Development of An Electrical Impedance Tomograph for Complex Impedance Imaging

Hing Tong Lucullus Leung, B.Sc., AMIEE

A dissertation submitted to the Council for National Academic Awards for the degree of Doctor of Philosophy

The Polytechnic of Wales

Department of Electronics and Information Technology

and

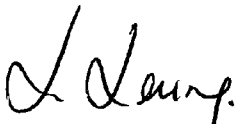
The University Hospital of Wales

Department of Medical Physics and Bio-engineering

August 1991

Declaration

This dissertation has not been, nor is being currently submitted for the award of any other degree or similar qualification.



Hing Tong Lucullus Leung

Acknowledgements

I wish to express my sincere gratitude to Mr. R.J.Williams from the Polytechnic of Wales and Dr. H.Griffiths from the University Hospital of Wales for their encouragement throughout my study, and for their invaluable advice and assistance in the compilation of this dissertation.

A special mention should also be made for all the help I received from the Departmental technicians.

I would like to express my appreciation to the Association of Commonwealth Universities (Overseas Development Administration Shared Scholarship Scheme) and the Polytechnic of Wales for providing the financial support which allowed me to complete this programme.

Finally, I would also like to acknowledge the support and encouragement of my wife, Maggie, throughout the duration of my work.

Abstract

This project concerns the development of electrical impedance tomography towards the production of complex impedance images. The prime intention was to investigate the feasibility of developing suitable instrumentation; but not clinical applications. It was aimed to develop techniques for the performance evaluation of data collection systems.

To achieve this it was necessary to design and develop a multi current source type impedance tomography system, to act as a platform for the current study and for future work. The system developed is capable of producing conductivity and permittivity images. It employs microprocessor based data collection electronics, providing portability between a range of possible host computers.

The development of the system included a study of constant amplitude current source circuits leading to the design and employment of a novel circuit. In order to aid system testing, a surface mount technology resistor-mesh test object was produced. This has been adopted by the EEC Concerted Action on Impedance Tomography (CAIT) programme as the first standard test object. A computer model of the phantom was produced using the industry standard ASTEC3 circuit simulation package. This development allows the theoretical performance of any system topology, at any level of detail, to be established.

The imaging system has been used to produce images from test objects, as well as forearm and lung images on humans. Whilst the conductivity images produced were good, the permittivity *in-vivo* images were noisy, despite good permittivity images from test objects.

A study of the relative merits of multiple and single stimulus type systems was carried out as a result of the discrepancies in the *in-vivo* and test object images. This study involved a comparison of the author's system with that of Griffiths at the University Hospital of Wales. The results showed that the multi current source type system, whilst able to reduce stray capacitance, creates other more significant errors due to circuit matching; future development in semiconductor device technology may help to overcome this difficulty. It was identified that contact impedances together with the effective capacitance between the measurement electrode pairs in four-electrode systems reduces the measurability of changes in phase. A number of benchmarking indices were developed and implemented, both for system characterisation and for practical/theoretical design comparisons.

Contents

Page

1 Introduction 1

2 A Review of the Developments of Electrical Impedance Tomography . 6
2.1 Software Developments 6
2.2 Aspects of Instrumentation Design 9
2.3 EIT Data Collection Systems and Aspects of Instrumentation Designs . 19
2.4 Relative Permittivity Imaging 35

3 An Investigation into Current Sources for EIT 38
3.1 Studies of Current Sources 42
3.1.1 A Three-VOA Current Source Circuit 42
3.1.2 A Transformer-coupled Current Source Circuit 44
3.1.3 A Howland's Equivalent Circuit 44
3.1.4 A VOA Supply Current Sensing Circuit 46
3.2 A Modified Current Source for Use in EIT 47
3.2.1 Computer Simulations 54
3.2.2 Practical Development 57
3.3 Discussion 60

4 Design and Development of the Polytechnic of Wales EIT System . . . 61
4.1 Principle of Operation of the Polytechnic of Wales EIT System 62
4.2 The Data Collection System 64
4.2.1 Digital Electronics Circuitry 65
4.2.2 Analogue Electronics Circuitry 67
4.3 The Host Computer Data Handling and Processing System 81
4.4 Data Collection System Software 91
4.5 System Functional Testings 97
4.6 Possible Improvements to the Data Collection System 101

Page

5 Performance Testings of Data Collection Systems 106
5.1 A Two-dimensional Resistor-mesh Phantom Using SMT 108
5.2 A Computer Model of the Phantom 110
5.2.1 An ASTEC3 Simulator Model 111
5.2.2 Use of the ASTEC3 Model 113
5.3 System Performance Characterisations 114
5.3.1 System Characterisation 114
5.3.2 Comparisons between Practical and Theoretical Data 118

6 Relative Permittivity Imaging 122
6.1 Effect of Contact Impedances on Relative Permittivity Imaging 123
6.2 Difficulties of Multiple Current Source Systems 129
6.3 A Proposed Configuration for Low Crosstalk and Low Feedthrough Analogue Multiplexers and Demultiplexers 133
6.4 Difference between Phase and Permittivity Images 135
6.5 *In-vivo* Imaging 136
6.5.1 Forearm Imaging 138
6.5.2 Lung Imaging 141

7 Discussion 146

Appendices

Appendix A References 154
Appendix B Operating Codes for the Polytechnic of Wales System 158
Appendix C Memory Map for the Data Collection System 159
Appendix D Program Listings of Modula-2 160
Appendix E A Program Listing of 6303 Assembly Languages 195
Appendix F Theoretical Data of the Two-dimensional Resistor-mesh Phantom (SMT) 205
Appendix G Circuit Diagrams of the Polytechnic of Wales System 211

Symbols

σ	electrical conductivity
ε_0	permittivity of free space
ε	relative permittivity
ω	angular frequency
s'	image of the magnitude of complex impedance
s''	image of the phase of complex impedance
α	image of conductivity
β	image of relative permittivity

Abbreviations

A-D	analogue-to-digital
APT	Applied Potential Tomography
ASIC	Application Specific Integrated Circuit
CAIT	Concerted Action on Impedance Tomography
CMFB	common-mode feedback
CMRR	common-mode rejection ratio
D-A	digital-to-analogue
DSP	digital signal processing
E_1	performance index - comparisons based on a uniform mesh
E_2	performance index - comparisons based on a perturbed mesh
E_{re}	performance index - reciprocity error
E_{rn}	performance index - random noise
E_{fde}	performance index - frequency dependent error
EIT	Electrical Impedance Tomography
FEM	finite element model
IEEE EMBS	The Institute of Electrical and Electronics Engineers, Engineering in Medicine and Biology Society
LSB	least significant bit
MSB	most significant bit
OXPACT	Oxford Polytechnic Adaptive Current Tomography
POW	The Polytechnic of Wales
SMT	surface mount technology
UHW	The University Hospital of Wales
VOA	voltage-mode operational amplifer

Chapter 1

Introduction

The aim of this investigation is the design and development of an electrical impedance tomography instrumentation system which is capable of producing images showing the distribution of resistivity (or conductivity) and relative permittivity of objects under test. This also includes the development of criteria by which performance characteristics of such impedance tomography systems may be critically compared. It is not intended to prove the advantages or disadvantages of impedance tomography compared with other clinical imaging methods, or to develop specific clinical investigations. That will be the remit of a future study. It is however intended to study various aspects which would influence *in-vivo* complex impedance imaging.

Electrical Impedance Tomography (EIT) [1,2,3], also called Applied Potential Tomography (APT), is a relatively new form of medical imaging technique. It makes use of the differences of electrical conductivity of human tissues to produce two-dimensional cross-sectional impedance images [4]. Some typical values of electrical conductivity and permittivity for mammalian tissues are shown in Table 1.1.

In order to successfully reconstruct impedance images, comprehensive and systematic sets of measurements are required. The usual measurement method is to place an array of electrodes (typically 16 or 32) around an object to be imaged. Small constant amplitude electrical alternating currents are then applied to a pair of electrodes and the resultant potentials are measured differentially at others. The impedance of the electrode contacts is a complicating issue in this process. However, the use of current stimuli and voltage measurement methods may be used to overcome difficulties associated with these varying contact impedances [5]. A minimum of two sets (frames) of measurements (208 voltage measurements for a sixteen-electrode data collection system) are collected [6]. Due to the application of electrical currents at a pair of electrodes, a pattern of equipotentials is set up within the region of interest. These equipotentials become altered when the electrical field is perturbed by inhomogeneities. An image reconstruction algorithm, e.g. based on a backprojection method [4], may then be used to compute

Tissue	σ	$2\pi f\varepsilon_o\varepsilon$	Ratio	Reference
Liver	0.09	0.059	0.66	Smith and Foster [7]
Kidney	0.20	0.06	0.30	Surowiec *et al.* [8]
Muscle	0.36	0.11	0.31	Surowiec *et al.* [8]
Bone	0.008	0.0013	0.16	Smith and Foster [7]
Spleen	0.20	0.02	0.10	Surowiec *et al.* [8]
Fat	0.033	0.0004	0.012	Smith and Foster [7]

Table 1.1 Typical values of electrical conductivity and permittivity for mammalian tissues obtained at f = 40.96 kHz

the conductivity distribution from these sets of data.

The spatial resolution (i.e. in the plane of electrodes) of impedance images is not as good as images produced by more traditional imaging techniques such as X-ray computed tomography. The resolution is however adequate for many clinical applications since it is not always necessary to display the organs in great detail for all diagnoses. Only a few papers regarding clinical applications have been published. Barber and Brown [4] demonstrated that it is possible to show the changes in resistivity in the lungs between inspiration and expiration. The images produced clearly showed the lungs and fluid within a lung. Mangnall *et al.* [9] showed EIT as a method of monitoring gastric emptying. As a clinical imaging system, impedance tomography has the potential advantages of being relatively low cost, non-invasive, and does not employ ionising radiation.

So far, the reported EIT data collection systems have been concerned with the imaging of conductivity only. This is actually because the conduction currents are the dominant components within the EIT frequency range, i.e. usually from 10 kHz to 100 kHz. From the studies conducted by Griffiths *et al.* [10], it has been evident that the ratios of permittivity ($2\pi f\varepsilon_o\varepsilon$) to conductivity ($\sigma$) of some human tissues for a given frequency are small (Table 1.1). Griffiths [11] has managed to demonstrate the possibility of displaying relative permittivity images by means of computer simulation studies and practical measurements. The intention of this programme is to continue Griffiths's preliminary work and to develop relative permittivity imaging to the point where measurements on human subjects will be possible.

In order to achieve this it was recognised that first it would be necessary to design and develop an EIT data collection system which would be suitable to assess the feasibility of relative permittivity imaging. A number of objectives were therefore projected. They were

(1) to review EIT system development

Considerable studies of the various instrumentation aspects were essential prior to the design and construction of such a system. Important aspects which would influence the system performance needed to be identified at an early stage of the programme. It was intended that a literature review would be maintained throughout the programme of work. A listing of the most important literature is appended to this thesis. A discussion of developments in EIT techniques is provided in chapter two. The areas of interest principally concerned the architecture of various data collection systems. An important aspect was, initially, to identify the best approach for the proposed system architecture. The options were for current or voltage stimuli; and serial or parallel methods for stimuli and measurements.

(2) to investigate current source circuits

In view of the magnitude of tissue impedances and the relative size, and variability, of electrode contact impedances, it was clear that high quality current sources would be essential. Thus, an intensive investigation into different current source circuits was projected. Various circuits presently being used by other centres would be reviewed; these are discussed in chapter three. Rigorous circuit analyses were planned to be conducted on each circuit. In addition to the normal performance analyses, feasibility studies of the production of multiple current sources with consistent performances would also be required since the implementation of parallel drive mode needs a number of identical circuits.

(3) to design and develop a complete impedance imaging system

Having undertaken the preliminary preparations, the design and construction of an impedance imaging system commenced. This work is detailed in chapter

four. It was proposed that the complete imaging system would be partitioned into two major functional systems, namely a Host Computer Data Handling and Processing System, and a Data Collection System. It was recognised that the former would require software developments concerning data communication protocols, imaging reconstruction algorithms and system performance characterisations. Due to the necessity of relatively large data transmissions, the employment of data transmission checking schemes was considered essential. These aspects were recognised to require both software and hardware developments. It was anticipated that the proposed system would need a degree of intelligence so that it would be flexible enough to allow for various types of investigations, and to be able to communicate with a host computer. This pointed to a microprocessor based system design. By introducing a microprocessor into the system, easy alteration of the system's operations or enhancement of its ability would result. This would provide considerable advantages as most of the electronics would be software controlled.

In addition to the construction work, some functional testing was planned in order to prove the integrity of the system. It was also intended to evaluate the relative advantages and disadvantages of multiple drive and multiplexed single current source systems.

(4) to develop methods for the assessment of system performance

Apart from the development of methods for functional testing, system performance characterisation was considered to be important; since a variety of circuit designs was proposed in order to address the problems exhibited in EIT. Thus, some methods for benchmarking system performances were considered to be essential. Chapter five shows a number of methods investigated to address this objective.

In order to benchmark EIT systems, an electrically stable test object (phantom) was required. In consideration of phantoms available at the time, a two-dimensional resistor-mesh originally designed by Griffiths [12] seemed to be the most appropriate hardware phantom. Further investigations into this phantom such as manufacturability were proposed. Additionally, the need for a software model having the same configuration as the hardware item was

deemed to be equally important, since this could provide a set of noise free data representing the ideal situation. It was intended to extend the phantom development to include this aspect.

(5) to study various aspects concerning complex impedance imaging

This would be a novel investigation in the author's studies. Details are documented in chapter six. As previously mentioned, a small number of permittivity images had been obtained by means of computer simulations [13] and from measurements on a resistor/impedance-mesh test object [12]. This section of work was intended to perform investigations into various aspects of hardware electronics development relating to permittivity imaging. *In-vivo* complex impedance imaging would be performed on a human subject in order to confirm the validity of electronics, and thereby to study the possibility of imaging complex impedances *in-vivo*.

(6) to produce a comprehensive report on the conduct and findings of the investigation

Chapter 2

A Review of the Developments of Electrical Impedance Tomography

In 1980, Bates *et al.* [14] reported the problem of reconstructing a unique two-dimensional impedance image from peripheral measurements. The technique was based on the concept of X-ray computed tomography. The reconstructed image was a representation of the distribution of electrical conductivity within the region of interest. It was pointed out that the stimuli (electrical currents) could not be forced to penetrate the object in a straight line, unlike X-rays. This situation creates ambiguities and makes a unique image reconstruction impossible. However, it was also reported that using extensive sets of measurements might provide a means to resolve the ambiguities.

Three years later, Brown, Barber and Freeston applied this type of tomography to develop a new medical imaging technique - electrical impedance tomography - to which a patent was issued in 1983 [1]. They reported in the same year that the evaluation of the distribution of conductivity was a three-dimensional problem and hence posed extra problems for reconstruction since the applied stimuli could not be constrained to flow in the plane of the surface electrodes [15]. Since then, a small number of research groups, internationally, has been undertaking various investigations. These have concerned software aspects such as studies of image reconstruction algorithms and hardware aspects such as the development of data collection systems. From the reported investigations, it appears that the majority of research groups originate in the European countries. Outside of Europe, several groups in U.S.A. have also engaged in investigations of impedance tomography, with a small number of papers reported from Japan [16] and People's Republic of China [17]. The latter principally concern algorithms and are thus not dealt with here. The first book devoted to impedance tomography is attributed to Webster [18]. The following review will mainly focus on recent developments in system hardware. A brief review of reconstruction algorithms potentially relevant to this study is also included.

2.1 Software Developments

In impedance tomography, software aspects mainly concern the investigations into

image reconstruction algorithms. To develop a reconstruction algorithm it often involves studies of the 'forward' and 'inverse' problems [2]. The former refers to the problem of predicting the distribution of potentials within a medium to which a known stimulus is applied; and hence the distribution of electrical conductivity. The latter refers to the problem of evaluating the distribution of electrical conductivity from sets of measured peripheral potentials as a result of applications of known current patterns at the boundary of an object under test. The solution of the 'forward' problem provides sets of theoretical data which are useful for the studies of the 'inverse' problem. However, methods of solving the 'forward' problem are not discussed as they are not an area of interest in the author's study. The 'reverse' problem may concern two- and three-dimensional reconstruction processes. Several three-dimensional reconstruction algorithms have been developed and details of these can be obtained from the work conducted by Kim *et al.* [19], Goble *et al.* [20], Liu *et al.* [21] and Zyia Ider *et al.* [22]. These are not further discussed because the two-dimensional reconstruction algorithms have been found to be adequate in the three-dimensional situation [2,23] and in this context of the author's work.

Generally, tackling the 'reverse' problem relates to the solution of Equation 2.1,

$$b = Tr \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

where b is a vector representing boundary measurements for a given set of drive configurations, r is a vector representing an object resistivity distribution, and T is a forward transformation. Thus, the image reconstruction problem leads to evaluation of the inverse transformation of Equation 2.1,

$$r = T^{-1}b \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.2)$$

Since T is a non-linear transformation, the evaluation of the inverse of T will be extremely difficult. To evaluate the resistivity distribution there are at least four types of algorithms available. They are *the perturbation method, the double constraint method, the modified Newton-Raphson method* and *the equipotential lines method.* A review of these methods and comparisons between them was

conducted by Yorkey *et al.* [24,25].

The perturbation method was proposed by Kim *et al.* [26]. The method is used to calculate current changes resulting from resistivity changes. They proposed that the perturbation matrix, which had a linear relationship between changes in current and resistivity, was only required to be evaluated once in the iteration process. The advantage of this method is fewer restrictions on *a priori* knowledge of the resistivity distribution. Yorkey *et al.* [24] showed that the computation time could be reduced substantially with a correct initial guess of the location of the object and its approximate resistivity. Even with a completely wrong initial guess, an identical image can be reconstructed at the expense of an increase in number of iterations in the reconstruction process.

The double constraint method was proposed by Wexler *et al.* [27] and named by Yorkey *et al.* [24]. It uses a finite element model (FEM) as a vehicle to evaluate the desired resistivity distribution. The method requires first to constrain the FEM with the known current source values for evaluations of the current density in the computer model elements and second to constrain the FEM with the observed voltages and the current source values for evaluations of the voltage gradient in each element. The resistivity of each element can then be computed based on these evaluations. A shorter time for each iteration is the advantage of this method. However, it produces bigger errors in comparison to other methods after the same number of iterations.

The modified Newton-Raphson method was developed by Yorkey *et al.* [24]. They employed a four-electrode method [28,29] to reduce measurement errors and a FEM to form an area of interest. Using a current source, a homogeneous relation between the resistivity distribution and the voltage measurements can be established. They mentioned that a similar relation was not true with conductivity as conductivity is inversely related to resistivity. The resistivity within each element in their FEM was assumed to be constant. With all these assumptions, the resistivity distribution can be evaluated. Yorkey *et al.* also showed that the modified Newton-Raphson method has the merit of fast convergence with the least error but has a demerit of sensitivity of measurement error.

The equipotential lines method proposed by Barber *et al.* [2,5] provides a means of reconstructing an object image from the measured potential profiles. A linear relationship between the resistance perturbations and the perturbed voltage

distribution is assumed. With the first order approximation, the ratios of the corresponding measurement voltages are backprojected along the equipotentials. Having completed the whole backprojection process, it is necessary to undergo a filtering process in order to smooth the image. In 1987, Barber and Seagar reported an algorithm for fast reconstruction of resistance images [23] based on an assumption that the required resistivity distribution is close to a known reference distribution so that reconstruction problems can be dealt with linearly but approximately. What is important about this assumption is that a single-pass backprojection method may be employed.

To summarise, Table 2.1 shows the characteristics of the four types of reconstruction methods. Note that all data were quoted from Yorkey *et al.* [24]. The errors shown in Table 2.1 were defined as the squared inner-product of the difference between the true resistivity profile and the estimate, and normalised by the squared inner-product between the true resistivity and its average value [24]. Each reconstruction process was started with the same initial value which was the background resistivity value.

	Perturbation	Double Constraint	Modified NR	Equipotential Line
normalised error after each iteration	0.8	1.03	0.3	0.78
normalised time taken for each iteration	19.8	2.5	18	1
No. of iterations required for error less than 0.01	No Convergence*	No Convergence*	4	Not Applicable

Table 2.1 Characteristics of the four reconstruction algorithms
*: Results after 20 iterations

2.2 Aspects of Instrumentation Design

When considering the design of a data collection system, strategies for the application of stimuli and the measurements of peripheral signals are the important topics to be discussed. These two aspects determine the configuration of the data collection system and also its operation modes, i.e. dynamic or static imaging. Dynamic imaging produces images representing changes in electrical conductivity with time. Static imaging produces images representing the anatomical structure of test objects. Static images are produced by evaluating the difference of absolute

conductivities obtained from two separate measurements.

The applied stimuli can be either voltages or currents. Barber *et al.* [1,4,5] have reported that using currents as the stimuli has the advantage of reducing the contact impedance problem to some extent particularly when measurements are not allowed immediately adjacent to the drive electrode pair, i.e. a four-electrode method. Additionally, a sink/source driving action is usually adopted in the four-electrode method (Figure 2.1(a)), although other drive methods (Figure 2.1(b)) such as drive and ground may be used. If the problems of stray capacitances in the demultiplexers could be tackled properly, the sink/source drive method may provide a better electric field pattern. Due to the presence of stray capacitance as shown in Figures 2.1(a) and (b), a current leakage occurs resulting in relatively serious deterioration of the field pattern in the latter case.

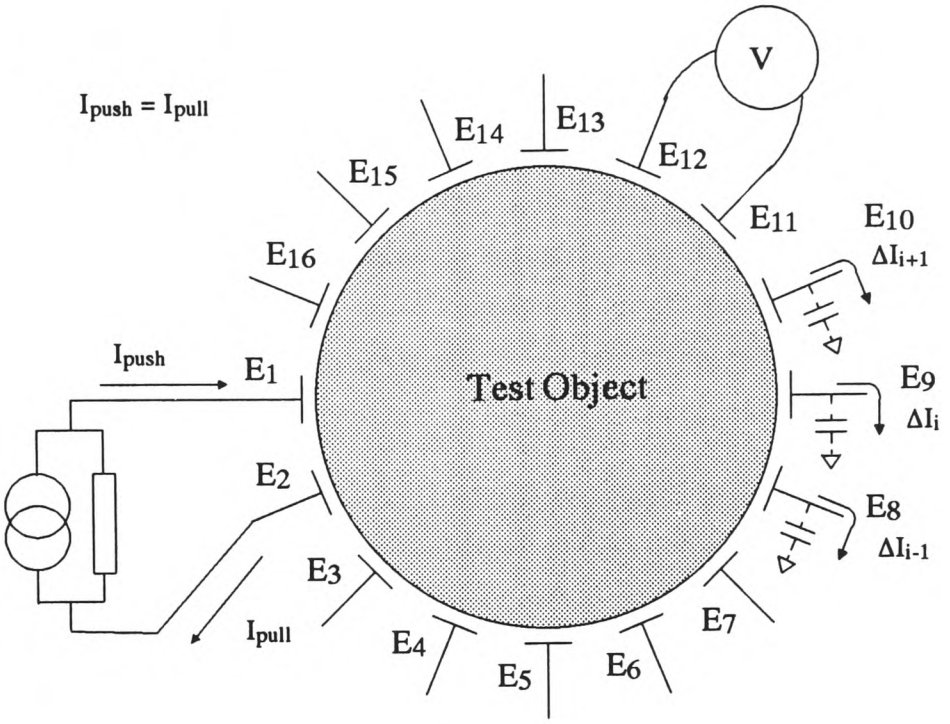
Voltage drive methods suffer acutely from electrode impedance problems. In *in-vivo* imaging, slight movements of a patient may give rise to a large change in contact impedance. An impedance variation of a few kΩ may result. In the presence of contact impedance, the true voltage signals to be applied to the periphery of the object would not be known and thus caused errors.

Basically, there are three aspects which need to be considered in order to determine the architectures of the data collection systems.

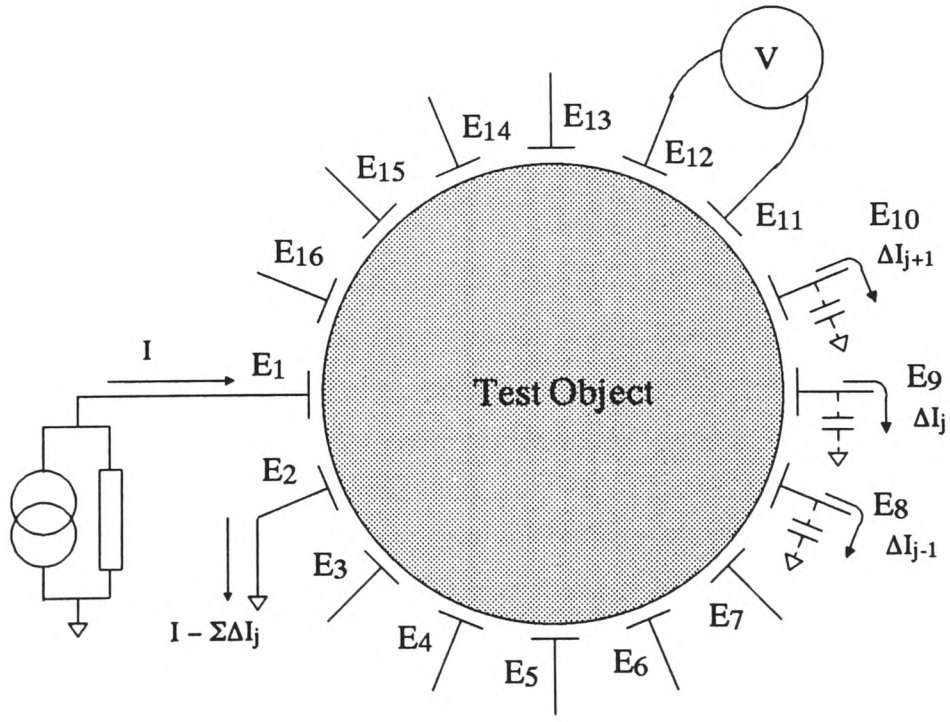
(a) Image quality

The image quality is generally governed by two quantities, namely spatial resolution and pixel resolution. The former represents the geometric resolution in the x-y plane (i.e. in the plane of surface electrodes) while the latter represents the resolution of the images parameter in the z plane. The spatial resolution of reconstructed images is determined by the number of independent pixels [3]. Clearly, the number of independent pixels is governed by the number of independent measurements which depends on the number of electrodes to be employed. Barber *et al.* [2,5] showed the relationship between the number of electrodes and the number of independent measurements. If N is the number of electrodes of a particular data collection system which employs the adjacent drive method and equally spaced electrodes configuration, a total number of possible measurements will be N(N–1).

(a) push-pull configuration



(b) push-ground configuration



ΔI denotes as leakage current, and

$\Delta I_i << \Delta I_j$

Figure 2.1 Possible adjacent drive configurations for a four-electrode method

However, due to the reciprocity between the drive and measurement electrode pairs, the maximum independent measurements reduces to N(N−1)/2 [2]. For instance, a 16-electrode data collection system can obtain a maximum of 120 independent measurements, but due to high electrode contact impedances no measurements are made adjacent to the drive pair. Thus, the independent measurements further reduce to 104, i.e. N(N−3)/2. In practice, 208 measurements are taken in order to improve the signal-to-noise ratio. Measurements performed with non-adjacent electrode pairs do not produce extra information as they can be obtained by superposition of the adjacent drive electrode profiles [2].

The pixel resolution is exhibited by the number of grey levels, usually sixteen grey levels are used. The quality of each pixel (defined as the accuracy of the representation of the actual resistivity value exhibited in the pixel) is determined partly by the system's accuracy and partly by the reconstruction algorithms. Brown and Seagar [6] reported that 5% changes in peripheral voltages in the lungs between maximum expiration and inspiration were detected with their data collection system. Patterson [30] also showed that changes in resistivity of the lungs was approximately 3% on cardiac systole. By correlating two sets of measured voltages, the maximum resistivity changes are approximately 0.15%. In order to be able to resolve these changes, peripheral measurements require to attain a signal-to-noise ratio of better than 0.1%, i.e. 60 dB.

An increase in the number of electrodes can improve the resolution of reconstructed images but has the drawback of reduction of the signal-to-noise ratio because the signal level also reduces dramatically by a factor of n^2 [2], where n is the ratio by which the number of electrodes is increased. According to the figures shown by Barber and Brown [2], the smallest signal of approximately 80 μV is obtained from a 32-electrode system as a result of applications of 1 mA current to the human arm. For a 128-electrode system, a signal of 5 μV can be expected. Thus, the spatial resolution and the pixel resolution of a system must be compromised.

(b) Data collection speed

This is the speed for a system to collect one frame of data, i.e. 208 differential

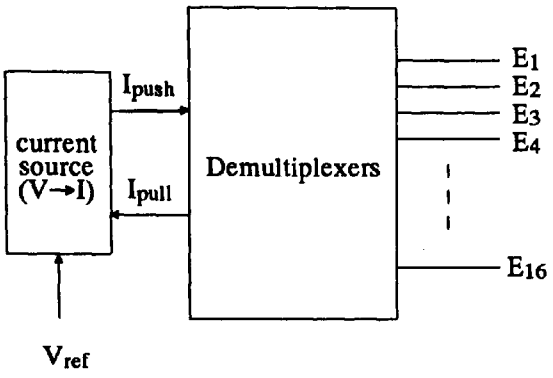
voltage measurements assuming a 16-electrode data collection system. This aspect determines the possible imaging modes viz static or dynamic imaging. Two major factors which influence the data collection speed are the architecture of voltage measurements and the methods of demodulation of measured voltages.

(i) Effect of architecture of voltage measurements

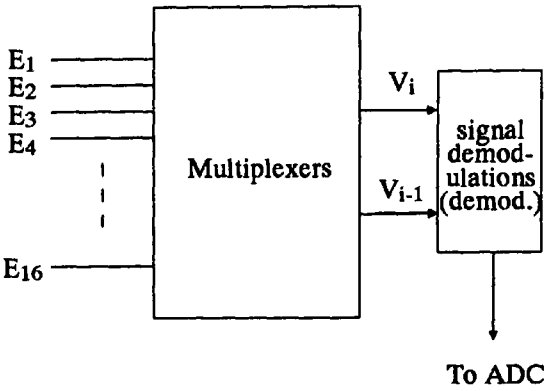
There are four basic configurations for the application of applied stimuli and the measurements of peripheral signals. These are shown in Figure 2.2. The simplest front-end architecture of a data collection system is the combination of two serial configurations, i.e. combining Figures 2.2(a) and (c). It involves one current source and one phase-sensitive detector for signal demodulation. The use of phase-sensitive detectors is to extract the real and imaginary parts of the measured peripheral voltage signals. Multiplexers and demultiplexers are used for switching drive and measurement electrode pairs. This configuration can allow both static and dynamic imaging but may not be suitable for "real-time" imaging due to limits of data acquisition speed. Imaging at real time requires a high data collection speed, e.g. at a rate approaching 25 frames per second, so parallel measurement methods are essential (Figure 2.2(d)). In addition to the high speed data collection capability, some means for fast image reconstruction, as well as rapid display of images, are essential. Figure 2.2(b) shows the configuration for the parallel application of stimuli. It cannot shorten the data collection time but provides a more flexible way to apply current patterns which may be attractive in future systems.

Each of the configurations mentioned above has its own advantages and disadvantages. The serial method of stimulus approach illustrated in Figure 2.2(a) needs only one current source but requires demultiplexers. It has the advantage that no current sources have to be matched in order to produce a sink/source drive pair. But, the employment of the demultiplexer gives rise to problems concerning stray capacitance effects within the demultiplexer device itself. The situation is aggravated when a system is intended to operate at different frequencies. This is because, in addition to the stray capacitance effects due to system layout, performances of semiconductor devices vary with frequency due to their inherent parasitic capacitance. The parallel method

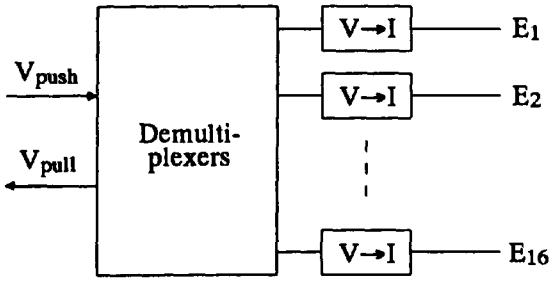
(a) serial drive



(c) serial measurement



(b) parallel drive



(d) parallel measurement

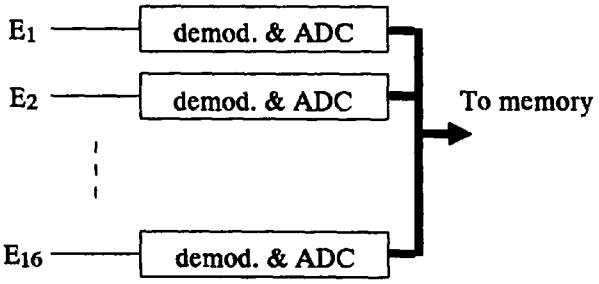


Figure 2.2 Possible drive and measurement configurations

illustrated in Figure 2.2(b) can reduce the problems of stray capacitance to a minimum but the problems of matching current sources becomes evident. This matching problem becomes very difficult when the number of electrodes increases. The characteristics of the semiconductor devices employed will inevitably exhibit deviations, which makes perfect matching impossible. Figure 2.3 shows that mismatched current sources will cause a small current to flow (possibly down the reference electrode) in the object, and hence to produce a common-mode error signal. In addition, employing multiple current sources may also produce large systematic errors such as reciprocity errors. Despite these problems, however, the method offers the maximum flexibility for driving any type of current patterns. The adaptive current drive configuration is one such application [31].

Figures 2.2(c) and (d) show the two possible configurations for the measurements of peripheral signals. The serial configuration requires less electronics. An analogue multiplexer is used in order to select the measurement electrode pairs. The effect of stray capacitance is not as critical as in the previous case because all peripheral voltage signals can be buffered. However, a longer data collection time is a disadvantage. The parallel configuration shortens the data collection time at the expense of the demand of a large amount of electronics. Again, large reciprocity errors may result due to the variations in the characteristics of the semiconductor devices.

(ii) Effects of methods of signal demodulation

Demodulation can be achieved by means of analogue or digital methods. Analogue demodulation usually employs analogue multipliers and low-pass filters to obtain the real (in-phase) and imaginary (quadrature) parts of measured voltage signals. Thus, the collection time is principally governed by the settling time required by the low-pass filters. The higher the break frequency, the shorter the setting time is, but the poorer the signal-to-noise ratio would be. So, the use of optimum break frequency filters is desirable in this application. An intensive investigation into the optimum break frequency was conducted by the author and the work was presented in chapter four.

Digital demodulation usually does not require low-pass filtering, instead sample-and-hold circuits are used. The aperture time of these circuits may be

$$I_{source} = I_{sink} + I_{imbalance}$$

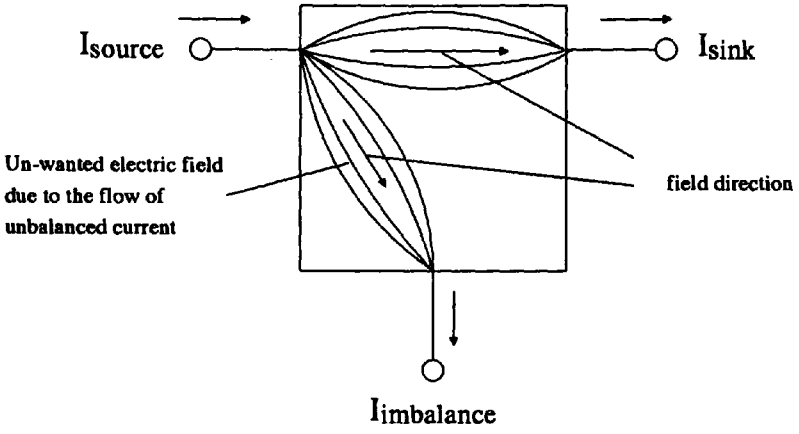


Figure 2.3 Effect of unbalanced current in EIT

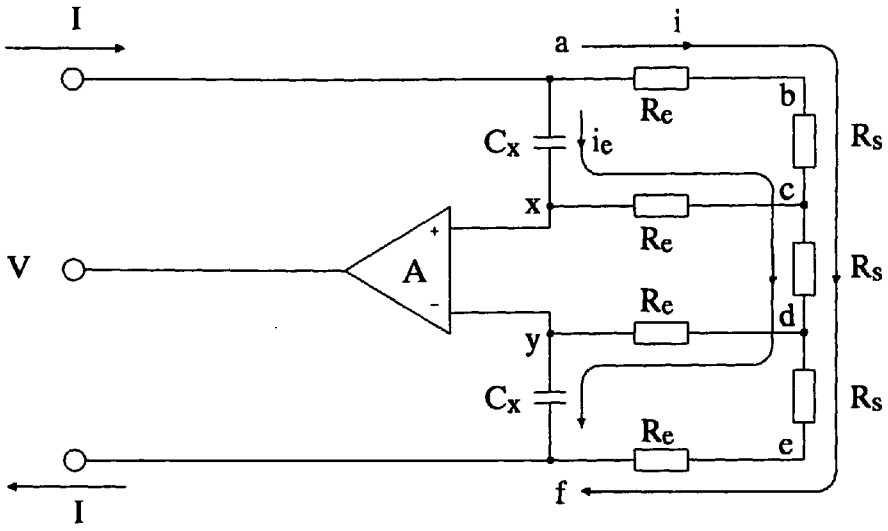
a cause of problems if the device is not carefully selected. The multiplication processes are performed digitally. A comparatively faster data collection speed can therefore be obtained, but high quality analogue-to-digital (A-D) conversions are essential.

(c) Image parameters

The peripheral voltages resulting from the application of known current patterns contain useful information concerning the resistivity and permittivity distributions within the object. The spatial distributions of these two electrical quantities can be evaluated by the measurements of the real and imaginary parts of the peripheral voltage signals respectively. Either of these two parameters may be used to create images. When employing the serial drive mode, systematic errors, due to the presence of stray capacitances, are dominant in the reactive parts of the measured voltages. These must be separated out even when only conductivity images are of interest. This can be shown in a four-electrode measurement system such as that illustrated in Figure 2.4 of which the object to be measured is a homogeneous medium of a resistivity R_S, e.g. a saline solution. In the presence of stray capacitances C_X between the electrodes, two major current paths can be identified. One is the desired current path along a,b,c,d,e,f. The other is the undesired current path along a,x,c,d,y,f. For the sake of simplicity, several assumptions are made. Firstly, all electrode resistances (R_e) are equal. Secondly, a perfect sink/source current pair (I) is used. Thirdly, the instrumentation amplifier has an infinite input impedance. Hence, an expression representing the voltage developed across x and y can be derived.

$$V_{xy} = I\left[\frac{2R_e(R_e+R_S)}{2R_e+R_S+(1/j\omega C_X)}\right] + IR_S \quad \ldots\ldots\ldots\ldots (2.3)$$

The measured voltage V_{xy} is clearly not the required signal which should be IR_S. For a typical stray capacitance C_X = 10 pF, an electrical contact resistance R_e = 1 kΩ and a load resistance R_S = 100 Ω, the unwanted resistive and reactive components at 100 kHz are calculated to be 0.18% and 13.8% of IR_S respectively. Thus, in order to measure the apparent changes in electrical



i : desired current path
ie : undesired current path

Figure 2.4 A four-electrode system

permittivity, the stray capacitance effect must be kept to a minimum. Parallel drive topologies may be appropriate in this application. If a system is intended to operate at different frequencies, the capability of producing a flat frequency response within the operating frequency range would be essential. Otherwise, large systematic errors would be produced due to variations in phase as well as magnitude at different frequencies.

2.3 EIT Data Collection Systems and Aspects of Instrumentation Designs

Internationally, several EIT data collection systems have been and are being developed and reported. A well organised EEC workshop specifically on hardware for EIT development was held in Leuven in December 1990. Some other systems being developed outside the EEC countries were also found. All these systems are reviewed. The architecture of each system will be discussed if available, and their features will be outlined.

(a) U.K. - Bristol group

The Bristol group reported a 16-electrode data collection system which is designed specially for breast imaging [32]. Serial stimuli application and data collection methods are adopted. A transformer-coupled current source capable of delivering 1 mA at 90 kHz alternating current is used as the applied stimulus circuit. Peripheral voltage measurements are taken with aid of a monolithic instrumentation amplifier type AD524. The selection of drive and measurement electrode pairs is achieved using analogue multiplexers. The signal demodulation process takes place in a RMS-to-DC converter AD536. All devices mentioned above are housed in a circular ring which encompasses the human breast. The demodulated voltages are then digitised and stored in a computer memory prior to reconstruction.

The fixed geometry of electrodes creates limitations in practice due to variations in breast sizes, but have the advantages that front-end electronics are close to electrodes resulting in smaller stray capacitance effects due to short cables. The employment of AD536 limited the accuracy since the accuracy of conversions was 0.5% (as quoted from Analogue Devices data sheet). No images have yet been reported.

(b) U.K. - Cardiff group

The Cardiff group is concentrating on the dual-frequency imaging technique proposed by Griffiths and Ahmed in 1987 [13]. The idea was verified by means of computer simulation as well as practical measurements [33]. The principle of dual-frequency technique is to produce conductivity images by making measurements at two distinct frequencies, say 40 kHz and 80 kHz, since the conductivity varies with frequency [34]. By comparing the changes, conductivity images can be reconstructed. This group also performed some studies on relative permittivity imaging [11,12].

Their system is under the control of a BBC micro-computer. The data collection system employs sixteen electrodes and is capable of operating at four distinct frequencies (10.24 kHz, 20.48 kHz, 40.96 kHz and 81.92 kHz). These frequencies are generated from a digital based sinusoidal voltage oscillator.

Serial methods for both the stimuli applications and the voltage measurements have been adopted. The adjacent drive and voltage measurement method is used. The selection of drive and measurement electrode pairs is achieved by two pairs of analogue multiplexers. A transformer-coupled current source circuit is used as the stimulus device. All electrodes are connected to the system by means of coaxial cables with driven screens in order to minimise the stray capacitance effect. The screen driver circuit is a voltage follower which is also used to sense the surface potentials. The surface potentials between adjacent electrodes are measured by means of a discrete instrumentation amplifier using the three operational amplifiers topology. The differential voltage signals then undergo a series of attenuation and amplification processes as well as band-pass filtering before the demodulation process takes place. Because of the necessity of multiple frequencies measurements, four different band-pass filters of different centre frequencies are required. A conventional voltage demodulation method is implemented by multiplying the sensing voltage signals with a reference signal. The resultants are low-pass filtered prior to the A-D conversions which take place in the BBC computer. Only the real part of the signals is used for image reconstructions.

Due to the employment of the serial data collection method and the slow data conversion time offered by the BBC computer, the time taken for collecting a complete image data sets requires 8 seconds. The feature of this system is to

produce static images without involving any homogeneous references. In order to meet the safety requirements [35], an isolation power supply is used for the patient circuitry. Isolation of analogue and digital signals is achieved by means of isolation transformers and opto-isolators respectively. Some conductivity images obtained with this dual-frequency technique were reported [33].

(c) U.K. - Keele group

Not all of the front-end circuitry has been reported. The overall system architecture was not found in the author's literature survey. They suggested that better system performances in terms of quality of measured data could be achieved when current sources and voltage detectors were mounted directly behind the electrodes [36]. This indicated that multiple current sources and voltage detectors were required. This method involves no multiplexing and hence alleviates the capacitance effects.

At the Leuven meeting, Record [37] pointed out that the input impedance of an ac-coupled voltage follower (also called a bootstrap buffer) as illustrated in Figure 2.5 is frequency dependent. Similar cautionary comments were also given by Lidgey [37]. With reference to Figure 2.3, the equivalent impedance Z_{in}' (defined as an impedance looking into the operational amplifier at point A) can be written as

$$Z_{in}' = R_1+R_2+j(\omega C_2R_1R_2-1/(\omega C_1)) \quad \ldots\ldots\ldots\ldots (2.4)$$

Suppose that the cable is not screen driven (i.e. switch at position '0'). The input impedance Z_{in} of the circuit can then be expressed as

$$Z_{in} = R_e + \frac{R_1+R_2+j(\omega C_2R_1R_2-1/(\omega C_1))}{1-\omega C_x(\omega C_2R_1R_2-1/(\omega C_1))+j\omega C_x(R_1+R_2)} \quad \ldots (2.5)$$

For values of $R_1 = R_2 = 100\ k\Omega$, $C_1 = 1\ \mu F$, $C_2 = 10\ nF$, $C_x = 100\ pF$ and $R_e = 1\ k\Omega$, the equivalent input impedance of the circuit was obtained by means

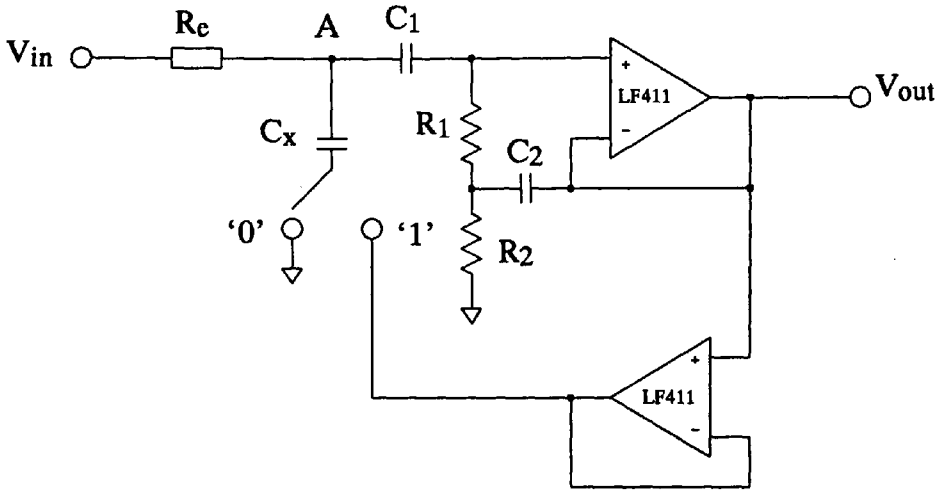
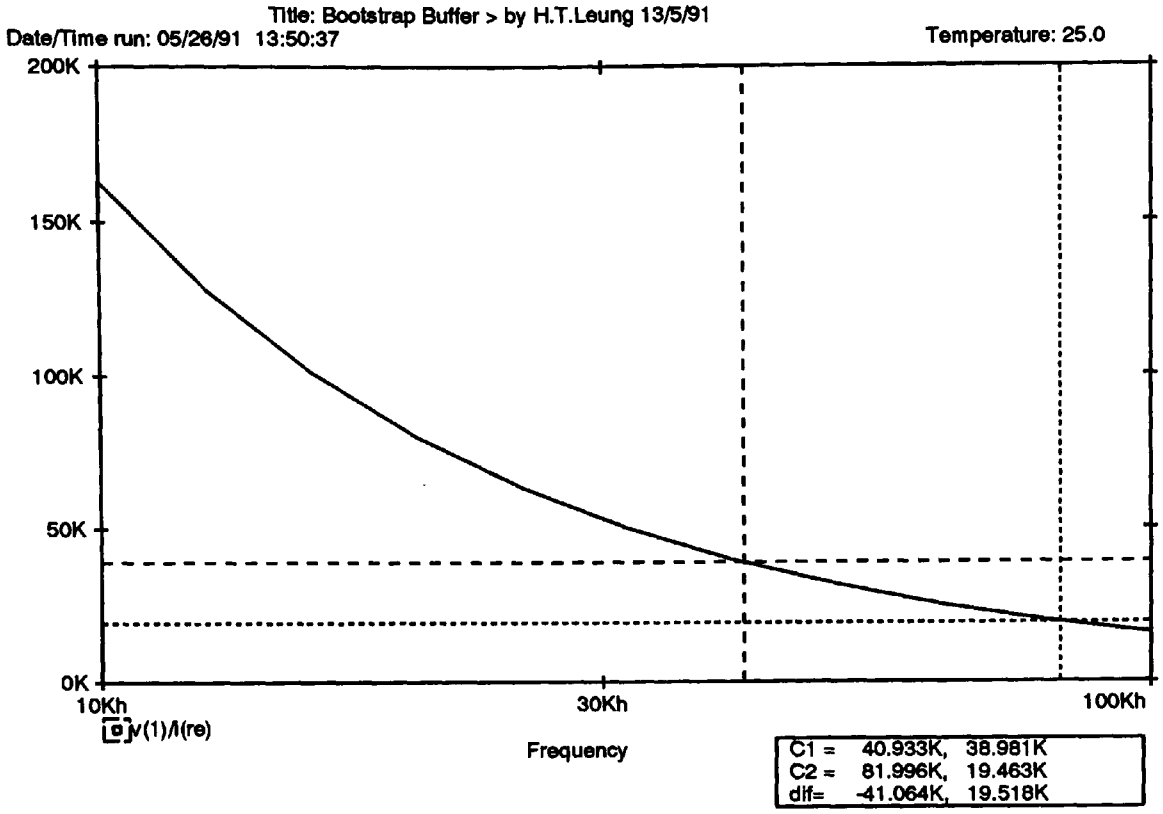


Figure 2.5 A bootstrap buffer

(a) Without driven screen



(b) With driven screen

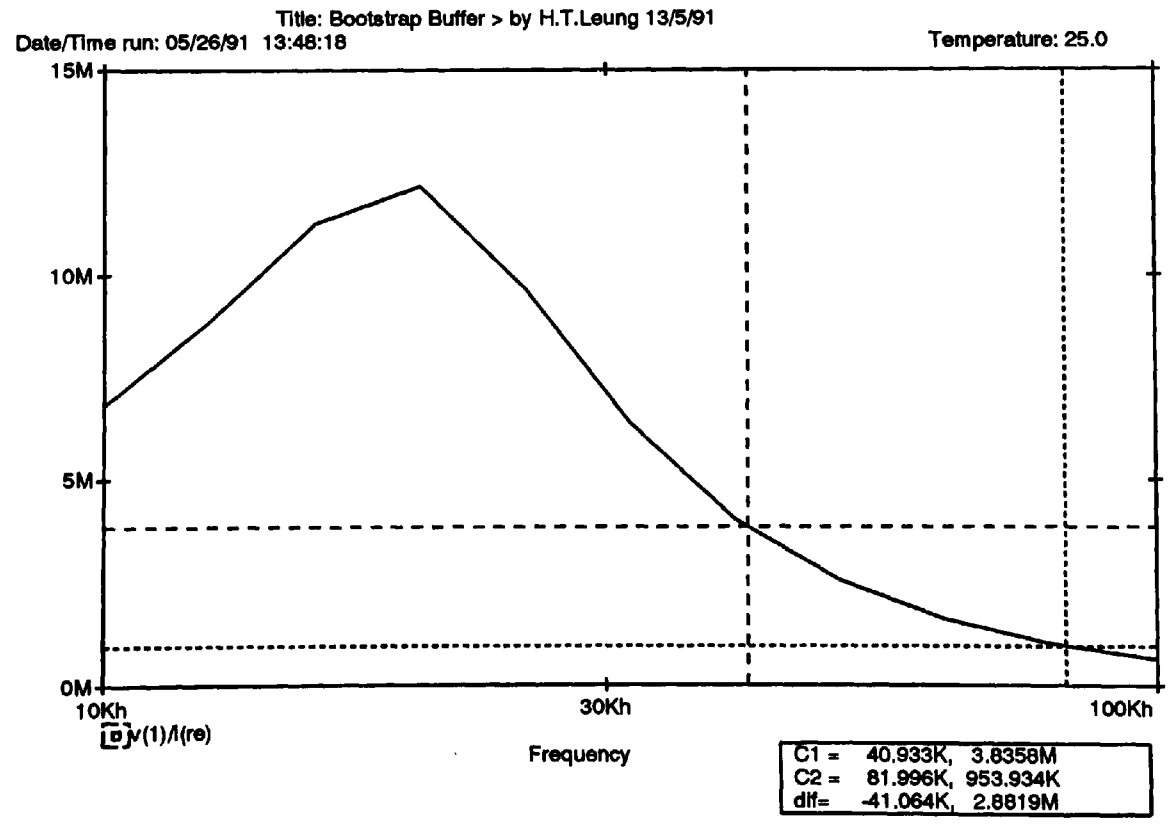


Figure 2.6 Performances of a bootstrap buffer
x-axis: frequency; y-axis: input impedance

of computer simulation. This work was conducted by the author and the PSpice analogue circuit simulator was used. Figure 2.6 shows the results at which the plots indicate the variations in input impedance with frequency. The input impedance of the buffer was found to be approximately 40 kΩ at 41 kHz and 19.5 kΩ at 81 kHz, Figure 2.6(a). This is within the EIT frequency band. However, with a screen driven cable (i.e. switch at position '1'), the input impedance increased significantly. Figure 2.6(b) shows 3.8 MΩ and 950 kΩ resulted at 41 kHz and 81 kHz respectively. This indicated that the driven screen mode is essential in EIT.

A modified voltage buffer was presented at the Leuven meeting [37]. The circuit was intended to introduce a negative capacitance to cancel the stray capacitance. The modified circuit performed well under an ideal situation where stray capacitances can be perfectly cancelled. However, it exhibits difficulty in reality as negative capacitances are difficult to obtain, and would introduce instability.

(d) U.K. - Oxford group

The Oxford group has endeavoured to exploit adaptive current tomography since this method of imaging would increase ability to distinguish inhomogeneities located far from the surface electrodes [31,38]. Their data collection system was named as Oxford Polytechnic Adaptive Current Tomography (OXPACT). According to Murphy's report [39], it is a 32-electrode imaging system and is expandable up to 96 electrodes. In this first reported version of OXPACT, a parallel stimuli applications configuration was employed. The system is a PC based unit. Each electrode has a dedicated single-ended current source which may be controlled independently by the PC, and has a driven screen. The current source circuit employed is made up of a complementary current mirrors pair and adopts the voltage operational amplifier (VOA) supply-current sensing technique to produce a constant amplitude alternating current. Details of this will be discussed in the next chapter.

The system was intended to operate at one frequency. 31 electrodes attempt to impose an optimum current pattern in an object to be imaged while the 32nd electrode is usually grounded to provide a path for residual current. Each

electrode is connected to an ac-coupled voltage follower which is intended to provide high input-impedance buffering to sense the surface potentials. It is important to have a high input impedance buffer as currents may leak through these buffers and the field pattern would therefore be disturbed. These surface potentials can be selected sequentially by means of a multiplexer network which is implemented with 8- and 16-channel analogue multiplexers. The selected signals are compared using an integrated instrumentation amplifier, AD524. The differential signals require sufficient amplification prior to the demodulation processes, otherwise, the signal-to-noise ratio would be poor. Demodulations are achieved with switch-mode demodulators. The demodulated signals are then low-passed filtered and digitised with a 12-bit A-D converter. Both real and imaginary parts of the peripheral voltages are measured. However, they have no intention to utilise the imaginary part for the reconstruction of relative permittivity images.

In June 1990, the progress on the second version of OXPACT was presented at the EEC Workshop in Copenhagen [40]. The new version features a battery powered PC and high data acquisition speed. But the serial data collection method is still employed. No images have yet been reported as the system is still in the development stage.

In addition to the development of the data collection system, the Oxford group also engages in investigations of hardware aspects which may have significant influence on front-end electronics design. For example, studies on circuit topologies of current drivers and instrumentation amplifiers have been executed. The discussion on the current source circuits will be continued in the next chapter while the most up-to-date development in the instrumentation amplifier is documented below.

A novel instrumentation amplifier was presented in the EEC Workshop held in Leuven in December 1990 [37]. The whole circuit comprises five operational amplifiers which is an improved version of the current-mode instrumentation amplifier previously developed by Toumazou and Lidgey [41,42]. The improved version is shown in Figure 2.7. The alteration is the replacement of the current mirrors with two additional operational amplifiers. Both amplifiers act as trans-resistance devices. The merit is that there are no current mirrors to be matched which is a well-known difficulty of using current mirrors. More discussions on current mirrors are given in the next chapter.

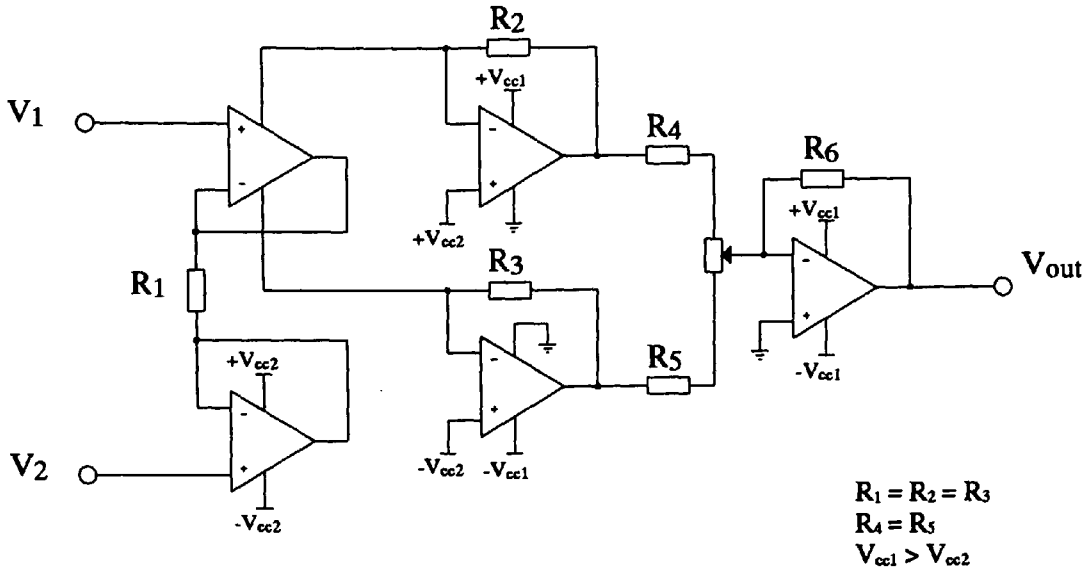


Figure 2.7 An improved instrumentation amplifier

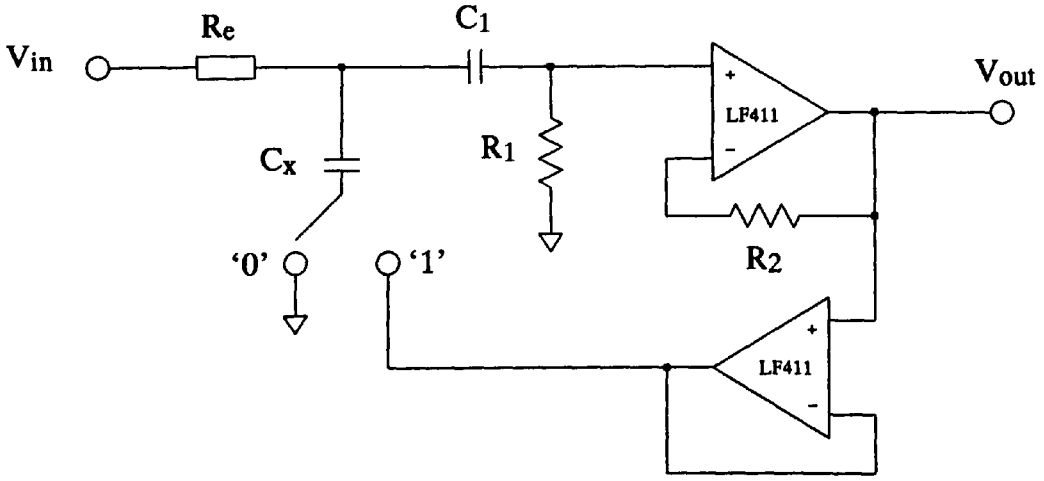
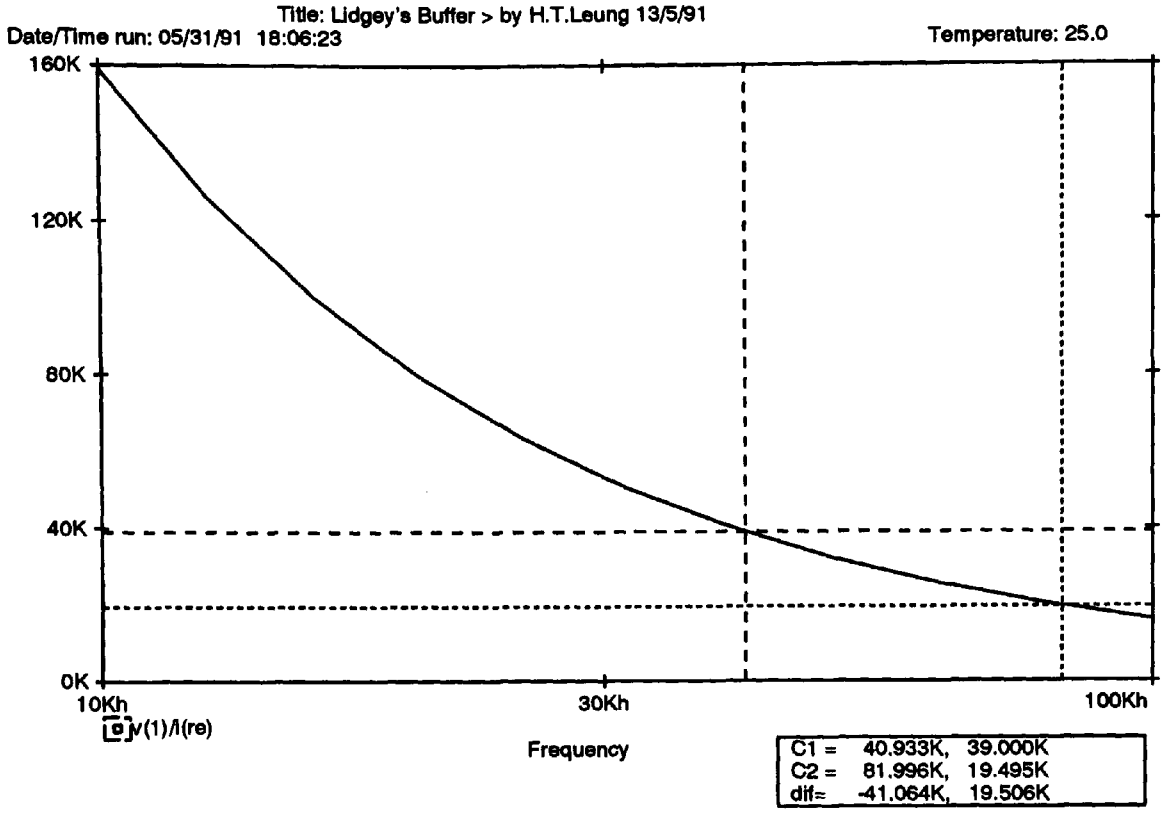


Figure 2.8 A simple ac-coupled voltage follower

(a) Without driven screen



(b) With driven screen

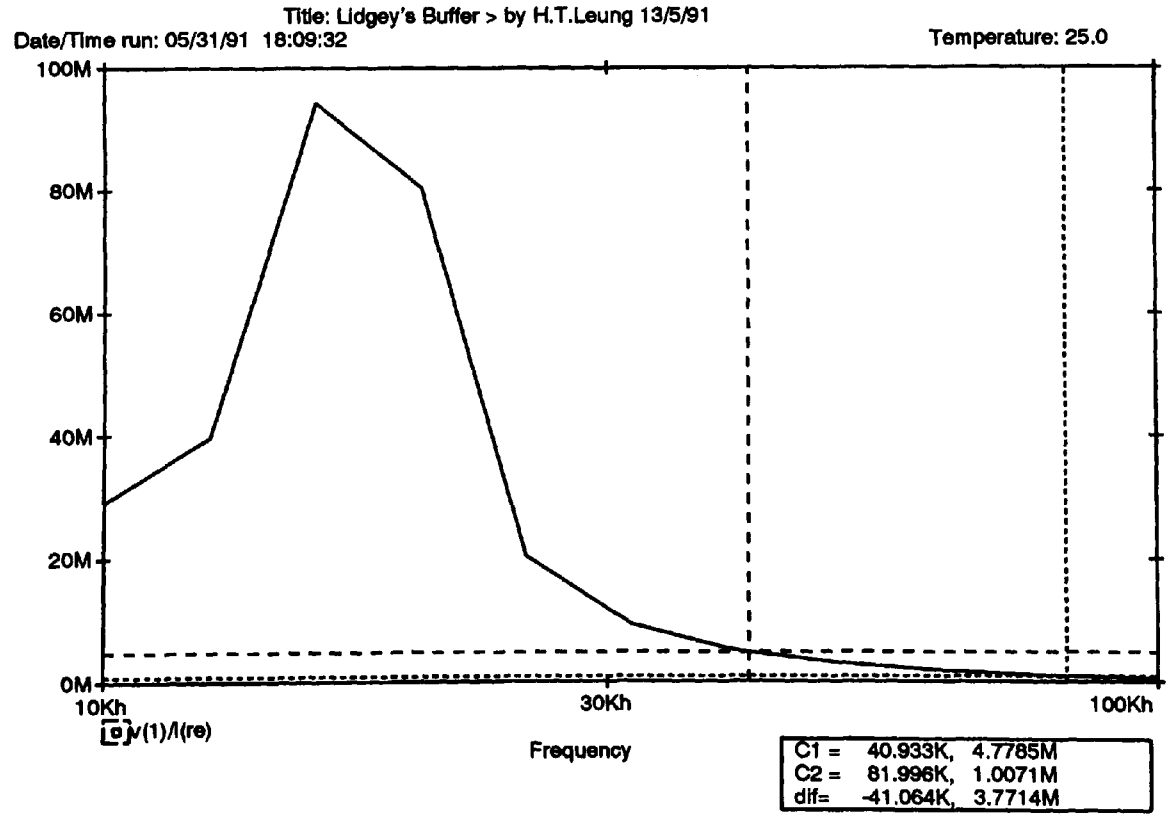


Figure 2.9 Performances of a simple buffer
x-axis: frequency; y-axis: input impedance

With the improved version, a high CMRR of above 60 dB was attained up to 200 kHz. One anticipated demerit is the increased complexity of the circuit. Using integration will reduce the physical size.

As mentioned previously, this group also pointed out that the bootstrap buffering circuit is peaky [37]. They suggested a simple voltage buffer as illustrated in Figure 2.8. Computer simulations were performed by the author in order to understand the practical circuit behaviour. Again, two simulations were conducted for R_e = 1 kΩ, $R_1 = R_2$ = 20 MΩ, C_1 = 1 μF and C_x = 100 pF. Figure 2.9(a) shows, under an undriven screen condition, the input impedance decreased from 39 kΩ to 19.5 kΩ with an increase in frequency from 41 kHz to 81 kHz. On the other hand, Figure 2.9(b) shows better results with a driven screen cable. Input impedances of 4.8 MΩ and 1.0 MΩ were obtained at 41 kHz and 81 kHz respectively.

In comparison to the bootstrap buffer, this simplified version produced better performances. Also, the simplicity of this circuit would be favourable for any future integrations.

(e) U.K. - Sheffield group

The Sheffield data collection system is a 16-electrode system, and utilises the serial stimuli applications and parallel data collection methods. The applied stimuli are generated from a transformer-coupled current source tuned to operate at one frequency only. The parallel data collection was implemented with an array of digital demodulators instead of the conventional analogue demodulation techniques [43]. Each of these digital demodulators consists of a low-pass filter, sample-and-hold, A-D converter and digital latch. Four of them are grouped together to form a module and the demodulation process takes place in a digital signal processing DSP microprocessor. Thus, four DSP devices are required for a 16-electrode data collection system. A fifth processor is used to measure the amplitude and phase of the current while a sixth acts as a central processing controller.

The digital demodulation process employs a so-called matched filtering technique [37]. This technique is a process of correlating two signals in the digital domain to yield a demodulated signal. Generally, the demodulated

signal is a sum of products which is formed by multiplying and adding the digitised incoming surface signal and the digitised reference sinusoidal signal. The resultant represents the in-phase component. When a cosinusoid is used as a reference, the quadrature component is obtained.

The drive electrode pair is selected by means of two 16-channel analogue multiplexers. Each electrode has a driven screen to reduce the stray capacitance effects. A bootstrap buffer is also connected to the electrode. Seagar and Brown [44] indicated that common-mode feedback (CMFB) is not suitable in this parallel data collection method. This is because the common mode voltage at one electrode pair can only be nulled at one time.

As an originator, the Sheffield group has made remarkable progress in the development of both data collection system and image reconstruction algorithms. In June 1990, they claimed the first functional "real-time" impedance imaging system capable of collecting 25 frames per second [45]. At the EEC Workshop in Leuven, it was indicated that an Application Specific Integrated Circuit (ASIC) was intended for their forthcoming demodulator implementation [37]. In addition to the "real-time" system, their first version imaging system has been used extensively for clinical applications.

(f) U.K. - Southampton group

The Southampton group used the Sheffield data collection system as a basis of their EIT investigations but they have been using transputers to enhance the computation power. No hardware development has been reported.

(g) Belgium - Leuven group

The Leuven group has presented an architecture for a voltage drive EIT system without involving any CMFB [37]. It will be a IBM compatible PC based 16-electrode system. The stimulus is to be produced operating in anti-phase. They have indicated that one signal generator may also be possible if it has a differential output. In addition, two resistors need to be connected in series with the voltage sources in order to measure the currents being applied to the object under test. Details of the circuitry are not available as the system is still in the development stage.

(h) Denmark - Lyngby group

The data collection system developed by the Lyngby group consists of sixteen electrodes [46]. The system operates at 40 kHz only. Sixteen current sources and voltage differential amplifiers are used. Each amplifier has a dedicated band-pass filter of centre frequency of 40 kHz. The demodulation process is implemented with a phase detector, a rectifier and a sample-and-hold circuit. Only seven-bit data conversion is performed.

The weakness of this system was the quality of the data since seven-bit data attains approximately 40 dB limiting the possible applications. To perform useful *in-vivo* imaging 60 dB accuracy is required.

(i) France - Lyon group

This group utilises a dual frequency technique for breast imaging [47]. Their system has been particularly designed for this purpose and works on a dual frequency basis. 32 kHz and 250 kHz are the two frequencies employed. Details of the system architecture are not available. However, features of their system can be found elsewhere [47,48]. It is a 32-electrode imaging system and is capable of collecting data in 800 ms. An array of current sources is permanently connected to the electrodes but only one pair is activated at one time. These current sources are placed near the electrodes in order to minimise the stray capacitance effects which have been experienced in the long electrode cables. Three phase sensitive detectors are used for signal demodulation. One is to extract the real part of the peripheral signals for 32 kHz while the other two are to extract the real and imaginary parts for 250 kHz. To obtain these real and imaginary components the resultant signals from these detectors undergo integrations by means of double slope integrators.

The idea of implementing the current sources in the electrode to reduce the stray capacitance effects was good. However, the use of multiple current sources exhibits difficulty of device matching. No images have yet been reported.

(j) France - Toulouse group

The Toulouse group presented their data collection system at IEEE EMBS conference in Seattle in 1989 [49]. It is also a 16-electrode system and can operate in multi-frequency mode for a frequency range from 10 kHz to 100 kHz. Their system has endeavoured to suit all different applications such as utilising the adjacent current drive method, the opposite current drive method and so on [49]. Most of the front-end electronics and the electrode itself form a complete module in order to reduce the stray capacitance effects. Each module may work in one of five operating modes which are "current generator", "short circuit to ground", "two voltage buffers", and "not connected". Due to the difficulties of physical component sizes, the implementation of these modules by means of ASIC was recommended [50]. The methods for applying current to the object to be imaged and recording peripheral potentials are not significantly different from the other reported systems. The A-D conversions are however performed in an interface board which allows digital data to be easily transmitted to a PC. Some details of the electrode module were also presented in the Leuven meeting [37].

Some conductivity images were reported [51]. The system also features minimization of stray capacitance effects by implementing the current source circuits in the electrodes.

(k) Greece - Thessaloniki group

The Thessaloniki group has developed their own data collection system but no details of system architecture was found in the literature survey. However, some circuit layouts concerning the demodulation process were reported at EEC EIT Workshop at Copenhagen and Leuven [37,52]. It was proposed that the demodulation process might take place prior to obtaining the differential signals. The demodulation processes was implemented by means of sample-and-hold circuits. By holding the captured signal at the crossover point of both sinusoidal and cosinusoidal reference signals, real and imaginary parts could be extracted.

The method of measuring the potential difference between the adjacent electrodes creates great problems. The capability of rejecting the common

mode voltages is poor since the possibility of performing high quality A-D conversions and the quality of sample-and-hold circuits are questionable.

(l) Spain - Barcelona group

A parallel 16-electrode data collection system was developed and reported by this group [37,53]. The system is capable of operating at one of four frequencies and is fully controlled by a PC-AT computer. The frequency can be selected from 10 kHz, 20 kHz, 40 kHz and 80 kHz. A serial stimuli application method was adopted in this system but the parallel data collection method was implemented using sixteen differential channels for voltage measurements. The differential surface potentials are measured by means of a discrete instrumentation amplifier of traditional three amplifier circuit topology. The demodulation process adopts the switching amplifier technique similar to the Oxford group. The output signals then undergo a low-pass filtering process to obtain the in-phase or quadrature d.c. equivalent voltages.

A data collection speed of 20 frames per second was claimed [37]. The rate of displaying image is one image per second due to the restriction of the computation speed. The architecture of this system is classic. Problems of using the parallel data collection method were again device matching, and difficulty of employing CMFB.

(m) Sweden - Lund group

This is also a parallel 16-electrode data collection system and is able to work at four different frequencies [37,54]. It is a microprocessor based system and communicates with a standard PC-AT computer. Image reconstruction and display are performed on the host computer. A sine wave generator is used as an applied stimuli device. Current is multiplexed to any electrode by means of two 16-channel analogue multiplexers. Each of the receiving channels consists of a monolithic instrumentation amplifier. However, the demodulation process has not been documented. The demodulated signals are then digitised. Neither the data collection speed nor the rate of image display have been reported. Again, this system adopted the classical front-end architecture.

(n) Turkey - Ankara group

The data collection system being developed by the Ankara group is also a 16-electrode imaging system [37]. Their current source circuit is battery powered. The voltage input of the source is opto-coupled by means of fast linear opto-isolators such as the 6N136. The isolated signals are then ac-coupled to an ordinary inverting amplifier with the feedback path connected to the electrodes. All connections are made with relay switches which are under computer control. Voltage measurements are made using the serial data collection method. A differential amplifier is used for this purpose. Both input signals are ac-coupled and buffered using a unity gain amplifier. A band-pass filter of centre frequency of 10 kHz and Q-factor of 5 is used to remove unwanted frequency components which may be picked up within the system and to bandlimit the noise. The demodulation process adopts a switch-mode technique and the output signal is low-pass filtered. A-D conversions are then performed.

The feature of this system is the battery power supply. Using battery power supply also removes the 50 Hz main frequency noise which is often experienced in any main-supply systems.

(o) U.S. - Madison group

This system consists of 32 electrodes and is claimed to be capable of injecting optimal currents patterns into an object to be imaged [55]. Details of its system architecture were not found. From the limited relevant literature concerning their hardware system, Webster *et al.* [56] showed that band-pass filters and a phase sensitive detector were used for the signal demodulation, and a Macintosh II computer was employed for image reconstructions. No images have yet been obtained from their hardware system but some conductivity images obtained from their computer model were reported [55].

(p) U.S. - New York group

This group invented the Adaptive Current approach [38]. A 32-electrode data collection system was reported by Newell *et al.* [31,57]. The system works at 15 kHz and is apparently able to apply an optimal pattern of currents to an

object under test. 32 current sources are used. Each current source has a dedicated digital-to-analogue (D-A) converter and a programmable voltage attenuator. Thus, only digital signals are to be multiplexed. In the receiving end, a classical voltage measurement arrangement is employed, i.e. a phase sensitive detector is used for the demodulation. The demodulated signal then undergo an A-D conversion. Again, no images have yet been reported.

To summarise, most of the developing data collection systems employ current stimuli, and measure the resultant voltages developed at the periphery of an object. Systems, using multiple currents, usually do so in order to facilitate adaptive impedance tomography or else for the purpose of reducing stray capacitance effects. The sole aim of employing parallel voltage measurements is to shorten the data collection time. This kind of configurations is essential for "real-time" imaging. As far as the demodulation process is concerned, most systems employ the conventional analogue demodulation technique. Additionally, the majority of them use sixteen electrodes because the signal-to-noise ratio decreases with increasing number of electrodes; as mentioned in the preceding section. Finally, since the development of EIT is still in its infancy, there is no approved universal configuration which would provide the best quality data.

As the systems described relate only to imaging conductivity, no specific problems regarding relative permittivity imaging were identified. However, informal discussions with the authors have revealed a number of problems which would be likely to arise in the latter case. Owing to signal feedthrough and crosstalk caused by stray capacitances in multiplexers [34], a method of the parallel stimuli application, i.e. using multiple current sources, seemed to provide the best solution to the stray capacitances problems, and hence was adopted. Since, it was not intended to perform "real-time" imaging, a serial data collection method was proposed. An analogue demodulation was adopted as there were no outstanding advantages of using digital demodulation apart from the possibility of improvement in data collection speed. Since "real-time" imaging was not intended, this type of demodulation was therefore not particularly useful in the proposed system. It was thought that the circuitry involved in the demodulation process should facilitate easy circuit substitutions. If an improved demodulation process was available at some later date, it could be easily implemented in the proposed system. Finally, sixteen electrodes were to be employed.

2.4 Relative Permittivity Imaging

In addition to imaging conductivity, relative permittivity images are also possible when complex variables are used as image reconstruction parameters in the backprojection process. The first relative permittivity image in EIT was shown by Griffiths using computer simulation [11] in 1987. A year later, he demonstrated the feasibility of imaging relative permittivity using a two-dimensional resistor mesh [12]. In his computer studies, using a backprojection reconstruction method, conductivity and permittivity images can be evaluated. These images can only be obtained when the reference data is homogeneous and non-reactive. With an abdominal cross-sectional computer model, liver explicitly appears in the permittivity image while both liver and kidney become the dominant tissues in the phase image (an image represents the changes in phase between the first and second sets of measurements) [11]. The distinction between permittivity and phase images will be explained in the following paragraphs. With aid of a resistor-mesh, results obtained from computer simulations [11] and practical measurements [12] were also compared. The reported images reveal the possibility of separating the conductivity and permittivity data contained in the measured voltages.

Another "permittivity" image for an impedance tomography route has also been reported by Scaife *et al.* [58]. The image was obtained by means of magnetically induced currents instead of injected currents. Voltage measurements at the periphery of the object were taken in a similar fashion as in EIT. However, this is another impedance imaging technique which will not be further discussed here.

In principle, any of the present reconstruction algorithms can be used to produce conductivity or permittivity images; the use of the image reconstruction parameters being the only difference. It would be necessary to implement the algorithms in complex numbers having measured the moduli and phases of the signals. As an example, Griffiths [11] demonstrated a method to obtain the permittivity image from his computer simulation studies. Using a backprojection algorithm implemented in complex numbers with $\ln(V_1^*/V_2^*)$ as the input data, where V_1^* and V_2^* are the reference and data signals, an image of complex electrical conductivity $\ln(\sigma_2^*/\sigma_1^*)$ can be reconstructed, where σ_1^* and σ_2^* are the reference and final complex conductivities. Instead of involving the computation of complex numbers, Griffiths [11] also suggested an alternative method to produce conductivity and permittivity images. By splitting the data prior to the backprojection process, the real and imaginary parts of s^*, where $s^* = s' + js''$, can

be reconstructed. The components of s^* can be written as

$$s' = \frac{1}{2} \ln \left(\frac{\sigma_2'^2 + (\omega \varepsilon_o \varepsilon_2)^2}{\sigma_1'^2 + (\omega \varepsilon_o \varepsilon_1)^2} \right) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

$$s'' = \tan^{-1}(\omega \varepsilon_o \varepsilon_2 / \sigma_2') - \tan^{-1}(\omega \varepsilon_o \varepsilon_1 / \sigma_1') \quad \ldots\ldots\ldots\ldots (2.7)$$

where σ' is the real part of the complex conductivity, ε is the relative permittivity, ε_o is the permittivity of free space and ω is the angular frequency.

Suppose that the real and imaginary signals are denoted as V_r and V_i. The parameters of its polar form are denoted as $|V|$ and θ. Using the transformation shown below, both $|V|$ and θ can be evaluated.

$$|V| = \sqrt{(V_r^2 + V_i^2)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.8)$$

and

$$\theta = \mathrm{Tan}^{-1}(V_i / V_r) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.9)$$

If $|V_1|$ and θ_1 are the reference and $|V_2|$ and θ_2 are the data, then each datum of $\ln(V_1^*/V_2^*)$ can be expressed as $\ln(|V_1|/|V_2|) + j(\theta_1 - \theta_2)$. The parameters for reconstructing the magnitude and phase images will be

$$M = \ln(|V_1|/|V_2|) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.10)$$

and

$$P = \theta_1 - \theta_2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.11)$$

Griffiths did apply these parameters in his preliminary studies and successfully obtained the phase images [11,12]. He also mentioned that conductivity and permittivity images might also be obtained when the reference had negligible reactance, i.e. $\omega \varepsilon_o \varepsilon_1 << \sigma_1'$. Using the transformations proposed by Griffiths [11], the conductivity and permittivity images can also be evaluated. These transformations are

conductivity : $\alpha \equiv \ln(\sigma_2'/\sigma_1') = s' + \ln\cos s''$ . . . . . (2.12)

and

permittivity : $\beta \equiv \ln(\omega\varepsilon_0\varepsilon_2/\sigma_1') = s' + \ln\sin s''$ . . . . . (2.13)

The real parts of the signals are affected by capacitance to much less extent than the imaginary parts. Thus, if only conductivity images are required, it is probably more accurate to backproject the ratios of the real parts than to evaluate $|V|$ and θ as above.

To summarise the whole review, only a few reported systems successfully produced images as most of the systems are still in their development stage. Apart from the Cardiff group - the collaborating group, none of the others produced relative permittivity images. Since there were no quantitative methods for assessing system performances, no approved system configurations, which would produce the best quality data, were available. The proposed system therefore adopted the "best" architecture for the investigations of relative permittivity imaging. As a result, a multiple drive and serial data collection architecture was adopted. The stimulus would be current and the number of electrodes would be sixteen. The demodulation processes would be achieved by means of analogue multipliers.

Chapter 3

An Investigation into Current Sources for EIT

In impedance tomography, the constancy of the voltage or current sources influences significantly the quality of the recorded data, and hence the reconstructed images. When a current source cannot produce a highly constant amplitude alternating current for a given load range, there will be no common reference between frames of data, which is essential. Owing to the finite output impedance of current sources, the applied currents would be load dependent. When this situation occurs in impedance imaging processes, the potentials across the surface electrode pairs will be affected. This results in an error because the differences in the peripheral voltages are not due to the perturbations of the electric field but due to the discrepancy in the drive currents. Thus, discussions will focus on the topology of various current sources. Considering a simple equivalent current source circuit illustrated in Figure 3.1, the output impedance Z_o determines the quality of the current source. Ideally, the output impedance should be infinite. In practice, they are finite and thus every effort is required to make the output impedance as high as possible. Since a desired constancy of 0.1% (i.e. equivalent to a 10-bit accuracy, refer to p.12) was considered to be necessary for impedance imaging, an output impedance of 1 MΩ is required for a maximum load change of 1 kΩ. Table 3.1 provides a chart of a minimum output impedance to sustain the required constancy. It is anticipated that reactive components contained in the impedance also affect the quality of the output current. This is discussed in conjunction with the stray capacitance effects in the measurement circuitry, chapter four. The effects of this on relative permittivity imaging are discussed in chapter six.

Theoretically, if the amount of current flowing through a test object can be measured accurately and precision voltage measurements are performed in every drive position, then the effect of the output impedance may be eliminated to a large extent as the recorded peripheral voltages could possibly be normalised with the observed currents. However, precision current measurements require extra data collection time and electronics. In the case of multiple current drive configurations, the complexity of the circuitry increases significantly with attendant noise problems. Thus, using high quality current sources may be the best alternative.

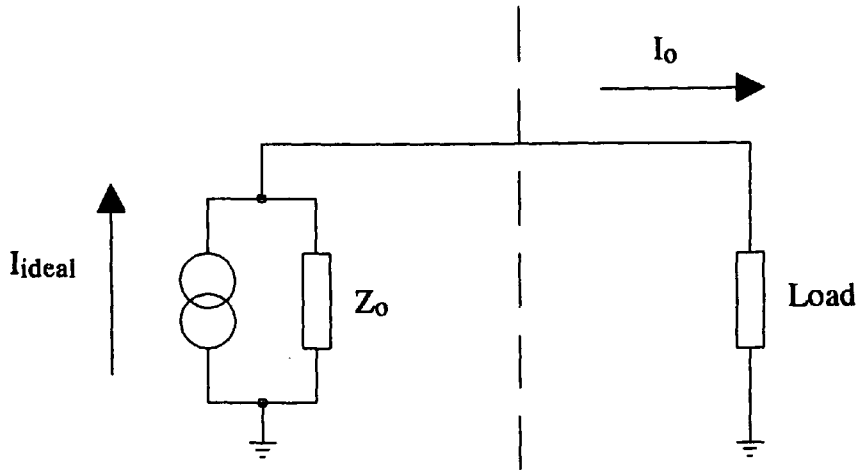


Figure 3.1 Equivalent constant current source

Initial Load = 0.5 kΩ

Increase in Resistance (kΩ)	Theoretically Required Output Impedance 12-bit(kΩ)	10-bit(kΩ)
+0.5	2047.0	511.0
+1.0	4094.5	1022.5
+1.5	6142.0	1534.0
+2.0	8189.5	2045.5
+2.5	10237.0	2557.0
+3.0	12284.5	3068.5
+3.5	14332.0	3580.0
+4.0	16379.5	4091.5
+4.5	18427.0	4603.0

Table 3.1 A chart of output impedances required to sustain 10-bit and 12-bit constancy

Apart from the requirement of a high output impedance, the capability of operating over a wide frequency band, i.e. from 10 kHz to 100 kHz, is also desirable. If the performance of a current source is consistent across the required frequency band, less problems would be exhibited in multi-frequency applications. The choice of output stage configuration between single-ended and double-ended needs to be considered. To perform peripheral voltage measurements, the drive electrode pair is required to produce a sink/source action. Multiple single-ended current sources exhibit difficulties in matching but benefit from the fact that no currents have to be multiplexed; with attendant stray capacitance problems. Double-ended current sources have virtually no current leakage problems due to mismatches of devices. In practice, when they are selected as the drive devices, a serial drive method is usually adopted. In the presence of multiplexers, problems of crosstalk and feedthrough thus arise. This could be serious in relatively high frequencies such as those used in EIT work. Suppose that the outputs of a 16-channel demultiplexer with channel capacitances of 10 pF each and channel-on resistances of 200 Ω each are individually connected to a load of 1 kΩ as illustrated in Figure 3.2. The effect of feedthrough can then be evaluated when considering that only one switch is closed and the others are open. A total leakage of about 1.13% of drive current at 10 kHz is calculated to flow due to feedthrough. It will be even worse when at 100 kHz where a total leakage current of 11.2% results.

To evaluate the influence due to crosstalk, it is convenient to analyse the situation by utilising the configuration of a four-electrode system. An equivalent stray capacitance network is illustrated in Figure 3.3(a). C_x is the capacitance of the combination of stray capacitances between electrodes and inter-channel parasitic capacitances within the multiplexer. Figure 3.3(b) shows a simpler equivalent network may be obtained by using a Delta-Wye circuit transformation. The network has been derived by considering the six capacitors in isolation. A detailed analysis results in $3C_x$ [28]. Suppose that the total load including the electrode contact resistance is 2 kΩ and C_x = 10 pF. A leakage current $I_{leakage}$ of approximately 0.25% occurs at 10 kHz due to crosstalk. At 100 kHz, a 2.5% leakage current is calculated. Although the above discussions have only considered the magnitudes of the signals, it is recognised that phase displacements also result. If only the conductivity of the object is of interest, the stray capacitance effects are not as vital as in dual-frequency imaging or in relative permittivity imaging. Thus, the use of single-ended current sources together with the multiple drive

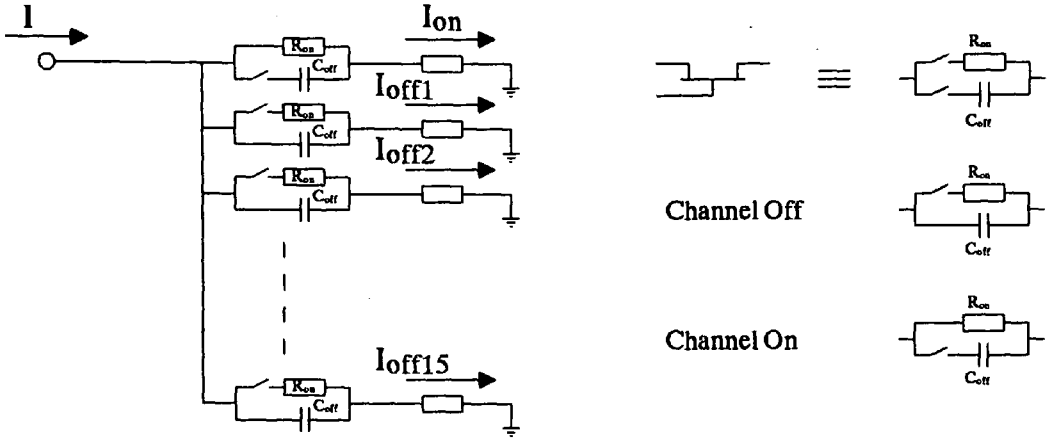
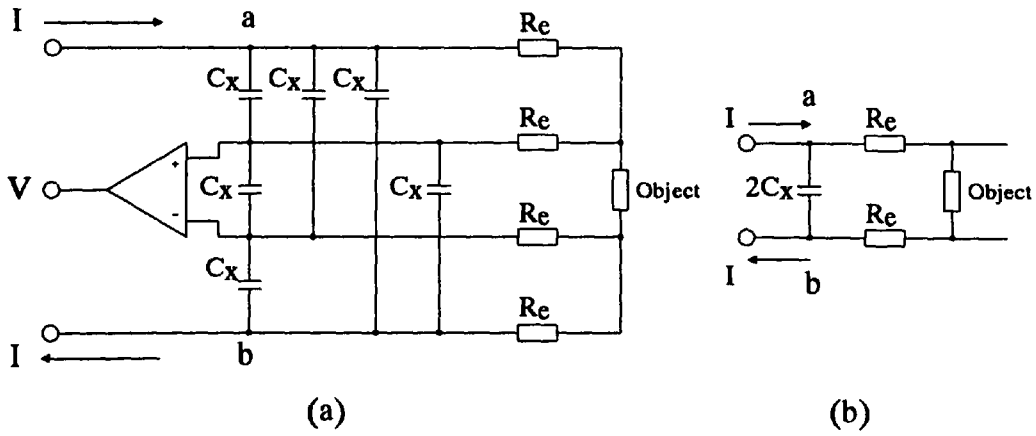


Figure 3.2 An equivalent circuit for the investigation of feedthrough



C_X: stray capacitance to be present between electrodes

Figure 3.3 An equivalent circuit for the investigation of crosstalk

configuration was proposed in this programme in order to minimise the capacitance effects.

Additionally, it is also necessary to ensure that no additional harmonics would be produced by the circuit, e.g. crossover distortion. Since these could be transformed into errors in an analogue demodulation process.

To summarise, the specification of the desired current sources was as follows :

(a) the output should be single-ended,
(b) the circuits should be bilateral, with no d.c. current in the output, (safety requirement [35]),
(c) it should have high output impedance; of the order of MΩ, or 0.1% constancy for a varying load within 1 kΩ,
(d) its performance should be consistent from 10 kHz to 100 kHz,
(e) the circuit should include provision for ease of matching,
(f) the circuit should be reproducible to close tolerance in small quantity, i.e.16,
(g) the circuit should not produce additional harmonics.

3.1 Studies of Current Sources

A current source can be developed by means of voltage-mode operational amplifier (VOA) circuits or current conveyors (CC). In the past decades, a number of current sources were developed. A comprehensive review of all kinds of current source circuits was conducted by Toumazou *et al.* [59]. It is not intended to repeat the detailed discussion of these circuits in here. However, a discussion on other current sources presently in use in the EIT applications is provided below.

3.1.1 A Three-VOA Current Source Circuit

Figure 3.4 shows a single-ended bilateral current source described by Wojslaw and Moustakas [60]. The output current is tracked by a "sensing" resistor R_s and maintains invariable by means of a "bootstrapping" effect which makes the current independent of the load. Using linear circuit theory, the output current I_o can be expressed in the form as shown in Equation 3.1.

$$I_o = - V_{in}/R_s \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$

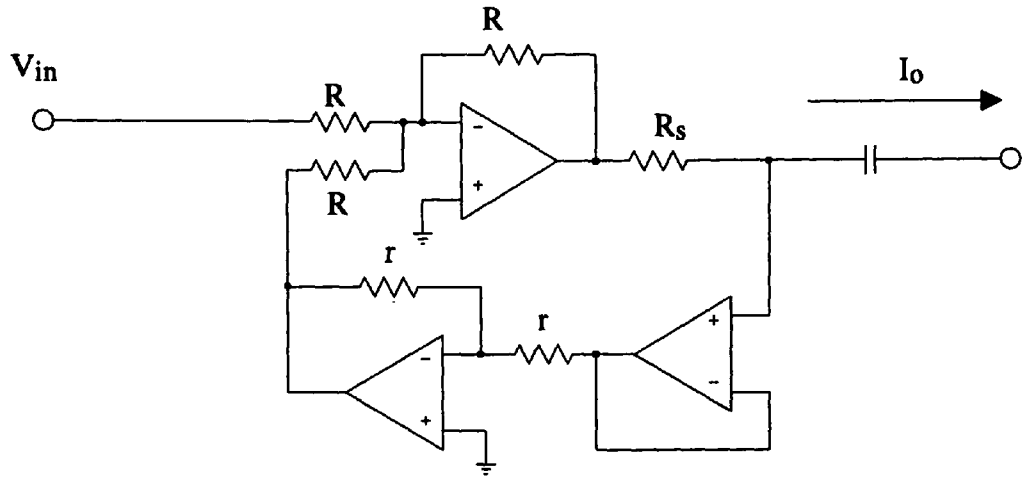


Figure 3.4 A three-VOA current source

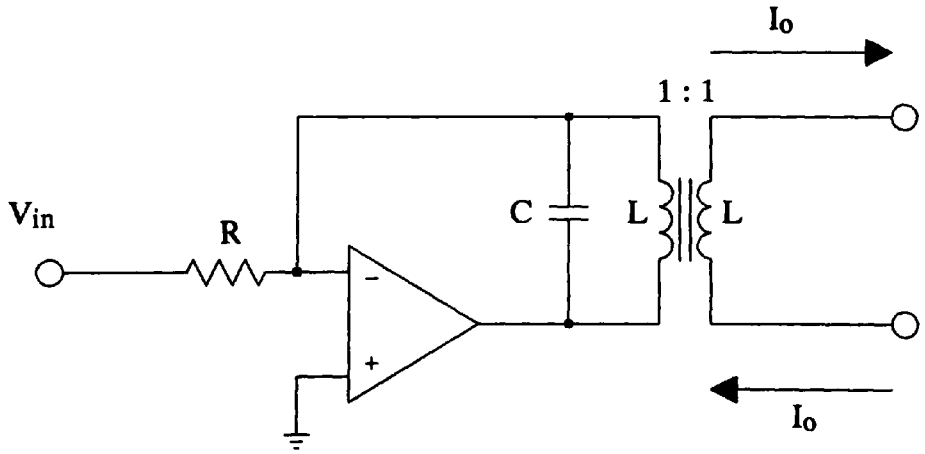


Figure 3.5 A transformer-coupled current source

The circuit was used by Newell in their EIT data collection system [31]. Precision resistors matching is essential if the circuit is to work correctly. Slight mismatches of the resistors lead to a significant reduction in output impedance. In order to attain the desired constancy of 0.1%, the tolerance of each resistor must not exceed 0.016% [39]. It is also bandlimited by the VOAs. Despite all these weaknesses, conductivity images were however successfully reconstructed using this kind of current sources in EIT [31]. Since the consistency of the current relies heavily on the quality of resistor matching, it would be inadequate in a multiple current source system and hence was not employed.

3.1.2 A Transformer-coupled Current Source Circuit

Figure 3.5 shows probably the most common current source circuit presently in use in EIT. It is a double-ended bilateral current source. This circuit makes use of the fact that a simple negative feedback inverter implemented with a VOA is a good current source when a load is placed in the feedback path, provided that the input voltage V_{in} and the input resistor R remain unchanged. In this case, an isolation transformer is placed in the negative feedback loop for safety purposes. A sink/source current pair at its secondary winding is produced naturally when an object is connected to the winding.

However, the disadvantage of this type of circuit is frequency dependence. Since any load connected in the secondary winding can be transferred to the primary side by multiplying the square of the turn ratio, (for an isolation transformer, the turn ratio is 1), the circuit then becomes a R-L parallel network if the load is purely resistive. L is the total inductance in the primary side including the leakage inductance. The current flowing in the negative feedback path then splits and also causes phase displacement. This clearly reduces the output impedance of the circuit. To minimize this effect, a tuning capacitor C (Figure 3.5) is needed. When adjusting C such that $C = 1/\omega^2 L$, the effect of L could then be cancelled by C, and hence the output impedance increases. The need of tuning creates a problem in multi-frequency operations so that the circuit was not considered.

3.1.3 A Howland's Equivalent Circuit

The circuit illustrated in Figure 3.6 was designed by Nowicki [61]. It exploits the identical configuration developed by Howland [62] but replaces the resistive

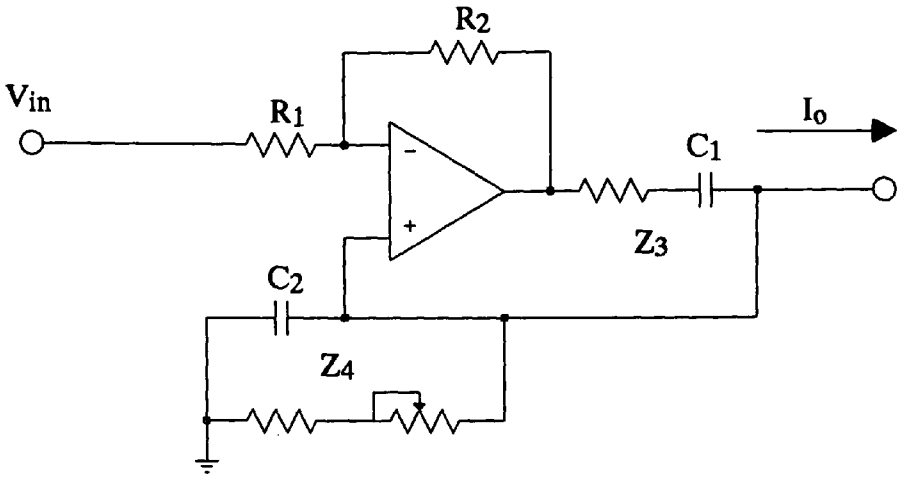


Figure 3.6 Howland's equivalent current source

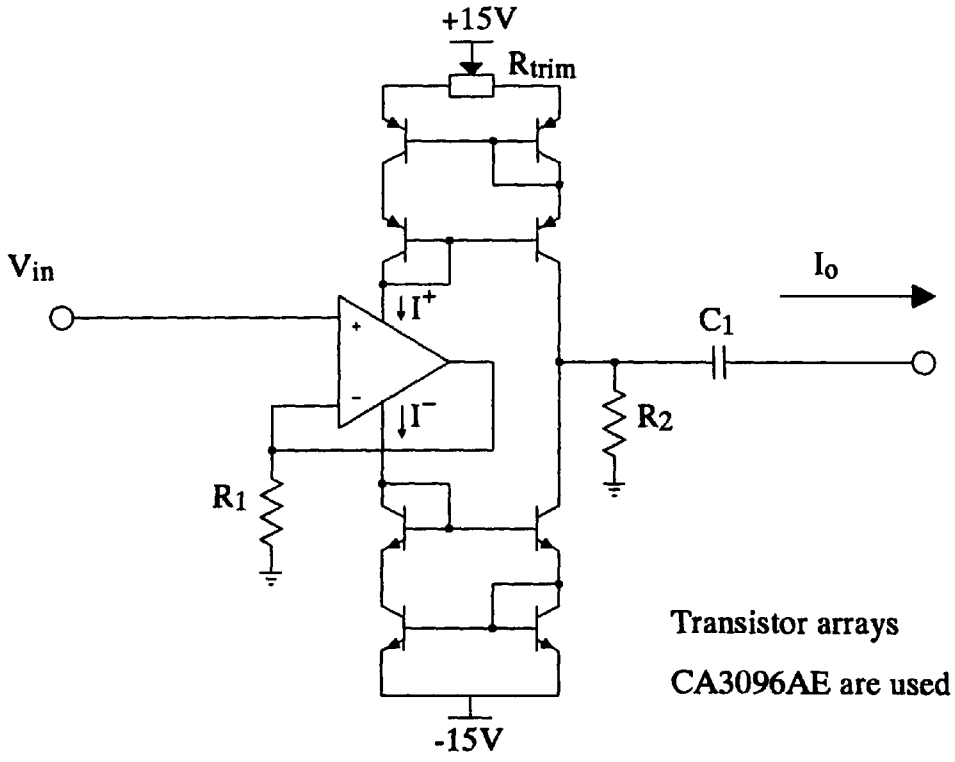


Figure 3.7 VOA supply current sensing current source

components with complex impedance components in the positive feedback path. Due to the employment of positive feedback, instability may be a potential problem. Thus, extreme care needs to be taken in selection of components. In order to limit low frequency currents or even d.c. current to flow in the patient, a.c. coupling is implemented with capacitor C_1. A second capacitor C_2 is required to balance the phase shift caused by Z_3. With reference to the Figure, the output current I_o can be derived using the linear circuit theory of VOA which gives

$$I_o = \frac{-R_2Z_4V_{in}}{Z_{load}(R_1Z_3-R_2Z_4) + R_1Z_3Z_4} \quad \ldots\ldots\ldots\ldots \quad (3.2)$$

where Z_{load} denotes the load impedance. To eliminate Z_{load} from the Equation it is necessary to make $R_1Z_3 - R_2Z_4 = 0$. For ease of implementation, making $R_1 = R_2$, tuning of $Z_4 = Z_3$ is therefore required. In the presence of R and C in Z's, multi-frequency applications again exhibit difficulties.

3.1.4 A VOA Supply Current Sensing Circuit

A single-ended bilateral current source illustrated in Figure 3.7 is the only circuit making use of the VOA supply current sensing technique together with current conveyor concepts to produce theoretically high constancy currents [63]. This circuit was adopted and implemented by the Oxford group in their EIT development work [39]. The function of the current mirrors is to sense the supply line currents I^+ and I^- of the VOA and duplicate a known amount of current, which is the difference of I^+ and I^-, to flow in a load unconditionally. This implies that one property of current mirrors is its high output impedance. The VOA is used to control the output current value. As can be seen from Figure 3.7 and assuming the VOA is perfect, the output current I_o can be expressed in the form show below.

$$I_o = -V_{in}/R_1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.3)$$

In practice, due to a discrepancy of current transfer ratios between the top and bottom modified Wilson current mirrors and the presence of a large resistor R_2, usually of the order of MΩ, situated at the output stage, a standing voltage appears at the output which may cause the current source not to operate properly. In order

to reduce the standing voltage, the value of the output resistor may be made smaller or trimming the transfer ratios with R_{trim} in Figure 3.7. Either method could possibly alleviate the effect of the unequal current transfer ratios. This circuit has an advantage that no resistor matching is required. Instead, matching of transistors needs to be dealt with. Using closely matched transistors in monolithic integrated circuits, such as CA3096AE, may ease matching difficulties. Bearing in mind that these transistors are still not perfectly matched but are nevertheless just adequate in this application.

This circuit possesses many advantages such as single-ended bilateral outputs, anticipated high output impedance and relatively wideband. It was therefore considered to be employed in the proposed system.

3.2 A Modified Current Source for Use in EIT

As mentioned in the preceding paragraphs, it was thought that the single-ended current source using the VOA supply current sensing technique would be suitable in the author's applications. A replica was therefore constructed using two transistor-arrays CA3096AE and a VOA AD548. With proper adjustment of R_{trim}, the circuit was found to be functional but limited to operate at a low current level such as 1.4 mA (r.m.s.) approximately if a 10 kΩ trim pot was used. When the circuit attempted to deliver current greater than 1.4 mA, the signal started to distort. This signal distortion was due to the voltage developed across the R_{trim} which limited the possible voltage swing at the output of the VOA. When a 4.7 kΩ pot was used as R_{trim}, the situation was improved. The circuit was also found to be limited to operate up to 30 kHz approximately. Signal distortions occurred when the frequency exceeded 30 kHz. The same situation resulted for another type of VOA, such as LF441. Additionally, the d.c. standing voltage at the output varied with the output current values even if the input was ac-coupled. When the output current levels increased or decreased, re-adjusting the value of R_{trim} was required. This causes difficulties in the proposed system as applications of different current levels may be necessary for some circumstances. If the circuit was not properly adjusted after alterations of the current values, clipping would occur for high resistance. Thus, this current source was not used eventually. As a result, it was necessary to design and develop a new wideband current source with the specifications stated above. The prototype circuit, however, showed a good performance in phase responses. Phase displacement between the input and output

signals was measured to be approximately 4° at 30 kHz. A new current source circuit based upon the principle of the supply current sensing technique with the complementary current mirrors topology was designed and shown in Figure 3.8. This is unique to the author's system and was presented at an international conference held in Copenhagen in June 1990 [64]. This is described below following a discussion of current mirror circuits.

Current Mirror Circuits

The circuit shown in Figure 3.9(a) is the basic circuit. Its output current (I_o) is controlled by the input current (I_i). The amount of the input current to be copied to the output is dependent on current transfer ratio λ (defined as I_o/I_i) and in this case $\lambda \approx 1 - 2/\beta$ for $\beta >> 1$. Clearly, λ is dependent on β. Due to the difference in β for n-p-n and p-n-p transistors, a serious problem occurs if this type of current mirror is used to form a complementary current mirror pair; this will be further discussed later. In addition, λ is also affected by the matching of V_{BE} and the difference in the collector-base voltages of the two transistors Q_1 and Q_2 and the Early intercept voltage at the operating point [69,70].

Fortunately, a better current transfer ratio can be obtained by including an additional transistor resulting the well-known Wilson current mirror, as illustrated in Figure 3.9(b). Its transfer ratio becomes

$$\lambda \approx 1 - 2/\beta^2 \qquad \text{for } \beta >> 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots (3.4)$$

and is much closer to unity and less influenced by changes of β. Also the effect of the difference in collector-base voltages becomes smaller as the difference is about 0.7 V.

Introducing a fourth transistor results in the modified Wilson current mirror as depicted in Figure 3.9(c). The performance of this current mirror is even better as the influence of collector-base voltages becomes negligible. The problems of matching of β and V_{BE} however remain.

Using the modified Wilson current mirror to form a complementary current mirror pair still exhibits a trivial problem. A simple circuit illustrated in Figure 3.10 can

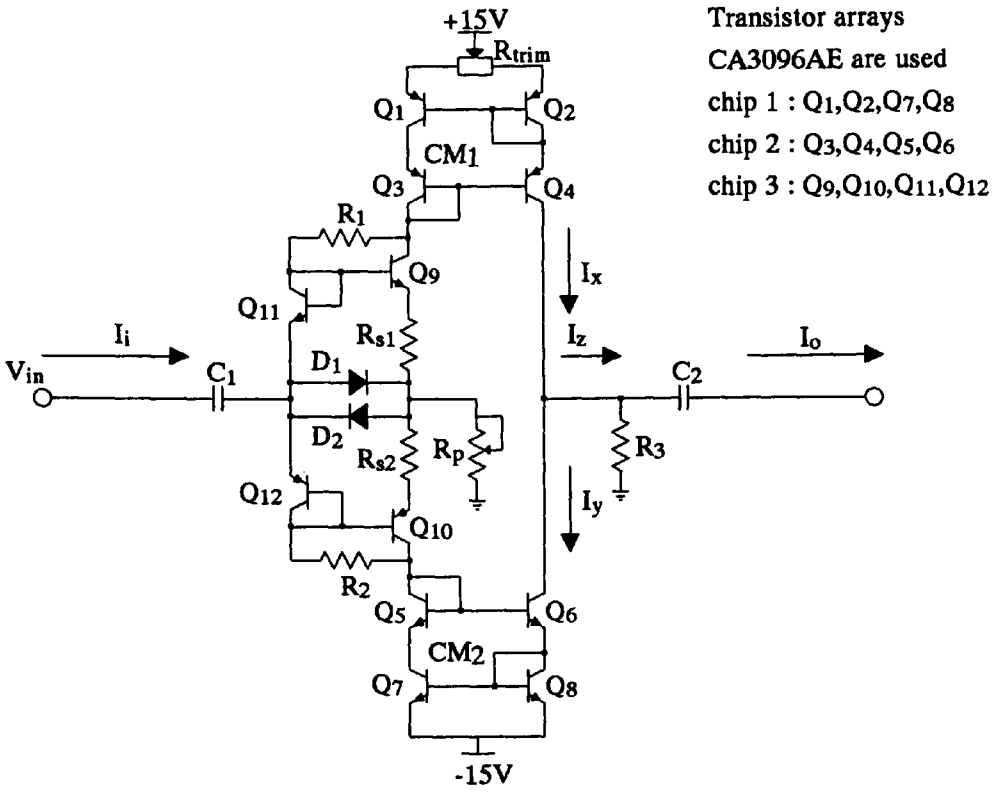


Figure 3.8 Leung's current source

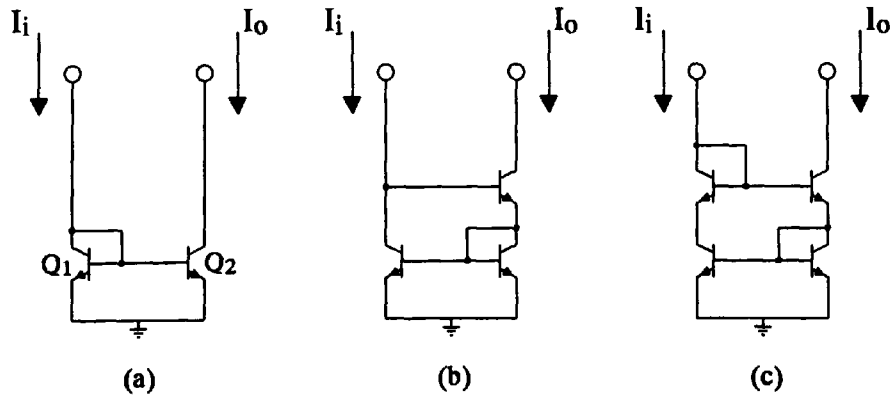


Figure 3.9 Current mirrors

easily show the problem. Suppose that β of n-p-n is 400 and β of p-n-p is 50. The current transfer ratios of n-p-n and p-n-p current mirrors are calculated as 0.99999 and 0.99923. If R_1 = 30 kΩ is used, this will cause a current of approximately 1 mA to flow at the inputs of the current mirror pair. Due to the discrepancy of the current transfer ratios of the n-p-n and p-n-p current mirrors, an unbalanced current may flow through the output resistor R_3. If the output resistor is very large, say of the order of MΩ, a standing voltage would be developed at the output. This would limit the possible output voltage swing. From this, it indicates when this supply current sensing technique is used in implementations of a current source, the quiescent current (defined as the current flows when the input voltage is zero) should therefore be kept as small as possible.

Leung's Current Source

Referring to Figure 3.8, it can be seen that a complementary current mirror pair is employed. The principle of operation is identical to those discussed previously and thus is not repeated here. In this case, R_3 is 10 MΩ. Suppose that a standing output voltage of 1V is allowable for a quiescent current of 1 mA. The unbalanced current is therefore less than 0.1 μA. This indicates the discrepancy of current transfer ratios between p-n-p and n-p-n current mirrors must not exceed 0.01%. For a monolithic transistor array CA3096AE, n-p-n transistors have a typical value of β of 390 and p-n-p transistors have β of 47. This results in a difference in current transfer ratio of about 0.085%, hence gives a standing voltage of 8.5 V. In order to meet the above specification for β of n-p-n transistors of 390, β of p-n-p must not be less than 132.

Present fabrication technology allows the gain of n-p-n transistors and p-n-p transistors to be matched within types. Therefore, an unbalanced current always exists in this application. To alleviate the effect of the unbalanced current a biased supply current altered by R_{trim} (Figure 3.8) is necessary. To reduce the quiescent current would be another alternative; this can be done by reducing the supply voltages. This method of reducing the quiescent current is not applicable when VOA is used, since this current is less susceptible to the power supply voltages in these circumstances.

The resistively biased complementary emitter follower used and illustrated in Figure 3.8 is a very common circuit topology often used in the output stage of

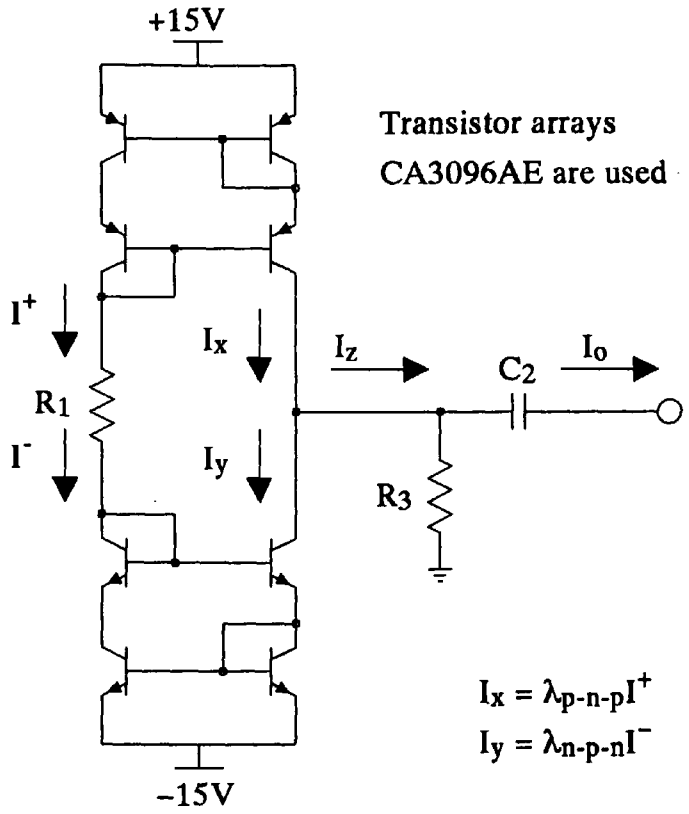


Figure 3.10 Evaluation of imbalance current due to the discrepancy of current transfer ratios

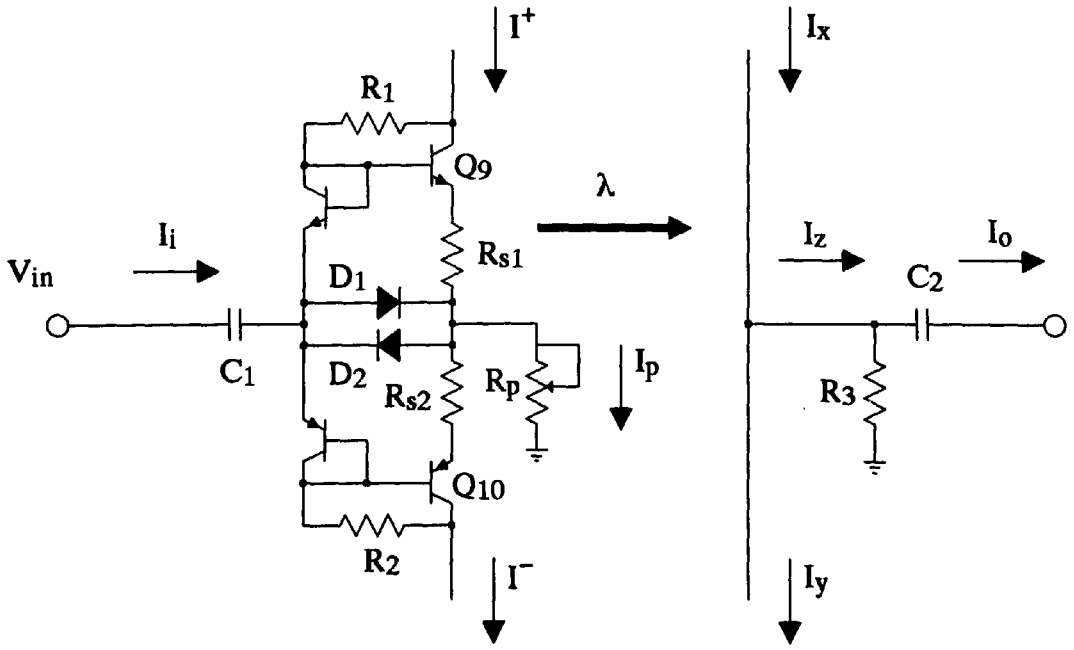


Figure 3.11 Simplified circuit for analyses

integrated amplifier devices [71]. The role of this emitter follower is similar to that of VOA in previous circuit (Figure 3.7, p.45). It operates in class AB to avoid crossover distortion. For the sake of clarity, the current mirrors are removed from the circuit and the circuit operation can be discussed according to Figure 3.11. Applying Kirchhoff's Current Law, the sum of currents in Figure 3.11 must be equal to zero, i.e.

$$I^+ - I^- = I_p - I_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.5)$$

Since it is an emitter follower, the input voltage can be written as

$$V_{in} = I_i R_{in} = I_p R_p \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.6)$$

for the ideal case.

R_{in} represents the input impedance of the follower and is dominated by the parallel combination of the biasing resistor R_1 and R_2. Since, the transistor-connected diodes exhibit negligible resistance under forward bias condition and a small amount of base currents flows into transistors Q_9 or Q_{10} indicating high impedances, therefore, the two transistor-connected diodes as well as the output circuitry can be neglected for the purpose of analysis. With these approximations, the circuit can be simplified and is shown in Figure 3.12. Using maximum power transfer theorem, the input impedance R_{in} is found approximately to be the parallel combination of R_1 and R_2.

Manipulating Equations 3.5 and 3.6 gives

$$I^+ - I^- = (1/R_p - 1/R_{in})\, V_{in} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.7)$$

Suppose that the current transfer ratios of the complementary current mirrors in Figure 3.8 are identical and designated as λ, then Equation 3.7 can be rewritten and gives

$$I_o = \lambda\,(1/R_p - 1/R_{in})\, V_{in} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.8)$$

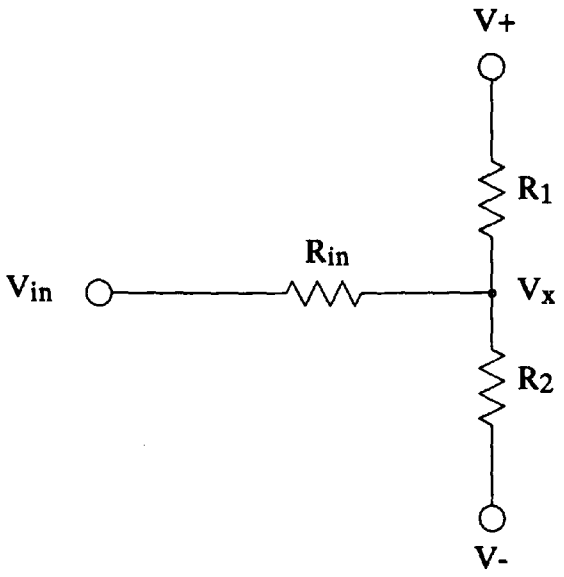


Figure 3.12 Equivalent input circuit

From Equation 3.8, it can be seen that I_o can be produced in-phase or anti-phase and is entirely dependent on the choice of R_p. When R_p is large, the output current I_o is dominated by I_{in}. Effect of this is a net current transfer to the output which produces an output current in anti-phase to the drive voltage. When R_p is very small, I_p becomes the dominant component and an in-phase output current is produced. R_p can therefore be used to control the output current I_o. In practice, R_p is used for the matching of sink/source pairs; the control of the output current I_o is effected by the input voltage V_{in}.

Resistors R_{s1} and R_{s2} are used to avoid occurrences of thermal runaway while the two diodes D_1 and D_2 help to avoid excessive collector currents to flow in the current mirrors, i.e. the circuit incorporates a current limit. In the configuration shown in Figure 3.8, the current limit does not occur until the collector current exceeds 27 mA approximately (i.e. 0.6V/22Ω). Note that this is not for safety purposes but for device protection.

To conclude, the output current (I_o) should therefore only be affected by the drive voltage V_{in}, programme resistor R_p, and the current transfer ratios. It should be independent of the load. In practice, the output current does vary for different loads indicating a finite output impedance for the circuit. The output impedance is governed by the degree of matching of transistor parameters (β and V_{BE}) and the value of resistor R_3. One disadvantage of this configuration is its susceptibility to changes in the supply rails which, essentially, need to be highly stable. The circuit must be driven from an adequately stable, low output impedance, voltage signal source.

3.2.1. Computer Simulations

Prior to commencing hardware developments, comprehensive computer simulations of the circuit were executed. The SIA-ASTEC3 package was chosen as the design tool due to its comprehensive package of functions. This simulator operates on a VAX11/785 computer.

Two kinds of simulations were performed. These were "functional simulations", to assess the theoretical validity of circuits, and "statistical simulations", to assess the feasibility of real circuits. During statistical analyses, the simulator was used to exercise the circuit with the nominal component values followed by a large

number of statistical runs with randomly selected values from within specified tolerance ranges. This was extremely useful for predicting the behaviour of a set of current sources, especially since the need for matching had been predicted to be very important.

ASTEC3 did not contain a model of the CA3096AE transistor array utilised. Employing a standard h-parameter transistor model, two computer models of n-p-n and p-n-p devices were created. The values of h_{ie}, h_{re}, h_{fe} and h_{oe} are obtained from the manufacturer's data sheet. Non-linearity characteristics, such as the variation of transistor current gains with collector current, were however not implemented in the model. By using these models in the current source, simulations were performed for a load range of 500 Ω to 5 kΩ across a frequency range of 10 kHz to 100 kHz. During statistical analyses, the transistor parameters were given a Gaussian distribution based on 0.5% (intra device) and 1% (inter device) tolerances; figures as quoted by the manufacturer.

Figure 3.13 shows the simulation results with nominal component values. The output impedance of the current source is inversely proportional to the current/load gradient and can be calculated from the data shown. The values obtained were 2.63 MΩ at 10 kHz and 2.52 MΩ at 100 kHz indicating a variation of approximately 4% with frequency within the operating range. With an output impedance of 2.5 MΩ, an 11-bit constancy is obtained for a load variation of less than 1 kΩ. Note that this result was the best case simulation in which all transistors involved were perfectly matched. In addition, the results also indicated that a 0.32% reduction in the output current results for a constant amplitude drive voltage, and load, when the signal frequency increases from 10 kHz to 100 kHz.

For the statistical analyses, involving a population of 100 circuits, the results showed an output impedance variation of 55% at 10 kHz (mean = 1.5 MΩ and SD = 830 kΩ) and 26% at 100 kHz (mean = 1.2 MΩ and SD = 310 kΩ). These figures reflect the deterioration in output impedance which results from variations in transistor parameters relative to the perfectly-matched case.

Although the results were not ideal, the circuit was nevertheless adopted for the initial system developments. The reasons for this are: (i) the complexity of circuit is reduced significantly as a whole in comparison to the VOA supply sensing current source discussed in section 3.1.4 (p.46). The basic components utilised are just transistors and diodes plus a few passive components, and thus is favourable for

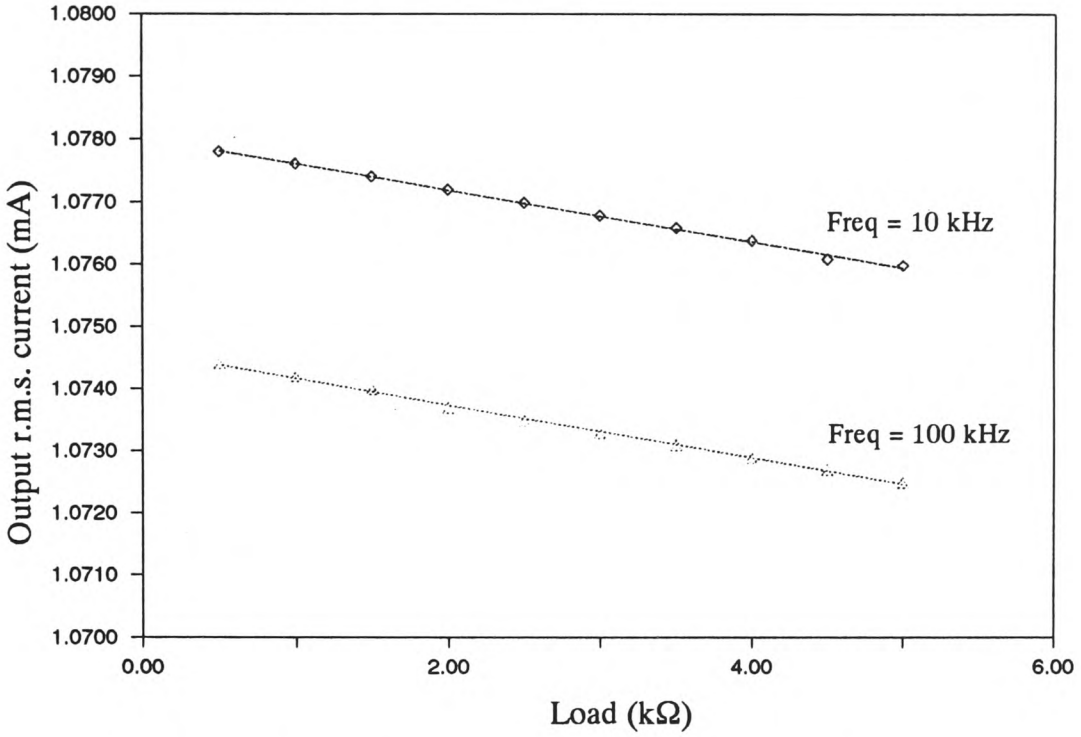


Figure 3.13 Simulation results of Leung's current source with nominal component values

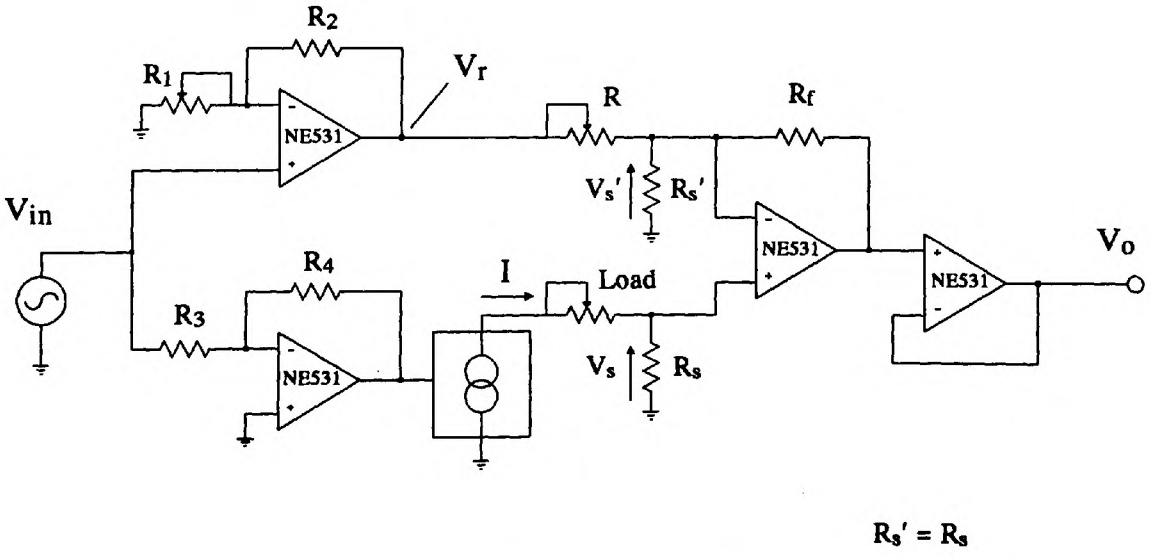


Figure 3.14 Test circuit

integration. As the circuit consists of twelve transistors and two diodes, the die size of the integrated circuit will be very small, provided that all passive components are connected externally. In fact, the circuit (Figure 3.8, p.49) can be further simplified by the removal of the two diodes (D_1 and D_2) and the two small resistors (R_{s1} and R_{s2}) since they are not essential to the current source. If the circuit is fabricated onto a monolithic integrated circuit, a better performance is expected as V_{BE} and β are likely to be more closely matched, and all transistors will possess the similar thermal variations. (ii) the mean output impedance was found to be of the order of MΩ within the specified frequency range, which was considered to be adequate in EIT applications.

3.2.2 Practical Development

The circuit was translated to printed circuit board (p.c.b.) using the RACAL CADSTAR p.c.b. design package. An Eurocard format was adopted with two current sources accommodated per board. Hence, eight circuit boards were produced for the proposed sixteen-electrode data collection system. The source was constructed and tested with the circuit illustrated in Figure 3.14. This circuit uses a null measurement technique to produce a voltage representing the corresponding change of current for different loads and frequencies. The principle of operation is to match the "sensing" voltage V_s developed across a resistor R_s in the Figure with a reference voltage V_r so that a null voltage results at the output of the test circuit, i.e. $V_s' = V_s$ and $V_o = 0$. Once the best voltages matching is established, the load can then be altered. For an ideal current source, i.e. infinite output impedance, no voltage variations at the output of the test circuit should be detected when the load changes. In practice, an increase in load results in a decrease in output current. A resultant anti-phase voltage is obtained at the output of the circuit, since V_s' is slightly greater than V_s. From these variations, the equivalent output impedance of the current source under test can be evaluated using Equation 3.9. Since the expected changes of currents are very small, the test circuit must have sufficient gain (of the order of thousands) in order to produce measurable signals. The relationship between the variables in Figure 3.14 can be expressed in the form shown below.

$$I = \frac{V_o R + R_f V_r}{R_s R_f + R_f R + R_s R} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3.9)$$

If a 1 mA r.m.s. current is generated from the source and R_S = 10 Ω, a voltage of 10 mV is sensed. To measure a 0.1% constancy the circuit must be able to detect the smallest possible change which is, in this case, 1 μV. R_f = 10 kΩ was therefore used. The value of R_S cannot be too large due to the problem of Johnson noise. If the bandwidth is limited to 1 kHz with a suitable band-pass filter, then the voltage noise level was calculated as approximately 0.01 μV at 300 °K for R_S = 10 Ω. Note that this method also takes phase variations into account, hence it is used for evaluation of output impedances instead of output resistances.

Using the test circuit mentioned above, sixteen current sources were tested. The results are shown in Table 3.2. Referring to Table 3.2, phase variations were also measured. These were done by comparing the phase difference between the drive voltage signal and the output voltage for a fixed load of 1 kΩ. The mean value of phase difference was evaluated as approximately 6.8°, which was due to the characteristics of the complementary emitter follower. When comparing the voltage at the output of the emitter follower with the output voltage of the current source, no significant phase shift (within 1°) was detected at 10 kHz and 100 kHz by using an oscilloscope. Additionally, it can be seen from Table 3.2 that the phase variations between circuits are within 2° at 100 kHz. The phase shift is independent of load and therefore does not indicate a shunt capacitance in the output of the circuit. This phase shift could be cancelled out by shifting the input drive voltage in the hardware, or cancelled in the software.

The output impedances of the best circuit, as evaluated from Figure 3.15, were 1.1 MΩ and 2.1 MΩ at 10 kHz and 100 kHz respectively over the specified load range. However, an output impedance variation of 12.9% (mean = 830 kΩ and SD = 100 kΩ) at 10 kHz was found over the batch of sixteen circuits. A variation of 2.6% (mean = 2.0 MΩ and SD = 50 kΩ) was apparent at 100 kHz. The discrepancies between the simulation and the practical results may only be explained by frequency dependence in the transistor parameters such that the transistors became more closely matched at the higher frequency, the sources approaching their theoretical performance. In addition, it was also found that the output current fell by a mean value of 0.11% between 10 kHz and 100 kHz with a standard deviation of only 0.02%.

This work has been accepted for publication [64].

Circuit	10 kHz Z_{10} (kΩ)	θ	100 kHz Z_{100} (kΩ)	θ
1	974	≈0°	1901	≈6°
2	851	≈0°	2035	≈7°
3	702	≈0°	2087	≈7°
4	832	≈0°	2080	≈8°
5	911	≈0°	2053	≈7°
6	736	≈0°	2117	≈6°
7	874	≈0°	2023	≈6°
8	867	≈0°	1987	≈6°
9	711	≈0°	2048	≈6°
10	624	≈0°	2066	≈7°
11	738	≈0°	2013	≈8°
12	1068	≈0°	2060	≈7°
13	804	≈0°	2119	≈8°
14	854	≈0°	2053	≈7°
15	869	≈0°	2007	≈7°
16	787	≈0°	2065	≈6°

Table 3.2 Practical results of sixteen Leung's current sources

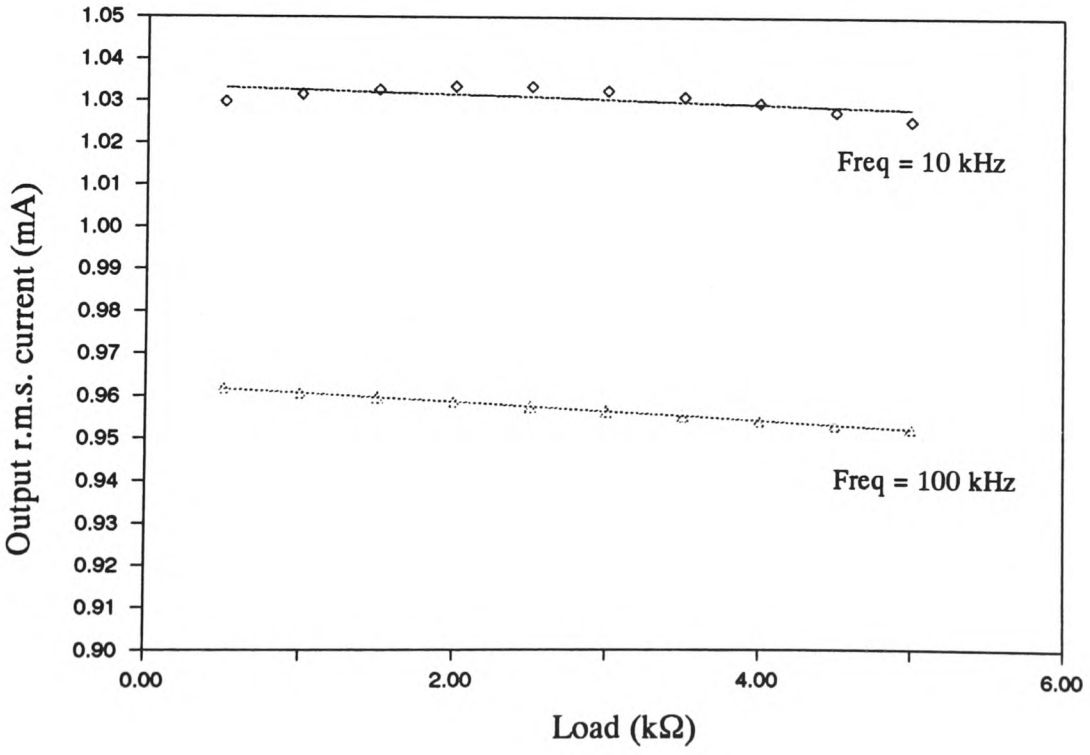


Figure 3.15 Practical results of Leung's current source :
best circuit among the sixteen current source samples

3.3 Discussion

To conclude, according to the simulation results shown in section 3.2.1, the performance was consistent across the frequency band for a perfectly matched current source. However, it is no longer true when parameter tolerances were introduced to the circuit. This is how the circuit is expected to behave in the real situation. In practice, from the testing of the sixteen current sources, the results indicate that the mean performance of the sources is better at the high frequency end. This is an abnormal phenomenon experienced in these particular transistor circuits. Due to the presence of frequency dependent parasitic components inside the transistors, it is usually expected that the performance will be degraded when operating at higher frequencies.

The result of the statistical analyses showed that this circuit constructed with discrete components is not ideal for use in this multiple drive application. However, it was thought that the multiple drive configuration would provide better results in studies of the relative permittivity imaging. Thus, this type of circuit was adopted in the author's system and has been evident to perform well. It is however expected that fabrications of this circuit on a monolithic integrated circuit should have a higher consistency in terms of their performances since they were fabricated simultaneously in the fabrication process on the same silicon wafer.

Chapter 4

Design and Development of the Polytechnic of Wales EIT System

The primary objective of this programme was the design and development of an EIT data collection to produce relative permittivity images of test objects and human subjects. Since other data collection systems being developed elsewhere are concerned with conductivity imaging only, it was therefore necessary to design a system which would be capable of extracting phase information as well as magnitude information from the peripheral voltages. With such data, the distribution of resistivity (or conductivity) and relative permittivity of the test objects could potentially be evaluated and displayed. As mentioned in chapter two, the minimisation of stray capacitance effects in the drive circuitry was considered to require a multiple current drive configuration. In the light of the various data collection systems reviewed, it was considered that the front-end architecture adopted in the OXPACT system would be a good starting point for the author's work. The front-end design of the author's system was thus based upon the OXPACT system with some modifications being made to satisfy the needs for relative permittivity imaging.

It was considered that the data collection system should be a stand-alone versatile apparatus and be able to communicate with a personal computer, so that it could be used as a basis for a variety of investigations in the future. This suggested that it should be a programmable system employing a microprocessor controlled front-end. The system should also be able to perform both "static" and "dynamic" imaging although "real-time" imaging would not be considered in this particular investigation. Since "real-time" imaging was not intended, a serial data collection method was considered to be adequate. However, voltage measurements at a relatively high speed were considered necessary since cardiac gated imaging was identified as a possible application. In such an application, a system capable of collecting data at a rate approaching ten frames per second was deemed necessary, although initially system capable of collecting two frames per second was considered to be adequate. The capability to operate across a wide range of frequencies (from 10 kHz to 80 kHz) was also considered necessary although imaging relative permittivity does not necessarily require multi-frequency operations. Implementations of this capability would provide a basis for future

investigations of other aspects of impedance tomography, such as dual-frequency imaging. In addition, since poor signal-to-noise ratios are usually obtained for the smallest voltage signals present in EIT, facilities for signal averaging would be of benefit. This implied that the data collection system itself required data storage.

The system is henceforth referred to as the Polytechnic of Wales (POW) EIT System, and is detailed below.

4.1 Principle of Operation of the Polytechnic of Wales EIT System

With reference to Figure 4.1, the data collection system is linked to a host computer via a data communication line. Thus, the complete imaging system can be broadly partitioned into two major functional systems viz the host computer data handling/processing system, and the data collection system. The work carried out on the host computer required software development; the data collection system required hardware and software developments due to the microprocessor control intended.

The host computer acts as a user/machine interface to initiate each operation. The interface is in fact a computer program, which runs on a standard IBM PC, developed by the author specifically for this application. Communications between the data collection system and the host computer is established by means of a RS232C data link. By employing this communication standard, the data collection system becomes machine independent because other types of computers may also be used provided that the same protocols are adopted. The major roles of the computer are: (i) to send codes to control the data collection system, (ii) to process the received measurement data in order to evaluate the distributions of conductivity and relative permittivity of test objects, i.e. to reconstruct images.

The data collection block is an intelligent data gathering system under microprocessor control. A typical microprocessor architecture is thus employed. With reference to Figure 4.1, instruction sequences are stored in the EPROM. The microprocessor executes these instructions at the hardware level. The setting of the magnitude and frequency of the applied stimuli, the selection of drive and measurement electrode pairs, the initiation of A-D conversions are examples of operations executed. The converted data are temporarily stored in the static RAM, pending transmission to the host computer.

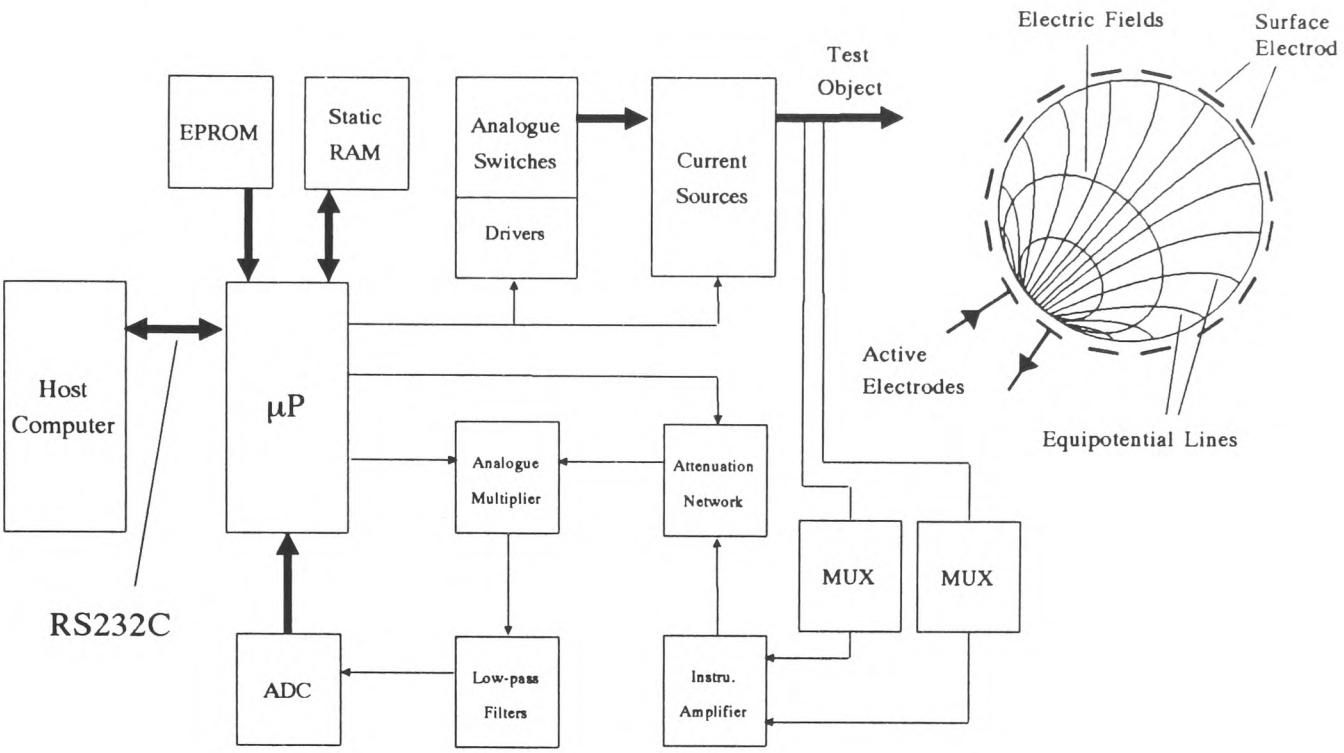


Figure 4.1 Architecture of the Polytechnic of Wales EIT System

Figure 4.2 The Polytechnic of Wales EIT System

The basic operation entails the selection and activation of a pair of adjacent current sources which, as a sink/source pair, cause a current of the order of 1 mA, at say 40 kHz, to flow in the object. This current generates an electric field which results in voltages being developed at all the other undriven electrodes. These voltages are measured differentially by means of an instrumentation amplifier via two analogue multiplexers which are used for the selection of a measurement electrode pair. In order to prevent overload and over-range problems, a process of signal attenuation is carried out intelligently by the microprocessor. The measured voltage signals are multiplied with two reference voltage signals in the two analogue multipliers. The outputs of this multiplication process provide the data necessary to reconstruct an impedance map in the host computer. The in-phase component represents the conductivity and the quadrature component represents relative permittivity. The required components are obtained from the outputs of low-pass filters. The d.c. signals are then converted into a digital format for transmissions to the host computer. To reconstruct an impedance image the stimulus is successively applied to each pair of electrodes in turn, and the resulting sets of data are transmitted to the host computer. A 12-bit A-D converter is used in order to avoid accuracy deterioration in data conversions.

To summarise, the microprocessor handles four major operations. They are :-

(a) setting the amplitude and frequency of the stimulus currents,

(b) performing measurements of electrode voltages,

(c) organising buffered data storage,

(d) effecting data and control communications between the EIT front-end and the host computer.

The following texts details the design and operational aspects of the system produced, a photograph of which is shown in Figure 4.2.

4.2 The Data Collection System

The production of the data collection system involved digital and analogue circuits design. The system can be partitioned into two functional units. One contains

mainly the digital electronics while the other contains the analogue electronics. They are dealt with separately in turn.

4.2.1 Digital Electronics Circuitry

A detailed block diagram of the digital part of the data collection system is illustrated in Figure 4.3; a complete circuit diagram is shown in Appendix G (G.1, p.212). An operating code is used to identify different operations throughout the following descriptions. A complete list of the codes and their defined operations is shown in Appendix B. To request an action the corresponding operating code must be transmitted from the host computer to the Receive Shift Register in the microprocessor μP via a serial data link RS232C (Figure 4.3). The microprocessor then interprets the received code and executes the requested operation. Detailed descriptions of the program are documented in section 4.4.

An 8-bit microprocessor (HD63B03RP) was chosen as the central processing unit. This incorporates an internal 128-byte static memory, for fast data manipulations, and a serial communication port. Two built-in I/O ports provide an easy means to interface it to other peripheral devices. With a quartz crystal of 4.9152 MHz, it provides a serial data transmission rate of 9600 baud.

An 8 kilo-byte eprom (M2764AFI) is used to store all the operation sequences while a 32 kilo-byte static ram (MS62256) provides the data storage. The address decoder was implemented with a PAL device (PAL20L10) which not only reduced the number of standard logic devices but also increased the potential reliability of the decoder as less circuit wiring and soldering joints are needed. In addition, signals within an integrated circuit have a much smaller chance to pick up electrical noise. A memory map for use in programming the address decoder is shown in Appendix C.

A drive electrode pair is selected by a 4-to-16 digital decoder (74LS154) while the measurement pair is selected by two 16-channel analogue multiplexers (HI506). Two 8-bit digital latches (74LS373) are used to hold the electrode selection codes in both cases. The frequency and magnitude of applied stimuli are selected by loading the corresponding code to another 8-bit latch (74LS373).

The serial interface comprise two voltage level shifting devices (DS14C88 and DS14C89) which are particularly designed for interfacing TTL and RS232C

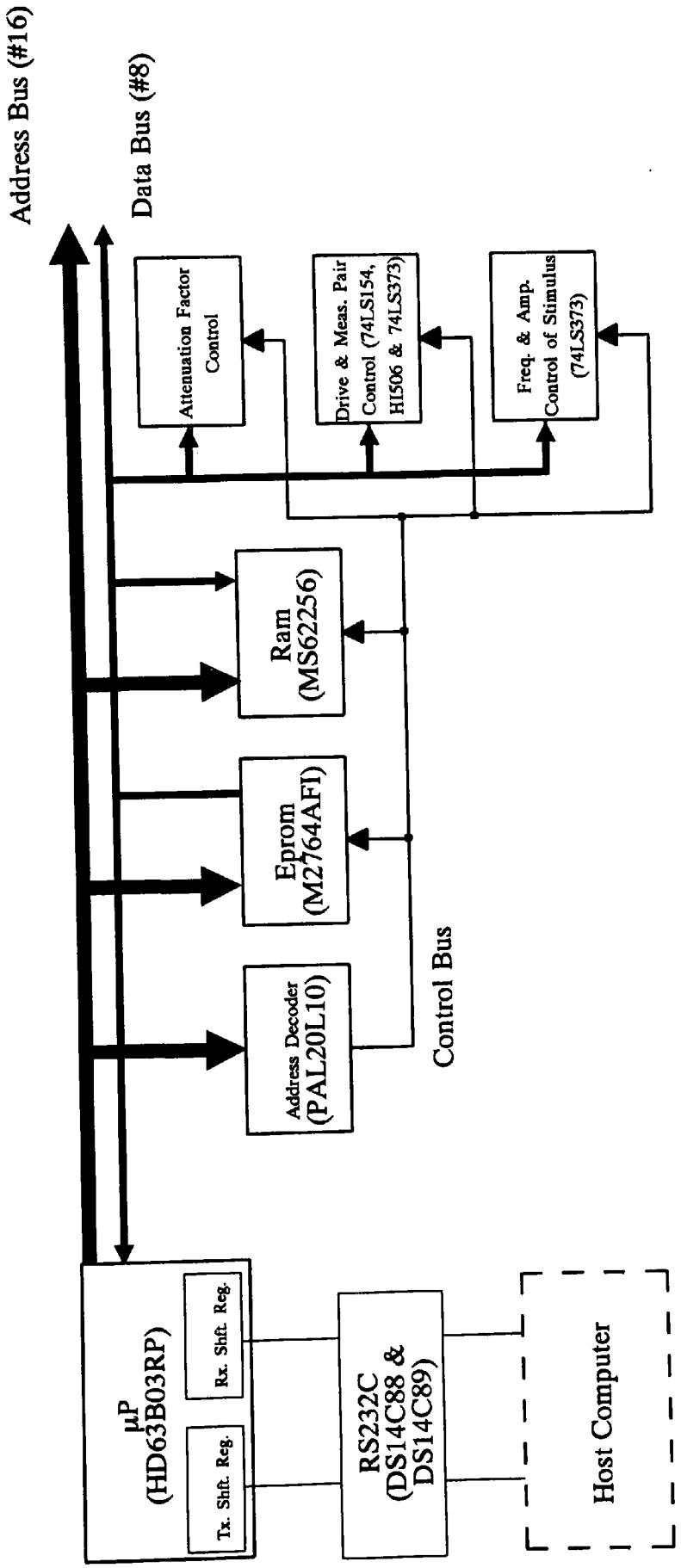


Figure 4.3 Digital part of the data collection system

signals. Note that logic '1' and '0' represent 5 V and 0 V in TTL devices respectively while −12 V and +12 V in RS232C standard respectively [72].

4.2.2 Analogue Electronics Circuitry

The analogue circuitry is illustrated in the block diagram in Figure 4.4; a complete set of circuit diagrams is shown in Appendix G. With reference to Figure 4.4, the voltage signal generator is used to provide sinusoidal and cosinusoidal signals, the frequency of which is controlled by the user via the user interface in the host computer. Since they are produced digitally, they thus need to be low-pass filtered in order to remove high frequency components (quantisation noise). The filtered sinusoidal signal is used as the control voltage drive signal for the current sources. An inverted version of the drive signal is needed in order to produce a sink/source action in a pair of current sources. Only one pair of adjacent current sources is activated at any time and operates as a sink/source current pair to drive the electrodes. The voltages developed at all other electrodes are recorded sequentially via the two 16-channel analogue multiplexers. The differential peripheral voltages between any two adjacent electrodes are sensed with an instrumentation amplifier. Each differential signal may be attenuated under microprocessor control depending upon its associated attenuation factor. Two stages of signal amplification then take place after the attenuation process. The total gain of these two amplifications is approximately 100. The role of the demodulation process is to extract the real and imaginary components of the signal, and is achieved by multiplying the sensed voltage signals with reference drive voltage signals. The demodulated signals are low-pass filtered to remove the double frequency components. These real and imaginary components are now in the forms of d.c. voltages and are digitised in the 12-bit A-D converter. The digital data are stored in the static ram and may then be transmitted to the host computer.

Having described the principle of operation of the front-end electronics generally, it is now necessary to discuss each of the individual functional units in detail. The discussion follows the flow of the signal.

(a) Voltage signal generator and four-pole low-pass filters

A circuit diagram of this generator is shown in Figure 4.5. The circuit is similar to that employed in the University Hospital of Wales EIT system [10]. The master

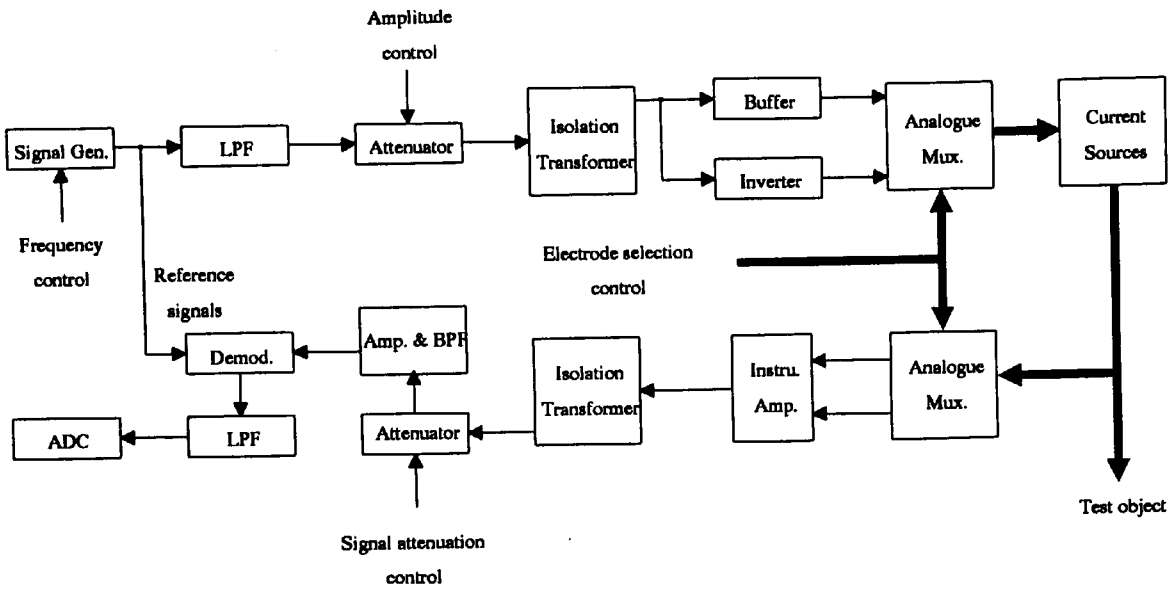


Figure 4.4 Analogue part of the data collection system

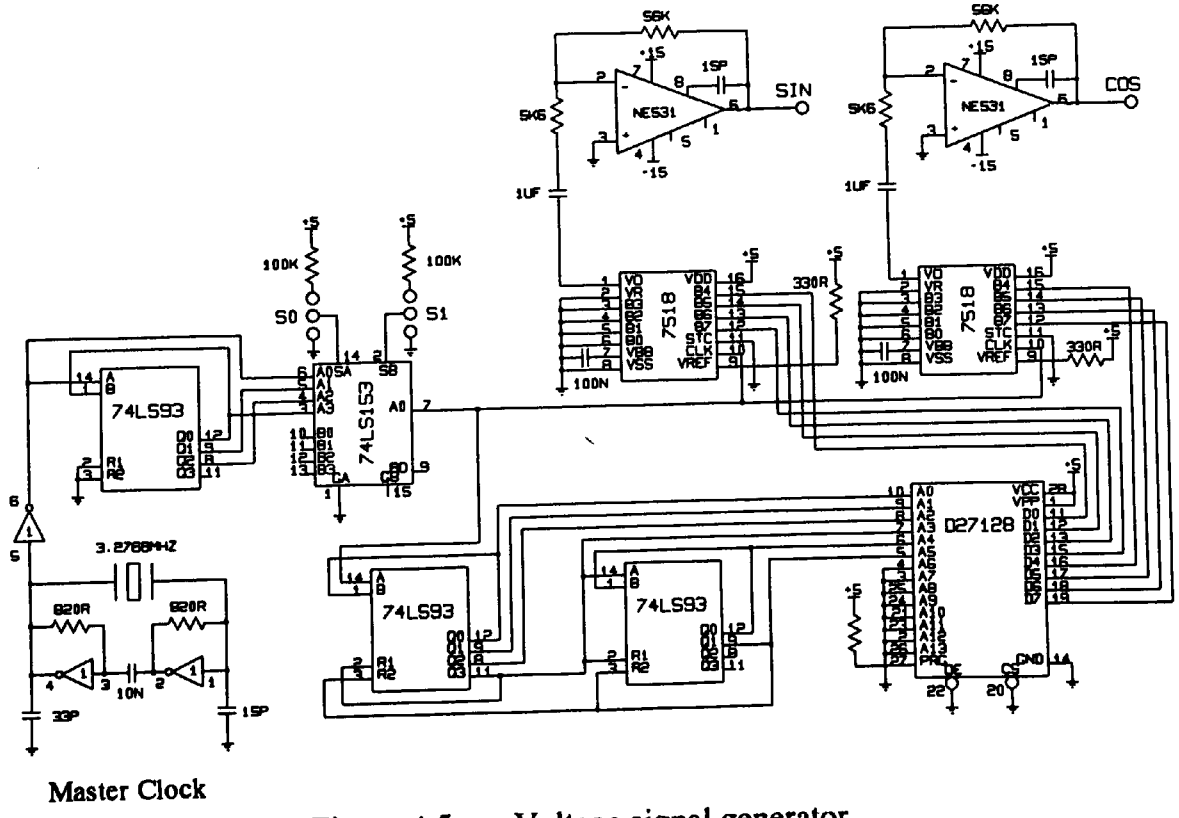


Figure 4.5 Voltage signal generator

clock is a crystal oscillator circuit running at 3.2768 MHz. Four distinct frequencies of 10.24 kHz, 20.48 kHz, 40.96 kHz and 81.92 kHz are derived from this clock. The selection code for these four frequencies is shown in Table 4.1.

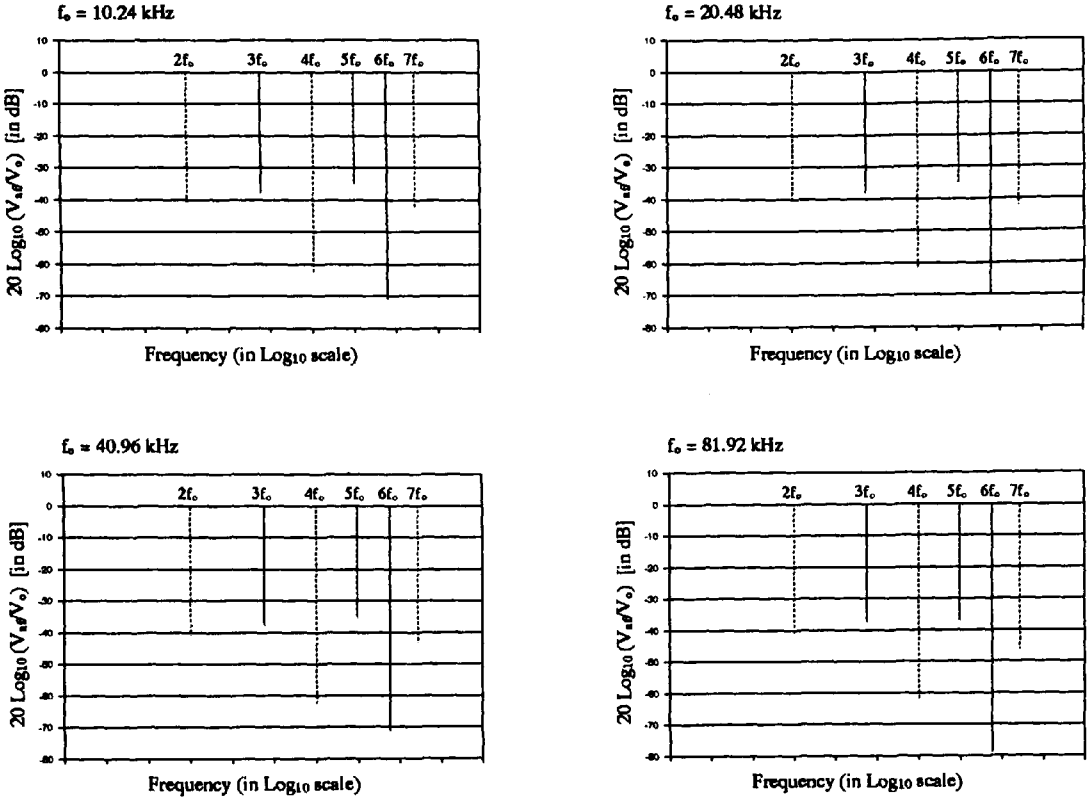
Selection code (in binary)	Frequency (kHz)
00	81.92
01	20.48
10	10.24
11	40.96

Table 4.1 Frequency selection codes

A 4-to-1 digital multiplexer (74LS153) is used to select one of the above frequencies. The selected frequency is applied to a set of two binary counters (74LS93). The outputs of these counters are used as address signals for an eprom (D27128) which contains a look-up table of sinusoidal and cosinusoidal signals. Each complete cycle of the signal is sampled 40 times and an amplitude resolution of 4 bits or 16 voltage levels is used. To image relative permittivity the effect of variation of the phase is more important than the variation of amplitude of the imaginary signals. Therefore a high sampling rate of 40 samples per cycle is used. The amplitude resolution is less important and it was considered that a four-bit amplitude resolution may prove adequate. Since the amplitude variation is not 6.25% which is a figure representing the missing code situation, instead it is dependent on the quality of the D-A converter (PNA7518). The output accuracy is ±1/2 LSB of the input. The constancy of phase is mainly dependent on the quality of the crystal. According to the figure quoted from RS Component Ltd., the crystal has a frequency tolerance of 10 ppm (i.e. 0.001%) at 25 °C. With this high quality crystal, it is acceptable in this application.

From the experimental spectral analyses, the amplitude of the second harmonics of the signals in all cases was found to be about 40 dB down on the fundamental as illustrated in Figure 4.6(a). This indicated that a four-pole low-pass filter would be required in order to reduce the amplitude of these harmonics. An active Butterworth low-pass filter implemented with the Brenan-Bridgeman configuration was used (Appendix G, G.3, p.214). If these reference signals are not filtered before the demodulation process, it is necessary to use band-pass filters elsewhere to

(a) Before filtering



(b) After filtering

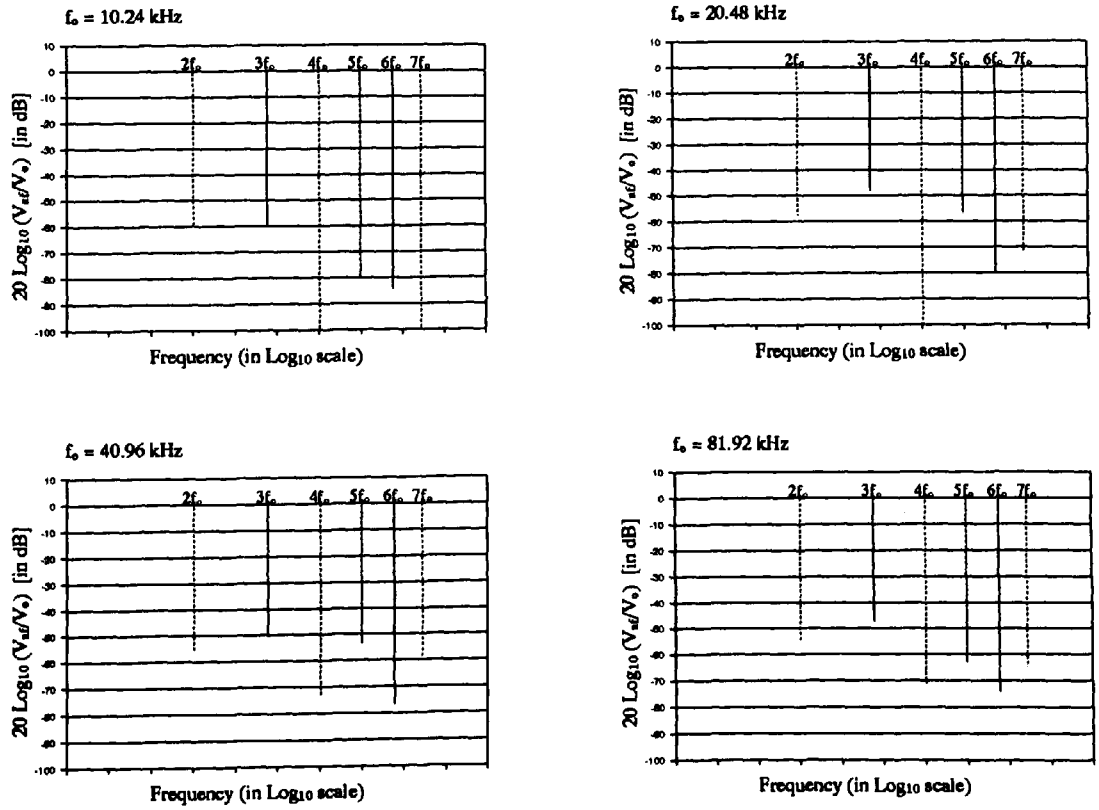


Figure 4.6 Spectrum analyses of the voltage drive signals

remove the high frequency components. Filtering the reference signals before utilising them in the data collection system could possibly improve the noise performance of the system. The higher harmonics signals may produce significant systematic errors in the demodulation process. It is because all harmonics present in the reference signal are also present in the sensed signals. If the band-pass filter does not reduce the amplitude of these harmonics significantly (below –60 dB), an error d.c. component signal would be produced. Figure 4.6(b) shows the spectra of filtered signals at the four frequencies.

(b) Voltage driver for the current sources (Appendix G, G.4, p.215)

The magnitude of the output current is controlled by the voltage drive signal. The aim of changing the magnitude of the current is to deliver the possible maximum current (within the safety limit [35]) to an object in order to obtain a bigger peripheral voltages, hence to improve the signal-to-noise ratio. To vary the current level the drive signal needs to have different amplification factors. The voltage signal is first attenuated by a set of known factors and then amplified with a constant gain. These processes must be in the correct order otherwise voltage clipping may occur in the secondary of the high frequency pulse transformer which is used for safety isolation purposes. Since the output voltage time product of the transformer used in the circuit is quoted as 200 Vμs, a maximum of 2 V pk at 10 kHz is the limit or 4 V pk at 20 kHz. Voltages which exceed these limits result in clipping.

The attenuation process is implemented using a simple passive voltage divider. A signal of the desired attenuation is selected by using the analogue multiplexer (DG508). After the isolation transformer, the signal is amplified. This signal needs to be inverted in order to produce an anti-phase drive voltage signal for the current sources. A buffer is also used for the in-phase signal to keep a minimum phase deviation between these two signals. The total voltage gain after the attenuation is about three (AD711(U2), Appendix G, G.5, p.216).

(c) Current sources selection (Appendix G, G.5, p.216)

Sink/source current source selection is achieved using a multiplexer comprising an array of analogue switches (DG211). The control of the switches is by a 4-to-16 line decoder (74LS154, Appendix G, G.7, p.218). Note that two 16-channel multiplexers may be used in lieu of the analogue switches. Isolation of digital

signals is achieved using opto-isolators (ILQ74, Appendix G, G.7, p.218) which have an approved isolation voltage of at least 1500 V.

(d) Measurement electrode pair selection (Appendix G, G.6, p.217)

The peripheral voltages are buffered by dc-coupled voltage followers. These buffered signals are in turn selected by two 16-channel analogue multiplexers (HI506) and the selection sequence is controlled by the microprocessor. The followers are implemented with a dual precision high speed BiFET input operational amplifier (AD712, Appendix G, G.10, p.221) which has a high input impedance of 3×10^{12} Ω in parallel with 5.5 pF. It exhibits extremely small input bias current and input offset currents of 25 pA and 10 pA respectively. It is vitally important to keep these currents small in EIT applications as they influence the peripheral potentials. Since the input impedance of these amplifier contains capacitive components, and stray capacitances are always present at unscreened electrodes, a study of the capacitance effect on the variation of the output current was carried out which was based on a current source with an output impedance of 1 MΩ. The equivalent circuit for this analyses is shown in Figure 4.7 and the result is shown in Figure 4.8.

In the presence of a total capacitance (C_x) of 1 pF, the deterioration of the output impedance of a current source is small across the EIT frequency range. However, a 10% drop in the output impedance is calculated if a total capacitance of 10 pF is present. Although the input capacitance of the operational amplifier cannot be avoided, a common mode voltage screened electrode can reduce the stray capacitance effect, see chapter two (pp.21,25).

In the Leuven meeting, some groups have pointed out that dc-coupled voltage followers did not incorporate safety precautions. Under a normal condition, it is anticipated that only a minimal current (of the order of pA) would flow in the human subject due to input bias currents. However, in case of faulty devices, unconditional flow of current may result which is not allowed.

(e) Instrumentation amplifier (Appendix G, G.6, p.217)

A standard three-VOA instrumentation amplifier is used to perform voltage measurements. The CMRR ratio was found to be at least -60 dB within the EIT

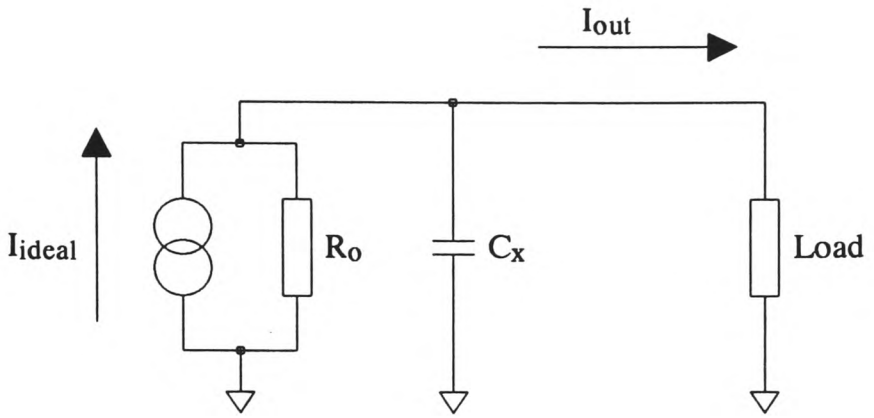


Figure 4.7 Equivalent circuit for studies of the stray capacitance effects

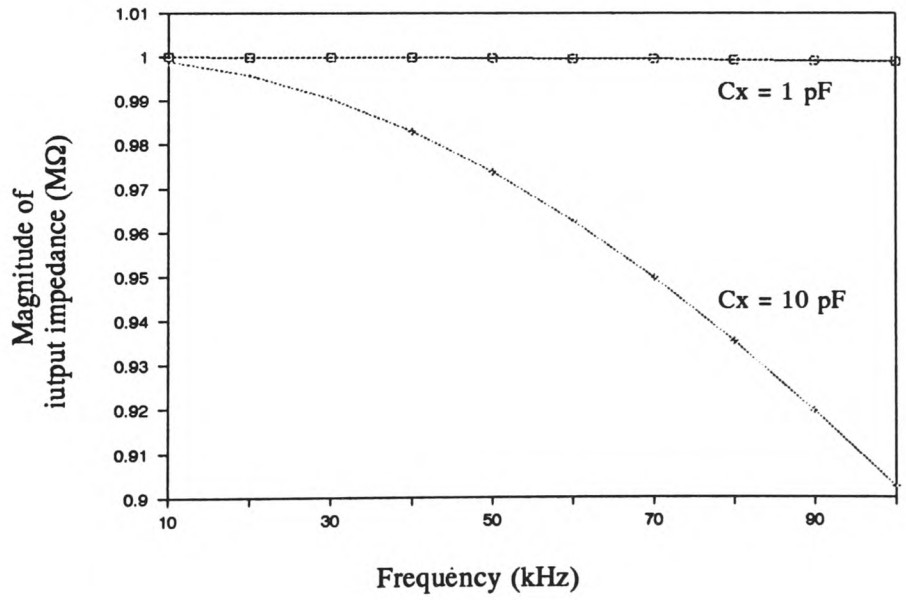


Figure 4.8 Variations of equivalent output impedance with frequency due to stray capacitance

frequency range and no measurable phase shifts were detected when the device was trimmed properly. The CMRR for different input voltage signals at various frequencies is shown in Table 4.2.

Frequency (kHz)	CMRR in dB 5V	10V	
10	—	—	
20	—	-73	where '—' means
40	-64	-68	signals are not
80	-62	-66	measurable

Table 4.2 CMRR of the instrumentation amplifier

(f) Signal attenuation network (Appendix G, G.8, p.219)

At this stage, signals are returned to the computer circuitry, so that the output signal from the instrumentation amplifier requires electrical isolation which is provided by another pulse transformer. In order to maintain the system's sensitivity to 0.1% changes, the smallest sensed voltage needs to be amplified to span the full scale input range of the A-D converter. Since the ratio of the maximum and minimum peripheral potential differences between adjacent electrodes is approximately 40:1 [6], the ratio of the system gains between the maximum and minimum potentials is therefore 1:40.

Suppose that the maximum potential difference between the two adjacent electrodes is 5 V. The expected minimum potential is 0.125 V. When a 12-bit A-D converter with an input span of ±5 V is used, the accuracy of the full scale input is 0.05%. For the minimum potential, however, the accuracy reduces to 1.95% which is not adequate. In order to resolve the minimum potential to a 0.1% accuracy with a constant system gain, a 17-bit A-D converter would be needed; this provides an accuracy of 0.06%. To minimise cost and maintain a low conversion time, the use of different system gains for different potentials provides a better solution.

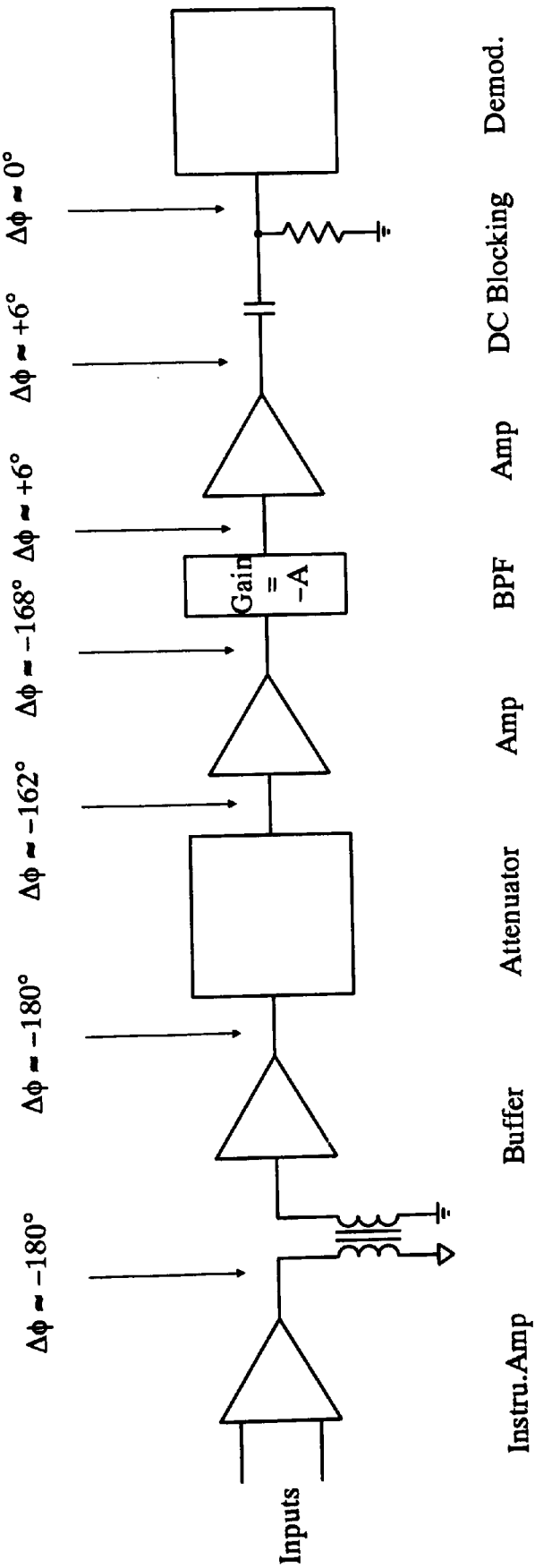
In order to obtain different gains, the best method is to use a high system gain and then to attenuate the amplified signals so that all signals span the input range of the A-D converter.

To implement such a mechanism a R-2R resistor network was used as a signal attenuation network. Thus, the input voltage signal could be gradually attenuated along the resistor chain by a factor of two. By applying an attenuation factor code at an 8-bit analogue multiplexer (DG508), the desired attenuation was obtained. It was very important to maintain a zero phase difference between the reference drive voltage signal and the sensed voltage signal at the input of the analogue multiplier (AD534, Appendix G, G.9, p.220). Phase shift introduced by the electronics cannot be separated from the actual phase displacement caused by the object reactances. There are two methods to be used to maintain a minimum phase difference between these two signals. One is to ensure a zero phase variation of the signal in all front-end electronics; this is a laborious task especially for a circuit involving a large amount of electronics devices. The other is to maintain a zero phase difference between these two signals by means of a band-pass filter. This was the method adopted in the author's system. The amount of phase shift introduced by the electronics at 40.96 kHz was measured at various points in the measurement path and is shown in Figure 4.9. The roles of the band-pass filter are discussed in (g) below.

According to the figures, a significant amount of phase shift was found to originate from the R-2R resistor network. The analogue multiplexer in the attenuator only introduced a negligible phase shift. The variation of phase at different attenuation factors is small (within 3° at 40.96 kHz). This indicates that the Polytechnic of Wales EIT system may not currently be suitable for multi-frequency imaging as the phase attribute is frequency dependent. However, it is possible to perform relative permittivity imaging as only one frequency is required, provided that the alignment of the reference and the sensed voltage signals is done properly.

(g) Voltage amplification and band-pass filter (Appendix G, G.8, p.219)

In the early stage of the development work, wide-band system operation was projected. In tests with a system gain of about 100, the system experienced no difficulties in acquiring data at the four frequencies (10.24 kHz, 20.48 kHz, 40.96 kHz and 81.92 kHz). Although the gain of 100 is achievable with one amplification, it is more practical to amplify the signal in two stages. The gain of the first amplification is about 3 while the second amplification is about 33. To perform *in-vivo* measurements an overall system gain (defined as the gain increases from the output of the instrumentation amplifier to the input of the analogue multipliers)



Δφ = Ref. voltage signal – Phase at pointed position

Figure 4.9 Phase shift at various points in the measurement path (40.96 kHz)

of the order of thousands is however required. Two-stage amplification, gain of 34 was used in this amplification circuit.

An R-L-C band-pass filter with Q-factor of 10 was tuned at about 40.96 kHz. The filter is used to avoid any un-wanted signals being amplified in the second stage of amplification. If this is not done, the system becomes potentially unstable as all other frequency components are also amplified which could cause oscillations.

As mentioned in (g), the tuning of the band-pass filter is critical. To properly tune the system to operate at the desired frequency it is necessary to compare the phase difference between the reference and measured signals with an oscilloscope. For a purely resistive load, the reference sinusoidal signal, applied stimuli and the measurement voltages should be in-phase. Any phase displacement between the reference and measurement voltage signals must be due to the different phase/frequency responses in the electronics. If the filter is correctly tuned, only a slight variation of filter gain is observed at the centre frequency but the phase is very sensitive to frequency variations at this point. Thus, adjusting the centre frequency of the band-pass filter will have no significant gain variation but may be able to align the phases of the reference and the measurement voltage signals.

When a large system gain and wide-band operations are required, a number of band-pass filters of different cut-off frequencies must be used.

(h) Demodulation process - analogue multiplication (Appendix G, G.9, p.220)

The demodulation process is used to extract the real and imaginary parts of the peripheral voltage signals. It is achieved by multiplying the measured voltage with the in-phase and quadrature signals which are generated by the voltage signal generator. The multiplications are performed by two analogue multipliers (AD534).

Prior to the multiplication processes, it is necessary to ensure the input signals contain no d.c. voltages since the desired voltage signals are also at d.c.. Any offset voltages present in the sensed voltage signals cannot not be separated. To avoid these offset voltages, three passive high-pass filters of break frequency of about 4 kHz are required. For relative permittivity imaging, a signal frequency of 40.96 kHz is used. Using high-pass filters each with a 4 kHz break frequency will produce

negligible phase shift at 40.96 kHz but has attenuation of about -36 dB for 50 Hz power supply signals.

Although the input CMRR of the multiplier is about -80 dB, the resistor used in the high-pass filter must be small. However it must be large enough to avoid exceeding the output current capability of the operational amplifiers. If too large a resistor is used, the bias current of the multiplier will produce a standing voltage at its inputs and thus cause errors. For the AD534 multipliers, an input bias current of 0.8 μA is quoted. When a 10 kΩ resistor is used, a standing voltage of 8 mV results.

(i) Four-pole low-pass filter for rejection of double frequency components (Appendix G, G.9, p.220)

At this final stage, the characteristics of the low-pass filters are very critical as the data collection time is influenced by their settling times. The accuracy of recorded data is dependent on the break frequency and the number of poles. Consider that a four-pole low-pass filter is implemented by means of cascading two identical two-pole low-pass filters of the same break frequency. If the transfer function of the filters is assumed to be

$$F(S) = \frac{\omega_n^2}{S^2+2\xi\omega_n S+\omega_n^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots (4.1)$$

where ξ is the damping ratio and ω_n is the angular break frequency, the settling time of the four-pole filter is not the same as the two-pole filter. Accurately evaluating the minimum settling time involves the evaluation of the convolution integrals. The inverse Laplace transform of Equation 4.1 yields

$$f(t) = \frac{\omega_n^2}{\omega_d} \mathrm{Exp}(-\xi\omega_n t)\mathrm{Sin}(\omega_d t) \quad \ldots\ldots\ldots\ldots\ldots\ldots (4.2)$$

where $\omega_d = \omega_n\sqrt{(1-\xi^2)}$. Suppose that a unit step input signal is applied to the first second-order filter. The output of the filter can be obtained by evaluating the convolution integral

$$O_1(t) = \int_0^t \frac{\omega_n^2}{\omega_d} \text{Exp}(-\xi\omega_n\tau)\text{Sin}(\omega_d\tau)d\tau \quad \ldots\ldots\ldots\ldots (4.3)$$

Solving Equation 4.3, gives

$$O_1(t) = 1 - \frac{\omega_n}{\omega_d} \text{Exp}(-\xi\omega_n t)\text{Sin}(\omega_d t+\phi) \quad \ldots\ldots\ldots\ldots (4.4)$$

where $\phi = \text{Cos}^{-1}\xi$. If the second convolution integral is evaluated with $O_1(t)$ as the input signal, the output of the second second-order filter $O_2(t)$, i.e. the output of the fourth-order low-pass filter, can be obtained. The second convolution integral becomes

$$O_2(t) = \int_0^t h_1(\tau)h_2(t-\tau)d\tau \quad \ldots\ldots\ldots\ldots\ldots\ldots (4.5)$$

where

$$h_1(\tau) = 1 - \frac{\omega_n}{\omega_d} \text{Exp}(-\xi\omega_n\tau)\text{Sin}(\omega_d\tau+\phi)$$

and

$$h_2(t-\tau) = \frac{\omega_n^2}{\omega_d} \text{Exp}[-\xi\omega_n(t-\tau)]\text{Sin}[\omega_d(t-\tau)+\phi]$$

Decomposing Equation 4.5, gives

$$I_1(t) = \int_0^t \frac{\omega_n^2}{\omega_d} \text{Exp}[-\xi\omega_n(t-\tau)]\text{Sin}[\omega_d(t-\tau)]d\tau \quad \ldots\ldots\ldots (4.6)$$

and

$$I_2(t) = \int_0^t \frac{\omega_n^3}{\omega_d^2} \text{Exp}(-\xi\omega_n t)\text{Sin}(\omega_d\tau+\phi)\text{Sin}[\omega_d(t-\tau)+\phi]d\tau \quad \ldots (4.7)$$

Equation 4.6 yields

$$I_1(t) = 1 - \frac{\omega_n}{\omega_d} \text{Exp}(-\xi\omega_n t)\text{Sin}(\omega_d t - \phi) \quad \ldots\ldots\ldots\ldots (4.8)$$

and Equation 4.7 yields

$$I_2(t) = \frac{\omega_n^3}{2\omega_d^3} \text{Exp}(-\xi\omega_n t)\text{Sin}(\omega_d t)\text{Cos}\phi$$

$$- \frac{t\omega_n^3}{2\omega_d^2} \text{Exp}(-\xi\omega_n t)\text{Cos}(\omega_d t + \phi) \quad \ldots\ldots\ldots\ldots (4.9)$$

By combining Equations 4.8 and 4.9 and substituting $\phi = \text{Cos}^{-1}\xi$, $\omega_d = \omega_n\sqrt{(1-\xi^2)}$, and $\omega_n = 1/T$

$$O_2(t/T) = 1 - \frac{1}{\sqrt{(1-\xi^2)}} \text{Exp}[-\xi(t/T)]\text{Sin}[(t/T)\sqrt{(1-\xi^2)} - \text{Cos}^{-1}\xi]$$

$$+ \frac{t/T}{2(1-\xi^2)} \text{Exp}[-\xi(t/T)]\text{Cos}[\text{Cos}^{-1}\xi + (t/T)\sqrt{(1-\xi^2)}]$$

$$- \frac{\xi}{2[\sqrt{(1-\xi^2)}]^3} \text{Exp}[-\xi(t/T)]\text{Sin}[(t/T)\sqrt{(1-\xi^2)}] \quad \ldots (4.10)$$

where T is the time constant of the filter, the output of the fourth-order low-pass filter for a unit step input can be expressed in the form as shown in Equation 4.10.

There is no easy algebraic means to solve Equation 4.10 for a value of t/T such that $O_2(t/T)$ settles within 0.1%. Consequently, it is necessary to plot the Equation graphically by means of computer in order to locate the value of t/T. A set of results using various ξ is tabulated in Table 4.3. Referring to Table 4.3, the shortest settling time is obtained when $\xi = 0.8$ at which the filter is slightly underdamped. The choice of ξ determines the length of settling time and hence determines the data acquisition time. The magnitude of the double frequency ripples is determined by the selection of break frequency of the filter.

ξ	No. of time constant required	
0.05	231.65	
0.1	109.15	
0.2	50.41	
0.3	32.22	
0.4	23.48	
0.5	18.01	
0.6	15.43	
0.7	13.18	
0.8	11.14	(shortest settling time)
0.9	11.28	
0.999	13.02	(close to critical damping for two-pole)
0.707	13.20	(maximally flat)

Table 4.3 Minimum number of time constants to attain a 0.1% accuracy

If the four-pole low-pass filter is realised using two Butterworth second-order low-pass filters, the minimum settling time is about 14-times of the time constant since ξ is 0.707. With a 10 kHz stimulus, the lowest magnitude frequency component is 20.48 kHz after the demodulation process. To attain a 0.1% accuracy the highest break frequency of a four-pole low-pass filter is about 3.5 kHz for which the attenuation of about -60 dB is obtained at 20 kHz. The minimum settling time is calculated to be 600 μs.

However, in the author's system, the break frequency was set to 2.7 kHz in order to improve the noise performances (attenuation of about -70 dB at 20 kHz) at the expense of a longer settling time. This is not very critical to the author's system since "real-time" imaging was not intended. The required settling time was calculated as 778 μs.

4.3 The Host Computer Data Handling and Processing System

As mentioned previously, the operating system is formed by a complete suite of software. This was mainly written in Modula-2. The reason for using this language is because of its ability to access the serial port easily and to handle a variety of data forms. The languages modelling ability, i.e. to integrate a number of

sub-modules (implementation modules), is also an attractive feature. Modula-2 can also integrate executable programs which are written in other programming languages, and thus allow e.g. the importing of existing reconstruction algorithms. Another useful feature is the co-routine operation, which allows the processing of two different processes in a time-multiplex mode. This was considered to be a useful feature for the development of the next evolution of the Polytechnic of Wales EIT system with the possibility of implementing "full-duplex" communications between the host computer and the data collection system.

It was considered that the system should be menu-driven and simple to use so that only limited expertise would be required to operate the system. The software was designed in modular form as shown in Figure 4.10. The operations enclosed in the shaded boxes represent "implementation modules" and the clear boxes are procedures in the main module. Figure 4.11 shows an example screen display of the main menu and operating window.

Each functional operation shown in Figure 4.11 can be invoked by selecting its associated function key. The procedure to operate the system and hence to obtain an object image are as follows. The magnitude and frequency of the applied stimuli to be chosen by pressing <F6>. After the setting, the updated values are shown in the display window as shown in Figure 4.11. Voltage measurements are invoked using <F1>. Upon the completion of the measurements, the recorded data can be transmitted from the data collection system to the host computer by using option <F2>. The received data are temporarily stored in an array which will be overwritten during the next transmission process. They must therefore be stored onto files for later use. They can however be examined before putting onto files by pressing <F3>. If more than one frame of measurements is required, the resulting data can be stored as separate files, or averaged and stored in a single file (<F4>). The reconstruction process can now be activated by pressing <F7>. During the reconstruction process, data are first read from the system files. Reconstructing the various images such as magnitude, phase and so on is optional. The whole process is completed when displaying the reconstructed images in one of the colour monitor modes.

A DOS shell may be created by striking <Alt> and <D> simultaneously. To return to the EIT program 'EXIT' is typed in the DOS environment. This is an option providing a means for users to handle other DOS operations. To terminate the EIT program <Alt> and <X> are pressed simultaneously. It is necessary to confirm this

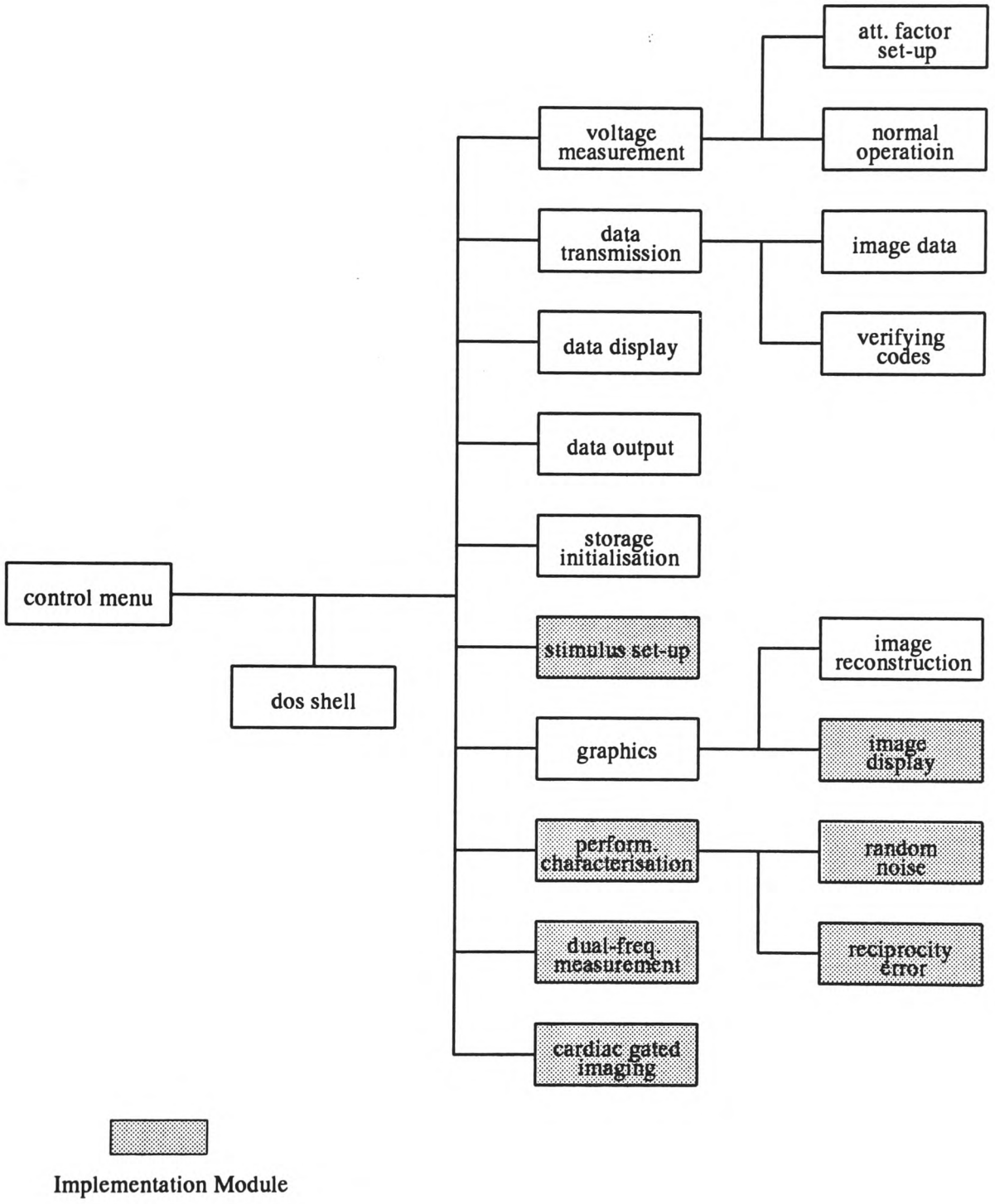


Figure 4.10 Modular structure of the host computer data handling and processing system

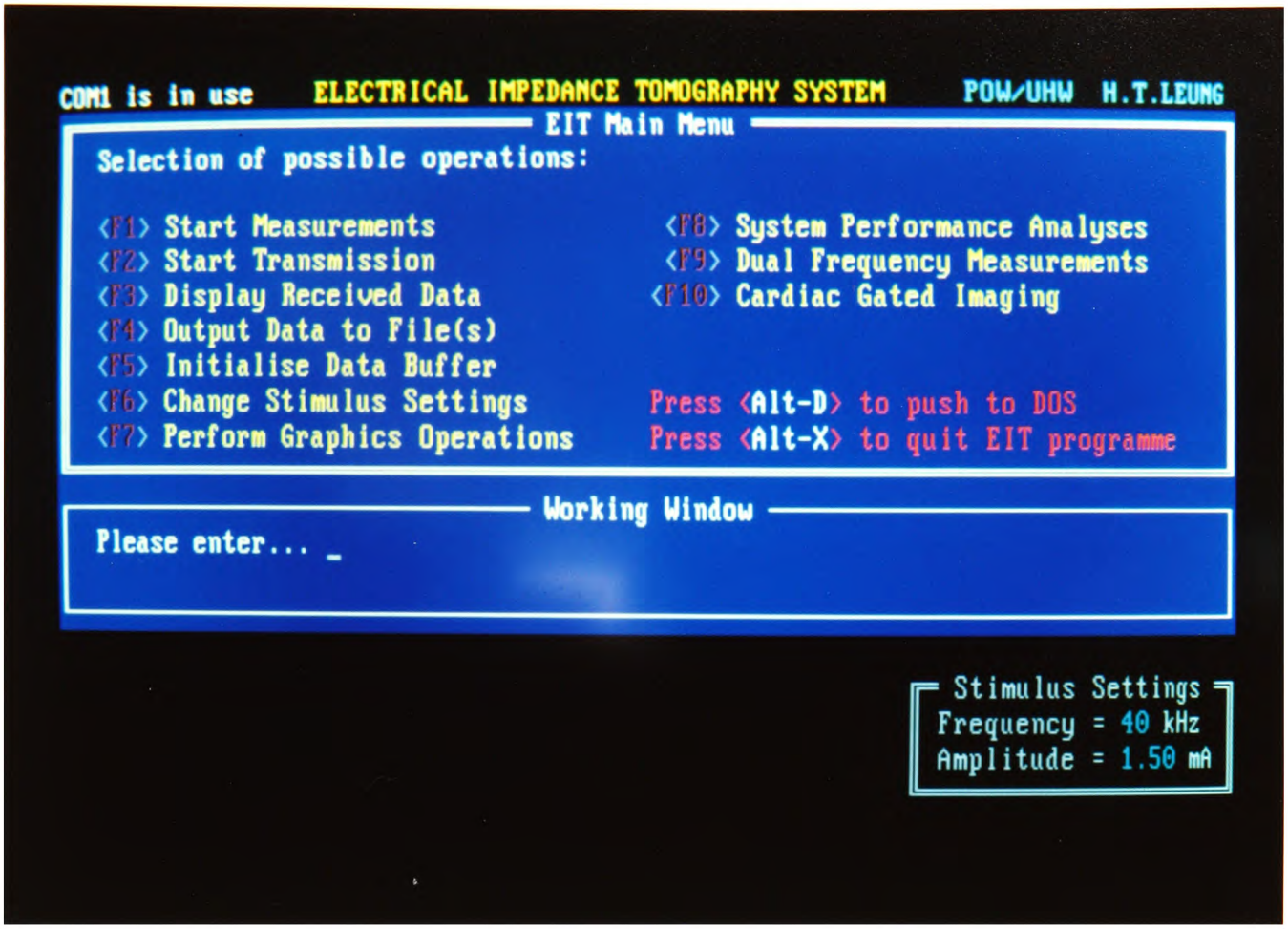


Figure 4.11 Screen format of the host computer data handling and processing system

instruction before leaving the EIT program and this is done by striking <esc>. Any other keys will return the user to the EIT program.

Having outlined the operation sequences, each functional operation and its program structure are now described individually. A set of operating codes is listed in Appendix B and the complete suite of the EIT programs can be found in Appendix D.

(a) Peripheral voltage measurements (Appendix D, D.9, p.182)

Voltage measurements start upon the transmission of the appropriate operating code to the data collection system. Upon the receipt of the operating code by the microprocessor in the data collection system, the received code will be sent back to the host computer for acknowledgement. If the transmitted and received codes are identical, the transmission is successful and voltage measurements commence. The host computer then polls the serial data receive port continuously until a completion code is detected. If the acknowledgement is negative, measurements abort.

Before the commencement of voltage measurements, it is necessary for the user first to determine a valid set of attenuation factors. This is essential for every test object to be imaged. Without an appropriate set of attenuation factors, over-range problems occurring in the data conversion processes may result which would cause difficulties in reconstructing a correct distribution of conductivity or relative permittivity. The factor is set by answering the questions prompted by the computer. The number of frames required can also be input up to a maximum of ten. This limitation is due to the memory restriction of the PC which allows a maximum of 64 kilo-bytes memory space for program variables.

(b) Serial data transmissions (Appendix D, D.9, p.182)

The data transmissions process is invoked by sending a correct operating code to the data collection system. In this process, no operating code acknowledgement is implemented. Thus, as long as no data corruption occurs during the transmission of the operating code, the code will be interpreted correctly by the microprocessor and all recorded data will be transmitted from the data collection system to the host computer. The order of data transmission is from the last measurement to the first

one. The total numbers of frames to be received are displayed for the convenience of the user. At the end of the data transmission, a completion code is received. The host computer then transmits another operating code to request the transmission of another three additional data, which are generated by processing the actual data with a mathematical algorithm. These data are used for checking the validity of the data received in the previous transmission. If all three data are consistent with those generated by the host computer, the transmission is considered to be successful. If it is not the case then data is considered to have been corrupted during the transmission process and re-transmissions are suggested.

The three additional data words used for verification in the above case are created by processing the actual measurement data with a simple algorithm as shown in Figure 4.12. The first byte data is used as a reference (Ref) and is compared with the next byte data (Data). If the next data is greater than the reference data, the reference data is modified by subtracting itself from the data, i.e. Ref = Data – Ref, and this updated Ref is named as GTRef. If the data is less than the reference data, the reference data is modified by subtracting the data from itself, i.e. Ref = Ref – Data, and the updated Ref is now named as LTRef. The reference data will not be altered if and only if the data is equal to the reference data or zero. This process continues until it finishes with the last data. Therefore, at the end of this process, three data bytes namely Ref, GTRef and LTRef are created. These can be used for data validity checking when the same process is carried out in both the host computer and the data collection system. This is a unique data checking scheme developed by the author and has advantages that it is not limited by word length and virtually no extra time is taken in the transmission process.

(c) Display of received data (Appendix D, D.9, p.182)

This option is used to inspect the raw data which are received from the data collection system. In addition to the display of actual data including real and imaginary parts of peripheral potential signals, the associated measurement parameters such as attenuation factors are also shown. Data for each frame of measurements occupy a total of sixteen screens of display. Each screen consists of a heading showing the page number, the frame number, and the drive electrode pair position. The corresponding thirteen measurements are displayed in the same order as they were collected in the data collection system. Thus, under a normal condition, a decrease in attenuation factors from a maximum value (in

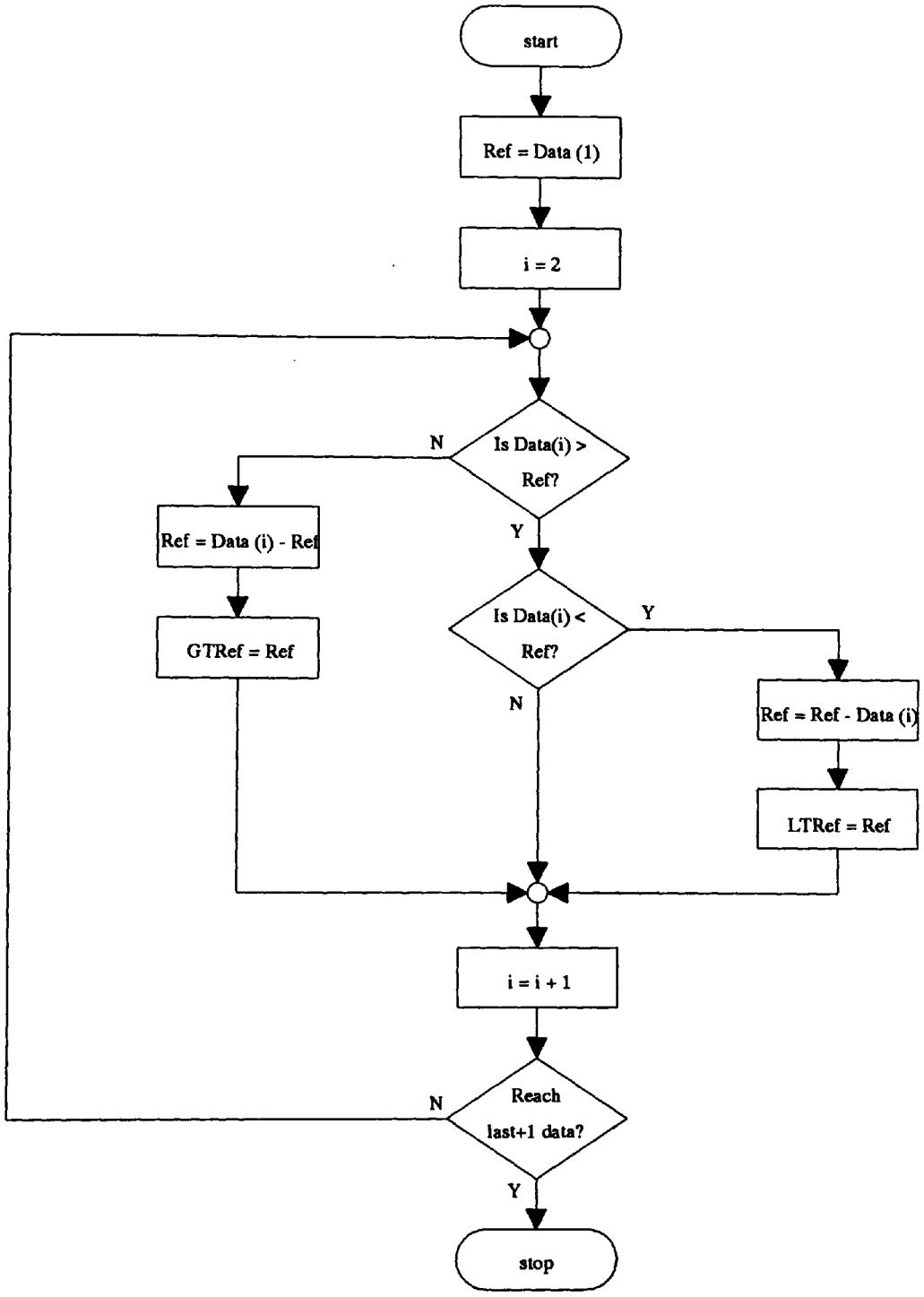


Figure 4.12 Novel data transmission checking scheme

measurements close to the drive electrode pair) to a minimum (which is usually at the opposite position of the drive pair) is expected. An upward trend then occurs after passing the minimum. Eventually it reaches another maximum value which is the measurement close to the other drive electrode. If malfunctions of some electronics occur in the drive or measurement circuitry, a display of this kind would be useful for trouble-shooting. This is the main purpose of this option.

(d) Creation of data files (Appendix D, D.9, p.182)

Having received data from the data collection system, it is necessary to save them onto files, otherwise they will be lost when the next data transmissions commence. Real and imaginary data are stored in two separate files. File names are specified by the user, but file extensions are automatically generated by the computer. The extension starts with a letter 'R' representing real part data and 'I' representing imaginary part data. In case of more than one frame data, all data can be stored separately onto different files and thus, in addition to the letters 'R' or 'I', numbers are also appended. However, storing mean values of each individual measurements is possible too. The name of each file to hold the data is displayed during the process.

(e) Initialisation of data storage (Appendix D, D.9, p.182)

This option offers a means to reset the content of the entire data storage to zero. This includes the host computer storage as well as memories in the data collection system. The principal use is to initialise all voltage attenuation factors to unity, i.e. 2^0, so that maximum system gain occurs during the first cycle of the attenuation factor set-up process.

(f) Alteration of applied stimuli settings (Appendix D, D.7, p.179)

This interactive module allows the frequency and amplitude of the current stimuli to be selected by the user. Both frequency and amplitude are adjusted in preset steps by means of the Up/Down cursor keys. The screen display shows the present set value, together with the possible alternatives. Both settings are invoked by pressing the relevant function key.

(g) Graphics operations

This option comprises two major functions; routines for invoking reconstruction algorithms (Appendix D, D.7, p.179), and image displays (Appendix D, D.9, p.182). The image reconstruction process is implemented as procedures in the main module. The algorithm described by Griffiths [11] was implemented. The display process was designed using the modelling technique. Since it is essential to use the backprojection map every time to reconstruct images, implementing the reconstruction process in the main module eases data manipulation and avoids massive data transfer from one module to another module. In order to reconstruct images, two pairs of input data files are required. One pair contains reference files of real and imaginary data. The other pair contains image files of real and imaginary object data. There are three kinds of images which can be reconstructed. These represent the distributions of the magnitude, phase and real part of the recorded object data. Conductivity and relative permittivity images can be derived from the magnitude and phase images. The method of such reconstructions has been described in section 2.4 (pp.34-7). All reconstructed images are automatically stored onto files with appropriate file extensions. There are five files containing the images; these are

* IMAGE.S : magnitude (s′)
* IMAGE.SS : phase (s″)
* IMAGE.A : alpha (α)
* IMAGE.B : beta (β)
* IMAGE.R : real part

These images can be displayed by using the image display routine. A selection of three common monitor display modes viz CGA, EGA and VGA is provided to suit different monitors. Sixteen gray-levels are employed. The maximum and minimum numerical values present in the reconstructed image are shown in the display for references only. The image is automatically gray scaled, however, manual control is also possible. A facility to make hardcopies was also developed. To control the printer to print images a dot graphics mode needs to be used. Since the "BASIC" programming language facilitates easy control of printers, a programme was therefore written in "GWBasic" and integrated to the Modula-2 program.

(h) System performance characterisation (Appendix D, D.6, p.176)

The main role of this option is to carry out system performance analyses including those relating to systemic errors, e.g. reciprocity errors, and random noise. Two quantitative methods for evaluation of these two entities will be fully discussed in the next chapter. In order to perform the analyses, two peripheral voltages data files which were created previously using the file creation option are needed. Having read the data from the files, a series of mathematical computations is performed, see chapter five. As a result, corresponding analyses provide overall values of root-mean-square and standard deviation to a resolution of 0.1%. Individual variations of data are also tabulated for inspections.

(i) Dual frequency measurement (Appendix D, D.3, p.167)

This is not an essential option in this research programme but has been implemented for future studies. This option is made similar in part to the peripheral voltage measurements routine but using two different frequencies for the applied stimuli. A correct operating code is transmitted to the data collection system. Prior to the peripheral voltage measurements, two frequencies for the reference and image data sets are defined. The program will request this information automatically. These stimuli settings are transmitted to the data collection system. The system will first perform measurements at the reference frequency. As soon as the first frame of measurements is completed, the second frequency stimuli is used for the next frame of measurements. This process allows two frames of data to be completed in less than 0.5 second. This reduces the possibility of errors caused by the possible movement of electrodes during measurements.

(j) Cardiac gated imaging (Appendix D, D.1, p.161)

The design of this implementation module was intended for use in cardiac gated imaging. The module structure is again similar to the peripheral voltage measurements. Only one frequency is used in this operation and two frames of measurements are made in one cardiac cycle. Two successive frames of measurements will start when the microprocessor in the data collection system detects a R-wave related signal from an ECG monitor. The present memory size limits the number of frames of cardiac data to five due to the memory limitation of running program on the PC. The reference and image data frames may be collected

at various intervals within one cardiac cycle. This may be set by the program inside the eprom in the data collection system.

(k) Examination of data collection electronics (Appendix D, D.9, p.182)

During the system development stage, it was considered that the peripheral voltage signals should be able to be monitored during the measurements so that the performance of the electronics inside the data collection system could be monitored. This is not possible when using the normal measurements routine since the time taken for the completion of one frame measurements is less than 0.25 second. Thus, this module allows the data collection time at each measurement position to be set by the user in order to make the signals observable. Specific drive and measurement electrode pairs may also be set up for electronics examinations. Therefore, this is a useful function for system debugging. To invoke this special operation <Alt> and <F1> keys need to be pressed simultaneously.

For example, if the measurement dwell time was set to one second, one frame of data would take 416 seconds to acquire since recording real and imaginary data requires two seconds. The data collection process may be interrupted by striking <esc>. When this routine is running, the active drive and measurement electrode pairs are shown on the screen. Each particular drive and measurement combination is denoted by a data number in the program. Thus, any combination may be set up again when the associated data number is entered in the second option.

4.4 Data Collection System Software

The data collection system was based upon a HD63B03RP 8-bit microprocessor as shown (Figure 4.3, p.66), and assembly language development was executed using a Phillips PMDS11 microprocessor development system. The program structure is illustrated in Figure 4.13. As it can be seen, the system is at "idle" for most of the time until an operation code is received from the host computer. When the system is reset, the microprocessor will be initialised. This includes the definition of the input port, the setting of the data transmission baud rate, and the activation of the serial data transmission port. In addition, the contents of all memory devices are cleared. When the initialisation is completed, the microprocessor polls its internal receive shift register continuously (Figure 4.3, p.66) and checks whether an operating code has been received from the host computer. If so, the code will be

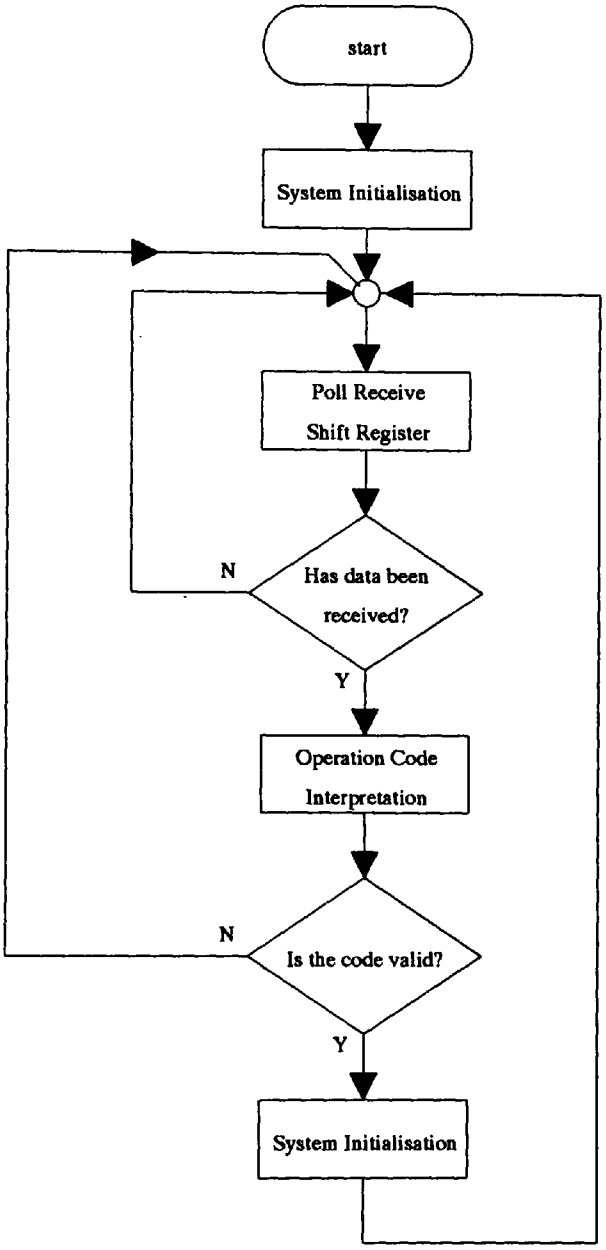


Figure 4.13 Program structure of the data collection system

interpreted which is achieved by comparing the received code with the pre-defined operating codes (Appendix B). After the interpretation, the microprocessor will take the appropriate action. A code which is not recognised will be ignored and the microprocessor then resumes polling. For example, if a code (03_{16}) representing the operation of data transmission is received, the microprocessor will jump to the correct instruction routine and execute the instructions in the hardware level. Once the operation is completed, the microprocessor resumes the polling.

Eight different operations have been implemented. Additional operations may be easily implemented without affecting other operations. This is an attractive feature of the Polytechnic of Wales EIT system facilitating easier future development. Brief descriptions of each operation are given below.

(a) Peripheral voltage measurements (Appendix E, option 2, p.196)

The number of frame to be recorded defaults to one initially. During the process of attenuation factors set-up, one frame of data is enough. However, it may be changed under the normal operation. As many as ten frames of data can be recorded for a single operation code. The microprocessor requires further information before the measurements start. It will examine the next code and check whether normal operations are requested or not. If it is not, the microprocessor will undergo the attenuation factor set-up by default.

Prior to the commencement of the attenuation factor set up process, all data storage location (from 1000_{16} to $5fff_{16}$) are set to zero so that the maximum system gain is used, and a maximum of seven iterations is allowed. An acknowledge code (the frame number which is 01_{16} in this case) is sent back to the host computer. All attenuation factors are kept in the static ram. A valid flag designated as 'atvald' is used to check the validity of each recorded data. If 'atvald' equal to $f0_{16}$ is obtained after one frame of measurements, all recorded data are considered to be fell within the limits of the A-D converter and the set-up process is completed. Otherwise, the process continues unless it reaches the maximum iteration limit.

The microprocessor initially sets up the electrodes 16 and 1 as the first drive pair by loading a special code to the decoder (74LS154, Appendix G, G.7, p.218). A complete set of codes for drive and measurement electrode pairs is shown in Table 4.4. To select the drive and measurement electrode pairs, the desired drive position

Electrode Pairs	Drive (in hex)	Measurement (in hex)
16 - 1	10	0f
1 - 2	0f	fe
2 - 3	0e	ed
3 - 4	0d	dc
4 - 5	0c	cb
5 - 6	0b	ba
6 - 7	0a	a9
7 - 8	09	98
8 - 9	08	87
9 - 10	07	76
10 - 11	06	65
11 - 12	05	54
12 - 13	04	43
13 - 14	03	32
14 - 15	02	21
15 - 16	01	10

Table 4.4 Equivalent code for selections of drive and measurement electrode pairs

is required to be loaded in the address 0024_{16} and measurement position in 0025_{16}, refer to Appendix C. The first measurement pair is the electrodes 2 and 3 while the last measurement electrode pair is 14 and 15. Each of the data conversions is performed after a time delay used to allow enough settling time for the low-pass filters in the demodulation processes. The time delay of approximately one milli-second was achieved by a simple delay loop in the software. This delay loop is only used once for every single measurement position although both real and imaginary parts signals are required to perform the A-D conversions. The microprocessor first selects the real part signal for conversions by resetting the flip-flop (74F74, Appendix G, G.9, p.220). Pre-setting the flip-flop is for conversions of imaginary part signals. Each voltage data will occupy two successive locations in the static memory.

Having completed these thirteen measurements, the drive pair moves to the electrodes 1 and 2 and another thirteen measurements are taken. Shifting of drive

and measurement pairs continues until one frame of measurements is completed. In each real part voltage measurement, the recorded data is examined by the microprocessor in order to check whether or not the data are valid. The criterion for valid data is that the converted data should not exceed the value indicated below.

MSB LSB
0aaa 1111 1111 xxxx

A voltage is recorded as a 16-bit (two bytes) data word. The most significant bit is not used and is always equal to '0'. The next three bits 'aaa' show the value of the attenuation factor. The rest indicate the actual data resulting from the A-D conversions. To make data comparisons the first four bits '0aaa' are not taken into account. Thus, as long as the actual data is not greater than or equal to '1111 1111 0000' in binary, the data is considered to be valid and the "atvald" is not clear. Due to the presence of electrical noise, the last four LSB are not compared.

In normal operation, the drive and measurement electrode pairs are shifted in the similar fashion as in the previous case. The only differences in this process are that no data validations need to be performed. All data are stored in the static memory. In addition, the collection of multiple frames is also possible.

(b) Serial data transmissions (Appendix E, option 3, p.199)

The data transmission starts with data which is the last recorded data in the voltage measurements, and finishes at the beginning of the data storage which is 1000_{16}. A baud rate of 9600 was used with no handshaking. When all data are transmitted, a completion signal ff_{16} is sent.

Having transmitted the data, another transmission is started upon the receipt of operating code 05_{16} (Appendix E, option 3a, p.200). When the microprocessor receives this code, it will undergo an identical mathematical computation which has been mentioned in section 4.3 (pp.85-6). As a result, three verifying codes are created and then transmitted to the host computer. The whole transmission process is therefore completed.

(c) Alteration of applied stimuli settings (Appendix E, option 1, p.196)

This operation allows users to change the magnitude and frequency of the applied stimuli. This is an interactive process. Each single setting is represented by a setting code (Table 4.1, p.69), and responds immediately at the hardware level. This can be done by latching the appropriate code to the D-type latch (74LS373(U3), Appendix G, G.7, p.218) which is used to hold the desired magnitude and frequency settings. The effect of the applied stimuli changes on the peripheral voltages can be observed simultaneously. This operation can be terminated by receiving the completion code which is again ff_{16}.

(d) Initialisation of data storages (Appendix E, option 4, p.200)

All data storage from 1000_{16} to $5fff_{16}$ are reset to zero using the command 'clr'. All attenuation factors will be equal to unity after this operation.

(e) Dual-frequency measurement (Appendix E, option 5, p.200)

The entire structure is similar in part to the normal voltage measurements. But no attenuation factor set-up process is available in this option. The attenuation factors will be the same as those assigned in the process in (a). The only difference between the normal voltage measurements and this measurements is that the stimuli setting code is altered automatically after the completion of the first frame of measurements. These two settings must be defined prior to the commencement of measurements.

(f) Cardiac gated imaging (Appendix E, option 6, p.201)

The structure is similar in part to the normal voltage measurements. Only one applied stimuli setting is used and the attenuation factors are the same set as in normal operation. Having set up the applied stimuli, the microprocessor polls the I/O port (dr1) and checks whether or not a R-wave related signal generated by an ECG monitor is detected. As soon as this signal is detected, the microprocessor initiates the voltage measurements. Only two frames measurements per cardiac cycle are taken. When the measurements are completed, the microprocessor will

continue the polling of the I/O port until another R-wave related signal is detected. A maximum measurement of five cardiac cycles is allowed.

(g) Examination of data collection electronics (Appendix E, option 7, p.202)

The purpose of this operation is to sweep through the 208 measurements at a user defined speed. The program sequences are identical to those in the normal voltage measurements except that the delay loop employed in the former case is replaced by a user controlled delay loop. The delay loop is in fact a process to receive a data from the host computer via the serial input port (trcsr). This is a continue loop if no data is received. Thus, the data collection time for each measurement can be controlled by the user on the host computer. In order to sweep through all measurements and be able to observe the variation of peripheral voltages, the dwell time of the order of seconds is needed. Therefore, the minimum dwell time for each measurement is set to 0.1 second.

4.5 System Functional Testings

The host computer data handling and processing system and the data collection system were successfully combined to produce a complete impedance imaging system. The Polytechnic of Wales EIT system has been developed to the point where it is capable of correctly applying stimuli to an object, making voltage measurements, transmitting data and displaying images. In this section, a few conductivity images are shown in order to prove the system is functional. Voltage measurements were performed on the two kinds of phantoms, i.e. a two-dimensional resistor mesh and three-dimensional cylindrical saline tank.

The resistor mesh phantom to be used in here will be further discussed in chapter five including its theory, advantages and disadvantages, and the method of implementation. Using an overall system gain of about 300, voltage measurements were performed at four frequencies but only two images obtained from measurements taken at 10.24 kHz and 40.96 kHz are shown. A stimulus of 1 mA was used in all cases unless otherwise specified. Two experiments were performed on the resistor-mesh. Figure 4.14 shows a complete layout of the resistor mesh. The first experiment involved the placement of two plug-boards at locations A and B. Increases in conductivity were simulated in both regions. The second

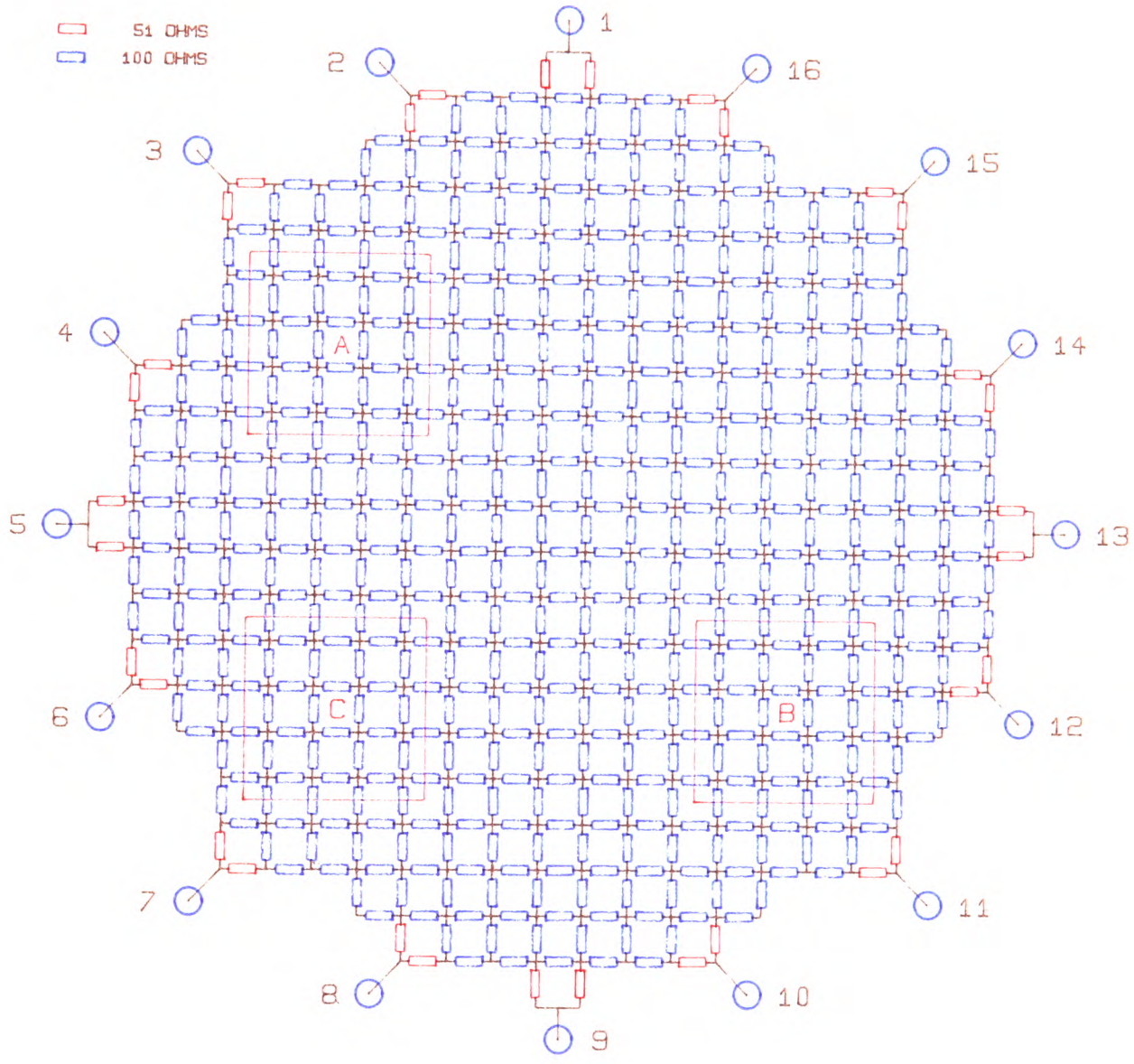


Figure 4.14 Layout of the two-dimensional resistor-mesh for conductivity imaging

experiment was similar to the first one but a decrease in conductivity was simulated in Region B. The conditions relating to these are shown in Table 4.5.

		ratio of final to initial conductivity (m)		
	Freq (kHz)	Region A	Region B	Region C
Experiment I	10.24	1.5	2.2	no change
Experiment II	40.96	1.5	0.67	no change

Table 4.5 Experiment set-up for conductivity imaging

Figure 4.15 illustrates a reconstructed conductivity image for experiment I. It can be seen that the changes in conductivity were detected and appeared in the correct positions. The image in region A is lighter than in region B which is as expected. The local maximum image values in regions A and B are approximately 0.11 and 0.23. Note that the conductivity scale increases from minimum (blue) to maximum (red).

Figure 4.16 illustrates positive and negative changes in conductivity and is consistent with the pre-set values. The displayed image shows clearly that similar values of absolute local maximums are obtained. The numerical values in regions A and B are 0.15 and −0.17. The experiments were repeated five times and similar minimum and maximum image values were obtained. The standard deviations of the minimum and maximum values were evaluated as 0.004 and 0.002 respectively. This indicated that the difference in the magnitudes of these image values was not due to random noise. The difference was probably due to the reconstruction algorithm [34].

Finally, in addition to the two experiments carried out on the resistor mesh, voltage measurements were performed at 40.96 kHz on a cylindrical tank containing saline solution. The internal diameter of the tank of was about 190 mm, and the conductivity of the saline solution was set to 0.2 S/m. In this experiment, the overall system gain (defined as the gain increases from the output of the instrumentation amplifier to the input of the analogue multipliers) was increased to about 1000 and the stimulus of 2.5 mA was used. Figure 4.17 shows a conductivity image representing two glass bottles each of diameter 65 mm immersed in the tank. This is an experiment mimicing human expiration and inspiration situations. The

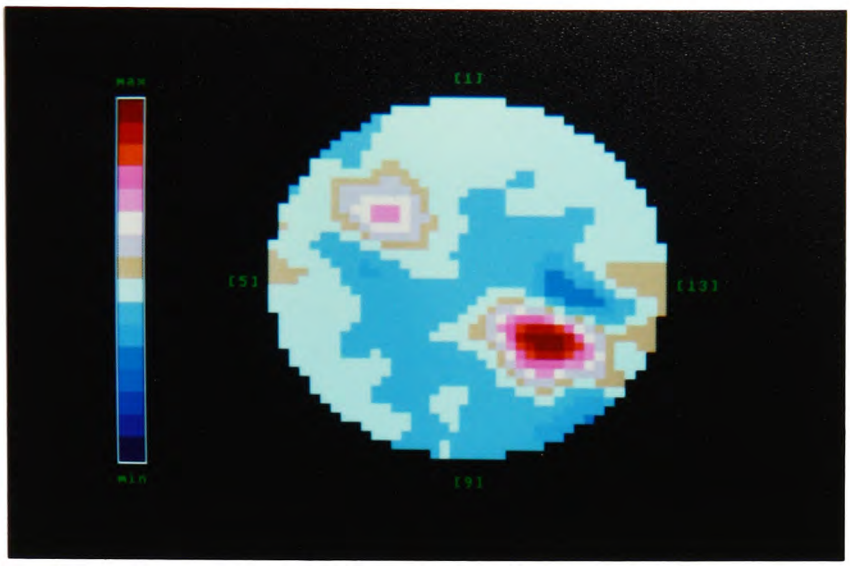


Figure 4.15 Conductivity image at 10.24 kHz
Region A : m = 1.5; Region B : m = 2.2

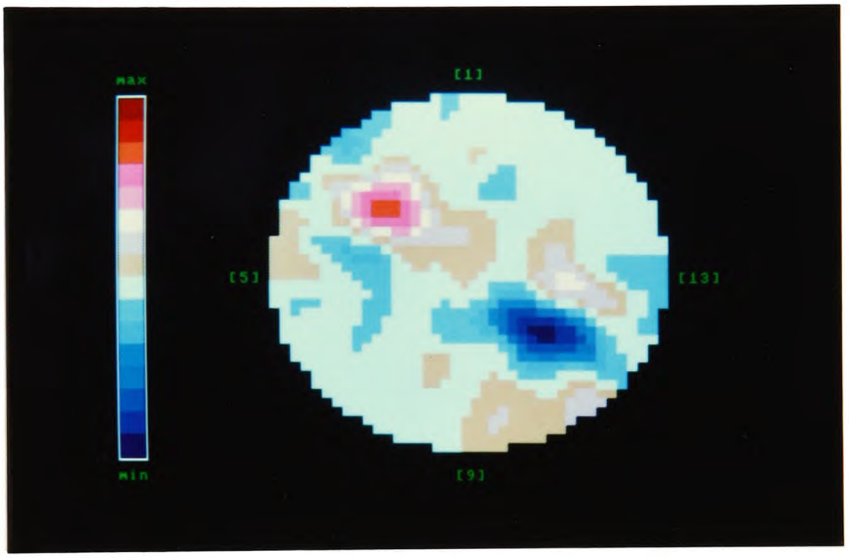


Figure 4.16 Conductivity image at 40.96 kHz
Region A : m = 1.5; Region B : m = 0.67

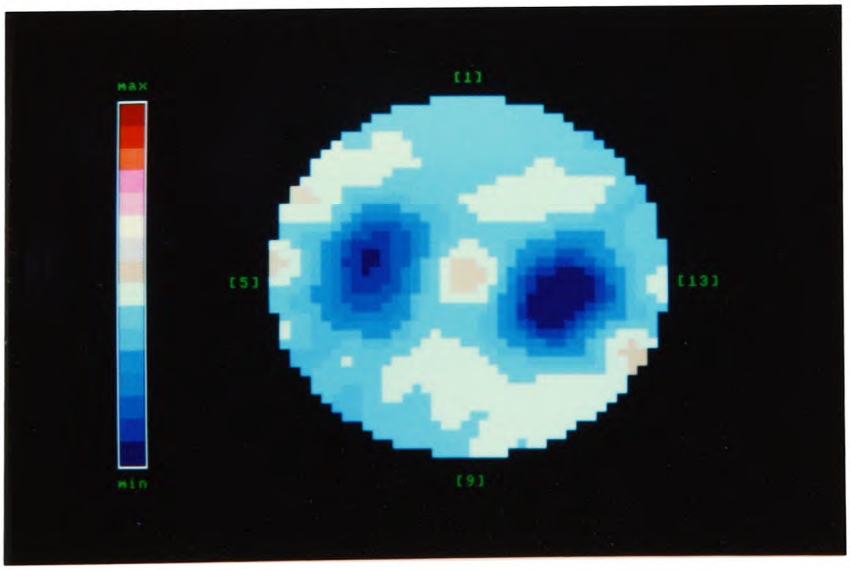


Figure 4.17 Conductivity image at 40.96 kHz
Saline solution of conductivity = 0.2 S/m

reconstructed image was obtained by using the data sets for the immersed bottles as the reference. Thus, the two bottles shows high resistivity (appeared in dark blue) in comparison with the saline (appeared in light blue).

To conclude, conductivity imaging can be performed with the Polytechnic of Wales EIT system. With an overall system gain of 100, imaging at four frequencies was possible. However, the system became bandlimited when inserting a band-pass filter. It is because when the system was tested on human subject at the later stage of development, it was necessary to increase the system gain significantly, e.g. gain of the order of thousands. Under these circumstances, stability problems dictated that the system should be bandlimited to the frequency of the applied stimuli.

4.6 Possible Improvements to the Data Collection System

Having successfully completed the construction of an EIT system, consideration was given to improvements of data collection methods. Two methods for data acquisitions are proposed and discussed. No actual implementations of these proposals have yet been carried out, but it is considered that further investigations could prove useful.

(a) Peak and phase measurements

Direct measurements of the peak value and phase difference of the voltages may be one alternative. Figure 4.18 shows the proposed method of measuring the peak of the differential voltages signals. Since the sinusoidal voltage signal is digitally generated, its peak can be easily predicted if its zero-crossing point is known. A binary counter (CNT) and a zero-crossing detector (ZCD1) are the essential elements for the peak value measurements. In order to implement the circuit, a sample-and-hold circuit (S/H) together with a fast 12-bit A-D converter (ADC) is required. With reference to Figure 4.18, as soon as a negative-to-positive transition of the monitored differential voltages is detected, the counter initiates a counting process. This counter is pre-programmed in such a way that it will stop and trigger the sample-and-hold circuit to capture the signal. The best position for sampling is not exactly at the peak of the signal but one clock pulse before the signal reaches its peak (see Figure 4.19). This is because the circuit requires a short period of time to respond and the rate of change of the signal is close to a minimum when it is

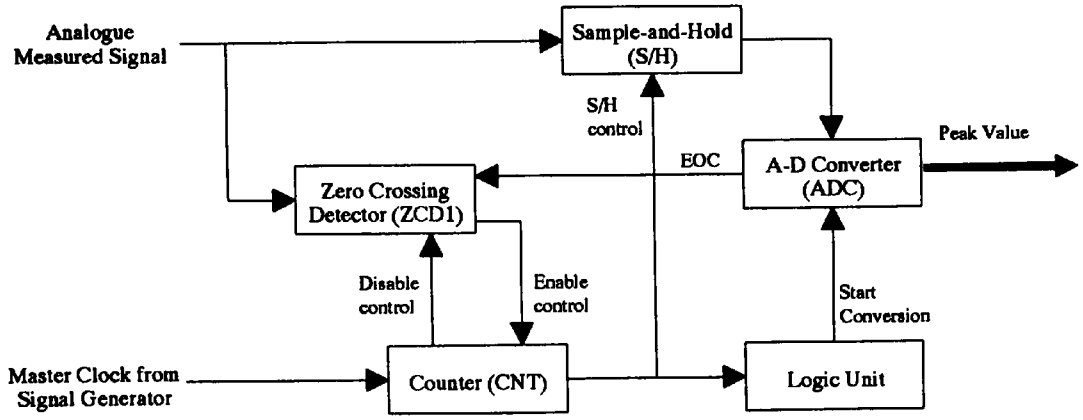


Figure 4.18 Proposed alternative arrangement of peak measurement

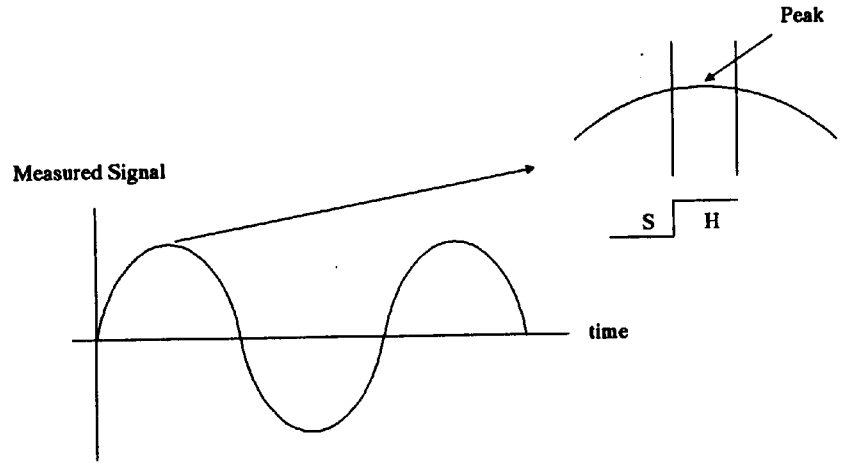


Figure 4.19 Best position for sampling

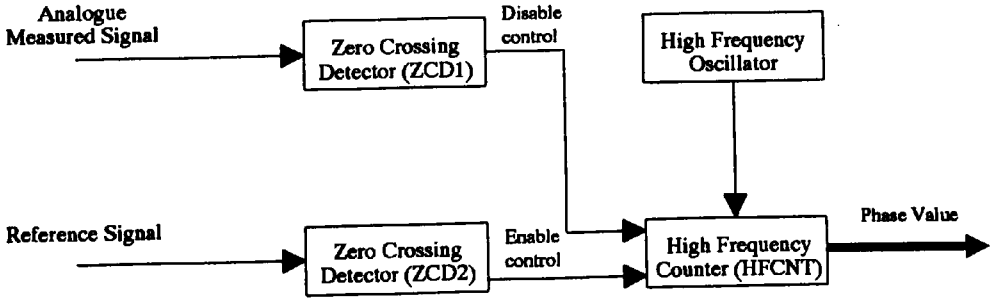


Figure 4.20 Proposed alternative arrangement for phase measurement

slightly off the peak. Once the sample-and-hold circuit retains the signal, the A-D conversion starts. The zero-crossing detector is disabled upon the detection of the first zero-crossing point. As soon as the conversion is completed, the zero-crossing detector is enabled again and the sample-and-hold circuit is reset. The whole circuit is then ready for the next data collection.

In Figure 4.20, a similar arrangement is shown for the measurements of phase. Another zero-crossing detector (ZCD2) and a high frequency counter (HFCNT) are used instead of the sample-and-hold circuit and the A-D converter in the previous case. The use of the second zero-crossing detector is to detect a negative-to-positive transition of the reference signal while the similar transition of the measured signal is detected by the first detector. When the transition of the reference signal is detected, the high frequency counter commences the counting process. The counting is stopped upon the detection of the transition of the measured signal. The output of the counter representing the phase difference between the reference and measured signals can be read by the microprocessor.

(b) Real and imaginary measurements

This is an alternative method of extracting the real and imaginary components from the peripheral voltage signals in lieu of the conventional analogue or digital demodulation processes. Figure 4.21 illustrates the configuration of the proposed method in which only one zero-crossing detector and two sample-and-hold circuits are needed. The zero-crossing detector is used to detect the positive-to-negative transition of the reference signal while the two sample-and-hold circuits are used to capture the measurement signals at two defined positions as illustrated in Figure 4.22. If the sinusoidal voltage signal is sampled at a rate of 64 samples per cycle, the sampling points for the real and imaginary components are at 32 and 48 clock pulses respectively after the detection.

At these sampling points, the sample-and-hold circuits hold the measurement signal at two different positions at which the real and imaginary components are presented. The captured voltages can then be digitised by two A-D converters or by one converter if the conversions are carried out in a time-multiplexing mode.

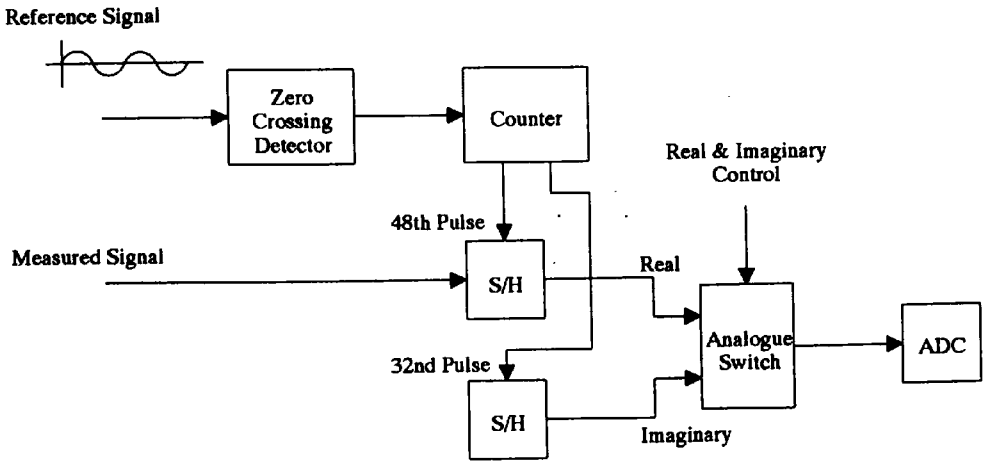


Figure 4.21 Arrangement for real and imaginary measurement

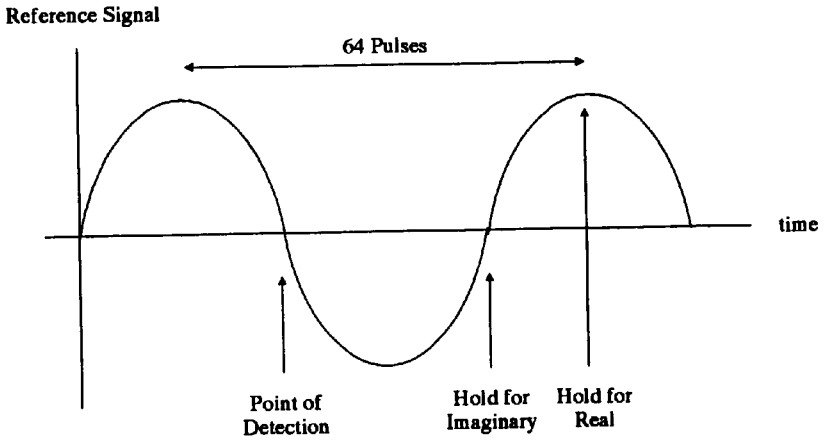


Figure 4.22 Positions of the signal to be captured

To conclude, both data acquisition methods require high quality sample-and-hold circuits, i.e. low droop rate, short aperture time and small feedthrough. The major improvement of using either type is the fast data collection speed since no low-pass filters are involved in the data acquisition process and hence long settling times are not required. "Noisy" devices, such as the analogue multiplier, are no longer employed in the process, therefore accuracy may also be improved.

Good quality sample-and-hold circuits are commercially available. For example, a monolithic integrated sample-and-hold AD683 offers the aperture delay of 2.5 ns, the droop rate of 0.01 μV/μs and the feedthrough of −100 dB. With an aperture time of 2.5 ns, only a 0.13% of voltage changes results in operations at 80 kHz while a 0.06% changes results at 40 kHz. These are calculated at the maximum rate of change of a sinusoidal signal, i.e. crossover point. Thus, for operations running at or below 40 kHz, the error caused by the aperture time is within the allowable tolerances, i.e. 0.1%. Considering the effect of the droop rate on the captured signal, it is obvious that the droop rate becomes insignificant for fast data conversions. For a data conversion time less than 50 μs, the voltage leakage of only 0.5 μV results. This is again for a minimum voltage of 0.5 V. Therefore using these methods may significantly improve the data collection speed as well as the accuracy.

Chapter 5

Performance Testings of Data Collection Systems

In order to assess performances of data collection systems, it is desirable to employ phantoms (test objects) which can be used as a common basis for comparisons. An EIT phantom is one in which localised changes in conductivity and permittivity can be achieved so as to provide data to enable impedance images to be reconstructed. Phantoms may be implemented in hardware or software, and their structures may be two- or three-dimensional. The use of hardware phantoms is primarily for confirming correct system operation and performance. For performance evaluations, the phantom must be highly electrically stable. Software phantoms are primarily used for investigations of reconstruction algorithms since the simulated voltages are free from electrical noise. In such cases, investigations of the effectiveness of different current drive layouts may be possible.

Since EIT phantom development is not the prime remit of this programme, a full discussion is not presented, however, the reader is directed to other to the work of groups, e.g. Griffiths *et al.* [12,65], Murai *et al.* [16], for an indication of the current status. This chapter focuses on the recent development of a two-dimensional resistor-mesh phantom originally reported by Griffiths [12].

The phantom [12] consists of a resistance mesh containing 592 resistors. The value of each resistor in the mesh is 100 Ω except those connected to electrodes which are 47 Ω. It is a 20×20 pixel mesh with circular outline as shown in Figure 5.1. The resistance between adjacent electrodes varies between 131 Ω and 174 Ω depending on the position. This does not create problems since the mesh is fixed and stable.

To enable measurements of sensitivity and resolution it is necessary to create changes in conductivity and permittivity at any region within the phantom. This is achieved by shunting the resistors in the region with additional resistors and/or capacitors respectively. These components are mounted on small boards containing sockets so as to enable them to be plugged onto the pins in the mesh.

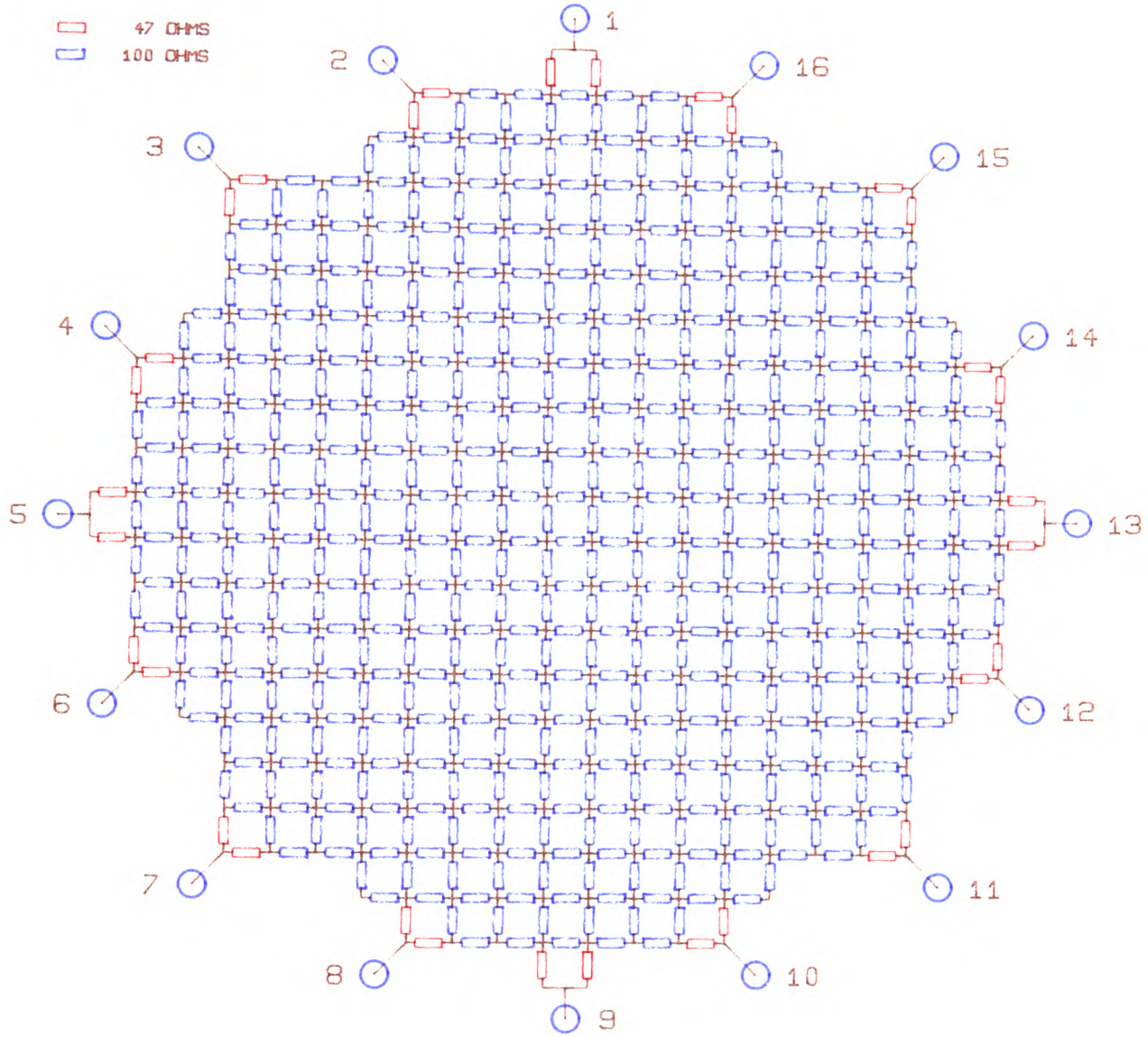


Figure 5.1 A layout of a two-dimensional resistor-mesh

5.1 A Two-dimensional Resistor-mesh Phantom Using SMT

The first version of this phantom was produced by using conventional resistors mounted on "Veroboard". Despite the tedious manufacturing process, the phantom does give several advantages for EIT development work. These advantages are summarised below:

* accurate knowledge of the impedance distribution within the test object;
* the ability to effect precise changes in conductivity and permittivity;
* high electrical stability with respect to time and temperature (0.005%/°C);
* reproducibility.

The total number of resistors in the mesh is limited only by physical size. This type of mesh is anisotrophic. When a finite element method of reconstruction algorithm is employed, the resistor-mesh is therefore not ideally suited to investigations of image reconstruction algorithms. Nevertheless, it is a suitable phantom for hardware development.

In view of the difficulty of small quantity manufacture, few centres employ such a phantom. To redress this, a surface mount technology (SMT) version was developed and implemented. The prototype was presented at the EEC EIT Workshop at Cardiff on 23th-24th November, 1990. This phantom has been accepted as a standard test object for assessments of system performance within the CAIT research programme of the EEC. A photograph of the phantom together with a plug-board is illustrated in Figure 5.2. The SMT resistors are mounted on a 22 cm × 22 cm printed circuit board. Turret tags are used for electrode contacts. Veropins are used to allow a small plug-board containing resistors and/or capacitors to be plugged onto the mesh, as shown in Figure 5.2. This provides the means to change the conductivity and/or permittivity values locally. The plug-board is also constructed using SMT. Terminal jack sockets (Farnell Electronic Components) are used in this case. Both the mesh and the plug-board are protected by perspex covers. The whole unit is rigid, robust, and therefore suitable for transferring between centres. The total weight of the mesh is approximately 0.5 kg.

To obtain pure conductivity changes only resistors are required to form the plug-board. Consider that the electrical conductivity in the four pixels illustrated in Figure 5.3(a) to be increased by a factor of m, then the inner resistors R_1 and

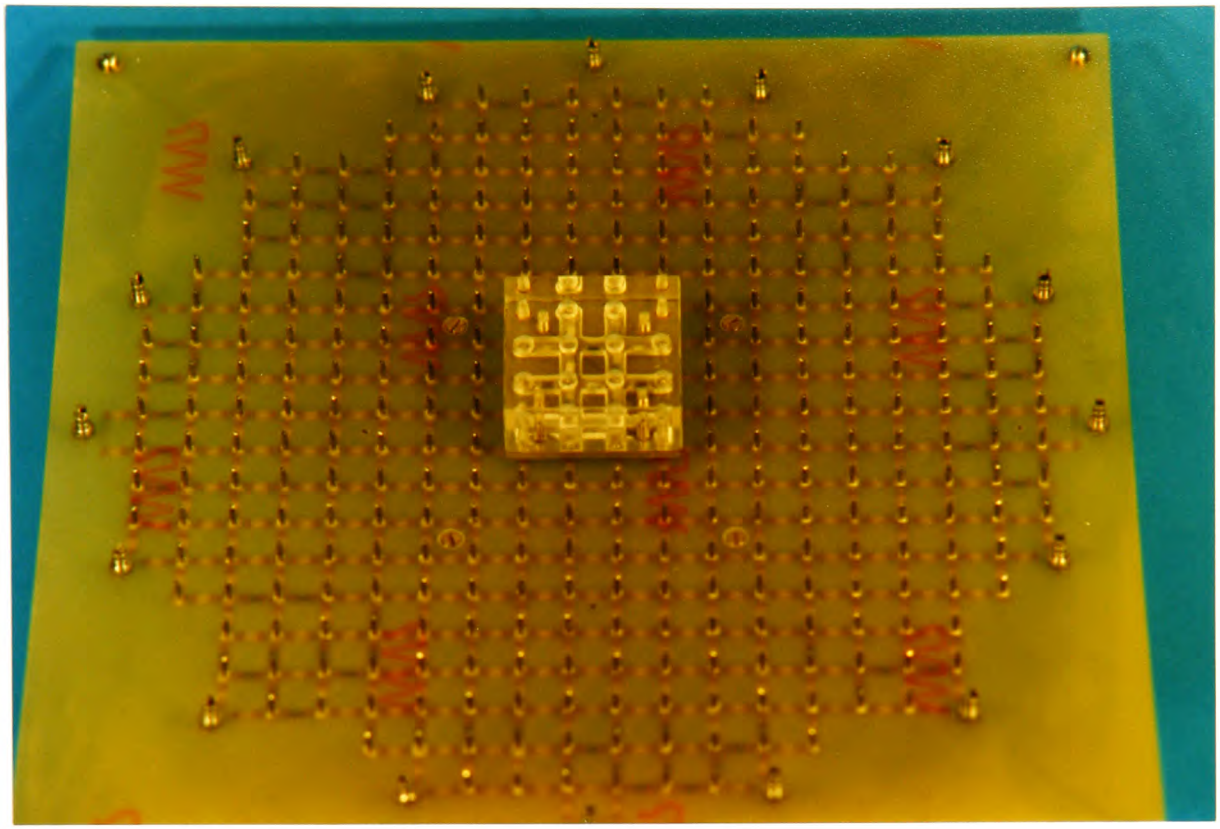
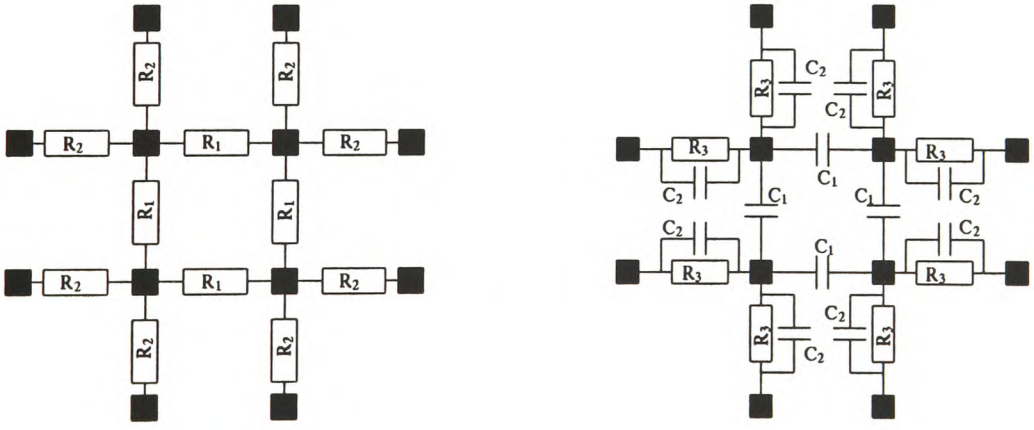


Figure 5.2 A two-dimensional resistor-mesh using SMT



(a) (b)

Figure 5.3 Plug-boards for resistive and reactive shunting

■ Plug contact

outer resistors R_2 may be evaluated according to Equations 5.1 and 5.2 [12], where R is the resistor value in the phantom.

$$R_1 = R/(m-1) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.1)$$

and

$$R_2 = R(m+1)/(m-1) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.2)$$

For the configuration illustrated in Figure 5.3(b), the reactive current can be increased by n times the resistive current. The required component values can be calculated using Equations 5.3 to 5.5 [12], where ω is the angular frequency of the applied current.

$$C_1 = n/(\omega R) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.3)$$

$$C_2 = 2n/[\omega R(4+n^2)] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.4)$$

and

$$R_3 = R(1+4/n^2) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.5)$$

In the case that n is much smaller than one, R_3 can be omitted.

There are two additional advantages of using the surface mount version. More resistors may be included without significant increases in the overall size. Also, stray capacitance is relatively small in comparison with the conventional version in which a large amount of copper tracks and metal wires are present. In order to maintain the performance consistency between SMT phantoms, ±1% or better tolerance resistors must be used. With a high consistency, it has an advantage that one set of theoretical data may be adequate for all phantoms. Effects due to stray capacitance are constant as the track layout is identical for all boards.

5.2 A Computer Model of the Phantom

Software phantoms (computer models) are used to generate a set of vectors which represents boundary measurements for a given set of drive configurations. The usual means of producing software phantoms employs conventional programming language techniques. Broadly speaking, these models can be divided into two categories viz analytical models and numerical models. The shape of an analytical model cannot be arbitrarily defined and it must be the same as the physical model; the shape of a numerical model can be arbitrarily defined to suit individual

applications. The computer model discussed in this chapter is an analytical model based upon an electronic circuit CAD tool. Such a technique has not previously been reported.

5.2.1 An ASTEC3 Simulator Model

A Fortran model of the phantom has been developed by Griffiths but has yet to be reported. This program is capable of providing the necessary data for system performance characterisations. However, due to the nature of the hardware phantom, it was considered that the development of an alternative computer model using an electronic circuit simulator could prove advantageous. A computer model was therefore implemented using the SIA-ASTEC3 analogue circuit simulator running on a VAX11/785 mainframe computer. It is, however, not confined to the ASTEC3 simulator as long as a simulator which is capable of handling large numbers of components and connection nodes is used, e.g. PSpice (Extended memory version) would possibly provide a suitable alternative. The model is a significantly large resistor network whose configuration is identical to the hardware phantom. The current drive circuitry can be in any form. In an ideal situation, ideal current sources may be used. The use of actual or equivalent circuits of practical current sources is also possible. In the case of simulating a multiple drive system, it is suggested that a modelling technique described in [66] should be employed. The modelling technique is a method used to ease the circuit descriptions in the program, and is particularly useful when a circuit is frequently used throughout the program. Once a drive circuit is modelled, it can be used anywhere without the need to duplicate circuit descriptions. Figure 5.4 shows a sample arrangement of such computer model. Note that the model described here is a software representation of the resistor-mesh only. A model is created by entering a netlist via the interactive programming language specific to ASTEC3. This is however similar to that employed in other industry standard simulators.

The model when exercised produces data sets which replicate a physical system. Two sets of data are required for image reconstruction. The first set of data, known as reference data V_1, is obtained without any perturbation, i.e. only simulating the resistor-mesh. The second set of data, known as image data V_2, is required to have a certain amount of perturbation. As an example, a reconstructed image illustrated in Appendix F shows an increase in conductivity in the perturbed region. The image was produced based on the simulated data. Additionally, the logarithmic ratios of

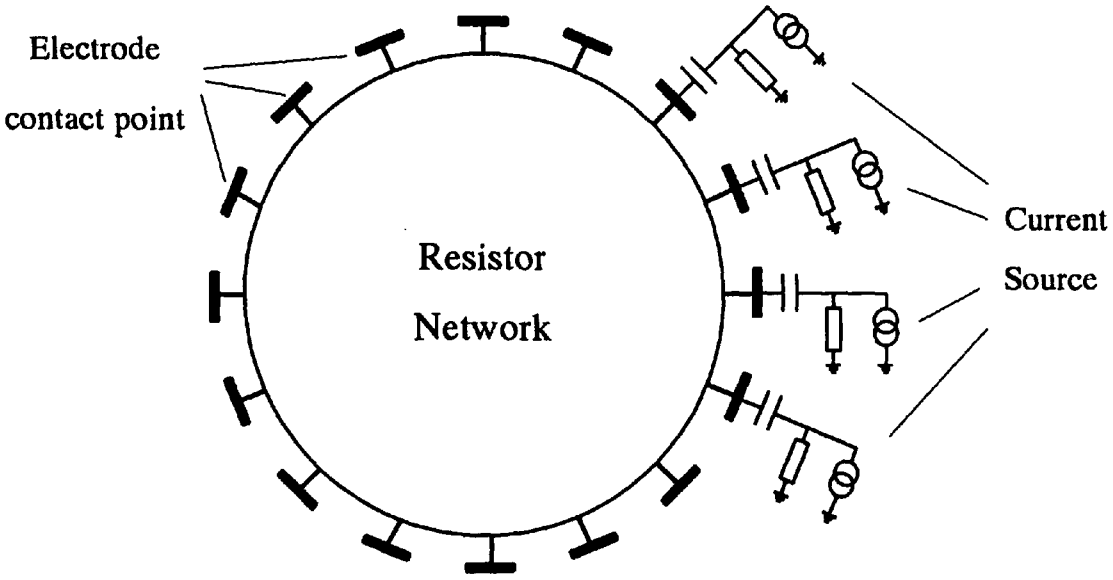


Figure 5.4 A sample arrangement of the ASTEC3 computer model

these two sets of data, i.e. $\ln(V_1/V_2)$, have been compared with the data generated from Griffiths's computer model. Only a small discrepancy (at the fourth significant figure) between any corresponding ratios resulted. This could be probably due to the rounding errors of different types of computers.

5.2.2 Use of the ASTEC3 Model

In addition to the capability of generating sets of theoretical data, this kind of computer model also offers two major functions. One is the statistical analyses of the performance of the hardware phantom. The other is the study of various drive topologies which are currently under general debate.

Firstly, using sophisticated analogue simulation, the model allows component tolerances existing in the physical item to be included in the computations. The mean value and standard deviation of data from simulated experiments can then be evaluated for a given set of component tolerances. This is important as each practical measured potential is not only contaminated by electrical noise but also deviates from the true value due to the component tolerances. The estimated standard deviation would indicate the degree of variation between physical models, and also indicate whether or not one set of reference data may be employed for a set of phantoms; as opposed to the need to generate a reference data set for each individual.

Secondly, circuit interactions in the current drive and peripheral voltage measurement systems are also able to be simulated. Since the phantom is just a passive component network, there is no reason why the practical behaviour of various drive and measurement topologies cannot be studied using this model. It was the intention that the creation of the modelling process would produce a platform for future work.

In the statistical analyses executed by the author, a component tolerance of 1% (Gaussian) was assigned to each resistor. Two analyses, with and without the plug-board attached, were carried out. Results of the simulations are shown in Appendix F. The results showed that values of SD/mean are within the 0.1% accuracy. Therefore the simulated peripheral voltages data using nominal resistor values can be applied to all SMT resistor-mesh phantoms whose resistor tolerances

are less than 1% (SD). This is of particular value where a number of test objects are used as a standard.

5.3 System Performance Characterisations

To test whether an imaging system is functional it is only necessary to perform voltage measurements on the SMT phantom and to display the reconstructed image based on the measured data. A visual comparison between the theoretical and practical images is adequate for this purpose. When adopting this visual comparison method, both images must be reconstructed using the same algorithm, otherwise, the reconstructed images cannot be compared. However, the visual comparison method may not be adequate for system performance characterisations since the comparison is subjective. Quantitative methods are thus essential for this particular task. Since the reconstructed images are created from sets of boundary measurement data, it is wise to use the un-processed measurement data as a basis for the comparison. Basically, system performance characterisations can be put into two major categories.

* system characterisation - this provides a means to compare systems which may not have identical configuration, and
* comparison between practical and theoretical data - this allows a single system design to be evaluated.

5.3.1 System Characterisation

To assess system performance only the real parts of the measured signals were used for comparisons. A figure which is used to evaluate system performances with respect to various comparison methods is called the performance index. Three performance indices were defined. The comparison methods suggested here are based on a 16-electrode EIT configuration employing the Sheffield data collection method with adjacent current drive and voltage measurement. These indices are:

(a) random noise (E_{rn})

Random noise, which occurs in the front-end electronics and is defined as electrical noise in this context, effects the accuracy of the measured peripheral voltages; and hence effects the quality of the reconstructed images. The

evaluation of random noise considering here does not include any noise introduced into the image through the reconstruction processes. To evaluate the random noise level in the measured data it is necessary to collect two frames of data taken on a uniform resistor-mesh in the shortest possible time. For a perfect system, both sets of data would be identical. Suppose that the level of random noise (E_{rn}) is indicated by a root-mean-square (rms) value. If x is a variable representing the difference between two peripheral voltage measurements at a single electrode pair, then

$$x_i = \frac{2\,(V_{1,i} - V_{2,i})}{V_{1,i} + V_{2,i}}$$

where $V_{1,i}$ and $V_{2,i}$ are the ith voltage data in the first and second measurements respectively. Since 208 voltage measurements are taken, the mean value of x is

$$\bar{x} = \frac{1}{208} \Sigma\, x_i$$

The value of rms is

$$E_{rn} = \text{rms of } x = \sqrt{\left[\frac{1}{208} \Sigma\, x_i^2\right]} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.6)$$

Typically, the noise level is high at positions opposite to the drive electrode pair. This statement is only valid for the adjacent drive and measurement data collection method. Results of such tests are shown later in this section.

(b) systematic error - reciprocity error (E_{re})

Reciprocity error, which does not vary with time, is inevitable and is produced by small mismatches between drive or measurement electronics. This error is defined as the proportional difference between reciprocal measurements. For example, considering two pairs of system electrodes, when a current is injected into one pair a voltage results across the other. If the stimulus and measurement are reversed, the voltage measurement should be the same.

To evaluate the reciprocity error one data set is required to be taken on a uniform resistor-mesh. This error is constant in time for a given set-up and therefore its rms value can be calculated. In practice, different measurement channels would have different attenuation factors in the collection system, so only the reciprocity channels with the same attenuation factors are used in the comparisons. If y is the reciprocity error and is defined as

$$y_{d,m} = \frac{2(V_{d,m} - V_{m,d})}{V_{d,m} + V_{m,d}}$$

where suffixes d and m denote the channel number of the drive and measurement electrode pairs. The rms value of y can then be expressed as

$$E_{re} = \text{rms of } y = \sqrt{\left[\frac{1}{N} \Sigma\Sigma\, y_{d,m}^2\right]} \quad \ldots\ldots\ldots\ldots\ldots\ldots (5.7)$$

where N is the number of reciprocity channels with the same attenuation factors. The size of the rms value is mainly dependent on the front-end architecture of the data collection system although the random noise would be involved in the calculation. For a system not involving matching difficulties, a smaller reciprocity error would be expected; in comparison with a system which suffers from matching problems. The results of tests using this index are shown later in this section.

(c) systematic error - frequency dependent error (E_{fde})

This comparison was intended to give a measure of the systematic errors arising when changing the frequency of the stimulus, e.g. from 40 kHz to 80 kHz. Two sets of measurements are needed on the uniform mesh but they are collected at two different frequencies. Suppose $V(f_1)$ and $V(f_2)$ are the measured voltage data obtained at frequencies f_1 and f_2. Let z_i be a variable of the ith voltage measurement which is defined as

$$z_i = \ln \frac{V(f_1)}{V(f_2)}$$

and the rms value of z is

$$E_{fde} = \text{rms of } z = \sqrt{[\frac{1}{208} \Sigma z_i^2]} \quad \ldots\ldots\ldots\ldots\ldots\ldots (5.8)$$

If a system has a flat frequency/phase response, then E_{fde} will be zero. Since it was not the intention of this programme to develop dual-frequency work, this index was not developed beyond this stage. Note that this error estimation is applicable only for systems intended to perform multi-frequency operations.

Use of Performance Indices

Having defined these performance indices, it was intended to use (a) and (b) to quantitatively compare two data collection systems of different front-end architectures. One was constructed by Griffiths using the serial drive and measurement configuration - the UHW system. The other one was constructed by the author using the parallel drive and serial measurement configuration - the POW system. Two frames of data were collected at 40.96 kHz in both systems on a uniform two-dimensional resistor mesh.

	Random Noise rms (×0.1%)	Reciprocity Error rms (×0.1%)	
		1st set	2nd set
POW system	2.37	58.45	58.67
UHW system	0.62	0.70	0.64
POW system*	1.25	58.48	58.51

Table 5.1 System performance assessments of the UHW system and the POW system - evaluations of E_{rn} and E_{re} using two successive frames of measurements
* : average of five frames

From the results shown in Table 5.1, it can be seen that the rms value of random noise of the UHW system is lower than that of the POW system. This is possibly due to the employment of longer integration time in the demodulation process. Although the noise level was lower, the UHW system exhibits a larger settling time

for each voltage measurement. By using the same integration time as the UHW system, the rms value of random noise of the POW system can be estimated as approximately 0.49 (×0.1%) which is close to the UHW system. Note that the integration time is approximately 0.85 ms for the POW system and 20 ms for the UHW system.

As far as the reciprocity error is concerned, the UHW system showed better results. This may be because the UHW system employs a serial drive and measurement configuration - no devices need to be matched. In the POW system, using parallel drive configuration to avoid stray capacitance effects between drive and measurement electrodes requires the current sources to be matched.

By averaging five frames of measurements, it is possible to reduce the noise level in the POW system. However, the reciprocity error showed approximately the same value. This provided an indication that serial type collection systems may provide better performance, in terms of reciprocity error, than parallel type systems with present technology. Unless the problem of device matching is solved, otherwise, the reciprocity error would be a hindrance to the parallel type system developments. Further discussion of this error can be found in chapter six.

5.3.2 Comparisons between Practical and Theoretical Data

The three performance indices described in the preceding section are useful for comparisons between systems. However, they do not indicate how close a system performs in comparison to its theoretical limits. In order to evaluate the discrepancy between practical and perfect systems, it is necessary to use sets of theoretical data which are free from electrical noise and systematic errors, for comparisons. With these theoretical data, two comparison methods were proposed and are discussed below:

(a) Comparisons based on a uniform mesh

This comparison was intended to evaluate the absolute error (defined as the difference of the computed and measured voltage data) resulting from the data collection system. One measured data set is adequate and is obtained by performing voltage measurements on the uniform mesh without a plug-board attached. Suppose that a single data comparison is defined as

$$p_i = \frac{V_{m,i} - V_{c,i}}{V_{c,i}}$$

where suffixes m and c denote measured and computed values of the ith peripheral voltage data respectively. Due to the reciprocity theorem, only 104 data are generated and compared although 208 voltage measurements are usually taken in practice. It is however possible to effect comparisons for all practical data by re-arranging the 104 computed data to form a set of 208 computed data. The first performance index E_1 is now defined as the rms absolute measurement error and is given by

$$E_1 = \sqrt{[\frac{1}{104} \Sigma p_i^2]} \times 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots (5.9)$$

(b) Comparisons based on a perturbed mesh

This comparison was intended to evaluate the relative error of the image reconstruction data, i.e. the difference of the ratios of the computed and measured voltage data. Two sets of measured data are needed. One is a reference data set (V_1) obtained from measurements taken on the uniform mesh. The other is an image data set (V_2) obtained from measurements taken on a perturbed mesh, i.e. with the plug-board attached. When the Sheffield image reconstruction algorithm is employed, uncertainties in the gain of the data collection system will be rejected from the calculations since $\ln(V_1/V_2)$ is the quantity to be backprojected. These uncertainties are due to the setting of the attenuation factors (section 4.2.2, p.74). Thus, each entry for the evaluation of the second performance index E_2 is defined as

$$q_i = \ln \frac{V_{1,m,i}}{V_{2,m,i}} - \ln \frac{V_{1,c,i}}{V_{2,c,i}}$$

Hence, the performance index E_2 is

$$E_2 = \sqrt{[\frac{1}{104} \Sigma q_i^2]} \times 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots (5.10)$$

Both E_1 and E_2 indicate the percentages of total errors (random noise and reciprocity error) produced from the data collection system. These values thus provide a means to benchmark a data collection system.

Use of Performance Indices

The value of E_1 was calculated as approximately 5.5% on the POW system. This 5.5% deviation from the theoretical prediction data was attributed to the presence of reciprocity errors and the variations of true attenuations in the data collection system. It was assumed that the attenuation was reduced by a factor of two along the resistor chain (section 4.2.2, p.74). It should be noted that precision attenuations were difficult to achieve although these could be measured. When evaluating E_1, each reading needed to be converted back to its original value by multiplying the measured value with its associated attenuation factor. Thus, a slight variation in the practical attenuation would produce a significant error in the estimation of E_1. However, this is the prime objective of E_1 - to trace the possible cause of errors which leads to the deviation of measured data from the ideal. A similar estimation of E_1 could not be performed on the UHW system since the individual gains relating to each set of measured data had not been recorded.

As far as E_2 is concerned, two sets of measurements were performed using the UHW system and the POW system. One was taken on a uniform resistor-mesh and the other was taken on a perturbed mesh. The amount of the conductivity changes was same as that used in the computed model and in the same location (Appendix F). Values of E_2 are shown in Table 5.2.

	E_2 (%)	
POW system	0.3089	0.2258*
UHW system	0.0816	

Table 5.2 System performance assessments of the UHW system and the POW system - evaluation of E_2
* : average of five frames

From the results, it reveals that the UHW system produced data which were much closer to the theoretical data than those produced by the POW system. The

discrepancy between the practical data and the theoretical data was calculated as approximately 0.08% for the UHW system and 0.23% for the POW system. These results were logical as the UHW system showed better performances in the previous assessments. The stray capacitance effect in the current multiplexing circuitry was the sole difficulty exhibited in the UHW systems; no device matching was required. Thus, the small deviation was attributed to random noise and the stray capacitance effects. In contrast, the POW system suffered from the problem of device matching in addition to the above two difficulties. A larger deviation from the theoretical data was therefore expected. Further discussion concerning the problem of device matching is given in chapter six.

To summarise, it was identified (section 2.2, p.12) that a low noise system (less than –60 dB) should be used in order to produce *in-vivo* images, e.g. lungs images. Using the performance index (E_{rn}) would be adequate to estimate the noise performance of a system and hence to justify whether or not the system is capable of performing *in-vivo* imaging. The estimation of reciprocity error would indicate the effects of employing serial or parallel drive and measurement system configurations. The manifestation of reciprocity error in reconstructed images will be discussed in chapter six. Using the reciprocity error index (E_{re}), the drive and measurement configuration may be quantitatively compared.

Using E_1 and E_2, the discrepancy between the theoretical and practical system performances can be evaluated. These provide an indication of how a system performs in comparison to its theoretical limits. Hence, it is possible to justify whether improvements could be made for a particular front-end architecture. All these indices are thus useful for EIT hardware development work.

Measurements of errors for comparing different systems using the standard Cardiff test phantom are still under discussion in CAIT. The evaluation of E_{rn}, E_{re}, E_1 and E_2 contained in this thesis (this chapter and chapter six), to the author's knowledge, are the first documented attempts at specifying error measurements in two data collection systems.

Chapter 6

Relative Permittivity Imaging

In addition to imaging spatial distributions of resistivity, another parameter of tissues, namely permittivity, may also be evaluated by means of EIT. Some preliminary work was conducted by Griffiths [11,12] and the reported images have shown the possibility of separating conductivity and permittivity contained in the measured data. Scaife *et al.* [58] showed phase images but they called them permittivity images. Under an ideal situation such as in computer simulations, perfect data for image reconstructions can be obtained. In practice, however, undesired phase displacements in the data collection electronics are a major problem since permittivity manifests itself as phase in the measured data. Problems exhibited in conductivity imaging, such as the effects of stray capacitance in multiplexers and matching of current sources, would also appear in relative permittivity imaging but to different extent. Additionally, the author's system has been designed specifically as a basis to investigate relative permittivity imaging, in particular *in-vivo* work. Thus, the effect of skin contact impedance on the measured data would also be investigated.

Before the *in-vivo* imaging studies, experiments on the resistor-mesh would be essential in order to confirm correct system operation and performance. Human forearm and lung imagings were intended to be performed as *in-vivo* studies. Some measurements were also performed with the UHW system in order to make comparisons between systems in *in-vivo* relative permittivity imaging.

The image reconstruction algorithm adopted is that described by Barber [23] but has been modified according to Griffiths's suggestions [11] so that reconstructions of relative permittivity images are possible. The detail of this is provided in chapter two. For a convenience, the important equations are repeated below.

$$s' = \frac{1}{2} \ln \left(\frac{\sigma_2'^2 + (\omega \varepsilon_0 \varepsilon_2)^2}{\sigma_1'^2 + (\omega \varepsilon_0 \varepsilon_1)^2} \right) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

$$s'' = \tan^{-1}(\omega \varepsilon_0 \varepsilon_2 / \sigma_2') - \tan^{-1}(\omega \varepsilon_0 \varepsilon_1 / \sigma_1') \quad \ldots\ldots\ldots\ldots (2.7)$$

conductivity : $\alpha = \ln(\sigma_2'/\sigma_1') = s' + \ln\cos s''$ . . . . . (2.12)

and

permittivity : $\beta = \ln(\omega\varepsilon_0\varepsilon_2/\sigma_1') = s' + \ln\sin s''$ . . . . . (2.13)

6.1 Effect of Contact Impedances on Relative Permittivity Imaging

In *in-vivo* conductivity imaging, the effect of skin contact impedance on the measured data is well-known. Using a four-electrode system technique would minimise this effect in conductivity imaging [5]. As mentioned in chapter two, the stray capacitance between drive and measurement electrodes creates parasitic current paths which produce measurement errors. In the previous discussions, the influence of the stray capacitance between the measurement electrode pair was not considered which is quite acceptable when dealing with conductivity imaging. Figure 6.1 shows a simple four-electrode system which models the object (R & C) to be imaged, and takes the effective capacitance C_{xm} between the measurement electrode pair into account. The stray capacitance C_x (Figure 2.4, p.18) is not shown in Figure 6.1 as it is assumed that C_x can be neglected when a parallel applied stimuli method is adopted.

Suppose that an object to be imaged contains conductivity and permittivity components. These components are represented by R and C in the four-electrode system (Figure 6.1). For the sake of simplicity, it is assumed that all electrodes have equal contact resistance R_e and the effective capacitance between measurement electrodes is C_{xm}. Using this simplified model, a value for C_{xm} can be evaluated. The true value of equivalent impedance of R and C, denotes as Z_{true}, may be expressed as $R/(1+j\omega RC)$. When passing a unity constant amplitude alternating current at an angular frequency ω and measuring the voltage developed across a and b in Figure 6.1, the magnitude $|M_{true}|$ and phase θ_{true} of the true voltage signal can be written as

$$|M_{true}| = \frac{R}{\sqrt{[1+(\omega RC)^2]}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.1)$$

$$\theta_{true} = -\tan^{-1}(\omega RC) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.2)$$

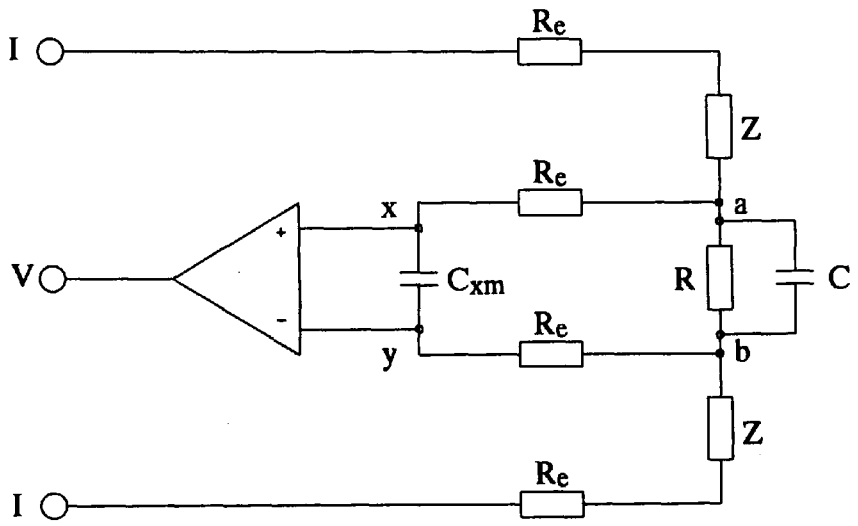


Figure 6.1 A four-electrode system taken into account of the stray capacitance between measurement electrodes

In the presence of R_e and C_{xm}, the magnitude $|M_{zc}|$ and phase θ_{zc} of the voltage developed across x and y can be written as

$$|M_{zc}| = \sqrt{\left[\frac{R^2+(2\omega R_e R C_{xm})^2}{(1-2\omega^2 R_e C C_{xm})^2+(\omega R C_{xm}+\omega RC+2\omega R_e C_{xm})^2}\right]} \quad . . (6.3)$$

$$\theta_{zc} = \tan^{-1}(2\omega R_e C_{xm}) - \tan^{-1}\left[\frac{\omega R C_{xm}+\omega RC+2\omega R_e C_{xm}}{1-2\omega^2 R_e C C_{xm}}\right] \quad . (6.4)$$

Now, to evaluate the effect of the contact impedance on the measured voltage it is necessary to let $R_e = 0$. In the absence of R_e, the capacitance C_{xm} can be considered to be a part of the reactive component of the object. If the magnitude and phase are denoted as $|M_c|$ and θ_c, then the voltage developed across x and y is

$$|M_c| = \frac{R}{\sqrt{[1+(\omega RC+\omega RC_{xm})^2]}} \quad . . . . . (6.5)$$

$$\theta_c = -\tan^{-1}(\omega RC+\omega RC_{xm}) \quad . . . . . (6.6)$$

As an example, assuming that the reactive current is 1/10th of the resistive current (i.e. n = 0.1), values of C can be calculated for R = 200 Ω. Using Equations 6.1 to 6.6, values of magnitude and phase for two different object reactances are calculated for I = 1 mA. Results are shown in Table 6.1. With reference to Table 6.1, it can be seen that the effect of contact impedance R_e on the true and detected magnitude values is small. The changes in magnitude, i.e. $(1-|M|/|M_{true}|)\times 100\%$, are evaluated as 0.063% (with R_e) and 0.026% (without R_e) for C = 800 pF. The value of $|M|$ is referred to either $|M_{zc}|$ or $|M_c|$ in Table 6.1. It can be seen that $|M|$ are approximately equal to R to within 0.1%. If $|M|$ is used for conductivity imaging, C_{xm} can be ignored.

However, C_{xm} and R_e are significant in relative permittivity imaging. For a perfect system with a 100% increase in C, a phase change of about 100% is calculated. When taking C_{xm} into account, a phase change of only 95.7% results, although the actual phase shift of 5.63° is shown. The situation is worse if the contact impedance is non-zero. With a contact impedance of 1 kΩ, a phase change of 76.9% can be detected for a 100% increase in the object's capacitance.

C(pF)	R_e(kΩ)	$\|M_{true}\|$(mV)	θ_{true}(in deg.)	$\|M_{zc}\|$(mV)	θ_{zc}(in deg.)	$\|M_c\|$(mV)	θ_c(in deg.)
800	1	198.9969	-5.74	198.8714	-7.32	——	——
1600	1	196.0760	-11.37	195.9066	-12.95	——	——
800	0	198.9969	-5.74	——	——	198.9460	-5.88
1600	0	196.0760	-11.37	——	——	195.9802	-11.51

Table 6.1 Theoretical true and detected values of magnitude and phase for R = 200 Ω and C_{xm} = 20 pF at frequency f = 100 kHz

From this analysis, it shows that, in the presence of contact impedance, the capacitance effect on the measurement pair cannot be ignored in relative permittivity imaging, unlike conductivity imaging. The presence of R_e and C_{xm} reduces the detectability of phase changes and hence makes the collection of phase information very difficult.

To investigate the effect of contact impedance and effective capacitance on reconstructed images two experiments of different set-up were conducted by means of the computer simulation. The model used was the one described in chapter five. Experiment I was performed without contact resistance R_e connected nor the effective capacitance C_{xm}. Experiment II was performed with R_e = 1 kΩ connected to each electrode and C_{xm} = 20 pF connected between measurement electrodes. With reference to the resistor-mesh layout (Figure 4.14, p.98), conductivity and relative permittivity changes in regions A and B were defined and are shown in Table 6.2.

	Region A	RegionB
ratio of final to initial conductivity (σ_2'/σ_1')	1.0	2.0
relative permittivity parameter ($\omega\varepsilon_0\varepsilon_2/\sigma_1'$)	0.1	0.1

Table 6.2 Simulation set-up for evaluations of the effect of contact impedance, R_e = 1 kΩ, f = 100 kHz

When using the results obtained under a homogeneous condition, the magnitudes of the peripheral voltages in Experiment II were found to be different from those

in Experiment I by (0.02 ± 0.15)%, mean ± SD. The phase shifts from Experiment II were 0.12° ± 0.74°. From these figures, it can be seen that the differences in magnitude and phase were insignificant. The reconstructed images are very similar to those obtained in practice (refer to Figure 6.2) and hence are not shown. Note that the images illustrated in Figure 6.2 are mirrored along the horizontal plane relative to Figure 4.14 (p.98), and the darkness of the image increases in the positive direction. It can be seen that features of both images were clearly and correctly shown in regions A and B.

In addition to the computer simulation, the effect of contact resistance was evaluated practically. Using the identical set-up (Table 6.2), measurements were performed at 40.96 kHz. The reconstructed images are shown in Figures 6.2 and 6.3. Note that these images are mirrored along the horizontal plane relative to Figure 4.14 (p.98). The same limits of the display window were used in corresponding images for comparison purposes. It can be noticed that the appearance of s′ image with contact impedance (Figure 6.3(a)) is very similar to that without contact impedance (Figure 6.2(a)). However, the image of s″ with contact impedance (Figure 6.3(b)) was contaminated by noise and systematic errors in addition to the image distortion of the two plug-boards. The effect of contact impedance can be significant in the images of β (Figure 6.3(d)). Minimum and maximum image values for Figures 6.2 and 6.3 are summarised in Table 6.3.

	Figure 6.2				Figure 6.3			
	s′	s″	α	β	s′	s″	α	β
minimum	−0.037	−0.01	−0.037	−23.1	−0.038	−0.023	−0.038	−23.1
maximum	0.181	0.033	0.181	−3.40	0.176	−0.035	0.176	−3.35

Table 6.3 Image values of Figures 6.2 and 6.3

For the practical system without contact resistance connected, the phase variation was found to be 11.65° ± 2.34°, and the minimum and maximum phase shift were 1.9° and 19.12°. With contact resistance connected, the phase variation was 11.50° ± 6.24°, its minimum and maximum phase shift were −8.2° and 31.6°. It can be noticed that a large discrepancy between the simulated system and the practical system is resulted. This indicates that the model illustrated in Figure 6.1 was probably too simple to explain the phenomenon. Nevertheless, the model shows that the detectability of phase change does decrease.

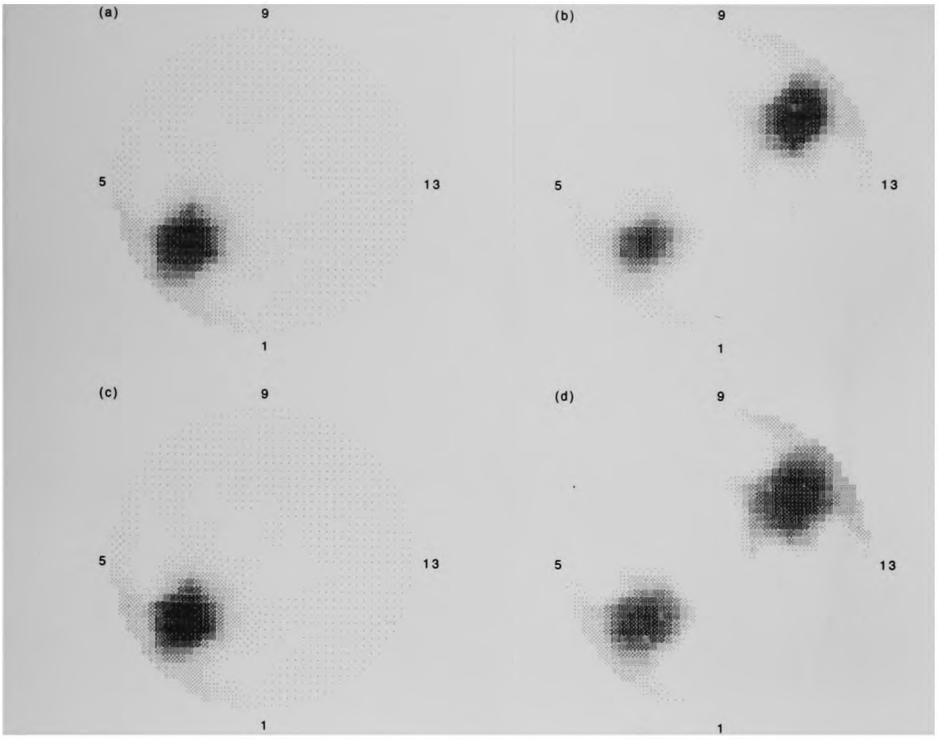


Figure 6.2 Images without contact resistance
(a) s′, (b) s″, (c) α and (d) β

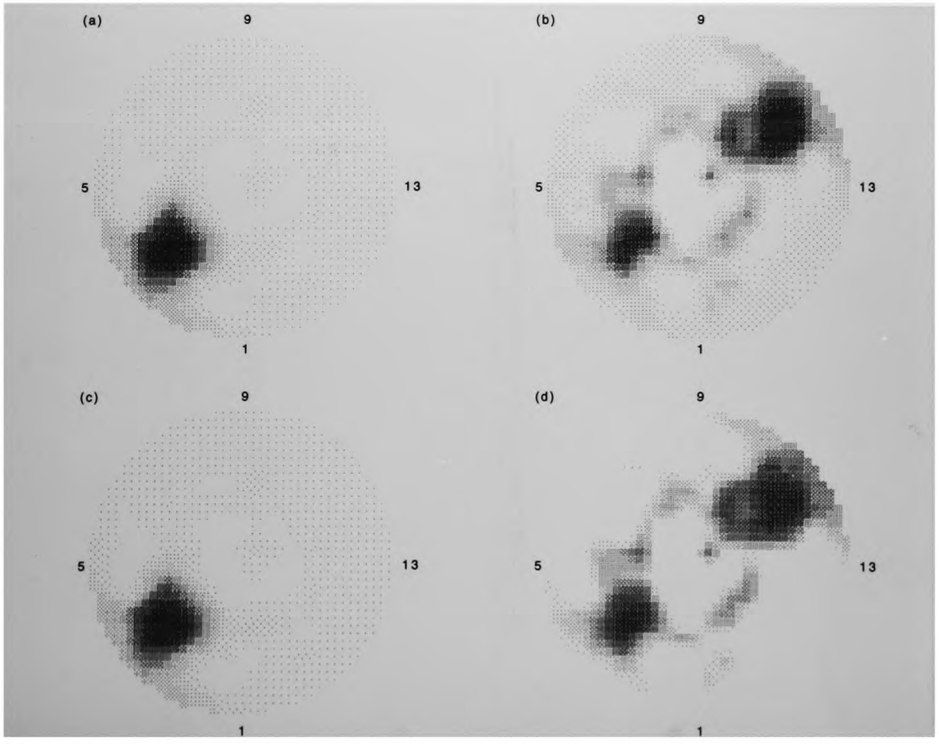


Figure 6.3 Images with contact resistance of 1 kΩ
(a) s′, (b) s″, (c) α and (d) β

Identical experiments were repeated using the UHW system, similar results were obtained. With contact resistance connected, a "noisy" image of s" was obtained. This implies that the contact resistance does affect the quality of relative permittivity images. However, no practical solution to this problem is yet derived. To ensure good electrode-surface contact, i.e. to reduce contact impedance, would be the sole method at the present time.

6.2 Difficulties of Multiple Current Source Systems

It was thought that the multiple current source systems would provide a better solution for relative permittivity imaging, since the effect of stray capacitance between the drive and measurement electrodes reduced to a minimum. However, employing multiple current sources would create other difficulties, e.g. the flow of unbalanced current results from mismatched current pairs. A high standard of current source matching is therefore required. Figure 6.4 shows a simple equivalent circuit of a multiple current source system. Single-ended current sources are usually employed in this application. Z_3 and Z_4 in the T-network represent the object impedance and Z_{imb} provides a path for the unbalanced current. Suppose that Z_1 and Z_2 are their output impedances and I_1 and I_2 are their output currents to be delivered to the loads (Z_3 and Z_4). To ease the analysis a Norton equivalent circuit can be used as illustrated in Figure 6.5. It can be seen from Figure 6.5 that I_{Nort} must be equal to zero in order to make $I_{imb} = 0$. From Figure 6.4, I_{Nort}, Z_{Nort} and I_{imb} can be derived and written as

$$I_{Nort} = \frac{Z_1}{Z_1 + Z_3} I_1 - \frac{Z_2}{Z_2 + Z_4} I_2 \quad \ldots\ldots\ldots\ldots (6.7)$$

$$Z_{Nort} = \frac{(Z_1+Z_3)(Z_2+Z_4)}{Z_1+Z_2+Z_3+Z_4} \quad \ldots\ldots\ldots\ldots\ldots\ldots (6.8)$$

and

$$I_{imb} = \frac{Z_{Nort}}{Z_{Nort} + Z_{imb}} I_{Nort} \quad \ldots\ldots\ldots\ldots\ldots\ldots (6.9)$$

It is anticipated from Equations 6.7 and 6.9 that I_{imb} would always appear in the

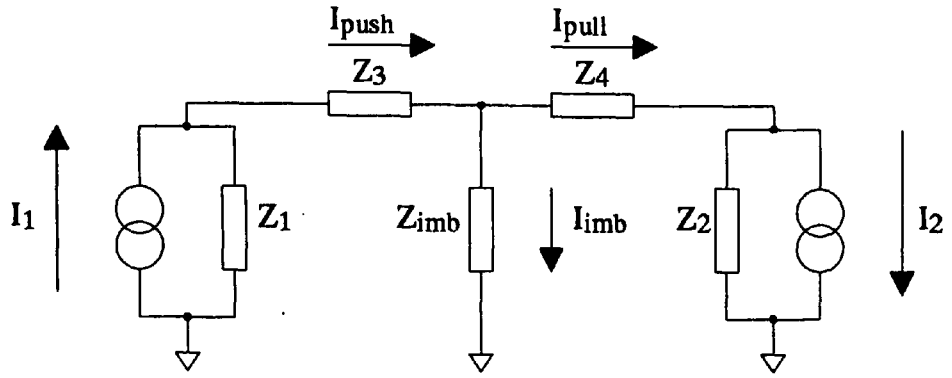


Figure 6.4 A simplified circuit for investigations of unbalanced current using two non-ideal single-ended current sources in EIT

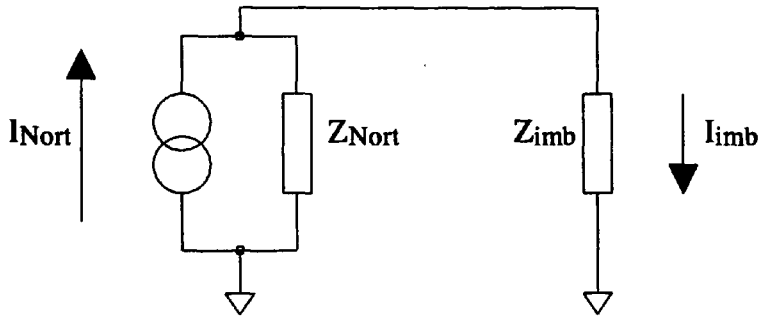


Figure 6.5 Norton equivalent circuit for two single-ended current drives

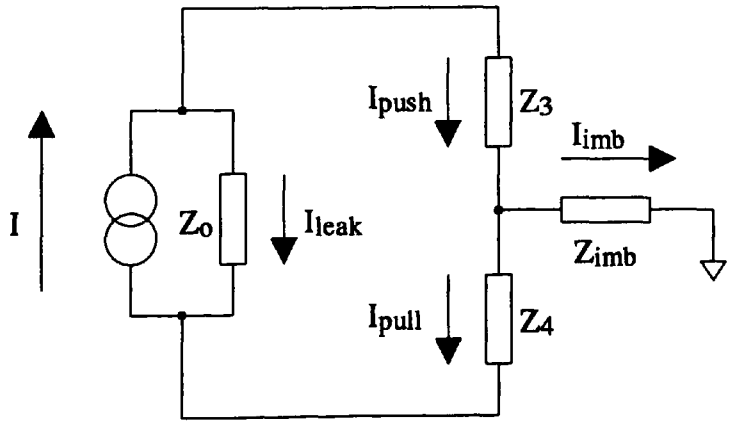


Figure 6.6 A simplified circuit for investigations of unbalanced current using double-ended current sources in EIT

EIT applications even if perfect matching of current sources was possible. Since Z_3 and Z_4 are usually not equal to each other in reality, I_{imb} would thus be non-zero. For instance, assuming that $I_1 = I_2 = 1mA$, $Z_1 = Z_2 = 1\ M\Omega$, and $Z_{imb} = 1\ k\Omega$, $\Delta Z = |Z_3 - Z_4| = 1\ k\Omega$ and Z_3 and Z_4 are much smaller than Z_1. $I_{imb}Z_{imb}$ was calculated to be approximately 1 mV. Although the common mode voltage of 1 mV is tolerable in the EIT applications, this is however for a situation where current sources were perfectly matched. Difficulties of matching of current sources was shown from the previous statistical analyses. Thus, I_{imb} would be expected to be significant. In the POW system, I_{imb} for sixteen channels varied from 0.8 μA to 10.6 μA for a drive current of 1 mA at 40.96 kHz when a uniform resistor-mesh was used as a test object.

However, if near ideal double-ended current sources are used, I_{imb} would be negligible as can be seen from Figure 6.6. Although it seems that double-ended current sources would be better in that sense, it however poses other problems when monolithic analogue multiplexers are used. Employing multiple double-ended current sources may be an alternative, but this requires further investigations.

To have a zero unbalanced current in all cases I_{push} must be equal to I_{pull}. Now, assuming that $I_{push} = ASin(\omega t)$ and $I_{pull} = BSin(\omega t+\phi)$, where ω is the angular frequency and ϕ is the phase difference between I_{push} and I_{pull}. The unbalanced current I_{imb} can then be written as

$$I_{imb} = -2ASin(\phi/2)Cos(\omega t+\phi/2) \pm \Delta ASin(\omega t+\phi) \quad . . . . . (6.10)$$

where $\Delta A = |A - B|$. Considering that a unity load ($Z_3 = Z_4 = 1\ \Omega$) is used (Figure 6.4). The voltage developed across Z_3 and Z_4 is

$$V_{34} = 2ACos(\phi/2)Sin(\omega t+\phi/2) \pm \Delta ASin(\omega t+\phi) \quad . . . . . (6.11)$$

In an ideal situation where $\Delta A = 0$ and $\phi = 0$, the true voltage value is $2ASin(\omega t)$. However, due to the presence of I_{imb}, the accuracy of the magnitude of the detected voltage is dependent upon ΔA. In conductivity imaging, to maintain a 0.1% accuracy for $\Delta A \leq 0.1\%$ of I_{push}, ϕ must not exceed 2.5°. In relative permittivity

imaging, considering that magnitudes of currents are perfectly matched and ϕ = 2.5°, the erroneous value of the imaginary part signal would be approximately 4% of A; ideally this should be zero.

In order to show the effect of unbalanced currents in the EIT applications with all sixteen sources connected to a phantom, computer simulations were performed. The changes of conductivity and permittivity were the same as shown in Table 6.2 (p.126). However, the current sources were deliberately made to be imperfect. The mean and SD values of the magnitude of unbalanced currents were set to 4.4 μA and 3.4 μA for a 1 mA drive. The mean and SD values of the phase of unbalanced currents were set to 6.8° and 0.73°. These were the values quoted from the author's system for 100 kHz (Table 3.2, p.59).

Reciprocity error (E_{re}) was evaluated as 1.6% for the simulated system with unbalanced current sources, and 0.09% for the simulated ideal system with identical sources. For the practical system, it was 5.8% (p.117). The discrepancy of E_{re} was partly due to difficulties of current source matching, and partly due to stray capacitances within the front-end electronics. The possible source of errors could be in the multiplexers at the receiving end although signals were buffered.

As a figure of measurement accuracy comparing against a theoretical ideal system, E_1 was evaluated as 1.8%. This value was obtained by taking the simulated voltages with balanced sources as $V_{c,i}$ and the simulated voltages with unbalanced sources as $V_{m,i}$ in Equation 5.9 (pp.118-9). With practical measurements of $V_{m,i}$ on the phantom, a value of 5.5% was obtained. The discrepancy of E_1 could be due to the characteristics of the attenuation network, and unbalanced current sources. However, both systems successfully produced data for image reconstructions. The reconstructed images were very similar to those illustrated in Figure 6.2, and thus are not shown. From this simulation study, it can be noticed that the unbalanced current does produce significant reciprocity error but they seem to be cancelled out with the Sheffield algorithm. Even so, the unbalanced current should be minimised to prevent deterioration of the image quality.

6.3 A Proposed Configuration for Low Crosstalk and Low Feedthrough Analogue Multiplexers and Demultiplexers

It was indicated from the preceding discussion that employment of double-ended current sources would alleviate the effect of unbalanced current and hence the accuracy of magnitude and phase measurements would be improved. However, using double-ended circuits would require the currents to be multiplexed which in turn poses the problems of crosstalk and feedthrough. This is the hindrance of using multiplexed current methods. If an analogue switch which features low crosstalk and low feedthrough, it would be ideal for EIT applications since device matching would be no longer required. In view of this, the author has proposed a configuration for such analogue switches.

It was considered that both multiplexer and demultiplexer should be made up of discrete components, such as 2N4858 which has a small channel capacitance of 10 pF and a leakage current of 250 pA. Although 16-channel analogue multiplier and demultiplexer such as HI506-5 and DG526CJ are available in the market, the switches are all built on a common substrate and so crosstalk remains due to the stray capacitance. If discrete components are used to implement a 16-channel multiplexer and demultiplexer, the problems of crosstalk between channels may be considerably reduced.

It is not thought to be possible to tackle the problem of feedthrough with just a careful selection of devices. Some special circuit topologies may be required. The proposed circuit configuration of a single switch is shown in Figure 6.7. A sample connection of a single channel in the EIT application is illustrated in Figure 6.8. A simplified ac-coupled voltage follower described by Lidgey [37] is used and the current is ac-coupled using C_1 for safety reason.

With reference to Figure 6.7, two switches (S_1 and S_2) which are connected in series are gated by a common signal (G) while the third one is controlled by the inverse of G, i.e. $\overline{G}$. When S_1 and S_2 are closed, a signal can pass through these switches to reach the output since S_3 is open. On the other hand, when S_1 and S_2 are open and S_3 is closed, the output is floating. Since the input of S_2 is at zero potential, the output of S_2 should not contain a fraction of the input signal resulting from feedthrough of S_1. A logic control unit is required to ensure a break-before-make action for all switches. Note that since this circuit configuration is used for multiplexing a current which generated from a current source, the channel

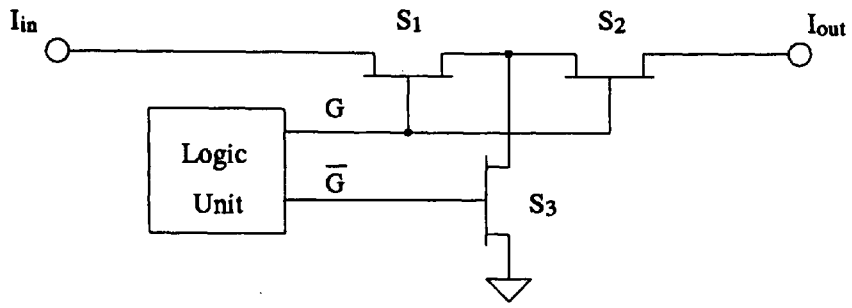


Figure 6.7 A proposed circuit topology of a single analogue switch

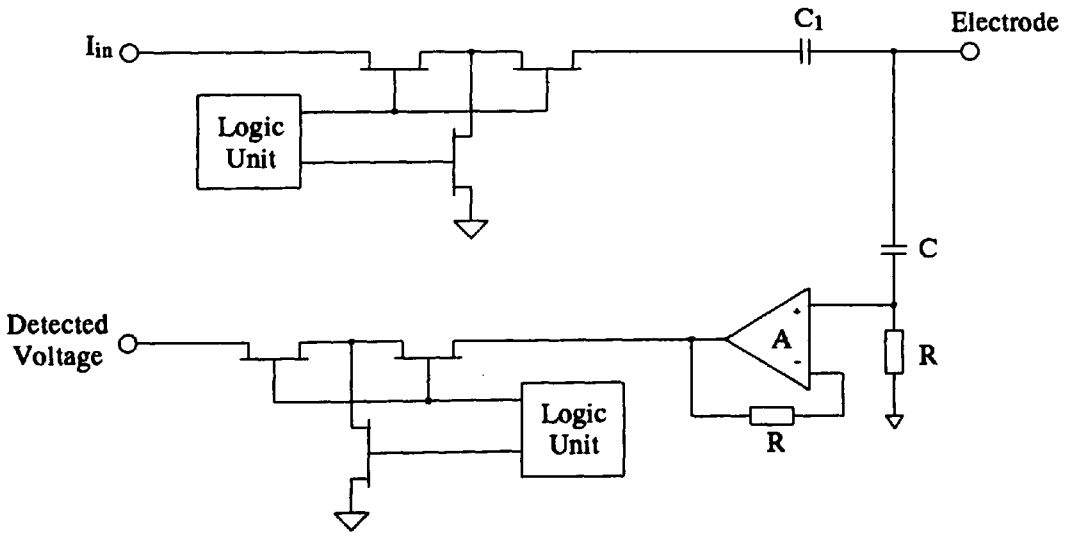


Figure 6.8 A sample arrangement for a single channel in EIT

resistance of each switch is therefore not critical.

6.4 Difference between Phase and Permittivity Images

As mentioned in chapter two, phase images are different from permittivity images. By definitions [12], phase images represent the distribution of the difference of phase angles (i.e. $\tan^{-1}(\omega\varepsilon_o\varepsilon/\sigma')$). Permittivity images represent the distribution of $\ln(\omega\varepsilon_o\varepsilon_2/\sigma_1')$. Thus, phase images are obtainable in all circumstances. Permittivity images, on the other hand, can only be produced when the reference has negligible reactance. Furthermore, the features of these two kinds of images are not identical.

This can be shown by re-considering the previous experiments (Table 6.2 and Figure 6.2). According to the set-up, the values of s′, s″, α and β were evaluated using Equations 2.6, 2.7, 2.12 and 2.13 and shown in Table 6.4.

	s′	s″	α	β
Region A	0.00498	0.0997	0	−2.30
Region B	0.694	0.0500	0.693	−2.30

Table 6.4 Images values for (A) n = 0.1 and m = 0.0;
(B) n = 0.1 and m = 2.0, at f = 40.96 kHz

As anticipated, the simulated permittivities of the two plug-boards were designed to be equal but an increase in conductivity was also simulated in region B. From the reconstructed image of s′ illustrated in Figure 6.2(a), only changes in conductivity in region B was observed as consistent with the true values of s′. The image of α showed similar features as can be noticed from their true values in Table 6.4. Figures 6.2(b) and (d) illustrate the distinction between imaging s″ and β. The image of s″ in Figure 6.2(b) shows changes in permittivities in both regions but the board in region B is fainter than that in region A as consistent with the true values of s″. On the image of β as illustrated in Figure 6.2(d), much closer local maximums are obtained in both regions A and B and thus have the similar intensity.

A similar experiment was performed using the UHW system. Similar results were

produced. This indicates that images of s″ are definitely different from images of β. This work has been accepted for publication [67] and will be presented in an international conference held in Florida, 1991.

6.5 *In-vivo* Imaging

In practice, *in-vivo* imaging poses additional problems for reconstruction since the distribution of equipotentials is not confined to the plane of electrodes; it also extends to the third dimension. As a result, the reconstructed image is distorted such that the features of the object are compressed toward the centre of the image. This is true for all reconstructed images using this two-dimensional reconstruction algorithm when the data has been acquired on a three-dimensional object. This phenomenon was investigated by making measurements on two metal rods each of diameter 18 mm immersed in a cylindrical tank containing saline solution of conductivity 0.1 S/m. The diameter of the tank was 19 cm and the separation between the two bars was 10 cm approximately. The reconstructed image was shown in Figure 6.9. It can be seen from the image that the bars tended to merge together when using a two-dimensional image reconstruction algorithm. To improve the image quality three-dimensional image reconstruction algorithms would be required.

Although this difficulty of measurements and reconstructions exists in *in-vivo* imaging, some static *in-vivo* conductivity images on a forearm [2,15,68] and thorax [68] have been reported. The image of a human forearm was obtained by taking two frames of measurements. One of them was taken from a homogeneous reference such as saline solution. This kind of imaging benefits from minimum electrode contact impedances. The lung imaging produced an image representing the changes in resistivity of the lungs between expiration and inspiration. Electrodes are required and hence contact impedances were a major problem in measurements. Another kind of static *in-vivo* conductivity images can be obtained by means of the dual-frequency imaging technique [13,34]. This image is reconstructed using two frames of data which are collected at two different frequencies. An *in-vivo* image was reported using this technique [33]. All these published *in-vivo* images represent the distribution of resistivity of different tissues. In the followings, several experiments were carried out and intended to display the distribution of relative permittivity of the human tissues as well as conductivity. An alternating current of 2.5 mA r.m.s at 40.96 kHz was used in all *in-vivo*

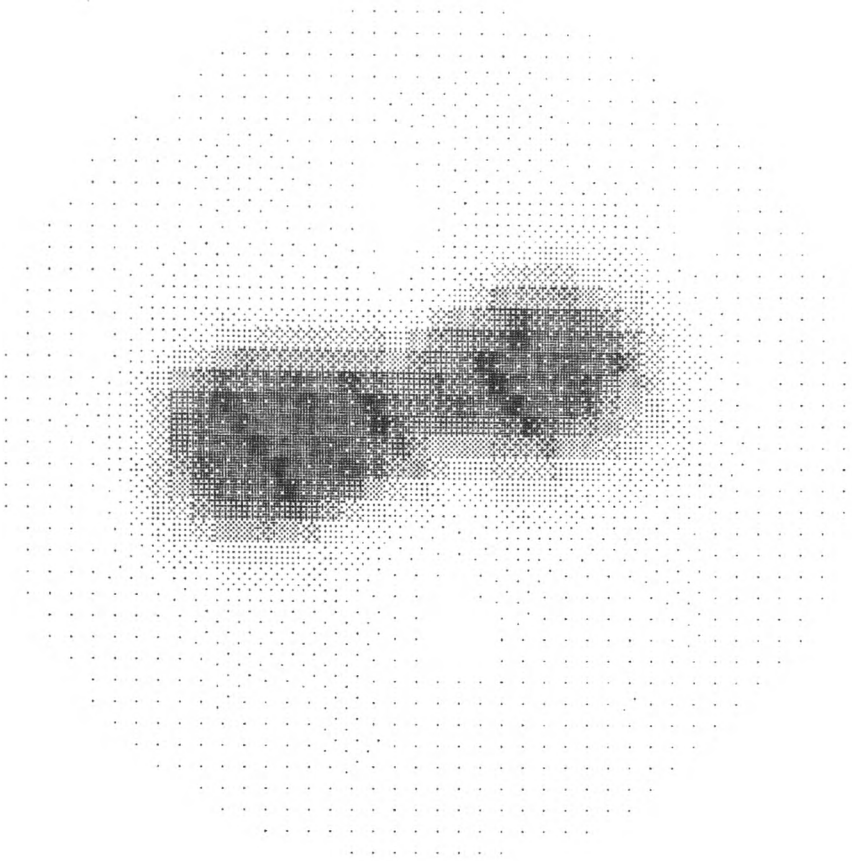


Figure 6.9 A conductivity image of two metal bars immersed in a saline tank

measurements performed with the author's system.

6.5.1 Forearm Imaging

The experiment was carried out using an array of sixteen electrodes mounted on a cylindrical tube of diameter 90 mm. The tube was immersed in a saline bath. The conductivity of the saline was 0.1 S/m. The first frame of data was collected with the saline solution alone and was used as a reference frame. The second frame of data was collected with a human forearm positioned in the middle of the electrodes. Images of s′, s″, α and β were then reconstructed. Using the reported values of electrical conductivity and permittivity for mammalian tissues, the values of s′, s″, α and β are calculated and shown in Table 6.5.

	s′	s″	α	β	
muscle	1.34	0.307	1.28	0.131	Surowiec *et al.* [8]
muscle(T)	1.61	0.66	1.37	1.12	Zheng *et al.* [74]
muscle(L)	0.51	0.038	0.51	−2.76	Zheng *et al.* [74]
bone	−2.51	0.161	−2.53	−4.34	Smith and Foster [7]
fat	−1.12	0.012	−1.12	−5.54	Smith and Foster [7]

Table 6.5 Image values of different tissues at 40.96 kHz with reference to a saline solution of conductivity of 0.1 S/m; T : transverse; L : longitudinal

Figure 6.10(a) illustrates the image of s′. The boundary of the forearm was clearly defined. The image was dominated by the muscle which appeared as increased conductivity. The minimum and maximum image values were 0.00623 and 1.27. A lower conductivity region was detected in the positions of the bones. However, two bones were not clearly separated in the image which could be probably due to the perturbation of equipotentials outside the plane of electrodes as was evident for the metal rods in Figure 6.9. On the left hand side of the image, reconstruction artifacts did occur when the separation between the surface of the forearm and the electrodes reduced. Similar features were obtained in the image of α, Figure 6.10(c). Its image values were −0.106 and 1.18.

Figure 6.10(b) illustrates the image of s″. The boundary of the forearm was still

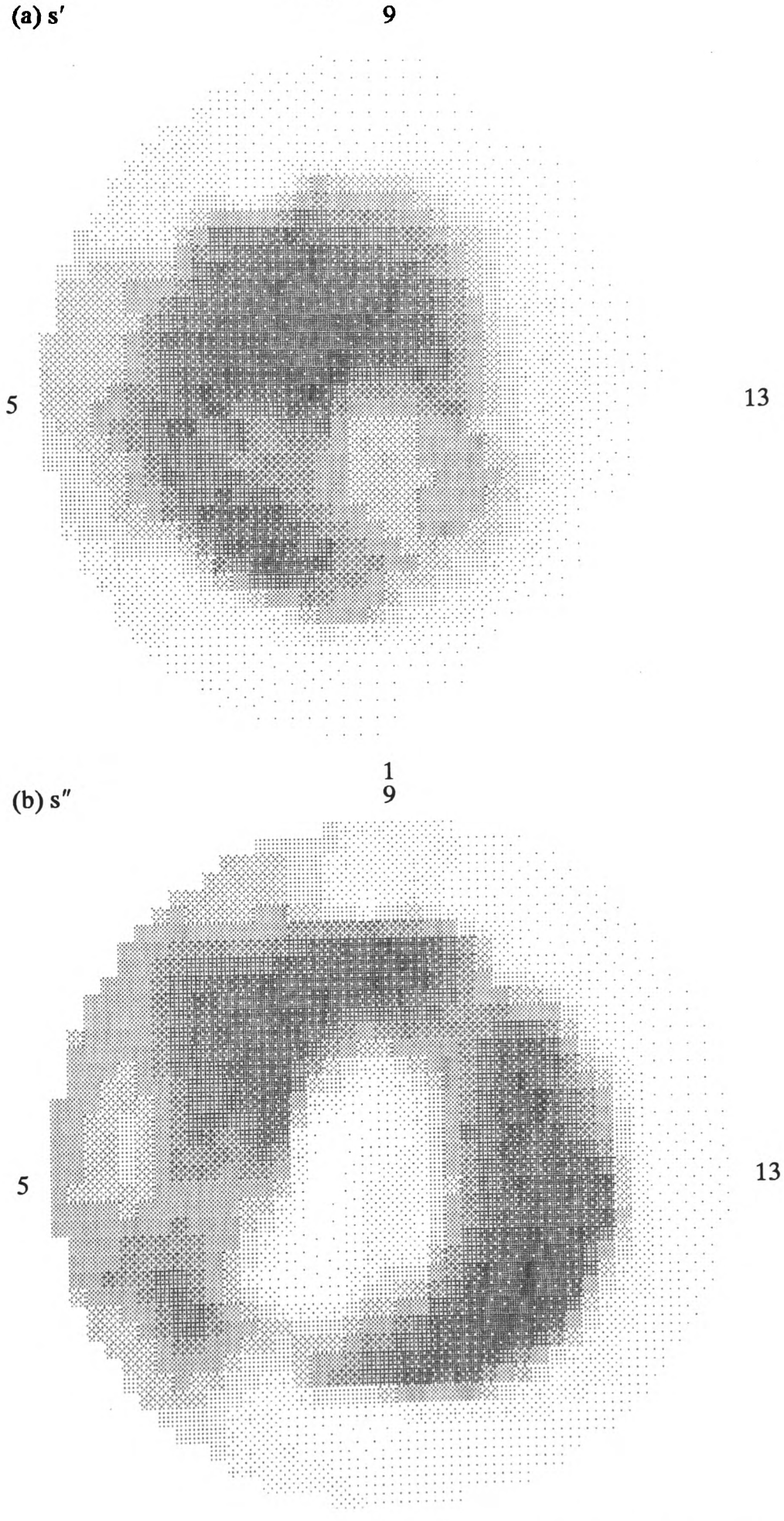
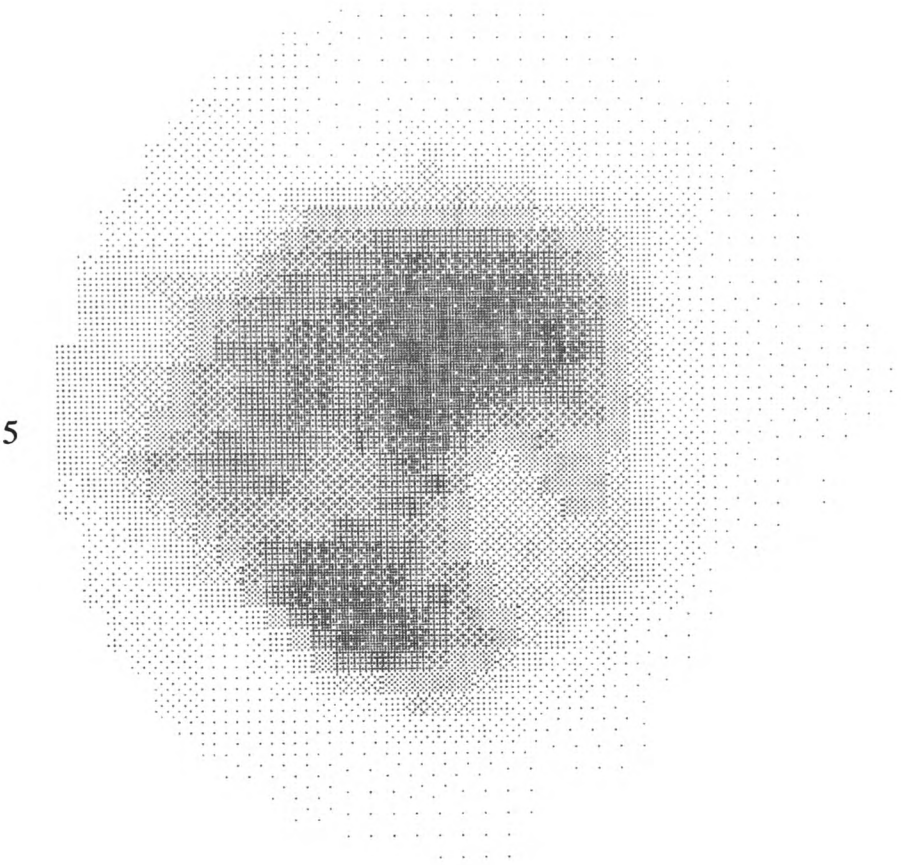


Figure 6.10 Forearm images

(c) α



(d) β

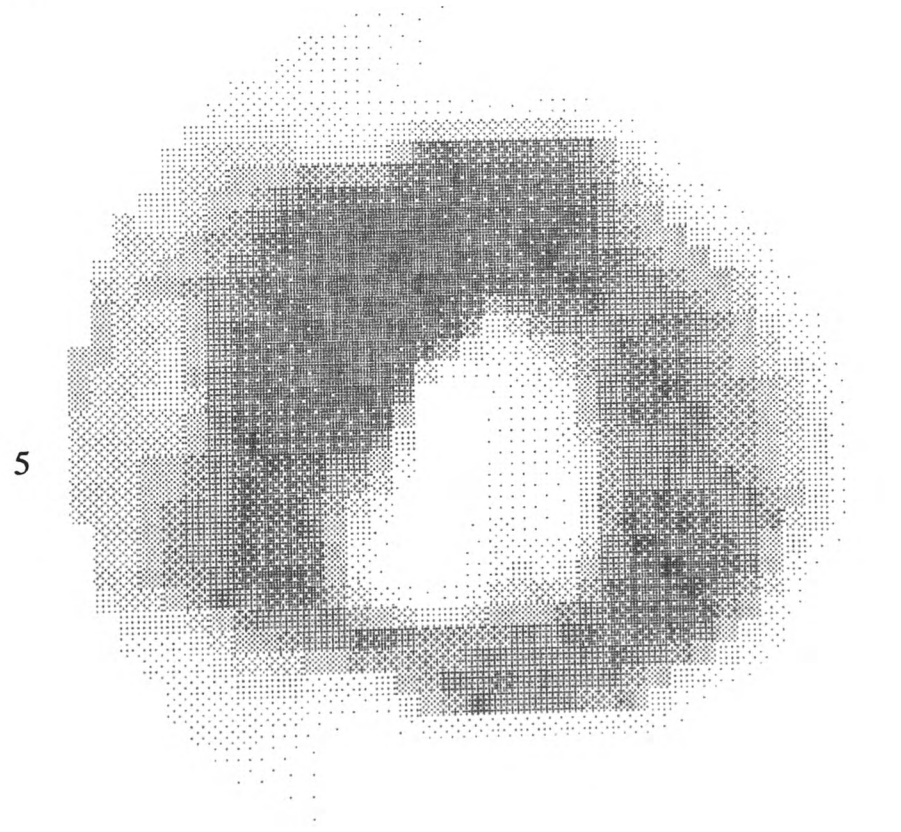


Figure 6.10 Forearm images

clearly defined. The image values were 0.104 and 0.723. According to the figures shown in Table 6.5, an increase in phase angle was resulted due to the perturbation by muscle. Again, the image was dominated by muscle. Since muscle with electric field transverse to the fibres exhibits a much large phase change than with electric field aligned with the fibres across the EIT frequency band [74], the un-even phase change appearing in the image could be probably due to the complex structure of fibres in the plane of electrodes. The location of bones did not show distinctly although a slight increase in phase angle was expected.

The image of β was illustrated in Figure 6.10(d). Muscle was the dominant substance appeared in the image. The central region showed a reduction in permittivity could be partly due to the electrical property of bones, and partly due to the difficulty of measurements performed on a three-dimensional object.

In comparison with other reported conductivity images of the human forearm, better images were shown when data was collected using serial drive methods although this may have been due to a better electrode arrangement [68,73]. In addition, some kinds of geometric correction processes were performed on the reported data but the method of the correction was not reported [68].

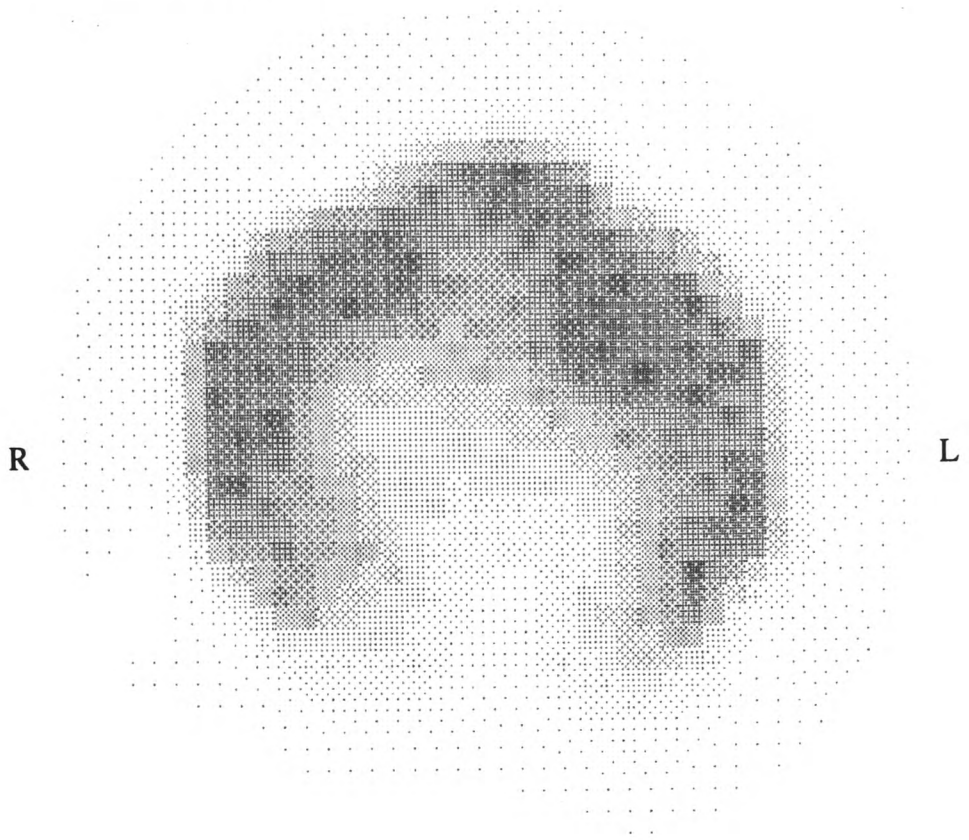
6.5.2 Lung Imaging

A ring of sixteen ECG electrodes was attached around the thorax 2 cm below nipple level on a volunteer. The attenuation factors were set after inspiration. Two frames of measurements were taken after expiration and inspiration. Images were reconstructed from data measured at total lung capacity referenced to a measurement after expiration. Both the UHW system and the POW system were used to collect the data with the same set of electrodes in order to make comparisons.

Figures 6.11(a) and 6.11(c) show the reconstructed images of s' obtained from the UHW system and the POW system respectively. The lungs themselves were well visualised as two regions of increased conductivity. In Figure 6.11(a), the lungs seemed to merge together at the anterior position. Figure 6.11(c) illustrates a similar feature at that position. Therefore, this could be due to the actual shape of the lungs of the volunteer. Image values of all images are shown in Table 6.6.

Figures 6.11(b) and 6.11(d) illustrate images of s''. The boundary of the lungs was clearly defined in Figure 6.11(b). According to the reconstructed image, both

(a) s′: The UHW system



(b) s″: The UHW system

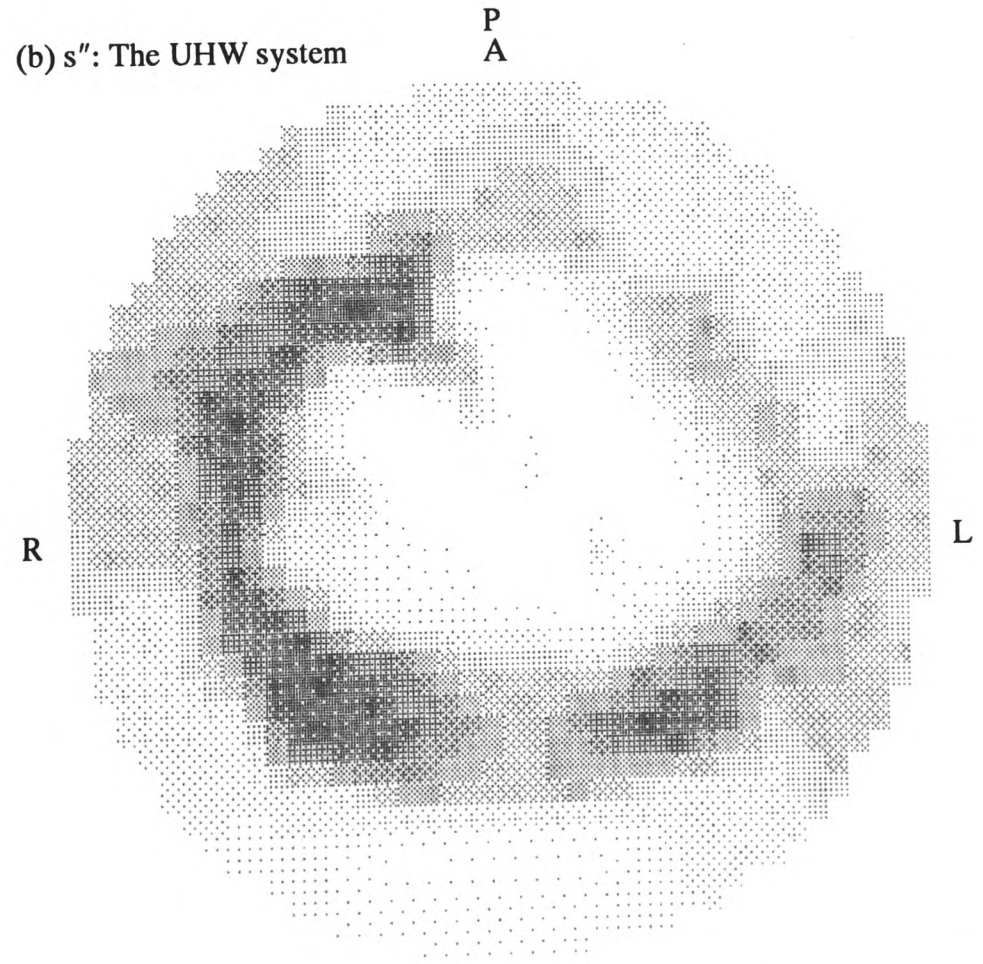
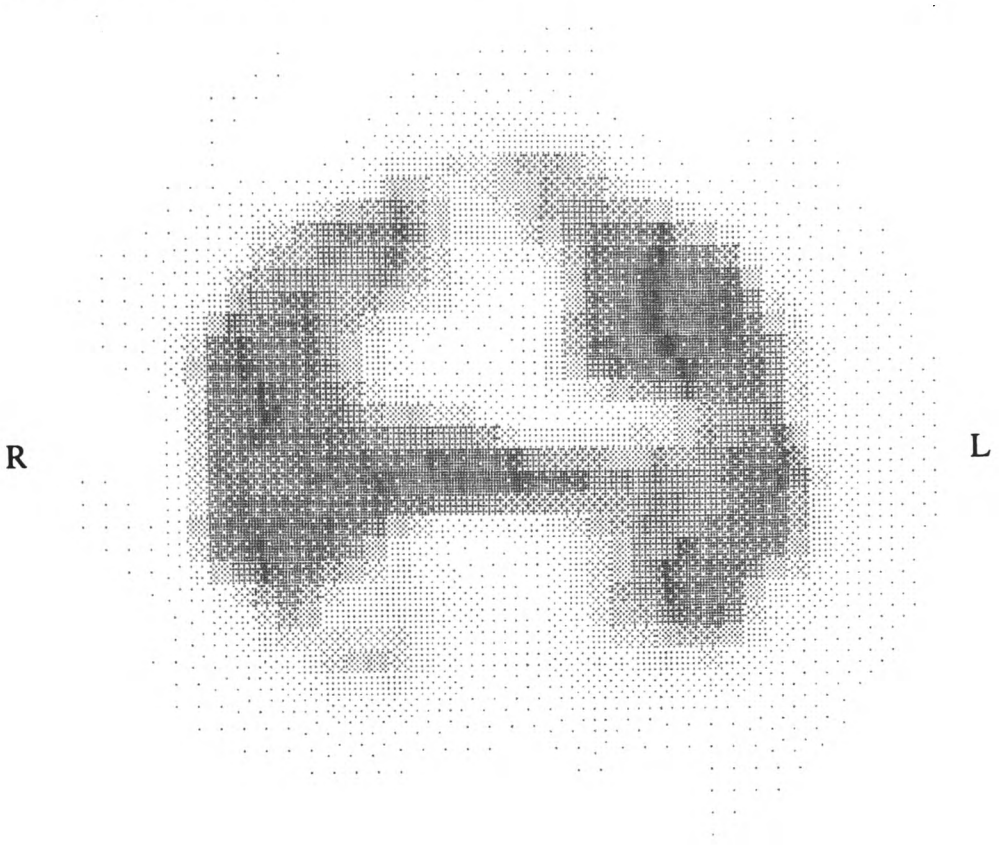


Figure 6.11 Lung images

(c) s′: The POW system



(d) s″: The POW system

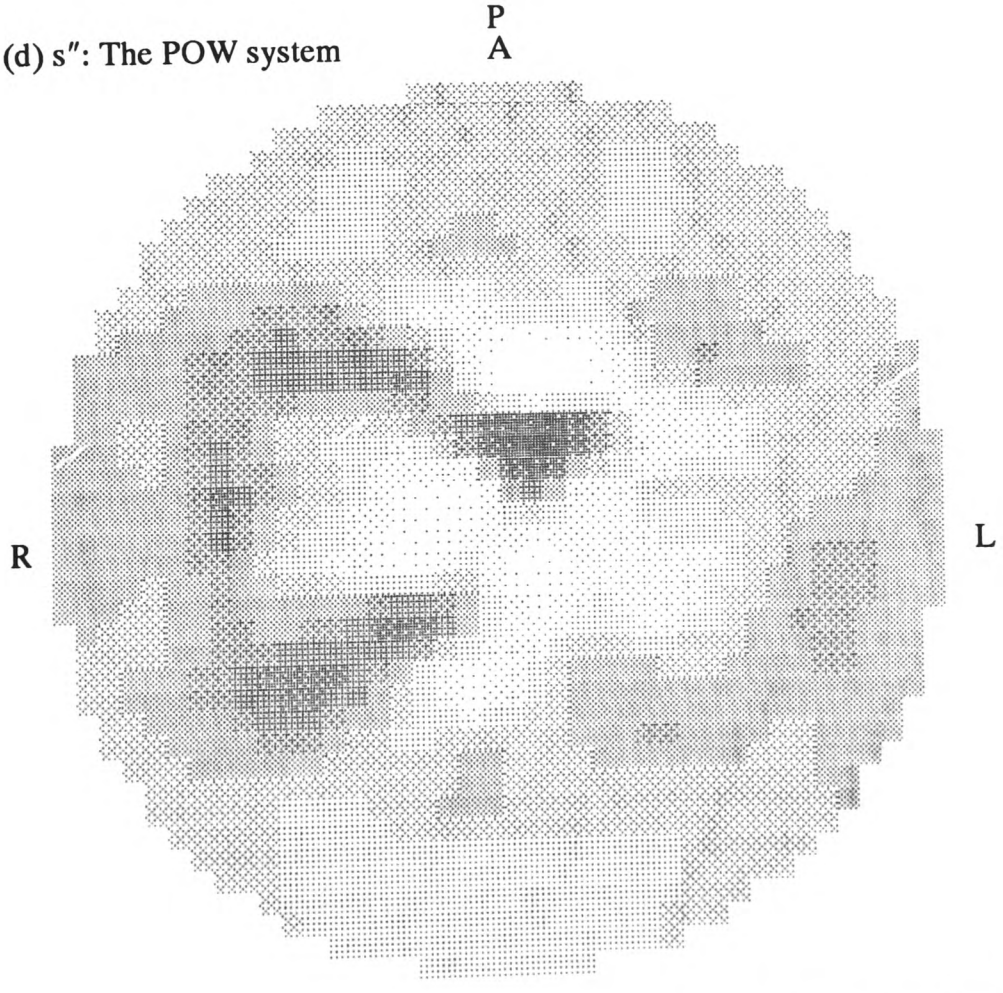


Figure 6.11 Lung images

	UHW system		POW system	
	s′	s″	s′	s″
minimum	-0.0418	-0.248	-0.0749	-0.363
maximum	0.423	0.1	0.558	0.345

Table 6.6 Image values of s′ and s″ of lung

positive and negative changes of significant magnitude in comparison with the value of s′ were detected. There is a central negative region surrounded by a positive region extending out to the boundary of the lungs. In contrast, Figure 6.11(d) just managed to show the boundary of the lungs. In the central region, the appearance was slightly different from that illustrated in Figure 6.11(b) since negative changes were also detected. If Figure 6.11(b) indicated the true changes, the un-even changes in Figure 6.11(d) might be explained due to the higher random noise level of the POW system. In addition to random noise, this could also be due to inconsistent expiration and inspiration between measurements.

	UHW system		POW system	
	s′	s″	s′	s″
minimum	-0.0327(78.2%)	-0.0209(8.4%)	-0.181(242%)	-0.0907(25.0%)
maximum	0.0568(13.4%)	-0.0143(14.3%)	-0.105(18.8%)	0.128(37.1%)

Table 6.7 Image values of s′ and s″ of noise in lung imaging

Noise estimations of these lung images were performed by reconstructing images of two measurements which were taken after inspiration. Image values together with a coarse estimation of noise for both systems are shown in Table 6.7. The reconstructed images were not displayed as they do not contain any useful information. A large percentage error occurred at the minimum image value of s′ is because the smallest change in conductivity is close to zero, and hence the size of these signals is comparable to that of noise. With reference to the figures of s′, both systems contained error of less than 20%. Thus, the reconstructed images were comparable. However, when comparing s″, a much larger error was obtained with the POW system. The image shown in Figure 6.11(d) could therefore be a "noisy" version of that in Figure 6.11(b).

A few experiments were carried out on a group of volunteers using the UHW system. Images of s″ show similar features on all volunteers but cannot be fully interpreted due to lack of information on the permittivities of tissues in this frequency range. This work has been accepted for publication [75].

Finally, similar experiments were also performed using the POW system. However, the results were not consistent. This could be due to the combinational effect of skin contact impedances which makes detectable phase changes smaller, and random noise.

From all experiments conducted, it seems that the effect of skin contact impedance together with the effective capacitance between the measurement electrodes creates difficulties in *in-vivo* imaging. The unbalanced current due to device matching does cause a significant increase in reciprocity error. However, it was evident by means of computer simulation that the reciprocity error does not seriously affect the quality of the reconstructed images. The effect of unbalanced currents seems to be cancelled out in the reconstruction processes.

Chapter 7

Discussion

At the outset of the project, a number of aspects of work were projected. They were (i) to review EIT system development, (ii) to investigate constant amplitude current source circuits, (iii) to design and develop a complete impedance tomography system, (iv) to develop methods for the assessment of system performance, and (v) to study various aspects concerning complex impedance imaging. These have been addressed as follows.

(i) Literature review

A comprehensive literature survey has been maintained throughout the project. From the review of the EIT system development, it was found that the front-end architecture employed by the Oxford group was suitable for the proposed system. It was thus adopted and modified to suit relative permittivity imaging. During the course of studies, one recently reported ac-couple voltage follower [37] seems to be useful in EIT applications. The circuit requires only one VOA and a few passive components. Although its input impedance does vary with frequency, under a screen driven condition, its value is in the order of MΩ within the EIT frequency band. It incorporates safety precaution and benefits from a high input impedance. Thus, it can be expected that it could be the best buffering circuit presently available for use in EIT. The author's system did not employ this circuit. It is because the circuit was reported at the later stage of the project.

(ii) Investigations into current source circuits

A current source circuit review was conducted. In the light of the reviewed circuits, the VOA current supply sensing circuit [63] was found to be the best one among the others due to its anticipated high output impedance, relatively wideband performance, and bilateral single-ended output configuration. A replica of such circuit was tested but unsatisfactory results were obtained. Consequently, a new single-ended current source circuit was designed and developed based upon the supply current sensing technique.

Development of a new single-ended current source - this work was presented in an EEC workshop in Copenhagen in 1990 [64]. The circuit adopted the VOA supply current sensing technique but replaced the VOA with a resistively biased complementary emitter follower. With this substitution, the d.c. offset at the circuit output was found to be insensitive to changes of input drive voltage. This is important when a current source was chosen as a stimulus circuit in the EIT applications. This is especially so when a current source is intended for use in adaptive current tomography. This circuit, however, required a highly stable power supply.

Under an ideal situation where transistor parameters are perfectly matched, the output impedance was found to be approximately 2.5 MΩ at 10 kHz and 100 kHz. In practice, typical output impedances of 830 kΩ and 2.0 MΩ were obtained. The better performance occurring at the higher frequency could not be explained logically. The sole explanation for this rare phenomenon was the transistors became more closely matched at the higher frequency.

In addition to functional studies, statistical analyses were also performed. The simulation results revealed that a variation of output impedance of 55% was obtained at 10 kHz in comparison with 26% at 100 kHz. In practice, percentages of the output impedance variation were found to be 12.9% and 2.6% at 10 kHz and 100 kHz.

Within the sixteen current source circuits, a mean phase shift of about 6.8° was obtained. It was found to be due to the characteristics of the complementary emitter follower. It was also found that the phase difference between the output of the emitter follower and the output of the current source was negligible at 10 kHz and 100 kHz. This indicated that the output impedance of the current source contains negligible capacitive component.

(iii) Design and development of a complete impedance tomography system

Completion of the design and construction of an electrical impedance tomography system - a fully functional instrumentation system for investigations of electrical impedance tomography was developed. The architecture of the author's system has been designed in such a way that future studies on other aspects, such as dual-frequency imaging, will be possible. Since the data collection system is under microprocessor control, emerging

ideas may be implemented without significant changes in hardware. The system operations are fully controlled by the user via a host computer. The system is capable of applying stimulus, collecting peripheral voltages, and reconstructing and displaying impedance images.

Completion of the design and construction of a user interface program - an ninterface program used to control the data collection system was developed on a IBM compatible PC. The program is menu driven and simple to use. Thus, only limited expertise is required to operate the system. A modular programming structure is employed for ease of future developments.

Since data transmissions between the data collection system and the PC are required, a novel data checking scheme was proposed and implemented. Basically, the scheme generates three additional bytes of data based upon the transmission data. Both the data collection system and the host computer perform the same arithmetic checking operations. If these three additional data generated in the data collection system are consistent with those generated in the host computer, the transmission is considered to be successful. Otherwise, it would be thought that data corruption has occurred in the transmission process. No limitation on the word length and virtually no extra transmission time are the merit of this checking scheme.

(iv) Development of methods for the assessment of system performance

In order to address this aspect, a suitable phantom was required. As a result, hardware and software phantoms were developed. The methods for the assessment of system performance were then devised.

Fabrication of a two-dimensional resistor-mesh phantom using surface mount technology (SMT) - in view of the advantages of the two-dimensional resistor-mesh phantom and its difficulty of low volume manufacture using conventional technology, an improved version was fabricated by means of SMT. The advantages of this SMT phantom are numerous - reliability, resistance to shock and relative ease of production. In addition, more resistors may be included without significant increases in the overall size. Stray capacitance is small and its effects are identical for all boards.

This phantom was presented in the EEC EIT Workshop at Cardiff in 1990 and

was accepted as a standard test object for the assessment of system performance within the CAIT research programme of the EEC.

A novel application of analogue circuit simulator - since the phantom mentioned above is made up of electrical passive components, it was considered that a sophisticated analogue circuit simulator might be used in order to develop a computer model. Such a model offers additional capabilities which other reported software models cannot be readily achieved, e.g. statistical analyses relating to parameter variation. This is a useful feature as evaluations of the percentage variations of the predicted peripheral voltages for a given component tolerance value are often required. If the variations are within an allowable tolerance (0.1%), a set of computed data may be adequate for a set of manufactured resistor-mesh phantoms provided that the same tolerance is used throughout.

Development of criteria for assessment of system performances - by virtue of the hardware and software phantoms, some criteria for system characterisations were devised. Random noise and systematic errors are the quantities to be measured. In addition to these quantities, a measure of the deviation of recorded data from the theoretical expectation was also defined. The author has used these criteria to benchmark two data collection systems; Griffiths's system and the author's system. It was found that Griffiths's system produced better results in both the random noise and systematic error estimations. Lower random noise level in Griffiths's system was due to longer integration time in the demodulation process. Smaller systematic errors were due to the employment of a multiplexed current configuration.

According to these results, it indicates that a serial type drive configuration (e.g. Griffiths's system) gives better noise performances in comparison to a parallel type drive configuration (e.g. the author's system). If the effect of stray capacitance exhibited in the analogue demultiplexers can be reduced, the serial type system should be the best front-end drive configuration.

(v) Various aspects concerning complex impedance imaging

A few investigations relating to this issue were performed. They are summarised as follows.

Effect of contact impedances (resistance) on complex impedance imaging - the effect of stray capacitance between the drive and measurement electrode pairs is well known in impedance imaging. The influence of the stray capacitance between the measurement electrode pairs on the detected voltage is negligible in conductivity imaging. This has been confirmed by means of theoretical analyses and practical experiments. However, it is not the case when relative permittivity imaging is intended. The degree of influence was found to be dependent upon both the contact impedance and the effective capacitance between the measurement pair.

The author has investigated the influence on the detected voltages. From the theoretical analyses, an approximate 77% of the true phase change could be detected for typical contact impedances and stray capacitance values. In a physical phantom study, the reconstructed images showed that the appearance of the conductivity images were similar with or without the simulated contact impedance. However, the phase and permittivity images were contaminated by noise as well as systematic errors when contact impedances were added.

According to the theoretical and practical investigations, it is evident that good quality *in-vivo* permittivity images would be very difficult to obtain with the present technology unless the skin-electrode contact impedance could be reduced to a few hundred ohms.

Identification of the difficulties of multiple current source systems - in addition to the problems of the stray capacitance and contact impedance, topologies of the drive methods were found to significantly influence the quality of the detected peripheral voltages. Using a multiple current source system, stray capacitances between the drive and measurement electrode pairs were, in principle, reduced to a minimum. However, the production of identical current sources was understood to be impossible due to variations of electrical characteristics in electronic components. Even though good matched single-ended current sources may be produced, problems are still present in the EIT applications. When two sources combined to produce a sink/source current pair, an unbalanced current was produced unconditionally due to finite output impedances of the current sources.

It was evident from the computer simulation and practical experiment that the reciprocity error increased from 0.09% (ideal system) to 1.6% (simulated

system) or to 5.8% (practical system). Fortunately, the quality of reconstructed images seems not to be affected. This is due to the normalised feature of the reconstruction algorithm adopted which virtually eliminates the common errors.

On the other hand, a double-ended current source could, in principle, produce a good sink/source current pair and negligible unbalanced current. When one current source is used and a multiplexing technique is employed, stray capacitance will be the sole problem to be tackled. Thus, if good demultiplexers were found to be available, a single current source operating in a multiplexing mode seems to be the best configuration.

In-vivo imaging - measurements were also performed on human subjects. Experiments were carried out using two different data collection systems. As a result, the system developed by Griffiths produced better images as compared to those of the author's system. For images of s′, a noise estimation showed that both systems contained noise of less than 20% of their maximum image values, and so images of s′ produced by both systems were comparable. However, the image of s″ produced by the author's system was noisy. This could be due to the difference of dwell time between the two systems. The dwell time of Griffith's system was about 20 times longer than that of the author's system.

In addition, the following aspects were addressed during the course of studies.

Proposal for two alternative data collection methods - two methods of data collection were proposed. Neither methods require analogue nor digital multiplication processes. One uses sample-and-hold and counter type circuits to extract the peak and phase information contained in the peripheral voltage signals. The other uses sample-and-hold circuits only; samplings at quadrature intervals produces the real and imaginary information. These kinds of data collection method potentially provide a means for fast information extraction since no low-pass filtering is required. Additionally, "noisy" multipliers are not needed and thus accuracy of the measured voltages would be improved. Due to the time constraints, these two methods have not be implemented in the author's system for further investigations. One consideration in implementing these methods is the quality of the sample-and-hold circuits. The author suggested that the monolithic type sample-and-hold circuit (AD683) is used due to its low droop rate, short

aperture time and small feedthrough.

Proposed arrangement for low crosstalk and low feedthrough analogue multiplexers and demultiplexers - a special arrangement for low crosstalk and low feedthrough analogue multiplexers was proposed. The proposed arrangement is to use discrete analogue switches, e.g. field effect transistors, to construct a multiplexer in order to minimise the crosstalk. To alleviate the effect of feedthrough three switches are connected in such a way that the effect of the channel-off capacitance becomes insignificant. Thus, when the switch is open, the undesired signal which will be present at the output due to feedthrough is negligible.

Note that the stray capacitance effect is still present in the proposed topology since parasitic capacitance to ground can never be removed completely in practice. Nevertheless, it is a constant quantity and will be eliminated in the EIT applications. This proposal may provide a solution to the problem of stray capacitance effect in the analogue multiplexers and demultiplexers. Therefore, the serial type drive configuration may be re-considered in relative permittivity imaging.

Appendices

		Page
Appendix A	References	154
Appendix B	Operating Codes for the Polytechnic of Wales System	158
Appendix C	Memory Map for the Data Collection System	159
Appendix D	Program Listings of Modula-2	160
Appendix E	A Program Listing of 6303 Assembly Languages	195
Appendix F	Theoretical Data of the Two-dimensional Resistor-mesh Phantom (SMT)	205
Appendix G	Circuit Diagrams of the Polytechnic of Wales System	211

Appendix A References

[1] Brown,B.H.; Barber,D.C. and Freeston,I.L. "Potential tomography", U.K.Patent 2119520, April 1982

[2] Barber,D.C. and Brown,B.H. "Applied potential tomography", *J.Phys.E:Sci.Instrum.*, Vol.17, 1984, pp.723-33

[3] Seagar,A.D.; Barber,D.C. and Brown,B.H. "Electrical impedance imaging", *IEE Proceedings*, Vol.134, Pt.A, No.2, Feb., 1987, pp.201-10

[4] Barber,D.C. and Brown,B.H. "Recent developments in applied potential tomography - APT", *Proc. 9th Conf. - Information Processing in Medical Imaging*, Washington D.C., Jun.10-14, 1985, pp.106-21

[5] Barber,D.C.; Brown,B.H. and Freeston,I.L. "Experimental results of electrical impedance imaging", *Proc. 6th Int. Conf. on Electrical Bio-impedance (ICEBI)*, Zadar, Yugoslavia, 1983, pp.1-5

[6] Brown,B.H. and Seagar,A.D. "The Sheffield data collection system", *Clin.Phys.Physiol.Meas.*, Vol.8, suppl.A, 1987, pp.91-7

[7] Smith,S.R. and Foster,K.R. "Dielectric properties of low-water-content tissues", *Phys.Med.Biol.*, Vol.30, No.9, 1986, pp.965-73

[8] Surowiec,A.; Stuchly,S.S.; Keaney,M. and Swarup,A. "*In vivo* and *in vitro* dielectric properties of feline tissues at low radiofrequencies", *Phys.Med.Biol.*, Vol.31, No.8, 1986, pp.901-9

[9] Mangnall,Y.F.; Baxter,A.J.; Avill,R.; Bird,N.C.; Brown,B.H.; Barber,D.C.; Seagar,A.D.; Johnson,A.G. and Read,N.W. "Applied potential tomography: a new non-invasive technique for assessing gastric function", *Clin.Phys.Physiol.Meas.*, Vol.8, suppl.A, 1987, pp.119-29

[10] Griffiths,H. and Zhang,Z. "A dual-frequency electrical impedance tomography system", *Phys.Med.Biol.*, Vol.34, No.10, 1989, pp.1465-76

[11] Griffiths,H. "The importance of phase measurement in electrical impedance tomography", *Phys.Med.Biol.*, Vol.32, No.11, 1987, pp.1435-44

[12] Griffiths,H. "A phantom for electrical impedance tomography", *Clin.Phys.Physiol.Meas.*, Vol.9, suppl.A, 1988, pp.15-20

[13] Griffiths,H. and Ahmed,A. "A dual-frequency applied potential tomography technique: computer simulations", *Clin.Phys.Physiol.Meas.*, Vol.8, suppl.A, 1987, pp.103-7

[14] Bates,R.H.T.; Mckinnon,G.C. and Seagar,A.D. "A limitation on systems for imaging electrical conductivity distributions", *IEEE Transactions on Biomedical Engineering*, Vol.BME-27, No.7, Jul., 1980, pp.418-20

[15] Barber,D.C.; Brown,B.H. and Freeston,I.L. "Imaging spatial distributions of resistivity using applied potential tomography", *Electronics Letters*, Vol.19, No.22, Oct.27, 1983, pp.933-5

[16] Murai,T. and Kagawa,Y. "Electrical impedance computed tomography based on a finite element model", *IEEE Transactions on Biomedical Engineering*, Vol.BME-32, No.3, Mar., 1985, pp.177-84

[17] Gong,L.; Liu,Y.; Liu,P. and Zhang,K. "Algorithm and simulation of electrical impedance imaging", *Electronics Letters*, Vol.25, No.7, Mar.30, 1989, pp.474-6

[18] Webster,J.G. (1990) **Electrical impedance tomography**, U.K.:Adam Hilger IOP Publishing Ltd.

[19] Kim,Y.; Tompkins,W.J. and Webster,J.G. "A three-dimensional modifiable body model for biomedical applications", *Proc. IEEE Conf. on Frontiers of Engineering in Health Care*, 1981, pp.8-12

[20] Goble,J. and Isaacson,D. "Optimal current patterns for three-dimensional electric current computed tomography", *IEEE EMBS*, Vol.11, 1989, pp.463-4

[21] Liu,W.P.; Hua,P. and Webster,J.G. "Three-dimensional reconstruction in electrical impedance tomography", *Clin.Phys.Physiol.Meas*, Vol.9, suppl.A, 1988, pp.131-5

[22] Ziya Ider,Y. and Gencer,N.G. "An algorithm for compensating for 3D effects in electrical impedance tomography", *IEEE EMBS*, Vol.11, 1989, pp.465-6

[23] Barber,D.C. and Seagar,A.D. "Fast reconstruction of resistance images", *Clin.Phys.Physiol.Meas.*, Vol.8, suppl.A, 1987, pp.47-54

[24] Yorkey,T.J.; Webster,J.G. and Tompkins,W.J. "Comparing reconstruction algorithms for electrical impedance tomography", *IEEE Transactions on Biomedical Engineering*, Vol.BME-34, No.11, Nov., 1987, pp.843-52

[25] Yorkey,T.J and Webster,J.G. "A comparison of impedance tomographic reconstruction algorithms", *Clin.Phys.Physiol.Meas*, Vol.8, suppl.A, 1987, pp.55-62

[26] Kim,Y.; Webster,J.G. and Tompkins,W.J. "Electrical impedance imaging on the thorax", *Journal of Microwave Power*, Vol.18, 1983, pp.245-57

[27] Wexler,A.; Fry,B. and Neuman,M.R. "Impedance-computed tomography algorithm and system", *Appl. Opt.*, Vol.24, 1985, pp.3985-92

[28] van Oosterom,A.; de Boer,R.W. and van Dam,R.Th. "Intramural resistivity of cardiac tissue", *Med.&Biol.Eng.&Comput.*, Vol.17, May, 1979, pp.337-43

[29] Gielen,F.L.H. and Bergveld,P. "Comparison of electrode impedances of Pt, PtIr (10%Ir) and Ir-AIROF electrodes used in electrophysiological experiments", *Med.&Biol.Eng.&Comput.*, Vol.20, Jan., 1982, pp.77-83

[30] Patterson,R.P. "Sources of the thoracic cardiogenic electrical impedance signal as determined by a model", *Med.Biol.Eng.Comput.*, Vol.23, 1985, pp.411-7

[31] Newell,J.C.; Gisser,D.G. and Isaacson,D. "An electric current tomograph", *IEEE Transactions on Biomedical Engineering*, Vol.35, No.10, Oct., 1988, pp.828-33

[32] Skidmore,R.; Evans,J.M.; Jenkins,D. and Wells,P.N.T. "A data collection system for gathering electrical impedance measurements from the human breast", *Clin.Phys.Physiol.Meas*, Vol.8, suppl.A, 1987, pp.99-102

[33] Griffiths,H and Zhang,Z. "Dual-frequency impedance tomography *in-vitro* and *in-vivo*", *IEEE EMBS*, Vol.11, 1989, pp.476-7

[34] Zhang,Z. "Dual-frequency electrical impedance tomography", Ph.D. dissertation, The University Hospital of Wales, Cardiff, U.K., 1991

[35] British Standard BS5724

[36] Record,P.M.; Gadd,R. and Rolfe,P. "A signal conditioning electrode for use in EIT - an integrated approach", *Proc. of a meeting on Electrical Impedance Tomography*, ? pp.168-74

[37] *Proc. of EEC Workshop on Hardware for EIT*, Leuven, CAIT, Dec., 1990

[38] Gisser,D.G.; Issacson,D. and Newell,J.C. "Current topics in impedance imaging", *Clin.Phys.Physiol.Meas*, Vol.8, suppl.A, 1987, pp.39-46

[39] Murphy,D. "OXPACT system development", *Oxford Polytechnic Internal Report*, Department of Computing & Mathematical Sciences, Oxford Polytechnic, 1987

[40] McLeod,C.N.; Breckon,W.R. and Murphy,D. "OXPACT: Development of an adaptive current tomography", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.191-7

[41] Lidgey,F.J.; Toumazou,C. and Markis,C. "Instrumentation amplifiers", *Electronics & Wireless World*, Vol.96, No.1638, Apr., 1989, pp.344-5

[42] Toumazou,C. and Lidgey,F.J. "Novel current-mode instrumentation amplifier", *Electronics Letters*, Vol.25, No.3, Feb.2, 1989, pp.228-30

[43] Smith,R.W.M.; Brown,B.H. and Freeston,I.L. "Electrical impedance tomography", *Biomedical Application of Digital Signal Processing, IEE Electronics Division Colloquium*, Nov.30, 1989

[44] Seagar,A.D. and Brown,B.H. "Limitations in hardware design in impedance imaging", *Clin.Phys.Physiol.Meas*, Vol.8, suppl.A, 1987, pp.85-90

[45] Smith,R.W.M.; Brown,B.H.; Freeston,I.L. and McArdle,F.J. "Real-time electrical impedance tomography", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.212-6

[46] Andersen,O.T.; Dawids,S. and Aa Serensen,J. "A system for electrical impedance tomography with circular finite-element reconstruction", *Abstract of EEC Workshop on Hardware for EIT*, Copenhagen, CAIT, 14-6 June, 1990

[47] Jossinet,J. and Trillaud,C. "A high contrast dual frequency multi-electrode system for electrical impedance tomography", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.144-9

[48] Jossinet,J.; Jarry,R. and Theillere,Y. "A hardware design for imaging the electrical impedance of the breast", *Clin.Phys.Physiol.Meas*, Vol.9, suppl.A, 1988, pp.25-8

[49] Shi,Y.; Rigaud,B.; Anah,J. and Morucci,J.P. "A data collection system for applied potential tomography", *IEEE EMBS*, Vol.11, 1989, pp.453-4

[50] Rigaud,B.; Anah,J.; Givelin,P.; Graziotin,P. and Morucci,J.P. "Multi-function electrode module for electrical impedance tomography", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.217-25

[51] Shi.Y.; Rigaud,B.; Marsili,P.M. and Morucci,J.P. "An electrical impedance tomography, hardware, software and imaging by dual frequency measurements", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.150-7

[52] Kyriacou,G.; Koukourlis,C. and Sahalos,J.N. "Evolution of the reconstruction algorithm and data collection for electrical impedance tomography", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.226-33

[53] Rosell,J.; Riu,P. and Pallas-Areny,R. "A parallel data acquisition system for electrical impedance tomography", *IEEE EMBS*, Vol.11, 1989, pp.459-60

[54] Blad,B.; Bertenstam,L. and Persson,R.B.R. "A multi-frequency, parallel electrical impedance tomography system for non-invasive temperature monitoring and others applications", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.158-61

[55] Woo,E.J.; Hua,P.; Tompkins,W.J. and Webster,J.G. "32-electrode electrical impedance tomography-software design and static images", *IEEE EMBS*, Vol.11, 1989, pp.455-6

[56] Hua,P.; Woo,E.J.; Webster,J.G. and Tompkins,W.J. "An electrical impedance tomograph using compound electrodes", *IEEE EMBS*, Vol.11, 1989, pp.472-3

[57] Gisser,D.G.; Issacson,D. and Newell,J.C. "Theory and performance of an adaptive current tomography system", *Clin.Phys.Physiol.Meas*, Vol.9, 1988, pp.35-41

[58] Scaife,J.M.; Tozer,R.C. and Freeston,I.L. "Real and imaginary impedance images using induced currents", *IEEE EMBS*, Vol.12, 1990, pp.116-7

[59] Toumazou,C.; Lidgey,F.J. and Makris,C.A. "Extending voltage-mode op amps to current-mode performance", *IEE Proceedings*, Vol.137, Pt.G, No.2, Apr., 1990, pp.116-30

[60] Wojslaw,C.F. and Moustakas,E.A. (1986) **Operational amplifiers**, U.S.:John Wiley & Sons, Inc. pp.212-3

[61] Nowicki,D.J. and Webster,J.G. "A one op-amp current source for electrical impedance tomography", *IEEE EMBS*, Vol.11, 1989, pp.457-8

[62] Tobey,G.E.; Graeme,J.G. and Huelsman,L.P. (1971) **Operational amplifiers: design and applications**, New York:McGraw-Hill, pp.225-7

[63] Toumazou,C.; Lidgey,F.J. and Haigh,D.G. (1990) **Analogue IC design: the current-mode approach**, England:Peter Peregrinus Ltd., pp.93-115

[64] Leung,H.T.L.; Williams,R.J. and Griffiths,H. "A wide-band current source for electrical impedance tomography", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.206-11

[65] Griffiths,H.; Zhang,Z. and Watts,M. "A constant-perturbation saline phantom for electrical impedance tomography", *Phys.Med.Biol.*, Vol.34, No.8, 1989, pp.1063-71

[66] Charnley,M. "A model design approach for electronic CAE", *Computer-Aided Engineering Journal*, Dec., 1985, pp.192-9

[67] Griffiths,H; Leung,H.T.L. and Williams,R.J. "Imaging permittivity in electrical impedance tomography", *IEEE EMBS*, Vol.13, 1991, In press

[68] Brown,B.H.; Barber,D.C. and Seagar,A.D. "Applied potential tomography: possible clinical applications", *Clin.Phys.Physiol.Meas.*, Vol.6, No.2, 1985, pp.109-21

[69] Wilson,B. "Current mirrors, amplifiers and dumpers", *Wireless World*, Vol.87, No.1551, Dec., 1981, pp.47-50

[70] Hart,B.L. and Barker,R.W.J. "D.C. matching errors in the Wilson current source", *Electronics Letters*, Vol.12, No.15, Jul.22, 1976, pp.389-90

[71] Roberge,J.K. (1975) **Operational amplifiers - theory and practice**, Canada: John Wiley & Sons Inc., pp.327-35

[72] Cluley,J.C. (1983) **Interfacing to Microprocessors**, London: The Macmillan Press Ltd, pp.125-7

[73] Zhang,Z. and Griffiths,H. "Dual-frequency electrical impedance tomography of the forearm", *Proc. of a meeting on Electrical Impedance Tomography*, Copenhagen, CAIT, Jun.14-16, 1990 pp.256-65

[74] Zheng,E.; Shao,S. and Webster,J.G. "Impedance of skeletal muscle from 1Hz to 1MHz", *IEEE Transactions on Biomedical Engineering*, Vol.BME-31, No.6, Jun., 1984, pp.477-81

[75] Griffiths,H.; Leung,H.T.L. and Williams,R.J. "Imaging complex impedance of the thorax by electrical impedance tomography", *Clin.Phys.Physiol.Meas.*, In press

Appendix B Operating Codes for the Polytechnic of Wales System

'01' : to alter stimulus settings
'02' : to start voltage measurements
'03' : to start data transmission
'04' : to initialise data storage
'05' : to transmit verification codes
'06' : to initiate dual-frequency measurements
'07' : to initiate cardiac gated imaging
'ff' : a special code for completion of operation

Appendix C Memory Map for the Data Collection System

0000_{16}	internal registers
$001F_{16}$	
0020_{16}	sc : start conversion
0021_{16}	ub : load upper byte from the ADC
0022_{16}	lb : load lower byte from the ADC
0023_{16}	latch : latch the attenuation factor
0024_{16}	m0 : select drive pair
0025_{16}	m1 : select measurement pair
0026_{16}	m3 : switch to the real part signal
0027_{16}	m2 : switch to the imaginary part signal
0028_{16}	m4 : set frequency and amplitude of stimuli
0029_{16}	not used
0080_{16}	
	internal rams
$00FF_{16}$	
	not used
$0FFF_{16}$	
	data storage
$5FFF_{16}$	
	not used
$EFFF_{16}$	
	operation program sequences
$FEFF_{16}$	
	interrupt/reset vectors
$FFFF_{16}$	

Appendix D Program Listings of Modula-2

Page

D.1 EITCGI (Cardiac gated imaging) 161

D.2 EITCom (Serial communication) 164

D.3 EITDFreq (Dual-frequency imaging) 167

D.4 EITDisp (Image display) 170

D.5 EITGraph (Graphic definitions) 173

D.6 EITNoise (System performance characterisations) 176

D.7 EITSetUp (Applied stimulus set-up) 179

D.8 EITUtil (Ordinary utilities) 181

D.9 EITSW5 (EIT interface main program) 182

```
IMPLEMENTATION MODULE EITCG1;

FROM Window IMPORT
    WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,
    SetTitle,Close,Clear,CursorOn,CursorOff,TextColor,Use,Hide,Used,
    CenterUpperTitle,PutOnTop,GotoXY,FullScreenDef,ClrEol,SetFrame;
FROM IO IMPORT
    WrInt,WrStr,WrLn,RdChar,RdInt,RdKey,RdItem,KeyPressed,WrShtHex;
FROM Lib IMPORT Delay;
FROM EITCom IMPORT
    Install,Init,RxCount,Send,Receive,Even,None;
FROM EITUtil IMPORT
    Alarm,ChColor,ClrKeyBuffer,ClrRxCount;

IMPORT FIO;

CONST  cr = 15C; esc = 33C;
TYPE   PtrType = (amp,freq);

VAR    pause            : CHAR;
    Yposition,numset    : CARDINAL;
    First            : BOOLEAN;
    numfm            : INTEGER;
    parameter        : PtrType;
    setupcode,ccode,fcode : BYTE;
    work_window    : WinType;
    setup_window   : WinType;
    cstr,fstr: ARRAY [1..78] OF CHAR;

PROCEDURE InitData;
VAR cdata,fdata : CHAR; initfile : FIO.File;
BEGIN
  initfile := FIO.Open('initial.dat');
  cdata := FIO.RdChar(initfile);
  fdata := FIO.RdChar(initfile);
  CASE fdata OF
  |'1': fcode := 08H; fstr := '10' |'2': fcode := 10H; fstr := '20';
  |'3': fcode := 00H; fstr := '40' |'4': fcode := 18H; fstr := '80';
  END;
  CASE cdata OF
  |'1': ccode := 00H; cstr := '0.00' |'2': ccode := 01H; cstr := '0.00';
  |'3': ccode := 02H; cstr := '0.00' |'4': ccode := 03H; cstr := '0.00';
  |'5': ccode := 04H; cstr := '1.00' |'6': ccode := 05H; cstr := '1.50';
  |'7': ccode := 06H; cstr := '2.00' |'8': ccode := 07H; cstr := '2.50';
  END;
  FIO.Close(initfile);
END InitData;

PROCEDURE SetAmp;
VAR select : CHAR; amp_window : WinType;
BEGIN
  amp_window := Open(WinDef(47,10,77,24,LightGray,Black,TRUE,FALSE,
            FALSE,TRUE,SingleFrame,LightGray,Black));
  SetTitle(amp_window," Amplitude Set Up ",CenterUpperTitle);
  Use(amp_window); Clear; WrLn;
  WrStr('  Selection of amplitudes'); WrLn; WrLn;
  WrStr('  ['); ChColor('1',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
  WrStr('  ['); ChColor('2',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
  WrStr('  ['); ChColor('3',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
  WrStr('  ['); ChColor('4',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
  WrStr('  ['); ChColor('5',Brown,LightGray); WrStr('] 1.00 mA rms'); WrLn;
  WrStr('  ['); ChColor('6',Brown,LightGray); WrStr('] 1.50 mA rms'); WrLn;
  WrStr('  ['); ChColor('7',Brown,LightGray); WrStr('] 2.00 mA rms'); WrLn;
  WrStr('  ['); ChColor('8',Brown,LightGray); WrStr('] 2.50 mA rms');
  LOOP
   GotoXY(3,13); WrStr('Please enter... ');
   select := RdKey();
   CASE select OF
   |'1': ccode := 00H; cstr := '0.00'; EXIT
   |'2': ccode := 01H; cstr := '0.00'; EXIT
   |'3': ccode := 02H; cstr := '0.00'; EXIT
   |'4': ccode := 03H; cstr := '0.00'; EXIT
   |'5': ccode := 04H; cstr := '1.00'; EXIT
   |'6': ccode := 05H; cstr := '1.50'; EXIT
   |'7': ccode := 06H; cstr := '2.00'; EXIT
   |'8': ccode := 07H; cstr := '2.50'; EXIT
   |esc: EXIT
   ELSE Alarm(3500,100,2);
   END;
  END; (* LOOP *)
  setupcode := ccode+fcode;
  Close(amp_window);
END SetAmp;

PROCEDURE SetFreq;
VAR select : CHAR; freq_window : WinType;
BEGIN
  freq_window := Open(WinDef(47,14,77,24,LightGray,Black,TRUE,FALSE,
            FALSE,TRUE,SingleFrame,LightGray,Black));
  SetTitle(freq_window," Frequency Set Up ",CenterUpperTitle);
  Use(freq_window); Clear; WrLn;
  WrStr('  Selection of frequencies'); WrLn; WrLn;
  WrStr('  ['); ChColor('1',Brown,LightGray); WrStr('] 10 KHz'); WrLn;
  WrStr('  ['); ChColor('2',Brown,LightGray); WrStr('] 20 KHz'); WrLn;
  WrStr('  ['); ChColor('3',Brown,LightGray); WrStr('] 40 KHz'); WrLn;
  WrStr('  ['); ChColor('4',Brown,LightGray); WrStr('] 80 KHz');
  LOOP
   GotoXY(3,9); WrStr('Please enter... ');
   select := RdKey();
   CASE select OF
   |'1': fcode := 08H; fstr := '10'; EXIT
   |'2': fcode := 10H; fstr := '20'; EXIT
   |'3': fcode := 00H; fstr := '40'; EXIT
   |'4': fcode := 18H; fstr := '80'; EXIT
   |esc: EXIT
   ELSE Alarm(3500,100,2);
```

```
    END;
  END;
  setupcode := ccode+fcode;
  Close(freq_window);
END SetFreq;

PROCEDURE SendSignal(sendcode : BYTE);
BEGIN
  ClrRxCount; Send(sendcode,1)
END SendSignal;

PROCEDURE Increment (VAR input : PtrType);
BEGIN
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,Yellow,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,Yellow,LightGray)
  END;
  CASE input OF
  |amp : input := freq; Yposition := 3;
  |freq: input := amp;  Yposition := 2;
  END;
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,LightRed,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,LightRed,LightGray)
  END;
END Increment;

PROCEDURE Decrement (VAR input : PtrType);
BEGIN
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,Yellow,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,Yellow,LightGray)
  END;
  CASE input OF
  |amp : input := freq; Yposition := 3;
  |freq: input := amp;  Yposition := 2;
  END;
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,LightRed,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,LightRed,LightGray)
  END;
END Decrement;

PROCEDURE CGISettings (VAR numfm : INTEGER);
LABEL Alter;
VAR ans,select : CHAR;
  exit_window : WinType;
BEGIN
  InitData;
  exit_window := Open(WinDef(47,22,63,24,Cyan,Black,TRUE,FALSE,
                 FALSE,TRUE,SingleFrame,LightGray,Black));
  WrStr(' <'); ChColor('esc',LightRed,Cyan); WrStr('> to exit');
  work_window := Open(WinDef(0,12,79,15,LightGray,Blue,TRUE,FALSE,
                 FALSE,TRUE,SingleFrame,LightGray,Blue));
  SetTitle(work_window," Cardiac Gated Imaging ",CenterUpperTitle);
  setup_window := Open(WinDef(0,19,36,24,LightGray,Black,FALSE,FALSE,
                  FALSE,TRUE,SingleFrame,LightGray,Black));
  SetTitle(setup_window," Cardiac Gated Imaging Set Up ",CenterUpperTitle);
  PutOnTop(setup_window); CursorOn;
  GotoXY(8,2); WrStr('Amplitude = ');
  ChColor(cstr,Yellow,LightGray); WrStr(' mA');
  GotoXY(8,3); WrStr('Frequency =  ');
  ChColor(fstr,Yellow,LightGray); WrStr(' kHz');
  parameter := freq;
Alter:
  PutOnTop(setup_window); CursorOn;
  LOOP
    CASE parameter OF
    |amp : GotoXY(20,2); TextColor(LightRed); WrStr(cstr);
        SetAmp; GotoXY(20,2); WrStr(cstr); Yposition := 2;
    |freq: GotoXY(22,3); TextColor(LightRed); WrStr(fstr);
        SetFreq; GotoXY(22,3); WrStr(fstr); Yposition := 3;
    END; (* of CASE *)
  PutOnTop(setup_window); CursorOn;
  select := RdKey();
  REPEAT
   IF select = CHR(0) THEN
     select := RdKey();
     CASE select OF
     |CHR(80) : Increment(parameter); select := RdKey();
     |CHR(72) : Decrement(parameter); select := RdKey();
     END; (* of CASE *)
   ELSIF select = esc THEN EXIT;
   ELSIF select = cr THEN GOTO Alter;
   ELSE select := RdKey();
   END;
  UNTIL FALSE;
  END; (* of LOOP *)

  PutOnTop(work_window);
  Clear;
  WrStr('  Change measurements settings? [Y/N]... '); ans := RdKey();
  IF ans = 'y' THEN ans := 'Y' END;
  IF ans = 'Y' THEN GOTO Alter END;

  Clear; GotoXY(3,1); WrStr('Number of data sets [max.=5] : ');
  numset := RdInt();
  LOOP
  IF (numset>=1) AND (numset<=5) THEN numfm := 2*numset; EXIT ELSE
    Alarm(3500,100,2); GotoXY(34,1); ClrEol; numset := RdInt()
  END;
  END; Close(exit_window);
```

```
END CGISettings;

PROCEDURE Acknowledgment (n : INTEGER; VAR error : BOOLEAN);
VAR rdbuffer : BYTE;
BEGIN
   LOOP
    IF KeyPressed() THEN ClrKeyBuffer; error := TRUE; EXIT END;
    IF RxCount() # 0 THEN
      Receive(rdbuffer,1); error := FALSE;
      CASE n OF
      0 : IF rdbuffer # BYTE(7) THEN error := TRUE; END;
      1 : IF rdbuffer # BYTE(numset) THEN error := TRUE; END;
      2 : IF rdbuffer # setupcode THEN error := TRUE; END;
      END; EXIT
    END;
   END;
END Acknowledgment;

PROCEDURE CardiacGatedImaging;
LABEL Error,EndMeasurement;
VAR  rdbuffer : BYTE; error : BOOLEAN;
BEGIN
   PutOnTop(work_window); CursorOn; error := FALSE;
   (* initiate Cardiac Gated Imaging routine in EIT system *)
   SendSignal(07H); Acknowledgment(0,error);
   IF error = TRUE THEN GOTO Error END;
   SendSignal(BYTE(numset)); Acknowledgment(1,error);
   IF error = TRUE THEN GOTO Error END;
   SendSignal(setupcode); Acknowledgment(2,error);
Error:
   IF error = TRUE THEN
     Clear; WrStr('  ERROR in RS232 link'); GOTO EndMeasurement;
   END;

(* wait for completion *)
   GotoXY(3,2);
   WrStr('Cardiac gated imaging start, please wait... ');
   LOOP
    IF KeyPressed() THEN ClrKeyBuffer; error := TRUE; EXIT END;
    IF RxCount() # 0 THEN
      Receive(rdbuffer,1);
      IF rdbuffer = BYTE(255) THEN EXIT ELSE error := TRUE; EXIT END
    END;
   END;

EndMeasurement:
   IF error = FALSE THEN
     GotoXY(3,2); ClrEol; TextColor(Yellow);
     WrStr('Cardiac gated imaging has been completed.');
     Alarm(4000,100,1); Delay(1000);
   ELSE
     GotoXY(3,2); ClrEol; TextColor(LightRed);
     WrStr('Process aborted!'); Alarm(3500,100,2); pause := RdKey();
   END; Close(setup_window); Close(work_window); CursorOn;
END CardiacGatedImaging;

(*
BEGIN
   First := TRUE;
   Install(1,First);
   Init(9600,8,None,TRUE,FALSE);
   CGISettings(numfm);
   CardiacGatedImaging;
   CursorOn;
*)
END EITCGI.
```

```
IMPLEMENTATION MODULE EITCom;
IMPORT SYSTEM, Lib, IO;

CONST Com1     = 3F8H;
    Com1INr  = 4;
    Com2     = 2F8H;
    Com2INr  = 3;
VAR IOBase,ComINr : CARDINAL;

CONST
    TXBuffer   ::= IOBase;
    RXBuffer   ::= IOBase;
    DivisorLsb ::= IOBase;

    BufferSize = 256;
    TRMax      = BufferSize-16;
    Noise      = FALSE;

VAR WorkSpace          : ARRAY[0..7FFH] OF CHAR;
  MainProc             : ADDRESS;
  IP                   : ADDRESS;
  OldC                 : PROC;
  Term                 : PROC;
(*$W+*)
  TXReady,CTS,CTSH     : BOOLEAN;
  DTR                  : BOOLEAN;
  TXCount,TXI,TXO      : CARDINAL;
  RXCount,RXI,RXO      : CARDINAL;
(*$W-*)
  IBuf,OBuf            : ARRAY[0..BufferSize-1] OF SHORTCARD;

PROCEDURE TX;
BEGIN
 IF CTS OR NOT CTSH THEN
   TXReady := FALSE;
   SYSTEM.Out( TXBuffer,OBuf[TXO] );
   TXO := (TXO+1) MOD BufferSize;
   DEC(TXCount);
 END;
END TX;

PROCEDURE RX;
TYPE b = SET OF SHORTCARD[0..7];
BEGIN
 IF RXCount >= TRMax THEN
   SYSTEM.Out( IOBase+4 (*ModemCont*), SHORTCARD( b(SYSTEM.In(IOBase+4 (*ModemCont*)))-b{0,1} ) );
   (* DTR,RTS := FALSE *)
   DTR := FALSE;
 END;
 IBuf[RXI] := SYSTEM.In( RXBuffer );
 IF Noise AND (Lib.RANDOM(500)=0) THEN
   IBuf[RXI] := SHORTCARD(Lib.RANDOM(256));
 END;
 RXI := (RXI+1) MOD BufferSize;
 INC(RXCount);
END RX;

VAR GotBreak : BOOLEAN;

(*$C FF,J+*)
PROCEDURE Int;
TYPE b = SET OF SHORTCARD[0..7];
VAR i  : SHORTCARD;
   s : b;
BEGIN
(*  SYSTEM.EI; *)
 LOOP
   i := SYSTEM.In( IOBase+2 (*IntId*) );
   IF i=1 THEN EXIT; END;
   CASE i DIV 2 OF
   | 0 : (* Modem status *)

       s := b(SYSTEM.In( IOBase+6 (*ModemStatus*) ));

       IF 0 IN s THEN (* CTS changed *)
         CTS :=4 IN s;
         IF CTS AND (TXCount >0) AND TXReady THEN TX END;
       END;


   | 1 : (* TXEmpty     *)
       TXReady := TRUE;
       IF TXCount > 0 THEN TX; END;

   | 2 : (* RXReady     *)

       RX;

   | 3 : (* LineStatus   *)

       s := b(SYSTEM.In( IOBase+5 (*LineStatus*) ));
       (*
       IF 1 IN s THEN IO.WrStr('OR-Error'); END;
       IF 2 IN s THEN IO.WrStr('PE-Error'); END;
       IF 4 IN s THEN IO.WrStr('Break');
       ELSIF 3 IN s THEN IO.WrStr('FE-Error'); END;

       SYSTEM.Out(20H,20H);
       HALT;*)
   ELSE (*IO.WrStr('Unknown Int'); HALT; *)
   END;
 END;
 SYSTEM.Out(20H,20H);
END Int;

(*$C F0,J-*)
```

```
PROCEDURE RxCount():CARDINAL;
BEGIN
 RETURN RXCount;
END RxCount;

PROCEDURE TxCount():CARDINAL;
BEGIN
 RETURN TXCount;
END TxCount;

VAR lc : SHORTCARD;

PROCEDURE Init( Baud      : CARDINAL;
          WordLength : wl;
          Parity     : pt;
          OneStopBit : BOOLEAN;
          HandShake  : BOOLEAN);

VAR d  : CARDINAL;
BEGIN
 CTSH := HandShake;
 lc := (WordLength-5) MOD 4;
 IF NOT OneStopBit THEN lc := lc+4; END;
 CASE Parity OF
  None  :;
  Even  : lc := lc+18H;
  Odd   : lc := lc+ 8H;
  Mark  : lc := lc+38H;
  Space : lc := lc+28H;
 END;
 SYSTEM.Out( IOBase+3 (*LineCont*),80H );
 d := CARDINAL( 115200 DIV LONGCARD( Baud ) );
 SYSTEM.Out( DivisorLsb, SHORTCARD(d));
 SYSTEM.Out( IOBase+1 (*DivisorMsb*), SHORTCARD(d DIV 100H));
 SYSTEM.Out( IOBase+3 (*LineCont*),lc );
END Init;

(*$V-*)

PROCEDURE Receive(VAR Buf : ARRAY OF BYTE; Len : CARDINAL );
TYPE b = SET OF SHORTCARD{0..7};
VAR i : CARDINAL;
BEGIN
 FOR i := 0 TO Len-1 DO
    WHILE RXCount = 0 DO END;
    Buf[i] := IBuf[RXO];
    DEC( RXCount );
    RXO := (RXO+1) MOD BufferSize;
 END;
 IF NOT DTR AND (RXCount < TRMax-16) THEN
   SYSTEM.Out( IOBase+4 (*ModemCont*), SHORTCARD( b(SYSTEM.In(IOBase+4 (*ModemCont*)))+b{0,1} ) );
   DTR := TRUE;
 END;
END Receive;

PROCEDURE Send( Buf : ARRAY OF BYTE; Len : CARDINAL );
VAR i : CARDINAL;
BEGIN
 FOR i := 0 TO Len-1 DO
    OBuf[TXI] := Buf[i];
    INC(TXCount);
    TXI := (TXI+1) MOD BufferSize;
    IF TXReady THEN TX; END;
 END;
END Send;

TYPE dp=POINTER TO PROC;

PROCEDURE CloseDown;
BEGIN
(*  IO.WrShtHex( SYSTEM.In( IOBase+5 (*LineStatus*) ),4 ); *)
 [0:4*(8+ComINr) dp]^ := OldC;
 SYSTEM.Out( IOBase+1 (*IntEnable*),00H );
 Term;
END CloseDown;

PROCEDURE Install2( Port,Intr : CARDINAL; VAR first : BOOLEAN);
TYPE bs = SET OF SHORTCARD{0..7};
VAR  s  : SHORTCARD; i : CARDINAL;
BEGIN
 IOBase := Port;
 ComINr := Intr;
 SYSTEM.Out( IOBase+1 (*IntEnable*),00H );
 SYSTEM.Out( IOBase+5 (*LineStatus*),0 );
 TXI    := 0;
 TXO    := 0;
 RXI    := 0;
 RXO    := 0;
 RXCount := 0;
 TXCount := 0;
 OldC   := [0:4*(8+ComINr) dp]^;
 [0:4*(8+ComINr) dp]^   := PROC(ADR(Int));
 IF first = TRUE THEN
   Lib.Terminate( CloseDown,Term ); first := FALSE
 END;
 SYSTEM.NewPriority( CARDINAL(BITSET( SYSTEM.CurrentPriority() )-{ComINr} ) );
 FOR i := 1 TO 10 DO
    s := SYSTEM.In( RXBuffer );
    s := SYSTEM.In( IOBase+5 (*LineStatus*) );
    Lib.Delay(10);
 END;
 TXReady := TRUE;
 CTS    := 4 IN bs(SYSTEM.In( IOBase+6 (*ModemStatus*) ));
 DTR    := TRUE;
 SYSTEM.Out( IOBase+1 (*IntEnable*),03H );
```

```
 SYSTEM.Out( IOBase+4 (*ModemCont*),0BH );
END Install2;

PROCEDURE Install( Port : CARDINAL; VAR first : BOOLEAN );
BEGIN
 IF Port= 1 THEN
   Install2( Com1,Com1INr,first );
 ELSE
   Install2( Com2,Com2INr,first );
 END;
END Install;

BEGIN
 lc := SYSTEM.In( IOBase+3 (*LineCont*) );
END EITCom.
```

```
IMPLEMENTATION MODULE EITDFreq;

FROM Window IMPORT
     WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,
     SetTitle,Close,Clear,CursorOn,CursorOff,TextColor,Use,Hide,Used,
     CenterUpperTitle,PutOnTop,GotoXY,FullScreenDef,ClrEol,SetFrame;
FROM IO IMPORT WrInt,WrStr,WrLn,RdChar,RdInt,RdKey,RdItem,KeyPressed,WrShtHex;
FROM EITCom IMPORT Install,Init,RxCount,Send,Receive,Even,None;
FROM Lib IMPORT Delay;
FROM EITUtil IMPORT Alarm,ChColor,ClrKeyBuffer,ClrRxCount;
IMPORT FIO;

CONST  cr = 15C; esc = 33C;
TYPE   PtrType = (amp1,freq1,amp2,freq2);

VAR    pause              : CHAR;
     First              : BOOLEAN;
     numfm              : INTEGER;
     Yposition,numset   : CARDINAL;
     parameter          : PtrType;
     setupcode,ccode,fcode : BYTE;
     setupcode1,setupcode2 : BYTE;
     ccode1,ccode2,fcode1,fcode2 : BYTE;
     setup_window,work_window    : WinType;
     cstr,cstr1,cstr2: ARRAY [1..4] OF CHAR;
     fstr,fstr1,fstr2: ARRAY [1..4] OF CHAR;

PROCEDURE InitData;
VAR cdata,fdata : CHAR; initfile : FIO.File;
BEGIN
   initfile := FIO.Open('initial.dat');
   cdata := FIO.RdChar(initfile); fdata := FIO.RdChar(initfile);
   CASE fdata OF
   |'1': fcode1 := 08H; fstr1 := '10'; fcode2 := 08H; fstr2 := '10';
   |'2': fcode1 := 10H; fstr1 := '20'; fcode2 := 10H; fstr2 := '20';
   |'3': fcode1 := 00H; fstr1 := '40'; fcode2 := 00H; fstr2 := '40';
   |'4': fcode1 := 18H; fstr1 := '80'; fcode2 := 18H; fstr2 := '80';
   END;
   CASE cdata OF
   |'1': ccode1 := 00H; cstr1 := '0.00'; ccode2 := 00H; cstr2 := '0.00';
   |'2': ccode1 := 01H; cstr1 := '0.00'; ccode2 := 01H; cstr2 := '0.00';
   |'3': ccode1 := 02H; cstr1 := '0.00'; ccode2 := 02H; cstr2 := '0.00';
   |'4': ccode1 := 03H; cstr1 := '0.00'; ccode2 := 03H; cstr2 := '0.00';
   |'5': ccode1 := 04H; cstr1 := '1.00'; ccode2 := 04H; cstr2 := '1.00';
   |'6': ccode1 := 05H; cstr1 := '1.50'; ccode2 := 05H; cstr2 := '1.50';
   |'7': ccode1 := 06H; cstr1 := '2.00'; ccode2 := 06H; cstr2 := '2.00';
   |'8': ccode1 := 07H; cstr1 := '2.50'; ccode2 := 07H; cstr2 := '2.50';
   END;
   FIO.Close(initfile);
   ccode := ccode1; fcode := fcode1;
   setupcode := ccode+fcode;
   setupcode1 := setupcode; setupcode2 := setupcode;
END InitData;

PROCEDURE SetAmp;
VAR select : CHAR; amp_window : WinType;
BEGIN
   amp_window := Open(WinDef(47,10,77,24,LightGray,Black,TRUE,FALSE,
                 FALSE,TRUE,SingleFrame,LightGray,Black));
   SetTitle(amp_window," Amplitude Set Up ",CenterUpperTitle);
   Use(amp_window); Clear; WrLn;
   WrStr(' Selection of amplitudes'); WrLn; WrLn;
   WrStr('  ['); ChColor('1',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('2',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('3',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('4',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('5',Brown,LightGray); WrStr('] 1.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('6',Brown,LightGray); WrStr('] 1.50 mA rms'); WrLn;
   WrStr('  ['); ChColor('7',Brown,LightGray); WrStr('] 2.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('8',Brown,LightGray); WrStr('] 2.50 mA rms');
   LOOP
    GotoXY(3,13); WrStr('Please enter... ');
    select := RdKey();
    CASE select OF
    |'1': ccode := 00H; cstr := '0.00'; EXIT
    |'2': ccode := 01H; cstr := '0.00'; EXIT
    |'3': ccode := 02H; cstr := '0.00'; EXIT
    |'4': ccode := 03H; cstr := '0.00'; EXIT
    |'5': ccode := 04H; cstr := '1.00'; EXIT
    |'6': ccode := 05H; cstr := '1.50'; EXIT
    |'7': ccode := 06H; cstr := '2.00'; EXIT
    |'8': ccode := 07H; cstr := '2.50'; EXIT
    |esc: EXIT
    ELSE Alarm(3500,100,2);
    END;
   END; (* LOOP *)
   setupcode := ccode+fcode;
   Close(amp_window);
END SetAmp;

PROCEDURE SetFreq;
VAR select : CHAR; freq_window : WinType;
BEGIN
   freq_window := Open(WinDef(47,14,77,24,LightGray,Black,TRUE,FALSE,
                  FALSE,TRUE,SingleFrame,LightGray,Black));
   SetTitle(freq_window," Frequency Set Up ",CenterUpperTitle);
   Use(freq_window); Clear; WrLn;
   WrStr(' Selection of frequencies'); WrLn; WrLn;
   WrStr('  ['); ChColor('1',Brown,LightGray); WrStr('] 10 KHz'); WrLn;
   WrStr('  ['); ChColor('2',Brown,LightGray); WrStr('] 20 KHz'); WrLn;
   WrStr('  ['); ChColor('3',Brown,LightGray); WrStr('] 40 KHz'); WrLn;
   WrStr('  ['); ChColor('4',Brown,LightGray); WrStr('] 80 KHz');
   LOOP
    GotoXY(3,9); WrStr('Please enter... ');
    select := RdKey();
    CASE select OF
```

```
      '1': fcode := 08H; fstr := '10'; EXIT
      '2': fcode := 10H; fstr := '20'; EXIT
      '3': fcode := 00H; fstr := '40'; EXIT
      '4': fcode := 18H; fstr := '80'; EXIT
      esc: EXIT
      ELSE Alarm(3500,100,2);
      END;
    END;
    setupcode := ccode+fcode;
    Close(freq_window);
END SetFreq;

PROCEDURE SendSignal(sendcode : BYTE);
BEGIN
    ClrRxCount; Send(sendcode,1)
END SendSignal;

PROCEDURE Increment (VAR input : PtrType);
BEGIN
    IF (Yposition = 2) OR (Yposition = 4) THEN
      GotoXY(26,Yposition); ChColor(cstr,Yellow,LightGray)
    ELSE
      GotoXY(28,Yposition); ChColor(fstr,Yellow,LightGray)
    END;
    CASE input OF
    amp1 : input := freq1; Yposition := 3; fstr := fstr1;
    freq1: input := amp2;  Yposition := 4; cstr := cstr2;
    amp2 : input := freq2; Yposition := 5; fstr := fstr2;
    freq2: input := amp1;  Yposition := 2; cstr := cstr1;
    END;
    IF (Yposition = 2) OR (Yposition = 4) THEN
      GotoXY(26,Yposition); ChColor(cstr,LightRed,LightGray)
    ELSE
      GotoXY(28,Yposition); ChColor(fstr,LightRed,LightGray)
    END;
END Increment;

PROCEDURE Decrement (VAR input : PtrType);
BEGIN
    IF (Yposition = 2) OR (Yposition = 4) THEN
      GotoXY(26,Yposition); ChColor(cstr,Yellow,LightGray)
    ELSE
      GotoXY(28,Yposition); ChColor(fstr,Yellow,LightGray)
    END;
    CASE input OF
    amp1 : input := freq2; Yposition := 5; fstr := fstr2;
    freq1: input := amp1;  Yposition := 2; cstr := cstr1;
    amp2 : input := freq1; Yposition := 3; fstr := fstr1;
    freq2: input := amp2;  Yposition := 4; cstr := cstr2;
    END;
    IF (Yposition = 2) OR (Yposition = 4) THEN
      GotoXY(26,Yposition); ChColor(cstr,LightRed,LightGray)
    ELSE
      GotoXY(28,Yposition); ChColor(fstr,LightRed,LightGray)
    END;
END Decrement;

PROCEDURE DualFreqSettings(VAR numfm : INTEGER);
LABEL Alter;
VAR ans,select : CHAR;
    exit_window : WinType;
BEGIN
    InitData;
    exit_window := Open(WinDef(47,22,63,24,Cyan,Black,TRUE,FALSE,
                   FALSE,TRUE,SingleFrame,LightGray,Black));
    WrStr(' <'); ChColor('esc',LightRed,Cyan); WrStr('> to exit');
    work_window := Open(WinDef(0,12,79,15,LightGray,Blue,TRUE,FALSE,
                   FALSE,TRUE,SingleFrame,LightGray,Blue));
    SetTitle(work_window," Dual Frequency Measurements ",CenterUpperTitle);
    setup_window := Open(WinDef(0,17,36,24,LightGray,Black,FALSE,FALSE,
                   FALSE,TRUE,SingleFrame,LightGray,Black));
    SetTitle(setup_window," Dual Frequency Set Up ",CenterUpperTitle);
    PutOnTop(setup_window); CursorOn;
    GotoXY(4,2); WrStr('First set amplitude = ');
    ChColor(cstr1,Yellow,LightGray); WrStr(' mA');
    GotoXY(14,3); WrStr('frequency =  ');
    ChColor(fstr1,Yellow,LightGray); WrStr(' kHz');
    GotoXY(3,4); WrStr('Second set amplitude = ');
    ChColor(cstr2,Yellow,LightGray); WrStr(' mA');
    GotoXY(14,5); WrStr('frequency =  ');
    ChColor(fstr2,Yellow,LightGray); WrStr(' kHz');
    parameter := amp1;
Alter:
    PutOnTop(setup_window); CursorOn;
    LOOP
      CASE parameter OF
      amp1 : GotoXY(26,2); TextColor(LightRed); WrStr(cstr1);
             fstr := fstr1; cstr := cstr1;
             fcode := fcode1; ccode := ccode1;
             SetAmp;
             fstr1 := fstr; cstr1 := cstr;
             fcode1 := fcode; ccode1 := ccode;
             setupcode1 := setupcode;
             GotoXY(26,2); WrStr(cstr); Yposition := 2;
      freq1: GotoXY(28,3); TextColor(LightRed); WrStr(fstr1);
             fstr := fstr1; cstr := cstr1;
             fcode := fcode1; ccode := ccode1;
             SetFreq;
             fstr1 := fstr; cstr1 := cstr;
             fcode1 := fcode; ccode1 := ccode;
             setupcode1 := setupcode;
             GotoXY(28,3); WrStr(fstr); Yposition := 3;
      amp2 : GotoXY(26,4); TextColor(LightRed); WrStr(cstr2);
             fstr := fstr2; cstr := cstr2;
             fcode := fcode2; ccode := ccode2;
             SetAmp;
```

```
            fstr2 := fstr; cstr2 := cstr;
            fcode2 := fcode; ccode2 := ccode;
            setupcode2 := setupcode;
            GotoXY(26,4); WrStr(cstr); Yposition := 4;
        |freq2: GotoXY(28,5); TextColor(LightRed); WrStr(fstr2);
            fcode := fcode2; ccode := ccode2;
            fstr := fstr2; cstr := cstr2;
            SetFreq;
            fstr2 := fstr; cstr2 := cstr;
            fcode2 := fcode; ccode2 := ccode;
            setupcode2 := setupcode;
            GotoXY(28,5); WrStr(fstr); Yposition := 5;
        END; (* of CASE *)
      PutOnTop(setup_window); CursorOn;
        select := RdKey();
        REPEAT
        IF select = CHR(0) THEN
          select := RdKey();
          CASE select OF
          |CHR(80) : Increment(parameter); select := RdKey();
          |CHR(72) : Decrement(parameter); select := RdKey();
          END; (* of CASE *)
        ELSIF select = esc THEN EXIT;
        ELSIF select = cr THEN GOTO Alter;
        ELSE select := RdKey();
        END;
        UNTIL FALSE;
      END; (* of LOOP *)

      PutOnTop(work_window); Clear;
      WrStr('  Change measurements settings? [N]... '); ans := RdKey();
      IF ans = 'y' THEN ans := 'Y' END;
      IF ans = 'Y' THEN GOTO Alter END;

      Clear; GotoXY(3,1); WrStr('Number of data sets [max.=5] : ');
      numset := RdInt();
      LOOP
      IF (numset>=1) AND (numset<=5) THEN numfm := 2*numset; EXIT ELSE
        Alarm(3500,100,2); GotoXY(34,1); ClrEol; numset := RdInt()
      END;
      END; Close(exit_window);
END DualFreqSettings;

PROCEDURE Acknowledgment (n : INTEGER; VAR error : BOOLEAN);
VAR rdbuffer : BYTE;
BEGIN
    LOOP
      IF KeyPressed() THEN ClrKeyBuffer; error := TRUE; EXIT END;
      IF RxCount() # 0 THEN
        Receive(rdbuffer,1);
        CASE n OF
        |0 : IF rdbuffer = BYTE(6) THEN
               error := FALSE; EXIT ELSE error := TRUE; EXIT END;
        |1 : IF rdbuffer = BYTE(numset) THEN
               error := FALSE; EXIT ELSE error := TRUE; EXIT END;
        |2 : IF rdbuffer = setupcode1 THEN
               error := FALSE; EXIT ELSE error := TRUE; EXIT END;
        |3 : IF rdbuffer = setupcode2 THEN
               error := FALSE; EXIT ELSE error := TRUE; EXIT END;
        END;
      END;
    END;
END Acknowledgment;

PROCEDURE DualFreqMeasurements;
LABEL Error,EndMeasurement;
VAR  rdbuffer : BYTE; error : BOOLEAN;
BEGIN
    PutOnTop(work_window); CursorOn; error := FALSE;
    (* initiate Dual Freq Routine in EIT system *)
    SendSignal(06H); Acknowledgment(0,error);
    IF error = TRUE THEN GOTO Error END;
    SendSignal(BYTE(numset)); Acknowledgment(1,error);
    IF error = TRUE THEN GOTO Error END;
    SendSignal(setupcode1); Acknowledgment(2,error);
    IF error = TRUE THEN GOTO Error END;
    SendSignal(setupcode2); Acknowledgment(3,error);
Error:
    IF error = TRUE THEN
      Clear; WrStr('  ERROR in RS232 link'); GOTO EndMeasurement;
    END;

(* wait for completion *)
    GotoXY(3,2); WrStr('Dual frequency measurement(s) start, please wait... ');
    LOOP
      IF KeyPressed() THEN ClrKeyBuffer; error := TRUE; EXIT END;
      IF RxCount() # 0 THEN
        Receive(rdbuffer,1);
        IF rdbuffer = BYTE(255) THEN EXIT ELSE error := TRUE; EXIT END
      END;
    END;

EndMeasurement:
    IF error = FALSE THEN
      GotoXY(3,2); ClrEol; TextColor(Yellow);
      WrStr('Dual frequency measurement(s) have been completed.');
      Alarm(4000,100,1); Delay(1000);
    ELSE
      GotoXY(3,2); ClrEol; TextColor(LightRed);
      WrStr('Process aborted!'); Alarm(3500,100,2); pause := RdKey();
    END; Close(setup_window); Close(work_window); CursorOn;
END DualFreqMeasurements;

END EITDFreq.
```

```
IMPLEMENTATION MODULE EITDisp; (* *)

FROM Window IMPORT
     WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,
     SetTitle,Close,Clear,CursorOn,CursorOff,TextColor,Use,Hide,
     CenterUpperTitle,PutOnTop,FullScreenDef,GotoXY,ClrEol;

IMPORT IO,FIO,Lib,Storage,GraphTxt,EITUtil,EITGraph,Graph;

(*$I+,O+,S+,R+*)

CONST null = 0C; esc = 33C; numberofcolor = 16;

VAR   a             : ADDRESS;
      portnum       : CARDINAL;
      comport       : CARDINAL;
      i,j,k         : CARDINAL;
      x,y,lx,ly,color : CARDINAL;
      xstart,ystart   : CARDINAL;
      barxstart,barlx : CARDINAL;
      barystart,barly : CARDINAL;
      freq          : CARDINAL;
      pause,redo      : CHAR;
      monitor         : CHAR;
      max,min         : REAL;
      automax,automin : REAL;
      lowinlev,upwinlev : REAL;
      resolution      : REAL;
      scaleddata      : REAL;
      scalefactor     : REAL;
      mapindex        : ARRAY [1..1936] OF CHAR;
      imagedata       : ARRAY [1..1936] OF REAL;
      imagecolor      : ARRAY [1..1936] OF CARDINAL;
      filename        : ARRAY [1..78] OF CHAR;
      infile,imagefile: FIO.File;
      menu_window,graph_window,FullScreen   : WinType ;

PROCEDURE DisplayData;
VAR change,default : CHAR; graylevel : INTEGER;
    winltd : REAL;
BEGIN
    REPEAT
    GotoXY(1,9); ClrEol; GotoXY(1,11); ClrEol; GotoXY(1,12); ClrEol;
    GotoXY(3,9); IO.WrStr('Do you want to alter the window limits? [N] ');
    change := IO.RdKey();
    IF (change = 'Y') OR (change = 'y') THEN
      GotoXY(1,9); ClrEol;
      GotoXY(3,9); IO.WrStr('Do you want to use default window limits? [Y] ');
      default := IO.RdKey();
      IF (default = 'N') OR (default = 'n') THEN
        GotoXY(3,11); IO.WrStr('Window upper limit = '); upwinlev := IO.RdReal();
        GotoXY(3,12); IO.WrStr('Window lower limit = '); lowinlev := IO.RdReal();
      ELSE
        upwinlev := automax; lowinlev := automin;
      END;
      GotoXY(40,6); IO.WrStr('Window maximum = '); IO.WrReal(upwinlev,4,9);
      GotoXY(40,7); IO.WrStr('Window minimum = '); IO.WrReal(lowinlev,4,9);
    END;
    UNTIL (change # 'y') AND (change # 'Y');
    scalefactor := (upwinlev-lowinlev)/resolution;
(*
    IF ABS(max) > ABS(min) THEN winltd := max ELSE winltd := ABS(min) END;
    lowinlev := -winltd*0.9;
    upwinlev := winltd*0.9;
    scalefactor := (upwinlev-lowinlev)/resolution;
*)
    FOR i := 1 TO 1936 DO
      IF mapindex[i] # '0' THEN
        scaleddata := (imagedata[i]-lowinlev)/scalefactor;
        graylevel := TRUNC(scaleddata);
        IF graylevel > 15 THEN imagecolor[i] := 15
        ELSIF graylevel < 0 THEN imagecolor[i] := 0
        ELSE imagecolor[i] := graylevel
        END
      END;
    END;
END DisplayData;

PROCEDURE PrintData;
VAR i,graylevel : INTEGER; printfile : FIO.File;
BEGIN
(* N.B. lowinlev and upwinlev are the same as in the monitor display mode
    lowinlev := min+(max-min)*0.05;
    upwinlev := min+(max-min)*0.95;
*)
    scalefactor := (upwinlev-lowinlev)/resolution;
    FOR i := 1 TO 1936 DO
      IF mapindex[i] # '0' THEN
        scaleddata := (imagedata[i]-lowinlev)/scalefactor;
        graylevel := TRUNC(scaleddata);
        IF graylevel > 15 THEN imagecolor[i] := 15
        ELSIF graylevel < 0 THEN imagecolor[i] := 0
        ELSE imagecolor[i] := graylevel
        END
      ELSE imagecolor[i] := 16
      END
    END;
    Clear; IO.WrLn; IO.WrStr('  Encoding image data for printing.  ');
    IO.WrStr('Please wait... ');
    printfile := FIO.Create('image.prn');
    FOR i := 1 TO 1936 DO
      FIO.WrCard(printfile,imagecolor[i],3);
      FIO.WrLn(printfile);
    END;
    FIO.Close(printfile);
END PrintData;
```

```
PROCEDURE GetImageData;
VAR winltd : REAL;
BEGIN
   max := -999.; min := 999.;
   FOR i := 1 TO 1936 DO imagecolor[i] := 16 END;
   imagefile := FIO.Open(filename);
   Clear; IO.WrLn;
   IO.WrStr('  Image data are being read from ');
   TextColor(Yellow); IO.WrStr(filename); TextColor(White);
   IO.WrLn; IO.WrLn; IO.WrStr('  Please wait... ');
   FOR i := 1 TO 1936 DO
     imagedata[i] := FIO.RdReal(imagefile);
     IF mapindex[i] # '0' THEN
       IF imagedata[i] > max THEN max := imagedata[i]
       ELSIF imagedata[i] < min THEN min := imagedata[i]
       END;
     END;
   END; EITUtil.Alarm(3500,100,1);
   FIO.Close(imagefile);
   IF ABS(max) > ABS(min) THEN winltd := max ELSE winltd := ABS(min) END;
   automin := -winltd*0.9; lowinlev := automin;
   automax := winltd*0.9; upwinlev := automax;
END GetImageData;

PROCEDURE MonitorColor(displaycolor: CARDINAL);
BEGIN
   IF monitor='1' THEN
     CASE displaycolor OF
       0: color :=  0|
       1: color :=  3|
       2: color := 12|
       3: color :=  4|
       4: color :=  1|
       5: color :=  5|
       6: color :=  7|
       7: color := 13|
       8: color := 15|
       9: color :=  6|
      10: color :=  9|
      11: color := 14|
      12: color := 11|
      13: color :=  2|
      14: color :=  8|
      15: color := 10|
      16: color :=  0
     END
   ELSE
     CASE displaycolor OF
       0: color :=  17|
       1: color := 288|
       2: color := 305|
       3: color := 313|
       4: color := 307|
       5: color := 377|
       6: color := 355|
       7: color := 383|
       8: color := 190(*495*)|
       9: color := 191(*510*)|
      10: color := 447|
      11: color := 445|
      12: color := 444|
      13: color := 412|
      14: color := 404|
      15: color := 400|
      16: color :=  0
     END
   END
END MonitorColor;

PROCEDURE ColorBar;
VAR xmin,xmax,ymin,ymax : CARDINAL;
BEGIN
   xmin := barxstart-2;
   ymin := barystart-2;
   xmax := barxstart+barlx+2;
   ymax := barystart+barly*numberofcolor+2;
   FOR i := 0 TO (numberofcolor-1) DO
     y := barystart+barly*i; MonitorColor(numberofcolor-i-1);
     EITUtil.Paize(barxstart,y,barlx,barly,color)
   END;
   EITGraph.Line(xmin,ymin,xmax,ymin,15);
   EITGraph.Line(xmax,ymin,xmax,ymax,15);
   EITGraph.Line(xmax,ymax,xmin,ymax,15);
   EITGraph.Line(xmin,ymax,xmin,ymin,15);
END ColorBar;

PROCEDURE MakeImage;
BEGIN
   ColorBar;
   FOR i := 0 TO 43 DO
   FOR j := 1 TO 44 DO
     k := j+44*i;
     x := xstart+j*lx;
     y := ystart+i*ly;
     MonitorColor(imagecolor[k]);
     EITUtil.Paize(x,y,lx,ly,color);
   END;
   END;
END MakeImage;

PROCEDURE WindowSetting;
BEGIN
END WindowSetting;

PROCEDURE SetUpMode;
```

```
VAR endgraph: BOOLEAN;
BEGIN
   endgraph := TRUE; resolution := 15.;
   PutOnTop(graph_window);
   REPEAT
   Clear; GotoXY(3,2); IO.WrStr('Possible monitor types:');
   GotoXY(3,4); EITUtil.ChColor('1',Brown,Yellow); IO.WrStr('. CGA');
   GotoXY(3,5); EITUtil.ChColor('2',Brown,Yellow); IO.WrStr('. EGA');
   GotoXY(3,6); EITUtil.ChColor('3',Brown,Yellow); IO.WrStr('. VGA');
   GotoXY(3,7); EITUtil.ChColor('8',Brown,Yellow); IO.WrStr('. Print image');
   TextColor(White);
   GotoXY(40,4); IO.WrStr('Maximum value = '); IO.WrReal(max,4,9);
   GotoXY(40,5); IO.WrStr('Minimum value = '); IO.WrReal(min,4,9);
   GotoXY(40,6); IO.WrStr('Window maximum = '); IO.WrReal(upwinlev,4,9);
   GotoXY(40,7); IO.WrStr('Window minimum = '); IO.WrReal(lowinlev,4,9);
   GotoXY(3,9); IO.WrStr('Press <'); EITUtil.ChColor('esc',LightRed,White);
   IO.WrStr('> to quit graphics mode');
   GotoXY(3,11); IO.WrStr('Please enter ... '); monitor := IO.RdKey();
   IF monitor # esc THEN
   CASE monitor OF
   |'1' : EITGraph.InitCGA; DisplayData;
        barxstart := 15; barystart := 16;
        barlx := 10; barly := TRUNC(160./(resolution+1.));
        xstart := 50; ystart := 10; lx := 5; ly := 4;
        Graph.InitCGA; Graph.GraphMode;
        MakeImage; pause := IO.RdKey();
        EITGraph.TextMode; PutOnTop(FullScreen);
   |'2' : EITGraph.InitEGA; DisplayData;
        barxstart := 50; barystart := 45;
        barlx := 20; barly := TRUNC(240./(resolution+1.));
        xstart := 150; ystart := 30; lx := 8; ly := 6;
        Graph.InitEGA; Graph.GraphMode; MakeImage;
        GraphTxt.LrgWrStr('max',48,28,2);
        GraphTxt.LrgWrStr('min',48,294,2);
        GraphTxt.LrgWrStr('[1]',322,25,2);
        GraphTxt.LrgWrStr('[5]',142,162,2);
        GraphTxt.LrgWrStr('[9]',322,294,2);
        GraphTxt.LrgWrStr('[13]',502,162,2);
        pause := IO.RdKey();
        EITGraph.TextMode; PutOnTop(FullScreen);
   |'3' : EITGraph.InitVGA; DisplayData;
        barxstart := 50; barystart := 52;
        barlx := 20; barly := TRUNC(360./(resolution+1.));
        xstart := 150; ystart := 30; lx := 9; ly := 9;
        Graph.InitVGA; Graph.GraphMode; MakeImage;
        GraphTxt.LrgWrStr('max',48,34,2);
        GraphTxt.LrgWrStr('min',48,420,2);
        GraphTxt.LrgWrStr('[1]',346,25,2);
        GraphTxt.LrgWrStr('[5]',144,228,2);
        GraphTxt.LrgWrStr('[9]',346,426,2);
        GraphTxt.LrgWrStr('[13]',546,228,2);
        pause := IO.RdKey();
        EITGraph.TextMode; PutOnTop(FullScreen);
   |'8' : PrintData;
        Storage.ALLOCATE(a,65535);
        i := Lib.Execute("hcopy.exe"," ",a,65535 DIV 8);
        IF i # 0 THEN IO.WrStr('Failed'); END;
        Storage.DEALLOCATE(a,65535);
        EITGraph.TextMode; PutOnTop(FullScreen)
   |'9' : WindowSetting;
   ELSE EITUtil.Alarm(3500,100,2) END;
   ELSE endgraph := FALSE END;
   PutOnTop(menu_window); PutOnTop(graph_window);
   UNTIL endgraph = FALSE; Hide(graph_window)
END SetUpMode;

PROCEDURE Main(portnum : CARDINAL);
BEGIN
   comport := portnum;
   FullScreen := Open(FullScreenDef);
   EITUtil.Banner(comport); TextColor(White);
   infile := FIO.Open('apt01.dat');
   FOR i:=1 TO 1936 DO mapindex[i] := FIO.RdChar(infile) END;
   FIO.Close(infile);
   menu_window := Open(WinDef(0,1,79,9,White,Blue,TRUE,FALSE,
                  FALSE,TRUE,DoubleFrame,White,Blue));
   SetTitle (menu_window," Graphics Routines ",CenterUpperTitle);
   graph_window := Open(WinDef(0,10,79,25,White,Blue,TRUE,FALSE,
                  TRUE,TRUE,DoubleFrame,White,Blue));
   SetTitle (graph_window," Graphics Window ",CenterUpperTitle);
   PutOnTop(menu_window);
   REPEAT
   Clear; IO.WrLn;
   IO.WrStr('  Please enter the name of the image information file... ');
   IO.RdItem(filename);
   IF FIO.Exists(filename) THEN
     GetImageData;
     SetUpMode;
   ELSE
     Clear; GotoXY(3,2);
     EITUtil.ChColor('ERROR: ',Red,Yellow); IO.WrStr(filename);
     EITUtil.ChColor(' does not exist!',Red,White); IO.WrLn;
     EITUtil.Alarm(3500,100,2);
   END; PutOnTop(menu_window); GotoXY(3,4);
   IO.WrStr('Another go? [Y] '); redo := IO.RdKey();
   UNTIL (redo = 'N') OR (redo = 'n') OR (redo = esc);
   Close(menu_window); Close(graph_window); Close(FullScreen);
   CursorOn;
END Main;
(*
BEGIN Main(1)
*)
END EITDisp.
```

```
IMPLEMENTATION MODULE EITGraph;

IMPORT Lib, SYSTEM;

(*$N,V-,I-,R-,A-,S-*)

TYPE
 tinyint    = [0..7];
 bs         = SET OF tinyint;
 bp         = POINTER TO bs;
VAR
 EGAScreen [0A000H:0] : ARRAY[0..0] OF bs;

(* == CGA specific routines == *)

PROCEDURE CGAGraphMode;
VAR r : SYSTEM.Registers;
BEGIN
 r.AX := 5;
 Lib.Intr( r,10H );
END CGAGraphMode;

PROCEDURE CGATextMode;
VAR r : SYSTEM.Registers;
BEGIN
 r.AX := 3;
 Lib.Intr( r,10H );
END CGATextMode;

PROCEDURE CGAPlot(x,y:CARDINAL;c:CARDINAL);
VAR off,seg,tmp : CARDINAL;
BEGIN
 IF (x >= CGAWidth) OR (y >= CGADepth) THEN RETURN END;
 off := x >> 2;
 IF ODD(y) THEN INC( off, 2000H - 40 ) END;
 INC( y, y << 2 );
 INC( off, y << 3 );
 x := 3 - CARDINAL( BITSET(x) * BITSET(3) );
 x := x << 1;
 tmp := 0B800H; seg := tmp;
 [seg:off bp]^ := ( [seg:off bp]^ - bs(3<<x) ) + bs(c<<x);
END CGAPlot;

PROCEDURE CGAHLine ( x,y,x2 : CARDINAL; c:CARDINAL );
VAR off,seg,tmp,n : CARDINAL; w,mask : bs; fillc: SHORTCARD;
BEGIN
 IF y > CGADepth-1 THEN RETURN END;
 IF INTEGER(x) >= INTEGER(CGAWidth) THEN RETURN END;
 IF INTEGER(x) < 0 THEN x := 0; END;
 IF x2 >= CGAWidth THEN x2 := CGAWidth-1 END;
 n := ( x2 - x ) + 1;
 off := x >> 2;
 IF ODD(y) THEN
  INC( off, 2000H - 40 );
  c := ( c >> 2 + c << 2 ) MOD 16;
 END;
 c := c + c * 16;
 INC( y, y << 2 );
 INC( off, y << 3 );
 x := 3 - CARDINAL( BITSET(x) * BITSET(3) );
 x := x << 1;
 tmp := 0B800H; seg := tmp;
 w := [seg:off bp]^;
 REPEAT
  mask := bs(3 << x);
  w := ( w - mask ) + bs(c)*mask;
  DEC(n);
  DEC(x,2);
 UNTIL (n=0) OR (x=CARDINAL(-2));
 [seg:off bp]^ := w;
 INC(off);
 Lib.Fill( [seg:off], n >> 2, SHORTCARD(c) );
 INC( off, n >> 2 );
 n := n MOD 4;
 x := 6;
 w := [seg:off bp]^;
 WHILE n # 0 DO
  mask := bs(3 << x);
  w := ( w - mask ) + bs(c)*mask;
  DEC(n);
  DEC(x,2);
 END;
 [seg:off bp]^ := w;
END CGAHLine;

PROCEDURE InitCGA ;
BEGIN
 Width := CGAWidth ;
 Depth := CGADepth ;
 NumColor := 4 ;
 TextMode := CGATextMode ;
 GraphMode := CGAGraphMode ;
 Plot := CGAPlot ;
 HLine := CGAHLine ;
END InitCGA ;

(* == EGA/VGA specific routines == *)

PROCEDURE EGAGraphMode; (* Also VGA *)
VAR r : SYSTEM.Registers;
BEGIN
 IF Depth=480 THEN r.AX := 12H ELSE r.AX := 10H END ;
 Lib.Intr( r,10H );
END EGAGraphMode;

PROCEDURE EGAPlot( x,y,c : CARDINAL); (* Also VGA *)
```

```
VAR t:ba; p,b,s:CARDINAL;
BEGIN
  IF (x < EGAWidth) AND (y < Depth) THEN
    b := 1 << (7-(x MOD 8));
    p := y*80+(x DIV 8);
    SYSTEM.Out( 3CEH,8);SYSTEM.Out( 3CFH,SHORTCARD(b));
    SYSTEM.Out( 3C4H,2);SYSTEM.Out( 3C5H,0FH);
    s := 0A000H;
    t := [s:p bp]^;
    [s:p bp]^ := bs{};
    SYSTEM.Out( 3C4H,2);SYSTEM.Out( 3C5H,SHORTCARD(c));
    [s:p bp]^ := bs{0..7};
    SYSTEM.Out( 3CEH,8);SYSTEM.Out( 3CFH,0FFH);
    SYSTEM.Out( 3C4H,2);SYSTEM.Out( 3C5H,0FH);
  END;
END EGAPlot;

PROCEDURE EGAHLine ( x,y,x2 : CARDINAL; c:CARDINAL );
VAR c1,c2,nc : CARDINAL;
BEGIN
 IF y > Depth-1 THEN RETURN END;
 IF INTEGER(x) > EGAWidth-1 THEN RETURN END;
 IF INTEGER(x) < 0 THEN x := 0; END;
 IF x2 > Width-1 THEN x2 := Width-1 END;

(* nc = 2 ;  16 colors *)
(* nc = 4 ;  32 colors *)
(* nc = 8 ;  64 colors *)
(* nc = 16; 128 colors *)
(* nc = 32; 256 colors *)

 nc := 32;
 c1 := c MOD nc;
 c2 := c DIV nc;
 WHILE (x <= x2) DO
     IF ODD(x+y) THEN
       Plot( x  , y , c1 );
     ELSE
       Plot( x  , y , c2 );
     END;
     INC( x );
 END;
 IF INTEGER(x) > INTEGER(x2) THEN RETURN; END;
 SYSTEM.Out( 3CEH,8);SYSTEM.Out( 3CFH,0FFH);
 SYSTEM.Out( 3C4H,2);SYSTEM.Out( 3C5H,0FH);
 SYSTEM.Out( 3CEH,5);SYSTEM.Out( 3CFH,2);
 y := y*80;
 x := x DIV 8;
 x2 := x2 DIV 8;
 WHILE x <= x2 DO
  EGAScreen[y+x] := bs(c);
  INC( x );
 END;
 SYSTEM.Out( 3CEH,5);SYSTEM.Out( 3CFH,0);
END EGAHLine;

PROCEDURE InitEGA ;
BEGIN
 Width    := EGAWidth ;
 Depth    := EGADepth ;
 NumColor := 16 ;
 TextMode := CGATextMode ;
 GraphMode := EGAGraphMode ;
 Plot     := EGAPlot ;
 HLine    := EGAHLine ;
END InitEGA ;

PROCEDURE InitVGA ;
BEGIN
 InitEGA ;
 Depth := VGADepth ; (* Width same as EGA *)
END InitVGA ;

(* ——— Device independant routines ——— *)

PROCEDURE Line(x1,y1,x2,y2: CARDINAL; c: CARDINAL);
VAR dx,dy,e,tmp : INTEGER;
BEGIN
 IF x1 > x2 THEN (* ensure that x2 >= x1 *)
  tmp := x1; x1 := x2; x2 := tmp;
  tmp := y1; y1 := y2; y2 := tmp;
 END;
 dx := x2-x1;
 e := 0;
 IF y1 <= y2 THEN (* case where y increases *)
  dy := (y2-y1);
  IF dx >= dy THEN
   LOOP
    Plot( x1,y1,c );
    IF x1 = x2 THEN EXIT END;
    INC(x1);
    INC(e,dy);
    INC(e,dy);
    IF e > dx THEN
     DEC(e,dx);
     DEC(e,dx);
     INC(y1);
    END;
   END;
  ELSE
   LOOP
    Plot( x1,y1,c );
    IF y1 = y2 THEN EXIT END;
    INC(y1);
    INC(e,dx);
    INC(e,dx);
```

```
    IF e > dy THEN
      DEC(e,dy);
      DEC(e,dy);
      INC(x1);
    END;
   END;
  END;
 ELSE
  (* case where y decreases *)
  dy := (y1-y2);
  IF dx >= dy THEN
   LOOP
     Plot( x1,y1,c );
     IF x1 = x2 THEN EXIT END;
     INC(x1);
     INC(e,dy);
     INC(e,dy);
     IF e > dx THEN
       DEC(e,dx);
       DEC(e,dx);
       DEC(y1);
     END;
    END;
   ELSE
    LOOP
     Plot( x1,y1,c );
     IF y1 = y2 THEN EXIT END;
     DEC(y1);
     INC(e,dx);
     INC(e,dx);
     IF e > dy THEN
       DEC(e,dy);
       DEC(e,dy);
       INC(x1);
     END;
    END;
   END;
  END;
 END Line;

END EITGraph.
```

```
IMPLEMENTATION MODULE EITNoise;(* *)

FROM Window IMPORT
     WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,
     SetTitle,Close,Clear,CursorOn,CursorOff,TextColor,Use,Hide,Used,
     CenterUpperTitle,PutOnTop,GotoXY,FullScreenDef,ClrEol,SetFrame;
FROM IO IMPORT
     WrInt,WrStr,WrLn,RdKey,RdItem,KeyPressed;
FROM Lib IMPORT Delay;
FROM EITUtil IMPORT
     Alarm,ChColor,WrDec;

IMPORT FIO,MATHLIB;

CONST cr = 15C; esc = 33C;

VAR ansop : CHAR; abort : BOOLEAN;
   optdata : ARRAY [1..16] OF ARRAY [1..16] OF REAL;
   datamap : ARRAY [1..16] OF ARRAY [1..16] OF CHAR;
   file1,file2 : FIO.File;
   filename1,filename2 : ARRAY [1..78] OF CHAR;
   menu_window,work_window,option_window,result_window : WinType;

PROCEDURE DispOpt1;
VAR data,m,n : INTEGER;
BEGIN
   GotoXY(7,1);
   FOR m := 1 TO 16 DO TextColor(Green); WrInt(m,4) END;
   FOR n := 1 TO 16 DO
     GotoXY(1,n+1); TextColor(Green); WrInt(n,4); WrStr('  ');
   FOR m := 1 TO 16 DO
     data := TRUNC(optdata[m,n]);
     IF datamap[m,n] = '1' THEN TextColor(Black); WrStr(' ---')
     ELSIF datamap[m,n] = '2' THEN TextColor(Yellow); WrStr(' NNN')
     ELSE
       IF (n=1) AND (m=16) THEN TextColor(Black); WrStr(' ---')
       ELSIF m >= n+2 THEN
       IF (data >= 999) OR (data <= -999)
       THEN TextColor(LightRed); WrStr(' XXX')
       ELSE
        IF m MOD 2 = 0 THEN
          TextColor(Cyan); WrInt(data,4)
        ELSE
          TextColor(LightGray); WrInt(data,4)
       END
       END
      ELSE TextColor(Black); WrStr(' ---')
      END
     END;

   END; GotoXY(71,n+1); TextColor(Green); WrInt(n,4);
   END
END DispOpt1;

PROCEDURE ReciError;
VAR skip : BOOLEAN; anstry : CHAR; m,n,number : INTEGER;
   data : ARRAY [1..16] OF ARRAY [1..16] OF REAL;
   attfactor : ARRAY [1..16] OF ARRAY [1..16] OF INTEGER;
   sumsqerror,denumerator,RMSerror : REAL;
BEGIN
   SetTitle(option_window," Reciprocity Errors ",CenterUpperTitle);
   SetFrame(option_window,SingleFrame,LightGray,Blue);
   PutOnTop(option_window); Clear; skip := FALSE;
   LOOP
   GotoXY(3,1); WrStr('Enter a data file : '); RdItem(filename1);
   IF FIO.Exists(filename1) THEN
     file1 := FIO.Open(filename1); GotoXY(23,1);
     ChColor(filename1,Yellow,White);
     GotoXY(3,3); WrStr('Please wait... ');
     EXIT
   ELSE Alarm(3500,100,2); Clear;
     ChColor('  File does not exist!',LightRed,White);
     WrStr('  Try again? [Y] '); anstry := RdKey(); Clear;
     IF (anstry = 'n') OR (anstry = 'N') THEN skip := TRUE; EXIT END;
   END;
   END; (* of LOOP *)
   IF skip = FALSE THEN
   FOR n := 1 TO 16 DO
   FOR m := 1 TO 16 DO
     data[m,n] := FIO.RdReal(file1); attfactor[m,n] := FIO.RdInt(file1);
     IF attfactor[m,n] = 0 THEN datamap[m,n] := '1' ELSE datamap[m,n] := '0' END;
   END; END; FIO.Close(file1);
   number := 0;
   FOR n := 1 TO 14 DO
   FOR m := n+2 TO 16 DO
     IF attfactor[m,n] = attfactor[n,m] THEN
       IF NOT ((n=1) AND (m=16)) THEN
       number := number+1;
       IF datamap[m,n] = '0' THEN
         denumerator := data[m,n]+data[n,m];
         IF denumerator = 0. THEN
           optdata[m,n] := 0.
         ELSE
           optdata[m,n] := 2.E3*(data[m,n]-data[n,m])/denumerator
         END;
       END;
       END;
     ELSE datamap[m,n] := '2'
     END;
   END END;
   sumsqerror := 0.;
   FOR n := 1 TO 14 DO
   FOR m := n+2 TO 16 DO
     IF datamap[m,n] = '0' THEN
       IF NOT ((n=1) AND (m=16)) THEN
         sumsqerror := sumsqerror+(optdata[m,n]*optdata[m,n])
```

```
      END;
    END;
  END END;
  RMSerror := REAL(MATHLIB.Sqrt(LONGREAL(sumsqerror/REAL(number))));
  GotoXY(1,2); ClrEol;
  GotoXY(3,2); ChColor('Number of term compared = ',White,Yellow);
  WrInt(number,-3);
  GotoXY(49,3); ChColor('RMS error (x0.1%) = ',White,Yellow);
  WrDec(LONGREAL(RMSerror),2);
  PutOnTop(result_window); Clear; DispOpt1;
  PutOnTop(option_window); Alarm(4000,100,1);
  GotoXY(3,3); TextColor(White); WrStr('Strike any key to continue');
  CursorOn; anstry := RdKey();
  Hide(result_window); Hide(option_window)
  END; (* of IF skip *)
END RecError;

PROCEDURE DispOpt2;
VAR data,i,j,m,n : INTEGER;
  vdata : ARRAY [1..13] OF ARRAY [1..16] OF INTEGER;
BEGIN
  FOR n := 1 TO 16 DO
  FOR i := 1 TO 13 DO
    m := n+1+i;
    IF m <= 16 THEN
      vdata[i,n] := TRUNC(optdata[m,n])
    ELSE
      vdata[i,n] := TRUNC(optdata[m-16,n])
    END
  END END;
  GotoXY(11,1);
  FOR m := 1 TO 13 DO TextColor(Green); WrInt(m,4) END;
  FOR n := 1 TO 16 DO
    GotoXY(4,n+1); TextColor(Green);
    IF n = 1 THEN WrInt(16,2) ELSE WrInt(n-1,2) END;
    WrStr(' - '); WrInt(n,-2);
  FOR m := 1 TO 13 DO
    IF (vdata[m,n] >= 999) OR (vdata[m,n] <= -999)
    THEN TextColor(LightRed); WrStr(' XXX')
    ELSE
      IF m MOD 2 = 0 THEN
        TextColor(Cyan); WrInt(vdata[m,n],4)
      ELSE
        TextColor(LightGray); WrInt(vdata[m,n],4)
      END
    END;
  END;
    GotoXY(65,n+1); TextColor(Green);
    IF n = 1 THEN WrInt(16,2) ELSE WrInt(n-1,2) END;
    WrStr(' - '); WrInt(n,-2);
  END
END DispOpt2;

PROCEDURE NoiseLevel;
VAR m,n,attfactor : INTEGER; anstry : CHAR; skip : BOOLEAN;
  data1,data2 : ARRAY [1..16] OF ARRAY [1..16] OF REAL;
  sqdiff,SD,SUM,MEAN,RMSerror,denumerator : REAL;
  outfile : FIO.File;
BEGIN
  SetTitle(option_window," Noise Level on Images Data ",CenterUpperTitle);
  SetFrame(option_window,SingleFrame,LightGray,Blue);
  PutOnTop(option_window); Clear; skip := FALSE;
  LOOP
  WrStr(' Enter the first data file : '); RdItem(filename1);
  WrStr(' Enter the second data file: '); RdItem(filename2);
  IF (FIO.Exists(filename1)) AND (FIO.Exists(filename2)) THEN
    file1 := FIO.Open(filename1); file2 := FIO.Open(filename2); Clear;
    WrStr(' Data files are '); ChColor(filename1,Yellow,White);
    WrStr(' and '); ChColor(filename2,Yellow,White); WrLn;
    WrStr(' Please wait... '); EXIT
  ELSE Alarm(3500,100,2); Clear;
     ChColor(' File(s) do not exist!',LightRed,White);
     WrStr(' Try again? [Y] '); anstry := RdKey(); Clear;
     IF (anstry = 'n') OR (anstry = 'N') THEN skip := TRUE; EXIT END;
  END;
  END; (* of LOOP *)
  IF skip = FALSE THEN
  FOR n := 1 TO 16 DO
  FOR m := 1 TO 16 DO
    data1[m,n] := FIO.RdReal(file1); attfactor := FIO.RdInt(file1);
    data2[m,n] := FIO.RdReal(file2); attfactor := FIO.RdInt(file2);
    IF attfactor = 0 THEN datamap[m,n] := '1' ELSE datamap[m,n] := '0' END;
  END END; FIO.Close(file1); FIO.Close(file2);
  SUM := 0.;
  FOR n := 1 TO 16 DO
  FOR m := 1 TO 16 DO
    denumerator := data1[m,n]+data2[m,n];
    IF denumerator = 0. THEN optdata[m,n] := 0.
    ELSE optdata[m,n] := 2.E3*(data1[m,n]-data2[m,n])/denumerator
    END;
    SUM := SUM+optdata[m,n];
  END END;
  MEAN := SUM/208.; SUM := 0.;
  FOR n := 1 TO 16 DO
  FOR m := 1 TO 16 DO
    SUM := SUM+optdata[m,n]*optdata[m,n];
  END END;
  RMSerror := REAL(MATHLIB.Sqrt(LONGREAL(SUM)/208.)); SUM := 0.;
  FOR n := 1 TO 16 DO
  FOR m := 1 TO 16 DO
    sqdiff := (MEAN-optdata[m,n])*(MEAN-optdata[m,n]);
    SUM := SUM+sqdiff
  END END;
  SD := REAL(MATHLIB.Sqrt(LONGREAL(SUM)/208.));
  GotoXY(1,2); ClrEol; GotoXY(49,1);
  ChColor('Mean (x0.1%) = ',White,Yellow); WrDec(LONGREAL(MEAN),2);
```

```
    GotoXY(49,2); ChColor('RMS (x0.1%)  = ',White,Yellow);
    WrDec(LONGREAL(RMSerror),2);
    GotoXY(49,3); ChColor('SD (x0.1%)  = ',White,Yellow);
    WrDec(LONGREAL(SD),2);
    PutOnTop(result_window); Clear; DispOpt2;
    PutOnTop(option_window); Alarm(4000,100,1);
    GotoXY(3,3); TextColor(White); WrStr('Strike any key to continue');
    CursorOn; anstry := RdKey();
    Hide(result_window); Hide(option_window)
    END; (* of IF skip *)
END NoiseLevel;

PROCEDURE main;
BEGIN
    menu_window := Open(WinDef(0,1,79,10,White,Blue,FALSE,FALSE,
                    FALSE,TRUE,DoubleFrame,White,Blue));
    SetTitle(menu_window," System Performance Analyses ",CenterUpperTitle);
    work_window := Open(WinDef(0,11,79,14,White,Blue,TRUE,FALSE,
                    FALSE,TRUE,SingleFrame,White,Blue));
    SetTitle(work_window," Working Window ",CenterUpperTitle);
    result_window := Open(WinDef(0,6,79,24,White,Blue,TRUE,FALSE,
                    TRUE,TRUE,DoubleFrame,Yellow,Blue));
    SetTitle(result_window," Proportional Errors (x0.1%) ",CenterUpperTitle);
    option_window := Open(WinDef(0,1,79,5,White,Blue,TRUE,FALSE,
                     FALSE,TRUE,SingleFrame,White,Blue));
    PutOnTop(menu_window);
    GotoXY(3,2); WrStr('Selection of possible operations:');
    GotoXY(3,4); ChColor('1',Brown,Yellow);
    WrStr('.  Reciprocity Errors');
    GotoXY(3,5); ChColor('2',Brown,Yellow);
    WrStr('.  Noise Level on Images Data');
    GotoXY(3,7); ChColor('Press <',White,LightRed); WrStr('esc');
    TextColor(White); WrStr('> to quit System Performance Analyses');
    REPEAT
    PutOnTop(menu_window);
    PutOnTop(work_window); TextColor(White); CursorOn;
    GotoXY(3,1); WrStr('Please enter...'); ansop := RdKey();
    GotoXY(1,2); ClrEol;
    CASE ansop OF
    |'1': Hide(menu_window); Hide(work_window); ReclError;
    |'2': Hide(menu_window); Hide(work_window); NoiseLevel;
    |esc: ansop := esc
    ELSE Alarm(3500,100,2); GotoXY(3,2); WrStr('Either <1> or <2>.');
    END;
    UNTIL ansop = esc;
    Close(menu_window); Close(work_window); Close(result_window);
    Close(option_window)
END main;

(*
BEGIN
   main
*)
END EITNoise.
```

```
IMPLEMENTATION MODULE EITSetUp;

FROM IO IMPORT WrChar,WrLn,WrStr,WrShtCard,WrShtHex,RdChar,RdKey,KeyPressed;
FROM EITCom IMPORT Install,Init,RxCount,TxCount,Send,Receive,Even,None;
FROM Window IMPORT
    WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,SetTitle,Close,Clear,
    CursorOn,CursorOff,TextColor,Use,GotoXY,CenterUpperTitle,PutOnTop,ClrEol;
FROM Lib IMPORT Delay;
FROM EITUtil IMPORT Alarm,ChColor,ClrRxCount;
IMPORT FIO;

TYPE char_array = ARRAY [1..78] OF CHAR;
   PtrType = (amp,freq);

CONST esc = 33C; cr = 15C;

VAR   Yposition     : CARDINAL;
     ccode,fcode,setupcode     : BYTE;
     parameter     : PtrType;
     code_window,setup_window   : WinType;
     initfile      : FIO.File;
     cdata,fdata,pause    : CHAR;
     cstr,fstr     : char_array;

PROCEDURE InitData;
BEGIN
   initfile := FIO.Open('initial.dat');
   cdata := FIO.RdChar(initfile); fdata := FIO.RdChar(initfile);
   CASE fdata OF
   '1': fcode := 08H; fstr := '10' |'2': fcode := 10H; fstr := '20';
   '3': fcode := 00H; fstr := '40' |'4': fcode := 18H; fstr := '80';
   END;
   CASE cdata OF
   '1': ccode := 00H; cstr := '0.00' |'2': ccode := 01H; cstr := '0.00';
   '3': ccode := 02H; cstr := '0.00' |'4': ccode := 03H; cstr := '0.00';
   '5': ccode := 04H; cstr := '1.00' |'6': ccode := 05H; cstr := '1.50';
   '7': ccode := 06H; cstr := '2.00' |'8': ccode := 07H; cstr := '2.50';
   END; FIO.Close(initfile);
   code_window := Open(WinDef(0,22,35,24,LightGray,Black,FALSE,FALSE,
            FALSE,TRUE,SingleFrame,LightGray,Black));
   SetTitle(code_window," Transmission Code ",CenterUpperTitle);
   Use(code_window); Clear; WrLn;
   WrStr('  The setting code (in HEX) = ');
   setupcode := ccode+fcode;
   TextColor(Cyan); WrShtHex(setupcode,1); TextColor(LightGray);
   Send(setupcode,1);
END InitData;

PROCEDURE SetAmp;
VAR select : CHAR; amp_window : WinType;
BEGIN
   amp_window := Open(WinDef(47,10,77,24,LightGray,Black,TRUE,FALSE,
            FALSE,TRUE,SingleFrame,LightGray,Black));
   SetTitle(amp_window," Amplitude Set Up ",CenterUpperTitle);
   Use(amp_window); Clear; WrLn;
   WrStr('  Selection of amplitudes'); WrLn; WrLn;
   WrStr('  ['); ChColor('1',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('2',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('3',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('4',Brown,LightGray); WrStr('] 0.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('5',Brown,LightGray); WrStr('] 1.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('6',Brown,LightGray); WrStr('] 1.50 mA rms'); WrLn;
   WrStr('  ['); ChColor('7',Brown,LightGray); WrStr('] 2.00 mA rms'); WrLn;
   WrStr('  ['); ChColor('8',Brown,LightGray); WrStr('] 2.50 mA rms');
   LOOP
    GotoXY(3,13); WrStr('Please enter... ');
    select := RdKey();
    CASE select OF
    '1': cdata := '1'; ccode := 00H; cstr := '0.00'; EXIT
    '2': cdata := '2'; ccode := 01H; cstr := '0.00'; EXIT
    '3': cdata := '3'; ccode := 02H; cstr := '0.00'; EXIT
    '4': cdata := '4'; ccode := 03H; cstr := '0.00'; EXIT
    '5': cdata := '5'; ccode := 04H; cstr := '1.00'; EXIT
    '6': cdata := '6'; ccode := 05H; cstr := '1.50'; EXIT
    '7': cdata := '7'; ccode := 06H; cstr := '2.00'; EXIT
    '8': cdata := '8'; ccode := 07H; cstr := '2.50'; EXIT
    esc: EXIT
    ELSE Alarm(3500,100,2);
    END;
   END; (* LOOP *)
   setupcode := ccode+fcode;
   Send(setupcode,1);
   Close(amp_window);
END SetAmp;

PROCEDURE SetFreq;
VAR select : CHAR; freq_window : WinType;
BEGIN
   freq_window := Open(WinDef(47,14,77,24,LightGray,Black,TRUE,FALSE,
            FALSE,TRUE,SingleFrame,LightGray,Black));
   SetTitle(freq_window," Frequency Set Up ",CenterUpperTitle);
   Use(freq_window); Clear; WrLn;
   WrStr('  Selection of frequencies'); WrLn; WrLn;
   WrStr('  ['); ChColor('1',Brown,LightGray); WrStr('] 10 KHz'); WrLn;
   WrStr('  ['); ChColor('2',Brown,LightGray); WrStr('] 20 KHz'); WrLn;
   WrStr('  ['); ChColor('3',Brown,LightGray); WrStr('] 40 KHz'); WrLn;
   WrStr('  ['); ChColor('4',Brown,LightGray); WrStr('] 80 KHz');
   LOOP
    GotoXY(3,9); WrStr('Please enter... ');
    select := RdKey();
    CASE select OF
    '1': fdata := '1'; fcode := 08H; fstr := '10'; EXIT
    '2': fdata := '2'; fcode := 10H; fstr := '20'; EXIT
    '3': fdata := '3'; fcode := 00H; fstr := '40'; EXIT
    '4': fdata := '4'; fcode := 18H; fstr := '80'; EXIT
    esc: EXIT
```

```
    ELSE Alarm(3500,100,2);
    END;
  END;
  setupcode := ccode+fcode;
  Send(setupcode,1);
  Close(freq_window);
END SetFreq;

PROCEDURE SendSignal(sendcode : BYTE);
BEGIN
  ClrRxCount; Send(sendcode,1)
END SendSignal;

PROCEDURE Increment (VAR input : PtrType);
BEGIN
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,Yellow,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,Yellow,LightGray)
  END;
  CASE input OF
  |amp : input := freq; Yposition := 3;
  |freq: input := amp;  Yposition := 2;
  END;
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,LightRed,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,LightRed,LightGray)
  END;
END Increment;

PROCEDURE Decrement (VAR input : PtrType);
BEGIN
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,Yellow,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,Yellow,LightGray)
  END;
  CASE input OF
  |amp : input := freq; Yposition := 3;
  |freq: input := amp;  Yposition := 2;
  END;
  IF Yposition = 2 THEN
    GotoXY(20,Yposition); ChColor(cstr,LightRed,LightGray)
  ELSE
    GotoXY(22,Yposition); ChColor(fstr,LightRed,LightGray)
  END;
END Decrement;

PROCEDURE SetUp;
LABEL Alter;
VAR ans,select : CHAR;
  exit_window : WinType;
BEGIN
  exit_window := Open(WinDef(47,22,63,24,LightGray,Black,TRUE,FALSE,
                 FALSE,TRUE,SingleFrame,LightGray,Black));
  WrStr(' <'); ChColor('esc',LightRed,LightGray); WrStr('> to exit');
  setup_window := Open(WinDef(0,15,35,20,LightGray,Black,FALSE,FALSE,
                 FALSE,TRUE,SingleFrame,LightGray,Black));
  SetTitle(setup_window,' Measurement(s) Set Up ',CenterUpperTitle);
  Use(setup_window); CursorOn;
  GotoXY(8,2); WrStr('Amplitude = ');
  ChColor(cstr,Yellow,LightGray); GotoXY(25,2); WrStr('mA');
  GotoXY(8,3); WrStr('Frequency =  ');
  ChColor(fstr,Yellow,LightGray); GotoXY(25,3); WrStr('kHz');
  parameter := freq;
Alter:
  PutOnTop(setup_window); CursorOn;
  LOOP
    CASE parameter OF
    |amp : GotoXY(20,2); TextColor(LightRed); WrStr(cstr);
        SetAmp; GotoXY(20,2); WrStr(cstr); Yposition := 2;
    |freq: GotoXY(22,3); TextColor(LightRed); WrStr(fstr);
        SetFreq; GotoXY(22,3); WrStr(fstr); Yposition := 3;
    END; (* of CASE *)
  PutOnTop(code_window); GotoXY(31,1); ClrEol;
  TextColor(Yellow); WrShtHex(setupcode,1);
  PutOnTop(setup_window); CursorOn;
  select := RdKey();
  REPEAT
   IF select = CHR(0) THEN
    select := RdKey();
    CASE select OF
    |CHR(80) : Increment(parameter); select := RdKey();
    |CHR(72) : Decrement(parameter); select := RdKey();
    END; (* of CASE *)
   ELSIF select = esc THEN EXIT;
   ELSIF select = cr THEN GOTO Alter;
   ELSE select := RdKey();
   END;
  UNTIL FALSE;
  END; (* of LOOP *)
  Close(setup_window); Close(code_window); Close(exit_window);
END SetUp;

PROCEDURE CurrentSetUp;
BEGIN
  InitData; SetUp;
  initfile := FIO.Create('initial.data');
  FIO.WrChar(initfile,cdata); FIO.WrChar(initfile,fdata);
  FIO.Close(initfile);
END CurrentSetUp;

END EITSetUp.
```

```
IMPLEMENTATION MODULE EITUtil;

FROM IO IMPORT
     WrCard,Wrlst,WrStr,WrLn,RdKey;
FROM Lib IMPORT
     Delay,Sound,NoSound;
FROM Window IMPORT
     WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,
     SetTitle,Close,Clear,CursorOn,CursorOff,TextColor,Use,
     CenterUpperTitle,PutOnTop;
IMPORT EITGraph,rs;
CONST esc = 33C;

PROCEDURE Alarm(freq,period,times : CARDINAL);
VAR i : CARDINAL;
BEGIN
   FOR i := 1 TO times DO
   Sound(freq); Delay(50); NoSound; Delay (period);
   END
END Alarm;

PROCEDURE ChColor(temp : ARRAY OF CHAR; new,old : Color);
BEGIN
   TextColor(new); WrStr(temp); TextColor(old);
END ChColor;

PROCEDURE Psize (x,y,lx,ly,color : CARDINAL);
VAR x2,y2,i: CARDINAL;
BEGIN
   x2 := x + lx; y2 := y + ly;
   FOR i := y TO y2 DO EITGraph.HLine(x,i,x2,color) END
END Psize;

PROCEDURE ClrRxCount;
VAR clrbuffer : BYTE;
BEGIN
   LOOP
   IF rs.RxCount() # 0 THEN rs.Receive(clrbuffer,1) ELSE EXIT END
   END
END ClrRxCount;

PROCEDURE ClrKeyBuffer;
VAR clrkeyb : CHAR; intrpt_window : WinType;
BEGIN
   intrpt_window := Open(WinDef(27,17,53,19,LightGray,LightRed,FALSE,FALSE,
                    FALSE,TRUE,DoubleFrame,LightGray,LightRed));
   WrStr(' press <esc> to continue'); Alarm(3500,100,2);
   REPEAT
      clrkeyb := RdKey();
   UNTIL clrkeyb = esc;
   Close(intrpt_window);
END ClrKeyBuffer ;

PROCEDURE WrDec(indata : LONGREAL; decimal : INTEGER);
VAR residual : LONGREAL; i,integer1 : INTEGER;
   data : ARRAY[1..16] OF INTEGER;
BEGIN
   integer1 := TRUNC(indata);
   residual := ABS(indata-LONGREAL(integer1));
   FOR i := 1 TO decimal DO
     residual := residual*10.;
     data[i] := TRUNC(residual);
     residual := residual-LONGREAL(data[i]);
   END;
   IF (indata < 0.) AND (indata > -1.) THEN WrStr('-') END;
   Wrlut(integer1,1);WrStr('.');
   FOR i := 1 TO decimal DO Wrlut(data[i],1) END;
END WrDec;

PROCEDURE Banner(portnum : CARDINAL);
BEGIN
   WrStr('COM'); WrCard(portnum,1); WrStr(' is in use   ');
   ChColor('ELECTRICAL IMPEDANCE TOMOGRAPHY SYSTEM    ',Yellow,LightCyan);
   WrStr('POW/UHW  H.T.LEUNG');
END Banner;

END EITUtil.
```

```
MODULE EITSW5;

FROM Window IMPORT
    WinType,Color,WinDef,TitleMode,SingleFrame,DoubleFrame,Open,
    SetTitle,Close,Clear,CursorOn,CursorOff,TextColor,Use,Hide,
    CenterUpperTitle,PutOnTop,GotoXY,FullScreenDef,SetFrame,ClrEol,
    TextBackground;
FROM Lib IMPORT
    Sound,NoSound,Delay,Execute;
FROM IO IMPORT
    WrCard,WrChar,WrLn,WrStr,WrShtHex,RdShtHex,RdKey,RdCard,RdChar,RdInt,
    RdItem,RdShtInt,KeyPressed,EndOfRd,WrShtCard,WrInt,WrReal,WrLngInt;
FROM EITCom IMPORT
    Install,Init,RxCount,Send,Receive,Even,None;
FROM EITUtil IMPORT
    Alarm,ChColor,ClrKeyBuffer,ClrRxCount,Banner;

IMPORT IO,FIO,MATHLIB,Storage,EITSetUp,EITDisp,EITNoise,EITDFreq,EITCGI;

(*$I+,$O+,$S+,$R+,$Z+*)
(*$S A000H*)

TYPE file_array = ARRAY [1..78] OF CHAR;
   data_array = ARRAY [1..8323] OF BYTE; (* 3 byte for verification *)
   map_array  = ARRAY [1..30976] OF CHAR;

LABEL alarm,exit,next;

CONST  null = 0C; esc = 33C; cr = 15C;
    frmdata = 832;

VAR   address       : ADDRESS;
    clr_rx_count    : BYTE;
    i,iaddress      : CARDINAL;
    portnum,currflags: CARDINAL;
    numfm,maxdata   : INTEGER;
    integer,attfactor: INTEGER;
    ansop           : CHAR;
    choice,confirm  : CHAR;
    First           : BOOLEAN;
    abort,mapexists : BOOLEAN;
    menu_window     : WinType;
    work_window     : WinType;
    update_window   : WinType;
    imagemenu_window : WinType;
    FullScreen      : WinType;
    DosScreen       : WinType;
    map             : map_array; (* image map *)
    potential       : data_array; (* peripheral voltages *)
    infile,outfile  : FIO.File;
    R_data,I_data   : ARRAY [1..2080] OF REAL;  (* 10 frames *)

PROCEDURE LocalAlarm(filename : ARRAY OF CHAR);
BEGIN
   WrLn; Alarm(3500,100,2);
   ChColor(' ERROR: ',Red,Yellow); WrStr(filename);
   ChColor(' does not exist! ',Red,White); WrLn;
END LocalAlarm;

PROCEDURE MakeMap;
VAR p,q : CARDINAL; filename : file_array;
BEGIN
   Clear; WrLn; WrStr('  Data are being read from APTn.DAT');
   WrLn; WrStr('  File(s)');
   LOOP
   FOR p := 0 TO 15 DO
     CASE p OF
     0 : filename := 'apt01.dat'|  1 : filename := 'apt02.dat'|
     2 : filename := 'apt03.dat'|  3 : filename := 'apt04.dat'|
     4 : filename := 'apt05.dat'|  5 : filename := 'apt06.dat'|
     6 : filename := 'apt07.dat'|  7 : filename := 'apt08.dat'|
     8 : filename := 'apt09.dat'|  9 : filename := 'apt10.dat'|
     10: filename := 'apt11.dat'|  11: filename := 'apt12.dat'|
     12: filename := 'apt13.dat'|  13: filename := 'apt14.dat'|
     14: filename := 'apt15.dat'|  15: filename := 'apt16.dat'
     END; WrCard(p+1,3);
     IF NOT FIO.Exists(filename) THEN
       LocalAlarm(filename); abort := TRUE; EXIT
     ELSE
       infile := FIO.Open(filename);
       FOR q := 1 TO 1936 DO map[p*1936+q] := FIO.RdChar(infile) END;
       FIO.Close(infile);
     END
   END; Alarm(4000,100,1); EXIT
   END
END MakeMap;

(*  *  *  *  *  *  *  *  *  *  *)

(*$S+*)

PROCEDURE RPImaging;

CONST slimit = 1.E-10;

LABEL Check1,Check2,Check3,Check4,
    Backprojection,Manipulation,SelectReconstruction,EndRPImaging;

VAR first,imageflag,Rflag,Iflag,ResiFlag : BOOLEAN;
   it,p,q : INTEGER;
   drive,code,i,j,k,l,m,n,attfactor : INTEGER;
   x,ans,imageoption,esckey,realcon : CHAR;
   Rv1,Rv2,Iv1,Iv2 : REAL;
   npix,npixreal,sum : REAL;
   vn : LONGREAL;
   magnitude1,magnitude2,phase1,phase2,tmps : LONGREAL;
```

```
  spixel : ARRAY [1..16] OF INTEGER;
  sumay,corr : ARRAY [1..16] OF REAL;
  signew : ARRAY[1..1936] OF REAL;
  vlog,Real : ARRAY [1..256] OF REAL;
  Magnitude,Phase : ARRAY [1..256] OF REAL;
  z,a,m : ARRAY [1..1936] OF REAL;
  outname,filename : file_array;
  Rfile1,Rfile2,Ifile1,Ifile2 : file_array;
  infile1,infile2,infile3,infile4,outfile : FIO.File;

BEGIN
  FOR i := 1 TO 256 DO vlog[i] := 0. END;
  FOR i := 1 TO 78 DO Rfile1[i] := null; Ifile1[i] := null;
            Rfile2[i] := null; Ifile2[i] := null;
  END;
  Clear; GotoXY(3,2);
  ChColor('Image Reconstruction Routine',Yellow,White);
  GotoXY(3,4); WrStr('Reference File [Real]: ');
  GotoXY(26,4); RdItem(Rfile1);
Check1:
  IF NOT FIO.Exists(Rfile1) THEN
    GotoXY(10,10); ClrEol;
    GotoXY(3,9); LocalAlarm(Rfile1); WrLn;
    ChColor('  <esc> to abort the process',LightRed,White);
    WrStr('  Another go ? [Y]'); ans := RdKey();
    IF ans = esc THEN GOTO EndRPImaging;
    ELSE GotoXY(26,4); ClrEol; RdItem(Rfile1); GOTO Check1 END
  ELSE GotoXY(26,4); ChColor(Rfile1,Yellow,White)
  END;
  GotoXY(13,5); WrStr('[Imaginary]: ');
  Ifile1 := Rfile1;
  LOOP
  FOR i := 1 TO 78 DO
    IF Rfile1[i] = '.' THEN
      Ifile1[i+1] := 'I';
      Ifile1[i+2] := Rfile1[i+2];
      Ifile1[i+3] := Rfile1[i+3];
      Ifile1[i+4] := 0C;
      GotoXY(26,5); WrStr(Ifile1); esckey := RdKey();
      IF esckey = esc THEN GotoXY(26,5); ClrEol; RdItem(Ifile1) END;
      EXIT
    ELSIF i = 77 THEN GotoXY(26,5); ClrEol; RdItem(Ifile1); EXIT
    END;
  END;
  END;
Check2:
  IF NOT FIO.Exists(Ifile1) THEN
    GotoXY(10,10); ClrEol;
    GotoXY(3,9); LocalAlarm(Ifile1); WrLn;
    ChColor('  <esc> to abort the process',LightRed,White);
    WrStr('  Another go ? [Y]'); ans := RdKey();
    IF ans = esc THEN GOTO EndRPImaging;
    ELSE GotoXY(26,5); ClrEol; RdItem(Ifile1); GOTO Check2 END
  ELSE GotoXY(26,5); ChColor(Ifile1,Yellow,White)
  END;
  GotoXY(7,6); WrStr('Image File [Real]: ');
  GotoXY(26,6); RdItem(Rfile2);
Check3:
  IF NOT FIO.Exists(Rfile2) THEN
    GotoXY(10,10); ClrEol;
    GotoXY(3,9); LocalAlarm(Rfile2); WrLn;
    ChColor('  <esc> to abort the process',LightRed,White);
    WrStr('  Another go ? [Y]'); ans := RdKey();
    IF ans = esc THEN GOTO EndRPImaging;
    ELSE GotoXY(26,6); ClrEol; RdItem(Rfile2); GOTO Check3 END
  ELSE GotoXY(26,6); ChColor(Rfile2,Yellow,White)
  END;
  GotoXY(13,7); WrStr('[Imaginary]: ');
  Ifile2 := Rfile2;
  LOOP
  FOR i := 1 TO 78 DO
    IF Rfile2[i] = '.' THEN
      Ifile2[i+1] := 'I';
      Ifile2[i+2] := Rfile2[i+2];
      Ifile2[i+3] := Rfile2[i+3];
      Ifile2[i+4] := 0C;
      GotoXY(26,7); WrStr(Ifile2); esckey := RdKey();
      IF esckey = esc THEN GotoXY(26,7); ClrEol; RdItem(Ifile2) END;
      EXIT
    ELSIF i = 77 THEN GotoXY(26,7); ClrEol; RdItem(Ifile2); EXIT
    END;
  END;
  END;
Check4:
  IF NOT FIO.Exists(Ifile2) THEN
    GotoXY(10,10); ClrEol;
    GotoXY(3,9); LocalAlarm(Ifile2); WrLn;
    ChColor('  <esc> to abort the process',LightRed,White);
    WrStr('  Another go ? [Y]'); ans := RdKey();
    IF ans = esc THEN GOTO EndRPImaging;
    ELSE GotoXY(26,7); ClrEol; RdItem(Ifile2); GOTO Check4 END
  ELSE GotoXY(26,7); ChColor(Ifile2,Yellow,White)
  END;

  GotoXY(1,12); ClrEol; GotoXY(1,10); ClrEol;
  Rflag := FALSE; Iflag := FALSE; RealFlag := FALSE;
  infile1 := FIO.Open(Rfile1); infile2 := FIO.Open(Ifile1);
  infile3 := FIO.Open(Rfile2); infile4 := FIO.Open(Ifile2);
  WrStr('  Reading real and imaginary data from files.  Please wait... ');
  FOR i := 1 TO 256 DO
    Rv1 := FIO.RdReal(infile1); attfactor := FIO.RdInt(infile1);
    Iv1 := FIO.RdReal(infile2); attfactor := FIO.RdInt(infile2);
    Rv2 := FIO.RdReal(infile3); attfactor := FIO.RdInt(infile3);
    Iv2 := FIO.RdReal(infile4); attfactor := FIO.RdInt(infile4);

    IF Rv2=0. THEN
```

```
      GotoXY(1,10); ClrEol;
      ChColor('  ERROR: Arithmatic overflows!',LightRed,White);
      Alarm(3500,100,2); Delay(1000); GOTO EndRPImaging;
    ELSIF (Rv1/Rv2)<=0. THEN
      GotoXY(1,11); ClrEol; GotoXY(1,12); ClrEol; GotoXY(1,10); ClrEol;
      ChColor('  ERROR: MATHLIB.Log()!',LightRed,White);
      Alarm(3500,100,2); Delay(1000); GotoXY(3,11);
      WrStr('Error detected!  Real Part reconstruction may not be possible.');
      GotoXY(3,12);  WrStr('Do you want to continue? [Y]');
      realcon := RdKey();
      IF (realcon='n') OR (realcon='N') THEN GOTO EndRPImaging
      ELSE Real[i] := 0.; WrStr('  Please wait... '); END;
    ELSE
      Real[i] := REAL(MATHLIB.Log(LONGREAL(Rv1/Rv2)))
    END;

    magnitude1 := MATHLIB.Sqrt(LONGREAL(Rv1*Rv1+Iv1*Iv1));
    magnitude2 := MATHLIB.Sqrt(LONGREAL(Rv2*Rv2+Iv2*Iv2));
    IF Rv1 = 0. THEN
      GotoXY(3,12); ClrEol;
      WrStr('Warning: Real part is equal to zero!');
      phase1 := MATHLIB.ACos(0.);
    ELSE
      phase1 := MATHLIB.ATan(LONGREAL(Iv1/Rv1));
    END;
    IF Rv2 = 0. THEN
      GotoXY(3,12); ClrEol;
      WrStr('Warning: Real part is equal to zero!');
      phase2 := MATHLIB.ACos(0.);
    ELSE
      phase2 := MATHLIB.ATan(LONGREAL(Iv2/Rv2));
    END;

    Phase[i] := REAL(phase1-phase2);
    IF magnitude2=0. THEN
      GotoXY(1,10); ClrEol;
      ChColor('  ERROR: Arithmatic overflows!',LightRed,White);
      Alarm(3500,100,2); Delay(1000); GOTO EndRPImaging;
    END;
    vn := LONGREAL(magnitude1/magnitude2);
    IF vn<=0. THEN
      GotoXY(1,10); ClrEol;
      ChColor('  ERROR: MATHLIB.Log()!',LightRed,White);
      Alarm(3500,100,2); Delay(1000); GOTO EndRPImaging;
    END;
    Magnitude[i] := REAL(MATHLIB.Log(vn))
  END;
  FIO.Close(infile1); FIO.Close(infile2);
  FIO.Close(infile3); FIO.Close(infile4);
  WrLn;

(* added for testing only *)
(*
  outfile := FIO.Create('Phase.txt');
  FOR i := 1 TO 256 DO FIO.WrReal(outfile,Phase[i],5,10);
  FIO.WrLn(outfile); END; FIO.Close(outfile);
  outfile := FIO.Create('Mag.txt');
  FOR i := 1 TO 256 DO FIO.WrReal(outfile,Magnitude[i],5,10);
  FIO.WrLn(outfile); END; FIO.Close(outfile);
*)
(* end *)

  FOR drive := 1 TO 16 DO
    k := drive-2;
    IF k < 1 THEN k := k+16 END;
    l := drive-1;
    IF l < 1 THEN l := l+16 END;
    m := drive+1;
    IF m > 16 THEN m := m-16 END;
    n := drive+2;
    IF n > 16 THEN n := n-16 END;
    j := 16*(drive-1);
    Magnitude[j+l] := 0.75*Magnitude[j+k]+0.25*Magnitude[j+n];
    Magnitude[j+m] := 0.25*Magnitude[j+k]+0.75*Magnitude[j+n];
    Phase[j+l] := 0.75*Phase[j+k]+0.25*Phase[j+n];
    Phase[j+m] := 0.25*Phase[j+k]+0.75*Phase[j+n]
  END; (* FOR drive END *)

(* == Options for image reconstruction == *)
SelectReconstruction:
  Alarm(4000,100,1);
  REPEAT
  Clear; GotoXY(3,2);
  WrStr('Selection of image to be reconstructed:');
  GotoXY(3,4); ChColor('1',Brown,Yellow); WrStr('.  Magnitude');
  GotoXY(3,5); ChColor('2',Brown,Yellow); WrStr('.  Phase');
  GotoXY(3,6); ChColor('3',Brown,Yellow); WrStr('.  Alpha');
  GotoXY(3,7); ChColor('4',Brown,Yellow); WrStr('.  Beta');
  GotoXY(3,8); ChColor('5',Brown,Yellow); WrStr('.  Real Part');
  GotoXY(3,10); ChColor('Press <',White,LightRed);
  ChColor('esc',LightRed,White);
  WrStr('> to quit image reconstruction routines');
  GotoXY(3,12); WrStr('Please enter ... ');
  LOOP
   imageoption := RdKey();
   IF imageoption = esc THEN imageflag := FALSE; EXIT END;
   IF (imageoption > '0') AND (imageoption < '6')
   THEN imageflag := TRUE; EXIT
   ELSE Alarm(3500,100,2) END;
  END;

  IF imageflag = TRUE THEN
   Clear; GotoXY(3,2); WrStr('Image parameter : ');
   CASE imageoption OF
   |'1' : ChColor('Magnitude',Yellow,White);
       IF Rflag = FALSE THEN
```

```
        FOR i := 1 TO 256 DO vlog[i] := Magnitude[i] END;
      ELSE GOTO Manipulation
      END
  |'2' : ChColor('Phase',Yellow,White);
      IF Iflag = FALSE THEN
        FOR i := 1 TO 256 DO vlog[i] := Phase[i] END;
      ELSE GOTO Manipulation
      END
  |'3' : ChColor('Alpha',Yellow,White);
      IF (Rflag = TRUE) AND (Iflag = TRUE) THEN GOTO Manipulation
      ELSIF Iflag = FALSE THEN
          FOR i := 1 TO 256 DO vlog[i] := Phase[i] END;
          first := TRUE;
      ELSIF Rflag = FALSE THEN
          FOR i := 1 TO 256 DO vlog[i] := Magnitude[i] END;
          first := FALSE;
      END;
  |'4' : ChColor('Beta',Yellow,White);
      IF (Rflag = TRUE) AND (Iflag = TRUE) THEN GOTO Manipulation
      ELSIF Iflag = FALSE THEN
          FOR i := 1 TO 256 DO vlog[i] := Phase[i] END;
          first := TRUE;
      ELSIF Rflag = FALSE THEN
          FOR i := 1 TO 256 DO vlog[i] := Magnitude[i] END;
          first := FALSE;
      END;
  |'5' : ChColor('Real Part',Yellow,White);
      IF RealFlag = FALSE THEN
        RealFlag := TRUE;
        FOR i := 1 TO 256 DO vlog[i] := Real[i] END;
      ELSE GOTO Manipulation
      END;
  END; (* CASE *)

Backprojection:
  WrLn; WrLn;
  WrStr('  Backprojection routine starts'); WrLn;
  WrStr('  Iterate');
  FOR i := 1 TO 1936 DO z[i] := 0. END;
  FOR it := 1 TO 3 DO WrCard(it,3);
  FOR i := 1 TO 16 DO corr[i] := 0. END;
  FOR drive := 1 TO 16 DO
    FOR i := 1 TO 16 DO npixel[i] := 0; sumay[i] := 0. END;
    n := 1936*(drive-1);
    FOR i := 1 TO 1936 DO
      x := map[n+i];
      IF x # '0' THEN
        code := ORD(x)-48;
        npixel[code] := npixel[code]+1;
        sumay[code] := sumay[code]+z[i];
      END;
    END;
    m := 16*(drive-1);
    FOR i := 1 TO 16 DO
      npixreal := REAL(npixel[i]);
      IF npixel[i] # 0 THEN
        corr[i] := vlog[m+i]-sumay[i]/npixreal
      END
    END;
    FOR i := 1 TO 1936 DO
      x := map[n+i];
      IF x # '0' THEN
        code := ORD(x)-48;
        z[i] := z[i]+corr[code]
      END
    END
  END (* drive *)
  END; (* it *)

(*  SigFilter *)

  FOR p := 1 TO 1936 DO signew[p] := z[p] END;
  FOR q := 1 TO 3 DO
  FOR p := 1 TO 1936 DO
    IF map[p] # '0' THEN
      sum := z[p]; npix := 1.;
      IF map[p-1] # '0' THEN sum := sum+z[p-1]; npix := npix+1. END;
      IF map[p+1] # '0' THEN sum := sum+z[p+1]; npix := npix+1. END;
      IF map[p+44] # '0' THEN sum := sum+z[p+44]; npix := npix+1. END;
      IF map[p-44] # '0' THEN sum := sum+z[p-44]; npix := npix+1. END;
      IF map[p-45] # '0' THEN sum := sum+z[p-45]; npix := npix+1. END;
      IF map[p-43] # '0' THEN sum := sum+z[p-43]; npix := npix+1. END;
      IF map[p+45] # '0' THEN sum := sum+z[p+45]; npix := npix+1. END;
      IF map[p+43] # '0' THEN sum := sum+z[p+43]; npix := npix+1. END;
      signew[p] := sum/npix
    END
  END; FOR p := 1 TO 1936 DO z[p] := signew[p] END
  END;

  IF imageoption = '1' THEN
    FOR i := 1 TO 1936 DO s[i] := z[i] END; Rflag := TRUE;
  ELSIF imageoption = '2' THEN
    FOR i := 1 TO 1936 DO
      ss[i] := z[i]
    END; Iflag := TRUE;
  END;

  IF ((imageoption = '3') OR (imageoption = '4')) THEN
    IF first = TRUE THEN
      FOR i := 1 TO 1936 DO ss[i] := z[i] END; Iflag := TRUE;
      FOR i := 1 TO 256 DO vlog[i] := Magnitude[i] END;
      first := FALSE; GOTO Backprojection
    ELSE
      FOR i := 1 TO 1936 DO s[i] := z[i] END; Rflag := TRUE;
    END
  END;
```

```
Manipulation:
   CASE imageoption OF
  |'3' : LOOP
          FOR i := 1 TO 1936 DO
            IF map[i] # '0' THEN
              tmps := MATHLIB.Cos(LONGREAL(ss[i]));
              IF tmps <= 0. THEN
                ChColor(' ERROR: MATHLIB.Log()!',LightRed,White);
                Alarm(3500,100,2); Delay(1000);
                GOTO SelectReconstruction;
              ELSE
                z[i] := s[i]+REAL(MATHLIB.Log(tmps))
              END;
            ELSE z[i] := 0.;
            END;
          END; EXIT;
        END;
  |'4' : FOR i := 1 TO 1936 DO
          IF map[i] # '0' THEN
          tmps := MATHLIB.Sin(LONGREAL(ss[i]));
          IF tmps<=slimit THEN tmps := slimit END;
            z[i] := s[i]+REAL(MATHLIB.Log(tmps))
          ELSE z[i] := 0.
          END;
        END;
   END;

(* create image file *)

   WrLn; WrLn; WrStr(' Creating image file: ');
   CASE imageoption OF
   |'1' : outname := 'IMAGE.s';
        FOR i := 1 TO 1936 DO z[i] := s[i] END;
   |'2' : outname := 'IMAGE.ss';
        FOR i := 1 TO 1936 DO z[i] := ss[i] END;
   |'3' : outname := 'IMAGE.a';
   |'4' : outname := 'IMAGE.b';
   |'5' : outname := 'IMAGE.r';
   END;
   ChColor(outname,Yellow,White); WrStr(' Please wait... ');
   outfile := FIO.Create(outname);
   FOR i := 1 TO 1936 DO
     IF map[i] # '0' THEN
       FIO.WrReal(outfile,z[i],5,8);
     ELSE
       FIO.WrReal(outfile,0.,5,8);
     END;
     FIO.WrLn(outfile)
   END; FIO.Close(outfile);
   WrLn; Alarm(4000,100,1);

   END; (* IF imageflage *)
   UNTIL imageflag = FALSE;

EndRPImaging: Clear
END RPImaging;

(*  *  *  *  *  *  *  *  *  *  *)

PROCEDURE SendSignal(sendcode : BYTE);
BEGIN
   ClrRxCount; Send(sendcode,1)
END SendSignal;

PROCEDURE ChnFrmNum;
VAR setcodeflag : CHAR; setflag : BYTE;
BEGIN
   WrStr(' Set attenuation factor codes? [N] '); setcodeflag := RdKey();
   IF (setcodeflag = 'Y') OR (setcodeflag = 'y') THEN
     ChColor(setcodeflag,Yellow,White);
     setflag := 0F0H; SendSignal(setflag); numfm := 1
   ELSE
     setflag := 0FH; SendSignal(setflag);
     LOOP
       GotoXY(3,1); ClrEol;
       WrStr('Number of frames [max.=10] : ');
       numfm := RdInt();
       IF (numfm > 0) AND (numfm < 11) THEN EXIT
       ELSE Alarm(3500,100,2); END;
     END; SendSignal(BYTE(numfm));
   END;
   maxdata := numfm*frmdata;
END ChnFrmNum;

PROCEDURE InitBuffer;
VAR i : CARDINAL;
BEGIN
   WrStr(' Initialisation of the data buffer is proceeding. ');
   WrStr('Please wait...');
   FOR i := 1 TO 8320 DO potential[i] := 00H END; WrLn;
   ChColor(' Initialisation completed.',Yellow,White);
   Alarm(4000,100,1); Delay(2000)
END InitBuffer;

PROCEDURE WaitAcknowledge;
VAR rdbuffer : BYTE; status_window : WinType;
BEGIN
   status_window := Open(WinDef(0,22,79,24,White,Black,FALSE,FALSE,
                    FALSE,TRUE,SingleFrame,White,Black));
   SetTitle(status_window," Voltage Measurements Status ",CenterUpperTitle);
   WrStr(' Waiting for an acknowledgement from the EIT system. ');
   WrStr('Please wait...');
   LOOP
    IF KeyPressed() THEN
      abort := TRUE; PutOnTop(status_window);
```

```
      SetFrame(status_window,SingleFrame,LightRed,Black); Clear;
      TextColor(LightGray); WrStr(' Communications have been interrupted!');
      ClrKeyBuffer; EXIT
    END;
    IF RxCount() # 0 THEN
      Receive(rdbuffer,1);
      IF rdbuffer = BYTE(sumfm) THEN Clear; EXIT ELSE
        Alarm(3500,100,2); Clear;
        WrStr(' ERROR in RS232 link.');
        Delay(2000); EXIT
      END;
    END;
  END;
  Close(status_window);
END WaitAcknowledge;

PROCEDURE FshMeasurement;
VAR rdbuffer : BYTE; pass,final : INTEGER; status_window : WinType;
BEGIN
  LOOP
    IF KeyPressed() THEN
      status_window := Open(WinDef(0,22,79,24,White,Black,FALSE,FALSE,
                            FALSE,TRUE,SingleFrame,White,Black));
      SetTitle(status_window," Voltage Measurements Status ",
      CenterUpperTitle);
      abort := TRUE; PutOnTop(status_window);
      SetFrame(status_window,SingleFrame,LightRed,Black); Clear;
      TextColor(LightGray); WrStr(' Communications have been interrupted!');
      ClrKeyBuffer; Close(status_window); EXIT
    END;
    IF RxCount() # 0 THEN
      Receive(rdbuffer,1);
      IF rdbuffer = BYTE(255) THEN Clear;
        WrStr(' Measurements have been completed.');
        Alarm(4000,100,1); EXIT
      ELSE
        Clear; WrStr(' ERROR in RS232 link');
        Alarm(3500,100,2); EXIT
      END;
    END;
  END; Delay(1000); Clear
END FshMeasurement;

PROCEDURE ComPort;
VAR comport : CHAR;
BEGIN
  FullScreen := Open(FullScreenDef);
  LOOP
    WrStr('Please enter COM-port to be used (1 or 2) ');
    comport := RdKey();
    CASE comport OF
    |'1' : portnum := 1; EXIT
    |'2' : portnum := 2; EXIT
    ELSE Alarm(3500,100,2); Clear
    END
  END; Clear; Banner(portnum);
  TextColor(White); First := TRUE;
  Install(portnum,First);
  Init(9600,8,None,TRUE,FALSE);
END ComPort;

PROCEDURE InitData;
VAR cdata,fdata : CHAR; fstr,cstr : file_array;
  fcode,ccode,setupcode : BYTE;
BEGIN
  SetFrame(update_window,DoubleFrame,LightGray,Black);
  PutOnTop(update_window); Clear; WrLn;
  infile := FIO.Open('Initial.dat');
  cdata := FIO.RdChar(infile);
  fdata := FIO.RdChar(infile);
  WrStr(' Frequency = ');
  CASE fdata OF
  |'1' : fcode := 08H; fstr := '10'
  |'2' : fcode := 10H; fstr := '20'
  |'3' : fcode := 00H; fstr := '40'
  |'4' : fcode := 18H; fstr := '80'
  END;
  ChColor(fstr,Cyan,LightGray);
  WrStr(' kHz'); WrLn; WrStr(' Amplitude = ');
  CASE cdata OF
  |'1' : ccode := 00H; cstr := '0.00'
  |'2' : ccode := 01H; cstr := '0.00'
  |'3' : ccode := 02H; cstr := '0.00'
  |'4' : ccode := 03H; cstr := '0.00'
  |'5' : ccode := 04H; cstr := '1.00'
  |'6' : ccode := 05H; cstr := '1.50'
  |'7' : ccode := 06H; cstr := '2.00'
  |'8' : ccode := 07H; cstr := '2.50'
  END;
  ChColor(cstr,Cyan,LightGray);
  WrStr(' mA'); FIO.Close(infile);
  SendSignal(01H);
  setupcode := ccode+fcode; Send(setupcode,1);
  SendSignal(0FFH);
END InitData;

PROCEDURE Menu;
BEGIN
  menu_window := Open(WinDef(0,1,79,11,White,Blue,FALSE,FALSE,FALSE,
                 TRUE,DoubleFrame,White,Blue));
  SetTitle(menu_window," EIT Main Menu ",CenterUpperTitle); Clear;
  WrStr(' Selection of possible operations:'); TextColor(LightGray);
  GotoXY(3,3); WrStr('<'); ChColor('F1',Brown,LightGray); WrStr('>');
  ChColor(' Start Measurements',Yellow,LightGray);
  GotoXY(3,4); WrStr('<'); ChColor('F2',Brown,LightGray); WrStr('>');
  ChColor(' Start Transmission',Yellow,LightGray);
```

```
    GotoXY(3,5); WrStr('<'); ChColor('F3',Brown,LightGray); WrStr('>');
    ChColor(' Display Received Data',Yellow,LightGray);
    GotoXY(3,6); WrStr('<'); ChColor('F4',Brown,LightGray); WrStr('>');
    ChColor(' Output Data to File(s)',Yellow,LightGray);
    GotoXY(3,7); WrStr('<'); ChColor('F5',Brown,LightGray); WrStr('>');
    ChColor(' Initialise Data Buffer',Yellow,LightGray);
    GotoXY(3,8); WrStr('<'); ChColor('F6',Brown,LightGray); WrStr('>');
    ChColor(' Change Stimulus Settings',Yellow,LightGray);
    GotoXY(3,9); WrStr('<'); ChColor('F7',Brown,LightGray); WrStr('>');
    ChColor(' Perform Graphics Operations',Yellow,LightGray);
    GotoXY(41,3); WrStr('<'); ChColor('F8',Brown,LightGray); WrStr('>');
    ChColor(' System Performance Analyses',Yellow,LightGray);
    GotoXY(41,4); WrStr('<'); ChColor('F9',Brown,LightGray); WrStr('>');
    ChColor(' Dual Frequency Measurements',Yellow,LightGray);
    GotoXY(40,5); WrStr('<'); ChColor('F10',Brown,LightGray); WrStr('>');
    ChColor(' Cardiac Gated Imaging',Yellow,LightGray);
    GotoXY(40,8); ChColor('Press <',LightRed,White);
    ChColor('Alt-D',White,LightRed); WrStr ('> to push to DOS');
    GotoXY(40,9); ChColor('Press <',LightRed,White);
    ChColor('Alt-X',White,LightRed); WrStr ('> to quit EIT programme');
    update_window := Open(WinDef(57,17,79,20,LightGray,Black,FALSE,FALSE,
                     FALSE,TRUE,DoubleFrame,LightGray,DarkGray));
    SetTitle(update_window," Stimulus Settings ",CenterUpperTitle);
    Use(update_window);
  END Menu;

  (*$I+*)
  PROCEDURE ReceiveData;
  VAR i,j,cntprt : INTEGER; data_window,status_window : WinType;
  BEGIN
    status_window := Open(WinDef(0,22,79,24,White,Black,FALSE,FALSE,
                     FALSE,TRUE,SingleFrame,White,Black));
    SetTitle(status_window," Transmission Status ",CenterUpperTitle);
    WrStr(' PC is now receiving data from the EIT system.');
    WrStr(' Please wait...');
    data_window := Open(WinDef(8,17,23,19,White,Black,FALSE,FALSE,
                     FALSE,TRUE,DoubleFrame,White,Black));
    SetTitle (data_window," Byte Count ",CenterUpperTitle);
    Use(data_window);
    i := maxdata; j := 1;
    LOOP
     IF RxCount() # 0 THEN
       Receive(potential[i],1);
       cntprt := maxdata-i;
       IF cntprt = j*100 THEN WrLn; WrInt(cntprt,13); j := j+1 END;
       i := i-1;
     END;
     IF KeyPressed() THEN
       abort := TRUE;
       LOOP
        IF RxCount() # 0 THEN
          Receive(potential[i],1)
        ELSE EXIT
        END
       END; PutOnTop(status_window);
       SetFrame(status_window,SingleFrame,LightRed,Black); Clear;
       TextColor(LightGray); WrStr(' Transmission has been interrupted!');
       ClrKeyBuffer; EXIT
     END;
     IF i = 0 THEN WrLn; WrInt(maxdata,13); Delay(2000); EXIT END;
    END;
    Close(data_window); Close(status_window);
  END ReceiveData;

  PROCEDURE ReceiveVerifyData;
  VAR i : INTEGER;
  BEGIN
    i := maxdata+1;
    LOOP
     IF RxCount() # 0 THEN
       Receive(potential[i],1); i := i+1
     END;
     IF i = maxdata+4 THEN EXIT END;
    END;
  END ReceiveVerifyData;

  PROCEDURE VerifyRecvdData;
  VAR t,d,tg,tl : CARDINAL; i : INTEGER;
  BEGIN
    t := 0; tg := 0; tl := 0; Clear;
    FOR i := 1 TO maxdata+1 DO
      d := CARDINAL(potential[i]);
      LOOP
       IF (d = 0) OR (d = 128) THEN EXIT END;
       IF d > 128 THEN d := 256-d END;
       IF d = t THEN EXIT
       ELSIF d < t THEN t := t-d; tl := t;
       ELSIF d > t THEN t := d-t; tg := t;
       END; EXIT
      END;
    END;
    IF (t=CARDINAL(potential[maxdata+1])) AND
     (tg=CARDINAL(potential[maxdata+2])) AND
     (tl=CARDINAL(potential[maxdata+3]))
    THEN
     Alarm(4000,100,1);
     ChColor(' Data were successfully received.',Yellow,White)
    ELSE
     Alarm(3500,100,2);
     SetTitle(work_window," Serial Transmission ",CenterUpperTitle);
     SetFrame(work_window,SingleFrame,LightRed,Black);
     TextBackground(Black); Clear; TextColor(LightGray);
     WrStr(' Data were corrupted. Need re-transmission.');
     ClrKeyBuffer;
    END; Delay(1000); Clear;
  END VerifyRecvdData;
```

```
PROCEDURE DisplayData;
VAR i,j,ub,lb,drive,frame,pagestart,linestart : INTEGER; key,RorI : CHAR;
    disp_window,instruct_window : WinType; smallflag,satflag,warnflag : BOOLEAN;
    attfactor : ARRAY [1..4160] OF INTEGER; intdata : LONGINT;
BEGIN
    instruct_window := Open(FullScreenDef); CursorOff;
    Banner(portnum); TextColor(LightGray);
    GotoXY(3,23); WrStr('<');
    ChColor('PgUp',Cyan,LightGray); WrStr('> : Previous page');
    GotoXY(3,24); WrStr('<');
    ChColor('PgDn',Cyan,LightGray); WrStr('> : Next page');
    GotoXY(30,23); WrStr('<');
    ChColor('Home',Cyan,LightGray); WrStr('> : Page 1');
    GotoXY(30,24); WrStr('<');
    ChColor('End',Cyan,LightGray); WrStr('> : Last page of the frame');
    GotoXY(66,24); WrStr('<');
    ChColor('esc',LightRed,LightGray); WrStr('> : Exit');
    disp_window := Open(WinDef(0,1,79,21,White,Blue,FALSE,FALSE,
                   FALSE,TRUE,DoubleFrame,Yellow,Blue));
    SetTitle(disp_window," Attenuation Factors & Peripheral Potentials ",
        CenterUpperTitle);
    Use (disp_window); Clear; warnflag := FALSE;
    j := 1; abort := FALSE; RorI := 'R';
    FOR i := 1 TO maxdata BY 2 DO
      ub := INTEGER(potential[i]);
      lb := INTEGER(potential[i+1]);
      IF ub < 16 THEN attfactor[j] := 1;
      ELSIF ub < 32 THEN attfactor[j] := 2; ub := ub-16
      ELSIF ub < 48 THEN attfactor[j] := 4; ub := ub-32
      ELSIF ub < 64 THEN attfactor[j] := 8; ub := ub-48
      ELSIF ub < 80 THEN attfactor[j] := 16; ub := ub-64
      ELSIF ub < 96 THEN attfactor[j] := 32; ub := ub-80
      ELSIF ub < 112 THEN attfactor[j] := 64; ub := ub-96
      ELSIF ub < 128 THEN attfactor[j] := 128; ub := ub-112
      ELSE attfactor[j] := 0; Alarm(3500,100,2);
         ChColor('Error detected in data stream',LightRed,White); WrLn;
      END;
      intdata := LONGINT(256*ub+lb);
      (* 1V software offset = 1638 *)
      (* 0V software offset = 2048 *)
      IF RorI = 'R' THEN
        R_data[j] := REAL(intdata-2048)*5./2048.; RorI := 'I';
        IF ABS(R_data[j]) < 0.5 THEN warnflag := TRUE; END;
        IF ABS(R_data[j]) > 4.96 THEN warnflag := TRUE; END;
      ELSE
        I_data[j] := REAL(intdata-2048)*5./2048.; RorI := 'R'; j := j+1;
      END;
      IF KeyPressed() THEN key := RdKey(); abort := TRUE;
                   IF key = esc THEN i := maxdata-2 END
      END;
    END;
    pagestart := 0;
    IF abort = FALSE THEN
    LOOP
      Clear; smallflag := FALSE; satflag := FALSE;
      IF warnflag = TRUE THEN
        IF pagestart = 0 THEN Alarm(3500,100,2) END;
        GotoXY(65,21); ChColor('Data Warning',LightRed,Green);
      END;
      frame := TRUNC(REAL(pagestart)/16.+1.);
      GotoXY(13,2); ChColor('Frame ',Green,LightRed);
      WrInt(frame,1); GotoXY(23,2);
      ChColor('Page ',Green,LightRed); WrInt(pagestart+1,1);
      GotoXY(33,2); ChColor('Driving Pair : ',Green,LightRed);
      drive := pagestart-16*(frame-1)+1;
      IF drive = 1 THEN
        WrInt(16,1) ELSE WrInt(drive-1,1) END;
      ChColor(' - ',Green,LightRed); WrInt(drive,-2);
      TextColor(Cyan);
      GotoXY(33,3); WrStr('Real Part');
      GotoXY(52,3); WrStr('Imaginary Part'); TextColor(Brown);
      GotoXY(13,4); WrStr('Data'); GotoXY(23,4); WrStr('Factor');
      GotoXY(33,4); WrStr('Pot. Diff (V)');
      GotoXY(52,4); WrStr('Pot. Diff (V)');
      linestart := (pagestart)*13;
      FOR i := 1 TO 13 DO
        TextColor(Green);
        GotoXY(7,i+5); WrInt(i,2); GotoXY(68,i+5); WrInt(i,2);
        IF (i = 5) OR (i = 10) THEN
          TextColor(Cyan) ELSE TextColor(LightGray)
        END;
        GotoXY(13,i+5); WrInt(linestart+i,4);
        GotoXY(23,i+5); WrInt(attfactor[linestart+i],4);
        (* real part *)
        IF ABS(R_data[linestart+i]) < 0.5 THEN
          smallflag := TRUE;
          TextColor(LightRed);
          GotoXY(34,i+5); WrReal(R_data[linestart+i],5,8);
          IF (i = 5) OR (i = 10) THEN
          TextColor(Cyan) ELSE TextColor(LightGray)
          END;
        ELSIF ABS(R_data[linestart+i]) > 4.96 THEN
          satflag := TRUE;
          TextColor(LightRed);
          GotoXY(34,i+5); WrReal(R_data[linestart+i],5,8);
          IF (i = 5) OR (i = 10) THEN
          TextColor(Cyan) ELSE TextColor(LightGray)
          END;
        ELSE GotoXY(34,i+5); WrReal(R_data[linestart+i],5,8);
        END;
        (* imaginary part *)
        IF ABS(I_data[linestart+i]) > 4.96 THEN
          TextColor(LightGreen);
          GotoXY(53,i+5); WrReal(I_data[linestart+i],5,8);
          IF (i = 5) OR (i = 10) THEN
```

```
        TextColor(Cyan) ELSE TextColor(LightGray)
        END;
      ELSE GotoXY(53,i+5); WrReal(I_data[linestart+i],5,8);
      END;
    END; (* FOR *)
    IF smallflag = TRUE THEN
      GotoXY(3,19); ChColor('Warning: Data too small',LightRed,Green);
    END;
    IF satflag = TRUE THEN
      GotoXY(35,19); ChColor('Warning: saturations occurred!',LightRed,Green);
    END;
    key := RdKey();
    CASE key OF
    |CHAR(0) : key := RdKey();
          CASE key OF
          |CHAR(73) : IF pagestart = 0 THEN pagestart := 16*numfm-1
                ELSE pagestart := pagestart-1 END
          |CHAR(81) : IF pagestart = 16*numfm-1 THEN pagestart := 0
                ELSE pagestart := pagestart+1 END
          |CHAR(79) : pagestart := frame*16-1
          |CHAR(71) : pagestart := 0
          ELSE Alarm(3500,100,2)
          END;
    |esc   : EXIT
    ELSE Alarm(3500,100,2)
    END; (* of CASE *)
   END; (* of LOOP *)
   END; (* of IF abort *)
   Close(instruct_window); Close(disp_window)
END DisplayData;

PROCEDURE WrUnity(times : CARDINAL);
CONST unity = 1.0; decimal = 5; placement = 10; attfactor = 0;
VAR i : CARDINAL;
BEGIN
   FOR i := 1 TO times DO
     FIO.WrReal(outfile,unity,decimal,placement);
     FIO.WrInt(outfile,attfactor,placement);
     FIO.WrLn(outfile)
   END
END WrUnity;

PROCEDURE OutputFormat(avgsep : CHAR; frame : INTEGER;
            Rname,Iname : ARRAY OF CHAR);
CONST unity = 1.0; decimal = 5; placement = 10;
VAR i,j,k,x,ub,pointer : INTEGER; RorI : CHAR;
   attfactor : ARRAY [1..4160] OF INTEGER;
BEGIN
   pointer := (frame-1)*208; j := pointer+1;
   FOR i := 1 TO frmdata*2 BY 4 DO
    ub := INTEGER(potential[i]);
    IF ub < 16 THEN attfactor[j] := 1;
    ELSIF ub < 32 THEN attfactor[j] := 2;
    ELSIF ub < 48 THEN attfactor[j] := 4;
    ELSIF ub < 64 THEN attfactor[j] := 8;
    ELSIF ub < 80 THEN attfactor[j] := 16;
    ELSIF ub < 96 THEN attfactor[j] := 32;
    ELSIF ub < 112 THEN attfactor[j] := 64;
    ELSIF ub < 128 THEN attfactor[j] := 128;
    ELSE attfactor[j] := 0; Alarm(3500,100,2);
      ChColor('Error detected in data stream',LightRed,White); WrLn;
    END; j := j+1;
   END;
(*
   IF (numfm > 1) AND (avgsep = '1') THEN
     FOR i := 1 TO 208 DO attfactor[i] := 0 END
   END;
*)
   RorI := 'R';
   FOR k := 1 TO 2 DO
    IF RorI = 'R' THEN
      outfile := FIO.Create(Rname); GotoXY(16,10);
      ChColor(Rname,Yellow,White);
    ELSE
      outfile := FIO.Create(Iname); GotoXY(16,11);
      ChColor(Iname,Yellow,White);
    END;
    WrUnity(2);
    FOR i := pointer+1 TO pointer+13 DO
      IF RorI = 'R' THEN
        FIO.WrReal(outfile,R_data[i],decimal,placement)
      ELSE
        FIO.WrReal(outfile,I_data[i],decimal,placement)
      END;
      FIO.WrInt(outfile,attfactor[i],placement);
      FIO.WrLn(outfile)
    END;
    WrUnity(4);
    FOR i := pointer+14 TO pointer+26 DO
      IF RorI = 'R' THEN
        FIO.WrReal(outfile,R_data[i],decimal,placement)
      ELSE
        FIO.WrReal(outfile,I_data[i],decimal,placement)
      END;
      FIO.WrInt(outfile,attfactor[i],placement);
      FIO.WrLn(outfile)
    END;
    j := 0; x := 3;
    REPEAT
     FOR i := pointer+(13*x-j) TO pointer+x*13 DO
      IF RorI = 'R' THEN
        FIO.WrReal(outfile,R_data[i],decimal,placement)
      ELSE
        FIO.WrReal(outfile,I_data[i],decimal,placement)
      END;
       FIO.WrInt(outfile,attfactor[i],placement);
```

```
        FIO.WrLn(outfile)
      END; WrUnity(3);
      FOR i := pointer+((x-1)*13+1) TO pointer+(x*13-(j+1)) DO
       IF Rorl = 'R' THEN
         FIO.WrReal(outfile,R_data[i],decimal,placement)
       ELSE
         FIO.WrReal(outfile,I_data[i],decimal,placement)
       END;
        FIO.WrInt(outfile,attfactor[i],placement);
        FIO.WrLn(outfile)
      END; j := j+1; x := x+1;
    UNTIL j = 12;
    FOR i := pointer+183 TO pointer+195 DO
       IF Rorl = 'R' THEN
         FIO.WrReal(outfile,R_data[i],decimal,placement)
       ELSE
         FIO.WrReal(outfile,I_data[i],decimal,placement)
       END;
       FIO.WrInt(outfile,attfactor[i],placement);
       FIO.WrLn(outfile)
    END; WrUnity(4);
    FOR i := pointer+196 TO pointer+208 DO
       IF Rorl = 'R' THEN
         FIO.WrReal(outfile,R_data[i],decimal,placement)
       ELSE
         FIO.WrReal(outfile,I_data[i],decimal,placement)
       END;
       FIO.WrInt(outfile,attfactor[i],placement);
       FIO.WrLn(outfile)
    END; WrUnity(2);
    FIO.Close(outfile);
    IF Rorl = 'R' THEN
      GotoXY(16,10); ChColor(Rname,LightGray,White);
    ELSE
      GotoXY(16,11); ChColor(Iname,LightGray,White);
    END;
    Rorl := 'I'
  END;
END OutputFormat;

PROCEDURE OutputFile;
LABEL ReEnter,EndOutputFile;
VAR i,j,k,ub,lb,pointer : INTEGER; realdata,imaginarydata : REAL;
    intdata : LONGINT; output_window : WinType; Rorl,ext1,ext2,avgsep : CHAR;
    attfactor : ARRAY [1..4160] OF INTEGER;
    inname,R_filename,I_filename,R_tmpfilename,I_tmpfilename : file_array;
BEGIN
    output_window := Open(WinDef(0,12,79,25,White,Blue,TRUE,FALSE,
                     FALSE,TRUE,SingleFrame,LightGray,Blue));
    SetTitle(output_window," Output Data to File(s) ",
          CenterUpperTitle);
    Clear; WrLn;

(* == data conversion == *)

(*
    outfile := FIO.Create('hex.out');
    FOR i := 1 TO maxdata DO
       FIO.WrShtHex(outfile,potential[i],1); FIO.WrLn(outfile);
    END;
    FIO.Close(outfile);
*)

    i := 1; j := 1; Rorl := 'R';
    FOR i := 1 TO maxdata BY 2 DO
      ub := INTEGER(potential[i]);
      lb := INTEGER(potential[i+1]);
      IF ub < 16 THEN attfactor[j] := 1;
      ELSIF ub < 32 THEN attfactor[j] := 2; ub := ub-16
      ELSIF ub < 48 THEN attfactor[j] := 4; ub := ub-32
      ELSIF ub < 64 THEN attfactor[j] := 8; ub := ub-48
      ELSIF ub < 80 THEN attfactor[j] := 16; ub := ub-64
      ELSIF ub < 96 THEN attfactor[j] := 32; ub := ub-80
      ELSIF ub < 112 THEN attfactor[j] := 64; ub := ub-96
      ELSIF ub < 128 THEN attfactor[j] := 128; ub := ub-112
      ELSE attfactor[j] := 0; Alarm(3500,100,2);
         ChColor('Error detected in data stream',LightRed,White); WrLn;
      END;
(*     intdata := LONGINT(256*ub+lb)*LONGINT(attfactor[j]);*)
      intdata := LONGINT(256*ub+lb);
      (* 1V software offset = 1638 *)
      (* 0V software offset = 2048 *)
      IF Rorl = 'R' THEN
        R_data[j] := REAL(intdata-2048)*5./2048.; Rorl := 'I';
      ELSE
        I_data[j] := REAL(intdata-2048)*5./2048.; Rorl := 'R'; j := j+1;
      END;
    END;
(* == output data == *)
    GotoXY(3,2);
    WrStr('Total number of frame(s) recorded '); TextColor(Yellow);
    WrCard(numfm,1); TextColor(White); ext1 := '1'; ext2 := '0';
    GotoXY(3,4); WrStr('Enter filename ');
    ChColor('[No extension]',LightGray,White); WrStr(' : ');
ReEnter:
    FOR j := 1 TO 78 DO inname[j] := 0C END;
    RdItem(inname);
    FOR j := 1 TO 78 DO
      IF inname[j] = '.' THEN
        GotoXY(18,4); ChColor('[No extension]',LightRed,White);
        WrStr(' : '); Alarm(3500,100,2); GotoXY(35,4); ClrEol;
        GOTO ReEnter
      END;
    END;
    R_filename := inname; I_filename := inname;
    j := 1;
```

```
  LOOP
    IF inname[j] = null THEN
      R_filename[j] := '.';
      R_filename[j+1] := 'R'; R_filename[j+2] := 0C;
      I_filename[j] := '.';
      I_filename[j+1] := 'I'; I_filename[j+2] := 0C; EXIT
    END; j := j+1
  END;
  IF numfm = 1 THEN avgsep := '1' ELSE
    GotoXY(3,6); ChColor('1',Brown,White); WrStr('.  Average');
    GotoXY(3,7); ChColor('2',Brown,White); WrStr('.  Separate');
    GotoXY(3,8);WrStr('Please enter... '); avgsep := RdKey();
  END;
  IF avgsep = esc THEN GOTO EndOutputFile END;
  GotoXY(3,10); WrStr('File [real]: ');
  GotoXY(3,11); WrStr('[imaginary]: ');
  LOOP
  CASE avgsep OF
  '1','A','a' : (* == average == *)
      FOR i := 1 TO 208 DO
        realdata := 0.; imaginarydata := 0.;
        FOR k := 1 TO numfm DO
          pointer := (k-1)*208+i;
          realdata := realdata+R_data[pointer];
          imaginarydata := imaginarydata+I_data[pointer];
        END;
        R_data[i] := realdata/REAL(numfm);
        I_data[i] := imaginarydata/REAL(numfm);
      END;
      OutputFormat(avgsep,1,R_filename,I_filename);
      Alarm(4000,100,1); EXIT|
  '2','S','s' : (* == separate == *)
      FOR k := 1 TO numfm DO
        R_tmpfilename := R_filename;
        I_tmpfilename := I_filename;
        LOOP
         FOR j := 1 TO 78 DO
          IF R_tmpfilename[j] = '.' THEN
            IF k < 10 THEN
              R_tmpfilename[j+2] := ext1; R_tmpfilename[j+3] := 0C;
              I_tmpfilename[j+2] := ext1; I_tmpfilename[j+3] := 0C;
              ext1 := CHAR(ORD(ext1)+1); EXIT
            ELSE
              R_tmpfilename[j+2] := '1'; I_tmpfilename[j+2] := '1';
              R_tmpfilename[j+3] := ext2; R_tmpfilename[j+4] := 0C;
              I_tmpfilename[j+3] := ext2; I_tmpfilename[j+4] := 0C;
              ext2 := CHAR(ORD(ext2)+1); EXIT
            END;
          END;
         END;
        END;
        OutputFormat(avgsep,k,R_tmpfilename,I_tmpfilename);
      END; Alarm(4000,100,1); EXIT|
    esc : EXIT
    ELSE Alarm(3500,100,2); GotoXY(3,6);
      ChColor('Either <1> or <2>',LightRed,White);
      WrStr('.  Please enter... '); avgsep := RdKey();
    END;
    END;
  (*
    tmpfile := FIO.Create('Hex.out');
    j := 1;
    FOR i := 1 TO 416 BY 2 DO
      FIO.WrShtHex(tmpfile,potential[i],3);
      FIO.WrShtHex(tmpfile,potential[i+1],3); FIO.WrLn(tmpfile); j := j+1;
      IF j = 14 THEN FIO.WrLn(tmpfile); j := 1 END;
    END; FIO.Close(tmpfile);
  *)
  EndOutputFile:
    Close(output_window);
  END OutputFile;

  PROCEDURE SlowVoltMeas;
  VAR mode,pause : CHAR; dwell,drive,measurement,data : CARDINAL;
    meas_window : WinType;
  LABEL Redo;
  BEGIN
    meas_window := Open(WinDef(0,12,79,25,LightGray,Blue,TRUE,FALSE,
                        FALSE,TRUE,SingleFrame,LightGray,Blue));
    SetTitle(meas_window," Slow Voltage Measurements ",CenterUpperTitle);
    GotoXY(3,2); WrStr('Selection of possible operation modes:');
    GotoXY(3,4); ChColor('1',Brown,LightGray); WrStr('.  Automatic');
    GotoXY(3,5); ChColor('2',Brown,LightGray); WrStr('.  Manual');
    GotoXY(3,7); WrStr('Please enter... ');
    REPEAT
     mode := RdKey();
     CASE mode OF
     |'1','A','a' : SendSignal(0F0H);
                GotoXY(3,9); ClrEol; GotoXY(3,10); ClrEol;
                GotoXY(3,11); ClrEol; GotoXY(3,7); ClrEol;
                WrStr('Dwell time between each measurement (x 0.1 second): ');
                dwell := RdCard();
                IF dwell<1 THEN dwell := 1 END;
                GotoXY(3,9);  WrStr('Data Number     : ');
                GotoXY(3,10); WrStr('Drive pair      : ');
                GotoXY(3,11); WrStr('Measurement pair : ');
                GotoXY(40,9); WrStr('Data Conversion');
                GotoXY(40,10); WrStr('Real');
                GotoXY(46,10); WrStr('Imaginary');
                FOR drive := 0 TO 15 DO
                FOR measurement := 1 TO 13 DO
                  GotoXY(22,10); ClrEol; GotoXY(22,11); ClrEol;
                  GotoXY(22,9); WrCard(drive*13+measurement,1);
                  GotoXY(22,10);
                  IF drive = 0 THEN WrCard(16,2)
                  ELSE WrCard(drive,2) END;
```

```
                GotoXY(25,10); WrStr('-'); GotoXY(27,10);
                WrCard(drive+1,1);
                GotoXY(22,11); WrCard(measurement,1);
                GotoXY(46,10); WrStr('Imaginary');
                GotoXY(40,10); ChColor('Real',Yellow,LightGray);
                Delay(100*dwell); SendSignal(00H);
                IF KeyPressed() THEN pause := RdKey();
                  GotoXY(40,11); WrStr('Strike any key to continue');
                  pause := RdKey(); GotoXY(40,11); ClrEol;
                  IF pause = esc THEN SendSignal(0FFH); GOTO Redo END;
                END;
                GotoXY(40,10); WrStr('Real');
                GotoXY(46,10); ChColor('Imaginary',Yellow,LightGray);
                Delay(100*dwell); SendSignal(00H);
                IF KeyPressed() THEN pause := RdKey();
                  GotoXY(40,11); WrStr('Strike any key to continue');
                  pause := RdKey(); GotoXY(40,11); ClrEol;
                  IF pause = esc THEN SendSignal(0FFH); GOTO Redo END;
                END;
              END;
              END;
              GotoXY(3,9); ClrEol; GotoXY(3,10); ClrEol;
              GotoXY(3,11); ClrEol;
    |'2','M','m' : SendSignal(0F0H); GotoXY(3,7); ClrEol;
              WrStr('Data to be monitored [enter data number] : ');
              data := RdCard();
              FOR drive := 0 TO 15 DO
              FOR measurement := 1 TO 13 DO
                IF drive*13+measurement=data THEN
                  GotoXY(3,9);  WrStr('Data Number     : ');
                  GotoXY(3,10); WrStr('Drive pair      : ');
                  GotoXY(3,11); WrStr('Measurement pair : ');
                  GotoXY(22,9); ClrEol; WrCard(drive*13+measurement,1);
                  GotoXY(22,10);
                  IF drive = 0 THEN WrCard(16,2)
                  ELSE WrCard(drive,2) END;
                  GotoXY(25,10); WrStr('-'); GotoXY(27,10);
                  WrCard(drive+1,1);
                  GotoXY(22,11); WrCard(measurement,1);
                  GotoXY(40,9); WrStr('Data Conversion');
                  GotoXY(40,10); ChColor('Real',Yellow,LightGray);
                  GotoXY(46,10); WrStr('Imaginary');
                  pause := RdKey(); SendSignal(00H);
                  GotoXY(46,10); ChColor('Imaginary',Yellow,LightGray);
                  GotoXY(40,11); WrStr('Strike any key to continue');
                  pause := RdKey(); SendSignal(0FFH); GOTO Redo;
                ELSE
                  Delay(10); SendSignal(00H);
                  Delay(10); SendSignal(00H);
                END;
              END;
              END;
    |esc     : SendSignal(0FFH);
    ELSE Alarm(3500,100,2);
    END;
Redo:  GotoXY(3,9); ClrEol; GotoXY(3,10); ClrEol;
    GotoXY(3,11); ClrEol; GotoXY(3,7); ClrEol; WrStr('Please enter... ');
  UNTIL mode=esc;
  Close(meas_window);
END SlowVoltMeas;


PROCEDURE ResetWorkWindow;
BEGIN
  SetTitle(work_window," Working Window ",CenterUpperTitle);
  SetFrame(work_window,SingleFrame,White,Blue);
  TextColor(White); TextBackground(Blue); Clear
END ResetWorkWindow;

BEGIN (* main program *)
  mapexists := FALSE; ComPort; Menu;
  numfm := 2; maxdata := numfm*frmdata;
  FOR i := 1 TO 8320 DO potential[i] := 00H END;
  work_window := Open(WinDef(0,12,79,15,White,Blue,TRUE,FALSE,FALSE,
                                                    TRUE,SingleFrame,White,Blue));
  ResetWorkWindow; WrStr('  Please enter... ');
  REPEAT
   ansop := RdKey();
   LOOP IF RxCount() # 0 THEN Receive(clr_rx_count,1) ELSE EXIT END END;
   IF ansop = CHAR(0) THEN
     ansop := RdKey();
     CASE ansop OF
     |CHAR(32) : DosScreen := Open(FullScreenDef);
              TextColor(Yellow);
              WrStr('Type EXIT to return to Electrical Impedance Tomography');
              Storage.ALLOCATE(address,65535);
              iaddress := Execute("C:\COMMAND.COM","",address,65535 DIV 8);
              Storage.DEALLOCATE(address,65535);
(*
              Storage.ALLOCATE(address,65535);
              iaddress := Execute("C:\COMMAND.COM","",address,65535 DIV 16);
              Storage.DEALLOCATE(address,65535);
*)
              Close(DosScreen); PutOnTop(FullScreen);
              PutOnTop(menu_window); PutOnTop(update_window);
              PutOnTop(work_window); Clear;
     |CHAR(45) : Clear; ChColor('  Press <esc> to confirm',LightRed,White);
              confirm := RdKey(); Clear;
              IF confirm = esc THEN GOTO exit ELSE GOTO alarm END
     |CHAR(59) : InitData; abort := FALSE; PutOnTop(work_window);
              SetTitle(work_window," Voltage Measurements ",
              CenterUpperTitle);
              SetFrame(work_window,SingleFrame,LightGray,Blue);
              Clear; SendSignal(02H); ChnFrmNum;
             WaitAcknowledge;
             IF abort = FALSE THEN
```

```
                WrStr('  Number of frames required = ');
                TextColor(Yellow); WrCard(numfm,2); WrLn;
                TextColor(White);
               WrStr('  Measurements are in progress...');
                FabMeasurement;
              END; ResetWorkWindow
        |CHAR(60) : SetTitle(work_window," Serial Transmission ",
              CenterUpperTitle);
              SetFrame(work_window,SingleFrame,LightGray,Blue);
              abort := FALSE; Clear; SendSignal(03H);
              WrStr('  Number of frames to be received     = ');
              TextColor(Yellow); WrCard(numfm,1); WrLn;
              ChColor('  Total number of bytes to be received = ',White,Yellow);
              WrCard(maxdata,3); TextColor(White); CursorOff;
              ReceiveData;
              IF abort # TRUE THEN SendSignal(05H);
                ReceiveVerifyData; VerifyRecvdData
              END; ResetWorkWindow; CursorOn;
        |CHAR(61) : Clear; DisplayData; Clear
        |CHAR(62) : Clear; OutputFile; Clear
        |CHAR(63) : SetTitle(work_window," Initialisation of Memory Buffer ",
              CenterUpperTitle);
              SetFrame(work_window,SingleFrame,LightGray,Blue);
              Clear; SendSignal(04H); InitBuffer; ResetWorkWindow
        |CHAR(64) : Hide(update_window); Hide(work_window);
              SendSignal(01H); EITSetUp.CurrentSetUp;
              InitData; PutOnTop(work_window); SendSignal(0FFH);
              PutOnTop(update_window); PutOnTop(work_window);
        |CHAR(65) : imagemenu_window := Open(WinDef(0,1,79,15,White,Blue,TRUE,
              FALSE,FALSE,TRUE,DoubleFrame,White,Blue));
              SetTitle(imagemenu_window," Graphics Operations ",
              CenterUpperTitle);
              Use (imagemenu_window); Clear;
              LOOP
               GotoXY(3,2); WrStr('Selection of possible operations:');
               WrLn; WrLn; ChColor('  1',Brown,Yellow);
               ChColor('.  Start Image Reconstruction',Yellow,Brown);
               WrLn; WrStr('  2');
               ChColor('.  Display Image',Yellow,White); WrLn;
               WrLn; WrStr('  Press <'); ChColor('esc',LightRed,White);
               WrStr('> to quit graphics routines');
               GotoXY(3,9); WrStr('Please enter... ');
               choice := RdKey();
               CASE choice OF
               |'1' : abort := FALSE;
                    IF mapexists = FALSE THEN MakeMap;
                      IF abort # TRUE THEN mapexists := TRUE END
                    END;
                    IF abort # TRUE THEN RPImaging END; Clear
               |'2' : EITDisp.Main(portnum); Install(portnum,First);
                    PutOnTop(imagemenu_window); Clear
               |esc : EXIT
               ELSE
                 Alarm(3500,100,2); GotoXY(3,11);
                 WrStr('Either <1> or <2>.  ');
               END; (* CASE *)
              END; (* LOOP *)
              Close(imagemenu_window);
              PutOnTop(work_window); Clear
        |CHAR(66) : Hide(menu_window); Hide(update_window); Hide(work_window);
              EITNoise.main; PutOnTop(menu_window);
              PutOnTop(update_window); PutOnTop(work_window); Clear
        |CHAR(67) : Hide(update_window);
              EITDFreq.DualFreqSettings(numfm);
              EITDFreq.DualFreqMeasurements;
              maxdata := numfm*frmdata;
              PutOnTop(update_window); PutOnTop(work_window); Clear
        |CHAR(68) : Hide(update_window);
              EITCG1.CGISettings(numfm);
              EITCG1.CardiacGatedImaging;
              maxdata := numfm*frmdata;
              PutOnTop(update_window); PutOnTop(work_window); Clear
        |CHAR(104): InitData; SendSignal(08H); SlowVoltMeas; SendSignal(0FFH);
              PutOnTop(work_window); ResetWorkWindow
        ELSE GOTO alarm
        END; (* of CASE *) Clear; GOTO next
      END;
alarm: Alarm(3500,100,2); Clear; ChColor('  ERROR:',LightRed,Yellow);
      WrStr(' Input should be one of the above choices.');
      TextColor(White);
next:  WrStr('  Please enter... ');
    UNTIL FALSE;
exit:Close(update_window);
    Close(menu_window);
    Close(work_window);
    Close(FullScreen); CursorOn;
END EITSW5.
```

Appendix E A Program Listing of 6303 Assembly Languages

```
*****************************************************************************
*
*                PROGRAM          : EITSW5
*                TARGET           : Electrial Impedance Tomography System
*                FUNCTION         : To control the operation of the EIT system
*                DATE             : May 16, 1991        AUTHOR : Hing Tong LEUNG
*
*****************************************************************************
*
                 model            m6800              ;call relevant assembler
*
eitsw5           asect            rom                ;identify target memory type
*
ddr1             equ              h'0000'
ddr2             equ              h'0001'
dr1              equ              h'0002'
dr2              equ              h'0003'
rmcr             equ              h'0010'
trcsr            equ              h'0011'
rdr              equ              h'0012'
tdr              equ              h'0013'
*
sc               equ              b'0020'            ;conversion start
ub               equ              h'0021'            ;upper byte
lb               equ              h'0022'            ;lower byte
latch            equ              h'0023'            ;attenuation factor latch
m0               equ              h'0024'            ;driving electrode control
m1               equ              h'0025'            ;monitoring electrode control
m2               equ              h'0026'            ;real and imaginary selection control
m3               equ              h'0027'            ;R and I switch reset
m4               equ              h'0028'            ;amplitude and frequency control
*
numfm            equ              h'0080'            ;number of frame(s)
numit            equ              h'0081'            ;number of iteration(s)
drpost           equ              h'0082'            ;active driving electrode position
atvald           equ              h'0083'            ;status of recorded data (att. factor)
setatt           equ              h'0084'            ;set att. factor flag ('f0' : adjust)
nummon           equ              h'0085'            ;number of monitoring positions
enaddr           equ              h'0086'            ;address pointer of the last+1 data
verdat           equ              h'0088'            ;transmission verifying data
vergt            equ              h'0089'            ;verifying data (greater than)
verlt            equ              h'008a'            ;verifying data (less than)
attadd           equ              h'5ffc'            ;address pointer for next att. factor
rtnadd           equ              h'5ffe'            ;address pointer for actual data
atcode           equ              h'5f00'            ;att. factor codes (start address)
abort            equ              h'5ff8'            ;abort flag
dffm             equ              h'5ff9'            ;no. of sets for dual freq.
setup1           equ              h'5ffa'            ;first set up code for dual freq.
setup2           equ              h'5ffb'            ;second set up code for dual freq.
*
sel1             equ              h'01'              ;code for change current setting
sel2             equ              h'02'              ;code for conversion
sel3             equ              h'03'              ;code for transmission
sel4             equ              h'04'              ;code for ram initialisation
sel3a            equ              h'05'              ;code for verification code
sel5             equ              h'06'              ;code for dual frequency measurements
sel6             equ              h'07'              ;code for cardiac gated imaging
sel7             equ              h'08'              ;code for stepping thro' volt. meas.
fshopt           equ              h'ff'              ;end of change current setting code
*
first            equ              b'1000'            ;first address of ram
last             equ              b'6000'            ;last+1 address of ram
*
                 org              h'f000'            ;set base address for assembly
*
*****************************************************************************
*
*                SYSTEM INITIALISATION
*
*****************************************************************************
*
*
start            lds              #h'00ff'           ;initialise stack pointer
                 ldaa             #h'00'             ;set port one as an input port
                 staa             ddr1
                 ldaa             #h'10'             ;set port two bit 4 as an output
                 staa             ddr2               and bits 0, 1, 2, 3 as inputs
                 ldaa             #h'05'             ;set baud rate
                 staa             rmcr
                 ldaa             #h'2a'             ;enable both receive and transmit
                 staa             trcsr
*
*****************************************************************************
*
*                INITIAL SET UP
*
*****************************************************************************
*
                 ldaa             #h'07'             ;set initial freq =      40 kHz
                 staa             m4                 and amp = 2.5 mA
                 ldaa             #h'10'             ;set initial active electrodes 16/1
                 staa             m0
                 ldaa             #h'cb'             ;set initial recording electrodes 4/5
                 staa             m1
                 clra
                 staa             latch
                 staa             m3                 ;set to R part
                 ldaa             #h'02'             ;defealt number of frame = 2 : R
                 staa             numfm
```

```
                ldx             #h'1680'        ;default last+1 address (i.e 2 frames)
                stx             enaddr
                ldx             #first
                clra
                clrb
lpclr           staa            x
                inx
                clr             x
                incb
                cmpb            #h'0d'
                beq             suma
backsm          inx
                cpx             #h'1680'
                bmi             lpclr
                jmp             mstart
suma            clrb
                adda            #h'11'
                cmpa            #h'88'
                bne             backsm
                clra
                bra             backsm
*
*****************************************************************************
*
*               MAIN PROGRAM SEQUENCES
*
*****************************************************************************
*
mstart          nop
*               jsr             optn0           ;added on 20/9/90 (temp. removed!)
                clc                             ;clear carry flag
                ldaa            trcsr           ;polling trcsr
                rola                            ;rotate content to left
                bcc             mstart          ;if set, poll again
                ldaa            trcsr           ;clear RDRF by reading trcsr and rdr
                ldaa            rdr             ;get data from rdr
                cmpa            #sel1           ;interpret command and branch to the
                beq             optn1           ;right subroutine
                cmpa            #sel2
                beq             optn2
                cmpa            #sel3
                beq             optn3
                cmpa            #sel4
                beq             optn4
                cmpa            #sel3a
                beq             optn3a
                cmpa            #sel5
                beq             optn5
                cmpa            #sel6
                beq             optn6
                cmpa            #sel7
                beq             optn7
                bra             mstart
*
optn1           jsr             routcc          ;goto the alteration of current
                bra             mstart          ;setting routine
optn2           jsr             routsc          ;goto the measurement routine
                bra             mstart
optn3           jsr             routst          ;goto the serial transmission routine
                bra             mstart
optn4           jsr             routin          ;goto the initialisation routine
                bra             mstart
optn3a          jsr             routve          ;goto the transmission of verifying
                bra             mstart          ;bytes routine
optn5           jsr             routdf          ;goto the dual frequency measurement
                bra             mstart          ;routine
optn6           jsr             routcg          ;goto the cardiac gated imaging routine
                bra             mstart
optn7           jsr             routsm          ;goto the slow meas. routine
                bra             mstart
*
*****************************************************************************
*
*               SUBROUTINES
*
*****************************************************************************
*
*****************************************************************************
*
*****************************************************************************
*
*               Option 1 : CHANGE CURRENT SETTING ROUTINE
*
*****************************************************************************
*
routcc          clc
                ldaa            trcsr
                rola
                bcc             routcc
                ldaa            trcsr
                ldaa            rdr
                cmpa            #fshopt
                beq             endcc
                staa            m4
                bra             routcc
endcc           nop
                rts
*
*****************************************************************************
*
*               Option 2 : START CONVERSION SUBROUTINE
*
*****************************************************************************
*
routsc          ldaa            #h'01'          ;default no. of frame = 1
                staa            numfm
                clc                             ;get att. factor flag
chsatt          ldaa            trcsr           ;if flag = 'f0' then set att. factor
```

```
                rola
                bcc             chnatt
                ldaa            trcar
                ldaa            rdr
                staa            setatt
                cmpa            #h'f0'
                bne             gfmnum

                ldx             #first          ;initialise att. factors,
clear           clr             x               ;actual data and address pointers
                inx
                cpx             #last
                bne             clear
                ldaa            #h'07'          ;set max. iterations to 7 times
                staa            numit
                jsr             chkcpt
                ldaa            trcar
                ldab            numfm
                stab            tdr
samefm          ldx             #atcode
                ldaa            #h'10'
                staa            drpost
                ldaa            #h'f0'
                staa            atvald
attrec          ldaa            drpost
                staa            m0
                bsr             record
                dec             drpost
                bne             attrec
                ldaa            atvald
                cmpa            #h'f0'
                beq             endset
                dec             numit
                bne             samefm
endset          clr             setatt
                bra             fframe

gfmnum          clc                             ;acquire a number of frame(s) to be
                ldaa            trcar           ;recorded
                rola
                bcc             gfmnum
                ldab            trcar           ;save the frame number in [numfm]
                ldab            rdr
                stab            numfm
stcon           jsr             chkcpt
                ldaa            trcar
                ldab            numfm
                stab            tdr             ;acknowledge the frame number
fframe          ldx             #first          ;set first return address
                stx             rtnadd
nextfm          ldx             #atcode         ;load first att. factor code to x
                ldaa            #h'10'
                staa            drpost
rocimp          ldaa            drpost
                staa            m0
*               jsr             conver          ;temporarily removed!
                bar             record
                dec             drpost
                bne             rocimp
donext          dec             numfm
                bne             nextfm
*
endsc           ldx             rtnadd
                stx             cnaddr          ;save the last+1 data storage address
                ldah            #fahopt
                jsr             chkcpt
                ldaa            trcar
                stab            tdr
                ldaa            #h'10'
                staa            m0
                ldaa            atcode+7
                staa            latch
                ldaa            #h'87'          ;changed from 'ed' to '87'
                staa            m1
                rts
*
*****************************************************************************
*
record          ldaa            #h'0d'          ;reset the number of recording signals
                staa            nummon
                ldaa            drpost          ;check the active driving electrodes
                cmpa            #h'10'
                beq             lab1
                cmpa            #h'0f'
                beq             lab2
                cmpa            #h'0e'
                beq             lab3
                cmpa            #h'0d'
                beq             lab4
                cmpa            #h'0c'
                beq             lab5
                cmpa            #h'0b'
                beq             jmp6
                cmpa            #h'0a'
                beq             jmp7
                cmpa            #h'09'
                beq             jmp8
                cmpa            #h'08'
                beq             jmp9
                cmpa            #h'07'
                beq             jmp10
                cmpa            #h'06'
                beq             jmp11
                cmpa            #h'05'
                beq             jmp12
                cmpa            #h'04'
```

```
                beq             jmp13
                cmpa            #h'03'
                beq             jmp14
                cmpa            #h'02'
                beq             jmp15
                cmpa            #h'01'
                beq             jmp16
jmp6            jmp             lab6
jmp7            jmp             lab7
jmp8            jmp             lab8
jmp9            jmp             lab9
jmp10           jmp             lab10
jmp11           jmp             lab11
jmp12           jmp             lab12
jmp13           jmp             lab13
jmp14           jmp             lab14
jmp15           jmp             lab15
jmp16           jmp             lab16
jmpstp          jmp             stprec
*
lab1            ldaa            #h'ed'          ;sequences of monitoring position
                jsr             action          ;select monitoring electrodes and
                dec             nummon          ;carry out the A-D conversion. Upon
                beq             jmpstp          ;the completion of 13th conversion,
lab2            ldaa            #h'dc'          ;go back to alter the position of the
                jsr             action          ;driving electrode pair.
                dec             nummon
                beq             jmpstp
lab3            ldaa            #h'cb'
                jsr             action
                dec             nummon
                beq             jmpstp
lab4            ldaa            #h'ba'
                jsr             action
                dec             nummon
                beq             jmpstp
lab5            ldaa            #h'a9'
                jsr             action
                dec             nummon
                beq             stprec
lab6            ldaa            #h'98'
                jsr             action
                dec             nummon
                beq             stprec
lab7            ldaa            #h'87'
                jsr             action
                dec             nummon
                beq             stprec
lab8            ldaa            #h'76'
                jsr             action
                dec             nummon
                beq             stprec
lab9            ldaa            #h'65'
                jsr             action
                dec             nummon
                beq             stprec
lab10           ldaa            #h'54'
                jsr             action
                dec             nummon
                beq             stprec
lab11           ldaa            #h'43'
                jsr             action
                dec             nummon
                beq             stprec
lab12           ldaa            #h'32'
                jsr             action
                dec             nummon
                beq             stprec
lab13           ldaa            #h'21'
                jsr             action
                dec             nummon
                beq             stprec
lab14           ldaa            #h'10'
                jsr             action
                dec             nummon
                beq             stprec
lab15           ldaa            #h'0f'
                jsr             action
                dec             nummon
                beq             stprec
lab16           ldaa            #h'fe'
                jsr             action
                dec             nummon
                beq             stprec
                jmp             lab1
stprec          rts
*
**************************************************************************
*
action          ldab            setatt
                cmpb            #h'f0'
                bne             normal          ;If set att. factor then jmp to normal
*
                staa            m1              ;initiate the ADC
                ldaa            x
                staa            latch
                staa            m3              ;switch to R part
                ldaa            #h'06'          ;added on 19/2/90 to increase the
fdly1           ldab            #h'70'          ;settling time due to low-pass filters
fset1           decb
                bne             fset1
                deca
                bne             fdly1           ;end: 19/2/90
                staa            sc
                staa            sc
chkf1           clc                             ;check whether the conversion is
```

```
                ldaa    drl             ;finished
                rola
                bcc     chkf1
                ldab    #h'03'          ;allow data to settle down
sdly1           decb
                bne     sdly1
                ldab    ub
                ldaa    ub
                rora
                bcc     store
                rora
                bcc     store
                rora
                bcc     store
                rora
                bcc     store
                ldaa    lb
                rola
                bcc     store
                rola
                bcc     store
                rola
                bcc     store
                rola
                bcc     store
                addb    #h'10'
                stab    x
                inx
                clr     stvald
                jmp     valid
store           stab    x
                inx
                jmp     valid
*
normal          staa    m1              ;initiate the ADC
                ldaa    x
                staa    latch
                inx                     ;increment att. factor address pointer
                stx     attadd          ;store address pointer
                ldx     rtnadd          ;get actual data address
                staa    m3              ;switch to R part
                ldaa    #h'06'          ;added on 19/2/90 to increase the
fdly2           ldab    #h'70'          ;settling time due to low-pass filters
fset2           decb
                bne     fset2
                deca
                bne     fdly2           ;end: 19/2/90
                staa    sc
                staa    sc
chkf2           clc                     ;check whether the conversion is
                ldaa    drl             ;finished
                rola
                bcc     chkf2
                ldab    #h'03'          ;allow data to settle down
sdly2           decb
                bne     sdly2
                ldab    ub
                stab    x
                inx
                ldab    lb
                stab    x
                inx
*               stx     rtnadd          ;store actual address pointer
*               ldx     attadd          ;get att. factor address
*
                stab    m2              ;switch to Imaginary Part
*
*               ldab    #h'01'
*long1          ldaa    #h'ff'
*long2          deca
*               bne     long2
*               decb
*               bne     long1
*
                ldab    #h'03'          ;allows signal to settle
sdly3           decb
                bne     sdly3
                staa    sc
                staa    sc
chkf3           clc
                ldaa    drl
                rola
                bcc     chkf3
                ldab    #h'03'
sdly4           decb
                bne     sdly4
                ldaa    ub
                staa    x
                inx
                ldaa    lb
                staa    x
                inx
                stx     rtnadd          ;store actual address pointer
                ldx     attadd          ;get att. factor address
valid           rts
*
*****************************************************************************
*
*               Option 3 : START SERIAL TRANSMISSION ROUTINE
*
*****************************************************************************
*
routst          ldx     ensddr          ;start the serial transmission process
                dex
lptx            ldab    x
                bar     chkcpt
```

```
                ldaa        trcsr
                stab        tdr
                dex
                cpx         #h'0fff'
                bne         lptx
                rts
*
*****************************************************************************
*
*               Option 3a : TRANSMISSION OF VERIFYING DATA SUBROUTINE
*
*****************************************************************************
*
routve          clr         verdat
                clr         vergt
                clr         verlt
                idx         #first
lpver           ldaa        x
                bmi         negate
                tsta
                beq         nextda
bkneg           cmpa        #h'80'
                beq         nextda
                cmpa        verdat
                bgt         amm
                cmpa        verdat
                blt         mma
nextda          inx
                cpx         enaddr
                bne         lpver
                jmp         stpgen
mma             tab
                ldaa        verdat
                sba
                staa        verdat
                staa        verlt
                bra         nextda
amm             ldab        verdat
                sba
                staa        verdat
                staa        vergt
                bra         nextda
negate          nega
                bra         bkneg
stpgen          idx         #verdat
lpgen           idab        x
                bar         chkcpt
                ldaa        trcsr
                stab        tdr
                inx
                cpx         #verlt+1
                bne         lpgen
                rts
*
*****************************************************************************
*
chkcpt          clc                             ;clear carry flag
                ldaa        trcsr               ;get transmission status
                rola
                rola
                rola
                bcc         chkcpt              ;if set, data sent to shift register
                rts
*
*****************************************************************************
*
*               Option 4 : INITIALISE STATIC RAM SUBROUTINE
*
*****************************************************************************
*
routin          idx         #first
lpinit          clr         x
                inx
                cpx         #last
                bne         lpinit
                rts
*
*****************************************************************************
*
*               Option 5 : DUAL FREQUENCY MEASUREMENTS
*
*****************************************************************************
*
routdf          idx         #first
clrdf           clr         x
                inx
                cpx         #atcode
                bne         clrdf
*
                bar         chkcpt              ;acknowledgment
                ldaa        trcsr
                ldab        #sel5
                stab        tdr
*
lpdf1           clc
                ldaa        trcsr
                rola
                bcc         lpdf1
                ldaa        trcsr
                ldaa        rdr
                staa        dffm
                bar         chkcpt
                ldaa        trcsr
                ldab        dffm
                stab        tdr
*
```

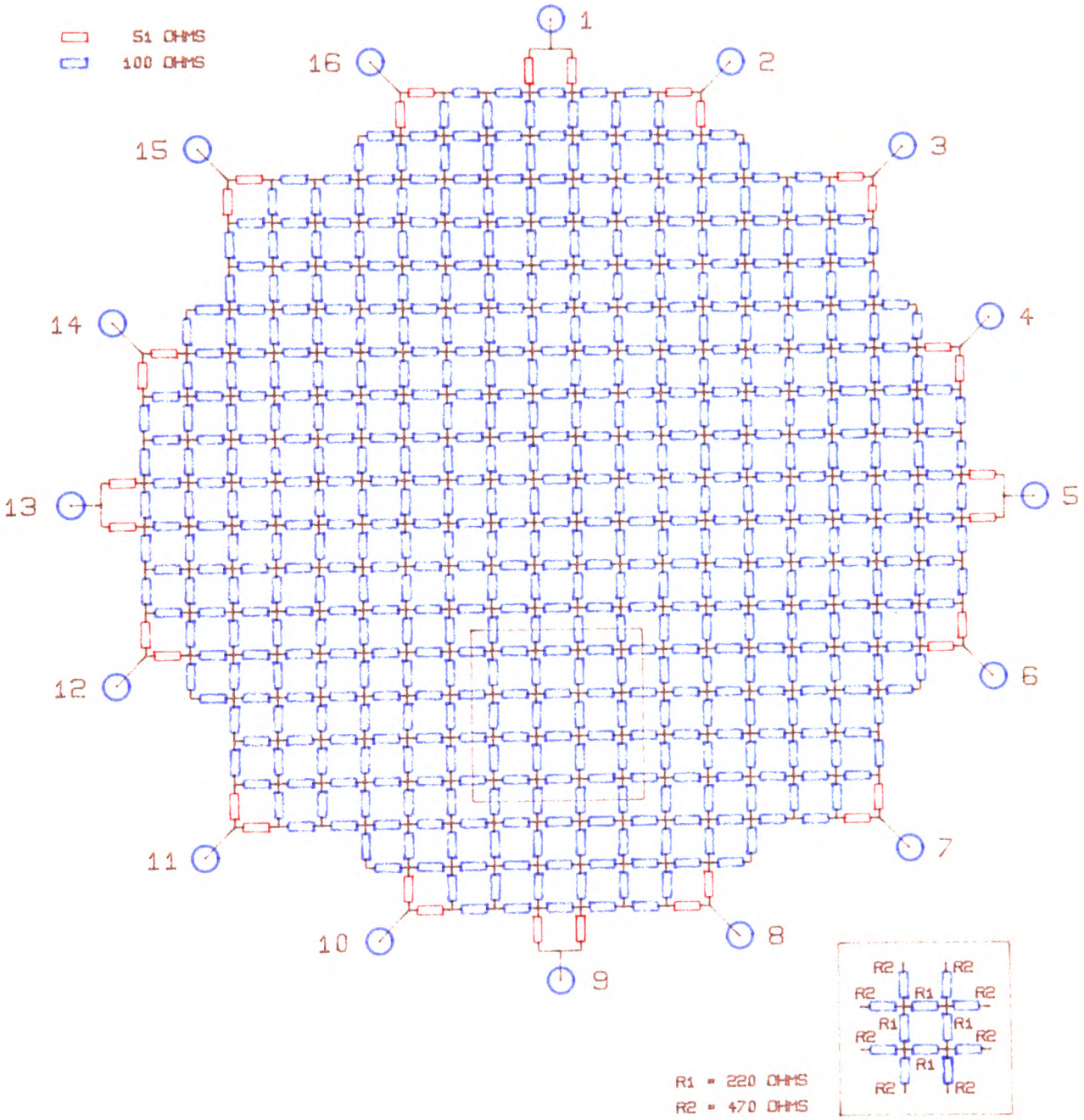
```
lpdf2           clc
                ldaa        trcar
                rola
                bcc         lpdf2
                ldaa        trcar
                ldaa        rdr
                staa        setup1
                bsr         chkcpt
                ldaa        trcar
                ldab        setup1
                stab        tdr
*
lpdf3           clc
                ldaa        trcar
                rola
                bcc         lpdf3
                ldaa        trcar
                ldaa        rdr
                staa        setup2
                bsr         chkcpt
                ldaa        trcar
                ldab        setup2
                stab        tdr
*
                clr         setatt
                ldx         #first          ;set first return address
                stx         rtnadd
*
real            ldaa        setup1
                staa        m4
                ldx         #atcode         ;load first att. factor code to x
                ldaa        #h'10'
                staa        drpost
rreal           ldaa        drpost
                staa        m0
                jsr         record
                dec         drpost
                bne         rreal
*
imag            ldaa        setup2
                staa        m4
                ldx         #atcode         ;load first att. factor code to x
                ldaa        #h'10'
                staa        drpost
rimag           ldaa        drpost
                staa        m0
                jsr         record
                dec         drpost
                bne         rimag
*
nextdf          dec         dffm
                bne         real
*
enddf           ldx         rtnadd
                stx         enaddr          ;save the last+1 data storage address
                ldab        #fshopt
                jsr         chkcpt
                ldaa        trcar
                stab        tdr
                ldaa        #h'10'
                staa        m0
                ldaa        atcode+7
                staa        latch
                ldaa        #h'87'          ;changed from 'ed' to '87'
                staa        m1
                rts
*
*****************************************************************************
*
*               Option 6 : CARDIAC GATED IMAGING
*
*****************************************************************************
*
routcg          ldx         #first
clrcg           clr         x
                inx
                cpx         #atcode
                bne         clrcg
*
                jsr         chkcpt          ;acknowledgment
                ldaa        trcar
                ldab        #sel6
                stab        tdr
*
lpcg1           clc
                ldaa        trcar
                rola
                bcc         lpcg1
                ldaa        trcar
                ldaa        rdr
                staa        dffm
                jsr         chkcpt
                ldaa        trcar
                ldab        dffm
                stab        tdr
*
lpcg2           clc
                ldaa        trcar
                rola
                bcc         lpcg2
                ldaa        trcar
                ldaa        rdr
                staa        setup1
                jsr         chkcpt
                ldaa        trcar
                ldab        setup1
```

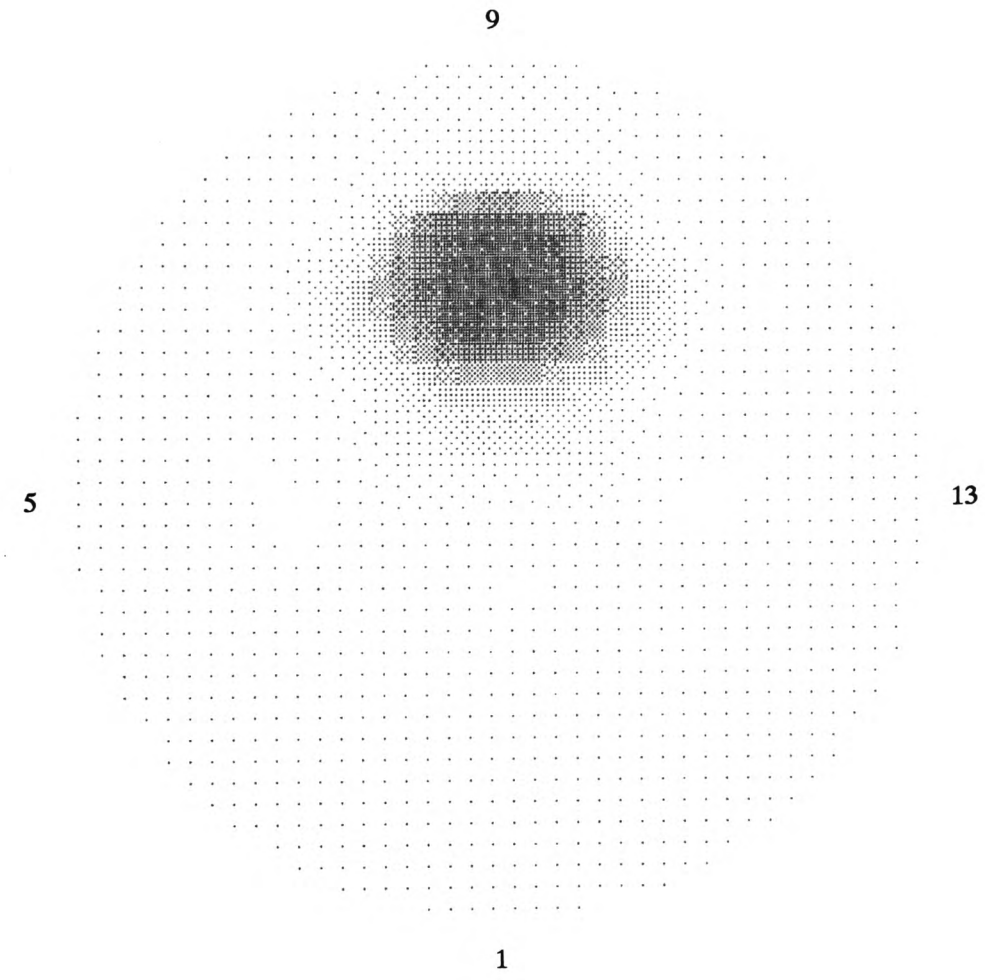
```
*               stab            tdr
*
                clr             setatt
                ldx             #first          ;set first return address
                stx             rtnadd
                ldaa            setup1
                staa            m4
                ldx             #atcode         ;load first att. factor code to x
                ldaa            #h'10'
                staa            drpost
*
detect          clc                             ;detect R pulse from ECG
                ldaa            dr1
                rora
                bcs             detect
*
cgi1            ldaa            drpost
                staa            m0
                jsr             record
                dec             drpost
                bne             cgi1
*
                ldx             #atcode         ;load first att. factor code to x
                ldaa            #h'10'
                staa            drpost
cgi2            ldaa            drpost
                staa            m0
                jsr             record
                dec             drpost
                bne             cgi2
*
nextcg          ldx             #atcode
                ldaa            #h'10'
                staa            drpost
                dec             diffm
                bne             detect
*
endcgi          ldx             rtnadd
                stx             enaddr          ;save the last+1 data storage address
                ldab            #fixhopt
                jsr             chkcpt
                ldaa            trcsr
                stab            tdr
                ldaa            #h'10'
                staa            m0
                ldaa            atcode+7
                staa            latch
                ldaa            #h'87'          ;changed from 'ed' to '87'
                staa            m1
                rts
*
*****************************************************************************
*
*               Option 7 : SLOW MEASUREMENTS ROUTINE
*
*****************************************************************************
*
routsm          clc
                ldaa            trcsr
                rola
                bcc             routsm
                ldaa            trcsr
                ldaa            rdr
                cmpa            #h'f0'
                beq             ch01
                cmpa            #h'ff'
                beq             endsm
                bra             routsm
*
ch01            ldaa            #h'00'          ;set abort = false
                staa            abort
                ldx             #first          ;set first return address
                stx             rtnadd
                ldx             #atcode         ;load first att. factor code to x
                ldaa            #h'10'
                staa            drpost
smrec           ldaa            drpost
                staa            m0
                bsr             smvolt
                dec             drpost
                bne             smrec
                bra             routsm
*
endsm           ldx             rtnadd
                stx             enaddr          ;save the last+1 data storage address
                ldab            #fixhopt
                jsr             chkcpt
                ldaa            trcsr
                stab            tdr
                ldaa            #h'10'
                staa            m0
                ldaa            atcode+7
                staa            latch
                ldaa            #h'87'          ;changed from 'ed' to '87'
                staa            m1
                rts
*
*****************************************************************************
*
smvolt          ldaa            #h'0d'          ;reset the number of recording signals
                staa            numsmes
                ldaa            drpost          ;check the active driving electrodes
                cmpa            #h'10'
                beq             smlb1
                cmpa            #h'0f'
                beq             smlb2
```

```
                cmpa    #h'0e'
                beq     smlb3
                cmpa    #h'0d'
                beq     smlb4
                cmpa    #h'0c'
                beq     smlb5
                cmpa    #h'0b'
                beq     smjp6
                cmpa    #h'0a'
                beq     smjp7
                cmpa    #h'09'
                beq     smjp8
                cmpa    #h'08'
                beq     smjp9
                cmpa    #h'07'
                beq     smjp10
                cmpa    #h'06'
                beq     smjp11
                cmpa    #h'05'
                beq     smjp12
                cmpa    #h'04'
                beq     smjp13
                cmpa    #h'03'
                beq     smjp14
                cmpa    #h'02'
                beq     smjp15
                cmpa    #h'01'
                beq     smjp16
smjp6           jmp     smlb6
smjp7           jmp     smlb7
smjp8           jmp     smlb8
smjp9           jmp     smlb9
smjp10          jmp     smlb10
smjp11          jmp     smlb11
smjp12          jmp     smlb12
smjp13          jmp     smlb13
smjp14          jmp     smlb14
smjp15          jmp     smlb15
smjp16          jmp     smlb16
smjstp          jmp     smstp
*
smlb1           ldaa    #h'ed'          ;sequences of monitoring position
                jsr     smact           ;select monitoring electrodes and
                dec     nummon          ;carry out the A-D conversion. Upon
                beq     smjstp          ;the completion of 13th conversion,
smlb2           ldaa    #h'dc'          ;go back to alter the position of the
                jsr     smact           ;driving electrode pair.
                dec     nummon
                beq     smjstp
smlb3           ldaa    #h'cb'
                jsr     smact
                dec     nummon
                beq     smjstp
smlb4           ldaa    #h'ba'
                jsr     smact
                dec     nummon
                beq     smjstp
smlb5           ldaa    #h'a9'
                jsr     smact
                dec     nummon
                beq     smstp
smlb6           ldaa    #h'98'
                jsr     smact
                dec     nummon
                beq     smstp
smlb7           ldaa    #h'87'
                jsr     smact
                dec     nummon
                beq     smstp
smlb8           ldaa    #h'76'
                jsr     smact
                dec     nummon
                beq     smstp
smlb9           ldaa    #h'65'
                jsr     smact
                dec     nummon
                beq     smstp
smlb10          ldaa    #h'54'
                jsr     smact
                dec     nummon
                beq     smstp
smlb11          ldaa    #h'43'
                jsr     smact
                dec     nummon
                beq     smstp
smlb12          ldaa    #h'32'
                jsr     smact
                dec     nummon
                beq     smstp
smlb13          ldaa    #h'21'
                jsr     smact
                dec     nummon
                beq     smstp
smlb14          ldaa    #h'10'
                jsr     smact
                dec     nummon
                beq     smstp
smlb15          ldaa    #h'0f'
                jsr     smact
                dec     nummon
                beq     smstp
smlb16          ldaa    #h'fe'
                jsr     smact
                dec     nummon
                beq     smstp
                jmp     smlb1
```

```
smstp       rts
*
*****************************************************************************
*
smact       staa    m1          ;initiate the ADC
            ldaa    x
            staa    latch
            inx                 ;increment att. factor address pointer
            stx     attadd      ;store address pointer
            ldx     rtnadd      ;get actual data address
            ldaa    abort
            cmpa    #h'ff'
            beq     stpact
*
            staa    m3          ;switch to Real part
*
smdlyR      clc                 ;delay controlled by PC
            ldaa    trcsr
            rola
            bcc     smdlyR
            ldaa    trcsr
            ldaa    rdr
            staa    abort
            cmpa    #h'ff'
            beq     stpact
*
            staa    sc
            staa    sc
smchk1      clc                 ;check whether the conversion is
            ldaa    drl         ;finished
            rola
            bcc     smchk1
            ldab    #h'03'      ;allow data to settle down
smdly1      decb
            bne     smdly1
            ldab    ub
            stab    x
            inx
            ldab    lb
            stab    x
            inx
*
            stab    m2          ;switch to Imaginary Part
*
smdly1      clc                 ;delay controlled by PC
            ldaa    trcsr
            rola
            bcc     smdly1
            ldaa    trcsr
            ldaa    rdr
            staa    abort
            cmpa    #h'ff'
            beq     stpact
*
            staa    sc
            staa    sc
smchk2      clc
            ldaa    drl
            rola
            bcc     smchk2
            ldab    #h'03'
smdly2      decb
            bne     smdly2
            ldaa    ub
            staa    x
            inx
            ldaa    lb
            staa    x
            inx
            stx     rtnadd      ;store actual address pointer
            ldx     attadd      ;get att. factor address
stpact      rts
*
*****************************************************************************
*
*           SET UP ADDRESS POINTER FOR RESET
*
*****************************************************************************
*
            org     h'fffe'     ;set RESET vectoring address
            acon    start
*
            end
```

Appendix F Theoretical Data of the Two-dimensional Resistor-mesh Phantom (SMT)





A sample reconstructed conductivity image for m = 1.5

Potentials in μV for 1mA drive
Channel n = electrodes n, n-1

Without plug board

Drive channel	Receive channel															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	11427	3913	1419	1089	1413	1211	716	734	1327	1688	1419	2026	6969	0
2	0	0	0	6969	2026	1419	1689	1327	734	716	1211	1412	1089	1419	3913	11427
3	11433	0	0	0	6970	3914	3929	2651	1327	1212	1893	2002	1413	1690	3929	7700
4	3914	6971	0	0	0	11432	7699	3929	1689	1413	2002	1893	1212	1327	2652	3929
5	1419	2027	6969	0	0	0	11427	3912	1419	1089	1412	1211	716	734	1327	1689
6	1089	1419	3913	11428	0	0	0	6968	2026	1419	1688	1326	734	716	1212	1413
7	1413	1690	3930	7700	11433	0	0	0	6969	3914	3929	2651	1327	1212	1893	2002
8	1212	1327	2652	3929	3914	6971	0	0	0	11432	7699	3929	1689	1413	2002	1893
9	716	734	1327	1689	1419	2027	6969	0	0	0	11427	3912	1419	1089	1413	1211
10	734	716	1211	1413	1089	1418	3912	11426	0	0	0	6969	2026	1419	1689	1327
11	1328	1211	1893	2002	1413	1688	3928	7698	11431	0	0	0	6971	3914	3929	2652
12	1689	1413	2002	1893	1212	1327	2651	3929	3914	6969	0	0	0	11433	7700	3929
13	1419	1089	1413	1212	716	734	1326	1689	1419	2025	6968	0	0	0	11428	3913
14	2026	1419	1689	1327	734	716	1211	1413	1089	1418	3912	11426	0	0	0	6969
15	6971	3914	3929	2652	1327	1211	1893	2002	1413	1688	3928	7698	11431	0	0	0
16	0	11433	7700	3929	1689	1413	2002	1893	1212	1327	2651	3928	3913	6970	0	0

With plug board

Drive channel	Receive channel															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	11432	3918	1422	1093	1418	1204	696	715	1321	1695	1424	2030	6974	0
2	0	0	0	6974	2030	1424	1695	1321	715	696	1204	1418	1093	1422	3917	11431
3	11436	0	0	0	6977	3923	3944	2648	1296	1176	1875	2008	1419	1694	3937	7706
4	3919	6974	0	0	0	11444	7721	3933	1658	1368	1974	1895	1217	1332	2659	3937
5	1422	2029	6975	0	0	0	11447	3922	1399	1055	1386	1209	719	737	1332	1694
6	1093	1423	3920	11437	0	0	0	6989	2007	1376	1650	1319	735	718	1216	1418
7	1418	1696	3944	7722	11452	0	0	0	6965	3821	3815	2610	1319	1210	1895	2008
8	1204	1322	2647	3933	3923	6991	0	0	0	11372	7508	3815	1650	1386	1974	1874
9	696	715	1296	1657	1399	2008	6965	0	0	0	11366	3819	1375	1055	1368	1175
10	715	696	1175	1368	1055	1375	3819	11366	0	0	0	6965	2007	1399	1657	1296
11	1322	1204	1874	1974	1387	1649	3814	7507	11371	0	0	0	6991	3924	3933	2648
12	1696	1418	2008	1895	1210	1319	2611	3815	3821	6965	0	0	0	11452	7721	3944
13	1423	1093	1418	1216	718	735	1319	1650	1376	2007	6989	0	0	0	11438	3921
14	2030	1423	1694	1332	737	718	1209	1386	1055	1399	3922	11447	0	0	0	6975
15	6975	3919	3937	2659	1332	1217	1895	1974	1369	1658	3933	7721	11444	0	0	0
16	0	11437	7707	3937	1695	1419	2008	1875	1176	1296	2647	3944	3923	6977	0	0

STATISTICAL ANALYSES (1% tolerance of each resistor; Mean is in V; S.D. is in mV)

	Homogenuous (V_1)		Inhomogenuous (V_2)		Homogenuous (V_1)		Inhomogenuous (V_2)	
	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)
	Active electrodes 16/1				Active electrodes 1/2			
1	0.0687	0.0782	0.0686	0.0784	-0.0686	0.0845	-0.0687	0.0748
2	0.0208	0.0272	0.0208	0.0252	0.0673	0.08	0.0673	0.0901
3	0.00938	0.0166	0.00938	0.0164	0.0145	0.0239	0.0145	0.0276
4	0.00547	0.015	0.00546	0.015	0.00757	0.0195	0.00756	0.0234
5	0.00405	0.0146	0.00404	0.0143	0.00554	0.0184	0.00553	0.0227
6	0.00296	0.0145	0.00295	0.014	0.00413	0.0182	0.00411	0.0218
7	0.00155	0.0144	0.00153	0.0138	0.00244	0.018	0.00241	0.0209
8	0.000339	0.0145	0.000323	0.0137	0.00111	0.0177	0.00109	0.0206
9	-0.00037	0.0145	-0.00037	0.0136	0.000376	0.0176	0.000376	0.0205
10	-0.00111	0.0146	-0.00109	0.0136	-0.00034	0.0176	-0.00032	0.0204
11	-0.00244	0.0148	-0.00241	0.0137	-0.00155	0.0178	-0.00152	0.0202
12	-0.00413	0.0152	-0.0041	0.0141	-0.00296	0.018	-0.00294	0.0201
13	-0.00555	0.0157	-0.00553	0.0144	-0.00405	0.0182	-0.00403	0.0205
14	-0.00757	0.017	-0.00756	0.0154	-0.00547	0.0187	-0.00546	0.0205
15	-0.0145	0.023	-0.0145	0.0205	-0.00938	0.0204	-0.00937	0.0213
16	-0.0673	0.0754	-0.0673	0.0798	-0.0208	0.0305	-0.0208	0.0303
	Active electrodes 2/3				Active electrodes 3/4			
1	-0.0291	0.0484	-0.0291	0.0563	-0.014	0.0808	-0.014	0.0768
2	-0.0819	0.0917	-0.0819	0.107	-0.021	0.0827	-0.021	0.077
3	0.0969	0.0886	0.0969	0.105	-0.0969	0.117	-0.0969	0.112
4	0.021	0.0491	0.021	0.0549	0.0819	0.1	0.0819	0.121
5	0.014	0.0456	0.014	0.0525	0.0291	0.0825	0.029	0.0848
6	0.0101	0.0452	0.0101	0.0516	0.0177	0.0821	0.0176	0.0803
7	0.00617	0.0443	0.00613	0.0508	0.00995	0.0809	0.00987	0.0782
8	0.00352	0.0437	0.00348	0.0505	0.00602	0.0805	0.00594	0.0775
9	0.00219	0.0433	0.00219	0.0503	0.00433	0.0806	0.00428	0.0771
10	0.000978	0.0433	0.00101	0.0503	0.00292	0.0808	0.00291	0.0772
11	-0.00091	0.043	-0.00086	0.0501	0.000915	0.0807	0.000938	0.077
12	-0.00292	0.0431	-0.00287	0.0502	-0.00097	0.0803	-0.00095	0.0768
13	-0.00433	0.0429	-0.00429	0.0503	-0.00219	0.0803	-0.00217	0.0768
14	-0.00602	0.043	-0.00599	0.0503	-0.00352	0.0803	-0.00351	0.0766
15	-0.00995	0.0434	-0.00993	0.0507	-0.00617	0.0807	-0.00617	0.0766
16	-0.0177	0.0435	-0.0176	0.0529	-0.0101	0.0804	-0.0101	0.077
	Active electrodes 4/5				Active electrodes 5/6			
1	-0.00554	0.0765	-0.00553	0.0685	-0.00405	0.0587	-0.00408	0.0487
2	-0.00756	0.0768	-0.00756	0.0687	-0.00547	0.059	-0.0055	0.0488
3	-0.0145	0.0775	-0.0145	0.0705	-0.00938	0.0597	-0.00942	0.0499
4	-0.0673	0.104	-0.0673	0.098	-0.0208	0.0627	-0.0209	0.0556
5	0.0687	0.105	0.0686	0.11	-0.0686	0.0784	-0.0687	0.0955
6	0.0208	0.0811	0.0208	0.0738	0.0673	0.0852	0.0673	0.0894
7	0.00939	0.0784	0.00934	0.0691	0.0145	0.06	0.0144	0.0488
8	0.00548	0.0772	0.00542	0.0683	0.00758	0.0586	0.00745	0.0476
9	0.00406	0.0769	0.00402	0.0683	0.00555	0.0585	0.00545	0.0475
10	0.00297	0.0768	0.00297	0.0681	0.00413	0.0586	0.00407	0.0477
11	0.00156	0.0766	0.00158	0.0682	0.00244	0.0584	0.00242	0.0477
12	0.000347	0.0767	0.000372	0.068	0.00112	0.0585	0.0011	0.0479
13	-0.00036	0.0766	-0.00034	0.068	0.000381	0.0584	0.000369	0.0479
14	-0.0011	0.0767	-0.00108	0.0681	-0.00033	0.0584	-0.00034	0.0482
15	-0.00243	0.0764	-0.00241	0.0681	-0.00155	0.0585	-0.00157	0.0482
16	-0.00412	0.0765	-0.00411	0.0683	-0.00296	0.0586	-0.00298	0.0485

STATISTICAL ANALYSES (1% tolerance of each resistor; Mean is in V; S.D. is in mV)

	Homogenuous (V_1)		Inhomogenuous (V_2)		Homogenuous (V_1)		Inhomogenuous (V_2)	
	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)
	Active electrodes 6/7				Active electrodes 7/8			
1	-0.00434	0.0929	-0.00434	0.0816	-0.00221	0.05	-0.00227	0.0558
2	-0.00603	0.0935	-0.00604	0.0818	-0.00354	0.0497	-0.00359	0.0559
3	-0.00996	0.0932	-0.00999	0.0814	-0.00619	0.0495	-0.00624	0.0557
4	-0.0177	0.0959	-0.0177	0.0813	-0.0101	0.0488	-0.0102	0.0569
5	-0.0291	0.0994	-0.0292	0.083	-0.014	0.0497	-0.0141	0.058
6	-0.0819	0.133	-0.082	0.121	-0.021	0.0513	-0.0211	0.0604
7	0.0969	0.132	0.0967	0.127	-0.0969	0.105	-0.0971	0.0961
8	0.021	0.0941	0.0207	0.0837	0.0819	0.0969	0.0815	0.102
9	0.014	0.0938	0.0138	0.0839	0.0291	0.0573	0.0285	0.0628
10	0.0101	0.0936	0.00993	0.0833	0.0176	0.0541	0.0172	0.0583
11	0.00616	0.093	0.00612	0.0826	0.00993	0.0518	0.00964	0.0566
12	0.00351	0.093	0.00351	0.0822	0.006	0.0511	0.00582	0.0563
13	0.00218	0.093	0.00219	0.0821	0.00431	0.0507	0.00417	0.0558
14	0.000973	0.0931	0.000978	0.0822	0.0029	0.0506	0.00279	0.0557
15	-0.00092	0.0931	-0.00091	0.082	0.000895	0.0504	0.000813	0.0557
	Active electrodes 8/9				Active electrodes 9/10			
1	-0.00037	0.0219	-0.00039	0.0219	0.000375	0.0176	0.000397	0.0184
2	-0.00111	0.0219	-0.00111	0.022	-0.00034	0.0177	-0.00029	0.0183
3	-0.00244	0.0219	-0.00241	0.022	-0.00155	0.0179	-0.00147	0.0185
4	-0.00412	0.0224	-0.00406	0.0223	-0.00297	0.0179	-0.00284	0.0186
5	-0.00554	0.0227	-0.00546	0.0226	-0.00406	0.0184	-0.0039	0.0186
6	-0.00757	0.0235	-0.00747	0.0228	-0.00547	0.0186	-0.00527	0.0189
7	-0.0145	0.0271	-0.0144	0.0267	-0.00939	0.0208	-0.00909	0.0209
8	-0.0673	0.0757	-0.0674	0.0732	-0.0208	0.0312	-0.0205	0.0305
9	0.0687	0.0824	0.0684	0.0862	-0.0687	0.0792	-0.0684	0.0725
10	0.0208	0.0354	0.0205	0.0336	0.0673	0.0887	0.0674	0.0829
11	0.00939	0.0251	0.00909	0.0246	0.0145	0.0233	0.0144	0.0244
12	0.00547	0.0228	0.00527	0.0231	0.00757	0.019	0.00747	0.0205
13	0.00405	0.0223	0.0039	0.0229	0.00554	0.0183	0.00546	0.0199
14	0.00297	0.0221	0.00284	0.0225	0.00412	0.018	0.00406	0.0194
15	0.00155	0.0219	0.00148	0.0221	0.00244	0.0177	0.00241	0.019
16	0.000341	0.0219	0.0003	0.022	0.00111	0.0177	0.00111	0.0185
	Active electrodes 10/11				Active electrodes 11/12			
1	0.0022	0.0205	0.00225	0.0218	0.00432	0.0194	0.00435	0.0206
2	0.000991	0.0202	0.00105	0.0219	0.00291	0.0196	0.00294	0.0208
3	-0.00090	0.0199	-0.00082	0.0221	0.000907	0.0196	0.000927	0.0207
4	-0.00291	0.0196	-0.0028	0.0223	-0.00098	0.0199	-0.00096	0.021
5	-0.00432	0.0198	-0.00418	0.0224	-0.0022	0.0201	-0.00218	0.0212
6	-0.00601	0.0203	-0.00583	0.0227	-0.00352	0.0206	-0.0035	0.0214
7	-0.00994	0.0214	-0.00965	0.0229	-0.00618	0.0214	-0.00611	0.0222
8	-0.0176	0.0247	-0.0172	0.0277	-0.0101	0.0231	-0.00993	0.0245
9	-0.0291	0.0343	-0.0285	0.0356	-0.014	0.0256	-0.0137	0.0254
10	-0.0819	0.0812	-0.0815	0.0923	-0.021	0.0299	-0.0207	0.0302
11	0.0969	0.0893	0.0971	0.0841	-0.0969	0.0854	-0.0967	0.0991
12	0.021	0.0288	0.0211	0.028	0.0819	0.0831	0.082	0.0806
13	0.014	0.0246	0.0141	0.0239	0.0291	0.0311	0.0292	0.0329
14	0.0101	0.0226	0.0102	0.0227	0.0176	0.0236	0.0177	0.0234
15	0.00618	0.0215	0.00622	0.022	0.00994	0.0202	0.00999	0.0209
16	0.00353	0.0206	0.00358	0.0214	0.00601	0.0194	0.00605	0.0204

STATISTICAL ANALYSES (1% tolerance of each resistor; Mean is in V; S.D. is in mV)

	Homogenuous (V_1)		Inhomogenuous (V_2)		Homogenuous (V_1)		Inhomogenuous (V_2)	
	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)	Mean	S.D.(×0.1%)
	Active electrodes 12/13				Active electrodes 13/14			
1	0.00405	0.0118	0.00406	0.0117	0.00555	0.0143	0.00555	0.0137
2	0.00296	0.0116	0.00297	0.0113	0.00413	0.0135	0.00413	0.0129
3	0.00155	0.0111	0.00155	0.0106	0.00244	0.0128	0.00243	0.0124
4	0.00034	0.0109	0.000336	0.0103	0.00111	0.0125	0.0011	0.0123
5	-0.00037	0.011	-0.00038	0.0102	0.000377	0.0125	0.000365	0.0122
6	-0.00111	0.011	-0.00112	0.0102	-0.00033	0.0123	-0.00035	0.0123
7	-0.00244	0.0111	-0.00244	0.0103	-0.00155	0.0124	-0.00156	0.0124
8	-0.00413	0.012	-0.00408	0.0106	-0.00296	0.0128	-0.00295	0.0129
9	-0.00554	0.0125	-0.00546	0.0109	-0.00405	0.0131	-0.004	0.0133
10	-0.00757	0.0135	-0.00747	0.0123	-0.00547	0.0136	-0.0054	0.0139
11	-0.0145	0.0179	-0.0145	0.0191	-0.00938	0.0164	-0.00932	0.0172
12	-0.0674	0.0821	-0.0673	0.0771	-0.0208	0.0275	-0.0208	0.0287
13	0.0687	0.0838	0.0687	0.082	-0.0687	0.0768	-0.0686	0.0878
14	0.0208	0.0235	0.0208	0.0288	0.0673	0.075	0.0674	0.0859
15	0.00938	0.0144	0.00941	0.0153	0.0145	0.0212	0.0146	0.0208
16	0.00547	0.0124	0.00549	0.0127	0.00757	0.0158	0.00758	0.0147
	Active electrodes 14/15				Active electrodes 15/16			
1	0.014	0.0183	0.014	0.0207	0.0291	0.0319	0.0291	0.035
2	0.0101	0.0177	0.0101	0.0184	0.0176	0.0232	0.0176	0.023
3	0.00618	0.0164	0.00617	0.0171	0.00994	0.018	0.00993	0.0195
4	0.00353	0.0162	0.00351	0.0173	0.00601	0.0171	0.00599	0.0182
5	0.0022	0.0161	0.00218	0.0171	0.00432	0.0169	0.0043	0.0176
6	0.00099	0.0158	0.000963	0.0171	0.00291	0.0164	0.00288	0.0175
7	-0.00090	0.0161	-0.00093	0.0172	0.000908	0.0161	0.00087	0.0176
8	-0.00291	0.0165	-0.00291	0.0177	-0.00098	0.0163	-0.00101	0.0177
9	-0.00432	0.0169	-0.00428	0.0178	-0.0022	0.0162	-0.00218	0.0176
10	-0.00601	0.017	-0.00593	0.018	-0.00352	0.0163	-0.00348	0.0178
11	-0.00994	0.0186	-0.00987	0.0193	-0.00618	0.0168	-0.00613	0.0182
12	-0.0176	0.0225	-0.0176	0.0238	-0.0101	0.0179	-0.0101	0.0193
13	-0.0291	0.0326	-0.029	0.0345	-0.014	0.0198	-0.014	0.0213
14	-0.0819	0.078	-0.0818	0.078	-0.021	0.0245	-0.021	0.025
15	0.0969	0.0909	0.0969	0.0971	-0.0969	0.0878	-0.0969	0.0804
16	0.021	0.0237	0.021	0.0254	0.0819	0.0741	0.0819	0.0814

Appendix G Circuit Diagrams of the Polytechnic of Wales System

Page

G.1 Microprocessor controller 212

G.2 A-to-D conversion circuit 213

G.3 Low-pass filtering of drive signals circuit 214

G.4 Amplitude and frequency selection circuit 215

G.5 Voltage driver circuit configuration 216

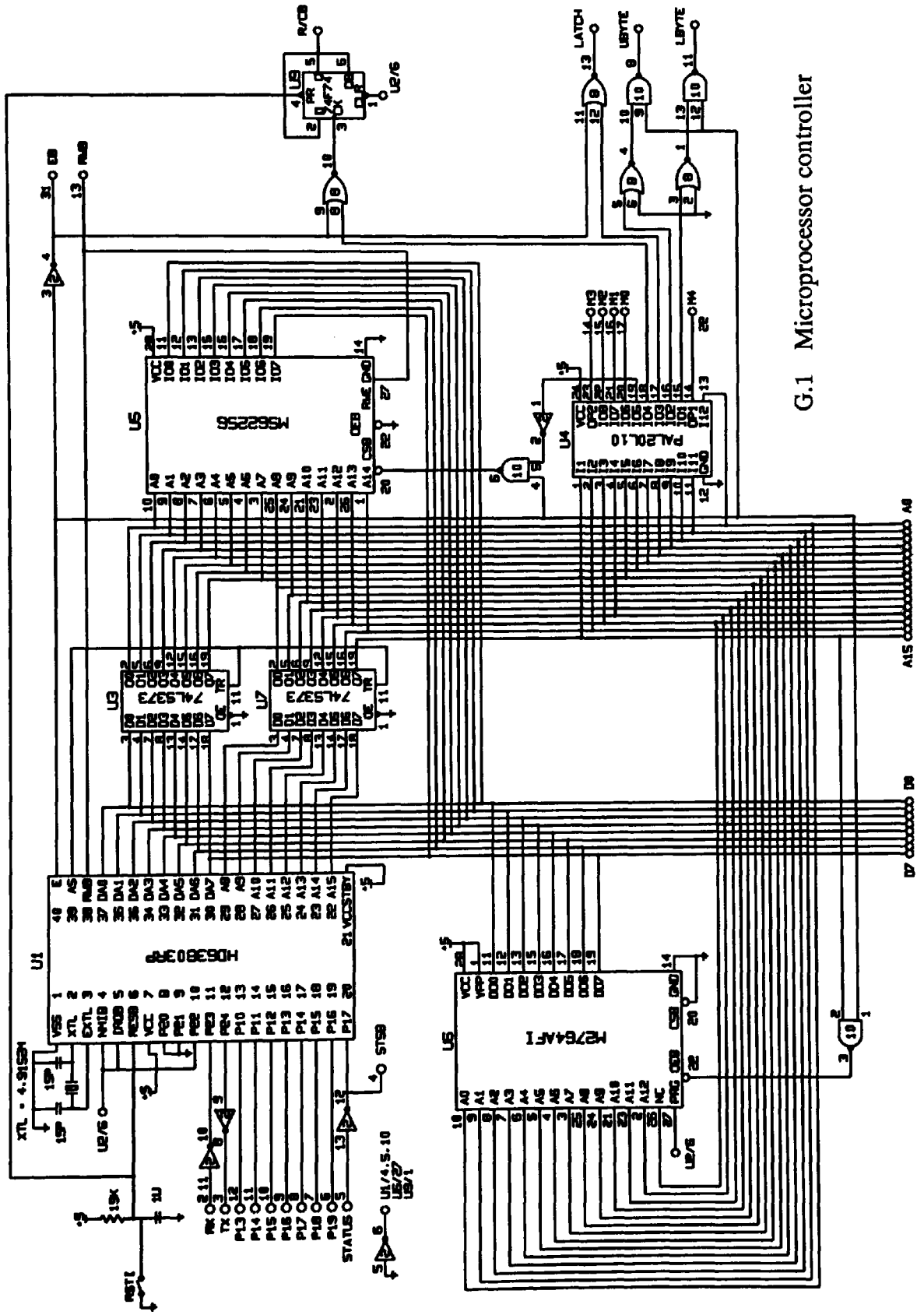
G.6 Measurement circuit configuration 217

G.7 Digital isolation circuitry 218

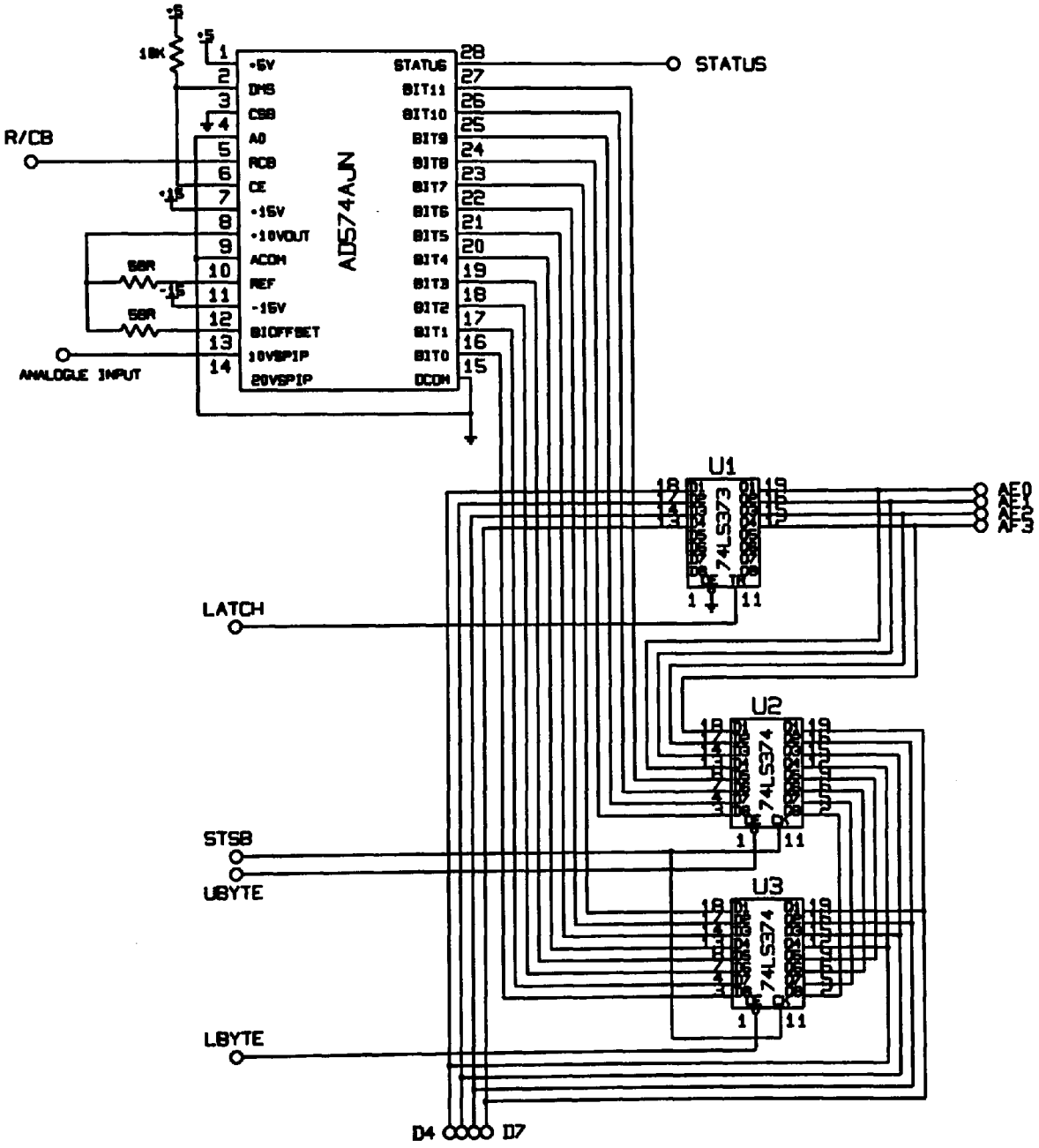
G.8 Attenuation, band-pass filtering and amplification circuit 219

G.9 Demodulation circuit 220

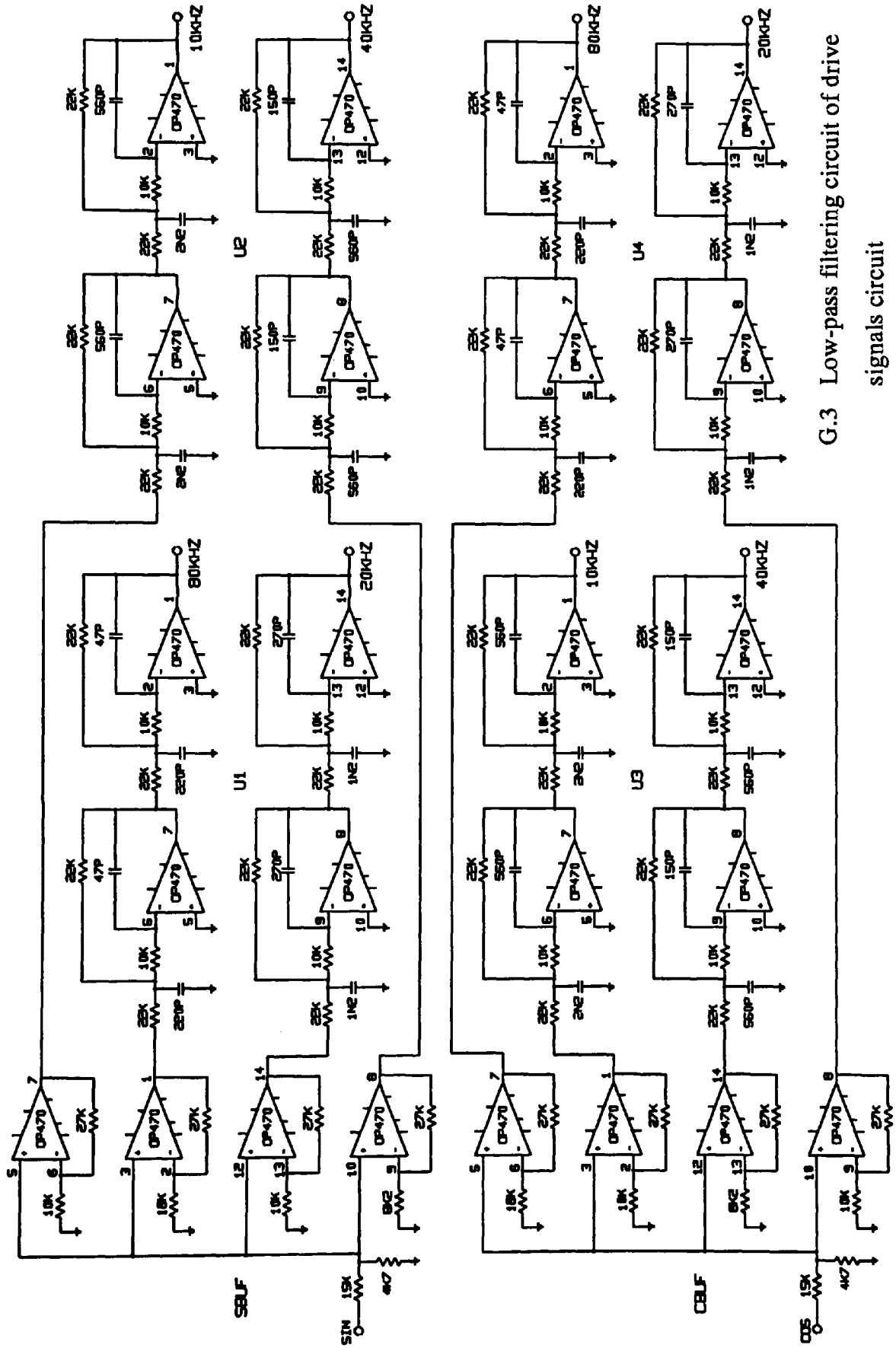
G.10 Current source and screen driver 221



G.1 Microprocessor controller

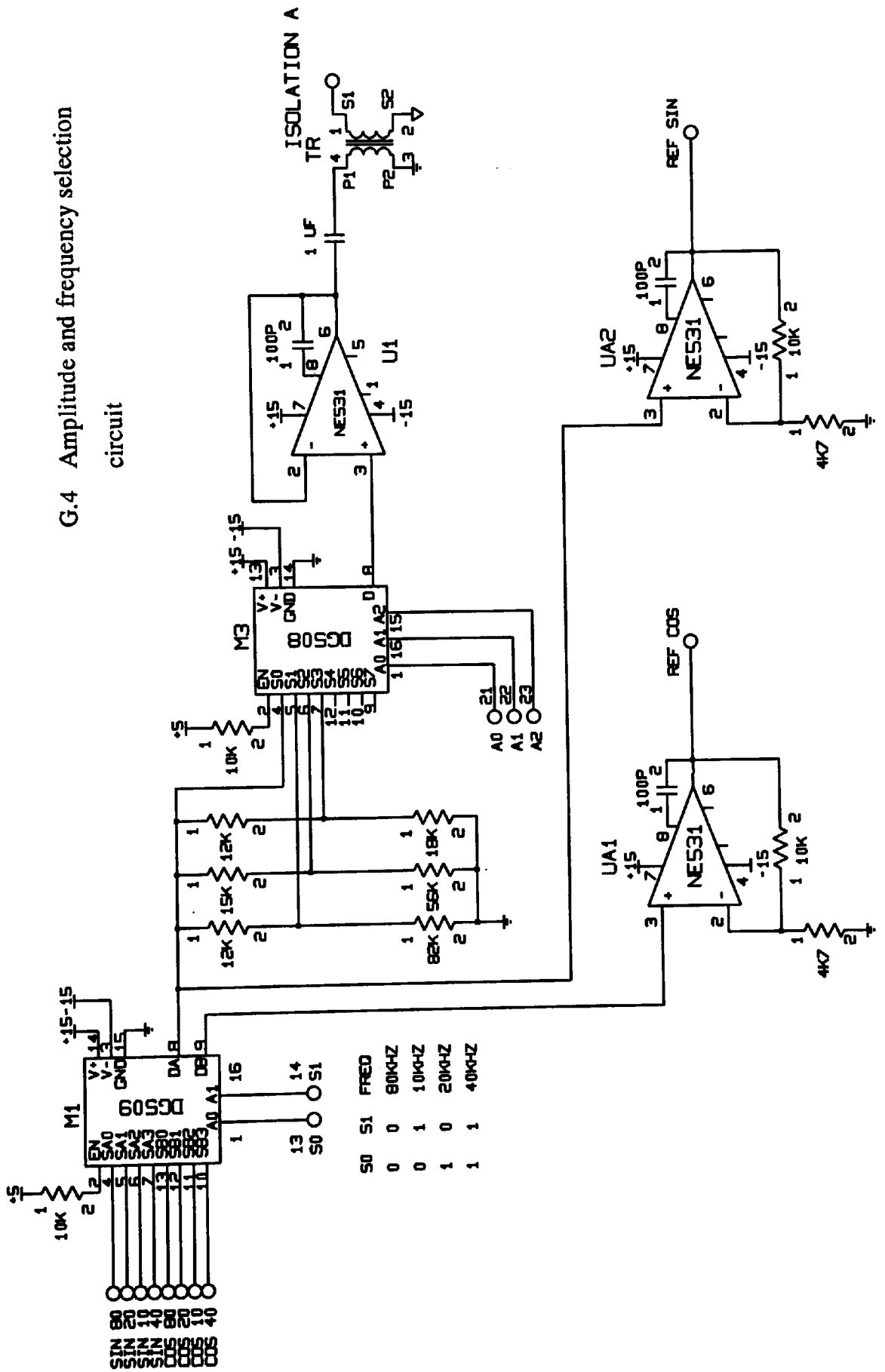


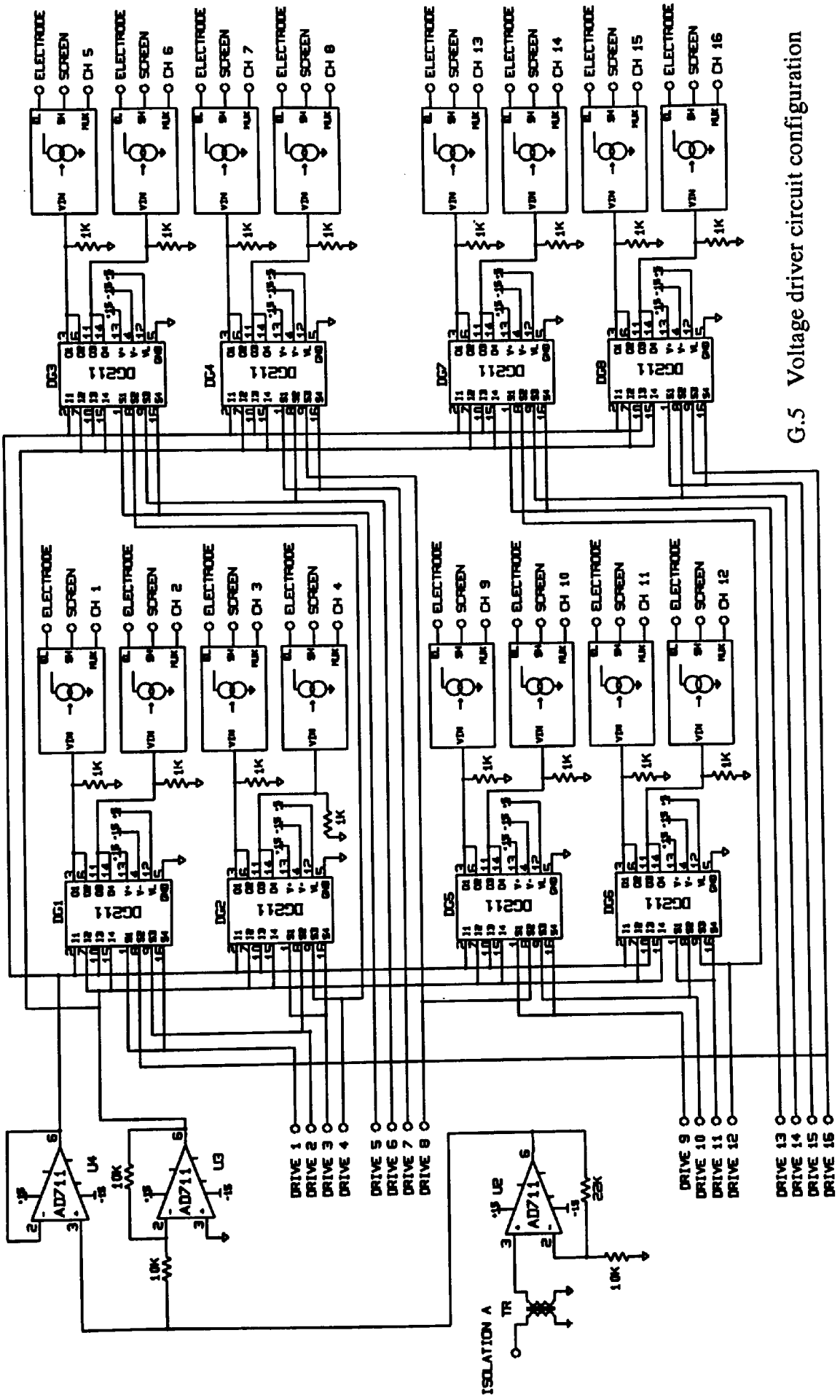
G.2 A-to-D conversion circuit



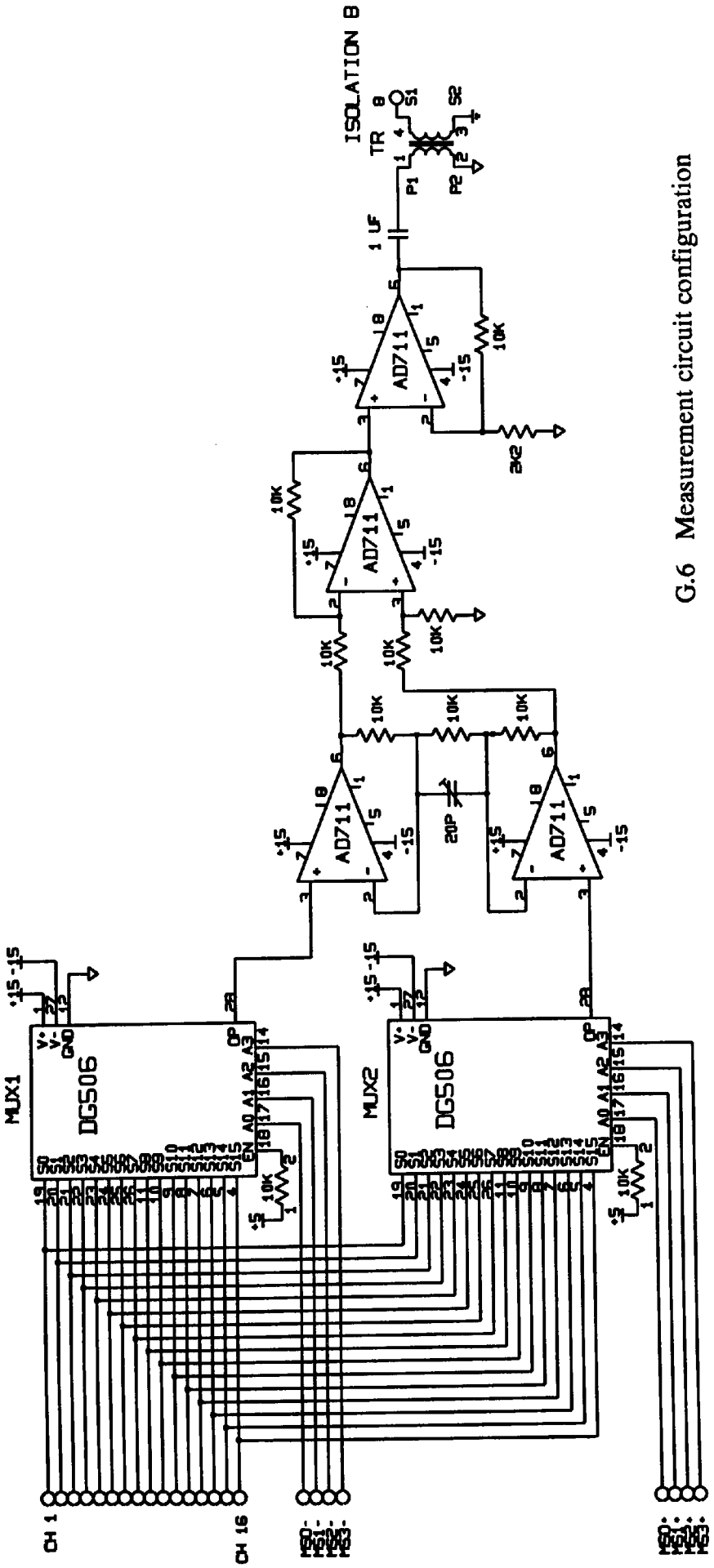
G.3 Low-pass filtering circuit of drive signals circuit

G.4 Amplitude and frequency selection circuit

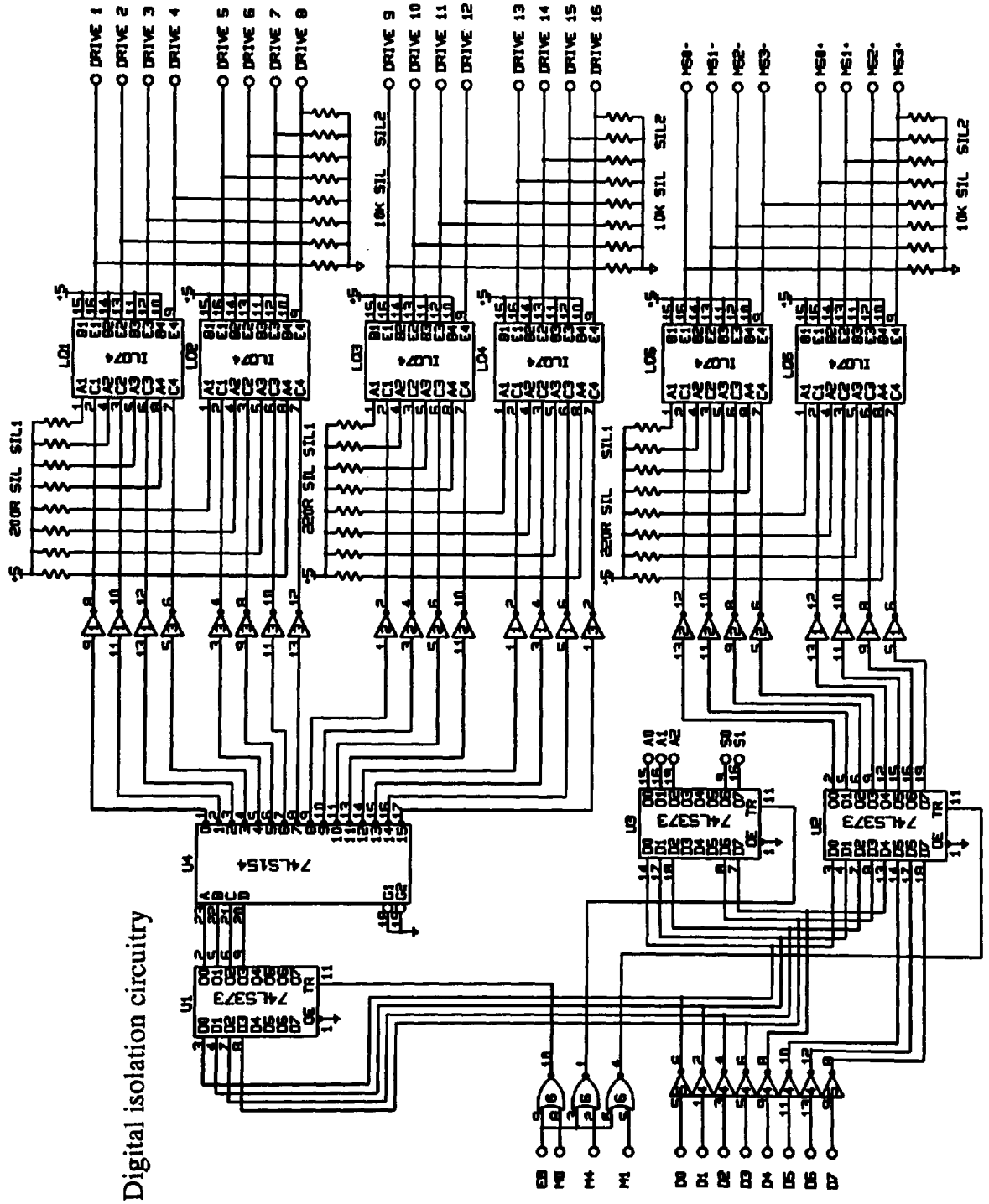




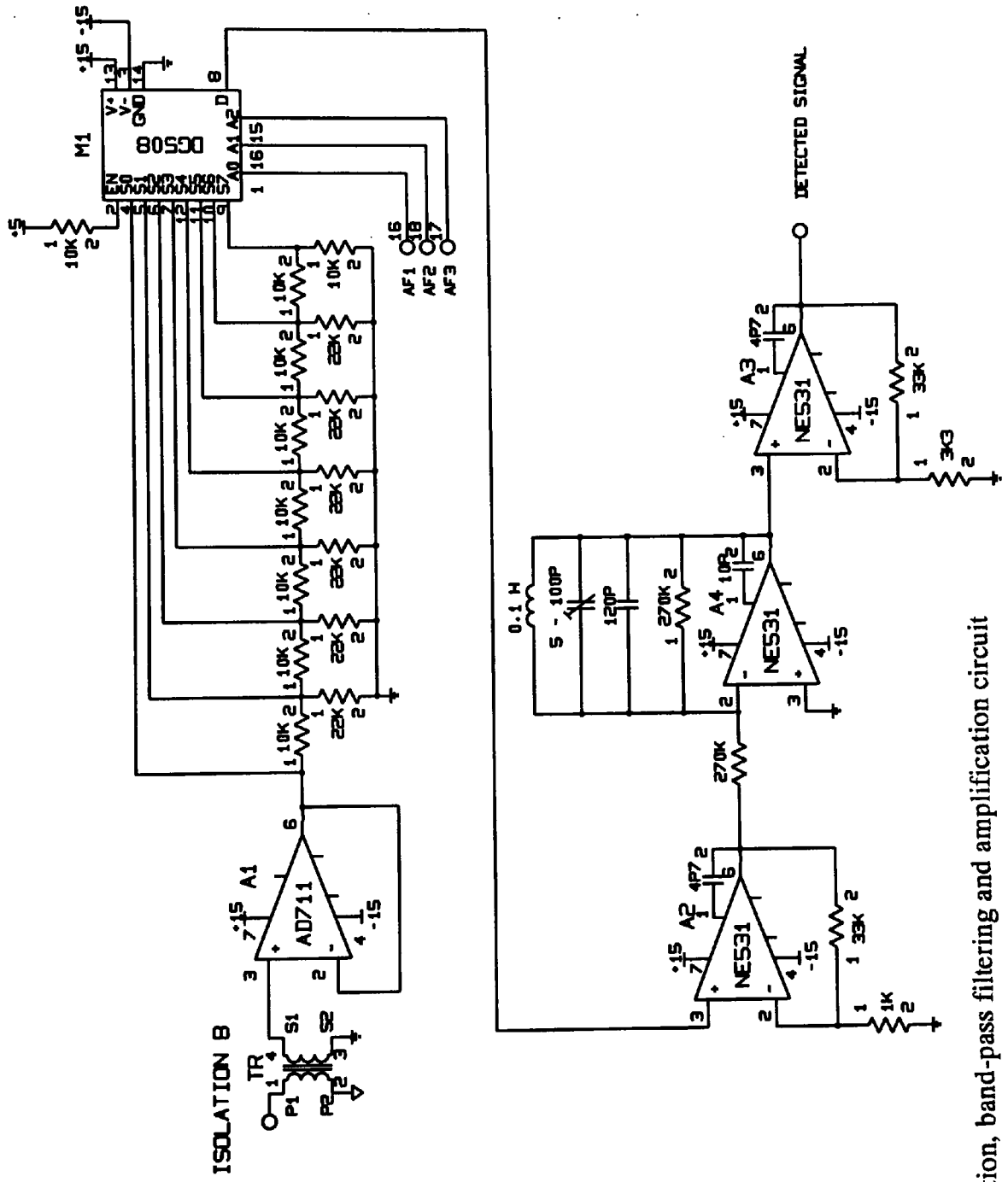
G.5 Voltage driver circuit configuration



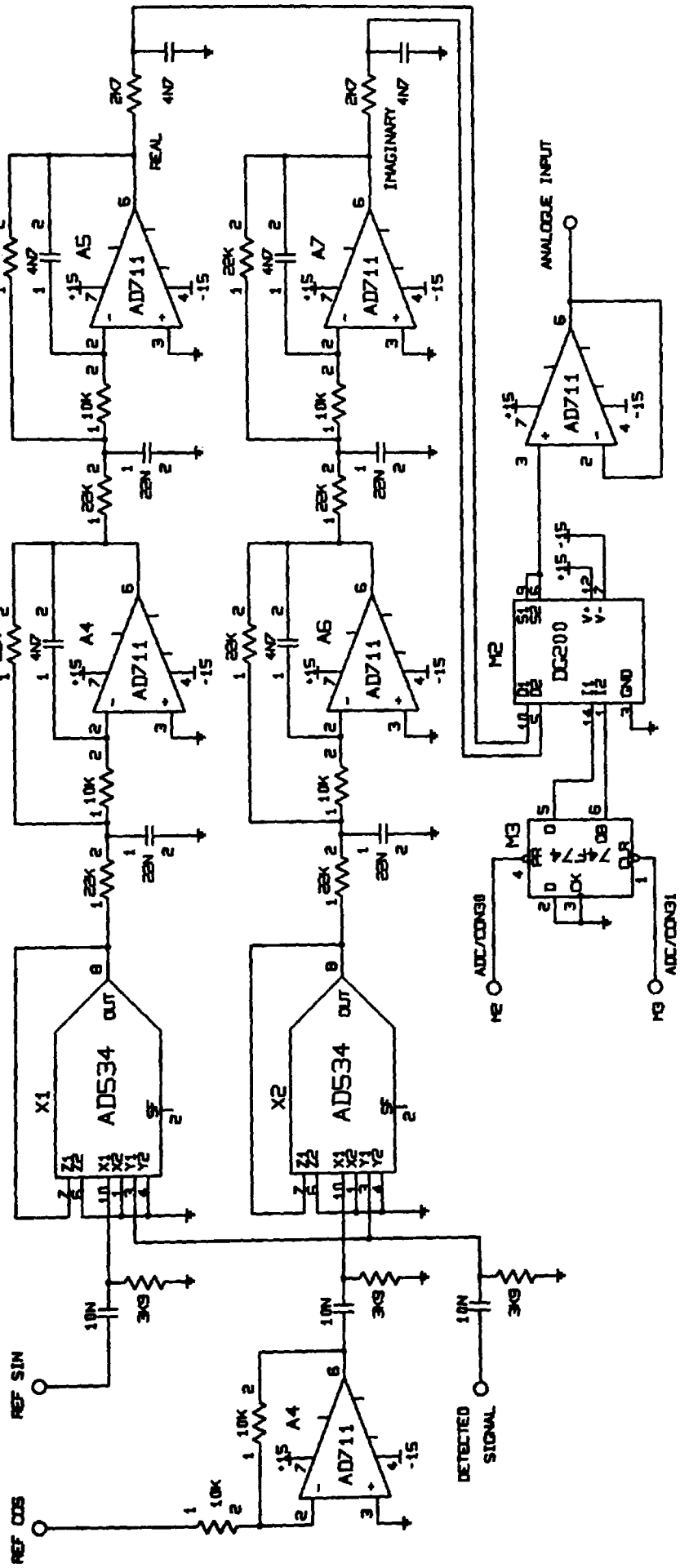
G.6 Measurement circuit configuration



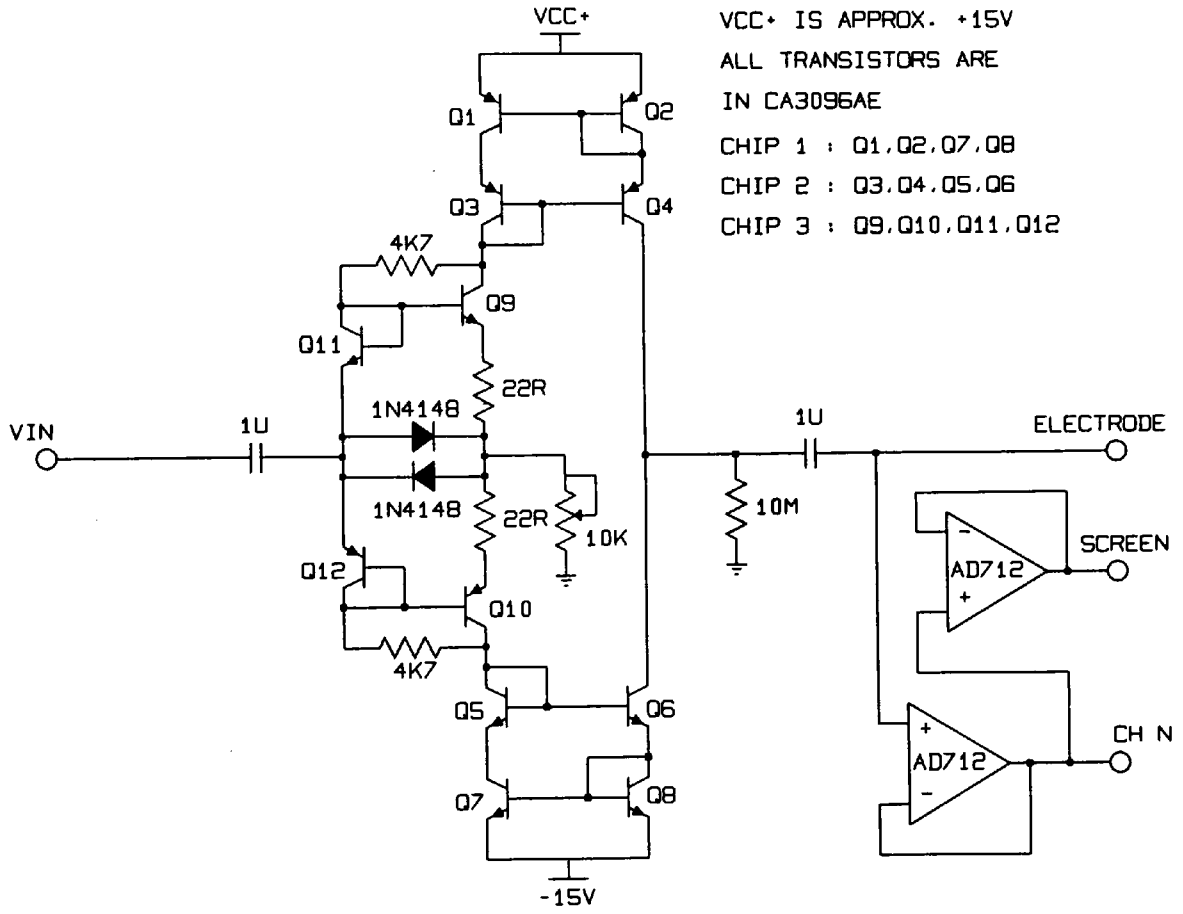
G.7 Digital isolation circuitry



G.8 Attenuation, band-pass filtering and amplification circuit



G.9 Demodulation circuit



G.10 Current source and screen driver