

**THE DESIGN AND IMPLEMENTATION OF
DIGITAL WAVE FILTER ADAPTORS**

NEIL PETRIE B.Sc.

**Doctor of Philosophy
University of Edinburgh**

August 1985



ABSTRACT

In the field of telecommunications, there is a growing need for high performance electronic filters. Initially, these circuits were realised as ladder structures composed of lumped or distributed filter elements which are passive, unconditionally stable, and which are insensitive to small variations in component value. However, these filters suffer from large physical size and a drift in component values due to time and temperature. Clearly, an integrated electronic filter which retains the advantages of such structures, whilst offering a response which will not drift, is highly desirable.

In this thesis, a number of circuits will be developed which are capable of modelling a wide range of distributed ladder reference filters using digital techniques. These structures, based on the wave digital filter, have been designed specifically for realisation as integrated circuits. In particular, two general purpose wave filter circuits, which may be externally programmed, will be presented. These structures can be directly cascaded to realise filters of any order.

A silicon integrated circuit, based on one of these structures, has been designed and fabricated in NMOS technology. Results of a practical filter implemented using this integrated circuit will be presented, and show close agreement with the expected response. This circuit has been further developed to produce a more efficient filter structure which can be multiplexed across one or more filters. This novel structure, called a *universal adaptor element*, has been extensively simulated and results are presented for filters based on this concept, showing its insensitivity to coefficient variation.

Declaration of Originality

This thesis, composed entirely by myself, reports work carried out in the Department of Electrical Engineering at the University of Edinburgh predominantly by myself. The sole exception to this is the silicon layout of the integrated adaptor circuit, chip number Eu372, presented in Chapter 6. This work was carried out by Mr. C. H. Lau of the Department of Electrical Engineering, University of Edinburgh, based on a design presented by myself.

9th August 1985

Acknowledgements

The author gratefully acknowledges the help and encouragement extended to him by his supervisor, Professor John Mavor.

The valuable assistance of many colleagues in the Department of Electrical Engineering has been greatly appreciated. In particular, the help of Drs. P. B. Denyer, H. M. Reekie and D. Renshaw has been of great benefit.

The authors work, reported in this thesis, has been funded by the Science and Engineering Research Council, to whom thanks are also extended.

Finally, thanks are due to my family for their help and encouragement during the course of this work.

Table of Contents

| | |
|---|-----|
| Abstract | i |
| Declaration of Originality | ii |
| Acknowledgements | iii |
| Table of Contents | iv |
| CHAPTER 1 : INTRODUCTION | |
| 1.1 Introduction | 1 |
| 1.2 Passive Filters | 2 |
| 1.2.1 RLC ladder filters | 3 |
| 1.2.2 Alternative forms of passive filter | 6 |
| 1.3 Sampled-Data Filters | 8 |
| 1.3.1 Bilinear z-transform | 10 |
| 1.4 Switched Capacitor Filters | 11 |
| 1.5 Digital Filters | 14 |
| 1.5.1 Non-recursive digital filters | 15 |
| 1.6 Introduction to Wave Filters | 17 |
| 1.7 Thesis Objectives | 18 |
| CHAPTER 2 : WAVE FILTER THEORY | 21 |
| 2.1 Introduction | 21 |
| 2.2 Choice of Frequency Variable | 22 |
| 2.3 Incident and Reflected Waves | 23 |
| 2.4 Circuit elements | 24 |
| 2.4.1 Inductor | 24 |
| 2.4.2 Capacitor | 26 |
| 2.4.3 Resistor | 27 |
| 2.4.4 Short circuit | 27 |
| 2.4.5 Open circuit | 28 |
| 2.4.6 Transmission line or unit-element | 28 |
| 2.4.7 FDNR elements | 30 |
| 2.4.7.1 Super capacitors | 32 |
| 2.4.7.2 Super inductors | 33 |
| 2.5 Interconnection of Circuit Elements | 34 |
| 2.5.1 Two-port adaptors | 35 |
| 2.5.2 Multi-port parallel adaptors | 37 |
| 2.5.3 Multi-port series adaptors | 38 |
| 2.6 Cascaded Two-Port Adaptor Filters | 39 |
| 2.7 Wave Filter Implementations of Ladder Structures | 42 |
| 2.7.1 Wave filter design using Kuroda Transforms | 43 |
| 2.7.2 Wave filter design using synthesis techniques | 46 |
| 2.7.3 Wave filter design using matched port techniques | 48 |
| 2.7.4 Wave lattice filters | 51 |
| 2.8 Summary | 53 |

| | |
|--|-----|
| CHAPTER 3 : WAVE FILTER IMPLEMENTATION | 56 |
| 3.1 Introduction | 56 |
| 3.2 Analogue Wave Filters | 57 |
| 3.3 Digital Wave Filters | 58 |
| 3.3.1 DWF implementation using general purpose hardware | 67 |
| 3.3.2 Integrated digital wave filters | 75 |
| 3.3.3 Wave filters implemented using FDNR elements | 85 |
| 3.4 Summary | 89 |
| CHAPTER 4 : DIGITAL WAVE FILTER ADAPTOR REALISATIONS | 92 |
| 4.1 Introduction | 92 |
| 4.1.1 Signal representation | 93 |
| 4.2 Bit-Serial Arithmetic | 95 |
| 4.2.1 The FIRST Silicon Compiler | 98 |
| 4.3 Two-Port Adaptors | 102 |
| 4.3.1 Hardware requirements | 102 |
| 4.3.2 Adaptor latency, sample rate and signal wordlength | 103 |
| 4.4 Three-Port Adaptors | 107 |
| 4.4.1 Parallel three-port adaptors | 107 |
| 4.4.2 Series three-port adaptors | 109 |
| 4.4.3 Adaptor latency, sample rate and signal wordlength | 111 |
| 4.5 General Purpose Wave Filter Adaptor | 114 |
| 4.5.1 Adaptor latency, sample rate and signal wordlength | 115 |
| 4.6 Universal Wave Filter Adaptor | 116 |
| 4.6.1 Adaptor latency, sample rate and signal wordlength | 121 |
| 4.7 Summary | 122 |
| CHAPTER 5 : SIMULATION OF DIGITAL WAVE FILTERS | 125 |
| 5.1 Introduction | 125 |
| 5.2 Reference Filter for Simulation of Filter System | 126 |
| 5.3 Signal and Control Information | 132 |
| 5.3.1 Signal information | 132 |
| 5.3.2 Control information | 135 |
| 5.4 Simulation Results | 138 |
| 5.4.1 12 bit multiplier coefficient | 140 |
| 5.4.2 10 bit multiplier coefficient | 143 |
| 5.4.3 8 bit multiplier coefficient | 146 |
| 5.4.4 6 bit multiplier coefficient | 149 |
| 5.4.5 4 bit multiplier coefficient | 152 |
| 5.4.6 2 bit multiplier coefficient | 155 |
| 5.5 Insertion Loss, Signal Scaling and Suppression of Parasitic oscillations | 159 |
| 5.5.1 Insertion loss in wave filters | 159 |
| 5.5.2 Signal Scaling | 160 |
| 5.5.3 Suppression of parasitic oscillations | 162 |

| | |
|---|-----|
| 5.6 Summary | 165 |
| CHAPTER 6 : NMOS REALISATION OF WAVE FILTER ADAPTOR | 167 |
| 6.1 Introduction | 167 |
| 6.2 Binary Multiplication | 168 |
| 6.2.1 CSD multiplier | 170 |
| 6.2.2 Modified Booth's algorithm multiplier | 172 |
| 6.2.3 Pipelined multipliers | 174 |
| 6.3 Circuit Elements for Integrated Wave Filter Adaptor | 176 |
| 6.3.1 Multiplexer circuit | 176 |
| 6.3.2 Bit-delay circuit | 178 |
| 6.3.3 Binary addition | 180 |
| 6.3.4 Binary mutliplier circuit | 183 |
| 6.3.5 Integrated adaptor circuit | 183 |
| 6.4 Realisation of Example Filter | 185 |
| 6.4.1 Example reference filter | 186 |
| 6.4.2 Test circuit requirement | 187 |
| 6.4.3 Practical results | 191 |
| 6.5 Summary | 193 |
| CHAPTER 7 : CONCLUSIONS | 195 |
| 7.1 Summary | 195 |
| 7.2 Bit-Serial Wave Filters | 197 |
| 7.3 Applications of Digital Wave Filters | 201 |
| 7.4 Suggestions for Future Work | 204 |
| APPENDIX I : COMPUTER AIDED FILTER SYNTHESIS | 206 |
| APPENDIX II : FIRST COMPILER FILE FOR UNIVERSAL ADAPTOR | 210 |
| II.1 Introduction | 210 |
| II.2 FIRST Language File for Universal Adaptor | 212 |
| APPENDIX III : CANDIDATES PUBLISHED WORK | 217 |
| APPENDIX IV : REFERENCES | 228 |

Chapter 1

INTRODUCTION

1.1 Introduction

In modern electronic systems, there is a growing need for high performance electronic filters. In the past, these filters were realised using conventional, discrete components. However, the advance of silicon integrated circuit processing techniques has permitted the development of alternative filter structures using digital electronic methods. These alternative structures are based on a variety of filter algorithms, and offer several advantages over previous forms of implementation, including linear- phase characteristics, the ability to accurately filter low frequency signals, a lack of drift in the filter response and small physical size.

Originally, integrated filters were designed to satisfy pre-determined filter responses for specific applications. In this thesis, general purpose circuits will be presented which permit a wide variety of filter responses to be realised. These circuits are based on one particular type of digital filter, known as the wave digital filter. An integrated circuit has been designed to realise one of these structures, and practical results from this design are presented.

In this chapter, a general background to filtering will be given, with an emphasis on the variety of filter types that are currently being used to meet the need of filter designers. These filter types can be divided into two main sub-groups of continuous-time filters and sampled-data filters. Within these groups, a variety of filter types are available; including passive filters, active filters, switched-capacitor filters and digital filters. These filter groups are considered separately and also a comparison between the different ways in which filters may be designed to meet the desired response is presented.

1.2 Passive Filters

Passive filters are defined as a linear network in which the energy delivered to the network is non-negative for any arbitrary excitation, and if no voltages or currents appear between any two terminals before an excitation is applied [1]. Passive filters are continuous signal filters, and therefore the input variable $x(t)$, and the output variable $y(t)$ are related by the convolution integral

$$y(t) = \int_0^{\infty} h(t - \tau)x(\tau) d\tau \quad (1.1)$$

where $h(t - \tau)$ is the impulse response of the network to some excitation at time t . This time domain relationship

can be converted to the frequency domain through the Laplace transform, giving

$$Y(s) = H(s) X(s) \quad (1.2)$$

If the network is composed of lumped elements, such as resistors (R), inductors (L) or capacitors (C), then $H(s)$ will be a rational function, whose poles are the resonant frequencies of the system. If the filter network is to be stable, then these poles must lie in the left half of the complex s-plane. Complex frequencies for which $H(s) = 0$ are known as zeros of transmission [2], and are dependent on the topology of the filter.

The construction of passive filters is dependent on the range of frequencies over which the filter is to operate, and the required transfer function of the filter.

1.2.1 RLC ladder filters

RLC ladder filters are the dominant type of filter implementation in the frequency range from d.c. to 10MHz, although they are being widely replaced by integrated structures. They are designed using lumped resistors, inductors and capacitors and, because they are passive, they cannot become unstable [3].

Consider the reactance ladder filter shown in Figure

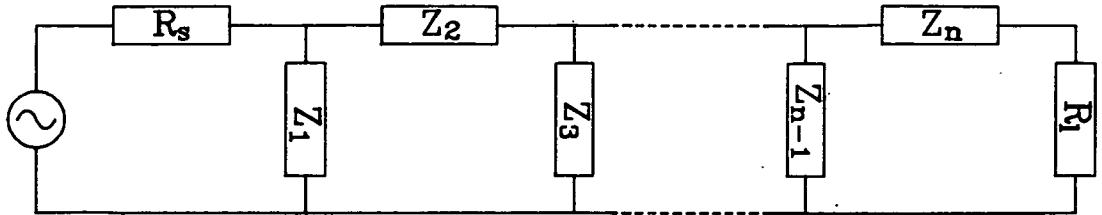


Figure 1-1: Resistively terminated ladder structure

1-1, terminated with source and load resistances R_s and R_L . If $R_s = R_L$, then the structure is maximally insensitive to small variations in the component values [4]. This insensitivity can be explained in terms of the power transmitted from the source to the load. In the filter passband, there are points at which the source and load impedances are matched by the reactance ladder network. These points are referred to as points of maximum power transmission as shown in Figure 1-2 (b). At these points, the rate of change of power transfer must be zero, and therefore small variations in the positions of these points are negligible. This property allows the implementation of large filter circuits with practical components.

Lumped RLC filters can realise lowpass, highpass, bandpass and bandstop frequency responses. The transfer function for these responses can take a variety of forms. The most widely used types of responses are the Butterworth, Chebyshev and elliptic responses. Butterworth responses are maximally flat in the passband. Chebyshev

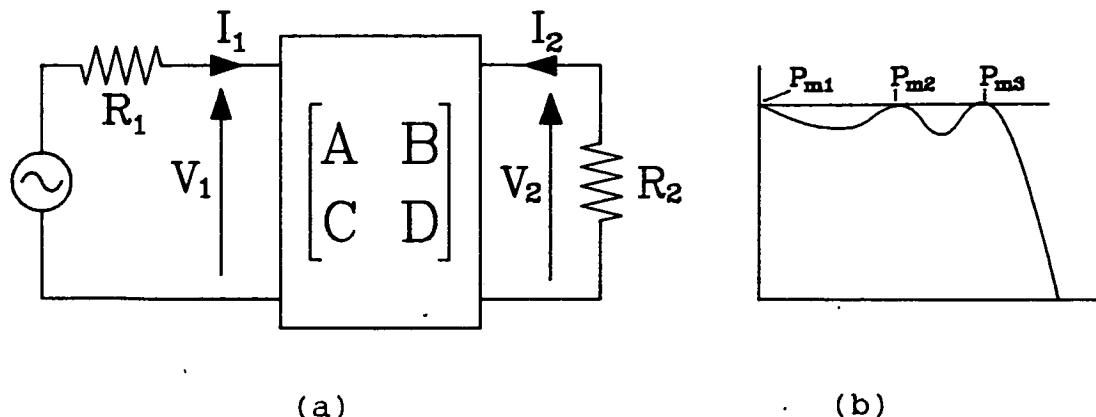


Figure 1-2: Power transfer response of a reactance network

filters offer a narrower transition range between pass and stop bands, but with the introduction of ripple in the passband. Elliptic filters have a Chebyshev response in both the passband and the stopband, and therefore offer even narrower transition ranges but with ripples in pass and stopbands. Tables of element values for these responses are available [5], and greatly simplify this type of filter design.

Filter transfer functions can be described in the form of a matrix, known as the *chain matrix*. This matrix relates the input voltage and current to the output voltage and current by

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \quad (1.3)$$

where V_1 , I_1 , V_2 and I_2 are as shown in Figure 1-2 (a).

Although these filters offer many benefits, they suffer from three major disadvantages.

1. They are physically large, especially at low frequencies. This inhibits their widespread use in applications where size is important, such as in portable communication systems.
2. The component values drift with time and temperature. Although the filters are insensitive to small variations, extremes of temperature can considerably affect their performance.
3. They cannot be used in situations where the filter characteristics are to be programmable, such as in adaptive systems for communications.

As integrated circuit techniques have been developed, filter designers have sought alternative forms of filter which are physically smaller and more versatile, while trying to retain the desirable characteristics of ladder filters. Examples of these developments are the active RC filter [6], and the wave filter [7].

1.2.2 Alternative forms of passive filter

In the frequency ranges beyond those covered by lumped element filters, two types of filter are widely

used; piezoelectric crystal filters and microwave filters.

In the frequency range between 5 and 150MHz, piezoelectric crystal filters are widely used. These crystals have very stable resonant frequencies, and when incorporated in filters, offer very small passband losses. For filters requiring sharp transitions between passband and stopband, these properties are clearly desirable.

Crystal filters are generally used for very narrow bandpass filters, where the passband width is a small fraction of a percent of the resonant frequency and large stopband attenuations are required. These filters are therefore not suitable for implementing lowpass and bandpass filters which generally require a wide band filter response. However, crystal units can be used to realise loss peaks extremely close to the passband edge of LC lowpass and highpass filters.

Microwave filters are generally considered to operate in the frequency range above 1GHz, although the design techniques used are applicable in the range 0.1 to 20GHz. The filtering elements involved in microwave filters are distributed inductors and capacitors, rather than the lumped elements of the filters described in section 1.2.1. A distributed network is defined as one in

which the physical dimensions of the elements are comparable to the wavelength of the signal inside the element [8].

Microwave filters are designed by using lumped element prototypes and transforming the reference filter to a distributed network by the Richards' transformation [9]:

$$\psi = \tanh sT \quad (1.4)$$

where s is the complex frequency variable. This has the effect of transforming lumped capacitors to open-circuit lines and inductors to short-circuit lines. Resistors, being frequency independent, remain as resistors in the ψ -plane. An additional effect of this transformation is that the distributed network is periodic in its response, with period π .

1.3 Sampled-Data Filters

In communications and control, there is often a requirement to transmit several independent signals along a single channel. Historically, this was achieved by using amplitude, frequency and phase modulating techniques. However, more recently, these methods have been further developed to permit time-domain multiplexing of the system by means of sampled-data systems [10]. These data samples can be transmitted as analogue signals or

converted to binary form for processing in purely digital systems.

Sampling may be considered as the multiplication of a continuous time signal, $x(t)$, by a train of impulses, $p(t)$, as shown in Figure 1-3.

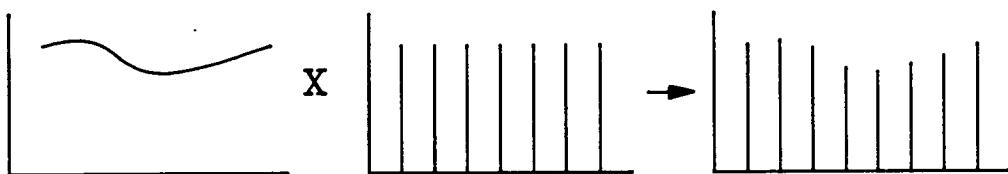


Figure 1-3: Effect of sampling on a continuous signal

The signal $p(t)$ is periodic, with an interval between samples of T , giving a sample frequency $F_s = 1/T$. The effect of sampling is to convert the continuous-time filters of the previous section, to discrete-time filters, where samples are processed individually. Apart from this effect, the transfer functions involved are essentially the same as those in passive systems.

At any point in a sampled-data system, the continuous signal input can only be reconstructed when the following conditions are satisfied.

1. The sampling frequency, F_s , must be more than twice the maximum frequency component of the input signal.

2. The reconstructed signal must be passed through a continuous time lowpass filter with a maximum cutoff frequency of $F_S/2$.

These conditions follow directly from the Shannon sampling theorem [10].

The maximum frequency, $F_S/2$, below which a signal can be recovered from a system sampling at F_S , is often referred to as the Nyquist frequency. Signals above this frequency, entering a sampled-data system, are reflected about the Nyquist frequency, causing unwanted components in the system. This "aliasing" of signals can only be removed by increasing the sample frequency, or by pre-filtering the input to the system. Similarly, low frequency components are reflected about the Nyquist point, causing high frequency reflections which must be removed by post-filtering the system output.

1.3.1 Bilinear z-transform

Given the transfer function of a continuous-time filter, we require to map the complex frequency variable, s , to the discrete-time complex variable, z , in order to obtain the transfer function of the sampled data filter. This can be achieved [11], using the bilinear z-transform of equation (1.5), which maps the left-hand side of the s -plane to inside the unit-circle in the z -plane:

$$s \rightarrow (1 - z^{-1}) / (1 + z^{-1}) \quad (1.5)$$

One drawback of this method is that this mapping is non-linear, and critical frequencies must be pre-warped using the following function.

$$\begin{aligned} \omega_C &= \tan(\omega_d T / 2) \\ &= \tan(\pi f_d / f_c) \end{aligned} \quad (1.6)$$

where ω_C is the continuous time frequency, ω_d is the discrete time frequency, and T is the sample period.

1.4 Switched-Capacitor Filters

The realisation of conventional RC-active filters in integrated form has been difficult because stable, precisely valued RC-products cannot be realised in monolithic circuits. As a result, these filters have been realised in the past using discrete components or thick-film hybrid circuits. However, Fried [12] has shown that resistors can be replaced by MOS switches and a capacitor. Consider the circuit of Figure 1-4 (b).

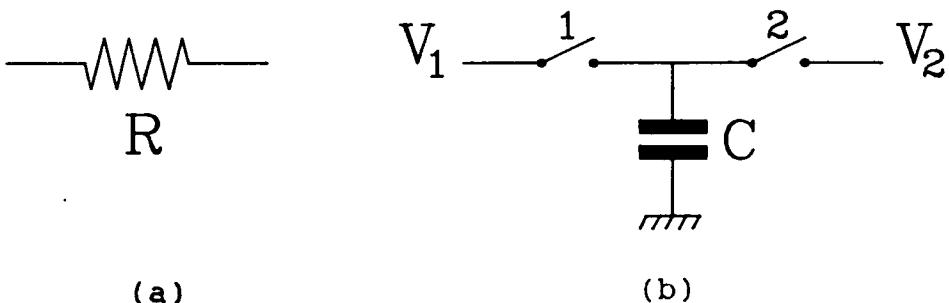


Figure 1-4: Switched capacitor realisation of a resistance

If switch 1 is closed, and switch 2 opened, the capacitor C will be charged to V_1 . When switch 2 is closed, and switch 1 opened, the capacitor is discharged to some voltage V_2 . The amount of charge which flows into (or from) V_2 is $Q = C(V_2 - V_1)$. If the switches are operated at a clock frequency f_C , then the average current flow from V_1 to V_2 will be $C(V_2 - V_1) f_C$. From this, we can see that the discrete resistor value which would give this average current is

$$R = 1 / C f_C \quad (1.7)$$

For this system, the switching time, $T = 1/f_C$, must be much greater than the RC product of the on-resistance of the switch and the holding capacitor C, to allow maximum charge transfer.

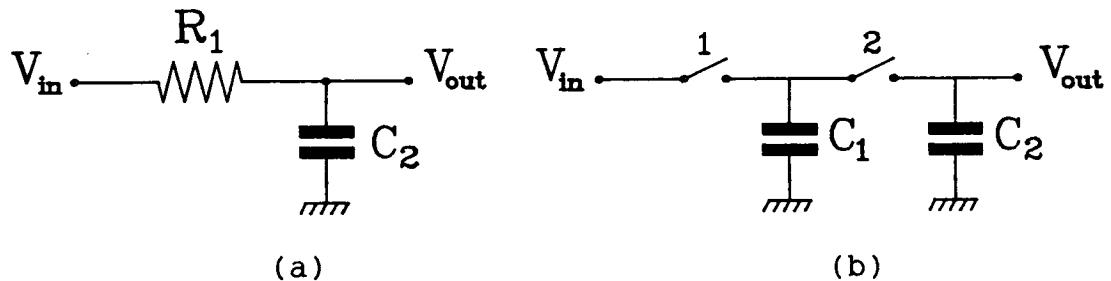


Figure 1-5: (a) Single-pole lowpass RC filter
(b) Switched capacitor equivalent

As an example of this type of filter, consider the single-pole RC filter of Figure 1-5 (a). The cutoff frequency of the RC filter is given by

$$\omega_0 = 1 / R_1 C_2$$

The cutoff of the equivalent switched capacitor filter of Figure 1-5 (b), is therefore,

$$\omega_0 = f_c C_1 / C_2 \quad (1.8)$$

Equation (1.8) shows that the cutoff frequency is proportional to the clock frequency. This property is common to all sampled-data filters. The equation also implies that the cutoff frequency is defined by a ratio of capacitors, rather than a particular value of capacitor. This feature is of particular importance to integrated filters, where it is difficult to maintain specific values of capacitors, but ratios can be achieved within close tolerances (< 1%).

Using MOS technology, these circuits can be combined with integrated operational-amplifiers to realise filters with similar properties to the RLC ladder filters of section 1.2.1.

The original switched capacitor filter structures suffered several drawbacks, such as sensitivity to stray capacitances and the requirement for a high clock rate relative to the cutoff frequency. However, recent developments in filter design techniques have resulted in circuits that are immune to the effects of stray capacitance [13]. Exact design methods have also been developed for switched capacitor filters, allowing maximum operating frequencies of up to half the switching

frequency [14]. The high density circuits available through MOS technology mean that the physical size of switched capacitor filters can be very small, making this type of filter an attractive alternative to the conventional RLC ladder filter. One significant drawback of these filters that has not yet been fully overcome, is that these filters cannot be fully programmed in the way digital filters can. A certain amount of programmability has been incorporated in a design by Allstot *et al* [15], but this is a very limited solution to an important problem.

1.5 Digital Filters

Digital filters are digital signal processing structures which approximate the frequency or time response of continuous time filters. Such a system is shown in Figure 1-6.

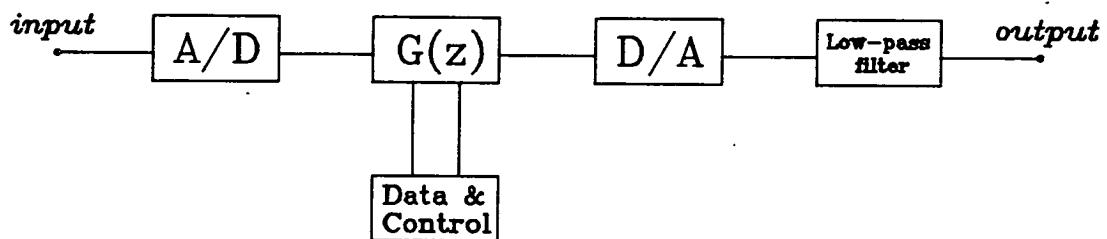


Figure 1-6: Digital filter system

In this system, the discrete time transfer function, $G(z)$, can be implemented in two ways. The algorithm can be implemented in software in digital signal processing

microprocessors, or by using dedicated hardware in discrete or integrated form. The implementation form is determined by the particular application for the filter.

Digital filter systems can be divided into two classes.

1. Non-recursive systems.

2. Recursive systems

Again, the choice of system is dependent on the particular application.

1.5.1 Non-recursive digital filters

Non-recursive filters are systems which have no inherent feedback i.e. the system output, $y(n)T$ is dependent only on the present input $x(n)T$, and previous inputs $x(n-1)T$, $x(n-2)T$..., but with no dependence on previous outputs. For this reason, these filters will have an impulse response which is of finite duration, and are frequently referred to as Finite Impulse Response (FIR) filters. The transversal filter [16] given in Figure 1-7 is an example of this type of system. This figure shows how input samples are successively delayed and multiplied by weighting coefficients (tap weights). The products are summed within one time period to form the output sample. The output $y(t)$ of this filter is therefore given by

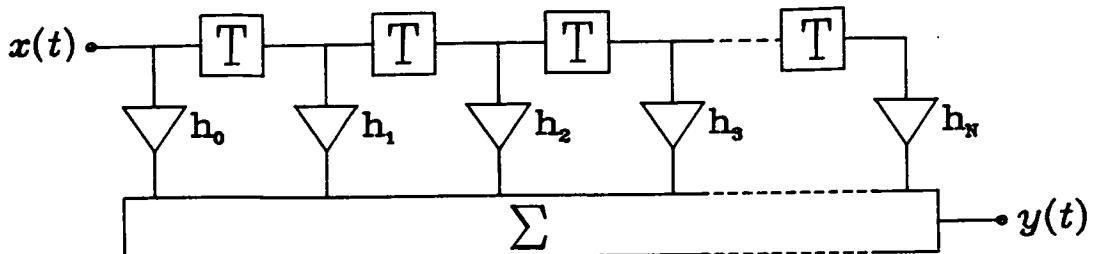


Figure 1-7: Transversal digital filter

$$y(t) = \sum_{m=1}^n h_m x(t-mT)$$

This type of filter has a response containing zeros of transmission but no attenuation poles.

Transversal filters offer several attractive features, not generally available in recursive structures. They are always stable, because of the lack of feedback loops in the system. Both amplitude and phase characteristics can be controlled by the weighting function applied to the taps of the filter, allowing linear phase filters to be designed.

A major disadvantage in the design of non-recursive filters is that they cannot be designed from RLC ladder filters, and therefore the desirable properties of the RLC filters cannot be retained. This type of structure also uses more hardware than recursive digital structures because of the large number of taps, often up to 256 or more, that may be required to satisfy a particular transfer function. For this reason, transversal filters

have been better suited for implementation in analogue form than by using digital techniques.

1.6 Introduction to Wave Filters

Wave filters are a class of sampled-data filters first proposed by Bingham [63], and developed by Fettweis in 1971.[7]. They were originally developed as digital structures which exactly imitated classical ladder filters. However, it was found that conventional voltages and currents could not be used as signal variables, but that incident and reflected voltage (or current) wave variables were well suited to this type of structure (see chapter 2).

Wave filters are exact models of RLC ladder filters. This means that if a reference filter can give a particular frequency response, then allowing for the frequency transformation present in all sampled-data systems, a wave filter can be designed with exactly the same response. This also has the effect that the wave filter will have the same insensitivity to coefficient variation that is available in the reference filter.

Although these filters were originally developed to be digital structures, hence the name wave digital filter, the general theory is independent of any particular implementation. Wave filters have been implemented in both analogue [17], and digital [18] form. For this

reason, throughout the rest of this thesis, wave filters realised using analogue techniques will be referred to as analogue wave filters, and those implemented in digital form as digital wave filters.

1.7 Thesis Objectives

The aim of this work has been to develop digital wave filter structures which are completely general purpose in their design. This allows the implementation of a suitable reference filter which satisfies any desired frequency or phase response. The structures were to be suitable for realisation in silicon integrated circuit form, and be capable of direct cascading to permit the realisation of a filter of any order. Further, the circuits were to require a minimum of external control to achieve their desired response.

Chapter 2 of the thesis presents the general background theory for wave filters. This includes a definition of the frequency variable used in wave filter design and the methods used to guarantee that possible filter structures can be realised as digital wave filters. Wave filter models of the widely used filter elements and possible methods for their interconnection are also presented.

A review of previous implementations of both analog-

gue and digital wave filters is given in chapter three. Wave filters have been realised in integrated circuit form, and as circuits composed from discrete building blocks such as adders and multipliers. Both of these forms are considered, and the advantages and disadvantages of both methods are outlined.

The different techniques used to implement general purpose filter sections as part of this work, are detailed in chapter 4. Examples of possible implementations of different types of wave filters are presented, with details of the suite of computer programmes used to verify the filter structures. This chapter gives details of a general purpose wave filter circuit, capable of modelling a wide variety of continuous time reference filters.

Chapter 5 presents simulation results for an example filter, implemented using the wave filter structure developed in the previous chapter. The filter has been simulated under a variety of conditions, with multiplier coefficient accuracies of between two and twelve bits. Comparisons between the theoretical and practical responses are made, and potential hazards such as limit cycle oscillations and arithmetic overflow are considered.

An integrated circuit has been designed to meet the

requirements of a general purpose wave filter adaptor system. Details of this design are given in chapter 6, considering such aspects as binary multiplication, suppression of bit growth and the layout of the silicon cells corresponding to the requirements of the adaptor system. Practical results of a filter implemented using this integrated circuit are also given.

Chapter 7 presents a discussion of the advantages and disadvantages of digital wave filters. Conclusions are drawn from both the computer simulations and the practical results from the integrated filter, and suggestions are given regarding further areas of work in this field.

Chapter 2

WAVE FILTER THEORY

2.1 Introduction

In order to form a digital model of some passive reference filter, we must first develop a series of general equations relating the flow of signals through the system. The first part of this process is to find some frequency variable which can be used to model the desired reference filter. A system must then be developed with which to represent the values of the signals, at specific times, throughout the filter network. Once this is known, wave filter models of the various filtering elements can be developed. Finally, some means of interconnecting the filter elements is required, which both satisfy the required equations and are physically realisable. Finally, a method must be developed from which all these aspects may be brought together to form a complete filter system.

In this chapter, the necessary theory for the development of wave filters is presented. Wave filter models of widely used filtering elements, are given, and possible methods for their connection to form complete wave filter systems are developed.

2.2 Choice of Frequency Variable

Let the transfer function of an ordinary LC ladder filter be $H(s)$, which is a rational function of the complex frequency s . This corresponds to a single differential equation between the input variable x and the output variable y . However, for the transfer function of a sampled-data filter, the frequency variable s must be replaced by $Z = e^{sT}$, where $F_C = 1/T$ is the clock rate of the filter. This corresponds to a single difference equation between the input and output variables.

In order that a structure can be realised as a digital filter, the following condition must be observed;

In the signal flowgraph of the digital structure, every feedback loop must contain at least one delay element.

Clearly, if this rule was violated in a signal flowgraph, then it would be impossible to find a sequence in which the appropriate calculations could be carried out.

The complex frequency variable used in developing the wave filter theory is ψ [9], defined by

$$\begin{aligned}\psi &= \tanh(sT/2) \\ &= (Z - 1) / (Z + 1) \quad \text{or} \\ &= [1 - e^{-sT}] / [1 + e^{-sT}]\end{aligned}\tag{2.1}$$

for $s = j\omega$, we can therefore write

$$\psi = j\varphi \quad \text{where} \quad \varphi = \tan(\omega T/2) \quad (2.2)$$

This choice of ψ as our frequency variable means that we cannot use voltages or currents as our signal quantities. In order to show this, we shall consider the case of an arbitrary linear capacitance C . The equation to be realised is

$$V = RI / \psi \quad \text{where} \quad R = 1 / C$$

using equation (2.1), this becomes

$$\begin{aligned} V &= RI(1 / \psi) \\ &= RI(Z + 1) / (Z - 1) \end{aligned}$$

giving

$$\begin{aligned} V - VZ^{-1} &= R(IZ^{-1} + I) \\ V[nT] - V[(n-1)T] &= R(I[nT] + I[(n-1)T]) \end{aligned} \quad (2.3)$$

This shows that the computation of V at $t=nT$ requires knowledge of V at $t=nT$, the present sample time, in addition to that of the previous sample. This is clearly a violation of the "delay free" rule given above.

2.3 Incident and Reflected Waves

We can define two new variables, A and B , from linear combinations of the conventional voltage and current variables, V and I . These are defined as follows,

$$A = V + R_0 I \quad (2.4)$$

$$B = V - R_0 I$$

where R_O is an arbitrary constant with the dimension of resistance, and known as the port normalising resistance. These new variables will be known as voltage wave variables as they represent the 'incident' (A) and 'reflected' (B) waves at some point in a circuit. These definitions may also be written in the form of a matrix,

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 1 & R_O \\ 1 & -R_O \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} \quad (2.5)$$

By representing signals as combinations of voltage or current wave variables, rather than voltages or currents, we can develop signal flowgraphs which will contain the required delay elements, and retain the use of ψ as our frequency variable. These signal flowgraphs will also be known as wave flow diagrams.

2.4 Circuit Elements

In this section, we shall determine the wave flow diagrams of typical filtering elements and signal sources.

2.4.1 Inductor

For the inductor shown in Figure 2-1 (a), the steady state voltage-current relation to be realised is

$$\begin{aligned} V &= \psi RI \\ &= RI(Z - 1) / (Z + 1) \end{aligned} \quad (2.6)$$

where R is the inductance in Henries.

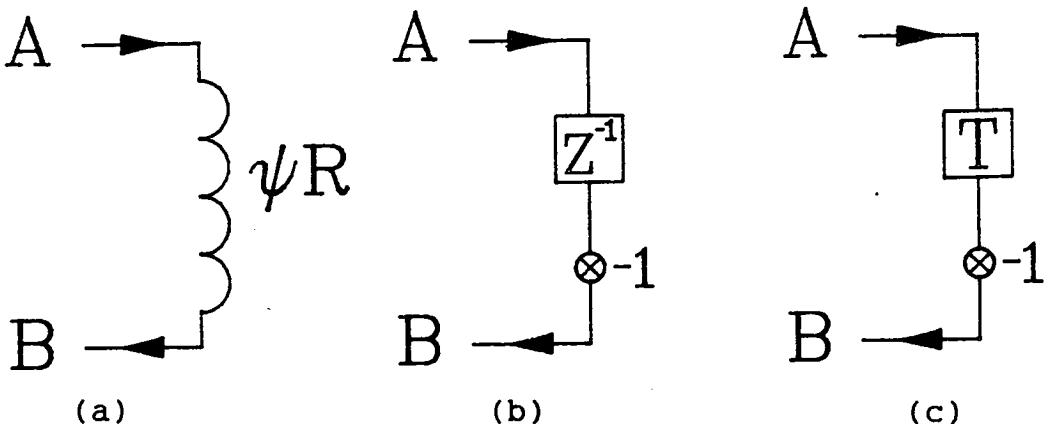


Figure 2-1: Wave filter model of an inductor

$$\text{Now } B = V - RI$$

$$\begin{aligned} &= RI(\psi - 1) \\ &= RI[(Z-1) / (Z+1) - 1] \\ &= RI[-2/(Z+1)] \\ &= -IRZ^{-1}[2Z/(Z+1)] \\ &= -IRZ^{-1}[(Z-1)/(Z+1) + 1] \\ &= -IRZ^{-1}[\psi + 1] \\ &= -Z^{-1}[V + IR] \end{aligned}$$

giving

$$B = -Z^{-1} A \quad (2.7)$$

This states that a voltage (or current) wave entering one terminal of an inductor is delayed and inverted before emerging at the other terminal. The steady-state and instantaneous wave flow diagrams corresponding to equation (2.7) are given in Figures 2-1 (b) and (c)

respectively.

2.4.2 Capacitor

For the capacitor of Figure 2-2 (a), the steady state voltage-current relationship to be realised is,

$$V = RI / \psi \quad (2.8)$$

where R is the capacitance in Farads. Using the same methods as the previous section, we can show that

$$B = Z^{-1} A \quad (2.9)$$

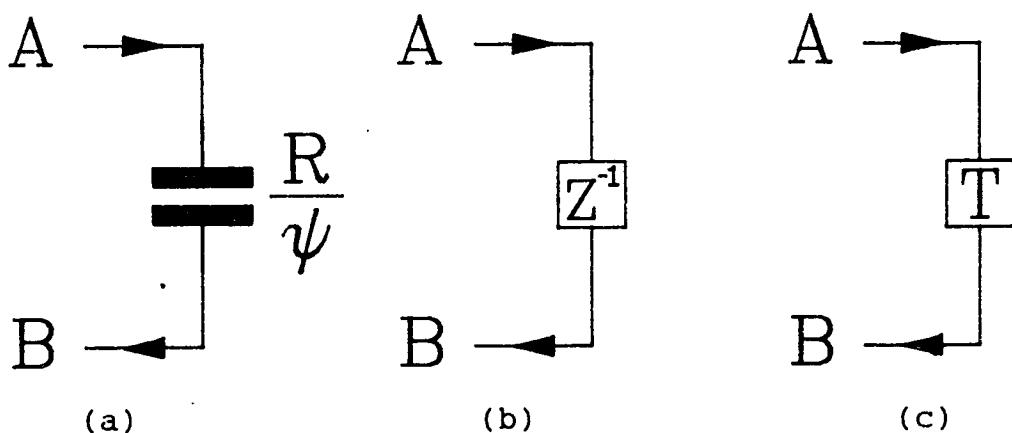


Figure 2-2: Wave filter model of a capacitor

This shows that a wave entering one terminal of a capacitor is delayed before emerging from the other terminal. The steady state and instantaneous wave flow diagrams are given in Figures 2-2 (b) and (c) respectively.

2.4.3 Resistor

For the resistor of Figure 2-3 (a), the steady state voltage-current relationship to be realised is

$$V = IR \text{ or} \quad (2.10)$$

$$V - IR = 0$$

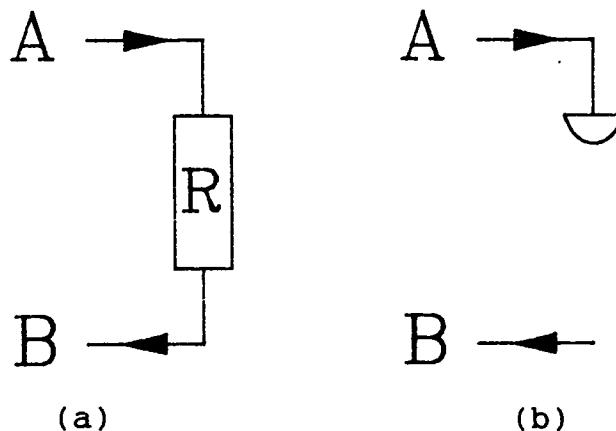


Figure 2-3: Wave filter model of a resistor

Comparison of this result with equation (2.4), gives

$$B = 0 \quad (2.11)$$

This shows that there is no reflected wave from a resistor. Hence, the wave flow diagram of Figure 2-3 (b) consists of a wave sink.

2.4.4 Short circuit

At any short circuit, the voltage present is zero.

$$V = 0$$

If we define R to be some arbitrary positive constant,

equation (2.4) reduces to

$$\begin{aligned} B &= -IR \\ &= -A \end{aligned} \quad (2.12)$$

The wave flow equation of a short circuit consists only of a sign inversion between the incident and reflected waves.

2.4.5 Open circuit

For any open circuit, the current flow, I, is zero. Again, choosing R to be some arbitrary positive constant, we obtain

$$\begin{aligned} B &= V - IR \\ &= V \\ &= V + IR \\ &= A \end{aligned} \quad (2.13)$$

The wave flow equation corresponding to an open circuit is a simple through connection from A to B.

2.4.6 Transmission line or unit-element

A section of transmission line can also be considered as a two-port network. Such networks are often referred to as unit-elements. The characteristic impedance of a unit- element, R_O , is given by

$$R_O = \sqrt{L/C} \quad (2.14)$$

where L and C are the inductance and capacitance per unit

length. Unit elements can be represented as shown in Figure 2-4 (a).

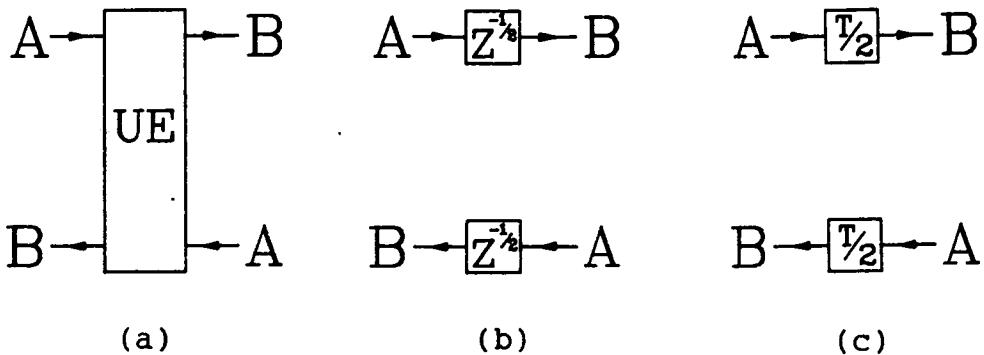


Figure 2-4: Wave filter model of a unit-element

The transfer matrix of a unit-element [3] is given by

$$(1 - \lambda^2)^{\frac{1}{2}} \begin{bmatrix} 1 & \lambda R \\ \lambda/R & 1 \end{bmatrix} \quad (2.15)$$

In this matrix, $\lambda = \tanh(st/2)$ and R is the characteristic impedance of the line. Replacing $st/2$ by X , the transfer matrix becomes

$$\frac{(\cosh X)^{-1}}{(1 - \tanh^2 X)^{\frac{1}{2}}} \begin{bmatrix} \cosh X & R \sinh X \\ \sinh X/R & \cosh X \end{bmatrix} \quad (2.16)$$

Now, $(\cosh X)^{-1} / (1 - \tanh^2 X)^{\frac{1}{2}} = 1$, so the transfer matrix reduces to

$$\begin{bmatrix} \cosh X & R \sinh X \\ \sinh X/R & \cosh X \end{bmatrix} \quad (2.17)$$

Using the transfer matrix definition from section 1.2.1, we find that

$$V_1 = V_2 \cosh X + I_2 R \sinh X$$

$$I_1 R = V_2 \sinh X + I_2 R \cosh X$$

Therefore,

$$\begin{aligned} V_1 + I_1 R &= (V_2 - I_2 R) * (\cosh X + \sinh X) \\ &= (V_2 - I_2 R) \exp(X) \end{aligned}$$

Using equation (2.4), we see that

$$\begin{aligned} A_1 &= B_2 \exp(X) \\ \text{or } B_2 &= A_1 \exp(-X) \\ &= A_1 \exp(-pT/2) \\ &= A_1 Z^{-\frac{1}{2}} \end{aligned} \tag{2.18}$$

Similarly,

$$\begin{aligned} B_1 &= A_2 \exp(-pT/2) \\ &= A_2 Z^{-\frac{1}{2}} \end{aligned} \tag{2.19}$$

The steady state and instantaneous wave flow diagrams for unit-elements are given in Figures 2-4 (b) and (c) respectively. The wave flow diagrams show that voltage (or current) waves entering a unit element are delayed for half a sample period before emerging at the output terminal.

2.4.7 FDNR elements

In wave filters, the circuit element count can sometimes be reduced by employing a pair of impedance transforms, first proposed by Bruton [19], and developed

by Fettweis [64]. These transformations were originally conceived to simplify active filter design, but are also applicable to wave filter design. The transformations give rise to ψ^2 -impedance and ψ^2 -admittance elements known collectively as frequency-dependent negative resistances (FDNRs).

FDNR filters are based on the premise that the voltage transfer function of a passive network is unchanged if the impedance of each element is multiplied by an arbitrary scaling factor. If this scaling factor is chosen to be K/ψ then an inductance will be transformed into a resistance, resistors into capacitors and capacitors to ψ^2 -admittance elements, often referred to as *super-capacitors*.

Similarly, with a scaling factor of ψK , then capacitors are transformed to resistors, resistors to inductors and inductors to ψ^2 -impedance elements known as *super-inductors*.

Of the two FDNR elements introduced above, the super-capacitor element is of the greater importance in active filter design, because it allows the elimination of inductors from reference filters. Similarly, in wave filters, the transformation of any element to a resistor means that the hardware required can be reduced because the wave filter model of a resistor is merely a wave

sink.

2.4.7.1 Super-capacitors

For the ψ^2 -admittance element shown in Figure 2-5 (a), the steady state voltage-current equation to be realised is

$$V = IR / \psi^2 \quad (2.20)$$

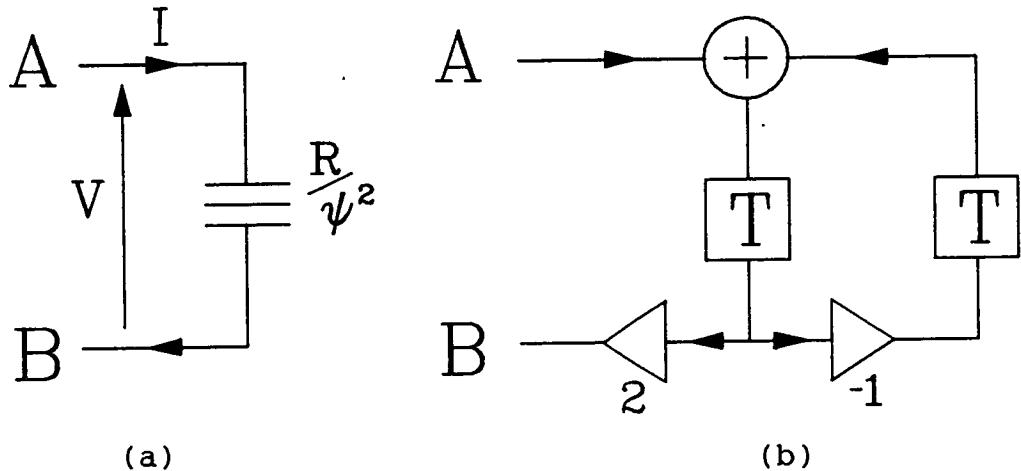


Figure 2-5: Wave filter model of a ψ^2 -admittance element

Again, eliminating V from equation (2.4), we get

$$\begin{aligned} A &= IR(1 + 1/\psi^2) \\ B &= IR(1/\psi^2 - 1) \\ A(1/\psi^2 - 1) &= B(1 + 1/\psi^2) \\ B &= A[1 - (Z-1)^2 / (Z+1)^2] / [1 + (Z-1)^2 / (Z+1)^2] \\ &= 2AZ/(Z^2 + 1) \end{aligned} \quad (2.21)$$

This corresponds to the difference equation

$$B[nT] = 2A[(n-1)T] + B[(n-2)T] \quad (2.22)$$

The steady state wave flow diagram corresponding to equation (2.22) is given in Figure 2-5 (b).

2.4.7.2 Super-inductors

If we consider the ψ^2 -impedance element as shown in Figure 2-6 (a), the steady state voltage-current relationship to be realised is

$$V = \psi^2 IR \quad (2.23)$$

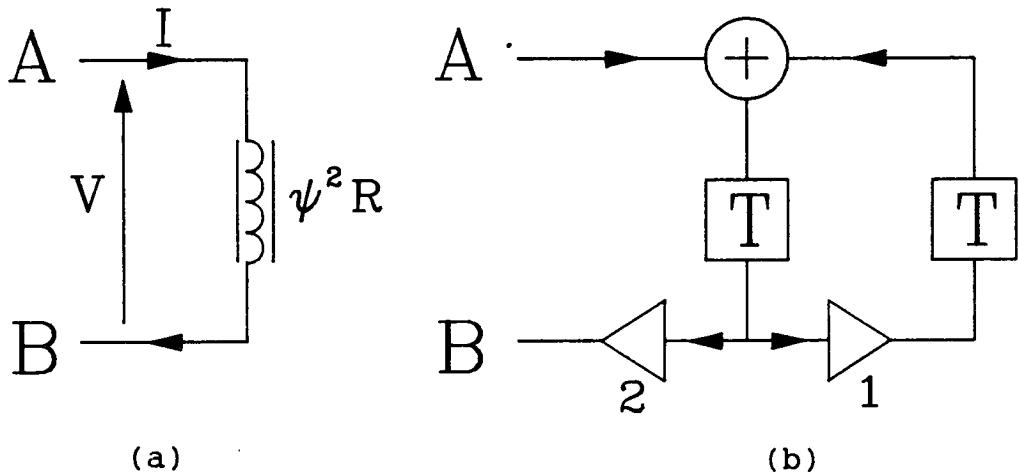


Figure 2-6: Wave filter model of a ψ^2 -impedance element

Using the same methods as before, we find

$$\begin{aligned} A &= IR(\psi^2 + 1) \\ B &= IR(\psi^2 - 1) \\ &= A(\psi^2 - 1) / (\psi^2 + 1) \\ &= -2AZ/(Z^2 + 1) \end{aligned} \quad (2.24)$$

This corresponds to the difference equation

$$B[nT] = -2A[(n-1)T] + B[(n-2)T] \quad (2.25)$$

The wave flow diagrams corresponding to equation (2.25) is given in Figures 2-6 (b). An example of wave filters implemented using FDNR elements is given in section 3.3.3.

2.5 Interconnection of Circuit Elements

Generally, networks are constructed by interconnecting the ports of circuit elements and signal sources. In the previous sections, we have defined the port resistance, R , to be some arbitrary positive constant. However, within any network, the port resistance will be fixed by the element or signal source to which the port is connected. Hence, we must develop some method of interconnecting these circuit elements by changing the port resistances. This can be achieved by using *adaptors* [7].

Within a network, adaptors may be required to interconnect two ports, in which case they are called *two-port adaptors*, or they may interconnect several ports in which case they are called *multi-port adaptors*. Circuit elements may be connected in series or in parallel in a network, and the resulting network equations must be satisfied by different multi-port adaptors. Hence, within any filtering network, there are three types of

adaptor that may be used to interconnect circuit elements.

1. Two-port adaptors.
2. Multi-port *series* adaptors.
3. Multi-port *parallel* adaptors.

2.5.1 Two-port adaptors

A two-port adaptor is used to connect two ports of different port resistances, R_1 and R_2 . Consider the two-port connection of Figure 2-7 (a).

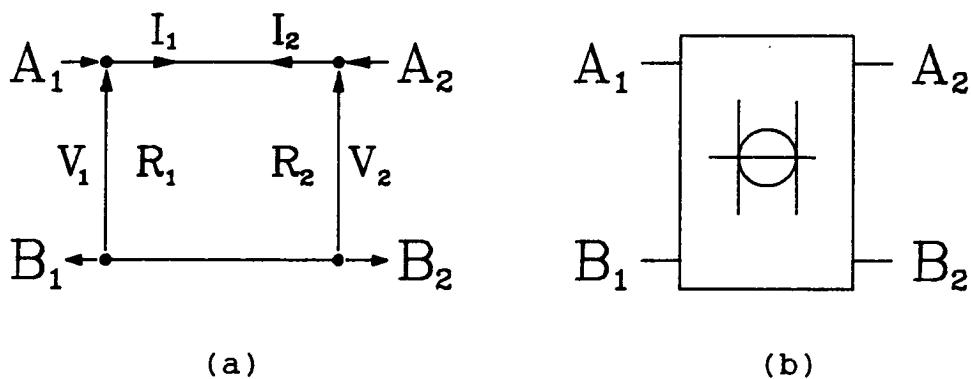


Figure 2-7: Two-port adaptor

In this figure, port 1, with port resistance R_1 , is connected to port 2, of port resistance R_2 . From Kirchhoff's Laws, V_1 must equal V_2 , and the currents I_1 and I_2 must be equal and opposite.

$$V_1 = V_2 = V \quad \text{and} \quad I_1 = -I_2 = I \quad (2.26)$$

The wave signal quantities are related to the voltages and currents using equation (2.4) as follows,

$$A_k = V_k + I_k R_k \quad (2.27)$$

$$B_k = V_k - I_k R_k \quad \text{for } k=1,2$$

Combining equations (2.26) and (2.27) gives,

$$A_1 = V + IR_1 \quad A_2 = V - IR_2 \quad (2.28)$$

$$B_1 = V - IR_1 \quad B_2 = V + IR_2$$

Eliminating V And I from equation (2.28) gives,

$$B_1 = A_2 + \alpha(A_2 - A_1) \quad (2.29)$$

$$B_2 = A_1 + \alpha(A_2 - A_1)$$

$$\text{where } \alpha = (R_1 - R_2)/(R_1 + R_2) \quad (2.30)$$

A possible realisation of a two-port adaptor corresponding to equation (2.29) is given in Figure 2-8.

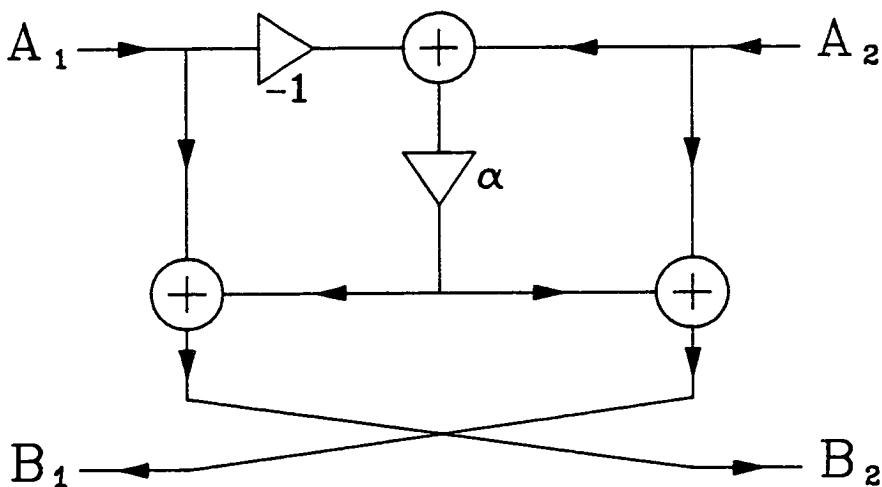


Figure 2-8: Signal flowgraph of two-port adaptor

Examination of this signal flowgraph shows a signal path from A_1 to B_1 , and from A_2 to B_2 , but with no delay-free feedback loops. The signal flowgraph is therefore

realisable.

Equation (2.30) states that if R_1 and R_2 are both positive, then $|\alpha| < 1$. This condition can simplify hardware requirements for realisations of two-port adaptors.

The symbolic representation of a two-port adaptor is given in Figure 2-7 (b). Clearly, for a two-port adaptor, there is no distinction between parallel and series connections.

2.5.2 Multi-port parallel adaptors

Consider n ports, with port resistances R_1, R_2, \dots, R_n respectively. If these ports are connected in parallel, then by Kirchhoff's laws,

$$V_1 = V_2 = \dots = V_n \quad \text{and} \quad (2.31)$$

$$I_1 + I_2 + \dots + I_n = 0$$

Eliminating V and I between equations (2.4) and (2.31) we find,

$$B_k = A_O - \frac{A_k}{N} \quad (2.32)$$

$$\text{where } A_O = \sum_{m=1}^n \alpha_m A_m \quad (2.33)$$

$$\text{and } \alpha_k = 2G_k / (G_1 + G_2 + \dots + G_n) \quad (2.34)$$

$$\text{with } G_k = 1/R_k$$

These equations define an n -port parallel adaptor, shown

symbolically in Figure 2-9 (a).

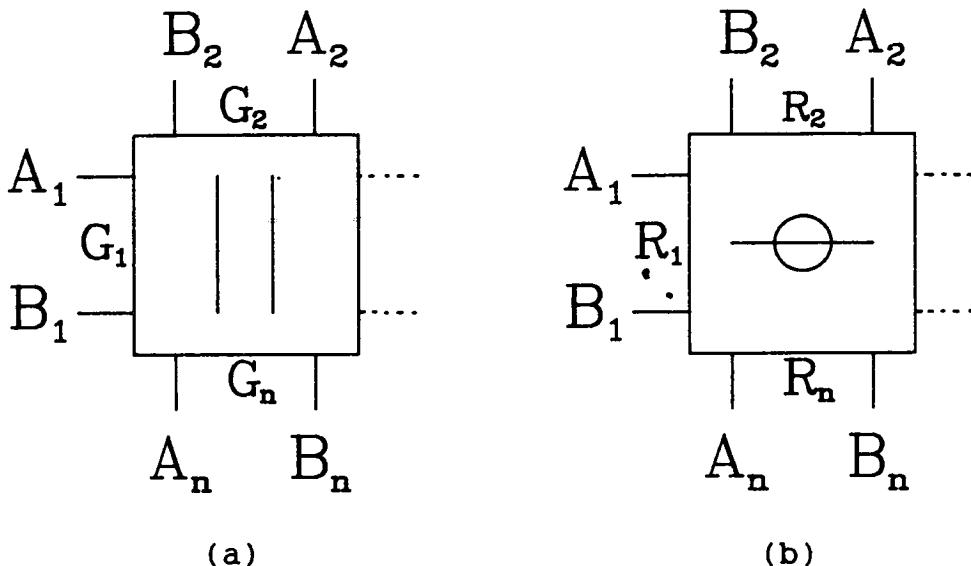


Figure 2-9: Multi-port parallel and series adaptors

2.5.3 Multi-port series adaptors

Again, consider n ports $1, 2, \dots, n$, but this time connected in series. For this network, we have

$$V_1 + V_2 + \dots + V_n = 0 \quad \text{and} \quad (2.35)$$

$$I_1 = I_2 = \dots = I_n$$

Again, eliminating V and I , we find

$$B_k = A_k - \frac{\beta_k A_O}{N} \quad (2.36)$$

$$\text{where } A_O = \sum_{m=1}^N A_m \quad (2.37)$$

$$\text{and } \beta_k = 2R_k / (R_1 + R_2 + \dots + R_n) \quad (2.38)$$

These equations define an n -port series adaptor, shown symbolically in Figure 2-9 (b).

2.6 Cascaded Two-Port Adaptor Filters

Using the theory of microwave transmission line filters, it is always possible to build filters that are a cascade of unit-elements of equal length, but of different characteristic impedance [20]. This type of filter is periodic in its frequency response, with period $2F_0$, where F_0 is the frequency at which the length of the transmission line, L , is one quarter wavelength [21].

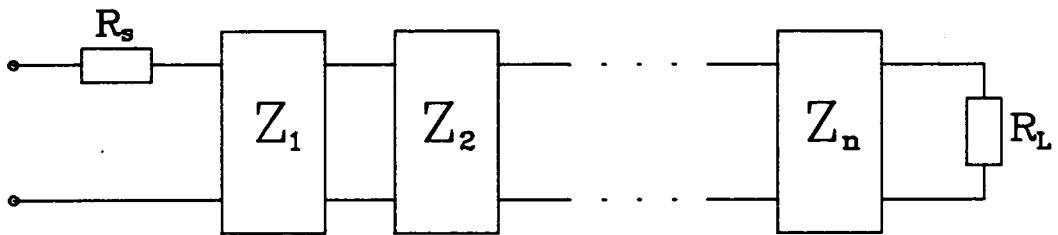


Figure 2-10: Cascaded unit-element filter

Figure 2-10 shows an example of a cascade of n unit-elements with characteristic impedance Z_1, Z_2, \dots, Z_n . The electrical length of these unit-elements is equal, and the structure is terminated with a source resistance of R_S and a load resistance R_L . This type of structure can be considered as a distributed form of the lumped RLC ladder filter. Cascaded unit-element filters can realise maximally flat or Butterworth responses of the form [21]:

$$|H(\omega)|^2 = 1/(1 + (\sin \omega / \sin \omega_0)^{2n}) \quad (2.39)$$

where ω is the frequency variable, and n is the order of

the filter. The 3dB cutoff point is at $\omega = \omega_o$ as described in chapter 1.

Similarly, these structures can produce Chebyshev responses of the form

$$|H(\omega)|^2 = 1/(1 + \epsilon^2 T_n^2(\sin \omega / \sin \omega_o)) \quad (2.40)$$

where $T_n(x)$ is the Chebyshev polynomial of degree n , and ϵ is related to the passband ripple, S , by

$$S = 10 \log_{10}(1 + \epsilon^2)$$

Again, $\omega = \omega_o$ is the cutoff point of the filter.

The variables ω and ω_o must lie in the region between 0 and $\pi/2$, because the filter response is periodic about $\omega = \pi/2$.

The maximum attenuation of these filters is at $\omega = \pi/2$. For a Butterworth response, the maximum attenuation, X_b , is given by

$$X_b = 10 \log_{10} [1 + (1 / \sin \omega_o)^{2n}] \quad (2.41)$$

and for a Chebyshev response,

$$X_C = 10 \log_{10} [1 + \epsilon^2 T_n^2 (1 / \sin \omega_o)] \quad (2.42)$$

Clearly, the attenuation is not infinite at $\omega = \pi/2$.

To implement these cascaded unit-element filters as wave filters, the reference structure must be transformed into a suitable form. In section 2.4.6, the transformation of unit-elements into wave filter models showed that

unit-elements can be represented by delays of $T/2$, where T is the sample period. This is summarised in Figure 2-4. However, the characteristic impedances of the unit-elements in a filter are not equal, so a two-port adaptor will be required to allow the unit-elements to be interconnected. For a reference structure of n unit-elements, with load and source resistances R_S and R_L , $n + 1$ two-port adaptors will be required.

As an example of the transformation from a cascaded unit-element filter to a wave filter, consider the network of Figure 2-11 (a). This is a third-order filter with three unit-elements of characteristic impedance Z_1 , Z_2 and Z_3 . The wave filter of Figure 2-11 (b) shows the result of the transformation, with four two-port adaptors and three groups of time delays corresponding to the unit-elements.

To simplify the nomenclature of these wave filter diagrams, the following nomenclature will be used throughout the rest of this thesis.

- Terminals labelled $A_{i,j}$ are those accepting an incident wave on the i^{th} port of adaptor j .
- Terminals labelled $B_{i,j}$ are those which output a reflected wave from port i of adaptor j .

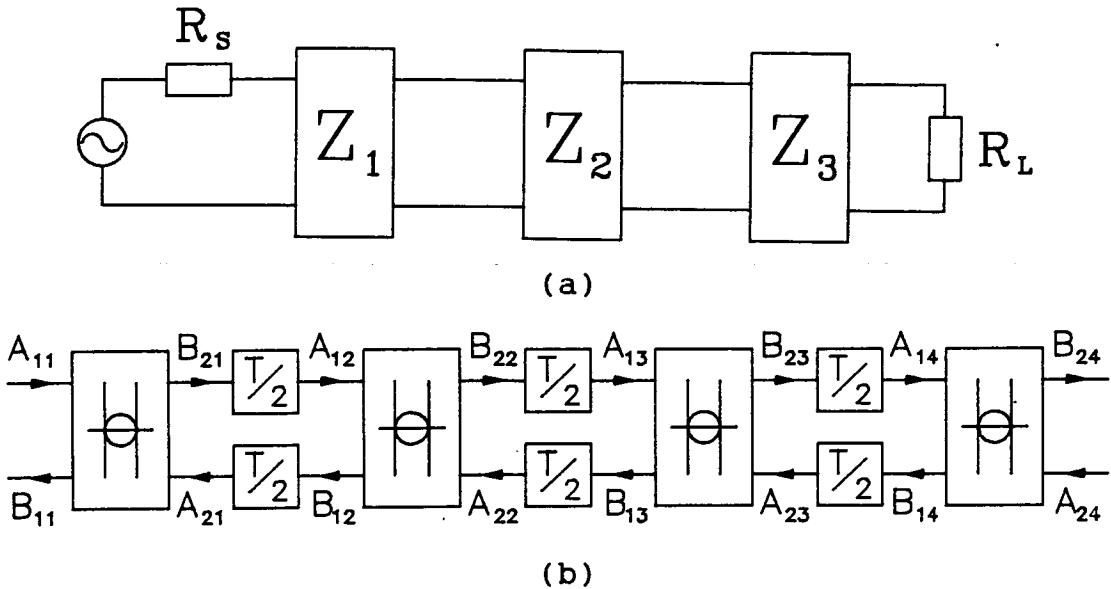


Figure 2-11: (a) Cascaded unit element filter and
(b) Wave filter equivalent

If these descriptions are applied to Figure 2-11 (b), then the filter input is at terminal A_{11} , and the output at B_{24} . The filter input at A_{24} will normally be grounded as there is no wave reflected from a pure resistance. These descriptions can clearly be extended to include adaptor nodes for multi-port connections.

2.7 Wave Filter Implementations of Ladder Structures

Using multi-port adaptors with only three ports, it is possible to model RLC ladder filters with low-pass, high-pass or band-pass responses [22]. This type of filter can realise Butterworth responses of the form:

$$|H(\omega)|^2 = 1 / [1 + (\tan \omega / \tan \omega_0)^{2n}] \quad (2.43)$$

and Chebyshev responses of the form:

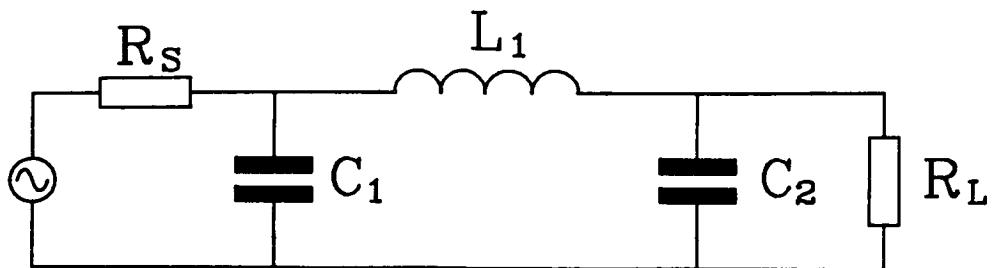
$$|H(\omega)|^2 = 1 / [1 + \epsilon^2 T_n^2(\tan \omega / \tan \omega_0)] \quad (2.44)$$

For this type of filter, the transfer function is a function of $\tan \omega$, rather than $\sin \omega$ as is the case for the cascaded unit-element filter. This function gives a faster roll-off than the sine function and gives a theoretically infinite attenuation at $\omega = \pi/2$.

2.7.1 Wave filter design using Kuroda transforms

In principle, RLC ladder filters may be realised by direct interconnection of three-port adaptors which model the parallel or series connection of inductors or capacitors. However, the filter theory developed in sections 2.4 and 2.5 shows that each adaptor must carry out its internal calculations instantaneously. This is clearly impossible in practice, and suitable time delays are required between each adaptor. The addition of these time delays is achieved by inserting unit-elements into the reference structure at suitable places. However, these unit-elements must be inserted in some way which does not affect the desired response of the filter. This can be achieved by using Kuroda Transforms [4].

Figure 2-12 shows a third order low-pass filter with a Chebyshev response. The passband ripple for this



$$R_s = R_L = 1\Omega \quad C_1 = C_2 = 3.077 \text{ F}$$

$$L_1 = 6.154 \text{ H}$$

Figure 2-12: Third order low-pass reference filter

filter is 0.177dB. The cutoff frequency for this filter is to be at 10% of the sample clock rate. Note that this filter will be used as an example throughout the remainder of this chapter.

A number of unit-elements of characteristic impedance Z_0 may be inserted between a lumped element network and a termination of impedance Z_0 , without affecting the amplitude response [20]. The phase response of the network will however be changed. Kuroda transforms allow circuit elements to have their order interchanged without affecting their circuit function. By continuing the process, the desired filter structure can be obtained.

The process of inserting unit-elements into lumped networks is shown in detail for the example filter in Figure 2-13. Figure 2-13 (d) can then be transformed

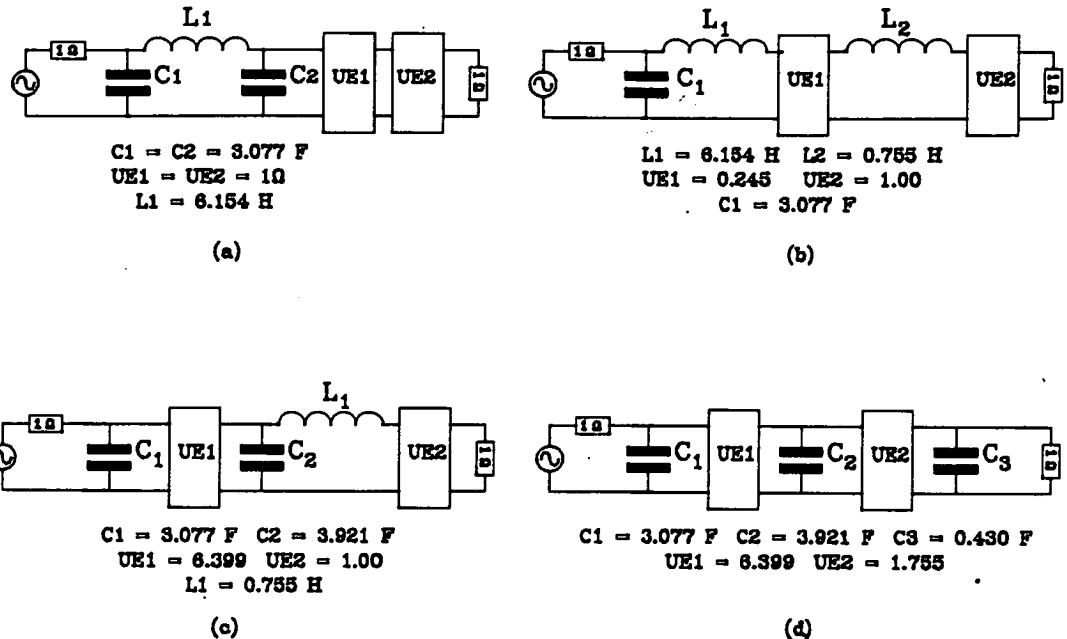


Figure 2-13: Insertion of unit-elements using Kuroda transforms

directly into the wave filter of Figure 2-14. This wave filter structure can be implemented using a two-phase clocking scheme, with each adaptor operating independently. This circuit structure can be described as a half-synchronous system.

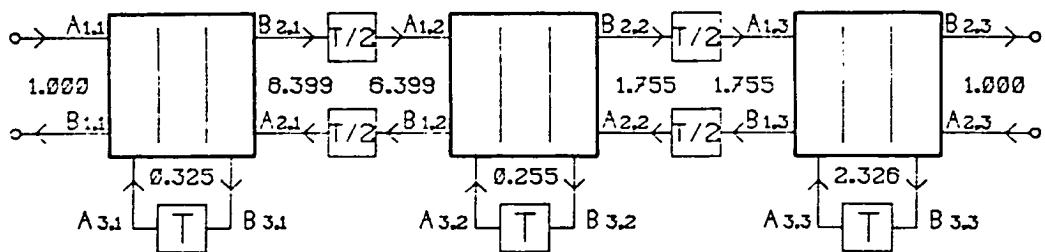


Figure 2-14: Wave filter implemented using Kuroda transforms

2.7.2 Wave filter design using synthesis techniques

In the filter networks of the previous section, the unit-elements inserted into the wave filter are redundant, in that they are present only to simplify system timing, and do not add anything to the order of the filter. However, from section 2.6, we know that filters can be designed using only cascaded unit-elements, so it is reasonable to assume that an optimised transfer function can be derived.

A suitable transfer function for a band-pass filter which does not contain redundant unit-elements is given below [23]

$$|S_{2,1}(\Omega^2)|^2 = 1 / [1 + \epsilon^2 f^2(X, Y, Z)] \quad (2.45)$$

with

$$\begin{aligned} f(X, Y, Z) &= \cosh[N \cosh^{-1}(X) + K \cosh^{-1}(Y) \\ &\quad + L \cosh^{-1}(Z)] \end{aligned} \quad (2.46)$$

$$X^2 = [(1 + \Omega_2^2)(\Omega_1^2 - \Omega^2)] / [(\Omega_1^2 - \Omega_2^2)(1 + \Omega^2)] \quad (2.47)$$

$$Y^2 = [\Omega_2^2(\Omega_1^2 - \Omega^2)] / [\Omega^2(\Omega_1^2 - \Omega_2^2)] \quad (2.48)$$

$$Z^2 = (\Omega_1^2 - \Omega^2) / (\Omega_1^2 - \Omega_2^2) \quad (2.49)$$

$$\Omega_1 = \tan(\varphi_1) \quad \Omega_2 = \tan(\varphi_2) \quad (2.50)$$

where N is the number of unit-elements, K is the number of zeros of transmission at d.c. and L is the number of zeros of transmission at the Nyquist frequency. φ_1 and φ_2 are the lower and upper cutoff frequencies of the

filter. Low-pass and high-pass filters can be obtained by setting $K=0$ and $\phi_1 = 0^\circ$ for a low- pass response, and $L=0$ and $\phi_2 = 90^\circ$ for a high-pass response.

No closed form solution to this equation is known, so the filter synthesis is performed using high precision arithmetic by a computer program called SYNTH [23]. Apart from the high precision arithmetic, the synthesis is quite conventional. However, although the program can guarantee to synthesise a transfer function to the required specification, no such guarantee can be given regarding the topology of the resulting filter network. This synthesis procedure is explained in more detail in Appendix I.

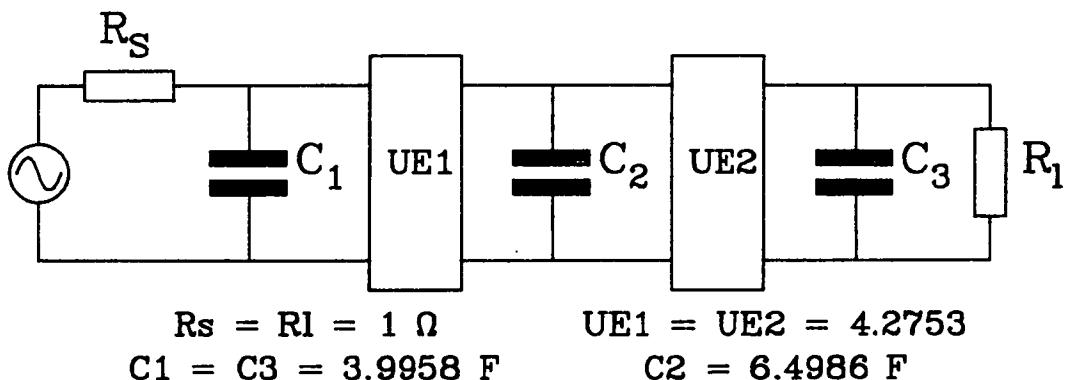


Figure 2-15: Third order reference filter designed using synthesis techniques

As an example of wave filter design using synthesis techniques, consider the reference filter of Figure 2-13

(d). This is a third order, low-pass filter with two unit-elements, three transmission zeros at Nyquist, no transmission zeros at d.c., and with an upper cutoff frequency of 10% of the sample rate, F_S . If this filter is generated using the synthesis techniques detailed above, then the unit-elements can be used to add to the order of the filter. The resulting reference structure of Figure 2-15 shows that although the element values have been changed, the overall structure remains the same. This method allows an increase in the filter order corresponding to the number of unit-elements employed, but with no corresponding increase in hardware requirements.

2.7.3 Wave filter design using matched-port techniques

The adaptor theory of section 2.5 showed how parallel and series three-port adaptor equations can be developed. However, these equations require that three multiplications be carried out within each adaptor. For any implementation of wave filters, it is desirable to reduce as far as possible the number of multipliers used in the circuit. In a wave filter, this can be achieved using the *matched-port* technique.

Consider a situation in which two port resistances of a three-port parallel adaptor are fixed, but the other can be set arbitrarily to any value. The equations to be realised are:

$$B_K = A_O - A_K$$

$$\text{where } A_O = \alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3$$

$$\text{and } \alpha_m = 2G_m / (G_1 + G_2 + G_3)$$

If we choose

$$G_2 = G_1 + G_3$$

$$\text{then } \alpha_2 = 2G_2 / (G_1 + G_2 + G_3)$$

$$= 2(G_1 + G_3) / (G_1 + G_1 + G_3 + G_3)$$

$$= 1$$

From equation (2.34), we find

$$\alpha_1 + \alpha_2 + \alpha_3 = 2$$

Since $\alpha_2 = 1$, we have

$$\alpha_3 = 1 - \alpha_1$$

and the three multiplier coefficients will have the following values

$$\alpha_1 = \alpha_1$$

$$\alpha_2 = 1 \quad (2.51)$$

$$\alpha_3 = 1 - \alpha_1$$

and the adaptor can be built with only one multiplier, α_1 , rather than two as before.

Similarly, this method can be applied to series three-port adaptors. Again we can show that

$$\beta_1 = \beta_1$$

$$\beta_2 = 1 \quad (2.52)$$

$$\beta_3 = 1 - \beta_1$$

and that the adaptor can be built with only one multiplier, β_1 .

If equation (2.51) is substituted into equation (2.32), we find

$$B_1 = (1 - \alpha_1)(A_3 - A_1) + A_2 \quad (2.53)$$

$$B_2 = \alpha_1(A_1 - A_3) + A_3 \quad (2.54)$$

$$B_3 = \alpha_1(A_1 - A_3) + A_2 \quad (2.55)$$

Equation (2.54) for port 2 of the parallel adaptor shows that B_2 , the reflected wave from port 2, is independent of A_2 , the incident wave. This allows adaptors to be connected directly to each other, without the need for time delays between each adaptor. A similar property can be shown for series three-port adaptors.

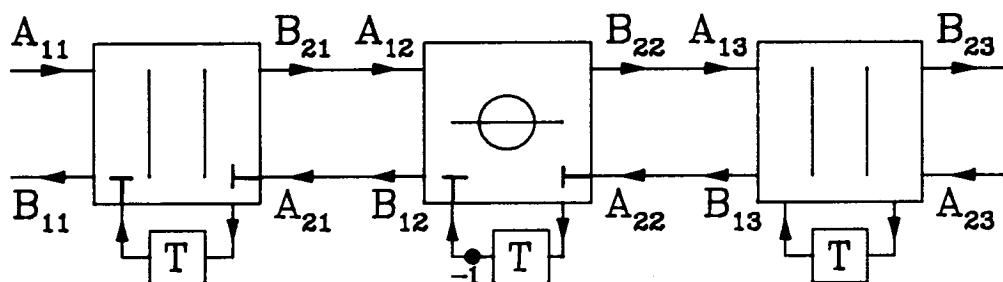


Figure 2-16: Example wave filter implemented using the matched port technique

Figure 2-16 shows the result of implementing a wave filter using the matched-port technique. The adaptor symbols for parallel and series three-port adaptors have

been modified to indicate that the matched-port system has been used. It should be noted that the wave filter requires a conventional three-port adaptor to connect the load to the filter network. This is because all the port resistances are defined for this node, and the matched-port technique can therefore not be used.

A major disadvantage of this system is that adaptors cannot be separated in the same way as in the half-synchronous system. The values of the adaptor outputs must be calculated in a well defined order, depending on the complexity of the filter. Therefore, all internal calculations must be carried out within one sample period, T . This method of system timing is often referred to as a full-synchronous system.

2.7.4 Wave lattice filters

Lattice filters are also known to possess the low passband sensitivity of conventional RLC ladder filters, although they are very sensitive to component variation in the stopband region. Digital wave lattice filters [24] have also been shown to retain these properties. As a brief example of the wave filter implementation of lattice filters, consider the lattice reference filter of Figure 2-17 (a), and the corresponding wave flow diagram (b). The transfer function of the structure, S_{21} , is given by



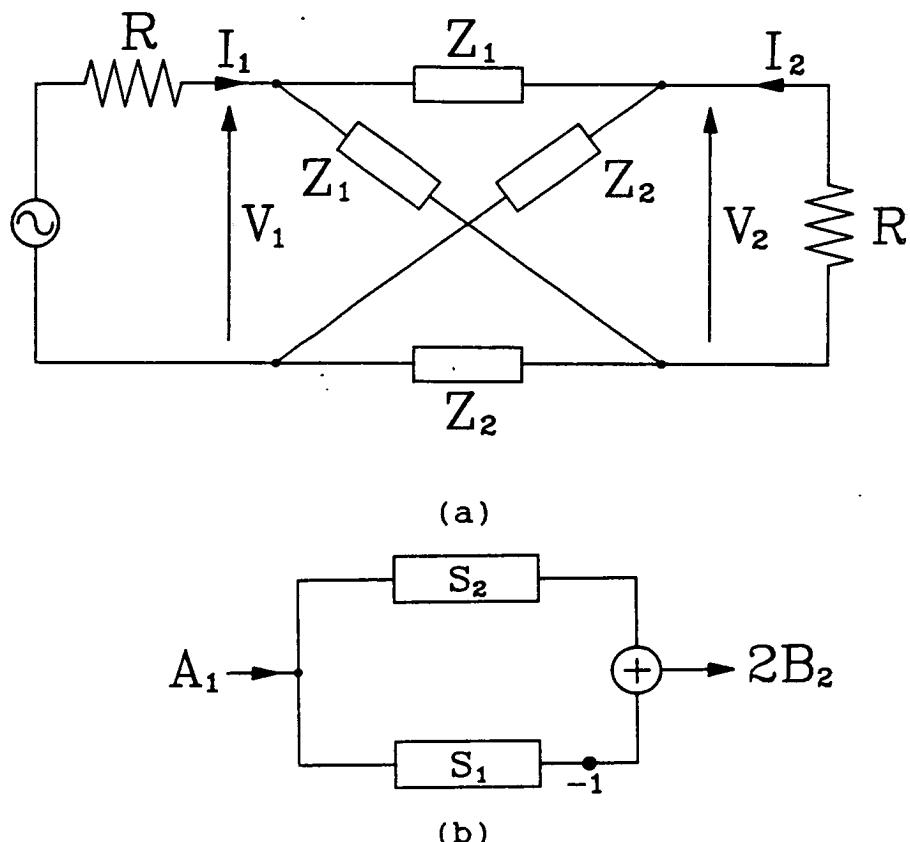


Figure 2-17: (a) Lattice reference filter
 (b) Digital wave lattice filter

$$S_{21} = 2V_1 / V_2 \quad (2.56)$$

$$= B_2 / A_1 \quad (2.57)$$

$$= (S_2 - S_1) / 2 \quad (2.58)$$

where S_1 and S_2 are the reflectances corresponding to the impedances Z_1 and Z_2 .

$$S_1 = (Z_1 - R) / (Z_1 + R) \quad (2.59)$$

$$S_2 = (Z_2 - R) / (Z_2 + R) \quad (2.60)$$

From equation (2.58),

$$2B_2 = [S_2 - S_1]A_1 \quad (2.61)$$

as shown in Figure 2-17 (b). Equations (2.59) and (2.60) can be realised by the interconnection of suitable unit-elements using two-port adaptors, resulting in a wave filter structure such as that shown in Figure 2-18.

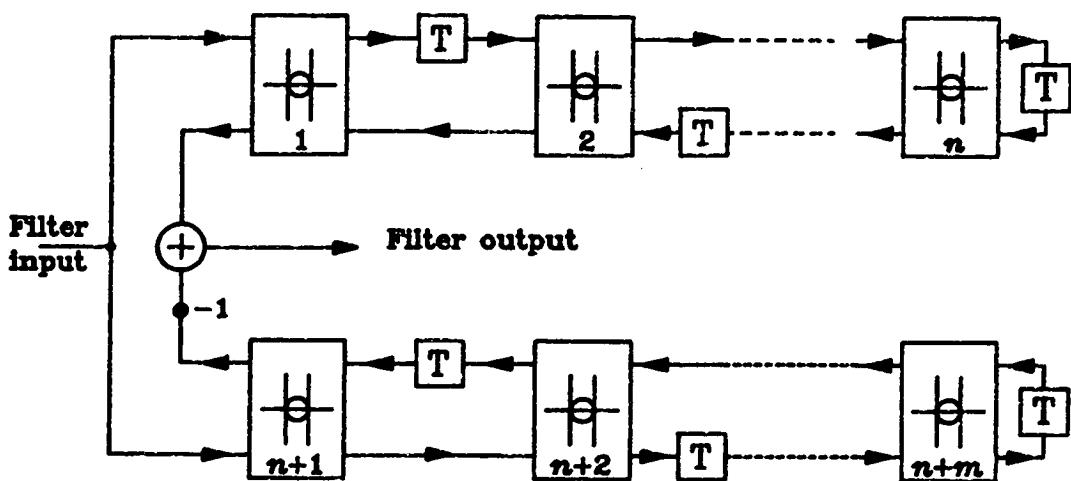


Figure 2-18: *m*-th order digital wave lattice filter structure

2.8 Summary

In this chapter, wave filter models have been developed for each of the elements required for the implementation of a variety of reference structures. Details of the possible implementation of these circuits have been given, and a synthesis procedure has been presented which can optimise the filter response without increasing the hardware requirements in any way.

The wave filter equivalents for the widely used

filtering elements, inductors, capacitors and unit-elements, can be realised using simple components such as delays and sign-inverters. More complex filtering elements such as the FDNR elements of section 2.4.7 can also be modelled, although the circuits require more hardware than the wave equivalents of inductors and capacitors. Frequency independent elements, such as resistors and open- and short-circuits can be modelled without increasing the hardware in any way. Wave filter circuits for other filter elements such as circulators, gyrators, transformers and resonant circuits can be found in Reference [7].

The implementation of wave filters is dependent on the reference filter required to achieve the desired response. The connections between the elements of the reference filter can be modelled by using wave filter adaptors. These adaptors are used to connect two or more ports with different port reference resistances. Equations have been presented for the simple two-port case, and for the more general n -port parallel and series adaptors. These adaptors may be used to model any continuous time reference filter. Examples of the realisation of wave filters using these circuits will be given in the following chapter.

Finally, the filter theory for a variety of continu-

ous time filters has been developed. The different classes of reference filters include cascaded unit-element filters, RLC ladder filters and wave lattice filters. Synthesis techniques have been presented for optimising the response of RLC ladder filters containing unit-elements to simplify the timing of the wave filter equivalent.

Chapter 3

WAVE FILTER IMPLEMENTATION

3.1 Introduction

In any signal processing system, the method chosen to implement the required algorithms is strongly dependent on the particular application for the system. If the system requires a high sample rate, then the algorithms may be implemented using dedicated hardware for each section. Alternatively, if the system is to operate at relatively low speeds, then digital signal processing (DSP) microprocessor systems may be used. Finally, if the system is required to have low power consumption, or be physically small, then the system may be designed to fit on one or more custom or semi-custom integrated circuits.

The wave filter theory, developed in Chapter 2, is independent of the particular method chosen to realise the filter structures. Although wave filters were originally conceived to be implemented using digital methods, these filters can be realised using digital or analogue means. Analogue wave filters can be implemented as custom integrated circuits, or they can be built using integrated switches, operational amplifiers, and discrete capacitors. In contrast, digital wave filters can be

realised using a wider variety of methods. Digital wave filters can also be realised as integrated circuits and by using discrete hardware, such as adders and multipliers. In addition, they can be realised using software algorithms in minicomputers or DSP microprocessors.

In this Chapter, the various methods used to realise wave filters will be discussed. Initially, analogue wave filters will be considered, and different techniques will be presented for both the design and implementation of filter structures. Secondly, a variety of digital wave filter systems will be given. Using digital circuits, a wider variety of reference structures have been realised as wave filters, and examples are given of each case.

3.2 Analogue Wave Filters

Analogue implementations of wave filters have used a variety of methods to realise adaptor structures. These methods are all based on switched-capacitor techniques whereby charge is stored on capacitors under the control of switches operating on two or more clock phases. In these circuits, operational amplifiers are used to provide integrating structures and to maintain voltage levels within the circuit.

The earliest reported work in the analogue implementation of wave filters was carried out by Reekie *et al*

[25,26]. In this work, analogue methods were used to realise cascaded unit-element filters of the form detailed in section 2.6. As an example of this method, consider the two port adaptor of Figure 3-1.

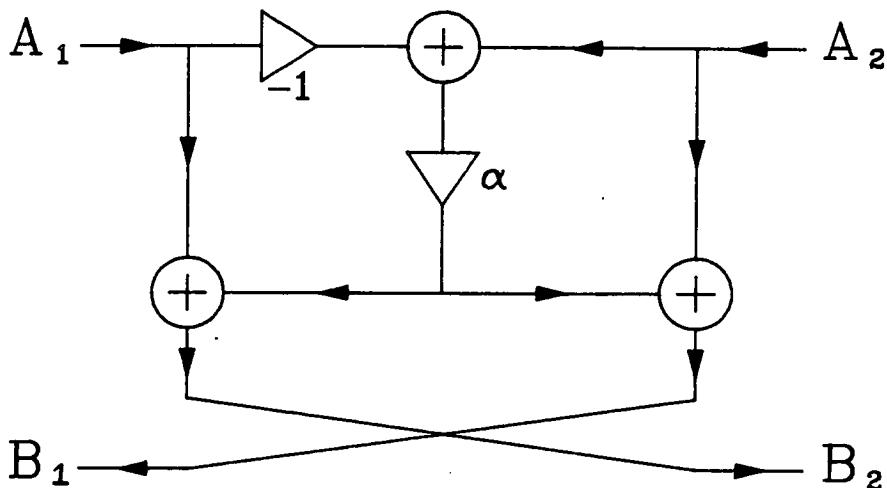


Figure 3-1: Two port adaptor wave flow diagram

The adaptor equations to be realised are

$$B_1 = A_2 + \alpha(A_2 - A_1)$$

$$B_2 = A_1 + \alpha(A_2 - A_1)$$

$$\alpha = (R_1 - R_2) / (R_1 + R_2)$$

In the implementation of these equations, the incident and reflected wave variables A_1 , A_2 , B_1 and B_2 , can be represented by voltages stored on capacitors. However, to ensure correct operation of the filter, the adaptor equations must be evaluated in time $T/2$.

In order to evaluate the reflected wave variables, methods had to be developed for the addition, subtraction

and multiplication of signal values. To add or subtract wave variable values, the following techniques may be used. This method uses only switches and capacitors, but unity-gain buffers are often required to maintain voltage levels and to provide sufficiently large currents to drive the following stages.

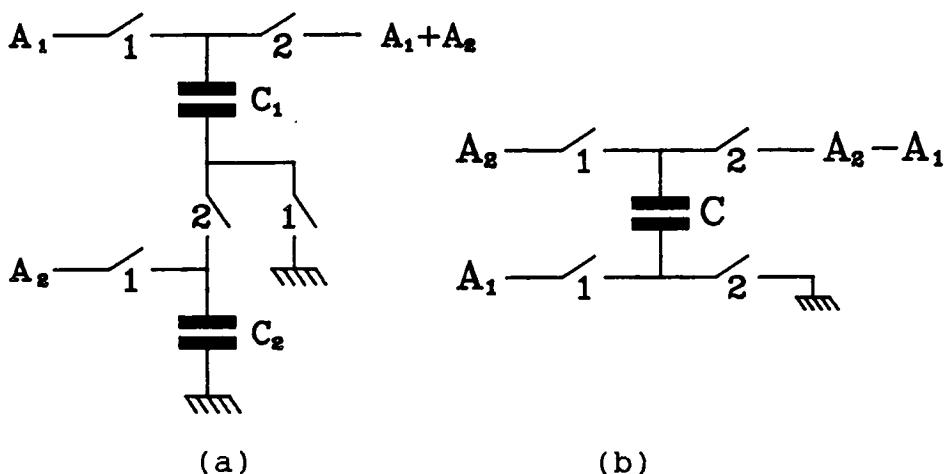


Figure 3-2: (a) Addition and (b) Subtraction of wave variables using analogue techniques

Figure 3-2 (a) shows how, during clock phase 1, two capacitors, C_1 and C_2 , are charged to voltages that represent the wave variables, A_1 and A_2 . During phase 2 of the clock, the capacitors are connected in series and the voltage sum $A_1 + A_2$ is available at the output. Similarly, voltages can be subtracted using the circuit of Figure 3-2 (b). During phase 1 of the clock, the voltages to be subtracted are applied across the capacitor C. The bottom plate of the capacitor is grounded, and the result, $A_2 - A_1$ made available during clock phase 2.

In two port adaptor systems, the voltage reflection coefficient, α , must lie within the range $0 < \alpha < 1$. Multiplication of a signal variable by α can therefore be achieved using the circuit of Figure 3-3. Again, this circuit is controlled by a two-phase clocking scheme.

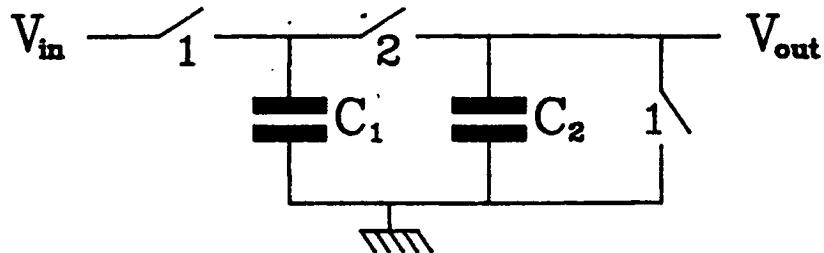


Figure 3-3: Multiplication of a wave variable by coefficient α

During clock phase 1, capacitor C₁ is charged to the input voltage V_{in} , and capacitor C₂ is discharged. The total charge in the system must therefore be

$$Q = C_1 V_{in}$$

On clock phase 2, the charge, Q, is shared between both capacitors, giving

$$Q = (C_1 + C_2)V_{out} \quad (3.1)$$

As the total charge in the system cannot change,

$$Q = C_1 V_{in} = (C_1 + C_2)V_{out} \quad (3.2)$$

$$V_{out} / V_{in} = C_1 / (C_1 + C_2) \quad (3.3)$$

If C₂ is chosen such that C₂ = ξC_1 , then

$$V_{out} / V_{in} = C_1 / C_1(1 + \xi) \quad (3.4)$$

$$= 1 / (1 + \xi)$$

which is dependent only on the ratio of C_1 to C_2 , rather than on an actual capacitor value. Thus, the coefficient, ξ , must be chosen such that

$$\alpha = 1 / (1 + \xi) \quad (3.5)$$

By combining each of the above circuits, the circuit diagram for an analogue realisation of a two port adaptor can be designed. In the circuit of Figure 3-4, unity gain buffers have been included at the B_1 and B_2 outputs to

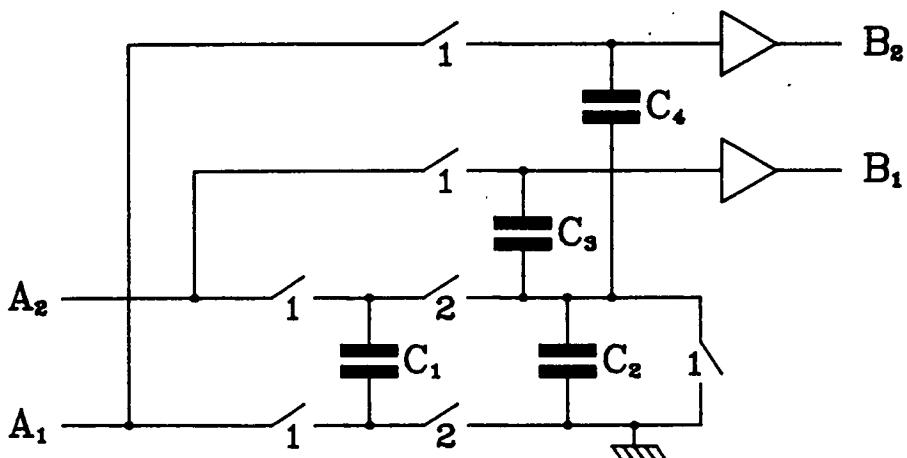


Figure 3-4: Analogue realisation of a two-port adaptor

maintain voltage levels and to provide suitable currents to drive the following stages of the filter.

These circuits can be cascaded to form filter structures. A cascade of $(N+1)$ adaptors will provide a N th order filter response. Because these wave filters are

half-synchronous systems, a four-phase clocking scheme is required, with odd-numbered adaptors operating on clock phases 1 and 2, and even-numbered adaptors working on phases 3 and 4.

An integrated circuit realisation of analogue wave filters using the techniques detailed above has been reported in [27]. The circuit contained three independent lowpass wave filters of third, fifth and seventh order, based on cascaded unit-element reference filters. Each reference filter was designed to have a cutoff frequency of 12.5% of the sample clock rate. The passband ripple of each filter was to 0.177dB. The circuit was designed to have a maximum clocking frequency of 60kHz and operate on a four-phase clocking system. The integrated circuit measured 5 mm by 6mm and had a power consumption of 135mW. Practical results of these filters are given in [17] and show close agreement with the desired response.

The circuit techniques developed for the implementation of two-port adaptors may also be applied to the design of three-port structures. However, in section 2.5, it was shown that two possible circuit arrangements are required to meet the three-port adaptor needs; parallel adaptors and series adaptors. In addition, the adaptor design must allow for the termination present at port

3 of the structure, which will normally be inductive or capacitive. Therefore, a minimum of four circuits are required for the implementation of three port analogue wave filter adaptors. Examples of such circuits have been presented in [17].

A major disadvantage of the circuits detailed above, is that parasitic capacitances, present in the realisation of the analogue wave filter, can affect the response of the filter. In filter realisations using discrete components, stray capacitances are predominantly caused by the capacitive effect of the interconnection between the components of the filter. However, the magnitude of this stray capacitance is small in comparison to the size of the capacitors used in the adaptor structure, and may be considered negligible. In contrast, the stray capacitance present in integrated wave filters is generated from both the interconnection between components and from the top and bottom plate of the integrated capacitors. These parasitic capacitances can be as large as 20% of the designed value [28], and must be considered in the design of the filter. In conventional switched-capacitor filters, several circuits have been presented which are immune from these effects, and are often referred to as "stray-free" circuits [13].

To overcome this problem, James, [29], has proposed

adaptor circuits which are insensitive to the effects of parasitic capacitances in MOS circuits. These circuits have been simulated using SWITCAP [30], a switched-capacitor filter simulation program, and results presented in [29]. In addition, a practical implementation of the filter has been developed using discrete capacitors and integrated switches. Practical results obtained from these circuits are again presented in [29].

Analogue implementations of wave filters were the earliest reported examples of integrated wave filters. At the time, analogue techniques were best suited to integrated wave filters for several reasons. The most important advantage was the physical size of the structures involved. Operational amplifiers, capacitors and switches were ideal candidates for integration, in comparison to the areas required for adders, subtractors and, more particularly, multipliers. Other advantages of analogue wave filters include greatly simplified system control and the need for fewer additional components over digital circuits of similar functional complexity. The original benefit of structures well suited to integration has been developed to include circuits immune to the effects of stray capacitance, [29], and this has also resulted in a reduction in the hardware requirement for filters implemented using three-port adaptors.

Although both design and fabrication techniques have improved since analogue wave filters were first realised as integrated circuits, the advantages of these methods have diminished. The main reason for this is that as the minimum dimensions of MOS transistors are reduced, the presence of random noise begins to affect the filter response. As a result, the minimum transistor dimension used in analogue MOS circuits will reach a practical limit at around $3\mu\text{m}$ [31]. In contrast, minimum dimensions in digital MOS circuits are predicted to reach a limit at around $0.2\mu\text{m}$ [31], offering much higher packing densities in digital circuits. For these reasons, it seems unlikely that wave filters will be produced commercially using the techniques discussed above.

3.3 Digital Wave Filters

The methods used to implement digital wave filters are more varied than those used for their analogue equivalents. Generally there are three approaches which may be considered.

1. The filter may be implemented using general purpose hardware such as integrated multiplier chips or DSP microprocessors. This approach offers rapid system development, and allows the filter to be included as part of a larger signal processing system with no major increase in hardware requirements.

2. A filter may be custom designed on silicon to fit certain required conditions. Integrated filters offer small physical size, low power consumption and generally higher system clocking rates than other approaches.
3. The filter structure may be simulated by a computer program. This technique is widely used in the system development stage. A wide range of programs are available commercially, or have been reported, which offer the filter designer the opportunity to ensure the correct filter response under a variety of input conditions, or to minimize the hardware required to achieve the desired response.

A further subdivision of methods arises when it is noted that digital circuits can be designed using either bit- serial or word-parallel arithmetic. The latter offers faster circuit operation, but the complexity, physical size and power consumption of the resulting circuits weigh against it.

The continuous-time reference structures on which the digital wave filters are based also differ widely, and cover the following classes.

- Cascaded unit-element filters.
- Lattice filters.
- RLC ladder filters.
- FDNR ladder filters.

In this section, examples of the implementation of each of these classes of filter will be given.

3.3.1 DWF implementation using general purpose hardware

Digital wave filters can be implemented by using general purpose building blocks such as integrated multipliers, adders and random access memory devices (RAM) or by using DSP microprocessors. This method has been used to realise wave filters by Lawson [32,33,34] and by Renfors [35].

The wave filters implemented by Lawson were again based on cascaded unit-element reference filters. Because of the regular nature of this type of filter structure, the general purpose hardware method is particularly well suited to the realisation of such filters. Figure 3-5 shows the digital structure corresponding to a cascaded unit-element reference filter. In this structure, the two half-unit delays between each adaptor have been combined. This operation does not affect the magnitude response, but will add a linear shift to the overall phase response [21].

Figure 3-6 shows the basic section that is multi-

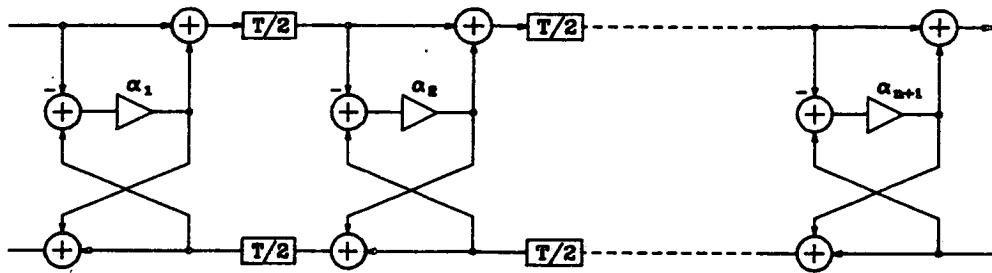


Figure 3-5: Digital wave filter based on unit-element cascade

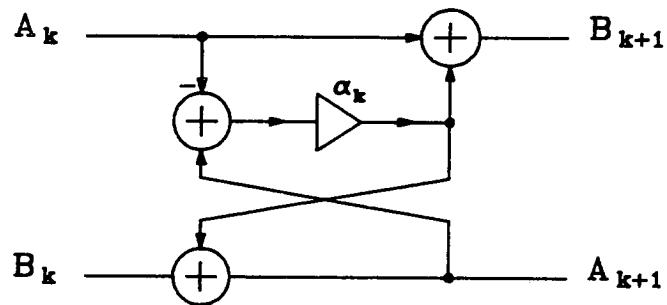


Figure 3-6: Implementation using one basic filter section

plexed. As before, the difference equations to be realised are

$$B_k = A_{k+1} + \alpha (A_{k+1} - A_k)$$

$$B_{k+1} = A_k + \alpha (A_{k+1} - A_k)$$

These equations are evaluated for each stage of the filter from $k=1, \dots, N$ where N is the number of adaptor sections to be multiplexed. Thus, to implement such a structure, there will be a requirement for a hardware multiplier, an adder/subtractor, memory to hold intermediate signal values, and some means of controlling the

system to ensure correct circuit timing and to provide multiplier coefficients.

The hardware structure used by Lawson to implement the wave filters is shown in Figure 3-7. The internal arithmetic of this system is 12-bit 2's complement, although the A/D and D/A converters only use 8-bit accuracy. A 12-bit parallel LSI multiplier was used in the system. The use of a multiplier/accumulator chip was considered, although it was found to be unsuitable for efficient realisation of the required equations [33]. Throughout the structure, 12-bit stores have been used to store internal signal values, and an 8-bit shift register to store the outputs from other adaptor stages. System control and coefficient storage were both achieved by using EPROMs.

The sample rate of this filter is determined by the number of system clock cycles required to perform all the required arithmetic operations. In addition, several "book-keeping" actions must be carried out to maintain the correct operation of the filter. For the system of Figure 3-7, a total of 9 system clock cycles are required for the computation of each filter section and the appropriate maintenance operations.

In a later design, the shift register sections of the above structure were replaced by RAM and the system

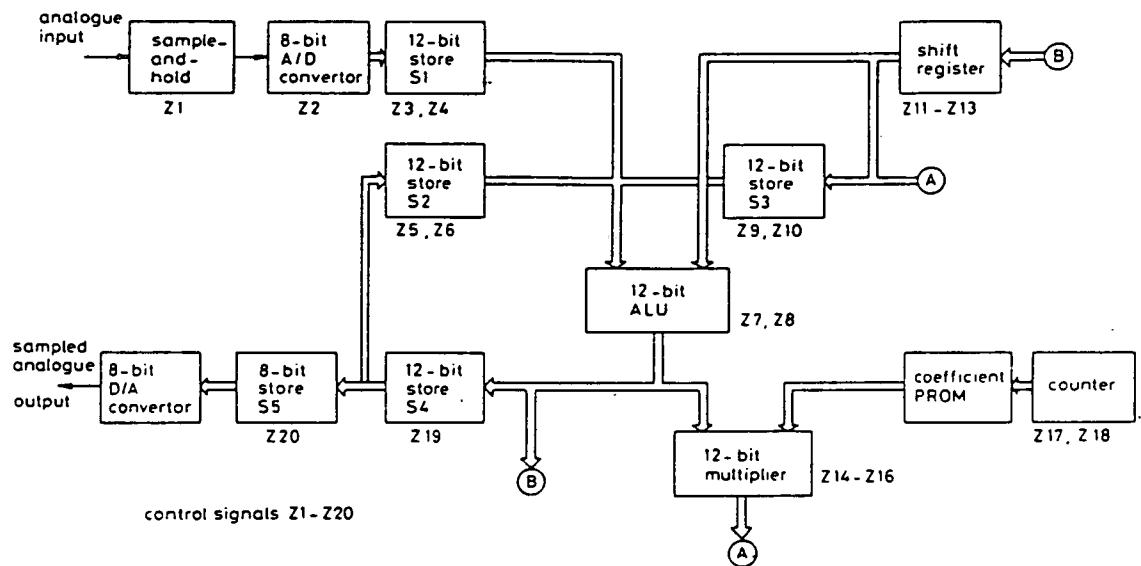


Figure 3-7: Filter hardware structure

control provided by a microprocessor development system [34]. To increase the data rate within the filter, extra adder circuits were included to provide a parallel processing path, and the action of reading from, and writing to memory were carried out simultaneously with arithmetic operations. These improvements in the design of the filter enabled the number of clock cycles required for each filter section to be reduced from 9 to 4. The sample rate of the wave filter is dependent on the order of the filter, N , and the system clock rate, F_C . Therefore, for the structures presented in [33] and [34], the filter sample rate will be

$$F_S = F_C / C(N + 1)$$

where C is the number of clock cycles required to evaluate each adaptor section.

Practical results of filters implemented using these methods are given in [33] and [34]. The filter example chosen in [33] was a 6th order Chebyshev lowpass filter with a passband ripple of 0.5dB and minimum stopband attenuation of 45dB. Passband and stopband edges were at 12.5% and 25% of the sampling frequency respectively. In the implementation of this filter, a system clock of 1MHz was used. This allowed a maximum sample rate of 14.9 kHz. The measured attenuation characteristic of the filter showed good agreement with the theoretical curve.

The example filters implemented in [34] were as follows:

- A 9th order Chebyshev lowpass filter with a passband ripple of 1dB and a minimum stopband attenuation of 43dB. Passband and stopband edges were to be at 25% and 35% of F_S respectively.
- A 5th order Chebyshev lowpass filter with 0.5 dB passband ripple. For this filter, the passband and stopband edges were at 10% and 30% of F_S respectively. The minimum stopband attenuation was to be 55 dB.

Practical results for the 9th order structure also considered the truncation of the multiplier coefficient, α_k , from 12 bits to 8, 6 and 4 bits. With the exception of the 4 bit coefficient, the filter showed good agreement between the theoretical and practical responses.

When the multiplier coefficient was reduced to 4 bits, there was a significant deviation in both the passband edge and the passband ripple from the predicted values. In the case of the 5th order filter, quantisation of the coefficient to 4 bits again produced an unacceptable frequency response. As before, responses with multiplier coefficients of 12, 8 and 6 bits showed good agreement with the expected results.

In [35], Renfors and Zigouris have used a signal processing microprocessor to implement a class of wave lattice filters as described in section 2.7.4. The wave lattice filter realised in [35] is based on a class of lattice reference filters where both lattice branches are realised as a cascade of first and second order allpass sections [36].

The microprocessor used in this implementation was the Signal Processor Interface (SPI) chip μ PD 7220 produced by NEC. The processor contains a 16×16 bit 2's complement multiplier capable of producing a 31 bit result in 2 system clock cycles. The SPI has on-chip PROM and RAM for both program and data storage. System development for the SPI can be simplified by a hardware emulation package which allows the designer to optimize the system performance by examination of the program and data memories and by tracing the program execution. The

SPI also offers facilities for detecting arithmetic overflow within the processor. In the implementation of the wave filters, an 8-bit parallel bus was used for controlling the system and updating the multiplier coefficients. The input and output of digital signals was achieved through the use of a serial I/O line on the processor.

A significant feature of implementing wave filters using a DSP microprocessor is that the number of multiplications to be carried out is not the most critical factor as in other types of implementation. Using this technique, a multiply operation requires the same number of system clock cycles as an addition, allowing the conventional two-port adaptor wave flow diagrams such as Figure 3-1 to be replaced by structures which are more efficient in their use of hardware. Examples of these alternative signal flow-graphs are presented in [35].

The wave filter realised using this method, was a tenth order bandpass filter based on a cascade of fifth order highpass and lowpass elliptic filters. The frequency and attenuation characteristics of the filter were as follows.

| | |
|----------------------|--------------|
| Sample rate | 32kHz |
| Passband edges | 200Hz, 10kHz |
| Stopband edges | 125Hz, 12kHz |
| Passband ripple | 0.25dB |
| Stopband attenuation | 47dB |

The DSP program required to execute the filter algorithm also detected and corrected arithmetic overflow in the system. This required 120 instruction cycles, offering a maximum sample rate of 33.33kHz. In the wave filter, both the data and the coefficient wordlengths were 16 bits, offering a maximum dynamic range of 96dB. However, the arithmetic techniques used in the filter did not suppress zero input quantization limit cycles, and the measured limit cycle amplitude corresponded to a 7 bit signal. This reduced the dynamic range of the filter to 54dB, which satisfied the desired specification of the filter.

Implementations of digital wave filters using these techniques offer several advantages. A major benefit of these systems is that the frequency response of the filter is programmable and can be altered by simply changing the control program and the values of the multiplier coefficients. Due to the regular structure of the wave filter, these types of implementation offer rapid system development. In addition, this approach allows the wave filter to be incorporated in a more complex DSP system where circuits may be used for several applications. Finally, this method may also be used to implement reference structures designed using the matched port technique described in section 2.7.3. These advantages mean that a hardware implementation such as that

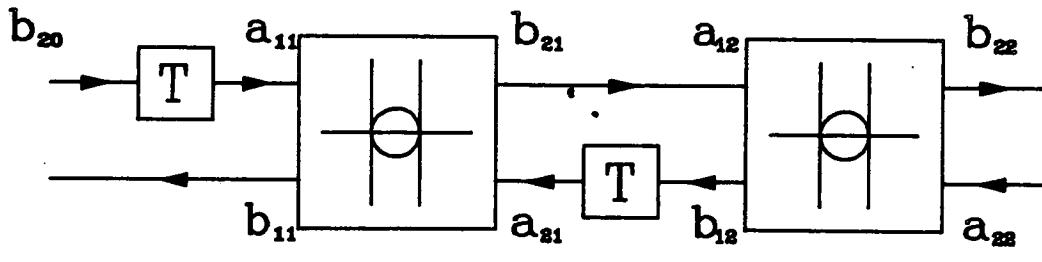
presented in [34] and [35] offer a versatile system for digital filtering. The main drawback of this type of implementation is the limitation on the sample rate, F_S , caused by the fact that all adaptor sections must be evaluated during one sample period, T. Clearly, this will have the effect of reducing the sample rate as the order of the filter increases, which may be a limiting factor where a high order filter is required to sample at high frequencies.

3.3.2 Integrated digital wave filters.

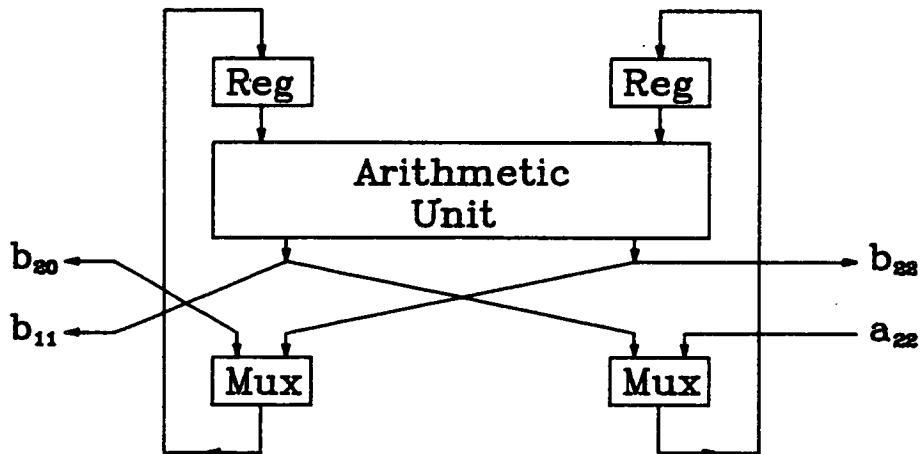
In situations where low power consumption and small physical size are required, digital wave filters may be custom designed as integrated circuits. Several custom realisations of wave filters have been reported, using a variety of reference structures. Wave filters, based on cascaded unit-elements filters have been reported, [37], using distributed arithmetic techniques. Wave lattice filters have also been realised in integrated circuit form [38], [39]. In each of these cases, the filter response was fixed at the design stage, and was not programmable.

In [37], Sikstrom reports the design of a digital wave filter based on a fifth order cascaded unit-element reference filter. In the digital system, all arithmetic was performed using bit-serial techniques to minimize the

hardware requirements. Because of the regular nature of the reference filter, the wave flow diagram of an Nth order filter can be easily partitioned into $(N+1)/2$ identical subsections, as shown in Figure 3-8 (a).



(a)



(b)

Figure 3-8: (a) Basic substructure of filter section
(b) Block diagram of computational unit

As before, the two $T/2$ delays have been combined in one branch of the wave flow diagram. By combining the adaptors as shown in the diagram, the same hardware may be

used for each two-port adaptor in the section, without the need for registers to hold intermediate values. The computational unit used to achieve this is shown in Figure 3-8 (b). For the fifth order example filter, three of these units are required, corresponding to six conventional two-port adaptors. This technique for reducing the hardware requirements is explained in more detail in Chapter 4.

The adaptor equations for these computational units were evaluated using distributed arithmetic techniques, based on the computation of inner products (sum of products). This method uses a shift-accumulator and a ROM look-up table, such as that shown in Figure 3-9. The shift-accumulator is realised using a chain of carry-save adders and generates an inner product in $(W_C + W_d)$ cycles, where W_C and W_d are the coefficient and data wordlengths respectively. A full precision product is generated, although the computation of two inner products may be overlapped in time to efficiently use the available hardware.

The realisation of one computational unit requires two shift-accumulators, two ROMs and two 2-input multiplexers. The registers shown in Figure 3-8 (b) are incorporated in the shift-accumulator, and can therefore be ignored. Both ROMs are controlled by the same address

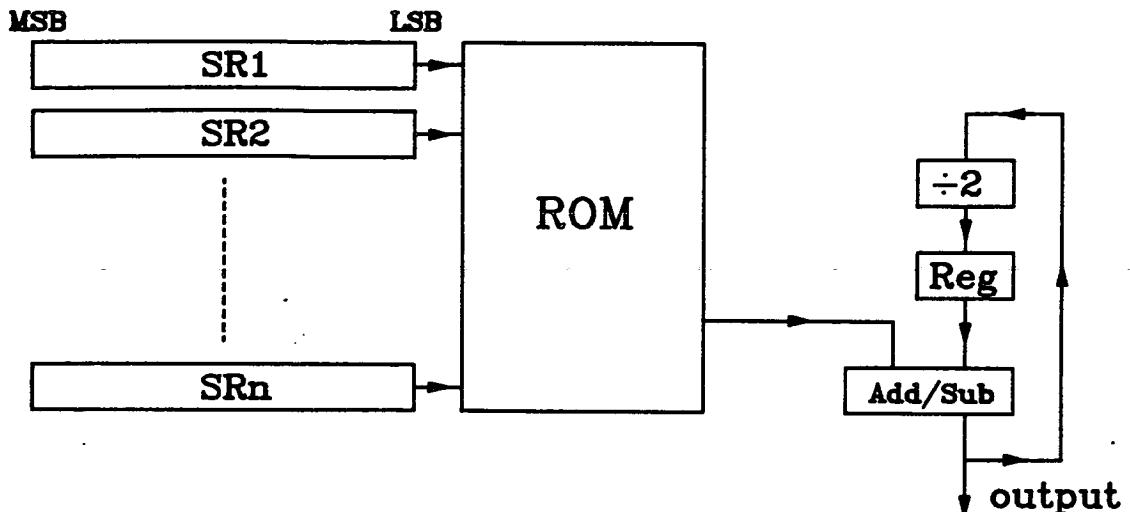


Figure 3-9: A distributed arithmetic unit

lines, and can therefore use the same address decoder. The complete computational unit, shown in Figure 3-8 (b), can then be repeated $N/2$ times to produce an N th order filter.

Of the circuits detailed in [37], only the shift-accumulate sections have been implemented as integrated circuits. Using a semi-custom aluminium-gate CMOS process, 1.1mm^2 of silicon area was required per coefficient bit for the shift- accumulator. Using a full custom $6\mu\text{m}$ CMOS process, this area was reduced to 0.57mm^2 . A $6\mu\text{m}$ NMOS shift- accumulate circuit has been reported, [40], requiring only 0.15mm^2 per bit.

Digital wave filters based on lattice reference structures have also been realised in integrated circuit

form. Van Ginderdeuren *et al* have reported the design of a third order elliptical lowpass lattice filter [38]. Similar designs, presented by Matsumura *et al*, detail the implementation of a 32 channel PCM transmit filter based on lattice reference filters.

In [38], Van Ginderdeuren reports the realisation of a third order lowpass lattice filter. In this design, the hardware requirements for the filter were defined by a suite of programs called DIGEST [41]. These programs offer two levels of operation. High level programs are available for modelling the filters as linear networks, without any finite wordlength effects. The designer can use these programs to determine the time and frequency response of the filter, and to perform noise and sensitivity analyses. The digital structure may also be modelled at the bit-level to evaluate small scale limit cycle and overflow responses. Further software tools have been reported for the optimization of the signal flow diagrams in an effort to minimize the hardware requirements and to generate suitable control signals for the filter.

The signal flowgraph for the third order wave lattice filter is given in Figure 3-10, and consists of three two- port adaptors, a subtractor and suitable delay stages. The circuit was initially designed for an NMOS

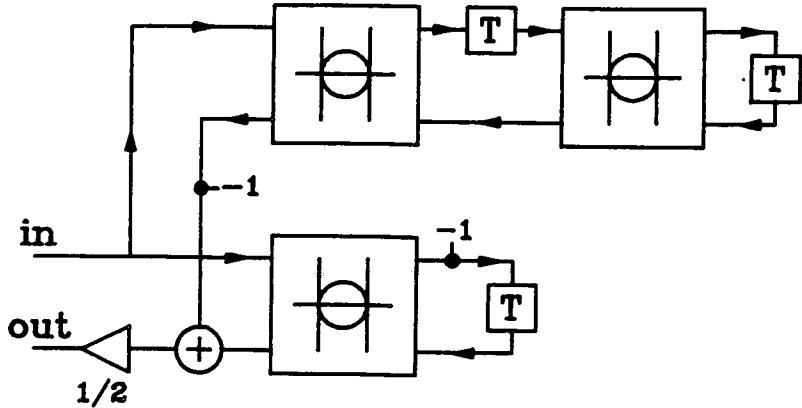


Figure 3-10: 3rd order elliptic wave lattice filter

fabrication process with a minimum linewidth of $6\mu\text{m}$. The multiplier coefficients for the adaptors have been optimized using Canonic Signed Digit (CSD) code, [42], to minimize the multiplier area required. The silicon area required for this design was 1.83mm^2 . However, this is predicted to reduce to 0.46mm^2 if a $3\mu\text{m}$ NMOS process was used. The circuit achieved a maximum clock rate of 5 MHz. As the data wordlength within the filter was 16 bits, this corresponds to a maximum sample rate of 312 kHz. Practical results for this circuit, as presented in [38], show close agreement with the predicted response.

Wave lattice filters have also been implemented as integrated circuits by Matsumura *et al* [39]. In this work, comparisons were made between implementing wave lattice filters using distributed arithmetic and by using serial/parallel multipliers with programmable coefficients. The distributed arithmetic technique described

in [39] is identical to that in [37] and uses shift-accumulator circuits and ROM to achieve the sum of products necessary to satisfy the two-port adaptor equations. The serial/parallel multiplier circuit is based on a circuit proposed by Lyon [43], and implemented by Myers [44]. This circuit uses programmable coefficients which allows the multiplier to be multiplexed between different adaptors or between different channels. This multiplier technique is again described in detail in Chapter 6.

As a comparison of the different multiplier techniques described above, a two-port adaptor was realised in integrated circuit form. Four separate adaptor designs were considered, two using fixed coefficient multipliers and two using programmable coefficient multipliers. Comparisons were made between the sampling period, T , in clock cycles and the silicon area required, A . The adaptors were designed on a $4\mu\text{m}$ Al-gate process using dynamic logic techniques. The multiplier coefficient wordlength, W_C , was 9 bits, and the data wordlength, W_d , 18 bits for each of the adaptor circuits. Results for the different adaptor implementations are summarised in Table 3-1. The table shows how the distributed arithmetic approach requires considerably greater silicon area than the other forms of implementation, producing the highest AT product. The sample rate of each

| Adaptor | T | A (mm ²) | AT |
|--------------------------|----|----------------------|----|
| Fixed coefficient | | | |
| dedicated multiplier | 28 | 0.68 | 19 |
| distributed arithmetic | 19 | 1.61 | 30 |
| Programmable coefficient | | | |
| without pipelining | 28 | 0.79 | 22 |
| with pipelining | 19 | 1.41 | 26 |

Table 3-1: Comparison of adaptor implementations
After Matsumura et al [39]

adaptor stage is defined by the time required to evaluate the adaptor equations. The dedicated multiplier adaptor and the programmable coefficient multiplier both evaluate a full precision product ($W_C + W_d$ bits), and therefore require 28 clock cycles between samples. In contrast, the distributed arithmetic multiplier and the programmable pipelined multiplier produce a truncated product of only W_d bits, and therefore require only 19 cycles between samples.

For a fixed sample rate system such as a PCM transmit filter, the system clock frequency is determined by the number of cycles required to evaluate each adaptor section. Clearly, the minimum system clock frequency can be achieved using either the distributed arithmetic multiplier or the programmable pipelined multiplier. Of these, the latter system requires less silicon area and this was one of the methods chosen for the design of the 32 channel PCM transmit filter. This filter operates with a 32 kHz input sample rate, which is then reduced to the standard telephony sample rate of 8kHz in two stages

using half-band wave lattice filters, as shown in Figure 3-11. The third stage of the filter provides the required bandpass response. The multiplier coefficients for the filter were optimized and lie between 3 and 8 bits of accuracy. The data wordlength for the system is 19 bits (including a guard bit), and a system clock frequency of over 8MHz is required to achieve the desired response.

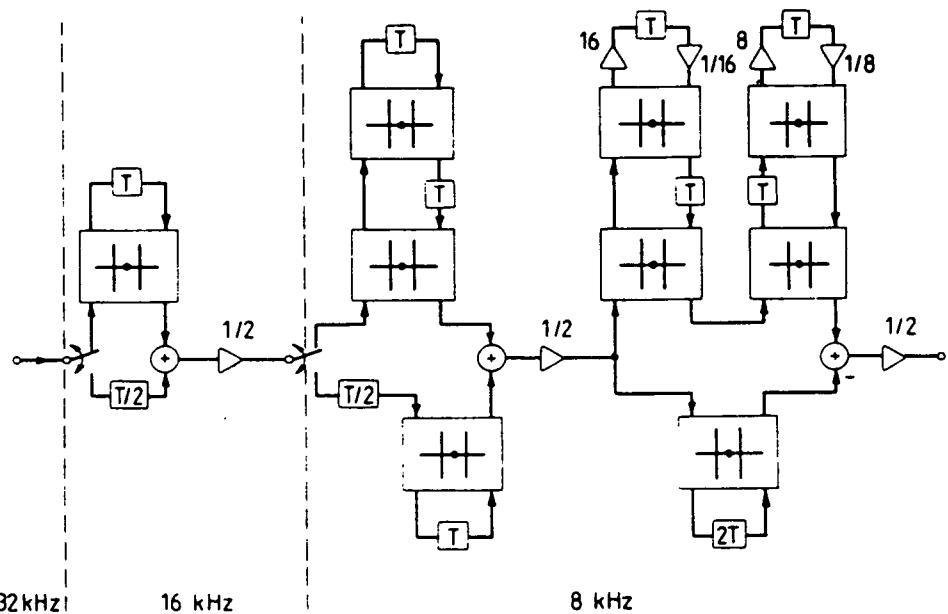


Figure 3-11: Wave flow diagram of PCM transmit filter

An alternative method for a complex filter such as that of Figure 3-11 is to use a state-space implementation as proposed by Wanhammar [45]. Using this method, the number of inner products to be evaluated is minimized with a consequent reduction in hardware requirements. This is achieved by only computing the filter output and

the signal values at the delay branches, and neglecting the difference equations at the other nodes in the filter. Distributed arithmetic techniques have been shown to be the most suitable way of computing the required inner products for this type of implementation [39]. Using this process, the important stability properties of the wave filter are retained.

This method was also used to design a PCM transmit filter. Due to the reduction in hardware requirements, a data wordlength of only 18 bits was possible, reducing the system clock frequency to 4.6MHz. Using this technique results in a more regular silicon layout with simplified testing procedures, but with the disadvantage that the response of the filter cannot be altered after the design stage. A summary of the alternative forms of PCM transmit filter implementation is given in Table 3-2.

| | Clock frequency MHz | Circuit area mm ² Arithmetic units | Total memory etc |
|--|---------------------------|--|---------------------|
| Fixed coefficients implementation | | | |
| Stage 1 | 6.144 | 3.7 | |
| Stage 2 | 6.656 | 5.8 | 33.9 |
| Stage 3 | 8.192 | 9.4 | |
| State-space Implementation | | | |
| Stage 1 | 4.608 | 5.8 | |
| Stage 2 | 4.608 | 5.8 | 32.5 |
| Stage 3 | 4.608 | 10.9 | |

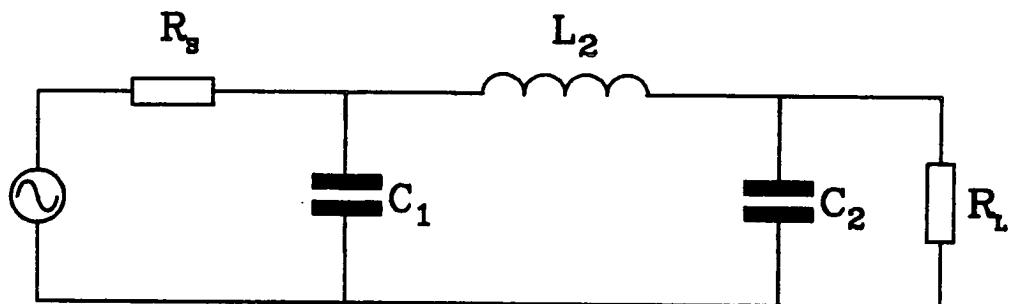
Table 3-2: Comparison of PCM filter implementation
After Matsumura et al [39]

3.3.3 Wave filters implemented using FDNR elements

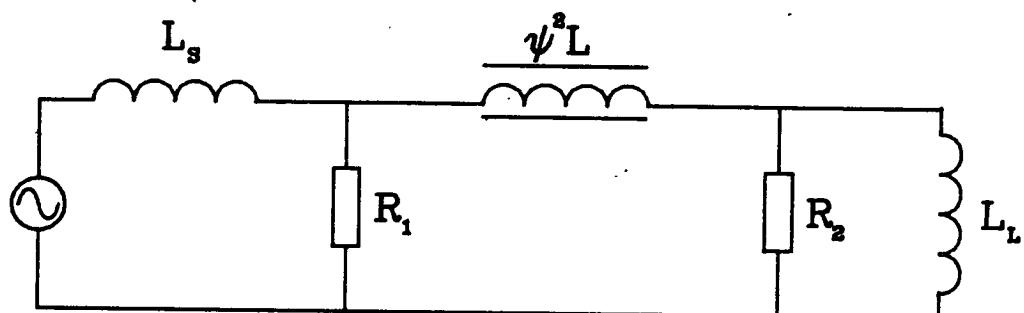
In section 2.4.7, a method of reducing the hardware requirements for certain classes of wave filters was proposed. Using this method, reference filters which had large numbers of inductors (or capacitors), could be transformed into filters which contained FDNR elements. Two types of FDNR element are possible. ψ^2 -impedance elements are generated when the reference filter is multiplied by an arbitrary scaling factor $K\psi$, and ψ^2 -admittance elements created when this factor is K/ψ . Wave filter equations and models for each of these elements were also given in section 2.4.7.

Digital wave filters designed using this approach have been simulated by McIntosh [46], using DINAP [47], a digital network analysis program. As an example of a wave filter implemented using FDNR elements, consider the third order, lowpass filter of Figure 3-12 (a). When the components of this filter are scaled by $K\psi$, the structure of Figure 3-12 (b) is generated. In this structure, the capacitors have been transformed to resistors, the resistors to inductors and the inductors to ψ^2 -impedance elements. This structure can be further transformed to a wave filter using the matched port technique given in Chapter 2, as shown in Figure 3-12 (c). The matched port technique has been used in preference to the Kuroda

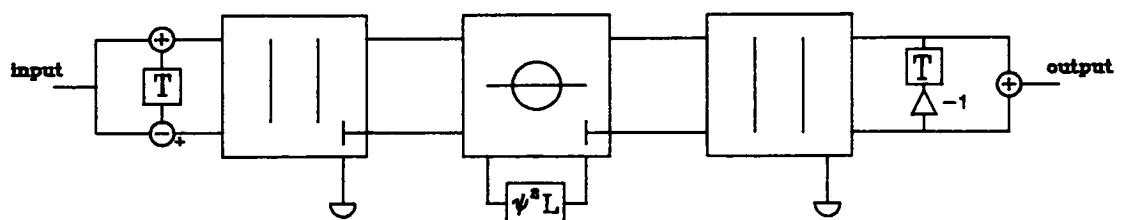
transform method, because transformation of unit-elements by $K\psi$ or K/ψ results in unrealisable structures.



(a)



(b)



(c)

Figure 3-12: FDNR wave filter realisation

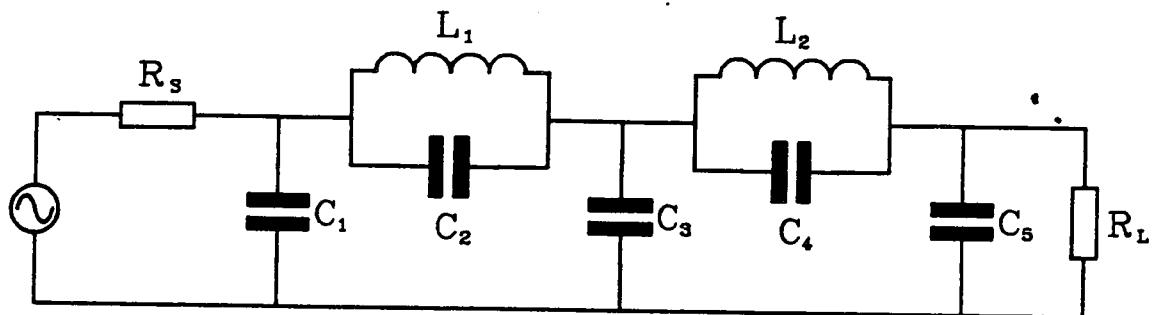
This technique can also be extended to higher order filters, and elliptic response filters.

The example filters presented in [46] included third order and fifth order Butterworth and Chebyshev lowpass filters. In addition, a fifth order elliptic lowpass filter was considered. Each filter was designed to have a cutoff frequency of 10% of the sample frequency. The passband ripple of the Chebyshev filters was 1.25dB. Component values for each of these filters were found from Saal [7].

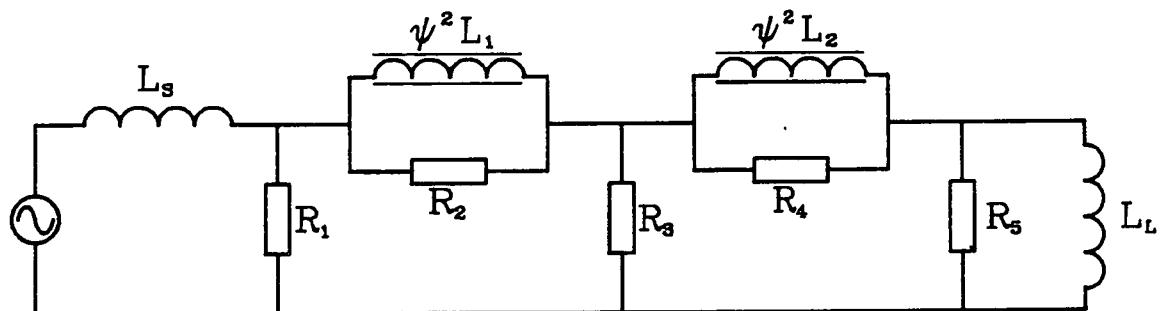
Comparison of the simulation results of the FDNR wave filters and conventional wave filters showed no difference between the frequency responses for both third and fifth order filters. Similarly, simulation of the FDNR and conventional elliptic wave filters resulted in identical frequency responses. The low sensitivity property of wave filters was retained in the FDNR structures.

The major advantage of the FDNR technique is that reference filters which have large numbers of inductors (capacitors), and few capacitors (inductors), can be transformed to ψ^2 -admittance (ψ^2 -impedance) type filters. However, for the Butterworth and Chebyshev filters simulated in [46], no such component imbalance exists, and no advantage is gained in transforming the filter. However,

if the fifth order elliptic filter of Figure 3-13 (a) is transformed by a factor of $K\psi$, then, as each capacitor is transformed to a resistor, the total hardware requirements can be reduced.



(a)



(b)

Figure 3-13: (a) 5th order elliptic lowpass filter
(b) transformation by factor ψ

A comparison of the hardware requirements for the example filters is given in Table 3-3.

From Table 3-3, it can be seen that hardware penalties exist where the number of capacitors and inductors are similar. However, when an imbalance in these component numbers exists, the FDNR transformations offer a reduction in the hardware required, but with a penalty of

| Filter | Delays | Multipliers | Adders | Total |
|------------------|--------|-------------|--------|-------|
| 3rd Order | | | | |
| WDF | 3 | 4 | 10 | 17 |
| FDNR($1/\psi$) | 6 | 4 | 19 | 29 |
| FDNR(ψ) | 4 | 4 | 14 | 22 |
| 5th order | | | | |
| WDF | 5 | 6 | 20 | 31 |
| FDNR($1/\psi$) | 8 | 6 | 27 | 41 |
| FDNR(ψ) | 6 | 6 | 22 | 34 |
| Elliptic | | | | |
| WDF | 7 | 10 | 28 | 45 |
| FDNR($1/\psi$) | 12 | 10 | 37 | 59 |
| FDNR(ψ) | 6 | 10 | 25 | 41 |

Table 3-3: Comparative element count

a more complex timing system.

3.4 Summary

In this Chapter, various methods used to realise wave filters have been presented. A wide range of techniques, from computer simulation to custom i.c. design have been used to implement wave filters based on a variety of reference structures from cascaded unit-element filters to FDNR ladder filters. Each of these techniques has advantages and disadvantages and the choice of implementation method is strongly dependent on the particular application for the filter.

Analogue wave filters offer a complete filter system on a chip, with a minimum number of external components required. They are efficient in their use of silicon area, and the introduction of circuits which are immune

to the effects of stray capacitance permits the use of very small integrated capacitors. However, these circuits suffer from the effects of noise as feature sizes are reduced, limiting the maximum possible dynamic range of the integrated filter. A further disadvantage is that the filter response is fixed at the time of design, preventing the filter being used in a multiplexed system.

Digital wave filters may be implemented in a wide variety of ways. Filters realised using general purpose hardware such as DSP microprocessors or dedicated multiplier circuits offer rapid system development and programmable filter response. This method may be used to realise filters which are part of a more complex signal processing system, with no significant increase in hardware. The principal drawbacks of this type of implementation are the amount of hardware required (and associated power consumption), and the limitations placed on the sample rate of the filter by factors such as the filter order and the complexity of other tasks which the central processor must perform. Integrated digital wave filters are more suitable where small physical size and low power consumption are required. Such filters can be designed to operate at very high sample rates, with no limitations such as filter order or "book-keeping" operations. They can be designed to model a wide range of reference filters with a programmable frequency response.

However, the design of integrated filters is expensive and time-consuming and many ways of evaluating the required filter equations have had to be developed in an effort to reduce the silicon area required to implement even simple, low order filters.

In the various implementations described above, only the filters based on general purpose hardware have offered a filter response which is truly programmable. Each of the integrated circuit filters had a frequency response which was fixed at the time of design for a particular application. Furthermore, each of the filters was based on a specific class of reference filter such as cascaded unit- element or lattice filter. Clearly, there exists a need for an integrated circuit capable of modelling a variety of reference filters in integrated circuit form. The development of such an integrated circuit forms the basis for the rest of this work.

Chapter 4

DIGITAL WAVE FILTER ADAPTOR REALISATIONS

4.1 Introduction

In Chapter 2, the theory of digital wave filters was developed to show how continuous-time passive filters could be modelled by sampled-data equivalent circuits. Examples of practical implementations of these circuits were presented in Chapter 3. However, in general, these filters have a fixed response, determined at the time of design. Any change in the filter order, cutoff frequency or passband ripple would require the filter structure to be re-designed. For dedicated filters such as analogue wave filters, or the wave lattice filters described in the previous chapter, this would involve a significant redesign of the overall wave filter structure. In the case of wave filters realised using general purpose hardware, the control program would have to be altered to account for such changes, with possible reductions in the filter sample rate. Clearly, some general purpose building block from which a variety of reference filters may be modelled by means of simple external programming, is highly desirable.

In the design of such a system, several criteria must be satisfied. The most important requirement is

that the overall response of the filter must be externally programmable. To achieve this, the building blocks must preferably be capable of evaluating adaptor equations for both two and three-port adaptors. This would enable any reference filter to be implemented as a wave filter. Furthermore, these adaptors should be directly cascadable, such that the order of the filter may be increased without affecting the sample rate of the system in any way. Finally, the adaptors must be capable of suppressing parasitic oscillations using the techniques described in [48], [49] and [50].

In this chapter, the first two criteria will be discussed. It can be shown that the suppression of parasitic oscillations can be achieved without altering the filter hardware in any way, and this process will be dealt with in Chapter 5, where the simulation of digital wave filters is considered.

4.1.1 Signal representation

In the implementation of wave filters, signal values may be realised in several ways. If the filter is to be simulated on a computer, the signals will be evaluated to the precision determined by the program. This will normally be to several decimal places, and can therefore be considered as a "perfect", theoretical value. Alternatively, the filter may be realised using microprocessors

or dedicated hardware, in which the signals must be realised as finite length binary words. Binary words may be realised as fixed-point numbers or as floating-point numbers.

Fixed-point numbers are such that the location of the binary point remains fixed for all arithmetic operations. These numbers may be represented in several ways, the most common of which are

1. sign and magnitude representation
2. two's complement representation

In a sign and magnitude representation, the binary word is given as a positive magnitude part with a sign bit as the most significant bit (MSB). For positive numbers, this sign bit is '0', and for negative numbers, the sign bit is a '1'. In two's complement form, decimal numbers are represented as follows.

$$N_{10} = -c_0 + \sum c_j 2^{-j} \quad (4.1)$$

where c_0 is the sign bit, and $c_j = 0$ or 1 for a $(w+1)$ bit binary number. In this system, the largest positive number that can be represented is $1 - 2^{-j}$, and the largest negative number is -1 .

Alternatively, these binary words may be represented as floating-point numbers, where

$$N_{10} = 2^{c_m} \quad (4.2)$$

where c is the exponent, and m is the mantissa of the number. The main advantage of floating-point numbers is that the range of numbers available is large, relative to fixed-point number systems. However, although in floating-point arithmetic multiplication and division are relatively simple, addition and subtraction are difficult to achieve due to the complexity of the system required [51].

In integrated digital wave filters, the circuit complexity and area requirements of the floating-point arithmetic systems outweigh the benefits of the large dynamic range and ease of multiplication that such a system offers. Therefore, the simpler, fixed-point two's complement system has been chosen for the implementation of these wave filter adaptors.

4.2 Bit Serial Arithmetic

In the evaluation of the adaptor equations for an integrated digital wave filter, two types of arithmetic may be used: word parallel arithmetic or bit-serial arithmetic. In parallel arithmetic systems, the data is processed one word at a time, by an array of arithmetic elements corresponding to the number of bits in the data word. The chief advantage of this system is that it offers the maximum data throughput as each arithmetic operation requires the minimum number of system clock

cycles. However, parallel arithmetic systems require large areas of silicon for both the arithmetic operations and the associated routing of both data and control signals.

In contrast, bit-serial arithmetic, as proposed by Lyon [52], processes data one bit at a time. Using this technique, data flows serially through the system, least significant bit first, with the data only moving 1 bit in every clock cycle. This method offers several advantages over the parallel arithmetic system discussed above. The most significant advantage of bit-serial arithmetic is the dramatic reduction in the silicon area required to implement the required algorithms. Because data is processed one bit at a time, instead of one word at a time, the area requirement for both the arithmetic operators and the signal routing is greatly reduced. In addition, this hardware reduction also leads to a lower power consumption and the minimum number of pins on the package in which the system is finally housed.

Implementing signal processing algorithms using serial arithmetic is similar to the methods used in parallel arithmetic systems. The signal flowgraphs required by the particular system are unchanged, but additional bit-delays are often required in order to synchronise signals as they progress through the system. In

bit-serial systems, signals move through the processing elements least significant bit first, and each arithmetic element is reset by a control signal as the LSB of each word reaches it. This feature of bit-arithmetic means that the data wordlength in such systems can be varied by altering the number of system clock cycles between each LSB signal. For these systems, there is no upper limit on the data wordlength. The bit delays required to synchronise the LSB of the signals are determined by the latency, or intrinsic delay, of each element in the signal flowgraph. Simple operators such as adders, subtractors and multiplexers, have a latency of 1 cycle, whereas a complex operator, such as a multiplier, has a latency determined by the arithmetic accuracy required.

As an example of the insertion of synchronising, or "shimming" delays [53], consider the addition of three bit-serial signals, $A_1 + A_2 + A_3$. The order in which these additions is carried out is unimportant, but in this example, A_1 is added to A_2 , and the sum added to A_3 . The addition (A_1+A_2) requires one system clock cycle, so A_3 must be delayed by one cycle before the second addition can be carried out. As the least significant bit of each word enters the adders, the CARRY part of each adder is cleared by the LSB control signal. The synchronised signal flowgraph and the appropriate timing diagrams are shown in Figure 4-1 (a) and (b).

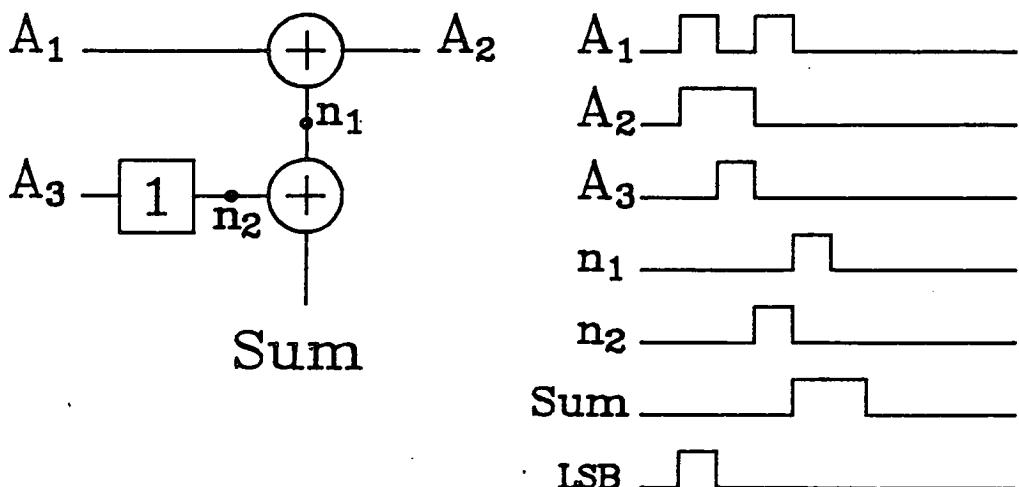


Figure 4-1: Example of synchronising delays in a bit-serial system

4.2.1 The FIRST silicon compiler

In the design of integrated circuits, two techniques may be used to generate the required silicon layout. Circuits may be custom designed to produce systems which are optimal in their use of silicon area, but require a long design time. Alternatively, semi-custom techniques which use automatic routing between pre-defined logic cells can be used to generate silicon layout in a much shorter design time than full-custom systems, but which generally contain redundant silicon area. Two semi-custom techniques are available - gate-array systems for low-level logic design using small numbers of gates, and silicon compilers for producing higher level circuits such as DSP systems. Silicon compilers use a language or pictorial representation of the particular system or

algorithm and produce a silicon design using pre-defined rules about placement and routing. A comprehensive description of compiler systems can be found in Reference [54].

The FIRST silicon compiler [55], developed at Edinburgh University, is a compiler system based on bit-serial arithmetic. FIRST uses a language description of a signal flow diagram as its input. From this, FIRST generates an automated layout of the desired system and provides suitable simulation data. A logic simulator is also included in the FIRST system to verify correctness of design and provides both a functional simulation and a verification that signals are properly synchronised through the system.

To implement a signal processing algorithm using FIRST, a language description of the signal flowgraph must be generated. Each node in the system is assigned an arbitrary name, and these names used to describe the interconnections of the elements in the network. The syntax of this description is given in the FIRST Simulators Users Manual [56]. System parameters such as the length of bit-delays or the number of coefficient bits in the multiplier primitives are passed to the primitives as parameters in the language description. Control information for each primitive, such as LSB control or data

selection in multiplexers, is also defined in the description of the system. As an example of the FIRST description of a signal flowgraph, again consider the addition as shown in Figure 4-1. The following FIRST input file will describe the network, with the adders being reset at times "clear1" and "clear2".

```
ADD [1,0,0,0] (clear1) A1, A2, GND -> n1, NC  
ADD [1,0,0,0] (clear2) n1, n2, GND -> SUM, NC  
BITDELAY [1] A3 -> n2
```

A more detailed example of the FIRST implementation of a signal flowgraph is presented in Appendix II.

In the FIRST system, a wide range of primitives are offered. In the implementation of wave filters, the range of operators required is limited to

ADDERS
SUBTRACTORS
MULTIPLIERS
BIT-DELAYS
MULTIPLEXERS

The adders used in by FIRST are carry-save, whereby a carry, generated by an addition, is stored in a latch, and used by the adder in the subsequent operation. Similarly, the subtractors retain borrow signals until the next subtraction. These latched signals are cleared by the LSB signal of each new data word. The multiplier architecture used in the FIRST system is the modified Booth's algorithm, as proposed by Rubenfield [57], and implemented by Lyon [43]. The range of multiplier

coefficients, m , available from this FIRST operator is

$$-1 \leq m < 1$$

The coefficient wordlength for the multiplier is determined by the arithmetic accuracy required, and is always an even number of bits long. This is a characteristic of the multiplier algorithm used. As with other arithmetic operators, the multiplier is reset as each new data word enters the structure. To prevent race hazards, the multiplexers used by FIRST have an internal delay of at least one system clock cycle. The clocking scheme used by FIRST is a two-phase, non-overlapping clock, with signals clocked into the system on φ_2 , and clocked out on φ_1 . This convention is used to simplify the interfacing of FIRST systems to the positive-edge triggered clocking scheme of TTL logic, where signals are stable when the system clock, φ_1 is low (and φ_2 is high).

The silicon layout generated by FIRST is based on a series of pre-defined building blocks such as adders, subtractors, multipliers, bit-delays and multiplexers. These elements are initially custom designed according to the design rules defined by the fabrication process involved. The FIRST system places each required operator in a suitable arrangement, and then routes the nodal connections through one central routing channel. An example of this routing system is given in Appendix II.

The use of one central channel for the routing interconnection guarantees 100% routing between the circuit nodes, as the width of the channel may be increased to accommodate all the required connections.

The simulation of networks implemented using FIRST is detailed in Chapter 5.

4.3 Two-Port Adaptors

4.3.1 Hardware requirements

In the implementation of digital wave filters, the simplest adaptor form is the two-port adaptor. This adaptor is used in the realisation of both cascaded unit-element filters and wave lattice filters, as described in the previous chapter. The two-port adaptor is used to provide a simple change of reference resistance.

The adaptor equations to be satisfied are

$$B_1 = A_2 + \alpha(A_2 - A_1) \quad (4.3)$$

$$B_2 = A_1 + \alpha(A_2 - A_1) \quad (4.4)$$

where

$$\alpha = (R_1 - R_2) / (R_1 + R_2) \quad (4.5)$$

Equation (4.5) shows that for positive port resistances, the magnitude of the multiplier coefficient $|\alpha| < 1$, which is within the range of the multiplier offered by

FIRST. Possible realisations of these adaptor equations are given in Figure 4-2.

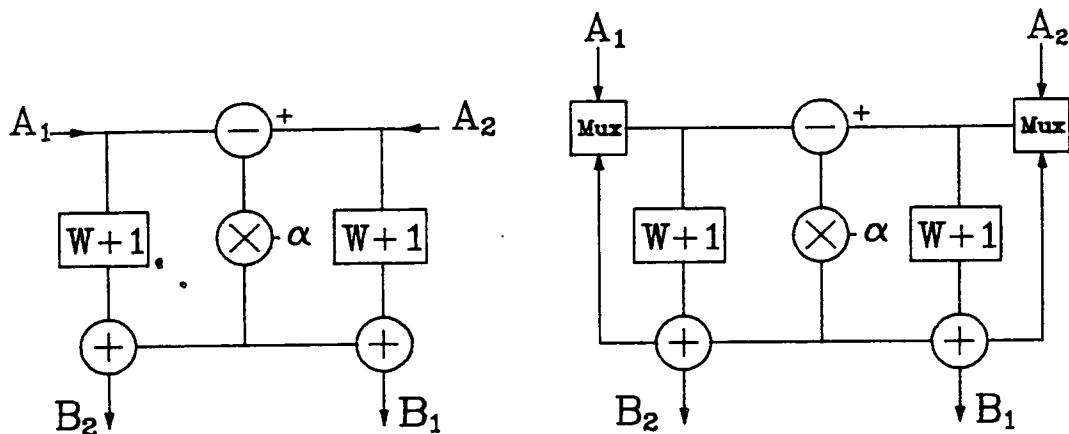


Figure 4-2: Two port adaptor signal flowgraphs

In these diagrams, the synchronising delays required by the system have been included. The length of these delays is determined by the latency of the multiplier. For the modified Booth's algorithm, the multiplier latency, W , is given by

$$W = 3 * m / 2 + 2 \text{ cycles} \quad (4.6)$$

where m is the coefficient wordlength. Therefore, a delay of $W + 1$ bits is required to synchronise the signals in the adaptor.

4.3.2 Adaptor latency, sample rate and signal wordlength

The total latency of the two port adaptor is given by summing the individual latencies of each arithmetic operator. For a correctly synchronised system, the

latency measured between any input and any output must be constant. For the adaptor of Figure 4-2 (a), the latency is given by

| | |
|------------|----------------|
| Subtractor | 1 bit |
| Multiplier | $3*m/2+2$ bits |
| Adder | 1 bit |
| Total | $3*m/2+4$ bits |

As shown in Chapter 2, these operations must be carried out in $T/2$ cycles, where T is the number of system clock cycles between samples.

The maximum sample frequency, F_S , of a filter using the two port adaptor structure of Figure 4-2 (a) would therefore be

$$F_S = F_C / T \quad (4.7)$$

$$T = 2 * (3 * m / 2 + 4) \text{cycles} \quad (4.8)$$

where F_S is the system clock rate.

The minimum data wordlength within a bit-serial adaptor is determined by the latency of the adaptor type used in the filter. However, several of the most significant bits are usually required to allow for wordgrowth within the system. The number of "guard bits" required is determined by an analysis of the signal flowgraph of the adaptor. For the two port adaptor of Figure 4-2, guard bits are required by the subtractor, and the adders at the adaptor outputs. Two guard bits are also required by the multiplier to ensure correct arithmetic operation,

although these bits are passed through unchanged by the multiplier. Therefore, a total of three guard bits are required at the MSB end of the data word - one for the subtractor, and two for the multiplier. The guard bit required by the adders is incorporated in the multiplier requirement. For an adaptor with a latency of $T/2$ cycles, the data wordlength will be $T/2$ bits, with a maximum signal wordlength of $T/2 - 3$ bits.

As an example of a filter implemented using the adaptor structure of Figure 4-2 (a), consider the implementation of a 3rd order lowpass filter, with a multiplier coefficient wordlength of 8 bits. An N th order filter requires $N + 1$ adaptors, so four sets of adaptor equations must be evaluated for the example filter. If four adaptors are used, the filter sample rate is given by

$$F_S = F_C / T \quad (4.9)$$

For this example, with a coefficient wordlength, m , of 8 bits, the sample period T is given by

$$\begin{aligned} T &= 2 (3 * m / 2 + 4) \\ &= 2 (12 + 4) \\ &= 32 \text{ cycles} \end{aligned}$$

With a typical system clock frequency of 8 MHz, a filter of this type could sample at a maximum frequency of

$$\begin{aligned}F_S &= F_C / T \\&= 8 * 10^6 / 32 \\&\approx 250 \text{ kHz}\end{aligned}$$

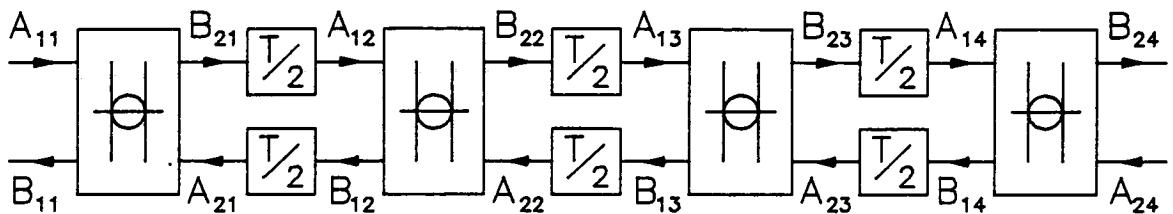


Figure 4-3: 3rd order lowpass wave filter

In situations where the hardware requirement is to be minimised, and a high sample rate is not necessary, the filter sample rate can be reduced by multiplexing the available hardware. Initially, this can be achieved by using one adaptor to evaluate two sets of adaptor equations in the sample period, T . As cascaded unit-element filters are half-synchronous systems, this can be achieved without any corresponding reduction in the filter sample rate. Consider the 3rd order example filter of Figure 4-3. In this structure, all odd numbered adaptors operate in the first half of the sample period, and all the even numbered adaptors in the second half of T . Adaptors (1) and (2) may therefore be combined to use the same hardware, without affecting the operation of the filter in any significant way. This technique can be achieved by adding a multiplexer to each signal input of

the adaptor, as shown in Figure 4-2 (b). The adaptors can then be used to either accept new data from outside the adaptor in the first $T/2$, or re-cycle the previous outputs in the second $T/2$. This multiplexing scheme offers a two-fold reduction in the hardware requirement for the filter adaptors, but one extra cycle of latency, caused by the clocked multiplexer, is added to the latency of the adaptor. This effect will increase the number of system clock cycles between samples by two.

This technique can be extended to further reduce the hardware required to achieve the desired filter response. Intermediate adaptor outputs may be stored in RAM, and recalled under the command of a stored program, to minimise the hardware, but reduce the sample rate. For half-synchronous filters, the sample rate may be reduced by a factor of $(n + 1) / 2$ where n is the filter order.

4.4 Three-Port Adaptors

4.4.1 Parallel three-port adaptors

To implement a parallel three-port adaptor, the following equations must be satisfied.

$$B_k = A_0 - A_k \quad (4.10)$$

where

$$A_0 = \sum_{k=1}^3 \alpha_k A_k \quad (4.11)$$

$$\alpha_k = 2G_k / (G_1 + G_2 + G_3) \quad (4.12)$$

$$G_k = 1/R_k \quad (4.13)$$

We can reduce the number of multipliers required by choosing port n as a *dependent* port. If we choose port 3 as the dependent port, then equation 4.11 reduces to

$$A_0 = 2A_3 + \alpha_1(A_1 - A_3) + \alpha_2(A_2 - A_3) \quad (4.14)$$

From equation 4.12, we can show that the sum of all the α coefficients is 2. Although this sum is predictable, the magnitude of any individual coefficient is dependent on the nature of the reference filter. For this reason, we cannot guarantee that the coefficients will fall within the range of coefficients available from the FIRST multiplier. However, if all the α_k coefficients were replaced by $(\alpha_k - 1)$, then the range of coefficients would always fall within the range offered by the multiplier. i.e. $-1 \leq \alpha_k - 1 < 1$.

A possible realisation of a parallel three-port adaptor, using multiplier coefficients ($m_k = \alpha_k - 1$), is presented in Figure 4-4. As before, bit-delays have been added to the circuit to correctly synchronise the signals. These bit delays are again dependent on the coefficient wordlength of the multiplier.

This adaptor circuit is therefore capable of modelling any parallel three-port connection, depending only

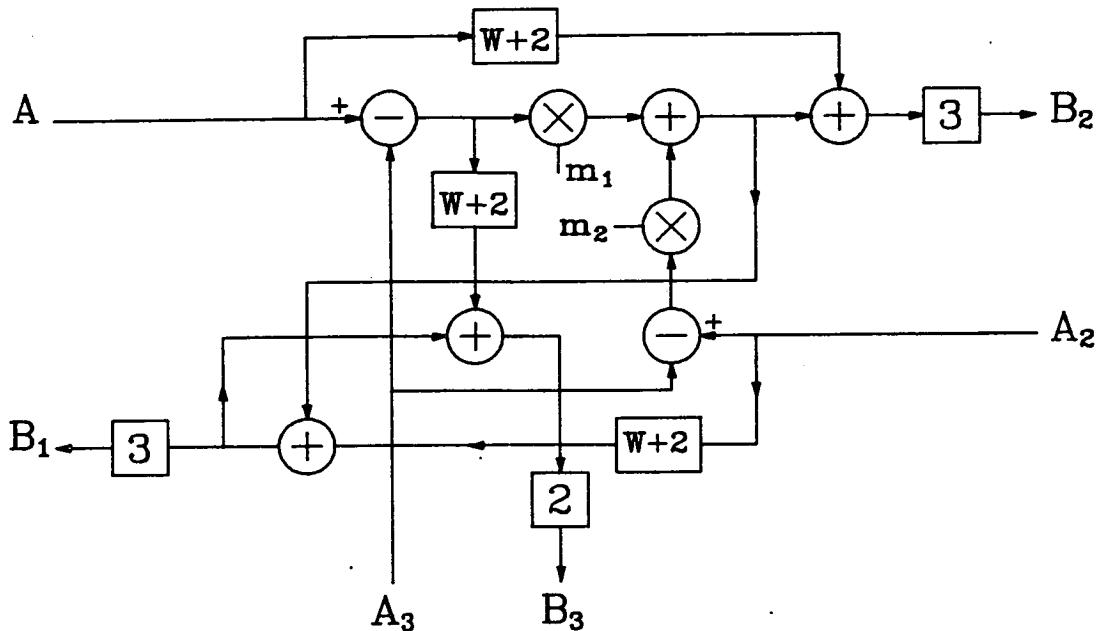


Figure 4-4: Parallel 3-port adaptor circuit

on the filtering element required on port 3 of the structure. The sample period delay required by a capacitive filtering element can be realised by adding a $T/2$ bit shift register between B_3 and A_3 . This delay is combined with the $T/2$ cycles delay inherent in the adaptor. For an inductive filtering element, the data word must be multiplied by -1. This can be achieved by subtracting the data word from zero. As this operation requires one clock cycle, the length of the shift register must be reduced to $(T/2 - 1)$ bits.

4.4.2 Series three-port adaptors

In the implementation of a series three-port adaptor, the following equations must be satisfied.

$$B_k = A_k - \beta_k A_O \quad (4.15)$$

where

$$A_O = A_1 + A_2 + A_3 \quad (4.16)$$

and

$$\beta_k = 2R_k / (R_1 + R_2 + R_3) \quad (4.17)$$

If port three is again chosen as the dependent port, then we can write,

$$B_n = - A_O - (B_1 + B_2) \quad (4.18)$$

and the adaptor can be realised using only two multipliers.

As in the case of the parallel adaptor, we know that $\sum \beta_k = 2$, but we cannot guarantee that $\beta < 1$. Therefore, the adaptor equations must be altered to bring the multiplier coefficient within the range $-1 \leq m < 1$. If β_k is replaced by $(1 - \beta_k)$, then this condition is assured, as $-1 \leq 1 - \beta_k < 1$ is always true for positive port resistances.

A possible realisation of these equation is given in Figure 4-5, with multiplier coefficients $m_k = 1 - \beta_k$. As before, the length of the synchronising bit-delays is dependent on the multiplier latency, W .

This circuit is capable of modelling the adaptor equations for any series three-port connection. Again,

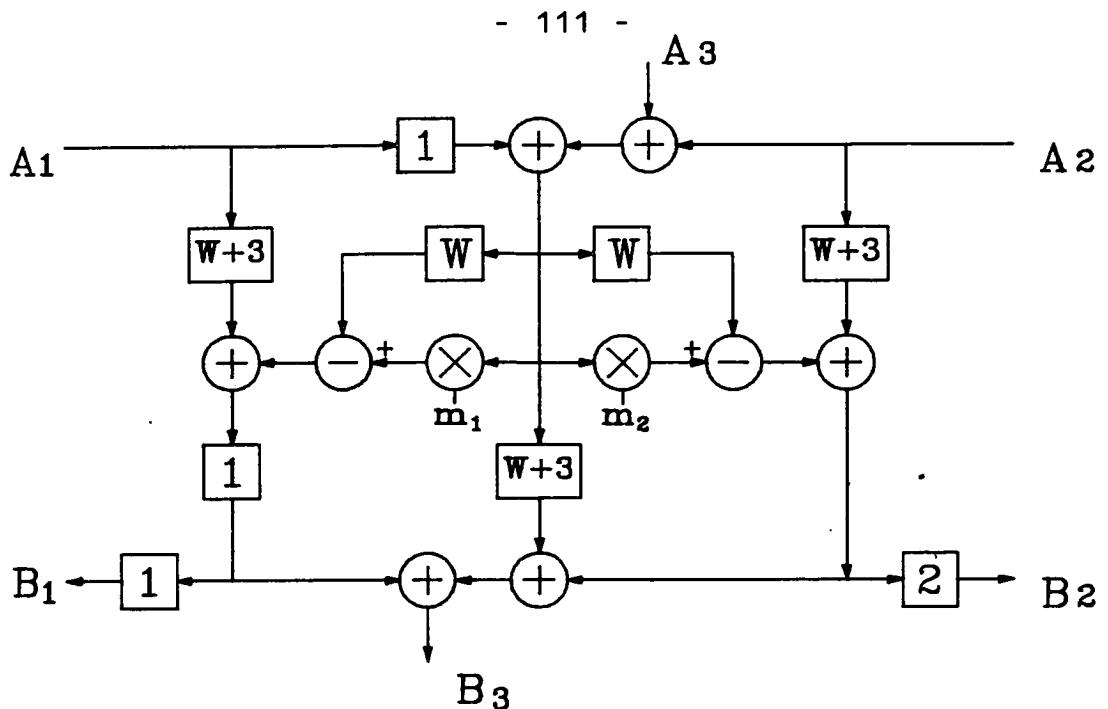


Figure 4-5: Series 3-port adaptor circuit

the filtering element, present on port 3, must be properly synchronised with the signals in the adaptor itself.

4.4.3 Adaptor latency, sample rate and signal wordlength

In the wave filter realisation of RLC ladder filters, it is often necessary to use both parallel and series adaptors. For this reason, it is essential that both three-port adaptors have equal latencies. If the latencies are the same, then the system sample rate will be constant for any reference filter (assuming no time-division multiplexing).

In the circuit diagrams of Figures 4-4 and 4-5, the latencies of both adaptors have been made equal by adding additional bit delays at the outputs of the parallel

adaptor. For the series adaptor circuit, the latency, measured between A_1 and B_1 is given by

| | |
|----------------------|--------------------|
| 1 multiplier | $3*m/2 + 2$ cycles |
| 3 adder/subtractors | 3 cycles |
| 3 synchronising bits | 3 cycles |

giving a total of $W + 6$ cycles. As this is larger than the minimum latency of the parallel adaptor, $W + 3$ cycles, additional bit delays have been added to the outputs of the parallel adaptor to equalise the latencies.

As these adaptors are designed for implementing half-synchronous systems, the system sample rate, F_S , is dependent only on the adaptor latency, L_{ad} , and is given by

$$F_S = F_C / 2 * L_{ad}$$

The time-division multiplexing system described for two-port adaptors in the previous section, can also be applied to three-port adaptors. However, this is only feasible if all the adaptors in the wave filter are parallel (or series). Time-division multiplexing of wave filters implemented with parallel and series adaptors is theoretically possible, but would require complex control circuitry.

The data wordlength within the adaptors of Figures 4-4 and 4-5 is defined by the adaptor latency, L_{ad} . However, as in the case of two-port adaptors, guard bits

must be allowed at the MSB end of the word, to allow for wordgrowth in the filter system. For both adaptors, guard bits are required for each adder and subtractor, and two bits required for the multiplier. Again the guard bits required by the multiplier are passed unchanged, and may be used for subsequent operators.

Within both adaptor systems, the number of guard bits required is determined by analysing the "worst-case" signal path between any input and any output. The number of guard bits required by the parallel adaptor circuit is given by following the signal path from A_1 , A_2 , or A_3 to B_3 . In this path, guard bits are required for three adders, one subtractor and one multiplier, giving a total of 4 bits. In contrast, the worst-case signal path of the series adaptor, from A_2 to B_3 , contains five adders, one subtractor and one multiplier. As the guard bits required by the multiplier can be used by the other operators, 6 guard bits are required in the series adaptor.

For filters using only parallel adaptors, the maximum permissible signal wordlength will therefore be $L_{ad} - 4$ bits. For filters implemented using only series adaptors, or series and parallel adaptors, the maximum signal will reduce to $L_{ad} - 6$ bits. It should be noted that these maximum signal levels are instantaneous values

and do not correspond directly to the maximum signal level that may be applied to the filter input.

For an example filter, implemented using parallel and series adaptors, with multiplier coefficient wordlengths of 8 bits and a system clock frequency of 8 MHz, the sample rate will be given by

$$\begin{aligned} F_s &= F_c / 2 * L_{ad} \\ &= 8 * 10^6 / 2 (18) \\ &\approx 210 \text{ kHz} \end{aligned}$$

This sample rate assumes that no time-division multiplexing is being used.

4.5 General Purpose Wave Filter Adaptor

The realisation of integrated wave filters using both parallel and series adaptors is both inefficient and expensive, as two separate circuits are required. If these circuits could be combined to form a general purpose adaptor, the design of both the adaptor circuit and the resulting wave filter could be greatly simplified.

A circuit capable of implementing both parallel and series three-port adaptor equations is presented in Figure 4-6. Throughout the circuit, multiplexers have been used to control the flow of data between the arithmetic operators, depending on the adaptor type required, parallel (P) or series (S). In this circuit, the multiplexers

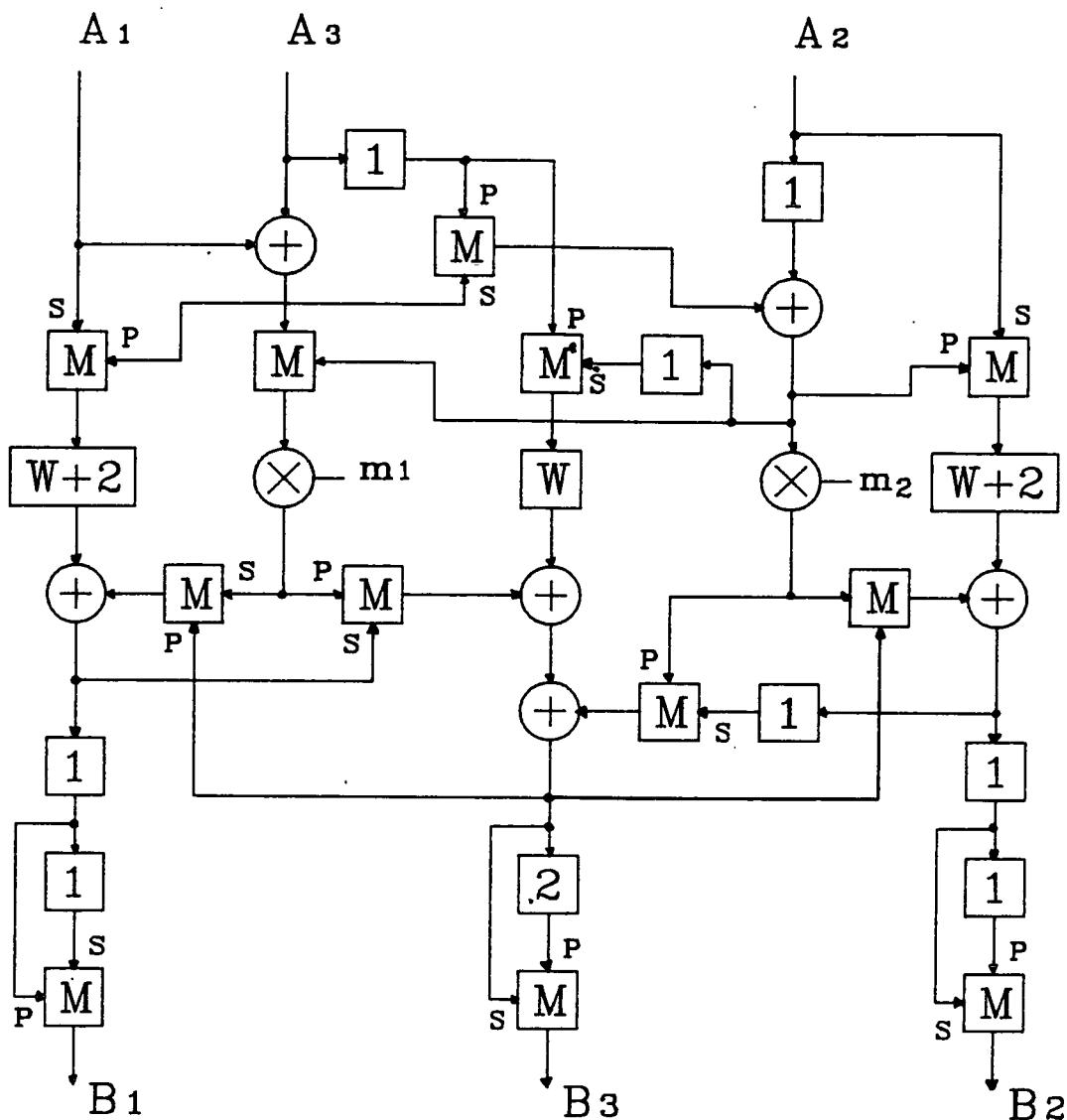


Figure 4-6: General purpose three-port adaptor

should be considered merely as switches, and do not have any latency.

4.5.1 Adaptor latency, sample rate and signal wordlength

The latency of the general purpose adaptor of Figure 4-6 is given by summing only the latencies of the

arithmetic elements in the circuit. If we again assume that an 8 bit modified Booth's algorithm is to be used, the circuit latency will be 19 cycles - 14 cycles for the multiplier, and 5 cycles for the other arithmetic elements and the required synchronising delays.

An example filter realised using this circuit, and with a system clock rate of 8MHz, would have a maximum sample rate of

$$F_S = 8 * 10^6 / (2 * 19)$$

$$\approx 210 \text{ kHz}$$

The general purpose adaptor circuit of Figure 4-6 has been realised in integrated circuit form, and is detailed in Chapter 6.

4.6 Universal Wave Filter Adaptor

In section 4.1, conditions were specified for a completely general purpose wave filter adaptor. Such a circuit would have to satisfy the following conditions:

1. The circuit must be externally programmable to realise parallel or series adaptors, with capacitive or inductive filtering elements. This permits any reference filter to be realised as a wave filter.

2. The adaptors must be capable of being cascaded to permit filters of any order to be implemented. Furthermore, the structures should be designed so that the adaptor could be multiplexed in situations where the hardware requirement was to be minimised.
3. The circuit must be capable of operating in both halves of the sample period, i.e. it must have a 100% duty cycle.

A structure which satisfies these conditions is given in Figure 4-7. As in the case of the general purpose adaptor, extensive use of multiplexers has been made to direct the flow of data between the appropriate elements for implementing the parallel (P) or series (S) adaptor equations. In this circuit, an internal connection has been provided between B_3 and A_3 to model inductive or capacitive filtering elements. This is controlled by a multiplexer which selects B_3 or $-B_3$ depending on the filter requirements. An additional multiplexer has been included in this path to allow the feedback loop to be broken to simplify the testing of the circuit. To ensure a 100% duty cycle, this adaptor has multiplexers at its inputs to determine whether data is to be accepted from another universal adaptor, or whether the data is to be re-cycled during the second half of the sample period. The multiplexers used in the circuit of

Figure 4-7 differ from those in the general purpose adaptor, in that they have a minimum latency of 1 cycle. This permits the adaptor to change between implementing parallel and series equations in synchronism with the LSB of each data word.

This circuit has been developed to model three-port parallel or series circuits. However, the adaptor equations for two-port adaptors are just a special case of the more general n -port parallel or series adaptor equations, so an additional requirement of a completely general purpose adaptor is that it should be capable of realising the two-port adaptor equations directly. This condition can be achieved by the circuit of Figure 4-7 by configuring the circuit as a parallel adaptor, and setting the m_1 coefficient to be α , and the m_2 coefficient to be $-\alpha$. Furthermore, the multiplexer which breaks the $B_3 - A_3$ connection should have its external connection grounded.

Because of the completely general nature of this adaptor, the term Universal Adaptor will be used to describe the circuit in the remainder of this work. The symbol used to represent the universal adaptor is given in Figure 4-8. The universal adaptor concept has been developed to minimise the number of external connections required to control the wave filter. For this reason, no

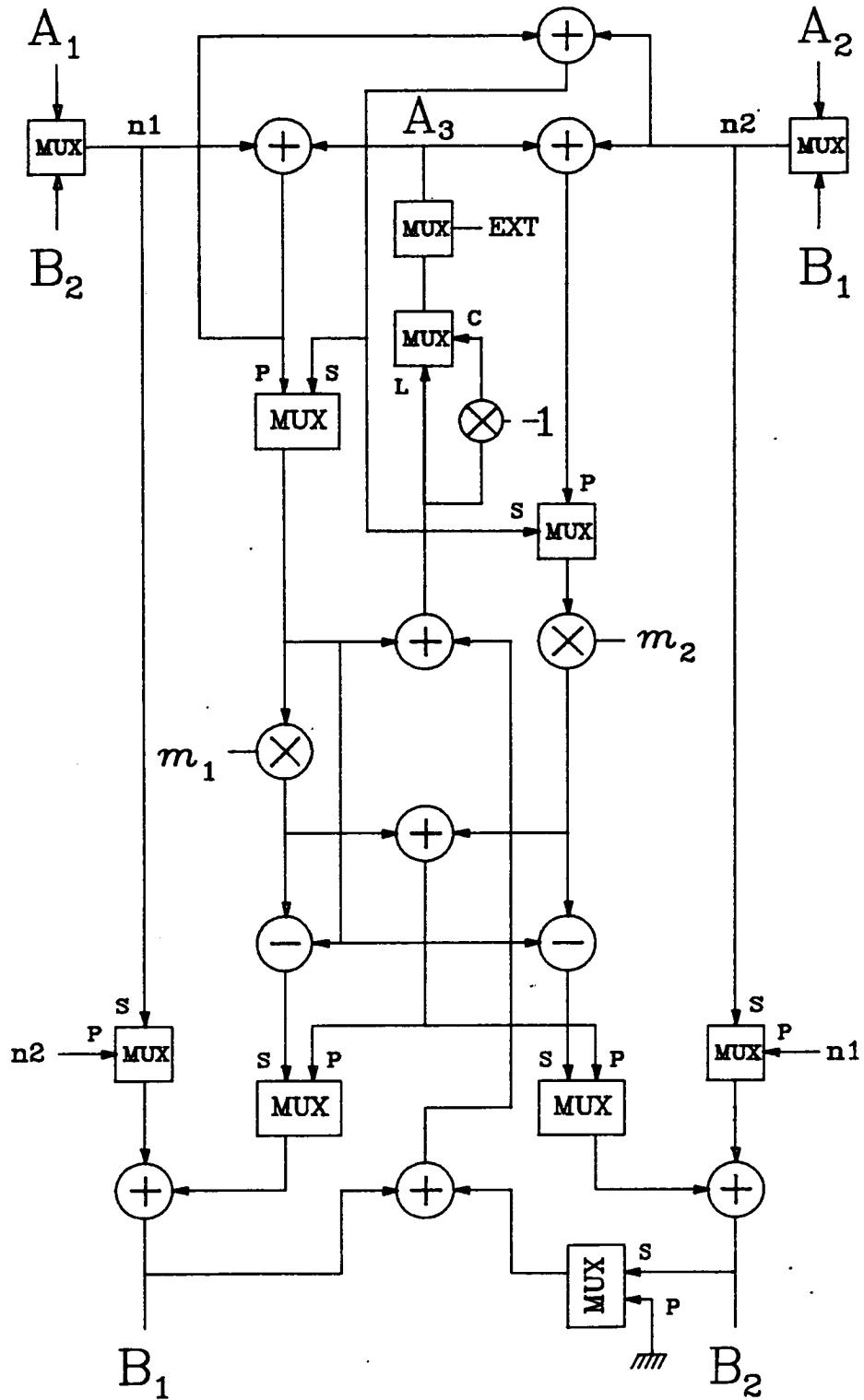


Figure 4-7: Universal wave filter adaptor

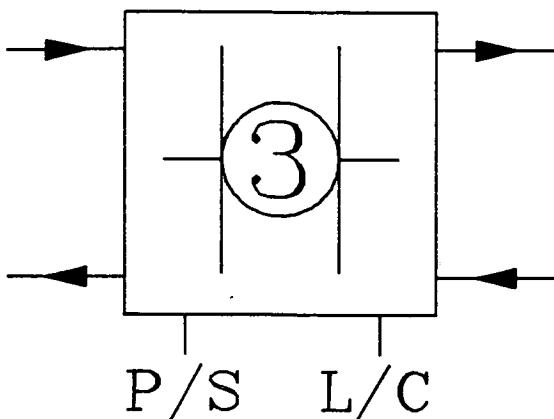


Figure 4-8: Universal adaptor symbol

port 3 connections have been included in the new adaptor symbol. The number in the center of the symbol shows that this universal adaptor can only model two and three-port adaptor equations. However, this concept may be extended to any number of ports and the symbol can be used to indicate the complexity of the adaptor circuit.

Due to the complexity of the adaptor, the synchronising bit-delays required by the circuit have not been given in Figure 4-7. However, a more detailed timing diagram of the adaptor is given in Appendix II. The FIRST input file describing this circuit is also given in this appendix.

In the universal adaptor circuit, the multiplier coefficients used are $(\alpha_k - 1)$ and $(1 - \beta_k)$, to ensure that all reference structures can be modelled using the FIRST system.

4.6.1 Adaptor latency, sample rate and signal wordlength

The latency of the universal adaptor can again be calculated using the methods described previously. In the signal path between either A_1 or A_2 and the appropriate output node, there are 3 adders, 3 multiplexers and a multiplier. A further one cycle is required for correct synchronisation (see Appendix II). As this circuit has been developed entirely using the FIRST system, an extra one cycle is required for the signals going into and coming out of the system, giving a total latency of

$$L_{ad} = 3 * m / 2 + 10 \text{ cycles}$$

where m is the multiplier coefficient wordlength.

The sample rate of the filter, assuming no time-division multiplexing, will therefore be given by:

$$\begin{aligned} F_s &= F_C / 2 * L_{ad} \\ &= F_C / (3m + 20) \end{aligned}$$

Again, for typical coefficient wordlengths and system clock rates of 8 bits and 8 MHz respectively, a maximum sample rate of approximately 180 kHz is possible.

The data wordlength for the universal adaptor is again given by the adaptor latency. The maximum signal levels within the adaptor can be determined by the number of guard bits required by the parallel and series circuit configurations. For parallel adaptors, 4 guard bits are

required, whilst 6 bits are required by series adaptors. As before, wave filters realised using parallel and series adaptors must limit signal levels to L_{ad} - 6 bits.

4.7 Summary

In this chapter, the implementation of digital wave filter adaptors has been discussed. Circuits have been proposed for conventional two and three port adaptors, and new circuits developed for general purpose wave filter adaptors. The realisation of wave filters using digital adaptors can be divided into two main classes of arithmetic : word-parallel and bit-serial. The circuits described in this chapter have been developed for use in bit-serial systems, although the general architectures may be used in parallel arithmetic circuits.

The use of bit-serial arithmetic has introduced the concept of circuit latencies which determine the sample rate of the filter. The latency of an adaptor is given by summing the inherent delays of all the arithmetic operators and any synchronising delays which may be required. The sample rate of the resulting wave filter is given by dividing the system clock rate, F_C , by twice the latency of the adaptors used in the circuit. Although the data wordlength is given by the adaptor latency, the maximum signal level possible within the adaptor is determined by the arithmetic elements used in

the circuit. Guard bits are required at the MSB end of the data word to allow for bit-growth in adders and subtractors, and to ensure arithmetic correctness in the multiplier circuit.

In situations where high sample rates are not required, the hardware requirement can be reduced. This process, known as time-division multiplexing, can be used to reduce the hardware required by a factor of 2 with no significant reduction in the filter sample rate. The hardware can be further reduced using time-division multiplexing, but the filter sample rate is reduced in direct proportion to the hardware reduction.

Circuits have been described in detail for two-port adaptors and both series and parallel three-port adaptors. These circuits have been extended to use multipliers whose coefficients, m , lie within the range $-1 \leq m < -1$. This condition allows wave filters, modelling any reference filter, to be designed using the FIRST silicon compiler system. The circuit requirements for parallel and series adaptors have been combined to form a general purpose wave filter adaptor which can be controlled externally to realise adaptors of either type. This general purpose circuit has been developed further to provide a Universal Adaptor, capable of modelling parallel and/or series adaptors with inductive or capaci-

tive filtering elements connected across port 3 of the adaptor. The universal adaptor has a 100% duty cycle, reducing the hardware requirements for practical implementations of wave filters. These features of the universal adaptor satisfy the criteria defined in section 4.1 for a completely general purpose "building-block" for the rapid development of wave filter systems.

The adaptor latencies, data and signal wordlengths and typical sample rates of filters implemented using the adaptors described in this chapter are summarised in Table 4-1. The figures quoted are for adaptors using a system clock rate of 8 MHz and multiplier coefficient wordlengths of 8 bits.

| Adaptor type | L_{ad} | Data wordlengths | Signal wordlengths | Sample rate kHz |
|-----------------|----------|------------------|--------------------|-----------------|
| 2 port | 14 | 14 | 11 | 285 |
| series 3 port | 20 | 20 | 14 | 200 |
| parallel 3-port | 20 | 20 | 16 | 200 |
| general purpose | 19 | 19 | 14 | 210 |
| universal | 22 | 22 | 16 | 180 |

Table 4-1 : Summary of digital adaptor implementations

Chapter 5

SIMULATION OF DIGITAL WAVE FILTERS

5.1 Introduction

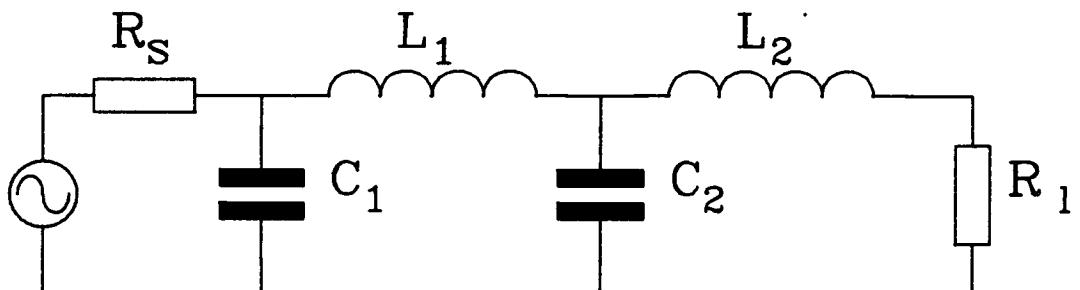
In the previous chapter, a number of circuits were presented for the digital implementation of wave filter adaptors. Initially, two-port adaptors were examined, but the range of reference filters which may be realised using these structures is limited. A wider range of reference filters may be implemented using parallel and series three-port adaptors, and circuits were also presented for these structures. These circuits were then combined to form a general purpose wave filter adaptor, capable of implementing the adaptor equations for parallel and series connections. Finally, a universal adaptor circuit was proposed, which can be configured to operate as a parallel or series adaptor, with a capacitive or inductive filtering element connected across port 3, and which operates with a 100% duty cycle.

In this chapter, wave filters implemented using universal adaptors will be considered. A seventh-order lowpass filter will be realised as a wave filter and the resulting structure simulated using the FIRST compiler system described in the previous chapter. Details of the frequency response, signal levels and insertion loss will

be given, and techniques presented for the elimination of parasitic oscillations.

5.2 Reference Filter for Simulation of Filter System

As an example of a wave filter implemented using the universal adaptor of section 4.6, consider the fourth order lowpass reference filter of Figure 5-1. The component values given are for a Chebyshev response with a passband ripple of 0.644dB. The cutoff frequency of the filter was designed to be 20% of the sample clock frequency, F_S .



$$\begin{array}{ll} C_1 = 1.807 \text{ F} & C_2 = 2.513 \text{ F} \\ L_1 = 1.152 \text{ H} & L_2 = 0.828 \text{ H} \\ R_S = 1.000 \Omega & R_1 = 1.000 \Omega \end{array}$$

Figure 5-1: 4th order lowpass reference filter

This reference filter can be transformed into a structure suitable for implementing as a wave filter, using the Kuroda transform method described in Chapter 2. Alternatively, a suitable filter may be directly synthesised from the transfer function given in section 2.7.

If Kuroda transforms are used, three unit-elements of characteristic impedance 1Ω are inserted between L_2 and the load resistance, R_L . After transforming the structure, the resultant filter is as shown in Figure 5-2. The component values are again given in Table 5-1. It should be noted that no unit-element is required between capacitor C_4 and the load resistance as there is no reflected wave from a pure resistance.

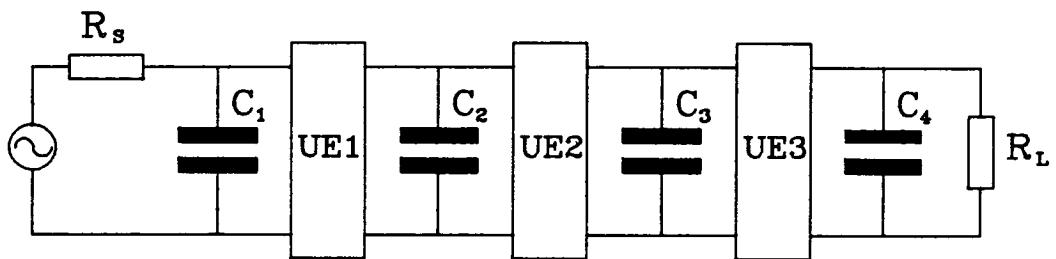


Figure 5-2: 4th order lowpass filter after Kuroda transforms

The major drawback of using Kuroda transforms to produce suitable reference filters, is that the unit-elements do not contribute to the overall response of the filter. However, in chapter 2, it was shown that reference filters could be generated using synthesis techniques in which unit-elements could be used to increase the order of the filter.

The reference filter of Figure 5-2 was synthesised using the transfer function given in section 2.7.2. The synthesis program SYNTH [23], was used to generate a

reference filter with an identical topology to that of Figure 5-2. The filter was to have similar passband ripple and cutoff frequency as that of the structure generated using Kuroda transforms. The sequence of commands to run the program is given below.

RUN SYNT

| | |
|--|----------|
| Enter the number of UNIT ELEMENTS, N: | 3 |
| Enter the number of ZEROS AT NYQUIST, L: | 4 |
| Enter the number of ZEROS AT DC, K: | 0 |
| Enter the lower cutoff frequency as % of Fc: | 0 |
| Enter the upper cutoff frequency as % of Fc: | 20 |
| Enter the value of epsilon: | 0.400 |
| Enter the number of significant figures: | 30 |
| Enter the name of the output file: | LPF4.DAT |

The program SYNT generates an input impedance function, Z_{in} , from which elements may be extracted in the conventional way [21] to produce the circuit topology. A further program, XTRACT [23], can be used to perform the element extraction. As stated previously, SYNT will generally be able to produce an impedance function corresponding to the required filter parameters, but no guarantee can be given that XTRACT will generate a particular filter topology. Further details on the synthesis and element extraction programs are given in Appendix I.

As the filter order increases, the accuracy required

to determine the exact position of the zeros of transmission in the s -plane also increases. To ensure that the required transfer function is generated correctly, high precision arithmetic is used in the synthesis program. For the synthesised filter, SYNTH was run with 30 decimal places of accuracy. The total CPU time for the synthesis of the filter was 1 minute 5 seconds. When the element extraction program was run, it was found that an identical filter topology to that of Figure 5-2 was possible. However, the component values of the filtering elements were found to be different. A comparison of the component values for the fourth and seventh order filter is given in Table 5-1.

| Element name | Kuroda transform | Synthesised filter |
|--------------|------------------|--------------------|
| RS | 1.000 Ω | 1.000 Ω |
| C1 | 2.487 F | 2.570 F |
| C2 | 3.220 F | 3.777 F |
| C3 | 0.402 F | 3.777 F |
| C4 | 0.093 F | 2.570 F |
| UE1 | 1.851 Ω | 1.891 Ω |
| UE2 | 4.680 Ω | 1.986 Ω |
| UE3 | 2.739 Ω | 1.891 Ω |
| RL | 2.182 Ω | 1.000 Ω |

Table 5-1: Component values for 4th & 7th order filters

The lowpass reference filter of Figure 5-2 can be realised as a wave filter implemented using universal adaptors as shown in Figure 5-3. If the reference structure of (a) is transformed to a wave filter in the con-

ventional manner, then four parallel adaptors are required, as shown in (b). However, by implementing the wave filter using universal adaptors, (c), only two adaptor structures are required.

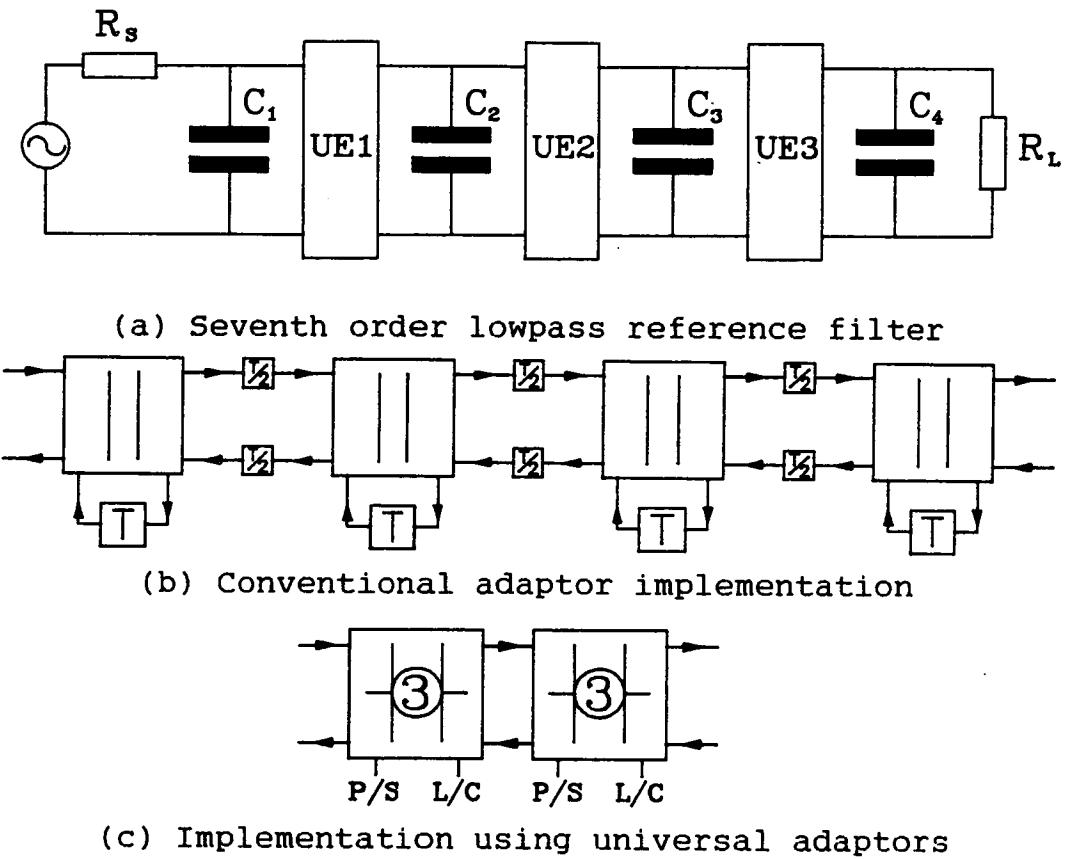


Figure 5-3 : Implementation of 7th-order lowpass filter using conventional and universal adaptors

External control of the filter is greatly simplified because all the adaptors to be implemented are parallel, and all port 3 connections are capacitors.

For the example filter, eight separate multiplier coefficients are required, as given in Table 5-2. The

table shows the α_{ij} coefficients for multiplier j of adaptor i , for both the fourth and the seventh order filters. For the universal adaptor implementation, these coefficients are also evaluated as $\alpha - 1$ for each parallel adaptor. The coefficients are used by the universal adaptor in the same order as they would have been used in the filter of Figure 5-3 (b).

| Coefficient | 4th order | 7th order | Universal Adaptor |
|---------------|-----------|-----------|-------------------|
| α_{11} | 0.4966107 | 0.4879371 | -0.5120629 |
| α_{12} | 0.2682276 | 0.2579995 | -0.7420005 |
| α_{21} | 0.2718462 | 0.2198710 | -0.7801290 |
| α_{22} | 0.1075471 | 0.2093644 | -0.7906356 |
| α_{31} | 0.4357325 | 0.2093644 | -0.7906356 |
| α_{32} | 0.7444813 | 0.2198710 | -0.7801290 |
| α_{41} | 0.7964836 | 0.2579995 | -0.7420005 |
| α_{42} | 1.0000000 | 0.4879371 | -0.5120629 |

Table 5-2: Coefficient values for reference filter

The synthesised reference filter was generated from an input impedance function with four zeros of transmission at the Nyquist frequency, and three unit-elements. In Figure 5-3 (a), this impedance function has been realised by four parallel capacitors, separated by three unit-elements. However, the element extraction process can remove zeros of transmission as parallel capacitors, or as series inductors. In this situation, the element values remain unchanged, and the filter response is unaltered. It is therefore possible to interchange parallel capacitors and series inductors in an effort to simplify the design of the system. If series inductors are used

in place of the parallel capacitors, then the multiplier coefficients of Table 5-2 would be realised in the form of $1 - \beta_{ij}$ for the universal adaptor. The magnitude of the multiplier coefficient β_{ij} would be the same as that for α_{ij} . The order in which the multiplier coefficients are required is the same for the parallel or series configuration.

5.3 Signal and Control Information

5.3.1 Signal information

In Section 4.3.2, it was shown how the system wordlength, and hence the filter sample rate was determined by the total latency of the universal adaptor. The latency of each adaptor system is

$$L_{ad} = 3/2 * m + 10 \text{ cycles} \quad (5.1)$$

where m was the multiplier coefficient wordlength. This latency corresponds to the $T/2$ delay required to compute the internal calculations of the adaptor. As the system hardware is used twice in each sample period, the minimum signal wordlength will correspond directly to the latency of the system.

Within the data wordlength, provision must be made for signal growth through the adaptor system. For the universal adaptor, six guard bits are required for the worst-case condition. These guard bits appear at the MSB

end of the binary word, and prevent arithmetic overflow which may cause instability. The requirement for these guard bits means that the maximum signal level available in the filter system at any time will be

$$\begin{aligned} \text{maximumsignal} &= \text{system wordlength} - \text{guardbits} \quad (5.2) \\ &= 3/2 * m + 10 - 6 \text{ bits} \\ &= 3/2 * m + 4 \text{ bits} \end{aligned}$$

The relationship between system wordlength, maximum signal level and multiplier coefficient wordlength is summarised in Table 5-3.

| Multiplier size | System wordlength | Maximum signal |
|-----------------|-------------------|----------------|
| 4 | 16 | $2^{11} - 1$ |
| 6 | 19 | $2^{14} - 1$ |
| 8 | 22 | $2^{17} - 1$ |
| 10 | 25 | $2^{20} - 1$ |
| 12 | 28 | $2^{23} - 1$ |

Table 5-3: Comparison of multiplier coefficients, signal wordlengths and levels

All signals in the adaptor are in 2's complement, bit-serial form, with the signal entering the system LSB first. The modified Booth multipliers used in the adaptor system use coefficients in the range $-1 \leq m < 1$. However, within the FIRST simulator, these numbers are represented by integers in the range between

$$-2^{k-1} \leq m < 2^{-1} \quad (5.3)$$

where k is the number of coefficient bits available in

the multiplier. To represent these numbers correctly within the adaptor, the multiplier coefficients must have the same overall wordlength as the data, although there will still only be k significant bits. This is achieved by repeating the MSB of the actual data word throughout the remainder of the system word. If the coefficient lies in the range $0 \leq m < 2^{k-1}$, then the signal word will be padded with zeros to equal the system wordlength. If, however, the coefficient lies in the range $-2^{k-1} < m < 0$, then the signal word will be padded with ones.

As an example of this technique, consider the representation of two coefficients, m_1 and m_2 , where an eight bit multiplier is used. Let $m_1 = 0.375$ and $m_2 = -0.375$. The eight bit multiplier can operate in the range

$$-2^7 \leq m_x \leq 2^7$$

The integer representation of the coefficients will therefore be

$$m'_1 = m_1 * 2^7 = +48$$

$$m'_2 = m_2 * 2^7 = -48$$

From Table 5-3, we know that the system wordlength corresponding to an adaptor with an eight bit multiplier, is 22 cycles. Therefore, after extending the MSB of the coefficients, m'_1 and m'_2 , we find that m'_1 remains unchanged such that $m''_1 = m'_1 = 48$. However, m_2 is

negative, and, if the MSB is repeated to fill the system wordlength, then m'_2 is realised as

$$\begin{aligned}m'_2 &= 2^{SWL} + m'_2 \\&= 2^{22} - 48 \\&= 4194256\end{aligned}\tag{5.4}$$

In the universal adaptor system, two independent coefficients are required for each multiplier in every sample period, T. The detailed timing diagram of Figure 5-4 shows how within the adaptors, multiplier A uses coefficients m_{11} followed by m_{21} , and multiplier B uses m_{12} followed by m_{22} for each pair of coefficients. Similarly, multiplier C uses m_{13} and m_{14} , whilst multiplier D uses m_{23} and m_{24} .

5.3.2 Control information

The internal control of the universal adaptor is controlled by four signals.

1. LSB control signal.
2. Parallel or series connection.
3. Inductive or capacitive termination.
4. Sample or recycle data control.

The LSB control signal is required by all the arithmetic units within the system, in order to reset each

operator for every new data word entering the adaptor. The PARALLEL/SERIES signal is used to control the flow of data through the adaptor, in order to satisfy the equations required for parallel or series adaptors. This signal proceeds through the adaptor in synchronism with the LSB signal, and is required whether the adaptor is accepting new data or recycling data from the previous time period. The INDUCTOR/CAPACITOR signal determines the termination that is present at port 3 of the adaptor. This signal is also synchronised with the LSB signal. Finally, the SAMPLE/RECYCLE signal is used to control whether the adaptor is accepting data from some external source, or recycling data from the output generated in the previous T/2 time period. Again, this signal is synchronised with the LSB signal.

For the implementation of the example filter, the control requirements are simplified because all the external connections are parallel capacitors. Therefore, the PARALLEL/SERIES control input can be connected to GROUND, and the INDUCTOR/CAPACITOR input connected to VDD. The SAMPLE/RECYCLE control input is driven by a square wave of period T, as each adaptor alternates between accepting new data and recycling the previous output. Finally, the LSB signal is used to synchronise all the signals going into, and coming out of the system. The control signals for the example filter realisation

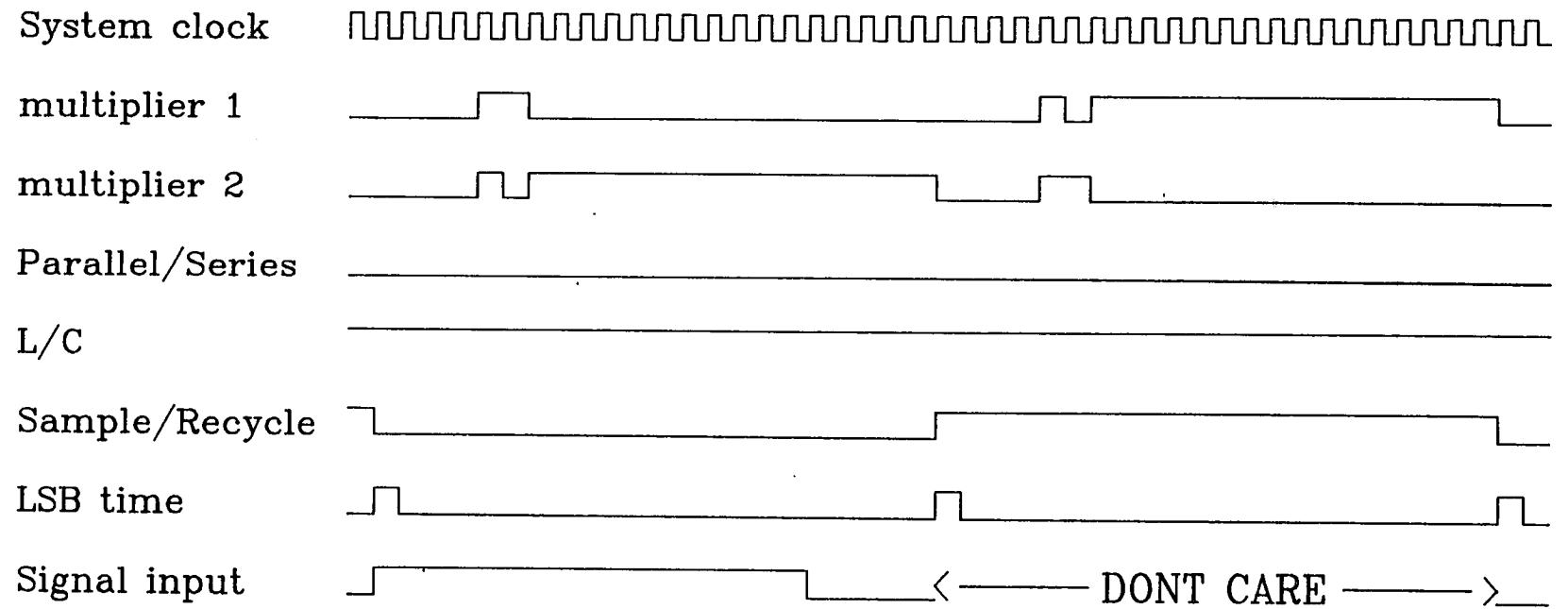


Figure 5-4: Signal & control timing for example filter implementation

are given in Figure 5-4.

In this figure, the multiplier coefficient wordlength is 8 bits, giving a system wordlength of 22 bits. The signal stimulating the filter is an impulse of magnitude $2^{17} - 1$, corresponding to the maximum signal permissible in the system. This signal is used to determine the impulse, or natural, response of the filter, from which the frequency response can be evaluated using Fourier analysis.

5.4 Simulation Results

In this section, simulation results are given for a universal adaptor implementation of the lowpass filter described in section 5.2. The wave filter has been implemented using the FIRST compiler system, and simulated using the FIRST simulator. This models the circuit in the time domain, and provides both a functional verification and ensures that the correct synchronising delays have been included in the circuit. A range of universal adaptors have been used, with multiplier coefficient wordlengths of between two and twelve bits. Each filter system was excited by an impulse corresponding to the maximum signal permissible in the particular system. The exception to this was the simulation of a wave filter using multiplier coefficients of only 2 bits. This case will be considered separately in section 5.4.6. The

impulse response of each circuit was recorded and the time-domain results converted to the frequency domain using a 256-point Discrete Fourier Transform (DFT).

In the simulation of the filters, 1024 sets of adaptor equations were evaluated, corresponding to 512 filter sample periods. Although only 256 time domain results are required by the DFT process, details of the parasitic oscillations generated by the adaptors can be determined by extending the simulation period.

In the simulation of these filters, a comparison is required between the theoretical response and the response generated by the wave filter. Ideally, these responses will be indistinguishable, and the wave filter will exactly match the reference filter in both amplitude and phase response. However, as the accuracy of the multiplier coefficients is reduced, the response of the wave filter should start to deviate from that of the reference filter. The theoretical response is constrained by both the cutoff frequency, and the passband ripple. In order to satisfy these constraints, the wave filter response should not deviate in either of these limitations. However, as will be seen later, both the cutoff frequency and the magnitude of the passband ripple are affected as the multiplier coefficient accuracy is reduced.

5.4.1 12 bit multiplier coefficient

The simulation of a 7th-order wave filter with multiplier coefficients of 12 bits requires a data wordlength of 28 bits. The largest signal permissible in such a system will therefore be

$$28 - 6 \text{ bits}$$

$$= 22 \text{ bits}$$

$$= 2^{23} - 1$$

For the example filter, the multiplier coefficients required can be found using the techniques described in section 5.3.1. For a 12 bit multiplier, these coefficients will be

$$\begin{array}{ll} m_{11} = 268434409 & m_{31} = 268433839 \\ m_{12} = 268433860 & m_{32} = 268433938 \\ m_{21} = 268433938 & m_{41} = 268433860 \\ m_{22} = 268433839 & m_{42} = 268434409 \end{array}$$

The full frequency response and magnified passband response of the example filter, simulated using 12 bit multipliers, is given in Figures 5-5 (a) and (b) respectively.

The frequency responses of Figures 5-5 (a) and (b) show how the practical (solid line) and theoretical (broken line) responses are virtually identical for a filter with multiplier coefficient wordlengths of 12 bits. Both the cutoff frequency and the passband ripple are unaffected, producing an acceptable frequency response.

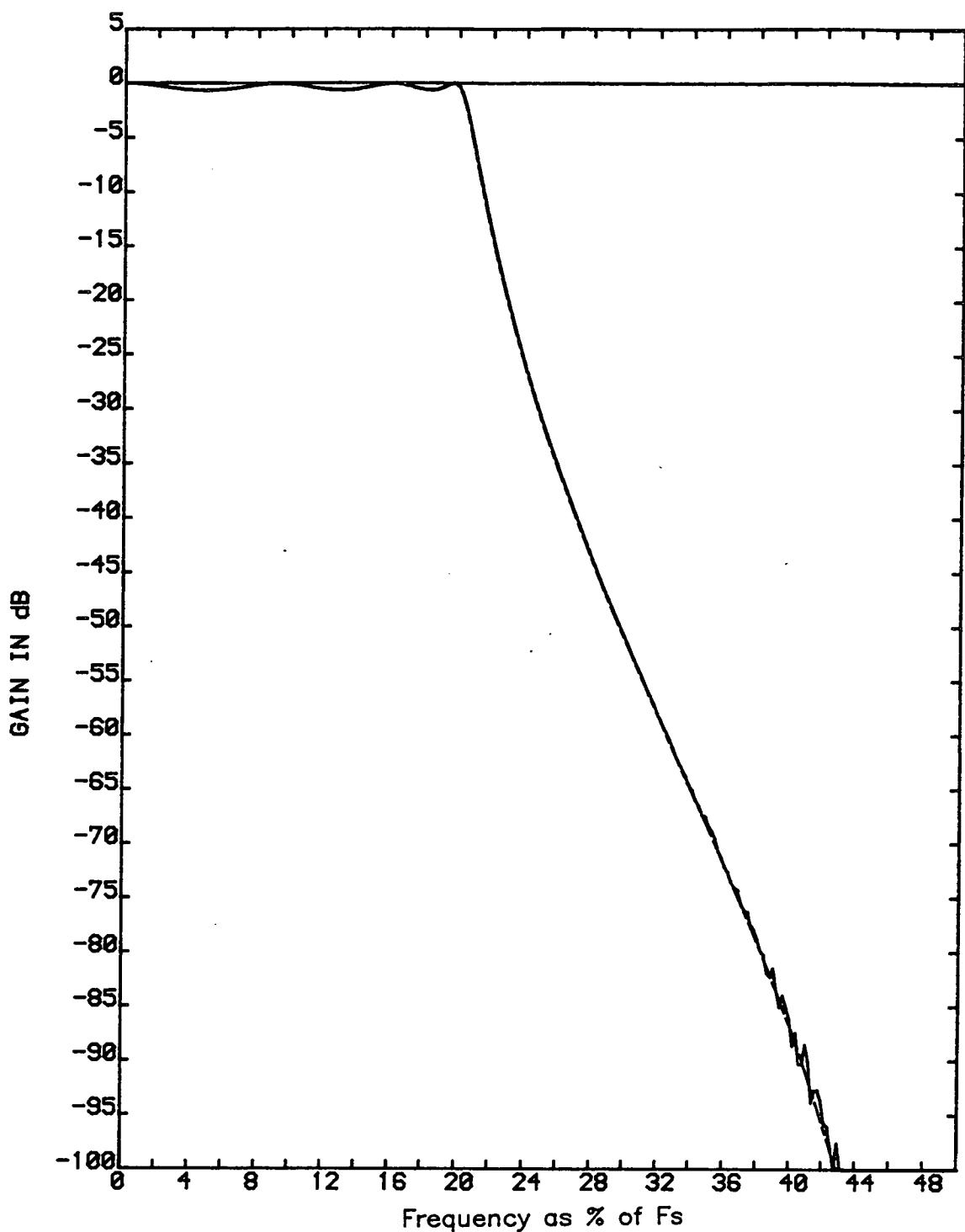


Figure 5-5 : (a) Example filter frequency response
(12 bit multiplier coefficient)

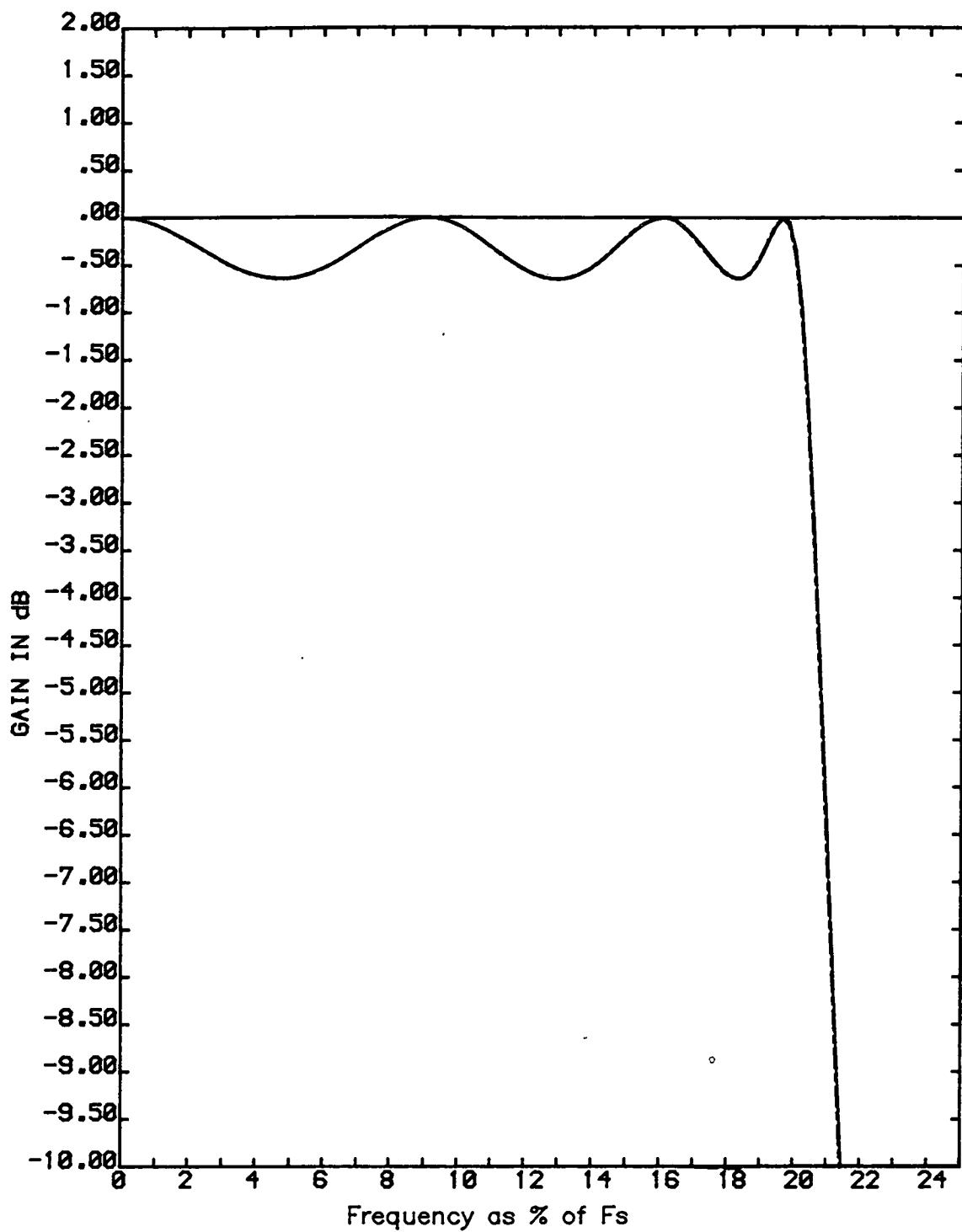


Figure 5-5 : (b) Magnified passband response
(12 bit multiplier coefficient)

5.4.2 10 bit multiplier coefficient

For 10 bit multiplier systems, the maximum signal level will be reduced to

25 - 6 bits

= 19 bits

= $2^{20} - 1$

For this system the multiplier coefficients will be as follows

$$m_{11} = 33554170 \quad m_{31} = 33554028$$

$$m_{12} = 33554033 \quad m_{32} = 33554053$$

$$m_{21} = 33554053 \quad m_{41} = 33554033$$

$$m_{22} = 33554028 \quad m_{42} = 33554170$$

The frequency response and magnified passband response of the example filter simulated with 10 bit multiplier coefficients are given in Figures 5-6 (a) and (b). Once again, the practical response (solid line) agrees closely with the theoretical response (broken line) in both passband ripple and cutoff frequency.

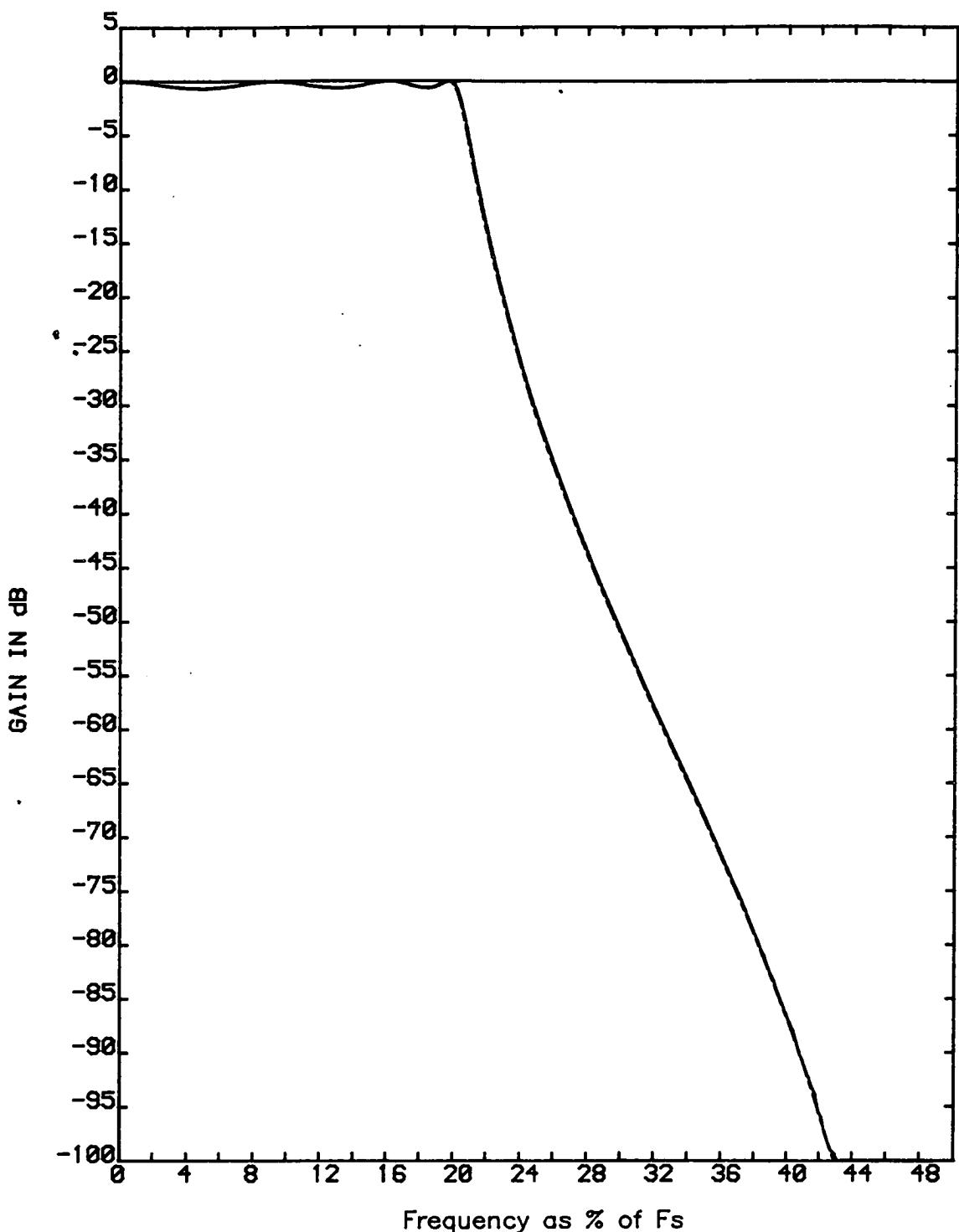


Figure 5-6 : (a) Frequency response of example filter
(10 bit multiplier coefficient)

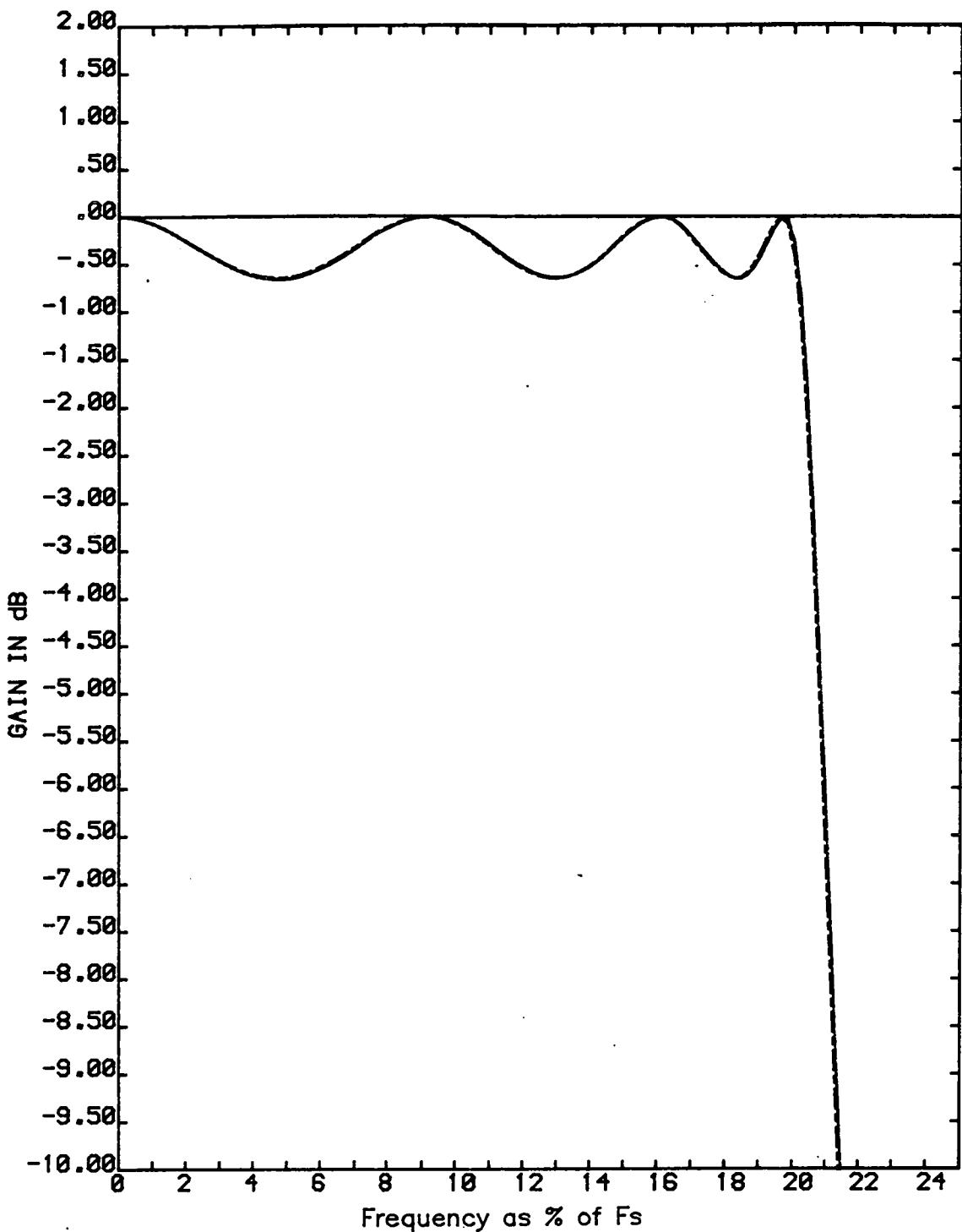


Figure 5-6 : (b) Magnified passband response
(10 bit multiplier coefficient)

5.4.3 8 bit multiplier coefficient

If the multiplier coefficient is further reduced to 8 bits of accuracy, the minimum data wordlength will also reduce to 22 bits. Of these 22 bits, only (22 - 6) bits may be used for valid signals. The multiplier coefficients will also be affected, and will be used as follows

$$m_{11} = 4194239 \quad m_{31} = 4194203$$

$$m_{12} = 4194204 \quad m_{32} = 4194209$$

$$m_{21} = 4194209 \quad m_{41} = 4194204$$

$$m_{22} = 4194203 \quad m_{42} = 4194239$$

The complete frequency response and magnified passband response of the example filter implemented using 8 bits of multiplier coefficient are given in Figures 5-7 (a) and (b) respectively. As before, the theoretical and practical responses are in close agreement, producing an acceptable frequency response. However, the magnified passband response of Figure 5-7 (b) shows how the passband ripple is starting to deviate from the theoretical response as the accuracy of the multiplier coefficients is reduced. The cutoff frequency of the filter remains unaffected by quantisation of the multiplier coefficients to 8 bits.

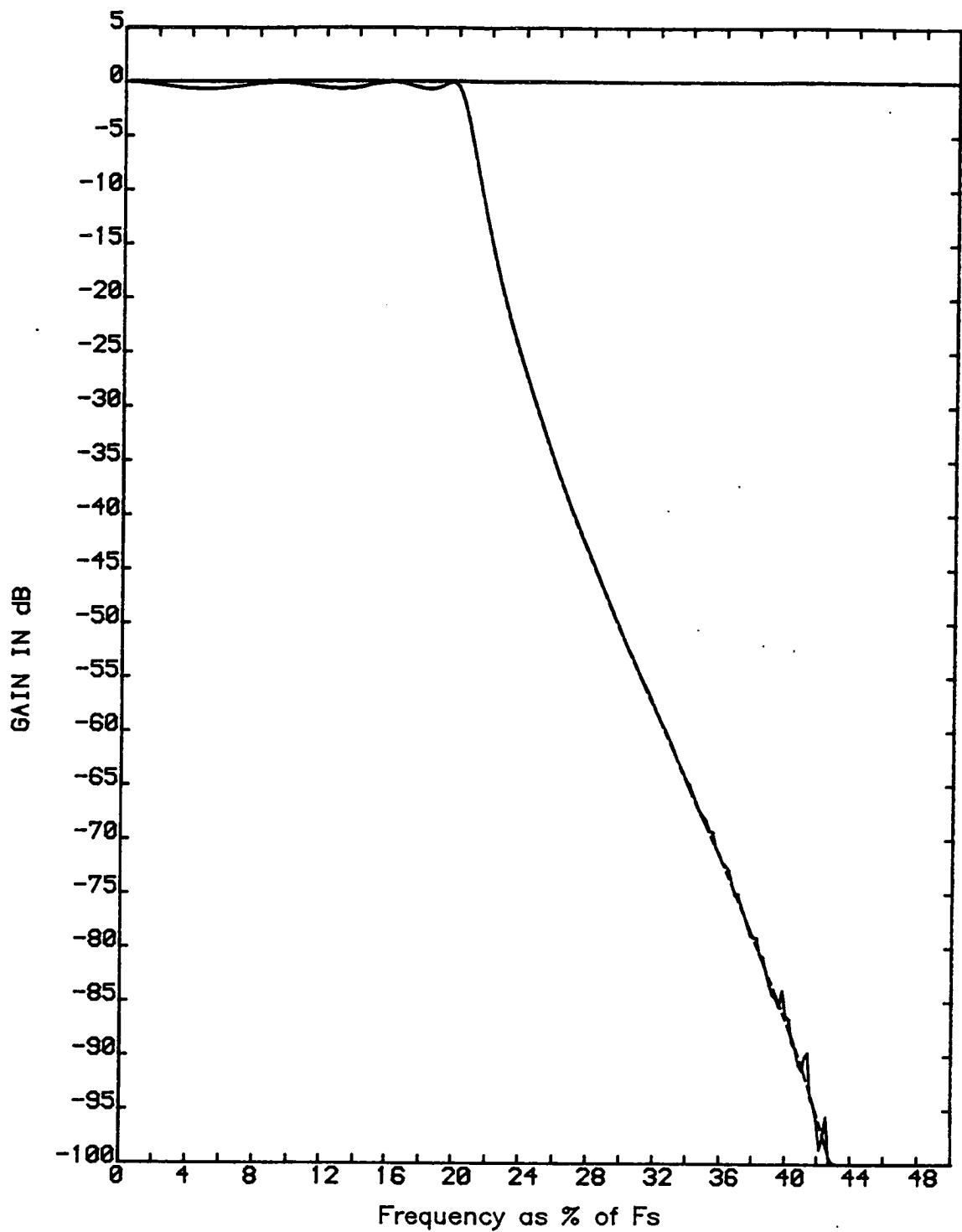


Figure 5-7 : (a) Example filter frequency response
(8 bit multiplier coefficient)

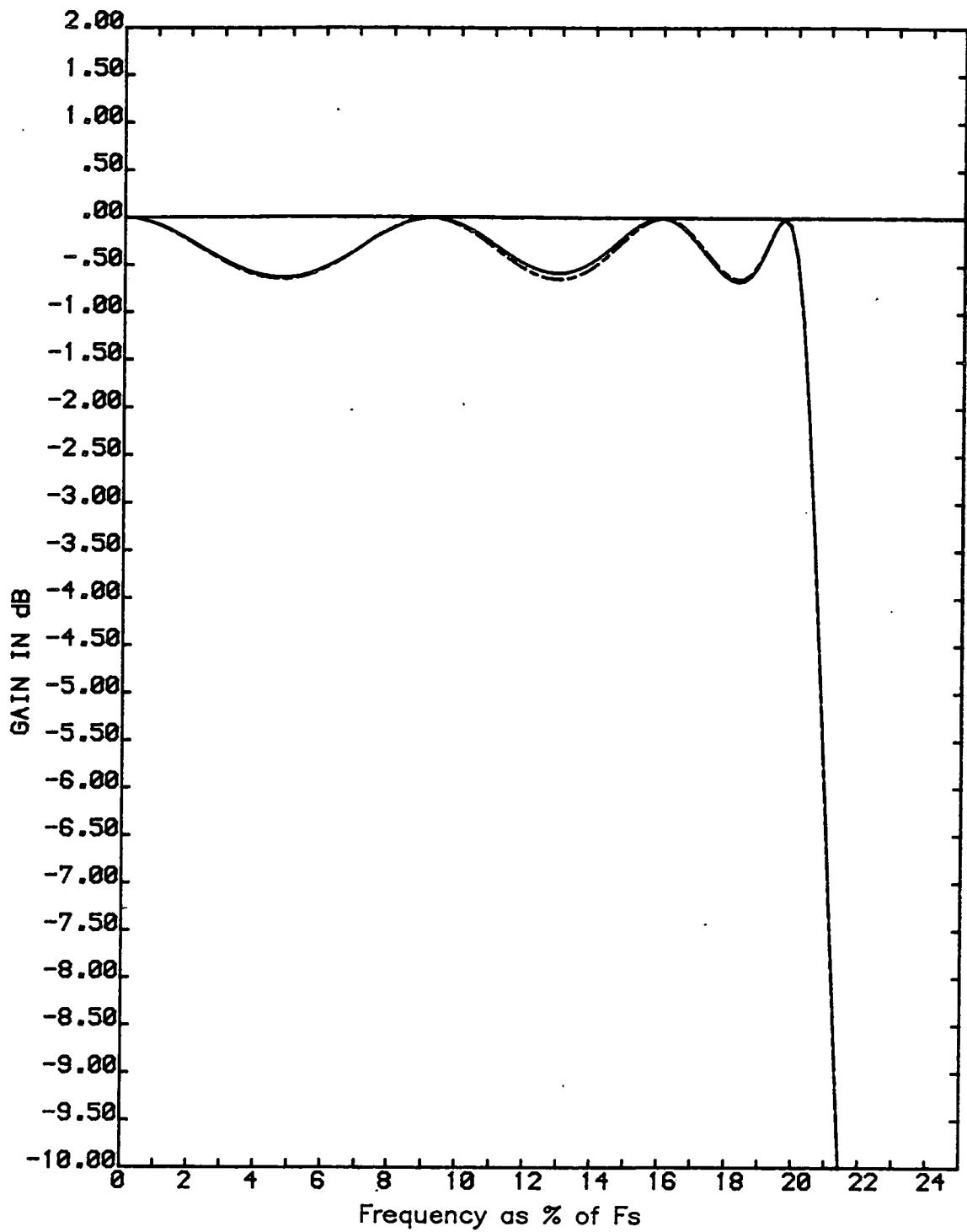


Figure 5-7 : (b) Magnified passband response
(8 bit multiplier coefficient)

5.4.4 6 bit multiplier coefficient

When the multiplier accuracy is reduced to six bits, the data wordlength within the system reduces to 19 bits, of which only 13 bits are allowed to be used to represent the signals. For such a system, the multiplier coefficients can be represented by the following numbers.

$$m_{11} = 524272 \quad m_{31} = 524263$$

$$m_{12} = 524264 \quad m_{32} = 524265$$

$$m_{21} = 524265 \quad m_{41} = 524264$$

$$m_{22} = 524263 \quad m_{42} = 524272$$

Again, the full frequency response and magnified passband response can be seen in Figures 5-8 (a) and (b) respectively. The passband region of the full response given in Figure 5-8 (a) shows how the passband ripple of the filter starts to deviate further from the theoretical response as the coefficient accuracy is again reduced. However, the cutoff frequency of the filter still remains within an acceptable range of the theoretical value.

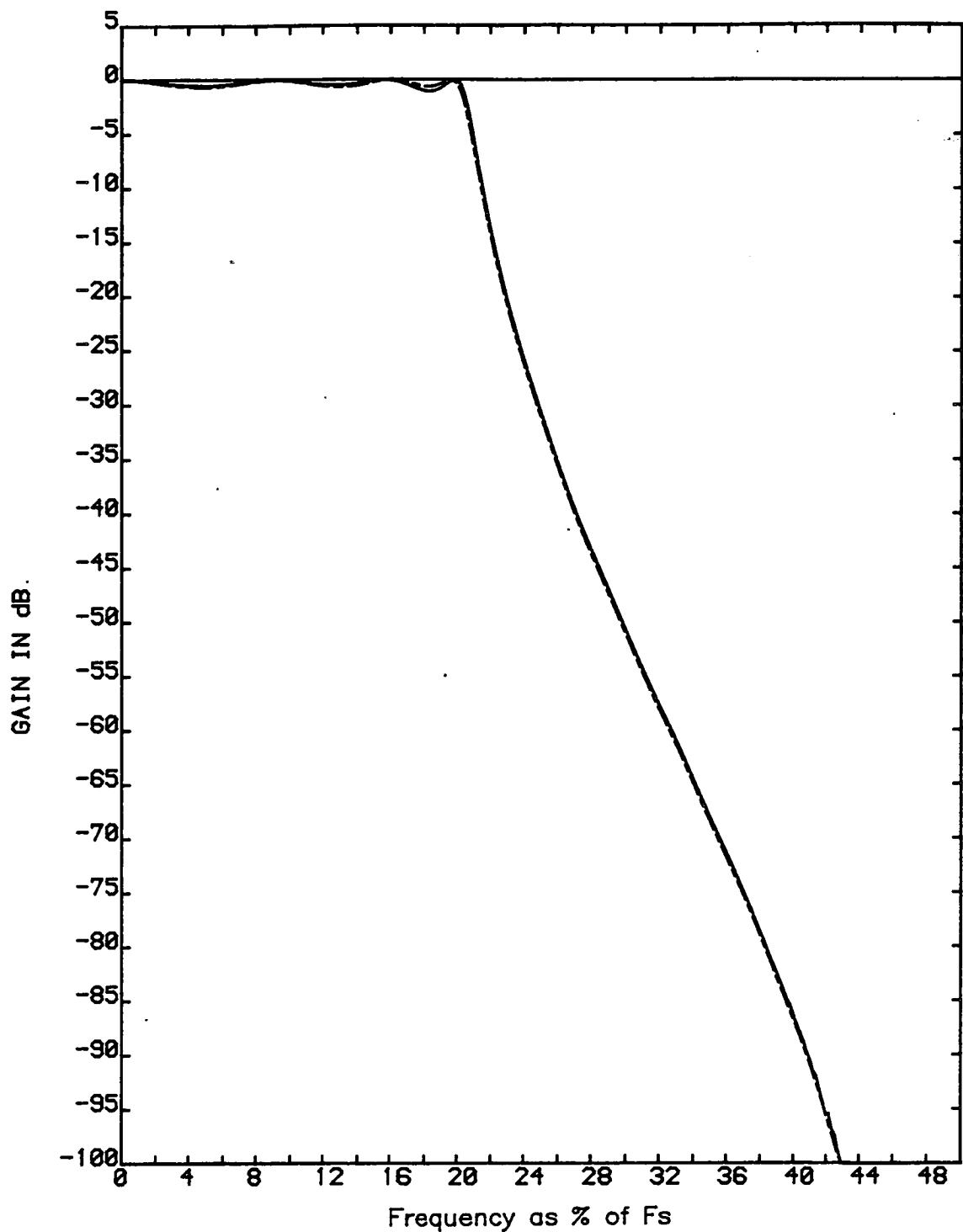


Figure 5-8 : (a) Example filter frequency response
(6 bit multiplier coefficient)

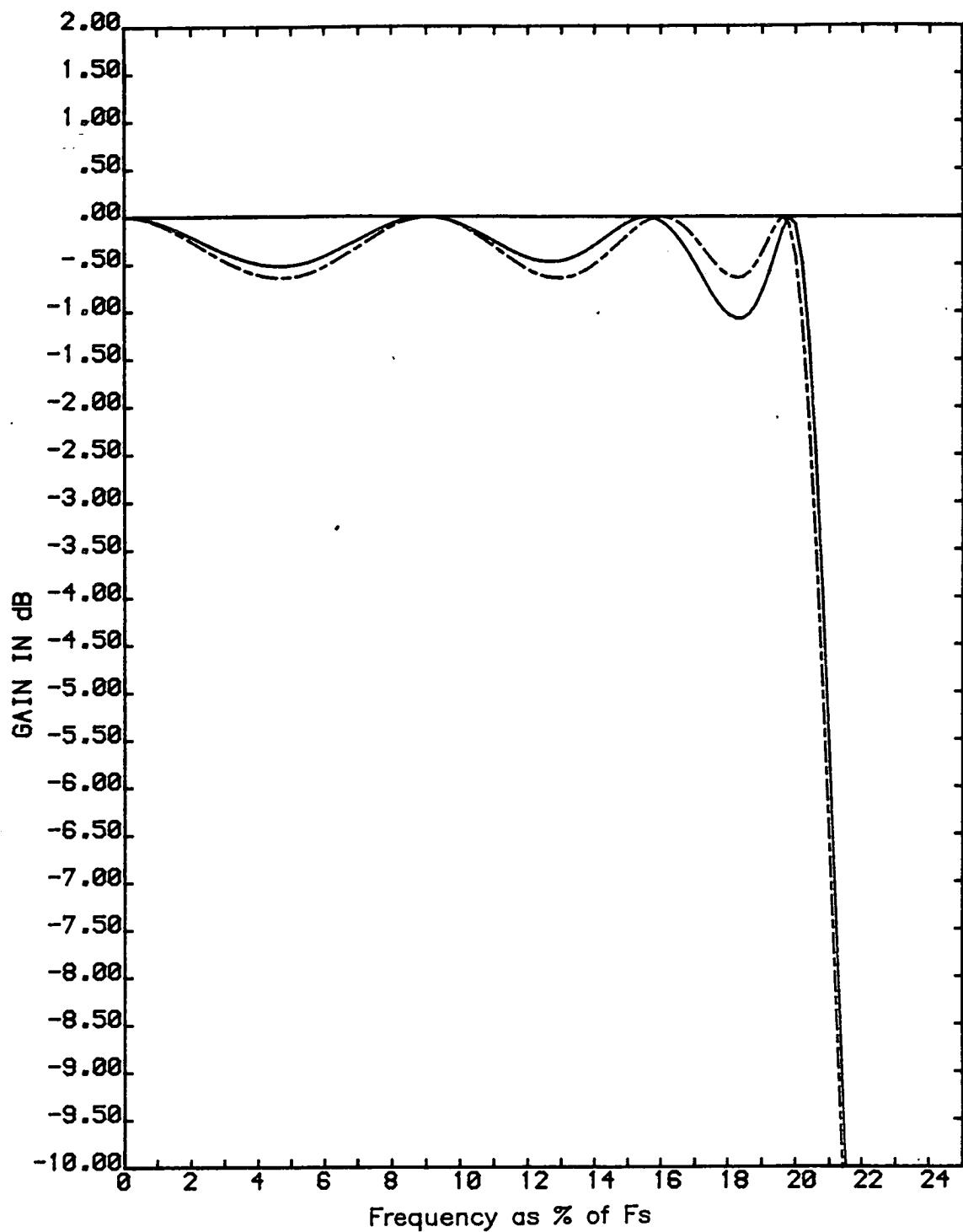


Figure 5-8 : (b) Magnified passband response
(6 bit multiplier coefficient)

5.4.5 4 bit multiplier coefficients

If the multiplier coefficients are reduced to only four bits, then the corresponding system wordlength will decrease in length to 16 bits. Within this wordlength, only 10 bits may be used to represent signals, giving a maximum signal level of $2^{11} - 1$. For such a system, the multiplier coefficients for the example filter will be

$$m_{11} = 65533 \quad m_{31} = 65530$$

$$m_{12} = 65532 \quad m_{32} = 65531$$

$$m_{21} = 65531 \quad m_{41} = 65532$$

$$m_{22} = 65530 \quad m_{42} = 65533$$

The full frequency response and magnified passband response are given in Figures 5-9 (a) and (b) respectively. These figures show how both the passband ripple and cutoff frequency are affected by reducing the coefficient wordlength to only four bits. The filter still has a basic lowpass response, but the passband region is unacceptable due to a large deviation in passband ripple and a drift in the cutoff frequency.

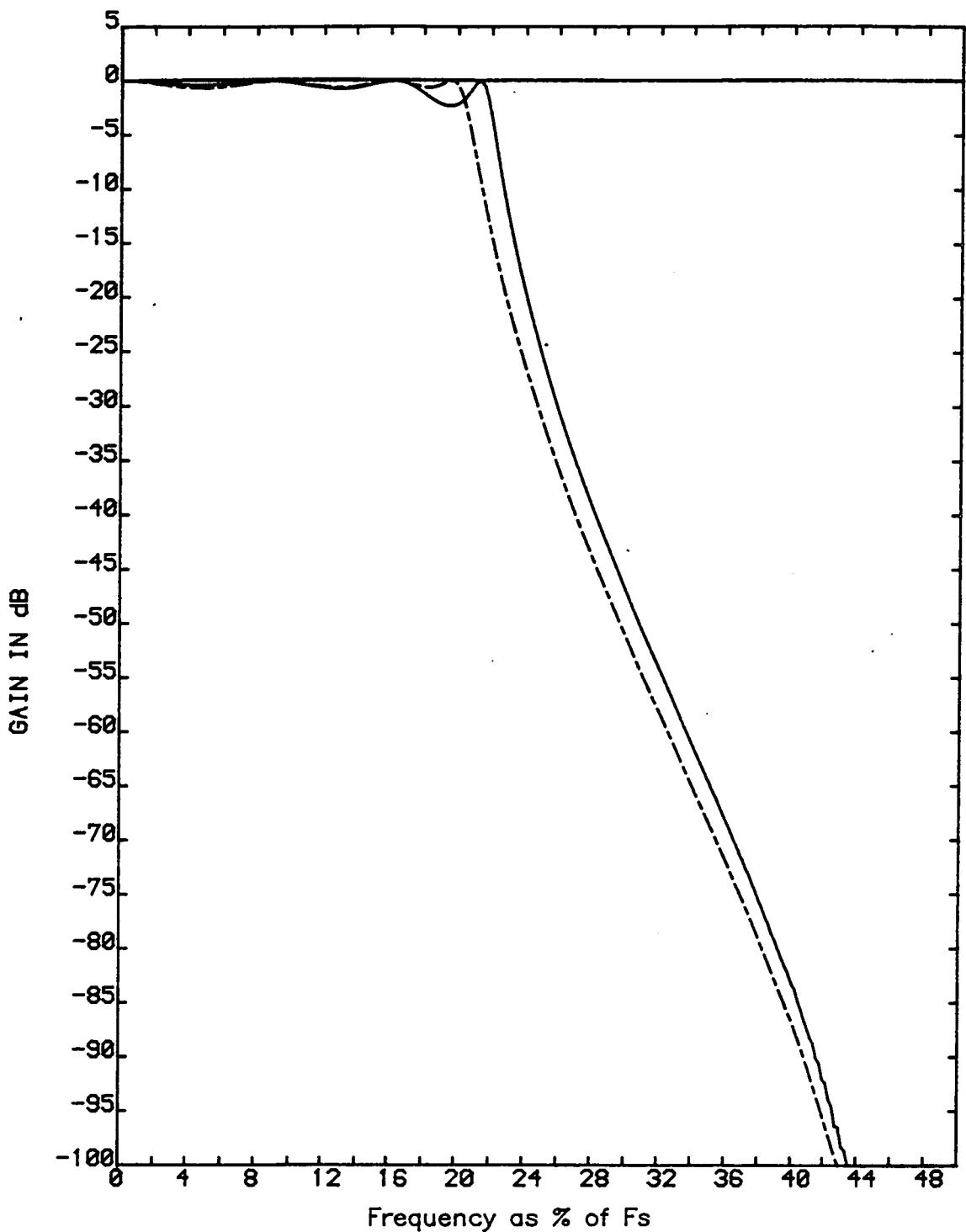


Figure 5-9 : (a) Example filter frequency response
(4 bit multiplier coefficient)

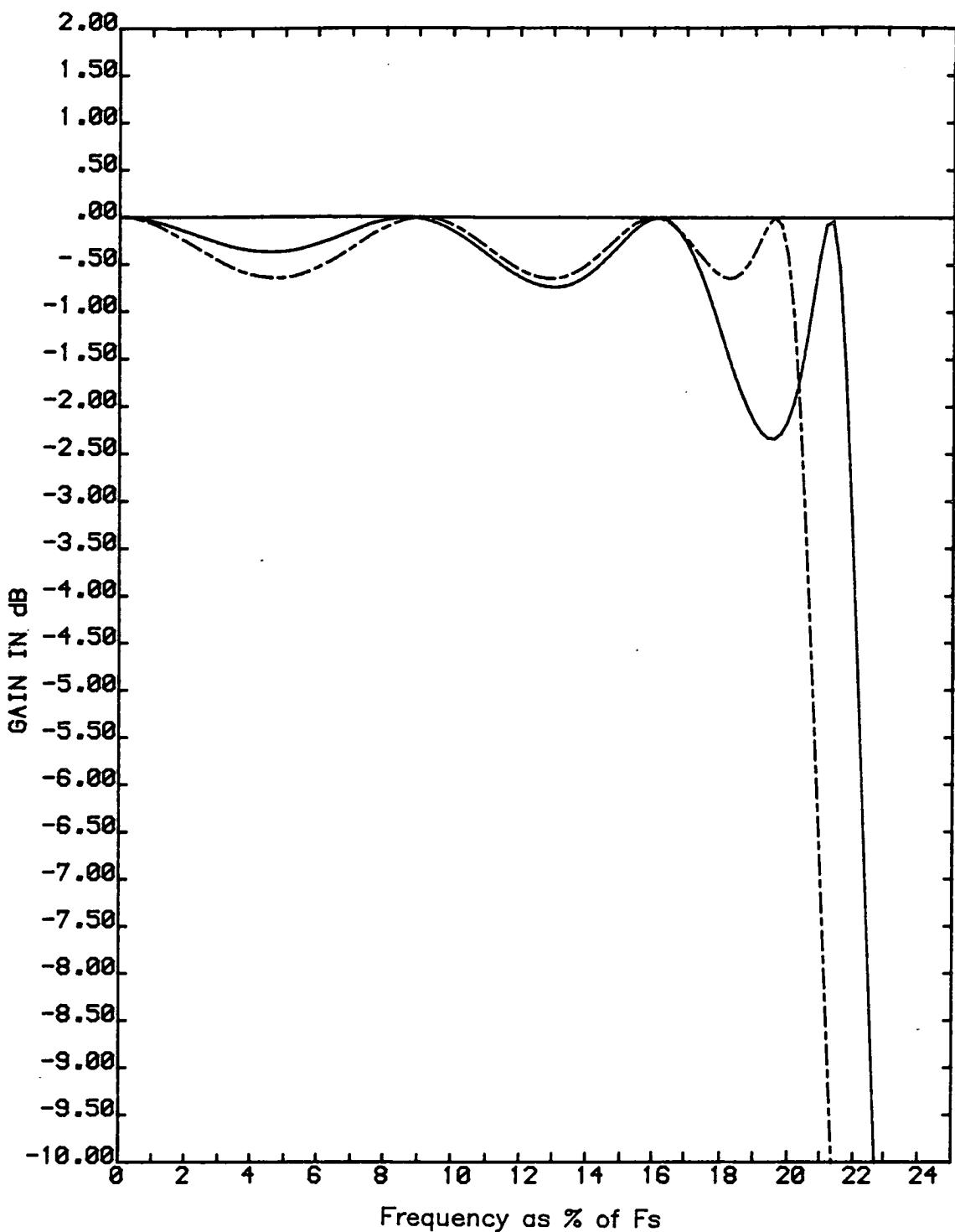


Figure 5-9 : (b) Magnified passband response
(4 bit multiplier coefficient)

5.4.6 2 bit multiplier coefficient

For a wave filter realised using multiplier coefficients of only 2 bits, the data wordlength will be only 13 bits. The largest signal that can exist within such a system will therefore be

$$\begin{aligned} & (13 - 6) \text{ bits} \\ & = 2^8 - 1 \\ & = 255 \end{aligned}$$

Unfortunately, no accurate frequency response could be generated from this range of numbers, so the filter was simulated using a 6-bit multiplier system, but with the coefficients being represented by only the 2 most significant bits in the 6-bit word. The remaining 4 bits of the word are set to zero. This increased the available number range to $2^{14} - 1$, which proved sufficient to attain a suitable frequency response. The same effect could be achieved by adding shift register stages to the inputs or outputs of the adaptor to increase the data wordlength.

For the modified 2-bit system, the multiplier coefficients were

| | |
|-------------------|-------------------|
| $m_{11} = 524240$ | $m_{31} = 524256$ |
| $m_{12} = 524240$ | $m_{32} = 524256$ |
| $m_{21} = 524256$ | $m_{41} = 524240$ |
| $m_{22} = 524256$ | $m_{42} = 524240$ |

Again, the full frequency response and magnified passband response are given in Figures 5-10 (a) and (b) respectively. These figures show how reduction of the coefficient wordlength to two bits produces a completely unacceptable response. The cutoff frequency has drifted substantially from the desired value, and the error in the passband ripple is over 4 dB as the filter approaches the cutoff frequency.

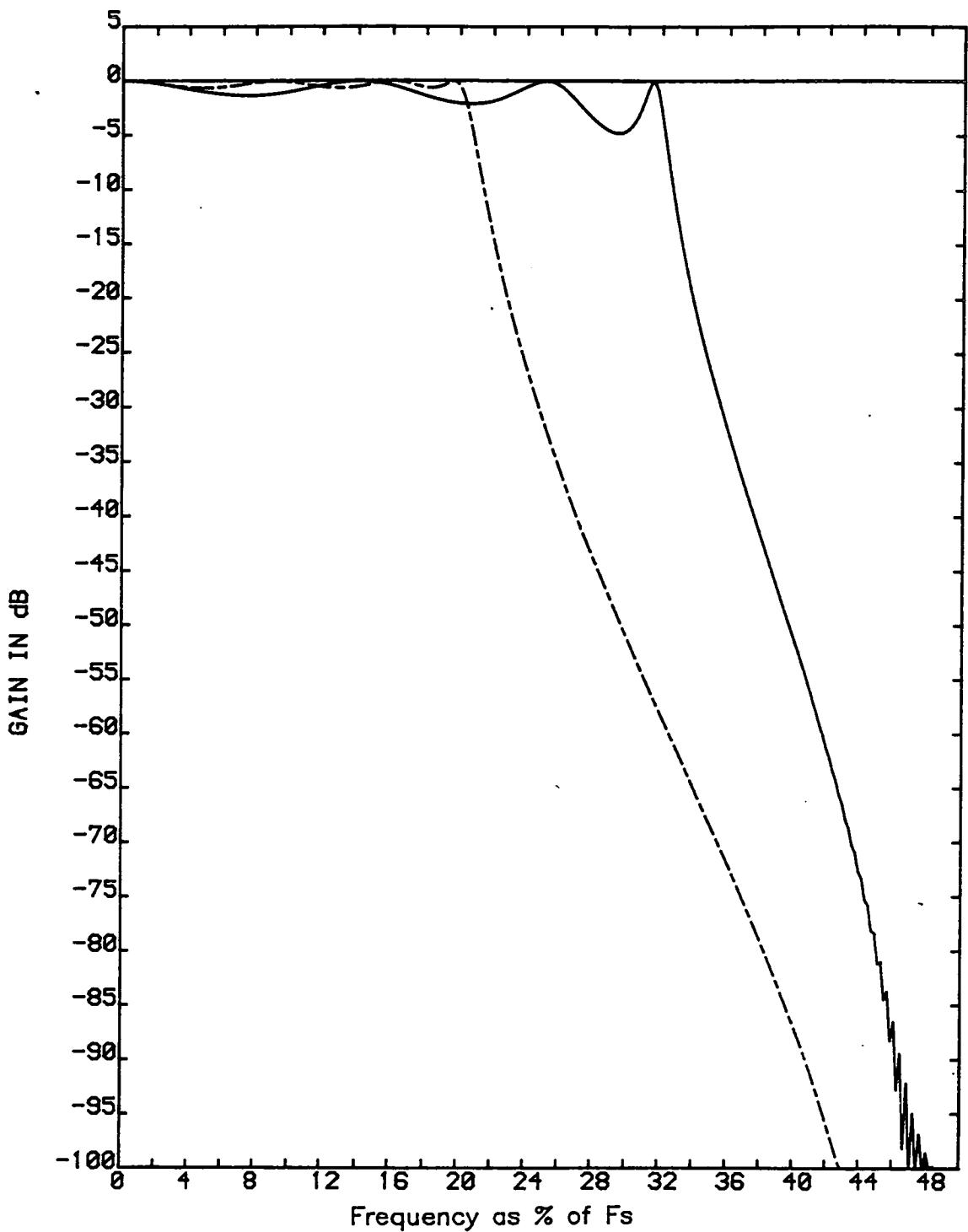


Figure 5-10 : (a) Example filter frequency response
(2 bit multiplier coefficient)

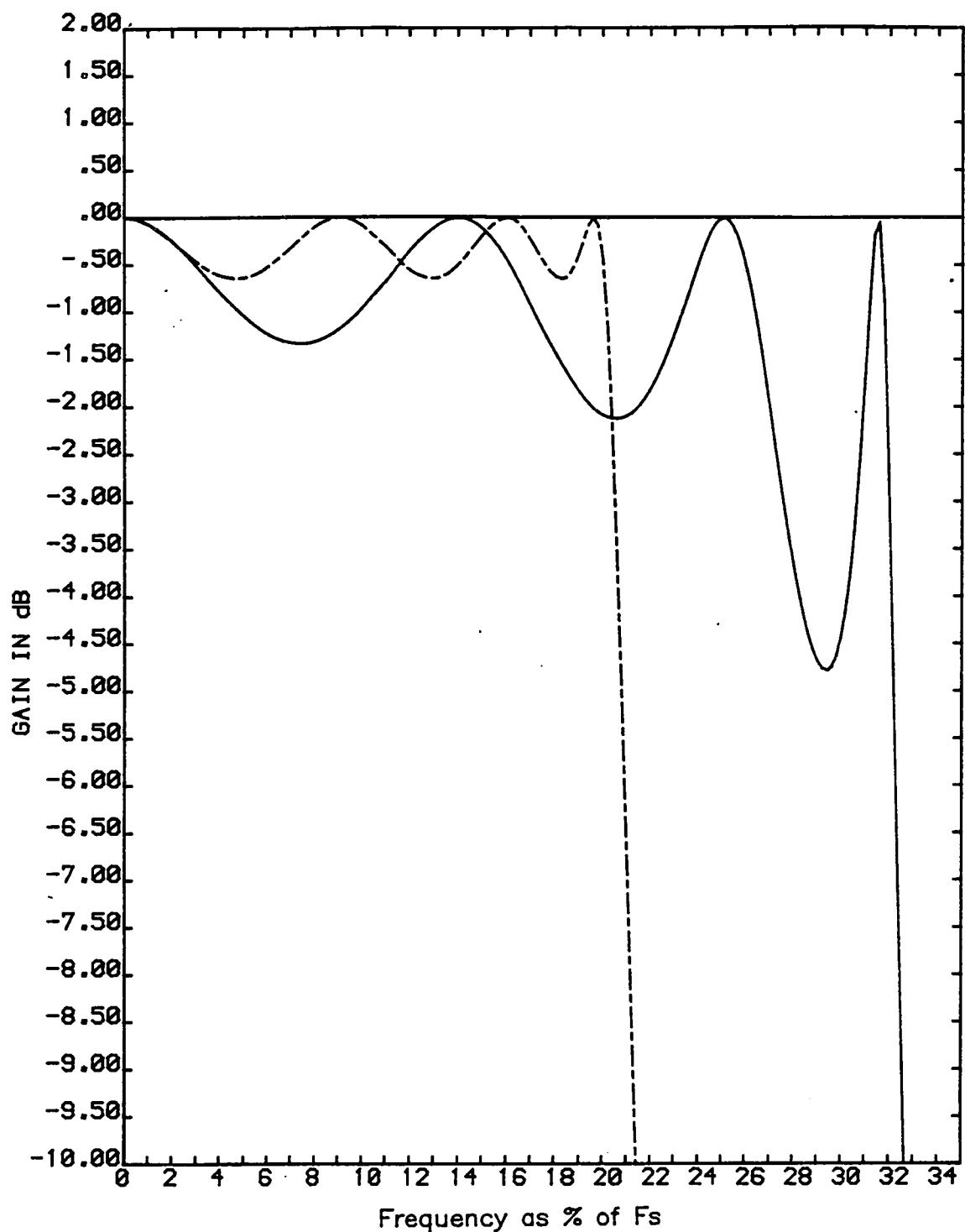


Figure 5-10 : (b) Magnified passband response
(2 bit multiplier coefficient)

5.5 Insertion Loss, Signal Scaling and Suppression of Parasitic Oscillations

5.5.1 Insertion loss in wave filters

In continuous time reference filters, such as the lowpass filter of Figure 5-1, there is an associated insertion loss at points of maximum power transfer. This loss is determined by the resistive terminations of the filter, and corresponds to a 6dB loss in the voltage transfer function when the source resistance is equal to the load resistance. However, in the realisation of wave filters, this loss is eliminated by the filter hardware to give zero insertion loss.

This feature of wave filters is associated with the final adaptor section of any filter. At this stage, the adaptor outputs a reflected wave corresponding to the filter output, incident on the load resistor, R_L . This incident wave is of the form

$$A = V + IR \quad (5.5)$$

In this case, the port resistance, R , is equivalent to the load resistance, R_L , giving

$$A = V + IR \quad (5.6)$$

$$= V + IR_L \quad (5.7)$$

$$= V + V \quad (5.8)$$

$$= 2V \quad (5.9)$$

which eliminates the insertion loss caused by the resistive terminations.

5.5.2 Signal scaling

In reference [7], a technique was presented for increasing the dynamic range of wave filters by the insertion of ideal transformers between appropriate elements in the continuous-time reference structure. The effect of these transformers is to scale signal levels up or down by suitable values to prevent arithmetic overflow or to avoid parasitic oscillations caused by arithmetic errors in very small signals. This can be achieved by scaling all the inputs to an adaptor by a factor of N and the outputs by a factor of $1/N$. Clearly, this process can be greatly simplified if N is some power of 2.

In the universal adaptor realisation of wave filters, no facility has been included for the local scaling of signals. There are two major reasons for this. The first reason is that universal adaptors were originally conceived to be completely general purpose structures and, as such, could not be constrained to offer a signal scaling facility unless scaling by an arbitrary factor could be offered. Clearly this is inefficient in the use of hardware, and the idea was therefore rejected.

The second, but more important reason, is due to an inherent feature of bit-serial systems. Realisations of universal wave filter adaptors using bit-serial arithmetic have, by definition, relatively long data wordlengths. Within this wordlength, guard bits are required at the MSB end of the word to allow for bit-growth in the arithmetic elements. However, after allowing for guard bits, the number of possible signal bits is still generally larger than the maximum number of bits available from A/D and D/A converters. As an example, consider a universal adaptor realised using multiplier coefficient wordlengths of 8 bits. The minimum data wordlength for this adaptor will be 22 bits, with a guard bit requirement of 6 bits (G). There will therefore be 16 available signal bits (D). In a typical filter application, where 12 bit A/D and D/A converters are being used, 4 bits of the signal word will remain unused. This capability can be utilised by shifting the LSB of the output of the A/D converter away from the LSB of the data word. This, in effect, is signal scaling at the input of the filter. Provided the new position of the LSB of the signal word is known at the filter output, the appropriate data bits may be extracted from the output data word. This process, known as *bit-masking*, is shown in Figure 5-11.

This process allows the signal to move up and down

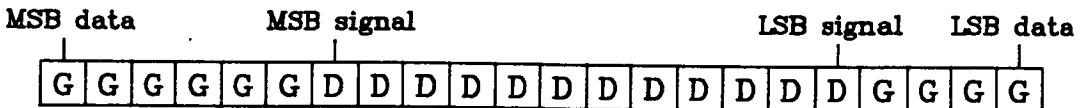


Figure 5-11 : Bit-masking technique in bit-serial arithmetic systems

inside the data word without the need for transformers to be included in the reference filter. As there is no insertion loss associated with wave filters, the signal word will appear in the output data word with the same significance as it when it went into the system. A further benefit of this technique is the automatic suppression of parasitic oscillations.

5.5.3 Suppression of parasitic oscillations.

Techniques for the suppression of parasitic oscillations in wave filters have been presented in references [48] and [49]: These effects are created by signal quantisation and rounding errors in arithmetic operations and appear as small oscillations at the output of digital networks. In practical filter systems, these oscillations can be measured by grounding the input to the filters, and measuring the resulting oscillations at the filter output. For this reason, these effects are often referred to as zero-input oscillations.

The methods presented for the suppression of these oscillations are very similar to the techniques described

above for the scaling of signals in wave filters. The methods proposed for complete suppression involve the use of guard bits at both the MSB end and the LSB end of the data word. Guard bits are required at the MSB end to prevent arithmetic overflow and to accomodate bit growth in the system, and additional bits are required at the LSB end of the data word to allow for rounding errors in the arithmetic elements.

It should be noted at this point that these techniques were initially developed for wave filters realised using parallel arithmetic systems and therefore reduce the dynamic range of systems where the data word size is fixed by the multiplier used in the circuit. In recursive bit-serial systems, data wordlengths are often longer than their parallel equivalent and can therefore offer guard bits at both ends of the data word *without* affecting the dynamic range of the filter.

In the wave filter example of section 5.2, the magnitude of the parasitic oscillations in the system can be determined from the impulse response of the filter. The impulse response of the lowpass filter of section 5.2 will be of the form

$$y(t) = \frac{\sin(t)}{t} \quad (5.10)$$

As wave filters are infinite impulse response filters

(IIR), the output of the filter will never settle to zero even though the input is grounded. The output will continue to oscillate, with a fixed period, about the t axis. In cases where the rounding errors are considerable, such as universal adaptors with only 2 bits of multiplier coefficient, these oscillations may also have a small offset from the t axis. The magnitude of these oscillations can be used to indicate the number of guard bits required at the LSB end of the data word.

For the example filter, with multiplier coefficient wordlengths of 6 and 8 bits, three bits are required at the LSB end of the data word to allow for parasitic oscillations. This requirement reduces to only two for the more accurate 10 and 12 bit multiplier coefficients. This technique is similar in principle to the signal scaling process, and can be implemented using the bit-masking method to select the appropriate signal word from the output data word.

Both techniques described above, for signal scaling and suppression of parasitic oscillations can be achieved in bit-serial systems with no increase in the hardware requirement, and with no reduction in the dynamic range of the filter.

5.6 Summary

In this chapter, a seventh-order lowpass reference filter has been synthesised and modelled as a digital wave filter using the universal adaptors developed in the previous chapter. This structure has been simulated with a range of multiplier coefficients between two and twelve bits, and details of the simulation techniques given. Finally, a method has been developed for the suppression of parasitic oscillations in bit-serial adaptor systems.

The simulation of wave filters using a wide range of multiplier coefficient wordlengths has demonstrated the insensitivity to coefficient variations that these filters possess. The seventh-order wave filter response with coefficients of 12, 10 and 8 bits were virtually indistinguishable from the theoretical response. When the coefficient was reduced to 6 bits, deviations from the theoretical results were noted. Reductions in the coefficient to 4 and 2 bits resulted in unacceptable responses. These simulations have shown that even for high-order structures such as the example filter, 8 bits of multiplier coefficient are sufficient to produce an acceptable response.

Techniques for the suppression of parasitic oscillations have been described, and it has been shown that no additional hardware is required to prevent such effects

in bit-serial wave filters. In addition, these techniques can be used to avoid the use of ideal transformers to maintain the dynamic range of these filters. Again, filters realised using only 8 bits of multiplier coefficient can efficiently suppress parasitic oscillations without affecting the dynamic range. The bit-masking techniques used to achieve this results are well suited to bit-serial arithmetic and offer a further advantage over parallel arithmetic systems.

Chapter 6

NMOS REALISATION OF WAVE FILTER ADAPTOR

6.1 Introduction

In chapter 4, a variety of circuits were presented for the realisation of digital wave filter adaptors. Examples of these circuits include two-port adaptors and series and parallel three-port adaptors. In chapter 4, much emphasis was placed on the development of more versatile adaptor circuits, capable of realising both parallel and series three-port adaptor equations under external control. To achieve this requirement, two circuits were proposed - a general purpose wave filter adaptor and the universal adaptor. The universal adaptor has been presented in detail in chapter 5, where it was used to implement a seventh-order lowpass reference filter. In this chapter, details will be given of an integrated circuit realisation of the general purpose adaptor of section 4.5.

The description of the general purpose adaptor will concentrate on the arithmetic architectures used, rather than on the design techniques required to produce silicon layout which are well documented elsewhere [58,59]. In particular, the key process of binary multiplication, which is central to the adaptor implementation and not

widely reported, will be considered here in some detail.

The integrated general purpose adaptor circuit has been fabricated on a 5 μ m, silicon-gate NMOS process at Edinburgh University, and has been used in the realisation of a practical filter. A third-order lowpass filter has been synthesised using the transfer function given in section 2.7, and has been implemented as a wave filter using the integrated adaptors. The complete filter system and practical results for the filter are also presented in this chapter.

6.2 Binary Multiplication

In the design of any digital wave filter system the method chosen to implement the binary multiplier will generally determine the physical size of the integrated circuit. When any attempt is made to reduce the overall size of the chip, alternative techniques have been developed to replace conventional multiplier architectures by ROM-based distributed arithmetic structures (such as those described in chapter 3). However, these methods have been shown to be unsuitable for applications where programmable multiplier coefficients are required - such as the universal adaptor or the general purpose adaptor. A wide variety of multiplier architectures have been proposed for both word-parallel and bit-serial arithmetic systems. However, only bit-serial multipliers

will be considered here.

In serial multiplier systems, the coefficient wordlength and the data wordlength need not be equal; allowing the adaptor circuits to operate with short coefficient wordlengths, but with data wordlengths of suitable length to accommodate guard bits at both the MSB and LSB end of the word. In general, an n -bit coefficient and an m -bit data word will produce an $n + m$ -bit product. In two's complement fractional binary notation, the bottom n bits of the product represent numbers in the range

$$2^{-m} \leq x \leq 2^{-(n+m-1)}$$

As this is beyond the range of the LSB of the data word, these bits have little significance and the data word may be rounded or truncated to limit the data wordlength to m bits. However, these bits still require to be calculated to ensure arithmetic accuracy, but may be discarded to achieve a multiplier throughput of one product in every m clock cycles.

The simplest bit-serial multiplier algorithm is the "shift-and-add" system which uses a serial data word and a parallel multiplier coefficient. For an n -bit multiplier coefficient and an m bit data word, n adders and $n + m$ clock cycles are required to evaluate the correct, full-precision product. However, both the hardware requirement and the number of system clock cycles can be

reduced by using alternative multiplier algorithms which involve the recoding of either the coefficient or the data word. This recoding takes place within the multiplier system, allowing coefficients to be presented to the multiplier in the conventional format, and generally recodes the multiplier coefficient into either a three-level or five-level code. Three-level codes use add, shift or subtract, (+1, 0, -1), signals to control the flow of data through the adder/subtractor elements. Alternatively, five-level recoding, using add*2, add, shift, subtract or subtract*2, (+2, +1, 0, -1, -2), signals may be employed. Each arithmetic operation is also accompanied by a logical shift.

Of these alternative multiplier systems, only two bit-serial algorithms will be considered - the Canonic Signed Digit (CSD) multiplier and the modified Booth's Algorithm multiplier.

6.2.1 CSD multiplier

The Canonic Signed Digit multiplier [42] is a three-level recoding system in which no two consecutive recoded elements are non-zero. As an example of this recoding method, consider the multiplier coefficient Q0111010. In CSD recoding, any consecutive string of 1's are replaced by a +1 in the bit directly to the left of the string, and a -1 at the lowest bit in the string, i.e.

00111010 → 0+100-10+10

Within this multiplier system, four possible operations can occur.

1. Add nothing, shift right 1 bit
2. Add * 1, shift right 1 bit
3. Subtract nothing, shift right 1 bit
4. Subtract * 1, shift right 1 bit

To realise these conditions for an n -bit multiplier coefficient, a maximum of $n/2$ adder/subtractor operators are required. This is because the number of non-zero coefficient bits is always $\leq n/2$.

A possible circuit realisation for a CSD multiplier with only four bits of multiplier coefficient is presented in Figure 6-1. In this circuit, only two adder/subtractor elements are required. However, a large amount of combinational logic is required to determine which of the four possible arithmetic operations is to be carried out in each particular system clock cycle. This combinational logic operates on pairs of coefficients to ensure the correct coding sequence is achieved. To generate an $n+m$ -bit product using this type of multiplier architecture, $n+m$ system clock cycles will be required. The latency associated with this process is zero.

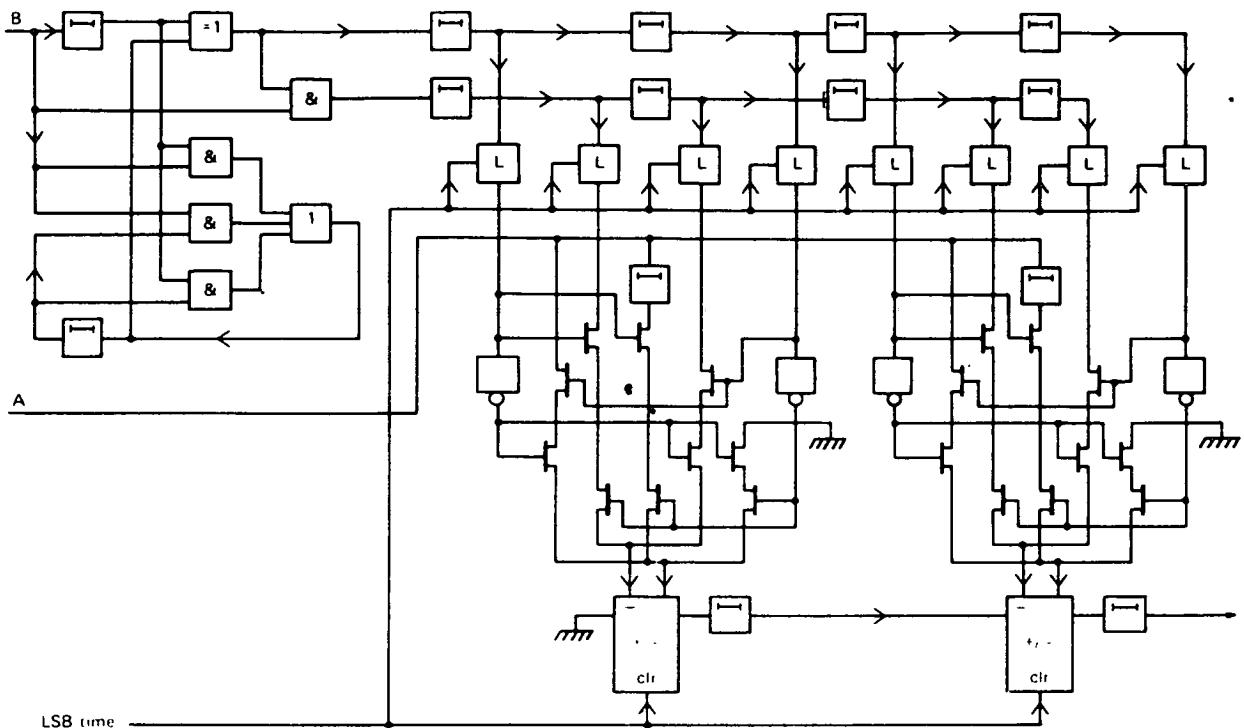


Figure 6-1 : A CSD multiplier architecture
(After Myers & Ivey [60])

6.2.2 Modified Booth's algorithm multiplier

The original binary multiplication algorithm which employed the recoding of either the data word or the coefficient, was the Booth's Algorithm Multiplier [61]. The Booth's algorithm uses a three-level recoding technique, similar to that of the CSD multiplier described above. The major drawback of this particular system is that an n -bit multiplier requires n adder/subtractor elements, similar to the "shift-and-add" system. However, this method has been developed by Rubinfield [57] to produce a five-level recoding system, known as the modified Booth's algorithm.

In the modified Booth's multiplier system, six possible arithmetic operations can occur:

1. Add nothing, shift right 2 bits.
2. Add * 2, shift right 2 bits.
3. Add * 1, shift right 2 bits.
4. Subtract nothing, shift right 2 bits.
5. Subtract * 2, shift right 2 bits.
6. Subtract * 1, shift right 2 bits.

As in the case of the CSD multiplier, combinational logic is used to determine which of these operations is required. Again, coefficients are examined in pairs, and the adder/subtractor elements controlled by the recoding circuitry.

The use of recoding logic also reduces the number of adder/subtractor elements required by the multiplier. For the modified Booth multiplier, $n/2$ elements are required to implement an n -bit coefficient. This multiplier will again produce an $n+m$ -bit product in $n+m$ system clock cycles. As before, the latency of this multiplication process is zero. A possible circuit configuration for such a multiplier is given in Figure 6-2.

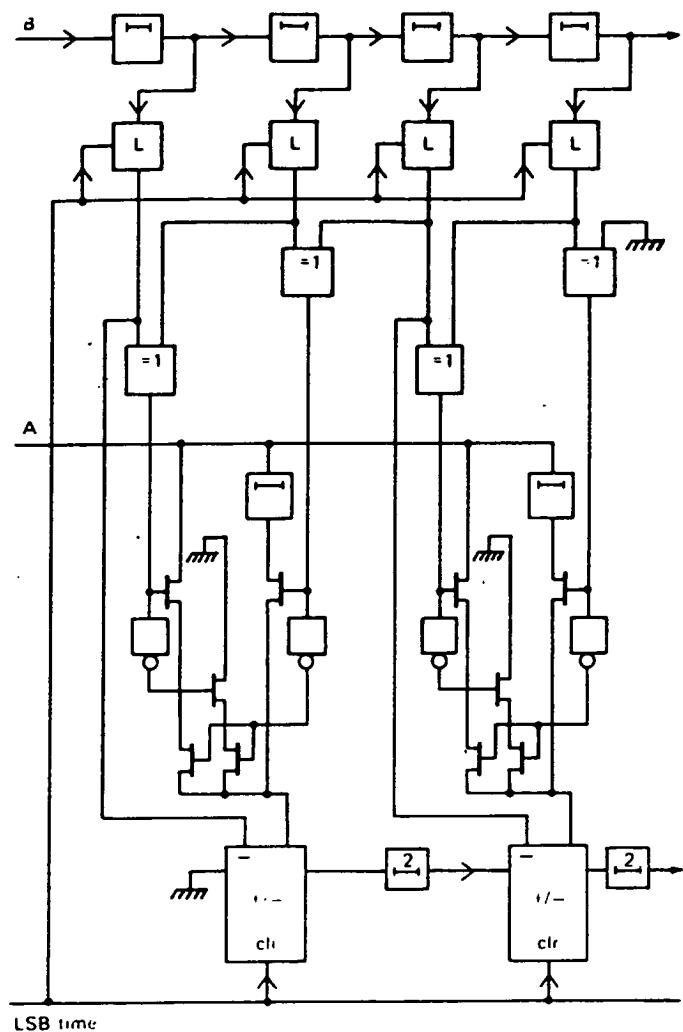


Figure 6-2 : A modified Booth's algorithm multiplier
(After Myers & Ivey [60])

6.2.3 Pipelined multipliers

A direct implementation of the circuits of Figures 6-1 and 6-2 with a coefficient wordlength of n -bits and a data wordlength of m bits will produce an $n+m$ -bit product. However, the data wordlength may be restricted to m -bits with no significant loss of arithmetic accuracy. This can be achieved by simple truncation of the

data word to remove the lower n bits, or rounding and truncation in which a 1 is added to the $m+1$ th bit of the data word to round up the product before truncating to m bits.

Lyon [43] has proposed a series of circuits to generate rounded or truncated multiplier products. These circuits use a "pipelined" system to allow successive data words to enter the multiplier system without affecting the multiplication currently in progress. If this pipelining technique is used in conjunction with either the CSD multiplier or the modified Booth's algorithm, a structure may be developed to produce a rounded, truncated product in only m system clock cycles. This process uses only $n/2$ adder/subtractor elements. The major drawback inherent in such a system is that a latency of $3*n/2 + 2$ cycles is developed within the multiplier system. This latency is generated from two sources. The circuit itself causes a latency of $3*n/2$ cycles, due to the arithmetic elements and the associated internal delays. In addition, one clock cycle is required as the MSB of the data word must be repeated internally in the multiplier architecture to prevent arithmetic overflow. A further clock cycle is required to constrain the multiplier to operate with coefficients in the range $-1 \leq x < 1$.

6.3 Circuit Elements for Integrated Wave Filter Adaptor

In this section, the various circuit elements required for an integrated wave filter adaptor will be described. The adaptor circuit used in this design was the general purpose wave filter adaptor described in section 4.5. This circuit is capable of modelling parallel or series three-port adaptors and has been designed for use in bit-serial arithmetic systems. The circuit diagram corresponding to this adaptor system is given in Figure 6-3.

In the realisation of the general purpose adaptor, four separate circuit elements are required.

1. Multiplexer
2. Bit-delay
3. Adder/subtractor
4. Multiplier

6.3.1 Multiplexer circuit

The simplest circuit element used in the general purpose adaptor is the two-input multiplexer. In this circuit, the multiplexer unit is used merely to switch between two inputs, depending on whether a parallel or series circuit configuration is required. This process can be achieved using two MOS transistors and an inverter, as shown in Figure 6-4.

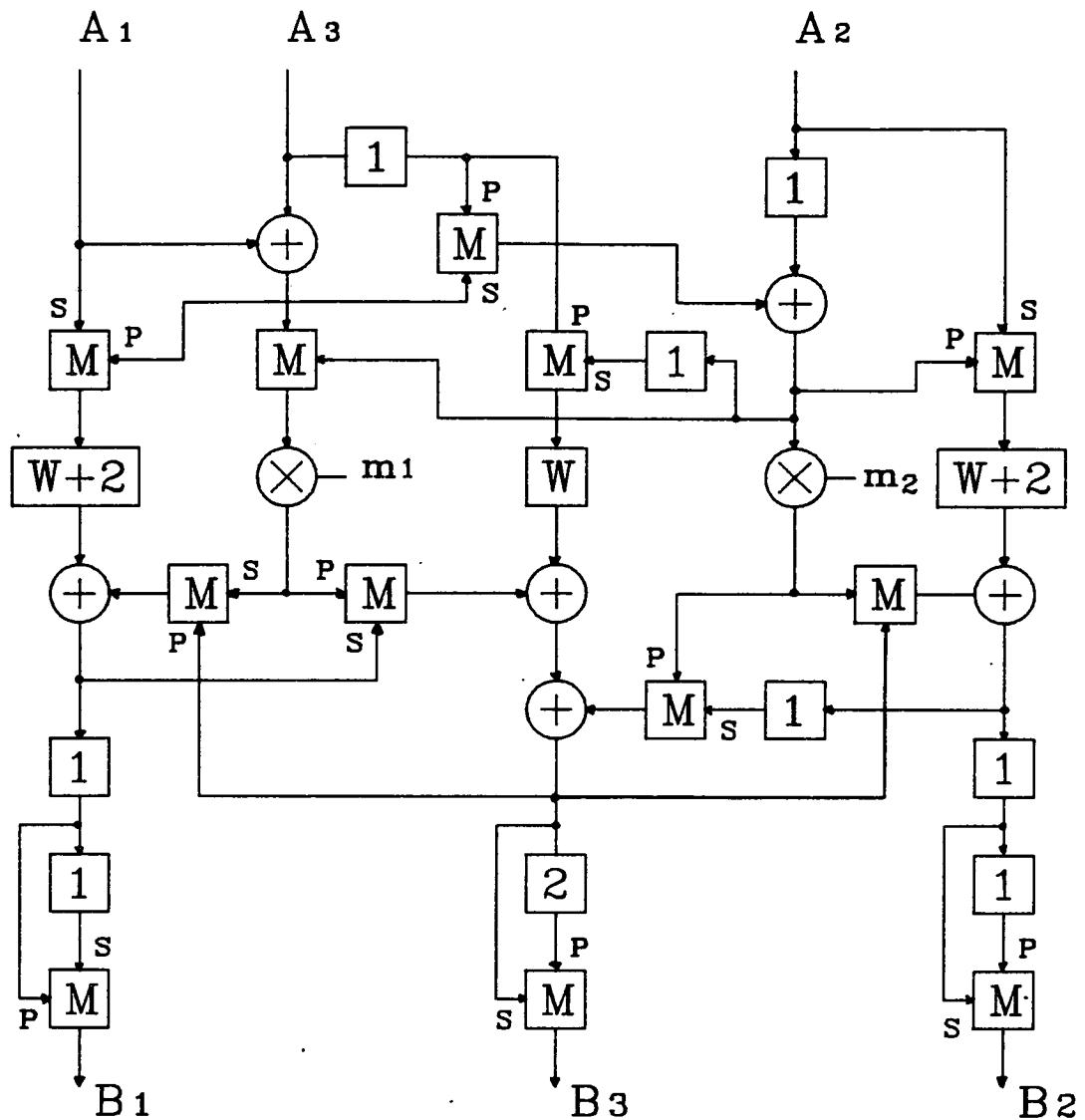


Figure 6-3 : General purpose wave filter adaptor

This circuit operates independently of the system clock, and therefore has no inherent latency. This feature of the multiplexer prevents the adaptor being reconfigured between the parallel and series modes whilst the filter is in operation.

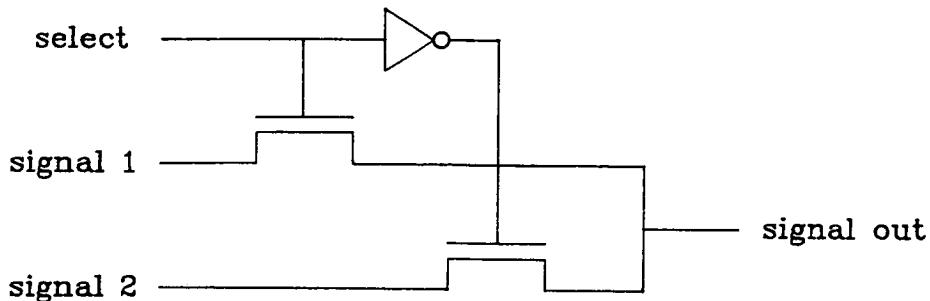


Figure 6-4: Simple multiplexer circuit

6.3.2 Bit-delay circuit

The implementation of bit-delay circuits in integrated circuits can be achieved using two methods - read/write memories and shift registers. For signal processing applications where large amounts of data need to be stored or delayed, the most efficient method of realising this is to use read/write memory cells, using random or sequential access systems. Of the two systems, random access memories are more versatile in their operation, but require large overheads in the row and column decoders required to access each memory location. Sequential access memories use a much simplified control structure whereby memory locations are read from, or written to, sequentially. This reduces the hardware requirement for the decoder circuitry. However, both systems are unsuitable for situations where only small bit-delays are required - such as those required by the general purpose adaptor. These smaller delays can be

more efficiently realised using shift register cells.

The simplest form of shift register which may be realised as an integrated circuit is a series of inverters, separated by pass-transistors which are controlled by a two-phase, non-overlapping system clock. An example of a 2-stage shift register is given in Figure 6-5.

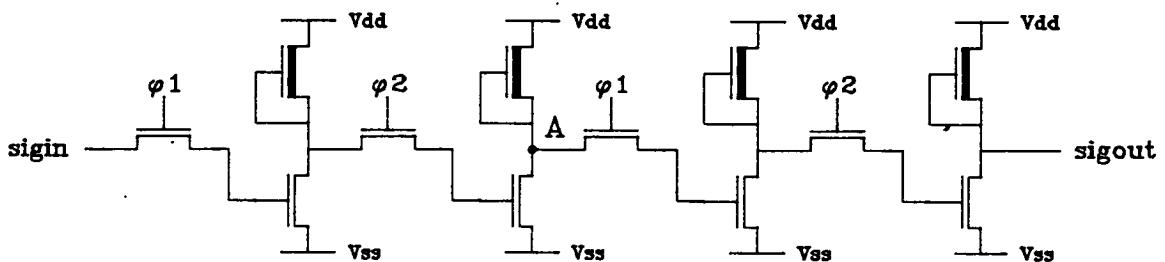


Figure 6-5: Two-stage shift register circuit

In this diagram, a signal present at the *sigin* input is transferred to the *sigout* output under the control of the system clock. On φ_1 , the signal is clocked into the system and inverted. On φ_2 , the inverted signal is applied to the next inverter in the chain. At this stage, the signal present on node *A* is the input signal, delayed by one system clock cycle. This process can be repeated m times to realise an m -stage shift register. The silicon layout of a two stage shift register is given in Figure 6-6.

The detailed silicon layouts presented in this chapter were all generated by Mr C. H. Iau, Dept. Electrical Engineering, University of Edinburgh. These designs have been produced using the generalised layout rules for silicon-gate NMOS design. A detailed description of these rules can be found in Mead & Conway [59].

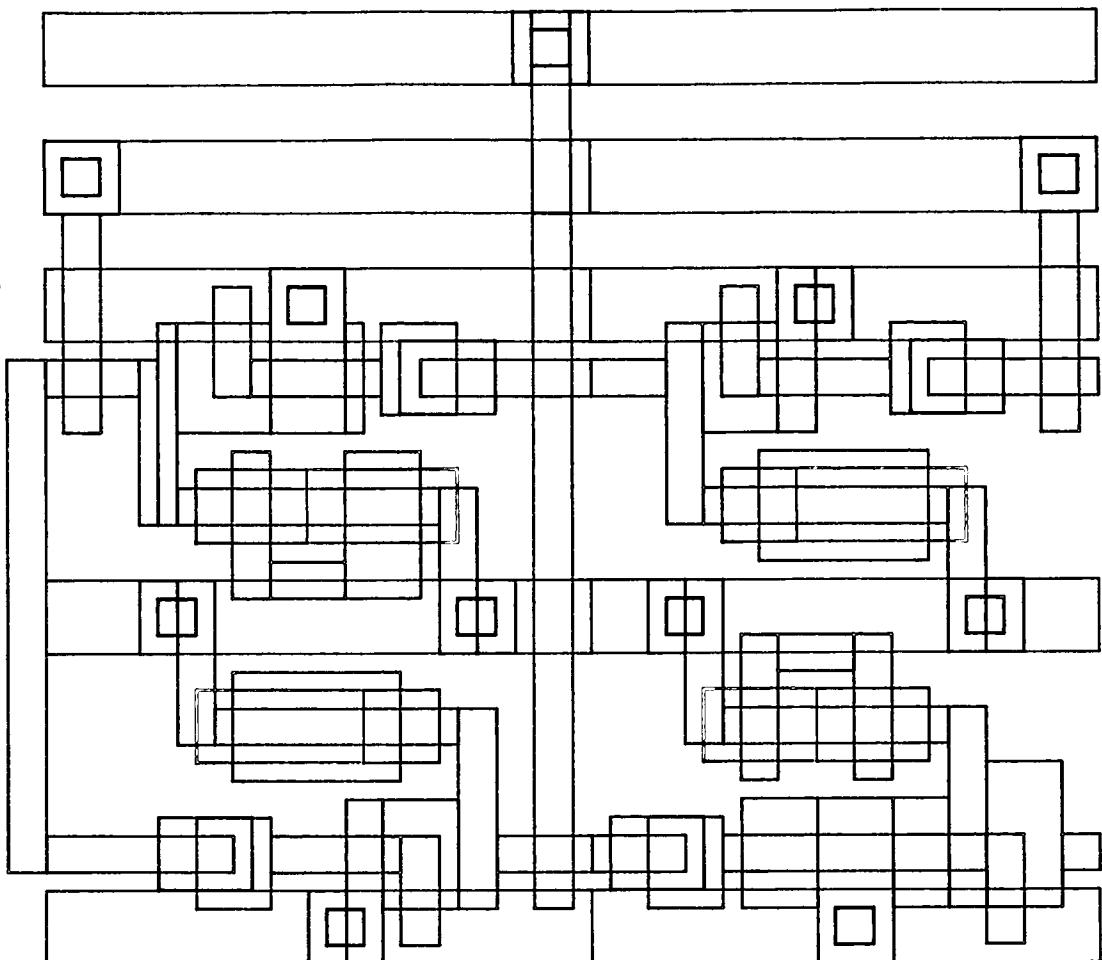


Figure 6-6: Silicon layout corresponding to a two-stage shift register

6.3.3 Binary addition

The addition of binary numbers in bit-serial arithmetic can be achieved using a wide variety of methods. Of these techniques, the most widely used circuits are *carry-save* adders, in which the *carry* generated by the addition of two signals, A and B , is saved in a latch and used in the subsequent addition. In this way, the addition of two bit-serial words can be correctly evaluated

if the words enter the adder *LSB* first. Any carry generated by the addition of the *MSB* of two binary words must be cleared when the *LSB* of the new words enter the adder.

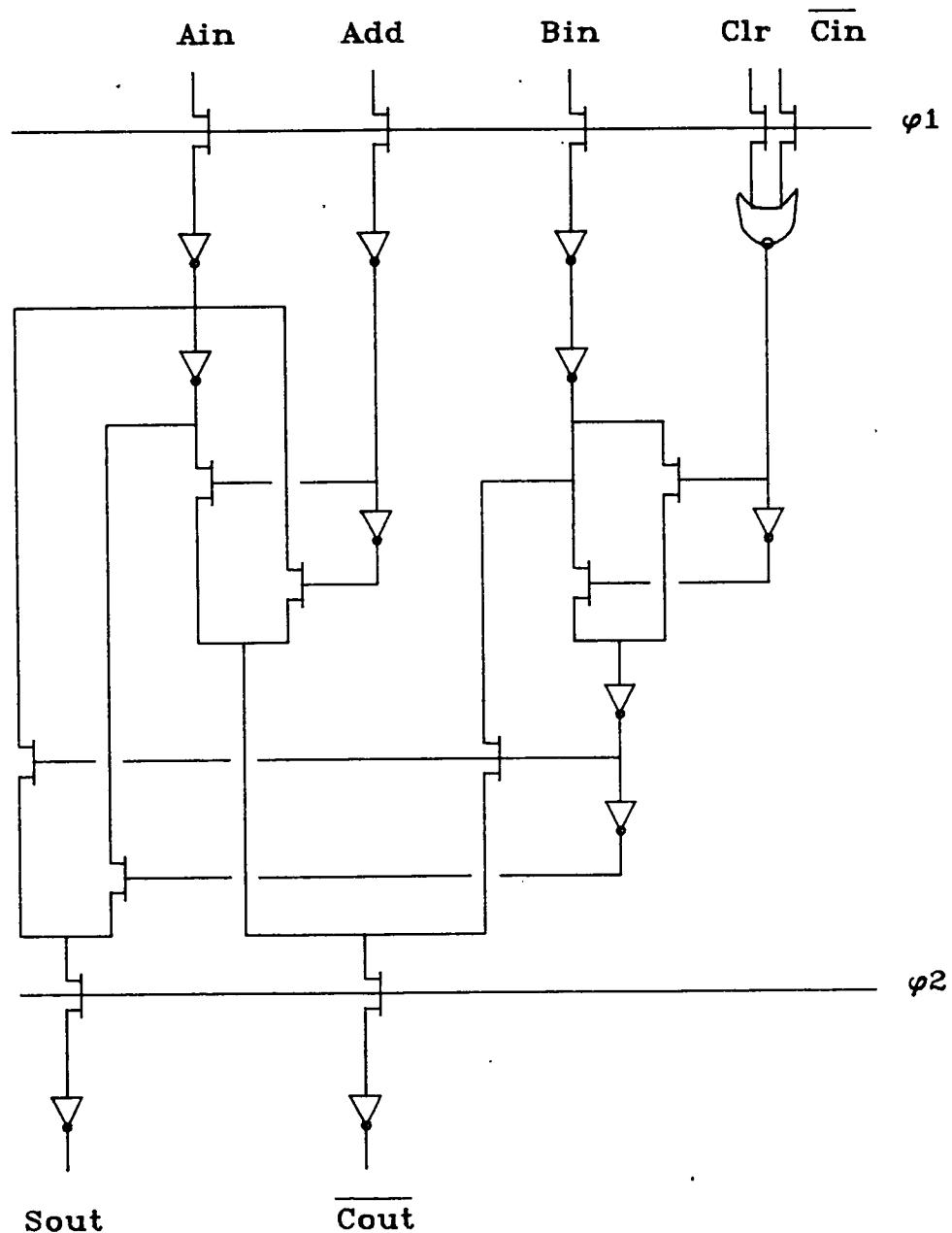


Figure 6-7 : Two-input carry-save adder

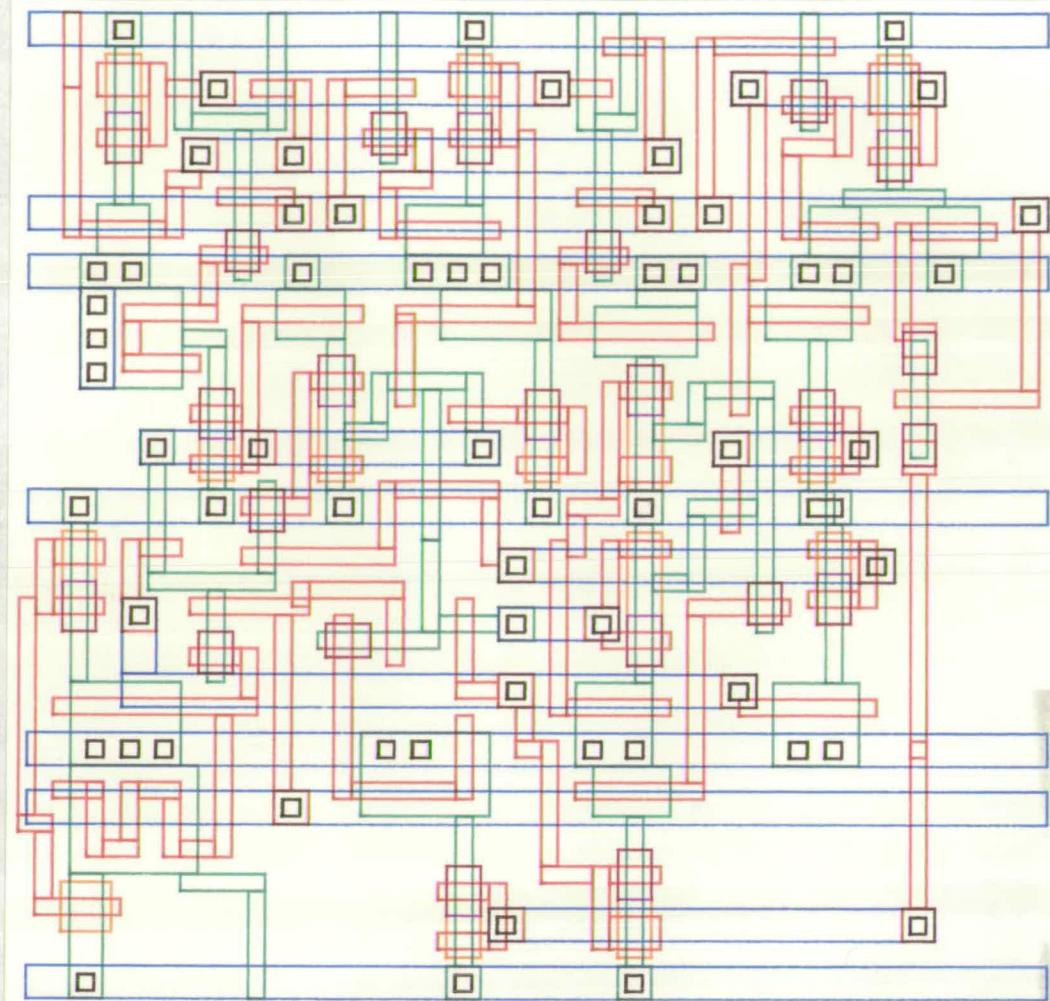


Figure 6-8 : Silicon layout corresponding to adder of Figure 6-7

A large number of possible circuit configurations for carry-save adders have been presented [44]. The adder circuit used in the design of the general purpose adaptor was based on a *data-selector* adder [44]. This circuit makes extensive use of pass-transistors to select the correct *sum* and *carry* depending on both the inputs to the adder and the carry generated in the previous

addition. The carry-save adder circuit and the corresponding silicon layout are given in Figures 6-7 and 6-8 respectively.

6.3.4 Binary multiplier circuit

In section 6.2, the possible circuit configurations for a bit-serial binary multiplier were presented. The multiplier architecture chosen for the integrated wave filter adaptor was the modified Booth's algorithm multiplier. After extensive simulation of the complete filter, a coefficient wordlength of 8-bits was chosen, as this was the minimum multiplier size which gave an acceptable filter response for high order filters, and which could be easily quantised to measure the sensitivity of the adaptor structures. This choice of architecture gives a multiplier latency, L_{mult} of 14 cycles, but generates a rounded and truncated m bit product in only m clock cycles.

The silicon layout corresponding to a pipelined, rounding, modified Booth's algorithm multiplier with an 8-bit multiplier coefficient is given in Figure 6-9.

6.3.5 Integrated adaptor circuit

The complete adaptor circuit may be produced by connecting the circuit elements according to the signal flowgraph of Figure 6-5. In this process, the

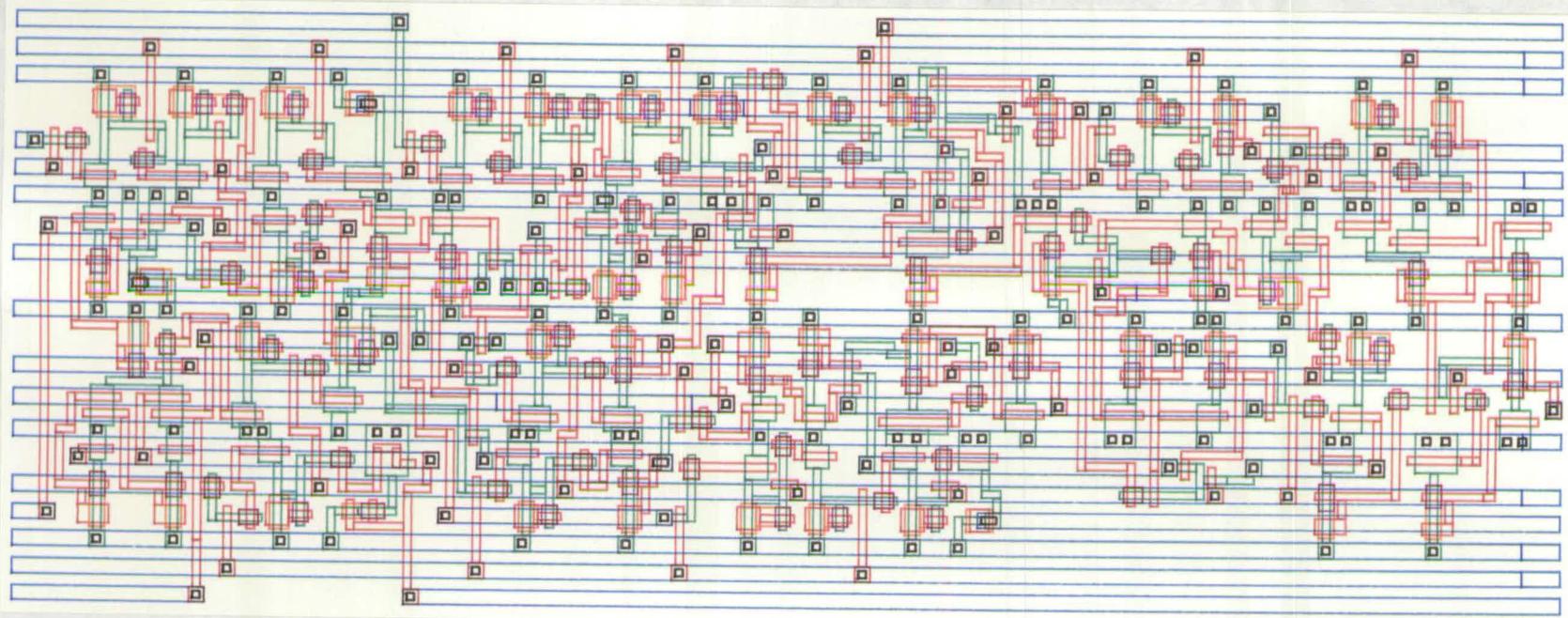


Figure 6-9 : Silicon layout of pipelined, rounding modified Booth's algorithm multiplier

appropriate synchronising delays are inserted where required, and do not form a specific part of the overall circuit. A photograph of the complete integrated wave filter adaptor is given in Figure 6-10. The complete circuit measures 5mm by 5mm. This size is determined by the standard frame in which the circuit must fit, rather than on the minimum size of circuit required.

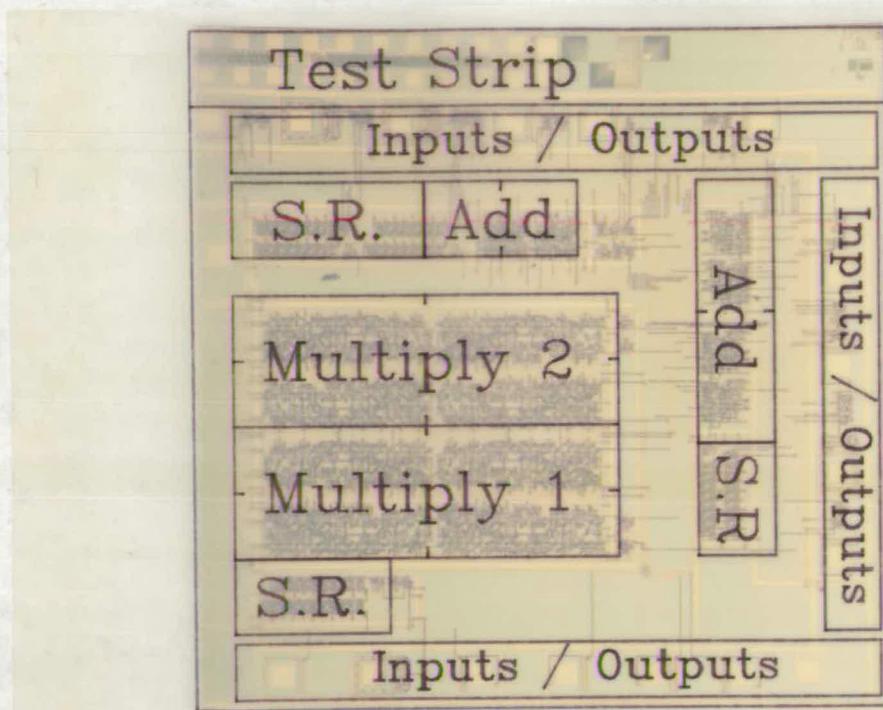


Figure 6-10 : Photograph of complete integrated adaptor circuit

6.4 Realisation of Example Filter

The integrated circuit described in section 6.2 has been used to realise a third-order lowpass wave filter. In this section, the external circuits required to imple-

ment this filter will be discussed, and practical results presented.

6.4.1 Example reference filter

The example reference filter chosen to demonstrate the use of the integrated adaptor circuit was a third-order lowpass filter with a Chebyshev response. The reference structure was generated using the synthesis techniques described in chapter 2, and had a passband ripple of 1.0 dB and a cutoff frequency of 15% of the sample clock rate. In the synthesis procedure, two zeros of transmission were required at the Nyquist frequency, and one unit element required. The zeros of transmission have been realised by parallel capacitors as shown in Figure 6-11 (a).

The component values for the reference filter were as follows.

$$R_L = R_S = 1 \Omega$$

$$C_1 = C_2 = 3.935 F$$

$$UE1 = 2.109 \Omega$$

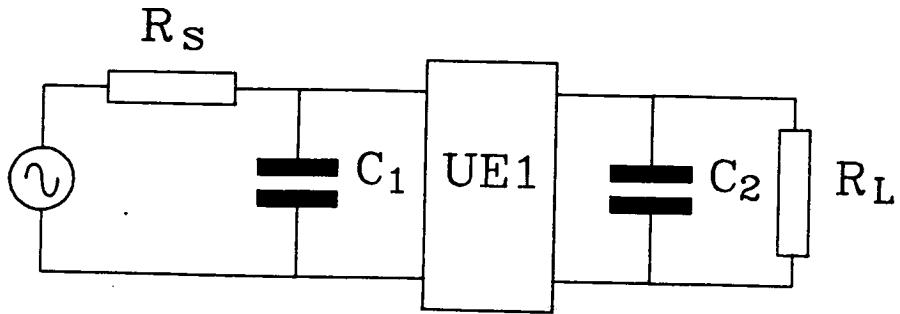
The multiplier coefficients for each adaptor, found in the usual way, are as follows

$$\alpha_{11} = 0.3697123$$

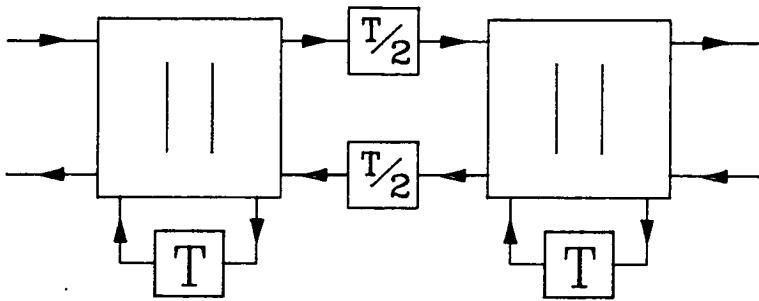
$$\alpha_{21} = 0.1753353$$

$$\alpha_{12} = 0.1753353$$

$$\alpha_{22} = 0.3697123$$



(a)



(b)

Figure 6-11 : (a) Lowpass reference filter
(b) Wave filter equivalent

For the parallel adaptor circuit, these coefficients are required by the circuit in the form

$$m_{i,j} = -\alpha_{i,j}$$

As these modified coefficients are all negative numbers, their binary word representation will be padded with 1's.

6.4.2 Test circuit requirement

To realise the wave filter as a practical circuit, a method must be developed for producing the required bit-serial binary sequence from some arbitrary analogue

input. Similarly, the filter output must also be converted to a corresponding analogue signal. This can be achieved using the circuit of Figure 6-12.

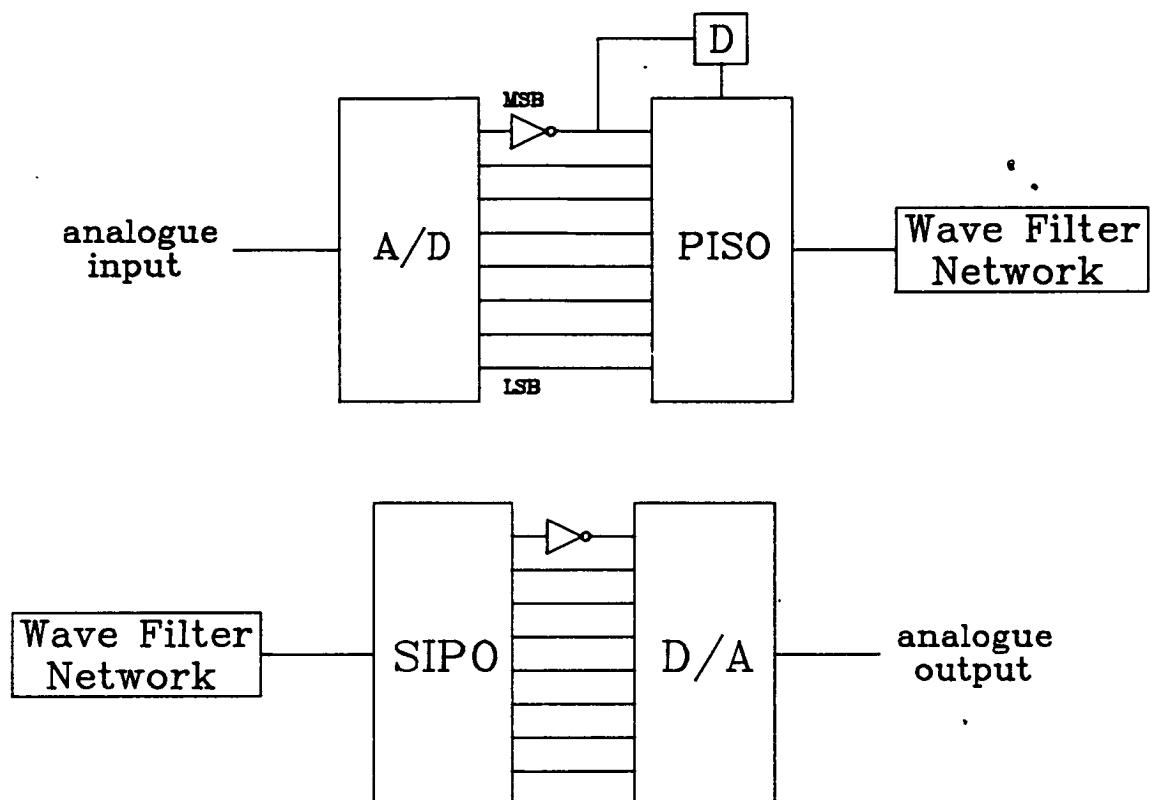


Figure 6-12 : Complete filter system

In this system, additional circuitry is required to convert the analogue input into binary form, convert this parallel signal word into a serial data word of appropriate length, and to reverse the entire process at the filter output. Let us consider the various subcircuits separately.

The first stage of the filtering process is the

analogue to digital (A/D) conversion. The device used for this was the National Semiconductor 8-bit converter, ADC0820. This device has been designed to operate at sample rates of less than 7 kHz without the need for an external sample-and-hold circuit. The 8-bit output of the ADC can be converted into two's-complement form by inverting the MSB of the parallel output word. The two's-complement output is then loaded into an 8-bit parallel-in series-out (PISO) shift register. To ensure that the MSB of the signal word is correctly extended, a D-type flip-flop has been included to latch the MSB of the signal word and present it at the serial input of the shift register. The conversion from parallel-load to series-shift requires one system clock cycle, so the sample period, T, has been increased from 38 to 39 cycles.

The bit-serial data is then processed by the wave filter and the binary output converted back to an analogue signal. This is achieved by passing the filter output through a serial-in/parallel-out (SIPO) shift register, and selecting the appropriate 8-signal bits in the data word. The digital to analogue converter (DAC) contains an 8-bit latch which is used to hold the signal word whilst the DAC conversion takes place. For this reason, the shift register may run continuously, and need not disable its input while the DAC is actually producing an analogue output. Once again, the MSB of the signal

word must be inverted to produce the correct magnitude-only output.

In order to program the wave filters response, data must be supplied to the adaptors to provide the coefficients for the multipliers and to determine whether the adaptors are to operate as parallel or series circuits with inductive or capacitive terminations. To satisfy this requirement, all the timing and control signals required by the filter system were generated by a Tektronix 9100 series Digital Analysis System (DAS). This machine can operate in two modes ; as a logic analyser and as a pattern generator. In the logic analyser mode, the DAS can monitor signals in a digital system to verify the correct operation of the circuit. This process is often referred to as data acquisition. In its pattern generator mode, the DAS can generate serial data signals which can be used to control the wave filter system. These modes may be combined to test circuits in isolation, and was the method used to initially verify that the adaptor chip was functionally correct.

The system clock used to drive the circuits was generated internally by the DAS system. However, for a more exhaustive test at higher system clock rates, an external clock source may be used to synchronise the pattern generation and data acquisition sections of the DAS. For

the example filter, the system clock frequency was 100 kHz, giving a sample frequency of 2.56 kHz. The clock frequencies involved in the testing of this device were merely to verify the system design, and were not intended to discover the maximum clock frequency of the adaptor, which are dependent on the parameters of the silicon fabrication process.

The frequency response of the filter system was measured on a Hewlett-Packard HP-3585A Spectrum Analyser. The analogue signal used to stimulate the filter system was provided by the tracking generator of the spectrum analyser. This facility provides a swept frequency signal between the defined frequency limits. For the example filter, the start and stop frequencies used were 20.0 Hz and 1.277 kHz respectively. The stop frequency is the Nyquist frequency of the filter system. The accuracy of the measurement system is given by the resolution bandwidth (RBW) of the spectrum analyser, and was 10 Hz for the example filter.

6.4.3 Practical results for example filter

The measured frequency response of the filter realisation is given in Figure 6-13.

The frequency response of Figure 6-13 shows how the measured response closely matches the desired response (shown as crosses on the response). The measu-

REF 14.2 DBM
10 DB/DIV

MARKER 382.1 HZ
RANGE 15.0 DBM 13.7 DBM

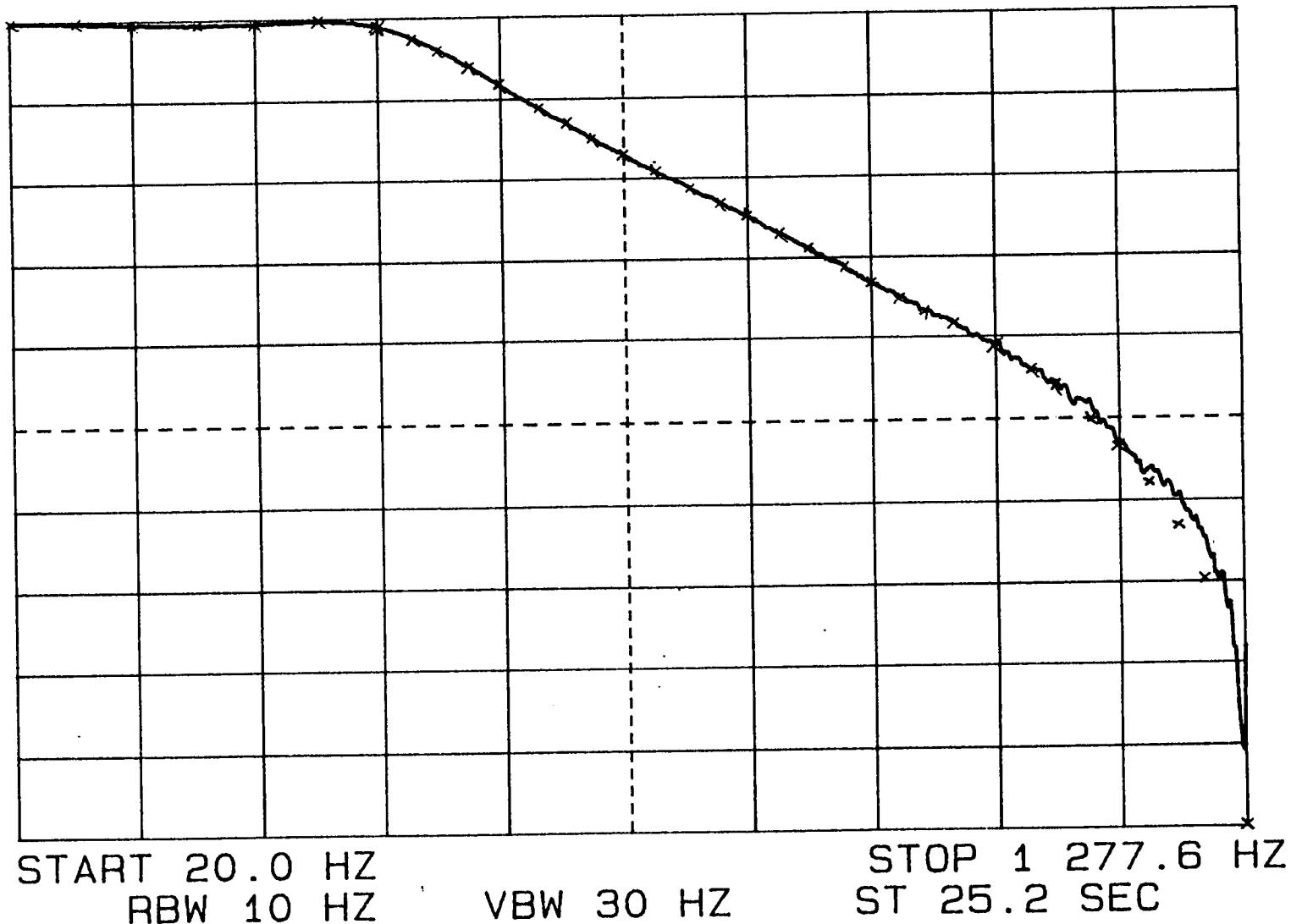


Figure 6-13 : Practical response of third-order filter

ured passband ripple was 0.96 dB and does not allow for the $\sin x/x$ roll-off effect caused by the sampling process. The marker frequency quoted on the measured response corresponds to the cutoff point of the lowpass filter. For the example filter, this should be 15% of the system clock frequency, or 384.6 Hz. The measured figure of 383.4 Hz is within an acceptable range of the desired value.

6.5 Summary

In this chapter, a design for an integrated circuit realisation of a general purpose wave filter adaptor has been presented. This circuit can model the adaptor equations for both parallel and series three-port networks, and also provides an inductive or capacitive filtering element connected across port 3 of the structure. Design aspects such as binary addition and multiplication have been considered and the resulting integrated circuit has been fabricated and used in a practical filter design. The additional circuitry required for such a filter has been given, and practical results presented.

Various architectures have been developed for the realisations of the four function blocks required by the adaptor circuit - multiplexer, bit-delay, adder and multiplier. Of these circuits, emphasis has been placed on the design of the adder and the multiplier as these are

the circuit elements which determine the physical size of the resulting circuit. Two multiplier algorithms have been considered; Canonic Signed Digit and the modified Booth's algorithm multiplier. Both techniques offer advantages and disadvantages which must be considered in the design of such a system. The CSD multiplier offers a small area requirement in dedicated situations, but requires complex control circuitry where a programmable multiplier is required. In contrast, the modified Booth's multiplier requires a slightly larger silicon area, but is better suited to programmable multipliers and offers a smaller multiplier latency when realised as a pipelined structure.

The integrated circuit of section 6.2 has been designed using a modified Booth's algorithm multiplier with an 8-bit coefficient wordlength. This circuit has been fabricated on a $5 \mu\text{m}$ NMOS process at Edinburgh University, and functionally tested using digital techniques.

A third-order lowpass filter has been realised using the integrated general purpose adaptor. The additional circuitry required to interface analogue signals to and from the filter has also been presented. The frequency response of the filter has been measured on a spectrum analyser, and was within acceptable limits in both passband ripple and cutoff frequency.

Chapter 7

CONCLUSIONS

7.1 Summary

The aim of the work presented in this thesis has been to develop a digital wave filter adaptor circuit which can be used as a general purpose building block, capable of realising *any* wave filter based on a distributed element reference filter.

The filter theory required to design such a circuit has been presented in Chapter 2. Wave filter models have been developed for all the widely used filter elements - resistors, inductors, capacitors and unit-elements, and also for FDNR based ψ^2 -impedance elements. The range of reference filters which may be realised using these elements has also been presented. In particular, a synthesis procedure for an optimised filter response based on distributed element reference filters has been reported.

The various methods used to implement wave filters have been discussed in Chapter 3, and include the use of both analogue and digital techniques to model wave filter adaptors. However, each of the filter systems presented in this chapter dealt with a particular filtering

requirement, and could not be easily changed to deal with situations where a programmable filter response was required.

To achieve such a programmable response, a series of digital adaptor circuits was presented in Chapter 4. These circuits dealt with the detailed circuit and timing requirements of both two- and three-port adaptors. The parallel and series three-port adaptors were then combined to produce two new adaptor circuits - the *general purpose* adaptor and the *universal* adaptor. These circuits are capable of modelling parallel or series adaptors, with inductive or capacitive terminations connected across port 3 of the structure. In addition, the universal adaptor operates with a 100% duty cycle and can be easily multiplexed to reduce the hardware requirement of a filter system.

The universal adaptor has been used to implement a seventh-order lowpass filter system. The reference filter, generated using the synthesis techniques presented in Chapter 2, can be realised using only two universal adaptors. An implementation based on conventional parallel adaptors would require four such structures. This filter has been extensively simulated in both the time and frequency domain, and results presented in Chapter 5.

A silicon integrated circuit realisation of the general purpose adaptor of Chapter 4 has been presented in Chapter 6. This circuit has been designed and fabricated on a 5 μm silicon-gate NMOS process at Edinburgh University. A third-order lowpass filter has been implemented using this circuit, and practical results presented in this chapter.

7.2 Bit-Serial Digital Wave Filters

The implementation of integrated digital wave filters using bit-serial arithmetic offers many advantages over equivalent parallel architectures. The major advantage is clearly in the physical size of the resulting integrated circuit. Parallel arithmetic architectures require large areas of silicon area to evaluate the required adaptor equations, particularly for the multiplication process. For this reason, several alternative parallel arithmetic architectures which do not use "true" multiplication, have been suggested. However, these alternative systems are generally unsuitable for use in situations where a programmable filter response is required. For this reason, the use of digital signal processing microprocessors would seem to be the only feasible way of implementing programmable digital wave filters using parallel arithmetic.

In contrast, bit-serial arithmetic realisations of

digital wave filters offer a small physical size due to the reduction in complexity of both the arithmetic elements and the interconnections between them. Similarly, the power consumption of the circuit, and the pin-count of the resulting packages will both be smaller. A further benefit of bit-serial chips is that the yield of working devices from a wafer of silicon will also increase. This is due to the smaller die size of the chip which spreads the defect density of the wafer across a larger number of possible candidates.

Although the use of bit-serial techniques generally allows an increase in the system clock rate, this is not sufficient to compensate for the increase in data throughput that results from using parallel arithmetic in the filter system. Parallel arithmetic offers word-wide communication within the system, with which bit-serial circuits cannot compete.

More specific advantages of bit-serial filters are derived from the long data wordlength that results in such a system. The use of pipelined bit-serial multipliers results in a long adaptor latency, where the sign bit of the signal word must be extended to fill the rest of the data word. This results in "redundant" bits in the data word, which can be used to allow automatic signal scaling and to suppress the effect of parasitic

oscillations in the filter system. In parallel arithmetic architectures, where the data wordlength and the signal wordlength are equal, these guard bits are an integral part of the signal wordlength, and their use results in a reduction in the effective dynamic range of the system.

The use of bit-serial arithmetic in the design of the universal adaptor enables high-order filters, with a programmable response, to be realised on one silicon chip. As an example of such a filter system, it is predicted that a single integrated circuit, measuring 5mm by 4mm, which contains four independent universal adaptors, could be designed using 2.5 μm CMOS technology. These could be configured, under external control, to implement any combination of filter between one fifteenth-order filter, and four second-order filters. The sample rate of these filters would be unaffected by the filter order, and would be determined only by the system clock rate and the latency of the universal adaptor. An equivalent implementation in parallel arithmetic would clearly be impractical.

The major disadvantage of bit-serial implementations of digital wave filters, is, however, the long latency associated with the adaptors. For a typical universal adaptor using 8-bit multiplier coefficients, the filter

may only sample every 44 clock cycles. For an integrated circuit with a system clock rate of 20 MHz, this gives a sample rate of 455 kHz. The Nyquist frequency for such a system would therefore be ~225 kHz. Wave filters use exact design techniques, and as such are capable of filtering accurately with cutoff frequencies of up to 50% of the sample clock rate. However, to avoid aliasing problems, sampled-data filters have upper cutoff frequencies which are a small fraction of the sample clock rate, typically up to 25% of F_s . In the wave filter system described above, the cutoff point of the filter will therefore be limited to frequencies around 110 kHz. In an equivalent parallel arithmetic system, sample frequencies of over 1 MHz could be achieved.

The limitations arising from this adaptor latency problem can be significantly reduced if the filter specification is well defined. For such a filter, the hardware requirement can be reduced in two ways. Firstly, extensive simulation of the filter can be employed to determine the minimum coefficient wordlength for each adaptor, which will still provide an acceptable filter response. For the example filter of Chapter 5, 8 bits of coefficient were required to satisfy the filter specification. However, for lower-order structures, or filters where the passband response is not so critical, smaller coefficient wordlengths can provide an acceptable

response.

The alternative method for increasing the sample rate is to used different means of evaluating the required arithmetic operations. Alternative multiplier algorithms such as the CSD multiplier [42] or ROM-based structures [45] are more suitable for dedicated filters than the more general purpose modified Booth's algorithm. Combining these features can result in a significant reduction in the latencies of the adaptors. As an example of this, consider a low order filter implemented using a CSD multiplier with only 4 bits of multiplier coefficient. The adaptor latency of such a filter could be reduced to only 13 cycles (6 for the multiplier and 7 for the other hardware), giving an increase in sample rate of 40%. It should be emphasised at this point that the filters described here are based on distributed element reference filters with unit-elements between each inductor and/or capacitor. Wave filters implemented using matched-port techniques cannot offer sample frequencies which are comparable to filters designed using synthesis techniques or the Kuroda transform method [20] using similar hardware requirements.

7.3 Applications of Digital Wave Filters

Originally, wave filters were developed as a means of retaining the benefits of classical RLC filters in a

digital environment. For this reason, the obvious application for digital wave filters is as a direct replacement for existing lumped element filters. Using synthesis techniques or filter tables, and the universal adaptor, digital wave filters can be designed to replace lowpass, highpass, bandpass and bandstop filters quickly and easily. Clearly, this cannot be applied as a general rule due to sample frequency limitations, but the direct replacement of RLC filters in the frequency range up to 150 kHz is possible using current technology.

The main limitation of direct replacement is, as previously mentioned, the range of sample frequencies over which digital wave filters can operate. Even with system clock rates of 20 MHz, wave filters based on the universal adaptor have a sample rate of only ~450 kHz. For this reason, the direct replacement of existing RLC filters will probably be limited to situations where the RLC filter is used in conjunction with an existing digital signal processing system. If, however, the system does not require a particularly high sample rate, then the application of digital wave filters becomes more appropriate.

In the fields of telecommunications and audio signal processing, there is a large market for high performance digital filters. In all applications where RLC filters

are currently in use, such as telephony, radio, television and hi-fi music systems, digital wave filters present an attractive alternative. Because of the ease of design of wave filters using the universal adaptor, potential users of the circuit would require no detailed knowledge of the internal operation of the device, greatly simplifying its application.

One particular application which is well suited to the digital wave filter is its use in filter banks. As the universal adaptor can be easily multiplexed to realise a wide range of programmable filter responses in one sample period, it is particularly well suited for use where a bank of filters is required to determine different spectral power components. Examples of systems requiring filter banks are in speech recognition and secure communication systems. As these systems operate in the audio frequency range, universal adaptor wave filter systems offer a simple alternative to conventional RLC filters. The versatile nature of the universal adaptor will also permit its use in application-specific integrated circuits (ASICs), where it could form one part of a more complex system such as a channel vocoder [62], for digital transmission of speech in low bandwidth environments such as telephony.

7.4 Suggestions for Future Work

In this thesis, an integrated circuit has been developed to provide a general purpose building block for the rapid development of digital wave filter systems. The circuit has been designed to allow any ladder filter based on distributed elements to be implemented in a digital network. The main limitation of this structure is not in its internal operation, but in the semiconductor technology used to realise the final circuit. In situations where this medium is insufficient to achieve the desired result, a semiconductor which can operate at higher frequencies is required. At present, the practical limits of speed in large scale integrated circuits such as the universal adaptor, is between 20 and 40 MHz. For digital filter applications at higher frequencies, alternative semiconductors such as Gallium Arsenide (GaAs) must be used. Such materials have higher carrier mobilities, resulting in higher operating bandwidths. The signal flowgraph for the universal adaptor can be greatly simplified where a dedicated filter response is required, making it an ideal candidate for the development of large structures in a developing semiconductor field such as GaAs.

With the adoption of better semiconductor technology, the applications of digital wave filters will become

more diverse. As the operating frequency of semiconductor devices increases, there will be a growing need for high speed digital communications. For this reason, the author suggests that future work in the world of digital wave filters should also concentrate on the applications of such devices in this field. Potential applications in this field include simultaneous speech and data transmission in telephony, video conferencing and the use of GaAs wave filters in fibre-optic based communication systems. With the current development in semiconductor technology, the future of digital wave filters seems assured.

Appendix I

COMPUTER AIDED FILTER SYNTHESIS

A suitable transfer function which will synthesise reference filters which have unit-elements which add to the order of the filter is given below.

$$|S_{2,1}(\Omega)|^2 = 1 / [1 + \epsilon^2 f^2(X, Y, Z)] \quad (I.1)$$

with

$$f(X, Y, Z) = \cosh [N \cosh^{-1}(X) + K \cosh^{-1}(Y) + L \cosh^{-1}(Z)] \quad (I.2)$$

$$X^2 = \frac{(1 + \Omega_2^2)(\Omega_1^2 - \Omega^2)}{(\Omega_1^2 - \Omega_2^2)(1 + \Omega^2)} \quad (I.3)$$

$$Y^2 = \frac{\Omega_2^2(\Omega_1^2 - \Omega^2)}{\Omega^2(\Omega_1^2 - \Omega_2^2)} \quad (I.4)$$

$$Z^2 = \frac{(\Omega_1^2 - \Omega^2)}{(\Omega_1^2 - \Omega_2^2)} \quad (I.5)$$

$$\Omega_1 = \tan(\phi_1) \quad (I.6)$$

$$\Omega_2 = \tan(\phi_2) \quad (I.7)$$

In this series of equations, the following notation has been used.

- N is the number of unit-elements.
- K is the number of zeros of transmission at d.c.
- L is the number of zeros of transmission at the Nyquist frequency.
- ψ_1 and ψ_2 are the lower and upper cutoff frequencies of the filter.

Low-pass and high-pass filters can be obtained by setting $K=0$ and $\psi_1 = 0^\circ$ for a low-pass response, and $L=0$ and $\psi_2 = 90^\circ$ for a high-pass response.

To perform the required filter synthesis, this expression for $|S_{21}|^2$ must be put in the form

$$|S_{21}(\Omega)|^2 = N(\Omega) / D(\Omega) \quad (\text{I.8})$$

where $Num(\Omega)$ and $Den(\Omega)$ are ratioed polynomials in the variable Ω . Then, Ω must be replaced by λ using

$$\lambda^2 = -\Omega^2 \quad (\text{I.9})$$

and evaluating $|S_{11}|^2 = 1 - |S_{21}|^2$ assuming that the network is lossless and reciprocal.

Next, the roots of $|S_{11}|^2$ must be found, and those in the left hand plane chosen. We can thus find $|S_{11}|(\lambda)$. We now require to calculate

$$Z_{in}(\lambda) = \frac{1 + S_{11}(\lambda)}{1 - S_{11}(\lambda)} \quad (\text{I.10})$$

and extract elements from Z_{in} in the conventional way to get element values and the filter topology.

Unfortunately, this filter synthesis is numerically ill-conditioned. There are several ways of dealing with this difficulty. The most obvious solutions are;

- Change the frequency variable in such a way that the problem becomes well-conditioned.
- Increase the accuracy of the arithmetic.

The first method is best if one particular problem is to be solved. However, in the completely general case, it may be difficult to find a suitable alternative frequency variable. For this reason the second approach has been shown to be the more appropriate solution.

The most efficient way of increasing the numerical accuracy of a computer is to employ extra hardware. However, this is impractical, and the problem must therefore be solved using high precision arithmetic routines in software. Such routines have been written in FORTRAN on a DEC-10 machine. The routines work to the degree of accuracy specified by the user with a current limit of 100 decimal places. The library of routines includes the following operations or tests.

$+, -, *, /, \sqrt{x}, \sin x, \cos x, \tan x, =,$
 $>, \geq, \leq, <, <=, =0, \pi/2, \pi/4, 2\pi$

All these routines have been tested and work to the

required accuracy. They all round their outputs to the nearest decimal place to ensure that the answer is correct to the accuracy specified.

The arithmetic routines have been used in the program SYNTH, which gives Z_{in} from the filter specification. $|S_{12}|^2$ is found as a ratio of rational polynomials efficiently and accurately because much of the problem has been solved algebraically. The calculation of $|S_{11}(\lambda)|^2$ is also fast, but the real difficulty lies in finding the roots of $Num(\lambda)$ and $Den(\lambda)$ where

$$|S_{11}(\lambda)|^2 = Num(\lambda) / Den(\lambda) \quad (I.11)$$

This root finding operation must be carried out to high accuracy if the final element values are to be accurate. Consequently, this section of the synthesis problem takes a great deal of computer time compared to the other sections.

The roots of the expression are first found using double precision arithmetic, and then further iterations are carried out using the Newton-Raphson technique to improve their accuracy to the desired level. Once the roots have been found, the input impedance function Z_{in} can be generated, enabling the filter designer to extract filtering elements in the usual way in order to satisfy the required transfer function.

Appendix II

FIRST COMPILER FILE FOR UNIVERSAL ADAPTOR

II.1 Introduction

The universal adaptor presented in chapter 4, has had it's physical layout in silicon technology developed entirely using the FIRST silicon compiler [55]. The FIRST suite of computer programs permits users to design signal processing systems by placing pre-defined blocks of silicon layout around a central routing channel under the control of the program. The compiler can guarantee 100% routing for each system as the routing channel can be expanded as the number of connecting nodes increases.

The FIRST description of a filter system is based on a hierarchical structure of user- and program-defined operators. In a filter based on the universal adaptor, there are three levels of hierarchy.

1. Operator level description
2. Chip level description
3. System level description

The *operator* level description of the universal adaptor gives details of the interconnection between the various arithmetic elements in the adaptor. The *chip*

level description shows how these operators are placed to form individual integrated circuits. Large signal processing systems may require more than one chip description to satisfy the algorithmic description of the system. Finally, the *system* level description shows how all the individual chips are connected to realise the complete system. Within the system description for the example filter of chapter 5, there will be both universal adaptor chips, and control generator chips to generate the required control signals for the correct operation of the filter.

For the universal adaptor system, a range of designs was required to allow for the variation in multiplier coefficient wordlength. To simplify this requirement, the FIRST language file which describes the universal adaptor has been parameterised. The multiplier coefficient wordlength determines the overall timing of the system as both the system wordlength and some of the synchronising delays are dependent on the coefficient wordlength. The parameter used in the FIRST language description, *coeff*, can be used wherever the description of a particular operator is dependent upon it.

II.2 FIRST language file for universal adaptor

```
OPERATOR EITHER[coeff] (c1wt0,intext,pors,clorc,a3orex)
    a1,a2,a3,m1,m2 -> b1,b2

SIGNAL s1 THROUGH 24,m1delay,m2delay
CONTROL c1wt1 THROUGH 8,cpors1 THROUGH 4,clorc3

CONSTANT w=3*coeff/2+2

MULTIPLY [1,coeff,0,0] (c1wt3 -> c1wt4)
    s6,m1delay -> s9,s8
MULTIPLY [1,coeff,0,0] (c1wt3 -> NC)
    s7,m2delay -> s10,NC

MULTIPLEX [w+5,0,0] (cpors1) s2,s5 -> s14
MULTIPLEX [w+5,0,0] (cpors1) s5,s2 -> s15
MULTIPLEX [w+5,1,0] (clorc3) s19,s20 -> s21

ADD [1,0,0,0] (c1wt1) s5,s24,GND -> s1,NC
ADD [1,0,0,0] (c1wt1) s2,s24,GND -> s4,NC
ADD [1,0,1,0] (c1wt2) s1,s2,GND -> s3,NC
ADD [1,0,0,0] (c1wt7) s17,s18,GND -> s19,NC
ADD [1,0,0,0] (c1wt4) s9,s10,GND -> s11,NC
ADD [1,0,0,0] (c1wt5) s14,s12,GND -> b1,NC
ADD [1,0,0,0] (c1wt5) s13,s15,GND -> b2,NC
ADD [1,1,0,0] (c1wt6) b1,s16,GND -> s17,NC

SUBTRACT [1,0,0,0] (c1wt4) s9,s8,GND -> s22,NC
SUBTRACT [1,0,0,0] (c1wt4) s10,s8,GND -> s23,NC
SUBTRACT [1,0,0,0] (c1wt8) GND,s19,GND -> s20,NC

BITDELAY [5] s8 -> s18
BITDELAY [4] m1 -> m1delay
BITDELAY [4] m2 -> m2delay

MULTIPLEX [1,1,0] (intext) b2,a1 -> s5
MULTIPLEX [1,0,1] (intext) a2,b1 -> s2
MULTIPLEX [1,1,0] (cpors2) s1,s3 -> s6
MULTIPLEX [1,1,0] (cpors2) s4,s3 -> s7
MULTIPLEX [1,0,0] (cpors3) s11,s22 -> s12
MULTIPLEX [1,0,0] (cpors3) s11,s23 -> s13
MULTIPLEX [1,0,0] (cpors4) GND,b2 -> s16
MULTIPLEX [1,0,0] (a3orex) s21,a3 -> s24

CBITDELAY[1] (c1wt0 -> c1wt1)
CBITDELAY[1] (c1wt1 -> c1wt2)
CBITDELAY[2] (c1wt2 -> c1wt3)
CBITDELAY[2] (c1wt4 -> c1wt5)
CBITDELAY[2] (c1wt5 -> c1wt6)
CBITDELAY[1] (c1wt6 -> c1wt7)
CBITDELAY[1] (c1wt7 -> c1wt8)
```

```
CBITDELAY[1] (pors -> cpors1)
CBITDELAY[2] (cpors1 -> cpors2)
CBITDELAY[w+2] (cpors2 -> cpors3)
CBITDELAY[2] (cpors3 -> cpors4)
CBITDELAY[3] (clorc -> clorc3)
```

END

!=====

```
CHIP PSAD (cc1,cinex,cpors,clorc,ca3orex)
           aa1,aa2,aa3,mm1,mm2 -> bb1,bb2
```

```
SIGNAL a1,a2,a3,b1,b2,m1,m2
CONTROL c1,intext,pors,lorc,a3orex
```

```
PADIN (cc1,cinex,cpors,clorc,ca3orex ->
        c1,intext, pors,lorc,a3orex)
        aa1,aa2,aa3,mm1,mm2 -> a1,a2,a3,m1,m2
PADOUT b1,b2 -> bb1,bb2
```

```
PADORDER VDD,cc1,cinex,cpors,clorc,ca3orex,
          GND,CLOCK,mm1,mm2,aa1,aa2,aa3,bb1,bb2
```

```
EITHER[10] (c1,intext,pors,lorc,a3orex)
            a1,a2,a3,m1,m2 -> b1,b2
```

END

!=====

```
CHIP CGEN (-> pc1, pcinex, pcpors, pclorc, pca3orex)
```

```
CONTROL c1, c2, c2a
```

```
PADOUT (c1, c2, c2, c2 ->
         pc1, pcinex, pcpors, pclorc, pca3orex)
```

```
PADORDER VDD, pc1, pcinex, pcpors, pclorc,
          pca3orex, GND, CLOCK
```

```
CONTROLGENERATOR ( -> NC, c1, c2)
                  CYCLE[w+8]
                  CYCLE[2]
ENDCONTROLGENERATOR
```

END

!=====

```
SYSTEM FILTER signin -> sigout
```

```
CONTROL sc1, scinex, scpors, sclorc, sca3orex
SIGNAL s1,s2,sm1,sm2,sm3,sm4,s2a2,sa31,sa32

PSAD (sc1,scinex,scpors,sclorc,sca3orex)
      sigin,s2,sa31,sm1,sm2 -> NC,s1
PSAD (sc1,scinex,scpors,sclorc,sca3orex)
      s1,s2a2,sa32,sm3,sm4 -> s2,sigout

CGEN ( -> sc1,scinex,NC,NC,NC)
```

END

ENDOFPROMGRAM

The fully synchronised signal flowgraph for the universal adaptor is given in Figure II-1. In this diagram, single bit-delays have been represented by a double arrow on the input to an operator. The FIRST floorplan for this adaptor is given in Figure II-2.

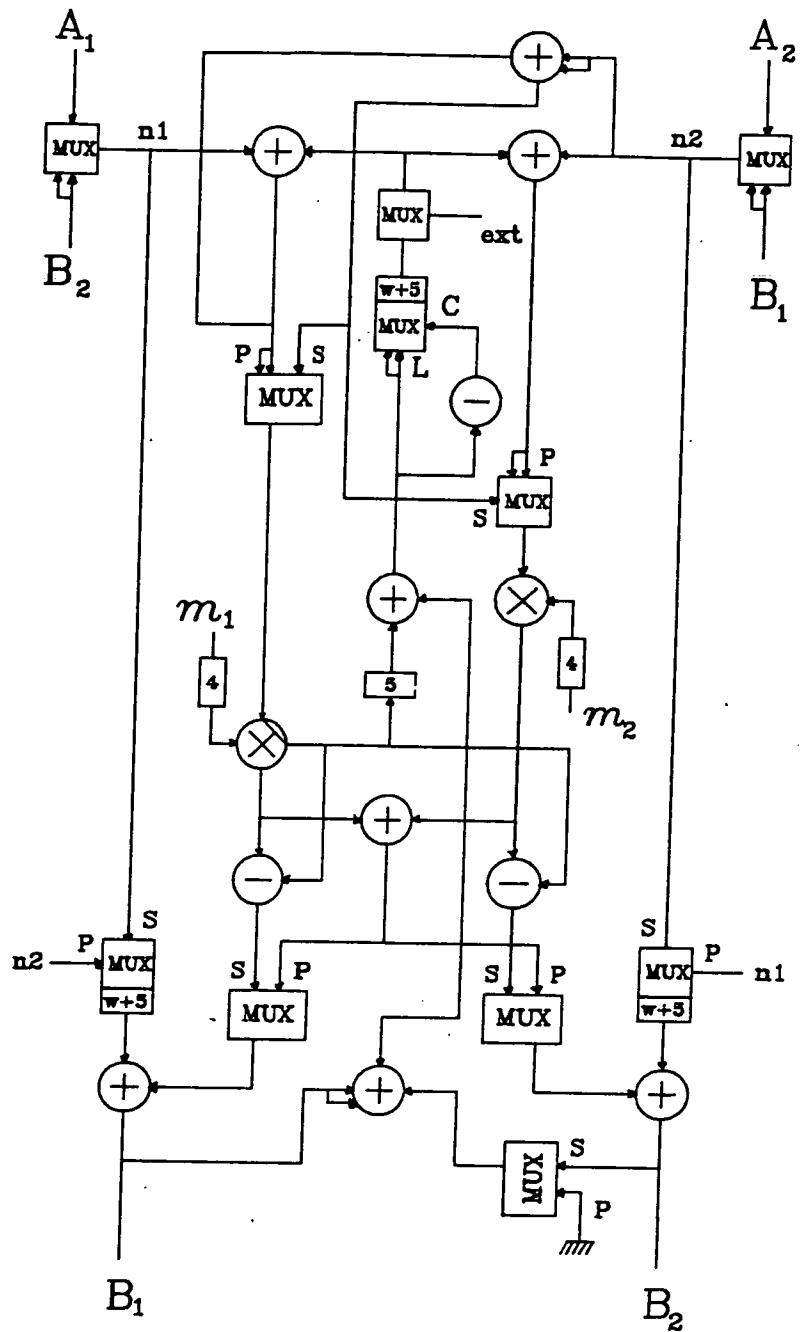


Figure II-1 : Synchronised signal flowgraph for universal adaptor

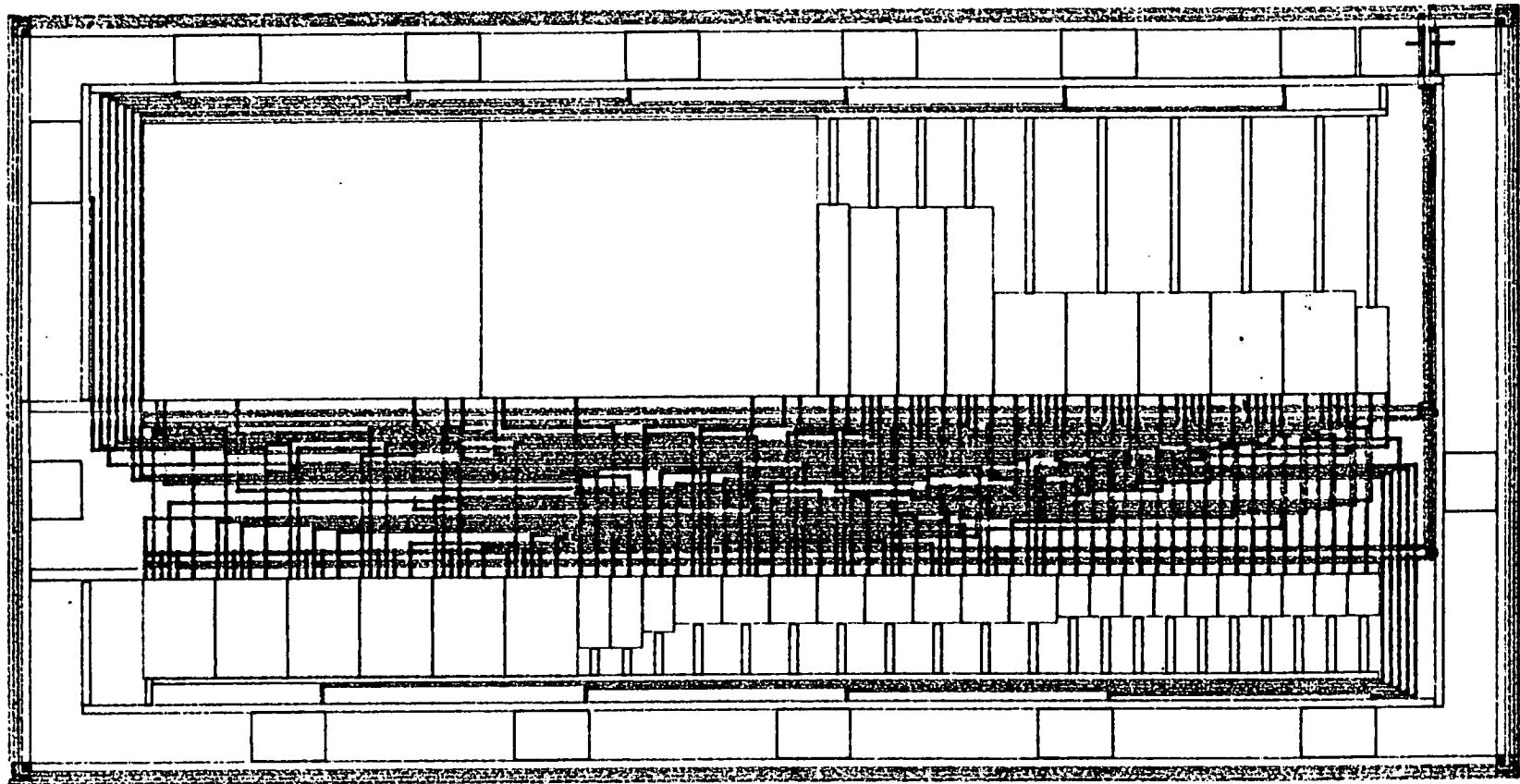


Figure II-2 : FIRST floorplan for universal adaptor

Appendix III

CANDIDATES PUBLISHED WORK

References

1. J. Mavor, H. M. Reekie, and N. Petrie, "Recent Developments in Wave Filters Implemented in MOS Technology," in *Design & Fabrication of Integrated Circuit Filters*, IEEE (September 1982).
2. N. Petrie, J. Mavor, and S. G. Smith, "General Purpose Adaptor Structure for Wave Digital Filter Realisation," *Electronics Letters*, Vol. 19, (24) pp. 1038-1039 (November 1983).
3. H. M. Reekie, J. Mavor, N. Petrie, and P. B. Denyer, "An Automated Design Procedure for Frequency Selective Wave Filters," pp. 258-261 in *ISCAS*, IEEE (May 1983).
4. H. M. Reekie, N. Petrie, and J. Mavor, "An Automated Design Procedure for Frequency Selective Wave Filters," in *Saraga Colloquium on Electronic Filters*, IEE (March 1983).
5. J. Mavor, N. Petrie, and C. H. Lau, "The Design of a Universal Wave Filter Adaptor using Dynamic CMOS Techniques," in *ESSCIRC 84*, (September 1984).
6. J. Mavor, N. Petrie, and H. M. Reekie, "Progress in Digital Wave Filter Design and Implementation," in *Digital Signal Processing - 84*, Elsevier Science Publications (September 1984).
7. H. M. Reekie, N. Petrie, J. Mavor, P. B. Denyer, and C. H. Lau, "Design and implementation of digital wave filters using universal adaptor structures," *Proc IEE Communications, Radar & Signal Processing*, Vol. 131, (6) pp. 615-622 (October 1984).

Design and implementation of digital wave filters using universal adaptor structures

H.M. Reekie, N. Petrie, J. Mavor, P.B. Denyer and C.H. Lau

Indexing terms: Filters and filtering, Digital circuits, Circuit theory and design

Abstract: The paper discusses progress on the topic of wave filters realised by a variety of approaches in digital technology and presents, in summary, an introduction to relevant theoretical concepts. The implementation of these filters is dependent on the efficient realisation of elements known as adaptors. This paper describes a universal adaptor concept which allows the realisation of either series or parallel adaptors. The Edinburgh FIRST silicon compiler has been used to design the universal adaptor structures, and example circuit layouts are given, in 5 µm NMOS technology, of filters with common frequency-selective characteristics. The future potential of the wave-filter approach to digital filtering is discussed, particularly with reference to CMOS integration.

1 Introduction

In any signal-processing system a requirement exists for a frequency-selective network, and this need was fulfilled in the past by RLC filters [1]. This type of filter has performed well for many years as it has significant advantages: it is passive and, therefore, guaranteed to be stable, it introduces a minimum of noise and when correctly designed it is insensitive to small variations in its component values. It has only one real disadvantage: it is physically large and heavy. This has not been a significant problem in the past because the systems in which these filters have been fitted have themselves been bulky. However, for integrated electronic systems the sheer size of the RLC filter has outweighed its many advantages, and other means have been developed to provide the frequency-selective function required.

Since the earliest days of the transistor active devices have been used to eliminate the need for inductors, which are generally the most difficult circuit elements to realise. However these early methods did not lend themselves to fabrication in integrated-circuit form, and different methods were developed to overcome the restrictions imposed by this construction medium. One of the most successful techniques used in recent years has been the switched-capacitor filter [2] which is now firmly established in the field of audio-frequency filtering. Though the initial theoretical background of these structures was poorly understood, more recent circuit theoretical developments [3] have permitted the design of excellent switched-capacitor filters. Unfortunately they suffer from one potential disadvantage which will become more of a problem as technology advances. Any system which employs analogue signals cannot take advantage of the reduction in the minimum on-chip dimensions that will be offered by future improved fabrication techniques. If an attempt is made to 'shrink' a switched-capacitor circuit (or any other analogue circuit) then increasing noise levels diminish the signal/noise ratio for very fundamental reasons [4]. Consequently most filter design aimed at systems of the future will be performed in the digital domain, and, certainly, this will be the case where electronically tunable characteristics are required.

Paper 3347F (E4, E10), first received 22nd February 1984 and in revised form 12th July 1984

The authors are with the Integrated Systems Group, Department of Electrical Engineering, University of Edinburgh, Kings Buildings, Mayfield Road, Edinburgh EH9 3JL, Scotland

IEE PROCEEDINGS, Vol. 131, Pt. F, No. 6, OCTOBER 1984

During the initial development of digital filters many different design techniques were used [5]. The first was the direct implementation of a filter transfer function using digital circuits, but this approach proved to be unsuitable as the resulting structures were extremely sensitive to small variations in their coefficients. The next method to be used was to split the filter into a cascade of first- and second-order sections, which greatly reduced the filter sensitivity to coefficient variation and allowed the design of fairly-high-order structures. It was not optimum by any standards, as this approach did not address problems which are unique to the digital environment; e.g. limit cycles, the effect of coefficient quantisation and finite signal word lengths. In an attempt to find a structure that was in some sense optimal, Fettweis [6] developed the wave-filter approach, in which all the electrical advantages of existing RLC filters were to be preserved in a digital structure. This was accomplished by developing a digital filter which was a direct transformation of a microwave reference filter. This reference filter is a standard, doubly terminated, commensurate microwave ladder network which, being passive, is guaranteed to be stable. Also, as it is a doubly terminated ladder structure, it is maximally insensitive to variations in its component values. The transformation used is exact, so the advantages of the continuous-time reference filter are retained in the digital structure. Historically, realisations of these structures using digital technology were referred to as wave digital filters (WDFs); however the theory that has been developed is independent of implementation. Thus here we refer to implementations of these filters using analogue techniques as analogue wave filters and using digital circuits as digital wave filters (synonymous with WDFs).

In this paper the elements of wave filter theory are summarised in Section 2, whereas implementation aspects and a review of practical filter realisations are given in Section 3.

2 Wave filter theory

A brief review of wave filter theory will be given here, but the reader is recommended to consult Reference 6 for further details.

The frequency variable used in the design of these filters is

$$\psi = \tanh(pT/2) \quad (1)$$

where T is the filter sample period, and p is the Laplace frequency variable [7]. By the above definition

$$\psi = (1 - e^{-pT})/(1 + e^{-pT}) \quad (2)$$

When an attempt is made to realise a digital structure based on this frequency variable, and using conventional voltages and currents as the digital variables, a set of equations will result which is unrealisable because of the presence of delay-free loops; i.e. the answer must be known before it may be calculated. This apparently insoluble problem is overcome when, instead of using V and I as variables, the wave variables A and B are used, where

$$A = V + RI \quad (3)$$

$$B = V - RI \quad (4)$$

where R is a positive constant known as the port reference resistance.

The variables A and B are found as linear combinations of V and I , the voltage across and the current through the component, respectively. However, A is in fact the wave incident on the component, and B is the wave reflected. The name generally associated with this type of filter (wave digital filter) arose from this fact.

2.1 Element realisation

All the normal components required in filter design can be realised as wave digital elements. It should be remembered that all the reference components are based on microwave structures and are thus distributed rather than lumped elements. Thus, for example, the actual representation of a capacitor is a section of open circuited transmission line.

2.1.1 Capacitors: The steady-state voltage/current relationship to be realised is

$$V = RI/\psi \quad (5)$$

where $R = 1/C$ = the port resistance. Using eqns. 3 and 4 this gives

$$B = AZ^{-1} \quad (6)$$

where $Z = e^{pT}$. These relationships are shown in Fig. 1.

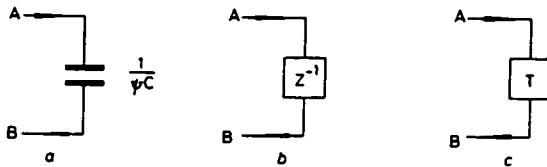


Fig. 1 Element realisation of capacitors

2.1.2 Inductors: The steady-state voltage/current relationships to be realised are

$$V = \psi RI \quad (7)$$

where R is the inductance value L . Application of eqns. 3 and 4 gives

$$B = -AZ^{-1} \quad (8)$$

These relationships are shown in Fig. 2.

2.1.3 Unit elements: The steady-state voltage/current relationships to be satisfied are

$$B_1 = A_2 e^{-pT/2} \quad (9)$$

$$B_2 = A_1 e^{-pT/2}$$

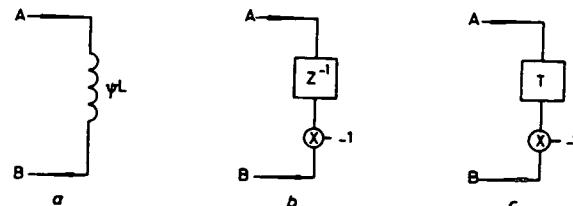


Fig. 2 Element realisation of inductors

where

$$A_k = V_k + RI_k$$

$$B_k = V_k - RI_k$$

$$k = 1, 2$$

(10)

The resulting wave structure is shown in Fig. 3c.

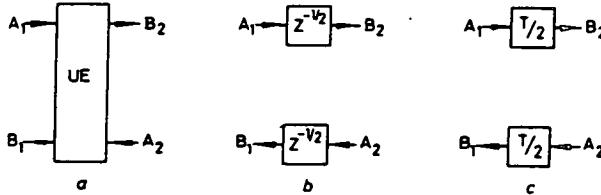


Fig. 3 Element realisation of unit elements

2.1.4 Other elements: The derivation of the wave equivalents for other circuit elements can be found in Fettweis' original paper [6]. Other possible reference elements include voltage sources, resistances, transformers, gyrators and circulators.

2.2 Interconnection of elements

The interconnection of elements is the heart of the problem in the realisation of a wave filter. This function is performed by adaptors which may be either of a series or a parallel type. First consider a parallel adaptor.

Consider an n -port network with port i having port reference resistance R_i . The ports are connected in parallel, so

$$V_1 = V_2 = V_3 = \dots = V_n$$

$$I_1 + I_2 + I_3 + \dots + I_n = 0$$

This information plus eqns. 3 and 4 with $k = 1, \dots, n$ gives

$$B_k = A_0 - A_k$$

where

$$A_0 = \sum_{k=1}^n \alpha_k A_k$$

$$\alpha = 2G_k/(G_1 + G_2 + \dots + G_n)$$

$$G_k = 1/R_k$$

(11)

Similar techniques give the equations for an n -port series adaptor and these are

$$B_k = A_k - \beta_k A_0$$

where

$$A_0 = \sum_{k=1}^n A_k$$

$$\beta_k = 2R_k/(R_1 + R_2 + \dots + R_n)$$

(12)

2.3 Wave filter networks

The circuit elements previously developed are interconnected using adaptors to give filter networks. An example of the transformation from a reference filter to a digital wave filter is shown in Fig. 4.

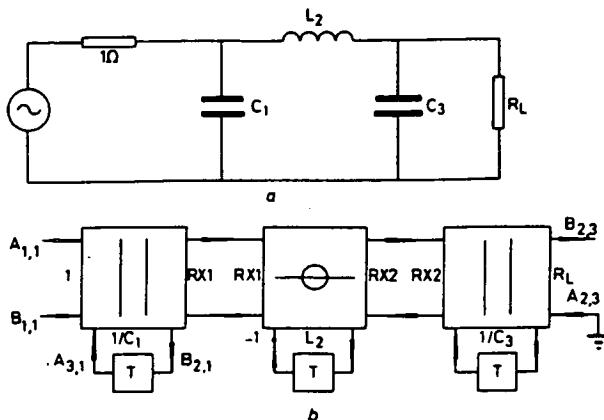


Fig. 4 Transformation of reference filter to wave filter

The input is $A_{1,1}$, and there are two outputs $B_{1,1}$ and $B_{2,3}$. The former represents the wave reflected on to (i.e. incident on) the source resistor R_s , and the latter represents the wave reflected onto the load resistor R_L . Hence $B_{2,3}$ is the 'normal' filter output (in this case lowpass), and $B_{1,1}$ is an extra output which is proportional to the inverse of $B_{2,3}$ and so is, in this case, a highpass output. All port resistances are marked on Fig. 4, but it should be noted that those marked $RX1$ and $RX2$ can be arbitrarily defined to be any positive constant. It should also be noted that at any given time no adaptor has more than two of its inputs known. As any adaptor output is normally determined from all three of its inputs, it would appear that an order in which the calculations are carried out cannot be found.

Consider the equations for adaptor 3:

$$B_k = \alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3 - A_k \quad k = 1, 2, 3 \quad (13)$$

If $\alpha = 1$ then

$$B_1 = \alpha_2 A_2 + \alpha_3 A_3$$

Now α_1 would be equal to one if

$$2G_1/(G_1 + G_2 + G_3) = 1$$

thus

$$G_1 = G_2 + G_3$$

$$1/RX2 = 1/R1 + C3$$

As $RX2$ can be defined arbitrarily, it can be set equal to this value, and $B_{1,3}$ can be calculated. By a similar argument applied to adaptor 2, it can be seen that if $RX1$ is defined as

$$RX1 = RX2 + L2$$

then

$$B_{1,2} = -A_{2,2} - A_{3,2} \quad (14)$$

which can be calculated from known data.

At this point all three inputs are known for adaptor 1 and all its outputs can be calculated. After performing this arithmetic enough information to calculate $B_{2,2}$ and $B_{3,2}$ is known. Lastly $B_{3,3}$ and $B_{2,3}$ may be calculated to give the filter output.

In one sample period five sets of sequential calculations have to be performed. For a filter of the form shown in Fig. 4, but containing m adaptors, $(2m - 1)$ sets of sequential equations would have to be calculated. This has a dramatic effect on the filter sample rate relative to the system clocking rate.

2.3.1 Half-synchronous wave filters: The problem discussed in the previous Section would be avoided if time delays could be inserted between each of the adaptors. Such a structure is known as a half-synchronous wave filter [8], and an example is shown in Fig. 5B. Inserting time delays in this way greatly affects the reference filter which is being modelled. In fact the wave filter shown in Fig. 5B is modelling the reference filter shown in Fig. 5A.

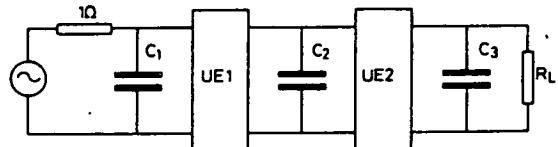


Fig. 5A Reference filter

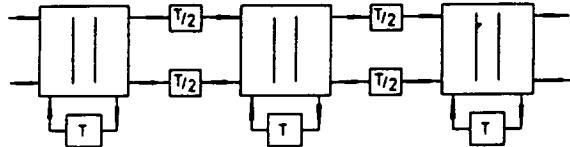


Fig. 5B Half-synchronous wave filter

The extra elements which have been inserted are unit elements, and they may be inserted when the reference filter is being designed in either of two ways: they can be added using the techniques described in Reference 9 (in which case they are redundant and do not add to the order of the filter), or the filter may be designed in such a way that they do contribute to its performance.

The graph shown in Fig. 6 illustrates the improvement

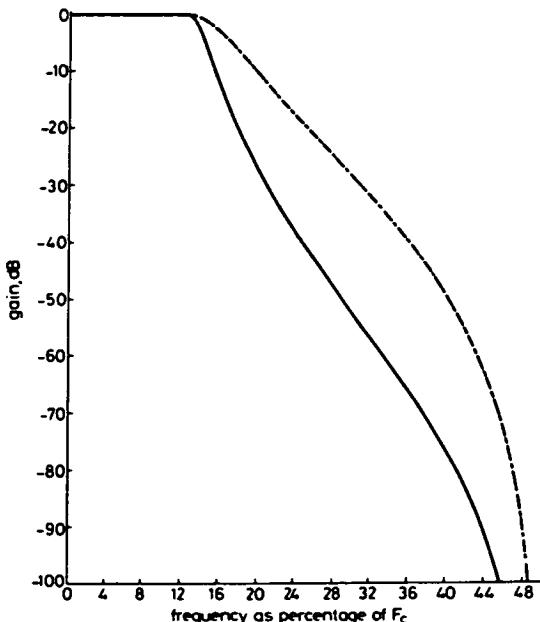


Fig. 6 Frequency response of filter of Figs. 5A and 5B

in performance that can be achieved by utilising the unit elements in the filter of Fig. 5A in such a way that they add to its order, in addition to providing the required segmentation effect. Clearly it is more efficient to synthesise the filter so that it does not include redundant unit elements.

A suitable transfer function for a bandpass-filter response which does not include redundant unit elements is

$$|S_{2,1}(\Omega^2)|^2 = 1/[1 + \epsilon^2 f^2(X, Y, Z)] \quad (15)$$

with

$$\begin{aligned} f(X, Y, Z) &= \cosh [N \cosh^{-1}(X) \\ &\quad + K \cosh^{-1}(Y) + L \cosh^{-1}(Z)] \\ X^2 &= [(1 + \Omega_2^2)(\Omega_1^2 - \Omega^2)]/[(\Omega_1^2 - \Omega_2^2)(1 + \Omega^2)] \\ Y^2 &= [\Omega_2^2(\Omega_1^2 - \Omega^2)]/[\Omega^2(\Omega_1^2 - \Omega_2^2)] \\ Z^2 &= (\Omega_1^2 - \Omega^2)/(\Omega_1^2 - \Omega_2^2) \\ \Omega_1 &= \tan(\phi_1) \\ \Omega_2 &= \tan(\phi_2) \end{aligned}$$

where N is the number of unit elements, K is the number of transmission zeros at DC, and L is the number of transmission zeros at infinity (Nyquist). ϕ_1 is the lower cutoff of the filter and ϕ_2 is the upper cutoff of the filter in degrees.

The filter synthesis following from this transfer function is performed with high-precision arithmetic but is otherwise quite conventional. Lowpass and highpass filter forms can be obtained from this synthesis procedure by setting $K = 0$ and $\phi_1 = 0^\circ$ for a lowpass filter, and $L = 0$ and $\phi_2 = 90^\circ$ for a highpass response.

Synthesis of the reference filter from this transfer function is a nontrivial task, and a computer program called SYNTH [10] has been written to do this automatically.

It can be seen from the transfer function given above that it is not possible to include zeros of transmission at frequencies other than DC or Nyquist. This difficulty can be overcome using the techniques detailed in the paper by Horton and Wenzel [11]. In this paper an iterative method is used to find a transfer function for lowpass and highpass filters which is equiripple in both the passband and the stopband.

A computer program has been written to perform the synthesis required to produce a filter which has zeros of transmission at finite frequencies, and an example filter is given in Fig. 7. The filter shown is a fifth-order Chebyshev

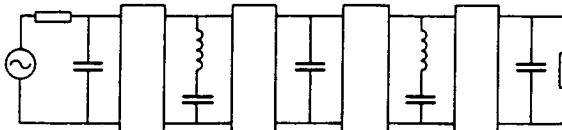


Fig. 7 Fifth-order lowpass filter

design with a passband ripple of 0.177 dB and a minimum stopband attenuation of 45 dB.

The computer program developed allows the very quick and accurate synthesis of some of the best known filters in such a way that they have topologies which allow them to be realised as half-synchronous wave filters. A half-synchronous wave filter is a structure which is inherently parallel in operation, and advantage of this fact must be taken in its implementation.

Deriving the wave filter from the reference filter is an extremely easy operation if the reference filter is of the

correct form. It is the implementation of the adaptor structures which gives rise to problems.

3 Implementation of wave filters

3.1 Review of implementations of wave filters

Since wave filters were first proposed several different approaches to their practical realisation have been reported. The most widely used form of wave filter has been the cascaded unit-element filter, in which the reference filter is composed of a series of unit elements with suitable characteristic impedances. As with conventional ladder filters, unit-element values can be determined from standard texts [12]. The wave filter derived from this type of structure is a cascade of two-port adaptors and has been implemented using both analogue and digital methods.

3.1.1 Analogue-wave filters: Analogue wave filters use switched-capacitor methods to implement wave filter adaptors and usually exploit the near perfect switches and capacitors formed in MOS technology. Signal values are determined by charge stored on capacitors. Within the adaptors addition is performed by charge redistribution between capacitors and is controlled by clocked switches. Multiplication uses similar principles, but the multiplier coefficient is determined by the ratio of the two capacitors involved. In the early work on analogue wave filters, buffers were used in preference to differential input operational amplifiers to maintain voltage levels within the system. However, as device sizes are reduced, these structures become more sensitive to integral stray capacitances, and the dynamic range is reduced. To avoid this problem, circuits using operational amplifiers have been proposed* which will allow stray-free wave-filter adaptors to be built.

Analogue techniques have been used in the past to implement adaptors, and fairly good results have been obtained [13], but it seems likely that, as integrated-circuit fabrication technology develops further, all-digital circuits will be the primary contenders for higher-order filtering operations. Lower-order filters may still be fabricated using analogue techniques, although it is most likely that switched-capacitor filters will dominate in this field. For this reason current research has focused almost entirely on digital wave filters.

3.1.2 Digital wave filters: In all digital signal-processing systems the method chosen to implement the required algorithms is important. Generally there are three approaches that can be used:

(i) A system can be custom designed to fit certain required conditions. This approach is expensive and time consuming although such custom designs in integrated-circuit form have been created for signal-processing systems.

(ii) General-purpose building blocks, such as DSP microprocessors, can be used to implement the system. This approach offers rapid system development, but for a commercial system this method of implementation is still bulky in terms of hardware.

(iii) Alternatively a method which combines aspects of both the methods mentioned above can be used. This is a semicustom technique in which predefined building blocks are used to implement the required signal-processing algorithm. Two examples of this method are the British Telecom FAD chip for digital filtering [14] and the FIRST silicon compiler [15] for more general DSP applications.

* TAYLOR, J.T.: Private communication

A further subdivision of methods arises when it is noted that digital circuits can be designed using either series or parallel arithmetic. The latter gives the fastest circuit operation, but the size and power consumption of the resulting chip weigh against it.

Digital wave filters have been realised using each of the different approaches mentioned above. However the wave filters have been based on a variety of reference filters. As in the analogue wave filter cascaded unit-element structures have been the most widely used reference, but, in contrast to the analogue domain where operational amplifiers, switches and capacitors are relatively compact, digital structures such as adders and multipliers require large areas of silicon. For this reason most of the initial work on practical wave filters [16, 17] used general-purpose structures such as multipliers, accumulators and memory chips for their implementation.

An alternative approach is to integrate the wave filter system into one or more chips. This can be achieved using distributed arithmetic methods [18] or by using conventional integrated-circuit 'building blocks' such as adders and multipliers. In the distributed arithmetic approach, extensive use of ROM is made to minimise the combinational logic required and offers high clocking rates as serial arithmetic is employed. It also has a highly regular compact structure which can be readily integrated. Comparisons of different types of wave filter implemented using distributed arithmetic methods can be found in Reference 19.

When the adaptor is integrated using standard cells the system can again offer high clock rates, but the system interconnection and control is reduced from that of the distributed arithmetic approach. Such an adaptor has been developed at Edinburgh University. In its design the following criteria had to be satisfied:

- (a) it was to be possible to configure the circuit as either a series or a parallel three-port adaptor
- (b) all the coefficients were to be externally programmable.

This generalised approach to the implementation implied that it would be possible to use one of these circuits whenever a wave filter called for a three-port (or two-port) adaptor. This rather simplistic approach would imply that each adaptor would work only on a 50% duty cycle. Also, since only a 5 μm NMOS process was available at the time, only one adaptor circuit could be placed on a chip, even when laid out by hand. These factors implied that, to realise a 2^m th-order filter, m chips would be required. Despite the problems the results obtained from these chips were very encouraging, and efforts were then made to overcome the limitations. A photograph of the general-purpose adaptor chip is given in Fig. 8. Fig. 9 shows the practical results for a third-order Chebyshev filter. The filter was designed to have a cutoff of 15% of the clock frequency, with a passband ripple of 0.5 dB.

These efforts led to the concept of a universal adaptor [20], which is actually a cascade of two adaptors each of which can be configured as either series or parallel. The two adaptors work in antiphase, and thus the circuit works with a 100% duty cycle. The universal adaptor is described in the next Section.

3.2 Universal adaptor chip

As in the case of the general-purpose adaptor there were three main points to consider in the design of a universal adaptor chip:

- (i) the chip had to be externally programmable to

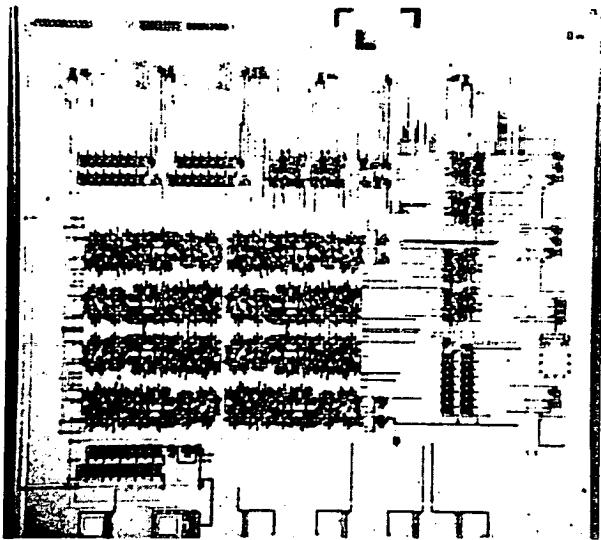


Fig. 8 Photograph of general purpose adaptor chip

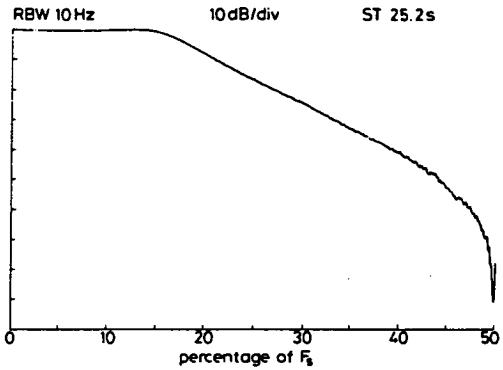


Fig. 9 Practical response of 3rd-order lowpass filter

realise parallel or series adaptors, with capacitive or inductive filtering elements

- (ii) the adaptor chips had to be directly cascadable to permit general-purpose filter design
- (iii) the structures had to be capable of operating in both periods of the sample clock; i.e. it had to have a 100% duty cycle.

A structure which satisfies these conditions is given in Fig. 10 and is shown symbolically in Fig. 11. In the adaptor structure extensive use of multiplexers has been made to direct the flow of data between suitable elements for implementing the parallel or series adaptor equations. Each of these multiplexers requires external signals to control this data flow. Multiplexers are also used to determine whether data is to be accepted from another universal adaptor, or whether the data is to be recycled during the second half of the sample period. Finally a multiplexer is used to select whether the adaptor has an inductive or capacitive filtering element on port 3. To simplify the testing of the adaptor structures, and to allow the construction of more complex wave filters, an extra multiplexer can be added to disconnect the B_3 output from the A_3 input and to allow external signals to be input to A_3 .

In the universal adaptor of Fig. 10, the multiplier coefficients used are $(\alpha_v - 1)$, and $(1 - \beta_v)$. It should be noted that this universal adaptor has been modified from that

originally published in Reference 20. The original adaptor structure was designed to use multiplier coefficients of

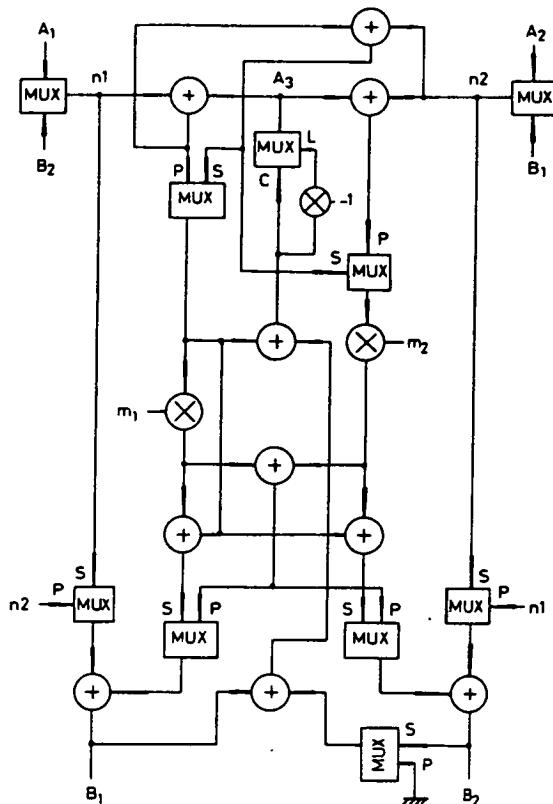


Fig. 10 Signal flowgraph of universal adaptor system

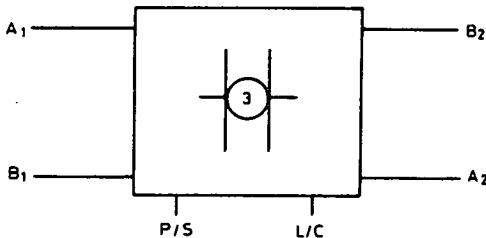


Fig. 11 Universal adaptor symbol

$(\alpha - 1)$ and $-\beta$. However, this was found to be unsuitable for integration using the available CAD tools (the FIRST silicon compiler) as series adaptors which had coefficients with a modulus greater than one could not have been realised. If this restriction had been accepted it would have implied that certain bandpass structures could not have been implemented. The method now used simplifies the system timing and provides a closed number system for multiplier operation; i.e. the range of the output signal will be in the same range as the input signal. From eqns. 11 and 12 it is clear that $(\alpha_v - 1)$ and $(1 - \beta_v)$ will always lie between plus and minus one. The new adaptor structure contains two extra adders, and the value of the β coefficients has been modified. These changes allow the new adaptor to deal with any possible wave filter coefficient values. The system timing is identical in both this and the previously published structures.

3.2.1 Adaptor latency and sample rate: The latency of any digital structure is determined by the total delay of all the arithmetic operators in the signal path through the structure. In the adaptor structure of Fig. 10 the latency can be determined by following the signal path from either of the input nodes, A₁ or A₂, through to the appropriate output node. In the adaptor each adder (or subtractor) has a latency of one cycle. Similarly each multiplexer contributes one cycle to the total delay. As there are three adders and three multiplexers in the path from A₁ to B₁, these structures will add six cycles to the system latency. Using the modified Booth multiplier algorithm, the latency of the multiplier is dependent on the coefficient word length and is given by

$$L_{\text{mult}} = (3/2) * m + 2 \quad (16)$$

where m is the coefficient word length. In addition, extra bit-delays are often required to synchronise signals coming from different parts of the circuit. These also add to the system latency. In the adaptor structure two bit delays are required for synchronisation. Therefore, for an adaptor system using a multiplier coefficient of length m , the latency of the structure will be

$$L_{\text{ad}} = (3/2) * m + 10 \text{ cycles} \quad (17)$$

As the adaptor system is used twice in every sampling period, the total period between samples is $2 * L_{\text{ad}}$ cycles. Knowing the adaptor latency L_{ad} and the process clock rate r_p , the maximum sample rate f_s can easily be deduced:

$$f_s = r_p / (2 * L_{\text{ad}}) \quad (18)$$

As an example, if we consider a filter network with a system clock of 8 MHz and a multiplier with 8-bit coefficients, the sample rate will be approximately 180 kHz. This assumes that no further multiplexing of the adaptors takes place. If the universal adaptor is to be multiplexed n times, then the overall sample rate will reduce to f_s/n .

3.2.2 Word growth and signal word lengths: To achieve a completely general-purpose condition, each adaptor in the filter system must be capable of suppressing word growth, which can lead to parasitic oscillations or instability. Within the adaptor system word growth can occur in the adder and subtractor elements. However this effect can be minimised by increasing the number of guard bits in the system word length SWL. The system word length is determined by the total latency of the adaptor structure. The number of guard bits required can be determined from an analysis of the longest signal path in the adaptor structure. The longest signal loop in the universal adaptor structure is for the case of a series adaptor connection with a capacitive element connected to the third port. In this path there are six adders, one multiplier and one subtractor. Adders can contribute to signal word growth, so one additional guard bit must be allowed for each of them. The sign inversion is provided by a subtractor element which subtracts the signal from zero. Therefore no word growth can occur from that element. Finally, to ensure correct multiplier operation, two further guard bits are necessary, although these are not affected by the multiplier and appear unchanged. Consequently the total number of guard bits required in the system is six. Therefore, to allow a small permissible signal growth in the loop, the signal word length must be limited to a maximum of (SWL - 6) bits. For the case where an 8-bit multiplier coefficient is being used, the maximum permissible signal anywhere in the filter system is $2^{17} - 1$.

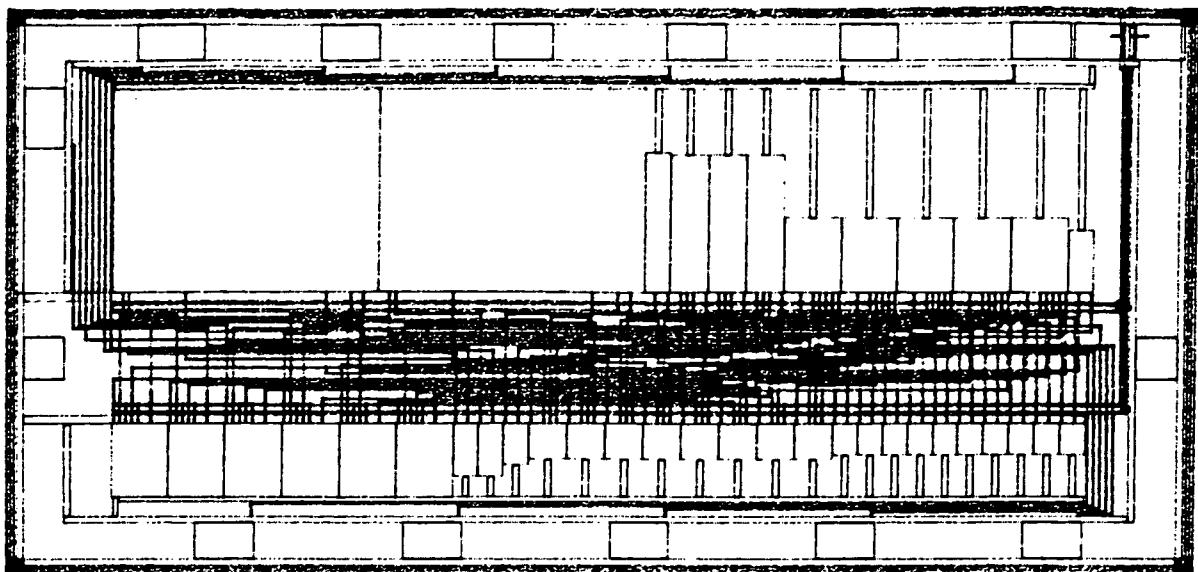


Fig. 12 FIRST floorplan for universal adaptor

4 FIRST silicon compiler

4.1 Introduction

A silicon compiler is a rapid method of implementing a particular system or algorithm on a silicon chip and is used to convert a language description of the required system into a silicon design using certain rules about placement and routing. A comprehensive description of silicon compilers can be found in Reference 21.

The FIRST silicon compiler, developed at Edinburgh University, uses a bit-serial architecture, as proposed by Lyon [22]. Using this method data flows serially through the system, with the least significant bit first. Because the data only moves by 1 bit in every clock cycle, the amount of hardware required can be significantly reduced. Similarly, the interconnection between individual elements in the system is dramatically reduced.

FIRST uses a system description of a signal-flow diagram as its input. From this the compiler generates suitable simulation data and provides an automated layout of the desired system. The signal flow graph shown in Fig. 10 represents a universal adaptor, and the corresponding layout, as generated by FIRST, can be seen in Fig. 12. All system routing is performed in one central routing channel, and, for this reason, FIRST compiled chips tend to have aspect ratios greater than unity.

4.2 Filter system hardware

Digital wave filters can consist of a regular array of universal adaptors, with additional multiplexing of adaptors possible. Clearly, for any given filter order n , the structure may be implemented using $n/2$ universal adaptors or one universal adaptor multiplexed $n/2$ times. One disadvantage of using the multiplexed adaptor is the requirement to store intermediate results in RAM during the computational cycle. However the major drawback is the corresponding $n/2$ reduction in sample rate that such a method would require.

External control of the filter system can be arranged easily by using conventional methods such as EPROMs or RAM. In a dedicated wave filter chip the control requirement could be built into the chip design in the form of RAM or ROM.

4.2.1 Example: As an illustration of wave-filter implementation using universal adaptors, a seventh-order lowpass filter has been chosen. The example filter has a Chebyshev response with a passband ripple of 0.644 dB and a designed cutoff frequency of 20% of the sample clockrate. The reference filter has been designed using the synthesis procedure described in Section 2, and the resulting structure of the ladder type can be seen in Fig. 13.

If the filter of Fig. 13 was transformed directly into a conventional wave-filter structure, four parallel adaptors would be required as shown in Fig. 13. However, to ensure the correct timing for the system, the odd numbered adaptors would have to operate in the first half of the sample period, and the even adaptors would work in the second half of the sample period. Clearly this is inefficient in hardware, and the universal adaptor flow diagram of Fig. 10 has been designed to use the available hardware during both halfcycles of the sample period. Thus, only

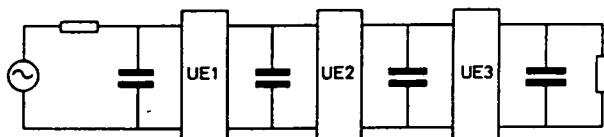


Fig. 13A Seventh-order lowpass filter

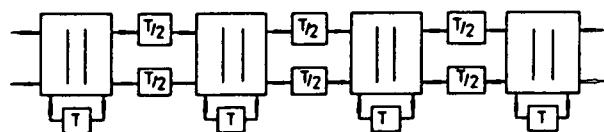


Fig. 13B Implementation using conventional adaptors

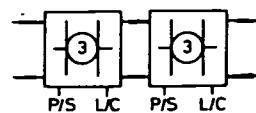


Fig. 13C Implementation using universal adaptors

two universal adaptors will be required to implement the seventh-order example filter.

External control of the filter is greatly simplified because all the adaptors to be implemented are parallel, and all the port 3 connections are capacitors. However, for the situation where a bandpass structure was being realised, the parallel/series (P/S) and inductor/capacitor (L/C) control pins of the adaptor could be programmed as required with reference to Fig. 11.

For the example filter, eight separate multiplier coefficients are required. These are evaluated as $(\alpha - 1)$ for each particular parallel adaptor, and they are used by the universal adaptors in the same order as they would have been used in the filter of Fig. 13.

The filter has been simulated with different coefficient word lengths of 8 and 12 bits, and results confirming the insensitivity of wave filters to coefficient word length can be seen in Fig. 14.

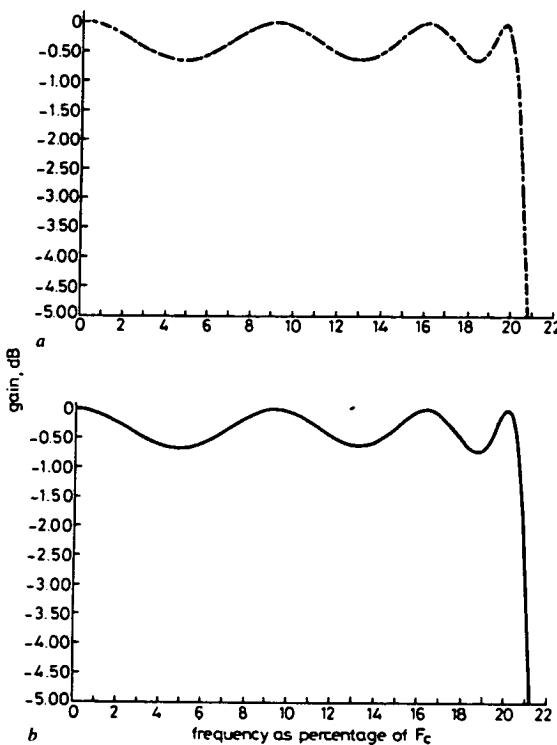


Fig. 14 Simulation results using different coefficient word lengths
a 12 bit coefficient
b 8 bit coefficient

5 Conclusions and future potential of wave filters

This paper has shown how efficient digital realisations of classical LC filters can be achieved. Synthesis techniques have been developed to remove redundancy in reference filter structures, offering a reduction in hardware requirements. Extensive use of computers has allowed the development of an automated procedure for designing and simulating digital wave filters quickly and easily. The combination of the synthesis techniques and the development of the universal adaptor has allowed the construction of half-synchronous wave filters based on reference filters which do not contain redundant unit elements. This, together with the fact that a universal adaptor works with a 100%

duty cycle, results in an almost fourfold reduction in hardware compared to a conventional nonoptimised design.

As an example of this improvement an integrated circuit, implemented using $2.5 \mu\text{m}$ linewidths in CMOS technology, is predicted to contain four independent universal adaptors. Such a device could offer any filter combination between one 15th-order filter and four separate second-order filters. For this approach, no time-division multiplexing would be necessary.

We believe that the automated design approach and the universal adaptor concept makes a significant contribution to the development of frequency-selective filters for use in digital signal-processor systems.

6 Acknowledgments

The authors acknowledge the support of the UK Science & Engineering Research Council in funding this research.

7 References

- TEMES, G.C., and LAPATRA, J.W.: 'Introduction to circuit synthesis and design' (McGraw-Hill, 1977)
- JACOBS, G.M., ALLSTOT, D.J., BRODERSEN, R.W., and GRAY, P.R.: 'Design techniques for MOS switched capacitor ladder filters', *IEEE Trans.*, 1978, **CAS-25**, pp. 1014-1021
- SCANLAN, S.O.: 'Analysis and synthesis of switched capacitor state variable filters', *ibid.*, 1981, **CAS-28**, pp. 85-92
- HSU, S.T.: 'Surface-related 1/f noise in MOS transistors', *Solid State Electron.*, 1970, **13**, pp. 1451-1459
- RABINER, L.R., and GOLD, B.: 'Theory and application of digital signal processing' (Prentice-Hall, 1975)
- FETTWEIS, A.: 'Digital filter structures related to classical filter networks', *Arch. Elect. Übertragungstech.*, 1971, **25**, pp. 79-89
- RICHARDS, P.I.: 'Resistor-transmission-line circuits', *Proc. IRE*, 1948, **37**, pp. 217-220
- FETTWEIS, A., and MEERKOTTER, K.: 'Suppression of parasitic oscillations in half-synchronous wave digital filters', *IEEE Trans.*, **CAS-23**, pp. 125-126
- FRAITURE, L., and NEIRYNCK, J.J.: 'Theory of unit element filters', *Rev. HF*, 1969, **7**, pp. 325-340
- REEKIE, H.M., MAVOR, J., PETRIE, N., and DENYER, P.B.: 'An automated design procedure for frequency selective wave filters', *ISCAS, IEEE*, May 1983, pp. 258-261
- HORTON, M.C., and WENZEL, R.J.: 'Realisation of microwave filters with equal ripple response in both pass and stop bands', Symposium on generalised networks, Polytechnic Institute of Brooklyn, April 1966, pp. 257-288
- LEVY, R.: 'Tables of element values for the distributed low-pass prototype filter', *IEEE Trans.*, 1965, **MTT-13**, pp. 514-536
- MAVOR, J., REEKIE, H.M., DENYER, P.B., SCANLAN, S.O., CURRAN, T., and FARRAG, A.: 'A prototype switched-capacitor voltage wave filter realised in NMOS technology', *IEEE J. Solid State Circuits*, 1981, **SC-16**, pp. 716-723
- ADAMS, P.F., HARBIDGE, J.R., and MACMILLAN, R.H.: 'A MOS integrated circuit for digital filtering and level detection', *ibid.*, 1981, **SC-16**, pp. 183-190
- DENYER, P.B., RENSHAW, D., and BERGMANN, N.: 'A silicon compiler for VLSI signal processors', *ESSCIRC '82*, Vrije Universiteit, Brussels, September 1982, pp. 215-218
- LAWSON, S.S.: 'On the hardware implementation of a multiplexable wave digital filter', *ISCAS, IEEE*, June 1980, pp. 124-127
- LAWSON, S.S.: 'A wave digital filter hardware structure', *Proc. IEE*, 1981, **128**, (6), pp. 307-312
- WANHAMMAR, L.: 'Implementation of wave digital filters using distributed arithmetic', *Signal Process.*, 1980, **2**, pp. 253-260
- WANHAMMAR, L.: 'A comparison of wave digital lowpass filters from an implementation point of view', *ISCAS, IEEE*, May 1982, pp. 776-779
- PETRIE, N., MAVOR, J., and SMITH, S.G.: 'General purpose adaptor structure for wave digital filter realisation', *Electron. Lett.*, 1983, **19**, (24), pp. 1038-1039
- AYRES, R.F.: 'VLSI: silicon compilation and the art of automatic microchip design' (Prentice-Hall, 1983)
- LYON, R.F.: 'A bit-serial VLSI architectural methodology for signal processing', *VLSI 81*, August 1981, pp. 131-140

GENERAL-PURPOSE ADAPTOR STRUCTURE FOR WAVE-DIGITAL-FILTER REALISATION

Indexing terms: Filters, Digital filters, Wave filters

A general purpose wave-digital-filter adaptor is presented which is capable of operating in a parallel or series configuration and can implement inductive or capacitive filtering elements. The structure can be multiplexed to increase the filter order for no significant increase in hardware.

Universal adaptor concept: Wave filters, as proposed by Fettweis,¹ are known to possess certain advantages over other types of digital filter, including insensitivity to small variations in coefficients and suppression of parasitic oscillations. Implementations of wave filters require systems which are half-synchronous, i.e. adaptors which operate on a two-phase timing sequence, with the odd and even adaptors operating on different time periods. Clearly, this requires hardware to be 'idle' for half the time, and is therefore inefficient. If an adaptor structure could be multiplexed, as in the case of cascaded unit-element filters,² then the required hardware can be minimised. However, for reference filters containing inductors, capacitors and unit elements, no general structure can be guaranteed, so there is a requirement for an adaptor structure that can be used as either a parallel or series adaptor and can simulate inductors or capacitors. This 'universal adaptor' would be suitable as a multiplexed structure provided the analogue reference filter contained unit elements between each of the filtering elements. This can be achieved using either Kuroda transforms or the synthesis method proposed by Reekie *et al.*³

Such an adaptor structure would have to satisfy the following equations for parallel and series adaptors:

Parallel adaptors:

$$B_k = A_0 - A_k$$

$$A_0 = 2A_n + \sum_{k=1}^{n-1} \alpha_k(A_k - A_n)$$

Series adaptors:

$$B_k = A_k - \beta_k A_0$$

$$B_n = -A_0 - \sum_{k=1}^{n-1} B_k$$

$$A_0 = \sum_{k=1}^n A_k$$

where n is the dependent port.

Universal adaptor realisation: The three-port adaptor shown symbolically in Fig. 1 represents a general-purpose wave-filter adaptor that can be configured to operate as a parallel or series adaptor, with either an inductive or capacitive termination at port 3. The signal flow diagram of the adaptor, shown in Fig. 2, is controlled using multiplexers (MUX) to direct the data along the series (S) or parallel (P) adaptor paths. These multiplexers determine whether the adaptor is implementing parallel or series adaptor equations and the type of termination connected to port 3. Control of these multiplexers is

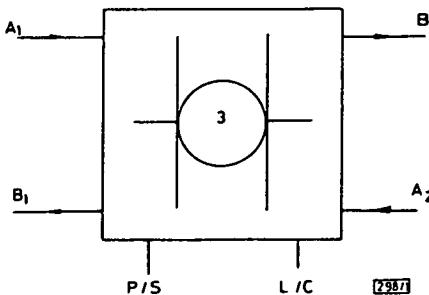


Fig. 1 Symbolic representation of universal adaptor

determined externally, as shown in Fig. 2. The structure can only simulate inductors (L) or capacitors (C), but the addition of a single multiplexer would enable separate signals to be injected into node A_3 for testing purposes, or to enable structures which implement zeros of transmission at finite frequencies to be designed. Time delays for the L or C terminations are provided internally, using the latency of the adaptor ($T/2$) and the latency of the control circuitry with additional delay to satisfy the T delay required. Additional multiplexers are available at the A_1 and A_2 signal inputs. These enable the outputs B_2 and B_1 to be made accessible internally so the structure can be used for multiplexed adaptor filters.

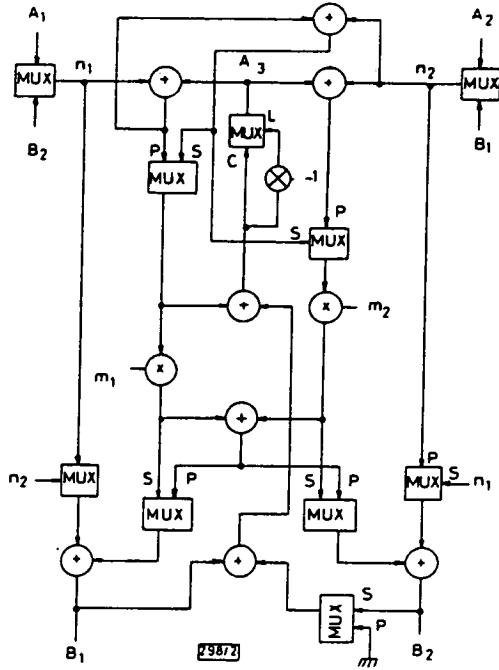


Fig. 2 Signal flow diagram for universal adaptor

Potential of approach: The structure is suitable for either parallel or bit-serial implementation, although only the serial form has been investigated. As with all bit-serial systems, the latency of the system determines the sample rate F_s . The latency of adders and multiplexers is equal, so an extra adder has been added to the minimised hardware to reduce the system latency and the complexity of the adaptor structure. The multiplier coefficients used in the structure, m_1 and m_2 , are determined in the usual manner,¹ but for parallel adaptors $\alpha_k - 1$ is used, and for series adaptors $-\beta_k$ is used. The overall latency of the adaptor is controlled by the type of digital multiplier employed: for a modified Booth algorithm multiplier,⁴ with an 8-bit coefficient, the system latency will be 21 cycles, allowing a sample rate of up to

$$T_s = \frac{\text{system clock rate}}{2 \times 21 \text{ cycles}}$$

depending on the level of multiplexing of the adaptor.

Use of the multiplexers at the A_1 and A_2 inputs means that we can use the available hardware to implement both odd- and even-numbered adaptors and, therefore, double the filter order for the given hardware and sample rate conditions.

Using the synthesis method proposed by Reekie *et al.*³ a seventh-order filter could be achieved using two universal adaptors utilising two inductors (or capacitors), and two unit elements per adaptor. Such a filter has been simulated and the results, given as the solid line in Fig. 3 agree well with the theoretical response (broken line). Also, because the adaptor structures can be cascaded, any order of filter can be achieved using either several universal adaptors or one adaptor multiplexed several times. For example, an integrated circuit of typical size, using $2.5 \mu\text{m}$ linewidths in either NMOS or

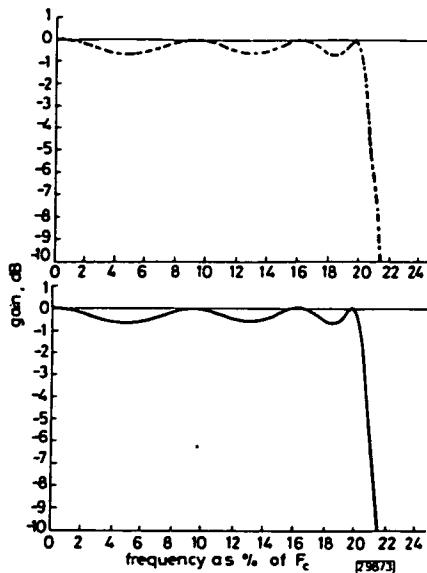


Fig. 3 Simulation results for 7th-order lowpass filter

CMOS technology, could contain four independent universal adaptors, offering any filter combination between one 15th-order filter and four separate second-order filters, with no time-division-multiplexing. We believe that this universal adaptor development is a significant advance in the practical realisation of precision digital filters.

Acknowledgments: The authors acknowledge the UK Science & Engineering Research Council and the contribution of Dr. H.M. Reekie in developing the wave-filter theory.

N. PETRIE
J. MAVOR
S. G. SMITH

14th October 1983

*Integrated Systems Group
Department of Electrical Engineering
University of Edinburgh
Edinburgh EH9 3JL, Scotland*

References

- 1 FETTWEIS, A.: 'Digital filter structures related to classical filter networks'. *Arch. Elect. Übertrag.*, 1971, **25**, pp. 79-89
- 2 LAWSON, S. S.: 'Implementation of wave digital filter structures'. *Proc. IEE*, 1981, **128**, pp. 224-226
- 3 REEKIE, H. M., MAVOR, J., PETRIE, N., and DENYER, P. B.: 'An automated design procedure for frequency selective wave filters'. *ISCAS, IEEE*, May 1983, pp. 258-261
- 4 RUBINFIELD, L. P.: 'A proof of the modified Booth's algorithm for multiplication'. *IEEE Trans.*, 1975, **C-24**, pp. 1014-1015

APPENDIX IV

BIBLIOGRAPHY

References

1. G. Raisbeck, "A Definition of Passive Linear Networks in terms of Time and Energy," *J. Applied Physics*, (25) pp. 1510-1514 (December 1954).
2. F. F. Kuo, *Network Analysis and Synthesis*, Wiley, New York (1962).
3. J. O. Scanlan and R. Levy, *Circuit Theory*, Oliver and Boyd, Edinburgh (1973).
4. G. C. Temes and S. K. Mitra, *Modern Filter Theory and Design*, Wiley, New York (1973).
5. Rudolph Saal, *Handbook of Filter Design*, AEG-Telefunken (1979).
6. R. P. Sallen and E. L. Key, "A Practical Method of Designing RC Active Filters," *IRE Trans. Circuit Theory*, Vol. CT-2, pp. 74-85 (March 1955).
7. A. Fettweis, "Digital Filter Structures Related to Classical Filter Networks," *Arch. Elect. Übertrag.*, Vol. 25, (2) pp. 79-89 (February 1971).
8. G. C. Temes and J. W. LaPatra, *Introduction to Circuit Synthesis and Design*, McGraw-Hill (1977).
9. P. I. Richards, "Resistor-transmission-line circuits," *Proc. IRE*, Vol. 37, pp. 217-220 (February 1948).
10. C. E. Shannon, "Communication in the Presence of Noise," *Proc. IRE*, pp. 10-22 (January 1949).
11. C. M. Rader and B. Gold, "Digital Filter Design Techniques in the Frequency Domain," *Proc. IEEE*, Vol. 55, (2) pp. 149-171 (1967).
12. D. L. Fried, "Analog Sampled-Data Filters," *IEEE Journal of Solid-State Circuits*, pp. 302-304 (August 1972).

13. K. Martin, "Improved Circuits for the Realisation of Switched Capacitor Filters," *Trans. IEEE Circuits and Systems*, Vol. CAS-27, (4) pp. 237-244 (April 1980).
14. J. T. Taylor and J. Mavor, "Exact design of stray-insensitive switched-capacitor lowpass ladder filters from unit-element tables," *Electronics Letters*, Vol. 20, (16) pp. 665-667 (August 1984).
15. D. J. Allstot, R. W. Broderson, and P. R. Gray, "An Electrically Programmable Switched Capacitor Filter," *IEEE Journal of Solid State Circuits*, Vol. SC-16, pp. 1034-1041 (September 1979).
16. R. E. Bogner and A. G. Constantinides, *Introduction to Digital Filtering*, Wiley (1975).
17. H. M. Reekie, *Monolithic Frequency Filter Designs Based on a Sampled Data Analogue Wave Filter.*, University of Edinburgh (October 1981). Ph.D. Thesis
18. S. S. Lawson, "Computer Simulation and Implementation of a Wave Digital Filter," in *Colloquium on Electronic Filters*, IEE Group E10 (May 1977).
19. L. T. Bruton, "Network Transfer Functions Using the Concept of Frequency-Dependent Negative Resistances," *Trans. IEEE Circuit Theory*, Vol. SC-16, pp. 406-408 (August 1969).
20. L. Fraiture and J. J. Neirynck, "Theory of Unit Element Filters," *Rev. H. F.*, Vol. 7, (12) pp. 325-340 (1969).
21. J. D. Rhodes, *Theory of Electrical Filters*, Wiley, London (1976).
22. A. Fettweis, "Canonic Realization of Ladder Wave Digital Filters," *Circuit Theory and Applications*, Vol. 3, pp. 321-332 (1975).
23. H. M. Reekie, J. Mavor, N. Petrie, and P. B. Denyer, "An Automated Design Procedure for Frequency Selective Wave Filters..," pp. 258-261 in *ISCAS*, IEEE (May 1983).
24. A. Fettweis, H. Levin, and A. Sedlemeyer, "Wave Digital Lattice Filters," *Int. Journal Circuit Theory*, Vol. 2, pp. 203-211 (June 1974).

25. J. Mavor, H. M. Reekie, P. B. Denyer, J. O. Scanlan, T. Curran, and A. Farrag, "Switched-Capacitor Voltage-Wave Filter," pp. 122-123 in 1981 IEEE International Solid-State Circuits Conference, Lewis Winner, New York (February 1981).
26. H. M. Reekie, J. Mavor, P. B. Denyer, J. O. Scanlan, T. Curran, and A. Farrag, "Realisation of a switched-capacitor voltage wave filter," pp. 5/1-5/4 in Colloquium on Electronic Filters, IEE Group E10, London (April 1981).
27. J. Mavor, H. M. Reekie, P. B. Denyer, S. O. Scanlan, T. Curran, and A. Farrag, "A Prototype Switched-Capacitor Voltage Wave Filter Realized in NMOS Technology," IEEE Journal of Solid State Circuits, Vol. SC-16, (6) pp. 716-723 (December 1981).
28. H. A. Rafat, *Design and Implementation of Switched-Capacitor Frequency-Selective Filters in MOS Technology*, University of Edinburgh (February 1983). Ph.D. Thesis
29. P. D. James, *A Stray Insensitive Analogue Implementation of a Three-Port Parallel Wave Filter Adaptor*, B.Sc. Honours Report HSP382, University of Edinburgh May 1984.
30. Y. Tsividis and K. Fang, *SWITCAP Users Guide*, Columbia University, New York
31. E. K. Cheng, "Problems and Promise of Analog MOS/LSI," *Lambda*, pp. 60-63 (1980). Fourth quarter
32. S. S. Lawson, "The hardware implementation of a multiplexable wave digital filter structure," pp. 6/1-6/6 in Colloquium on Electronic Filters, IEE Group E10 (May 1980).
33. S. S. Lawson, "A Wave Digital Filter Hardware Structure," *Proc. IEE*, Vol. 128, (6) pp. 307-312 (December 1981). Part G
34. S. S. Lawson and S. Terplan, "High-speed single-board flexible hardware implementation of a wave digital filter," *IEE Proceedings - Part G.*, Vol. 131, pp. 17-23 (February 1984).
35. M. Renfors and E. Zigouris, "Signal Processor Implementation of Lattice Wave Digital Filters," pp. 233-237 in *Digital Signal Processing - 84*, Elsevier Science Publishers B. V. (September 1984).

36. L. Gazsi, "Single Chip Filter Bank with Wave Digital Filters," *IEEE Trans. Acoust. Speech & Signal Processing*, Vol. ASSP-30, pp. 709-718 (October 1982).
37. B. Sikstrom, "Some Aspects of the LSI-Implementation of Wave Digital Filters," pp. 780-783 in *ISCAS*, IEEE (May 1982).
38. J. Van Ginderdeuren, H. De Man, F. Catthoor, and S. Beckers, "A Design Methodology for Compact Integration of Wave Digital Filters," pp. 210-213 in *ESSCIRC '84*, University of Edinburgh (September 1984).
39. S. Matsumura, M. Renfors, B. Sikstrom, and L. Wanhammar, "LSI Implementation of Lattice Wave Digital Filters," pp. 303-307 in *Digital Signal Processing - 84*, Elsevier Science Publications (September 1984).
40. H. De Man, C. Vanderbulcke, and M. Van Cappelen, "High Speed NMOS Circuits for ROM Accumulator and Multiplier-Type Digital Filters," pp. 200-201 in *Proc. ISSCC*, IEEE (February 1978).
41. L. J. M. Claesen, H. J. De Man, and J. Vanderwalle, "DIGEST: A Digital Filter Evaluation and Simulation Tool for MOSVLSI Filter Implementations," *IEEE Journal of Solid State Circuits*, Vol. SC-19, pp. 414-424 (June 1984).
42. A. Peled, "On the Hardware Implementation of Digital Signal Processors," *IEEE Trans. Acoustics Speech and Signal Processing*, Vol. ASSP-24, (1) pp. 76-86 (February 1976).
43. R. F. Lyon, "Two's Complement Pipeline Multipliers," *IEEE Trans. Communications*, Vol. COM-24, pp. 418-425 (April 1976).
44. D. J. Myers, *Multipliers for LSI and VLSI Signal Processing Applications*, University of Edinburgh (September 1981). M.Sc. Thesis
45. L. Wanhammar, "Implementation of Wave Digital Filters using Distributed Arithmetic," *Signal Processing*, Vol. 2, (3) pp. 253-260 (July 1980).
46. C. S. McIntosh, *Frequency Dependent Negative Resistances (FDNRs) applied to Wave Digital Filters*, B.Sc. Honours Report HSP341, University of Edinburgh May 1983.

47. *DINAP Users Manual*, Circuits and Systems Group, School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907, USA
48. A. Fettweis and K. Meerkotter, "Suppression of Parasitic Oscillations in Half-Synchronous Wave Digital Filters," *Trans. IEEE*, Vol. CAS-23, pp. 125-126 (February 1976).
49. A. Fettweis and K. Meerkotter, "Suppression of Parasitic Oscillations in Wave Digital Filters," *IEEE Trans. Circuits and Systems*, Vol. CAS-22, pp. 239-246 (March 1975).
50. Stanley L. Freeny, "Special-Purpose Hardware for Digital Filtering," *Proc. IEEE*, Vol. 63, (4) pp. 633-648 (April 1975).
51. T. J. Terrell, *Introduction to Digital Filters*, Macmillan Press, London (1980).
52. R. F. Lyon, "A Bit-Serial VLSI Architectural Methodology for Signal Processing," pp. 131-140 in *VLSI 81*, (August 1981).
53. M. Renfors and Y. Neuvo, "The Maximum Sample Rate of Digital Filters under Hardware Speed Restrictions," *Trans. IEEE Circuits & Systems*, Vol. CAS-28, pp. 196-202 (March 1981).
54. R. F. Ayres, *VLSI: Silicon Compilation and the art of Automatic Microchip Design*, Prentice-Hall (1983).
55. P. B. Denyer, D. Renshaw, and N. Bergmann, "A Silicon Compiler for VLSI Signal Processors," pp. 215-218 in *ESSCIRC '82*, Vrije Universiteit, Brussel (September 1982).
56. D. Renshaw, "On Using The FIRST Software Package," University of Edinburgh (1983).
57. L. P. Rubinfield, "A Proof of the Modified Booth's Algorithm for Multiplication," *IEEE Transactions on Computers*, Vol. C-24, pp. 1014-1015 (October 1975).
58. J. Mavor, M. A. Jack, and P. B. Denyer, *Introduction to MOS LSI Design*, Addison-Wesley (1983).
59. C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley (1980).

60. D. J. Myers and P. A. Ivey, "Circuit elements for VLSI signal processing," *British Telecom Technology Journal*, Vol. Volume 2, pp. 67-78 (July 1984).
61. A. D. Booth, "A Signed Binary Multiplication Technique," *Journal of Mech. & Appl Math.*, Vol. 4, (1951).
62. J. N. Holmes, "The JSRU channel vocoder," *IEE Proceedings, Part F*, Vol. 127, pp. 53-60 (February 1980).
63. J. A. C. Bingham, "A new type of Digital Filter with a Reciprocal Ladder Configuration," pp. 129-130 in *Proc. IEEE Int. Symp. Circuit Theory*, Lewis Winner (December 1970).
64. A. Fettweis, "Wave Digital Filters with Reduced Number of Delays," *Int. Journal of Circuit Theory & Application*, Vol. 2, pp. 319-330 (December 1974).