



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform 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 an inferior photocopy.

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

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme 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 qualité inférieure.

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

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

**DIGITAL FILTER DESIGN
USING STORED PRODUCT ROMs**

by

HENDRA TATANGSURJA

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical Engineering

Ottawa, Ontario, 1987



Hendra Tatangsurja, Ottawa, Canada, 1987.

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-40745-X



UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ABSTRACT

A digital filter design technique using stored-product ROMs replacing conventional hardware multipliers is presented. The objectives are minimization of the coefficient roundoff errors and the product roundoff errors, and enhancement of the filtering speed. The results of the roundoff error accumulation are compared with those obtained from the same filter with short coefficient wordlength, and also with those obtained from the wave digital filter with multipliers. The bit-slice implementation of the stored-product digital filters is given in order to reduce the ROM storage size required. Finally, a method for reducing the accumulation of product roundoff errors in stored-product digital filters using an error cancellation technique is given.

ACKNOWLEDGMENT

The author wishes to thank his thesis supervisor, Professor Willem Steenaart, for his guidance, inspiration and continuous encouragement throughout this work.

The author also gratefully acknowledges the funding of this work by the National Research Council and the University of Ottawa.

Perhaps the deepest appreciation and thanks should go to the author's family for their patience and moral support throughout this work. To my wife, Rita Susanto, and to my parents, for all their help I am especially grateful.

TABLE OF CONTENTS

Abstract	iii
Acknowledgment	iv
Chapter I: INTRODUCTION	1
General	1
Summary	3
Chapter II: STORED-PRODUCT MULTIPLICATION TECHNIQUE	5
Introduction	5
Concept	6
Error Analysis for Fixed Point Arithmetic	9
Error analysis for floating point arithmetic	15
ROM Size Requirement	21
Speed Considerations	27
Chapter III: STORED-PRODUCT DIGITAL FILTERING	29
Introduction	29
Coefficient Accuracy	30
Roundoff Error Accumulation	41
Proposed Structure for Stored-Product Digital Filters	48
Chapter IV: A COMPARISON OF STORED-PRODUCT AND WAVE DIGITAL FILTERS	51
Introduction	51
Wave Digital Filters	52
General Principles	52
Digital Realization of Analog Elements.	54
3-Port Adaptors	58
Design Example	65
Frequency Response Accuracy in Wave and Stored-Product Digital Filters	69
Roundoff Error Accumulation in Wave and Stored-Product Digital Filters	73
Hardware Realization and Speed Consideration	77

Chapter V: STORED-PRODUCT DIGITAL FILTERS WITH ERROR SPECTRUM SHAPING	83
Introduction	83
Approach	84
Analysis of The Error Cancelling Technique	88
Architectures For SPDFs With Error Spectrum Shaping	92
Chapter VI: CONCLUSIONS	96
Appendix A: COEFFICIENTS OF THE DIRECT FORM DIGITAL FILTERS	98
Appendix B: COEFFICIENTS OF THE WAVE DIGITAL FILTERS	99
BIBLIOGRAPHY	100

LIST OF FIGURES

2.1	Design concept of stored-product ROM multipliers.	7
2.2	Stored-product ROM multipliers.	8
2.3	Fixed point multiplication using a conventional digital multiplier.	10
2.4	Variance of output error for fixed point multiplication.	14
2.5	Floating point multiplication using a conventional digital multiplier.	16
2.6	Variance of relative output error for floating point multiplication.	20
2.7	Parallel architecture of bit-slice stored-product ROM multiplier.	24
2.8	Sequential architecture of bit-slice stored-product ROM multiplier.	25
3.1	"Direct form 1" realization of an N-th order digital filter.	31
3.2	Frequency response of a 3-rd order Chebyshev LPF.	35
3.3	Frequency response of a 5-th order Chebyshev LPF.	36
3.4	Frequency response of a 7-th order Chebyshev LPF.	37
3.5	RMS magnitude error of a 3-rd order direct filter.	38
3.6	RMS magnitude error of a 5-th order direct filter.	39

3.7	RMS magnitude error of a 7-th order direct filter.	40
3.8	Technique for measuring roundoff error accumulation in SPDFs.	44
3.9	Average output error of a 3-rd order direct filter.	45
3.10	Average output error of a 5-th order direct filter.	46
3.11	Average output error of a 7-th order direct filter.	47
3.12	"Direct form 2" realization of an N-th order direct filter.	49
3.13	Proposed structure for stored-product digital filters.	50
4.1	Analog n-port network.	53
4.2	Digital realization of analog elements.	57
4.3	Type P2 Adaptor.	61
4.4	Type S2 Adaptor.	62
4.5	Type P1 Adaptor.	63
4.6	Type S1 Adaptor.	64
4.7	Conceptual design of wave digital filter.	68
4.8	RMS magnitude error in the 3-rd order WDF and SPDF.	70
4.9	RMS magnitude error in the 5-th order WDF and SPDF.	71
4.10	RMS magnitude error in the 7-th order WDF and SPDF.	72
4.11	Average output error in the 3-rd order WDF and SPDF.	74
4.12	Average output error in the 5-th order WDF and SPDF.	75
4.13	Average output error in the 7-th order WDF and SPDF.	76
4.14	Block diagram of the 3-rd order wave digital filter.	78
4.15	Block diagram of the 7-th order wave digital filter.	79
4.16	Speed comparison for wave and stored-product digital filters.	82
5.1	(a) Recursive section of an N-th order digital filter.	86
5.2	(a) Block diagram of the filter with error cancelling technique.	87
5.3	Parallel architecture of a high speed SPDF with ESS.	93
5.4	Sequential architecture of a high speed SPDF with ESS.	94

Chapter I

INTRODUCTION

1.1 General

Digital filtering is the process of spectral shaping by means of computational algorithms. A digital filter can be constructed using digital hardware such as adders, multipliers and shift registers, or can be simulated on a general purpose computer.

Despite the many advantages offered by digital filters, there is an inherent limitation on the accuracy of the filters due to finite wordlength constraints in the implementation of the filters. Digital filters with coefficients obtained directly from the transfer functions are sensitive to coefficient quantization. Although filter structures with low sensitivity to coefficient quantization, such as the wave digital filters [1]-[6] have been developed, this particular solution trades off filtering speed and hardware complexity for increased accuracy. Wave digital filter structures are highly recursive and use a large number of adders, thus, requiring a relatively large number of arithmetic operations. Subsequently, the speed of wave digital filters is low. Therefore, it is of interest to find alternatives for achieving fast and accurate digital filters.

Basically, the source of inaccuracy in a digital filter lies in the coefficient wordlength restriction, resulting in predictable deviations from the ideal filter characteristics, as well as contributing to the probabilistic roundoff error accumulation in the output signals. Thus, it is possible to increase the accuracy of a digital filter characteristic and reduce the

roundoff error accumulation by increasing the wordlength of the multiplier coefficients used.

The simplest way of increasing a digital filter's accuracy is by use of stored-product ROMs replacing hardware multipliers [12]-[17]. In this technique, the filter's coefficients can be implemented with very large wordlength as they appear only indirectly in each ROM. Therefore, the coefficient roundoff errors are minimized to arbitrarily low values, and errors in the products from each multiplier are reduced also. Another advantage obtainable from this technique comes from the availability of a faster multiplication rate of the stored-product ROM in parallel arithmetic mode. A 16-bit stored-product digital filter realized in the transposed configuration of "direct form 2" has been reported capable to operate with sampling frequency in the range of 10 MHz [15].

The use of ROM elements in digital filters has been introduced in distributed arithmetic filters [10]. These filters generally process the input signal serially, so that the filtering speed is limited by the signal wordlength. Although distributed arithmetic filters can be used also for parallel processing, the limitation in filtering speed still exists due to the propagation of the output signal through an adder tree of $\log_2 B$ stages, where B is the signal wordlength. On the other hand, stored-product digital filters are designed for parallel processing, and the length of the sequence of arithmetic operations per clock cycle can be minimized so that the filtering speed is high.

In the process of circuit integration, the necessary ROM storage size can be reduced substantially through the use of a bit-slice technique [13][16] so that the implementation cost is reduced as well. The many advantages (high coefficient accuracy, potential high speed, minimal hardware complexity and low implementation cost) offered by the stored-product technique make stored-product digital filters attractive.

The performance of stored-product digital filters has not been compared previously to that obtained from wave digital filters. The major objective in undertaking this research work is to make a comparison in term of filter's accuracy, speed and hardware requirement. Also, since the stored-product digital filtering, which minimizes coefficient roundoff errors, only reduces product roundoff error accumulation slightly, a technique for obtaining further reduction in product roundoff error accumulation in digital filters is explored as well.

1.2 Summary

This thesis deals with the analysis of errors in stored-product ROM multipliers and in recursive stored-product digital filters. For the stored-product ROM multipliers, the error analysis is carried out for fixed point arithmetic as well as for floating point arithmetic. In either case, it will be shown that the stored-product multiplication technique improves the accuracy of the products obtained. Since the fixed point arithmetic mode is the most commonly used in real time application, only the evaluation for fixed point stored-product digital filters is presented.

In chapter II, the concept of the stored-product multiplication technique is presented. An error analysis is carried out, and the results indicate that lower product error is achieved by increasing the coefficient wordlength. Methods for reducing the ROM storage size required are discussed, and the sequential architecture of bit-slice stored-product ROM multipliers is proposed, taking the advantage of the fast ROM access time.

Chapter III is devoted to the design and analysis of recursive stored-product digital filters. The frequency response accuracy and product roundoff error accumulation are evaluated theoretically and by computer simulation, and compared to that obtained from the fil-

ters with shorter coefficient wordlength. The realization for high speed filtering is discussed.

In chapter IV, a performance comparison for stored-product and wave digital filters is presented. This includes the deviations from the ideal filter characteristics, the roundoff error accumulation in the output signals, the hardware requirement and the filtering speed of the two filter structures. Again, the filtering performance is carried out by means of a computer simulation.

In chapter V, an error cancelling technique to obtain further reduction in product roundoff error accumulation in stored-product digital filters is proposed. Finally, the conclusions is given in chapter VI.

Chapter II

STORED-PRODUCT MULTIPLICATION TECHNIQUE

2.1 Introduction

The basis of the stored-product multiplication technique is the use of a Read Only Memory (ROM) as a look up table to obtain the product of two numbers. As described in [12]-[17], this technique, when used to obtain the product of the multiplication of a digital signal and a constant coefficient, gives a more accurate result than that available from conventional digital multipliers. In addition, this technique offers higher speed and less hardware complexity.

An analysis is given of the accuracy of the stored-product ROM multiplier for fixed point arithmetic, as well as for floating point arithmetic. It will be shown that the overall error in the product obtainable using this technique is reduced as the coefficient can be implemented with largewordlength. The only source of error in this technique is the product quantization error. The results obtained from computer simulations confirm this analysis.

From the hardware point of view, the stored-product ROM multiplier realization cost is dependent on the ROM storage size. It can be formulated as a function of the required accuracy (the product wordlength) and of the input signal wordlength. In the case of an input signal with large wordlength, it may be necessary to reduce the ROM size required in order to reduce the related cost. This task can be accomplished by two alternatives, the nonlinear code conversion technique and the bit-slice technique.

2.2 Concept

A look-up table ROM multiplier requires values of the multiplicand and the multiplier to address the corresponding product. However, when the multiplier value is constant, only the multiplicand value is needed to address the ROM. A stored-product ROM multiplier is generally applied in digital filters with constant coefficients. For applications where both signal and coefficient vary, the use of stored-product ROM multiplier is feasible, requiring large ROM sizes.

The design stage of the stored-product ROM multiplier with constant coefficient is shown in Fig. 2.1. It requires initial computation of all possible products of the signal and the given coefficient with high accuracy using a general purpose computer. Assuming that the signal is of wordlength L_s bits, there will be 2^{L_s} possible signal values. The coefficient is of wordlength L_c bits ($L_c \gg L_s$), and each generated product is rounded into L_p bits and stored in a ROM using the rounded signal value as its address. The coefficient wordlength L_c can be as large as the double precision wordlength of the computer used, so that the products stored in the ROM are based on a near infinite precision coefficient, thus essentially free from coefficient roundoff error.

At the table look-up stage, the product is retrieved by addressing the ROM with the corresponding value of the input signal. This technique is illustrated in Fig. 2.2.

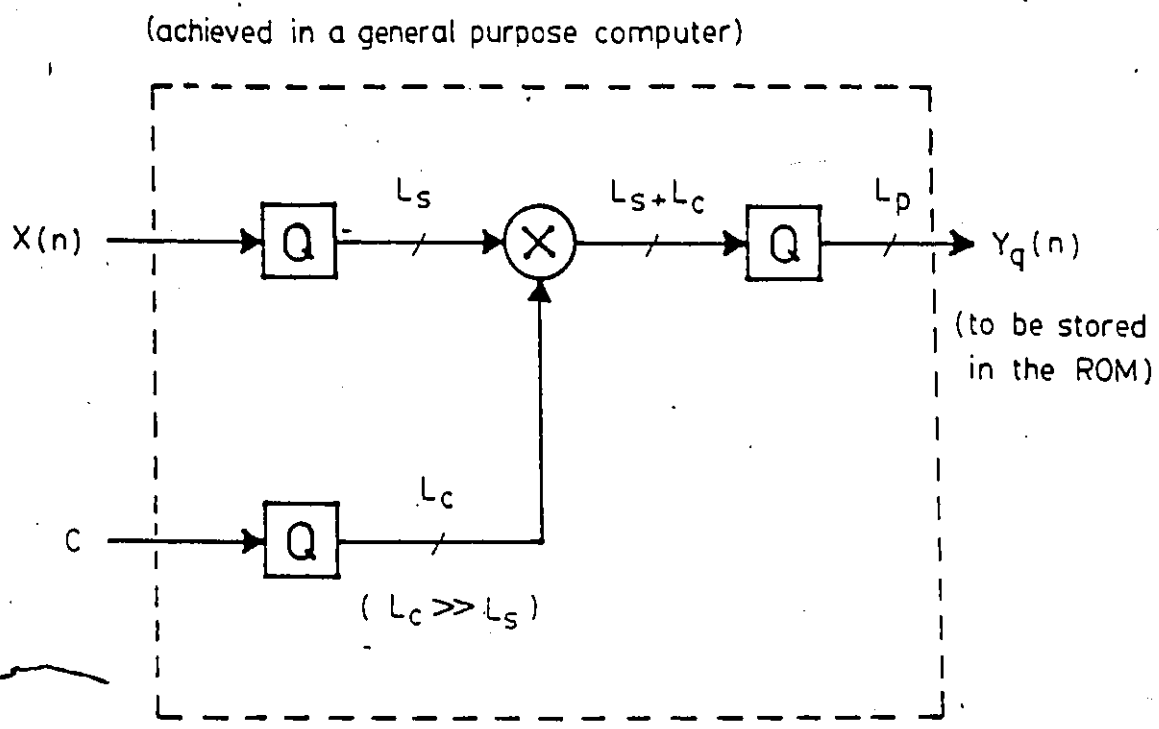


Figure 2.1: Design concept of stored-product ROM multipliers.

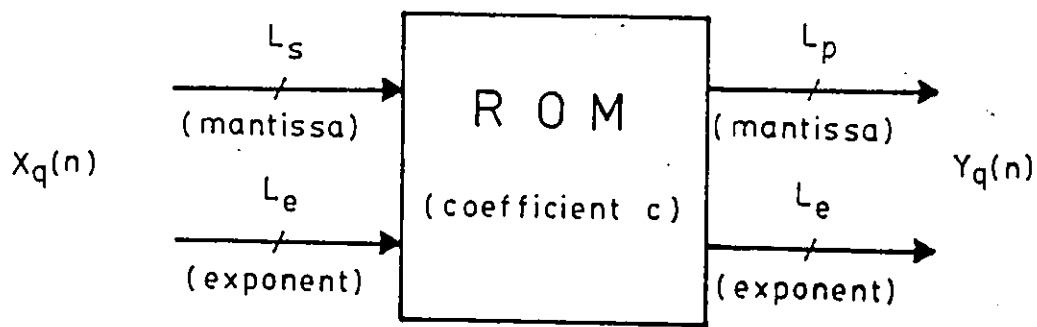
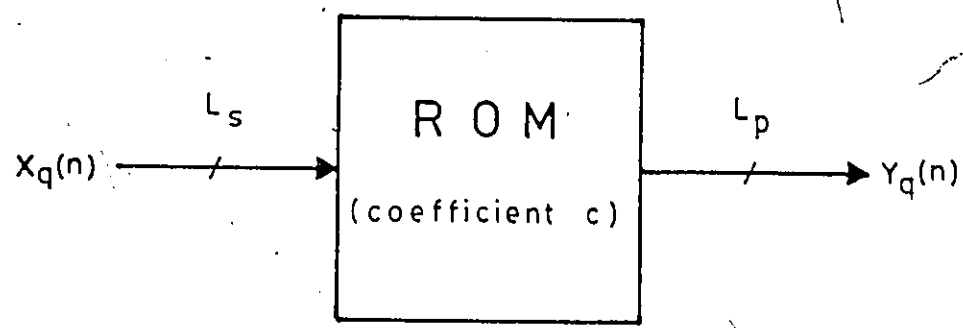


Figure 2.2: Stored-product ROM multipliers. (a) For fixed point arithmetic. (b) For floating point arithmetic.

2.3 Error Analysis for Fixed Point Arithmetic

Consider a multiplication between a signal $x(n)$ and a constant coefficient c using fixed point arithmetic. For infinite precision multiplication, the product or the output signal is

$$y(n) = c \cdot x(n) \quad (2.1)$$

When using a conventional hardware multiplier as shown in Fig. 2.3, because of wordlength limitation, the signal, the coefficient and the product are quantized. Assuming that they are rounded to L_s , L_c and L_p bits respectively, excluding the sign bit, the actual product obtained from the hardware multiplier will be

$$\begin{aligned} y_M(n) &= \{c + e_c\} \{x(n) + e_x(n)\} + e_p(n) \\ &= y(n) + e_M(n) \end{aligned} \quad (2.2)$$

where $e_x(n)$, e_c and $e_p(n)$ are the roundoff errors of the signal, the coefficient and the product respectively, and are bounded by

$$\begin{aligned} -\frac{1}{2} \cdot 2^{-L_s} &\leq e_x(n) \leq \frac{1}{2} \cdot 2^{-L_s} \\ -\frac{1}{2} \cdot 2^{-L_c} &\leq e_c \leq \frac{1}{2} \cdot 2^{-L_c} \\ -\frac{1}{2} \cdot 2^{-L_p} &\leq e_p(n) \leq \frac{1}{2} \cdot 2^{-L_p} \end{aligned} \quad (2.3)$$

The overall multiplier output error $e_M(n)$, after neglecting the second order error terms, is given by

$$e_M(n) = c \cdot e_x(n) + e_c x(n) + e_p(n) \quad (2.4)$$

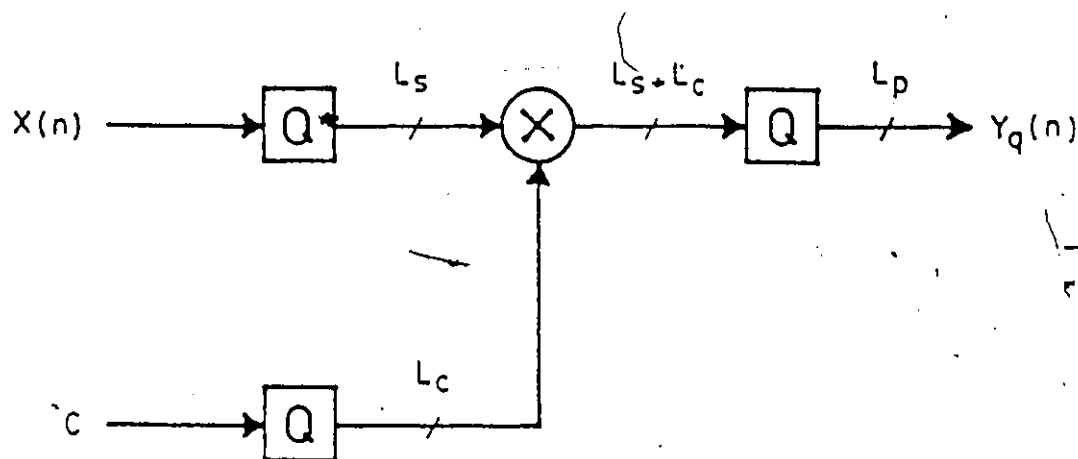


Figure 2.3: Fixed point multiplication using a conventional digital multiplier.

Notice that, since the coefficient c is a constant, the coefficient roundoff error e_c is deterministic, while the signal and the product roundoff errors, $e_x(n)$ and $e_p(n)$, are probabilistic. However, the coefficient c can have any possible value, thus, this analysis can be generalized by considering the coefficient roundoff error e_c as being probabilistic with a uniform probability density function [14]. By assuming that these roundoff errors are uncorrelated to the signal and the coefficient, the variance of the overall output error will be

$$\sigma_{e_M}^2 = \sigma_c^2 \sigma_{e_x}^2 + \sigma_x^2 \sigma_{e_c}^2 + \sigma_{e_p}^2 \quad (2.5)$$

If the multiplication is carried out using the stored-product ROM technique as illustrated in Fig. 2.2-a, the coefficient roundoff error is arbitrarily small and can be neglected. The actual product obtained from the stored-product ROM can be expressed as

$$\begin{aligned} y_R(n) &= c\{x(n) + e_x(n)\} + e_p(n) \\ &= y(n) + e_R(n) \end{aligned} \quad (2.6)$$

with the overall output error given by

$$e_R(n) = c \cdot e_x(n) + e_p(n) \quad (2.7)$$

Therefore, the variance of output error of the stored-product ROM will be

$$\sigma_{e_R}^2 = \sigma_c^2 \sigma_{e_x}^2 + \sigma_{e_p}^2 \quad (2.8)$$

From (2.5) and (2.8), it can be seen that the overall output error in both cases contains roundoff error due to the rounded input signal, which is equal for either the stored-product ROM or the hardware multiplier with short coefficient wordlength. Apart from the input signal roundoff, the overall error in the product from a stored-product ROM is due to product roundoff only (2.8), while the overall error in a hardware multiplier with short coefficient wordlength is due to coefficient and product roundoff errors and is also dependent on the input signal power (2.5).

The effect of the input signal roundoff to the actual product can be dealt with as if it contributes a known noise power due to the analog to digital conversion. Thus, assuming that the input signal $x(n)$ of wordlength L_s is accurate [14], the variance of overall output error introduced by the hardware multiplier with short coefficient wordlength is

$$\sigma_{e_M}^2 = \sigma_x^2 \sigma_{e_c}^2 + \sigma_{e_p}^2 \quad (2.9)$$

$$= \sigma_x^2 \frac{2^{-2L_c}}{12} + \frac{2^{-2L_p}}{12}$$

and the variance of output error introduced by the stored-product ROM is

$$\begin{aligned} \sigma_{e_R}^2 &= \sigma_{e_p}^2 \\ &= \frac{2^{-2L_p}}{12} \end{aligned} \quad (2.10)$$

Shown in Fig. 2.4 is the plot of the variance of output errors available from a conventional hardware multiplier and from a stored-product ROM, where the input signal $x(n)$ with wordlength L_s bits is assumed to be without error. The results from computer simulation are plotted as well, where the input signal used was a random sequence, uniformly distributed within $[-1.0, 1.0]$. It can be seen that the output error for the stored-product ROM (represented by the curve for $L_c = 30$) is smaller than that available for the hardware multiplier. Here, the output error of the stored-product ROM is dependent on the choice of product wordlength L_p , where an increase in L_p by one bit gives a decrease of 6 dB in the output noise power. On the other hand, the output error of the hardware multiplier is dependent on both the coefficient and product wordlengths, and nothing will be gained by increasing L_p alone as the overall error will then be controlled by the coefficient roundoff error.

It can be seen also that, the output error of the hardware multiplier is influenced by the input signal power. When the input signal power is $\sigma_x^2 = 1/3$, the output error of the hardware multiplier is 1.25 dB larger than that of the stored-product ROM, and this value decreases to 0.35 dB when the input signal power is reduced to $\sigma_x^2 = 1/12$. Although the output error of the hardware multiplier is small when the input signal power is reduced, the output error of the stored-product ROM is still smaller than that of the hardware multiplier. In conclusion, the stored-product ROM multiplier gives more accurate products than the digital multiplier due to its use of coefficients with very long wordlength.

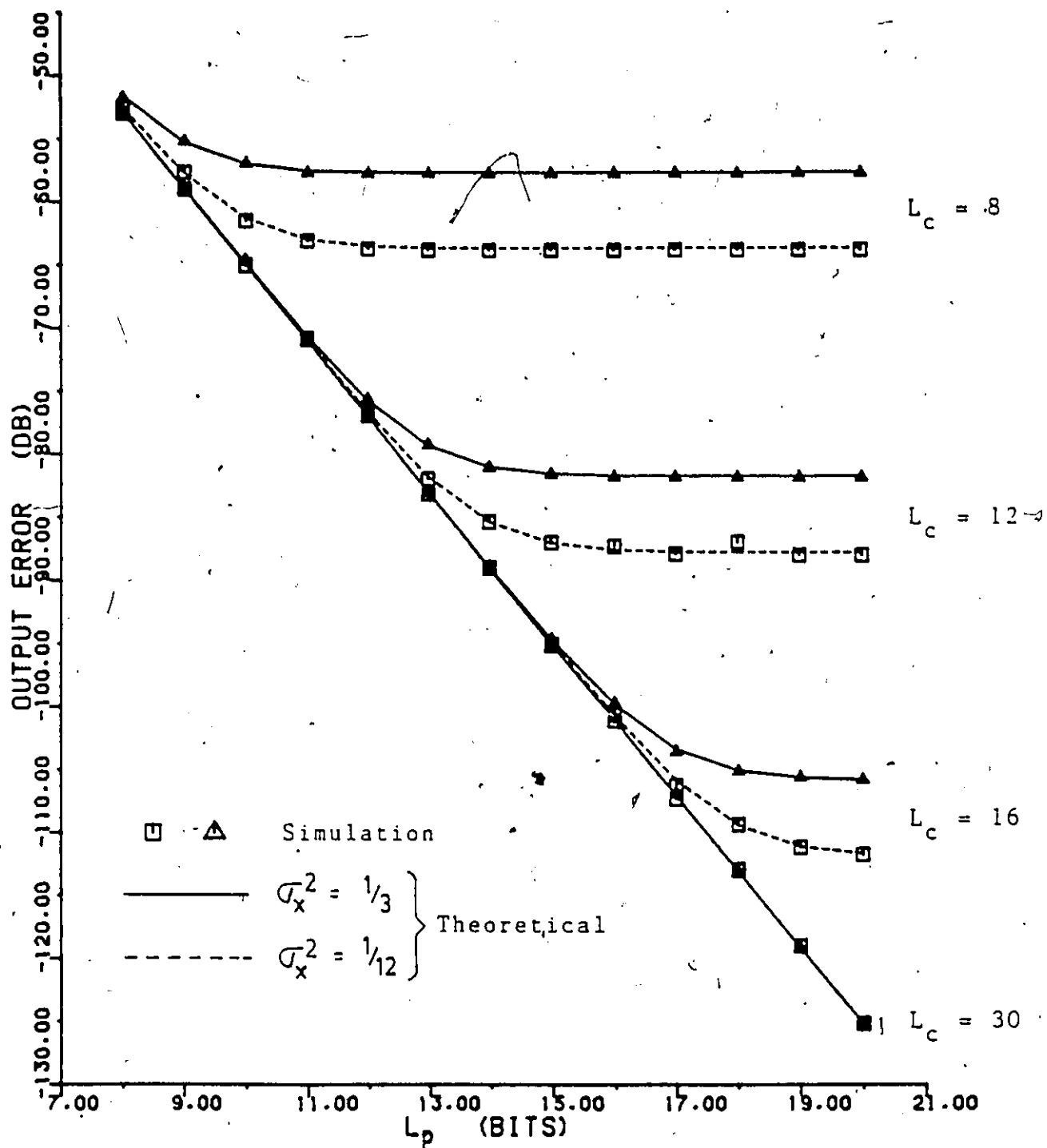


Figure 2.4. Variance of output error for fixed point multiplication.

2.4 Error analysis for floating point arithmetic

In floating point systems, a number is represented by two fixed point numbers, the mantissa and the exponent, where the magnitude of the mantissa is normalized to the range

$$0.5 \leq |m| < 1.0 \quad (2.11)$$

and the floating point binary number x is obtained as the product of the mantissa m with the given base raised to the power denoted by the exponent e , that is

$$x = m \cdot 2^e \quad (2.12)$$

Since number quantization in floating point systems occurs in the mantissa [21], it is more convenient to express the resulting roundoff error in terms of the relative roundoff error, which is defined as

$$\epsilon_x = \frac{\Delta x}{x} = \frac{x_q - x}{x} \quad (2.13)$$

so that a rounded floating point number can be expressed as

$$x_q = x \cdot (1 + \epsilon_x) \quad (2.14)$$

where the relative roundoff error ϵ_x due to rounding the mantissa to B bits is bounded by

$$-2^{-B} \leq \epsilon_x \leq 2^{-B} \quad (2.15)$$

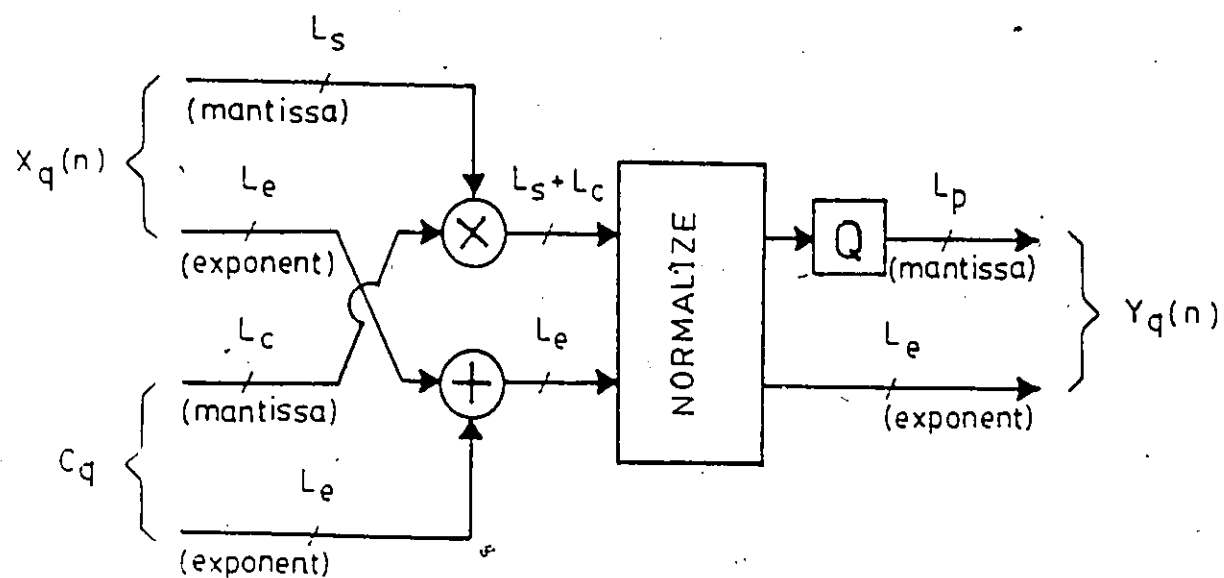


Figure 2.5: Floating point multiplication using a conventional digital multiplier.

In the following, consider the multiplication between the signal $x(n)$ and the coefficient c using a conventional hardware multiplier as shown in Fig. 2.5. It is assumed that all quantization is by rounding, and the product is normalized prior to rounding. The product obtained from this multiplier can be expressed as

$$\begin{aligned}
 y_M(n) &= [c_q \cdot x_q(n)]_q \\
 &= [c \cdot (1 + \epsilon_c) \cdot x(n) \cdot \{1 + \epsilon_x(n)\}] [1 + \epsilon_p(n)] \quad (2.16) \\
 &= y(n) \{1 + \epsilon_M(n)\}
 \end{aligned}$$

where the relative roundoff errors $\epsilon_x(n)$, $\epsilon_p(n)$ and ϵ_c are bounded by

$$\begin{aligned}
 -2^{-L_s} &\leq \epsilon_x(n) \leq 2^{-L_s} \\
 -2^{-L_p} &\leq \epsilon_p(n) \leq 2^{-L_p} \quad (2.17) \\
 -2^{-L_c} &\leq \epsilon_c \leq 2^{-L_c}
 \end{aligned}$$

The relative output error $\epsilon_M(n)$, after neglecting the second order error terms, is given by

$$\epsilon_M(n) = \epsilon_x(n) + \epsilon_p(n) + \epsilon_c \quad (2.18)$$

It follows that the variance of relative output error is

$$\sigma_{\epsilon_M}^2 = \sigma_{\epsilon_x}^2 + \sigma_{\epsilon_p}^2 + \sigma_{\epsilon_c}^2 \quad (2.19)$$

where the usual assumptions regarding the statistical independence of the signal and roundoff errors have been made.

If the multiplication is carried out using the stored-product ROM technique as shown in Fig. 2.2-b, the resulting product will be

$$\begin{aligned}
 y_R(n) &= [c \cdot x_q(n)]_q \\
 &= [c \cdot x(n) \{1 + \epsilon_x(n)\}] [1 + \epsilon_p(n)] \\
 &= y(n) \{1 + \epsilon_R(n)\}
 \end{aligned} \tag{2.20}$$

with the relative output error given by

$$\epsilon_R(n) = \epsilon_x(n) + \epsilon_p(n) \tag{2.21}$$

so that the variance of relative output error will be

$$\sigma_{\epsilon_R}^2 = \sigma_{\epsilon_x}^2 + \sigma_{\epsilon_p}^2 \tag{2.22}$$

Again, it can be seen from (2.19) and (2.22), the effect of input signal roundoff can be considered as if it contributes a known relative error at the multiplier output. Assuming that the input signal $x(n)$ with the mantissa of wordlength L_s is accurate, the variance of the relative output error obtainable from the hardware multiplier will be

$$\begin{aligned}
 \sigma_{\epsilon_M}^2 &= \sigma_{\epsilon_c}^2 + \sigma_{\epsilon_p}^2 \\
 &= \frac{2^{-2L_c}}{3} + \frac{2^{-2L_p}}{3}
 \end{aligned} \tag{2.23}$$

and, the variance of the relative output error obtainable from the stored-product ROM will be

$$\begin{aligned}\sigma_{\epsilon_R}^2 &= \sigma_{\epsilon_P}^2 \\ &= \frac{2^{-2L_P}}{3}\end{aligned}\quad (2.24)$$

Thus, in floating point arithmetic, the stored-product ROM multiplier also gives more accurate results than those obtained using conventional hardware multipliers.

Shown in Fig. 2.6 is the plot of the variance of the relative output error for the stored-product ROM and the hardware multiplier where the input signal $x(n)$ is assumed to be accurate. It can be seen that, for $L_C = L_P$, the relative output error obtainable from the stored product ROM is 3 dB smaller than that from the conventional hardware multiplier and is independent of the input signal power.

The relative output error of the stored-product ROM is entirely controllable by the choice of the product wordlength, which is one of the advantages offered by the stored-product ROM technique. Computer simulation results confirm those expressed by (2.23) and (2.24).

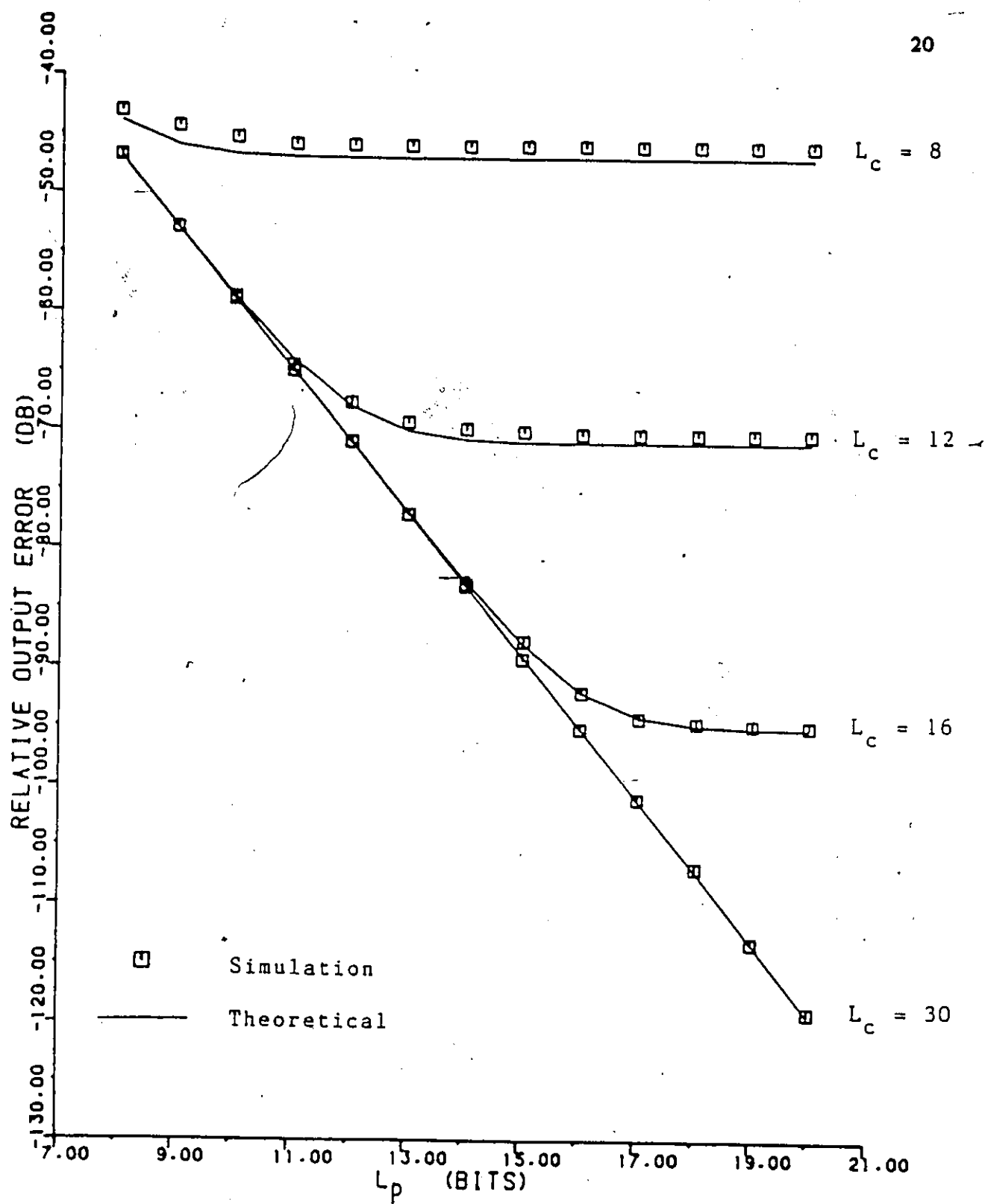


Figure 2.6: Variance of relative output error for floating point multiplication.

2.5 ROM Size Requirement

It has been pointed out that the stored-product multiplication technique requires storing all possible products in a ROM with the input signal as the address code. Analysis shows that the number of bits to be stored in the ROM for an input signal with wordlength of up to 12 bits is still manageable at reasonable cost. However, for signals with wordlength larger than 12 bits, the necessary ROM size will be large, so that the direct implementation as shown in Fig. 2.2 is no longer suitable.

ROM size requirement is dependent on the input signal wordlength and on the choice of product wordlength (the required accuracy). It can be formulated as

$$C = 2^{L_s} \cdot L_p \text{ bits} \quad (2.25)$$

For input signal and product wordlength of 16 bits, the necessary ROM storage size will be $2^{16} \times 16 = 1 \text{ M bits}$, which is very large. Therefore, it is imperative to explore ways of reducing the required ROM size in order to minimize the multiplier realization cost.

A first alternative to handle a signal with large wordlength is to use a nonlinear quantization technique [12][13], thus the signal is represented by fewer bits per sample than in the linear quantization technique. This technique reduces the memory capacity requirement, and allows the multiplier to be used directly for applications involving nonlinearly quantized signals, such as PCM signals, thus gives additional hardware advantage.

Another alternative is to use a bit-slice technique, which can be implemented either with the parallel architecture or the split address technique [13][15], or with the sequential architecture [16].

In the bit-slice technique, the input signal wordlength, and hence the ROM address code is divided into two or more parts. For example, consider a signal $x(n)$ with wordlength L_s , which can be expressed as

$$x(n) = \sum_{k=1}^{L_s} b_k 2^{-k} \quad (2.26)$$

where b_k is either 0 or 1. Alternatively, it can be expressed as

$$x(n) = x_{msb}(n) + x_{lsb}(n) \quad (2.27)$$

where $x_{msb}(n)$ and $x_{lsb}(n)$ are the most significant bits and the least significant bits respectively. That is

$$\begin{aligned} x_{msb}(n) &= \sum_{k=1}^{L_s/2} b_k 2^{-k} \\ &= x_1(n) \end{aligned} \quad (2.28)$$

$$\begin{aligned} x_{lsb}(n) &= \sum_{k=1}^{L_s/2} c_k 2^{-k} \cdot 2^{-L_s/2} \\ &= x_2(n) \cdot 2^{-L_s/2} \end{aligned} \quad (2.29)$$

so that, the product of $x(n)$ and a constant c can be written as

$$\begin{aligned} y(n) &= c \cdot x(n) \\ &= c \cdot x_1(n) + c \cdot x_2(n) \cdot 2^{-L_s/2} \end{aligned} \quad (2.30)$$

Thus, the product $y(n)$ can be obtained through the use of two stored-product ROMs, where each ROM is addressed by $L_s/2$ bit codes. Since the number of bits to be stored is largely determined by the address wordlength, therefore, the necessary ROM size is reduced substantially.

Shown in Fig. 2.7 is the realization of (2.30) in parallel architecture, requiring two ROMs and one parallel adder, which is also known as the split address technique. Notice that, from the ROM addressed by the least significant bits, only $(L_p - L_s/2)$ bits are needed to obtain the actual product. Therefore, the necessary ROM storage size will be

$$C' = 2^{L_s/2} (2L_p - L_s/2) \text{ bits} \quad (2.31)$$

Since, the contents of the two ROMs are related to each other by a factor of $2^{L_s/2}$, therefore, they can be time-shared. This forms the basis for the sequential architecture of the realization of (2.30). As shown in Fig. 2.8, there is only one ROM is used, and the product is obtained through two sequences of operations. During the first cycle, the ROM is addressed by the least significant bits, and its output will be saved in the latch and right-shifted by $L_s/2$ bits. During the second cycle, the ROM is addressed by the most significant bits, and the ROM's output is added with the data previously saved in the latch to give the actual product. Therefore, by time-sharing the same ROM, the memory capacity required is reduced to

$$C'' = 2^{L_s/2} \cdot L_p \text{ bits} \quad (2.32)$$

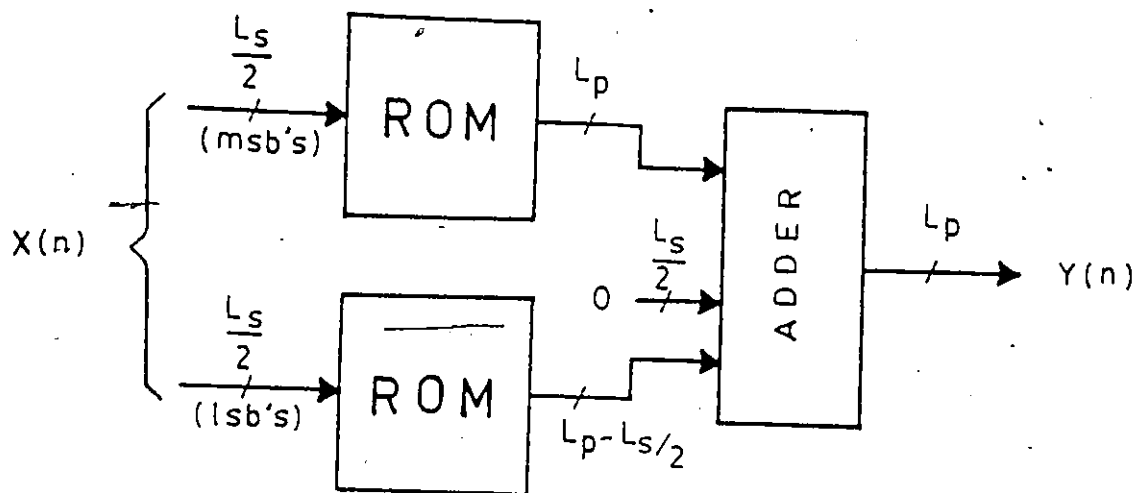


Figure 2.7: Parallel architecture of bit-slice stored-product ROM multiplier.

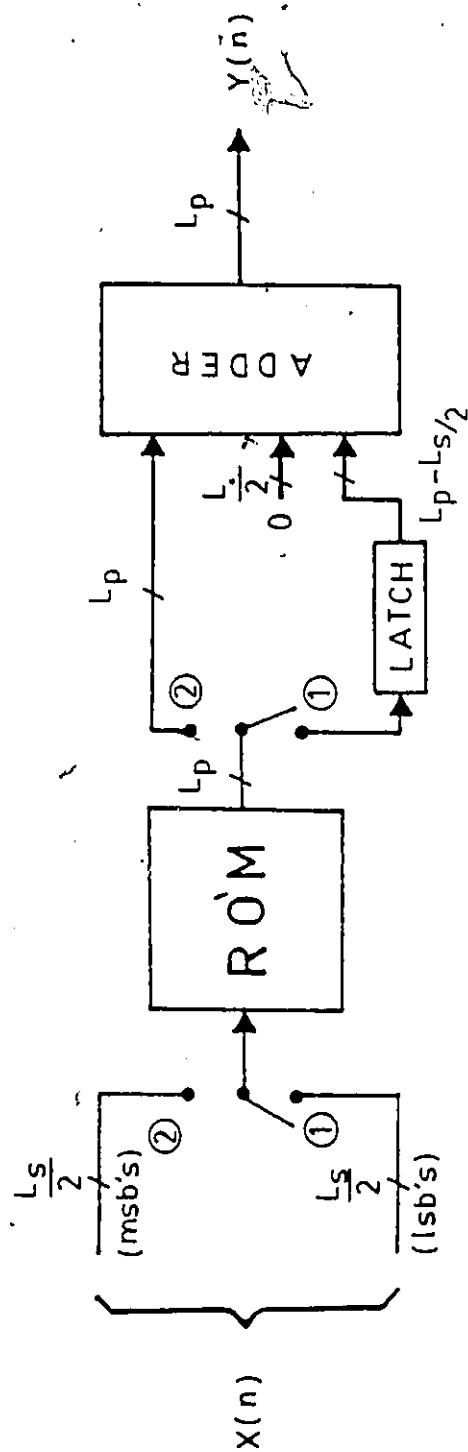


Figure 2.8: Sequential architecture of bit-slice stored-product ROM multiplier.

A comparison of the ROM size requirement for the direct implementation and the bit-slice implementation is given in Table 2.1. It can be seen that the bit-slice technique reduces the number of bits to be stored in a ROM substantially. In principle, the ROM size requirement can be reduced further by increasing the number of slices. However, this requires the use of more parallel adders, thus increases hardware complexity of the ROM multiplier. In order to minimize implementation cost, the number of slices used may not be too large. The fact that the application of the bit-slice technique affects the multiplication speed will be discussed in the next section.

L_s	DIRECT IMPLEMENTATION (BITS)	BIT-SLICE IMPLEMENTATION (BITS)	
		PARALLEL	SEQUENTIAL
8	2K	192	128
10	10K	480	320
12	49K	1.2K	768
14	229K	2.7K	1.2K
16	1M	6.1K	4.1K
24	402M	147K	98K
32	137G	3.1M	2M

Table 2.1: ROM size Requirement for $L_s = L_p$.

2.6 Speed Considerations

The multiplication speed of a stored-product ROM is dependent on the way it is implemented. In direct implementation, ROM is the only hardware used. Thus, the multiplication speed is entirely determined by the ROM access time, i.e.

$$T_M = T_{ROM} \quad (2.33)$$

On the other hand, the bit-slice implementation requires more hardware, and the multiplication speed is also dependent on the architecture used. In other words, it trades off implementation cost with speed. In the parallel architecture, the multiplication is achieved by addressing the two ROMs simultaneously and followed by an addition. The multiplication speed for the parallel architecture is given by

$$T'_M = T_{ROM} + T_{ADD} \quad (2.34)$$

where T_{ADD} is the time needed for the addition. In the sequential architecture the ROM is addressed sequentially twice, and it is followed by an addition. Therefore, the time required for the multiplication with this architecture is

$$T''_M = 2T_{ROM} + T_{LATCH} + T_{ADD} \quad (2.35)$$

where T_{LATCH} is the time needed to shift data into the latch.

In general, the multiplication speed for a bit-slice implementation is lower than for a direct implementation. However, small ROMs usually need less access time than large ROMs, so that the decrease in multiplication speed sometimes is not significant. Table 2.2

shows typical multiplication speed for direct implementation and bit-slice implementation of a stored-product ROM multiplier using bipolar ROMs and standard TTL devices.

L_s	DIRECT IMPLEMENTATION	BIT-SLICE IMPLEMENTATION	
		PARALLEL	SEQUENTIAL
12	50	71	115
16	65	83	127
24	85	122	181
32	105	171	245

Table 2.2 : Typical Multiplication Speed (in nano-second).

Chapter III.

STORED-PRODUCT DIGITAL FILTERING

3.1 Introduction

Thus far the stored-product multiplication technique has been shown to minimize the coefficient and product roundoff errors occurring in digital multiplication. Such advantages make this technique useful for implementing a digital filter where highly accurate coefficients are necessary.

In this chapter the implementation of a digital filter using stored-product ROM multipliers will be presented. The filter is realized in direct form and is implemented using fixed point arithmetic, assuming all quantization is by rounding and two's complement numbers are used for negative values. The resulting filter, also known as the stored-product digital filter (SPDF), is examined for various signal wordlengths and several filter orders. The evaluation is carried out theoretically and by computer simulation, using a random sequence with uniform distribution as the input signal. The result is compared with the performance based on the same realization with shorter coefficient wordlength. It will be shown that stored-product digital filters have very high frequency response accuracy. Furthermore, error in the output signal is reduced as well, due to the increase in coefficient accuracy. This result is similar to that obtained for the single stored-product ROM multiplier.

An additional advantage is the availability of high speed multiplication, allowing stored-product digital filters to be operated with very high throughput rates [15]. The design simplicity, the high arithmetic accuracy and the high speed are attractive features of stored-product digital filters.

3.2 Coefficient Accuracy

Due to the wordlength limitation, digital filters usually are implemented with quantized coefficients. However, the coefficient quantization causes a change in the pole and zero locations of the transfer function, which in effect causes error in the frequency response. If the filter has high Q poles, that is, poles that are close to the unit circle in the z-plane, instability may occur as one or more poles may move outside the unit circle due to coefficient errors.

In this section an analysis for the accuracy of coefficients in stored-product digital filters is given. Basically, this analysis is based on [18]-[20], where the effect of coefficient quantization is observed in the frequency response of the filter.

Consider a general N-th order digital filter shown in Fig. 3.1 with the input-output relationship given by the difference equation

$$y(n) = \sum_{k=0}^N a_k x(n-k) - \sum_{k=1}^N b_k y(n-k) \quad (3.1)$$

where a_k and b_k are the infinite precision coefficients. When the coefficients are quantized, the input-output relationship of the actual filter becomes

$$v(n) = \sum_{k=0}^N \tilde{a}_k x(n-k) - \sum_{k=1}^N \tilde{b}_k v(n-k) \quad (3.2)$$

$$v(n) = y(n) + e(n) \quad (3.3)$$

$$\tilde{a}_k = a_k + \alpha_k ; \quad \tilde{b}_k = b_k + \beta_k \quad (3.4)$$

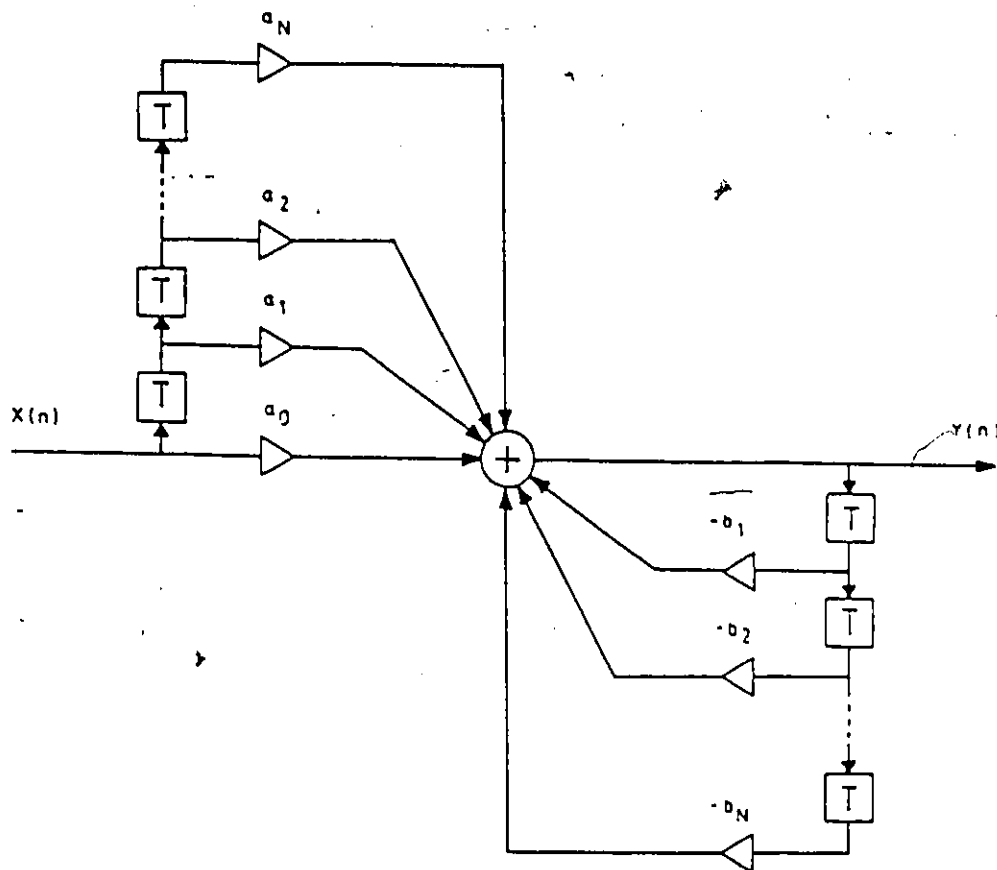


Figure 3.1: "Direct form 1" realization of an N-th order digital filter.

where $e(n)$ is the output error due to the coefficient quantization, and the quantities α_k and β_k are the coefficient roundoff errors. Note that only the coefficient error (the deterministic error) and its influence upon the filter magnitude characteristic is considered here. The probabilistic error accumulation in the the output signal due to product roundoff will be considered in section 3.3.

In the z-domain, equation (3.3) is written as

$$V(z) = Y(z) + E(z) \quad (3.5)$$

Upon dividing both sides by the z-transform of the input signal $X(z)$ we obtain the actual transfer function of the filter with quantized coefficients

$$\frac{V(z)}{X(z)} = \frac{Y(z)}{X(z)} + \frac{E(z)}{X(z)} \quad (3.6)$$

or

$$H(z) = H_{\infty}(z) + \varepsilon(z) \quad (3.7)$$

That is, the transfer function of a digital filter with quantized coefficients can be represented by the transfer function of the ideal filter in parallel with the stray transfer function $\varepsilon(z)$ based on the coefficient errors.

Notice that, from (3.1)-(3.4), the output error $e(n)$ can be expressed as

$$e(n) = \sum_{k=0}^N \alpha_k x(n-k) - \sum_{k=1}^N \beta_k y(n-k) - \sum_{k=1}^N b_k e(n-k) \quad (3.8)$$

in which the second order error terms have been neglected. Taking the z-transform of (3.8) and dividing both sides by $X(z)$ yields

$$\epsilon(z) = \frac{\alpha(z) - \beta(z)H_{\infty}(z)}{B(z)} \quad (3.9)$$

where

$$\alpha(z) = \sum_{k=0}^N \alpha_k z^{-k} \quad (3.10-a)$$

$$\beta(z) = \sum_{k=1}^N \beta_k z^{-k} \quad (3.10-b)$$

$$B(z) = 1 + \sum_{k=1}^N b_k z^{-k} \quad (3.10-c)$$

In stored-product digital filters, the coefficients are implemented with very large wordlength and their precision approaches the infinite precision coefficients, so that

$$\alpha(z) \approx 0$$

$$\beta(z) \approx 0$$

and subsequently the stray transfer function due to coefficient roundoff approaches zero:

$$\epsilon(z) \approx 0$$

and,

$$H_{SPDF}(z) \approx H_{\infty}(z)$$

As a result, the frequency response of a stored-product digital filter is very close to the frequency response in the original design.

To measure the frequency response accuracy, consider the root mean square deviations in the frequency response with respect to that of the original design, which is defined as

$$\text{RMS dev.} = \sqrt{\frac{1}{N} \sum_{k=1}^N \{ |H_{\infty}(f_k)| - |H(f_k)| \}^2} \quad (3.11)$$

where

$|H_{\infty}(f_k)|$ = magnitude response of the ideal filter at frequency f_k .

$|H(f_k)|$ = magnitude response of the filter with finite coefficient wordlength at frequency f_k .

N = Number of computations (frequencies f_k 's) taken within the passband.

Three digital low pass filters of different orders, derived from Chebyshev filters by means of the bilinear transformation, were evaluated using (3.11) for various coefficient wordlengths L_c . It can be seen from Fig. 3.2 to 3.4, the deviations in frequency response for stored-product digital filters, which can be represented by the filter with $L_c = 30$ bits, are very small. These results indicate that the SPDF design technique yields digital filters with frequency responses that approach the ideal response for the original design. Fur-

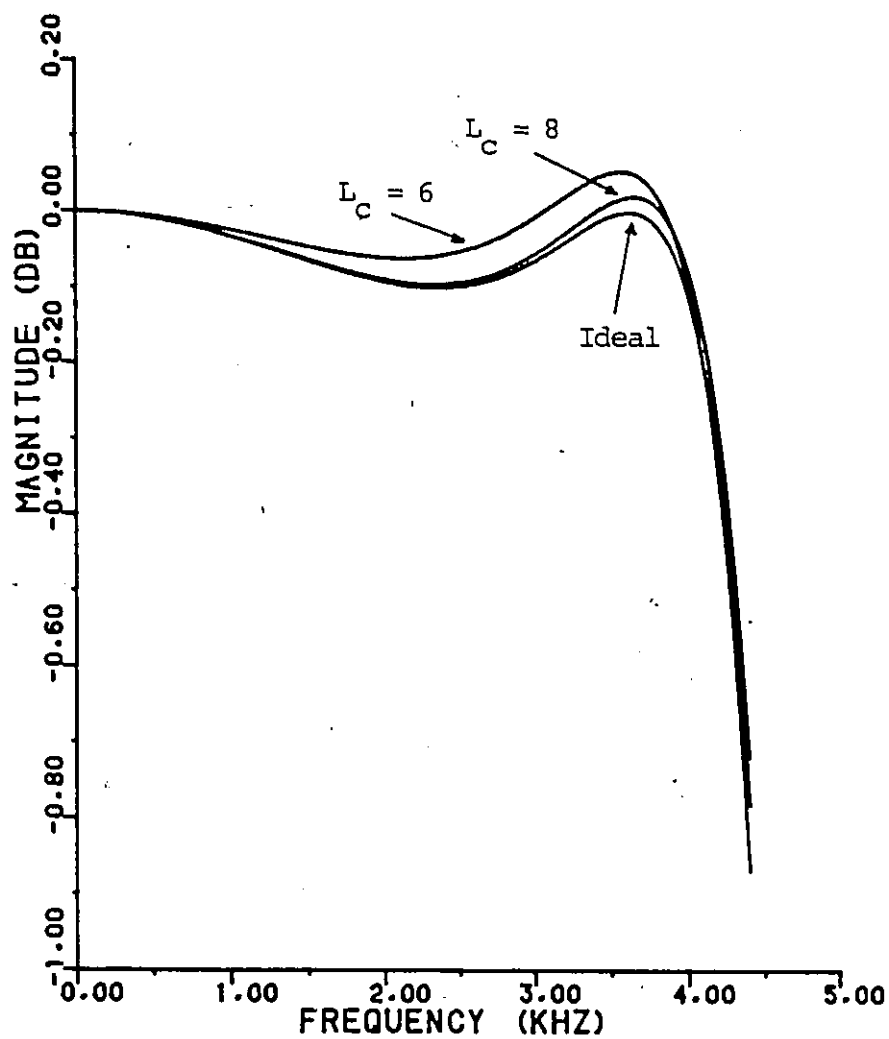


Figure 3.2: Frequency response of a 3-rd order Chebyshev LPF.

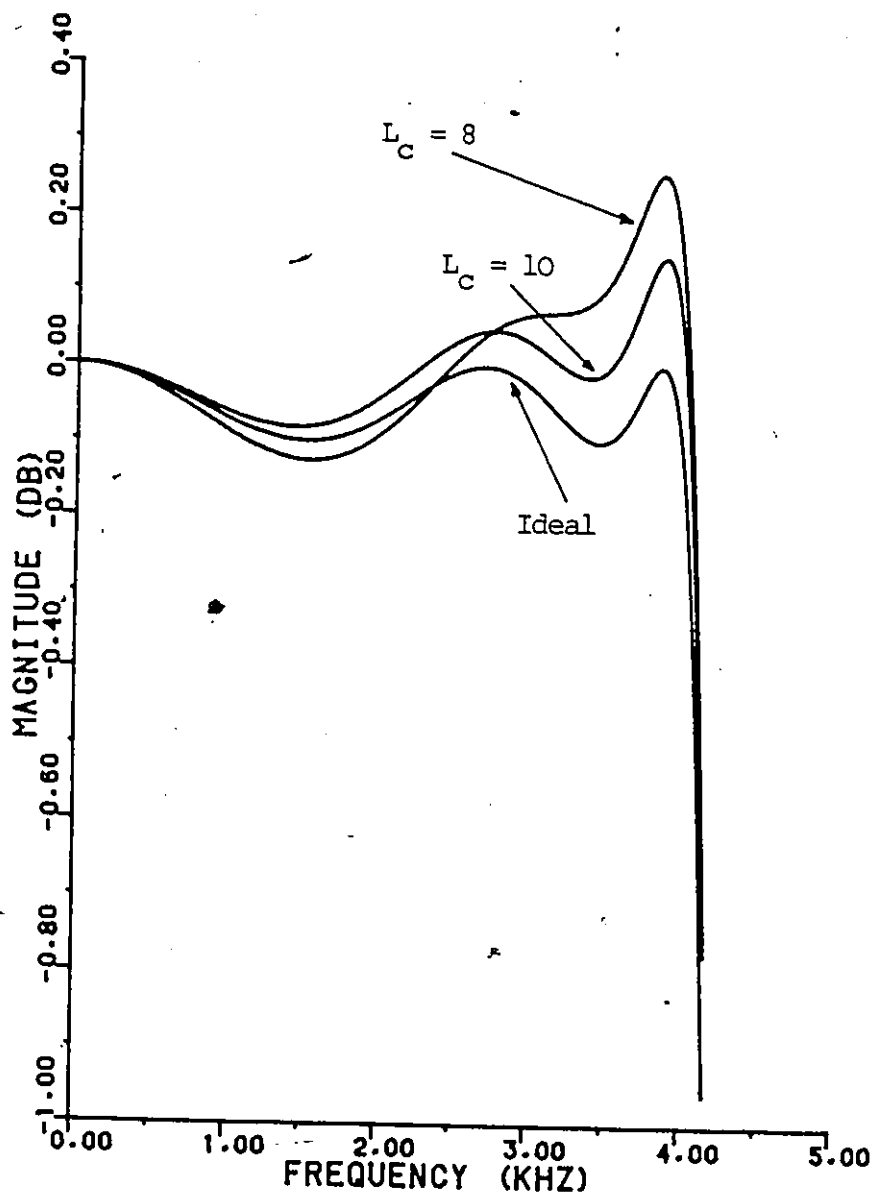


Figure 3.3° Frequency response of a 5-th order Chebyshev LPF.

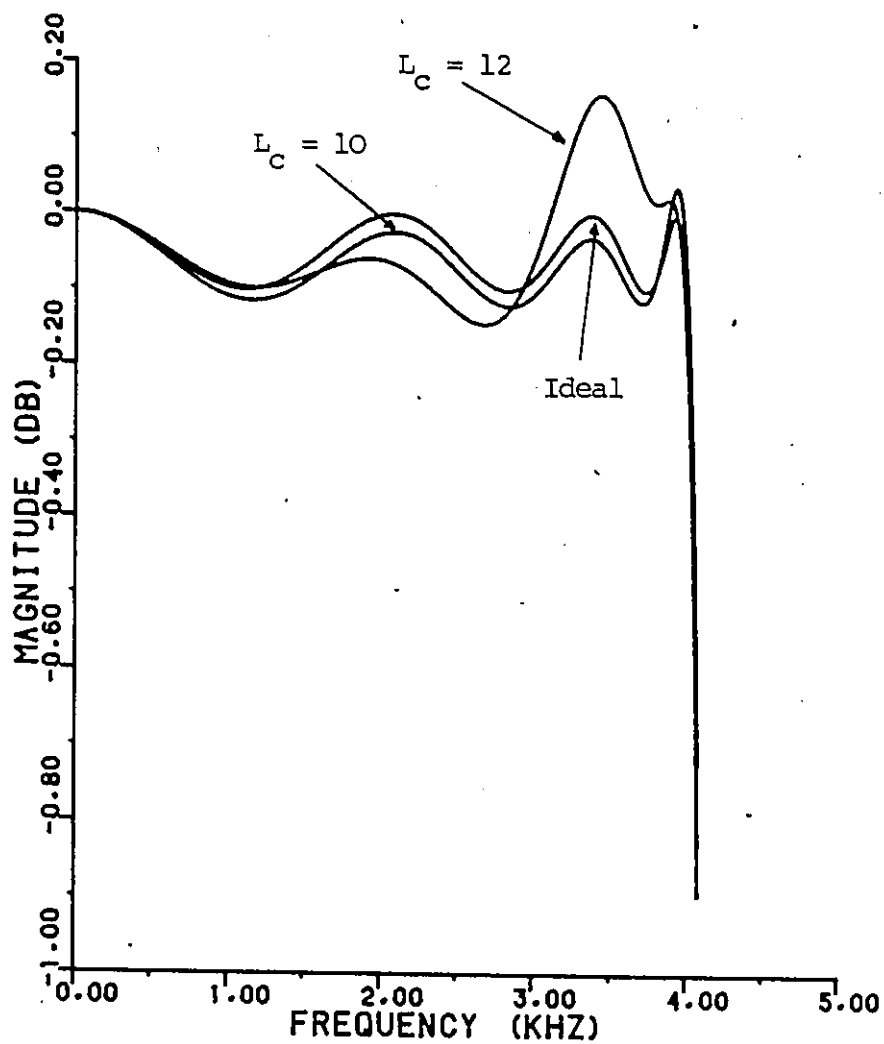


Figure 3.4: Frequency response of a 7-th order Chebyshev LPF.

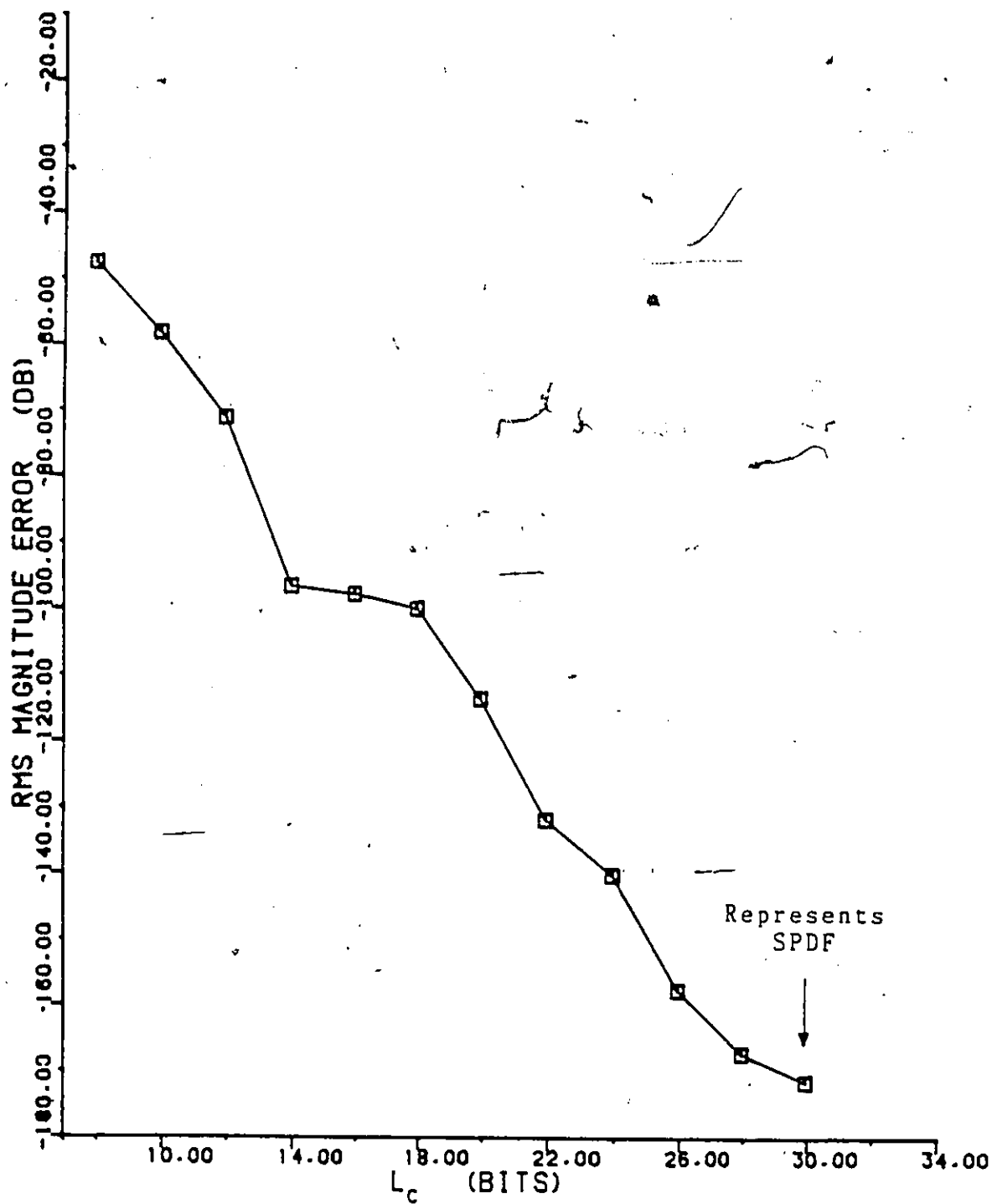


Figure 3.5: RMS magnitude error of a 3-rd order direct filter.

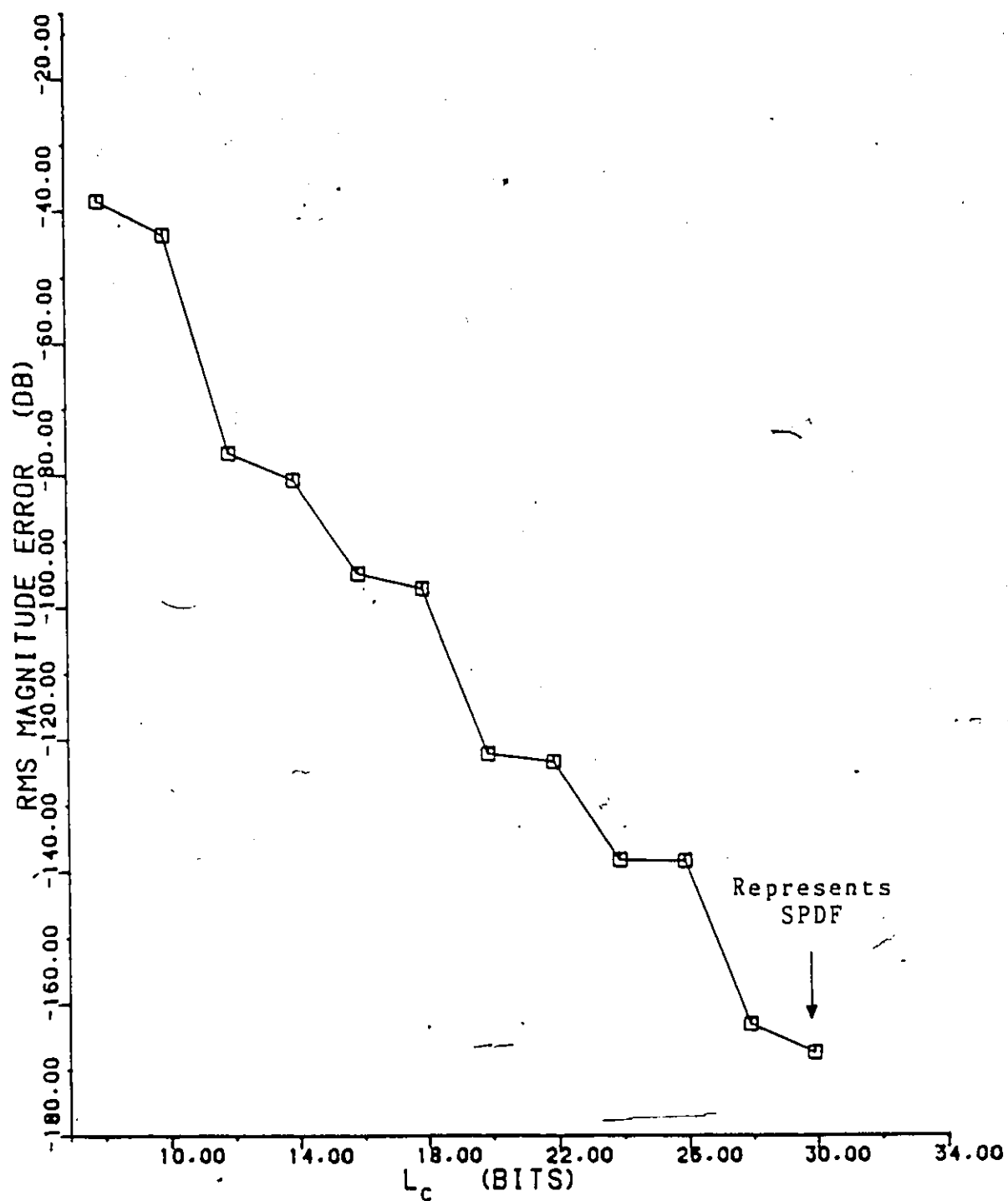


Figure 3.6: RMS magnitude error of a 5-th order direct filter.

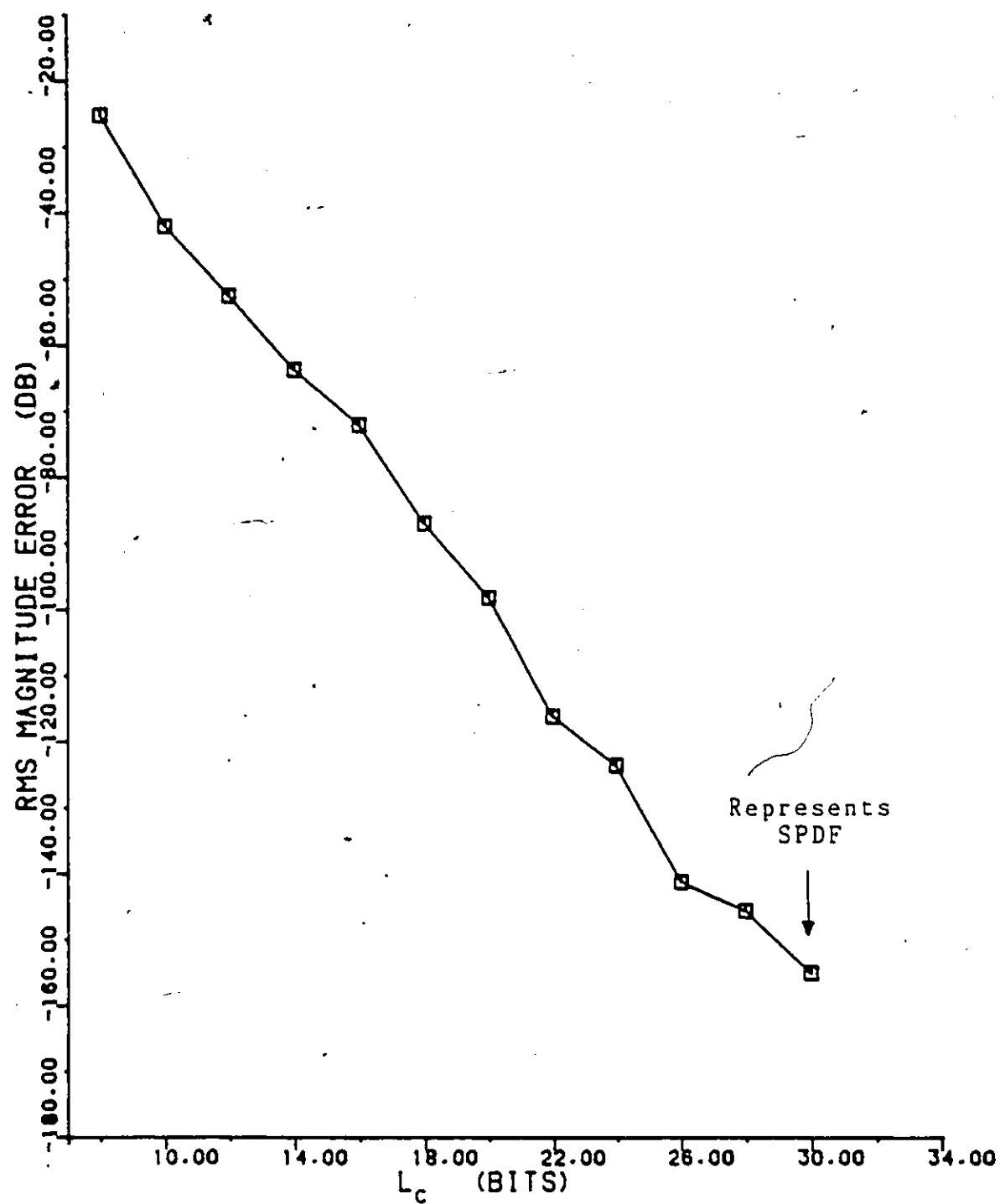


Figure 3.7: RMS magnitude error of a 7-th order direct filter.

thermore, since the coefficients in stored-product digital filters are practically unchanged from those in the original design, the position of the poles within the unit circle is preserved, thus the filter's stability obtained in the original design is maintained.

3.3 Roundoff Error Accumulation

Other sources of errors in a digital filter are due to the arithmetic operations. For fixed point arithmetic, if the input signal is kept within a given range, addition will not introduce error in the sum, otherwise overflow error will occur. Multiplication is followed by product quantization, causing roundoff error. Therefore, the output signal of a digital filter is corrupted by an output error due to the accumulation of product roundoff errors.

Product quantization produces additive error sources [18]-[20], since these error sources are assumed to be statistically independent. The variance of the error in the output signal can be computed as the sum of the variance of individual product roundoff errors weighted by the transfer functions from their point of injection to the filter output. That is

$$\sigma_o^2 = \sum_{k=1}^M \sigma_{e_k}^2 G_k \quad (3.12)$$

$$G_k = \frac{1}{2\pi j} \oint B_k(z) B_k(z^{-1}) z^{-1} dz \quad (3.13)$$

where $\sigma_{e_k}^2$ is the variance of error from the k -th error source, M is the number of error sources in the filter, and $B_k(z)$ is the transfer function from the k -th error source to the filter output.

Suppose the digital filter is implemented using the SPDF design technique. Basically, the error gain G_k remains the same, since the structure of the filter is unchanged. On the other hand, the accuracy of the coefficients is increased. As the result, the product error in each multiplier is reduced, thus the output error of the SPDF is reduced as well.

The following analysis is based on the computational technique given in [19]. The average error energy in the output signal is defined as

$$\bar{E} = \frac{1}{N} \sum_{n=1}^N \{y(n) - v(n)\}^2 \quad (3.14)$$

where $v(n)$ and $y(n)$ are the output signals for the actual and the ideal filters respectively. The ideal filter is simulated with signal and coefficient wordlength of more than 32 bits. A random sequence with uniform distribution within -1.0 and +1.0 is used as the input signal and is applied to the ideal and the actual filters as shown in Fig. 3.8.

The accumulation of roundoff error in stored-product digital filters of the direct form was examined for various product wordlengths and with a sufficiently large number of signal samples. The results were plotted in Fig. 3.9 to 3.11. It can be seen, that the SPDFs (represented by filters with $L_c = 30$ bits) exhibit lower output error than filters with shorter coefficient wordlength. In the SPDFs the output error is determined by the product wordlength only, and an increase in the product wordlength by one bit gives a decrease in the output error energy by 6 dB. On the other hand, the output error in the

filters with shorter coefficient wordlength is dependent on the product and the coefficient wordlengths, and is influenced also by the input signal power. This analysis shows similar results to that for the single stored-product ROM.

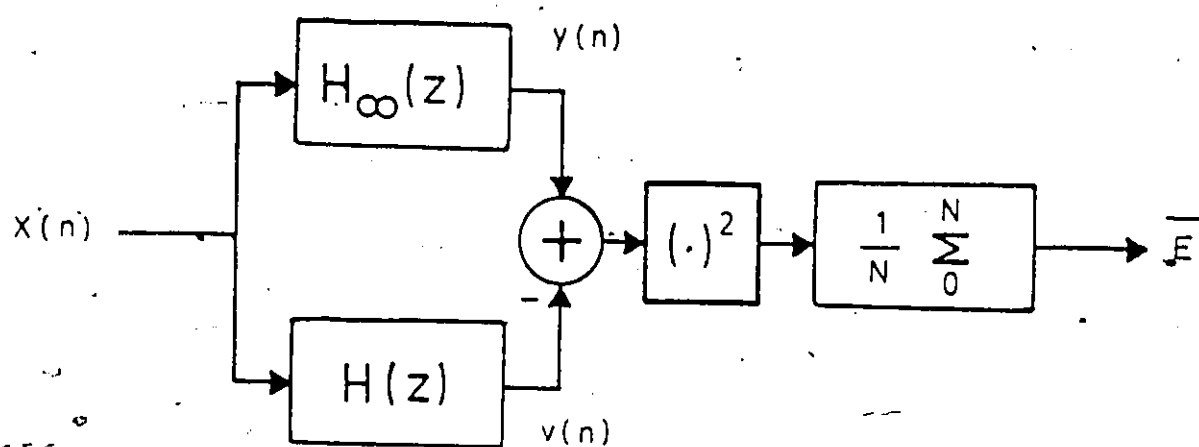


Figure 3.8: Technique for measuring roundoff error accumulation in SPDFs.

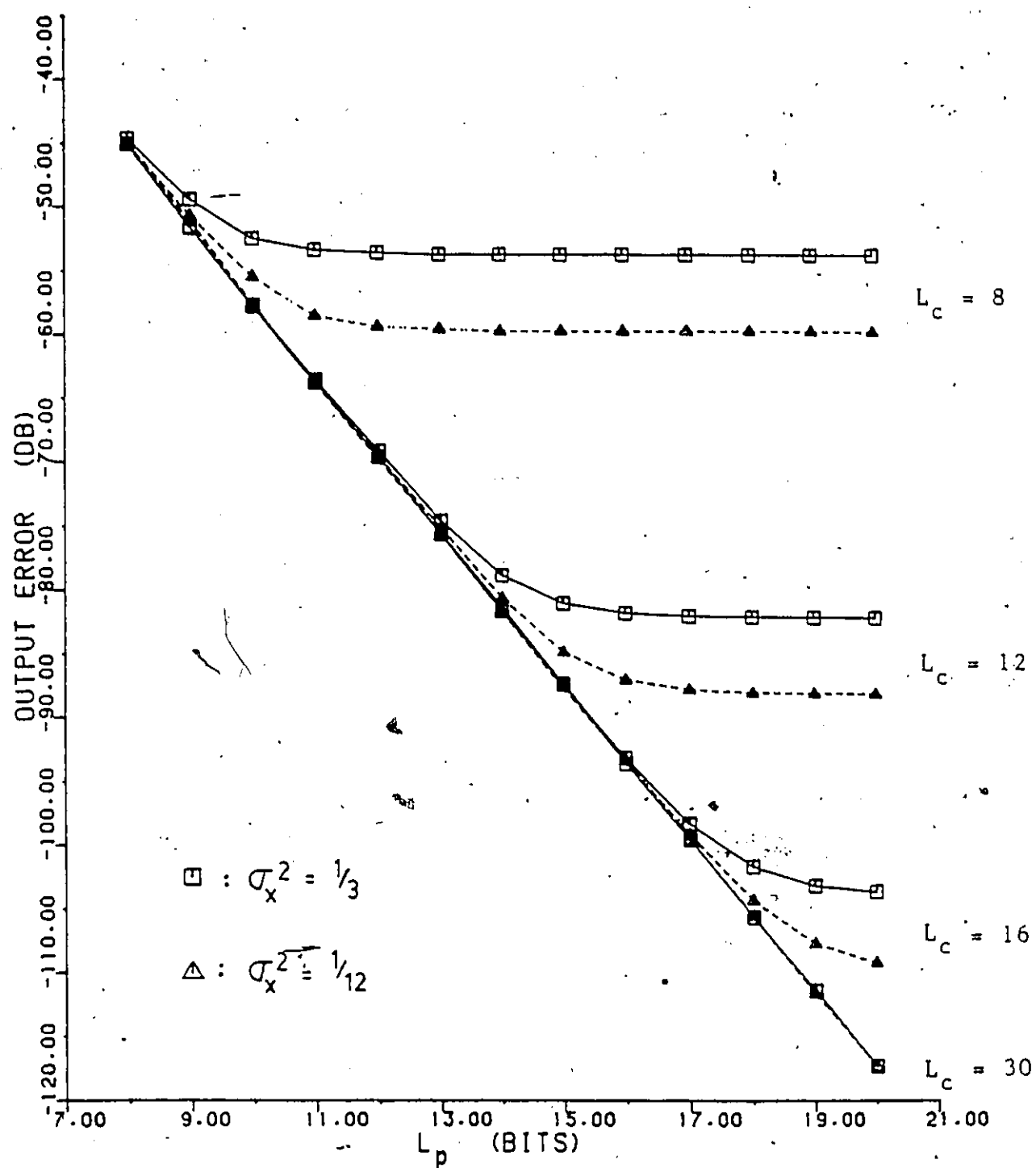


Figure 3.9. Average output error of a 3-rd order direct filter.

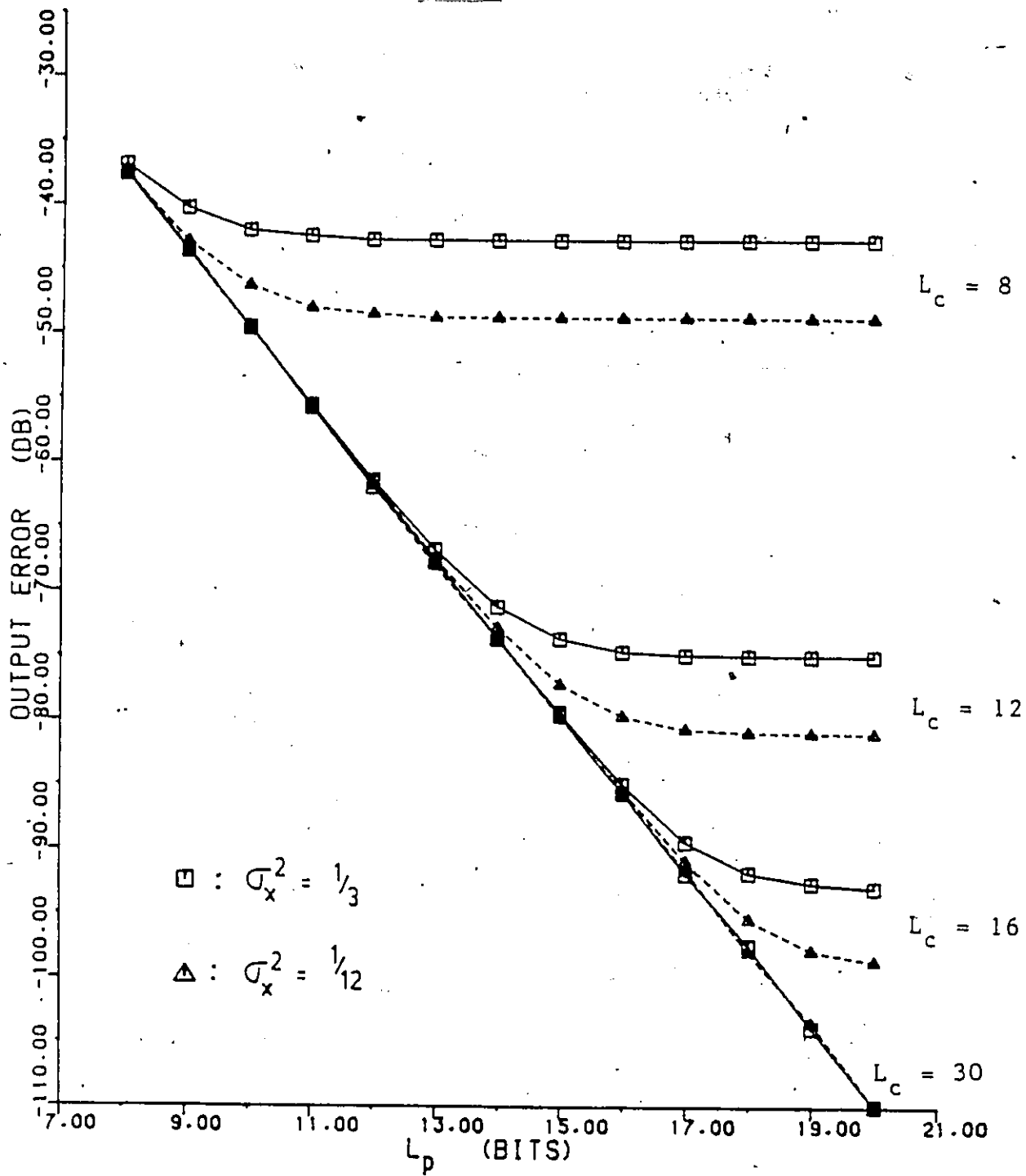


Figure 3.10: Average output error of a 5-th order direct filter.

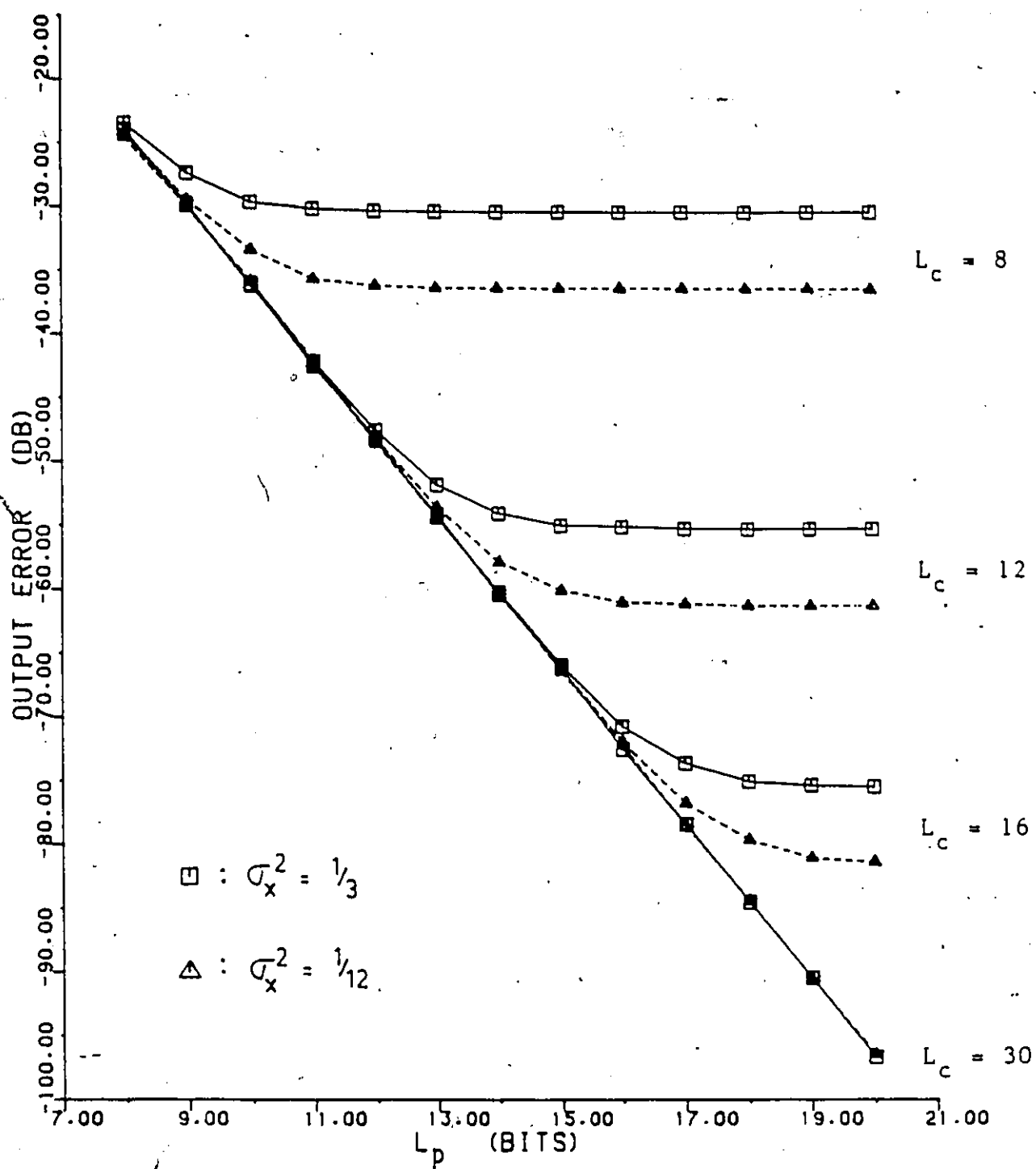


Figure 3.11: Average output error of a 7-th order direct filter.

3.4 Proposed Structure for Stored-Product Digital Filters

In general, a stored-product digital filter can be realized in a variety of forms, and the maximum speed of an SPDF varies from one realization to another, since it is determined by the duration of the longest sequence of arithmetic operations between any two unit-delay elements to be completed in one cycle. An N -th order filter realized in "direct form 1" (Figure 3.1) requires the completion of 1 multiplication and $\log_2 (2N+1)$ additions per clock cycle, and the one realized in "direct form 2" (Figure 3.12) requires the completion of 2 multiplications and $1+\log_2 (N+1)$ additions per clock cycle.

Digital filters intended for high speed operations should be realized in a structure requiring a minimum number of arithmetic operations per cycle. A suitable structure for stored-product digital filters is shown in Fig. 3.13, in which the longest sequence of arithmetic operations between any two unit-delay elements is minimized to one multiplication and one addition. A 16-bit SPDF with this structure has been reported [15] to have potential throughput rates in the range 10 MHz using standard TTL devices and bipolar ROMs. In this structure, all ROMs are addressed simultaneously by the same address code, thus allowing to use a single input multiple output ROM. Moreover, when two or more coefficients are identical, some outputs can be shared. In this case there will be cost benefit from hardware saving.

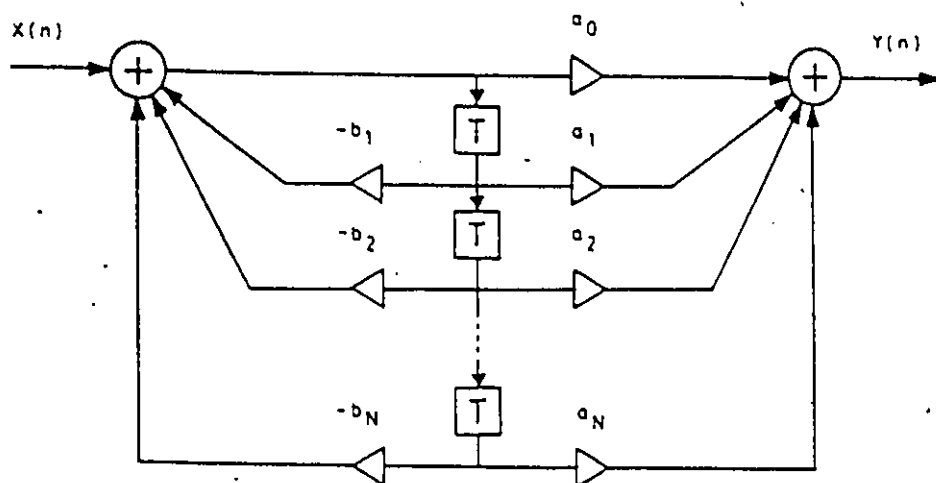


Figure 3.12: "Direct form 2" realization of an N-th order direct filter.

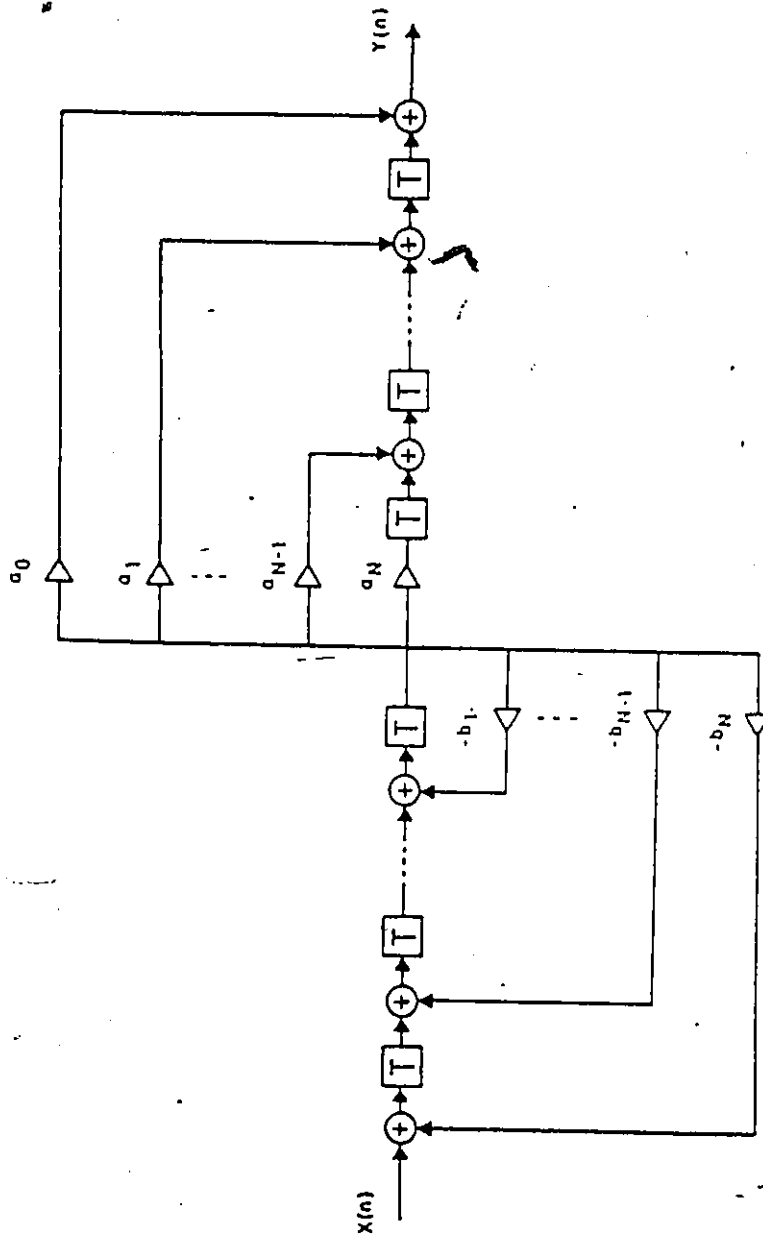


Figure 3.13: Proposed structure for stored-product digital filters.

Chapter IV
A COMPARISON OF STORED-PRODUCT AND WAVE DIGITAL
FILTERS

4.1 Introduction

Since much of the research in the field of digital filtering has emphasized the search for structures with low sensitivity to coefficient roundoff errors, a digital filter structure, called the wave digital filter [1]-[3], has evolved. In addition to the low coefficient sensitivity property, wave digital filters have low roundoff error accumulation [4]-[6]. Since in stored-product digital filters the coefficient roundoff errors are minimized to arbitrarily low values and roundoff error accumulation is reduced, therefore, it is reasonable to make a comparison between stored-product and wave digital filter.

In this chapter, a comparison between stored-product and wave digital filters is presented. The comparison will be based on frequency response accuracy, roundoff error accumulation, hardware complexity, and speed of operation. These aspects are considered, because for other digital filters, the filter performance is usually improved at the expense of hardware complexity and a decrease in filtering speed, while in stored-product digital filters, improved performance is achieved with minimal hardware complexity at high speed.

4.2 Wave Digital Filters

The design method for wave digital filters is somewhat different from conventional digital filters. A wave digital filter structure is derived from the classical analog LC filter structure by replacing all analog elements by their corresponding digital realizations. The low sensitivity property is obtained because LC ladder filters are insensitive to element variations [4]. In this section a review of the wave digital filter design concept is given. Much of the introductory material in this section can be found in [1]-[3] and [20].

4.2.1 General Principles

An analog n -port network like that in Fig. 4.1 can be represented by a set of equations

$$\begin{aligned} A_k &= V_k + R_k I_k \\ B_k &= V_k - R_k I_k \end{aligned} \quad (4.1)$$

$$k = 1, 2, \dots, n$$

where the parameters A_k and B_k are the incident and reflected wave quantities respectively, and R_k is the port resistance. Since an LC ladder filter can be regarded as consisting of a number of one-port impedances (R , sL , or $1/sC$), and a number of 3-port networks, therefore, by realizing the digital equivalent of these elements the corresponding wave digital filter can be obtained. Use is made of the analogy between transmission line filters and digital filters. The basic design sequence thus is: (1) LC filter, (2) transmission line filter obtained by replacing each L by a shorted transmission line and each C by an open-circuited transmission line, (3) digital filter obtained directly from a modified version of the transmission line filter as explained in the next section.

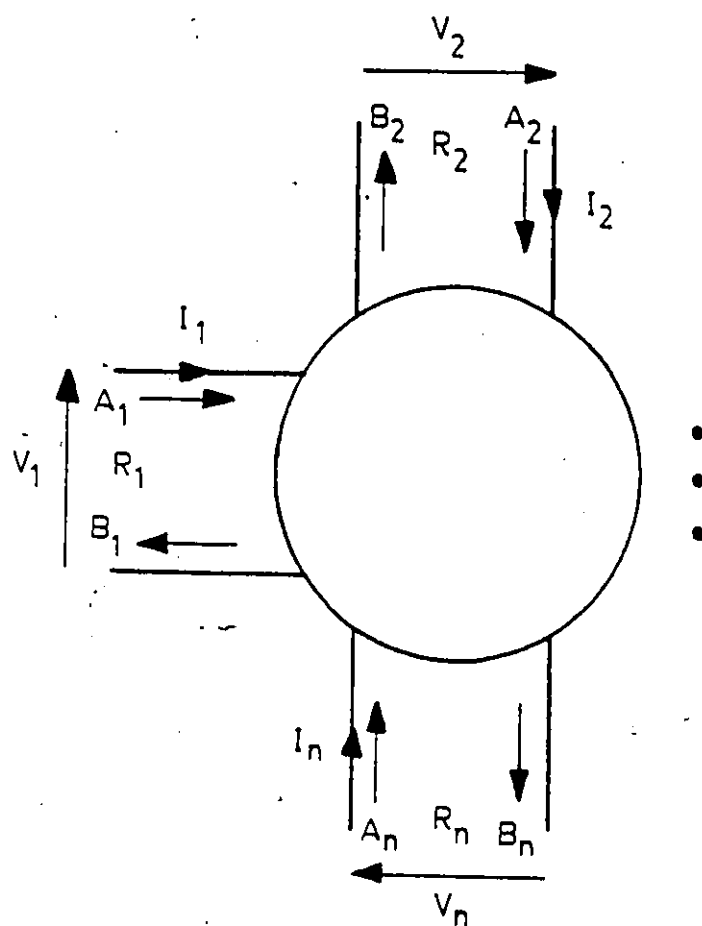


Figure 4.1: Analog n-port network.

4.2.2 Digital Realization of Analog Elements.

For an inductor, the steady-state voltage-current relation to be realized is given by

$$V = j\omega L \cdot I \quad (4.2)$$

and for a shorted transmission line

$$V = \psi R I \quad (4.3)$$

$$\begin{aligned} \psi &= \frac{1 - e^{-sT}}{1 + e^{-sT}} \\ &= \tanh(sT/2) \end{aligned} \quad (4.4)$$

where s is the complex frequency, T is the sampling period which is equal to the propagation time in the shorted line, and $R = \sqrt{L/C}$ is the characteristic impedance of the transmission line. Substituting (4.4) and (4.3) into (4.1) yields

$$A = R I (\psi + 1) \quad (4.5)$$

$$B = R I (\psi - 1)$$

so that

$$\begin{aligned}
 B &= \frac{\psi - 1}{\psi + 1} \\
 &= -e^{-sT} A \\
 &= -z^{-1} A
 \end{aligned}
 \tag{4.6}$$

Therefore, digital realization of an inductor is a unit-delay element and a sign inverter to be connected to a port of resistance R (Fig. 4.2-a).

For a capacitor, the steady-state voltage-current relation to be realized is given by

$$V = (1/j\omega C) I \tag{4.7}$$

and for an open-circuited transmission line

$$V = R I / \psi \tag{4.8}$$

where $R = C$ is the characteristic impedance of the transmission line. Upon using (4.1) and (4.4) one obtains

$$B = z^{-1} A \tag{4.9}$$

which means that the digital realization of a capacitor is a unit-delay element (Fig. 4.2-b).

For a resistor, the equation to be realized is

$$V = R I \quad (4.10)$$

From (4.1) and (4.4), it follows that

$$B = 0 \quad (4.11)$$

Thus, the digital realization of a resistor is a "wave sink" (Fig. 4.2-c), as it is well matched to the port of resistance R .

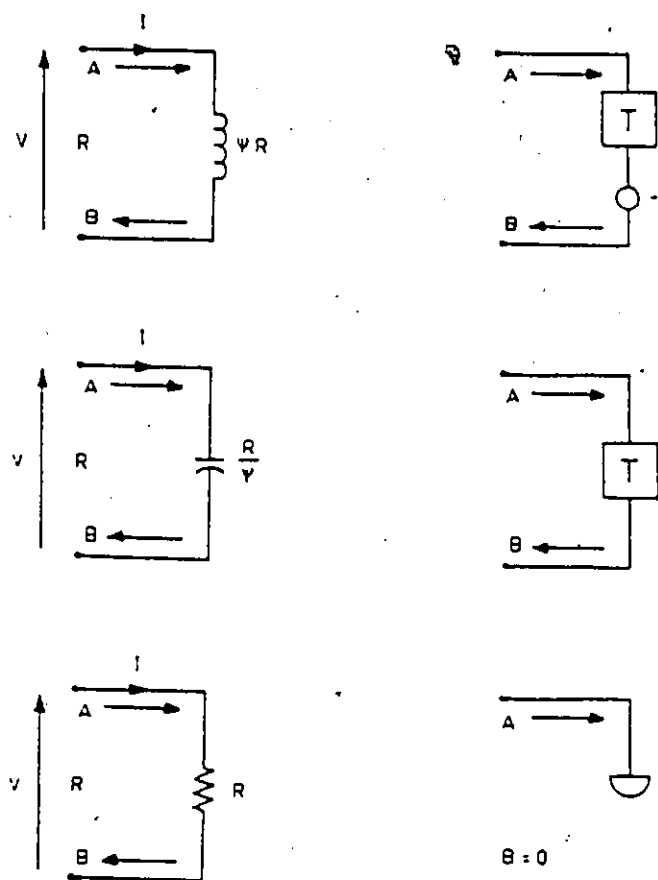


Figure 4.2: Digital realization of analog elements. (a) Inductance of impedance ψR . (b) Capacitance of impedance R/ψ . (c) Resistance R .

4.2.3 3-Port Adaptors

In a wave digital filter, the adders and multipliers can be grouped into 3-port building blocks called adaptors or interconnectors. Parallel adaptors serve to simulate the parallel connections in the LC ladder filter, and series adaptors the corresponding series connections.

If ports 1, 2, and 3 are connected in parallel, then

$$V_1 = V_2 = V_3 \quad (4.12)$$

$$I_1 + I_2 + I_3 = 0$$

From (4.1), it follows that, for port with resistance R_k

$$B_k = \sum_{j=1}^3 c_j A_j - A_k \quad (4.13)$$

$$k = 1, 2, 3$$

where c_1 , c_2 , and c_3 are the coefficients for this particular adaptor and are derived from the value of the resistance R_k

$$c_k = \frac{2 G_k}{G_1 + G_2 + G_3} \quad (4.14)$$

$$G_k = 1 / R_k \quad (4.15)$$

$$c_1 + c_2 + c_3 = 2 \quad (4.16)$$

This parallel adaptor is depicted in Fig. 4.3. The realization of this adaptor requires 2 multipliers, thus also called type P2 adaptor [20].

If these ports are connected in series, then

$$I_1 = I_2 = I_3 \quad (4.17)$$

$$V_1 + V_2 + V_3 = 0$$

so that

$$B_k = A_k - c_k (A_1 + A_2 + A_3) \quad (4.18)$$

where

$$c_k = \frac{2 R_k}{R_1 + R_2 + R_3} \quad (4.19)$$

and (4.16) still holding. This 3-port series adaptor also requires 2 multipliers (Fig. 4.4), and hence called type S2 adaptor.

Of particular interest are the 3-port adaptors for which one of the ports is "reflection-free" [3] as it is connected to its matched resistance R_k . These adaptors require only 1 multiplier. For a 3-port parallel adaptor with port 3 being the reflection-free port

$$G_3 = G_1 + G_2 \quad (4.20)$$

so that

$$c_k = \frac{G_k}{G_3} \quad (4.21)$$

The realization of this adaptor is shown in Fig.4.5. It is also called type P1 adaptor.

Similarly for a 3-port series adaptor with port 3 being the reflection-free port,

$$R_3 = R_1 + R_2 \quad (4.22)$$

and

$$c_k = \frac{R_k}{R_3} \quad (4.23)$$

This adaptor is called type S1 adaptor and is shown in Fig.4.6.

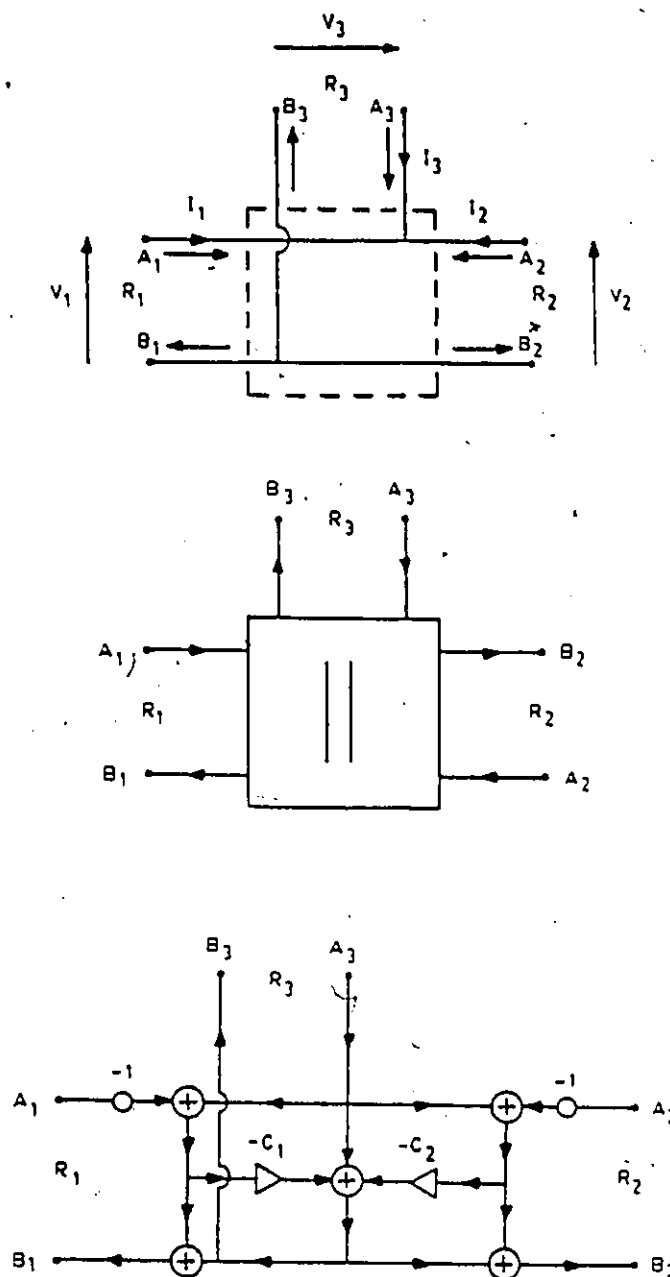


Figure 4.3: Type P2 Adaptor. (a) Parallel connection of 3 ports. (b) Block diagram of type P2 adaptor. (c) Realization of type P2 adaptor.

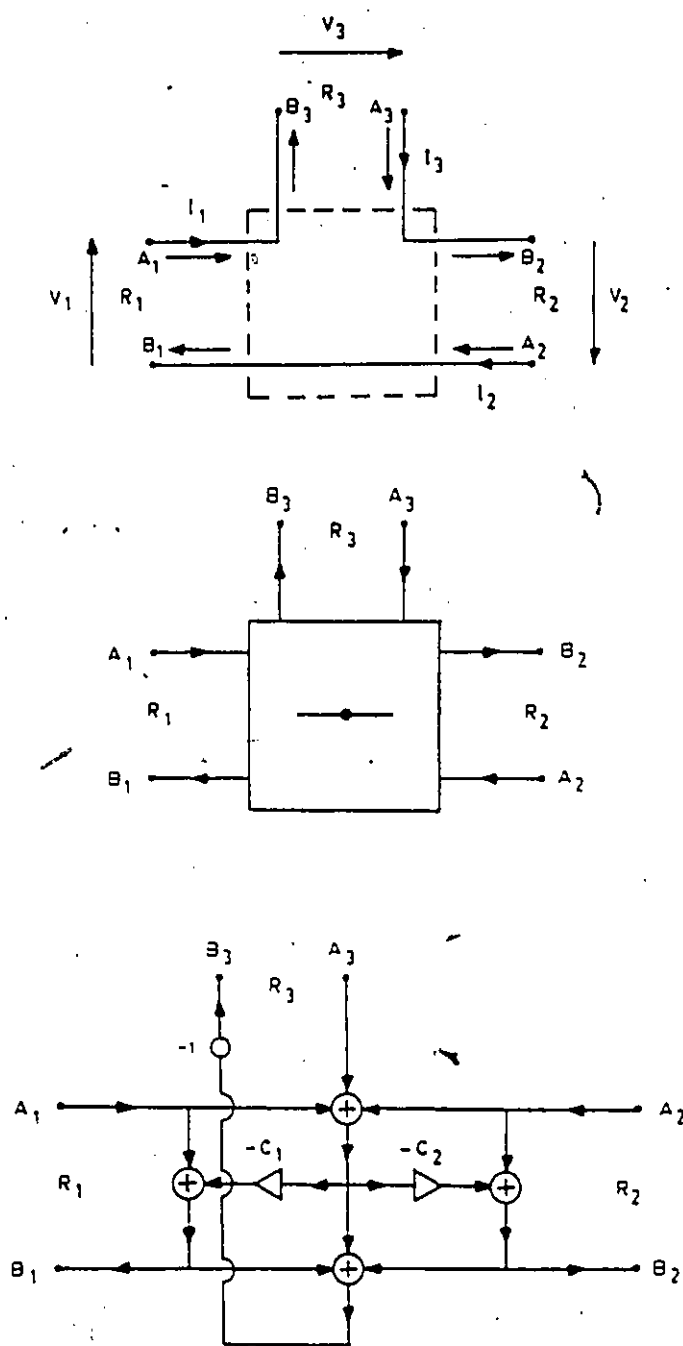


Figure 4.4: Type S2 Adaptor. (a) Series connection of 3 ports. (b) Block diagram of type S2 adaptor. (c) Realization of type S2 adaptor.

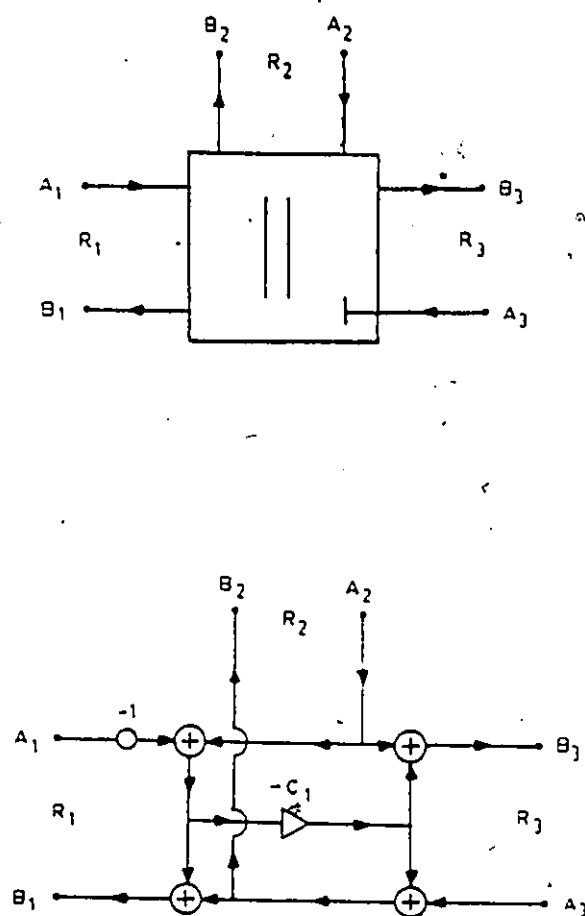


Figure 4.5: Type P1 Adaptor. (a) Block diagram of type P1 adaptor. (b) Realization of type P1 adaptor.

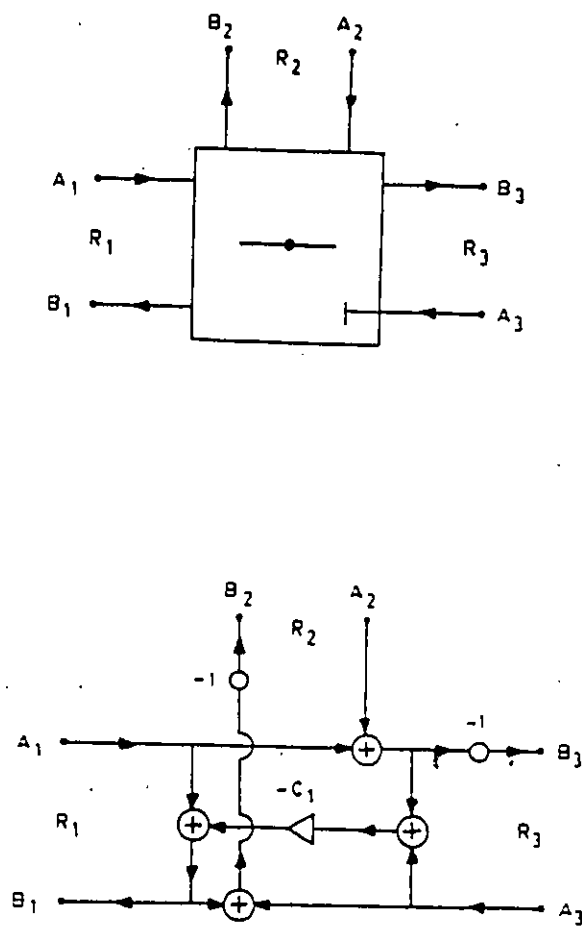


Figure 4.6: Type S1 Adaptor. (a) Block diagram of type S1 adaptor. (b) Realization of type S1 adaptor.

4.2.4 Design Example

With digital equivalents for the various analog elements available, given an LC filter, one can derive a corresponding wave digital filter by identifying the various 3-port connections in the LC filter and assigning port resistances by the corresponding impedances of the analog elements [20]. That is

$$R_k = \begin{cases} R & ; \text{ Resistor} \\ 2L/T & ; \text{ Inductor} \\ T/2C & ; \text{ Capacitor} \end{cases}$$

As the design example consider the LC ladder filter depicted in Fig. 4.7-a which is a 5-th order Chebyshev lowpass filter with a cut-off frequency at 4 KHz and a maximum passband ripple of 0.1 dB. The element values of the filter (with the termination resistances R_S and R_L equal 1 ohm) are

$$C_1 = C_5 = 35.8375 \text{ } \mu\text{F}$$

$$C_3 = 61.71875 \text{ } \mu\text{F}$$

$$L_2 = L_4 = 42.85 \text{ } \mu\text{H}$$

The corresponding wave digital filter with a sampling frequency of 16 KHz is shown in Fig. 4.7-b. The coefficients for the wave digital filter are calculated as follows:

Adaptor 1 (type P1 adaptor):

$$G_{11} = \frac{1}{R_s} ; G_{21} = \frac{2C_1}{T} ; G_{31} = G_{11} + G_{21}$$

$$c_{11} = 0.4658$$

Adaptor 2 (type S1 adaptor):

$$R_{12} = \frac{1}{G_{31}} ; R_{22} = \frac{2L_2}{T} ; R_{32} = R_{12} + R_{22}$$

$$c_{12} = 0.2536$$

Adaptor 3 (type P1 adaptor):

$$G_{13} = \frac{1}{R_{32}} ; G_{23} = \frac{2C_3}{T} ; G_{33} = G_{13} + G_{23}$$

$$c_{13} = 0.2161$$

Adaptor 4 (type S1 adaptor):

$$R_{14} = \frac{1}{G_{33}} ; R_{24} = \frac{2L_4}{T} ; R_{34} = R_{14} + R_{24}$$

$$c_{14} = 0.2245$$

Adaptor 5 is type P2 adaptor. This is because the resistance at port 3 is pre-assigned and equal to the termination resistance R_L so that

$$G_{35} = \frac{1}{R_L} ; G_{25} = \frac{2C_5}{T} ; G_{15} = \frac{1}{R_{34}}$$

Thus in adaptor 5 there are 2 multipliers with coefficients given by (4.14), that is

$$c_{15} = 0.4170$$

$$c_{25} = 0.7374$$

Notice that the wave digital filter in Fig. 4.7-b is in the form of block diagram of adaptors and resistor reference values. The actual realization employs adders and multipliers in place of these adaptors.

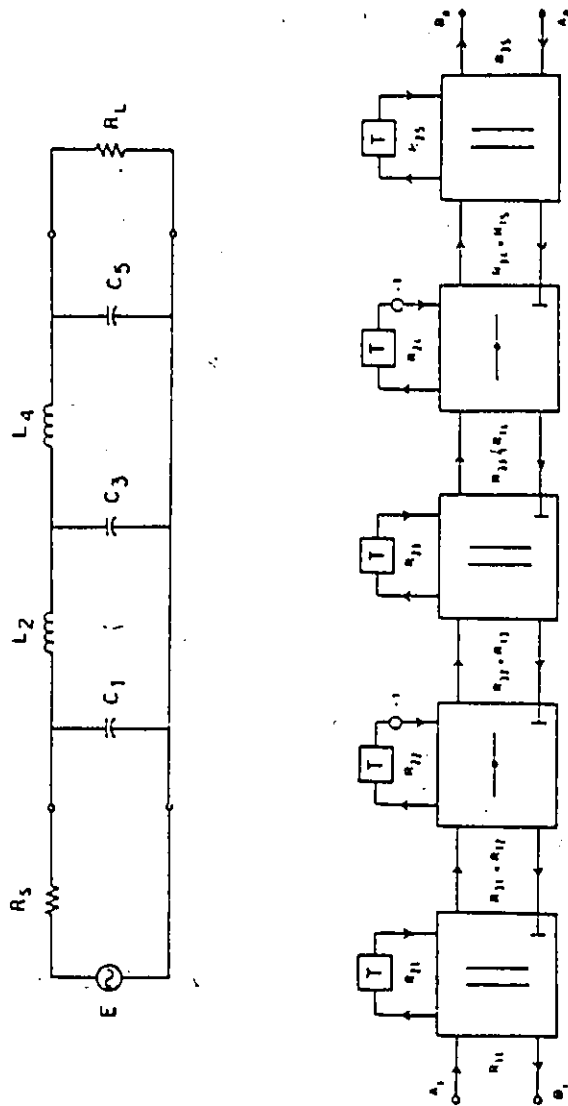


Figure 4.7: Conceptual design of wave digital filter. (a) Analog LC filter. (b) Block diagram of the corresponding wave digital filter.

4.3 Frequency Response Accuracy in Wave and Stored-Product Digital Filters

In this section an evaluation of the effect of the coefficient quantization error to the frequency response of wave digital filters is presented. Three WDFs with the same characteristics as the filters in Chapter III were designed and evaluated for various coefficient wordlengths using the measurement technique given by (3.11).

It is found that, for the same coefficient wordlength, the rms deviation in the frequency response for the WDFs are between 15 to 47 dB better than for the direct filter (Figures 4.8 to 4.10). These results indicate that wave digital filters are indeed less sensitive to coefficient quantization than direct form filters. On the other hand, the frequency response accuracy for the SPDFs with an assumed wordlength of 30 bits is very high, and that the WDFs should be implemented with coefficient wordlength of approximately 24 bits in order to achieve the same accuracy as the SPDFs. Wave digital filters are intended for the design using short coefficient wordlengths, but here we see that only those with rather long coefficient wordlengths exhibit such an accuracy as stored-product filters. From the design simplicity point of view, this shows that the stored-product digital filtering is a better alternative for designing digital filters with high frequency response accuracy than wave digital filters.

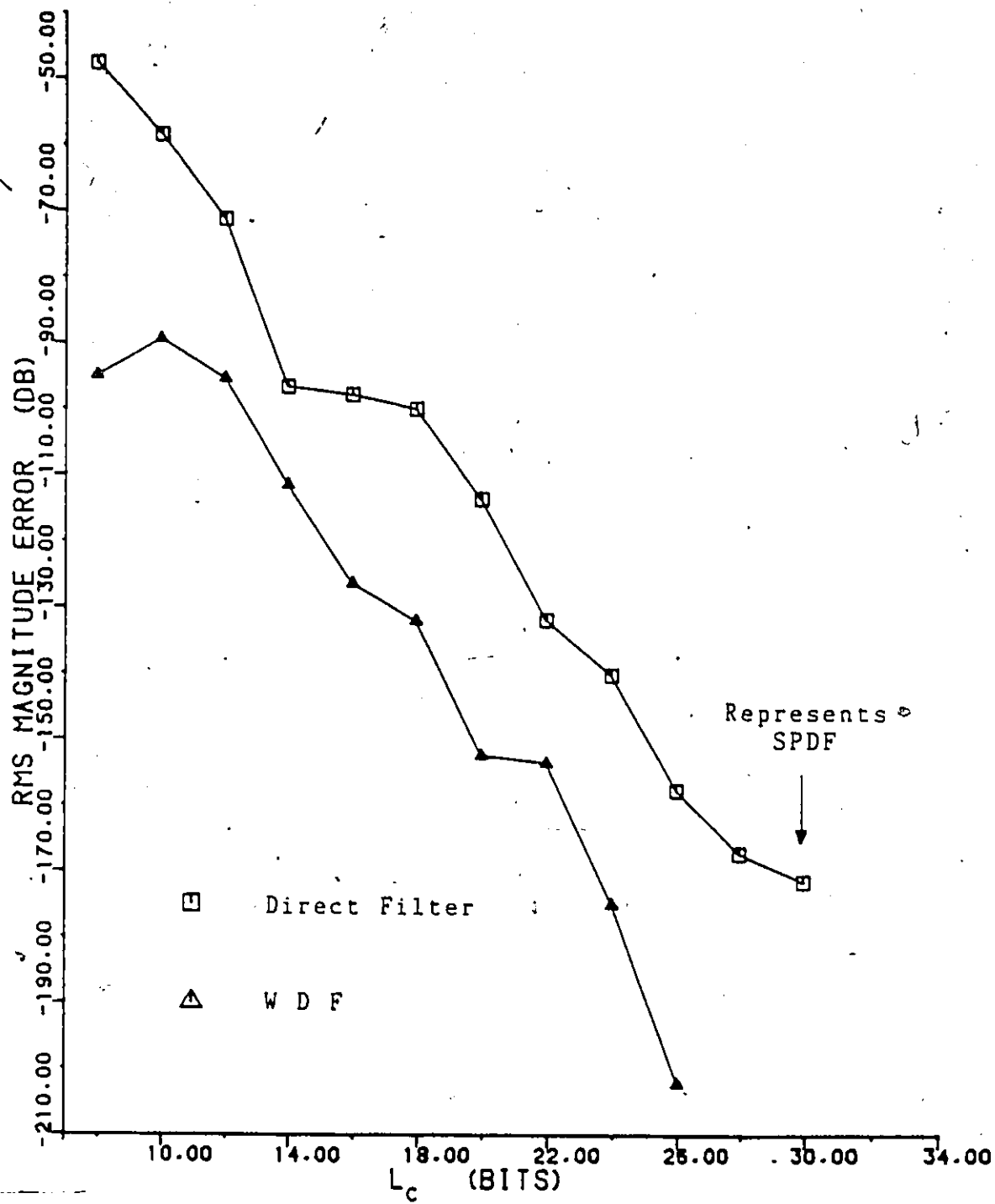


Figure 4.8: RMS magnitude error in the 3-rd order WDF and SPDF.

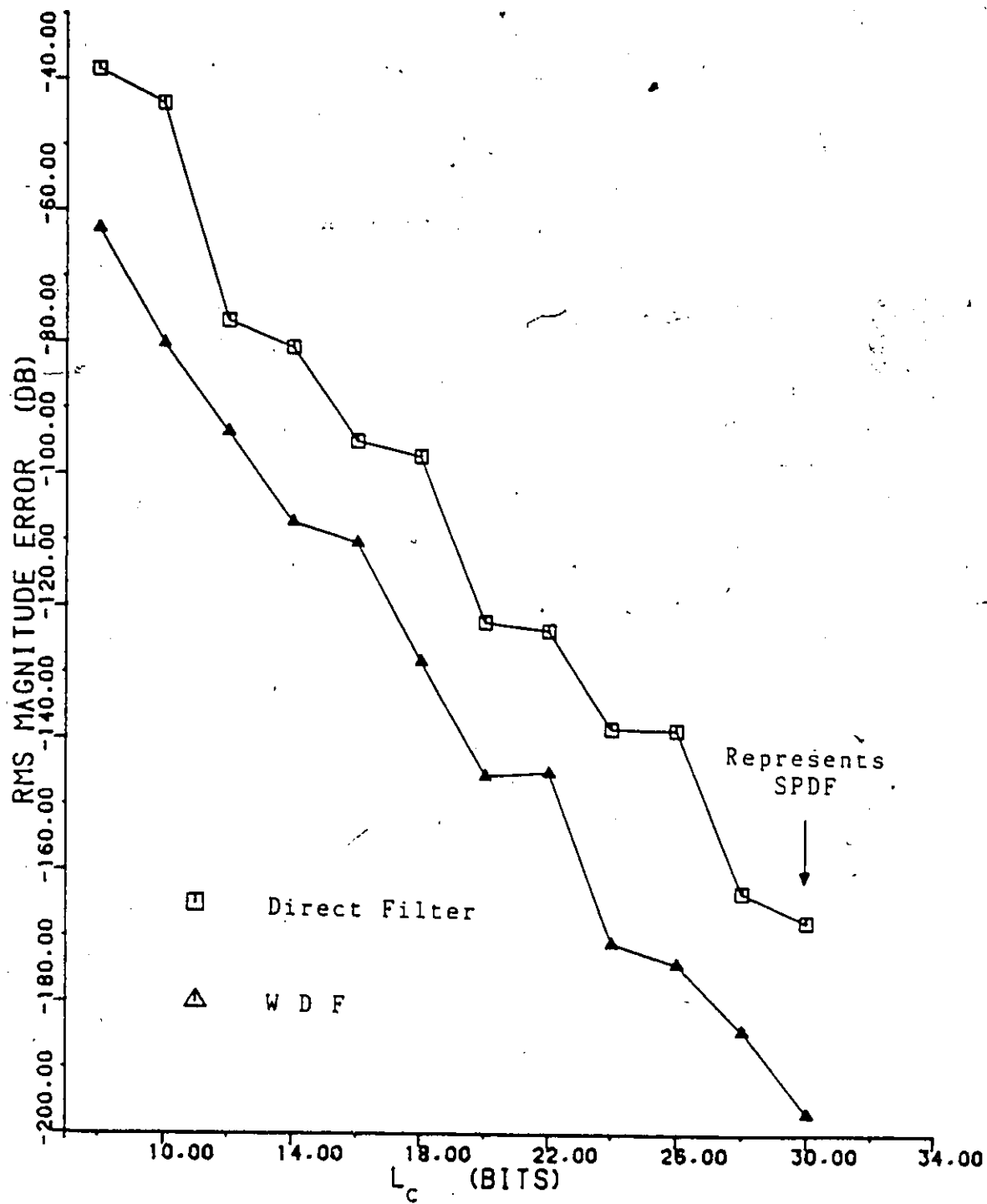


Figure 4.9. RMS magnitude error in the 5-th order WDF and SPDF.

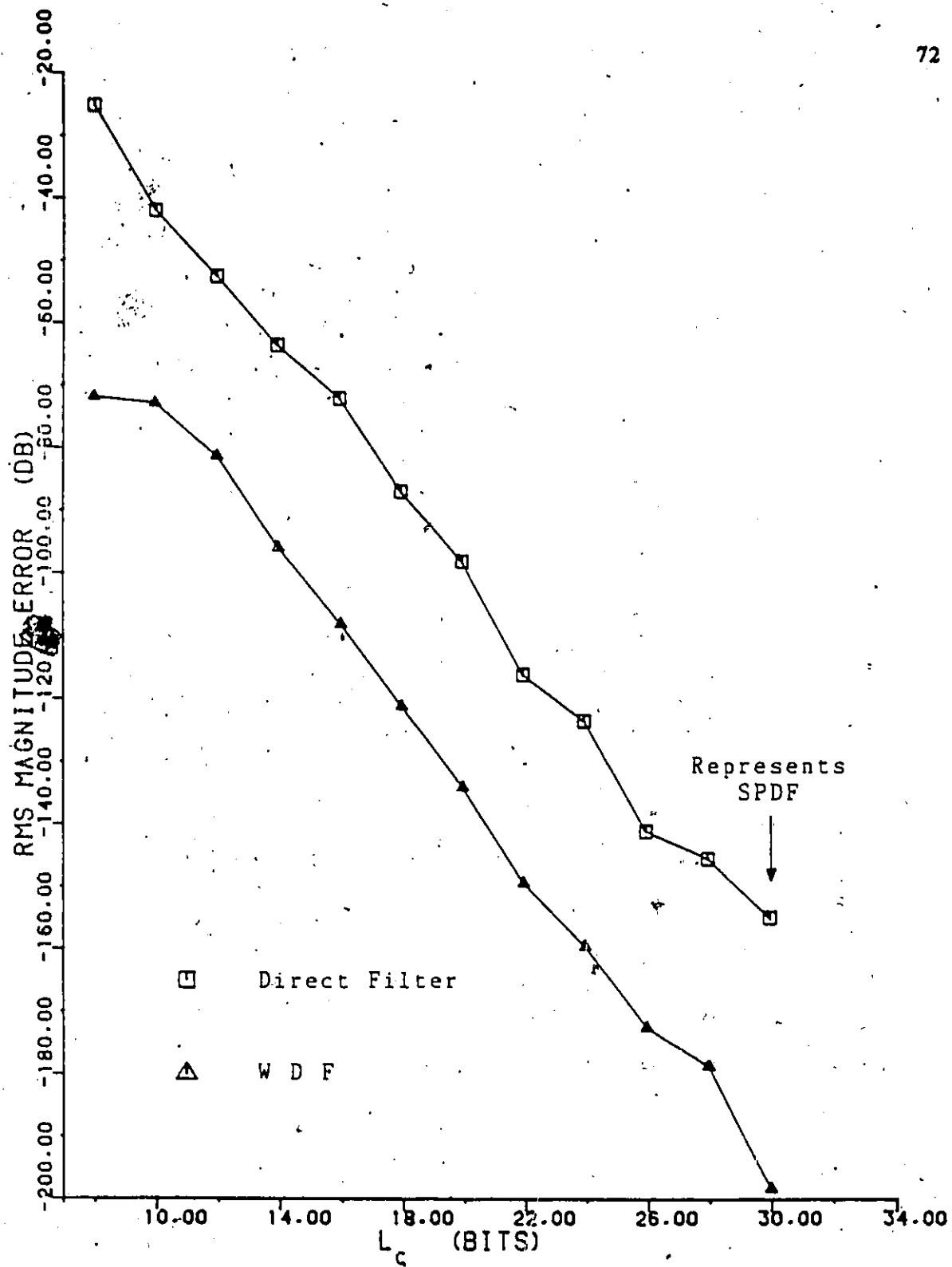


Figure 4.10: RMS magnitude error in the 7-th order WDF and SPDF.

4.4 Roundoff Error Accumulation in Wave and Stored-Product Digital Filters

In this section, the roundoff error accumulation in wave digital filters was evaluated using the measurement technique shown in Fig. 3.8. The WDFs were implemented with coefficient wordlengths of 8, 12 and 16 bits, and evaluated for various product wordlengths. The results were compared with those for the SPDFs and shown in Figures 4.11 to 4.13.

It is found that, for the same product wordlength, the WDFs yield lower roundoff error than the SPDFs. However, the roundoff error in the SPDFs can be reduced by increasing the product wordlength used, and in general, the SPDFs with $L_p = L_c + 3$ will achieve the same performance as the WDFs. On the other hand, for $L_p > L_c + 3$, the WDFs show higher roundoff error than the SPDFs. This is because, as described in Chapter II, the product error in individual multipliers in the WDFs is dominated by the coefficient error.

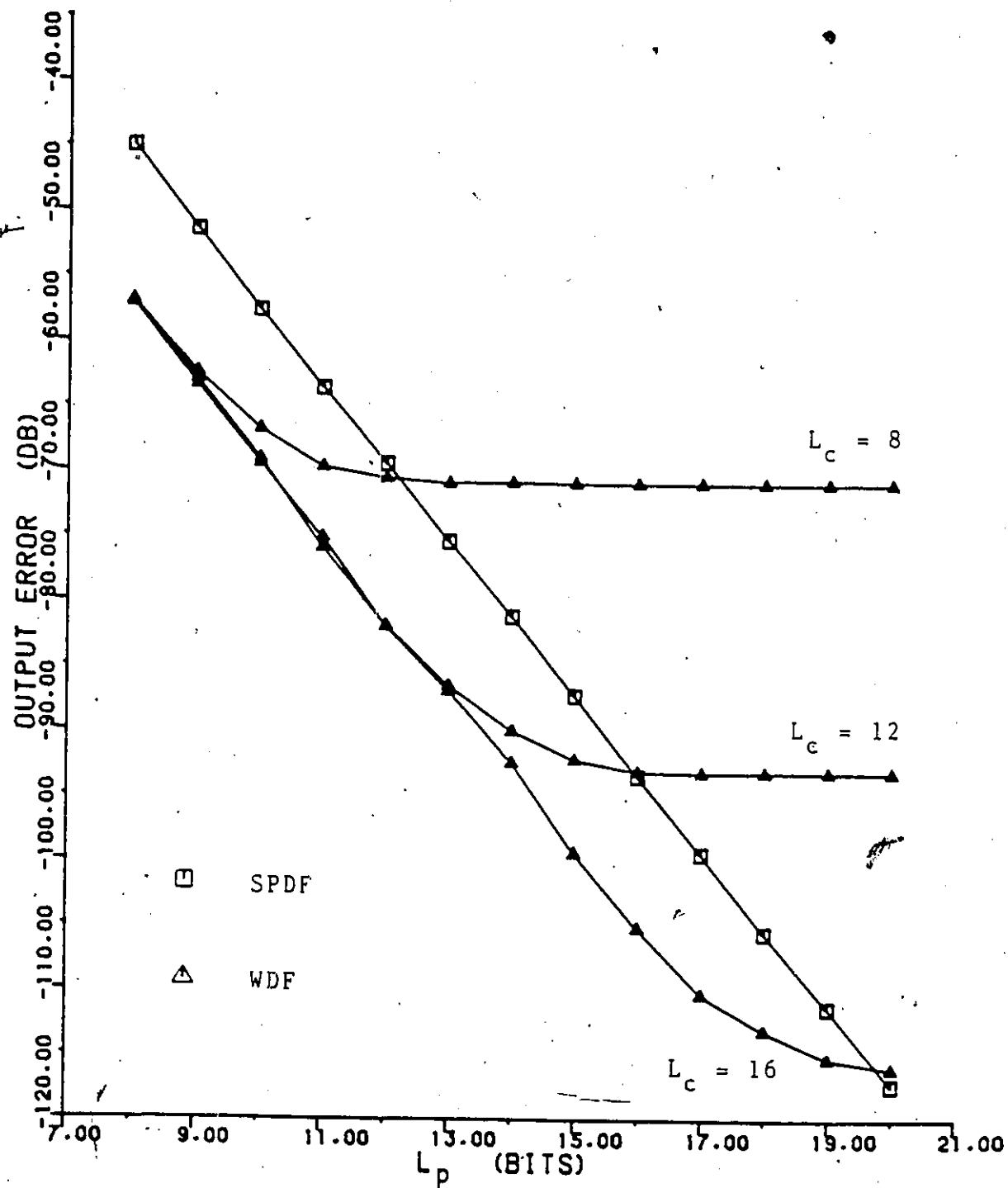


Figure 4.11: Average output error in the 3-rd order WDF and SPDF.

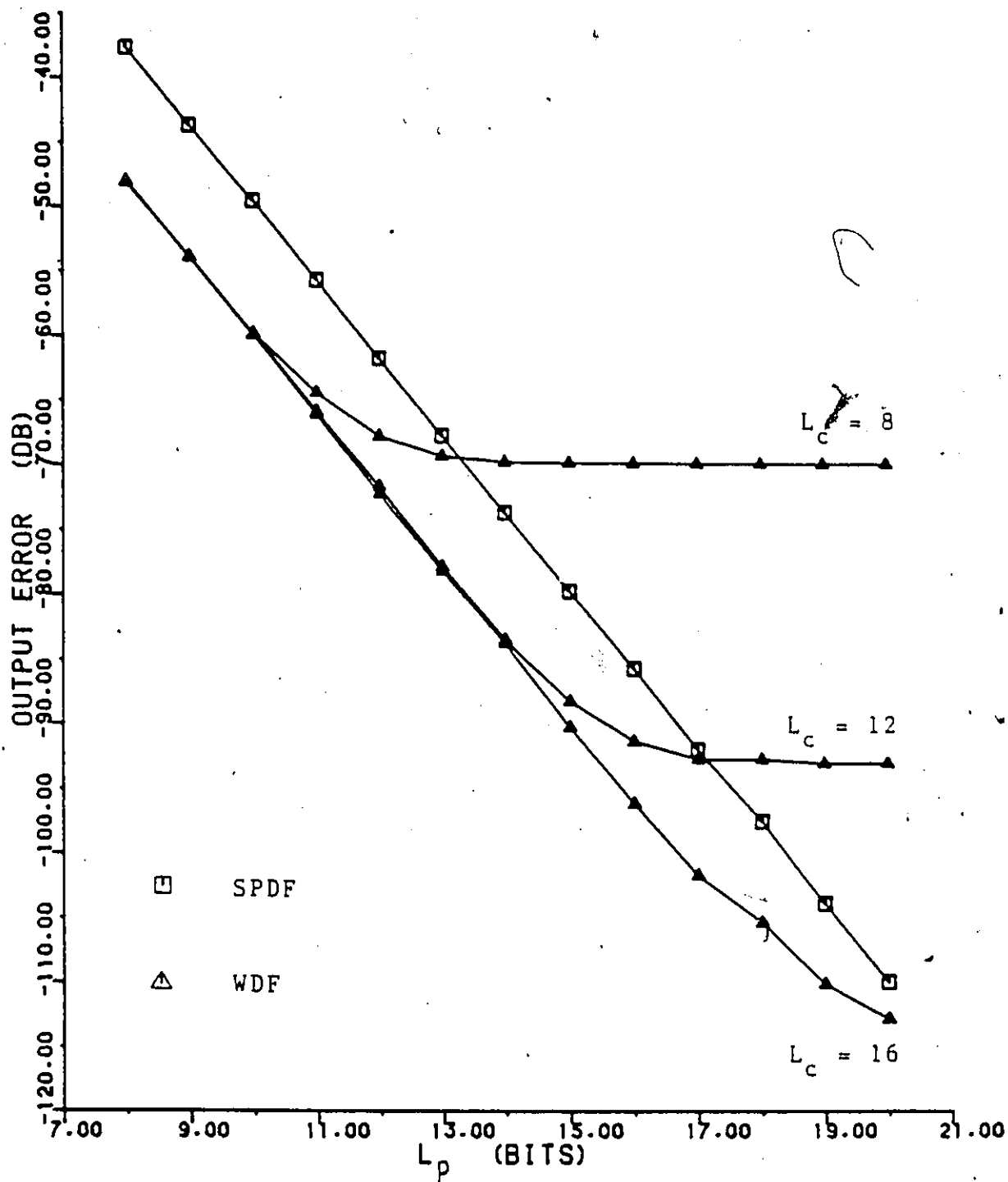


Figure 4.12: Average output error in the 5-th order WDF and SPDF.

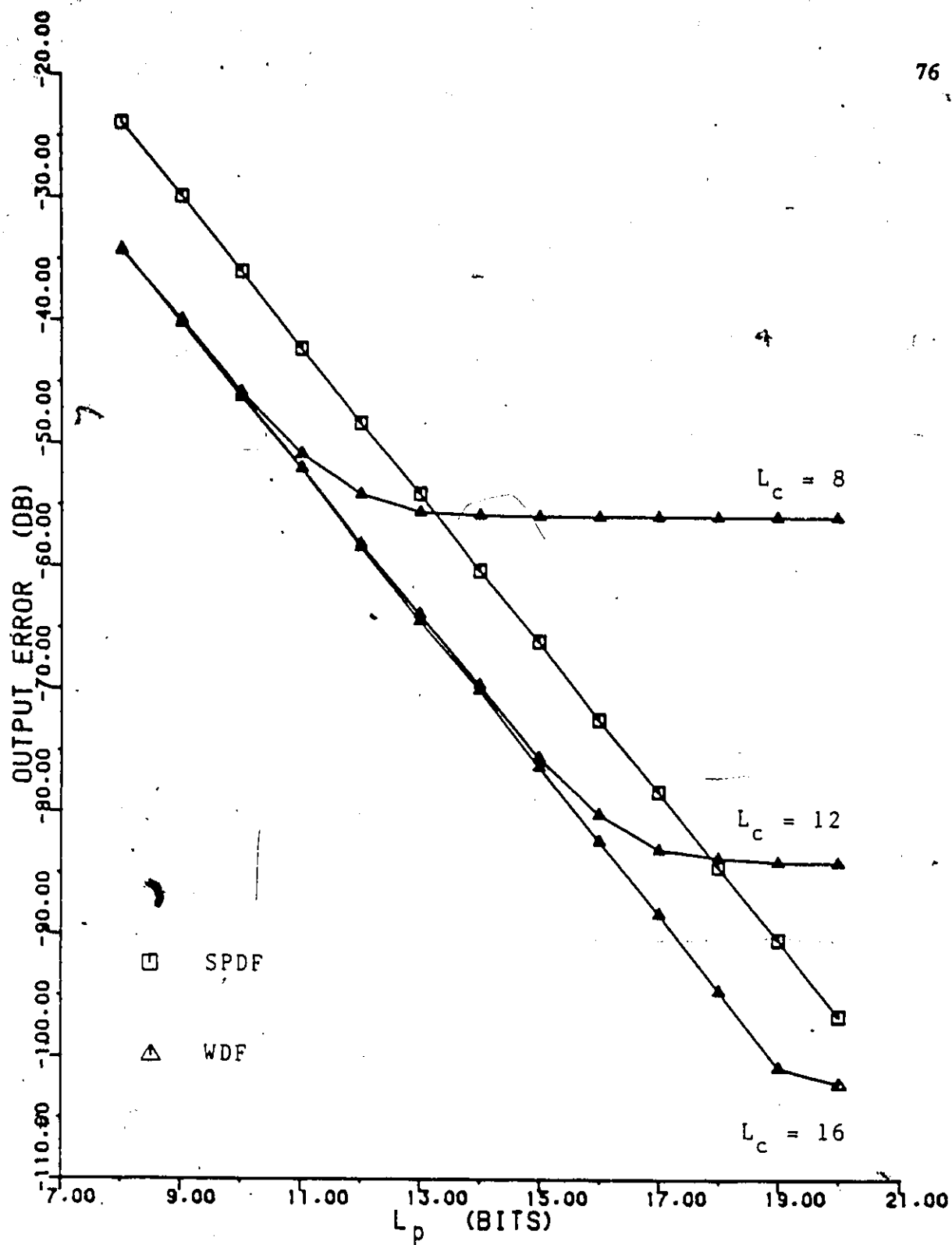


Figure 4.13: Average output error in the 7-th order WDF and SPDF.

4.5 Hardware Realization and Speed Consideration

So far the comparison between wave and stored-product digital filters has been done in terms of arithmetic accuracy. In this section, aspects of the realization and the speed capability are discussed. The block diagrams of the 3-rd and 7-th order WDFs are shown in Figures 4.14 and 4.15 respectively, while the block diagram of the 5th order WDF is the same as that in Fig.4.7-b. The hardware required for the WDFs is shown in these block diagrams and the details of the component are shown in Figures 4.2 to 4.6. The hardware count is listed in Table 4.1. It is shown in this table that the WDFs require fewer multipliers and unit-delay elements but more adders than the SPDFs of the "direct form 1". In general, an N -th order WDF requires $N+2$ multipliers, N unit-delay elements and $4N$ adders, while the SPDF requires $2N+1$ multipliers, $2N$ unit-delay elements and $2N$ adders. However, if stored-product digital filters are realized in canonic form, there will be a reduction by a factor of two in the number of unit-delay elements used.

The speed comparison between the WDFs and the SPDFs is given in terms of the longest sequence of arithmetic operations between two unit-delay elements to be completed in one filter cycle. It can be seen that (Table 4.2), the SPDFs are capable of higher filtering speed than the WDFs. This can be understood since the structures of the WDFs are highly recursive due to the complexity of the adaptors.

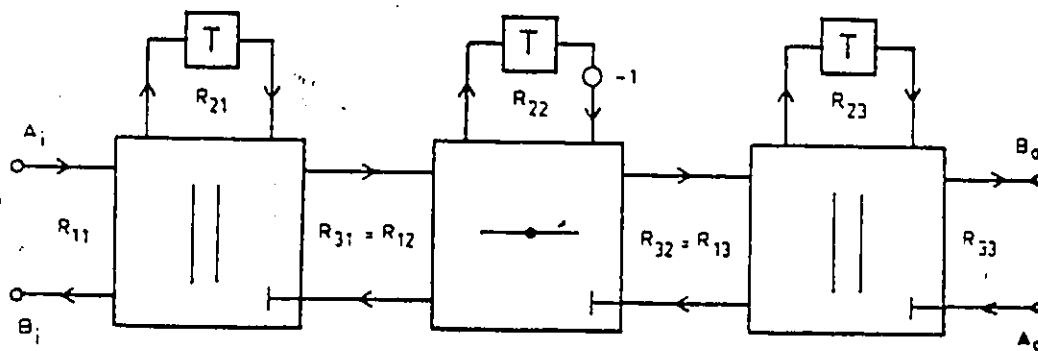
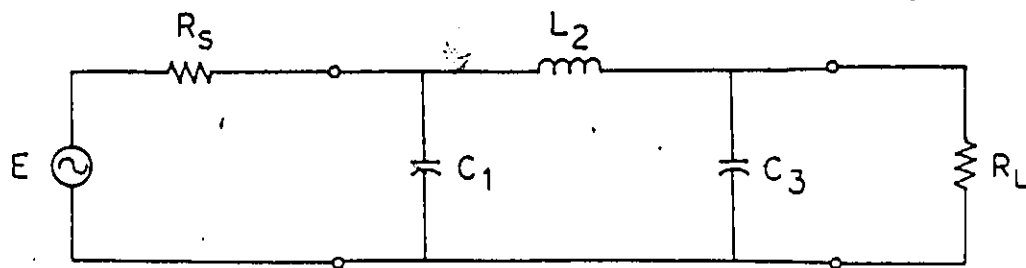


Figure 4.14: Block diagram of the 3rd order wave digital filter.

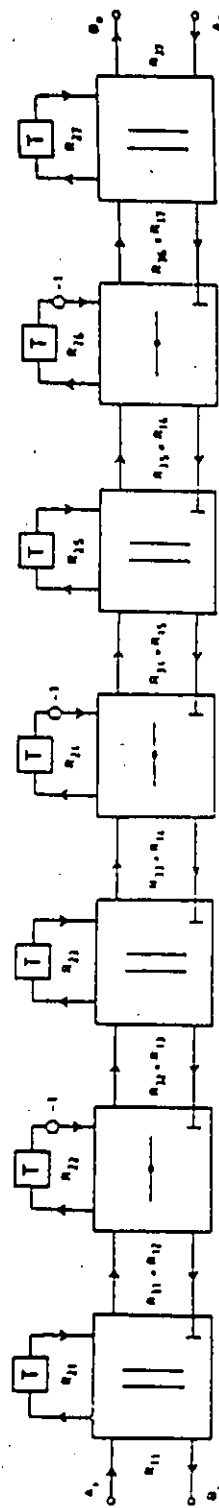
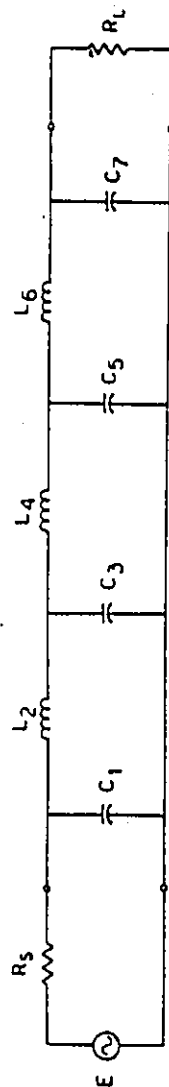


Figure 4.15. Block diagram of the 7th order wave digital filter.

	3rd order		5th order		7th order	
	WDF	SPDF	WDF	SPDF	WDF	SPDF
Adders	12	6	20	10	28	14
Multipliers	5	7	7	11	9	15
Delay Elements	3	6	5	10	7	14

Table 4.1: Hardware Required by WDFs and SPDFs of the "direct form 1"

	3rd order		5th order		7th order	
	WDF	SPDF	WDF	SPDF	WDF	SPDF
Multiplications	4	1	6	1	8	1
Additions	9	3	16	4	23	4
Total	13	4	22	5	31	5

Table 4.2: The Longest Sequence of Arithmetic Operations To Be Completed in One Cycle (the SPDFs are realized in the "direct form 1")

As an example, Fig. 4.16 shows the speed comparison chart for the WDFs and the SPDFs for signal wordlength of 8 bits, where typical time needed for multiplication and addition in commercially available ROMs and TTL adders are approximately 35 ns and 25 ns respectively. It is assumed that the time needed for the multiplication in the WDFs is also 35 ns. Notice that the filtering speed in SPDFs realized in high speed structure (Figure 3.13) is independent of the filter length. This chart shows the speed advantage of stored-product digital filters over wave digital filters.

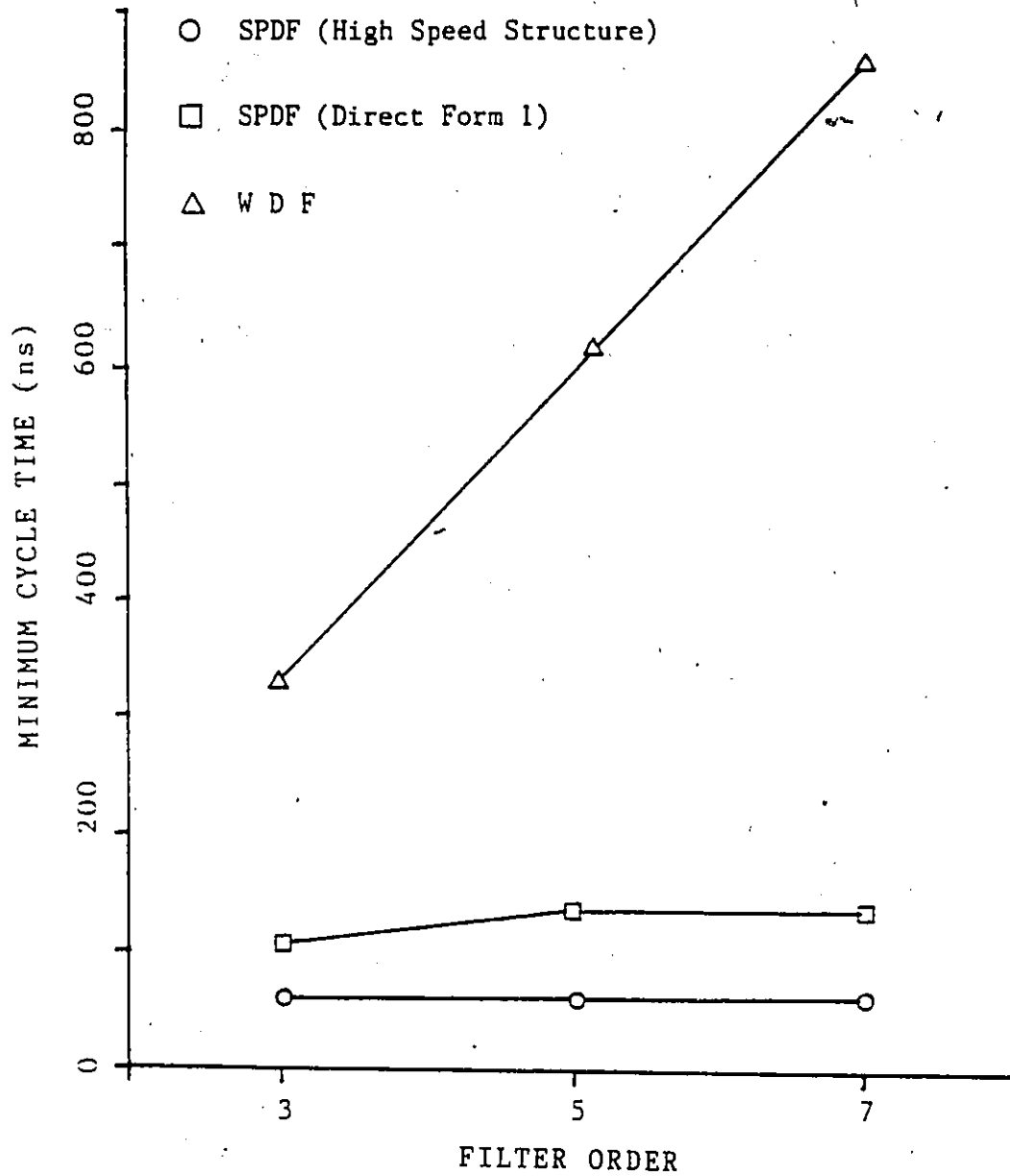


Figure 4.16: Speed comparison for wave and stored-product digital filters.

Chapter V

STORED-PRODUCT DIGITAL FILTERS WITH ERROR SPECTRUM SHAPING

5.1 Introduction

Basically, the stored-product digital filtering technique described in chapter III eliminates the effects of coefficient quantization errors from the performance of a digital filter and gives a slight reduction in the output signal error. The output signal error can be reduced further by increasing the product wordlength. However, due to signal recursion in the filter, this approach requires an increase in the wordlength of ROM address codes, thus increases ROM size requirement. Since the bit-slice technique described in chapter II trades off ROM size required with filtering speed, therefore, it is necessary to explore an alternative for reducing output signal error that does not reduce the filtering speed.

In this chapter, a simple technique for reducing error in the output signal of stored-product digital filters without decreasing filtering speed is presented. This technique is based on an error cancelling technique, where the existing error at the filter's output is cancelled with its own replica. The error replica is generated using the same transfer function as seen by the product roundoff error to the filter's output. In principle, this technique is equivalent to extending the filter's internal wordlength with arithmetic operations done in bit-slice fashion [24]. It will be shown that this technique reduces the error in filter's output signal substantially.

Since accumulation of roundoff errors in a digital filter exists in recursive part of the filter, throughout this chapter the discussion will be emphasized in that part only. Second order filters will be used as examples and can be used to generalize the effectiveness of this technique for higher order filters.

5.2 Approach

Consider the recursive section of an N-th order digital filter shown in Fig. 5.1-a, with the transfer function

$$H(z) = \frac{1}{1 + \sum_{k=1}^N b_k z^{-k}} \quad (5.1)$$

Assuming that the input and output signal wordlengths are L_s , and the product wordlength is L_p ($L_p > L_s$), the quantization of the sum of the products introduces a roundoff error $e(n)$, so that the actual output signal of the filter can be written as

$$v(n) = y(n) + e(n) \quad (5.2)$$

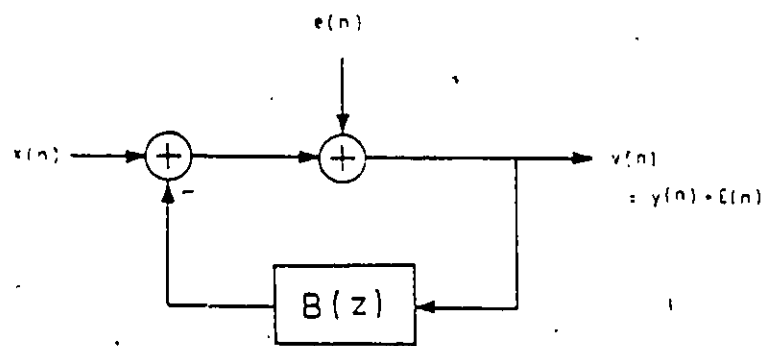
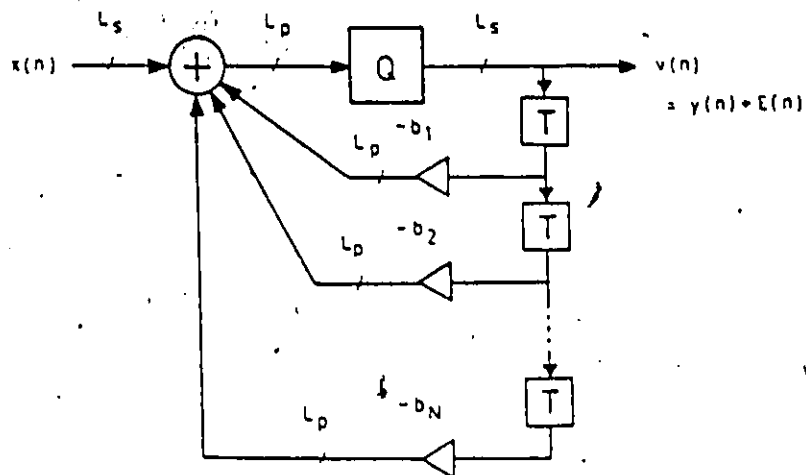
where $y(n)$ is the ideal output signal, and $e(n)$ is the output error due to propagation of the roundoff error $e(n)$ within the filter.

Notice that, during each filter cycle, the quantity of $e(n)$ can be computed by taking the difference between the quantities at the input and the output of quantizer Q. Since the transfer function from the error source $e(n)$ to the filter's output is known, i.e. equal to (5.1), a replica of $e(n)$ can be formed. Thus, the output error $e(n)$ can be cancelled by sub-

strating this replica from the output signal $v(n)$. This error cancelling technique is illustrated in Fig. 5.2. The final output signal of the filter is then

$$w(n) = y(n) + \epsilon(n) - \hat{\epsilon}(n) \quad (5.3)$$

Mathematically, the output error $\epsilon(n)$ will be cancelled completely if $\hat{\epsilon}(n)$ in (5.3) is an exact replica of $\epsilon(n)$. This technique will be analyzed in the next section.



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

$$B(z) = \sum_{k=1}^N b_k z^{-k}$$

Figure 5.1:

(a) Recursive section of an N -th order digital filter. (b) Error model of the filter.

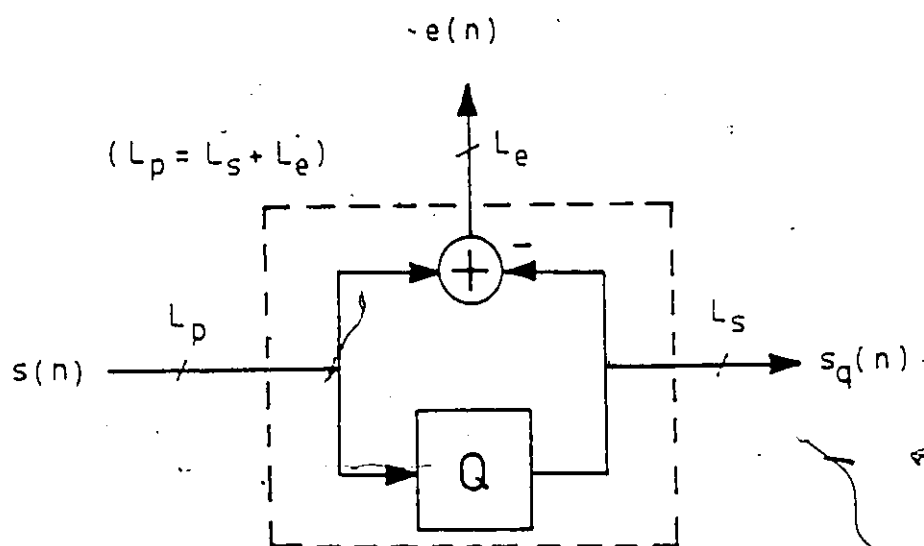
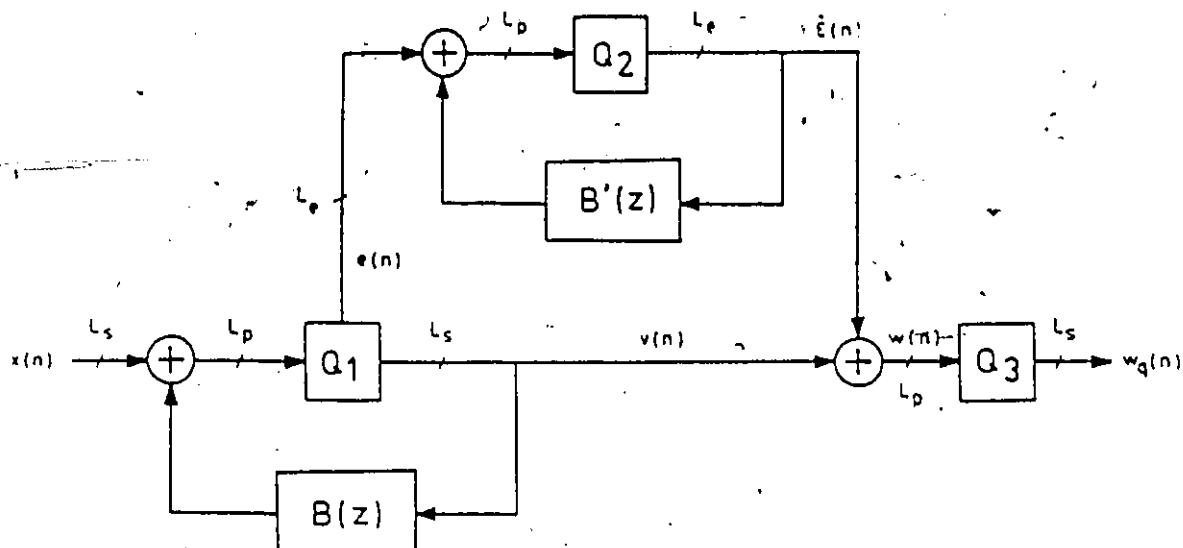


Figure 5.2:

(a) Block diagram of the filter with error cancelling technique. (b) Realization of the quantizer Q_1 .

5.3 Analysis of The Error Cancelling Technique

Consider the filter shown in Fig. 5.1-a. From (3.12) and (3.13), the variance of output error of this filter is

$$\sigma_{\epsilon}^2 = \frac{2^{-2L_s}}{12} H_0 \quad (5.4)$$

where H_0 is the roundoff error gain and is given by

$$H_0 = \frac{1}{2\pi j} \oint H(z) H(z^{-1}) z^{-1} dz \quad (5.5)$$

When the filter is implemented using the error cancelling technique as shown in Fig. 5.2-a, the roundoff error $\epsilon(n)$ obtained from the quantizer Q1 is fed to a filter with transfer function $H'(z)$, which is identical to $H(z)$. However, the existence of the quantizer Q2 causes the replica generated to be only an approximation of $\epsilon(n)$, which can be expressed as

$$\hat{\epsilon}(n) = \epsilon(n) + \epsilon'(n) \quad (5.6)$$

where $\epsilon'(n)$ is the difference between $\epsilon(n)$ and $\hat{\epsilon}(n)$. Therefore, the output signal obtained after the error cancellation is

$$w(n) = y(n) - \epsilon'(n) \quad (5.7)$$

Notice that, $e(n)$ and $\hat{e}(n)$ in Fig. 5.2 are shown to be of wordlength L_e bits. Actually both are of wordlengths $L_s + L_e$ bits with a string of zeros in the first L_s most significant bits. Subsequently, the resulting output signal $w(n)$ is also of wordlength $L_s + L_e$ bits, and the variance of error in the output signal $w(n)$ essentially is given by

$$\sigma_{\epsilon'}^2 = \frac{2^{-2(L_s + L_e)}}{12} H_0 \quad (5.8)$$

Suppose the desired wordlength in the output signal of the filter is L_s bits, i.e. the same as the input signal wordlength, then another quantizer (the quantizer Q3 in Fig. 5.2) is introduced at the output. This gives the final output signal $w_q(n)$, which can be expressed as

$$\begin{aligned} w_q(n) &= w(n) + \epsilon''(n) \\ &= y(n) - \hat{\epsilon}(n) \end{aligned} \quad (5.9)$$

where $\hat{\epsilon}(n)$ is the total error in the output signal, $\epsilon''(n)$ is the error introduced by the quantizer Q3 and has the variance given by

$$\sigma_{\epsilon''}^2 = \frac{2^{-2L_s}}{12} \quad (5.10)$$

Therefore, the variance of the final output error will be

$$\begin{aligned}
 \sigma_{\delta}^2 &= \sigma_{\epsilon'}^2 + \sigma_{\epsilon''}^2 \\
 &= \frac{2^{-2L_s}}{12} (2^{-2L_e} H_o + 1) \quad (5.11) \\
 &= \frac{2^{-2L_s}}{12} H_o'
 \end{aligned}$$

Define the reduction factor in roundoff error gain as the ratio of the initial and the final roundoff error gains, that is

$$r = \frac{H_o}{H_o'} \quad (5.12)$$

Substituting (5.11) into (5.12) yields

$$r = \frac{H_o}{\frac{2^{-2L_e} H_o + 1}{2^{-2L_s}}} \quad (5.13)$$

and

$$r < H_o \quad (5.14)$$

It can be seen that r is dependent on L_e . Hence, the choice of wordlength L_e to accommodate the roundoff error $\epsilon(n)$ determines the performance of this error cancelling technique.

In general, the wordlength L_e can be formulated as a function of the initial roundoff error gain H_0 and the desired reduction factor r , and is given by

$$L_e = -\frac{1}{2} \log_2 \left(\frac{1}{r} - \frac{1}{H_0} \right) \quad (5.15)$$

By using (5.15), the necessary wordlength L_e that reduces the final output error to the desired level can be determined.

To give an illustration for the performance of the error cancelling technique, consider a second order all pole filter, i.e. a filter consisting the recursive section only. Two sets of coefficients representing different Q 's are given for these examples (Tables 5.1 and 5.2). It will be assumed that L_s and L_e are 8 bits. This technique is evaluated by computer simulation using a random sequence, uniformly distributed between -1.0 and +1.0, for the input signal. It can be seen from Tables 5.1 and 5.2 the output error of the filter is reduced substantially.

	INITIAL OUTPUT ERROR	FINAL OUTPUT ERROR	REDUCTION FACTOR r
THEORETICAL	-45.21 dB	-58.96 dB	13.75 dB
SIMULATION	-41.55 dB	-58.99 dB	17.44 dB

Table 5.1: Output Error Variance of The 2nd Order All-Pole Filter
($b_1 = -1.5$, $b_2 = 0.875$; $Q = 4.8$)

	INITIAL OUTPUT ERROR	FINAL OUTPUT ERROR	REDUCTION FACTOR r
THEORETICAL	-35.84 dB	-58.94 dB	23.10 dB
SIMULATION	-37.10 dB	-58.86 dB	21.76 dB

Table 5.2: Output Error Variance of The 2nd Order All-Pole Filter
($b_1 = -1.516$, $b_2 = 0.988$; $Q = 59.7$)

5.4 Architectures For SPDFs With Error Spectrum Shaping

Basically, error spectrum shaping reduces product roundoff error accumulation by extending the product wordlength and operates in bit-slice fashion. Thus, digital filters incorporating error spectrum shaping can be implemented in either parallel or sequential architectures.

Shown in Fig. 5.3 is the parallel implementation of a second order SPDF incorporating the error cancelling technique. Basically, it is similar to the split address SPDF described in [15]. The only difference is that, in the split address SPDF, the longest sequence of arithmetic operations between two unit-delay elements increases to one multiplication and two additions. This is because the products from the ROM addressed by the msb's are added to the products from the ROM addressed by the lsb's directly at the output of the ROMs (Fig. 2.7). On the other hand, in the implementation shown in Fig. 5.3 the arithmetic operations for the msb's are done separately from those for the lsb's. Therefore, the longest sequence of arithmetic operations between two unit-delays elements is kept to a minimum, i.e. one multiplication and one addition.

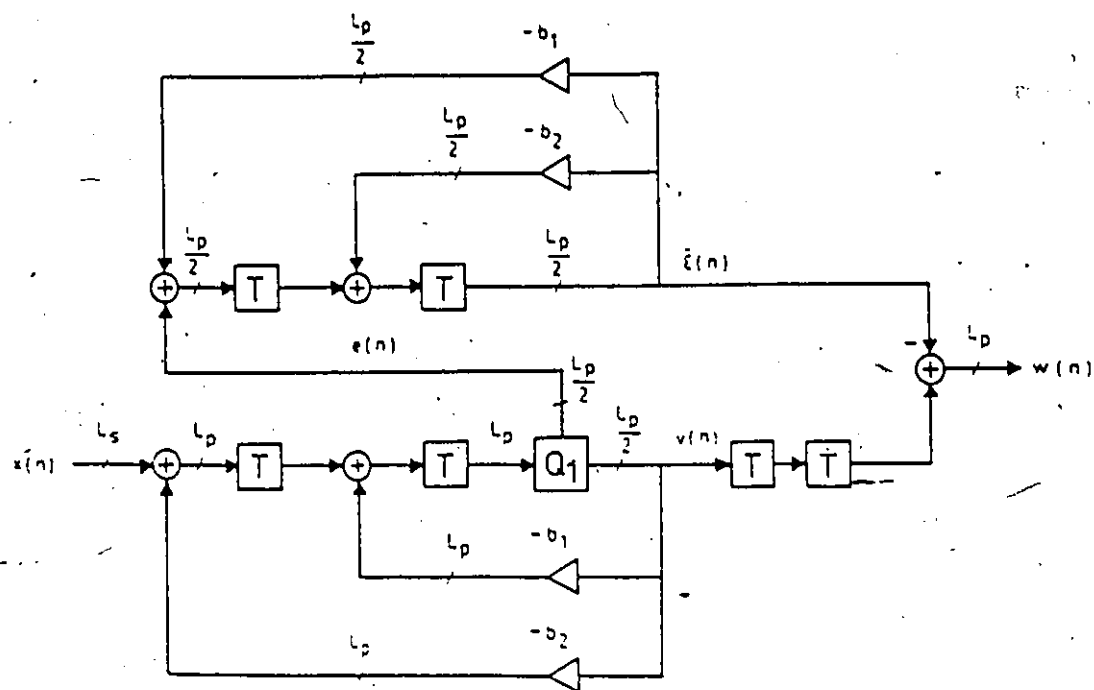


Figure 5.3: Parallel architecture of a high speed SPDF with ESS.

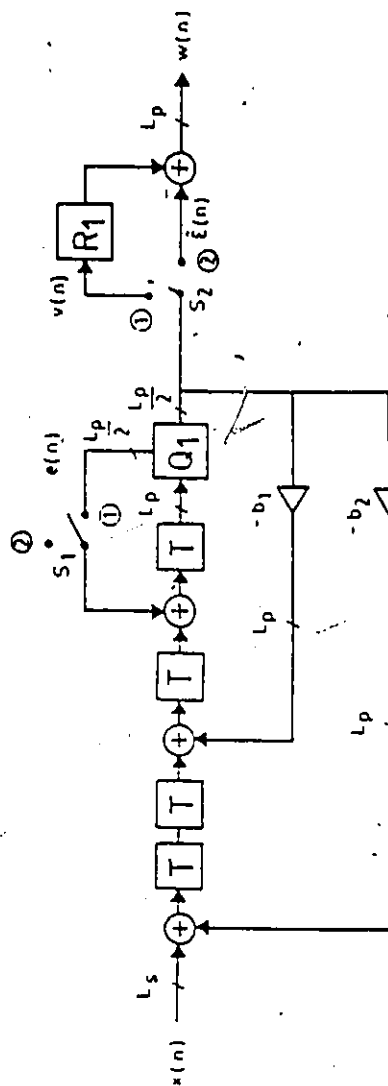


Figure 5.4: Sequential architecture of a high speed SPDF with ESS.

The sequential architecture of a second order SPDF incorporating the error cancelling technique is shown in Fig. 5.4. Here a reduction in the ROM size required is obtained by time-sharing the same ROM, for the msb's and the lsb's. However, since the filter is alternately used for the actual filtering operation and for the output error replica generation, the maximum throughput rate that can be handled by the system decreases to one half of the maximum throughput rates for the parallel architecture. Notice that, during the first cycle, the filter processes the input signal. The switches S1 and S2 are at position 1. The roundoff error $e(n)$ is computed and fed back as shown in Fig. 5.4 and the output signal $v(n)$ is saved in the latch R1. During the second cycle, the switches S1 and S2 are at position 2, and the roundoff error $e(n)$ is processed by the filter. The output error replica $\hat{e}(n)$ is subtracted from the output signal $v(n)$ from the latch R1. Therefore, the output error $e(n)$ is cancelled.

Chapter VI

CONCLUSIONS

In this thesis, the use of stored-product ROMs as a replacement for conventional binary multipliers in digital filters has been studied. This technique, also known as the stored-product digital filtering, requires fixed filter coefficients to determine the stored product values. It has advantages over multiplier based digital filters, such as high filtering speed and minimal hardware complexity, and can be applied to any filter structures.

The stored-product ROM multiplication technique has been evaluated for fixed point and for floating point arithmetic. For either type of arithmetic, this technique exhibits lower product error due to the elimination of coefficient error. In general, the product error obtained from a stored-product ROM multiplier is controllable by the choice of product wordlength used. While in a conventional hardware multiplier with short coefficient wordlength, it is dependent on the product and coefficient wordlengths used and it is also influenced by the signal power.

For an address wordlength longer than 12 bits, the ROM storage size requirement can be reduced through the use of the nonlinear quantization technique, and also using the bit-slice technique. A large reduction in the ROM storage size can be obtained using the sequential bit-slice architecture by trading off the multiplication speed.

The use of stored-product ROM multipliers in a digital filter increases the accuracy of the filter coefficients. Therefore, the frequency response accuracy of a stored-product digi-

tal filter is very high and is it close to that of the original design. It has been shown that the stored-product digital filtering also reduces product roundoff error accumulation. Furthermore, the computational speed of stored-product digital filters is very high. With the transposed configuration of "direct form 2", stored-product digital filters can be used for filtering signals with sampling frequencies in the range 10 MHz or more.

Compared to wave digital filters, stored-product digital filters show several advantages. The frequency response accuracy in stored-product digital filter is very high, and that it requires to use coefficient wordlength of about 24 bits for wave digital filters to achieve equal accuracy. In terms of product roundoff error accumulation, stored-product digital filters also show an advantage, as the product roundoff error accumulation in wave digital filters is dependent on the coefficient and product wordlengths used. Wave digital filters are intended for the design using short coefficient wordlengths. However, only those with rather long coefficient wordlengths have lower product roundoff error accumulation than stored-product digital filters. Finally, stored-product digital filters exhibit higher speed and require less hardware than wave digital filters.

The application of an error cancelling technique, known as error spectrum shaping, to stored-product digital filters is evaluated. It is shown that this technique reduces the accumulation of product roundoff errors in stored-product digital filters substantially, without affecting filtering speed.

It is clear that the stored-product digital filtering technique provides a solution for achieving fast and accurate filters at low cost. As large ROMs are now available at low cost, it is expected that the stored-product digital filtering will become a major method for designing digital filters.

Appendix A

COEFFICIENTS OF THE DIRECT FORM DIGITAL FILTERS

3-rd order filter:

$$\begin{aligned}a_0 &= 0.227306 \\a_1 &= 0.681919 \\a_2 &= 0.681919 \\a_3 &= 0.227306\end{aligned}$$

$$\begin{aligned}b_1 &= 0.361464 \\b_2 &= 0.464291 \\b_3 &= -0.007304\end{aligned}$$

5-th order filter:

$$\begin{aligned}a_0 &= 0.041972 \\a_1 &= 0.209862 \\a_2 &= 0.419725 \\a_3 &= 0.419725 \\a_4 &= 0.209862 \\a_5 &= 0.041972\end{aligned}$$

$$\begin{aligned}b_1 &= -0.435742 \\b_2 &= 1.037118 \\b_3 &= -0.465380 \\b_4 &= 0.274341 \\b_5 &= -0.067218\end{aligned}$$

7-th order filter:

$$\begin{aligned}a_0 &= 0.007366 \\a_1 &= 0.051564 \\a_2 &= 0.154692 \\a_3 &= 0.257820 \\a_4 &= 0.257820 \\a_5 &= 0.154692 \\a_6 &= 0.051564 \\a_7 &= 0.007366\end{aligned}$$

$$\begin{aligned}b_1 &= -1.259612 \\b_2 &= 2.193684 \\b_3 &= -1.955560 \\b_4 &= 1.554096 \\b_5 &= -0.846151 \\b_6 &= 0.328569 \\b_7 &= -0.072142\end{aligned}$$

Appendix B

COEFFICIENTS OF THE WAVE DIGITAL FILTERS

3-rd order filter:

$$c_{11} = 0.492223$$

$$c_{13} = 0.461781$$

$$c_{12} = 0.300205$$

$$c_{23} = 0.757167$$

5-th order filter:

$$c_{11} = 0.465810$$

$$c_{14} = 0.224490$$

$$c_{12} = 0.253569$$

$$c_{15} = 0.417030$$

$$c_{13} = 0.216072$$

$$c_{25} = 0.737362$$

7-th order filter:

$$c_{11} = 0.458463$$

$$c_{15} = 0.196205$$

$$c_{12} = 0.243700$$

$$c_{16} = 0.212252$$

$$c_{13} = 0.202247$$

$$c_{17} = 0.404891$$

$$c_{14} = 0.194731$$

$$c_{27} = 0.731299$$

** c_{jk} denotes the j -th coefficient of the k -th adaptor.

BIBLIOGRAPHY

1. A. Fettweiss : "Digital Filter Structures Related to Classical Filter Networks", Arch. Electron. Uebertrag., vol. 25, pp 79-89, Feb. 1971.
2. A. Sedlemeyer and A. Fettweiss : "Digital Filters With True Ladder Configuration", Int. J. Circuit Theory Appl., vol. 1, pp 5-10, Mar. 1973.
3. A. Fettweiss and K. Meerkotter : "On Adaptors for Wave Digital Filters", IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-23, pp 516-525, Dec. 1975.
4. R.E. Crochiere : "Digital Ladder Structures and Coefficient Sensitivity", IEEE Trans. Audio Electroacoust., vol. AU-20, pp 240-246, Oct. 1972.
5. A. Fettweiss : "On The Connection Between Multiplier Wordlength Limitation and Roundoff Noise in Digital Filters", IEEE Trans. Circuit Theory, vol. CT-19, pp 486-491, Sep. 1972.
6. D. Van Haften and P.M. Chirlian : "An Analysis of Errors in Wave Digital Filters", IEEE Trans. Circuits Syst., vol. CAS-28, pp 154-160, Feb. 1981.
7. S.A. White : "ROM-Based Parallel Digital Arithmetic Device", U.S. Patent 4 313 174, Jan. 26 1982.
8. T. Tjahjadi and W. Steenaart : "On The Accuracy of ROM Stored Square Multipliers", IEEE Trans. Circuits Syst., vol. CAS-29, pp 441-447, Jul. 1982.
9. T.A. Brubaker and J.C. Becker : "Multiplication Using Logarithms Implemented with Read-Only Memory", IEEE Trans. Comput., vol. C-24, pp 761-765, Aug. 1975.
10. 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.

11. A. Croisier, D.J. Esteban, M.E. Levilion and V. Rizo : "Digital Filters for PCM encoded Signals", U.S. Patent 3 777 130, Dec. 3 1973.
12. O. Monkewich and W. Steenaart : Stored Product Digital Filtering With Nonlinear Quantization", Proc. 1976 IEEE Int. Symp. Circuits Syst., pp 157-160.
13. W. Steenaart, D. Dubois and O. Monkewich : "Stored-Product Digital Filtering: Structures, Potential and Applications", Proc. 1981 European Conference on Circuit Theory and Design, pp 118-126.
14. D. Dubois and W. Steenaart : "Coefficient Accuracy and Error Accumulation in Stored-Product Digital Filters" Colloque sur le Traitement du Signal, Conference Proceedings pp 391-394, 1983.
15. D. Dubois and W. Steenaart : "High Speed Stored-Product Digital Filters", IEEE Trans. Circuits Syst., vol. CAS-29, pp 390-393, June 1982.
16. H. Tatangsurja and W. Steenaart : "Bit Slice Stored-Product Digital Filters", Proc. 13th Biennial Symp. on Communications, pp B4.14-B4.17, 1986.
17. H. Tatangsurja and W. Steenaart : "A Comparison of Stored-Product and Wave Digital Filters: Coefficient Accuracy and Roundoff Noise", Proc. 27th Midwest Symp. on Circuits Syst., pp 612-614, 1984.
18. B. Liu : "Effect of Finite Wordlength On The Accuracy of Digital Filters - A Review", IEEE Trans. Circuit Theory, vol. CT-18, pp 670-677, Nov. 1971.
19. J.B. Knowles and E.M. Olcayto : "Coefficient Accuracy and Digital Filter Response", IEEE Trans. Circuit Theory, vol. CT-15, pp 31-41, Mar. 1968.
20. A. Antoniou : "Digital Filters: Analysis and Design", McGraw-Hill, New York, 1979.
21. L.R. Rabiner and B. Gold : "Theory and Application of Digital Signal Processing", Prentice-Hall, New Jersey, 1975.

22. A.I. Abu-El-Haija and A.M. Peterson : "An Approach to Eliminate Roundoff Errors in Digital Filters", IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-27, pp 195-198, Apr. 1979.
23. T.L. Chang and S.A. White : "An Error Cancellation Digital Filter Structure and Its Distributic Arithmetic Implementation", IEEE Trans. Circuits Syst., vol. CAS-28, pp 339-342, Apr. 1981.
24. C.T. Mullis and R.A. Roberts : "An Interpretation of Error Spectrum Shaping in Digital Filters", IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-30, pp 1013-1015, Dec. 1982.