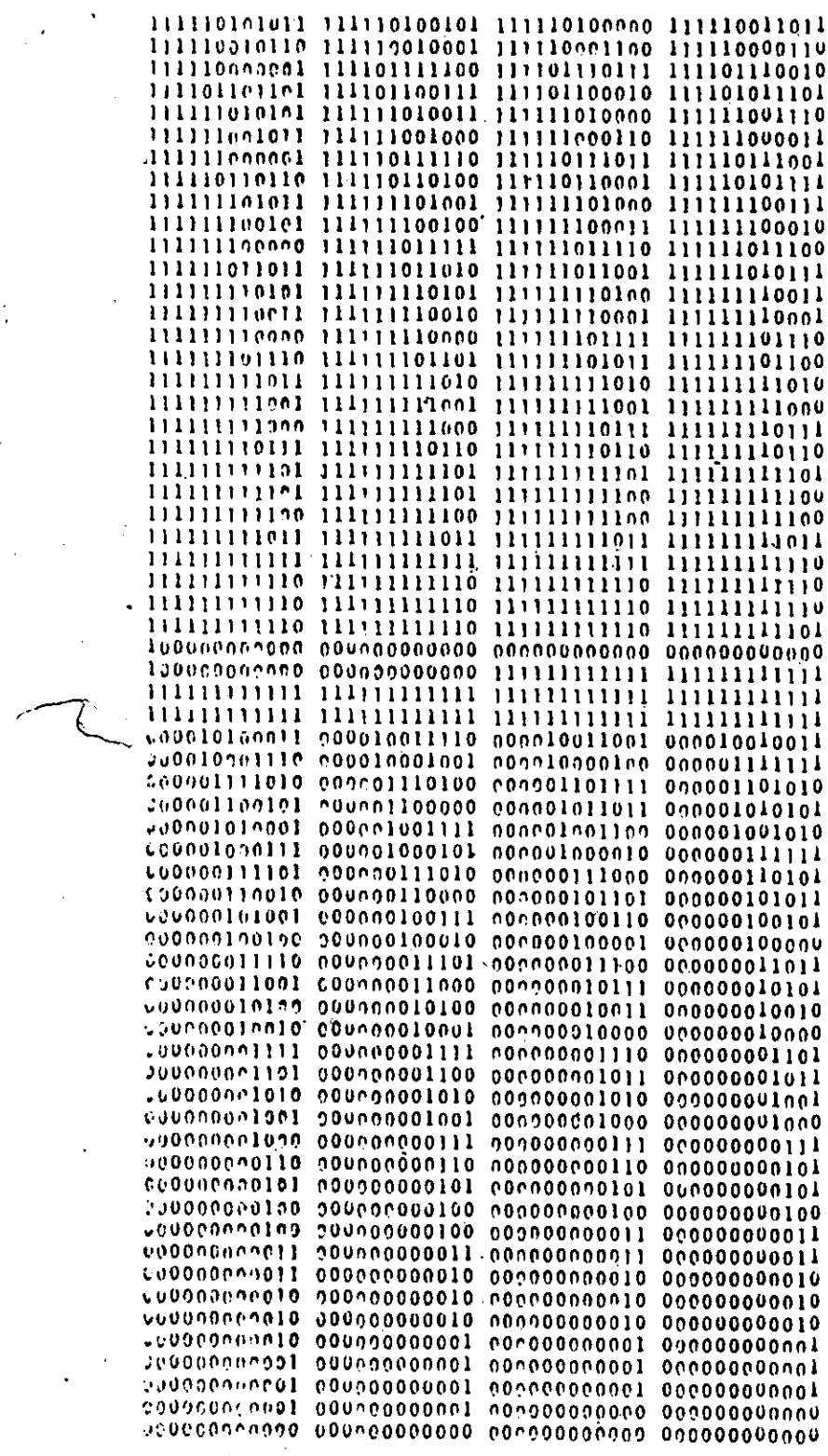
$a_1 = a_2 = 0.0808208$ 7C-3

$b_1 = -0.9877751$ 7C-4

$b_2 = 0.7787396$ 7C-5

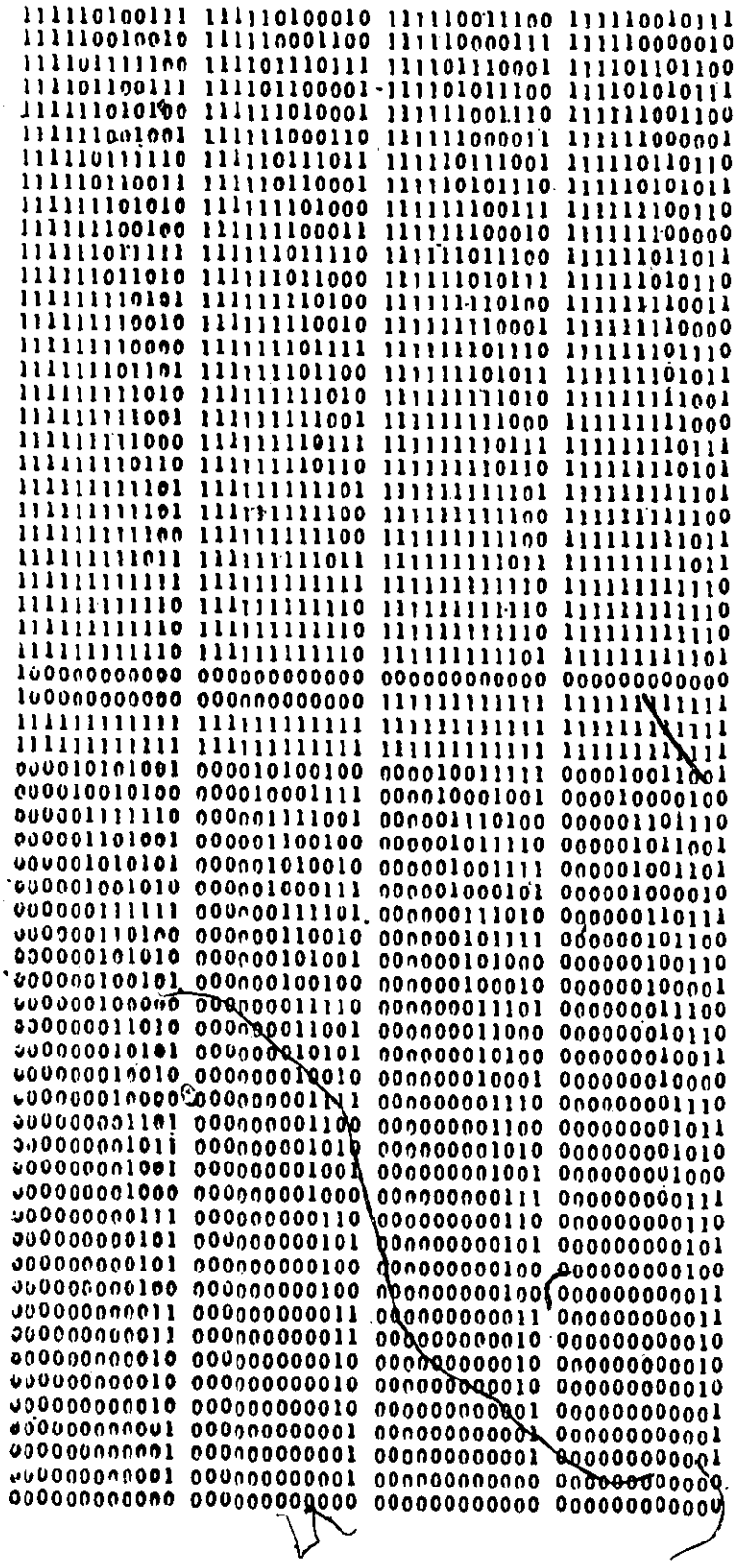
$b_3 = 0.084076B$ 7C-6

```
111011100000 111011001111 111010111101 111010101100
111010011010 111010001001 111001110111 111001100110
111001010101 111001000011 111000110010 111000100000
110000001111 110111111101 110111101100 110111011010
111101110000 111101100111 111101011111 111101010110
111101001101 111101000100 111100111100 111100110011
111100101010 111100100010 111100011001 111100010000
111100000111 111011111111 111011110110 111011101101
111110111000 111110110100 111110101111 111110101011
111110100111 111110100010 111110011110 111110011001
111110010101 111110010001 111110001100 111110001000
111110000100 111101111111 111101111011 111101110111
111111011100 111111011010 111111011000 111111010101
111111010011 111111010001 111111001111 111111001101
111111001011 111111001000 111111000110 111111000100
111111000010 111111000000 111110111011 111110111011
111111101110 111111101101 111111101100 111111101011
111111101010 111111101001 111111100111 111111100110
111111100101 111111100100 111111100011 111111100010
111111100001 111111100000 111111011111 111111011110
111111110111 111111110110 111111110110 111111110101
111111110101 111111110100 111111110100 111111110011
111111110011 111111110010 111111110010 111111110001
111111110000 111111110000 111111101111 111111101111
111111111100 111111111011 111111111011 111111111011
111111111010 111111111010 111111111010 111111111010
111111111001 111111111001 111111111001 111111111001
111111111000 111111111000 111111111000 111111110111
000000000000 000000000000 111111111111 111111111111
111111111111 111111111111 111111111110 111111111110
111111111110 111111111101 111111111101 111111111101
111111111101 111111111100 111111111100 111111111100
001000100110 001000010100 001000000011 000111110001
000111100000 000111001110 000110111101 000110101011
000110011010 000110001001 000101110111 000101100110
000101010100 000101000011 000100110001 000100100000
000100010011 000100001010 000100000001 000011111001
000011110000 000011100111 000011011110 000011010110
000011001101 000011000100 000010111100 000010110011
000010101010 000010100001 000010011001 000010010000
000010001001 000010000101 000010000001 000001111100
000001111000 000001110100 000001101111 000001101011
000001100111 000001100010 000001011110 000001011001
000001010101 000001010001 000001001100 000001001000
000001000101 000001000011 000001000000 000000111110
000000111100 000000111010 000000111000 000000110101
000000110011 000000110001 000000101111 000000101101
000000101011 000000101000 000000100110 000000100100
000000100010 000000100001 000000100000 000000011111
000000011110 000000011101 000000011100 000000011011
000000011010 000000011001 000000010111 000000010110
000000010101 000000010100 000000010011 000000010010
000000010001 000000010001 000000010000 000000010000
000000001111 000000001110 000000001110 000000001101
000000001101 000000001100 000000001100 000000001011
000000001011 000000001010 000000001010 000000001001
000000001001 000000001000 000000001000 000000001000
000000000111 000000000111 000000000111 000000000111
000000000110 000000000110 000000000110 000000000110
000000000101 000000000101 000000000101 000000000100
000000000100 000000000100 000000000100 000000000011
000000000011 000000000011 000000000011 000000000010
000000000010 000000000010 000000000001 000000000001
000000000001 000000000001 000000000000 000000000000
```



```
101111101101 101110101110 101101101110 101100101111
101011110000 101010110001 101001110010 101000110010
100111110011 100110110100 100101110101 100100110110
100011110110 100010110111 100001111000 100000111001
110111110110 110111010111 110110110111 110110011000
110101111000 110101011000 110100111001 110100011001
110011111010 110011011010 110010111010 110010011011
110001111011 110001011100 110000111100 110000011100
111011111011 111011101011 111011011100 111011001100
111010111100 111010101100 111010011100 111010001101
111001111101 111001101101 111001011101 111001001101
111000111110 111000101110 111000011110 111000001110
110111111110 110111101110 110111011110 110111001110
110101111110 110101101110 110101011110 110101001110
110100111110 110100101110 110100011110 110100001110
110011111111 110011101111 110011011111 110011001111
110010111111 110010101111 110010011111 110010001111
110001111111 110001101111 110001011111 110001001111
110000111111 110000101111 110000011111 110000001111
101111101111 101110101111 101101101111 101100101111
101011110000 101010110001 101001110010 101000110010
100111110011 100110110100 100101110101 100100110110
100011110110 100010110111 100001111000 100000111001
011111101101 011110101110 011101101110 011100101111
011011110000 011010110001 011001110010 011000110010
010111110011 010110110100 010101110101 010100110110
010011110110 010010110111 010001111000 010000111001
001111101101 001110101110 001101101110 001100101111
001011110000 001010110001 001001110010 001000110010
000111110011 000110110100 000101110101 000100110110
000011110110 000010110111 000001111000 000000111001
```

```
001100110110 001101101000 001110011010 001111001100
001111111110 010000110000 010001100001 010010010011
010011000101 010011110111 010100101001 010101011011
010110001100 010110111110 010111110000 011000100010
000110011011 000110110100 000111001101 000111100110
000111111111 001000011000 001000110001 001001001010
001001100011 001001111011 001010010100 001010101101
001011000110 001011011111 001011111000 001100010001
000011001110 000011011010 000011100111 000011110011
000011111111 000100001100 000100011000 000100100101
000100110001 000100111110 000101001010 000101010111
000101100011 000101110000 000101111100 000110001000
000001100111 000001101101 000001110011 000001111001
000010000000 000010000110 000010001100 000010010010
000010011001 000010011111 000010100101 000010101011
000010110010 000010111000 000011000110 000011000100
000000110011 000000110111 000000111010 000000111101
000001000000 000001000011 000001000110 000001001001
000001001100 000001001111 000001010011 000001010110
000001011001 000001011100 000001011111 000001100010
000000011010 000000011011 000000011101 000000011110
000000100000 000000100001 000000100011 000000100101
000000100110 000000101000 000000101001 000000101011
000000101100 000000101110 000000110000 000000110001
000000001101 000000001110 000000001110 000000001111
000000010000 000000010001 000000010010 000000010010
000000010011 000000010100 000000010101 000000010101
000000010110 000000010111 000000011000 000000011001
000000000000 000000000001 000000000010 000000000011
000000000100 000000000100 000000000101 000000000110
000000000111 000000000111 000000001000 000000001001
000000001010 000000001011 000000001011 000000001100
100111011110 101000010000 101001000010 101001110100
101010100101 101011010111 101100001001 101100111011
101101101101 101110011111 101111010000 110000000010
110000110100 110001100110 110010011000 110011001010
110011101111 110100001000 110100100001 110100111010
110101010011 110101101100 110110000101 110110011101
110110110110 111011001111 111011101000 111000000001
111000011010 111000110011 111001001100 111001100101
111001111000 111010000100 111010010000 111010011101
111010101001 111010110110 111011000010 111011001111
111011011011 111011101000 111011110100 111100000001
111100001101 111100011001 111100100110 111100110010
111100111100 111101000010 111101001000 111101001110
111101010101 111101011011 111101100001 111101100111
111101101110 111101110100 111101111010 111110000000
111110000111 111110001101 111110010011 111110011001
111110011110 111110100001 111110100100 111110100111
111110101010 111110101101 111110110001 111110110100
111110110111 111110111010 111110111101 111111000000
111111000011 111111000110 111111001001 111111001101
111111001111 111111010000 111111010010 111111010100
111111010101 111111010111 111111011000 111111011010
111111011011 111111011101 111111011111 111111100000
111111100010 111111100011 111111100101 111111100110
111111100111 111111101000 111111101001 111111101010
111111101011 111111101011 111111101100 111111101101
111111101110 111111101110 111111101111 111111110000
111111110001 111111110010 111111110010 111111110011
111111110100 111111110101 111111110101 111111110110
111111110111 111111111000 111111111001 111111111001
111111111010 111111111011 111111111100 111111111100
111111111101 111111111110 111111111111 000000000000
```



APPENDIX 7D

This appendix lists the F/LOGIC simulation of the response of the third-order SPDF section to the sampled sinewave input of Fig.7.08.

TIME	INV CLA	AND K	XOR Z12	XOR Z11	XOR Z10	XOR Z9	XOR Z8	XOR Z7	XOR Z6	XOR Z5	XOR Z4	XOR Z3	XOR Z2	XOR Z1
369600	--1	0	0	X	X	X	X	X	X	X	X	X	<	<
371000	--1	0	0	0	--1	--1	--1	--1	--1	0	--1	--1	0	0
372400	--1	0	0	0	--1	--1	--1	--1	--1	0	--1	--1	0	0
373800	--1	0	0	0	--1	--1	--1	--1	--1	0	--1	--1	0	0
375200	--1	0	0	0	--1	--1	--1	--1	--1	0	--1	--1	0	0
376600	--1	0	0	0	--1	--1	--1	--1	--1	0	--1	--1	0	0
378000	0	0	0	0	--1	--1	--1	--1	--1	0	--1	--1	0	0
379400	>	0	0	0	0	0	0	0	0	0	0	0	0	0
380800	--1	0	0	0	0	0	>	>	0	0	0	>	0	0
382200	--1	0	0	0	0	0	X	<	0	>	X	<	0	0
383600	--1	0	0	0	X	X	X	>	X	X	X	X	0	>
385000	--1	0	0	0	>	<	X	X	X	X	0	0	>	<
386400	--1	0	0	X	X	X	X	X	<	0	X	X	X	X
387800	--1	0	0	X	X	0	X	X	X	X	X	X	<	0
389200	--1	0	0	X	X	X	X	X	0	0	>	X	>	0
390600	--1	0	0	0	--1	--1	--1	0	--1	--1	--1	0	--1	0
392000	--1	0	0	0	--1	--1	--1	0	--1	--1	--1	0	--1	0
393400	--1	0	0	0	--1	--1	--1	0	--1	--1	--1	0	--1	0
394800	--1	0	0	0	--1	--1	--1	0	--1	--1	--1	0	--1	0
396200	--1	0	0	0	--1	--1	--1	0	--1	--1	--1	0	--1	0
397600	0	0	0	0	--1	--1	--1	0	--1	--1	--1	0	--1	0
399000	>	0	0	0	0	0	0	0	0	0	0	0	0	0
400400	--1	0	0	0	0	0	0	>	>	0	>	0	>	>
401800	--1	0	0	0	0	0	X	X	X	X	<	X	X	X
403200	--1	0	0	0	0	0	X	X	X	X	X	X	X	--1
404600	--1	0	0	0	0	X	X	X	X	X	X	X	X	<
406000	--1	0	0	X	X	0	X	X	X	X	X	X	X	X
407400	--1	0	0	X	X	X	X	X	<	X	X	>	>	>
408800	--1	0	0	X	X	X	X	X	X	X	X	X	X	<
410200	--1	0	0	0	--1	0	--1	--1	--1	0	0	--1	0	0
411600	--1	0	0	0	--1	0	--1	--1	--1	0	0	--1	0	0
413000	--1	0	0	0	--1	0	--1	--1	--1	0	0	--1	0	0
414400	--1	0	0	0	--1	0	--1	--1	--1	0	0	--1	0	0
415800	--1	0	0	0	--1	0	--1	--1	--1	0	0	--1	0	0
417200	0	0	0	0	--1	0	--1	--1	--1	0	0	--1	0	0
418600	>	0	0	0	0	0	0	0	0	0	0	0	0	0
420000	--1	0	0	0	0	0	0	0	0	0	0	0	0	0
421400	--1	0	0	0	0	0	0	0	>	0	0	0	0	0
422800	--1	0	0	0	0	0	0	>	<	>	>	0	>	0
424200	--1	0	0	0	0	>	X	X	X	X	X	X	X	>
425600	--1	0	0	X	X	X	X	X	X	<	<	X	X	<
427000	--1	0	0	X	X	X	X	X	X	>	X	X	X	>
428400	--1	0	0	X	X	X	X	X	X	>	>	X	<	<
429800	--1	0	0	0	0	0	--1	--1	0	--1	--1	--1	0	0
431200	--1	0	0	0	0	0	--1	--1	0	--1	--1	--1	0	0
432600	--1	0	0	0	0	0	--1	--1	0	--1	--1	--1	0	0
434000	--1	0	0	0	0	0	--1	--1	0	--1	--1	--1	0	0
435400	--1	0	0	0	0	0	--1	--1	0	--1	--1	--1	0	0
436800	0	0	0	0	0	0	--1	--1	0	--1	--1	--1	0	0
438200	>	0	0	0	0	0	0	0	0	0	0	0	0	0
439600	--1	0	0	>	>	>	>	0	0	>	0	>	0	>
441000	--1	0	--1	X	X	X	X	0	X	X	X	X	X	X
442400	--1	0	--1	--1	--1	--1	--1	0	>	--1	0	--1	0	--1
443800	--1	0	--1	--1	--1	--1	X	X	X	--1	>	<	0	--1
445200	--1	0	0	X	X	X	X	X	X	X	X	X	X	<
446600	--1	0	0	X	X	0	>	>	X	X	X	X	>	X
448000	--1	0	--1	X	X	X	X	X	X	X	X	X	<	>
449400	--1	0	0	0	0	0	0	0	--1	--1	0	0	0	--1
450800	--1	0	0	0	0	0	0	0	--1	--1	0	0	0	--1
452200	--1	0	0	0	0	0	0	0	--1	--1	0	0	0	--1
453600	--1	0	0	0	0	0	0	0	--1	--1	0	0	0	--1
455000	--1	0	0	0	0	0	0	0	--1	--1	0	0	0	--1
456400	0	0	0	0	0	0	0	0	--1	--1	0	0	0	--1
457800	>	0	0	0	0	0	0	0	0	0	0	0	0	0
459200	--1	0	0	>	>	>	0	0	>	>	>	>	0	0
460600	--1	0	--1	X	X	X	X	X	X	X	X	X	0	0
462000	--1	0	--1	X	X	X	X	X	X	--1	--1	--1	0	0
463400	--1	0	--1	--1	--1	--1	0	X	X	<	<	--1	>	>
464800	--1	0	--1	--1	--1	--1	X	X	X	X	X	X	X	X
466200	--1	0	--1	X	X	--1	<	X	X	X	X	<	<	<
467600	--1	0	--1	X	X	X	X	<	X	X	X	X	X	>
469000	--1	0	--1	--1	--1	0	--1	0	0	--1	--1	--1	--1	--1
470400	--1	0	--1	--1	--1	0	--1	0	0	--1	--1	--1	--1	--1
471800	--1	0	--1	--1	--1	0	--1	0	0	--1	--1	--1	--1	--1
473200	--1	0	--1	--1	--1	0	--1	0	0	--1	--1	--1	--1	--1
474600	--1	0	--1	--1	--1	0	--1	0	0	--1	--1	--1	--1	--1
476000	0	0	--1	--1	--1	0	--1	0	0	--1	--1	--1	--1	--1

```
485800 --1 0   --1  X   X   X   X   X   X  0   X   X   >  --1
487200 --1 0   --1  X   X   X   X   X   X  X   X   X   X   <
488600 --1 0   --1 --1  0   --1  0   --1  0   0   0   0   0   0
490000 --1 0   --1 --1  0   --1  0   --1  0   0   0   0   0   0
491400 --1 0   --1 --1  0   --1  0   --1  0   0   0   0   0   0
492800 --1 0   --1 --1  0   --1  0   --1  0   0   0   0   0   0
494200 --1 0   --1 --1  0   --1  0   --1  0   0   0   0   0   0
495600 0   0   --1 --1  0   --1  0   --1  0   0   0   0   0   0
497000  >  0   0   0   0   0   0   0   0   0   0   0   0   0   0
498400 --1 0   0   >   >   0   >   >   >  0   >   >   0   >   >
499800 --1 0   --1  X   X   X   X   X   X  X   X   <   X   X
501200 --1 0   --1 --1 --1  0   X   X   X  X   X   >   >  --1
502600 --1 0   --1 --1 --1  0   X   X  --1  >   X   X   X   <
504000 --1 0   --1  X   X   X   X   X   X  X   X   X   X   X
505400 --1 0   --1  X   <  --1  X   X   >  --1  >  --1  0   0
506800 --1 0   --1  X   X   X   X   X   X  X   X   <   0   0
508200 --1 0   --1 --1  0   0   0  --1 --1 --1  0   0   0   0
509600 --1 0   --1 --1  0   0   0  --1 --1 --1  0   0   0   0
511000 --1 0   --1 --1  0   0   0  --1 --1 --1  0   0   0   0
512400 --1 0   --1 --1  0   0   0  --1 --1 --1  0   0   0   0
513800 --1 0   --1 --1  0   0   0  --1 --1 --1  0   0   0   0
515200 0   0   --1 --1  0   0   0  --1 --1 --1  0   0   0   0
516600  >  0   0   0   0   0   0   0   0   0   0   0   0   0
518000 --1 0   0   >   >   0   >   >   >   >   >   >   >   >
519400 --1 0   --1  X   X   X   X   X   X   X   X   X   X   <
520800 --1 0   --1 --1 --1  0   <   >   X   X   <   <   X   X
522200 --1 0   --1 --1 --1  0   X   <  --1  0   X   >  --1  0
523600 --1 0   --1  X   X   X   X   X   X   X   X   X   <   >
525000 --1 0   --1  X   X   X   X   X   >  --1  X   X   X   X
526400 --1 0   --1  X   X   X   X   X   X   X   X   0   >  --1
527800 --1 0   --1 --1  0   0   0   0   0  --1  0   0  --1 --1
529200 --1 0   --1 --1  0   0   0   0   0  --1  0   0  --1 --1
530600 --1 0   --1 --1  0   0   0   0   0  --1  0   0  --1 --1
532000 --1 0   --1 --1  0   0   0   0   0  --1  0   0  --1 --1
533400 --1 0   --1 --1  0   0   0   0   0  --1  0   0  --1 --1
534800 0   0   --1 --1  0   0   0   0   0  --1  0   0  --1 --1
536200  >  0   0   0   0   0   0   0   0   0   0   0   0   0
537600 --1 0   0   >   >   >   0   0   >   >   >   >   0   0
539000 --1 0   --1  X   X   <   >   X   X   X   X   <   0   0
540400 --1 0   --1 --1  X   X   X   <  --1  X   X   >   >   >
541800 --1 0   --1 --1  <   >   X   X   X   X   X   X   X   <
543200 --1 0   --1  X   X   X   X   X   >  --1 --1  <   X   X
544600 --1 0   --1  X   X  --1  X   X   X   X   X   X   <   0
546000 --1 0   --1  X   X   X   <   >   X   X   X   >   >   0
547400 --1 0   --1 --1  0   0   0   0  --1  0   0   0  --1 --1  0
548800 --1 0   --1 --1  0   0   0   0  --1  0   0   0  --1 --1  0
550200 --1 0   --1 --1  0   0   0   0  --1  0   0   0  --1 --1  0
551600 --1 0   --1 --1  0   0   0   0  --1  0   0   0  --1 --1  0
553000 --1 0   --1 --1  0   0   0   0  --1  0   0   0  --1 --1  0
554400 0   0   --1 --1  0   0   0   0  --1  0   0   0  --1 --1  0
555800  >  0   0   0   0   0   0   0   0   0   0   0   0   0   0

557200 --1 0   0   >   >   >   >   0   0   >   0   >   0   >
558600 --1 0   --1  X   X   X   <   >   X   X   X   X   X   X
560000 --1 0   --1 --1 --1 --1  X   X   X   X   X   X   X  --1
561400 --1 0   --1 --1 --1  X   X   X   X   X   X   X   <   <
562800 --1 0   --1  X   X  --1  X   X   X   X  --1  <   X   X
564200 --1 0   --1  X   X   X   X   X   >  --1  X   X   0   >
565600 --1 0   --1  X   X   X   X   X   X   >  --1  >   X   <
```

```
567000 --1 0   --1 --1 0   --1 0  0  0  --1 --1 --1 0  0
564400 --1 0   --1 --1 0   --1 0  0  0  --1 --1 --1 0  0
569800 --1 0   --1 --1 0   --1 0  0  0  --1 --1 --1 0  0
571200 --1 0   --1 --1 0   --1 0  0  0  --1 --1 --1 0  0
572600 --1 0   --1 --1 0   --1 0  0  0  --1 --1 --1 0  0
574000 0   0   --1 --1 0   --1 0  0  0  --1 --1 --1 0  0
575400 >   0   0  0  0  0  0  0  0  0  0  0  0  0  0
576800 --1 0   0  0  0  0  0  0  0  0  0  0  0  0  0
578200 --1 0   0  >  >  >  >  >  >  0  0  0  0  0  0
579600 --1 0   --1 X  X  X  X  X  X  0  0  >  >  >  0
581000 --1 0   --1 --1 --1 <  X  X  X  X  X  X  <  >
582400 --1 0   --1 X  X  X  X  X  X  0  X  X  X  <
583800 --1 0   --1 X  X  X  X  X  X  X  X  X  X  >
585200 --1 0   --1 X  X  X  X  X  X  <  X  X  X  <
586600 --1 0   --1 --1 --1 0  0  --1 0  0  --1 0  0  0
588000 --1 0   --1 --1 --1 0  0  --1 0  0  --1 0  0  0
589400 --1 0   --1 --1 --1 0  0  --1 0  0  --1 0  0  0
590800 --1 0   --1 --1 --1 0  0  --1 0  0  --1 0  0  0
592200 --1 0   --1 --1 --1 0  0  --1 0  0  --1 0  0  0
593600 0   0   --1 --1 --1 0  0  --1 0  0  --1 0  0  0
595000 >   0   0  0  0  0  0  0  0  0  0  0  0  0  0
596400 --1 0   0  0  0  0  0  >  >  0  >  0  >  >
597800 --1 0   0  0  0  0  X  X  X  X  X  X  X  X
599200 --1 0   0  X  X  X  X  X  <  0  --1 0  --1 --1
600600 --1 0   0  0  0  0  X  X  X  X  <  >  --1 --1
602000 --1 0   --1 X  X  X  X  X  X  X  X  <  <
603400 --1 0   --1 X  X  X  X  X  <  X  X  X  X  X
604800 --1 0   0  X  X  X  X  X  X  X  X  X  X  >
606200 --1 0   --1 --1 --1 --1 --1 --1 0  0  --1 --1 --1 --1
607600 --1 0   --1 --1 --1 --1 --1 --1 0  0  --1 --1 --1 --1
609000 --1 0   --1 --1 --1 --1 --1 --1 0  0  --1 --1 --1 --1
610400 --1 0   --1 --1 --1 --1 --1 --1 0  0  --1 --1 --1 --1
611800 --1 0   --1 --1 --1 --1 --1 --1 0  0  --1 --1 --1 --1
613200 0   0   --1 --1 --1 --1 --1 --1 0  0  --1 --1 --1 --1
614600 >   0   0  0  0  0  0  0  0  0  0  0  0  0
616000 --1 0   0  0  0  0  >  >  0  0  0  >  0  0
617400 --1 0   0  0  0  0  X  X  >  0  X  X  0  0
618800 --1 0   0  0  0  0  X  X  X  0  0  --1 0  0
620200 --1 0   0  X  X  X  <  --1 --1 >  >  <  0  >
621600 --1 0   0  0  0  0  X  X  X  X  X  X  X  X
623000 --1 0   0  X  X  0  >  X  X  X  <  X  X  <
624400 --1 0   0  X  X  X  X  X  X  X  X  X  X  >
625800 --1 0   0  0  0  --1 0  --1 --1 0  0  0  0  --1
627200 --1 0   0  0  0  --1 0  --1 --1 0  0  0  0  --1
628600 --1 0   0  0  0  --1 0  --1 --1 0  0  0  0  --1
630000 --1 0   0  0  0  --1 0  --1 --1 0  0  0  0  --1
631400 --1 0   0  0  0  --1 0  --1 --1 0  0  0  0  --1
632800 0   0   0  0  0  --1 0  --1 --1 0  0  0  0  --1
634200 >   0   0  0  0  0  0  0  0  0  0  0  0  0
635600 --1 0   0  0  0  >  0  0  0  0  0  0  0  >
637000 --1 0   0  0  X  X  0  0  >  >  >  X  X  X
638400 --1 0   0  0  0  0  --1 0  X  X  X  X  X  --1
639800 --1 0   0  0  0  --1 X  X  X  --1 --1 0  --1 --1
641200 --1 0   0  0  >  <  >  <  >  --1 --1 >  --1 --1
642600 --1 0   0  X  X  X  X  X  X  <  <  X  <  --1
644000 --1 0   0  X  X  X  X  X  X  X  X  X  X  <
645400 --1 0   0  0  --1 0  --1 --1 0  0  0  0  0  0
646800 --1 0   0  0  --1 0  --1 --1 0  0  0  0  0  0
648200 --1 0   0  0  --1 0  --1 --1 0  0  0  0  0  0
```

```
649600 --1 0  0  0 --1 0 --1 --1 0  0  0  0  0  0  0
651000 --1 0  0  0 --1 0 --1 --1 0  0  0  0  0  0  0
652400 0   0  0  0 --1 0 --1 --1 0  0  0  0  0  0  0
653800 >   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
655200 --1 0  0  0  0  > 0  0  0  >  0  0  >  >
656600 --1 0  0  0  X  X 0  >  X  X  X  X  X  X
658000 --1 0  0  0  0 --1 X  X  X  X  X  X  <  --1
659400 --1 0  0  0  0 --1 X  X  0  <  X  X  X  <
660800 --1 0  0  X  X  X  X  X  X  X  X  X  >  >
662200 --1 0  0  X  >  0  X  <  >  X  X  X  X  X
663600 --1 0  0  X  X  X  X  X  X  X  >  >  --1 --1
665000 --1 0  0  0 --1 --1 --1 0  --1 0  --1 --1 --1 --1
666400 --1 0  0  0 --1 --1 --1 0  --1 0  --1 --1 --1 --1
667800 --1 0  0  0 --1 --1 --1 0  --1 0  --1 --1 --1 --1
669200 --1 0  0  0 --1 --1 --1 0  --1 0  --1 --1 --1 --1
670600 --1 0  0  0 --1 --1 --1 0  --1 0  --1 --1 --1 --1
672000 0   0  0  0 --1 --1 --1 0  --1 0  --1 --1 --1 --1
673400 >   0  0  0  0  0  0  0  0  0  0  0  0  0  0
674800 --1 0  0  0  0  >  0  0  0  0  0  0  0  >
676200 --1 0  0  0  X  X  0  >  0  >  0  0  >  <
677600 --1 0  0  0  0 --1 X  <  X  X  X  X  X  X
679000 --1 0  0  0  0 --1 X  >  0  X  X  <  --1 0
```