1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service / Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

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

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

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

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

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

AVIS

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

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

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

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

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

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

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

National Library
of Canada

Canadian Theses
on Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Services des thèses canadiennes
sur microfiches

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

Please print or type — Ecrire en lettres moulees ou dactylographier

Full Name of Author — Nom complet de l'auteur

IRENA STREIBL

Date of Birth — Date de naissance

10-10-55

Country of Birth — Lieu de naissance

CZECHOSLOVAKIA

Permanent Address — Résidence fixe

3410 ALBION RD.
OTTAWA, ONT. K1V 8W3

Title of Thesis — Titre de la thèse

SAW DIFFRACTION COMPENSATION FOR LiNbO_3

University — Université

CARLETON

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M. ENG.

Year this degree conferred — Année d'obtention de ce grade

1984

Name of Supervisor — Nom du directeur de thèse

B. A. SYRETT

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication, ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

APRIL 3, 1984

Signature

Irena Streibl

SAW DIFFRACTION COMPENSATION FOR LiNbO_3

by

© Irena Streibl, B.A.Sc.

A thesis submitted to the Faculty of
Graduate Studies and Research in partial fulfilment
of the requirements for the degree of
Master of Engineering

Department of Electronics

Carleton University
Ottawa, Ontario
January 9, 1984

LICENCE TO CARLETON UNIVERSITY

In the interests of facilitating research by others at this institution and elsewhere, I hereby grant a licence to:

CARLETON UNIVERSITY

to make copies of my thesis:

SAW DIFFRACTION COMPENSATION FOR LiNbO_3

or substantial parts thereof, the copyright which is invested in me, provided that the licence is subject to the following conditions:

Only singled copies shall be made or authorized to be made at any one time, and only in response to a written request from the library of any university or similar institution or its own behalf or on behalf of one of its users.

This licence shall continue for the full term of the copyright, or for so long as may be legally permitted.

The Universal Copyright Notice shall appear on the title page of all copies made under the authority of this licence.

This licence does not permit the sale of authorized copies at a profit, but does permit the collection by the institution or institutions concerned of charges covering actual costs.

All copies under the authority of this licence shall bear a statement to the effect that the copy in question "is being made available in this form by the authority of the copyright owner solely for the purpose of private study and research and may not be copied or reproduced except as permitted by the copyright laws without written authority from the copyright owner."

The foregoing shall in no way preclude the granting by the author of a licence to the National Library of Canada to reproduce the thesis and to lend or to sell copies of the same. For this purpose it shall also be permissible for Carleton University to submit the abovementioned thesis to the National Library of Canada.

J. Wang
Signature of Witness

Yvonne Strubel
Signature of Student

April 3/84 MENG
Date Degree

Electronics
Department of Student

The undersigned recommend to the Faculty of Graduate Studies
and Research, the acceptance of the thesis:

"SAW Diffraction Compensation for LiNbO_3 "

submitted by Irena Streibl, B.A. Sc., in partial fulfillment of
the requirements for the degree of Master of Engineering.

Thesis Supervisor

Chairman, Department of Electronics

Carleton University •

March 1984

ABSTRACT

The work in this thesis presents an experimental and analytical study of the effects of diffraction on the frequency response of surface acoustic wave filters.

An algorithm which models and compensates for surface acoustic wave diffraction on Y-cut Z-propagating lithium niobate substrates has been developed. The approach to the diffraction model is based on a physical understanding of the problem and uses the concept of Fresnel integrals with a novel approach to modelling the anisotropic velocity behaviour of the crystal. Based on the diffraction model a diffraction compensation algorithm is developed via time domain considerations.

The approach taken in this thesis is accurate enough to give a 9 dB improvement in side lobe levels for YZ-LiNbO₃ substrates.

ACKNOWLEDGEMENT

I would like to express my deep appreciation for the support and guidance to my thesis' supervisor Dr. Mark S. Suthers of Bell-Northern Research, Ottawa. Without his expertise, continued encouragement and involvement, the completion of this work would not be possible.

The author recognizes the privileges extended to her by Bell-Northern Research, Ottawa; namely Dr. Dave R. Conn of department 1C52 who suggested the thesis topic and provided her with access to the laboratory at the Transmission Development Division 1C00 of Bell-Northern Research, and shared an enthusiastic interest in this work.

Finally, the author is most grateful for the constant guidance and encouragement provided, throughout the entire program, by her thesis supervisor from Carleton University, Professor Barry A. Syrett.

TABLE OF CONTENTS

	<u>PAGE</u>
CHAPTER I: INTRODUCTION	
1.1 Introduction to the Thesis Topic	1
1.2 Thesis Objective	7
1.3 Thesis Outline	10
CHAPTER II: PHYSICAL THEORY OF SAW DEVICES	
2.1 Principle of Operation	12
2.2 Design Considerations	18
2.2.2 Temperature Coefficient of Delay	20
2.2.3 Loss Mechanisms	20
2.2.4 Electromagnetic Feedthrough	21
2.2.5 Triple Transit Echo	22
2.2.6 Edge Reflections	24
2.2.7 Bulk Wave Coupling	24
2.3 Principles of Diffraction Theory	26
CHAPTER III: DESIGN ALGORITHM	
3.1 Design Analysis	31
3.2 Design Procedure	33
3.3 Selection of the Window Function	37
3.3.1 Rectangular Window	37
3.3.2 Hamming Window	39
3.4 Computer Design Algorithms	42

TABLE OF CONTENTS (Continued)PAGE

CHAPTER IV: DIFFRACTION MODELLING

4.1	Introduction	44
4.2	Diffraction Integral	46
4.3	Anisotropy of LiNbO_3	50
4.4	Diffraction Integral for LiNbO_3	55
4.5	Modelling of Diffraction	58

CHAPTER V: COMPENSATION ALGORITHM

5.1	Introduction	63
5.2	Development of the Diffraction Compensation Algorithm	65
5.3	Computer Algorithm	69

CHAPTER VI: TESTING AND EXPERIMENTAL RESULTS

6.1	Time Domain Measurements	73
6.2	Practical Implementation	76
6.3	Experimental Results	78
6.3.1	Diffraction Un-compensated Filter	79
6.3.2	Diffraction Compensated Filter	82

CHAPTER VII: CONCLUSIONS AND FUTURE WORK

7.1	Conclusions	87
7.2	Future Work	89

v

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
REFERENCES	91
APPENDIX I: FRESNEL INTEGRAL	94
APPENDIX II: COMPUTATION OF FRESNEL INTEGRALS	98
APPENDIX III: FRESNEL INTEGRAL WITH THE ANISOTROPIC APPROXIMATION FOR YZ-LiNbO ₃	100
APPENDIX IV: FREQUENCY DIVIDER CIRCUIT	104
APPENDIX V: FABRICATION SUMMARY	105
APPENDIX VI: COMPUTER PROGRAM LISTINGS	108

LIST OF SYMBOLS

A_m	amplitude of the m -th sample
B_a	reactive component of the acoustic admittance
C_T	static capacitance
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
e	base $e = 2.718\ 281\ 828$
f	frequency
f_0	center frequency
f_e	lower passband frequency
f_s	sampling frequency
f_u	upper passband frequency
f_{se}	lower stopband frequency
f_{su}	upper stopband frequency
G_a	radiation conductance
$h(t)$	impulse response
$H(f)$	frequency response
IDT	interdigital transducer
IL	insertion loss
$I(x)$	current distribution on the radiating element

LIST OF SYMBOLS (Continued)

k	acoustic wave propagation constant
k_e	acoustic energy propagation constant
k^2	coupling coefficient
L	length of the radiating element scaled to the acoustic wavelength
$+L/2, -L/2$	integration limits for the radiating element
LiNbO_3	lithium niobate
N	number of transducer fingers
NP	number of transducer finger pairs
N_s	number of samples required
P	field point
S_n	sign of the n -th sample
t	time
t_n	time of the n -th sample
Δt	time increment from finger to finger
t_+, t_-	limits of the Fresnel integral
TTE	triple transit echo
v, v_p	acoustic phase velocity
v_e	acoustic energy velocity
v_f	unmetallized surface acoustic velocity
v_g	group velocity

LIST OF SYMBOLS (Continued)

v_s	metallized surface acoustic velocity
v_o	acoustic velocity at the pure mode axis
v_+^2, v_-^2	limits of the Fresnel integral
X	distance from the center of the radiating element to the field point along the x-axis, scaled by the acoustic wavelength
y_a	acoustic admittance
Z	distance from the radiating element of the field point along the pure mode axis, scaled by the acoustic wavelength.
$\sigma(t+\tau_m)$	Dirac delta function
ϕ	angle of propagation
λ	free space wavelength
λ_a	acoustic wavelength
λ_e	energy wavelength
λ_g	group wavelength
ρ	distance from the center of the radiating element to the field point
ρ_o	distance from the radiating element to the field point along the pure mode axis
τ_D	delay between two transducers

LIST OF SYMBOLS (Continued)

τ_I	delay between the center of the input transducer and the edge of the substrate
τ_O	delay between the center of the output transducer and the edge of the substrate
ω	angular frequency
ω_S	sampling angular frequency

CHAPTER I

INTRODUCTION

1.1 Introduction to the Thesis Topic

Surface acoustic waves have developed as an important new technology since the publication of the first delay line by White [1] in 1965. There are many advantages to using surface acoustic wave (SAW) devices in place of conventional components. Low cost, small size, uniformity of performance and ease of manufacture have contributed to making this a successful technology. To illustrate these points a SAW IF band pass filter as shown in Figure 1.1 fits into a TO-8 header. In large quantity production runs (>10,000/yr.) it would cost about \$5-\$20 per unit to manufacture.

The techniques used in the fabrication of SAW devices are compatible with the planar technology used for integrated circuits. Since the substrate is a chemically pure single crystal and since the device pattern is highly reproducible, the performance from device to device is virtually identical. This results in devices which do not need to be tuned once manufactured. A simple visual screening of the devices ensures a yield approaching 100%.

A few of the limitations of SAW devices will be discussed below.

A conventional interdigital transducer (IDT)¹ has a simplified equivalent electrical circuit consisting of the acoustic admittance $Y_a = G_a + jB_a$ and the static capacitance C_T (Figure 1.2).

¹ The IDT is described in Chapter 2, Section 2.1.

FIGURE 1.1: 70 MHz filter IMS-102 mounted in a TO-8 header

FIGURE 1.2: An equivalent circuit of an interdigital transducer (IDT). Y_{in} is the input admittance of an IDT.

The real component of the acoustic admittance, G_a , is the radiation conductance and it represents the electrical to mechanical energy transformation. The imaginary component B_a is the reactive component of the acoustic admittance. Both G_a and B_a are frequency dependent terms. C_T is the static capacitance which is frequency independent. C_T depends on the number of fingers, the finger overlap, the spacing between fingers and the dielectric constant of the substrate.

When the input transducer is matched at the electrical port, half of the acoustic power is transmitted towards the output transducer and the other half is dissipated in an acoustic load (Figure 1.3). By reciprocity, the same applies to the output transducer. Hence there

FIGURE 1.3: Signal transmission with bi-directional IDT's

is a minimum built-in loss of 3 dB ($\frac{1}{2}$ power) for each transducer. This inherent loss is usually compensated for by amplification.

Since it is very difficult to obtain a perfect acoustic match, part of the signal travelling to the left of the input IDT will be reflected back to the receiving IDT from the substrate edge. This unwanted delayed image will cause ripple in the filter's passband. Cutting the substrate edge at an angle helps to disperse these reflections. Applying an acoustic absorber at the edges also significantly attenuates these reflections. The problem of reflections applies to the output transducer as well.

There exists another reflection inherent to bi-directional transducers, the triple transit echo (TTE) which can be suppressed but never completely eliminated. The transmitted signal of Figure 1.3 reaches the output IDT and part of it is reflected back towards the input IDT where it gets re-reflected to the output. This delayed signal is typically 40 to 50 dB down from the main signal and causes a passband ripple in the order of ± 0.05 dB [2].

The metallization of the IDT is another source of reflections, however, these may be suppressed by the use of split fingers. Fingers spaced at a half wavelength separation will cause the reflections to build up in phase. By selecting the split fingers design, the reflections from successive fingers will cancel each other [2].

Diffraction is a source of wavefront modification related to propagation in any medium. In SAW devices, the acoustic beam spreads out as it propagates away from the transmitting IDT. As a result, the

receiving transducer will intercept only a fraction of the transmitted power. This problem occurs predominantly in apodized¹ transducers where there exist regions of very small overlap. To overcome this problem, the finger overlap must be adjusted. This is the subject of Chapter 5.

The distortion due to diffraction is most noticeable in the filter's stopband as an increase in the level of the sidelobes (Figure 1.4). Also some degradation of the passband ripple and slope is attributed to diffraction.

In the next section the thesis objective is defined and the sequence of work presented is outlined.

¹ In an apodized IDT the fingers are not of uniform length, that is their length varies from finger to finger.

FIGURE 1.4: Predicted frequency response of an ideal transversal filter compared to the experimental result

1.2 Thesis Objective

Many authors have addressed the topic of diffraction modelling and diffraction compensation for SAW devices in recent years [3,4,5,6,7,8]. However, no universally successful solution has been proposed. The investigations reported have been unique to specific substrates and the results have not been generalized.

Lithium Niobate (LiNbO_3) is a substrate of great interest for fabrication of surface acoustic wave filters. It has a high electro-mechanical coupling coefficient k^2 which permits high transducer conversion efficiency for wideband filters. However, modelling the effects of diffraction on this material has proven to be difficult because of the highly anisotropic nature of the velocity versus the direction of propagation.

It is the objective of this thesis to isolate the effects of diffraction in the response of a SAW IF filter built on a LiNbO_3 substrate and to develop a computer model for this effect. Based on the computer model a new filter is designed, fabricated and tested. This new filter has correction for diffraction incorporated into its design. The frequency response of the latter device is compared to the desired response and the model and the compensation technique are discussed and evaluated. The errors in the design algorithms are quantified and suggestions for refinement of the numerical techniques are presented.

Figure 1.5 outlines the work that has been done to meet the thesis objective. As the first step a design without diffraction considerations was completed. A filter based on those specifications was fabricated. Using this first device as a benchmark, the effects of diffraction were measured and diffraction profiles, showing the

FIGURE 1.5: Flowchart of work presented in this thesis

spreading of the SAW wave energy as the SAW wave travelled from the input IDT to the output IDT, were predicted on the computer. The theoretical model, based on the diffraction profiles, was developed and used to compare this design to the experimental frequency response. Based on this model a new diffraction-compensated filter design was completed. Another filter was fabricated from the compensated design and its predicted frequency response was compared to the experimental result. Finally the effectiveness of the model and compensation algorithm were evaluated.

1.3 Thesis Outline

Chapter 2 outlines the basic operation of surface acoustic waves and presents terminology specifically related to SAW devices. The physical requirements for a surface wave to propagate are stated and wave propagation in an isotropic medium is compared to that in an anisotropic medium. Further, the design considerations in SAW filters are discussed as well as the radiation from a slit according to classical optics [9].

Chapter 3 describes the procedure used to design a SAW filter without any consideration of diffraction effects. A flowchart of the computer design algorithm is included.

Chapter 4 starts with a discussion of the literature covering the topic of diffraction modelling and diffraction compensation. The theory developed in this thesis for modelling diffraction on LiNbO_3 is summarized. The diffraction integral is derived for an isotropic medium and modifications to the integral are presented in order to include the anisotropy on LiNbO_3 . Computer modelling for both cases is discussed.

In Chapter 5 the diffraction compensation algorithm is derived. The computer implementation of the compensation algorithm is discussed and the predicted diffraction frequency response for both the isotropic and the anisotropic model are compared.

The experimental results of all the devices fabricated are presented and discussed in Chapter 6. A description of the experimental test set-up used to isolate the effects of diffraction is also described.

Finally, in Chapter 7, conclusions are drawn from the experimental results and the modelling and compensation techniques are reviewed. Recommendations for future work are discussed as a completion of this work.

CHAPTER II

PHYSICAL THEORY OF SAW DEVICES

2.1 Principle of Operation

Surface acoustic waves are generated by an interdigital transducer (IDT) deposited on a piezoelectric substrate (Figure 2.1).

FIGURE 2.1: Prototype of a SAW IDT on YZ-LiNbO_3

The term piezoelectricity means an interdependence of elastic and electric properties in certain materials. A material is piezoelectric if a mechanical deformation induces an electric polarization (direct effect) or conversely if it experiences a mechanical deformation (strain) when an electric field is applied (inverse effect). The discovery of the direct piezoelectric effect was made by Pierre and Jacques Curie in 1880 [10].

The interdigital transducer (IDT) is composed of two metal comb-shaped electrodes. The voltage applied to these electrodes or fingers, produces an electric field which creates a stress-strain relationship in the vicinity of the surface giving rise to various types of elastic waves. A surface wave is, in principle, any deforma-

tion that when propagating affects only a small depth of material near the surface. The elastic wave of interest is the Rayleigh wave (Figure 2.2) which is generated in the direction orthogonal to the comb fingers, i.e., the Z-direction of Figure 2.1.

Rayleigh waves consist of a longitudinal displacement and a $\pi/2$ phase shifted vertical displacement (Figure 2.3). The extremity of the polarization vector describes an ellipse and the particle displacement within the sagittal plane¹ vanishes at a depth of two wavelengths (Figure 2.2). Since Rayleigh waves are confined to the surface of the crystal, successive wave crests may be tapped or generated by an interdigital transducer which is deposited on the surface plane.

As far as the Rayleigh waves are concerned, the transducer can be modelled as a sequence of ultrasonic line sources (Figure 2.4). For an applied sinusoidal voltage, all vibrations interfere constructively only if the distance d (Figure 2.1) between two neighbouring fingers is close to half the elastic wavelength. Any deformation created at time t by a finger pair, for a given polarity of the applied voltage, travels at the Rayleigh velocity V_R along a distance $\lambda/2$ during the half period $T/2$. At time $t + T/2$ this deformation has reached the next pair of fingers precisely when the voltage, which has changed sign, generates an in-phase deformation; the elastic excitation due to the second pair re-enforces that of the first pair. The frequency $f_0 = V_R/2d$ that corresponds to this cumulative effect is called the synchronous frequency or the resonant frequency of the transducer. If the frequency departs from this value, the interference between the elastic signals due to various finger pairs is less constructive and the overall vibration is weaker. Thus the bandwidth of a transducer is dependent on the number of fingers.

¹ The sagittal plane is perpendicular to the plane of propagation in the direction of propagation.

FIGURE 2.2: The propagation of a Rayleigh wave in a crystal [10]

FIGURE 2.3: Amplitude of displacement components as a function of depth. Displacement components are in phase quadrature [2].

FIGURE 2.4: Method of discrete sources. The transducer is regarded as a discrete sequence of sources, each one located midway between two fingers, and of amplitude proportional to the overlap length.

In the discrete source approach each finger is considered as an infinitely narrow ultrasonic source or receiver located midway between two fingers, as shown in Figure 2.4. The amplitude A assigned to each source is proportional to the overlap length of the two fingers, with positive or negative sign depending on the direction of the electric field.

With these assumptions the impulse response $h(t)$ of a transducer is a series of Dirac functions:

$$h(t) = \sum_n S_n A_n \delta(t - t_n) \quad (2.1)$$

The sign $S_n = (-1)^n$ implies that the electric field is reversed from one interdigital interval to the next.

The frequency response is obtained as:

$$H(f) = \sum_n S_n A_n e^{-j 2\pi f t_n} \quad (2.2)$$

which is the Fourier transform of the impulse response equation (2.1):

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j 2\pi f t} dt \quad (2.3)$$

A sequence of delta functions represents not only the sine wave at the center frequency f_0 , but also all the odd harmonic sine waves at frequencies $(2m + 1) f_0$, $m = 0, 1, 2 \dots n$ (Figure 2.5).

FIGURE 2.5: A sequence of delta functions. Only odd harmonics are present.

Due to the half wavelength spacing between the fingers of the IDT, all of the even harmonic components cancel completely.

The electric field associated with the elastic wave, when passing under the electrodes, induces a time varying voltage. The shape of this signal depends on the number of receiver fingers (Figure 2.6). It follows from Figure 2.6 that the overall filter's impulse response is the convolution of the impulse responses of the input IDT and of the receiving IDT. The frequency response is the Fourier transform of the filter's impulse response.

FIGURE 2.6: Impulse response of a delay line consisting of two interdigital transducers [10].

- (a) The receiver has only one finger pair
- (b) The receiver is identical to the emitter.

2.2 Design Considerations

There are several important parameters and corruptions affecting the performance of a SAW device.

2.2.1 The Coupling Coefficient

It is customary to define a coupling constant for surface waves [10] by

$$k^2 \cong 2 \left(\frac{v_f - v_s}{v_s} \right) \quad (2.4)$$

where v_f = the free unmetallized surface velocity
 v_s = the metallized surface velocity.

This electromechanical coupling constant k^2 measures the relative effect of a metal film on the propagation velocity v . It characterizes the material's ability to transform an electric signal into an elastic surface wave through a transducer consisting of metal fingers deposited on the surface.

For Y-cut lithium niobate (Figure 2.7) the maximum difference between the two velocities occurs when the propagation is along the Z-axis: $v_f = 3488$ m/s and $v_s = 3404$ m/s. The resulting value of $k^2 \cong 4.94\%$ is quite high compared to other piezoelectric substrates [10].

FIGURE 2.7: Rayleigh wave in lithium niobate (Y-cut). Velocities v_s on metallized surface and v_f on unmetallized surface vs. propagation direction in the XZ plane [10].

2.2.2 Temperature Coefficient of Delay

The temperature coefficient of delay is another important consideration in SAW devices. A change in the temperature causes a change in the surface wave velocity and the transducer's center frequency will change correspondingly. The time delay between the input and the output transducer will also vary. The value of this coefficient is $-90 \text{ ppm}/^\circ\text{C}$ for Y-cut Z-propagating LiNbO_3 .

2.2.3 Loss Mechanisms

Acoustic attenuation is a fundamental loss mechanism where the attenuation constant is approximately proportional to the square of the frequency. As the surface wave propagates, part of its energy is scattered by the solid and dissipates as heat in the substrate. This loss is of importance at frequencies above 500 MHz; at standard IF frequencies it is negligible. The same holds true for the ohmic losses in the transducer electrodes. There is little loss due to the coupling of surface waves to bulk waves for the crystal cuts normally used, since bulk waves travel faster than surface waves thus binding the energy to the surface. The attenuation of a surface-wave by coupling to ultrasonic waves in the air can be ignored for frequencies below 1 GHz.

2.2.4 Electromagnetic Feedthrough

There is always a certain amount of electromagnetic (EM) feedthrough. This means that not all of the electrical signal follows the acoustic path between the input and output IDT. Some of it is transmitted by electromagnetic waves which travel at the speed of light.

The feedthrough signal occurs "instantly" when the input voltage is applied while the desired signal is delayed by travelling at the acoustic velocity along the surface of the crystal (Figure 2.8) [11]. Beating of these two signals will cause ripples in the frequency response. By careful packaging of the device the EM feedthrough noise floor can be brought down to be better than -60 dB.

FIGURE 2.8: The delay between signals in time domain.

The EM feedthrough arrives at the receiver "instantly" followed by the longitudinal bulkwave. The main SAW signal coincides with the shear bulkwave. Reflections from the edges arrive later and finally the triple transit echo.

2.2.5 Triple Transit Echo

The triple transit echo (TTE) causes a further interference for bidirectional transducers (Figure 2.9). When a voltage is applied to the electrodes of the input IDT half of the signal is transmitted in the desired propagation direction towards the output IDT. As the main signal reaches the output IDT part of it is reflected back towards the input IDT where it is re-reflected again to the output IDT. This delayed and attenuated image will beat with the main signal thus causing ripples in the magnitude and phase of the frequency response. The frequency of this ripple is determined by the inverse of twice the delay time τ_D between the two transducers [11].

$$f_{TTE} = \frac{1}{2\tau_D} \quad (2.5)$$

The level of the TTE interference relates to the level of the main signal by the following expression [12]:

$$\frac{\text{TTE power}}{\text{main signal power}} \text{ (dB)} = 6 \text{ (dB)} + 2 \text{IL (dB)} \quad (2.6)$$

where IL is the insertion loss of the transducer.

The triple transit echo can be eliminated by building unidirectional transducers. However these designs are much more involved and are more costly to fabricate than the bidirectional transducers. The system application of the filter will dictate if such an approach should be used.

FIGURE 2.9: SAW delay line with two bidirectional interdigital transducers

2.2.6 Edge Reflections

The reflections from the substrate edges are also a source of passband ripple (Figure 2.10). The frequency of the ripple due to the input IDT is equal to the inverse of twice the delay time between the center of the IDT and the substrate edge:

$$f_I = \frac{1}{2\tau_I} \quad (2.7)$$

and similarly for the output IDT

$$f_O = \frac{1}{2\tau_O} \quad (2.8)$$

However, these reflections can be quite easily eliminated by applying acoustic absorber at the edges and by cutting the substrate at an angle with respect to the transducer fingers. In the latter case the signal will be reflected away from the transducer and the possibility of performance degradation due to reflections is eliminated.

2.2.7 Bulk Wave Coupling

For YZ-LiNbO₃ crystals, the bulk wave coupling is another source of performance degradation. The transducer excites not only Rayleigh waves but also different modes of bulk waves [13,14,15]. As the frequency increases the bulk waves propagate steeper into the substrate. But the slow shear bulk wave travels at a velocity only 3% higher than the Rayleigh wave and it can propagate near the surface [13]. Hence at the high frequency side of the passband the coupling of the bulk wave significantly increases the sidelobe levels.

FIGURE 2.10(a): Passband response showing ripples due to end of substrate reflections.

FIGURE 2.10(b): Passband response without ripples from end of substrate reflections.

2.3 Principles of Diffraction Theory

Diffraction is a phenomenon inherent to wave propagation regardless of the propagation medium.

Diffraction is the spreading of the acoustic beam as shown in Figure 2.10.

FIGURE 2.11: Beam spreading due to diffraction.

It is apparent from the above Figure 2.10 that not all of the radiated energy is intercepted by the output IDT. This introduces an additional source of performance degradation in SAW devices.

Diffraction in SAW devices is analogous to the diffraction encountered in optics when a plane wave illuminates a slit (Figure 2.11) [9,16,17]. In contrast to the classical optical case, the propagation medium of SAW devices is highly anisotropic and the resulting variation of phase velocity with the angle of propagation must be taken into account in diffraction calculations [8].

In the case of isotropic diffraction the phase velocity remains constant regardless of the direction of propagation. Figure 2.12(a) illustrates how the isotropic diffraction causes the beam profile to

FIGURE 2.12: Plane wave incident on a slit in a screen. The edge diffraction leads to spreading of the radiation from the slit [16].

quickly lose its uniform amplitude and phase as it travels away from an input transducer. The profile undergoes significant fluctuations in shape and it eventually broadens to an infinite width. For the anisotropic case (Figure 2.12(b)) the direction of phase and group velocity vectors coincide only for specific values of θ defined as the pure mode axes. Thus the acoustic beam is shown propagating off at an angle ϕ , the power flow angle. For an anisotropic substrate the diffraction patterns are unique to the particular anisotropy and do not occur in the isotropic manner.

Now consider the situation shown in Figure 2.11 in which an infinite plane wave is incident on an infinite flat sheet. The sheet has a slot of width a and of infinite length in the direction normal to the page. The field distribution across the slot may be assumed to be uniform and the field everywhere to the right of the sheet is the result of the section of the wave that passes through the slot.

By Huygens' principle [17] the field everywhere to the right of the sheet is the same as though each point in the plane of the slot is the source of a new spherical wave (Figure 2.11). Each of these point sources is of equal magnitude and phase. Thus by Huygens' principle the slotted sheet with a uniform field across the opening can be replaced by a continuous array of point sources which just fill the opening. The field pattern in the X-Z plane is then calculated in the same way as for a continuous linear array of point sources.

FIGURE 2.13: (a) Schematic representation of launching and propagation of acoustic surface wave on an isotropic substrate. Note diffraction effects. θ defines direction of propagation with respect to reference crystalline axis.

(b) Schematic representation of launching and propagation of acoustic surface wave on anisotropic substrate. Note diffraction effects and how the phase velocity and power flow directions are separated by angle ϕ . θ defines the direction of propagation while ϕ is power flow angle [8].

The diffraction patterns can be computed by applying Huygens' principle to the linear radiator (Figure 2.13):

FIGURE 2.14: Linear radiator configuration

The field at point P is expressed as

$$E(x_1, R) = K \int_{-L/2}^{+L/2} I(x) \frac{e^{-jk\rho}}{\rho} dx \quad (2.9)$$

where $k = 2\pi/\lambda$ is the wave number

K = constant along x-axis

$I(x)$ = current distribution on the element

ρ = distance from the radiating element to the field point P.

The solution in the region near the radiating element is given by the Fresnel integral and the patterns are called the Fresnel patterns (Figures 2.14(a) and (b)). The far field pattern takes on the $\sin x/x$ shape (Figure 2.14(c)) and is called the Fraunhofer diffraction pattern [18]. It is shown in polar coordinates in Figure 2.14(d).

FIGURE 2.15: Fresnel and Fraunhofer patterns of a slot of width a [17].

Now that the principles of operation of SAW devices as well as some of the physical design considerations have been reviewed, one can proceed with the development of the design algorithm in Chapter 3.

CHAPTER III

DESIGN ALGORITHM

3.1 Design Analysis

Given the desired frequency specification of a filter (Figure 3.1) a design procedure must be determined which will satisfy the required criteria.

The filter's impulse response is the Inverse Fourier Transform [19] of the desired frequency response (Figure 3.2). The infinite impulse response is then truncated by applying the appropriate window function [20].

FIGURE 3.1: Design Specification of a 70 MHz Filter

The width of the window in the time domain is determined from the transition width in the frequency domain, i.e., the convolving function of Figure 3.1. The multiplication of the impulse response by the specified window function yields the filter's finite impulse response. Taking the Discrete Fourier Transform (DFT) [20] of the finite impulse response results in the final frequency response. The resulting frequency response can be alternatively obtained by convolving the desired frequency response with the convolving function, i.e., the transition width in the frequency domain (Figure 3.1).

FIGURE 3.2: Generalized design process [11]

3.2 Design Procedure

As the first step in the filter design the number of fingers of the input non-apodized transducer will be determined. The finger overlaps are of constant width and hence carry uniform weighting of ± 1 depending on the polarity of the connecting rail (ref. Figure 2.1). The term "apodized" means a variation in the length of the finger overlaps which will be the case for the output receiving transducer.

The number of fingers required for the input IDT is given by the position of the first null of the filter's frequency response. Figure 3.1 specifies the first nulls to be at 63 and 77 MHz. Since the uniform non-apodized transducer has a rectangular profile in the time domain, its frequency response will be the $\sin x/x$ function. The position of the nulls of the sinc function (Figure 3.3) is related to the number of finger pairs by the following expression:

$$\left(1 \pm \frac{n}{NP}\right) f_0 = f_n ; n = \pm 1, \pm 2, \dots \quad (3.1)$$

FIGURE 3.3: Position of nulls of $\sin x/x$ dependence on the number of finger pairs NP.

From (3.1) follows the choice of 9.5 finger pairs or 19 fingers with the first nulls positioned at 62.6 MHz and 77.4 MHz.

The specification given by Figure 3.1 requires the roll-off to be less than 1 dB at 66.5 MHz and 73.5 MHz. However, with the present design it is -3.6 dB at those frequencies. This discrepancy will be compensated for in the design of the output apodized IDT.

Figure 3.4(b) shows the desired rectangular filter response while Figure 3.4(a) shows the $\sin x/x$ frequency response of the input IDT. Dividing the position of the $\sin x/x$ bounded by $f_0 \pm \Delta f$ into the ideal response yields the required frequency response of the apodized IDT Figure 3.4(c). The "horns" of this response compensate for the roll-off due to the $\sin x/x$ response of the non-apodized IDT. Since the filter's frequency response is obtained by multiplying the frequency response of the input IDT with that of the output IDT, one can see that multiplying the waveforms of Figures 3.4(a) and 3.4(c) yields the required filter response shown in 3.4(b).

The next step is to determine the output IDT's impulse response by using the Inverse Fast Fourier Transform [21]. This yields the infinite impulse response which is truncated by multiplication with the window function (Figures 3.4(d) and (e)). The apodization weighting of the output IDT is specified by the sample weights of the truncated impulse response.

The convolution of the input IDT's impulse response, Figure 3.5(a), with that of the output IDT, Figure 3.5(b), yields the filter's finite impulse response (FIR) as indicated in Figure 3.5(c). The FFT of the finite impulse response results in the final frequency response of the filter, Figure 3.5(d).

The design sequence outlined above may have to be repeated 2 or 3 times to achieve the required frequency specification. The width Δf of Figure 3.4 has to be adjusted by trial and error to compensate for the effect of the window function at the "horns" of the apodized IDT's frequency response Figure 3.4(c).

FIGURE 3.4: Design process of a SAW filter

f_0 = center frequency

$2\Delta f$ = passband width

τ_D = delay between the input and output transducers

FIGURE 3.5: Basic steps in filter design

3.3 Selection of the Window Function

The selection of the right window function is an important consideration in the filter design. The procedure used here is based on that developed by Harris [22].

It is evident from Figure 3.1 that the transition width of the filter is specified by the position of the first null of the window function.

3.3.1 Rectangular Window

For a rectangular window the resolution between the first nulls is equal to 2 bins (Figure 3.6), where a bin is the fundamental frequency resolution.

$$1 \text{ bin} = \frac{\omega_s}{N} \quad (3.2)$$

where ω_s = sampling frequency

N = number of samples, one sample corresponds to one finger

FIGURE 3.6: Fourier transform of rectangular window is $\sin x/x$. The first side lobe level is -13.4 dB. A bin is the fundamental frequency resolution.

The sampling frequency is equal to $2f_0$ since the analysis assumes two fingers per wavelength.

Figure 3.1 indicates that 1 bin = 3.5 MHz and one can calculate the necessary number of fingers using (3.2) as:

$$N = \frac{\omega_s}{1 \text{ bin}} \quad (3.3)$$

Substituting the above values into (3.3) and letting $f_0 = 70$ MHz yields:

$$\begin{aligned} N &= \frac{2\pi (2f_0)}{2\pi (1 \text{ bin})} \\ &= \frac{2 (70)}{3.5} \end{aligned}$$

$$N = 40 \text{ fingers}$$

Hence the transition width of 3.5 MHz corresponds to 40 fingers. As a safeguard the transition width should be reduced by 20% corresponding to 700 kHz. The new transition width is 2.8 MHz and corresponds to 50 fingers.

The truncation of the impulse response by a rectangular window causes significant ripple in the filter's passband as indicated in Figure 3.2. Figure 3.1 specifies the sidelobe level to be below -40 dB. Here again the rectangular window is unsuitable since it has the first sidelobe at -13.4 dB. Another window function must be considered then to meet the filter's criteria.

3.3.2 Hamming Window

The Hamming window belongs to the $\cos \alpha(x)$ family of windows. The parameter $\alpha = 0.54$ and the first side lobe level is -43 dB.

The general form of the Hamming window is [22]:

$$\omega(n) = \begin{cases} 0.54 + 0.46 \cos \left[\frac{2\pi}{N} n \right]; & n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \\ 0.54 - 0.46 \cos \left[\frac{2\pi}{N} n \right]; & n = 0, 1, 2, \dots, N-1 \end{cases} \quad (3.4)$$

where N = total number of samples (fingers)

n = n -th sample (finger)

As in the case of the rectangular window the first zero occurs at 3.5 MHz and it is defined by:

$$\omega \cong 2.6 \left[\frac{2\pi}{N} \right] f_s \quad (3.5)$$

where ω = position of the first null

f_s = sampling frequency

N_s = number of samples (fingers) required

From equation (3.5) the number of fingers required follows as:

$$N = 2.6 (2\pi f_s) \frac{1}{\omega} \quad (3.6)$$

Substituting for $f_s = 140$ MHz and $\omega = 3.5$ MHz (2π) yields:

$$N = 2.6 (2\pi) (140) \frac{1}{2\pi * 3.5}$$

$$N = 104 \text{ fingers}$$

Applying a 20% guardband increases the number of fingers required to 125.

Figure 3.7 compares the truncation of the infinite impulse response as done by the rectangular window and by the Hamming window. The Hamming window, a cosine function on a pedestal, reduces the width of the main lobe as well as the level of the adjacent lobes. This translates into the frequency domain as reduced ripple in the passband and sidelobe levels below -43 dB.

The conclusion can be drawn that the Hamming window is the most suitable one to use to meet the required filter specification indicated by Figure 3.1.

FIGURE 3.7: The truncation of the infinite impulse response by multiplication with the rectangular and Hamming windows.

3.4 Computer Design Algorithm

The flowchart of the computer design routines is indicated in Figure 3.8.

The vectors representing the input and output IDT consist of 512 samples each. The spacing between samples corresponds to half a wavelength which can be converted to $\Delta t = \frac{1}{2f_0}$ in the time domain. The total length of 512 samples is based on the amount of work space required to perform numerical convolution and the FFT. For an easier application of the FFT it is recommended that the vector be selected as a power of two [21], i.e., $512 = (2)^9$.

In step 1, the 512 sample vector representing the input IDT is generated and it consists of a sequence of samples with weighting +1 or -1 and zeros. The initial sequence of ± 1 represents the 19 fingers of the non-apodized IDT and the remaining sample values are set to zero.

In the next step the frequency response of the non-apodized IDT is computed using the FFT algorithm. The output IDT's frequency response is predistorted in order to compensate for the input IDT's frequency response.

Next the Inverse Fourier Transform is applied to the output IDT's frequency response and its time domain response is evaluated.

Step 4 applies the Hamming window function to truncate the infinite impulse response of the output IDT and the 125 sample output IDT is positioned at the correct delay with respect to the input IDT.

Finally, the numerical convolution of the two IDT's is performed which yields the finite impulse response of the filter. The last check in the design sequence is the frequency response obtained by applying the FFT to the filter's impulse response.

FIGURE 3.8: Flowchart of design sequence of a SAW filter

CHAPTER IV

DIFFRACTION MODELLING

4.1 Introduction

Diffraction modelling in SAW devices is an area of great interest. Because of the highly anisotropic behaviour of the materials used for the fabrication of these devices it is by no means a trivial task to model the diffraction effects. Since the anisotropy, the change in velocity with the propagation direction, is unique to each substrate, there exists no universal solution to this problem.

Szabo and Slobodnik [8] used the parabolic velocity surface theory to model diffraction on quartz and the angular spectrum of waves theory to model diffraction on YZ-LiNbO₃ [36]. They reported good agreement between the predicted and experimental diffraction profiles for quartz but not for YZ-LiNbO₃.

An efficient numerical technique for simulating the effect of diffraction in SAW devices was developed by Penuri [7]. It was tested for parabolic diffraction and the experimental and theoretical frequency responses were typically within 2 dB on ST-quartz.

Savage and Mattheef [5] modelled diffraction using the Fresnel and parabolic anisotropy approximations for ST-quartz substrate. They published good improvement in the filter's frequency response when diffraction has been compensated for.

Mader [6] developed a compensation algorithm which computes corrections to the tap weights from the deviation of the measured frequency response from the desired one. They achieved excellent agreement between theory and experiment since their method compensates for other second-order effects as well.

Fast computation of diffraction in general anisotropic media by use of the Geometrical Theory of Diffraction (GTD) was presented by Radasky and Matthaei [4]. Their technique provides a means for computing diffraction patterns on a medium like YZ-LiNbO_3 , where the parabolic anisotropic approximation is not valid. Comparisons between the diffraction profiles obtained by using the angular spectrum calculations and the GTD on YZ-LiNbO_3 substrate indicate that the GTD technique is very good, however the angular spectrum of waves solution did not give good results for YZ-LiNbO_3 when used by Szabo and Slobodnik [8].

The technique for diffraction compensation on YZ-LiNbO_3 developed in this thesis is based on approximating the velocity versus the angle of propagation curve (Figure 2.7) by an eighth-order polynomial [3]. A correction for the velocity change is then incorporated into the diffraction integral. The approach uses the concept of Fresnel integrals which are used to calculate diffraction patterns of subelements of a finger in an apodized transducer. The complete diffraction pattern is obtained by shifting and integrating this pattern over the output aperture. This approach minimizes the error introduced by moving the $\exp(-jk_0)$ term outside of the integral.

4.2 Diffraction Integral

The diffraction integral evaluates the complex field strength at a certain point P away from the radiating element, i.e., finger of a transducer (Figure 4.1).

FIGURE 4.1: The complex field strength at point P is evaluated by shifting the radiating element of length L along the x -axis from start to end.

The complex field at point P is expressed as [18]:

$$E(x_1, \rho_0) = K \int_{\text{line}} A(x) \frac{e^{-jk\rho}}{\rho} dx \quad (4.1)$$

where K = a constant along the x-axis, equal to unity

A(x) = acoustic distribution on the element,

A(x) = 1 for all x

k = the wave number = $2\pi/\lambda_0$

The distance from the incremental radiating element dx to the field point P is:

$$\rho = \sqrt{\rho_0^2 + (x - x_1)^2} \quad (4.2)$$

From Figure 4.1 follows that $x_1 = 0$, hence:

$$\begin{aligned} \rho &= \sqrt{\rho_0^2 + x^2} \\ &= \rho_0 \sqrt{1 + \frac{x^2}{\rho_0^2}} \end{aligned} \quad (4.3)$$

In general ρ_0 is very much greater than the array length, i.e., the finger length. Consequently, the denominator of (4.1) varies only slightly over the integral and may be replaced by ρ_0 . The effect of the variation of ρ in the exponent is quite different, however, since $\exp(-jk\rho)$ rotates 360° when the distance changes by one wavelength.

Expanding the expression under the square root in (4.3) in a binomial series and preserving only the dominant terms yields:

$$\left[1 + \frac{x^2}{\rho_0^2}\right]^{\frac{1}{2}} = 1 + \frac{1}{2} \frac{x^2}{\rho_0^2} \quad (4.4)$$

Substituting (4.4) into (4.3) gives:

$$\rho = \rho_0 + \frac{x^2}{2\rho_0} \quad (4.5)$$

Hence (4.1) can be expressed as:

$$E(x_1, \rho_0) = \int_{\text{line}} \frac{1}{\rho_0} \exp(-jk(\rho_0 + \frac{x^2}{2\rho_0})) dx \quad (4.6)$$

The limits of the integral are $X - L/2$ and $X + L/2$ from Figure 4.1.

Hence,

$$E(x_1, \rho_0) = \frac{\exp(-jk\rho_0)}{\rho_0} \int_{X-L/2}^{X+L/2} \exp(-jk \frac{x^2}{2\rho_0}) dx \quad (4.7)$$

The term $\exp(-jk \frac{x^2}{2\rho_0})$ pertains to the near field radiation pattern, the Fresnel zone of diffraction theory, hence the expression the Fresnel integral. The Fresnel integral can be shown to asymptotically approach the far-field distribution [9]. Using the substitution $t = \pi v^2/2$, (4.7) becomes

$$E(x_1, z) = \frac{\exp(-j2\pi\hat{z})}{\sqrt{2\hat{z}}} \int_{\frac{\pi}{2} v_-^2}^{\frac{\pi}{2} v_+^2} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt \quad (4.8)$$

and the integration limits are

$$v_+ = \frac{\hat{x} + \hat{L}/2}{\sqrt{\hat{z}/2}} \quad v_- = \frac{\hat{x} - \hat{L}/2}{\sqrt{\hat{z}/2}} \quad (4.9)$$

where the wavelength scaled parameters are adopted:

$$\hat{L} = L/\lambda_a \quad \hat{Z} = z/\lambda_a \quad \hat{X} = X/\lambda_a$$

and λ_a is the acoustic wavelength at center frequency.

For each computation the following form of integral (4.8) is adopted:

$$E(x_1, z) = C \left[\int_0^{\frac{\pi}{2} \sqrt{z}} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt - \int_0^{\frac{\pi}{2} \sqrt{z}} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt \right] \quad (4.10)$$

$$\text{where } C = \frac{\exp(-j2\pi\hat{Z})}{\sqrt{2\hat{Z}}}$$

The derivation of the Fresnel integrals (4.8) and (4.10) is shown in Appendix I.

The above form of Fresnel integral is evaluated by the series expansion by J. Boersma [23]:

$$f(x) = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}} dt = C(x) - j S(x) \quad (4.11)$$

For $0 < x < 4$

$$f(x) = e^{-jx} \sqrt{\frac{x}{4}} \sum_{n=0}^{11} (a_n + jb_n) \left(\frac{x}{4}\right)^n \quad (4.11a)$$

For $x > 4$

$$f(x) = \frac{1-j}{2} + e^{-jx} \sqrt{\frac{4}{x}} \sum_{n=0}^{11} (c_n + jd_n) \left(\frac{4}{x}\right)^n \quad (4.11b)$$

The values of coefficients a_n , b_n , c_n and d_n are listed in Appendix II.

4.3 Anisotropy of LiNbO_3

Now consider the anisotropy of the LiNbO_3 crystal (Figure 2.7). The velocity versus the angle of propagation curve is symmetrical about the pure mode axis 0° . It will be shown that this curve can be approximated by an eighth-order polynomial.

The LiNbO_3 crystal exhibits yet another kind of anisotropy [22]. The direction and velocity of the energy propagation do not coincide with the direction and velocity of the phase propagation (Figure 4.2).

FIGURE 4.2: Acoustic beam trajectory in an anisotropic solid [37]

The wave phase fronts travel along the direction of $\vec{k}$ which is normal to the transmitting transducer, but the wave packet energy travels in the direction $\vec{v}_e$ which is at an angle to $\vec{k}$. This means that the receiving transducer must be offset in order to intercept the acoustic pulse [37]. The angle between the two vectors has to be determined in order to get an accurate model for diffraction effects on lithium niobate crystals.

FIGURE 4.3(a):
Slowness surface and energy velocity. The energy velocity is always orthogonal to the slowness surface:

$$\vec{v}_e \cdot d\vec{L} = 0. [10]$$

FIGURE 4.3(b):
Wave surface and propagation direction. The wave plane P is tangential to the wave surface at the extremity of energy velocity vector $\vec{v}_e$ [10]

The energy velocity vector is normal to the slowness surface (Figure 4.3(a)), the reciprocal of the phase velocity as a function of the angle of propagation. For an isotropic substrate the wave vector $\vec{k}$ and the energy velocity $\vec{v}_e$ point in the same direction.

where

$$\begin{aligned} a &= -1.26 \times 10^{-4} \\ b &= 3.6892 \times 10^{-7} \\ c &= -5.0887 \times 10^{-10} \\ d &= 2.7557 \times 10^{-13} \end{aligned}$$

These values of a , b , c and d coefficients give the best fit to the phase velocities between -30° to $+30^\circ$ reported by Szabo and Slobodnik [22].

Now consider

$$k(\theta) = \frac{\omega}{v(\theta)} \quad (4.15)$$

Going back to (4.14) and using (4.13) and (4.15)

$$\begin{aligned} \frac{dk(\theta)}{d\theta} &= \frac{d}{d\theta} \left[\frac{\omega}{v_o (1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8)} \right] \\ &= \frac{\omega}{v_o} \left(\frac{1}{(1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8)} \right)^2 \cdot \\ &\quad - (+2a\theta + 4b\theta^3 + 6c\theta^5 + 8d\theta^7) \end{aligned} \quad (4.16)$$

Thus substituting (4.16) as well as (4.14) and (4.13) into (4.14) yields:

$$\begin{aligned} \tan \phi &= \frac{1}{k(\theta)} \frac{dk(\theta)}{d\theta} \\ &= \frac{v(\theta)}{\omega} \frac{dk(\theta)}{d\theta} \\ &= \frac{v_o (1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8)}{\omega} \cdot \\ &\quad \frac{\omega}{v_o} \left(\frac{1}{1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8} \right)^2 \cdot \\ &\quad - (2a\theta + 4b\theta^3 + 6c\theta^5 + 8d\theta^7) \\ \tan \phi &= \frac{-(2a\theta + 4b\theta^3 + 6c\theta^5 + 8d\theta^7)}{(1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8)} \end{aligned} \quad (4.17)$$

Result (4.17) agrees with the expression by Crabb [25], hence:

$$\phi = \tan^{-1} \left[\frac{1}{v(\theta)} \frac{dv(\theta)}{d\theta} \right] \quad (4.18)$$

where ϕ is the "power flow angle", the angle between the phase vector $\vec{k}(\theta)$ and the energy flow vector $\vec{P}(\theta)$ of Figure 4.4.

FIGURE 4.5: Propagation velocities on the surface of YZ-LiNbO₃ crystal

From Figure 4.5 it can be seen that energy propagates at an angle $(\theta - \phi)$ on LiNbO₃ substrates. The energy velocity vector can be resolved into two components $\vec{v}_p$ and $\vec{v}_{p\perp}$ where $\vec{v}_p$ is equal to the plane wave phase velocity. Since the phase fronts for an infinitely wide wave front would remain constant, one can conclude from Figure 4.5 that:

$$v_g(\theta) = v_e(\theta) = \frac{v_p(\theta)}{\cos\phi} \quad (4.19)$$

$$\text{or } v_p(\theta) = v_e(\theta) \cos\phi \quad (4.19a)$$

$$\text{and } \lambda_p = \lambda_g \cos\phi \quad (4.19b)$$

4.4 Diffraction Integral for LiNbO_3

The approach outlined here is based on the analysis of Cohen [26].

Since it is the energy that has to be intercepted by the output transducer, the velocity and direction of propagation of the energy vector $\vec{v}_e$ must be considered.

The dependence of the phase velocity $\vec{v}_p$ on the angle of propagation θ was stated in (4.14) as:

$$v_p = v_0 (1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8)$$

$$\text{now } v_e = \frac{v_p}{\cos\phi}$$

$$= \frac{v_0}{\cos\phi} (1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8) \quad (4.20)$$

$$\text{and } k_e = \frac{\omega}{v_e}$$

$$= \frac{\omega}{v_0} \cdot \frac{\cos\phi}{1 + a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8} \quad (4.21)$$

$$\text{where } \frac{\omega}{v_0} = k_0$$

For small values of $a\theta^2$, $b\theta^4$, $c\theta^6$, $d\theta^8$ the small number approximation can be made as follows:

$$k_e \approx k_0 \cos\phi (1 - a\theta^2 - b\theta^4 - c\theta^6 - d\theta^8) \quad (4.22)$$

$$\text{Let } (-a\theta^2 - b\theta^4 - c\theta^6 - d\theta^8) = \gamma$$

$$\text{hence } k_e \approx k_0 \cos\phi (1 + \gamma) \quad (4.23)$$

Now one must modify the Fresnel integral to accommodate the energy velocity parameters.

FIGURE 4.6: Geometrical configuration for evaluating the effect of power flow angle ϕ on the complex field strength at point P.

From Figure 4.6 it is apparent that

$$\tan(\theta - \phi) = x_1 / \rho_0 \quad (4.24)$$

where ϕ is a negative angle

$$\text{hence } x_1 = \rho_0 \tan(\theta - \phi) \quad (4.24a)$$

x_1 now has to appear in the limit of the integral as $x_1 \pm L/2$.

Hence substituting the energy velocity parameters into (4.7) yields:

$$E(x_1, \rho_0) = \frac{\exp(-jk_e \rho_0)}{\rho_0} \int_{x_1 - L/2}^{x_1 + L/2} \exp\left\{-jk_e \frac{x^2}{2\rho_0}\right\} dx \quad (4.25)$$

Transforming the above Fresnel integral into its classical form where $z \equiv \rho_0$ and using acoustic wavelength scaled parameters:

$$E(x_1, z) = \frac{\exp(-j2\pi\hat{z} \cos\phi(1+\gamma))}{\sqrt{2\hat{z}_i \cos\phi(1+\gamma)}} \int_{\frac{\pi}{2}v_-^2}^{\frac{\pi}{2}v_+^2} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt \quad (4.26)$$

where the limits v_- and v_+ are:

$$v_+ = \frac{\hat{x}_1 + \hat{L}/2}{\sqrt{\hat{z}/2}} [\cos\phi(1+\gamma)]^{\frac{1}{2}} \quad (4.26a)$$

$$v_- = \frac{\hat{x}_1 - \hat{L}/2}{\sqrt{\hat{z}/2}} [\cos\phi(1+\gamma)]^{\frac{1}{2}} \quad (4.26b)$$

The step by step derivation of these results is given in Appendix 3.

4.5 Modelling of Diffraction [3]

The effects of diffraction of the output apodized transducer are the dominant ones. The diffraction due to the input non-apodized transducer is of lesser importance. The approach taken here assumes that the diffraction pattern due to the fingers of the apodized IDT will vary only a little from finger to finger of the input non-apodized IDT. Hence the diffraction pattern of each finger of the output IDT is computed at the center of the non-apodized IDT (Figure 4.7).

FIGURE 4.7: The diffraction pattern of each finger of the output transducer is evaluated at the center of the input transducer

Each complex field point of the diffraction pattern is computed using the Fresnel integral, i.e., if 100 points are required across the output aperture then 100 Fresnel integrals must be solved. In order to get the final complex finger weighting the values of the field points across the whole aperture have to be totalled, i.e., integrated. This yields one complex value for each finger of the output IDT. However, the phase of each finger has to be corrected for by subtracting the phase shift between each finger and the center of the input IDT. Hence, the complex weights as computed at the center of the input IDT are placed back at their original position in the output apodized IDT.

The next step is the complex numerical convolution between the input and output IDT to obtain the filter's complex impulse response. Applying the FFT for complex numbers to the impulse response yields the filter's frequency response with the effects of diffraction included.

For the anisotropic case the modelling becomes more complicated due to the introduction of the variation in phase velocity and power flow angle with the direction of propagation.

The term outside the integral of (4.26) is the phase delay term. There is an error in this delay time for long transmitting fingers. It is assumed that this delay is approximately constant for the limits of integration, i.e., between the center of the finger and the extreme ends. For long transmitting fingers this assumption is no longer true and a significant phase error is encountered. To avoid this problem the long finger is subdivided into segments no longer than 2 wavelengths each (Figure 4.8). For distances of $z \approx 100 \lambda_0$, the width of $L = 2\lambda_0$ gives a maximum phase error of approximately $\pm 20^\circ$ on YZ-LiNbO₃ [28].

FIGURE 4.8: A composite diffraction profile is obtained by adding the contribution of the diffraction profiles of the individual subelements

From Figure 4.8, it follows that the delays $D1$ to $D4$ are equal. Thus adding up the contributions of the individual segments $P1$ to $P4$ over the output aperture width, $-L/2$ to $+L/2$, will yield the correct finger weight after diffraction. The rest of the process is the same as for the isotropic case: 1) shift the diffracted weights back to their original position; 2) convolve the two IDT's; and 3) take the FFT of the impulse response to get the desired frequency response.

Finally, compare the two models with the experimental results (Figure 4.9, 4.10). There is a discrepancy in both cases between the model and experiment on the high frequency side. This is due to the coupling of the surface wave with the shear bulk wave [13]. This coupling peaks at about 77 MHz which explains the loss of the first null of the $\sin x/x$ at 77.3 MHz.

The anisotropic model appears to be the better one since the individual nulls follow the experimental results more closely.

FIGURE 4.9: The isotropic model versus experimental results of 70 FIL02

FIGURE 4.10: The anisotropic model versus experimental results of 70 FIL102

CHAPTER V

COMPENSATION ALGORITHM

5.1 Introduction

Compensation for the effects of diffraction can be accomplished in the time domain or in the frequency domain. In this thesis the problem has been approached via time domain considerations [28]. The diffraction compensation algorithm has been based on the work of Mitchell and Parker [29] and [30,31,32,33,34].

A single transmitting finger can be decomposed into two split fingers which are separated spatially by a quarter wavelength as shown in Figure 5.1. This arrangement offers an additional degree of freedom in the filter design.

Having calculated the diffraction weights for these split fingers and comparing them to the original transmitted weight, a phase and an amplitude error is noted. The amplitude correction is achieved by adjusting the finger overlaps of both split fingers simultaneously until the diffracted weights take on the desired value. The compensation for the phase error is accomplished by adjusting the relative amplitudes of the split fingers.

FIGURE 5.1: (a) Single electrode structure IDT
(b) Double electrode (split finger) structure IDT

The successive fingers in the single-electrode IDT are separated spatially by a half wavelength while the successive fingers in the double electrode IDT are separated spatially by a quarter wavelength.

5.2 Development of the Diffraction Compensation Algorithm [28]

The transmitted single vector sample TW (Figure 5.2) is decomposed into the left split finger TWL and the right split finger TWR. The two split fingers are 90 degrees out of phase with respect to each other. The left split finger TWL is leading TW by 45 degrees while the right split finger TWR is lagging TW by 45 degrees. At a certain observation point, i.e., the center of the receiving transducer, the change in the transmitted sample's weighting due to diffraction has to be evaluated (Figure 5.3).

From Figure 5.3 it follows that both samples TWL and TWR travel the same distance $R\theta$, when their change in weighting due to diffraction is evaluated. The new sample weights are DWL for the left split finger and DWR for the right one. To obtain the single weight DWSN, which represents TW after diffraction, the contributions of DWL and DWR have to be added together. Due to diffraction there is a phase offset ϕ_k between TW and DWSN as well as a difference in their respective magnitudes. In order to compensate for these discrepancies, the phasor CWSN is transmitted in place of the phasor TW. The phasor CWSN is the complex conjugate of DWSN such that after diffraction, approximately TW is received. CWSN is decomposed into the left and right component CWL and CWR respectively. These two phasors represent the new weighting of the split fingers TWL and TWR, originally designed to transmit TW.

From the trigonometry of Figure 5.2 the relationships between all the phasors under consideration are derived.

$$\overline{DWL} = |DWL| \angle 45^\circ + \phi_k \quad (5.1a)$$

$$\overline{DWR} = |DWR| \angle \phi_k - 45^\circ \quad (5.1b)$$

FIGURE 5.3: Evaluation of the diffracted weights DWL and DWR of the split transmitting fingers TWL and TWR.

Since the magnitudes of $\overline{DWL}$, and $\overline{DWR}$ are equal, a common term can be used.

$$|\overline{DWL}| = |\overline{DWR}| = |\overline{DW}|$$

The magnitude of $|\overline{DW}|$ relates to the magnitude of $\overline{DWSN}$ in the following way:

$$|\overline{DW}| = |\overline{DWSN}| / \sqrt{2} \quad (5.2)$$

To generate the single phasor $\overline{DWSN}$, the contributions of $\overline{DWL}$ and $\overline{DWR}$ have to be summed:

$$\begin{aligned} \overline{DWSN} &= \overline{DWL} + \overline{DWR} \\ &= |\overline{DW}| \angle 45^\circ + \phi_k + |\overline{DW}| \angle \phi_k - 45^\circ \end{aligned} \quad (5.3)$$

Performing this summation in complex form yields:

$$\begin{aligned}\overrightarrow{DWSN} &= |DW| [\cos(45^\circ + \phi_k) + j\sin(45^\circ + \phi_k) \\ &\quad + \cos(\phi_k - 45^\circ) + j\sin(\phi_k - 45^\circ)] \\ &= |DW| [2 \cos 45^\circ \cos \phi_k + j2 \cos 45^\circ \sin \phi_k]\end{aligned}\quad (5.4)$$

but $|DW| = |\overrightarrow{DWSN}| / \sqrt{2}$ from equation (5.2)

hence:

$$\overrightarrow{DWSN} = |\overrightarrow{DWSN}| [\cos \phi_k + j\sin \phi_k] \quad (5.5)$$

The result (5.5) agrees with Figure 5.2.

Next, derive the expression for $\overrightarrow{CWL}$ and $\overrightarrow{CWR}$. Their sum must be the complex conjugate of $\overrightarrow{DWSN}$.

$$\begin{aligned}\overrightarrow{CWSN} &= \overrightarrow{CWL} + \overrightarrow{CWR} \\ &= |CWL| \angle +45^\circ + |CWR| \angle -45^\circ\end{aligned}\quad (5.6)$$

Find the expression for $|CWL|$ and $|CWR|$ from Figure 5.2.

$$TWL = |CWL| \cos 45^\circ \quad (5.7a)$$

$$\text{and } TWL = |CWL| \cos(45^\circ + \phi_k) \quad (5.7b)$$

equating (5.7a) and (5.7b) yields:

$$|CWL| = |DWL| \frac{\cos(45^\circ + \phi_k)}{\cos 45^\circ} \quad (5.7c)$$

Substituting for $|DWL|$ expression (5.2)

$$|CWL| = |\overrightarrow{DWSN}| \cos(45^\circ + \phi_k) \quad (5.8)$$

Similarly for $|CWR|$:

$$TW2 = |CWR| \cos(-45^\circ) \quad (5.9a)$$

$$TW2 = |DWR| \cos(\phi_k - 45^\circ) \quad (5.9b)$$

$$\text{hence } |CWR| = |DWR| \frac{\cos(\phi_k - 45^\circ)}{\cos(-45^\circ)} \quad (5.9c)$$

$$\text{and } |CWR| = |DWSN| \cos(\phi_k - 45^\circ) \quad (5.10)$$

Substituting (5.8) and (5.10) into (5.6)

$$\begin{aligned} \overrightarrow{CWSN} &= |DWSN| [\cos(45^\circ + \phi_k) (\cos 45^\circ + j \sin 45^\circ) \\ &\quad + \cos(\phi_k - 45^\circ) (\cos 45^\circ - j \sin 45^\circ)] \\ &= |DWSN| (\cos \phi_k - j \sin \phi_k) \end{aligned} \quad (5.11)$$

Equation (5.11) is the complex conjugate of equation (5.5) and hence

$$\overrightarrow{CWSN} = \overrightarrow{DWSN}^*$$

The phase correction for diffraction is implemented by using expressions (5.8) and (5.10) which adjust the relative length of the left and right split fingers. The magnitude correction is done by simple subtraction of the desired and received sample magnitude:

$$|CWL| = |CWL| [1 + |TW| - |DWSN|] \quad (5.12)$$

5.3 Computer algorithm

The flow graph of the computer compensation algorithm is shown in Figure 5.4. It outlines the steps in the sequence they were taken, however, not all the details are included.

FIGURE 5.4: Flow graph of diffraction compensation algorithm

71

First, the maximum number of iterations ITMAX is established to avoid excessive computing time. The magnitude relaxation factor RLXM and the phase relaxation factor RLXP are also initiated as well as the minimum acceptable amplitude error. All the transmitting fingers TW are generated and then each one of them is decomposed into the left and right split fingers TWL and TWR respectively (Figure 5.2). Next, the diffracted weights DWL and DWR of the split fingers TWL and TWR are computed at the center of the output transducer.

The diffracted weight sum normalized DWSN is then obtained as the phasor addition of DWL and DWR and the amplitude and phase of DWSN are compared to those of TW. The error in magnitude ERRLIN and the error in phase ERRPHS are hence established for each iteration. Once the amplitude and phase error reach a sufficiently low value, e.g., the amplitude error should be less than 10^{-8} and the phase error one degree maximum, or the number of allowable iterations is exceeded, the impulse response and the frequency response of the compensated design are generated.

The predicted diffraction compensated frequency response based on the isotropic model is shown in Figure 5.5 and the one based on the anisotropic model is shown in Figure 5.6. The isotropic compensation yields sidelobe levels of -53 dB while the anisotropic compensation indicates -46 dB. The anisotropic compensation algorithm diverged after the fifteenth iteration and the frequency response shown in Figure 5.6 is the best one obtained before divergence. Further refinement of the numerical techniques is necessary to improve the compensated frequency response based on the anisotropic model.

However, the evaluation of the two compensated designs must be based on experimental results given in Chapter VI.

FIGURE 5.5: The predicted compensated frequency response based on the isotropic model compared to the Dirac delta model

FIGURE 5.6: The predicted compensated frequency response based on the anisotropic model compared to the Dirac delta model

CHAPTER VI

TESTING AND EXPERIMENTAL RESULTS

6.1 Time Domain Measurements

A measurement set-up is described here which enables time domain testing of a SAW device's pulse response [35] to be performed. The same set-up isolates spurious time domain signals from the swept frequency amplitude and phase measurements, hence, the effects of diffraction on the frequency response are isolated.

The test set consists of a network analyzer and two time gates, one at the input and one at the output of the SAW filter as shown in Figure 6.1. The time gates each contain a pulse generator and a balanced mixer which are used to pulse modulate the input and the output signals of the device under test.

• FIGURE 6.1: Block diagram of the test set for pulse response measurements

The step-by-step description of the time domain processor follows.

The RF output of the tracking generator generates a 70 MHz sinewave (Figure 6.2(a)). Time gate 1 sends out a narrow pulse (Figure 6.2(b)), approximately 12 RF cycles wide, which multiplies with the RF signal and the final result is an RF pulse (Figure 6.2(c)) which excites the SAW filter's impulse response. The pulse of time gate 2 is delayed with respect to that of time gate 1 as can be seen in Figure 6.2(e). The amount of delay corresponds directly to the time response of the SAW filter (Figures 6.2(d) and (f)). The position of the second pulse is such as to isolate the desired SAW impulse response, the TTE or reflections. The output of time gate 2 is connected to the RF input of the spectrum analyzer and the frequency domain response is observed.

o

FIGURE 6.2: Time gated frequency response measurement

- a) RF output from tracking generator
- b) Time gate 1
- c) Input signal to the test sample
- d) Output signal from the test sample
- e) Time gate 2
- f) RF input to the spectrum analyzer
- g) Signal evaluated in the spectrum analyzer

6.2 Practical Implementation

In order to realize the block diagram of Figure 6.1 some practical aspects of the set-up had to be considered. Pulse generator 1 has to be triggered externally from the tracking generator so that the pulse generator is synchronized. Pulse generator 2 is also triggered externally but from pulse generator 1. Hence the source and the two pulse generators are in synchronism. The pulse rate of generator 1 has to allow an adequate amount of time for pulse 2 to window on to the desired SAW response (Figure 6.2(f)). If the pulse rate were 70 MHz then the separation of pulses would be $.014 \mu\text{s}$. But the TTE is $6.26 \mu\text{s}$ delayed from the instant the time domain response is excited. If the pulse rate were to remain 70 MHz the window of pulse 2 could not be applied. Hence the 70 MHz RF signal has to be divided down to allow for the full SAW time domain response including the TTE to be observed between pulses. A division by 1000 satisfies this requirement. The pulse rate is then 70 kHz which corresponds to $14.3 \mu\text{s}$ between pulses. The block diagram is now like that of Figure 6.3. There is a 3 dB power splitter at the output of the tracking generator which adjusts the power level to the frequency divider consisting of the Emitter Coupled Logic (ECL) gates. Each gate is a MC10137 decade counter. Three of these will be connected sequentially in order to achieve the desired division count (Appendix IV). The mixers used are HP10514A and HP10534A balanced mixers and for better isolation between the local and RF ports two mixers are connected in series. There is a 20 dB C-COR 3580 amplifier both at the input and at the output of the SAW filter. Both the input pulse and the output of the SAW filter are displayed on the oscilloscope. By moving the window of pulse 2 and adjusting its width (Figure 6.4) the SAW impulse response is isolated from the secondary reflections and its frequency response is observed on the spectrum analyzer (Figure 6.7).

FIGURE 6.3: Block diagram of RF pulse measurement technique to isolate main signal from spurious coupling in SAW filter

FIGURE 6.4: Delay of the different signals of a 70 MHz SAW filter

6.3 Experimental Results

In this section the experimental results are presented. The mask of one of the devices fabricated is shown in Figure 6.5. The bars around the outside are for alignment of the mask on the wafer. They also serve as a guideline for the sawing of the wafer into the individual dice. The two diagonal bars between the two transducers are electrically grounded in order to reduce the electromagnetic feed-through.

FIGURE 6.5: Mask of a 70 MHz SAW filter

6.3.1 Diffraction Un-compensated Filter

Figure 6.6(a) shows the frequency response of the 70 MHz filter. The vertical scale is 10 dB/div and the sweep is 70 ± 25 MHz. All of the spurious signals are included. Figure 6.6(b) depicts the significant ripples due to the reflections from the end of substrate in the passband of the device. The vertical scale is 2 dB/div and the sweep about $f_0 = 70$ MHz is 10 MHz.

Figures 6.7(a) and (b) present the frequency response of the 70 MHz filter without the spurious interferences. The window of pulse 2 is positioned exactly as indicated in Figure 6.4, hence, the response of the main SAW signal has been isolated from the delayed reflections and the TTE. Diffraction raises the level of the sidelobes (see Figure 1.4) and reduces the depth of the nulls of the sinc function due to the non-apodized transducer.

There is a significant slope in the passband of the 70 MHz filter fabricated on the YZ-LiNbO₃ substrate (Figure 6.7(b)). Figure 6.7(a) shows a noticeable difference between the level of the sidelobes at the high frequency side of the passband and the low frequency one. The sidelobes at the high frequency side are quite a bit higher than those at the low frequency side.

The cause of this asymmetry was established to be due to the strong coupling with the surface shear bulk wave [13,14,15]. This shear bulk wave travels at a velocity only 3 percent higher than the center frequency of the filter. The increase in the coupling level causes the slope in the passband to increase with frequency and the high frequency sidelobe level to increase. The bulk wave signal arrives at almost the same delay as the main SAW signal and therefore cannot be therefore isolated by the window of pulse 2. The only way to avoid it is to use a different crystal cut of LiNbO₃, for example the 128° rotated Y-cut.

FIGURE 6.6(a): The measured frequency response of the 70 FIL102 device with spurious signal interference

FIGURE 6.6(b): The measured passband response of the 70 FIL102 device showing ripples due to end of substrate reflections

FIGURE 6.7(a): The measured frequency response of the 70 FIL102 device with spurious interference removed

FIGURE 6.7(b): The measured passband response of the 70 FIL102 device showing response without ripples from end of substrate reflections

Another diffraction un-compensated filter 70 FIL102 was fabricated (Figure 6.8), but this time the substrate bottom was roughened in order to disperse the bulk wave. Note the improvement in the shape of the sidelobes as compared to Figure 6.6(a) and the depth of the nulls due to the un-apodized transducer. However, the bulk wave coupling is still degrading the frequency response of 70 FIL102 in the passband and at the high frequency side. The intensity of the bulk wave is shown in Figure 6.9. The surface of the substrate has been coated with viscous material which absorbs the Rayleigh wave, hence only the bulk wave response is observed on the network analyzer. The upper frequency cutoff is due to the backside roughening. The bulk wave frequency response peaks at about 77 MHz where it is only 40 dB below the main SAW signal.

This strong bulk wave coupling masks the first upper null of 70 FIL102 which occurs at 77.3 MHz. The second null is at 84.6 MHz where the bulk wave coupling is 60 dB below the main SAW response, hence there is a well-pronounced deep null at that frequency.

6.3.2 Diffraction Compensated Filter

Three types of diffraction compensated filters were fabricated. Two are based on the anisotropic diffraction model (Figures 6.11 and 6.12) and one is based on the isotropic diffraction model (Figure 6.10). All three filters show an improvement in the passband and in the sidelobe suppression when compared to the un-compensated design (Figure 6.8). The filters of Figures 6.10 and 6.11 have a 30 degree incline in the apodized IDT. The filter in Figure 6.12 has no incline in the apodization. All three filters show the loss of the first upper null due to bulk wave coupling. The filter 70 FIL201 which was designed according to the isotropic model shows an 8 dB improvement in the first sidelobe suppression. 70 FIL202, the diffraction compensated filter based on the anisotropic model indicates a 5 dB improvement in the first sidelobe level. 70 FIL203

FIGURE 6.8: The measured frequency response of the filter 70 FIL102 with the substrate bottom roughened to disperse the bulk wave.

FIGURE 6.9: The measured bulk wave coupling affecting the frequency response of filter 70 FIL102

27. R. Veith, G. Eberharter, H.P. Feuerbaum and U. Knauer, "Visualization of SAW Propagation with the Scanning Electron Microscope", IEEE Ultrasonics Sump. Proc., Vol. 1, 80CH1602-2, pp. 348-351, 1980.
28. Mark S. Suthers, private communication.
29. R.F. Mitchell and D.W. Parker, "Synthesis of Acoustic-Surface-Wave Filters Using Double Electrodes", Electron. Lett., 10, pp. 512, 1974.
30. M. Hikita et al, "Phase Weighting for Low Loss SAW Filters", IEEE Ultrasonics Symposium, Vol. 1, 80CH1602-2 pp. 308-312, 1980.
31. R.W. Cohen and C.M. Panasik, "Sidelobe Compensation of Double Electrode SAW Transducers on Lithium Niobate", IEEE Ultrasonics Symposium, Vol. 1, 82CH1823-4, pp. 18-22, 1982.
32. E.B. Savage and G.L. Mattheaei, "Compensation for Diffraction in SAW Filters", IEEE Ultrasonics Symposium, 79CH1482-9, pp. 527-532, 1979.
33. M.S. Suthers and C.K. Campbell, "Use of Charge Distribution Model in SAW Bandpass Filter Design", IEEE Ultrasonics Symposium, 79CH1482-9, pp. 565-569, 1979.
34. M.S. Suthers, "A Design Algorithm for Arbitrary Phase and Amplitude Surface Acoustic Wave Filters", Ph.D. Thesis, McMaster University, Hamilton, Canada 1978.
35. K. Langecker and R. Veith, "Automatic Time Domain Testing of SAW Devices", IEEE Ultrasonics Symp. Proc., Cat. no. 80 Ch. 1602-2, p. 396, 1980.
36. M.S. Kharusi and G.W. Farnell, "Diffraction and Beam Stearing for Surface-Wave Comb Structures on Anisotropic Substrates", IEEE Trans. on Sonics and Ultrasonics, SU-18, pp. 35-42, 1971.
37. B.A. Auld, Acoustic Fields and Waves in Solids, John Wiley & Sons, New York, N.Y., Vol. I, ch. 7, 1973.
38. A.B. Glaser, G.E. Subak-Sharpe, Integrated Circuit Engineering, Addison-Wesley Publishing Corp., Reading, Mass., 1977.

FIGURE 6.10: The measured frequency response of the diffraction compensated filter based on the isotropic diffraction model (70 FIL201)

FIGURE 6.11: The measured frequency response of the diffraction compensated filter based on the anisotropic diffraction model (70 FIL202)

FIGURE 6.12: The measured frequency response of the diffraction compensated filter based on the anisotropic diffraction model (70 FIL203). There was no incline in the apodized IDT.

has the best frequency response of all the diffraction compensated filters. The first sidelobes of 70 FIL203 are 9 dB below the level of the un-compensated 70 FIL102.

The shape of the passband of 70 FIL203 appears to be slightly different from that of 70 FIL201 and 70 FIL202. However, the nulls are at exactly the same frequencies. The change in the passband characteristic is attributed to the fact there is no incline in the design of the apodized transducer. Due to lack of time there was no diffraction un-compensated filter with no incline fabricated for comparison.

The diffraction compensated filter 70 FIL202, which is based on the anisotropic model, had an error in the mask layout. There is a slight discrepancy in the magnitude of the finger overlaps along the

apodization incline and there is a good chance of an improved sidelobe suppression if this error is corrected.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis contains a basic treatment of the diffraction effects which are common to surface acoustic wave devices. The study has been carried out at both the theoretical and experimental levels, with good agreement.

The experimental results, however, do not yield a clear indication as to which diffraction compensation algorithm is more effective - the isotropic one or the anisotropic one. The anisotropic diffraction model is thought to be far superior to the isotropic one but the performance of the compensated filter based on the anisotropic model was anticipated to be better than what the experimental results indicate.

At present, an interesting problem arises when using the anisotropic model in the diffraction compensation algorithm, namely, the algorithm diverges after the fifteenth iteration. This causes the predicted first sidelobe level to be at only -46 dB and the experimental results indicate -41 dB for 70 FIL 203. Compared to this predicted frequency response the experiment is in good agreement.

The predicted frequency response for the isotropic diffraction correction indicates sidelobe levels of -53 dB but the experiment shows the first sidelobes at -40 dB. From this point of view the anisotropic compensated design is more successful and further refinement of the numerical techniques may bring significant improvements in the experimental results based on the anisotropic model.

A diffraction model based on the understanding of the physics of the SAW propagation behaviour on a highly anisotropic crystal such as YZ-LiNbO₃ has been developed. The novel approach taken to modelling the velocity versus the angle of propagation characteristic of YZ-LiNbO₃ with an 8th-order polynomial is considered successful. The experimental results indicate a 9 dB improvement in sidelobe levels for YZ-LiNbO₃.

7.2 Future Work

Future work in the area of this thesis may include some, or all of the following points.

1. A comprehensive analysis of the velocity profile of the YZ-LiNbO₃ substrate is a worthwhile task to be undertaken. The velocities used are based on those published by Szabo, and Slobodnik in 1973 [24]. It is felt that with today's technology more accurate velocities for YZ-LiNbO₃ may be obtained.
2. An improvement of the diffraction model, by including the effect of the velocity change as the wavefront propagates across the metallized and unmetallized parts of the transducer, is recommended.
3. To further improve the diffraction model, the diffraction profiles may be generated at each receiving finger rather than just at the center of the receiving transducer. A more complex convolutional process will have to be implemented but the procedure will remain analogous to that outlined in Chapter IV.
4. To establish the diffraction model and the diffraction compensation algorithm for YZ-LiNbO₃, developed in this thesis, as a solid benchmark further experimental results must be obtained for a variety of distances between transducers and for a number of different aperture widths as well as transducer lengths.
5. There is a great deal of room left for improving the compensation time for the diffraction model. This can be achieved by using advanced numerical techniques and structured

programming. In the diffraction compensation algorithm there can be a significant saving in computation time achieved by developing a more efficient algorithm which will reduce the number of iterations before convergence.

6. A major contribution towards improving the filter's frequency response can be made by developing a design algorithm to counteract the effects of shear bulk wave coupling on the high frequency side of the passband. This may be accomplished by modifying the design algorithm presented in Chapter III.
7. Finally, it is felt that the techniques developed in this thesis for diffraction compensation on YZ-LiNbO₃ may be extended to other cuts of LiNbO₃ or to other substrate materials. It may be an interesting task to investigate the validity of this suggestion both theoretically and experimentally for a number of different substrate materials.

References

1. R.M. White and F.W. Voltmer, "Direct Piezoelectric Coupling to Surface Elastic Waves", Applied Physics Letters, Vol. 7, pp. 314-316, December 1965.
2. G.W. Farnell and E.L. Adler, "An Overview of Acoustic Surface-Wave Technology", Final Report to Communications Research Centre on DSS Contract 36001-3-4406, August 12th, 1974.
3. I. Streibl, B. Syrett, M. Suthers, D. Conn, "SAW Diffraction Compensation on YZ-LiNbO_3 ", IEEE Ultrasonics Symposium, 1983.
4. W.A. Radasky and G.L. Matthaei, "Fast Computation of Diffraction in General Anisotropic Media by Use of the Geometrical Theory of Diffraction", IEEE Trans. on Sonics and Ultrasonics, Vol. 30, No. 2, March 1983.
5. Edward B. Savage and George L. Matthaei, "A Study of Some Methods for Compensation for Diffraction in SAW IDT Filters", IEEE Trans. on Sonics and Ultrasonics, Vol. SU-28, No. 6, pp. 439-448, November 1981.
6. W.R. Mader, "Universal Method for Compensation of SAW Diffraction and Other Second Order Effects", IEEE Ultrasonics Symp. Proc., Vol. 1, pp. 23-28, 1982.
7. David Penuri, "A Numerical Technique for SAW Diffraction Simulation", IEEE Trans. on Microwave Theory and Techniques, Vol. MTT-26, No. 4, April 1978.
8. Thomas L. Szabo and Andrew Y. Slobodnik, Jr., "The Effect of Diffraction on the Design of Acoustic Surface Wave Devices", IEEE Trans. on Sonics and Ultrasonics, SU-20, pp. 240-251, 1973.
9. John M. Stone, Radiation and Optics: An Introduction to the Classical Theory, McGraw-Hill Inc., New York, N.Y., 1983.
10. E. Dieulesaint, D. Royer, "Elastic Waves in Solids", John Wiley & Sons Ltd., Chichester, G.B., 1980.
11. H. Matthews ed., Surface Acoustic Wave Filters, Design, Construction and Use, John Wiley & Sons Inc., New York, N.Y., 1977.
12. E.L. Adler and G.W. Farnell, "Bandpass Filters Using Surface Acoustic Waves", BNR Technical Report TR 1E91-79-95, BNR, March 1979.

13. Richard F. Mitchell and Eileen Read, "Suppression of Bulk Wave Radiation from Surface Acoustic Wave Devices", IEEE Trans. on Sonics and Ultrasonics, SU-22, pp. 263-270, 1975.
14. M.H.C. Reilly, R.F. Milson, M. Redwood, "Generation of Rayleigh and Bulk Waves by Interdigital Transducers on Y-cut, Z-propagating Lithium Niobate", Electron. Lett., 9, pp. 419-420, 1973.
15. R. Larosa, G.F. Vasilis, "Broadband Bulk Wave Cancellation in Acoustic-Surface-Wave Devices", Electron. Lett., 8, pp. 478-479, 1972.
16. W.L. Stutzman & G.A. Thiele, Antenna Theory and Design, John Wiley & Sons Inc., New York, N.Y., 1981.
17. John D. Kraus, Antennas, McGraw-Hill Inc., New York, N.Y., 1950.
18. B.D. Steinberg, Principals of Aperture and Array System Design, including random and adaptive arrays, John Wiley & Sons Inc., New York, N.Y., 1976.
19. Erwin Kreyszig, Advanced Engineering Mathematics 3rd edition, John Wiley & Sons Inc., New York, N.Y., 1972.
20. Alan V. Oppenheim, Ronald W. Schaffer, Digital Signal Processing, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1985.
21. E.O. Brigham, R.E. Morrow, "The Fast Fourier Transform", IEEE Spectrum, pp. 63-70, December 1967.
22. Frederic J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proceedings of the IEEE, Vol. 66, No. 1, pp. 51-83, January 1978.
23. J. Boersma, "Computation of Fresnel Integrals", J. Math. Comp., V. 14, p. 380, 1960.
24. T.L. Szabo, A.J. Slobodnik, Jr., "Acoustic Surface Wave Diffraction and Beam Steering", United States Air Force Technical Report, AFCL-TR-73-0302, May 1973.
25. J.C. Crabb, J.D. Maines, N.R. Ogg, "Surface-wave Diffraction on LiNbO_3 ", Electron. Lett., 7, pp. 253-255, 1971.
26. M.G. Cohen, "Optical Study of Ultrasonic Diffraction and Focusing in Anisotropic Media", Journal of Applied Physics, Vol. 38, No. 10, pp. 3821-3828, September 1967.

27. R. Veith, G. Eberharter, H.P. Feuerbaum and U. Knauer, "Visualization of SAW Propagation with the Scanning Electron Microscope", IEEE Ultrasonics Sump. Proc., Vol. 1, 80CH1602-2, pp. 348-351, 1980.
28. Mark S. Suthers, private communication.
29. R.F. Mitchell and D.W. Parker, "Synthesis of Acoustic-Surface-Wave Filters Using Double Electrodes", Electron. Lett., 10, pp. 512, 1974.
30. M. Hikita et al, "Phase Weighting for Low Loss SAW Filters", IEEE Ultrasonics Symposium, Vol. 1, 80CH1602-2 pp. 308-312, 1980.
31. R.W. Cohen and C.M. Panasik, "Sidelobe Compensation of Double Electrode SAW Transducers on Lithium Niobate", IEEE Ultrasonics Symposium, Vol. 1, 82CH1823-4, pp. 18-22, 1982.
32. E.B. Savage and G.L. Matthaui, "Compensation for Diffraction in SAW Filters", IEEE Ultrasonics Symposium, 79CH1482-9, pp. 527-532, 1979.
33. M.S. Suthers and C.K. Campbell, "Use of Charge Distribution Model in SAW Bandpass Filter Design", IEEE Ultrasonics Symposium, 79CH1482-9, pp. 565-569, 1979.
34. M.S. Suthers, "A Design Algorithm for Arbitrary Phase and Amplitude Surface Acoustic Wave Filters", Ph.D. Thesis, McMaster University, Hamilton, Canada 1978.
35. K. Langecker and R. Veith, "Automatic Time Domain Testing of SAW Devices", IEEE Ultrasonics Symp. Proc., Cat. no. 80 Ch. 1602-2, p. 396, 1980.
36. M.S. Kharusi and G.W. Farnell, "Diffraction and Beam Steering for Surface-Wave Comb Structures on Anisotropic Substrates", IEEE Trans. on Sonics and Ultrasonics, SU-18, pp. 35-42, 1971.
37. B.A. Auld, Acoustic Fields and Waves in Solids, John Wiley & Sons, New York, N.Y., Vol. I, ch. 7, 1973.
38. A.B. Glaser, G.E. Subak-Sharpe, Integrated Circuit Engineering, Addison-Wesley Publishing Corp., Reading, Mass., 1977.

APPENDIX I

FRESNEL INTEGRAL

Consider the Fresnel integral,

$$E(x_1, \rho_o) = \frac{\exp(-jk\rho_o)}{\rho_o} \int_{x=L/2}^{x=L/2} \exp\left(-jk \frac{x^2}{2\rho_o}\right) dx \quad (1)$$

where $k = \frac{2\pi}{\lambda_a}$. Substituting for k into (1) yields:

$$\begin{aligned} E(x_1, \rho_o) &= \frac{\exp(-jk\rho_o)}{\rho_o} \int_{x=L/2}^{x=L/2} \exp\left(-j \frac{2\pi}{\lambda_a} \cdot \frac{x^2}{2\rho_o}\right) dx \\ &= \frac{\exp(-jk\rho_o)}{\rho_o} \int_{x=L/2}^{x=L/2} \exp\left(-j\pi \frac{x^2}{\lambda_a \rho_o}\right) dx \end{aligned} \quad (2)$$

$$\text{Let } \frac{v^2}{2} = \frac{x^2}{\lambda_a \rho_o}$$

$$\text{or } x = \sqrt{\frac{\lambda_a \rho_o}{2}} v \quad (\text{note } \rho_o \equiv z) \quad (3)$$

$$\text{Then } \frac{2v dv}{2} = \frac{2x dx}{\lambda_a \rho_o}$$

$$dx = \frac{\lambda_a \rho_o}{2} \frac{v dv}{x}$$

$$\text{Using (3): } dx = \frac{\lambda_a \rho_o v dv}{2 \sqrt{\frac{\lambda_a \rho_o}{2}} v} = \sqrt{\frac{\lambda_a \rho_o}{2}} dv \quad (4)$$

Substituting (4) into (2) gives:

$$\begin{aligned}
 E(x_1, \rho_o) &= \frac{\exp(-jk\rho_o)}{\rho_o} \int_{\text{line}} \exp\left(-j\pi \frac{v^2}{2}\right) dv \sqrt{\frac{\lambda \rho_o}{2}} \\
 &= \frac{\exp(-jk\rho_o)}{\rho_o} \sqrt{\frac{\lambda \rho_o}{2}} \int_{\text{line}} \exp\left(-j\pi \frac{v^2}{2}\right) dv \quad (5)
 \end{aligned}$$

Now transform the limits using (3)

When $x = X + L/2$

$$\begin{aligned}
 \sqrt{\frac{\lambda \rho_o}{2}} v &= X + L/2 \\
 \text{or } v_+ &= \frac{X + L/2}{\sqrt{\frac{\lambda \rho_o}{2}}} = \frac{X + L/2}{\sqrt{\frac{\lambda z}{2}}} \quad (\rho_o \equiv z)
 \end{aligned}$$

Similarly, the lower limit is derived as

$$v_- = \frac{X - L/2}{\sqrt{\frac{\lambda z}{2}}}$$

and integral (5) can be put in the form:

$$E(x_1, z) = \frac{\exp(-jkz)}{2} \sqrt{\frac{\lambda z}{2}} \int_{v_-}^{v_+} \exp\left(-j\pi \frac{v^2}{2}\right) dv \quad (6)$$

For convenience acoustic wavelength scaled parameters are adopted:

$$\hat{L} = L/\lambda_a \quad \hat{Z} = z/\lambda_a \quad \hat{X} = x/\lambda_a \quad (7)$$

So that the integration limits of (6) become

$$v_+ = \frac{\hat{X} + \hat{L}/2}{\sqrt{\hat{Z}/2}} \quad v_- = \frac{\hat{X} - \hat{L}/2}{\sqrt{\hat{Z}/2}} \quad (8)$$

and the integral (6) becomes

$$E(x_1, z) = \frac{\exp(-j2\pi\hat{Z})}{\sqrt{2\hat{Z}}} \int_{v_-}^{v_+} \exp(-j\pi \frac{v^2}{2}) dv \quad (9)$$

One further transformation is needed to put (9) into the form required for the numerical evaluation of the Fresnel integral (ref. Appendix II).

$$\text{From (9), let } C = \frac{\exp(-j\pi\hat{Z})}{\sqrt{2\hat{Z}}}$$

and (9) is rewritten in the following form:

$$E(x_1, z) = C \int_0^{v_+} \exp(-j\pi \frac{v^2}{2}) dv - C \int_0^{v_-} \exp(-j\pi \frac{v^2}{2}) dv \quad (10)$$

Now apply the transformation

$$t = \frac{\pi v^2}{2}$$

$$\text{or } v = \sqrt{\frac{2t}{\pi}} = \sqrt{\frac{2}{\pi}} t^{1/2}$$

$$\text{for which } dv = \frac{dt}{\sqrt{2\pi t}}$$

The limits are transformed according to:

$$\text{when } v = v_+ \rightarrow t_+ = \frac{\pi v_+^2}{2}$$

$$\text{and } v = v_- \rightarrow t_- = \frac{\pi v_-^2}{2}$$

hence (10) becomes

$$E(x_1, z) = C \int_0^{t_+} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt - C \int_0^{t_-} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt \quad (11)$$

The above form of Fresnel integral (11) can be evaluated by the series expansion listed in Appendix II.

APPENDIX II

COMPUTATION OF FRESNEL INTEGRALS [23]

Two approximations, one valid for x less than 4 and the other valid for x larger than 4, are listed here for the Fresnel integrals defined in the form:

$$f(x) = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}} dt = C(x) - jS(x)$$

These approximations are the following:

(1) For $0 < x < 4$

$$f(x) = e^{-jx} \sqrt{\frac{x}{4}} \sum_{n=0}^{11} (a_n + jb_n) \left(\frac{x}{4}\right)^n$$

(2) For $x > 4$

$$f(x) = \frac{1-j}{2} + e^{-jx} \sqrt{\frac{4}{x}} \sum_{n=0}^{11} (c_n + jd_n) \left(\frac{4}{x}\right)^n$$

The numerical values of the coefficients a_n , b_n , c_n and d_n are given by:

$a_0 = +1.595769140$	$b_0 = -0.000000033$	$c_0 = 0$	$d_0 = +0.199471140$
$a_1 = -0.000001702$	$b_1 = +4.255387524$	$c_1 = -0.024933975$	$d_1 = +0.000000023$
$a_2 = -6.808568854$	$b_2 = -0.000092810$	$c_2 = +0.000003936$	$d_2 = -0.009351341$
$a_3 = -0.000576361$	$b_3 = -7.780020400$	$c_3 = +0.005770956$	$d_3 = +0.000023006$
$a_4 = +6.920691902$	$b_4 = -0.009520895$	$c_4 = +0.000689892$	$d_4 = +0.004851466$
$a_5 = -0.016898657$	$b_5 = +5.075161298$	$c_5 = -0.009497136$	$d_5 = +0.001903218$
$a_6 = -3.050485660$	$b_6 = -0.138341947$	$c_6 = +0.011948809$	$d_6 = -0.017122914$
$a_7 = -0.075752419$	$b_7 = -1.363729124$	$c_7 = -0.006748873$	$d_7 = +0.029064067$
$a_8 = +0.850663781$	$b_8 = -0.403349276$	$c_8 = +0.000246420$	$d_8 = -0.027928955$
$a_9 = -0.025639041$	$b_9 = +0.702222016$	$c_9 = +0.002102967$	$d_9 = +0.016497308$
$a_{10} = -0.150230960$	$b_{10} = -0.216195929$	$c_{10} = -0.001217930$	$d_{10} = -0.005598515$
$a_{11} = +0.034404779$	$b_{11} = +0.019547031$	$c_{11} = +0.000233939$	$d_{11} = +0.000838386$

The maximum error is 1.6×10^{-9} for the first approximation and 0.5×10^{-9} for the second approximation.

APPENDIX III

FRESNEL INTEGRAL WITH THE ANISOTROPIC APPROXIMATION FOR YZ-LiNbO₃

Consider the Fresnel Integral

$$E(x_1, \rho_0) = \frac{\exp(-jk_e \rho_0)}{\rho_0} \int_{x_1-L/2}^{x_1+L/2} \exp\left(-jk_e \frac{x^2}{2\rho_0}\right) dx \quad (1)$$

where k_e is the energy propagation vector equal to [3]:

$$k_e \cong k_0 \cos\phi (1 - a\theta^2 - b\theta^4 - c\theta^6 - d\theta^8) \quad (2)$$

$$\text{Let } \gamma = - (a\theta^2 + b\theta^4 + c\theta^6 + d\theta^8)$$

$$\text{hence } k_e \cong k_0 \cos\phi (1 + \gamma) \quad (3)$$

where ϕ is the power flow angle and k_0 is the wave number along the pure mode axis. Next, the Fresnel integral (1) is modified to include the energy propagation parameters:

$$E(x_1, \rho_0) = \frac{\exp(-jk_0 \cos\phi (1+\gamma) \rho_0)}{\rho_0} \int_{x_1-L/2}^{x_1+L/2} \exp\left(-jk_0 \cos\phi (1+\gamma) \cdot \frac{x^2}{2\rho_0}\right) dx \quad (4)$$

$$\text{where } k_0 = \frac{2\pi}{\lambda_a}$$

Now, to transform (4) into Fresnel integral classic form:

$$\text{let } \frac{v}{2} = \frac{x^2}{\lambda_a \rho_0} \cos\phi (1+\gamma) \quad \underline{\text{note:}} \quad \rho_0 \equiv z$$

$$\text{or } x_1 = \sqrt{\frac{\lambda_a \rho_0}{2}} \frac{v}{\sqrt{\cos \phi (1+\gamma)}} \quad (5)$$

$$\text{Then, } dx = \sqrt{\frac{\lambda_a \rho_0}{2}} \frac{dv}{\sqrt{\cos \phi (1+\gamma)}} \quad (6)$$

Substituting (5) into (1) yields:

$$E(x_1, \rho_0) = \frac{\exp(-j \frac{2\pi}{\lambda} \rho_0 \cos \phi (1+\gamma))}{\rho_0 \sqrt{\cos \phi (1+\gamma)}} \sqrt{\frac{\lambda_a \rho_0}{2}} \int_{v_-}^{v_+} \exp(-j\pi \frac{v^2}{2}) dv \quad (7)$$

Now, transforming the limits using (5):

When $x = x_1 + L/2$

$$\sqrt{\frac{\lambda_a \rho_0}{2}} \frac{v}{\sqrt{\cos \phi (1+\gamma)}} = x_1 + L/2$$

$$\text{or } v_+ = \frac{x_1 + L/2}{\sqrt{\lambda_a \rho_0 / 2}} \sqrt{\cos \phi (1+\gamma)}$$

Similarly, the lower limit is derived as

$$\text{or } v_- = \frac{x_1 - L/2}{\sqrt{\lambda_a \rho_0 / 2}} \sqrt{\cos \phi (1+\gamma)}$$

where $\rho_0 \equiv z$

For convenience acoustic wavelength scaled parameters are adopted:

$$\hat{L} = L/\lambda_a \quad \hat{Z} = z/\lambda_a \quad \hat{X}_1 = X_1/\lambda_a$$

The integration limits become:

$$v_+ = \frac{\hat{X}_1 - \hat{L}/2}{\sqrt{\hat{Z}/2}} [\cos\theta (1+\gamma)]^{1/2} \quad (8)$$

$$v_- = \frac{\hat{X}_1 - \hat{L}/2}{\sqrt{\hat{Z}/2}} [\cos\theta (1-\gamma)]^{1/2} \quad (9)$$

Using (8) and (9), integral (7) becomes:

$$E(x_1, z) = \frac{\exp(-j2\pi\hat{Z}\cos\theta(1+\gamma))}{\sqrt{2\hat{Z}\cos\theta(1+\gamma)}} \int_{v_-}^{v_+} \exp(-j\pi\frac{v^2}{2}) dv \quad (10)$$

Integral (10) has to be put into the form required for the numerical evaluation, as outlined in Appendix II.

$$\text{From (10), let } C = \frac{\exp(-j2\pi\hat{Z}\cos\theta(1+\gamma))}{\sqrt{2\hat{Z}\cos\theta(1+\gamma)}}$$

and (10) is rewritten in the following form:

$$E(x_1, z) = C \int_{v_-}^{v_+} \exp(-j\pi\frac{v^2}{2}) dv - C \int_0^{v_-} \exp(-j\pi\frac{v^2}{2}) dv \quad (11)$$

Now apply the transformation

$$t = \frac{\pi v^2}{2}$$

$$\text{or } v = \sqrt{\frac{2t}{\pi}}$$

$$\text{for which } dv = \frac{dt}{\sqrt{2\pi t}}$$

The limits are transformed according to:

$$\text{when } v = v_+ \rightarrow t_+ = \frac{\pi}{2} v_+^2$$

$$v = v_- \rightarrow t_- = \frac{\pi}{2} v_-^2$$

hence (11) takes on the following form:

$$E(x_1, z) = c \int_0^{t_+} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt - c \int_0^{t_-} \frac{\exp(-jt)}{\sqrt{2\pi t}} dt \quad (12)$$

Integral (12) is evaluated by the series expansion of Appendix II.

APPENDIX IV

FREQUENCY DIVIDER CIRCUIT

The 70 MHz input signal is first amplified and then de-coupled to the pulse shaping network consisting of two Schottky diodes. The ECL switching threshold V_{BB} is provided by the line driver. The synchronous counter divides the 70 MHz input into a 70 kHz output pulse which is low pass filtered and triggers pulse generator 1 of Figure 6.3.

APPENDIX V

FABRICATION SUMMARY [38]

1) Cleaning of Wafers

Wafers are placed in a plastic holder and lowered into the ultrasonic bath for about 5 minutes. The ultrasonic movement will shake off any dust particles attached to the surface of the wafer. Next, the wafers are lowered into a beaker with propanol to rinse off the ultrasonic bath residue. Then the wafers are microcleaned in a liquid soap and de-ionized water bath for about 10 minutes, followed by rinsing in a cascaded tank with de-ionized water only, for 5 minutes. To clean away any residual soap, the wafers are then rinsed in a beaker with chromic acid, followed by a thorough rinse in de-ionized water in the cascaded tank for 15 minutes.

2) Drying Process

Wafers are dried in the reflux tank for 3 to 4 minutes.

3) Metal Deposition

Metal may be deposited on the wafer surface either by evaporating or by sputtering. Sputtering is used for depositions thicker than 10 microns and the metallized surface is not as smooth as that achieved by using an evaporator.

For the fabrication of 70FIL devices the electron beam evaporator was used and a very smooth and even layer of aluminum was obtained. To measure the thickness of deposition, a quartz crystal, which changes its resonant frequency as the deposition layer increases, is used. The deposition rate was approximately 9Å/sec for aluminum and the total layer deposited was 2000Å.

4) Coating with photoresist and developing

The wafer is placed on a spinning table where it is held down by a vacuum. A layer of adhesive material is applied by spinning the table. This first layer ensures good adhesion of the photoresist itself since the aluminum surface is very smooth and shiny and the photoresist does not adhere uniformly otherwise. Next, photoresist is applied and spun to yield about a 1 micron layer. The thinner the layer, the better for the developing stage.

Positive photoresist is used to coat 70FIL devices. The positive photoresist defines lines as narrow as 1 micron while the negative photoresist defines line width starting at 3 microns.

After the application of the photoresist, the wafer is baked in the oven at 85 - 90°C for approximately 25 minutes. Since LiNbO_3 is very fragile with sudden temperature changes, there must be a gradual increase and decrease in the oven temperature. Baking of the photoresist improves the success of the developing process.

The mask is then mounted on to the mask aligner and the flat edge of the wafer is lined up with the mask boundary. When the mask touches the substrate it is exposed to light. The exposure time is about 3 seconds and the light intensity approximately 7 mW/cm². Commercial developers are used, the developing time is about 45 to 60 seconds.

After developing, the wafers are baked again for approximately 20 minutes, this ensures good adhesion of the photoresist which will then be easier to etch.

5) Etching

The photoresist is etched away in an etching bath which is at a temperature of about 40°C. The etching should be uniform so that some areas do not overetch while others may be underetched.

The etched wafers are rinsed in de-ionized water and then in propanol to remove any remaining photoresist.

Lastly the wafers are rinsed in ultrasonic bath and then inspected under a microscope. There are approximately 12 dice per wafer of 70 MHz filter "70FIL102".

The wafers are diced with a standard IC saw with a special diamond tip for LiNbO_3 . The individual dice are mounted in TO-8 packages where the wire bonding is done with thermal compression.

APPENDIX VI

COMPUTER PROGRAM LISTINGS

FILE FILTR FORTRAN A BELL-NORTHERN RESEARCH

```

C
C THIS PROGRAM PERFORMS NUMERICAL CONVOLUTION BETWEEN THE INPUT
C AND OUTPUT IDT'S TO GENERATE THE FILTER'S IMPULSE RESPONSE.
C THE FAST FOURIER TRANSFORM OF THE IMPULSE RESPONSE IS COMPUTED
C TO YIELD THE FILTER'S FREQUENCY RESPONSE.
C
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION FR(512),K(9),TD(512),WK(1),Z(512),WM(512)
  DIMENSION X(512),T(512),YW(512)
  COMPLEX*16 FFT(257),FTR(512),FT2(512),Y(512)
  COMMON FT2,Y,X,T,YW
  COMMON /DATA/ FO,F1,FSL,FSU,FU,FL,PI,N,NS,NFI,NFO,NDATA

C
C DEFINE CONSTANTS /DATA/ OF COMMON
C
  FO = 700+08
  F1 = 3.50+08
  FSL = 630+08
  FSU = 770+08
  FU = 73.50+08
  FL = 66.50+08
  PI = DABS(DARCOS(-1.000))
  N = 100
  NS = 18
  NFI = 19
  NFO = 125
  NDATA = 512

C
C DEFINE PARAMETERS
C
  ND = NDATA/2+1
  ND2 = NDATA/2
  MFI = (NFI+1)/2
  NF = NFI+1
  NO = NFO+1

C
  OMEGAS = 2.000*FO
  DOMECA = OMEGAS/NDATA
  DTIME = 1/OMEGAS
  ETIME = DTIME/2.000
  TLENG = DFLOAT(NFO)*DTIME

C
C SUBROUTINE 'INPUT' GENERATES THE INPUT NON-APODIZED IDT
C CALL INPUT
C
C SUBROUTINE 'TRUNC' GENERATES THE OUTPUT IDT'S IMPULSE RESPONSE
C CALL TRUNC
C
C DO 5 I=1,NDATA
C WRITE(9,907) FT2(I),Y(I),X(I),T(I),YW(I)
C 5 CONTINUE

```

```

FIL00010
FIL00020
FIL00030
FIL00040
FIL00050
FIL00060
FIL00070
FIL00080
FIL00090
FIL00100
FIL00110
FIL00120
FIL00130
FIL00140
FIL00150
FIL00160
FIL00170
FIL00180
FIL00190
FIL00200
FIL00210
FIL00220
FIL00230
FIL00240
FIL00250
FIL00260
FIL00270
FIL00280
FIL00290
FIL00300
FIL00310
FIL00320
FIL00330
FIL00340
FIL00350
FIL00360
FIL00370
FIL00380
FIL00390
FIL00400
FIL00410
FIL00420
FIL00430
FIL00440
FIL00450
FIL00460
FIL00470
FIL00480
FIL00490
FIL00500
FIL00510
FIL00520
FIL00530
FIL00540
FIL00550

```


FILE TRUNC FORTRAN A BELL-NORTHERN RESEARCH

```

SUBROUTINE TRUNC
C
C   THIS PROGRAM USES THE WINDOW FUNCTION TO TRUNCATE THE IMPULSE
C   RESPONSE OF THE OUTPUT IDT. THE TRUNCATED RESPONSE IS THEN
C   POSITIONED ACCORDING TO DELAY.
C
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION TW(512),W(512),X(512),T(512),YW(512)
  COMPLEX*16 FT2(512),Y(512)
  COMMON FT2,Y,X,T,YW
  COMMON /DATA/ FO,F1,FSL,F5U,FU,FL,PI,N,NS,NFI,NFO,NDATA
C
C   DEFINE PARAMETERS
C
  ND2 = NDATA/2
  MFI = (NFI+1)/2
  NF = NFI+1
C
  OMEGAS = 2.000*FO
  DOMEGA = OMEGAS/NDATA
  DTIME = 1/OMEGAS
  ETIME = DTIME/2.000
  TLENG = DFLOAT(NFO)*DTIME
C
  KDATA = NDATA/2+1
  LDATA = NDATA/2-1
  MDATA = NDATA+1
C
  KFO = (NFO-1)/2
  MFO = (NFO+1)/2
C
  NFO1 = KDATA-MFO
  NFO2 = NFO1+1
  NFO3 = KDATA+KFO
  NFO4 = NFO3+1
C
  TDO = DFLOAT(N)*DTIME
  TDE = DFLOAT(N)*DTIME+ETIME
C
  NFDO = (NFI+1)/2+N
  NFDE = NFI/2+1+N
C
  NFOE1 = KDATA-NFO/2-1
C
  NNE1 = NFDE-NFO/2-1
  NNE2 = NFDE-NFO/2
C
  NN1 = NFDO-MFO
  NN2 = NN1+NFO+1
C
  NO1 = NFDE-MFO
  NO2 = KDATA-MFO
  NO3 = KDATA+MFO
C
  TRU00010
  TRU00020
  TRU00030
  TRU00040
  TRU00050
  TRU00060
  TRU00070
  TRU00080
  TRU00090
  TRU00100
  TRU00110
  TRU00120
  TRU00130
  TRU00140
  TRU00150
  TRU00160
  TRU00170
  TRU00180
  TRU00190
  TRU00200
  TRU00210
  TRU00220
  TRU00230
  TRU00240
  TRU00250
  TRU00260
  TRU00270
  TRU00280
  TRU00290
  TRU00300
  TRU00310
  TRU00320
  TRU00330
  TRU00340
  TRU00350
  TRU00360
  TRU00370
  TRU00380
  TRU00390
  TRU00400
  TRU00410
  TRU00420
  TRU00430
  TRU00440
  TRU00450
  TRU00460
  TRU00470
  TRU00480
  TRU00490
  TRU00500
  TRU00510
  TRU00520
  TRU00530
  TRU00540
  TRU00550

```

FILE TRUNC FORTRAN A BELL-NORTHERN RESEARCH

```

      NE1 = NFD0-NF0/2
      NE2 = ND2-NF0/2
      NE3 = ND2+NFO/2+1
C
C      SUBROUTINE INPUT GENERATES THE INPUT AND THE TIME VECTORS
C
C      CALL INPUT
C
C      SUBROUTINE 'APOD' GENERATES THE OUTPUT IDT IMPULSE RESPONSE
C
C      CALL APOD
C
C      PRINT HEADINGS
C      WRITE(9,906)
C
C      CALCULATE TIME OF WINDOW
C
      DO 100 I=1,NDATA
C
C      CHECK FOR EVEN NFO
C
      IF((NFO/2)*2.EQ.NFO) GO TO 110
C
C      CALCULATION FOR ODD NFO
C
      J=I-KDATA
      CMULT=DFLOAT(J)
      TIME=CMULT*OTIME
      TW(I)=TIME
C
      GO TO 100
C
C      CALCULATION FOR EVEN NFO
C
      110 J=I-ND2
      CMULT=DFLOAT(J)
      TIME=CMULT*DTIME-ETIME
      TW(I)=TIME
C
      100 CONTINUE
C
C      COMPUTE THE AMPLITUDE OF THE HAMMING WINDOW FUNCTION
C
      DO 200 I=1,NFO1
      W(I)=0.000
      200 CONTINUE
C
      DO 210 I=NFO2,NFO3
      PAR=TW(I)
      W(I)=5.40-01+4.60-01*DCOS(2.000*PI*(PAR/TLENG))
      210 CONTINUE
C
      DO 220 I=NFO4,NDATA
      W(I)=0.000
      220 CONTINUE

```

TRU00560
 TRU00570
 TRU00580
 TRU00590
 TRU00600
 TRU00610
 TRU00620
 TRU00630
 TRU00640
 TRU00650
 TRU00660
 TRU00670
 TRU00680
 TRU00690
 TRU00700
 TRU00710
 TRU00720
 TRU00730
 TRU00740
 TRU00750
 TRU00760
 TRU00770
 TRU00780
 TRU00790
 TRU00800
 TRU00810
 TRU00820
 TRU00830
 TRU00840
 TRU00850
 TRU00860
 TRU00870
 TRU00880
 TRU00890
 TRU00900
 TRU00910
 TRU00920
 TRU00930
 TRU00940
 TRU00950
 TRU00960
 TRU00970
 TRU00980
 TRU00990
 TRU01000
 TRU01010
 TRU01020
 TRU01030
 TRU01040
 TRU01050
 TRU01060
 TRU01070
 TRU01080
 TRU01090
 TRU01100

FILE TRUNC FORTRAN A BELL-NORTHERN RESEARCH

C		TRUO1110
C		TRUO1120
C	TRUNCATE IMPULSE RESPONSE WITH HAMMING WINDOW	TRUO1130
C		TRUO1140
C	DO 400 I=1,NDATA	TRUO1150
C		TRUO1160
C	CHECK FOR EVEN NFO	TRUO1170
C		TRUO1180
C	IF((NFO/2)=2.EQ.NFO) GO TO 410	TRUO1190
C		TRUO1200
C	ODD NFO	TRUO1210
C		TRUO1220
C	YW(I)=DREAL(Y(I))*W(I)	TRUO1230
C		TRUO1240
C	GO TO 400	TRUO1250
C		TRUO1260
C	EVEN NFO	TRUO1270
C		TRUO1280
C	410 IF(I EQ.NDATA) GO TO 420	TRUO1290
C		TRUO1300
C	YW(I)=((DREAL(Y(I))+DREAL(Y(I+1)))/2)*W(I)	TRUO1310
C		TRUO1320
C	GO TO 400	TRUO1330
C		TRUO1340
C	420 YW(I)=((DREAL(Y(I))+DREAL(Y(1)))/2)*W(I)	TRUO1350
C		TRUO1360
C	400 CONTINUE	TRUO1370
C		TRUO1380
C	CHECK FOR ODD OR EVEN INPUT OR OUTPUT IDT	TRUO1390
C		TRUO1400
C	IF((NFI/2)=2.EQ.NFI.AND.(NFO/2)=2.EQ.NFO) GO TO 590	TRUO1410
C		TRUO1420
C	IF((NFI/2)=2.EQ.NFI) GO TO 690	TRUO1430
C		TRUO1440
C	IF((NFO/2)=2.EQ.NFO) GO TO 790	TRUO1450
C		TRUO1460
C	POSITION ODD - ODD IDT'S	TRUO1470
C		TRUO1480
C	DO 500 I=1,NFO	TRUO1490
C	J=I+NN1	TRUO1500
C	K=I+NFO1	TRUO1510
C	YW(J)=YW(K)	TRUO1520
C	500 CONTINUE	TRUO1530
C		TRUO1540
C	DO 510 I=1,NN1	TRUO1550
C	YW(I)=0.000	TRUO1560
C	510 CONTINUE	TRUO1570
C		TRUO1580
C	DO 520 I=NN2,NDATA	TRUO1590
C	YW(I)=0.000	TRUO1600
C	520 CONTINUE	TRUO1610
C		TRUO1620
C	GO TO 900	TRUO1630
C		TRUO1640
C	POSITION EVEN - EVEN IDT'S	TRUO1650

FILE TRUNC FORTRAN A BELL-NORTHERN RESEARCH

C		TRU01660
590	DO 600 I=1,NFO	TRU01670
	J=I+NNE1	TRU01680
	K=J+NFOE1	TRU01690
	YW(J)=YW(K)	TRU01700
600	CONTINUE	TRU01710
C		TRU01720
	DO 610 I=1,NNE1	TRU01730
	YW(I)=0.000	TRU01740
610	CONTINUE	TRU01750
C		TRU01760
	DO 620 I=NNE2,NFO	TRU01770
	YW(I)=0.000	TRU01780
620	CONTINUE	TRU01790
C		TRU01800
C	POSITION EVEN - ODD IDT'S	TRU01810
C		TRU01820
690	DO 700 I=1,NFO	TRU01830
	J=I+NO1	TRU01840
	K=I+NO2	TRU01850
	YW(J)=YW(K)	TRU01860
700	CONTINUE	TRU01870
C		TRU01880
	DO 710 I=1,NO1	TRU01890
	YW(I)=0.000	TRU01900
710	CONTINUE	TRU01910
C		TRU01920
	DO 720 I=1,NO3	TRU01930
	YW(I)=0.000	TRU01940
720	CONTINUE	TRU01950
C		TRU01960
	GO TO 900	TRU01970
C		TRU01980
C	POSITION ODD - EVEN IDT'S	TRU01990
C		TRU02000
790	DO 800 I=1,NFO	TRU02010
	J=I+NE1	TRU02020
	K=I+NE2	TRU02030
	YW(J)=YW(K)	TRU02040
800	CONTINUE	TRU02050
C		TRU02060
	DO 810 I=1,NE2	TRU02070
	YW(I)=0.000	TRU02080
810	CONTINUE	TRU02090
C		TRU02100
	DO 820 I=NE3,NFO	TRU02110
	YW(I)=0.000	TRU02120
820	CONTINUE	TRU02130
C		TRU02140
C	CHECK FOR OVERLAP OF INPUT AND OUTPUT IDT'S	TRU02150
C		TRU02160
900	DO 910 I=1,NFO	TRU02170
	PAR=X(I)+YW(I)	TRU02180
	IF(PAR.LT.-1.00-12.AND.PAR.GT.1.00-12) GO TO 920	TRU02190
	GO TO 910	TRU02200

FILE TRUNC FORTRAN A BELL-NORTHERN RESEARCH

C		TRU02210
920	WRITE(6,907)	TRU02220
	STOP	TRU02230
C		TRU02240
910	CONTINUE	TRU02250
C		TRU02260
	AMAX=DABS(YW(1))	TRU02270
	DO 950 I=1,NDATA	TRU02280
	IF(DABS(YW(I)).GT.AMAX) AMAX=DABS(YW(I))	TRU02290
950	CONTINUE	TRU02300
C		TRU02310
C	PRINT RESULTS	TRU02320
C		TRU02330
	DO 1000 I=1,NDATA	TRU02340
	YW(I)=YW(I)/AMAX	TRU02350
C	WRITE(9,905) I,T(I),X(I),YW(I)	TRU02360
	WRITE(11,1009) YW(I)	TRU02370
1000	CONTINUE	TRU02380
C		TRU02390
C		TRU02400
905	FORMAT(I10,1P3D20.6)	TRU02410
906	FORMAT(9X,'I',8X,'TIME',16X,'INPUT IDT',11X,'OUTPUT IDT')	TRU02420
907	FORMAT(10X,'DELAY ERROR-INCREASE THE DELAY')	TRU02430
1009	FORMAT(1PD20.10)	TRU02440
C		TRU02450
	RETURN	TRU02460
C	STOP	TRU02470
	END	TRU02480

FILE, APOD FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE APOD	AP000010
C		AP000020
C	THIS PROGRAM EVALUATES THE OUTPUT IDT APODIZATION	AP000030
C	BY PERFORMING THE INVERSE FOURIER TRANSFORM OF THE	AP000040
C	FREQUENCY RESPONSE.	AP000050
C		AP000060
	IMPLICIT REAL*8(A-H,O-Z)	AP000070
	DIMENSION IWK(10),X(512),T(512),YW(512)	AP000080
	COMPLEX*16 FT2(512),Y(512)	AP000090
	COMMON FT2,Y,X,T,YW	AP000100
	COMMON /DATA/ FO,F1,FSL,FSU,FU,FL,PI,N,NS,NF1,NFO,NDATA	AP000110
C		AP000120
C	DEFINE PARAMETERS	AP000130
C		AP000140
	M=9	AP000150
	KDATA=NDATA/2+1	AP000160
	LDATA=NDATA/2-1	AP000170
	MDATA=NDATA+1	AP000180
C		AP000190
C	SUBROUTINE 'INPUT' GENERATES THE INPUT NON-APODIZED IDT	AP000200
C	AND THE TIME VECTOR	AP000210
C		AP000220
C	CALL INPUT	AP000230
C		AP000240
C	SUBROUTINE 'FREQ' GENERATES THE FREQUENCY	AP000250
C	RESPONSE OF THE OUTPUT IDT	AP000260
C		AP000270
C	CALL FREQ	AP000280
C		AP000290
C	PRINT HEADINGS I,TIME,Y	AP000300
C	WRITE(9,906)	AP000310
C		AP000320
C		AP000330
C	IMSL SUBROUTINE FFT2C COMPUTES	AP000340
C	THE INVERSE FOURIER TRANSFORM	AP000350
C		AP000360
C	CALL FFT2C(FT2,M,IWK)	AP000370
C		AP000380
C	REARRANGE FFT2C OUTPUT VECTOR FOR SYMMETRY	AP000390
C		AP000400
	DO 10 I=1,KDATA	AP000410
	J=I-1	AP000420
	K=KDATA-J	AP000430
	Y(K)=FT2(I)	AP000440
10	CONTINUE	AP000450
C		AP000460
	DO 20 I=1,LDATA	AP000470
	J=MDATA-I	AP000480
	K=KDATA+I	AP000490
	Y(K)=FT2(J)	AP000500
20	CONTINUE	AP000510
C		AP000520
C	PRINT RESULTS	AP000530
C		AP000540
C	DO 50 I=1,NDATA	AP000550

FILE APOD FORTRAN A BELL-NORTHERN RESEARCH

C	WRITE(9,905) I,T(I),X(I),Y(I)	AP000560
C50	CONTINUE	AP000570
C		AP000580
C		AP000590
905	FORMAT(I3,3X,1PD10.3,1P4D15.4)	AP000600
906	FORMAT(2X,'I',4X,'TIME',11X,'X(I)',11X,'Y(I)-REAL',5X,'Y(I)-IMAG')	AP000610
C		AP000620
C		AP000630
	RETURN	AP000640
C	STOP	AP000650
	END	AP000660

FILE FREQ FORTRAN A BELL-NORTHERN RESEARCH

```

SUBROUTINE FREQ
C
C   THIS PROGRAM COMPUTES THE FAST FOURIER TRANSFORM (FFT)
C   OF THE INPUT UNIFORM TRANSDUCER.
C   THE DESIRED FREQUENCY RESPONSE OF THE OUTPUT IDT IS
C   GENERATED.
C
C   IMPLICIT REAL*8(A-H,O-Z)
C   DIMENSION K(9),WK(1),FR(512),X(512),T(512),YW(512)
C   COMPLEX*16 FT1(512),FFT(512),FT2(512),Y(512)
C   COMMON FT2,Y,X,T,YW
C   COMMON /DATA/ FO,F1,FSL,FSU,FU,FL,PI,N,NS,NFI,NFO,NDATA
C
C   DEFINE PARAMETERS
C
C   ND2=NDATA/2
C   OMEGAS=2.000*FO
C   DOMEGA=OMEGAS/NDATA
C
C   NS1 = ND2-NS
C   NS2 = NS1+1
C   NS3 = NS2+2*NS
C   NS4 = NS3+1
C
C   SUBROUTINE 'INPUT' GENERATES THE INPUT NON-APODIZED IDT
C   CALL INPUT
C
C   IMSL SUBROUTINE FFTRC COMPUTES THE FAST FOURIER TRANSFORM
C   OF THE INPUT NON-APODIZED TRANSDUCER.
C   CALL FFTRC(X,NDATA,FT1,K,WK)
C
C   PRINT HEADINGS
C   WRITE(9,906)
C
C   CALCULATE THE REMAINING VALUES OF FT1
C
C   DO 30 I=2,ND2
C   J=NDATA+2-I
C   FT1(J)=DCONJG(FT1(I))
30  CONTINUE
C
C   CALCULATE THE MAX VALUE OF FT1 TO NORMALIZE FT1
C
C   AMAX=CDABS(FT1(1))
C   DO 40 I=1,NDATA
C   IF(CDABS(FT1(I)).GT.AMAX) AMAX=CDABS(FT1(I))
40  CONTINUE
C
C   DO 60 I=1,NDATA
C
C   NORMALIZE FT1
C   FT1(I)=FT1(I)/AMAX
C

```

FRE00010
 FRE00020
 FRE00030
 FRE00040
 FRE00050
 FRE00060
 FRE00070
 FRE00080
 FRE00090
 FRE00100
 FRE00110
 FRE00120
 FRE00130
 FRE00140
 FRE00150
 FRE00160
 FRE00170
 FRE00180
 FRE00190
 FRE00200
 FRE00210
 FRE00220
 FRE00230
 FRE00240
 FRE00250
 FRE00260
 FRE00270
 FRE00280
 FRE00290
 FRE00300
 FRE00310
 FRE00320
 FRE00330
 FRE00340
 FRE00350
 FRE00360
 FRE00370
 FRE00380
 FRE00390
 FRE00400
 FRE00410
 FRE00420
 FRE00430
 FRE00440
 FRE00450
 FRE00460
 FRE00470
 FRE00480
 FRE00490
 FRE00500
 FRE00510
 FRE00520
 FRE00530
 FRE00540
 FRE00550

FILE FREQ FORTRAN A BELL-NORTHERN RESEARCH

60	CONTINUE	FREQ0560
C		FREQ0570
C	COMPUTE MAG OF FT1 AND SET PHASE = 0.0	FREQ0580
C		FREQ0590
	DO 61 I=1,NDATA	FREQ0600
	FMAG=CDABS(FT1(I))	FREQ0610
	FT1(I)=DCMPLX(FMAG,0.000)	FREQ0620
61	CONTINUE	FREQ0630
C		FREQ0640
C	CALCULATE THE DESIRED OUTPUT IDT RESPONSE	FREQ0650
C		FREQ0660
	DO 65 I=1,NS1	FREQ0670
	FT2(I)=0.000	FREQ0680
65	CONTINUE	FREQ0690
C		FREQ0700
	DO 66 I=NS2,NS3	FREQ0710
	FT2(I)=1.000/FT1(I)	FREQ0720
C	FT2(I)=1.000	FREQ0730
66	CONTINUE	FREQ0740
C		FREQ0750
	DO 67 I=NS4,NDATA	FREQ0760
	FT2(I)=0.000	FREQ0770
67	CONTINUE	FREQ0780
C		FREQ0790
C	COMPUTE THE MAX VALUE OF FT2 TO NORMALIZE FT2	FREQ0800
C		FREQ0810
	BMAX=CDABS(FT2(1))	FREQ0820
	DO 68 I=1,NDATA	FREQ0830
	IF(CDABS(FT2(I)).GT.BMAX) BMAX=CDABS(FT2(I))	FREQ0840
68	CONTINUE	FREQ0850
C		FREQ0860
C	NORMALIZE FT2	FREQ0870
C		FREQ0880
	DO 69 I=1,NDATA	FREQ0890
	FT2(I)=FT2(I)/BMAX	FREQ0900
69	CONTINUE	FREQ0910
C		FREQ0920
	DO 80 I=1,NDATA	FREQ0930
C		FREQ0940
C	COMPUTE THE MAGNITUDE OF FT1 IN DB'S	FREQ0950
	ABSVAL=CDABS(FT1(I))	FREQ0960
	IF(ABSVAL.LE.0.0) ABSVAL=1D-12	FREQ0970
	AMAG=20.000+DLOG10(ABSVAL)	FREQ0980
C		FREQ0990
C	COMPUTE THE PHASE OF FT1 IN DEGREES	FREQ1000
	APHASE=0.000	FREQ1010
C		FREQ1020
C		FREQ1030
C	COMPUTE THE MAGNITUDE OF FT2 IN DB'S	FREQ1040
	BBSVAL=CDABS(FT2(I))	FREQ1050
	IF(BBSVAL.LE.0.0) BBSVAL=1D-12	FREQ1060
	BMAG=20.000+DLOG10(BBSVAL)	FREQ1070
C		FREQ1080
C	COMPUTE THE PHASE OF FT2 IN DEGREES	FREQ1090
	BPHASE=0.000	FREQ1100

FILE: FREQ FORTRAN A BELL-NORTHERN RESEARCH

C		FREQ1110
C	COMPUTE FREQUENCIES CORRESPONDING TO SAMPLE NUMBERS	FREQ1120
C		FREQ1130
	J=I-1	FREQ1140
	CMULT=DFLOAT(J)	FREQ1150
	FREQ=CMULT*QMEGA	FREQ1160
	FR(I)=FREQ	FREQ1170
C		FREQ1180
C	WRITE(9,905) I,FR(I),AMAG,APHASE,BMAG,BPHASE	FREQ1190
C		FREQ1200
80	CONTINUE	FREQ1210
C		FREQ1220
C		FREQ1230
905	FORMAT(I3,X,1PD9.3,X,1P4D11.3)	FREQ1240
906	FORMAT(2X,'I',X,'FREQ(HZ)',4X,'AMAG(DB)',3X,'APHAS(DEG)',	FREQ1250
	XX,'BMAG(DB)',3X,'BPHAS(DEG)')	FREQ1260
907	FORMAT(I3,X,1P4E11.3)	FREQ1270
C		FREQ1280
C		FREQ1290
C	STOP	FREQ1300
	RETURN	FREQ1310
	END	FREQ1320

FILE: INPUT FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE INPUT	INP00010
C		INP00020
C	THIS PROGRAM GENERATES COMPLETE INPUT IDT DATA FILE	INP00030
C	AND THE CORRESPONDING TIME VECTOR.	INP00040
C		INP00050
	IMPLICIT REAL*8(A-H,O-Z)	INP00060
	DIMENSION X(512),T(512),Y(512)	INP00070
	COMPLEX*16 FT2(512),Y(512)	INP00080
	COMMON FT2,Y,X,T,YW	INP00090
	COMMON /DATA/ FO,F1,FSL,FSU,FU,FL,PI,N,NS,NFI,NFO,NDATA	INP00100
C		INP00110
C	ABBREVIATIONS	INP00120
C		INP00130
C	FO = CENTRE FREQUENCY	INP00140
C	F1 = BANDWIDTH TO ROLLOFF (2*F1)	INP00150
C	NFI = NUMBER OF FINGERS OF INPUT TRANSDUCER	INP00160
C	NFO = NUMBER OF FINGERS OF OUTPUT TRANSDUCER	INP00170
C	DTIME= STEP IN TIME FROM FINGER TO FINGER	INP00180
C	TLENG= TOTAL LENGTH OF TRANSDUCER IN TIME	INP00190
C	DELAY= SEPARATION OF INPUT AND OUTPUT IDT,CENTER TO CENTER	INP00200
C		INP00210
C	DEFINE PARAMETERS	INP00220
C		INP00230
	KFI=NFI+1	INP00240
	NFI=(NFI+1)/2	INP00250
C		INP00260
	DTIME=1/(2.000*FO)	INP00270
	ETIME=DTIME/2.000	INP00280
C		INP00290
C		INP00300
C	PRINT HEADINGS I,TIME,XIN	INP00310
C	WRITE(9,906)	INP00320
C		INP00330
C		INP00340
C	GENERATE INPUT IDT DATA	INP00350
C		INP00360
	DO 101 I=1,NFI	INP00370
	J=(-1)**(I+1)	INP00380
	CMULT=DFLOAT(J)	INP00390
	X(I)=CMULT	INP00400
101	CONTINUE	INP00410
C		INP00420
	DO 102 I=KFI,NDATA	INP00430
	X(I)=0.000	INP00440
102	CONTINUE	INP00450
C		INP00460
C	GENERATE TIME FILE	INP00470
C		INP00480
	DO 103 I=1,NDATA	INP00490
C		INP00500
C	TEST FOR EVEN NFI	INP00510
	IF((NFI/2)*2.EQ.NFI) GO TO 10	INP00520
C		INP00530
C	CALCULATION FOR ODD NFI	INP00540
C		INP00550

FILE INPUT FORTRAN A BELL-NORTHERN RESEARCH

J=I-NFI	INP00560
CMULT=DFLOAT(J)	INP00570
TIME=CMULT*ETIME	INP00580
T(I)=TIME	INP00590
GO TO 103	INP00600
C	INP00610
C CALCULATION FOR EVEN NFI	INP00620
C	INP00630
10 J=I-NFI/2	INP00640
CMULT=DFLOAT(J)	INP00650
TIME=CMULT*ETIME-ETIME	INP00660
T(I)=TIME	INP00670
C	INP00680
103 CONTINUE	INP00690
C	INP00700
C	INP00710
C PRINT RESULTS	INP00720
C	INP00730
DO 104 I=1,NDATA	INP00740
WRITE(9,905) I,T(I),X(I)	INP00750
104 CONTINUE	INP00760
C	INP00770
C	INP00780
905 FORMAT(I10,1P3D20.6)	INP00790
906 FORMAT(9X,'I',10X,'TIME(I)',13X,'XIN(I)')	INP00800
C	INP00810
RETURN	INP00820
C STOP	INP00830
END	INP00840

FILE ANISOT FORTRAN A BELL-NORTHERN RESEARCH

```

C
C THIS ROUTINE COMPUTES A NEW DIFFRACTION COMPENSATED DESIGN OF
C SURFACE ACOUSTIC WAVE FILTER
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION YW(125),NCOUNT(125),SUBL(125),ALENG(125),XOLD(125,410)
C DIMENSION XLIN(512),XPHS(512),TWLIN(512),TWPHS(512),XNEW(500)
C DIMENSION XX(500),XPROF(500),TXMAG(512),TXPHS(512),YPHS(500)
C DIMENSION TXNLIN(512),TXNMAG(512),TXNPHS(512),YLIN(500),YMAG(500)
C DIMENSION IWK(10),FR(512),QLIN(512),OMAG(512),OPHS(512),XSUB(19)
C DIMENSION FFTLIN(512),FFTMAG(512),FFTPHS(512),XLENG(19),MXCNT(19)
C DIMENSION YYLIN(500),YYMAG(500),YYPHS(500),YNEWL(500),YNEWN(500)
C DIMENSION YNEWP(500),XDLIN(512),XDMAG(512),XDPHS(512)
C COMPLEX*16 X(512),Y(512),SINE(125),YD(125),YOLD(125,410),TX(512)
C COMPLEX*16 YNEW(500),YY(500),YPROF(500),TOTL(125),SUM,TXN(512)
C COMPLEX*16 Q(512),FFT(512),XSINE(19),XDIF(512)
C COMMON /DATA/ PI,FO,FLAMB0,DELAY,AP1,AP2,NFO,NFI,NDATA
C
C DEFINE CONSTANTS /DATA/ OF COMMON
C PI = DABS(DARCOS(-1.000))
C FO = 70.00+06
C FLAMB0 = 49.20+06
C DELAY = 5415.520+06/FLAMB0
C AP1 = 2461.450+06/FLAMB0
C AP2 = 45.000
C NFO = 125
C NFI = 19
C NDATA = 512
C
C CONSTANT OF INSL ROUTINE 'FFT2C'
C MM = 9
C
C DEFINE PARAMETERS
C KFO = -(NFO-1)/2
C KFI = (NFI-1)/2
C DTR = PI/180.000
C RTD = 180.000/PI
C THTA = 45.000+DTR
C ITMAX = 8
C ITMIN = 7
C IT = 1
C NN1 = 57
C RLXM = 0.7500
C RLXP = 0.000
C
C IS YOUR SUBSTRATE ISOTROPIC??? (1 = Y.O = N)
C 1 WRITE(8,1000)
C READ(6,*) ISOT
C IF(ISOT.NE.1.AND.ISOT.NE.0) GO TO 1
C IF(ISOT.EQ.0) GO TO 2
C
C WRITE(10,1006)
C WRITE(9,1006)
C WRITE(7,1006)

```

FILE ANISOT FORTRAN A BELL-NORTHERN RESEARCH

```

      AR = 0.000
      BR = 0.000
      CR = 0.000
      DR = 0.000
      GO TO 3
C
C      DO POLYNOMIAL CURVEFIT OF VELOCITY VS. ANGLE OF PROPAGATION
C      ON LINGO3 SUBSTRATES (BASED ON SLOBODNIK'S DATA)
C      A,B,C,D ARE COEFFICIENTS OF 8TH ORDER POLYNOMIAL
C
2    WRITE(10,1006)
      WRITE(9,1006)
      WRITE(7,1006)
      A = -1.26000-04
      B = 3.68920-07
      C = -5.08870-10
      D = 2.75510-13
C
      AR = A*(RTD**2)
      BR = B*(RTD**4)
      CR = C*(RTD**6)
      DR = D*(RTD**8)
C
      WRITE(6,1001) A
      WRITE(10,1001) A
      WRITE(9,1001) A
C
      WRITE(6,1002) B
      WRITE(10,1002) B
      WRITE(9,1002) B
C
      WRITE(6,1003) C
      WRITE(10,1003) C
      WRITE(9,1003) C
C
      WRITE(6,1004) D
      WRITE(10,1004) D
      WRITE(9,1004) D
C
C      DO YOU WISH TO HAVE AN INCLINE IN THE APODIZED IDT?(1=Y,0=N)
C
3    WRITE(6,1007)
      READ(6,*) INCLN
      IF(INCLN.NE.1.AND.INCLN.NE.0) GO TO 3
      IF(INCLN.EQ.1) GO TO 4
      IF(INCLN.EQ.0) BETA = 0.000
      GO TO 5
C
C      HOW MANY DEGREES???
C
4    WRITE(6,1008)
      READ(6,*) BETA
      WRITE(6,1009) BETA
      WRITE(10,1009) BETA
      WRITE(9,1009) BETA
      BETA = DTR*BETA
C

```

ANI00560
 ANI00570
 ANI00580
 ANI00590
 ANI00600
 ANI00610
 ANI00620
 ANI00630
 ANI00640
 ANI00650
 ANI00660
 ANI00670
 ANI00680
 ANI00690
 ANI00700
 ANI00710
 ANI00720
 ANI00730
 ANI00740
 ANI00750
 ANI00760
 ANI00770
 ANI00780
 ANI00790
 ANI00800
 ANI00810
 ANI00820
 ANI00830
 ANI00840
 ANI00850
 ANI00860
 ANI00870
 ANI00880
 ANI00890
 ANI00900
 ANI00910
 ANI00920
 ANI00930
 ANI00940
 ANI00950
 ANI00960
 ANI00970
 ANI00980
 ANI00990
 ANI01000
 ANI01010
 ANI01020
 ANI01030
 ANI01040
 ANI01050
 ANI01060
 ANI01070
 ANI01080
 ANI01090
 ANI01100

FILE: ANISOT FORTRAN A BELL-NORTHERN RESEARCH

C	GENERATE THE INPUT UNAPODIZED VECTOR 'X'	ANIO1110
5	CALL INPIDT(X,XLIN,XPHS)	ANIO1120
C		ANIO1130
C	APODIZATION SIGN VECTOR OF THE INPUT UNAPODIZED IDT	ANIO1140
	DO 30 I = 1,NFI	ANIO1150
	XLENG(I) = AP2*XLIN(I)	ANIO1160
	CMULT = DSIGN(1.000,DREAL(X(I)))	ANIO1170
	XSINE(I) = DCMPLX(CMULT,0.000)	ANIO1180
30	CONTINUE	ANIO1190
C		ANIO1200
C	SUBLENGTH OF THE TRANSMITTING FINGERS OF THE INPUT UNAPODIZED IDT	ANIO1210
	N = 25	ANIO1220
	DO 31 I = 1,NFI	ANIO1230
C		ANIO1240
	DO 32 J = 1,N	ANIO1250
	XSUB(I) = DABS(XLENG(I))/DFLOAT(J)	ANIO1260
	IF(XSUB(I).LE.2.000) GO TO 33	ANIO1270
32	CONTINUE	ANIO1280
C		ANIO1290
33	NXCNT(I) = J	ANIO1300
C	WRITE(10,2004) I,NXCNT(I),XLENG(I),XSUB(I)	ANIO1310
C		ANIO1320
31	CONTINUE	ANIO1330
C		ANIO1340
C	COMPUTE DIFFRACTED WEIGHTS OF THE INPUT IDT	ANIO1350
	GAMA = 0.000	ANIO1360
	NDAT = NFI	ANIO1370
	CALL DIF(NDAT,KFI,AR,BR,CR,DR,GAMA,XSUB,XOLD,YOLD,NBR)	ANIO1380
C		ANIO1390
C	DO 900 I = 1,NFI	ANIO1400
C	DO 901 J = 1,NBR	ANIO1410
C		ANIO1420
C	WRITE(10,2004) I,J,XOLD(I,J),YOLD(I,J)	ANIO1430
C		ANIO1440
C901	CONTINUE	ANIO1450
C900	CONTINUE	ANIO1460
C		ANIO1470
C	GO THRU SORTING,SLIDING AND INTEGRATION	ANIO1480
	DO 51 I=1,NFI	ANIO1490
	SUB = XSUB(I)	ANIO1500
	NVAL = NXCNT(I)	ANIO1510
C		ANIO1520
	DO 52 J = 1,NBR	ANIO1530
	XX(J) = XOLD(I,J)	ANIO1540
	YY(J) = YOLD(I,J)	ANIO1550
52	CONTINUE	ANIO1560
C		ANIO1570
	CALL SORT(NBR,SUB,XX,YY,XNEW,YNEW,INT)	ANIO1580
C		ANIO1590
	CALL SLIDE(NVAL,INT,XNEW,YNEW,XPROF,YPROF,NTOT)	ANIO1600
C		ANIO1610
	CALL INTEG(NTOT,XPROF,YPROF,SUM)	ANIO1620
C		ANIO1630
	XDIF(I) = SUM*XSINE(I)	ANIO1640
C		ANIO1650

FILE ANISOT FORTRAN A BELL-NORTHERN RESEARCH

51	CONTINUE	ANIO1660
C		ANIO1670
C	REMAINING VALUES OF X-VECTOR(512) ARE ZERO	ANIO1680
	NFF = NFI+1	ANIO1690
	DO 53 I = NFF,NDATA	ANIO1700
	XDIF(I) = DCMLPX(0.000,0.000)	ANIO1710
53	CONTINUE	ANIO1720
C		ANIO1730
	DO 54 I = 1,NFI	ANIO1740
	XDIF(I) = XDIF(I)/XDIF(NFI)	ANIO1750
54	CONTINUE	ANIO1760
C		ANIO1770
	CALL MAGPHS(XDIF,XDLIN,XDMAG,XDPHS)	ANIO1780
C		ANIO1790
C	DO 55 I = 1,NDATA	ANIO1800
C	WRITE(10,2002) I,XDLIN(I),XDPHS(I)	ANIO1810
C55	CONTINUE	ANIO1820
C		ANIO1830
C		ANIO1840
C		ANIO1850
C	READ ORIGINAL APODIZATION WEIGHTING OF OUTPUT IDT	ANIO1860
	DO 50 I=1,NFO	ANIO1870
	READ(1,*) YW(I)	ANIO1880
	ALENG(I) = AP2*YW(I)	ANIO1890
50	CONTINUE	ANIO1900
C		ANIO1910
C	OUTPUT IDT 125 SAMPLE COMPLEX VECTOR YO(125)	ANIO1920
	CALL OUTIDT(YW,YO)	ANIO1930
C		ANIO1940
C	CREATE APODIZATION SIGN VECTOR OF THE OUTPUT IDT	ANIO1950
	DO 56 I=1,NFO	ANIO1960
	CMULT = DSIGN(1.000,DREAL(YO(I)))	ANIO1970
	SINE(I) = DCMLPX(CMULT,0.000)	ANIO1980
56	CONTINUE	ANIO1990
C		ANIO2000
C	POSITION THE OUTPUT IDT AT THE CORRECT DELAY AND GET TW(512)	ANIO2010
	CALL POSITY(YO,TW,TWLIN,TWPHS)	ANIO2020
C		ANIO2030
	N = 25	ANIO2040
C	COMPUTE SUBLLENGTH OF THE TRANSMITTING FINGER	ANIO2050
	DO 60 I = 1,NFO	ANIO2060
C		ANIO2070
	DO 65 J = 1,N	ANIO2080
C		ANIO2090
	SUBL(I) = DABS(ALENG(I))/DFLOAT(J)	ANIO2100
C		ANIO2110
	IF(SUBL(I).LE.2.000) GO TO 6	ANIO2120
C		ANIO2130
65	CONTINUE	ANIO2140
C		ANIO2150
6	NCOUNT(I) = J	ANIO2160
C	WRITE(10,2004) I,NCOUNT(I),ALENG(I),SUBL(I)	ANIO2170
C		ANIO2180
60	CONTINUE	ANIO2190
C		ANIO2200

FILE ANISOT FORTRAN A BELL-NORTHERN RESEARCH

	NDAT = NFO	ANIO2210
	CALL DIF(NDAT,KFO,AR,BR,CR,DR,BETA,SUBL,XOLD,YOLD,NUMBER)	ANIO2220
C		ANIO2230
C	GO THRU SORTING,SLIDING & INTEGRATION	ANIO2240
C		ANIO2250
	DO 100 I = 1,NFO	ANIO2260
	SUB = SUBL(I)	ANIO2270
	NVAL = NCOUNT(I)	ANIO2280
C		ANIO2290
	DO 200 J = 1,NUMBER	ANIO2300
C		ANIO2310
	XX(J) = XOLD(I,J)	ANIO2320
C		ANIO2330
	IF(NVAL.EQ.1) GO TO 11	ANIO2340
	GO TO 13	ANIO2350
C		ANIO2360
11	DUMMY = DABS(XX(J))	ANIO2370
	IF(DUMMY.GT.22.5000) GO TO 12	ANIO2380
	GO TO 13	ANIO2390
12	YY(J) = DCNPLX(0.000,0.000)	ANIO2400
	GO TO 200	ANIO2410
C		ANIO2420
13	YY(J) = YOLD(I,J)	ANIO2430
C		ANIO2440
200	CONTINUE	ANIO2450
C		ANIO2460
	IF(NVAL.EQ.1) GO TO 10	ANIO2470
	GO TO 15	ANIO2480
C		ANIO2490
10	CALL INTEG(NUMBER,XX,YY,SUM)	ANIO2500
	GO TO 20	ANIO2510
C		ANIO2520
15	CALL MGPHS(NUMBER,YY,YLIN,YMAG,YPHS)	ANIO2530
C		ANIO2540
	CALL SORT(NUMBER,SUB,XX,YY,XNEW,YNEW,INT)	ANIO2550
C		ANIO2560
	CALL MGPHS(INT,YNEW,YNEWL,YNEWM,YNEWP)	ANIO2570
C		ANIO2580
	CALL COMPLX(INT,YNEWL,YNEWP,YNEW)	ANIO2590
C		ANIO2600
	CALL SLIDE(NVAL,INT,XNEW,YNEW,XPROF,YPROF,NTOT)	ANIO2610
C		ANIO2620
	CALL MGPHS(NTOT,YPROF,YLIN,YMAG,YPHS)	ANIO2630
C		ANIO2640
	DO 240 K = 1,INT	ANIO2650
C		ANIO2660
	WRITE(10,2002) K,XX(K),YLIN(K),YYPHS(K),XNEW(K),YNEWL(K),YNEWP(K)	ANIO2670
C	&,XPROF(K),YLIN(K),YPHS(K)	ANIO2680
C240	CONTINUE	ANIO2690
C		ANIO2700
	DO 241 L = 1,NTOT	ANIO2710
C		ANIO2720
	WRITE(10,2002) L,XPROF(L),YLIN(L),YPHS(L)	ANIO2730
C241	CONTINUE	ANIO2740
C		ANIO2750
	WRITE(6,*) NTOT	

FILE ANISOT FORTRAN A BELL-NORTHERN RESEARCH

C	CALL INTEG(NTOT,XPROF,YPROF,SUM)	ANIO2760
C		ANIO2770
20	TOTL(I) = SUM*SINE(I)	ANIO2780
C		ANIO2790
C	WRITE(10,2002) I,TOTL(I)	ANIO2800
C		ANIO2810
C		ANIO2820
100	CONTINUE	ANIO2830
C		ANIO2840
C	CALL POSITN(TOTL,TX,TXMAG,TPHS)	ANIO2850
C		ANIO2860
C	CALL NORM(TX,TXN)	ANIO2870
C		ANIO2880
C	CALL MAGPHS(TXN,TXNLIN,TXNMAG,TXNPHS)	ANIO2890
C		ANIO2900
C	CALL CONVOL(TXN,XDIF,Q,QLIN,QMAG,QPHS)	ANIO2910
C		ANIO2920
C	DO 242 I = 1,NDATA	ANIO2930
C	WRITE(10,2002) I,XDLIN(I),XDPHS(I),TXNLIN(I),TXNPHS(I),QLIN(I)	ANIO2940
C	&,QPHS(I)	ANIO2950
242	CONTINUE	ANIO2960
C		ANIO2970
C	FFT2C FOR EXP(-JBD)	ANIO2980
C	DO 243 I = 1,NDATA	ANIO2990
C	Q(I) = DCONJG(Q(I))	ANIO3000
243	CONTINUE	ANIO3010
C		ANIO3020
C	CALL FFT2C(Q,MN,IWK)	ANIO3030
C		ANIO3040
C	DO 244 I = 1,NDATA	ANIO3050
C	Q(I) = DCONJG(Q(I))/NDATA	ANIO3060
244	CONTINUE	ANIO3070
C		ANIO3080
C	CALL FNORM(Q,FFT)	ANIO3090
C		ANIO3100
C	CALL MAGPHS(FFT,FFTLIN,FFTMAG,FFTPHS)	ANIO3110
C		ANIO3120
C	CALL FRVECT(FR)	ANIO3130
C		ANIO3140
C	DO 250 I = 1,NDATA	ANIO3150
C	WRITE(9,2002) I,FR(I),FFTMAG(I),FFTPHS(I)	ANIO3160
250	CONTINUE	ANIO3170
C		ANIO3180
C	DO 500 I = 184.330	ANIO3190
C	FR(I) = FR(I)/1.00+06	ANIO3200
C	WRITE(3,2001) FR(I),FFTMAG(I)	ANIO3210
500	CONTINUE	ANIO3220
C		ANIO3230
1000	FORMAT(/,5X,'IS YOUR SUBSTRATE ISOTROPIC??? (1 = Y,0 = N)')	ANIO3240
1001	FORMAT(5X,'A = ',1PD16.7)	ANIO3250
1002	FORMAT(5X,'B = ',1PD16.7)	ANIO3260
1003	FORMAT(5X,'C = ',1PD16.7)	ANIO3270
1004	FORMAT(5X,'D = ',1PD16.7,/))	ANIO3280
1005	FORMAT(5X,'THIS SUBSTRATE IS ISOTROPIC!!!',/))	ANIO3290
1006	FORMAT(5X,'THIS SUBSTRATE IS AN-ISOTROPIC!!!',/))	ANIO3300

FILE ANISOT FORTRAN A BELL-NORTHERN RESEARCH

1007	FORMAT(/,5X,'DO YOU WISH TO HAVE AN INCLINE IN THE APODIZED IDT??')	ANIO3310
	&(1=YES,0=NO)')	ANIO3320
1008	FORMAT(5X,'HOW MANY DEGREES??')	ANIO3330
1009	FORMAT(5X,'THE APODIZED IDT HAS AN INCLINE OF BETA(DEG) = '	ANIO3340
	&1PD10.4,/)')	ANIO3350
C		ANIO3360
2001	FORMAT(1P3020.10)	ANIO3370
2002	FORMAT(18,1P10012.4)	ANIO3380
2003	FORMAT(1H1)	ANIO3390
2004	FORMAT(215,1P10012.4)	ANIO3400
C		ANIO3410
C		ANIO3420
	STOP	ANIO3430
	END	ANIO3440

FILE COMPEN FORTRAN A BELL NORTHERN RESEARCH

```

C
C   THIS ROUTINE COMPUTES A NEW DIFFRACTION COMPENSATED DESIGN OF
C   SURFACE ACOUSTIC WAVE FILTER
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION YW(125),DWNLIN(512),DWNMAG(512),DWNPHS(512)
C   DIMENSION FFTLIN(512),FFTMAG(512),FFTPHS(512),FR(512),T(512)
C   DIMENSION TWK(10),XLIN(512),XPHS(512),TWLIN(512),DWER(512)
C   DIMENSION TWPHS(512),HLIN(512),HMAG(512),HPS(512),TWMAG(512)
C   DIMENSION DWLSNL(512),DWLSNM(512),DWLSNP(512),DWRSNL(512)
C   DIMENSION DWRSNM(512),DWRSNP(512),DWSNL(512),DWSNM(512)
C   DIMENSION DWSNP(512),DWSL(512),DWSM(512),DWSP(512),DWP(512)
C   DIMENSION DWLNL(512),DWLNM(512),DWLNP(512),ERRLIN(512),ERRMAG(512)
C   DIMENSION DWRNL(512),DWRNM(512),DWRNP(512),ERRPHS(512)
C   DIMENSION CWLNL(512),CWLNM(512),CWLNP(512),CWRNL(512),CWRNM(512)
C   DIMENSION CWRNP(512),CWSNL(512),CWSNM(512),CWSNP(512),DUMRP(512)
C   DIMENSION TWLL(512),TWLM(512),TWLP(512),TWRL(512),TWRM(512)
C   DIMENSION TWRP(512),DWLL(512),DWLM(512),DWLP(512),DWRL(512)
C   DIMENSION DWRM(512),DWRP(512),CWLL(512),CWLM(512),CWLP(512)
C   DIMENSION CWRL(512),CWRM(512),CWRP(512),CWLDL(512),CWLDM(512)
C   DIMENSION CWLDP(512),CWRDL(512),CWRDM(512),CWRDP(512),TWPHSS(512)
C   DIMENSION TWNEWM(512),TWNEWL(512),TWNEWP(512),TWLINN(512)
C   DIMENSION XLENG(512),ALENG(125),DUMLL(512),DUMLP(512),DUMRL(512)
C   COMPLEX*16 TW(512),DW(512),DWN(512),SINE(512),H(512),FFT(512)
C   COMPLEX*16 FFTN(512),X(512),DWL(512),DWR(512),DWS(512),DWERR(512)
C   COMPLEX*16 DWLS(512),DWLSN(512),DWRS(512),DWRSN(512),DWSN(512)
C   COMPLEX*16 YO(125),TWL(512),TWR(512),CWL(512),CWR(512),CWLD(512)
C   COMPLEX*16 CWRD(512),CWS(512),CWSN(512),SINL(512),SINR(512)
C   COMPLEX*16 DWLN(512),DWRN(512),TWLN(512),TWRN(512),TWNEWN(512)
C   COMPLEX*16 CWLN(512),CWRN(512),TWNEW(512),TXL(125),TXR(125)
C   COMPLEX*16 TWW(2,512),TWTX(512),XDIF(512),XSINE(512)
C   COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA
C
C   DEFINE CONSTANTS /DATA/ OF COMMON
C   PI      = DABS(DARCOS(-1.0D0))
C   FO      = 70 0D+06
C   FLAMBD  = 49 2D-06
C   DELAY   = 5415 52D-06/FLAMBD
C   AP1     = 2461 45D-06/FLAMBD
C   AP2     = 2215 31D-06/FLAMBD
C   AP2     = 45 0D0
C   NFO     = 125
C   NFI     = 19
C   NDATA   = 512
C
C   CONSTANT OF IMSL ROUTINE 'FFT2C'
C   MM      = 9
C
C   DEFINE PARAMETERS
C   DTR     = PI/180 0D0
C   RTD     = 180 0D0/PI
C   THTA    = 45 0D0-DTR
C   ITMAX   = 3
C   ITMIN   = 2

```

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

	RLXM = 0.0000	COM00560
	RLXP = 0.100	COM00570
	IT = 1	COM00580
	NN1 = 57	COM00590
	KFO = -(NFO-1)/2	COM00600
	KFI = (NFI-1)/2	COM00610
C		COM00620
C	IS YOUR SUBSTRATE ISOTROPIC??? (1 = Y, 0 = N)	COM00630
1	WRITE(6,1000)	COM00640
	READ(6,*) ISOT	COM00650
	IF(ISOT NE 1 AND ISOT NE 0) GO TO 1	COM00660
	IF(ISOT EQ 0) GO TO 2	COM00670
C		COM00680
	WRITE(10,1021)	COM00690
	WRITE(9,1021)	COM00700
	WRITE(7,1021)	COM00710
	AR = 0.000	COM00720
	BR = 0.000	COM00730
	CR = 0.000	COM00740
	DR = 0.000	COM00750
	GO TO 3	COM00760
C		COM00770
C	DO POLYNOMIAL CURVEFIT OF VELOCITY VS. ANGLE OF PROPAGATION	COM00780
C	ON LINBO3 SUBSTRATES (BASED ON SLOBODNIK'S DATA)	COM00790
C	A,B,C,D ARE COEFFICIENTS OF 8TH ORDER POLYNOMIAL	COM00800
C		COM00810
2	WRITE(10,1022)	COM00820
	WRITE(9,1022)	COM00830
	WRITE(7,1022)	COM00840
	A = -1.2600D-04	COM00850
	B = 3.6892D-07	COM00860
	C = -5.0887D-10	COM00870
	D = 2.7551D-13	COM00880
C		COM00890
	AR = A*(RTD**2)	COM00900
	BR = B*(RTD**4)	COM00910
	CR = C*(RTD**6)	COM00920
	DR = D*(RTD**8)	COM00930
C		COM00940
	WRITE(6,1029) A	COM00950
	WRITE(7,1029) A	COM00960
	WRITE(9,1029) A	COM00970
C		COM00980
	WRITE(6,1010) B	COM00990
	WRITE(7,1010) B	COM01000
	WRITE(9,1010) B	COM01010
C		COM01020
	WRITE(6,1027) C	COM01030
	WRITE(7,1027) C	COM01040
	WRITE(9,1027) C	COM01050
C		COM01060
	WRITE(6,1028) D	COM01070
	WRITE(7,1028) D	COM01080
	WRITE(9,1028) D	COM01090
C		COM01100

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

C	DO YOU WISH TO HAVE AN INCLINE IN THE APODIZED IDT?(1-Y,0-N)	COMO1110
3	WRITE(6,1023)	COMO1120
	READ(6,*) INCLN	COMO1130
	IF(INCLN.NE.1.AND.INCLN.NE.0) GO TO 3	COMO1140
	IF(INCLN.EQ.1) GO TO 4	COMO1150
	IF(INCLN.EQ.0) BETA = 0.000	COMO1160
	GO TO 5	COMO1170
C		COMO1180
C	HOW MANY DEGREES???	COMO1190
4	WRITE(6,1024)	COMO1200
	READ(6,*) BETA	COMO1210
	WRITE(6,1025) BETA	COMO1220
	WRITE(7,1025) BETA	COMO1230
	WRITE(9,1025) BETA	COMO1240
	BETA = DTR*BETA	COMO1250
C		COMO1260
C	GENERATE THE INPUT UNAPODIZED VECTOR 'X'	COMO1270
5	CALL INPIDT(X,XLIN,XPHS)	COMO1280
C		COMO1290
C	APODIZATION SIGN VECTOR OF THE INPUT UNAPODIZED IDT	COMO1300
	DO 30 I = 1,NDATA	COMO1310
	XLENG(I) = AP2*XLIN(I)	COMO1320
	CMULT = DSIGN(1.000,DREAL(X(I)))	COMO1330
	XSINE(I) = DCMPLX(CMULT,0.000)	COMO1340
30	CONTINUE	COMO1350
C		COMO1360
C	COMPUTE DIFFRACTED WEIGHTS OF THE INPUT IDT	COMO1370
	GAMA = 0.000	COMO1380
	NDAT = NFI	COMO1390
	CALL DIFCOR(NDAT,KFI,AR,BR,CR,DR,GAMA,X,XDIF)	COMO1400
C		COMO1410
C	NORMALIZE X-VECTOR WRT. LARGEST WEIGHT	COMO1420
	DO 54 I = 1,NFI	COMO1430
	XDIF(I) = (XDIF(I)/XDIF(NFI))*XSINE(I)	COMO1440
54	CONTINUE	COMO1450
C		COMO1460
	CALL MAGPHS(XDIF,XDLIN,XDMAG,XDPHS)	COMO1470
C		COMO1480
C	READ ORIGINAL APODIZATION WEIGHTING OF OUTPUT IDT	COMO1490
	DO 50 I=1,NFO	COMO1500
	READ(1,*) YW(I)	COMO1510
	ALENG(I) = DABS(YW(I))*AP2	COMO1520
50	CONTINUE	COMO1530
C		COMO1540
C	OUTPUT IDT 125 SAMPLE COMPLEX VECTOR YO(125)	COMO1550
	CALL OUTIDT(YW,YO)	COMO1560
C		COMO1570
C	POSITION THE OUTPUT IDT AT THE CORRECT DELAY AND GET TW(512)	COMO1580
	CALL POSITN(YO,TWTX,TWLIN,TWPHSS)	COMO1590
C		COMO1600
C	CREATE APODIZATION SIGN VECTOR	COMO1610
	DO 55 I=1,NDATA	COMO1620
	TW(I) = TWTX(I)	COMO1630
	CMULT = DSIGN(1.000,DREAL(TW(I)))	COMO1640
	SINE(I) = DCMPLX(CMULT,0.000)	COMO1650

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

55	CONTINUE	COMO1660
C		COMO1670
C	COMPUTE TRANSMITTING SPLIT FINGERS	COMO1680
	DO 60 I = 1, NDATA	COMO1690
	TWL(I) = (TW(I)/DSORT(2.000))*DCMPLX(DCOS(THTA), DSIN(THTA))	COMO1700
	TWR(I) = (TW(I)/DSORT(2.000))*DCMPLX(DCOS(THTA), DSIN(-THTA))	COMO1710
60	CONTINUE	COMO1720
C		COMO1730
C	READ IN LAST SET OF RELAXED VALUES	COMO1740
	DO 200 I = 1, NFO	COMO1750
	READ(11, 2001) TWLL(I), TWRL(I)	COMO1760
	TXL(I) = DABS(TWLL(I))*SINE(I)*DCMPLX(DCOS(THTA), DSIN(THTA))	COMO1770
	TXR(I) = DABS(TWRL(I))*SINE(I)*DCMPLX(DCOS(THTA), DSIN(-THTA))	COMO1780
200	CONTINUE	COMO1790
C		COMO1800
	CALL POSITN(TXL, TWL, TWLL, TWLP)	COMO1810
	CALL POSITN(TXR, TWR, TWRL, TWRP)	COMO1820
C		COMO1830
	CALL NORML(TWL, TWLN)	COMO1840
	CALL NORML(TWR, TWRN)	COMO1850
C		COMO1860
C	DIFFRACTED WEIGHTS OF VECTOR TW(512)=DW(512)	COMO1870
	NDAT = NFO	COMO1880
10	CALL DIFCOR(NDAT, KFO, AR, BR, CR, DR, BETA, TWLN, DWL)	COMO1890
	CALL DIFCOR(NDAT, KFO, AR, BR, CR, DR, BETA, TWRN, DWR)	COMO1900
C		COMO1910
	DO 66 I=1, NDATA	COMO1920
	DWL(I) = DWL(I)*SINE(I)	COMO1930
	DWR(I) = DWR(I)*SINE(I)	COMO1940
	TWL(1, I) = TW(I)	COMO1950
C		COMO1960
	TWL2(1, I) = TWLN(I)	COMO1970
C		COMO1980
	TWR2(1, I) = TWRN(I)	COMO1990
66	CONTINUE	COMO2000
C		COMO2010
	CALL NORML(DWL, DWLN)	COMO2020
	CALL NORML(DWR, DWRN)	COMO2030
C		COMO2040
	CALL MAGPHS(DWL, DWLL, DWLM, DWLP)	COMO2050
	CALL MAGPHS(DWR, DWRL, DWRM, DWRP)	COMO2060
C		COMO2070
	CALL MAGPHS(TWLN, TWLL, TWLM, TWLP)	COMO2080
	CALL MAGPHS(TWRN, TWRL, TWRM, TWRP)	COMO2090
C		COMO2100
	DO 65 I = 1, NDATA	COMO2110
	DUMLL(I) = TWLL(I)	COMO2120
	DUMLP(I) = TWLP(I)	COMO2130
	DUMRL(I) = TWRL(I)	COMO2140
	DUMRP(I) = TWRP(I)	COMO2150
65	CONTINUE	COMO2160
C		COMO2170
	CALL MAGPHS(TW, TWLN, TWLM, TWPHS)	COMO2180
C		COMO2190
C	PUT DWL IN THE CORRECT QUADRANT	COMO2200
	DO 68 J = 1, NFO	
	I = J+NN1	

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

```

DWLP(I) = DWLP(I)+90.000
IF(DWLP(I) GT.180.000) DWLP(I) = DWLP(I)-360.000
DWL(I) = DWLL(I)*DCMPLX(DCOS(DWLP(I)*DTR),DSIN(DWLP(I)*DTR))
68 CONTINUE
C
C NORMALIZE DWL WRT. ABSOLUTE VALUE OF CENTER WEIGHT
C CALL NORM(DWL,DWLN)
C
C WRITE(10,2003)
C
C ADD DWL & DWR TO GET A SINGLE COMBINED WEIGHT
C DO 70 I = 1,NDATA
DWS(I) = DWL(I)+DWR(I)
70 CONTINUE
C
C NORMALIZE DWS(512) WRT. CENTER WEIGHT = 1
C CALL NORM(DWS,DWSN)
C
C COMPUTE MAGNITUDE AND PHASE OF DWS(512)
C CALL MAGPHS(DWSN,DWSNL,DWSNM,DWSNP)
C
C ERRMIN = 1.00-02
C
C CALL ERROR(TWLINN,TWPHSS,DWSNL,DWSNP,ERRLIN,ERRPHS,ERRMAX)
C
C12 TW(I) = TWW(1,I)
C
C COMPUTE COMPLEX WEIGHTS OF NEW LEFT AND RIGHT SPLIT FINGERS
C DO 100 I=1,NDATA
C
C MAGNITUDE RELAXATION FACTOR 'RLXM'
C RLXM = 0.2500
C PHASE RELAXATION FACTOR 'RLXP'
C RLXP = 0.0000
C
C IF(TWLINN(I) EQ.0.000) GO TO 14
C RATIO = (TWLINN(I)-DWSNL(I))/TWLINN(I)
C IF(DABS(RATIO).LE.1.00-03) ERRLIN(I) = 0.000
C CMULT = DSIGN(1.000,ERRPHS(I))
C IF(DABS(ERRPHS(I)).GE.5.000) ERRPHS(I) = 5.000*CMULT
C IF(DABS(ERRPHS(I)).GE.5.000) RLXP = 1.000
C IF(DABS(ERRPHS(I)).LE.1.000) ERRPHS(I) = 0.000
C
C PHIK = DTR*ERRPHS(I)*RLXP
C ABSL = CDABS(TWLN(I))+RLXM*ERRLIN(I)
C ABSR = CDABS(TWRN(I))+RLXM*ERRLIN(I)
C TWL(I)=ABSL*DCOS(THTA+PHIK)*SINE(I)*DCMPLX(DCOS(THTA),DSIN(THTA))
C TWR(I)=ABSR*DCOS(PHIK-THTA)*SINE(I)*DCMPLX(DCOS(THTA),DSIN(-THTA))
C
C 100 CONTINUE
C
C CALL NORML(TWL,TWLN)
C CALL NORML(TWR,TWRN)
C

```

COMO2210
 COMO2220
 COMO2230
 COMO2240
 COMO2250
 COMO2260
 COMO2270
 COMO2280
 COMO2290
 COMO2300
 COMO2310
 COMO2320
 COMO2330
 COMO2340
 COMO2350
 COMO2360
 COMO2370
 COMO2380
 COMO2390
 COMO2400
 COMO2410
 COMO2420
 COMO2430
 COMO2440
 COMO2450
 COMO2460
 COMO2470
 COMO2480
 COMO2490
 COMO2500
 COMO2510
 COMO2520
 COMO2530
 COMO2540
 COMO2550
 COMO2560
 COMO2570
 COMO2580
 COMO2590
 COMO2600
 COMO2610
 COMO2620
 COMO2630
 COMO2640
 COMO2650
 COMO2660
 COMO2670
 COMO2680
 COMO2690
 COMO2700
 COMO2710
 COMO2720
 COMO2730
 COMO2740
 COMO2750

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

	CALL MAGPHS(TWLN,TWLL,TWLM,TWLP)	COMO2760
	CALL MAGPHS(TWRN,TWRL,TWRM,TWRP)	COMO2770
C		COMO2780
	DO 52 I=1,NDATA	COMO2790
C	TWL2(2,I) = TWLN(I)	COMO2800
C	TWR2(2,I) = TWRN(I)	COMO2810
	TWNEW(I) = TWLN(I)+TWRN(I)	COMO2820
	TWW(2,I) = TWNEWN(I)	COMO2830
52	CONTINUE	COMO2840
C		COMO2850
	CALL NORM(TWNEW,TWNEWN)	COMO2860
C		COMO2870
	CALL MAGPHS(TWNEWN,TWNEWL,TWNEWM,TWNEWP)	COMO2880
C		COMO2890
	WRITE(10,2003)	COMO2900
C		COMO2910
	DO 67 I = 58,182	COMO2920
	RATIO = (TWLINN(I)-DWSNL(I))/TWLINN(I)	COMO2930
	WRITE(10,2002) I,TWLINN(I),DUMLL(I),DUMRL(I),DWLL(I),DWLP(I),	COMO2940
	&DWRL(I),DWRP(I),DWSNL(I),DWSNP(I),RATIO,ERRPHS(I)	COMO2950
67	CONTINUE	COMO2960
C		COMO2970
	DO 69 I = 1,NDATA	COMO2980
	TW(I) = TWNEWN(I)	COMO2990
69	CONTINUE	COMO3000
C		COMO3010
	IT = IT+1	COMO3020
C		COMO3030
	WRITE(6,1003) IT,ERRMAX,ERRMIN	COMO3040
C		COMO3050
	IF(ERRMAX.LT.ERRMIN) GO TO 20	COMO3060
	IF(IT.EQ.ITMAX) GO TO 20	COMO3070
C	IF(IT.EQ.ITMIN) GO TO 15	COMO3080
C		COMO3090
	WRITE(6,1003) IT,ERRMAX	COMO3100
C	GO TO 10	COMO3110
		COMO3120
C		COMO3130
		COMO3140
C15	RLXM = 0.0000	COMO3150
C	RLXP = 1.0000	COMO3160
C	WRITE(6,2001) RLXP,RLXM	COMO3170
C	GO TO 12	COMO3180
C		COMO3190
	PERFORM NUMERICAL CONVOLUTION BETWEEN THE INPUT AND OUTPUT IDT'S	COMO3200
20	CALL CONVOL(DWSN,XDIF,H,HLIN,HMAG,HPHS)	COMO3210
C		COMO3220
	COMPUTE THE FAST FOURIER TRANSFORM OF IMPULSE RESPONSE	COMO3230
C	TO OBTAIN THE FREQUENCY RESPONSE	COMO3240
C	CALL FFT2C(H,MM,IWK)	COMO3250
C		COMO3260
	NORMALIZE 'FFT' WRT. CENTER VALUE	COMO3270
C	CALL FNORM(H,FFTN)	COMO3280
C		COMO3290
	COMPUTE MAGNITUDE LINEAR & DB AND PHASE	COMO3300

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

```

      CALL MAGPHS(FFTN,FFTLIN,FFTMAG,FFTPHS)
C
C      GENERATE FREQUENCIES CORRESPONDING TO ALL 512 SAMPLES
      CALL FRVCT(FR)
C
C      GENERATE TIME DELAY VECTOR 'T' CORRESP. TO ALL 512 SAMPLES
      CALL TIME(T)
C
      WRITE(10,2003)
C
C      PRINT APODIZATION DATA
      DO 11 J = 1,NFO
        I = J+NN1
        TWLL(I) = DUMLL(I)*SINE(I)
        TWRL(I) = DUMRL(I)*SINE(I)
        RATIO = (TWLINN(I)-DWSNL(I))/TWLINN(I)
        WRITE(11,2001) TWLL(I),TWRL(I)
        WRITE(10,2002) I,TWLINN(I),TWPSS(I),TWLL(I),TWLP(I),TWRL(I),
&TWRP(I),DWSNL(I),DWSNP(I),RATIO,ERRPHS(I)
11      CONTINUE
C
C      PRINT RESULTS OF FREQUENCY RESPONSE
      DO 500 I=1,NFO
        WRITE(9,2002) I,FR(I),FFTMAG(I),FFTPHS(I),HLIN(I),HPSH(I)
500      CONTINUE
C
C      GENERATE PLOTTING FILE
      DO 550 I = 184,330
        FR(I) = FR(I)/1.0D+06
        WRITE(3,2001) FR(I),FFTMAG(I)
550      CONTINUE
C
2001 FORMAT(1P2D20.10)
2002 FORMAT(1B,1P1D11.3)
2003 FORMAT(1H1)
C
1000 FORMAT(/,5X,'IS YOUR SUBSTRATE ISOTROPIC??? (I = Y,O = N)')
1001 FORMAT(110,15,2D16.7,1P2D16.6)
1002 FORMAT(9X,'I',4X,'J',8X,'F-RE',10X,'F-IM',10X,'Z-LIMIT',10X,'THTA'
&)
1003 FORMAT(15,1P9D14.6)
1004 FORMAT(1P3D20.10)
1005 FORMAT(///,9X,'I',4X,'J',6X,'FMAG(DB)',8X,'FPHS(DEG)',7X,'Z-LIMIT'
&,9X,'THTA(DEG)')
1006 FORMAT(2X,130(' '))
1007 FORMAT(7X,105(' '))
1008 FORMAT(///,9X,'I',4X,'J',5X,'MAG(ABS)',8X,'FPHS(DEG)',7X,'Z-LIMIT'
&,9X,'THTA(DEG)')
1009 FORMAT(5X,'A = ',1PD16.7)
1010 FORMAT(5X,'B = ',1PD16.7)
1011 FORMAT(/,5X,'I AM NOW COMPUTING DIFFRACTION EFFECT OF THE APODIZED'
& IDT !!!',/)
1012 FORMAT(110,1P10D12.4)
1013 FORMAT(9X,'I',5X,'XX',10X,'RO',10X,'ZU',10X,'ZL',10X,'F-REAL',
&6X,'F-IMAG',6X,'BBSVAL')

```

FILE COMPEN FORTRAN A BELL-NORTHERN RESEARCH

```

1014 FORMAT(4X,'I',4X,'THTA(DEG)',5X,'PHE(DEG)',6X,'VEL (M/S)',5X,'XX',COMO3860
      &12X,'XX1',12X,'K-NBR',8X,'GAM',11X,'ABSVAL',8X,'PHS(DEG)') COMO3870
1015 FORMAT(9X,'I',4X,'FREQ(MHZ)',3X,'MAG(DB)',5X,'PHS(DEG)',4X, COMO3880
      &'MAG(DB)',4X,'PHASE(DEG)',4X,'WT-WITH DIF',8X,'WT-NO DIF') COMO3890
1016 FORMAT(5X,'INPUT VECTOR X(512)') COMO3900
1017 FORMAT(5X,'TIME VECTOR T(512)') COMO3910
1018 FORMAT(5X,'OUTPUT IDT VM(512)') COMO3920
1019 FORMAT(5X,'IMPULSE RESPONSE W(512)') COMO3930
1020 FORMAT(5X,'FAST FOURIER TRANSFORM FFT(512)') COMO3940
1021 FORMAT(5X,'THIS SUBSTRATE IS ISOTROPIC!!!'//) COMO3950
1022 FORMAT(5X,'THIS SUBSTRATE IS AN-ISOTROPIC!!!'//) COMO3960
1023 FORMAT(/,5X,'DO YOU WISH TO HAVE AN INCLINE IN THE APOOIZED IDT???COMO3970
      &(1=YES,0=NO)') COMO3980
1024 FORMAT(5X,'HOW MANY DEGREES??') COMO3990
1025 FORMAT(5X,'THE APOOIZED IDT HAS AN INCLINE OF BETA(DEG)' = ' COMO4000
      &1PD10.4,/) COMO4010
1026 FORMAT(10X,'BE PATIENT, PLEASE!!!'//) COMO4020
1027 FORMAT(5X,'C = ',1PD16.7) COMO4030
1028 FORMAT(5X,'D = ',1PD16.7) COMO4040
1029 FORMAT(5X,'A = ',1PD16.7) COMO4050
C COMO4060
C COMO4070
C COMO4080
C COMO4090

      STOP
      END

```


FILE CONVOL FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE CONVOL (VM,XI,W,WLIN,W MAG,WPHS)	CON00010
C		CON00020
C		CON00030
	IMPLICIT REAL*8(A-H,O-Z)	CON00040
	DIMENSION WSVAL(512),WMAG(512),WPHS(512),WLIN(512)	CON00050
	COMPLEX*16 VM(512),SUM,W(512),XI(512)	CON00060
	COMMON /DATA/ PI,FO,FLAMBDA,DELAY,AP1,AP2,NFO,NFI,NDATA	CON00070
C		CON00080
	CK2 = 180.000/PI	CON00090
C		CON00100
C	PERFORM NUMERICAL CONVOLUTION OF THE REAL INPUT VECTOR 'XX'	CON00110
C	AND THE COMPLEX OUTPUT VECTOR 'AMAG'	CON00120
C		CON00130
	SUM = DCMLPX(0.000,0.000)	CON00140
C		CON00150
	DO 100 I = 1,NDATA	CON00160
	DO 200 J = 1,NDATA	CON00170
C		CON00180
	IJ = J-I+1	CON00190
	IF(IJ LT 1) IJ = IJ+NDATA	CON00200
C		CON00210
	SUM = SUM+XI(IJ)*VM(J)	CON00220
C		CON00230
200	CONTINUE	CON00240
C		CON00250
	W(I) = SUM	CON00260
	SUM = DCMLPX(0.000,0.000)	CON00270
C		CON00280
100	CONTINUE	CON00290
C		CON00300
	AMAX = CDABS(W(1))	CON00310
	DO 300 I = 1,NDATA	CON00320
	IF(CDABS(W(I)).GT.AMAX) AMAX = CDABS(W(I))	CON00330
300	CONTINUE	CON00340
C		CON00350
	DO 400 I = 1,NDATA	CON00360
C	W(I) = W(I)/AMAX	CON00370
C		CON00380
	MAGNITUDE LINEAR	CON00390
	WLIN(I) = CDABS(W(I))	CON00400
C		CON00410
	MAGNITUDE (DB)	CON00420
	WSVAL(I) = CDABS(W(I))	CON00430
	IF(WSVAL(I).LE.0.000) WSVAL(I) = 1.00-12	CON00440
	WMAG(I) = 20.000+DLOG10(WSVAL(I))	CON00450
C		CON00460
	PHASE(DEG)	CON00470
	IF(DIMAG(W(I)).EQ.0.000) GO TO 10	CON00480
	WPHS(I) = CK2*DATAN2(DIMAG(W(I)),DREAL(W(I)))	CON00490
	GO TO 400	CON00500
10	IF(DREAL(W(I)).GE.0.000) WPHS(I) = 0.000	CON00510
	IF(DREAL(W(I)).LT.0.000) WPHS(I) = 180.000	CON00520
C		CON00530
400	CONTINUE	CON00540
C		CON00550

FILE CONVOL FORTRAN A BELL-NORTHERN RESEARCH

C
RETURN
END

CON00560
CON00570
CON00580

FILE DIF FORTRAN A BELL-NORTHERN RESEARCH

```

SUBROUTINE DIF(NNN,KF,AR,BR,CR,DR,BETA,SUBL,XOLD,YOLD,NUMBER)
C
C   THIS PROGRAM MODELS DIFFRACTION ON LINBO3 SUBSTRATES IT CAN BE
C   USED IN DESIGNING SURFACE ACOUSTIC WAVE FILTERS
C
  IMPLICIT REAL*8(A-H,O-Z)
  REAL*4 XXLOW
  DIMENSION X(401),Y(401),XDLN(512),XDMG(512),XDPH(512),RX(512)
  DIMENSION ALENG(125),RO(125),BMAG(512),BPHS(512),CBSVAL(512)
  DIMENSION IWK(10),H(7),YW(125)
  DIMENSION XOLD(125,410),SUBL(125)
  COMPLEX*16 F,F1,F2,C2,HZ(125),SUM,TW(512),TWN(512),DW(512),W(512)
  COMPLEX*16 VM(512),FFT(512),XI(512),WO(125),WD(512),DWN(512)
  COMPLEX*16 FFT1(512),O(512),XDI(512),AN(401),YOLD(125,410)
  COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NF1,NDATA
C
C   DEFINE PARAMETERS
C
  KDATA  = NDATA/2+1
  KFC    = (NFO+1)/2
  DELL   = (AP1-AP2)/2 ODO
  ROO    = DELAY*DFLOAT(KF)*O 500
  APR    = 45 ODO
  NN1    = 57
  CK1    = PI/180 ODO
  CK2    = 180 ODO/PI
  CK3    = PI/2 ODO
C
  WRITE(6,1001) NNN,KF,AR,BR,CR,DR,BETA
C
C   INITIALIZE THE PLOTTING ROUTINE
C
  CALL $PLOTV('PROFILES')
  CALL $PRPLT
  CALL $DATE
  CALL $BOX
  CALL $PN
  CALL $XAXIS('WAVELENGTH (LAMDA)',19,+25,-25,0,10)
  CALL $YAXIS('MAG',3,2,0,2,10)
C
C   EXPLANATION OF SYMBOLS
C
  ALENG = TOTAL LENGTH OF RADIATING ELEMENT (FINGER OF IDT)
  RO    = DISTANCE ALONG DIRECTION OF PROPAGATION (Z-AXIS)
          FROM POINT X=0,Z=0 TO POINT P WHERE THE COMPLEX
          FIELD STRENGTH IS CALCULATED
          RO=0 AT X=0,Z=0
  XX    = DISTANCE ALONG THE X-AXIS FROM X=0 TO THE CENTER OF
          THE RADIATING ELEMENT, FINGER OF IDT, DETERMINES THE
          OFFSET OF THE FINGER POSITION WRT. X=0,Z=0
  XXMAX,XXMIN = EXTREMES OF DISTANCE XX
  DELLX = INCREMENT OF XX WHICH DEFINES A NEW POSITION OF RADIATING
          FINGER WRT X=0,Z=0 FOR THE COMPUTATION OF COMPLEX
          FIELD STRENGTH AT POINT P

```

FILE DIF FORTRAN A BELL-NORTHERN RESEARCH

```

C      Z      = LIMIT OF INTEGRATION                      DIF00560
C      F      = COMPLEX FIELD STRENGTH, RESULT OF FRESNEL'S INTEGRAL DIF00570
C      F(X)=C(X)*JS(X)                                     DIF00580
C                                                         DIF00590
C      XXLIM = -50 000                                     DIF00600
C      XXS   = 50 000                                     DIF00610
C      DELLX = 0 2500                                     DIF00620
C                                                         DIF00630
C                                                         DIF00640
C      I AM NOW COMPUTING DIFFRACTION EFFECT OF THE APODIZED IDT DIF00650
C      WRITE(6,1011)                                       DIF00660
C      WRITE(6,1026)                                       DIF00670
C                                                         DIF00680
C      PRINT HEADINGS - TO PRINT OUT, PROFILES FI 7DISK ID-PROF OUT DIF00690
C      WRITE(7,1014)                                       DIF00700
C      WRITE(7,1006)                                       DIF00710
C                                                         DIF00720
C      DO 100 I= 1,NNN                                     DIF00730
C      II = I-1                                           DIF00740
C      IF(KF GT 0) II = -(I-1)                             DIF00750
C      IF(KF LT 0) KF1 = IABS(KF)                         DIF00760
C                                                         DIF00770
C      INITIALIZE PLOTTER COUNT                           DIF00780
C      N = 1                                              DIF00790
C      M = 1                                              DIF00800
C      SET PLOT PAGE COUNT                                DIF00810
C      NPLOT = 1                                          DIF00820
C                                                         DIF00830
C      COMPUTE RADIATING FINGER'S LENGTH                 DIF00840
C      ALENG(I) = SUBL(I)                                DIF00850
C      BLENG   = ALENG(I)*0 500                          DIF00860
C                                                         DIF00870
C      ABC= DFLOAT(KF1-II)                                DIF00880
C      XD =0 500+DFLOAT(KF1-II)*DTANIBETA               DIF00890
C                                                         DIF00900
C      RO(I) = ROO+DFLOAT(II)*0 500                     DIF00910
C      XXL0W = XXLIM+XD-0 500                             DIF00920
C      INTG  = IFIX(XXLOW)                                DIF00930
C      XX    = DFLOAT(INTG)                               DIF00940
C      IF(BETA EQ 0 000) XX=XXLIM                         DIF00950
C                                                         DIF00960
C      XXSTOP = XXS+XD                                    DIF00970
C      XXMAX  = APR+0 500+XD                              DIF00980
C      XXMIN  = -APR+0 500+XD                             DIF00990
C                                                         DIF01000
C      COMPUTE ANGLE OF PROPAGATION 'THTA'              DIF01010
C      THTAO = DATAN(XX/RO(I))                            DIF01020
C      THTA  = CK2*THTAO                                  DIF01030
C      THTAOA = DABS(THTAO)                               DIF01040
C      WRITE(6,1001) I,N,THTAO,THTA,THTAOA              DIF01050
C                                                         DIF01060
C      THTAO2 = THTAO*THTAO                              DIF01070
C      THTAO3 = THTAO2*THTAO                             DIF01080
C      THTAO4 = THTAO2*THTAO2                             DIF01090
C      THTAO5 = THTAO4*THTAO                             DIF01100

```

FILE DIF FORTRAN A BELL-NORTHERN RESEARCH

	THTAO6 = THTAO4+THTAO2	DIF01110
	THTAO7 = THTAO6+THTAO	DIF01120
	THTAO8 = THTAO6+THTAO2	DIF01130
C	GAMA = -(AR+THTAO2+BR+THTAO4+CR+THTAO6+DR+THTAO8)	DIF01140
	GAMM = 1.000+GAMA	DIF01150
C		DIF01160
C	COMPUTE THE PHASE VELOCITY OF PROPAGATION	DIF01170
	VEL = 3487.7600*(1.000-GAMA)	DIF01180
C		DIF01190
	DVEL = 2.000*AR+THTAO+4.000*BR+THTAO3	DIF01200
	DVEL = 3487.7600*(DVEL+6.000*CR+THTAO5+8.000*DR+THTAO7)	DIF01210
C		DIF01220
C	COMPUTE THE POWER FLOW ANGLE 'PHEE'	DIF01230
	PHEE = DATAN(DVEL/VEL)	DIF01240
C		DIF01250
	PHE = CK2+PHEE	DIF01260
	PHI = DCOS(PHEE)	DIF01270
	GAM = GAMM+PHI	DIF01280
C		DIF01290
C	COMPUTE ANGLE 'ALPHA' OF POWER FLOW PROPAGATION	DIF01300
	ALPHA = THTAO+PHEE	DIF01310
	ALPH = CK2+ALPHA	DIF01320
C		DIF01330
C	COMPUTE DISTANCE XX1 WHERE POWER FLOW ARRIVES	DIF01340
	XX1 = RO(1)*DTAN(ALPHA)	DIF01350
C		DIF01360
	IF(XX1 LT XXLIM) GO TO 10	DIF01370
	IF(XX1 GT XXSTOP) GO TO 10	DIF01380
	GO TO 15	DIF01390
C		DIF01400
10	F = DCMLX(0.000,0.000)	DIF01410
	GO TO 16	DIF01420
C		DIF01430
C	UPPER LIMIT 'Z'	DIF01440
C		DIF01450
15	VU = ((XX1+BLENG)/DSORT(RO(1)+0.500))*DSORT(GAM)	DIF01460
	ZU = CK3+VU+VU	DIF01470
C		DIF01480
C	COMPUTE FRESNEL INTEGRAL 'F1' WITH UPPER LIMIT	DIF01490
	CALL FRES(ZU,F1)	DIF01500
C		DIF01510
C	LOWER LIMIT 'Z'	DIF01520
	VL = ((XX1-BLENG)/DSORT(RO(1)+0.500))*DSORT(GAM)	DIF01530
	ZL = CK3+VL+VL	DIF01540
C		DIF01550
C	COMPUTE FRESNEL INTEGRAL 'F2' WITH LOWER LIMIT	DIF01560
	CALL FRES(ZL,F2)	DIF01570
C		DIF01580
C	CHECK FOR SIGN OF F1 AND F2	DIF01590
	IF((XX1+BLENG).LE.0.000) F1=-F1	DIF01600
	IF((XX1-BLENG).LE.0.000) F2=-F2	DIF01610
C		DIF01620
C	SUBTRACT F2 FROM F1 TO OBTAIN FRESNEL INTEGRAL F	DIF01630
	F = F1-F2	DIF01640
		DIF01650

FILE DIF FORTRAN A BELL-NORTHERN RESEARCH

C		DIF01660
C	ABSOLUTE VALUE OF FRESNEL INTEGRAL F=F1-F2	DIF01670
	BBSVAL=CDABS(F)	DIF01680
C		DIF01690
C	MULTIPLY F BY CONSTANTS IN FRONT OF INTEGRAL SIGN	DIF01700
	C1 = 2.000*PI*RO(I)*GAMA*PHI	DIF01710
	C2 = DCMLX(0.000,-C1)	DIF01720
	F = F*(CDEXP(C2)/DSORT(2.000*RO(I)*GAMM*PHI))	DIF01730
C		DIF01740
C	MULTIPLY 'F' BY COUPLING COEFFICIENT 'K SQUARED'	DIF01750
C16	CKSQ = 1.000*((1.000/CK3)*DABS(THTAO))	DIF01760
C	F = F*CKSQ	DIF01770
C		DIF01780
C	DEFINE 'Y' VARIABLE FOR INTEGRATION	DIF01790
16	AN(N) = F	DIF01800
C		DIF01810
C	COMPUTE MAGNITUDE(LINEAR)	DIF01820
	ABSVAL = CDABS(F)	DIF01830
C		DIF01840
C	COMPUTE PHASE IN DEGREES	DIF01850
	IF(DIMAG(F) EQ 0.000) GO TO 8	DIF01860
	APHS = CK2*DATAN2(DIMAG(F),DREAL(F))	DIF01870
	GO TO 9	DIF01880
C		DIF01890
8	IF(DREAL(F) GE 0.000) APHS=0.000	DIF01900
	IF(DIMAG(F) LT 0.000) APHS=180.000	DIF01910
C		DIF01920
C		DIF01930
C	COMPUTE INTENSITY OF DIFFRACTION PROFILE AT POINT P	DIF01940
9	VALI = ABSVAL*ABSVAL	DIF01950
C		DIF01960
C	ESTABLISH X AND Y AXIS OF PLOTTER	DIF01970
	X(N) = XX1	DIF01980
	Y(N) = ABSVAL	DIF01990
C		DIF02000
	XOLD(I,N) = XX1	DIF02010
	YOLD(I,N) = F	DIF02020
C		DIF02030
C	WRITE(7,1003) I,THTA,PHE,VEL,XX,XX1,GAMM,GAM,ABSVAL,APHS	DIF02040
C	WRITE(7,1003) I,THTA,PHE,ALPH,XX1,RO(I),GAMA1,GAMA2,GAMM,GAM	DIF02050
C		DIF02060
C	ESTABLISH COUNT FOR PLOTTER	DIF02070
	N = N+1	DIF02080
C		DIF02090
C	INCREMENT DISTANCE XX TO COMPUTE NEW COMPLEX FIELD STRENGTH	DIF02100
C	AT POINT P	DIF02110
	XX = XX+DELLX	DIF02120
C		DIF02130
C	CHECK IF ALL COMPLEX FIELDS ALONG DISTANCE XX HAVE BEEN	DIF02140
C	COMPUTED,REFER TO XX=XX+DELLX	DIF02150
	IF(XX.LE.XXSTOP) GO TO 5	DIF02160
C		DIF02170
C	TOTAL NUMBER OF POINTS TO BE PLOTTED	DIF02180
	N = N-1	DIF02190
	NUMBER = N	DIF02200

```

C      WRITE(6,*) N,NUMBER
C
C      CALL ON PLOTTING ROUTINES(SUPPLIED BY BNR'S SYSTEM)
C      CALL $LINE(X,Y,N,O,O,O)
C      CALL $TITLE('DIFRN  PROFILE NO      ',21,0 15)
C      CALL $SYNUM(FLOAT(NPLOT),O)
C      CALL $NPLOTV
C100  CONTINUE
C
C      CLOSE PLOTTING FILE
C      CALL $PLTND
C
C1001 FORMAT(2I5,1P8D12 4)
C1002 FORMAT(9X,'I',4X,'J',8X,'F-RE',10X,'F-IM',10X,'Z=LIMIT',10X,'THTA'
C      &)
C1003 FORMAT(1I5,1P9D14 6)
C1004 FORMAT(1P5D15 6)
C1005 FORMAT(///,9X,'I',4X,'J',6X,'FMAG(DB)',8X,'FPHS(DEG)',7X,'Z=LIMIT'
C      &,9X,'THTA(DEG)')
C1006 FORMAT(2X,130(' '))
C1007 FORMAT(7X,105(' '))
C1008 FORMAT(///,9X,'I',4X,'J',5X,'MAG(ABS)',8X,'FPHS(DEG)',7X,'Z=LIMIT'
C      &,9X,'THTA(DEG)')
C1009 FORMAT(5X,'A = ',1PD16 7)
C1010 FORMAT(5X,'B = ',1PD16 7)
C1011 FORMAT(/,5X,'I AM NOW COMPUTING DIFFRACTION EFFECT OF THE APODIZED'
C      & IDT ' ',/)
C1012 FORMAT(1I10,1P10D12 4)
C1013 FORMAT(9X,'I',5X,'XX',10X,'RO',10X,'ZU',10X,'ZL',10X,'F-REAL',
C      &6X,'F-IMAG',6X,'BBSVAL')
C1014 FORMAT(4X,'I',4X,'THTA(DEG)',5X,'PHE(DEG)',6X,'VEL (M/S)',5X,'XX',
C      &12X,'XX1',12X,'K-NBR',8X,'GAM',11X,'ABSVAL',8X,'PHS(DEG)')
C1015 FORMAT(9X,'I',4X,'FREQ(MHZ)',3X,'MAG(DB)',5X,'PHS(DEG)',4X,
C      &'MAG(DB)',4X,'PHASE(DEG)',4X,'WT-WITH DIF',8X,'WT-NO DIF')
C1025 FORMAT(5X,'THE APODIZED IDT HAS AN INCLINE OF BETA(DEG) = ',
C      &1PD10 4,/)
C1026 FORMAT(10X,'BE PATIENT,PLEASE ' ' ',/)
C
C      RETURN
C      END

```

FILE DIFCOR FORTRAN A BELL-NORTHERN RESEARCH

```

SUBROUTINE DIFCOR(NDAT,KF,AR,BR,CR,DR,BETA,TW,DW)
C
C THIS ROUTINE COMPUTES A NEW DIFFRACTION COMPENSATED DESIGN OF
C SURFACE ACOUSTIC WAVE FILTER
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
C
C DIMENSION NCDUNT(125),SUBL(125),ALENG(125),XOLD(125,410)
C DIMENSION XX(500),XPROF(500),TXMAG(512),TXPHS(512),YPHS(500)
C DIMENSION TXNLIN(512),TXNMAG(512),TXNPHS(512),YLIN(500),YMAG(500)
C DIMENSION IWK(10),FR(512),QLIN(512),OMAG(512),OPHS(512),XSUB(19)
C DIMENSION YYLIN(500),YYMAG(500),YYPHS(500),YNEWL(500),YNEWM(500)
C DIMENSION YNEWP(500),XNEW(500),DWLIN(512),DWPHS(512)
C COMPLEX*16 TW(512),YOLD(125,410),DW(512)
C COMPLEX*16 YNEW(500),YY(500),YPROF(500),TOTL(125),SUM
C COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA
C
C
C DEFINE PARAMETERS
C DTR = PI/180.000
C RTD = 180.000/PI
C NN1 = 57
C
C
C COMPUTE SUBLNGTH OF THE TRANSMITTING FINGER
C N = 25
C DO 60 I = 1,NDAT
C
C   II = I
C   IF(NDAT EQ NFO) II = I+NN1
C   ALENG(I) = AP2*CDABS(TW(II))
C
C   DO 61 J = 1,N
C
C     SUBL(I) = ALENG(I)/DFLOAT(J)
C     IF(SUBL(I) LE 2.000) GO TO 6
C
C 61 CONTINUE
C
C 6 NCDUNT(I) = J
C   WRITE(10,2004) I,NCDUNT(I),ALENG(I),SUBL(I)
C
C 60 CONTINUE
C
C
C CALL DIF(NDAT,KF,AR,BR,CR,DR,BETA,SUBL,XOLD,YOLD,NUMBER)
C
C GO THRU SORTING,SLIDING & INTEGRATION
C
C DO 100 I = 1,NDAT
C   SUB = SUBL(I)
C   NVAL = NCDUNT(I)
C
C DO 200 J = 1,NUMBER
C

```

DIF00010
DIF00020
DIF00030
DIF00040
DIF00050
DIF00060
DIF00070
DIF00080
DIF00090
DIF00100
DIF00110
DIF00120
DIF00130
DIF00140
DIF00150
DIF00160
DIF00170
DIF00180
DIF00190
DIF00200
DIF00210
DIF00220
DIF00230
DIF00240
DIF00250
DIF00260
DIF00270
DIF00280
DIF00290
DIF00300
DIF00310
DIF00320
DIF00330
DIF00340
DIF00350
DIF00360
DIF00370
DIF00380
DIF00390
DIF00400
DIF00410
DIF00420
DIF00430
DIF00440
DIF00450
DIF00460
DIF00470
DIF00480
DIF00490
DIF00500
DIF00510
DIF00520
DIF00530
DIF00540
DIF00550

FILE DIFCOR FORTRAN A BELL-NORTHERN RESEARCH

XX(J) = XOLD(I,J)	DIF00560
C	DIF00570
IF(NVAL EQ 1) GO TO 11	DIF00580
GO TO 13	DIF00590
C	DIF00600
11 DUMMY = DABS(XX(J))	DIF00610
IF(DUMMY GT 22 5000) GO TO 12	DIF00620
GO TO 13	DIF00630
12 YY(J) = DCMLX(0 000,0 000)	DIF00640
GO TO 200	DIF00650
C	DIF00660
13 YY(J) = YOLD(I,J)	DIF00670
C	DIF00680
200 CONTINUE	DIF00690
C	DIF00700
IF(NVAL EQ 1) GO TO 10	DIF00710
GO TO 15	DIF00720
C	DIF00730
10 CALL INTEG(NUMBER,XX,YY,SUM)	DIF00740
GO TO 20	DIF00750
C	DIF00760
15 CALL MGPHS(NUMBER,YY,YYLIN,YYMAG,YYPHS)	DIF00770
C	DIF00780
CALL SORT(NUMBER,SUB,XX,YY,XNEW,YNEW,INT)	DIF00790
C	DIF00800
CALL MGPHS(INT,YNEW,YNEWL,YNEWM,YNEWP)	DIF00810
C	DIF00820
CALL SLIDE(NVAL,INT,XNEW,YNEW,XPROF,YPROF,NTOT)	DIF00830
C	DIF00840
CALL MGPHS(NTOT,YPROF,YLIN,YMAG,YPHS)	DIF00850
C	DIF00860
DO 240 K = 1,INT	DIF00870
C	DIF00880
WRITE(10,2002) K,XX(K),YYLIN(K),YYPHS(K),XNEW(K),YNEWL(K),YNEWP(K)	DIF00890
C	DIF00900
8,XPROF(K),YLIN(K),YPHS(K)	DIF00910
C240 CONTINUE	DIF00920
C	DIF00930
DO 241 L = 1,NTOT	DIF00940
C	DIF00950
WRITE(10,2002) L,XPROF(L),YLIN(L),YPHS(L)	DIF00960
C241 CONTINUE	DIF00970
C	DIF00980
WRITE(6,*) NTOT	DIF00990
C	DIF01000
CALL INTEG(NTOT,XPROF,YPROF,SUM)	DIF01010
C	DIF01020
20 TOTL(I) = SUM	DIF01030
C	DIF01040
WRITE(10,2002) I,TOTL(I)	DIF01050
C	DIF01060
100 CONTINUE	DIF01070
C	DIF01080
IF(NDAT.EQ.NFI) GO TO 7	DIF01090
GO TO 8	DIF01100
C	
7 CALL XPOS(NDAT,TOTL,DW)	
GO TO 9	

FILE DIFCOR FORTRAN A BELL-NORTHERN RESEARCH

C		DIFO1110
	8 CALL POSITN(TOTL,DW,DWLIN,DWPHS)	DIFO1120
C		DIFO1130
	1000 FORMAT(/,5X,'IS YOUR SUBSTRATE ISOTROPIC??? (1 = Y,0 = N)')	DIFO1140
	1001 FORMAT(5X,'A = ',1PD16.7)	DIFO1150
	1002 FORMAT(5X,'B = ',1PD16.7)	DIFO1160
	1003 FORMAT(5X,'C = ',1PD16.7)	DIFO1170
	1004 FORMAT(5X,'D = ',1PD16.7,/))	DIFO1180
	1005 FORMAT(5X,'THIS SUBSTRATE IS ISOTROPIC',/,/)	DIFO1190
	1006 FORMAT(5X,'THIS SUBSTRATE IS AN-ISOTROPIC',/,/)	DIFO1200
	1007 FORMAT(/,5X,'DO YOU WISH TO HAVE AN INCLINE IN THE APODIZED IDT???	DIFO1210
	&(1=YES,0=NO)')	DIFO1220
	1008 FORMAT(5X,'HOW MANY DEGREES??')	DIFO1230
	1009 FORMAT(5X,'THE APODIZED IDT HAS AN INCLINE OF BETA(DEG) = ',	DIFO1240
	&1PD10.4,/,/)	DIFO1250
C		DIFO1260
	2001 FORMAT(1P3D20.10)	DIFO1270
	2002 FORMAT(1B.1P10D12.4)	DIFO1280
	2003 FORMAT(1H1)	DIFO1290
	2004 FORMAT(215.1P10D12.4)	DIFO1300
C		DIFO1310
	9 RETURN	DIFO1320
C		DIFO1330
	END	DIFO1340

FILE ERROR FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE ERROR(TWLIN,TWPHS,CWSNL,CWSNP,ERRLIN,ERRPHS,ERRMAX)	ERR00010
C		ERR00020
C		ERR00030
	IMPLICIT REAL*8 (A-H,O-Z)	ERR00040
	DIMENSION TWLIN(512),TWPHS(512),CWSNL(512),CWSNP(512),ERRLIN(512)	ERR00050
	DIMENSION ERRPHS(512)	ERR00060
	COMMON /DATA/PI,FO,FLAMBDA,DELAY,AP1,AP2,NFO,NFI,NDA	ERR00070
C		ERR00080
C		ERR00090
C	ERRM = 0.000	ERR00100
C		ERR00110
C	DO 100 I=1,NDA	ERR00120
C		ERR00130
C		ERR00140
C	ERRLIN(I) = TWLIN(I)-CWSNL(I)	ERR00150
C		ERR00160
C	ERRPHS(I) = TWPHS(I)+CWSNP(I)	ERR00170
C		ERR00180
C	IF(CWSNP(I) LE -90.000 AND CWSNP(I) GE -180.000) ERRPHS(I) =	ERR00190
	&TWPHS(I)+CWSNP(I)	ERR00200
	IF(CWSNP(I) GE 90.000 AND CWSNP(I) LE 180.000) ERRPHS(I) =	ERR00210
	&CWSNP(I)-TWPHS(I)	ERR00220
C		ERR00230
C	IF(DABS(ERRPHS(I)) GT ERRM) ERRM = DABS(ERRPHS(I))	ERR00240
C		ERR00250
C	100 CONTINUE	ERR00260
C		ERR00270
C	ERRMAX = DABS(ERRM)	ERR00280
C		ERR00290
C	RETURN	ERR00300
	END	ERR00310
		ERR00320

FILE FNORM FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE FNORM(FFT,FFTN)	FN000010
C		FN000020
C	THIS ROUTINE NORMALIZES THE FAST FOURIER TRANSFORM(FFT) WRT	FN000030
C	ITS CENTER VALUE	FN000040
C		FN000050
C		FN000060
	IMPLICIT REAL*8(A-H,O-Z)	FN000070
	COMPLEX*16 FFT(512),FFTN(512),FMAX	FN000080
	COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NF1,NDATA	FN000090
C		FN000100
C	KDATA = NDATA/2+1	FN000110
C		FN000120
C		FN000130
C		FN000140
	FMAX = FFT(KDATA)	FN000150
	DO 100 I=1,NDATA	FN000160
	FFTN(I) = FFT(I)/FMAX	FN000170
100	CONTINUE	FN000180
C		FN000190
C		FN000200
	RETURN	FN000210
	END	FN000220

FILE FRES FORTRAN A BELL-NORTHERN RESEARCH

```

C      SUBROUTINE FRES(Z,F)
C      THIS SUBROUTINE COMPUTES THE FRESNEL INTEGRAL
C      BY USING A SERIES EXPANSION APPROXIMATION
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION A(12),B(12),C(12),D(12)
C      COMPLEX*16 P(12),O(12),F,FIND,G,SUM,DUMMY
C      COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA
C
C      THE NUMERICAL VALUES OF THE FRESNEL COEFFICIENTS
C
C      A(1) = 1.5957691400D0
C      A(2) = -1.70200-06
C      A(3) = -6.80856885400D0
C      A(4) = -5.7636100-04
C      A(5) = 6.92069190200D0
C      A(6) = -1.689865700-02
C      A(7) = -3.0504856600D0
C      A(8) = -7.575241900-02
C      A(9) = 8.5066378100-01
C      A(10) = -2.563904100-02
C      A(11) = -1.502309600-01
C      A(12) = 3.440477900-02
C
C      B(1) = -3.300-08
C      B(2) = 4.25538752400D0
C      B(3) = -9.28100-05
C      B(4) = -7.780020400D0
C      B(5) = -9.52089500-03
C      B(6) = 5.07516129800D0
C      B(7) = -1.3834494700-01
C      B(8) = -1.36372912400D0
C      B(9) = -4.0334927600-01
C      B(10) = 7.0222201600-01
C      B(11) = -2.1619592900-01
C      B(12) = 1.954703100-02
C
C      C(1) = 0.000
C      C(2) = -2.493397500-02
C      C(3) = 3.93600-06
C      C(4) = 5.77095600-03
C      C(5) = 6.8989200-04
C      C(6) = -9.49713600-03
C      C(7) = 1.194880900-02
C      C(8) = -6.74887300-03
C      C(9) = 2.464200-04
C      C(10) = 2.10296700-03
C      C(11) = -1.2179300-03
C      C(12) = 2.3393900-04
C
C      D(1) = 1.994711400-01
C      D(2) = 2.300-08
C      D(3) = -9.35134100-03

```

FRE00010
 FRE00020
 FRE00030
 FRE00040
 FRE00050
 FRE00060
 FRE00070
 FRE00080
 FRE00090
 FRE00100
 FRE00110
 FRE00120
 FRE00130
 FRE00140
 FRE00150
 FRE00160
 FRE00170
 FRE00180
 FRE00190
 FRE00200
 FRE00210
 FRE00220
 FRE00230
 FRE00240
 FRE00250
 FRE00260
 FRE00270
 FRE00280
 FRE00290
 FRE00300
 FRE00310
 FRE00320
 FRE00330
 FRE00340
 FRE00350
 FRE00360
 FRE00370
 FRE00380
 FRE00390
 FRE00400
 FRE00410
 FRE00420
 FRE00430
 FRE00440
 FRE00450
 FRE00460
 FRE00470
 FRE00480
 FRE00490
 FRE00500
 FRE00510
 FRE00520
 FRE00530
 FRE00540
 FRE00550

FILE FRES FORTRAN A BELL-NORTHERN RESEARCH

D(4) = 2 300600-05	FRE00560
D(5) = 4 85146600-03	FRE00570
D(6) = 1 90321800-03	FRE00580
D(7) = -1 712291400-02	FRE00590
D(8) = 2 906406700-02	FRE00600
D(9) = -2 792895500-02	FRE00610
D(10) = 1 649730800-02	FRE00620
D(11) = -5 59851500-03	FRE00630
D(12) = 8 3838600-04	FRE00640
C	FRE00650
C	FRE00660
N = 12	FRE00670
G = DCMLPX(1.000,-1.000)	FRE00680
C	FRE00690
Z=DABS(Z)	FRE00700
C	FRE00710
LIMIT 'Z' FOR OVERFLOW ERROR 'IMO 2081'	FRE00720
IF(Z LE 1 D-12) Z=1 D-12	FRE00730
IF(Z GE 1 D+12) Z=1 D+12	FRE00740
C	FRE00750
FIND = DCMLPX(0.000,-Z)	FRE00760
C	FRE00770
C	FRE00780
C	FRE00790
CHECK IF LIMIT 'Z' IS GREATER THAN 4.0	FRE00800
C	FRE00810
IF(Z GE 4.000) GO TO 110	FRE00820
C	FRE00830
C	FRE00840
C	FRE00850
PERFORM SERIES EXPANSION FOR FRESNEL INTEGRAL	FRE00860
C	FRE00870
SUM=DCMLPX(0.000,0.000)	FRE00880
Z4=1.000	FRE00890
C	FRE00900
DO 100 I=1,N	FRE00910
C	FRE00920
P(I)=(DCMLPX(A(I),B(I)))*Z4	FRE00930
C	FRE00940
WRITE(3,1000) I,Z,Z4,P(I)	FRE00950
C	FRE00960
Z4=(Z/4.000)*Z4	FRE00970
C	FRE00980
SUM=SUM+P(I)	FRE00990
C	FRE01000
C	FRE01010
IF(Z4 LE 1 D-20) GO TO 110	FRE01020
C	FRE01030
100 CONTINUE	FRE01040
C	FRE01050
110 F=SUM*(DSQRT(Z/4.000))*DCMLPX(DCOS(Z),-DSIN(Z))	FRE01060
C	FRE01070
DUMMY = CDEXP(FIND)	FRE01080
C	FRE01090
WRITE(6,1000)	FRE01100
C	
GO TO 20	
C	
C	
FRESNEL INTEGRAL FOR Z>4.0	
C	
10 SUM=DCMLPX(0.000,0.000)	
C	
Z4=1.000	
C	
DO 200 J=1,N	
C	
P(J)=(DCMLPX(C(J),D(J)))*Z4	

FILE FRES FORTRAN A BELL-NORTHERN RESEARCH

C	WRITE(3,1000) J,Z,Z4,P(J)	FREQ1110
	Z4=(4.000/Z)*Z4	FREQ1120
C		FREQ1130
	SUM=SUM+P(J)	FREQ1140
C		FREQ1150
	IF(Z4.LE 1.D-20) GO TO 210	FREQ1160
C		FREQ1170
200	CONTINUE	FREQ1180
C		FREQ1190
C		FREQ1200
210	F=SUM*(DSQRT(4.000/Z))*DCMLX(DCOS(Z),-DSIN(Z))+G/2.000	FREQ1210
C	DUMMY = CDEXP(FIND)	FREQ1220
C	WRITE(6,1000) FIND,DUMMY	FREQ1230
CFILE	FORMAT(I10,1P4D20,10)	FREQ1240
C		FREQ1250
20	RETURN	FREQ1260
	END	FREQ1270

FILE FRVECT FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE FRVECT(FR)	FRV00010
C		FRV00020
C		FRV00030
C	THIS ROUTINE GENERATES THE FREQUENCIES CORRESPONDING TO ALL	FRV00040
C	512 SAMPLES OF 70MHZ FILTERDESIGN	FRV00050
C		FRV00060
C		FRV00070
	IMPLICIT REAL*8 (A-H,O-Z)	FRV00080
	DIMENSION FR(512)	FRV00090
	COMMON /DATA/PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA	FRV00100
C		FRV00110
C		FRV00120
	OMEGAS = 2.000*FO	FRV00130
	DOMEGA = OMEGAS/NDATA	FRV00140
C		FRV00150
C		FRV00160
	DO 100 I=1,NDATA	FRV00170
	J = I-1	FRV00180
C		FRV00190
	CMULT = DFLOAT(J)	FRV00200
	FREQ = CMULT*DOMEGA	FRV00210
	FR(I) = FREQ	FRV00220
C		FRV00230
C		FRV00240
C	100 CONTINUE	FRV00250
C		FRV00260
C		FRV00270
	RETURN	FRV00280
	END	FRV00290

FILE INPIDT FORTRAN A BELL-NORTHERN RESEARCH

```

SUBROUTINE INPIDT(XI XLIN,XPHS)
C
C THIS PROGRAM GENERATES COMPLETE INPUT IDT DATA FILE
C
C IMPLICIT REAL*8(A-H O-Z)
C DIMENSION XLIN(512),XPHS(512)
C COMPLEX*16 XI(512)
C COMMON /DATA PI FO,FLAMBDA,DELAY AP1,AP2,NFO NFI NDATA
C
C ABBREVIATIONS
C
C FO = CENTRE FREQUENCY
C FL = BANDWIDTH TO ROLLOFF (2*F1)
C NFI = NUMBER OF FINGERS OF INPUT TRANSDUCER
C DELAY = SEPARATION OF INPUT AND OUTPUT IDT, CENTER TO CENTER
C
C DEFINE PARAMETERS
C CK2=180.000/PI
C KFI=NFI+1
C
C GENERATE INPUT IDT DATA
C
C DO 101 I=1,NFI
C U=(I+1)**(I+1)
C CMULT=DFLOAT(U)
C XI(I) = DCMPLX(CMULT,0.000)
101 CONTINUE
C
C DO 102 I=KFI,NDATA
C XI(I)=DCMPLX(10.000,0.000)
102 CONTINUE
C
C DO 103 I=1,NDATA
C
C MAGNITUDE(LINEAR)
C XLIN(I) = CDABS(XI(I))
C
C PHASE(DEG)
C IF(DREAL(XI(I)) GE 0.000) XPHS(I)=0.000
C IF(DREAL(XI(I)) LT 0.000) XPHS(I)=180.000
103 CONTINUE
C
C RETURN
C
C END

```

INP00010
INP00020
INP00030
INP00040
INP00050
INP00060
INP00070
INP00080
INP00090
INP00100
INP00110
INP00120
INP00130
INP00140
INP00150
INP00160
INP00170
INP00180
INP00190
INP00200
INP00210
INP00220
INP00230
INP00240
INP00250
INP00260
INP00270
INP00280
INP00290
INP00300
INP00310
INP00320
INP00330
INP00340
INP00350
INP00360
INP00370
INP00380
INP00390
INP00400
INP00410
INP00420
INP00430
INP00440
INP00450


```

FILE MAGPHS FORTRAN A BELL-NORTHERN RESEARCH

SUBROUTINE MAGPHS(X ALIN AMAG APHS)

IMPLICIT REAL*8(A-H D-Z)
DIMENSION AMAG(512) APHS(512) ALIN(512)
COMPLEX*16 X(512)
COMMON DATA PI EQ 3.14159265358979323846
      DELAY AP1 AP2 NFO NF1 NDATA

DO BC = 1, NDATA

  COMPUTE THE MAGNITUDE IN LINEAR
  ABSVAL = CDABS(X(BC))
  ALIN(BC) = ABSVAL

  MAGNITUDE IN DB'S
  IF ABSVAL LE 0.01 ABSVAL = 0.01
  AMAG(BC) = 20.000 - DLOG10(ABSVAL)

  COMPUTE THE PHASE OF FTR IN DEGREES
  IF DIMAG(X(BC)) EQ 0.000 GO TO 50
  APHS(BC) = (180.000 PI) + DATAN2(DIMAG(X(BC)) DREAL(X(BC)))
  GO TO 80

50 IF DREAL(X(BC)) GE 0.000 APHS(BC) = 0.000
   IF DREAL(X(BC)) LT 0.000 APHS(BC) = 180.000

80 CONTINUE

RETURN
END

```

MAG00010
 MAG00020
 MAG00030
 MAG00040
 MAG00050
 MAG00060
 MAG00070
 MAG00080
 MAG00090
 MAG00100
 MAG00110
 MAG00120
 MAG00130
 MAG00140
 MAG00150
 MAG00160
 MAG00170
 MAG00180
 MAG00190
 MAG00200
 MAG00210
 MAG00220
 MAG00230
 MAG00240
 MAG00250
 MAG00260
 MAG00270
 MAG00280
 MAG00290
 MAG00300
 MAG00310
 MAG00320

FILE MGPHS FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE MGPHS(NN,X,ALIN,AMAG,APHS)	MGPO0010
C		MGPO0020
C		MGPO0030
	IMPLICIT REAL*8(A-H,O-Z)	MGPO0040
	DIMENSION AMAG(500),APHS(500),ALIN(500)	MGPO0050
	COMPLEX*16 X(500)	MGPO0060
	COMMON /DATA/ PI,FO,FLAMBDA,DELAY,AP1,AP2,NFO,NFI,NDATA	MGPO0070
C		MGPO0080
C		MGPO0090
	DO 80 I=1,NN	MGPO0100
C		MGPO0110
C	COMPUTE THE MAGNITUDE IN LINEAR	MGPO0120
	ABSVAL = CDABS(X(I))	MGPO0130
	ALIN(I)= ABSVAL	MGPO0140
C		MGPO0150
C	MAGNITUDE IN DB'S	MGPO0160
	IF(ABSVAL LE 0.0) ABSVAL=1D-12	MGPO0170
	AMAG(I) =20.000-DLOG10(ABSVAL)	MGPO0180
C		MGPO0190
C	COMPUTE THE PHASE OF FTR IN DEGREES	MGPO0200
C		MGPO0210
	IF(DIMAG(X(I)) EQ 0.000) GO TO 50	MGPO0220
	APHS(I) =(180.000/PI)*DATAN2(DIMAG(X(I)),DREAL(X(I)))	MGPO0230
	GO TO 80	MGPO0240
C		MGPO0250
50	IF(DREAL(X(I)) GE 0.000) APHS(I) =0.000	MGPO0260
	IF(DREAL(X(I)) LT 0.000) APHS(I) =180.000	MGPO0270
C		MGPO0280
80	CONTINUE	MGPO0290
C		MGPO0300
	RETURN	MGPO0310
	END	MGPO0320

FILE MODPLT FORTRAN A BELL-NORTHERN RESEARCH

```

C      THIS IS A PLOTTING ROUTINE USED WITH THE T4013 GRAPHICS TERMINAL
C      AFTER PLOTTING COMPLETED TO DISPLAY PLOT TYPE ;
C      DISPLAY 'PLOTFILE' T4013 TEXT
C      TO GET A HARD COPY FROM THE PLOTTER DOWNSTAIRS, TYPE
C      MERPLOT 'PLOTFILE' ( PEN 80
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      REAL*4 X(500),Y(500),FREQ(10)
C
C      CALL $PLOTH('MODPLLOT ')
C      CALL $DATE
C      CALL $GRD
C
C      CALL $XAXIS('FREQ (MHZ)',11,90,50,0.20)
C      CALL $YAXIS('MAG (DB)',9,5,-105,0.11)
C
C      NPLOT=1
C
C      WRITE(6,1)
C      1 FORMAT('O*ENTER THE NUMBER OF LINES(10 MAX)')
C      READ(1,*,END=30)NF
C      DO 7 I=1,NF
C      WRITE(6,2)I
C      2 FORMAT('O*WHAT IS LINE NO ',1X,I10,1X 'IN MA?')
C      READ(1,*)FREQ(I)
C      WRITE(6,3)FREQ(I)
C      3 FORMAT('O*HOW MANY POINTS AT ',1X,F10,2,1X,'MA')
C      READ(1,*)NP
C      WRITE(6,4)
C      4 FORMAT('O*ENTER DATA POINTS(FREQ(MHZ),MAG(DB))')
C      DO 6 J=1,NP
C      READ(1,*)X(J),Y(J)
C      WRITE(6,*) X(J),Y(J)
C      6 CONTINUE
C      IF(I EQ.NF) GO TO 8
C      CALL $LINE(X,Y,NP,0,0)
C      7 CONTINUE
C      GO TO 9
C      8 CALL $LINE(X,Y,NP,0,0)
C      GO TO 7
C
C      9 CALL $TITLE('70MHZ FILTER ISOT COMPEN VS IDEAL PLOT NO
C      &,45,0.15)
C      CALL $SYNTH(FLOAT(NPLOT),0)
C      30 CALL $PLTNO
C
C      C900 FORMAT(1P2D10.2)
C
C      STOP
C      END

```

FILE NORM FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE NORM(A,AN)	NOR00010
C		NOR00020
C	THIS SUBROUTINE NORMALIZES A COMPLEX VECTOR WRT ITS LARGEST VALUE	NOR00030
C		NOR00040
	IMPLICIT REAL*8 (A-H,O-Z)	NOR00050
	COMPLEX*16 A(512),AN(512),AMAX	NOR00060
	COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA	NOR00070
C		NOR00080
C		NOR00090
	INTDEL = 110	NOR00100
	NFIC = (NFI+1)/2	NOR00110
	NFOC = NFIC+INTDEL	NOR00120
C		NOR00130
	AMAX = A(NFOC)	NOR00140
	DO 200 I=1,NDATA	NOR00150
	AN(I) = A(I)/AMAX	NOR00160
200	CONTINUE	NOR00170
C		NOR00180
C		NOR00190
	RETURN	NOR00200
	END	NOR00210

FILE NORML FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE NORML(A,AN)	NOR00010
C		NOR00020
C	THIS SUBROUTINE NORMALIZES A COMPLEX VECTOR WRT. ITS LARGEST VALUE	NOR00030
C		NOR00040
	IMPLICIT REAL*8 (A-H,O-Z)	NOR00050
	COMPLEX*16 A(512),AN(512)	NOR00060
	COMMON /DATA/ PI,FO,FLAMBO,DELAY,AP1,AP2,NFO,NFI,NDATA	NOR00070
C		NOR00080
C		NOR00090
	INTDEL = 110	NOR00100
	NFIC = (NFI+1)/2	NOR00110
	NFOC = NFIC+INTDEL	NOR00120
C		NOR00130
	AMAX = CDABS(A(NFOC))	NOR00140
	DO 200 I=1,NDATA	NOR00150
	AN(I) = A(I)/AMAX	NOR00160
200	CONTINUE	NOR00170
C		NOR00180
C		NOR00190
	RETURN	NOR00200
	END	NOR00210

FILE OUTIDT FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE OUTIDT(YW,WO)	OUT00010
C		OUT00020
C	THIS PROGRAM GENERATES COMPLETE OUTPUT IDT DATA FILE	OUT00030
C		OUT00040
	IMPLICIT REAL*8(A-H,O-Z)	OUT00050
	DIMENSION YW(125)	OUT00060
	COMPLEX*16 WO(125)	OUT00070
	COMMON /DATA/ PI,FO,FLAMBDA,DELAY,AP1,AP2,NFO,NFI,NDATA	OUT00080
C		OUT00090
C	ABBREVIATIONS	OUT00100
C		OUT00110
C	FO = CENTRE FREQUENCY	OUT00120
C	F1 = BANDWIDTH TO ROLLOFF (2*F1)	OUT00130
C	NFI = NUMBER OF FINGERS OF INPUT TRANSDUCER	OUT00140
C	DELAY = SEPARATION OF INPUT AND OUTPUT IDT, CENTER TO CENTER	OUT00150
C		OUT00160
C	DEFINE PARAMETERS	OUT00170
	KFO=NFO+1	OUT00180
C		OUT00190
C	GENERATE OUTPUT IDT DATA	OUT00200
C		OUT00210
	DO 101 I=1,NFO	OUT00220
	WO(I) = DCMLPX(YW(I),0.000)	OUT00230
101	CONTINUE	OUT00240
C		OUT00250
C	DO 102 I=KFO,NDATA	OUT00260
C	WO(I)=DCMLPX(0.000,0.000)	OUT00270
C102	CONTINUE	OUT00280
C		OUT00290
C		OUT00300
	RETURN	OUT00310
	END	OUT00320

FILE POSITN FORTRAN A BELL-NORTHERN RESEARCH

```

C          SUBROUTINE POSITN(AMAG,VM,VMAG,VPHS)
C
C          THIS PROGRAM POSITIONS THE OUTPUT IDT AT THE SPECIFIED DELAY
C          WRT. THE INPUT IDT
C
C          IMPLICIT REAL*8(A-H,O-Z)
C          REAL*4 DELAYS
C          DIMENSION VMAG(512),VPHS(512)
C          COMPLEX*16 AMAG(125),VM(512)
C          COMMON /DATA/PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA
C
C          DEFINE PARAMETERS
C
C          CK2      = 180.000/PI
C          DELAYS    = SNGL(DELAY)
C          INTDEL    = IFIX(DELAYS)
C          IFC       = (NFI+1)/2
C          KFC       = (NFO+1)/2
C          NCD       = IFC+INTDEL
C          NN1       = NCD-KFC
C          NN2       = NN1+NFO+1
C
C          DO 100 I=1,NFO
C             J=I+NN1
C             VM(J) = AMAG(I)
C          100 CONTINUE
C
C          DO 200 I=1,NN1
C             VM(I) = DCMLPX(0.000,0.000)
C          200 CONTINUE
C
C          DO 300 I=NN2,NDATA
C             VM(I) = DCMLPX(0.000,0.000)
C          300 CONTINUE
C
C          AMAX = CDABS(VM(1))
C          DO 400 I=1,NDATA
C             IF(CDABS(VM(I)).GT.AMAX) AMAX=CDABS(VM(I))
C          400 CONTINUE
C
C          DO 500 I=1,NDATA
C             VM(I) = VM(I)/AMAX
C
C          MAGNITUDE(LINEAR)
C          VMAG(I) = CDABS(VM(I))
C
C          PHASE(DEG)
C          IF(DIMAG(VM(I)).EQ.0.000) GO TO 50
C          VPHS(I) = CK2*DATAN2(DIMAG(VM(I)),DREAL(VM(I)))
C          GO TO 500
C
C          50  IF(DREAL(VM(I)).GE.0.000) VPHS(I)=0.000
C             IF(DREAL(VM(I)).LT.0.000) VPHS(I)=180.000

```

POS00010
 POS00020
 POS00030
 POS00040
 POS00050
 POS00060
 POS00070
 POS00080
 POS00090
 POS00100
 POS00110
 POS00120
 POS00130
 POS00140
 POS00150
 POS00160
 POS00170
 POS00180
 POS00190
 POS00200
 POS00210
 POS00220
 POS00230
 POS00240
 POS00250
 POS00260
 POS00270
 POS00280
 POS00290
 POS00300
 POS00310
 POS00320
 POS00330
 POS00340
 POS00350
 POS00360
 POS00370
 POS00380
 POS00390
 POS00400
 POS00410
 POS00420
 POS00430
 POS00440
 POS00450
 POS00460
 POS00470
 POS00480
 POS00490
 POS00500
 POS00510
 POS00520
 POS00530
 POS00540
 POS00550

FILE POSITN FORTRAN A BELL-NORTHERN RESEARCH

500 CONTINUE
C
RETURN
END

POS00580
POS00570
POS00580
POS00590

FILE SLIDE FORTRAN A BELL-NORTHERN RESEARCH

```

SUBROUTINE SLIDE(NCOUNT,INT,XOLD,YOLD,XPROF,YPROF,NTOT)      SLI00010
C                                                                SLI00020
C THIS ROUTINE LINES UP AND ADDS Y-AXIS VALUES OF INDIVIDUAL PROFILE SLI00030
C IN ORDER TO OBTAIN THE FINAL DIFFRACTION PROFILE           SLI00040
C                                                                SLI00050
C                                                                SLI00060
C IMPLICIT REAL*8(A-H,O-Z)                                    SLI00070
C DIMENSION XOLD(500),XNEW(500),XPROF(500)                   SLI00080
C COMPLEX*16 YOLD(500),Y(500),YNEW(500),YPROF(500)          SLI00090
C COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA      SLI00100
C                                                                SLI00110
C                                                                SLI00120
C NCENTR = (INT+1)/2                                          SLI00130
C                                                                SLI00140
C NINC = 8                                                    SLI00150
C NHALF = NINC/2                                              SLI00160
C                                                                SLI00170
C N = 1                                                       SLI00180
C M = 1                                                       SLI00190
C                                                                SLI00200
C                                                                SLI00210
C INITIALIZE THE VALUES OF Y(500)                           SLI00220
C DO 50 I = 1,INT                                             SLI00230
C Y(I) = DCMPLEX(0.000,0.000)                                SLI00240
50 CONTINUE                                                  SLI00250
C                                                                SLI00260
C DO 100 I = 1,NCOUNT                                         SLI00270
C IM1 = I - 1                                                 SLI00280
C                                                                SLI00290
C NSTEP = NHALF-NCOUNT-NHALF-IM1+NINC                         SLI00300
C                                                                SLI00310
C DO 200 J = 1,INT                                             SLI00320
C                                                                SLI00330
C XX = DABS(XOLD(J))                                           SLI00340
C IF(XX GT.22.5000) GO TO 200                                  SLI00350
C                                                                SLI00360
C JJ = J + NSTEP                                               SLI00370
C                                                                SLI00380
C XNEW(N) = XOLD(J)                                             SLI00390
C                                                                SLI00400
C YNEW(N) = YOLD(JJ)+ Y(N)                                     SLI00410
C                                                                SLI00420
C Y(N) = YNEW(N)                                               SLI00430
C N = N + M                                                     SLI00440
C                                                                SLI00450
C 200 CONTINUE                                                 SLI00460
C                                                                SLI00470
C NTOT = N-1                                                  SLI00480
C                                                                SLI00490
C N = 1                                                       SLI00500
C                                                                SLI00510
C 100 CONTINUE                                                 SLI00520
C                                                                SLI00530
C DO 300 I = 1,NTOT                                           SLI00540
C XPROF(I) = XNEW(I)                                           SLI00550

```

FILE SLIDE FORTRAN A BELL-NORTHERN RESEARCH

YPROF(I) = YNEW(I)	SL100560
300 CONTINUE	SL100570
C	SL100580
C WRITE(10,2002)	SL100590
C	SL100600
C IF((NCOUNT/2)+2 EQ NCOUNT) GO TO 49	SL100610
C	SL100620
C DO 400 I = 1,NTOT	SL100630
C IM1 = I-1	SL100640
C J = 159+8*IM1	SL100650
C K = 159+IM1	SL100660
C L = 159-8*IM1	SL100670
C	SL100680
C WRITE(10,2001) I,J,K,L,XPROF(I),YOLD(J),YOLD(K),YOLD(L),YPROF(I)	SL100690
C	SL100700
C400 CONTINUE	SL100710
C	SL100720
C GO TO 48	SL100730
C	SL100740
C 49 DO 500 I = 1,NTOT	SL100750
C IM1 = I-1	SL100760
C J = 127+4*IM1	SL100770
C K = 127-4*IM1	SL100780
C	SL100790
C WRITE(10,2001) I,I,J,K,XPROF(I),YOLD(J),YOLD(K),YPROF(I)	SL100800
C	SL100810
C500 CONTINUE	SL100820
C	SL100830
C	SL100840
C2001 FORMAT(4I5,1P9D12.4)	SL100850
C2002 FORMAT(1H1)	SL100860
C	SL100870
C RETURN	SL100880
C END	SL100890

FILE SORT FORTRAN A BELL-NORTHERN RESEARCH

C	SUBROUTINE SORT(NBR,SUB,XOLD,YOLD,XNEW,YNEW,INTEG)	SOR00010
C	IMPLICIT REAL*8(A-H,O-Z)	SOR00020
C	DIMENSION XOLD(500),XNEW(500),YLIN(500),YPHS(500)	SOR00030
C	COMPLEX*16 YOLD(500),YNEW(500)	SOR00040
C	COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA	SOR00050
C		SOR00060
C	XSTOP = 50 000	SOR00070
C		SOR00080
C		SOR00090
C		SOR00100
C		SOR00110
C	GENERATE NEW X-AXIS GRID FOR FINGERS LONGER THAN TWO WAVELENGTH	SOR00120
C	DELXO = SUB/8.000	SOR00130
C	INT = XSTOP/DELXO+0.5000	SOR00140
C	INTM1 = INT-1	SOR00150
C	INTEG = INT+INTM1	SOR00160
C		SOR00170
C	DO 100 J = 1,INT	SOR00180
C	JM1 = J - 1	SOR00190
C	I = J+INTM1	SOR00200
C	K = INT-JM1	SOR00210
C		SOR00220
C	DUMMY = DFL0AT(JM1)*DELXO	SOR00230
C	IF(DUMMY.GT.XSTOP) GO TO 100	SOR00240
C		SOR00250
C	XNEW(I) = DUMMY	SOR00260
C	XNEW(K) = -DUMMY	SOR00270
C		SOR00280
C	100 CONTINUE	SOR00290
C		SOR00300
C	PERFORM LINEAR INTERPOLATION OF Y-AXIS VALUES	SOR00310
C		SOR00320
C	DO 150 I = 1,INTEG	SOR00330
C	IP1 = I+1	SOR00340
C		SOR00350
C	XDUM = XNEW(I)	SOR00360
C		SOR00370
C	DO 200 J = 1,NBR	SOR00380
C	JP1 = J+1	SOR00390
C		SOR00400
C	YO = DREAL(YOLD(J))	SOR00410
C	Y1 = DREAL(YOLD(JP1))	SOR00420
C		SOR00430
C	YOO = DIMAG(YOLD(J))	SOR00440
C	Y11 = DIMAG(YOLD(JP1))	SOR00450
C		SOR00460
C	XO = XOLD(J)	SOR00470
C	X1 = XOLD(JP1)	SOR00480
C		SOR00490
C	DELX = DABS(X1-XO)	SOR00500
C		SOR00510
C	IF(XDUM.GE.XO.AND.XDUM.LE.X1) GO TO 20	SOR00520
C		SOR00530
C	GO TO 200	SOR00540
		SOR00550

FILE SORT FORTRAN A BELL-NORTHERN RESEARCH

```
C
20  R = DABS(XDUM-XO)/DELX
C
      DELY1 = Y1 - YO
      DELY2 = Y11 - YOO
C
      YNEWL = YO + R*DELY1
      YNEWP = YOO+ R*DELY2
C
      GO TO 15
C
200  CONTINUE
C
      15  YNEW(I) = DCMPLX(YNEWL,YNEWP)
C
150  CONTINUE
C
      RETURN
      END
```

SOR00560
SOR00570
SOR00580
SOR00590
SOR00600
SOR00610
SOR00620
SOR00630
SOR00640
SOR00650
SOR00660
SOR00670
SOR00680
SOR00690
SOR00700
SOR00710
SOR00720
SOR00730
SOR00740

FILE TIME FORTRAN A BELL-NORTHERN RESEARCH

SUBROUTINE TIME(T)

THIS PROGRAM GENERATES COMPLETE TIME VECTOR CORRESPONDING
TO ALL 512 SAMPLES

IMPLICIT REAL*8(A-H,O-Z)

DIMENSION T(512)

COMMON DATA/PI FO,FLMBD,DELAY,AP1,AP2 NFO,NFI,NDATA

ABBREVIATIONS

FO = CENTRE FREQUENCY

F1 = BANDWIDTH TO ROLLOFF (2*F1)

NFI = NUMBER OF FINGERS OF INPUT TRANSDUCER

NFO = NUMBER OF FINGERS OF OUTPUT TRANSDUCER

DTIME = STEP IN TIME FROM FINGER TO FINGER

TLENG = TOTAL LENGTH OF TRANSDUCER IN TIME

DELAY = SEPARATION OF INPUT AND OUTPUT IOT,CENTER TO CENTER

DEFINE PARAMETERS

NFI=(NFI+1)/2

DTIME=1/(2.000*FO)

ETIME=DTIME/2.000

GENERATE TIME VECTOR

DO 103 I=1,NDATA

TEST FOR EVEN NFI

IF((NFI,2)=2 EQ NFI) GO TO 10

CALCULATION FOR ODD NFI

J=I-NFI

CMULT=DFLOAT(J)

TIME=CMULT*DTIME

T(I)=TIME

GO TO 103

CALCULATION FOR EVEN NFI

10 J=I-NFI/2

CMULT=DFLOAT(J)

TIME=CMULT*DTIME-ETIME

T(I)=TIME

103 CONTINUE

RETURN

END

TIM00010

TIM00020

TIM00030

TIM00040

TIM00050

TIM00060

TIM00070

TIM00080

TIM00090

TIM00100

TIM00110

TIM00120

TIM00130

TIM00140

TIM00150

TIM00160

TIM00170

TIM00180

TIM00190

TIM00200

TIM00210

TIM00220

TIM00230

TIM00240

TIM00250

TIM00260

TIM00270

TIM00280

TIM00290

TIM00300

TIM00310

TIM00320

TIM00330

TIM00340

TIM00350

TIM00360

TIM00370

TIM00380

TIM00390

TIM00400

TIM00410

TIM00420

TIM00430

TIM00440

TIM00450

TIM00460

TIM00470

TIM00480

TIM00490

TIM00500

TIM00510

FILE XPOS FORTRAN A BELL-NORTHERN RESEARCH

	SUBROUTINE XPOS(N,X,XN)	XP000010
C		XP000020
C	THIS ROUTINE WILL POSITION THE INPUT IDT AT THE FIRST N SAMPLES	XP000030
C	OF THE 512 SAMPLE VECTOR XN(512)	XP000040
C		XP000050
	IMPLICIT REAL*8 (A-H,O-Z)	XP000060
	COMPLEX*16 X(125),XN(512)	XP000070
	COMMON /DATA/ PI,FO,FLAMBD,DELAY,AP1,AP2,NFO,NFI,NDATA	XP000080
C		XP000090
C		XP000100
	DO 100 I = 1,N	XP000110
	XN(I) = X(I)	XP000120
100	CONTINUE	XP000130
C		XP000140
C		XP000150
	NP1 = N+1	XP000160
C		XP000170
	DO 200 I = NP1,NDATA	XP000180
	XN(I) = DCMLX(0.000,0.000)	XP000190
200	CONTINUE	XP000200
C		XP000210
C		XP000220
	RETURN	XP000230
	END	XP000240

END

1 2 0 6 8 6

FIN